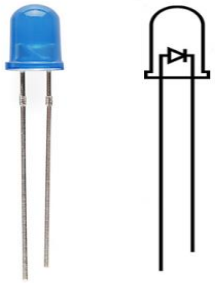
មេរៀនទី២

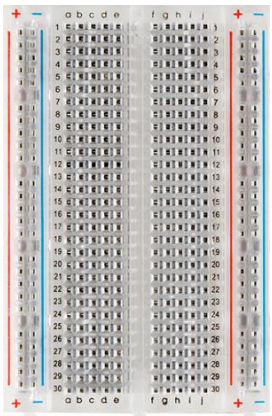
Programming in Arduino

I. BLINK LED

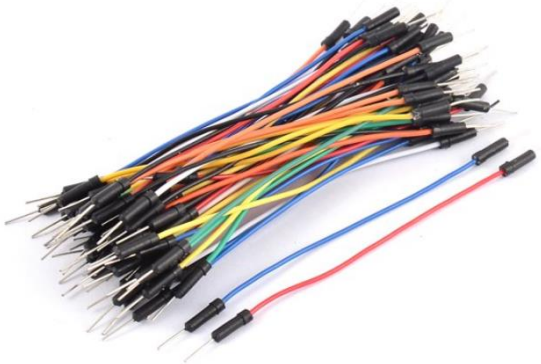
ការតភ្ជាប់សៀគ្វីតម្រូវការគ្រឿងបង្គុំ Bread-Board, LED, Resistor 220Orm Arduino Uno និងខ្សែភ្លើង



Resistor

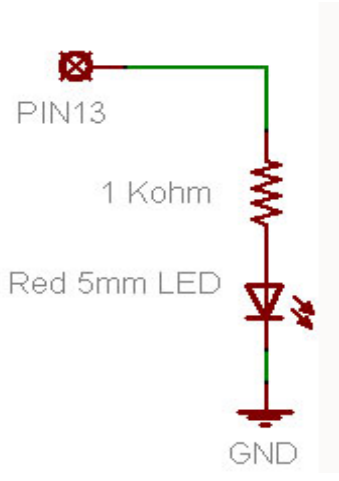


Bread Board

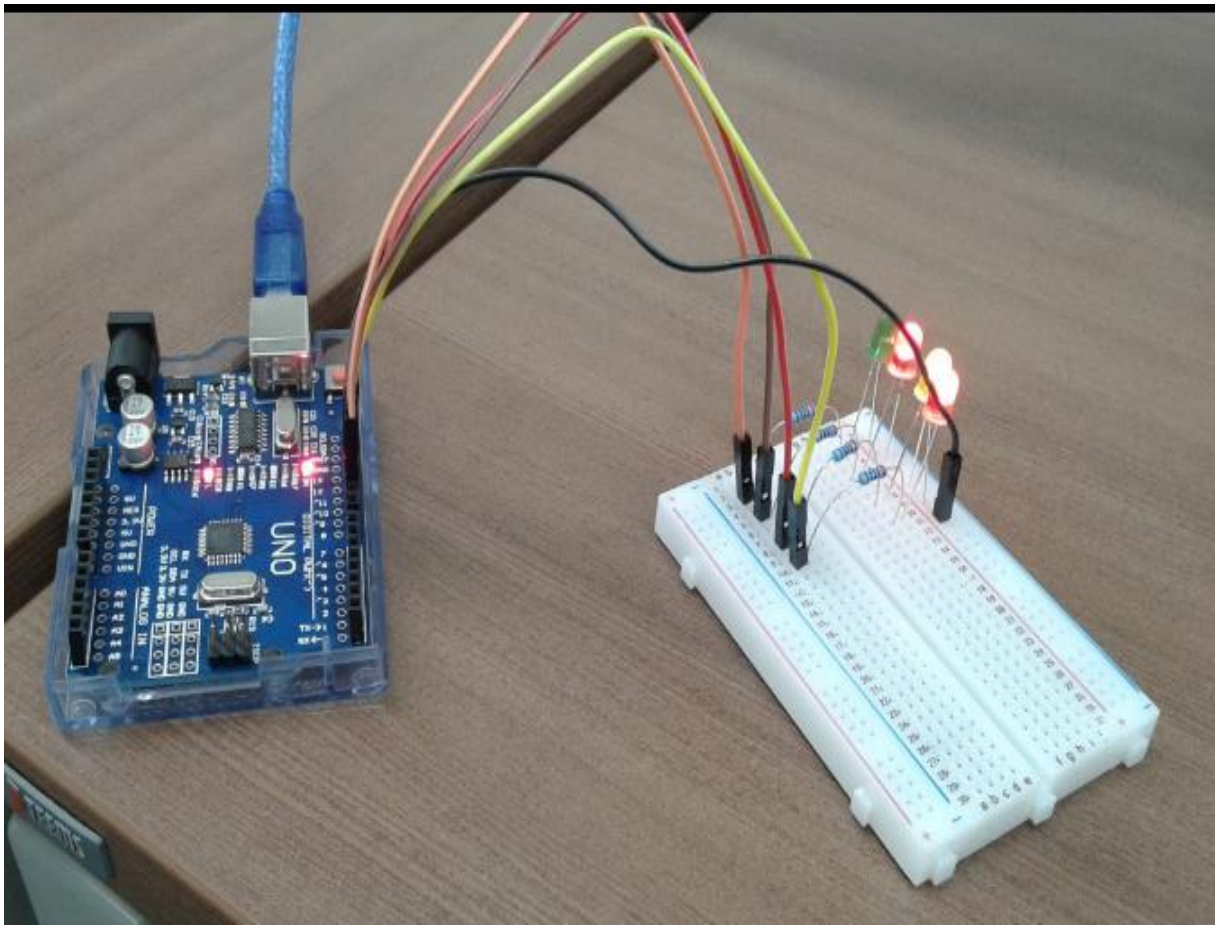


Jumper Wires

របៀបតភ្ជាប់



របៀបតភ្ជាប់ជាក់ស្តែង



ដើម្បី Write code ចូលក្នុង Board Arduino ត្រូវសរសេរ Code ដូចខាងក្រោម

```
void setup() {
        // put your setup code here, to run once:
         pinMode(13,OUTPUT);
}
void loop() {
         // put your main code here, to run repeatedly:
         digitalWrite(13,HIGH);
         delay(10000);
         digitalWrite(13,LOW);
         delay(10000);
}
```

កូដខាងលើនឹងធ្វើអោយអំពូលភ្លឺរយៈពេល១០វិនាទីរួចរលត់ទៅវិញ(ភ្លឺរលត់ៗ)

Syntax & Parameter

+pinMode(pin,mode)

-pin ជាលេខជើងដែលប្រើប្រាស់ជាមួយ Arduino Board

-mode INPUT,OUTPUT,INPUT_PULLUP & INPUT_PULLDOWN

+digitalWrite(pin,value)

-pin ជាលេខជើង Output ដែលបានតភ្ជាប់ជាមួយ Arduino

-value ជាតម្លៃ Digital ដែលត្រូវបញ្ចេញទៅពិត(HIGH : 1) ឬមិនពិត(LOW: 0)

+delay(ms)

-ms រយៈពេលគិតជាមីលីវិនាទី

+delayMircroseconds(us)

