

საქართველოს ტექნიკური უნივერსიტეტი

ჯ. გრიგალაშვილი



მეთოდური მითითებები Arduino - ს შესწავლისათვის

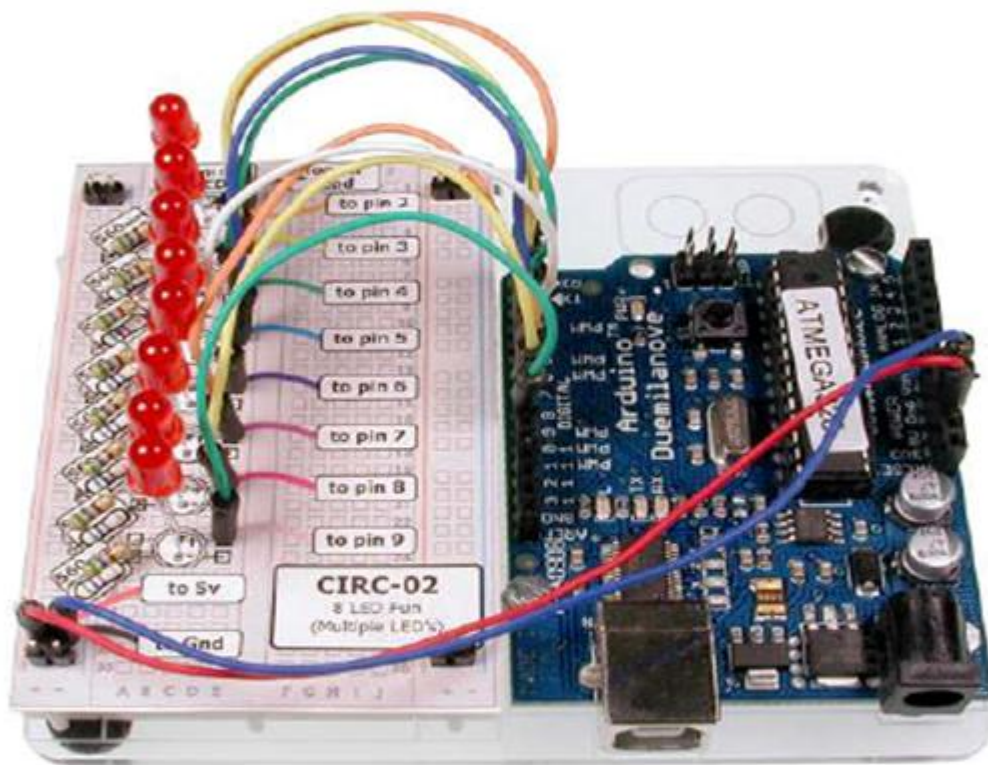
დამტკიცებულია სახელმძღვანელოდ
სტუ-ს სარედაქციო-საგამომცემლო
საბჭოს მიერ, ოქმი №

თბილისი 2016

ARDX

კომპლექტი Arduino ექსპერიმენტებისათვის

მეთოდური მითითებები Arduino-ს შესწავლისათვის



ARDX

Arduino ღია საწყისი კოდით
დამზარე სახელმძღვანელო

 **SOLARBOTICS** Ltd



Document Revision: May 12012

შესავალი

ამ კომპლექტის შესახებ

ეს კომპლექტი შესაძლებლობას გაძლევთ მსურველებს საინტერესოდ და სასარგებლოდ გაატაროთ დრო, აითვისოთ სხვადასხვაგვარი ელექტრონული კომპონენტები პატარა მარტივი და საინტერესო მოწყობილობების აგების გზით. შედეგად, კი თქვენ ხელში შეგრჩებათ მუშა მოწყობილობა და ინსტრუმენტი, რაც საშუალებას მოგცევთ გაიგოთ მისი მუშაობის პრინციპები.

თუკი თქვენ რაიმე არ გამოგადით, არის კითხვები ანდა აუცილებელია დამატებითი დახმარება - მოგვწერეთ ელექტრონულ მისამართზე help@oomlout.com



ღია აპარატული უზრუნველყოფის შესახებ

.:oomlout:. - ის ყველა პროექტი ღიაა. რას ნიშნავს ეს? ეს ნიშნავს იმას, რომ ყველა მონაცემი, რაც საჭიროა ამ კომპლექტის გამოყენებისათვის ხელმისაწვდომია უფასო ჩამოტვირთვისთვის. უფრო მეტიც - თქვენც შეგიძლიათ შეცვალოთ უშუალოდ ნებისმიერი მასალა თქვენი შეხედულებისამებრ და შემდეგ გაავრცელოთ იგი. რა აზრი აქვს ამას?

ყველაფერი მაღიან მარტივია, ეს მასალები გამოცემულია Creative Commons ლიცენზიით, ანუ თუ გამოიყენებთ **.:oomlout:.**-ს, თქვენ მოვალე ხდებით მოიხსენიოთ იგი თქვენს პროექტებში და გაავრცელოთ იგი ამდაგვარად. რატომ? თქვენ უკვე გაიზარდეთ ღია პროგრამულ უზრუნველყოფასთან თანაშითა და მისი შესწავლით, ამით თქვენ შეიძინეთ კარგი გამოცდილება და დადგა დრო თქვენც მიიღოთ მონაწილეობა პროექტის განვითარებაში. ჩვენ ვფიქრობთ, რომ ასეთი მიდგომა გაამართლებს და შესაძლებლობა გვექნება მივიღოთ რეალური პროექტები და რალური მათი კომპონენტები.

უფრო დაწვრილებით Creative Commons ლიცენზიაზე თქვენ შეგიძლიათ გაიგოთ ელექტრონულ მისამართზე <http://ardx.org/CCLI>

.: OOMLOUT :. - ის შესახებ

დამმუშავებლების ჩვენი პატარა, მაგრამ მამაცი კომპანიაა, რომელიც სპეციალიზირდება "შესანიშნავი ღია პროდუქციის" წარმოებაზე

ჩვენი უახლესი პროექტები შეგიძლიათ ნახოთ მისამართზე:

<http://www.oomlout.com>

Solarbotics - ის შესახებ

ჩვენ დავიწყეთ რობოტოტექნიკური BEAM კომპლექტების წარმოება ამ 15 წლის წინათ, ამჟამად კი ჩვენ ვაზნადებთ აგრეთვე მაღალი კლასის ელექტრონულ კომპონენტებს

<http://www.solarbotics.com/>

პრობლემების შესახებ

ჩვენ ვცდილობთ მოგაწოდოთ უმაღლესი ხარისხის საქონელი. თუ თქვენ აღმოაჩენთ არათვალნათელ ინსტრუქციას, დეტალების ნაკლებობას ანდა უზრალოთ მოისურვებთ შეგვეკითხოთ რაზე, ჩვენ გავაკეთებთ ყველაფერს რათა დაგეხმაროთ თქვენ

help@oomlout.com/help@solarbotics.com

(ჩვენთვის უმჯობესია ვიცოდეთ პრობლემების შესახებ, რათა გავაუმჯობესოთ შემდეგი ვერსიები)

მადლობა იმისათვის, რომ აირჩიეთ **.:oomlout:.**

(და Solarbotics)

.: შინაარსი .:

შესავალი/მზადება პროექტისათვის

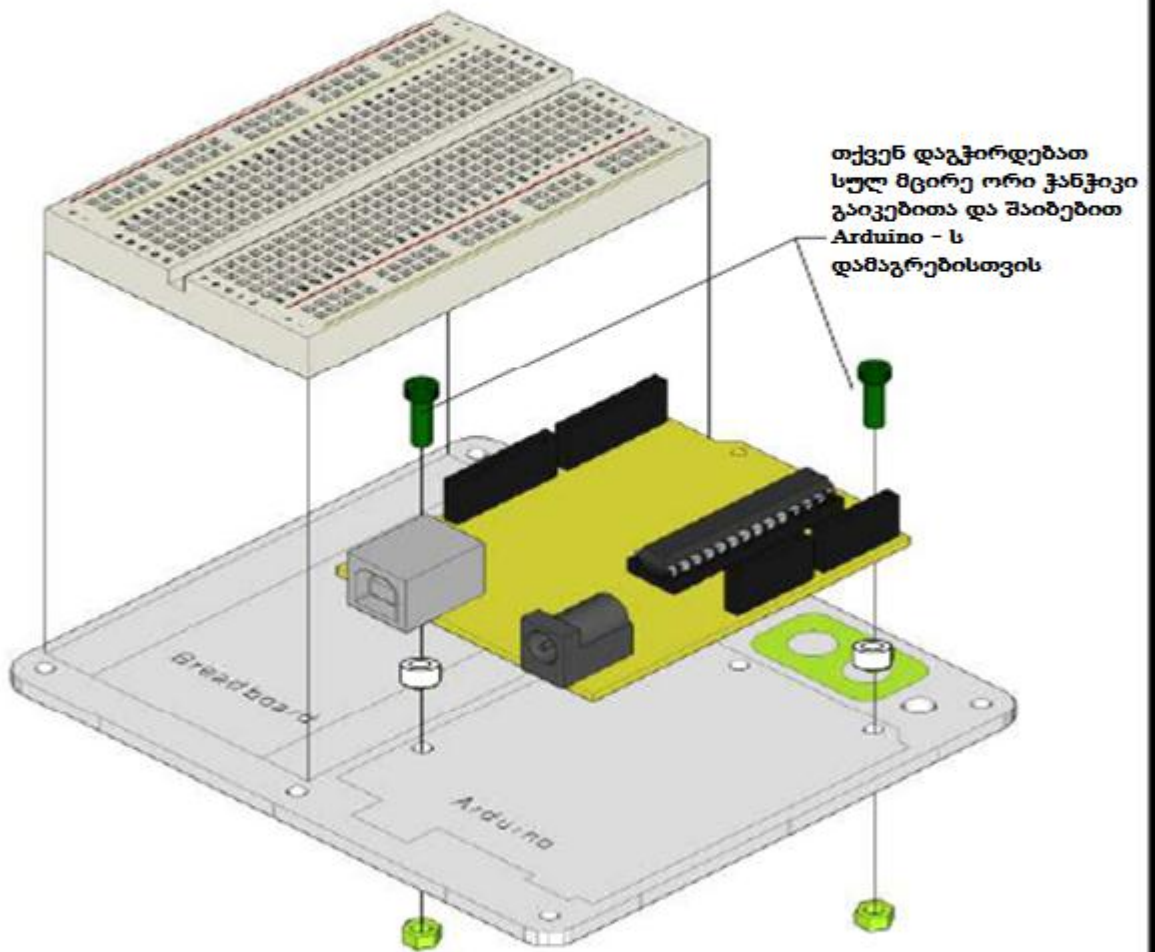
{ASEM}	მოწყობილობის აწყობა	02
{INST}	პროგრამული უზრუნველყოფის დაყენება	03
{PROG}	მცირეოდენი ცნობები პროგრამირებაში	04
{ELEC}	მცირეოდენი ცნობები ელექტრონიკაში	06
<u>მოწყობილობები</u>		
{CIRC01}	ვიწყებთ მუშაობას (მოციმციმე შუქდიოდი)	08
{CIRC02}	სქემა 8 შუქდიოდზე	10
{CIRC03}	ბრუნვა (ტრანზისტორი და მოტორი)	12
{CIRC04}	ერთეული სერვოდრავი (სერვოდრავები)	14
{CIRC05}	8 შუქდ. (წანაცვლების რეგისტრი 74HC595)	16
{CIRC06}	მუსიკა (პიეზოელემენტი)	18
{CIRC07}	ლილაკზე დაჭერა (ლილაკები)	20
{CIRC08}	გრეხვები (პოტენციომეტრები)	22
{CIRC09}	შუქი (ფოტორეზისტორები)	24
{CIRC10}	ტემპერატურა (ტემპ. გადამწოდი TMP56)	26
{CIRC11}	დიდი დატვირთვები (რელეები)	28
{CIRC12}	მრავალფერა ნათება (შუქდიოდები RGV)	30

01 ASEM

კომპონენტების
აწყობა

:: მოწყობილობის აწყობა ::

 საფენი Arduino -ხათვის x1	 სამაკეტო დაფა x1	 Arduino x1
 4-40 x 3/8" ჭანჭიკი x3	 4-40 გაიკა x3	 #4 x 1/16" შაიბა x3



:: Arduino - ხ საფუძვლების შესწავლისათვის ეწვიეთ ::

:: <http://ardx.org/INTR> ::

პროგრამული უზრუნველყოფის დაყენება

02 INST

ინსტალიაცია

პროგრამული და აპარატული უზრუნველყოფა

(სამომხმარებლო ინტერფეისის IDE) .:

სამომხმარებლო ინტერფეისის დანიშნულება Arduino-ს დაპროგრამებაა. დასაწყისში ეს მოგეჩვენებათ მხელ საქმეთ, მაგრამ პროგრამის დაყენებისა და მასზედ მცირეოდენი წამუშავეების შემდეგ პროგრამის ყველა საიდუმლო გასაკები იქნება თქვენთვის.

ბიჯი 1: პროგრამული უზრუნველყოფის ჩამოტვირთვა
 გახსენით გვერდი
<http://arduino.cc/en/Main/Software> და ჩამოტვირთეთ ვერსია თქვენი ოს-ისათვის

Windows XP

ბიჯი 2: არქივის გახსნა პროგრამა

arduino-00xx-win.zip (xx-ვერსიის ნომერი)
 უმჯობესია საქალაქდემში c:\program Files\

ბიჯი 3: იარლიყის შექმნა გახსენით

c:\program Files\arduino-00xx\9xx - ვერსიის ნომ.)
 დააწკაპეთ თავის მარჯვ ღილაკით
 Arduino.exe (send to.Desktop (შექმ. იარლიყი)

ბიჯი 4: მიერთება მიაერთეთ თქვენი arduino USB

კაბელით თავისუფალ USB პორტს
 ეკრანზე უნდა გამოჩნდეს
 დიალოგური ფანჯარა

ბიჯი 5: ახალი აპარატურის დაყენება

გამოტოვეთ ოფცია "დრაივერების ძებნა
 ინტერნეტში" (უბრალდ დაწკაპება "next"-ზე)
 შემდეგში

დააწკაპეთ "Install from a list or specific location
 (Advanced)"

დასაყენებელი დირექტორია Uno Board
 c:\program files\arduino-00xx/drivers/

Vista, Seven

ბიჯი 5: ახალი აპარატურის დაყენება

გაუშვით Device Manager

Stat>Run>devmgmt.msc

აირჩიეთ Arduino

Other Devices>Arduino Uno (Uno)

განაახლეთ დრაივერი

დააწკაპეთ "Update Driver"

აირჩიეთ დრაივერი

დააწკაპეთ "Browse My Computer for Driver Software"

c:\program Files\ arduin-00xx/drivers

Mac OSX

ბიჯი 2: გახსენით.dmg

გახსენით (ჩატვირთეთ)
 arduino-00xx-mac.dmg
 (xx-ვერსიის ნომერი)

ბიჯი 3: გადაიღეთ ასლი დანართის

მოძებნეთ "Arduino" (მონწყობილობის
 ბრაუზერში "Devices" გადაადგილეთ
 დანართი "Arduino" საქალაქდემში
 "Applications" (დანართები)

ბიჯი 4: დააყენეთ დრაივერები

..მხოლოდ Duamilanove dafebisaTvis.:
 მოძებნეთ
 მონწყობილობა "Arduino"
 თავის ორმაგი დაწკაპებით დააყენეთ
 FTDI Drivers for Intel Macs (X X XX).pkg
 (FTDI Drivers for PPC Macs (X_X_XX).pkg
 გადატვირთეთ

ბიჯი 5: ჩართეთ Arduino

შეერთეთ თქვენი Arduino USB-კაბელის
 დახმარებით თავისუფალ პორტს USB

..დაყენებისას წარმოიშვა პრობლემები..

.. საჭიროა დაწვრილებითი ცნობები?

იყენებთ Linux? ..

..:http://ardx.org/LINU ..

Arduino-ს დაპროგრამების საფუძვლები

Arduino-ს დაპროგრამირება ხდება C ენაზე. ეს განყოფილება ორიენტირებულია მათზე, რომლებსაც აქვთ დაპროგრამირების გარკვეული გამოცდილება და ჭირდებათ მხოლოდ C ენის თავისებურებებისა და Arduino-ს ინტერფეისის განმარტება. თუ ეს ყველაფერი მოგეჩვენებათ თქვენ რამდენადმე რთულად, ნუ შეწუხდებით, დაიწყეთ მუშაობა მაკალითებით მოწყობილობებზე და გაკება მოვა თავისთავად. საფუძვლების უფრო ღრმა შესწავლისათვის მიმართეთ საიტს arduino.cc.

სტრუქტურა

Arduino-ს პროგრამებს უწოდებენ აგრეთვე "სკეჩ"-ს და ისინი შეიცავს ორ სავალდებულო ფუნქციას ანუ ქვეპროგრამას

void setup() { }
ის გამოსახულებები, რომლებიც ფიგურულ ფრჩხილებშია მოთავსებული სრულდება მხოლოდ ერთხელ, პროგრამის პირველი გაშვებისას

void loop() { }
ეს პროგრამა სრულდება უსასრულოდ, ციკლურად თვის კვების გამორთვამდე, setup ქვეპროგრამის დაშთავრების შემდეგ

სინტაქსი

C ენაში მოთხოვნები ფორმატირების შესახებ იწვევს სიძნელებს დაწყებებისათვის (მეორეს მხრივ თავისი სტრუქტურისგან გამომდინარე C ენას გააჩნია დიდი შესაძლებლობები). თუ კი თქვენ დაიხსომებთ შემდეგ წესებს ეს სრულიად საკმარისი იქნება დასაწყისში

// (ცალმხრივი კომენტარი) ხშირად გამოიყენება პროგრამის ტექსტში კომენტარებისათვის. აქ შეიძლება განმარტებულ იქნას თუ რას ნიშნავს თვითოეული სტრიქონი პროგრამაში. რაც ჩაიწერება ორმაგი ხაზის შემდეგ სტრიქონის ბოლომდე კომპილიატორის მიერ იგნორირდება.

