

# ՍԵՌՈԲ 3-ի կիրառումը արտաքին սարքերի հետ



## Պարզ հրամանների կիրառումը Սեռոբ-3-ի համար (python)

- 1) Ստեղծում ենք համապատասխան կիչքը Desktop-ի վրա `անուն.py
- 2) Բացում ենք terminal-ը և հավաքում` cd Desktop և սեղմում enter հրամանը
- 3) sudo python անուն.py և սեղմում enter հրամանը

### Անալոգային (սև գծին հետևող տվիչների դեպքում)

```
#!/usr/bin/python
# Ներմուծել class-ներ SERob3 փաթեթից
from SERob3 import Analog

# Ստեղծում է անալոգային class-ի օբյեկտը և կանչում է վերջինիս մեթոդը
myAnalog = Analog()
# կարդում է SERob3-ի 5-րդ անալոգային մուտքի արժեքը
print(myAnalog.In(5))
# After closing the myAnalog instance all the I2C bus will be released
myAnalog.Close()
```

### Ultrasonic (հեռաչափեր)

```
#!/usr/bin/python
# Import classes from SERob3 package
from SERob3 import Usrf
# import time

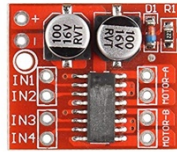
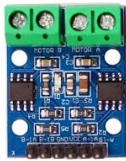
# Create an object of Usrf class & call a method of it
myUsrf = Usrf()
# Read distance in cm from SERob3 GPIO 8 and 9
print(myUsrf.Distance(8))
# Close the myUsrf instance before exiting the program myUsrf.Close()
```

## Շարժիչների իմպուլս

```
#!/usr/bin/python
# Import classes from SERob3 package
from SERob3 import Pwm
import time

# Create an object of Pwm class & call a method of it
myPwm = Pwm()
# Output 1600 us pulse width to SERob3 PWM 2 pin
myPwm.Out(2, 70)
time.sleep(3)
myPwm.Out(2, 0)
time.sleep(1)
myPwm.Out(2, -70)
time.sleep(2)
# After closing the myPwm instance all the PWMs will output 1350 us pulse width to stop
motors
myPwm.Close()
```

