

საქართველოს ტექნიკური უნივერსიტეტი

ჯემალ გრიგალაშვილი



პროექტების კრებული კონტროლერებში

არდუინოს დაპროგრამება ვიზუალურ გარემოში

1 ნაწილი

დამტკიცებულია სახელმძღვანელოდ
სტუ-ს სასწავლო და სამეცნიერო
ლიტერატურის სარედაქციო-საგამომცემლო
საბჭოს მიერ 2020, ოქმი №

უაკ 681.3.06

როგორც ცნობილია, არდუინო ეს არის რამოდენიმე ტექნოლოგიის ნაკრები, რომელთა დახმარებითაც შესაძლებელია სხვადასხვა სახის ჭკვიანი გაჯეტების, ავტომატიკისა და რობოტოტექნიკის მოწყობილობების აგება. იგი შექმნილია უნივერსიტეტის სტუდენტურ გარემოში, შესაბამისად მისი პლატფორმა ღია და მნიშვნელოვნად ხელმისაწვდომია, ამიტომ სარგებლობს უზარმაზარი პოპულარობით როგორც დამწყებებში, ასევე პროფესიონალებში. მაგრამ, აქ საყურადღებოა ის გარემოება, რომ არდუინოს მოწყობილობების დაპროგრამების გარემო Processing Wiring დაფუძნებულია C/C++ ენაზე. ეს ენა კი შექმნილია არა ტექნიკური სისტემების დაპროგრამებისათვის, არამედ რთული პროგრამული პროდუქტების ოპერაციული სისტემების, მონაცემთა ბაზების, საოფისე პროგრამების, ელექტრონული თამაშების შექმნისთვის და მისი გამოყენება არდუინოს მოწყობილობების დაპროგრამებისათვის არაეფექტურია.

გარდა ამისა, როგორც ცნობილია არდუინოს ძირითადი გამომყენებლები პროფესიონალ პროგრამისტები კი არ არიან, არამედ ინჟინრერ ელექტრონიკოსები, ელექტრიკოსები, სქემოტექნიკოსები, სკადა სისტემების სპეციალისტები, რადიომოყვარულები, გამომგონებლები, რომლებსაც არ აქვთ საფუძვლიანი განათლება ტექსტურ პროგრამირებაში, არამედ ისინი ტექნიკური სისტემების შექმნისას ორიენტირებულნი არიან გრაფიკული ენების გამოყენებაზე. ამ შეუსაბამობამ, არდუინოს შემთხვევაში, გამოიწვია მთელი რიგი გარაფიკული/ვიზუალური პროგრამების გარემოების შექმნა, რომელთა შორის გამოირჩევა უფასოდ გავრცელებადი პროგრამა FLProg. *ამ პროგრამის დახმარებით შეიძლება ვიზუალურად დავაპროგრამიროთ არდუინოს კონტროლერი ისე, რომ გამოიყურებოდეს ეს პროცესი, როგორც ელექტრონული ანდა ელექტრული სქემების შექმნა და საერთოდ არ დაგვჭირდეს დაპროგრამირების ტექსტური ენების გამოყენება.*

ამ მიზნის მისაღწევად FLProg იყენებს სამრეწველო კონტროლერების დაპროგრამებისათვის კარგად, ცნობილ საერთაშორისო სტანდარტებით გამყარებულ, გრაფიკულ ენებს FBD და LED, რომლებიც გამოიყენება პრაქტიკულად ყველა სახის ლოგიკური რელეების და სამრეწველო კონტროლერების (პროგრამირებადი ლოგიკური კონტროლერების) დაპროგრამებისათვის მთელ მსოფლიოში.

წინამდებარე სახელმძღვანელო წარმოადგენს პირველ მცდელობას არდუინოს კონტროლერების პროექტების კრებულის შექმნას ვიზუალური დაპროგრამების ენის

გამოყენებით. იგი განკუთვნილია უმაღლესი ტექნიკური სასწავლებლის სტუდენტებისათვის, შედგენილია ამჟამად მოქმედი პროგრამის მიხედვით, რომელიც დახმარებას გაუწევს ინფორმატიკისა და მართვის სისტემების ფაკულტეტის შესაბამისი სპეციალობების ბაკალავრიატისა და მაგისტრატურის სტუდენტებს.

ამ წიგნით შეიძლება ისარგებლონ აგრეთვე უმაღლესი პროფესიული განათლებისა და რობოტოტექნიკისა და ელექტრონიკის სკოლების პედაგოგებმა და მოსწავლეებმა.

რეცენზენტები: კომპიუტერული ინჟინერიის დეპარტამენტის

პროფესორი

მართვის ავტომატიზებული სისტემების დეპარტამენტის

პროფესორი

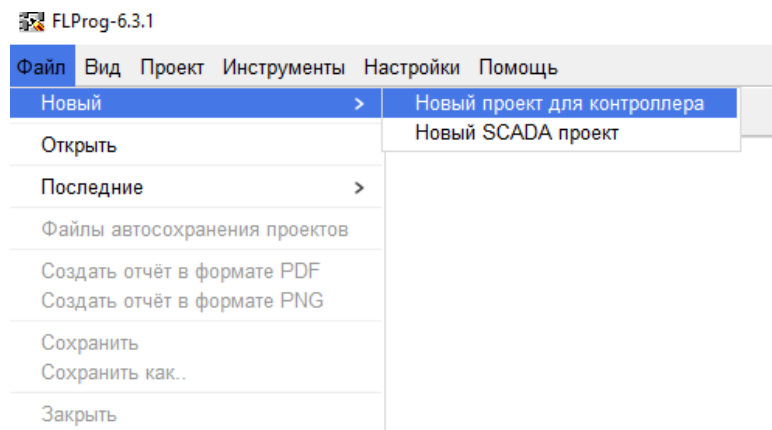
© საგამომცემლი სახლი „ტექნიკური უნივერსიტეტი“, 2020

სარჩევი	4
შესავალი	6
პროექტი_1 მისალმება მსოფლიოს Hello World!	26
პროექტი_2 ბაზური ბლოკების მართვა	33
პროექტი_3 ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი_1	38
პროექტი_4 ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი_2	44
პროექტი_5 ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი_3	47
პროექტი_6 ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი_4	50
პროექტი_7 მასშტაბირების ბლოკი	52
პროექტი_8 ტრიგერების მართვა	57
პროექტი_9 ორობითი მთვლელი	62
პროექტი_10 სიხშირების გამყოფი	66
პროექტი_11 ტაიმერების მართვა	69
პროექტი_12 მთვლელები პარამეტრიზაციის გარეშე	72
პროექტი_13 რევერსიული მთვლელი	78
პროექტი_14 რიცხვების გენერატორი	82
პროექტი_15 მათემატიკა_1	86
პროექტი_16 მათემატიკა_2	91
პროექტი_17 ალგებრა_1	97
პროექტი_18 ალგებრა_2	104
პროექტი_19 შედარების ბლოკები	110
პროექტი_20 გადამრთველები	114
პროექტი_21 სერვო დაბიჯური ძრავების მართვა	123
პროექტი_22 ერთი სერვო ძრავის მართვა	129
პროექტი_23 ოთხი სერვოძრავის მართვა	134
პროექტი_24 რვა სერვოძრავის მართვა	142
პროექტი_25 ორი მუდმივი დენის ძრავის პირდაპირი მართვა	148
პროექტი_26 ერთი მუდმივი დენის ძრავის რევერსული მართვა	151
პროექტი_27 განივ იმპულს მოდულირებული (გიმ) მართვა	156
პროექტი_28 ორი მუდმივი დენის ძრავის მიკროსექმური მართვა	160
პროექტი_29 სერვო ძრავი და ბიჯური ძრავი რევერსით	168
პროექტი_30 საათი და სტრიქონების შეკრება_1	170

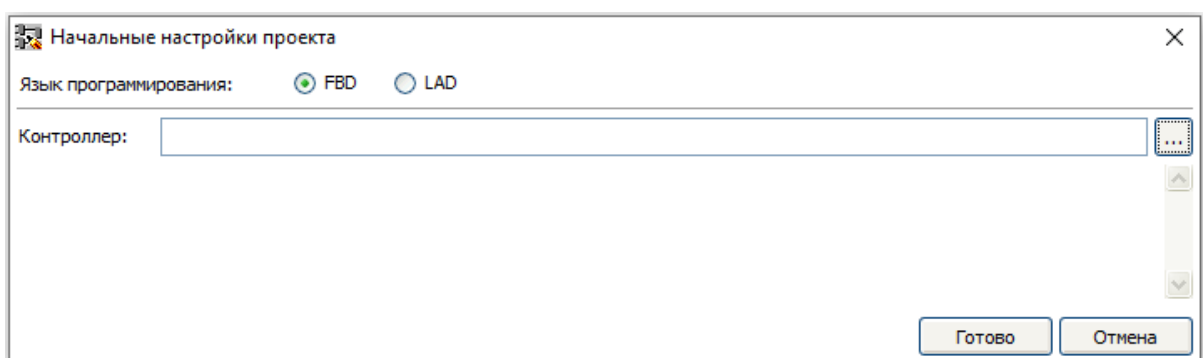
პროექტი_31	საათი და სტრიქონების შეკრება_2	175
პროექტი_32	საათი და სტრიქონების შეკრება_3	179
პროექტი_33	ინფრაწითელი გადამცემისა და მიმღების ტრანზისტორული მართვა	183
პროექტი_34	დო, რე, მი, ფა, სოლ, ლია, სი, დო	187
პროექტი_35	ორი ერთეულოვანი და ერთი RGB შუქდიოდები	190
პროექტი_36	ტემპერატურის (LM35DZ), შეხების და დახრის გადამწოდები	199
პროექტი_37	ტემპერატურული და ფოტორეზისტული გადამწოდები	206
პროექტი_38	ტემპერატურის ციფრული გადამწოდი DS18B20	214
პროექტი_39	ტენიანობის გადამწოდი DH11	220
პროექტი_40	წნევის გადამწოდი [BMP-085]	227
პროექტი_41	ბგერისა და ულტრაბგერის გადამწოდები	233
პროექტი_42	დისპლეი HD44780 ჩიპზე და შვიდსეგმანტა ინდიკატორი	238
პროექტი_43	ულტრაბგერითი (HC-SR04) და რელეური მოდულები	245
პროექტი_44	5 ვოლტის გაზომვა და ძაბვის დონის ინდიკატორი	253
პროექტი_45	ობიექტის შორიდან მართვა ინფრაწითელი პულტით	258
პროექტი_46	სვეტაფორი	268
პროექტი_47	EEPROM	272
პროექტი_48	გირლანდა და ორობითი მთვლეელი	280
პროექტი_49	SD ბარათი	289
პროექტი_50	ეკვალაიზერი	301
პროექტი_51	შუქდიოდური მატრიცა, ნათება ცალკეულ დიაგონალებზე	308
პროექტი_52	შუქდიოდური მატრიცა, ნათება ორივე დიაგონალებზე	316
პროექტი_53	რადიოსიხშირული იდენტიფიკაცია	320
პროექტი_54	ისრიანი თერმომეტრი	328
პროექტი_55	პატრულის ციმციმა	333
პროექტი_56	დეშიფრატორი	337
პროექტი_57	მდგომარეობათა ცხრილი	341
პროექტი_58	გაფართოების მიკროცქემა 74HC595	349
პროექტი_59	K-ტიპის თერმოწყვილის სიგნალის გარდამქმნელი MAX6675	356
პროექტი_60	ფერების გამოცნობის სენსორი TCS230	360
ლიტერატურა	364
დანართი_1	365

შესავალი

დასაწყისში რაც უნდა გავაკეთოთ არის ის, რომ გავუშვათ FLProg-ის პროგრამა, დავაწკაპოთ მთავარი მენიუს ფაილის მენიუზე-Файл, ჩამოშლადი სიიდან ავირჩიოთ ახალი-Новый და გამოჩენილ ფანჯარაში (ფიგ. შ.1) ჯერ ავირჩიოთ ახალი პროექტი კონტროლერისათვის-Новый проект для контроллера, შემდეგ დაპროგრამების ენად-Язык программирования, ავირჩიოთ ფუნქციონალური ბლოკების დიაგრამა-FBD (Function Block Diagram), (ფიგ. შ.2). იმავე ფანჯარაში კონტროლერის ველის მრავალწერტილის ... ღილაკზე თავის მაჩვენებლის მიყვანისა და მარცხენა ღილაკზე თითის დაჭერის (მომავალში ამ პროცედურის მაგივრად ვიხმართ სიტყვას-თითის დაჭერა ან დაწკაპუნება) შედეგად კონტროლერის-Контроллер ჩამოშლად სიაში ვირჩევთ იმ მოდელს, რომელიც ხელთ გვაქვს- (Arduino Uno ჩვენ შემთხვევაში ფიგ. შ.3).

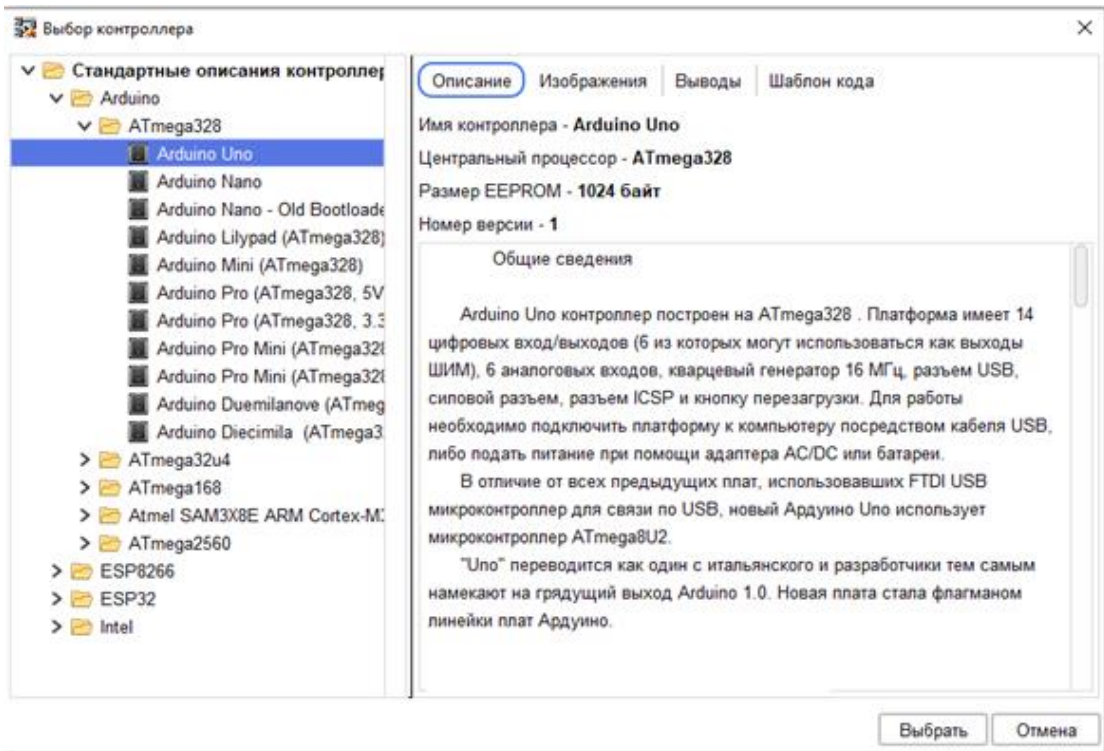


ფიგ. შ.1 ახალი პროექტის გახსნის ფანჯარა



ფიგ. შ.2 დაპროგრამების ენის არჩევა

ამის შემდეგ ვაჭერთ არჩევის ღილაკს-Выбрать და ეკრანზე გამოვა Arduino Uno კონტროლერის გამოსახულება და მისი ტექნიკური მახასიათებლები (ფიგ. შ.4).



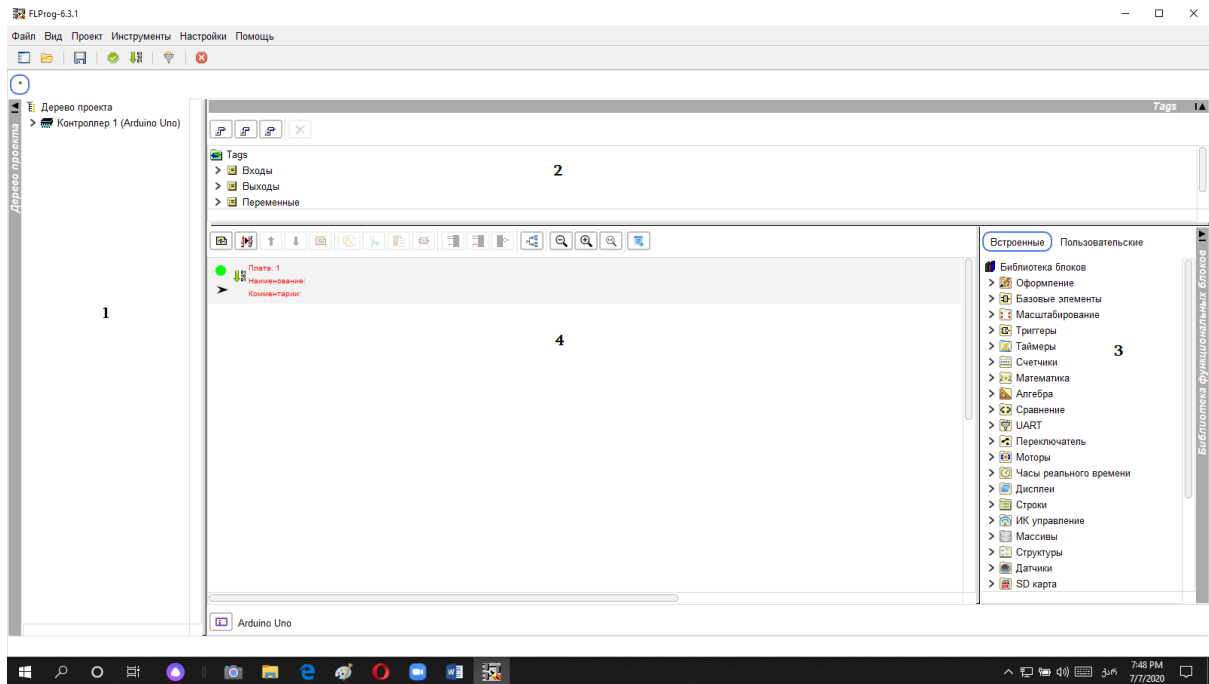
ფიგ. შ.3 Arduino Uno კონტროლერის არჩევის ფანჯარა



ფიგ. შ.4 Arduino Uno-ს გამოსახულება და მისი ტექნიკური მახასიათებლები

აქ კი კოდევ ერთჯერ ვაჭერთ თითს მზადყოფნის ლილავს-Готово და ეკრანზე გამოვა FLProg-ის ფანჯარა, რომელიც წარმოადგენს არდუინოს კონტროლერებისა და აგრეთვე სხვა

ტიპის კონტროლერების (იხ. ფიგ. შ.3) დაპროგრამების გრაფიკულ-ვიზუალურ გარემოს (ფიგ. შ.5).



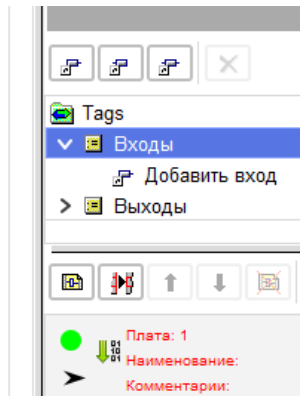
ფიგ. შ.5 FLProg დამუშავების გრაფიკულ-ვიზუალური გარემოს ფანჯარა

აღნიშნული გარემო შედგება ოთხი ზონისაგან (არესაგან): 1) პროექტის ხის ზონისაგან; 2) ტეგების ზონისაგან; 3) ბლოკების ბიბლიოთეკის ზონისაგან და 4) სამუშაო ზონისაგან. პირველ ზონას ჯერ არ განვიხილავთ, მეორე ზონის დანიშნულებაა: ა) ციფრული სიგნალებისათვის არდუინოს შესასვლელებისა და/ან გამოსასვლელების შექმნა; ბ) ანალოგიური სიგნალებისათვის არდუინოს შესასვლელების შექმნა; გ) არდუინოს მუშაობისათვის საჭირო ცვლადების შექმნა.

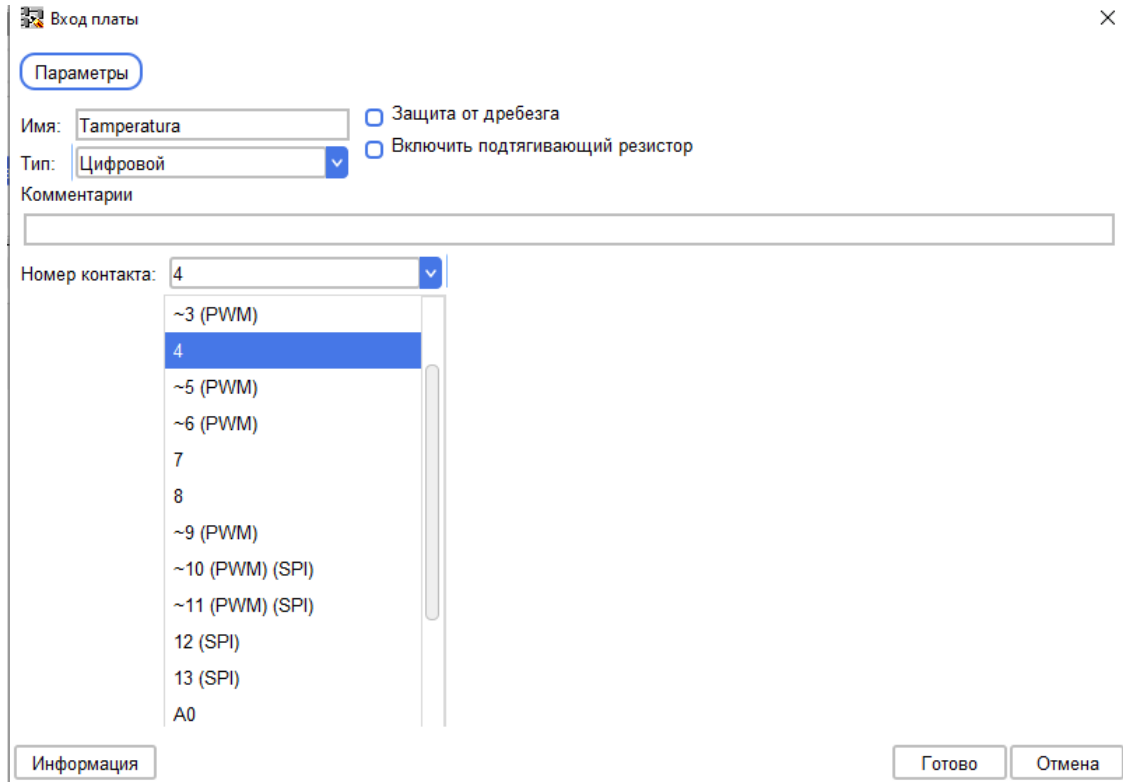
ზოგადად, არდუინოსთან მუშაობისათვის განიხილება ორი სახის სიგნალები: 1) ციფრული სიგნალები, როდესაც იგი იღებს მხოლოდ ორ მნიშვნელობას: ა) დაბალი დონის მნიშვნელობას, ანუ იგივე 0 ვოლტის მნიშვნელობას, ანუ იგივე ლოგიკური ნოლის მნიშვნელობას, ანუ იგივე ლოგ. „0“-ის მნიშვნელობას, ანუ იგივე False-ს; ბ) მაღალი დონის მნიშვნელობას, ანუ იგივე 5 ვოლტის მნიშვნელობას, ანუ იგივე ლოგიკური ერთიანის მნიშვნელობას, ანუ იგივე ლოგ. „1“-ის მნიშვნელობას, ანუ იგივე True-ს; 2) ანალოგიური სიგნალები, სადაც სიგნალი იღებს ნებისმიერ მნიშვნელობას 0...5 ვოლტის დიაპაზონში.

ციფრული სიგნალების შესასვლელების შექმნისათვის შევდივართ ტეგების ზონაში და თავის მარცხენა დილაკის შესასვლელების პიქტოგრამაზე-Входы ორჯერ დაჭერით, ეკრანზე გამოვა შესასვლელის დამატების პიქტოგრამა-Добавить вход (ფიგ. შ.6). შემდეგ თუ ამ

პიქტოგრამაზე დააწკაპებთ ორჯერ ეკრანზე გამოვა დაფის შესასვლელის-Вход платы პარამეტრების ფანჯარა-Параметры, სადაც სახელის ველში-Имя: ჩავწერთ ჩვენთვის სასურველ სახელს, მაგ. Temperatura, ჩამოვშლით ტიპის სიას-Тип: ისარზე დაწკაპებით და ორი შესაძლებელი ტიპიდან ციფრული ანალოგური-Цифровой Аналоговый, ციფრულის არჩევით, ეკრანზე გამოჩნდება კონტაქტის ნომრის ველი- Номер Контакта, ჩამოვშლით ამ ველს ისარზე დაწკაპებით და ავირჩევთ ჩვენთვის საჭირო არდუინოს კონტაქტის ნომერს (პინს), მაგ. ჩვენ შემთხვევაში მე-4-ს (ფიგ. შ.7).

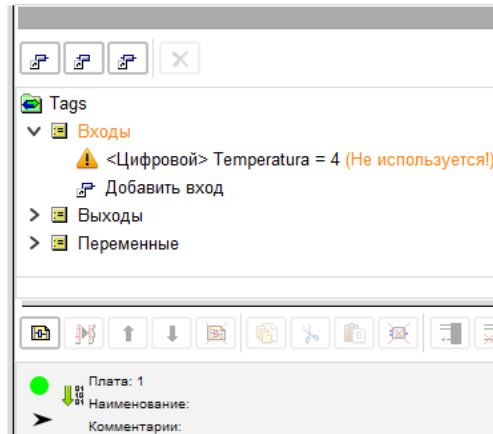


ფიგ. შ.6 შესასვლელის დამატების ფანჯარა



ფიგ. შ.7 ციფრული შესასვლელი კონტაქტის არჩევა

აქვე უნდა შევნიშნოთ, რომ თუ ავირჩიეთ რომელიმე კონტაქტი, მაგ. მე-4 ჩვენს შემთხვევაში, მაშინ მეორედ ამ კონტაქტს ამ პროექტში ვეღარ ავირჩევთ. გარდა ამისა, ციფრული კონტაქტების მაქსიმალური რაოდენობა Arduino Uno-სთვის შეზღუდულია და შეადგენს 14-ს (ნომრებით 0...13) და ეს კონტაქტები შეიძლება გამოყენებულ იქნას ან როგორც ციფრული სიგნალების შეყვანისათვის არდუინოში, ან როგორც ციფრული სიგნალების გამოყვანისათვის არდუინოდან. შემდეგ ვაჭერთ ამ ფანჯარაში მზადყოფნის ლილაკს-готово, ეს ფანჯარა გაქრება და ჩანაწერი აისახება ტეგების ზონაში (ფიგ. შ.8)

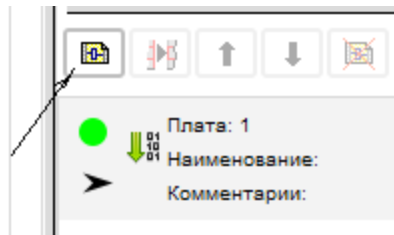


ფიგ. შ.8 არჩეული ციფრული კონტაქტის ასახვა ტეგების ზონაში.

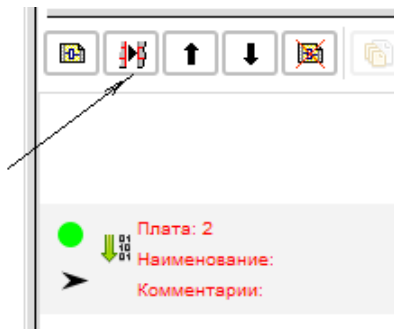
აქ ამ ახლად არჩეული კონტაქტის გასწვრივ ფრჩხილებში წერია არ არის გამოყენებული-(Не используется!), მაგრამ როდესაც მას ჩავიტანთ სამუშაო ზონაში მაშინ ეს ჩანაწერი გაქრება. თუ კიდევ გვინდა ახალი შესასვლელის შექმნა, მაშინ უნდა გავიმეოროთ ზემოთაღნიშნული პროცედურები ანალოგიურად.

შენიშვნა: ზემოთ ნახსენები იყო ტერმინი „დაფა“, FLProg პროგრამაში შესაქმნელი პროექტის აგება ხდება დაფებზე, თუ პროექტი მარტივია, მაშინ იგი დაეტევა პირვდელივე დაფაზე-Плата: 1, მაგრამ თუ პროექტი მოცულობითია და ვერ ეტევა ერთ დაფაზე, მაშინ მისი განლაგება შეიძლება ორ ან რამდენიმე დაფაზე. ჩვეულებრივად, ახლად გახსნილ FLProg-ში მუშაობის დროს მუშაობა წარმოებს ეგრეთ წოდებულ პირველ დაფაში-Плата: 1, რომელსაც შეგვიძლია დავარქვათ სახელწოდებაც-Наименование და გავუკეთოთ კომენტარებიც-Комментарии შესაბამის ველებში დაწკაპუნებითა და ჩაწერით (ფიგ. შ.9). მაგრამ ხშირ შემთხვევაში, ნახაზების გადატვირთვების თავიდან აცილების მიზნით, მოსახერხებელია ნაწილი ნახაზებისა, განსაკუთრებით ისეთი ნახაზების, რომლებიც ლოგიკურად გამოყოფილია სხვა ნახაზებისაგან, მიზანშეწონილია შესრულდეს სხვა დაფაზე. ამიტომ საჭირო ხდება, რომ ჩვენ უნდა შევქმნათ სხვა, დამატებითი დაფა და ეს შეიძლება მოხდეს

ფიგურა შ.9-ზე წვრილი ისრით ნაჩვენებ დაფის დამატების -Добавить плату ღილაკზე დაწკაპუნებით (ფიგ. შ.10).

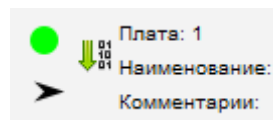


ფიგ. შ.9 პროექტში დაფის დამატება



ფიგ. შ.10 დაფის წაშლა

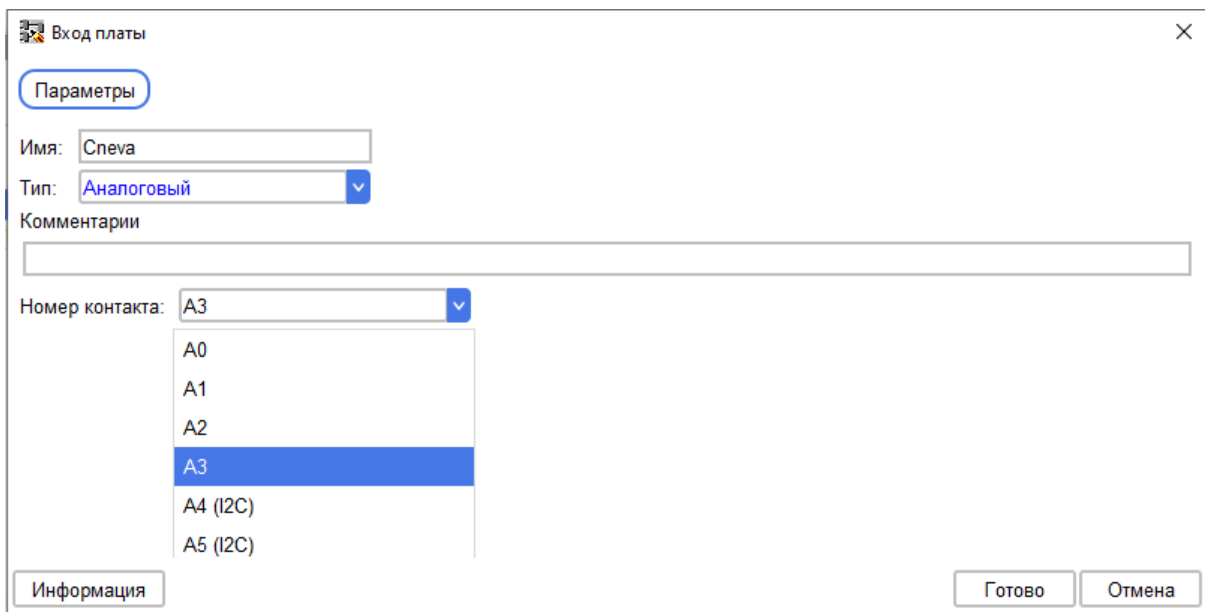
იმ შემთხვევაში თუ გვინდა დაფის დამატება მონიშნული დაფის წინ-Вставить плату перед выделенной, მაშინ უნდა დაწკაპუნდეს ფიგ. შ.10 წვრილი ისრით აღნიშნულ ნიშანზე, იმ შემთხვევაში, თუ გვინდა შევუცვალოთ ადგილები დაფებს ზევით ან ქვევით, მაშინ უნდა გამოვიყენოთ ზემოთ და ქვემოთ მიმართული ისრებიანი ღილაკები, იმ შემთხვევაში კი თუ რომელიმე დაფის წაშლაა საჭირო მაშინ უნდა მოინიშნოს ეს დაფა და როდესაც დაფის ნომერი და წარწერები: Наименование: და Комментарии: გაწითლდება, მაშინ უნდა დაწკაპუნდეს წითელი ხაზებით გადახაზულ ღილაკზე (ფიგ. შ.10). თუ ჩვენ გვინდა, რომ დაფა კი არ წავშალოდ, არამედ ჩაიკეცოს, რომ ხელი არ შეგვიშალოს სხვა დაფებთან მუშაობის დროს, ამიტომ უნდა დავაწკაპუნოთ დაფის ზემოთნახსენები წარწერების წინ მოთავსებულ მარჯვნივ მიმართულ მსხვილ შავ ისარზე (ფიგ. შ.11).



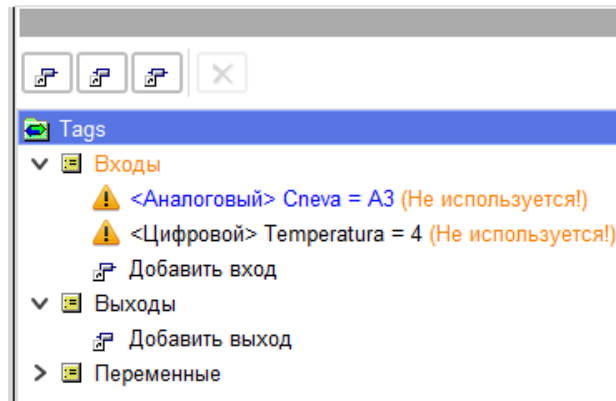
ფიგ. შ.11 დაფის ჩაკეცვის ისარი (ღილაკი)

ეს ისარი ამ შემთხვევაში გადაიქცევა ქვემოთმიმართულ ისრად და ამ დაფის აღდგენის საჭიროებისათვის უნდა დაწკაპუნდეს აღნიშნულ ისარზე ხელმეორედ. გარდა ამისა, მხედველობაში უნდა იქნას მიღებული, რომ სხვადასხვა დაფებზე განლაგებული ბლოკების ერთმანეთთან პირდაპირი შეერთება არ შეიძლება, სხვადასხვა დაფებზე განთავსებული ბლოკების შეერთებისთვის საჭიროა შეიქმნეს დამატებითი ცვლადი. ეს ცვლადი უნდა ჩამოტანილ იქნას ორთავე (ან რამდენიმე) დაფაზე და შეერთდეს შესაბამისი ბლოკების გამომავალ და შემავალ გამომყვანებს. მასასადამე ცვლადების შექმნითა და მისი გამოყენებით შესაძლებელია დაფებსშორისი ნებისმიერი რაოდენობის შეერთებების რეალიზება.

ახლა შევქმნათ კიდევ ერთ შესასვლელი, სახელად დავარქმევთ Cneva და ტიპად ავირჩევდით ანალოგური-Аналоговый, მაშინ პარამეტრიზაციის ფანჯარა მიიღებდა სახეს (ფიგ. შ.12) და თუ აქ კონტაქტის ნომრად ავირჩიეთ მაგალითად A3 კონტაქტს და დავაჭერთ ღილაკზე Готово ეს კონტაქტიც აისახება ტეგების ზონაში (ფიგ. შ.13). აქვე უნდა აღინიშნოს, რომ Arduino Uno-სთვის ანალოგური შესასვლელი კონტაქტების რაოდენობა, ისევე როგორც ციფრული კონტაქტების რაოდენობა, შეზღუდულია და შეადგენს ექვსს (ნომრებით A0...A5).

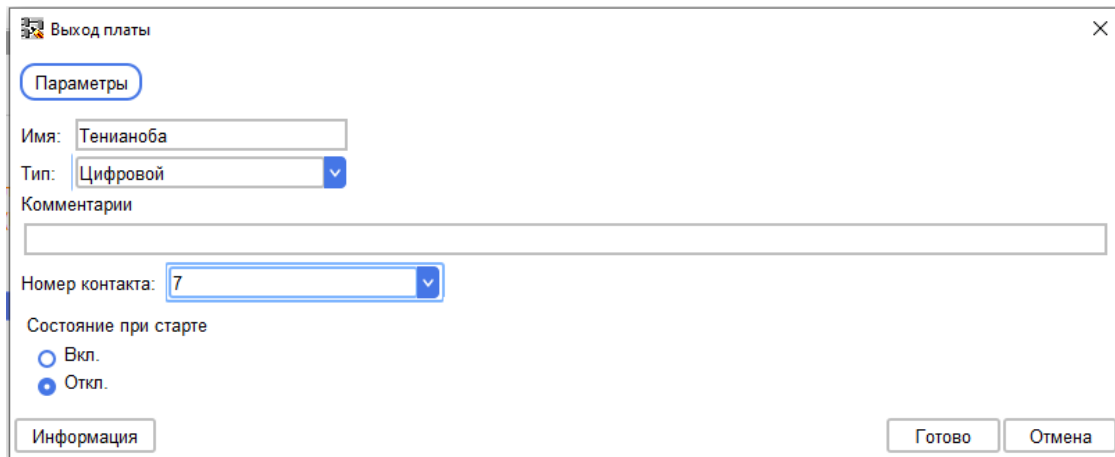


ფიგ. შ.12 ანალოგური შესასვლელი კონტაქტის არჩევა



ფიგ. შ.13 არჩეული ანალოგური შესასვლელი კონტაქტის ასახვა ტეგების ზონაში

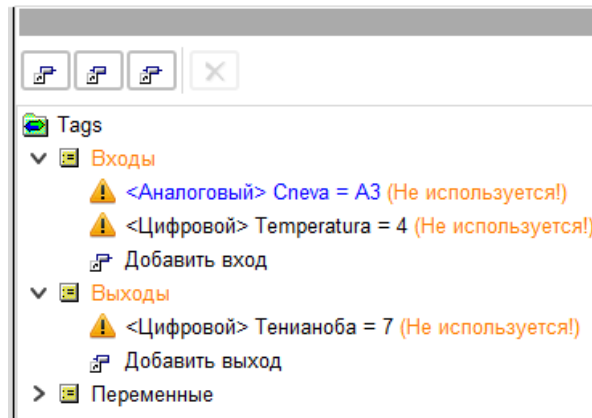
ახლა განვიხილოთ თუ როგორ უნდა შეიქმნას არდუინოს გამოსასვლელი. გადავიდეთ ისევ ტეგების ზონაში და დავაჭიროთ ორჯერ პიქტოგრამას გამოსასვლელი-Выходы, როგორც შესასვლელის შემთხვევაში, აქაც გამოვა ახალი პიქტოგრამა სახელწოდებით დაემატოს გამოსასვლელი-Добавить выход, დავაჭიროთ ამ პიქტოგრამაზე ორჯერ თითო და ეკრანზე გამოვა დაფის გამოსასვლელის-Выход платы პარამეტრების ფანჯარა-Параметры (ფიგ. შ.14), სადაც სახელის ველში-Имя: ვწერთ რაიმე, ჩვენთვის საჭირო პარამეტრის მნიშვნელობას, მაგ. Тенианоба, ტიპის ველში-Тип: ჩამოშლადი სიიდან ვირჩევთ ციფრულს-Цифровой, კონტაქტის ნომრის ველში ჩამოშლადი სიიდან ვირჩევთ ნებისმიერს, მაგ. მე-7 კონტაქტს.



ფიგ. შ.14 ციფრული გამოსასვლელი კონტაქტის არჩევა

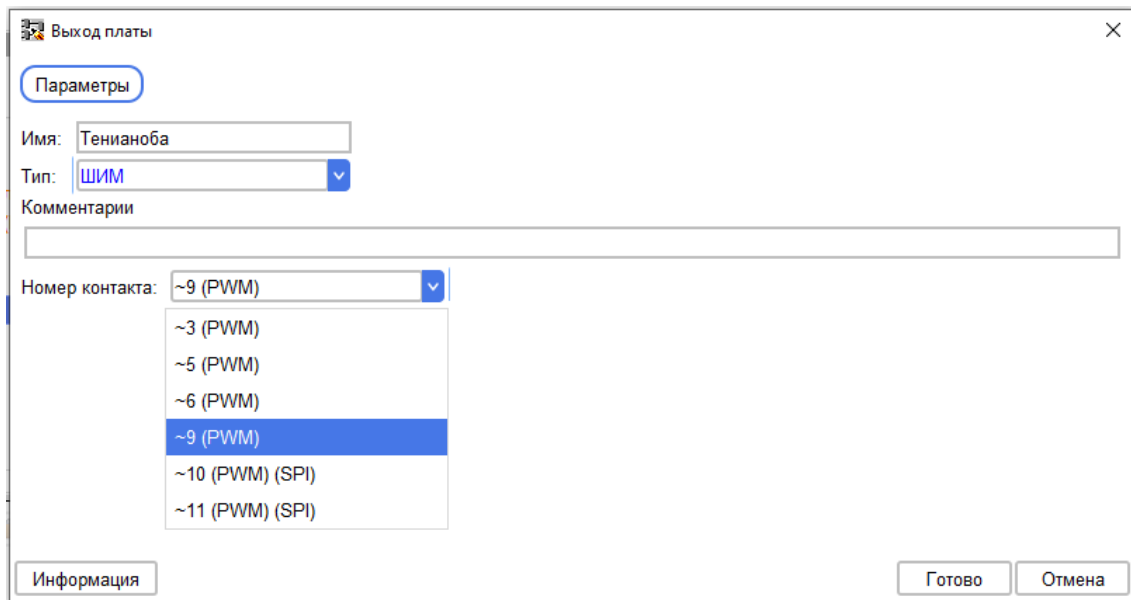
მივაქციოთ ყურადღება, რომ ჩამოშლად სიაში მე-4 ნომერი უკვე აღარ არსებობს, ეს იმიტომ, რომ ჩვენ იგი უკვე გამოვიყენეთ როგორც შესასვლელი კონტაქტი. ამრიგად არდუინოს ამ თოთხმეტი ციფრული კონტაქტიდან ნებისმიერი მათგანი შეიძლება გამოყენებულ იქნას ან როგორც ციფრული შესასვლელი, ან როგორც ციფრული

გამოსასვლელი, მაგრამ მხოლოდ ერთჯერ, შემდეგ დავაჭიროთ მზადყოფნის ლილავს-Готово და ტეგების ზონა მიიღებს სახეს (ფიგ. შ.15).

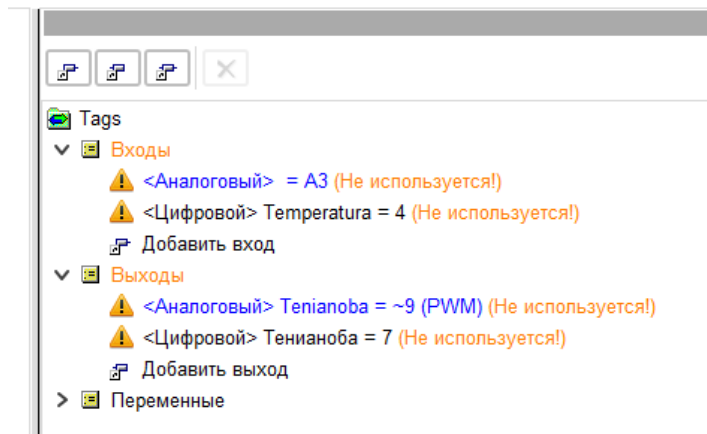


ფიგ. შ.15 არჩეული დისკრეტული გამოსასვლელის ასახვა ტეგების ზონაში

ამის შემდეგ, თუ ჩვენ შევქმნიდით კიდეც ერთ გამოსასვლელს და მის ტიპად ავირჩევდით ШИМ (Широтно-импульсная модуляция)-(განივ იმპულსური-მოდულაცია), ხოლო კონტაქტის ნომრად-ცხრას, მაშინ უკვე მივიღებდით ახალ ფანჯარას (ფიგ. შ.16). ტეგების ზონა კი მიიღებდა სახეს (ფიგ. შ.17). ШИМ სიგნალების რაობა განხილული იქნება ქვემოთ შესაბამის პროექტში.

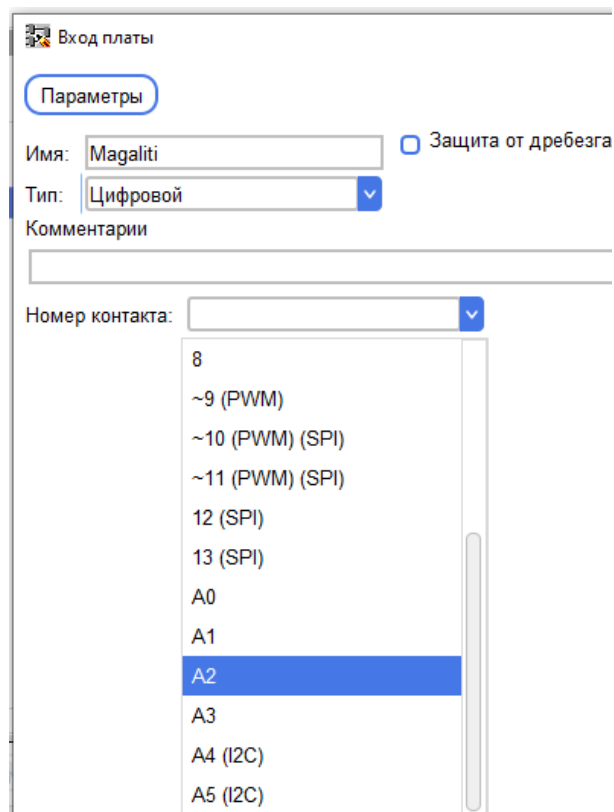


ფიგ. შ.16 ШИМ გამოსასვლელი კონტაქტის არჩევა



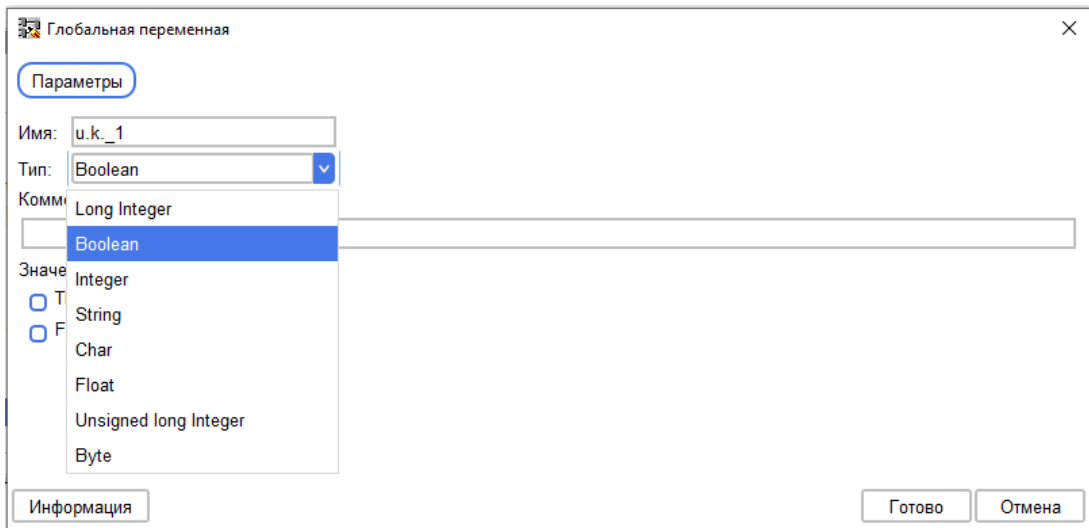
ფიგ. შ.17 არჩეული ШИМ გამოსასვლელის ასახვა ტეგების ზონაში

გარდა ამისა, FLProg-ს აქვს შესაძლებლობა, ციფრული კონტაქტების გაზრდის მიზნით, არდუინოს ანალოგური კონტაქტები გამოყენებულ იქნას როგორც ციფრული შესავლელები, ანდა როგორც ციფრული გამოსასვლელები. ასეთი საჭიროების დადგომის შემთხვევაში, ჩვენ უბრალოდ პარამეტრიზაციის ფანჯარაში ვირჩევთ შესასვლელს (გამოსასვლელს) ციფრულს-Цифровой, მაგრამ კონტაქტის ნომრად ვირჩევთ ერთერთს A0...A5-იდან. ჩვენ შემთხვევაში ვირჩევთ A2 კონტაქტს (ფიგ. შ.18).

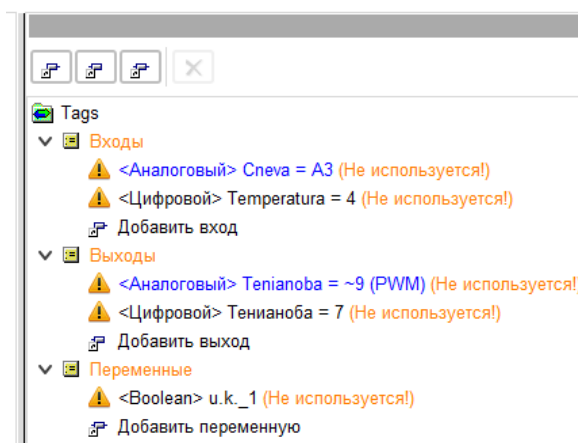


ფიგ. შ.18 ანალოგური კონტაქტების გამოყენება როგორც ციფრულების

ახლა უკვე შეგვიძლია გადავიდეთ ცვლადის შექმნაზე, პროექტის შექმნის მომენტში შეიძლება დაგვჭირდეს მაგალითად უკუკავშირის შექმნა, სადაც უკვე საჭირო იქნება ცვლადების გამოყენება. ცვლადის შექმნის პროცედურა ასეთია: ტეგების ზონაში ვაწკაპებთ ცვლადების პიქტოგრამას-Переменные და შემდეგ პიქტოგრამას, დაემატოს ცვლადი-Добавить переменную. ეკრანზე გამოვა გლობალური ცვლადის-Глобальная переменная პარამტრების-Параметры ფანჯარა. სახელის ველში უნდა ჩავწეროთ ის სახელი რასაც ჩვენ ჩავთვლით საჭიროთ მაგ. Ukukavshiri_1 (შემოკლებით u.k._1) (ფიგ. შ.19). ტიპის ველში ჩამოშლადი სიიდან უნდა ავირჩიოთ მონაცემთა ის ტიპი, რა ტიპისაც შეიძლება იყოს ჩვენს მიერ შექმნილი ცვლადი. უკუკავშირის შემთხვევაში უფრო ხშირად გამოიყენება ბულის ტიპის ცვლადი, რომელმაც ზოგადად, შეიძლება მიიღოს ან ერთიანის ან ნოლის მნიშვნელობა. ავირჩიოთ ჩვენც Boolean ტიპის მონაცემი, მზადყოფნის დილაკზე-Готово დაჭერის შემდეგ ჩვენს მიერ შექმნილი ცვლადი აისახება ტეგების ზონაში (ფიგ. შ. 20).



ფიგ. შ. 19 ცვლადის შექმნის ფანჯარა

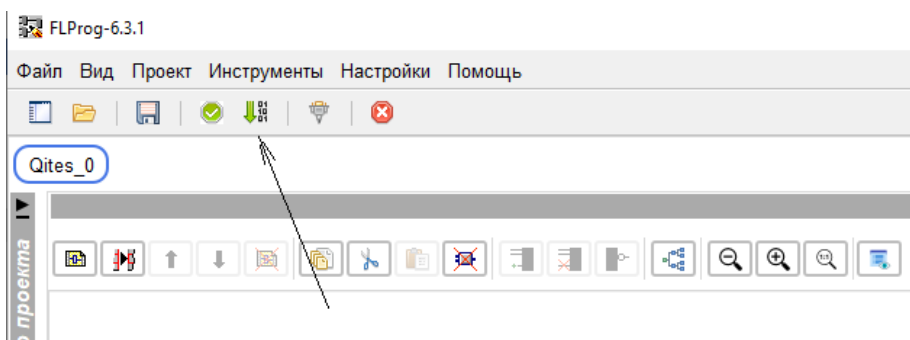


ფიგ. შ. 20 შექმნილი ცვლადის ასახვა ტეგების ზონაში

ასევე, თუ საჭირო იქნება პროექტში ჩვენ შეგვიძლია შევექმნათ ნებისმიერი რაოდენობისა და ტიპის ცვლადები. ნებისმიერი ჩვენს მიერ შექმნილი კონტაქტი თუ ცვლადი (ბლოკი) ტეგების ზონიდან, ან მზა ბლოკი ბლოკების ბიბლიოთეკიდან-Библиотека блоков (ზონა 3) შეიძლება გადმოტანილ იქნას სამუშაო ზონაში (ზონა 4) შემდეგნაირად: მიგვაქვს თავის მაჩვენებელი ისარი ნებისმიერ ბლოკთან და მარცხენა ღილაკით მოვნიშნავთ მას, შემდეგ დავაჭერთ იმავე მარცხენა ღილაკით მონიშნულ ბლოკს და თითის აუშვებლად გადმოგვაქვს მე-4 ზონაში, აქ კი აისახება კიდეც ამ ბლოკის შესაბამისი გრაფიკული ნიშანი.

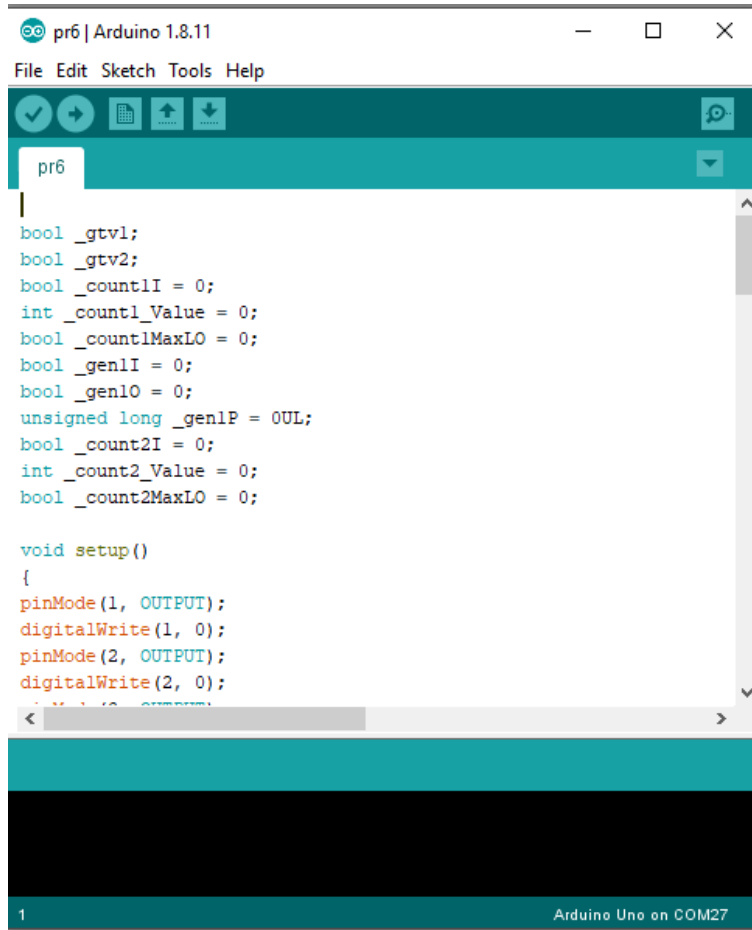
იმისათვის, თუ როგორ უნდა შეიქმნას პროგრამა ცალკეული პროექტისათვის განხილული იქნება თვითოეული პროექტისათვის ცალცალკე, აქ კი შევხვით თუ როგორ მოვახერხოთ ამ ჩვენი დამუშავებული FLProg პროგრამის ჩატვირთვა არდუინოში.

პირველ რიგში არდუინოს კომპლექტში მოყოლილი კაბელით უნდა შევაერთოთ არდუინო პერსონალური კომპიუტერის ერთერთ USB პორტს. ამის შემდეგ საჭიროა თავის მარცხენა ღილაკით დავაჭიროთ პროექტის კომპილირების ღილაკს- Компилировать проект (ფიგ. შ.21). (ეს ღილაკი ნაჩვენებია ისრით).



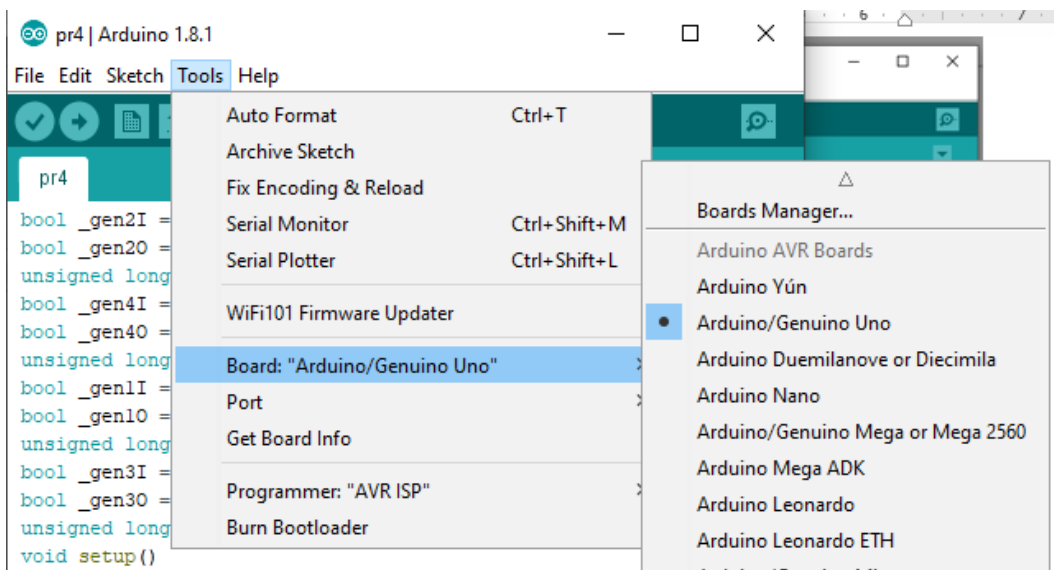
ფიგ. შ.21 პროექტის კომპილაციის ღილაკი

კომპილიაციის პროცესში ხორციელდება ჩვენს მიერ FLProg-ში დამუშავებული პროგრამის გადათარგმნა არდუინოს კომპანის მიერ შემოთავაზებულ დამუშავების გარემოში-Arduino IDE (ფიგ. შ.22). სადაც რიცხვითი მნიშვნელობა 1.8.11 აღნიშნავს ამ პროგრამის ვერსიას და შეიძლება იყოს სხვა მნიშვნელობის.



ფიგ. შ. 22 FLProg პროგრამის კომპილაცია Arduino-ს პროგრამაში

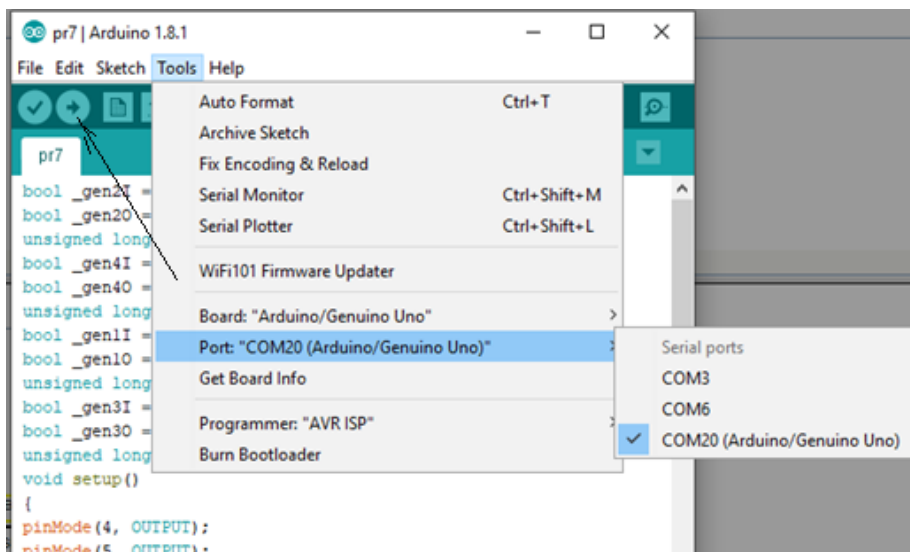
ამ ახალ ფანჯარაში უნდა შევამოწმოთ ორი ჩანაწერი. დავაჭიროთ ხელი ღილაკს Tools და გამომავალ ფანჯარაში ჩამოშლადი სიიდან: 1) შევამოწმოთ მონიშნულია თუ არა Board: „Arduino/Genuio Uno” (ფიგ. შ.23).



ფიგ. შ.23 არდუინოს მოდელის შემოწმება

თუ მონიშნულია არდუინოს სხვა რომელიმე მოდელი უნდა შევცვალოთ და მოვნიშნოთ „Arduino/Genuio Uno“, რადგან ჩვენს მიერ დასაწყისშივე ეს მოდელი იქნა არჩეული;

2) შევამოწმოთ ჩანაწერი Port და დავრწმუნდეთ მინიჭებული აქვს თუ არა ჩვენს მიერ გამოყენებულ არდუინოს ვირტუალური ნომერი პერსონალური კომპიუტერის სისტემის მიერ და არის თუ არა ეს ვირტუალური პორტი მონიშნული ალმით, (Port “Com20 (Arduino/Genuio Uno)” ჩვენს შემთხვევაში (ფიგ. შ.24), აქ Com პორტის ნომრად გამოიყენება მე-20 ნომერი, მაგრამ შეიძლება სხვა რომელიმე რიცხი იყოს მითითებული. თუ არ არის მონიშნული ალმით-მოვნიშნოთ ეს პორტი. ამის შემდეგ კი უნდა ჩავტვირთოთ კონტროლერში ეს ჩვენი კომპილირებული პროგრამა. ჩატირთვა მოხდება ისარზე თითის დაჭერით, რაც ფიგ. შ.24-ზე ისრითაა მითითებული.



ფიგ. შ.24 ვირტუალური ნომრის მინიჭება არდუინოს პორტს

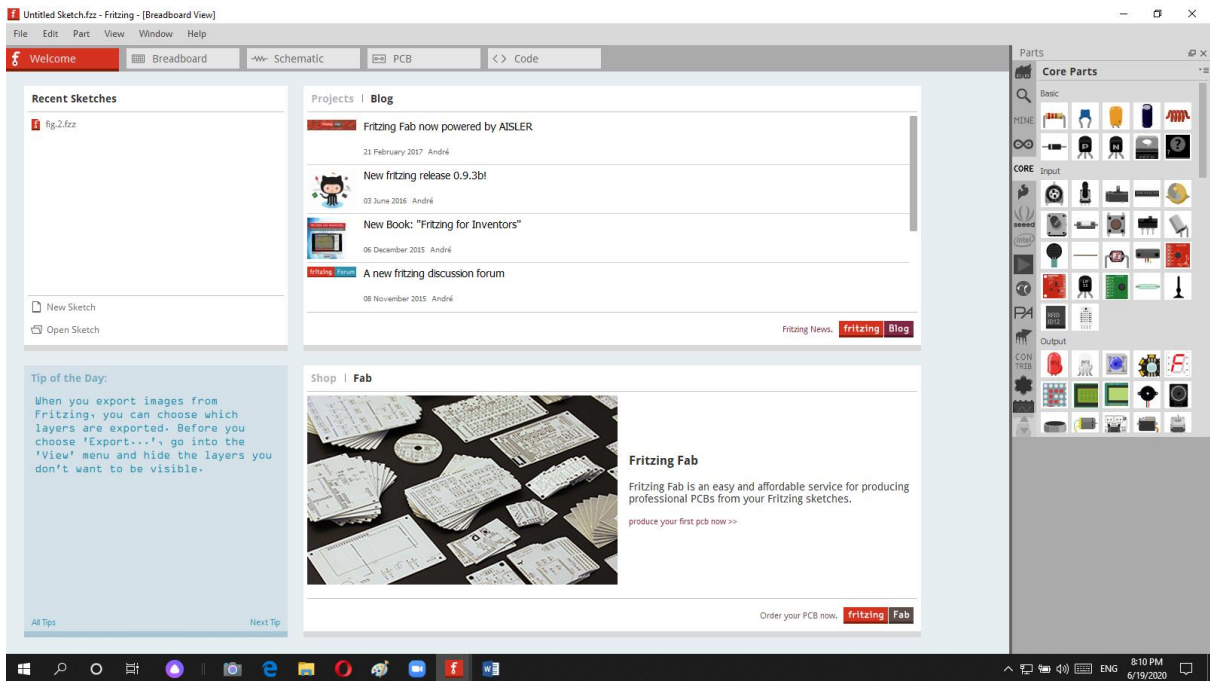
როგორც FLProg დამუშავების გრაფიკულ-ვიზუალური გარემოს ფანჯარაში (ფიგ. შ.5) ჩანს, იქ არის აგრეთვე გამოყენებული ჩაკეცვის ღილაკები (ფიგ. შ.25). ამ ღილაკებზე დაჭერით შესაბამისი ზონები ჩაიკეცება და გაიზრდება სამუშაო ზონის ფართი. ამით შეიქმნება შესაძლებლობა უფრო დიდი მოცულობის პროგრამები დავატიოთ სამუშაო ზონაში. ჩაკეცილი ზონები მარტივად შეგვიძლია აღდგეს იმავე ღილაკებზე (თუმცა ისრის მიმართულებები შეცვლილი იქნება საწინააღმდეგოზე) დაწკაპუნებით.



ფიგ. შ.25 ზონების ჩაკეცვისა და აღდგენის ღილაკები


ახლა განვიხილავთ თუ როგორ უნდა შევავროთ არდუინოს კონტაქტები პროექტში გამოყენებულ ელექტრონულ ელემენტებს. პროექტის სამონტაჟო სქემის დასახაზათ ვიყენებთ პროგრამა Fritzing-ს, რომელიც სპეციალურად არის შექმნილი არდუინოს კომპლექტებთან სამუშაოდ და უფასოდაა გავრცელებული.


გადმოვიწეროთ ინერნეტიდან პროგრამა Fritzing და გაუშვათ. ეკრანზე გამოვა ამ პროგრამის სამუშაო არე (ფიგ. შ.26).

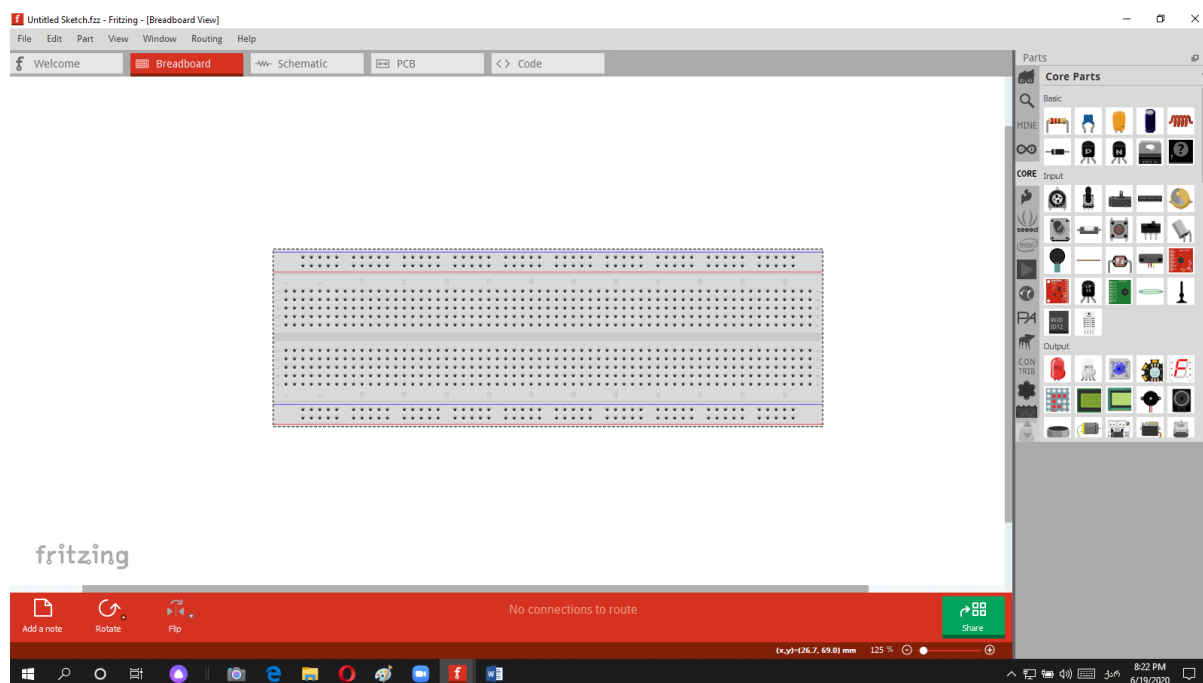


ფიგ. შ.26 Fritzing პროგრამის სამუშაო არე

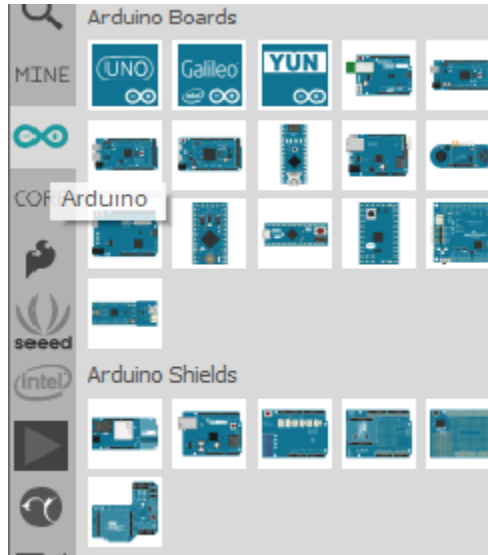
როგორც ფიგურიდან ჩანს Fritzing პროგრამა შედგება ორი ძირითადი ნაწილისაგან, მარჯვენასაგან სადაც მოთავსებულია ელემენტების ბიბლიოთეკა და მარცხენასაგან-სადაც მოთავსებულია ელექტრული/სამონტაჟო სქემის განთავსების ზონა.

ამ პროგრამის მენიუში დავაწკაპუნოთ ლილავს  Breadboard და სამუშაო არეში აისახება სამონტაჟო დაფის გამოსახულება (ფიგ. შ.27). თუ მოვნიშნავთ ამ დაფას და დავატრიალებთ თავის გორგოლაქს წინ და უკან, მაშინ დავინახავთ, რომ სამონტაჟო დაფის ზომები იცვლება ზრდისა და კლების მიმართულებით შესაბამისად, ასევე შეგვიძლია დაფის გადაადგილება სამუშაო ზონაში მისი მონიშვნითა და თითის აულებლად აქეთ იქით გადაადგილებით.

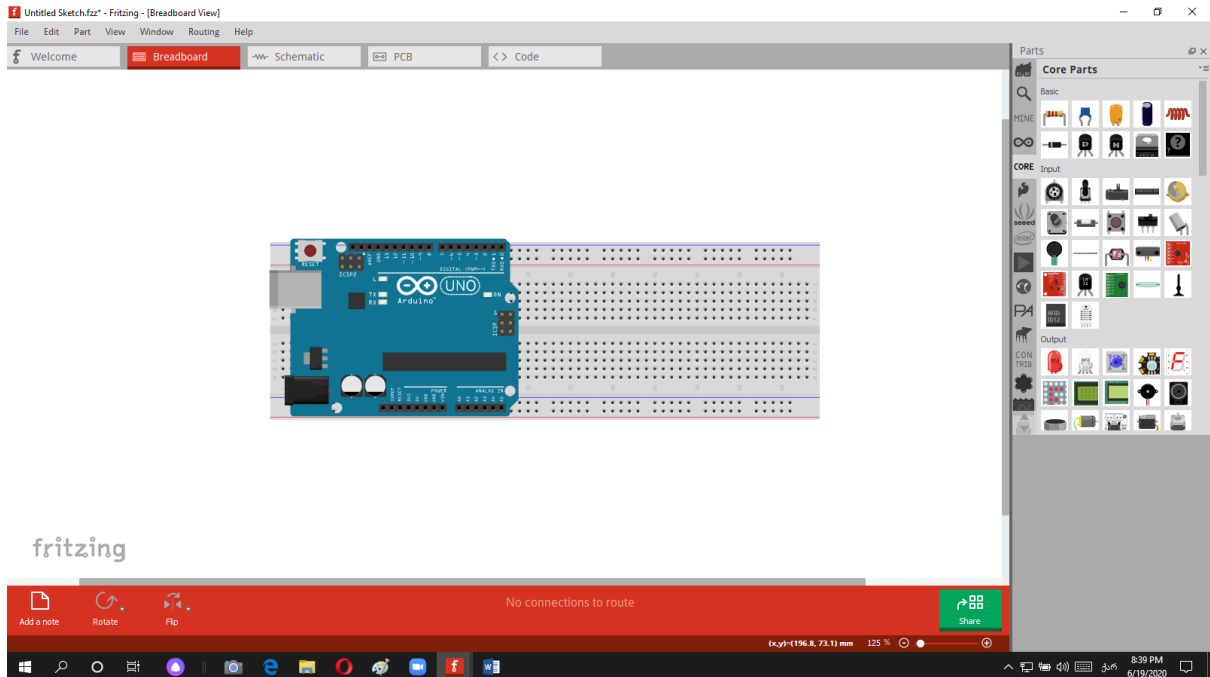
ახლა, გადავიდეთ პროგრამის ბიბლიოთეკის არეში, დავაწკაპოთ არდუინოს დაფების საქაღალდეს  (ფიგ. შ.28) ჩამოიშლება არდუინოს დაფების ჩამონათვალი, საიდანაც მოვნიშნოთ UNO თავის მარცხენა ლილაკით, თითის აუშვებლად გადმოვიტანოთ პროგრამის სამუშაო ზონაში და განვალაგოთ ისე, როგორც ფიგ. შ.29-ზეა გამოსახული.



ფიგ. შ.27 სამონტაჟო დაფის განთავსება პროგრამის სამუშაო არეში



ფიგ. შ.28 არდუინოს დაფები

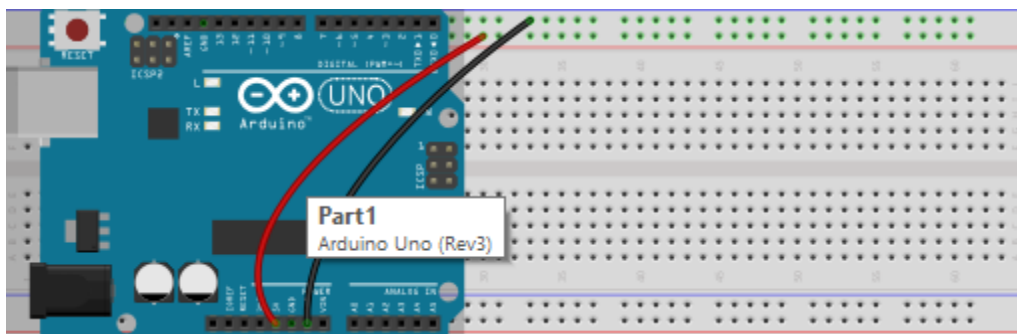


ფიგ. შ.29 არდუინოს დაფის გადმოტანა სამუშაო ზონაში

შევაერთოთ არდუინოს კვების კონტაქტები GND-ს და 5v-ს სამონტაჟო დაფის ჰორიზონტალურ ლურჯ და წითელ ხაზებთან შესაბამისად. აღსანიშნავია, რომ ამ სამონტაჟო დაფის წითელი ხაზები ერთმანეთთან ელექტრულადაა გაერთიანებული, ისევე როგორც ლურჯი ხაზებია ერთმანეთთან ელექტრულად გაერთიანებული, მაგრამ თვითონ წითელი და ლურჯი ჰორიზონტალური ხაზები ერთმანეთთან არ არიან გადაერთებული. ასევე ვერტიკალური ხაზების ყოველი ხუთ-ხუთი ბუდე ერთმანეთთანაა ელექტრულად

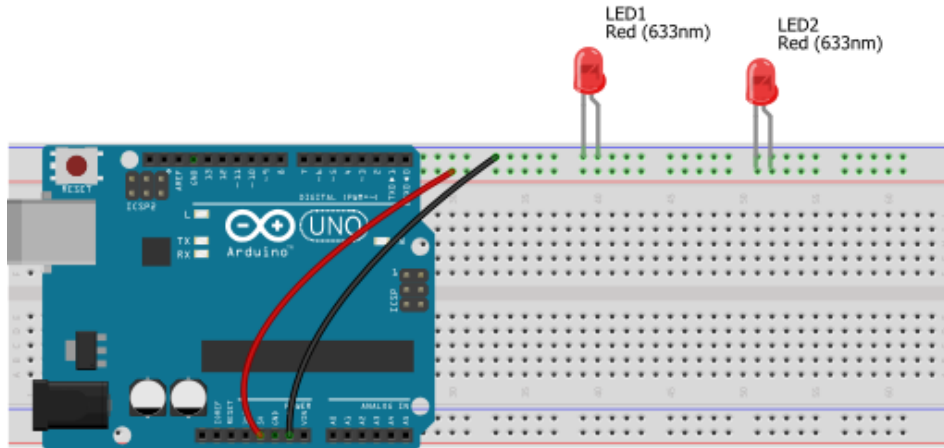
გაერთიანებული, მაგრამ ეს ბუდეები სხვა ვერტიკალურ ბუდეებთან არ არიან დაკავშირებულნი.

საერთოდ, ორი რომელიმე ბუდე, ან არდუინოს კონტაქტი რომ შევაერთოთ გამტარით ერთკმანეთთან საჭიროა თავის მაჩვენებელი მივიყვანოთ ერთერთ ბუდესთან ან კონტაქტთან, დავაჭიროთ თავის მარცხენა ღილაკს თითი და თითის აუშვებლად გავაგრძელოთ მეორე ბუდემდე და აუშვათ ღილაკს თითი. ამ დროს ამ ბუდეებს, ან ბუდესა და კონტაქტს შორის გაივლება სწორი ხაზი, ანუ გამტარი, რომლის ფერიც შეგვიძლია შევცვალოთ. ამისთვის მივიყვანოთ თავის მაჩვენებელი ხაზთან, დავაჭიროთ თავის მარჯვენა ღილაკს, ჩამოშლილი სიიდან ავირჩიოთ გამტარის ფერი-Vire Color და ჩამოშლადი სიიდან ავირჩიოთ ჩვენთვის სასურველ ფერი, მაგალითად წითელი Red. ასევე, ჩვენ შეგვიძლია გავამრუდოთ ხაზი თუ მივიყვანთ თავის მაჩვენებელს ხაზზე და და როდესაც გამოჩნდება სპეციალური, სამკუთხედის მსგავსი ნიშანი, მაშინ დავაწვეთ კლავიატურის Ctrl ღილაკს და გავქაჩოთ ხაზი იმ მხარეს, სადაც უფრო მისახერხებელი იქნება. ახლა დავუბრუნდეთ ჩვენს ამოცანას. არდუინოს მიწა (GND) შევაერთოთ სამონტაჟო დაფის ლურჯ ჰორიზონტალურ ხაზთან, ხოლო არდუინოს 5v შევაერთოთ სამონტაჟო დაფის წითელ ჰორიზონტალურ ხაზთან (ფიგ. შ.30).



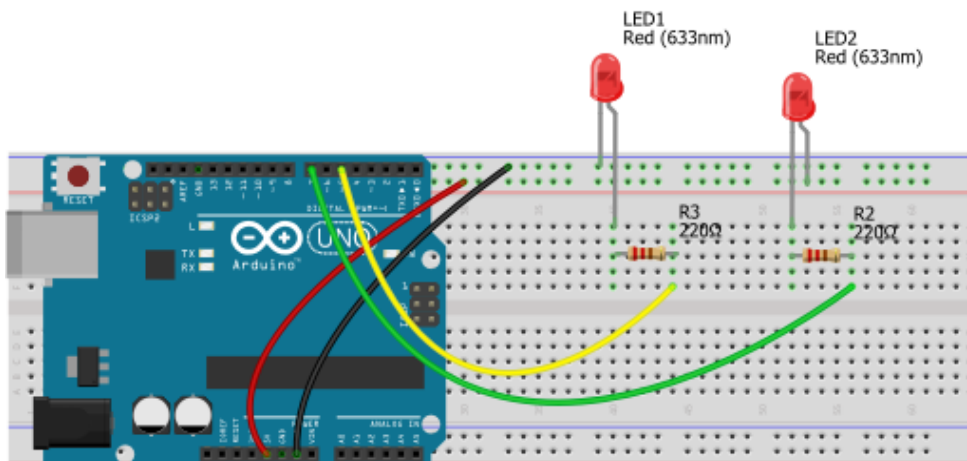
ფიგ. შ.30 მიწისა და 5 ვოლტის გადაყვანა სამონტაჟო დაფაზე

შევცვალოთ გამტარების ფერები 5v-ის გამტარი გავხადოთ წითელი, ხოლო მიწის გამტარი შავი. ამის შემდეგ თუ ჩვენ დავაჭიროებთ რომელიმე ელექტრული კომპონენტისათვის კვება მიწა და +5 ვოლტი ამისთვის არდუინოს კი არ მოვიკითხავთ, არამედ სამონტაჟო დაფის ჰორიზონტალურ ხაზებს. მართლაც გავხსნათ ბიბლიოთეკის CORE საქაღალდე და სამუშაო არეში გადმოვიტანოთ ორი წითელი LED1, LED2 შუქდიოდი სათითაოდ და დავსვათ ისინი ისე, როგორც ეს ფიგ. შ.31-ზეა ნაჩვენები ერთი ლურჯ ხაზზე და მეორე წითელ ხაზზე.



ფიგ. 2.31 შუქდიოდების გადმოტანა და ერთერთი შესაძლო განლაგება დაფაზე

შევიდეთ მათ კონტექსტურ მენიუში მათზედ თავის მარჯვენა ღილაკის დაწკაპუნებით, ჩამოშლილი სიდან ავირჩიოთ Show part label, რის შემდეგაც პროგრამა ავტომატურად წააწერს ამ შუქდიოდებს ნომრებს LED1 და LED2. (ფიგ. შ.31). ახლა მივაქციოთ ყურადღება, რომ პირველი შუქდიოდის ანოდის და კათოდის ერთიდაიგივე ნულოვან პოტენციალზე ზიან, ხოლო მეორე შუქდიოდის + 5 ვოლტზე. ეს კი შეცდომაა, ამოიტომ მივიყვანოთ თავის მაჩვენებელი პირველი შუქდიოდის ანოდთან და გამოვაგრძელოთ და დავსვათ ვერტიკალური ხუთეულის ერთერთ ბუდეზე, ისევე როგორც მეორეს კათოდი ასევე მის სხვა ვერტიკალური ბუდეების ერთერთ ბუდეზე (ფიგ. შ.32). ახლა Fritzing პროგრამის ბიბლიოთეკის იმავე **CORE** საწაღალდედან გადმოვიტანოთ ორი წინააღმდეგობა და მათი ერთერთი გამომყვანები დავსვათ ისე, როგორც ეს ნახაზზეა ნაჩვენები. მეორე გამომყვანები კი მივუერთოთ არდუინოს მე-5 და მე-7 კონტაქტზე გამტარებით, შევცვალოთ მათი ფერები და ფორმები.

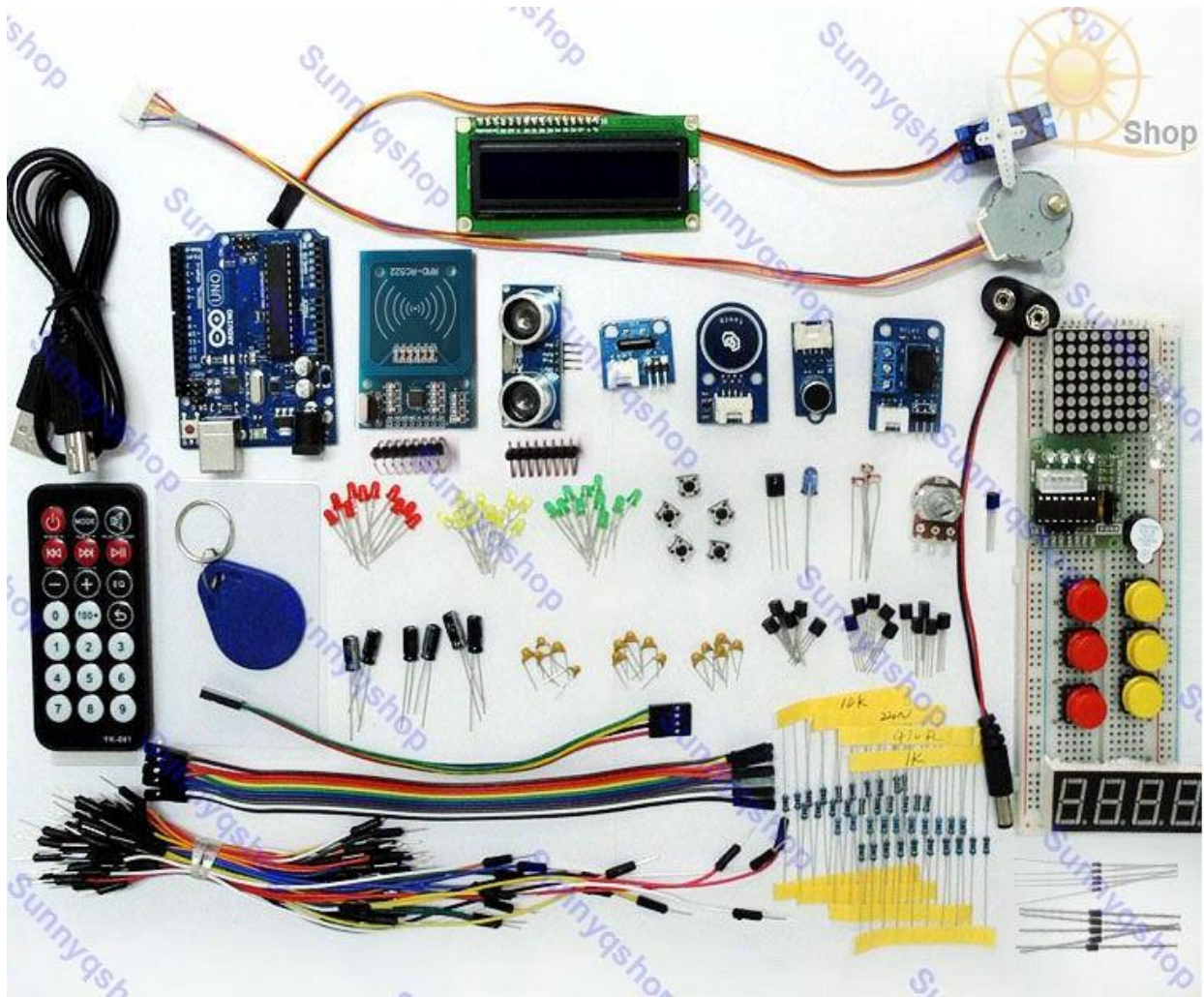


ფიგ. შ.32 პროექტის ელექტრულ-სამონტაჟო Fritzing სქემის მაგალითი

ამრიგად ჩვენ შევქმენით უმარტივესი სქემა ორი შუქდიოდისა და ორი წინააღმდეგობის გამოყენებით. ახლა ავაწყით ეს სქემა რეალური კომპონენტების გამოყენებით. შემდეგ, თუ ჩვენ შევძლებთ დავაჯინოთ +5 ვოლტი მაგალითად არდუინოს მე-5 კონტაქტზე მაშინ ეს +5 ვოლტი ამ კონტაქტიდან ყვითელი გამტრის გავლით მიეწოდება პირველ წინააღმდეგობას R3 და მის გავლით LED1 შუქდიოდის ანოდს, კათოდი კი ზის მიწაზე, მაშინ ეს დიოდი ჩართული იქნება პირდაპირი მიმართულებით ძაბვების მიმართ, მასში გაივლის დენი და იგი გაანათებს. ასევე თუ კი ჩვენ შევძლებთ არდუინოს მე-7 კონტაქტზე ვაფორმირთ ნოლოვანი პოტენციალი, მაშინ ეს პოტენციალი მწვანე გამტარის გავლით მიეწოდება მეორე წინააღმდეგობს R2-ს გამოყვანს, შემდეგ, ამ წინააღმდეგობის გავლით მეორე შუქდიოდის LED2 კათოდს, რომლის ანოდიც მიერთებულია +5 ვოლტზე, მაშინ ეს დიოდი იქნება ჩართული პირდაპირი მიმართულებით, მასში გაივლის დენი და იგი აინთება. თუ როგორ ვაფორმირთ ასეთი ძაბვები (სიგნალები) არდუინოს კონტაქტებზე ამას განვიხილავთ შემდეგ პროექტებში.

აქვე უნდა აღინიშნოს, რომ FLProg-ში შექმნილ პროგრამებში შავი ხაზები გამოიყენება მხოლოდ დისკრეტული სიგნალების ლოგიკური ნოლისა და ლოგიკური ერთის გადასაცემად, ხოლო ლურჯი ხაზები გამოიყენება ეგრეთ წოდებული ანალოგური სიდიდეების გადასაცემად, როგორცაა Integer-მთელი, Long Integer-გრძელი მთელი, Float-მცოცავ მძიმის რიცხვი და სხვ. FLProg-თან მუშაობის დროს ჩვენ მუდამ უნდა გვახსოვდეს, რომ მისი მუშაობის თანმიმდევრობა არის ბლოკებში ჩადებული ფუნქციების შესრულება ბლოკების მიხედვით მარცხნიდან მარჯვნივ და ზემოდან ქვემოთ ციკლურად, ე.ი. ჯერ ასრულებს ერთი მწკრივის ბლოკებს, მერე მეორე მწკრივის ბლოკებს, და ა.შ. ბოლო მწკრივის ბლოკებს და შემდეგ ისევ პირველი მწკრივის ბლოკებს და ა.შ.

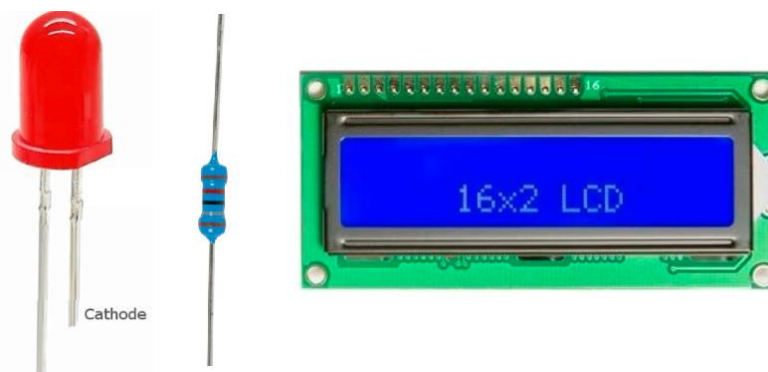
ქვემოთ მოყვანილ პროექტებში გამოყენებული იქნება ელექტრონული კომპონენტების სასტარტო კომპლექტი Arduino Starter Kit (ფიგ. შ.33).



ფიგ. შ.33 არდუინოს სასტარტო კომპლექტი

პროექტი_1

მისაღმება მსოფლიოს-Hello World!

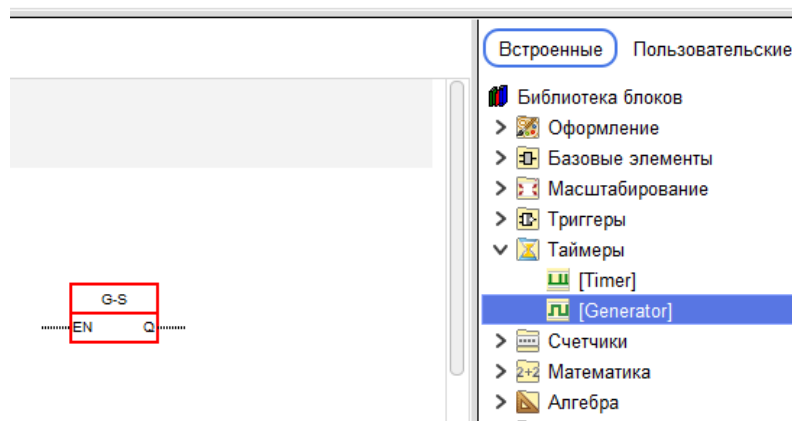


ტრადიციულად Hello World! ტიპის პროგრამები გამოიყენება კოდირების პროცესის მუშაობის ილუსტრირებისათვის, ასევე სისტემის ან ენის გამართული მუშაობის

უზრუნველყოფისათვის. ჩვეულებრივად კი, ასეთ პროგრამებს წერენ ახალგამომცხვარი პროგრამისტები იმიტომ, რომ იმათაც კი, ვისაც აქვთ მცირე გამოცდილება პროგრამების წერაში, ანდა სულაც არა აქვთ, შეუძლიათ ადვილად და სწორად შეასრულონ Hello World!

არდუინოს შემსწავლელებისთვისაც გამოიძებნა საშუალება დაამუშავონ ვიზუალური პროგრამა გრაფიკული ენების გამოყენებით, რაც თანასწორუფლებიანი აღმოჩნდა პროგრამისტების მიერ დაწერილი პროგრამისა Hello World! ეს არის შუქდიოდის ციმციმი ერთი ჰერცის სიხშირით, რითაც დამწყები არდუინისტი ესალმება მსოფლიოს, აი მეც ვიწყებ დაპროგრამების შესაწავლას და ელოდეთ ჩემს წარმატებებსო. ამ პრინციპებიდან გამომდინარე, პირველ პროექტში წარმოდგენილი იქნება შუქდიოდის ციმციმის მარტივი პროგრამა შესრულებული FLProg გარემოში, მაგრამ პროგრამის მცირედ გართულების ხარჯზე დავამატებთ დისპლეის ჩიპზე HD44780, რომლის პირველ სტრიქონშიც ჩავწერთ ჩვენთვის უკვე კარგად ცნობილ მისალმებას Hello World! ხოლო მეორე სტრიქონში პროგრამის შემქმნელი პიროვნების სახელს, მაგ. I am Jemal.

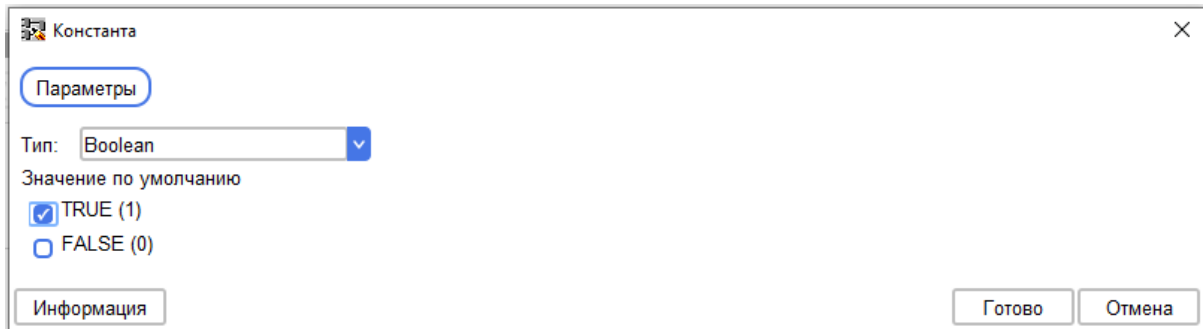
ახლა შევუდგეთ პროგრამის შექმნას. გავუშვათ FLProg, საუშაო ენად ავირჩიოთ FBD, ხოლო კონტროლერად Arduino Uno. გადავიდეთ ბლოკების ბიბლიოთეკის ზონაში-3 და დავაწკაპუნოთ ტაიმერების საქადალდეს-Таймеры, გამოსული ორი პიქტოგრამიდან ავირჩიოთ გენერატორი-[Generator] და გადმოვიტანოთ პროგრამის სამუშაო ზონაში (ფიგ.1.1).



ფიგ. 1.1 გენერატორის გადმოტანის პროცედურა (შექმნა) FLProg-ის სამუშაო ზონაში

ახლა შევუდგეთ ამ ბლოკის პარამეტრიზაციას (დაპროგრამებას). ჯერ დავაპროგრამოთ გენერატორის მუშაობის ნების დართვის შესასვლელი EN. მივიტანოთ ამ შესასვლელის ხაზზე თავის მაჩვენებელი და როდესაც იგი შეიცვლის ფერს და გახდება წითელი მაშინ გავხსნათ მისი კონტექსტური მენიუ, ანუ დავაჭიროთ თავის მარჯვენა ღილაკს თითი და ავირჩიოთ კონსტანტის ჩასმა-Вставить константу რის შედეგადაც ეკრანზე გამოვა კონსტანტის ფანჯარა-

Константа და მისი პარამეტრები-Параметры (ფიგ. 1.2). ამ ფანჯარაში კონსტანტის ტიპის ველი Tip: Boolean დავტოვოთ უცვლელად. თავის მარცხენა ღილაკით ჩავრთოთ ალამი TRUE(1) და მზადყოფნის ღილაკზე-Готово დაჭერის შემდეგ გენერატორის ბლოკის EN შესასვლელზე გაჩნდება წარწერა True. ეს იმას ნიშნავს, რომ პროგრამის არდუინოში ჩატვირთვის შემდეგ გენერატორი იმუშავებს უწყვეტად ყოველგვარი მართვის გარეშე.



ფიგ. 1.2 გენერატორის მუშაობაზე ნების დართვის ფანჯარა

ახლა საჭიროა დავაპროგრამოთ გენერატორის სხვა პარამეტრები. შევიდეთ ამ ბლოკის G-S ველში, დავაწკაპოთ მასზედ ორჯერ თავის მარცხენა ღილაკით და გამოსულ პარამეტრების ფანჯარაში (ფიგ. 1.3) გენერატორის ტიპად-Тип генератора: ჩამოშლადი სიიდან ავირჩიოთ სიმეტრიული მულტივიბრატორი-Симметричный мультивибратор.



ფიგ. 1.3 გენერატორის პარამეტრიზაცია

კონსტანტის ჩამრთველი-Константа და მილიწამების ჩამრთველი-Миллисекунды დავტოვოთ უცვლელად, Миллисекунды-ის ველში ჩავწეროთ 500 დავაჭიროთ ღილაკს Готово და ამით გენერატორის პარამეტრიზაცია დამთავრებული იქნება. შეფეგად დავინახავთ, რომ გენერატორის ბლოკზე G-S ველში შეიცვლება წარწერა და ჩაიწერება G-SM, ანუ საკმე გვექნება

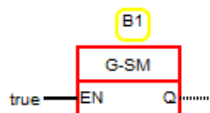
სიმეტრიულ მულტივიბრატორთან, რომელიც უწყვეტად გამოიმუშავებს იმპულსების და პაუზების თანმიმდევრობას 500 მილიწამის იმპულსისა და 500 მილიწამის პაუზის ხანგრძლიობებით, ანუ ერთწამიანი სიხშირის იმპულსებს. (იმპულსი ნიშნავს ძაბვას +5 ვოლტს, ხოლო პაუზა-ძაბვას 0 ვოლტს).

დავუსვათ ამ ბლოკს თავზე რაიმე წარწერა, რომ შემდეგში ადვილი იყოს მასზედ მითითება. ამისათვის დავაწკაპუნოთ ისევ ამ ბლოკზე ორჯერ თავგის მარცხენა ღილაკით, გამოსულ ფანჯარაში მოვნიშნოთ ველი წარწერა ბლოკზე-Надпись к блоку და გამოსულ ფანჯარაში მარცხნივ, შუაში ან მარჯვნივ ჩავწეროთ მაგ. ბლოკის ნომერი B1 (ფიგ. 1.4).



ფიგ. 1.4 იდენტიფიკაციის წარწერის დატანა ბლოკზე

ეს წარწერა შემდგომში აისახება ბლოკის თავზე (ფიგ. 1.5).

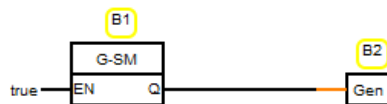


ფიგ. 1.5 წარწერის B1 ასახვა ბლოკზე

ახლა ამ გენერატორის გამოსასვლელი გავიტანოთ არდუინოს ერთერთ კონტაქტზე. ამიტომ, ჯერ უნდა შევქმნათ ასეთი გამოსასვლელი. შევიდეთ ტეგების ზონაში და დავაწკაპუნოთ გამოსასვლელების ნიშანს-Выходы და შემდეგ გამოსასვლელის დამატებას-Добавить Выход, სახელის ველში ჩავწეროთ ჩვენთვის მოსახერხებელი სახელი, მაგალითად Gen, ტიპის ველში ავირჩიოთ ციფრული-Цифровой, კონტაქტის ნომრის ველში ნებისმიერი კონტაქტი, რასაც პროგრამა შემოგვთავაზებს, მაგალითად 7, შემდეგ ჩამოვიტანოთ ეს ჩვენი შექმნილი გამოსასვლელი პროგრამის ტეგების ზონიდან სამუშაო ზონაში და მოვათავსოთ გენერატორის ბლოკის გასწვრივ. დავარქვათ ამ ბლოკს B2. ახლა საჭიროა B1 ბლოკის

გამოსასვლელი შევუერთოთ B2 ბლოკის შესასვლელს. ამისათვის, მივიტანოთ თავის მარჯვენა B1 ბლოკის გამოსასვლელზე და როდესაც ეს გამოსასვლელი ფერს შეიცვლის და გახდება წითელი, მაშინ დავაჭიროთ თავის მარცხენა ღილაკს თითი და გადავადგილოთ B2 ბლოკის შესასვლელამდე. როდესაც ეს შესაცვლელიც შეიცვლის ფერს და გახდება წითელი აუშვათ თითი თავის ღილაკს და დავრწმუნდეთ, რომ ეს ორი ბლოკი ერთმანეთთან შეერთებული იქნება შავი ხაზით (ფიგ. 1.6). ეს ოპერაცია ასევე წარმატებით შეგვეძლო ჩაგვეტარებინა უკუ მიმართულებით, ე.ი. B2 ბლოკის შესასვლელიდან B1 ბლოკის გამოსასვლელამდე.

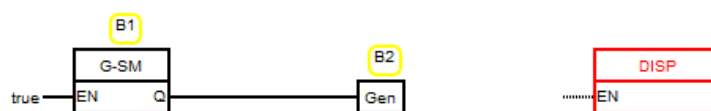
იმ შემთხვევაში თუ ჩვენ გვინდა უკვე გავლებული შემაერთებელი ხაზის გაუქმება, უნდა წავიდეთ უკუ მიმართულებით. უნდა მივიყვანოთ თავის მარჯვენა ბლოკის შესასვლელთან და როდესაც ეს შესასვლელი გაწითლდება გადავადგილოთ თავის მარჯვენა სხვა მიმართულებით და შემაერთებელი ხაზი წაიშლება.



ფიგ. 1.6 გენერატორის ბლოკის გამოსასვლელის შეერთება არდუინოს გამოსასვლელი ბლოკის შესასვლელთან.

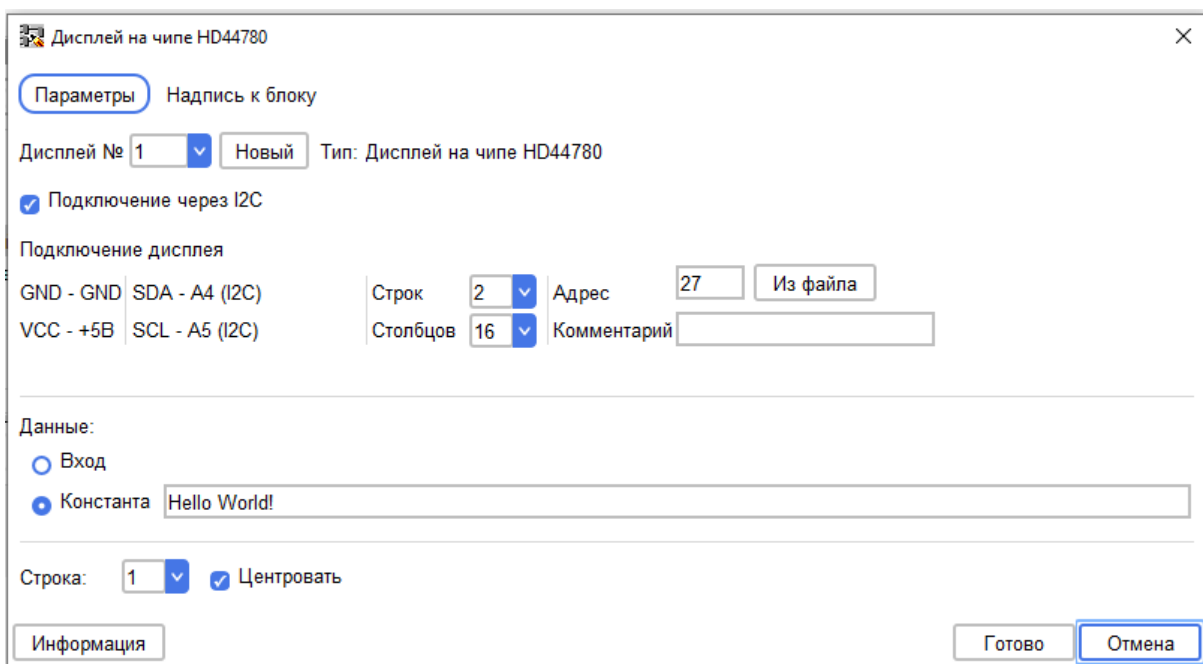
ამით გენერატორის ბლოკის დაპროგრამება და სიგნალების გაყვანა არდუინოს კონტაქტზე დამთავრებულია. ჩვენ უნდა შევაერთოთ არდუინოს ეს მე-7 გამოსასვლელი ფიზიკურად შუქდიოდის ანოდს შემზღვეველი წინააღმდეგობის გავლით, ნომინალით 220 ომი, კათოდი დავსვათ მიწაზე. ჩავტვირთოთ ეს პროგრამა არდუინოში და დავინახავთ, რომ შუქდიოდი დაიწყებს ციმციმს ერთი ჰერცის სიხშირით, ანუ ნახევარი წამი ანთებული იქნება და ნახევარი წამი ჩამქვრალი. ამით ჩვენ შევასრულეთ ჩვენი პროექტის პირველი ნაწილი და აპარატურით გამოვსახეთ Hello World!-ის შინაარსის შემცველი მისალმება შუქდიოდის ციმციმით.

ახლა კი გადავიდეთ პროექტის მე-2 ნაწილის განხორციელებაზე. ამ მიზნით შევიდეთ ბლოკების ბიბლიოთეკის დისპლეების საქაღალდეში-Дисплей და სამუშაო ზონაში გადმოვიტანოთ დისპლეი ჩიპზე-Дисплей на чипе HD4480 (ფიგ. 1.7).



ფიგ. 1.7 დისპლეის ბლოკის გადმოტანა სამუშაო ზონაში

დავაწკაპუნოთ ამ ბლოკზე ორჯერ და ეკრანზე გამოვა მისი პარამეტრიზაციის ფანჯარა, სადაც დავაჭიროთ ღილაკს ახალი-Новый, რაც გამოიწვევს დისპლეის ნომრის ველში-№ Дисплей ავტომატურად ერთიანის ჩაწერას, შემდეგ ჩავრთოთ მიერთების ალამი I2C-ს გავლით-Подключение через I2C, მისამართის ველში-Адрес ჩავწეროთ 27, ჩავრთოთ ცენტრირების ალამი-Центровать, მონაცემთა-Данные-ს კონსტანტის ველში-Константа ჩასვწეროთ ჩვენთვის უკვე კარგად ნაცნობი მისალმება ლათინური ასოებით Hello World! სტრიქონის ველში-Строка ავირჩიოთ პირველ სტრიქონი და ჩავწეროთ ერთიანი (ფიგ. 1.8) და დავაჭერთ მზადყოფნის ღილაკს-Готово.

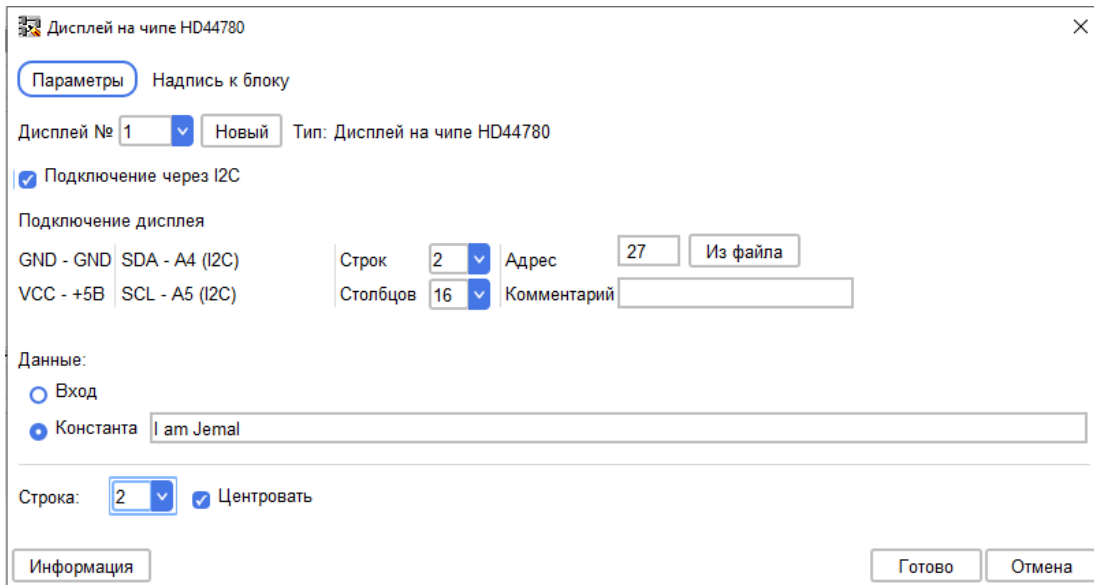


ფიგ. 1.8 დისპლეის ჩიპზე HD4480 პირველი სტრიაქონის პარამეტრიზაციის ფანჯარა

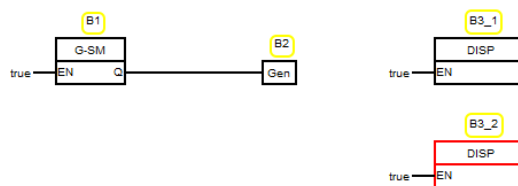
აქ გასათვალისწინებელია ის, რომ თვითონ დისპლეი შეიძლება იყოს ორი მოდიფიკაციის, ჩამუნებული ინტერფეისით I2C-ით და მის გარეშე. უმჯობესია შევიძინოთ და გამოვიყენოთ ჩამუნებული ინტერფეისით, რადგან იგი უზრუნველყოფს უფრო ნაკლებ შეერთებებს არდუინოსთან და მარტივ დაპროგრამებას და კიდევ ის, რომ დისპლეის მისამართი-Адрес 27 ნომერი ქარხნულად აქვს მინიჭებული და ყველა დისპლეისთვის ერთიდაიგივეა. ამის შემდეგ დისპლეის ბლოკის შესასვლელზე დავსვათ True, ისევე როგორც ეს გავაკეთეთ გენერატორის პარამეტრიზაციის დროს.

იმისათვის, რომ ვაფორმიროთ მეორე სტრიქონიც, უნდა დავაკოპიროთ ეს ბლოკი და დავსვათ მისი ასლი პირველ ბლოკთან მახლობლობაში, გავხსნათ მისი პარამეტრიზაციის

ფანჯარა და აქ შევიტანოთ მხოლოდ ორი ცვლილება: 1) კონსტანტას ველში Hello World!-ის ნაცვლად ჩავწეროთ I am Jemal (თქვენ უნდა ჩაწეროთ თქვენი სახელი; 2) სტრიქონის ველში აირჩიოთ ორიანი (ფიგ. 1.9.) და დავაჭიროთ ღილაკს Готово. არ დაგვავიწყდეს, რომ მისამართის ველში-Адрес, აქაც, თუ არ არის ჩაწერილი, უნდა ჩავწეროთ რიცხვი27. ამით დისპლეის პარამეტრიზაცია დამთავრებული იქნება და პროგრამა მიიღებს საბოლოო სახეს (ფიგ. 1.10).

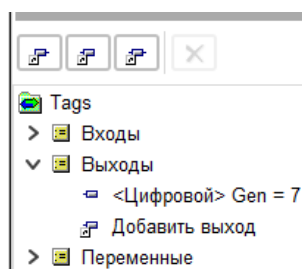


ფიგ. 1.9 დისპლეის ჩიპზე HD4480 მეორე სტრიაქონის პარამეტრიზაციის ფანჯარა



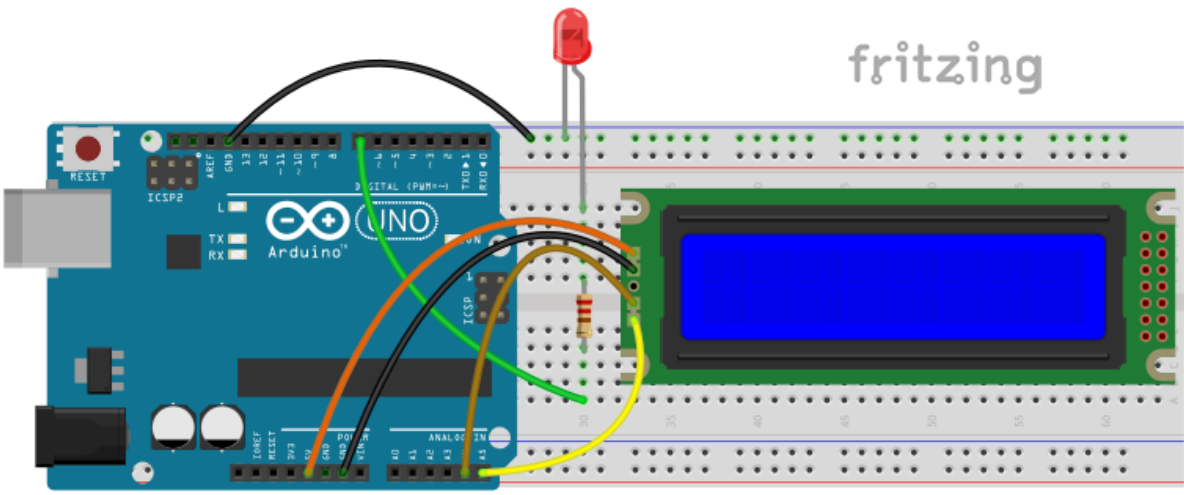
ფიგ. 1.10 პროგრამის საბოლოო სახე

ფიგ. 1.11-ზე გამოსახულია პროექტის ტეგების ზონის ფანჯარა



ფიგ. 1.11 პროექტის ტეგების ზონის ფანჯარა

დისპლეის შეერთებისათვის არდუინოსთან გამოიყენება ოთხი კონტაქტი. ორი კონტაქტი-GND და 5v გამოიყენება დისპლეის კვებისათვის 5 ვოლტით და ისინი უნდა შეურთდნენ გამტარებით არდუინოს GND და 5v კონტაქტებზე შესაბამისად, ორი შემდეგი კონტაქტი კი SDA და SCL გამოიყენება I2C ინტერფეისის განხორციელებისათვის ანუ დისპლეის არდუინოსთან ინფორმაციის გაცვლა/გამოცვლისათვის და როგორც გვავალდებულებს FLProg პროგრამა ისინი უნდა შეურთდნენ გამტარებით Arduino Uno-ს A4 და A5 კონტაქტებთან შესაბამისად (იხ. ფიგ. 1.12). ამის შემდეგ უნდა ჩავტვირთოთ პროგრამა არდუინოში და გაუშვათ. გაშვების შემდეგ დისპლეის პირველ სტრიქონში უნდა აისახოს წარწერა-Hello World!, მეორე სტრიქონში კი-I am Jemal და ამასთანავე შუქდიოდი უნდა ციმციმებდეს ერთი ჰერცის სიხშირით 500 მილიწამი ანთებითა და 500 მილიწამი ჩაქრობით. პროექტში გამოყენებული ელექტრონული კომპონენტების შეერთებები ასახულია Fritzing სქემაზე (ფიგ. 1.12).



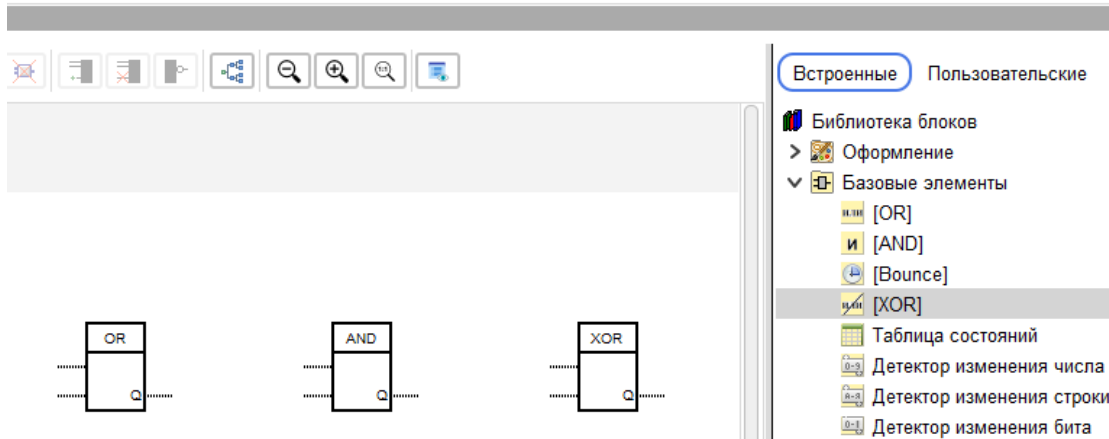
ფიგ. 1.12 პირველი პროექტის Fritzing სქემა

პროექტი_2

ბაზური ელემენტების მართვა

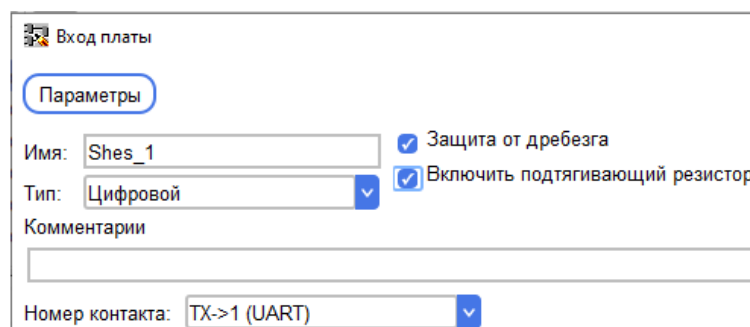


გავუშვათ FLProg პროგრამა, დაპროგრამების ენად ავირჩიოთ-FBD, ხოლო კონტროლერად-Arduino Uno. გადავიდეთ ბლოკების ბიბლიოთეკის ზონაში გავხსნათ ბაზური ელემენტების საქაღალდე-Базовые элементы და პროგრამის სამუშაო ზონაში გადმოვიტანოთ ლოგიკური ბაზური ელემენტები [OR], [AND] და [XOR] (ფიგ. 2.1.).



ფიგ. 2.1. ბაზური ელემენტების გადმოტანა

ტეგების ზონაში შევქმნათ ექვსი ციფრული შესასვლელი და სამი ციფრული გამოსასვლელი: Shes_1...Shes_6, Gam_1...Gam_3. რადგან ჩვენ მექანიკურ დილაკებთან გვაქვს საქმე, ამიტომ უმჯობესია თვითოეულ ამ შესასვლელზე მოვნიშნოთ ალმები:



ფიგ. 2.2 ალმების მონიშვნა შესასვლელზე

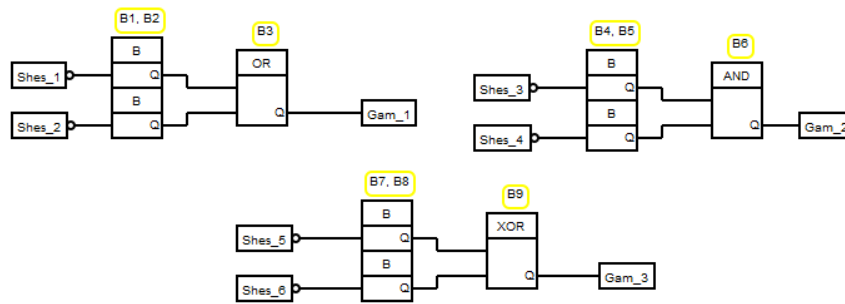
1.. ალამი Защита от дребезга-დაცვა ყანყალისაგან (ფიგ. 2.2). აქ იგულისხმება დილაკის მექანიკური კონტაქტების ყანყალი ანუ მათი მექანიკური მრავალჯერადი შეკვრა-გახსნა როგორც დილაკზე თითის დაჭერის, ასევე აშვების შემთხვევაში, რამაც შეიძლება წარმოქმნას ხელშეშლების დამატებითი იმპულსები და უარყოფითი გავლენა იქონიოს ელექტრონული სქემის მუშაობაზე;

2. ალამი Включить подтягивающий резистор-ამომქაჩავი რეზისტორის ჩართვა. აქ იგულისხმება, რომ როდესაც დილაკზე თითის დაჭერა ხდება მაშინ დილაკის გამოსასვლელზე ჯდება ნულოვანი პოტენციალი, ხოლო როდესაც თითის დაჭერა არ ხდება, მაშინ დილაკის

გამოსასვლელი ჰაერში რჩება და ეს არ შეიძლება, საჭიროა როგორღაც დავსვათ მაღალი პოტენციალი ლოგ. „1“ ამ გამოსასვლელზე, ამიტომ საჭიროა მიუერთდეს რეზისტორის ერთი ბოლო ამ გამოსასვლელზე, რომლის მეორე ბოლოც მიუერთდება +5 ვოლტს და ეს ხდება პროგრამულად ფიგ. 2.2-ზე ნაჩვენები აღმის მონიშვნით.

აქ უნდა აღინიშნოს კიდევ ერთი მომენტი, როდესაც ღილაკის ყანყალი ხანგრძლივია, მაშინ ზემოთმოყვანილი აღმის მონიშვნები არ არის საკმარისი ხელშეშლების იმპულსების სრულად ჩახშობისათვის. ამიტომ სქემაში შემოაქვთ ყანყალის ჩახშობის დამატებითი ბლოკი [Bounce], რომელიც მდებარეობს იმავე საქალაღდეში, რომელშიც ბაზური ბლოკებია მოთავსებული და რომელიც ახშობს ხელშეშლის იმპულსებს 40 მილიწამის განმავლობაში. ამ ბლოკს ათავსებენ შესასვლელელი კონტაქტების გამოსასვლელებზე (B1, B2, B4, B5, B7, B8).

დავამატოთ ზემოთმოყვანილ გამოსახულებას (ფიგ. 2.1) შესასვლელები, გამოსასვლელები და ზემოთნახსენები ბლოკები [Bounce] და მიუერთოთ ისინი ერთმანეთს ნაჩვენები სქემის მიხედვით (ფიგ. 2.3).

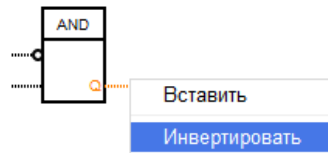


ფიგ. 2.3 ბაზური ელემენტების მართვის პროგრამა

ამ პროგრამაში არის კიდევ ერთი ნიუანსი. რადგან საწყის მომენტში ღილაკებზე თითის დაჭერა არ ხდება და შესასვლელი კონტაქტებზე Shes_1 ... Shes_6 დაყენებული გვაქვს ამომქაჩავი რეზისტორები, ამიტომ ამ კონტაქტებზე წარმოიქმნება მაღალი პოტენციალები, ანუ ლოგიკური ერთიანები, რაც იწვევს არასწორ ფუნქციონირებას. ჩვენ კი გვინდა, რომ როდესაც ღილაკს დავაწვებით მაშინ გამოიმუშავდეს სასიგნალო იმპულსები, ანუ აღნიშნულ კონტაქტებზე დაჯდეს ლოგიკური ერთიანები და შესრულდეს ჩვენს მიერ ალგორითმში ჩადებული მოქმედებები. ამიტომ ჩვენ ინვერტირება უნდა მოვახდინოთ ამ ბლოკების გამოსასვლელი სიგნალების, რათა უზრუნველყოთ მათ გამოსასვლელებზე ლოგიკური ერთიანები მხოლოდ მაშინ, როდესაც ღილაკებზე მოხდება თითების დაჭერა. სწორედ რომ ეს პატარა რგოლები Shes_1...Shes_6 ბლოკების გამოსასვლელებზე ნიშნავს სიგნალების

ინვერტირებას. თუ წერტილში დგას ლოგ. „1“ მაშინ სიგნალის ინვერტირების შემდეგ წარმოიქმნება ლოგ. „0“ და პირიქით.

სიგნალის ინვერტირება შეძლება მოვახდინოთ ნებისმიერი ბლოკის როგორც შესასვლელზე ასევე გამოსასვლელზე. ამისათვის უნდა მოვნიშნოთ ეს გამომეყვანი, დავაჭიროთ თავის მარჯვენა ღილაკს და გამოსული წარწერიდან-Вставить, Инvertировать ავირჩიოთ ეს უკანასკნელი (ფიგ. 4.).

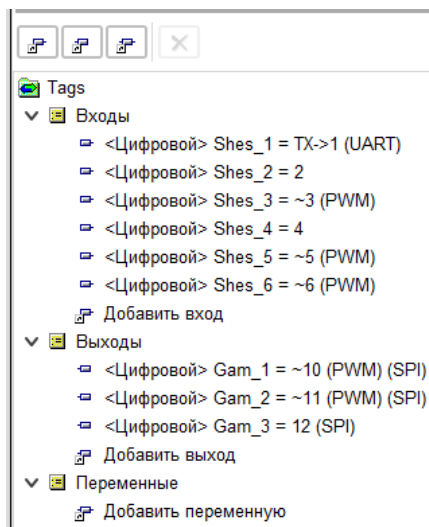


ფიგ. 2.4. სიგნალების ინვერტირება

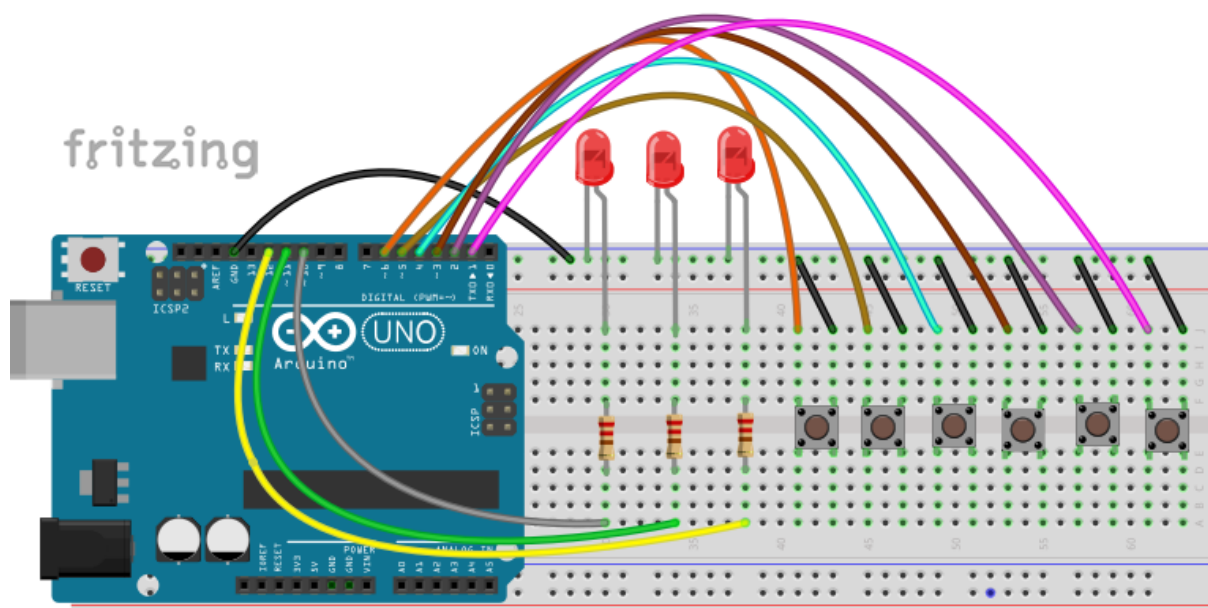
ახლა შევაერთოთ ფიზიკური ღილაკების კონტაქტების პირველი კონტაქტები მიწასთან ხოლო მეორე კონტაქტები-არდუინოს ჩვენს მიერ ფორმირებულ შესასვლელებთან Shes_1 ... Shes_6. მივუერთოთ აგრათვე არდუინოს გამოსასვლელები Gam_1...Gam_3 შემზღულდავი წინაქაღმდეგობების გავლით, ნომინალებით 220 ომი, შუქდიოდების ანოდებს, ხოლო კათოდები მივაერთოთ მიწას GND-ს. ამის შემდეგ მოვახდინოთ პროგრამის კომპილირება და არდუინოში ჩაწერა. ჩაწერის წარმატებით მოხდენის შემთხვევაში შევამოწმოთ ჩვენი ლოგიკური ბაზისური ელემენტების მუშაობის სისწორე. ჯერ შევამოწმოთ B3 ბლოკის მუშაობა. რადგან ეს ბლოკი ასრულებს ლოგიკური შეკრების ფუნქციას, ამიტომ თუ დავაწვებით მარტო Shes_1 კონტაქტთან შეერთებულ ღილაკს, მაშინ B3 ბლოკის გამოსასვლელზე დაჯდება ლოგ. „1“ რაც გამოიწვევს პირველი შუქდიოდის ანთებას, რომელიც Gam_1 გამოსასვლელზეა მიერთებული. თუ დავაწვებით მარტო Shes_2 კონტაქტთან შეერთებულ ღილაკს, მაშინ უნდა აინთოს იგივე პირველი შუქდიოდი და თუ დავაწვებით ორთავე Shes_1 და Shes_2 ღილაკებს, ამ შემთხვევაშიც უნდა აინთოს იგივე პირველი შუქდიოდი, თუ ორთავე ღილაკიდან ავუშვებთ თითებს, მაშინ შუქდიოდი უნდა იყოს ჩამქვრალი.

ახლა გადავიდეთ B6 ბლოკის შემოწმებაზე. რადგან ეს ბლოკი ასრულებს ლოგიკური გამრავლების ფუნქციას, ამიტომ მხოლოდ Shes_3-ს შესაბამის ღილაკზე ტიტის დაჭერისას ამ ბლოკის გამოსასვლელზე იქმნება ლოგ. „0“ და მე-2 შუქდიოდი, რომელიც Gam_2 გამოსასვლელზეა მიერთებული არ აინთება, ასევე მხოლოდ Shes_4-ის შესაბამის ღილაკზე დაჭერისას ეს მეორე შუქდიოდი ასევე არ აინთება და როდესაც ორთავე Shes_3 და Shes_4 ერთდროულად დავაჭერთ თითებს მხოლოდ ამ შემთხვევაში აინთება ეს შუქდიოდი.

ახლა გადავიდეთ B9 ბლოკის შემოწმებაზე. რადგან ეს ბლოკი ასრულებს ლოგიკურ ფუნქციას XOR, ამიტომ მხოლოდ Shes_5-ის შესაბამის ღილაკზე თითის დაჭერისას უნდა აინთოს Gam_2 გამოსასვლელთან მიერთებული შუქდიოდი ანდა მხოლოდ Shes_6 შესაბამის ღილაკზე თითის დაჭერისას უნდა აინთოს იგივე შუქდიოდი, მაგრამ თუ ორთავე ღილაკს დავაჭერთ ერთდროულად თითებს, ისევე როგორც თუ ორთავე ღილაკიდან მოვაშორებთ თითებს, მაშინ ეს მესამე შუქდიოდი უნდა იყოს ჩამქვრალი. ბაზური ბლოკების მართვის პროექტის ტეგების ზონის ფანჯარა ნაჩვენებია ფიგ. 2.6-ზე, ხოლო ბაზური ბლოკების მართვის Fritzing სქემა ასახულია ფიგ. 2.6-ზე.



ფიგ. 2.6 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 2.6 ბაზური ბლოკების მართვის Fritzing სქემა

პროექტი_3

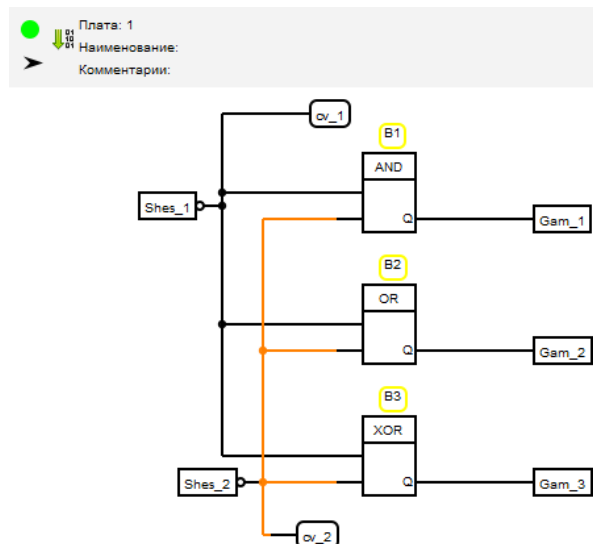
ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი

რ. ტოკპეიმის მიხედვით_1



ამ და კიდევ რამოდენიმე შემდგომ პროექტში ჩვენ განვიხილავთ მიკროპროცესორების არქიტექტურის, ფუნქციონირებისა და დაპროგრამების ცნობილი ამერიკელი სპეციალისტის რ. ტოკპეიმის სახელმძღვანელოში მოყვანილ სავარჯიშოებს ბაზური ლოგიკური ელემენტების ფუნქციონირების შესახებ, სადაც მათი მუშაობა განიხილება არა სტატიურ არამედ დინამიურ რეჟიმში.

გავუშვათ FLProg პროგრამა, დაპროგრამების ენად ავირჩიოთ FBD, ხოლო კონტროლერად- Arduino Uno. გადავიდეთ ბლოკების ბიბლიოთეკის ზონაში გავხსნათ ბაზური ელემენტების საქადალდე-Базовые элементы და პროგრამის სამუშაო ზონაში გადმოვიტანოთ ლოგიკური ბაზური ელემენტები [AND], [OR] და [XOR] (ფიგ. 3.1).



ფიგ. 3.1. ბაზური ელემენტების [AND], [OR] და [XOR] მართვის სქემა, პირველი დაფა

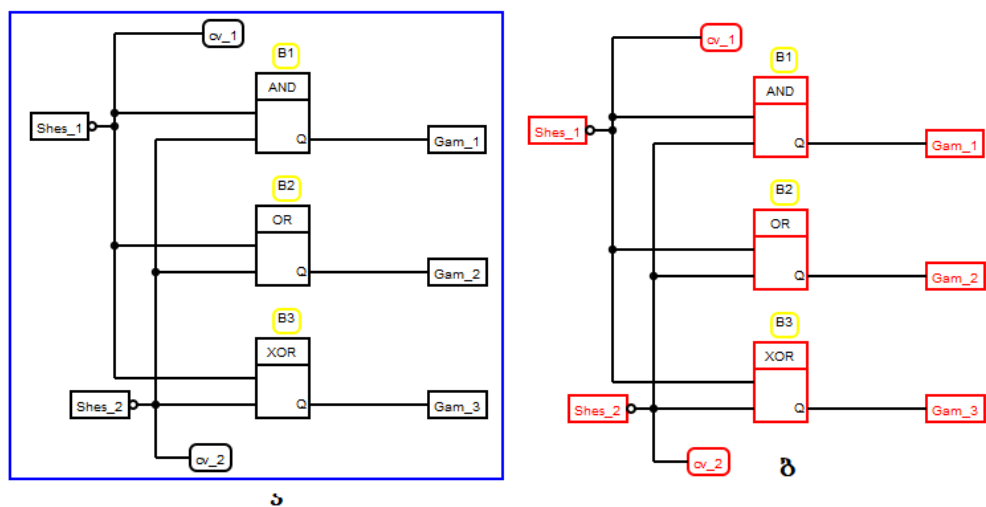
შეკმნათ: ა) ორი ციფრული შესასვლელი Shes_1 და Shes_2 სადაც მოვნიშნოთ ყანყალისაგან დაცვისა და ამომქაჩავი რეზისტორის აღმები (ფიგ. 3.2). მივეუერთოთ ეს შესასვლელები არდუინოს მე-2 და მე-3 კონტაქტებს; ბ) სამი ციფრული გამოსასვლელი: (ფიგ. 3.3) Gam_2, Gam_3 და Gam_3 და მივეუერთოთ ისინი არდუინოს მე-5, მე-6 და მე-7 კონტაქტებს; ორი Boolean-ის ტიპის ცვლადი: cv_1 და cv_2 (ფიგ. 3.4).

ფიგ. 3.2 შესასვლელების შექმნა

ფიგ. 3.3 გამოსასვლელების შექმნა

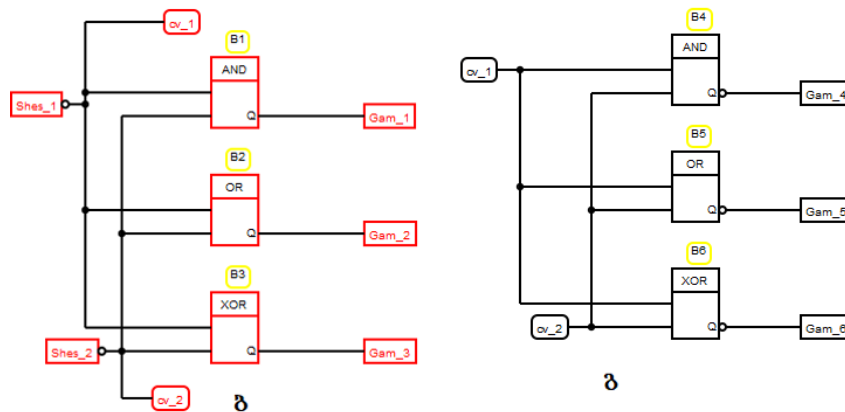
3.4 ცვლადების შექმნა

ახლა მოვნიშნოთ მთელი სქემა პირველ დაფაზე (ფიგ. 3.1) და დავაკოპიროთ. ამისათვის უნდა მივიყვანოთ თავგის მაჩვენებელი ნახაზის მარცხენა ზედა კუთხეში, დავაჭიროთ თავგის მარცხენა ღილაკს და გადავაადგილოთ მაჩვენებელი ნახაზის მარჯვენა ქვედა კუთხისაკენ თითის აულებლად. ამ დროს შეიქმნება ლურჯი ოთხკუთხედი (ფიგ. 3.5). რომელმაც უნდა მოიცვას მთელი ის ნახაზი (ფიგ. 3.5ა), ანდა ნახაზის ის ნაწილი რომლის კოპირებაც გვაქვს განზრახული. ამის შემდეგ აუშვებთ თითს თავგის ღილაკს და ისევ დავაჭერთ მას, რაც გამოიწვევს მთელი მონიშნული ნაწილის გაწითლებას (ფიგ. 3.5ბ), ამის შემდეგ ვაწკაპუნებთ ამ გაწითლებულ არეში ნებისმიერ ბლოკზე თავგის მარცხენა ღილაკით, ანუ შევდივართ კონტექსტურ მენიუში და ვირჩევთ კოპირებას-Копировать.



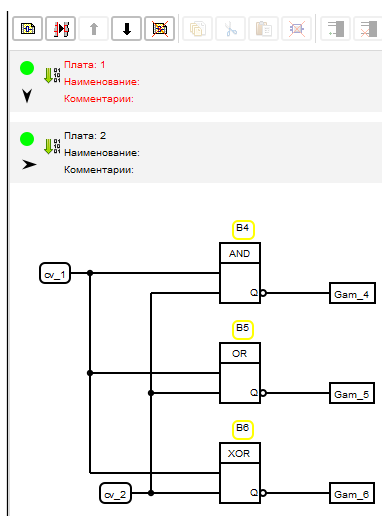
ფიგ. 3.5 ნახაზის კოპირების პროცედურა

ჩავსვათ ეს კოპირებული ნახაზი მე-2 დაფაში (ფიგ. 3.6ა) და მოვახდინოთ მისი კორექტირება. წავშალოთ ბლოკები Shes_1 და Shes_2, შევუცვალოთ ნუმერაცია B1,B2,B3 ბლოკებს და დავარქვათ მათ B4, B5 და B6 ბლოკები, შევუერთოთ ამ ბლოკების შესასვლელები cv_1 და cv_2 ცვლადებს, ისევე როგორც ისინი შეერთებული იყვნენ Shes_1 და Shes_2 შესასვლელებზე. წავშალოთ ბლოკები Gam_1, Gam_2 და Gam_3, ჩამოვიტანოთ Gam_4, Gam_5 და Gam_6 გამოსასვლელები თეგების ზონიდან და ისინი შევაერთოთ B4, B5 და B6 ბლოკების გამოსასვლელებზე.



ფიგ. 3.6 ახალი დაფის შექმნა და კოპირებული სქემის ჩასმა და კორექტირება

საბოლოოდ მე-2 დაფა მიიღებს ფიგურა 3.7-ზე ნაჩვენებ სახეს. ამრიგად ჩვენ გვაქვს ორი იდენტური ლოგიკური ბლოკების წყობა, რომლებსაც შესასვლელზე მიეწოდებათ აბსოლუტარად ერთნაირი ინფორმაციული თანმიმდევრობები, მაგრამ რომლებიც განსხვავდებიან იმით, რომ მათი გამოსასვლელი ინვერტირებული არიან და შესაბამისად მათი გამოსასვლელი სიგნალები იქნება განსხვავებული.



ფიგ. 3.7 ბაზური ელემენტების [AND], [OR] და [XOR] კონტროლის სქემა, მეორე დაფა

ახლა ვნახოთ თუ რაში მდგომარეობს ბაზური ელემენტების დინამიური კონტროლის მეთოდი ტოკპეიმის მიხედვით. შესასვლელი სიგნალები ამ შემთხვევაში განიცდიან ცვლილებებს უწყვეტად, კომბინაციებად, დროის თანაბარი ან არათანაბარი შუალედების მიხედვით. მაგ. ორი შესასვლელის შემთხვევაში ჯერ მიეწოდება ვთქვათ ლოგიკური ერთიანი და ერთიანი, შემდეგ ნოლი და ნოლი, შემდეგ ერთი და ნოლი შემდეგ ნოლი და ერთი, შემდეგ

კიდევ ნოლი და ნოლი, შემდეგ ერთი და ერთი და ა.შ. ჩავტვირთოთ არდუინოში ჩვენს მიერ დამუშავებული პროგრამა და დავტესტოთ იგი.

ასეთი მიდგომა შეგვიძლია გამოვსახოთ ცხრილების მიხედვით. ფიგ. 3.1.-ზე გამოსახული პროგრამის მიხედვით ცხრილი მიიღებს ასეთ სახეს (ცხრ. 3.1);

ცხრ. 3.1.

Shes_1	Shes_2	Gam_1	Gam_2	Gam_3
1	1	1	1	0
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

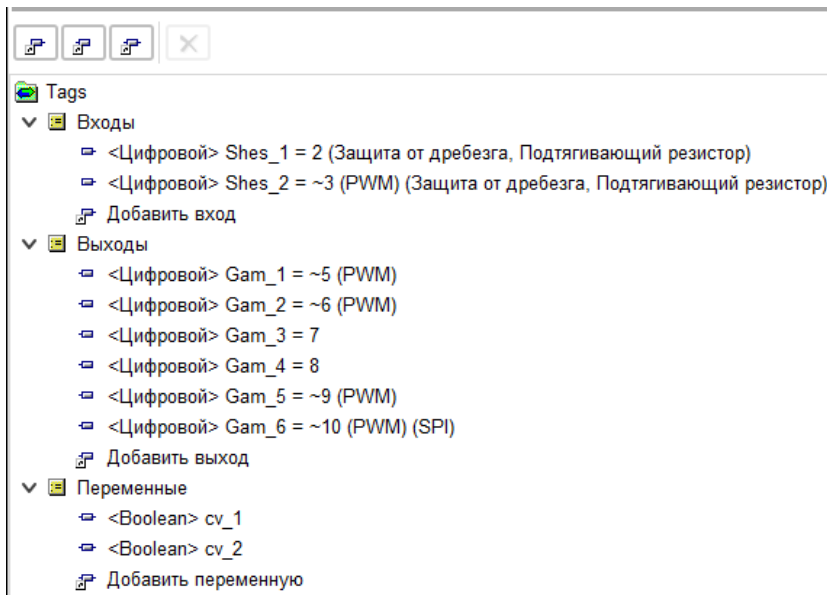
ამ შემთხვევაში უნდა წარმოვიდგინოთ, რომ არდუინოს პირველ და მეორე კონტაქტებზე (Shes_1 და Shes_2) შესასვლელებზე მიერთებულნი არიან დილაკები და დილაკზე თითის დაჭერა ცხრილში Shes_1 და Shes_2-ის უჯრაში ნიშნავს 1-ს, ხოლო დილაკიდან თითის აღება ნიშნავს 0-ს. ამასთანავე არდუინოს მე-5, მე-6 და მე-7 კონტაქტზე (Gam_1, Gam_2 და Gam_3) ჩვეულებრივ მიერთებულნი არიან შუქდიოდის ანოდები შემზღუდავი წინააღმდეგობების გავლით (220 ომი თვითოეული), ხოლო კათოდები შეერთებულნი არიან მიწასთან, მაშინ ცხრილის Gam_1, Gam_2 და Gam_3 უჯრაში 1 ნიშნავს შუქდიოდი ანთებულ მდგომარეობაშია, ხოლო 0 ნიშნავს, რომ შუქდიოდი ჩამქვრალია.

იგივე ცხრილი ფიგ. 3.7-ის შემთხვევისთვის მიიღებს სახეს (ცხრ. 3.2.).

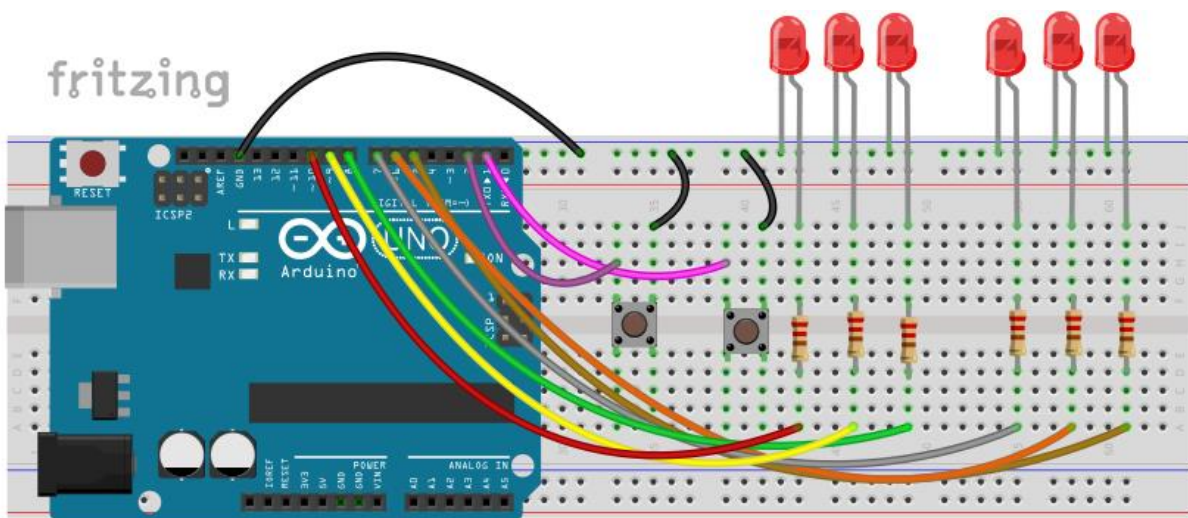
ცხრ. 3.2.

Shes_1	Shes_2	Gam_4	Gam_5	Gam_6
1	1	0	0	1
0	0	1	1	1
1	0	1	0	0
0	1	1	0	0
0	0	1	1	1

ფიგ. 3.8 ასახულია პროექტის ტეგების ფანჯარა, ხოლო ფიგ. 3.9-ზე წარმოდგენილია ბაზური ბლოკების კონტროლის Fritzing სქემა რ. ტოკხაიმის მიხედვით.



ფიგ. 3.8 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 3.9 ბაზური ბლოკების კონტროლის Fritzing სქემა რ. ტოკხაიმის მიხედვით.

პროექტი_4

ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი

რ. ტოკჰეიმის მიხედვით_2



ამ პროექტში განხილული იქნება ბაზური ლოგიკური ელემენტების კომბინაციები. როგორც ცნობილია ციფრული სისტემების აგება ხდება ლოგიკური ელემენტების კომბინაციებიდან. ასეთი კომბინაციები შეიძლება აღწერილ იქნას ჭეშმარიტების ცხრილით, ბულის ფუნქციით ანდა ლოგიკური სქემის მიხედვით.

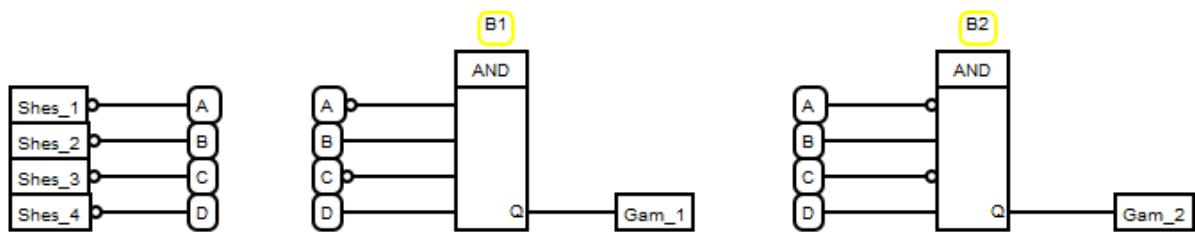
გავუშვათ FLProg პროგრამა, დაპროგრამების ენად ავირჩიოთ-FBD, ხოლო კონტროლერად-Arduino Uno. შევქმნათ ოთხი ციფრული შესასვლელი Shes_1, Shes_2, Shes_3, Shes_4 და ორი ციფრული გამოსასვლელი Gam_1 და Gam_2. შევეერთოთ ისინი არდუინოს მე-2, მე-3, მე-4, მე-5 და მე-7, მე-8 კონტაქტებს (პინებს) შესაბამისად. მივაერთოთ არდუინოს ეს შესასვლელი კონტაქტები გარე დილაკებს, ხოლო გამოსასვლელები კი შუქდიოდებს.

აქ კიდევ ერთჯერ უნდა გავუსვათ ხაზი იმ გარემოებას, რომ პროექტში გამოყენებული დილაკები წარმოადგენენ ნორმალურად ღია მექანიკურ კონტაქტებს, ანუ თუ დილაკზე თითი არ არის დაჭერილი, მაშინ კონტაქტები გახსნილია, თითის დაჭერის შემთხვევაში კი კონტაქტები შეიკვრება. თუ კი ჩვენ ამ კონტაქტის ერთ ბოლოზე დავსვავთ ნულოვან პოტენციალს (GND), და მეორე ბოლოს შევაერთებთ ერთერთ შესასვლელს, მაგ. Shes_1-ს, რომელიც თავის მხრივ წარმოადგენს არდუინოს მე-2 კონტაქტს, მაშინ დილაკზე თითის დაჭერის შემთხვევაში ამ მეორე კონტაქტზე მიეწოდება ნულოვანი პოტენციალი ანუ ლოგ. „0“, თითის აშვების შემთხვევაში კი, გამომდინარე იქიდან, რომ ამ მეორე კონტაქტზე პროგრამულად დასმულია ამომქაჩავი რეზისტორი - დაჯდება მაღალი პოტენციალი (5 ვ), ანუ ლოგ. „1“. ჩვენ კი გვინდა საწინააღმდეგოდ. ანუ თითის დაჭერის შემთხვევაში შევიდეს არდუინოში ლოგ. „1“, ხოლო აშვების შემთხვევაში კი ლოგ. „0“. ამის მისაღწევად საჭიროა ინვერტირება გავუკეთოთ დილაკებიდან გამოსულ სიგნალებს Shes_1, Shes_2, Shes_3, Shes_4 (ფიგ. 4.1.) და დავარქვათ ამ მათ A, B, C, D. ამის მისაღწევად შევიდეთ ტეგების ზონაში და

შევქმნათ ბულის ტიპის ცვლადები A, B, C, D, შემდეგ ჩამოვიტანოთ ეს ცვლადები პროგრამის სამუშაო ზონაში და მივეუერთოთ ჩვენს მიერ ინვერტირებულ სიგნალებს (ფიგ. 4.1). ახლა ავაგოთ კომბინაციური ლოგიკური სქემა ქვემოთ მოყვანილი გამოსახულების მიხედვით, რომელიც წარმოადგენს ლოგიკური გამრავლების ფორმულას.

$$\text{Gam}_1 = \overline{A} * B * \overline{C} * D$$

როგორც ამ გამოსახულებიდან ჩანს A და C სიმბოლოები ფორმულაში წარმოდგენილნი არიან თავზე გასმული ხაზებით, რაც ბულის ალგებრაში აღნიშნავს ინვერტირებას, ანუ A და C შესასვლელი სიგნალები გამოსახულების მიხედვით უნდა იქნენ ინვერტირებულნი. გადმოვიტანოთ პროგრამის სამუშაო ზონაში ბაზური ელემენტების ბლოკი [AND], ბლოკზე თავგის მარცხენა ღილაკით დაწკაპუნებით შევიდეთ მის კონტექსტურ მენიუში და ავირჩიოთ შესასვლელის დამატება-Добавить вход, კიდევ ერთჯერ გავიმეოროთ ეს პროცედურა და მივიღოთ ოთხშესასვლელიანი ლოგიკური ელემენტი AND (B1 ფიგ. 4.1).



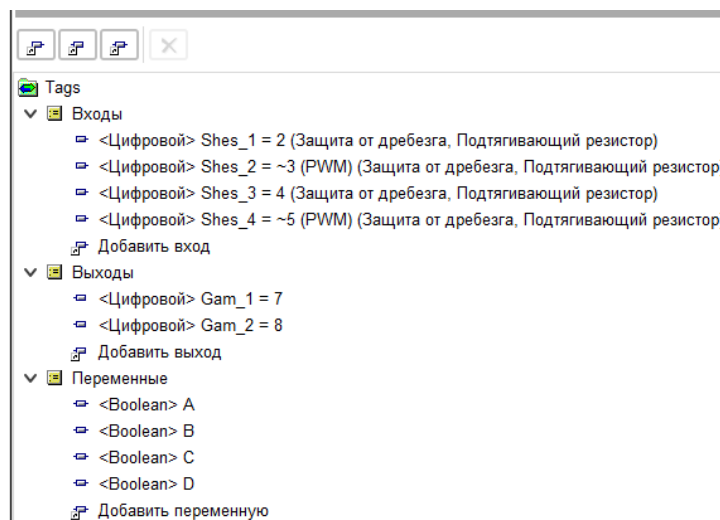
ფიგ. 4.1.

ტეგების ზონიდან კიდევ ერთჯერ გადმოვიტანოთ ოთხივე, ჩვენს მიერ უკვე შექმნილი ცვლადები A, B, C, D და შევუერთოთ ამ ოთხშესასვლელიანი ლოგიკური გამრავლების ბლოკის შესასვლელს. ამის შემდეგ გამოვიძახოთ A და C-ის შესასვლელის კონტექსტური მენიუები და მოვახდინოთ მათი ინვერტირება. ტეგების ზონიდან გადმოვიტანოთ აგრეთვე Gam_1 გამოსასვლელი და შევუერთოთ იგი B1 ბლოკის გამოსასვლელს.

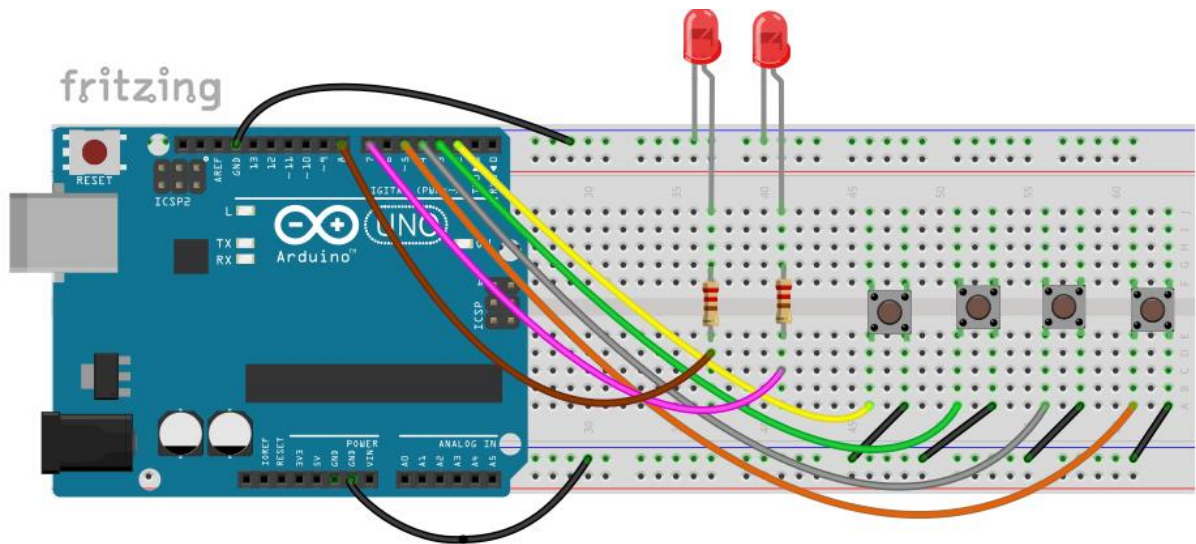
ახლა შევასრულოთ კიდევ შემდეგი სახის მოქმედება. კიდევ ერთჯერ შევქმნათ ოთხშესასვლელიანი ლოგიკური ელემენტი AND (B2). შევუერთოთ მისი შესასვლელები იგივე ცვლადებს A, B, C, D ხოლო გამოსასვლელი Gam_2 გამოსასვლელს. მოვაშოროთ A და C ცვლადების გამოსასვლელზე ინვერსირების რგოლები, ამისათვის შევიდეთ მათი გამოსასვლელის კონტექსტურ მენიუებში და გავიმეოროთ ისევ მათი ინვერტირების პროცედურა. ინვერსიის ინვერსია მოგვცევს არაინვერსირებულ სიგნალს. სამაგიეროდ ინვერსია გავუკეთოთ B2 ბლოკის ის შესასვლელს, რომლებიც გამოდიან A და C ცვლადებიდან (ფიგ. 4.1.).

გადავიდეთ ამ პროგრამის ტესტირებაზე. ამისათვის ჩავტვირთოთ იგი არდუინოს კონტროლერში და დავიწყოთ ტესტური კომბინაციების მიწოდება შესასვლელებზე. რადგან შესასვლელების რაოდენობა ოთხია Shes_1, Shes_2, Shes_3, Shes_4, ამიტომ კომბინაციების რაოდენობა იქნება ორი ხარისხად ოთხი ანუ თექვსმეტი, ესენია: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. გავიხსენოთ, რომ ამ კომბინაციებში სიმბოლო „0“ აღნიშნავს რომ ღილაკზე თითი არ არის დაჭერილი, ხოლო სიმბოლო „1“ აღნიშნავს, რომ ღილაკზე თითია დაჭერილი. მივაწვდით რა ამ კომბინაციებს თანმიმდევრობით ამ შესასვლელებზე დავადგენთ, რომ ორივე შუქდიოდი, რომლებიც Gam_1 და Gam_2 გამოსასვლელებზეა მიერთებულნი აინთებიან ზემოთმოყვანილი კომბინაციებიდან მხოლოდ და მხოლოდ ერთ კომბინაციაზე, კერძოდ 0101-ზე, იმიტომ რომ სქემაში პირველი ნოლიანიც ინვერტირდება და გადაიქცევა ერთიანად და მესამე ნოლიანიც ინვერტირდება და ისიც გადაიქცევა ერთიანად, ოთხი ერთიანის ლოგიკური ნამრავლი კი ერთიანია, რომელიც გადის გამოსასვლელზე და აანთებს შუქდიოდებს. ამით მტკიცდება აგრეთვე, რომ ორთავე სქემა რომელიც შეიცავს B1 ბლოკსაც და B2 ბლოკსაც აბსოლუტურად იდენტურია და ერთნაირი წარმატებით შეიძლება გამოყენებულ იქნას ლოგიკური სქემების აგების დროს.

პროექტის ტეგების ზონა ასახულია ფიგ. 4.2-ე, ხოლო პროექტის Fritzing სქემა ფიგ. 4.3-ზე.



ფიგ. 4.2 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 4.3 ბაზური ბლოკების კონტროლის Fritzing სქემა_2 რ. ტოკხაიმის მიხედვით

პროექტი_5

ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი

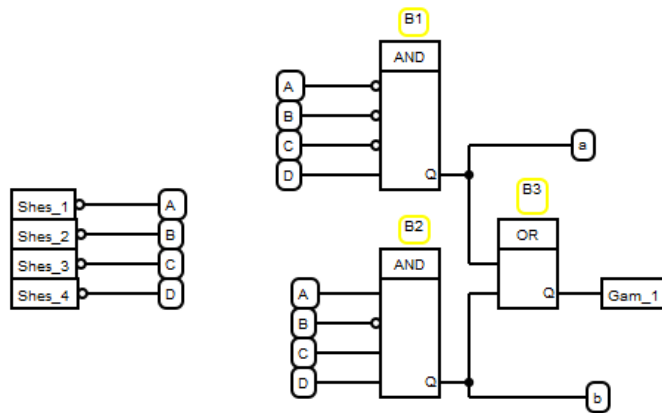
რ. ტოკჰეიმის მიხედვით_3



გავუშვათ FLProg პროგრამა, აქაც დაპროგრამების ენად ავირჩიოთ-FBD, ხოლო კონტროლერად-Arduino Uno. შევქმნათ ოთხი ციფრული შესასვლელი Shes_1, Shes_2, Shes_3, Shes_4 და ერთი ციფრული გამოსასვლელი Gam_1 შევუერთოთ ეს შესასვლელები არდუინოს მე-2, მე-3, მე-4, მე-5 კონტაქტებს, ხოლო გამოსასვლელი კი მე-7კონტაქტს. მივაერთოთ არდუინოს ეს შესასვლელი კონტაქტები ღილაკებს, ხოლო გამოსასვლელი კი შუქდიოდს. წინა პროექტის მსგავსად დავაინვერტოთ ეს შესასვლელები მათ კონტექსტურ მენიუში შესვლის გზით (ფიგ. 5.1), ტეგების ზონაში შევქმნათ A, B, C და D ორობითი (ბულის) ცვლადები ჩამოვიტანოთ პროექტის სამუშაო ზონაში და მივუერთოთ Shes_1, Shes_2, Shes_3 და Shes_4 კონტაქტებს.

ახლა შევქმნათ საკონტროლებელი ლოგიკური სქემა. ამ მიზნით ელემენტების ბიბლიოთეკიდან გადმოვიტანოთ ორი ლოგიკური ელემენტი [AND] და ერთი ლოგიკური

ელემენტი [OR]. გავზარდოთ ლოგიკური ელემენტების [AND]-ის შესასვლელები ოთხამდე თვითოეული. ავავოთ ლოგიკური კომბინაციური სქემა ფიგ. 5.1-ზე წარმოდგენილი სქემის მიხედვით.



ფიგ. 5.1 კომბინაციური ლოგიკური სქემა [AND] და [OR] ძირითადი ლოგიკური ელემენტების გამოყენებით

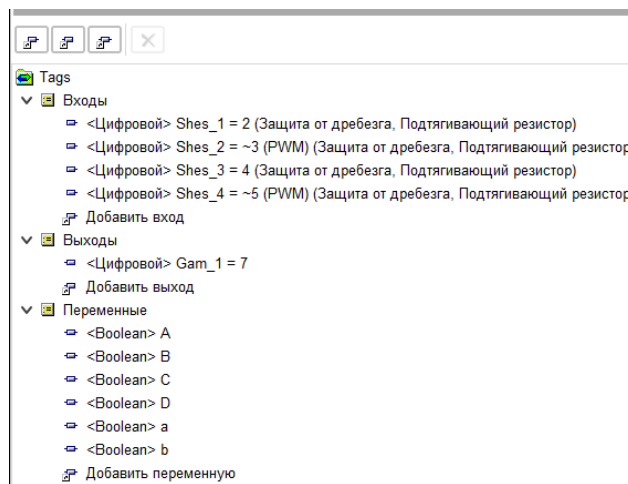
ამ პროექტში წინა პროექტიდან გამოიყენება იგივე ოთხშესასვლელიანი ლოგიკური ელემენტები B1 და B2, იგივე ოთხი შესასვლელი Shes_1, Shes_2, Shes_3 და Shes_4, რომლებიც შეერთებულია არდუინოს იგივე მე-2, მე-3, მე-4, მე-5 კონტაქტებთან და რომლებზედაც შეერთებულნი არიან იგივე ცვლადები A, B, C, D, მაგრამ განსხვავებით წინა პროექტისაგან A, B და C შესასვლელები პირველ [AND] ელემენტზე B1 ინვერტირებულია, ისევე როგორც B შესასვლელი ინვერტირებულია მეორე [AND] ელემენტზე B2, შემდეგ ამ ელემენტების გამოსასვლელები კი მიერთებულნი არიან ლოგიკური შეკრების ელემენტის [OR] შესასვლელებზე - ბლოკი B3.

ახლა დავწეროთ ლოგიკური გამოსახულებები ჯერ B1 ბლოკის მიერ შესასრულებელი ლოგიკური ფუნქციის, შემდეგ B2 ბლოკის მიერ შესასრულებელი ლოგიკური ფუნქციის, ხოლო შემდეგ B3 ბლოკის მიერ შესასრულებელი ლოგიკური ფუნქციის, ესენია:

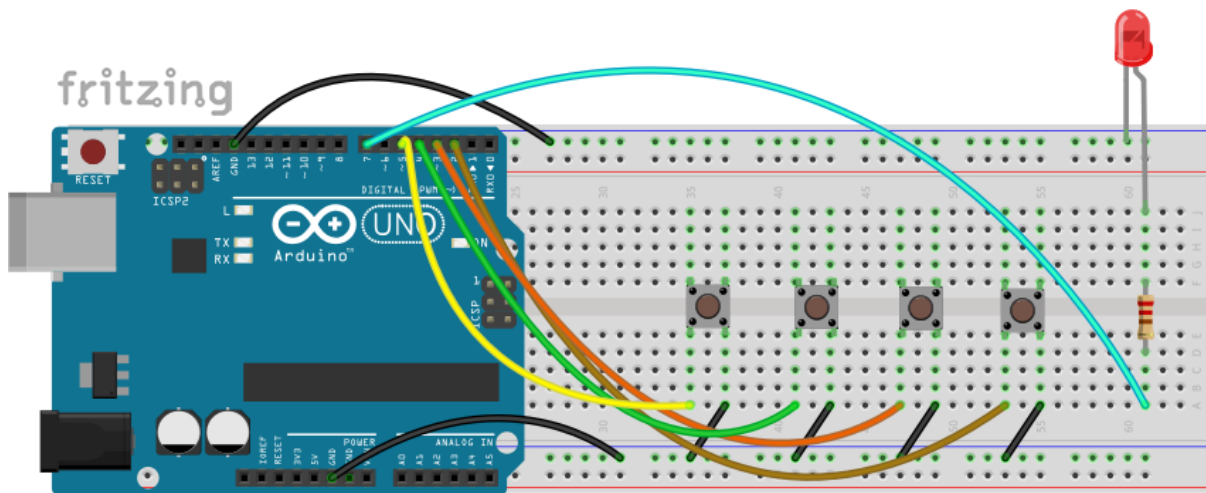
$$a = \bar{A} \bar{B} \bar{C} D \quad b = A \bar{B} C D \quad \text{Gam}_1 = a + b = \bar{A} \bar{B} \bar{C} D + A \bar{B} C D$$

შევუდგეთ ახლა პროგრამის ტესტირებას. ჩავტვირთოთ იგი არდუინოში და დავიწყოთ ტესტური კომბინაციების მიწოდება შესასვლელებზე. რადგან შესასვლელების რაოდენობა ოთხია Shes_1, Shes_2, Shes_3, Shes_4, ამიტომ ისევე როგორც წინა პროექტში, კომბინაციების რაოდენობა იქნება ორი ხარისხად ოთხი ანუ თექვსმეტი: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. გავიხსენოთ, რომ ამ კომბინაციებში სიმბოლო „0“ აღნიშნავს რომ ღილაკზე თითი არ არის დაჭერილი, ხოლო სიმბოლო „1“

აღნიშნავს, რომ ღილაკზე თითია დაჭერილი. მივაწვდით რა ამ კომბინაციებს შესასვლელებს დავადგენთ, რომ შუქდიოდი, რომლებიც Gam_1 გამოსასვლელზეა მიერთებული ანთება ზემოთმოყვანილი კომბინაციებიდან მხოლოდ ორ კომბინაციაზე, კერძოდ ესენია 0001-ზე და 1011, ეს იმიტომ ხდება, რომ B1 ბლოკი თავის გამოსასვლელზე გამოსცევს ლოგ. „1“-ს 0001 კომბინაციაზე, ხოლო ბლოკი B2-1011 კომბინაციაზე, B3 ბლოკი კი ლოგიკურად აჯამებს ამ ლოგ. „1“-ებს და თავის გამოსასვლელზე Gam_1 გამოსცემს ლოგ. „1“ ორივე ამ კომბინაციებზე, რაც იწვევს მასზედ მიერთებული შუქდიოდის ანთებას მხოლოდ ამ კომბინაციებზე. პროექტის ტეგების ზონა ასახულია ფიგ. 5.2-ზე, ხოლო პროექტის Fritzing სქემა-ფიგ. 5.3-ზე.