-us រយៈពេលគិតជាមិក្រូវិនាទី

a. if (conditional)and ==,!=,<,>

if ត្រូវបានគេប្រើប្រាស់នៅក្នុងលក្ខខណ្ឌដែលមានការប្រៀបធៀបគ្នាពីរ ឬច្រើនជាងនឹង។

Syntax : if (variable < *20)*

```
 {
        //do something
  }
```

Example

```
if (x<20)
{
digitalWrite(led,HIGH);
}
if (x<20) digitalWrite(led,HIGH);
if(x<20){digitalWrite(led,HIGH);}
```

ការសរសេរទាំងបីរបៀបនេះអ្នកអាចជ្រើរើសការសរសេរតាមរបៀបមួយណាក៏បាន។

b. if / else

else ត្រូវបានគេប្រើប្រាស់ជាមួយ if ដែលវាដំណើរការបានដែលព្រឹត្តិការណ៍នៅក្នុង if មិនពិតនោះវានឹងរំលងកូននៅក្នុង if ហើយចូលទៅដំណើរការកូដនៅក្នុង else វិញ។

Syntax

```
if (condition)
{
        // do something(statement)
}else if (condition)
{
                // do something(statement)
        }else
        {
                //do something(statement)
        }
```

C. FOR STATEMENT

វាជាកន្សោមការងារមួយដែលដំណើរការដដែលៗ ប៉ុន្តែស្ថិតនៅក្នុងចំនួនកំណត់ជាក់លាក់មួយ។

Syntax

```
For (initialization, condition, increment)
{
        //do something(statement)
}
```

Example

```
for (int x=0; x<=100; x++)
{
        println(x+1);
}
```

D. SWITCH /CASE STATEMENT

ដំណើរការរបស់ Switch/case statement វាមានភាពស្រដៀងគ្នាទៅនឹង if ដែរខុសគ្នាត្រង់ថាវាយកតម្លៃនៃអថេរនោះ មានតម្លៃដូចគ្នាទៅនឹងតម្លៃដែលបានកំណត់ជាមួយវា នឹងប្រតិបត្តិការកូដនៅកនុងចំណុចនោះ។ នៅពេលដែលគេចង់អោយប្រតិបត្តិការកូដត្រង់ចំណុចណាមួយបញ្ចប់គេត្រូវប្រើពាក្យគន្លឹះ Break (ចុងបញ្ចប់កូដ)។

Syntax

```
Switch(variable)
{
        case value:
                //do something(statement)
                break;
        case value
                //do something(statement)
                break;
        case value
                //do something(statement)
                break;
        .
        .
        default:
        //do something(statement)
        break;

}
```

Example

```
int x=10;
switch(x)
{
        case 2:
                println(“Hello”);
                break;
        case 7:
                println(“No”);
                break;
        case 10:
                println(“Yes”);
                break;
        default:
                println(“OK”);
                break;
}
```

E. WHILE LOOPS

while loops ជាការកំណត់លក្ខខណ្ឌមួយដែលដំណើរការដដែលៗនៅពេលដែល Expression របស់វាពិត នៅពេលដែល Expression របស់វាមិនពិតវានឹងបញ្ឈប់ដំណើរការនោះភ្លាម។

Syntax:

```
while(expression)
{
    //do something(statement)
}
```

Example

```
x=0;
while(x<100)
{
    //do something(statement)
    x++;
}
```

F. DO WHILE

ដំណើរការស្រដៀងគ្នាទៅនឹង While Loops ដែរខុសគ្នាត្រង់ថាវាដំណើរការរួចហើយទើបទៅត្រួតពិនិត្យលក្ខខណ្ឌតាមក្រោយ។

```
do
{
    //do something(statement)
}while(condition);
```

Example

```
int x=0;
do
{
    delay(50);
    //do something(statement)
}while(x<12);
```

អនុវត្តន៍ ជ្រើសរើសអំពូល LED 5 គ្រាប់មកតភ្ជាប់ជាមួយ Arduino UNO ដែលត្រូវដាក់ជាជួរៗអ្នកភ្ជាប់ដូចខាងក្រោម

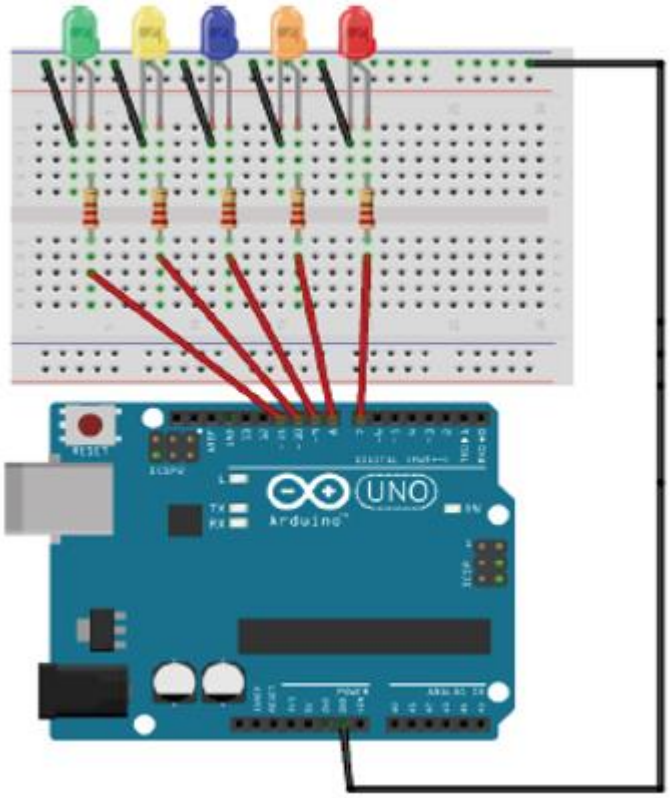
-អំពូលទី១ ទៅជើងទី៧របស់ Arduino

-អំពូលទី២ ទៅជើងទី៨របស់ Arduino

-អំពូលទី៣ ទៅជើងទី៩របស់ Arduino

-អំពូលទី៤ ទៅជើងទី១០របស់ Arduino

-អំពូលទី៥ ទៅជើងទី១១របស់ Arduino



បន្ទាប់មកទៀតអ្នកត្រូវសរសេរកូដអោយអំពូលទាំង ៥ នោះភ្លឺប្រដេញគ្នាពីឆ្វេងទៅស្តាំរួចពីស្តាំមកឆ្វេងវិញដែលវានឹងដំណើរការដដែលនៅក្នុង Loop មួយ។

Code

```
void setup() {
  // put your setup code here, to run once:
            for (int i=7;i<=11;i++)
            {
              pinMode(i,OUTPUT);
            }
}

void loop() {
  // put your main code here, to run repeatedly:
  for (int i=7;i<=11;i++)
  {
           if(i>7)
          {
            digitalWrite(i-1,LOW);
          }
          digitalWrite(i,HIGH);
          delay(1000);
  }
  for (int i=11;i>=7;i--)
  {
              if(i<11)
              {
                digitalWrite(i-1,LOW);
```

```
                }
                digitalWrite(i,HIGH);
                delay(1000);
            }
        }
```

II . DIGITAL INPUT

Digital Input ជារបៀបដែល Arduino Board ទទួលទិន្នន័យប្រភេទ Digital តារយៈជើង Digital របស់វា។Arduino មានច្រើនប្រភេទនិងមានជើង Digital ទៅតាមជើងរបស់Board មួយៗ របស់Arduino។យើងនឹងលើកយក Arduino UNO យកមកនិយាយ។Arduino UNOមានជើង Digital input ចំនួន១៤ជើង(0 ដល់ ១៣)។