[] (ფიგურული ფრჩხილები) გამოიყენება ბრძანებების ბლოკის საწყისისა და ბოლოს აღნიშნისათვის (გამოიყენება ფუნქციებში და ციკლებში)

/* */ (მრავალსტრიქონი კომენტარები). თქვენ შეგიძლიათ გამოიყენოთ ეს სტრუქტურა, თუ გნებავთ შექმნათ კომენტარები რამდენიმე სტრიქონზე, ყველაფერი რაც ამ სიმბოლოებს შორისაა იგნორირდება კომპილიატორით.

; (წერტილმძიმე) თვითოეული ბრძანება უნდა დამთავრდეს ამ სიმბოლოთი (დაკარგული წერტილმძიმე - ის მეტად გავრცელებული შეცდომაა, რაც იწვევს კომპილიაციის შეუძლებლობას).

ცვლადები

ნებისმიერი ბრძანება მხოლოდ და მხოლოდ მანიპულირებს რიცხვებზე. ცვლადები გვებმარებიან ციფრებით ფონგლირებაში.

int (მთლიანრიიცხვანი) ხშირად გამოიყენება, ინახება მეხსიერებაში ორ ბაიტად (16 ბიტი) შეიძლება მოთავსებულ იქნას რიცხვი -32 768 ... 32 767 დიაპაზონში)

long (გრძელი) გამოიყენება იმ შემთხვევაში, როდესაც არ ყოფნის int-ის მოცულობა. მეხსიერებაში იკავებს 4 ბაიტს და აქვს -2 147 483 648 ... 2 147 483 647 დიაპაზონი

boolean (ორობითი) ცვლადის მარტივი ტიპია ჭეშმარიტი/მცდარი-ს სახის, იკავებს მხოლოდ 1 ბიტს მეხსიერებაში

float (მცოცავი მძიმით) გამოიყენება მცოცავი მძიმით გაანგარიშებისათვის. იკავებს მეხსიერებაში 4 ბაიტს და აქვს დიაპაზონი -3. 4028235E+38.

char (სიმბოლო), გამოიყენებს ASCII კოდირებას (მაგ. 'A' = 65). მოიცავს მეხსიერების ერთ ბაიტს (8 ბიტი). Arduino ოპერეირებს სტრიქონებთან როგორც სიმბოლოების მასივთან.

∴ უფრო დაწვრილებითი ცნობებისათვის დაპროგრამებისათვის შესახებ მიაკითხეთ ∴

<http://ardx.org/PROG>

03 PROG დაპროგრამება

მათემატიკური ოპერატორები

ოპერატორები გამოიყენება რიცხვთა გარდაქმნებისათვის

= (მიკუთვნება) გადააქცევს რაიმეს რისიმეს ტოლად. (მაგ. $x=10^*2$ ახდენს რიცხვის 20 ჩაწერას ცვლად x-ში)
 % (ნაშთი გაყოფისაგან), მაგ. $12\%10$ იძლევა შედეგს 2 (შევრება)
 + (გამოვლება)
 - (გამოვლება)
 * (გამრავლება)
 / (გაყოფა)

შედარების ოპერატორები

ოპერატორები გამოიყენება ლოგიკური შედარებებისათვის

== (ტოლია) (მაგ. $12==10$ მცდარია (FALSE), $5==5$ ჭეშმარიტია)
 != (არ არის ტოლი) (მაგ. $12!=10$ (TRUE), $5!=5$ მცდარია),
 < (ნაკლებია) (მაგ. $12<10$ მცდარია, $12<24$ ჭეშმარიტია (TRUE)
 > (მეტია) (მაგ. $12>10$ ჭეშმარიტია (TRUE), $12>12$ მცდარი)

მმართველი ოპერატორები

ბრძანებების (ბრძანებების ბლოკების) შესრულების თანმიმდევრობის გასარკვევად გამოიყენება მმართველი სტრუქტურები. აქ მოყვანილია მხოლოდ ძირითადი მათგანი. უფრო დაწვრილებით საიტზე- Arduino.

if (პირობა 1) { }
else if (პირობა 2) { }
else { }
 თუ პირობა 1 სწორია სრულდება ბრძანება I ფიგურულ ფრჩხილებში. თუ პირობა 1 მცდარია მაშინ მოწმდება პირობა 2. თუ იგი სწორია, მაშინ სრულდება ბრძანება II ფიგურულ ფრჩხილებში, თუ იგი მცდარია, მაშინ სრულდება ბრძანება III ფიგურულ ფრჩხილებში

for (int i = 0; f < განმეორებათა რიცხვი; f++) { }
 ეს სტრუქტურა გამოიყენება ციკლის განსაზღვრისათვის. ციკლი მეორდება მოცემულ რიცხვჯერ. ცვლადი i შეიძლება გაიზარდოს ანდა შემცირდეს.

ციფრული სიგნალები

digitalWrite (pin, value);
 თუ პორტი დაყენებულია OUTPUT რეჟიმში, მასში შეიძლება ჩაიწეროს HIGH (ლოგ. ერთიანი, +5ვ) ან LOW (ლოგ. ნოლი, GND)

pinMode (pin, mode);
 გამოიყენება პორტების რეჟიმის განსაზღვრისათვის. პორტების მისამართებად შეიძლება გამოიყენებულ იქნას ნომრები 0-დან 19-მდე. აქედან 14 ... 19 გამოიყენება ანალოგური პორტების 0 ... 5 აღწერისათვის). რეჟიმი შეიძლება იყოს ან INPUT ან OUTPUT

digitalRead (pin, mode);
 თუ პორტი დაყენებულია INPUT რეჟიმში, ეს ბრძანება აბრუნებს სიგნალის მნიშვნელობას HIGH ან LOW შესასვლელიდან

ანალოგური სიგნალები

Arduino - ციფრული მოწყობილობაა, მაგრამ შეუძლია იმუშაოს ანალოგურ სიგნალებთანაც შემდეგი ორი ბრძანების დახმარებით:

analogWrite (pin, value) ;
 Arduino-ს ზოგიერთი პორტი მხარს უჭერს განვიმპულსურ მოდულაციას (3,5,6,9,10,11). ამ რეჟიმში რეზულტირებული მახვ პორტში დამოკიდებულ იქნება ლოგიკური ერთებისა და ნულების ხალხსზე, რომლებიც გაიგზავნება პორტში მალზედ დიდი სიჩქარით და შეიძლება იცვლებოდეს 0-დან (0ვ) 255-მდე (+5ვ)

analogRead (pin) ;
 თუ ანალოგური პორტი INPUT რეჟიმშია აწყობილი, მაშინ შესაძლებელია მასზედ არსებული მახვის გაზომვა. იგი შეიძლება იყოს 0-დან (0ვ) ... 1023 - მდე (+5ვ)

ელექტრონიკის საფუძვლები

ამ კომპლექტთან მუშაობისათვის თქვენ არ დაგჭირდებათ ელექტრონიკაში მუშაობის გამოცდილება. ქვემოთ მოყვანილია ზოგიერთი ცნობები ელექტრონული კომპონენტების შესახებ, რაც შესაძლებლობას მოგცემთ თქვენ მარტივად შეძლოთ მათი იდენტიფიცირება და მათი მოქმედების პრინციპი კი, თუ თქვენთვის რაიმეა გაუგებარი, ანდა კომპონენტი არ მუშაობს როგორც სჭირთა, მაშინ მიაკითხეთ ჩვენი ხელშეწყობის ჯგუფს მისამართზე help@oomlout.com

კომპონენტების აღწერა

LED (შუქდიოდი)



რას აკეთებს:
ასხივებს შუქს თუ მასში გავატარებთ მცირე დენს. დენი მასში შეიძლება გად-
იოდეს მხოლოდ ერთი მიმართულებით
შეხედულება:
გავს პატარა ნათურას

გამომყვანების რაოდენობა:
(უფრო გრძელი გამომყვანი (ანოდი),
მოუერთდება დადებით პოტენციალს)
საყურადღებოა:
მუშაობს მხოლოდ სწორი ჩართვის
შენთხვევაში. ითხოვს რეზისტორს
დამატებითი ინფორმაცია:
<http://ar dx.org/LED>

დიოდი



რას აკეთებს:
ცალმხრივ გამტარი ვენტილის ელექტრ-
ონული ექვივალენტი. დენმა დიოდში
შეიძლება მხოლოდ ერთ მხარეს გაიაროს.
შეხედულება:
ზოგადად ცილინდრული ფორმისაა
გამომყვანებით მოპირდაპირე მხარეებზე.
(ერთერთ მხარეს დატანილი აქვს
პოლიარობის მარკირებული ზოლი)

გამომყვანების რაოდენობა:2
საყურადღებოა:
ატარებს დენს მხოლოდ ერთი მიმართუ-
ლებით. დენი გაივლის მაშინ, როდესაც
ზოლიან გვერდს მივართებთ დაბალ
პოტენციალზე ნოლო მეორეს მაღალზე
დამატებითი ინფორმაცია:
<http://ar dx.org/DIOD>

რეზისტორი



რას აკეთებს:
ზღუდავს წრედში გამავალი დენის
მაღას.
შეხედულება: ზოგადად ცილინდ-
რული ფორმის გამომყვანებით
ურთიერთსაწინააღმდეგო მხარეებზე.
ნომინალის მნიშვნელობა ფერადი
ზოლების საშუალებით (იხ. შემდეგში)

გამომყვანების რაოდენობა:2
საყურადღებოა:
ადვილიდ შეიძლება აკერიოთ ნომინალი.
მკაცრად შეამოწმეთ მნიშვნელობა.
დამატებითი ინფორმაცია:
<http://ar dx.org/RESI>

ტრანზისტორი



რას აკეთებს:
გამოიყენება სიგნალების კომუტაციის
ანდა გაძლიერებისათვის.
შეხედულება: მზადდება მრავალ-
გვაროვან კორპუსებში. დასახელება
ჩვეულებრივად დაიტანება კორპუსზე
მოცემულ კომპლექტში გამოიყენება
P2N2222AG).

გამომყვანების რაოდენობა:3
(ზაზა, კოლექტორი, ემიტერი)
საყურადღებოა:
არ აურიოთ ვონტაქტები. დენის შეზღუ-
დვისათვის გამოიყენება რეზისტორები.
დამატებითი ინფორმაცია:
<http://ar dx.org/TRAN>

ხერვამპრავი



რას აკეთებს:
ელექტრულ სიგნალებს გარდაქმნის
ლერმის მოზრუნების კუთხეში.
შეხედულება:
პლასტმასის ყუთი სამი
გამოსასვლელით და მეტალის
ლერმით კროსტეინთან ერთად

გამომყვანების რაოდენობა:3
დარწმუნდით სწორად არის
მიერთებული თუ არა (გასართი
გასალების გარეშე).
საყურადღებოა:
დამატებითი ინფორმაცია:
<http://ar dx.org/SERV>

კოლექტორული -- მრავი



რას აკეთებს:
ზრუნავს. მასში ელექტრული
დენის გატარების დროს.
შეხედულება:
იგი მარტივი კონსტრუქციისაა,
ცილინდრული ფორმის, შუაში
ლერმით და ჩვეულებრივი მოტორია

გამომყვანების რაოდენობა:2
საყურადღებოა:
გამოიყენება შესაბამისი სიმძლავრის
რელე ან ტრანზისტორი მრავის
მიერთებისათვის
დამატებითი ინფორმაცია:
<http://ar dx.org/MOTO>

კომპონენტების აღწერა (გაგრძელება)

პიეზოელემენტი



რას აკეთებს:
დენის იმპულსი გარდაიქმნება ტკაცანში. იმპულსების თანმიმდევრობა კი გარდაიქმნება მუსიკალურ ტონად.
შეხედულება: ამ კომპლექტში ის გამოიყურება როგორც შავი ზალონი. ზოგჯერ შეიძლება გამოიყურებოდეს როგორც მოოქროვილი დისკო.

გამომყვანების რაოდენობა: 2
საყურადღებოა: შეუძლებელია არასწორად ჩართვა
დამატებითი ინფორმაცია:
<http://ardx.org/PIEZ>

IC (ინტეგრალური მიკროსქემა)



რას აკეთებს:
თავისთავში შეიცავს ნებისმიერი სირთულის ელექტრონიკას.
შეხედულება:
კომპონენტის სახელწოდება უზრალოდ დატანილია კორპუსზე (ზშირად მის წასაკითხად საჭიროა გამადიდებელი შუშა).

გამომყვანების რაოდენობა:
ორიდან რამდენიმე ასეულამდე. ამ კომპლექტში TMP36-ს აქვს 3 გამომყვანი და 74HC595-ს კი 16 გამოსასვლელი.
საყურადღებოა: არ აურიოთ მიკროსქემის ორიენტაცია
დამატებითი ინფორმაცია:
<http://ardx.org/ICIC>

დილაკი



რას აკეთებს:
ვრავს კონტაქტს დილაკზე თითის დაჭერის დროს.
შეხედულება:
მცირე ზომის ვეადრაკტი გამომყვანებით ქვევიდან და დილაკით ზევიდან

გამომყვანების რაოდენობა: 4
საყურადღებოა: პრაქტიკულად ვეადრატულია, შეიძლება დაყენდეს 90 გრადუსით მობრუნებით
დამატებითი ინფორმაცია:
<http://ardx.org/BUTT>

პოტენციომეტრი



რას აკეთებს:
რეზისტორი ნომინალით, რომლის სიდიდეც დამოკიდებულია ღერძის მობრუნების ვლთხეზე.
შეხედულება:
გამოდის სხვადასხვა კორპუსებში.

გამომყვანების რაოდენობა: 3
საყურადღებოა: შეიძლება ჰქონდეს ხაზოვანი ანდა ლოკარიტმული სკალა
დამატებითი ინფორმაცია:
<http://ardx.org/POTE>

ფოტორეზისტორი



რას აკეთებს:
რეზისტორი ნომინალით, რომლის სიდიდეც დამოკიდებულია მასზედ დაცემული შუქის ინტენსივობისაგან.
შეხედულება:
ჩვეულებრივად გამოიყურება როგორც დისკო გამჭირვალე დაფარვით და ზიგზაგული გამტარით მის ქვეშ.

გამომყვანების რაოდენობა: 2
საყურადღებოა: სასარგებლო სივანლის მიღებისათვის, აუცილებელია ფოტორეზისტორი გამოყენებულ იქნას როგორც ძაბვის გამყოფის ნაწილი.
დამატებითი ინფორმაცია:
<http://ardx.org/PHOT>

რეზისტორების ფერადი კოდირება

მაკალიტები:
მწვანე-ლურჯი-ყავისფერი = 560 ომი,
წითელი-წითელი-წითელი = 2 200 ომი,
ყავისფერი-შავი-ნარინჯის ფერი = 10 კომი.

მირველი ციფრი
მეორე ციფრი
წილების რაოდენობა
ნომინალის სიზუსტე

0 - შავი	5 - მწვანე	20% - უზოლო
1 - ყავის ფ.	6 - ლურჯი	10% - ვერცხ. ფ.
2 - წითელი	7 - იისფერი	5% - ოქროს ფ.
3 - ნარინჯ. ფ.	8 - რუხი	
4 - ყვითელი	9 - თეთრი	

გამომყვანების წაჭრა

ზოგიერთ კომპონენტს ან კომპლექტში აქვს ძალზედ გრძელი გამომყვანები. მათი გამოყენების მოხერხებულობისათვის შეიძლება გაკეთდეს შემდეგი:

შუქდიოდები:
წაჭრით გრძელი გამომყვანი 10 88-მდე, მოკლე 7 მდე.

რეზისტორები:
მონარეთ გამომყვანები ქვემოთ 90° და წაჭრით 6 მდე.

დანარჩენი კომპონენტები:
დანარჩენ კომპონენტებს შეგიძლიათ შეუმოკლოთ ფეხები თქვენი შეხედულებისამებრ.

CIRC-01

∴ ვიწყებთ მუშაობას∴
∴(მოციმციმე შუქდიოდი)∴



დავალების აღწერა:

შუქდიოდები ფართოდ გამოიყენება სხვადასხვა მოწყობილობებში, ამიტომ არის ისინი დამატებული ამ ნაკრებში. დავიწყეთ რაიმე მარტივიდან, მაგალითად ჩავრთოთ და გამოვრთოთ შუქდიოდი დაუსრულებელ ციკლში.

ავირჩიოთ დეტალები სიის მიხედვით, დავაყენოთ ყველა სამაკეტო დაფაზე. ჩატვირთეთ პროგრამა. ამისათვის მიუერთეთ Arduino თავისუფალ USB პორტს. შემდეგ დააყენეთ პორტი პროგრამაში Tools>Serial Port> (პორტი, დანიშნული Arduino-სათვის). პროგრამის ჩატვირთვა ხდება მენიუსგან File>Upload to I/O Board (ctrl+U). ბოლოს კი, დატკბით შუქდიოდის მართვის შესაძლებლობით!

