

ຊື່ ແລະ ນາມສະກຸນ: ທ. ຄິມຫັນ ສີປະເສດ ລຳດັບ: 1 ຫ້ອງ:3COM2

## ວຽກข້າມ MCU & MPU



## 🔌 core boards & interfaces

## 1. Arduino Uno board

**ການໃຊ້ງານຂອງອຸປະກອນ:** Arduino Uno ແມ່ນບອດ Microcontroller ມັນເຮັດວຽກໂດຍການອ່ານຄ່າຈາກອຸປະກອນຮັບເຂົ້າ (Input) ເຊັ່ນ: ເຊັ່ນເຊີ້ງ ຫຼື Button, ປະມວນຜົນຕາມໂປຣແກຣມທີ່ຂຽນໄວ້, ແລະ ສັ່ງການໄປຫາອຸປະກອນ Output ເຊັ່ນ: LED, Monitor ຫຼື ຈໍສະແດງຜົນ.

🔌 Arduino Uno มีขาเข้า-ออก (PINOUT) แบบห้องเป็น 3 พากส์รวมถึง:  
ขาดิจิตอล (Digital Pins)

- มີທັງໝົດ **14 21** (0 ເຖິງ 13).
  - ໃຊ້ສໍາລັບການປ້ອນ/ສົ່ງຂໍ້ມູນແບບ **ປຶດ (HIGH)** ຫຼື **ປຶດ (LOW)**.
  - **PWM Pins (~)**: 6 ຂົ້ວ (3, 5, 6, 9, 10, 11) ທີ່ສາມາດຄວບຄຸມຄ່າແບບ Analog ໄດ້ (ເຊັ່ນ: ປັບຄວາມສະຫວັງຂອງໄຟ).
  - **ສໍ່ສານ Serial:**
    - RX (0): ຮັບຂໍ້ມູນ.
    - TX (1): ສົ່ງຂໍ້ມູນ.

## ୧୨ Analog (Analog Input Pins)

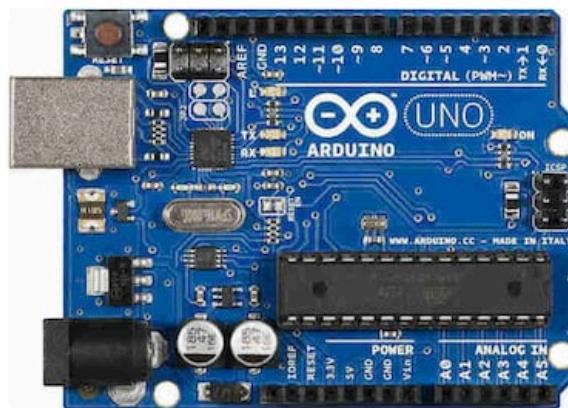
- มีทั้งหมด **6 แผ่น** (A0 ถึง A5).
  - ให้สำลับงานอ่านได้ทิ่ปูนแบบย่อๆ เมื่อจะใช้งาน (เช่น: วัดเส้น, วัดอุณหภูมิ).

## ឧបផែង (Power Pins)

- GND: ຂາພື້ນດິນ (Ground) ຫຼື ຂາລີບ.
  - 5V ແລະ 3.3V: ຂາສະໜອງໄຟຟ້າອອກນອກໃຫ້ອຸປະກອນອື່ນ.
  - VIN: ຂາສໍາລັບຕໍ່ເຫົາຈ່າຍໄຟພາຍນອກ (7V ຫາ 12V).

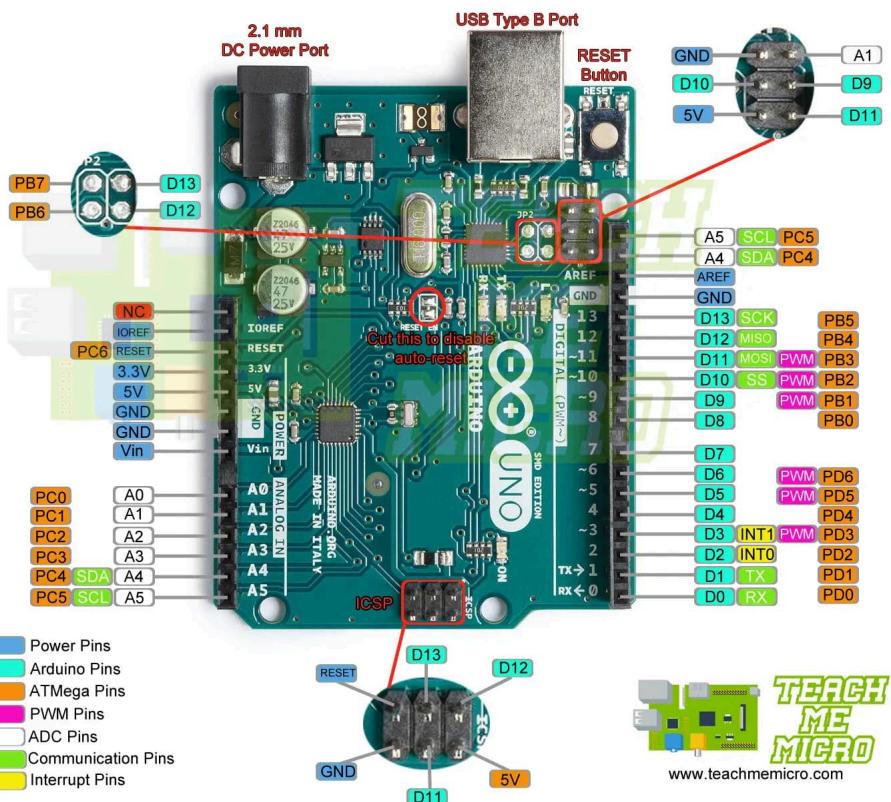
- RESET: ຂາສໍາລັບຮູບສະຕາດໂປຣແກຣມ.

## ຮູບພາບອຸປະກອນຕົວຈິງ:



## ទូរស័ព្ទបច្ចនមនុប Schematic Diagram :

# **ARDUINO UNO R3 SMD PINOUT**



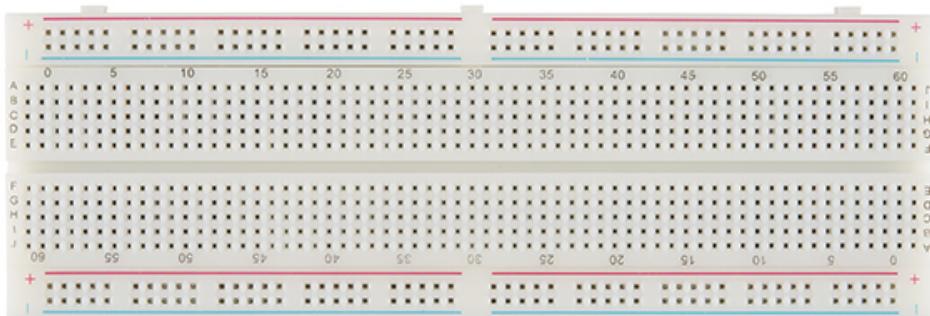
## ຕົວຢ່າງການນຳໃຊ້:

Blink LED: ໃຊ້ຂາ Digital Output ເພື່ອຄວບຄຸມການເປີດ-ປິດໄຟ LED ເປັນຈັງຫວະ

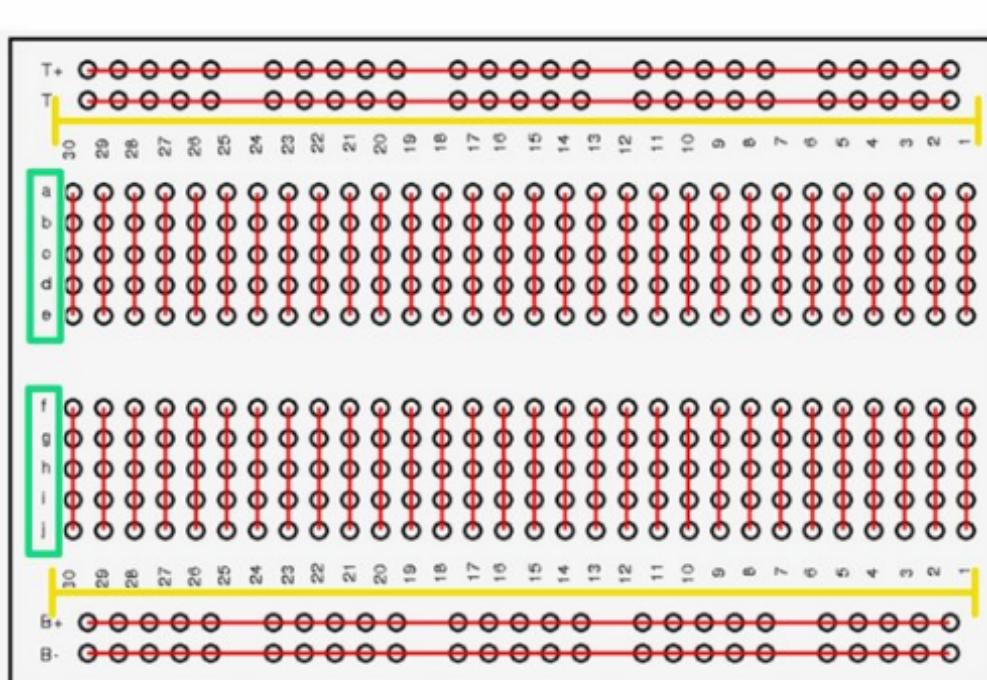
## 2. Breadboards

ការໃຊ້ງານຂອງອຸປະກອນ: Breadboard ແມ່ນຄື່ອງມີສໍາລັບການສ້າງແບບຈໍາລອງວິຈອນເອເລັກໂຕຣນິກ ເຮັດໃຫ້ສາມາດເຊື່ອມຕໍ່ອຸປະກອນຕໍ່າງໆ (ເຊັ່ນ: Arduino, Resistors, LEDs, ICs) ເຂົ້າກັນ ເພື່ອທິດສອບ ແລະ ປັບປຸງວິຈອນໄດ້ຢ່າຍ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram :



ຕົວຢ່າງການນຳໃຊ້:

ສ້າງວິຈອນໄຟ LED ກະພົບໂດຍການວາງ LED, ຕົວຕ້ານທານ, ແລະ ສາຍເຊື່ອມຕໍ່ກັບ Arduino Uno.