## L9110/L298N շարժիչների ղեկավարումը Այգեստանի միջոցով



```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)

pinForward = 23
pinReverse = 24

GPIO.setup(pinForward, GPIO.OUT)
GPIO.setup(pinReverse, GPIO.OUT)

p = GPIO.PWM(pinForward, 50)
q = GPIO.PWM(pinReverse, 50)

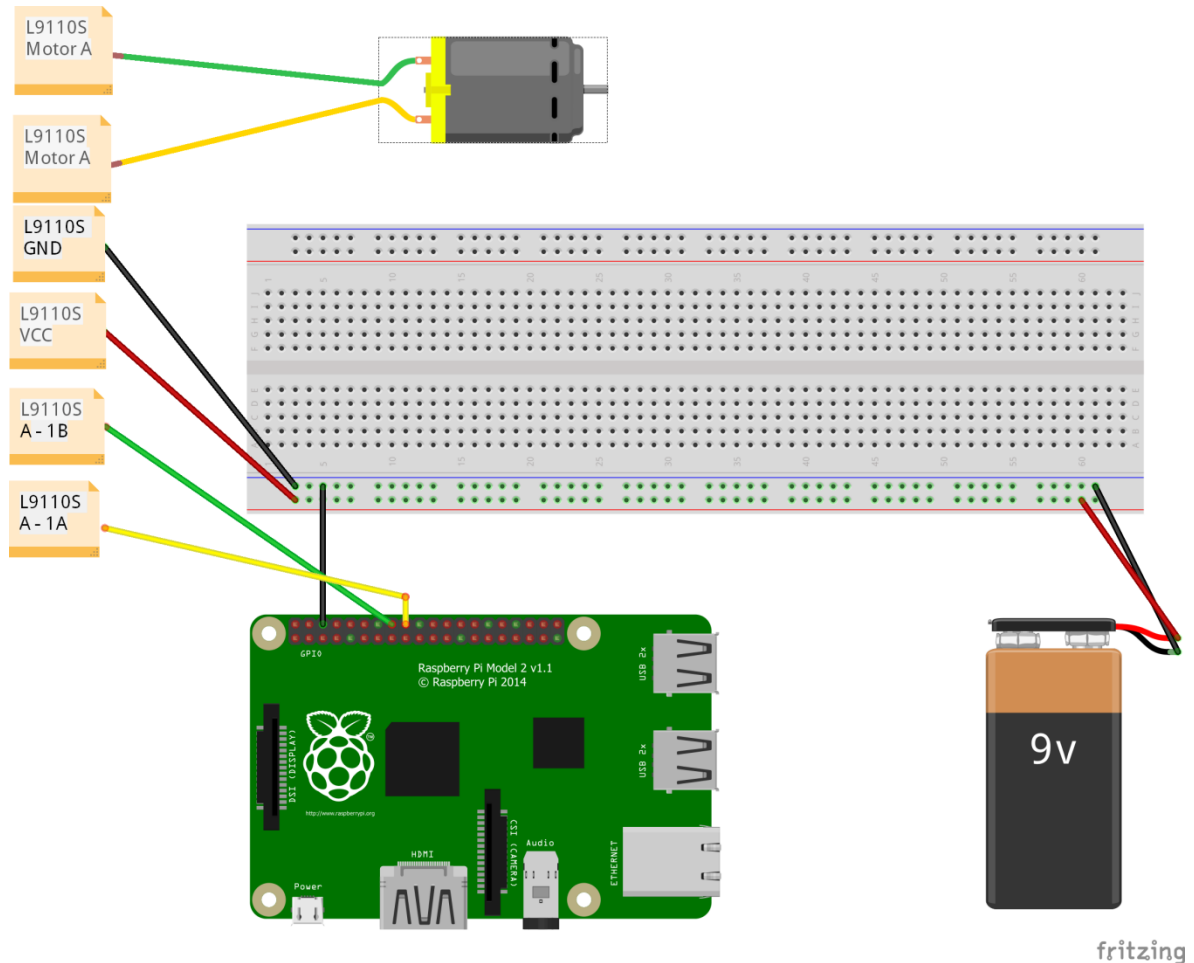
p.start(0)
p.ChangeDutyCycle(50)
sleep(5)
p.stop()

q.start(0)
q.ChangeDutyCycle(50)
sleep(5)
q.stop()
GPIO.cleanup()

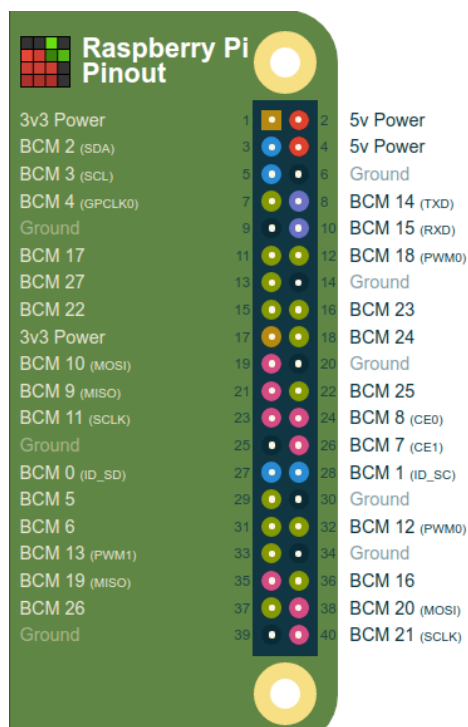
#ArmathTechCommunity
```

Կից նկարում կարող եք տեսնել միացման սխեման (հիմնականում տարածված դեղին գույնի փոքր dc շարժիչի միացման դեպքում մարտկոց անհրաժեշտ չէ)`