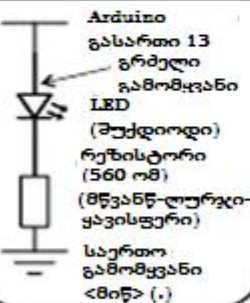
თუ თქვენ შეგქმნათ პრობლემები ჩატვირთვის დროს-დაუკავშირდით: <http://ardx.org/TRBL>

მოწყობილობა:

კომპონენტები:

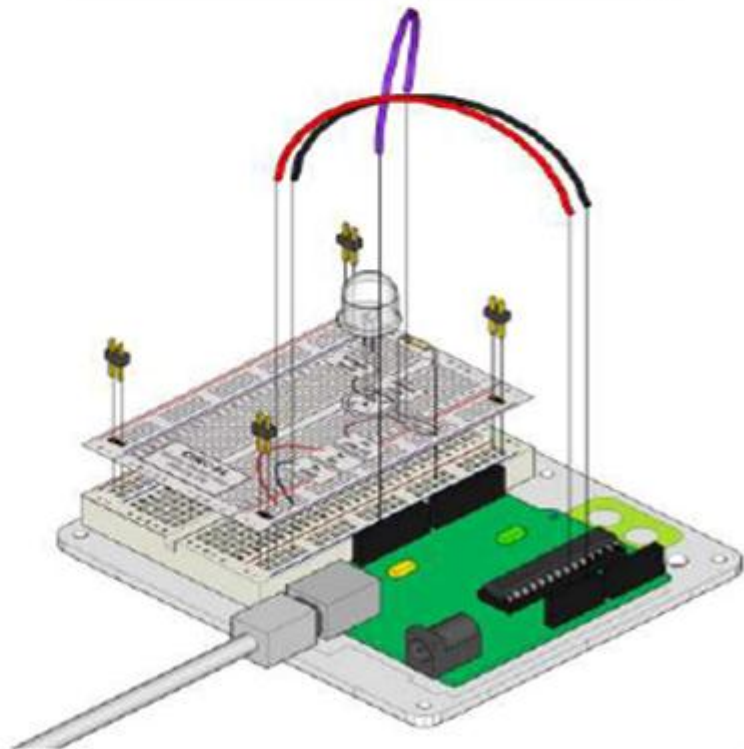
- | | | | |
|--|---|--|--|
|  დავალების ბარათი CIRC-01
x1 |  2 კონტაქტური გასართი
x4 |  შუქდიოდი 1088
x1 |  გამტარი |
|  რეზისტორი 560 ომ
მწვანე-ლურჯი-ყავისფერი
x1 | | | |

სქემა



ინტერნეტში:

•გადმოქაჩეთ, დავალების ბარათი <http://ardx.org/BBL.S01> .ნახეთ, ანაწყოების ვიდეო <http://ardx.org/VIDE01>



File>Examples>1.Basic>Blink

(ეს მაგალითი arduino.cc-სგანაა სადაც შეიძლება სხვების ნახვაც)

/* ციმციმი

- * ჩართავს შუქდიოდს 1 წამით, გამორთავს ასევე 1 წამით,
- * ფუნქციის გამოძახება ხდება ციკლურად (წრეზე)
- * Created 1 June 2005 By David Cuartielles
- * <http://arduino.cc/en/Tutorial/blink>
- * based on an original by H. Barragon for the wiring i/o board

```
*/
int ledpin = 13; // შუქდიოდი მიერთებულია მე-13 გამოსასვლელთან
//საწყისი დაყენებების ფუნქცია setup() გამოიძახება 1-ჯერ სვერის დასაწყისში
void setup() { // მე-13 კონტაქტის დაყენებით გამოყვანის რეჟიმში:
  pinMode(ledpin, OUTPUT); }
// ფუნქცია loop()-ის გამოძახება ხდება ციკლურად (წრეზე)
// მანამ, სანამ arduino მიერთებულია კვებასთან
void loop() {
  digitalWrite(ledpin, HIGH); //შუქდიოდის ჩართვა
  delay(1000); //დაყოვნება 1 წამით
  digitalWrite(ledpin, LOW); // შუქდიოდის გამორთვა
  delay(1000); //დაყოვნება 1 წამით
}
```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტები)

შუქდიოდი არ ინთება?

იგი შეიძლება პოლარობით არასწორად იყოს ჩართული. ამოიღეთ და ჩასვით შებრუნებით. სუ შეგეშინდებათ არასწორი ჩართვა მას არ დააზიანებს. შეამოწმეთ, ნამდვილად მე-13 პორტზე თუ არის იგი მიერთებული.

პროგრამა არ იტვირთება?

ეს შეიძლება ზოგჯერ მოხდეს. უფრო ხშირად მიზეზი შეიძლება იყოს არასწორად მითითებული მიმდევრობითი პორტი. პორტის შეცვლა შეიძლება მენიუსი tools>serial port>

მაინც არ მუშაობს?

აპარატურის უწყესრიგობა ეს სამწუხაროა, e-mail-ზე გამოგვიგზავნეთ ჩვენ პრობლემის აღწერა და ჩვენ დაგიკავშირდებით თქვენ რაც შეიძლება ჩქარა. help@oomlout.com

ვაუმჯობესებთ მოწყობილობას

მმართველი პორტის გამოცვლა:

შუქდიოდი მიერთებულია მე-13 პორტზე, მაგრამ ჩვენ შეგვიძლია გამოვიყენოთ Arduino-ს ნებისმიერი პორტი, გადაბურთელ მავთული ნებისმიერ პორტზე (0-დან 13-მდე), ანდა ანალოგურ პორტზე - (14-დან 19-მდე). შემდეგ შევცვალოთ ზრძანება:

```
int ledpin = 13; int ledpin = თქვენი ნომერი;
```

ციმციმის სიხშირის ცვლილება:

ჩართვისა და გამორთვის დროის შეცვლისათვის ვარედაქტირებთ ფუნქცია delay(x)-ის არგუმენტს. დაყოვნება ტოლია x = წამების რიცხვი * 1000.

```
მაგ, უფრო დაბალი სიხშირის ციმციმის მიღებისათვის პროგრამაში უნდა მიიღოს სახე:
digitalWrite(ledpin,HIGH);
delay(2000);
digitalWrite(ledpin,LOW);
delay(2000);
```

ნათების სიკაშკაშის მართვა:

მართვ ჩართვა გამორთვისთან ერთად შეიძლება აგრეთვე შუქდიოდის ნათების სიკაშკაშის მართვაც (უფრო დაწვრილებით ეს განმარტებული იქნება შემდეგ სავარჯიშოებში. მიაერთეთ შუქდიოდი მე-9 პორტს:

```
int ledpin = 13; შევცვალოთ int ledpin = 9;
შევცვალოთ კოდი პროცედურა loop()-ის ფიგურულ ფრჩხილებში შემდეგნაირად:
```

```
analogWrite(ledpin, new number); new number -
ნებისმიერი რიცხვია დიაპაზონიდან 0 ... 255.
0 შეესაბამება გამორთულ შუქდიოდს, 255 კი -
მაქსიმალურ სიკაშკაშეს .
```

სიკაშკაშის მდოვრე ცვლილება:

გახსენით მაგალითი:

File>Examples>3.Analog>Fading

ჩატვირთეთ ეს პროგრამა Arduino-ში და ნახეთ მიღებული ეფექტი .

კიდევ არის შეკითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC01>

CIRC-02

..სქემა 8 შუქდიოდზე..



დავალების აღწერა:

ჩვენ ვაიბუღეთ შუქდიოდი იციმციმოს, ახლა უკვე საჭიროა თამასის მაღლა აწევვა! მოდი მივაერთოთ ერთბაშად 8 შუქდიოდი. ჩვენ შეგვეძლება აგრეთვე შევქმნათ რამდენიმე სხვადასხვა შუქეფექტი. ეს მოწყობილობა კარგი საფუძველია შემდგომი ექსპერიმენტებისა და Arduino-ს ათვისებისათვის. შუქდიოდების მართვასთან ერთად განვიხილავთ პროგრამის ოპტიმიზაციის მეთოდებს.

for() loops - გამოიყენება, თუ თქვენთვის აუცილებელია პროგრამის ნაწილის მსრულებზე მრავალჯერ.

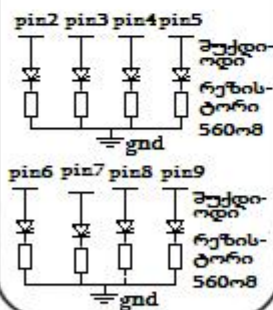
arrays [] - გამოიყენება ცვლადებთან მუშაობის გამარტივებისათვის (ფაქტიურად ეს ცვლადების ჯგუფია).

მოწყობილობა:

კომპონენტები

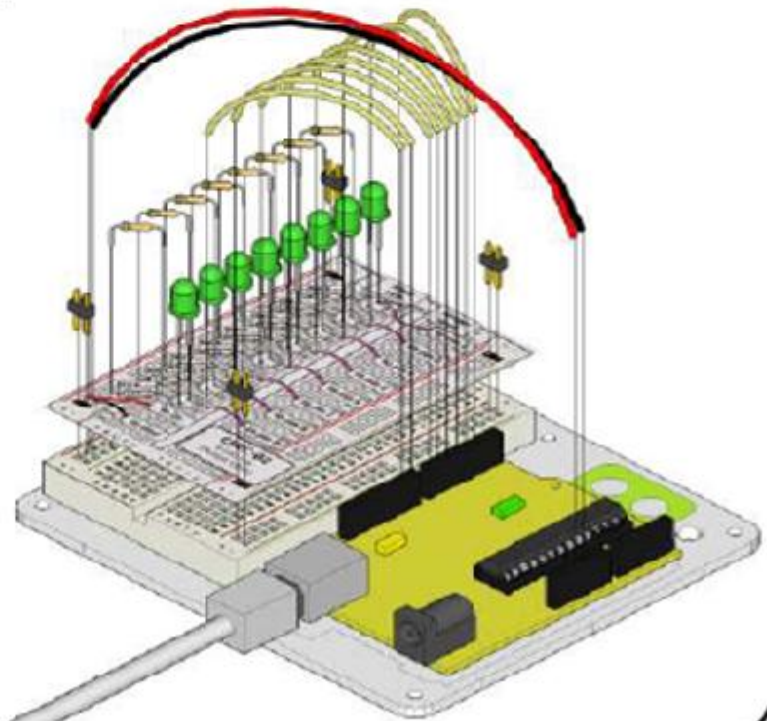
- | | | | |
|--|--|---|--|
|  დავალების ზარათი
CIRC - 02
x1 |  2 კონტაქტა
გასართი
x4 |  შუქდიოდი 5 მმ,
მწვანე
x8 |  გამტარი |
|  რეზისტორი 560 ომ
მწვანე - ლურჯი - ყავისფერი
x8 | | | |

სქემა



ინტერნეტში:

..ჩამოტვირთვა:
დავალების ზარათი
<http://ardx.org/BBS02>
..ნახეთ:
ანაწყოების ვიდეო
<http://ardx.org/VIDE02>



ის შეიძლება ჩამოკვირთოთ <http://ardx.org/CODE02>
გადმოაკვირეთ ტექსტი და ჩასვით სვეჩის ახალ ფანჯარაში

```
//შუქდიოდების ცვლადები
int ledPins[] = {2,3,4,5,6,7,8,9};
//სიმრავლე, რომელიც თვითოეულ
//შუქდიოდს აერთებს თვითოეულ
//პორტთან, მაგ. შუქდიოდს #0 მე-2
//პორტთან

void setup()
{
  for(int i = 0; i < 8; i++){
    //ციკლი მეორდება 8-ჯერ
    pinMode(ledPins[i],OUTPUT);
    //აყენებს პორტებს, რომლებზედაც შუქდიოდები
    //მიერთებული, გამოყვანის რეჟიმში
  }
}

void loop() //უსასრულო ციკლი
{
  oneAfterAnotherNoloop():
  //ჩართავს თვითოეულ შუქდიოდს ცალცალკე
  // და შემდეგ ასევე გამოართავს
  //oneAfterAnotherloop():
  //იგივეს აკეთებს, რასაც oneAfterAnotherNoloop
  //მაგრამ ბრძანება უფრო მოკლეა
  //oneOnAtATime():
  //inAndout():
}
/*
oneAfterAnotherNoloop() - აანთავს ერთ

```

```
შუქდიოდს.
დააყოვნებს delayTime
დროზე და აანთებს
შემდეგ შუქდიოდს.
გამორთავს
void oneAfterAnotherNoloop() {
  int delayTime = 100;
  //დროის დაყოვნება შუქდიოდის ჩართვებს
  //შორის მილიწამებში
  digitalWrite(ledpins[0], HIGH);
  // #0 შუქდიოდის ჩართვა მე-2 გამოსასვლელზე
  delay(delayTime); //დაყოვნება delayTime
  // დროში (მილიწამებში)
  ...
  digitalWrite(ledpins[7], HIGH);
  // #7 შუქდიოდის ჩართვა მე-9 გამოსასვლელზე
  delay(delayTime);
  //დაყოვნება delayTime დროში (მილიწამებში)
  //გამორთავს თვითოეული შუქდიოდის
  digitalWrite(ledpins[7], LOW); // #7 შუქდიოდის
  //გამორთავს
  delay(delayTime);
  //დაყოვნება delayTime დროში (მილიწამებში)
}

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

ზოგიერთი შუქდიოდი არ ანათებს
ადვილი შესაძლებელია შექნიშანი უკუღმა ჩართული. შეამოწმეთ პოლარობა ყველა შუქდიოდის, რომლებიც არ მუშაობენ

შუქდიოდები ინთება არასწორი თანმიმდევრობით
8 გამტარის მიერთებისას ადვილად შეიძლება წყვილების არეუდარება, დარწმუნდით, რომ პირველი შუქდიოდი შეერთებულია მე-2 პორტზე და ყოველი შემდეგი კი შემდგომ თანმიმდევრულ პორტზე.

დაიწყეთ თავიდან
ხშირად უფრო ადვილია ყველაფერი დაშალოთ და თავიდან ააწყოთ, ვიდრე ეძებოთ უწესრიგობანი

ვაუმჯობესებთ მოწყობილობას

ციკლების გამოყენება:
ფუნქცია loop-ში ჩაწერილია 4 ბრძანება. ბოლო სამი იწყება /* */ სიმბოლოთი. ამიტომ, კომპილიატორი აიგნორირებს ამ სტრიქონებს (განიზილავს მათ როგორც კომენტარიებს). ციკლების გამოყენებისთვის შეცვალეთ პროგრამის ტექსტი შემდეგნაირად:

```
//oneAfterAnotherNoloop();
oneAfterAnotherloop();
//oneOnAtATime();
//inAndout();
```

ჩატვირთეთ პროგრამა Arduino-ში და მიაქციეთ ყურადღება, რომ პროგრამა სრულდება ისევე, ე.ი. ეს ორი ფუნქცია ასრულებს ერთსადაიგივეს, მაგრამ სხვადასხვა პრინციპების გამოყენებით

მე-2 ფუნქცია იყენებს სტრუქტურას for). დამატებითი ეფექტები:

დაიდალეთ ამ ეფექტისაგან? გამოიყენეთ ორი სხვა ნიმუში. მოაცილეთ მორიგეობით კომენტარის ნიშნები (/*) ორი ბოლო სტრიქონისგან, ჩატვირთეთ Arduino-ს პროგრამა და დატკბით ახალი ეფექტებით შექმენით თქვენი საკუთარი ეფექტები: შეეცადეთ შეცვალოთ რაიმე პროგრამაში. მთელი იდეა შუქდიოდების ჩართვაშია ბრძანებით digitalWrite(pinNumber, High); და გამორთვაში ბრძანებით digitalWrite(pinNumber, Low); ნუ შეგეშინდებათ ექსპერიმენტების ჩატარებით, იმის მიუხედავად რასაც კი შეცვლით, არაფერი არ უნდა გაფუჭდეს.

კიდევ არის შეკითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC02>

CIRC-03

..იბრუნე მობოტრო,
 იბრუნე..
 ..ტრანზისტორი
 და მობოტრო:.



დავალების აღწერა:

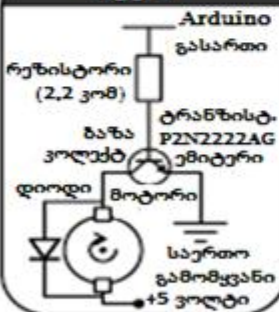
Arduino-ს პორტები იდეალურია მცირე მოხმარებული სიმძლავრის მქონე დეტალების სამართავად (მაგ. შუქდიოდები). მაგრამ, დიდი დატვირთების შემთხვევაში საჭიროა დამატებითი ელემენტის გამოყენება, მაგ. ტრანზისტორის. ტრანზისტორი ძალიან სასარგებლო ელემენტია, მისი დახმარებით შესაძლებელია დიდი დატვირთვების მართვა, თვითონ კი ითხოვს ძალზედ მცირე დენს. ტრანზისტორს აქვს სამი გამოსასვლელი. n-p-n ტიპის ტრანზისტორისათვის საჭიროა დატვირთვა მიერთებულ იქნას კოლექტორზე ხოლო "მიწა" ემიტერზე. თუ კი ბაზიდან ემიტერისაკენ გაივლის მცირე დენი (თუ კი მის ბაზას მივაერთებთ Arduino-ს HIGH სიგნალს, მაშინ ტრანზისტორი "გაიღება", დენი გაივლის ტრანზისტორს და მობოტრო დაიწყებს ბრუნვას. ჩვენს კომპლექტში გამოიყენება საერთო გამოყენების P2N2222AG ტრანზისტორები, რომელთა მაქსიმალური დასაშვები ძაბვა 40 ვოლტია, ხოლო მაქსიმალური დენი - 600 მა. ეს პარამეტრები საშუალებას გვაძლევს გამოვიყენოთ იგი ჩვენი მობოტროისათვის. უფრო დაწვრილებით ტრანზისტორის პარამეტრები მოყვანილია მისამართზე: <http://ardx.org/2222>.
 უკუ დენებისაგან დაცვის მიზნით გამოიყენება დიოდი 1N4001. ამის შესახებ უფრო დაწვრილებით: <http://ardx.org>

მოწყობილობა:

კომპონენტები

- | | | | |
|--------------------------------|----------------------------|---|------------------------------|
| დავალების ბარათი CIRC-01
x1 | 2 კონტაქტური გასართი
x4 | ტრანზისტორი P2N2222AG (TO92)
x1 | გამტარი დიოდი (1N4001)
x1 |
| კოლექტორული მრავი
x1 | | რეზისტორი 2.2 კომ
წითელი-წითელი-წითელი
x1 | |

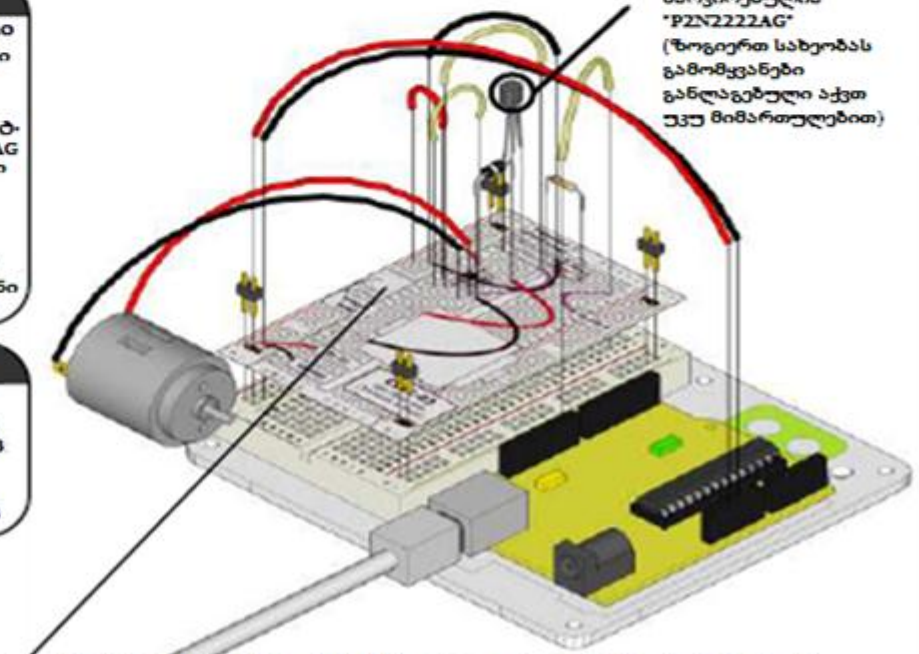
სქემა



ტრანზისტორი მარკირებულია "P2N2222AG" (ზოგიერთ სახეობას გამოყვანები განლაგებული აქვთ უკუ მიმართულებით)

ინტერნეტში:

..ჩამოტვირთეთ:
 დავალების ბარათი <http://ardx.org/BBSO3>
 .. ნახეთ:
 ანაწყობის ვიდეო <http://ardx.org/VIDEO3>



..შენიშვნა: თუ მობოტროს გაშვებისას Arduino დაევიდება ან გადაიტვირთება, მაშინ აუცილებელია დამატებითი კონდენსატორის დაყენება..

კოდი არ არის საჭირო ხელით აკრეფვა

CIRC-03

ის შეიძლება ჩამოტვირთოთ <http://ardx.org/CODE02>

გადმოაკომპირეთ ტექსტი და ჩასვით სვეჩის ახალ ფანჯარაში