ფიგ. 5.2 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 5.3 ლოგიკური ელემენტების კომბინაციის Fritzing სქემა

პროექტი_6

ციფრული ტექნიკის ძირითადი ელემენტების კონტროლი

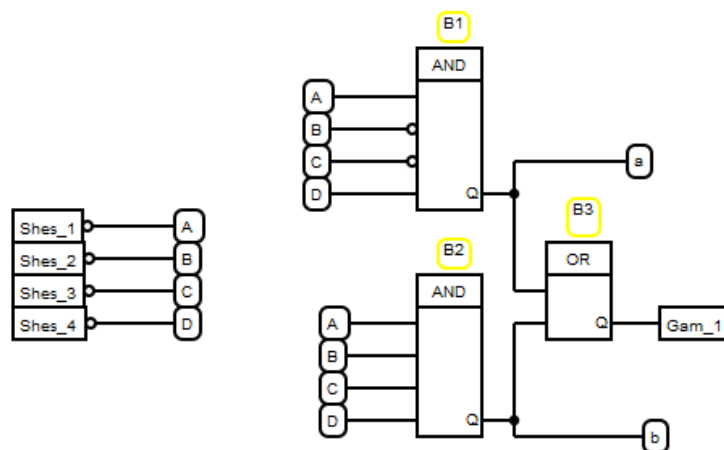
რ. ტოკპეიმის მიხედვით_4



ამ პროექტში განიხილება ლოგიკური კომბინაციური სქემა, რომელიც მეტნაკლებად ანალოგიურია წინა პროექტში განხილული სქემისა, იმ განსხვავებით, რომ B1 და B2 ბლოკების შესასვლელების ინვერტირებების ფორმა სხვადასხვაა. აქ ინვერტირებულანია B1 ბლოკის B და C შესასვლელები, ხოლო B2 ბლოკის შესასვლელები ინვერტირებას არ განიცდიან (ფიგ. 5.1). აღნიშნული სქემის მიერ შესრულებული ლოგიკური ფუნქცია იღებს შემდეგ სახეს:

$$a = A \cdot \bar{B} \cdot \bar{C} \cdot D \quad b = A \cdot B \cdot C \cdot D \quad \text{Gam}_1 = a + b = A \cdot \bar{B} \cdot \bar{C} \cdot D + A \cdot B \cdot C \cdot D$$

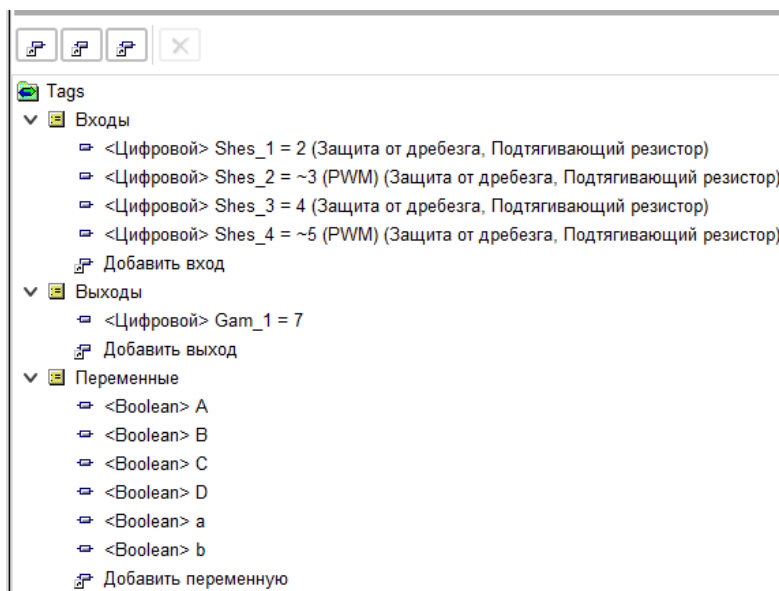
შევქმნათ პროექტი ფიგ. 5.1-ზე მოყვანილი სქემის მიხედვით და გადავიდეთ ამ პროგრამის ტესტირებაზე. ჩავტვირთოთ იგი არდუინოში და დავიწყოთ ტესტური კომბინაციების მიწოდება შესასვლელებზე. გავიხსენოთ, რომ რადგან შესასვლელების რაოდენობა ოთხია Shes_1, Shes_2, Shes_3, Shes_4, ამიტომ, კომბინაციების რაოდენობა იქნება ორი ხარისხად ოთხი ანუ თექვსმეტი, ესენია: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.



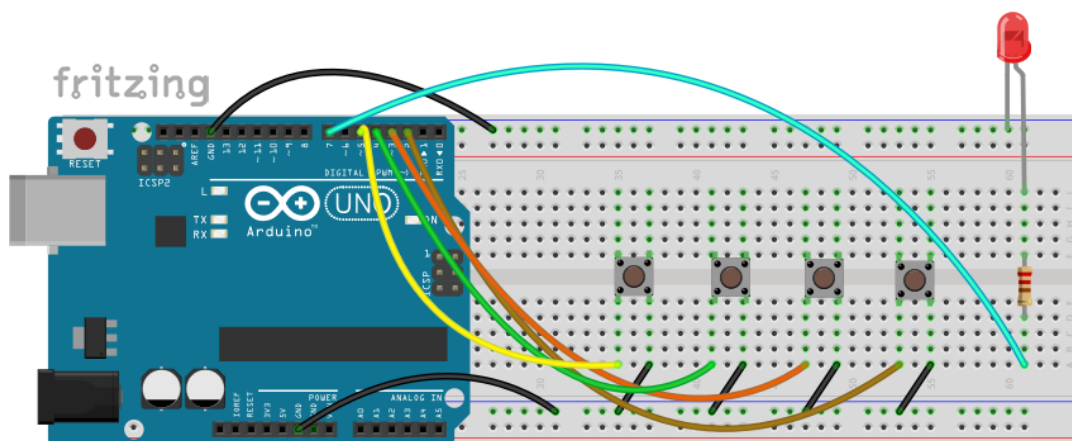
ფიგ. 5.1. კომბინაციური ლოგიკური სქემა [AND] და [OR] ძირითადი ლოგიკური ელემენტების გამოყენებით

როგორც წინა პროექტებიდან ცნობილია, ამ კომბინაციებში სიმბოლო „0“ აღნიშნავს იმას, რომ დილაკზე თითი არ არის დაჭერილი, ხოლო სიმბოლო „1“ აღნიშნავს იმას, რომ დილაკზე თითია დაჭერილი. თუ მივაწვდით ამ კომბინაციებს შესასვლელებს დავადგენთ, რომ შუქდიოდი, რომლებიც Gam_1 გამოსასვლელზეა მიერთებული აინთება ზემოთმოყვანილი კომბინაციებიდან მხოლოდ ორ კომბინაციაზე, კერძოდ ესენია 1001-ზე და 1111, ეს იმიტომ ხდება, რომ B1 ბლოკი თავის გამოსასვლელზე გამოსცევს ლოგ. „1“ -ს 1001 კომბინაციაზე, ხოლო ბლოკი B2-1111 კომბინაციაზე, B3 ბლოკი კი ლოგიკურად აჯამებს ამ ლოგ. „1“-ებს და გამოსასვლელზე გამოსცემს ლოგ. „1“ მხოლოდ ან ერთი ან მეორე ამ კომბინაციებიდან მიწოდების შემთხვევაში, რაც იწვევს მასზედ მიერთებული შუქდიოდის ანთებას.

პროექტის ტეგების ზონა ასახულია ფიგ. 5.2-ზე, ხოლო პროექტის Fritzing სქემა - ფიგ. 5.3-ზე.



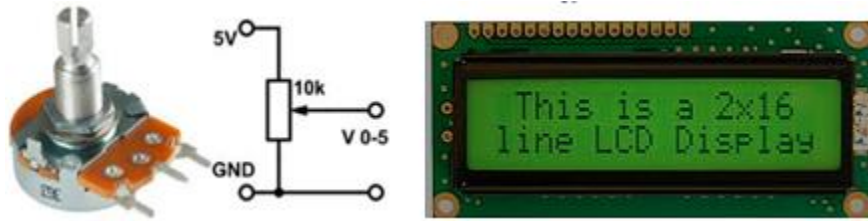
ფიგ. 5.2. პროექტის ტეგების ზონის ფანჯარა



ფიგ. 5.3 ლოგიკური ელემენტების კომბინაციის Fritzing სქემა

პროექტი_7

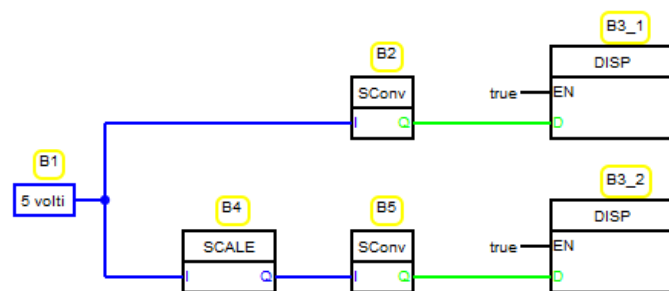
მასშტაბირების ბლოკი



გავუშვით პროგრამა FLProg, დაპროგრამების ენად ავიჩიოთ-FBD ხოლო კონტროლერის ტიპად-Arduino UNO. ტეგების ზონაში შევექმნათ ერთი ანალოგური შესასვლელი არდუინოს A2 კონტაქტის ბაზაზე, რომელსაც შემდგომში ფიზიკურად მიუერთდება პოტენციომეტრის შუალედური გამომყვანი, ხოლი განაპირა გამომყვანები მიუერთდება არდუინოს 5 ვოლტისა და მიწის (GND) კონტაქტებს. ჩამოვიტანოთ ეს A2 შესასვლელი პროგრამის სამუშაო არეში - ბლოკი B1 (ფიგ. 7.1).

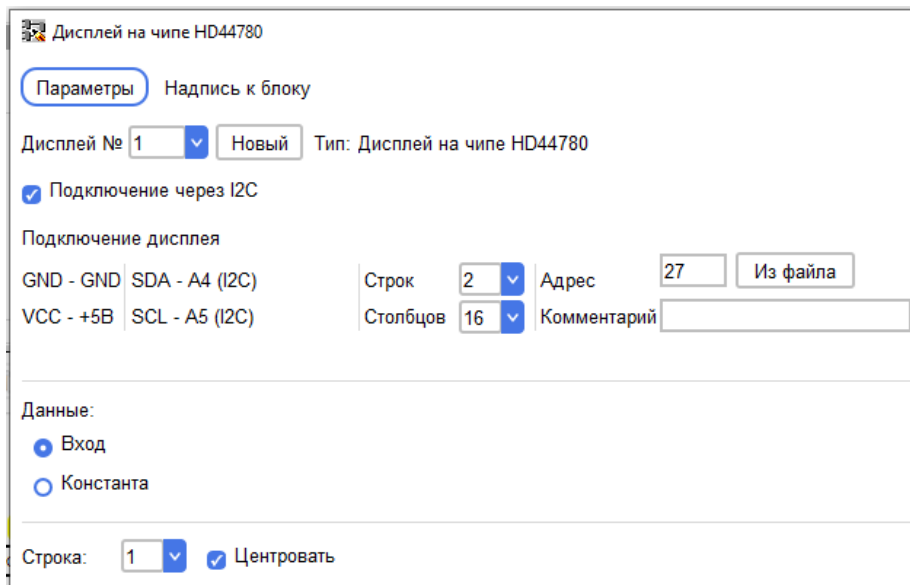
გავიხსენოთ, რომ პოტენციომეტრის ცოცხას გადაადგილების დროს მის შუალედურ გამომყვანზე, როდესაც ნაპირა გამომყვანებზე არსებობს მიწა და 5 ვოლტი, ძაბვა იცვლება 0 ... 5ვ-ის დიაპაზონში. ამ რეგულირებად ძაბვას არდუინო გარდაქმნის ათთანრიგა ორობითი კოდის სიზუსტით-ათობით კოდში 0...1023-მდე დიაპაზონში შესაბამისად. კოდის ეს მნიშვნელობები გამოდის B1 ბლოკის გამოსასვლელზე, რომელიც უნდა შევუერთოთ ერთის მხრივ დისპლეის (ბლოკი B3_1) პირველი სტრიქონის შესასვლელს კონვერტაციის ბლოკის B2 გავლით, ხოლო მეორეს მხრივ, მასშტაბირების ბლოკისა B4 და მეორე კონვერტაციის ბლოკების გავლით-დისპლეის (ბლოკი B3_2) მე-2 სტრიქონის შესასვლელს.

ახლა შევუდგეთ დისპლეის ბლოკის პარამეტრიზაციას. ამ მიზნით ბლოკების ბიბლიოთეკის დისპლეების საქაღალდედან-Дисплей პროგრამის სამუშაო ზონაში გადმოვიტანოთ დისპლეი ჩიპზე -Дисплей на чипе HD4480 - ბლოკი B3_1 (ფიგ. 7.1).



ფიგ. 7.1 მასშტაბირების ბლოკის მართვის პროგრამა

დავაწკაპუნოთ ამ ბლოკზე ორჯერ და ეკრანზე გამოვა მისი პარამეტრიზაციის ფანჯარა, სადაც დავაჭიროთ ღილაკს ახალი-Новый (ფიგ. 7.2), რაც გამოიწვევს დისპლეის ნომრის ველში-№ Дисплей ავტომატურად ერთიანის ჩაწერას, შემდეგ ჩავრთოთ ალამი-მიერთება I2C-ს გავლით-Подключение через I2C, მისამართის ველში-Адрес ჩავწეროთ 27, ჩავრთოთ ცენტრირების ალამი-Центровать. ამით დისპლეის პირველი სტრიქონის პარამეტრირება დამთავრებული იქნება.

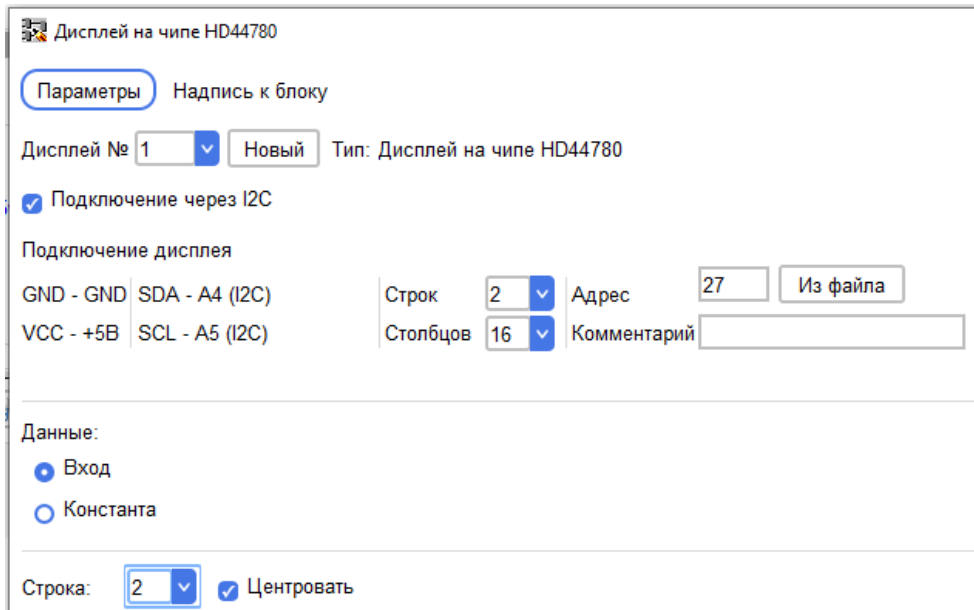


ფიგ. 7.2 დისპლეის ჩიპზე HD44780 პირველი სტრიქონის პარამეტრიზაციის ფანჯარა

აქ გასათვალისწინებელია ის გარემოება, რომ თვითონ დისპლეი შეიძლება იყოს ორი მოდიფიკაციის, ჩაშენებული ინტერფეისით I2C-ით და მის გარეშე. უმჯობესია შევიძინოთ და გამოვიყენოთ ჩაშენებული ინტერფეისით, რადგან იგი უზრუნველყოფს უფრო ნაკლებ შეერთებებს არდუინოსთან და მარტივ დაპროგრამებას, გარდა ამისა, მხედველობაშია მისაღები, რომ დისპლეის მისამართის ნომერი-Адрес ყველა დისპლეის მინიჭებული აქვს ქარხნულად (სხვა ნომრით ჩვენ არ შეგვხვედრია, თუმცა არ არის გამორიცხული) და ყველა დისპლეისთვის ერთიდაიგივეა და არის 27. ამიტომ მისამართის ველში-Адрес უნდა ჩავწეროთ 27. მონაცემთა ველში ჩავრთოთ შესასვლელი-Вход.

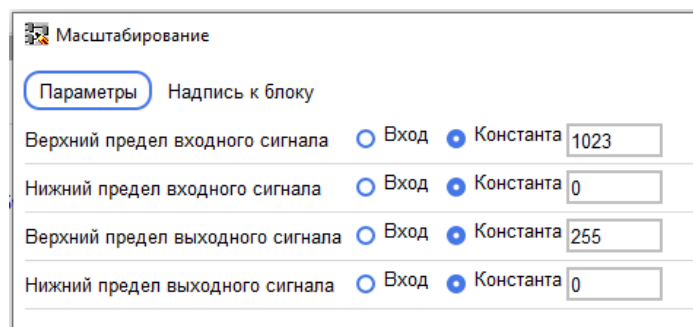
ამის შემდეგ გადავიდეთ დისპლეის მეორე სტრიქონის დაპროგრამებაზე, ამისათვის მოვნიშნოთ ამ დისპლეის ბლოკი B3_1 დავაკოპიროთ და ჩავსვათ მის მახლობლად. შევუცვალოთ ბლოკს აღნიშვნა და დავარქვათ B3_2. დავაწკაპუნოთ მასზედ და გავხსნათ მისი პარამეტრიზაციის ფანჯარა (ფიგ. 7.3). ამ ფანჯარაში ყველა პარამეტრს დავტოვებთ

უცვლელად, შევცვლით მხოლოდ სტრიქონის ველს-Строка და შესაბამის ველში ჩავწერთ ორს. ამით დისპლეის ორივე სტრიქონის პარამეტრიზაცია დამთავრებული იქნება.



ფიგ. 7.3 დისპლეის ჩიპზე HD4480 მეორე სტრიქონის პარამეტრიზაციის ფანჯარა

ახლა უკვე უნდა შევუდგეთ ჩვენი პროექტის მთავარი მიზნის, მასშტაბირების ბლოკის დაპროგრამებას. შევიდეთ პროგრამის ბლოკების ბიბლიოთეკის მასშტაბირების საქალაქში -Масштабирование და გადმოვიტანოთ ბლოკი [Scale] პროგრამის სამუშაო ზონაში B4. დავაწკაპუნოთ მასზე და გავხსნათ მისი პარამეტრიზაციის ფანჯარა (ფიგ. 7.4).



ფიგ. 7.4 მასშტაბირების ბლოკის პარამეტრიზაციის ფანჯარა

როგორც ფიგურიდან ჩანს, აქ წარმოდგენილია ოთხი პარამეტრი დასაპროგრამებლად: 1) შესასვლელი სიგნალის ზედა ზღვარი-Верхний предел входного сигнала; 2) შესასვლელი სიგნალის ქვედა ზღვარი-Нижний предел входного сигнала; 3) გამოსასვლელი სიგნალის ზედა ზღვარი-Верхний предел выходного сигнала; 4) გამოსასვლელი სიგნალის ქვედა ზღვარი-Нижний предел выходного сигнала. ამ ოთხ ველში მონაცემები შეიძლება ჩაიწეროს როგორც

კონსტანტები თუ ჩვენ ეს მონაცემები წინასწარ არის ცნობილი ჩვენი ამოცანებიდან გამომდინარე ანდა შეიძლება მიწოდებულ იქნას შესასვლელებიდან-Вход. ჩვენ ამ პროექტში შემოვიფარგლებით კონსტანტებს წინასწარ ჩაწერით.

რადგან ჩვენ არდუინოს ანალოგურ შესასვლელზე A2 მიერთებული გვაქვს ცვლადი წინააღმდეგობა, რომლის შუალედურ გამოსასვლელზეც, წინააღმდეგობის ღერძის ბრუნვით, იცვლება ძაბვა 0...5 ვოლტის დიაპაზონში, ამ ძაბვას ზომავს არდუინო და იძლევა შედეგს 0...1023-ის დიაპაზონში რაც დაფიქსირდება A2 პინის გამოსასვლელზე, ამიტომ ჩვენ შესასვლელი სიგნალის ზედა და ქვედა ველებში უნდა დავტოვოთ ის რიცხვები, რომლებიც სიჩუმით არიან ჩაწერილნი ანუ 1023 და 0. აქვე აღვნიშნავთ, რომ ეს რიცხვები სხვა ამოცანისთვის შესაძლებელია იყოს სხვადასხვა და არ არის აუცილებელი რომ ზედა ზღვარი ყოველთვის მეტი იყოს ქვედა ზღვარზე. დავტოვოთ უცვლელად ის რიცხვებიც რომლებიც სიჩუმით არის მოცემული გამოსასვლელი სიგნალის ზედა და ქვედა ზღვრებისთვის ანუ 255 და 0, თუმცა აქაც აღვნიშნავთ, რომ ეს რიცხვები სხვა ამოცანისთვის შესაძლებელია იყოს სხვადასხვა. დავაჭიროთ მზადყოფნის ღილაკს-Готово, რითაც მასშტაბირების ბლოკის დაპროგრამებზე ამ ჩვენი პროექტისათვის დამთავრებული იქნება.

ახლა შევაერთოთ ეს შესასვლელი და გამოსასვლელი სიგნალები ჩვენს მიერ უკვე შექმნილ დისპლეის შესასვლელებთან შესაბამისად, მაგრამ აქ უნდა გავითვალისწინოთ ის გარემოება, რომ დისპლეი პირდაპირ რიცხვებს ვერ აღიქვავს, საჭიროა ეს რიცხვები გარდავექმნათ სტრიქონებათ. ამ მიზნის მისაღწევად პროგრამის ბლოკების ბიბლიოთეკის ტიპების კონვერტაციის საქაღალდედან-Конвертация типов, გადმოვიტანოთ სტრიქონების კონვერტაციის ბლოკი-Конвертация строк პროგრამის სამუშაო ზონაში პირველი სტრიქონისათვის, ბლოკი B2, შევექმნათ მისი ასლი მეორე სტრიქონისათვის და დავარქვათ მას სახელი B5, შევაერთოთ ეს ბლოკები ისე როგორც ეს ფიგ. 7.1-ზეა ნაჩვენები. ამის შემდეგ ვახდენთ პროგრამის კომპილიაციას, შემდეგ არდუინოში გადატვირთვას და გავუშვებთ სამუშაო პროგრამას. დავაბრუნოთ ცვლადი წინააღმდეგობის ღერძი წაღმა უკუღმა თავიდან ბოლომდე და დავაკვირდეთ როგორ იცვლება რიცხვები დისპლეის პირველ სტრიქონში ნოლიდან 1023-მდე, ხოლო მეორე სტრიქონში ნოლიდან 255-მდე.

Масштабирование

Параметры Надпись к блоку

Верхний предел входного сигнала Вход Константа 1023

Нижний предел входного сигнала Вход Константа 0

Верхний предел выходного сигнала Вход Константа 50

Нижний предел выходного сигнала Вход Константа 0

ფიგ. 7.5 მასშტაბირების ბლოკის პარამეტრიზაციის ფანჯარა

ახლა ჩავატაროთ ექსპერიმენტები და შევცვალოთ გამოსასვლელი სიგნალის ზედა და ქვედა ზღვრები 0-ზე და 50-ზე და დავაკვირდეთ ციგნალების ცვლილებებს (ფიგ. 7.5).

Масштабирование

Параметры Надпись к блоку

Верхний предел входного сигнала Вход Константа 1023

Нижний предел входного сигнала Вход Константа 0

Верхний предел выходного сигнала Вход Константа 50

Нижний предел выходного сигнала Вход Константа 0

ფიგ. 7.5 მასშტაბირების ბლოკის პარამეტრიზაციის ფანჯარა

და ბოლოს, შევცვალოთ კიდევ ერთჯერ გამოსასვლელი სიგნალის ზღვრები 4000 დან 1000 (ფიგ. 7.6) და დავაკვირდეთ სიგნალების ცვლილებებს.

Масштабирование

Параметры Надпись к блоку

Верхний предел входного сигнала Вход Константа 1023

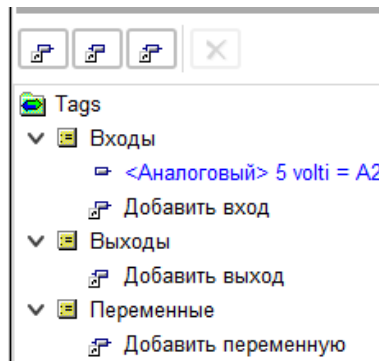
Нижний предел входного сигнала Вход Константа 0

Верхний предел выходного сигнала Вход Константа 4000

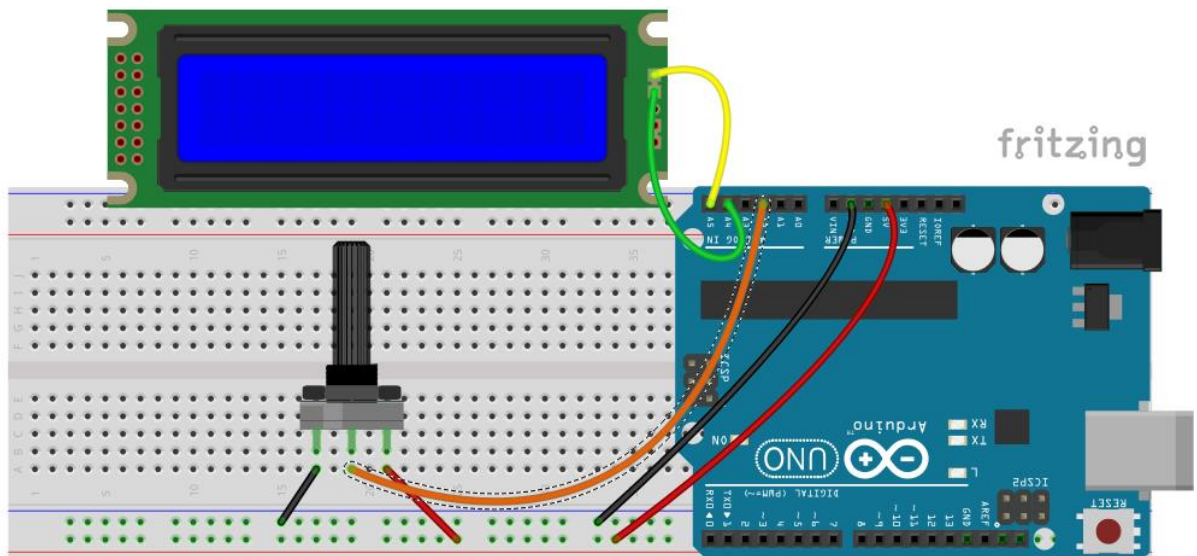
Нижний предел выходного сигнала Вход Константа 1000

ფიგ. 7.6 მასშტაბირების ბლოკის პარამეტრიზაციის ფანჯარა

პროექტის ტეგების ზონის ჩანაწერები ნაჩვენებია ფიგ. 7.7-ზე, ხოლო მასშტაბირების ბლოკის მართვის Fritzing სქემა ნაჩვენებია ფიგურა 7.7-ზე.



ფიგ. 7.7 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 7.8 მასშტაბირების ბლოკის მართვის Fritzing სქემა

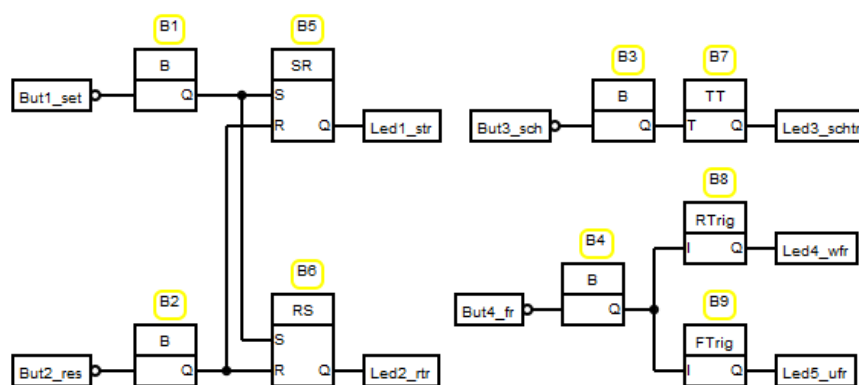
პროექტი_8

ტრიგერები



გავუშვათ პროგრამა FLProg, დაპროგრამების ენად ავიჩიოთ FBD ხოლო კონტროლერის ტიპად Arduino UNO. შევიდეთ ელემენტების ბიბლიოთეკის ზონაში და გავხსნათ ტრიგერების საქაღალდე-Триггеры. ზოგადად არსებობს ტრიგერების მრავალი სახეობა, D ტრიგერები, JK ტრიგერები, RS ტრიგერები და სხვ. ისინი განეკუთვნებიან ციფრული სქემების მიმდევრობით

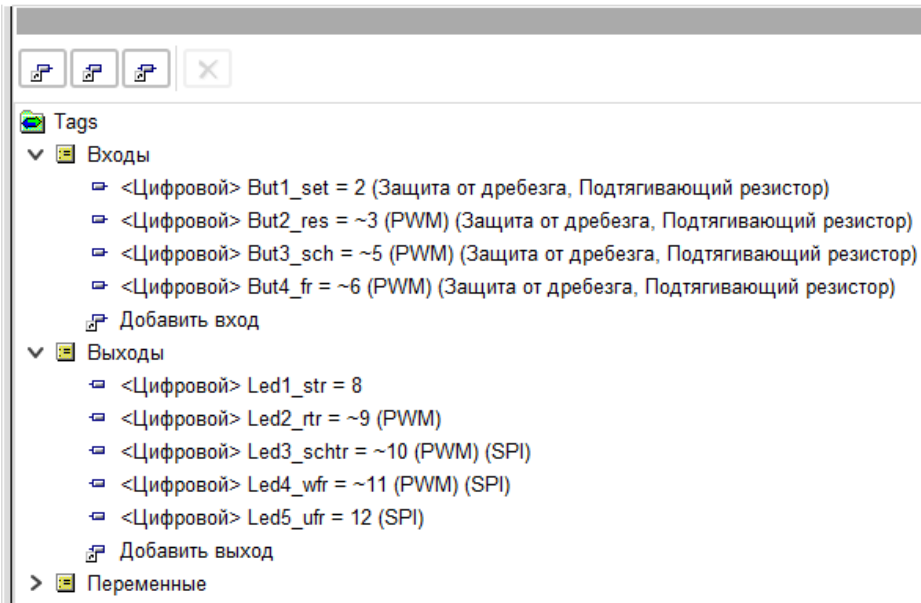
კლასს და თამაშობენ უმნიშვნელოვანეს როლს ელექტრონული მოწყობილობების დამუშავების საქმეში. განსხვავებით კომბინაციური სქემებისაგან მიმდევრობით სქემებს გააჩნიათ შიგა მეხსიერება და გამომავალი სიგნალები დამოკიდებულია არა მარტო შემავალი სიგნალებზე, არამედ შიგა მეხსიერების მდგომარეობებზეც. განსაკუთრებული ხაზი უნდა გაესვას იმას, რომ ტრიგერი არის ის ელემენტი, რომელიც შეიძლება იმყოფებოდეს ორ მდგომარეობაში: ერთიანის მდგომარეობაში, როდესაც მის გამოსასვლელზე ძაბვის მაღალი დონეა (+5ვ), ანუ იგი იმყოფება ლოგ. „1“-ის მდგომარეობაში და ნულოვან მდგომარეობაში, როცა მის გამოსასვლელზე ძაბვის დაბალი დონეა (0 ვოლტი), ანუ ლოგ. „0“-ის მდგომარეობაში.



ფიგ. 8.1 SR, RS, TT, RTrig, FTrig ტრიგერების მართვის პროგრამა

როგორც ტრიგერების საქალაქედან ჩანს პროგრამა FLProg შეიცავს ტრიგერების ხუთ სხვადასხვა ტიპს, აქედან ორ-ორი SR და RS ტრიგერები, აგრეთვე RTrig და Ftrig მცირედ განსხვავდებიან ერთმანეთისაგან. გადმოვიტანოთ ყველა ეს ტრიგერი პროგრამის სამუშაო ზონაში (ფიგ. 8.1).

ტეგების ზონაში შევქმნათ (ფიგ. 8.2): ა) ციფრული შესასვლელი დილაკებისათვის But1_set და But2_res SR და RS ტრიგერების მართვისათვის, ხოლო ტრიგერების მუშაობის ვიზუალური კონტროლის შუქდიოდებისათვის ციფრული გამოსასვლელი Led1_str და Led_rst; ბ) ციფრული შესასვლელი დილაკისათვის But3_sch მთვლელი TT ტრიგერის მართვისთვის და მისი მუშაობის ვიზუალური კონტროლის შუქდიოდისათვის-ციფრული გამოსასვლელი Led3_schtr; გ) ციფრული შესასვლელი But4_fr დილაკისათვის Rtrig და Ftrig ტრიგერების მართვისთვის და მათი მუშაობის ვიზუალური კონტროლის შუქდიოდებისათვის-ციფრული გამოსასვლელი Led4_wfr და Led5_ufr. როგორც ფიგ. 8.1-დან ჩანს ციფრული გამოსასვლელის გამოსასვლელი სიგნალები ყველანი ინვერტირებულია. ინვერტირების დანიშნულება განხილული იყო წინა პროექტებში.



ფიგ. 8.2. შესასვლელებისა და გამოსასვლელების შექმნა ტეგების ზონაში

გადმოვიტანოთ ეს ჩვენი შექმნილი შესასვლელ-გამოსასვლელები ტეგების ზონიდან პროგრამის სამუშაო ზონაში, ყველა შესასვლელის გამოსასვლელზე მივუერთოთ კონტაქტების ყანყალისაგან დაცვის Bounce ბლოკები B1, B2, B3, B4. შევავერთოთ შესასვლელები, Bounce ბლოკები, ტრიგერები SR, RS, TT, RTrig, FTrig და გამოსასვლელები ერთმანეთთან, როგორც ეს FLProg პროგრამაშია წარმოდგენილი (ფიგ. 8.1). გავუშვათ პროგრამის კომპილაცია და გადავტვირთოთ სამუშაო პროგრამა არდუინოში.

ახლა შევუდგეთ ჩვენს მიერ შექმნილი პროგრამის ტესტირებას. დასაწყისში ყველა შუქდიოდი უნდა იყოს ჩამქვრალი. თუ დავაჭერთ But1_set ღილაკს თითოს, მაშინ ორივე ტრიგერი უნდა დაყენდეს ერთიანის მდგომარეობაში და შესაბამისად, უნდა აინთოს ორივე შუქდიოდი Led1_str და Led_rst, ახლა ავუშვათ თითი ღილაკს და დავინახავთ რომ ორივე შუქდიოდი ანთებული დარჩება, ეს ნიშნავს რომ ტრიგერები არ შეიცვლიან მდგომარეობას და დარჩებიან ლოგიკური ერთიანის მდგომარეობაში. დავაჭიროთ ამავე ღილაკს რამდენჯერმე თითი და დავრწმუნდეთ, რომ ანთებული შუქდიოდების მდგომარეობა არ იცვლება რამდენი ხანიც არ უნდა გავიდეს და რამდენჯერაც არ უნდა დავაჭიროთ, ტრიგერები ინარჩუნებენ თავის მდგომარეობებს. ახლა დავაჭიროთ ღილაკს But2_res და დავინახავთ, რომ ორივე ანთებული შუქდიოდი ჩაქრება, ანუ ეს ნიშნავს, ფრომ ტრიგერები გადავარდებიან ერთიანის მდგომარეობაში და ინარჩუნებენ ამ მდგომარეობადს რამდენჯერაც არ უნდა დავაჭიროთ ამ ღილაკებს, შესაბამისად ამ შუქდიოდების მდგომარეობებიც დარჩება უცვლელი. ახლა დავაჭიროთ ორთავე But1_set და But2_res ღილაკს თითი, მაშინ დავინახავთ, რომ SR

ტრიგერთან შეერთებული Led1_str უქდიოდი აინთება ხოლო RS ტრიგერთან შეერთებული Led_rst ჩაქრება, თუ იგი ანთებული იყო ანდა ინარჩუნებს ჩამქვრალ მდგომარეობას, თუ იგი ჩამქვრალი იყო. მოვაცილით თითები ღილაკებს და დავრწმუნდეთ, რომ ტრიგერები შეინარჩუნებენ მდგომარეობას. დავაჭიროთ რამდენჯერმე ამ ორივე ღილაკს თითი და დავინახოთ რომ ტრიგერების მდგომარეობა არ შეიცვლება. აი მხოლოდ ამაში მდგომარეობს SR და RS ტრიგერებს შორის განსხვავება, SR ტრიგერში პრიორიტეტი ენიჭება ტრიგერის ერთზე დაყენების შესასვლელს S, ხოლო RS ტრიგერში ნულზე დაყენების შესასვლელს R, ორთავე S და R შესასვლელზე სიგნალების ერთდროული მოსვლის შემთხვევაში SR ტრიგერი დგება ერთიანის მდგომარეობაში და ინარჩუნებს ამ მდგომარეობას სიგნალების მოხსნის შემდეგაც, ხოლო RS ტრიგერი დგება ნოლოვან მდგომარეობაში და ინარჩუნებს ამ მდგომარეობას სიგნალების მოხსნის შემდეგაც.

ახლა გადავიდეთ TT ტრიგერის ტესტირებაზე. ამისათვის დავაჭიროთ მე-3 ღილაკის But3_sch გამოყენება და მე-3 უქდიოდი Led3_schtr-ის მუშაობის თვალთვალი. TT ტრიგერის დანიშნულებაა შეიცვალოს მდგომარეობა მის შესასვლელზე ყოველი იმპულსის დადგომის შემთხვევაში, წინა ფრონტის წარმოშობის მომენტში, ანუ თუ ტრიგერი ნოლოვან მდგომარეობაშია მაშინ იგი გადავარდება ერთიანის მდგომარეობაში, ხოლო თუ ერთიანის მდგომარეობაშია, მაშინ ჩამოვარდება ნულოვან მდგომარეობაში. იმპულსების წყაროდ გამოვიყენებთ ღილაკს But3_sch, რომელიც B3 (Bounce) ბლოკის გავლით მიეწოდება ტრიგერის ბლოკს B7, რომლის გამოსავალიც თავის მხრივ მიერთებულია Led3_schtr უქდიოდს წინააღმდეგობის გავლით.

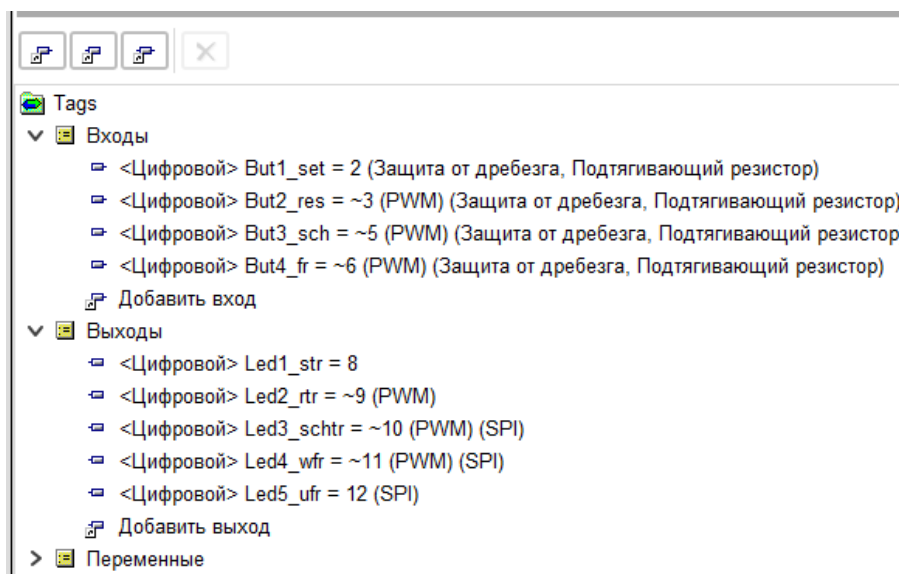
ახლა დავაჭიროთ თითი But3_sch ღილაკს, რომელიც გამოიწვევს B7 ტრიგერის ერთიანის მდგომარეობაში დაყენებას, რასაც მოჰყვება Led3_schtr უქდიოდის ანთება, აუშვათ თითი ღილაკს და დავინახავთ, რომ ტრიგერმა შეინარჩუნა მდგომარეობა, რაც იქიდან ჩანს, რომ Led3_schtr უქდიოდი განაგრძობს ნათებას, დავაჭიროთ ისევ თითი But3_sch ღილაკს და დავინახავთ, რომ B7 ტრიგერი ჩამოვარდება ნულში, რაც იქიდან გამოჩნდება, რომ Led3_schtr უქდიოდი ჩაქრება, ავუშვათ თითი ღილაკს და დავრწმუნდეთ, რომ ტრიგერმა შეინარჩუნა ნულოვანი მდგომარეობა და Led3_schtr უქდიოდი ჩამქვრალია, დავაჭიროთ თითი ისევ But3_sch ღილაკს და დავინახავთ, რომ Led3_schtr უქდიოდი ისევ აინთება და ა.შ. უსასრულოდ.

გადავიდეთ ახლა Rtrig და Ftrig ტრიგერების შემოწმებაზე. წინასწარ შევნიშნავთ, რომ პროგრამა FLProg ისევე, როგორც სხვა პროგრამები, რომლებიც მუშაობენ ტექნიკური სისტემებისა და ტექნოლოგიური პროცესების მართვის ავტომატიზირებულ სისტემებში,

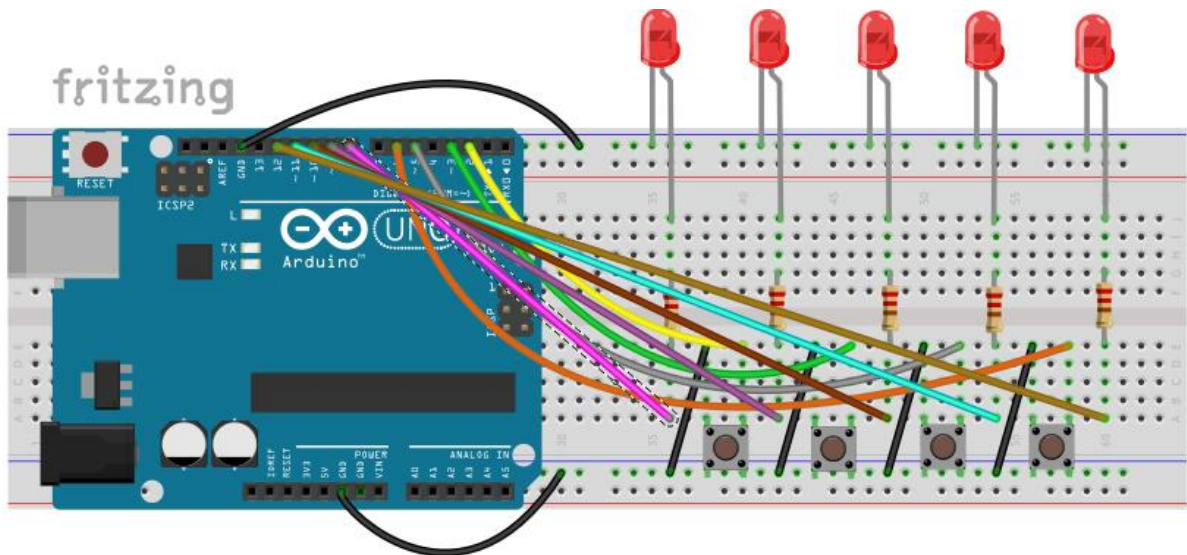
მუშაობენ ციკლურად, ანუ იწყებენ მუშაობას და ასრულებენ მიმდევრობით პროგრამულ ბლოკებს მარცხნიდან მარჯვნივ და ზემოდან ქვემოთ, როდესაც შეასრულებენ სულ ქვედა მარჯვენა ბლოკს, მუშაობას იწყებენ ისევ თავიდან ზედა მარცხენა ბლოკიდან. პროგრამის ამ ერთ გავლას ეწოდება მისი ციკლი, რომლის ხანგრძლივობა სხვადასხვაა სხვადასხვა სირთულის პროგრამებიდან გამომდინარე.

ჩვენი ზემოთმოხსენებული ტრიგერები Rtrig და Ftrig, ბლოკები ნომრით R8 და R9 დაკავშირებულნი არიან ამ პროგრამულ ციკლებთან. მათ შესასვლელელებზე სასიგნალო იმპულსის მოსვლის შემთხვევაში, ამ იმპულსის წინა ფრონტზე, ე.ი. როდესაც შესასვლელი სიგნალი გადადის დაბალი დონიდან მაღალ დონეზე, მაშინ ამუშავდება Rtrig და თავის გამოსასვლელზე გამოიმუშავებს ერთი ციკლის ტოლ იმპულსს, ხოლო შესასვლელი იმპულსის უკანა ფრონტზე, ანუ როცა იმპულსი გადადის მაღალი დონიდან დაბალ დონეზე, ამუშავდება Ftrig და თავის გამოსასვლელზე გამოიმუშავებს ასევე ერთი ციკლის ტოლ იმპულსს. ამ ტრიგერების შემოწმებისთვის გამოვიყენებთ But4_fr ღილაკს, რომელიც B4 (Bounce) ბლოკის გავლით მიერთებულია R8 და R9 ტრიგერების ბლოკებზე. ტრიგერების გამოსასვლელები კი არდუინოს Led4_wfr და Led5_ufr კონტაქტებისა და შემზღუდავი წინააღმდეგობების გავლით დაკავშირებულია შესაბამის შუქდიოდებთან. როდესაც დავაჭერთ თითო But4_fr ღილაკს დავინახავთ, რომ აინთება Led4_wfr შუქდიოდი მცირე ხნით, ერთი ციკლის განმავლობაში, ხოლო როდესაც ავუშვებთ But4_fr ღილაკს თითო, მაშინ აინთება Led5_ufr შუქდიოდი, ასევე მცირე ხნით, ერთი ციკლის განმავლობაში.

პროექტის ტეგების ზონა ასახულია ფიგ. 8.3-ზე, ხოლო პროექტის SR, RS, TT, RTrig, FTrig ტრიგერების მართვის Fritzing სქემა მოცემულია ფიგ. 8.4-ზე.



ფიგ. 8.3. პროექტის ტეგების ზონის ფანჯარა



ფიგ. 8.4. SR, RS, TT, RTrig, FTrig ტრიგერების მართვის Fritzing სქემა

პროექტი_9

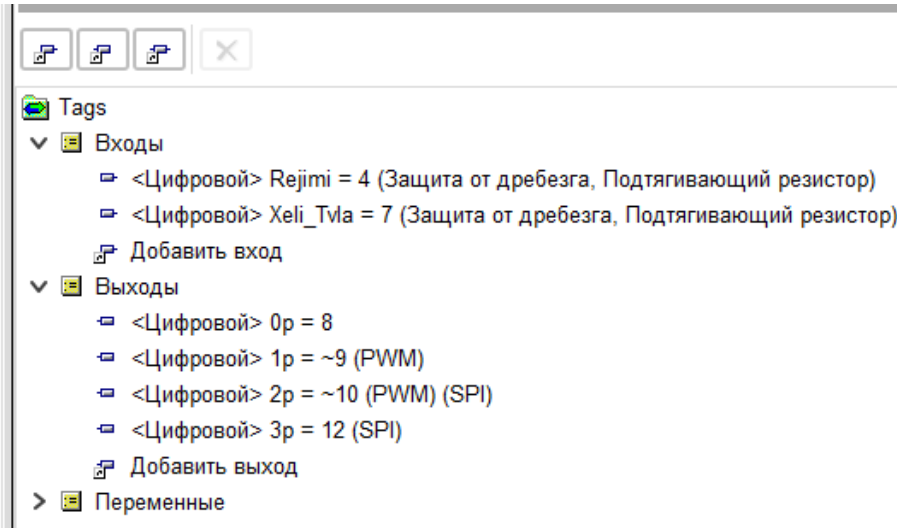
ორობითი მთვლელი



ორობითი მთვლელი ფართოდ გამოიყენება ციფრულ ტექნოლოგიებში. შეიძლება ითქვას, რომ პრაქტიკულად ვერ ნახავთ მეტნაკლებად სერიოზულ ელექტრონულ დამუშავებას სადაც ორობითი მთვლელი არ იყოს გამოყენებული. ტრიგერული ბლოკები ის ელემენტებია, რომელთა დახმარებითაც შეიძლება აგებულ იქნას ორობითი მთვლელი, ყველაზე უფრო მარტივად ეს ამოცანა წყდება TT ტრიგერების გამოყენებით. ამჯერად ჩვენ შემოვიფარგლებით მარტივი ოთხთანრიგიანი მთვლელის აგებით, თუმც ასევე მარტივად შეიძლება აგებულ იქნას რვა თანრიგიანი, თორმეტთანრიგიანი და ა.შ. ნებისმიერი თანრიგიანობის ორობითი მთვლელი.

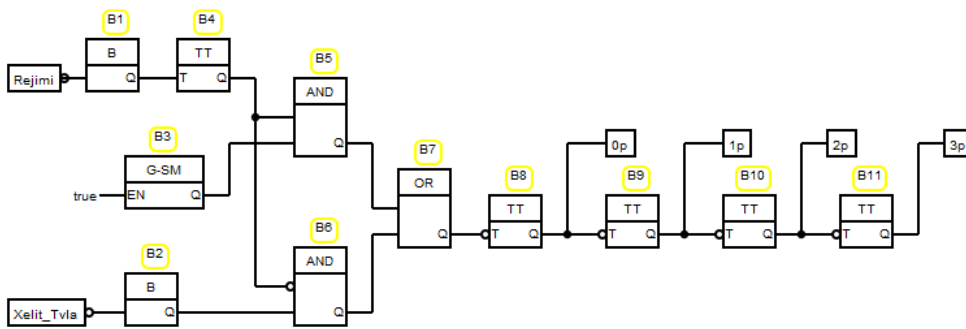
დავუშვათ, ჩვენ მთვლელი გვინდა ვამუშაოთ ორ რეჟიმში: ხელით თვლის რეჟიმში და ავტომატური თვლის რეჟიმში. ამისათვის დაგვჭირდება ორი ღილაკი, პირველი ღილაკი რომელიც შეცვლის რეჟიმს და მეორე ღილაკი, რომელიც განკითვნილი იქნება თითის

დაჭერით თვლის იმპულსების გამომუშავებლად. შევქმნათ ორი ციფრული შესასვლელი-Вход მაგალითად არდუინოს მე-4 და მე-7 კონტაქტების გამოყენებით (ფიგ. 9.1), რომლებზედაც უნდა მივუერთოთ ორო გარე მექანიკური ღილაკი: რეჟიმის ცვლილების ღილაკი Rejimi და ხელით თვლის ღილაკი Xelit_Tvla. აქვე შევქმნათ ოთხი ციფრული გამოსასვლელი-Выходы, თანრიგებისათვის 0p, 1p, 2p, 3p.



ფიგ. 9.1 შემსავალი და გამომავალი კონტაქტების შექმნა ტეგების ზონაში

პროგრამა FLProg-ის ბლოკების ბიბლიოთეკიდან პროგრამის სამუშაო ზონაში გადმოვითანოთ ორობითი მთვლელის ასაგებად შემდეგი სახის ბლოკები (ფიგ. 9.2). ჩვენ დაგვჭირდება ოთხი TT ტრიგერი თვითონ ორობითი მთვლელის აგებისათვის B8, B9, B10, B11 და დანარჩენი ბლოკები თვითონ ამ მთვლელის მართვისათვის, ესენია ორი Bounce ბლოკი B1, B2, ერთი გენერატორის ბლოკი B3, ერთი TT ტრიგერი B4, ორი ლოგიკური გამრავლების (AND) ბლოკი B5, B6 და ერთი ლოგიკური შეკრების (OR) ბლოკი B7. შევაერთოთ ეს ბლოკები ერთმანეთთანჯ მითითებულ ფიგურაზე წარმოდგენილი სქემის მიხედვით.



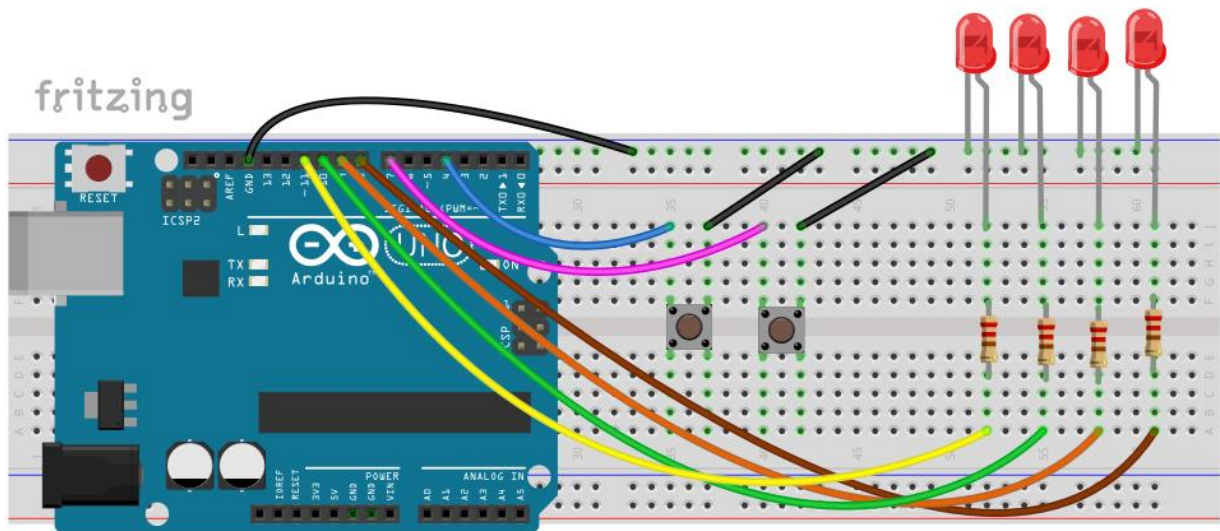
ფიგ. 9.2 ორობითი მთვლელის პროგრამა

განვიხილოთ ახლა თუ რაში მდგომარეობს ამ პროექტისათვის ორობითი მთვლელის მართვა. განსხვავება იმაშია თუ როგორ მიეწოდება სათვლელი იმპულსები მთვლელს. ამისათვის არსებობს რეჟიმის ღილაკი Rejimi. ამ ღილაკით შეიძლება თვლის იმპულსების გამომუშავების წყაროს მართვა. საწყის მდგომარეობაში, როდესაც ამ ღილაკზე არა გვაქვს დაჭერილი თითი, მაშინ B4 ტრიგერი იმყოფება ნულის მდგომარეობაში, ეს ნიშნავს იმას, რომ ლოგ. „0“ მისი გამოსასვლელიდან კეტავს ლოგიკური გამრავლების ბლოკს B5 და აღებს ასევე ლოგიკური გამრავლების ბლოკს B6, რადგან ამ ტრიგერის გამოსასვლელი ინვერტირებას განიცდის B6-ის შესასვლელზე. ამით გზა ეხსნება მეორე ღილაკის მოქმედებას Xelit_Tv1a. ამ შემთხვევაში, თითის ყოველი დაჭერა ამ ღილაკზე გამოიწვევს ლოგ. „1“-ის ფორმირებას მის გამოსასვლელზე, რომელიც B2, B6 და ლოგიკური შეკრების ბლოკის B7 გავლით მიეწოდება ორობითი მთვლელის ნულოვანი თანრიგის შესასვლელის ტრიგერს B8. ახლა, თუ დავაჭერთ ღილაკს რეჟიმი, მაშინ ლოგ. „1“ მისი გამოსასვლელიდან B1 ბლოკის გავლით დააყენებს ერთიანის მდგომარეობაში B4 ტრიგერს, რაც გამოიწვევს B5 ბლოკის გაღებას და B6 ბლოკის ჩაკეტვას. ახლა უკვე გენერატორი შედის საქმეში და მის მიერ გამომუშავებული იმპულსები ავტომატურად, ღილაკზე Xelit_Tv1a დაჭერის გარეშე, B5 და B7 ბლოკების გავლით მიეწოდება ორობითი მთვლელის ნულოვანი თანრიგის შესასვლელის ტრიგერს B8.

ამის შემდეგ უნდა განვიხილოთ, თუ როგორ მუშაობს თვითონ ოთხთანრიგიანი ორობითი მთვლელი. წინასწარ შევნიშნავთ, რომ ჩვეულებრივად თანრიგები განლაგებულნი არიან მარჯვნიდან მარცხნივ თანრიგების ზრდის მიხედვით. ანუ ნულოვანი უმცროსი თანრიგი მარჯვნივაა, შემდეგ მოდის პირველი თანრიგი მის მარცხნივ და ასევე სხვა თანრიგებიც. სქემაზე კი თანრიგების შესაბამისი ტრიგერები განლაგებულია მარცხნიდან მარჯვნივ ზრდის მიხედვით. ამიტომ, რომ ჩვენ პირველი ტრიგერის გამოსასვლელს დავაწერეთ, რომ ის არის ნულოვანი თანრიგი p0, მეორე ტრიგერის გამოსასვლელს p1 და ა.შ.შ. სამაგიეროდ შუქდიოდებზე ეს შებრუნებულობა გამოსწორებულია და ნოლოვანი თანრიგი შეერთებულია მარჯვენა პირველ შუქდიოდზე, პირველი თანრიგი შეერთებულია იმის წინა მარცხენა შუქდიოდზე და ა.შ.შ ანუ შუქდიოდები განლაგებულია თანრიგების მიხედვით მარჯვნიდან მარცხნივ ზრდის მიხედვით, ისევე როგორც თანრიგები განლაგებულია ათობითი ათვლის სისტემაში.

საწყის მომენტში ყველა ტრიგერი ნულოვან მდგომარეობაში იმყოფება, ანუ მთვლელის თანრიგებზე დაფიქსირებულია 0000 და ამიტომ ყველა შუქდიოდი ჩამქვრალია. ნულოვანი თანრიგის ტრიგერზე B8 პირველი იმპულსის მოსვლის შემდეგ, მის უკანა ფრონტზე (რადგან ტრიგერის შესასვლელზე რგოლია, ანუ შესასვლელი სიგნალი ინვერტირებულია), რადგან

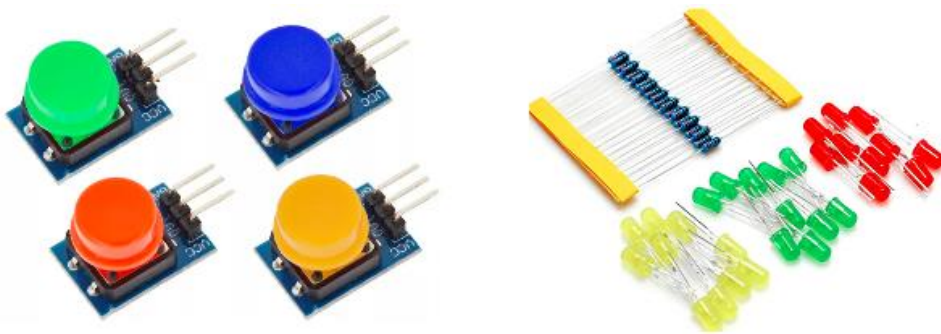
ყოველი იმპულსის მოსვლის შემდეგ TT ტრიგერი მდგომარეობას იცვლის, ამიტომ ეს ტრიგერი დაყენდება ერთის მდგომარეობაში, რაც გამოიწვევს P0 გამოსასვლელზე ძაბვის მაღალი დონის გაჩენას, რაც თავის მხრივ გამოიწვევს მასზედ მიერთებული შუქდიოდის ანთებას ე.ი. ოთხ შუქდიოდზე დაფიქსირდა 0001. ანუ მთვლელმა დაითვალა ერთი იმპულსი და აანთო უმცროსი თანრიგის შუქდიოდი. მე-2 იმპულსის მოსვლის შემდეგ, პირველი ტრიგერი მდგომარეობას იცვლის და დადგება ნოლიანის მდგომარეობაში, მაგრამ მის გამოსასვლელზე გაჩენილი იმპულსის უკანა ფრონტი დააყენებს მეორე ტრიგერს B9 ერთიანის მდგომარეობაში, მთვლელში დაფიქსირდება 0010, რაც ორობითი ათვლის სისტემაში ნიშნავს ორს, ამასთანავე p1 გამოსასვლელის გავლით აინთება მეორე თანრიგის შუქდიოდი, ანუ მთვლელმა დაითვალა ორი იმპულსი. მესამე იმპულსის მოსვლისას მდგომარეობას შეიცვლის მხოლოდ პირველი თანრიგის ტრიგერი B8, ხოლო დანარჩენი ტრიგერები მდგომარეობებს არ იცვლიან, რადგან იმპულსის უკანა ფრონტი არცერთზე არ მიეწოდება, ესეიგი აინთება მხოლოდ ორი ბოლო შუქდიოდი 0011, ეს კი ორობით ათვლის სისტემაში ტოლია სამის, ანუ მთვლელმა დაითვალა სამი იმპულსი. მეოთხე იმპულსის მოსვლის შემდეგ მდგომარეობას შეიცვლის B8 ტრიგერი და დადგება ნულზე, ეს კი გამოიწვევს B9 ტრიგერის მდგომარეობის შეცვლასაც რადგან მის შესასვლელზეც ინვერტირების შემდეგ წარმოიშობა იმპულსის წინა ფრონტი B8 ტრიგერის ნულზე დაყენების გამო და იგი დადგება ნულოვან მდგომარეობაში, ასევე შეიცვლება B9 ტრიგერის მდგომარეობაც, რაც თავის მხრივ გამოიწვევს B10 ტრიგერის მდგომარეობის შეცვლას B9 ტრიგერის ნულზე დაყენების გამო, რადგან მის შესასვლელზეც წარმოიშობა იმპულსის წინა ფრონტი და აინთება მხოლოდ მე-3 შუქდიოდი, ანუ მთვლელში დაფიქსირდება 0100, ეს კი ორობით ათვლის სისტემაში არის ოთხი, ე.ი ორობითმა მთვლელმა დაითვალა ოთხი იმპულსი და ასე შემდეგ, შემდეგი იმპულსების მოსვლის შედეგად მთვლელში დაფიქსირდება მდგომარეობები: 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111 ესენი კო ორობითი ათვლის სისტემაში ნიშნავს: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15-ს. მე-16 იმპულსის მოსვლის შედეგად მთვლელი გადაივსება და ყველა თანრიგი დადგება ნულოვან მდგომარეობაში, ანუ ორობით მთვლელში დაფიქსირდება 0000. შემდგომი იმპულსების მოსვლის შემთხვევაში ზემოთაღწერილი პროცესი განმეორდება. ორობითი მთვლელის Fritzing სქემა ნაჩვენებია ფიგურა 9.3-ზე.



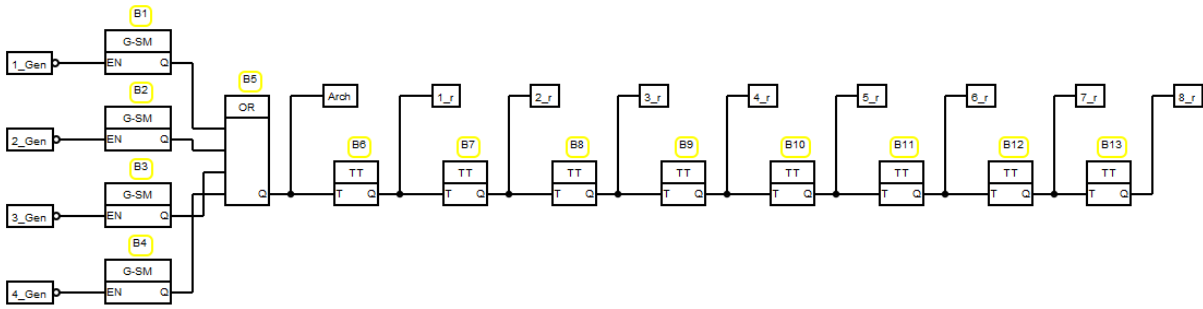
ფიგ. 9.3 ორობითი ოთხთანრიგანი მთვლელის Fritzing სქემა

პროექტი_10

სიხშირების გამყოფი

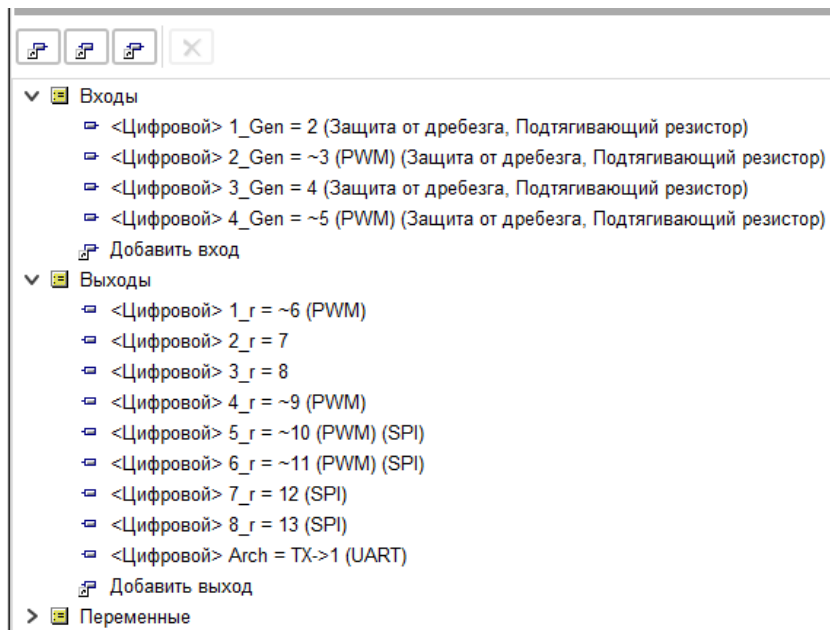


ამ პროექტში განხილული იქნება რვათანრიგანი სიხშირული გამყოფი, რომელიც მუშაობს ოთხი სხვადასხვა სიხშირის გაყოფაზე. სიხშირების გამყოფის პროგრამა მოცემულია ფიგ. 10.1-ზე. პროგრამაში გამოიყენება ოთხი სხვადასხვა სიხშირე, რომლებსაც გამოიმუშავენ ოთხი სხვადასვა გენერატორი (სიმეტრიული მულტივიბრატორები) B1, B2, B3, B4. ეს გენერატორები აწყობილია სხვადასხვა სიხშირების გამომუშავებაზე (იმპულსების ხანგრძლივობა 100, 200, 300, 400 მლწ). მათი გაშვება ხდება ოთხი სხვადასხვა გარე მექანიკური ღილაკით, რომლებიც შეერთებულია არდუინოს შესასვლელ კონტაქტებზე 2,3,4,5. ღილაკზე თითის დაჭერა იწვევს შესაბამისი გენერატორის ამუშავებას. გენერატორების იმპულსების ლოგიკური შეკრება ხდება ლოგიკური შეკრების B5 ბლოკის გამოყენებით. იმპულსების დაყოფა ხდება TT ტრიგერების გამოყენებით (B6...B13), რომელთანგან თვითოეული ჰყოფს წინა ტრიგერიდან მიღებული იმპულსების სიხშირეს ორჯერ.



ფიგ. 10.1 სიხშირების დამყოფის პროგრამა

პროგრამის შესასვლელ/გამოსასვლელი ტეგები ნაჩვენებია ფიგ. 10.2-ზე, სადაც ტეგები 1_Gen...4_G ის შესასვლელი კონტაქტებია, რომელზედაც უნდა შეერთდნენ გენერატორის მუშაობის ნების დართვის დილაკები. ტეგი Arch-ის გამოსასვლელია, რომელზედაც გამოდიან არჩეული გენერატორის მიერ გენერირებული სიგნალები, ხოლო ტეგები 1_r...8_r გამოსასვლელია, რომლებზედაც გამოდიან სიხშირების ორზე დამყოფი ტრიგერების თანრიგები.



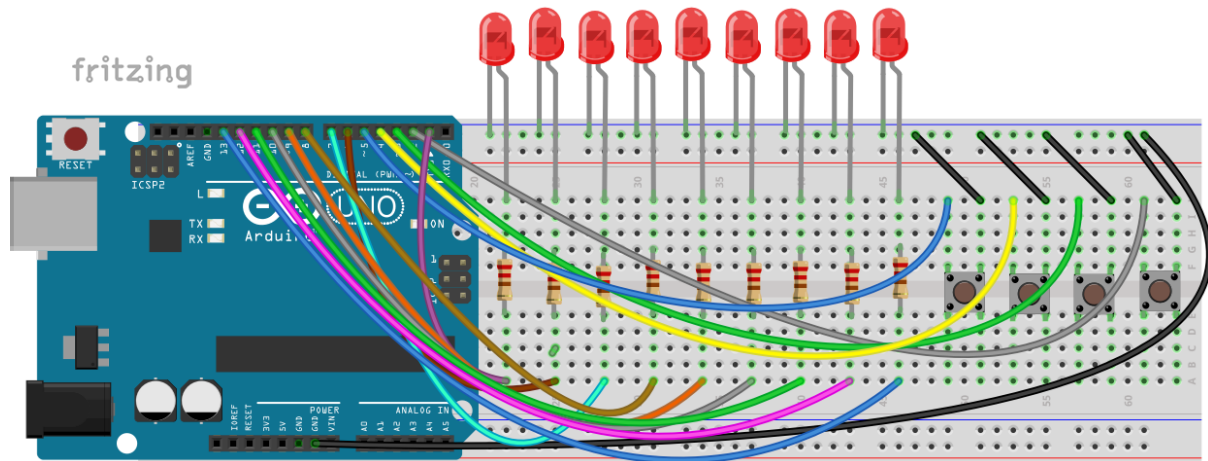
ფიგ. 10.2 სიხშირის დამყოფის შესასვლელ/გამოსასვლელის ტეგები

ავაწყობთ პროგრამა FLProg-ში, გადავწეროთ არდუინოში და გავუშვათ. საწყის მომენტში არცერთი შუქდიოდი არ უნდა ანათებდეს, რადგან ყველა გენერატორი ჩაკეტილია მათ ნებისდართვის EN შესასვლელზე დილაკებიდან მოდებული ლოგიკური ნოლებით.

დავაჭიროთ პირველ დილაკს თითი, რომელიც მიაწვდის ლიგიკურ ერთიანს პირველი გენერატორის B1-ის EN შესასვლელს, აამუშავებს მას, რის შედეგად მის მიერ გამოიმუშავებული

იმპულსების თანმიმდევრობა 100 მლწ იმპულსითა და 100 მლწ პაუზით B5 ბლოკის გავლით გამოჩნდება Arch გამოსასვლელზე და მასზედ მიერთებული შუქდიოდი დაიწყებს ციმციმს შესაბამისი სიხშირით. იგივე იმპულსები მიეწოდება სიხშირის ოროჯერ დამყოფ TT ტრიგერების მიმდევრობას, პირველი ამ ტრიგერისა სიხშირის გაყოფის შემდეგ გამოიმუშავებს იმპულსების თანმიმდევრობას 200 მლწ იმპულსითა და 200 მლწ პაუზით. ეს კარგად გამოჩნდება ამ ტრიგერის 1_r-ზე გამოსასვლელზე მიერთებული შუქდიოდის ციმციმით, რომლის სიხშირეც ორჯერ ნელი იქნება Arch გამოსასვლელზე მიერთებულ შუქდიოდის ციმციმზე, მეორე TT ტრიგერი გამოიმუშავებს იმპულსების თანმიმდევრობას, რომლის სიხშირეც თავის მხრივ ორჯერ ნაკლები იქნება წინა ტრიგერის იმპულსურ თანმიმდევრობასთან შედარებით, ანუ 400 მლწ იმპულსითა და 400 მლწ პაუზით, მესამე ტრიგერი გამოიმუშავებს იმპულსების თანმიმდევრობას 800 მლწ იმპულსით და 800 მლწ პაუზით და ა.შ. ყოველი შემდეგი ტრიგერი ორჯერ გაყოფს სიხშირეს წინასთან შედარებით.

ახლა, დავაწვთ მეორე ღილაკს, თითი და გავუშვათ მეორე გენერატორი B2, დავინახავთ, რომ შუქდიოდების ციმციმის სიხშირე შემცირდება, რადგან ეს გენერატორი თვითონ გამოიმუშავებს იმპულსებს უფრო ნაკლები სიხშირით, ასევე შემცირდება სიხშირეები მე-3 და მე-4 ღილაკებზე თითის დაჭერის დროს, რადგან ამ შემთხვევაში გაიშვება უფრო ნაკლები სიხშირეებით მომუშავე B3 და B4 გენერატორები შესაბამისად. სიხშირეების გამყოფის Fritzing სქემა ნაჩვენებია ფიგ. 10.3-ზე.



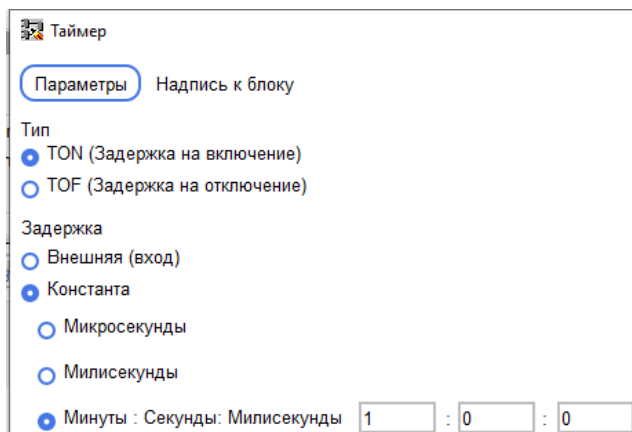
ფიგ. 10.3 სიხშირეთა დამყოფის Fritzing სქემა

პროექტი_11

ტაიმერები

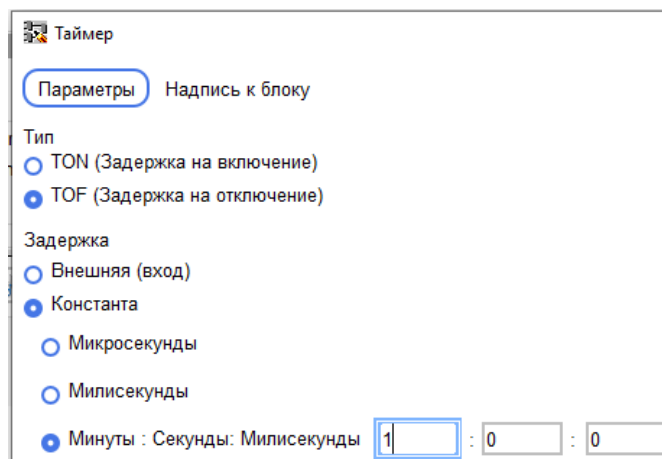


ტაიმერები გამოიყენება რაიმე მოქმედებებზე ჩვენს მიერ განსაზღვრული დროითი დაყოვნებით რეაგირებისთვის. არდუინოში გამოიყენება ორი სახის ტაიმერი: ა) ტაიმერი რომელიც მუშაობას ჩართვაზე დაყოვნებით TON; ბ) ტაიმერი, რომელიც მუშაობს გამორთვაზე დაყოვნებით TOF. ამ პროექტში განვიხილავთ ორივე ტაიმერის მუშაობას ცალცალკე. ამისათვის ჯერ უნდა გავუშვათ პროგრამა FLProg, სამუშაო ენად ავირჩიოთ FBD, ხოლო კონტროლერის ტიპად Arduino UNO. ბლოკების ბიბლიოთეკის ზონის ტაიმერების საქაღალდედან-Таймеры პროგრამის სამუშაო ზონაში გადმოვიტანოთ ბლოკი ტაიმერი-[Timer]. ეს ბლოკი საწყისში სიჩუმით წარმოადგენს დაყოვნების ტაიმერს ჩართვაზე TON. დავაწკაპუნოთ ამ ბლოკზე და გადავიდეთ ტაიმერის პარამეტრიზაციის ფანჯარაში -Таймер (ფიგ. 11.1). ყველა აწყობები დავტოვებთ უცვლელად, მხოლოდ წუთები: წამები : მილიწამების ველებში-Минуты: Секунды: Миллисекунды მოვნიშნოთ წუთები-Минуты: და შესაბამის ველში ჩავწეროთ ერთიანი, რაც იმას ნიშნავს, რომ ჩვენ ტაიმერის დაყოვნების სიდიდით ვირჩევთ 1 წუთს, თუმცა თავისუფლად შეგვეძლო მიგვეთითებინა სხვა ნებისმიერი სიდიდეც, მათ შორიც წამებიც და მიკროწამებიც. დავაჭიროთ მზადყოფნის ღილაკს-Готово და დავხუროთ TON ტაიმერის პარამეტრიზაციის ფანჯარა.



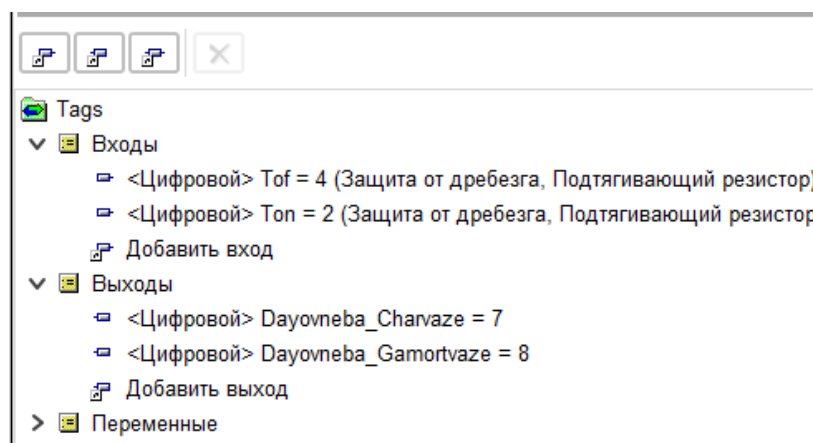
ფიგ. 11.1 TON ტაიმერის პარამეტრიზაცია

ახლა გავაკეთოთ ამ ტაიმერის კოპირება და ჩავსვათ ეს კოპირებული ბლოკი იქვე წინა ტაიმერის მახლობლობაში. შევიდეთ ამ ახალი ტაიმერის ბლოკის პარამეტრიზაციის ფანჯარაში (ფიგ. 11.2), სადაც TON შევცვალოთ TOF-ით (ანუ ვირჩევთ დაყოვნების რეჟიმს გამორთვაზე), წუთები: წამები : მილიწამების ველებში-Минуты: Секунды: Миллисекунды მოვნიშნოთ წუთები და წუთების ველში ჩავწეროთ ისევ ერთიანი. ეს იმას ნიშნავს, რომ გამორთვაზეც ჩვენ დაყოვნების სიდიდით ვირჩევთ ერთი წუთს, თუმც ტავისუფლად შეგვეძლო მიგვეთითებინა სხვა ნებისმიერი სიდიდე. დავაჭიროთ მზადყოფნის ლილავს-Готово და დავხუროთ TOF ტაიმერის პარამეტრიზაციის ფანჯარაც.



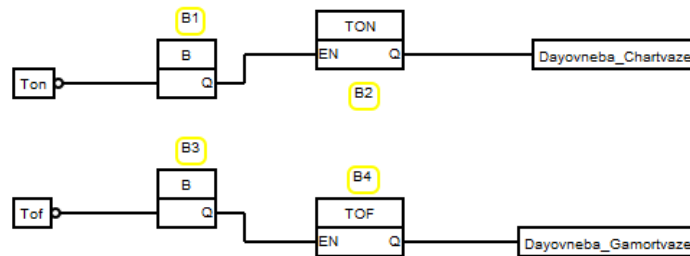
ფიგ. 11.2 TOF ტაიმერის პარამეტრიზაცია

ახლა გადავიდეთ არდუინოს ტეგების ზონაში და შევასრულოთ არდუინოს შესასვლელებისა და გამოსასვლელების შექმნის პროცედურები, რომელსაც ვახორციელებთ წინა პროექტებში უკვე შესწავლილი მეთოდების გამოყენებით (ფიგ. 11.3).



ფიგ. 11.3 არდუინოს შესასვლელებისა და გამოსასვლელების შექმნა ტეგების ზონაში

ჩვენ დაგვჭირდება ორი შესასვლელი Ton და Tof, თითო ცალკეული ტაიმერისათვის, რომლებზედაც უნდა შევუერთოთ შესაბამისი ფიზიკური ღილაკები და ორი გამოსასვლელი: დაყოვნება ჩართვაზე და დაყოვნება გამორთვაზე-Dayovneba_CharTvaze და Dayovneba_Gamortvaze, რომლებსაც თავის მხრივ უნდა შევუერთოთ შუქდიოდები შემზღუდავი წინააღმდეგობების გავლით სიდიდით 220 ომი. გადმოვიტანოთ ეს შესასვლელ/გამოსასვლელები პროგრამის სამუშაო ზონაში, აგრეთვე ბლოკების ბიბლიოთეკიდან ორი ღილაკების კონტაქტების ყანყალისათვის დაცვის ბლოკი B1 და B3, ორი ტაიმერის ბლოკი B2 და B4, რომელთაც დავაპროგრამებთ ზემოთ აღნიშნული მეთოდების გამოყენებით და შევაერთოთ ეს ბლოკები ერთმანეთთან ფიგურა 11.3-ზე მაჩვენები სქემის მიხედვით.

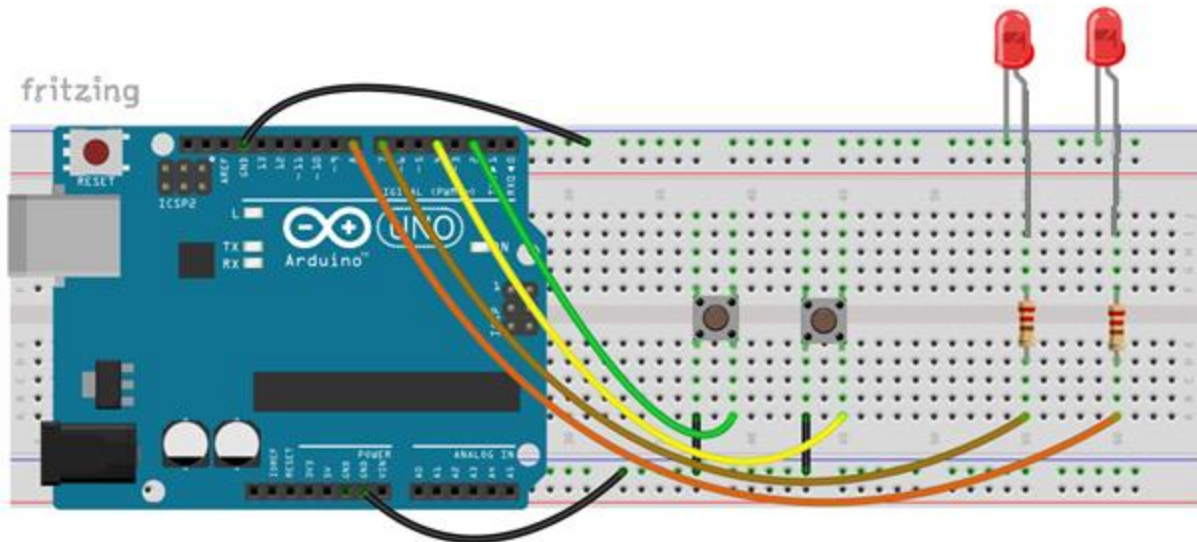


ფიგ. 11. 3 ტაიმერების მართვის პროგრამა

ჩავტვირთოთ ეს პროგრამა არდუინოში და შევამოწმოთ ტაიმერების მუშაობები. დავაჭიროთ ღილაკს Ton თითი დიდი ხნის განმავლობაში. ერთი წუთის შემდეგ უნდა აინთოს Dayovneba_Chartvaze შუქდიოდი. ავუშვათ ღილაკს თითი და დავრწმუნდეთ, რომ შუქდიოდი უნდა ჩაქრეს ღილაკზე თითის აშვებისთანავე. ექსპერიმენტის განმეორებისათვის საჭიერია ავუშვათ ღილაკს თითი და შემდეგ ისევ დავაჭიროთ.

ახლა დავაჭიროთ თითი მე-2 ღილაკს Tof თითი დიდი ხნის განმავლობაში, დავინახავთ, რომ მეორე შუქდიოდი Dayovneba_Gamortvaze აინთება თითის დაჭერისთანავე და ანათებს ერთი წუთის განმავლობაში, შემდეგ კი შუქდიოდი ჩაქრება რამდენ ხანსაც არ უნდა გვეკონდეს დაჭერილი ამ ღილაკზე. ექსპერიმენტის განმეორებისათვის საჭიერია ავუშვათ ღილაკს თითი და შემდეგ ისევ დავაჭიროთ.

ტაიმერების მართვის Fritzing სქემა ნაჩვენებია ფიგურა 11.4 სქემაზე.



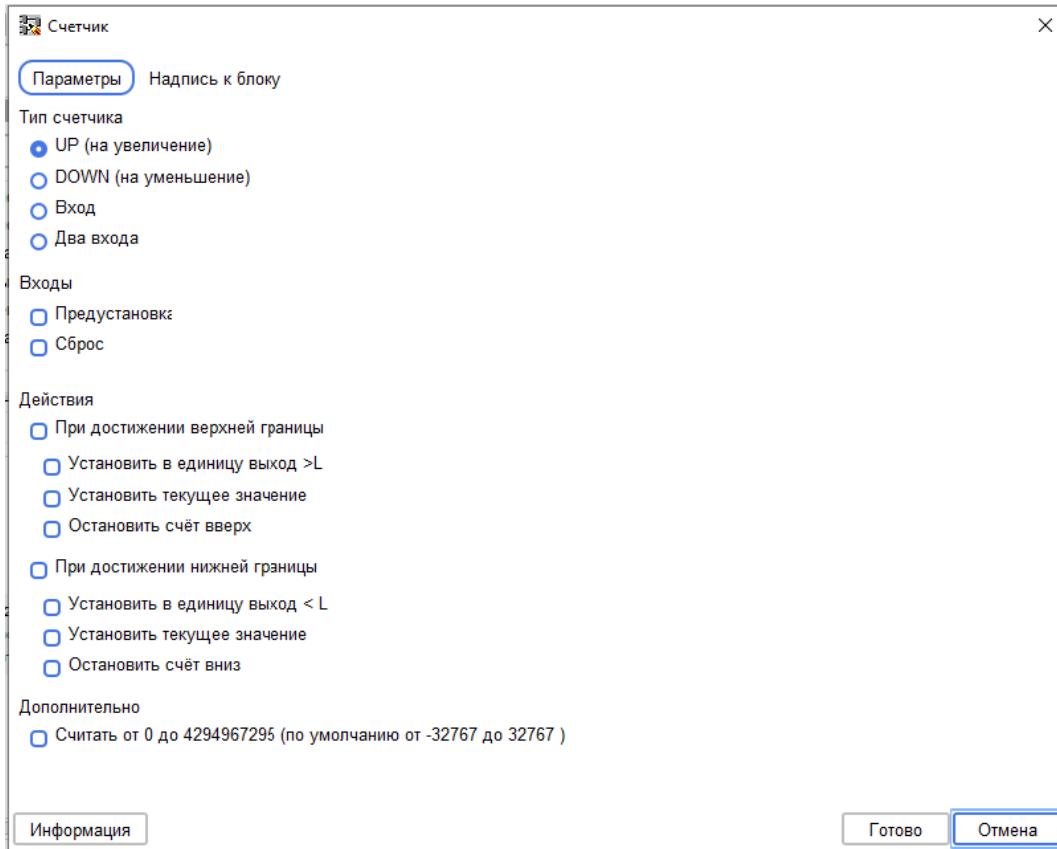
ფიგ. 11.4 ტაიმერების მართვის Fritzing სქემა

პროექტი_12

მთვლელები პარამეტრიზაციის გარეშე

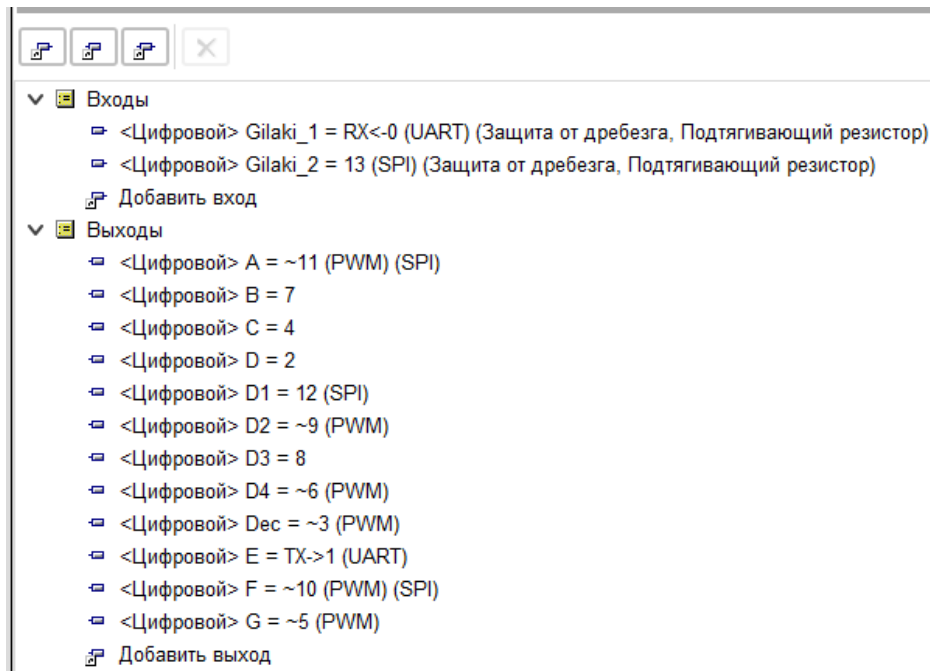


განვიხილოთ მთვლელები რომელსაც არ გავუკეთებთ პარამეტრიზაციას და მივიღებთ ისე, როგორც ეს სიჩუმით არის შემოთავაზებული პროგრამის მიერ. ამ შემთხვევაში მთვლელები მუშაობენ პირდაპირი თვლის რეჟიმში (მატების რეჟიმი) და არ აქვთ უკუ თვლის რეჟიმში გადასვლის, რაიმე ინფორმაციის წინასწარი ჩაწერის ანდა ნულზე დაყენების ფუნქციები განსაზღვრული. ამ შემთხვევაში მთვლელის პარამეტრიზაციის ფანჯარა გამოიყურება ფიგ. 12.1 ნაჩვენები გამოსახულების მსგავსად, რომელსაც დავტოვებთ უცვლელად კორექტირების გარეშე.



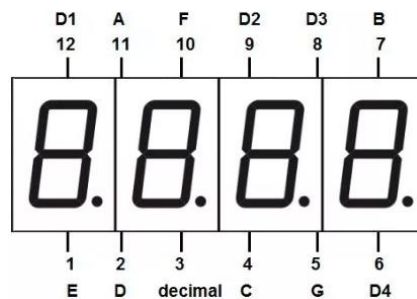
ფიგ. 12.1 მთვლელის პარამეტრიზაციის ფანჯარა

როგორც ფიგურიდან ჩანს, ფანჯარაში მთვლელის ტიპად მონიშნულია მხოლოდ UP (მატებაზე-На Увеличение) ჩამრთველი, რაც იმას ნიშნავს, რომ მთვლელი უბრალოდ ითვლის მასზედ მიწოდებულ იმპულსებს და შესაბამისად იცვლის თავის მდგომარეობას ნოლიდან 32767-მდე, შემდეგი იმპულსი აყენებს მას ნულის მდგომარეობაში და შემდეგ კი თვლის პროცესი ისევ მეორდება.



ფიგ. 12.2 ტეგების ზონა შესასვლელ/გამოსასვლელების შექმნა

რადგან ჩვენ ამ პროექტში დისპლეიდ გამოყენებული გვაქვს როგორც ორსტრიქონიანი თექვსმეტანრიგისანი LED დისპლეი ჩიპზე HD44780, ასევე ოთხტანრიგისანი შვიდსეგმენტა ინდიკატორი, ამიტომ ტეგების ზონაში გამოსასვლელებად გათვალისწინებული უნდა იყოს ამ უკანასკნელის მართვის კონტაქტები, რომლებსაც შევარჩევთ ისეთნაირად, რომ მისი კონტაქტების ნომრები დაემთხვას არდუინოს ნომრებს მაგ. 5 დაემთხვეს არდუინოს მე-5 კონტაქტს, 8 დაემთხვეს არდუინოს მე-8 კონტაქტს და ა.შ.შ. იხილეთ ფიგურები 12.2 და 10.3.

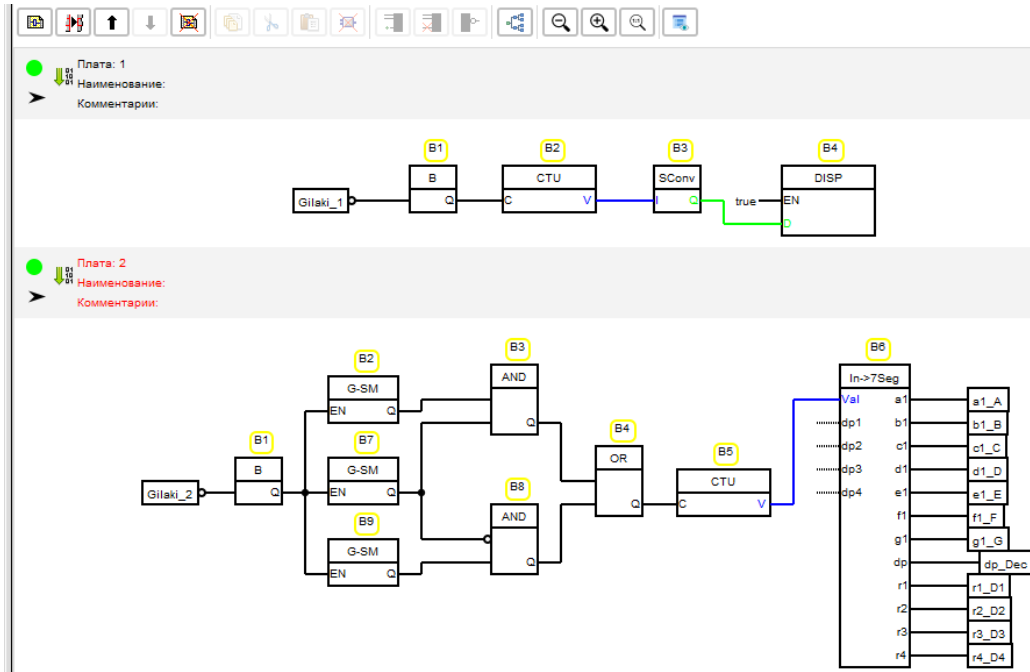


ფიგ. 12.3 ოთხტანრიგა შვიდსეგმენტა ინდიკატორის კონტაქტების ნუმერაცია

ტეგების ზონაში უნდა შევქმნათ აგრეთვე ორი შესასვლელი Gilali_1 და Gilaki_2 ამ მთვლელების მართვისათვის და დავაპროგრამოთ ისინი, როგორც ტეგებში ზონაშია ნაჩვენები, ანუ ყანყალისაგან დაცვისა და ამომქაჩავი რეზისტორის აღმების მონიშვნით. ჩამოვიტანოთ ეს შესასვლელები პროგრამის სამუშაო ზონაში (ფიგ. 12.4) და მასთან ერთად ამავე ზონაში გადმოვიტანოთ პროგრამის ელემენტების ბიბლიოთეკიდან ბლოკები Bounce -

B1, მთვლელი-Счетчик - B2, სტრიქონების კონვერტაციის ბლოკი-Конвертация строк - B3 და დისპლეი ჩიპზე-дисплей на чипе HD44780 - B4.

განვალაგოთ ეს ბლოკები პროგრამის პირველ დაფაზე-Плата: 1, ისე როგორც ეს ფიგ. 12.4-ზეა ნაჩვენები, შევადროთ ისინი ამავე ნახაზის მიხედვით და შრემდეგ დავაპროგრამოთ დისპლეი ისე, როგორც ეს ფიგ. 12.5-ზეა ნაჩვენები.



ფიგ. 12.4 მთვლელის მუშაობის პროგრამა

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № 1 Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND SDA - A4 (I2C) Строк 2 Адрес 27 Из файла

VCC - +5B SCL - A5 (I2C) Столбцов 16 Комментарий

Данные:

Вход

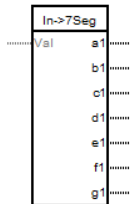
Константа

Строка: 1 Центровать

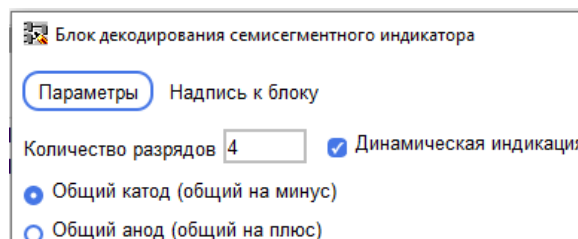
ფიგ. 12.5 დისპლეის ჩიპზე HD44780 პარამეტრიზაცია

მეორე მთვლელის მართვის პროგრამის ბლოკების განლაგებისთვის გამოვიყენოთ პროგრამის მეორე დაფა-Плата: 2. გადმოვიტანოთ ამ დაფაზე ბლოკები: Bounce - B1, სამი

გენერატორის ბლოკი - B2, B7 და B9, ორი ლოგიკური გამრავლების ბლოკი - B3 და B8, ერთი ლოგიკური შეკრების ბლოკი - B4, ერთი მთვლელის ბლოკი - B5 და ერთი შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკი-Блок декодирования семисегментного индикатора - B6, რომელსაც დასაწყისში აქვს სახე (ფიგ. 10.6).



ფიგ. 12.6 შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკი



ფიგ. 12.7 შვიდსეგმენტა ინდიკატორის დეკოდირების პარამეტრიზაცია

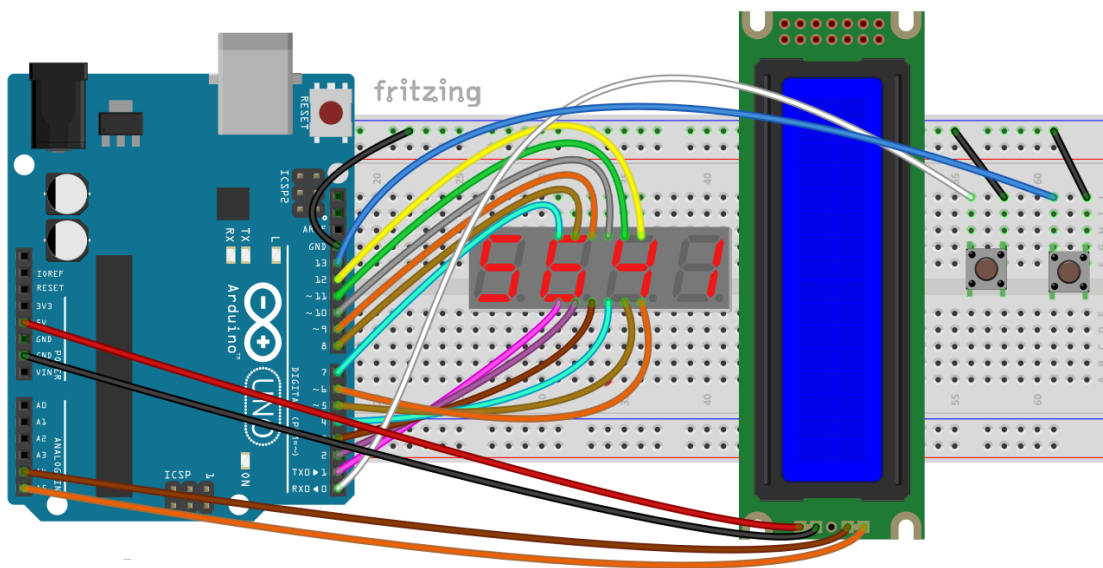
მაგრამ როდესაც ჩვენ დავაპროგრამებთ ამ ბლოკს ფიგ. 10.7-ზე ნაჩვენები გამოსახულების მიხედვით, მაშინ ბლოკი მიიღებს ფიგ. 12.4-ზე ნაჩვენებ სახეს-B6. შევაერთოთ ყველა ეს ბლოკი ერთმანეთთან ფიგ. 12.4 - ზე მოცემული ნახაზის მიხედვით. შევუერთოთ აგრეთვე ყველა გარე ელემენტი: ორი დილაკი, დისპლეი ჩიპზე HD44780 და შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკი არდუინოს კონტაქტებს, შემდეგ ჩავტვირთოთ პროგრამა არდუინოში და შევუდგეთ პროგრამის ტესტირებას.

დავაჭიროთ თითო დილაკს Gilali_1, როგორც აღნიშნული იყო წინა პროექტებში, B1 ბლოკი მუშაობს სტანდარტულად, უბრალოდ 40 მილიწამის განმავლობაში ანეიტრალებს დილაკის კონტაქტების მიერ წარმოშობილ ხელშემლის იმპულსებს მასზე თითის დაჭერის შემთხვევაში. შევამოწმოთ დისპლეი ჩიპზე HD44780. მის მირველ სტრიქონში ცენტრალურ ნაწილი უნდა დაიწეროს რიცხვი 1, შემდეგ კიდევ დავაჭიროთ თითი Gilali_1 და ახლა უკვე უნდა დაიწეროს რიცხვი 2 და ასე შემდეგ მრავალჯერ, ანუ მასზედ უნდა აისახოს იმპულსების თვლის პროცესი.

ახლა გადავიდეთ მეორე ნაწილის განხილვაზე. როგორც ფიგ. 12.4-დან ჩანს, აქ გვაქვს სამი გენერატორი - B2, B7 და B9, რომლებიც უნდა დავაპროგრამოთ როგორც სიმეტრიული მულტივიზბრატორები, იმ განსხვავებით, რომ პირველი გენერატორს B2-ს ვაპროგრამირებთ 100

მლწ იმპულსის ხანგრძლიობაზე, მეორეს B7-ს 2 წამიან იმპულსზე, ხოლო მესამეს B9 კი 500 მილიწამიან იმპულსის ხანგრძლიობაზე. როდესაც Gilaki_2-ზე თითო დაჭერილი არ არის, მაშინ ლოგ. „0“ მისი ინვერტირებული გამოსასვლელიდან კეტავს ამ სამთავე გენერატორს, რადგან ნებისდართვის შესასვლელებზე EN ლოგიკური ნული არ აძლევს საშუალებას გენერატორებს იმუშავონ. ამ ფაქტორებიდან გამომდინარე მთვლელის B5 შესასვლელზე იმპულსები არ მიეწოდება, იგი იმყოფება ნულოვან მდგომარეობაში და ინდიკატორი ასახავს 0000 ციფრს. თუ ჩვენ დავაჭერთ გარკვეული დროის განმავლობაში Gilaki_2-ზე თითოს, მაშინ დავინახავთ, რომ როდესაც B7 გენერატორის გამოსასვლელზე ლოგიკური ნოლია, მისი ინვერტირებული სიგნალი B8 ბლოკის შესასვლელზე იქნება ერთიანი, რაც გამოიწვევს B8 ლოგიკური გამრავლების ბლოკის გაღებს და იგი გაატარებს თავის გამოსასვლელზე 500 მლწ იმპულსებს. ამ დროს B3 ლოგიკური გამრავლების ბლოკი ჩაკეტილია B7 ბლოკის გამოსასვლელზე არსებული ლოგიკური ნულით და ამიტომ B4 ლოგიკური შეკრების ბლოკის გამოსასვლელზეც იქნება მხოლოდ 500 მლწ-იანი იმპულსები. ორი წამის გავლის შემდეგ როდესაც B7 ბლოკის გამოსასვლელზე ლოგიკური ერთიანი გახდება, მაშინ ჩაიკეტება B8 ბლოკი და გაიხსნება B3 ბლოკი, რომელიც გაატარებს 100 მლწ იმპულსებს და რომელიც შემდეგ ლოგიკური შეკრების ბლოკს B4 გავლით მიეწოდება იგივე მთვლელის შესასვლელს. მაშასადამე მთვლელის შესასვლელზე მიეწოდება იმპულსები თანმიმდევრობა ორი სხვადასხვა სიხშირით, ორ-ორი წამიანი გადართვებით და ეს თვლის სიხშირეების ცვლილება აისახება შვიდსეგმენტური ოთხთანრიგა ინდიკატორის შუქდიოდებზე და იქნება თვალისათვის შესამჩნევი.

მთვლელის მუშაობის Fritzing სქემა ნაჩვენებია ფიგ. 12.8-ზე.



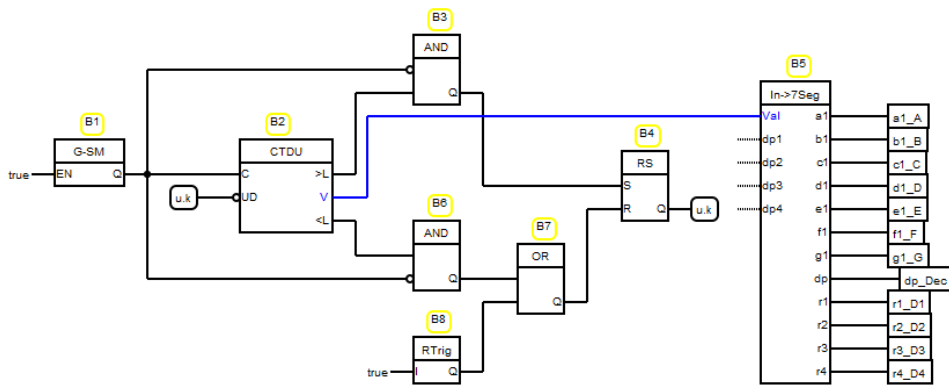
ფიგ. 12.8 მთვლელის მართვის ელემენტების შეერთების Fritzing სქემა

პროექტი_13

რევერსიული მთვლელი

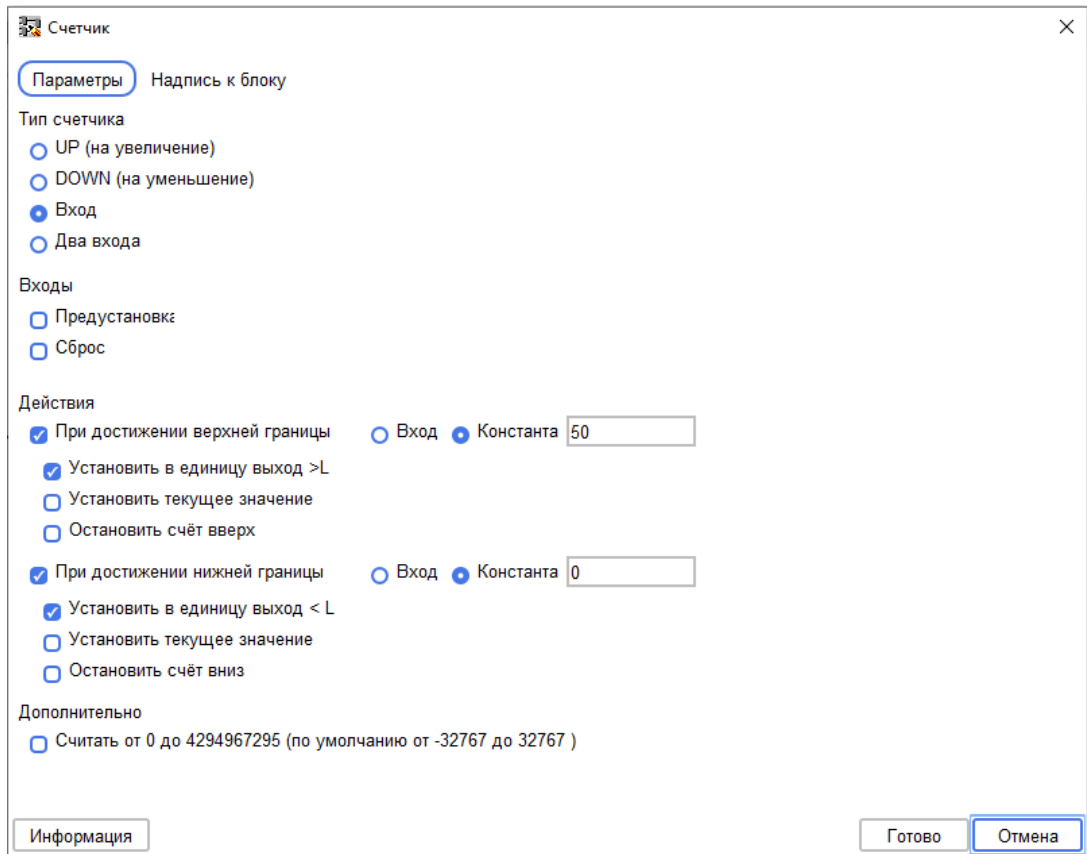


FLP-ში გამოყენებულ მთვლელის ბლოკს-Счетчик აქვს შესაძლებლობა იმუშაოს რევერსიულ რეჟიმში. ამ პროექტში დავაპროგრამებთ მთვლელს, რომელიც ჯერ ითვლის პირდაპირი თვლის რეჟიმში რაღაც ნებისმიერ ადებულ სიდიდემდე (ამ კონკრეტულ შემთხვევაში ადებული გვაქვს რიცხვი 50) შემდეგ გადადის უკუ თვლის რეჟიმში და ჩამოდის ნულამდე, შემდეგ გადადის ისევ პირდაპირი თვლის რეჟიმში და ა.შ. უსასრულობამდე. აღნიშნული რევერსიული მთვლელის პროგრამა ნაჩვენებია ფიგ. 13.1-ზე.



ფიგ. 13.1. რევერსიული მთვლელის პროგრამა

სათვლელი იმპულსების გამომუშავებისათვის პროექტში გამოყენებულია სიმეტრიული მულტივიბრატორი B1, რომელიც დაპროგრამებულია იმპულსის ხანგრძლიობაზე სიდიდით 250 მლწ. მთვლელის-Счетчик B2 პარამეტრიზაციის ფანჯარა ასახულია ფიგ. 13.2-ზე. როგორც ამ ფანჯრიდან ჩანს, მთვლელის ტიპად-Тип счетчика არჩეულია შესასვლელი-вход, ეს იმას ნიშნავს, რომ როდესაც ამ შესასვლელზე ლოგიკური ერთიანია, მაშინ მთვლელი მუშაობს ზრდაზე ხოლო როდესაც ლოგიკური ნოლია, მაშინ მთვლელი მუშაობს კლებაზე (თუ მთვლელის C შესასვლელზე მიეწოდება იმპულსები რათქმაუნდა).

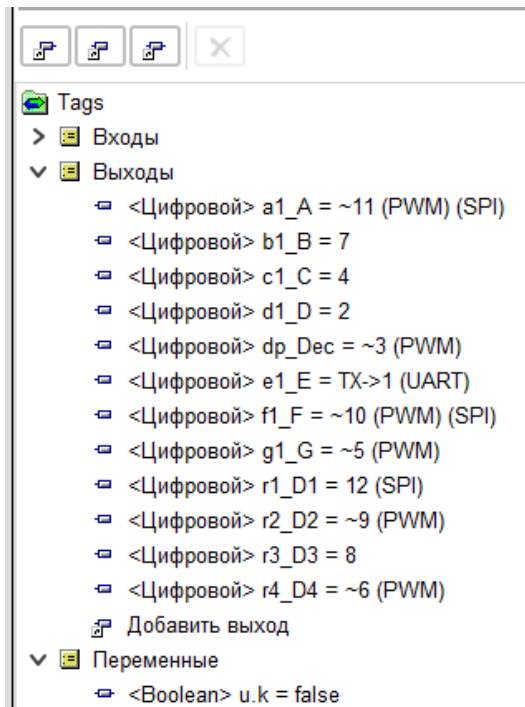


ფიგ. 13.2. მთვლელის პარამეტრიზაციის ფანჯარა

ამავე ფანჯრიდან ჩანს, რომ როდესაც მთვლელი მიაღწევს ზედა ზღვარს პირდაპირი თვლის რეჟიმში, ანუ დათვლის 50 იმპულსს, მაშინ გამომუშავდება სიგნალი >L, ხოლო როდესაც მიაღწევს ქვედა ზღვარს უკუ თვლის რეჟიმში, ანუ ჩამოვა ნოლამდე, მაშინ გამომუშავდება <L სიგნალი. სწორედ ეს სიგნალები უნდა გამოვიყენოთ ჩვენ რევერსული მთვლელის მართვისათვის. ამისათვის დავუბრუნდეთ ისევ ჩვენს პროგრამას ფიგურაზე 13.1.

შევქმნათ ტეგების ზონაში გამოსასვლელები და u.k ცვლადი (ფიგ. 13.3.).

გადმოვიტანოთ პროგრამის სამუშაო ზონაში ელემენტების ბიბლიოთეკიდან ორი ლოგიკური გამრავლების ბლოკი [END]-B3 და B6, ერთი ლოგიკური შეკრების ბლოკი [OR]-B7, ერთი Rtrig ტრიგერი-B8 და ერთი [RS] ტრიგერი- B4 (ფიგ. 13.1).



ფიგ. 13.3. ტეგების ზონის ფორმირება

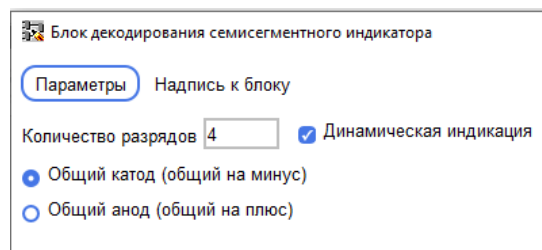
რევერსიული მთვლელი მუშაობს შემდეგნაირად. შევავერთოთ ეს ბლოკები ერთმანეთთან ისე როგორც ეს ფიგ. 13.1-ზეა აღნიშნული და გადავტვირთოთ პროგრამა არდუინოში. პროგრამის გაშვების შემდეგ პირველად და უკანასკნელად გაიშვება Rtrig ტრიგერი, რომელიც გამოიმუშავებს თავის გამოსასვლელზე ერთეულოვან იმპულსს პროგრამის ერთი ციკლის ხანგრძლიობით. ეს იმპულსი B7 ბლოკის გავლით დააყენებს B4 ტრიგერს ნოლიანის მდგომარეობაში. როგორც ფიგურიდან 13.1 ჩანს, ნოლოვანი სიგნალი ამ ტრიგერის გამოსასვლელიდან u.k უკუკავშირის ცვლადის გავლით მიეწოდება მთვლელის ud შესასვლელზე, ინვერტირდება და დააყენებს მას პირდაპირი თვლის რეჟიმზე, ანუ მატებაზე.

ამის შემდეგ, მთვლელი დაიწყებს მასზედ მიწოდებულ იმპულსების თვლას გენერატორის ბლოკიდან B1 და როდესაც იგი დაითვლის 50-მდე, მის >L გამოსასვლელზე გამოიმუშავდება ლოგიკური ერთიანი, რომელიც მიეწოდება ლოგიკური გამრავლების ბლოკს B3-ს და როდესაც გენერატორის B1 გამოსასვლელზე იმპულსი შეიცვლება პაუზით (დადგება ლოგიკური ნოლი), მაშინ მისი ინვერტირებული სიგნალი B3 ბლოკის შესასვლელზე გამოიწვევს მის გამოსასვლელზე ლოგიკური ერთიანის წარმოშობას, ეს უკანასკნელი კი დააყენებს [RS] ტრიგერის B4 ერთიანის მდგომარეობაში. ამ ტრიგერის ინვერტირებული სიგნალი კი მთვლელის ud შესასვლელზე შეუცვლის ამ მთვლელს მუშაობის რეჟიმს და

დააყენებს მას კლების რეჟიმზე, რის შედეგადაც გენერატორის ყოველი შემდგომი იმპულსი გამოიწვევს მთვლელის შემცველობის კლებას 50-დან ნოლამდე.

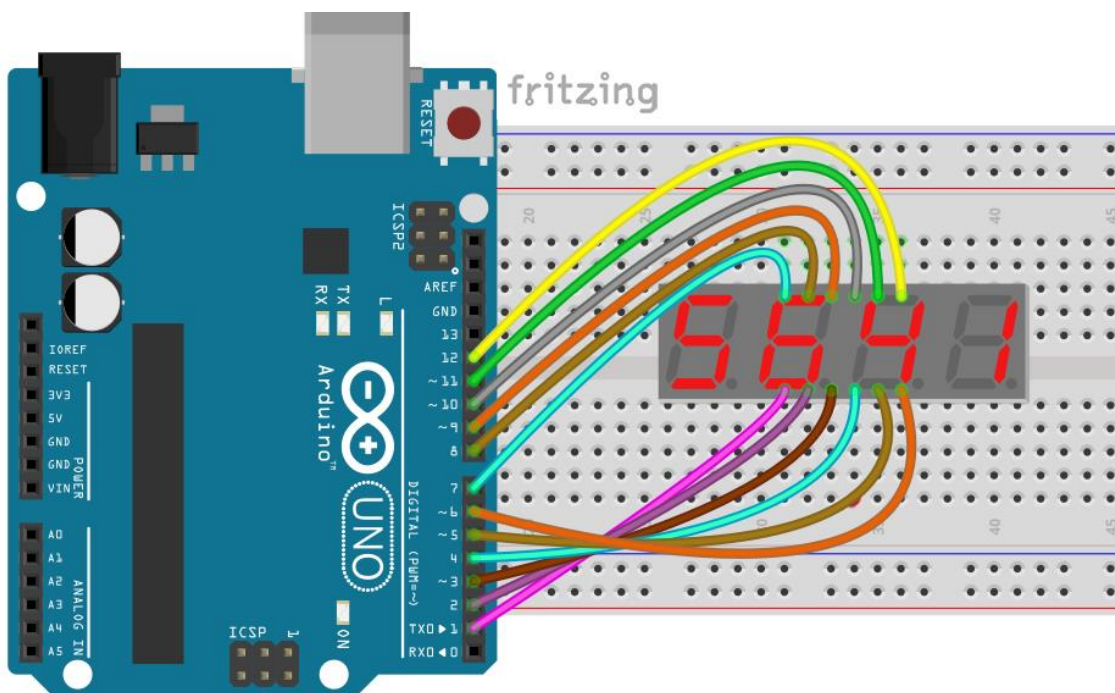
როდესაც მთვლელში ჩაიწერება ნულოვანი სიდიდე მაშინ გამოუმუშავდება ლოგიკური ერთიანის სიგნალი მთვლელის <L გამოსასვლელზე, და როდესაც გენერატორის გამოსასვლელზე დადგება პაუზა, მაშინ ამუშავდება B6 ლოგიკური გამრავლების ბლოკი, რომელიც B7 ლოგიკური შეკრების ბლოკის სახით ისევ დააყენებს [RS] ტრიგერის ნულიანის მდგომარეობაში და ა.შ.შ პროცესი განმეორდება ისევ და ისევ.

მთვლელის მუშაობის ვიზუალური კონტროლისათვის გამოყენებული გვაქვს ოთხთანრიგაშვიდსეგმენტა ინდიკატორი, რისთვისაც ელემენტების ზონიდან გადმოვიტანთ შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკს-Блок декодирования семисегментного индикатора B5 და დავაპროგრამირებთ მას ფიგურა 13.4. წარმოდგენილი სქემის მიხედვით.



ფიგ. 13.4 შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკის პარამეტრიზაცია

რევერსიული მთვლელის Fritzing სქემა ასახულია ფიგ. 13.5-ზე



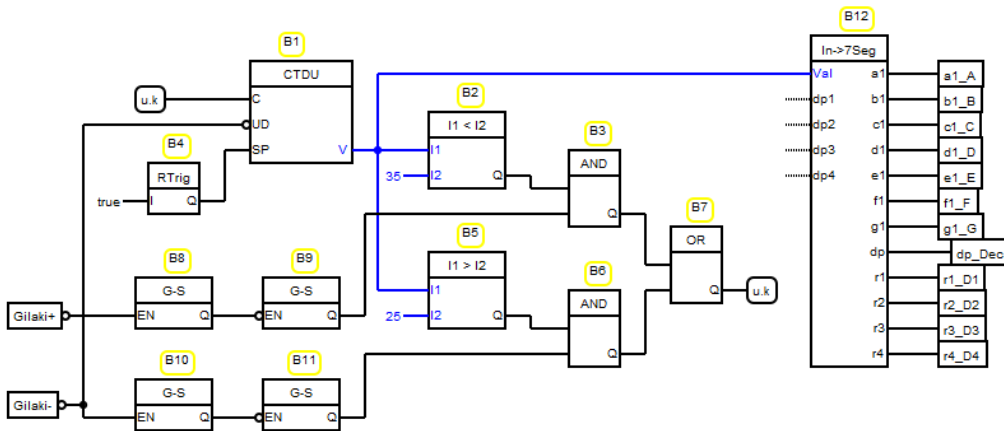
ფიგ. 13.5 რევერსიული მთვლელის Fritzing სქემა

პროექტი_14

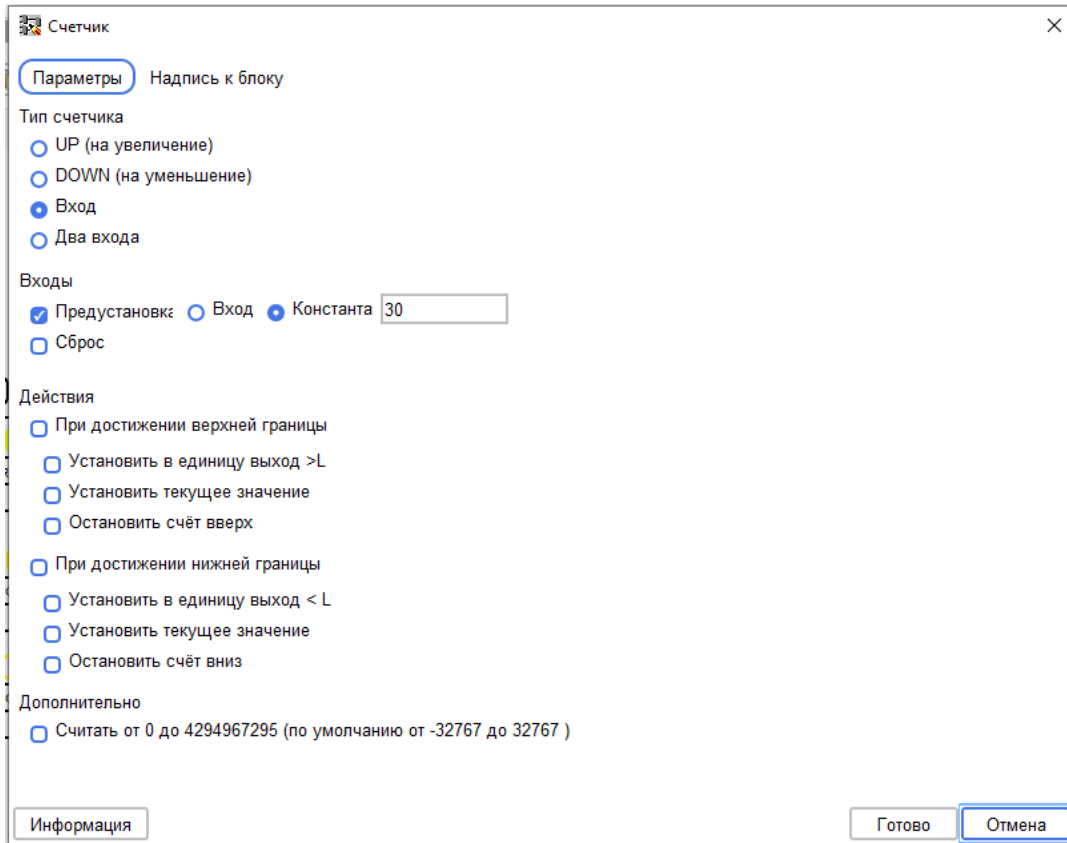
რიცხვების გენერატორი



ზოგ შემთხვევაში საჭირო ხდება რაიმე ბაზური რიცხვის გენერირება მის ირგვლივ არსებულ რიცხვების რაიმე დიაპაზონთან ერთად. ეს ამოცანა ადვილად შეიძლება გადაწყდეს მთვლელის გამოყენებით და მისი შესაბამისი დაპროგრამებით. ამ პროექტში ნაჩვენებია ასეთი ამოცანის გადაწყვეტის ერთერთი გზა. პროექტში ბაზურ რიცხვად აღებულია ჩვენს მიერ შერჩეული ნებისმიერად აღებული რიცხვი, მაგალითად 30, ხოლო რიცხვების დიაპაზონი აღებული გვაქვს 25-დან 35-ის საზღვრებში. პროექტში მთავარ ელემენტს წარმოადგენს მთვლელის ბლოკი-Счетчик B1 (ფიგ. 14.1), რომელიც დაპროგრამებულია ფიგურა 14.2-ზე გამოსახული ფანჯრის შესაბამისად.

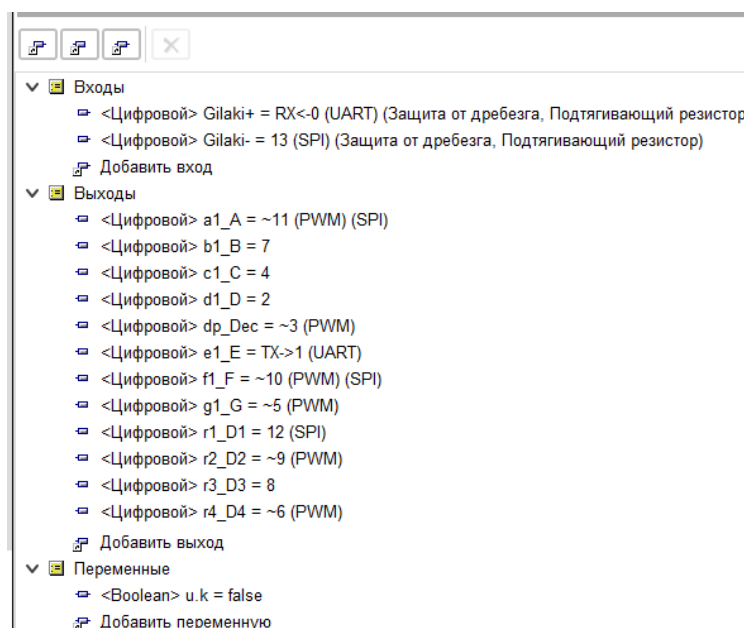


ფიგ. 14.1 რიცხვების გენერატორის პროგრამა



ფიგ. 14.2 მოვლელის პარამეტრიზაციის ფანჯარა

გარდა ამისა ჩვენ დაგვჭირდება არდუინოს შესასვლელ/გამოსასვლელების შექმნა ტეგების ზონაში და აგრეთვე ერთი ცვლადის შექმნა სახელწოდებით u.k, რომლებსაც შევქმნით ფიგ. 14.3-ზე მოცემული სქემის მიხედვით.



ფიგ. 14.3 ტეგების ზონის პარამეტრიზაციის ფანჯარა

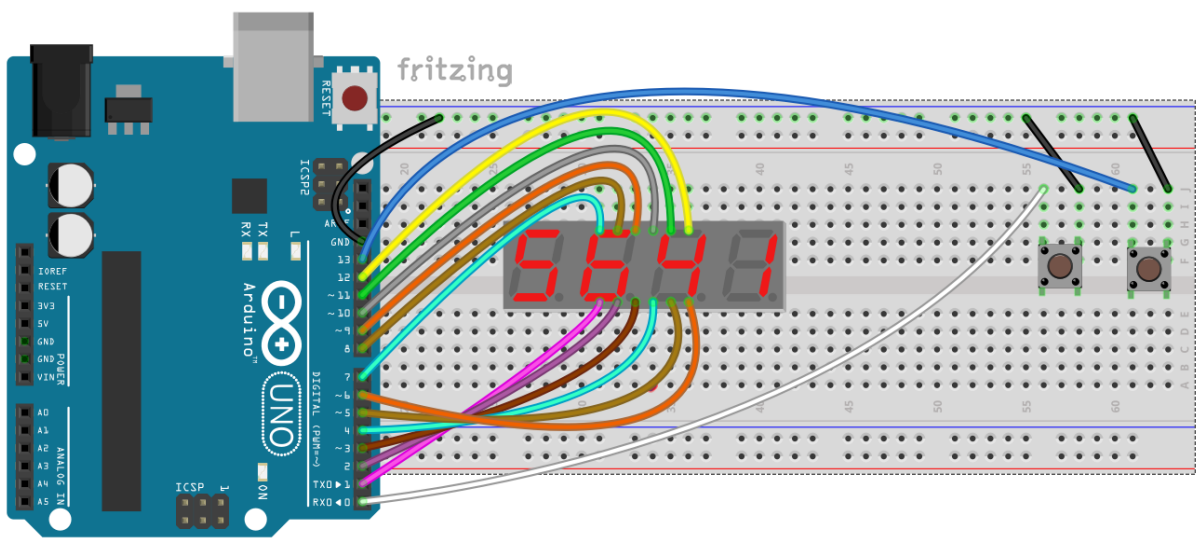
როგორც ფიგურა 14.2-დან ჩანს მთვლელის ტიპად-Тип счетчика აღებულია ისეთი მთვლელი, რომლის თვლის მიმართულება ზრდაზე თუ კლებაზე იმართება შესასვლელით-Вход ე.ი. ამ შესასვლელით განისაზღვრება თუ როგორი მიმართულებით იმუშავებს მთვლელი ზრდადობაზე თუ კლებადობაზე. როგორც უკვე ვიცით, თუ ამ შესასვლელზე ლოგიკური ერთიანია, მაშინ იგი მუშაობს ზრდადობაზე, ხოლო თუ ლოგიკური ნოლია მაშინ იგი მუშაობს კლებადობაზე. ამავე ფიგურაზე ჩართული გვაქვს წინასწარ დაყენების რეჟიმი-Предустановка, რისთვისაც მონიშნული გვაქვს კონსტანტა-Константа და შესაბამის ველში ჩაწერილი გვაქვს ჩვენს მიერ წინასწარ შერჩეული რიცხვი 30, დანარჩენი ალმები დატოვილი გვაქვს უცვლელად.

ახლა გადავიდეთ თვითონ პროგრამის განმარტებაზე, როგორც აღნიშნული იყო აქ გამოყენებული გვაქვს მთვლელის ბლოკი B1, რომელმაც პარამეტრიზაციის შემდეგ მიიღო ფიგურა 14.1-ზე ნაჩვენები სახე. რიცხვების დიაპაზონის ფორმირებისათვის ჩვენ დაგვჭირდება ორი კომპარატორის ბლოკი B2 და B5, რომელთაგან პირველს ვაპროგრამირებთ 35-ზე ნაკლებობაზე, ხოლო მეორეს 25-ზე მეტობაზე. პროექტში B1 მთვლელის მართვისთვის გამოყენებული გვაქვს არდუინოს ორი შესასვლელი Gilaki+ და Gilaki-, რომლებიც გამოიყენება მთვლელის რიცხვის მატებისათვის ან კლებისათვის.

ამ შესასვლელებზე მიერთებული ფიზიკური ღილაკებიდან ღილაკებიდან გამოსული სიგნალები გაივლიან ორორ გენერატორს, რომლებიც ერთვიბრატორის მუშაობის რეჟიმშია დაპროგრამებული ათ ათი მილიწამიანი სიგნალებით, ესენია: B8, B9 და B10, B11, ეს ერთვიბრატორები ასრულებენ მხოლოდ და მხოლოდ ხელშეშლის იმპულსების ჩახშობის ფუნქციას ღილაკზე თითის დაჭერის ან გათიშვის მომენტში.

განვიხილოთ ახლა მოწყობილობის მუშაობის პრინციპი. პროგრამის არდუინოში ჩატვირთვის შემდეგ, საწყის მომენტში ამუშავდება Rtrig - B4 და მის მიერ გამომუშავებული სიგნალით ჩაწერს რიცხვს 30-ს მთვლელში B1. ამავდროულად ეს რიცხვი მიეწოდება შვიდსეგმენტა ინდიკატორს B12 და მასზედ აისახება რიცხვი 30. ახლა დავიწყით ღილაკზე Gilaki+ თითის დაჭერა. როდესაც Gilaki- - ზე თითის დაჭერილი არ არის, მაშინ მის

გამოსასვლელზე ნოლოვანი სიგნალია, ხოლო მთვლელის UD შესასვლელზე-მისი ინვერსია, ანუ ლოგიკური ერთი და ამიტომ მთვლელი იმყოფება პირდაპირი თვლის რეჟიმში, Gilaki+-ზე თითის პირველი დაჭერისას B2 კომპარატორის გამოსასვლელზე ლოგიკური ერთია, რადგან ის რიცხვი ანუ 30 რაც მთვლელშია ჩაწერილი ნაკლებია 35-ზე, ამიტომ ლოგიკური გამრავლების გამოსასვლელზე B3 გაჩნდება ლოგიკური ერთიანი, რომელიც ლოგიკური შეკრების ბლოკის B7-ის და ცვლადის u.k-გავლით მიეწოდება მთვლელის თვლის შესასვლელს C-ს და დაუმატებს მის შემცველობას ერთს, ანუ მთვლელში დაფიქსირდება რიცხვი 31, რაც აისახება კიდეც B12 ინდიკატორზე. თუ გავაგრძელებთ ჩვენ ღილაკზე Gilaki+ თითის რამდენჯერმე დაჭერას, მაშინ ინდიკატორზე აისახება რიცხვები, 32, 33, 34, 35, მაგრამ ამის შემდეგ მეტის მომატება აღარ დაფიქსირდება, რადგან B2 კომპარატორი შეიცვლის მდგომარეობას, მის გამოსასვლელზე დადგება ლოგიკური ნულიანი და ჩაკეტავს შემდეგი იმპულსების გატარებას მთვლელის C შესასვლელზე. ახლა თუ დავაჭერთ Gilaki-, მაშინ ერთეულოვანი სიგნალი მისი გამოსასვლელიდან ინვერტირების შემდეგ გადაიყვანს მთვლელს B1 კლებადობის რეჟიმში, მაგრამ იგივე სიგნალი B6, B7 ბლოკებისა და ცვლადის u.k-ს გავლით ისევ მიეწოდება მთვლელის C შესასვლელს, რაც გამოიწვევს მთვლელის შემცველობიდან ერთი ერთეულის გამოკლებას. თუ მაგალითად ჩვენ Gilaki- ღილაკს დავაჭირებთ თითი მაშინ, როდესაც მთვლელში ჩაწერილი იყო 33, მაშინ თითის დაჭერის შემდეგ მასში დაფიქსირდება რიცხვი 32, რაც აისახება კიდეც ინდიკატორზე. ამრიგად, აღნიშნული პროგრამით ჩვენ შეიძლება ვაგენერიროთ ნებისმიერი რიცხვი მთვლელის გამოსასვლელზე და ოპერატულად ვცვალოთ იგი ცვენს მიერ შერჩეულ დიაპაზონში. მოწყობილობის Fritzing სქემა ასახულია ფიგურა 14.4-ზე.



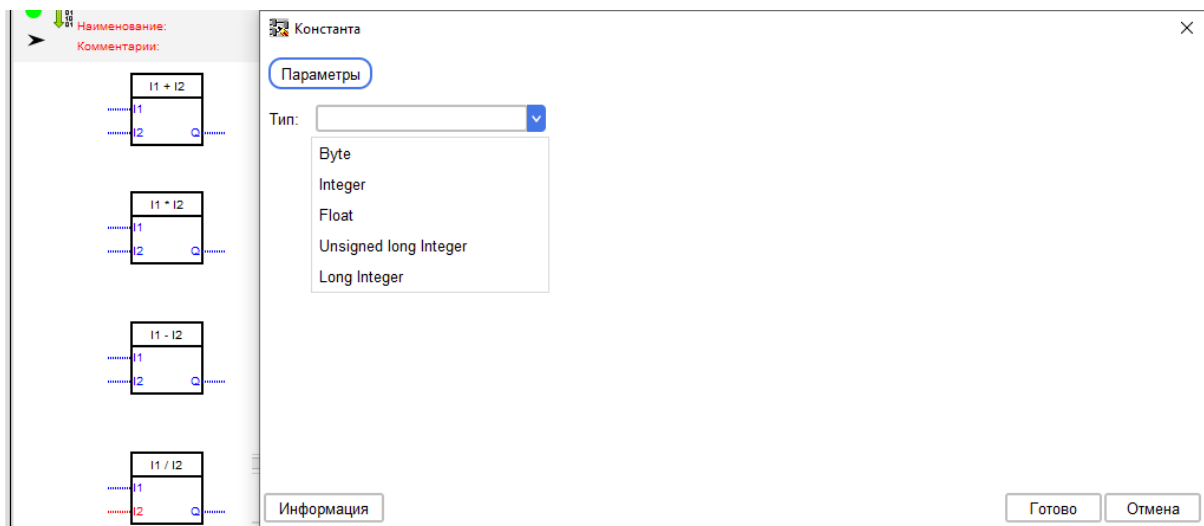
ფიგ. 14.4 რიცხვების გენერატორის Fritzing სქემა

პროექტი_15

მათემატიკა_1



ამ პროექტში განხილული იქნება ის მათემატიკური ოპერაციები, რომლებიც შეიძლება რეალიზებულ იქნას FLProg-ში. ესენია სტანდარტული შეკრების, გამრავლების, გამოკლების და გაყოფის ოპერაციები. შევიდეთ პროგრამის ბლოკების ბიბლიოთეკის საქალაქო მათემატიკა-Математика და პროგრამის სამუშაო ზონაში გადმოვიტანოთ ბლოკები: [SUM], [MUL], [SUB] და [DIV] (ფიგ. 15.1).



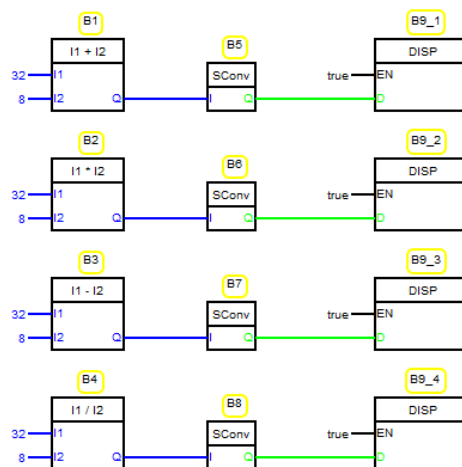
ფიგ. 15.1 პროგრამის ძირითადი ოთხი მათემატიკური ოპერაცია შეკრება, გამრავლება, გამოკლება, გაყოფა

როგორც ფიგურა 15.1-დან ჩანს, თვითოეული ბლოკი გაანგარიშებულია მხოლოდ ორ ოპერანდზე მოქმედებების საწარმოებლად. თუ გვინდა მრავალ ოპერანდული ოპერაციების შესრულება, მაშინ უნდა გამოვიყენოთ კასკადური სქემები. ახლა ვნახოთ თუ როგორი ტიპის მონაცემები შეიძლება იყოს მიწოდებული ამ შესასვლელზე, ამისთვის შევიდეთ ნებისმიერ ამ შესასვლელის კონტექსტურ მენიუში თავის მარჯვენა ღილაკის დაწკაპუნებით, მოვნიშნოთ იქ კონსტანტის ჩასმა-Вставить константу, დავაჭიროთ მასზედ თავის მარცხენა

ლილაკით და ეკრანზე გამოვა კონსტანტის პარამეტრები-Параметры. როგორც ვხედავთ აქ დასაშვებია მონაცემთა ხუთი სხვადასხვა ტიპი: ბაიტი-Bayte-სადაც ინფორმაციამ შეიძლება მიიღოს მნიშვნელობა 0-დან 255-მდე, მთელი-Intiger სადაც ინფორმაციამ შეიძლება მიიღოს მნიშვნელობა -32768-დან +32767-მდე; მცოცავმძიმისანი-Float, სადაც ინფორმაციამ შეიძლება მიიღოს მნიშვნელობა -3.4028235e38-დან 3.4028235e38-მდე, უნიშნო გრძელი მთელი-Usigned long Integer სადაც ინფორმაციამ შეიძლება მიიღოს მნიშვნელობა 0-დან 0-დან 4294967295-მდე და გრძელი მთელი-Long Integer სადაც ინფორმაციამ შეიძლება მიიღოს მნიშვნელობა -2147483648-დან 2147483647-მდე.

შენიშვნა: აქ ლაპარაკი იყო კონსტანტის ჩასმაზე, მაგრამ ამ შესასვლელზე შეიძლება მიწოდებულ იქნას აგრეთვე ერთერთი ამ ზემოთმოხსენებული ტიპის ცვლადი იმფორმაცია ნებისმიერი ინფორმაციის წყაროდან.

მაგალითის სახით ჩავუწეროთ თვითოეულ ამ მათემატიკური ბლოკის B1...B4 შესასვლელზე ერთიდაიგივე კონსტანტები: კერძოთ I1 შესასვლელზე ჩავწეროთ 32, ხოლო I2 შესასვლელზე - 8 (ფიგ. 15.2).



ფიგ. 15.2 მათემატიკური ბლოკების დაპროგრამება

ახლა პროგრამის ბლოკების ბიბლიოთეკის საქალაქედან დისპლეები-Дисплей პროგრამის სამუშაო ზონაში გადმოვიტანოთ ორსტრიქონიანი 16 პოზიციანი ბლოკი დისპლეი ჩიპზე-HD44780-Дисплей на чипе HD44780 რომელიც ნახაზზე აღნიშნული გვაქვს მე-9 ნომრით. რადგანაც ამ დისპლეიზე ასახული უნდა იყოს ოთხი სხვადასხვა ინფორმაცია, ამიტომ ეს ბლოკი გაყოფილი გვაქვს ოთხ ნაწილად და აღნიშნული გვაქვს როგორც B9_1...B9_4. მათი პარამეტრიზაციის ფანჯრები ასახულია ფიგ. 15.3 ... ფიგ. 15.6. როგორც ამ ფიგურებიდან ჩანს თვითოეულ სტრიქონში პირველი ინფორმაციის ასახვა იწყება პირველი

სვეტიდან, ხოლო მეორე ინფორმაციის ასახვა მე-9 სვეტიდან. დანარჩენი პარამეტრები ერთნაირია ყველა ბლოკისათვის B9_1...B9_4.

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № 1 Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	2	Адрес	27	Из файла
VCC - +5V	SCL - A5 (I2C)	Столбцов	16	Комментарий		

Данные:

Вход

Константа

Строка: 1 Центровать

Столбец: 1

ფიგ. 15.3. დისპლეი ჩიპზე HD4478 პირველი სტრიქონის პირველი ნაწილის პარამეტრზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № 1 Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	2	Адрес	27	Из файла
VCC - +5V	SCL - A5 (I2C)	Столбцов	9	Комментарий		

Данные:

Вход

Константа

Строка: 1 Центровать

Столбец: 9

ფიგ. 15.4. დისპლეი ჩიპზე HD4478 პირველი სტრიქონის მეორე ნაწილის პარამეტრზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № 1 Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	2	Адрес	27	Из файла
VCC - +5В	SCL - A5 (I2C)	Столбцов	16	Комментарий		

Данные:

Вход

Константа

Строка: 2 Центровать

Столбец: 1

ფიგ. 15.5. დისპლეი ჩიპზე HD4478 მეორე სტრიქონის პირველი ნაწილის პარამეტრზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № 1 Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	2	Адрес	27	Из файла
VCC - +5В	SCL - A5 (I2C)	Столбцов	16	Комментарий		

Данные:

Вход

Константа

Строка: 2 Центровать

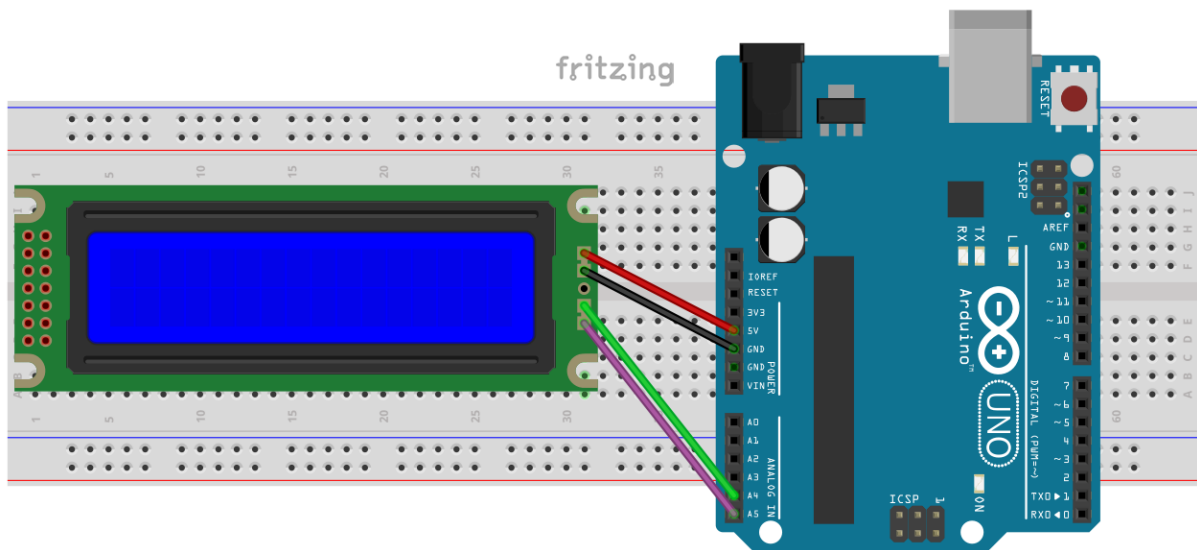
Столбец: 9

ფიგ. 15.6. დისპლეი ჩიპზე HD4478 მეორე სტრიქონის მეორე ნაწილის პარამეტრზაცია

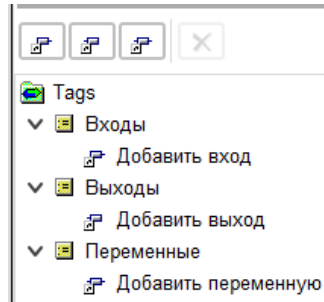
ახლა უნდა შევაერთოთ არითმეტიკული ბლოკების B1...B4 გამოსასვლელები დისპლეის ბლოკების შესასვლელებს B9_1...B9_4, მაგრამ გავიხსენოთ, რომ დისპლეი ვერ აღიქვავს პირდაპირ რიცხობრივ ინფორმაციას, ამიტომ საჭიროა მათი კონვერტაცია სტრიქონულ ფორმატში. ამისათვის ჩვენ დაგვჭირდება ბლოკების ბიბლიოთეკის

საქალაქიდან ტიპების კონვერტაცია-Конвертация типов პროგრამის სამუშაო ზონაში გადმოვიტანოთ ოთხი სტრიქონების კონვერტაციის ბლოკი-Конвертация строк B5...B8 და შევართოთ ყველა ეს ბლოკი დანარჩენ ბლოკებთან ფიგ. 15.2-ზე ნაჩვენები სქემის მიხედვით.

პროგრამის კომპილაციისა და არდუინოში გადატვირთვის შემდეგ დისპლეის პირველ სტრიქონში, დაწყებული პირველი და მეცხრე სვეტებიდან უნდა აისახოს რიცხვები 40 და 256 შესაბამისად (ე.ი. ბლოკმა B1-მა მოახდინა ორი რიცხვის 32-ის და 8-ის შეკრება, ხოლო ბლოკმა B2-მა ამავე ორი რიცხვის გამრავლება) და რეზულტატები გაიგზავლა დისპლეის პირველ სტრიქონში ასახვისათვის). ამის მსგავსად დისპლეის მე-2 სტრიქონში დაწყებული პირველი და მეცხრე სვეტებიდან აისახება B3 და B4 ბლოკების მუშაობის შედეგები, ანუ რიცხვები-24 და 4 შესაბამისად. გთავაზობთ ცვალოთ რიცხვები ნებისმიერად, მაგრამ ესე რომ ოპერაციის შედეგმა არ უნდა გადააჭარბოს რვა თანრიგს, წინააღმდეგ შემთხვევაში მოხდება გადავსება დისპლეისათვის და შედეგი დამახინჯდება, გარდა ამისა უნდა გვახსოვდეს, რომ გაყოფის ოპერაციის შესრულების დროს მთელი რიცხვების შემთხვევაში, მიღებული შედეგი მრგვალდება ნაკლებობით, წილადური რიცხვის უბრალოდ გადაგდებით. მოწყობილობის Fritzing სქემა მოცემულია ფიგ. 15.7-ზე. ხოლო პროექტის ტეგების ზონა გამოუყენებელია, უბრალოდ, არ დაგვჭირვებია მისი გამოყენება (ფიგ. 15.8).



ფიგ. 15.7 მოწყობილობის Fritzing სქემა



ფიგ. 15.8 ტეგების ზონა

პროექტი_16

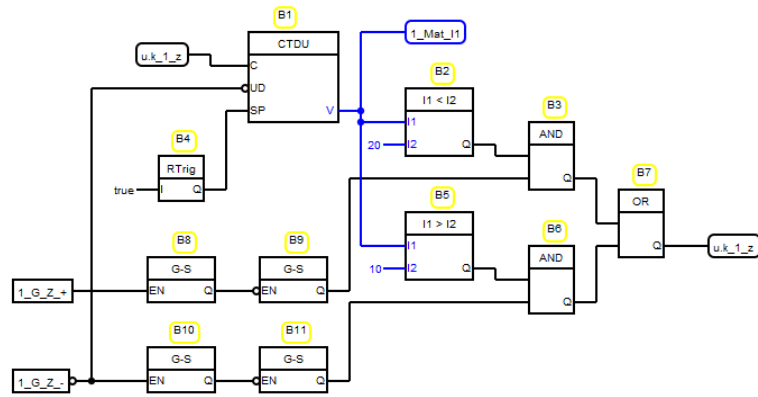
მათემატიკა_2



ამ პროექტში კიდევ უფრო დავუახლოვდებით რეალობას და განვიხილავთ იმ შემთხვევებს, როდესაც მათემატიკური ბლოკების შესასვლელებზე I1 და I2-ზე არა კონსტანტები, არამედ რეალური რიცხვებია მიწოდებული, თანაც ეს რიცხვები შესაძლებელია ოპერატულად იქნას შეცვლილები. ამ მიზნით გამოვიყენებთ წინა ორ პროექტში გავლილ მასალას. კონკრეტულად, ოპერანდების ოპერატორულად ცვლილებისათვის გამოვიყენებთ რიცხვების გენერატორებს და მიღებული შედეგების ასახვის მეთოდებს. აქვე აღვნიშნავთ, რომ რიცხვების გენერატორები განზრახული გვაქვს გამოვიყენოთ ყველა შესასვლელისათვის, რაც მნიშვნელოვნად გაზრდის პროექტის მოცულობას და გაართულებს მის აღქმას. აქედან გამომდინარე, გადავწყვიტეთ ოთხი არითმეტიკული ოპერაციის ნაცვლად განვიხილოთ მხოლოდ ორი არითმეტიკული ოპერაცია, შეკრება და გამრავლება, დანარჩენი ორი არითმეტიკული ოპერაციის გამოკლებისა და გაყოფის მართვა კი შეიძლება განხორციელდეს ანალოგიურად შეკრებისა და გამრავლების ოპერაციების მსგავსად.

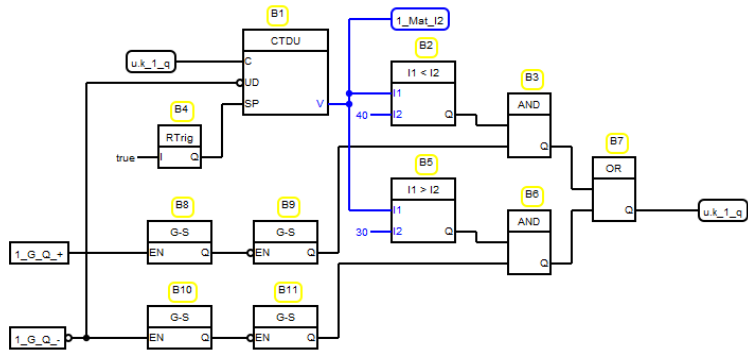
მაშასადამე ჩვენ გვექნება ორი ოპერაცია ორორი შესასვლელით, პირველი ოპერანდითა და მეორე ოპერანდით. თვითოეული ამ შესასვლელისთვის გამოყენებული გვაქვს ცალკე რიცხვების გენერატორი, რომლებიც განლაგებულია FLProg-ის ცალცალკე დაფაზე დაფა: 1, დაფა: 2-Плата: 1...Плата: 4 (ფიგ. 1 ... ფიგ. 4).

● Плата: 1
▼ Наименование:
▶ Комментарий:



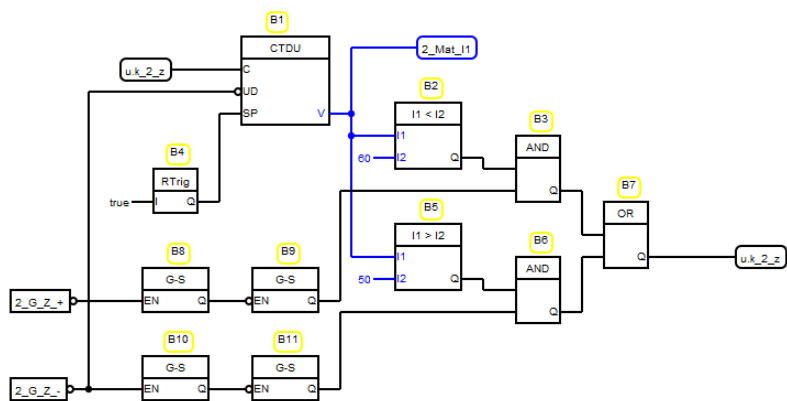
ფიგ. 16.1 რიცხვების გენერატორი_1

● Плата: 2
▼ Наименование:
▶ Комментарий:

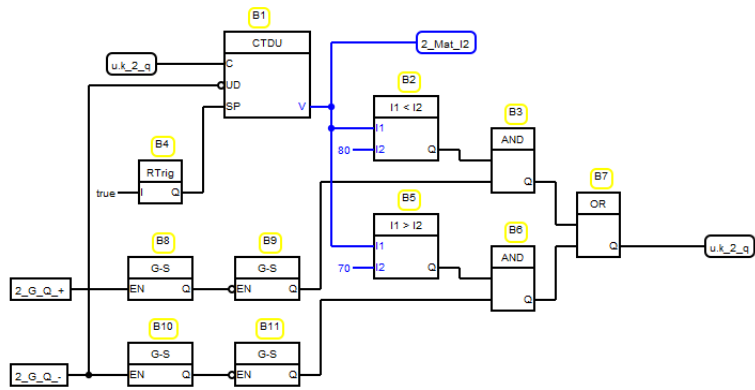


ფიგ. 16.2 რიცხვების გენერატორი_2

● Плата: 3
▼ Наименование:
▶ Комментарий:

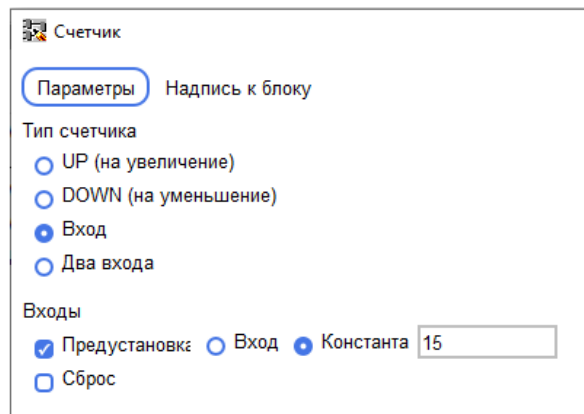


ფიგ. 16.3 რიცხვების გენერატორი_3

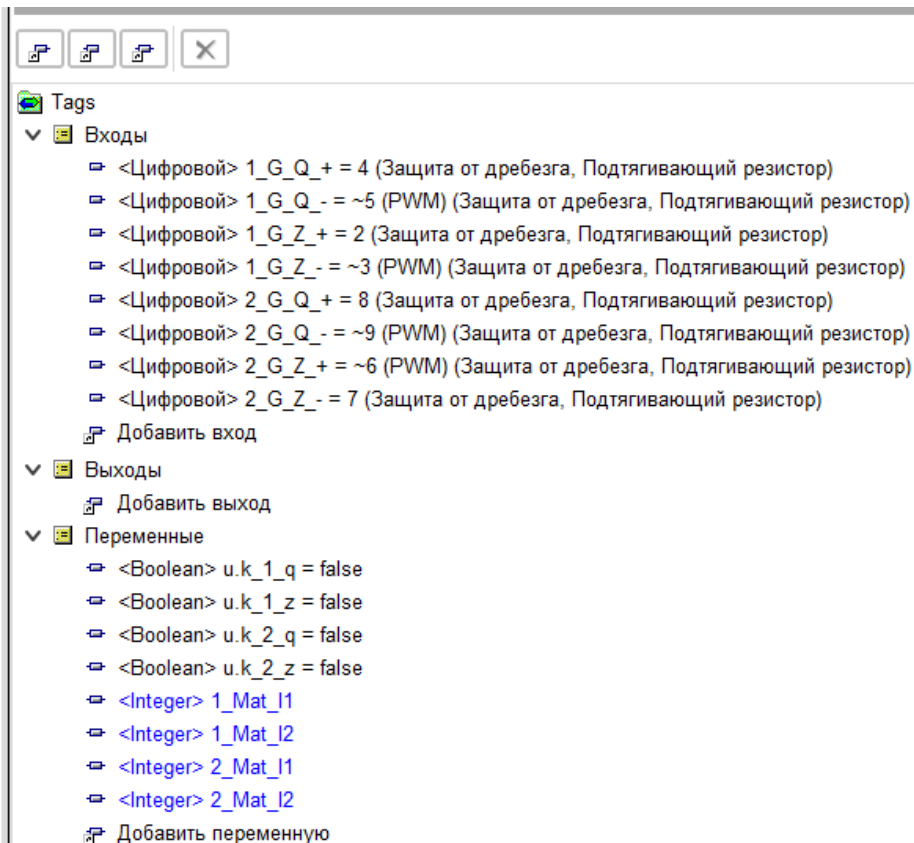


ფიგ. 16.4 რიცხვების გენერატორი_4

როგორც რიცხვების გენერატორების კომპარატორის ბლოკებიდან (B2 და B5) ჩანს, პირველ რიცხვების გენერატორის რიცხვების დიაპაზონი ნებისმიერად აღებული გვაქვს 20-დან 10-მდე, მეორის -40-დან 30-მდე, მესამის -60-დან 50-მდე, მეოთხის კი - 80-დან 70-მდე. თუმცა, პროექტის დამუშავების დროს შეიძლება გამოყენებულ იქნას ნებისმიერი სხვა დიაპაზონები. რიცხვების გენერატორების კორექტულად მუშაობისათვის საჭიროა თვითოეულ მთვლელში B1 საწყისი მომენტში ჩაიწეროს რაიმე შუალედური რიცხვი, ჩვენს შემთხვევაში ვწერთ 15-ს პირველი გენერატორისათვის და 35, 55, 75 - მეორე, მესამე და მეოთხე გენერატორებებისათვის შესაბამისად (ფიგ. 16.5), ანუ ეს რიცხვები ჩვენ გამოყენებული გვაქვს მთვლელებში საწყისი ინფორმაციების ჩასაწერად.

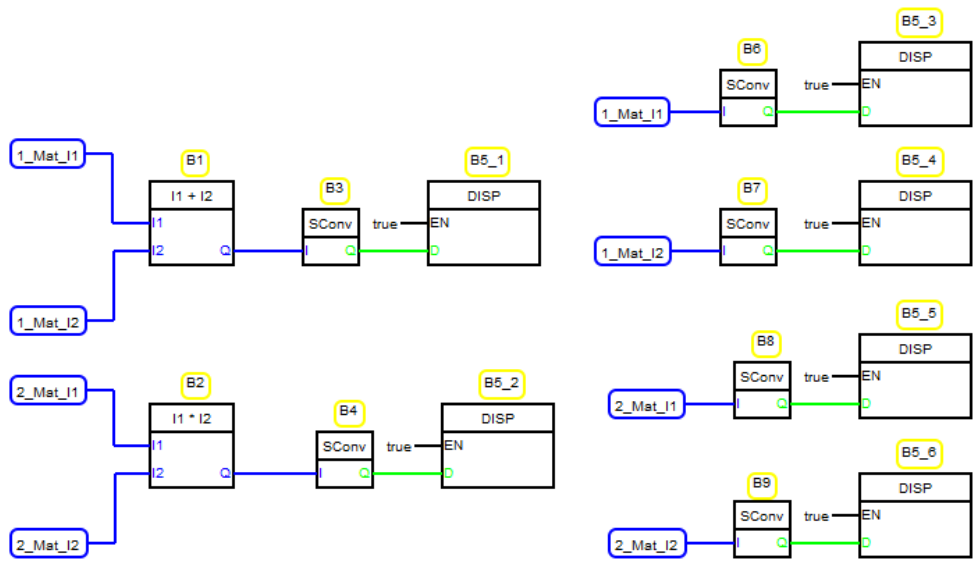


ფიგ. 16.5 საწყისი რიცხვის ჩაწერა რიცხვების გენერატორის მთვლელში.



ფიგ. 16.6 ტეგების ზონის ფორმირება

ახლა ტეგების ზონაში (ფიგ. 16.6). შევქმნათ ციფრული შესასვლელები 1_G_Q_+, 1_G_Q_-, 1_G_Z_+, 1_G_Z_-, 2_G_Q_+, 2_G_Q_-, 2_G_Z_+, 2_G_Z_- თვითოეული გენერატორისათვის კლებისა და მატების გარე დილაკების მისაერთებლად და Integer ტიპის ცვლადები მათემატიკური ბლოკების I1 და I2 შესასვლელებზე გენერირებული რიცხვების გადაცემისათვის, ესენია: 1_Mat_I1, 1_Mat_I2, 2_Mat_I1, 2_Mat_I2. შევუერთოთ ეს ცვლადები რიცხვების გენერატორების მთვლელების B1 ბლოკების გამოსასვლელებს, სადაც ამ რიცხვების ფორმირება ხდება. შევქმნათ აგრეთვე ბულის ცვლადები u.k_1_q, u.k_1_z, u.k_2_q, u.k_2_z რიცხვების გენერატორებში არსებული მთვლელების უკუკავშირების ფორმირებისათვის. ამის შემდეგ შევქმნათ მეხუთე დაფა-Плата: 5 და გადმოვიტანოთ ამ დაფაზე შეკრებისა და გამრავლების მათემატიკური ბლოკები B1 და B2 (ფიგ. 16.7).



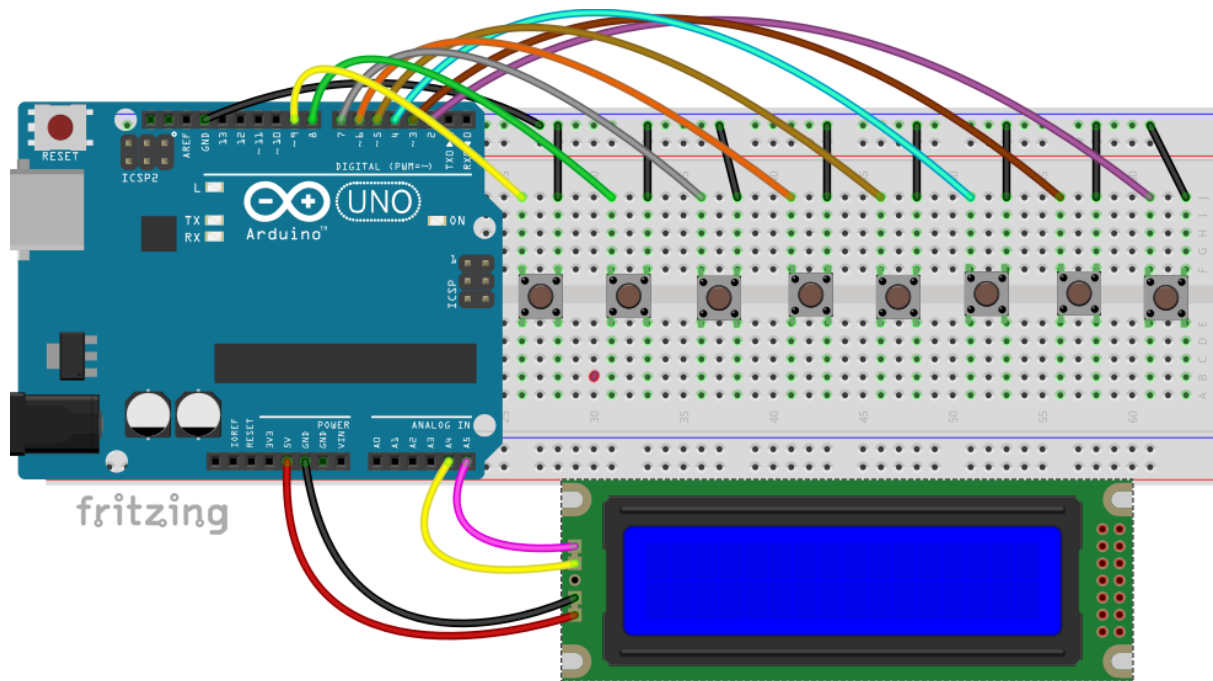
ფიგ. 16.7. მათემატიკური ბლოკებისა და ოპერაციების შედეგების ასახვის დაფა

შეფუერთოთ ამ ბლოკების I1 და I2 შესასვლელებს ჩვენს მიერ წინა დაფებში შექმნილი Integer ტიპის ცვლადები 1_Mat_I1, 1_Mat_I2, 2_Mat_I1, 2_Mat_I2, რაც შესაძლებელია ამ ცვლადების ტეგების ზონიდან მეხუთე დაფაზე გადმოტანით. ამის შემდეგ ამავე დაფაზე უნდა გადმოვიტანოთ კონვერტაციის ბლოკები B3 და B4 ბლოკების ბიბლიოთეკიდან და შეფუერთოთ ისინი ერთის მხრივ მათემატიკური ბლოკების გამოსასვლელებს, ხოლო მეორე მხრივ დისპლეის ჩიპზე HD77480 შესასვლელებს. ჩვენ დაგვჭირდება ორი რიცხვის ასახვა ამ დისპლეიზე პირველი - შერების ბლოკისთვის, მეორე - გამრავლების ბლოკისთვის. ამისათვის საკმარისი იქნება ამ დისპლეის მხოლოდ ერთი, პირველი სტრიქონის გამოყენება, სადაც პირველი სვეტიდან ჩაიწერება B1 ბლოკის ოპერაციის შედეგები, ხოლო მე-9 სვეტიდან კი მეორე ბლოკის B2 ოპერაციის შედეგები (ბლოკები B5_1 და B5_2).

გარდა ამისა, იმისათვის რომ ჩვენ შეგვეძლოს გავაკონტროლოთ არითმეტიკული ბლოკების მუშაობის შედეგების სისწორე, ჩვენ უნდა შეგვეძლოს ცალკე ავსახოთ პირველი და მეორე შესაკრებების, სამრავლისა და მამრავლის მნიშვნელობები. ამისათვის გამოვიყენოთ დისპლეის მე-2 სტრიქონი. ეს სტრიქონი დავყოთ ოთხ ნაწილად B5_3, B5_4, B5_5, B5_6 თვითოეულში ზემოთ მოხსენებული ოპერანდის ასახვისათვის. დისპლეის ეს სტრიქონი დავაპროგრამიროთ გარედან შემოსული რიცხვების ასახვაზე ისე, რომ პირველი რიცხვი დაიწყოს პირველი სვეტიდან, მეორე - მე-5 სვეტიდან, მესამე - მე-9 სვეტიდან და მეოთხე - მე-13 სვეტიდან. დისპლეის სტრიქონების დაპროგრამება ისევე უნდა მოხდეს, როგორც წინა

პროექტში. შევავრთოთ დისპლეის მეორე სტრიქონის შესასვლელები ჩვენს მიერ მე-5 დაფაში უკვე გადმოტანილ Integer ტიპის ცვლადებთან 1_Mat_I1, 1_Mat_I2, 2_Mat_I1, 2_Mat_I2 კონვერტაციის ბლოკების B6...B9 გავლით (ფიგ. 16.7).

გადავტვირთოთ ეს ჩვენი დამუშავებული პროგრამა არდუინოში და შევუდგეთ პროგრამის ტესტირებას. ვცვალოთ შესასვლელი რიცხვები შესაბამისი ღილაკების გამოყენებით მატებაზე თუ კლებაზე, წავიკითხოთ ეს რიცხვები დისპლეის მეორე სტრიქონში და შევამოწმოთ მათხედ შესრულებული შეკრებისა თუ გამრავლების ოპერაციების შედეგები დისპლეის პირველ სტრიქონში. მოწყობილობის Fritzing სქემა იხილეთ ფიგურა 16.8-ზე.



ფიგ. 16.8 მათემატიკური ოპერაციების მართვის Fritzing სქემა.

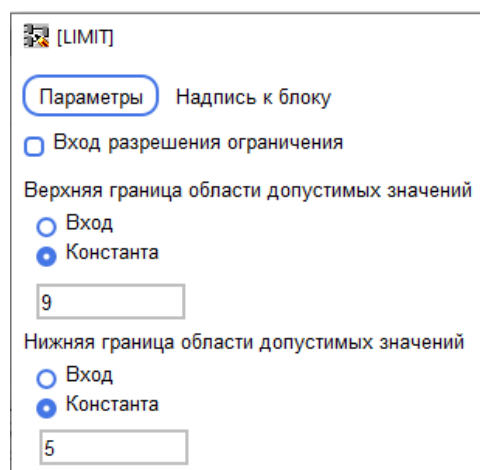
პროექტი_17

ალგებრა_1



FLProg-ის ალგებრის ნაწილში ჩვენ დაგვჭირდება 12 ფუნქციის მართვა. ერთ პროექტში ამ რაოდენობის ფუნქციის განხილვა მნიშვნელოვნად გაართულებდა პროექტს, ამიტომ გადაწყვიტეთ იგი გაგვეყო ორ ნაწილად. პირველ ნაწილში წარმოდგენილი იქნება 6 ფუნქცია, ანუ ბლოკი: [Limit], [POW], [ABS], [SQRT], [RAND] და [MAX].

ა. მნიშვნელობების შემოფარგვლის [Limit] ბლოკის დანიშნულებაა თავის შესასვლელზე “Value” მოდებული მნიშვნელობების შემოფარგვლა საზღვრებით, რომლებსაც ჩვენ დავუდგენთ ბლოკის აწყობებში, ანდა მივაწვდით “Max” და “Min” შესასვლელებზე. თუ “Value” შესასვლელზე მოდებული მნიშვნელობა გამოვა ამ საზღვრებიდან, მაშინ “Q” გამოსასვლელზე გამომავალი მნიშვნელობა შესაბამისი იქნება იმ მნიშვნელობის, რომელიც გადაკვეთა სიგნალმა. წინააღმდეგ შემთხვევაში გამოსასვლელზე იქნება შესასვლელზე არსებული სიგნალის მნიშვნელობა. ბლოკის გაწყობა უნდა მოხდეს რედაქტორის ბლოკის გამოყენებით. ჩვენს შემთხვევაში გამოვიყენებთ საზღვრების დადგენის პირველ გზას და დავუდგენთ მას ზედა ზღვარს ტოლს 9, ხოლო ქვედა ზღვარს ტოლს 5 (ფიგ. 17.1).



