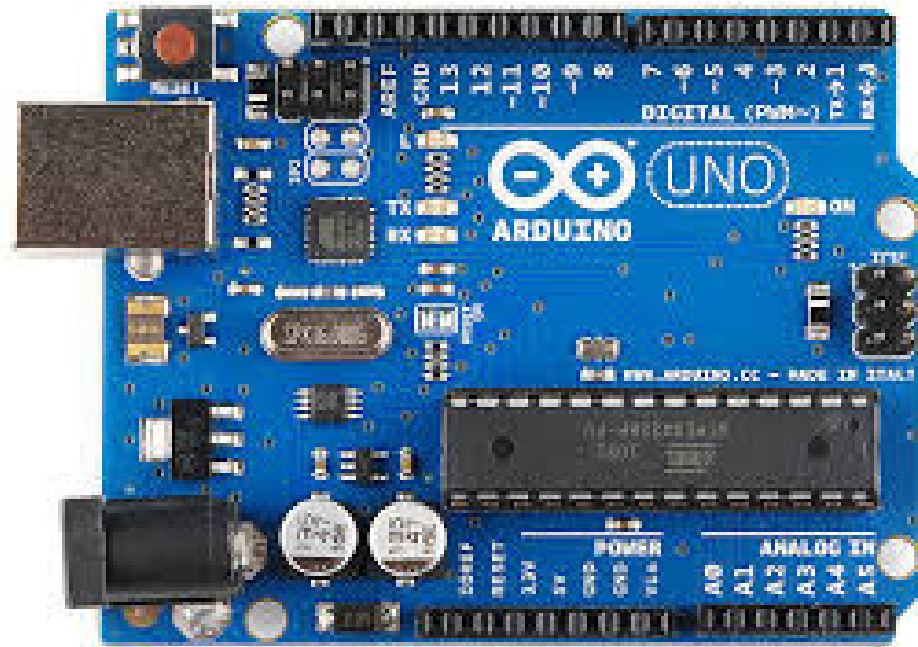


# Core Boards & Interfaces

## 1. Arduino Uno Board

- ເປັນບອດຄອມພິວເຕີຂະໜາດນ້ອຍ (**Microcontroller Board**) ທີ່ເປັນສະໜອງໃຫຍ່ຂອງໂຄງການ
- ມີໄມໂຄຄອນໂທລເລີ **ATmega328P** ຢູ່ພາຍໃນ
- ມີພິນດິຈິຕອລ **14** ພິນ ແລະ ພິນອານາລ໌ອກ **6** ພິນ
- ສາມາດໂປຣແກຣມໄດ້ງ່າຍດ້ວຍພາສາ **Arduino (C/C++)**

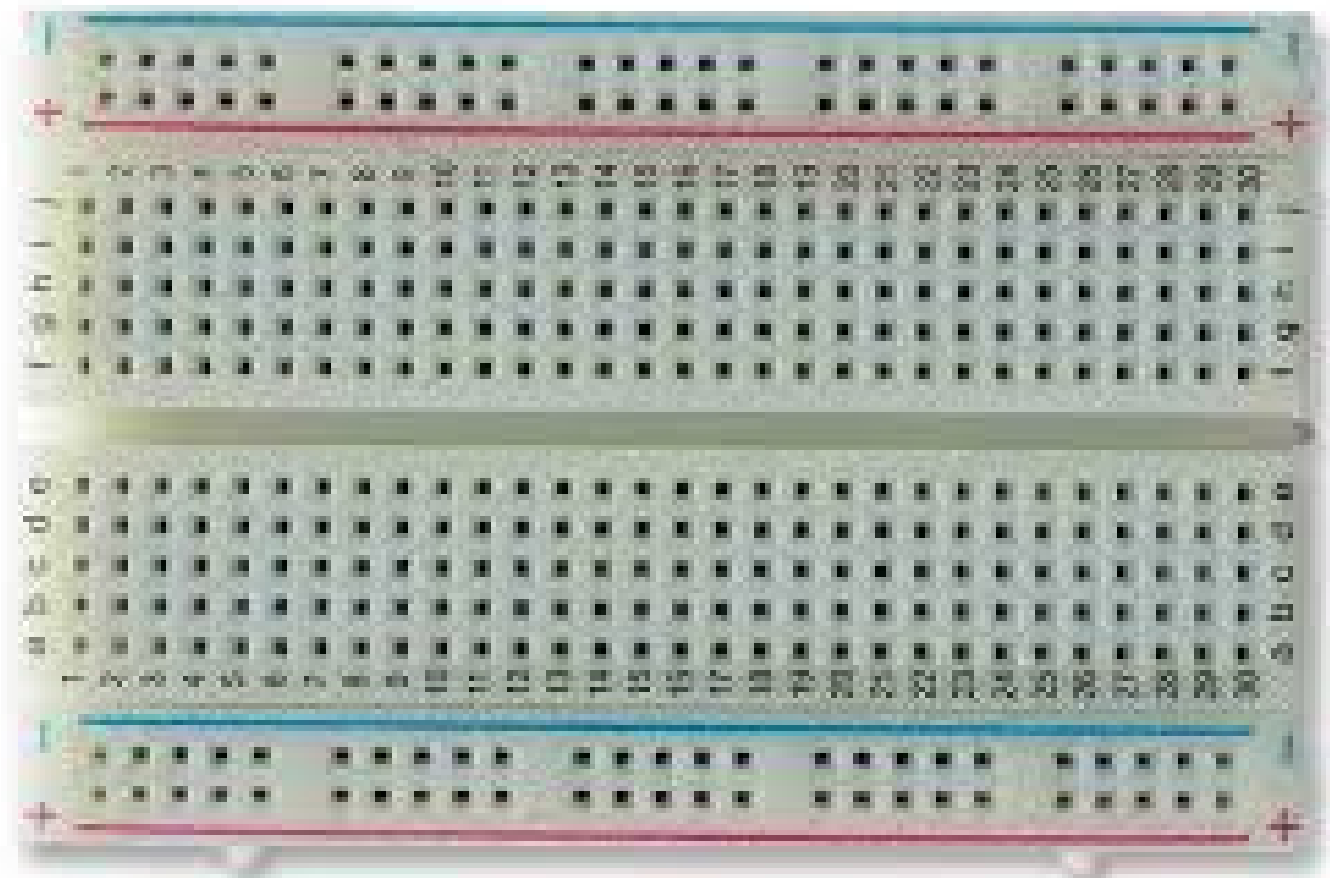


### ວິທີໃຊ້ງານ

- ປະມວນຜົນຄໍາສັ່ງຕ່າງໆຕາມໂປຣແກຣມທີ່ກຳນົດໄວ້
  - ຮັບສັນຍານເຂົ້າ (**Input**) ຈາກເຊັ່ນເຊີ, ປຸ່ມກົດ
  - ສົ່ງສັນຍານອອກ (**Output**) ໄປຄວບຄຸມ **LED**, ມໍເຕີ, ລຳໂພງ
- ຕົວຢ່າງການນຳໃຊ້:
- ຄວບຄຸມໄຟ **LED** ໃຫ້ກະພິບ
  - ອ່ານອຸນຫະພູມຈາກເຊັ່ນເຊີ
  - ຄວບຄຸມຫຸ່ນຍົນ, ໂຄງການ **IoT**

## 2. Breadboard

- ແຜ່ນສຳລັບຕໍ່ວົງຈອນໄຟຟ້າແບບບໍ່ຕ້ອງບັດກີ
- ມີຮູນ້ອຍໆຫຼາຍຮູສຳລັບສຽບສາຍ ແລະ ອຸປະກອນອີເລັກໂທຣນິກ
- ສາຍພາຍໃນຕໍ່ຮູແຕ່ລະແຖວເຂົ້າກັນ
- ເໝາະສຳລັບທົດລອງວົງຈອນກ່ອນຕໍ່ແບບຖາວອນ



### ວິທີໃຊ້ງານ:

- ວິທີການຕໍ່ວົງຈອນ:
- ສຽບອຸປະກອນອີເລັກໂທຣນິກ (**LED**, ຕົວຕ້ານທານ) ເຂົ້າຮູ
- ໃຊ້ສາຍຈັມເປີ້ (**Jumper Wire**) ຕໍ່ເຊື່ອມວົງຈອນ
- ແຖວແນວນອນບາງແຖວເປັນ + (ບວກ) ແລະ - (ລົບ) ສຳລັບແຈກຈ່າຍໄຟ
- ຈຸດດີ:
- ບໍ່ຕ້ອງບັດກີ, ປ່ຽນແປງວົງຈອນໄດ້ງ່າຍ
- ເໝາະກັບການຮຽນຮູ້ ແລະ ທົດລອງ