## 3. USB Cable

ການໃຊ້ງານຂອງອຸປະກອນ:

- ການສໍ່ສານຂໍ້ມູນ: ໃຊ້ສໍາລັບການອັບໂຫຼດໂປຣແກຣມ (Sketch) ຈາກຄອມພິວເຕີໄປຍັງ Microcontroller ຂອງ Arduino. ນອກຈາກນີ້ຢັ້ງໃຊ້ສໍາລັບການສໍ່ສານແບບ Serial (ເຊັ່ນ: ການສ້ົງຂໍ້ມູນຈາກ Arduino ກັບໄປ

ຄອມພິວເຕີເພື່ອດີບກ ຫຼື ສະແດງຜົນ).

- **ການຈ່າຍໄຟ:** ໃຊ້ສໍາລັບຈ່າຍໄຟຟ້າ (+5V) ຈາກຄອມພິວເຕີໄປບັງ Arduino Uno ເພື່ອໃຫ້ບອດສາມາດຮັດວຽກໄດ້.

ຮູບພາບອຸປະກອນຕົວຈີງ:



ຕົວຢ່າງການນຳໃຊ້:

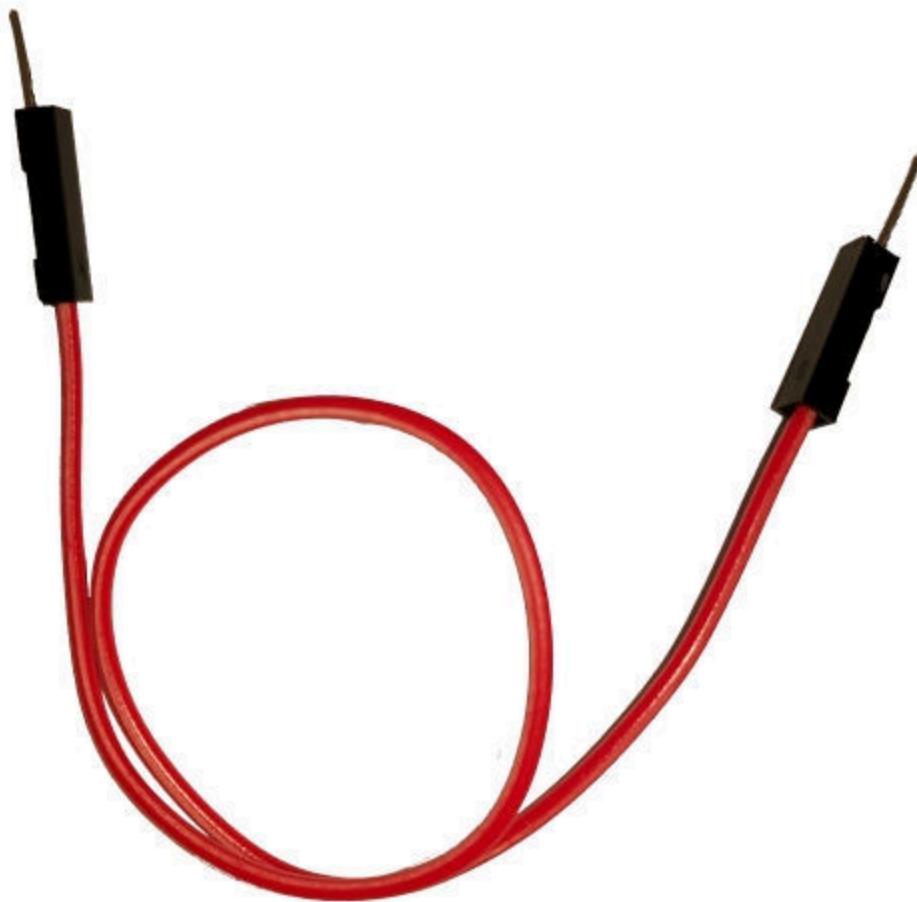
- Programming: ເຊື່ອມຕໍ່ Arduino ເຂົ້າກັບຄອມພິວເຕີເພື່ອຂຽນ ແລະ ອັບໂຫຼດໂປຣແກຣມໃໝ່.
- Serial Monitor: ໃຊ້ສາຍນີ້ເພື່ອເປີດ Serial Monitor ໃນ Arduino IDE ເພື່ອເປັ້ນຂໍ້ຄວາມ ຫຼື ຄ່າຈາກເຊັນເຊີ້ມທີ່ Arduino ສິ່ງມາ.

## **Wiring & Connectors**

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

**ການໃຊ້ງານຂອງອຸປະກອນ:** Jumper Wires ແມ່ນສາຍສໍາລັບເຊື່ອມຕໍ່ຫາງໄຟຟ້າລະຫວ່າງອີງປະກອບຕ່າງໆໃນວິຈອນເອເລັກໄຕເນີກ, ໂດຍສະເພາະແມ່ນການເຊື່ອມຕໍ່ລະຫວ່າງ Arduino Uno, Breadboard, ແລະ Modules/Components ອື່ນໆ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



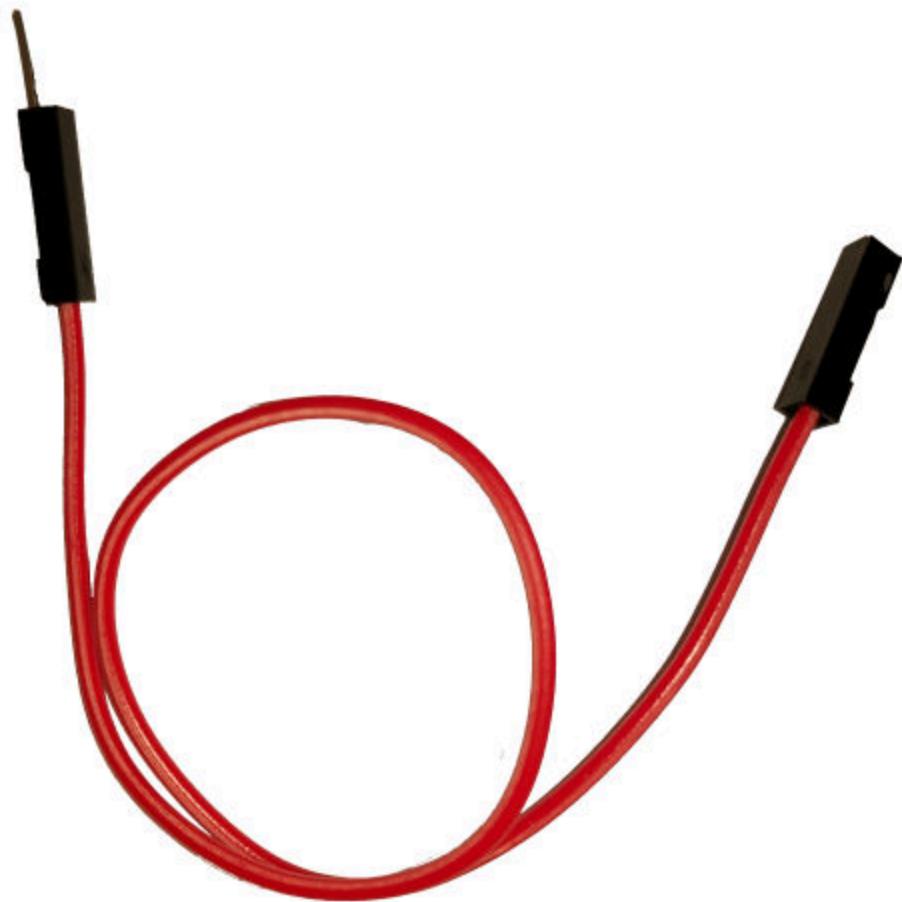
#### ຕົວຢ່າງການນຳໃຊ້:

**M-to-M:** ເຊື່ອມຕໍ່ຂາ Digital ຂອງ Arduino ໄປຫາຈຸດໜຶ່ງໃນ Breadboard.

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

**ການໃຊ້ງານຂອງອຸປະກອນ:** Jumper Wires ແມ່ນສາຍສໍາລັບເຊື່ອມຕໍ່ທາງໄຟຟ້າລະຫວ່າງອີງປະກອບຕ່າງໆໃນ  
ວົງຈອນເອເລັກໂຕຣນິກ, ໂດຍສະເພາະແມ່ນການເຊື່ອມຕໍ່ລະຫວ່າງ Arduino Uno, Breadboard, ແລະ  
Modules/Components ອື່ນໆ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