III. DEVICE

1.SERIAL ADAPTOR

Serial Adaptorគឺជាខ្សែមួយសម្រាប់តភ្ជាប់ពីរ Arduino មក Computer។ពេលយើងធ្វើការបញ្ជូនព័ត៌មានពី computer ចូលទៅក្នុង Arduino និងពីរ Arduino ទៅកុំព្យូទ័រវិញ។តាមរយៈ Serial Adaptor ។

https://www.pngkit.com/view/u2w7w7o0o0u2r5t4_arduino-usb-cable-usb-serial-cable-arduino/

Function of Serial

-If(Serial)
-available()
-begin()
-end()
-find()
-print()
-println()
-readString()
-setTimeout()
-write()

Function Begin

សម្រាប់បើក Serial port *របស់* Arduino ដំណើរការជាមួយ Baud rate ណាមួយ។

Syntax:

```
Serial.begin(Baudrate);
```

Example

```
void setup() {
    Serial.begin(9600);
}
```

Function Available

*សម្រាប់តេស្តមើលថាមានទិន្នន័យនៅក្នុង*Buffer *មុននិងអានយកទិន្នន័យនោះ។*

```
Example:
void setup() {
  Serial.begin(9600);
}
void loop() {
  if (Serial.available()>0){
  int k=Serial.read();
 }}
```

Function Read()

សម្រាប់ទាញយកទិន្នន័យ 1byte ពីBuffer។ទិន្នន័យដែលទទួលបានជាតម្លៃលេខ។

Example:

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  if (Serial.available()){
  int k=Serial.read();
 }}
```

Function ReadString()

សម្រាប់ទាញទិន្នន័យទាំងអស់ក្នុង Buffer រួចបំលែងទិន្នន័យជាពាក្យ។

Example

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  if (Serial.available()>0){
  String k=Serial.readString();
 }}
```

Function setTimeout()

កំណត់រយៈពេលសម្រាប់*Arduino* ក្នុងការធ្វើការប្រតិបត្តិការអានទិន្នន័យប្រសិនជាវាមិនអាចអានក្នុងរយៈពេលនោះអោយអស់ទេវាងនិងបាត់បង់ទិន្នន័យ។

```
void setup() {
  Serial.begin(9600);
   Serial.setTimeout(100);
}
void loop() {
    if (Serial.available()){
        String k=Serial.readString();
    }
}
```

Function Write()

សម្រាប់បញ្ជូនទិន្នន័យ 1byte ចេញពីArduino ទៅក្រៅ

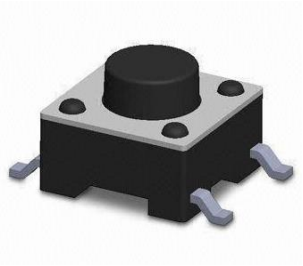
Example

```
void setup() {
        Serial.begin(9600);
}
void loop() {

        Serial.write(x);
}
```

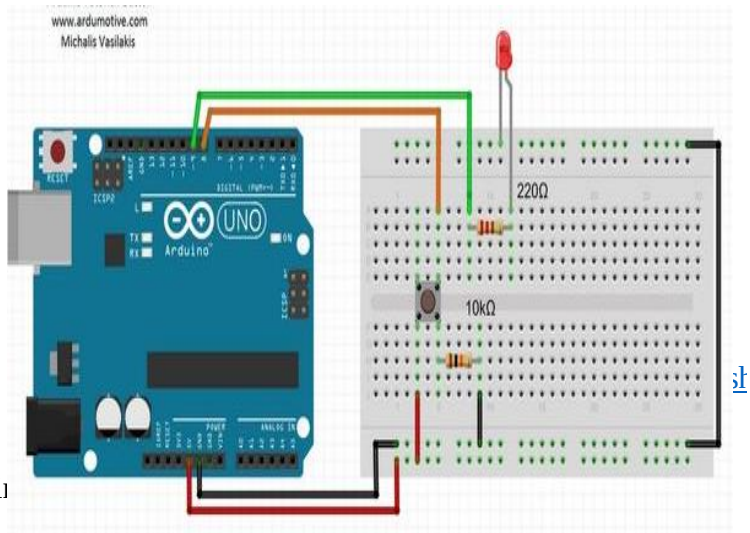
2. PUSHBUTTON

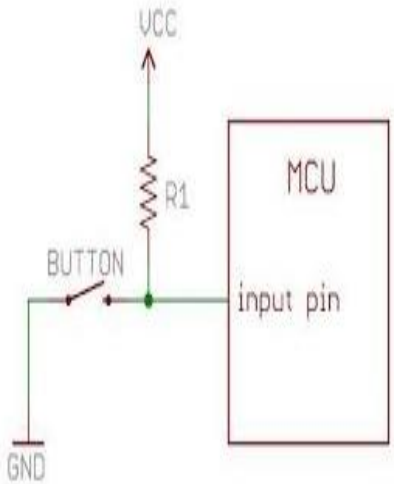
pushButton មានតួនាទីសម្រាប់ផ្តាច់ចរន្ត និងភ្ជាប់ចរន្ត។



https://www.sparkfun.com/products/97

ការតភ្ជាប់






```
void setup() {
  // put your setup code here, to run once:
 // pinMode(13,OUTPUT);
  pinMode(BTN_PIN,INPUT_PULLUP);
  Serial.begin(9600);
}
```

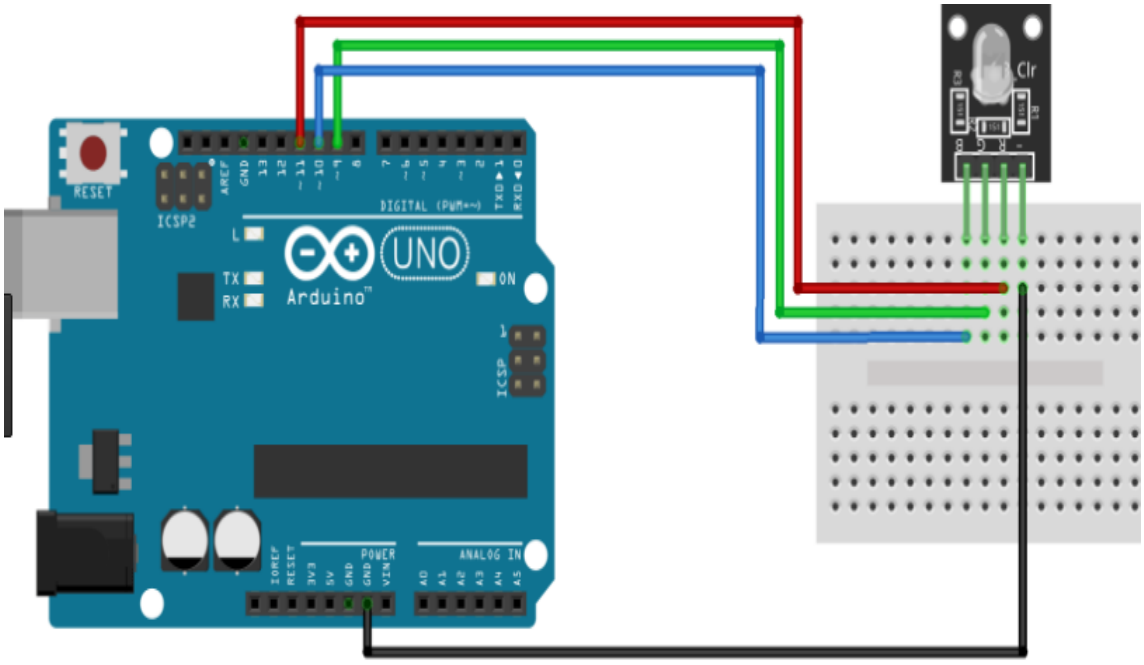
```
void loop() {
        // put your main code here, to run repeatedly:
         Serial.println(digitalRead(BTN_PIN));
         delay(1000);
       }
```