ფიგ. 17.1 “Limit” ბლოკის ზღვრების დადგენა

ბ. ხარისხში აყვანის ბლოკი “POW” ახდენს მოცემულ ხარისხში აყვანილი მნიშვნელობის გაანგარიშებას. ეს ხარისხი შეიძლება იყოს წილადურიც.

ბლოკის შესასვლელია:

Base-რიცხვი, რომელიც საჭიროა იყოს აყვანილი ხარისხში;

Exp-ხარისხი, რომელშიც უნდა იყოს აყვანილი რიცხვი;

ბლოკის გამოსასვლელი:

ხარისხში აყვანის რეზულტატი.

გ. რიცხვის მოდულის ბლოკი [ABS] აბრუნებს რიცხვის მოდულს.

შესასვლელი:

I - რიცხვი

გამოსასვლელი:

I თუ I მეტია ან ტოლია ნოლზე, ან-I თუ I ნაკლებია ნოლზე.

დ. რიცხვის კვადრატული ფესვის ბლოკი [SQRT] ახდენს რიცხვის კვადრატული ფესვის გაანგარიშებას.

შესასვლელი:

ნებისმიერი ტიპის რიცხვი.

გამოსასვლელი:

რიცხვის კვადრატული ფესვი.

ე. შემთხვევითი რიცხვის ბლოკი [RAND] გამოსცევს ფსევდოშემთხვევით რიცხვს.

შესასვლელი:

EN-შემთხვევითი რიცხვის გენერაციის ნების დართვის შესასვლელი. ამ შესასვლელზე მაღალი დონის მიწოდების შემთხვევაში, პროგრამის თვითოეულ ციკლში გენერირდება შემთხვევითი რიცხვი, დაბალი დონის შემთხვევაში ნარჩუნდება ბოლო გენერირებული რიცხვი;

MIN - შემთხვევითი მნიშვნელობების ქვედა ზღვარი;

MAX - შემთხვევითი მნიშვნელობების ზედა ზღვარი.

გამოსასვლელი:

შემთხვევითი რიცხვი MIN და (MAX – 1) შორის.

ვ. დიდი რიცხვის არჩევის ბლოკი [MAX] დანიშნულებაა ორი რიცხვიდან უფრო დიდის დაბრუნება.

შესასვლელი:

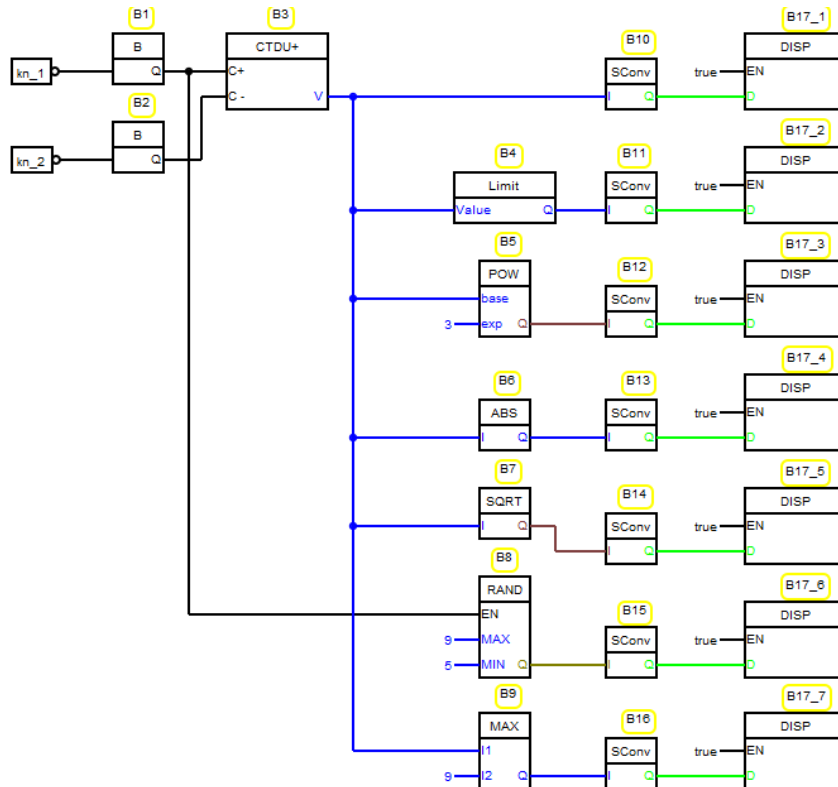
I1 – პირველი რიცხვი;

I2 - მეორე რიცხვი.

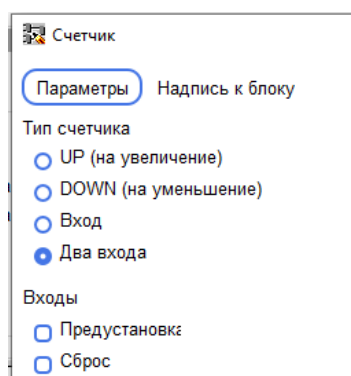
გამოსასვლელი:

ორ შესადარებელ რიცხვს შორის უფრო დიდი.

ჩამოთვლილი ალგებრული ბლოკების მუშაობის სამართავად გამოყენებული გვაქვს რევერსიული მთვლელი B3 (ფიგ. 17.2) რომელსაც შეუძლია მუშაობა რევერსიული თვლის რეჟიმში. მთვლელის პარამეტრიზაციის ფანჯარა მოცემულია ფიგ. 17.3-ზე.

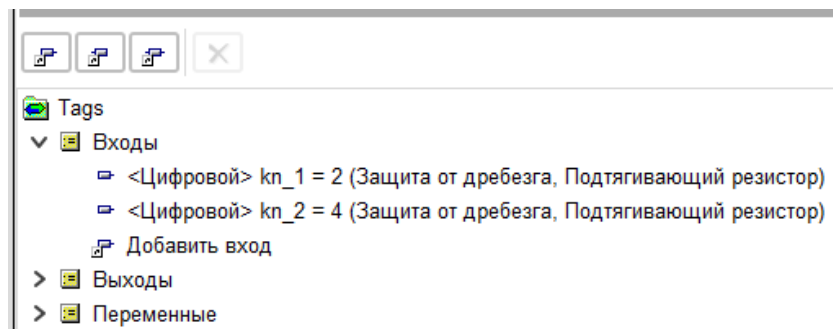


ფიგ. 17.2 ალგებრული ბლოკების [Limit], [POW], [ABS], [SQRT], [RAND], [MAX] მართვის პროგრამა



ფიგ. 17.3 მთვლელის პარამეტრიზაციის ფანჯარა

ამ ფანჯარაში ერთადერთი რაც პარამეტრებშია ჩართული არის ორი შესასვლელის ჩამრთველი-*Два входа* ეს ნიშნავს იმას, რომ მთვლელი მუშაობს ორი შესასვლელის რეჟიმში, ერთი-შეკრების რეჟიმი, მეორე-გამოკლების რეჟიმი. მთვლელის მართვისათვის გამოიყენება ორი ღილაკი *kn_1* და *kn_2*, რომლებიც შეერთებული გვაქვს არდუინოს მე-2 და მე-4 კონტაქტებთან, თუმცა შეგვეძლო შეგვეერთებინა ნებისმიერ სხვა თავისუფალ კონტაქტთან. ფიგ. 17.4-ზე ასახულია ტეგების ზონა ამ პროექტისათვის, სადაც *kn_1* და *kn_2* ღილაკებზე პროგრამულად ჩართულნი არიან ყანყალისაგან დაცვის რეჟიმი და ამომქაჩავი რეზისტორი.



ფიგ. 17.4 ტეგების ზონის ფანჯარა

ალგებრული ბლოკების მუშაობის მართვის ალგორითმი ამ პროექტისათვის მდგომარეობს იმაში, რომ მათ შესასვლელებს მეწოდებათ რევერსულო მთვლელის გამოსასვლელი სიგნალები და დისპლეიზე ხდება მათზე ამ ბლოკების რეაგირების, ანუ შედეგების ასახვა. დისპლეიდ გამოყენებული გვაქვს ჩვენთვის უკვე კარგად ცნობილი ორსტრიქონიანი თექვსმეტპოზიციანი დისპლეი ჩიპზე HD44780. აქ მართვის პრობლემა მდგომარეობს იმაში, რომ ყველა ბლოკის მუშაობის შედეგი ერთდროულად უნდა აისახოს დისპლეის სტრიქონებში.

ამისათვის ამ ბლოკებს შორის დისპლეის სტრიქონები და სვეტები განაწილებულია შემდეგნაირად, თვითონ მთვლელის შემცველობა აისახება პირველი სტრიქონის პირველი სვეტიდან, [Limit] ბლოკის მუშაობის შედეგი აისახება პირველი სტრიქონის მე-4 სვეტიდან, [POW] ბლოკის შედეგი აისახება პირველი სტრიქონის მე-6 სვეტიდან, [ABS] ბლოკის შედეგი-პირველი სტრიქონის მე-15 სვეტიდან, [SQRT] ბლოკის შედეგი - მე-2 სტრიქონის პირველი სვეტიდან, [RAND] ბლოკის შედეგი - მე-2 სტრიქონის მე-6 სვეტიდან და [MAX] ბლოკის შედეგი - მე-2 სტრიქონის მე-8 სვეტიდან მე-16 სვეტის ჩათვლით. აღნიშნული გრადაციები ნათლად ჩანს B17_1...B17_7 ქვებლოკების პარამეტრიზაციის ფანჯრებიდან. ფიგ. 17.5-ზე წარმოდგენილია ერთერთი ბლოკის, მაგ. [ABS] ბლოკს-B6 პარამეტრიზაციის ფანჯარა.

ფიგ. 17.5 [ABS] ბლოკის პარამეტრიზაციის ფანჯარა

ახლა შევუდგეთ ალგებრული ბლოკების მუშაობის შემოწმებას. ამისათვის ჯერ გამოვიყენოთ kn_1 ლილაკი მოვახდინოთ მასზედ მრავალჯერადი დაჭერა, მთვლელის მდგომარეობას დავაკვირდეთ პირველი სტრიქონის პირველ და მეორე სვეტებში, ავიდეთ მაგ. 15-მდე. შემდეგ გადავიდეთ kn_2 ლილაკის მრავალჯერ დაჭერაზე და ვაკონტროლოთ მთვლელის მუშაობა, მისი მდგომარეობა უნდა იცვლებოდეს კლებაზე, უნდა ჩამოვიდეს ნოლამდე და ამ ლილაკზე თითის დაჭერის განგრძობით, მთვლელმა უნდა დააფიქსიროს მინუს ერთი, მინუს ორი, მინუს სამი და ა.შ. ნებისმიერ რიცხვამდე.

ახლა kn_1 ლილაკის დაჭერით ისევ მივიდეთ ამ მთვლელის ნოლოვან მდებარეობამდე და ამის შემდეგ დავიწყოთ შემოწმება ჩვენს მიერ ამ პროექტში წარმოდგენილი ბლოკების.

ა. როგორც ფიგ. 17.1-ზე იყო წარმოდგენილი B4 [Limit] ბლოკის სალომიტო ზღვრები იყო 9-ზედა ზღვარი და 5-ქვედა ზღვარი, დავიწყოთ kn_1 ლილაკზე მრავალჯერადი დაჭერით, ვაკონტროლოთ მთვლელის გამოსასვლელი პირველი სტრიქონის პირველი სვეტიდან და B4 ბლოკის გამოსასვლელი პირველი სტრიქონის მე-4 სვეტიდან, მანამ მთვლელი არ ავა 5-მდე B4 ბლოკის გამოსასვლელი იქნება ყოველთვის 5. ამის შემდეგ B4 ბლოკი გაიმეორებს მთვლელის მაჩვენებელს: 6, 7, 8, 9, ამის შემდეგ კი მთვლელი კი გააგრძელებს ზემოთ თვლას, მაგრამ B4 ბლოკის გამოსასვლელზე იქნება მუდამ ცხრა. ისევ დავაკლოთ მთვლელს და ჩამოვცდეთ ხუთს, გადავიდეთ მინუსებში და დავრწმუნდეთ, რომ B4 ბლოკის გამოსასვლელი რჩება მუდმივად ხუთიანის მდგომარეობაში.

ბ. B5 ბლოკის - [POW] შემოწმებისთვის ვიწყებთ მთვლელის ნულოვანი მდგომარეობიდან. როგორც პროექტიდან ჩანს, ჩვენ ბაზად აღებული გვაქვს მთვლელის მნიშვნელობა, ხოლო ხარისხად უცვლელი რიცხვი 3. მთვლელის ნოლოვანი მდგომარეობის დროს დისპლეის მე-6, მე-7, მე-8 და მე-9 პოზიციებში იქნება რიცხვი 0.00, ამის შემდეგ თუ მთვლელში ჩავწერთ ერთიანს, მაშინ ერთი ხარისხად სამი იქნება ერთი და ზემოთ მითითებული პოზიციები მიიღებს მნიშვნელობებს 1.00, თუ მთვლელში ჩავწერთ 2-ს, მაშინ ორი ხარისხად სამი იქნება რვა და ზემოთმოხსენებული პოზიციები მიიღებს სახეს 8.00, მთვლელის სამის დროს უკვე მე-10 პოზიციაც დაემატება, ანუ რეზულტატი გამონათდება მე-6, მე-7, მე-8, მე-9 და მე-10 პოზიციებში და შედეგი იქნება 27.00, მთვლელის ოთხიანის დროს - 64.00, 5-იანის დროს -125.00, 6-იანის დროს-216.00, 7-იანის დროს 343.00 და შემდეგ 20-ანის დროს 8000.00 და ა.შ.შ. მთვლელის მნიშვნელობის 46-ის დაფიქსირების შემდეგ რეზულტატი ბლოკის მუშაობის რეზულტატი იქნება 97335.91 და უფრო დიდი რიცხვები ამ პროექტში ვეღარ დაფიქსირდება დისპლეის შესაძლებლობების გამო.

გ. B6 ბლოკის - [ABS] შემოწმებისთვის ვიწყებთ მთვლელის ნულოვანი მდგომარეობიდან. ამ ბლოკის მუშაობის რეზულტატის ასახვისათვის გამოიყენება პირველი სტრიქონის მე-15 და მე-16 სვეტები. ვიწყებთ მთვლელში თითო-თითო იმპულსის დამატებას kn_1 ღილაკის გამოყენებით და ვაკვირდებით ამავე სტრიქონის პირველ და მეორე სვეტებს. ვღებულობთ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11... იგივე რიცხვები უნდა მოგვცეს [ABS] ბლოკმა მე-15 და მე-16 სვეტებში. ახლა გადავდივართ kn_2 ღილაკზე თითის დაჭერაზე, ჩამოვალთ მთვლელის ნულოვან მდგომარეობამდე და ვაგრძელებთ თითის დაჭერას. პირველ და მეორე სვეტებში ვღებულობთ -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11 მაგრამ მე-15 და მე-16 სვეტებში ისევ ვიღებთ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 რიცხვებს, ანუ რიცხვების აბსოლუტურ მნიშვნელობებს.

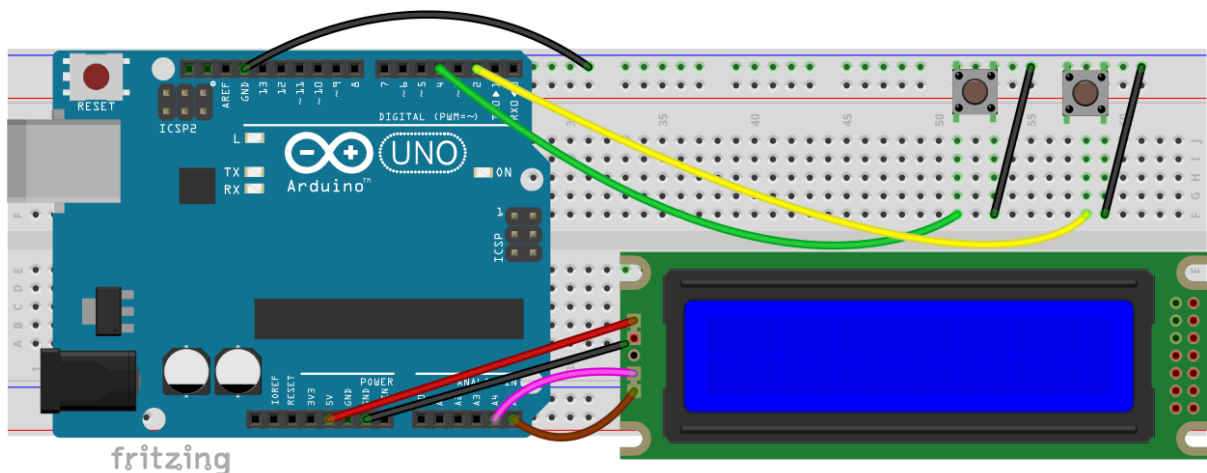
დ. კვადრატული ფესვის ამოღების [SQRT] ბლოკის B7 მუშაობის შედეგი ფიქსირდება მე-2 სტრიქონის პირველი სვეტიდან და მისთვის ოთხი სვეტია გამოყოფილი. საწყის მომენტში როდესაც მთვლელში ნოლია ჩაწერილი ბლოკი იძლევა შედეგს 0.00, როდესაც ერთს ჩავწერთ მაშინ რეზულტატი იქნება 1.00, 2-ის დროს 1.41, 3-ის დროს - 1.73, 4-ის დროს 2.00 და ა.შ.შ 40-ის დროს 6.32 და ა.შ. 98-ის დროს-9.90, მეტის შესაძლებლობა ამ პროექტს უკვე აღარ აქვს.

ე. შემთხვევითი სიდიდეების გენერირების ბლოკის B8 შემოწმებისათვის ზედა და ქვედა ზღვარი აქაც 9 და 5 გვაქვს დაფიქსირებული, მაგრამ შეიძლება სხვა ნებისმიერი რიცხვების ჩაწერა. ამ რიცხვების ჩაწერა უნდა მოხდეს თვითონ ბლოკის გამომყვანებზე სტანდარტული მეთოდით, კონტექსტურ მენიუში შესვლით, კონსტანტის ჩასმის არჩევით, რიცხვის ტიპის არჩევით და შესაბამის ველში საჭირო რიცხვის ჩაწერით. ამ ბლოკის

შესასვლელზე, სხვა ბლოკებისაგან განსხვავებით უნდა იქნას მიწოდებული არა რიცხვი, არამედ სიგნალი. ამისთვის ჩვენ ვიყენებთ kn_1 დილაკის იმპულსს. ამ დილაკზე თითის დაჭერის დროს ხდება შემთხვევითი რიცხვების უწყვეტი გენერირება 5-დან 8-მდე დიაპაზონში. თითის ადების შემდეგ ბლოკის გამოსასვლელზე რჩება ბოლო გენერირებული რიცხვი. ამ რიცვის ასახვისათვის გამოყოფილ გვაქვს ერთი სვეტი, კერძოდ მე-6. kn_1 დილაკზე თითის დაჭერის დროს თვალნათლივ ჩანს შემთხვევითი რიცხვების გენერაცია და ასახვა, თითის აშვების შემდეგ გამონათდება ნებისმიერი ერთერთი შემთხვევითი რიცხვი 5, 6, 7, 8.

ვ. B9 ბლოკი შედარებას ახდენს მთვლელში ჩაწერილი რიცხვის, რომელიც მიეწოდება ამ ბლოკის I1 შესასვლელს და I2-ზე ჩაწერილ რიცხვს შორის და მათ შორის მაქსიმალურ რიცხვს გამოსცევს ბლოკის გამოსასვლელზე. მისი მუშაობის რეზულტატი აისახება მეორე სტრიქონის მე-8 სვეტიდან. ბლოკის შემოწმებას ვიწყებთ მთვლელის ნულთან პოზიციიდან. ამ შემთხვევაში მე-8 სვეტიდან მე-16 სვეტის ჩათვლით აისახება 9.000000, ამის შემდეგ ვამატებთ მთვლელს თითო იმპულსს, მაგრამ რეზულტატი ცხრა იმპულსამდე იგივეა 9.000000, მთვლელში 10-ის ჩაწრის შემთხვევაში დაწყებული მე-18 სვეტიდან, დამთავრებული მე-16-ს ჩათვლით-ჩაიწერება 10.000000 მე-11 იმპულსზე-11.000000 და ა.შ. 99-ზე-99.000000, მე-100 იმპულსზე იქნება - 100.00000 და ა.შ.

პროექტის Fritzing სქემა ასახულია ფიგ. 17.6-ზე



ფიგ. 17.6 ალგებრული ბლოკების მართვის Fritzing სქემა

პროექტი_18

ალგებრა_2



ამ პროექტში განხილული იქნება შემდეგი სახის ალგებრული ბლოკები: [TAN], [SIN], [COS], [modulo], [SQ] და [MIN].

ა. [TAN]-ის ბლოკი აბრუნებს კუთხის ტანგენსს, რომელიც მოცემულია რადიანებში. შესასვლელი:

I – კუთხე რადიანებში.

გამოსასვლელი:

Q – კუთხის ტანგენსი.

ბ. [SIN]-ის ბლოკი აბრუნებს კუთხის სინუსს, რომელიც მოცემულია რადიანებში, ბლოკის რეზულტატი დევს $-1 \dots +1$ დიაპაზონში.

შესასვლელი:

I – კუთხე რადიანებში.

გამოსასვლელი:

Q – კუთხის სინუსი.

გ. [COS]-ის ბლოკი აბრუნებს კუთხის კოსინუსს, რომელიც მოცემულია რადიანებში, ბლოკის რეზულტატი დევს $-1 \dots +1$ დიაპაზონში.

შესასვლელი:

I – კუთხე რადიანებში.

გამოსასვლელი:

Q – კუთხის კოსინუსი.

დ. ნაშთი გაყოფისაგან (modulo).

გაყოფის ოპერაცია მოდულით გამოიყენება მთელრიცხვა არითმეტიკაში. მისი რეზულტატია ნაშთი, რომელიც მიიღება ოპერაციის მარცხნივ მდებარე რიცხვის გაყოფით ოპერაციის მარჯვნივ მდებარე რიცხვზე. მაგ. $13 \% 5$ (წაიკითება როგორც 13 მოდულით 5) გვაძლევს მნიშვნელობას 3, რადგან სამართლიანია ტოლობა $13 = 2*5+3$.

ბლოკს არ აქვს პარამეტრები.

შესასვლელები:

I1 – გასაყოფი მთელი რიცხვი;

I2 - მოდული.

გამოსასვლელი:

Q - ნაშთი მთელი რიცხვის გაყოფისა მოდულზე.

ე. [SQ] - კვადრატში აყვანის ბლოკი ასრულებს რიცხვის კვადრატში აყვანის ოპერაციას.

შესასვლელი:

I – ნებისმიერი ტიპის რიცხვი.

გამოსასვლელი:

Q – რიცხვის კვადრატი.

ვ. [MIN] ბლოკი აბრუნებს უმცირესს ორი რიცხვისგან.

შესასვლელები:

I1 – პირველი რიცხვი;

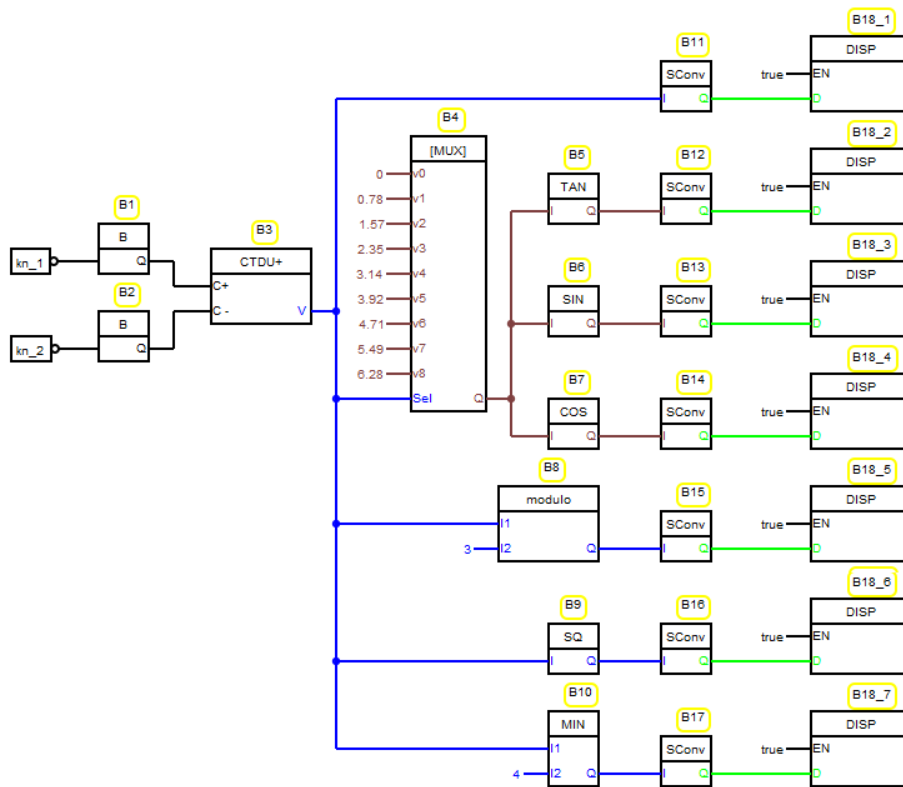
I2 - მეორე რიცხვი.

გამოსასვლელი:

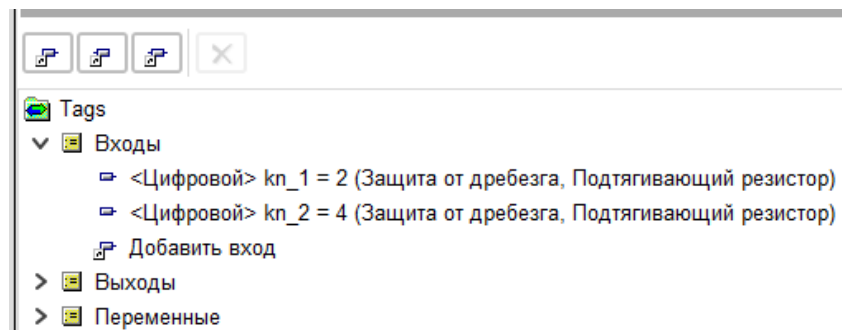
უფრო მცირე რიცხვი ორი შესადარებელი რიცხვისგან.

ალგებრული ბლოკების [TAN], [SIN], [COS], [modulo], [SQ] და [MIN] მართვის პროგრამა წარმოდგენილია ფიგ. 18.1-ზე.

ამ ბლოკებზე ტესტური ზემოქმედების მიწოდებისათვის გამოყენებული გვაქვს რევერსული მთვლელი B3, რომლის C+ და C- შესასვლელებზე მიერთებულია kn_1 და Kn_2 დილაკები ხელშემლელის იმპულსების ჩახშობის B1 და B2 ბოუნცე ბლოკების გავლით. თვითონ kn_1 და Kn_2 დილაკები მიერთებულნი არიან არდუინოს მე-2 და მე-4 კონტაქტებზე, არდუინოს ტეგების ზონაში შექმნილი შესასვლელების გამოყენებით (ფიგ. 18.2). ჩვენი ამოცანაა ტესტური ზემოქმედებების მიზნით გამოვიკვლიოთ თუ როგორ რეაგირებს ჩვენი ალგებრული ბლოკები როდესაც მთვლელი ითვლის პლიუსებში და მინუსებში, ზრდადობაზე და კლებადობაზე და ავსახოთ მათი რეაგირების შედეგები ორსტრიქონიან 16 სვეტიან დისპლეიზე.

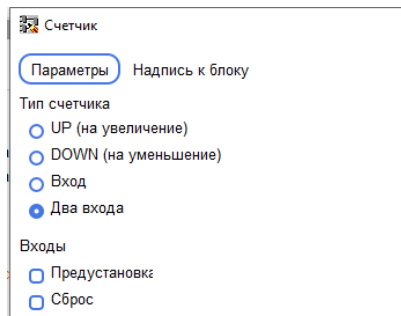


ფიგ. 18.1. ალგებრული ბლოკების [TAN], [SIN], [COS], [modulo], [SQ] და [MIN] მართვის პროგრამა



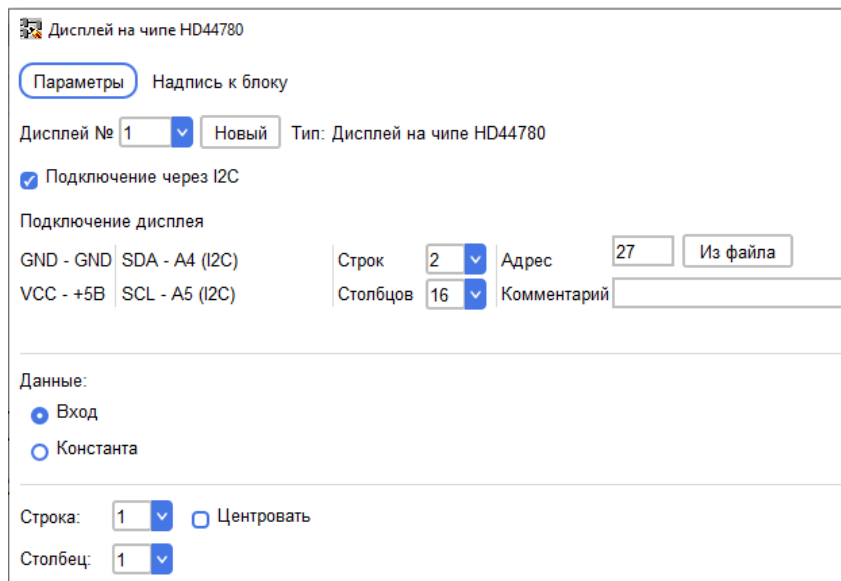
ფიგ. 18.2. ტეგების ზონა

იმისათვის, რომ მთვლელმა შეძლოს მუშაობა როგორც მატებაზე, ასევე კლებაზე და აგრეთვე მინუსებშიც, საჭიროა მისი პარამეტრიზაცია ფიგ. 18.3-ზე ნაჩვენები ფანჯრის მიხედვით.



ფიგ. 18.3 მთვლელის ბლოკის პარამეტრიზაციის ფანჯარა

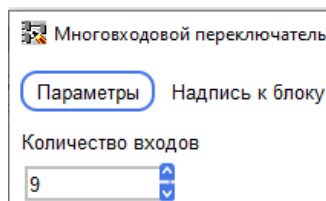
აღგებრული ბლოკების მართვის პროცესში აუცილებელია ჩვენ ხელთ გვექონდეს ინფორმაცია ამ მთვლელის მდგომარეობის შესახებ. ეს ინფორმაცია სასურველია აისახოს დისპლეის პირველი სტრიქონის პირველ და მეორე სვეტში. ამ მიზნით მთვლელის ბლოკის B3-ს გამოსასვლელი B11 კონვერტაციის ბლოკის გავლით შეერთებულია დისპლეის B18 ბლოკის პირველნაწილთან B18_1, რომელიც დაპროგრამებულია ინფორმაციის ასახვაზე პირველი სტრიქონის პირველ სვეტიდან (ფიგ. 18.4).



ფიგ. 18.4. მთვლელის მდგომარეობის ასახვა პირველი სტრიქონის პირველი სვეტიდან

ახლა გადავიდეთ ტრიგონომეტრიული ბლოკების მუშაობის შემოწმებაზე და მათი მუშაობის რეზულტატების ასახვაზე დისპლეიზე. ახლავე უნდა ითქვას, რომ მათი მუშაობის შედეგები B1... B14 კონვერტაციის ბლოკების გავლით მიეწოდება B18_2...B18_4 დისპლეის ნაწილებზე და აისახება პირველი სტრიქონის მეოთხე სვეტიდან, პირველი სტრიქონის მე-12 სვეტიდან და მეორე სტრიქონის პირველი სვეტიდან შესაბამისად.

ვინაიდან ეს ტრიგონომეტრიული ბლოკები მუშაობენ კუთხის რადიანულ ზომებზე, ამიტომ ჩვენ გვჭირდება კუთხეების რადიანებში გენერირება. ამ მიზნით შემოტანილი გვაქვს მულტიპლექსორული გადამრთველი ბლოკი [MUX]–B4, რომლის ცხრა შესასვლელზე მიწოდებული გვაქვს ცხრა კუთხე რადიანებში: 0, 0.78, 1.57, 2.35, 3.14, 3.92, 4.71, 5.49, 6.28. [MUX] ბლოკი B4 მუშაობს შემდეგნაირად, თუ sel შესასვლელზე იცვლება ანალოგური ინფორმაცია: 0, 1, 2, 3, 4, 5, 6, 7, 8, მაშინ მის Q გამოსასვლელზე გამოვა შესაბამისი ანალოგური სიგნალები v0, v1, v2, v3, v4, v5, v6, v7, v8 შესასვლელებიდან. თვითონ [MUX] ბლოკის პარამეტრზაცია, ანუ შესასვლელთა რაოდენობის არჩევა ხდება ამ ბლოკზე დაწკაპუნებითა და გამოსულ ფანჯარაში გამოსასვლელების რაოდენობის ჩასწერით (ფიგ. 18.5). დისპლეის შეერთება კი არდუინოსთან უნდა მოხდეს სტანდარტული შეერთებების გამოყენებით.



ფიგ. 18.5 [MUX] ბლოკის პარამეტრზაციის ფანჯარა

ავაწყობთ სქემა წარმოდგენილი პროგრამის მიხედვით და ჩავტვირთოთ ეს პროგრამა არდუინოში და გავუშვათ. საწყის მდგომარეობაში მთვლელი დადგება ნულოვან მდგომარეობაში. მე-4 სვეტი ტანგენსის ბლოკიც [TAN] გვიჩვენებს 0-ს.

ა. [TAN] ბლოკის შემოწმებისთვის დავაჭიროთ kn_1 ღილაკს თითო ერთჯერ, რითაც მთვლელში ჩაიწერება ერთიანი, რაც აისახება პირველი სტრიქონის პირველ სვეტში, იგივე სტრიქონის მე-4,5 სვეტებში აისახება ტანგენსის მნიშვნელობა 0.78 რადიან კუთხეზე, ანუ 0.99. დავაჭიროთ kn_1 ღილაკს მეორეჯერ, ტანგენსის მნიშვნელობა 1.57 რადიან კუთხეზე აისახება მე-4,5,6,7,8,9,10 სვეტებში და იქნება 1255.85, დავაჭიროთ მესამეჯერ kn_1 ღილაკს, ტანგენსის მნიშვნელობა მე-4,5,6,7,8 სვეტებში 2.35 რადიან კუთხეზე ტოლი იქნება -1.01. კიდევ ერთჯერ დავაჭიროთ და მე-4,5,6,7,8 სვეტებში მივიღებთ ინფორმაციას -0.00.

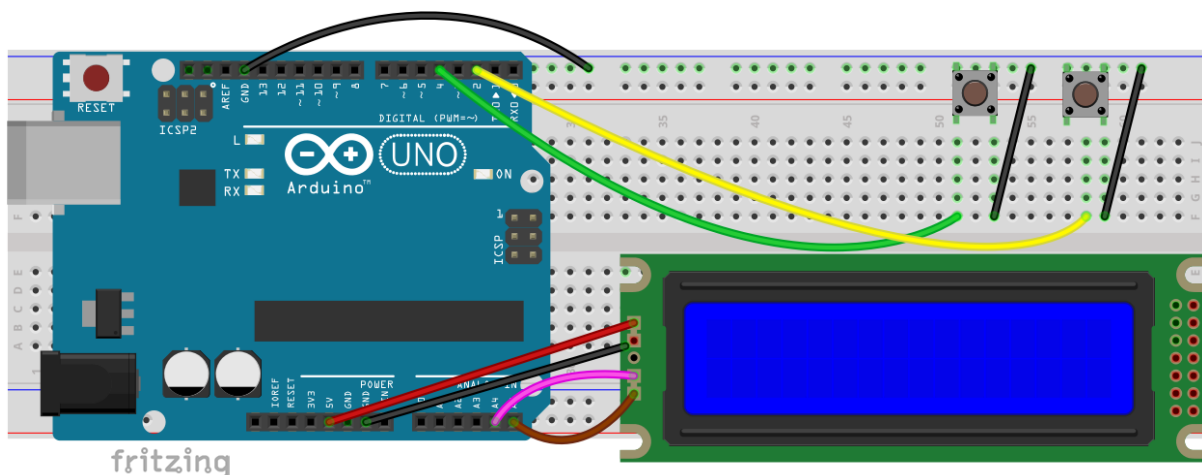
ბ. [SIN] ბლოკის B6 შემოწმება. მთვლელი როცა იმყოფება ნულოვან მდგომარეობაში, მაშინ პირველი სტრიქონის მე-12,13,14,15 სვეტებში ჩაწერილია 0.00, როცა მთვლელში ჩავწერთ 1-ს მაშინ იმავე სვეტებში ჩაიწერება 0.70 და ა.შ. მთვლელის მნიშვნელობის გაზრდით ამ სვეტებში გამოვა: 1.00, 0.71, 0.00, -0.70, -1.00, -0.71, -0.00.

გ. [COS] ბლოკის B7 შემოწმება. მთვლელი როცა იმყოფება ნულოვან მდგომარეობაში, მაშინ მეორე სტრიქონის 1,2,3,4 სვეტებში ჩაწერილი იქნება 1.00, მთვლელში თითო იმპულსების მიმტებით ამავე სვეტებში მივიღებთ 0.71, 0.00, -0.70, -1.00, -0.71, -0.00, 0.70, 1.00.

დ. [modulo] ბლოკის B8 მუშაობის რეზულტატი გამოდის მეორე სტრიქონის მე-7 სვეტში, საწყის მდგომარეობაში, როდესაც მთვლელი ნულოვან მდგომარეობაშია, ამ ბლოკის რეზულტატი არის 0, ერთიანის შემთხვევაში - ერთი, ორიანის შემთხვევაში - ორი, სამიანის შემთხვევაში - ნოლი, რადგან ჩვენ I2-ზე ჩაწერილი გვაქვს 3, ოთხიანის შემთხვევაში - ისევ ერთი და ა.შ.შ.

ე. [SQ] ბლოკის B8 მუშაობის რეზულტატი გამოდის მეორე სტრიქონის მე-9,10,11,12,13 სვეტებში და აისახება მთვლელში დაფიქსირებული რიცხვის კვადრატი. ამის შესაბამისად ამ სვეტებში ასახული ბლოკის მუშაობის რეზულტატები იქნება 0.000, შემდეგ 1.000, 4.000, 9.000, 16.00, 25.00, 36.00, 49.00, 64.00, 81.00, 100.0 და ა.შ.შ.

ვ. [MIN] ბლოკის B9 მუშაობის რეზულტატი გამოდის მეორე სტრიქონის მე-15 და 16 სვეტებში, თუ მთვლელის რიცხვი = 0, მე-15, 16 სვეტებში ასახული იქნება 0. ერთის შემთხვევაში -1., ორის შემთხვევაში - 2.0, სამის შემთხვევაში - 3.0, ოთხის შემთხვევაში - 4.0, ხუთის შემთხვევაში - 4.0, ექვსის შემთხვევაში ისევ 4.0 და ა.შ.შ. თუ მთვლელის მინუსებში წავალთ, მაშინ აისახება: -1, -2, -3, -4, -5, -6, -7 და ა.შ.შ. პროექტის Fritzing სქემა ასახულია ფიგ. 18.6-ზე.



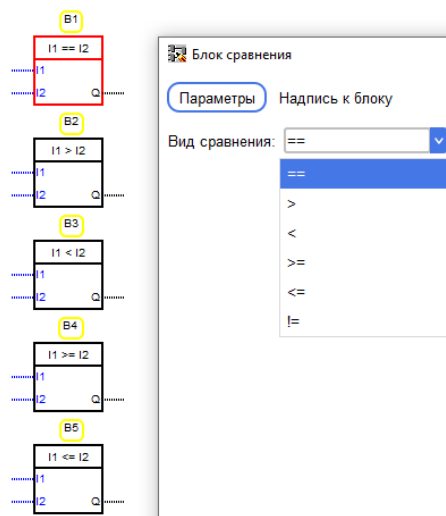
ფიგ. 18.6 ალგებრული ბლოკების მართვის Fritzing სქემა

პროექტი_19

შედარებები

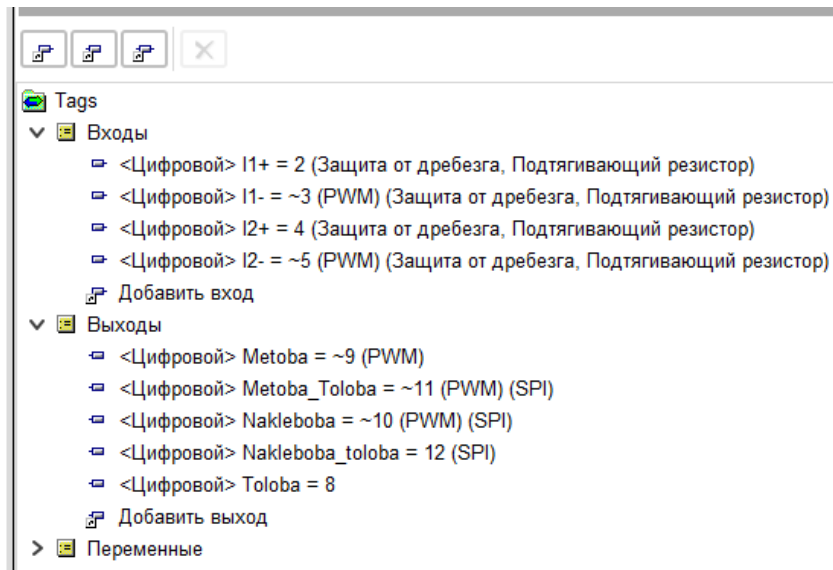


შევიდეთ FLProg-ის ბიბლიოთეკაში და შედარების საქალაქედან-Сравнение პროგრამის სამუშაო ზონაში ზონაში გადმოვიტანოთ შედარების ბლოკი-Comparator და ეს გადმოტანა გავიმეოროთ ხუთჯერ B1...B5 (ფიგ. 19.1). დავაწკაპუნოთ თვითოეულ ბლოკზე თავის მარცხენა ღილაკით ორჯერ სათითაოდ და გამოსული პარამეტრების ფანჯარაში შედარების ტიპის ველში-Вид сравнения თვითოეული ბლოკისთვის ავირჩიოთ განსხვავებული ტიპი, კერძოდ B1 ბლოკისთვის-ტოლობა ==, B2 ბლოკისთვის-მეტობა >, B3 ბლოკისთვის- ნაკლებობა <, B4 ბლოკისთვის-მეტობა ან ტოლობა >= და B5 ბლოკისთვის- ნაკლებობა ან ტოლობა <=.



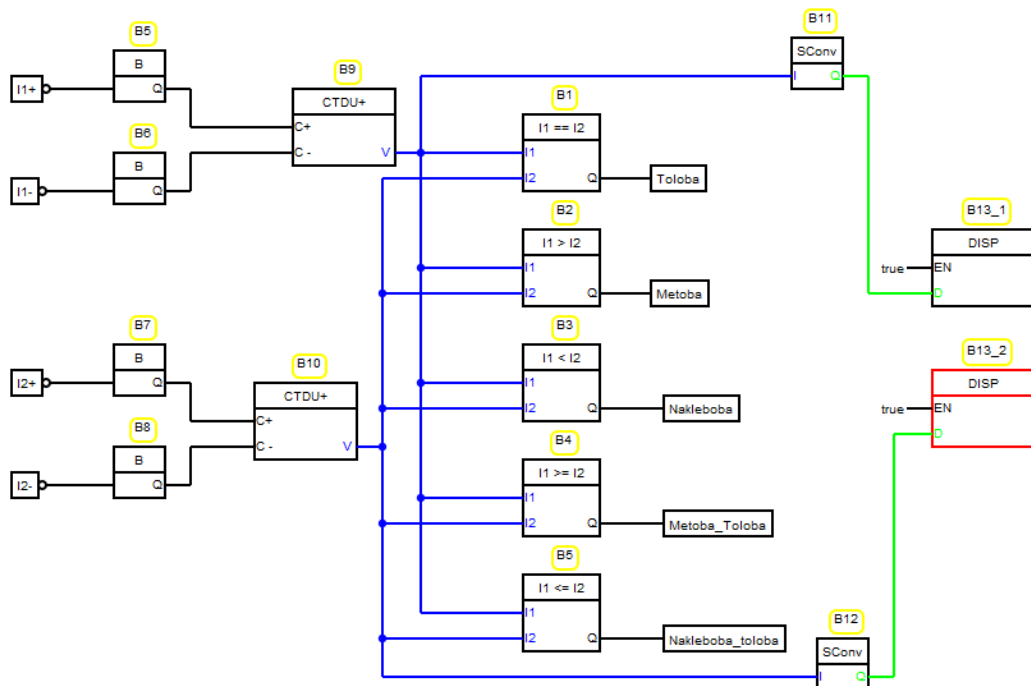
ფიგ. 19.1 შედარების ბლოკის პარამეტრიზაცია

ამის შემდეგ შევუდგეთ ამ ბლოკების მართვის პროგრამის შედგენას. ტეგების ზონაში (ფიგ. 19.2) შევქმნათ ოთხი ციფრული შესასვლელი: ორი ამ ბლოკების I1 შესასვლელისათვის და ორიც I2 შესასვლელისათვის. გადმოვიტანოთ ეს ოთხივე შესასვლელი პროგრამის სამუშაო ზონაში. იგივე ტეგების ზონაში შევქმნათ ხუთი ციფრული გამოსასვლელი რომლებსაც შევუერთებთ ჩვენი გამოსაკვლევი B1...B5 ბლოკების გამოსასვლელს და რომლებზედაც შემდეგ შევუერთებთ ცალკეულ შუქდიოდს წინააღმდეგობის გავლით.



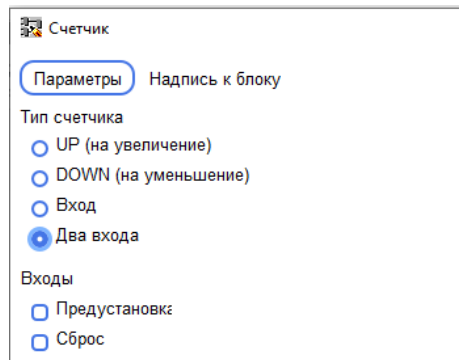
ფიგ. 19.2 ციფრული შესასვლელებისა და გამოსასვლელების შექმნა ტეგების ზონაში

ბლოკების ბიბლიოთეკის ბაზური ელემენტების საქალაქედან-Базовые элементы პროგრამის სამუშაო ზონაში გადმოვიტანოთ ოთხი ხელშეშლებისაგან დაცვის ბლოკი Bounce B5...B9 და შევუერთოთ ისინი ტეგების ზონიდან გადმოტანილ ციფრულ შესასვლელებს. ბიბლიოთეკიდან დაგვჭირდება კიდევ ორი ბლოკის გადმოტანა B9 და B10, ორივე მთვლელი-Счетчик და ამის შემდეგ შევუერთოთ ეს ყველა ბლოკი და შესასვლელ/გამოსასვლელები ერთმანეთთან ისე, როგორც ეს ფიგ. 19.3-ზეა წარმოდგენილი.



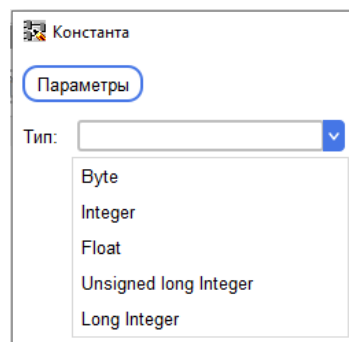
ფიგ. 19.3 შედარების ბლოკების ტესტირების პროგრამა

აქ სიახლე მდგომარეობს მთვლელების დაპროგრამებაზე. მათი პარამეტრიზაციის ფანჯარაში ჩართული გვაქვს მხოლოდ ერთი ჩამრთველი ორი შესასვლელი რაც იმას ნიშნავს, რომ მათში გამორთულია ყველა ჩვენთვის უკვე ცნობილი ფუნქცია: წინასწარი დაყენების ფუნქცია, ნულზე დაყენების ფუნქცია, ერთი შესასვლელით თვლის მიმართულების ცვლილების ფუნქცია და სხვ. მაგრამ შემოტანილია ახალი ფუნქცია, ორი შესასვლელით თვლის მიმართულების ცვლის ფუნქცია, რაც მიიღწევა მისი პარამეტრების ფანჯარაში ორი შესასვლელის-Два входа ჩამრთველის ჩართვით (ფიგ. 19.4). ამ შემთხვევაში მთვლელების ბლოკზე გამოჩნდება ორი შესასვლელი C+ და C-, სწადაც თუ მივაწვდით იმპულსებს პირველ მათგანზე, მაშინ მთვლელი იმუშავებს შეკრების რეჟიმში, ხოლო თუ მივაწვდით მეორეზე, მაშინ მთვლელი იმუშავებს გამოკლების რეჟიმში. თუ შევწყვეტთ იმპულსების მიწოდებას, მაშინ მთვლელი შეინარჩუნებს მასში ჩაწერილ ინფორმაციას.



ფიგ. 19.4 მთვლელების პარამეტრიზაციის ფანჯარა

ამ ორი მთვლელის ანალოგური გამოსასვლელიდან პირველი შეერთებულია შედარების ბლოკების პირველ I1 შესასვლელზე, ხოლო მეორეს გამოსასვლელი-იმავე ბლოკების მეორე I2 შესასვლელზე. აღვნიშნოთ აგრეთვე, რომ ამ ბლოკების შესასვლელზე შეიძლება მიწოდებულ იქნას ნებისმიერი ტიპის ინფორმაცია, რომელიც ნაჩვენებია ფიგ. 19.5-ზე.

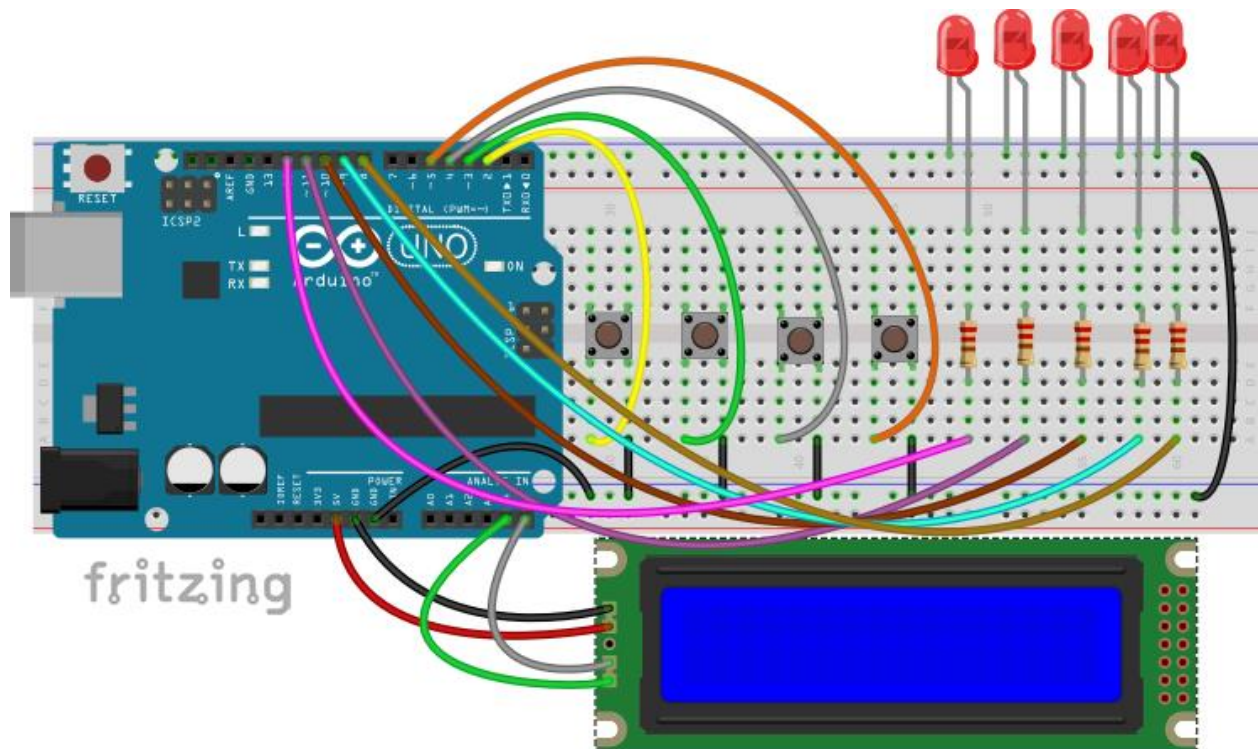


ფიგ. 19.5 შედარების ბლოკებზე მოდებული ინფორმაციის ტიპები.

ამის შემდეგ ჩავტერთოთ პროგრამა არდუინოში და დავიწყოთ ამ ბლოკების ტესტირება. მაგრამ წინასწარ ყოველთვის დავგჭირდება იმის ცოდნა თუ რა რიცხვებია ჩაწერილი მთვლელებში. ამის მისაღწევად გადმოვიტანოთ კიდეც ერთი ბლოკი ინდიკაციის ჩიპზე HD77480 და დავაპროგრამოთ ისე, რომ მის პირველ სტრიქონის ცენტრალურ ნაწილში ჩაიწეროს პირველი მთვლელის ინფორმაცია, ხოლო მეორე სტრიქონის ცენტრალურ ნაწილში- მეორე მთვლელის ინფორმაცია.

I1+, I1-, I2+, I2- დილაკების დახმარებით ჩავწერთ მთვლელებში ნებისმიერი ერთნაირი რიცხვი, მაგალითად 17 და დავაკვირდით შუქდიოდების მდგომარეობას. ამ შემთხვევაში უნდა აინთოს სამი შუქდიოდი: Toloba, Metoba_Toloba, Nakleboba_Toloba. ახლა ჩავწერთ პირველ მთვლელში მეტი რიცხვი ვიდრე მეორეში, მაგალითად 20 და 19. მაშინ უნდა აინთოს ორი შუქდიოდი: Metoba, Metoba_Toloba, ახლა ჩავწერთ პირველში უფრო ნაკლები რიცხვი ვიდრე მეორეში, მაგალითად 18 და 20, მაშინ უნდა აინთოს ორი შუქდიოდი Nakleboba, Nakleboba_Toloba. და ა.შ. ცვალებით რიცხვები მთვლელებში და გამოიკვლიეთ ყოველ კომბინაციაზე შუქდიოდების მდგომარეობა.

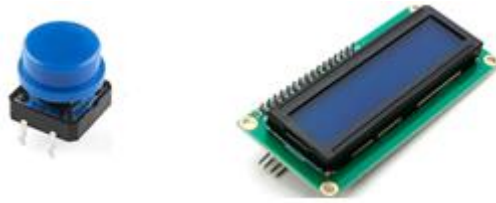
შედარების სქემების მართვის Fritzing სქემა ნაჩვენებია ფიგ. 19.6-ზე.



ფიგ. 19.6 შედარების სქემების მართვის Fritzing სქემა

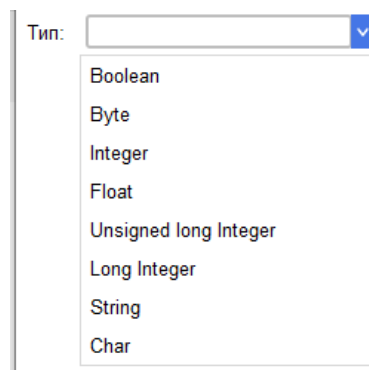
პროექტი_20

გადამრთველები



შევიდეთ FLProg პროგრამის ელემენტების ბიბლიოთეკის საქაღალდეში გადამრთველი-Переключатель და სამუშაო ზონაში გადმოვიტანოთ ბლოკი [SWITSH]. ეს ბლოკი წარმოადგენს ანალოგურ გადამრთველს, რომლის დანიშნულებაცაა გადასცეს გამოსასვლელზე ერთერთი ანალოგური სიგნალი, რომლებიც მიეწოდება „0“ ან „1“ ანალოგურ შესასვლელზე. თუ “S” შესასვლელზე ლოგიკური ნოლია, მაშინ სიგნალი გადაეცემა „0“ შესასვლელიდან, ხოლო თუ ლოგიკური ერთია, მაშინ სიგნალი გადაეცემა „1“ შესასვლელიდან.

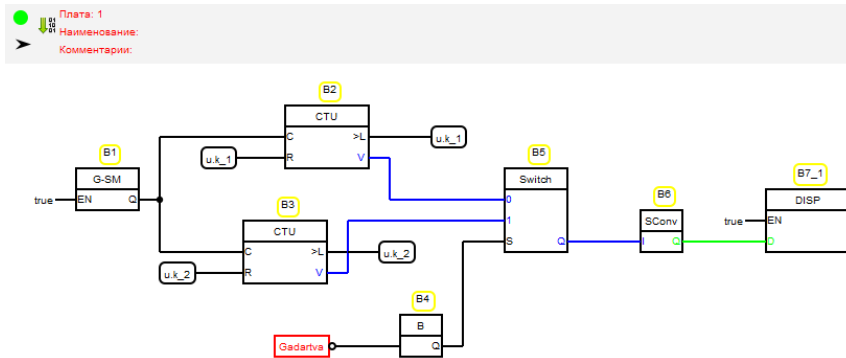
შეგახსენებთ, რომ ანალოგური სიგნალი FLProg-ში შეიძლება იყოს შემდეგი ტიპის (ფიგ. 20.1), ბულის - როდესაც სიგნალი იღებს ლოგიკურ მნიშვნელობებს ნოლს ან ერთს, ბაიტი, სიგნალი იღებს მნიშვნელობებს 0-დან ... 255-მდე, მთელი, რიცხვი იღებს მნიშვნელობებს - 32768-დან ... 32767-მდე, მცოცავმძიმისანი, რიცხვი იღებს მნიშვნელობებს -3.4028235e38-დან ... 3.4028235e38-მდე, უნიშნო გრძელი მთელი, რიცხვი იღებს მნიშვნელობებს 0-დან ... 4294967295-მდე, გრძელი მთელი, რიცხვი იღებს მნიშვნელობებს -2147483648-დან ... 2147483647-მდე, სტრიქონი და ბოლოს სიმბოლო, რომელიც შეესაბამება ერთ-ერთ რომელიმე სიმბოლოს ლათინური ალფაბეტიდან.



ფიგ. 20.1 ანალოგური სიგნალების სახეები გადამრთველებში

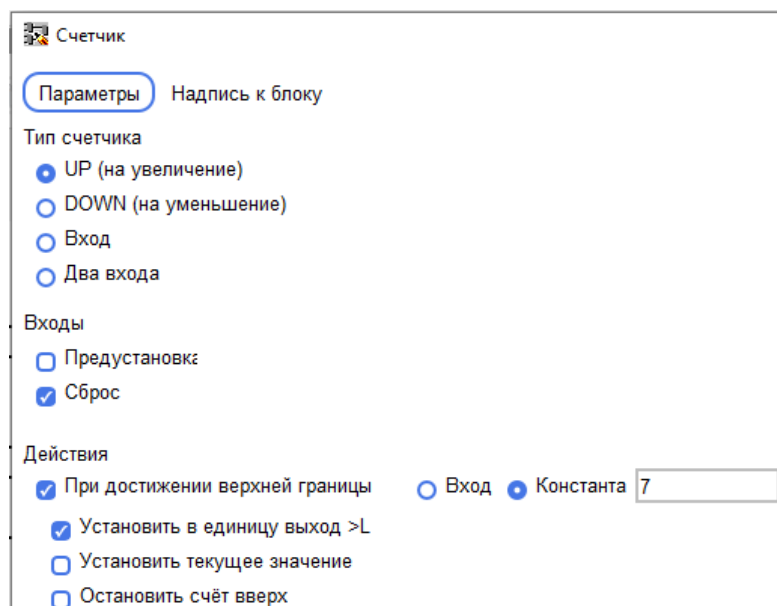
[SWITSH] ბლოკის ტესტირებისათვის უმჯობესია ავაწყოთ ორი მთვლელი, რომლებიც ციკლურად ითვლიან სხვადასხვა სიდიდემდე, მაგ: პირველი ექვსამდე, მეორე - ცხრამდე და

გარე ლილაკით მოვახდინოთ მაგათი მუშაობის გადართვა. ამისათვის, შევქმნათ ერთი ციფრული შესასვლელი სახელწოდებით ‘Gadartva’ არდუინოს მე-2 კონტაქტის გამოყენებით. რადგან ამ კონტაქტზე უნდა მიერთებულ იქნას გარე ლილაკი, ამიტომ ავაწყოთ ეს კონტაქტი სტანდარტულად, მოვნიშნოთ ყანყალისაგან დაცვისა და ამომქაჩავი რეზისტორის ალმები, ჩამოვიტანოთ ეს შესასვლელი პირველი დაფის სამუშაო ზონაში, დავაინვერტიროთ მისი გამოსასვლელი და Bounce ბლოკის გავლით B4 შევუერთოთ [SWITSH] ბლოკის S შესასვლელს B5 (ფიგ. 20.2).



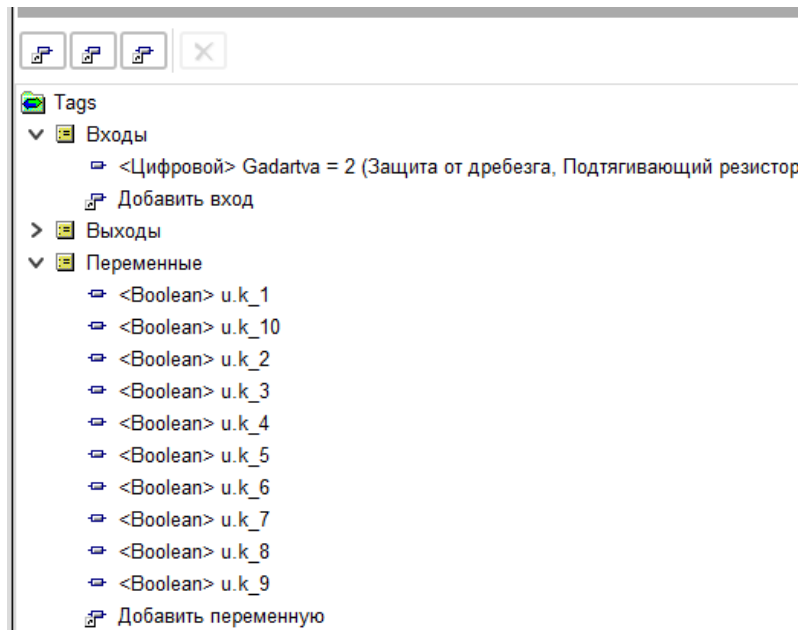
ფიგ. 20.2 [SWITSH] ბლოკის მართვა

ამავე დაფის სამუშაო ზონაში გადმოვიტანოთ ორი მთვლელი B2 და B3 და ავაწყოთ ისინი ციკლური თვლის რეჟიმზე შვიდამდე პირველი მთვლელისათვის და ათამდე მეორე მთვლელისათვის, იხილეთ ფიგ. 20.3 პირველი მთვლელისთვის, მეორე მთვლელიც დაპროგრამდება ანალოგიურად, მაგრამ კონსტანტაში 7-ის მაგივრად უნდა ჩაიწეროს 10.



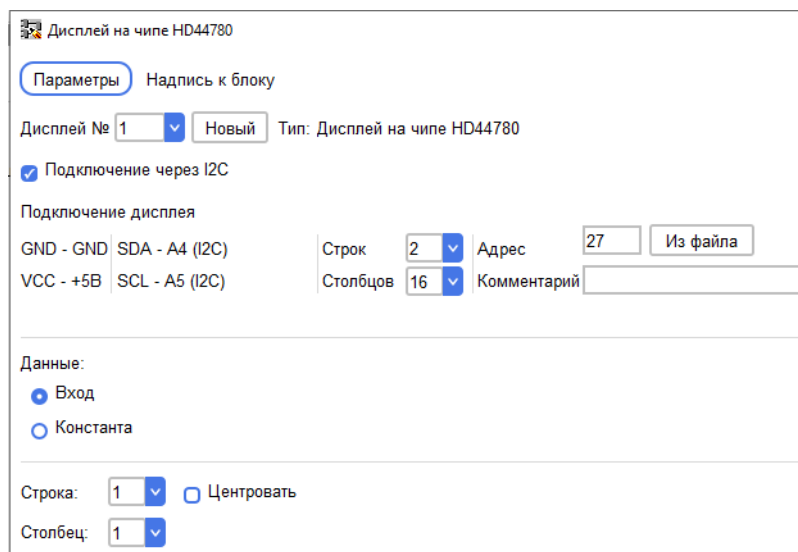
ფიგ. 20.3 პირველი მთვლელის პარამეტრიზაცია

გარდა ამისა, დაგჭირდება კიდეც ორი, ბულის ტიპის ცვლადის შექმნა უკუკავშირებისათვის, ერთი u.k_1 პირველი მთვლელისთვის და მეორე-u.k_2 მეორე მთვლელისათვის, იხ. ტეგების ზონა (ფიგ. 20.4).



ფიგ. 20.4 ტეგების ზონის შექმნა

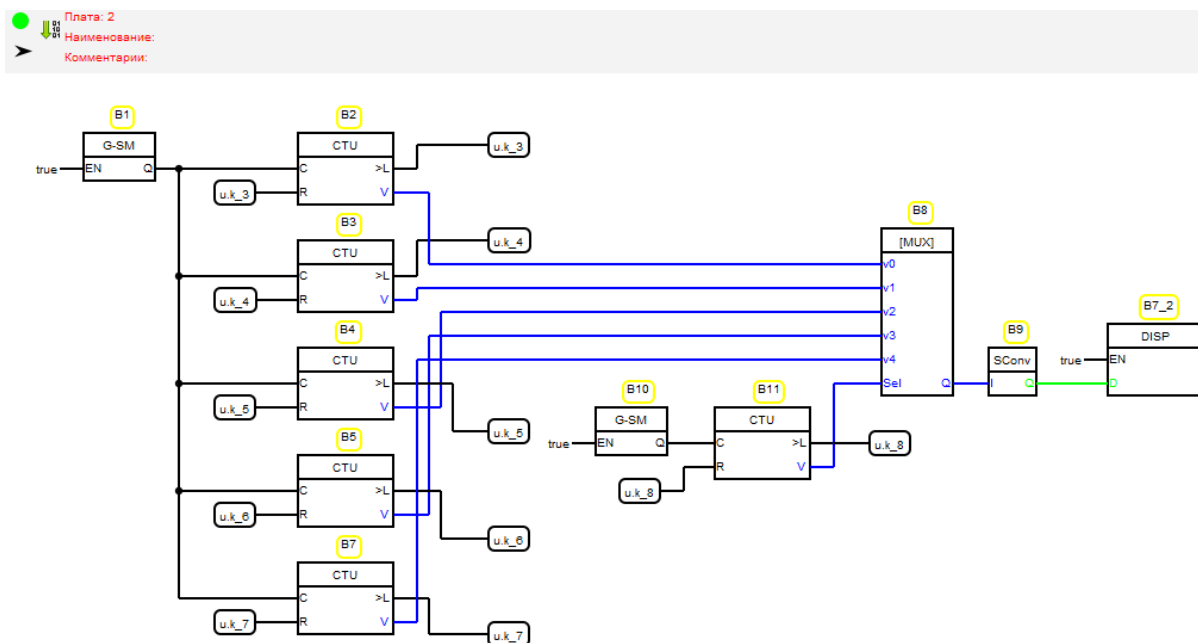
შევაერთოთ ეს უკუკავშირის ცვლადები მთვლელებზე ისე, როგორც ეს ფიგურა 1-ზეა ნაჩვენები. გარდა ამისა, შევაერთოთ აგრეთვე ამ მთვლელების ანალოგური გამოსასვლელები v [SWITSH] ბლოკის 0 და 1 შესასვლელებს შესაბამისად, ხოლო ამ ბლოკის გამოსასვლელი, კონვერტაციის ბლოკის გავლით B6 შევაერთოთ დისპლეის ბლოკის შესასვლელს B7_1, რომელიც დაპროგრამირებულია ფიგ. 20.5 ფანჯრის შესაბამისად.



ფიგ. 20.5 დისპლეის ბლოკის პირველი ნაწილის დაპროგრამება

ახლა დაგვჩა გენერატორის ბლოკის B1 პარამეტრიზაცია, რომელსაც დავაპროგრამებთ როგორც სიმეტრიულ მულტივიზრატორს 200 მილიწამიანი იმპულსის ხანგრძლივობით და შემდეგ მის გამოსასვლელს შევუერთებთ ჩვენს მიერ დაპროგრამებული მთვლელების შესასვლელებს. არდუინოს ჩატვირთვისა და გაშვების შემთხვევაში, თუ დისპლეიც შეერთებულია არდუინოს კონტაქტებს სტანდარტულად (ფიგ. 4). მაშინ [SWITSH] ბლოკი თავის გამოსასვლელზე გაატარებს სიგნალს თავისი 0-ის შესასვლელიდან, რადგან ღილაკის კონტაქტზე ნულოვანი სიგნალია, ანუ B2 ბლოკიდან. ამ შემთხვევაში დისპლეის პირველი სტრიქონის პირველ პოზიციაზე, რადგან გენერატორი მუშაობს და მთვლელები ითვლიან, აისახება პირველი მთვლელის მუშაობის პროცესი ანუ დისპლეიზე აისახება მიმდევრობით და ციკლურად რიცხვები 0,1,2,3,4,5,6,0,1,2... და ა.შ. თუ ჩვენ დავაწვებით ღილაკს 'Gadartva", მაშინ [SWITSH] ბლოკი თავის გამოსასვლელზე გაუშვებს სიგნალს ახლა უკვე თავისი 1 შესასვლელიდან, ანუ მე-2 მთვლელის სიგნალს და დისპლეის იგივე პოზიციაზე აისახება მიმდევრობით 0,1,2,3,4,5,6,7,8,9,0,1,2... და ა.შ.

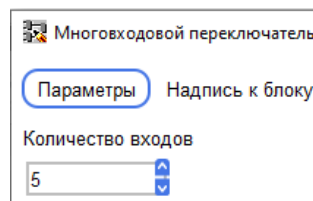
ახლა გადავიდეთ მრავალშესასვლელიანი გადამრთველისა მულტიპლექსორის [MUX] ბლოკის დაპროგრამებასა და აღწერაზე. მისი მართვის პროგრამა განვათავსოთ პროგრამის მე-2 დაფაზე (ფიგ. 20.6).



ფიგ. 20.6 [MUX] ბლოკის მართვის დაპროგრამება

[MUX] ბლოკი B8 ფაქტიურად იგივე ფუნქციას ასრულებს, რასაც [SWITSH] ბლოკი იმ განსხვავებით, რომ მას შეუძლია არა მხოლოდ ორი, არამედ ნებისმიერი რაოდენობის

ანალოგური სიგნალის კომუტაცია და გაგზავნა თავის გამოსასვლელზე. ბლოკის შესასვლელების რაოდენობის შერჩევა შეიძლება ბლოკის პარამეტრიზაციის ფანჯრიდან, რომელიც მიიღება ბლოკზე თავის მარჯვენა ღილაკის ორჯერ დაწკაპებით, იხ. ფიგ. 20.7, სადაც შეიძლება ჩაიწეროს შესასვლელების ნებისმიერი რაოდენობა. ჩვენს შემთხვევაში ჩაწერილი გვაქვს ციფრი 5, ეს ნიშნავს იმას, რომ ანალოგურ გარდაქმნელს აქვს ხუთი ანალოგური შესასვლელი.



ფიგ. 20.7 მრავალშესასვლელიან გადამრთველის პარამეტრიზაციის ფანჯარა

რადგან ჩვენს მიერ ჩაწერილ იქნა 5 შესასვლელი, ამიტომ [MUX] ბლოკის ფუნქციონირების კონტროლისათვის დაგჭირდება ხუთი მთვლელი B2 ... B7, რომლებიც აწყობილია 10, 13, 16, 19, 22 იმპულსების ციკლურ თვლაზე შესაბამისად. ამ მთვლელების პარამეტრიზაცია ისევე ხდება როგორც პირველ დაფაზე არსებული მთვლელებში. ამ მთვლელების უკუკავშირების უზრუნველსაყოფად ტეგების ზონაში შექმნილი გვაქვს u.k_3 ... u.k_7 ბულის ტიპის ცვლადები, რომლებიც შეერთებულნი არიან ფიგ. 20.6-ზე ნაჩვენები სქემის მიხედვით. მთვლელების იმპულსებით უზრუნველყოფისათვის შექმნილი გვაქვს სიმეტრიული გენერატორი B1 (მულტიპლექსორი) 200 მილიწამიანი იმპულსის ხანგრძლივობით. მთვლელების გამოსასვლელები შეერთებულია მულტიპლექსორის v0, v1, v2, v3 და v4 შესასვლელებთან. მათგან [MUX] ბლოკის გამოსასვლელზე შესაძლებელია გამოსული იყოს მხოლოდ ერთი რომელიმეს სიგნალი. არჩევა ხდება Sel შესასვლელზე მიწოდებული ანალოგური რიცხვის კოდით, თუ მიეწოდება 0, მაშინ აირჩევა v0 სიგნალი და გადაეწოდება გამოსასვლელზე, თუ მიეწოდება 1, მაშინ აირჩევა v1 სიგნალი და ა.შ. თუ მიეწოდება 4, მაშინ აირჩევა v4 სიგნალი. ამ Sel შესასვლელზე საჭირო ზემოთაღნიშნული სიგნალების ფორმირებისათვის აღებული გვაქვს B11 მთვლელი, რომელიც ციკლურად მუშაობს და ითვლის იმპულსებს 0-დან...4-ის ჩათვლით. ამ გენერატორის უკუკავშირისათვის შექმნილი გვაქვს u.k_8 ცვლადი, ხოლო მისი თვლის იმპულსების გენერირებისათვის გამოყენებული გვაქვს სიმეტრიული მულტივიბრატორი B10, რომელიც მუშაობს 10 წამიანი იმპულსის ხანგრძლივობით. მაშასადამე [MUX] ბლოკი არდუინოს ჩატვირთვისა და გაშვების შემთხვევაში მორიგეობით გაატარებს იმპულსებს v0, v1, v2, v3, v4 შესასვლელებიდან 10

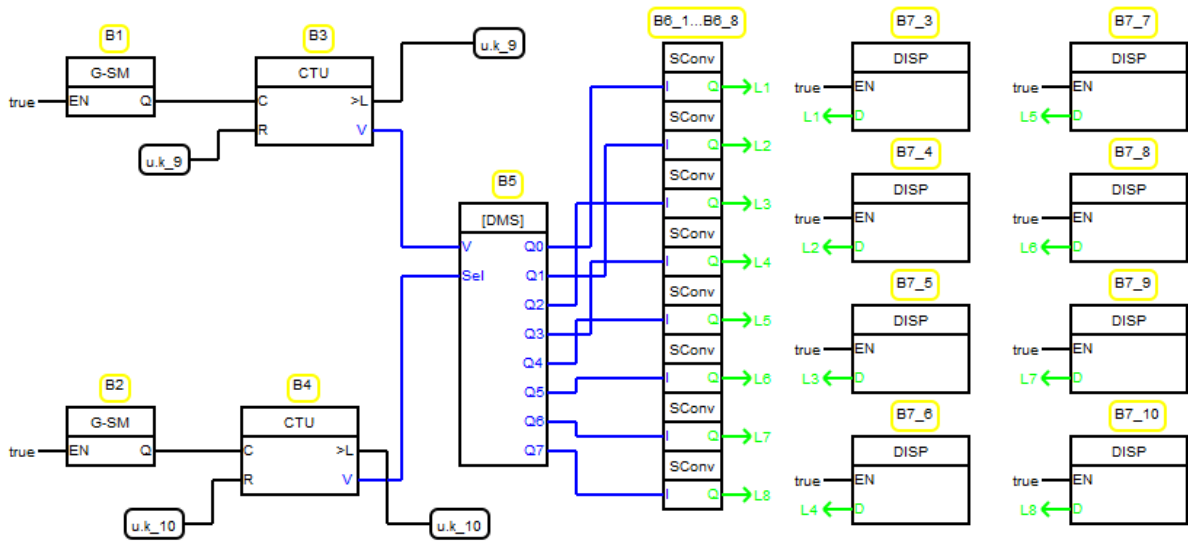
წამიანი ინტერვალებით. ეს კომპიუტერული სიგნალები B9 კონვერტაციის ბლოკის გავლით მიეწოდება B7 დისპლეის მე-2 ნაწილს B7_2, რომელიც დაპროგრამებულია ფიგ. 20.8-ზე ასახული სქემის მიხედვით.

ფიგ. 20.8 დისპლეის მე-2 ნაწილის პარამეტრიზაცია

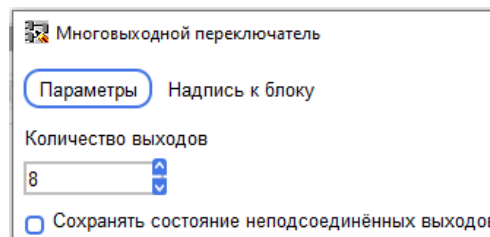
არდუინოს ჩატვირთვისა და გაშვების შემდეგ პირველი ათი წამის განმავლობაში დისპლეის მე-15 და მე-16 პოზიციებზე აისახება რიცხვების ცვლა 0-დან 9-მდე პირველი მთვლელიდან (B2 ბლოკი), მეორე 10 წამის განმავლობაში - რიცხვების ცვლა 0-დან 12-მდე მეორე მთვლელიდან (B3 ბლოკი) და ა.შ. მე-5 ათი წამის განმავლობაში - რიცხვების ცვლა 0-დან 21 (B6 ბლოკი), მე-6 10 წამის განმავლობაში-ისევ 0-დან 9-მდე პირველი მთვლელიდან და ა.შ. ციკლურად არდუინოს გამორთვამდე.

ახლა გადავიდეთ მესამე მრავალგამოსასვლელიანი გადამრთველის-Многовыходной Переключатель მართვის აღწერაზე, ესაა დემულტიპლექსორი [DMS] (ფიგ. 20.9), რომელიც ასრულებს მულტიპლექსორის შებრუნებულ ფუნქციას, ანუ ანალოგური სიგნალი ერთი შესასვლელიდან V გადააქვს ერთ-ერთ რომელიმე გამოსასვლელზე Q1, Q2... და თუ რომელზე კონკრეტულად, ეს დამოკიდებულია Sel შესასვლელზე მოდებული ანალოგური რიცხვის კოდით, თუ მიეწოდება 0, მაშინ სიგნალი გამოვა Q0 გამოსასვლელზე, თუ მიეწოდება 1, მაშინ სიგნალი გამოვა Q1 გამოსასვლელზე და ა.შ. თუ მიეწოდება 7, მაშინ სიგნალი გამოვა Q7 გამოსასვლელზე. ამ გამოსასვლელების რაოდენობა შეიძლება შერჩეულ იქნას ნაბისმიერად ფიგ. 20.10-ზე ნაჩვენები პარამეტრიზაციის ფანჯარის მიხედვით, რომელიც მიიღება [DMS]

ბლოკზე თავგის მარცხენა ღილაკის ორჯერ დაწკაპუნებით. ჩვენ შემთხვევაში აღებული გვაქვს რვა გამოსასვლელი Q1 ... Q7.

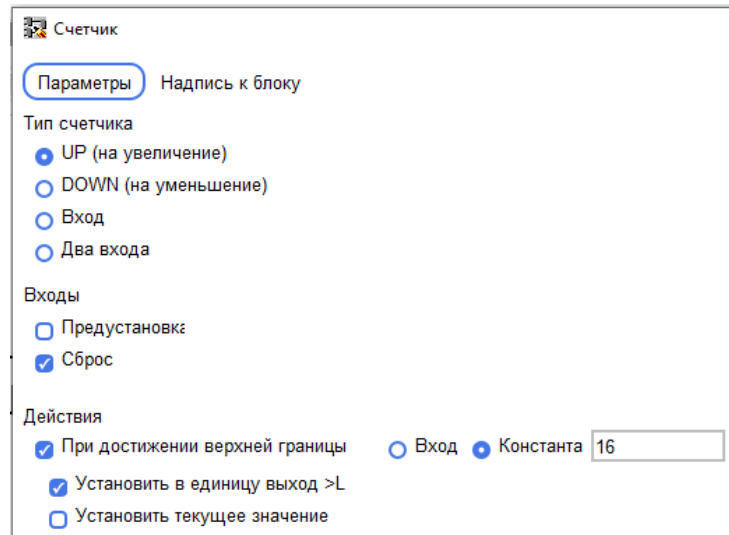


ფიგ. 20.9 დემულტიპლექსორის [DMS] მართვის პროგრამა



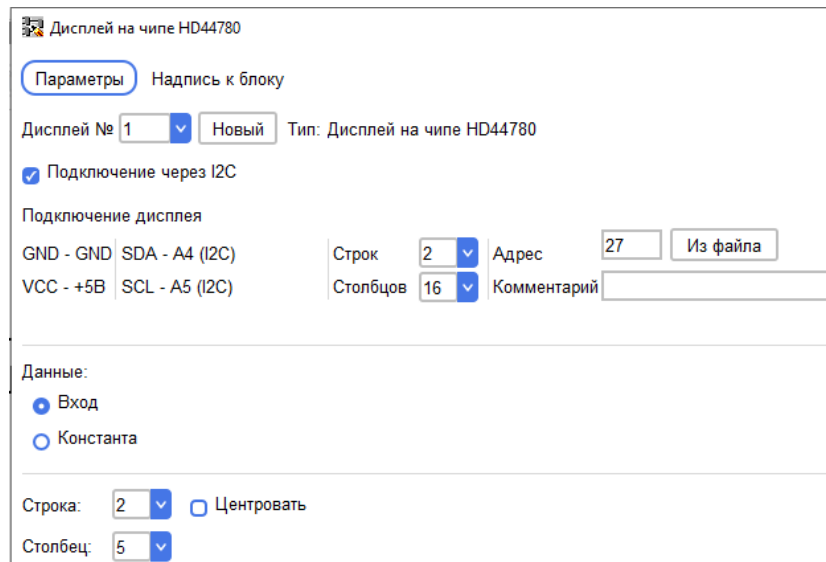
ფიგ. 20.10 მრავალგამოსასვლელიანი გადამრთველის პარამეტრიზაციის ფანჯარა

ამ პროექტში დემულტიპლექსორის ბლოკის მართვისთვის გამოყენებული გვაქვს ორი სიმეტრიული მულტივიზრატორი B1 და B2 და ორი მთვლეელი B3 და B4. მულტივიზრატორები დაპროგრამებულნი არიან 200 მილიწამიან და 10 წამიან იმპულსების ხანგრძლივობებზე შესაბამისად, ხოლო მთვლელები ითვლიან 16-მდე და 8-მდე, რაც მიიღწევა სტანდარტულად უკუკავშირის ბულის ტიპის ცვლადების u.k_9 და u.k_10 შექმნითა და გამოყენებით. გთავაზობთ ერთერი მთვლელის, კერძოდ B3-ის პარამეტრიზაციის ფანჯარას (ფიგ. 20.11).



ფიგ. 20.11 ერთერთი მთვლელის პარამეტრიზაციის ფანჯარა

პროგრამა ისე გვაქვს შედგენილი, რომ პირველი 10 წამის განმავლობაში პირველი მთვლელის მუშაობის თვის პროცესი 0-დან 15-მდე გამოვა B5 ბლოკის Q1 გამოსასვლელზე, მე-2 ათი წამის განმავლობაში-Q2 გამოსასვლელზე და ა.შ. მე-8 ათი წამის განმავლობაში Q7 გამოსასვლელზე, მე-9 ათი წამის განმავლობაში - ისევ B5 ბლოკის Q1 გამოსასვლელზე და ა.შ. ციკლურად არდუინოს გამორთვამდე. ეს პროცესი აისახება ჩვენს მიერ შემოტანილი დისპლეის ბლოკის B7 მე-2 სტრიქონში.

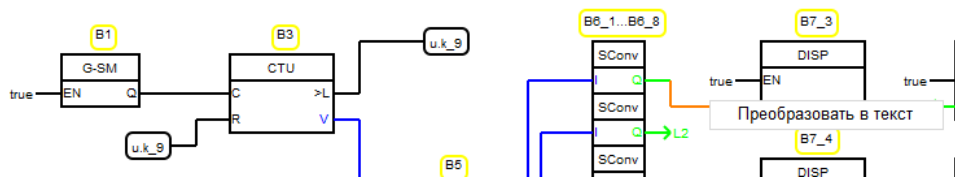


ფიგ. 20.12 დისპლეის მეორე სტრიქონის მესამე ნაწილის პარამეტრიზაციის ფანჯარა

ეს მეორე სტრიქონი პირობითად დაყოფილი გვაქვს რვა ნაწილად B7_3, B7_4 ... B7_10 ორ-ორი სვეტით თვითოეულში. Q1 გამოსასვლელის სიგნალები უნდა აისახოს დისპლეის

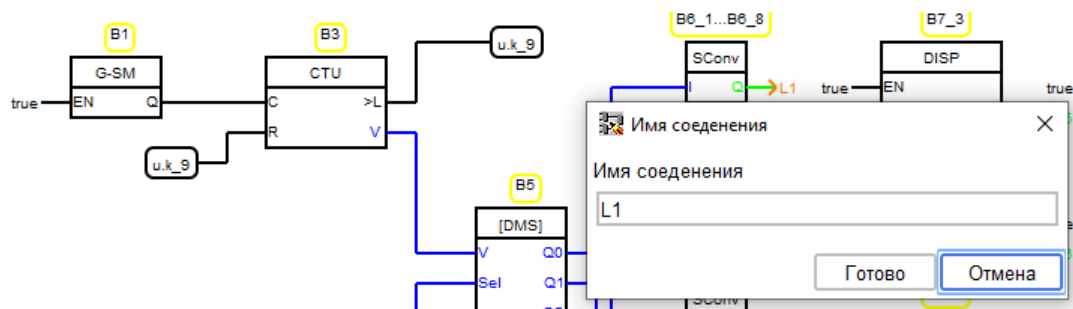
მეორე სტრიქონის პირველ ნაწილში ანუ პირველ და მეორე სვეტებში, Q1 გამოსასვლელის სიგნალები - დისპლეის მეორე ნაწილში ანუ მესამე და მეოთხე სვეტში და ა.შ.მ Q7 გამოსასვლელის სიგნალები - მე-15, მე-16 სვეტებში. ამ მიზნის მისაღწევად საჭირო იქნება დისპლეის ამ ნაწილების სათანადო პარამეტრიზაცია. ჩვენ გაჩვენებთ ერთერთი მათგანის, მაგ. მე-3 ნაწილის პარამეტრიზაციის ფანჯარას B7_5, დანარჩენი ნაწილების პარამეტრიზაცია მოხდება ანალოგიურად, მხოლოდ სვეტის-Столбец ველში იქნება სხვადასხვა ციფრები ჩასაწერი: 1,3,5,7,9,11,13,15 (ფიგ. 20.12). ეს სტრიქონი პირობითად დაყოფილი გვაქვს რვა ნაწილად B7_3, B7_4 ... B7_10 ორ-ორი სვეტით თვითოეულში. Q0 გამოსასვლელის სიგნალები უნდა აისახოს დისპლეის მეორე სტრიქონის პირველ ნაწილში ანუ პირველ და მეორე სვეტებში, Q1 გამოსასვლელის სიგნალები - დისპლეის მეორე ნაწილში ანუ მესამე და მეოთხე სვეტში და ა.შ.მ Q7 გამოსასვლელის სიგნალები - მე-15, მე-16 სვეტებში.

დისპლეის ამ ნაწილების [MUX] ბლოკის გამოსასვლელებთან შეერთების ხაზების განტვირთვის მიზნით, გამოყენებული გვაქვს შესაერთებელი წერტილების შეერთების დანომვრის მექანიზმი, რისთვისაც შევდივართ შემაერთებელი ხაზის კონტექსტურ მენიუში და ვაჭერთ ტექსტად გარდაქმნის შეტყობინებას-Преобразовать в текст (ფიგ. 20.13).

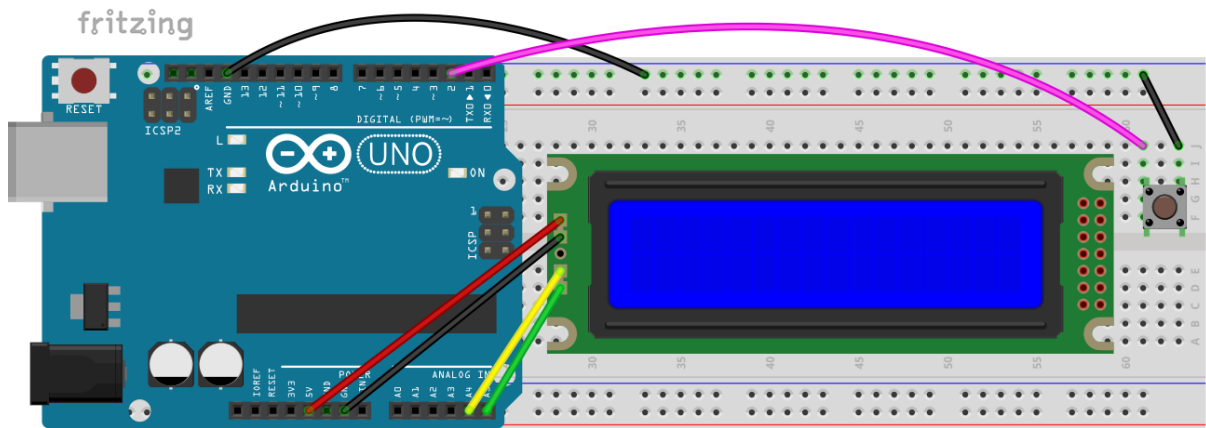


ფიგ. 20.13 შემაერთებელი ხაზის ტექსტად გარდაქმნის პროცედურა

ამის შემდეგ ეკრანზე გამოდის შემაერთებელი ხაზის დანომვრა (მაგ. L1), რომელსაც უნდა დავეთანხმოთ ან ჩავწეროთ ჩვენი ვერსია (ფიგ. 20.14), რის შემდეგაც შემაერთებელი ხაზი გაწყდება და ხაზის თავში და ბოლოში დაეწერება ერთნაირი ნომრები (ფიგ. 20.9). დემულტიპლექსორის მართვის Fritzing სქემა მოცემულია ფიგ. 20.15-ზე.



ფიგ. 20.14 შემაერთებელი ხაზის დანომვრა



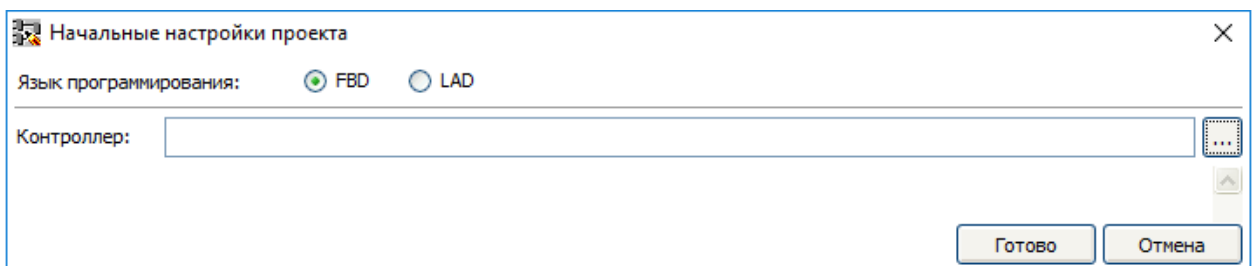
ფიგ. 20.15 დემულტიპლექსორის მართვის Fritzing სქემა

პროექტი_21

სერვო და ბიჯური ძრავების მართვა

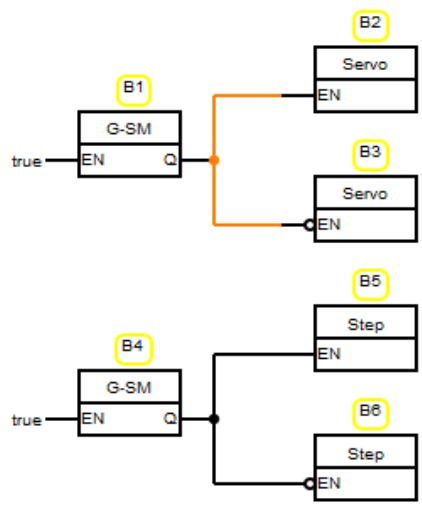


დასაწყისში რაც უნდა გავაკეთოთ არის ის, რომ გავუშვათ FLProg-ის პროგრამა, დავაწკაპუნოთ მთავარი მენიუს ფაილის მენიუზე (Файл), ჩამოშლადი სიიდან ავირჩიოთ ახალი-Новый პროექტი და გამოჩენილ ფანჯარაში (ფიგ. 21.1) დაპროგრამების ენა-Язык программирования ავირჩიოთ FBD. გარდა ამისა, კონტროლერის ველში ... დილაკზე დაჭერის შედეგად კონტროლერის-Контроллер: ჩამოშლად სიაში ვირჩევთ იმ კონტროლერს, რომელიც გვაქვს ხელთ (Arduino Uno ჩვენ შემთხვევაში ფიგ. 21.2). ამის შემდეგ კი ვაჭერთ არჩევის დილაკს-Выбрать, შემდეგ კი მზადყოფნის დილაკს-Готово.



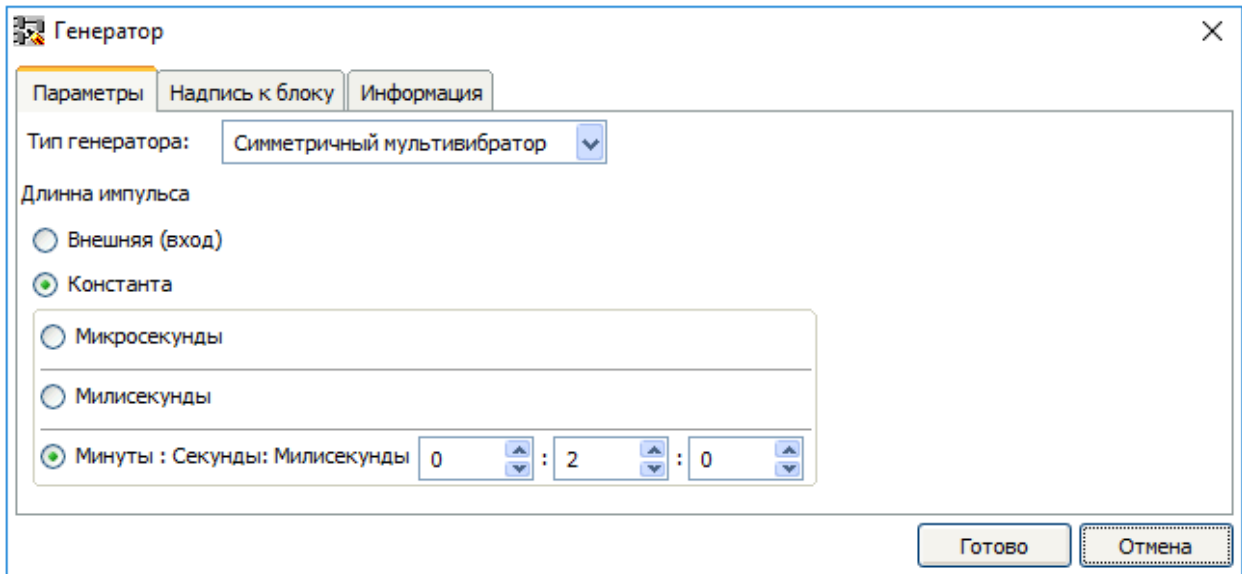
ფიგ. 21.1 პროექტის საწყისი გაწყობის ფანჯარა

ფიგურა 21.2-ზე წარმოდგენილია სერვომძრავისა და ბიჯური ძრავი და მათი მართვის პროგრამები, რომლებიც შესრულებულია Arduino-ს დაპროგრამების გარემოში FLProg (B1, B2, B3 და B4, B5, B6 ბლოკები შესაბამისად). სერვომძრავის მართვის დაპროგრამებისათვის საჭიროა FLProg-ის ელემენტების ბიბლიოთეკის-Библиотека элементов ტაიმერების საქაღალდედან-Таймеры თავის მარცხენა ღილაკის ჩავლებით პროგრამის სამუშაო ზონაში გადმოვაცოცოთ იმპულსების გენერატორის ბლოკი-Generator (B1) და მოვახდინოთ მისი დაპროგრამება. ამისათვის დავაწკაპუნოთ ამ ბლოკზე თავის მარცხენა ღილაკით ორჯერ და გამოჩენილ ფანჯარაში (იხ. ფიგ. 21.3) მოვახდინოთ მისი პარამეტრიზაცია. გენერატორის ტიპის ველში-Тип генератора ავირჩიოთ სიმეტრიული მულტივიბრატორი-Симетричный мультивибратор, იმპულსის სიგრძის ველში-длина импульса ჩავრთოთ ჩამრთველი კონსტანტა-константа და შემდეგ ჩავრთოთ წუთები: წამები: მილიწამები-ს ველის-Минуты: Секунды: Миллисекунды შესაბამის ფანჯარაში ჩავწეროთ ჩვენთვის საჭირო დროის ხანგრძლიობა, რა პერიოდულობითაც გვინდა შეასრულოს სერვომძრავმა წინდაუკან გადაადგილება. (ჩვენ მაგალითში აღებული გვაქვს 2 წამი).



ფიგ. 21.2 სერვო და ბიჯური ძრავების მართვა

გარდა ამისა, ვაწკაპუნებთ ამ ბლოკის EN შესასვლელზე და ვირჩევთ ნიბისდართვის ოპციას-True (1), რაც იმას ნიშნავს, რომ პროგრამის გაშვების შემთხვევაში გენერატორის მუშაობა ნებადართული იქნება განუწყვეტელი დროის განმავლობაში. ამის შემდეგ ვაწკაპებთ მზადყოფნის ღილაკს-Готово, რითაც გენერატორის დაპროგრამება დამთავრებული იქნება.

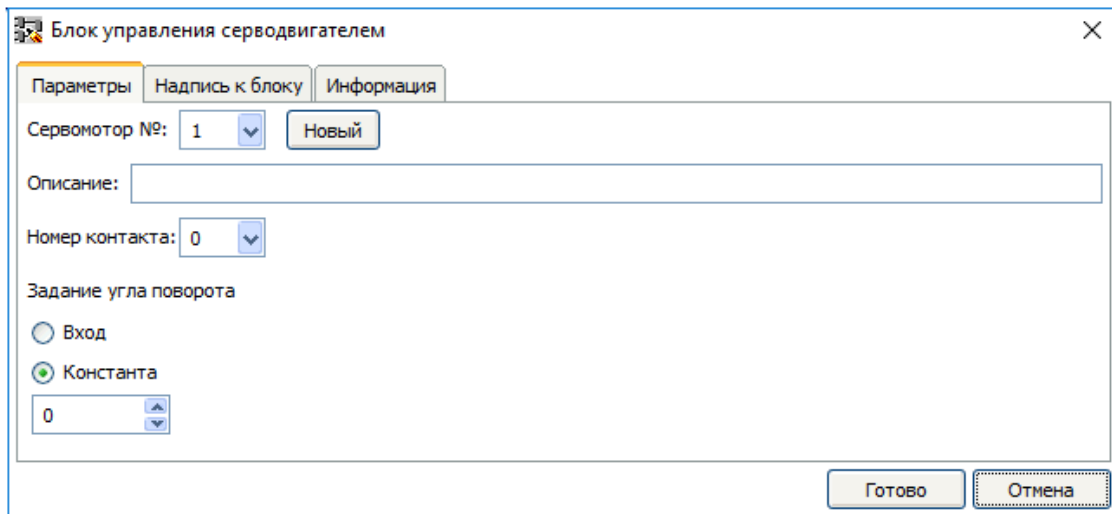


ფიგ. 21.3 გენერატორის დაპროგრამების (პარამეტრიზაციის) ფანჯარა

ამის შემდეგ გადავდივართ სერვომძრავის პარამეტრიზაციაზე. ამ მიზნით FLProg-ის ელემენტების ბიბლიოთეკის-Библиотека элементов ძრავების საქაღალდედან-Моторы, პროგრამის სამუშაო ზონაში გადმოვაცოცოთ სერვო ძრავის ბლოკი-ServoMotor B2, (B3) და მოვახდინოთ მისი დაპროგრამება. აქ უნდა გავითვალისწინოთ ის, რომ B2, B3 ბლოკები ფაქტიურად ერთი ფიზიკური სერვომძრავის აღმნიშვნელი ბლოკებია, მაგრამ განსხვავდებიან იმით, რომ თუ ერთი ბლოკი დაპროგრამებულია იმგვარად, რომ უზრუნველყოფს სერვომძრავის ღერძის ერთი მიმართულებით მოძრაობას, მეორე ასეთ შემთხვევაში, უზრუნველყოფს იგივე ძრავის მეორე მიმართულებით მოძრაობას. ამის განსახორციელებლად ჯერ ვაწკაპუნებთ B2 ბლოკზე და ვავსებთ სერვომძრავის მართვის ბლოკის პარამეტრიზაციის ფანჯარას (ფიგ. 21.4).

ფანჯრის პარამეტრების ჩანართში-Параметры ჯერ ვაწკაპუნებთ ღილაკს ახალი-Новый, რითაც სერვომძრავის ნომრის ველში-Сервомотор №: ავტომატურად ჩაიწერება ციფრი 1, რადგან სხვა რომელიმე ძრავი ამ პროექტში ჩვენს მიერ ჯერ არ ყოფილა გამოყენებული. კონტაქტის ნომრის ველში-Номер контакта: უნდა ჩავწეროთ ის კონტაქტი, რომელზედაც უნდა იქნას მიერთებული სერვომძრავის მართვის გამტარი (ნულოვანი კონტაქტი ჩვენს მაგალითში). მობრუნების კუთხის დავალების ველში-Задание угла поворота უნდა ჩავრთოთ ჩამრთველი კონსტანტა-Константа და შესაბამის ველში ჩავწეროთ მობრუნების რაიმე კუთხე (ჩვენ შემთხვევაში ვწერთ ციფრს 0, რაც მიუთითებს იმაზე, რომ სერვომძრავის ღერძი, სადაც უნდა იყოს ის, უნდა მობრუნდეს და დადგეს თავის ნულოვან მდგომარეობაში). ამის შემდეგ ვაჭერთ

მზადყოფნის ლილაკს-Готово, რითაც სერვოდრავის დაპროგრამების პირველი ეტაპი დამთავრებული იქნება.



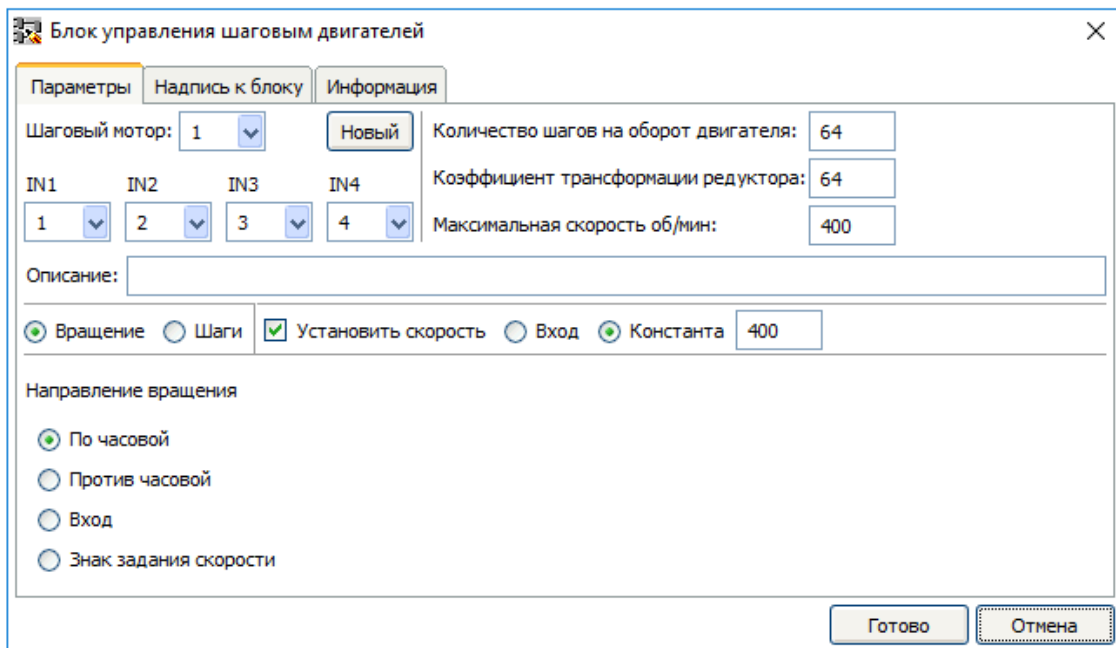
ფიგ. 21.4 სერვოდრავის მართვის ბლოკის პარამეტრიზაცია

ახლა გადავდივართ სერვოდრავის დაპროგრამების მეორე ეტაპზე. ამ მიზნით ვაკოპირებთ B2 ბლოკს და ჩავსვავთ პროგრამის სამუშაო ზონაში და მას დავარქმევთ B3 ბლოკს. გავხსნით ამ ბლოკის პარამეტრიზაციის ფანჯარას სადაც ყველა პარამეტრს უცვლელად დავტოვებთ, გარდა მობრუნების კუთხის დავალების ველისა-Задание угла поворота, სადაც ჩავწერთ ჩვენთვის სასურველ მობრუნების კუთხეს (180 გრადუსი ჩვენ შემთხვევაში). ვაწკაპებთ მზადყოფნის ლილაკს და ვამთავრებთ სერვოდრავის დაპროგრამებას.

ამის შემდეგ ვახდენთ B2, B3 ბლოკების EN შესასვლელების შეერთებას გენერატორის ბლოკთან B1. გარდა ამისა, ვაწკაპებთ B3 ბლოკის EN შესასვლელზე თავის მარჯვენა ლილაკით და გამოსულ ფანჯარაში ვირჩევთ ინვერტირებას-Инвертировать, რის შედეგადაც ამ შესასვლელზე გაჩნდება მცირე ზომის რგოლი, რაც მიუთითებს იმას, რომ ამ შესასვლელზე ხდება გენერატორიდან გამოსული სიგნალის ინვერსია. ამ პროგრამის გაშვების შემდეგ გენერატორი დაიწყებს იმპულსებისა და პაუზების განუწყვეტლივ გენერირებას. იმ დროს როდესაც მის გამოსასვლელზე იმპულსია (ლოგიკური ერთიანია) ნება ერთვება B2 ბლოკის მუშაობას, რაც იწვევს სერვოდრავის ღერძის დაყენებას 0 გრადუსზე, (B3 ბლოკი ამ დროს გაჩერებულია, რადგან მის შესასვლელზე მოდებული გენერატორის სიგნალი ინვერსიას განიცდის და მასზედ მოდებული ლოგიკური ნოლი ბლოკავს ამ ბლოკის მუშაობას). ხოლო როდესაც გენერატორის გამოსასვლელზე ორი წამის შემდეგ დადგება პაუზა, მაშინ ნება დაერთვება B3 ბლოკის მუშაობას (რადგანაც მის EN შესასვლელზე არსებული ლოგიკური

ნოლი ინვერსირების ოპერაციით გადაიქცევა ლოგიკურ ერთიანად და ნებას რთავს ამ ბლოკის მუშაობას). შედეგად კი სერვომძრავის ღერძი დადგება 180 გრადუსის მდგომარეობაში და ა.შ.შ.

ახლა გადავიდეთ ბიჯური ძრავის ბლოკის მართვაზე. FLProg-ის ელემენტების ბიბლიოთეკიდან პროგრამის სამუშაო ზონაში გადმოგვაქვს კიდეც ერთი გენერატორი B4, რომელსაც ზუსტად ისევე ვაპროგრამებთ როგორც B1 ბლოკს, იმ განსხვავებით, რომ ჩვენი მაგალითის შემთხვევაში იმპულსებისა და პაუზების ხანგრძლივობებს ვიღებთ 4 წამის სიდიდისას. ამის შემდეგ ძრავების საქალაქედან გადმოგვაქვს ბიჯური ძრავის ბლოკი-StepMotor-B5.



ფიგ. 21.5 ბიჯური ძრავის მართვის ბლოკის პარამეტრიზაცია

ვხსნით ამ ბლოკის პარამეტრიზაციის ფანჯარას (ფიგ. 21.5) და აქაც ვაწკაპებთ ღილაკს ახალი-Новый, რაც გამოიწვევს ველში ბიჯური ძრავი ერთიანის ავტომატურ ჩაწერას და ასევე ველებს: ძრავის ერთ ბრუნვაზე ბიჯების რაოდენობა-Количество шагов на оборот двигателя.; რედუქტორის ტრანსფორმაციის კოეფიციენტი-Коефициент трансформации редуктора: და მაქსიმალური სიჩქარე ბრ/წთ-Максимальная скорость об/мин: ავტომატურ შევსებას. ძრავის მართვის სიგნალების ველებში IN1, IN2, IN3, IN4 ვწერთ არდუინოს იმ კონტაქტების ნომრებს რომლებზედაც ვაერთებთ ბიჯური ძრავის ზემოთ აღნისნულ მართვის კონტაქტებს (არდუინოს 1, 2, 3, 4 კონტაქტები ჩვენი მაგალითის შემთხვევაში). ამის შემდეგ, ბრუნვასა-Вращение და ბიჯებს-Шаги შორის ვირჩევთ ბრუნვას, მოვნიშნავთ სიჩქარის დამყარებას-установить скорость, ჩავრთავთ კონსტანტას-Константа და შესაბამის ველში ჩავწერთ იმ სიჩქარეს, რა სიჩქარითაც გვინდა იბრუნოს ბიჯურმა ძრავმა (400 ბრ/წთ-ში ჩვენი მაგალითის

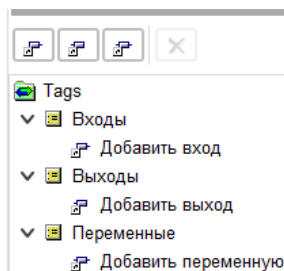
შემთხვევაში. ველში ბრუნვის მიმართულება-Направление Вращения ჩავრთავთ საათის ისრის მიმართულებას-По часовой და ვაჭერთ მზადყოფნის ღილაკს-Готово, რითაც ბიჯური ძრავის ბლოკის დაპროგრამება ერთი მიმართულებით მოძრაობისათვის დამთავრებული იქნება.

ამის შემდეგ გადავდივართ იგივე ძრავის დაპროგრამებაზე უკუ მიმართულებით მოძრაობისათვის. ამისათვის ვაკოპირებთ B5 ბლოკს და ჩავსვავთ იქვე პროგრამის სამუშაო ზონაში დავარქმევთ მას ბლოკს B6, გავხსნით მისი პარამეტრიზაციის ფანჯარას და ვცვლით მხოლოდ მოძრაობის მიმართულებას საათის ისრის საწინააღმდეგოდ-Против часовой და დავხურავთ ამ ფანჯარასაც, რითაც ბიჯური ძრავის პარამეტრიზაცია დამთავრებული იქნება. შემდეგ ვაერთებთ ამ B5 და B6 ბლოკების EN შესასვლელებს გენერატორის გამოსასვლელებზე. B6 ბლოკის შესასვლელებზე, ისევე როგორც B3 ბლოკის შესასვლელებზე მოვახდენთ სიგნალის ინვერტირებას და ბოლოს ღილაკზე Готово დაჭერით დავამთავრებთ მთელი პროექტის დაპროგრამირებას.

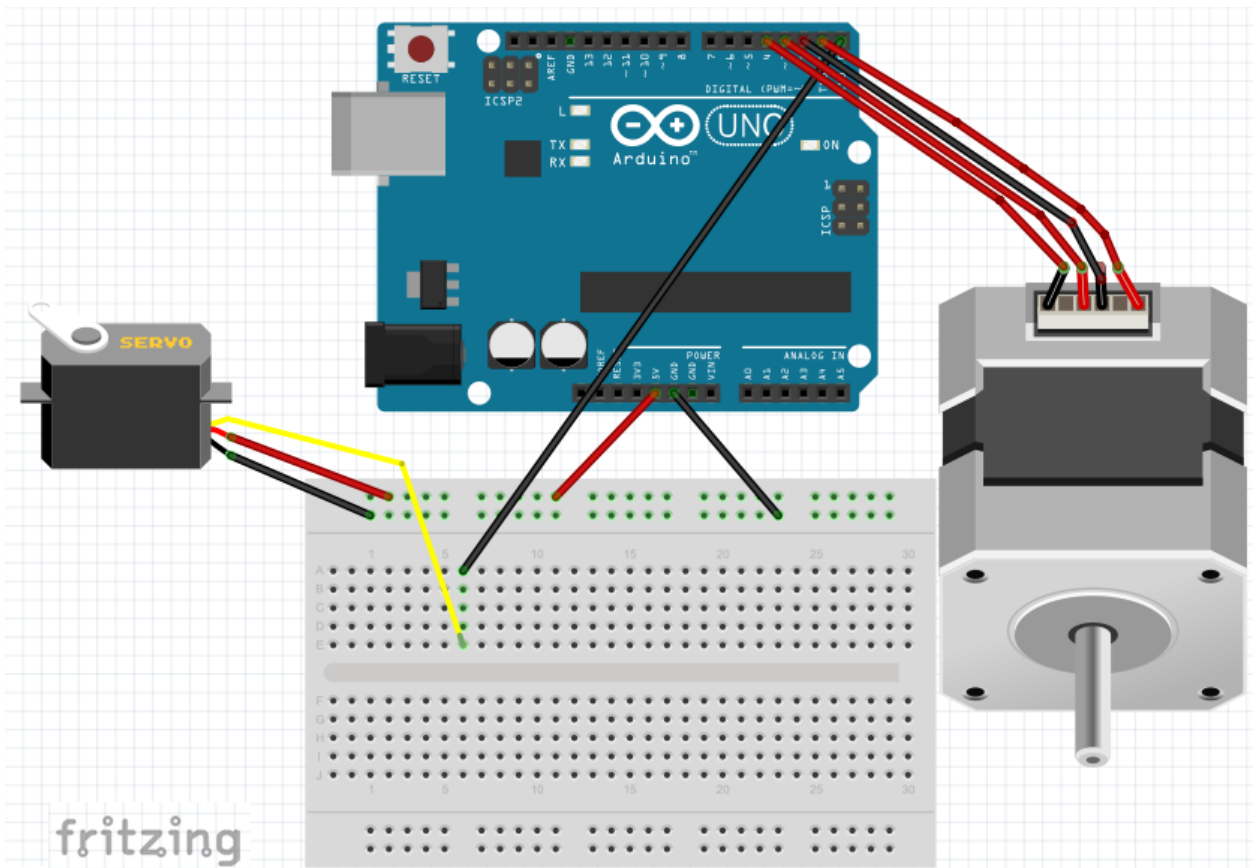
პროგრამის გაშვების შემდეგ ბიჯური ძრავის ღერძი დაიწყებს წინდაუკან მოძრაობას ოთხ-ოთხი წამის შუალედებით, ხოლო სერვოდრავის ღერძი ასევე შეიცვლის თავისი მდებარეობის კუთხეებს ორორი წამის შუალედებით და ეს მოძრაობები გაგრძელდება უსასრულოდ პროგრამის მუშაობის შეწყვეტამდე.

პროექტში ტეგების ზონაში ჩანაწერების გაკეთება არ არის საჭირო (ფიგ.21.6), რადგან ძრავების არდუინოსთან შეერთებების ფორმირება ხდება მათს შესაბამის პარამეტრიზაციის ფანჯრებში.

ფიგ. 21.7-ზე წარმოდგენილია პროექტის ბლოკ სქემა, რომელიც შესრულებულია Fritzing პროგრამის გამოყენებით.



ფიგ. 21.6 ტეგების ზონა



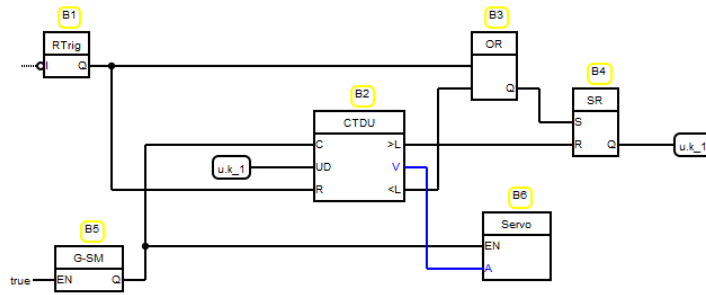
ფიგ. 21.7 პროექტის Fritzing სქემა.

პროექტი_22

ერთი სერვო ძრავის მართვა

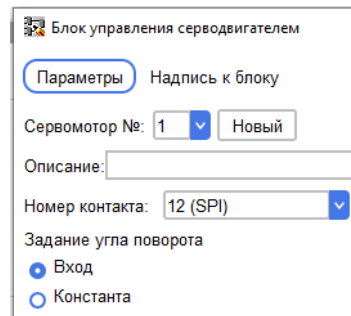


წინა პროექტში ჩვენ განვიხილეთ სერვო ძრავის დისკრეტული მართვა კონსტანტებით მარტო ორ კუთხეზე 0-დან 180 გრადუსამდე მისი ღერძის მობრუნებით. მაგრამ პრაქტიკაში უფრო ფართოდ გამოიყენება ამ ძრავის მართვა მდორე გადაადგილებით კუთხეების გარკვეული სიდიდეებზე, ასე ვთქვათ ანალოგიურად. ეს ამოცანა მოითხოვს სერვოდრავის შესაბამის პარამეტრიზაციას. ფიგ. 22.1-ზე ნაჩვენებია ერთი სერვოდრავის მართვა აღნიშნული მეთოდით.



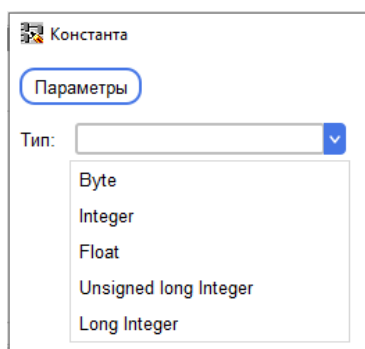
ფიგ. 22.1 სერვომოტორის ანალოგიური მართვა

გადმოვიტანოთ FLProg-ის სამუშაო ზონაში ბლოკების ბიბლიოთეკის ძრავების საქაღალდედან Motorы სერვომოტორის ბლოკი-ServoMotor-B6 და დავაწკაპუნოთ მასზედ ორჯერ თავის მარცხენა ღილაკით. გაიხსნება სერვომოტორის პარამეტრიზაციის ფანჯარა სადაც ჩავრთავთ კუთხის მობრუნების დავალებს - Задание угла поворота ორი ჩამრთველიდან ერთერთს, (ფიგ. 22.2) კერძოთ შესასვლელს-Вход და დავაჭერთ მზადყოფნის ღილაკს - Готово თავის მარცხენა ღილაკით.



ფიგ. 22.2 სერვომოტორის პარამეტრიზაციის ფანჯარა

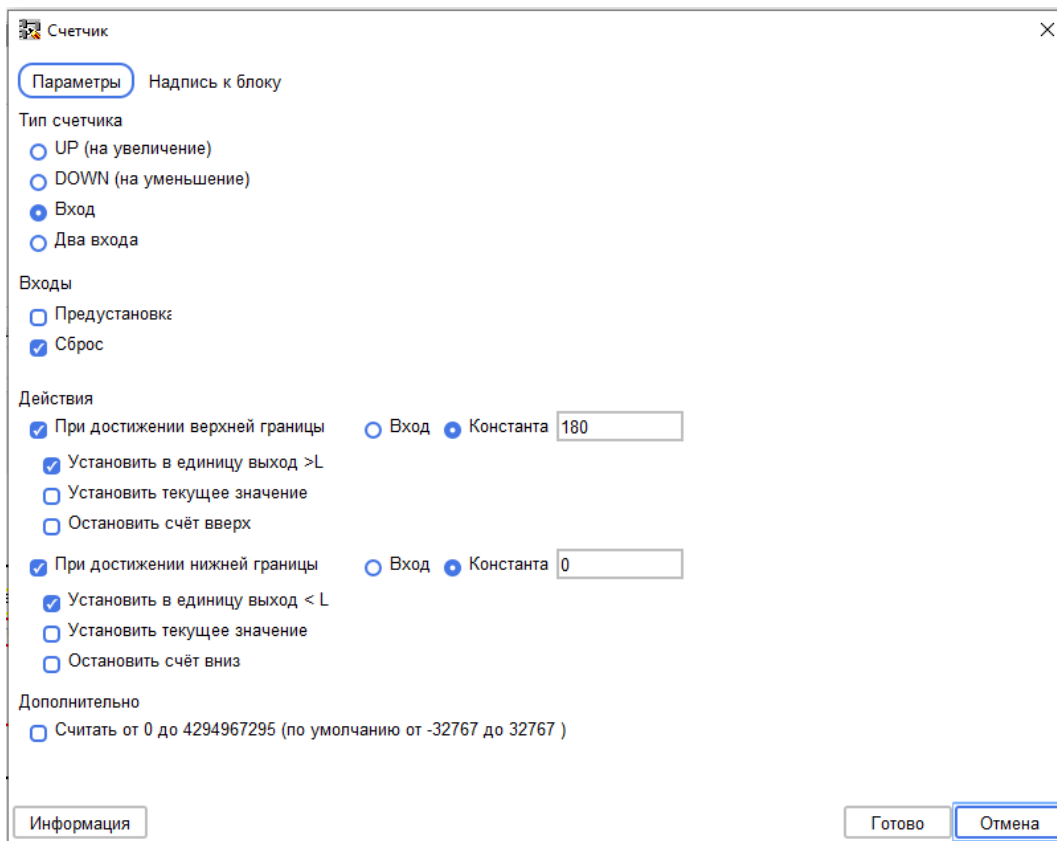
ამით დავინახავთ, რომ სერვომოტორის ბლოკს პროექტში გაუჩნდება მე-2 შესასვლელიც A (ფიგ. 22.1). ამ შესასვლელზე შეიძლება მიწოდებულ იქნას სიდიდეები 0-დან 180-მდე. ამ რიცხვს აღიქვამს სერვომოტორის ბლოკი და რა რიცხვიც მიეწოდება ამ შესასვლელზე იგივე კუთხით მოაბრუნებს სერვომოტორის ღერძს. ეს შესასვლელი არის ანალოგიური ტიპის შესასვლელი, რომელზედაც შეიძლება მიწოდებულ იქნას ნებისმიერი ტიპის რიცხვი (ფიგ. 22.3).



ფიგ. 22.3 სერვოდრავის შესასვლელზე მიწოდებული დასაშვები რიცხვების ტიპები

ამ პროექტში ჩვენ შემოვიფარგლებით Integer ტიპის რიცხვებით, რომელთა გენერირება ხდება რევერსიული მთვლელის B2 გამოყენებით. ეს მთვლელი დადებით თვლის რეჟიმში გადაითვლის მასზედ მიწოდებულ იმპულსებს და გამოსცევს თავის V გამოსასვლელზე რიცხვებს 0-დან 180-მდე, შემდეგ გადადის უარყოფითი თვლის რეჟიმში და ახლა უკვე იმავე სიჩქარით ჩამოდის 0-მდე, შემდეგ ისევ გადადის დადებითი თვლის რეჟიმში და ისევ ადის 180-მდე და ა.შ. დაუსრულებლად.

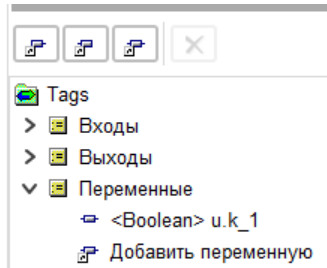
რევერსული მთვლელის მუშაობისათვის საჭირო იმპულსების გამომუშავებისათვის გამოყენებული გვაქვს გენერატორის ბლოკი B5, რომელიც დაპროგრამებულია ფიგ. 22.4-ზე წარმოდგენილი პარამეტრიზაციის ფანჯრის მიხედვით. როგორც ფიგურიდან ჩანს როდესაც გენერატორი დაითვლის 180-მდე, მაშინ მის >L გამოსასვლელზე გამომუშავდება ლოგიკური ერთიანი, ხოლო როდესაც გენერატორი ჩამოვა 0-მდე, მაშინ მის <L გამოსასვლელზე გამომუშავდება ლოგიკური ერთიანი. სწორედ ეს გამოსასვლელი უნდა გამოვიყენოთ ჩვენ მთვლელის მუშაობის გადასართველად შეკრების რეჟიმიდან გამოკლების რეჟიმზე და პირიქით.



ფიგ. 22.4 მთვლელის ბლოკის პარამეტრიზაციის ფანჯარა

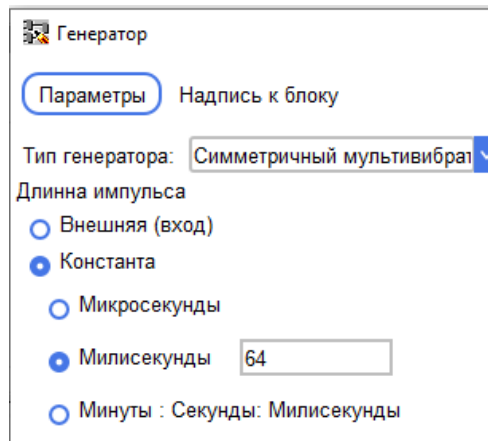
ახლა განვიხილოთ ფიგ. 22.1-ზე წარმოდგენილი მართვის სქემის მუშაობა. არდუინოს ჩატვირთვისა და გაშვების შემდეგ ჯერ ამუშავდება B1 ტრიგერი და გამოიმუშავებს ერთჯერად იმპულსს, რომელიც გაგრძელდება პროგრამის მუშაობის ერთი ციკლის განმავლობაში და რომელიც ლოგიკური შეკრების ბლოკის B3 გავლით დააყენებს SR ტრიგერს B4 ერთიანის მდგომარეობაში. ამ ერთიანმა უნდა დააყენოს მთვლელი B2 დადებითი თვლის რეჟიმში, მაგრამ რადგან პირდაპირი შეერთება უკუკავშირის ცვლადის გამოუყენებლად არ შეიძლება ჩვენი სამუშაო პროგრამის მოთხოვნების გათვალისწინებით, ამიტომ უნდა შევქმნათ ცვლადი და ამ ცვლადის გამოყენებით უნდა შევასრულოთ ეს შეერთება.

ამ მიზნით ტეგების ზონაში შევქმნათ ცვლადი და დავარქვათ მას პირობითად u.k_1 (უკუკავშირი_1), იხ. ფიგ. 22.5. გადმოვიტანოთ ეს ცვლადი ორჯერ სამუშაო ზონაში, ერთი შევუერთოთ SR ტრიგერის გამოსასვლელს, მეორე კი მთვლელის რეჟიმების გადართვის UD შესასვლელს და ამრიგად ეს ჩვენი უკუკავშირი მართალია შემოვლითი გზით, მაგრამ მაინც შესრულებული იქნება.



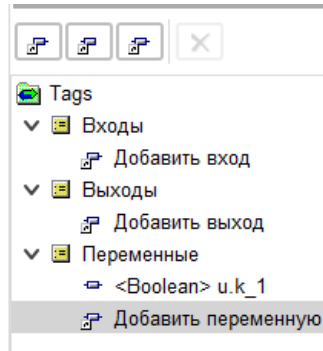
ფიგ. 22.5 ცვლადის u.k_1-ის შექმნა ტეგების ზონაში

ამავდროულად ამუშავდება იმპულსების გენერატორი B5, რომელიც დაპროგრამებული გვაქვს როგორც სიმეტრიული მულტივიბრატორი, 64 მილიწამის ხანგრძლიობის იმპულსის გამომუშავებაზე (ფიგ. 22.6) და რადგან გენერატორის ტიპი სიმეტრიულია, ამიტომ ჰაუზაც იმპულსებს შორის სიმეტრილი იქნება.

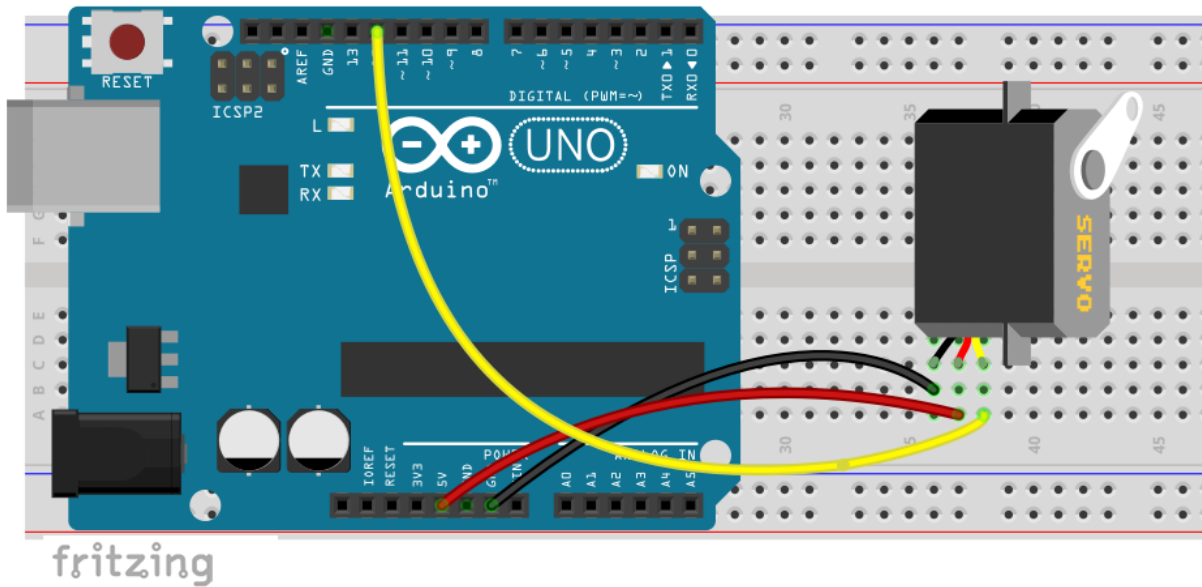


ფიგ. 22.6 იმპულსების გენერატორის პარამეტრიზაცია

მთვლელი B2 დაიწყებს იმპულსების თვლას და რადგან მის შესასვლელზე ლოგიკური ერთიანია SR ტრიგერიდან, ამიტომ ის ითვლის პირდაპირი (შეკრების) თვლის რეჟიმში. როდესაც მთვლელი დაითვლის 180 იმპულსს, მაშინ მის $>L$ დადგება ერთიანი, რომელის დააყენებს SR ტრიგერს ნულიანის მდგომარეობაში. ეს ნული მიეწოდება მთვლელის რეჟიმების გადართვის UD შესასვლელს და გადართავს მას უკუ თვლის (გამოკლების) რეჟიმში. როდესაც მთვლელი ჩამოვა ნულამდე, მაშინ გამომუშავდება სიგნალი $<L$, რომელიც ისევ დააყენებს RS ტრიგერს, ისევ ლოგიკური შეკრების ბლოკის B3 გავლით და .შ.შ პროცესი გაგრძელდება დაუსრულებლად არდუინოს გამორთვამდე. მთვლელის მუშაობა რევერსიულ რეჟიმში 0-დან 180-მდე და 180-დან ნულამდე იწვევს სერვომრავის ღერძის მდორე გადაადგილებას წინ და იუკან აღნიშნულ დიაპაზონში. პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 22.7, ხოლო ერთი ძრავის მართვის Fritzing სქემა მოცემულია ფიგ. 22.8-ზე.



ფიგ. 22.7 პროექტის ტეგების ზონის ფანჯარა



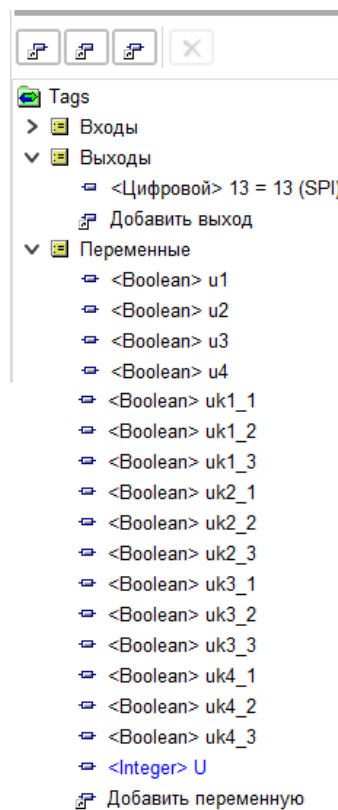
ფიგ. 22.8 ერთი ძრავის მართვის Fritzing სქემა

პროექტი_23

ოთხი სერვო ძრავის მართვა



ოთხი სერვოდრავის მართვის პროგრამა განთავსებულია FLProg დამუშავების გარემოს ხუთ დაფაზე, აქედან პირველ დაფაზე განლაგებულია ყველა, ოთხივე სერვოდრავისთვის საერთო მართვის პროგრამა, რომელიც გამოიმუშავებს სიგნალებს, რომლებიც გადაეწოდება ყველა დანარჩენ ოთხ დაფას და გამოიყენება ამ დაფებზე განლაგებული სერვოდრავების ინდივიდუალური მართვის პროგრამების სამართავად. პირველ დაფაზე მართვის სიგნალების ფორმირებისათვის შექმნილი გვაქვს u1, u2, u3 და u4 ბულის ტიპის ცვლადები. იხილეთ ფიგ. 23.1 სადაც გამოსახულია ამ პროექტის FLProg პროგრამის ტეგების ზონა.

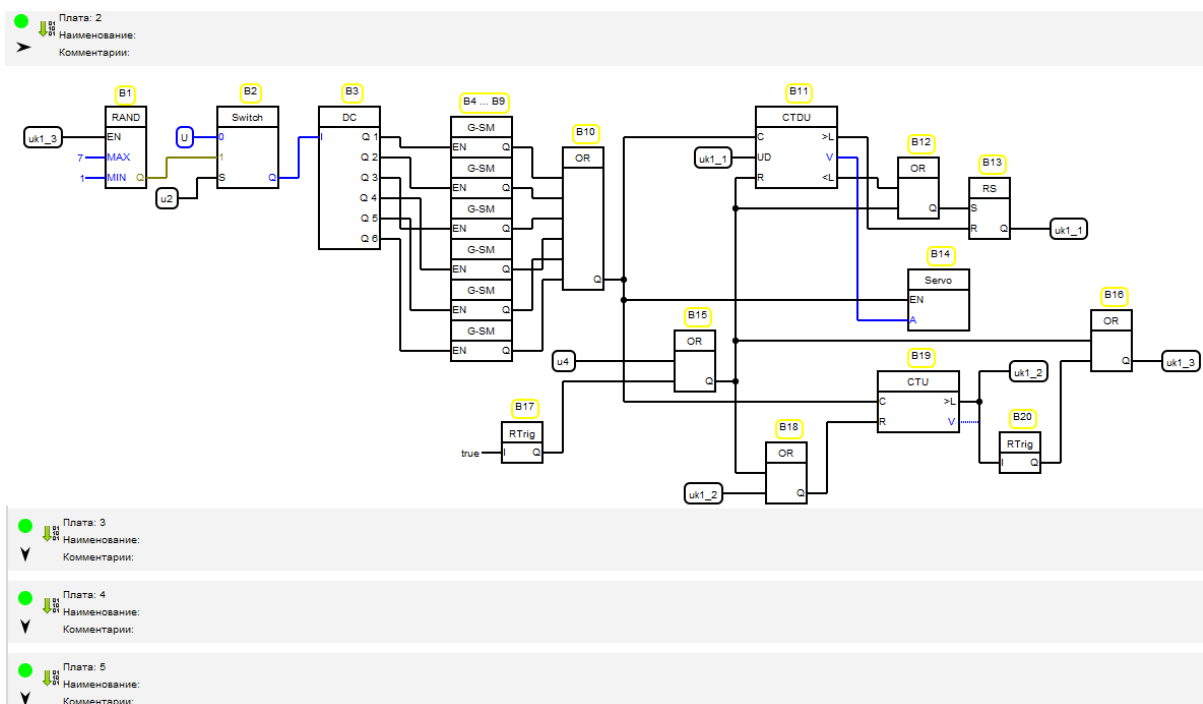


ფიგ. 23.1. პროგრამის ტეგების ზონა

ამ ცვლადებიდან u1, და u3 ცვლადები არიან ინდივიდუალური მოხმარების, ისინი უბრალოდ მთვლელების უკუკავშირების შესაქმნელად არიან გამოყენებული, ხოლო u2 და u4 არიან საერთო დანიშნულების ცვლადები და გადაეცემა სხვა ოთხ დაფას ინდივიდუალური სერვოდრავების მართვის ფუნქციების შესასრულებლად. ტეგების ზონაში შექმნილია აგრეთვე Integer ტიპის ცვლადი U, რომელიც აგრეთვე ინდივიდუალური სერვოდრავების მართვაში გამოიყენება და სხვა ოთხ დაფაშია გამოყენებული.

გარდა ამისა, ტეგების ზონაში შექმნილია აგრეთვე ბულის ტიპის სხვა ცვლადები, რომლებიც ინდივიდუალური სერვოდრავების მართვისათვის არიან გათვალისწინებული. ესენია uk1_1, uk1_2, uk1_3 პირველი სერვოდრავისათვის, uk2_1, uk2_2, uk2_3 - მეორე სერვოდრავისათვის, uk3_1, uk3_2, uk3_3 - მესამე სერვოდრავისათვის და uk4_1, uk4_2, uk4_3 - მეოთხე სერვოდრავისათვის.

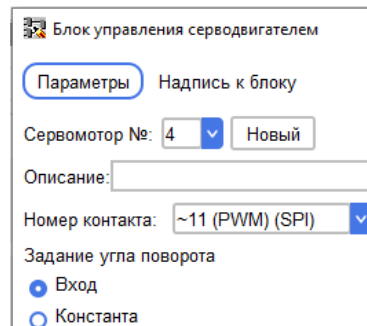
სერვოდრავების ინდივიდუალური მართვის პროგრამები ნაჩვენებია ფიგ. 23.2-ზე, სადაც პირველი სერვოდრავის მართვის პროგრამა განთავსებულია დაფა 2-ზე-Плата: 2, მეორის-დაფა 3-ზე, მესამის-დაფა 4-ზე, ხოლო მეოთხის-დაფა 5-ზე. ისინი ფუნქციონალურად არაფრით არ განსხვავდებიან ერთმანეთისაგან, უბრალოდ ინდივიდუალური ცვლადების დასახელებისა, აგრეთვე სერვოდრავების ნომერაციისა 1,2,3,4 და მათი არდუინოსთან შეერთების კონტაქტების ნომრებსა.



ფიგ. 23.2 სერვოდრავების ინდივიდუალური მართვის პროგრამები

ზემოთმოყვანილ ტეგების ზონაში არ არის ნაჩვენები სერვოდრავების შესასვლელების შეერთებები არდუინოს გამოსასვლელებთან, რადგან ისინი მითითებულია თვითონ სერვოდრავების პარამეტრიზაციის ფანჯრებში. კონკრეტულად ჩვენ პროექტში გამოყენებული

ოთხი სერვომძრავისთვის ისინი შეერთებული არიან არდუინოს მე-8 ... მე-11 კონტაქტებზე შესაბამისად. თვალსაჩინოებისათვის ფიგ. 23.3-ზე წარმოდგენილია პროექტის მეოთხე სერვომძრავის პარამეტრიზაციის ფანჯარა. დანარჩენი სერვომძრავების პარამეტრიზაცია უნდა მოხდეს ანალოგიურად.

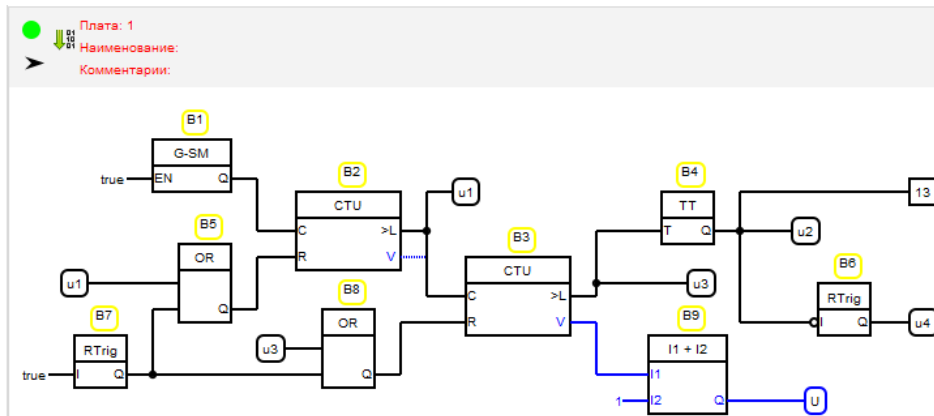


ფიგ. 23.3 მე-4 სერვომძრავის პარამეტრიზაციის ფანჯარა

ახლა გადავიდეთ პროექტის აღწერაზე. სერვომძრავების მართვის ალგორითმი ამ პროექტში მდგომარეობს იმაში, რომ ისინი მუშაობენ ორ რეჟიმში: დეტერმინებულ რეჟიმში და ფსევდომთხვევით რეჟიმში. პირველ რეჟიმში მუშაობისას სერვომძრავის A შესასვლელზე მიეწოდება ჯერ რიცხვების ზრდადი თანმიმდევრობა ნოლიდან 180-მდე რაღაც გარკვეული სიხშირით, შემდეგ კლებადი თანმიმდევრობა 180-დან ნოლამდე იმავე სიხშირით და ეს ხდება რამდენჯერმე, შემდეგ სიხშირე კლებულობს და აღწერილი პროცესი განმეორდება მაგრამ უკვე ნაკლები სიხშირით, და ა.შ.შ. საბოლოოდ ეს პროცესი განმეორდება ექვს სხვადასხვა სიხშირეებზე. ამ სიხშირეების იმპულსების ხანგრძლიობებია (4, 8, 16, 32, 64, 128) მლწ. ძრავის ღერძი ამ შემთხვევაში გადაადგილდება წინ და უკან დასაწყისში მაღალი სიჩქარით, შემდეგ სიჩქარე ორჯერ შენელებს, შემდეგ კიდევ ორჯერ შენელებს და ა.შ.შ სულ ექვს სხვადასხვა სიჩქარეზე იმუშავებს იგი. ამის შემდეგ მართვა გადადის ფსევდომთხვევითი მუშაობის რეჟიმში. აქაც ძრავის შესასვლელზე მიეწოდება იგივე სიხშირეები 4, 8, 16, 32, 64, 128, მაგრამ ეს სიხშირეები იცვლება არა ისეთივე თანმიმდევრობით როგორც ეს პირველ რეჟიმში იყო, არამედ შემთხვევით (რნდომ), მაგ. დასაწყისში შეიძლება იყოს 32, შემდეგ 4, შემდეგ 128 და ა.შ.შ. ამის შემდეგ კი მართვა გადავა ისევ პირველ-დეტერმინებულ რეჟიმში და ა.შ.შ.

გადავიდეთ ახლა სერვომძრავების საერთო მართვის პროგრამის აღწერაზე, ფიგ. 23.5. აქ ჩვენ გვაქვს იმპულსების გენერატორი B1, რომელიც მუშაობს სიმეტრიული მულტივიზორატორის რეჟიმში და გამოიმუშავებს იმპულსებს ხანგრძლივობით 6 მლწ. ეს იმპულსები ჯერ მიეწოდება პირველ მთვლელს B2, რომელიც მუშაობს ციკლური თვლის

რეჟიმში, ითვლის 360-მდე, შემდეგ დგება ნულზე u1 უკუკავშირის ცვლადის დახმარებით და შემდეგ იმეორებს ციკლს. აღნიშნული უკუკავშირი u1 მიეწოდება აგრეთვე მეორე მთვლელს B3, რომელიც ითვლის 6-მდე (გაიხსენეთ სიხშირეების რაოდენობა, რომელზეც მუშაობს სერვოდრავი) და შემდეგ დგება ნოლზე u3 უკუკავშირის ცვლადის დახმარებით და თავიდან იწყებს ახალ ციკლს. ეს მთვლელი აფორმირებს თავის V გამოსასვლელზე რიცხვებს 0, 1, 2, 3, 4, 5.



ფიგ. 23.5 სერვოდრავების საერთო მართვის პროგრამა

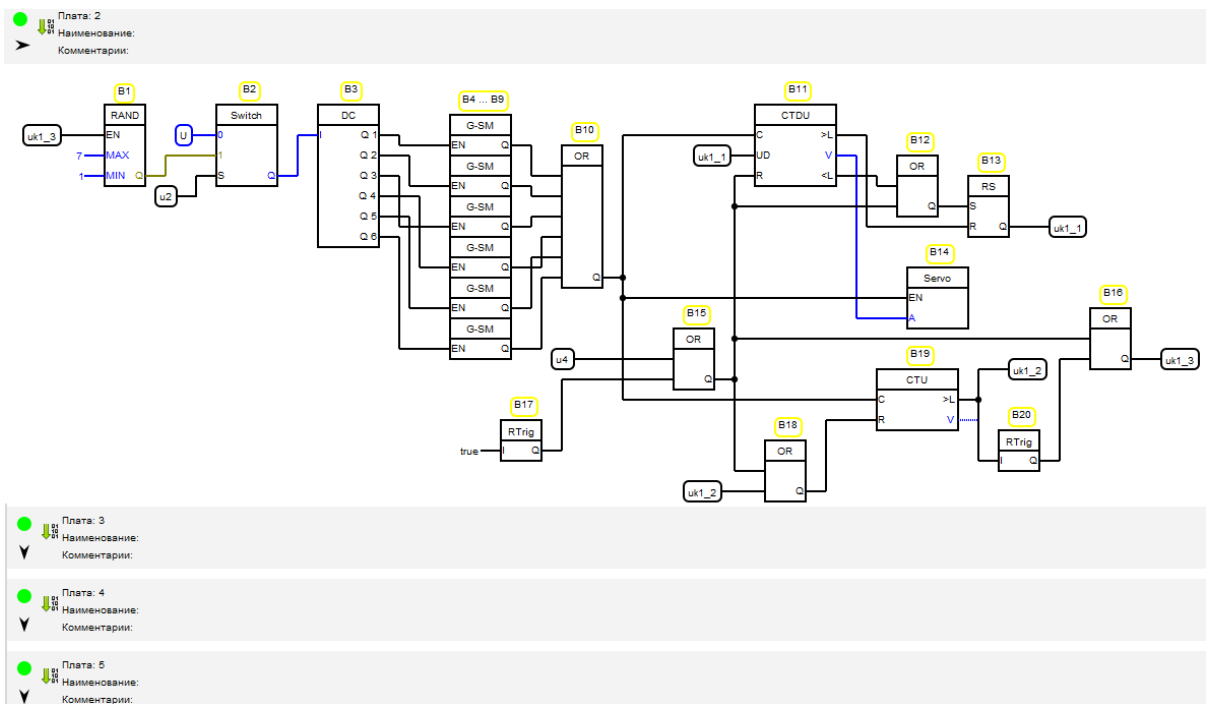
ჩვენ კი შემდეგ დაფებში დემიფრაციისათვის გვჭირდება რიცხვები 1, 2, 3, 4, 5, 6, ამიტომ შემოტანილი გვაქვს მათემატიკური შეკრების ბლოკი B9, სადაც მთვლელის მიერ ფორმირებულ რიცხვს ვუმატებთ ერთიანს და ტეგების ზონაში შექმნილი U ცვლადით გადავცევთ შემდეგ დაფებში.

ამავდროულად B3 მთვლელიდან მიღებული u3 სიგნალი აყენებს TT ტრიგერს ერთიანის მდგომარეობაში და გამოიმუშავებს u2 სიგნალს, რომელიც გადაეცემა შემდეგ ოთხ დაფას და რითაც მათში ხდება დეტერმინებული რეჟიმის შეცვლა ფსევდომთხვევითზე. ამავდროულად ეს სიგნალი მიეწოდება არდუინოს მე-13 კონტაქტს. აინთება არდუინოს დაფაზე ჩამონტაჟებული და მე-13 კონტაქტთან შეერთებული ჩამონტაჟებული შუქდიოდი რაც ვიზუალურად გვატყობინებს რომ რეჟიმი შეიცვალა და მიმდინარეობს სერვოდრავის მუშაობის სიჩქარეების ფსევდომთხვევითი ცვლილების რეჟიმი.

ჩვენს მიერ განხილულ ამ მართვის სქემას (ფიგ. 23.5) ჭირდება საწყის მომენტში ნოლზე დაყენება. ამ ფუნქციის შესასრულებლად გამოყენებული გვაქვს RTrig-B7, რომელიც არდუინოს ჩატვირთვის, გადატვირთვის, ანდა კვების მიწოდების შემთხვევაში გამოიმუშავებს პროგრამის ერთი ციკლის ხანგრძლიობის სიგნალებს, რომელიც B5 და B8 ლოგიკური შეკრების ბლოკების გავლით დააყენებს B2 და B3 მთვლელებს ნულოვან, საწყის

მდგომარეობაში. გარდა ამისა, მართვის ფსევდომემთხვევითი რეჟიმიდან ისევ დეტერმინებულ რეჟიმში დაბრუნების მომენტში ამუშავდება B6 RTrig, რომელიც u4 ცვლადის გავლით დააყენებს დანარჩენი დაფების სქემებს საწყის მდგომარეობაში.

ახლა დავუბრუნდეთ სერვერებისათვის ინდივიდუალური მართვის პროგრამების გარჩევას. ფიგ. 23.6-ზე გამოსახულია ერთერთი ბლოკის მართვის პროგრამა, დანარჩენის მართვის პროგრამები ანალოგიურია ფიგურაზე წარმოდგენილთან. როგორც უკვე აღნიშნული იყო, გამომდინარე მუშაობის ალგორითმიდან ძრავები მუშაობენ სიჩქარეების ან დეტერმინებული ან ფსევდომემთხვევითი შერჩევის პრინციპით. სიჩქარეების საერთო რაოდენობა ტოლია ექვსის. დეტერმინებული მართვის დროს ეს სიჩქარეები შეირჩევა მიმდევრობით: ჯერ პირველი სიჩქარე, მერე მე-2 და ა.შ. ამ რიცხვების ფსევდომემთხვევითი შერჩევისათვის გამოყენებული გვაქვს B1 ბლოკი RAND, რომელიც ყოველი uk1_3 სიგნალის დამთავრების შემდეგ თავის გამოსასვლელზე გამოიმუშავებს შემთხვევით რიცხვს ერთიდან ექვსამდე. თვალსაჩინოებისათვის ამ ნახაზს გავიმეორებთ ამ გვერდზეც.



ფიგ. 23.6 სერვოდრავების ინდივიდუალური მართვის პროგრამები

ამ რენდომ ტიპის ბლოკის B1 გამოსასვლელი შეერთებულია გადამრთველი ბლოკის B2 პირველ შესასვლელზე, რომლის ნულოვან შესასვლელზე მიერთებულია საერთო მართვის ბლოკის U ცვლადი, რომელზედაც როგორც უკვე აღვნიშნეთ, რიცხვების მიწოდება 1-დან 6-მდე ჩათვლით. რეჟიმებს შორის გადართვა ერთიდან მეორეზე, ხდება საერთო მართვის ბლოკში გამომუშავებული სიგნალით u2. როდესაც ეს სიგნალი ნულოვან მდგომარეობაშია,

მაშინ B2 ბლოკის გამოსასვლელზე გამოვა რიცხვები U შესასვლელიდან, ხოლო როცა ეს სიგნალი მიიღებს ერთიანის მდგომარეობას მაშინ B2 ბლოკის გამოსასვლელზე გამოვა რიცხვები მისი პირველი შესასვლელიდან, ანუ B1 რანდომ ბლოკიდან. B2 ბლოკის გამოსასვლელი შეერთებულია დემიფრატორის შესასვლელზე B3, რომელიც იმისდა მიხედვით, თუ რა რიცხვი მოვა მის შესასვლელზე 1-დან...6-მდე, გამოიმუშავებს ერთერთ მის გამოსასვლელზე შესაბამის Q1...Q6 ლოგიკურ ერთიანს, დანარჩენებზე კი ლოგიკურ ნოლს.

ლოგიკური ერთიანი დემიფრატორის გამოსასვლელიდან გაუშვებს შესაბამის სიმეტრიულ მულტივიბრატორს B4...B9, რომლებიც აწყობილია 2,4,8,16,32,64 მილიწამის ხანგრძლივობის იმპულსების გამომუშავებაზე და სწარედ ამით მიიღწევა სერვოდრავების სხვადასხვა სჩქარეებზე გადართვა. აქ ხაზი უნდა გაესვას იმას, რომ თუ დეტერმინებულ რეჟიმში ყველა სერვოდრავი მუშაობს სინქრონულად, ერთიდაიგივე სიხშირით, ფსევდოშემთხვევით რეჟიმში ისინი იმუშავებენ ასინქრონულად სხვადასხვა სიხშირეებზე, რადგან ჩვენ გვაქვს ოთხი ბლოკი B1 თვითოეული ძრავისათვის თავის საკუთარი. ამუშავებული გენერატორიდან გამომავალი სიგნალები B10 ლოგიკური შეკრების ბლოკის გავლით მიეწოდება რევერსიულ მთვლელს B11, რომელიც ითვლის ჯერ დადებითი თვლის რეჟიმში 180-მდე, შემდეგ უარყოფითი თვლის რეჟიმში ნოლამდე, შემდეგ ისევ დადებითი თვლის რეჟიმში და ა.შ.შ. ამ მთვლელის მართვა ხდება RS ტრიგერის გამოყენებით-ბლოკი B13, რომელიც საწყის მომენტში დგება ერთიანის მდგომარეობაში Rtrig ტრიგერის B17 სიგნალით, რომელიც ლოგიკური შეკრების ბლოკის B15 გავლით დააყენებს ამ მთვლელს B11 ნულოვან მდგომარეობაში, ხოლო B12 ლოგიკური შეკრების ბლოკის გავლით დააყენებს RS ტრიგერს ერთიანის მდგომარეობაში.

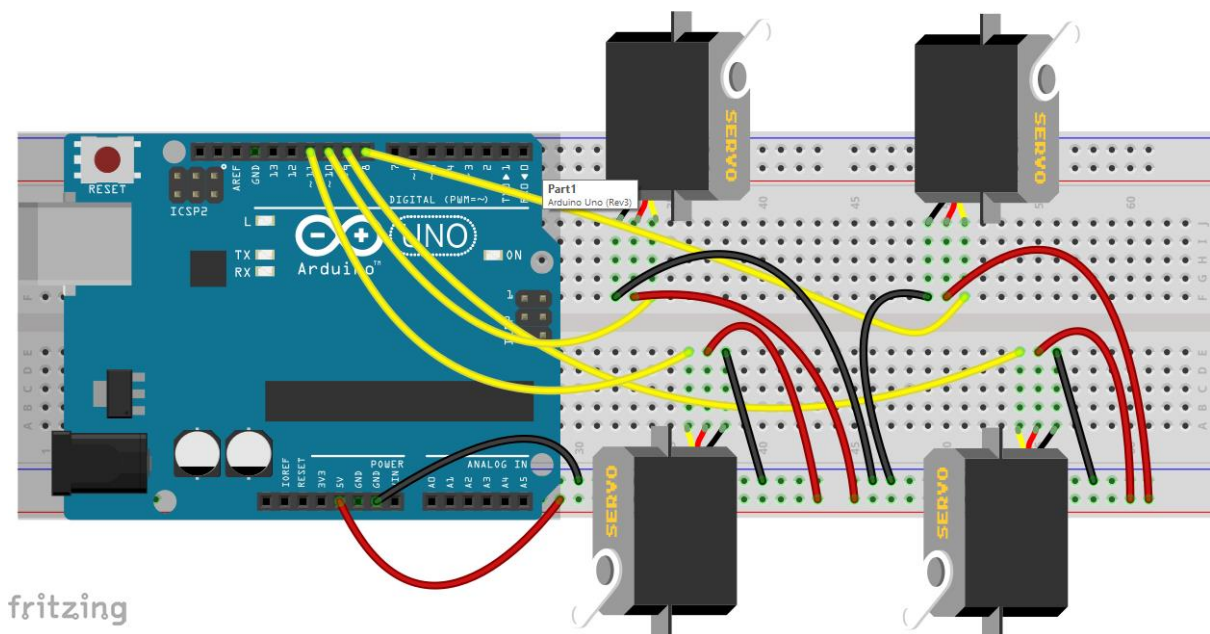
ლოგიკური ერთიანი ამ ტრიგერის გამოსასვლელიდან uk_1 ცვლადის გავლით მიეწოდება B11 მთვლელის UD შესასვლელს და დააყენებს მას დადებითი თვლის რეჟიმში. როდესაც ეს მთვლელი დაითვლის 180 იმპულსს მაშინ B11 მთვლელის გამოსასვლელზე გამომუშავდება >L სიგნალი, რომელიც დააყენებს ტრიგერს ნულოვან მდგომარეობაში და გადაიყვანს B11 მთვლელს გამოკლების რეჟიმში. ამის შემდეგ, როდესაც ეს მთვლელი ჩამოვა ნოლამდე მაშინ მის მიერ გამომუშავებული სიგნალი <L ლოგიკური შეკრების ბლოკის B12 გავლით ისევ დააყენებს RS ტრიგერს ერთიანის მდგომარეობაში, რაც ისევ გადაიყვანს B11 მთვლელს შეკრების რეჟიმში და ა.შ.შ.

B11 ბლოკის ანალოგური V გამოსასვლელი მიერთებულია სერვოდრავის ანალოგურ A შესასვლელზე და აყენებს მას იმ კუთხეზე, რა რიცხვიც არის მიწოდებული ამ შესასვლელზე.

ძრავის ამუშავებას კი ნებას რთავს ერთეულოვანი სიგნალი რომელიც მიეწოდება მის EN შესასვლელზე B10 ბლოკის გამოსასვლელიდან.

თვითონ ამ შემთხვევითი რიცხვის ცვლილება ხდება შემდეგნაირად. სიგნალი პროგრამის პირველი ციკლის შესრულებისას რომელიც გამომუშავდება B17 ბლოკში B15 და B18 ლოგიკური შეკრების ბლოკების გავლით აყენებს მთვლელს B19 ნულოვან მდგომარეობაში. ამის შემდეგ ეს მთვლელი იწყებს იმპულსების თვლას, რომლებიც მიეწოდება B10 ბლოკის გამოსასვლელიდან. იგი მთვლელი აწყობილია 100 იმპულსის თვლაზე. როდესაც მთვლელი დაითვლის 100-მდე გამომუშავდება სიგნალი uk1_2, რომელიც B18 ბლოკის გავლით ისევ დააყენებს ამ მთვლელს ნულოვან მდგომარეობაში და ა.შ.შ. იგივე სიგნალი uk1_2 ამუშავებს RTrig ტრიგერს, რომელიც ასევე გამოიმუშავებს იმპულსს პროგრამის ერთი ციკლის განმავლობაში და რომელიც B16 ლოგიკური შეკრების ბლოკის გავლით მიეწოდება B1 ბლოკის EN შესასვლელს, იგი დაიწყებს შემთხვევითი რიცხვების გენერირებას ნებადართული დიაპაზონიდან და EN შესასვლელზე იმპულსის დამთავრების შემდეგ ეს ბლოკი გამოსცევს ერთერთ რიცხვს დასაშვები დიაპაზონიდან.

აქვე უნდა აღინიშნოს, რომ იგივე პროცესი, ანუ შემთხვევითი რიცხვის გენერირების პროცესი ხდება მუშაობის რეჟიმის გადასვლის დროსაც ფსევდოშემთხვევითი რიცხვების გენერირების რეჟიმშიც. ეს კი ხდება u4 სიგნალის დახმარებით, რომელიც გამომუშავდება საერთო მართვის სქემაში. ეს სიგნალი B15 და B16 ლოგიკური შეკრების ბლოკების გავლით მიეწოდება B1 ბლოკის EN შესასვლელს. ოთხი სერვომძრავის მართვის Fritzing სქემა ასახულია ფიგ. 23.5-ზე.



ფიგ. 23.5 ოთხი სერვომძრავის მართვის Fritzing სქემა

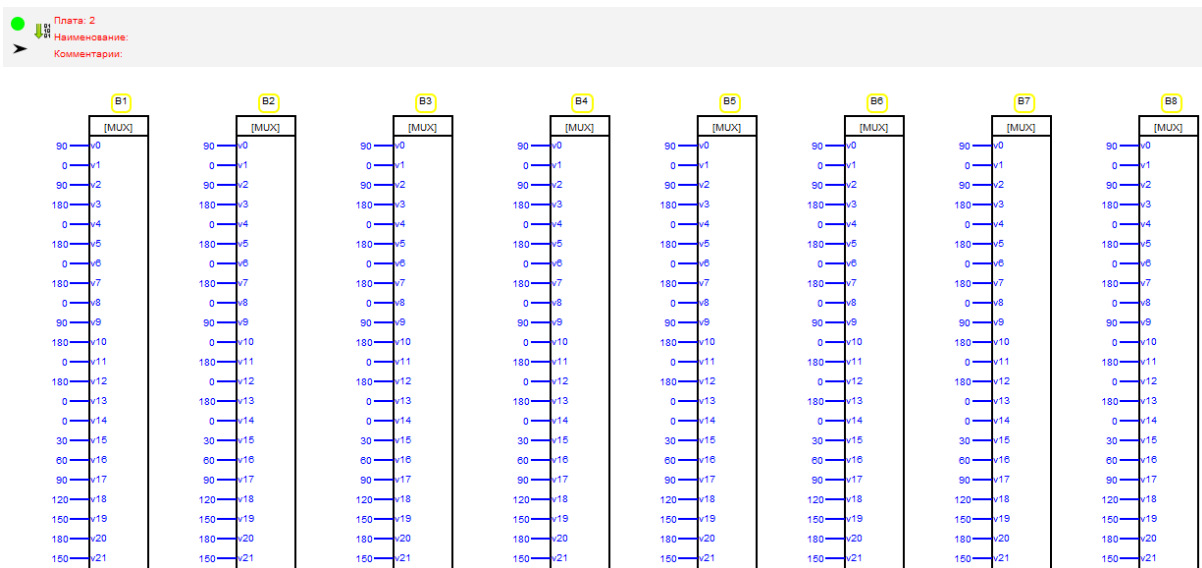
პროექტი_24

რვა სერვო ძრავის მართვა

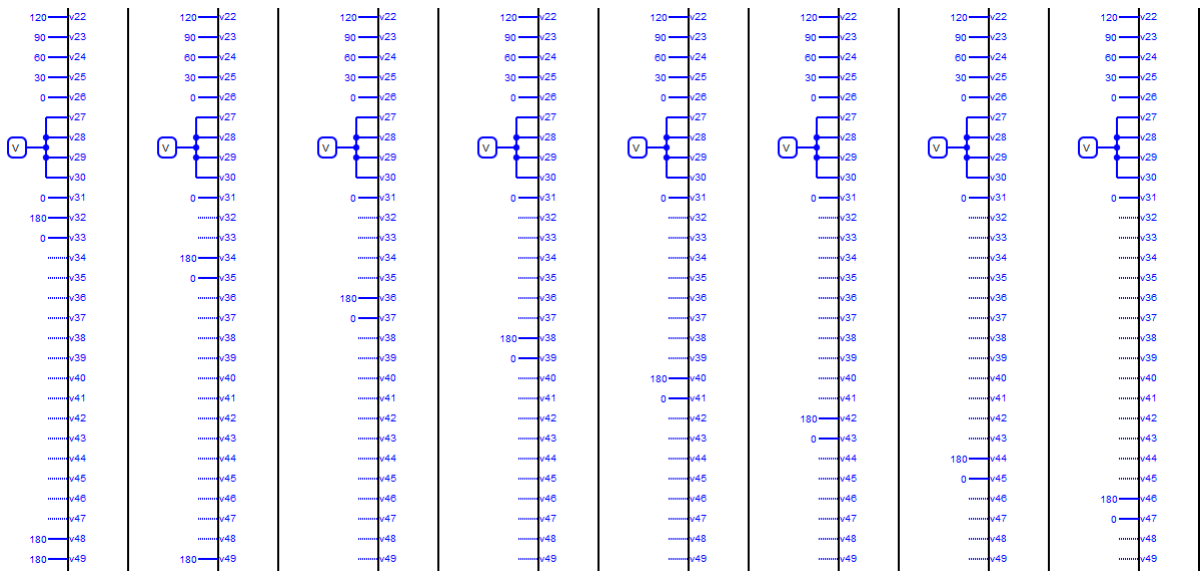


ამ პროექტში ჩვენ განვიხილავთ რვა სერვოდრავის მართვას, სადაც წინა პროექტისაგან განსხვავებით სერვოდრავის მობრუნების კუთხეები კი არ ფორმირდება მუშაობის პროცესში, არამედ ხდება მითითება, რომ ამათუ იმ პოზიციაზე ძრავის ლილვი უნდა დაყენდეს მოცემულ კუთხეზე. პროექტში ჩვენ პირობითად აღებული გვაქვს 64 პოზიცია, მაგრამ სხვა შემთხვევებში პოზიციები შეიძლება იყოს მეტი ან ნაკლები მუშაობის ალგორითმიდან გამომდინარე.

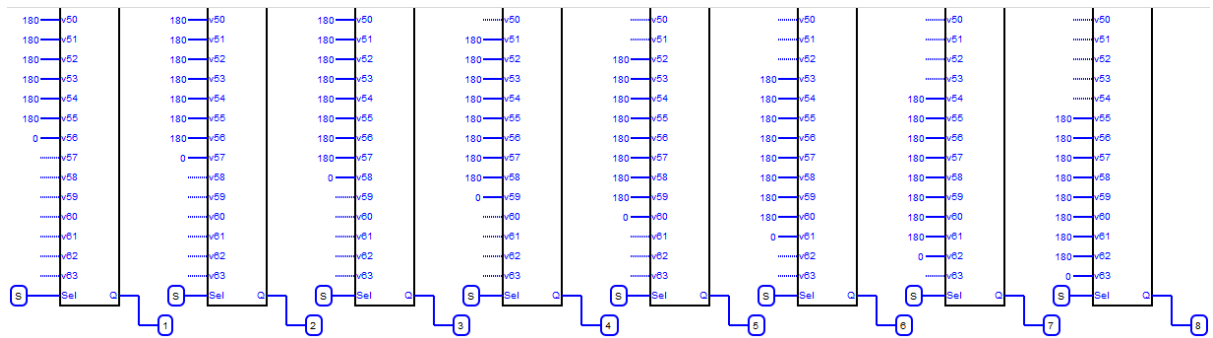
პოზიციების ფორმირებისათვის ჩვენ გამოყენებული გვაქვს მრავალშესასვლელიანი გადამრთველების, მულტიპლექსორების ბლოკი [MUX] რვა ერთეული, სამართავი სერვოდრავების რაოდენობის მიხედვით და თვითოეული ეს ბლოკი აწყობილია 64 შესასვლელზე, პოზიციები v0...v63. (ფიგ. 24.1.1...ფიგ. 24.1.3).



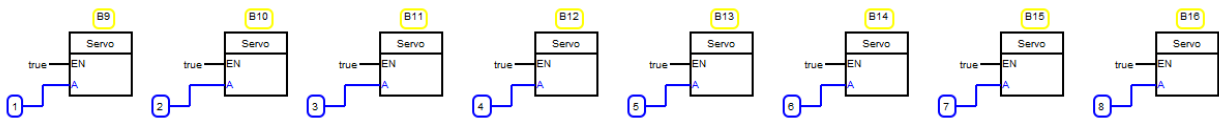
ფიგ. 24.1.1 რვა სერვოდრავის მართვის პროგრამა (1 ნაწილი)



ფიგ. 24.1.2 რვა სერვომძრავის მართვის პროგრამა (მე-2 ნაწილი)

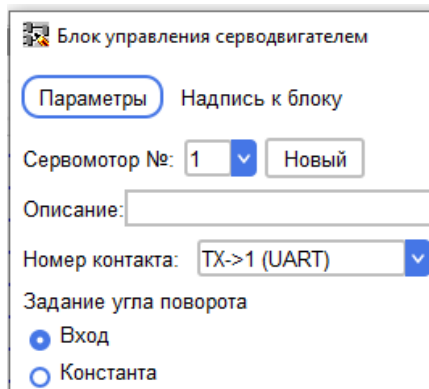


ფიგ. 24.1.3 რვა სერვომძრავის მართვის პროგრამა (მე-3 ნაწილი)



ფიგ. 24.1.4 რვა სერვომძრავის მართვის პროგრამა (მე-4 ნაწილი)

იმასთან დაკავშირებით, რომ 64 პოზიციანი [MUX] ბლოკები სიმალის მიხედვით კადრში ვერ ეტევა, ამიტომ ჩვენ მოგვიწია პროგრამის ვიზუალური ნახაზის ოთხ ნაწილად გაყოფა. პირველ ნაწილში წარმოდგენილია პოზიციები v0...v21, მეორეზე ნაწილში - v22... v49 (ფიგ. 24.1.2), მესამე ნაწილში - v50...v63 (ფიგ. 24.1.3), ხოლო მე-4 ნაწილში - თვითონ სამართავი სერვომძრავების ბლოკები.



ფიგ. 24.2 პირველი სერვომძრავის პარამეტრიზაციის ფანჯარა

უნდა აღინიშნოს, რომ სერვომძრავების შესაერთებელი კონტაქტები, ჩვეულებრივად, მიეთითება არა ტეგების ზონაში, არამედ აირჩევა თვითონ ამ სერვომძრავების ბლოკების პარამეტრიზაციის ფანჯრებში (იხ. პირველი სერვომძრავის პარამეტრიზაციის ფანჯარა ფიგ. 24.2) საიდანაც ჩანს, რომ პირველი სერვომძრავის მართვის შესასვლი მიერთებულია არდუინოს პირველ კონტაქტთან, დანაჩენი სერვომძრავების პარამეტრიზაცია სრულდება ანალოგიურად და შეერთებულნი არიან არდუინოს 2, 3, 4, 5, 6, 7, 8 კონტაქტებზე შესაბამისად.