- 4) Ստեղծում ենք համապատասխան նիշքը Desktop-ի վրա ` անուն. py
- 5) Բացում ենք terminal-ը և հավաքում` cd Desktop և սեղմում enter հրամանը
- 6) sudo python անուն. py և սեղմում enter հրամանը



fritzing



## Սեռոբ-3 -ից Այգեստան/Rpi ոտնակների փոխարկիչ

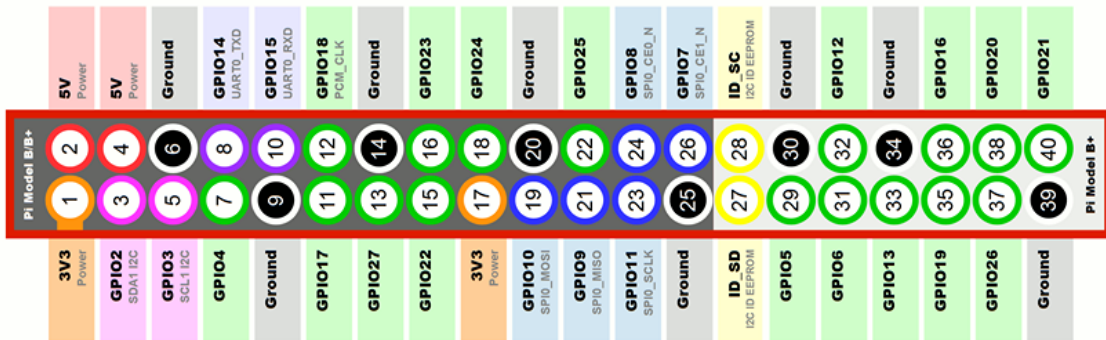
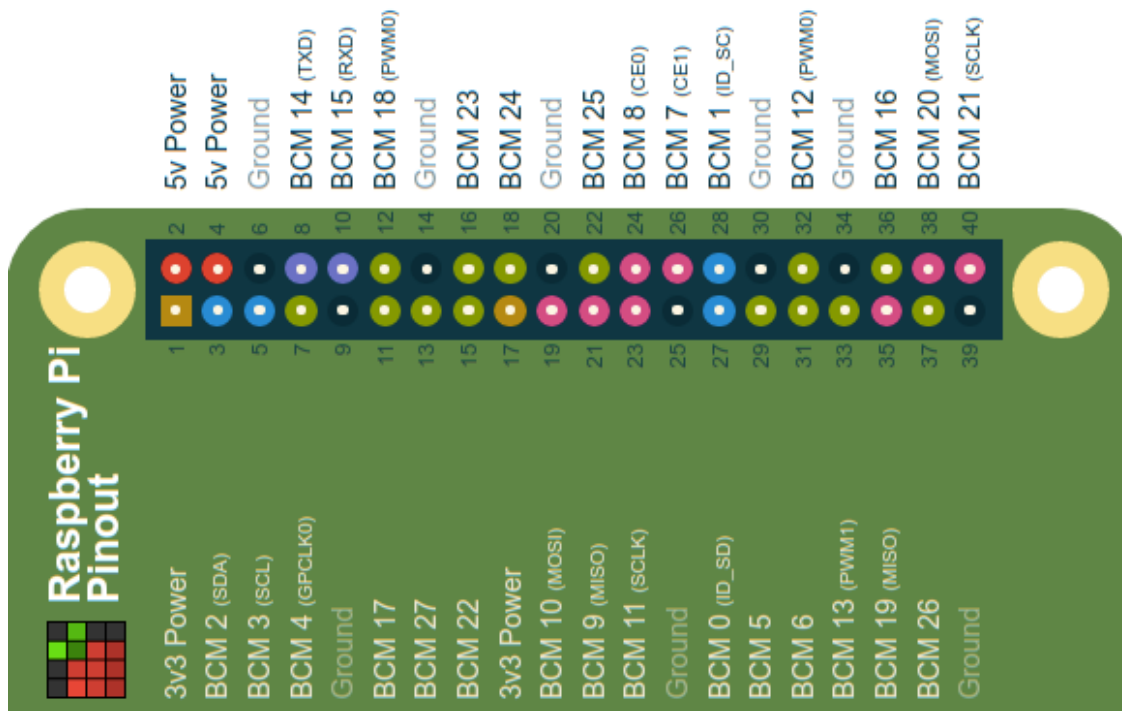
Pin numbers | Չերթական համարներով

motor-pins = 21, 22, 23, 24, 26, 27

Serob motors 0, 1, 2 3 4 5

Digital I/O (BCM) | Թվային մուտք/ելք, GPIO/BCM համարներով

'SERob':Raspberry={'0':1, '1':5, '2':6, '3':12, '4':13, '5':16, '6':19, '7':20, '8':26, '9':21}