3. RGB

ចូរសរសេរកូដបញ្ជាទៅលើ RGB LED ដើម្បីអោយចេញពណ៌បានច្រើនពណ៌។

https://duino4projects.com/rgb-color-detector-using-tcs3200-sensor-module/

ការតភ្ជាប់



Coding

```
int BTN_PIN_R=8;//pin Red
int BTN_PIN_G=9;//pin Green
int BTN_PIN_B=10;//Pin Blue
int speckerpin=13;
void setup() {
        // put your setup code here, to run once:
```

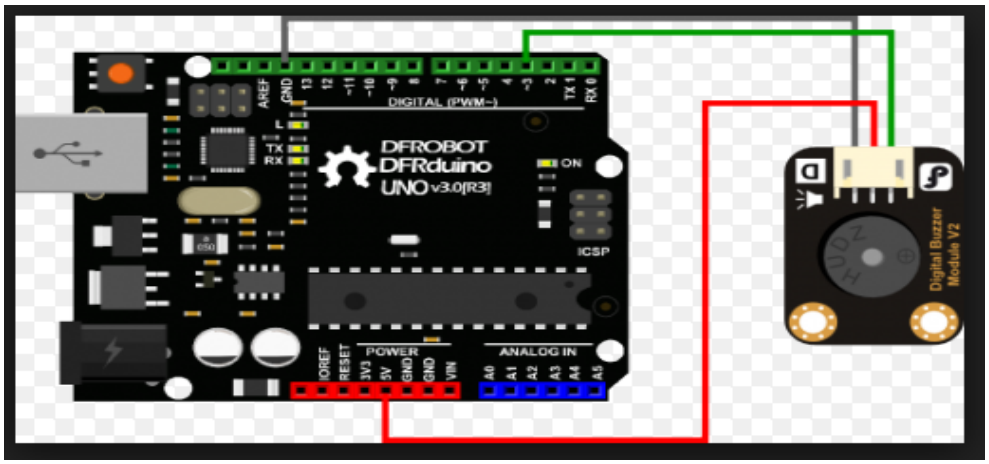
```
        pinMode(BTN_PIN_R,INPUT);
        pinMode(BTN_PIN_G,INPUT);
        pinMode(BTN_PIN_B,INPUT);
}
void loop()
{
        setColor(255, 0, 0); // red
        delay(1000);
        setColor(0, 255, 0); // green
        delay(1000);
        setClolor(23,44,56);//red 23,green 44,blu56
}
void setColor(int red, int green, int blue)
{
        analogWrite(BTN_PIN_R, red);
        analogWrite(BTN_PIN_G, green);
        analogWrite(BTN_PIN_B, blue);
}
```

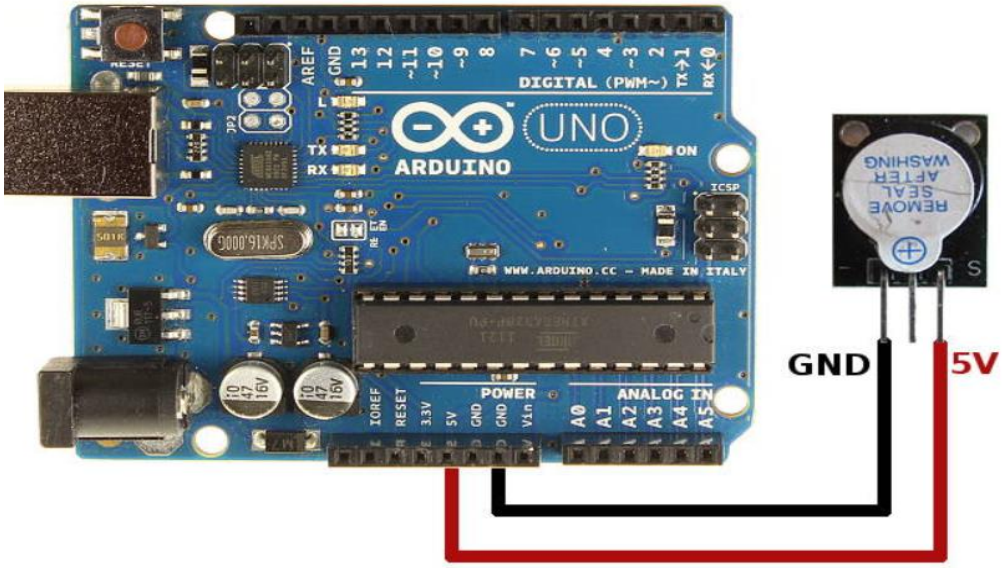
4. Digital buzzer module(speaker)

buzzer module ជាឧបករណ៍ប្រើសម្រាប់បង្ហាញបន្លឺសម្លេង។

https://www.fabtolab.com/active-buzzer-module

ការតភ្ជាប់





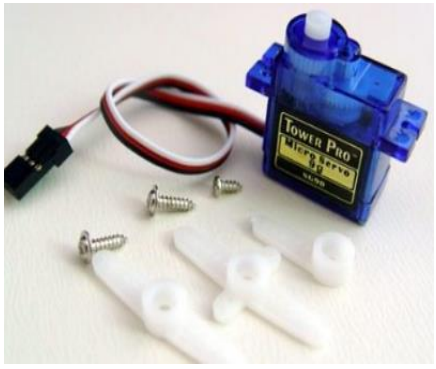
https://startingelectronics.org/tutorials/arduino/modules/active-buzzer/

Coding

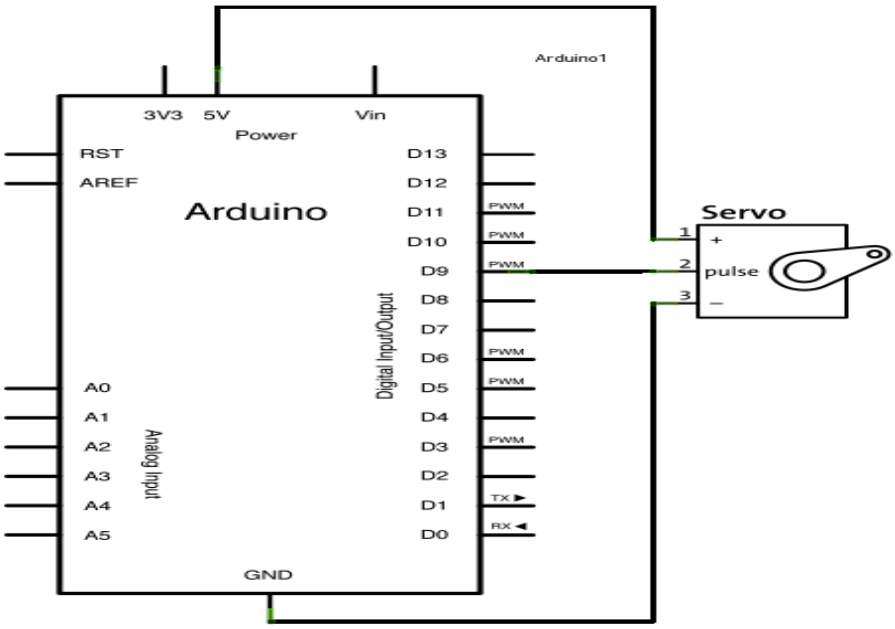
```
int speckerpin=10; //pin digital
void setup() {
}
void loop() {
        // put your main code here, to run repeatedly:
        tone(speckerpin,2,1000);
        delay(1000);
}
```