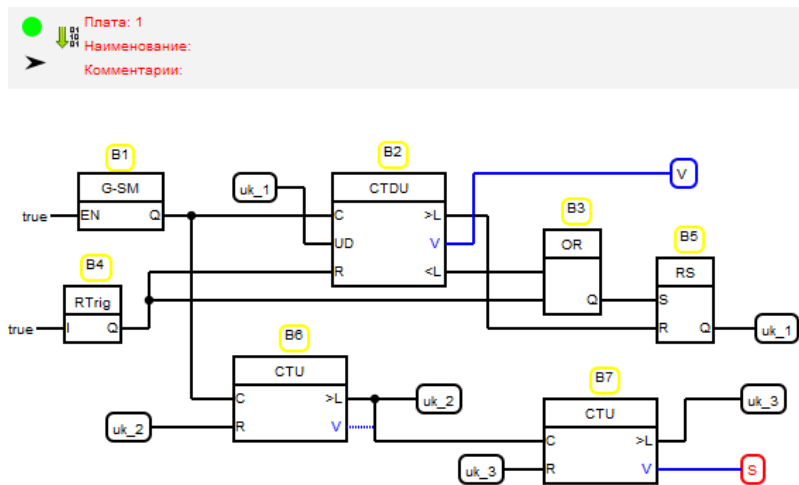
ახლა ავხსნათ თუ რას აღნიშნავს [MUX] ბლოკების შესასვლელებზე მითითებული რცხვები ფიგ. 24.1.1. წინა პროექტებიდან ვიცით, რომ რა რიცხვის კოდიც არის მიწოდებული ამ ბლოკის sel შესასვლელზე, იმის შესაბამისი v შესასვლელია არჩეული და მისგან ხდება რიცხვის გადაცემა Q გამოსასვლელზე. მუშაობის პროცესში sel შესასვლელზე მიეწოდება რიცხვების თანმიმდევრობა ზრდადობის მიხედვით, ეს ნიშნავს რომ ამ შესასვლელზე რიცხვები იცვლება 0-დან 63 თანდათანობით ზრდადობით, ანუ ამ ბლოკებზე ჯერ ხდება v0 შესასვლელების არჩევა, შემდეგ v1 და ა.შ.შ - v63-მდე, შემდეგ პროცესი იწყება თავიდან.

დავუშვად sel შესასვლელზე მიეწოდება ნოლი, ე.ი ყველა [MUX] ბლოკზე არჩეულია v0 შესასვლელი, ამ შესასვლელზე კი ყველგან აწერია რიცხვი 90, ეს სიდიდე გამოვა ამ ბლოკების გამოსასვლელებზე და მიეწოდება სერვომძრავების მართვის შესასვლელებს, რაც გამოიწვევს ამ ყველა სერვომძრავის ღერძებს დაყენებას 90 გრადუსის მდგომარეობაში.

შემდეგ sel შესასვლელებზე ჩაიწერა ერთიანი. ე.ი. ყველა ბლოკზე აირჩევა v1 პოზიცია, ამ პოზიციაზე ჩაწერილია ნოლები და ყველა სერვომძრავების ღერძები დადგება ნოლოვან მდგომარეობაში, ანუ ნულოვან პოზიციაზე ყველა სერვომძრავი დგება 90 გრადუსის მდგომარეობაში, v1 პოზიციაში კი ყველა სერვომძრავი მობრუნდა და დადგა ნულ გრადუსის მდგომარეობაში. v2 პოზიციაზე შესასვლელებს აწერიათ 90, ანუ ყველა სერვომძრავი ისევ დადგება 90 გრადუსის მდგომარეობაში, v4 პოზიციაზე ყველა სერვერი ისევ დადგება

ნულოვან მდგომარეობაში, v5 პოზიციაზე ყველა სერვერი დადგება ახლა უკვე 180 გრადუსის მდგომარეობაში, v6-ზე - ისევ ნულოვან პოზიციაში, v7-ზე-ისევ 180 გრადუსზე, v8-ზე-ისევ ნულოვან მდგომარეობაში, v9-ზე-ისევ 90 გრადუსის მდგომარეობაში. როგორც ვხედავთ v10 პოზიციამდე ყველა სერვერი მუშაობდა ერთნაირად, სინქრონულად, ამის შემდეგ კი იწყება სერვოდრავების ასინქრონული მუშაობა.

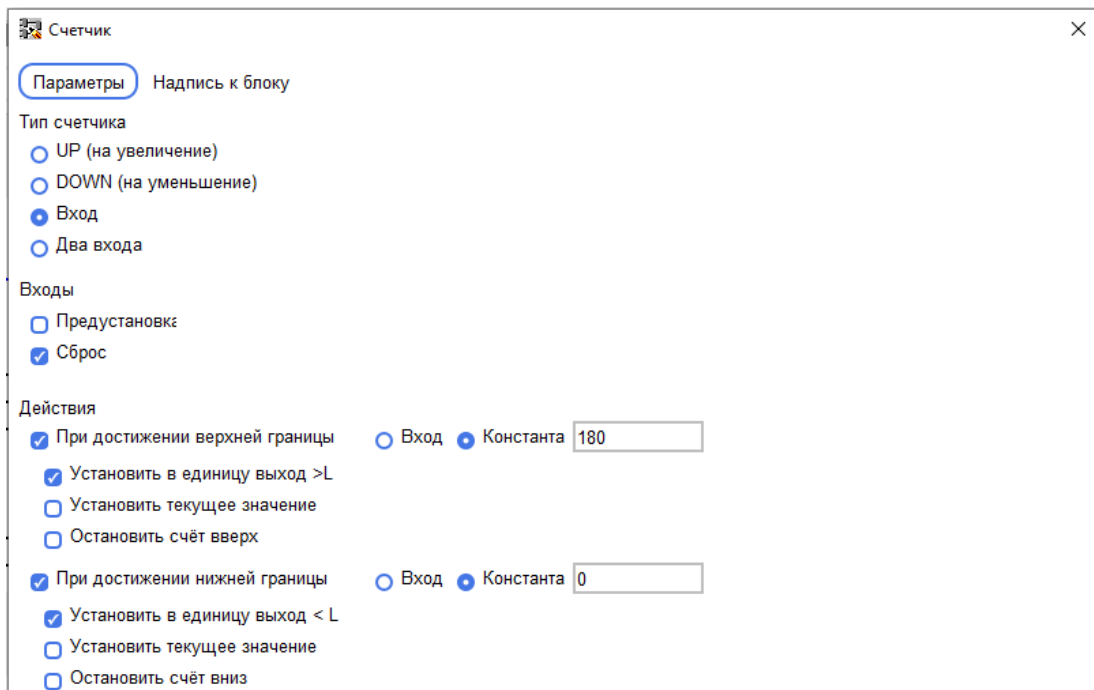
v10 პოზიციაზე 1, 3, 5, 7 სერვოდრავები დადგება 180 გრადუსის მდგომარეობაში, ხოლო 2,4,6,8 - ნული გრადუსის მდგომარეობაში, v11 კი პირიქით, 1, 3, 5, 7 სერვოდრავები დადგება ნული გრადუსის მდგომარეობაში, ხოლო 2,4,6,8 -180 გრადუსის მდგომარეობაში, v12 პოზიციაზე ისევ გამეორდება v10 პოზიციის მდგომარეობა და ა.შ.შ ასე ფორმირდება სერვოდრავების სინქრონულ-ასინქრონული მუშაობა არა მარტო ნულ და 180 გრადუსზე არამედ 30 გრადუსზე, 60 გრადუსზე, 90 გრადუსზე, 120 გრადუსზე, 150 გრადუსზე და 180 გრადუსებზე (იხილით და გაანალიზეთ ყველა პოზიცია, გარდა v27...v30 პოზიციებისა, ფიგ. 24_1_1...ფიგ. 24_1_3). ამ პოზიციებზე კი ყველა სერვოდრავი მუშაობს სინქრონულად, რომელთა მუშაობის პრინციპს გავეცნობით საერთო მართვის სისტემის აღწერის კვალდაკვალ. იგი განთავსებულია პირველ დაფაზე (ფიგ. 24.3).



ფიგ. 24.3 რვა ძრავის საერთო მართვის სისტემა

მთელი სისტემის მართვისთვის გამოყენებული გვაქვს გენერატორი B1, რომელიც მუშაობს სიმეტრიული მულტივიბრატორის რეჟიმში იმპულსის ხანგრძლიობით 10 მილიწამი. მართვის სქემის საწყის მდგომარეობაში მოყვანა ხდება Rtrig-ის გამოყენებით-B4, რომელიც გამოიმუშავებს პროგრამის ერთი ციკლის ხანგრძლიობის იმპულსს არდუინოს ჩატვირთვის მომენტში. ეს იმპულსი აყენებს მთვლელს B2-ნულოვან მდგომარეობაში და ლოგიკური შეკრების ბლიკის B3-ის გავლით - RS ტრიგერს ერთიანის მდგომარეობაში. ტრიგერის

გამოსასვლელი uk_1 უკუკავშირის ცვლადის დახმარებით მიეწოდება B2 მთვლელის UD შესასვლელზე და აყენებს მას დადებითი თვლის რეჟიმში. მთვლელის პარამეტრიზაციის ფანჯარიდან (ფიგ. 24.4 ჩანს), რომ როდესაც მთვლელი იმპულსების თვლის პროცესში მიაღწევს 180, მაშინ გამომუშავდება ლოგ. „1“ მის >L გამოსასვლელზე, რომელიც დააყენებს RS ტრიგერს ნოლიანის მდგომარეობაში და ახლა უკვე ნულოვანი სიგნალი მისი გამოსასვლელიდან, რომელიც მიეწოდება B2 მთვლელის UD შესასვლელზე გადაიყვანს მას უარყოფითი თვლის რეჟიმში და მთვლელი დაიწყებს მუშაობას კლებაზე. ამის შემდეგ, როდესაც მთვლელი მიაღწევს ნულოვან მდგომარეობას, მაშინ მის <L გამომუშავდება ლოგ. „1“, რომელიც B3 ლოგიკური შეკრების ბლოკის გავლით ისევ დააყენებს RS ერთიანის მდგომარეობაში, მთვლელი (ახლა უკვე რევერსული მთვლელი) გაიმეორებს პროცესს და ა.შ.შ. მუშაობა გაგრძელდება დაუსრულებლად.



ფიგ. 24.4 რევერსული მთვლელის პარამეტრიზაციის ფანჯარა

სწორედ რომ ამ B2 რევერსული მთვლელის ანალოგური გამოსასვლელი, რომელიც აღნიშნული გვაქვს V ცვლადის სახით მიეწოდება [MUX] ბლოკების v27, v28, v29 და v30 პოზიციებს. ანუ, როდესაც ამ პოზიციებზე გადავა სისტემა, მაშინ სერვოძრავები დაიწყებენ მდორე გადაადგილებას 0 გრადუსიდან 180 გრადუსამდე, შემდეგ ისევ ნოლ გრადუსამდე და ა.შ.შ. მანამ არ გადავა სისტემა v31 პოზიციაზე.

პოზიციების სათვლელად კი გამოყენებული გვაქვს ორი დამატებითი მთვლელი B6 და B7, რომელთაგან პირველი ითვლის ციკლურად 50-მდე uk_2 ცვლადის დახმარებით და რომლის გამოსასვლელზეც ვღებულობთ იმპულსებს ყოველ $10 \cdot 50 = 500$ მილიწამში. სწორედ ამ სიხშირით ხდება პოზიციების გადართვა [MUX] ბლოკების შესასვლელებზე, ხოლო ამ პოზიციების დასათვლელად კი გამოიყენება მეორე მთვლელი B7, რომელიც ითვლის 64-მდე და შემდეგ uk_3 ცვლადის დახმარებით ბრუნდება საწყის მდგომარეობაში და ასე ასრულებს ციკლური თვლის რეჯიმს. სწორედ, რომ ამ მეორე მთვლელის ანალოგური გამოსასვლელი S მიეწოდება [MUX] ბლოკების შესასვლელებზე და ხორციელდება ამ ბლოკების პოზიციების ცვლილება. პროექტზე მუშაობის დროს საჭიროა ტეგების ზონის გამოყენება ცვლადების შესაქმნელად, რომელიც ასახულია გვაქვს ფიგ. 24_5-ზე. ამავე ზონაშია შექმნილი და ასახული Integer ტიპის 1, 2, 3, 4, 5, 6, 7, 8 ცვლადები, რომლებიც გამოიყენება [MUX] ბლოკების დაკავშირებისათვის სერვოდრავებთან.

რვა სერვოდრავის მართვის Fritzing სქემა ასახულია ფიგ. 24_6-ზე. აქ უნდა გვახსოვდეს, რომ რვა სერვოდრავის კვება არდუინოდან დაუშვებელია, საჭიროა დამატებითი კვების წყარო, სადაც მათ კვების წყაროზე საჭირო იქნება სამი-ოთხი ფილტრის ელექტროლიტური კონდენსატორი არანაკლებ 100 მკფ/16v ტევადობის თვითოეული.

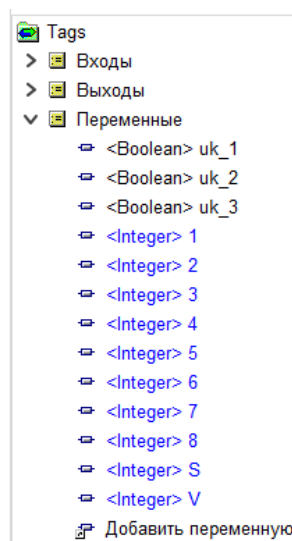
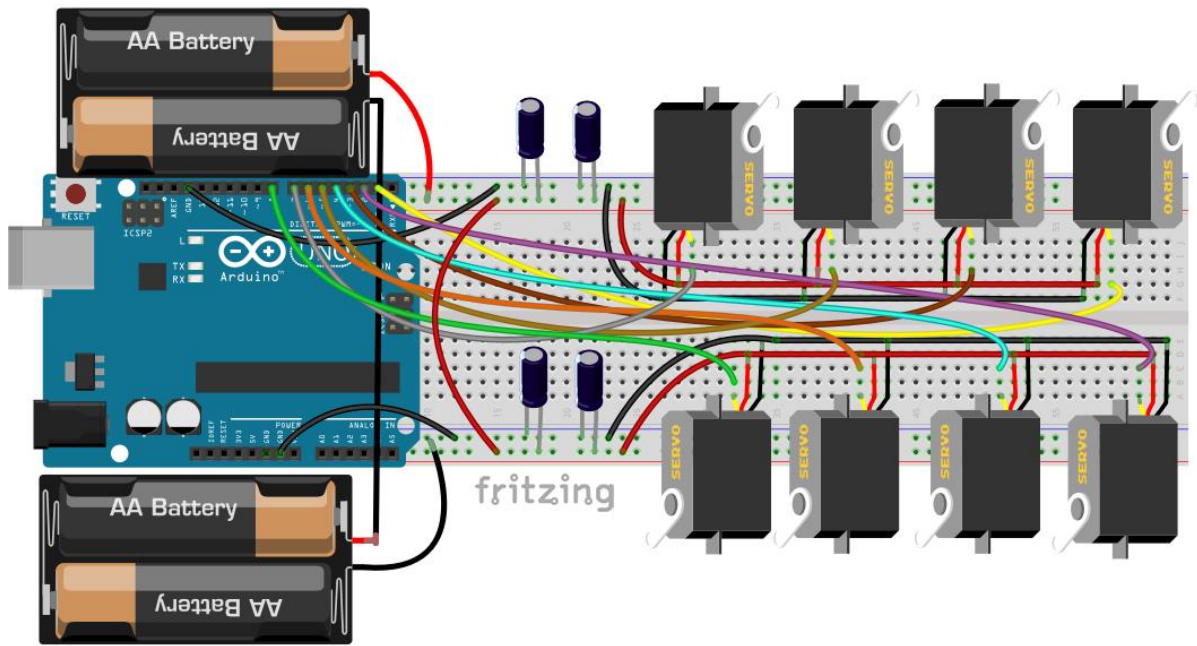


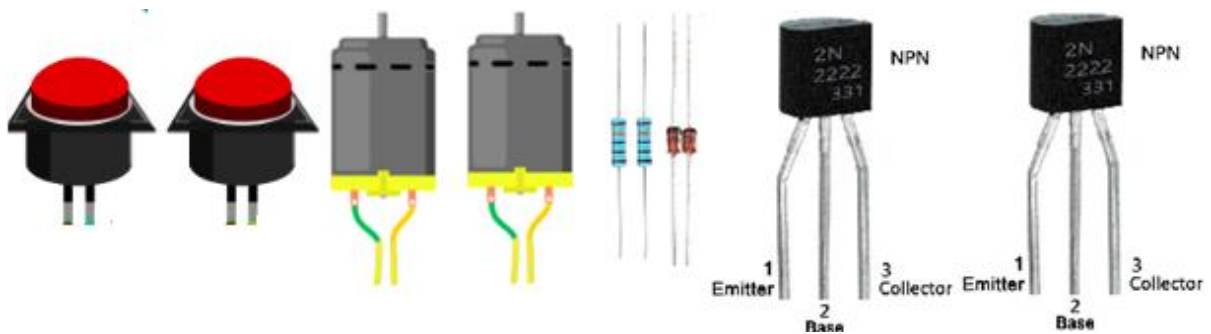
Fig. 24.5 ცვლადები ტეგების ზონაში



ფიგ. 24.6 რვა სერვომძრავის მართვის Fritzing სქემა

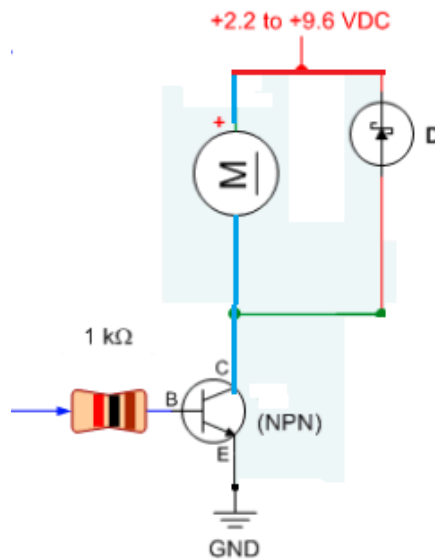
პროექტი_25

ორი მუდმივი ძრავის პირდაპირი მართვა



მუდმივი დენის ძრავების მართვისთვის სასურველია გამოყენებულ იქნას ან ტრანზისტორული მართვა, ან სპეციალური ძრავის დრაივერებით მართვა, განვიხილოთ მუდმივი დენის ძრავების ტრანზისტორული მეთოდი. პროექტში განხილული იქნება ყველაზე მარტივი ფორმა ძრავის მართვის, პერიოდული ჩართვა და გამორთვა. ამასთან გამოვიყენებთ ორ ძრავს, რომელსაც ვამუშავებთ სხვადასხვა სიხშირით. პირველი ძრავი ორი წამის განმავლობაში ჩართული იქნება და ორი წამის განმავლობაში გამორთული, მეორე ძრავი კი ჩართული იქნება ოთხი წამის განმავლობაში და გამორთული ასევე ოთხი წამის განმავლობაში.

ძრავების ჩართვა გამორთვისთვის გამოყენებული გვექნება ორი დილაკი, ერთი დილაკი, ჩართავს პირველ ძრავს, ხოლო მეორე-მეორე ძრავს. ელექტრული სქემა ერთერთი ძრავის შეერთებისა ტრანზისტორთან წარმოდგენილია ფიგ. 25.1-ზე.



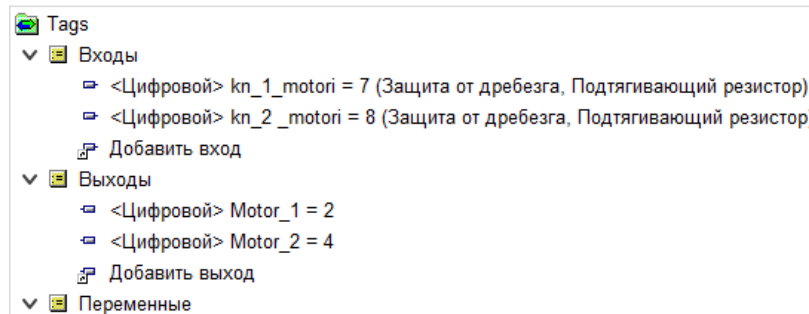
ფიგ. 25.1 ძრავის მართვის ტრანზისტორული სქემა

აქ ჩვენ გამოყენებული გვაქვს მცირე სიმძლავრის npn ტიპის ტრანზისტორი 2N2222, რომელიც ფართოდ გამოიყენება არდუინოზე დაფუძნებულ სქემებში. მუდმივი დენის ძრავების კვებისათვის გამოყენებულ უნდა იქნას გარეშე კვების წყარო 2.2 ვოლტიდან 9.2 ვოლტამდე. არდუინოს გამოსასვლელი უერთდება ტრანზისტორის ბაზას B სქემაზე ნაჩვენები 1 კომ წინააღმდეგობის გავლით, მაგრამ შეიძლება გამოყენებულ იქნას ნებისმიერი ნომინალის წინააღმდეგობა 220 ომიდან 1 კომ-მდე.

ტრანზისტორის ემიტერი უნდა შეერთდეს არდუინოსა და გარე კვების წყაროს გაერთიანებულ მიწასთან, ხოლო კოლექტორი - მუდმივი დენის ძრავის ერთერთ გამომყვანთან, რომლის მეორე გამომყვანი შეერთებული უნდა იყოს გარე კვების წყაროს პლიუსთან. ამავე პლიუსზე უნდა შეერთებულ იქნას აგრეთვე ზენერის დიოდის პლიუს გამომყვანი, ხოლო მისივე მინუს გამომყვანი-ტრანზისტორის კოლექტორთან. ამ დიოდის დანიშნულებაა დაიცვას ტრანზისტორი ძრავზე კვების მოხსნის მომენტში მის გრაგნილზე წარმოშობილი მაღალი ძაბვის ნახტომისაგან. ეს ძაბვა მიიღევა ძრავის გრაგნილის წინააღმდეგობაზე დიოდის გავლით და შესაბამისად ვერ გამოიყვანს ტრანზისტორს მწყობრიდან.

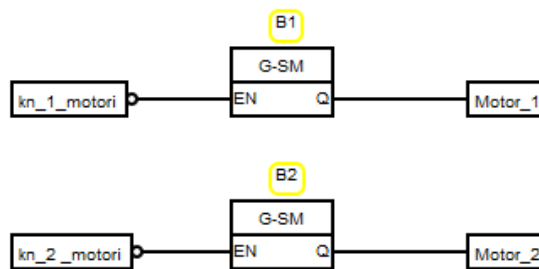
პროექტში გამოყენებული ორი ძრავის FLProg პროგრამა ძალიან მარტივია. შექმნილია ორი შესასვლელი, kn_1_motori და kn_2_motori, რომლებზეც უნდა მიუერთდეს მართვის გარე

ლილაკები და ორი გამოსასვლელი, Motor_1 და Motor_2 (ფიგ. 25.2), რომლებიც უნდა შეერთდეს ტრანზისტორების ბაზებთან შესაბამისი წინააღმდეგობების გავლით.



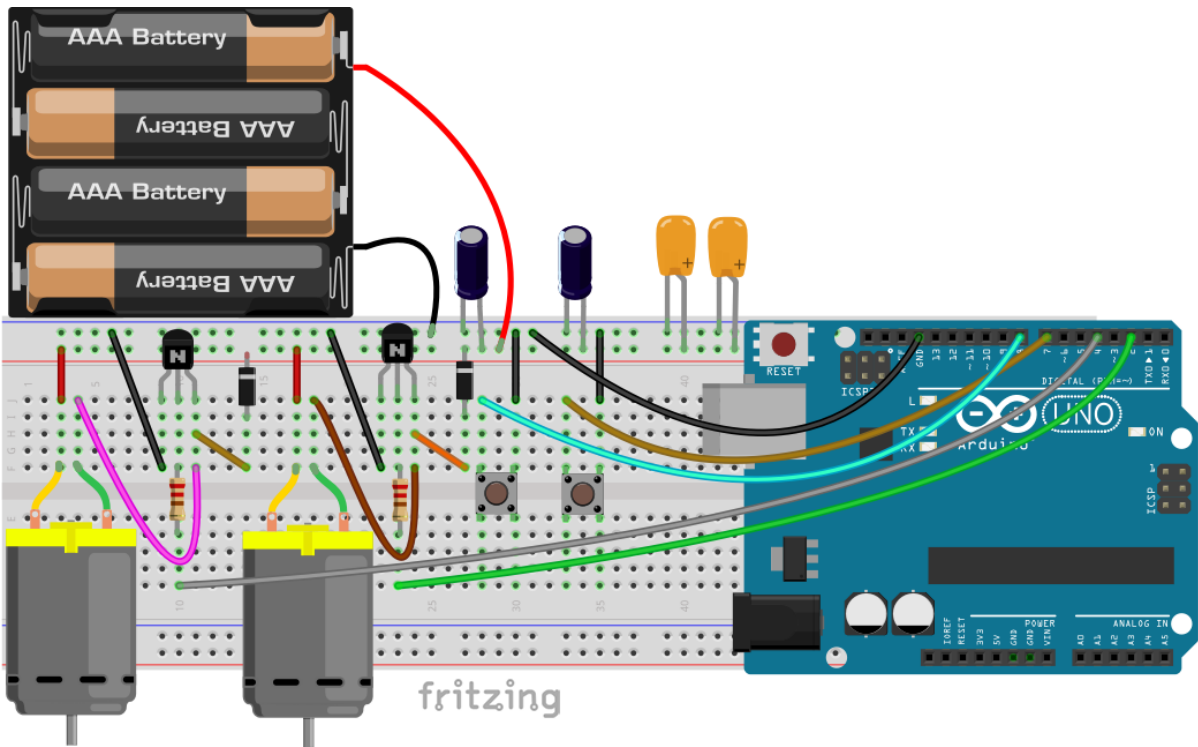
ფიგ. 25.2 ტეგების ზონა

პროგრამაში ღილაკები შეერთებულნი არიან ორ იმპულსების გენერატორების EN შესასვლელთან ფიგ. 25.3.



ფიგ. 25.3 მუდმივი დენის ძრავების მართვის პროგრამა

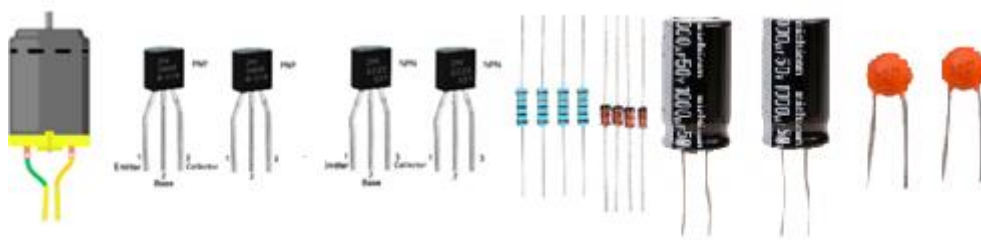
პირველ ღილაკზე თითის დაჭერა გაუშვებს B1 გენერატორს, რომელიც მუშაობს სიმეტრიული მულტივიბრატორის რეჟიმში 2 წამის ხანგრძლიობის იმპულსით. ეს იმპულსი გახსნის ტრანზისტორს და ძრავი ამუშავდება, ხოლო პაუზა ჩაკეტავს ტრანზისტორს და ძრავი შეჩერდება, რაც იმას ნიშნავს, რომ ძრავი მუშაობს 2 წამის განმავლობაში და გაჩერებულია ასევე 2 წამის განმავლობაში და ა.შ. ასევე მუშაობს მეორე ღილაკზე თითის დაჭერა, იმ განსხვავებით, რომ მეორე ძრავი იმუშავებს 4 წამის განმავლობაში და გაჩერებული იქნება ასევე ოთხი წამის განმავლობაში. მუდმივი დენის ძრავების არდუინოსთან და ელექტრონულ კომპონენტებთან შეერთების Fritzing სქემა ნაჩვენებია ფიგ. 25.4-ზე.



ფიგ. 25.4 მუდმივი დენის ძრავების არდუინოსთან და ელექტრონულ კომპონენტებთან შეერთების Fritzing სქემა

პროექტი_26

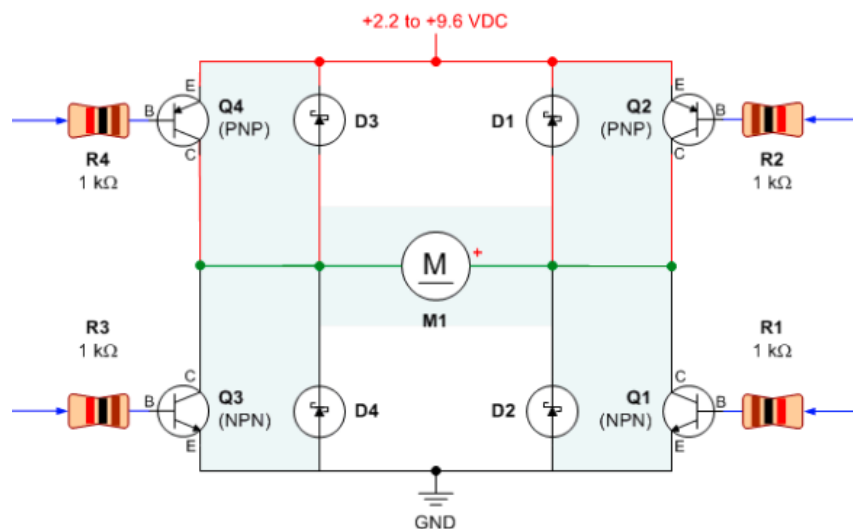
ერთი მუდმივი ძრავის რევერსული მართვა



მუდმივი დენის ძრავების რევერსული მართვისათვის ძირითადად გამოიყენება ტრანზისტორული ან მიკროსქემური მართვა. ტრანზისტორული მართვა შეიძლება განხორციელდეს როგორც ბიპოლარული ტრანზისტორების ასევე ველიანი ტრანზისტორების გამოყენებით. ბიპოლარული ტრანზისტორების გამოყენების შემთხვევაში შესაძლებელია როგორც ერთნაირი გამტარობის ისე სხვადასხვა გამტარობის ტრანზისტორების გამოყენება. ჩვენ არდუინოს სტარტერ კიტიში აღმოჩნდა სხვადასხვა გამტარობის ტრანზისტორები: NPN გამტარობის 2N2222 და PNP გამტარობის 2N3906 ტიპის

ტრანზისტორები, ამიტომ ჩვენს მართვაში გადავწყვიტეთ გამოგვეყენებინა ორივე ამ ტიპის ტრანზისტორები.

რევერსული რეჟიმის უზრუნველყოფისათვის ძირითადად გამოიყენება ოთხტრანზისტორული მართვა, ეგრეთ წოდებული H ბოგირები (სადაც ძრავი ჩართულია ბოგირის დიაგონალში, ბოგირზე მოდებულია ძრავის მკვებავი ძაბვა, მუშაობის პროცესში გაიხსნება ხან ერთი მხარი წყვილი ტრანზისტორების, ხან მეორე. დენი გაივლის ძრავში ხან ერთი მიმართულებით, ხან მეორე მიმართულებით, რითაც მიიღწევა რევერსული ბრუნვა მუდმივი დენის ძრავისა. კონკრეტულად, ჩვენ ბოგირულ სქემაზე NPN ტიპის ტრანზისტორები განლაგებულნი არიან ბოგირის ქვედა მხრებში (Q1, Q3), ხოლო PNP ტიპის ტრანზისტორები - ბოგირის ზედა მხრებში (Q2, Q4) (იხ. ფიგ. 26.1). ამ უკანასკნელი ტრანზისტორების ემიტერები გაერთიანებულია და მათზე მოდებული უნდა იყოს დადებითი ძაბვა (2,2 ... 9,6) ვოლტამდე. გაერთიანებულია აგრეთვე Q1, Q3 ტიპის ტრანზისტორების ემიტერები და მათზე მოდებულ უნდა იქნას ძაბვის ნულოვანი პოტენციალი, რომელიც ელექტრულად შეერთებული უნდა იყოს აგრეთვე არდუინოს ნულოვან პოტენციალთან GND.



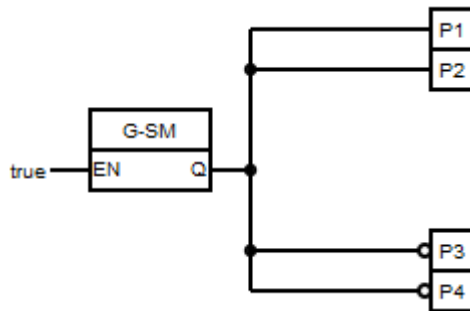
ფიგ. 26.1 ტრანზისტორული ბოგირული სქემა

Q1 და Q2, ტრანზისტორების კოლექტორები შეერთებულნი არიან ელექტრულად ერთმანეთთან და უერთდებიან მუდმივი ძრავის ერთერთ გამომყვანს, ისევე როგორც Q3 და Q4 ტრანზისტორების კოლექტორები შეერთებულნი არიან ელექტრულად ერთმანეთთან და უერთდებიან მუდმივი ძრავის მეორე გამომყვანს. ამას გარდა ყველა ტრანზისტორის კოლექტორ-ემიტერის გამომყვანებზე პარალელურად ჩართულია ზენერის დიოდები D1...D4,

რომელთა დანიშნულებათა დაიცვას ტრანზისტორები ძრავის მიმართულებების გადართვა გადმორთვის დროს მის გამომყვანებზე დაინდუქციურებული მაღალი ძაბვებისაგან.

H ბოგირის მუშაობის პრინციპი მარტივია, როდესაც R1 და R2 წინააღმდეგობების შესასვლელზე ლოგიკური ერთიანი დაჯდება, ხოლო R3 და R4 წინააღმდეგობების შესასვლელზე ლოგიკური ნულიანი, მაშინ გაიხსნება Q1 და Q4 ტრანზისტორები, ხოლო Q2 და Q3 ტრანზისტორები ჩაიკეტება. ამ შემთხვევაში ძრავის მარცხენა გამომყვანზე დაჯდება დადებითი პოტენციალი მარჯვენა გამომყვანზე-უარყოფითი პოტენციალი და ძრავი დაიწყებს მოზრაობას საათის ისრის მიმართულებით ან საწინააღმდეგოთ. ხოლო როცა R1 და R2 წინააღმდეგობების შესასვლელზე ლოგიკური ნულია, ხოლო R3 და R4 წინააღმდეგობების შესასვლელზე ლოგიკური ერთიანი, მაშინ გაიხსნება Q2 და Q3 ტრანზისტორები, ხოლო Q1 და Q4 ტრანზისტორები ჩაიკეტება. ამ შემთხვევაში ძრავის მარცხენა გამომყვანზე დაჯდება უარყოფითი პოტენციალი მარჯვენა გამომყვანზე კი დადებითი პოტენციალი და ძრავი დაიწყებს მოძრაობას საათის ისრის საპირისპირო მიმართულებით,

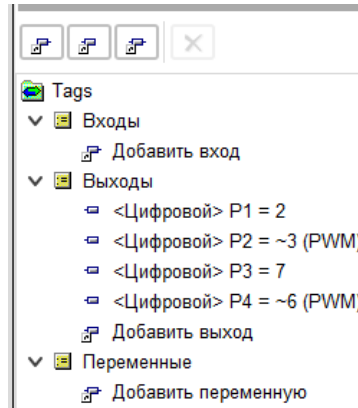
მუდმივი დენის ძრავის მართვის პროგრამა მარტივად შეიძლება განხორციელდეს. ამისათვის დაგჭირდება მხოლოდ ერთი გენერატორის ბლოკი, რომელიც აწყობილია სიმეტრიულ მულტივიბრატორად მუშაობის რეჟიმში იმპულსის ხანგრძლიობით 4 წმ, და ოთხი დისკრეტული გამოსასვლელი P1, P2, P3 და P4 (ფიგ. 26.2).



ფიგ. 26.2 მუდმივი დენის ძრავის რევერსული მართვის პროგრამა

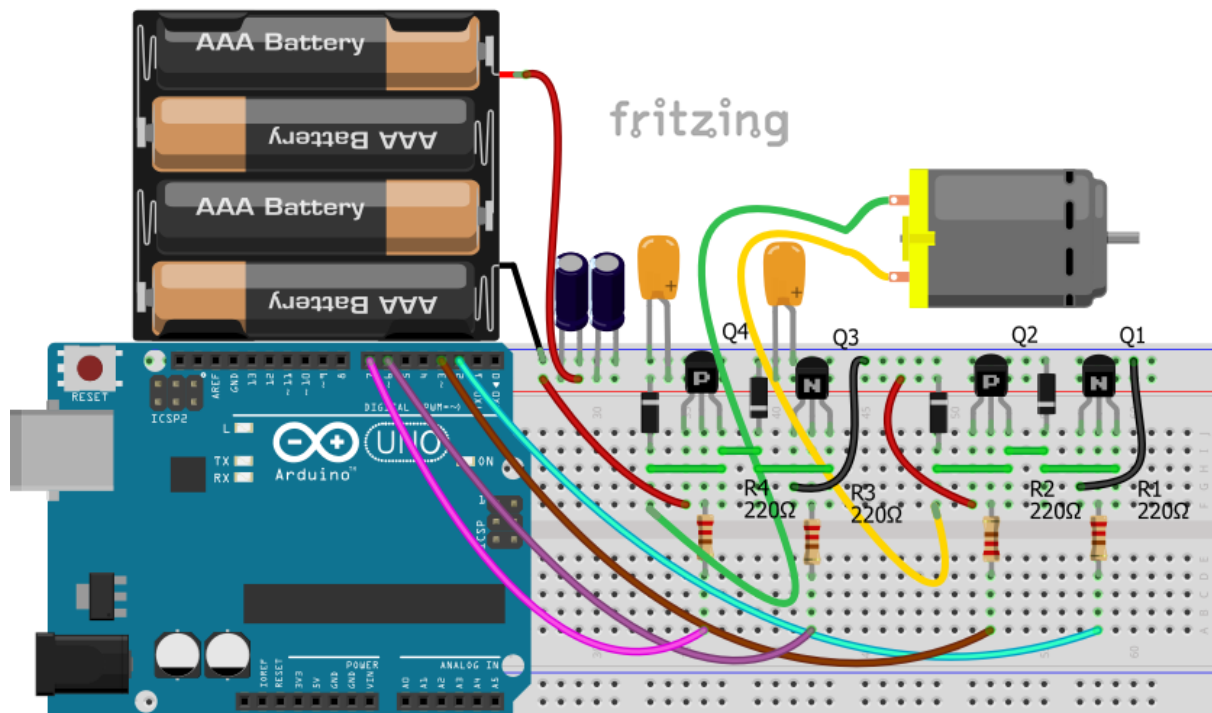
პროგრამა მუშაობს შემდეგნაირად გენერატორის გამოსასვლელზე, როდესაც მასზედ ლოგიკური ერთი ოთხი წამის განმავლობაში, მაშინ R1 და R2 წინააღმდეგობებზე მიეწოდება მაღალი პოტენციალი, ხოლო R3 და R4 წინააღმდეგობებზე-ნულოვანი პოტენციალი. ეს ნიშნავს რომ გაიხსნება Q1 და Q4 ტრანზისტორები და ძრავი დაიწყებს საათის ისრის მომართულებით მოზრაობას, ხოლო როდესაც გენერატორის გამოსასვლელზე დაჯდება პაუზა და ისიც გაგრძელდება 4 წამის განმავლობაში, მაშინ R1 და R2 წინააღმდეგობებზე

მიწოდება დაბალი პოტენციალი, ხოლო R3 და R4 წინააღმდეგობებზე-მაღალი. შესაბამისად ამისა, გაიხსნება Q2 და Q3 ტრანზისტორები და ძრავი დაიწყებს საათის ისრის მომართულებით მოძრაობას. პროექტის ტეგების ზონა ასახულია ფიგ. 26.3-ზე.



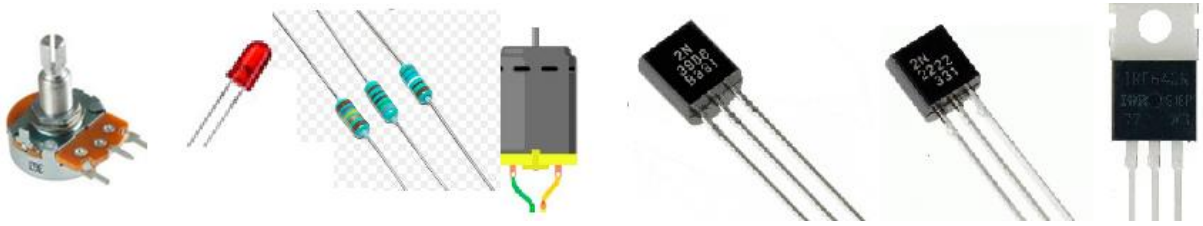
ფიგ. 26.3 ტეგების ზონა

მუდმივი დენის ძრავის რევერსული მართვის Fritzing სქემა ასახულია ფიგ. 26.4 - ზე.



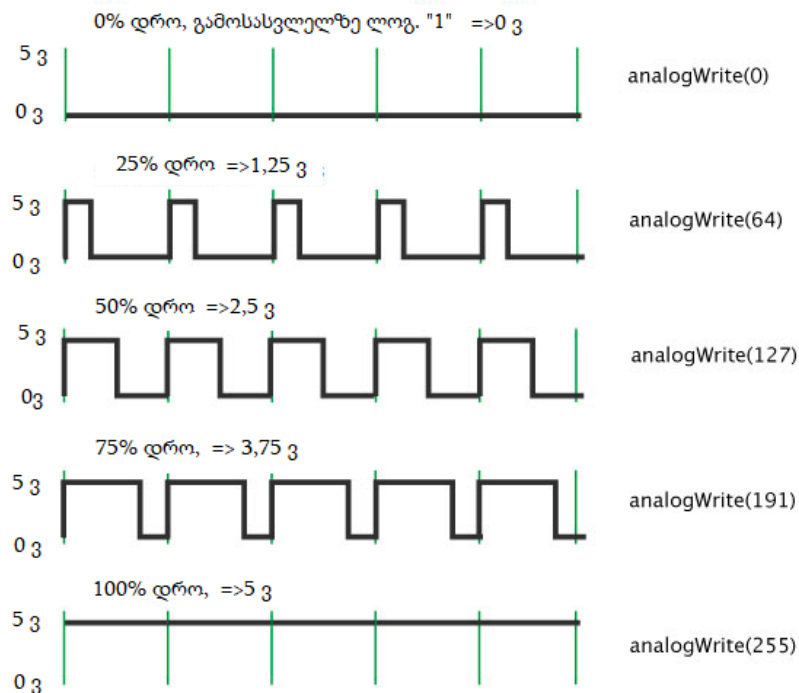
ფიგ. 26.4 მუდმივი დენის ძრავის რევერსული მართვის Fritzing სქემა

პროექტი_27



განივ იმპულსური მოდულიაცია (გიმ)

იმპულსის სიგანის მოდულიაცია (გიმ) ხასიათდება იმით, რომ იმპულსის სიგანე წარმოადგენს მკაცრად სწორკუთხოვანი ფორმის გრაფიკს. ეს მეთოდი ფართოდ გამოიყენება მუდმივი დენის ძრავების მართვაში. იგი მნიშვნელოვნად აუმჯობესებს მათ მართვას, რადგან მუდმივი დენის განუწყვეტელი სიგნალის ნაცვლად აქ გამოიყენება იმპულსების სერიით მართვა. იმპულსებმა შეიძლება მიაღწიონ მაქსიმალურ ძაბვას, მაგრამ წამების ნაწილების განმავლობაში. განმეორებითი ქმედებების შესრულებას მივყავართ იქამდე, რომ ძრავი იწყებს ბრუნვით მოძრაობას. ეს კი საშვალებას გვაძლევს უფრო კომფორტულად ვმართოთ მისი ბრუნვის სიჩქარე. განვიიმპულსური მოდულიაციის მოქმედების პრინციპი ასახულია ფიგ. 27.1-ზე.



ფიგ. 27.1 განვიიმპულსური მოდულიაციის პრინციპი

თუ რაღაც პერიოდში იმპულსის სიგანე პაუზასთან შედარებით არის 0%, მაშინ DC ძრავის გამომყვანებზე მიწოდებული ძაბვა არის => 0 ვოლტის ექვივალენტური და ძრავი არ

ბრუნავს, თუ კი იმპულსის სიგანეს გავხდით 25% მაშინ ძრავს მიეწოდება ექვივალენტური => 1,25 ვოლტი ძაბვა (აქ იგულისხმება, რომ ძრავის კვების წყაროდ გამოყენებულია 5 ვოლტიანი კვება) და ა.შ.შ, თუ იმპულსის სიგანეს ავიღებთ 100%, მაშინ ძრავს მიეწოდება მაქსიმალური ძაბვა => 5 ვოლტი.

თუ კარგად დავაკვირდებით, აღმოვაჩინებთ რომ გიმ სიგნალის სიხშირე ყველა პროცენტზე ერთნაირია, იცვლება მხოლოდ იმპულსის შევსების კოეფიციენტი პერიოდში, მაგ. 50%-ის შემთხვევაში იმპულსის სიდიდე და პაუზის სიდიდე უტოლდება ერთმანეთს. რაც უფრო იზრდება შევსების კოეფიციენტის პროცენტი მით უფრო იზრდება იმპულსის სიგანე, ხოლო პაუზის სიგანე პროპორციულად მცირდება. ამის შესაბამისად ჩვენ უნდა გვქონდეს სისტემაში რაიმე მექანიზმი, რომლის საშუალებითაც შეგვეძლებსა გავზომოთ იმპულსების და პაუზების პროცენტული მახასიათებლები.

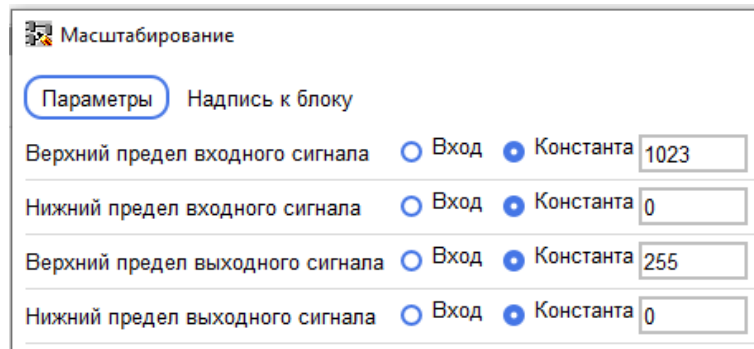
არდუინოში სისტემური გიმ-ის თანრიგაინობა მკაცრად არის დადგენილი და იგი შეადგენს 8 ბიტს. ამის შესაბამისად გიმ სიგნალი აქ შეიძლება იცვლებოდეს მინიმუმიდან ანუ 0-დან მაქსიმუმამდე, ანუ 255-მდე.

გამომდინარე ყოველივე ზემოთქმულიდან, ჩვენ ამ პროექტში შემოგთავაზებთ: შუქდიოდის სიკაშკაშის მდორე ცვლილებისა და DC ძრავის ბრუნვის სიჩქარის მდორე ცვლილების პროგრამებს 10 კომ ცვალებადი რეზისტორის გამოყენებით (ფიგ. 27_2).

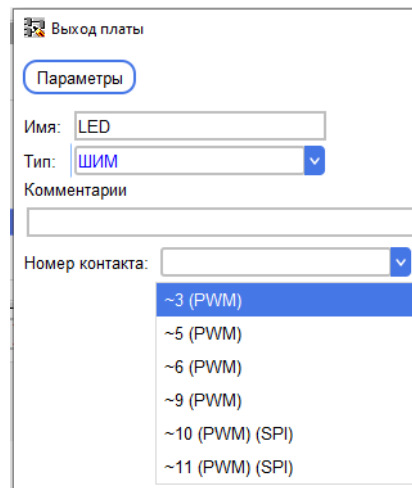
ამისათვის ტეგების ზონაში შექმნილი გვაქვს არდუინოს ერთი ანალოგური შესასვლელი A0, რომელზეც მიერთებულია ცვლადი რეზისტორის შუალედური გამომყვანი, განაპირა გამომყვანები შეერთებულია არდუინოს GND და 5v კონტაქტებთან. როგორც ცნობილია ეს შესასვლელი არდუინოს შიგნით მიერთებულია ათთანრიგაინ ანალოგურ-ციფრულ გარამქმნელთან და იგი ახდენს მონაცემის გამომუშავებას 0-დან 1024-მდე დიაპაზონში, მაშინ როდესაც ანალოგურ შესასვლელზე სიგნალი იცვლება ნოლიდან ხუთ ვოლტამდე. ჩვენ კი გვჭირდება, რომ როდესაც ანალოგურ შესასვლელზე სიგნალი იცვლება ნოლიდან ხუთ ვოლტამდე, ანუ ანალოგურ-ციფრული გარდაქმნის შემდეგ 0-დან 1023-მდე, მაშინ არდუინოში გამომუშავდეს ამის შესაბამისი პროპორციული გიმ სიგნალი 0-დან 255-მდე. ამას ჩვენ მივაღწევთ მასშტაბირების ბლოკის გამოყენებით B1, რომლის პარამეტრიზაციის ფანჯარაც წარმოდგენილი იქნება ფიგ. 27_2-ზე.

ამის შემდეგ უნდა შეიქმნას ერთი ციფრული გამოსასვლელი სახელწოდებით LED და მიექცეს ყურადღება იმას, რომ ეს გამოსასვლელი აუცილებლად უნდა იყოს გიმ ტიპის გამოსასვლელი, ასეთებია 3,5,6,9,10,11. ჩვენ შემთხვევაში არჩეულ იქნა მე-3 ციფრული

გამოსასვლელი, რომელიც უზრუნველყოფს გიმ სიგნალების გამომუშავებას შესასვლელზე მიწოდებული სიგნალის შესაბამისი შევსებულობით (ფიგ. 27_3).

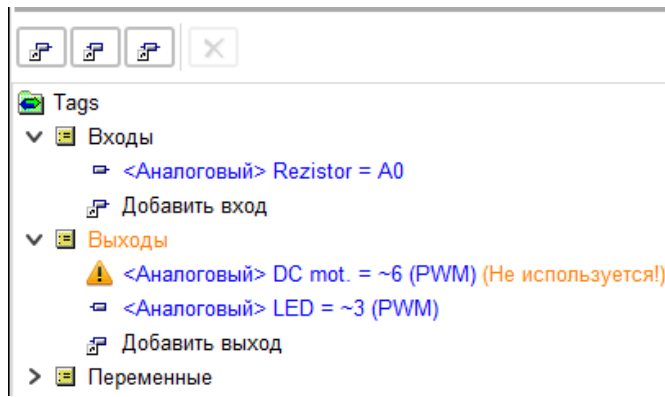


ფიგ. 27.2 მასშტაბირების ბლოკის პარამეტრიზაციის ფანჯარა

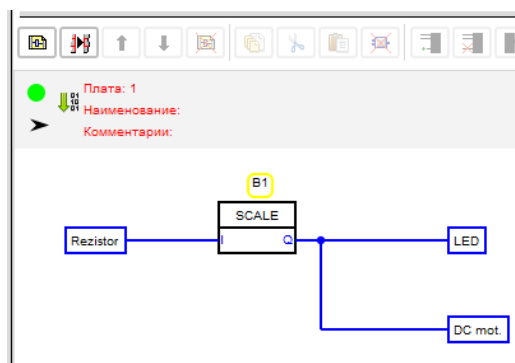


ფიგ. 27.3 გიმ ტიპის კონტაქტის შექნმა LED შუქდიოდის მართვისთვის

ეს გამოსასვლელი შემდეგ, მუდმივი წინააღმდეგობის გავლით უნდა შეუერთდეს შუქდიოდის ანოდს, რომლის კათოდის შეერთებულა არდუინოს GND კონტაქტთან. შევქმნათ კიდევ ერთი გიმ გამოსასვლელი DC mot მუდმივი დენის მოტორის მართვისთვის ჩამოვიტანოთ იგი პროექტის სამუშაო ზონაში და შევუერთოთ იგივე მასშტაბირების ბლოკის გამოსასვლელს (ფიგ. 27.5).

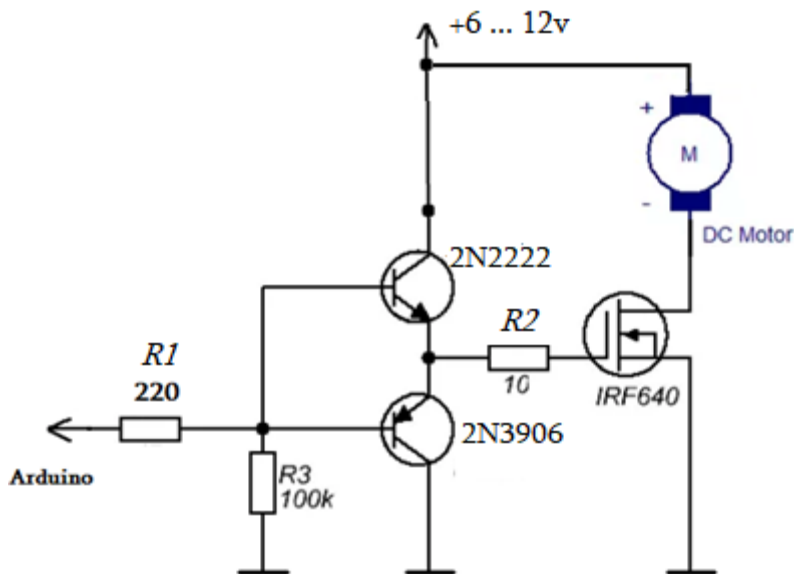


ფიგ. 27.4 გიმ ტიპის კონტაქტის შექმნა DC Motor-ის მართვისთვის



ფიგ. 27.5 შუქდიოდისა და მუდმივი დენის ძრავის მდორე მართვის პროგრამა

DC ძრავის მართვისთვის გამოყენებული გვაქვს ველიანი ტრანზისტორული მართვის ერთერთი ფართოდ გავრცელებული სქემა (ფიგ. 27.6).

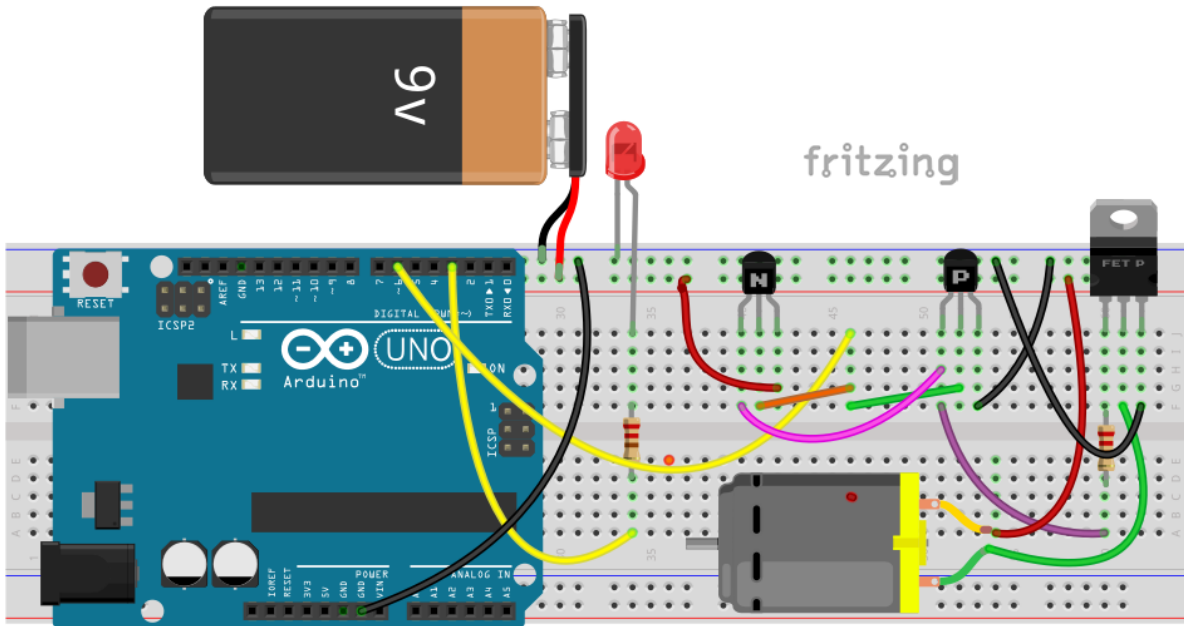


ფიგ. 27.6 DC ძრავის სქემა, ველიან ტრანზისტორის გამოყენებით

არდუინოს გამოსასვლელი R1 წინააღმდეგობის გავლით შეერთებულია ორი სხვადასხვა გამტარობის მქონე ბაზებით გაერთიანებულ pnp და npn ტრანზისტორების ბაზებზე, ამ ტრანზისტორების ემიტერებიც გაერთიანებულია და R2 წინააღმდეგობის გავლით შეერთებულია ველიანი ტრანზისტორის (MOSFET) საკეტონ. გარდა ამისა, დარჩენილი კოლექტორებიდან 2N3906 ტიპის ტრანზისტორის კოლექტორი შეერთებულია მიწასთან GND, ხოლო 2N2222 ტიპის ტრანზისტორის კოლექტორი - პლიუს ძაბვასთან. ველიანი ტრანზისტორის ჩასადინარი შეერთებულია მიწასთან, ხოლო სათავე DC ძრავის გავლით - პლიუს ძაბვასთან.

ჩავტვირთოთ პროგრამა არდუინოში და დავაკვირდეთ შუქდიოდის სიკაშკაშეს და ძრავის მოძრაობის სიჩქარეს. მოვატრიალოთ და დავაყენოთ ცვლადი წინააღმდეგობის ცოცია ერთერთ ნაპირა მდგომასრეობაში, შემდეგ იქიდან ნელნელა გადავიყვანოთ მეორე ნაპირა მდგომარეობაში. დავინახავთ, რომ შუქდიოდის სიკაშკაშე იცვლება, ან იზრდება ან კლებულობს ნელნელა მდორედ, ისევე როგორც ძრავის სიჩქარე ან იზრდება, ან კლებულობს ნელნელა მდორედ. არ დაგვავიწყდეს, რომ ცვლადი წინააღმდეგობის განაპირა გამომყვანები მიერთებულნი არიან მიწასთან GND და 5 ვოლტთან შესაბამისად. ცოციას ბრუნვით მის შუალედურ გამოსასვლელზე იცვლება მდორედ ძაბვა 0 ვოლტიდან 5 ვოლტამდე, ეს ანალოგური სიგნალი მიეწოდება არდუინოს ანალოგურ შესასვლელს A0-ს, სადაც ხდება ამ სიგნალის გარდაქმნა 0-დან 1023-მდე. ამის შემდეგ, მასშტაბირების ბლოკი მოახდენს ამ სიგნალის გარდაქმნას შესაბამისად 0-დან 255-მდე, რითაც მივიღებთ გიმ სიგნალებს. ეს გიმ სიგნალები არდუინოს გიმ გამოსასვლელებში გარდაიქმნება შესაბამის იმპულსების თანმიმდევრობად, რომლებიც მიეწოდება შუქდიოდსა და ძრავს და ახდენს მათ სიკაშკაშისა და სიჩქარის მდორე რეგულირებას.

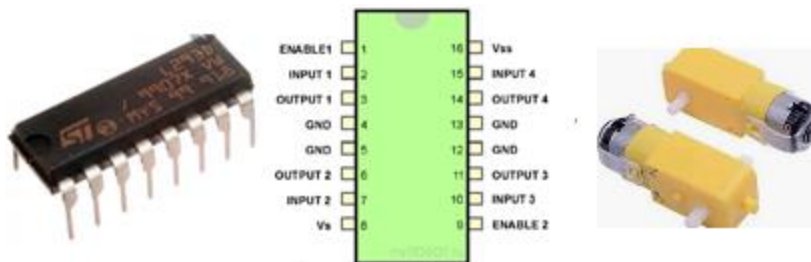
შუქდიოდისა და ძრავის გიმ მართვის Fritzing სქემა ასახულია ფიგ. 27.7-ზე.



ფიგ. 27.7 შუქდიოდისა და DC ძრავის გიმ მართვის Fritzing სქემა

პროექტი_28

ორი მუდმივი დენის ძრავის მიკროსქემული მართვა



ამ პროექტში ჩვენ გამოვიყენებთ ძრავების დრაივერის მიკროსქემას L293D-ს. რობოტებისა და მცირე ავტომატიკის ძრავების მართვისათვის, აუცილებელია ისეთი მოწყობილობა, რომელსაც შეეძლება მცირე სიმძლავრის მმართველი სიგნალების გარდაქმნა ძრავების მუშაობისათვის საკმარის დენის ძალებად. ასეთ მოწყობილობებს ეწოდებათ ძრავების დრაივერები. არსებობს საკმაო რაოდენობის სხვადასხვა სქემები, რომლებიც გამოიყენება ელექტროძრავების მართვისათვის. ისინი განსხვავდებიან ერთმანეთისაგან როგორც სიმძლავრით, ისე იმ ელემენტური ბაზით, რაზედაც ისინი არიან აგებული.

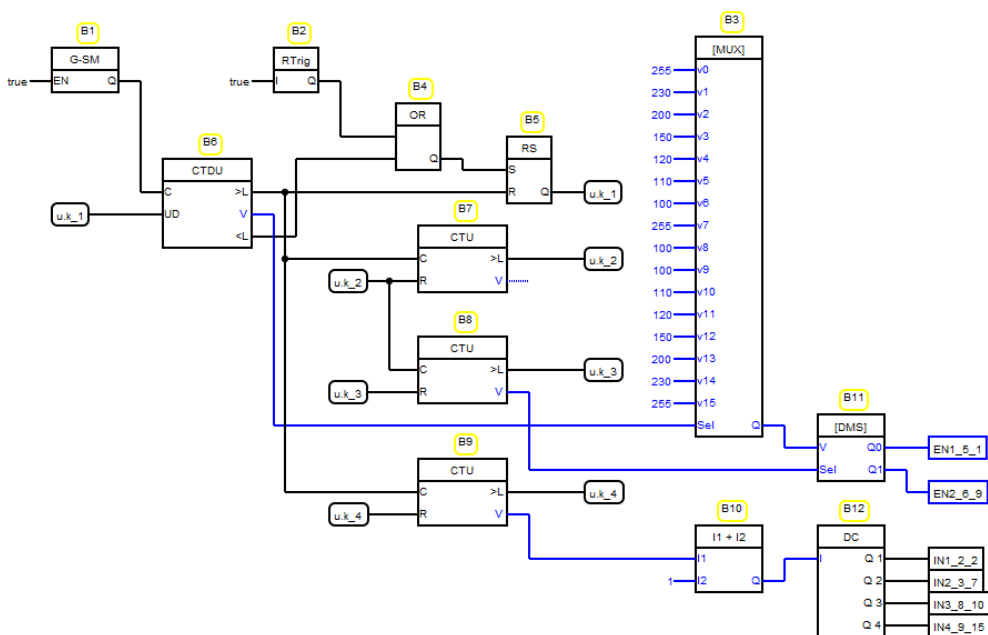
ჩვენ ამ პროექტში გამოვიყენებთ ყველაზე მარტივ მართვის დრაივერს, რომელიც დამზადებულია მუშაობისათვის სრულიად მზა მიკროსქემაზე. ამ მიკროსქემას ეწოდება L293D და წარმოადგენს ერთერთ, ყველაზე ფართოდ გავრცელებულ მიკროსქემას, რომლებიც აღნიშნული მიზნებისათვის არის განკუთვნილი.

L293D თავის შემადგენლობაში შეიცავს ერთბაშად ორ დრაივერს მცირე სიმძლავრის ორი ელექტროძრავის სამართავად. ამ დრაივერს აქვს ორი წყვილი შესასვლელი მმართველი სიგნალებისათვის და ორი წყვილი გამოსასვლელი ელექტროძრავების მიერთებისათვის. გარდა ამისა L293D-ს აქვს კიდევ ორი შესასვლელი თვითოეული ამ ძრავის მუშაობაზე ნების დართვისთვის. ეს შესასვლელები გამოიყენება ელექტრომტორების ბრუნვის სიჩქარის მართვისათვის განვივიმპულსურ მოდულირებული სიგნალებით (გიმ) მართვის შემთხვევაში.

L293D უზრუნველყოფს ძაბვების განმხილოებას თვითონ მიკროსქემის კვებისათვის და მის მიერ მართული ელექტროძრავების კვებისათვის, რაც შესაძლებლობას გვაძლევს მიუერთოთ ამ მიკროსქემას ელექტროძრავები უფრო დიდი კვების ძაბვებით, ვიდრე მიკროსქემის კვებაა, რაც დამატებით იძენს კიდევ უპირატესობას, რომ ამცირებს ხელშეშლებს, რომლებიც წარმოიშვება ძრავების მუშაობით გამოწვეული ძაბვების ნახტომებისაგან. მიკროსქემაში შემავალი თვითოეული დრაივერის მუშაობის პრინციპი ერთიდაიგივეა და ამიტომ განვიხილავთ ერთერთ მათგანს.

OUTPUT1 და OUTPUT2 გამოსასვლელებზე მიერთებულია პირველი ძრავი, როდესაც ENABLE1 შესასვლელზე მივაწვდით სიგნალს ლოგ. „1“ და ამასთანავე INPUT1-ზეც მივაწვდით ლოგ. „1“, ხოლო INPUT2-ზე-ლოგიკურ ნულს, მაშინ ძრავი იწყებს ბრუნვას ერთი მიმართულებით, ხოლო თუ INPUT1-ზე და INPUT2-ზე სიგნალებს შევცვლით საწინააღმდეგოებზე, მაშინ ძრავი იბრუნებს საწინააღმდეგო მიმართულებით.

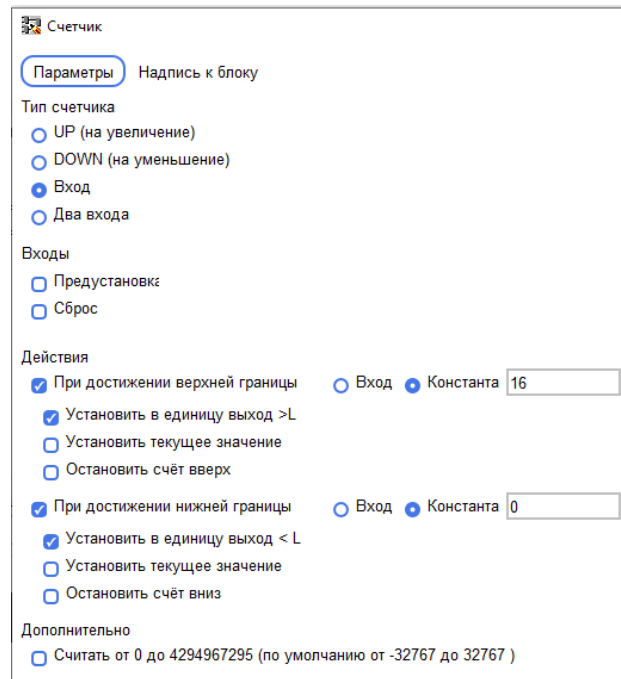
ახლა განვიხილოთ ჩვენს პროექტში აღწერილი ორი მუდმივი ელექტროძრავის მართვის პროგრამა (ფიგ. 28.1).



ფიგ. 28.1 ორი DC ძრავის მიკროსქემური მართვის პროგრამა

პროგრამის ძირითად ბლოკად შეიძლება ჩავთვალოთ მრავალშესასვლელიანი ანალოგური გადამრთველი [MUX] B3, რომელიც აწყობილია 16 შესასვლელზე. ამის შესაბამისად პროექტში შემოტანილი უნდა იყოს მთვლელი B6, რომელიც აწყობილია თექვსმეტამდე ერთის მხრივ და ნოლამდე-მეორეს მხრივ რევერსულ რეჟიმში სამუშაოდ (ფიგ. 28.2).

მთვლელის თვლის იმპულსები გამომუშავდება იმპულსების გენერატორის B1-ის მიერ, რომელიც მუშაობს სიმეტრიული მულტივიბრატორის რეჟიმში 500 მლწ იმპულსის ხანგრძლივობით. საწყის მდგომარეობაში RS ტრიგერი B5 დგება ერთიანის მდგომარეობაში B2 ბლოკის, ანუ RTrig ტრიგერის მიერ გამომუშავებული ერთჯერადი იმპულსით ლოგიკური შეკრების ბლოკის B4 გავლით, რაც იწვევს B6 მთვლელის დაყენებას დადებითი თვლის რეჟიმში უკუკავშირის სიგნალით u.k_1 დახმარებით.

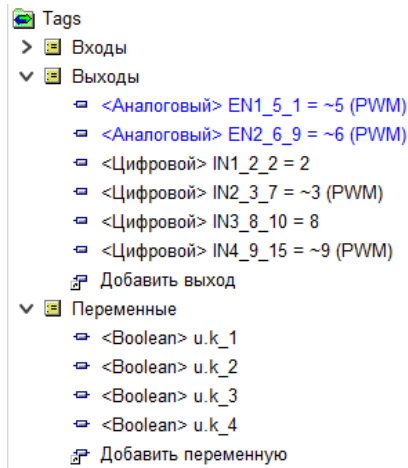


ფიგ. 28.2 მთვლელის პარამეტრიზაციის ფანჯარა

ამის შემდეგ შემდეგ როცა B6 მთვლელი დაითვლის 16-მდე, RS ტრიგერი მთვლელის >L სიგნალის ზემოქმედებით დადგება ნულოვან მდგომარეობაში და გადაიყვანს მთვლელს უკუ თვლის რეჟიმში. როდესაც მთვლელი ჩამოვა ნულამდე, მაშინ გამომუშავდება მთვლელის სიგნალი <L, რომელიც ლოგიკური შეკრების ბლოკის B4 გავლით ისევ დააყენებს RS ტრიგერს ერთიანის მდგომარეობაში და ა.შ.შ. დაუსრულებლად.

რადგან სიმეტრიული გენერატორი აწყობილია იმპულსის ხანგრძლიობაზე 500 მლწ და პაუზის ხანგრძლიობაც იგევე 500 მლწ-ია, ამიტომ გენერატორის გამოსასვლელზე იქნება ერთ წამიანი სიხშირის იმპულსები და ამიტომ აღმავალი გზის გავლას, ანუ ნოლიდან 15-ის ჩათვლით, მთვლელი B6 მოანდომებს 16 წამსა და დაღმავალი გზის გავლას ანუ 15-დან ნოლის ჩათვლით ასევე მოანდომებს 16 წამს. დავარქვათ ამ 32 წამიან ინტერვალს მთვლელის ერთი ციკლი. მთვლელის გამოსასვლელი V, სადაც ფიქსირდება ანალოგური სიგნალი 0...15, 15...0 მიეწოდება ანალოგური გადამრთველის [MUX] ბლოკის B5 არჩევის შესასვლელს Sel, რომელიც ახდენს ასე ვთქვათ შესასვლელი სიგნალების სკანირებას და Q გამოსასვლელზე შესასვლელზე არსებული სიგნალების v0... v15 გაგზავნას, მთვლელის აღმავალი გზის გავლის დროს და შემდეგ v15 ... v0 სიგნალებისა მთვლელის დაღმავალი გზის გავლის დროს.

B6 მთვლელის მიერ გენერირებული სიგნალი >1 მიეწოდება B7 მთვლელის შესასვლელს, რომლებიც u.k_2 უკუკავშირის ცლადის დახმარებით მუშაობს და ითვლის ორამდე. ეს მთვლელი მუშაობს ისე, რომ მთვლელის პირველი ციკლის განმავლობაში მასში ჩაწერილია 0, მეორე ციკლის განმავლობაში ჩაიწერება ერთიანი, მესამე ციკლის განმავლობაში მასში ისევ ჩაიწერება ნოლი და ა.შ. B7 მთვლელის u.k_2 სიგნალი მიეწოდება აგრეთვე B8 მთვლელის თვლის შესასვლელს, რომელიც ისევე როგორც B7 მთვლელი აწყობილია ორამდე თვლაზე, რისთვისაც იყენებს u.k_3 უკუკავშირის სიგნალს. ეს ბლოკი იმუშავებს შემდეგნაირად. B6 მთვლელის პირველი და მეორე ციკლის განმავლობაში მასში ჩაწერილი იქნება 0, ხოლო მესამე და მეოთხე ციკლის განმავლობაში - ჩაწერილი იქნება ერთი. B8 ბლოკის ანალოგური გამოსასვლელი V მიერთებულია ანალოგური დემულტიპლექსორის [DMS] Sel შესასვლელზე და რადგანაც ეს დემულტიპლექსორი B11 აწყობილია ორ გამოსასვლელზე, ამიტომ ის ანალოგური სკანირებული სიგნალები რომლებიც მიეწოდება V შესასვლელზე მთვლელის პირველი და მეორე ციკლის განმავლობაში გაშვებული იქნება B11 ბლოკის Q0 გამოსასვლელზე, ხოლო მესამე და მეოთხე ციკლის განმავლობაში კი ამავე ბლოკის Q1 გამოსასვლელზე. Q0 და Q1 გამომავალი სიგნალები კი თავის მხრივ მიერთებულნი არიან არდუინოს კონტაქტებზე სახელწოდებით EN1_5_1 და EN2_6_9 (იხილეთ ტეგების ზონა ფიგ. 28.3). ამ სახელწოდებებში პირველი ციფრები, ე.ი. 1 და 2 აღნიშნავენ პირველ და მეორე ძრავებს, მე-2 ციფრები, ე.ი. 5 და 6 აღნიშნავენ არდუინოს გიმ გამოსასვლელებიან კონტაქტებს, 1 და 9 აღნიშნავენ მიკროსქემის იმ ფეხების ნომრებს, რომლებზედაც უნდა იყოს მიერთებული აღნიშნული სიგნალები.



ფიგ. 28.3 პროექტის ტეგების ზონა

ანალოგური მულტიპლექსორის შესასვლელებზე v0...v15 მითითებულია გიმ სიგნალების სიდიდეები, რომლებიც შეიძლება შეიცვალოს 0-დან 255-მდე. ნოლოვანი გიმ სიგნალის მიწოდების დროს ძრავი გაჩერებულია, ხოლო 255 გიმ სიგნალის მიწოდების დროს ძრავი ბრუნავს მაქსიმალური სიჩქარით. ჩვენს მიერ ჩატარებული მრავალრიცხოვანი ექსპერიმენტების შედეგად დადგინდა იქნა, რომ ჩვენს ხელთ არსებული მუდმივი დენის ძრავები DC 3V-6V Small Gear Motor TT Motor 130 Motor for Arduino Smart Car Robot

არასტაბილურად მუშაობენ, ან საერთოდ ვერ მუშაობენ მცირე გიმ სიგნალებზე და გარდა ამისა გაშვების მომენტში სასურველია დასაწყისში მივცეთ მაქსიმალური სიგნალი და შემდეგ გადავიდეთ სასურველზე. ამის გათვალისწინებით ჩვენ შევარჩიეთ ის გიმ სიგნალები, რომლებიც ასახულია B3 ბლოკის შესასვლელებზე, რათა გვეჩვენებია მუდმივი ძრავის სიჩქარეების მდორე რეგულირების შესაძლებლობა. ეს გიმ სიგნალებია: 255, 230, 200, 150, 110, 100, ე.ი აქ წამდაწამ შევამცირეთ ნელნელა ძრავის ბრუნვები და ჩამოვედით მცირე ბრუნვებზე, შემდეგ უნდა ასევე ნელნელა უნდა ავიდეთ მაქსიმუმამდე, მაგრამ ჯერ სასურველია ავალეხნოთ ძრავს ინერცია და ამისათვის ვაწვდით მას მაქსიმალურ სიგნალს 255-ს ერთი წამის განმავლობაში და შემდეგ ვაგრძობთ სიჩქარეების მდორე ცვლას გიმ სიგნალების თანდათანობითი ზრდით: 100, 110, 120, 150, 200, 230, 255.

ახლა დაგვეჩა ერთი საკითხი. L293D-ის გამოყენებით ძრავების რევერსიული მართვის რეალიზაცია. ეს შეიძლება მოხდეს ამ მიკროსქემის INPUT1...INPUT4 სიგნალების გამოყენებით. როგორც ზემოთ იყო აღნიშნული, თუ INPUT1-ზე მივაწვდით ლოგ „1“ და INPUT2-ზე - ლოგიკურ „0“, მაშინ პირველი ძრავი ბრუნავს ერთი მიმართულებით, ხოლო თუ შევუცვლით INPUT1 და INPUT2-ს შესასვლელ სიგნალებს საწინააღმდეგოზე, მაშინ ძრავი დაბრუნდება მეორე მიმართულებით. ამ ფუნქციის რეალიზაციის მიზნით გამოყენებული

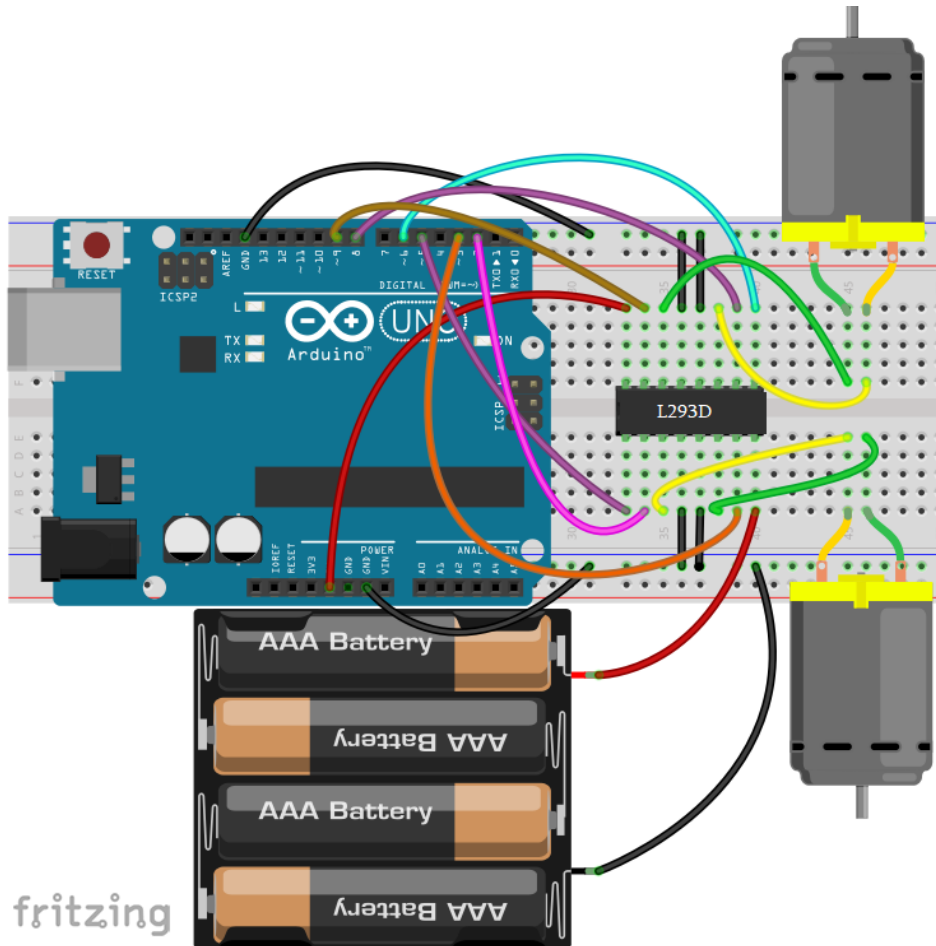
გვაქვს კიდეც ერთი მთვლელი B9, რომელიც ითვლის B6 მთვლელის >L სიგნალებს ოთხამდე. ამ მთვლელის ანალოგური გამოსასვლელი უნდა შევიყვანოთ დემიფრატორის შესასვლელზე B12. მაგრამ, რადგან დემიფრატორი აღიქვავს, შესასვლელ სიგნალებს: 1, 2, 3, 4, ჩვენ კი მთვლელის ანალოგურ გამოსასვლელზე V გვაქვს სიგნალები: 0, 1, 2, 3 ამიტომ, სიგნალების შეთანხმების მიზნით მთვლელის ანალოგურ გამოსასვლელს ვუმატებთ რიცხვს ერთს მათემატიკური შეკრების ბლოკის B10-ის გამოყენებით. შესაბამისად ამისა, როდესაც მთვლელი B6 იწყებს მუშაობას შეკრების რეჟიმში, მანამ 16-ს დაითვლიდეს B9 მთვლელი ნულოვან მდგომარეობაშია, ამიტომ B12-ის გამოსასვლელებზე Q1, Q2, Q3, Q4-ზე ლოგიკური სიგნალები მიიღებს სახეს 1 0 0 0 და პირველი ძრავი დაბრუნდება, მეორე გაჩერებულია, როდესაც B6 დაითვლის 16-მდე, მაშინ გამომუშავდება მისი >L სიგნალი, B9-ში ჩაიწერება ერთიანი და B12 -ის გამოსასვლელები მიიღებს სახეს: 0 1 0 0, რაც გამოიწვევს პირველი ძრავის ბრუნვის მიმართულების შეცვლას, ხოლო მეორე ძრავი ისევ გაჩერებული იქნება. როდესაც B6 მთვლელი ჩამოვა 0-მდე, შემდეგ ისევ დაბრუნდება და დაითვლის 16 მდე, ისევ გამომუშავდება მისი >L სიგნალი, B9-ში ჩაიწერება 2 და B12 -ის გამოსასვლელები მიიღებს სახეს: 0 0 1 0, რაც გამოიწვევს პირველი ძრავის გაჩერებას და ხოლო მეორე ძრავის დაბრუნებას ერთი მიმართულებით. ამის შემდეგ, როცა კიდეც გამომუშავდება >L სიგნალი, B9-ში ჩაიწერება 3 და B12 -ის გამოსასვლელები მიიღებს სახეს: 0 0 0 1, რაც გამოიწვევს მეორე ძრავის ბრუნვის მიმართულების შეცვლას, ხოლო პირველი ძრავი ისევ გაჩერებული იქნება. შემდეგი სიგნალი >L ისევ ჩაწერს B9-ში ნოლს და ა.შ.შ. აქ აუცილებლად მხედველობაში უნდა გვქონდეს მიღებული, რომ ძრავების დაბრუნებას იმ შემთხვევაში ექნებათ ადგილი, როცა შესაბამისი ნების დართვის სიგნალები EN_1 და EN2 იმყოფებიან ლიგიკური ერთიანის მდგომარეობაში. საბოლოოდ მივიღებთ, რომ B6 მთვლელის პირველი ორი ციკლის განმავლობაში მუშაობს პირველი ძრავი და ასრულებს წინ და უკან თითო მოძრაობას, ხოლო მეორე მესამე ციკლის განმავლობაში ამუშავდება მეორე ძრავი და ასრულებს ასევე წინ და უკან თითო მოძრაობას.

უნდა აღვნიშნოთ კიდეც, რომ B12 ბლოკის გამოსასვლელების დასახელებებში IN1_2_2, IN2_3_7, IN_3_8_10, IN4_9_15 (ფიგ. 28.3) პირველი რიცხვები: 1, 2, 3, 4 აღნიშნავენ უბრალოდ გამოსასვლელების ნომრებს, მეორე რიცხვები 2, 3, 8, 9 - არღუნიოს კონტაქტების ნომრებს სადაც გამოდიან ეს გამოსასვლელები, ხოლო მესამე რიცხვები - 2, 7, 10, 15 მიკროსქემის ფეხების ნომრებს რაზედაც მიერთებულნი არიან ეს გამოსასვლელები.

გასათვალისწინებელია აგრეთვე, რომ მიკროსქემის OUTPUT3, OUTPUT 6 და OUTPUT 11, OUTPUT 14 გამოსასვლელები უნდა შეერთდნენ პირველი და მეორე ძრავების გამომყვანებთან. მიკროსქემის მე-16 კონტაქტზე უნდა იქნას მიყვანილი ძაბვა +5 ვ,

მიკროსქემის კვებისათვის, მე-8 კონტაქტზე - ძრავების კვების ძაბვა +3...6 ვ. ხოლო 4, 5 და 12, 13-ზე კონტაქტებზე უნდა დაისვას ორივე კვების ძაბვისა და არდუინოს გაერთიანებული მიწები GND.

ორი DC ძრავის მიკროსქემური მართვის Fritzing სქემა ასახულია ფიგ. 28.4-ზე.



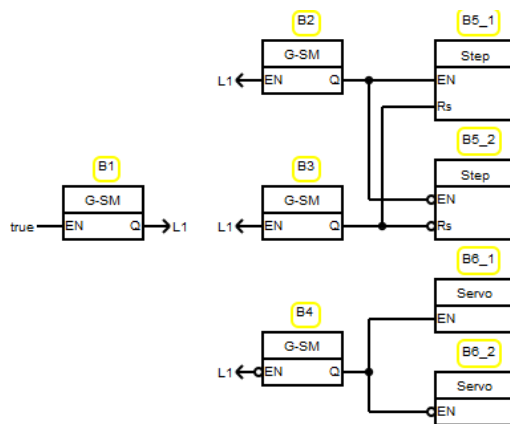
ფიგ. 28.4 ორი DC ძრავის მიკროსქემური მართვის Fritzing სქემა

პროექტი_29

სერვო ძრავი და ბიჯური ძრავი რევერსით

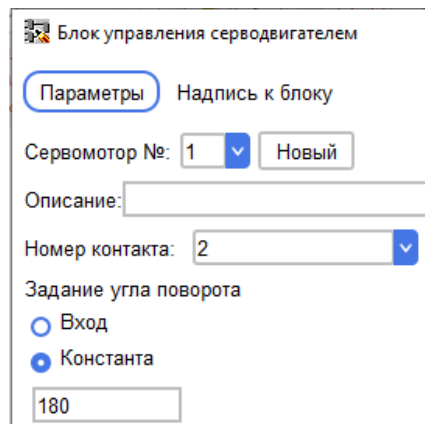


ბიჯური ძრავის მართვის ბლოკი წარმოადგენს მოხერხებულ ინტერფეისს ბიპოლარული ანდა უნიპოლარული ბიჯური ძრავების მართვისათვის. ბიჯური ძრავების მართვისათვის, მათი ტიპისაგან დამოკიდებულებით (ბიპოლარული თუ უნიპოლარული) და მიერთების მეთოდისაგან დამოკიდებულებით საჭიროა ზოგიერთი დამატებითი ელექტრონული კომპონენტების გამოყენება. უნიპოლიატრული ბიჯური ძრავების მიერთებისათვის საჭირო ხდება ითხი შესასვლელით მართვა. მიერთების ასეთი სქემისათვის გამოიყენება დატვირთვის დრაივერის მიკროსქემა ULN 2003 ან ULN2004. ბიპოლარული ბიჯური ძრავების მიერთებისათვის კი გამოიყენება ასევე ოთხშესასვლელიანი მართვა SN754410NE მიკროსქემის გამოყენებით. ფიგ. 29.1-ზე მოყვანილია ბიპოლარული ბიჯური ძრავის მართვის ვიზუალური პროგრამა სადაც ბიჯური ძრავის ბლოკი B5 პირობითად წარმოდგენილია ორი ქვებლოკის სახით B5_1 და B5_2, ერთი-საათის ისრის მიმართულებით მოძრაობის მართვისათვის და მეორე-საწინააღმდეგო მოძრაობის მართვისათვის. ხოლო სერვოძრავის ბლოკი B6 ასევე წარმოდგენილია ორი ქვებლოკით B6_1 და B6_2 ასევე რევერსული მოძრაობების მართვისათვის.



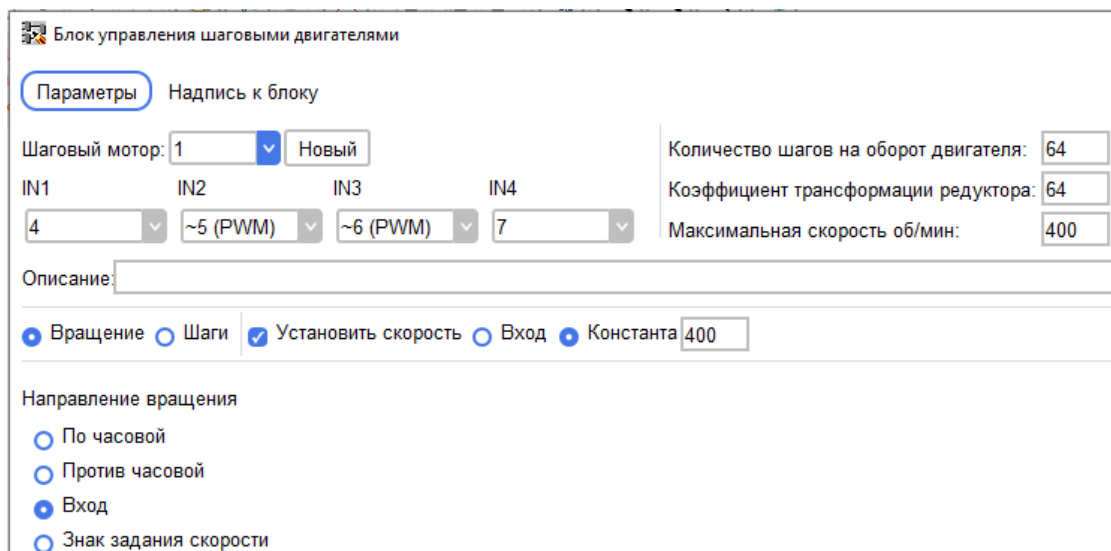
ფიგ. 29.1 სერვოძრავის და ბიჯური ძრავის რევერსული მართვის პროგრამა

სერვომძრავის B6_1 ბლოკის პარამეტრიზაციის ფანჯარა წარმოდგენილია ფიგ. 29.2-ზე, საიდანაც ჩანს, რომ ძრავის მობრუნების კუთხის კონსტანტის ველში ჩაწერილია 180 გრადუსი. მე-2 ბლოკის B6_2-ბლოკის პარამეტრიზაციის ფანჯარა უნდა იყოს ზუსტად ასეთივე იმ განსხვავებით, რომ ძრავის მობრუნების კუთხის კონსტანტის ველში ჩაწერილი უნდა იყოს ნოლი.



ფიგ. 29.2 სერვომძრავის პირველი ბლოკის მართვის პარამეტრიზაციის ფანჯარა

ბიჯური ძრავის პირველი ბლოკის პარამეტრიზაციის ფანჯარა წარმოდგენილია ფიგ. 29.3-ზე და მეორე ქვებლოკისათვისაც იგივეა.

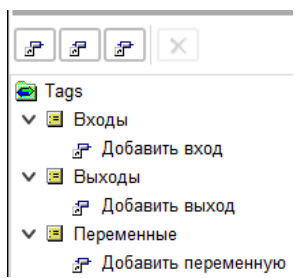


ფიგ. 29.3 ბიჯური ძრავის ბლოკის პარამეტრიზაციის ფანჯარა

სერვო და ბიჯური ძრავების მართვის ალგორითმი ამ პროექტისათვის მდგომარეობს იმაში, რომ აქ გამოიყენება მრავალგენერატორული მართვა. გენერატორი B1, რომელიც მუშაობს სიმეტრიული მულტივიზრატორის რეჟიმში გამოიმუშავებს 16 წამიანი იმპულსისა

და 16 წამიანი პაუზის იმპულსების თანმიმდევრობას. 16 წამიანი იმპულსის განმავლობაში ამუშავდება B2 და B3 გენერატორები, რომლებიც მუშაობენ ასევე სიმეტრიული მულტივიბრატორის რეჟიმში და გამოიმუშავენ რვაწამიან და ოთხწამიან იმპულსებისა და პაუზების თანმიმდევრობებს. რადგან B2 გენერატორის გამოსასვლელი, რომელზედაც გენერირდება სიმეტრიული პაუზები რვაწამიანი ხანგრძლიობებით, შეერთებულია ბიჯური ძრავის ქვებლოკების ნების დართვის შესასვლელებთან EN და ამასთანავე მეორე ქვებლოკთან ინვერსირებით, ამიტომ B2 ბლოკის გამოსასვლელზე რვაწამიანი იმპულსის განმავლობაში, როდესაც B3 ბლოკის გამოსასვლელზე ასევე იქნება იმპულსი ოღონდ 4 წამის განმავლობაში, მაშინ ბიჯური ძრავი იმოდრავებს საათის ისრის მიმართულებით, ხოლო როდესაც B3 ბლოკის გამოსასვლელზე დაჯდება პაუზა მაშინ ბიჯური ძრავი იმოდრავებს საათის ისრის საწინააღმდეგო მიმართულებით ასევე ოთხი წამის განმავლობაში. მაგრამ როდესაც B2 ბლოკის გამოსასვლელზე დადგება პაუზა, მაშინ სერვოდრავი იმუშავებს იგივე ტემპში მაგრამ საწინააღმდეგო ფაზაში, ე.ი. პირველი ოთხი წამის განმავლობაში ძრავი იმოდრავებს საათის ისრის საწინააღმდეგო მიმართულებით და მეორე ოთხი წამის განმავლობაში კი-პირდაპირი მიმართულებით. როდესაც B1 გენერატორი გადავა პაუზის რეჟიმში და B2 და B3 გენერატორების მუშაობა დაიბლოკება, მაშინ B5_2 ბლოკის ორთავე შესასვლელზე დაჯდება ლოგიკური ერთიანები, ამ შემთხვევაში ძრავი იმოდრავებს უწყვეტად ერთი მიმართულებით, საათის ისრის მოზრაობის საწინააღმდეგოდ.

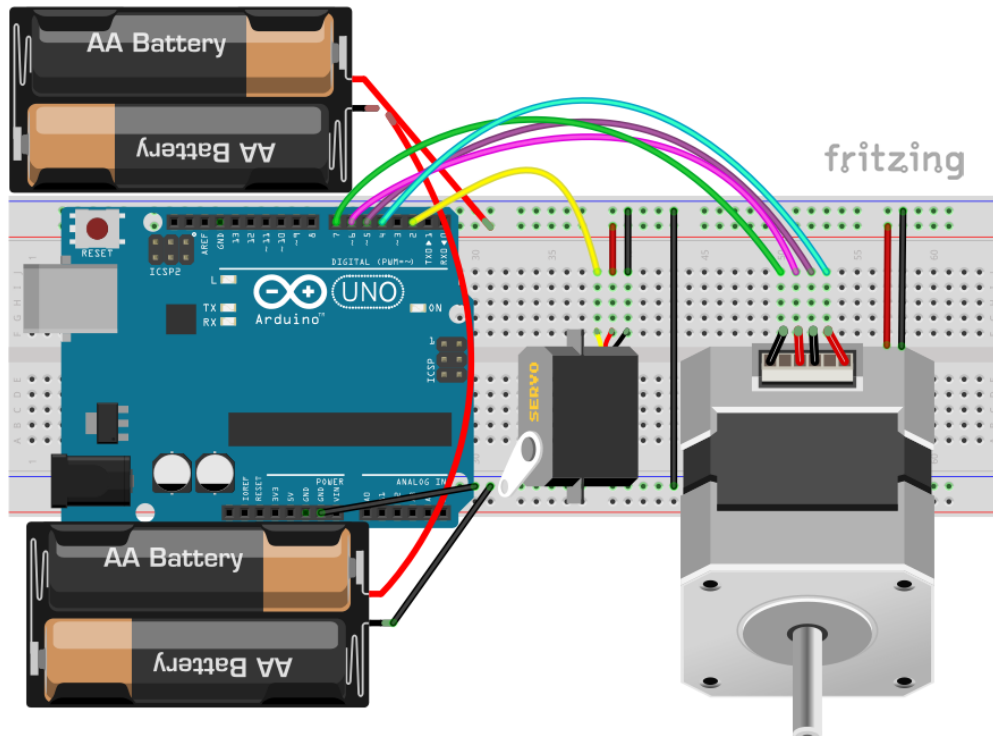
როდესაც B1 გენერატორის გამოსასვლელზე იმპულსია, მაშინ ამუშავდება აგრეთვე B4 გენერატორი, რომელიც გამოიმუშავებს ერთწამიან იმპულსისა და ერთწამიან პაუზის თანმიმდევრობას და ამ სიხშირით მოხდება B6 სერვოდრავის გადასვლა/გადმოსვლა ნოლი გრადუსის მდგომარეობიდან 180 გრადუსის მდგომარეობაში და პირიქით.. როდესაც B1 გენერატორის გამოსასვლელზე პაუზაა, მაშინ B4 გენერატორს გამოსასვლელზე იქნება მუდმივად ნოლი, რაც გამოიწვევს B6_2 ბლოკის ამუშავებას, სერვოდრავი დადგება ნოლი გრადუსის მდგომარეობაში და შეინარჩუნებს ამ მდგომარეობას თექვსმეტი წამის განმავლობაში.



ფიგ. 29.4 პროექტის ტეგების ზონა

სერვომრავისა და ბიჯური ძრავის რევერსიული მართვისთვის ფიგ. 29.4-ზე ტეგების ზონაში არავითარი ჩანაწერების გაკეთება საჭირო არ აღმოჩნდა.

სერვომრავისა და ბიჯური ძრავის რევერსიული მართვის Fritzing სქემა წარმოდგენილია ფიგ. 29.5-ზე.



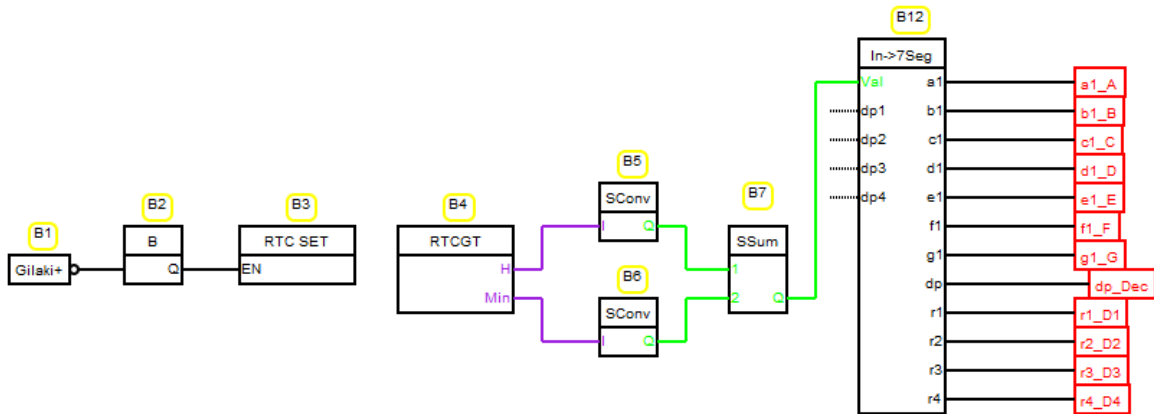
ფიგ. 29.5 სერვომრავისა და ბიჯური ძრავის რევერსიული მართვის Fritzing სქემა

პროექტი_30

საათი და სტრიქონების შეკრება_1

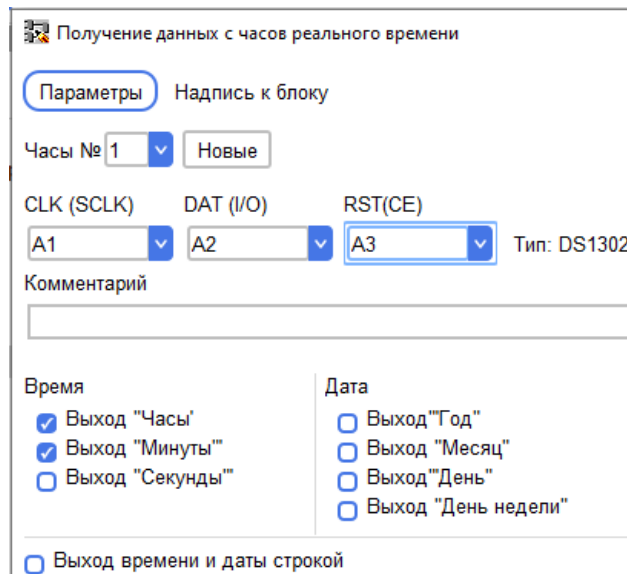


შევიდეთ FLProg -ის ბლოკების ბიბლიოთეკის რეალური დროის საათების საქალაქდემი-Часы реального времени და სამუშაო ზონაში გადმოვიტანოთ ორი ბლოკი: რეალური დროის საათისთვის დროის დაყენების ბლოკი B3-Установка времени для часов реального времени და რეალური დროის საათიდან დროის მიღების ბლოკი B4-Получение времени для часов реального времени (ფიგ. 30.1).

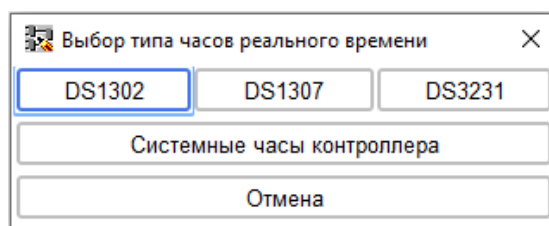


ფიგ. 30.1 რეალური დროის საათის მართვის პროგრამა

რეალური დროის საათში დროის დაყენების ბლოკის დანიშნულებაა საწყის მომენტში საათში დროის იმ მონაცემების შეყვანა, რომელიც მითითებული იყო ბლოკის აწყობის ფანჯარაში (ფიგ. 30.2). B3 ბლოკის EN შესასვლელზე მაღალი პოტენციალის მიწოდების მომენტში დროის ის მონაცემები, რომლებიც იყო ჩაწერილი ბლოკის აწყობებში (ფიგ. 30.2) ჩაიწერება რეალური დროის საათის მოდულში (ჩვენს შემთხვევაში 13 საათი და 50 წუთი).



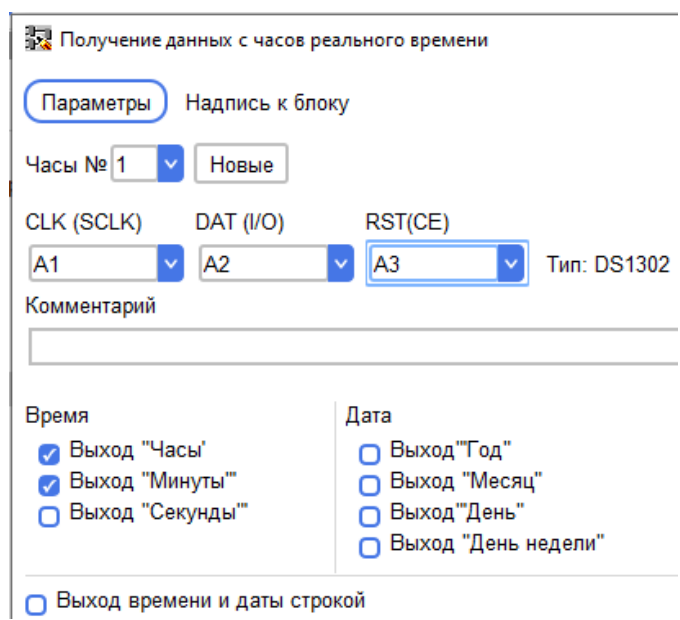
ფიგ. 30.2 რეალური დროის საათში დროის მონაცემების საწყისი დაყენება



ფიგ. 30.3 საათის ტიპის შერჩევის ფანჯარა

ახალი საათის შექმნისას არის შესაძლებლობა არჩეულ იქნას საათის ტიპი (ფიგ. 30.3). სხვადასხვა ტიპის საათისთვის მისი მიერთებების შესახებ ინფორმაცია სხვადასხვაა. ჩვენს პროექტში გამოყენებული გვაქვს DS1302 ტიპის საათი, რომლისთვისაც აწყობის დიალოგი წარმოდგენილია ფიგ. 30.4-ზე.

რეალური დროის საათიდან რეალური დროის მიღების ბლოკის დანიშნულებაა დროის შესახებ მონაცემთა მიღება. ახალი საათის შექმნის შემთხვევაში აქაც არის საათების ტიპის შერჩევის შესაძლებლობა. ბლოკის პარამეტრიზაციის დროს საჭიროა უბრალოდ ჩართული იყოს ის ალმები, რომელთა შესახებ გვჭირდება ინფორმაციის მიღება. ჩვენს შემთხვევაში ჩართული გვაქვს მხოლოდ ორი ალამი: გამოსასვლელი საათის გამოსასვლელი-Выход «Часы» და წუთის გამოსასვლელი -Выход «Минуты».

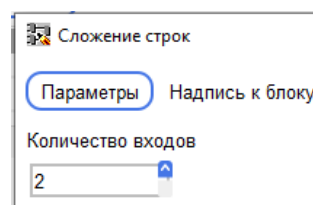


ფიგ. 30.4 რეალური დროის საათიდან მონაცემთა მიღების არჩევა

ახლა გადავიდეთ რეალური დროის საათის მართვის პროგრამის აღწერაზე. რადგან საათებისა და წუთების თანრიგთა რაოდენობა არ აჭარბებს ორ-ორს, ანუ სულ ოთხია, ამიტომ შესაძლებლად ვცანით ამ მონაცემების ასახვისათვის გამოგვეყენებინა ოთხთანრიგა შევიდსეგმენტა დისპლეი B12, რომელიც უკვე განხილული გვქონდა წინა პროექტებში.

როგორც ცნობილია, ამ ინდიკატორის ბლოკს FLProg პროგრამაში აქვს შესაძლებლობა მიიღოს ინფორმაცია როგორც Integer, ასევე String ფორმატში. აი, სწორედ ამ უკანასკნელის შესაძლებლობას გამოვიყენებთ ჩვენ ჩვენ პროექტში. ჯერ გადავაქცევთ რეალური დროის საათის B4 საათებისა (H) და წუთების (Min) გამოსასვლელებს (ფიგ. 30.1) სტრიქონებად B5 და

B6 სტრიქონების კონვერტაციის ბლოკების დახმარებით, რომლებიც აღებულია ბლოკების ბიბლიოთეკის ტიპების კონვერტაციის საქალაქდედან-Конвертация типов და ავიღებთ სტრიქონების კონვერტაციის ბლოკს-Конвертация строк ორჯერ B4 და B5 ბლოკებისათვის ცალცალკე, ხოლო შემდეგ იმავე ბიბლიოთეკის სტრიქონების საქალაქდედან-Строки სტრიქონების შეკრების ბლოკის-Сложение строк (B7) გამოყენებით გავაერთიანებთ მათ დისპლეიზე ასახვისათვის. სტრიქონების შეკრების ბლოკს ჭირდება პარამეტრიზაცია ანუ შესაკრები სტრიქონების რაოდენობის მითითება (ფიგ. 30.5). ჩვენ შემთხვევაში, რადგან მხოლოდ ორი სტრიქონი გვაქვს შესაკრები, ამიტომ შესასვლელების რაოდენობის-Количество входов ველში ვწერთ ორს.

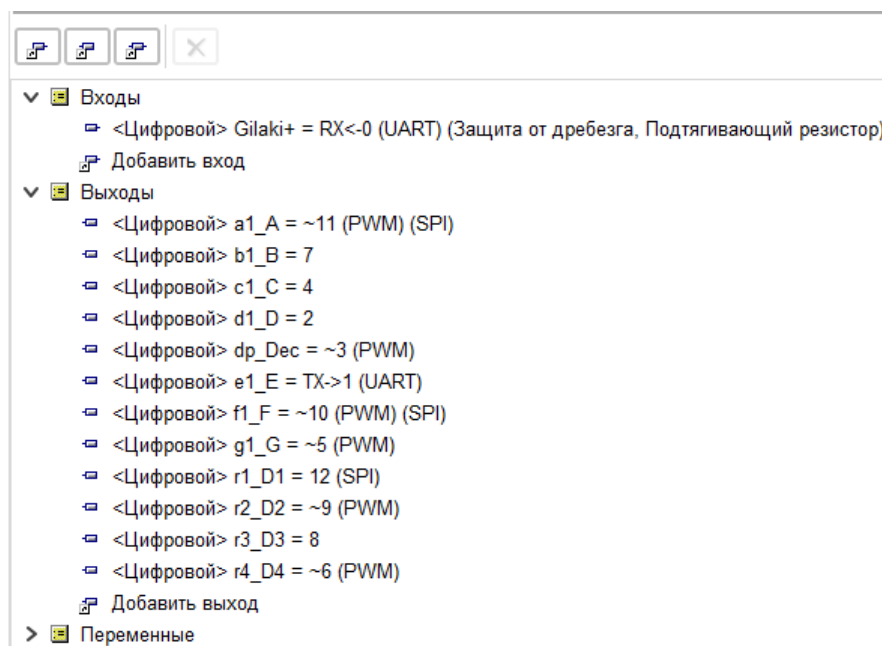


ფიგ. 30.5. შესაკრები სტრიქონების რაოდენობის მითითება

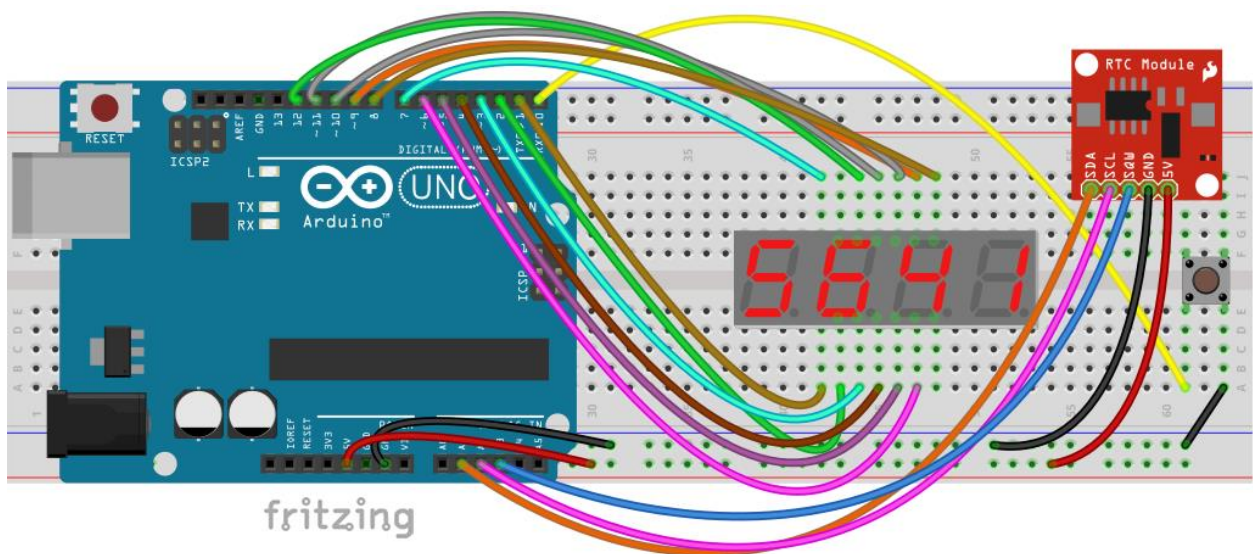
რეალური დროის საათში საწყისი დროის მაჩვენებლების ჩაწერისათვის ჯერ უნდა უნდა შვიდეთ B3 ბლოკის პარამეტრიზაციის ფანჯარაში (ფიგ. 30.2), დავაჭიროთ ღილაკს ახალი-Новые, რითაც საათის ნომრის ველში-Часы № ავტომატურად ჩაიწერება ერთიანი, შემდეგ ბლოკის მართვის სიგნალების ველებში-CLK, DAT, RST ჩავწერთ არდუინოს თავისუფალი ანალოგური კონტაქტები (A1, A2, A3 ჩვენს პროექტში), შემდეგ ჩავწერთ დროის მაჩვენებლების ველები: წელი, თვე, კვირის დღე, რიცხვი, საათი, წუთი და წამი. ჩვენ აღნიშნული გვაქვს მხოლოდ საათი და წუთი და შესაბამის კონსტანტის ველებში ჩაწერილი გვაქვს 13 საათი და 50 წუთი და დავაჭიროთ მზადყოფნის ღილაკს-Готово თავის მაჩვენებლით. ასევე უნდა დავაპროგრამოთ B4 ბლოკი (ფიგ. 30.4).

დისპლეის შეერთება არდუინოსთან ძალზედ მარტივია. საჭიროა უბრალოდ არდუინოს 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 კონტაქტები პირდაპირ, ერთი ერთზე შევაერთოთ დისპლეის 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 ფეხებთან. ტეგების ზონა იხილეთ ფიგ. 30.5-ზე. პროგრამის არდუინოში ჩატვირთვისა და გაშვების შემდეგ საჭიროა ერთჯერადად დავაფიქსიროთ დროის რეალური მონაცემები რეალურ საათის მოდულში. ამისათვის, ვაჭერთ თითს ღილაკს B1, რითაც გამომუშავდება მის

გამოსასვლელზე დადებითი სიგნალი, რომელიც B2 ბოუნცეს ბლოკის გავლით მიეწოდება B3 ბლოკის EN შესასვლელს და დროის საწყისი მონაცემები, რომლებიც ჩვენ დაფიქსირებული გვექონდა ამ ბლოკის პარამეტრიზაციის ფანჯარაში (ფიგ. 30.2) - გადაიწერება B3 ბლოკში და აისახება დისპლეიზე. ამის შემდეგ რეალური დროის საათის ბლოკი გააგრძელებს რეალური დროის მნიშვნელობების გამოთვლას და დისპლეი გააგრძელებს რეალური დროის ანათვალის ასახვას განუწყვეტლივ. რეალური დროის საათის Fritzing სქემა ასახულია ფიგ. 30.6-ზე.



ფიგ. 30.5. ტეგების ზონა



ფიგ. 30.6 რეალური დროის საათის Fritzing სქემა

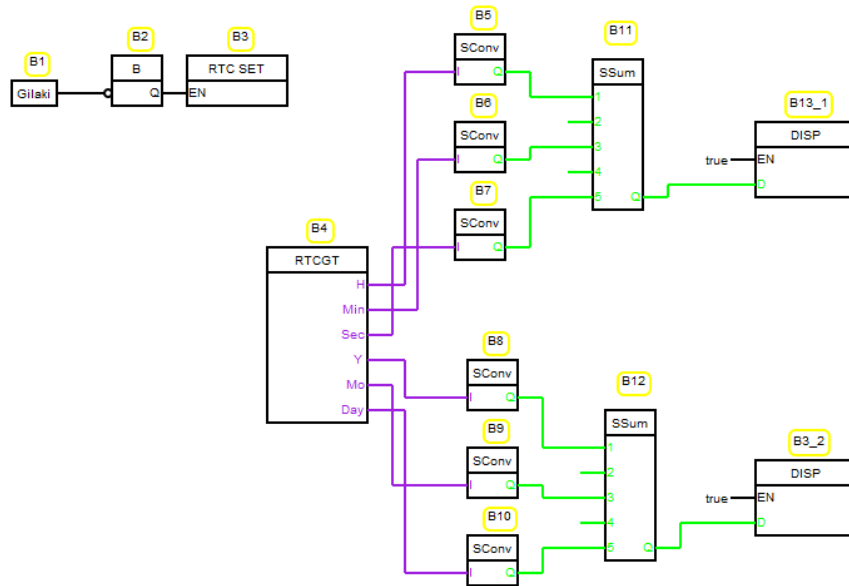
პროექტი_31

საათი და სტრიქონების შეკრება_2



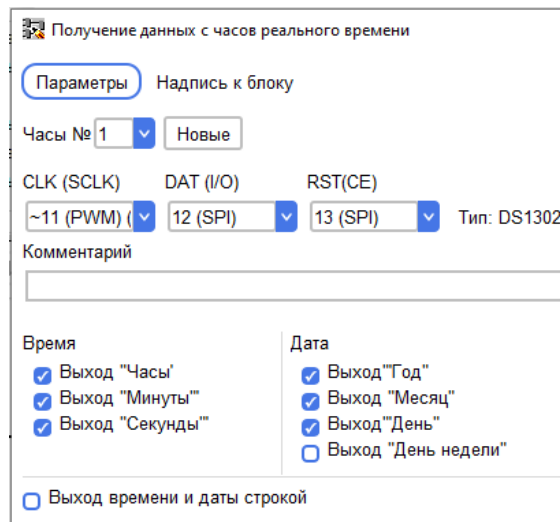
ახლა გავაუმჯობესოთ პროექტი და გამოვიტანოთ დისპლეიზე სრული ინფორმაცია დროის შესახებ, კერძოდ: საათი, წუთი, წამი, წელი, თვე, დღე ანუ: H, Min, Sek, Y, Mo, Day. ამისათვის ჩვენ დაგვჭირდება სხვა დისპლეის, კერძოდ დისპლეი ჩიპზე HD44780-ის გამოყენება და მისი გადმოტანა ბლოკების ბიბლიოთეკიდან FLProg-ის სამუშაო ზონაში B13_1, B13_2 (ფიგ. 31.1).

დასმული ამოცანის შესასრულებლად საჭირო ხდება რეალური დროის საათის ბლოკის B4 პარამეტრების კორექტირება და ამის შესაბამისად პარამეტრიზაციის ახალი ფანჯარა ასახული გვაქვს ფიგ. 31.2-ზე.



ფიგ. 31.1. საათის მონაცემების ასახვის პროგრამა HD44780 ჩიპზე

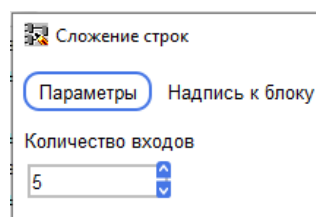
როგორც ფიგურიდან ჩანს, ფანჯარაში მონიშნულია არა მარტო „საათის“, „წუთების“, „წამების“ გამოსასვლელები, არამედ „წელის“ გამოსასვლელი-Выход «Год», თვის გამოსასვლელი-Выход «Месяц» და დღის გამოსასვლელი-Выход «День».



ფიგ. 31.2 რეალური დროის საათის მონაცემების პარამეტრიზაციის ფანჯარა

ვინაიდან ექვსივე მონაცემის განთავსება დისპლეის ერთ სტრიქონში შეუძლებელია, ამიტომ ჩვენ დაგვჭირდება მათი განთავსება ორ სტრიქონში სამ-სამი მონაცემისა თვითოეულში. ამიტომ უნდა გამოვიყენოთ სტრიქონების შეკრების ორი ბლოკი B11 და B12, რომელთა შესასვლელებზეც მივაწოდებთ რეალური დროის საათის ზემოთჩამოთვლილ ექვსივე მონაცემს სტრიქონების კონვერტაციის ბლოკების B5... B10-ის გავლით.

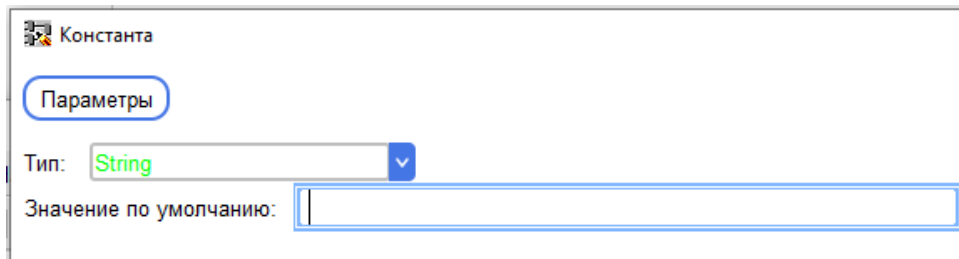
ახლა განვიხილოთ თუ როგორ გვაქვს აწყობილი სტრიქონების შეკრების ბლოკი. ჩვენ შეგვიძლია ეს სამივე მონაცემი პირდაპირ მივუწეროთ ერთმანეთს განმხლოების გარეშე, მაშინ საკმარისი იქნება სტრიქონების შეკრების ბლოკს შესაბამის ველში ჩავუწეროთ 3. მაგრამ ეს გამოიწვევს მონაცემების ერთმანეთზე გადაბმას და აღსაქმელად იქნება რთულად. ამიტომ საჭირო იქნება გამოიყოს ეს მონაცემები ერთმანეთისაგან პრაბელით. ე.ი. ჩვენ დაგვჭირდება სტრიქონების შეკრების ბლოკის ხუთი შესასვლელი (ფიგ. 31.3).



ფიგ. 31.3 სტრიქონების შეკრების ბლოკის პარამეტრიზაცია

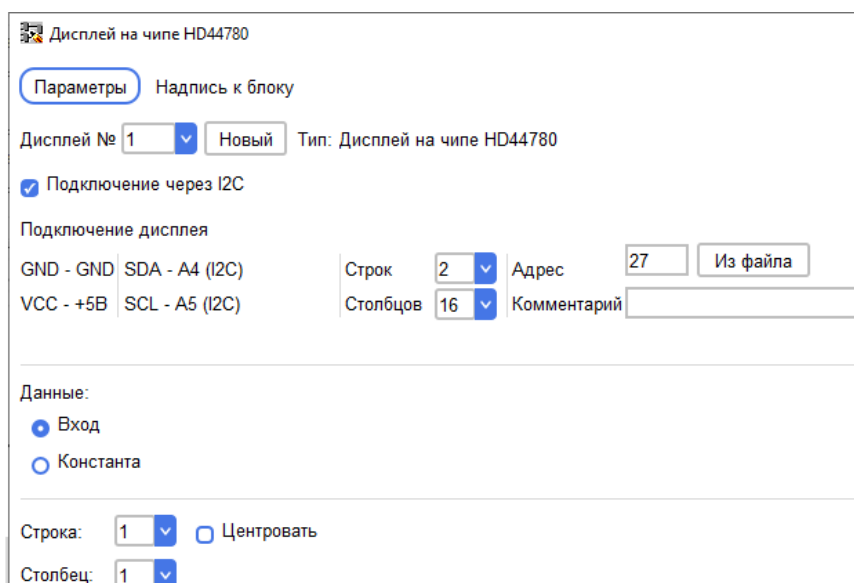
ამის შემდეგ ამ B11 ბლოკის პირველ შესასვლელზე პირდაპირ ვაერთებთ რეალური დროის საათის H გამოსასვლელს B5 ბლოკის გავლით, მეორე შესასვლელზე კი უნდა შევიდეთ მის კონტექსტურ მენიუში, გამოვა შეტყობინება ჩასვით კონსტანტა-Вставить константу და მასზედ დაწკაპუნებით გამოვა კონსტანტის ჩასმის ფანჯარა (ფიგ. 31.4).

ამ ფიგურაზე ველში მნიშვნელობა სიჩუმით-Значение по умолчанию ჩვენ შეგვიძლია ჩავსვათ ნებისმიერი სიმბოლოების ან/და ნიშნების ნაკრები, ან ნიშანი. ჩვენ გვჭირდება განვამზადოთ მონაცემები და ამისთვის გამოვტოვოთ ერთი პოზიცია, ამიტომ ვაჭერთ კლავიატურაზე პრაბელის ღილაკს, შემდეგ კი ფანჯარაში Готово-ს ღილაკს, შედეგად ვერტიკალური ხაზი ველში გადაიწევეს მარჯვნივ ერთ პოზიციაზე, რაც მაჩვენებელი იქნება იმისა, რომ ამ შესასვლელზე დაფიქსირდა პრაბელი.



ფიგ. 31.4. პრაბელის დასმის პროცედურა სტრიქონების შეკრების ბლოკის შესასვლელზე

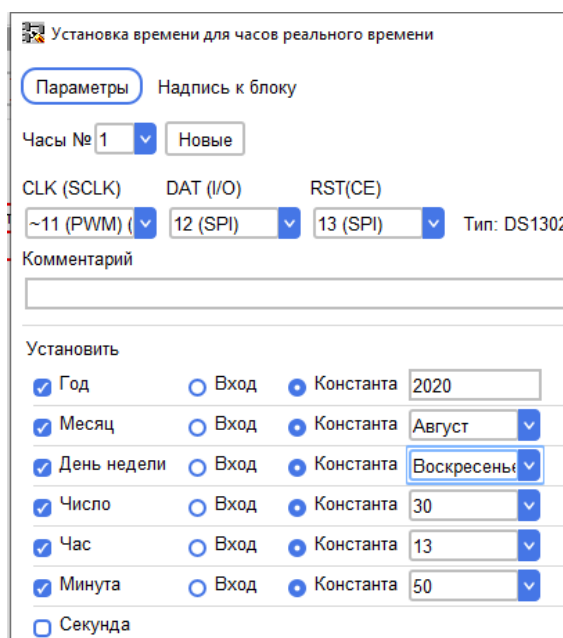
ფიგ. 31.4-ზე ეს ვერტიკალური ხაზი გადაწეულია მარჯვნივ ერთი პოზიციით, ეს ნიშნავს, რომ ამ შესასვლელზე დაფიქსირებულია პრაბელი. სტრიქონების შეკრების ბლოკის მე-3 შესასვლელზე ისევე როგორც პირველზე ჩვენ მივიყვანთ წუთების მონაცემს-Min კონვერტაციის ბლოკის B6-ის გავლით, მე-4 შესასვლელზე დავსვათ ისევ პრაბელს, შემდეგ შევიყვანთ წამების მონაცემს B7 ბლოკის გავლით და შესაბამისად პირველი სტრიქონის ასასახი ინფორმაციის ფორმირება დამთავრებული იქნება. ასევე მოვიქცევით მე-2 სტრიქონის შესასვლელებისთვისაც, ანჯღუ B12 ბლოკის შესასვლელების მიერთებები და პრაბელების ჩასმა.



ფიგ. 31.5 დისპლეის ჩიპზე HD44780 პირველი სტრიქონის პარამეტრიზაცია

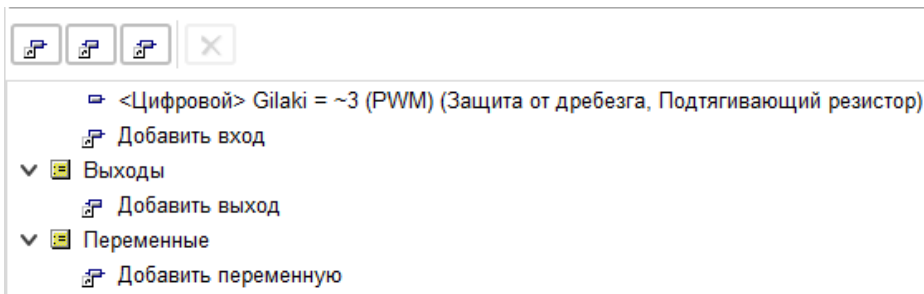
სტრიქონების შეკრების ბლოკების B11 და B12 გამოსასვლელები შეერთებულნი არიან ორსტრიქონიანი, თექვსმეტპოზიციანი დისპლეის ჩიპზე HD44780 პირველი და მეორე სტრიქონების ანალოგურ შესასვლელებთან ქვებლოკებზე B13_1 და B13_2. ამ ქვებლოკების პარამეტრიზაციის ფანჯრები მცირედ განსხვავდება ერთმანეთისაგან. ფიგ. 31.5-ზე მოცემულია დისპლეის პირველი ქვებლოკის პარამეტრიზაციის ფანჯარა. მეორე ქვებლოკისათვის სტრიქონის ველში-Строка უნდა ჩაიწეროს ორიანი.

რეალური დროის საათის დროების დაყენების ბლოკის B3-ის პარამეტრიზაცია ამ პროექტისთვის უნდა მოხდეს ფიგ. 3.6-ზე წარმოდგენილი ფანჯრის მიხედვით. აქ უკვე მითითებული უნდა იყოს წელი-Год მაგალითად-2020), თვე-Месяц მაგალითად აგვისტო-Август, კვირის დღე-День недели მაგალითად კვირა-Воскресение.

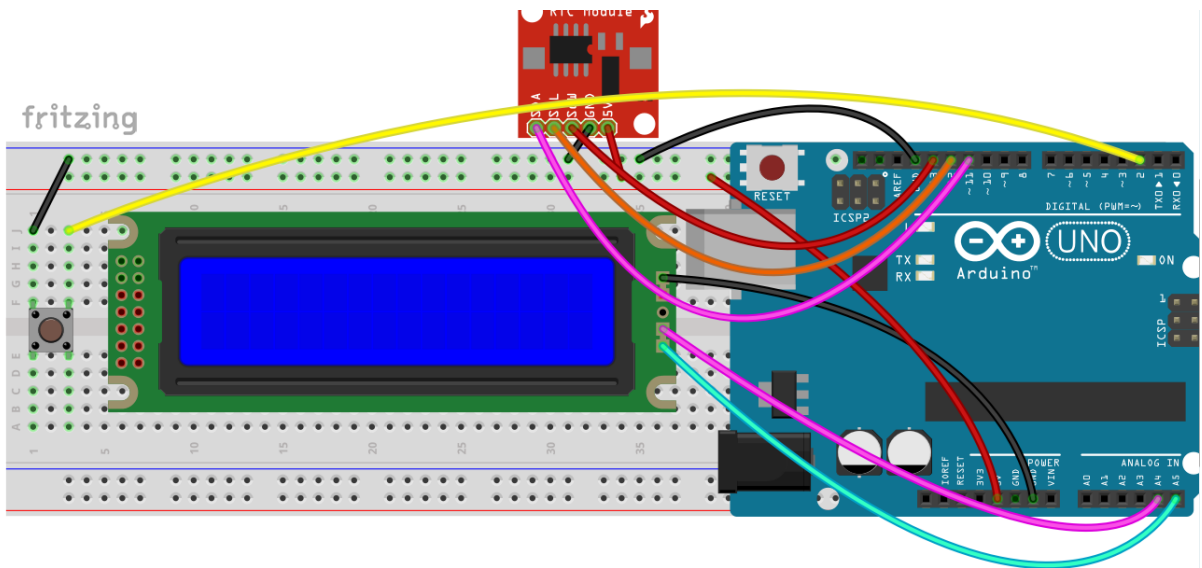


ფიგ. 3.6 რეალური დროის საათის დროების დაყენების ბლოკის პარამეტრიზაცია

როდესაც ჩვენ დავაჭერთ ღილაკს B1, რომელიც შეერთებულია არდუინოს მე-3 გამოსასვლელთან (იხ. ფიგ. 31.7 ტეგების ზონა) მაშინ ყველა ეს პარამეტრები რომლებიც ჩვენ მივუთითეთ რეალური დროის საათის დროების დაყენების ბლოკის პარამეტრიზაციის ფანჯარაში-გადაიწერება რეალურ დროის საათში და შემდეგ ეს საათი გააგრძელებს ავტომატურ რეჟიმში დროის პარამეტრების გენერირებასა და ასახვას დისპლეიზე. რეალურ დროის საათის Fritzing სქემა ასახულია ფიგ. 31.8-ზე



ფიგ. 31.7 ტეგების ზონის ფანჯარა



ფიგ. 31.8 რეალური დროის საათის მართვის Fritzing სქემა

პროექტი_32

საათი და სტრიქონების შეკრება_3



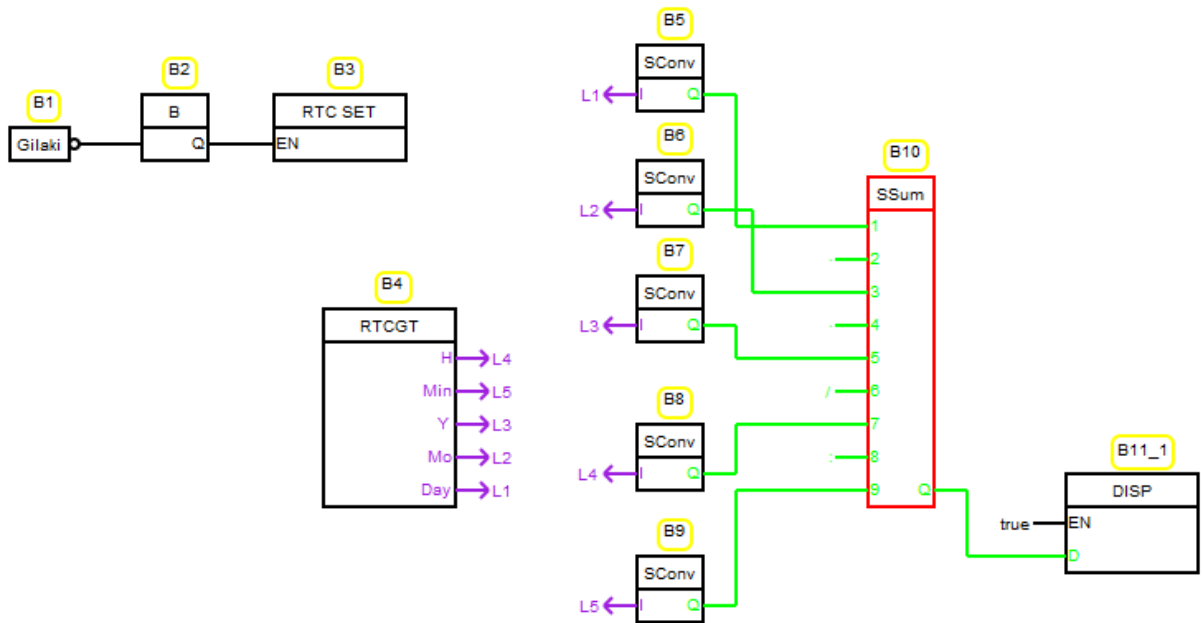
ხშირად საჭირო ხდება რეალური დროის საათიდან გამოსული მთელი ინფორმაცია ასახულ იყოს დისპლეის მხოლოდ ერთ სტრიქონში, რათა მეორე სტრიქონი გამოყენებულ იქნას სხვა ინფორმაციის-მაგალითად მეტეოროლოგიური ინფორმაციის ასახვისათვის. ამ ამოცანის გადაწყვეტა შესაძლებელია თუ კი უარს

ვიტყვიot წამების ასახვაზე. მათთვის უბრალოთ აღარ რჩება ადგილი სტრიქონში. რეალური დროის საათში დროების დაყენების ბლოკის B3 პარამეტრიზაციის ფანჯარას ექნება სახე, რომელიც წარმოდგენილია ფიგ. 32.1-ზე.

ფიგ. 32.1 რეალური დროის საათის დროის დაყენების ფანჯარა

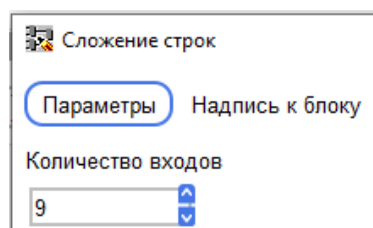
ფიგ. 32.2 რეალური დროის საათიდან მონაცემთა მიღების ბლოკის პარამეტრიზაცია.

რეალური დროის საათიდან მონაცემთა მიღების ბლოკის B4 პარამეტრიზაციის ფანჯარა კი მიიღებს სახრეს, რომელიც ნაჩვენებია ფიგ. 32.2-ზე. სტრიქონების კონვერტაციის ბლოკების რაოდენობა იქნება ხუთის ტოლი B5...B9 (ფიგ 32.3).



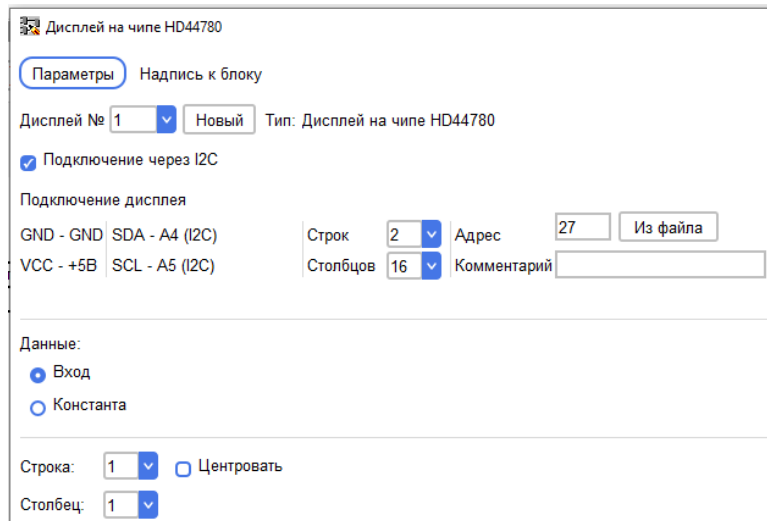
ფიგ. 32.3 რეალური დროის საათის მართვის პროგრამა

სტრიქონების შეკრების ბლოკი B10-სთვის შესასვლელების რაოდენობა Количество входов გაუტოლდება ცხრას (ფიგ. 32.4).

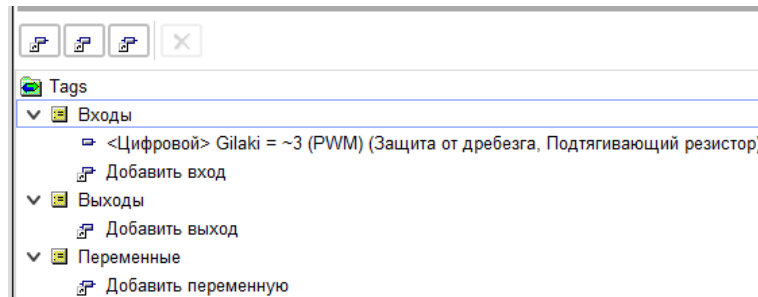


ფიგ. 32.4 სტრიქონების შეკრების ბლოკის პარამეტრიზაციის ფანჯარა

დისპლეის ბლოკის პარამეტრიზაციის ფანჯარაც B11_1 ქვებლოკისთვის იგივე იქნება როგორც წინა პროექტში (ფიგ. 32.5) და ტეგების ზონაც იგივე იქნება როგორც წინა პროექტში (ფიგ. 32.6).



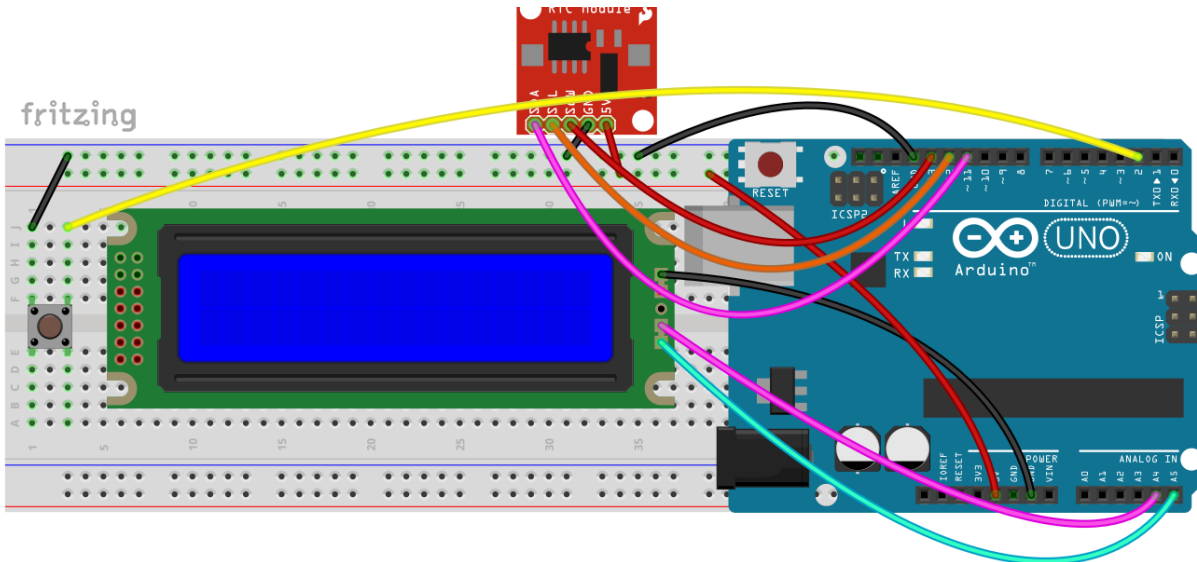
ფიგ. 32.5 დისპლეის ქვებლოკის პარამეტრიზაციის ფანჯარა



ფიგ. 32.6 პროექტის ტეგების ზონა

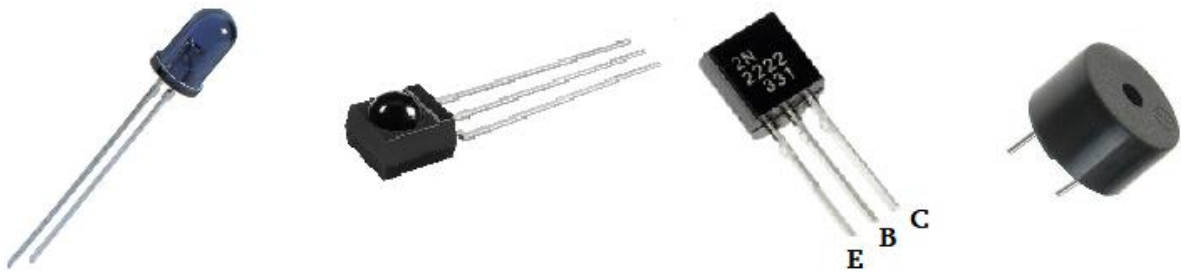
როგორც უკვე აღნიშნული იყო, სტრიქონების შეკრების ბლოკზე შესასვლელთა რაოდენობა არის 9. ამ შესასვლელებზე მონაცემების განთავსება ხდება შემდეგნაირად. პირველ შესასვლელზე მიეწოდება ორთაწრიანი დღეების რიცხობრივი მნიშვნელობა, მეორე შესასვლელზე უნდა ჩაიწეროს კონსტანტა-წერტილი, მესამე შესასვლელზე-ორთაწრიანი თვეების რიცხობრივი მნიშვნელობა, მეოთხე შესასვლელზე უნდა ჩაიწეროს ისევ კონსტანტა-წერტილი, მე-5 შესასვლელზე - წლის ოთხთაწრიანი რიცხობრივი მნიშვნელობა, მე-6 შესასვლელზე უნდა ჩაიწეროს კონსტანტა-დახრილი ხაზის ნიშანი, მე-7 შესასვლელზე-საათების ორთაწრიანი რიცხობრივი მნიშვნელობა, მე-8 შესასვლელზე უნდა ჩაიწეროს კონსტანტა-ორი წერტილი და ბოლოს, მე-9 შესასვლელზე უნდა ჩაიწეროს წუთების ორთაწრიანი რიცხობრივი მონაცემები.

რეალური დროის საათის მართვის Fritzing პროგრამა იხილეთ ფიგ. 32.7-ზე.



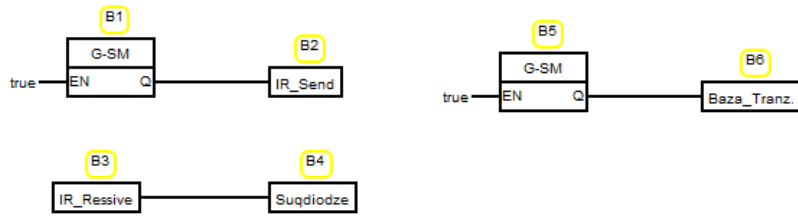
ფიგ. 32.7 რეალური დროის საათის მართვის Fritzing სქემა

პროექტი_33 ინფრაწითელი გადამცემისა და მიმღების ტრანზისტორული მართვა



გადამცემი და მიმღები მოწყობილობები ინფრასწითელი სხივების გამოყენებით ფართოდ გამოიყენება საყოფაცხოვრებო ტექნიკაში: ტელევიზორების, კონდიციონერების და სხვ. მართვაში. ასევე ფართოდ გამოიყენება წარმოების ავტომატიზაციაში. არსებობს მაგალითად ინფრაწითელი გადამწოდების მთელი სპექტრი, რომელთა გარეშეც შეუძლებელია ტექნოლოგიური პროცესების მართვა. ზემოთ მოყვანილ ფოტოზე პირველი და მეორე გამოსახულება წარმოადგენს ინფრაწითელ გადამცემსა და მიმღებს, მესამე n-p-n ტიპის ტრანზისტორია, ხოლო მეოთხე კი-ბუზერი (წრიპინა).

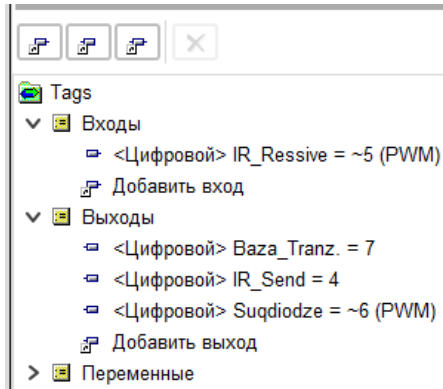
დასაწყისში განვიხილოთ ინფრაწითელი გადამცემისა და მიმღების წყვილის ერთობლივ მუშაობა (ფიგ. 33.1). ეს ელემენტები უშუალოდ FLProg პროგრამაში არ ჩანს, არ არიან ისინი ასახულნი პროგრამის ბიბლიოთეკაში. პროექტში მხოლოდ არდუინოს პინებია განსაზღვრული, რასთანაც დაკავშირებულია ამ მიმღებ გადამცემის გამოსასვლელი.



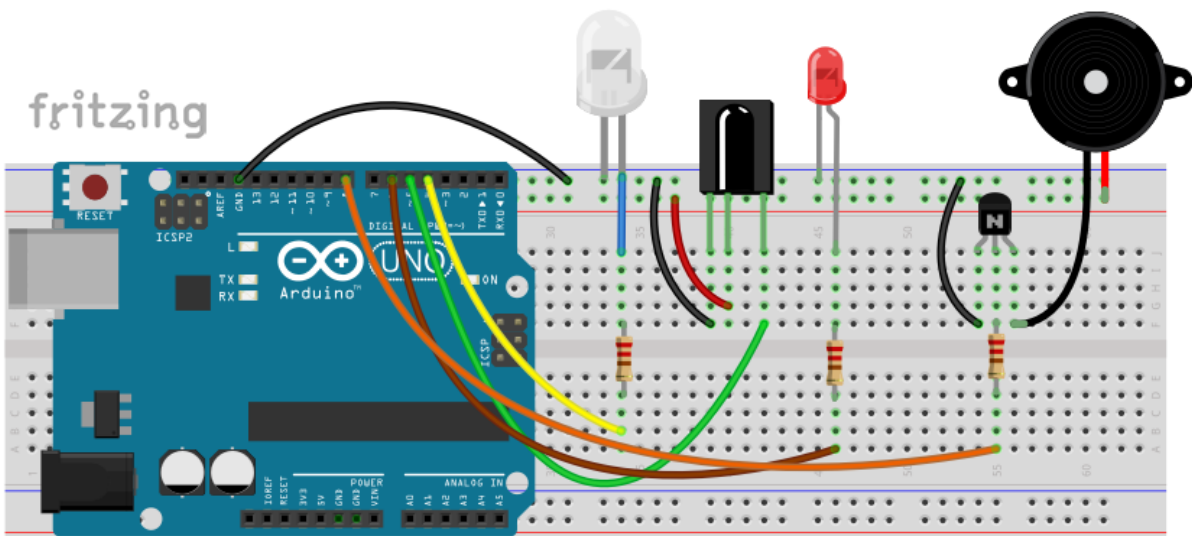
ფიგ. 33.1 ინფრაწითელი გადაცემა მიღების მართვა და ტრანზისტორული მართვის პროგრამა

ინფრაწითელი გადაცემის სამართავად აღებული გვაქვს სიმეტრიული მულტივიბრატორი (იმპულსების გენერატორი) B1, რომელიც მუშაობს 20-20 მიკროწამი იმპულსისა და პაუზის ხანგრძლიობებით. ეს იმპულსები B2 ბლოკის გავლით, ანუ არდუინოს მე-4 კონტაქტის გავლით მიეწოდება ინფრაწითელი სხივების გამომსხივებელს, რომელიც სხვა არაფერია, თუ არა ჩვეულებრივი შუქდიოდი, რომელიც გამოსხივებს სხივებს გარკვეული ტალღის სიგრძის დიაპაზონში (თვალთ უხილავი ინფრაწითელი სხივები). აღნიშნული სხივები დამიზნებულია ინფრაწითელი მიმღებზე, რომელიც რეაგირებს ამ სხივებზე და გამოიმუშავებს იგივე ინტენსიობის იმპულსებს, რომელიც მიეწოდება არდუინოს შესასვლელ კონტაქტს B3. მიიღებს რა ამ იმპულსებს არდუინო, იგი დაამუშავებს მათ და გაგზავნის თავის გამოსასვლელ კონტაქტზე B4. აქედან კი ეს იმპულსები შემზღუდავი წინააღმდეგობის გავლით მიეწოდება ჩვეულებრივი შუქდიოდის ანოდს და იგი დაიწყებს ციმციმს, მასზედ მოდებული იმპულსების სინქრონულად.

ახლა განვიხილოთ მეორე, ტრანზისტორული სქემის მუშაობა. აქ გამოყენებული გვაქვს მე-2 გენერატორი B5, რომელიც მუშაობს ორი წამი იმპულსისა და ორი წამი პაუზის პერიოდულობით. როდესაც გენერატორის გამოსასვლელზე იმპულსია, მაშინ იგი B6 ბლოკის და წინააღმდეგობის გავლით მიეწოდება ტრანზისტორის ბაზას. ტრანზისტორი გაიხსნება, კოლექტორზე დააჯენს ნულოვან პოტენციალს და მიაწვდის ამ პოტენციალს ბუზერის უარყოფით შესასვლელზე, რომლის დადებითი შესასვლელი მიერთებულია 5 ვოლტთან. ბუზერში გაივლის დენი და იგი დაიწყებს ხმის გამოცემას. როდესაც გენერატორის გამოსასვლელზე ნულია, მაშინ ტრანზისტორი ჩაიკეტება, ბუზერში დენი შეწყდება და ხმის გენერირება დამთავრდება. პროექტის ტეგების ზონის ფანჯარა ნაჩვენებია ფიგ. 33.2-ზე, ხოლო Fritzing სქემა ფიგურა 33.3-ზე.



ფიგ. 33.2 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 33.3 ინფრაწითელი გადამცემისა მიმღების მართვისა და ტრანზისტორული მართვის Fritzing სქემა

პროექტი_34

დო, რე, მი, ფა, სოლ, ლია, სი, დო



ქვემოთ ჩამოწერილია პირველი ოქტავის შვიდი ტონის სიხშირეები და პერიოდები:

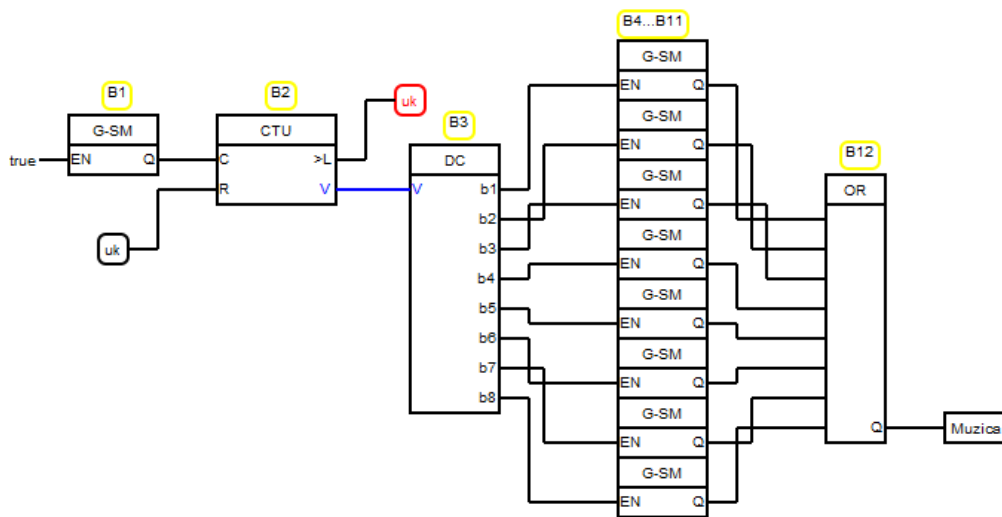
ტონი

სიხშირე (ჰც)

პერიოდი (მკს)

დო	261.63	3822
რე	293.67	3405
მი	329.63	3033
ფა	349.23	2863
სოლ	391.00	2557
ლია	440.00	2272
სი	493.90	2024
დო	523.25	1911

ჩვენი ამოცანა იქნება შევქმნათ FLProg პროგრამა, რომელიც მორიგეობით თითო წამის განმავლობაში გამოსცევს ბერას სათითაოდ აღნიშნული სიხშირეებით. ამ მიზნით გადმოვიტანოთ ბლოკების ბიბლიოთეკის ტაიმერების საქალაქედან-Таимеры FLProg-ის სამუშაო ზონაში გენერატორის ბლოკი-[Generator] B1 (ფიგ. 34.1).



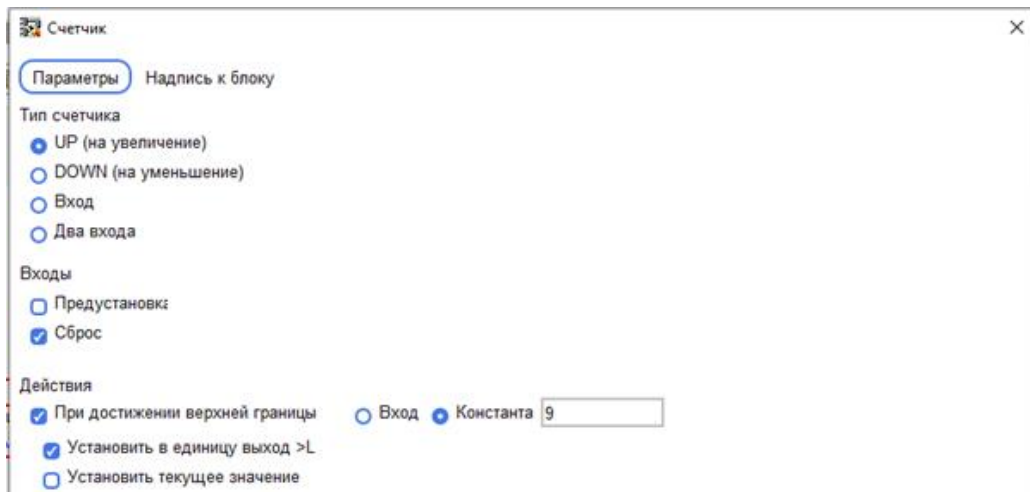
ფიგ. 34.1 დო, რე, მი, ფა, სოლ, ლა, სი, დო-ს პროგრამა

დავაპროგრამიროთ ეს გენერატორი განუწყვეტელი მუშაობის რეჟიმში (შესასვლელზე დავსვათ True). გენერატორის ტიპად ავირჩიოთ სიმეტრიული მულტივიზრატორი, იმპულსის ხანგრძლივობით 500 მილიწამი და რადგან გენერატორი სიმეტრიულია, ამიტომ პაუზაც 500 მილიწამი იქნება, პერიოდი კი გამოვა ერთი წამის ტოლი. (ფიგ. 34.2).



ფიგ. 34.2 გენერატორის პარამეტრიზაცია

ახლა სამუშაო ბლოკების ბიბლიოთეკის საქაღალდედან-Счетчики სამუშაო ზონაში გადმოვიტანოთ მთვლელის ბლოკი-Счетчик В2. რადგან ტონების რაოდენობა პირველ ოქტავაში შვიდია და სასურველია ავიღოთ პლიუს ერთი ტონი-დო შემდეგი ოქტავიდან და კიდევ სასურველია, რომ ავიღოთ კიდევ ერთი პოზიცია სიჩუმისთვის ახალი ციკლის დაწყებამდე, ამიტომ დავაპროგრამიროთ ეს მთვლელი რვამდე ჩათვლით მუშაობაზე. ამ მიზნით თეგების ზონაში შევქმნათ ცვლადი sk , რომელსაც გამოვიყენებთ გენერატორის უკუ კავშირის ფორმირებისათვის. დავაპროგრამიროთ ეს მთვლელი ფიგ. 34.3-ზე წარმოდგენილი პარამეტრიზაციის ფანჯრის შესაბამისად.



ფიგ. 34.3 მთვლელის პარამეტრიზაცია

ტეგების ზონიდან გადმოვიტანოთ ჩვენს მიერ შექმნილი ცვლადი sk ორჯერ სამუშაო ზონაში, ერთი მიუერთოთ მთვლელის $>L$ გამოსასვლელს, მეორე კი R შესასვლელს (ფიგ. 1). ახლა ბლოკების ბიბლიოთეკის ზონის ბიტებთან ოპერაციის საქაღალდედან-Операции с

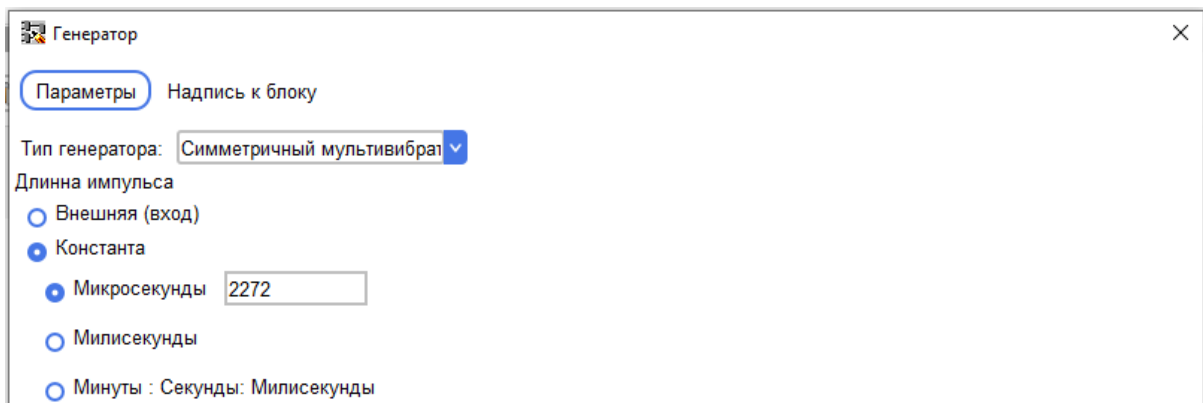
ბიტაში, სამუშაო ზონაში გადმოვიტანოთ დეშიფრატორის ბლოკი-Дешифратор В3, ავწყლოთ იგი რვა გამოსასვლელზე, ტონების მიხედვით (ფიგ. 34.4),



ფიგ. 34.4 დეშიფრატორის გამოსასვლელების არჩევა

ამის შემდეგ, მთვლელის ანალოგური გამოსასვლელი V შევუერთოთ დეშიფრატორის ერთსახელა შესასვლელს. როდესაც გენერატორი დაიწყებს უწყვეტად მუშაობას, ხოლო მთვლელი დაიწყებს იმპულსების გადათვლას 0-დან და 8-მდე, მე-9 იმპულსის მოსვლისას, რადგან ჩვენ კონსტანტის ველში ჩაწერილი გვაქვს 9 (ფიგ. 34.3) მთვლელის >L გამოსასვლელზე დადგება ლოგ. „1“, რომელიც უკუკავშირის uk გავლით მაშინვე მიეწოდება მთვლელის ნულზე დაყენების შესასვლელს R-ს და დააყენებს მას ნულოვან მდგომარეობაში. ამ მდგომარეობაში იგი იმყოფება შემდეგი იმპულსის მოსვლამდე და ა.შ.შ. პროცესი განმეორდება დაუსრულებლად. მთვლელის მუშაობის სინქრონულად დეშიფრატორის გამოსასვლელზეც ლოგ. „1“-ანები მორიგეობით გამოვა b1, b2 და ა.შ.შ. b8 და ამის შემდეგ გამოსასვლელებზე, შემდეგ ისევ b1-ზე და ა.შ.შ.

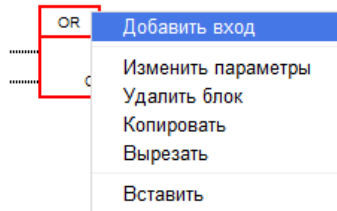
ახლა ჩვენ დაგვჭირდება რვა სიმეტრიული გენერატორის შექმნა. გადმოვიტანოთ რვა გენერატორი ბლოკების ბიბლიოთეკიდან და განვალაგოთ ისე, როგორც ეს ფიგ. 1-ზეა წარმოდგენილი (B4 ... B11).



ფიგ. 34.5 გენერატორის ბლოკის პარამეტრიზაცია „ლია“ ტონის გათვალისწინებით დავაპროგრამოთ ეს ბლოკები ტონების სიხშირეების (პერიოდების) გათვალისწინებით. ფიგ. 34.5-ზე წარმოდგენილია B9 ბლოკის პარამეტრიზაცია „ლია“ ტონის შესაბამისობაში.

დანარჩენი ბლოკების აწყობაც უნდა მოხდეს ანალოგიურად ამ ბლოკისა სხვა ტონებთან შესაბამისობაში.

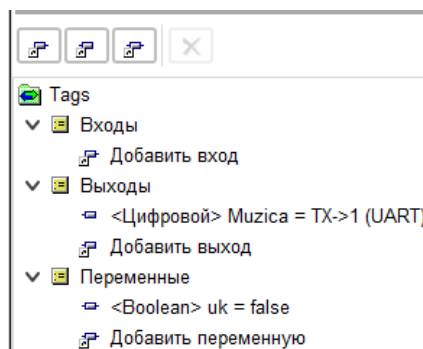
დეშიფრატორის გამოსასვლელები b1...b8 შევართოთ გენერატორის B4...B11 ბლოკების EN შესასვლელებთან, ხოლო იმავე გენერატორის ბლოკების გამოსასვლელები შევართოთ ლოგიკური OR ბლოკის B12 შესასვლელებთან, რომელიც დაპროგრამებულია 8 შესასვლელზე შესასვლელის დამატების Добавить вход მეხანიზმის გამოყენებით (ფიგ. 34.6).



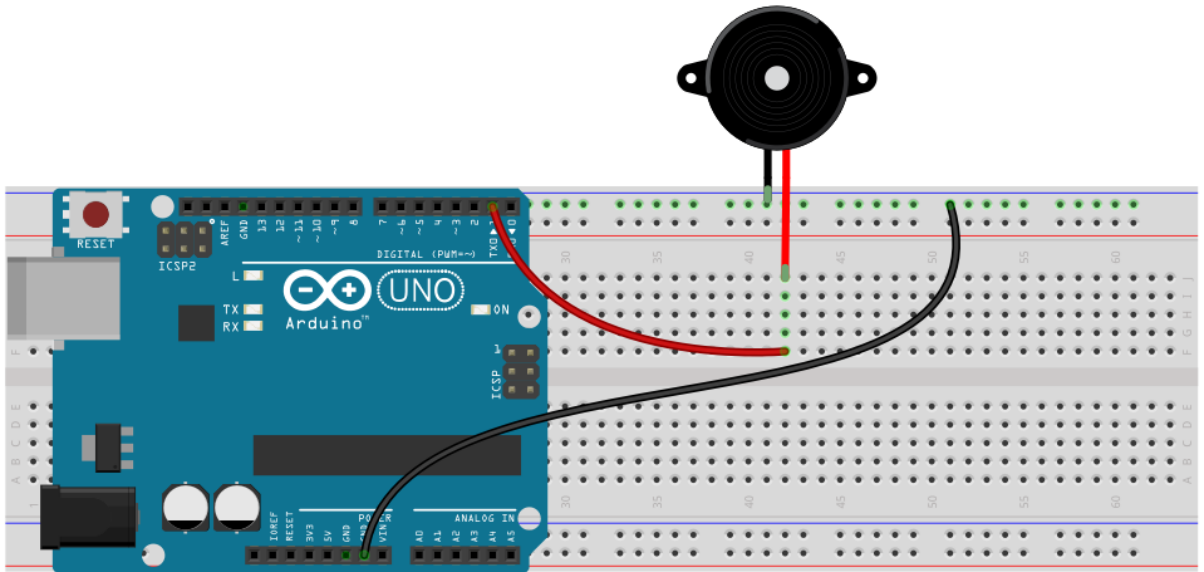
ფიგ. 34.6 ლოგიკურ ბლოკზე შესასვლელების დამატების მექანიზმი

B12 ბლოკში ხდება მაზედ მოდებული გენერატორების სიგნალების ლოგიკური შეკრება. ამ ბლოკის გამოსასვლელი კი არდუინოს პირველი კონტაქტის გავლით მიეწოდება ბუზერს.

FLProg-ში დამუშავებული ამ პროგრამის კომპილაციისა და არდუინოს კონტროლერში ჩატვირთვის შემდეგ გენერატორი B1 მაშინვე დაიწყებს სიგნალების გენერირებას 1 ჰერცის სიხშირით, რომლებსაც ითვლის მთვლელი B2 რვა-რვად, დეშიფრატორის ბლოკი ახდენს მორიგეობით და წანაცვლებით 8 ლოგ. „1“ სიგნალის გამომუშავებას თავის გამოსასვლელებზე (ჯერ პირველზე, მერე მეორეზე და ა.შ.შ.). დეშიფრატორის ეს გამოსასვლელები ნებას რთავს B4...B11 გენერატორების ბლოკებს მუშაობაზე ასევე მორიგეობით და წანაცვლებით, საბოლოოდ B12 ბლოკის გამოსასვლელზე ხდება მორიგეობით დო, რე, მი, ფა... ტონის შესაბამისი სიხშირეების გამოსვლა ერთი წამის მონაცვლეობით და ამ შერეული სიგნალის მიწოდება ბუზერზე. პროექტის ტეგების ზონის ფანჯარა ნაჩვენებია ფიგ. 34.7-ზე, ხოლო Fritzing სქემა-ფიგ. 34.8-ზე.



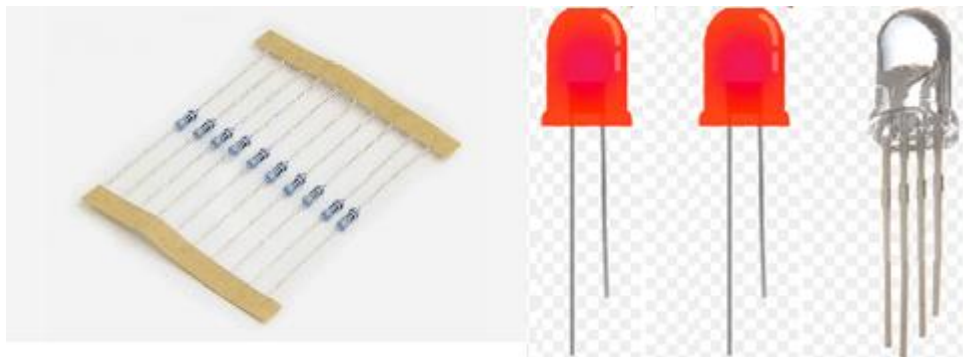
ფიგ. 34.7 პროექტის ტეგების ზონის ფანჯარა



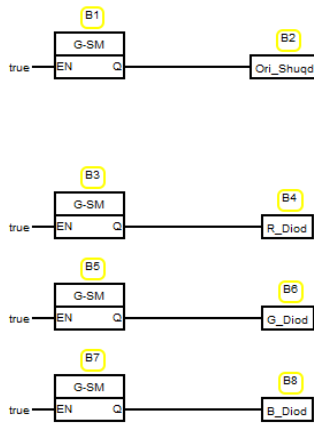
ფიგ. 34. მოწყობილობის მართვის Fritzing სქემა

პროექტი_35

ორი ერთეულოვანი და ერთი RGB შუქდიოდები

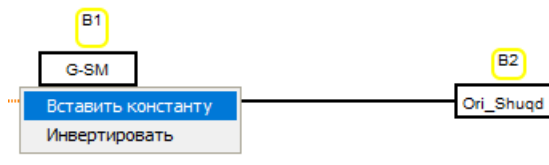


ქვემოთ ფიგ. 35.1-ზე წარმოდგენილია ორი ურთიერთ საწინააღმდეგო ფაზაში მომუშავე ერთეულოვანი შუქდიოდებისა და კომბინირებული სამფერო RGB შუქდიოდის მართვის სქემები FLProg პროგრამაში. მოწყობილობების დაპროგრამება ხდება შემდეგნაირად.



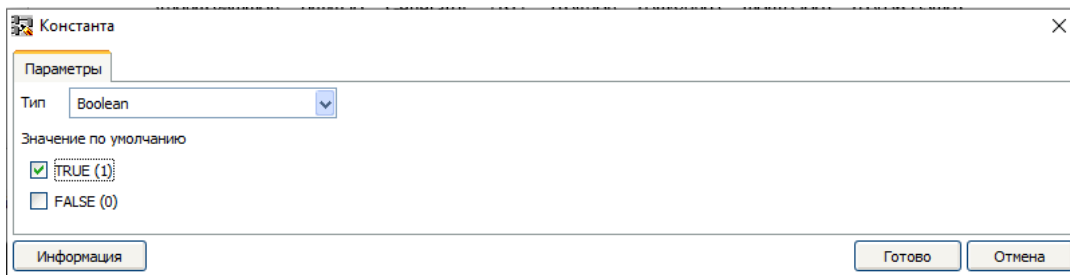
ფიგ. 35.1 ორი შუქდიოდისა და ერთი RGB შუქდიოდის მართვის პროგრამა

განვიხილოთ ორი შუქდიოდის დაპროგრამება. გავუშვათ FLProg პროგრამა, ავირჩიოთ სამუშაო ენად FBD ენა, ხოლო არდუინოს ტიპად Arduino Uno. FLProg პროგრამის ბლოკების ბიბლიოთეკის ტაიმერების საქაღალდედან-Таймеры სამუშაო ზონაში გადმოვიტანოთ გენერატორის ბლოკი-Generator (B1). თავის მარჯვენა დილაკით დავაწკაპუნოთ გენერატორის EN შესასვლელზე, გამოსულ ფანჯარაში (ფიგ. 35.2) ავირჩიოთ კონსტანტის ჩასმა-Вставить константу, და შემდეგ გამოსულ ახალ ფანჯარაში მოვნიშნოთ TRUE - ჭეშმარიტება (ფიგ. 35.2).



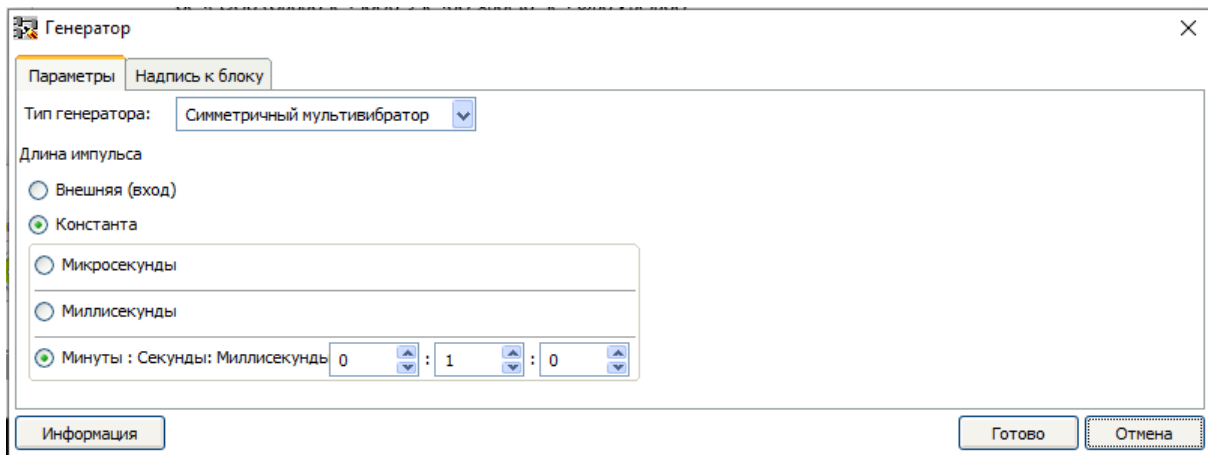
ფიგ. 35.2 გენერატორის კონსტანტის ჩასმის მონიშვნა

ამ პროცედურის დამთავრების შემდეგ გენერატორის ამ EN შესასვლელზე გაჩნდება წარწერა true, რაც იმას ნიშნავს, რომ გენერატორი, როდესაც პროგრამა ჩატვირთული იქნა არდუინოში, იმუშავებს განუწყვეტლივ, პროგრამის მუშაობის იძულებით შეწყვეტამდე, ანდა კვების წყაროს გამორთვამდე.