### 3. USB Cable

- ສາຍເຊື່ອມຕໍ່ລະຫວ່າງ **Arduino** ກັບຄອມພິວເຕີ
- ປົກກະຕິເປັນສາຍ **USB Type-A** ຫາ **Type-B**
- ໃຊ້ສໍາລັບ:
  - ອັບໂຫຼດໂປຣແກຣມເຂົ້າບອດ
  - ສົ່ງໄຟເລີ້ຍ **Arduino (5V)**
  - ສື່ສານຂໍ້ມູນລະຫວ່າງ **Arduino** ກັບຄອມ

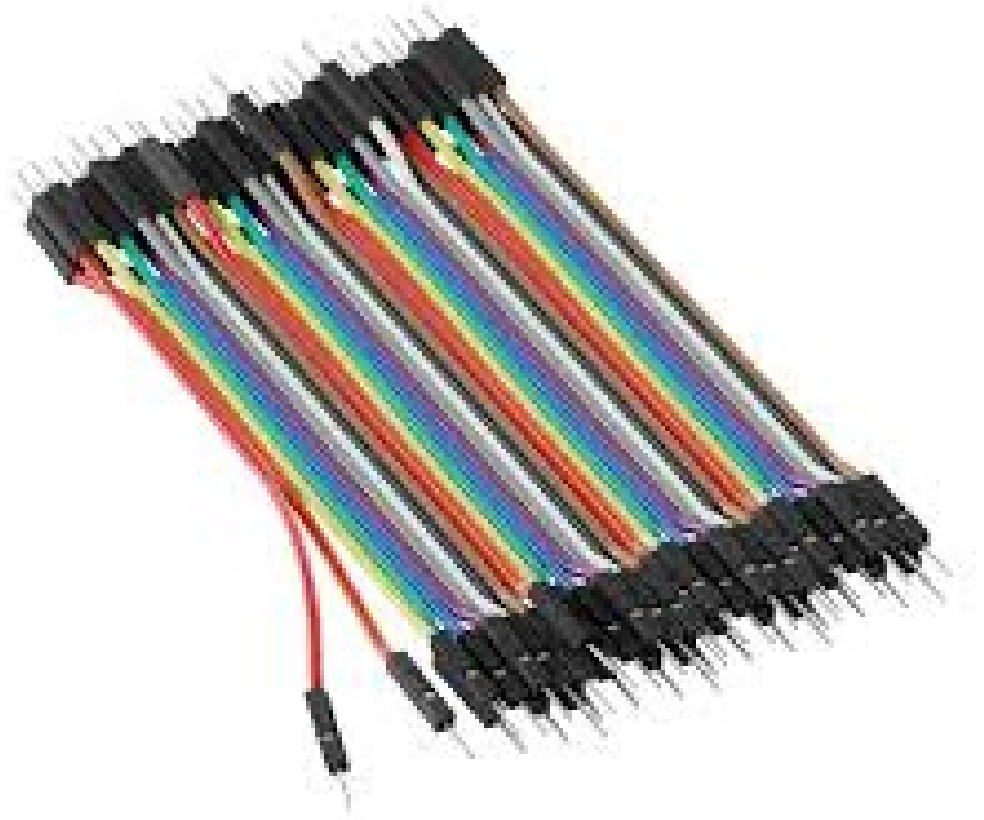


- ວິທີໃຊ້ງານ:
  - ຕໍ່ສາຍ: ປຽກ **USB** ດ້ານໜຶ່ງເຂົ້າກັບ **Arduino**, ອີກດ້ານເຂົ້າຄອມພິວເຕີ
  - ຕິດຕັ້ງໂປຣແກຣມ: ເປີດ **Arduino IDE** ເທິງຄອມ
  - ຂຽນໂຄ້ດ: ສ້າງໂປຣແກຣມທີ່ຕ້ອງການ
  - ອັບໂຫຼດ: ກົດປຸ່ມ **Upload** ເພື່ອສົ່ງໂປຣແກຣມໄປບອດ
  - ເຊັກຜົນລັບ: ເບິ່ງການເຮັດວຽກຂອງໂຄງການ
  - ຕົວຢ່າງໂຄງການງ່າຍໆ

# Wiring & Connectors

## 4. Jumper wires (male-to-male)

- ສາຍເຊື່ອມຕໍ່ທີ່ມີຫົວແຫຼມ (ພິນຜູ້) ສອງຂ້າງ
- ປົກກະຕິມີຫຼາຍສີ ເພື່ອງ່າຍຕໍ່ການຈຳແນກວົງຈອນ
- ຄວາມຍາວປະມານ **10-30** ຊັງຕີແມັດ



- ການນຳໃຊ້ຫຼັກ:
  - ຕໍ່ວົງຈອນເທິງ **Breadboard** (ສຽບເຂົ້າຮູ)
  - ເຊື່ອມຈຸດຕ່າງໆໃນ **Breadboard** ເຂົ້າຫາກັນ
  - ຕໍ່ພິນຜູ້ຂອງໂມດູນເຊັ່ນເຊີບາງຊະນິດ

## 5. Jumper wires (male-to-female)

- ສາຍເຊື່ອມຕໍ່ທີ່ມີຫົວແຫຼມ (ພິນຜູ້) ສອງຂ້າງ
- ປົກກະຕິມີຫຼາຍສີ ເພື່ອງ່າຍຕໍ່ການຈຳແນກວົງຈອນ
- ຄວາມຍາວປະມານ **10-30** ຊັງຕີແມັດ



- ການນຳໃຊ້ຫຼັກ:
  - ຕໍ່ວົງຈອນເທິງ **Breadboard** (ສຽບເຂົ້າຮູ)
  - ເຊື່ອມຈຸດຕ່າງໆໃນ **Breadboard** ເຂົ້າຫາກັນ
  - ຕໍ່ພິນຜູ້ຂອງໂມດູນເຊັ່ນເຊີບາງຊະນິດ

## 6. Jumper wires (female-to-female)

- ສາຍຂະໜາດນ້ອຍສີສັນຕ່າງໆ ປາຍທັງສອງເປັນຮູ (female).
- **Schematic / Pinout:**
- ສາຍໄຟທີ່ໃຊ້ເຊື່ອມຕໍ່ລະຫວ່າງ **PIN** ທີ່ເປັນ **male** (ເຊັ່ນ **Sensor, Module, Board**).



- ການເຮັດວຽກ:
- ໃຊ້ເຊື່ອມຕໍ່ສັນຍານຈາກອຸປະກອນໜຶ່ງໄປອີກໜຶ່ງ ຢ່າງປອດໄພ ໂດຍບໍ່ຕ້ອງບັດບອດ.
- ໃຊ້ກັບ **Breadboard** ທີ່ບອດທີ່ມີ **pin male** ຈະສະດວກທີ່ສຸດ.
- ຕົວຢ່າງການນຳໃຊ້:
- ເຊື່ອມ **DHT11 Sensor** ເຂົ້າກັບ **Breadboard** ທີ່ **LCD Module**.

## 7. 9V Battery Connector

- ສາຍຕໍ່ຖ່ານ **9V** ມີຝາປິດຄວາມແຂງແຮງ ແລະສາຍສອງສາຍ (ແດງແລະດຳ).
- **Schematic / Pinout:**
- ແດງ = **+9V**
- ດຳ = **GND**



- ການເຮັດວຽກ:
- ສົ່ງພະລັງງານຈາກຖ່ານ **9V** ເຂົ້າບອດ **Arduino** ຜ່ານຂາ **Vin** ຫຼືພອດພາວເຕີ.
- ຕົວຢ່າງການນຳໃຊ້:
- ໃຊ້ໃນໂຄງງານທີ່ບໍ່ມີຄອມພິວເຕີເຊື່ອມຜ່ານ **USB**.
- ຄຳແນະນຳພິເສດ:
- ຄວນໃຊ້ຖ່ານ **9V** ທີ່ມີຄຸນນະພາບດີ ເພື່ອຫຼຸດຄວາມສ່ຽງຂອງໄຟຕົກ.