5. Micro Servo(Motor)

Micro Servo ជាប្រភេទឧបករណ៍មួយប្រភេទដែលជាម៉ូទ័រសម្រាប់វិលបានមុំ ០ទៅ១៨០ដឺក្រេ។



ការតភ្ជាប់



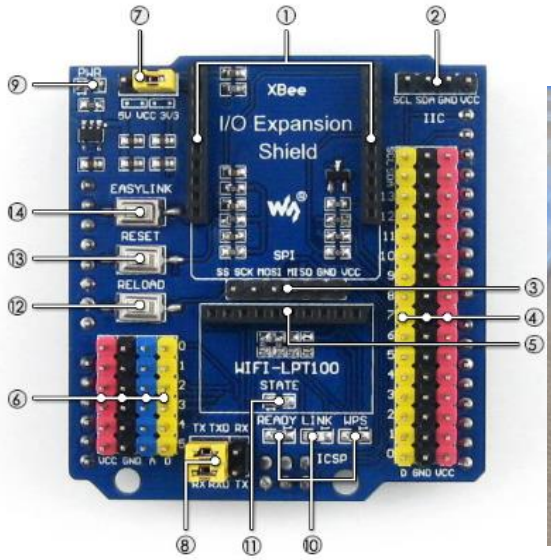
https://create.arduino.cc/projecthub/krivanja/working-with-a-micro-servo-86ec6b

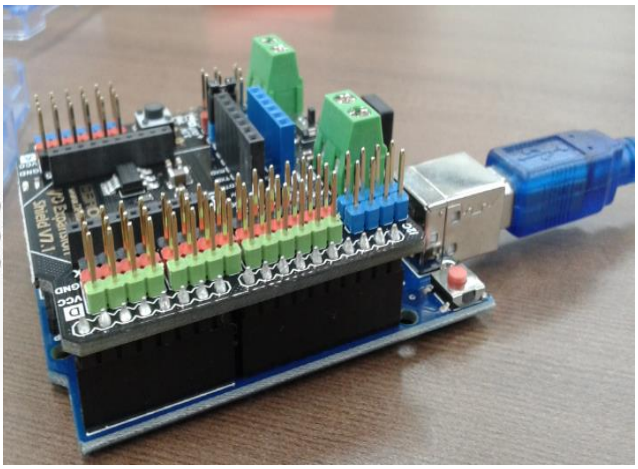
Coding

```
#include<Servo.h>//you want to see Servo Libraries you connect part
                        (C:\Users\203-01\Documents\Arduino\libraries\Servo\Servo.cpp)
Servo myServo;//create object
void setup() {
  myServo.attach(4);//pin number of connect with Arduino
  myServo.write(0);
}
void loop() {
        // put your main code here, to run repeatedly:
       for (int i=0 ;i<=180; i++)//mom 0-180
        {
              myServo.write(i);
               delay(15);
       }
       myServo.write(0);
}
```

6. I/O Expansion

I/O Expansion មានតួនាទីសម្រាប់តភ្ជាប់ពីBoard Arduino ដើម្បីអោយបានPin ច្រើន និង មានភាពងាយស្រួលជាងមុន។ហើយភ្ជាប់ជាមួយ Arduino