#### ຕົວຢ່າງການນຳໃຊ້:

**M-to-F:** ເຊື້ອມຕໍ່ຂາ Power (5V) ຂອງ Arduino ໄປຫາຂາ VCC ຂອງ Sensor Module.

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

**ການໃຊ້ງານຂອງອຸປະກອນ:** Jumper Wires ແມ່ນສາຍສຳລັບເຊື້ອມຕໍ່ທາງໄຟຟ້າລະຫວ່າງອີງປະກອບຕ່າງໆໃນ  
ວົງຈອນເອເລັກໂຕຣນິກ, ໂດຍສະເພາະແມ່ນການເຊື້ອມຕໍ່ລະຫວ່າງ Arduino Uno, Breadboard, ແລະ  
Modules/Components ອື່ນໆ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

**F-to-F:** ແຊ້ອມຕໍ່ຂາສັນຍານ I2C (SDA, SCL) ໂດຍກິຈລະຫວ່າງ Arduino ກັບ I2C LCD Module.

#### 7. 9V Battery Connector

**ການໃຊ້ງານຂອງອຸປະກອນ:** ໃຊ້ເປັນຫົວຕໍ່ ແບດເຕີຣີ 9V ກັບ Arduino Uno ເພື່ອຈ່າຍໄຟຟ້າໃຫ້ກັບ Arduino Uno ໂດຍກິຈຈາກແບດເຕີຣີ 9V

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

ໃຊ້ພື້ນຈ່າຍໄຟເປັນແຫຼ່ງພະລັງງານສໍາລັບຫຼຸນຍືນນ້ອຍ ຫຼື ລະບົບຄວບຄຸມໄລຍະໄກທີ່ໃຊ້ Arduino.

## Basic Components

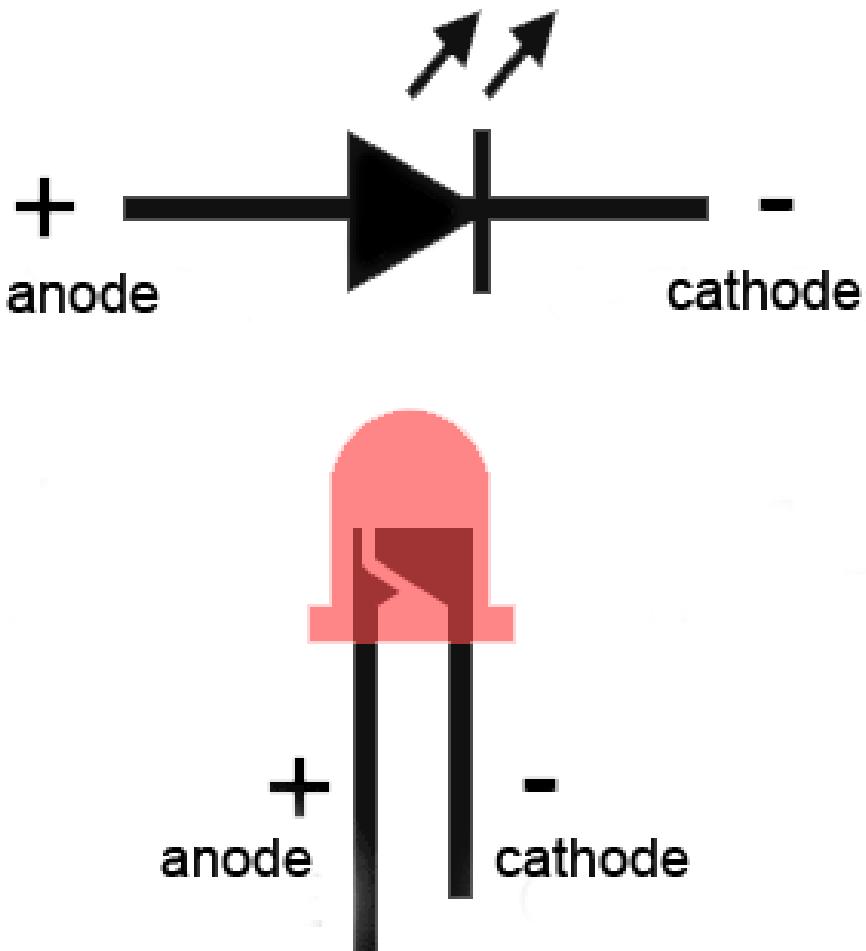
### 8. LEDs (Red, Yellow, Blue, RGB)

**ການໃຊ້ງານຂອງອຸປະກອນ:** LEDs ແມ່ນໄດ້ໂອດທີ່ປ່ອຍແສງອອກມາເມື່ອມີກະແສໄຟຟ້າໄຫຼຸ້ນຜ່ານໃນທິດທາງທີ່ຖືກຕ້ອງ (ຈາກ Anode ໄປຫາ Cathode). ໃຊ້ເປັນຕົວຊີ້ບອກສະຖານະພາບ (ເຊັ່ນ: ເປີດ/ປິດ, ເຮັດວຽກ), ຫຼື ເປັນອົງປະກອບຂອງການສະແດງຜົນທາງສາຍຕາ. ຕ້ອງຕໍ່ກັບຕົວຕ້າມທານ (ເຊັ່ນ:  $220\Omega$ ) ເພື່ອຈຳກັດກະແສໄຟຟ້າ ແລະ ບັງດັບກັບ LED ເສຍຫາຍ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