Աղբյուր՝ pinout.xyz

## GY - 31 տեսակի գուճալի տվիչի օգտագործումը Սեռոբ3/Այգեստան-ի միջոցով

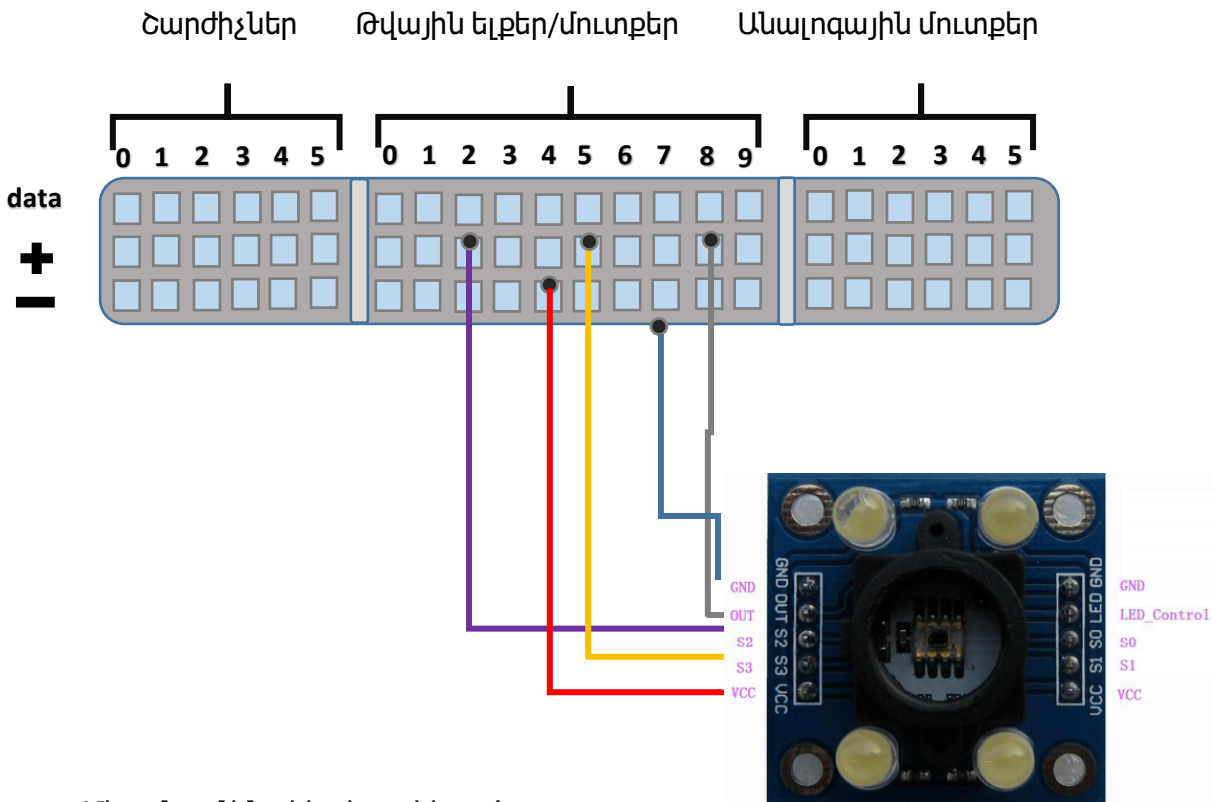
### Սեռոբ 3

Անհրաժեշտ է 5 հատ female to male jumper-եր (լարեր), 1 հատ GY-31 տեսակի գուճալի տվիչ:

Սեռոբ-ի մուտքերը/ելքերը բաժանված են հետևյալ կերպ՝

- 0-5 թվով 6 հատ շարժիչների միացման ելքեր, Motors
- 0-9 թվով 10 հատ թվային միացման մուտքեր կամ ելքեր, Digital I/O
- 0-5 հատ անալոգային մուտքեր, A In

Սեռոբ controller-ի բոլոր մուտքերը կամ ելքերը բաղկացած են 3 անցքերով հետևյալ կերպ՝