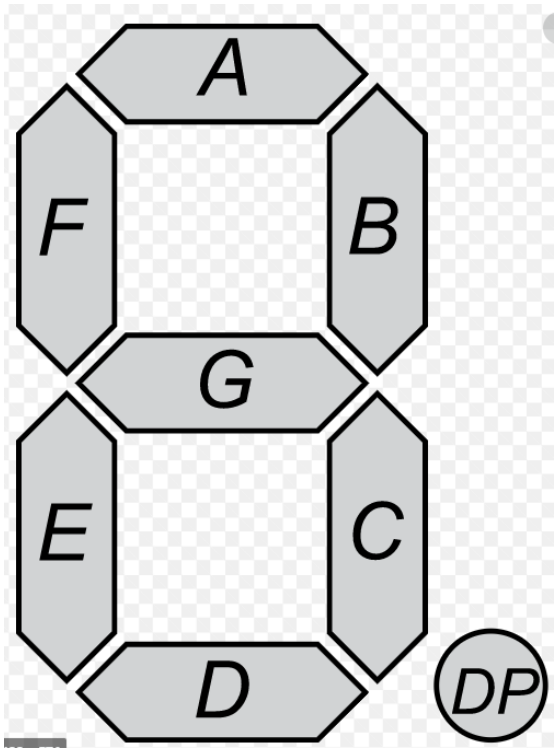


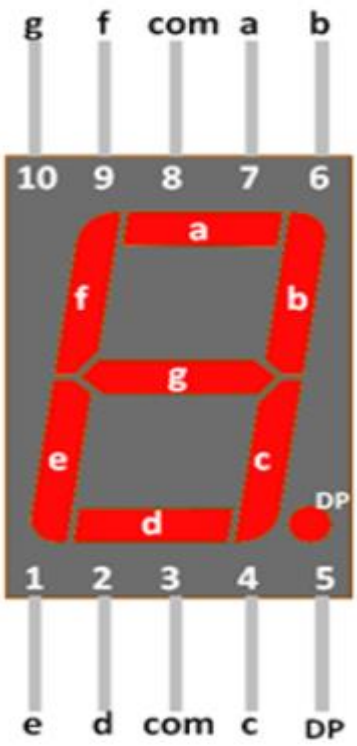


https://www.waveshare.com/io-expansion-shield.htm

7. Segment with Max 7219

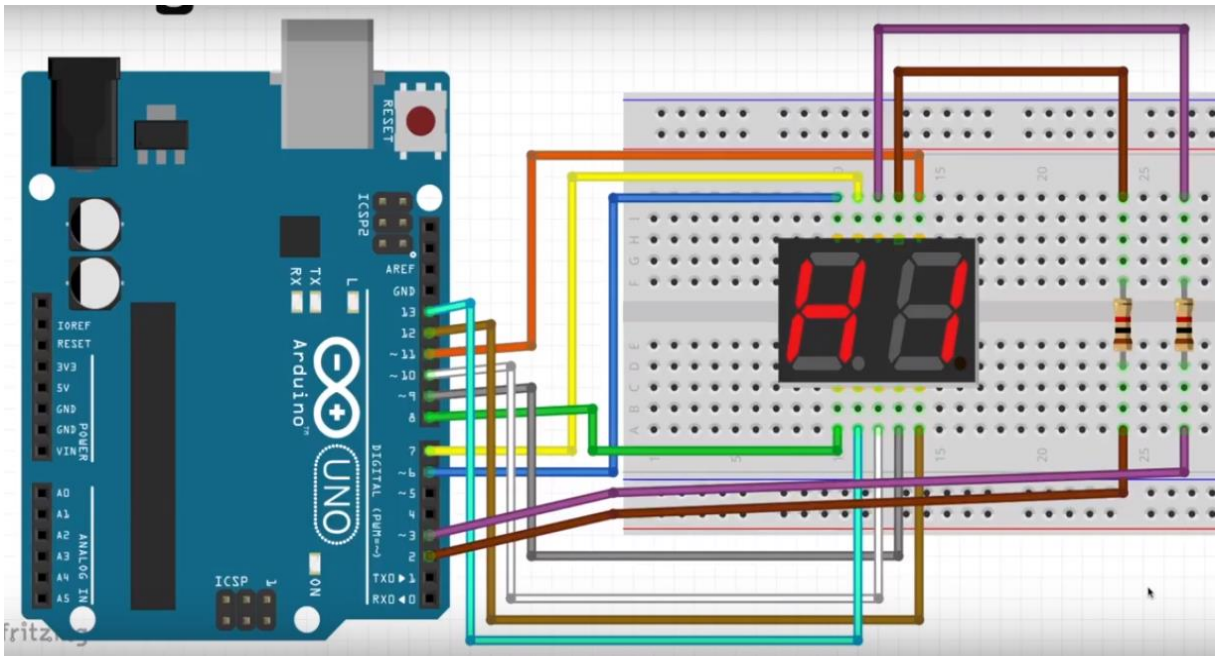
ដូចដែរយើងបានសិក្សារួចមកហើយនៅក្នុងការប្រើប្រាស់ 7 *segment* មួយយើងត្រូវការតភ្ជាប់ខ្សែជាច្រើនទៅកាន់ជើង *Arduino* ។សាកស្រមៃថាបើសិនជាយើងត្រូវការប្រើប្រាស់ 7 *segment* ចំនួន ៥ គ្រាប់តើយើងត្រូវការខ្សែប៉ុន្មានសម្រាប់ធ្វើការតភ្ជាប់លើសពីនេះទៅទៀតយើងនឹងប្រឈមមុខនូវបញ្ហាមួយគឺខ្វះខាតជើង *Arduino* នៅក្នុងការប្រើប្រាស់។ដូច្នេះទើបគេបង្កើត *IC* ម្យ៉ាងដែលមានឈ្មោះ *Max* 7129 ដែលត្រូវប្រើប្រាស់ជាមួយ ៧ *Segment* ដើម្បីដោះស្រាយបញ្ហាទាំងអស់នេះ ក៏ដូចជាសម្រួលនៅក្នុងការប្រើប្រាស់ ៧ *Segment* អោយងាយស្រួលជាងមុននៅក្នុងការសរសេរកូដ ក៏ដូចជាការតភ្ជាប់សៀគ្វីជាដើម។





https://components101.com/displays/7-segment-display-pinout-working-datasheet

ដើម្បីប្រើប្រាស់ *7 Segment with Max7219* ប្រភេទនេះបានយើងត្រូវការប្រើប្រាស់ *Library Led Control* ។



Coding

```
void setup() {
        for (int i=6;i<=13;i++)
          {
           pinMode(i,OUTPUT);
          }
          pinMode(1,OUTPUT);
          pinMode(2,OUTPUT);
          digitalWrite(2,0);
          digitalWrite(1,0);
          digitalWrite(13,1);
}
void loop() {
         // put your main code here, to run repeatedly:
        int x;
        for (int j=0;j<=9;j++)
        {
                x=Br(j);
                for (int i=6;i<13;i++)
               {
                        digitalWrite(i,bitRead(x,i-6));
               }
              delay(1000);
        }
}
```

អនុវត្តន៍ ១ ចូលសរសេរកូដអោយ 7 segment រាប់ពី0ដល់ ៩៩រួចរាប់ថយក្រោយពី៩៩មកវិញ។

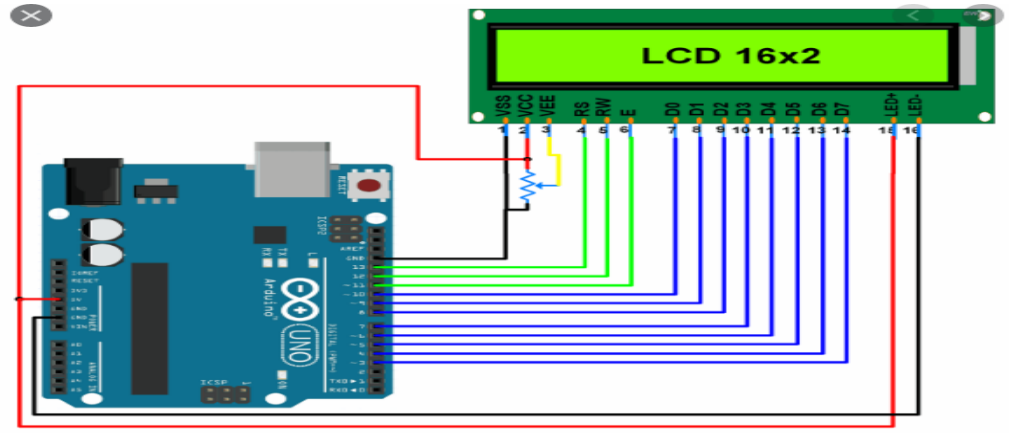
អនុវត្តន៍ ២ ចូលសរសេរកូដដែលមានដំណើរការដូចទៅនឹងភ្លើងស្តុបចរាចរណ៍។អ្នកត្រូវជ្រើសរើសអំពូល *LED* ចំនួន៣គ្រាប់ដែលមានពណ៌បៃតងលឿង និងក្រហម។

8. LCD (LIQUID CRYSTAL DISPLAY)

LCD(Liquid Crystal Display) ជាគ្រឿងអេឡិចត្រូនិកមួយប្រភេទដែលគេនិយមយកវាទៅប្រើប្រាស់សម្រាប់បង្ហាញរាល់ពត៌មានដែលបានមកពី Controller។ *LCD* មានសមត្ថភាពបង្ហាញតម្លៃលេខ និងអក្សរឡាតាំងស្ទើរតែគ្រប់អក្សរ ដែលមានភាពល្អប្រសើរជាង 7 *Segment* នៅក្នុងការបង្ហាញអក្សរឡាតាំង។LCD ដែលយើងនឹងលើកមកសិក្សាគឺជាប្រភេទ LCD *16 * 2* វាមានន័យថាវាអាចបង្ហាញទិន្នន័យចំនួន២ជួរដាច់ពីគ្នាដែលនៅក្នុងមួយជួរមានចំនួន Colum ចំនួន១៦។ ដោយសារតែយើងបានប្រើប្រាស់នៅ module បន្ថែមដើម្បីងាយស្រួលក្នុងការតភ្ជាប់ ដូច្នេះ យើងត្រូវទាញយក Library មួយយកមកប្រើប្រាស់

Pin	Name	Function
1	Anode	Backlight Anode (+)
2	Cathode	Backlight Cathode (-)
3	VSS	0V (GND)
4	VDD	+5V
5	V0	Contrast
6	RS	Register select
7	R/W	Read / Write
8	E	Enable
9	DB0	Data bit 0
10	DB1	Data bit 1
11	DB2	Data bit 2
12	DB3	Data bit 3
13	DB4	Data bit 4
14	DB5	Data bit 5
15	DB6	Data bit 6
16	DB7	Data bit 7

https://startingelectronics.org/beginners/components/LCD/



https://circuits4you.com/2016/05/15/how-to-lcd-display-arduino-uno/

Function ក្នុង object LiquidCrystal

-Blink() (សម្រាប់រក្សាទិន្នន័យអោយនៅ)