ფიგ. 35.2 TRUE-ს მონიშვნა

მაგრამ ამით გენერატორის დაპროგრამება ჯერ კიდევ არ დამთავრებულია. საჭიროა მითითებულ იქნას იმ სიგნალების ხანგრძლიობები, ანუ იმპულსისა და პაუზის ხანგრძლიობები, რაც უნდა გამოიმუშაოს გენერატორმა. ამისათვის თავის მარცხენა ღილაკით ვაწკაპებთ ორჯერ გენერატორის (B1) ბლოკის დასახელებაზე G-S, გამოვა გენერატორის პარამეტრების ფანჯარა, სადაც გენერატორის ტიპის ველში-Тип генератора ჩამოშლადი სიიდან ვირჩევთ სიმეტრიულ მულტივიბრატორს-Симметричный мультивибратор (ამ შემთხვევაში გენერატორის გამოსასვლელზე იმპულსის ხანგრძლიობა ტოლი იქნება პაუზის ხანგრძლიობის). იმპულსის სიგრძის ჩამრთველებში-Длина импульса ჩავრთავთ კონსტანტას-Константа (დავაწკაპუნებთ მასზედ თავის მარცხენა ღილაკით), შემდეგ ჩავრთავთ წუთების ჩამრთველს-Минуты და წამების შუა ველში-Секунды ჩავწერთ 1-ს.

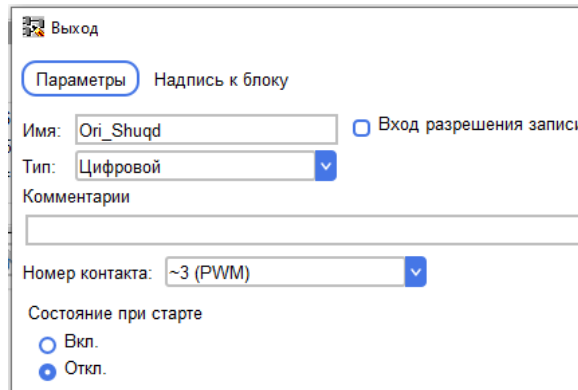


ფიგ. 35.3 გენერატორის პლოკის პარამეტრიზაცია

ეს ნიშნავს იმას, რომ გენერატორის გამოსასვლელზე მისი ამუშავების შემთხვევაში, გამოვა იმპულსისა და პაუზის თანმიმდევრობები თვითოეული 1 წამის ხანგრძლიობით. აქვე შევნიშნავთ, რომ იმპულსის ქვეშ იგულისხმება ძაბვა +5 ვ, ხოლო პაუზის ქვეშ - ძაბვა 0 ვ, ან თუ ვისაუბრებთ ლოგიკურ ერთეულებში - ლოგიკური ერთიანი და ლოგიკური ნოლი, ანუ ლოგ. „1“ და ლოგ. „0“. ამის შემდეგ დავაჭერთ მზადყოფნის ღილაკს - Готово თავის მარცხენა ღილაკით, რაც გამოიწვევს ამ ფანჯრის გაქრობას და გენერატორის დაპროგრამების დამთავრებას. ასევე დავაპროგრამებთ B3, B5 და B7 გენერატორებს, იმ განსხვავებით, რომ პირველში იმპულსების ხანგრძლიობას ვიღებთ 2 წმ-ს, მეორეში 3 წმ-ს, მესამეში კი 4 წმ-ს.

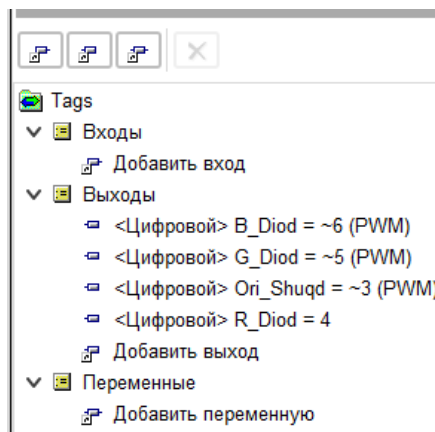
ამის შემდეგ გადავიდეთ ტეგების ველში და შევქმნათ გამოსასვლელები. ამისათვის დავაწკაპოთ გამოსასვლელის დამატების ნიშანზე-Добавить Выход ორჯერ თავის მარცხენა ღილაკით და გამოსულ დაფის გამოსასვლელის ფანჯარაში-Выход платы პარამეტრების ჩანართში-Параметры სახელის ველში-Имя ვწერთ ჩვენთვის მოსახერხებელ რაიმე სახელს,

ჩვენს შემთხვევაში-Ori_Shuqd (ფიგ. 35.4), ტიპის ველში-Тип ჩამოშლადი სიიდან ვირჩევთ ციფრულს-Цифровой, კომენტარების ველში შეგვიძლია არ ჩავწეროთ, ან ჩავწეროთ ნებისმიერი ინფორმაცია, რომელიც მიუთითებს მაგალითად, ამ გამოსასვლელის დანიშნულებას. კონტაქტის ნომრის ველში-Номер константа ვირჩევთ ნებისმიერ ნომერს, რომელსაც ნებას დაგვრთავს პროგრამა (ჩვენს შემთხვევაში მე-3-ს, ანუ Pin3) და ამის შემდეგ თავის მარცხენა ღილაკით ვაჭერთ მზადყოფნის ღილაკს - Готово და ამით გამოსასვლელის შექმნის პროცედურა დამთავრდება.



ფიგ. 35.4 არდუინოს გამოსასვლელის დაპროგრამება

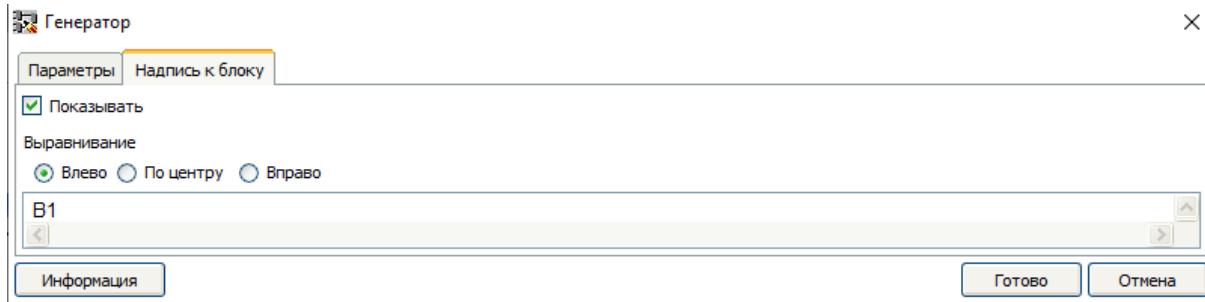
ასევე შევქმნით RGB შუქდიოდის მართვისათვის სამ გამოსასვლელს: R_Diod, G_Diod და B_Diod , ანუ Pin4, Pin5 Pin6 რაც საბოლოო ჯამში გამოიწვევს ტეგების ველში შესაბამისი ჩანაწერების გაჩენას (ფიგ. 35.5).



ფიგ. 35.5 ჩანაწერები ტეგების ველში

ამის შემდეგ, ჩვენ ეს ახლად შექმნილი გამოსასვლელის ბლოკი სახელწოდებით Ori_Shuqd უნდა გადმოვიტანოთ პროგრამის სამუშაო ზოლში, ამისათვის მივიყვანთ თავის მარცხენა ისარს ტეგების არეში ბლოკთან Ori_Shuqd, დავაწკაპებთ თავის მარცხენა ღილაკით, ე.ი.

მოვნიშნავთ მას და შემდეგ ამ თავის ღილაკიდან თითის აუშვებლად გადმოვიტანთ ამ ბლოკს პროგრამის სამუშაო ზონაში, დავსვავთ მას სადმე, მაგალითად B1 ბლოკიდან მარჯვნივ ორ-სამ სანტიმეტრში და აუშვებთ თითს ღილაკს. (ბლოკების დანომვრის წარწერები B1, B2 ... კეთდება შემდეგნაირად, მაგ. B1-ის დაწერისთვის ვაწკაპებთ ამ ბლოკზე და გამომავალ ფანჯარაში გადავდივართ მე-2 ჩანართში, წარწერა ბლოკზე - Надпись к блоку (ფიგ. 35.6), გამოჩენილ ველში ჩავწერთ B1, დავაჭერთ მზადყოფნის ღილაკს - Готово და ეს ჩანაწერი აისახება პირველი გენერატორის თავზედ.



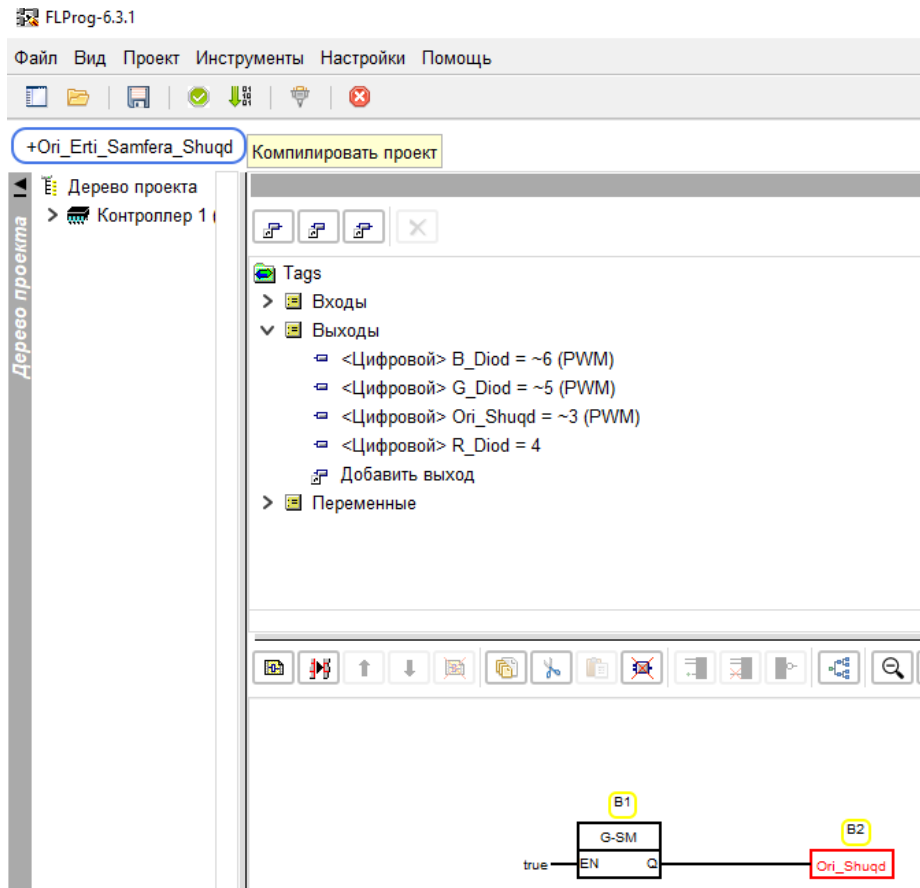
ფიგ. 35.6 წარწერის B1-ის გაკეთება ბლოკის ზემოთ

ახლა, იმისათვის რომ უზრუნველვყოთ გენერატორის სიგნალების გაყვანა არდუინოს ამ გამოსასვლელზე Pin4 უნდა შევასრულოთ გენერატორის გამოსასვლელის მიერთება აღნიშნულ გამოსასვლელთან. ამ მიზნით მიგვყავს თავის მაჩვენებელი გამოსასვლელი B2-ი ბლოკის შესასვლელ ხაზთან (ფიგ. 35.1), როდესაც ეს ხაზი ფერს შეიცვლის და გახდება წითელი ვაჭერთ თავის მარცხა ღილაკს თითს და გადავადგილებთ B1 ბლოკის გამოსასვლელისაკენ. ამ გადაადგილების შესრულების დროს მაჩვენებელ ისარს უნდა გამოჰყვეს ხაზი, რომელსაც მივიყვანთ ამ პირველი გენერატორის გამოსასვლელ ხაზთან. როცა ეს უკანასკნელიც შეიცვლის ფერს და გახდება წითელი ფერის, აუშვებთ თითს თავის ღილაკს და დავინახავთ რომ გენერატორის გამოსასვლელი შეერთებული იქნება არდუინოს Pin4 კონტაქტზე შავი ხაზით, რაც ნიშნავს იმას, რომ გენერატორის ამუშავების შემთხვევაში, მისი გამომავალი სიგნალები გამოვლენ არდუინოს Pin4 კონტაქტზე.

ასევე, მსგავსად ზემოთაღწერილისა, უნდა მოვახდინოთ R_Diod, G_Diod, B_Diod ბლოკების ჩამოტანა პროგრამის სამუშაო ზონაში და განლაგება B3, B5 და B7 გენერატორების გასწვრივ. შემდეგ კი მათაც ისევე შევაერთებთ არდუინოს გამოსასვლელ კონტაქტებთან (B4, B6 და B8 ბლოკებთან) როგორც B1 და B2 ბლოკები შევაერთეთ.

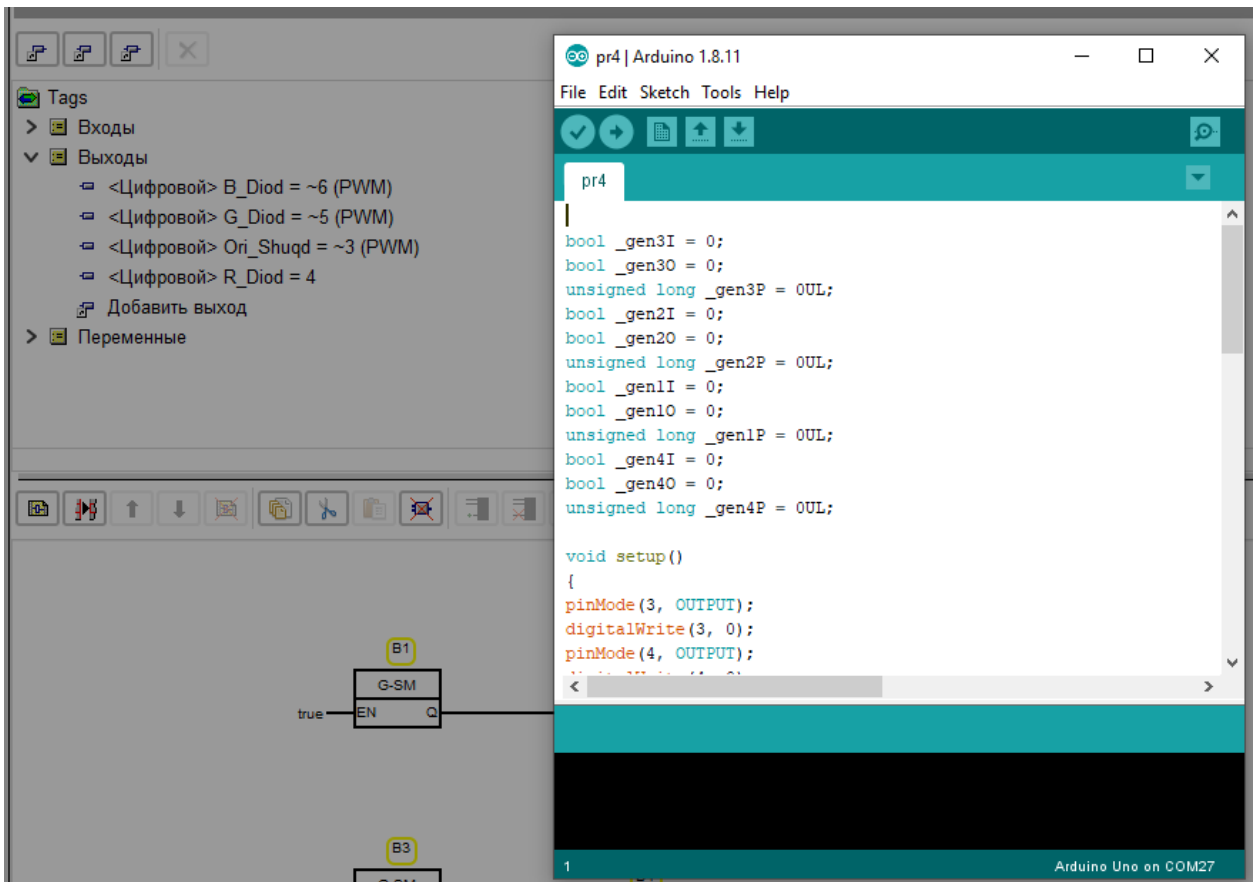
ამრიგად, ჩვენ მივიღეთ ოთხ სხვადასხვა სიხშირეზე მუშაობისათვის დაპროგრამებული გენერატორი, რომელთა გამოსასვლელებიც შეერთებულია არდუინოს

ოთხ გამოსასვლელ კონტაქტთან. ახლა საჭიროა ამ ჩვენი დამუშავებული FLProg პროგრამის ჩატვირთვა არდუინოში. ამისათვის თავის მარცხენა ღილაკით ვაჭერთ პროექტის კომპილირების მწვანე მსხვილ, ქვემოთმართულ კომპილიაციის ღილაკს თავის მარცხენა ღილაკით - Компилировать проект (ფიგ. 35.7).

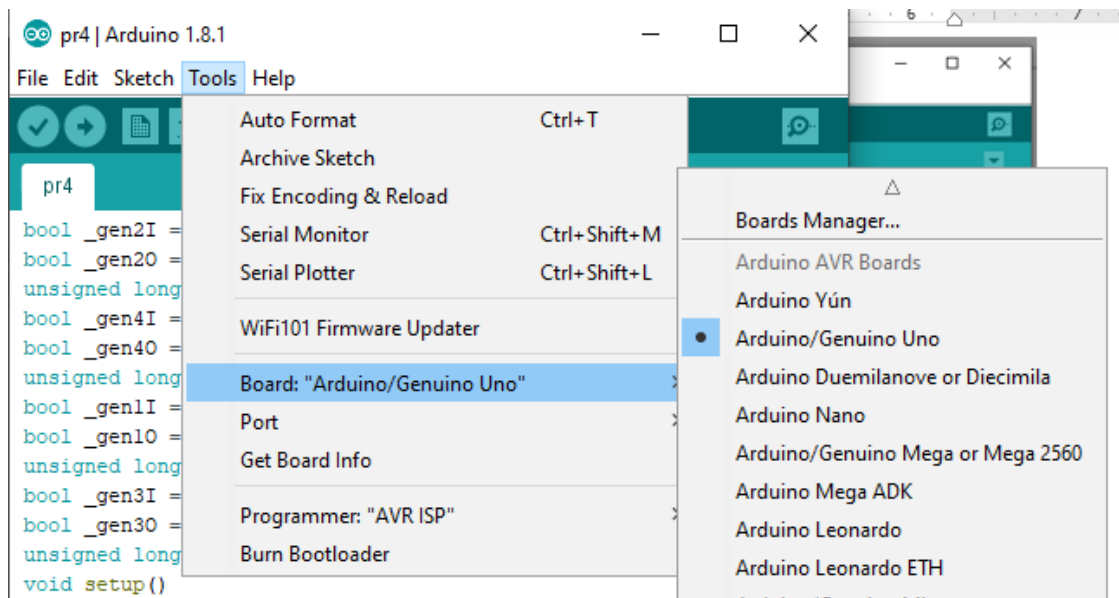


ფიგ. 35.7 პროექტის კომპილირების ღილაკი

კომპილიაციის პროცესში ხორციელდება ჩვენს მიერ FLProg-ში დამუშავებული პროგრამის გადათარგმნა არდუინოს კომპანის მიერ შემოთავაზებულ დამუშავების გარემოში Arduino IDE (ფიგ. 35.8), სადაც რიცხვითი მნიშვნელობა 1.8.11 აღნიშნავს ამ პროგრამის ვერსიას და შეიძლება იყოს სხვა მნიშვნელობის.



ფიგ. 35.8 FLProg პროგრამის კომპილაცია Arduino-ს პროგრამაში

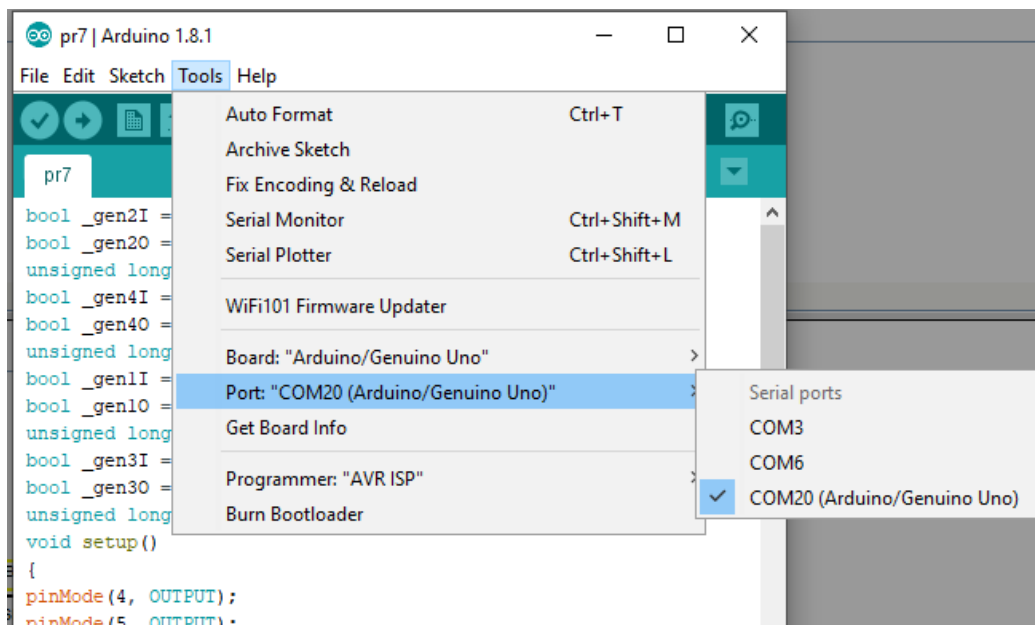


ფიგ. 35.9 არდუინოს მოდელის შემოწმება

ამ ახალ ფანჯარაში უნდა შევამოწმოთ ორი კომპონენტი. დავაჭიროთ ხელი ღილაკს Tools და გამომაგალ ფანჯარაში ჩამოშლადი სიიდან:

1) შევამოწმოთ მონიშნულია თუ არა Board: „Uno Arduino/Genuio” (ფიგ. 35.9). თუ მონიშნულია არდუინოს სხვა რომელიმე მოდელი, მაშინ უნდა მოვნიშნოთ „Uno Arduino/Genuio“, რადგან ჩვენს მიერ დასაწყისშივე ეს მოდელი იქნა არჩეული;

2) შევამოწმოთ ჩანაწერი Port და დავრწმუნდეთ მინიჭებული აქვს თუ არა ჩვენს მიერ გამოყენებულ არდუინოს ვირტუალური ნომერი (Port “Com20 (Arduino/Genuio Uno)” და არის თუ არა ეს ვირტუალური ნომერი მონიშნული მარჯვნივ ალმით (Port “Com20 (Arduino/Genuio Uno)” (ფიგ. 35.10), ჩვენს შემთხვევაში აქ Com პორტის ნომრად შეიძლება სხვა რიცხვი იყოს მითითებული.

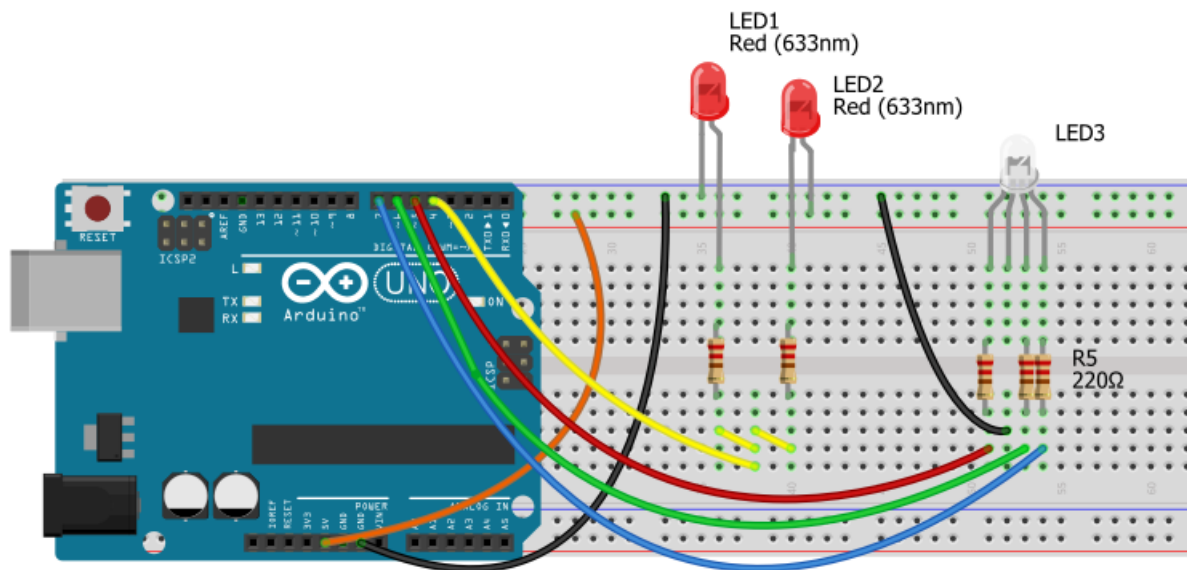


ფიგ. 35.10 ვირტუალური ნომრის მინიჭება არდუინოს პორტს

განვიხილოთ მუშაობის პრინციპი. პირველად ავხსათ ორი შუქდიოდის ერთობლივი მუშაობა. როგორც უკვე ვიცით, პროგრამის არდუინოში ჩატვირთვის შემდეგ, ეს უკანასკნელი მაშინვე იწყებს მასში ჩაწერილი პროგრამის შესრულებას. პირველ რიგში ჩვენ გვინტერესებს B1 გენერატორის მუშაობა. იგი ჩაწერილი პროგრამის შესაბამისად იწყებს იმპულსების გენერირებას თავის გამოსასვლელზე 1 წამი იმპულსისა და 1 წამი პაუზის თანმიმდევრობით და უწყვეტად. ეს იმპულსები არდუინოს Pin3 კონტაქტის გავლით გადაეცემა LED1 და LED2 შუქდიოდების შემზღუდავ რეზისტორებს. როდესაც Pin3 კონტაქტზე პლიუს 5 ვოლტია, მაშინ LED1 ჩართული ძაბვაზე პირდაპირი მიმართულებით, ხოლო LED2 უკუ მიმართულებით. შესაბამისად ამისა LED1-ში გაივლის დენი წრედით Pin3, რომელზედაც ხუთი ვოლტი ძაბვაა, შემზღუდავი წინააღმდეგობა, შუქდიოდის ანოდი, შუქრიოდის კათოდი, არდუინოს მიწის კონტაქტი (GND) და იგი აინთება, ხოლო LED2-ში დენი არ გაივლის, რადგან მის ანოდზე

მიწა ხოლო კოდზე 5 ვოლტი, ამიტომ მასზე მოდებული ძაბვა საწინააღმდეგოდ არის ორიენტირებული, იგი ჩაკეტილი და ჩამქვრალია. როდესაც Pin3 კონტაქტზე დაჯდება პაუზა, ანუ ნოლი ვოლტი, მაშინ LED1-ი უკუღმა იქნება ჩართული ხოლო LED2 პირდაპირ. შესაბამისად ამისა ეს პირველი შუქდიოდი ჩაკეტილი აღმოჩნდება, მასში დენი არ გადის და იგი ჩამქვრალი იქნება, ხოლო LED2 შუქდიოდი ღიაა დენის მიმართ მასში დენი გაივლის და იგი აინთება. ამრიგად ეს ორი შუქდიოდი იმუშავებენ ურთიერთსაწინააღმდეგო ფაზაში, როცა ერთი ანთებულია მაშინ მეორე ჩამქვრალია და პირიქით და ანთება ჩაქრობის დროის ინტერვალები იქნება 1 წამის ტოლი.

ანალოგიურად მუშაობს RGB შუქდიოდიც, ამ დიოდს აქვს სამი ანოდი და ერთი კათოდი. როდესაც ანოდებზე იმპულსებია მაშინ ისინი აინთება. როდესაც მაგალითად R შესასვლელზეა იმპულსი და დანარჩენზე პაუზები, მაშინ შუქდიოდი ანთია წითელ ფერად, როდესაც G-ზეა იმპულსი, ხოლო დანარჩენზე პაუზები, მაშინ შუქდიოდი ანთია მწვანე ფერად, როდესაც B-ზეა იმპულსი დანარჩენებზე კი პაუზა, მაშინ შუქდიოდი აინთება მწვანე ფერად. თუ რომელიმე ორზე ან სამთავე შესასვლელზე მიეწოდება იმპულსები, მაშინ შუქდიოდები გამოასხივებს შესაბამის ფერებს, მაგრამ საბოლოოდ მიიღება ამ ფერების ნარევი ნათება. შუქდიოდების მართვის Fritzing სქემა მოყვანილია ფიგ. 35.11-ზე

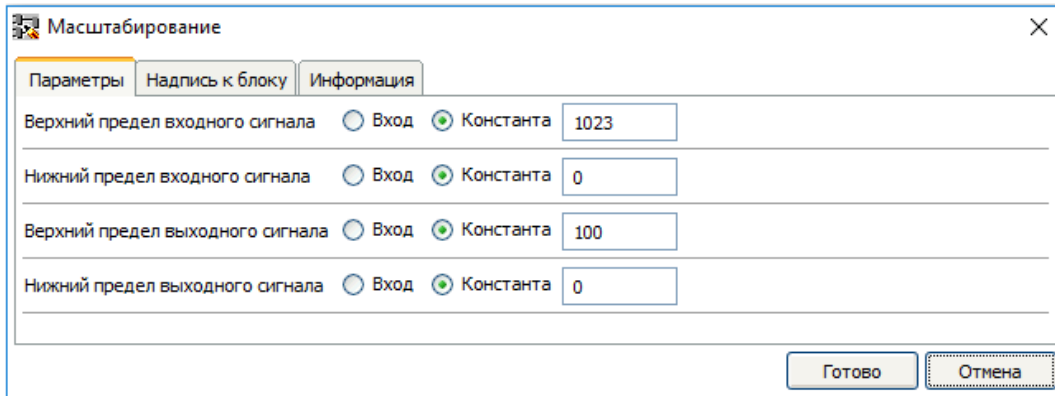


ფიგ. 35.11 პროექტის ელექტრულ-სამონტაჟო Fritzing სქემა

ფიგ. 36.2 სენსორის შესასვლელის პარამეტრიზაცია

ტიპის ველში-Тип: ვირჩევთ ანალოგიურს-Аналоговый, რადგან აღნიშნული სენსორი იძლევა ანალოგურ სიგნალს თავის გამოსასვლელზე, კონტაქტის ნომრის ველში-Номер контакта: ვწერთ ნებისმიერ ნომერს, რომლის შესაძლებლობასაც გვაძლევს პროგრამა არჩეული არდუინოს ტიპისთვის, (A2 კონტაქტი ჩვენს შემთხვევაში), შემდეგ ვაჭერთ მზადყოფნის ლილაკს- Готово, შემდეგ კი ეს შესასვლელი ტეგების ველიდან თავის მარცხენა ლილაკის ჩავლებით გადმოგვაქვს იგი პროგრამის სამუშაო ზონაში (ბლოკი B1 ფიგ. 36.1).

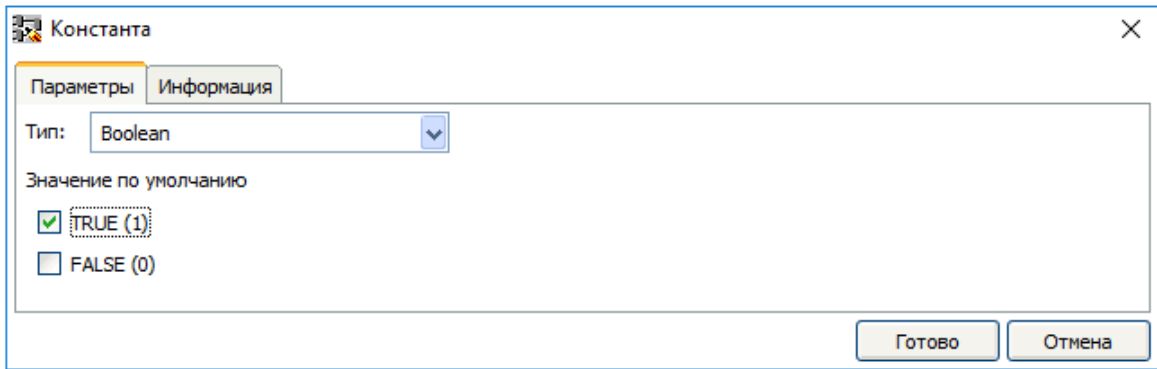
როგორც ცნობილია ეს სენსორი თავის გამოსასვლელზე, ტემპერატურისაგან დამოკიდებულებით, იძლევა ძაბვას 0-დან +5 ვოლტამდე, რაც შეესაბამება ტემპერატურის მნიშვნელობებს 0-დან 100 გრადუსამდე. ძაბვის ეს მნიშვნელობები შედის არდუინოს ანალოგურ შესასვლელზე გარდაქმნის მას ციფრულ ფორმატში და გამოიმუშავებს შესაბამის რიცხვით მნიშვნელობებს 0-დან 1023-მდე. ანუ ჩვენი პროგრამის შემთხვევაში სწორედ რომ B1 ბლოკი იძლევა თავის გამოსასვლელზე ტემპერატურის შესაბამის რიცხვით მნიშვნელობებს 0-დან 1023-მდე. (ტემპერატურა იცვლება 0-დან 100 გრადუსამდე, B1 ბლოკის გამოსასვლელი იცვლება 0-დან 1023-მდე შესაბამისად). ახლა, რომ მივიღოთ ტემპერატურის ადეკვატური მნიშვნელობა საჭიროა B1 ბლოკის გამოსასვლელზე შევასრულოთ გარკვეული მათემატიკური ოპერაციები. ამ ფუნქციას ასრულებს მასშტაბირების ბლოკი SCALE ბლოკი B2, რომელიც გადმოგვაქვს ელემენტების ბიბლიოთეკის-Библиотека элементов მასშტაბირების საქალაქედან- Масштабирование ბლოკი (Scale). თუ დავაწკაპებთ ამ ბლოკზე გაიხსნება მისი პარამეტრიზაციის ფანჯარა (ფიგ. 36.3), სადაც დასაპროგრამირებელია ორი დიაპაზონი:



ფიგ. 36.3 მასშტაბირების ბლოკის პარამეტრიზაციის ფანჯარა

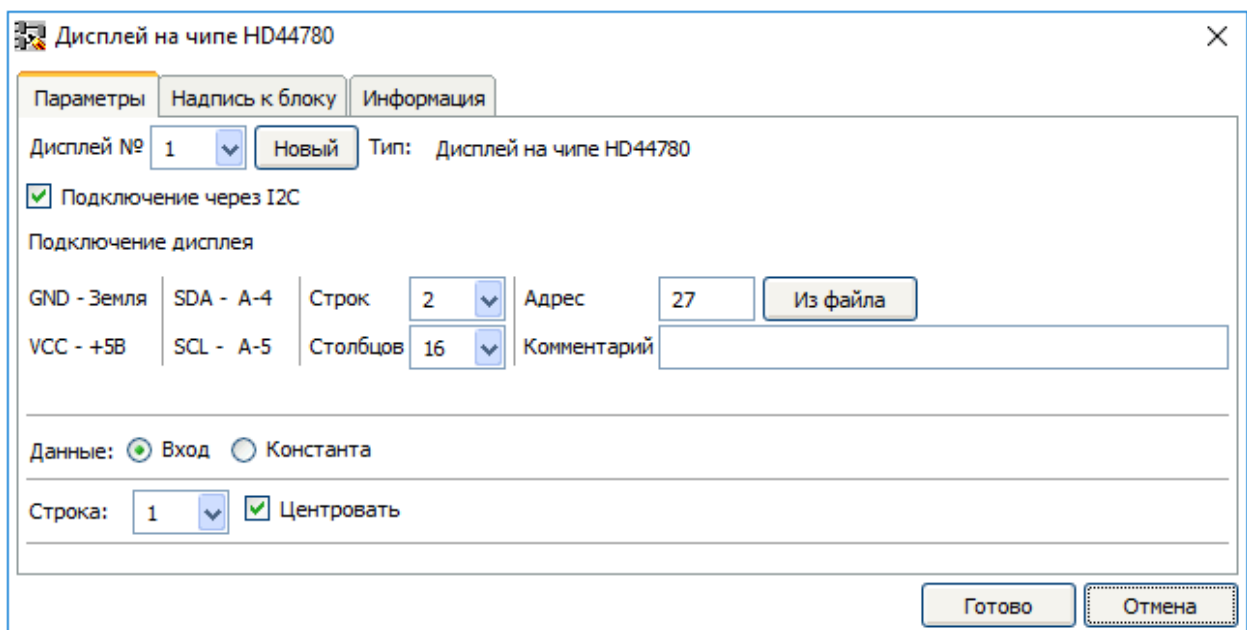
შესასვლელი სიგნალების დიაპაზონი და გამოსასვლელი სიგნალების დიაპაზონი. თვითოეული ეს დიაპაზონი წარმოდგენილია ზედა და ქვედა ზღვრების მნიშვნელობებად. ამის შესაბამისად, ჩვენ შესასვლელი სიგნალის ზედა ზღვარში- Верхний предел входного сигнала უნდა ჩავრთოთ ჩამრთველი კონსტანტა-Константа და გამოჩენილ ველში ჩავწეროთ 1023, ხოლო შესასვლელი სიგნალის ქვედა ზღვარში- Нижний предел входного сигнала ჩავწეროთ 0. შესაბამისად ამისა, გამოსასვლელი სიგნალის ზედა ზღვარის ველში-Верхний предел Выходного сигнала უნდა ჩავწეროთ 100, ხოლო გამოსასვლელი სიგნალის ქვედა ზღვარში-Нижний предел выходного сигнала კი 0. ამის შემდეგ ვაწვებით მზადყოფნის ღილაკს და მასშტაბირების ბლოკის დაპროგრამებას ამით დავამთავრებთ.

ახლა საჭიროა ამ გაზომილი და გარდაქმნილი ინფორმაციის ასახვა რაიმე მოწყობილობაზე. ამისათვის ვიყენებთ ჩვენს ხელთ არსებულ მოწყობილობას დისპლეის ჩიპზე HD7780, რომლის შესაბამისი ბლოკიც პროგრამის სამუშაო ზონაში გადმოგვაქვს იგივე ბლოკების ბიბლიოთეკის დისპლეის საქაღალდედან სახელწოდებით დისპლეი ჩიპზე- HD7780 Дисплей на чипе HD7780 (B4). დავაწკაპოთ მის EN შესასვლელზე და გამონათებულ ფანჯარაში ავიჩიოთ კონსტანტის ჩასმა-Вставить константу და შედეგად გამოჩენილ ახალ ფანჯარაში (ფიგ. 36.4) ჩავრთოთ ალამი TRUE (1) და დავაჭიროთ მზადყოფნის ღილაკს-Готово, რაც იმას ნიშნავს, რომ ამ დისპლეის პროგრამის გაშვების მომენტიდან ნება ეძლევა მას ინფორმაციის განუწყვეტელი ინდიცირებისათვის.



ფიგ. 36.4 დისპლეის შესასვლელის პარამეტრიზაცია

ამის შემდეგ, უნდა შევუდგეთ თვითონ დისპლეის ბლოკის პარამეტრიზაციას. ამისათვის დავაწკაპუნოთ ამ ბლოკზე-B4 და გამოჩენილ ფანჯარაში (ფიგ. 36.5) დავაწკაპუნოთ ლილავს ახალი-Новый, რის შედეგადაც დისპლეის ნომრის ფანჯარაში-Дисплей № ავტომატურად ჩაიწერება ერთიანი, ამით ვუჩვენებთ, რომ ჩვენი გამოყენებული დისპლეი იქნება პირველი.

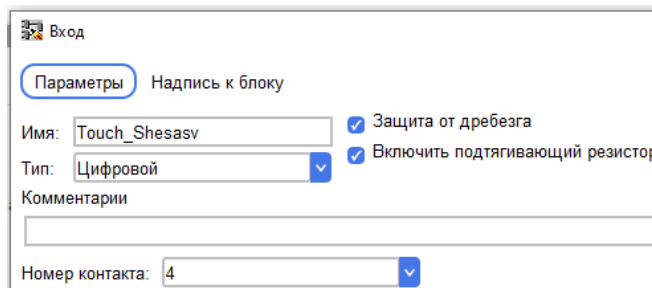


ფიგ. 36.5 დისპლეის ბლოკის პარამეტრიზაციის ფანჯარა

როგორც ფიგურიდან ჩანს, SDA და SCL სიგნალების მიერთების კონტაქტები განსაზღვრულია თვითონ FLProg პროგრამის მიერ სიჩუმით (A4 და A5 შესაბამისად) და ჩვენც უნდა დავეთანხმოთ და გამოვიყენოთ ეს შეერთებები. ასევე ვაერთებთ GND-ს - არდუინოს მიწასთან-Земля და VCC-ს არდუინოს +5 ვოლტთან (+5B). სტრიქონის ველში-Строк და სვეტების ველში-Столцов ვწერთ ჩვენს ხელთ არსებული დისპლეის მონაცემებს (2 და 16 ჩვენ

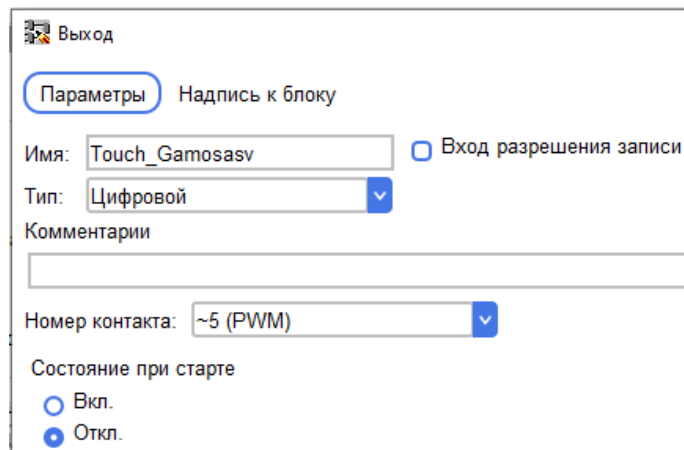
შემთხვევაში). სამისამართო ველში-Адрес ვწერთ ინტერნეტიდან მოძიებულ საიდენტიფიკაციო ნომერს. ჩვენი დისპლეისათვის იგი უდრის 27. ჩავრთავთ ჩამრთველს მონაცემები-Данные, დავაყენებთ ცენტრირების ალამს-Центровать, სტრიქონის ველში ჩავწერთ ერთიანს და შემდეგ ვაჭერთ მზადყოფნის ლილაკს-Готово. ამით დისპლეის ბლოკის პარამეტრიზაცია დათავრებული იქნება. ამის შემდეგ ვახდენთ B1 ... B4 ბლოკების შეერთებას ისე, როგორც ეს გამოსახულია ფიგ. 35.1. -ზე. პროგრამის გაშვების შემთხვევაში დისპლეიზე აისახება ტემპერატურული გადამწოდებიდან გაზომვილი ინფორმაცია. სენსორზე თითების მოკიდებით ვახდენთ გასაზომი ტემპერატურის მატებას რაც შესაბამისად აისახება დისპლეიზე.

ახლა გადავიდეთ Touch სენსორის მართვაზე. ტეგების ზონაში (Tags) გავხსნათ შესასვლელების საქალაღე-Входы და დავაწკაპოთ შესასვლელების დამატებას-Добавить вход. გამოსულ ფანჯარაში (ფიგ. 36.6) სახელის ველში-Имя: ჩავწერთ Touch_Shesasv, ტიპის ველში-Тип: ავირჩიოთ ციფრული-Цифровой, კონტაქტის ნომრის ველში-Номер контакта: ჩავწერთ რაიმე კონტაქტი, რომელიც ნებადართულია პროგრამის მიერ (4, ჩვენ შემთხვევაში), ჩავრთოთ ყანყალისაგან დაცვისა-Защита от дребезга და ამომქაჩავი რეზისტორის-Включить подтягивающий резистор ალმები და დავაწვეთ მზადყოფნის ლილაკს-Готово. ყოველივე ამის შემდეგ Touch სენსორის შესასვლელის პარამეტრიზაცია დამთავრებული იქნება.



ფიგ. 36.6 Touch სენსორის შესასვლელის პარამეტრიზაციის ფანჯარა

ამის შემდეგ გადავიდეთ იგივე ტეგების ზონის გამოსასვლელების საქალაღეში-Выходы, დავაწკაპოთ გამოსასვლელის დამატებას-Добавить выход გამონათებულ ფანჯარაში ჩავწერთ შემდეგი მონაცემები (ფიგ. 36.7). სახელის ველში-Имя: Touch_Gamosasv, ტიპის ველში-Тип: ციფრული-Цифровой, კონტაქტის ნომრის ველში-Номер контакта: – ნებისმიერი კონტაქტი, რომელსაც ნებას გვრთავს პროგრამა (5, ჩვენს შემთხვევაში) და დავაწვეთ მზადყოფნის ლილაკს-Готово. ყოველივე ამის შემდეგ Touch სენსორის პარამეტრიზაცია დამთავრებული იქნება.



ფიგ. 36.7 Touch სენსორის გამოსასვლელის პარამეტრიზაციის ფანჯარა

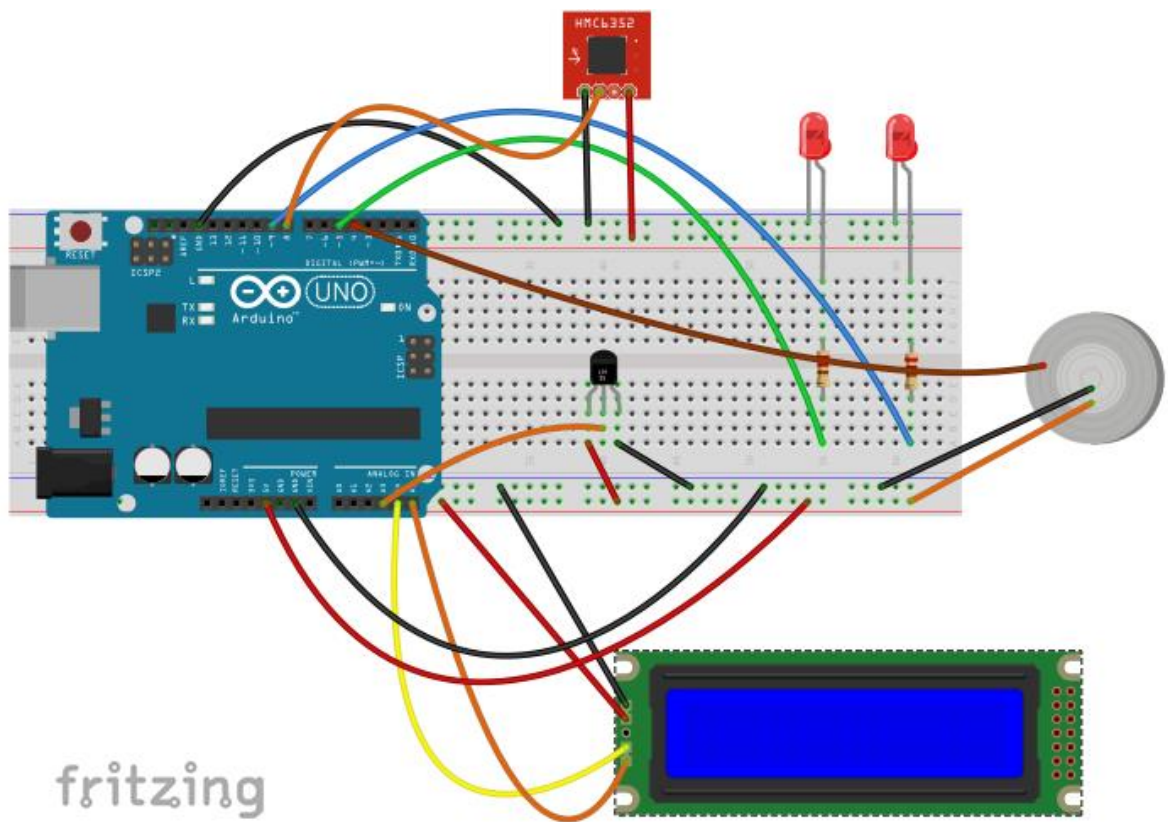
ამის შემდეგ, ეს ახლად შექმნილი შესასვლელი და გამოსასვლელი გადმოგვაქვს პროგრამის სამუშაო ზონაში, ვარქმევთ მათ B5 და B6 სახელებს და ვაერთებთ ერთმანეთთან ისე, როგორც ეს ფიგ. 36.1-ზეა ასახული. აქ ვაკეთებთ კიდევ ერთ ოპერაციას, კერძოდ: ვაწკაპუნებთ B5 ბლოკის გამოსასვლელზე და გამოსულ ფანჯარაში ვირჩევთ ინვერტირებას-Инвертировать, რის შედეგადაც ამ გამოსასვლელზე აისახება პატარა რგოლი, რაც მიუთითებს იმას, რომ ხდება სენსორიდან მიღებული სიგნალის ინვერტირება, ანუ თუ არის ლოგიკური ერთიანი ინვერტირების შემდეგ გახდება ლოგიკური ნული და პირიქით. კონტროლერის მე-5 გამოსასვლელზე უნდა იქნას შეერთებული შუქდიოდი რეზისტორის გავლით. პროგრამის გაშვების შემდეგ, Touch სენსორზე თითის დაჭერის შემთხვევაში, თუ კი პროგრამა სწორად იყო შედგენილი და ელექტრული სქემები სწორად აწყობილი, შუქდიოდი უნდა აინთოს, ხოლო თითის ადების შემთხვევაში - უნდა ჩაქრეს.

ახლა გადავიდეთ Tilt სენსორის მართვაზე. ამ სენსორის მართვა უმეტესწილად ანალოგიურია Tuch სენსორის მართვისა. ტეგების ზონაში გავხსნათ შესასვლელების საქაღალდე-Входы და დავაწკაპუნოთ შესასვლელის დამატებას-Добавить вход. გამოსულ ფანჯარაში სახელის ველში-Имя: ჩავწერთ Tilt_Shesasv, ტიპის ველში-Тип: ავირჩიოთ ციფრული-Цифровой, კონტაქტის ნომრის ველში-Номер контакта: ჩავწერთ რომელიმე კონტაქტი, დასაშვები პროგრამის მიერ (6, ჩვენ შემთხვევაში), ჩავრთოთ ყანყალისაგან დაცვისა-Защита от дребезга და ამომქაჩავი რეზისტორის-Включить подтягивающий резистор ალმები და დავაწვეთ მზადყოფნის ღილაკს-Готово). ყოველივე ამის შემდეგ Tilt სენსორის შესასვლელის პარამეტრიზაცია დამთავრებული იქნება.

ამის შემდეგ გადავიდეთ იგივე ტეგების ზონის გამოსასვლელების საქაღალდეში-Выходы, დავაწკაპუნოთ გამოსასვლელის დამატებას-Добавить выход გამონათებულ

ფანჯარაში ჩავწერთ შემდეგი მონაცემები. სახელის ველში-Имя: Tilt_Gamosasv, ტიპის ველში-Тип: ციფრული-Цифровой, კონტაქტის ნომრის ველში-Номер контакта: – ნებისმიერი კონტაქტი, რომელსაც ნებას გვრთავს პროგრამა (7, ჩვენს შემთხვევაში) და დავაწვეთ მზადყოფნის ლილაკს-Готово. ყოველივე ამის შემდეგ Tilt სენსორის პარამეტრიზაციაც დამთავრებული იქნება.

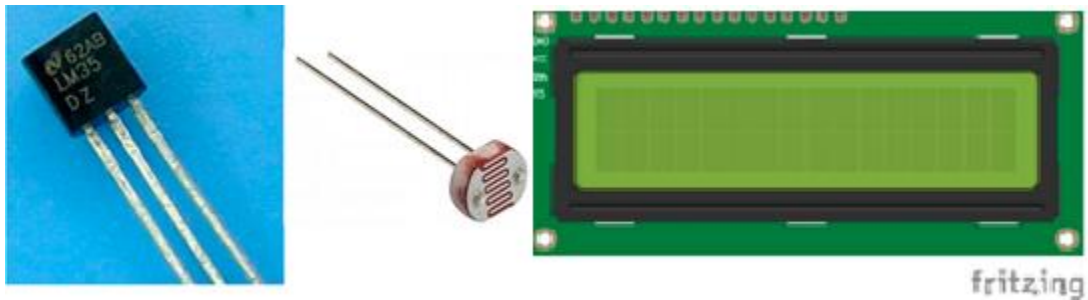
ამის შემდეგ, ეს ახლად შექმნილი შესასვლელი და გამოსასვლელი გადმოგვაქვს პროგრამის სამუშაო ზონაში, ვარქმევთ მათ B7 და B8 ბლოკებს შესაბამისად და ვაერთებთ ერთმანეთთან ისე, როგორც ეს ფიგ. 36.1-ზეა ასახული. აქ ვასრულებთ კიდეც ერთ ოპერაციას, კერძოდ: ვაწკაპებთ B7 ბლოკის გამოსასვლელზე და გამოსულ ფანჯარაში ვირჩევთ ინვერტირებას-Инвертировать, რის შედეგადაც ამ გამოსასვლელზე აისახება პატარა რგოლი, რაც მიუთითებს იმას, რომ ხდება სენსორიდან მიღებული სიგნალის ინვერტირება. კონტროლერის მე-7 გამოსასვლელზე უნდა იქნას შეერთებული შუქდიოდი რეზისტორის გავლით. პროგრამის გაშვების შემდეგ, სენსორის გადახრა-გადმოხრის შემთხვევაში, თუ კი პროგრამა სწორად იყო შედგენილი და ელექტრული სქემები სწორად აწყობილი, შუქდიოდი უნდა აინთოს და ჩაქრეს ხოლმე. ფიგ. 36.8-ზე ნაჩვენებია პროექტის ბლოკ სქემა, რომელიც შესრულებულია Fritzing პროგრამაში.



ფიგ. 36.8 პროექტის ბლოკ სქემა Fritzing პროგრამაში

პროექტი_37

ტემპერატურული და ფოტორეზისტორული გადამწოდები



ჯერ განვიხილოთ LM35 DZ გადამწოდი და მოვახდინოთ მისი გამოსასვლელი სიგნალის გარდაქმნა ცელსიუსის გრადუსებში (დაკალიბრება). სპეციფიკაციიდან გამომდინარე, (საფირმო ტექნიკური მახასიათებლები) იგი გამოსასვლელზე იძლევა ანალოგურ სიგნალს 0-5ვ დიაპაზონში და ტემპერატურის ცვლილებისას პლიუს-მინუს ერთი გრადუსის დიაპაზონში იწვევს მისი გამოსასვლელი სიგნალის ცვლილებას პლიუს მინუს 0.01 ვოლტის (10 მილივოლტი) სიდიდით. აქედან გამომდინარე, რადგან ცნობილია, რომ ამ გადამწოდის გამოსასვლელი მიერთებული უნდა იყოს არდუინოს ერთერთ ანალოგურ შესასვლელთან, რომელიც ასრულებს შესავალი ანალოგური სიგნალის გარდაქმნას ციფრულში (ორობით კოდში) ათ თანრიგიანი ორობითი სიზუსტით, ამიტომ თუ არდუინოს ამ ანალოგურ შესასვლელზე სიგნალი იცვლება 0...5ვ დიაპაზონში, მაშინ არდუინო ახდენს ამ სიგნალის გარდაქმნას 0 ... 1023 დიაპაზონში შესაბამისად, ანუ ახდენს სიგნალის ანალოგურ ციფრულ გარდაქმნას (აცვ).

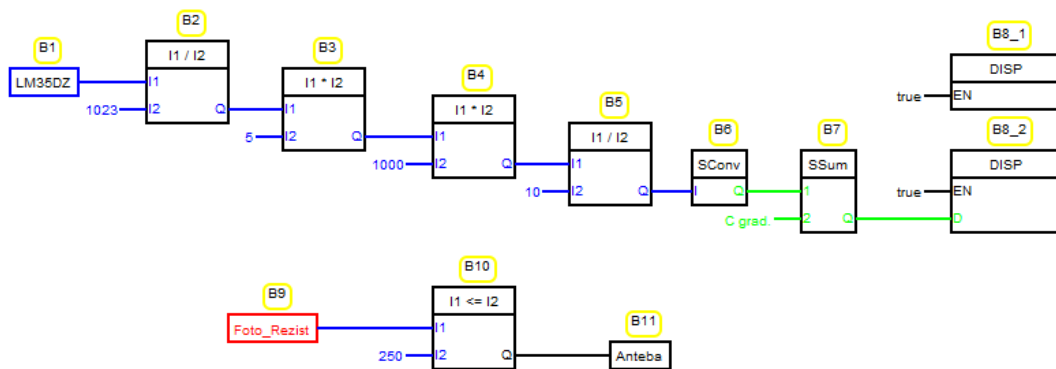
ახლა ჩვენ ამოცანაა დავადგინოთ თუ რამდენი ვოლტია მოცემულ მომენტში გადამწოდის გამოსასვლელზე. ამ მიზნის მისაღწევად გადამწოდის ეს მნიშვნელობა რომელიც იცვლება 0 ... 1023 დიაპაზონში უნდა გავყოთ მაქსიმალურ სიდიდეზე ანუ 1023-ზე და შემდეგ გავამრავლოთ 5-ზე. ამით ჩვენ დავადგენთ თუ რა სიდიდის ანალოგური სიგნალი გამოიმუშავა გადამწოდმა თავის გამოსასვლელზე ვოლტებში.

ამის შემდეგ უნდა დადგინდეს თუ რამდენი მილივოლტია ამ გამოანგარიშებულ ვოლტებში, ამიტომ ამ ვოლტების სიდიდეს ვამრავლებთ 1000-ზე. ამით ჩვენ მივიღეთ გასაზომი სიდიდის მნიშვნელობა მეასეის სიზუსტით. პრინციპში ჩვენ ეს სიდიდე პირდაპირ შეგვიძლია გავიყვანოთ დისპლეიზე და ავსახოთ, მაგრამ პრაქტიკაში ხშირად ხდება ხოლმე, რომ ასათი სიზუსტე საჭირო არ არის და მეათედის სიზუსტეც სრულიად საკმარისია, ამისათვის მიღებული შედეგი უნდა გავყოთ 10-ზე. შედეგად ჩვენ მივიღებთ გაზომილი

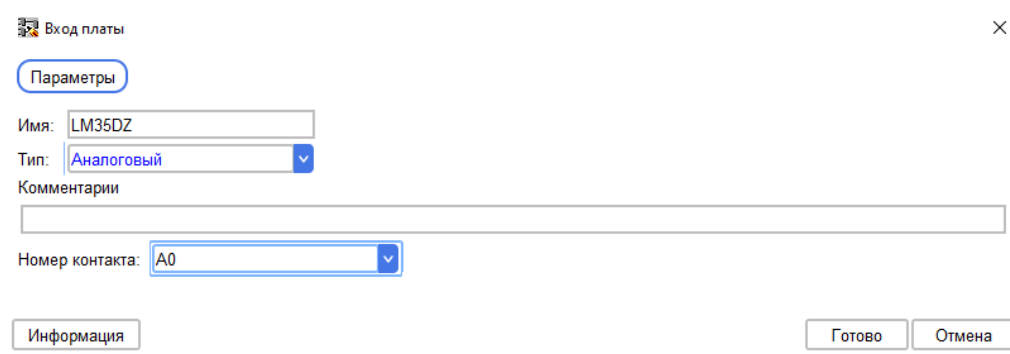
ტემპერატურის ზუსტ მნიშვნელობას ცელსიუსის გრადუსებში, რომელიც შემდეგ უნდა ასახულ უნდა იქნას დისპლეიზე.

ტემპერატურის გაზომვის ზემოთაღნიშნული ალგორითმის დაპროგრამება FLProg გარემოში უნდა განვახორციელოთ შემდეგნაირად (ფიგ. 37.1). ამისათვის საჭიროა:

- ა) გაუშვათ FLProg, დაპროგრამების ენად ავირჩიოთ FBD, კონტროლერად - Arduino UNO;
- ბ) ტეგების ზონაში გავხსნათ საქალაქდ შესასვლელი-Входы და შემდეგ დავაწკაპოთ დავმატოს შესასვლელი - Добавить вход, გამოსულ ფანჯარაში სახელის ველში - Имя ჩავწერთ ჩვენთვის მოსახერხებელი სახელი, მაგალითად - LM35DZ ჩვენს შემთხვევაში, ტიპის ველში ავირჩიოთ ანალოგური-Аналоговый და შემდეგ ახლად გამოსულ კონტაქტის ნომრის ველში-Номер контакта უნდა ჩავწეროთ ნებისმიერი კონტაქტის ნომერი A0...A5, რომელსაც პროგრამა დაგვრთავს ნებას. ჩვენს შემთხვევაში ვწერთ A0, შემდეგ ვაჭერთ თავვის მზადდოფნის ლილაკს - Готово თითს (ფიგ. 37.2), ეს ფანჯარა გაქრება და ამით არდუინოს შესასვლელის შექმნის პროცედურა ტემპერატურის გადამწოდისთვის დამთავრებული იქნება.

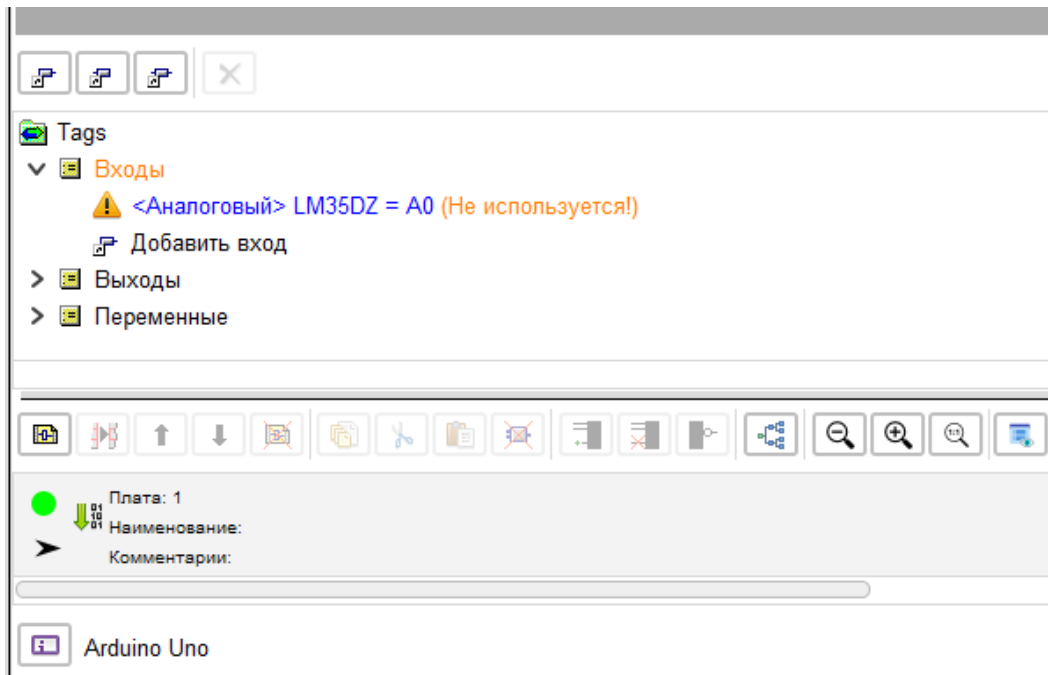


ფიგ. 37.1 ტემპერატურისა და ფოტორეზისტული გადამწოდის მართვის პროგრამა



ფიგ. 37.2 არდუინოს შესასვლელის შექმნა

ამ ფანჯარაში მზადდოფნის ლილაკის - Готово დაჭერის შემდეგ ახლად შექმნილი შესასვლელი აისახება ტეგების ზონაში (ფიგ. 37.3).



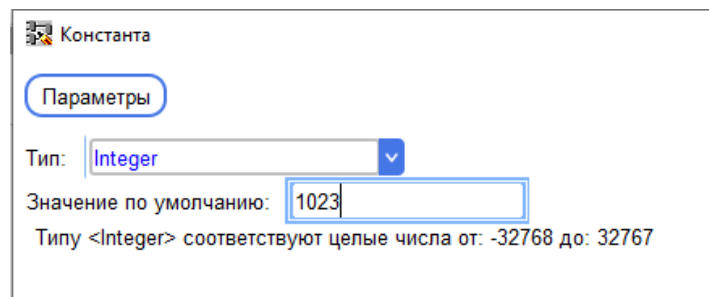
ფიგ. 37.3 ანალოგიური შესასვლელი სახელწოდებით LM35DZ

ახლა საჭიროა ეს შექმნილი შესასვლელი ბლოკი გადმოვიტანოთ სამუშაო არეში, ამიტომ თავის მარცხენა ღილაკის დაწკაპებით მოვნიშნავთ ამ შესასვლელის სახელს <Аналоговый>LM35DZ=A0 და თითის აუშვებლად გადმოვიტანთ სამუშაო ზონაში, იხილით ფიგ. 37.4. ბლოკი B1. აქვე შევნიშნავთ, რომ სწორედ ეს ბლოკი, რომელიც დაკავშირებულია არდუინოს ანალოგიურ შესასვლელთან ახორციელებს გადამწოდის მიერ გამომუშავებული ძაბვის გარდაქმნას ციფრულ ფორმატში. ზუსტად ასევე შევექმნათ მეორე ანალოგიურ შესასვლელი ფოტო- რეზისტორისთვის, დავარქვათ მას სახელს - Foto_rezist, შესასვლელ კონტაქტად ავირჩიოთ მაგალითად A1 და ჩამოვიტანოთ ეს ბლოკიც B9 პროგრამის სამუშაო ზონაში.

ამის შემდეგ, უნდა მოვახდინოთ ტემპერატურის გადამწოდის გაზომვის ალგორითმის რეალიზაცია. ამისათვის შევიდეთ პროგრამის ბლოკების ბიბლიოთეკაში-Библиотека блоков და გავხსნათ მათემატიკის საქალაღდე-Математика, სადაც მოთავსებულია ოთხი ძირითადი მათემატიკური მოქმედების ბლოკები. თავის მარცხენა ღილაკით მოვნიშნოთ გაყოფის ბლოკი [DIV(/)], თითის აუღებლად გადმოვიტანოთ პროგრამის სამუშაო ზონაში და მოვათავსოთ B1 ბლოკის მარჯვენა. დავარქვათ ამ ბლოკს B2 ბლოკი. შევაერთოთ ამ ბლოკის პირველი შესასვლელი I1 B1 ბლოკის გამოსასვლელს. ამ მიზნით მივიყვანოთ თავის მაჩვენებელი ისარი B2 ბლოკის შესასვლელ ხაზთან და როდესაც ეს ხაზი შეიცვლის ფერს და გახდება წითელი, მაშინ დავაჭიროთ თავის მარცხენა ღილაკით და თითის აუშვებლად

გადავიტანოთ ისარი B1 ბლოკის გამოსასვლელ ხაზთან. როცა ეს ხაზიც შეიცვლის ფერს და გახდება წითელი აუშვათ თითი ღილაკს და დავინახავთ, რომ B1 ბლოკის გამოსასვლელი და B2 ბლოკის შესასვლელი შეერთებული იქნებიან ერთმანეთთან შავი ხაზით.

ამით ჩვენ მოვახდინეთ გაყოფის ბლოკის გასაყოფის შესასვლელზე I1 გადამწოდის გამოსასვლელის მიერთება B1 ბლოკის გავლით. ეს რიცხვი უნდა გაიყოს 1023, ამიტომ მეორე შესასვლელზე უნდა დავსვათ კონსტანტა 1023. ამ მიზნით ვაწკაპებთ ამ მეორე შესასვლელზე I2, თავის მარჯვენა ღილაკით და გამოსულ კონსტანტის ჩასმის წაეწერაზე - Вставить константу ვაწკაპებთ ახლა უკვე მარცხენა ღილაკით, გამოსულ ფანჯარაში რიცხვის ტიპად ვირჩევთ მთელ რიცხვს Integer და შედეგად გამოსულ რიცხვის ველში ვწერთ 1023, ბოლოს კი ვაწკაპებთ Готово -ს (ფიგ. 37.4).



ფიგ. 3.4 გაყოფის ბლოკის დაპროგრამება

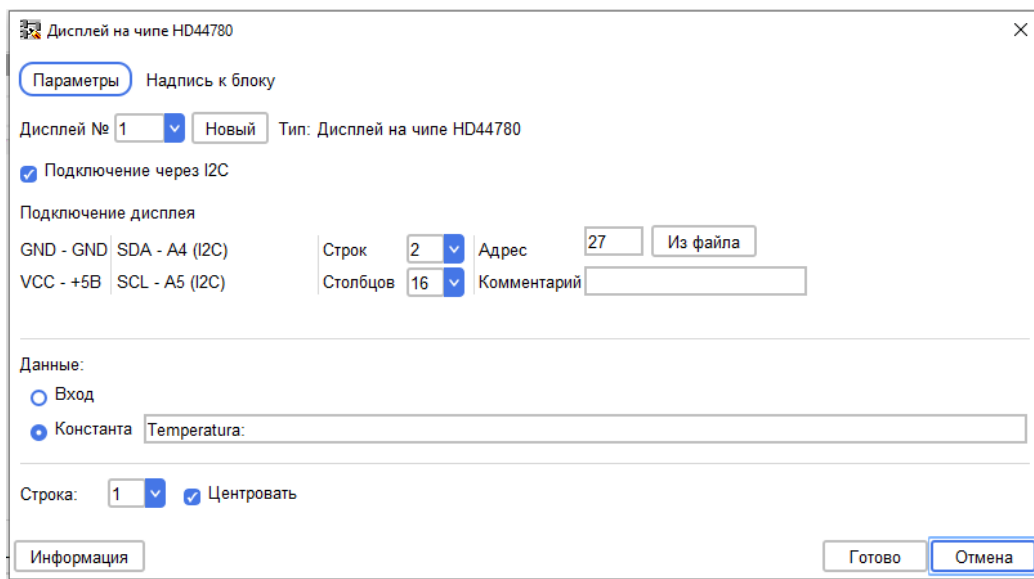
ამრიგად, ჩვენი ალგორითმის პირველი ნაწილის დაპროგრამება უკვე მოვახდინეთ B1 ბლოკის გამოყენებით და ტემპერატურის გადამწოდის მიერ გაზომილი ტემპერატურის სიდიდე, რომელიც გარდაქმნილია ათობით რიცხვად გავყავით მუდმივ რიცხვზე-1023. ამის შემდეგ ვახდენთ B3, B4, B5 ბლოკების გადმოტანას სამუშაო ზონაში და ანალოგიურად ზემოთაღწერილისა ვახდენთ მიღებული შედეგის გამრავლებას ჯერ ხუთზე, შემდეგ 1000-ზე და შემდეგ მიღებული შედეგის გაყოფას 10-ზე, რის შედეგადაც B5 ბლოკის გამოსასვლელზე მივიღებთ ტემპერატურის მნიშვნელობას ათობითი რიცხვის ფორმატში, ანუ დაკალიბრებულს ცელსიუსის გრადუსებში.

ამის შემდეგ ჩვენ გვჭირდება B5 ბლოკის გამოსასვლელზე მიღებული ტემპერატურის რიცხვითი მნიშვნელობა ავსახოთ ორსტრიქონიან 16 თანრიგიან ინდიკატორზე. მაგრამ ეს უკანასკნელი, როგორც უკვე ცნობილია, რეაგირებს მარტო ფრიცხვის სტრიქონულ ფორმატზე. ამიტომ შევდინართ ბლოკების ბიბლიოთეკის ტიპების გარდაქმნის საქალაქში - Преобразование типов და ვირჩევთ სტრიქონების კონვერტაციის ბლოკს-Конвертация строк და გადმოგვაქვს სამუშაო ზონაში B6, აღსანიშნავია, რომ ამ ბლოკს არ ჭირდება პარამეტრიზაცია,

იგი უბრალოდ მთელ რიცხვს გარდაქმნის სტრიქონულ ფორმატში, რომელიც მოსახერხებელია ასხვისათვის დისპლეიზე.

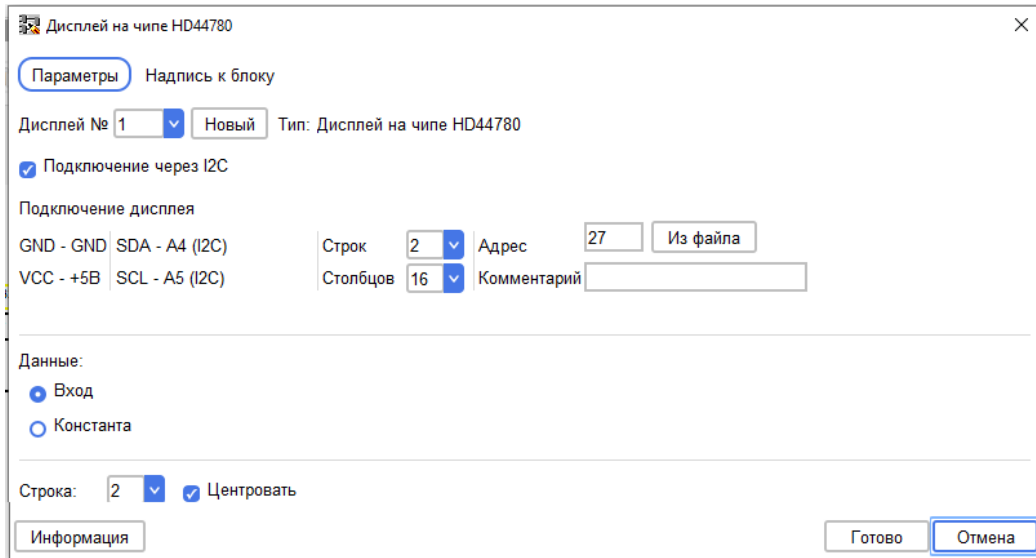
ახლა უმჯობესია ჯერ გადმოვიტანოთ შესაბამისი დისპლეი სამუშაო ზონაში. ამ მიზნით შევდივართ პროგრამის ბლოკების ბიბლიოთეკის დისპლეების საქაღალდეში - Дисплеи სადაც ვირჩევთ Дисплеи на чипе HD44780 და გადმოგვაქვს პროგრამის სამუშაო ზონაში.

მოვახდინოთ ამ ბლოკის პარამეტრიზაცია. მაგრამ რადგან იგი შეიცავს ორ სტრიქონს, ამიტომ ეს ბლოკი უნდა წარმოვადგინოთ ორი ქვებლოკის სახით B8_1 და B8_2. დავაწკაპუნოთ ამ ბლოკზე თავის მარცხენა ღილაკით ორჯერ და გამოსულ ფანჯარაში (ფიგ. 37.5) შევქმნათ ახალი დისპლეი-Новый ღილაკზე დაჭერით. დისპლეის ნომრის ველში-Дисплеи № ავტომატურად ჩაიწერება 1, რადგან იგი პირველი დისპლეია ჩვენს პროექტში (და უკანასკნელიც). სტრიქონების ველში - Строк ჩავწეროთ 2 რადგან ჩვენ დაგვჭირდება ორივე სწტრიქონის გამოყენება, სვეტების ველში Сталцов ჩავწეროთ 16, რადგან ვიყენებთ თექვსმეტივე პოზიციას იმფორმაციის ასახვისათვის. მისამართების ველში-Адрес ჩავწეროთ ამ კონკრეტული დისპლეისთვის ქარხნული წესით მინიჭებულ კოდი (ჩვენ შემთხვევაში 27), მონაცემების ველში-Данные ავირჩიოთ კონსტანტა-Константа და მის შესაბამის ველში ჩავწეროთ Temperatura., სტრიქონის ველში Строка ჩავწეროთ 1 (პირველი სტრიქონი), მოვნიშნოთ ალაში განლაგდეს ცენტრში - Центровать და დავაჭიროთ მზადყოფნის ღილაკს Готово. ამით დისპლეის პირველი სტრიქონი დაპროგრამებული იქნება, რის შემდეგაც უნდა დავაკოპიროთ ეს ბლოკი და შევქმნად მისი ასლი. პირველ ბლოკს დავაწეროთ B8_1, ხოლო მეორეს B8_2 და სწორედ ამ მეორე სტრიქონში უნდა ავსახოთ გასაზომი ტემპერატურის მნიშვნელობა.



ფიგ. 37.5 დისპლეის ბლოკის პირველი სტრიქონის დაპროგრამება

ამისთვის საჭირო იქნება ამ მეორე სტრიქონის დაპროგრამებაც (ფიგ. 37.6). აქ უკვე მონაცემებში - Данные ჩავრთავთ შესასვლელს-Вход, სტრიქონის ველში- Строка ჩავწერთ 2, ხოლო დანარჩენ ველებს იგივეს დავტოვებთ. ვაჭერთ ხელს მზადყოფნის ღილაკს და ვუბრუნდებით ისევ ფიგ. 37.1-ს.

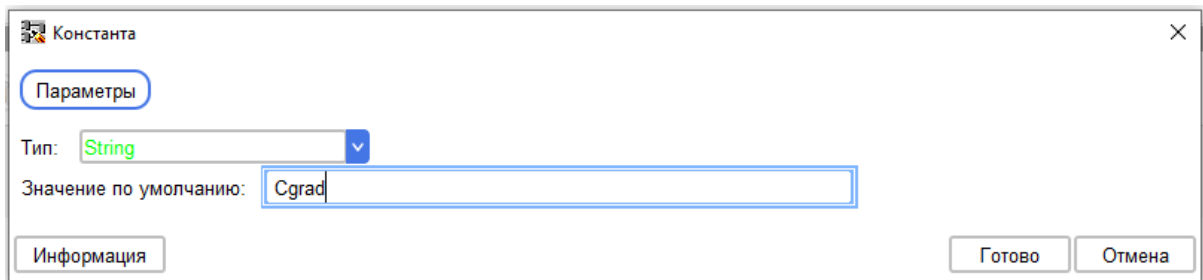


ფიგ. 37.6 დისპლეის ბლოკის მეორე სტრიქონის დაპროგრამება

ტემპერატურის გასაზომი სიდიდე ჩვენ მივიღეთ კონვერტაციის ბლოკის B6-ის გამოსასვლელზე. თუ პირდაპირ შეუერთებთ ამ ბლოკის გამოსასვლელს დისპლეის შესასვლელს, მაშინ დისპლეიზე აისახება ამ ტემპერატურის რიცხვითი მნიშვნელობა და მეტი არაფერი. უმჯობესი კი იქნებოდა ამ სიდიდის შემდეგ ასახულიყო მისი განზომილება. ამ მიზნით უნდა შემოვიტანოთ კიდევ ერთი ბლოკი. FLProg-ის ბლოკების ბიბლიოთეკის სტრიქონების-Строки საქალაქდებში, მოვნიშნოთ სტრიქონების შეკრების ბლოკი-Сложение строк გადმოვიტანოთ იგი პროგრამის სამუშაო ზონაში B7 და დავაყენოთ კონვერტაციის ბლოკის B6-ის შემდეგ. შევაერთოთ B6-ის გამოსასვლელი B7-ის პირველ შესასვლელთან. ხოლო მეორე შესასვლელი უნდა დავაპროგრამიროთ. ამ მიზნით უნდა მოვნიშნოთ ეს შესასვლელი თავის მარჯვენა ღილაკით, დავაწკაპოთ კონსტანტის ჩასმის ფანჯარაზე- Вставить константу და გამოსულ ფანჯარაში ჩავწეროთ - „პრაბელი Cgrad.“. (ფიგ. 37.7). დავაჭიროთ მზადყოფნის ღილაკს-Готово და ამით ტემპერატურის გადამწოდის მართვის დაპროგრამება დამთავრებული იქნება.

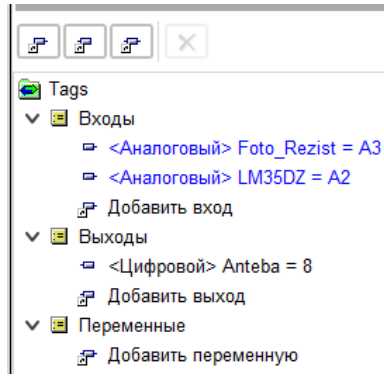
პროგრამის არდუინოში გადაწერისა და გაშვების შემდეგ დისპლეის პირველ სტრიქონში აისახება Temperatura:, ხოლო მეორე სტრიქონში ამ მომენტისათვის იმ ადგილას არსებული ტემპერატურის მნიშვნელობა, სადაც მოთავსებულია გადამწოდი, ჩვენს

შემთხვევაში - 22 Cgrad. აქ, მაგალითისათვის აღებულია ოთახის ნომინალური ტემპერატურა. გადამწოდის კორპუსზედ თითებით შეხების შემთხვევაში ასახული ტემპერატურა უნდა გაიზარდოს ადამიანის მიმდინარე ტემპერატურამდე.

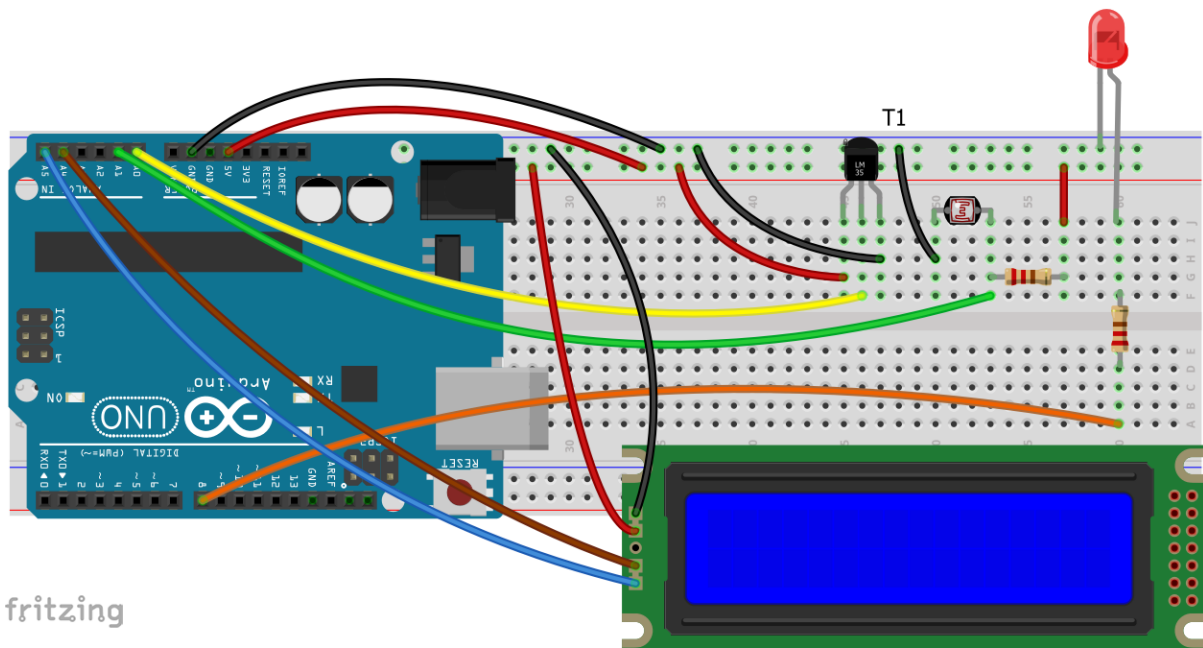


ფიგ. 37.7 სტრიქონების შეკრების ბლოკის დაპროგრამება

ახლა საჭიროა დავაპროგრამოთ ფოტო-რეზისტორული გადამწოდის მართვის სქემა. როგორც ზემოთ იყო აღნიშნული, ამ გადამწოდის გამოსასვლელი შეერთებულია არდუინოს ანალოგურ შესასვლელთან A1. როგორც არდუინოს სპეციფიკაციიდან ვიცით ეს შესასვლელი ახორციელებს მასზედ მოდებული ძაბვის (0...5ვ) გარდაქმნას ციფრულ ფორმატში (0...1023). ექსპერიმენტით დადგენილია, რომ ცუდი განათებულობის შემთხვევაში B9-ის გამოსასვლელზე 250 ერთეულზე ნაკლები სიდიდეა, ხოლო კარგი განათებულობის შემთხვევაში - 250 ერთეულზე მეტი. ამ სიდიდეების გამოჯენისათვის ვიყენებთ შედარების ბლოკს B10, რომელიც გადმოტანილი გვაქვს FLProg-ის შედარების საქალაქედან-Сравнение, ბლოკი Comparator. დავაწკაპუნოთ ამ ბლოკში და შედარების სახეობაში-Вид сравнения ავირჩიოთ ნაკლებია ან ტოლია ნიშანი, მის პირველ შესასვლელს მივუერთოთ B9 ბლოკის გამოსასვლელი, ხოლო მეორე შესასვლელზე კი კონტექსტური მენიუდან უნდა მოვნიშნოთ კონსტანტა და კონსტანტის ველში ჩავწეროთ 250. ამ ბლოკის გამოსასვლელი კი არდუინოს გამოსასვლელი კონტაქტის გავლით შევაერთოთ შუქდიოდის ანოდს 220 ომი შემზლუდავი წინააღმდეგობის გავლით. კარგი განათებულობის შემთხვევაში კომპარატორის ბლოკის გამოსასვლელზე იქნება ნულოვანი პოტენციალი და შუქდიოდი არ ანათებს, თუ ფოტო-რეზისტორს ხელს დავაფარებთ და შევუქმნით სიბნელის გარემოს, მაშინ კომპარატორის ბლოკის გამოსასვლელზე დაჯდება ერთიანი და გაანათებს შუქდიოდს. პროექტის ტეგების ზონის ფანჯრა ასახულია ფიგ. 37.8-ზე, ხოლო პროექტის Fritzing სქემა-ფიგ. 37.9-ზე.



ფიგ. 37.8 პროექტის ტეგების ზონის ფანჯარა



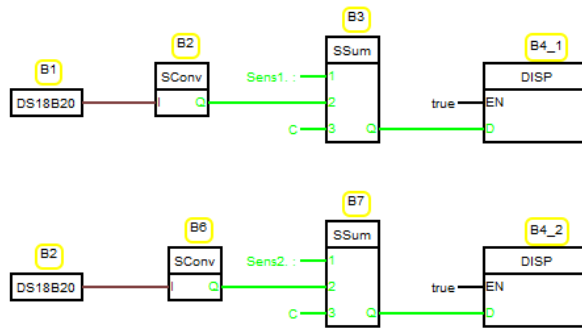
ფიგ. 37.8 ტემპერატურის გადაწოდისა და ფოტო-რეზისტორის შეერთებების Fritzing სქემა

პროექტი_38 ტემპერატურის ციფრული გადამწოდი DS18B20



ტემპერატურის ციფრული გადამწოდი DS18B20 - ეს სრულფასოვანი ციფრული თერმომეტრია, რომელსაც შეუძლია ტემპერატურის გაზომვა დიაპაზონში მინუს 55 გრადუსიდან პლიუს 125 გრადუსამდე 9-12 ბიტის პროგრამირებადი სიზუსტით. წარმოებაში დამზადების პროცესში თვითოეულ გადამწოდს მიაკუთვნებული აქვს თავისი უნიკალური 64 თანრიგიან მისამართი და ინფორმაციის გაცვლა გარეშე წამყვან მოწყობილობებთან (არდუინო) ხორციელდება 1-Wire სალტის დახმარებით. ასეთი მიდგომა საშუალებას გვაძლევს მივუერთოთ ერთ ხაზს გადამწოდების მთელი ჯგუფი, რაოდენობით თვით 2 ხარისხად 64-ისა. ეს ელექტრონული ხელსაწყო განეკუთვნება ინტელექტუალური გადამწოდების კლასს, შეიცავს თავის შემადგენლობაში მცირე ზომის მიკროსქემას, რომელიც შეიცავს თავის მხრივ ელექტრონული ბლოკებისა და მოდულების მთელ ნაკრებს. უნიკალურია ამ მიკროსქემის კვების რეჟიმი. იგი შეიძლება იკვებოს პირდაპირი კვების წყაროთი (მაგვა მიეწოდება გამომყვანებზე Vdd და GND), ანდა პარაზიტული კვების წყაროთი, ამ შემთხვევაში იგი იკვებება მონაცემთა ხაზიდან. ჩვენ ამ პროექტში გამოვიყენებთ გადამეწოდის პირდაპირი კვების რეჟიმს.

ფიგ. 38.1-ზე წარმოდგენილია ამ პროექტში განხორციელებული ორი DS18B20 გადამწოდის მართვის პროგრამა. როგორც ვხედავთ, ორივე გადამწოდის მართვა აბსოლუტურად იდენტურია, ოღონდ განსხვავდებიან ისინი ამ გადამწოდების საქარხნოდ მიკუთვნებული მისამართებით-Адрес. ჩვენს ხელთ არსებული გადამწოდებიდან პირველი ბლოკის-B1-ს ეს მისამართი არის 28 FF B4 7B 90 15 01 41, ხოლო მეორე ბლოკის- B2-ს არის 28 FF A3 5C 90 15 01 9A. ფიგ. 38.2 და ფიგ. 38.3 წარმოდგენილი არის ამ ორივე გადამწოდის პარამეტრიზაციის ფანჯრები. ამ ფრანჯრებიდან ჩანს, რომ პირველი გადამწოდის სასიგნალო გამომყვანი მიერთებულია არდუინოს მე-2 კონტაქტთან, ხოლო მეორესი-მე-3-სთან.



ფიგ. 38.1 ორი DS18B20 გადამწოდის მართვის პროგრამა

Датчик температуры серии DS18x2x

Параметры Надпись к блоку

Шина OneWire Шина OneWire на 2 Новая

Ручное задание адреса Адрес: 28 FF B4 7B 90 15 01 41 Загрузить из файла

Тип: DS18B20

Массив

Комментарий:

Паразитное питание

Вход разрешения опроса

Время опроса датчика

В каждом цикле (не рекомендуется)

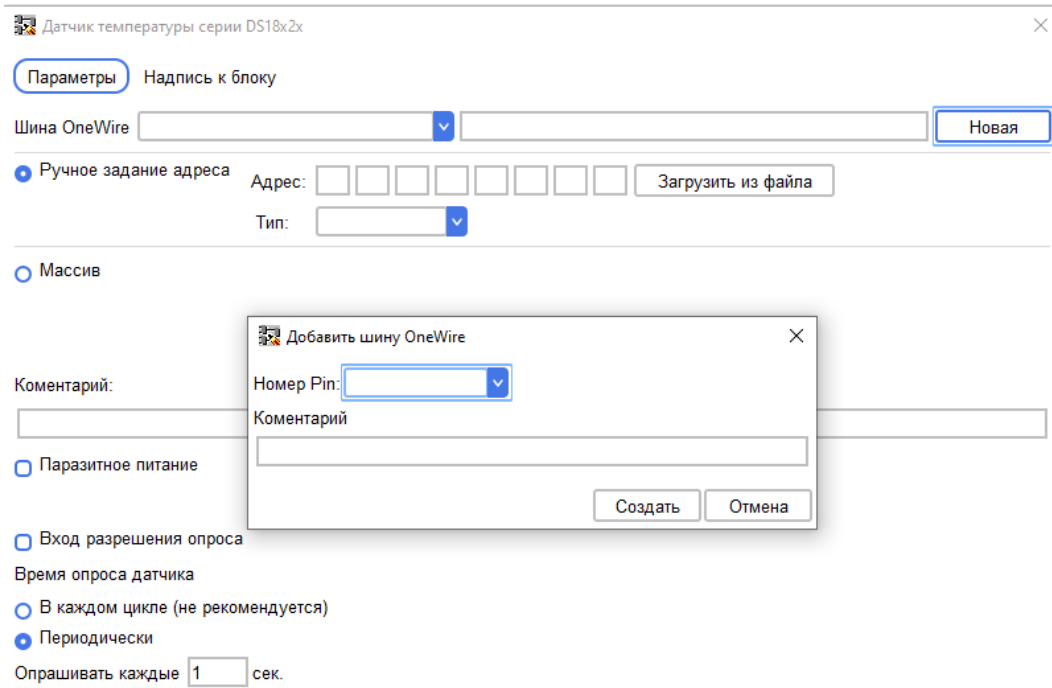
Периодически

Опрашивать каждые 1 сек.

ფიგ. 38.2 პირველი გადამწოდის პარამეტრიზაციის ფანჯარა

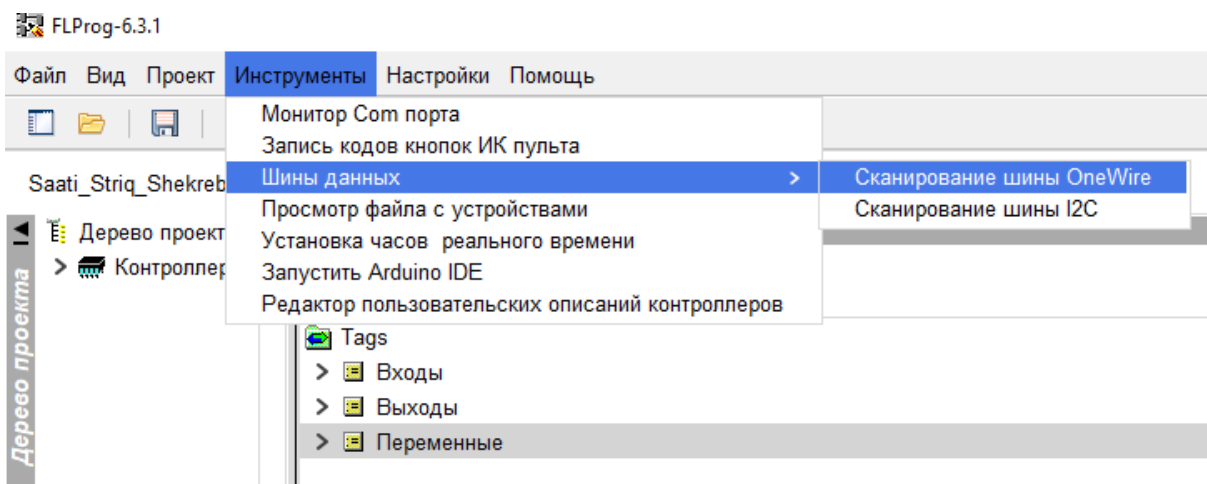
ფიგ. 38.3 მეორე გადამწოდის პარამეტრიზაციის ფანჯარა

კონტაქტების არჩევა ხდება პარამეტრიზაციის ფანჯრიდან. როდესაც გადამწოდის ბლოკს DS18x2x, ბლოკების ბიბლიოთეკის გადამწოდების საქალაქედან-Датчики გადმოვიტანთ FLProg-ის სამუშაო ზონაში, დავაწკაპუნებთ ამ ბლოკზე და გამოვაჩინთ მის პარამეტრიზაციის ფანჯარას ფიგ. 38.2 შემდეგ უნდა დავაწკაპოთ ახალი-Новая (ფიგ. 38.4) ეკრანზე გამოვა უფრო მცირე ზომის ფანჯარა სადაც ჩამოშლადი სიიდან Home Pin უნდა ამორჩეულ იქნას არდუინოს იმ კონტაქტის ნომერი, რომელზედაც უნდა მიუერთდეს გადამწოდის სასიგნალო გამომყვანი. ჩვენს შემთხვევაში, პირველი გადამწოდისთვის არჩეული გვაქვს მე-2 კონტაქტი, ხოლო მეორე გადამწოდისთვის მე-3 კონტაქტი (იხ ფიგ. 38.2 და ფიგ. 38.3). აქვე გადამწოდის ტიპის ველში-Тип უნდა ჩაიწეროს ჩვენს ხელთ არსებული გადამწოდის ტიპი DS18B20.

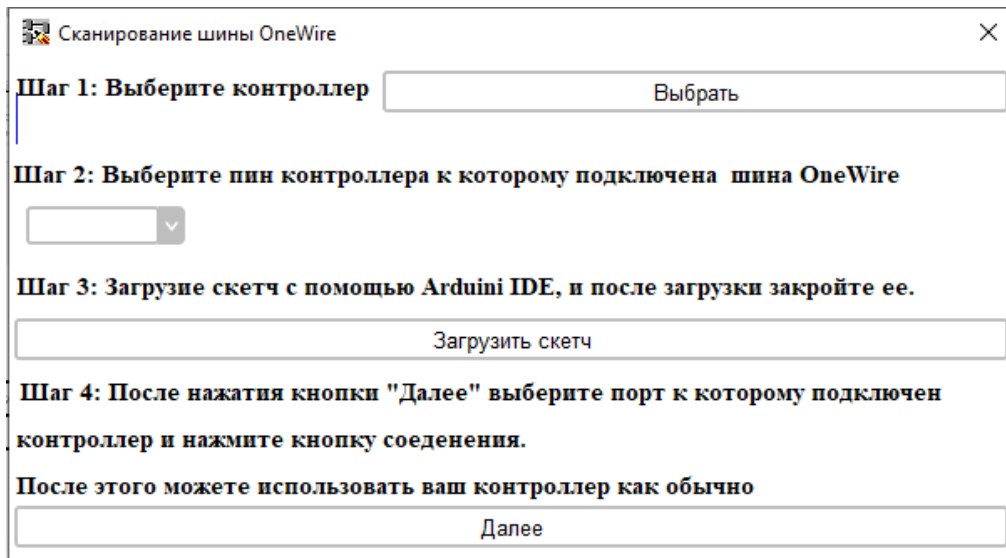


ფიგ. 38.4 OneWire სალტის არდუინოსთან მიერთების კონტაქტის არჩევა

ასევე მარტივია თვითოეული გადამწოდისათვის თავისი საქარხნო მისამართის დადგენა. ამისათვის უნდა დსავაწკაპუნოთ FLProg-ის ინსტრუმენტების-Инструменты მენიუს მონაცემთა სალტის-Шины данных → Сканирование шины OneWire სალტის სკანირებაზე (იხ. ფიგ. 38.5) და ახლად გამოსულ OneWire სალტის სკანირების ფანჯარაში (ფიგ. 38.6) შევასრულოთ რამდენიმე ნაბიჯი: 1) ავირჩიოთ კონტროლერი-Выбрать ველზე დაწკაპებით ჩამოშლილი სიიდან;



ფიგ. 38.5 DS18B20 ტრემპერატურის გადამწოდის მისამართის დადგენა

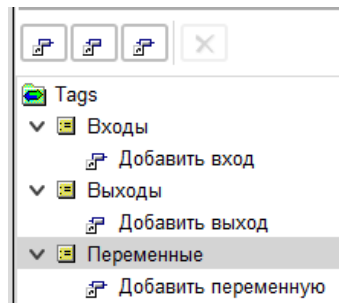


ფიგ. 38.6 DS18B20 ტემპერატურის გადაწოდის მისამართის დადგენა

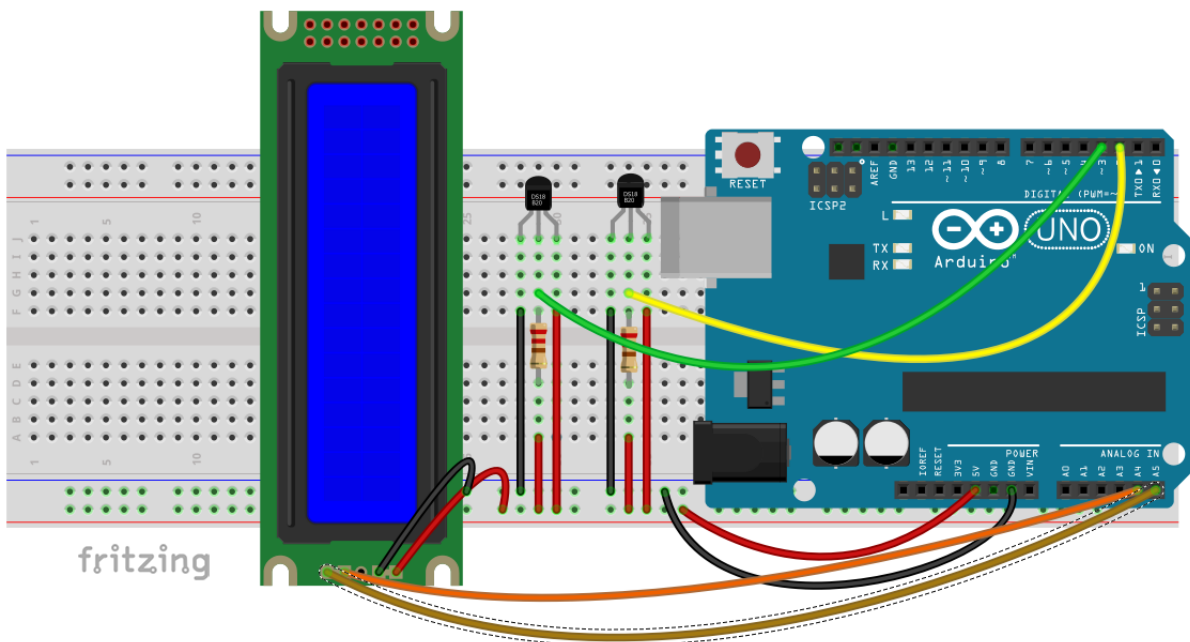
2) ავირჩიოთ კონტროლერის იმ კონტაქტის ნომერი, რომელზედაც მიერთებულია OneWire სალტე, ანუ DS18B20 გადაწოდის სასიგნალო გამომყვანი (მე-2 პინი B1 გადაწოდის შემთხვევაში; 3) ჩავტვირთოთ მიღებული სკეჩი არდუინოში-Загрузить скетч ველზე დაჭერით 4) შემდეგ დავაჭიროთ ველზე Далее, გამოვა ახალი ფანჯარა, სადაც უნდა მივუთითოთ პორტის ნომერი, რომელზედაც მიერთებულია არდუინო და უნდა დავაჭიროთ შეერთების ღილაკს თავგის ღილაკით. ეკრანზე გამოვა საძებნი გადაწოდის მისამართის კოდი. უნდა დავაფიქსიროთ ეს კოდი სადმე ფაილში და შემდეგ უნდა გადავწეროთ ფიგ. 38.4-ში მისამართის ველში-Адрес:. ამის შემდეგ უნდა გავთიშოთ შეერთება გათიშვის ღილაკზე დაჭერით და შემდეგ არდუინო შეგვიძლია გამოვიყენოთ დანიშნულების მიხედვით.

დავუბრუნდეთ ისევ ჩვენს პროგრამას, რომელიც ასახულია ფიგ. 38.1-ზე. როგორც პროგრამიდან ჩანს, გადაწოდის B1 და B2 გამოსასვლელები სტრიქონების კონვერტაციის ბლოკების გავლით B2 და B6 შეერთებულნი არიან სამშესასვლელიანი სტრიქონების შეკრების ბლოკების B3 და B7 მე-2 შესასვლელებზე. მათ პირველ შესასვლელებზე კი ჩაწეილია კონსტანტები: B3 ბლოკისთვის Sens1 : პრაბელი და B4 ბლოკისთვის Sens2 : პრაბელი. მე-3 შესასვლელებზე კი ჩაწეილია პრაბელი და ტემპერატურის აღმნიშვნელ ასო C. პროგრამის კომპილიაციისა და არდუინოში ჩატვირთვის შემდეგ დისპლეის პირველ და მეორე სტრიქონებში გამოვლენ ტემპერატურის სიდიდეები გაზომილნი შესაბამისად პირველი და მეორე გადაწოდების მიერ, რომლებიც შევსებულნი იქნებიან ზემოთაღნიშნულ კონსტანტებში მოყვანილი ჩანაწერებით.

აქ უნდა აღინიშნოს დამატებით, რომ გარდა აღნიშნული ელექტრონული კომპონენტებისა, გადამწოდების კორექტული მუშაობისათვის საჭიროა კიდევ 4.7 კომ-ის რეზისტორები, რომლებიც შეერთებულია ერთის მხრივ ძაბვასთან +5 ვ, ხოლო მეორეს მხრივ გადამწოდის სასიგნალო გამოყვანებთან. პროექტს ტეგების ზონაში ჩანანაწერები არა აქვს (ფიგ. 38.7), ხოლო მისი Fritzing სქემა მოყვანილია ფიგ. 38.8-ზე.

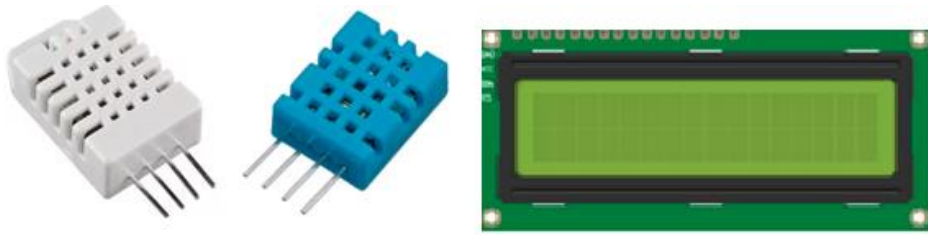


ფიგ. 38.7 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 38.8 პროექტის Fritzing სქემა

პროექტი_39 ტენიანობის გადამწოდები DHT11 და DHT12



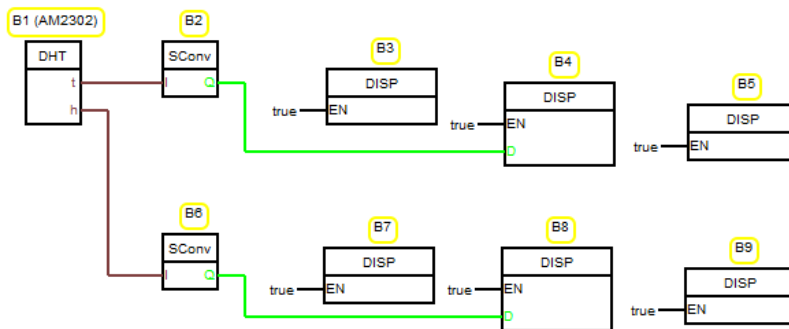
DHT11 და DHT12 გადამწოდები ძალედ პოპულარულები არიან არდუინოს გარემოში და ხშირად გამოიყენებიან მეტეოსადგურის ანდა ჭკვიანი სახლების პროექტებში. ისინი არ გამოირჩევიან მაღალი სწრაფმოქმედებითა და სიზუსტით, მაგრამ ძალზედ მარტივნი არიან თავისი შესრულებებით. ეს გადამწოდები შეიძლება თამამად იქნას გამოყენებული თქვენს პირველ პროექტებში, რადგან გამოირჩევიან დაბალი ფასით და ხელმისაწვდომობით.

გადამწოდი შედგება ორი ნაწილისაგან - ტევადური ტემპერატურის გადამწოდისგან და გიგრომეტრისგან. პირველი გამოიყენება ტემპერატურის გაზომვისათვის და მეორე-ჰაერის ტენიანობის გაზომვისათვის. გადამწოდის შიგნით მყოფმა ჩიპმა შეიძლება შეასრულოს ანალოგურ-ციფრული გარდაქმნა და გამოსცეს ციფრული სიგნალი რომელიც შემდეგ შეიძლება წაკითხულ იქნას არდუინოს კონტროლერის მიერ. DHT11 გადამწოდი გამოიყენება ტემპერატურის გაზომვისათვის დიაპაზონში 0...50 გრადუსი (სიზუსტე 2%) და ტენიანობის გაზომვისათვის დიაპაზონში 20%...80% (ცდომილება 5%-მდე), მისი კვება ხდება ძაბვით 3...5 ვოლტის დიაპაზონში. DHT22 ტიპის გადამწოდს აქვს რამდენადმე უკეთესი მახასიათებლები: ა) ტენიანობა 0%...100%, გაზომვის სიზუსტე 2%...5% ბ) ტემპერატურა - მინუს 40 გრადუსიდან ... პლიუს 125 გრადუსამდე, სიზუსტე-0,5%.

FLProg პროგრამის ბლოკების ბიბლიოთეკის გადამწოდების საქალაქის-Датчики ტენიანობის გადამწოდებიდან-Датчики Влажности პროგრამის სამუშაო ზონაში გადმოვიტანოთ ტემპერატურისა და ტენიანობის გადამწოდი-Датчик температуры и влажности DHT11 - B1 (ფიგ. 39.1). ამ ბლოკს აქვს ორი გამომყვანი: t და h. ორივე ეს გამომყვანი კონვერტაციის ბლოკების B2 და B6 გავლით მიერთებულია HD44780 დისპლეის პირველი და მეორე სტრიქონების შესასვლელებთან შემდეგნაირად: დისპლეის ორივე სტრიქონი დაყოფილია სამსამ ნაწილად B3, B4, B5 პირველი სტრიქონისთვის და B7, B8, B9 მე-2 სტრიქონისთვის.

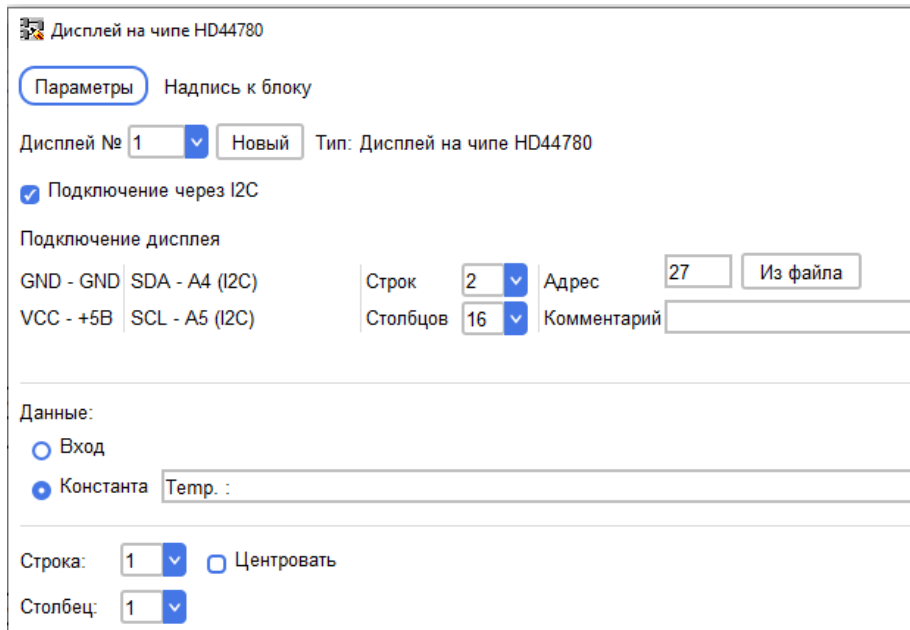
დისპლეის პირველი სტრიქონის პირველი ნაწილი B3 იწყება პირველი სვეტიდან და მასში ჩაწერილია კონსტანტის სახით Temp : (იხ. ფიგ. 39.2). დისპლეის მეორე ნაწილი იწყება

მე-9 სვეტიდან (ფიგ. 39.3) და მასზედ მიერთებულია გადამწოდის t-ს გამოსასვლელი B2 ბლოკის გავლით. დისპლეის მესამე ნაწილი იწყება მე-15 სვეტიდან (ფიგ. 39.4) და მასზედ ჩაწერილია კონსტანტის სახით ტემპერატურის განზომილება C.



ფიგ. 39.1 DHT11 ტენიანობის გადამწოდის მართვის პროგრამა

ზუსტად ასევეა პარამეტრიზებული დისპლეის მეორე სტრიქონიც. მისი პირველი ნაწილი იწყება პირველი სვეტიდან და კონსტანტის სახით ჩაწერილია Humid : (ფიგ. 39.5), მე-2 ნაწილი იწყება მე-9 სვეტიდან და მასზედ მიერთებულია გადამწოდის h გამოსასვლელი B6 ბლოკის გავლით (ფიგ.39.6), ხოლო დისპლეის მეორე სტრიქონის მე-3 მწილი იწყება მე-15 სტრიქონიდან და მასში ჩაწერილია ტენიანობის განზომილება % (ფიგ. 39.7).



ფიგ. 39.2 დისპლეის პირველი სტრიქონის პირველი ნაწილის პარამეტრიზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND SDA - A4 (I2C) Строк Адрес

VCC - +5B SCL - A5 (I2C) Столбцов Комментарий

Данные:

Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 39.3 დისპლეის პირველი სტრიქონის მეორე ნაწილის პარამეტრიზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Новый Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND SDA - A4 (I2C) Строк Адрес

VCC - +5B SCL - A5 (I2C) Столбцов Комментарий

Данные:

Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 39.4 დისპლეის პირველი სტრიქონის მესამე ნაწილის პარამეტრიზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/>	Комментарий	<input type="text"/>	

Данные:

Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 39.5 დისპლეის მეორე სტრიქონის პირველი ნაწილის პარამეტრიზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/>	Комментарий	<input type="text"/>	

Данные:

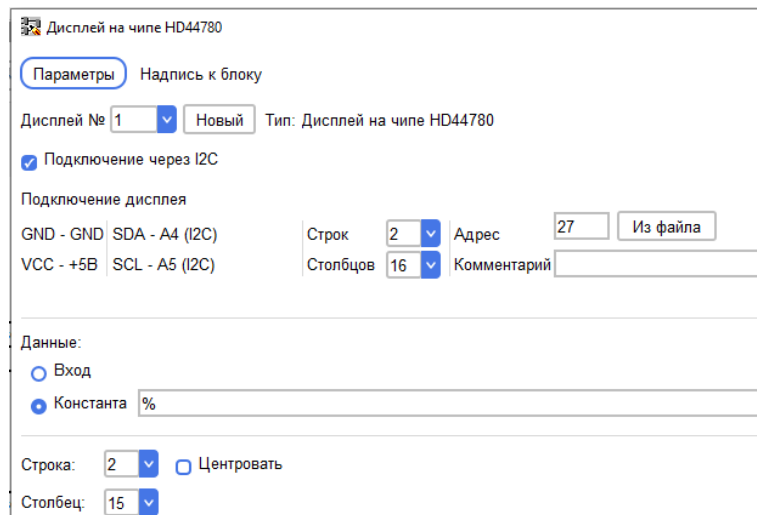
Вход

Константа

Строка: Центровать

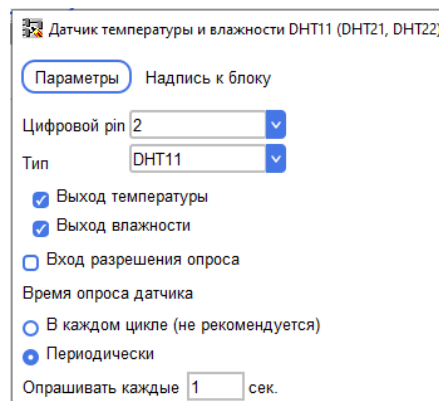
Столбец:

ფიგ. 39.6 დისპლეის მეორე სტრიქონის მეორე ნაწილის პარამეტრიზაცია



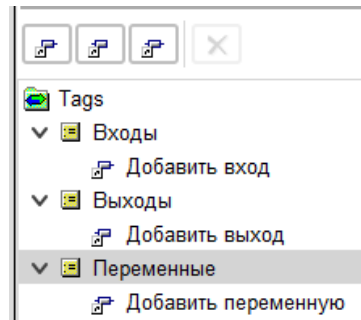
ფიგ. 39.7 დისპლეის მეორე სტრიქონის მესამე ნაწილის პარამეტრიზაცია

უნდა აღინიშნოს, რომ DHT11 (DHT12) ტიპის გადაწოდს აქვს ოთხი გამომყვანი, მათგან ორი გამომყვანი საჭიროა გადამცემის კვებისათვის (GND, +5 ვ) (პირველი და მე-4 გამომყვანები), ერთი გამომყვანი არის სასიგნალო (მეორე), რომელიც უერთდება არდუინოს ერთერთ ციფრულ გამომყვანს, რომელიც მიერთებულია აგრეთვე +5 ვოლტთან 4,7 კილოომიანი რეზისტორის გავლით, ხოლო მესამე გამომტვანი საერთოდ არ გამოიყენება საერთოდ არაფერში და დაკიდებულია ჰაერში. გადაწოდი საჭიროებს პარამეტრიზაციას, რომლის ფანჯარაშიც შევდივართ ტრადიციულად-ბლოკზე დაწკაპუნებით. პარამეტრიზაციის ფანჯარაში ციფრული კონტაქტის ველში-Цифровой pin უნდა ჩაიწეროს, თუ არდუინოს რომელ კონტაქტს უნდა შეუერთდეს მისი სასიგნალო გამომყვანი, ასევე მონიშნული უნდა იყოს ტემპერატურის გამოსასვლელისა-Выход температуры და ტენიანობის გამოსასვლელის-Выход влажности აღმები. ამავე ფანჯარაში შეძლება გადაწოდის ამოკითხვის პერიოდულობის განსაზღვრაც შესაბამისი ვლის-Опрашивать каждые შესვებით. ჩვენს შემთხვევაში ჩაწერილი გვაქვს 1 წამი (ფიგ. 39.8).

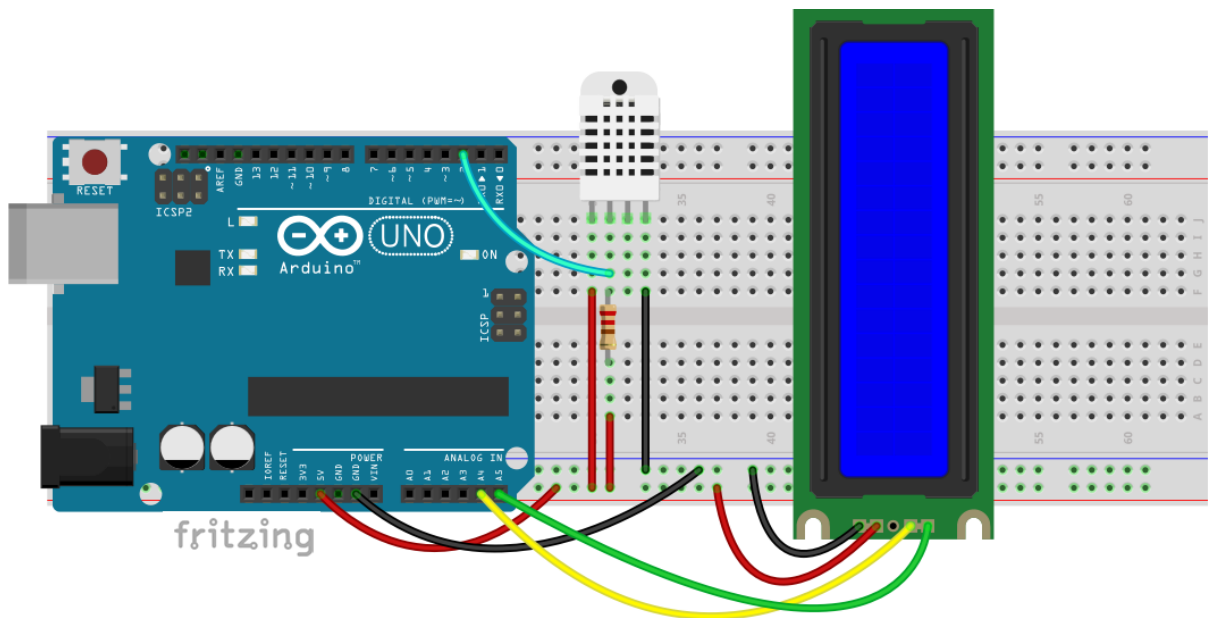


ფიგ. 39.8 DHT11 (DHT12) ტიპის გადაწოდის პარამეტრიზაციის ფანჯარა

პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 39.9-ზე, ხოლო Fritzing პროგრამა სახულია ფიგ. 39.10-ზე.



ფიგ. 39.9 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 39.10 პროექტის Fritzing პროგრამა

პროექტი_40 წნევის გადამწოდი [BMP-085]



ატმოსფერული წნევის გადამწოდის BMP-085 დანიშნულებათა ბარომეტრული წნევის კონტროლი, გარდა ამისა, იგი აკონტროლებს აგრეთვე ტემპერატურასაც. გადამწოდი გამოიყენება მრავალ პროექტში, მათ რიცხვში არდუინოს ბაზაზე შექმნილ პროექტებშიც, რადგანაც მას პრაქტიკულად არა აქვს ანალოგი მსოფლიოში. გარდა ამისა, მისი ღირებულება მცირეა რაც კიდევ უფრო ზრდის მისი გამოყენების მიმზიდველობას პროექტებში. მაგრამ, აქ შეიძლება დაისვას შეკითხვა-რაში შეიძლება დაჭირდეს ვინმეს ატმოსფერული წნევის გაზომვა? პასუხი მდგომარეობს იმაში, რომ აკონტროლოს სიმაღლე ზღვის დონიდან. ცნობილია, რომ სიმაღლის ზრდით ატმოსფერული წნევა მცირდება, ამიტომ იგი მეტად მოსახერხებელია ლაშქრობებში როგორც GPS ნავიგატორების ალტერნატივა. გარდა ამისა, ატმოსფერული წნევის მაჩვენებლით შესაძლებელია ამინდის პროგნოზირება.

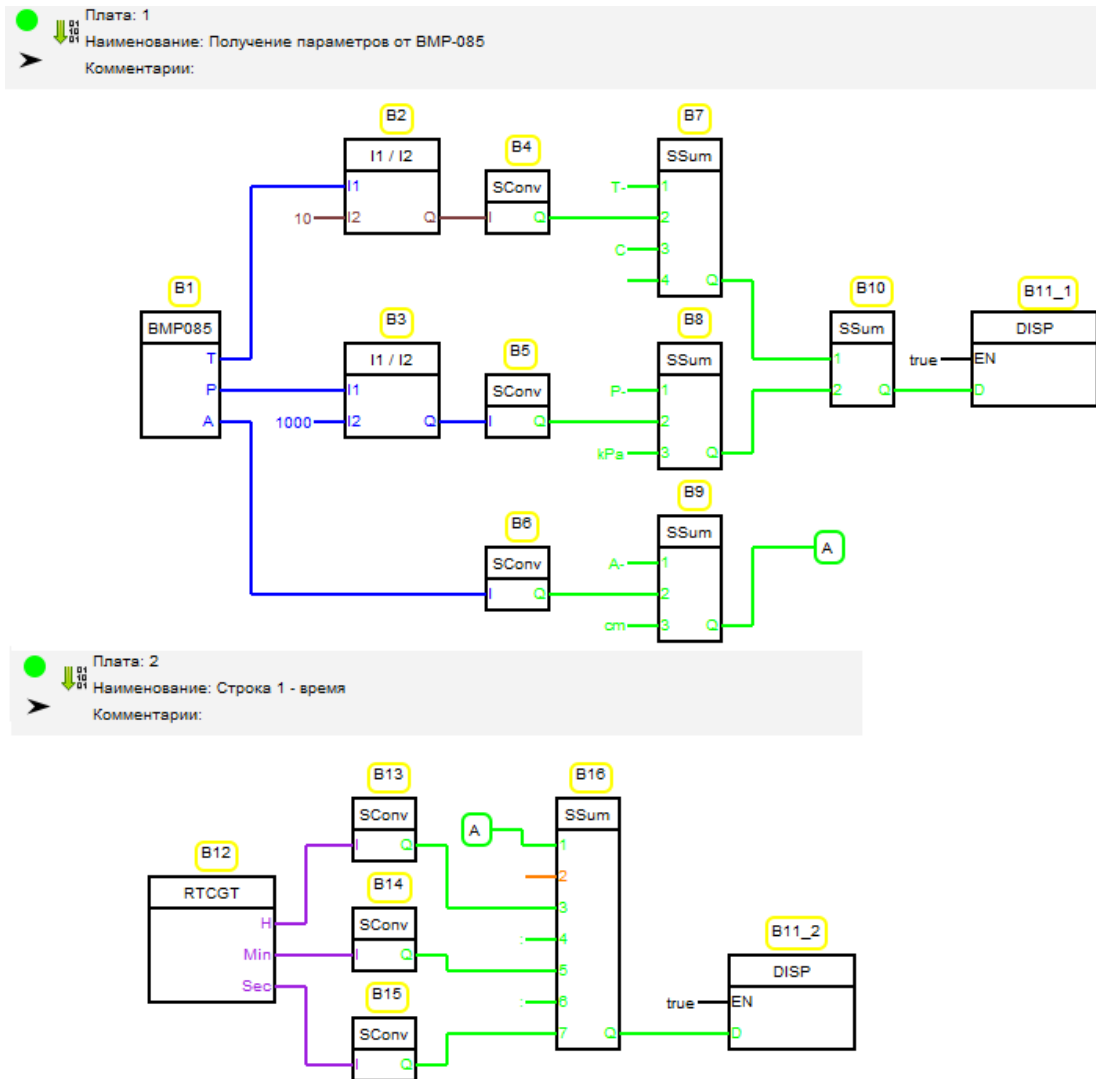
BMP-085 გადამწოდი შეცვალა შემდეგ BMP-180-მა, რომელიც ისევე უკავშირდება არდუინოსა და სხვა მიკროკონტროლერებს, მაგრამ ამავდროულად უფრო იაფი და პატარაა.

BMP-085 გადამწოდის ტექნიკური მახასიათებლებია:

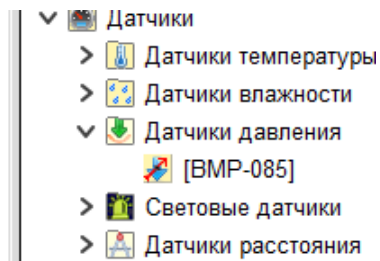
- მგრძობიარობის დიაპაზონი: (9000 მ ... 500 მ ზღვის დონიდან);
- გარჩევადობა: 0.25 მ;
- სამუშაო ტემპერატურა: -40...+85;
- ტემპერატურის გაზომვის სიზუსტე: +2 გრადუსი ცელსიუსით;
- მიერთების ინტერფეისი: I2C;
- კვება: +5 ვ.

BMP-085 გადამწოდის მართვის პროგრამა მოყვანილია ფიგ. 40.1-ზე. პროექტის სრულყოფილების გაზრდის მიზნით ჩვენს მიერ მასში შეტანილ იქნა რეალური დროის საათიც B11. იმისათვის, რომ პროგრამაში გამოვიყენოთ წნევის გადამწოდი უნდა შევიდეთ FLProg-ის ბლოკების ბიბლიოთეკის გადამწოდების საქალაქდემი-Датчики (ფიგ. 40.2), ავირჩიოთ იქ წნევის გადამწოდი-Датчик давления BMP-085 და გადავიტანოთ იგი პროგრამის სამუშაო ზონაში B1. ამის შემდეგ ჩვენ უნდა დავაპროგრამოთ იგი.

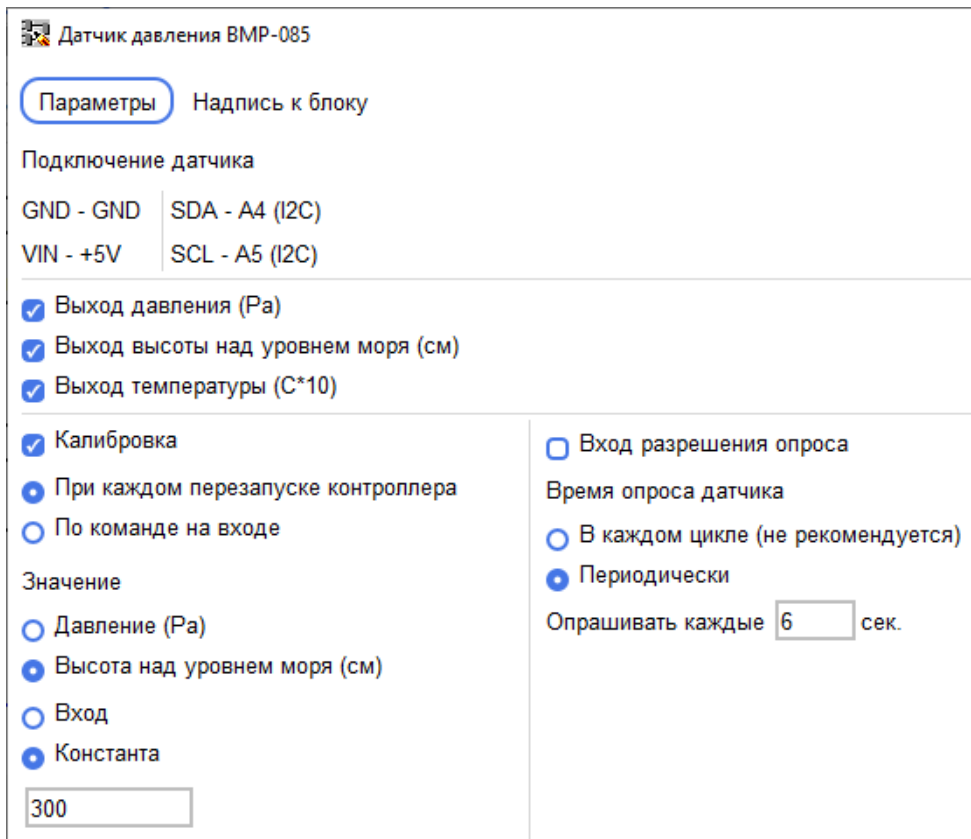
ამისათვის ვაწკაპუნებთ ამ ბლოკზე და შევდივართ მისი პარამეტრიზაციის ფანჯარაში (ფიგ. 40.3).



ფიგ. 40.1 BMP-085 გადამწოდის და RTCGT რეალური დროის საათის მართვის პროგრამა



ფიგ. 40.2. წნევის გადამწოდის არჩვა ბლოკების ბიბლიოთეკაში



ფიგ. 40.3 წნევის გადამწოდის ბლოკის პარამეტრიზაციის ფანჯარა

როგორც პარამეტრიზაციის ფანჯრიდან ჩანს გადამწოდის გამომყვანების შეერთებები არდუინოს კონტაქტებთან წინასწარაა გაწერილი FLProg პროგრამაში, ანუ თუ რომელი გამომყვანი უნდა შეერთდეს არდუინოს რომელ კონტაქტთან. ასე, რომ გადამწოდის მიერთების ველში-Подключение датчика: GND შეერებულია არდუინოს GND-სთან; VIN შეერთებულია +5 ვოლტთან; SDA-A4-თან და SCL -A5-თან.

B1 ბლოკს აქვს სამი გამოსასვლელი: წნევის გამოსასვლელი-Выход давления (Pa); ზღვის დონიდან სიმაღლის გამოსასვლელი-Выход высоты над уровнем моря (см); და ტემპერატურის გამოსასვლელი-Выход температуры (C*10). მოვნიშნოთ სამივე გამოსასვლელის ალამი. აქვე მოვნიშნოთ დაკალიბრების-Калибровка ალამი და ჩავრთოთ ჩამრთველი-კონტროლერის ყოველი გადატვირთვისას-При каждом перезапуске контроллера. შემდეგ, ფანჯრის მარცხენა ქვედა ნაწილში ჩავრთოთ ჩამრთველები სიმაღლე ზღვის დონიდან-Высота над уровнем моря (см) და კონსტანტა-Константа და შესაბამის ველში ჩავწერთ 300 (სიმაღლის ზღვის დონიდან საშუალო მაჩვენებელი თბილისისათვის). ფანჯრის მარჯვენა მხარეს ჩავრთოთ პერიოდული-Периодически, რითაც მივუთითებთ, რომ გადამწოდთან წვდომა და მონაცემების მოთხოვნა იქნება პერიოდულად და ველში-გამოიკითხოს ყოველ-Опрашивать каждые ველში ჩავწერთ 6, რითაც მიეთითება არდუინოს, რომ ამ გადამწოდის გამოკითხვა უნდა მოხდეს ყოველ 6

წამში ერთჯერ. ბლოკის პარამეტრიზაციის შემდეგ არ უნდა დაგვავიწყდეს მზადყოფნის დილაკზე-Горно თავის დილაკით დაჭერა.

ამის შემდეგ უნდა გადავიდეთ B1 ბლოკიდან გამომავალი სიგნალების დამუშავებაზე. ეს პროცესი გულისხმობს, რომ ტემპერატურის გამოსასვლელი სიდიდე T უნდა გაიყოს 10-ზე, ხოლო წნევის გამოსასვლელი სიდიდე P-1000, ამ ფუნქციებს ასრულებენ მათემატიკური ბლოკები B2 და B3, ხოლო მესამე სიდიდეს A-ს ანუ სიმაღლეს ზღვის დონიდან-დამუშავება არ ჭირდება. მიღებულ სიდიდეებს ჩვეულებრივად დაჭირდებათ სტრიქონებად გარდამნა და ამას შეასრულებენ B4, B5, B6 კონვერტაციის ბლოკები.

ამის შემდეგ, ჩვენ უნდა დავუმატოთ თვითოეულ ან სიდიდეს შესაბამისი აღნიშვნები, ამიტომ გვჭირდება სტრიქონების შეკრების ბლოკების B7, B8, B9 გადმოტანა. T სიგნალის გაფორმებისათვის გვჭირდება ოთხ შესასვლელიანი სტრიქონების შეკრების ბლოკი B7, რომლის პირველ შესასვლელზე ჩავწერთ კონსტანტა T., მეორე შესასვლელზე შეუერთოთ გასაზომი სიდიდე-ტემპერატურა, მესამე კონტაქტზე ჩავწერთ ტემპერატურის განზომილება C, მეოთხეზე შესასვლელზე კი - კონსტანტა პრაბელი, რათა სტრიქონში მოხდეს პირველი გასაზომი სიდიდის T-ს გამოყოფა მეორესაგან. მეორე ბლოკი B8 სამშესასვლელიანია, პირველ მის შესასვლელზე უნდა ჩავწერთ კონსტანტა P. მეორეზე შევავერთოთ გასაზომი სიდიდე-წნევა B5 ბლოკიდან, ხოლო მესამეზე-ჩავწერთ წნევის განზომილება kPa. ეს ორი გასაზომი სიდიდე თავის აღნიშვნებთან ერთად ავსებს პირველი სტრიქონის თექვსმეტივე პოზიციას, ამიტომ კიდევ ერთჯერ შევკრიბავთ ამ ორ მონაცემს B10 შეკრების ბლოკის მეშვეობით და მივაერთებთ დისპლეის B11_1 პირველი სტრიქონის ანალოგურ შესასვლელზე.

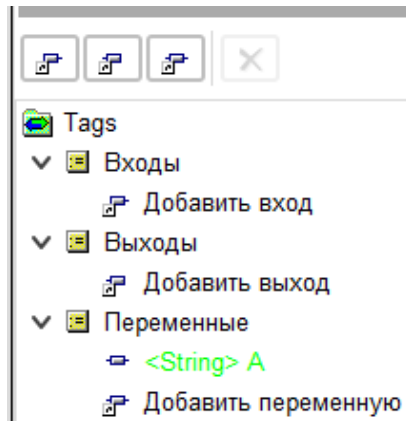
მესამე გასაზომი სიდიდის-სიმაღლის ზღვის დონიდან სტრიქონის ფორმირებისთვის ვიყენებთ სტრიქონების შეკრების ბლოკს B9-ს, რომლის პირველ ფეხზეც ჩავწერთ კონსტანტას A., მეორე ფეხზე შეგვყავს გასაზომი სიდიდე-სიმაღლე ზღვის დონიდან B6 ბლოკიდან, ხოლო მე-3 ფეხზე ჩავწერთ sm-ს კონსტანტას. რადგან ეს ახალი ფორმირებული გასაზომი სიდიდე დისპლეის პირველ სტრიქონში აღარ ეტევა, ამიტომ უნდა შევქმნათ String ტიპის ცვლადი A (ფიგ. 40.4) და შევუერთოთ იგი B9 ბლოკის გამოსასვლელს პირველ დაფაში.

შემდეგ, დავაკოპიროთ ეს ცვლადი, ჩავსვათ მეორე დაფაზე და გამოვიყენოთ იგი დისპლეის მეორე სტრიქონის ფორმირებისათვის. ამ მიზნით უნდა შევუერთოთ იგი სტრიქონების შეკრების ბლოკის B16-ს პირველ შესასვლელს. ეს ბლოკი შვიდშესასვლელიანია (ფიგ. 40.5), რომლის მე-3, მე-5 და მე-7 შესასვლელებზე მიერთებულია რეალური დროის საათის B12 საათების H, წუთების Min, წამების Sec-ის გამოსასვლელები B13, B14 და B15

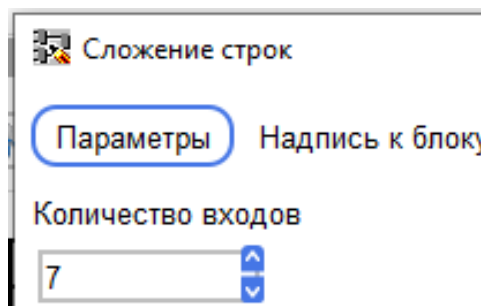
კონვერტაციის ბლოკების გავლით. ამავე სტრიქონების შეკრების ბლოკში მე-2, მე-4 და მე-6 შესასვლელელებზე ჩაწერილია კონსტანტები: პრაბელი, ორი წერტილი და კიდევ ორი წერტილი.

ამის შემდეგ უნდა მოვახდინოთ დისპლეის ქვებლოკების პარამეტრიზაცია, რაც უნდა შევასრულოთ ფიგ. 40.6 და ფიგ. 40.7-ზე ასახული პარამეტრიზაციის ფანჯრების შესაბამისად.

გავუშვათ პროგრამის კომპილაცია, შემდეგ გადავტვირთოთ იგი არდუინოში და ვაკონტროლოთ გადამწოდის მუშაობა. წნევის გადამწოდის Fritzing სქემა ასახულია ფიგ. 40.8-ზე.



ფიგ. 40.4 ტეგების ზონა



ფიგ. 40.5 სტრიქონების შეკრების ბლოკის პარამეტრიზაცია

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/> <input type="button" value="v"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/> <input type="button" value="v"/>	Комментарий	<input type="text"/>	

Данные:

Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 40.6 დისპლეის პირველი სტრიქონის პარამეტრიზაციის ფანჯარა

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/> <input type="button" value="v"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/> <input type="button" value="v"/>	Комментарий	<input type="text"/>	

Данные:

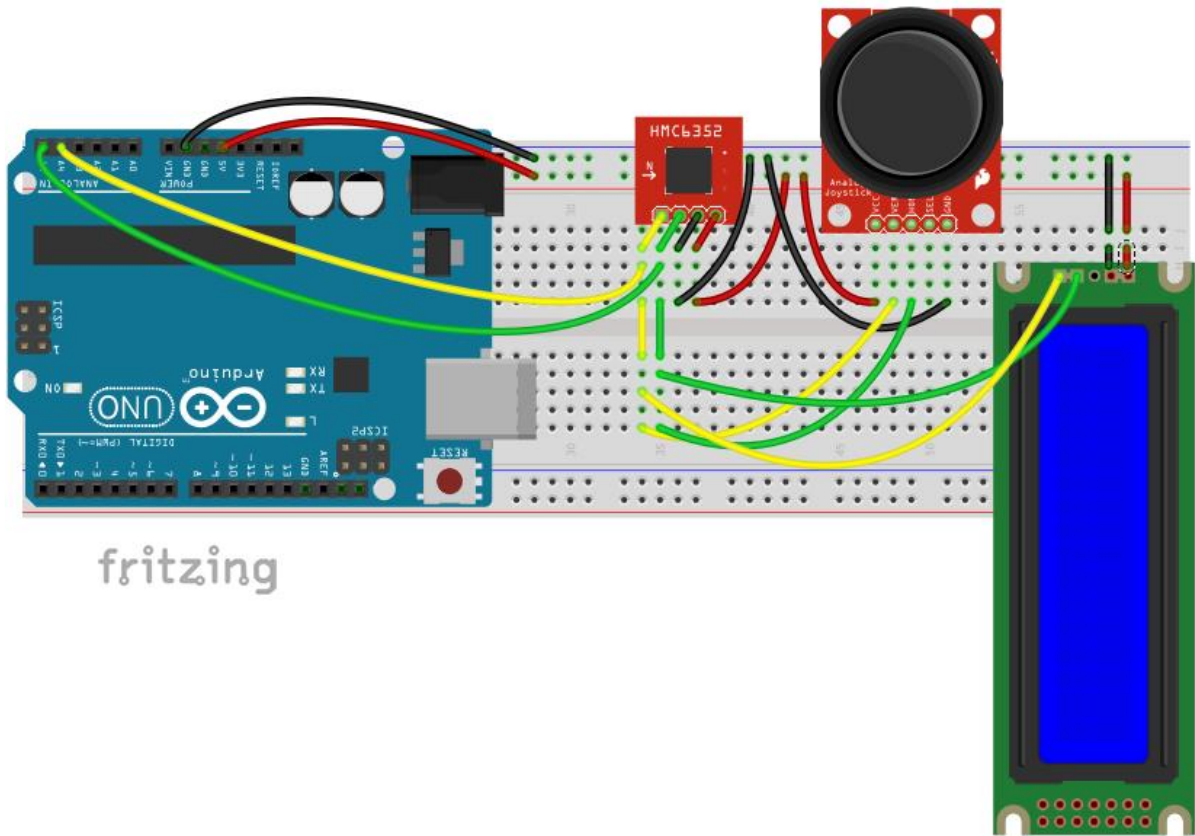
Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 40.7 დისპლეის მეორე სტრიქონის პარამეტრიზაციის ფანჯარა



ფიგ. 40.6-ზე წნევის გადამწოდის BMP-085-ის მართვის Fritzing სქემა

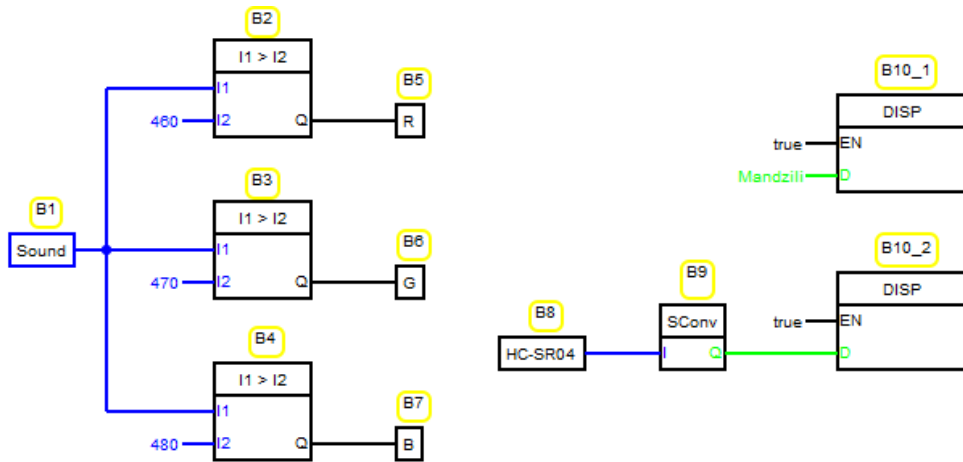
პროექტი_41

ბგერისა (Sound) და ულტრაბგერითი (Ultra Sound) გადამწოდები



პროექტში გამოყენებული იქნება მიკროფონური ბგერის სენსორის მოდული (ა), ულტრაბგერითი სენსორის მოდული HC-SR04 (ბ) და სამფერა RGB შუქდიოდი (გ). გავუშვებ პროგრამა FLprog, შევქმნათ ახალი პროექტი, ავირჩიოთ დაპროგრამების ენად FBD, ხოლო კონტროლერის ტიპად Arduino Uno. ტეგების ზონაში შევქმნათ ანალოგური შესასვლელი სახელწოდებით Sound და შეუერთოთ ის არდუინოს ანალოგურ შესასვლელ კონტაქტს A0.

თავის მარცხენა ღილაკით გადმოვიტანოთ ეს შესასვლელი პროგრამის სამუშაო ზონაში და დავსვათ ის ისე, როგორც ნახაზზეა ნაჩვენები (ფიგ. 41.1).



ფიგ. 41.1 Sound და Ultra Sound სენსორების მართვის პროგრამა

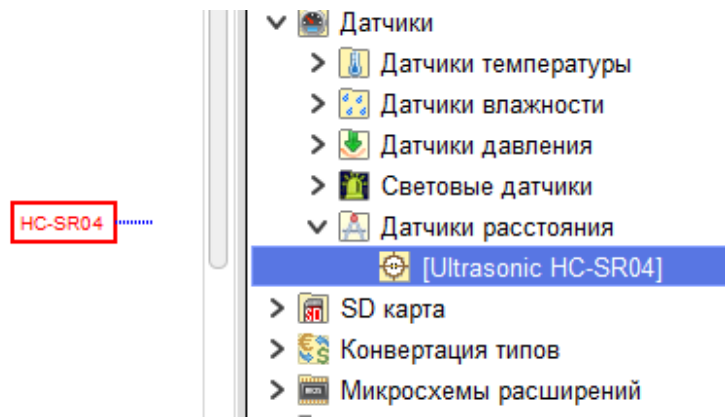
ტეგების ზონაში შევქმნათ აგრეთვე სამი გამოსასვლელი სახელწოდებით R, G და B სამფერა RGB დიოდთან მუშაობის საჭიროებისათვის და შეუერთოთ ისინი არდუინოს დისკრეტული შესასვლელების მე-3, მე-4 და მე-5 კონტაქტებს. ბლოკების ბიბლიოთეკის შედარებების საქალაქედან - Сравнение გადმოვიტანოთ ბლოკი Comparator B2, დავაპროგრამოთ იგი ისე, რომ პირველი შესასვლელი მეტი იყოს მეორე შესასვლელზე $I1 > I2$, მოვახდინოთ ამ ბლოკის კოპირება ორჯერ, აღვნიშნოთ ეს ახალი ბლოკები B3 და B4. განვალაგოთ ისინი ნახ. 41.1-ის შესაბამისად. შეუერთოთ ამ ბლოკების I1 შესასვლელები B1 ბლოკის გამოსასვლელთან.

ახლა გადმოვიტანოთ ტეგების ზონიდან სამი გამოსასვლელი R, G და B, მივანიჭოთ მათ ბლოკის ნომრები B5, B6, B7 და შეუერთოთ ისინი B2, B3, B4-ის გამოსასვლელებს შესაბამისად. ახლა უნდა დავაპროგრამოთ B2, B3, B4 ბლოკების I2 შესასვლელები. ჩვენ ვიცით, რომ B1 ბლოკის გამოსასვლელზე ციფრულმა სიგნალმა თეორიულად იძლება მიიღოს ნებისმიერი მნიშვნელობა 0...1023 დიაპაზონში. აი ამ დიაპაზონიდან უნდა იქნას შერჩეული I2 მნიშვნელობები.

ექსპერიმენტულად დადგენილ იქნა, რომ თუ ავიღებთ 460, 470, 480 მნიშვნელობებს, როგორც ეს ნახაზზეა წარმოდგენილი (ფიგ. 41.1), მაშინ მაგიდაზე, სადაც ეს Sound სენსორია განლაგებული, მის მახლობლობაში, მცირედ დაკაკუნებით, სიგნალი B1 ბლოკის გამოსასვლელზე მცირედ გადააჭარბებს 460-ს, ამუშავდება B2 ბლოკი და მაღალი ძაბვა გამოვა მხოლოდ ამ ბლოკის გამოსასვლელზე, თუ კაკუნს ცოტა უფრო გავაძლიერებთ, მაშინ სიგნალი გადააჭარბებს როგორც 460 ასევე 470-ს და 5ვ გამოვა როგორც B2, ასევე B3 ბლოკების

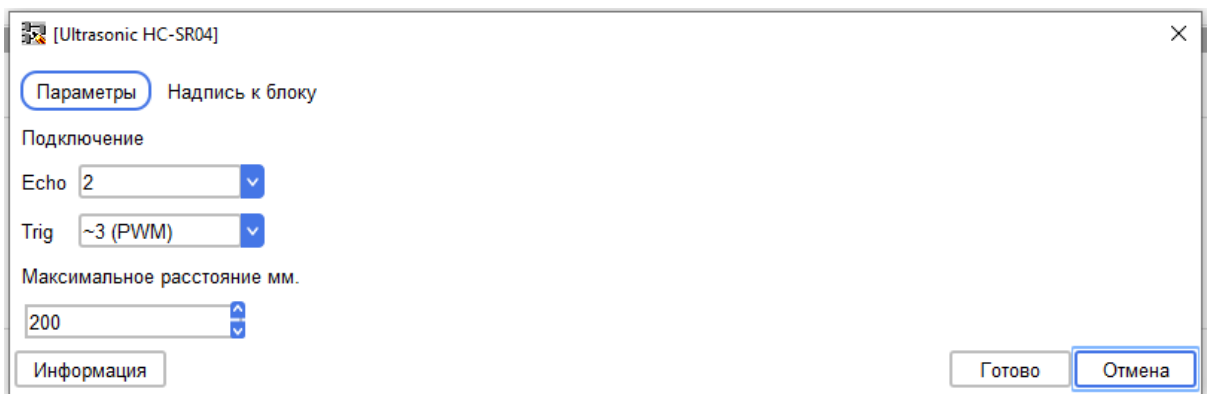
გამოსასვლელზე, თუ კი ძლიერად დავაკაკუნებთ მაშინ სიგნალი გადააჭარბებს სამთავე დონეს, ამუშავდება სამმივე ბლოკი და 5ვ ძაბვა გამოვა სამივე ბლოკის გამოსასვლელზე. რადგანაც ამ ბლოკის გამოსასვლელები B5, B6, B7 ბლოკების გავლით შეერთებული უნდა იყოს RGB შუქდიოდის შესაბამის გამომყვანებთან, ამიტომ პირველ შემთხვევაში უნდა აინთოს წითელი ფერი, მეორე შემთხვევაში - წითელი და მწვანე, ხოლო მესამე შემთხვევაში სამთავე წითელი, მწვანე და ლურჯი ერთად.

ამის შემდეგ ვიწყებთ ულტრაბგერითი სენსორის დაპროგრამებას. ამისათვის ბლოკების ბიბლიოთეკიდან, გადამწოდების საქალაღედან-Датчики შევიდეთ მანძილის გადამწოდებში-Датчики расстояния, იქ კი მოვნიშნოთ ბლოკი [Ultrasonic HC-SR04] და გადმოვიტანოთ პროგრამის სამუშაო ზონაში (ფიგ. 41.2). პროგრამაში ეს ბლოკი აღნიშნულია B8 ნიშნით. ვაწკაპუნებთ ამ ბლოკზე და ვახდენთ მის დაპროგრამებას (ფიგ. 41.3).



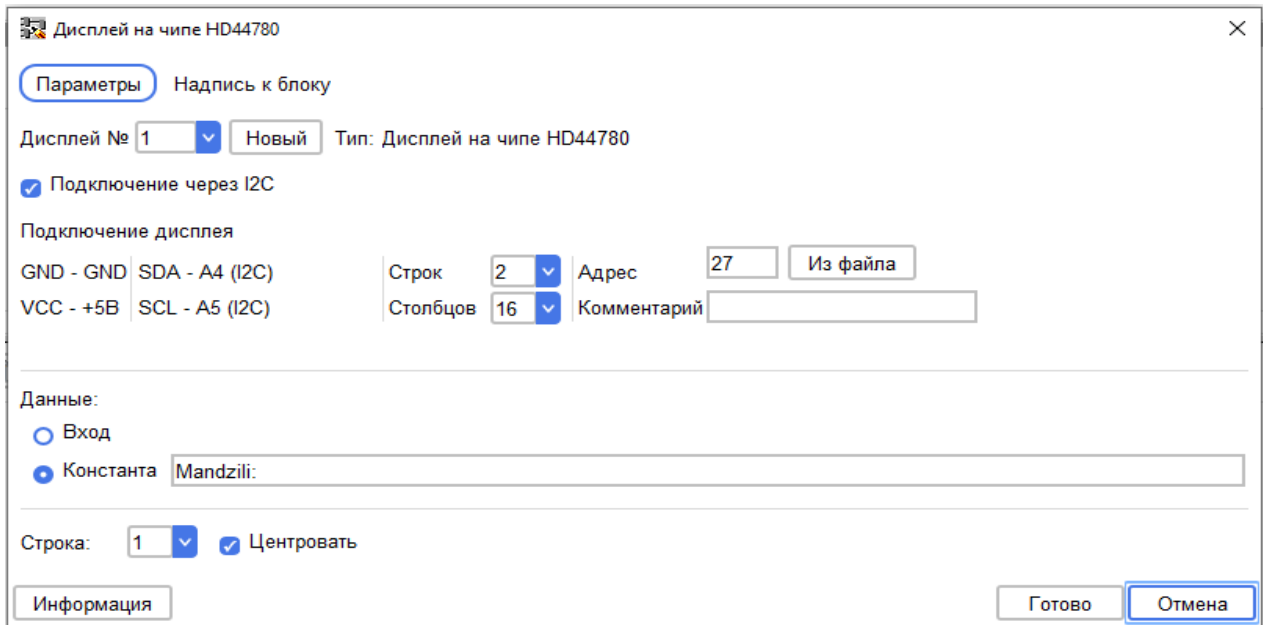
ფიგ. 41.2. ულტრაბგერითი გადამწოდის გადმოტანა სამუშაო ზონაში

Echo-სა და Trig ველებში ვწერთ არდუინოს რომელიმე თავისუფალ კონტაქტებს, ჩვენს შემთხვევაში მე-2 და მე-3 კონტაქტებს, მაქსიმალურ მანძილს მმ - Максимальное расстояние мм ვტოვებთ უცვლელად - 200 მმ და მზადყოფნის დილაკზე Готово დაწკაპუნებით გამოვდივართ ამ სენსორის დაპროგრამების ფანჯრიდან.



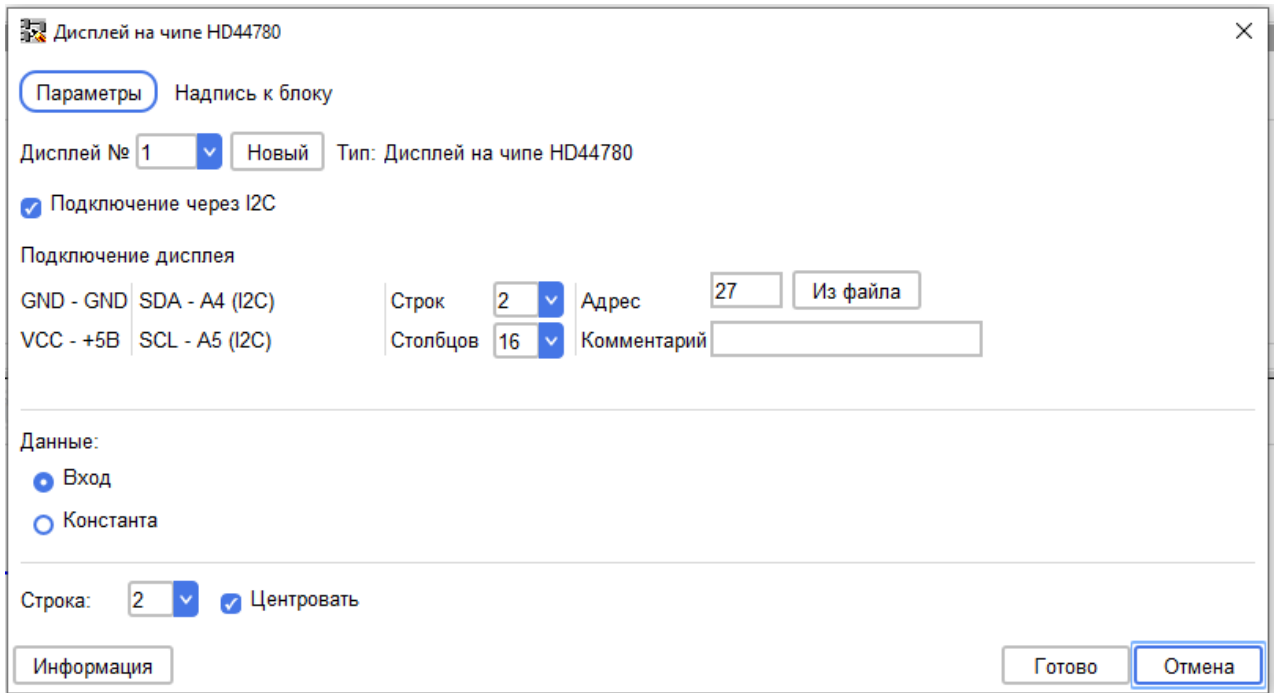
ფიგ. 41.3 ულტრაბგერითი გადამწოდის დაპროგრამების ფანჯარა

ახლა შევიდეთ ისევ პროგრამის ბიბლიოთეკაში, გავხსნათ დისპლეების საქაღალდე Дисплей, მოვნიშნოთ დისპლეი ჩიპზე - Дисплей на чипе HD44780 და გადმოვიტანოთ პროგრამის სამუშაო ზონაში.



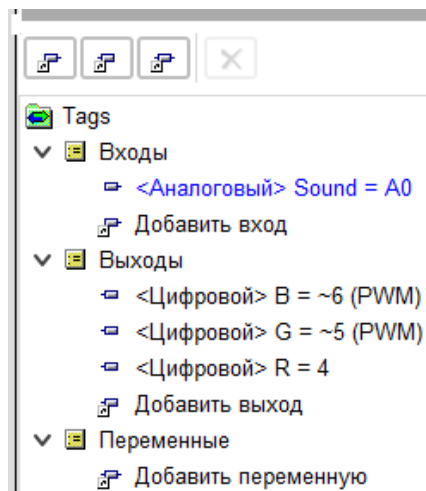
ფიგ. 41.4 დისპლეის პირველი სტრიქონის პარამეტრიზაციის ფანჯარა

აქ მოგვიწევს ორივე სტრიქონის ცალცალკე დაპროგრამება B10_1 და B10_2 ბლოკებში შესაბამისად. დავაწკაპოთ B10_1 ბლოკს, შევიდეთ მისი პარამეტრიზაციის ფანჯარაში და შევავსოთ ველები ისე, როგორც ფიგ. 41.4-ზეა ნაჩვენები. მიაქციეთ ყურადღება, რომ უნდა ჩაირთოს ჩამრთველი კონსტანტა- Константа შესაბამის ველში ჩაიწეროს ლათინური ასოებით მანძილი-Mandzili: დავაჭიროთ თავის თითით მზადყოფნის დილაკზე-Готово. ამის შემდეგ უნდა დაპროგრამირდეს მე-2 სტრიქონის ფანჯარა. ამისათვის ვაწკაპუნებთ B10_2 ბლოკს და შევდივართ მისი დაპროგრამების ფანჯარაში, ვავსებთ მის ველებს ისე, როგორც ეს ფიგურა 41.5-ზეა ნაჩვენები. ამის შემდეგ ვაწკაპებთ Готово-ს და ამით ჩვენი პროექტის დაპროგრამებას დავამთავრებთ.

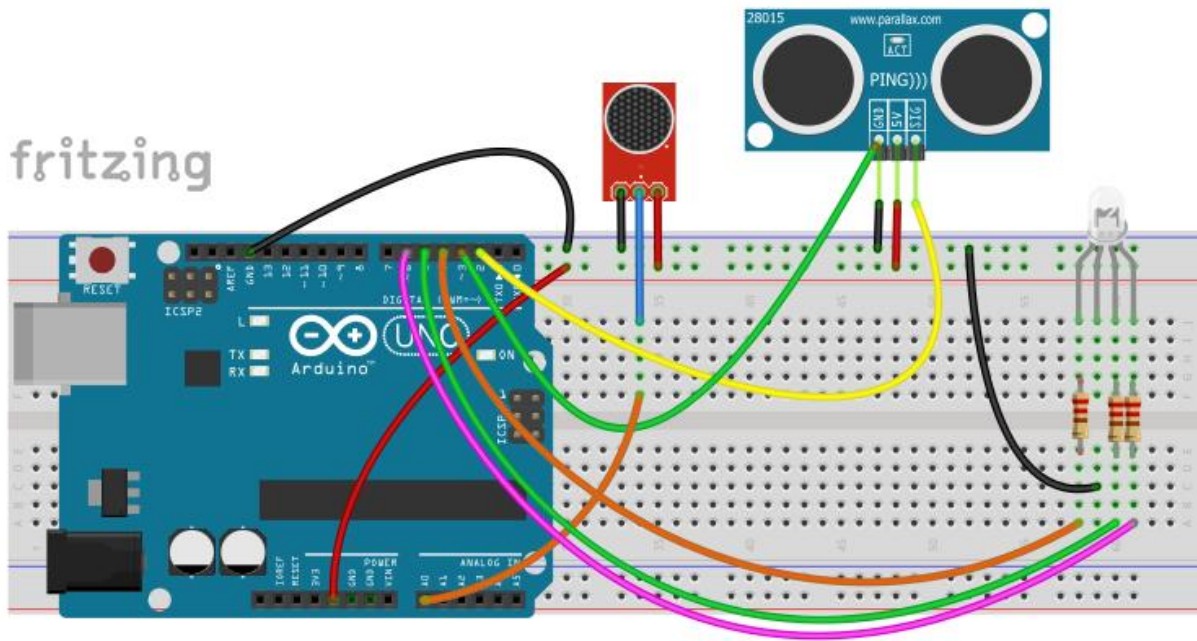


ფიგ. 41.5 დისპლეის მეორე სტრიქონის პარამეტრიზაციის ფანჯარა

პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 41.6-ზე, ხოლო მისი ფიზიკური აწყობა უნდა მოხდეს ფიგურა 41.7-ზე წარმოდგენილი Fritzing სქემის მიხედვით.

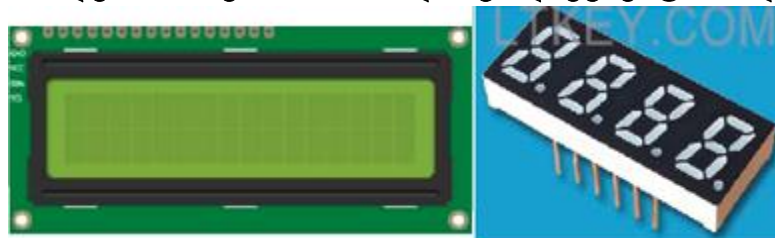


ფიგ. 41.6 პროექტის ტეგების ზონის ფანჯარა



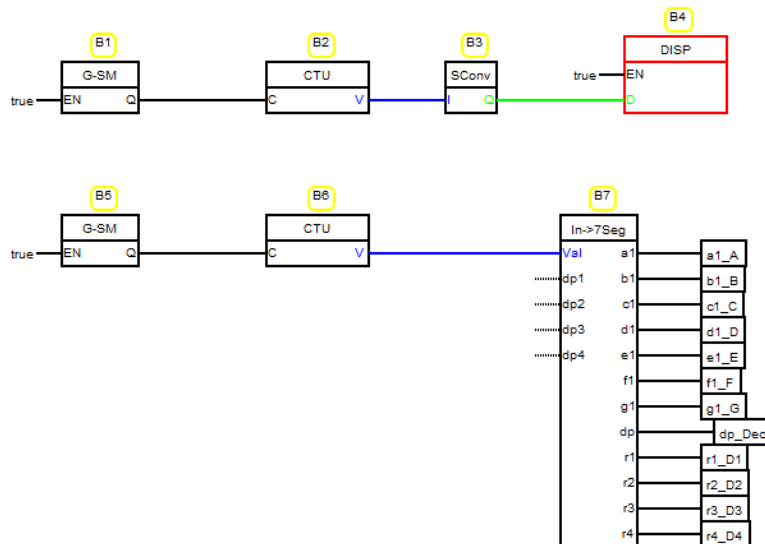
ფიგ. 41.7. ბგერის სენსორისა და ულტრაბგერითი სენსორის შეერთებების Fritzing სქემა

პროექტი_42 დისპლეი ჩიპზე HD44780 და შვიდსეგმენტა ინდიკატორი



ქვემოთ წარმოდგენილია ორი დისპლეის: ორსტრიქონიანი თექვსმეტსვეტიანი დისპლეის ჩიპზე-HD44780 და ოთხთანრიგა შვიდსეგმენტიანი-LED დისპლეის მართვის სქემები FLProg გარემოში. გავუშვით FLProg-ის პროგრამა, დაპროგრამების ენად-Язык программирования ავირჩიოთ FBD, ხოლო კონტროლერად-Контроллер: - Arduino Uno

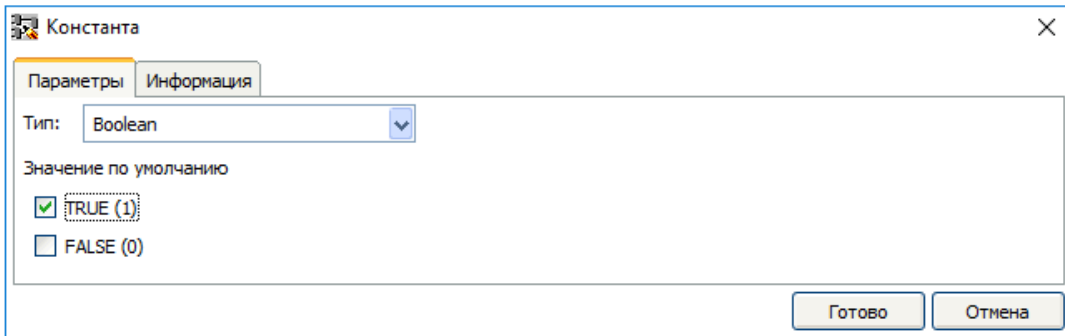
ამის შემდეგ, გადავიდეთ FLProg პროგრამის ელემენტების ბიბლიოთეკაში და დისპლეის საქადალდედან Дисплей პროგრამის სამუშაო ზონაში გადმოვიტანოთ ჯერ დისპლეის ბლოკი ჩიპზე-Дисплей на чипе HD7780 (B4) და შემდეგ შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკი-Блок декодирования семисегментного индикатора (B7) (ფიგ. 42.1).



ფიგ. 42.1 დისპლეის ჩიპზე HD7780 და შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკები

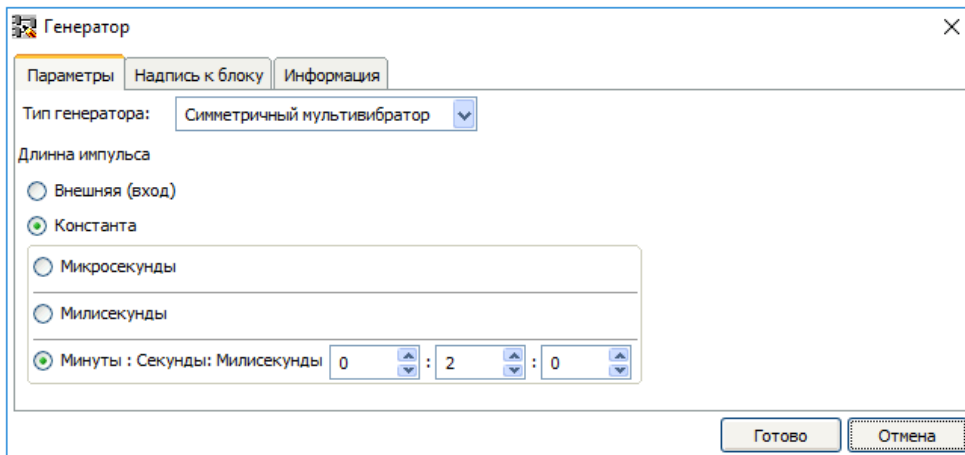
გარდა ამისა, პირველი დისპლეის მართვისათვის პროგრამის ბლოკების ბიბლიოთეკიდან სამუშაო ზონაში გადმოვიტანოთ: ტაიმერების საქაღალდედან-Таймеры-გენერატორის ბლოკი-Generator (B1), მთვლელების საქაღალდედან-Счетчики- მთვლელის ბლოკი-Counter (B2) და ტიპების კონვერტაციისათვის-Конвертация типов) - სტრიქონების კონვერტაციის ბლოკი-Конвертация строк (B3). შვიდსეგმენტა ინდიკატორის მართვისათვის პროგრამის ბლოკების ბიბლიოთეკიდან სანუშაო ზონაში გადმოვიტანოთ: ტაიმერების საქაღალდედან Таймеры-გენერატორის ბლოკი-Generator (B5) და მთვლელების საქაღალდედან-Счетчики-მთვლელის ბლოკი-Counter (B6).

ამის შემდეგ ვიწყებთ აღნიშნული ბლოკების პარამეტრიზაციას. დავაწკაპუნოთ B1 ბლოკის EN შესასვლელზე თავის მარჯვენა ღილაკით და გამოსულ ფანჯარაში ავირჩიოთ კონსტანტის ჩასმა-Вставить константу, რის შედეგადაც გამოჩნდება კონსტანტის არჩევის ფანჯარა (ფიგ. 42.2), სადაც ჩავრთავთ ალამს TRUE (1), შემდეგ ვაწკაპუნებთ მზადყოფნის ღილაკს-Готово თავის მარცხენა ღილაკით, რის შედეგადაც ეს ფანჯარა გაქრება ხოლო B1 ბლოკის EN შესასვლელზე გაჩნდება წარწერა true, რაც იმას ნიშნავს, რომ პროგრამის გაშვების შემდეგ ამ გენერატორის მუშაობა ნებადართულია მუდმივად ამ პროგრამის მუშაობის გაჩერებამდე.



ფიგ. 42.2 კონსტანტის პარამეტრიზაციის ფანჯარა

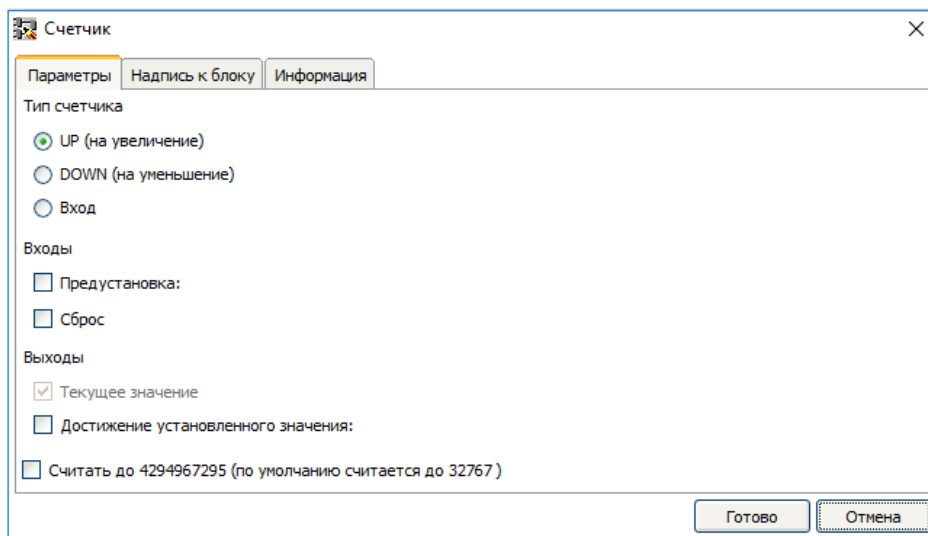
ამის შემდეგ დავაწკაპუნოთ თვითონ ამ ბლოკში B1 და გამოსულ ფანჯარაში (ფიგ. 42.3) გენერატორის ტიპის ველში-Тип генератора ავირჩიოთ სიმეტრიული მულტივიბრატორი-Симметричный мультивибратор, იმპულსის სიგრძის ველში-Длина импульса ჩავრთოთ კონსტანტის ჩამრთველი-Константа და წამების ველში-Секунды ჩავწეროთ ჩვენთვის სასურველ დროის ინტერვალს, რა ინტერვალითაც გვინდა განახლდეს ინფორმაცია დისპლეიზე. ჩვენს მაგალითში ჩავწეროთ ორი, დავაჭიროთ მზადყოფნის ღილაკს თავის მარცხენა ღილაკით და ამით დავამთავროთ B1 გენერატორის პარამეტრიზაცია. ასევე მოვიქცეთ B5 გენერატორის დაპროგრამებისთვისაც, იმ განსხვავებით, რომ იქ განსხვავებულობისთვის წამების ველში ჩავწეროთ სხვა დრო მაგალითად სამი (თუმც, იქაც რატემაუდა შეგვეძლო ჩავვწერა ორი).