Միացնում ենք հետևյալ կերպ՝

- GND - ն ցանկացած - ոտնակի
- VCC-ն ցանկացած + ոտնակի,
- OUT-ը 9-րդ data (տվյալներ) ոտնակին,
- S2-ը 2-րդ data ոտնակին,
- S3-ը 6-րդ data ոտնակին,

Սեռոբ-ի միջավայրում կատարում ենք հետևյալ քայլերը՝

- Ստեղծում ենք համապատասխան կիչքը Desktop-ի վրա՝ անուն.py
  - Բացում ենք terminal-ը և հավաքում՝ cd Desktop և սեղմում enter հրամանը
  - sudo python անուն.py և սեղմում enter հրամանը

### GY - 31Color sensor (գուՆային տվիչ)

```
#!/usr/bin/python
```

```
# Ներմուծել class-ներ SERob3 փաթեթից/գրադարանից
from SERob3.Color import ColorSensor
```

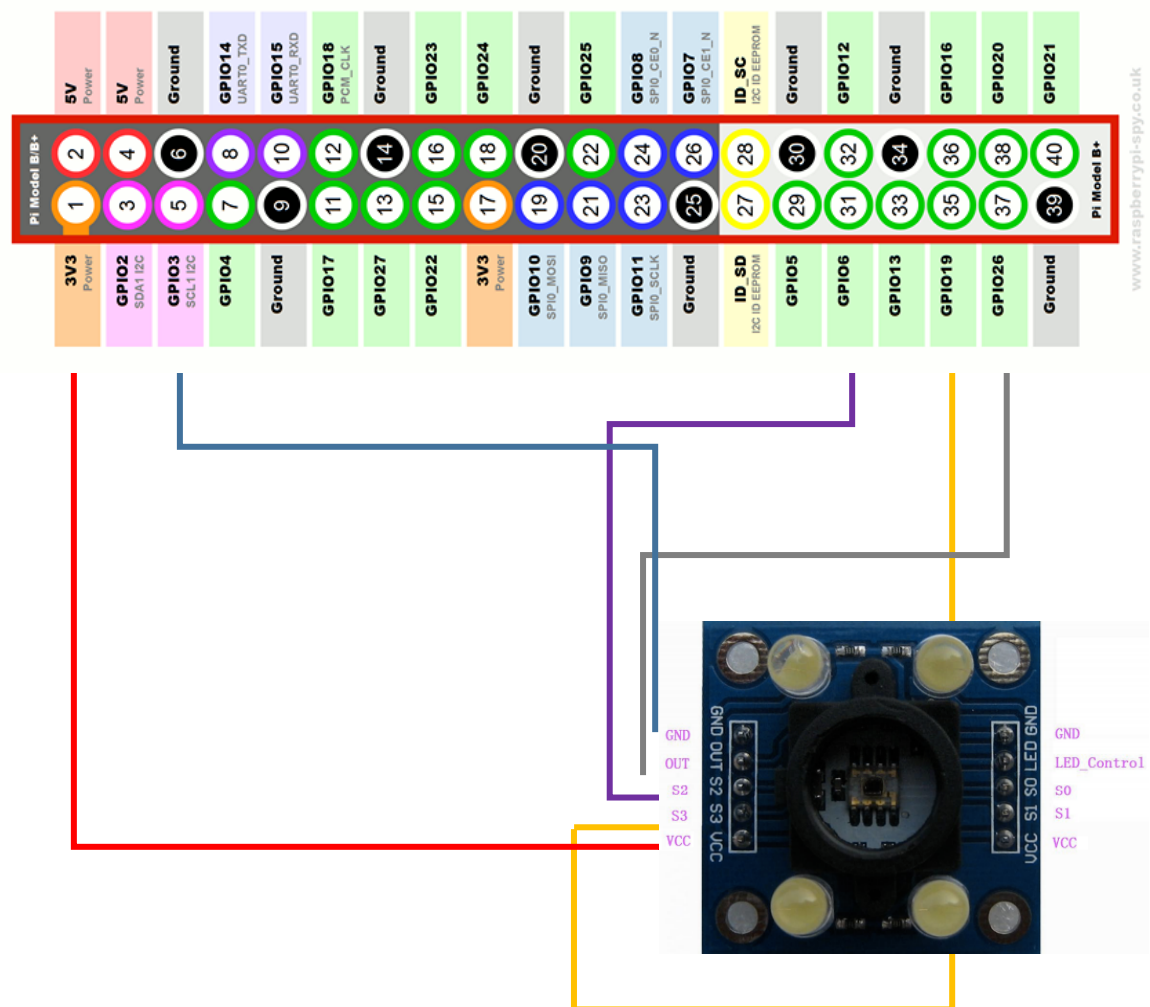
```
# Ստեղծել օբյեկտ գուՆի class-ի համար և կանչել վերջինիս մեթոդը
```

```
col_sens = ColorSensor(5, 16, 26)
```

```
# Կարդալ RGB գուՆային համակարգով SERob3-ի GPIO 6 and 19 and 26 ոտնակների միջոցով
```