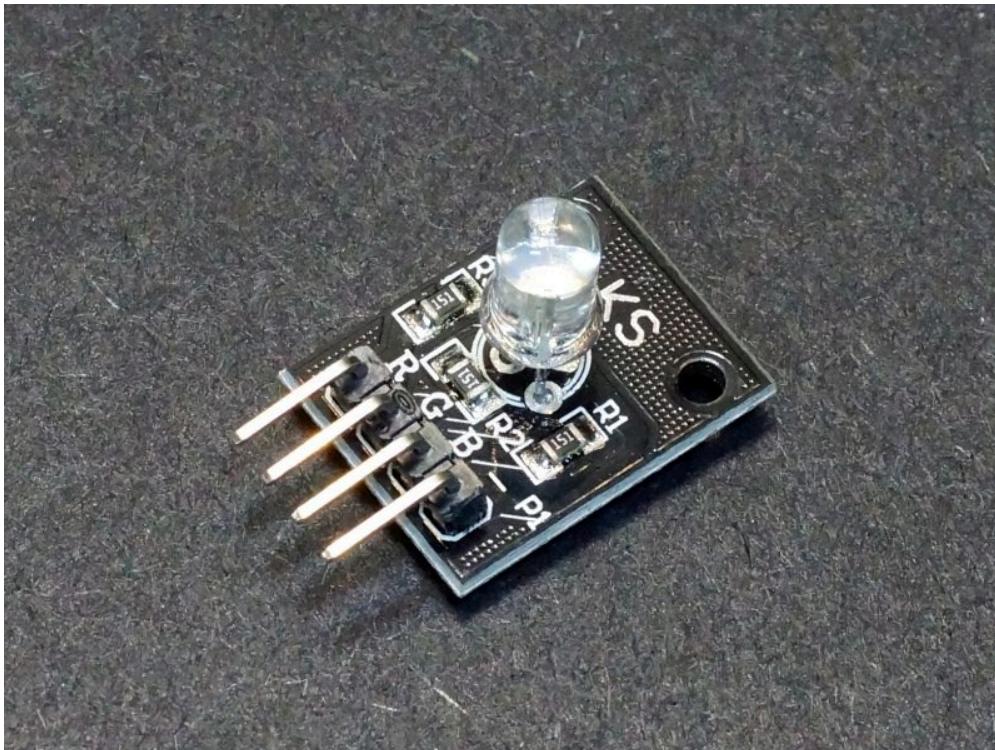
## ពិរយោគការមាໃຊ់

**Status Indicator:** ໃຊ້ໄຟສີແຕງເພື່ອຊື້ບອກວ່າລະບົບປິດຢູ່ ແລະ ສຶບຍົງວເພື່ອຊື້ບອກວ່າລະບົບປິດຢູ່.

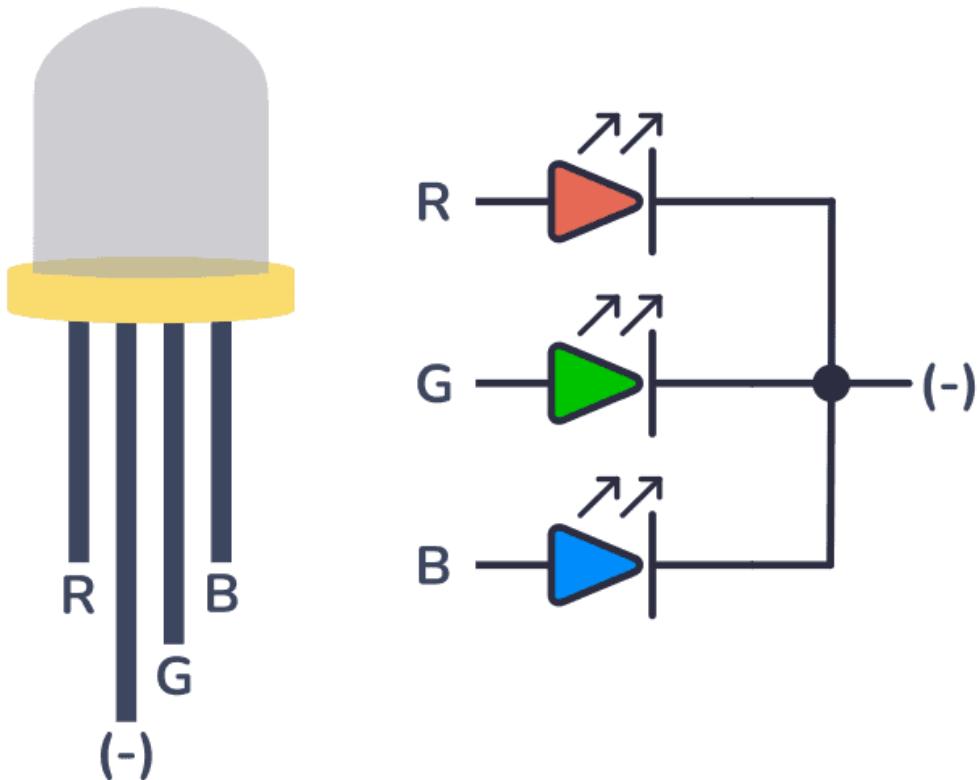
## 9. RGB module

**ການໃຊ້ງານຂອງອະປະກອນ:** RGB LED ແມ່ນການລວມເອົາໄຟ LED ສາມດອກ (ແດງ, ຂຽວ, ພ້າ) ໄວ້ໃນກ່ອງ ດຽວ. ມັນຮັດວຽກໂດຍອີງໃສ່ຫຼັກການປະສົມສືແບບພື່ມ (Additive Color Mixing). ໂດຍການຄວບຄຸມຄວາມເຂັ້ມ ຂອງແສງສາມສີຫຼັກນີ້, ມັນສາມາດຜະລິດສີທີ່ເຕັມຕ່າງກັນໄດ້ຫາຍລ້ານສີ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



ຕົວຢ່າງການນຳໃຊ້:

- **RGB Mood Light:** ໃຊ້ PWM ຂອງ Arduino ເພື່ອຄວບຄຸມຄວາມເຂັ້ມຂອງສີແດງ, ສີຂຽວ, ແລະ ສີຟ້າ ຂອງ RGB LED ເພື່ອສ້າງສີປະສົມທີ່ແຕກຕ່າງກັນ.

10. Resistors (220Ω, 1kΩ, 10kΩ)

**ການໃຊ້ງານຂອງອຸປະກອນ:** ຕົວຕ້ານທານມີຫັ້ນທີ່ຫຼັກຄືການ **ຕ້າມ** ການໄຫຼ້ຂອງກະແສໄຟຟ້າໃນວົງຈອນ. ມັນຖືກໃຊ້ເພື່ອ:

1. **ຈຳກັດກະແສໄຟຟ້າ:** ບ້ອງກັນບໍ່ໃຫ້ກະແສໄຟຟ້າຫຼາຍເກີນໄປໃຫ້ເຂົ້າໄປໃນອຸປະກອນທີ່ອ່ອມໄຫວ (ເຊັ່ນ: LED). (ມັກໃຊ້  $220\Omega$  ຫາ  $1k\Omega$  ສໍາລັບ LED).
2. **ແບ່ງແຮງດັນ (Voltage Divider):** ໃຊ້ສອງຕົວຕ້ານທານຮ່ວມກັນເພື່ອຫຼັດແຮງດັນລົງ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:

"resistor\_schematic.png|500" could not be found.

ຕົວຢ່າງການນຳໃຊ້:

**LED Protection:** ດໍາຕົວຕ້ານທານ  $220\Omega$  ກັບ LED ເພື່ອປ້ອງກັນບໍ່ໃຫ້ LED ເສຍຫາຍ.

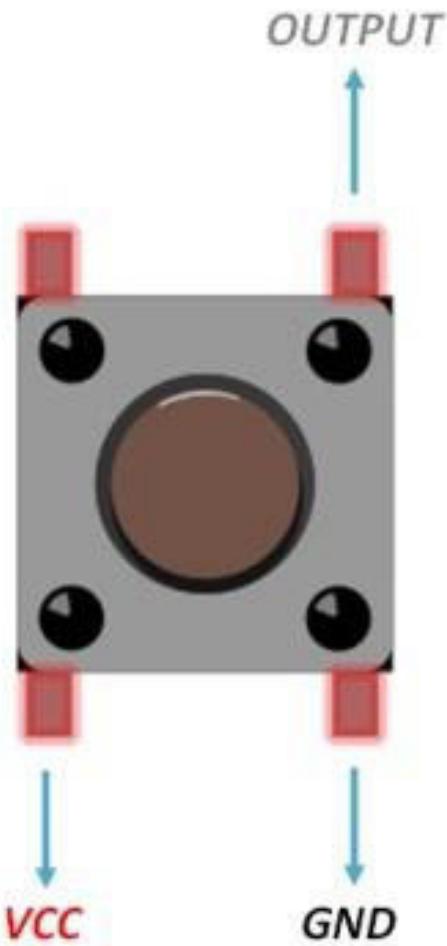
#### 11. Push Buttons (x4 with Lids)

**ການໃຊ້ງານຂອງອຸປະກອນ:** ບຸ່ມກົດຮັດໜ້າທີ່ເປັນອຸປະກອນຮັບເຂົ້າ (Input) ທີ່ປິດ ຫຼື ເປີດວົງຈອນຊົ່ວຄາວ. ເມື່ອກົດບຸ່ມ, ມັນຈະຮັດໃຫ້ກະແສໄຟຟ້າໃຫ້ ຫຼື ບຸ່ມໄຫ້ ເຊິ່ງ Arduino ສາມາດອ່ານການປ່ຽນແປງສະຖານະນີ້ໄດ້ຜ່ານຂາ Digital Input.

ຮູບພາບອຸປະກອນຕົວຈິງ:

![plush button|400](plush\_button.jpg |400)

ຮູບອຸປະກອນແບບ Schematic Diagram:



#### ຕົວຢ່າງການນຳໃຊ້:

- **Control Light:** ໃຊ້ບຸ່ມກົດເພື່ອເປີດ/ປິດໄຟ LED.
- **Mode Selector:** ໃຊ້ບຸ່ມກົດຫຼາຍອັນເພື່ອເລືອກໂທນາຄາມເຮັດວຽກທີ່ແຕກຕ່າງກັນສໍາລັບໂປຣເຈັກ.

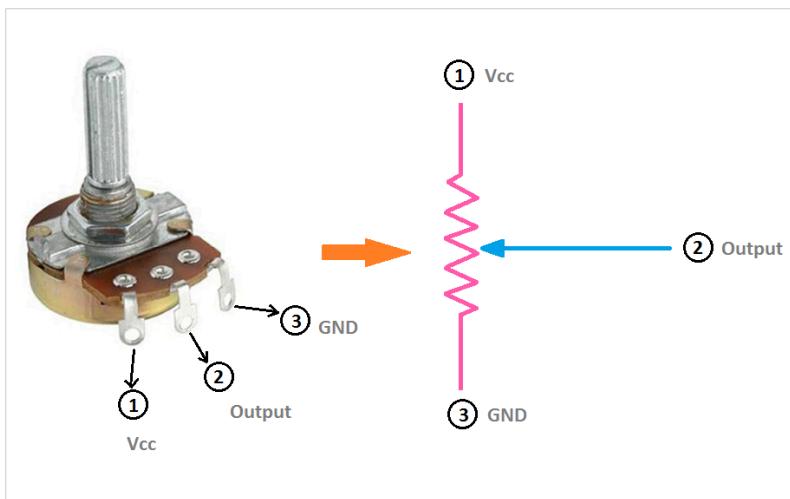
#### 12. Potentiometer ( $5k\Omega$ )

**ການໃຊ້ງານຂອງອຸປະກອນ:** Potentiometer ແມ່ນຕົວຕ້ານທານທີ່ສາມາດປັບຄ່າໄດ້. ໃນການນຳໃຊ້ກັບ Arduino, ມັນມັກຖືກໃຊ້ເປັນ **Voltage Divider** ເພື່ອຜະລິດສັນຍານແຮງຕັນອະນາລັອກທີ່ປຽນແປງໄດ້ (ຈາກ 0V ຫາ 5V) ໃຫ້ Arduino ອ່ານຄ່າຜ່ານຂາ Analog Input. ຜູ້ໃຊ້ສາມາດປັບຄ່າໄດ້ໂດຍການໝູນປຸ່ມ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



ຕົວຢ່າງການນຳໃຊ້:

**Input Control:** ໃຊ້ເພື່ອຄວບຄຸມຄວາມສະຫວ່າງຂອງ LED (ໂດຍການປັບດໍາ PWM) ຫຼື ຄວາມໄວຂອງມຶເຕີ.

### 13. Active Buzzer

**ການໃຊ້ງານຂອງອຸປະກອນ:** Buzzer ໃຊ້ສໍາລັບການຜະລິດສຽງ (Output).

**Active Buzzer:** ໃຊ້ສໍາລັບສັນຍານເຕືອນ ຫຼື ສຽງແຈ້ງເຕືອນແບບບ່າຍໆ (ເຊັ່ນ: ສຽງ Beep ເມື່ອມີຂໍຜິດພາດ). ແຊ້ອມຕໍ່ກັບຂາ Digital ຂອງ Arduino ເພື່ອເປີດ/ປິດ.