ფიგ. 42.3 გენერატორის პარამეტრიზაციის ფანჯარა

ახლა გადავიდეთ B2 მთვლელის პარამეტრიზაციაზე. ამ მიზნით დავაწკაპუნოთ ამ ბლოკზე ორჯერ თავის მარცხენა ღილაკით, რის შედეგადაც გამოჩნდება ფანჯარა (ფიგ. 42.4) სადაც ამ შემთხვევაში ჩვენი ამოცანის გადაწყვეტისათვის არაფერის ხელის ხლება არ არის საჭირო და დავტოვოთ ისე როგორც არის მოცემული სიჩუმით. ანუ მთვლელი დაიწყებს

თვლას პირდაპირი მიმართულებით მასზედ იმპულსების მიწოდების შედეგად, რადგან ჩართულია ჩამრთველი ზრდაზე UP-на увеличение. დანაჩენ ალმებს დავტოვებთ დაუყენებელს, რაც იმას ნიშნავს, რომ არ მოხდება მთვლელში არავითარი რიცხვის წინასწარი ჩაწერა და არ მოხდება მთვლელის წინასწარი განულება და გენერატორიდან იმპულსების მიწოდების შემთხვევაში მთვლელი დაითვლის თავის შესაძლებელ მაქსიმალურ მნიშვნელობამდე ანუ 32767-მდე და შემდეგი იმპულსი გაანულებს მთვლელს და შემდეგ თვლის პროცესი დაიწყება თავიდან. ამის შემდეგ დავაწვებით მზადყოფნის ღილაკს-Готово, რითაც მთვლელის დაპროგრამება დამთავრებული იქნება. ასევე მოვიქცევით მე-2 მთვლელის-B6 დაპროგრამების დროსაც.

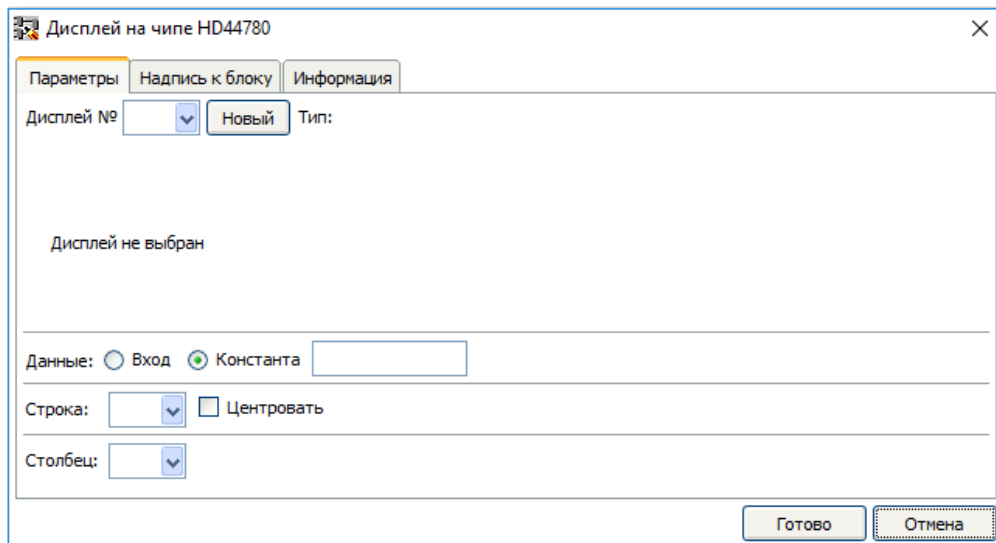


ფიგ. 42.4 მთვლელის პარამეტრიზაციის ფანჯარა

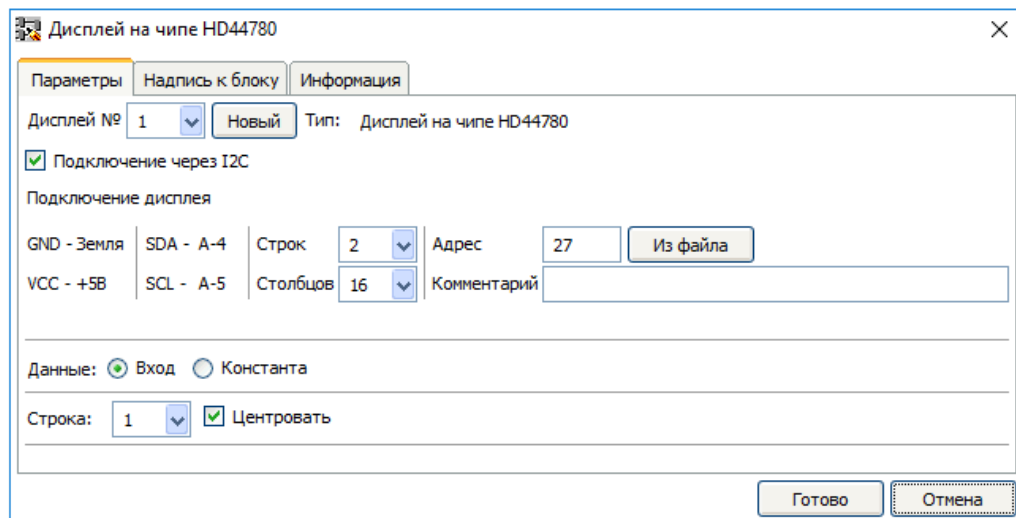
ახლა, ვინაიდან პირველი ინდიკაციის ბლოკს შეუძლია ინფორმაციის მიღება სტრიქონული ფორმატით, ამიტომ აუცილებელია კონვერტაციის ბლოკის B3 გამოყენება, რის პარამეტრიზაციასაც დავტოვებთ ისე, როგორც იგი სიჩუმით არის მოცემული ანუ რიცხვს გადააქცევს სტრიქონად.

ახლა გადავიდეთ დისპლეის ბლოკის-B4 პარამეტრიზაციაზე. ამისათვის, ჯერ საჭიროა მის EN შესასვლელზე ჩავწეროთ true და ამით ნება დავართოთ მის მუშაობას მუდმივად, შემდეგ დავაწკაპუნოთ ამ ბლოკზე ორჯერ და გამონათებულ ფანჯარაში (ფიგ. 42.5) დავაწვით ღილაკს - ახალი-Новый, რითაც დისპლეის ნომრის ველში-Дисплей № ავტომატურად ჩაიწერება რიცხვი 1 (რადგან პროექტში ერთი დისპლეია გამოყენებული ამიტომ მისი მნომერი იქნება პირველი). აქ ფანჯრის გამოსახულება შეიცვლება და ამ შეცვლილ ფანჯარაში (იმისათვის, რომ ჩვენს ხელთ არსებული დისპლეი მუშაობს მხოლოდ I2C პროტოკოლით) დავაყენოთ

ალამი მიერთება I2C-ს გავლით-Подключение через I2C, რითაც ფანჯარის გამოსახულება კვლავ შეიცვლება და მიიღებს ახალ სახეს (ფიგ. 42.6).



ფიგ. 5.5. დისპლეის პარამეტრიზაციის პირველი ფანჯარა

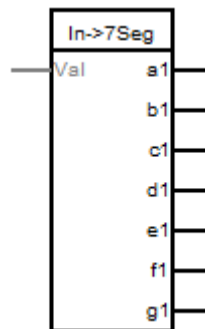


ფიგ. 42.6 დისპლეის პარამეტრიზაციის მესამე ფანჯარა

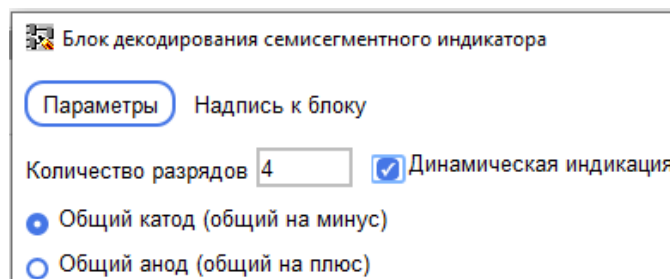
ფიგ. 42.6-ზე მოცემულ ფანჯარაში ზოგიერთი შეერთებები განსაზღვრულია სიჩუმით პროგრამის მიერ, მაგალითად SDA და SCL გამომყვანები უნდა შეერთდეს არდუინოს A4 და A5 კონტაქტებთან შესაბამისად, აგრეთვე GND და VCC უნდა შეერთდეს მიწასთან და +5B - თან შესაბამისად). სტრიქონების-Строк და სვეტების ველებში- Столцов ჩაწეროთ ჩვენ ხელთ არსებული დისპლეის შესაბამისი პარამეტრები-2 და 16 ჩვენს შემთხვევაში. მისამართის ველში-Адрес ჩაწეროთ 27, მონაცემთა ველში-Данные: ჩავრთოთ ჩამრთველი, შესასვლელის-

Вход ველში სტრიქონი-Строка ჩავწერთ ერთიანი, რადგან ჩვენ შემთხვევაში ერთი სტრიქონის გამოყენება სრულიად საკმარისი იქნება, ჩავრთოთ ცენტრირების ალამი Центровать, რათა გამონათებული ინფორმაცია იყოს გასწორებული ცენტრის მიმართ, შემდეგ დავაჭიროთ მზადყოფნის ლილაკს-Готово და ამით დავამთავროთ დისპლეის ბლოკის დაპროგრამება.

ახლა გადავიდეთ შვიდსეგმენტა ინდიკატორის დაპროგრამებაზე. ამისათვის პროგრამის ელემენტების ბიბლიოთეკიდის დისპლეის საქალაღდედან-Дисплей სამუშაო ზონაში გადმოვიტანოთ შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკი-Блок декодирования семисегментного индикатора (ფიგ. 42.7), დავაწკაპუნოთ მასზედ თავის მარცხენა ლილაკით ორჯერ. გამოსულ ახალ ფანჯარაში (ფიგ. 42.8) თანრიგების რაოდენობის ველში-Количество разрядов ჩავწეროთ ოთხი, რადგან ჩვენ ხელთ არსებული შვიდსეგმენტა ინდიკატორი ოთხ თანრიგანია და დავაყენოთ დინამიური ინდიკაციის ალამი (Динамическая индикация), საერთო კათოდის ჩამრთველი- Общий катод დავტოვოთ ჩართული (ანდა ჩავრთოთ Общий анод თუ ასეთი დისპლეი გვაქვს), რის შედეგადაც ბლოკი მიიღებს ახალ სახეს (ფიგ. 42.1- B7).



ფიგ. 42.7 შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკი



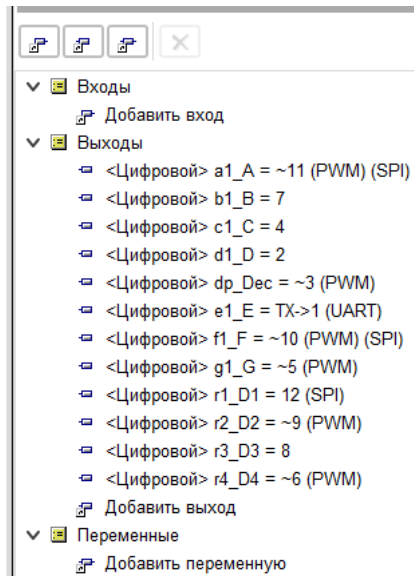
ფიგ. 42.8 შვიდსეგმენტა ინდიკატორის დეკოდირების ბლოკის პარამეტრიზაციის ფანჯარა

ამის შემდეგ გავხსნით ტეგების ზონის (Tags) საქალაღდეს - გამოსასვლელები-Выходы და გამოსასვლელების დამატებაზე-Добавить выходы დაწკაპუნებით თანდათანობით

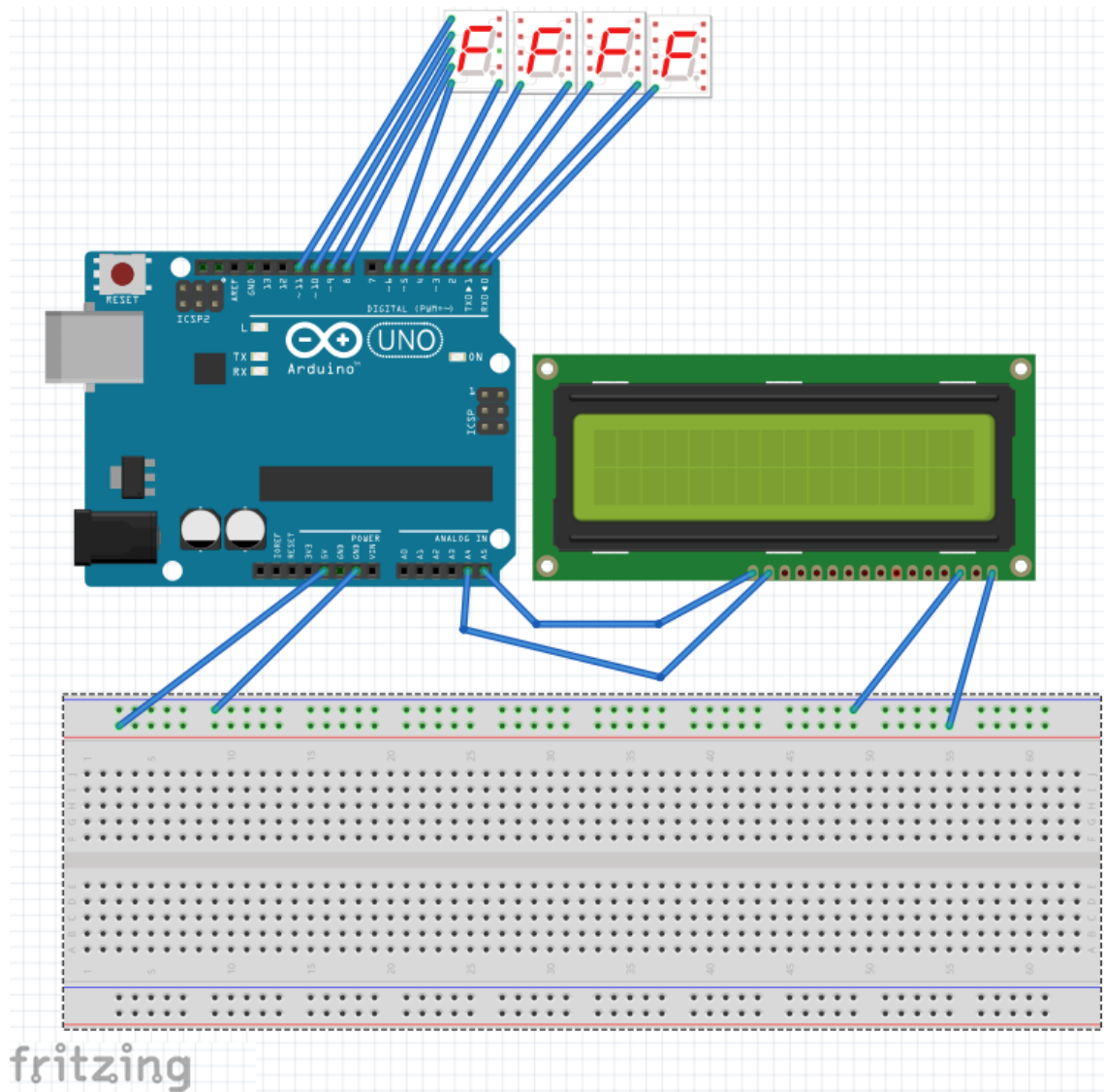
შევექმნათ ის გამოსასვლელები (a1, b1, c1, d1, e1, f1, g1, h1, r1, r2, r3, r4) რომლებიც უნდა შეუერთოთ ინდიკატორის შესაბამის გამომყვანებს (ფიგ. 42.9, ფიგ.42.1).

პროგრამის გაშვების შემდეგ როგორც დისპლეიმ ასევე ინდიკატორმა უნდა დაიწყოს იმპულსების თვლა, მაგრამ რადგან გენერატორები სხვადასხვა სიხშირეებზეა დაპროგრამებული ამიტომ ეს თვლებიც იქნება ასინქრონული ერთმანეთის მიმართებაში.

ფიგ. 42.9 -ზე პროექტის ტეგების ზონის ფანჯარა, ხოლო ფიგ.42.10-ზე-პროექტის ბლოკ სქემა Fritzing პროგრამაში.



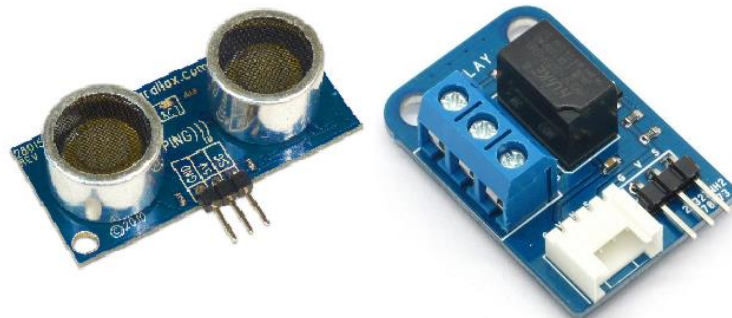
ფიგ. 42.9 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 42.10 პროექტის ბლოკ სქემა Fritzing პროგრამაში

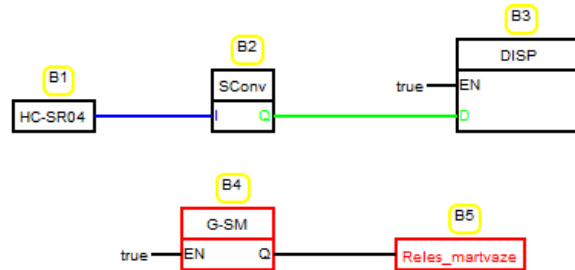
პროექტი_43

ულტრაბგერითი (HC-SR04) და რელური (RELAY) მოდულების მართვა



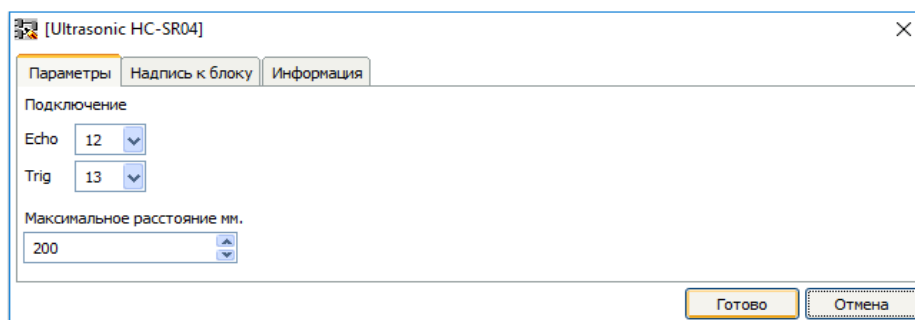
გავუშვათ FLProg-ის პროგრამა, დაპროგრამების ენად-Язык программирования ავირჩიოთ FBD, ხოლო კონტროლერად-Контроллер: - Arduino Uno.

ფიგურა 43.1.-ზე წარმოდგენილია ულტრაბგერითი გადამწოდისა (სენსორი) და რელეური მოდულის მართვის პროგრამები, რომლებიც შესრულებულია Arduino-ს დაპროგრამების გარემოში FLProg. პროგრამები აგებულია FLProg-ის B1, B2, B3 და B4, B5 ბლოკების გამოყენებით.



ფიგ. 43.1 ულტრაბგერითი სენსორისა და რელეური მოდულების მართვის პროგრამა

ულტრაბგერითი სენსორის მართვის დაპროგრამებისათვის საჭიროა FLProg-ის ბლოკების ბიბლიოთეკის-Библиотека элементов გადამწოდების საქაღალდედან-Датчики თავის მარცხენა ღილაკის ჩავლევით პროგრამის სამუშაო ზონაში გადმოვიტანოთ ულტრაბგერითი სენსორის ბლოკი Ultrasonioc-HC-SR04 (B1) და მოვახდინოთ მისი პარამეტრიზაცია (დაპროგრამება).

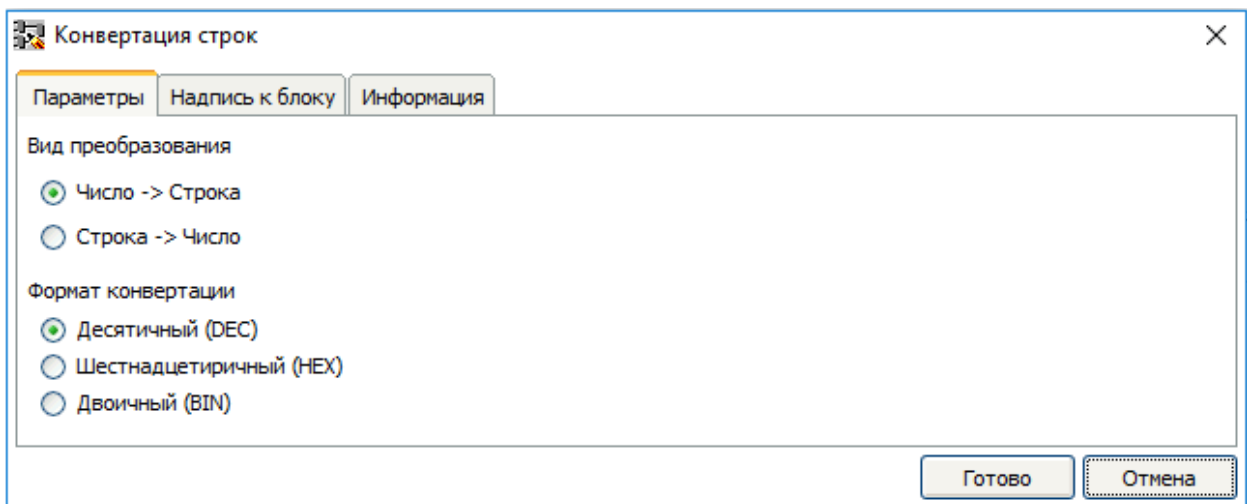


ფიგ. 43.2 ულტრაბგერითი სენსორის დაპროგრამების ფანჯარა

ამ მიზნის მისაღწევად დავაწკაპუნოთ ამ ბლოკზე თავის მარცხენა ღილაკით და შევიდეთ მისი დაპროგრამების ფანჯრაში (ფიგ. 43.2). სენსორს არდუინოსთან კავშირისათვის გააჩნია ოთხი გამომყვანი, ორი მათგანი გამოიყენება კვებისათვის (+5ვ და მიწა) ხოლო ორი წარმოადგენს გადამწოდის მართვის სიგნალებს Echo და Trig. ჩვენ უნდა ავირჩიოთ და ჩავწეროთ შესაბამის ველებში არდუინოს ის კონტაქტები, რომლებზედაც უნდა შევაერთოთ ეს მართვის სიგნალები (მე-12 და მე-13 კონტაქტები ჩვენი პროექტის შემთხვევაში). მაქსიმალური მანძილის ველში-Максимальное расстояние мм. ჩავწეროთ ჩვენითვის სასურველი მაქსიმალური მანძილის სიდიდე, რისი გაზომვაც არის მიზანშეწონილი (ჩვენ ვწერთ 200 მმ ამ

პროექტში) და ამის შემდეგ ვაჭერთ მზადყოფნის ღილაკს-Готово, რითაც ულტრაბგერითი სენსორის დაპროგრამება დამთავრებული იქნება.

ახლა გადავდივართ ულტრაბგერითი სენსორიდან გამოსული გაზომვილი სიდიდეების დამუშავებაზე. ეს ბლოკი-B1 თავის გამოსასვლელზე იძლევა მანძილის მნიშვნელობებს <Integer> ფორმატში, რაც შეესაბამება რიცხვებს -32768 -დან + 32767-მდე. იმისათვის, რომ შევძლოთ ამ რიცხვის გამოსახვა დისპლეიზე საჭიროა მისი გარდაქმნა სტრიქონულ ფორმატში რისთვისაც ვიყენებთ კონვერტორის ბლოკს-(B2). ამისათვის FLProg პროგრამის ელემენტების ბიბლიოთეკის ტიპების კონვერტაციის საქალაღდედან-Конвертация типов თავის მარჯვენა ღილაკით პროგრამის სამუშაო ზონაში გადმოგვაქვს სტრიქონების კონვერტაციის ბლოკი-Конвертация строк. ამ ბლოკზე თავის მარცხენა ღილაკით ორჯერ დაწკაპუნებით კომპიუტერის ეკრანზე აისახება მისი პარამეტრიზაციის ფანჯარა (ფიგ. 43.3).

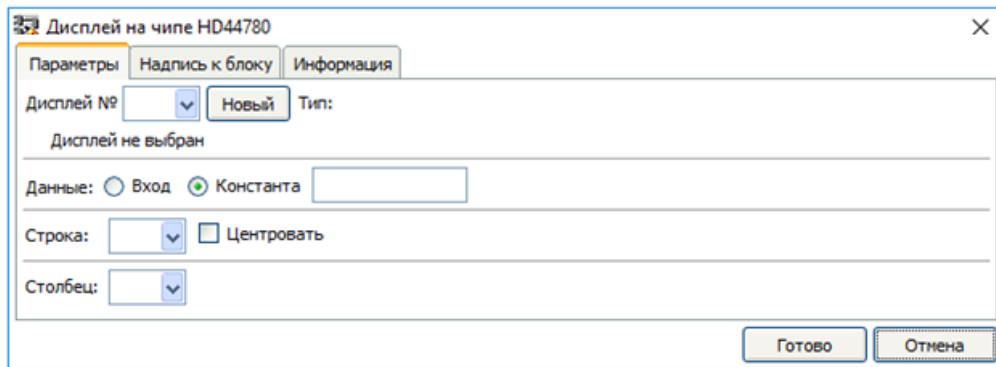


ფიგ. 43.3 ტიპების კონვერტაციის ბლოკის ფანჯარა

ამ ფანჯარაში, ველში - გარდაქმნის სახეობა-Вид преобразования უნდა ჩავრთოთ ჩამრთველი - რიცხვი -> სტრიქონი-Число -> Строка, ხოლო კონვერტაციის ფორმატის ველში-Формат конвертации უნდა ჩავრთოთ ჩამრთველი ათობითი-Десятичный (DEC). შემდეგ დავაჭიროთ თითი მზადყოფნის ღილაკს-Готово და ამით სტრიქონების კონვერტაციის ბლოკის დაპროგრამება დამთავრებული იქნება.

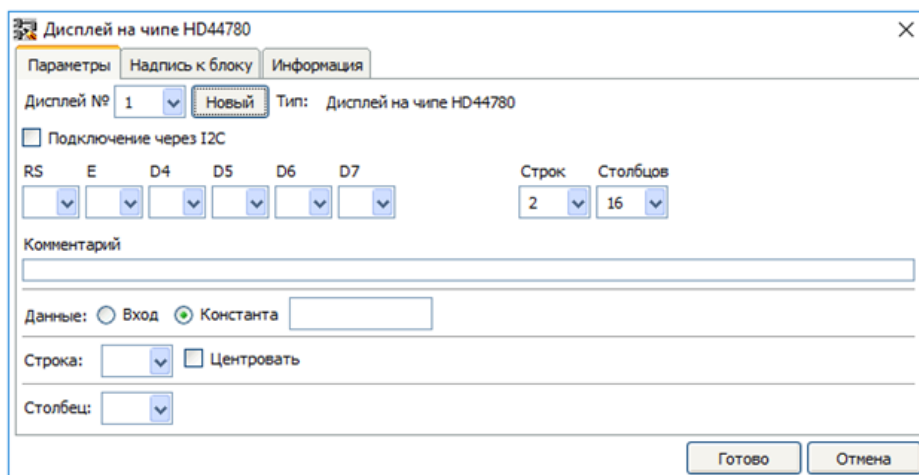
იმისათვის, რომ ავსახოთ გაზომვილი მანძილის სიდიდეები საჭიროა რაიმე დისპლეის გამოყენება. ჩვენ ვიყენებთ ჩვენს ხელთ არსებულ დისპლეის ჩიპზე HD44788. ამ მიზნით პროგრამის ელემენტების ბიბლიოთეკის დისპლეების საქალაღდედან-Дисплей ვირჩევთ დისპლეის ჩიპზე HD4478-Дисплей на чипе HD4478 და გადმოგვაქვს იგი პროგრამის სამუშაო

ზონაში. ამ ბლოკზე ორჯერ დაწკაპუნებით ეკრანზე აისახება მისი პარამეტრიზაციის საწყისი ფანჯარა (ფიგ. 43.4).

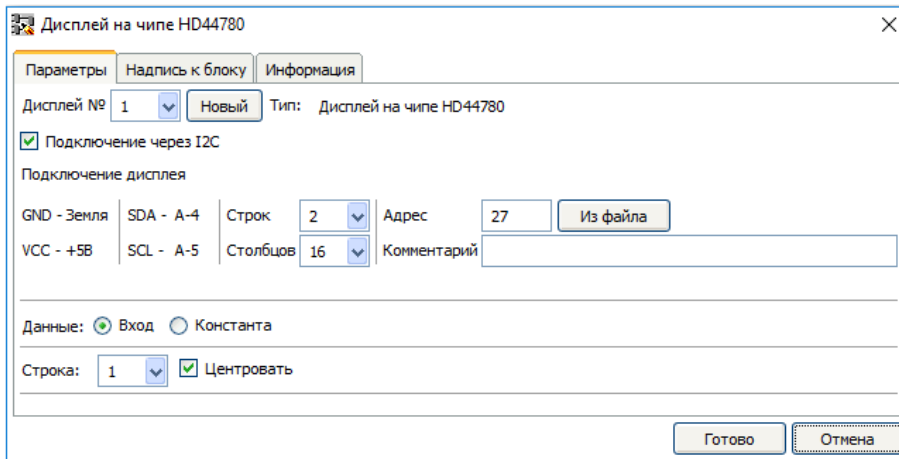


ფიგ. 43.4. დისპლეის ბლოკის პარამეტრიზაციის პირველი ფანჯარა

პარამეტრიზაციის პირველ ფანჯარაში ვაჭერთ ღილაკს ახალი-Новый, რაც გამოიწვევს დისპლეის ბლოკის ახალი ფანჯრის გამოჩენას (ფიგ. 43.5), სადაც დისპლეის ნომერის ველში-Дисплей № ჩაწერილი იქნება ერთიანი ავტომატურად (იგი იქნება პირველი დისპლეის ბლოკი, რადგან ჩვენ გამოვიყენეთ დისპლეი ამ პროექტში პირველად). ამის შემდეგ, რადგანაც ჩვენს ხელთ არსებულ დისპლეის შეუძლია მუშაობა მხოლოდ და მხოლოდ I2C პროტოკოლით, ჩვენ უნდა დავაყენოთ I2C-ს გავლით მიერთების ალამი-Подключение через I2C, რაც გამოიწვევს ახალი ფანჯრის გამოჩენას (ფიგ. 43.5).



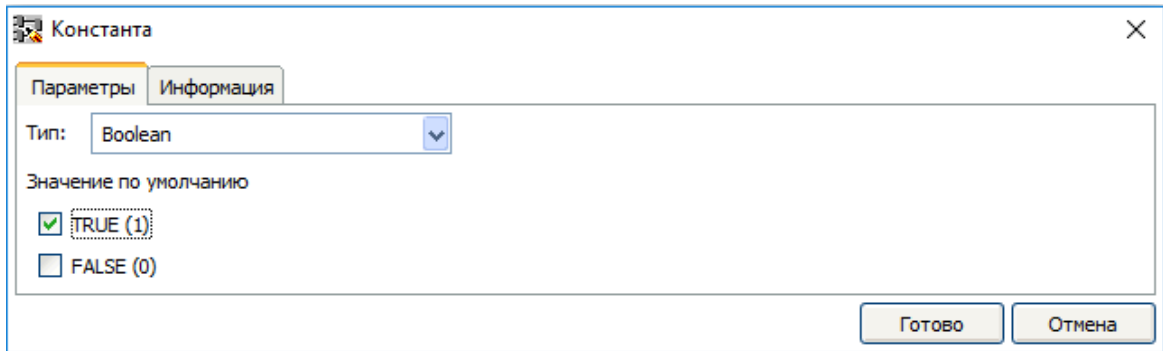
ფიგ. 43.5 დისპლეის ბლოკის პარამეტრიზაციის მეორე ფანჯარა



ფიგ. 43.6 დისპლეის ჩიპზე HD44788 პარამეტრიზაციის ახალი ფანჯარა

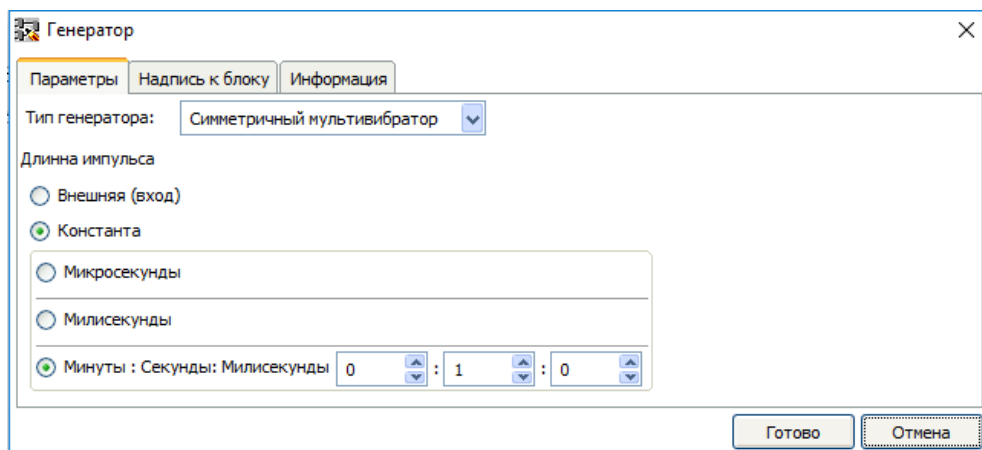
ფიგურა 43.6-ზე გამოსახულ ფანჯარაში სტრიქონის ველში-Строк ჩავწერეთ 2, ხოლო სვეტების ველში-Столбцов 16, ანუ ვწერთ იმ პარამეტრებს, რაც შეესაბამება ჩვენი დისპლეის ტექნიკურ მონაცემებს. მისამართის ველში-Адрес ჩავწერეთ 27 (ეს, საქარხნო მისამართია, რომელიც დადგენილი უნდა იყოს წინასწარ ინტერნეტის რესურსების გამოყენებით). შემდეგ, ველში მონაცემები-Данные: ჩავრთოთ ჩამრთველი შესასვლელი-Вход, სტრიქონის ველში-Строка: ჩავწერეთ ერთიანი, (რადგან ერთი სტრიქონის გამოყენება ამ პროექტში სრულიად საკმარისია), ჩავრთოთ ცენტრში მოთავსების ალამი-Центровать (რათა გაზომვილი მონაცემები განლაგდება სტრიქონის ცენტრალურ ადგილას) და დავაჭიროთ მზადყოფნის ლილაკს-Готово. ამით დისპლეის ბლოკის დაპროგრამება დამთავრებული იქნება. აქ დავგვრჩა კიდეც ორიოდე პროცედურის შესრულება: ჯერ დავაწკაპუნოთ (ფიგ. 43.1) დისპლეის ბლოკის (B3) EN შესასვლელზე თავის მარჯვენა ლილაკით და გამოსულ ფანჯარაში ავირჩიოთ კონსტანტის ჩასმა-Вставить константу და შემდეგ, ახალ გამოსულ ფანჯარაში ჩავრთოთ ალამი TRUE (1), ამის შემდეგ კი დისპლეის ანალოგურ 0 შესასვლელს შევავრთებთ კონვერტაციის ბლოკის (B2) გამოსასვლელზე თავის მარჯვენა ლილაკის გამოყენებით, ხოლო ამ უკანასკნელი ბლოკის შესასვლელს შეუერთებთ ულტრაბგერითი სენსორის (B1) გამოსასვლელზე.

ახლა გადავიდეთ პროექტის მეორე ნაწილის, ანუ რელეს მართვის პროგრამის შექმნაზე. ამ მიზნით FLProg-ის ელემენტების ბიბლიოთეკის-Библиотека элементов ტაიმერების საქაღალდედან-Таймеры პროგრამის სამუშაო ზონაში გადმოვიტანოთ გენერატორის ბლოკი (Generator) (B4 ფიგ. 43.1) და მოვახდინოთ მისი პარამეტრიზაცია. ჯერ დავაწკაპუნოთ თავის მარჯვენა ლილაკით მის შესასვლელზე EN და გამოსულ ფანჯარაში ახლა უკვე თავის მარცხენა ლილაკით ავირჩიოთ კონსტანტის ჩასმა-Вставить константу. შემდეგ გამოსულ ფანჯარაში დავაყენოთ ალამი (TRUE) (ფიგ. 43.7) და დავაწვეთ მზადყოფნის ლილაკს-Готово.



ფიგ. 43.7 გენერატორის პარამეტრიზაციის ფანჯარა

ამის შემდეგ თავის მარცხენა ღილაკით დავაწკაპუნოთ თვითონ გენერატორის ბლოკის შეგნით და გამოსულ ფანჯარაში (ფიგ. 43.8) გენერატორის ტიპის ველში-Тип генератора: ავირჩიოთ სიმეტრიული მულტივიბრატორი-Симметричный мультивибратор, იმპულსის სიგრძის ველში-Длина импульса, ჩავრთოთ კონსტანტის ჩამრთველი-Константа და შემდეგ დროის შერჩევის ველში ჩავწეროთ იმ დროის სიდიდე, რა პერიოდულობითაც გვინდა ჩაირთოს და გამოირთოს რელე. ჩვენ შემთხვევაში ვწერთ 1 წამს-Секунды, რაც ნიშნავს იმას, რომ რელე პროგრამის გაშვების შემდეგ ჩართული იქნება ერთი წამის განმავლობაში და გამორთული იქნება ასევე ერთი წამის განმავლობაში და ასე გაგრძელდება განუწყვეტლივ პროგრამის გაჩერებამდე. ამის შემდეგ ვაწვებით მზადყოფნის ღილაკს-Готово, რითაც გენერატორის პარამეტრიზაციის პროცედურაც დამთავრებული იქნება.

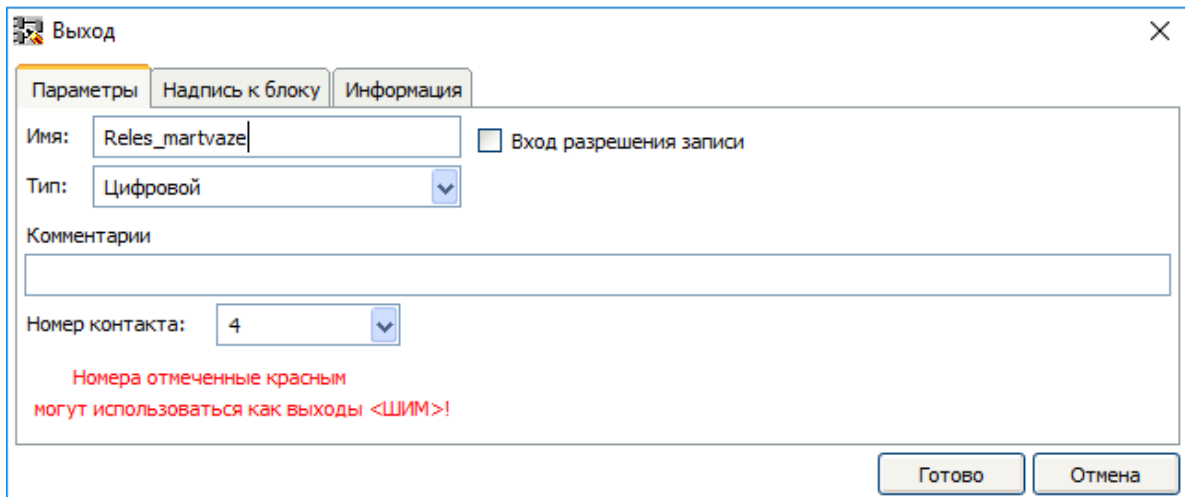


ფიგ. 43.8 გენერატორის პარამეტრიზაციის ფანჯარა

ამის შემდეგ გადავიდეთ პროგრამის ტეგების ზონაში, გავხსნათ გამოსასვლელების საქაღალდე-Выходы, დავაწკაპუნოთ გამოსასვლელის დამატებას-Добавить выход და გამოსულ

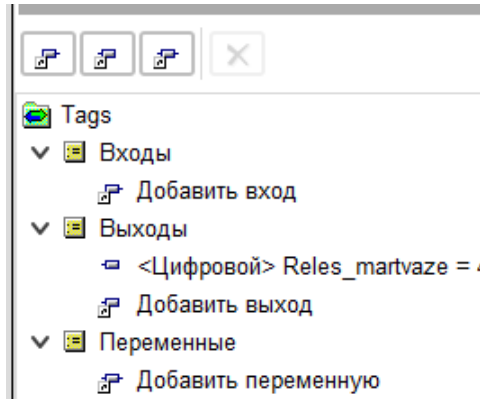
ფანჯარაში (ფიგ. 43.9) სახელის ველში-Имя: ჩავწერთ გამოსასვლელის ჩვენთვის სასურველი დასახელება, (ჩვენს პროექტში ვწერთ Reles_martvaze, ტიპის ველში-Тип: ავირჩიოთ ციფრული-Цифровой, კონტაქტის ნომრად ავირჩიოთ ის ნომერი რასთანაც უნდა იქნას მიერთებული რელეს მართვის კონტაქტი (მე-4 ნომერი ჩვენს შემთხვევაში) და დავაჭიროთ მზადყოფნის ღილაკს-Готово და ამით გამოსასვლელის დაპროგრამებაც დამთავრებული იქნება.

ამის შემდეგ საჭიროაა ეს ახლად შექმნილი გამოსასვლელი თავის დახმარებით ტეგების ზონიდან გადმოვიტანოთ პროგრამის სამუშაო ზონაში და მისი შესასვლელი შევუერთოთ გენერატორის გამოსასვლელს (ფიგ. 43.1). არდუინოს მე-4 გამოსასვლელი შევუერთოთ რელეს მოდულის (5v Relay Module) მართვის შესასვლელს (S), ხოლო ამავე მოდულის G და V შესასვლელები შევუერთოთ არდუინოს მიწისა (GND) და 5 ვოლტის (5v) გამომყვანებს შესაბამისად.

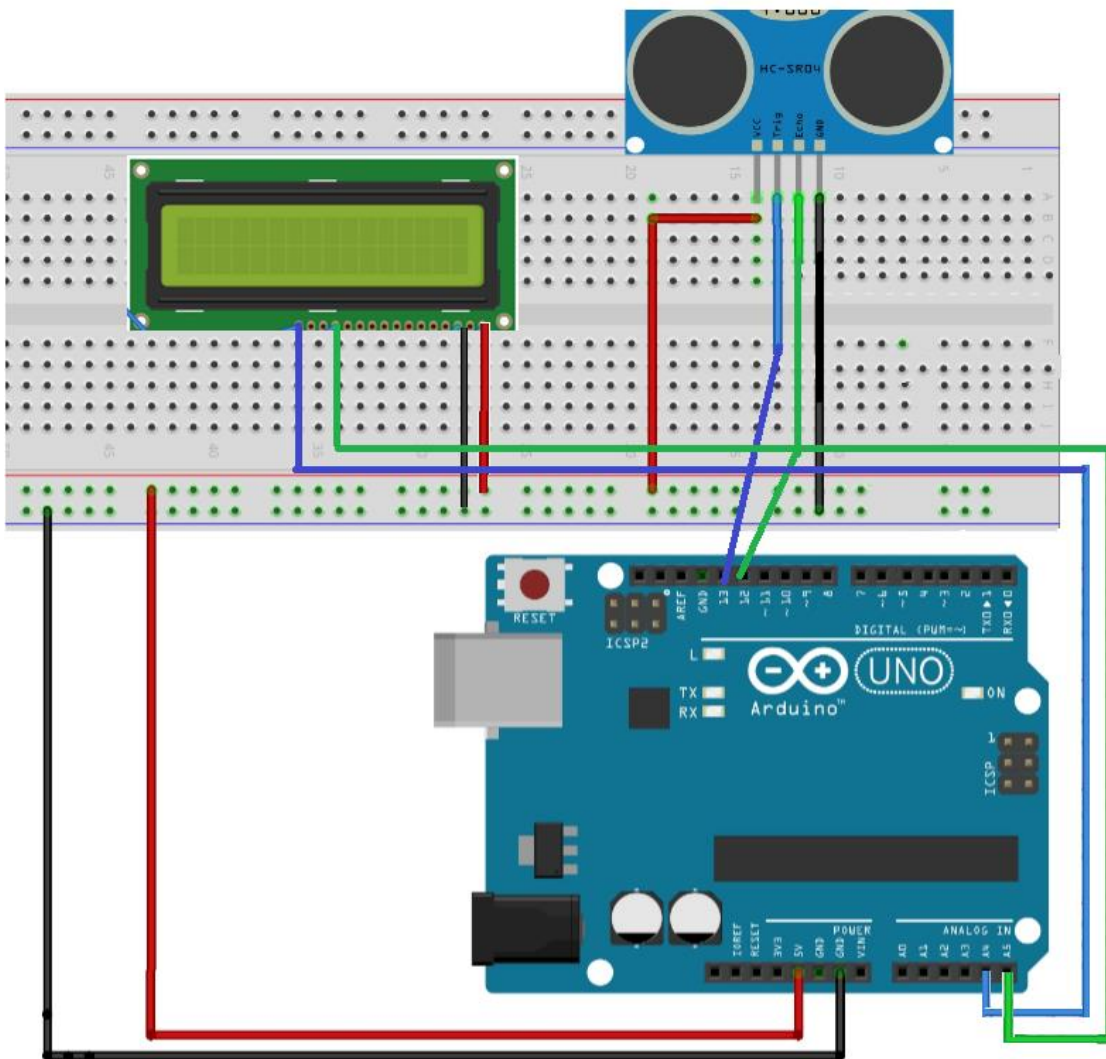


ფიგ. 43.9 გამოსასვლელის პარამეტრიზაციის ფანჯარა

ყოველივე ამ საქმიანობის შემდეგ გავუშვათ პროგრამა და მისი კორექტულობის შემთხვევაში ჩვენ გავიგონებთ რელეს კონტაქტების ჩართვა/გამორთვების ხმას ერთი წამის პერიოდულობით და ასევე დავინახავთ რელეს მოდულზე შუქდიოდის ციმციმს იმავე პერიოდულობით. ფიგ. 43.10-ზე წარმოდგენილია პროექტის ტეგების ზონის ფანჯარა, ხოლო ფიგ. 43.11-ზე წარმოდგენილია პროექტის Fritzing სქემა.



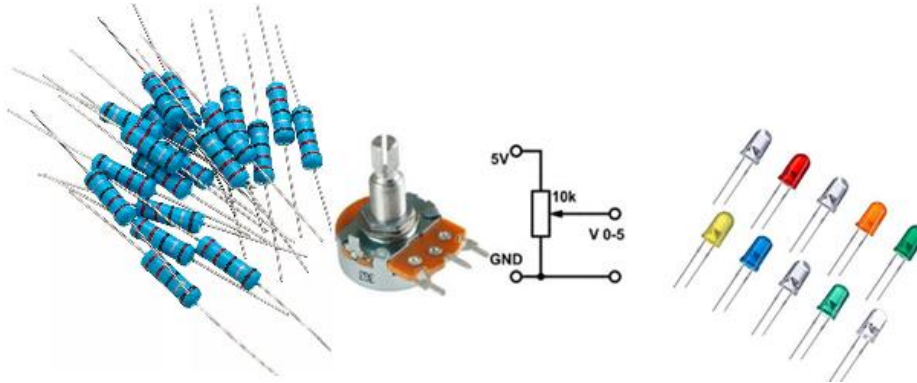
ფიგ. 43.10 პროექტის ტეგების ზონის ფანჯარა



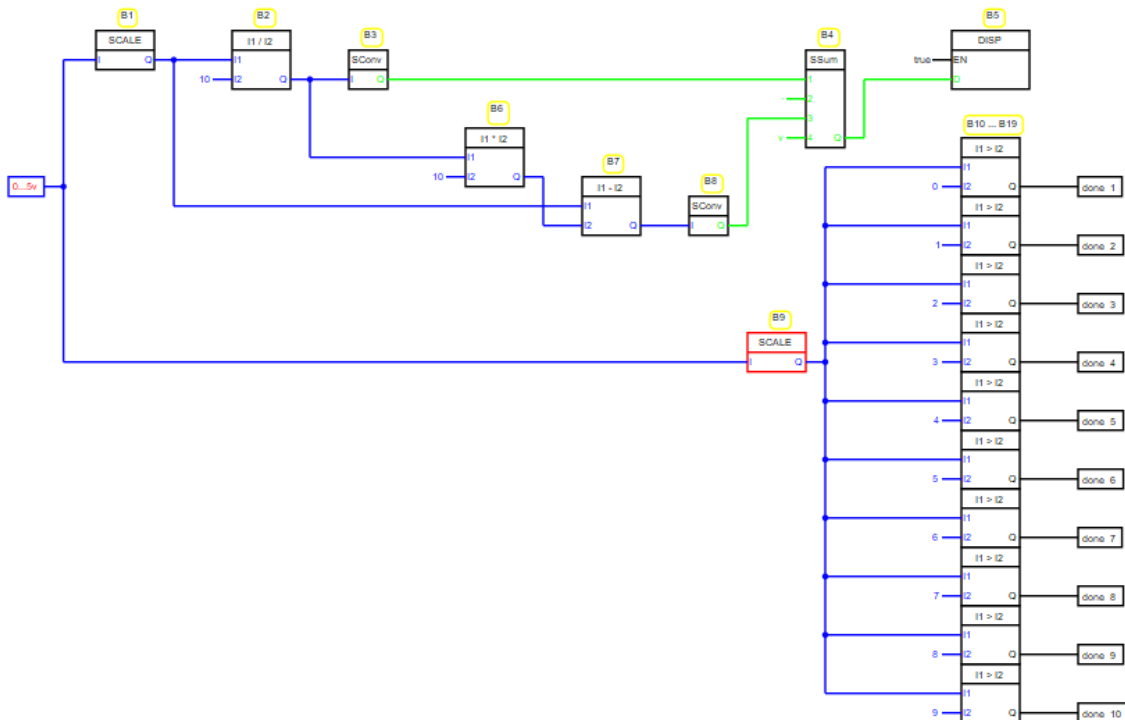
ფიგ. 43.11 პროექტის Fritzing სქემა

პროექტი_44

5 ვოლტის გაზომვა დონის ინდიკატორით



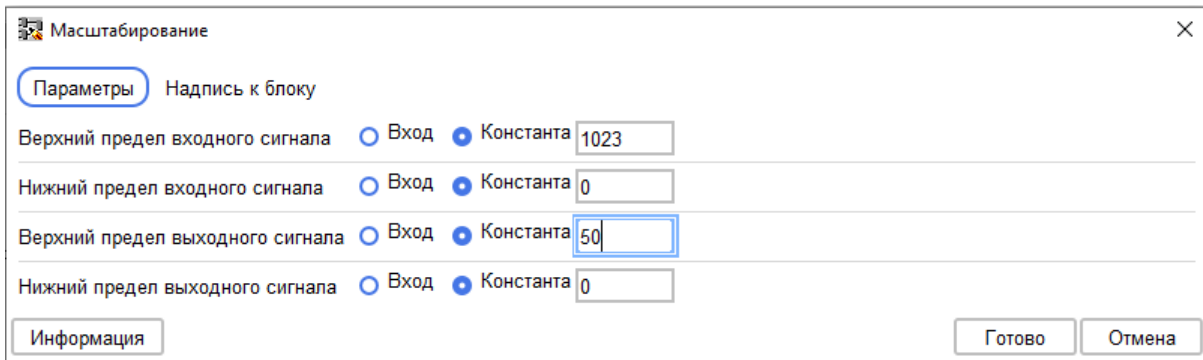
გავუშვით პროგრამა FLProg, დაპროგრამების ენად ავიჩიოთ FBD ხოლო კონტროლერის ტიპად Arduino UNO. პროგრამის ტეგების ზონაში შევქმნათ ერთი ანალოგური შესასვლელი არდუინოს A0 კონტაქტის გამოყენებით (0...5v), რომელზედაც უნდა მიუერთოთ პოტენციომეტრის შუალედური გამომყვანი, ხოლო მისივე განაპირა გამომყვანები უნდა შეუერთოთ არდუინოს 5 ვოლტისა და მიწის GND კონტაქტებს. ჩამოვიტანოთ ეს შესასვლელი პროგრამის სამუშაო არეში - ბლოკი (0...5v) (ფიგ. 44.1).



ფიგ. 44.1 5 ვოლტის გაზომვისა და ძაბვის დონის ინდიკატორის პროგრამა

პოტენციომეტრის ცოცხას გადაადგილების დროს, როდესაც მის განაპირა გამომყვანებზე მოდებულია + 5 ვოლტი ძაბვა, მაშინ მის შუალედურ გამომყვანზე ძაბვა შეიცვლება 0 ... 5ვ-ის დიაპაზონში. ამ ძაბვას არდუინო გარდაქმნის ათთანრიგა ორობითი

კოდის სიზუსტით ანუ ათობით კოში 0 ... 1023-მდე შესაბამისად. ეს გაზომილი სიდიდე მიეწოდება B1 ბლოკის შესასვლელზე, რომელმაც უნდა გააკეთოს გაზომილი სიდიდის მასშტაბირება ისეთნაირად, რომ შესაბამისი მათემატიკური გარდაქმნების შედეგად მივიღოთ გაზომილი ძაბვის მნიშვნელობა ათობით კოდში. ამიტომ, გადავიდეთ პროგრამის ბლოკების ბიბლიოთეკაში და სამუშაო ზონაში გადმოვიტანოთ მასშტაბირების ბლოკი B1 და დავაპროგრამოთ მისი დიაპაზონები ისე, როგორც ეს ფიგურა 44.2-ზეა ნაჩვენები.



ფიგ. 44.2 B1 მასშტაბირების ბლოკის დაპროგრამება

რადგან ჩვენ გაზომვა უნდა ვაწარმოოთ მეათედი სიზუსტით, ამიტომ როდესაც გასაზომი ძაბვა იცვლება 0-დან 5 ვოლტამდე და B1 ბლოკის შესასვლელი იცვლება 0-დან 1023-მდე, მაშინ მისი გამოსასვლელი უნდა იცვლებოდეს 0-დან 50-მდე შესაბამისად. ახლა ჩვენ ჯერ უნდა გამოვიანგარიშოთ გასაზომი ძაბვის მთელი მნიშვნელობა. ამისათვის B1 ბლოკის გამოსასვლელზე არსებული სიდიდე უნდა გავყოთ 10-ზე - B2 არითმეტიკული გაყოფის ბლოკის დახმარებით და რადგან ამ ბლოკის თვისება ისაა, რომ იგი გაყოფას აწარმოებს ნაშთის გადაგდებით, ამიტომ, მის გამოსასვლელზე დაფიქსირდება გასაზომი ძაბვის რიცხვითი მნიშვნელობის მთელი ნაწილი მძიმედ. ახლა უნდა გამოვითვალოთ გასაზომი ძაბვის მნიშვნელობა მძიმის შემდეგ, ანუ ათობითი სიდიდე. ამისთვის B2 ბლოკის გამოსასვლელის რიცხვითი მნიშვნელობა ისევ უნდა უნდა გავამრავლოთ 10-ზე და მიღებული მნიშვნელობა გამოვაკლოთ B1 ბლოკის მნიშვნელობისგან (მისი მნიშვნელობა მოიცავს ძაბვის როგორც მთელ ისე ათწილადის მნიშვნელობას). ათზე გამრავლება ხდება B6 ბლოკის გამოყენებით, ხოლო გამოკლება-B7 ბლოკის გამოყენებით. შედეგად მივიღებთ იმ რიცხვით მნიშვნელობას, რომლის გადაგდებაც მოხდა გაყოფის ოპერაციის შესრულების დროს B2 ბლოკში, ანუ გამოკლების ოპერაციის შესრულების შედეგად მივიღებთ ძაბვის ათწილადის რიცხვით მნიშვნელობას (მძიმის შემდეგ).

მიღებული გაზომვილი მნიშვნელობის ასახვისათვის ვიყენებთ დისპლეის ჩიპზე HD44780-ს. ავსახოთ ეს ინფორმაცია პირველ სტრიქონის ცენტრში, ასეთი ფორმატი: მთელი რიცხვი, წერტილი, წილადი რიცხვი, V (განზომილება. ამ რიცხვების ასახვისათვის საჭიროა გარდავქმნათ მთელი და ათწილადი რიცხვები სტრიქონებათ. ამ მიზნით ვიყენებთ B3 და B8 ბლოკებს. ახლა საჭიროა შევკრიბოთ ეს სტრიქონები, მაგრამ შუაში უნდა ჩავუმატოთ მთელისა და ათწილადის გამყოფი წერტილი და ბოლოში მივამატოთ განზომილება V. ამ ფუნქციის შესასრულებლად ვიყენებთ სტრიქონების შეკრების ბლოკს B4, რომელსაც ვიღებთ ბლოკების ბიბლიოთეკის სტრიქონის საქალაქედან-Строки სტრიქონების შეკრების-Сложение строк ბლოკს. შესასვლელების რაოდენობას მივუთითებთ 4-ს (ფიგ. 44.3). ამ ბლოკის პირველ შესასვლელს შეუერთებთ B3 ბლოკის გამოსასვლელს და ეს იქნება გაზომვილი ძაბვის მთელი ნაწილი, მეორე შესასვლელზე ჩავწერთ კონსტანტის მნიშვნელობას წერტილს, მესამეზე მივუერთებთ ბლოკ B8-ის გამოსასვლელს და ეს იქნება გაზომვილი ძაბვის ათწილადის რიცხვითი მნიშვნელობა, ხოლო მეოთხე შესასვლელზე ჩავწერთ კონსტანტას V ეს იქნება გაზომვილი სიდიდის განზომილება. ბლოკი B4 უზრუნველყოფს ამ ოთხი ჩანაწერის შეკრებას სტრიქონულ ფორმატში, რაც გადაეცემა დისპლეის ბლოკს B5, რომელიც ასახავს კიდეც ამ გასაზომი ძაბვის მნიშვნელობას.

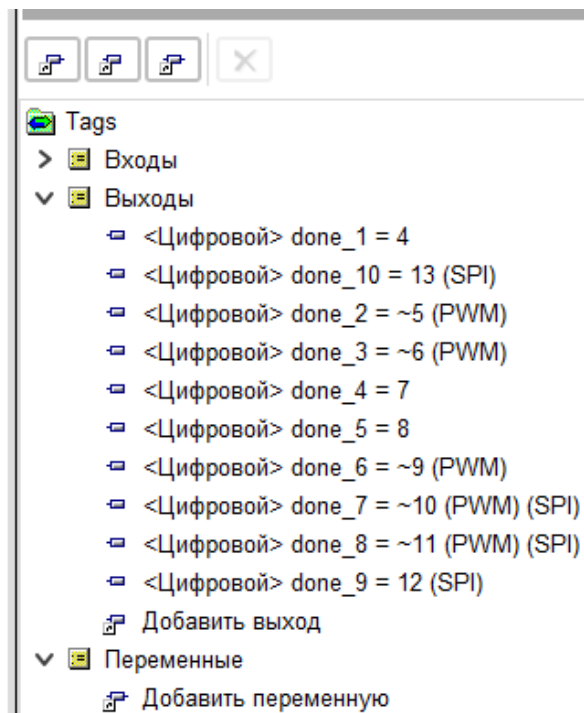


ფიგ. 44.3 სტრიქონთა შეკრების ბლოკის პარამეტრიზაცია

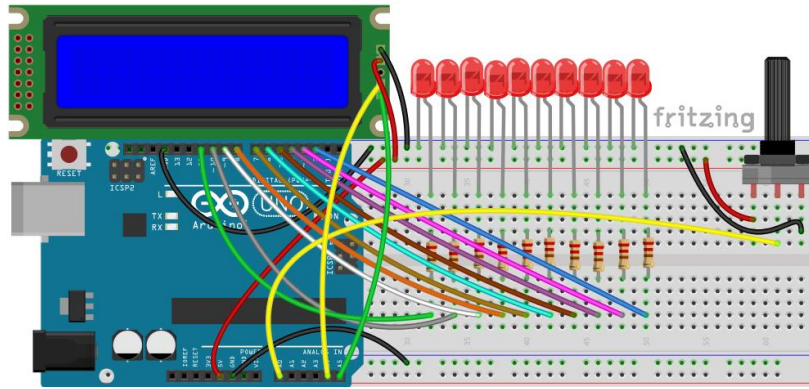
ახლა განვიხილოთ დონის ინდიკატორის შექმნა და მისი მუშაობის პრინციპი. იდეა მდგომარეობს იმაში, რომ აღებულია მწკრივად დალაგებული შუქდიოდების ნაკრები 10 ცალის რაოდენობით. პოტენციომეტრის ბრუნვის შედეგად, როდესაც მის შუალედურ გამოსასვლელზე ძაბვაა 0...1 ვოლტი, მაშინ უნდა აინთოს პირველი შუქდიოდი, როდესაც ძაბვა გაუტოლდება 1 ... 2 ვოლტს, მაშინ უნდა აინთოს პირველი და მეორე შუქდიოდი და ასე შემდეგ, როდესაც ძაბვა გაუტოლდება 4 ... 5 ვოლტს, მაშინ უნდა აინთოს ყველა შუქდიოდი. ე.ი. ძაბვის ცვლილება გამოსახული იქნება ანთებული შუქდიოდის რაოდენობით ანუ დონის ცვლილებით.

ამ ალგორითმის რეალიზაციისათვის ვიყენებთ კიდევ ერთ მასშტაბირების ბლოკს B9 (ფიგ. 44.1), რომლის შესასვლელი სიგნალის ზედა და ქვედა ზღვრებს ვტოვებთ უცვლელად, ხოლო გამოსასვლელი სიგნალის ზედა ზღვარში ვწერთ 10 (შუქდიოდების რაოდენობის მიხედვით) ხოლო ქვედა ზღვარში - ნულს. ამის შემდეგ ბლოკების ბიბლიოთეკიდან Сравнения-ს საქალაღდედან პროგრამის სამუშაო ზონაში გადმოგვაქვს კომპარატორის ბლოკი - Comparator, ვაპროგრამირებთ მას მეტობაზე >, შემდეგ ვაკოპირებთ ამ ბლოკს და ჩავსვავთ პროგრამაში კიდევ ცხრა ასეთ დაკოპირებელ ბლოკს B10 ... B19, შევავრთეთ ამ ბლოკების I1 შესასვლელებს B9 ბლოკის გამოსასვლელზე, ხოლო I2 შესასვლელებზე მივუთითებთ მთელ რიცხვს-Integer და ჩავწერთ თანმიმდევრობით 0,1,2,3,4,5,6,7,8,9, როგორც ეს ფიგ. 44.1-ზეა წარმოდგენილი.

ამის შემდეგ გადავდივართ ტეგების ზონაში და შევქმნით 10 გამოსასვლელს: done_1, done_2 ... done_10, რისთვისაც გამოვიყენებთ არდუინოს ციფრულ კონტაქტებს 4-13 ჩამოვიტანთ ამ გამოსასვლელებს სამუშაო ზონაში B10 ... B19 და შევავრთებთ კომპარატორის ბლოკებს ისე, როგორც ეს ნაჩვენებია მახაზზე (ფიგ. 44.1). ამის შემდეგ ავაწყობთ პროექტს, რომლის ტეგების ზონის ფანჯარა წარმოდგენილია ფიგ. 44.4-ზე, ხოლო fritzing სქემა-ფიგ. 44.5-ზე.



ფიგ. 44.4 პროექტის ტეგების ზონის ფანჯარა



ფიგ. 44.5 5 ვოლტის გამზომისა და ძაბვის დონის ინდიკატორის Fritzing სქემა

პროექტი_45

ობიექტების შორიდან მართვა ინფრაწითელი პულტით



ამ პროექტში ობიექტების შორიდან მართვის პულტად გამოყენებულია 21 ლილაკიანი YK-001 ტიპის პულტი, რომელსაც შეუძლია ინფრაწითელი სხივების გამოსხივება, დაჭერილი ლილაკის ნომრის შესაბამისი კოდის სახით. ეს კოდები ჩაწერილია პულტში ქარხანაში დამზადების პროცესში და მისი ცვლილება შეუძლებელია. პულტიდან გამოსული სხივები მიიღება 38 კჰც-იანი VS1838B მიმღები მოდულის მიერ, ხოლო ამ უკანასკნელიდან მიღებული კოდების გაშიფრა ხდება FLProg-ის IRReSSive ბლოკის მიერ მისი ბლოკების ბიბლიოთეკის ინფრაწითელი მართვის-ИК Управление-ს საქალაღედან პროგრამის სამუშაო ველში გადმოტანისა და გამოყენების შედეგად. ამ ბლოკის დანიშნულება არის პროექტში ინფრაწითელი პულტიდან სიგნალების მიღების რეალიზაციის შესაძლებლობა. ბლოკის აწყობებში შესაძლებელია საჭირო რაოდენობის გამოსასვლელების არჩევა, რომლებზედაც პულტის რომელიმე გარკვეულ ლილაკზე დაჭერისას გამომუშავდება მაღალი დონე.

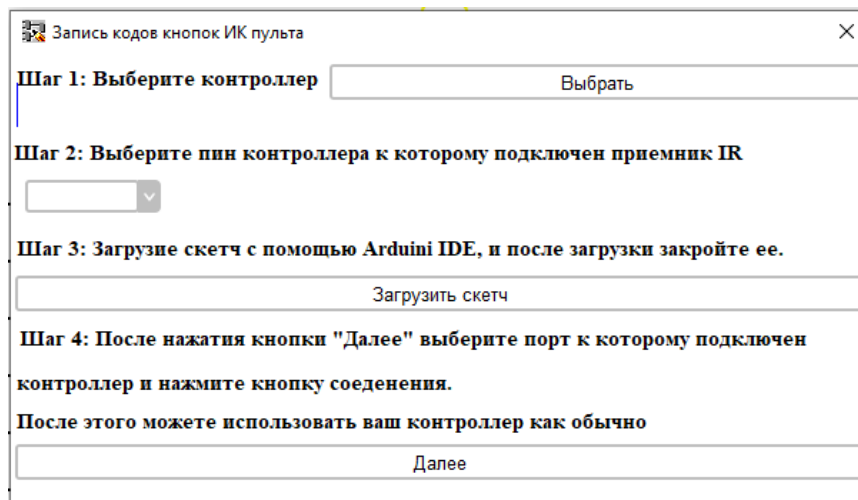
პულტის ლილაკების კოდების დასადენათ საჭიროა ჯერ შევიდეთ FLProg-ის მენიუს ინსტრუმენტებში - Инструменты და ჩამოშლილი სიიდან ავირჩიოთ ინფრაწითელი პულტის

ლილაკების კოდების ჩაწერა-Запись кодов кнопок ИК пульта, გამოსულ ფანჯარაში (ფიგ. 45.1) მორიგეობით შევსრულოთ ოთხივე საფეხური:

ა. ავირჩიოთ კონტროლერი-Выбрать контроллер, ჩვენ შემთხვევაში ვირჩევთ ჩვენს ხელთ არსებულ Arduino Uno-ს.

ბ. ავირჩიოთ კონტროლერის შემყვან/გამომყვანი კონტაქტი (პინი), რომელზედაც მიერთებულია ინფრაწითელი მიმღები-Выберите пин контроллера к которому подключен приемник IR, ჩვენ შემთხვევაში ვირჩევთ მე-5 კონტაქტს.

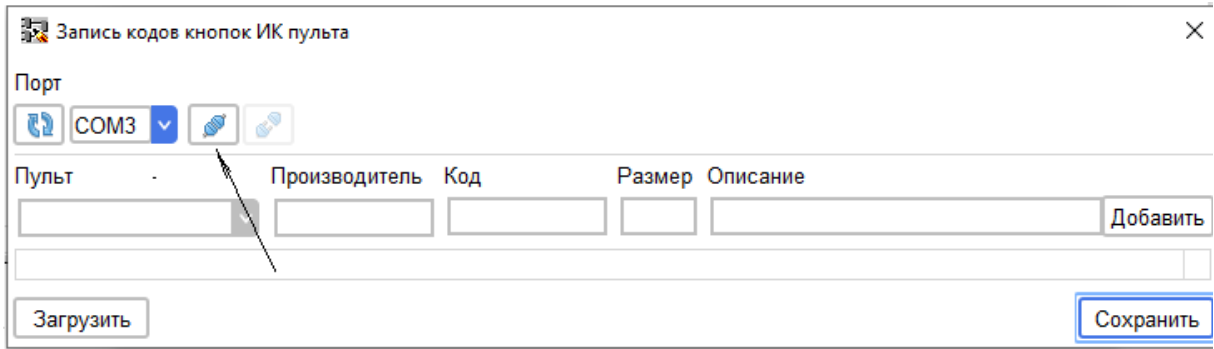
გ. ჩატვირთოთ სკეჩი Arduino IDE-ს გამოყენებით და ჩატვირთვის შემდეგ დავხუროთ იგი-Загрузите скетч с помощью Arduino IDE, и после загрузки закройте ее.



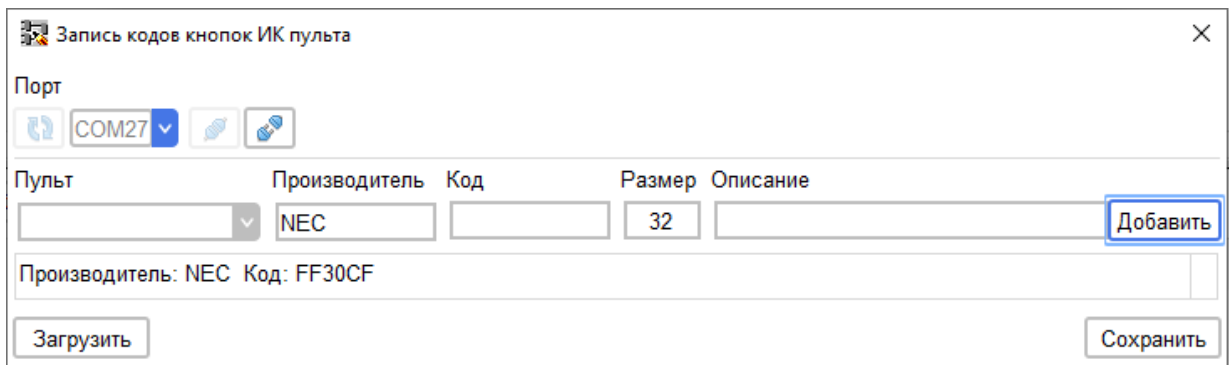
ფიგ. 45.1 ინფრაწითელი პულტის ლილაკების კოდების მიღების თანმიმდევრობა

დ. ლილაკზე „შემდეგ“-«Далее» დაჭერის შემდეგ ავირჩიოთ ის ვირტუალური პორტი, რომელზედაც მიერთებულია კონტროლერი (ჩვენს შემთხვევაში-27) იხილეთ ფიგ. 45.2 და დავაჭიროთ შეერთების ლილაკს-После нажатия кнопки «Далее» выберите порт к которому подключен контроллер и нажмите кнопку соединения.

ლილაკზე შემდეგ-Далее დაჭერის შემდეგ ეკრანზე გამოვა ფანჯარა (ფიგ. 45.2), სადაც COM3 პორტის ნაცვლად უნდა ჩავწეროთ იმ ვირტუალური პორტის ნომერი რომელიც სისტემამ მიაწიჭა არდუინოს (27 ჩვენს შემთხვევაში) და შემდეგ დავაჭიროთ შეერთების ლილაკს თითი (ნაჩვენებია ისრით). ამის შემდეგ ეკრანზე გამოვა უკვე ინფრაწითელი პულტის ლილაკების შესაბამისი კოდების განსაზღვრის ფანჯარა (ფიგ. 45.3).

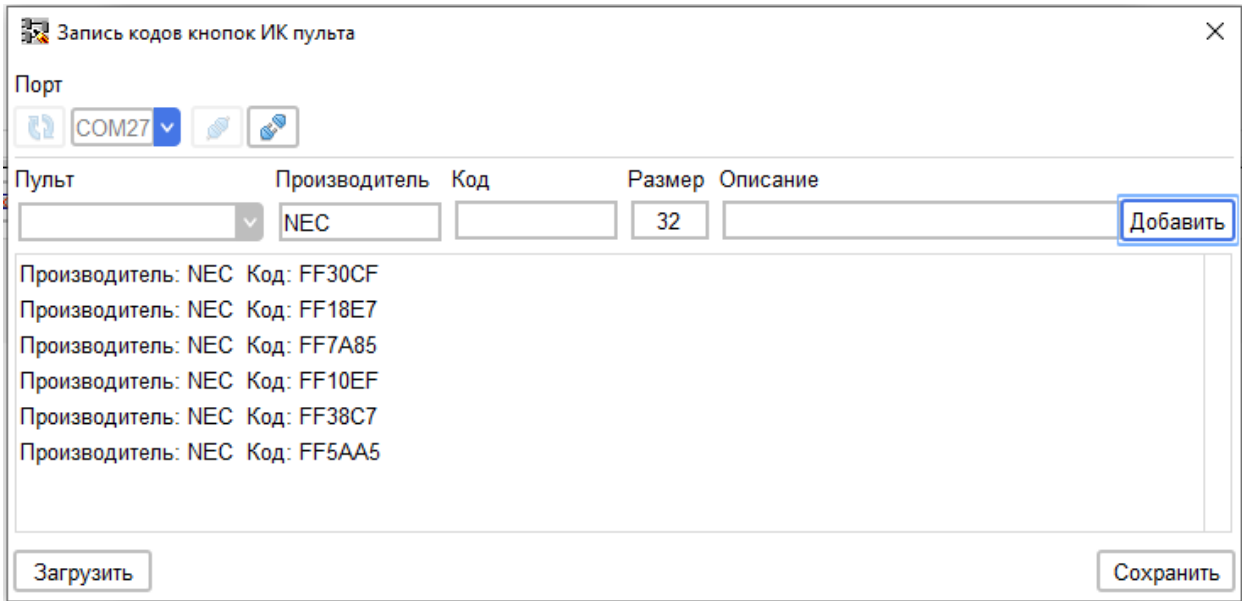


ფიგ. 45.2 ინფრაწითელი პულტის ღილაკების კოდების ჩაწერის ფანჯარა



ფიგ. 45.3 ინფრაწითელი პულტის ღილაკების კოდების პარამეტრიზაციის ფანჯარა

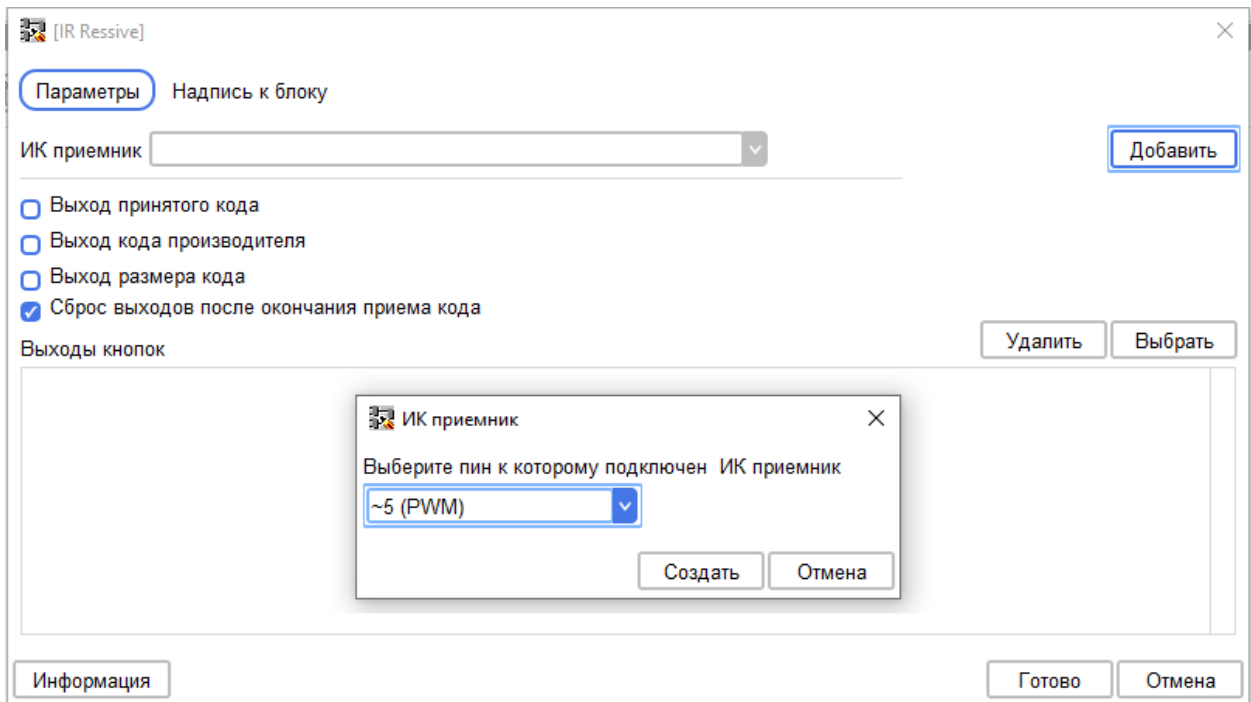
ამის შემდეგ ვიწყებთ ჩვენთვის საჭირო ღილაკების კოდების განსაზღვრას. პულტს ვუმიზნებთ ინფრაწითელი სხივების მიმღებ მოდულს, ვაწვევით რომელიმე ღილაკს პულტზე (მაგ. 1) და კომპიუტერის ეკრანზე აისახება ღილაკის კოდი, საიმედოობისათვის ამ პროცედურას გავიმეორებთ ორჯერ-სამჯერ და თუ ეს კოდი არ შეიცვალა ვაწვევით დამატების ღილაკს ეკრანზე-Добавить და ეკრანზე გამოსახული ფანჯარა მიიღებს სახეს (იხ. ფიგ. 45.3). აქ აისახება როგორც ამ პულტის მწარმოებელი ფირმა-Производитель NEC ისე პირველი ღილაკის შესაბამისი კოდი FF30CF. შემდეგ ვაწვევით პულტის მეორე ღილაკს და დამატების ღილაკს-Добавить და ასე შემდეგ, ჩვენს შემთხვევაში ყველა შერჩეული ღილაკებისათვის 1,2,3,4,5,6 გამოვა შესაბამისი კოდები (ფიგ. 45.4).



ფიგ. 10.4 ინფრაწითელი პულტის ღილაკების 1,2,3,4,5,6 შესაბამისი კოდები

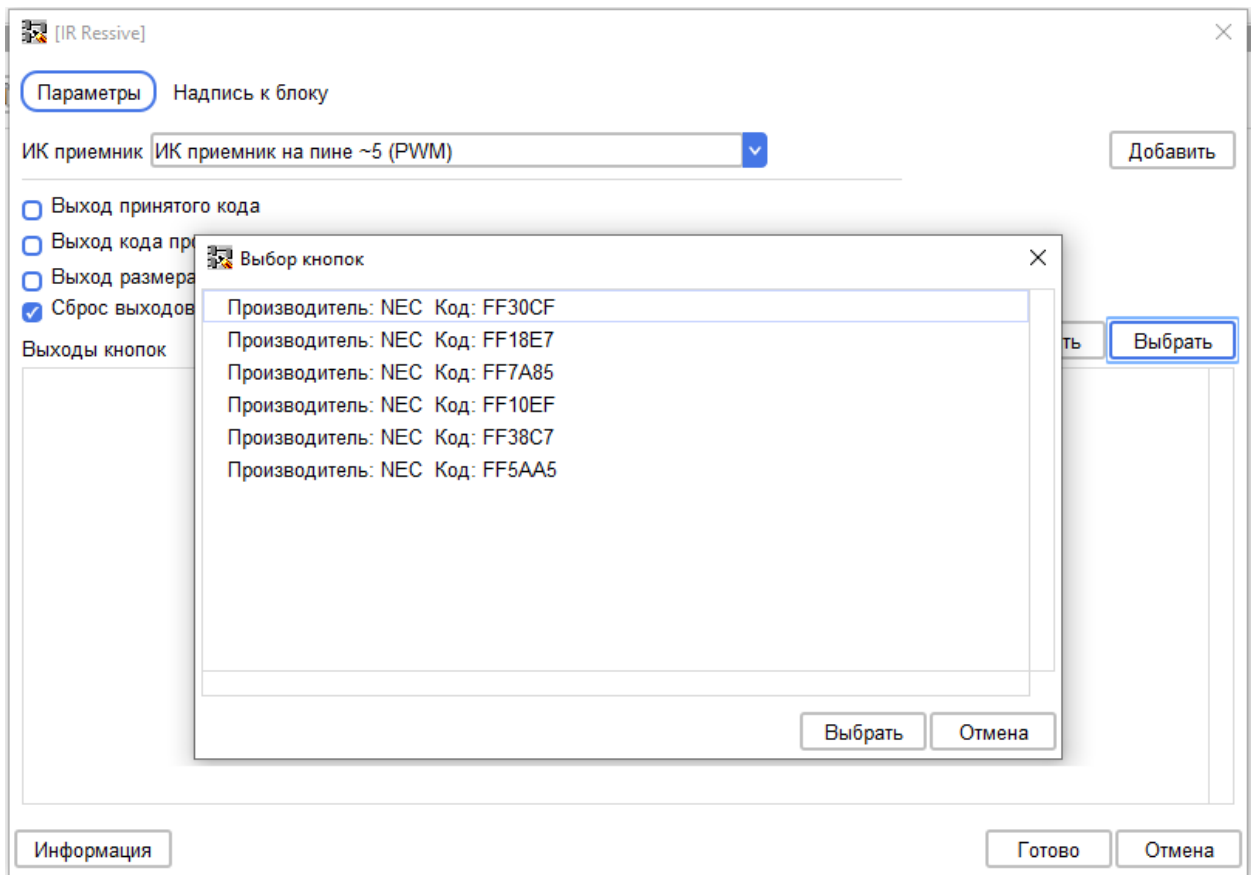
ამის შემდეგ უნდა შევწყვიტოთ COM პორტთან კავშირის სეანსი ფიგ. 45.2-ზე ისრით ნაჩვენები ღილაკის მარჯვნივ მოთავსებულ ღილაკზე დაჭერით, უნდა შევინახოთ მიღებული კოდების ცხრილი ფაილის სახით. ამ მიზნით ვაჭერთ შენახვის ღილაკს-Сохранить ვარქმევთ ფაილს რაიმე სახელს, ვუთითებთ დირექტორიას და ვინახავთ. ამის შემდეგ ჩვენ შეგვიძლია გამოვიყენოთ ჩვენი კონტროლერი ჩვეულებრივად.

მოვახდინოთ IRResive ბლოკის პარამეტრიზაცია, გადმოვიტანოთ (თუ უკვე გადმოტანილი არა გვაქვს) ეს ბლოკი პროგრამის ელემენტების ბიბლიოთეკის ინფრაწითელი მართვის საქალაქდედან-ИК Управление-ს პროგრამის სამუშაო ველში. დავაწკაპუნოთ მასზედ ორჯერ, რის შედეგადაც ეკრანზე გამოვა ფანჯარა სახელწოდებით IR Resive (ფიგ. 45.5).



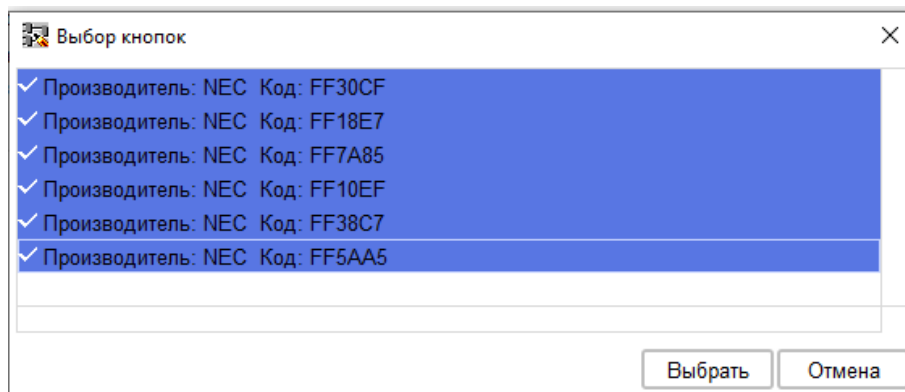
ფიგ. 45.5 IRRessive ბლოკის პარამეტრიზაციის ფანჯრები

დამატების ღილაკზე-Добавить დაწკაპუნებით ამ ფანჯრის შიგნით გამოვა მეორე უფრო მცირე ზომის ფანჯარა ინფრაწითელი მიმღები-ИК приемник, სადაც ჩამოშლითი სიიდან ვირჩევთ არდუინოს იმ კონტაქტს რომელზედაც უნდა მიუერთდეს ინფრაწითელი მოდული (მე-5 კონტაქტი ჩვენს შემთხვევაში), ვაჭერთ შექმნის ღილაკს-Создать და ეს ჩვენი არჩეული ნომერი გადაიწერება ИК приемник-ის ველში. ამის შემდეგ ამავე ფანჯარაში უნდა დავაჭიროთ არჩევის ღილაკს-Выбрать, რის შემდეგაც ფანჯარა მიიღებს ასეთ სახეს (ფიგ. 45.6).



ფიგ. 10.6 ინფრაწითელი მიმღების პარამეტრიზაციის ფანჯარა

აქ უკვე უნდა მოვნიშნოთ ის კოდები, რომლებიც აუცილებლად საჭიროა გამოსაყენებლად (ჩვენს შემთხვევაში მოვნიშნავთ ექვსივე კოდს) (ფიგ. 45.7), ვაჭერთ ამ ფანჯარაში არჩევის ლილაკს-**Выбрать**, შემდეგ წინა ფანჯარაში ლილაკს **Готово** და ამის შემდეგ ჩვენი ეს ბლოკი IR Ressive მიიღებს სახეს (ფიგ. 45.8).



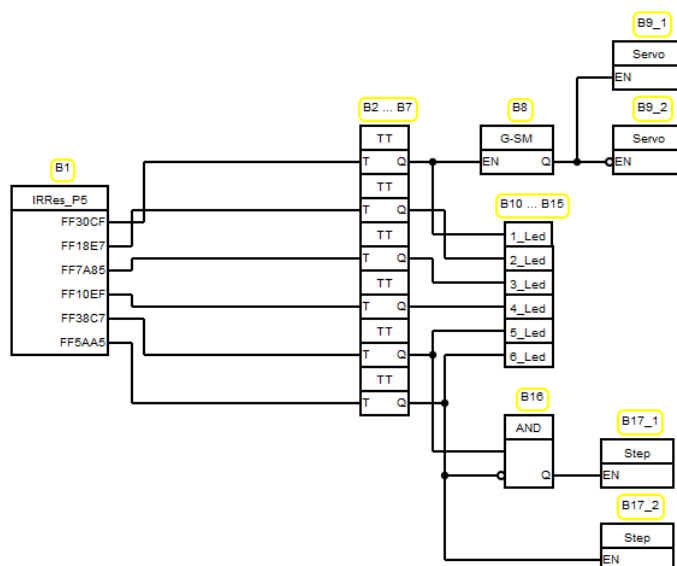
ფიგ. 45.7 IR Ressive ბლოკის კოდების მონიშვნის ფანჯარა

IRRes_P5	
FF30CF
FF18E7
FF7A85
FF10EF
FF38C7
FF5AA5

ფიგ. 10.8 IRResive ბლოკის სახე პარამეტრიზაციის შემდეგ

ახლა ჩამოვყალიბოთ ალგორითმი თუ რა უნდა გააკეთოს კონტროლერმა ამ პულტის ღილაკებზე დაჭერის შემდეგ:

- ა. ნებისმიერ ღილაკზე დაჭერით უნდა აინთოს შესაბამისი შუქდიოდი არდუინოს სამონტაჟო დაფაზე რომელთა ჩაქრობა უნდა მოხდეს შესაბამის ღილაკზე განმეორებით დაჭერის შემდეგ;
- ბ. პირველ ღილაკზე დაჭერამ უნდა გამოიწვიოს ასევე სერვომძრავის ამუშავება და მისი ღერძის ბრუნვა 0-180 გრადუსის ფარგლებში და ორი წამის სიხშირით, მისი გაჩერება კი უნდა მოხდეს ამავე ღილაკზე ხელმეორედ დაჭერით;
- გ. მე-5 ღილაკზე დაჭერით უნდა ამუშავდეს ასევე ბიჭური ძრავი, რომელმაც უნდა იმოძრაოს საათის ისრის მიმართულებით, ამის შემდეგ მე-6 ღილაკზე დაჭერით ბიჯურმა ძრავმა უნდა შეიცვალოს მოძრაობის მიმართულება საათის ისრის საწინააღმდეგოთ;
- დ. და ბოლოს ბიჯური ძრავის გაჩერება უნდა მოხდეს როგორც მე-5, ასევე მე-6 ღილაკზე განმეორებით თითის დაჭერით.



ფიგ. 45.9 ალგორითმის პროგრამული რეალიზაცია FLProg-ში

აღნიშნული ალგორითმის პროგრამული რეალიზაცია ასახულია ფიგ. 45.9-ზე. მასში დასმული ამოცანის გადასაწყვეტად FLProg-ის ბლოკების ბიბლიოთეკის ტრიგერების-საქადალდედან-Триггеры პროგრამის სამუშაო არეში გადმოგვაქვს ექვსი ცალი TT ტრიგერი (B2 ... B7) და ვაერთებთ IRResive B1 ბლოკის გამოსასვლელებზე. როგორც ცნობილია, ამ ტრიგერების მოქმედების პრინციპი მდგომარეობს იმაში, რომ იგი შეიძლება იმყოფებოდეს ორ მდგომარეობაში: ლოგიკური ერთიანის მდგომარეობაში-როცა მის გამოსასვლელებზე მაღალი პოტენციალია (+5ვ) და ლოგიკური ნოლის მდგომარეობაში-როცა მის გამოსასვლელებზე დაბალი პოტენციალია (0ვ). როდესაც ასეთი ტრიგერის შესასვლელებზე მიეწოდება იმპულსების სერია, მაშინ იმპულსის დადებითი ფრონტის შემთხვევაში (იმპულსის გადასვლა ლოგიკური ნოლიდან ლოგიკურ ერთიანში), ტრიგერი იცვლის მდგომარეობას, ანუ თუ იყო ლოგიკური ნოლის მდგომარეობაში მაშინ გადავა ლოგიკური ერთის მდგომარეობაში, ხოლო თუ იყო ლოგიკური ერთის მდგომარეობაში მაშინ გადავა ლოგიკური ნოლის მდგომარეობაში. გარდა ამისა ტეგების არეში უნდა შევქმნათ ექვსი გამოსასვლელი, ჩამოვიტანოთ სამუშაო ზონაში და მიუერთოთ ამ ტრიგერების გამოსასვლელებზე (B10 ... B15), რაზედაც შემდეგ შევაერთებთ ექვს შუქდიოდს (1_Led ... 6_Led) შემზღუდავი წინააღმდეგობების გავლით.

პროგრამის კომპილიაციისა და კონტროლერში ჩატვირთვის შემდეგ როდესაც თითხს დავაჭერთ ინფრაწითელი პულტის რომელიმე ღილაკს, მაგალითად პირველს, მაშინ B1 ბლოკის პირველ გამოსასვლელებზე-FF30CF წარმოიქმნება მაღალი პოტენციალი რაც გამოიწვევს B2 ტრიგერის ერთიანის მდგომარეობაში ჩაყენებას და შესაბამისად B10 გამოსასვლელებზე მაღალი პოტენციალის წარმოქმნას, რაც აანთებს კიდეც 1_Led შუქდიოდს. პულტის პირველ ღილაკზე თითხს აშვების შემთხვევაში B1 ბლოკის პირველ გამოსასვლელებზე წარმოიშობა დაბალი პოტენციალი და ეს მოვლენა ვერ შეცვლის B2 ტრიგერის მდგომარეობას, შესაბამისად შუქდიოდი ისევ განაგრძობს ნათებას.

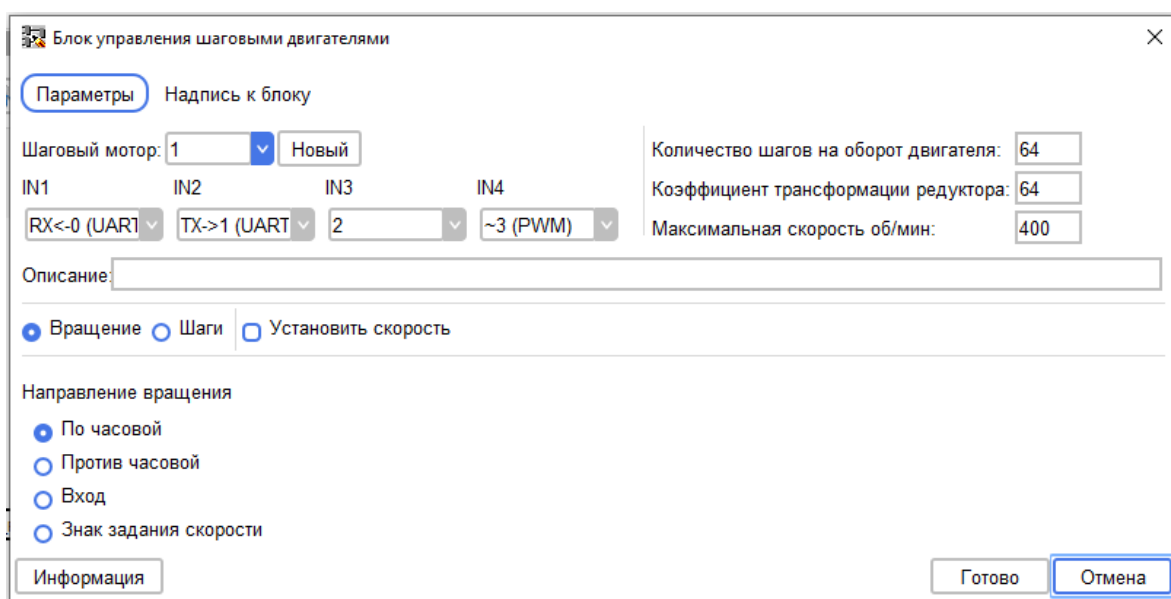
ახლა, პულტის ამ პირველ ღილაკზე ხელმეორედ თითხს დაჭერის შემთხვევაში B1 ბლოკის პირველ გამოსასვლელებზე ისევ წარმოიშობა მაღალი პოტენციალი რაც გამოიწვევს B2 ბლოკის მდგომარეობის ცვლილებას, ანუ იგი ლოგიკური ერთის მდგომარეობიდან გადავარდება ლოგიკური ნოლის მდგომარეობაში და შუქდიოდი ჩაქრება. პირველი ტრიგერის ერთიანის მდგომარეობაში ჩაყენებით გარდა 1_Led შუქდიოდის ანთებისა, გაიშვება აგრეთვე იმპულსების გენერატორი B8 (მის EN შესასვლელებზე დაჯდება გენერატორის მუშაობის ნების დართვის სიგნალი ლოგიკური ერთიანი), რაც გამოიწვევს მის გამოსასვლელებზე სიმეტრიული სიგნალების გენერირებას 2 წამი იმპულსითა და 2 წამი პაუზით. ამ გენერატორის გამოსასვლელი მიერთებულია სერვოდრავთან არდუინოს მე-4 კონტაქტის გამოყენებით.

სერვოდრავის ბლოკი B9 გაყოფილია ორ ქვებლოკად: B9_1 და B9_2. პირველი ქვებლოკის დანიშნულებაა, როდესაც გენერატორის გამოსასვლელზე იმპულსია, ანუ ლოგიკური ერთიანია, მაშინ ძრავის EN შესასვლელზე ლოგიკური ერთიანია, ანუ ნება ერთევა B9_1 -ს მუშაობაზე და ძრავის ლილვი უნდა მობრუნდეს 180 გრადუსზე (იხ. მისი პარამეტრიზაციის ფანჯარა ფიგ. 45.10-ზე), ამ დროს B9_2 გაჩერებულია რადგან მის EN შესასვლელზე გენერატორიდან გამოსული სიგნალი ინვერტირებულია ანუ ამ შემთხვევაში ლოგიკური ნოლია, რასაც ახორციელებს პატარა რგოლი რომელიც ამ მეორე ქვებლოკის შესასვლელზეა დასმული. როდესაც გენერატორის გამოსასვლელზე პაუზაა, მაშინ ეს ნულოვანი სიგნალი კრძალავს B9_1 ბლოკის მუშაობას, მაგრამ მისი ინვერტირებული სიგნალი, ანუ ლოგიკური ერთიანი, ნებას დართავს B9_2 ბლოკის მუშაობს და რადგან მისი პარამეტრიზაცია (ფიგ. 45.11) ითვალისწინებს ძრავის მობრუნებას 0 გრადუსზე, ამიტომ სერვოდრავის ლილვი მობრუნდება და დაყენდება 0 გრადუსზე.

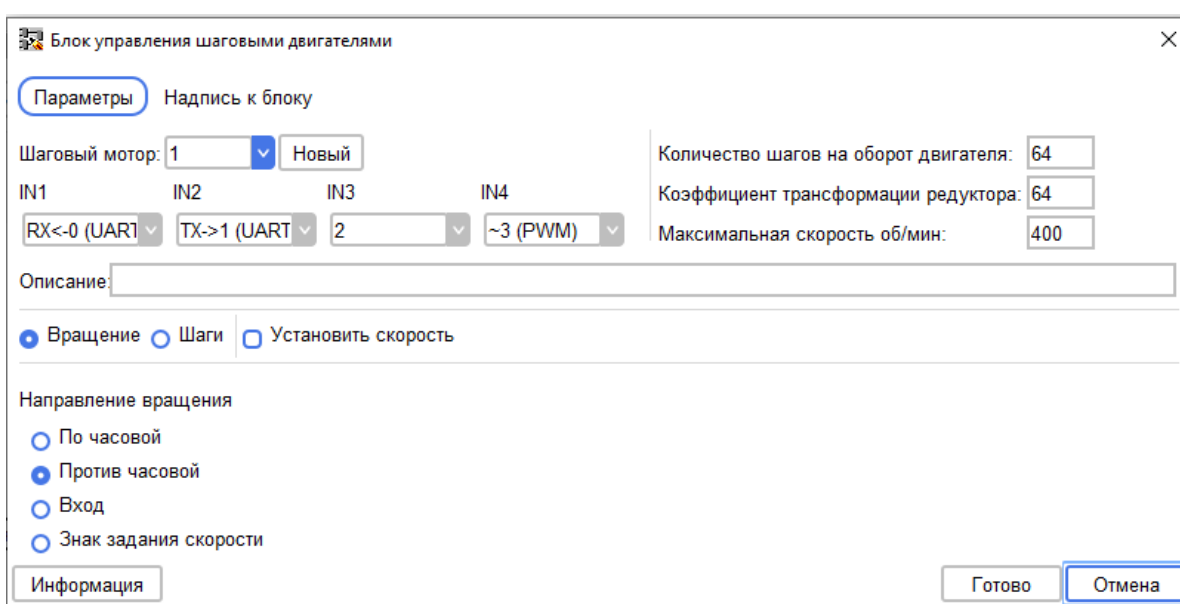
ფიგ. 45.10 სერვოდრავის დაპროგრამება 180 გრადუს მობრუნებაზე

ფიგ. 45.11 სერვოდრავის დაპროგრამება 0 გრადუს მობრუნებაზე

ამის შემდეგ უნდა გადავიდეთ ბიჯური ძრავის მართვაზე. აქაც მისი ბლოკი გაყოფილია ორ ქვებლოკად B17.1 და B17.2, იხილეთ პროგრამა ფიგ. B 45.9, რომელთაგან თუ პირველს მიეწოდა ნების დართვის სიგნალი EN შესასვლელზე მაშინ იგი მოძრაობს საათის ისრის მიმართულებით ამ შემთხვევის პარამეტრიზაციის ფანჯარა წარმოდგენილია ფიგ. 45.12-ზე, თუ კი მეორეს მიეწოდა ნების დართვის სიგნალი EN შესასვლელზე, მაშინ იგი მოძრაობს საათის ისრის საწინააღმდეგო მიმართულებით, ამ შემთხვევის პარამეტრიზაციის ფანჯარა კი წარმოდგენილია ფიგ. 45.13-ზე .



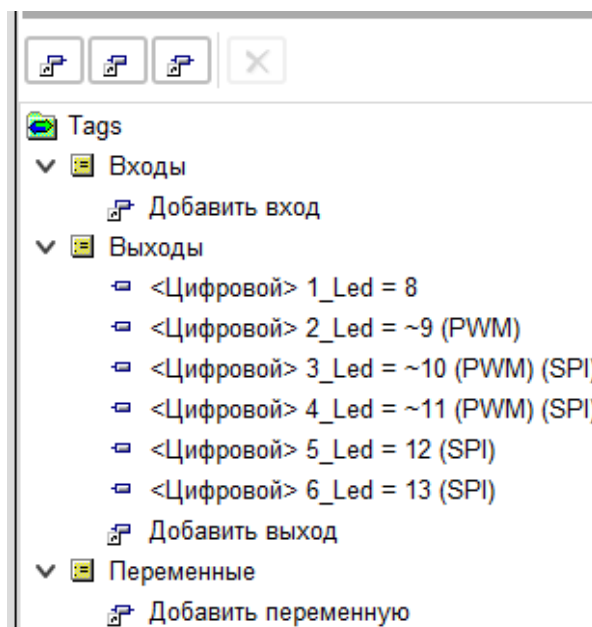
ფიგ. 45.12 სერვოდრავის დაპროგრამება საათის ისრის მიმართულებით მოძრაობაზე



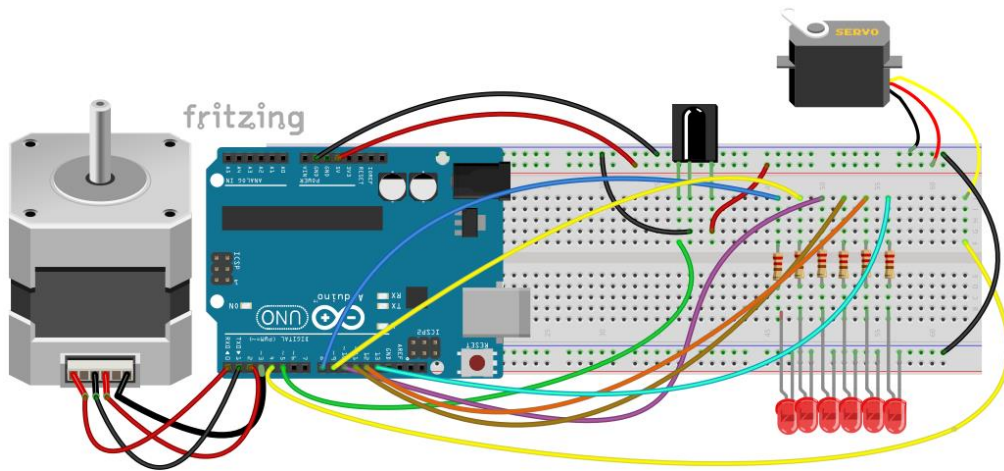
ფიგ. 45.13 სერვოდრავის დაპროგრამება საათის ისრის საწინააღმდეგოდ მოძრაობაზე

ახლა განვიხილოთ, თუ როგორ უნდა მივაწოდოთ მართვის სიგნალები ამ ბიჯურ ძრავას ალგორითმში დასმული ამოცანის გადასაწყვეტად. დავაჭიროთ პულტის მე-5 ღილაკს თითი. ეს გამოიწვევს B1 ბლოკის შესაბამის გამოსასვლელზე მაღალი პოტენციალის გამოჩენას, რომელიც დააყენებს მე-5 ტრიგერს ერთიანის მდგომარეობაში და ეს სიგნალი მიეწოდება ლოგიკური გამრავლების ბლოკის B16-პირველ შესასვლელს, რადგან ჩვენ მეექვსე ღილაკისთვის ჯერ თითი არ დაგვიჭერია, ამიტომ მეექვსე ტრიგერიდან გამოსული ნულოვანი სიგნალი ინვერტირდება B16 ბლოკის მეორე შესასვლელზე, რითაც ამუშავდება B16 ბლოკი, გამოიმუშავებს გამოსასვლელზე ლოგიკურ ერთიანს და ამუშავებს ბიჯური ძრავის პირველ ქვებლოკს B17_1, იგი დაიწყებს ბრუნვას საათის ისრის მოძრაობის მიმართულებით. თუ კი შემდეგ ჩვენ დავაწვებით პულტის მეექვსე ღილაკსაც თითს, მაშინ ერთიანის მდგომარეობაში გადავარდება მე-6 ტრიგერიც, იქიდან წამოსული სიგნალი ნებას დართავს B17_2 ქვებლოკს ამოდროს ძრავი საათის ისრის საწინააღმდეგო მიმართულებით და ამავე დროს ინვერტირების შედეგად ჩაკეტავს B16 ლოგიკურ ელემენტს და შეწყვეტს B17_1 ქვებლოკის ნებართვას ძრავის საათის ისრის მიმართულებით მოძრაობაზე. ამით მუშაობის ალგორითმი ორთავე ძრავისთვის შესრულებული იქნება.

პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 45.13-ზე, ხოლო მოწყობილობის Fritzing სქემა მოცემულია ფიგურა 45.14-ზე.



ფიგ. 45.14 პროექტის ტეგების ზონის ფანჯარა

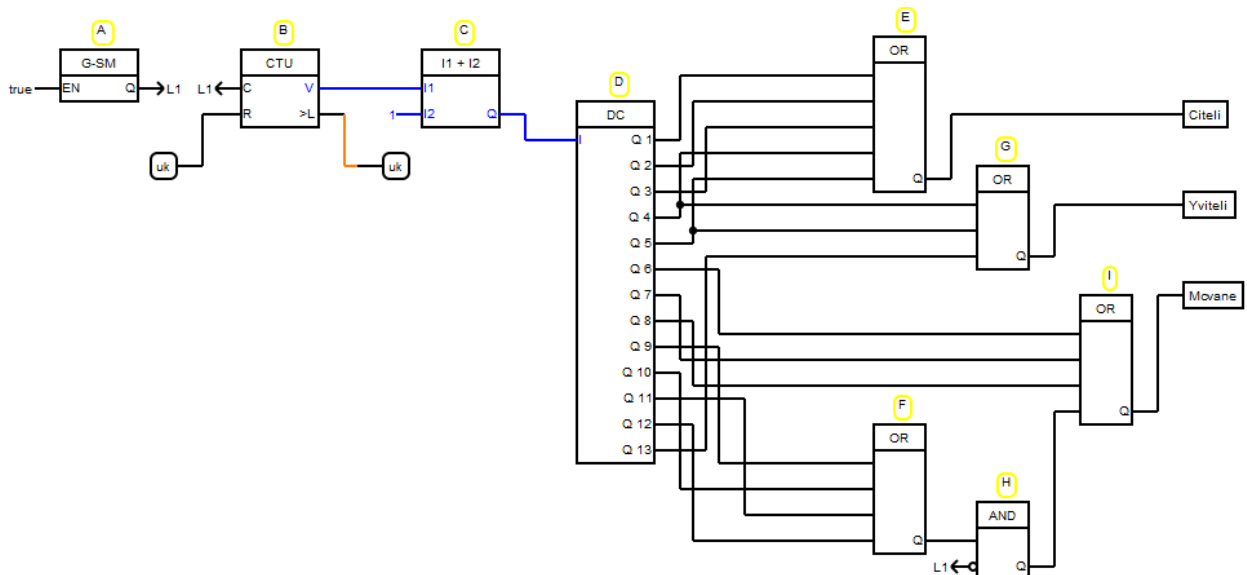


ფიგ. 45.14 ობიექტების შორიდან მართვის Fritzing სქემა

პროექტი_46 სვეტაფორი

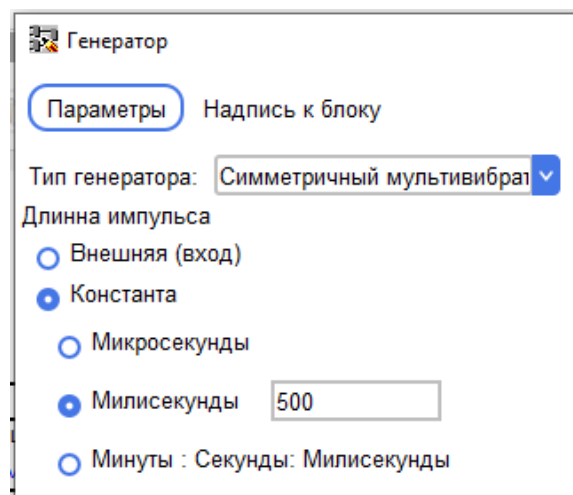


სვეტაფორის მუშაობის ალგორითმი ჩამოვყალიბოთ შემდეგნაირად. თავიდან უნდა აინთოს წითელი შუქნიშანი. სამი წამის შემდეგ დამატებით უნდა აინთოს ყვითელი შუქნიშანიც. ისინი ორთავე ერთად უნდა ანათებდეს ორი წამის განმავლობაში, შემდეგ ორივე უნდა ჩაქრეს და აინთოს მწვანე შუქდიოდი. ეს უკანასკნელი უნდა ანათებდეს სამი წამის განმავლობაში და შემდეგ უნდა აციმციმდეს, ოთხჯერ უნდა ჩაქრეს და ოთხჯერ აინთოს ნახევარ-ნახევარი წამის განმავლობაში. ამის შემდეგ იგი უნდა ჩაქრეს და ერთი წამის განმავლობაში ისევ აინთოს ყვითელი შუქდიოდი, ხოლო შემდეგ იგიც უნდა ჩაქრეს და ისევ აინთოს წითელი შუქდიოდი, და ასე შემდეგ თავიდან, ციკლურად.

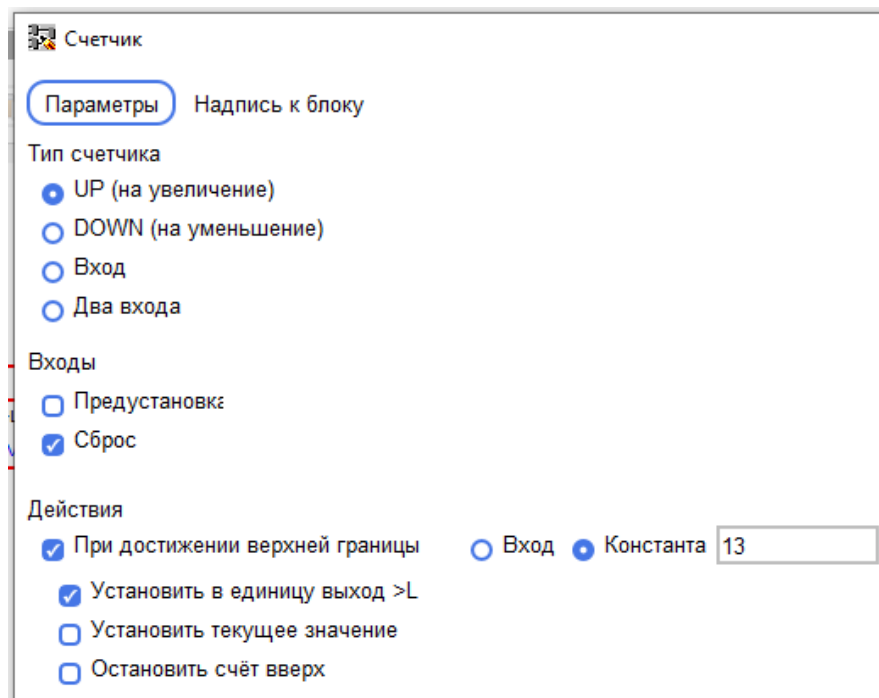


ფიგ. 46.1 სვეტაფორის სქემის ფანჯარა FLProg გარემოში

ამ ალგორითმის პროგრამული რეალიზაცია წარმოდგენილია ფიგ. 46.1-ზე. იგი ადვილად განსახორციელებელია თუ გამოვიყენებთ დემიფრატორს (D). მაგრამ დასაწყისში საჭიროა შეიქმნას გენერატორი (A), რომელიც უნდა დაპროგრამირდეს როგორც სიმეტრიული მულტივიბრატორი იმპულსის ხანგრძლიობით 500 მლწ. (ფიგ 46.2).



ფიგ. 46.2 გენერატორის პარამეტრიზაციის ფანჯარა



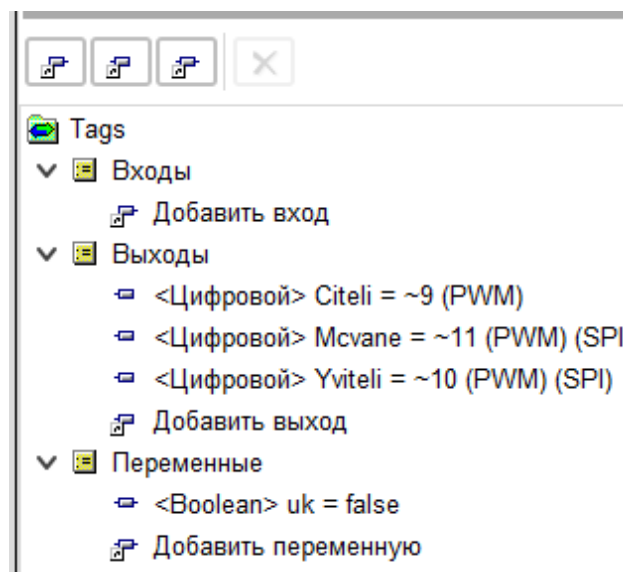
ფიგ. 46.3 მთვლელის პარამეტრიზაციის ფანჯარა

ამ გენერატორის გამოსასვლელი შეერთებულია იმპულსების მთვლელთან (B), რომელიც დაპროგრამებულია პირდაპირი თვლის რეჟიმში სამუშაოდ და ითვლის 13-მდე ციკლურად (ფიგ. 46.3). როდესაც მთვლელი დაითვლის მე-13 იმპულსს, მაშინვე ამუშავდება უკუკავშირის ცვლადის სქემა uk და მთვლელს დააყენებს საწყის ნულოვან მდგომარეობაში. ამის შემდეგ მისი მუშაობის ციკლი მის შესასვლელზე იმპულსების მოსვლის პროცესში - ისევ განმეორდება. მთვლელის მდგომარეობას უნდა დავუმატოთ 1 (C), რათა დემიფრატორმა შეძლოს მთვლელის ყველა პოზიციის დაფიქსირება. მთვლელის მუშაობის პროცესში დემიფრატორს გამოსასვლელებზე, ჯერ Q1-ზე გამოვა სიგნალი ლოგიკური ერთიანი, მერე Q2-ზე და ასე შემდეგ, ეს დსიგნალი მოივლის ყველა პოზიციას და დაუბრუნდება Q1-ს, შემდეგ ციკლი განმეორდება. დემიფრატორის გამოსასვლელებზე ლოგიკური ერთიანის გადაადგილება ხდება ყოველ ერთ წამში ერთჯერ.

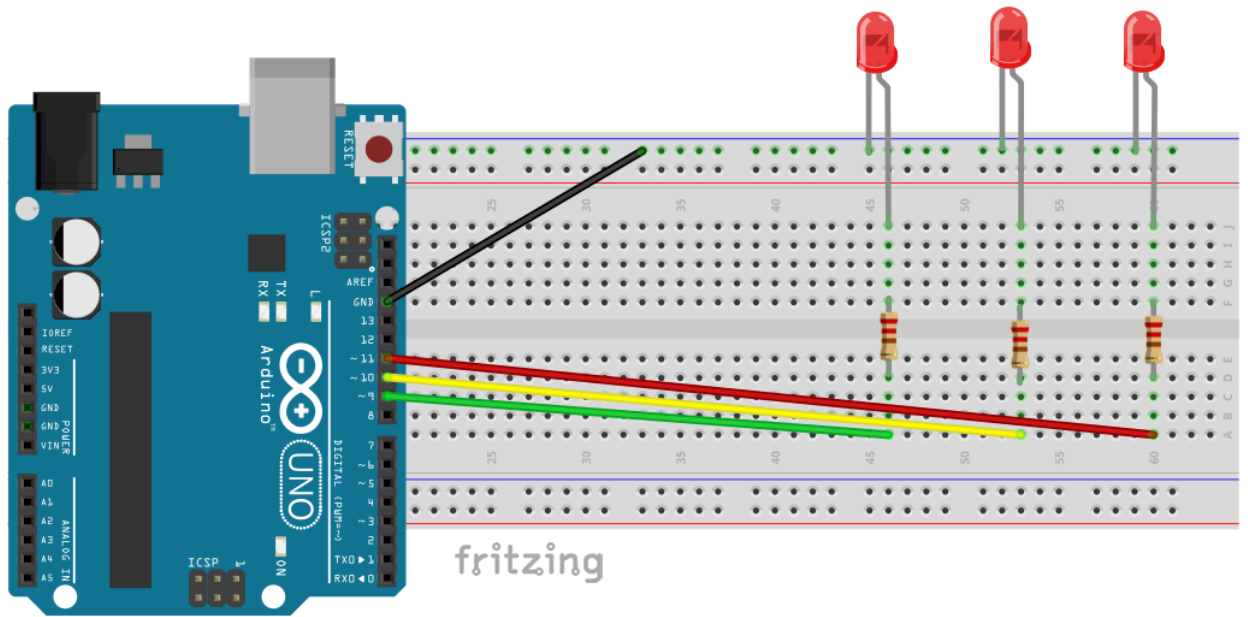
„ახლა დავუბრუნდეთ ჩვენს პროგრამას ფიგ. 46.1-ზე. როგორც ვხედავთ აქ გვაქვს შექმნილი სამი გამოსასვლელი: Citeli, Yviteli და Mcvane. მათი ფორმირებისათვის გამოყენებული გვაქვს ლოგიკური შეკრების ბლოკე ბი: E, G, F, H, I. ჯერ განვიხილოთ წითელი სიგნალის ფორმირება. ალგორითმის საფუძველზე იგი უნდა იყოს ანთებული პირველი ხუთი წამის განმავლობაში, მართლაც E ბლოკზე სიგნალები მიდიან დემიფრატორის Q1...Q5 გამოსასვლელიდან, რაც ნიშნავს იმას, რომ ეს ბლოკი მოახდენს ამ სიგნალების ლოგიკურ

შეკრებას მის გამოსასვლელზე სიგნალი იქნება მთელი პირველი ხუთი წამის განმავლობაში. ახლა ვნახოთ ყვითელი, ალგორითმიდან ცნობილია, რომ იგი უნდა აინთოს მე-4 და მე-5 წამის განმავლობაში და შემდეგ კიდევ მე-13 წამის დროს. ეს ხორციელდება ლოგიკური შეკრების ბლოკზე (G), რომელზედაც მიდის სიგნალები Q4, Q5 და Q13. ახლა გადავიდეთ მწვანეზე. ალგორითმის თანახმად იგი უნდა აინთოს მე-6, მე-7, მე-8 წამებზე, ამას ახორციელებს ლოგიკური შეკრების ბლოკი I, რომლის სამ შესასვლელზე მოდის სიგნალები Q6, Q7 და Q8, მაგრამ ეს არაა საკმარისი, ალგორითმის თანახმად მწვანემ უნდა დაიწყოს ციმციმი მე-9, მე-10, მე-11 და მე-12 წამებზე და ანთებული უნდა იყოს ნახევარი წამის განმავლობაში და ჩამქრალი-ასევე მეორე ნახევარი წამის განმავლობაში. ამას ახორციელებს ჯერ ლოგიკური შეკრების ბლოკი (F), შესასვლელებზეც მოდიან სიგნალები Q9...Q12, მაგრამ ამ პოზიციებზე ციმციმის უზრუნველყოფისათვის ამ (F) ბლოკის გამოსასვლელი შედის ლოგიკური გამრავლების ბლოკის (H) პირველ ფეხზე, რომლის მეორე ფეხიც შეერთებულია იმპულსების გენერატორის (A) გამოსასვლელთან ინვერსირებული ფორმით. ეს ნიშნავს იმას, რომ მე-9...მე-12 პოზიციებზე, როდესაც გენერატორის გამოსასვლელზე პაუზებია, მაშინ (H) გამოსასვლელზე იქნება ლოგიკური ერთი, რომელიც I ბლოკის გავლით აანთებს მწვანეს, ხოლო როდესაც გენერატორის გამოსასვლელზე იმპულსია, მაშინ ლოგიკური გამრავლების ბლოკი H ჩაკეტილია, მის გამოსასვლელზე ლოგიკური ნოლია, რაც გამოიწვევს მწვანეს ჩაქრობას. საბოლოო ჯამში კი მიიღება მწვანე შუქის ციმციმის ეფექტი მე-9...მე-12 წამების განმავლობაში.

ფიგ. 46.4-ზე ასახულია პროექტის ტეგების ზონის ფანჯარა, ხოლო ფიგ. 46.5-ზე პროექტის Fritzing სქემა.

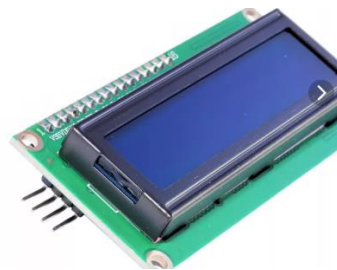


ფიგ. 46.4 პროექტის ტეგების ზონის ფანჯარა

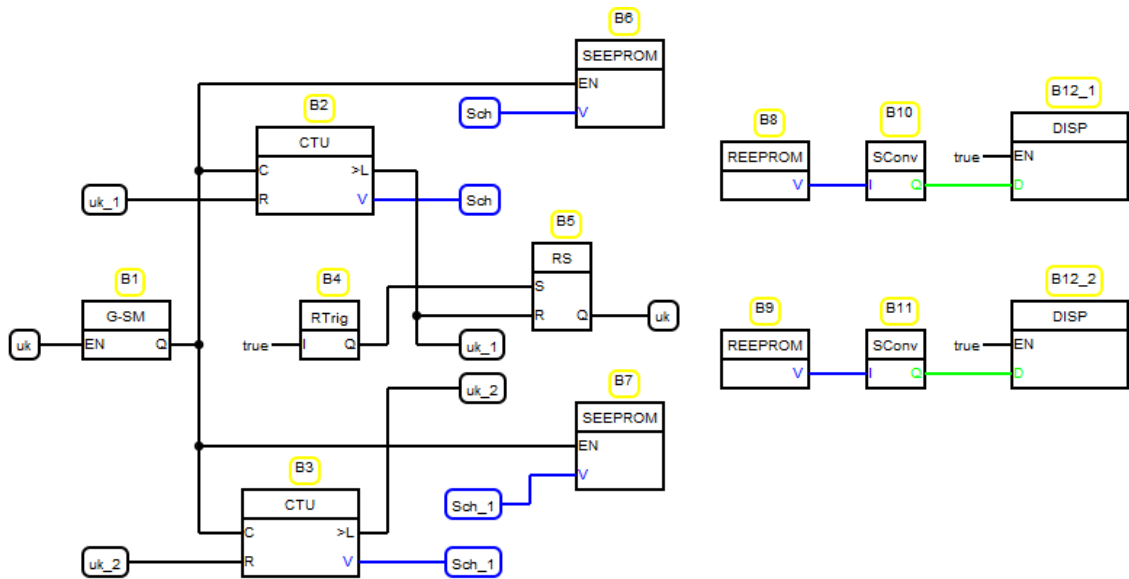


ფიგ. 46.5 პროექტის Fritzing სქემა

პროექტი_47 FFPROM



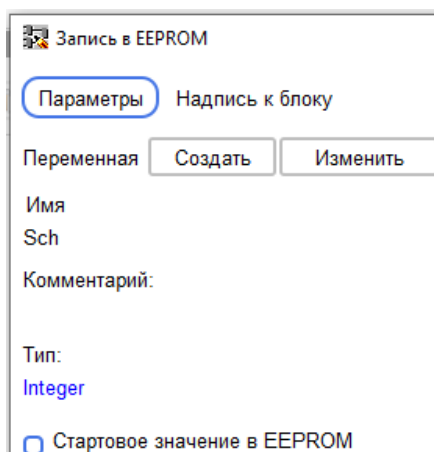
დასაწყისში უნდა ავირჩიოთ დაპროგრამირების ენა და კონტროლერის მოდელი. ამისათვის უნდა გავუშვათ FLProg-ის პროგრამა, შემდეგ უნდა დავაწკაპოთ მისი მთავარი მენიუს ფაილის მენიუზე (Файл), ჩამოშლადი სიიდან უნდა ავირჩიოთ ახალი პროექტი-Новый და გამოჩენილ ფანჯარაში სახელწოდებით პროექტის საწყისი აწყობები-Начальные настройки проекта დაპროგრამების ენად-Язык программирования ავირჩევთ FBD. კონტროლერის არეში ლილაკზე დაჭერის შედეგად კონტროლერის-Контроллер: ჩამოშლად სიაში ვირჩევთ იმ კონტროლერს, რომელიც გვაქვს ხელთ-Arduino Uno ჩვენს შემთხვევაში. ამის შემდეგ კი ვაჭერთ არჩევის ლილაკს-Выбрать, შემდეგ კი მზადყოფნის ლილაკს-Готово. ენერგოდამოუკიდებელ მეხსიერებაში-EEPROM ცვლადების ჩაწერისა და წაკითხვის პროგრამა წარმოდგენილია ქვემოთ ფიგ. 47_1-ზე.



ფიგ. 47_1 კონტროლერის ენერგოდამოუკიდებელი მეხსიერების ჩაწერის პროგრამა

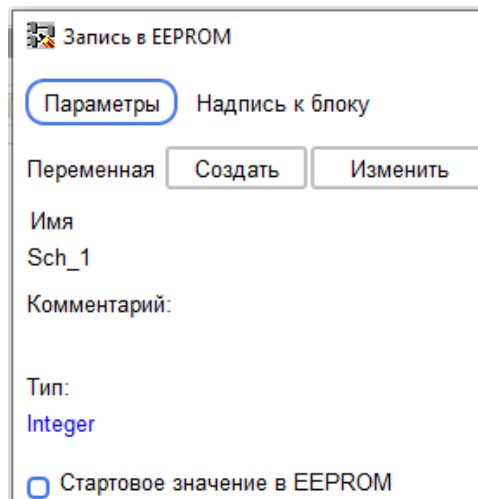
ცვლადის ჩაწერა EEPROM-ში.

SEEPROM ბლოკების დანიშნულებაა ცვლადის ჩაწერა კონტროლერის ენერგოდამოუკიდებელ მეხსიერებაში-ბლოკები B6 და B7 (ფიგ. 47_1). EN შესასვლელზე იმპულსის წინა ფრონტის დადგომისას ის მონაცემი, რაც დგას V შესასვლელზე ჩაიწერება კონტროლერის ენერგოდამოუკიდებელ მეხსიერებაში. სქემაში ბლოკის დამატების შემდეგ ბლოკის თვისებებში აუცილებელია ცვლადების შექმნა, (ფიგ. 47_2, 47_3), რომელთა მნიშვნელობების ჩაწერაა საჭირო კონტროლერში, გარდა ამისა, უნდა მივაკუთვნოთ მას სახელი და მივუთითოთ შესაბამისი ტიპი. მოხერხებულობისათვის შეგვიძლია მივაწეროთ აუცილებელი კომენტარებიც.



ფიგ. 47_2 EEPROM-ში ჩასაწერი პირველი ცვლადის შექმნა სახელწოდებით Sch

ენერგოდამოუკიდებელი მეხსიერების შევსებაზე კონტროლი ხდება ავტომატურად და თუკი მეხსიერება სრულად არის შევსებული, მაშინ მეხსიერებაში ჩასაწერი ცვლადის შექმნის მცდელობა, გამოიწვევს შესაბამისი შეტყობინობის გამოცემას და ცვლადი არ შეიქმნება.

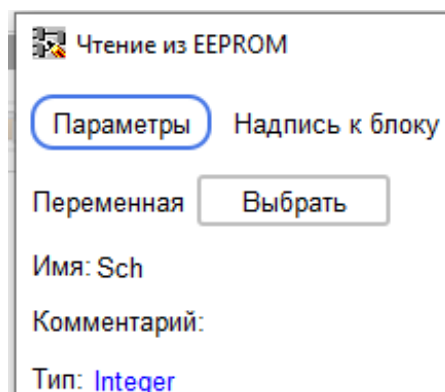


ფიგ. 47_3 EEPROM-ში ჩასაწერი მეორე ცვლადის შექმნა სახელწოდებით Sch_1

აქ ყურადღება უნდა მიექცეს იმ გარემოებასაც, რომ EEPROM-ში ჩაწერების საერთო რაოდენობა შეზღუდულია და არ უნდა გადააჭარბოს 100000. ამიტომ უნდა შევეცადოთ, რომ გამოვიყენოთ მნიშვნელობების ჩაწერა მეხსიერებაში, მხოლოდ აუცილებლობის შემთხვევაში. წაკითხვათა რაოდენობა მეხსიერებიდან - არ არის შეზღუდული.

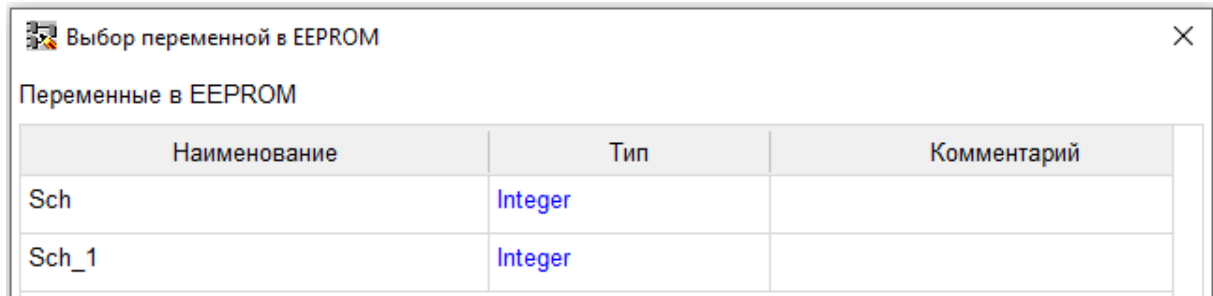
ცვლადის წაკითხვა EEPROM-იდან.

ბლოკის დანიშნულებაა კონტროლერის ენერგოდამოუკიდებელ მეხსიერებაში წინასწარ ჩაწერილი ინფორმაციის წაკითხვა. ბლოკის შექმნის შემდეგ აუცილებელია სიიდან იმ ცვლადის არჩევა, რომელიც შექმნილი იყო ცვლადის შექმნის ბლოკებში (ფიგ. 47_4).



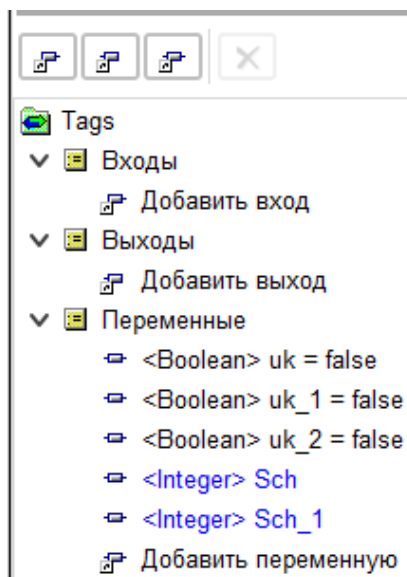
ფიგ. 47_4 ინფორმაციის წაკითხვა ენერგოდამოუკიდებელი მეხსიერებიდან

ცვლადის არჩევა EEPROM-იდან შესაძლებელია ფიგ. 47_5-ზე ასახული ფანჯრის მიხედვით-Выбор переменной в EEPROM.



ფიგ. 47_5 ცვლადის არჩევა EEPROM-იდან

როგორც ფიგ. 47_5-დან ჩანს ჩვენ EEPROM-ში ჩაწერილი გვაქვს ორი ცვლადი, რომელთაგანაც უნდა ავირჩიოთ, თუ რომლის წაკითხვაც გვინდა პირველის, მეორის თუ ორივესი ერთად, მაგრამ იმისათვის, რომ ამ ენერგოდამოუკიდებელ მეხსიერებაში მოხვდნენ ცვლადები, წინასწარ საჭიროა რომ ისინი შექმნილი იყოს პროგრამაში. მოვახდინოთ მათი შექმნა ტეგების ზონაში სტანდარტული პროცედურების გამოყენებით FLProg-ის (ფიგ. 47_6).



ფიგ. 47_6 ცვლადების Sch და Sch_1-ის შექმნა პროექტის ტეგების ზონაში

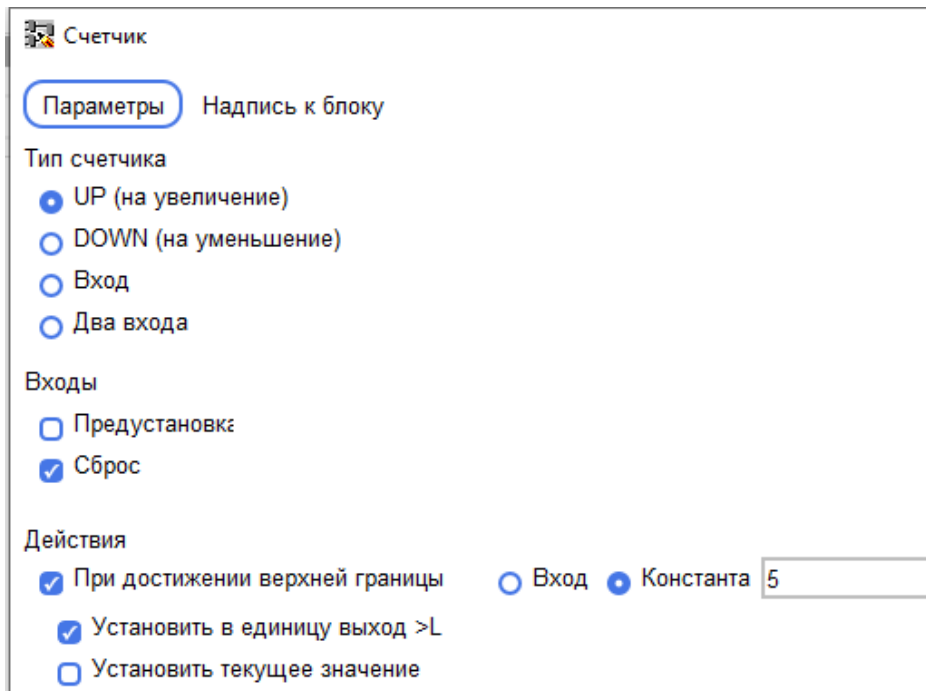
ახლა გადავიდეთ პროგრამის მუშაობის განხილვაზე, ანუ იმის განხილვაზე, თუ როგორ უნდა მოხდეს კონტროლერის ენერგოდამოუკიდებელ მეხსიერებაში ცვლადების ჩაწერა და წაკითხვა. ფიგ. 47_1-ზე წარმოდგენილია ჩვენი პროექტის პროგრამა. ძირითად მასინქრონიზებულ ელემენტად აქ წარმოდგენილია გენერატორის ბლოკი B1, რომელიც

წარმოადგენს სიმეტრიულ მულტივიბრატორს რომელიც დაპროგრამებულია ერთ წამიანი იმპულსების გამომუშავებაზე (ფიგ. 47_7).

ფიგ. 47_7 გენერატორის პარამეტრიზაციის ფანჯარა

გენერატორის გამოსასვლელი შეერებულია ორი მთვლელის შესასვლელებთან, რომელთაგან პირველი მუშაობას ცამეტამდე (ჩაუთვლელად) ციკლური თვლის რეჟიმში, ხოლო მეორე ხუთამდე (ჩაუთვლელად) ასევე ციკლური თვლის რეჟიმში და ამ რეჟიმების დამყარება ხდება უკუკავშირების uk_1 და uk_2 სიგნალების გამოყენებით (ფიგ. 47_8 და 47.9).

ფიგ. 47_8 პირველი მთვლელის პარამეტრიზაციის ფანჯარა



ფიგ. 47_9 მეორე მთვლელის პარამეტრიზაციის ფანჯარა


როდესაც პროგრამა ჩაიტვირთება არდუინოში და მოხდება მისი გაშვება, ჯერ ამუშავდება RTrig (B4), რომელიც გამოიმუშავებს ერთჯერად იმპულსს არდუინოს პირველი ციკლის განმავლობაში და დააყენებს RS ტრიგერს (B5) ერთიანის მდგომარეობაში. პარალელურად ამისა იმპულსების გენერატორი დაიწყებს იმპულსების გენერირებას ორი წამის პერიოდით და მთვლელები დაეწყებენ იმპულსების თვლას.

ახლა ვნახოთ რა მონაცემები დაფიქსირდება პირველ B2 და მეორე B3 მთვლელებში იმპულსების თვლის პროცესში: პირველი იმპულსის მოსვლის შემდეგ მთვლელებში იქნება B2-ში ერთიანი და B3 ერთიანი, რადგან ორთავემ დაითვალა თითო თითო იმპულსი მათ შესასვლელებზე ერთდროულად მიწოდებული. ჩაწეროთ ეს მდგომარეობა რიცხვების სახით: 1, 1. მეორე იმპულსის მოსვლის შემდეგ ამ მთვლელებში იქნება უკვე - 2, 2. მესამე იმპულსის მოსვლის შემდეგ - 3, 3. მეოთხე იმპულსის მოსვლის შემდეგ - 4, 4 მეხუთე იმპულსის მოსვლის შემდეგ უკვე ამუშავდება მეორე უკუკავშირი uk_2 და მეორე მთვლელს B3-ს დააყენებს ნულის მდგომარეობაში, ხოლო პირველი მთვლელი გააგრძელებს ჩვეულებრივ თვლას. ე.ი მთვლელებში იქნება - 5, 0, შემდეგ იმპულსზე - 6,1, შემდეგზე - 7, 2, შემდეგზე -8, 3, შემდეგზე - 9, 4 და შემდეგ ისევ ამუშავდება მეორე მთვლელის უკუკავშირი uk_2 და ისევ დააყენებს ამ მთვლელს ნულოვან მდგომარეობაში. ახლა უკვე მთვლელებში იქნება-10, 0, შემდეგ იმპულსზე - 11, 1, შემდეგ იმპულსზე - 12, 2. შემდეგი მე-13 იმპულსის მოსვლის წინა ფრონტზე უკვე ამუშავდება პირველი მთვლელის უკუკავშირი uk_2, რომელიც დააყენებს RS ტრიგერს B5 ნულოვან მდგომარეობაში, ნულოვანი სიგნალი ჩაკეტავს

იმპულსების გენერატორს B1-ს და იმპულსების გენერირება შეწყდება. ამავდროულად ეს უკუკავშირი დააყენებს ამ პირველ მთვლელს ნოლოვან მდგომარეობაში და საბოლოო ჯამში B1 მთვლელში დაფიქსირდება და დარჩება რიცხვი 0, ხოლო B2 მთვლელში კი რიცხვი 2.

B1 გენერატორის იმპულსების ყოველი წინა ფრონტი ახდენს აგრეთვე B6 და B7 ბლოკების ამუშავებას, ანუ მათ ანალოგურ შესასვლელებზე მიწოდებული ინფორმაციის, ანუ მთვლელების მდგომარეობათა ჩაწერას კონტროლერის ენერგოდამოუკიდებელ მეხსიერებაში. განერატორის დაბლოკვამდე ამ ბლოკებში მირიგეობით ჩაიწერება ის რიცხვები, რომლებიც ზემოთ იყო ჩამოთვლილი, ხოლო გენერატორის დაბლოკვის შემდეგ პირველ ბლოკში დარჩება ჩაწერილი რიცხვი 12, ხოლო მეორეში კი რიცხვი 2. ამ რიცხვების წაკითხვა იწარმოებს პარალელურად ჩაწერის პროცესისა B8 და B9 ბლოკების მიერ და მათი ასახვა მოხდება B10 და B11 კონვერტაციის ბლოკების გავლით B12_1 და B12_2 დისპლეის ბლოკებზე, ანუ პირველი მთვლელისა დისპლეის პირველ სტრიქონში და მეორე მთვლელისა დისპლეის მე-2 სტრიქონში. ამ ბლოკების პარამეტრიზაციის ფანჯრები ასახულია ფიგ. 47_9-ზე და ფიგ. 47_10.

აქ კიდევ ერთჯერ უნდა გაესვას ხაზი იმ გარემოებას, რომ EEPROM-ში ინფორმაციის ჩაწერა ხდება მხოლოდ ერთჯერ არდუინოს გაშვების მომენტში. ჩაწერის პროცესის დამთავრების შემდეგ B5 ტრიგერი დგება ნულოვან მდგომარეობაში და ჩაწერის პროცესი წყდება. თუ გვინდა, რომ ჩაწერის პროცესი განმეორდეს საჭიროა დავაწვეთ არდუინოს გადატვირთვის ღილაკს თითი, რომელიც თვითონ არდუინოს დაფაზეა მოთავსებული, ამით მთელი სქემა დადგება საწყის მდგომარეობაში და ჩაწერის მთელი პროცესი განმეორდება. ეს იმიტომაა საჭირო, რომ თუ ჩვენ უსასრულოდ გავაგძელებთ ჩაწერის პროცესს, მაშინ სწრაფად გადავაჭარბებთ კონტროლერის მუდმივ მეხსიერებაში ჩაწერისათვის განკუთვნილ ლიმიტს 100000 და ამით არდუინო დაკარგავს ამ ფუნქციის შესრულების უნარს. პროექტის Fritzing სქემა ასახულია ფიგ. 47_11-ზე.


Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/> <input type="button" value="v"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/> <input type="button" value="v"/>	Комментарий	<input type="text"/>	

Данные:


Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 47_9 HD44877 დისპლეის პირველი სტრიქონის პარამეტრიზაციის ფანჯარა


Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/> <input type="button" value="v"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/> <input type="button" value="v"/>	Комментарий	<input type="text"/>	

Данные:

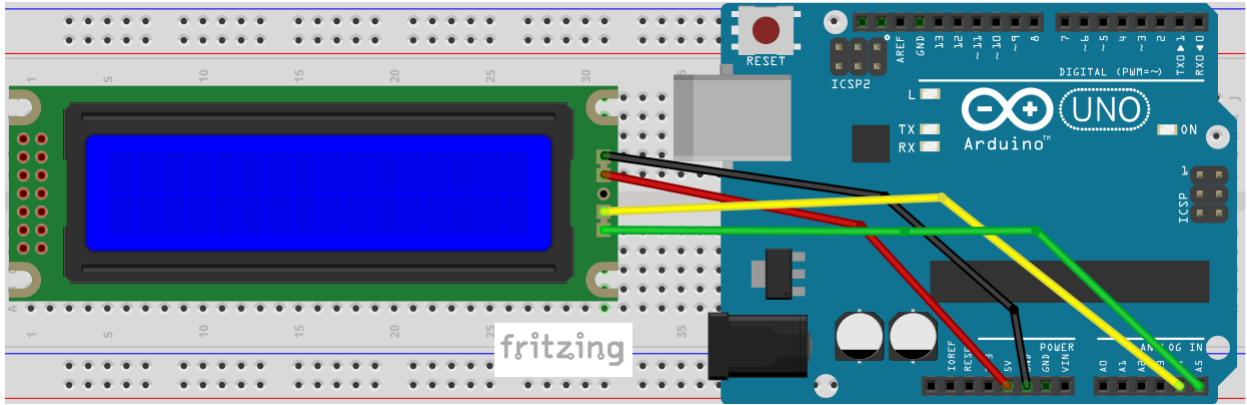
Вход

Константа

Строка: Центровать

Столбец:

ფიგ. 47_10 HD44877 დისპლეის პირველი სტრიქონის პარამეტრიზაციის ფანჯარა



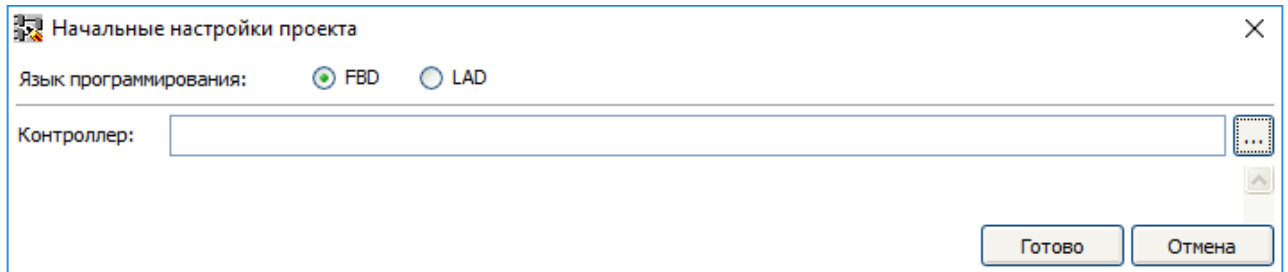
ფიგ. 47_11 პროექტის Fritzing სქემა

პროექტი_48

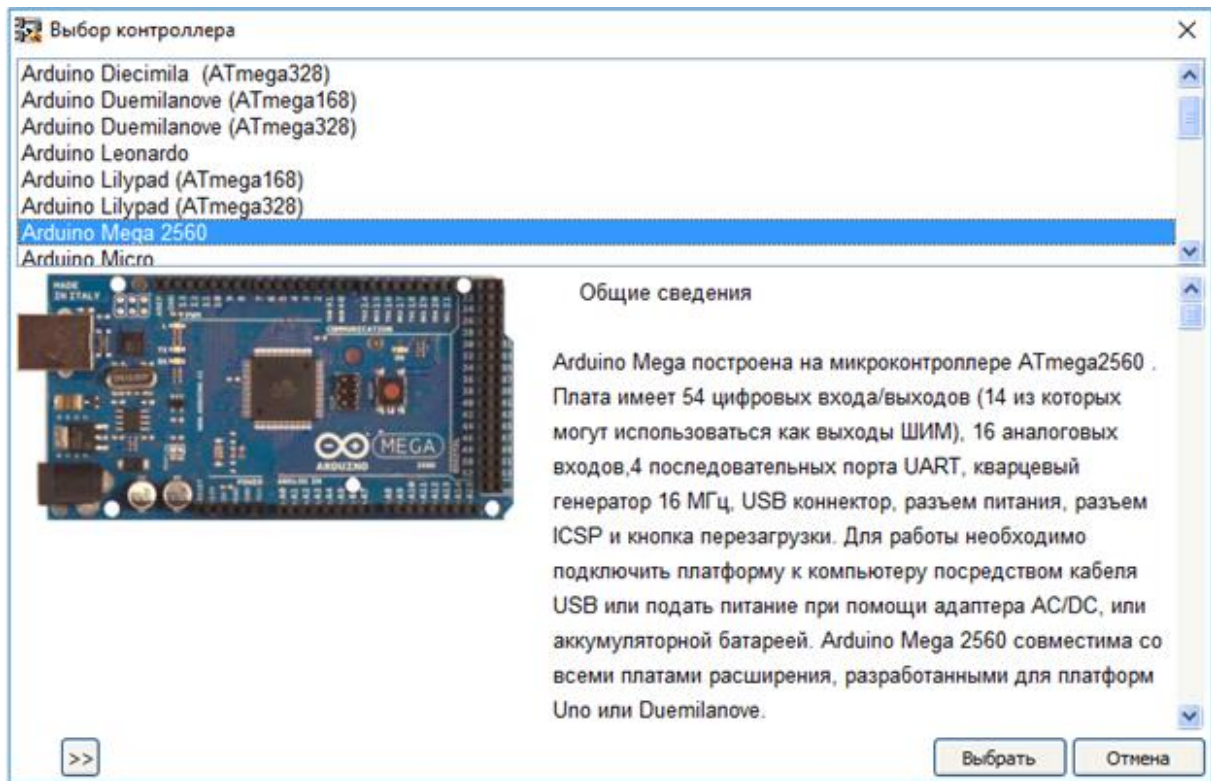
გირლანდა და ორობითი მთვლელი



გავუშვათ FLProg-ის პროგრამა, დავაწკაპუნოთ მთავარი მენიუს ფაილის მენიუზე (Файл), ჩამოშლადი სიიდან ავირჩიოთ ახალი-Новый და გამოჩენილ ფანჯარაში (ფიგ. 48.1) დაპროგრამების ენად-Язык программирования ავირჩიოთ FBD. ხოლო კონტროლერის ველში ... დილაკზე დაჭერის შედეგად კონტროლერის-Контроллер: ჩამოშლად სიაში ვირჩევთ იმ კონტროლერს, რომელიც გვჭირდება ამ პროექტის შესასრულებლად-Arduino Mega2560 ჩვენ შემთხვევაში (ფიგ. 48.2). ამის ემდეგ კი ვაჭერთ არჩევის დილაკს-Выбрать, შემდეგ კი მზადყოფნის დილაკს-Готово.



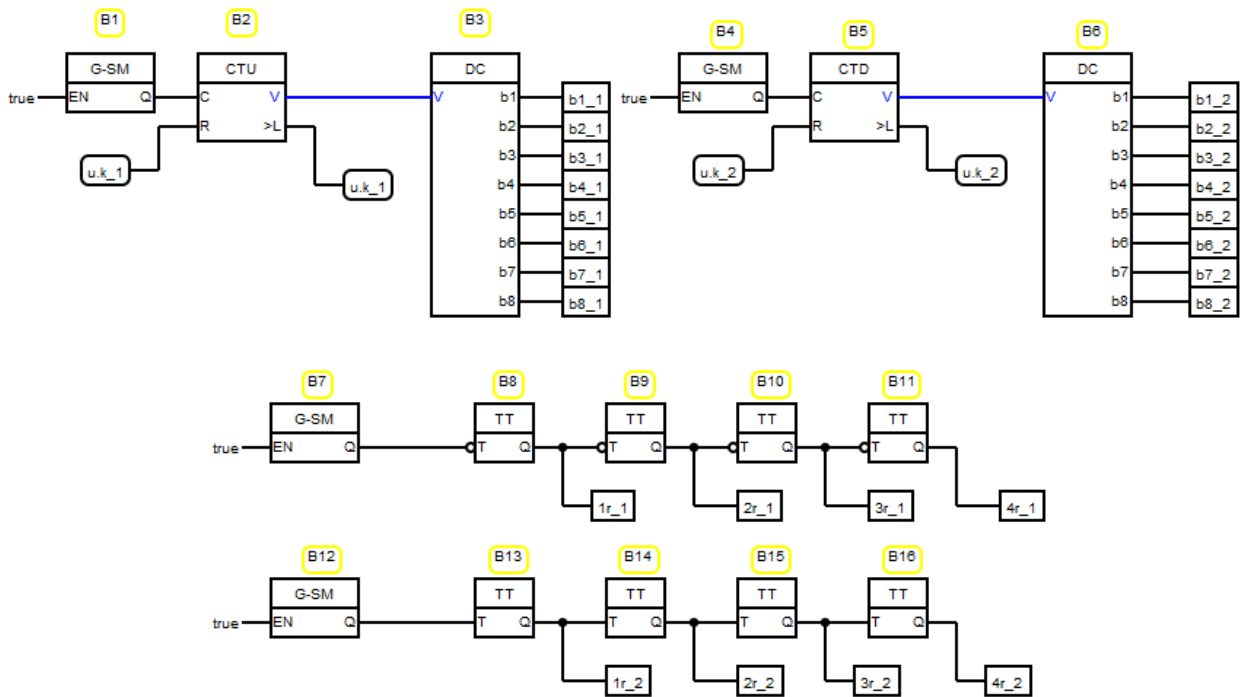
ფიგ. 48.1 პროექტის საწყისი გაწყობის ფანჯარა



ფიგ. 48.2 დაპროგრამების ენისა და კონტროლერის არჩევის ფანჯარა

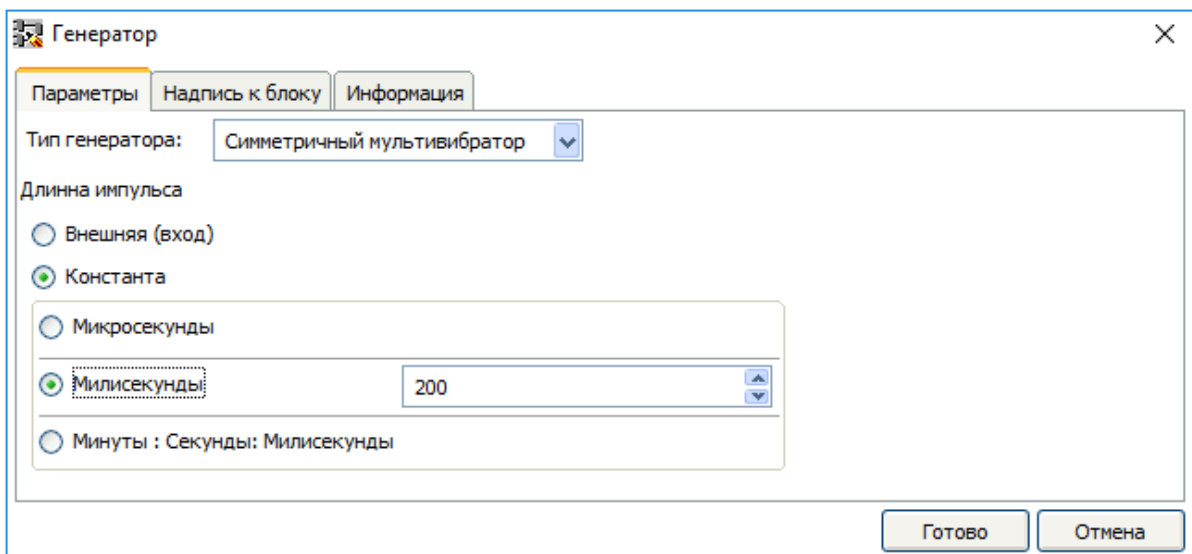
ფიგურა 48.3-ზე წარმოდგენილია ჩვენი პროექტის პროგრამა შესრულებული FLProg IDE გარემოში. გირლანდები წარმოადგენს რვა-რვა შუქდიოდებისაგან შემდგარ ორ სისტემას, რომლებიც ანათებენ წანაცვლებით ერთიდაიგივე სიხშირით და სადაც შუქდიოდების ნათებები გადაადგილდება ურთიერთსაწინააღმდეგო მიმართულებით. მათი დაპროგრამირებისათვის გამოიყენება B1, B2, B3 და B4, B5, B6 ბლოკები შესაბამისად.

პირველი გირლანდის მუშაობის უზრუნველყოფისათვის საჭიროა FLProg-ის ბლოკების ბიბლიოთეკის-Библиотека блоков ტაიმერების საქაღალდედან-Таймеры თავის მარცხენა ღილაკის ჩავლებით პროგრამის სამუშაო ზონაში გადმოვაცოცოთ იმპულსების გენერატორის ბლოკი-Generator (B1) და მოვახდინოთ მისი დაპროგრამება.



ფიგ. 48.3 გირლანდებისა და ორობითი მთვლელების დაპროგრამების ფანჯარა

ამისათვის დავაწკაპუნოთ ამ ბლოკზე თავის მარცხენა ლილაკით ორჯერ და გამოჩენილ ფანჯარაში (იხ. ფიგ. 48.4) მოვახდინოთ მისი პარამეტრიზაცია.



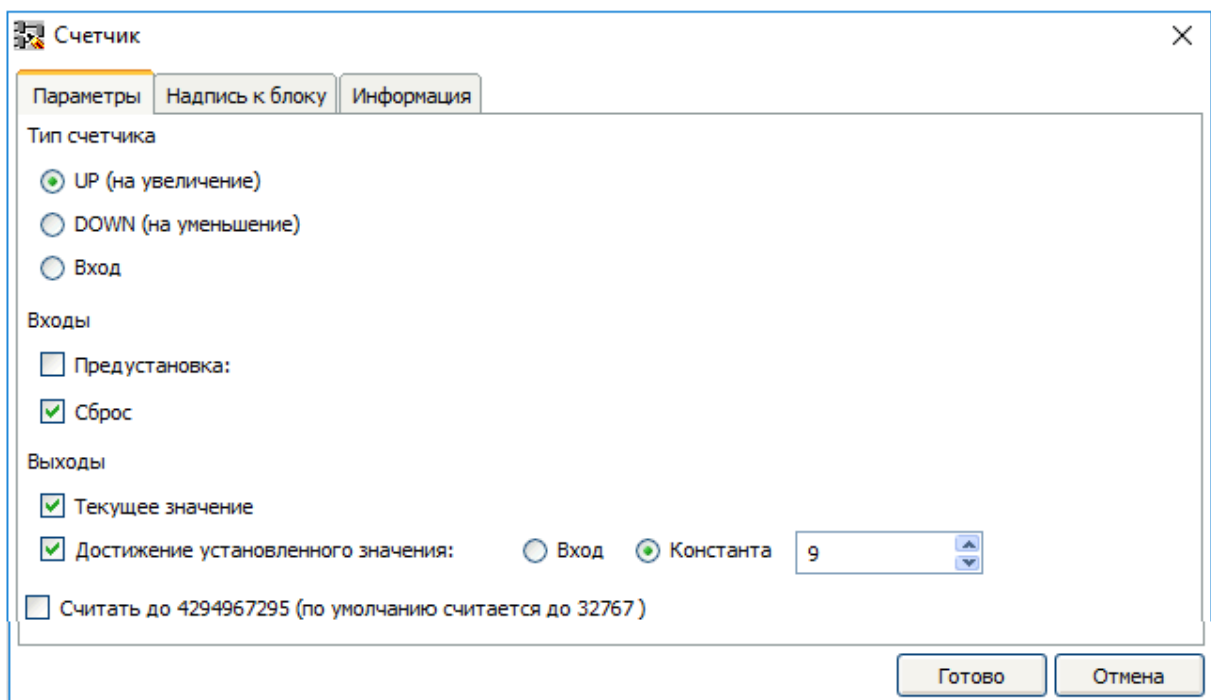
ფიგ. 48.4 გენერატორის დაპროგრამების ფანჯარა

გენერატორის ტიპის ველში-Тип генератора ავირჩიოთ სიმეტრიული მულტივიბრატორი-Симметричный мультивибратор, იმპულსის სიგრძის ველში-длина импульса ჩავრთოთ ჩამრთველი კონსტანტა-константа და შემდეგ ჩავრთოთ წუთები: წამები: მილიწამები-ს ველში-Миллисекунды შესაბამის ფანჯარაში ჩავწეროთ ჩვენთვის საჭირო

დროის ხანგრძლიობა, რა პერიოდულობითაც გვინდა შეასრულოს შუქდიოდების ნათებებმა გადაადგილება. (ჩვენ მაგალითში აღებული გვაქვს 200 მილიწამი). გარდა ამისა, ვაწკაპუნებთ ამ ბლოკის EN შესასვლელზე და ვირჩევთ ნიბისდართვის ოპციას-True (1), რაც იმას ნიშნავს, რომ პროგრამის გაშვების შემთხვევაში გენერატორის მუშაობა ნებადართული იქნება განუწყვეტელი დროის განმავლობაში, თვით პროგრამის მუშაობის დამთავრებამდე. ამის შემდეგ ვაწკაპებთ მზადყოფნის ლილაკს-Готово, რის შედეგადაც ფანჯარა გაქრება და გენერატორის დაპროგრამება დამთავრებული იქნება. ასევე უნდა მოხდეს (B4) გენერატორის დაპროგრამება, იმ განსხვავებით, რომ იქ მილიწამების სიდიდეები-Милисекунды შესაძლოა სხვა ნებისმიერი სიდიდით შეიცვალოს, მაგ. 400 მილიწამი ჩვენ შემთხვევაში.

ასევე მოხდება დანარჩენი ორი გენერატორების - (B7, B12) დაპროგრამება, იმ განსხვავებით, რომ იქ მილიწამები - Милисекунды შესაძლოა კიდევ სხვადასხვა ნებისმიერი სიდიდეები იყოს, ანუ ჩვენ შემთხვევაში 150 და 250 შესაბამისად.

ახლა, ბლოკების ბიბლიოთეკის მთვლელების საქალაქედან-Счетчики პროგრამის სამუშაო ზონაში თავის მარცხენა ლილაკის ჩავლებით გადმოვიტანოთ მთვლელის ბლოკი-Counter (B2) და მოვახდინოთ მისი პარამეტრიზაცია.



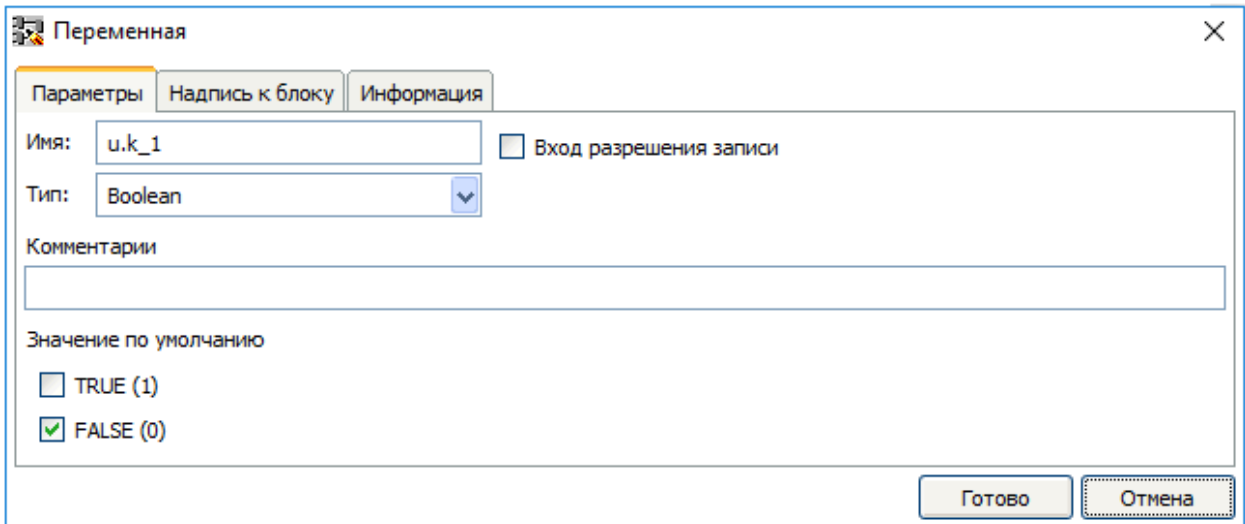
ფიგ. 48.5 მთვლელის პარამეტრიზაციის ფანჯარა

ამისათვის დავაწკაპუნოთ ამ ბლოკზე ორჯერ თავის მარცხენა ლილაკით და გამონათებულ ფანჯარაში (ფიგ. 48.5) ჩავრთოთ ჩამრთველი ზრდაზე-На увеличение (UP), დავაყენოთ ჩამოყრის (ნულზე დაყენების) ალამი- Сброс, გამოსასვლელების ველებში-Выходы

დავაცენოთ მიმდინარე მნიშვნელობისა-Текущее значение და დაცენებულ მნიშვნელობებთან მიღწევის-Достижение установленного значения: ალმები, აგრეთვე ჩავრთოთ კონსტანტის ჩამრთველი-Константа და შესაბამის ველში ჩავწეროთ 9 (9 იმიტომ ვწერთ, რომ ჩვენ რვა შუქდიოდი გვაქვს სამართავი და გადასათვლელი, ხოლო მეცხრე იმპულსმა უნდა გაანულოს მთვლელი და შემდეგი იპულსიდან თავიდან დაიწყოს თვლა) და შემდეგ დავაჭიროთ მზადყოფნის ღილაკს-Готово თავგის მარცხენა ღილაკით.

ამ მთვლელზე იმპულსების მიწოდების შემთხვევაში იგი დაითვლის მათ და ამ დათვლილი რაოდენობის მიმდინარე მნიშვნელობებს გამოსცევს V გამოსასვლელზე ანალოგური ფორმით.

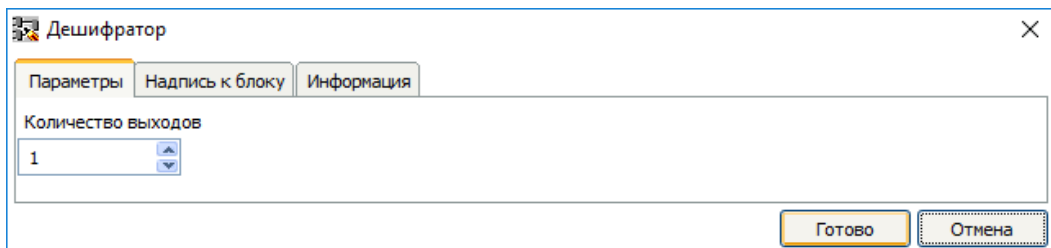
როდესაც მთვლელზე მიეწოდება მე-9 იმპულსი (რადგანაც ჩვენ კონსტანტის ველში 9 ჩავწერეთ), მაშინ ამ გენერატორის >L გამოსასვლელზე გამომუშავდება ლოგიკური ერთიანი და ჩვენი მიზანი უნდა იყოს, რომ ამ მომენტში მთვლელი დავაცენოთ ნულზე, რათა მან შემდგომში თავიდან დაიწყოს იმპულსების გადათვლა და ა.შ.შ. განუწყვეტლივ პროგრამის მუშაობის გაჩერებამდე. მთვლელის ნულზე დაცენებისათვის ლოგიკური გამოსავალია, რომ ეს >L გამოსასვლელი პირდაპირ შევაერთოთ ამავე მთვლელის ნულზე დაცენების R შესასვლელზე, მაგრამ ამ პირდაპირი შეერთების საშუალებას FLProg პროგრამა არ გვაძლევს და ამისათვის გამოვიყენოთ შეერთება ცვლადის დახმარებით. ამ მიზნით გავხსნათ პროგრამის ტეგების ზონაში-Tags ცვლადების-Переменные საქაღალდე და დავაწკაპოთ ცვლადის დამატების ტეგს-Добавить переменную (ფიგ. 48.6). გამოსულ ფანჯარაში დავარქვათ სახელწოდების ველში-Имя: დავარქვათ ცვლადს ჩვენთვის მოსახერხებელი სახელი (u.k_1 პირველი უკუ კავშირი ჩვენ შემთხვევაში), ცვლადის ტიპის ველში-Тип: ავირჩიოთ Boolean, რადგან ამ ცვლადს მხოლოდ ბულის მნიშვნელობები შეიძლება ჰქონდეს - ლოგიკური 0 ან ლოგიკური 1. სიჩუმით მნიშვნელობის ველში-Значение по умолчанию დავაცენოთ ალამი-False (0), რადგან საწყის მომენტში ამ ცვლადის მნიშვნელობა უნდა იყოს ლოგიკური ნულის ტოლი. შემდეგ, დავაწკაპუნოთ მზადყოფნის ღილაკზე-Готово, რითაც დავამთავრებთ ცვლადის შექმნასა და პარამეტრიზაციას. ამის შემდეგ ეს ცვლადი თავგის მარცხენა ღილაკის გამოყენებით ორჯერ გადმოვიტანოთ პროგრამის სამუშაო ზონაში და შევაერთოთ მთვლელის გამომყვანებს ისე, როგორც ეს ფიგ. 48.3-ზეა ნაჩვენები.



ფიგ. 48.6 ცვლადის პარამეტრიზაციის ფანჯარა

ამის შემდეგ უნდა შევუდგეთ მეორე მთვლელის (B5) დაპროგრამებას, რომელსაც ისევე განვახორციელებთ, როგორც პირველი მთვლელისათვის, იმ განსხვავებით, რომ თვლის მიმართულებად აქ უნდა ავირჩიოთ არა მატება-(UP) (ფიგ. 48.5), არამედ კლება - На уменьшение-(DOWN), ხოლო მისი უკუ კავშირისათვის უნდა შევქმნათ კიდეც ახალი ცვლადი სახელწოდებით (u.k_2) და დავაპროგრამოთ იგი ისევე, როგორც u.k_1 და შევაერთოთ იგი მე-2 მთვლელს ფიგ. 48.3 - ის მიხედვით.

ახლა გადავიდეთ დემიფრატორის დაპროგრამებაზე. ამ მიზნით FLProg პროგრამის ბლოკების ბიბლიოთეკის-Библиотека блоков ბიტებთან ოპერაციების-Операции с битами საქაღალდედან თავის მარცხენა დილაკის დაჭერით პროგრამის სამუშაო ზონაში უნდა გადმოვიტანოთ დემიფრატორის ბლოკი-Дешифратор, დავაწკაპოთ მასზე და მოვახდინოთ მისი პარამეტრიზაცია (ფიგ. 48.7), რომელიც ამ შემთხვევაში მდგომარეობს იმაში, რომ გამოსასვლელების რაოდენობის ველში-Количество выводов უნდა ჩავწეროთ ის რაოდენობა, რომელიც ჩვენ გვინტერესებს (8 ჩვენს შემთხვევაში).



ფიგ. 48.7 დემიფრატორის პარამეტრიზაციის ფანჯარა

ახლა საჭირო იქნება ტეგების ზონაში-Tags გავხსნათ გამოსასვლელების საქალაქო-Выходы და გამოსასვლელების დამატებაზე-Добавить выход თავის დილაკით დაჭერებით შევქმნათ გამოსასვლელები ჯერ პირველი დემიფრატორისთვის (B3) b1_1 ... b8_1 შემდეგ მეორე დემიფრატორისათვის (B6) b1_2 ... b8_2. შემდეგ ჩამოვიტანოთ ეს გამოსასვლელები და შეუერთოთ დემიფრატორების შესაბამის გამოსასვლელებს, ისე როგორც ეს ფიგ. 48.3 - ზეა ნაჩვენებია. ეს გამოსასვლელები შემდეგ შემზღუდავი წინააღმდეგობების გავლით შეერთებულ უნდა იქნას შუქდიოდების ანოდებთან, რომლის კათოდებიც უნდა იქნას შეერთებული არდუინოს მიწასთან (GND). აქ არ უნდა დაგვავიწყდეს ასევე დემიფრატორების შესასვლელების V შეერთება მთვლელების გამოსასვლელთან V, ხოლო ამ უკანასკნელთა შესასვლელები C გენერატორების გამოსასვლელთან Q.