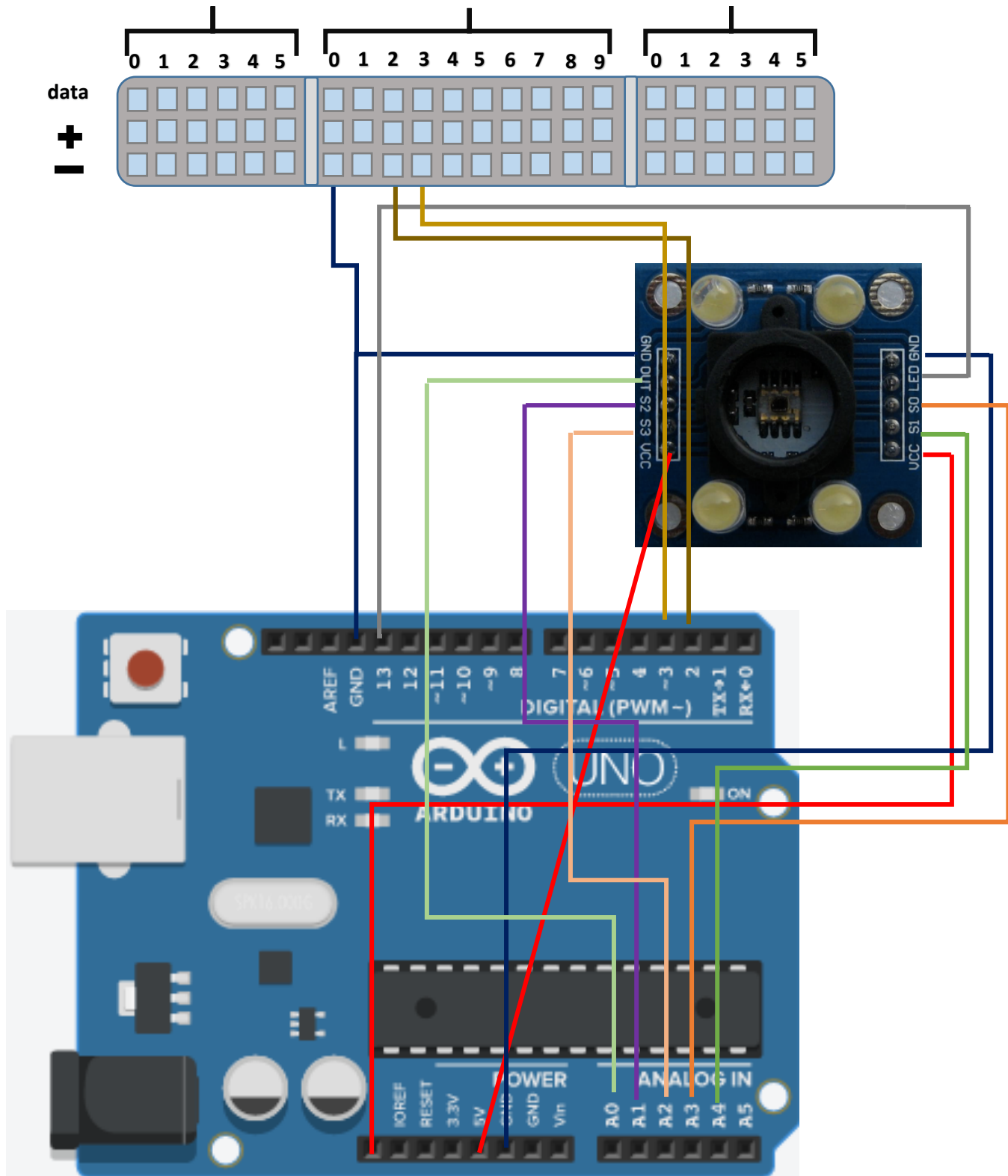
```
Col_sens.start()
```

### ԱՅԳԵՍԱՆ/RPi



**GY-31 գուՆային տվիչից ինֆորմացիա ուղարկել դեպի Սեռոբ 3 օգտագործելով «ԱրդուՆո» սալիկ**

-Միացումները s0 - A3; s1 - A4; s2 - A1; s3 - A2; out - A0; Սերոի D i/o մուկեր 2, 3 (data-ը) միանում է արդուինոյի համապատասխան 2,3-ին