### ຮູບພາບອຸປະກອນຕົວຈິງ:



### ຮູບອຸປະກອນແບບ Schematic Diagram:

“buzzer-pinout.jpg|400” could not be found.

#### ຕົວຢ່າງການນຳໃຊ້:

**Alarm System:** ໃຊ້ Active Buzzer ເພື່ອສິ່ງສັນຍານເຕືອນເມື່ອເຊັ້ນເຊີກວດພົບການເຄື່ອນໄຫວ.

#### 14. Passive Buzzer

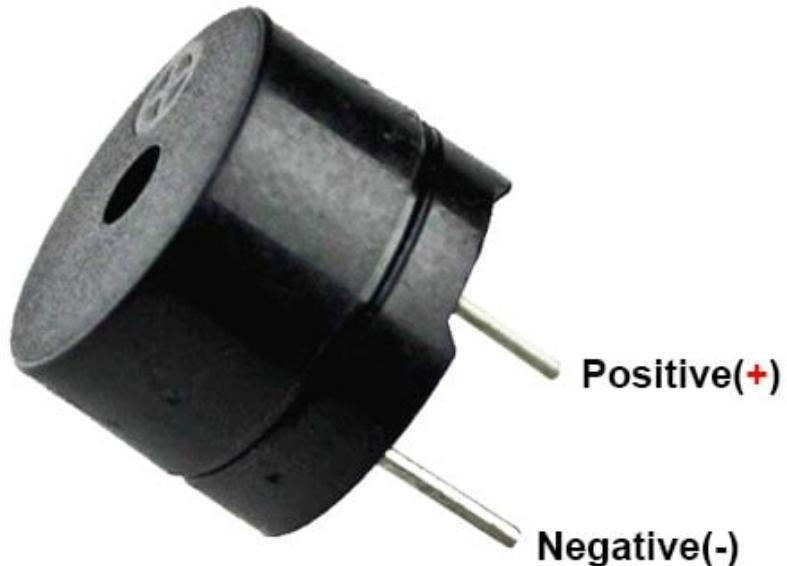
**ການໃຊ້ງານຂອງອຸປະກອນ:** Buzzer ໃຊ້ສໍາລັບການຜະລິດສຽງ (Output).

**Passive Buzzer:** ໃຊ້ສໍາລັບສຽງທີ່ສັບຊ້ອນກວ່າ ເຊັ່ນ: ສຽງດິນຕຣີ, ສຽງເອັບເຟັກ ຫຼື ສຽງເຕືອນທີ່ມີຄວາມຖືປ່ຽນແປງໄດ້. ເຊື່ອມຕໍ່ກັບຂາ PWM ຂອງ Arduino ເພື່ອຄວບຄຸມຄວາມທີ່.

### ຮູບພາບອຸປະກອນຕົວຈິງ:



### ຮູບອຸປະກອນແບບ Schematic Diagram:



## ពិរីយោងការមាំខ្លួន

**Playing Music:** ໃຊ້ Passive Buzzer ເພື່ອຫຼັ້ນເງິນ່າຍ່າໂດຍການປ່ຽນຄວາມຖືຂອງສັນຍານທີ່ປ້ອນເຂົ້າ.

# Displays & Output

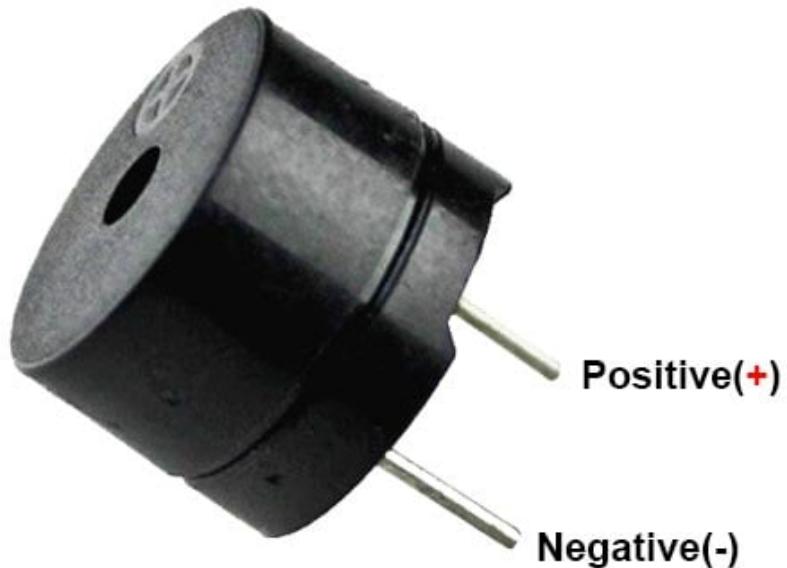
## 15. 16x2 LCD display

**ການໃຊ້ງານຂອງອຸປະກອນ:** ຈະ LCD 16x2 ໃຊ້ສໍາລັບການສະແດງເຢີນຂໍ້ຄວາມສັນ່ງ (16 ຕົວອັກສອນຕໍ່ແຖວ, ສອງແຖວ). ມັນເປັນອຸປະກອນ Output ທີ່ສໍາຄັນສໍາລັບການສະແດງຄ່າຈາກເຊັ້ນເຊື້, ສະຖານະພາບຂອງລະບົບ ຫຼື ຂໍ້ຄວາມຂອງຜູ້ໃຊ້ໂດຍກິງ. ເນື່ອງຈາກມັນໃຊ້ຂາຫຼາຍ (6-11 ຂາ) ໃນການເຊື່ອມຕໍ່ກັບ Arduino, ມັນມີກຈະຖືກໃຊ້ຄຸ້ກັບ I2C Serial Adapter Board ເພື່ອຫຼັດການໃຊ້ຂາຂອງ Arduino ລົງທຶນພຽງ 2 ຂາ (SDA, SCL).

## ຮູບພາບອຸປະກອນຕົວຈິງ:



## ຮູບອຸປະກອນແບບ Schematic Diagram:



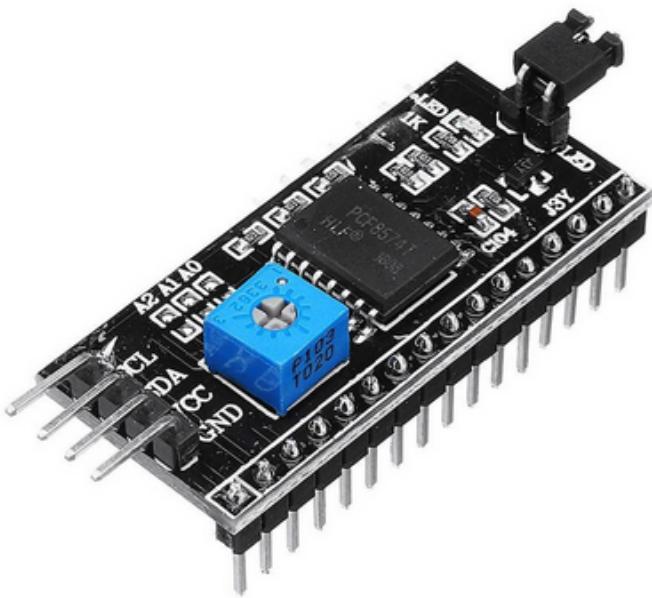
### ຕົວຢ່າງການນຳໃຊ້:

- **Weather Station:** ສະແດງຄ່າອຸນຫະພູມ ແລະ ຄວາມຊຸມຊື່ນທີ່ອ່ານໄດ້ຈາກ DHT11.
- **Counter:** ສະແດງຈຳນວນຄັ້ງທີ່ບຸ່ມກິດຖືກກິດ
- **Menu:** ສະແດງເມນຸເລືອກໂໝນດການເຮັດວຽກຂອງໂປຣເຈັກ.

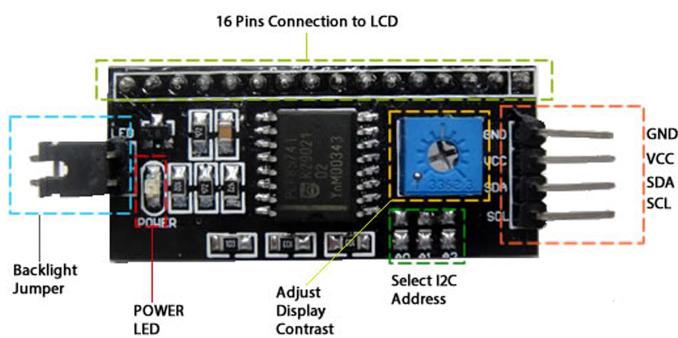
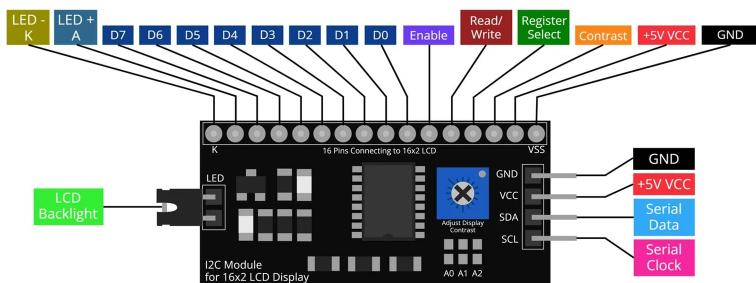
## 16. I2C Serial Adapter board module

**ການໃຊ້ງານຂອງອຸປະກອນ:** ໂມດຸນນີ້ມີ IC ຂັບ (ເຊັ່ນ: PCF8574) ທີ່ເຮັດຫຼາທີ່ປົງນການສື່ສານແບບ Parallel ຂອງຈຳ LCD ໃຫ້ເປັນການສື່ສານແບບອະນຸກິມ **I2C (Inter-Integrated Circuit)**. ການໃຊ້ໂມດຸນນີ້ຈະຫຼັດຈຳນວນຂາທີ່ຕ້ອງເຊື່ອມຕໍ່ຈາກ Arduino ໄປຫາ LCD ຈາກ 7-11 ຂາ ລົງໜູ້ອພງ **4 ຂົງ** (VCC, GND, SDA, SCL), ເຊິ່ງຊ່ວຍປະຍັດຂາ I/O ຂອງ Arduino ໄວສໍາລັບອຸປະກອນອໍ່ນັ້ງ.