პროგრამის გაშვების შემთხვევაში გენერატორები (B1 და B4) გამოიმუშავენ იმპულსების თანმიმდევრობას, რომლებიც მიეწოდება მთვლელების (B2 და B5) შესასვლელებს. ეს უკანასკნელნი დაიწყებენ ამ იმპულსების თვლას და თავიანთ გამოსასვლელებზე აფორმირებენ რიცხვით მნიშვნელობებს 0, 1, 2, 3, 4, 5, 6, 7, 8, 0, 1, 2, 3, 4, 5, 6, 7, 8 და ა.შ.შ. განუწყვეტლივ. ეს მნიშვნელობები თავის მხრივ გამოიწვევს დემიფრატორის გამოსასვლელებზე მორიგეობით ლოგიკური ერთიანების გამოიმუშავებას, რომლებიც მიეწოდება რა შუქდიოდებზე მოახდენს მათ მორიგეობით და თანმიმდევრობით ანთებას და ისეთი ეფექტი იქნება თითქოსდა გტანათებული წერტილები გადაადგილდება შუქდიოდების გასწვრივ თანდათანობით. აქ გასათვალისწინებელი იქნება ის გარემოება, რომ როცა მთვლელები განულდება მაშინ არცერთი შუქდიოდი ამ რვიდან არ იქნება ანთებული, აგრეთვე ის, რომ პირველ და მეორე დემიფრატორებთან შეერთებული შუქრიოდების ანთებების მოძრაობა იქნება ურთიერთსაწინააღმდეგო.

ახლა გადავიდეთ ორობითი მთვლელების შექმნაზე. სიმარტივისათვის დავაპროექტოთ ოთხთანრიგიანი მთვლელები რომლებიც მუშაობენ პირდაპირი და უკუ თვლის რეჟიმებში. ამ მიზნით პროგრამის ბლოკების ბიბლიოთეკის ტრიგერების საქალაქოდან-Триггеры გადმოვიტანოთ TT ტრიგერები, ოთხი პირდაპირი ორობითი მთვლელისთვის (B8 ... B11) და ოთხი შებრუნებული ორობითი მთვლელისთვის (B13 ... B16). შევართოთ ისინი ისე, როგორც ეს ფიგურაზე (ფიგ. 48.3)-ზეა ნაჩვენები, გარდა ამისა პირდაპირი მთვლელის შესასვლელები დავაინვერტიროთ.

ამის შემდეგ, ტეგების ზონაში ზემოთაღწერილის ანალოგიურად შევქმნათ კონტროლერების გამოსასვლელები 1r_1 ... 4r_1 პირველი ორობითი მთვლელისთვის და 1r_2 ... 4r_2 მეორე ორობითი მთვლელისთვის. ამ გამოსასვლელებს ერთის მხრივ შევუერთებთ

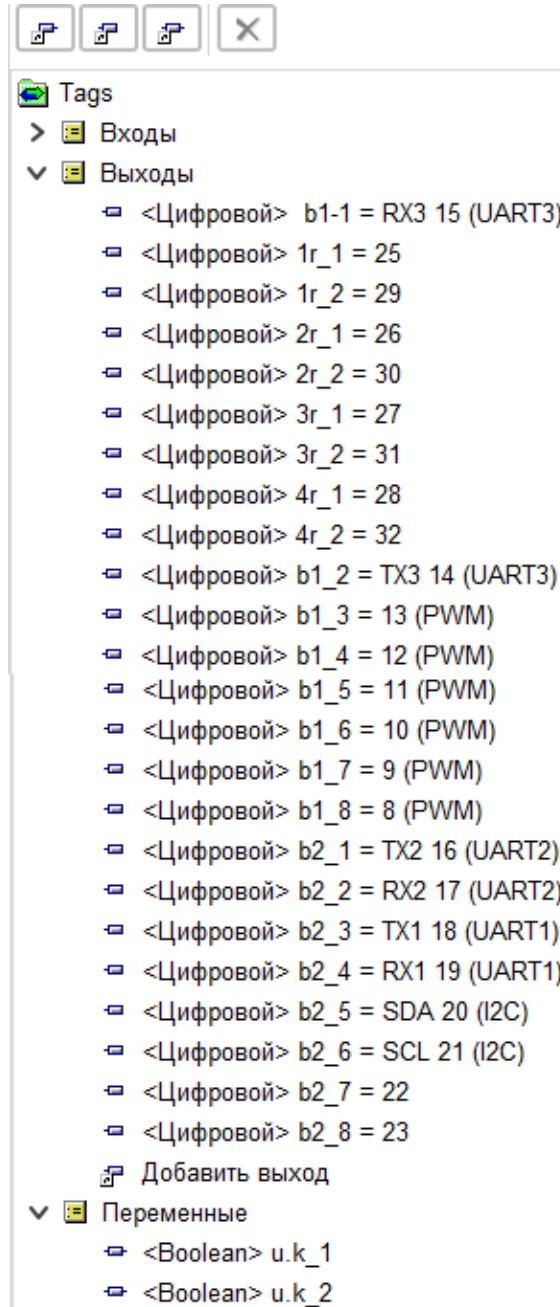
ტრიგერების გამოსასვლელებს, ხოლო მეორეს მხრივ შუქდიოდებს წინააღმდეგობების გავლით. თუ გაუშვებთ პროგრამას, მაშინ გენერატორები (B7 და B12) დაიწყებენ იმპულსების გენერირებას ხოლო მთვლელები მათ თვლას ორობით ფორმატში. იხ ცხრილი 1.

ცხრილი 1.

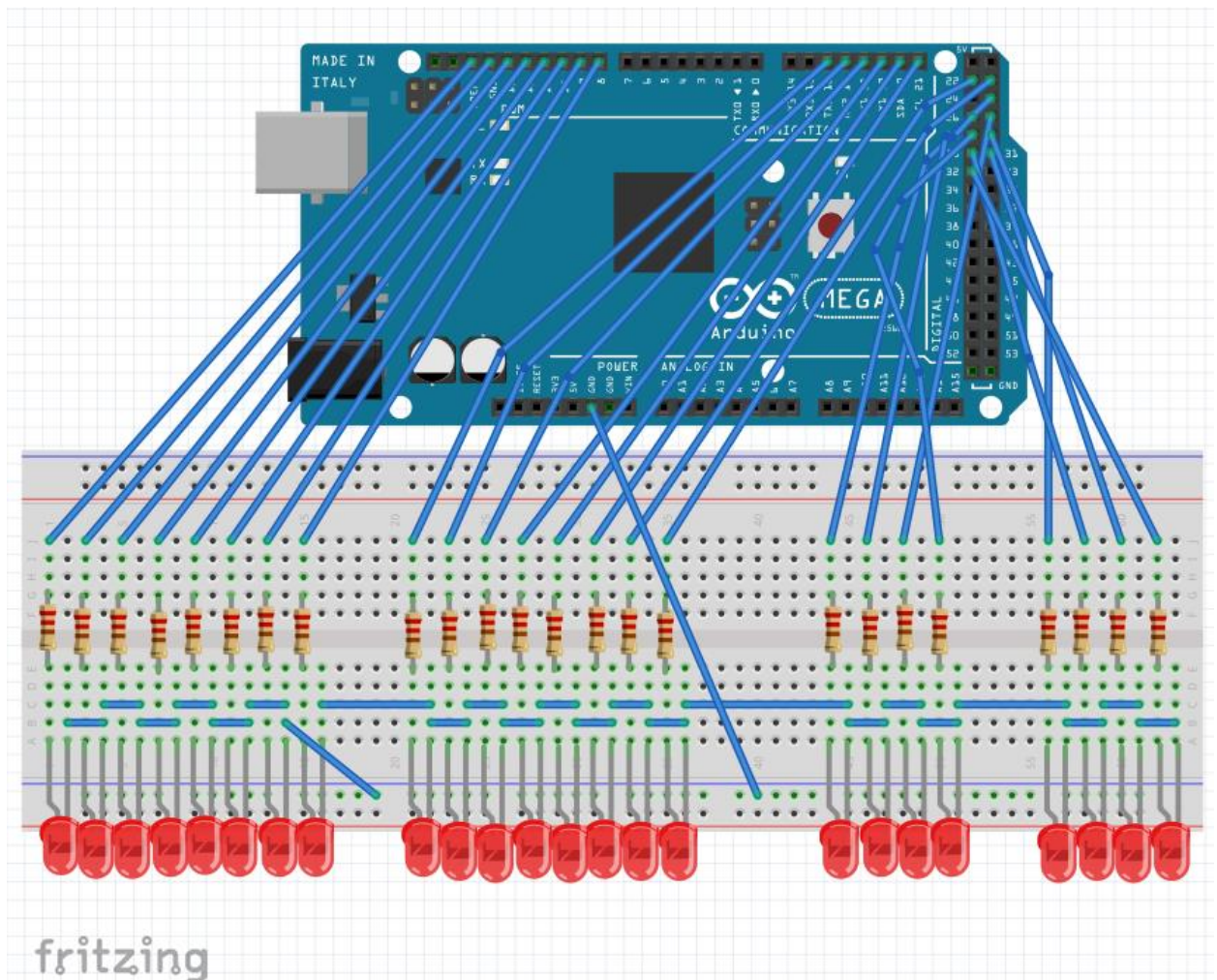
იმპულსის #	პირდაპირი მთვლელები				ათობითი ექვივალენტი	შებრუნებული მთვლელები				ათობითი ექვივალენტი
	4r_1	3r_1	2r_1	1r_1		4r_2	3r_2	2r_2	1r_2	
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	15
2	0	0	1	0	2	1	1	1	0	14
3	0	0	1	1	3	1	1	0	1	13
4	0	1	0	0	4	1	1	0	0	12
5	0	1	0	1	5	1	0	1	1	11
6	0	1	1	0	6	1	0	1	0	10
7	0	1	1	1	7	1	0	0	1	9
8	1	0	0	0	8	1	0	0	0	8
9	1	0	0	1	9	0	1	1	1	7
10	1	0	1	0	10	0	1	1	0	6
11	1	0	1	1	11	0	1	0	1	5
12	1	1	0	0	12	0	1	0	0	4
13	1	1	0	1	13	0	0	1	1	3
14	1	1	1	0	14	0	0	1	0	2
15	1	1	1	1	15	0	0	0	1	1
16	0	0	0	0	0	0	0	0	0	0

17	0	0	0	1	1	1	1	1	1	15
----	---	---	---	---	---	---	---	---	---	----

ფიგ. 48.8-ზე ნაჩვენებია პროექტის ტეგების ზონის ფანჯარა, ხოლო ფიგ. 48.9-ზე - იგივე პროექტის Fritzing სქემა.



ფიგ. 48.8 პროექტის ტეგების ზონა



ფიგ. 48.9 პროექტის ბლოკსქემა Fritzing პროგრამაში

პროექტი_49 SD ბარათი

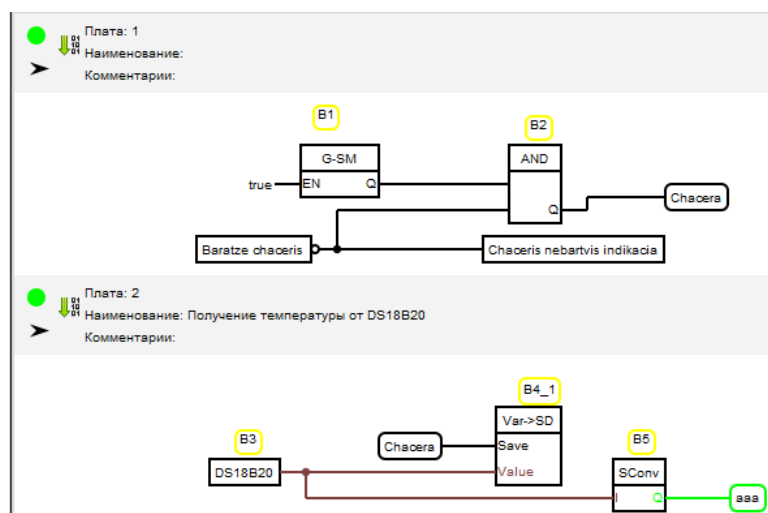


დასაწყისში რაც უნდა გავაკეთოთ არის ის, რომ გავუმვათ FLProg-ის პროგრამა, დავაწკაპოთ მთავარი მენიუს ფაილის მენიუზე-Файл, ჩამოშლადი სიიდან ავირჩიოთ ახალი პროექტი-Новый და გამოჩენილ ფანჯარაში დაპროგრამების ენად-Язык программирования ავირჩიოთ FBD, ხოლო კონტროლერის ველში ... დილაკზე დაჭერის შედეგად კონტროლერი-Контроллер: ჩამოშლად სიაში ვირჩევთ იმ კონტროლერს, რომელიც გვაქვს ხელთ-Arduino Uno

ჩვენ შემთხვევაში. ამის შემდეგ კი ვაჭერთ არჩევის ღილაკს-ВЫБРАТЬ, შემდეგ კი მზადყოფნის ღილაკს-ГОТОВО.

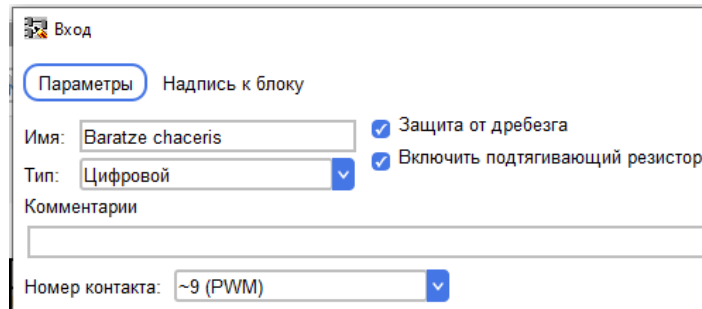
ჩვენს მიერ დასამუშავებელი პროექტის ამოცანა მდგომარეობს შემდეგში. ვთქვათ საჭიროა ოთხი სხვადასხვა ფიზიკური სიდიდის გაზომვა: ორი ტემპერატურის სხვადასხვა წერტილში, აგრეთვე ატმოსფერული წნევისა და სიმაღლის ზღვის გაზომვა დონიდან. გაზომვა უნდა მოხდეს გარე ღილაკზე თითის დაჭერის მომენტში, გაზომილი სიდიდეების შენახვა უნდა ხდებოდეს SD ბარათზე ბარათის წამკითხავით-CardReader, ისე რომ შემდეგში შესაძლებელი იყოს ამ ბარათიდან ჩაწერილი ინფორმაციის წაკითხვა კომპიუტერის საშუალებით. პარალელურად გაზომილი ინფორმაციის ასახვა უნდა ხდებოდეს დისპლეის- HD44780 ჩიპზე დახმარებით.

ამ ალგორითმის რეალიზაციის პროგრამა FLProg გარემოში განლაგებულია მის ოთხ დაფაზე ფიგ. 49_1 დაფა 1, 2 (Плата :1, Плата :2) და ფიგ. 49_7 დაფა 3,4 (Плата :3, Плата :4).



ფიგ. 49_1 დაფა 1 და დაფა 2

პროგრამა მუშაობს შემდეგნაირად: პირველ დაფაზე განლაგებულია გენერატორის ბლოკი B1, რომელიც დაპროგრამებულია სიმეტრიული მულტივიზრატორის რეჟიმში სამუშაოდ 5 წამის იმპულსის ხანგრძლიობებით გამომუშავებისათვის. ამავე დაფაზე ბარათზე ჩაწერის ნებართვის-Baratzze chaoeris დისკრეტული ერთი შესასვლელი ინვერსირებით, რომლის პარამეტრიზაციის ფანჯარაც ასახულია ფიგ. 49_2-ზე.

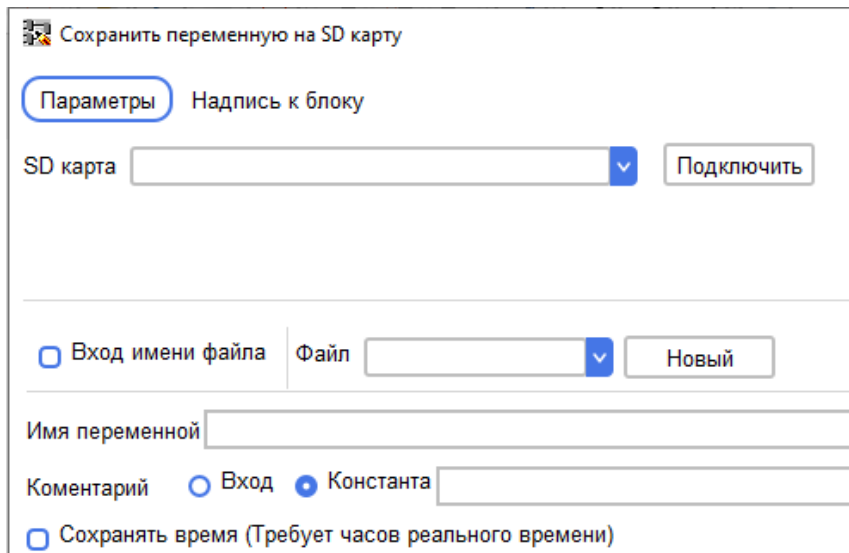


ფიგ. 49_2 ბარათზე ჩაწერის შესასვლელის პარამეტრიზაციის ფანჯარა

აქვეა განთავსებული ლოგიკური გამრავლების ბლოკი B2. ბარათზე ჩაწერის შესასვლელზე (არდუინოს მე-9 პინი) მიერთებულია გარე დილაკი, რომელზეც თითის დაჭერის შემთხვევაში ამ ბლოკის მე-2 შესასვლელზე დაჯდება ლოგიკური ერთიანი და გაატარებს მისსავე პირველ შესასვლელზე მიერთებული გენერატორის იმპულსებს. ეს იმპულსები, ანუ ჩაწერის ნებართვის ინდიკაციის იმპულსები ერთის მხრივ გადიან არდუინოს მე-7 პინზე და შემდეგ შუქდიოდზე სტანდარტული შეერთების სქემით და გვიჩვენებენ SD მეხსიერებაში მონაცემის ჩაწერაზე ნებართვის იმპულსებს-Chaceraze nebartvis indikacia. აქვეა გამოსახული ტეგების ზონაში შექმნილი ცვლადი Chacera, რომელიც შეერთებულია მეორე ბლოკის გამოსასვლელზე და გამოიყენება შემდეგ დაფებზე. იგი შეერთებულია SD მეხსიერების ბლოკის შენახვის-Save შესასვლელებთან (B4_1,2,3,4) და მასში ფორმირებული იმპულსების წინა ფრონტებზე ახორციელებს CD მეხსიერებაში მონაცემების ჩაწერას. კერძოდ, ამ მეხსიერებაში ჩაიწერება: მე-2 დაფაზე განთავსებული DS18B20 (ბლოკი B3) ტემპერატურული გადამწოდის მიერ გაზომილი სიდიდე, მე-3 დაფაზე განთავსებული BMP085 წნევის გადამწოდის (ბლოკი B6) მიერ გაზომილ სამ სხვადასხვა სიდიდეს. DS18B20 გადამწოდის გამოსასვლელი კონვერტაციის ბლოკის B5 გავლით ქმნის ცვლადს სახელწოდებით aaa, რომელიც შემდეგ გადადის მე-3 დაფაზე დისპლეიზე ტემპერატურის სიდიდის ასახვისათვის. ამ გადამწოდის ბლოკის პარამეტრიზაციის ფანჯარა ასახულია ფიგ. 49_3-ზე.

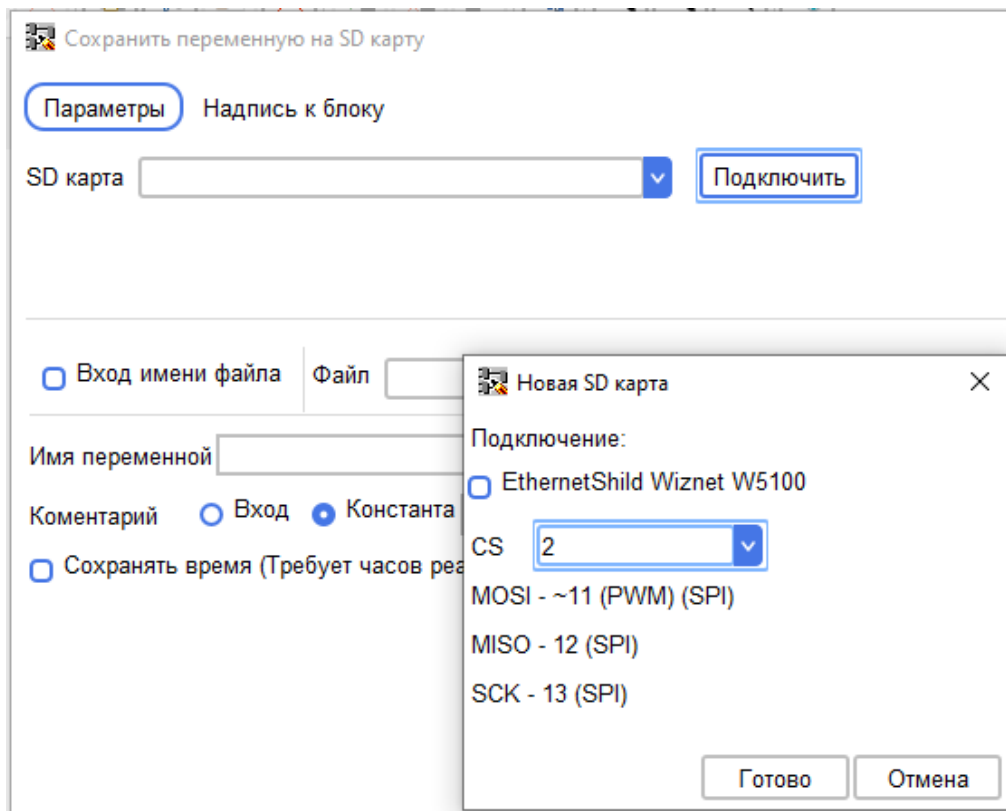
ფიგ. 49_3 DS18B20 გადამწოდის პარამეტრიზაციის ფანჯარა

ბლოკის, სახელწოდებით ცვლადის შენახვა CD ბარათზე-Сохранить переменную на CD карту, რომელიც მოთავსებულია FLProg-ის ბლოკების ბიბლიოთეკის საქალაქში SD Карта-დან იშნულეება CD ბარათის ლოგ. ფაილში ცვლადის ჩაწერის უზრუნველყოფა, რომელიც მიერთებულია არდუინოსთან სპეციალური შილდის საშუალებით. რადგან ეს ბლოკი დავამატეთ პროექტში, მაშინ აუცილებელია მისი პარამეტრიზაცია. მოდულის თვისებებში, თუ კი პროექტში ადრეც არსად არ გამოიყენებოდა CD ბარათი, აუცილებელია ის მიერთებულ იქნას არდუინოსთან. გადმოვიტანოთ ბლოკი პროგრამის სამუშაო ზონაში და დავაწკაპუნოთ მასზედ თავის მარცხენა ღილაკით. ეკრანზე გამოვა ბლოკის პარამეტრიზაციის ფანჯარა (ფიგ. 49_4).



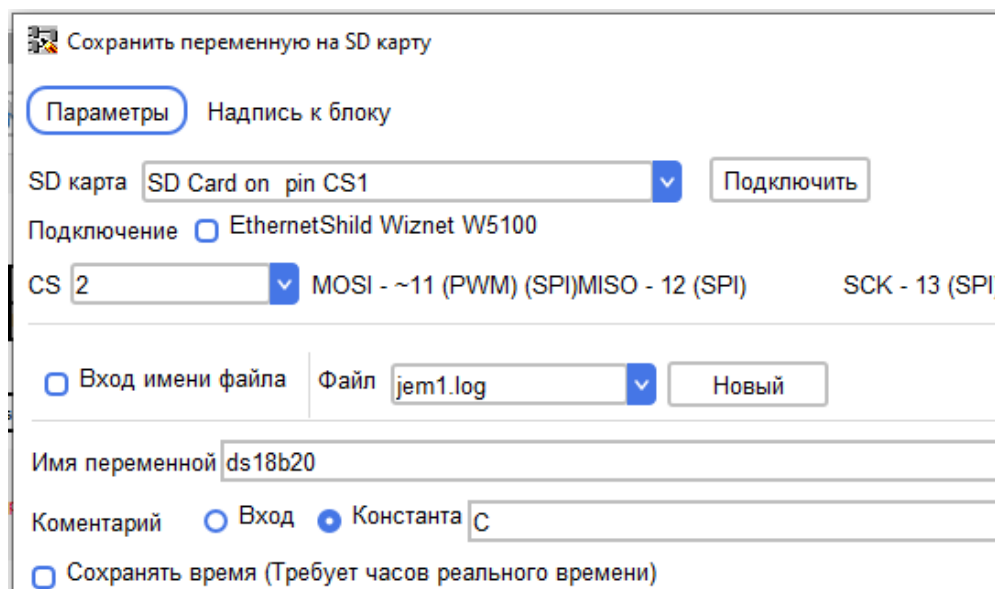
ფიგ. 49_4 ცვლადის შენახვის ბლოკის პარამეტრიზაციის ფანჯარა SD ბარათზე

დავაჭიროთ მიერთების ღილაკს-Подключить და გამოსულ დამატებით ფანჯარაში (ფიგ. 49_5) CS ველში ჩავწეროთ არდუინოს პინის ნომერი, რომელიც უნდა მიუერთდეს მოდულის CS შესასვლელს, ჩვენს შემთხვევაში ვწერთ 2. აქვე ჩანსა, რომ მოდულის MOSI, MISO და SCK კონტაქტების შეერთებები პროგრამულად არის განსაზღვრული და ისინი უნდა შეუერთდეს არდუინოს მე-11, მე-12 და მე-13 პინებს შესაბამისად.



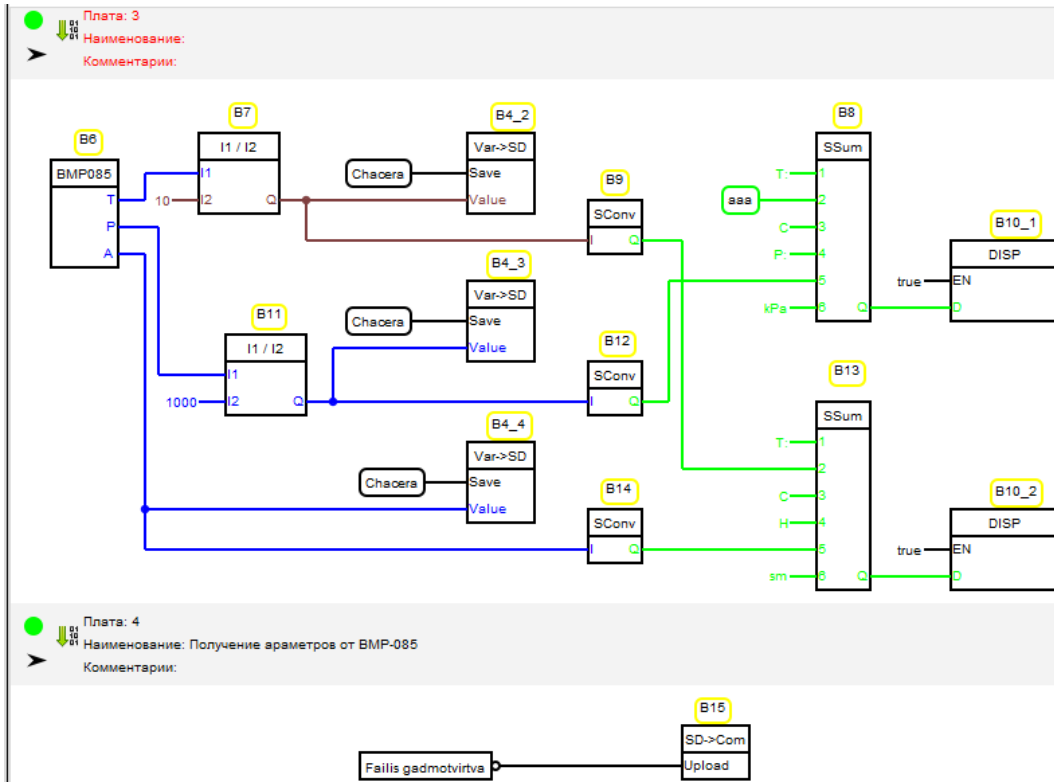
ფიგ. 49_5 SD ბარათის არდუინოსთან მიერთების ფანჯარა

დავაჭიროთ მზადდონის ღილაკს-Готово და ახლად გამოსულ ფანჯარაში ავირჩიოთ უკვე შექმნილი ფაილი ცვლადის ჩაწერისათვის ან უნდა შევქმნათ ახალი ფაილი და შევიყვანოთ ცვლადის სახელი. ჩვენს შემთხვევაში შევქმენით ახალი ცვლადის ფაილი სახელწოდებით - jem1 (ფიგ. 49_6).



ფიგ. 49_6 ფაილის სახელის დარქმევა

გადავიდეთ ახლა მე-3 დაფის აღწერაზე. როგორც ვხედავთ აქ წარმოდგენილია ატმოსფერული წნევის გადამწოდი BMP-085 (ბლოკი B6) (ფიგ. 49_7), რომელსაც შეუძლია ერთდროულად გაზომოს სამი პარამეტრი, ატმოსფერული ტემპერატურა, ატმოსფერული წნევა და სიმაღლე ზღვის დონიდან. აღნიშნულ ბლოკს ჭირდება პარამეტრიზაცია, რომლის ფანჯარაც წარმოდგენილია ფიგ. 49_8-ზე. როგორც პარამეტრიზაციის ფანჯარიდან ჩანს, გადამწოდის გამომყვანების შეერთებები არდუინოს პინებთან პროგრამულად არის განსაზღვრული და წარმოადგენენ GND და VIN გამოსასვლელებს, რომლებიც შეერთებულია არდუინოს GND და +5 ვ პინებთან შესაბამისად, ხოლო SDA და SCL სასიგნალო გამომყვანები შეერთებულია არდუინოს A4 და A5 პინებთან. გარდა ამისა გადამწოდის პარამეტრიზაციის ფანჯარაში უნდა მოვნიშნოთ სამივე პარამეტრის ალაში: წნევის გამოსასვლელი-Выход давления (Pa), ზღვის დონიდან სიმაღლის გამოსასვლელი-Выход Высоты над уровнем моря (см) და ტემპერატურის გამოსასვლელი-Выход температуры (C*10).



ფიგ. 49_7 დაფა 3 და დაფა 4

Датчик давления BMP-085

Параметры Надпись к блоку

Подключение датчика

GND - GND	SDA - A4 (I2C)
VIN - +5V	SCL - A5 (I2C)

Выход давления (Pa)

Выход высоты над уровнем моря (см)

Выход температуры (C*10)

Калибровка

При каждом перезапуске контроллера

По команде на входе

Значение

Давление (Pa)

Высота над уровнем моря (см)

Вход

Константа

Вход разрешения опроса

Время опроса датчика

В каждом цикле (не рекомендуется)

Периодически

Опрашивать каждые сек.

ფიგ. 49_8 BMP-089 გადამწოდის ბლოკის პარამეტრიზაციის ფანჯარა

აქვეა მონიშნული დაკალიბრების ალამი-Калибринка და ჩართულია ჩამრთველი თუ როდის უნდა მოხდეს გადამწოდის დაკალიბრებაა და მითითებულია კონტროლერის ყოველ გადატვირთვის დროს-При каждом перезапуске контроллера. აქვეა ჩართული ჩამრთველი-Высота над уровнем моря (см) და ჩართულია აგრეთვე ჩამრთველი კონსტანტა და შესაბამის ველში ჩაწერილია რიცხვი 300 რაც უჩვენებს იმ ადგილმდებარეობის სიმაღლეს ზღვის დონიდან სადაც წარმოებს გაზომვები (ჩვენს შემთხვევაში თბილისის მდგომარეობა). ამავე ფანჯარაშია ჩართული გადამწოდის გამოკითხვის დრო-Время опроса датчика და მითითებულია დრო 6 წამი, რაც უჩვენებს იმ პერიოდულობას, რა ტემპითაც უნდა მოხდეს გადამწოდის ამოკითხვა.

ახლა დავუბრუნდეთ ისევ ჩვენს პროგრამას ფიგ. 49_7. როგორც ნახაზიდან ჩანს SD მეხსიერების ბლოკი პირობითად დაყოფილია ოთხ ნაწილად B4_1, B4_2, B4_3, B4_4. აქედან პირველი ნაწილი B4_1 გამოყენებულია DS18B20 გადამწოდის ჩვენების ასახვისათვის. BMP-089 გადამწოდის ჩვენებების ასახვისათვის გამოყენებული გვაქვს CD ბარათის B4_2, B4_3, B4_4 ნაწილები. რომელთა Value შესასვლელზე მიეწოდება გადამწოდის მიერ გაზომილი პარამეტრების მნიშვნელობები, მაგრამ ტემპერატურის სიდიდე T მიეწოდება გაყოფილი 10-ზე ბლოკით B7, წნევის სიდიდე P-გაყოფილი 1000 ბლოკით B11 და სიმაღლის ზღვის დონიდან A უშუალოდ, მათემატიკური გარდაქმნის გარეშე.

თუ კი ჩვენ ჩავთვირთავთ აღნიშნულ პროგრამას არდუინოში და დავაწვებით Baratze Chaceris გარე ღილაკს თითოს, მაშინ თითოს დაჭერის განმავლობაში აინთება Chaceris nebartvis indikaciis შუქდიოდი, რომელიც ანალოგიური დასახელების არდუინოს პინზეა მიერთებული (ფიგ. 49_1), გარდა ამისა ლოგიკური გამრავლების ბლოკის B2-ის გამოსასვლელზე გამომუშავდება ცვლადი სახელწოდებით Chacera იმპულსების სახით და ეს მოხდება ყოველ 6 წამში ერთჯერ ღილაკზე თითოს დაჭერის განმავლობაში. ეს ჩაწერის სიგნალი Chacera გადადის ჯერ მეორე დაფაზე და მიეწოდება SD ბლოკის პირველი B4_1 ნაწილის Save შესასვლელს და გარდა ამისა გადადის მე-3 და მე-4 დაფებზე (ფიგ. 49_7) და მიეწოდება SD ბარათის დანარჩენ B4_2, B4_3, B4_4 ნაწილების Save შესასვლელს. ამ შესასვლელებზე ყოველი იმპულსის წინა ფრონტის დადგომის მომენტებში მოხდება მათ Value შესასვლელზე არსებული მონაცემების ჩაწერა და შენახვა SD მეხსიერების დაფაში. თუ დავაჭერთ ღილაკს თითოს დაახლოვებით ნახევარი წუთის განმავლობაში, მაშინ მეხსიერებაში ჩაიწერება 5,6 ანათვალის გაზომილი პარამეტრებისა და მოხდება მისი შენახვა.

ახლა, თუ ამოვიღებთ ჩვენ SD მეხსიერების დაფას და ჩავდგავთ კომპიუტერში, მაშინ ჩვენ გვექნება შესაძლებლობა წავიკითხოთ ამ მეხსიერებაში ჩაწერილი ინფორმაცია

სტანდარტული მეთოდებით. ფიგ. 49_9-ზე ასახულია ფრაგმენტი SD მეხსიერების ბარათში ჩაწერილი ინფორმაციისა.

```
JEM1.LOG - Notepad
File Edit Format View Help
|ds18b20t;Float;29-9-2016 22:32:26;20.88;C;
bmp085t;Float;;20.80; C;
bmp085p;Integer;;96;kPa;
bmp085a;Integer;;219;cm;
ds18b20t;Float;29-9-2016 22:32:28;20.88;C;
bmp085t;Float;;20.90; C;
bmp085p;Integer;;96;kPa;
bmp085a;Integer;;237;cm;
ds18b20t;Float;29-9-2016 22:32:31;20.88;C;
bmp085t;Float;;20.90; C;
bmp085p;Integer;;96;kPa;
bmp085a;Integer;;237;cm;
ds18b20t;Float;29-9-2016 22:32:33;20.88;C;
bmp085t;Float;;20.90; C;
bmp085p;Integer;;96;kPa;
bmp085a;Integer;;237;cm;
ds18b20t;Float;29-9-2016 22:32:35;20.88;C;
bmp085t;Float;;20.90; C;
```

ფიგ. 49-9 DS მეხსიერების ბარათში ჩაწერილი ინფორმაციის ფრაგმენტი

პროექტი იძლევა იმისა საშუალებას, რომ ის იმფორმაცია, რომელიც განიცდის ჩაწერას CD მეხსიერებაში ამავედროულად ასახულ იყოს ორსტრიქონიან თექვსმეტანრიგან დისპლეი HD44780 ჩიპზე. ამ მიზნით დისპლეის ბლოკი ჩიპზე HD44780 ბლოკების ბიბლიოთეკის დისპლეების საქალაქდგან გადმოტანილია პროგრამის სამუშაო ზონაში და წარმოდგენილია ორი ქვებლოკის B10_1 და B10_2 სახით პირველი და მეორე სტრიქონების ასახვისათვის. პირველი ქვებლოკის პარამეტრიზაციის ფანჯარა წარმოდგენილია ფიგ. 49_10-ზე, მეორე ქვებლოკის პარამეტრიზაციის ფანჯარაც ზუსტად ანალოგიურია პირველი ქვებლოკის ფანჯრის, იმ განსხვავებით, რომ სტრიქონის ველში-Строка ერთის მაგივრად ჩაწერილია ორი.

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/>	Комментарий	<input type="text"/>	

Данные:

Вход

Константа

Строка:

Столбец:

ფიგ. 49_10 დისპლეის პირველი ქვებლოკის პარამეტრიზაციის ფანჯარა

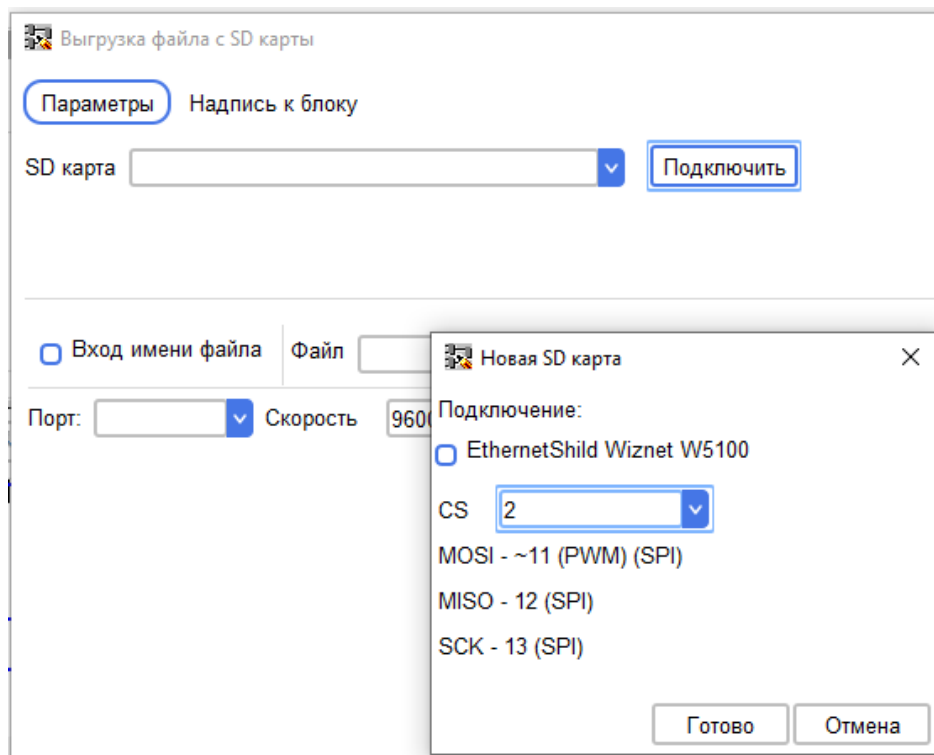
ჩვენ გვაქვს ასასახი ოთხი პარამეტრი, დავაჯგუთოთ ისინი შემდეგნაირად: დისპლეის პირველ სტრიქონში აისახოს ტემპერატურის მნიშვნელობა DS18B20 გადამწოდებიდან და წნევის მნიშვნელობა BMP-085 გადამწოდებიდან, ხოლო მეორე სტრიქონში - ტემპერატურის მნიშვნელობა BMP-085 გადამწოდებიდან და სიმაღლის მნიშვნელობა ზღვის დონიდან - იგივე გადამწოდებიდან. ამ მიზნით დავგჭირდება კონვერტაციის ბლოკების გამოყენება: B5, B9, B12, B14 და ორი სტრიქონების შეკრების ბლოკის გამოყენება ექვს-ექვსი შესასვლელით თვითოეულ ბლოკზე, ესენია: B8, B13. B8 ბლოკის ექვსი შესასვლელი დათვითულია შემდეგნაირად: პირველ შესასვლელზე ჩაწერილი გვაქვს ტემპერატურის აღნიშვნა T და პრაბელი, შემდეგ DS18B20 გადამწოდებიდან გაზომილი ტემპერატურის სიდიდე, რომელიც მიეწოდება aaa ცვლადის სახით და შემდეგ ტემპერატურის განზომილება-ლათინური ასო C, შემდეგ ისევ პრაბელი და წნევის აღმნიშვნელი P და პრაბელი, შემდეგ წნევის გაზომილი სიდიდე B9 კონვერტაციის ბლოკის გავლით, შემდეგ ისევ პრაბელი და წნევის განზომილება kPa.

ანალოგიურად დავტვირთავთ მეორე სტრიქონების შეკრების ბლოკსაც B13. მის პირველ შესასვლელზე ჩავწერთ ტემპერატურის მნიშვნელობას T და პრაბელს, მეორე შესასვლელზე მივიყვანთ გასაზომი ტემპერატურის სიდიდეს BMP-085 გადამწოდებიდან B12 კონვერტაციის ბლოკის გავლით და შემდეგ პრაბელსა და ტემპერატურის განზომილებას C და

პრაბელს, შემდეგ სიმაღლის აღმნიშვნელს H და პრაბელს, შემდეგ სიმაღლის გასაზომ სიდიდეს B14 კონვერტაციის ბლოკის სახით და პრაბელს და ბოლოს განზომილებას sm. B8 და B13 სტრიქონების შეკრების ბლოკების გამოსასვლელები რათქმაუნდა უნდა შევუერთოთ B10_1 და B10_2 დისპლეის ქვებლოკების ანალოგურ შესასვლელებს O, ავაწყობთ დისპლეის ფიგ. 49_10-ის მიხედვით, გაუშვებთ არდუინოს და დისპლეის სტრიქონებში აისახება გასაზომი პარამეტრების მნიშვნელობები.

ფაილის გადმოტვირთვა

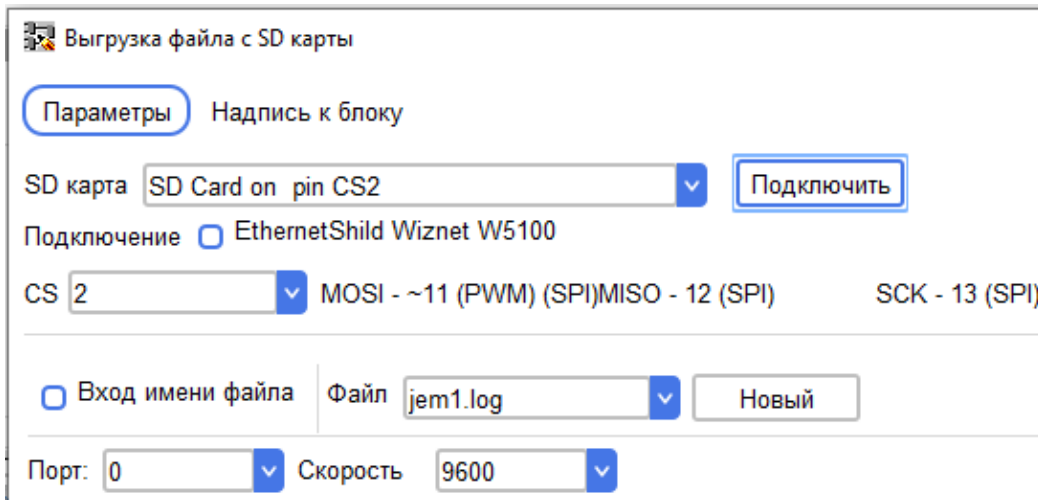
ახლა პროგრამის ბლოკების ბიბლიოთეკის CD ბარათის საქალაქედან-SD карта სამუშაო ზონაში გადმოვიტანოთ CD ბარათიდან ფაილის გადმოტვირთვის ბლოკი-Выгрузка файла с CD карты. ამ ბლოკის დანიშნულებაა CD ბარათში ჩაწერილი ფაილის შემცველობის გაგზავნა Com-პორტში. პროექტში ამ ბლოკის დამატების შემთხვევაში აუცილებელი ხდება მისი პარამეტრიზაცია (ფიგ. 49_11). მოდულის თვისებებში აუცილებელია მისი მიერთება, თუ კი ადრე პროექტში CD ბარათი არსად გამოყენებული არ ყოფილა.



ფიგ. 49_11 SD ბარათიდან ფაილის გადმოტვირთვის ბლოკის პარამეტრიზაციის ფანჯარა

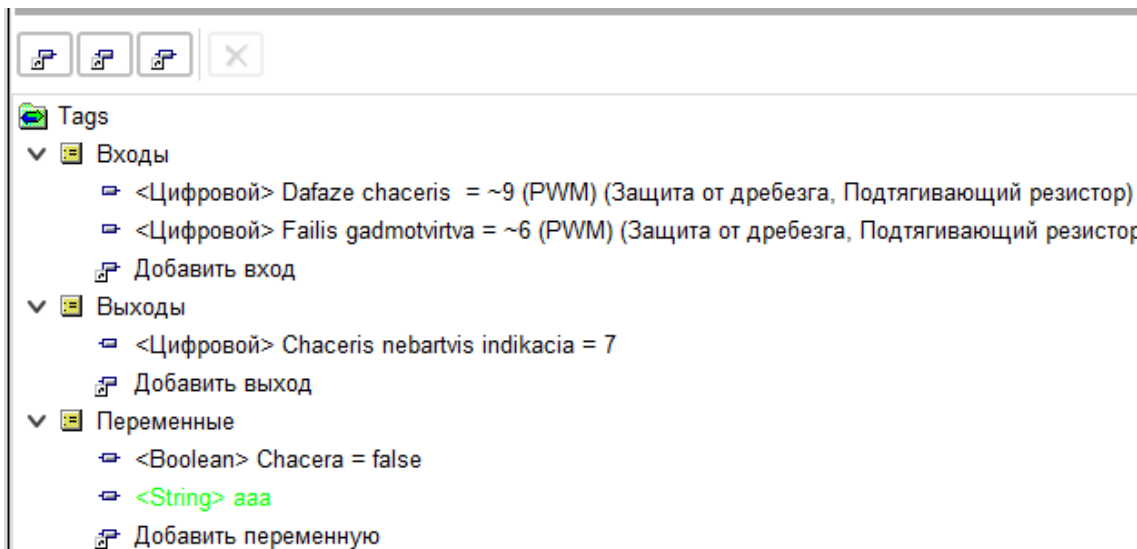
CD ბარათის შექმნის დიალოგში აუცილებელი იქნება არჩეულ იქნას კონტროლერის ის პინი, რომელზეც უნდა შეუერთდეს ბარათის წამკითხველის მართვის შესასვლელი. შემდეგ საჭიროა არჩეულ იქნას ის ფაილი პროექტში არსებული ფაილებიდან რომელიც საჭიროა

გადმოტვირთულ იქნას, ანდა უნდა შექმნილ იქნას ახალი ფაილი. შემდეგ არჩეულ უნდა იქნას პორტის ნომერი და მისი მუშაობის სიჩქარე.

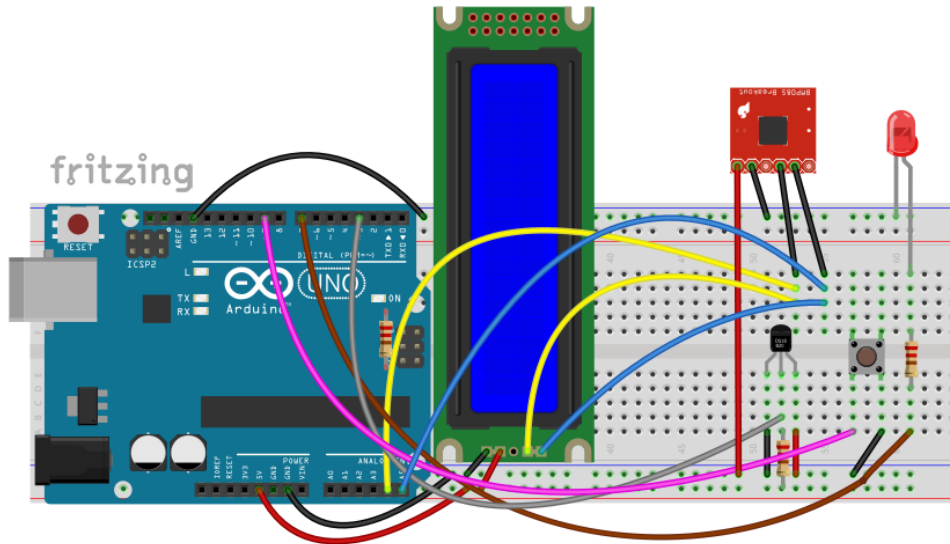


ფიგ. 49_12 გადმოსატვირთვი ფაილის, პორტის ნომერის და სიჩქარის არჩევის ფანჯარა

ბლოკის შესასვლელზე „Upload“ იმპულსის წინა ფრონტის დადგომის შემთხვევაში ფაილის შემცველობა გადაგზავნილ იქნება Com-პორტში. თუ ფაილის პოვნა ვერ მოხერხდა, მაშინ პორტში არ გაიგზავნება არაფერი. პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 49_13-ზე. პროექტის Fritzing სქემა ასახულია კი ფიგ. 49_14-ზე.

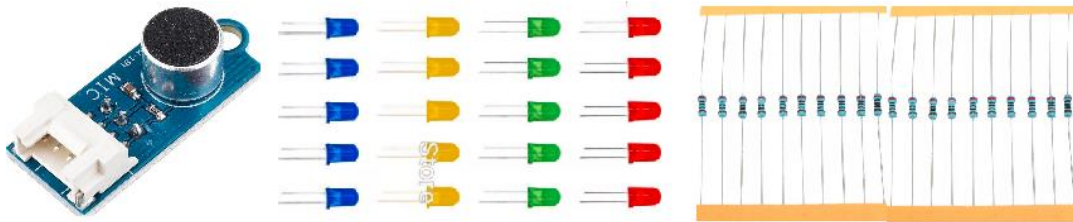


ფიგ. 49_13 პროექტის ტეგების ზონის ფანჯარა

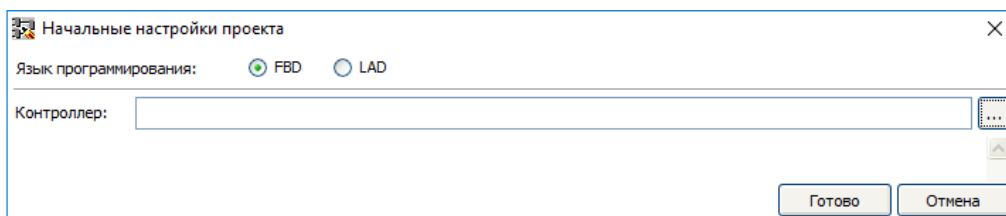


ფიგ. 4_14 პროექტის Fritzing სქემა

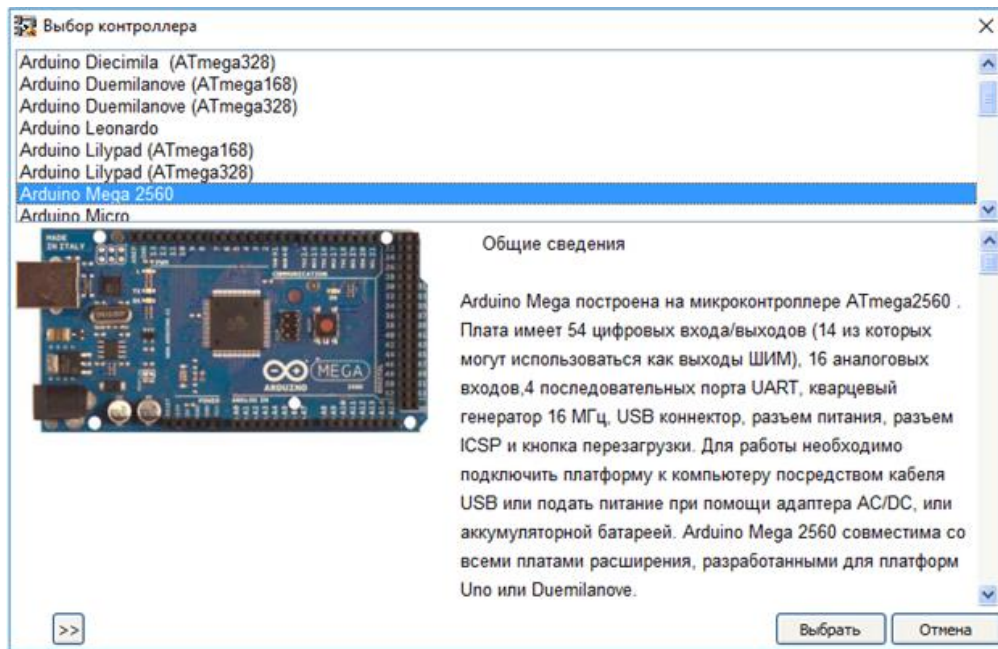
პროექტი_50 ექვალაიზერი



გავუშვით FLProg-ის პროგრამა, დავაწკაპოთ მთავარი მენიუს ფაილზე-Файл, ჩამოშლადი სიიდან ავირჩიოთ ახალი-Новый და გამოჩენილ ფანჯარაში (ფიგ. 50.1) დაპროგრამების ენად-Язык программирования ავირჩიოთ FBD. ხოლო კონტროლერის ველში ... ლილაკზე დაჭერის შედეგად კონტროლერის-Контроллер: ჩამოშლად სიაში ვირჩევთ იმ კონტროლერს, რომელიც გვჭირდება ამ პროექტში-Arduino Mega2560 ჩვენ შემთხვევაში (ფიგ. 50.2). ამის შემდეგ კი უნდა დავაჭიროთ არჩევის ლილაკს-Выбрать, შემდეგ კი მზადდონის ლილაკს-Готово და ამით პროგრამის საწყისი გაწყობა დამთავრებული იქნება.



ფიგ. 50.1 პროექტის საწყისი გაწყობის ფანჯარა



ფიგ. 50.2 კონტროლერის არჩევის ფანჯარა

პროექტში განხილულია ეკვალაიზერის ერთერთი ვარიანტი, მაგრამ სანამ დავიწყებდეთ უშუალოდ თემაზე საუბარს იქამდე გავაკეთოთ მცირე შესავალი ამ საკითხთან დაკავშირებით. ბოლო 10-15 წელია რაც შეიქმნა მიკროპროცესორული ტექნიკის ერთ-ერთი მიმდინარეობა მიკროკონტროლერები, რომლებიც ფართოდ გამოიყენება ჩვენს ყოფაცხოვრებაში. ეს მოწყობილობები თავისთავში მოიცავს ორ ერთმანეთისგან განსხვავებულ სპეციალიზაციას. ერთი არის ელექტრონიკა ხოლო მეორე - დაპროგრამირება. ეხლა მეტნაწილად ეს ორი მიმდინარეობა გაერთიანდა და ერთ სისტემაში მოეწყო, შესაბამისად ამისა მასთან სამუშაოდ გვჭირდება როგორც ელექტრონიკის ცოდნა ასევე დაპროგრამების ცოდნაც, მაგრამ რეალურად რთულია ისეთი სპეციალისტების მოძიება რომლებსაც ორივე საკითხში ექნებათ სათანადო ცოდნა და გამოცდილება.

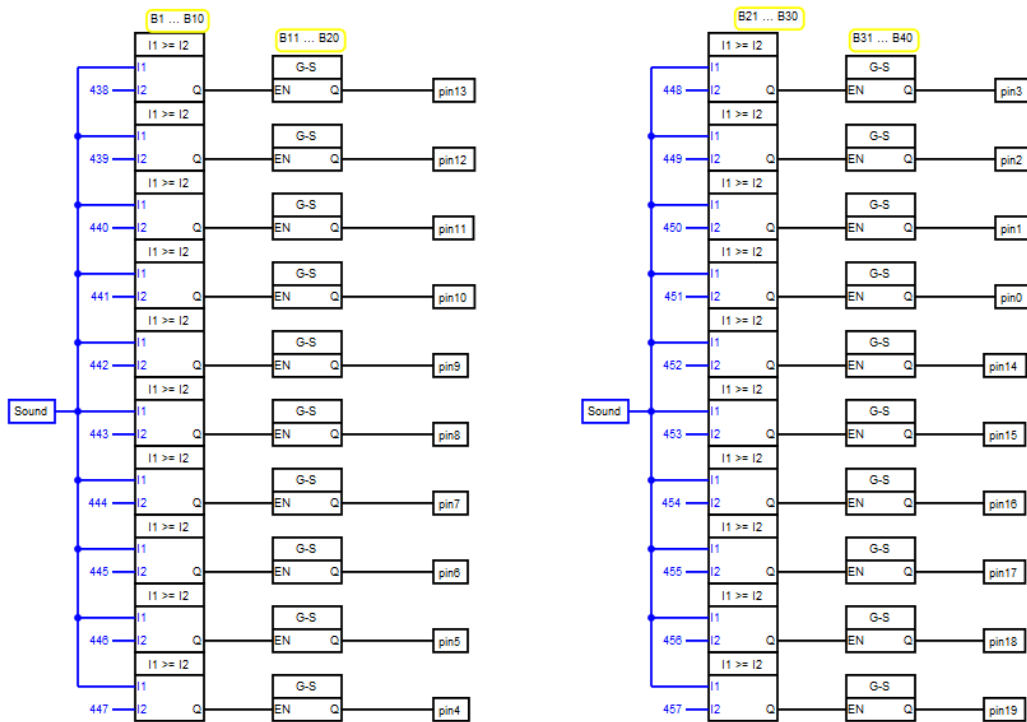
აქედან გამომდინარე, აუცილებელი გახდა არდუინოსა და მისი მსგავსი ტექნოლოგიების შექმნა რათა პროგრამისტებისთვის გაეადვილებინათ სქემებთან მუშაობა. მაგრამ ინჟინრისთვის ისევ დიდ სირთულეს წარმოადგენდა ტექნოლოგიის ტექსტური სახით დაპროგრამირება. იმისთვის რომ ინჟინრებისთვისაც გაეადვილებინათ არდუინოსთან მუშაობა შეიქმნა გრაფიკული დაპროგრამების ენები მაგრამ მათაც ვერ პოვეს სათანადო განვითარება რადგან ვერ აკმაყოფილებდნენ ინჟინრისთვის საჭირო მოთხოვნებს. ბოლო რამდენიმე წლის განმავლობაში წარმოიშვა და დღესაც განაგრძობს განვითარებას ახალი, ვიზუალური დაპროგრამების გარემო FLProg, რომელიც განირჩევა არსებული ვიზუალური

დაპროგრამების გარემოებებიდან მთელი რიგი უპირატესობებით და ამიტომ ძალიან ფართო გამოყენება ჰპოვა ამ სფეროში.

აღნიშნული ტექნოლოგია ორიენტირებულია ელექტრიკოსებზე, ელექტრონიკოსებზე, სქემოტექნიკოსებზე, სისტემოტექნიკოსებზე, რადიომოყვარულებზე, გამომგონებლებზე, მართვის ავტომატიზირებული სისტემების სპეციალისტებზე, რობოტოტექნიკოსებზე, მექატრონიკოსებზე ანუ სპეციალისტებზე, რომლებიც თავისი კვალიფიკაციით არ არიან პროგრამისტები. მაგრამ სწორედ რომ ეს ახალი ტექნოლოგია, ანუ FLProg ვიზუალურ/გრაფიკული გარამო აძლევს მათ საშუალებას შეასრულონ ის სამუშაოები რაც ადრე მათ არ შეეძლოთ შესრულებინათ, რადგან ჩვეულებრივად, მათ არ იცოდენ დაპროგრამების სტანდარტული უნივერსალური ენები. ახლა უკვე მათ საშუალება ეძლევათ დააპროექტონ ელექტრონული სქემები, გაჯეტები, ანუ ხაზონ ელექტრონული და ლოგიკური სქემები FLProg გარემოში, ე.ი. შეასრულონ ის სამუშაოები რაშიც არიან პროფესიონალები, ხოლო ყველა სხვა სამუშაოებს ანუ კომპილაციას, პროგრამული კოდის მიღებას, მის ჩატვირთვას მიკროკონტროლერში და გაშვებას ახდენს თვით FLProg გარემო.

ამ პროექტში საუბარია ეკვალაიზერზე და მისი მუშაობის მთავარი პრინციპი ის არის რომ ბგერის დონის შესაბამისად მოახდინოს ნათურების ანთება, ამისთვის ჩვენ უნდა მოვახდინოთ არდუინოს შესასვლელი სიგნალების დისკრეტიზაცია და მიღებული სიგნალის საშუალებით მოვახდინოთ გამოსასვლელი სიგნალების მართვა.

მთავარი განმსაზღვრელი დონის ეკვალაიზერში უნდა იყოს ხმის გადამწოდის ტექნიკური მახასიათებლები და მათთან გადამწოდის გამოსასვლელი სიგნალის ხმის სიძლიერის დიაპაზონის განსაზღვრა. ჩვენ პროექტში გამოყენებული გვაქვს HW-181 MIC ტიპის გადამწოდი, რომლის კვებისთვის გამოიყენება +5 ვ. ძაბვა და რომელიც ოპტიმიზირებულია არდუინოსთან მუშაობაზე. გადამწოდი თავის გამოსასვლელზე იძლევა ანალოგურ სიგნალს, რომელმაც თეორიულად შეიძლება მიიღოს 0-დან 5 ვოლტამდე. თუ ჩვენ ამ სიგნალს შევუშვებთ არდუინოს ერთერთ ანალოგურ შესასვლელზე (A0 ... A5), მაშინ არდუინოში განხორციელებული ანალოგურ-ციფრული გარდაქმნის შემდეგ სიგნალი მიიღებს სახეს 0-დან 1023-დე. ჩვენს მიერ ჩატარებული ექსპერიმენტის შედეგად დადგენილ იქნა, რომ ადამიანის ყურისთვის ნორმალური ხმის (მუსიკალური ნაწარმოების) დიაპაზონის შემთხვევაში გადამწოდის გარდაქმნილი სიგნალი არდუინოში საკმარისია იცვლებოდეს 430-დან 460-მდე.

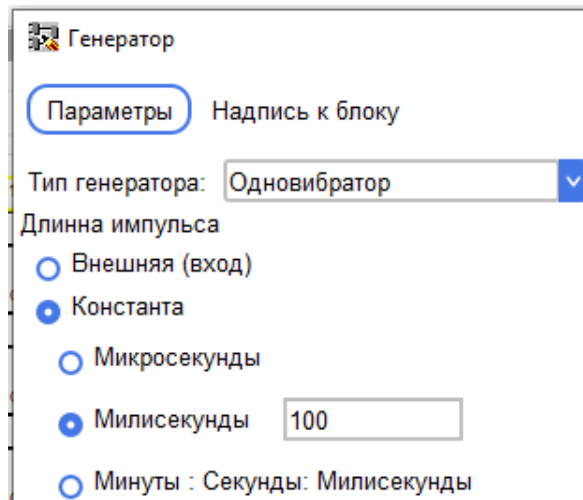


ფიგ. 50.1 ეკვალიზერის პროგრამა შევქმნილი FLPროგ გარემოში

უფრო კონკრეტულად, ჩვენ ავიღეთ დიაპაზონი 438-დან 457-მდე, ეს დიაპაზონი გავყავით ოც ნაწილად და შევქმენით პროგრამა ოცი კომპარატორის შემადგენლობით A1 ... B10, B21 ... B30. ყველა კომპარატორის პირველი შესასვლელი გაერთიანებულია და შეერთებულია არდუინოს იმ ანალოგურ კონტაქტთან რომელზედაც შეერთებულია ჩვენი გადამწოდი, კონკრეტულად A0-ზე, კომპარატორის მეორე ფეხებზე კი ჩაწერილი გვაქვს კონსტანტები 438, 439 ... 447, 448, 449 ... 457. კომპარატორები დაპროგრამებულია მეთობა ტოლობაზე >= და როდესაც სიგნალი მის პირველ ფეხზე I1 ტოლი გახდება ან გადააჭარბებს მეორე ფეხზე I2 ჩაწერილ კონსტანტას, მაშინ ამუშავდება ეს კომპარატორი და თავის გამოსასვლელზე გამოსცევს სიგნალს ლოგიკურ ერთიანს. კომპარატორის გამოსასვლელი შეერთებულია გენერატორების შესასვლელებთან B11 ... B20, B31 ... B40, რომლებიც დაპროგრამებულია ერთვიბრატორის რეჟიმში მუშაობისათვის იმპულსის ხანგრძლიობით 100 მლწ (ფიგ. 50.2). ანუ როდესაც კომპარატორის გამოსასვლელზე სიგნალი მიიღებს ლოგიკური ერთიანის მნიშვნელობას, მაშინ ამუშავდება შესაბამისი გენერატორი და გამოიმუშავებს ერთ იმპულსს 100 მლწ-ის ხანგრძლიობით, რა ხანგრძლიობისაც არ უნდა იყოს შესასვლელი სიგნალი.

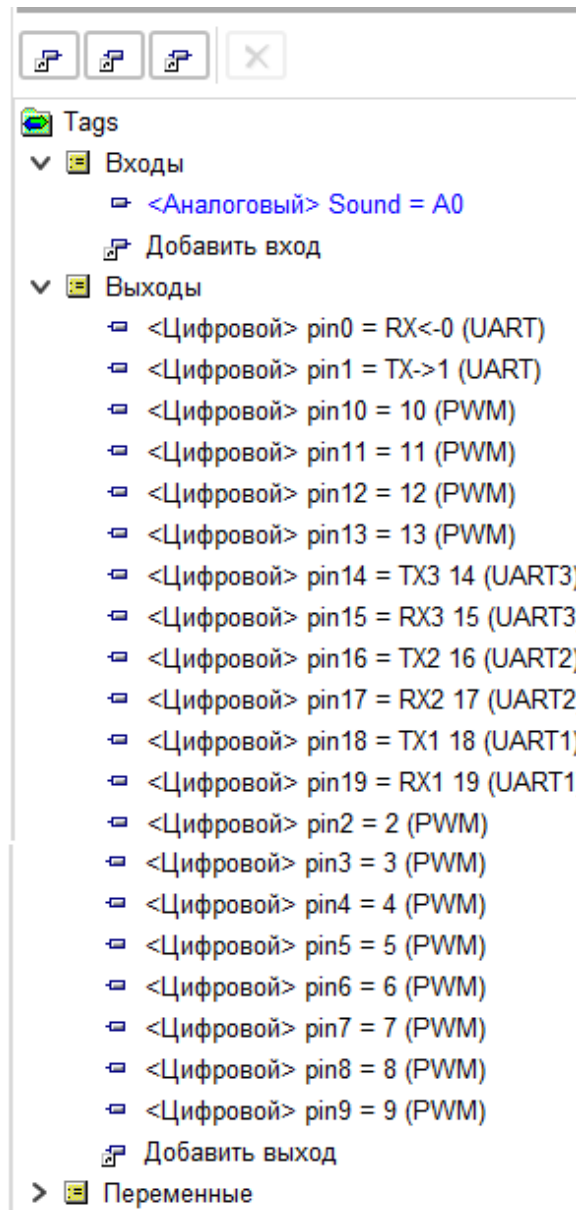
თუ პროგრამა უკვე კომპილირებული და ჩატვირთულია არდუინოში, მაშინ ხმის გადამწოდზე რაიმე ხმაურის, ან მუსიკის ზემოქმედების შედეგად ამუშავდება ხმის სიმაღლის

შესაბამისი რაოდენობის კომპარატორები, რომლებიც თავის მხრივ აამუშავებენ ერთვიბრატორებს. ეს ერთვიბრატორები კი დენის შემზღუდავი წინააღმდეგობების გავლით შეერთებულია შუქდიოდებთან და აანთებენ მათ გადამწოდზე მოდებული მუსიკის ტემპში.

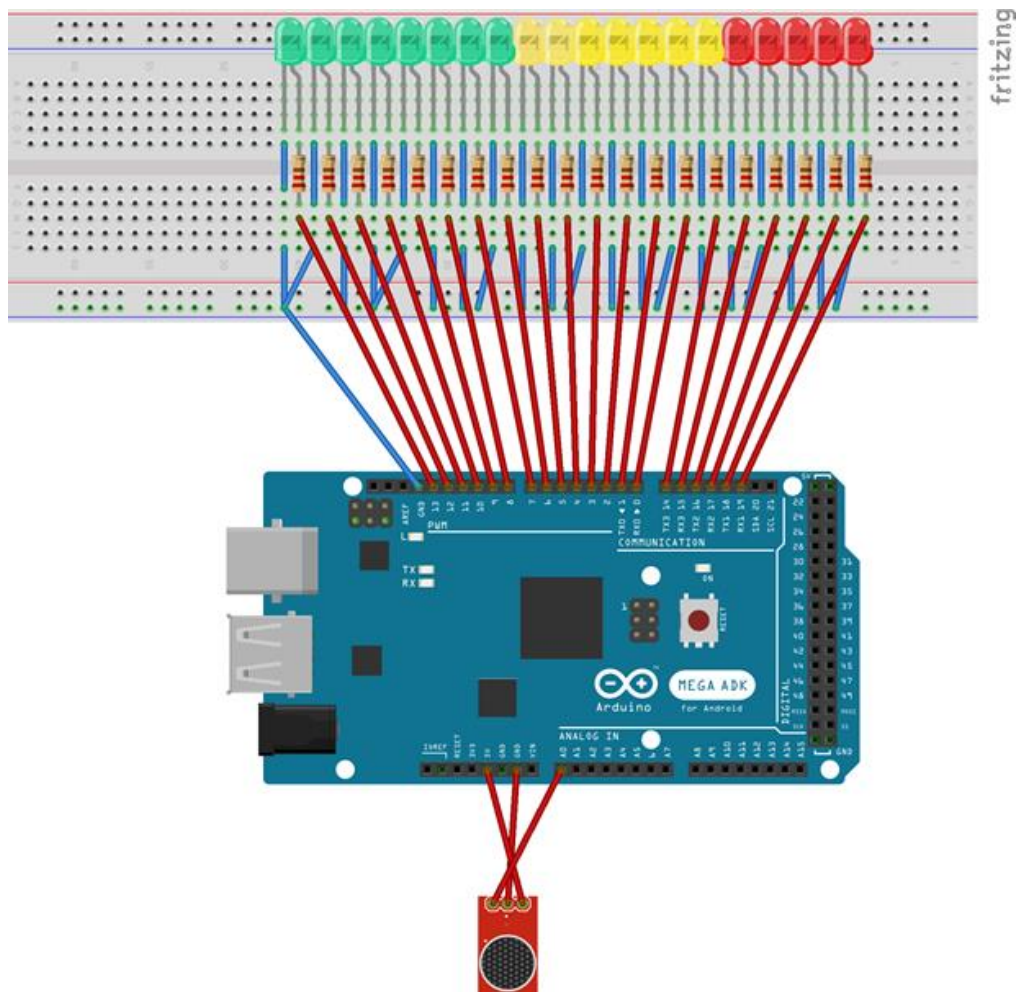


ფიგ. 50.2 გენერატორის პარამეტრიზაციის ფანჯარა

პროექტის ტეგების ზონის ფანჯარა წარმოდგენილია ფიგ. 50.3-ზე, ხოლო პროექტის Fritzing სქემა-ფიგ. 50.4-ზე.



ფიგ. 50.3 პროექტის ტეგების ზონის ფანჯარა



ნახ. 50.4 პროექტის Fritzing სქემა

პროექტი_51

შუქდიოდური მატრიცა, ნათება ცალკეულ დიაგონალდებზე



1388BSR ტიპისა და მისი მსგავსი დიოდური მატრიცები შეიცავს 64 შუქდიოდს (8*8), მაგრამ გამომყვანი ფეხების რაოდენობა სულ 16-ია 8 ერთმხარეს და 8 მეორე მხარეს, ასე რომ თვითოეული ამ შუქდიოდთაგანის პირდაპირი მართვა შეუძლებელია, საჭიროა გადავიდეთ დინამიურ მართვაზე, სადაც შუქდიოდები ინთება მიმდევრობით რაღაც დროის დაყოვნებით და ციკლურად, სადაც ეს დროის დაყოვნება შერჩეულია ისეთნაირად, რომ იგი თვალისთვის შეუმჩნეველი უნდა დარჩეს. თუ როგორ უნდა გავაკეთოთ ეს წარმოვაგენთ ქვემოთმოყვანილ პროექტში.

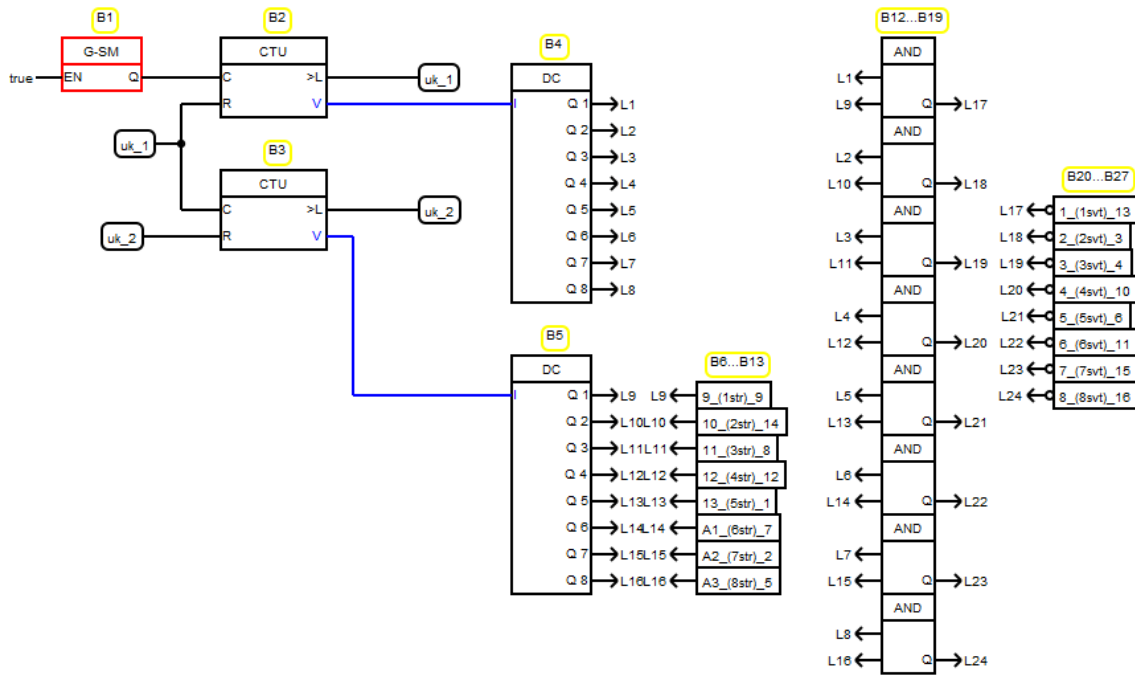
დავსვათ ასეთი ამოცანა, დაუშვათ ჩვენ გვჭირდება მატრიცის შუქდიოდების ნათება დიაგონალზე, ე.ი. ანთებული უნდა იყოს 1 სტრიქონის პირველი შუქდიოდი, მე-2 სტრიქონის მე-2 შუქდიოდი და ა.შ. მერვე სტრიქონის მე-8 შუქდიოდი (ფიგ. 51.1).



ფიგ. 51.1 პირველი დიაგონალის შუქდიოდების ნათება

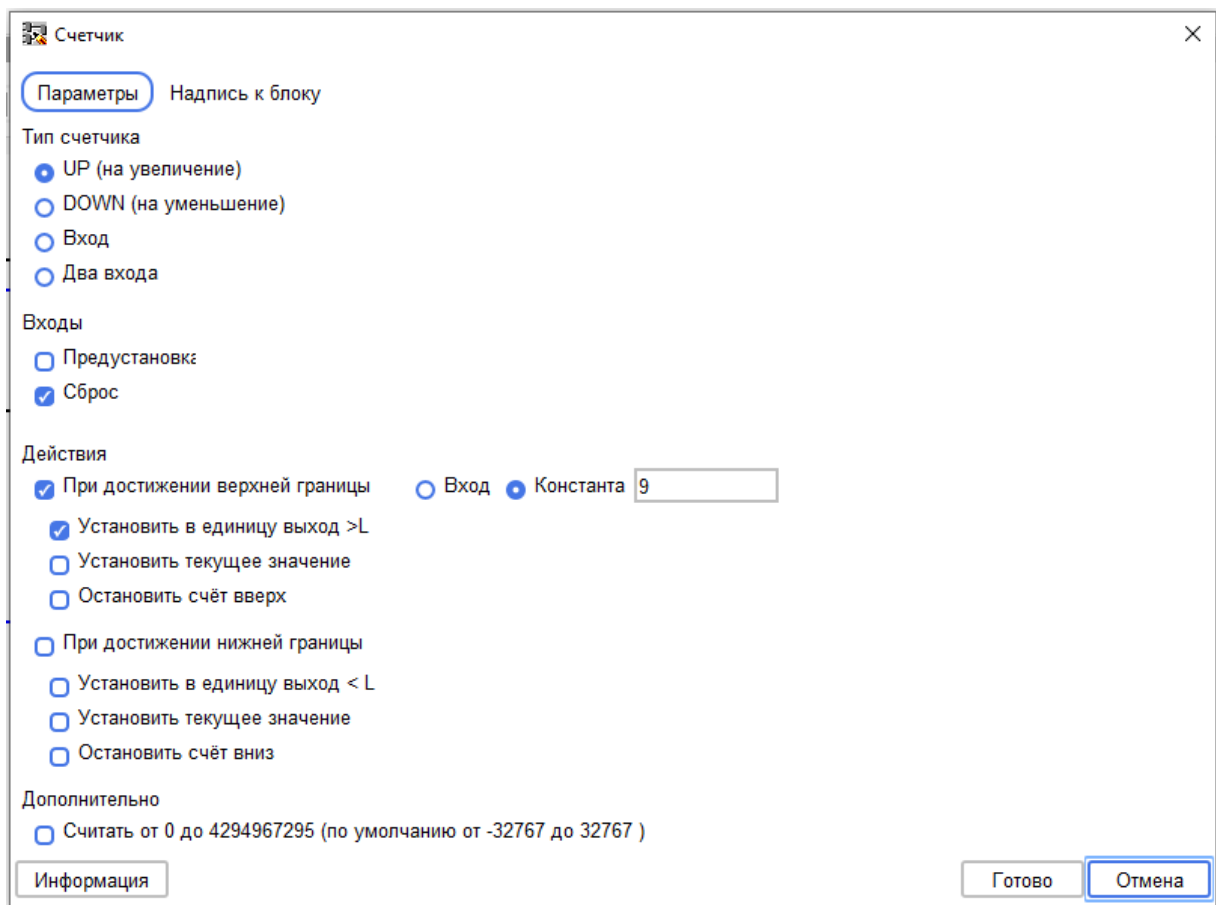
აღნიშნული ალგორითმის რეალიზაცია FLProg დამუშავების გარემოში ნაჩვენებია ფიგ. 51.2-ზე. ფიგურაზე B1 წარმოადგენს სიმეტრიულ მულტივიბრატორს რომლის მიერ გენერირებული სიგნალის იმპულსისა და პაუზის ხანგრძლივობა ტოლია და შეადგენს 10

მიკროწამს. გენერატორის გამოსასვლელი შეერთებულია B2 იმპულსების მთვლელის სინქრონიზაციის C შესასვლელზე, დავარქვათ ამ მთვლელს მატრიცის სვეტების მთვლელი.



ფიგ. 51.2 . პირველი დიაგონალის შუქდიოდების ნათების პროგრამა

სვეტების მთვლელის ბლოკის პარამეტრიზაციის ფანჯარა გამოსახულია ფიგურა 51.3-ზე, რომელიც მიიღება ამ ბლოკზე თავის მარცხენა ღილაკით ორჯერ დაწკაპუნებით. იმისათვის, რომ ავანთოთ რომელიმე შუქდიოდი საჭიროა შესაბამისი სვეტის შესაბამის ფეხზე (მატრიცის ფეხის ნომერაცია აითვლება მისი კორპუსის წინა წარწერდან მარცხენა პირველი ფეხიდან მარჯვენა მე-8 ფეხამდე და მერე გაგრძელდება უკანა მარჯვენა მე-9 ფეხიდან ბოლო მარცხენა მე-16 ფეხამდე. ეს B2 ბლოკი პროგრამაში ასრულებს მატრიცის სვეტების გადათვლის ფუნქციას. ამ ბლოკის პარამეტრიზაციის ფანჯრიდან ჩანს, რომ ბლოკი მუშაობს მატეზაზე რეჟიმში, იგი ითვლის 8-მდე რადგან მატრიცაში 8 სვეტია და როგორც კი მე-9 იმპულსი გამოჩნდება, რადგან ჩვენ კონსტანტის ველში-Константа ჩაწერილი გვაქვს ცხრიანი და მიცემული გვაქვს მითითებები (იხ. ფიგ. 51.3), (ჩართული გვაქვს ალმები)-ზედა ზღურბლის მიღწევის შემდეგ-При достижении верхней границы დადგეს ერთიანში L> გამოსასვლელი- Установить в единицу L> выход, ამიტომ ამ მთვლელის L> გამოსასვლელზე დადგება ლოგიკური ერთიანი და რადგანაც ჩვენ ჩართული გვაქვს ნულზე დაყენების ალამი-Снос, ამიტომ მთვლელის ბლოკის შესასვლელზე გაჩნდება აგრეთვე ამავე ბლოკის ნულზე დაყენების შესასვლელი R.



ფიგ. 51.3 სვეტების მთვლელის პარამეტრიზაცია

ახლა ჩვენ რომ მეცხრე იმპულსის მოსვლისას შეგვეძლოს დავაყენოთ B2 ბლოკი ნულოვან მდგომარეობაში ამიტომ მისი $L >$ გამოსასვლელი უნდა შევეუერთოთ R შესასვლელს, ე.ი შევქმნათ უკუკავშირი, მაგრამ თვითონ დამუშავების გარემო FLProg-ი გვიკრძალავს ასეთ პირდაპირ უკუკავშირების შექმნას. მაგრამ შეგვიძლია შევქმნათ ცვლადი ტეგების ზონაში დავარქვათ მას პირველი უკუკავშირი-uk_1, ჩამოვიტანოთ ეს ცვლადი, საუშაო ზონაში ორჯერ, ერთი შეუერთოთ B2 ბლოკის $L >$ გამოსასვლელს, მეორე კი ამავე ბლოკის R შესასვლელს და ამით ჩვენ მართალია შემოვლითი გზით, მაგრამ მაინც შევძელით უკუკავშირის ფორმირება და მთვლელის დაყენება ნულზე მასზედ იმპულსების სერიიდან მე-9 იმპულსის მოსვლის მომენტში, შემდეგ მთვლელი ისევ გააგრძელებს იმპულსების თვლას, ისევ დათვლის მე-9 იმპულსამდე, ისევ დაყენებდება ნულზე და ა.შ.შ. ციკლურად უსასრულობამდე. ანუ, ეს მთვლელი ასრულებს მატრიცის სვეტების ფორმირებას: დაითვალა ერთი იმპულსი, ე.ი აფორმირა პირველი სვეტი, დაითვალა მე-2 იმპულსი ე.ი აფორმირა მე-2 სვეტი და ა.შ.შ. ციკლურად.

ზუსტად ანალოგიურად არის დაპროგრამებული მეორე მთვლელიც B3 ბლოკი, იმ განსხვავებით, რომ მის სინხრონიზაციის შესასვლელზე მიეწოდება იმპულსები არა გენერატორის გამოსასვლელიდან, არამედ პირველი B2 მთვლელის uk_-ის გამოსასვლელიდან, რაც იმას ნიშნავს, რომ დაამთავრა რა ყველა სვეტის გადათვლა სვეტებისა პირველმა მთვლელმა, დაემატა მეორე მთვლელს ერთი იმპულსი და ასე შემდეგ ანუ მეორე მთვლელი ითვლის სტრიქონებს. გადაითვალა პირველმა მთვლელმა რვა სვეტი, მეორე მთვლელში ფორმირდა ახალი სტრიქონი და ა.შ.მ. მანამ მეორე მთვლელიც არ დაითვლის რვა სტრიქონს, მე-9 იმპულსის მოსვლის დროს კი ამუშავდება მეორე ცვლადი-uk_2 და ამ მომენტში ნულის მდგომარეობაში დადგება როგორც პირველი, ასევე მეორე მთვლელები. ესე იგი ამით დამთავრებული იქნება მატრიცის ყველა შუაქდიოდის გადათვლა, ანუ დამთავრებული იქნება ერთი კადრის ფორმირება. შემდეგ კი ეს პროცესი იმპულსების გადათვლის დაიწყება თავიდან. მაშასადამე, საბოლოო ჯამში მივიღეთ ის, რომ მატრიცის მართვისათვის გვაქვს ორი მთვლელი, რომლებიდანაც პირველი ითვლის სვეტებს, ხოლო მეორე - სტრიქონებს.

ახლა შევიდეთ ბლოკების ბიბლიოთეკის ბიტებთან ოპერაციის საქალაქდემი-Операция с битами და გადმოვიტანოთ ორი დემიფრატორი-Дешифратор B4, B5, ერთის შესასვლელი შევუერთოთ B2 ბლოკის გამოსასვლელს, ხოლო მეორეს კი B3 ბლოკის გამოსასვლელს. გავხსნათ მათი პარამეტრიზაციის ფანჯრები და გამოსასვლელების რაოდენობის ველში-Количество Выходов ორთავე ბლოკში ჩავწეროთ 8, 8 (ფიგ. 51.4).



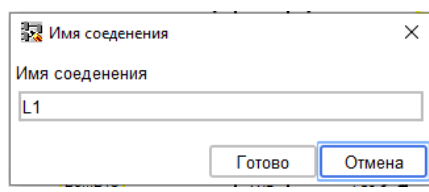
ფიგ. 51.4 სვეტებისა და სტრიქონის დემიფრატორების პარამეტრიზაციის ფანჯარა

დავაჭიროთ მზადყოფნის ლილაკს-Готово თითო და დავამთავროთ ამ დემიფრატორების პარამეტრიზაცია. დემიფრატორის მოქმედების პრინციპი მოკლედ, მდგომარეობს იმაში, რომ თუ მის შესასვლელზე მიწოდებულია რაიმე ანალოგური რიცხვი, მაგალითად 3, მაშინ დემიფრატორის მე-3 გამოსასვლელზე (Q3) დაჯდება ლოგიკური ერთიანი, ფიგ. 51.2, B4 ან B5, ხოლო ყველა დანარჩენზე-ლოგიკური ნოლი.

ახლა ჩვენ შეგვიძლია სტრიქონების დეშიფრატორის B5-ს გამოსასვლელები Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8 რომლების გაყვანილნი არიან არდუინოს 9, 10, 11, 12, 13, A1, A2, A3 კონტაქტებზე, პირდაპირ შევართოთ შუქდიოდური მატრიცის სტრიქონების შესასვლელებთან, რომლების ტექნოლოგიური მოსაზრებებიდან გამომდინარე მიერთებულნი არიან მატრიცის ფეხებთან არეულდარეულად, კონკრეტულად: 1 სტრიქონი შეერთებულია მატრიცის მე-9 ფეხთან, 2-14-სთან, 3-18-სთან, 4-12-თან, 5-1-თან, 6-მე-7, 7-მე-2-სთან და მე-8-მე-5 სთან, ასევე მივართებთ ჩვენც, მაგრამ აქვე უნდა გავითვალისწინოთ ის ამბავი, რომ FLProg პროგრამაში, იქ სადაც ბევრი დამაკავშირებელი ხაზების გავლება გვჭირდება, ნახაზის შემაერთებელი ხაზებით გადატვირთვის თავიდან აცილების მიზნით შესაძლებელია დავნომროთ ეს ხაზები და ამით გავარკვიოთ თუ რომელი წერტილი სადაა შეერთებული. ეს უნდა მოხდეს შემდეგნაირად. დავუშვათ ფიგ. 51.2-ზე B4 ბლოკის გამოსასვლელი შეერთებულია B12 ბლოკის შესასვლელთან (ფიგ. 51.5) და საჭიროა ეს ხაზი მოვიშოროთ თავიდან მაგრამ უნდა დაინომროს ამ ხაზის თავი და ბოლო, რომ რეალური შეერთება ამ ორი წერტილისა იგივე დარჩეს. ამიტომ თავგის მაჩვენებელი მიგვყავს ამ ხაზთან და როდესაც იგი შეიცვლის ფერს და გადაიქცევა წითლად, მაშინ დავაჭერთ თავგის მარცხენა ღილაკას, გამოვა წარწერა გარდაიქმნა ტექსტად-Преобразовать в текст, დავაჭიროთ ამ წარწერაზე თავგის მარჯვენა ღილაკით და ეკრანზე გამოვა ფანჯარა შეერთების სახელი-Имя соединения და იქვე შემოგვთავაზებს ავტომატურად შეერთების სახელს, (მაგალითად L1), ჩვენ შეგვიძლია დავთანხმდეთ ამ სახელს, (ანდა შევცვალოთ იგი სხვა რომელიმე სახელით) და მზადყოფნის-ღილაკზე Готово დაჭერის შემდეგ ეს ხაზი გაქრება და მის საწყის და ბოლო წერტილებს დაწერება სახელი L1 (ფიგ. 51.2). ეს იმას ნიშნავს, რომ B4 ბლოკის გამოსასვლელი და B12 ბლოკის შესასვლელი ელექტრულად დაკავშირებულია ერთმანეთთან.



ფიგ. 51.5 შემაერთებელი ხაზის მონიშვნა



ფიგ. 51.6 შემაერთებელი ხაზის დასახელებებით შეცვლა

ახლა შევუდგეთ სვეტების დემიფრატორის გამოსასვლელების შეერთებას შუქდიოდური მატრიცის სვეტების შესასვლელებთან. აქ უკვე დაგჭირდება ლოგიკური ბლოკების AND გამოყენება B12...B19 რვა ცალის რაოდენობით. შევუერთოთ სვეტების დემიფრატორის პირველი გამოსასვლელი პირველი AND ბლოკის (B12) პირველ შესასვლელს, სტრიქონების დემიფრატორის პირველი გამოსასვლელი ამავე პირველი AND ბლოკის მეორე შესასვლელს (იხ. ფიგ. 51.2) და ასე შემდეგ, ანალოგიურად, სვეტების დემიფრატორის მერვე გამოსასვლელი მერვე AND ბლოკის (B19) პირველ შესასვლელს ხოლო სტრიქონების დემიფრატორის მერვე გამოსასვლელი-იგივე AND ბლოკის მეორე შესასვლელს.

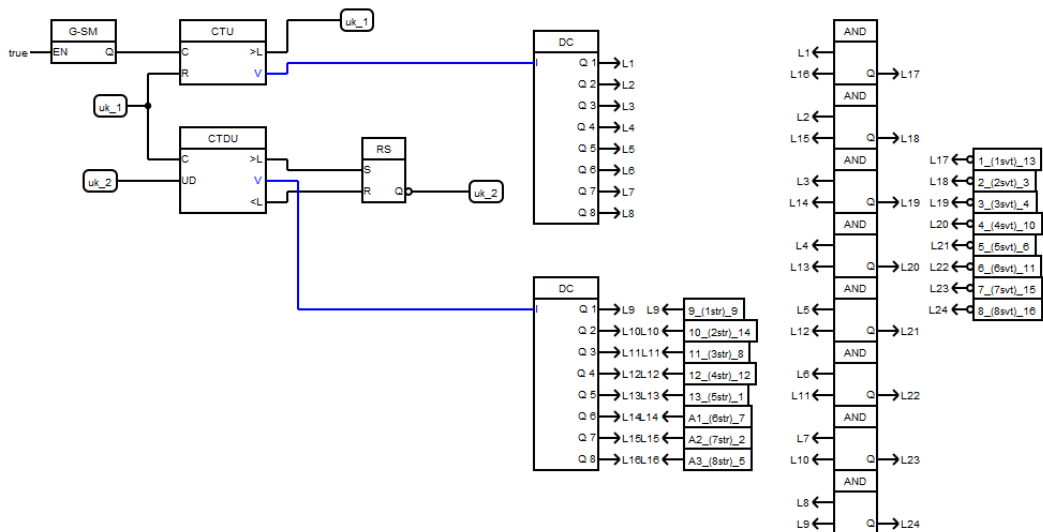
ამ AND ბლოკებზე მოხდება შესასვლელი სიგნალების ლოგიკური გამრავლება. მათ გამოსასვლელებზე ლოგიკური ერთიანის შესაბამის სიდიდეებს მივიღებთ: B12 ბლოკის გამოსასვლელზე მაშინ, როდესაც ორთავე დემიფრატორის პირველ გამოსასვლელებზე ერთიანებია, B13 ბლოკის გამოსასვლელზე მაშინ, როდესაც ორთავე დემიფრატორის მეორე გამოსასვლელებზე ერთიანებია და ასე შემდეგ B19 ბლოკის გამოსასვლელებზე მაშინ, როდესაც ორთავე დემიფრატორის მერვე გამოსასვლელებზე ერთიანებია. ახლა ჩვენ უნდა შევუერთოთ ამ ბლოკების გამოსასვლელები არდუინოს 1,2,3,4,5,6,6,8 გამოსასვლელებს, ხოლო ეს კონტაქტები კი შუქდიოდური მატრიცის პირველი სვეტის ფეხს, მეორე სვეტის ფეხს და ა.შ.შ. მერვე სვეტის ფეხს, ანუ ამ მატრიცის 13,3,4,10,6,11,15,16 ფეხებს შესაბამისად. ამით ჩვენ ავირჩიეთ ის შუქდიოდები, რომლებმაც უნდა გაანათოს განუწყვეტლივ ციკლურად მუშაობის დროს, ანუ დიაგონალზე განლაგებული შუქდიოდები (ფიგ. 51.1) მაგრამ როგორც ცნობილია, არჩეულმა შუქდიოდმა რომ გაანათოს საჭიროა ანოდის მხარეს უნდა იყოს დადებითი პოტენციალი, ანუ ლოგ. "1", ხოლო კათოდის მხარეს-ლოგ. „0“. ამის მისაღწევად საჭიროა ინვერსირება გავუკეთოთ სიგნალებს რომელსაც გამოიმუშავენს არდუინო მატრიცის სვეტების კონტაქტებზე შესაერთებლად, იხ. არდუინოს კონტაქტები 1,2,3,4,5,6,7,8 პატარა რგოლებით შესასვლელებზე. ამის შემდეგ ჩავტვირთოთ პროგრამა არდუინოში და შევამოწმოთ.

ახლა თუ ჩვენ გვინდა, რომ მატრიცის მეორე დიაგონალი ავანთოთ, მაშინ უნდა შევცვალოთ AND ლოგიკურ ბლოკებზე მიერთებული სიგნალები. (ფიგ. 51.7).



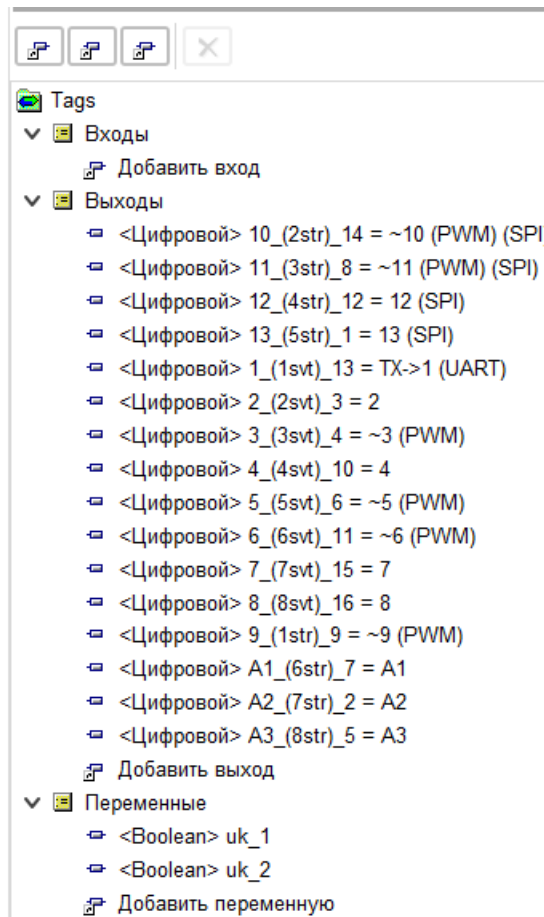
ფიგ. 51.7. მეორე დიაგონალის შუქდიოდების ნათება

კერძოთ უნდა დაწყვილდეს დემიფრატორების პირველი და მერვე, მეორე და მეშვიდე, მესამე და მეექვსე, მეხუთე და მეხუთე, მეექვსე და მესამე, მეშვიდე და მეორე და მერვე და პირველი გამომყვანები (ფიგ. 51.8).

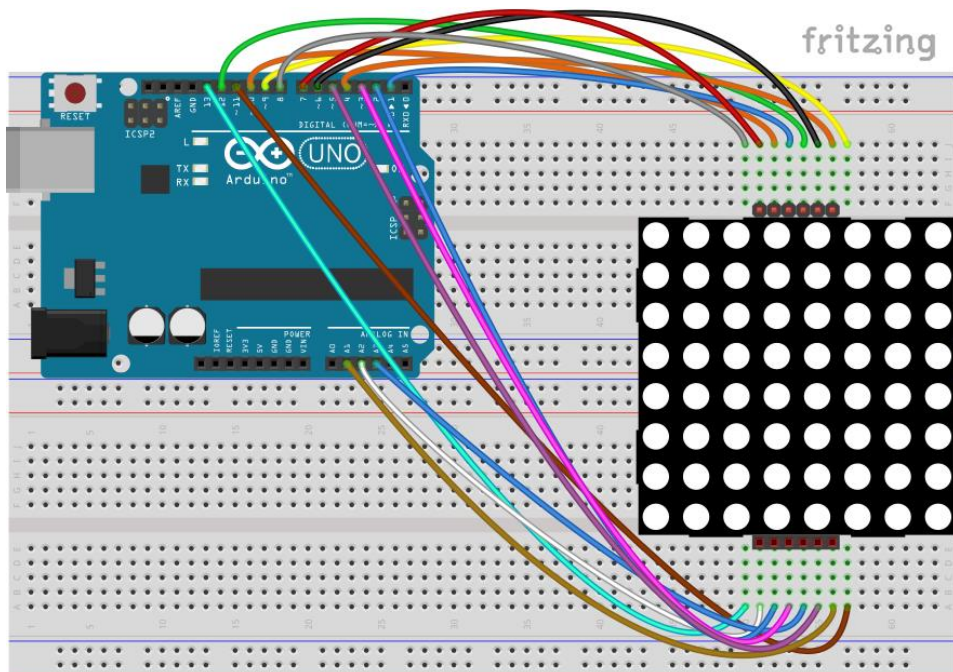


ფიგ. 51.8 მეორე დიაგონალის შუქდიოდების ნათების პროგრამა

პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 51.9-ზე, ხოლო Fritzing სქემა ფიგ. 51.10-ზე.



ფიგ. 51.10 პროექტის ტეგების ზონის ფანჯარა

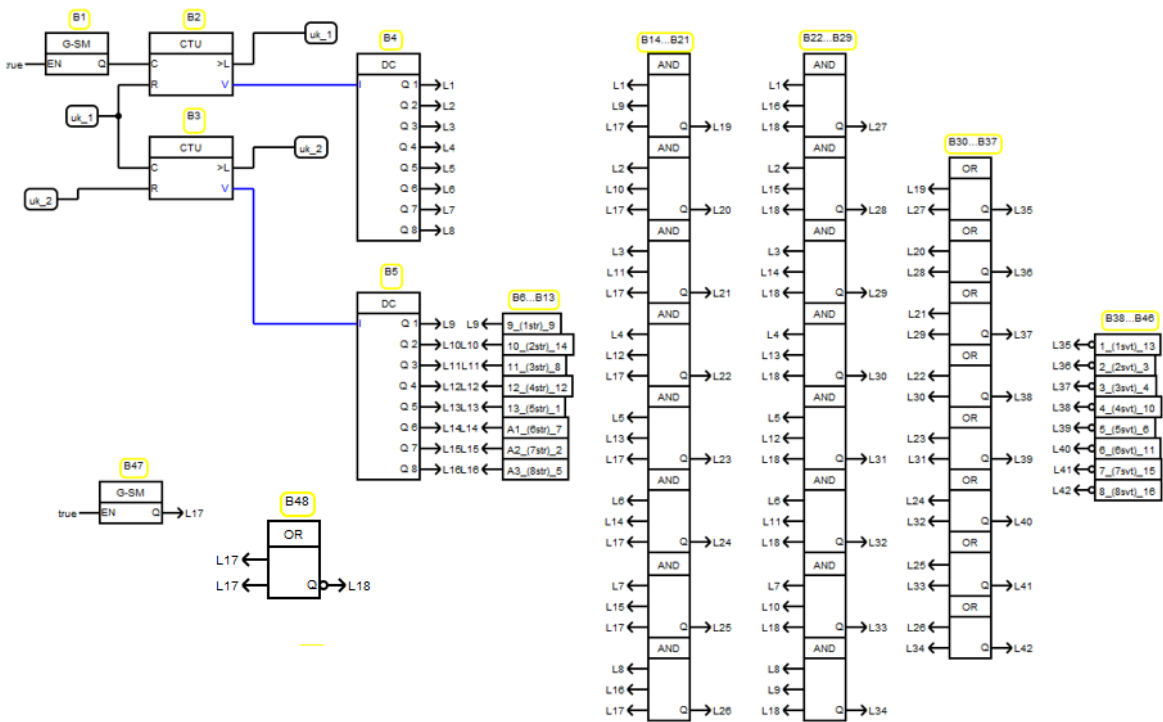


ფიგ. 51.10 პროექტის Fritzing სქემა

პროექტი_52 შუქდიოდური მატრიცა, ნათება ორთავე დიაგონალზე



ახლა შევექმნათ პროექტი მატრიცის ორთავე დიაგონალზე დიოდების ნათებისთვის მონაცვლეობით, ანუ როდესაც ერთი ნახევარი წამის განმავლობაში ერთი დიაგონალი ანთია და მეორე ნახევარი წამის განმავლობაში-მეორე დიაგონალი. აღნიშნული ალგორითმის რეალიზაციის პროგრამა მოცემულია ფიგურა ფიგ. 52.1-ზე.



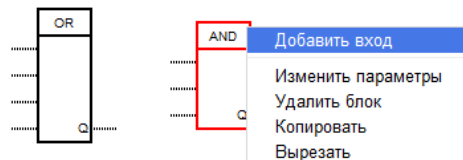
ფიგ. 52.1 შუქდიოდური მატრიცის ორთავე დიაგონალზე ნათების პროგრამა

როგორც ფიგურა 52.1-დან ჩანს ამ პროგრამის B1,B2,B3,B4,B5,B6...B13 ბლოკები ზუსტად იგივეა თავისი დასახელებითა და დანიშნულებით, პროექტ_51 პროგრამის ანალოგიურ ბლოკებთან შედარებით. განსხვავება მდგომარეობს იმაში, რომ აქ უნდა გამონათდეს ხან პირველი დიაგონალი ხან მეორე. ეს კი ხდება დამატებითი სიმეტრიული გენერატორისა B47 და ლოგიკური ბლოკის OR-ის B48 შემოტანით. გენერატორი

დაპროგრამებულია 500 მლწმ იმპულსისა და 500 მლწ პაუზის გამომუშავებაზე, ლოგიკური ბლოკი OR კი უბრალოდ ინვერტირებას უკეთებს გენერატორის გამომავალ სიგნალებს, როდესაც გენერატორის გამოსასვლელზე ლოგ. 1-ია, მაშინ B48 ბლოკის გამოსასვლელზე ლოგ. 0-ია და საწინააღმდეგოდ, როდესაც B47 ბლოკის გამოსასვლელზე ლოგიკური ნოლია, მაშინ B48 ბლოკის გამოსასვლელზე ლოგ. 1-ია. შესაბამისად ამისა, როდესაც გენერატორის გამოსასვლელზე ლოგიკური ერთიანია, მაშინ ჩვენ უნდა გამოვანათოთ მატრიცის პირველი დიაგონალი, ხოლო როდესაც ლოგიკური ერთია - მაშინ მეორე დიაგონალი.

თუ როგორ გამონათდეს პირველი დიაგონალი და როგორ გამონათდეს მეორე დიაგონალი განხილული იყო 51-ე პროექტში. შესაბამისად ამისა, ჩვენ ორივე სქემა დაგვჭირდება ამ ახალ პროგრამაში, მაგრამ ხან ერთ სქემას უნდა მიეცეს ნების დართვა მუშაობაზე ლოგიკური სქემების გამოყენებით, ხან მეორეს და შემდეგ ეს ნამუშევრები უნდა შეიკრიფოს ასევე დსამატებითი ლოგიკური სქემების გამოყენებით. ამის მიღწევა შეიძლება იმავე ლოგიკური ბლოკების END-ის გამოყენებით, რაც იყო 51-ე პროექტში, მაგრამ ამ ბლოკებს უნდა დაემატოს მესამე შესასვლელი მუშაობაზე ნებისდართვის რეალიზაციისათვის.

ლოგიკურ ბლოკებზე END, OR დამატებითი შესასვლელების ფორმირებისთვის უნდა დავაწკაპუნოთ თავის მარჯვენა ღილაკით და გამოსულ კონტექსტურ მენიუში უნდა დავაწკაპუნოთ დაემატოს შესასვლელზე-Добавить вход, რაც დაამატებს კიდევ ერთ შესასვლელს ბლოკზე (ფიგ. 52.2)



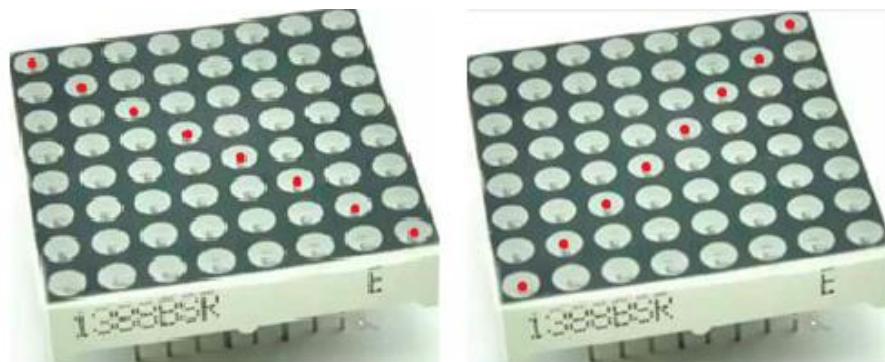
ფიგ. 52.2 შესასვლელების დამატება ლოგიკურ ბლოკებზე OR, END

ახლა გამოვიყენოთ 51-ე პროგრამის პირველი დიაგონალის ფორმირების END ბლოკები B12...B19 დავამატოთ ამ ბლოკებს თითო შესასვლელი და შევუერთოთ ისინი ყველანი სიმეტრიული გენერატორის B47 გამოსასვლელს (ფიგ.52.1, შემაერთებელი ხაზის ნომერი L17). მივიღებთ B14...B21 სამშესასვლელიან ლოგიკურ ბლოკებს, ეს ნიშნავს რომ ბლოკების ამუშავება და დიაგონალის ფორმირება შესაძლებელი იქნება მაშინ, როდესაც მათ მე-3 შესასვლელზე დაჯდება ლოგიკური ერთიანი, ანუ გენერატორის გამოსასვლელზე იქნება იმპულსი ხანგრძლიობით 500 მლწმ და სწორედ ამ პერიოდში მოხდება პირველი დიაგონალის გამონათების ნების დართვა, მეორე დიაგონალის ნების დართვა ამ დროს აკრძალული იქნება B48 ბლოკის გამოსასვლელზე ნულოვანი სიგნალის არსებობით, რომელიც კეტავს B22...B29

ბლოკებს. ანალოგიურად, აღებულია მეორე დიაგონალის მაფორმირებელი ლოგიკური END ბლოკები B22...B29, მათზედაც დამატებულია მე-3 ფეხები და შეერთებულია ისინი ლოგიკური OR ბლოკის B48 ინვერსირებულ გამოსასვლელზე ხაზებით L18. როგორც ვხედავთ უკანასკნელის შესასვლელი მიერთებულია გენერატორის ბლოკის B47 გამოსასვლელზე. ეს ნიშნავს, რომ როდესაც ამ გენერატორის გამოსასვლელზე პაუზაა ანუ ლოგიკური ნოლია, მაშინ B48 ბლოკის გამოსასვლელზე იქნება ლოგ. ერთიანი, და ამით ნება დაერთვება B22...B29 ამუშავებას, რადგან მათ მესამე შესასვლელებზე დაჯდა ლოგ. 1 და ამის შესაბამისად, ნება დაერთვება მეორე დიაგონალის გამონათებას.

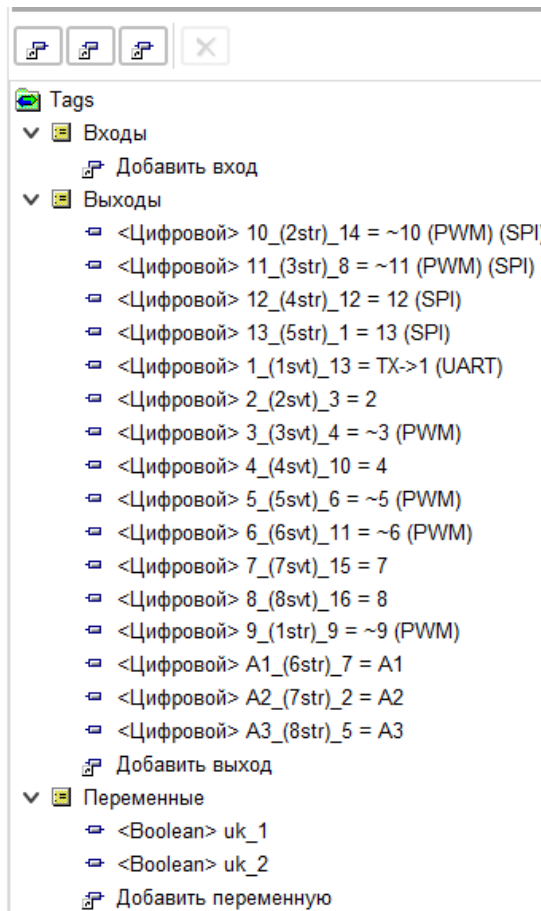
შევკრიბოთ ახლა ამ ორთავე დიაგონალის მაფორმირებელი END ბლოკების გამოსასვლელები ლოგიკურად, რათა შუქდიოდურ მატრიცაზე გაგზავნილ იყოს როგორც პირველი, ისე მეორე დიაგონალის მაფორმირებელი სიგნალები. ამისათვის გამოვიყენოთ ლოგიკური შეკრების ბლოკები OR (B30...B37), რომელთა გამოსასვლელებსაც შევაერთებთ უკვე არდუინოს გამოსასვლელ კონტაქტებთან 1,2,3,4,5,6,7,8, მათ კი როგორც ეს იყო 51-ე პროექტში - მატრიცის 1,2,3,4,5,6,7,8 სვეტებთან ანუ ამავე მატრიცის 13,3,4,10,6,11,15,16 ფეხებთან.

პროგრამის კომპილაციისა და გაშვების შემდეგ დავინახავთ, რომ აღნიშნული პროგრამა უზრუნველყოფს მორიგეობით ხან პირველი დიაგონალის გამონათებას ნახევარი წამის განმავლობაში, ხან მეორე დიაგონალის გამონათებას ასევე ნახევარი წამის განმავლობაში მორიგეობით და ა.შ. უსასრულოდ. (ფიგ.52.3).

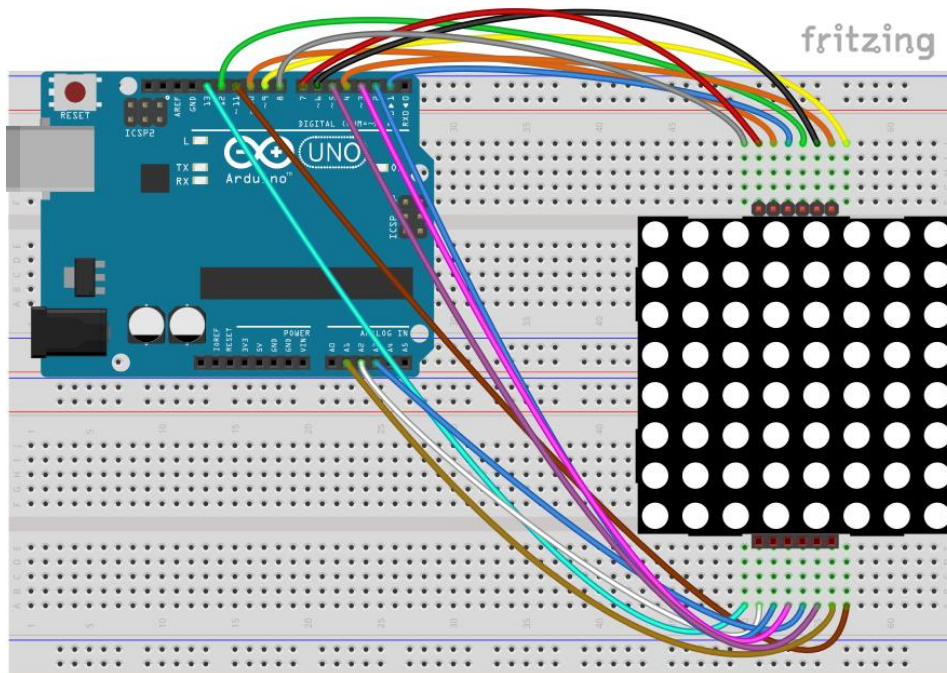


ფიგ. 52.3 პირველი და მეორე დიაგონალის გამონათება 1 ჰერცის სიხშირით

ამ პროექტის ტეგების ზონის ფანჯარა იგივეა რაც წინა ოროექტის და ასახულია ფიგ. 52.4-ზე, ხოლო ამავე პროექტის Fritzing სქემა იგივეა, რაც წინა პროექტში და ასახულია ფიგ.52.5-ზე.



ფიგ. 52.4 პროექტის ტეგების ზონის ფანჯარა



ფიგ.52.5 პროექტის Fritzing სქემა

პროექტი_53 რადიოსიხშირული იდენტიფიკაცია RFID



RFID (Radio Frequency Identification, რადიოსიხშირული იდენტიფიკაცია)-ეს არის მეთოდი მოწყობილობების უკონტაქტო ამოცნობისათვის, მონაცემთა უკონტაქტო გაცვლისათვის. მაგ. RFID შეიძლება გამოყენებულ იქნას იმ ადამიანთა იდენტიფიკაციისათვის, რომლებიც შედიან ან გამოდიან დაწესებულებებში.

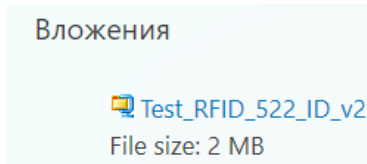
RFID-ის სისტემების საფუძველს წარმოადგენს წამკითხველები და მოდულები (ბრელოკები ან ბარათები). ეს უკანასკნელი ჩვეულებრივ მუშაობენ იმ მუხტის ხარჯზე, რომელსაც იღებენ წამკითხველებისაგან, რადგან თავიანთი კვების წყაროები არ გააჩნიათ. ამ მუხტების გადაცემა და მიღება ხორციელდება წამკითხველებში და მოდულებში ჩამონტაჟებული მიკრო ანტენებით. ამ პროექტში გამოყენებული გვაქვს MF522-AM ტიპის RFID წამკითხველი მოდული, რომლის გამომყვანებსაც ვაერთებთ არდუინოს კონტაქტებთან შემდეგნაირად:

- MISO 12
- SCK 13
- SS 10
- MOSI 11
- GND GND
- 3,3 v 3,3v
- RST 9

MF522-AM წამკითხველი მოდულის მართვისათვის FLProg-ში ვიყენებთ მომხმარებლის მიერ დამუშავებულ სამომხმარებლო ბლოკს MFRC522_ID(CODE), რომელიც განთავსებულია ინტერნეტში მისამართით:

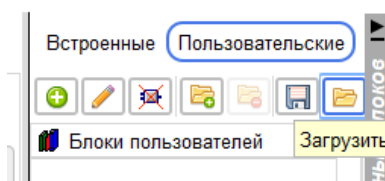
<https://flprog.ru/en/downloads/usersblocks/prochee/blok-raboty-so-schityvatelem-mfrc522/>

შევდივართ ამ ბმულზე, მოვძებნით ჩანართებში-Вложения ფაილს სახელწოდებით Test_RFID_522_ID_v2 (ფიგ. 53.1) და გადმოვიწერთ მას ჩვენს კომპიუტერში.



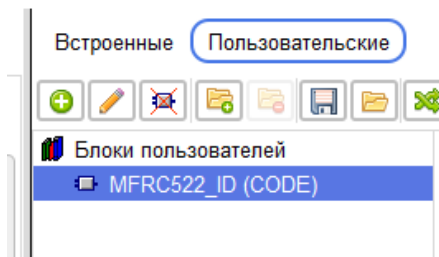
ფიგ. 53.1 მომხმარებლის ბლოკის ფაილი MFRC522_ID_(CODE).ubi

შემდეგ ჩვენ უნდა განვარხივით ეს ფაილი რომელიმე საქაღალდეში და შევიდეთ FLProg-ის განყოფილებაში სამომხმარებლო-Пользовательские, დავაჭიროთ ჩატვირთვის ღილაკს-Загрузить (ფიგ. 53.2) და გამოსულ საძიებო ფანჯარაში მივუთითოთ ჩვენ მიერ განაარხივებული საქაღალდე და ფაილი MFRC522_ID_(CODE).ubi.



ფიგ. 53.2 შეიცავს მომხმარებლის ბლოკის ჩატვირთვის ღილაკი

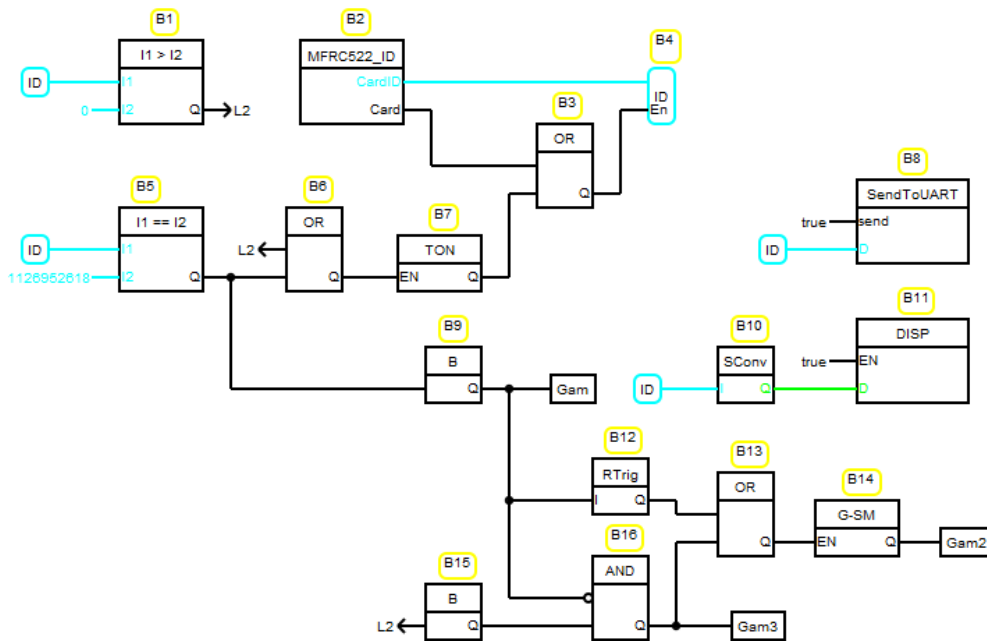
ამის შემდეგ ეს ბლოკი აისახება მომხმარებლის ბიბლიოთეკაში (ფიგ. 53.3), რომელიც შემდეგ ჩვეულებისამებრ შეგვეძლება გადავიტანოთ FLProg-ის სამუშაო გარემოში.



ფიგ. 53.3 მომხმარებლის ბლოკის ასახვა FLProg ბიბლიოთეკაში Пользовательские

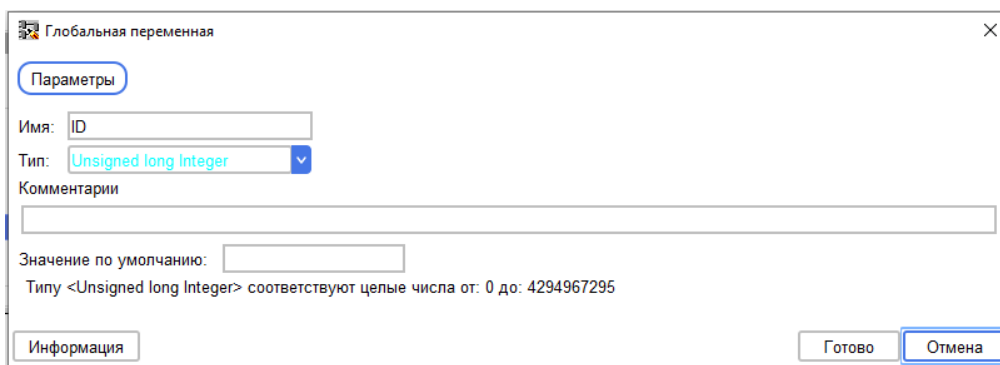
თუ გადმოვიტანთ ამ მომხმარებლის ბლოკს FLProg-ის სამუშაო ზონაში (ფიგ. 53.4) დავინახავთ რომ მას აქვს ორი გამოსასვლელი (ბლოკი B2): 1) ანალოგიური გამოსასვლელი CardID, ეს არის ანალოგიური გამოსასვლელი, რომელიც გვიჩვენებს ათათარიღიან ათობით კოდს - ბარათის (ბრელოკის) იდენტიფიკატორის ნომერს; 2) Card-ეს არის ციფრული გამოსასვლელი, რომელიც იღებს ორ მნიშვნელობას, ნოლს ან ერთს. ბრელოკის მიტანის დროს მიმდების მახლობლობაში ამ გამოსასვლელზე ერთიანია, ხოლო მოშორების დროს-ნოლიანი. ჩვენი პროგრამის მიზანია, იმ ბრელოკის მიახლოების დროს წამკითხველთან, რომლის იდენტიფიკატორიც დარეგისტრირებული გვაქვს პროგრამაში-აინთოს მწვანე შუქდიოდი, დისპლეიზე აისახოს ამ ბრელოკის იდენტიფიკატორის კოდი და ამასთანავე ბუზერმა

გამოსცეს მოკლე ხმოვანი შეტყობინება, ხოლო იმ ბრელოკის მიახლოების დროს, რომლის იდენტიფიკატორიც არ არის დარეგისტრირებული პროგრამაში-ანთოს წითელი შუქდიოდი, დისპლეიზე აისახოს ამ ბრელოკის იდენტიფიკატორის კოდი და ამასთანავე ბუზერმა გამოსცეს გრძელი ხმოვანი შეტყობინება.



ფიგ. 53.4 რადიოსიხშირული იდენტიფიკაციის პროგრამა

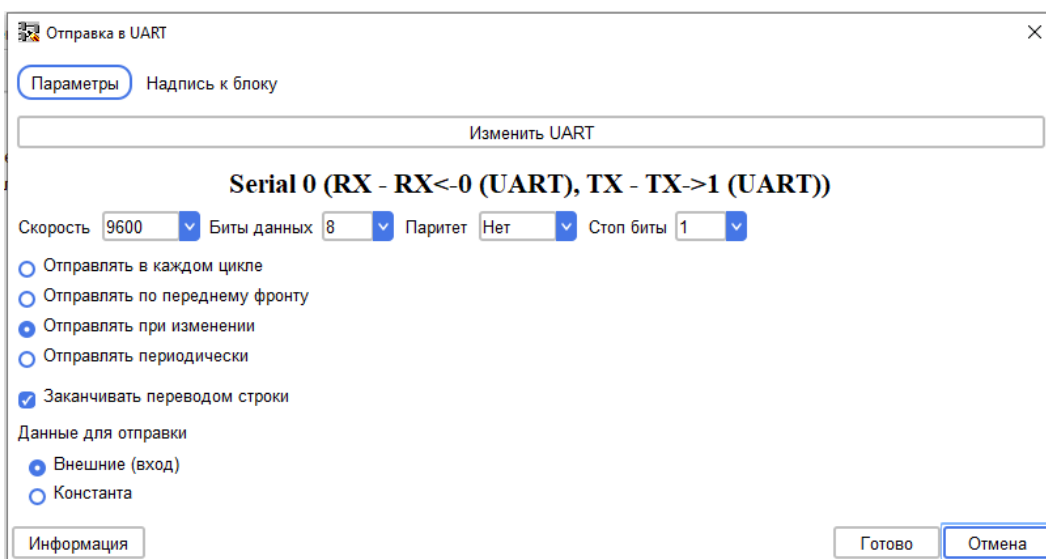
უნდა აღინიშნოს, რომ B2 ბლოკი კი ასრულებს თავის ფუნქციას, ანუ თავის გამოსასვლელზე გამოსცევს ანალოგურ და დისკრეტულ სიგნალებს, მაგრამ ეს სიგნალები განიცდიან წყვეტებს პროგრამის ციკლური მუშაობის ხასიათიდან გამომდინარე. ანუ დისპლეი და შუქდიოდი განიცდიან ციმციმს. ეს რომ ავიცილოთ თავიდან, უნდა შევიდეთ ტეგების ზონის ცვლადებში-Переменные და შევქმნათ გლობალური ცვლადი სახელწოდებით ID. ცვლადის ტიპის ველში ავირჩიოთ უნიშნო გრძელი მთელი-Uncsigned long Integer (ფიგ. 53.5).



ფიგ. 53.5 ცვლადის შექმნა სახელწოდებით ID

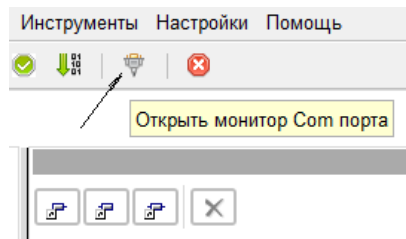
ჩამოვიტანოთ ეს ცვლადი პროგრამის სამუშაო ზონაში, დავაწკაპუნოთ მასზედ ორჯერ თავის მარჯვენა ღილაკით, გამოსულ ფანჯარაში ჩავრთოთ ჩაწერის ნებისდართვის შესასვლელი-Вход разрешения записи და დავაჭიროთ ღილაკს Готово. როგორც ფიგ. 53.4-იდან ჩანს ამ ახლად შექმნილ ცვლადს B4 გაუჩნდება ნების დართვის შესასვლელი En. ამ შესასვლელზე ყოველი იმპულსის მიწოდების დროს B4 ბლოკში ყოველთვის ფიქსირდება მის ანალოგურ შესასვლელზე მიწოდებული ინფორმაცია, ხოლო შემდეგ ინარჩუნებს ამ მდგომარეობას იმპულსის დამთავრების შემდეგაც. ახლა უკვე, როდესაც ბრელოკს მივიტანთ წამკითხავთან B2 ბლოკის დისკრეტული გამოსასვლელი იმპულსი B3 ბლოკის გავლით (ლოგიკური სქემა OR) ახდენს იდენტიფიკატორის კოდის ჩაწერას B4-ში და ამ კოდის შენარჩუნებას იმპულსის დამთავრების შემდეგაც მანამ, სანამ არ შეიცვლება შემავალი ინფორმაცია და არ დადგება ახალი იმპულსი B4 ბლოკის En შესასვლელზე წამკითხველთან იგინა, ან სხვა ბრელოკის მიახლოვებით გამომდინარე.

ახლა გავარკვიოთ ამბავი ბრელოკების საიდენტიფიკაციო კოდების შესახებ. ჩვენ გვჭირდება მინიმუმ ორი ბრელოკი, რომელთაგან ერთის კოდს დავაფიქსირებთ პროგრამაში ხოლო მეორესას არა. ამისათვის ჯერ ჩვენ უნდა დავადგინოთ ბრელოკების კოდები, რისთვისაც საჭიროა გავხსნათ Com მონიტორი, შემდეგ ჯერ პროგრამის ბიბლიოთეკის ზონიდან საქალაქედან UART სამუშაო ზონაში უნდა გადმოვიტანოთ UART-ში გაგზავნის ბლოკი Отправка в UART-B8. გავხსნათ ამ ბლოკის პარამეტრიზაციის ფანჯარა და დავაპროგრამოთ იგი, როგორც ეს ფიგ. 53.6-ზეა წარმოდგენილი.

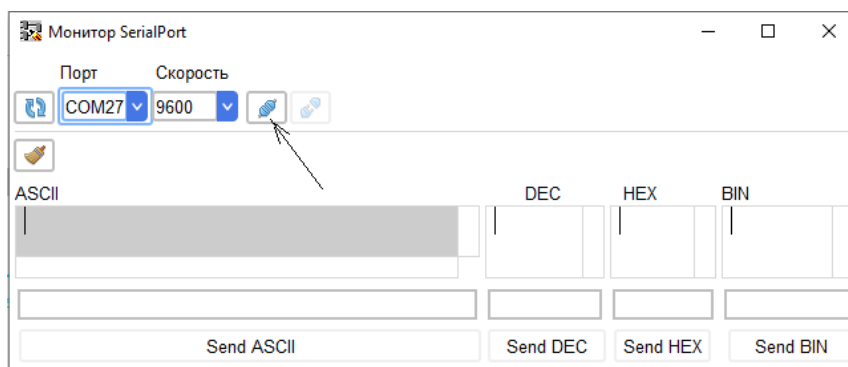


ფიგ. 53.6 UART-ში გაგზავნის ბლოკის პარამეტრიზაციის ფანჯარა

დავაჭიროთ ღილაკს Готово და გადავტვირთოთ პროგრამა არდუინოში. პროგრამის მენიუში დავაჭიროთ ღილაკს გაიღოს Com პორტის მონიტორი-Открыть монитор Com порта (ფიგ. 53.7).

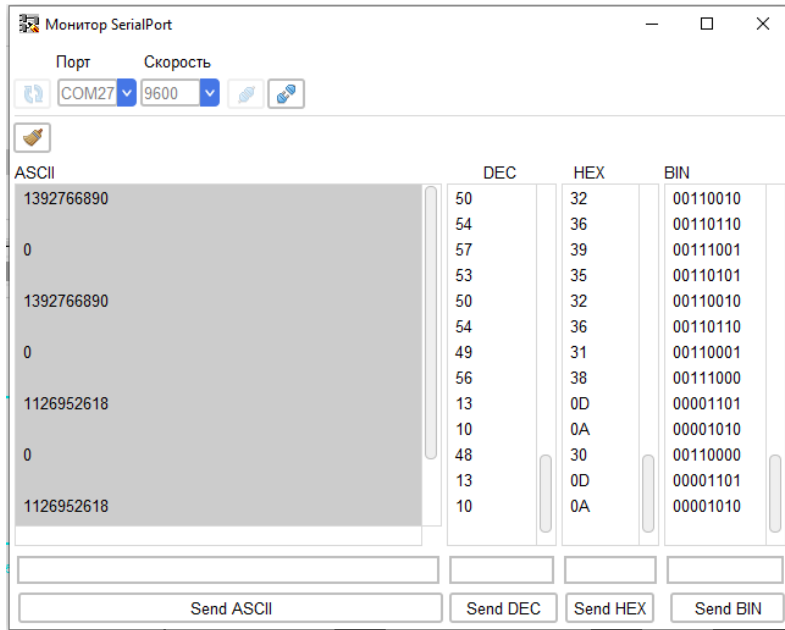


ფიგ. 53.7 Com პორტის გაღების ფანჯარა



ფიგ. 53.8 Com პორტთან დაკავშირების ფანჯარა.

პორტის გაღების ღილაკზე დაჭერით ეკრანზე გამოვა ახალი ფანჯარა (ფიგ. 53.8) სახელწოდებით მიმდევრობითი პორტის მონიტორი-Monitor Serial Port, სადაც პორტის ველში-Port უნდა ჩავწეროთ იმ ვირტუალური პორტის ნომერი, რომელიც მინიჭებული აქვს კომპიუტერის ოპერაციულ სისტემას ჩვენი არდუინოსთვის (COM27 ჩვენს შემთხვევაში), შემდეგ დავაჭიროთ შეერთების ღილაკს თითი (ნაჩვენებია ისრით) და ამის შემდეგ მოწყობილობა მზად არის ბრელოკების ID კოდების დასადგენათ. მივუახლოვოთ ორჯერ ზედიზედ MF522-AN წამკითხველს ჯერ პირველი ბრელოკი, შემდეგ მეორე და ეკრანზე გამოვა ჩვენი ბრელოკების საიდენტიფიკაციო კოდები ფიგ. 53.9).

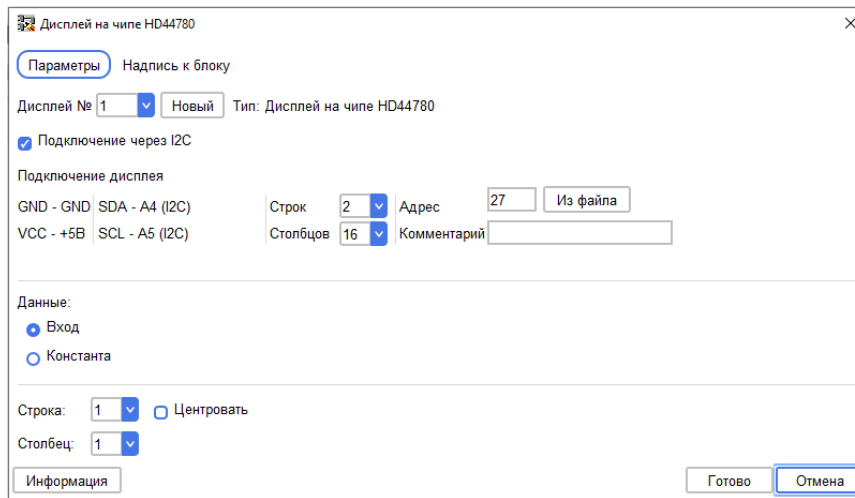


ფიგ. 53.9 ბრელოკების საიდენტიფიკაციო კოდების ამოკითხვა

ახლა გავთიშოთ მონიტორის მუშაობა ფიგ. 53.9-ზე ჩართვის მარჯვნივ მდებარე მეზობელი ღილაკის დაჭერით და გავაგრძელოთ ჩვენს პროგრამასთან მუშაობა. შევიტანოთ ერთერთი ბრელოკის მაგალითად მეორე ბრელოკის ნომერი 1126952618 ჩვენს პროგრამაში (ფიგ. 53.9). ამ მიზნით გადმოვიტანოთ კომპარატორის ბლოკი ბიბლიოთეკის შედარებების-Сравнение საქადალდედან Comparator პროგრამის სამუშაო ზონაში (B5) და მეორე შესასვლელზე ჩავწეროთ ამ მეორე ბრელოკის კოდი 1126952618, ამის შემდეგ თუ ჩვენ ამ მეორე ბრელოკს მივუახლოვებთ წამკითხველს, მაშინ ცვლადის ბლოკში B4 ჩაიწერება 1126952618 კოდი, B5 ბლოკის გამოსასვლელზე გამომუშავდება შედარების სიგნალი ერთიანი, რომელიც იმპულსების ყანყალის ბლოკის-Bounce B9 გავლის შემდეგ გამოვა არდუინოს მე-7 კონტაქტზე სახელწოდებით Gam და აანთებს მასთან მიერთებულ მწვანე შუქდიოდს.

ოგივე იმპულსი დააყენებს ლპოგიკური ერთიანის მდგომარეობაში B12 ტრიგერს, რომელიც რჩება ამ მდგომარეობაში პროგრამის მხოლოდ ერთი ციკლის განმავლობაში და შემდეგ დგება ნულოვან მდგომარეობაში. ეს ლოგ. ერთიანი B13 ბლოკის გავლით გაუშვებს B14 გენერატორს, რომელიც არდუინოს მე-6 კონტაქტის გავლით (Gam2 კონტაქტი) აამუშავებს ბუზერს და ეს გაგრძელდება მცირე ხნის განმავლობაში, პროგრამის მხოლოდ ერთი ციკლის განმავლობაში. მაშასადამე, პროგრამაში დარეგისტრირებული ბრელოკის მიახლოვების დროს წამკითხველთან მწვანე შუქდიოდი ანათებს მუდმივად სანამ ბრელოკი მიახლოვებულია მასთან, ხოლო ხმის სიგნალი კი ამუშავდება მცირე ხნით, მხოლოდ პროგრამის ერთი ციკლის განმავლობაში.

ID კოდის ასახისათვის გამოიყენება დისპლეი ჩიპზე HD44780 (B11), რომელიც დაპროგრამებულია ფიგ. 53.10 ასახული სქემის შესაბამისად.



ფიგ. 53.10 HD44780 ჩიპის პარამეტრიზაციის ფანჯარა

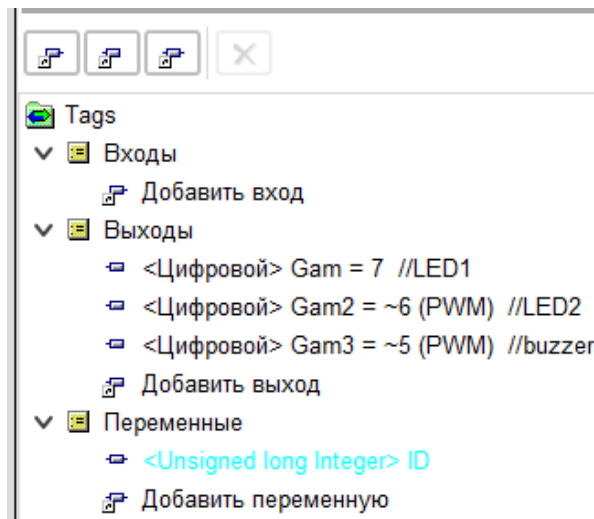
ბრელოკის წამკითხველთან მიახლოების მომენტში იდენტიფიკატორის კოდი ID ცვლადიდან B10 კონვერტირების ბლოკის გავლით აისახება B11 დისპლეის პირველ სტრიქონში, მაგრამ რაღაც დროის გავლის შემდეგ სასურველია ეს ასახული კოდი 1126952618 განულდეს. ამის მისაღწევად სიგნალი B5 კომპარატორის გამოსასვლელიდან ლოგიკური შეკრების ბლოკის B6 გავლით გაუშვებს ტაიმერის ბლოკს B7, რომელიც დაპროგრამებულია ფიგ. 53.11-ზე გამოსახული ნახაზის მიხედვით. ეს იმას ნიშნავს, რომ წამკითხველზედ ბრელოკის მოშორებიდან ერთი წუთის შემდეგ B7 ბლოკის გამოსასვლელზე გამოვა ერთიანი და რადგან ამ დროს B2 ბლოკის გამოსასვლელზე უკვე ნულოვანი კოდი იქნება დაფიქსირებული, ამიტომ B4 ბლოკში ჩაიწერება ნულოვანი კოდი, რაც აისახება B11 დისპლეიზე.



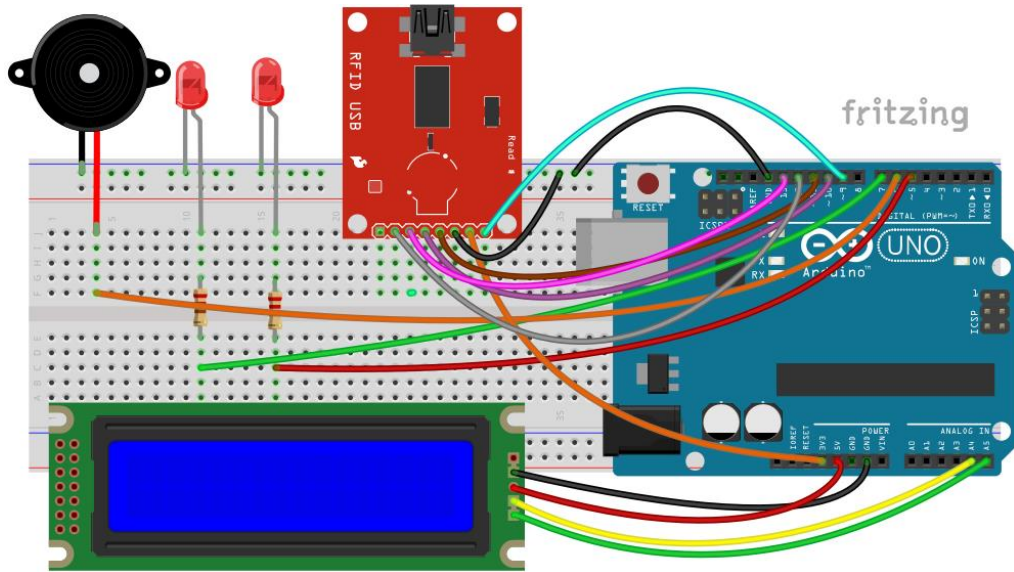
ფიგ. 53.11 ტაიმერის ბლოკის პარამეტრიზაცია

ახლა მივუახლოვით მე-2 ბრელოკი, ანდაც სრულად სხვა ბრელოკი, წამკითხავ მოწყობილობას. ამ შემთხვევაში ამუშავდება მეორე კომპარატორი B1, რადგან იგი დაპროგრამებულია მეტობაზე და მეორე შესასვლელზე დაფიქსირებული გვაქვს ნოლი. ლოგიკური ერთიანი ამ B1 ბლოკის გამოსასვლელზე ლოგიკური შეკრების ბლოკის B6-ის გავლით ჩაწერს ბრელოკის ID კოდს B4 ბლოკში, რაც მაშინვე აისახება დისპლეიზე, ამავე დროს იგივე სიგნალი ბოუნცე-bounce B15 ბლოკის გავლით მიეწოდება ლოგიკური გამრავლების B16 ბლოკის ერთერთ შესასვლელს, რომლის მეორე შესასვლელზე ზის პირველი კომპარატორის B5-ის ინვერტირებული სიგნალი. რადგან B5-ის გამოსასვლელზე ლოგიკური ნოლია და რადგან ახლა უკვე ჩვენი ნაცნობი ბრელოკი აღარ ფიგურირებს საქმეში, ამიტომ მისი ინვერტირებული სიგნალი ლოგიკური ერთიანი იქნება, ბლოკი B16-ს გამოსასვლელზე დაჯდება ლოგიკური ერთიანი, იგი მიეწოდება არდუინოს მე-5 კონტაქტს (Gam3), აინთება ამ კონტაქტზე მიერთებული წითელი შუქდიოდი და ამავე დროს იგივე სიგნალი B16-ს გამოსასვლელიდან B13 ლოგიკური შეკრების ბლოკის სახით გაუშვებს B14 სიმეტრიულ მულტივიბრატორს, რომელიც არდუინოს მე-5 კონტაქტის გავლით (Gam2) აამუშავებს ბუზერს მთელი იმ ხნის განმავლობაში, რა ხნითაც არარეგისტრირებული ბრელოკი მიახლოებული იქნება წამკითხველს.

პროექტის ტეგების ზონის ფანჯარა ასახულია ფიგ. 53.12-ზე, ხოლო მისივე სამონტაჟო Fritzing სქემა წარმოდგენილია ფიგურაზე 53.13-ზე.

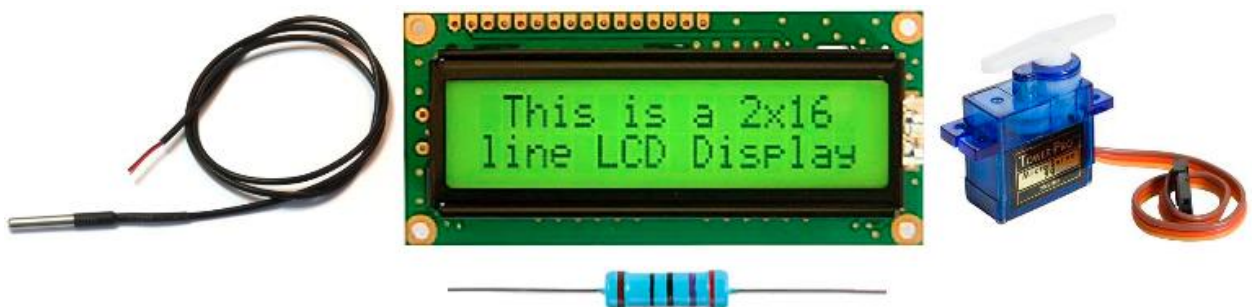


ფიგ. 53.12 პროექტის ტეგების ზონის ფანჯარა



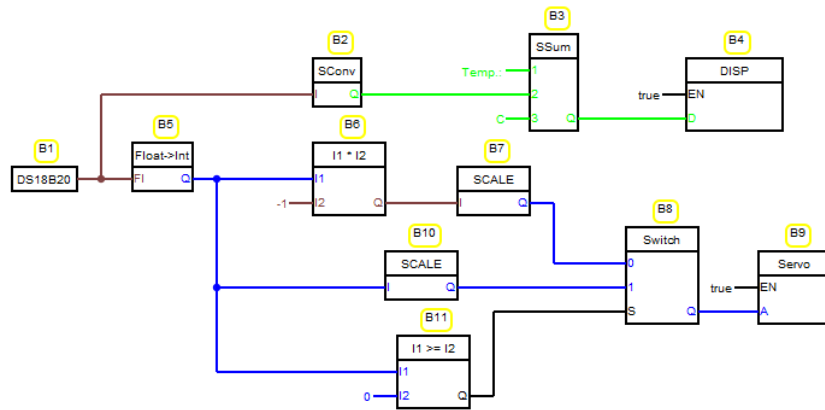
ფიგ. 53.13 RFID MF522-AN რადიოსიხშირული იდენტიფიკატორის Fritzing სქემა

პროექტი_54 ისრიანი თერმომეტრი



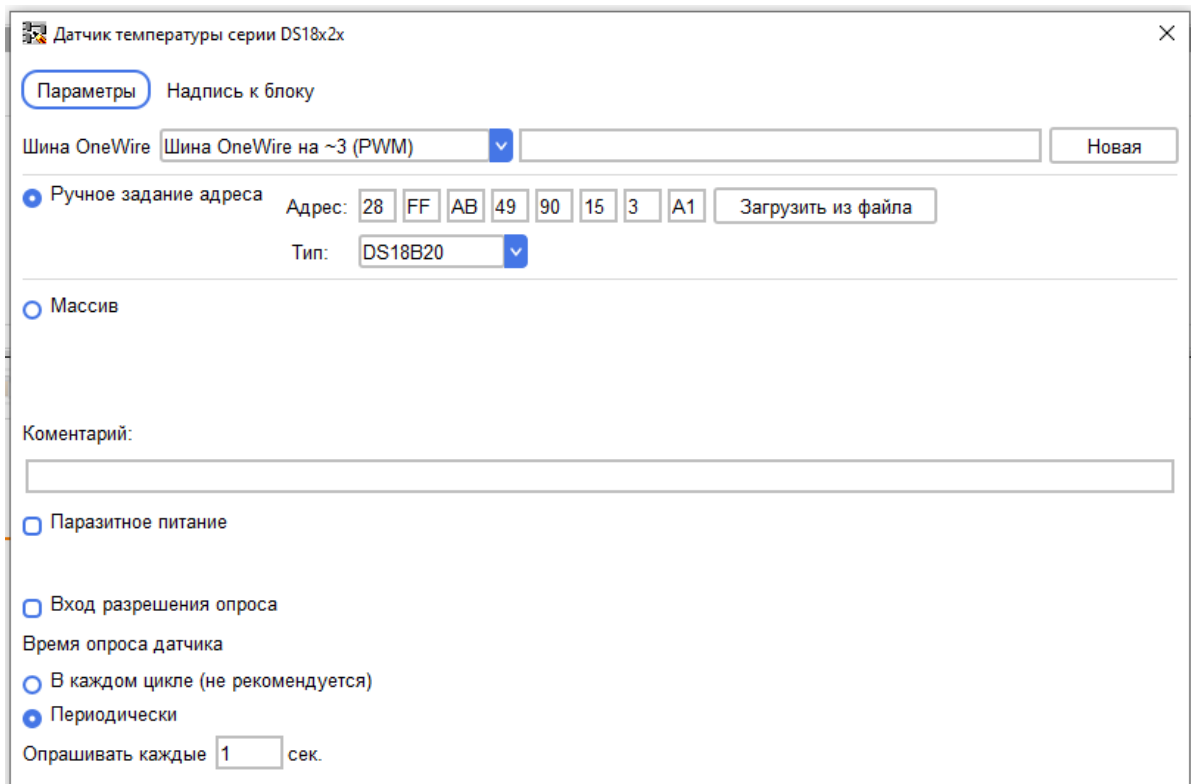
როგორც ცნობილია, ციფრული ინტელექტუალური გადამწოდი DS18B20 მუშაობს ტემპერატურის დიაპაზონში ცელსიუსის -55 გრადუსიდან +125 გრადუსამდე. ე.ი. ტემპერატურის გაზომვის დიაპაზონი შეადგენს 180 გრადუსს. არდუინოს პროექტებში გამოყენებული სერვომრავები კი მუშაობენ 0-დან 180 გრადუსამდე გადაადგილების კუთხის დიაპაზონში. ამ დიაპაზონების დამთხვევამ წარმოშვა იდეა გაგვერთიანებინა ამ ორი ხელსაწყოს ეს თვისებები და მიგველო ერთი ისეთი ხელსაწყო, ისრიანი თერმომეტრი, რომლის ისარიც გადაადგილდებოდა 0-დან 180 გრადუსამდე კუთხეზე, მაშინ როდესაც ტემპერატურა შეიცვლებოდა ცელსიუსის -55 გრადუსიდან +125 გრადუსამდე. ამ იდეის პროგრამული რეალიზაცია იხილეთ ფიგ. 54.1-ზე. პროგრამაში ტემპერატურის სიდიდის ასახვა წარმოებს ორი, ერთმანეთისაგან დამოუკიდებელი საშუალებით, პირველი ეს არის 2x16 დისპლეი B4 ჩიპზე HD44870, რომელზედაც ინფორმაცია აისახება სტანდარტულად, გადამწოდი B1,

სტრიქონების კონვერტაციის ბლოკი B2, სამშესასვლელიანი სტრიქონების შეკრების ბლოკი B3, სადაც პირველ შესასვლელზე ჩაწერილია კონსტანტა Temp : და პრაბელი, მეორე შესასვლელზე გასაზომი ტემპერატურის სიდიდე B2 ბლოკის გამოსასვლელიდან, ხოლო მე-3 შესასვლელზე ტემპერატურის განზომილება C.



ფიგ. 54.1 ისრიანი თერმომეტრის პროგრამა

B1 და B4 ბლოკების პარამეტრიზაციის ფანჯრები ასახულია ფიგ. 54.2 და 54.3-ზე შესაბამისად. აქ მხედველობაშია მისაღები, რომ B1 ბლოკის პარამეტრიზაციის ფანჯრის სამისამართო ველში-Адрес მისამართი ფიგ. 53.2 სხვადასხვა გადამწოდისთვის არის სხვადასხვა.



ფიგ. 54.2. DS18B20 გადამწოდის პარამეტრიზაციის ფანჯარა

Дисплей на чипе HD44780

Параметры Надпись к блоку

Дисплей № Тип: Дисплей на чипе HD44780

Подключение через I2C

Подключение дисплея

GND - GND	SDA - A4 (I2C)	Строк	<input type="text" value="2"/>	Адрес	<input type="text" value="27"/>	<input type="button" value="Из файла"/>
VCC - +5B	SCL - A5 (I2C)	Столбцов	<input type="text" value="16"/>	Комментарий	<input type="text"/>	

Данные:

Вход

Константа

Строка: Центровать

Столбец:

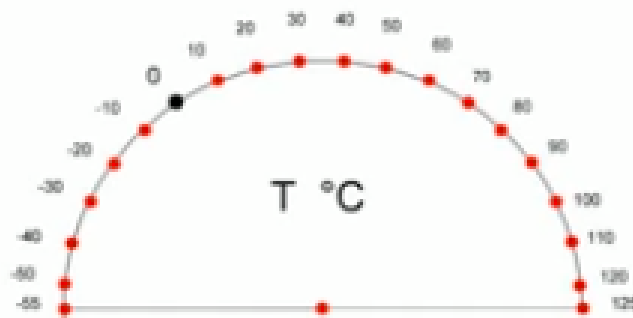
ფიგ. 54.3 დისპლეის ჩეკზე HD44780 პარამეტრიზაციის ფანჯარა

ახლა გადავიდეთ პროექტის მთავარ ნაწილზე, ანუ გაზომილი ტემპერატურის სიდიდესა და სერვომრავის ღუზის კუთხის მობრუნებას შორის თანაფარდობის დადგენაზე. აქ ჩვენ გამოვყოფთ ორ მდგომარეობას: თუ ტემპერატურა მინუსებშია და თუ ტემპერატურა პლიუსებშია. ამ მდგომარეობების განსაზღვრას მოვახერხებთ კომპარატორის ბლოკის B11 გამოყენებით. ეს ბლოკი ჩვენ აწყობილი გვაქვს მეტობა ან ტოლობაზე >=. კომპარატორის I2 შესასვლელზე თუ ჩავწერთ ჩვენ კონსტანტას ნოლს, მაშინ თუ I1 შესასვლელზე ტემპერატურის სიდიდე დადებითია ანუ ნოლზე მეტია, მაშინ B11-ის გამოსასვლელზე სიგნალის მნიშვნელობა იქნება ლოგიკური ერთი, ხოლო თუ ნოლზე ნაკლებია - მაშინ ლოგიკური ნოლი. აქ შევნიშნავთ, რომ ისრიანი თერმომეტრი მიიღება არცთუ ისე მაღალი სიზუსტის, ამიტომ წილადური რიცხვების მხედველობაში მიღება არაფერში არ გამოგადგება. ამიტომ ვიყენებთ კონვერტაციის ბლოკს B5-ს, რომელიც უბრალოდ გადააქცევს ათწილადურ რიცხვს მთელ რიცხვად წერტილის შემდგომი რიცხვების მოკვეთის გზით.

ახლა დავუშვათ ტემპერატურა დადებითია, მაშინ ორპოზიციანი გადამრთველის B8-ის S შესასვლელზე იქნება ლოგიკური ერთიანი და ეს გადამრთველი თავის Q გამოსასვლელზე გაუშვებს სიგნალს მისი "1" შესასვლელიდან. ამ შესასვლელზე კი მიეწოდება ტემპერატურის სიდიდის მნიშვნელობა B10 მასშტაბირების ბლოკის გავლით. თუ კი ტემპერატურა უარყოფითია მაშინ ამ B8 S შესასვლელზე იქნება ლოგიკური ნოლი და გადამრთველი თავის

გამოსასვლელზე გაუშვებს სიგნალს „0“ შესასვლელიდან. ამ შესასვლელზე კი მიეწოდება ტემპერატურის სიდიდის მნიშვნელობა B7 მასშტაბირების ბლოკის გავლით, მაგრამ წინასწარ ეს უარყოფითი სიდიდე ტემპერატურისა გადაიქცევა დადებით სიდიდით მინუს ერთზე გამრავლებით ბლოკში B6.

ახლა ვნახოთ, თუ როგორ უნდა დაპროგრამირდეს მასშტაბირების ბლოკები. ამისათვის შევხედოთ იმ შკალას, რომელზეც უნდა გვაჩვენოს ისრიანმა თერმომეტრმა ტემპერატურის მნიშვნელობა (ფიგ. 54.4).



ფიგ. 54.4 ისრიანი თერმომეტრის სკალა

როგორც წარმოდგენილი ნახაზიდან ჩანს, შკალა იწყება -55 გრადუსიდან მარცხნივ +125 გრაფდუსამდე მარჯვნივ. შკალის ნოლი მოთავსებულია 65 გრადუს კუთხეზე. ამის შესაბამისად სერვოდრავის ღუზა უნდა იდგეს განაპირა მარცხენა მდგომარეობაში -55 გრადუსის დროს, უნდა იდგეს 65 გრადუსის შესაბამის კუთხეზე ნოლოვანი ტემპერატურის დროს და უნდა იდგეს განაპირა მარჯვენა მდგომარეობაში პლიუს 125 გრადუსის დროს. სერვოდრავის კონსტრუქციიდან გამომდინარე, თუ მას წინა მხრიდან ვუყურებთ სერვოდრავის ღუზა განაპირა მარცხენა მდგომარეობაში დადგება თუ მას 180 გრადუსის შესაბამის სიგნალს მივაწვდით ანალოგურ შესასვლელზე. ჩვენი პროგრამის შემთხვევაში, თუ მის ანალოგურ შესასვლელზე რიცხვს 180-ს მივაწვდით და პირიქით, თუ მას რისხვს ნოლს მივაწვდით, მაშინ იგი დადგება განაპირა მარჯვენა მდგომარეობაში. საბოლოო ჯამში მივიღებთ, რომ როცა ტემპერატურა იცვლება - 55 გრადუსიდან ნოლ გრადუსამდე მაშინ სერვოდრავს უნდა მიეწოდოს ანალოგური სიგნალი 180-დან 125-მდე შესაბამისად, ხოლო როცა სიგნალი იცვლება 0 გრადუსიდან 125 გრადუსამდე, მაშინ სერვოდრავს უნდა მიეწოდოს ანალოგური სიგნალი 125-დან 0-მდე. ზემოთაღნიშნულის გათვალისწინებით არის დაპროგრამებული მინუს გრადუსების მასშტაბირების ბლოკი B7 (ფიგ. 54.5) და პლიუს გრადუსების მასშტაბირების ბლოკი B10 (ფიგ. 54.6).

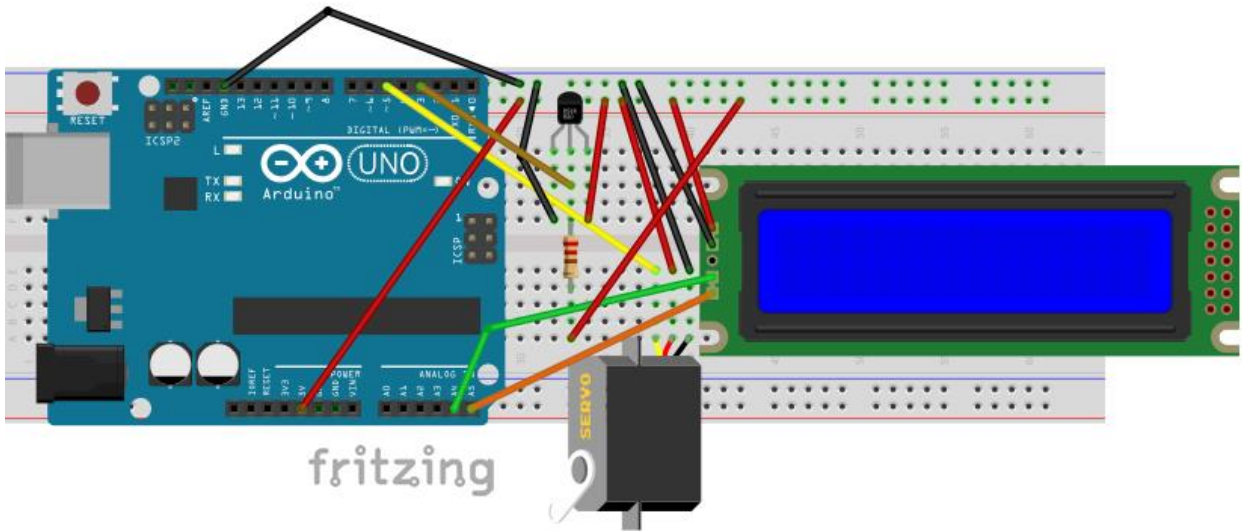
ფიგ. 54.5 მინუს გრადუსების მასშტაბირების ბლოკის პარამეტრიზაცია

ფიგ. 54.6 პლიუს გრადუსების მასშტაბირების ბლოკის პარამეტრიზაცია

ამრიგად, როდესაც ტემპერატურის გადამწოდზე ძაბვა იცვლება მინუს 55 გრადუსიდან პლიუს 125 გრადუსამდე, მაშინ სერვომრავის ანალოგურ შესასვლელზე რიცხვი იცვლება 180-დან 0-მდე და ამით მიიღწევა შესაბამისობა გასაზომი ტემპერატურის სიდიდისა ისრიანი თერმომეტრის ისრის მდგომარეობასთან. სერვომრავის პარამეტრიზაციის ფანჯარა ნაჩვენებია ფიგ. 54.7-ზე, ხოლო მოწყობილობის Fritzing სქემა ნაჩვენებია ფიგ. 54.8-ზე.

პროექტს არ ჭირდება ტეგების ზონის გამოყენება.

ფიგ. 54.7 სერვომრავის პარამეტრიზაციის ფანჯარა



ფიგ. 54.8 ისრიანი თერმომეტრის Fritzing სქემა

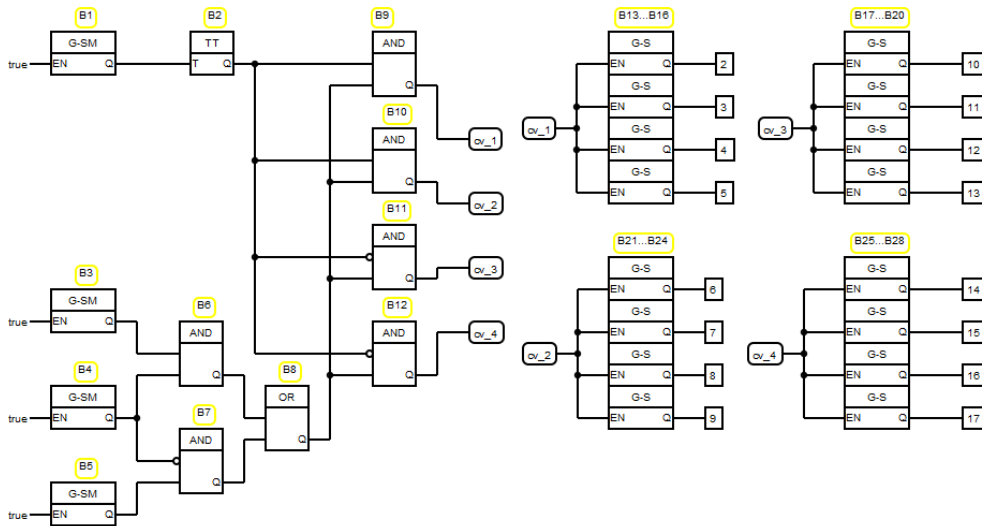
პროექტი_55 პატრულის ციმციმა



პოექტში ჩვენ გამოყენებული გვაქვს ოთხოთხი შუქდიოდებისაგან შემდგარი ოთხი ჯგუფი: პირველი ჯგუფი მიერთებულია არდუინოს მე-2, მე-3, მე-4, მე-5 კონტაქტებთან (ფიგ. 55.1) და წითელი ფერისანი არიან, მეორე ჯგუფი მიერთებულია მე-6, მე-7, მე-8, მე-9 კონტაქტებთან და ყვითელი ფერისანი არიან, მე-3 ჯგუფი მიერთებულია მე-10, მე-11, მე-12, მე-13 კონტაქტებთან და ლურჯი ფერისანი არიან და მეოთხე ჯგუფი მიერთებულნი არიან არდუინოს A1, A2, A3, A4 კონტაქტებთან და მწვანე ფერისანი არიან.

აქ გამოყენებული გვაქვს შუქდიოდების ციმციმის რეალურთან მიახლოებული ალგორითმი. კონკრეტულად, ეს ოთხი ჯგუფი გაერთიანებულია ორ ჯგუფად და ხან ერთი ჯგუფი გამოანათებს და ხან მეორე. ამ ნათების მონაცვლეობა ხდება ყოველ წამში ერთჯერ. ამ მიზანს ემსახურება სიმეტრიული მულტივიზბრატორი ბლოკი B1, რომელიც დაპროგრამებულია 250 მილიწამის ხანგრძლიობის იმპულსის გამომუშავებაზე. ამ მულტივიზბრატორის სიხშირე იყოფა ორზე TT ტრიგერის გავლით-ბლოკზე B2. მაშასადამე, ამ

ტრიგერის გამოსასვლელზე იმპულსისა და პაუზის ხანგრძლიობები ერთმანეთის ტოლია და იქნება 500 მილიწამის, ანუ ნახევარი წამის ტოლი თვითოეულის. როდესაც ამ ბლოკის გამოსასვლელზე იმპულსია ანუ ლოგ. „1“-ია, მაშინ გაღებულია B9 და B10 ლოგიკური გამრავლების ბლოკები, ხოლო B11 და B12 ლოგიკური გამრავლების ბლოკები ჩაკეტილია. როდესაც TT ტრიგერის გამოსასვლელზე პაუზაა ანუ ლოგ. „0“-ია, მაშინ ჩაკეტება B9 და B10 ბლოკები, ხოლო B11 და B12 გაიხსნება.



ფიგ. 55.1 პატრულის ციმციმციმას მუშაობის პროგრამა

ღია ლოგიკური გამრავლების ბლოკები თავიანთ გამოსასვლელებზე გაატარებენ იმ იმპულსებს, რომლებიც თავიანთ მეორე შესასვლელებზე მიეწოდებათ. ახლა ვნახოთ თუ რა სახის იმპულსები მიეწოდებათ ამ მეორე შესასვლელებზე. შევნიშნოთ, რომ ყველა ბლოკის მეორე შესასვლელი გაერთიანებულია და მიეწოდება ერთიდაიგივე სიგნალი B8 ლოგიკური შეკრების ბლოკიდან. ეს იმას ნიშნავს, რომ ეს ბლოკი გაატარებს სიგნალს ნებისმიერი შესასვლელიდან, ანუ ან B6 ან B7 ლოგიკური გამრავლების ბლოკებიდან. როგორც ფიგ. 55.1-დან ჩანს, მათ თითო შესასვლელზე მიეწოდება ერთიდაიგივე სიგნალი B4 სიმეტრიული მულტივიზრატორიდან, მაგრამ ერთერთის შესასვლელზე, კერძოდ B7-ის შესასვლელზე იგი განიცდის ინვერტირებას. ეს ნიშნავს, რომ როცა B4-ის გამოსასვლელზე იმპულსია, მაშინ ღიაა B6 ბლოკი, ხოლო როცა პაუზაა, მაშინ ღიაა B7 ბლოკი.

B4 მულტივიზრატორი დაპროგრამებულია 8 წამიან იმპულსზე და 8 წამიან პაუზაზე. მაშასადამე B4-ის გამოსასვლელზე იმპულსის დროს სიგნალი გადის B3 მულტივიზრატორიდან, B6 და B8 ბლოკის გავლით და მიეწოდება B9... B12 ბლოკების

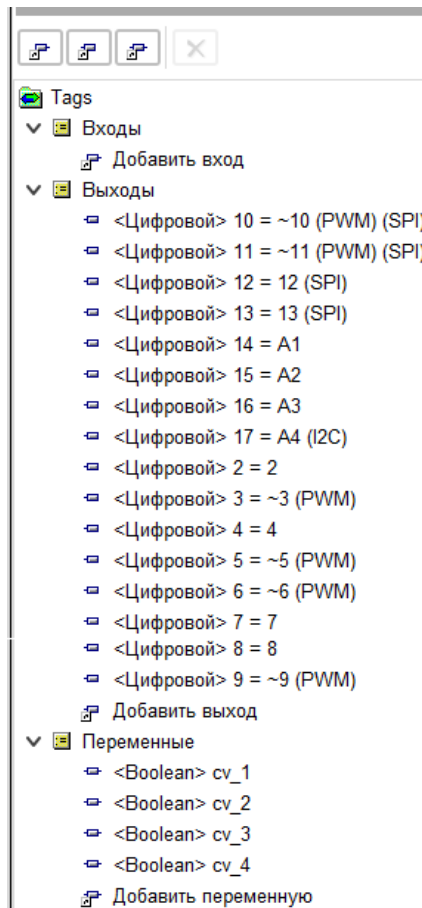
გაერთიანებულ შესასვლელებს. ეს სიგნალები კი არის 50 მილიწამიანი იმპულსები, რომელსაც გამოიმუშავებს B3 მულტივიბრატორი.

როდესაც B4-ის გამოსასვლელზე პაუზაა, მაშინ მისი ინვერსირებული სიგნალი გააღებს B7 ლოგიკური გამრავლების ბლოკს და B9... B12 ბლოკების გაერთიანებულ შესასვლელებზე სიგნალი გადის B5 მულტივიბრატორიდან B7 ბლოკისა და B8 ბლოკის გავლით. B5 მულტივიბრატორი კი მუშაობს 10 წამიან იმპულსისა და 10 წამიან პაუზის ხანგრძლიობებით.

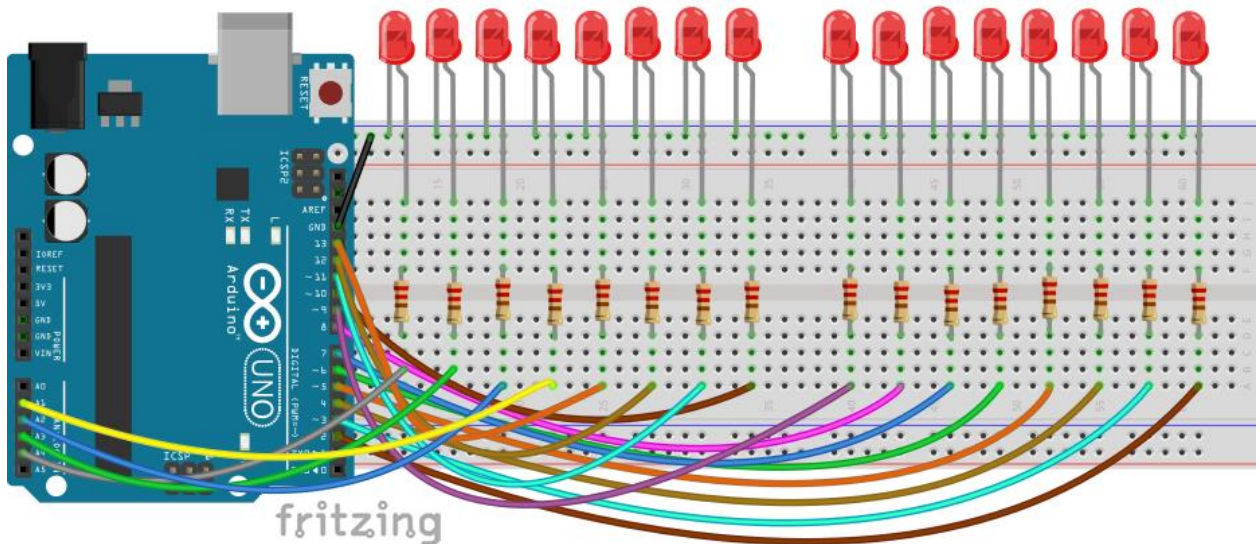
მაშასადამე B9... B12 ბლოკების გაერთიანებულ შესასვლელებზე გამომუშავდება სიგნალი პირველი 8 წამის განმავლობაში ყოველ 50 მილიწამში ერთჯერ, ხოლო მეორე 8 წამის განმავლობაში - ყოველ 10 მილიწამში ერთჯერ. ეს გამომუშავებული სიგნალები კი B9... B12 ბლოკებზე ისევ განიცდიან კომუტაციას და მიეწოდება ხან პირველ ჯგუფს შუქდიოდებისა და ხან მეორე ჯგუფს. მაგრამ ეს სიგნალები სხვადასხვა ხანგრძლიობისა იქნება თუ არ დავიყვანეთ ერთიდაიგივე ხანგრძლიობებზე. ამას ვახდენთ ერთვიბრატორების გამოყენებით B13... B16, B17... B20, B21 ... B24, B25 ... B28. ეს ბლოკები, იმისდამოუხედავად, თუ რა ხანგრძლიობის იმპულსები მიეწოდება მათ შესასვლელებზე, ყოველი შესასვლელი იმპულსის მიწოდებისას გამოიმუშავებს ერთიდაიგივე ხანგრძლიობის იმპულსს ხანგრძლიობით 100 მლწ. ამით მიიღწევა შუქდიოდებზე მოდებიული იმპულსების ხანგრძლიობების ერთნაირობა.