```
int motorPin = 9; //მრავი მიერთებულია მე-9 გამოს.
void setup() //გაიშვება მხოლოდ ერთჯერადად
{
  pinMode(motorPin, OUTPUT);
}
void loop() // სრულდება უსასრულოდ
{
  motorOnThenoff();
  //motorOnThenoffWithSpeed();
  //motorAcceleration();
}
/* motorOnThenoff(); - ჩართავს მრავს, შემდეგ კი
გამორთავს (ზიანქვით ყურადღება, რომ კოდი
იდენტურია როგორც მოსიარულე შუქდიოდისას)*/
void motorOnThenoff() {
  int onTime = 2500; //ამ დროით ჩართულია
  int offTime = 1000; //ამ დროით გამორთულია
  digitalWrite(motorPin, HIGH); //მრავის ჩართვა
  delay(onTime); //დაყოვნება onTime მილიწამით
  digitalWrite(motorPin, LOW); //მრავის გამორთვა
  delay(offTime); //დაყოვნება offTime მილიწამით
}
```

```
void motorTheOffwithSpeed() {
  int onSpeed = 200; // რიცხვი 0-დან 255-მდე (მაქ. სიჩქ.)
  int onTime = 2500;
  int offSpeed = 50; // რიცხვი 0-დან 255-მდე (მაქს. სიჩქ.)
  int offTime = 1000;
  analogWrite(motorPin, onSpeed);
  // მრავის ჩართვა
  delay(onTime); // დაყოვნება onTime მილიწამით
  analogWrite(motorPin, offSpeed);
  //მრავის გამორთვა
  delay(offTime); // დაყოვნება onTime მილიწამით }
void motorAcceleration() {
  int delayTime = 50; // დაყოვნება სიჩქარებს შორის
  for(int i = 0; i<256; i++) { // ცვლის სიჩქარეს 0...255-ში
    analogWrite(motorPin, i); // აყენებს ახალ სიჩქარეს
    delay(delayTime); //დაყოვნება delayTim მილიწამით
  }
  for(int i = 255; i >= 0; i--) { ნიშნავს სიჩქარეს 0 ... 155-ში
    analogWrite(motorPin, i); აყენებს ახალ სიჩქარეს
    delay(delayTime); // დაყოვნება delayTime მილიწამით
  }
}
```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

მრავი არ ბრუნავს?
თუ თქვენ იყენებთ სხვა ტრანზისტორს დარწმუნდით, რომ მისი გამომყვანები ემთხვება P2N222A-ს. ზოგიერთს აქვთ გამომყვანების სხვა განლაგება.

ახლა კიდევ არ მუშაობს?
თუ თქვენ იყენებთ სხვა მრავს, მაშინ დარწმუნდით, რომ მას შეუძლია მუშაობა 5 ვ-ზე და არ იხმარს ძალზედ ბევრ სიმძლავრეს

და კიდევ არ მუშაობს?
ზოგჯერ Arduino თვითნებურად გაითიშება კომპიუტერისგან. შეეცადეთ გამოთიშოთ და შემდეგ ხელმეორეთ ჩართოთ Arduino-ს USB კაბელი.

ვაუმჯობესებთ მოწყობილობას

სიჩქარის მართვა ადრე ჩვენ განვიხილეთ შუქდიოდის სივარჯის მართვას Arduino-ს საშუალებით. იგივე პრინციპი შეიძლება გამოყენებულ იქნას მრავის ბრუნვის სიჩქარის მართვისთვის. Arduino აფორმირებს გიმ (განვი-იმპულსური მოდულიაცია) სიგნალებს, გამოსასვლელს მაღალი სიხშირით მიეწოდება ლოგიკური ნულებისა და ერთების თანმიმდევრობები, ნოლებისა და ერთების რაოდენობათა ბალანსი განსაზღვრავს რეზულტირებულ ძაბვას. მაგალითად, 2,5 ვოლტის ძაბვის ფორმირებისათვის აუცილებელია, რომ დროის ერთეულში ნულებისა და ერთების რაოდენობა იყოს ერთმანეთის ტოლი. loop()-ის ფუნქციაში შევვალეთ პროგრამის ტექსტი შემდეგნაირად:

```
//motorOnThenOff();
motorOnThenOffWithSpeed();
//motorAcceleration();
```

ჩატვირთეთ პროგრამა Arduino-ში. თქვენ შეგიძლიათ ცვლით სიჩქარე onSpeed და offSpeed ცვლადების რედაქტირებით.

არქარება და შენელება
რატომ შევჩერდეთ მხოლოდ ორ ფიქსირებულ სიჩქარეზე? ამ ტექნოლოგიის გამოყენებით შეგიძლია ავარჯიროთ და შევანელოთ მრავი. ამის მისაღწევად loop() ფუნქციაში უნდა შევვალთ ტექსტი შემდეგნაირად:

```
//motorOnTheOff;
//motorOnTheOffWithSpeed();
motorAcceleration();
```

ჩატვირთეთ პროგრამა და თქვენ ნახავთ, მრავი როგორ მდორედ გაქანდება მაქსიმალურ სიჩქარემდე და შემდეგ მდორედ დამუხრუჭდება. თუ თქვენ გჭირდებათ გაქანების სიჩქარის შეცვლა, მაშინ შევვალთ delayTime ცვლადი (უფრო დიდ მნიშვნელობას შეესაბამება უფრო ნელი გაქანება).

კიდევ არის შეკითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC03>

CIRC-04

ერთეულოვანი სერვომძრავი



დავალების აღწერა :

მოტორის დაბრუნება საკმაოდ საინტერესოა, მაგრამ როდესაც ჩვენ ვაკეთებთ პროექტს, სადაც აუცილებელია მოძრაობის კონტროლი, მაშინ გვინდა უფრო რაღაც მეტის გაკეთება. განვიხილოთ სერვომძრავის მართვის მაგალითი. ეს ამძრავები მასიურად იწარმოება, ადვილად ხელმისაწვდომია, მისი ფასი იცვლება ერთიორი დოლარიდან რამდენიმე ასეულამდე. სერვომძრავის შიგნით ჩაშენებულია მცირე ზომის რედუქტორი (სიმძლავრის გასაზრდელად) და ელექტრონიკა (მართვის გამართივებისათვის). სტანდარტული სერვომძრავის პოზიციონირება ხდება 0-დან 180 გრადუსამდე. პოზიციის დავალება ხდება მმართველი იმპულსის ხანგრძლივობით 1,25 მწმ-დან (0 გრადუსი) 1,75 მწმ-მდე 180 გრადუსისათვის (1,5 მწმ - 90 გრადუსი). დროითი პარამეტრები შეიძლება განსხვავებულ იყოს სხვადასხვა მწარმოებლებისათვის. თუ კი გავაგზავნით იმპულსებს ყოველ 25 ... 50 მლწ-ში, მაშინ ამძრავი მდორედ დაიწყებს ბრუნვას. Arduino-ს ერთერთი უპირატესობა ისაა, რომ მას აქვს გამოსაყენებლად ქვეპროგრამების გამოსაყენებლად გამზადებული ბიბლიოთეკა, რომლითაც შეიძლება ორი სერვომძრავის ერთდროული მართვა (რომლებიც მიერთებულ უნდა იყოს მე-9 და მე-10 პორტებზე).

მოწყობილობა :

კომპონენტები:

- 

დავალების ზარათი
CIRC-01
x1
- 

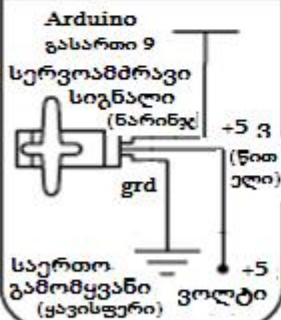
2-კონტაქტიანი
გასართი
x4
- 

3-კონტაქტიანი
გასართი
x1
- 

გამტარი
x1
- 

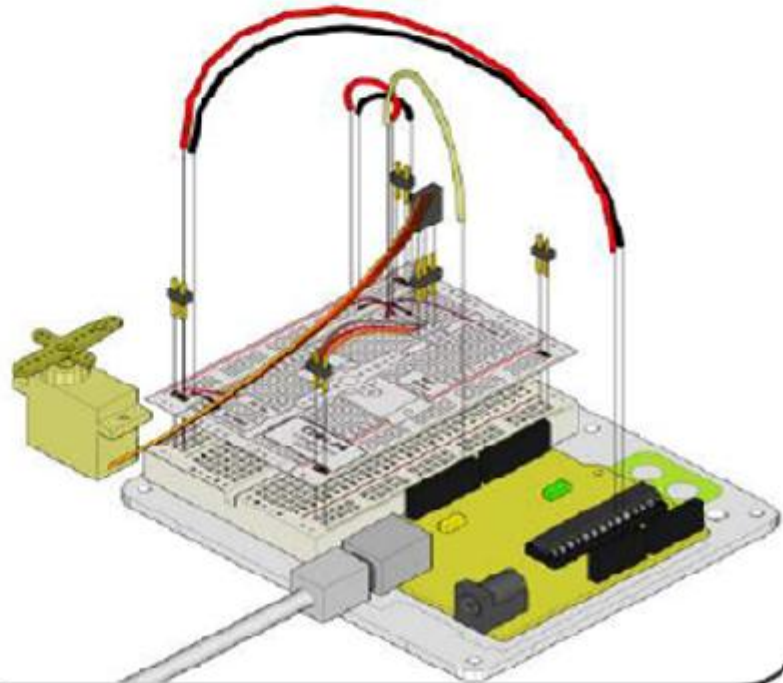
სერვომძრავი
x1

სქემა



ინტერნეტში:

• ჩანმოტირეთ: •
დავალების ზარათი
<http://arduino.org/BLSO4>
• ნახეთ: •
ანაწყობის ვიდეო
<http://arduino.org/VIDEO4>



კოდი არ არის საჭირო ხელით აკრეფა, იგი იმყოფება მისამართზე:

```
File > Examples > Servo > Sweep <http://barraganstudio.com>).
(ეს მაგალითი დევს arduino.cc საიტზე, ზევრ სხვა კარგ იდეებთან ერთად)
// სერვოამძრავის მობრუნება (ბარაგანის მიერ)
#include <Servo.h>
Servo myservo; // ობიექტის "სერვო"-ს შექმნა სერვოამძრავის მართვისათვის
int pos = 0; // ცვლადი სერვო პოზიციის შენახვისათვის
void setup() {
  myservo.attach(9); // დაუნიშნავს ობიექტს სერვოს მე-9 გამოსას, როგორც მმართველს
}
void loop() {
  for (pos = 0; pos < 180; pos += 1) // მობრუნდება 0-დან 180 გრადუსამდე 1 გრადუსის ბიჯით
    myservo.write(pos); // ბრძანება სერვოამძრავის მიმართ
    // გადასვლა პოზიციაში ცვლადით 'pos'
    delay(15); // დაყოვნება 15 მლწ-ით სერვოსათვის პოზიციასთან მიღწევისთვის
  }
  for (pos = 180; pos >= 1; pos -= 1) // მობრუნება 180-დან 0 გრადუსამდე
  {
    myservo.write(pos); // ბრძანება სერვოამძრავის მიმართ
    delay(15); // გადასვლა პოზიციაში ცვლადით 'pos'
  }
  // დაყოვნება 15 მლწ-ით სერვოსათვის პოზიციასთან მიღწევისთვის
}
```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

სერვოამძრავი არ ბრუნავს?
ფერადი მარკირებითაგ კი მისი მავთულები ადვილი ასარევაა. შეიძლება მიზეზი ამაშია.

მაინც არ ბრუნავს?
არ დაგავიწყდეთ კვება. წითელი და ყავისფერი გამტარები უნდა მიერთებულ იქნას +5ვ და "მიწას" შესაბამისად

თუ სერვოამძრავი მოწრობს რტყმევეებით თუ სერვოამძრავი მოძრაობს და Arduino-ს შუქდიოდი გიმციმებს, მაშინ შესაძლოა კვების წყაროს სიმძლავრე არაა საკმარისი. შეეცადეთ გამოიყენოთ კვება არა USB-დან არამედ გარე წყაროსგან.

ვაუმჯობესებთ მოწყობილობას

სავარჯიშო პოტენციომეტრით: ჩვენ ჯერ ვიდეო არ ჩაგვიტარებია ექსპერიმენტები შესასვლელზე, მაგრამ მოდით ვცადოთ. ჩატვირთოთ მაგალითი File > Servo > Knob. ამ მაგალითში სერვოამძრავის მართვისათვის გამოიყენება პოტენციომეტრი (CIRC08). განმარტებისთვის ნახეთ საიტი: <http://ardx.org/KNOB> პირდაპირი მართვა: სერვოამძრავის მართვა მარტივია ქვეპროგრამების ბიბლიოთეკის დახმარებით, მაგრამ შეიძლება მისი უშუალო მართვა თუ გვავაზავნით საჭირო ხანგრძლივობის იმპულსს არდუინოს ნებისმიერ პორტზე. პროგრამის ნიმუში:

```
int servoPin = 9;
void setup() {
  pinMode(servoPin, OUTPUT);
}
```

```
void loop() {
  int pulseTime = 2200;
  // დაყოვნების ხანგრძლიობა მივიწინაშეში
  // 900 0 გრადუსისთვის და 1500 90 გრადუსს,
  // 2100 180 გრადუსისთვის
  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulseTime);
  digitalWrite(servoPin, LOW);
  delay(25);
}
```

საინტერესო იდეები: სერვოამძრავი შეიძლება გამოყენებულ იქნას მრავალ საინტერესო პროექტში. მოვიყვანო რამდენიმე მაგალითს: საშობაო მთვლელი <http://ardx.org/XMAS> რობოტის მანიპულიატორი (ღია პროექტი სერვოამძრავებისა და Arduino-ს გამოყენებით) <http://ardx.org/EARM>

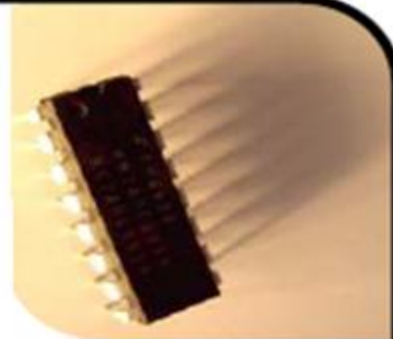
არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიყილოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC04>

CIRC-05

..კიდევ 8 შუქდიოდი (წანაცვლების რეგისტრი 74HC595):..



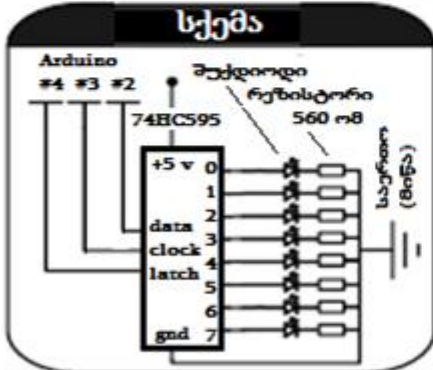
დავალების აღწერა:

მოვიდა უკვე დრო გავეცნოთ მიკროსქემებს. მიკროსქემის გარე ხედი შეიძლება ძალზედ დამაბნეველი იყოს. მაგ. კონტროლერის მიკროსქემა Arduino-ს დაფაზე და წანაცვლების რეგისტრის მიკროსქემა ძალზედ ჰგავნან ერთმანეთს, თუმც თავისი არსით მნიშვნელოვნად განსხვავდება ერთმანეთისაგან. ATmega მიკროკონტროლერის ფასი შეადგენს რამდენიმე დოლარს, მაშინ, როცა წანაცვლების რეგისტრის-რამდენიმე ცენტს. წანაცვლების რეგისტრის ათვისება, ეს კარგი დასაწყისია, და თუ თქვენ გაბარკვევთ, თუ როგორ უნდა მართოთ ის და როგორ იმუშაოთ დოკუმენტაციასთან (<http://arduino.org/74HC595>), თქვენ უკვე არ შეგეშინდებათ მიკროსქემების სამყაროსი, წანაცვლების რეგისტრი (მას ასევე უწოდებენ მიმდევრობითი ინტერფეისის გარდამქმნელს პარალელურში) თქვენ მოგვცემ 8 დამატებით ციფრულ გამოსასვლელს (შუქდიოდების, ანდა სხვა რამის მართვისათვის) Arduino-ს მხოლოდ სამი პორტის გამოყენებით. ეს რეგისტრები შეიძლება შეაერთოთ მიმდევრობით, რაც შესაძლებლობას მოგვცემ მიიღოთ პრაქტიკულად უსასრულო რაოდენობის გამოსასვლელი, იგივე სამი პორტის გამოყენებით. წანაცვლების რეგისტრის გამოყენებისათვის აუცილებელია მასში მნიშვნელოვან ჩაიწეროს მიმდევრობითი ინტერფეისის გამოყენებით და შემდეგ გაიღეს იგი გამოსასვლელზე პარალელური ინტერფეისით. მიმდევრობითი ინტერფეისი წარმოადგენს ორ შესასვლელს: ტაქტების შესასვლელს და მონაცემების შესასვლელს, მონაცემთა ბაიტის გადაცემისთვის ან რეგისტრის შესასვლელზე აუცილებელია მორიგეობით დაყენდეს ბაიტის ბიტების შესაბამისი დონეები და მიეწოდოს იმპულსები ტაქტურ შესასვლელზე. მას შემდეგ, რაც მთელი ბაიტი გაივლის წანაცვლების რეგისტრში, მიეწოდება ზრმანება ან ბაიტის გაცემისათვის პარალელური ინტერფეისით. უფრო დაწერილებითი ინფორმაცია შეიძლება ინახოს მისამართზე: <http://arduino.org/SHIFT>.

მოწყობილობა:

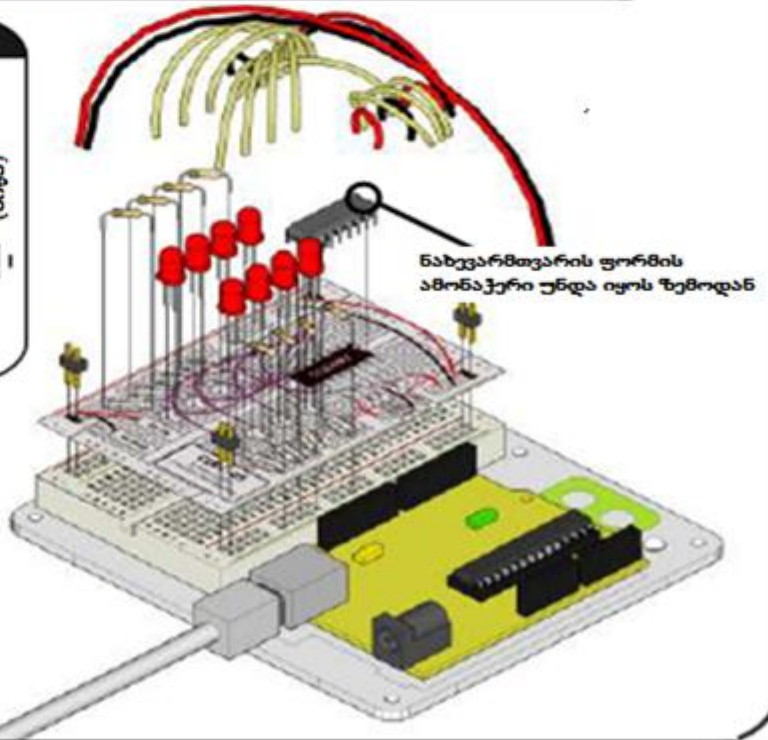
კომპონენტები:

-  დავალების ბარათი CIRC-01 x1
-  წითელი შუქდიოდი x8
-  2-კონტაქტიანი გასართი რეზისტორი 560 ომ x8
-  წანაცვლების რეგისტრის მიკროსქემა 74HC595 x4
-  გამტარი



ინტერნეტში:

..ჩამოტვირთეთ..
 დავალების ბარათი <http://arduino.org/BBS05>
 .. ნახეთ:..
 ანაწყობის ვიდეო <http://arduino.org/VIDE05>



ნახევარმთვარის ფორმის ამონაჭერი უნდა იყოს ზემოდან

კოდი არ არის საჭირო ხელით აკრეფა.

CIRC-05

იგი შეიძლება ჩამოიტვირთოს <http://ardx.org/CODE05>
(გადმოაკოპირეთ ტექსტი და ჩასვით Arduino sketch-ის ახალ ფანჯარაში)

```

გასართების განსაზღვრა
//74HC595 იყენებს SPI პროტოკოლს
//რომელსაც აქვს 3 კონტაქტი:
int data = 2;
int clock = 3;
int latch = 4;
void setup();
{
  pinMode(data, OUTPUT);
  pinMode(clock, OUTPUT);
  pinMode(latch, OUTPUT); }
void loop()
{
  int delayTime = 100;
  //დაყოვნება შუქდიოდების განახლებებს შორის
  for(int i = 0; i<256; i++) {
    updateLEDS(i); }
  }
/* updateLEDS() - აგზავნის შუქდიოდების
მდგომარეობებს ცვლადის მნიშვნელობის სახით

```

```

74HC595-ის მიმდევრობებზე
*/

void updateLEDS(int value) {

digitalWrite(latch, LOW);
// ჩაქერის სიგნალი დაყენებულია LOW-ზე

shiftOut(data, clock, MSBFIRST, value);
// გადასცევს 8 ბიტს წანაცვლების რეგისტრში

digitalWrite(latch, HIGH);
// ალუბს ჩამყერს - გამოსცევს ინფორმაციას
}

----- სრული პროგრამის ელექტრონულ ვერსიაში -----

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

Arduino-კვების ინდიკატორი არ ანათებს
ალბათ თქვენ წანაცვლების რეგისტრის მიკროსქემა ჩართეთ უკუღმა. თუ კი თქვენ სწააყად გამორთავთ კვებას და გადაადგილებთ მიკროსქემის კვებას მაშინ არაფერი არ უნდა გაფუჭდეს.

მუშაობს არასწორად?
ალბათ თქვენ ავერიათ რაიმე მავთულბი. ყურადღებით შეამოწმეთ შეერთებები.

მაინც არაფერი არ გამოვიდა?
მოგწერეთ ჩვენ წერილი პრობლემის აღწერით. შევცვდებით დაგეხმაროდ ისე სწრაფად როგორც შევძლებთ help@oomlout.com

ვაუმჯობესებთ მოწყობილობას

ვართულებთ ამოცანას:
Arduino-ს შეუძლია რთული საქმეების შესრულება შედარებით ადვილად. მონაცემთა გაცემა მიმდევრობითი ინტერფეისის გავლით - ეს ერთერთი ასეთი ამოცანაა. მიუხედავად ამისა ეს ამოცანა შესაძლოა გადაწყვეტილ იქნას უფრო რთული გზით. შეეცადათ შევალთ ტექსტი შემდეგნაირად:
updateLEDS(i); შევვალთ updateLEDSLong(i);
ჩატვირთეთ პროგრამა კონტროლერში და თქვენ დაინახავთ, რომ არაფერი არ შევლიდა. შეხედეთ ფუნქციის updateLEDSLong(i) ტექსტს და თქვენ ნახავთ, რომ მონაცემთა გადაცემა მიმდინარეობს თითო ბიტით (დანჭერილებით <http://ardx.org/SPI> -ზე).
შუქდიოდის დამოუკიდებელი მართვა:
ახლა შევცვალოთ ვმართოთ შუქდიოდები, ისევე როგორც დავალბაში CIRC02. ყველა რვა შუქდიოდის მდგომარეობა შეინახება ერთი ბიტის დახმარებით.

Arduino-ს შეუძლია ოპერირება ბაიტების თვითოეულ ბიტებთან. (იხ. მისამართზე: <http://ardx.org/BITW>). ჩაუტარეთ ტექსტს რედაქტირება ასეთნაირად:
int delayTime = 100;
//ციკლებს შორის დროის დაყოვნება მ.წამებში
for(int i = 0; i < 8; i++){
 changeLED(i,ON);
 delay(delayTime);
}for(int i = 0; i < 8; i++){
 changeLED(i,OFF);
 delay(delayTime); }
ჩატვირთეთ პროგრამა Arduino-ში. შუქდიოდები უნდა ინთებოდნენ და ქვრებოდნენ ერთმანეთის მიყოლებით.
დამატებითი ეფექტები:
თქვენ შეგიძლიათ გადმოაკოპიროთ ეფექტების ქვეპროგრამა COERC02 პროგრამიდან და შევალთ ბრძანება digitalWrite(led,state) - changeLED(led,state)-ზე.

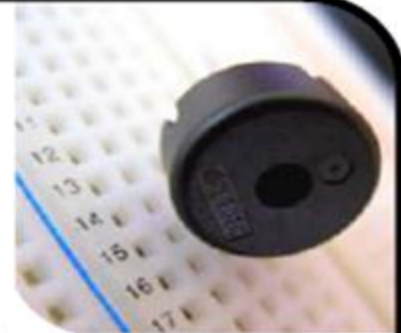
არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC04>

CIRC-06

.:მუსიკა (პიეზოელემენტი):.



დავალების აღწერა:

აქამდე ჩვენ ვმართავდით შუქს, მოძრაობას და ელექტრონებს. ახლა ვცადათ ვმართოთ ხმა. ხმა, ანალოგური მოვლენაა. შესაძლებელია თუ არა შევქმნათ ხმები ციფრული Arduino-ს მეშვეობით? შესაძლებელია, გამომდინარე მიკროკონტროლერის შედარებით მაღალი მუშაობის სიჩქარისაგან. პიეზოელემენტი ჩხაკუნებს ყოველთვის, როცა მასმედ მიეწოდება ელექტრული იმპულსი. თუ კი ამ იმპულსებს გავაგზავნით გარკვეული სიხშირით (მაგალითად 440 -ჯერ წამში ნოტა ლიას აღწარმოებისათვის), მაშინ ეს ჩხაკუნები გარდაიქმნებიან მუსიკალურ ტონად. მოდით ჩავატაროთ ექსპერიმენტები და ვაიძულოთ Arduino დაუკრას რაიმე მელოდია.

მოწყობილობა:

კომპონენტები:

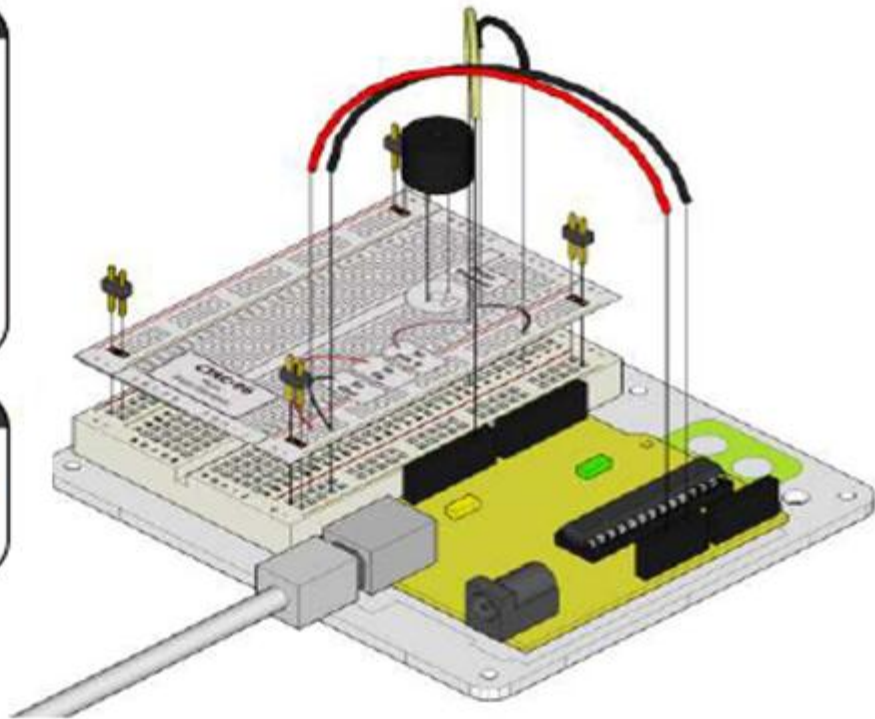
- 
 დავალების ხარატი
CIRC-01
x1
- 
 2-კონტაქტიანი
გასართი
x4
- 
 პიეზოელემენტი
x1
- 
 გამტარი

სქემა



ინტერნეტში:

.:ჩამოტვირთეთ.:
 დავალების ხარატი
<http://arduino.org/BBS05>
 .: ნახეთ.:
 ანაწყოების ვიდეო
<http://arduino.org/VIDE06>



მისი ჩამოტვირთვა შეიძლება <http://ardx.org/CODE06> -გან
(გადმოაკვირეთ ტექსტი და ჩასვით Sketch-ის ახალ ფანჯარაში)

/* მელოდია

```

ამ მაგალითში ჩვენ დაეტყობათ მელოდიაზე პიუზოლეუმების
გაითქვით. ეს მიუხედავად იქნება რომელიც გით-
ხეგნაღი, რის შედეგადაც გენერირდება მუსიკალური ტონი.
* საჭირო დაყოვნებების განგარამება ხდება შემდეგი
* ფორმულის გამოყენებით:
timeHigh = period / 2 - 1 / (2 * toneFrequency)
* დაყოვნებები განისაზღვრება შემდეგი ცხრილით:
* ნოტა ხობნიტე პერიოდი timeHigh
* e 261 Hz 3830 1915
* d 294 Hz 3400 1700
* e 329 Hz 3038 1519
* f 349 Hz 2864 1432
* g 392 Hz 2550 1275
* a 440 Hz 2272 1136
* b 493 Hz 2028 1014
* c 523 Hz 1912 956
*
http://www.arduino.cc/en/Tutorial/Melody
*/

```

```

int speakerPin = 9;
int length = 15; // ნოტების რაოდენობა
char notes[] = "cegaagffeeddc"; // მრავალი არის პაუნა
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;

```

```

void playTone(int tone, int duration) {
  for (long i=0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone); } }
void PlayNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
  for (int i=0; i<8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration); } } }
void setup() {
  pinMode(speakerPin, OUTPUT); }
void loop() {
  for (int i=0; i < length; i++) {
    if (notes[i] == ' ') {
      delay(beats[i] * tempo); // პაუნა
    } else {
      playNote(notes[i], beats[i] * tempo);
    } // პაუნა ნოტებს შორის
    delay(tempo / 2); } }

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

არ არის ხმა?
პიუზოლეუმენტის ფორმისა და ზომებისგან გამომდინარე ადვილი შესაძლებელია სწორი კონტაქტების არეგდარევა. შეამოწმეთ პიუზოლეუმენტის განლაგება

გიშლით ხელს ფიქრში მელოდის ხმა?
უბრალოდ, გამორთეთ პიუზოლეუმენტი სამაკეტო დაფისაგან, დაასრულეთ ფიქრი, და შემდეგ მიუერთეთ იგი ისევ უკან

მოგებურდათ ეს მელოდია?
პროგრამა მზადაა, თქვენ მხოლოდ გჭირდებათ შეცვალოთ ნოტები სხვებზე

გაუმჯობესებთ მოწყობილობას

```

სიჩქარის ცვლა:
მელოდის აღწარმოების სიჩქარის ცვლისათვის
საჭიროა სტრიქონის რედაქტირება
int tempo = 300; -->
int tempo = (ახალი მნიშვნელობა)
მელოდის ტემპის შენელებისათვის საჭიროა ამ
რიცხვის გაზრდა, ხოლო გაზრდისთვის - შემცირება.
ნოტების აწყობა:
თუ თქვენ გგონიათ, რომელიმე ნოტა ეღერს ყალბად,
შეგიძლიათ დააკორექტიროთ ისინი. ამისათვის
საჭიროა შესაბამისი ნოტის რედაქტირება მასივში
tones[]
char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136,
1014, 956 };

```