## ຮູບພາບອຸປະກອນຕົວຈິງ:



## ຮູບອຸປະກອນແບບ Schematic Diagram:



## ຕົວຢ່າງການນຳໃຊ້:

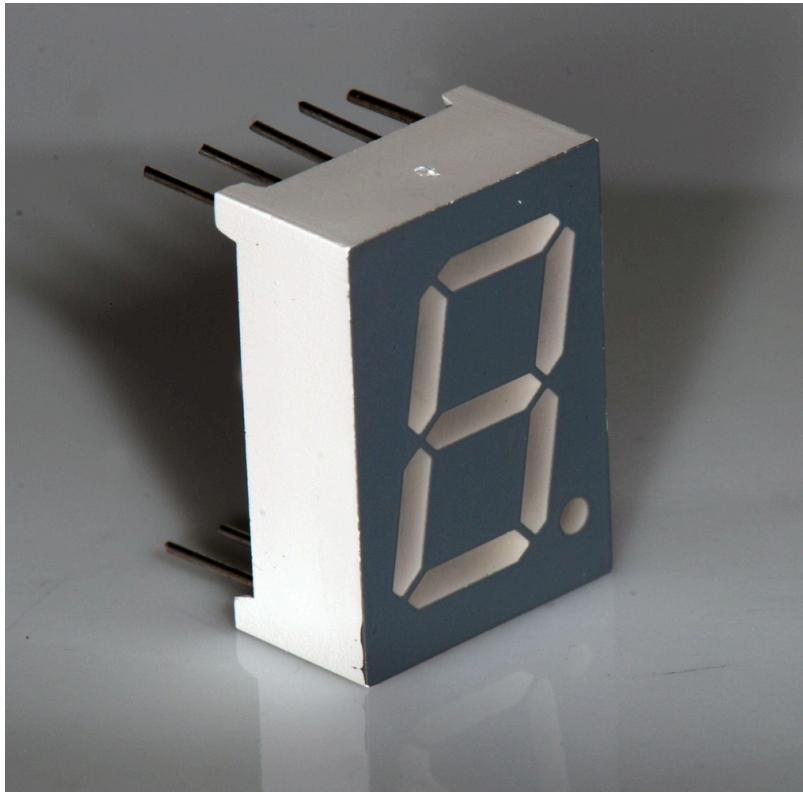
- **Minimal Wiring Project:** ເຊື້ອມຕໍ່າໆ LCD ກັບ Arduino ໂດຍໃຊ້ພຽງ 4 ສາຍເຝື້ອປະຫັດຂາ I/O.
- **Data Logging:** ໃຊ້ເຝື້ອສະແດງຜົນຂໍ້ມູນແຊັ້ນເຊີ້ງຫາຍອນ ພ້ອມທ່າງໃຊ້ຂາອື່ນໆສໍາລັບ SD Card Module ຫຼື Module ອື່ນໆ.

## 17. 7-segment display (Common Cathode +)

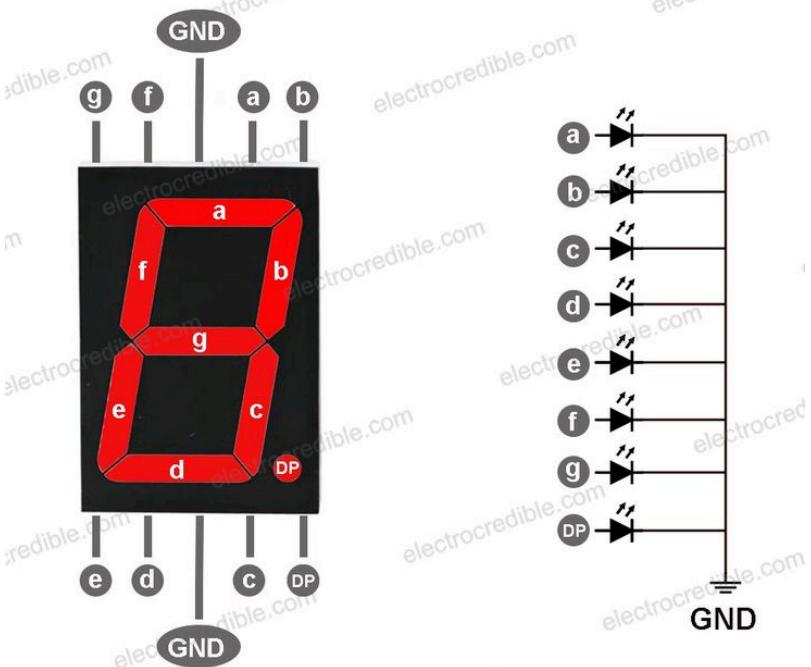
ການໃຊ້ງານຂອງອຸປະກອນ: ຈຳ 7-Segment ໃຊ້ສໍາລັບການສະແດງເີນຕົວເລກ (0-9) ແລະ ຕົວອັກສອນບາງຕົວ.

- **Common Cathode:** ເພື່ອເປີດໄຟ LED ໃນສ່ວນໃດໜຶ່ງ, ຕ້ອງຈ່າຍໄຟ **HIGH (5V)** ໄປຫາຂາຄວບຄຸມສ່ວນນັ້ນ. ຂາ Common ຕ້ອງຕໍ່ກັບ GND.
- ການຄວບຄຸມ: ຕ້ອງໃຊ້ 8 ຂາ Digital Output ຂອງ Arduino ເພື່ອຄວບຄຸມ Segment ຫັງໝົດ, ເຊິ່ງມັກຈະໃຊ້ IC ຂັບ (ເຊັ່ນ: 74HC595) ເພື່ອຫຼັດການໃຊ້ຂາ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



## ពិរយោគការមាиз់

- **Simple Counter:** ສະແດງຕົວເລກນັບຂຶ້ນ ຫຼື ນັບລົງ.
  - **Game Score:** ສະແດງຄະແນນໃນເກມແບປ່ງໆຢ່າງ.

## 18. 4-Digit 7-Segment Display

**ການໃຊ້ງານຂອງອຸປະກອນ:** ໃຊ້ສໍາລັບການສະແດງຜົນຕົວເລກ 4 ຫຼັກ. ເພື່ອໃຫ້ສາມາດສະແດງຜົນ 4 ຫຼັກໄດ້ພ້ອມກັນ, ມັນໃຊ້ເທິກິນິກທີ່ເອີ້ນວ່າ **Multiplexing** (ການເປີດ-ປິດຫຼັກຕ່າງໆຢ່າງໄວວາ), ເຮັດໃຫ້ຕາຂອງຄົນເຫັນຄືກັບວ່າສະແດງຜົນພ້ອມກັນຕະຫຼອດເວລາ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



ຕົວຢ່າງການນຳໃຊ້:

- **Digital Clock/Timer:** ສະແດງຊື່ວໂມງ ຫຼື ເວລານັບຖອຍຫຼັງ.
- **Voltmeter/Termometer:** ສະແດງຄ່າເຮັງດັນ ຫຼື ອຸນທະພູມທີ່ມີຕົວເລກ 4 ຫຼັກ.