Օգտագործում ենք «Արդուինո Ունո» կամ Նաիրի սալիկ և Arduino IDE միջավայրում գրում ենք հետևյալ ծրագիրը`



```
const int s0 = A3;
const int s1 = A4;
const int s2 = A1;
const int s3 = A2;
const int out = A0;
// ԳՈՒՅՆԵՐԻ ԵԼՔԱՅԻՆ ՈՏՆԱԿՆԵՐԸ ՍԵՌՈՒՐԻՆ ՄԻԱՆԱԼՈՒՑ
int redOut=2;
int greenOut=3;
int blueOut=4;
// ԿԱՐԴՈՒՄ Է ՏՎԻՉԻ ԱՐԺԵՔՆԵՐԸ
int red = 0;
int green = 0;
int blue = 0;
//
void setup()
{
    Serial.begin(9600);

    pinMode(redOut, OUTPUT);
    pinMode(greenOut, OUTPUT);
    pinMode(blueOut, OUTPUT);
    //ԶՐՈԱՑՆՈՒՄ ԵՆՔ ԳՈՒՅՆԵՐԻ ԵԼՔԱՅԻՆ ԱՐԺԵՔՆԵՐԸ
    digitalWrite(greenOut,LOW);
    digitalWrite(redOut,LOW);
    digitalWrite(blueOut,LOW);

    pinMode(s0, OUTPUT);
    pinMode(s1, OUTPUT);
    pinMode(s2, OUTPUT);
    pinMode(s3, OUTPUT);
    pinMode(out, INPUT);
    pinMode(A5, OUTPUT);
    //Միացնում ենք s0 և s1 ոտնակները
    digitalWrite(s0, HIGH);
    digitalWrite(s1, HIGH);
}

void loop()
{
    color();
    delay(50);
    //Տպում է գույների արժեքը(տես serial monitor-ում)
    Serial.println("RED");
    Serial.println(red, DEC);
    Serial.println(" GREEN ");
    Serial.println(green, DEC);
    Serial.println(" BLUE ");
    Serial.println(blue, DEC);
}
```





```
//Ստուգում է գույների ճշմարիտ լինելը
if (red < blue && red < green && red < 20)
{
    if (red <=10 && green <=10 && blue <=10){
        Serial.println("WHILE");
    } else{