```

საკუთარი მელოდის დაწერა:
პროგრამა უკრავს 'Twinkle Twinkle Little Star', მაგრამ ეს
მელოდია ადვილად შეიძლება შეიცვალოს. თვითოეული
მელოდია განისაზღვრება ერთი ცვლადით (ნოტების
რიცხვი - int length) და ორი მასივით, ერთი მასივი, notes[],
აწესებს ნოტების თანმიმდევრობას, მეორე, beats[] - მათი
ეღერადობის ხანგრძლივობას.
Twinkle Twinkle Little Star
int length = 15;
char notes[] = ("cegaagffeeddc");
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
Happy B irthday (first Line)
int length = 13;
char notes[] = {"ccdcfccdcgf "};
int beats[] = { 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 4};

```

არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC06>

CIRC-07

..:Нажатие на кнопку (кнопки):..



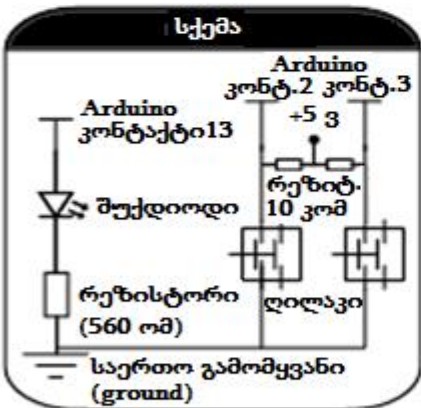
დავალების აღწერა:

აქამდე ჩვენ განვიხილავდით მართვის სხვადასხვა ვარიანტს. ახლა კი დადგა დრო ვასწავლოთ Arduino-ს ესმოდეს, ხედავდეს და გრძნობდეს. დავიწყეთ მარტივი დილაკით. დილაკის მიერთება ძალზედ მარტივია. საჭიროა მხოლოდ ერთი კომპონენტი - ამომქაჩავი რეზისტორი. იგი საჭიროა იმიტომ, არდუინო ისევე არ იღებს ინფორმაციას როგორც ჩვენ. იგი ვერ გრძნობს დაჭერილია დილაკზე თუ არა, ამის ნაცვლად იგი აკონტროლებს ძაბვის მნიშვნელობას თავის შესასვლელზე. დილაკის დაჭერა იწვევს შესასვლელზე ძაბვის განულებას (LOW), ხოლო დილაკის აშვებით ძაბვა ხდება მაღალი (HIGH). რეზისტორი საჭიროა იმიტომ, რომ არდუინოს შესასვლელზე ურყევად ფორმირებულ იქნას ლოგიკური ერთიანი როდესაც დილაკი აშვებულია. მიქციეთ ყურადღება იმას, რომ პირველ მაგალითში გამოიყენება მხოლოდ ერთი დილაკი.

მოწყობილობა:

კომპონენტები:

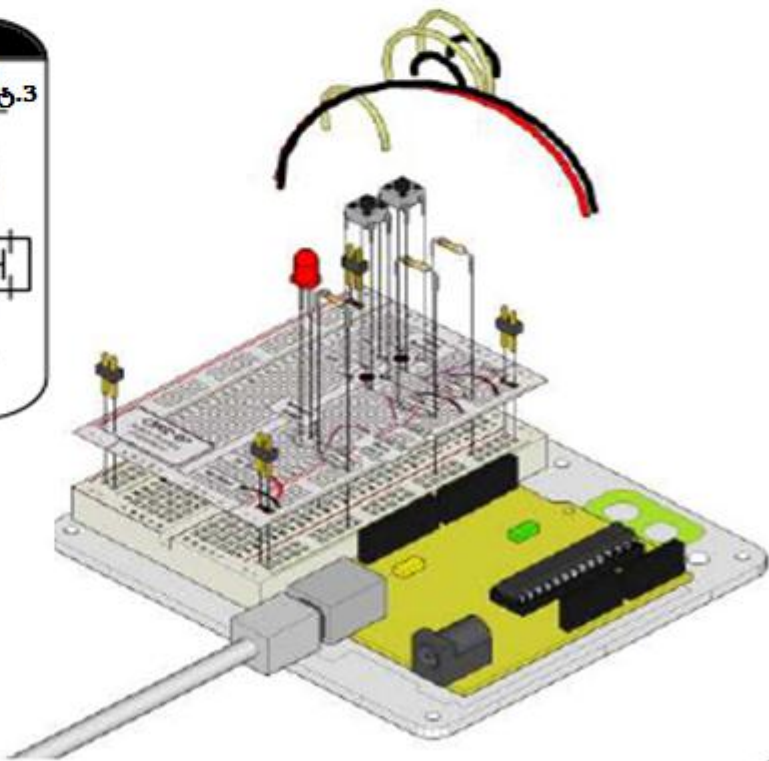
- დავალების ბარათი CIRC-07 x1
- დილაკი x2
- რეზისტორი 10 კომ ყავისფერ-შავი ნარინჯისფერი x2
- წითელი შუქდიოდი x1
- რეზისტორი 560 ომ მწვანე-ლურჯი ყავისფერი x1
- 2' კონტაქტური გასართი x4
- გამტარი



ინტერნეტში:

..:ჩამოტვირთეთ: დავალების ბარათი <http://arduino.org/BBLS07>

..:ნახეთ: ვიდეოები <http://arduino.org/VIDEO07>



File > Examples > 2.Digital > Button

(ეს მაგალითი არის arduino.cc საიტისა და, იქვე შეგიძლიათ ნახოთ სხვა იდეებიც)

```

^
* ღილაკი
* by DojoDave <http://www.0j0.org>
* ჩავრთავთ და გამოვრთავთ შუქდიოდს ღილაკის დაჭერით.
* http://www.arduino.cc/en/Tutorial/Button
*/
int ledPin = 13;           // ვადგენთ გაქმოსასვლელის ნომერს შუქდიოდისთვის
int inputPin = 2;         // ვადგენთ გაქმოსასვლელის ნომერს ღილაკისთვის
int val = 0;              // ცვლადი, ღილაკის მდგომარეობის შენახვისათვის
void setup () {
  pinMode(ledPin, OUTPUT);
  // ვახდენთ შუქდიოდთან მიერთებული კონტაქტის ინციალიზაციას, როგორც გამოსასვლელს
  pinMode(inputPin, INPUT);
  // ვახდენთ ღილაკთან მიერთებული კონტაქტის ინციალიზაციას, როგორც შესასვლელს
}
void loop () {
  val = digitalRead(inputPin); // ვკითხულობთ დავალებას ღილაკის გამოსასვლელიდან
  if (val == HIGH) {           // ვამოწმებთ, დაჭერილია თუ არა ღილაკი
    digitalWrite(ledPin, LOW); // შუქდიოდის გამორთვა
  } else {
    digitalWrite(ledPin, HIGH); // შუქდიოდის ჩართვა
  }
}

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტები)

შუქდიოდი ჩართვაზე

ღილაკი კვადრატული, ამიტომ მისი ჩასმა ადვილია თუ მოვამზადებთ 90 გრადუსით. შეამოწმეთ ღილაკის მიერთების სისწორე.

შუქდიოდი არ ინთება მდოვრედ (ქრება)

არ დაგავიწყდეთ შუქდიოდის გადართვა მე-13 პორტიდან მე-9 -ზე ამ დავალებისთვის.

იმედი გაგიცრუვდათ?

არ შეწუხდეთ, ყველა მოწყობილობა ამ ნაკრებში გამართ- ივებულია მაქსიმალურად ადვილად გასაგები რომ იყოს. შემდეგში კი თქვენს წინაშე გაიშლება ამოუწურავი შესაძლებლობები.

ვაუმჯობესებთ მოწყობილობას

ჩართვის ღილაკი და გამორთვის ღილაკი: გავართულოდ პირველი ამოცანა: ერთმა ღილაკმა ჩართოს შუქდიოდი მეორემ კი გამორთოს.

```

int ledPin = 13;
//შეირჩეს გასართი შუქდიოდისთვის
int inputPin1 = 3; //ღილაკი 1
int inputPin2 = 2; // ღილაკი 2
void setup () {
  pinMode (ledPin, OUTPUT);
  // შუქდიოდი განისაზღვრება როგორც გამოსასვლელი
  pinMode (inputPin1, INPUT); // ღილაკი 1- გამოსასვლ
  pinMode (inputPin2, INPUT); ღილაკი 2 - გამოსას.
}
void loop () {
  if (digitalRead(inputPin1 == LOW) {
    digitalWrite(ledPin, LOW); // შუქდიოდის გამორთვა
  } else if (digitalRead(inputPin2) == LOW {
    digitalWrite(ledPin, HIGH); //შუქდიოდის ჩართვა }
  } ჩატვირთეთ პროგრამა და შეამოწმეთ

```

შუქდიოდის მდორე ჩართვა და გამორთვა: მოდით გამოვიყენოთ შუქდიოდი ანალოგური სიგნალების მართვისთვის. ამისათვის გადავიტანოთ შუქდიოდი მე-13 პორტიდან მე-19-ზე და შევცვალოთ პროგრამა:

```

int ledPin = 13; ----> int ledPin = 9;
int value = 0;
void loop () {
  if (digitalRead(inputPin1) == LOW) {value --; }
  else if (digitalRead(inputPin2) == LOW) { value++ +; }
  value = constrain(value LOW, 0, 255);
  analogWrite(ledPin, value);
  delay(10);

```

შუქდიოდის ჩართვა/გამორთვის სიჩქარის ცვლილება: ამ შემთხვევაში საჭირო იქნება მხოლოდ ერთი ბრძანების შეცვლა პროგრამაში. delay(10); ----> delay(ახალი მნიშვნელობა) ეფექტის დაჩქარებისთვის აუცილებელია ამ რიცხვის შემცირება, ხოლო შენელებისთვის - გაზრდა.

კიდევ არის შეკითხვები?

წვრილმანები, სად შეიძლება დეტალების ყიდვა, სად შეიძლება შეკითხვების დასმა:

<http://ardx.org/CIRC07>

CIRC-08

.:ტრიალი (პოტენციომეტრი):.



დავალების აღწერა:

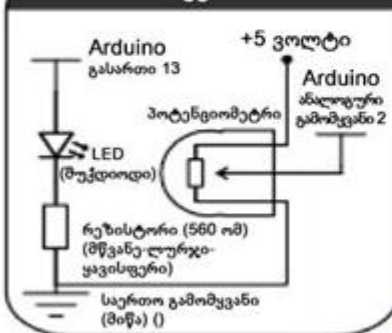
ციფრული პორტების გარდა Arduino-ს გააჩნია კიდევ ექვსი პორტი, რომლებიც შეიძლება იყოს ანალოგური შესასვლელი. ასეთი შესასვლელის საშუალებით შესაძლებელია 0 ... 5ვ - ის დიაპაზონში მდებარე მძვავი გარდაიქმნას ციფრულ კოდში დიაპაზონით 0 ... 1023 (რაც შეესაბამება 10 ბიტთან გარჩევადობას). პოტენციომეტრი (ცვლადი რეზისტორი) ეს ძალზედ სასარგებლო მოწყობილობაა), რომელიც შეიძლება გამოყენებულ იქნას ასეთ შესასვლელთან ერთად. თუ კი პოტენციომეტრი მიერთებული იქნება განაპირა გამოყვანებით "მიწასთან" და +5ვოლტთან, მაშინ შუალა გამოსასვლელზე იარსებებს რაღაც მძვავი 0 ... 5 ვ დიაპაზონში, რაც დამოკიდებულია პოტენციომეტრის ცოციალას მდგომარეობაზე (მოზრუნების კუთხეზე). შუალედურ მდგომარეობაში მძვავის მნიშვნელობა შუალა გამოსასვლელზე შესაბამისი იქნება 2,5 ვოლტის, ხოლო გარდაქმნილი მნიშვნელობა ტოლი იქნება 511, რაც ციფრულ ფორმაში (01111111) იქნება შეტანილი Arduino-ში. მძვავის ეს გაზომილი მნიშვნელობა პოტენციომეტრის შუალა გამოყვანაზე შეიძლება გამოყენებულ იქნას პროგრამაში როგორც ცვლადი სიდიდე.

მოწყობილობა:

კომპონენტები:

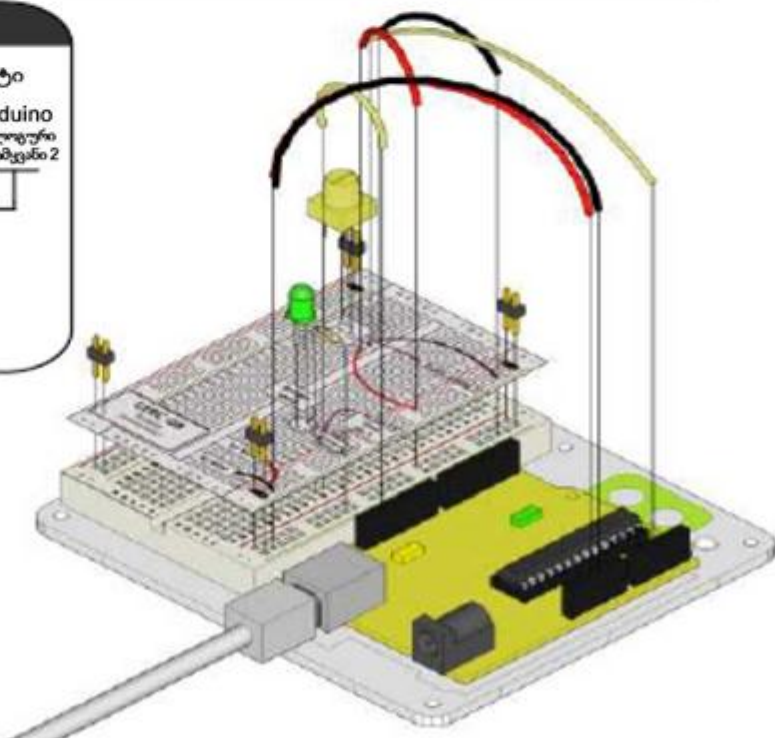
- დავალების ბარათი CIRC-08 x1
- მწვანე შუქდიოდი x1
- 2 კონტაქტური გასართი x4
- რეზისტორი 560 ომ მწვანე-ლურჯი-ყავისფერი x1
- პოტენციომეტრი 10 კომ x1
- გამტარი

სქემა



ინტერნეტში:

.გადმოქაჩეთ.
დავალების ბარათი
<http://ardx.org/BBL508>
.ნახეთ.
აწყობის ვიდეო
<http://ardx.org/VIDE08>



File > Examples > 3.Analog > AnalogInput

(ეს მაგალითი არის საიტდან arduino.cc)

```

/*
 *წაწერება ანალოგური შეერთება, მნიშვნელობის წაკითხვა ანალოგური სენსორიდან 0 და შუქდიოდის ჩართვა გათიშვა.
 *რომელიც შეერთებულია ციფრულ გამომცავთან 13. შუქდიოდის ჩართვა გამორთვის ხანგრძლივობა დამოკიდებულია
 *ცვლადზე, რომელიც მიღებულია analogRead()-ის დახმარებით
 *Created by David Cuatrecasas
 *Modified 16 Jun 2009
 *By Tom Igoe
 *http://arduino.cc/en/Tutorial/AnalogInput
 */
int sensorPin = 0; // ვანიჭებთ პორტის ნომერს პოტენციომეტრის წასაკითხად
int ledPin = 13; // ვანიჭებთ პორტის ნომერს შუქდიოდისთვის
int sensorValue = 0; // ვიცეხებთ სენსორიდან მიღებული მნიშვნელობის შესახად
void setup() {
  pinMode(ledPin, OUTPUT); //ვაყენებთ მე-13 კონტაქტს გამოცვანი რეჟიმში
}
void loop() {
  sensorValue = analogRead(sensorPin); //ანალოგური სიგნალის წაკითხვა
  digitalWrite(ledPin, HIGH); //შუქდიოდის ჩართვა
  delay(sensorValue); //დაყოვნება შესაბამისი ხანგრძლივობით
  digitalWrite(ledPin, LOW); //შუქდიოდის გამორთვა
  delay(sensorValue); //დაყოვნება შესაბამისი ხანგრძლივობით
}

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

პოტენციომეტრი დროდადრო წყვეტს მუშაობას

ერთერთი მიზეზი შეიძლება იყოს ცუდი კონტაქტი პოტენციომეტრის გამომსვლეულსა და ბარათის შორის შეეცადეთ მჭიდროდ მიაბეჭინოთ პოტენციომეტრი ბარათს.

არ მუშაობს

დარწმუნდით რომ არ შეერთეთ პოტენციომეტრის შუა გამომსვლეული ციფრულ გამომსვლეულთან ანალოგურის ნაცვლად (ანალოგური გამომსვლეული კვების გამომსვლეულის რიგში).

მაინც არ მუშაობს

შეეცადეთ შეცვალოთ პოტენციომეტრის შეერთების პოლარობა. ზოგჯერ ეს შევლის.

ვაუმჯობესებთ მოწყობილობას

ზღვრული ჩამრთველი:

ზოგჯერ საჭიროა რაიმეს გადართვა როდესაც დაკვირვებადი პარამეტრი მიაღწევს რაიმე ნიშნულს (მაგ. წყლის გათიშვა სარეზი მანქანაში, როცა წყალი მიაღწევს საკმარის დონეს), რომ შევქოთ პოტენციომეტრის გამოყენება, კოდი უნდა შეიცვალოს შემდეგნაირად:

```

void loop() {
  int threshold = 512;
  if(analogRead(sensorPin) > threshold) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}

```

შუქდიოდი ჩართვება/გამორთვება შუა მნიშვნელობის მიღწევისას, თქვენ შეგიძლიათ მგრძობაარობის შევლა threshold პარამეტრის ცვლილებით.

სიკამკამის კონტროლი:

სიკამკამის კონტროლი შესაძლებელია უშუალოდ პოტენციომეტრით. ამისათვის საჭიროა გადავრთოთ შუქდიოდი პორტი 13-დან პორტ 9-ზე და

შეცვალოთ პროგრამის ტექსტი:

```

int ledPin = 13; --> int ledPin =

```

შეცვალოთ ფუნქცია loop:

```

void loop() {
  int value = analogRead(potPin) / 4;
  analogWrite(ledPin, value);
}

```

ჩატვირთეთ პროგრამა მიკროკონტროლერში და დარწმუნდით, რომ შუქდიოდის სიკამკამე დამოკიდებულია პოტენციომეტრის თავის მდგომარეობაზე. ჩვენ გაცვაკით მიღებული მნიშვნელობა 4, რადგანაც სიკამკამე იცვლება მხოლოდ 0...255 ფარგლებში (8 ბიტი), ცოლო ფუნქცია analogRead() - 0...1024 ბარგებში (10 ბიტი).

სერვოამძრავის კონტროლი:

ეს საინტერესო მაგალითი, აერთებს ორ მოწყობილობას ერთში. შეერთედ სერვოამძრავი როგორც დაგალება CIRC-04-ში, გახსენით მაგალითი File>Examples>Servo>Kirob, შეცვალეთ ხაზი:

```

int potPin = 0; --> int potPin = 2;

```

ჩატვირთეთ პროგრამა მიკროკონტროლერში და დარწმუნდით რომ, პოტენციომეტრის მდგომარეობა განსაზღვრავს სერვოამძრავის მდგომარეობას.

არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წერილმანი

<http://ardx.org/CIRC08>

CIRC-09

..შუქი
(ფოტორეზისტორი)..



დავალების აღწერა:

წინა მაგალითში ჩვენ განვიხილეთ პოტენციომეტრი, რომელიც შეგვიძლია გამოვიყენოთ ინფორმაციის შესაყვანდ ადამიანის მიერ. არსებობენ მრავალნაირი სენსორები, რომლებიც საშუალებას გვაძლევს მივიღოთ ინფორმაცია გარემოს შესახებ. მაგალითად ფოტორეზისტორი რომელიც გამოიყენება განათების აღსაქმელად. Arduino-სთან მუშაობის პრინციპი არ იცვლება. Arduino-ს არ შეუძლია უშუალოდ გაზომოს წინააღმდეგობა (შესაძლებელია გაიზომოს ძაბვა), ამიტომ ფოტორეზისტორი ირთვება როგორც ძაბვის გამყოფის ნაწილი (<http://ardx.org/VODI>). შესაძლებელია ზუსტი ძაბვის გაზომვა ანალოგურ შესასვლელზე, მაგრამ ჩვენი დავალებისთვის ეს არ არის საჭირო. გავზომოთ ფარდობითი განათების სიმძლავრე. პატარა მნიშვნელობები შეესაბამება კამკაშა განათებას, ხოლო დიდი მნიშვნელობები ნაკლებ განათებას.

მოწყობილობა:

კომპონენტები:

- დავალის ბარათი CIRC-09 x1
- რეზისტორი 10 კომ ყავისფერი-შავი -სტაფილოსფერი x2

- 2 კონტაქტური გასართი x4

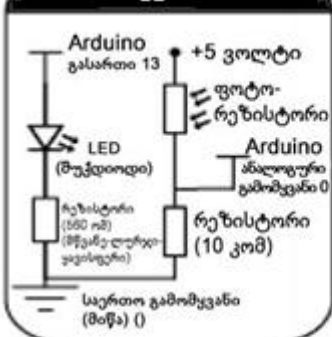
- ფოტორეზისტორი x1

- გამტარი

- რეზისტორი 560 ომ მწვანე-ლურჯი-ყავისფერი x1

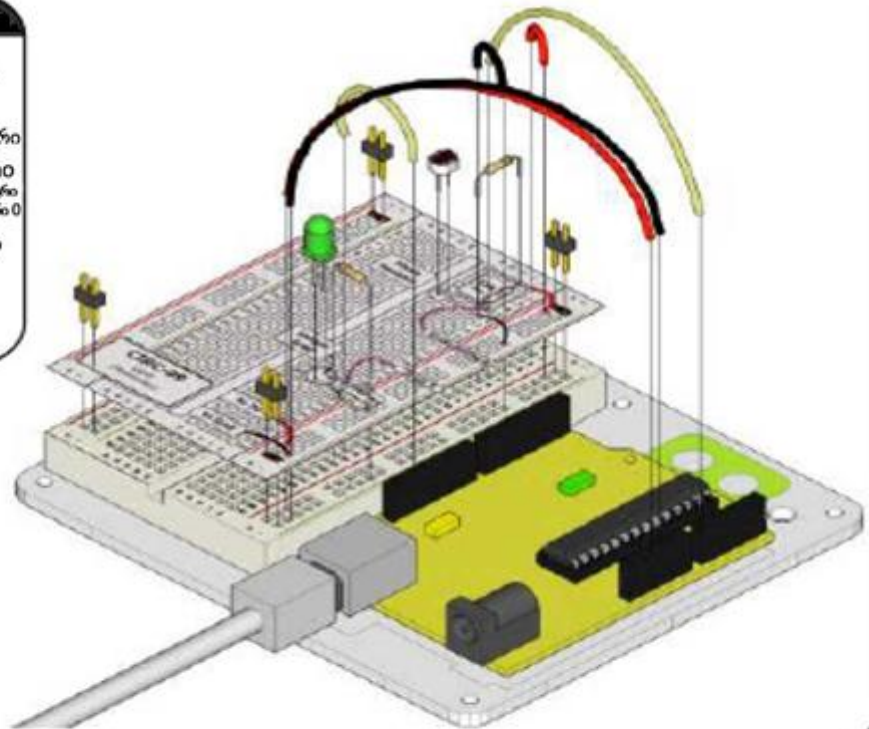
- მწვანე შუქდიოდი x1

სქემა



ინტერნეტში:

.გადმოქაჩეთ.
დავალის ბარათი
<http://ardx.org/BBL509>
ნახეთ.
აწყობის ვიდეო
<http://ardx.org/VIDE09>



კოდი არ არის საჭირო ხელით აკრეფა.

CIRC-09

მისი გადმოწერა შესაძლებელია მისამართზე: <http://ardx.org/CODE09>
(დააკოპირეთ ტექსტი და ჩასვით Arduino Sketch-ის ახალ ფანჯარაში)

```

/*
 * მარტივი პროგრამა, რომელიც ზომავს შუქდიოდის განათების ინტენსიურობას, რომელიც ეგება ფოტორეზისტორზე/
 */
//PhotoResistor Pin
int lightPin = 0; //ანალოგური შესასვლელი, რომელზეც შეერთებული ფოტორეზისტორი. ფოტორეზისტორი არ არის დაკალიბრებული
// რაიმე ერთეულზე, ის უბრალოდ გამოსცემს ძაბვას განათების სიმძლავრის პროპორციულად.
//value (relative light)
//LED Pin
int ledPin = 9;
//კონტაქტი რომელზეც შეერთებულია შუქდიოდი, ჩვენ ვაკონტროლებთ სიკაშკაშეს ამიტომ ვაერთებთ მას ანალოგურ გამოსასვლელზე.
void setup()
{
  pinMode(ledPin, OUTPUT);
  //ვინიციალიზირებთ შუქდიოდის კონტაქტს, როგორც გამომსვლელი
}
/*
 * ფუნქცია loop() იწყება ფუნქცია setup-ის დასრულების შემდეგ
 */
void loop()
{
  int lightLevel = analogRead(lightPin); //ვკითხულობთ განათების
  lightLevel = map(lightLevel, 0, 900, 0, 255); //გადაგვყვავს გვლადის მნიშვნელობა 0-900 შუალედში
  lightLevel = constrain(lightLevel, 0, 255); //შემოგსაზღვროთ მნიშვნელობა 0-255 შუალედით
  analogWrite(ledPin, lightLevel); //გამოგვყვავს გვლადის მნიშვნელობა
}

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

შუქდიოდი რჩება გათიშული

ზმირად შუქდიოდის შეერთება ხდება არასწორი პოლარობით, შეატრიალეთ ის 180 გრადუსით

მოწყობილობა არ რეაგირებს შუქის ვლილებაზე

შეამოწმეთ სწორედ არის თუარა შეერთებული ფოტორეზისტორი

მაინც არ მუშაობს?

შეიძლება იმ ოთახში რომელშიც იმყოფებით, ძალიან ბნელა ან პირიქით, ძალიან დიდი განათებაა. შეეცადეთ შუქის გათიშვა-ჩართვას, თუ თქვენ გაქვთ ფანარი, მიანათეთ ის სენსორს.

ვაუმჯობესებთ მოწყობილობას

გამომსვლელის ინვერტაცია:

თუ გინდათ, რომ შუქდიოდმა იმუშაოს შეტრიალებულ რეჟიმში. ამისათვის საჭიროა კოდის შეცვლა შემდეგნაირად:
analogWrite(ledPin, lightLevel); ->
analogWrite(ledPin, 255 - lightLevel);

ლაშპა:

იმის მაგივრად, რომ შევცვალოთ შუქდიოდის სიკაშკაშე გარემოს განათების მიხედვით, შეგვიძლია მხოლოდ გავთიშოთ და ჩავრთოთ ის. შევცვალოთ loop() ფუნქციის ტექსტი შემდეგნაირად:

```

void loop() {
  int threshold = 300;
  if(analogRead(lightPin) > threshold) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, Low);
  }
}

```

სერვოამძრავი, შუქით მართვა:

გამოვიყენოთ ეს სენსორი სერვოამძრავის სამართავად. შეაერთეთ სერვოამძრავი მე-9 პორტში (როგორც CIRC-04-ში). ჩატვირთეთ კონტროლერში პროგრამა Knob File > Examples > Servo > Knob (მაგალითი CIRC-08-dan). შეხედეთ როგორ იმუშავებს სერვოამძრავი.

თქვენ ალბათ შეამჩნიეთ რომ სერვოამძრავი არ მუშაობს სრულ დიაპაზონში. ეს ხდება იმიტომ, რომ დიაპაზონის გამყოფის გამო ჩვენ ვმუშაობთ ნაკლებ დიაპაზონში, ვიდრე 0...ნვ. ეს შეგვიძლია გამოვასწოროთ შემდეგი ბრძანებით: val = map(val, 0, 1023, 0, 179); ფუნქციის დამატებითი აღწერისთვის ნახეთ მისამართი: <http://arduino.cc/en/Reference/Map>.

არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC09>

CIRC-10

.:ტემპერატურა (ტემპერატურის საზომი TMP36):.



დავალების აღწერა:

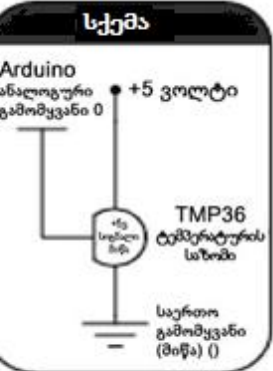
კიდევ რა შეგვიძლია გავზომოთ არდუინოს დახმარებით? ტემპერატურა!
 ტემპერატურის გასაზომად ჩვენ გამოვიყენებთ შედარებით რთულ მიკროსკემას, რომელიც დამალულია P2N22AG ტრანზისტორის კორპუსში. ამ მიკროსკემას გააჩნია სამი გამოსასვლელი: <<მიწა>>, სიგნალი და კვება +5ვ. ის გადმოსცემს 10მვ/გრადუსი ცელსიუსი (იმისათვის რომ გავზომოთ უარყოფითი ტემპერატურებიც, გათვალისწინებულია ძაბვის წანაცვლება 500 მვ, შესაბამისად: 25°C = 750 მვ, 0°C = 500 მვ). იმისთვის რომ, გარდავსახოთ ძაბვის მნიშვნელობა ტემპერატურაში, გამოვიყენოთ არდუინოს მათემატიკური შესაძლებლობები. ტემპერატურის გამოსატანად გამოვიყენოთ არდუინოს ინტერფეისის დამატებითი ფანჯარა. ჩვენ გვაგ ზავნით ტემპერატურის მნიშვნელობას მიმდევრობით პორტზე ტემპერატურის ეკრანზე გამოსატანად. ეს მოწყობილობა იყენებს მიმდევრობითი პორტის მონიტორს. იმისთვის რომ გავუშვათ ის, საჭიროა ჩავტვირთოთ პროგრამა არდუინოში, შემდეგ დავაკვიროთ ღილაკს რომელიც გავს მართკუთხედს ანტენით.



TMP36 საზომის დოკუმენტაცია: <http://ardx.org/TMP36>

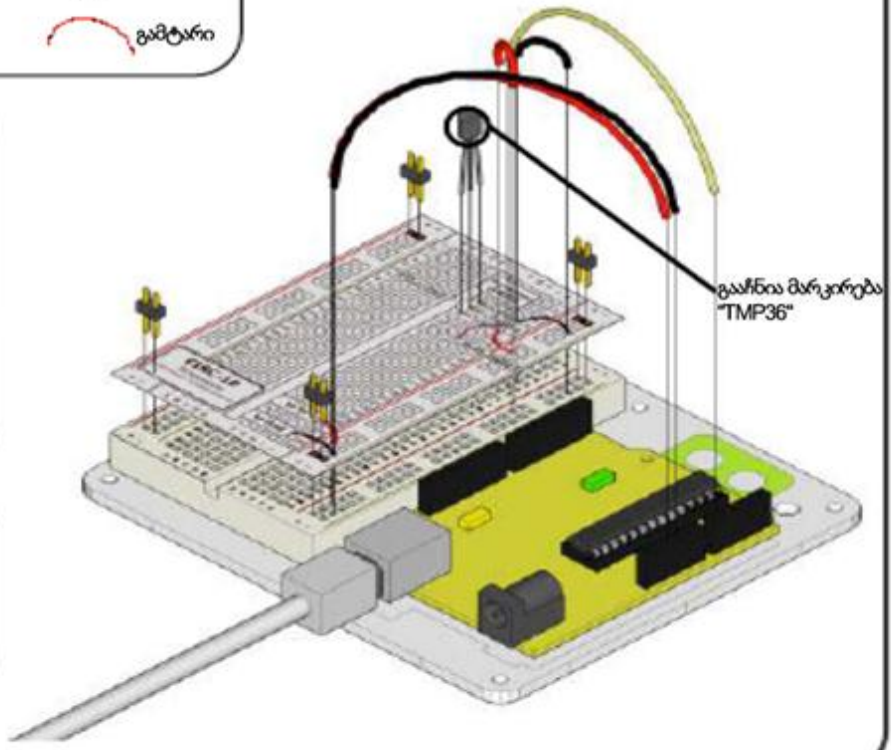
მოწყობილობა:

- კომპონენტები:**
- დავალების ბარათი CIRC-09 x1
 - ტემპერატურის საზომი TMP36 x1
 - 2 კონტაქტური გასართი x4
 - გამტარი



ინტერნეტში:

.გადმოქაჩეთ. დავალების ბარათი <http://ardx.org/BBL510>.
 .ნახეთ. აწყობის ვიდეო <http://ardx.org/VIDE10>



კოდი არ არის საჭირო ხელით აკრეფა.

CIRC-10

მისი გადმოწერა შესაძლებელია მისამართზე: <http://ardx.org/CODE09>
(დააკოპირეთ ტექსტი და ჩასეთ Arduino Sketch-ის ახალ ფანჯარაში)

```

//
// | არდუინოს ექვიმენტარული ნაკრების მაგალითი |
// | CIRC-10 : ტემპერატურა : (TMP36 ტემპერატურის საზომი) |
//
//
// * მართვი პროგრამა ტემპერატურის მიმდევრობითი პორტის ფანჯარაში
// * გამოსატანად
//
// * დამატებითი ინფორმაცია ამ სტუმარზე: http://inyurt.com/c89vd
//
//
//TMP36 Pin Variables
int temperaturePin = 0; //the analog pin the TMP36's Vout (sense) pin is connected to
//the resolution is 10 mV / degree centigrade
//500 mV offset) to make negative temperatures an option

//
// * setup() - this function runs once when you turn your Arduino on
// * We initialize the serial connection with the computer
//
void setup()
{
  Serial.begin(9600); //Start the serial connection with the copmuter
  //to view the result open the serial monitor
  //last button beneath the file bar (looks like a box with an antennae)
}

void loop() // run over and over again
{
  float temperature = getVoltage(temperaturePin);
  //getting the voltage reading from the temperature sensor

  temperature = (temperature - .5) * 100;
  //converting from 10 mv per degree with 500 mv offset
  //to degrees ((voltage - 500mV) times 100)

  Serial.println(temperature); //printing the result
  delay(1000); //waiting a second
}

//
// * getVoltage() - returns the voltage on the analog input defined by
// * pin
//
float getVoltage(int pin){
  return (analogRead(pin) * .004882814); //converting from a 0 to 1023 digital range
  // to 0 to 5 volts (each 1 reading equals ~ 5 millivolts)
}

```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

არაფერი არ ხდება?

ეს პროგრამა არ იყენებს იდენტიფიკაციას შედეგის გამოსატანად, თქვენ უნდა გაუშვათ მიმდევრობითი პორტის ფანჯარა.

გამოსახება უცნაური ინფორმაცია

შესაზღვრელია რომ მიმდევრობითი პორტის ფანჯარა იღებს მონაცემებს განსხვავებული სიხშირით, გადართეთ მისი მუშაობის სიჩქარე <<9600 baud>>

ტემპერატურის მნიშვნელობა არ იცვლება

შეეხეთ სენსორს რათა გაათბოთ, ან გამოიყენოთ ყინული მის გასაციებლად.