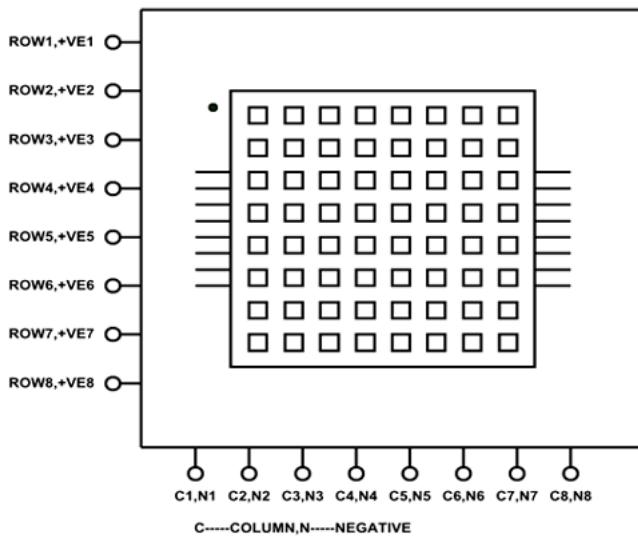
#### 19. 8x8 Dot Matrix display

ການໃຊ້ງານຂອງອຸປະກອນ: ຈະ 8x8 Dot Matrix ໃຊ້ສໍາລັບການສະແດງຜົນຮູບພາບ, ຮູບແບບ (Patterns), ຂໍ້ຄວາມແບບເລື່ອນ (Scrolling Text) ຫຼື ຮູບພາບແບບ Pixel Art ທີ່ມີຄວາມລະອຽດຕຳ. ເຮັດວຽກໂດຍການຄວບຄຸມແຕ່ລະຈຸດ (Pixel) ໃນຕາຫັ້ງ 8x8 ໂດຍແຍກກັນ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນເບີນ Schematic Diagram:



ຕົວຢ່າງການນຳໃຊ້:

- **Scrolling Text:** ສະແດງຂໍ້ຄວາມຂະໜາດສັນແບບເລື່ອນໄປມາ.
- **Simple Animation:** ສ້າງພາບເຄື່ອນໄຫວ່າຍ່າງ ຫຼື ຮູບພາບ Pixel Art.
- **Game Display:** ສະແດງຜົນໃນເກມແບບເກົ່າ (ເຊັ່ນ: Tetris ຫຼື Snake).

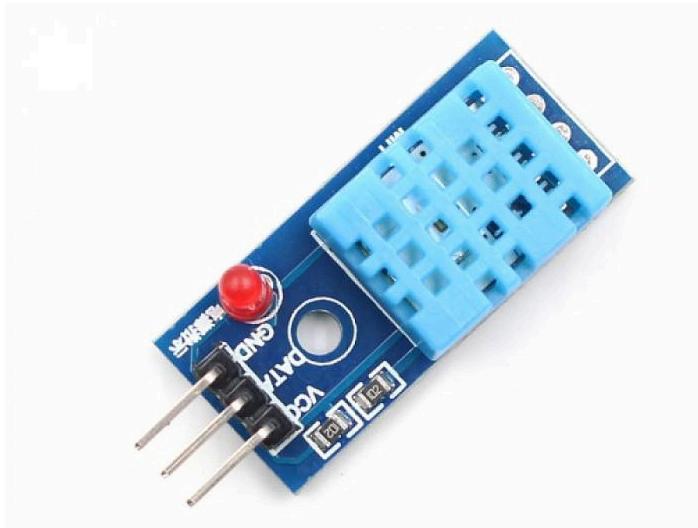


## Sensors & Input Modules

## 20. Temperature and humidity sensor (DHT11)

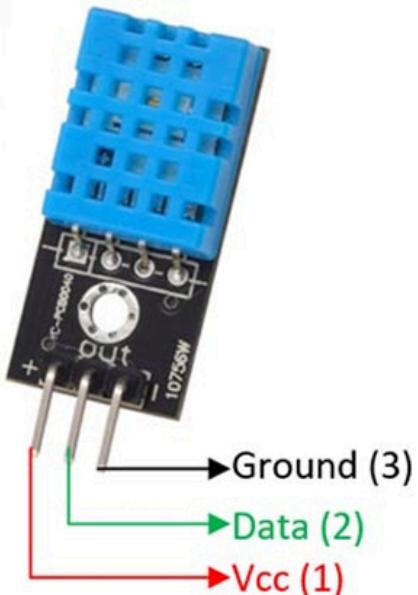
ການໃຊ້ງານຂອງອຸປະກອນ: DHT11 ໃຊ້ສໍາລັບວັດແທກຄ່າອຸນຫະພູມ (Temperature) ແລະ ຄວາມຊູມຊື່ນສໍາພັດ (Relative Humidity) ຂອງອາກາດອ້ອມຂ້າງ. ມັນສື່ງຂໍ້ມູນອອກມາເປັນສັນຍານດິຈິຕອລ (Digital Signal) ພຽງ 1 ສາຍ, ເຮັດໃຫ້ການເຊື່ອມຕໍ່ງໆຢັ້ງ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:

## DHT11 Pinout



[www.Circuits-DIY.com](http://www.Circuits-DIY.com)

### ຕົວຢ່າງການນຳໃຊ້:

- **Environmental Monitoring:** ສ້າງລະບົບຕິດຕາມສະພາບແວດລ້ອມໃນເຮືອນ ຫຼື ສວນ.
- **Thermostat:** ຄວບຄຸມເຄື່ອງປັບອາກາດ ຫຼື ພັດລົມໂດຍອີງໃສ່ຄ່າອຸນຫະພຸມ ແລະ ຄວາມຂຸ່ມຂຶ້ນ.

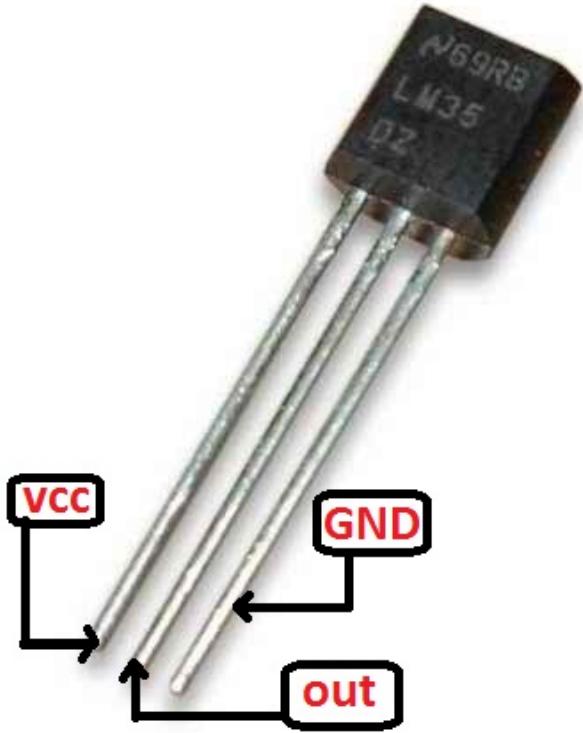
### 21. LM35 Temperature Sensor

#### ການໃຊ້ງານຂອງອຸປະກອນ:

LM35 ແມ່ນເຊັນເຊື່ອນຫະພຸມທີ່ມີຄວາມແມ່ນຢໍາສູງ ເຊິ່ງໃຫ້ຄ່າອອກມາເປັນສັນຍານ ອະນາລັອກ (Analog Voltage) ໂດຍກົງເປັນອີງສາເຊລຊວສ ( $^{\circ}\text{C}$ ).

- **ຫຼັກການ:** ແຮງດັນໄຟຟ້າທີ່ອອກມາຈະປ່ຽນແປງ  $10\text{mV}$  ຕໍ່  $1^{\circ}\text{C}$  (ເຊັ່ນ: ທີ່  $25^{\circ}\text{C}$ , ມັນຈະໃຫ້ຄ່າອອກ  $250\text{mV}$  ຫຼື  $0.25\text{V}$ ).
- ເຊື່ອມຕໍ່ກັບຂາ Analog Input (A0-A5) ຂອງ Arduino.

## ຮູບພາບອຸປະກອນຕົວຈິງ:



### ຕົວຢ່າງການນຳໃຊ້:

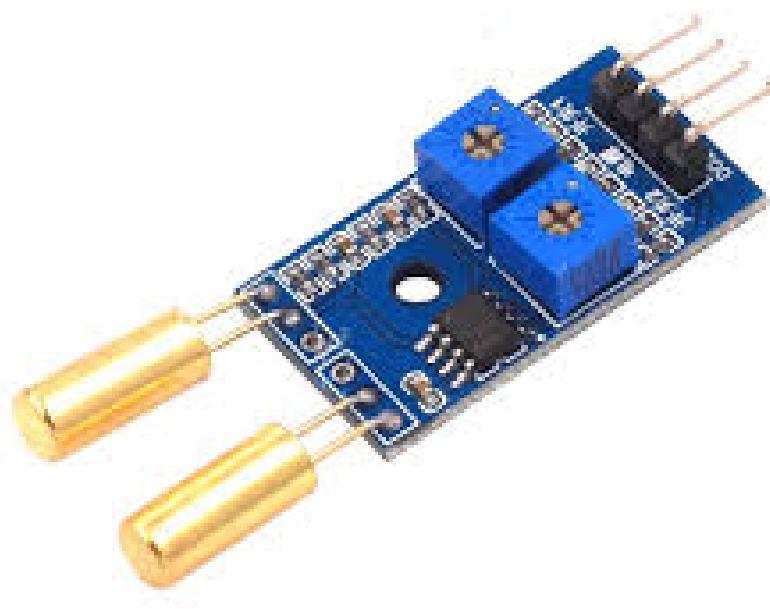
- **Precision Temperature Gauge:** ສ້າງເຄື່ອງວັດອຸນຫະພູມທີ່ສະແດງຜົນໃນຈຳ LCD ດ້ວຍຄວາມລະອຽດສູງ.
- **Overheat Protection:** ໃຊ້ຕິດຕາມອຸນຫະພູມຂອງອຸປະກອນເອເລັກໄຕຣນິກ ແລະ ສ້າງການປິດເມື່ອຮັອນເກີນໄປ.