# Basic Components

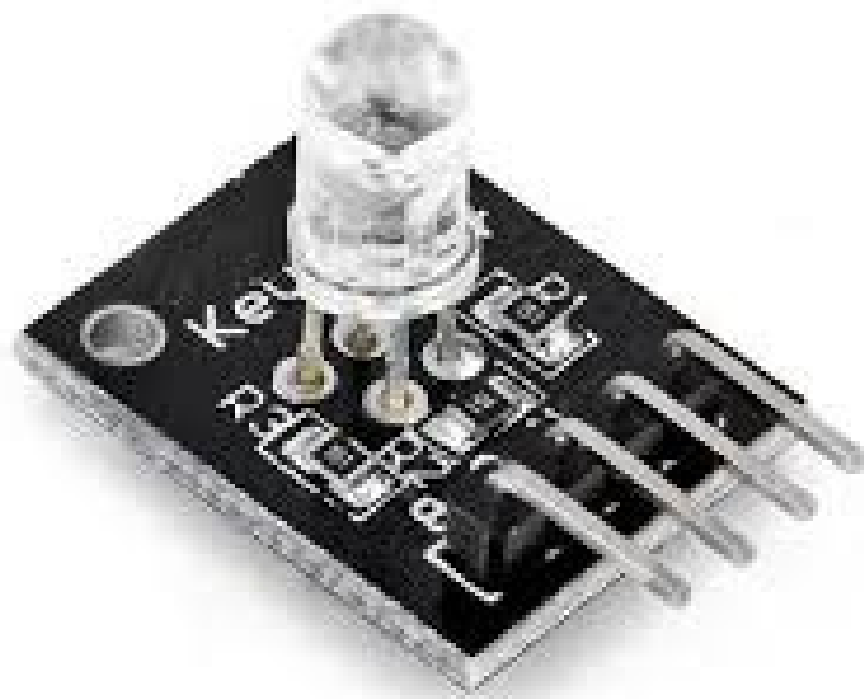
## 8. LEDs (Red: 5, Yellow: 5, Blue: 5, RGB: 1)

- ໄຟດວງນ້ອຍທີ່ມີຂາສອງຂາ (**Anode +, Cathode -**)
- **Schematic / Pinout:**
- **Long leg = Anode (+)**
- **Short leg = Cathode (-)**



- ການເຮັດວຽກ:
- ໃຫ້ແສງສະຫວ່າງເມື່ອມີໄຟໄຫຼຜ່ານໂດຍຜ່ານຕົວຕ້ານ (**Resistor**).
- ຕົວຢ່າງການນຳໃຊ້:
- ໃຊ້ໃນໂຄງງານ **Blink LED**, ໄຟສະຖານະ, ຕົວຊີ້ສະຖານະຂອງ **sensor**.
- ຄຳແນະນຳພິເສດ:
- ໃຊ້ຕົວຕ້ານ **220Ω** ຫຼື **330Ω** ເພື່ອປ້ອງກັນໄຟເກີນຈົນ **LED** ໄໝ້.

## 9. RGB LED Module



- ຄໍາອະທິບາຍ: ໂມດູນ **LED RGB** ທີ່ມີ **resistor** ໃນຕົວ
- **Pinout:**
- **R (Red), G (Green), B (Blue) pins**
- **Common Cathode:** ຕໍ່ **GND**
- ການໃຊ້ງານ: ຄວບຄຸມ **3 pins** ດ້ວຍ **PWM** ເພື່ອສ້າງສີຕ່າງໆ
- ຕົວຢ່າງການນຳໃຊ້: ສ້າງຜົນໄຟຟ້າສີ, ສະແດງສະຖານະ



## 10. Resistors (220Ω, 1kΩ, 10kΩ)



- ຄຳອະທິບາຍ: ຕົວຕ້ານທານເພື່ອຈຳກັດກະແສໄຟຟ້າ
- ການໃຊ້ງານ:
  - **220Ω**: ໃຊ້ກັບ **LED** ( $5V / 0.02A = 250\Omega$ )
  - **1kΩ**: ໃຊ້ກັບ **button pull-down**
  - **10kΩ**: ໃຊ້ກັບ **button pull-up**
- ວິທີອ່ານສີ:
  - **220Ω**: ແດງ-ແດງ-ນ້ຳຕານ-ທອງ
  - **1kΩ**: ນ້ຳຕານ-ດຳ-ແດງ-ທອງ
  - **10kΩ**: ນ້ຳຕານ-ດຳ-ສີເຫຼືອງ-ທອງ
- ຕົວຢ່າງການນຳໃຊ້: ປົກປ້ອງ **LED**, **pull-up/pull-down resistors**

## 11. Push Buttons (x4 with Lids)



- ຄໍາອະທິບາຍ: ປຸ່ມກົດແບບ **tactile switch**
- ການໃຊ້ງານ:
- 4 ຂາ: 2 ຄູ່ເຊື່ອມຕໍ່ກັນ
- ກົດແລ້ວຈະເຊື່ອມວົງຈອນ
- **Wiring:**
- ຂາໜຶ່ງຕໍ່ 5V
- ອີກຂາໜຶ່ງຕໍ່ **digital pin + 10k $\Omega$  resistor to GND**

ຕົວຢ່າງການນໍາໃຊ້:

```
int buttonState = digitalRead(2);  
if(buttonState == HIGH)
```

## 12. Potentiometer (5kΩ)



- ຄຳອະທິບາຍ: ຕົວຕ້ານທານປັບໄດ້ **5kΩ**
- **Pinout:**
- ຂາ **1, 3**: ຕໍ່ **5V** ແລະ **GND**
- ຂາ **2** (ກາງ): ຕໍ່ **Analog pin**
- ການໃຊ້ງານ: ຫມຸນເພື່ອປ່ຽນຄ່າແຮງດັນອອກ **0-5V**

ຕົວຢ່າງການນຳໃຊ້:

```
int value = analogRead(A0); // 0-1023  
int brightness = map(value, 0, 1023, 0, 255);
```

## 13. Active Buzzer



- ຄຳອະທິບາຍ: **Buzzer** ທີ່ມີ **oscillator** ໃນຕົວ ສົ່ງສຽງຄົງທີ່
- ການໃຊ້ງານ:
- ຕໍ່ + ກັບ **digital pin**
- ຕໍ່ - ກັບ **GND**
- ສົ່ງ **HIGH** ກໍ່ດັງ

ຕົວຢ່າງການນຳໃຊ້:

```
digitalWrite(8, HIGH); // ດັງ  
digitalWrite(8, LOW); // ຢຸດ
```

## 14. Passive Buzzer



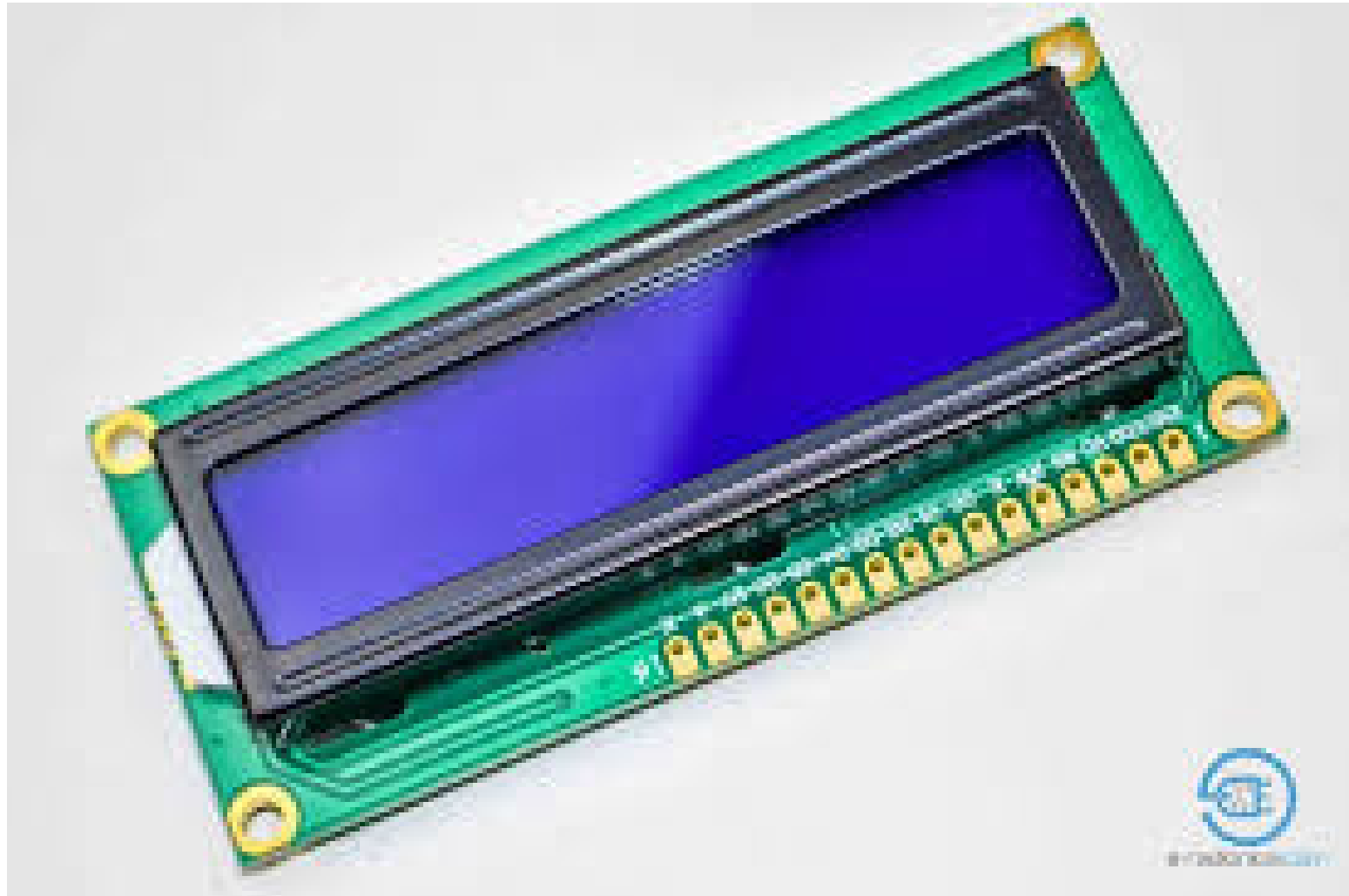
- ຄຳອະທິບາຍ: **Buzzer** ທີ່ຕ້ອງການສັນຍານ **PWM** ເພື່ອສົ່ງສຽງ
- ການໃຊ້ງານ: ສາມາດສ້າງສຽງຄວາມຖີ່ຕ່າງໆ ດ້ວຍ **tone() function**

ຕົວຢ່າງການນຳໃຊ້:

```
tone(9, 1000);    // ສົ່ງສຽງ  
1000HznoTone(9);    // ຢຸດສຽງ
```

# Displays & Output

## 15. 16x2 LCD Display

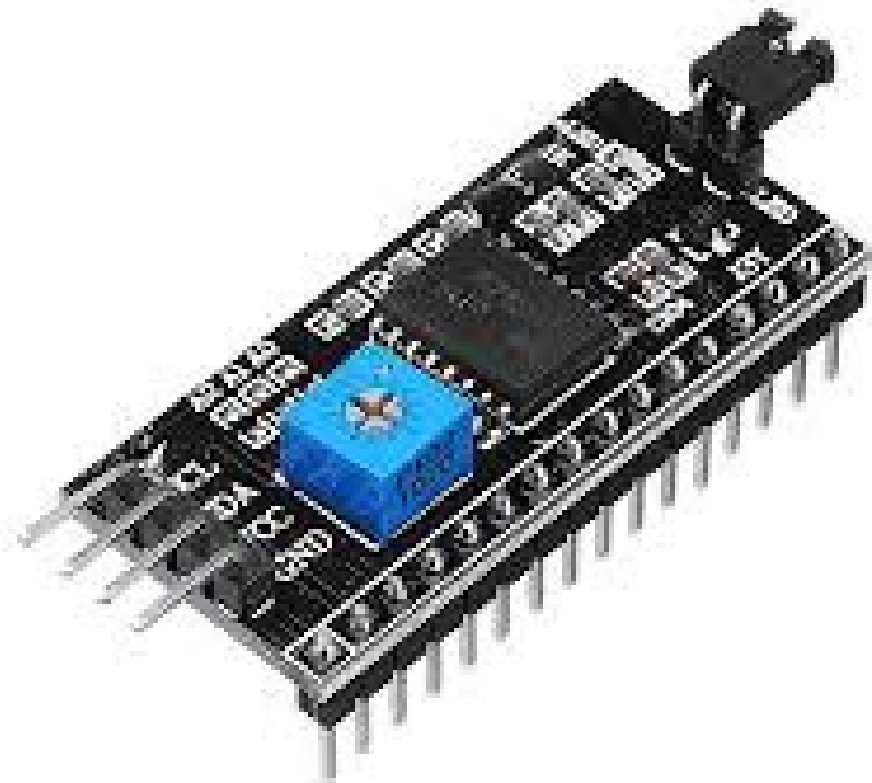


- ຄໍາອະທິບາຍ: ຈໍສະແດງຜົນ **LCD 16** ຕົວອັກສອນ x **2** ແຖວ
- **Pinout:**
- **VSS, VDD, VO, RS, RW, E**
- **D0-D7 (data pins)**
- **A (backlight +), K (backlight -)**
- ການໃຊ້ງານ: ສະແດງຂໍ້ຄວາມ, ຕົວເລກ, ສັນຍາລັກ

ຕົວຢ່າງການນຳໃຊ້:

```
#include <LiquidCrystal.h>  
LiquidCrystal lcd(12,11,5,4,3,2);  
lcd.begin(16, 2);  
lcd.print("Hello World!");
```

## 16. I2C Serial Adapter Module

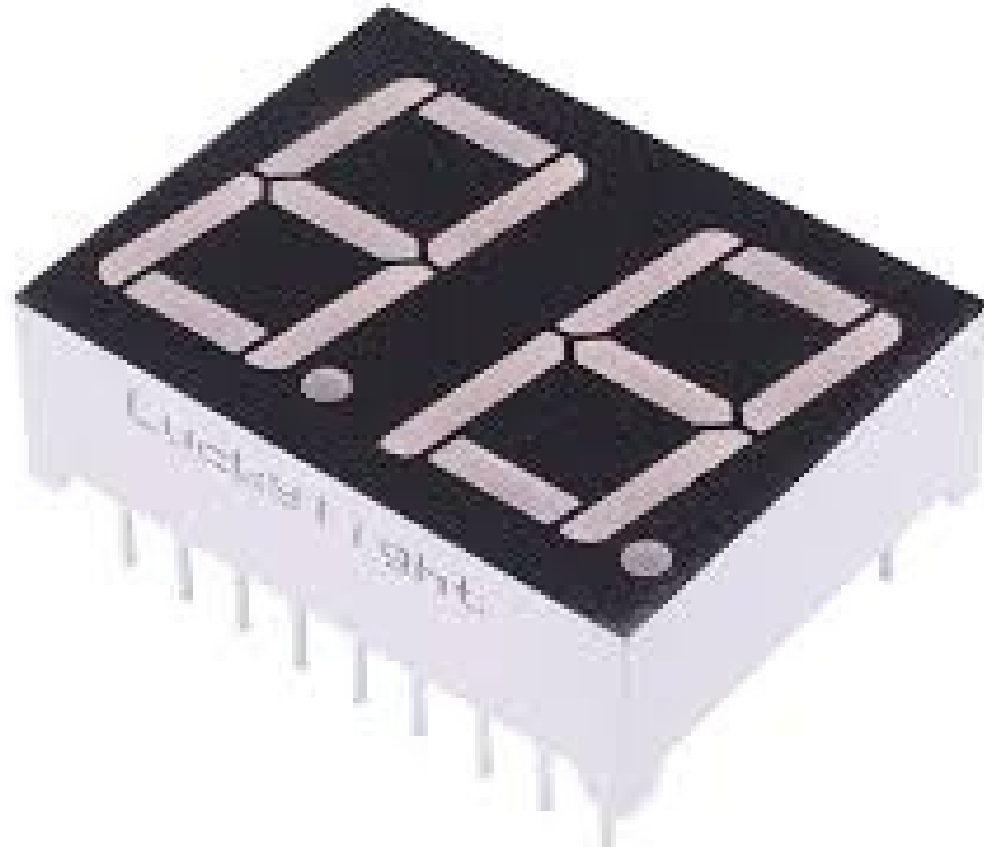


- ຄໍາອະທິບາຍ: ໂມດູນປ່ຽນ LCD ເປັນ I2C ເພື່ອຫຼຸດ pins
- **Pinout:**
- **GND, VCC, SDA (A4), SCL (A5)**
- ການໃຊ້ງານ: ໃຊ້ພຽງ 2 pins (SDA, SCL) ແທນ 6-8 pins

ຕົວຢ່າງການນຳໃຊ້:

```
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27, 16, 2);  
lcd.init();  
lcd.print("I2C LCD");
```

## 17. 7-Segment Display (Common Cathode)



- ຄຳອະທິບາຍ: ຈຳສະແດງຜົນຕົວເລກ **0-9** ແບບ **7** ເສັ້ນ
- **Pinout:**
- **Segments:** **a, b, c, d, e, f, g, dp**
- **Common Cathode:** ຕໍ່ກັບ **GND**
- ການໃຊ້ງານ: ສົ່ງ **HIGH** ໄປຍັງ **segment** ທີ່ຕ້ອງການໃຫ້ສະຫວ່າງ
- ຕົວຢ່າງການນຳໃຊ້: ສະແດງຕົວເລກ, **timer**



## 18. 4-Digit 7-Segment Display



- ຄໍາອະທິບາຍ: ຈໍສະແດງຜົນ 4 ຕົວເລກ ດ້ວຍ **TM1637 driver**
- **Pinout:**
- **CLK, DIO, VCC, GND**
- ການໃຊ້ງານ: ໃຊ້ **TM1637 library** ເພື່ອຄວບຄຸມ

ຕົວຢ່າງການນຳໃຊ້:

```
#include <TM1637Display.h>  
TM1637Display display(CLK, DIO);  
display.showNumberDec(1234);
```

## 19. 8x8 Dot Matrix Display



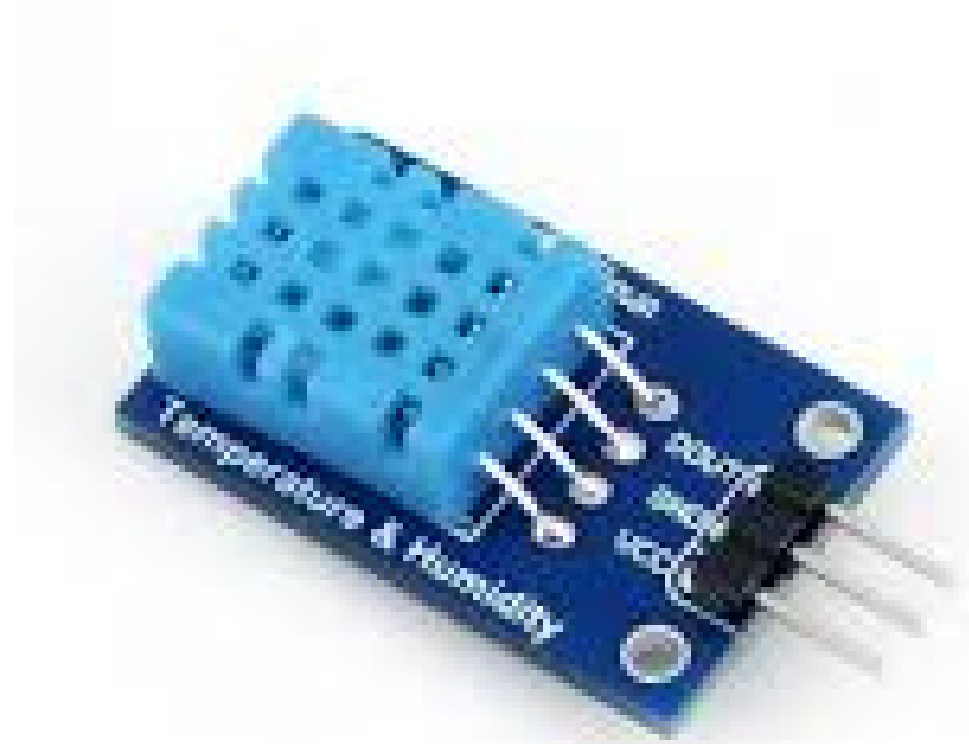
- ຄໍາອະທິບາຍ: ຈໍ **LED matrix 8x8** ດ້ວຍ **MAX7219 driver**
- **Pinout:**
- **VCC, GND, DIN, CS, CLK**
- ການໃຊ້ງານ: ສະແດງຮູບ, ຕົວອັກສອນ, **animation**

ຕົວຢ່າງການນຳໃຊ້:

```
#include <MD_MAX72xx.h>  
MD_MAX72XX mx = MD_MAX72XX(CS_PIN, 1);  
mx.setPoint(3, 3, true);
```

# Sensors & Input Modules

## 20. DHT11 Temperature & Humidity Sensor



- ຄຳອະທິບາຍ: ເຊັ່ນເຊີວັດອຸນຫະພູມ (**0-50°C**) ແລະຄວາມຊຸ່ມ (**20-90%**)
- **Pinout:**
- **VCC (3.3-5V), DATA, GND**
- ການໃຊ້ງານ: ອ່ານຂໍ້ມູນດ້ວຍ **DHT library**

ຕົວຢ່າງການນຳໃຊ້:

```
#include <DHT.h>  
DHT dht(PIN, DHT11);  
float temp = dht.readTemperature();  
float humidity = dht.readHumidity();
```

## 21. LM35 Temperature Sensor

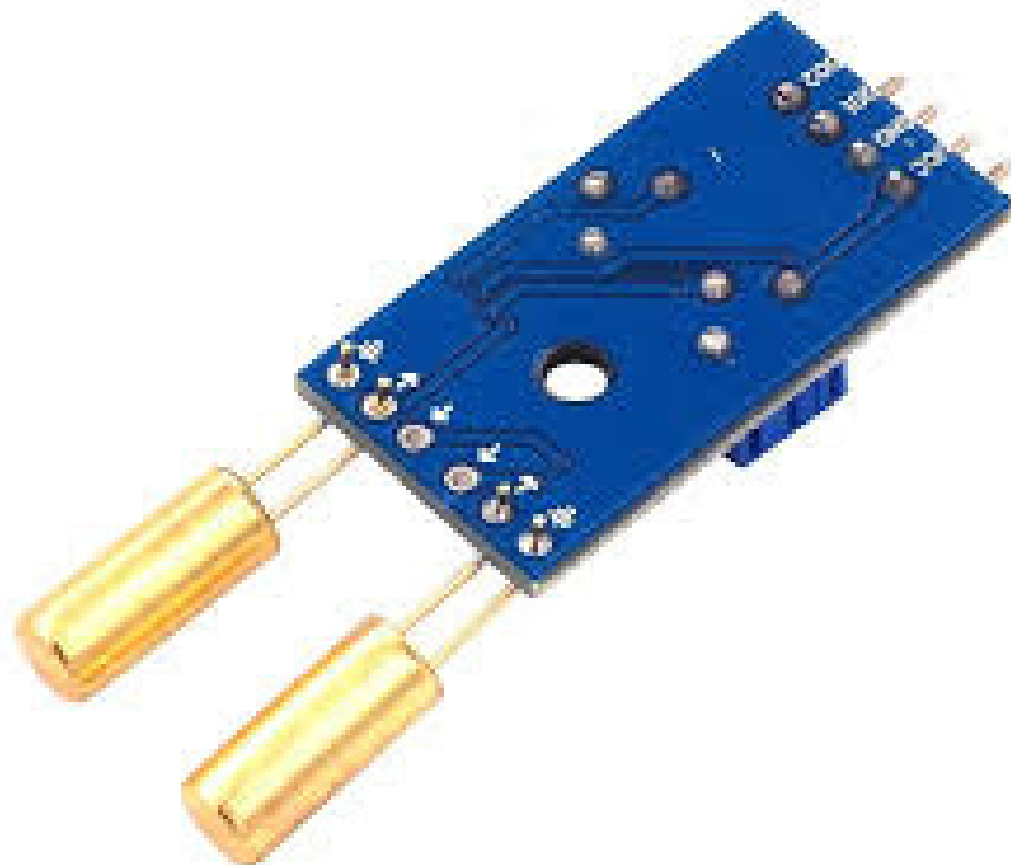


- ຄຳອະທິບາຍ: ເຊັ່ນເຊີວັດອຸນຫະພູມແບບ **analog** (-55 ຫາ 150°C)
- **Pinout:**
- **1: VCC, 2: Vout, 3: GND**
- ການໃຊ້ງານ: ອອກແຮງດັນ **10mV/°C**

ຕົວຢ່າງການນຳໃຊ້:

```
int value = analogRead(A0);  
float temp = (value * 5.0 * 100.0) / 1024.0;
```

## 22. Tilt Sensor (x2)



- ຄຳອະທິບາຍ: ເຊັ່ນເຊີຕວດຈັບການເອຽງ ມີລູກບານໃນຕົວ
- ການໃຊ້ງານ:
- ຕັ້ງຊື່: ເຊື່ອມວົງຈອນ (**LOW**)
- ເອຽງ: ເປີດວົງຈອນ (**HIGH**)
- ຕົວຢ່າງການນຳໃຊ້: ກະດິ່ງເຕືອນ, ເກມ, ຕວດຈັບທາງ

## 23. Photoresistor / LDR (x3)



- ຄໍາອະທິບາຍ: ຕົວຕ້ານທານທີ່ປ່ຽນຄ່າຕາມແສງສະຫວ່າງ
- ການໃຊ້ງານ:
- ແສງເຢັນ: ຄ່າສູງ ( $M\Omega$ )
- ແສງສະຫວ່າງ: ຄ່າຕໍ່າ ( $k\Omega$ )
- **Wiring: Voltage divider ກັບ  $10k\Omega$  resistor**

ຕົວຢ່າງການນຳໃຊ້:

```
int light = analogRead(A0);  
if(light < 500) { digitalWrite(LED, HIGH); }
```

## 24. PIR Motion Sensor



- ຄຳອະທິບາຍ: ເຊັ່ນເຊີຕວດຈັບການເຄື່ອນໄຫວ (**Passive Infrared**)
- **Pinout:**
- **VCC, OUT, GND**
- ປັບ **sensitivity** ແລະ **delay time**
- ການໃຊ້ງານ: ສົ່ງ **HIGH** ເມື່ອກວດພົບການເຄື່ອນໄຫວ

ຕົວຢ່າງການນຳໃຊ້:

```
if(digitalRead(PIR_PIN) == HIGH) {  
  Serial.println("Motion detected!");}
```

## 25. Ultrasonic Module (HC-SR04)



- ຄໍາອະທິບາຍ: ເຊັ່ນເຊີວັດໄລຍະດ້ວຍຄື້ນສຽງ **ultrasonic (2-400cm)**
- **Pinout:**
- **VCC, Trig, Echo, GND**
- ການໃຊ້ງານ:
- ສົ່ງ **pulse 10μs** ໃສ່ **Trig**
- ອ່ານເວລາ **pulse** ຈາກ **Echo**
- ໄລຍະ = ເວລາ × **0.034 / 2**

ຕົວຢ່າງການນຳໃຊ້:

```
digitalWrite(TRIG, HIGH);  
delayMicroseconds(10);  
digitalWrite(TRIG, LOW);  
long duration = pulseIn(ECHO, HIGH);  
float distance = duration * 0.034 / 2;
```

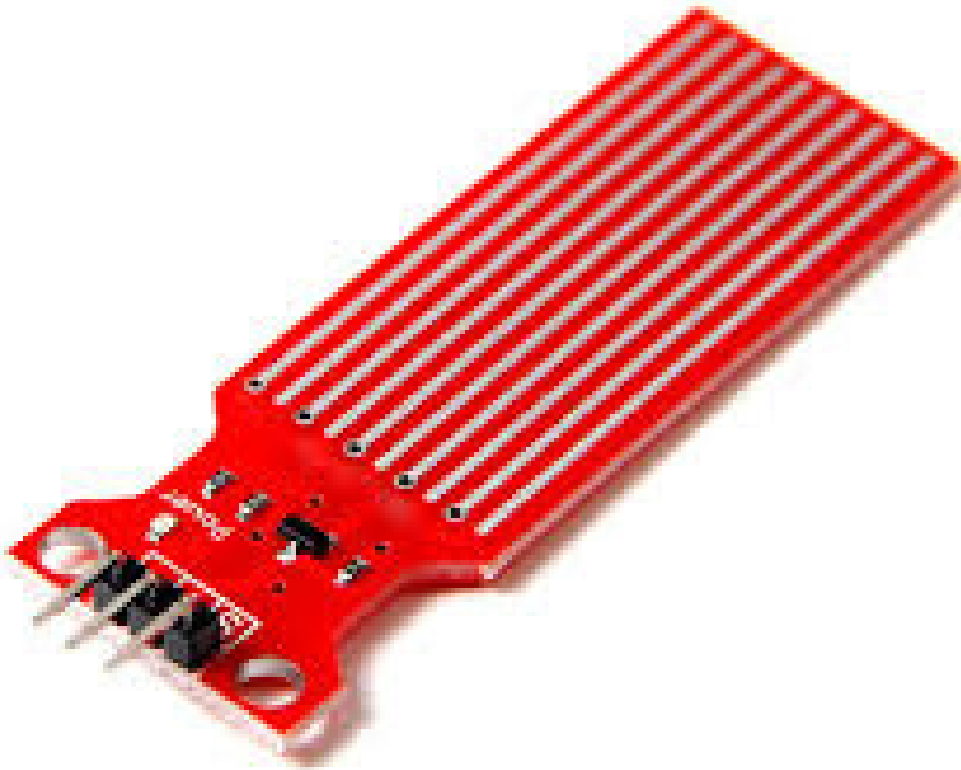


## 26. Sound Sensor



- ຄໍາອະທິບາຍ: ເຊັ່ນເຊີຕວດຈັບສຽງດ້ວຍໄມໂຄຣໂຟນ
- **Pinout:**
- **VCC, A0 (analog), D0 (digital), GND**
- ການໃຊ້ງານ:
- **Analog:** ລະດັບສຽງ **0-1023**
- **Digital:** ເກີນ **threshold** ຫລືບໍ່
- ຕົວຢ່າງການນຳໃຊ້: **Voice activated switch, sound level meter**

## 27. Water Sensor

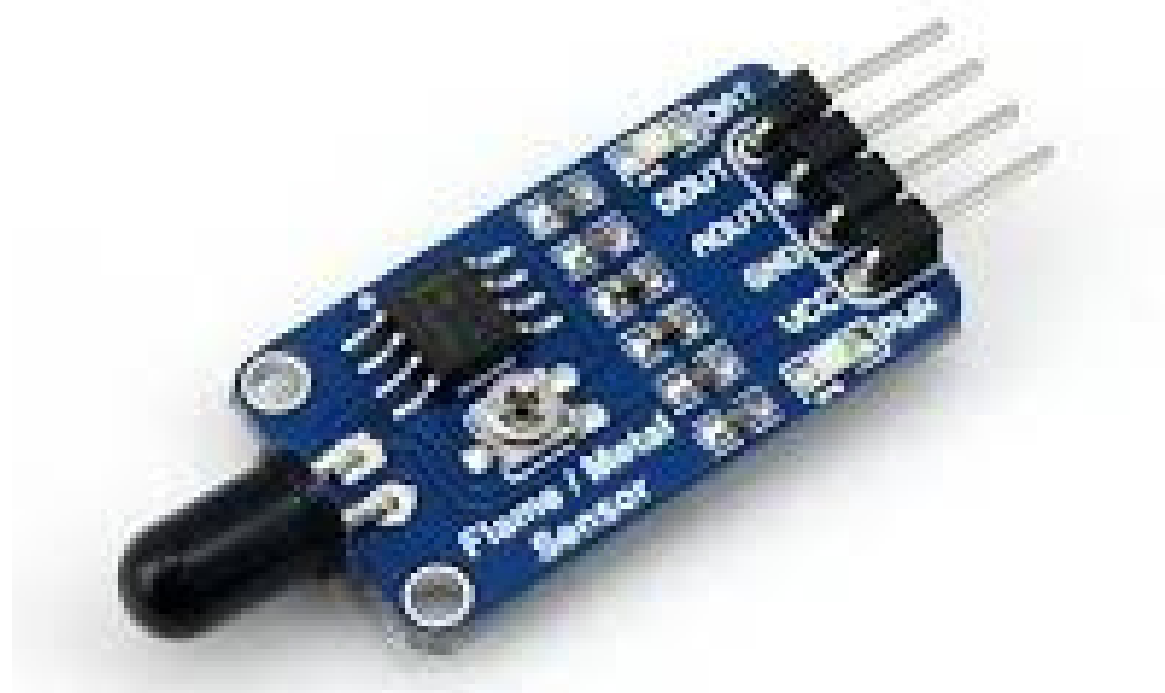


- ຄຳອະທິບາຍ: ເຊັ່ນເຊີຕວດຈັບລະດັບນ້ຳ
- **Pinout:**
- +, S (**signal**), - (**GND**)
- ການໃຊ້ງານ: ອ່ານຄ່າ **analog** ຕາມລະດັບນ້ຳ

ຕົວຢ່າງການນຳໃຊ້:

```
int waterLevel = analogRead(A0);  
if(waterLevel > 500) {  
  Serial.println("Water detected!");  
}
```

## 28. Flame Sensor

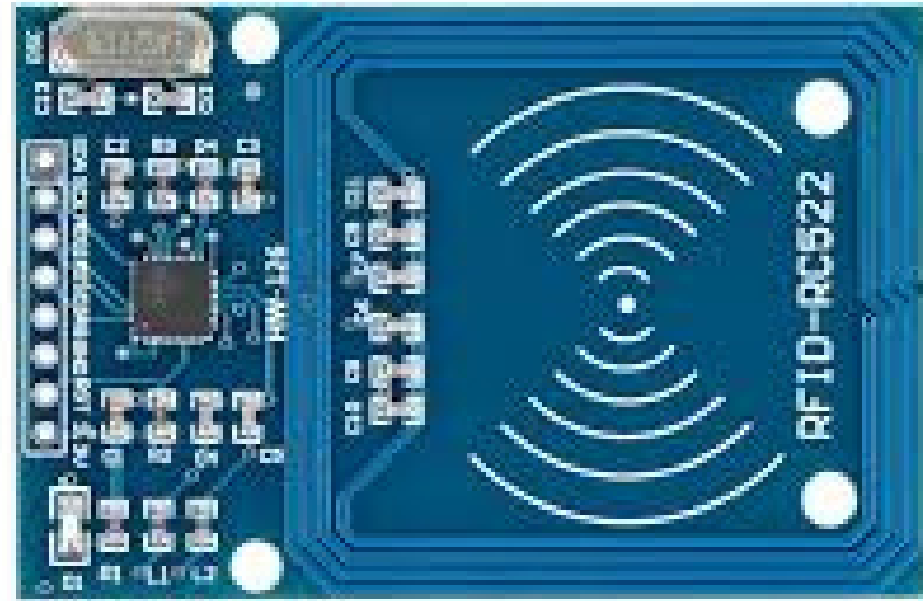


- ຄຳອະທິບາຍ: ເຊັ່ນເຊີຕວດຈັບແສງໄຟ / **infrared**
- **Pinout:**
- **VCC, D0 (digital), A0 (analog), GND**
- ການໃຊ້ງານ: ຕວດຈັບແສງໄຟໃນຊ່ວງ **760-1100nm**

ຕົວຢ່າງການນຳໃຊ້:

```
if(digitalRead(FLAME_PIN) == LOW) {  
  digitalWrite(BUZZER, HIGH);} 
```

## 29. RFID Module (RC522)



- ຄໍາອະທິບາຍ: ໂມດູນອ່ານບັດ **RFID 13.56MHz**
- **Pinout:**
- **SDA, SCK, MOSI, MISO, IRQ, GND, RST, 3.3V**
- ການໃຊ້ງານ: ສື່ສານດ້ວຍ **SPI protocol**

ຕົວຢ່າງການນຳໃຊ້:

```
#include <MFRC522.h>
MFRC522 rfid(SS_PIN, RST_PIN);
if(rfid.PICC_IsNewCardPresent())    {
Serial.println(rfid.uid.uidByte[0]);}
```

## 30. RFID Tag/Card



- ຄຳອະທິບາຍ: ບັດ **RFID** ຫລື **tag** ທີ່ມີ **UID** ເປັນເອກະລັກ
- ການໃຊ້ງານ: ເອົາໃກ້ກັບ **RFID reader** ເພື່ອອ່ານຂໍ້ມູນ
- ຕົວຢ່າງການນຳໃຊ້: ລະບົບຄວບຄຸມການເຂົ້າອອກ, ລະບົບຈ່າຍເງິນ

## 31. Infrared Receiver



- ຄຳອະທິບາຍ: ເຊັ່ນເຊີຮັບສັນຍານ **infrared 38kHz**
- **Pinout:**
- **OUT, GND, VCC**
- ການໃຊ້ງານ: ຮັບສັນຍານຈາກ **remote control**

ຕົວຢ່າງການນຳໃຊ້:

```
#include <IRremote.h>
IRrecv irrecv(IR_PIN);
decode_results results;
if(irrecv.decode(&results))      {
  Serial.println(results.value, HEX);}
```

# Remote & Control

## 32. Infrared Remote Control



- ຄຳອະທິບາຍ: ລືໂມດຄວບຄຸມດ້ວຍ **infrared**
- ການໃຊ້ງານ: ສົ່ງສັນຍານ **IR** ເພື່ອຄວບຄຸມອຸປະກອນ
- ຕົວຢ່າງການນຳໃຊ້: ຄວບຄຸມ **LED**, ມື້ເຕີ້, ຫລືໂປຣເຈັກຕ່າງໆຈາກໄກ

## 33. Joystick Module



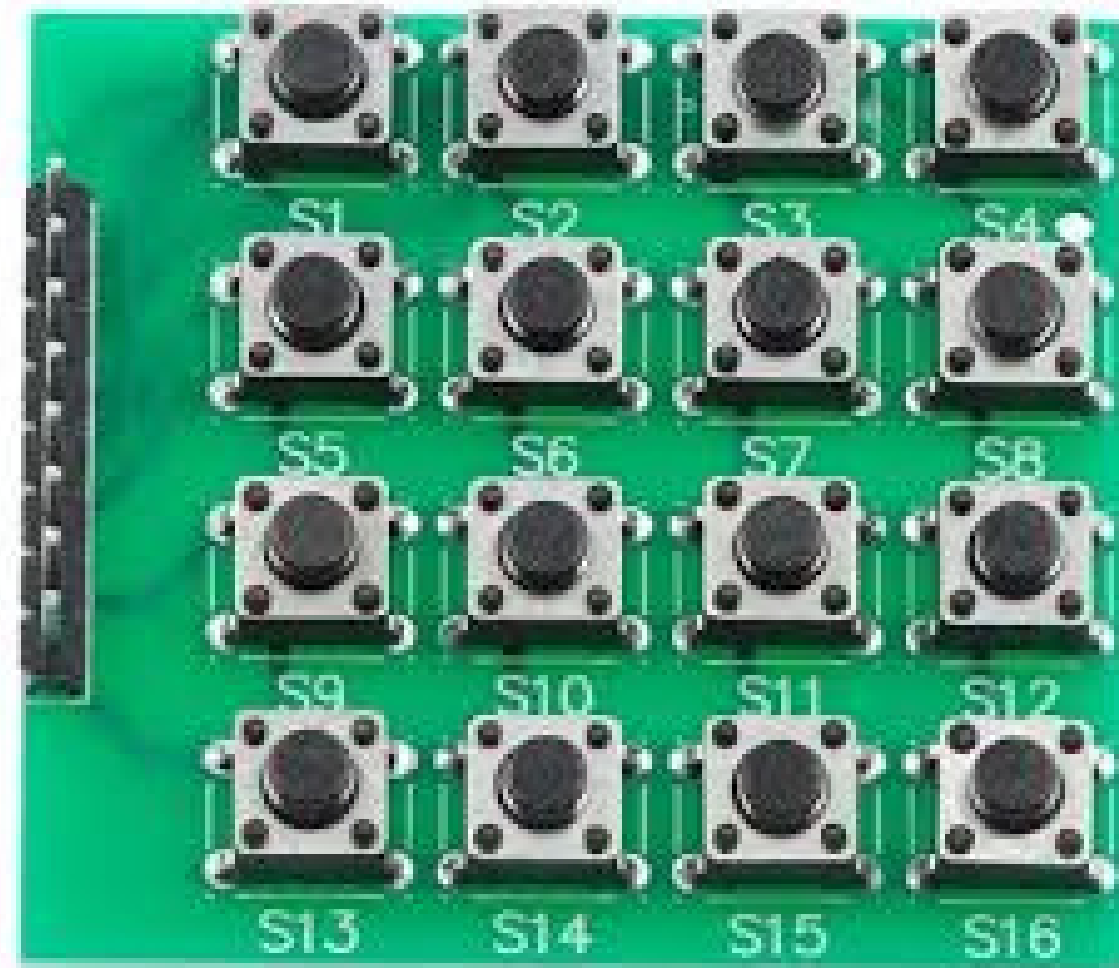
- ຄຳອະທິບາຍ: ໂມດູນຄວບຄຸມແບບ **analog 2** ແກນ (X, Y) + ປຸ່ມກົດ
- **Pinout:**
- **GND, +5V, VRx, VRy, SW**
- ການໃຊ້ງານ:
- **VRx/VRy: analog 0-1023** (ກາງ ~512)
- **SW: digital button (LOW ເມື່ອກົດ)**

ຕົວຢ່າງການນຳໃຊ້:

```
int xValue = analogRead(A0);  
int yValue = analogRead(A1);  
int button = digitalRead(2);
```



## 34. 4x4 Matrix Keyboard Module



- ຄໍາອະທິບາຍ: ແປ້ນພິມ **16** ປຸ່ມ (**0-9, A-D, \*, #**)
- **Pinout:**
- **8 pins: 4 rows, 4 columns**
- ການໃຊ້ງານ: ສະແກນ **rows/columns** ເພື່ອຕວດຈັບປຸ່ມທີ່ກົດ

ຕົວຢ່າງການນຳໃຊ້:

**cpp**

```
#include <Keypad.h>
```

```
char key = keypad.getKey();
```

```
if(key) { Serial.println(key);}
```

## 35. Relay Module



- ຄຳອະທິບາຍ: ໂມດູນສະວິດຄວບຄຸມໄຟແຮງດັນສູງ (**AC/DC**)
- **Pinout:**
- **VCC, GND, IN (control)**
- **COM, NO (Normally Open), NC (Normally Closed)**
- ການໃຊ້ງານ:
- ສົ່ງ **LOW** ເພື່ອປິດ **relay** (ເຊື່ອມ **COM-NO**)
- ສົ່ງ **HIGH** ເພື່ອເປີດ **relay**

ຕົວຢ່າງການນຳໃຊ້:

```
digitalWrite(RELAY_PIN, LOW);  
ACdelay(1000);  
digitalWrite(RELAY_PIN, HIGH);
```

# Motors & Drivers

## 36. Servo Motor



- ຄໍາອະທິບາຍ: ມໍເຕີຄວບຄຸມມຸມ **0-180°**
- **Pinout:**
- ສີນ້ຳຕານ/ດໍາ: **GND**
- ສີແດງ: **VCC (5V)**
- ສີສັ້ມ/ເຫຼືອງ: **Signal (PWM)**
- ການໃຊ້ງານ: ສົ່ງ **PWM signal** ເພື່ອຄວບຄຸມມຸມ

ຕົວຢ່າງການນຳໃຊ້:

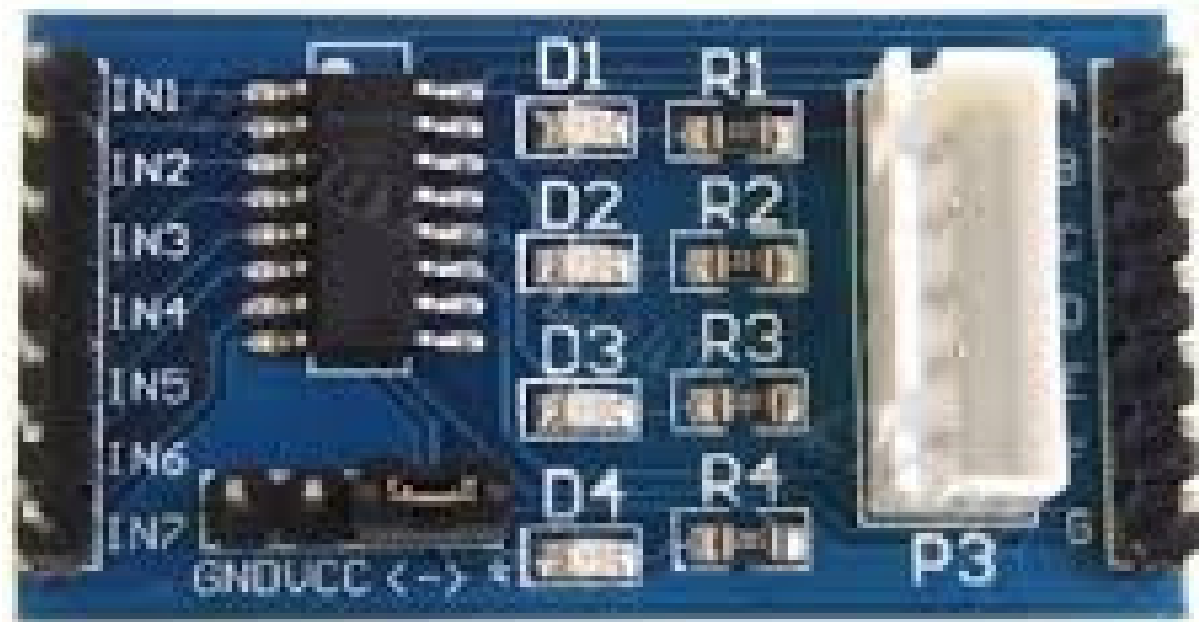
```
#include <Servo.h>  
Servo myservo;  
myservo.attach(9);  
myservo.write(90);
```

## 37. Stepper Motor



- ຄຳອະທິບາຍ: ມໍເຕີທີ່ໝຸນເປັນຂັ້ນຕອນ (**5.625°** ຕໍ່ **step**)
- **Pinout:**
- **5** ສາຍ: **4 coils + common**
- ການໃຊ້ງານ: ຕ້ອງໃຊ້ **driver board** ເພື່ອຄວບຄຸມ
- ຕົວຢ່າງການນຳໃຊ້: ໝຸນແມ່ນຍຳ, **CNC, 3D printer**

## 38. Stepper Motor Driver Board (ULN2003)



- ຄໍາອະທິບາຍ: **Driver board** ສໍາລັບຄວບຄຸມ **stepper motor 28BYJ-48**
- **Pinout:**
- **IN1, IN2, IN3, IN4: Control pins**
- **+, -: Power (5V, GND)**
- ການໃຊ້ງານ: ແປງສັນຍານ **logic** ເປັນກະແສຂັບມໍເຕີ

ຕົວຢ່າງການນໍາໃຊ້:

```
#include <Stepper.h>
Stepper myStepper(2048, 8,10,9,11);
myStepper.setSpeed(10);
myStepper.step(2048);
```

# ICs & Modules

## 39. Real-Time Clock Module (DS1302)

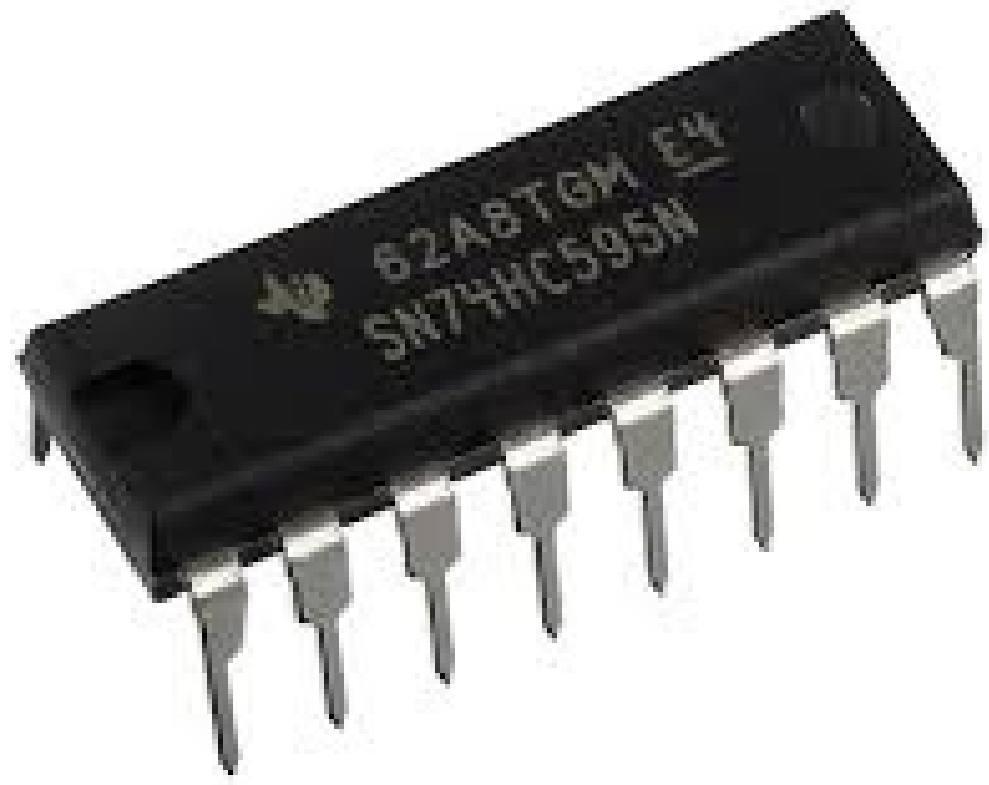


- ຄໍາອະທິບາຍ: ໂມດູນນາໂຄງການເວລາຈິງທີ່ມີແບັດເຕີຣີສໍາຮອງ
- **Pinout:**
- **VCC, GND, CLK, DAT, RST**
- ການໃຊ້ງານ: ເກັບຮັກສາເວລາ/ວັນທີແມ່ນວ່າປິດໄຟ

ຕົວຢ່າງການນໍາໃຊ້:

```
#include <DS1302.h>
DS1302 rtc(RST, DAT, CLK);
Time t = rtc.getTime();
Serial.print(t.hour);
Serial.print(":");
Serial.println(t.min);
```

## 40. 74HC595 Shift Register IC



- ຄຳອະທິບາຍ: IC ຂະຫຍາຍ **output** ຈາກ **3 pins** ເປັນ **8 pins**
- **Pinout:**
- **Data (DS), Clock (SHCP), Latch (STCP)**
- **Q0-Q7: 8 output pins**
- **VCC, GND**
- ການໃຊ້ງານ:
- ສົ່ງຂໍ້ມູນ **serial** ທາງ **DS**
- **Clock** ເພື່ອ **shift**
- **Latch** ເພື່ອອັບເດດ **output**

ຕົວຢ່າງການນຳໃຊ້:

```
digitalWrite(LATCH_PIN, LOW);  
shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, data);  
digitalWrite(LATCH_PIN, HIGH);
```