-setCursor(Column,Row) (សម្រាប់កំណត់ Cursor នៅទីតាំងណាមួយ)

-print(“”)(សម្រាប់បង្ហាញចេញទិន្នន័យក្នុង Screen)

- autoscroll() (សម្រាប់ទាញទិន្នន័យ)

-noAutoscroll()(សម្រាប់បញ្ឈប់ទាញទិន្នន័យ)

-clear()(សម្រាប់លប់ទិន្នន័យចេញពីScreen)

-scrollDisplayLeft()(សម្រាប់ទាញទិន្នន័យទៅខាងឆ្វេង)

-scrollDisplayRight() (សម្រាប់ទាញទិន្នន័យ
ទៅខាងស្តាំ)

-Write(“”)(សម្រាប់បង្ហាញទិន្នន័យក្នុងScreen)

-Home() (សម្រាប់ Reset)

-rightToLeft() go right for the next letter
-leftToRight() go left for the next letter

កូដ(សម្រាប់ Check Address ដែល Protocol I2C បានតភ្ជាប់អ្នកមិនចាំបាច់ប្រើប្រាស់កូដនេះក៏បាន)

```
#include<Wire.h>
void setup()
{
        Write.begin();
        Serial.begin(9600);
        while(!Serial);
        Serial.println(“\nI2C Scanner”);
}


Void Loop()
{
        byte error,address;
        int nDevices;
        Serial.println(“Scanning…”);
                nDevices=0;
                for (address =1; address<127;address++)
                {
            Wire.beginTransmission(address);
            error=Wire.endTransmission();
            if (error==0)
            {
                    Serial.print(“I2C device found at address Ox”);
                    if (address<16)
                    Serial.print(“0”);
                    Serial.print(address,HEX);
                    Serial.print(“  !”);
                    nDevices++;
            }
            else if(error==4)
            {
                    Serial.print(“unknown error at address Ox”);
                    if (address<16)
                    Serial.print(“0”);
                    Serial.println(address,HEX);
            }
            if (nDevices==0)
```

```
                Serial.println("No I2C devices found\n");
            else
                Serial.println("done\n");
            delay(5000);
        }
```

កូដ(សម្រាប់តេស្តបញ្ជាក់ពីការតភ្ជាប់ ក៏ដូចជាការបង្ហាញពាក្យនៅលើ LED)

```
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
LiguidCrystal_I2C lcd(0x27,16,2);
void setup()
{
    lcd.init();
    lcd.backlight();
}
void loop()
{
    lcd.setCursor(3,0);
    lcd.print("Automation");
    lcd.setCursor(4,1);
    lcd.print("Cambodia");
}
```

មុខងារមួយចំនួនដែលគួរតែយល់ដឹង

-liquidCrystal()
-begin()
-clear()
-home()
-setCursor()
-write()
-print()
-cusor()
-noCursor()
-blink()
-noBlink()
-display()
-nodisplay()
-scrollDisplayLeft()
-scrollDisplayRight()
-autoscroll()
-noAutoscroll()
-leftToRight()
-rightToLeft()
-createChar()

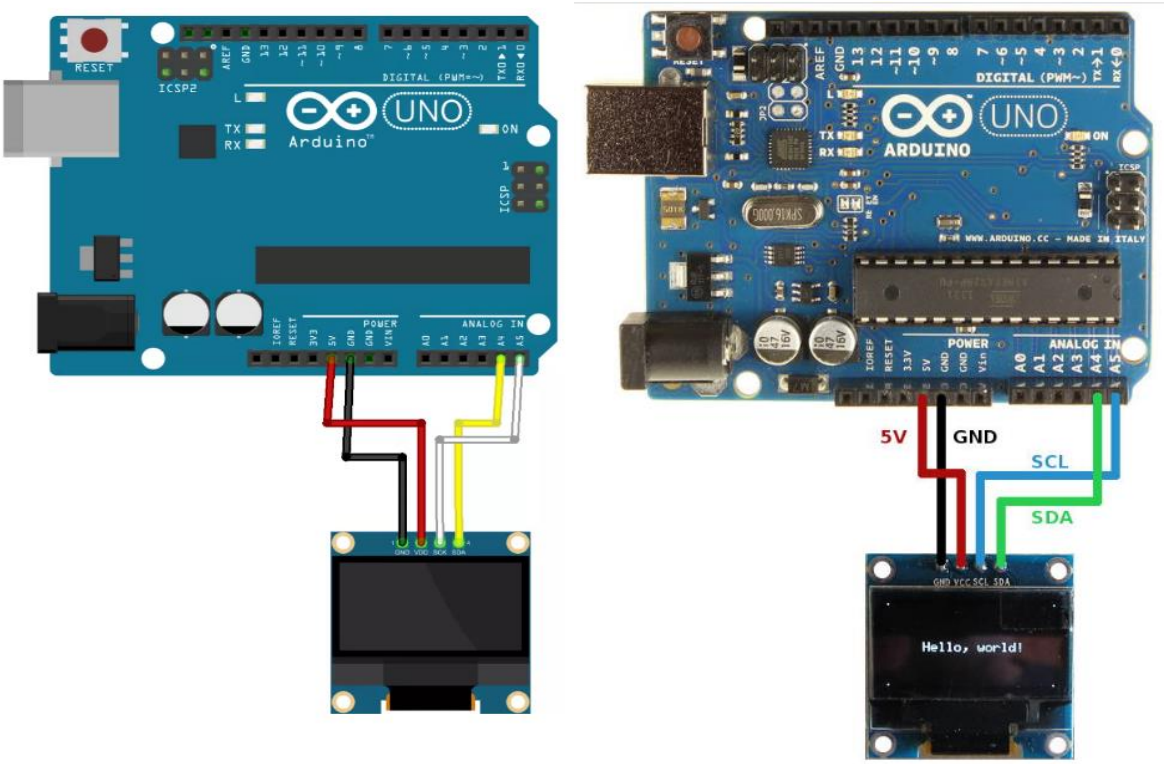
សម្រាប់ពត៌មានសម្អិតអាចចូលទៅកាន់ Link នេះ

https://www.arduino.cc/en/reference/liquidCrystal

អនុវគ្គន៍: អ្នកត្រូវតភ្ជាប់ Potentiometer ជាមួយ Arduino ។អ្នកត្រូវសរសេកូដមួយដែលអាចអោយយើញបម្រែបម្រួលតម្លៃលេខដែលទទួលបានតាមរយៈ Analog Read ហើយតម្លៃលេខដែលប្រែប្រួលទាំងអស់នោះនឹងត្រូវបង្ហាញលើអេក្រង់ LCD។

9. ផ្ទាំងអេក្រង់(OLED)

OLED ជាផ្ទាំង LCD ខ្នាតតូចតូចមួយប្រើប្រាស់សម្រាប់ទុកបង្ហាញដូចទៅនឹង Liquid Crystal ដែរប៉ុន្តែវាមានលក្ខណៈពិសេសច្រើនជាង Liquid Crystal។



https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/

Function ដែលមាននៅក្នុង Library (#include <Adafruit_SSD1306.h>)មួយចំនួនដូចខាង*ក្រោម*ៈ

-Adafruit_SSD1306() សម្រាប់ប្រកាស់អញ្ជាត្តរបស់ Library *នេះ។*

Syntax

Adafruit_SSD1306 OLED(RS_pin);

Example

Adafruit_SSD1306 myoled(13);

-begin() សម្រាប់ចាប់ផ្តើមដំណើរការអោយOLED ទៅ Address របស់វា។

Syntax OLED.begin(address)

Example

```
Adafruit_SSD1306 myoled(13);
myoled.begin(0x3c);
```

-display() សម្រាប់បញ្ជារអោយ*OLED*បង្ហាញអក្សរ ឬរូបចេញ

Syntax

OLED.display();

-clearDisplay() សម្រាប់បញ្ជារអោយ OLED សំអាត screen oled

Syntax

OLED.clearDisplay();

-drawPixel() សម្រាប់គូសចំនុចមួយPixel នៅត្រង់ទីតាំង X,Y *។*

Syntax

OLED.drawLPixel(x,y,color)

Example

myoled.drawPixel(64,32,WHITE);

-drawCircle() ប្រើសម្រាប់គូសរង្វង់មួយមានផ្ចិតនៅត្រង់ទីតាំង *x,y*និងមានកាំ *r* *។*

Syntax

OLED.drawCircle(x,y,r,color)

Example

myoled.drawCircle(64,32,20,WHITE);

-drawRect() ប្រើសម្រាប់គូសចតុកោណកែង។

Syntax

OLED.drawRect(x,y,w,h,color);

Example

myoled.drawRect(5,5,100,50,WHITE);

-fillRect() ប្រើសម្រាប់គូសចតុកោណកែងពេញសាច់ នៅត្រង់ចំនុច x,y ហើយបណ្តោយតាង*w* និងទទឹងតាង *h*។

Syntax

OLED.fillRect(x,y,w,h,color)

example

myoled.fillRect(10,10,70,30,WHITE);

-drawRoundRect() ប្រើសម្រាប់គូសចតុកោណកែងមានលុបជ្រុងកោង។

Syntax

OLED.drawRoundRect(x,y,w,h,r,color); (r is radius fillet)

Example

myoled.drawRoundRect(10,10,100,50,10,WHITE);

-drawTriangle() សម្រាប់គូសត្រីកោណ។

Syntax

OLED.drawTriangle(x1,y1,x2,y2,x3,y3,color);

Example

myoled.drawTriangle(10,5,100,10,50,40,WHITE);

-drawLine() *សម្រាប់គូសបន្ទាត់។*

Syntax

OLED.drawLine(x1,y1,x2,y2,color);

Example

myoled.drawLine(5,5,120,60,WHITE);

-setTextSize() *សម្រាប់កំណត់ទំហំអក្សរ។*

Syntax OLED.setTextSize(size);

-setTextColor() *សម្រាប់ដាក់ពណ៌អក្សរ។*

Syntax OLED.setTextColor(color);

-setCusor() *សម្រាប់កំណត់ទីតាំង* Cusor *។*

Syntax OLED.setCusor(x,y);

-print/println() *សម្រាប់សេរសេរអក្សរ*

Syntax

OLED.print(String);

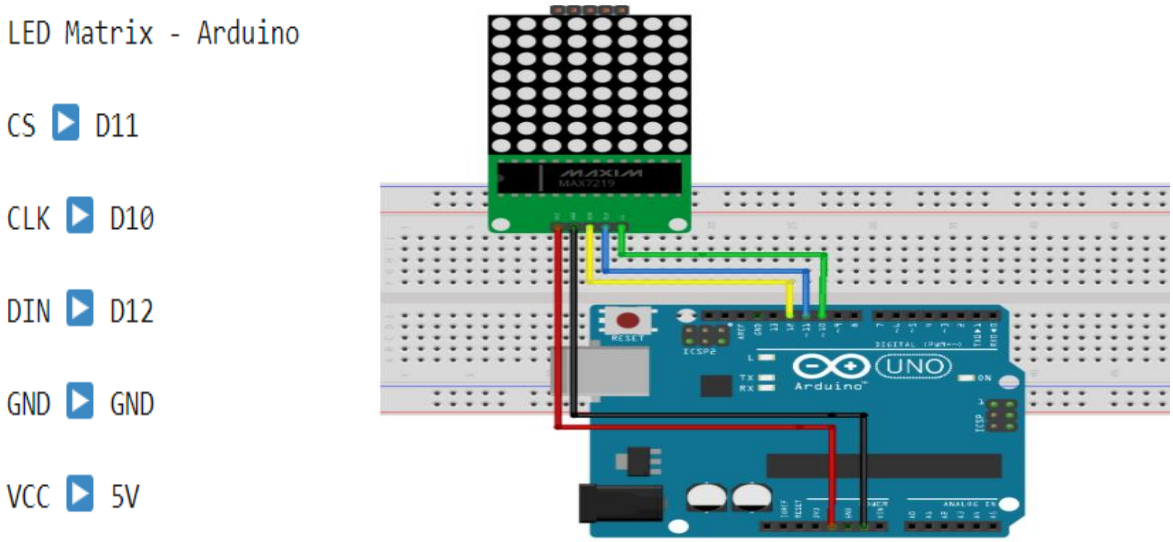
OLED.println(String);

coding

```
#include <Adafruit_GFX.h>
#include <gfxfont.h>
#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 myoled(13);
void setup() {
          myoled.begin(0x3c);
          myoled.clearDisplay();
          myoled.drawPixel(64,32,WHITE);
          myoled.drawLine(5,5,120,60,WHITE);
          myoled.drawCircle(64,32,20,WHITE);
          myoled.drawRect(5,5,100,50,WHITE);
          myoled.fillRect(10,10,70,30,WHITE);
          delay(3000);
          myoled.display();
          myoled.clearDisplay();
          myoled.drawRoundRect(10,10,100,50,10,WHITE);
          myoled.drawTriangle(10,5,100,10,50,40,WHITE);
          myoled.setTextSize(1);
          myoled.setTextColor(WHITE);
          myoled.setCursor(10,30);
          myoled.print("HELLO WORLD");
          myoled.display();
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

10. LED Matrix

LED Matrix ជាសំនុំ LED ដែលគេបានតម្រៀបឡើងមានរបៀបរៀបរយជាជួរឈរ និងជួរដេក។ Matrix ដែលយើងនឹងយកមកសិក្សានេះជាប្រភេទ Matrix 8x8 មានន័យថា 8 ជួរដេក និង8 ជួរឈរ។Matrix គេនិយមយកវាទៅប្រើប្រាស់ជាមួយគម្រោង Arduino បែបលក្ខណៈបង្ហាញ អក្សរ រូបនិមត្តិសញ្ញា និងរូបភាពជាដើម។



https://educ8s.tv/arduino-8x8-led-matrix-tutorial/

```
#include<LedControl.h>
int DIN=12;
int CS=11;
int ClK=10;
byte smile[8]={0x3C,0x42,0xA5,0x81,0xA5,0x99,0x42,0x3C};
byte neutral8]={0x3C,0x42,0xA5,0x81,0xBD,0x81,0x42,0x3C};
byte frown[8]={0x3C,0x42,0xA5,0x81,0x99,0xA5,0x42,0x3C};

LedControl lc=LedControl(DIN,CLK,CS,0);
void setup()
{
        lc.shutdown(0,false);
        lc.setIntensity(0,15);
        lc.clearDisplay(0);
}
void loop()
{
        printByte(smile);
        delay(1000);
        printByte(neutral);
        delay(1000);
        printByte(frown);
        delay(1000);
```

```
}
void printByte(byte character[] )
{
        int i=0;
        for (i=0; i<8;i++)
        {
                lc.setRow(0,i ,character[i]);
        }
}
```

The End