დასასრულს, ნახაზის გადატვირთვის აცილების მიზნით შემოტანილი გვაქვს ოთხი ცვლადი cv_1, cv_2, cv_3, cv_4 და თვითოეული ეს ცვლადი შეერთებულია ოთხი ერთვიბრატორისაგან შედგენილ თითო ჯგუფთან შესაბამისად.

საბოლოო ჯამში მივიღებთ ასეთ სურათს. პატრულის ციმციმა მუშაობს შემდეგნაირად: არის შუქდიოდების ორი ჯგუფი თვითოეულში რვა-რვა შუქდიოდით, სხვადასხვა ფერებით გაწყობილნი. პირველი ნახევარ წამის განმავლობაში ერთი ჯგუფი ანათებს, მეორე ნახევარი წამის განმავლობაში მეორე ჯგუფი, შემდეგ ისევ პირველი და ა.შ. ასე ხდება პირველი 8 წამის განმავლობაში, მეორე რვა წამის განმავლობაში ისევ ხდება ნახევარ წამიანი პერიოდებით შუქდიოდის ჯგუფების გადართვა გადმორთვა, მაგრამ ამას ემატება ის, რომ ამ მეორე რვა წამის განმავლობაში ანთებული შუქდიოდის ჯგუფები განიცდიან აგრეთვე რამდენჯერმე სწრაფ ჩაქრობა ანთებას და ისეთი ეფექტია, რომ თითქოსდა შუქები ფეთქავს (პარპალებს), როგორც სადგილი აქვს რეალურ პატრულის ციმციმაში. პროექტის ტეგების ზონა ასახულია ფიგ. 55.2-ზე, ხოლო პროექტის Fritzing სქემა ფიგ. 55.3-ზე.



ფიგ. 55.2 ტეგების ზონა

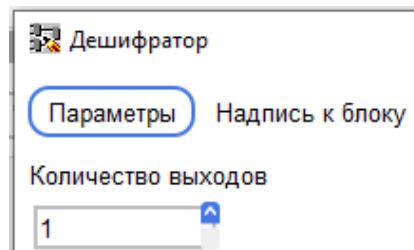


ფიგ. 55.3 პატრულის ციმციმას Fritzing სქემა

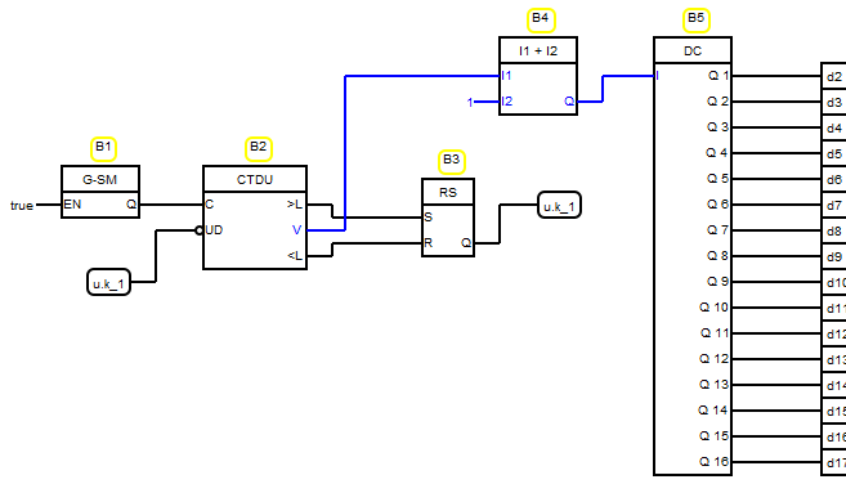
პროექტი_56 დეშიფრატორი



დეშიფრატორის ბლოკის დანიშნულებაა იმ გამოსასვლელზე ლოგიკური ერთიანის გამოცემა, რომლის ნომერიც შეესაბამება ბლოკის შესასვლელზე მიწოდებული ანალოგური სიგნალის მნიშვნელობას. მაგ. თუ ბლოკის I შესასვლელზე მიეწოდება ანალოგური სიგნალი 7, მაშინ ამ ბლოკის Q7 გამოსასვლელზე იქნება ლოგიკური ერთიანი, ხოლო ყველა დანარჩენზე - ლოგიკური ნოლი. დეშიფრატორის ბლოკი იმყოფება FLProg-ის ბლოკების ბიბლიოთეკის ბიტების ოპერაციის-Операции с битами საქალდეში სახელწოდებით დეშიფრატორი-Дешифратор. თუ გადმოვიტანთ ამ ბლოკს პროგრამის სამუშაო ზონაში და დავაწკაპუნებთ მასზედ თავის მარცხენა ღილაკით ორჯერ, მაშინ გაიხსნება ამ ბლოკის პარამეტრიზაციის ფანჯარა (ფიგ. 56.1), სადაც გამოსასვლელების რაოდენობის ველში-Количество выходов შეგვიძლია ჩავწეროთ ნებისმიერი რიცხვი. ჩვენს შემთხვევაში ჩაწერილი გვაქვს რიცხვი 16, ეს ნიშნავს, რომ ბლოკს გაუჩნდება 16 გამოსასვლელი Q1...Q16 იხილეთ ფიგ. 56.2 ბლოკი B5.



ფიგ. 56.1 დეშიფრატორის ბლოკის პარამეტრიზაციის ფანჯარა



ფიგ. 56.2 დეშიფრატორის მართვის პროგრამა

ფიგ. 56.2-ზე ასახული პროგრამა ფაქტიურად წარმოადგენს 16 შუქდიოდისაგან შექმნილ გირლანდას, სადაც ანთებული შუქდიოდები წანაცვლებიან სათითაოდ ჯერ პირველი გამოსასვლელიდან მე-16 მდე, შემდეგ შეიცვლიან მიმართულებას და წანაცვლებენ მე-16-დან პირველისკენ, შემდეგ ისევ შეიცვლიან მიმართულებას და ა.შ. აღნიშნული ალგორითმის რეალიზაციისთვის გამოყენებული გვაქვს გენერატორის ბლოკი B1, რომელიც დაპროგრამებული გვაქვს როგორც სიმეტრიული მულტივიზრატორი 500 მლიწამის იმპულსის ხანგრძლიობით. ამ გენერატორის გამოსასვლელი შეერთებულია მთვლელის ბლოკის B2 შესასვლელთან, რომელიც თავის მხრივ დაპროგრამებულია ფიგ. 56.3-ზე წარმოდგენილი პარამეტრიზაციის ფანჯრის შესაბამისად.

როგორც ფიგურიდან ჩანს, მთვლელის თვლის მიმართულება განსაზღვრულია შესასვლელზე-Bx0d მიწოდებული სიგნალით. თუ იგი ლოგიკური ერთია, მაშინ მთვლელი ითვლის დადებითი მიმართულებით (ზრდაზე), ხოლო ლოგიკური ნოლის შემთხვევაში მთვლელი დაითვლის უარყოფითი მიმართულებით ანუ კლებაზე. ამის შესაბამისად, თუ ჩვენ საწყის მომენტში შევძლებთ დავსვათ მთვლელის შესასვლელზე UD-ზე ლოგ. „1“ და ეს მოხდება ავტომატურად, რადგან არდუინოს გაშვების მომენტში ეს მთვლელიც და RS ტრიგერიც B3 ავტომატურად დგება ნოლზე, რაც გამოიწვევს ინვერსირების შემდეგ მთვლელის UD-ზე შესასვლელზე ლოგ. „1“-ის დაჯენას.

Счетчик

Параметры Надпись к блоку

Тип счетчика

UP (на увеличение)

DOWN (на уменьшение)

Вход

Два входа

Входы

Предустановка

Сброс

Действия

При достижении верхней границы Вход Константа

Установить в единицу выход >L

Установить текущее значение

Остановить счёт вверх

При достижении нижней границы Вход Константа

Установить в единицу выход < L

Установить текущее значение

Остановить счёт вниз

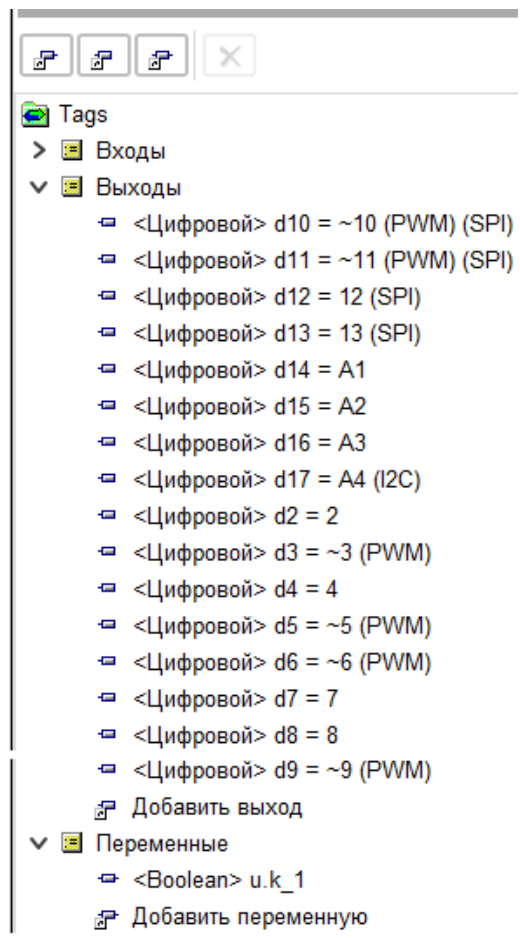
Дополнительно

Считать от 0 до 4294967295 (по умолчанию от -32767 до 32767)

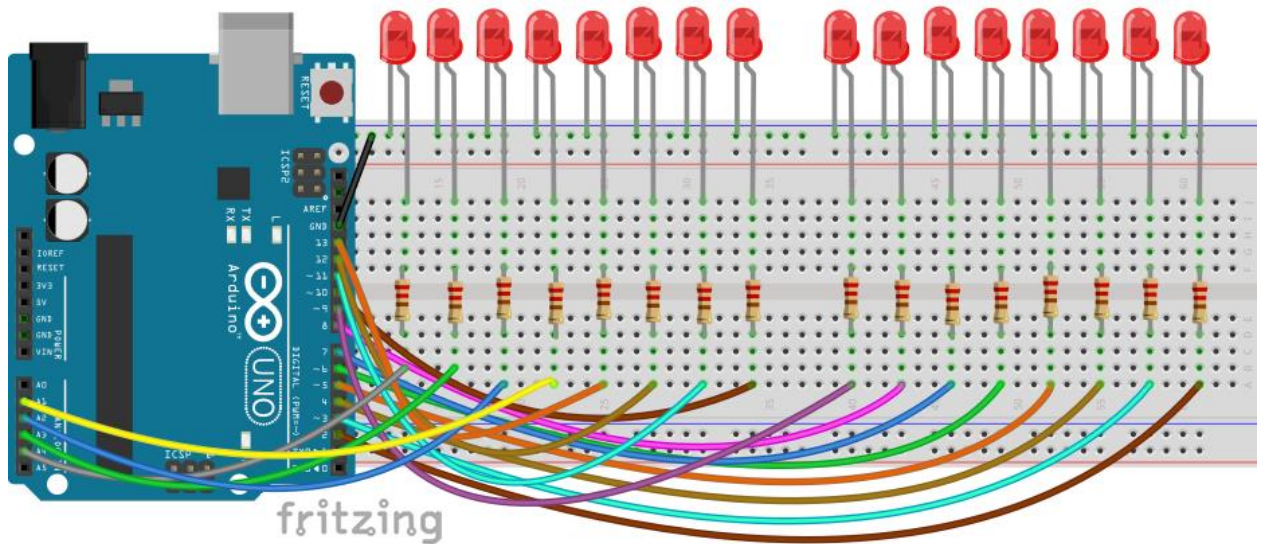
ფიგ. 56.3 მთვლელის პარამეტრიზაციის ფანჯარა

ამის შემდეგ, როდესაც მთვლელის C შესასვლელზე გამოჩნდება იმპულსები, მთვლელი დაითვლის ამ იმპულსებს პირდაპირი მიმართულებით და როდესაც მთვლელის მნიშვნელობა გაუტოლდება 16-ს, მაშინ მთვლელის გამოსასვლელზე გამომუშავდება >L სიგნალი, რომელიც დააყენებს RS ტრიგერს ერთიანის მდგომარეობაში, რაც გამოიწვევს მთვლელის UD-ზე შესასვლელზე ლოგ. „0“-ის დაჯენას და იგი გადავა უკუ თვლის რეჟიმში, როდესაც მთვლელი დაითვლის ნოლამდე, მაშინ მის გამოსასვლელზე გამომუშავდება <L სიგნალი, რომელიც ისევ დააენებს RS ტრიგერს ნოლოვან მდგომარეობაში, მთვლელი ისევ გადავა პირდაპირი თვლის რეჟიმში და ა.შ.შ. დაუსრულებლად. თუ დავაკვირდებით, მთვლელის ანალოგური გამოსასვლელის რიცხვითი მნიშვნელობა იცვლება ნოლის მდგომარეობიდან 15-მდე. დეშიფრატორს კი აქვს გამოსასვლელი ერთიდან თექვსმეტამდე. ამიტომ რომ გავაწონასწოროთ მთვლელის გამოსასვლელი და დეშიფრატორის შესასვლელი

ერთმანეთთან, საჭიროა მთვლელის რიცხვს დავიმატოთ ერთი, ამას ვახერხებთ შეკრების ბლოკის B4-ის გამოყენებით. შესაბამისად ამისა, როდესაც მთვლელის მდგომარეობა იცვლება ყოველი წამის განმავლობაში ნოლიდან 15-მდე დემიფრატორის გამოსასვლელებზე მიერთებული შუქდიოდების ნათებები წანაცვლდებიან პირველიდან მე-16-მდე, ხოლო როცა მთვლელი შეიცვლის თვლის მიმართულებას 15-დან ნოლამდე, მაშინ შუქდიოდის ნათებები წანაცვლდებიან მე-16 ნომრიდან ნოლამდე. დემიფრატორის გამოსასვლელები მიერთებულნი არიან არდუინოს კონტაქტებთან ფიგ. 56.4-ზე ასახული ტეგების ზონის მიხედვით, ხოლო დემიფრატორის მართვის Fritzing სქემა - ფიგ. 56.5-ზე.

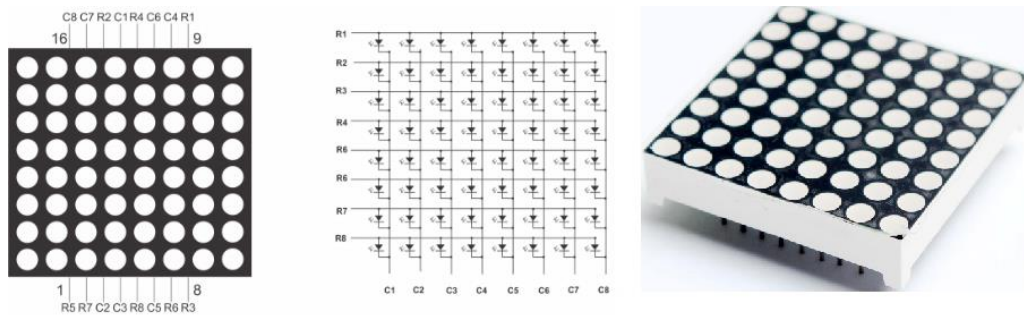


ფიგ. 56.4. დემიფრატორის მართვის ტეგების ზონა

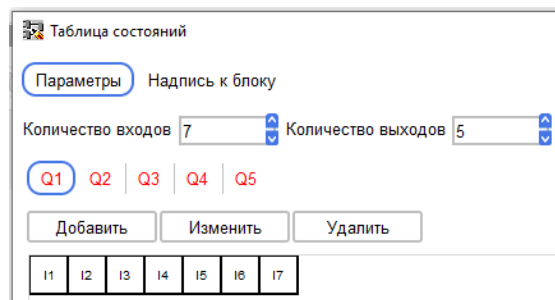


ფიგ. 56.5 დემოფრატორის მართვის Fritzing სქემა

პროექტი_57 მდგომარეობათა ცხრილი

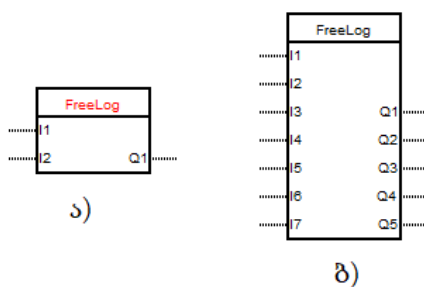


მდგომარეობათა ცხრილის ბლოკის დანიშნულებაა შესასვლელების მდგომარეობების გამოსასვლელების მდგომარეობებთან შესაბამისობის ცხრილის შექმნა. ბლოკი მდებარეობს ბაზური ელემენტების საქადალდეში-Базовые элементы სახელწოდებით მდგომარეობათა ცხრილი-Таблица состояний. ბლოკის გაწყობა შესაძლებელია ბლოკის რედაქტორის დახმარებით, რომელიც მიიღება პროგრამის სამუშაო ზონაში გადმოტანილ ბლოკზე თავის მარცხენა ღილაკით ორჯერ დაწკაპუნებით ფიგ. 57.1



ფიგ. 57.1 მდგომარეობათა ცხრილის ბლოკის პარამეტრიზაციის ფანჯარა

ბლოკის პარამეტრებში შეგვიძლია ჩავსვათ აუცილებელი რაოდენობის შესასვლელები და გამოსასვლელები-Количество входов და Количество выходов. ფიგ. 57.2 ა)-ზე ნაჩვენებია მდგომარეობათა ცხრილის ბლოკი სამუშაო ზოლში გადმოტანის საწყისწ მომენტში ბ)-ზე ბლოკის შესასვლელებისა და გამოსასვლელების ჩასმის შემდეგ. ამ შემთხვევაში შესასვლელების რაოდენობა განვსაზღვრეთ შვიდით, ხოლო გამოსასვლელების რაოდენობა ხუთით. აქ ყურადღება უნდა მიექცეს იმ გარემოებას, რომ შესასვლელთა რაოდენობის ცვლილების დროს იკარგება ყველა ადრე შეყვანილი მდგომარეობათა სტრიქონები.

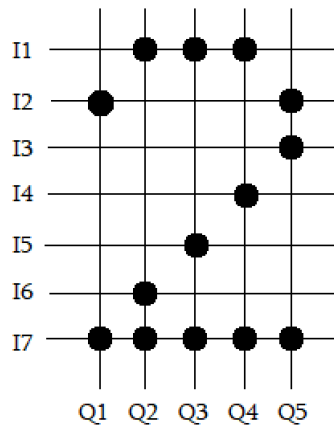


ფიგ. 57.2 მდგომარეობათა ცხრილის ბლოკი საწყისში ა) და შემდეგ ბ)

მდგომარეობათა ცხრილში შეიტანება სტრიქონები მდგომარეობათა იმ კომბინაციებით შესასვლელზე, რომლის დროსაც არჩეულ გამოსასვლელზე წარმოიშვება ლოგიკური ერთიანი. სხვა რომელიმე კომბინაციებზე კი არჩეულ გამოსასვლელზე უნდა იყოს ლოგიკური ნოლი. ანალოგიურად ხდება ცხრილის შევსება დანარჩენი გამოსასვლელებისთვისაც. ნებისმიერი გამოსასვლელისთვის შევსებულ უნდა იქნას მდგომარეობათა ერთი სტრიქონი მაინც. მდგომარეობათა სტრიქონების დამატება ან ცვლილება ხდება მდგომარეობათა სტრიქონების დიალოგით.

ახალი სტრიქონის შექმნისას სტრიქონების დიალოგი ივსება ნოლებით. თუ კი დავაწკაპუნებთ რომელიმე მნიშვნელობის უჯრედზე სტრიქონის მდგომარეობათა დიალოგში, მაშინ უჯრედის მნიშვნელობა იცვლება საწინააღმდეგოზე. ე.ი. თუ ჩაწერილი იყო უჯრედში ნოლი (საწყის მომენტში ყოველთვის ჩაწერილია ნოლი), მაშინ უჯრედში ჩაიწერება 1. თუ უჯრედში სადაც ჩაწერილია 1 გვინდა შეცვალოთ ნოლზე, მაშინაც უნდა დავაწკაპოთ ამ 1-ზე და იგი შეიცვლება ნოლით.

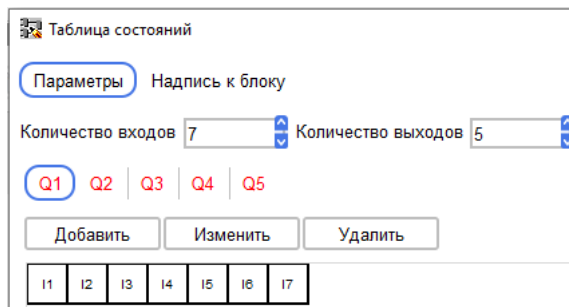
ცხრილის შევსების პროცედურების თვალსაჩინოებისათვის განვიხილოთ რომელიმე სიმბოლოს, მაგ. ორის მაგალითი (ფიგ. 57.3), რომელიც ჩაწერილი 7x5 მატრიცაში.



ფიგ. 57.3 სიმბოლოს კონტური

თუ მატრიცაში ჰორიზონტალურ კოორდინატებს შევუსაბამებთ მდგომარეობათა ცხრილის შესასვლელებს, ხოლო ვერტიკალურ კოორდინატებს მდგომარეობათა ცხრილის გამოსასვლელებს და სიმბოლოს კონტურის წერტილებს შევუსაბამებთ ლოგიკურ ერთიანებს, მაშინ ეს დაგვეხმარება მდგომარეობათა ცხრილის შევსების პროცედურაში.

დავიწყით Q1 გამოსასვლელით. მას შემდეგ როდესაც შევავსებთ ცხრილის შესასვლელოებისა და გამოსასვლელების ველებს და დავხურავთ ფანჯარას (ფიგ. 57.1), შემდეგ ისევ დავაწკაპებთ უკვე ახალი ფორმის ბლოკზე (ფიგ. 57.2 ბ), ეკრანზე გამოვა ფანჯარა (ფიგ. 57.4). ფანჯარაში Q1 გამოსასვლელი მონიშნულია სიჩუმით და დავტოვოთ როგორც არის და დავიწყით მისით.



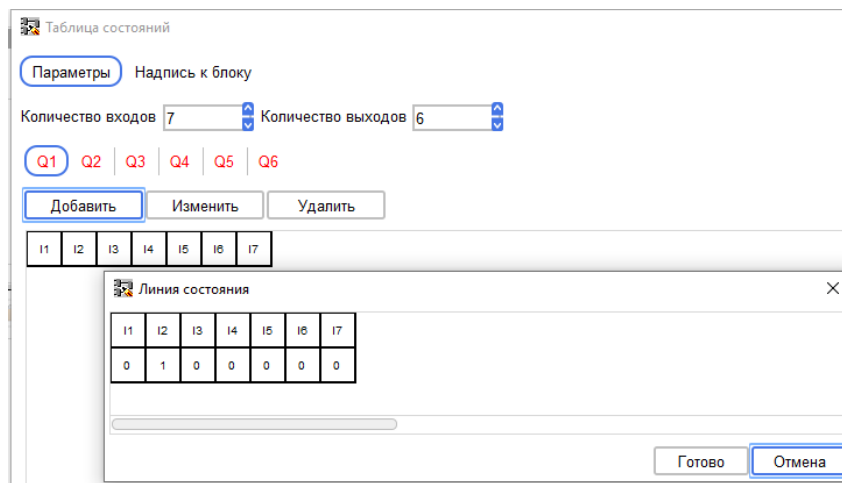
ფიგ. 57.4 მდგომარეობათა ცხრილი 7 შესასვლელსა და 5 გამოსასვლელზე

როგორც ფიგ. 57.3-დან ჩანს Q1 გამოსასვლელმა უნდა მიიღოს ლოგიკური ერთიანის მნიშვნელობა როდესაც არსებობს I2 შესასვლელი ან I7 შესასვლელი. ჩავსვათ ახლა ეს პირობები მდგომარეობათა ცხრილში. დავაწკაპუნოთ დამატების ღილაკზე-Добавить, ეკრანზე გამოვა ახალი ფანჯარა (ფიგ. 57.5).

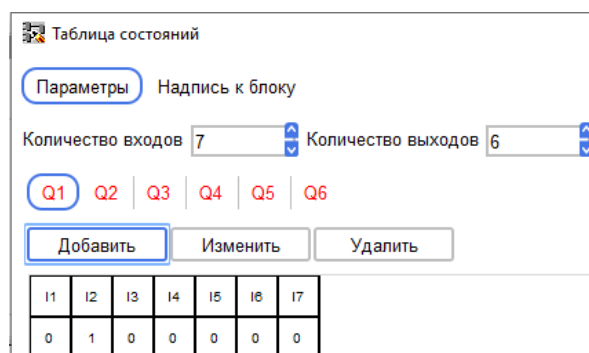


ფიგ. 57.5 მდგომარეობათა ცხრილში ჩაწერა Q1 გამოსასვლელისთვის

ამ ახალ ფანჯარაში დავაწკაპუნოთ ერთჯერ თავის მარცხენა ღილაკით იმ ნოლზე, რომელიც მოთავსებულია I2 შესასვლელის ქვემოთ მოთავსებულ უჯრედში. ამით ნოლიანი შეიცვლება ერთიანით (ფიგ. 57.6). დავაწვეთ ღილაკს Готово და ეკრანზე დარჩება ფანჯარა იხ. ფიგ. 57.7. ამის შემდეგ ხელმეორედ დავაწვეთ ღილაკს Добавить, კიდევ გამოვა ახალი ფანჯარა (ფიგ. 57.8).



ფიგ. 57.6 მდგომარეობათა ცხრილში ჩაწერა Q1 გამოსასვლელისთვის

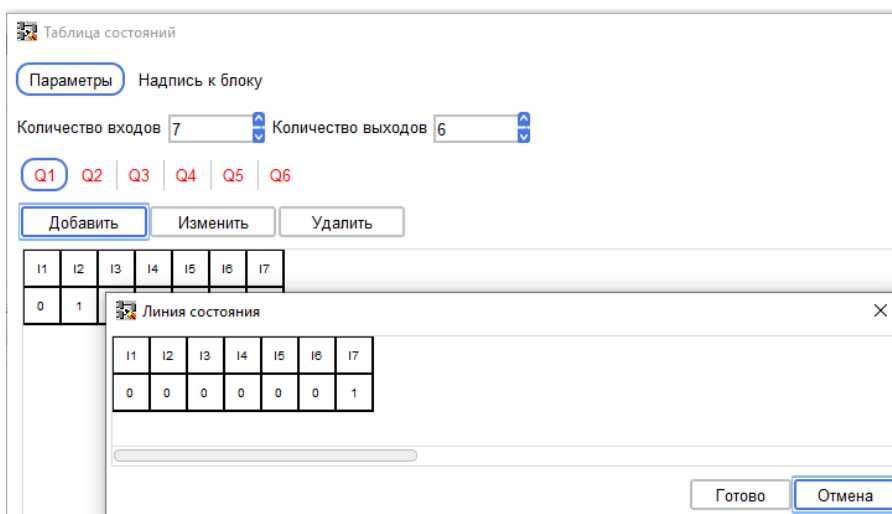


ფიგ. 57.7 მდგომარეობათა ცხრილში ჩაწერა Q1 გამოსასვლელისთვის

ამ ახალ ფანჯარაში დავაწკაპუნოთ ერთჯერ თავის მარცხენა ღილაკით იმ ნოლზე, რომელიც მოთავსებულია ამჯერად I7 შესასვლელის ქვემოთ მოთავსებულ უჯრედში. ამით ნოლიანი შეიცვლება ერთიანით (ფიგ. 57.8). დავაწვეთ ღილაკს Готово.



ფიგ. 57.8 მდგომარეობათა ცხრილში ჩაწერა Q1 გამოსასვლელისთვის



ფიგ. 57.9 მდგომარეობათა ცხრილში ჩაწერა Q1 გამოსასვლელისთვის

ფანჯარა მიიღებს საბოლოო სახეს ფიგ. 57.10, რითაც მდგომარეობათა ცხრილის შევსება Q1 გამოსასვლელისთვის დამთავრებული იქნება.

Таблица состояний

Параметры Надпись к блоку

Количество входов 7 Количество выходов 5

Q1 Q2 Q3 Q4 Q5

Добавить Изменить Удалить

I1	I2	I3	I4	I5	I6	I7
0	1	0	0	0	0	0
0	0	0	0	0	0	1

ფიგ. 57.10 მდგომარეობათა ცხრილში ჩაწერა Q1 გამოსასვლელისთვის

ე.ი სიმვოლოს კონტურს Q1 ვერტიკალურ ხაზზე (გამოსასვლელზე) აქვს ორი წერტილი: I2 და I7 ჰორიზონტალურ ხაზებზე (შესასვლელებზე) და ეს ადეკვატურად ასახულია ფიგ. 57.10 მდგომარეობათა ცხრილში.

ასევე აიგება მდგომარეობათა ცხრილები Q2, Q3, Q4 და Q5 გამოსასვლელისთვის (იხ. ფიგ. 57.11, ფიგ. 57.12, ფიგ. 57.13 და ფიგ. 57.14 შესაბამისად).

Таблица состояний

Параметры Надпись к блоку

Количество входов 7 Количество выходов 5

Q1 Q2 Q3 Q4 Q5

Добавить Изменить Удалить

I1	I2	I3	I4	I5	I6	I7
1	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

ფიგ. 57.11 მდგომარეობათა ცხრილში ჩაწერა Q2 გამოსასვლელისთვის

Таблица состояний

Параметры Надпись к блоку

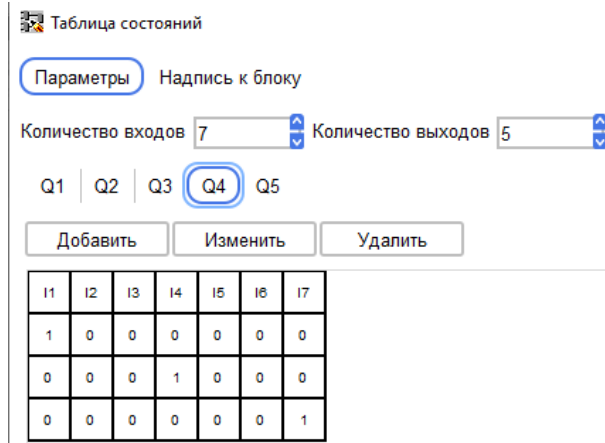
Количество входов 7 Количество выходов 5

Q1 Q2 Q3 Q4 Q5

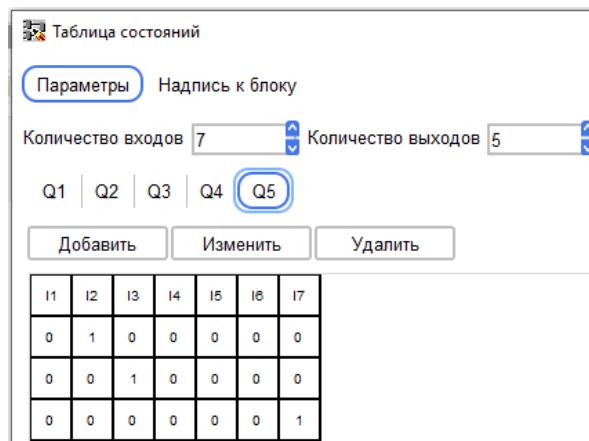
Добавить Изменить Удалить

I1	I2	I3	I4	I5	I6	I7
1	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	1

ფიგ. 57.12 მდგომარეობათა ცხრილში ჩაწერა Q3 გამოსასვლელისთვის

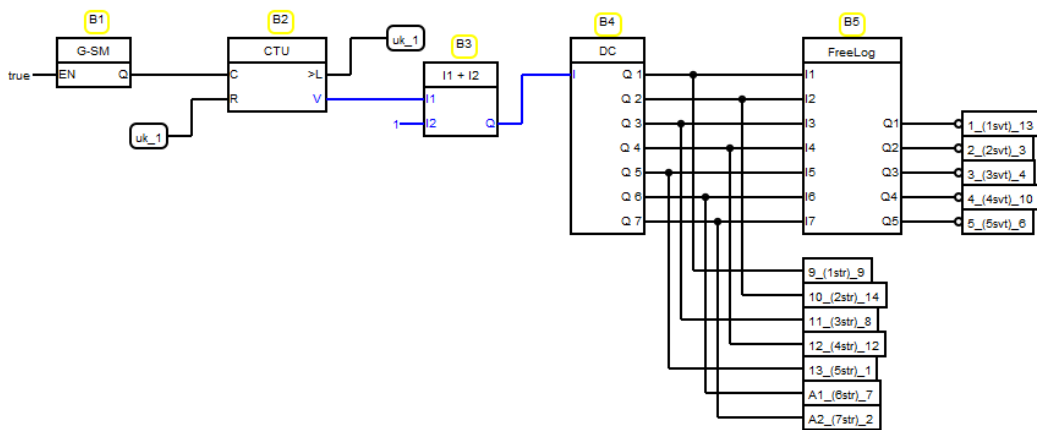


ფიგ. 57.13 მდგომარეობათა ცხრილში ჩაწერა Q4 გამოსასვლელისთვის



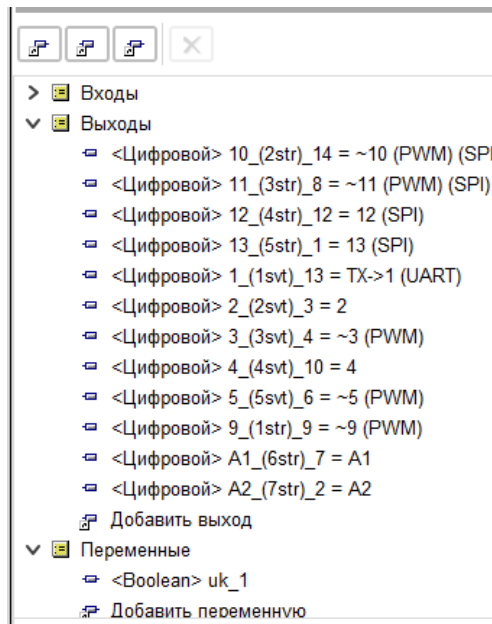
ფიგ. 57.14 მდგომარეობათა ცხრილში ჩაწერა Q5 გამოსასვლელისთვის

ამ პროექტის თვალსაჩინოებისათვის უკეთესი იქნება თუ ჩვენ ინდიკაციისათვის გამოვიყენებთ 8x8 შუქდიოდურ მატრიცას და შევადგენთ ისეთ პროგრამას, რომელიც დინამიურად და ციკლურად გამოიწვევს ყველა მნიშვნული შუქდიოდის ანთებას. ე.ი. მატრიცაზე იმ ორიანის სიმბოლოს კონტურის გამონათებას.



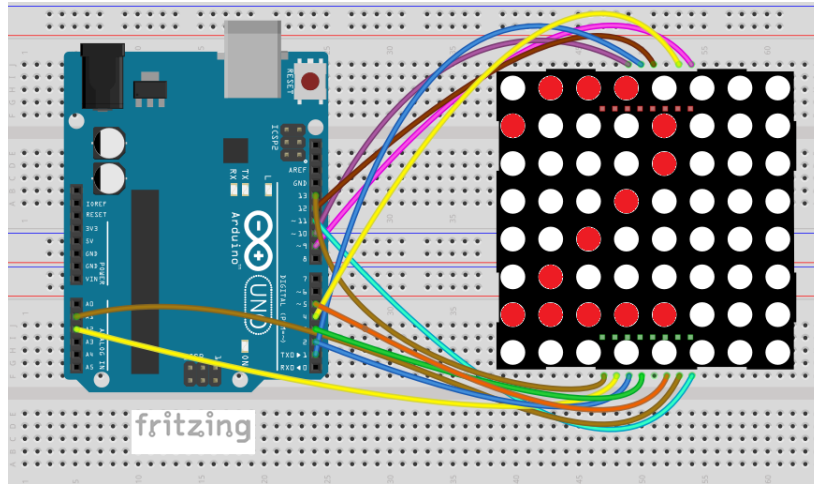
ფიგ. 57.15 მდგომარეობათა ცხრილის ბლოკის მართვის პროგრამა

ასეთი პროგრამა წარმოდგენილია ფიგ. 57.15-ზე. აქ ჩვენ აღებული გვაქვს გენერატორის ბლოკი B1, რომელიც დაპროგრამებულია სიმეტრიული მულტივიბრატორის რეჟიმში განუწყვეტელი მუშაობისათვის, იმპულსების ხანგრძლიობით 100 მკწ. გენერატორის მიერ გამომუშავებული იმპულსები მიეწოდება მთვლელს B2, რომელიც დაპროგრამებულია ციკლური მუშაობის რეჟიმში სამუშაოდ, რომელიც ითვლის პირდაპირი თვლის რეჟიმში და როდესაც მასზედ მიეწოდება მე-7 იმპულსი, გამომუშავდება მთვლელის სიგნალი $>L$, რომელიც uk_1 უკუკავშირის ცვლადის დახმარებით დააყენებს მთვლელს ნულოვან მდგომარეობაში და შემდეგ თვლა გაგრძელდება თავიდან. მთვლელის ანალოგურ გამოსასვლელზე დაფიქსირდება ანალოგური რიცხვები: 0, 1, 2, 3, 4, 5, 6. მაგრამ იმისათვის, რომ დეშიფრატორმა B4 გამოიმუშაოს სიგნალები Q1, Q2, Q3, Q4, Q5, Q6, Q7 საჭიროა შესაბამისობა მთვლელის გამოსასვლელისა დეშიფრატორის შესასვლელთან. ეს კი მიიღწევა თუ ჩვენ მთვლელის ანალოგურ გამოსასვლელს დავუმატებთ მუდმივ რიცხვს ერთს. ამისთვის ვიყენებთ არითმეტიკულ ბლოკს B3-ს. ახლა საჭიროა შევქმნათ არდუინოს გამოსასვლელები-Выходы, რომლებსაც შევუერთებთ შუქდიოდური მატრიცის ფეხებს. ასეთი გამოსასვლელები შექმნილია ტეგების ზონაში წარმოდგენილია ფიგ. 57.16. ჩვენს მიერ შექმნილი გამოსასვლელები შეიცავს შვიდ სტრიქონს: 1str, 2str, 3str, 4str, 5str, 6str, 7str და 5 სვეტს 1svt, 2svt, 3svt, 4svt, 5svt. შევაერთებთ სტრიქონებს დეშიფრატორის გამოსასვლელებთან ხოლო სვეტებს B5 ბლოკის გამოსასვლელებთან ფიგ. 57.16-ზე წარმოდგენილი სქემის მიხედვით. მიაქციეთ ყურადღება, რომ ამ გამოსასვლელებზე მაგ. 9_(1str)_9 პირველი ციფრი 9 აღნიშნავს არდუინოს კონტაქტის ნომერს, ხოლო მეორე ციფრი, ამ შემთხვევაში იგივე 9 აღნიშნავს შუქდიოდური მატრიცის ფეხის ნომერს.



ფიგ. 57.16 ტეგების ზონის ფანჯარა

ასეთი პროგრამის შექმნის შემდეგ ჩვენ უნდა ჩავტვირთოთ პროგრამა არდუინოში. გენერატორი დაიწყებს იმპულსების გენერირებას, მთვლელი ითვლის ამ იმპულსებს ციკლურ რეჟიმში, დეშიფრატორი დაიწყებს მატრიცის სტრიქონების სკანირებას, ჯერ პირველ სტრიქონზე დააჯენს ძაბვის მაღალ დონეს (+5 ვ) და დანარჩენებზე დაბალ დონეს (0 ვ), შემდეგ მეორე სტრიქონზე დააჯენს მაღალ დონეს და დანარჩენებზე დაბალ დონეს და ა.შ.შ. განუწყვეტლივ ციკლურად. ერთდროულად ამისა მდგომარეობათა ცხრილის ბლოკი B5 კითხულობს დეშიფრატორის მიერ ფორმირებული სტრიქონების ჩანაწერებს თავის თავში და გამოსცევს თავის გამოსასვლელებზე Q1, Q2, Q3, Q4, Q5 სიგნალებს რომლებიც შემდეგ ინვერტირდებიან და მიეწოდებიან მატრიცის სვეტებს. ბლოკის გამოსასვლელებზე, იქ სადაც ნოლები ფიქსირდება ინვერტირების პროცედურა გადააქცევს მას ერთიანად ანუ მაღალ დონედ. სტრიქონშიც მაღალი დონეა სვეტშიც, დენი შუქდიოდში არ გავა და შუქდიოდი არ ანათებს. მაგრამ როდესაც B5 ბლოკი თავის ჩანაწერში შეხვდება ერთიანს, ინვერტირების შემდეგ იგი გადაიქცევა ნოლად, ანუ დაბალ დონეთ სტრიქონში მაღალი დონეა, სვეტში დაბალი დონე, შესაბამის შუქდიოდში გაივლის დენი და შუქდიოდი გაანათებს. რადგან შუქდიოდის ნათებები ხდება ციკლურად ყოველ 100 მკწ-ში ერთჯერ, ამიტომ მთელი სიმბოლოს სკანირებას დასჭირდება 700 მიკროწამი და შემდეგ პროცესი განმეორდება ციკლურად თავიდან, ამიტომ ადამიანის თვალი ასეთ სწრაფ მონაცვლეობას აღიქვავს როგორც უწყვეტ კონტურს სიმბოლოსას, და ჩვენს შემთხვევაში მატრიცაზე გამონათდება სიმბოლო 2. მოწყობილობის Fritzing სქემა ასახულია ფიგ. 57.17-ზე.

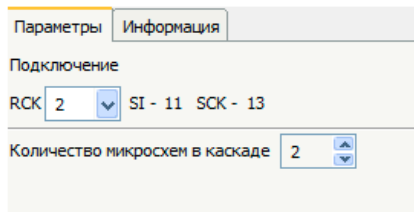
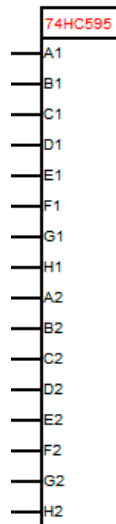


ფიგ. 57.18 მდგომარეობათა ცხრილის მართვის Fritzing სქემა

პროექტი_58 გაფართოების მიკროსქემა 74HC595

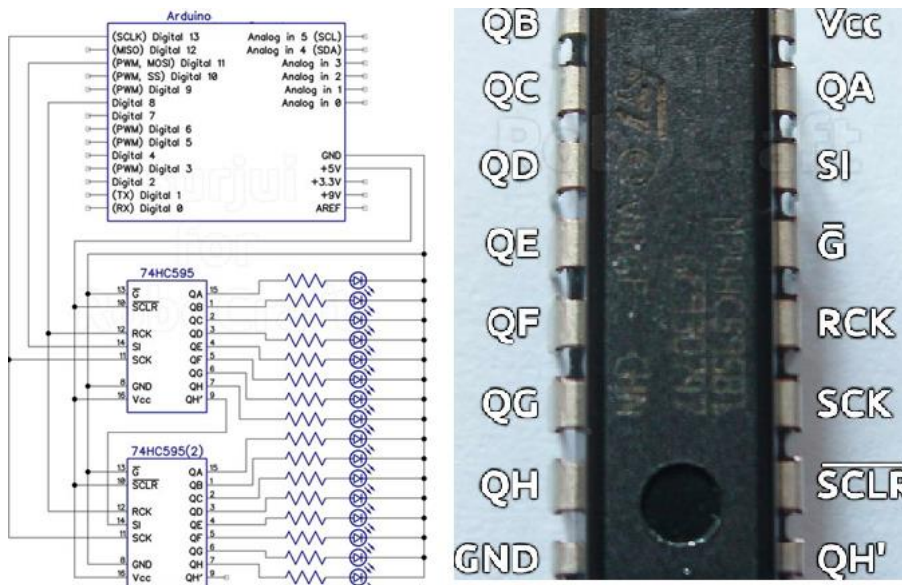


გამომყვანების გაფართოების ბლოკის დანიშნულებაა 74HC595 მიკროსქემის გამოყენების უზრუნველყოფა სხვადასხვა სახის პროექტებში, რომელიც წარმოადგენს წანაცვლების რეგისტრს და მუშაობს სალტეზე SPI. ბლოკის შესასვლელზე მიწოდებული ლოგიკური დონეები აისახება მიკროსქემის გამოსასვლელებზე. ეს მიკროსქემები შეიძლება შეერთებულნი იყოს კასკადურად და ამის შესაბამისად გამოსასვლელების რაოდენობა შეიძლება გაზრდილ იქნას მნიშვნელოვნად. ბლოკი მდებარეობს FLProg-ის ბლოკების ბიბლიოთეკის გაფართოების მიკროსქემების-Микросхемы расширения საქაღალდეში სახელწოდებით გამოსასვლელების გამაფართოებელი-Расширитель выходов 74HC595. ბლოკის გამოყენება ძალზედ მარტივია. ლოგიკური დონეები, რასაც მივაწვდით ბლოკის შესასვლელებზე აისახება მიკროსქემის გამოსასვლელებზე. მაგრამ ბლოკის აწყობებში აუცილებლად ნაჩვენები უნდა იქნას არდუინოს ის კონტაქტი, რომელზედაც მიერთებული იქნება მიკროსქემის მართვის შესასვლელი RCK და მიკროსქემების რაოდენობა კასკადში (ფიგ. 58.1).



ფიგ. 58.1. გამოსასვლელების გაფართოვების ბლოკის აწყობა

ამ პროექტში ჩვენ ვაჩვენებთ გამოსასვლელების გაფართოვების ორი მიკროსქემის 74HC595 გამოყენებას შემდეგი სახის ალგორითმის შესასრულებლად. ფიგ. 58.2-ზე ასახულია ორკასკადიანი ელექტრული პრინციპიალური სქემა, სადაც ნაჩვენებია მიკროსქემების ელექტრული კავშირები არდუინოსთან, ერთმანეთთან და შუქდიოდებთან. ნახაზიდან ჩანს რომ მხოლოდ არდუინოს სამი კონტაქტით შეიძლება ვმართოთ თექვსმეტი შუქდიოდი. აქვე თვალსაჩინოებისათვის და მიკროსქემის ფეხების ნომერაციისათვის მოყვანილია თვითონ მიკროსქემის ფიზიკური გამოსახულება და მისი ფეხების აღნიშვნები.



ფიგ. 58.2. არდუინოს შეერთება 74HC595 მიკროსქემებთან

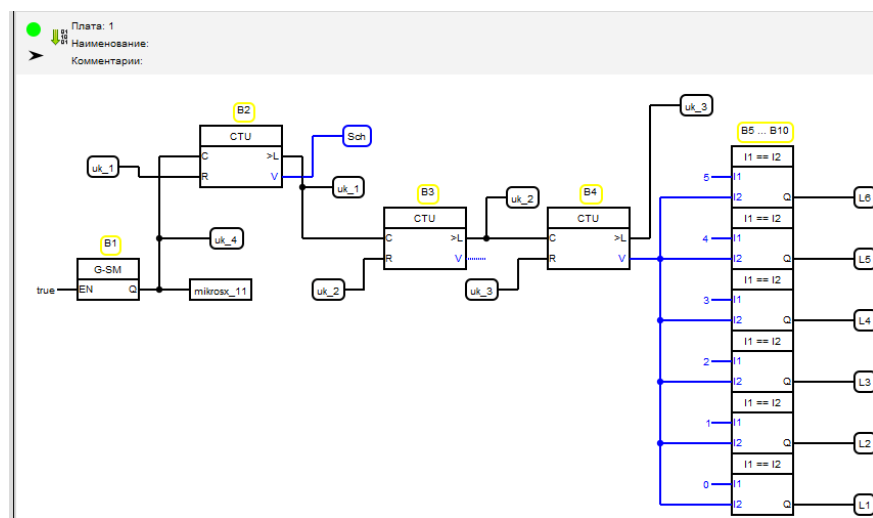
როგორც ნახაზიდან ჩანს, მიკროსქემების კვებები შეერთებულია არდუინოს კვების გამოსასვლელებთან. მიკროსქემების სინქრონიზაციის ფეხები SCK (11) გაერთიანებულია და შეერთებულია არდუინოს მე-13 კონტაქტთან. მიკროსქემის ფიქსატორები RCK (12) ასევე გაერთიანებულია და შეერთებულია არდუინოს მე-8 კონტაქტთან. პირველი მიკროსქემის მონაცემების შესასვლელი SI (14) შეერთებულია არდუინოს მე-11 კონტაქტთან, ხოლო მეორე მიკროსქემის მონაცემების შესასვლელი SI (14) შეერთებულია პირველი მიკროსქემის გადატანის კონტაქტთან QH აპოსტროფით (9). მიკროსქემაზე არის კიდევ სხვა კონტაქტები (10 და 13) რომლებიც არ გამოიყენება მართვაში და ამიტომ მათზედ მიყვანილნი არიან +5 ვ და GND შესაბამისად. როგორც ვხედავთ, თვითოეულ მიკროსქემის გამოსასვლელებზე მიერთებულნი არის რვა-რვა შუქდიოდი ანუ არდუინომ უნდა მართოს 16 შუქდიოდი, მხოლოდ თავისი სამი კონტაქტის გამოყენებით.

მოვიყვანოთ ახლა შუქდიოდების მართვის ალგორითმი. არდუინოს ჩართვის მომენტიდან ინთება შუქდიოდები ცალცალკე და მორიგეობით და ეს ანთებები გადაინაცვლებს პირველი შუქდიოდიდან მე-16-მდე, შემდეგ პაუზა რამდენიმე ხნის განმავლობაში და შემდეგ პროცესი მეორდება, აინთება ისევ პირველი შუქდიოდი და გადაინაცვლებს ანთებები მე-16 შუქდიოდისაკენ და ასე გაგრძელდება 5 ციკლი, ამით ალგორითმის პირველი ეტაპი დამთავრებული იქნება. ამის შემდეგ იწყება მეორე ეტაპი, რომელიც გულისხმობს, რომ ჯერ აინთება პირველი შუქდიოდი, შემდეგ მეორე შუქდიოდი, შემდეგ პირველი ჩაქრება მაგრამ სამაგიეროდ აინთება მესამე შუქდიოდი, შემდეგ მეორე ჩაქრება და სამაგიეროდ აინთება მე-4 შუქდიოდი და ასე შემდეგ გადაინაცვლება ხდება ორ-ორი ანთებული შუქდიოდის პირველიდან ბოლოსაკენ, როცა ბოლო შუქდიოდიც ჩაქრება, რაღაც პაუზის შემდეგ იგივე ციკლი განმეორდება და ა.შ.შ. აღწერილი ციკლი განმეორდება ასევე ხუთჯერ. ამის შემდეგ სამ-სამი ანთებული შუქდიოდი გადაინაცვლებს თავიდან ბოლოსკენ და ესეც ხუთჯერ, შემდეგ ოთხ-ოთხი ანთებული შუქდიოდი გადაინაცვლებს თავიდან ბოლოსკენ და ესეც ხუთჯერ, შემდეგ ხუთ-ხუთი ანთებული შუქდიოდი გადაინაცვლებს თავიდან ბოლოსკენ ხუთჯერ და ბოლოს ვექვსექვსი ანთებული შუქდიოდი ასევე ხუთჯერ და ამის შემდეგ მეორდება პირველი ეტაპი, ე.ი ისევ თითო ანთებული შუქდიოდი გადაინაცვლებს თავიდან ბოლოსკენ ხუთჯერ და ა.შ.შ. დაუსრულებლად.

აღნიშნული ალგორითმის რეალიზაციის პროგრამა განლაგებულია FLProg-ის ორ დაფაზე-Plata: 1 და Plata: 2 (იხ. 58.3 და ფიგ. 58.4) შესაბამისად. მთელი პროგრამის მუშაობას სინქრონიზაციას უწევს პირველ დაფაზე განლაგებული სიმეტრიული მულტიბვიბრატორი B1 (ფიგ. 58.3), რომელიც გამოიმუშავებს იმპულსების თანმიმდევრობას იმპულსისა და პაუზის

ხანგრძლიობებით 60 მილიწამი თვითოეული. ეს ნიშნავს, რომ ყოველი 120 მილიწამის განმავლობაში მოხდება ორი 74HC595 მიკროსქემით შედგენილი 16 თანრიგიან წანაცვლების რეგისტრში შეტანილი ერთიანების გადაადგილება მარცხნიდან მარჯვნივ და ამ ერთიანების შესაბამისი შუქდიოდების ანთება. ამ გენერატორის გამოსასვლელი მიერთებულია B2 მთვლელის სინქრონიზაციის შესასვლელთან, რომელიც დაპროგრამებულია 22-მდე ციკლურ თვლაზე, ანუ წანაცვლების ეს რაოდენობა 21 დაკავშირებულია წანაცვლების რეგისტრის წანაცვლებების რაოდენობასთან. ეს ნიშნავს, იმას, რომ თუ ჩვენ დასაწყისში წანაცვლების რეგისტრში შევიტანთ ერთიანს, მაშინ გენერატორის 15 იმპულსის განმავლობაში ეს ერთიანი მიაღწევს წანაცვლების რეგისტრის მე-16 თანრიგს, მე-16 იმპულსზე ეს ერთიანი გამოიღვენება რეგისტრიდან, ხოლო მე-17...21-ე იმპულსებზე წანაცვლების რეგისტრში იტრიალებს ნოლები (ე.ი. წარმოიშობა პაუზა შუქდიოდების ანთებაში), ხოლო 22 იმპულსი დაყენებს B2 მთვლელს ნულოვან მდგომარეობაში, ამ მდგომარეობაში ისევ ჩაიწერება წანაცვლების რეგისტრში ერთიანი და ა.შ. პროცესი უნდა გაგრძელდეს დაუსრულებლად.

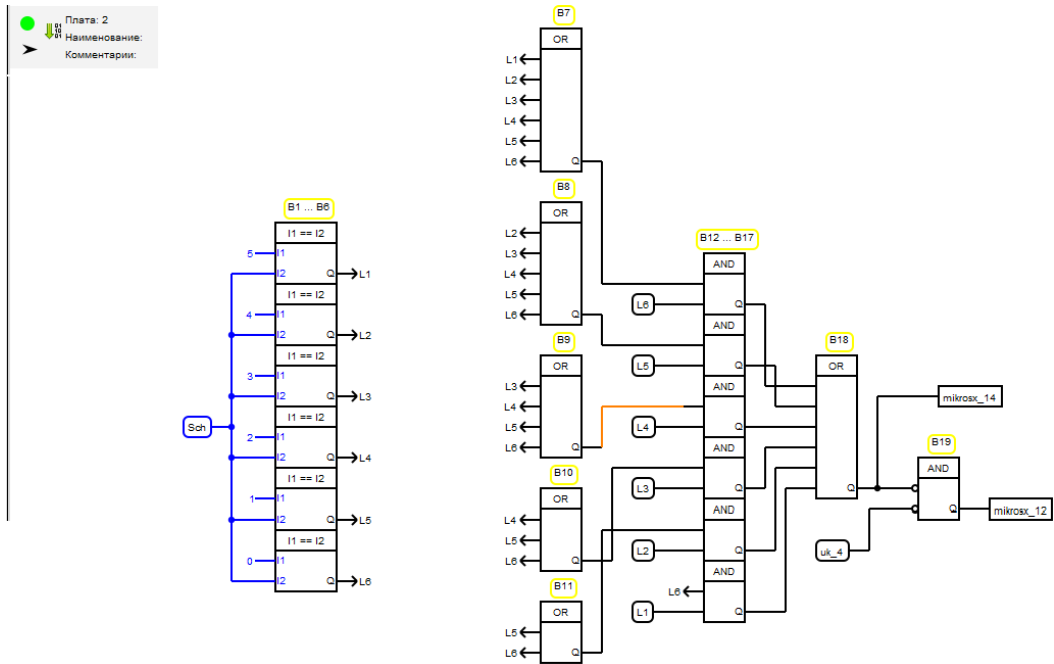
იმისათვის, რომ ჩვენ შეგვეძლოს წანაცვლების რეგისტრში ხან ერთი ერთიანის ჩაწერა, ხან ორის, ხან სამის, ხან ოთხის და ხან ხუთის და ეს უნდა მოხდეს B2 მთვლელის ხან ნოლოვან, ხან როგორც ნოლოვან ასევე ერთიანის მდგომარეობაში და ასე შემდეგ ჩვენ დაგვჭირდება ამ პოზიციების დაფიქსირება.



ფიგ. 58.3 FLProg-ის პირველ დაფაზე მოთავსებული მართვის პროგრამა

ამას ანხორციელებენ მე-2 დაფაზე მყოფი B1 ... B6 შედარების ბლოკები (ფიგ. 58.4) ტოლობაზე, რომელთა პირველი შესასვლელი შეერთებულია B2 მთვლელის ანალოგურ გამოსასვლელთან Sch მთელის ტიპის ცვლადის გამოყენებით. ბლოკების მეორე

შესასვლელზე ჩაწერილია კონსტანტები 0, 1, 2, 3, 4, 5, რაც ნიშნავს იმას, რომ როდესაც B2 მთვლელში ჩაწერილია ეს რიცხვები, მაშინ მათ გამოსასვლელზე L6, L5, L4, L3, L2, L1 გამომუშავდება ლოგიკური ერთიანები.



ფიგ. 58.4 FLProg-ის მეორე დაფაზე მოთავსებული მართვის პროგრამა

რადგან ჩვენ ზოგ შემთხვევაში გვჭირდება მხოლოდ ერთი ერთიანის ჩაწერა წანაცვლების რეგისტრში და ეს უნდა მოხდეს მთვლელის ნულოვან მდგომარეობაში. ზოგ შემთხვევაში ორი ერთიანის ჩაწერა და ეს უნდა მოხდეს მთვლელის ნოლოვან და ერთიანის მდგომარეობაში და ა.შ. საჭირო ხდება ამ მდგომარეობების გაერთიანება. ამას კი ასრულებენ B7 ... B12 ლოგიკური შეკრების ბლოკები. მაგალითად მთვლელის ხუთი მდგომარეობის 0...5 მდგომარეობის გაერთიანებას ემსახურება ბლოკი B7, რომელის შესასვლელზეც მიერთებულია L1 ... L6 გამოსასვლელები და მასაშადამე იგი აერთიანებს მთვლელის მდგომარეობებს 0-დან 5-მდე ჩათვლით, ანუ ამ ბლოკის გამოსასვლელზე ლოგიკური ერთიანი იქნება მთვლელის ყველა მდგომარეობის დროს დაწყებული ნოლიდან დამთავრებული ხუთამდე. ასევე მუშაობენ B6 ... B11, მაგრამ მთვლელის თითო-თითო მდგომარეობის გამოკლებებით. ამ ბლოკების გამოსასვლელები მიერთებულია B12 ... B17 ლოგიკური გამრავლების ბლოკების B12 ... B17 პირველ შესასვლელებთან (B7 ბლოკზე პირდაპირ L6-ია მიერთებული, რადგან ამ შემთხვევაში ჩვენ გვჭირდება მხოლოდ ერთი ერთიანის ჩაწერა მთვლელის მხოლოდ ნულოვანი მდგომარეობის დროს), რომლის მეორე შესასვლელების რაობის დასადგენად დაუზრუნდეთ ისევ პირველ დაფას (ფიგ. 58.3).

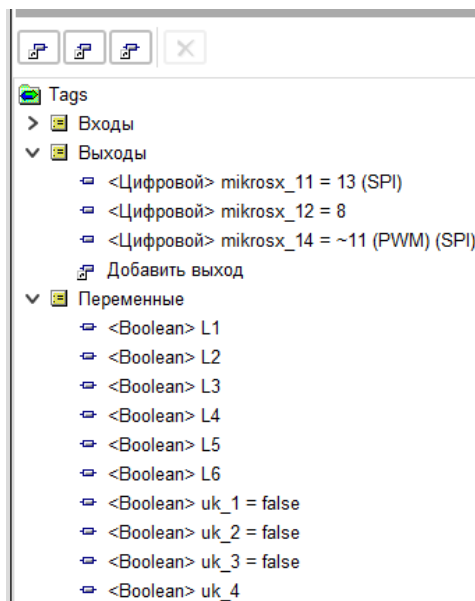
პირველ რიგში აღვნიშნოთ, რომ B1 გენერატორის გამოსასვლელი გადის არდუინოს მე-13 კონტაქტზე, რომელსაც მინიჭებული აქვს სახელი mikrosx_11, რაც ნიშნავს იმას, რომ არდუინოს ეს კონტაქტი მიერთებულია გაფართოების მიკროსქემების სინქრონიზაციის შესასვლელებზე მე-11 კონტაქტზე. ამის შემდეგ, B2 მთვლელის გამოსასვლელი მიდის B3 მთვლელის შესასვლელთან, რომელიც ისეა დაპროგრამებული, რომ ითვლის ხუთამდე ციკლურად. ხუთამდე იმიტომ, რომ ხუთჯერ უნდა მოხდეს თვითოეული ეტაპის განმეორება, მაგ. თითო ანთებული შუქდიოდის გადანაცვლება, ორორი ანთებული შუქდიოდის გადანაცვლება და ა.შ. ამის შემდეგ B3 მთვლელის გამოსასვლელი შეერთებულია B4 მთვლელის შესასვლელთან, რომელიც ისეა დაპროგრამებული, რომ ითვლის ციკლურად ექვსამდე. ექვსამდე იმიტომ, რომ ჩვენ გვჭირდება ჯერ ერთი ერთიანის გადაადგილება ხუთჯერ, მერე ორი ერთიანის - ასევე ხუთჯერ და ბოლოს ექვსი ერთიანის ასევე ხუთჯერ. მაშასადამე გამოდის, რომ B4 მთვლელში ჩაწერილი რიცხვის რაოდენობა განსაზღვრავს წანაცვლების რეგისტრში ჩასაწერი ერთიანების რაოდენობას.

იმისათვის, რომ კონკრეტულად გავიგოთ თუ რამდენი ერთიანის ჩაწერა უნდა მოვახდინოთ, საჭიროა გამოვყოთ მთვლელის ეს მდგომარეობები. ამას ახორციელებს B5 ... B10 კომპარატორის ბლოკები, რომელთა პირველ ფეხებზე ჩაწერილია რიცხვები ნოლიდან ხუთამდე, ხოლო მეორე ფეხები შეერთებულია B4 მთვლელის ანალოგურ გამოსასვლელთან. მაშასადამე B5 ... B10 კომპარატორის ბლოკების გამოსასვლელებზე (ბულის ცვლადები L6 ... L1) ხდება B4 მთვლელის მდგომარეობის დაფიქსირება და ლოგიკური ერთიანი ერთერთ L6 ... L1 გამოდსასვლელზე მიუთითებს იმას, თუ რამდენი ერთიანია ჩასაწერი წანაცვლების რეგისტრში. ეს გამოსასვლრელები გადადიან FRLProg-ის მეორე დაფაში და შეერთებულნი არიან ჩვენთვის უკვე ნაცნობ B12 ... B17 ლოგიკური გამრავლების ბლოკების მეორე შესასვლელებთან. მაშასადამე, ამ ბლოკებზე ერთის მხრივ მიეწოდება სიგნალები თუ რამდენი ერთიანი უნდა ჩაიწეროს რეგისტრში, ხოლო მეორე მხარეს მიეწოდება სიგნალები თუ რომელ პოზიციებზე უნდა ჩაიწეროს ეს ერთიანები. ამ სიგნალების თანმთხვევის შედეგად ბლოკების B12 ... B17 გამოსასვლელებზე წარმოიქმნება ლოგიკური ერთიანები, რომლებიც გაერთიანებულია ლოგიკური შეკრების ბლოკით B18. ამ უკანასკნელის გამოსასვლელი გადის არდუინოს მე-11 კონტაქტზე სახელწოდებით mikrosx_14, საიდანაც უერთდება პირველი მიკროსქემის მე-14 შესასვლელს. ეს შესასვლელი არის მონაცემების შესასვლელი და ერთიანების რაოდენობა, რომლებიც უნდა ჩაიწეროს რეგისტრში განისაზღვრება იმ სიგნალით რომელიც მიეწოდება ამ შესასვლელზე. თუ მაგ. რეგისტრში უნდა ჩაიწეროს სამი ერთიანი, მაშინ ამ მონაცემების შესასვლელზე სამი სინქრონიზაციის განმავლობაში ანუ იმ

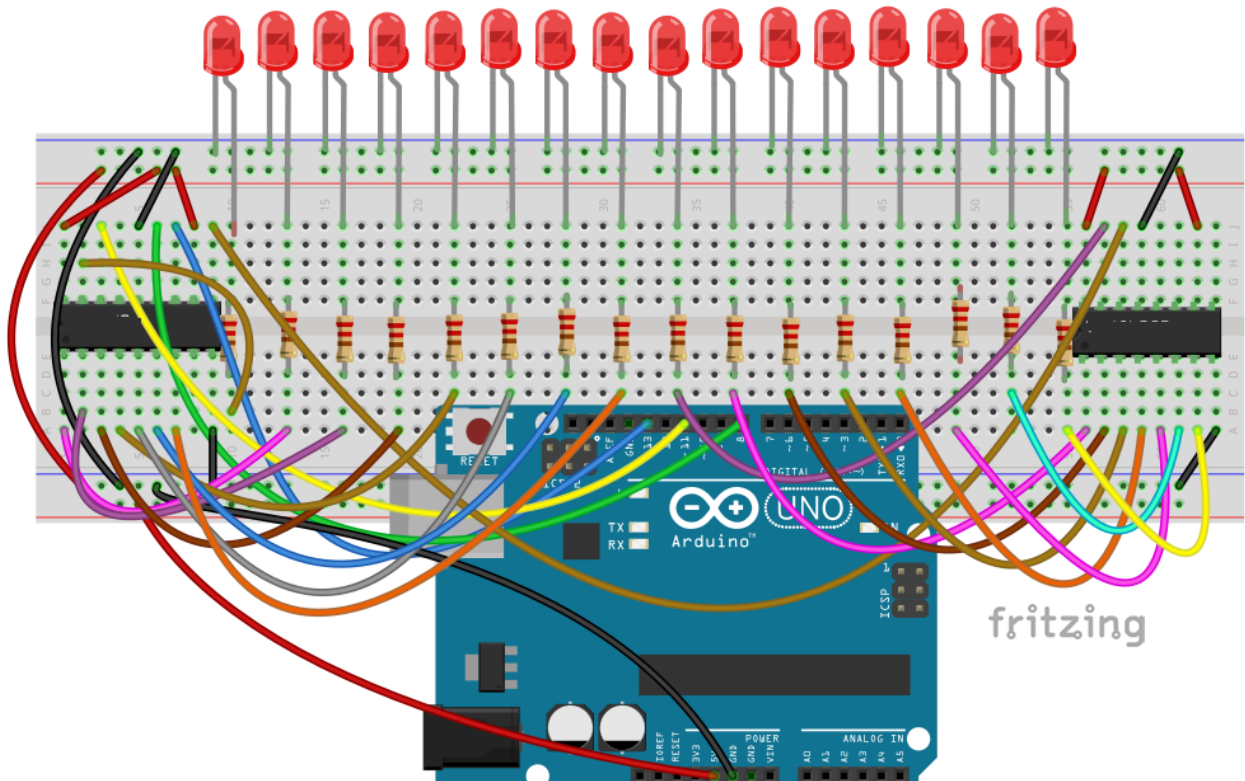
სინქრონიზაციის რომელიც მიეწოდება რეგისტრს მე-13 შესასვლელზე B1 გენერატორიდან, უნდა იდგეს ლოგიკური ერთიანი, დაწყებული B2 მთვლელის ნოლოვანი პოზიციიდან.

როდესაც საჭირო რაოდენობის ერთიანების ჩაწერა რეგისტრში დამთავრდება, მაშინ B18 ბლოკის გამოსასვლელზე დაჯდება ლოგიკური ნოლი და ამასთანავე, როდესაც იმპულსების გენერატორის გამოსასვლელზე იმპულსი დამთავრდება და დაიწყება პაუზა, მაშინ ამუშავდება B19 ლოგიკური გამრავლების ბლოკი, მის გამოსასვლელზე დაფიქსირდება ლოგიკური ერთიანი, რომელიც არდუინოს მე-8 კონტაქტის გავლით მიეწოდება მიკროსქემების ფიქსატორების შესასვლელებს მე-12 ფეხებზე და დააფიქსირებს წანაცვლების რეგისტრში ჩაწერილი მონაცემების მნიშვნელობას (ანუ ერთიანების რაოდენობას). აქ არ უნდა დაგვავიწყდეს, რომ მეორე მიკროსქემის მონაცემთა შესასვლელი, ანუ მისი მე-14 ფეხი უნდა იქნას შეერთებული პირველი მიკროსქემის გადატანის ფეხთან, ანუ მე-9 ფეხთან. შუქდიოდები კი შემზღულდავი 220 ომიანი წინააღმდეგობების გავლით უნდა იქნას შეერთებული შუქდიოდის ანოდებთან, რომელთა კათოდებიც უნდა იყოს შეერთებული არდუინოს GND-სთან.

პროექტის ტეგების ზონა ასახულია ფიგ. 58.5-ზე, ხოლო Fritzing სქემა-ფიგ. 58.6-ზე.



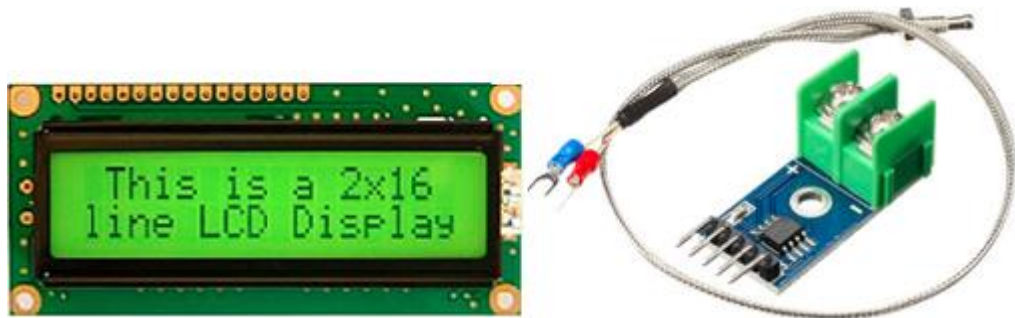
ფიგ. 58.5. ტეგების ზონა



ფიგ. 58. 6 პროექტის Fritzing სქემა

პროექტი_59

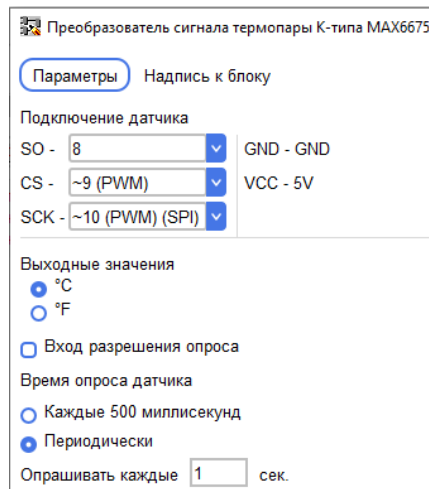
K-ტიპის თერმოწყვილის სიგნალის გარდაქმნელი MAX6675



FLProg-ის ბლოკების ბიბლიოთეკის ტემპერატურის გადამწოდების საქალაქდემი-
 Датчики температуры დევს ერთი ისეთი გადამწოდი რომელსაც ეწოდება K-ტიპის
 თერმოწყვილის სიგნალის გარდაქმნელი MAX6675-Преобразователь сигнала термопары K-
 типа MAX6675. ამ ბლოკის დანიშნულებაა K-ტიპის თერმოწყვილის სიგნალის გარდაქმნელი
 MAX6675-სგან გასაზომი ტემპერატურის მნიშვნელობების მიღება.

გადამწოდის სამუშაო კვების დიაპაზონია 3 ... 5 ვოლტი, გადამწოდს აქვს თავისი
 საკუთარი შიგა სქემა ცივი მინრჩილვის კომპენსაციისათვის და არ ჭირდება დამატებითი
 დამუშავება. ტემპერატურის გადამწოდი წარმოადგენს K-ტიპის თერმოწყვილს და ზომავს

ტემპერატურას 0-დან 800 გრადუს ცელსიუსის ფარგლებში. თერმოწყვილის არდუინოსთან შეერთების სქემა ასახულის ფიგურა 59.1-ზე.

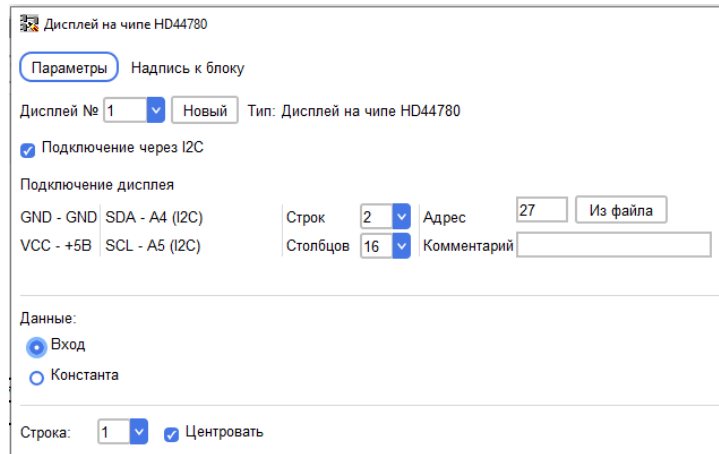


ფიგ. 59.1 სიგნალის K-ტიპის თერმოწყვილის გარდამქმნელის რედაქტორის ბლოკი

ბლოკის აწყობა ხდება ბლოკის რედაქტორის შევსებისად. აქ უნდა გვახსოვდეს, რომ აუცილებლად უნდა მოხდეს არდუინოს პინების შერჩევა, რომლებზეც უნდა შეერთდეს გადამწოდის სასიგნალო გამომყვანები: SO, CS და SCK, აგრეთვე GND-ს და VCC-ს გამოსასვლელები. ბლოკის აწყობებში შეიძლება მივუთითოთ აგრეთვე თუ რომელ ფორმატში გვინდა ტემპერატურის ანათვალეები ცელსიუსში თუ ფარენგეიტში.

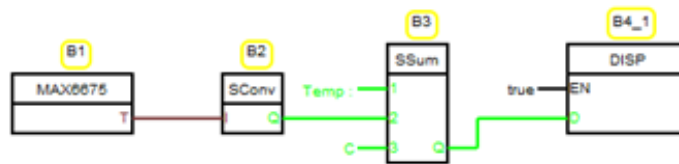
თუ ჩვენ ჩავრთავთ ასევე ოპციას გამოკითხვის ნებისდართვის შესასვლელი-Вход разрешения опроса, გარდამქმნელთან მიკითხვა შესაძლებელი იქნება მხოლოდ ლოგიკური ერთიანის არსებობის შემთხვევაში EN შესასვლელზე. გარდამქმნელის გამოკითხვათა რაოდენობა შეზღუდულია დროში და დასაშვებია არაუმეტეს ორი გამოკითხვისა ერთ წამში.

გარდამქმნელიდან მიღებული გამოკითხვის შედეგის ასახვა შეიძლება მოხდეს სხვადასხვა სახის დისპლეიზე: შვიდსეგმენტა ოთხთანრიგა დისპლეი, Nextion HMI პანელი, დისპლეი HD44780-ჩიპზე. ჩვენ გამოვიყენებთ ამ უკანასკნელს. მისი პარამეტრიზაციის ფანჯარა ასახულია ფიგ. 59.2-ზე.



ფიგ. 59.2 დისპლეის ჩიპზე HD44780 პარამეტრიზაციის ფანჯარა

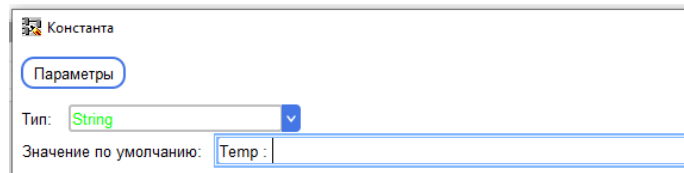
შევუერთოთ ახლა K-ტიპის თერმოწყვილის სიგნალის გარდამქმნელი MAX6675 დისპლეის შესასვლელს. მაგრამ ამისათვის დაგჭირდება დამატებითი ბლოკების შემოტანა.



ფიგ. 59.3 K-ტიპის თერმოწყვილის გარდამქმნელის მართვის პროგრამა

ფიგ. 59.3-ზე წარმოდგენილია სიგნალის K-ტიპის თერმოწყვილის გარდამქმნელის მართვის პროგრამა სადაც B1 ბლოკიდან, ანუ K-ტიპის MAX6675 თერმოწყვილიდან გამომავალი სიგნალი მანამ, სანამ დისპლეის შესასვლელზე შევიდოდეს განიცდის სტანდარტულ დამუშავებას. იგი ჯერ გაივლის სტრიქონების კონვერტაციის ბლოკს B2, სტრიქონების შეკრების ბლოკს B3 და შემდეგ შედის დისპლეის ბლოკის ანალოგურ შესასვლელზე B4.

სტრიქონების შეკრებისათვის ჩვენ აღებული გვაქვს სამშესასვლელიანი შეკრების ბლოკი, რომლის პირველ შესასვლელზე ჩაწერილი გვაქვს (იხ. ფიგ. 59.4) კონსტანტა Temp : (და ბოლოში პრაბელი), მეორე ფეხზე მიერთებული გვაქვს კონვერტაციის ბლოკის გამოსასვლელი, ანუ გასაზომი ტემპერატურის სიდიდე, ხოლო მესამე ფეხზეც ასევე ჩაწერილი გვაქვს კონსტანტა-პრაბელი და C, რომელიც მიუთითებს გასაზომი ტემპერატურის განზომილებას ცელსიუსის შკალაში (ფიგ. 59.5)

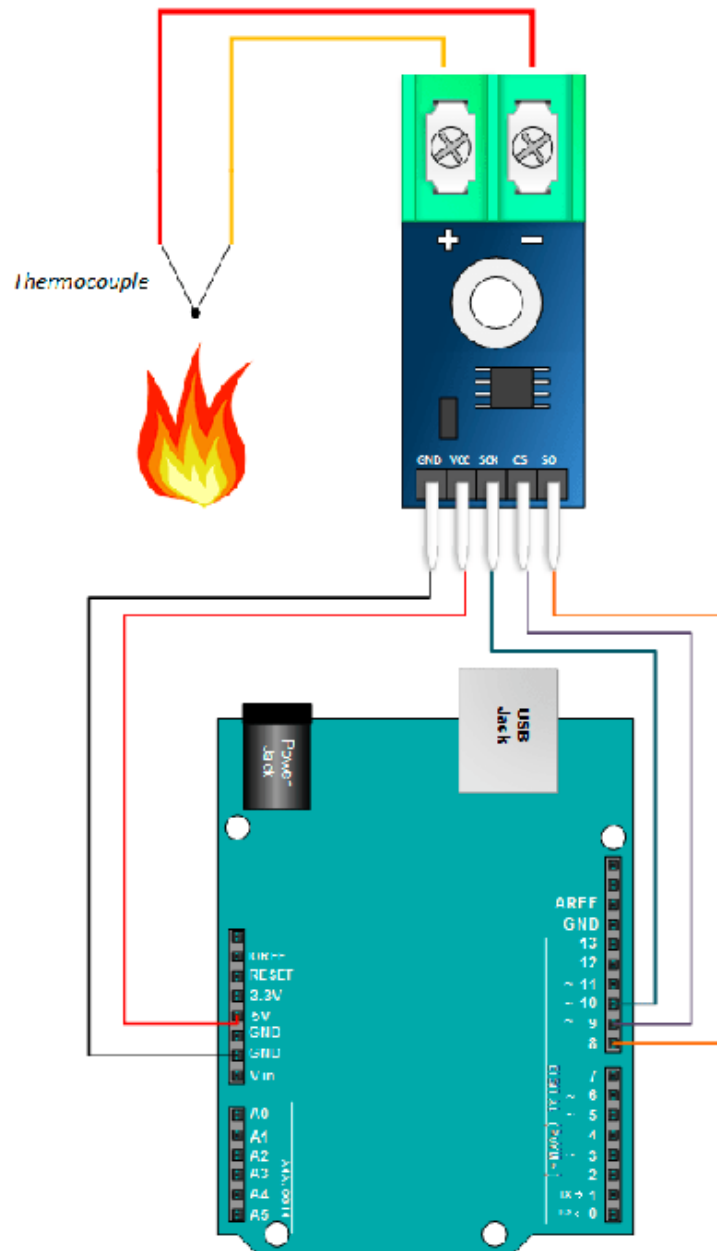


ფიგ. 59.4 კონსტანტის ჩაწერის ფანჯარა სტრიქონების შეკრების ბლოკის პირველ ფეხზე



ფიგ. 59.5 კონსტანტის ჩაწერის ფანჯარა სტრიქონების შეკრების ბლოკის მესამე ფეხზე

K-ტიპის MAX6675 თერმოწყვილის გარდამქმნელის გამოყვანების შეერთების Fritzing სქემა არდუინოს კონტაქტებთან ნაჩვენებია ფიგ. 59.6-ზე.



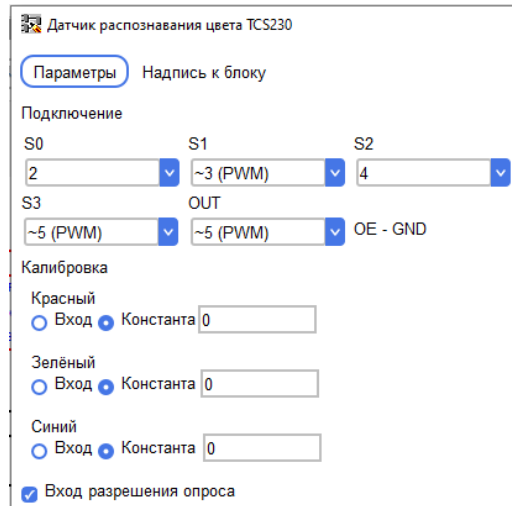
ფიგ. 59.6 არდუინოს შერთების სქემა K-ტიპის თერმოწყვილის გარდამქმნელთან MAX6675

პროექტი_60 ფერების გამოცნობის სენსორი TCS230



TCS230 მიკროსქემა ახდენს ფერის სპექტრის ინტენსივობის გარდაქმნას სიგნალად. ფერის განსაზღვრისათვის იგი ზომავს სამ სპექტრს, წითელს, ლურჯს და მწვანეს. მიკროსქემა

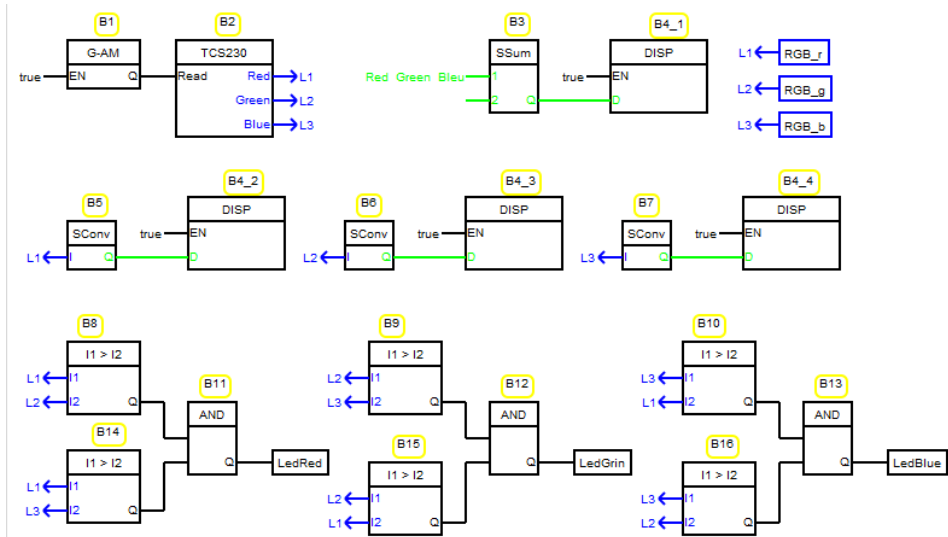
შედგება 64 ფოტოდiodისაგან რომლებიც განსაზღვრავენ ფილტრის რეჟიმის არჩევას: ლურჯს, მწვანეს, წითელს და უფილტროს. მიკროსქემის სამუშაო ძაბვაა: 2,7 ... 5,5 ვ. FLProg-ში ამ მიკროსქემასთან მუშაობისათვის გადამწოდების საქალაქდემი-Датчики და შემდეგ ფერის გადამწოდებში-Световые датчики ჩადებულია ბლოკი ფერის გამოცნობის გადამწოდი - Датчик распознавания цвета TSC230, რომლის დანიშნულებაა TSC230 მიკროსქემასთან მუშაობა. არდუინოსთან მიერთებისათვის ეს ბლოკი იყენებს ოთხ გამომყვანს: S0, S1, S2, S3, აგრეთვე გამოსასვლელ გამომყვანს OUT და პლიუს ამას ორ კონტაქტს კვებისათვის (ფიგ. 60.1).



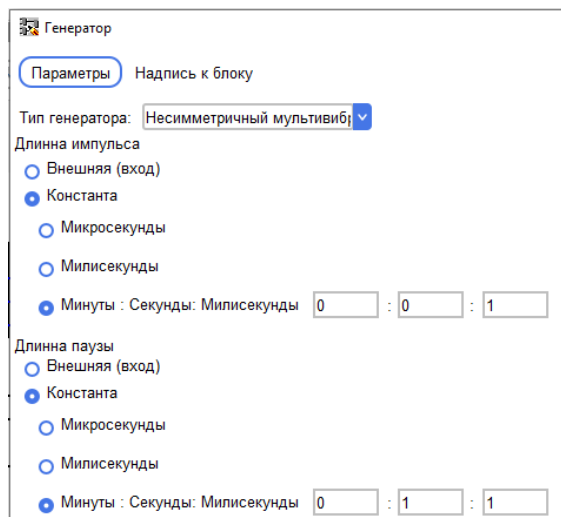
ფიგ. 60.1 ფერის გამოცნობის გადამწოდის TSC230 პარამეტრიზაციის ფანჯარა

ამავე ფანჯარაში სიგნალების დაკალიბრებისათვის-Калибровка გამოიყენება კონსტანტები, რომლებიც ჩვენს შემთხვევაში აღებულია ნოლის ტოლები. ამავე ფანჯრიდან ჩანს, რომ ჩართულია გამოკითხვის ნებისდართვის შესასვლელი, რაც იმას ნიშნავს, რომ გადამწოდის გამოკითხვა შესაძლებელი იქნება მაშინ, როდესაც Read შესასვლელზე დაჯდება ლოგიკური ერთიანი იმპულსში. ფიგ. 60.2-ზე ასახულია ფერების გამოცნობის გადამწოდის TSC230-ის მართვის პროგრამა FLProg გარემოში.

მართვის პროგრამაში ძირითად მასინქრონიზებელ ელემენტად გამოყენებულია არასიმეტრიული მულტივიბრატორი (გენერატორი) B1, იმპულსისა და პაუზის ხანგრძლიობები არასიმეტრიულ რეჟიმში, სადაც იმპულსის ხანგრძლიობა 1 მილიწამის ტოლია, ხოლო პაუზის ხანგრძლიობა 1,1 წამის (ფიგ. 60.3).



ფიგ. 60.2 ფერების გამოცნობის გადამწოდის TSC230-ის მართვის პროგრამა

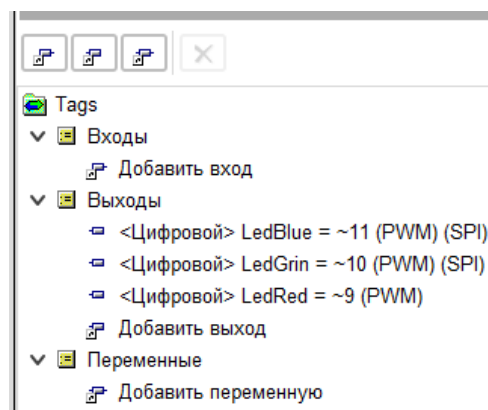


ფიგ. 60.3 არასიმეტრიული მულტივიბრატორის პარამეტრიზაციის ფანჯარა

მულტივიბრატორის იმპულსები მიეწოდება ფერების გამოცნობის გადამწოდს TSC230 Read შესასვლელზე, რაც იმას ნიშნავს, რომ ყოველ 1.1 წამის განმავლობაში ერთი მილიწამის ხანგრძლიობით ხდება ფერების ინტენსიობების ამოკითხვა TSC230 გადამწოდიდან. ამოკითხვა წითელი, ლურჯი და მწვანე ფერების შესაბამისი ინტენსიობაში: Red, Green, Blue, რომლებიც წარმოადგენენ მთელ რიცხვებს. ამის შემდეგ ხდება ჯერ ამ სიდიდეების (ინტენსიობების) ასახვა თექვსმეტანრიგა ორსტრიქონიან დისპლეიზე (ჩიპი HD44780) მეორე სტრიქონის პირველ, მე-7 და მე-13 სტრიქონებიდან (ფიგ. 60.2-ის B4_2, B4_3 და B4_4). ამ ბლოკის პირველი სტრიქონის ცენტრში უბრალოდ აისახება ინფორმაცია კონსტანტა: Red, Green, Blue (პრაბელი), რომელიც ჩაწერილი გვაქვს მისი პარამეტრიზაციის ფანჯარაში.

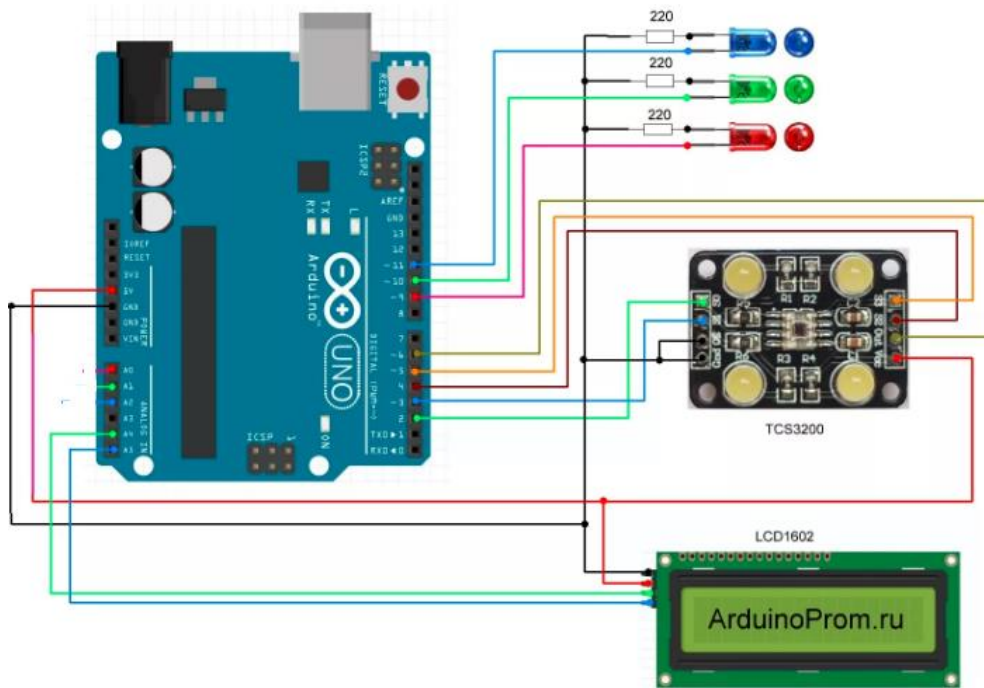
გარდა ამისა, გადამწოდებიდან ამოკითხული ინტენსიობების მაჩვენებლებზე შემდეგში ხდება მათემატიკური დამუშავება, იმის დასადგენად, თუ რომელი ფერის ინტენსიობა სჭარბობს სპექტრში. ამისთვის პროგრამაში შემოტანილია კომპარატორის ბლოკები (B8, B9, B10 და B14, B15, B16 და აგრეთვე ლოგიკური გამრავლების ბლოკები B12, B13, B14). როგორც ზემოთ გვქონდა აღნიშნული, გადამწოდის მიერ გამომუშავებული სიგნალები მუდმივი რიცხვებია ანუ ამ პროგრამაში ანალოგური სიდიდეებია და ეს სიდიდეები Red, Green, Blue აღნიშნულია L1, L2, L3 ხაზები.

ახლა ვნახოთ თუ რა მათემატიკურ-ლოგიკურ ოპერაციას ასრულებენ ბლოკები: B8, B14 და B11. თუ $L1 > L2$, $L1 > L3$ ანუ თუ წითელი ფერის ინტენსიობა მეტია ლურჯი ფერის ინტენსიობაზე და თუ წითელი ფერის ინტენსიობა მეტია მწვანე ფერის ინტენსიობაზე, ანუ როდესაც სრულდება ეს ორივე პირობა მაშინ უნდა გამომუშავდეს სიგნალი Ledred ანთოს წითელი შუქდიოდი, რომელიც მიუთიებს გამოცნობილ ფერზე. ასევე თუ მწვანე ფერის ინტენსიობა მეტია როგორც წითელი ფერის ინტენსიობაზე, მეტია აგრეთვე ლურჯი ფერის ინტენსიობაზე, მაშინ B9, B15 და B12 ბლოკების დახმარებით უნდა გამომუშავდეს სიგნალი LedGreen ანთოს მწვანე ფერის შუქდიოდი. ანალოგიურად ზემოთაღწერილისა, თუ ლურჯი ფერის ინტენსიობა მეტია წითელი ფერის ინტენსიობაზეც და მწვანე ფერის ინტენსიობაზეც, მაშინ B10, B16 და B13 ბლოკების დახმარებით უნდა გამომუშავდეს სიგნალი LedBlue და ანთოს ლურჯი შუქდიოდი.



ფიგ. 60. 4 ტეგების ზონა

ფერების გამოცნობის გადამწოდის TSC230-ის მართვის პროგრამისთვის ფორმირებული ტეგების ზონა ნაჩვენებია ფიგ. 60.4-ზე. ფერების გამოცნობის გადამწოდის TSC230 მიერთების სქემა არდუინოსთან წარმოდგენილია ფიგ. 60.5-ზე. მაგრამ ის შეერთებები, რომლებიც არ არიან ასახული ტეგების ზონაში განისაზღვრება თვითონ TSC230-გადამწოდის პარამეტრიზაციის ფანჯარაში (ფიგ. 60.1).



ფიგ. 60.5 ფერების გამოცნობის გადამწოდის TSC230 მიერთების სქემა არდუინოსთან

ლიტერატურა;

1. ჯ. გრიგალაშვილი. Arduino-ს ვიზუალური დაპროგრამება FLProg გარემოში // სახელმძღვანელო // სახელმძღვანელო, ბიბლიოთეკა სტუ, CD – 3869 // 2017 წ. 289 გვ.
2. <https://flprog.ru>
3. <http://arduinoprom.ru>
4. Руководство по освоению Arduino // SOLARBOTICS /Зарегистрирован 01.09.2015
http://robot-kit.ru/manual/Arduino_Sketch_Robot-kit.ru.pdf
5. Arduino Basic Connection – The Book // Зарегистрирован 01.09.2015 г.
<https://dl.dropboxusercontent.com/u/32245429/arduinoBasicConnections.pdf>
6. ARDUINO BASIC CONNECTIONS - THE BOOK on Vimeo
<https://box.glgprograms.it/CorsoArduino2016/didattica/Arduino-BC.pdf>

არდუინოს ძირითადი შეერთებები გარე ელექტრონულ კომპონენტებთან

THE UNOFFICIAL ARDUINO basic connections

ლილაკის 12 ვოლტთან დაჭერით
Should you need to connect Arduino inputs to a DC voltage higher than 5V

შესასვლელი ოპტოწყვილით
Use when galvanic separation between external circuit and arduino circuit is required

ლილაკის მიწასთან დაჭერით

გარეთა წვეტიანობის

ლილაკის 5 ვოლტთან დაჭერით

თაშიერი ან პოტენციომეტრი

ფოტორეზისტორი

When using this circuit for 24 volts input increase R to 2200Ω (C101)

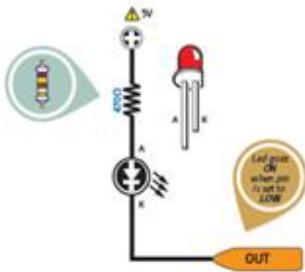
MAX +12V

V_{in} LOW

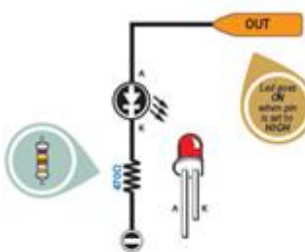
4N25

www.profblox.com
03 APR 2013
ver 1 rev 0

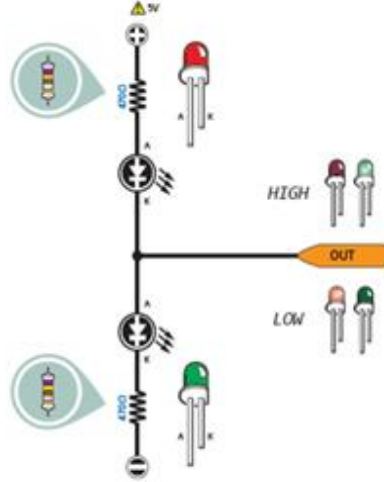
შუქდიოდის მიერთება



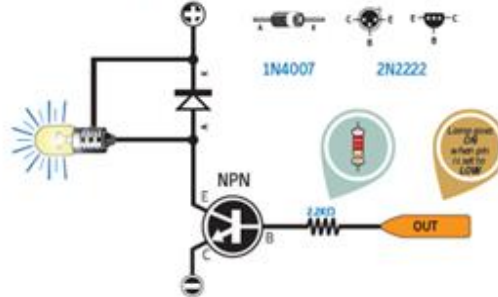
შუქდიოდის მიერთება



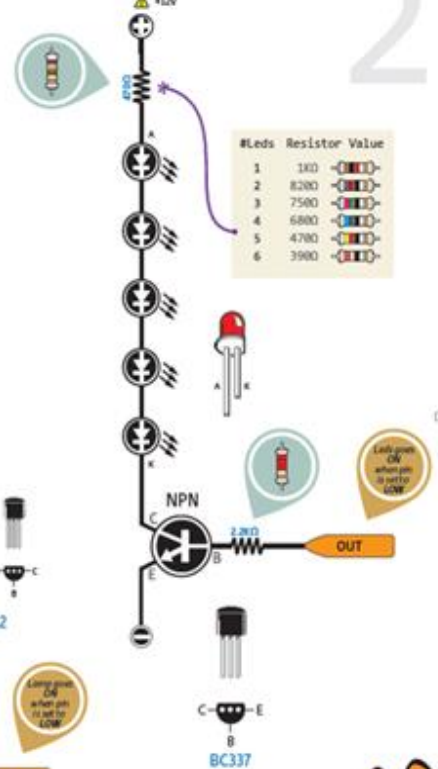
ორი შუქდიოდი



ნათურის მიერთება (დაბალვოლტაჟი)



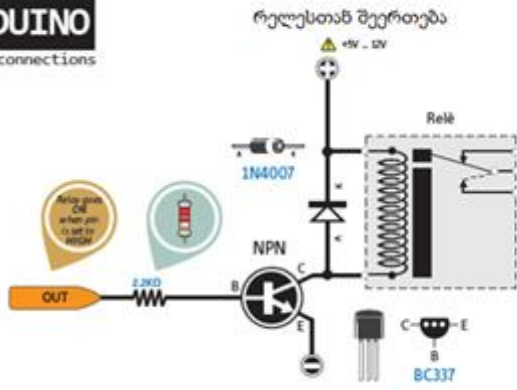
შუქდიოდების კლასტერი



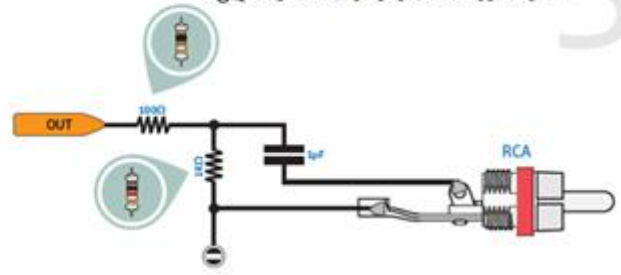
#Leds	Resistor Value
1	100
2	8200
3	7500
4	6800
5	4700
6	3900



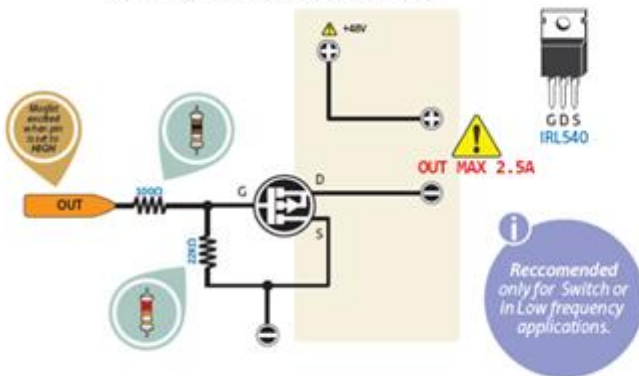
რელესთან შეერთება



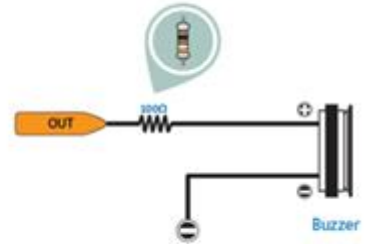
აუდიოგამაძლიერებელთან შეერთება



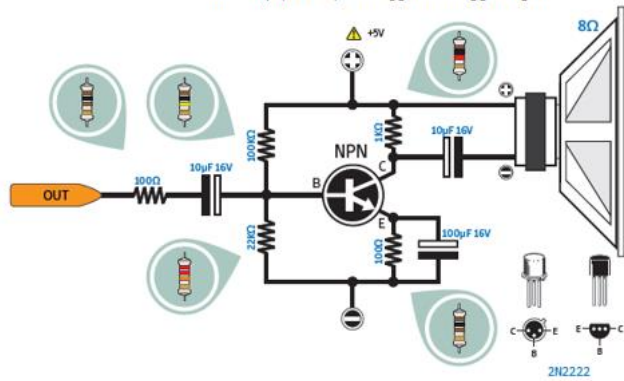
ველიან ტრანზისტორთან შეერთება



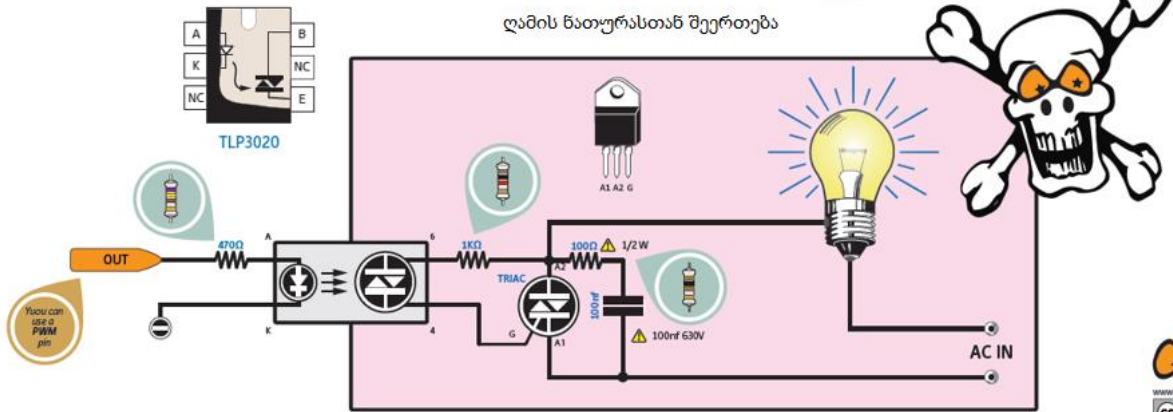
ბუზერთან შეერთება



ხმამალამოლაპარაკესთან შეერთება

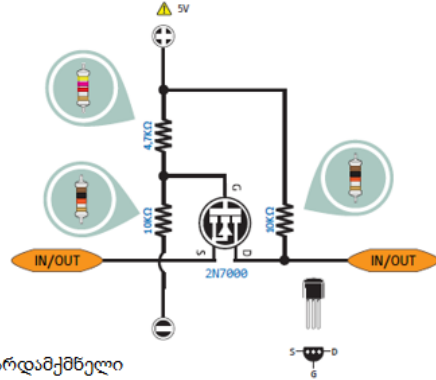
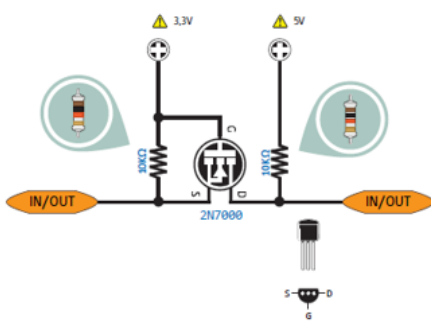


ღამის ნათურასთან შეერთება



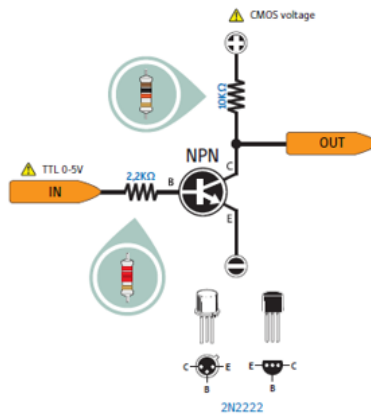
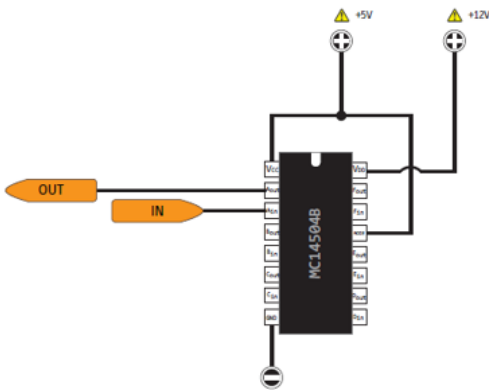
ორდონიანი ძაბვის გარდამქმნელი 3,3 ვოლტიდან 5 ვოლტად

ორდონიანი ძაბვის გარდამქმნელი 3,3 ვოლტიდან 5 ვოლტად ძაბვის გამოყოფის გამოყენებით



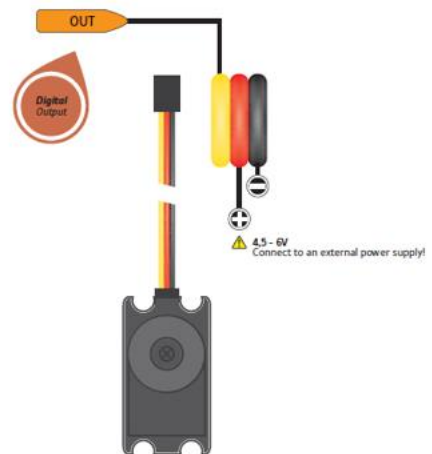
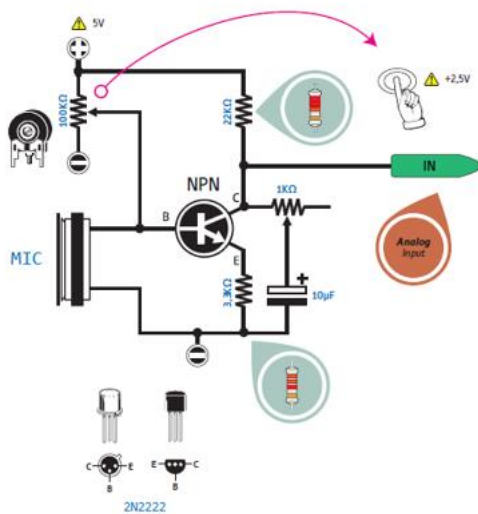
TTL/CMOS გარდამქმნელი (6 შესასვლელ/გამოსასვლელი)

TTL/CMOS გარდამქმნელი



მიკროფონის შეერთება

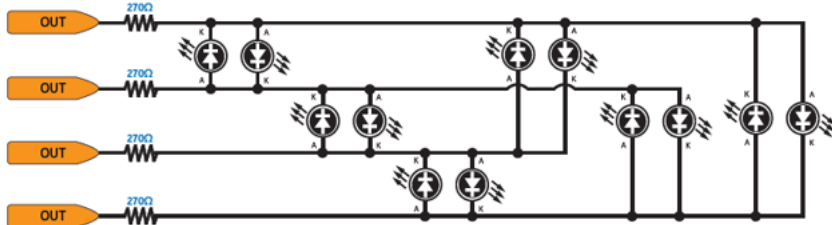
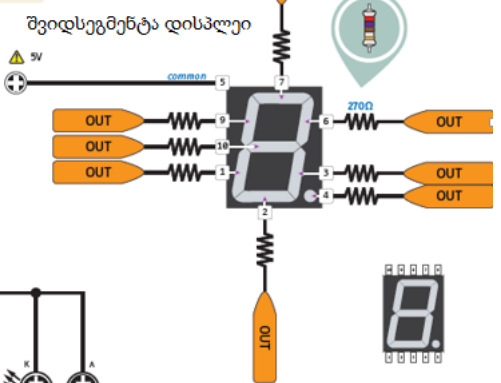
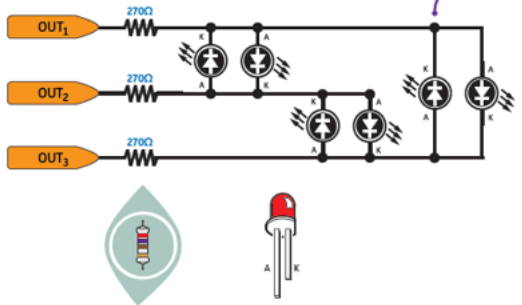
სერვომოტორის შეერთება



Charlieplexing

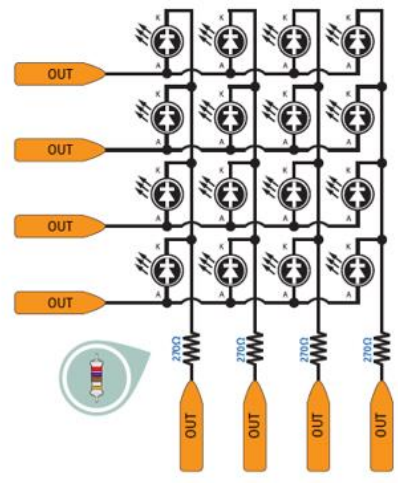
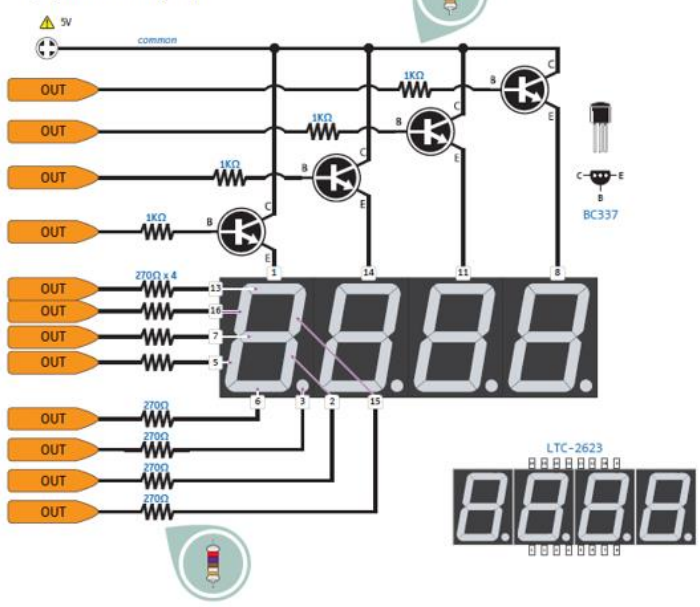
Out ₁	Out ₂	Out ₃
L	L	L
L	H	i
H	L	i
i	L	H
i	H	L
L	i	H
H	i	L

i=High, L=Low, i=Input

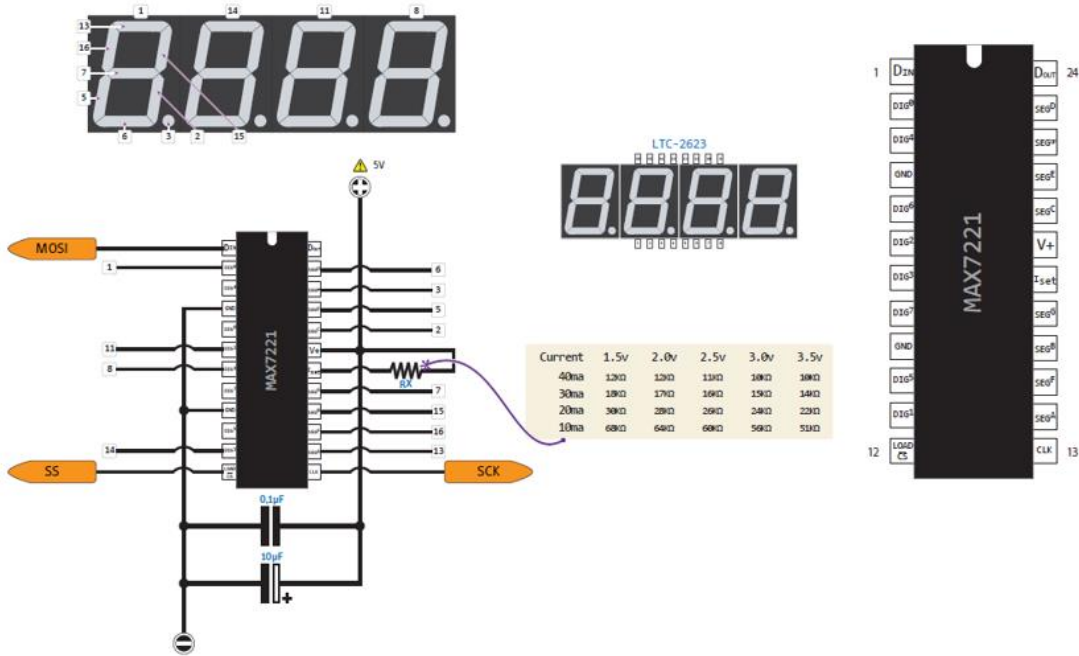


LED (შუქდიოდების) მატრიცა

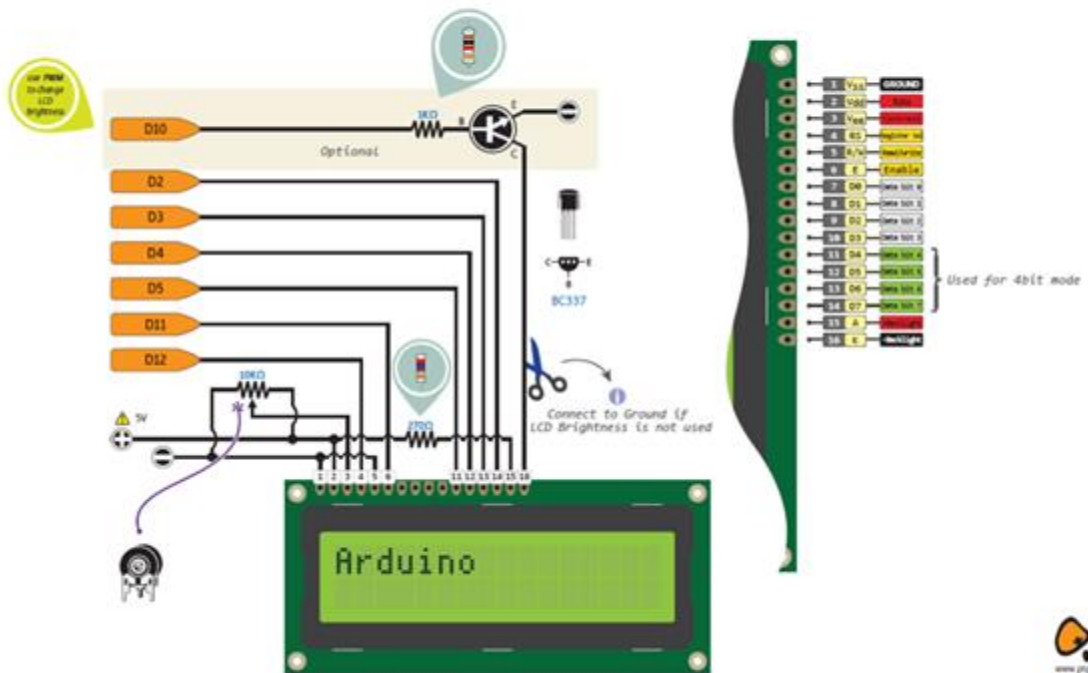
7 სეგმენტა მულტიპლექსური დისპლეი (საერთო ანოდით)



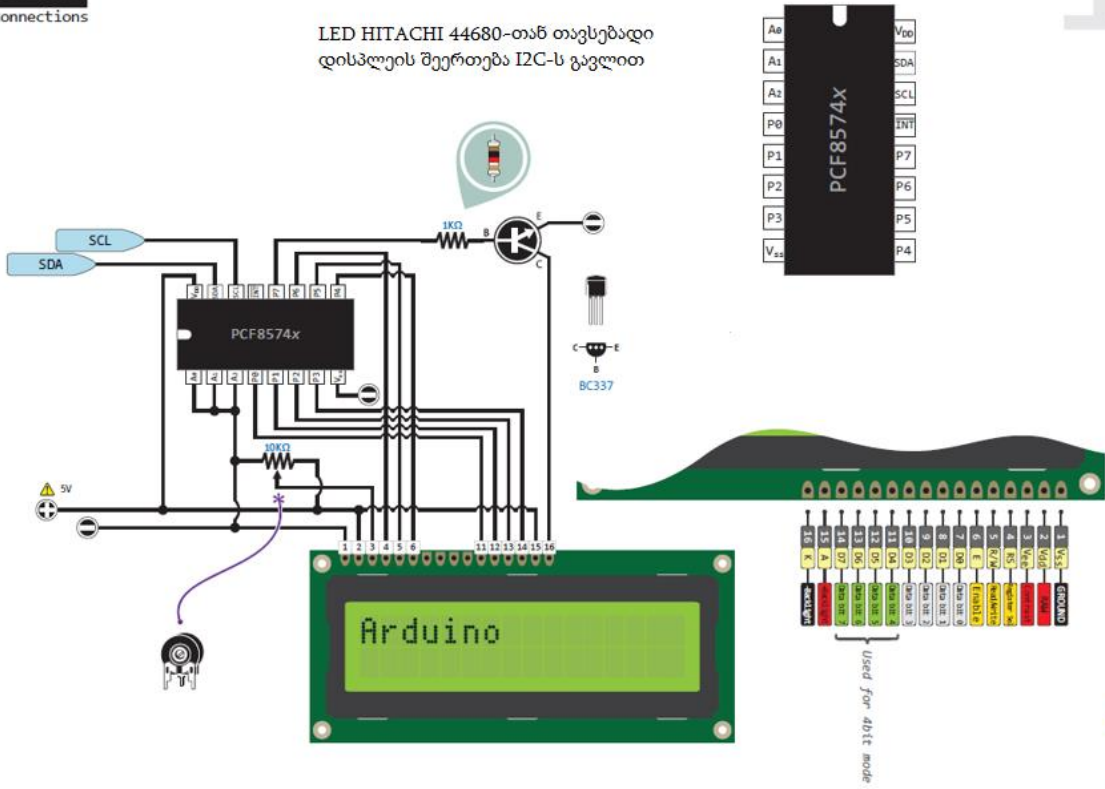
7 სეგმენტა დისპლეი (საერთო ანოდით) MAX7221 მიკროსკემის გამოყენებით



LED HITACHI-თან თავსებადი დისპლეის შერთება

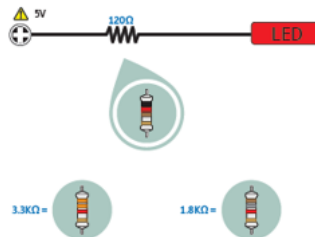
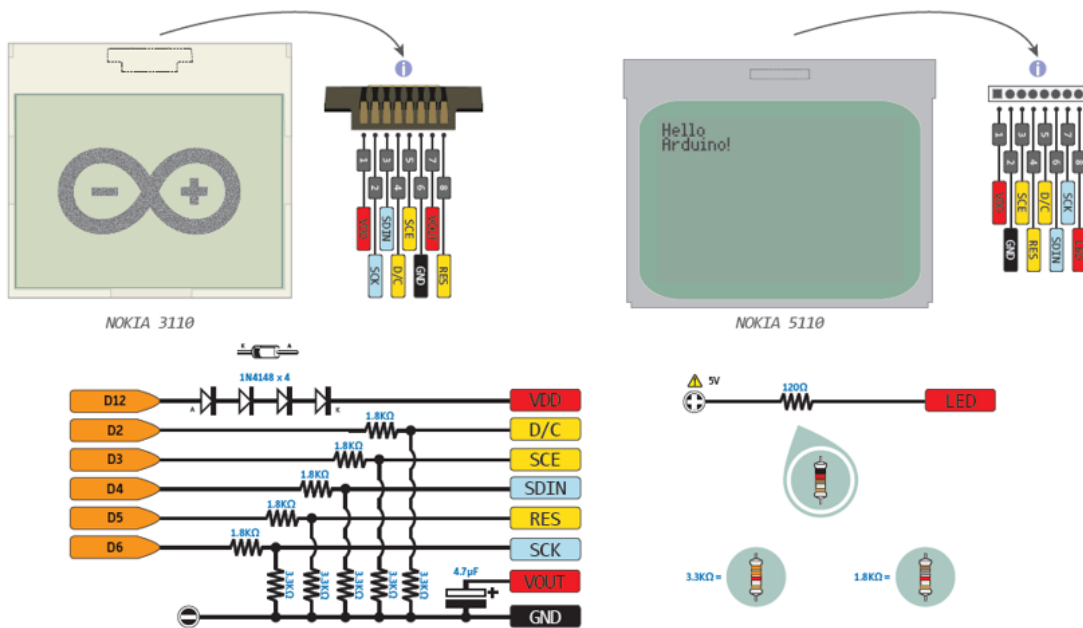


LED HITACHI 44680-თან თავსებადი
დისპლეის შეერთება I2C-ს გალით

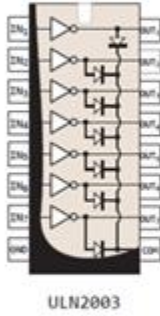
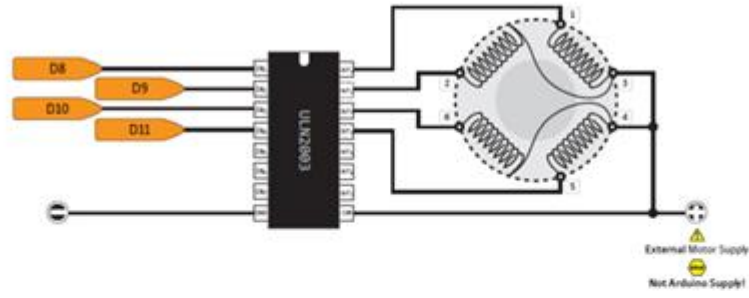


NOKIA LCD-სთან (ბაზური)
შეერთება

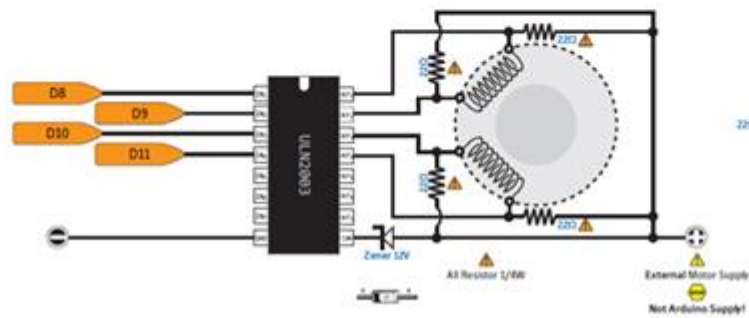
⚠ Only for 5V Arduino



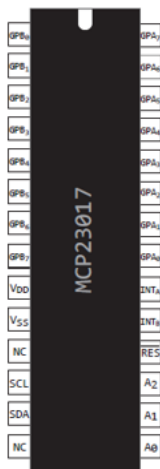
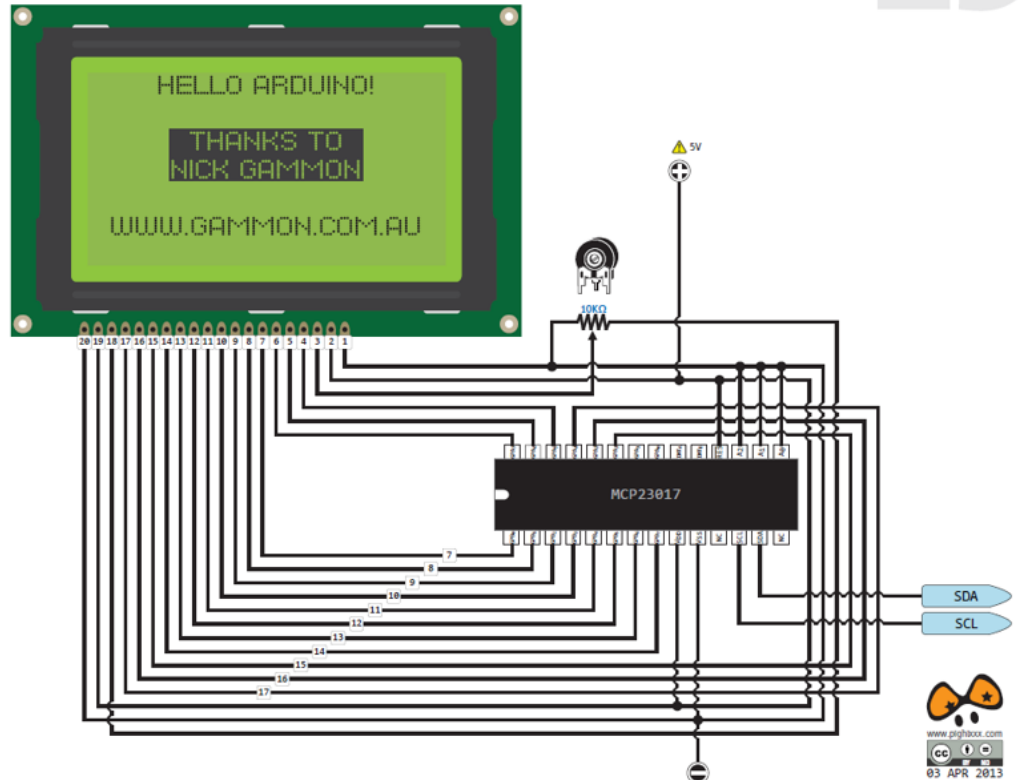
უნიპოლარული ზიგურტი ძრავის ამბრავი (ზაზისური 1)



ბიპოლარული ზიგურტი ძრავის ამბრავი (ზაზისური 2)



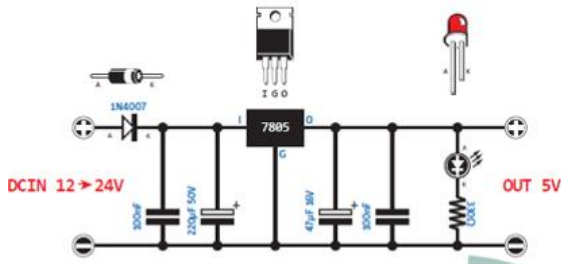
გრავიკული LCD დისპლეის შეერთება I2C-ს გავლით



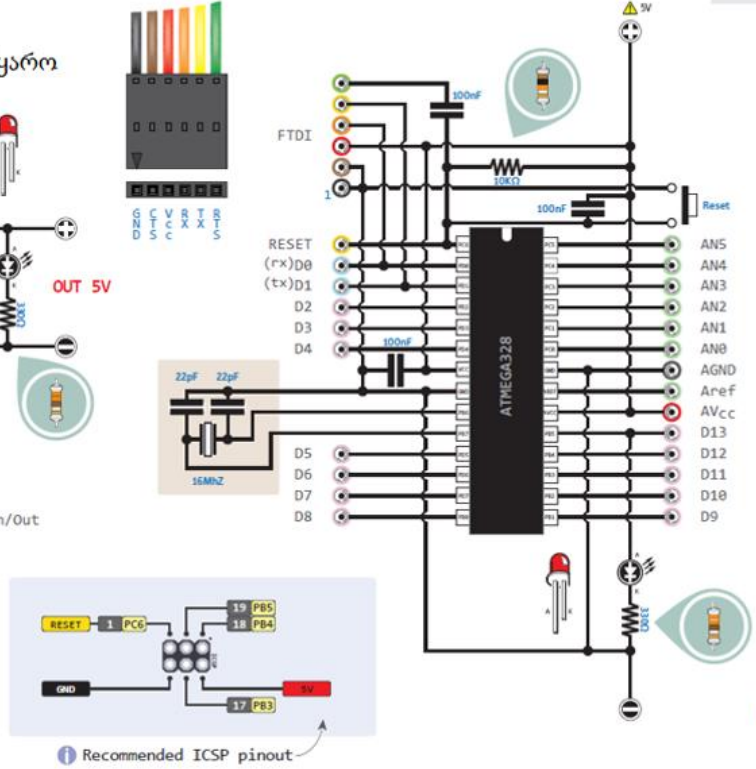
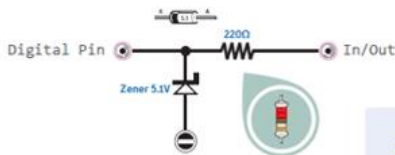
FTDI შემავრთველი

DIY Arduino

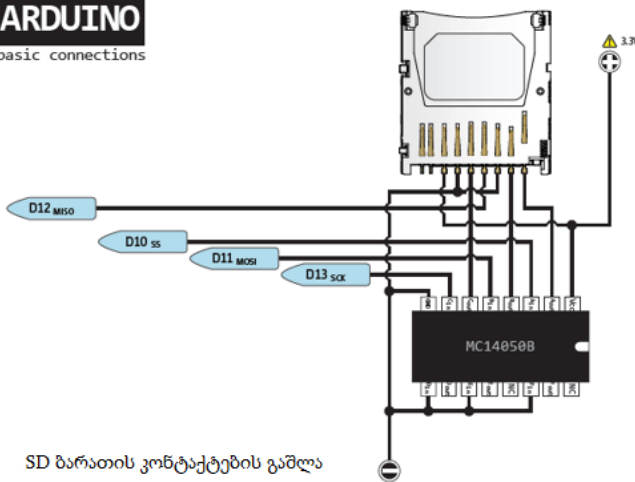
მარტივი 5 ვოლტიანი კვების წყარო



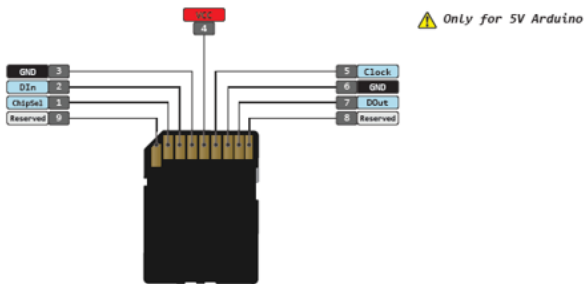
I/O კონტაქტების დაცვა



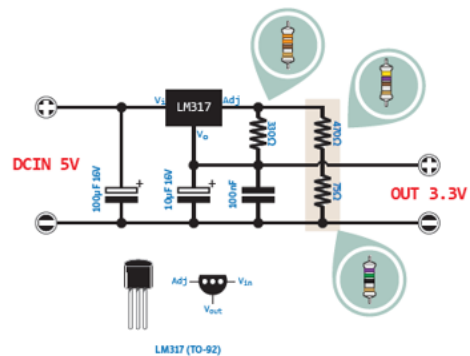
SD ბარათთან მიერთება



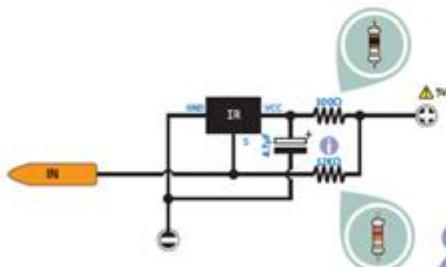
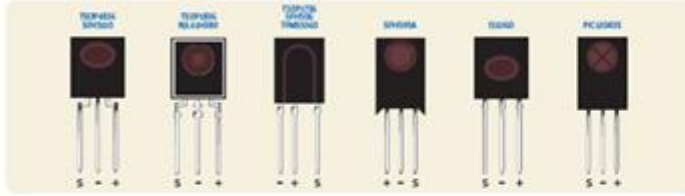
SD ბარათის კონტაქტების გამლა



მარტივი 3.3 ვოლტის კვების წყარო

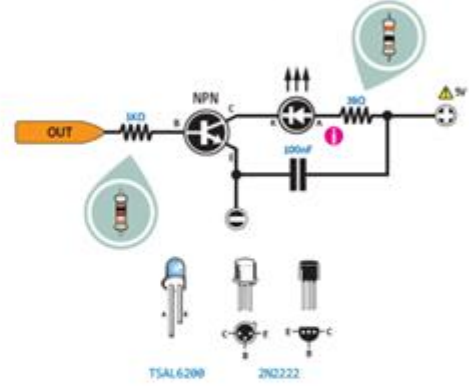


IR (ინფრაწითელი) სენსორებთან შეერთება



i Recommended to suppress power supply disturbances

IR (ინფრაწითელი) ემიტორთან შეერთება



i

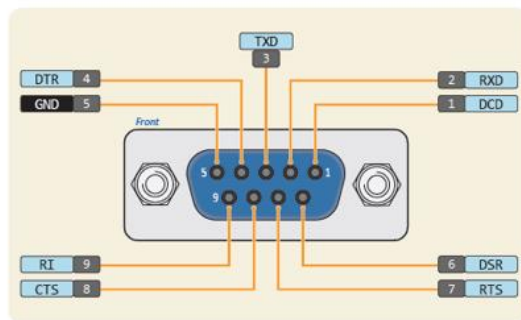
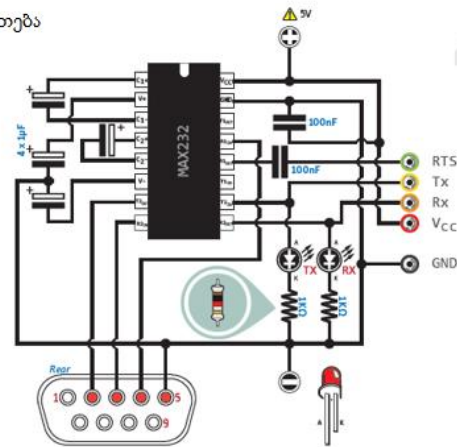
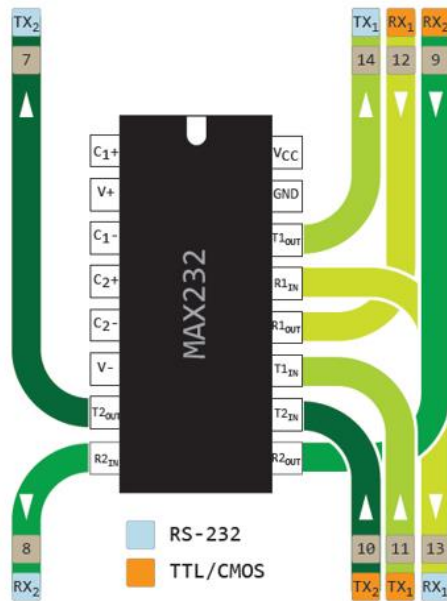
$$R = \frac{V_{in} - V_f}{I_f} + 1000$$

V_{in} Source Voltage
 V_f Forward Voltage Led
 I_f Forward Current Led



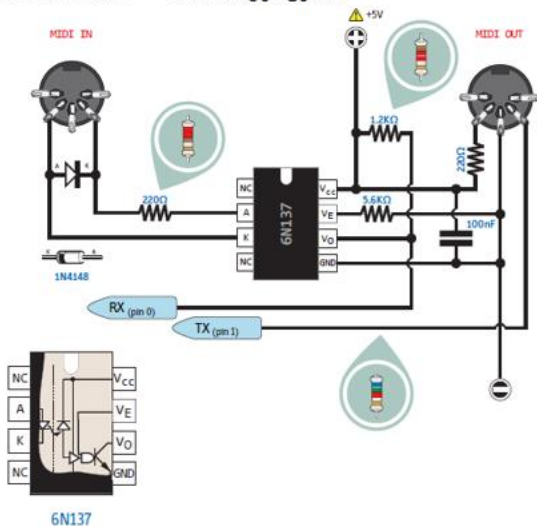
MAX232 - სიან შეერთება

21

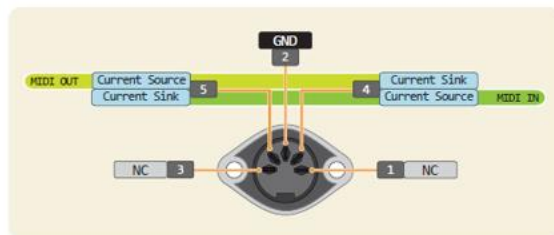
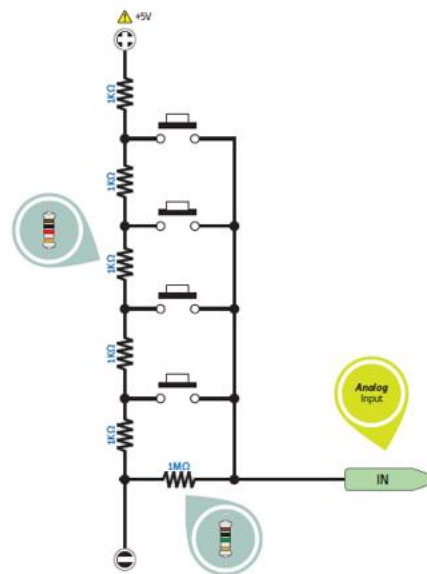


MIDI იბტერფისი

22

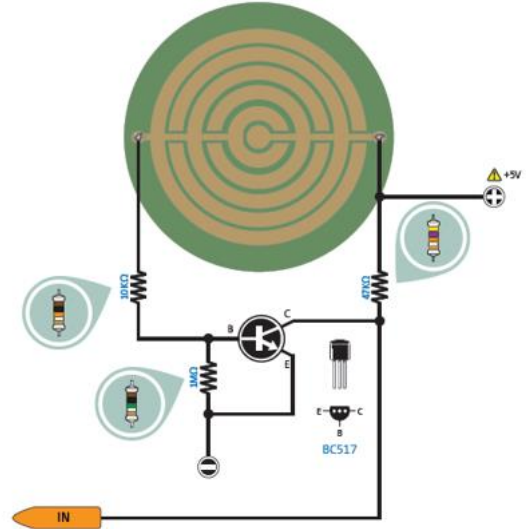
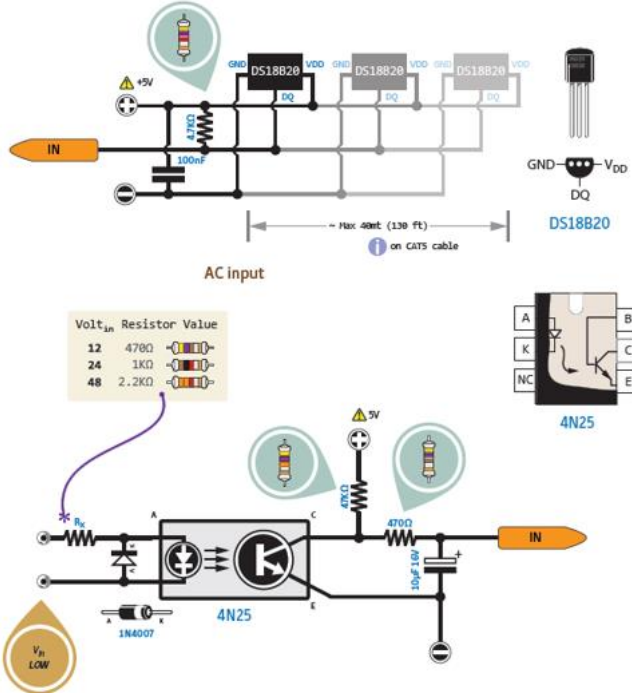


მრავალი დილაკი 1 ანალოგური შესასვლელის გამოყენებით

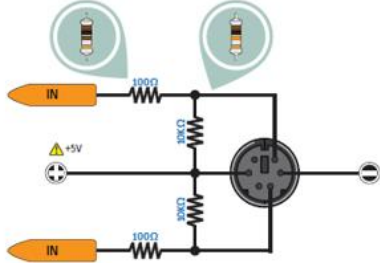


ციფრული ტემპერატურული გადამწოდის (DS18B20) მიერთება

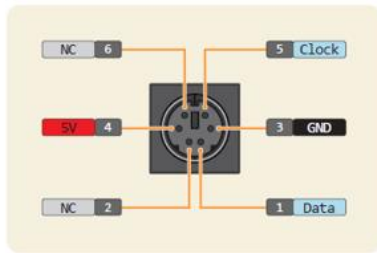
მარტივი წვიმის სენსორი არდუინოსთან



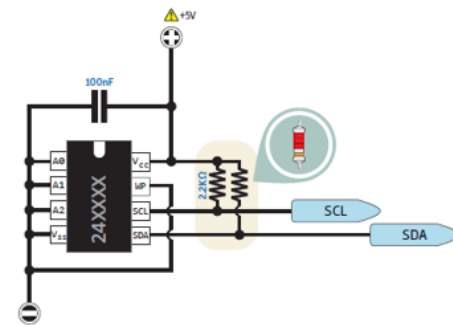
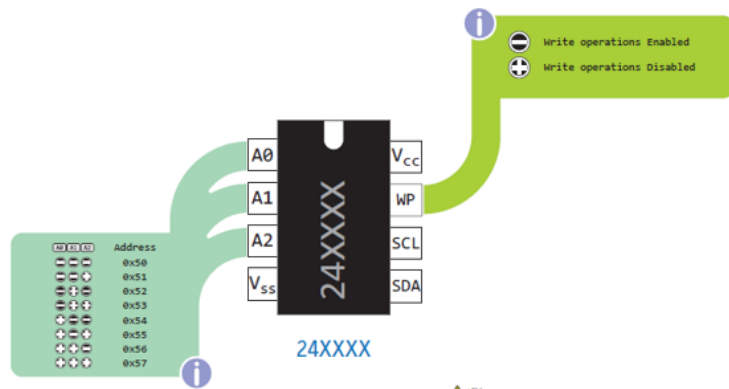
შეერთება PS2 კლავიატურასთან



სკან კოდი



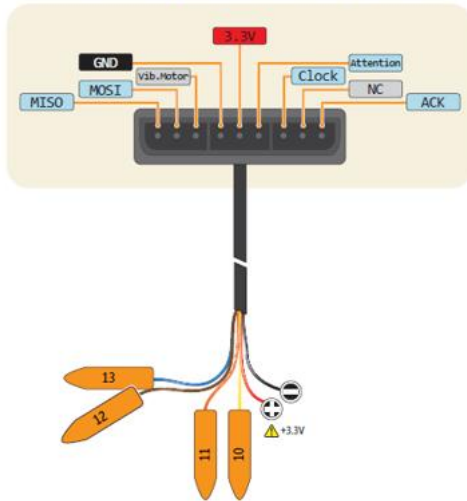
EEPROM - თან შეერთება I2C - ს გაგლით



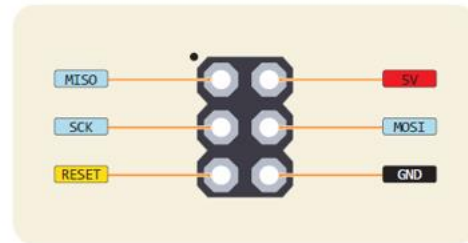
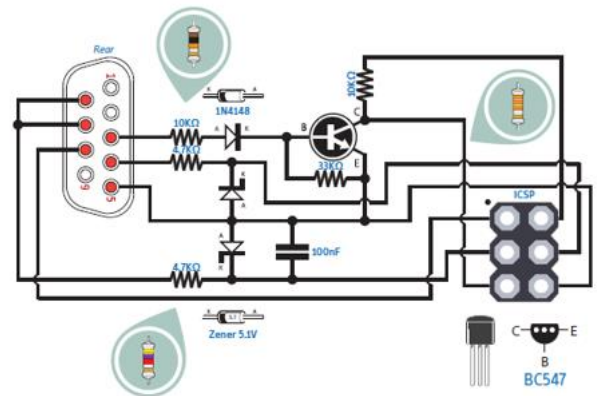
24x01x	1Kb	24x32x	32Kb
24x02x	2Kb	24x64x	64Kb
24x04x	4Kb	24x128x	128Kb
24x08x	8Kb	24x256x	256Kb
24x16x	16Kb	24x512x	512Kb



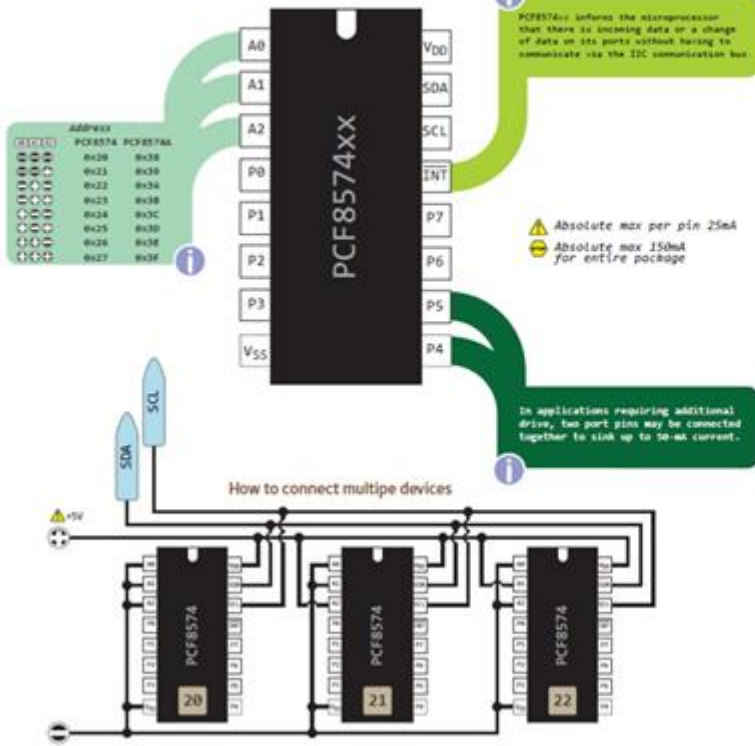
PS2 ორმაგ კონტროლერთან შეერთება



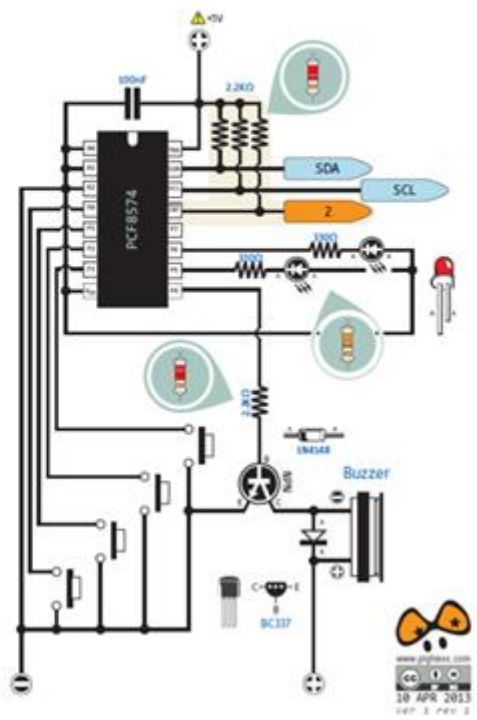
მარტივი პროგრამატორი



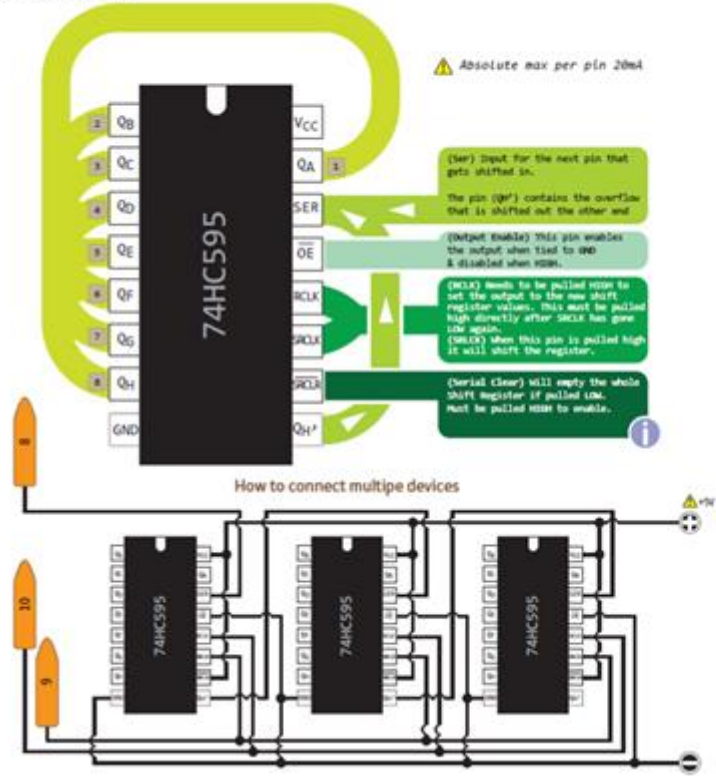
გაფართოების პორტი (PCF8574xx)



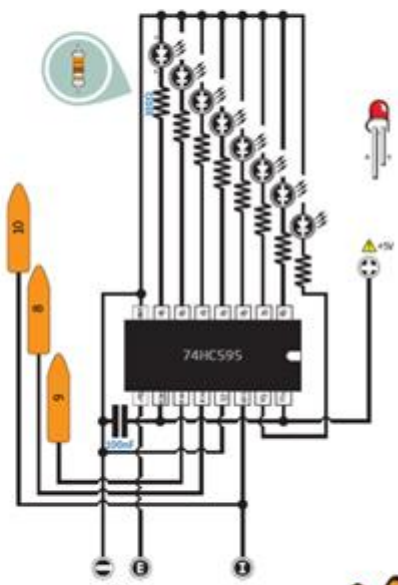
ტიპური გამოყენება



წანაცვლების რეგისტრი (74HC595)

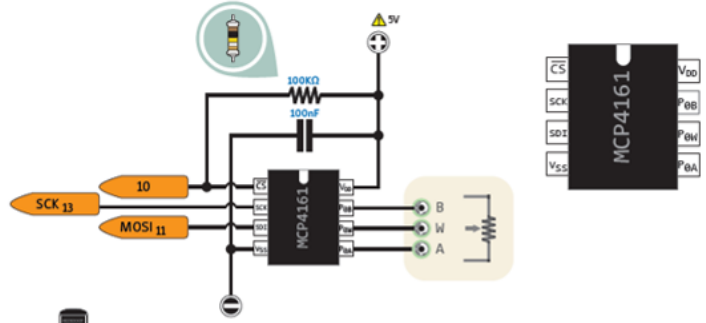
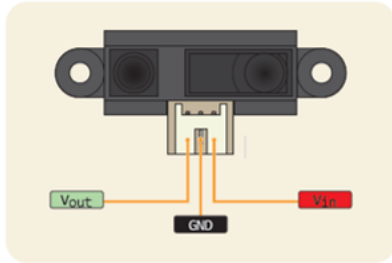


ტიპური აპლიკაცია

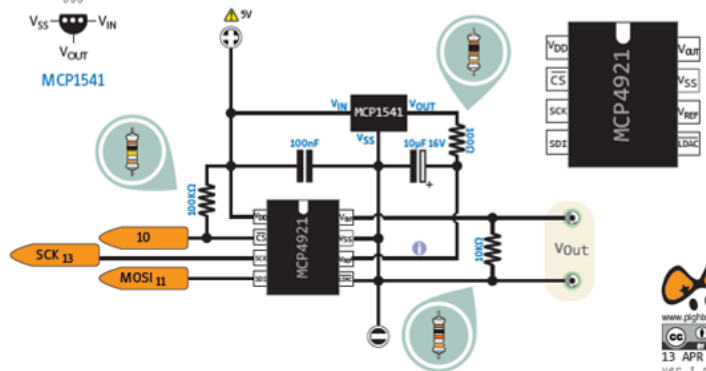
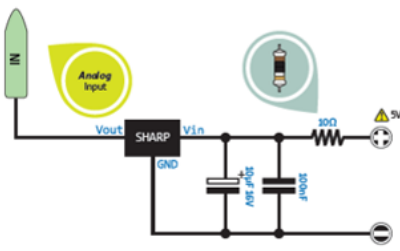


ციფრული პოტენციომეტრის (MCP4161) შეერთება

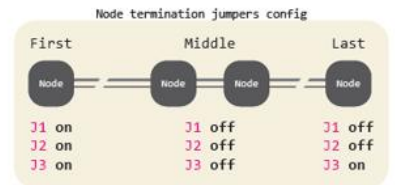
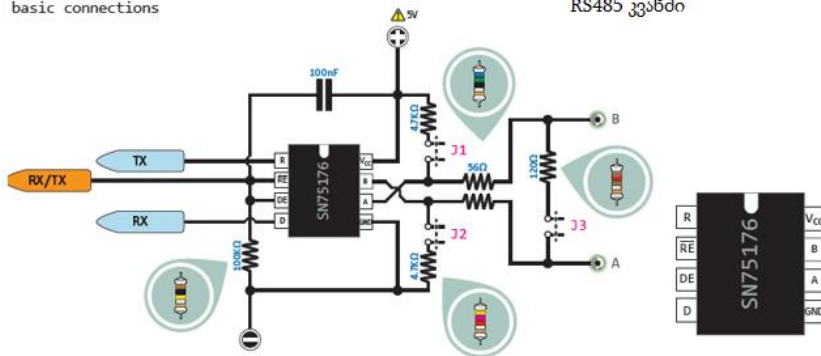
დისტანციური გადამწოდის (GP2Y0A21) შეერთება



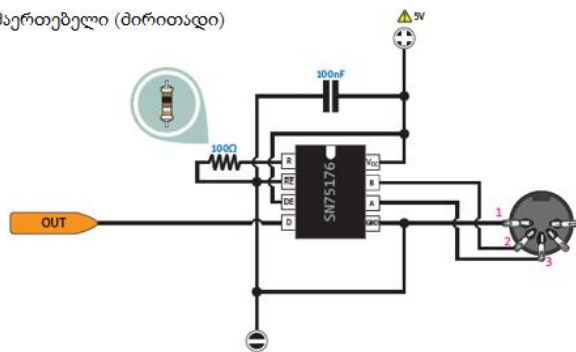
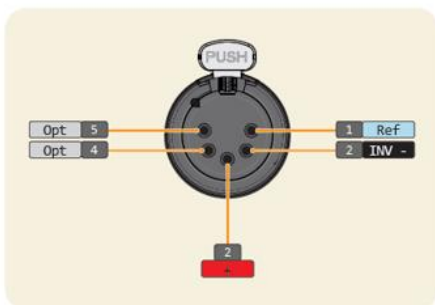
DAC (MCP 4921) - თან შეერთება



RS485 კვანძი



DMX შემავრთებელი (ძირითადი)



სიზუსტე (რეზისტორები)

ოქრო ±5%
ვერცხლი ±10%



შავი	0	0	x10
ყავისფერი	1	1	x100
წითელი	2	2	x1000
წარინჯისფერი	3	3	x1,0000
ყვითელი	4	4	x10,0000
მწვანე	5	5	x100,0000
ლურჯი	6	6	x1,000,0000
იისფერი	7	7	
რუხი	8	8	
თეთრი	9	9	

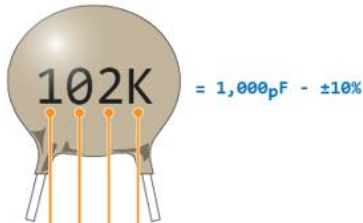
KΩ = x1,000Ω

MΩ = x1,000,000Ω

მამრავლი



კერამიკული ტევადობების კოდები



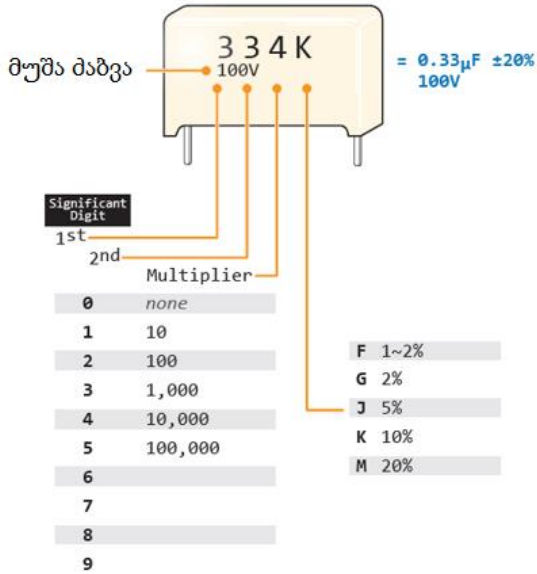
Significant Digit	1st	2nd	მამრავლი	
0	none			
1	10			C ±0.25pF
2	100			J ±5%
3	1,000			K ±10%
4	10,000			M ±20%
5	100,000			D ±0.5pF
6	1,000,000			Z +80% -20%
7				
8				
9				

1,000 pF 1 nF 0.001 μF

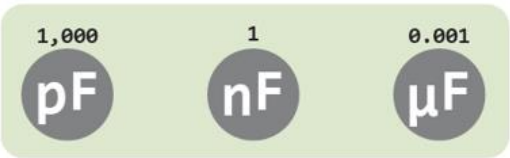
1 1p0 1pF	10 10pF	101 n10 100pF
1.2 1p2 1.2pF	12 12pF	121 n12 120pF
1.5 1p5 1.5pF	15 15pF	151 n15 150pF
1.8 1p8 1.8pF	18 18pF	181 n18 180pF
2.2 2p2 2.2pF	22 22pF	221 n21 220pF
2.7 2p7 2.7pF	27 27pF	271 n27 270pF
3.3 3p3 3.3pF	33 33pF	331 n33 330pF
3.9 3p9 3.9pF	39 39pF	391 n39 390pF
4.7 4p7 4.7pF	47 47pF	471 n47 470pF
5.6 5p6 5.6pF	56 56pF	561 n56 560pF
6.8 6p8 6.8pF	68 68pF	681 n68 680pF
8.2 8p2 8.2pF	82 82pF	821 n82 820pF



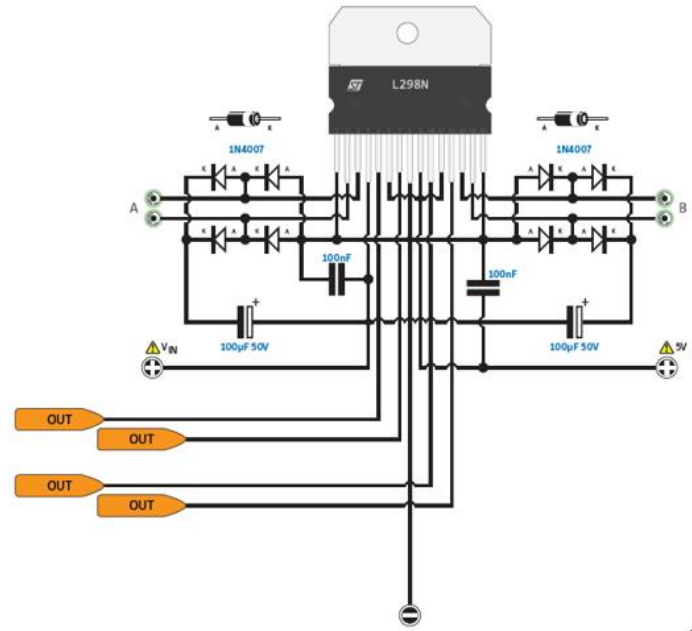
პოლისტიროლის აფსკიანი კონდესატორის კოდები

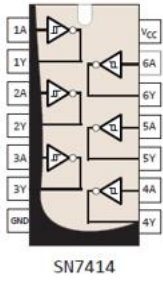


102	1n	.001	1,000pF	103	10n	.01	10,000pF	104	100n	.1	100,000pF
122	1n2	.0012	1,200pF	123	12n	.012	12,000pF	124	120n	.12	120,000pF
152	1n5	.0015	1,500pF	153	15n	.015	15,000pF	154	150n	.15	150,000pF
182	1n8	.0018	1,800pF	183	18n	.018	18,000pF	184	180n	.18	180,000pF
222	2n2	.0022	2,200pF	223	22n	.022	22,000pF	224	220n	.22	220,000pF
272	2n7	.0027	2,700pF	273	27n	.027	27,000pF	274	270n	.27	270,000pF
332	3n3	.0033	3,300pF	333	33n	.033	33,000pF	334	330n	.33	330,000pF
392	3n9	.0039	3,900pF	393	39n	.039	39,000pF	394	390n	.39	390,000pF
472	4n7	.0047	4,700pF	473	47n	.047	47,000pF	474	470n	.47	470,000pF
562	5n6	.0056	5,600pF	563	56n	.056	56,000pF	564	560n	.56	560,000pF
682	6n8	.0068	6,800pF	683	68n	.068	68,000pF	684	680n	.68	680,000pF
822	8n2	.0082	8,200pF	823	82n	.082	82,000pF	824	820n	.82	820,000pF

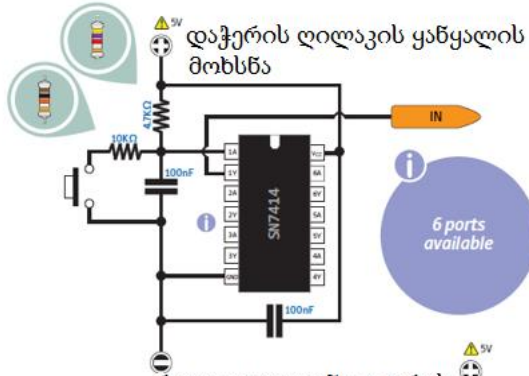


L298 ძრავის ამძრავი

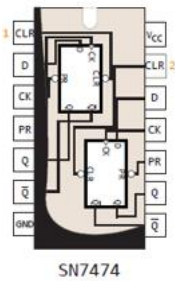
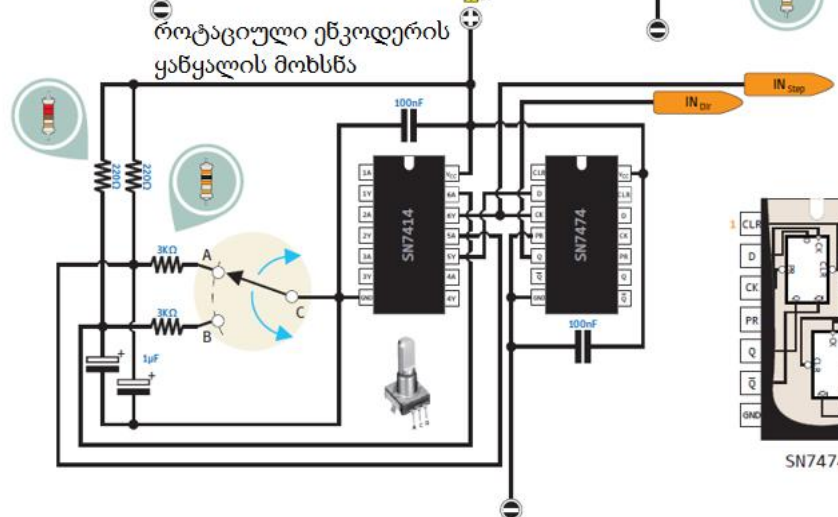
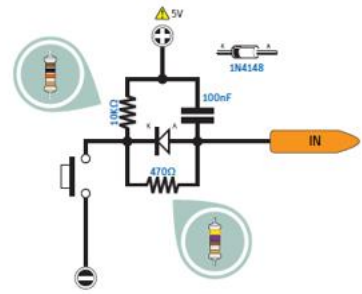




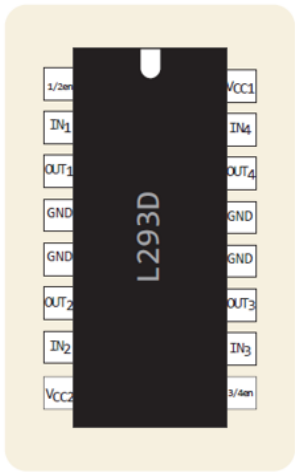
SN7414



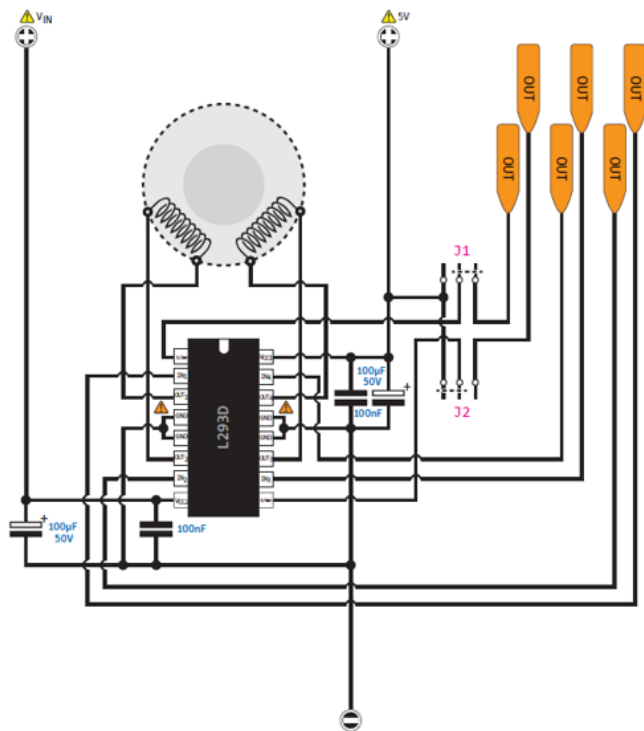
მარტივი ყანყალის მომხსნელი



SN7474

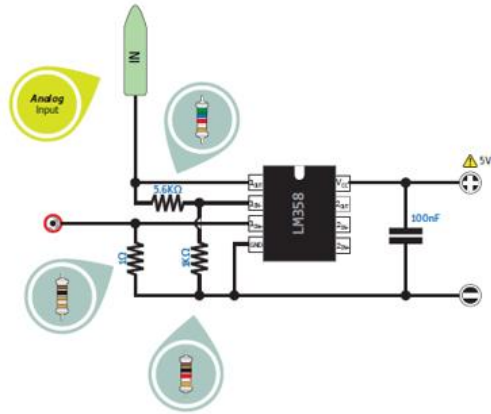
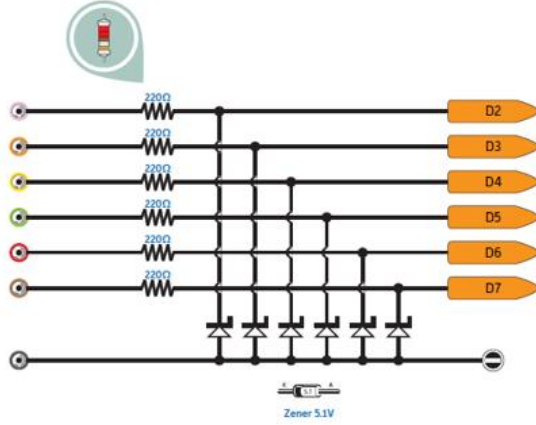


L293D ბიჯური ძრავის ამძრავი

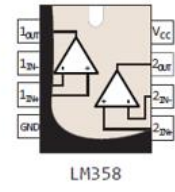


ლოგიკური ანალიზატორი არდუინოზე

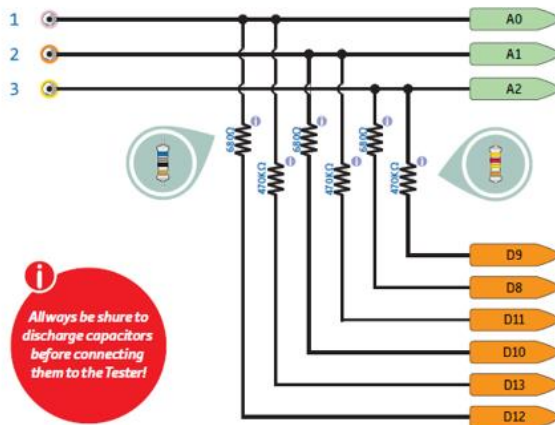
დენის მგრძობელი



See instructions at la.pighixx.com



კომპონენტების ტესტერი არდუინოზე (ბაზისური)

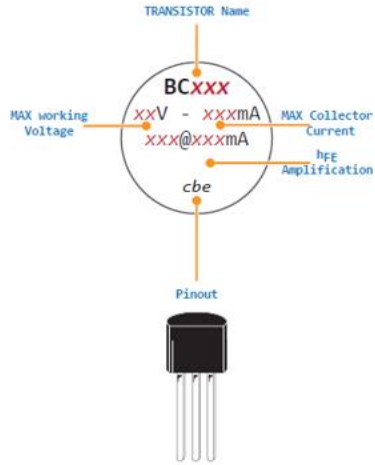


Allways be shure to discharge capacitors before connecting them to the Tester!

To get full accuracy use 1% tolerance resistors

Download sketch at at.pighixx.com





- NPN
- PNP