        Serial.println(" - (Red Color)");
    }

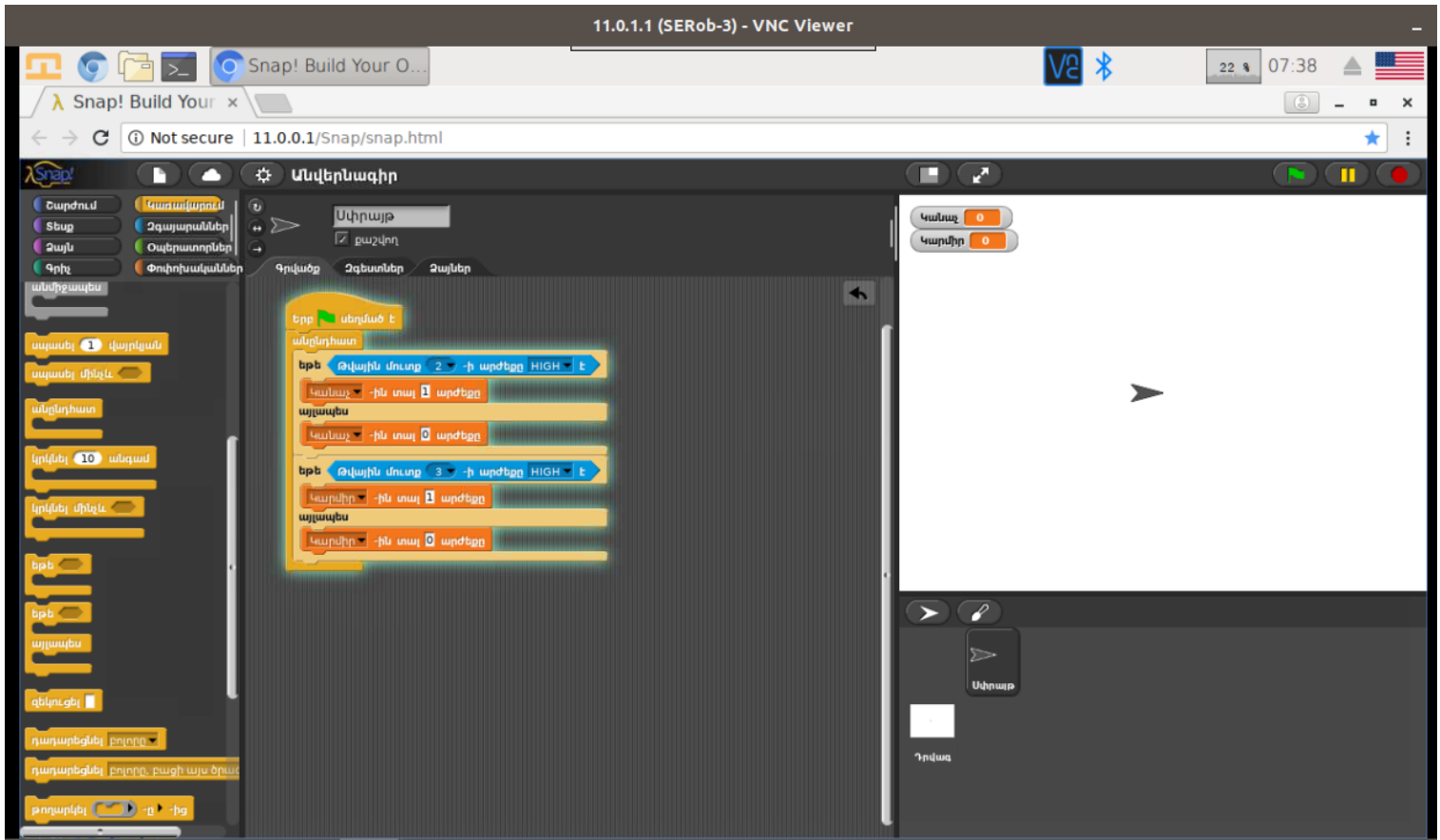
    digitalWrite(redOut,1);//analogWrite(A5,150);
}
}
else if (blue < red && blue < green)
{
    if (red <=10 && green <=10 && blue <= 10){
        Serial.println("WHILE");
    } else {
        Serial.println(" - (Blue Color)");
        digitalWrite(blueOut,1);
        //analogWrite(A5,200);
    }
}
}
else if (green < red && green < blue)
{
    if (red <= 10 && green <=10 && blue <= 10){
        Serial.println("WHILE");
    } else{
        Serial.println(" - (Green Color)");
        //analogWrite(A5,300);
        digitalWrite(greenOut,1);
    }
}
}
else{
    Serial.println();
    //analogWrite(A5,0);
    digitalWrite(greenOut,0);
digitalWrite(blueOut,0);
digitalWrite(redOut,0);
}
}
void color()
{
    digitalWrite(s2, LOW);
    digitalWrite(s3, LOW);
    red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
    digitalWrite(s3, HIGH);
    blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
    digitalWrite(s2, HIGH);
    green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}
```

Կից հղումով կարող եք օգտագործել գույնային տվիչի պատրաստի գրադարանը, որը կարող եք ներառել Arduino IDE միջավայրում՝ [RGB\\_library](#)

#ArmathTechCommunity



## Սեռոբի snap միջավայրում`



Ստեղծում ենք փոփոխական կանաչ և կարմիր գույների համար և օգտվում թվային մուտքի պայմանից:

