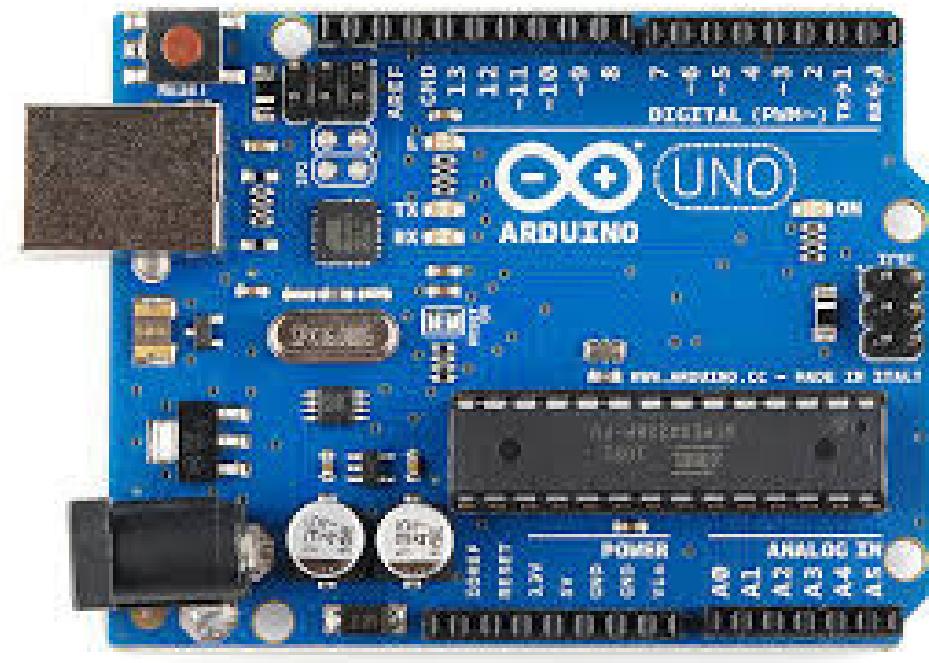


# Core Boards & Interfaces

## 1. Arduino Uno Board

- เป็นบอร์ดคอมพิวเตอร์ขนาดน้อย (**Microcontroller Board**) ที่เป็น symbology ให้กับโลก
- มีไมโครคอนโทรลเลอร์ ATmega328P อยู่ภายใน
- มีพอร์ต串รีสแลร์ 14 พอร์ต และ พอร์ตอ่านมาล็อก 6 พอร์ต
- สามารถโปรแกรมได้ง่ายด้วยภาษา **Arduino (C/C++)**



### วิธีใช้งาน

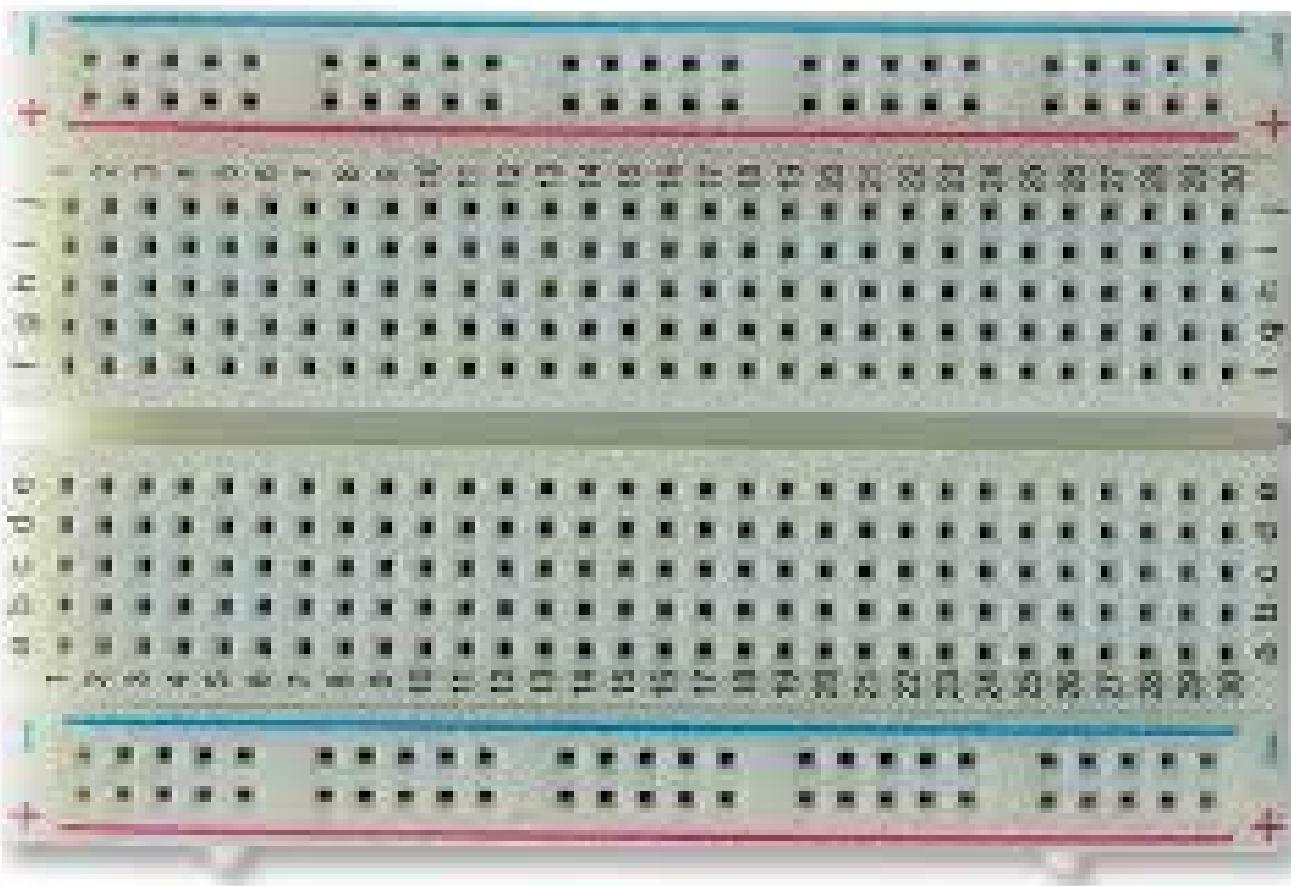
- ปะมวนผิวถาวรสั่งต่างๆตามโปรแกรมที่กำหนดไว้
- รับสัญญาณเข้า (**Input**) จากเซ็นเซอร์, บุ๊มเกิด
- ส่งสัญญาณออก (**Output**) ไปควบคุม LED, มีเติ, ลำโพง

### ตัวอย่างการนำใช้:

- ควบคุมไฟ LED ให้กะพริบ
- อ่านอุณหภูมิจากเซ็นเซอร์
- ควบคุมหุ่นยนต์, โครงการ IoT

## 2. Breadboard

- ແຜ່ນສໍາລັບຕໍ່ວົງຈອນໄຟຟ້າແບບບໍ່ຕ້ອງບັດກີ
- ມີຮູນໜ້ອຍໆຫຼາຍຮູ້ສໍາລັບສຽບສາຍ ແລະ ອຸປະກອນອີເລັກໂທຣນິກ
- ສາຍພາຍໃນຕໍ່ຮູ້ແຕ່ລະແຖວເຂົ້າກັນ
- ເໝາະສໍາລັບທົດລອງວົງຈອນກ່ອນຕໍ່ແບບຖາວອນ



ວິທີໃຊ້ງານ:

- ວິທີການຕໍ່ວົງຈອນ:
- ສຽບອຸປະກອນອີເລັກໂທຣນິກ (**LED**, ຕົວຕ້ານທານ) ເຂົ້າຮູ້
- ໃຊ້ສາຍຈັມເປີ້ (**Jumper Wire**) ຕໍ່ເຊື້ອມວົງຈອນ
- ແຖວແນວນອນບາງແຖວເປັນ + (ບວກ) ແລະ - (ລົບ) ສໍາລັບ ແຈກຈ່າຍໄຟ
- ຈຸດດີ:
- ບໍ່ຕ້ອງບັດກີ, ປ່ຽນແປງວົງຈອນໄດ້ຈ່າຍ
- ເໝາະກັບການຮຽນຮູ້ ແລະ ທົດລອງ

### 3. USB Cable

- สายเชื่อมต่อระหว่าง **Arduino** กับคอมพิวเตอร์
- ปีกกะตีเป็นสาย **USB Type-A** หรือ **Type-B**
- ใช้สำลับ:
  - อัปโหลดโปรแกรมเข้าบอร์ด
  - ส่งไฟเลี้ยง **Arduino (5V)**
  - สื่อสารข้อมูลระหว่าง **Arduino** กับคอม



- วิธีใช้งาน:
  - ต่อสาย: Plug **USB** ด้านหนึ่งเข้ากับ **Arduino**, อีกด้านเข้าคอมพิวเตอร์
  - ติดตั้งโปรแกรม: เปิด **Arduino IDE** เทิงคอม
  - ชุนໂຄ່ດ: ส້າງโปรแกรมທີ່ຕ້ອງການ
  - อัปโหลด: กົດປຸ່ມ **Upload** ເພື່ອສ່ົ່ງໂປຣແກຣມໄປບອດ
  - ເຊັກຜົນລັບ: ເບິ່ງການເຮັດວຽກຂອງໂຄງການ
  - ຕົວຢ່າງໂຄງການງ່າຍໆ

# Wiring & Connectors

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

- สายເຊື່ອມຕໍ່ທີ່ມີຫົວແຫຼມ (ພິນຜູ້) ສອງຂ້າງ
- ປົກກະຕິມີຫຼາຍສີ ເພື່ອຢ່າຍຕໍ່ການຈຳແນກວົງຈອນ
- ຄວາມຍາວປະມານ **10-30** ຊັງຕີແມັດ



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

- สายເຊື່ອມຕໍ່ທີ່ມີຫົວແຫຼມ (ພິນຜູ້) ສອງຂ້າງ
- ປົກກະຕິມີຫຼາຍສີ ເພື່ອຢ່າຍຕໍ່ການຈຳແນກວົງຈອນ
- ຄວາມຍາວປະມານ **10-30** ຊັງຕີແມັດ



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

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

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

- สายขดหมายด้วยสีสันต่างๆ ปลายทั้งสองเป็นรู **(female)**.
- Schematic / Pinout:**
- สายไฟที่ใช้เชื่อมต่อระหว่าง **PIN** ที่เป็น **male** (เช่น **Sensor, Module, Board**).



- การเริ่ดวง:
- ใช้เชื่อมต่อสัญญาณจากอุปกรณ์หนึ่งไปอีกหนึ่ง ย่างปอดไฟ โดยบล็อกบัดบอด.
- ใช้กับ **Breadboard** ห้องทดลองที่มี **pin male** จะสะดวกที่สุด.
- ตัวอย่างการนำไปใช้:
- เชื่อม **DHT11 Sensor** เข้ากับ **Breadboard** หรือ **LCD Module**.

## 7. 9V Battery Connector

- สายต่อที่ก้าน 9V มีฝาปิดความแข็งแรง และสายสองสาย (แดงและดำ).
- **Schematic / Pinout:**
- แดง = +9V
- 黑 = GND



- การเรียดวຽກ:
- สั่งพะลังงานจากก้าน 9V เข้าบอด Arduino ผ่านขา Vin หูพอดพาวเติ.
- ຕົວຢ່າງການນຳໃຊ້:
- ໃຊ້ໃນໂຄງງານທີ່ບໍ່ມີຄອມພິວເຕີເຊື່ອມຜ່ານ USB.
- ຄຳແນະນຳພິເສດ:
- ຄວນໃຊ້ກ้าน 9V ທີ່ມີຄຸນນະພາບດີ ເພື່ອຫຼຸດຄວາມສ່ຽງຂອງໄຟຕົກ.

# Basic Components

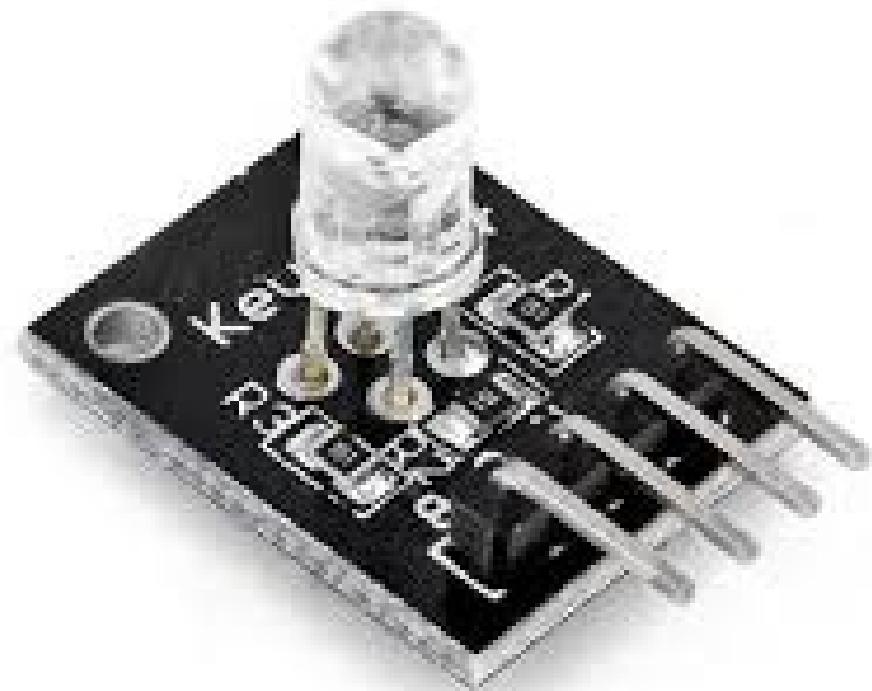
## 8. LEDs (Red: 5, Yellow: 5, Blue: 5, RGB: 1)

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



- การเร็ดวຽก:
- ใช้แสงสะท้อนเมื่อมีไฟใหญ่ผ่านโดยผ่านตัวต้าน (**Resistor**).
- ตัวอย่างการนำใช้:
  - ใช้ในโปรแกรม **Blink LED**, ไฟส่องทาง, ตัวเข้าส่องทาง [sensor].
  - คำแนะนำนำพิเศษ:
  - ใช้ตัวต้าน **220Ω** หรือ **330Ω** เพื่อป้องกันไฟเกินจัง LED ใหม่.

## 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)



1K (20pcs)



10K (20pcs)



22K (20pcs)

- 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-ສົມ-ທອງ
- ຕົວຢ່າງການນຳໃຊ້: ປົກປ້ອງ **LED**, **pull-up/pull-down resistors**

## 11. Push Buttons (x4 with Lids)



- ຄຳອະທິບາຍ: ປຸ່ມກົດແບບ **tactile switch**
- ການໃຊ້ງານ:
- 4 ຂົ: 2 ຄູ່ເຊື້ອມຕໍ່ກັນ
- ກົດແລ້ວຈະເຊື້ອມວົງຈອນ
- **Wiring:**
- ຂາໜຶ່ງຕໍ່ 5V
- ອີກຂາໜຶ່ງຕໍ່ **digital pin** + **10kΩ resistor to GND**

ຕົວຢ່າງການນຳໃຊ້:

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

## 12. Potentiometer ( $5k\Omega$ )



- ຄຳອະທິບາຍ: ຕົວຕ້ານທານປັບໄດ້  $5k\Omega$
- **Pinout:**
- ຂັ 1, 3: ຕໍ່ **5V** ແລະ **GND**
- ຂັ 2 (ກາງ): ຕໍ່ **Analog pin**
- ການໃຊ້ງານ: ຫມຸນເພື່ອປັບປຸງຄ່າແຮງດັນອອກ **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 ສັງສຽງ ຄົງທີ
- ການໃຊ້ງານ:
- ຕໍ່ + ກັບ **digital pin**
- ຕໍ່ - ກັບ **GND**
- ສັ່ງ **HIGH** ກໍ່ດັງ

ຕົວຢ່າງການນຳໃຊ້:

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

## 14. Passive Buzzer



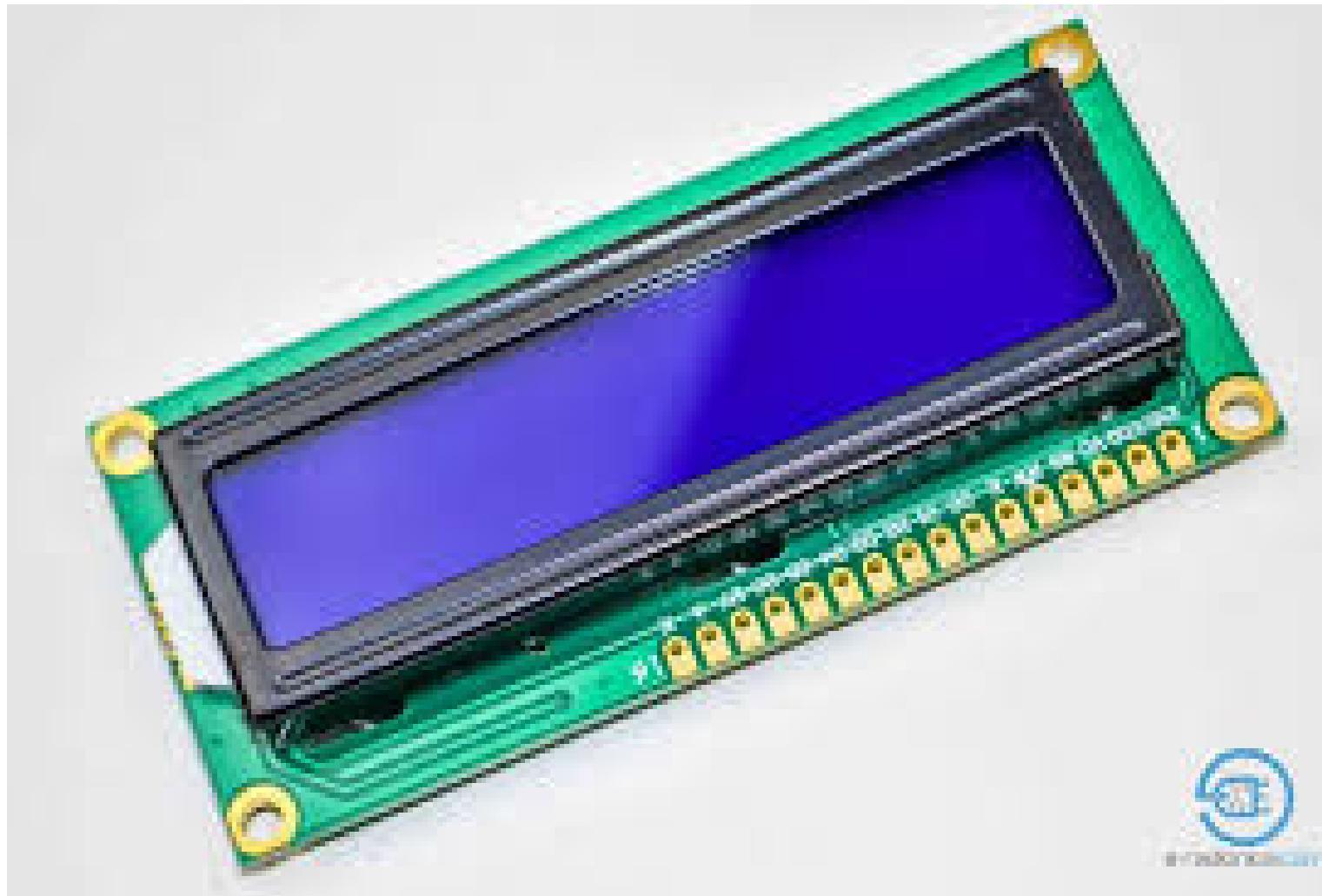
- ຄໍາອະທິບາຍ: **Buzzer** ທີ່ຕ້ອງການສັນຍານ **PWM** ເພື່ອສົ່ງສຽງ
- ການໃຊ້ງານ: ສາມາດສ້າງສຽງຄວາມຖືຕ່າງໆ ດ້ວຍ **tone()** **function**

ຕົວຢ່າງການນຳໃຊ້:

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

# Displays & Output

## 15. 16x2 LCD Display

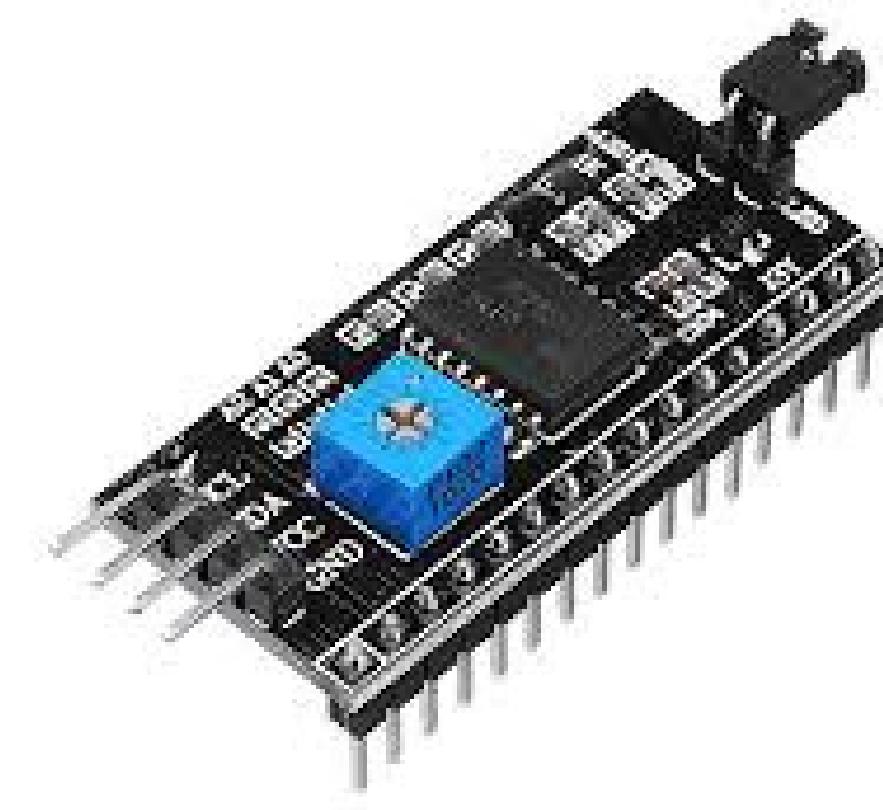


- ຄໍາອະທິບາຍ: ຈຳສະແດງຜົນ **LCD 16** ຕົວອັກສອນ x 2 ແຫວ
- **Pinout:**
- **VSS, VDD, VO, RS, RW, E**
- **D0-D7 (data pins)**
- **A (backlight +), K (backlight -)**
- ການໃຊ້ງານ: ສະແດງຂໍ້ຄວາມ, ຕົວເລກ, ສັນຍາລັກ

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

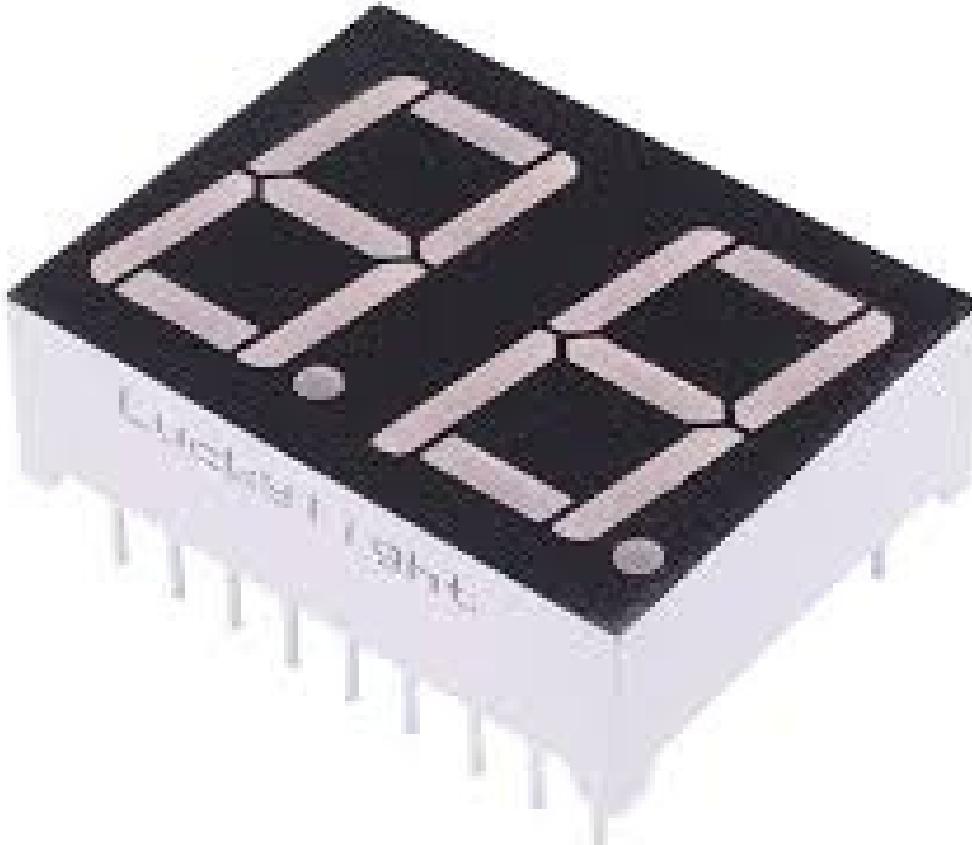


- តាមទិន្នន័យ: សម្រាប់បង្កើត LCD ជាន់ I2C ដើម្បីទូទាត់ pins
- **Pinout:**
- **GND, VCC, SDA (A4), SCL (A5)**
- ការងារ: ត្រួតពិនិត្យ **2 pins (SDA, SCL)** និង **6-8 pins**

ពិនិត្យការងារខាងក្រោម:

```
#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** ໄປຢັງ **segment** ທີ່ຕ້ອງການໃຫ້ ສະຫວັງ
- ຕົວຢ່າງການນຳໃຊ້: ສະແດງຕົວເລກ, **timer**

## 18. 4-Digit 7-Segment Display

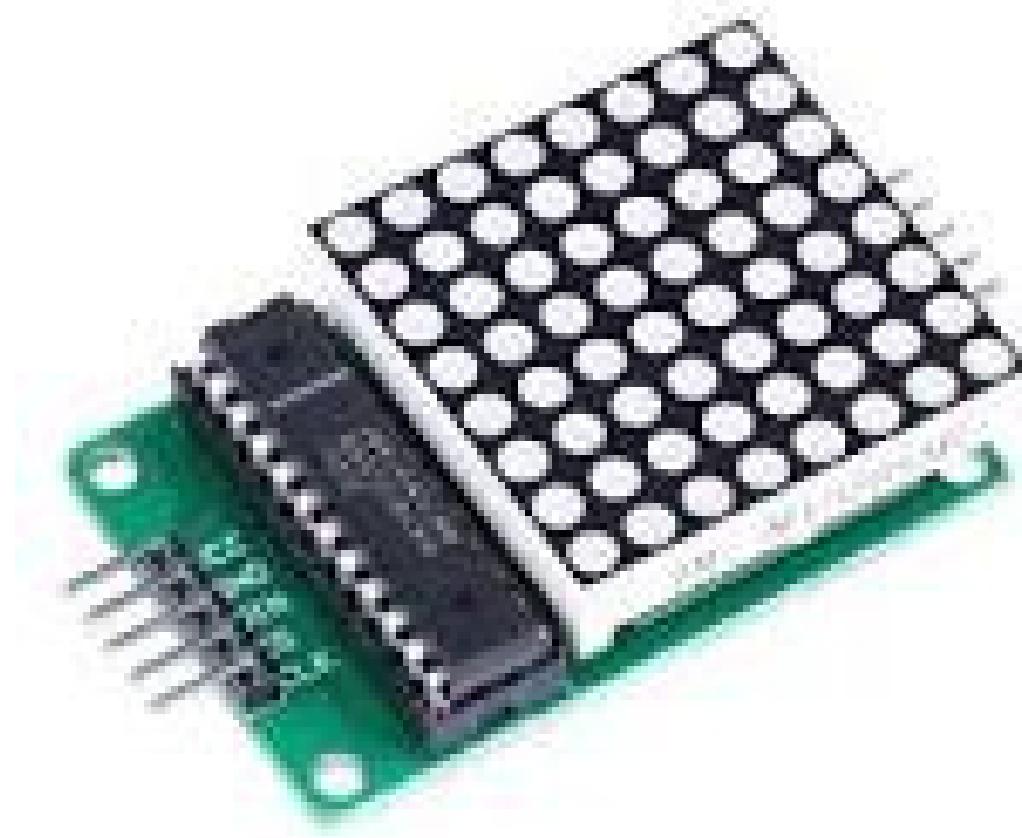


- ຄໍາອະທິບາຍ: ຈຳສະແດງຜົນ 4 ຕົວເລກ ດ້ວຍ **TM1637 driver**
- **Pinout:**
- **CLK, DIO, VCC, GND**
- ການໃຊ້ງານ: ໃຊ້ **TM1637 library** ເພື່ອຄວບຄຸມ

ຕົວຢ່າງການນຳໃຊ້:

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

## 19. 8x8 Dot Matrix Display



- ຄໍາອະທິບາຍ: ຈຳ 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



- តម្លៃ: ខ្សោយ (0-50°C) និងការបញ្ចូល (20-90%)
- Pinout:
- VCC (3.3-5V), DATA, GND
- ការងារ: ទាញរាយការងារជាមួយ DHT library

ព័ត៌មានលម្អិត:

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

## 21. LM35 Temperature Sensor

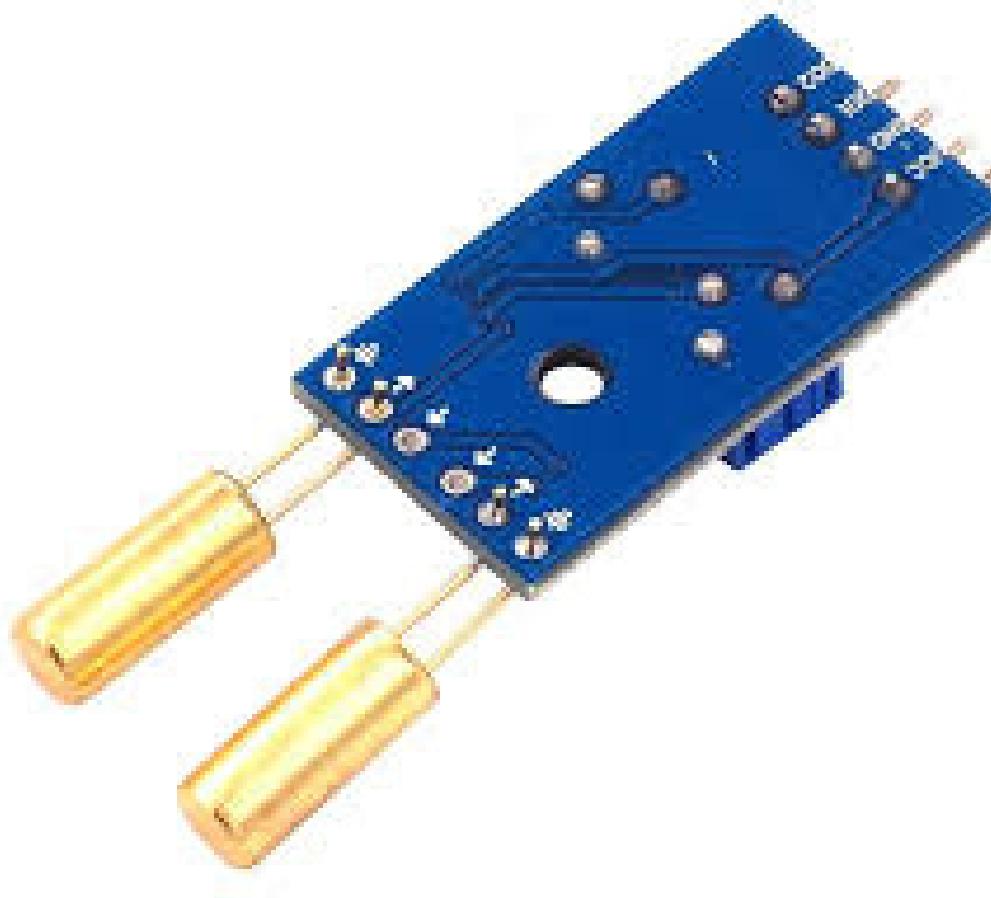


- តម្លៃពិបាយ: ខ្សោយខ្សោយតុនហេយូមបេប **analog** (-55 ហាត់ 150°C)
- **Pinout:**
- 1: VCC, 2: Vout, 3: GND
- ការងារ: ១៩៧រៀលដំណឹង **10mV/°C**

តើវាបានរាយការណា?

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

## 22. Tilt Sensor (x2)



- ຄໍາອະທິບາຍ: ເຊັນເຊີຕວດຈັບການເອງງ ມີລູກບານໃນຕົວ
- ການໃຊ້ງານ:
- ຕັ້ງຊື່: ເຊື່ອມວົງຈອນ (**LOW**)
- ເອງງ: ເປີດວົງຈອນ (**HIGH**)
- ຕົວຢ່າງການນຳໃຊ້: ກະດິ່ງເຕືອນ, ແນ, ຕວດຈັບຫາງ

## 23. Photoresistor / LDR (x3)



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

ຕົວຢ່າງການນຳໃຊ້:

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

## 24. PIR Motion Sensor



- คำอธิบาย: เดือนເຊີຕວດຈັບການເຄື່ອນໄຫວ (**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: ເຊັນເຊີວັດໄລຍະດ້ວຍຄື່ນສຽງ **ultrasonic** (**2-400cm**)
- **Pinout:**
- **VCC, Trig, Echo, GND**
- ການໃຊ້ງານ:
- ສົ່ງ **pulse**  $10\mu s$  ໃສ່ **Trig**
- ອ່ານເວລາ **pulse** ຈາກ **Echo**
- ໄລຍະ = ເວລາ  $\times$  **0.034 / 2**

ຕົວຢ່າງການນຳໃຊ້:

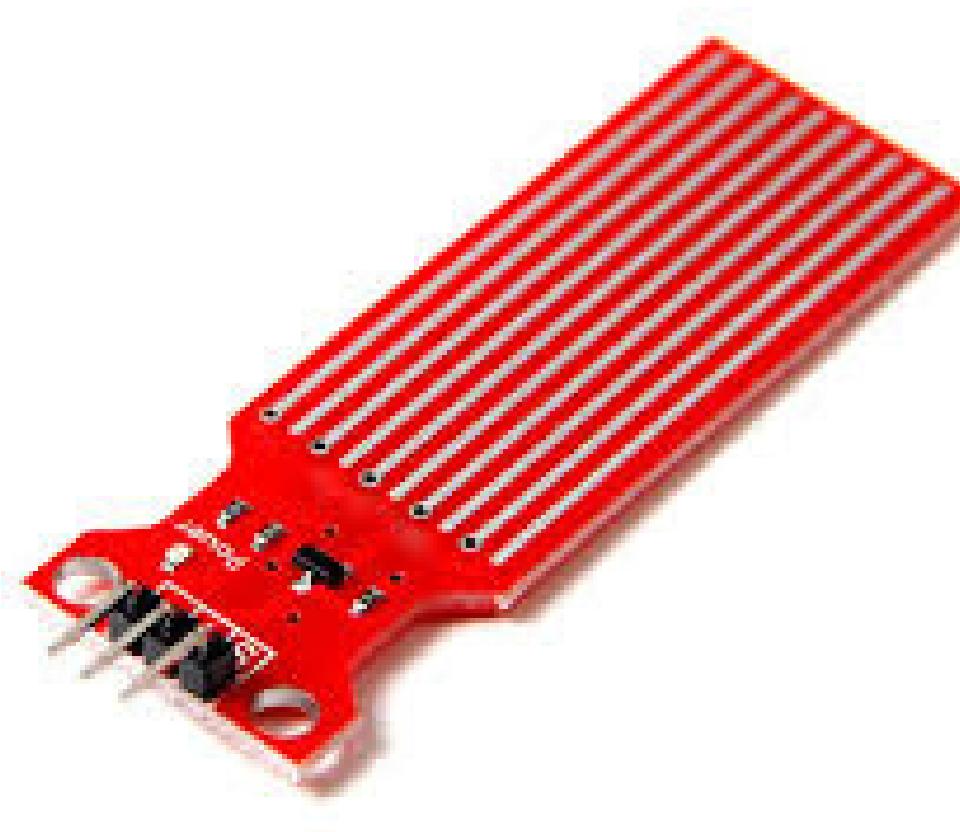
```
digitalWrite(TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG, LOW);
long duration = pulseIn(ECHO, HIGH);
float distance = duration * 0.034 / 2;
```

## 26. Sound Sensor



- คำอธิบาย: เดินเรือตัวดับสูงด้วยไมโครโฟน
- **Pinout:**
- **VCC, AO (analog), DO (digital), GND**
- งานใช้งาน:
  - **Analog:** ระดับสูง **0-1023**
  - **Digital:** เก็บ **threshold** หลัก
  - ตัวอย่างงานนำร่อง: **Voice activated switch, sound level meter**

## 27. Water Sensor



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

ຕົວຢ່າງການນຳໃຊ້:

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

## 28. Flame Sensor

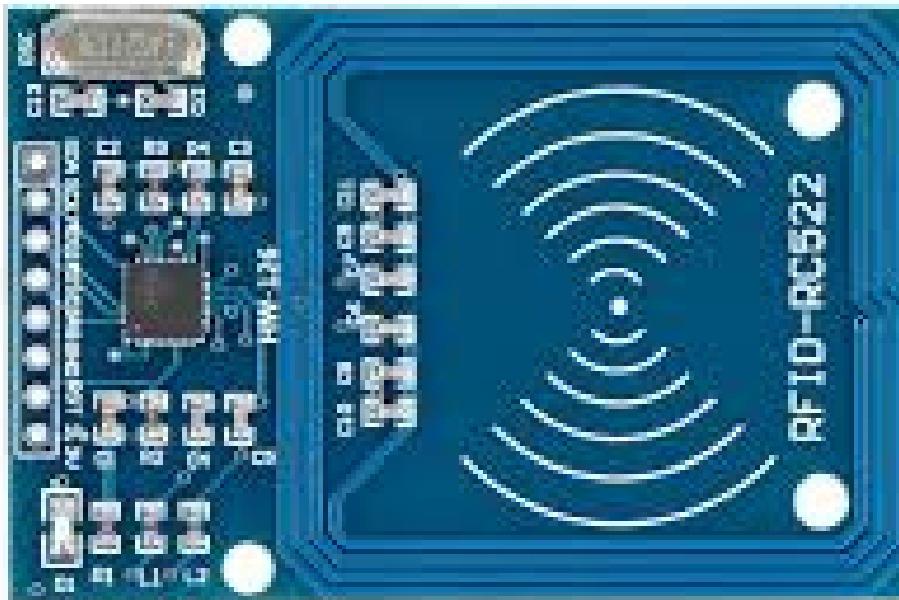


- ຄໍາອະທິບາຍ: ເຊັນເຊີຕວດຈັບແສງໄຟ / **infrared**
- **Pinout:**
- **VCC, DO (digital), AO (analog), GND**
- ການໃຊ້ງານ: ຕວດຈັບແສງໄຟໃນຊ່ວງ **760-1100nm**

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: ສື່ສາມດ້ວຍ **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: ເຊັນເຊີຮັບສັນຍານ **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



- ការប្រើប្រាស់: លើមិនគុមាមដែរ **infrared**
- ការងារ: សំណង់ **IR** ដើម្បីគុមាមអូបេភោន
- ពិនិត្យការងារ: គុមាម **LED**, មំពើ, ហលិវិត

## 33. Joystick Module

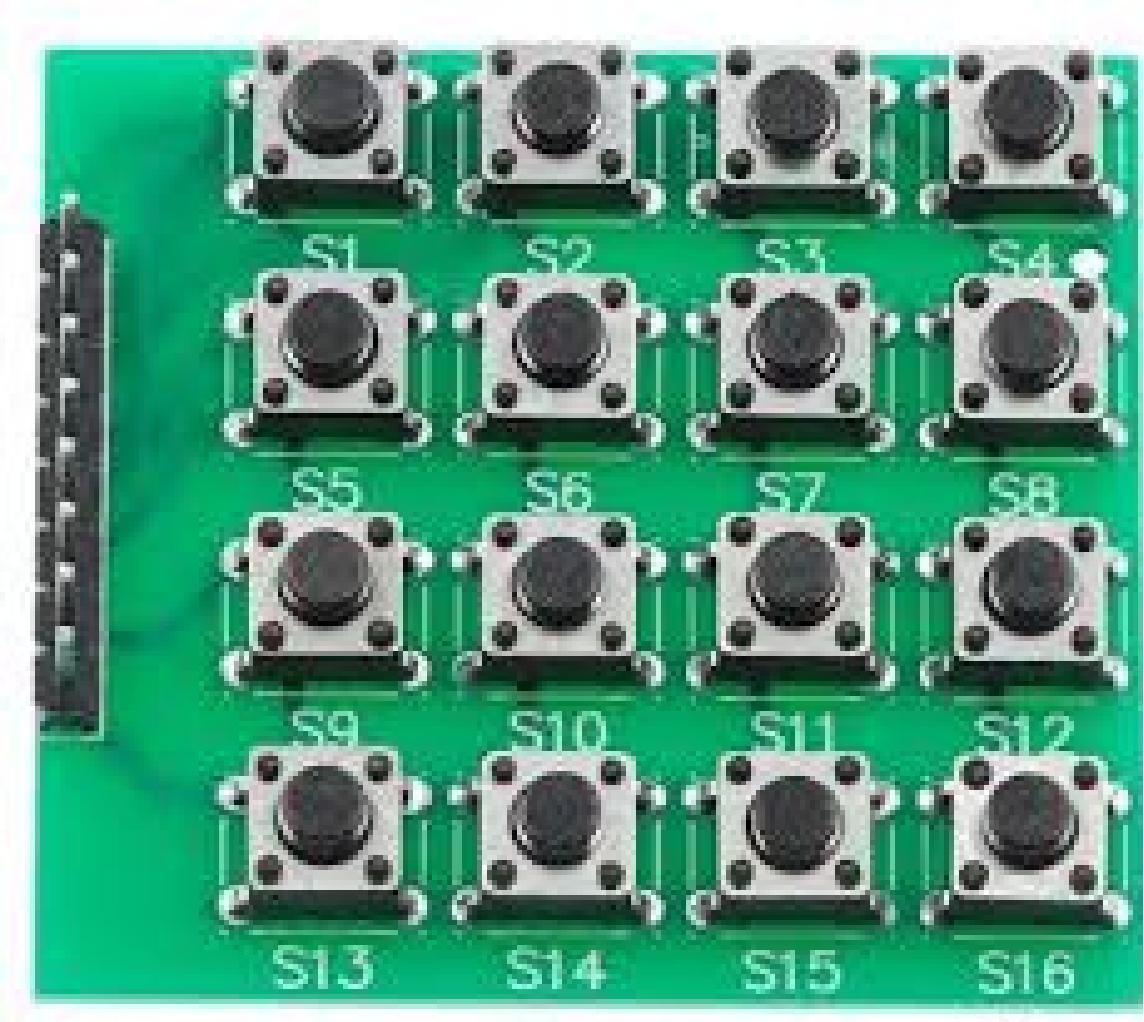


- ຄໍາອະທິບາຍ: ໂມດູນຄວບຄຸມແບບ **analog 2** ການ (X, Y)  
+ ປຸມກົດ
- **Pinout:**
- **GND, +5V, VRx, VRy, SW**
- ການໃຊ້ງານ:
- **VRx/VRy: analog 0-1023 (ນາງ ~512)**
- **SW: digital button (LOW ເມື່ອກົດ)**

ຕົວຢ່າງການນຳໃຊ້:

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

## 34. 4x4 Matrix Keyboard Module



- ຄໍາອະທິບາຍ: ແປ່ນພິມ **16** ປຸມ (**0-9, A-D, \*, #**)
- **Pinout:**
- **8 pins: 4 rows, 4 columns**
- ການໃຊ້ງານ: ສະແກນ **rows/columns** ເພື່ອຕວດຈັບປຸມທີ່ກົດ

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** ເພື່ອປີດ **relay** (ເຊື່ອມ **COM-NO**)
- ສົ່ງ **HIGH** ເພື່ອປີດ **relay**

ຕົວຢ່າງການນຳໃຊ້:

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

# Motors & Drivers

## 36. Servo Motor



- ຄໍາອະທິບາຍ: ມຳເຕີຄວບຄຸມມຸມ **0-180°**
- **Pinout:**
- ສິນ້າຕານ/ດຳ: **GND**
- ສີແດງ: **VCC (5V)**
- ສີສົ່ມ/ເຫຼືອງ: **Signal (PWM)**
- ການໃຊ້ງານ: ສັງ **PWM signal** ເພື່ອຄວບຄຸມມຸມ

ຕົວຢ່າງການນຳໃຊ້:

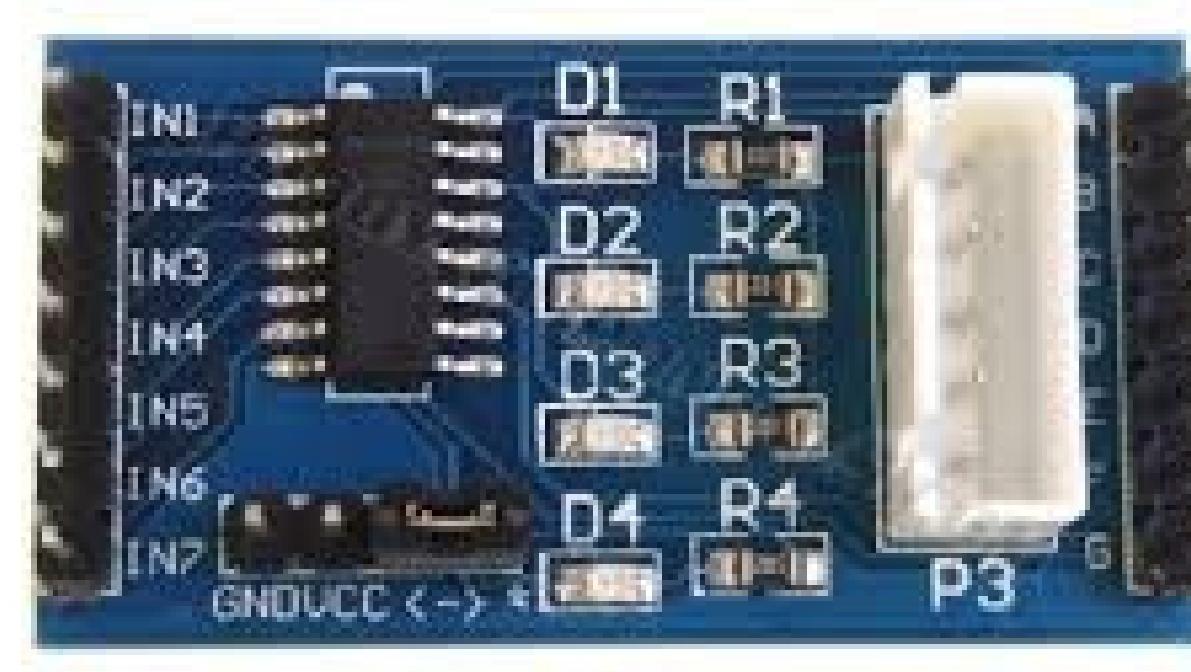
```
#include <Servo.h>
Servo myservo;
myservo.attach(9);
myservo.write(90);
```

## 37. Stepper Motor



- ຄຳອະທິບາຍ: ມີເຕີທີ່ໝູນເປັນຂັ້ນຕອນ ( $5.625^\circ$  ຕໍ່ step)
- **Pinout:**
- 5 ສາຍ: **4 coils + common**
- ການໃຊ້ງານ: ຕ້ອງໃຊ້ **driver board** ເພື່ອຄວບຄຸມ
- ຕົວຢ່າງການນຳໃຊ້: ໝູນແມ່ນຍຳ, 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** ເປັນກະແສຂັບມຳເຕີ

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: ໂມດູນນາໂນກາເວລາຈິງທີ່ມີແບັດເຕີຣີສໍາຮອງ
- **Pinout:**
- **VCC, GND, CLK, DAT, RST**
- ການໃຊ້ງານ: ເກັບຮັກສາເວລາ/ວັນທີແມ່ນວ່າປິດໄພ

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



- คำอธิบาย: IC จะขยาย **output** จาก 3 **pins** เป็น 8 **pins**
- **Pinout:**
- **Data (DS), Clock (SHCP), Latch (STCP)**
- **Q0-Q7: 8 output pins**
- **VCC, GND**
- งานใช้งาน:
- สั่งข้อมูล **serial** ทาง **DS**
- **Clock** เพื่อ **shift**
- **Latch** เพื่ออับเดด **output**

ตัวอย่างงานนำໃຊ້:

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