ვაუმჯობესებთ მოწყობილობას

ძაბვის ჩვენება:

ამის გაკეთება მარტივია, რადგანაც სენსორი უკვე გადმოგვცემს ძაბვას. საკმარისია წავშალოთ ხაზი
 $temperature = (temperature - .5) * 100;$

ტემპერატურის ჩვენება ფარენჰეიტებში:

ესეც არ არის რთული, რადგანაც ვებებით მხოლოდ გამოთვლით ნაწილს. იმისთვის რომ, გადავიღოთ ცელსიუსიდან ფარენჰეიტზე საჭიროა შემდეგი ფორმულის გამოყენება:
 $F = C * 1.8 + 32$
 დამატეთ ხაზი
 $temperature = (((temperature - .5) * 100) * 1.8) + 32;$
 $Serial.println(temperature);$ ხაზამდე

უფრო ინფორმაციული შედეგების ჩვენება:

დავამაოდ შეტყობინება შედეგს, რომელსაც ვაგზავნით მიმდევრობითი პორტით. დაუბრუნდით საწყის პროგრამას და შეცვალეთ:
 $Serial.println(temperature);$ → $Serial.print(temperature);$
 $Serial.println(" degrees centigrade");$
 ცვლილება პირველ ხაზში ნიშნავს რომ შემდეგი ინფორმაციის გამოტანა მოხდება იგივე ხაზზე. შემდეგი ხაზი კი იძახებს დამზარე ფუნქციას და ინფორმაციის გამოტანის შემდეგ გადადის ახალ ხაზზე.

ინფორმაციის გადაცემის სიჩქარის ცვლილება

თუ თქვენ აპირებთ გააგზავნოთ დიდი მოცულობის ინფორმაცია მიმდევრობითი პორტით, მაშინ საჭიროა სიჩქარის გაზრდა. ჩვენ მაგალითში გადაცემის სიჩქარეა 9600 ბიტი. შესაძლებელია გავილებით მეტი სიჩქარის გამოყენება.
 შეცვალეთ ხაზი: $Serial.begin(9600);$ → $Serial.begin(115200);$
 ამ შემთხვევაში ინფორმაციის გადაცემის სიჩქარე გაიზარდა 12-ჯერ. არ დაგავიწყდეთ შეცვალოთ მიმდევრობითი პორტის ფანჯრის სიჩქარის შეცვლა.

არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიპოვოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC10>

CIRC-11

..დიდი დატვირთვები (რელე):.




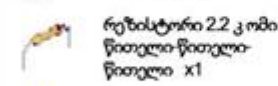

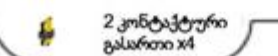


დავალების აღწერა:

ბოლო დავალება თავისმხრივ გამოსაცდელი დავალებაა. ჩვენ გამოვიყენებთ ჩვენ ცოდნას ტრანზისტორებზე (CIRC03) რელეს სამართავად. რელე - ეს არის ელექტრო-მექანიკური ჩამრთველი. პლასტიკური ხუფის ქვეშ არის ელექტრომაგნიტი, რომელზეც ძაბვის მიწოდებისას ხდება მექანიკური კონტაქტის ჩართვა (შესამჩნევი ხმით). განსხვავებული რელეები ძალიან განსხვავდებიან ზომით და პარამეტრებით (ძალიან პატარებიდან, რელემდე რომელიც მაცივრის ხელაა). ჩვეულებისამებ რაც უფრო დიდია რელე, მით უფრო მეთი დენის გატარება შეუძლია. რელეს და არდუინოს დახმარებით შეძლებთ მნიშვნელოვანი საგნების კონტროლს.

მოწყობილობა:

კომპონენტები:

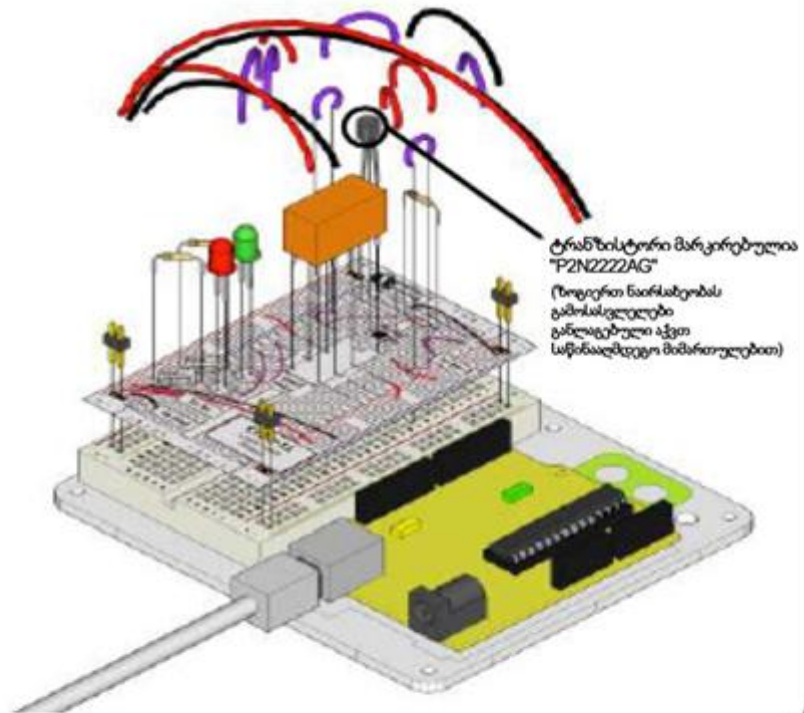
- | | | | |
|---|--|---|---|
|  დავალების ბარათი CIRC-11 x1 |  დიოდი (1N4001) x1 |  ტრანზისტორი P2N2222AG (TO92) x1 |  რელე (DPDT) X1 |
|  რეზისტორი 2.2 კომი წითელი-წითელი-წითელი x1 |  რეზისტორი 560 ომი მწვანე-ლურჯი-ყვითელი x2 |  წვანე შუქდიოდი x1 |  წითელი შუქდიოდი x1 |
|  2 კონტაქტური გასართი x4 | | | |

სქემა



ინტერნეტში

.გადმოქაჩეთ. დავალების ბარათი <http://ardx.org/BBL511>. ნახეთ. აწყობის ვიდეო <http://ardx.org/VIDE11>



კოდი არ არის საჭირო მისი ხელით აკრეფა, იგი იმყოფება მისამართზე:

CIRC-11

File > Sketchbook > 1.Basic > Blink

(მაგალითი საიტიდან arduino.cc, ასევე მანდვე შეგიძლიათ მოძებნოთ ბევრი იდეა) ციმციმი

ჩართე შუქდიოდი ერთი წამით, შეჰმდეგ გამორთე ერთი წამით, გამეორებით სტემა:

- * შუქდიოდი შეერთებულია გამომავალ 13-ზე და მიწაზე.
- * შენიშვნა: უმეტეს არდუინოს უკვე გააჩნია შუქდიოდი მე-13 გამოსასვლე ამიტომ თქვენ არ დაგჭირდებათ დამატებითი კომპონენტები

<http://arduino.cc/en/Tutorial/Blink>

```
int ledPin = 2; // რელე შეერთებულია გამოსავალ 2-ზე <--- Change this to pin 2
// The setup()
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

არაფერი არ ხდება?

პროგრამა იყენებს პორტ 13, ხოლო რელე შეერთებულია პორტ 2-ზე. არ დაგავიწყდეთ შესაბამისად შეცვალოთ პროგრამა.

რელე არ გამოსცემს ხმას

შესაძლებელია არ მუშაობს ტრანზისტორი, დიდი უზრუნველებით შეამოწმეთ მოწყობილობის სისწორე.

ზუსტად არ მუშაობს

ნაკრებში შემავალი რელე გათვლილია მიღუღებაზე, და არა მაკეტურ პლატაზე. დააჭირეთ რელეს ცოტაოდენი ძალით, ამან შეიძლება უშველოს.

ვაუმჯობესებთ მოწყობილობას

უკუ დენის იმპულსების ვიზუალიზაცია შეცვალეთ დიოდი შუქდიოდით. თქვენ დაინახავთ ნათებას ყოველ აქრზე, როცა რელე გათიშება.

ძრავის მართვა

CIRC03 ჩვენ ვმართავდით ძრავს ტრანზისტორის დახმარებით. თუმცა თუ თქვენ აპირებთ გაცილებით მძლავრი ძრავის მართვას, რელეს გამოყენება იქნება უკეთესი გადაწყვეტა. ამისათვის უბრალოდ მოხსენით წითელი შუქდიოდი და მის მაგივრად შეაერთეთ ძრავი. (არ დაგავიწყდეთ 650 ომ რეზისტორის მოხსნა).

ძრავის ბრუნვის მიმართულების ცვლილება

ბოლოსთვის ცოტათი გავართულოთ მოწყობილობა. მუდმივი დენის ძრავის ბრუნვის მიმართულების შესაცვლელად საჭიროა დენის მიმართულების შეცვლა. თუ ამას გავაკეთებთ ხელით - საკმარისია ადგილები გავუცვალოდ გასართებს. ელექტრონული კომუნიკაციისთვის საჭიროა გამოვიყენოთ H-ხიდის ნაირი რამ. ეს შეგვიძლია გავაკეთოთ DPDT რელეს გამოყენებით. ააწყეთ შემდეგი სტემა:



არის კიდევ კითხვები?

სად დავსვათ კითხვები, სად ვიყიდოთ დეტალები და სხვ. ნებისმიერი წვრილმანი

<http://ardx.org/CIRC11>

CIRC-12

:: მრავალფერა ნათება (RGB შუქდიოდები) ::



დავალების აღწერა:

თქვენ უკვე შეგიძლიათ აციმციმით შუქდიოდები და გამოიყენოთ განივიმპულსური მოდულიაციის პრინციპი მრავლების მართვისთვის. მოდით გამოვიყენოთ ეს ცოდნა და შევქნათ შუქდიოდი, რომელიც გაანათებს ნებისმიერი ფერითა და ნებისმიერი ინტენსივობით (RGB შუქდიოდების ბაზაზე).

RGB შუქდიოდი წარმოადგენს სამ შუქდიოდს - R წითელი, G მწვანე, B ლურჯი, რომლებიც მოთავსებულია ერთ კორპუსში. თუ კი გამოვიყენებთ ამ ფერების სხვადასხვა კომბინაციებს შეგვეძლება პრაქტიკულად ყველა ფერის სინთეზი.

RGB შუქდიოდის თვითოეული არხის კონტროლისათვის გამოიყენება ჩვეულებრივი შუქდიოდი, შესაბამისად, ფერების შერევა ხდება თვალსაჩინოდ.

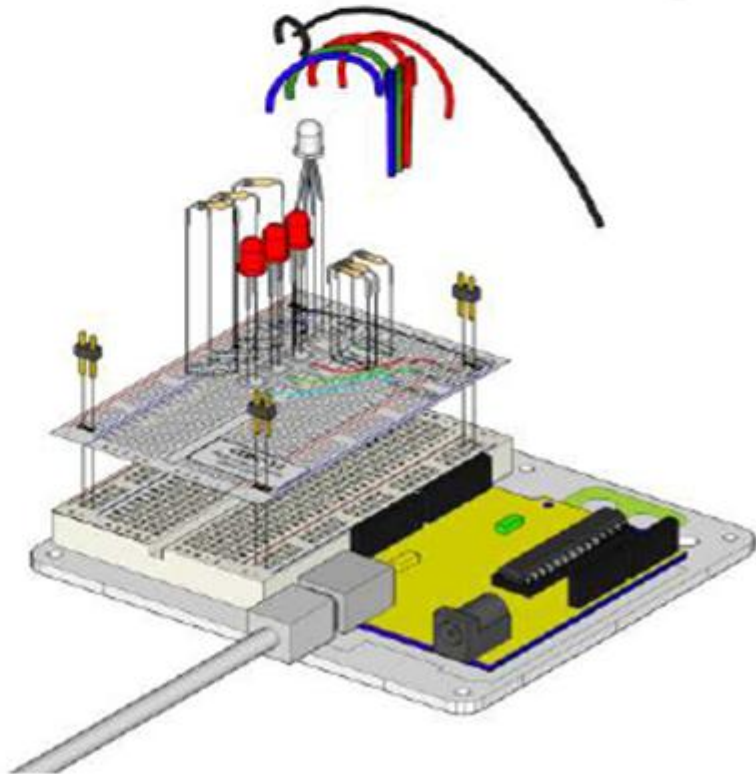
მოწყობილობა:

კომპონენტები:

- დავალების ბარათი CIRC-RGB x1
- რეზისტორი 560 ომ მწვანე-ლურჯი-ყავისფერი x 6
- 2 კონტაქტიანი გასართი x 4
- შუქდიოდი RGB 5 88 x1
- წითელი შუქდიოდი x1
- გამტარი

ინტერნეტში:

::გადმოქაჩეთ::
დავალების ბარათი
<http://ardx.org/BBL512R>



<http://ardx.org/CODE12R>

(დააკოპირეთ ტექსტი და ჩასვით Arduino sketch-ის ახალ ფანჯარაში)

```
// RGB LED pins
int ledDigitalOne90 = {9, 10, 110};
//3 შუქდიოდების ციფრული გამოსასვლელი
//9 - redPin, 10 - greenPin, 11 - bluePin
const boolean ON - LOW;
// გამოიყენება შუქდიოდები საერთო ანოდით
const boolean OFF - HIGH;
//გამორთული მდგომარეობისთვის ვაწვდით HIGH;
// ფერის მოცემულობა
const boolean RED() = {ON, OFF, OFF};
const boolean GREEN() = {OFF, ON, OFF};
const boolean BLUE() = {OFF, OFF, ON};
const boolean YELLOW() = {ON, ON, OFF};
const boolean CYAN() = {OFF, ON, ON};
const boolean MAGENTA() = {ON, OFF, ON};
const boolean WHITE() = {ON, ON, ON};
const boolean BLACK() = {OFF, OFF, OFF};
// ფერების მოცემულობის შემსახვე მასივი
[RED, GREEN, BLUE, YELLOW, CYAN, MAGENTA,
WHITE, BLACK];
void setup() {
  for(int i = 0; i < 3; i++) [
    pinMode(ledDigitalOne[i], OUTPUT);
  ]
  //3 შუქდიოდის გამომყვანების ინციალიზაცია ხდება
  //როგორც გამოსასვლელის
  ]
  void loop() [
    setColor(ledDigitalOne, CYAN) ;
    //მიუთითებს შუქდიოდს ფერზე
    // randomColor()
  ]
  void randomColor() [
    int rand = random90, sizeof(COLORS) / 2);
    // აბრუნებს შემთხვევით რიცხვს ფერთა დიაპაზონიდან
    setColor(ledDigitalOne, COLORS[rand]);
    //უთითებს შუქდიოდის ფერს როგორც შემთხვევის
    delay(1000);
  ]
  void setColor(int* led, boolean* color) [
    for(int i = 0; i < 3; i++) [
      digitalWrite(led[i], color[i]);
    ]
  ]
}
```

არ მუშაობს? (3 პრობლემა და მათი გადაწყვეტა)

შუქდიოდი არ ანათებს, ანდა აჩვენებს არასწორ ფერებს
 ყურადღებით შეამოწმეთ RGB შუქდიოდის შეერთება. არ დაგავიწყდეთ დიოდის გამომყვანის მიერთება საერთო "მიწასთან".

ყველა ფერი ნამეტანად გაწითლებულია
 შესაძლებელია RGB-ს წითელი შუქდიოდი ანათებს სხვებზე უფრო ნათლად. ამის კომპენსირება შესაძლებელია R არხში რეზისტორის გაზრდით

მალზედ ბევრი შუქდიოდებია?
 ერთფერა შუქდიოდები გამოიყენება RGB-ს არხების სიგნალების დროებითი ინდიკაციისათვის. შემდეგში ისინი შეგეძლებათ მოაშოროთ სქემიდან.

ვაუმჯობესებთ მოწყობილობას

მეტი ფერები
 თქვენ ალბათ მალზედ გაკვირვებული ხართ შუქდიოდის ღია ლურჯი ფერით. რომ შეიცვალოს მისი ფერი სხვა ფერით - შეცვალეთ პროგრამა:
 setColor(ledDigitalOne, CYAN); —>
 setColor(ledDigitalOne, "NEW COLOR");
 შემთხვევითი ფერების ასახვა რატომ უნდა, თქვენ შეგიძლიათ უფრო მეტი, ვიდრე მუდმივი ფერების ჩვენება. შეცვალეთ პროგრამა შემდეგნაირად

```
void loop() {
  // setColor(ledDigitalOne, CYAN);
  randomColor()
}
```

(დააკოპირეთ ტექსტი და ჩასვით იგი Arduino Sketch-ის ახალ ფანჯარაში)

ფერის ანალოგური კონტროლი
 ფიქსირებულ ფერებს შორის გადართვა საკმაოდ საინტერესოა. მაგრამ ანალოგური კონტროლის გამოყენებით შეიძლება პრაქტიკულად შეუზღუდავი რაოდენობის ფერის მიღება.
 ჩამოტვირთეთ მაგალითი მისამართიდან:
<http://ardx.org/MABE12R>

არის კიდევ კითხვები?

წვრილმანები, თუ სად შევიძინოთ დეტალები პროექტისათვის, სად დავსვათ შეკითხვები:

<http://www.solarbotics.com>

A large rectangular area with rounded corners, containing 25 horizontal lines for writing notes.

..ადგილი ჩანაწერებისათვის:

შენიშვნები

A large rectangular area with horizontal lines for writing notes. The lines are evenly spaced and cover most of the page's width and height, leaving margins at the top and bottom.

ARDX

Arduino-ს კომპლექტი ექსპერიმენტებისათვის



www.oomlout.com