## 22. Tilt sensor (x2)

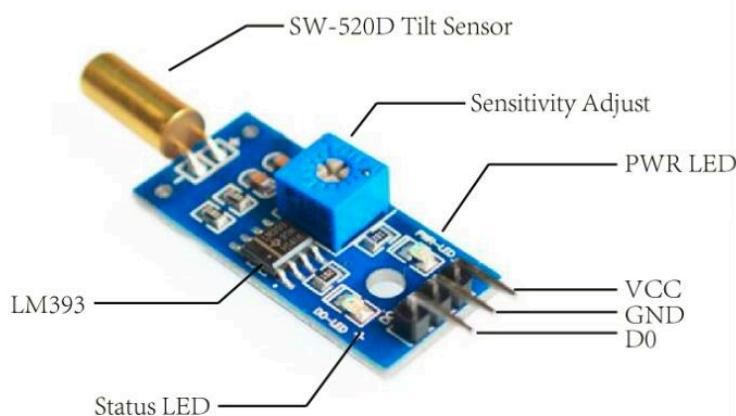
**ການໃຊ້ງານຂອງອຸປະກອນ:** ເຊັນເຊີວັດມຸມອງງາມເມັນສະວິດແບບງ່າຍໆທີ່ໃຊ້ກິນໄກຂອງແຮງໂນ້ມຖ່ວງ (Gravity). ມັນຮັດວຽກໂດຍການປ່ຽນສະຖານະຈາກ ເປີດ (Open Circuit) ເປັນ ປິດ (Closed Circuit) (ຫຼື ກົງກັນຂ້າມ) ເມື່ອ ມັນອງງາມໄປໃນມຸມໃດໜຶ່ງ.

- ມັນຮັດຫັນທີ່ຄືກັບບຸ່ມກົດ, ສັງເກດຕິກົດ (HIGH/LOW) ໄປຫາຂາ Digital Input ຂອງ Arduino.
- **ຂໍ້ສັງເກດ:** ມັນແຕກຕ່າງຈາກ Accelerometer (ເຊັນເຊີວັດຄວາມເລັ່ງ), ເຊັງວັດມຸມອງງາມທີ່ຊັດເຈນ. Tilt Sensor ພຽງແຕ່ບອກວ່າ "ອງງ" ຫຼື "ບໍ່ອງງ".

## ຮູບພາບອຸປະກອນຕົວຈິງ:



## ຮູບອຸປະກອນແບບ Schematic Diagram:



## ຕົວຢ່າງການນຳໃຊ້:

- **Motion Detector:** ໃຊ້ເພື່ອກວດພົບວ່າວັດຖຸຖືກຍົກຂຶ້ນ ຫຼື ວາງລົງ.
- **Security Alarm:** ຕິດຕັ້ງໃສ່ປອງຢູ່ມີ ຫຼື ປະຕຸ ເພື່ອສົ່ງສັນຍາມເຕືອນມື່ອມີການເຄື່ອນໄຫວ/ເປີດ.

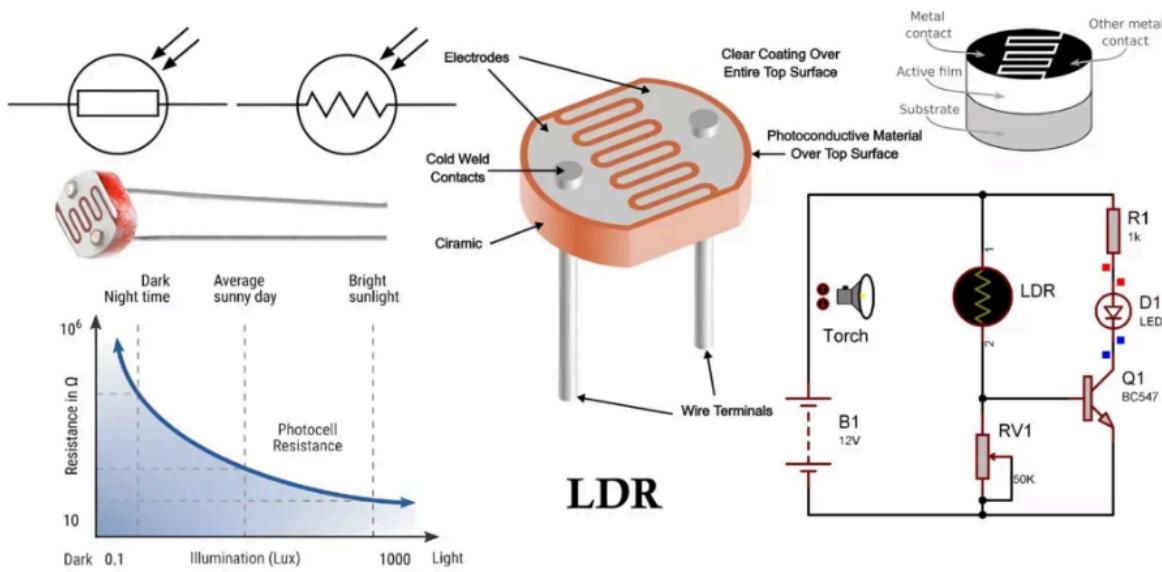
## 23. Photoresistor (LDRs x3)

### ການໃຊ້ງານຂອງອຸປະກອນ:

LDR ແມ່ນຕົວຕໍານທານທີ່ຄ່າຄວາມຕໍານທານຈະ ປ່ຽນແປງຕາມຄວາມເຂັ້ມຂອງແສງ.

- **ຄວາມມືດ:** ຄ່າຄວາມຕໍານທານສູງ (ກະແສໄຟຟ້າໃຫ້ຜ່ານຍາກ).
- **ຄວາມສະຫວັດ:** ຄ່າຄວາມຕໍານທານຕໍ່າ (ກະແສໄຟຟ້າໃຫ້ຜ່ານຢ່າຍ).
- **ການເຊື່ອມຕໍ່:** ມັນຕ້ອງຖືກໃຊ້ເປັນສ່ວນໜຶ່ງຂອງ **Voltage Divider** (ຕໍ່ອະນຸກົມກັບຕົວຕໍານທານຄົງທີ່, ແຊັ້ນ:  $10k\Omega$ ) ເພື່ອຜະລິດສັນຍາມແຮງດັນອະນາລັອກທີ່ປ່ຽນແປງໄດ້ໃຫ້ Arduino ອ່ານຄ່າຜ່ານຂາ Analog Input.

## ຮູບພາບອຸປະກອນຕົວຈິງ:



## ຕົວຢ່າງການນຳໃຊ້:

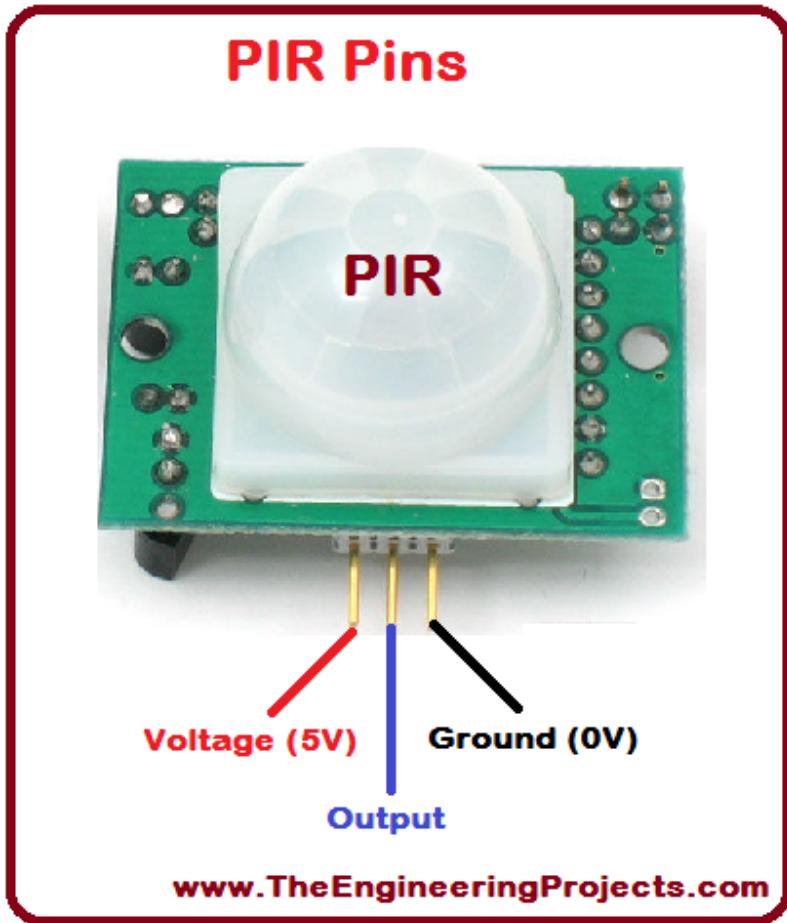
- **Automatic Night Light:** ຖັນຄ່າແສງຕໍ່ກວ່າທີ່ກໍານົດ, ໃຫ້ເປີດໄຟ LED.
- **Light Intensity Meter:** ວັດແທກຄວາມເຂັ້ມຂອງແສງ ແລະ ສະແດງຜົນ.

## 24. PIR Motion sensor (HC-SR501)

**ການໃຊ້ງານຂອງອຸປະກອນ:** PIR (Passive Infrared) sensor ໃຊັງກວດຈັບການເຄື່ອນໄຫວໂດຍອີງໃສ່ການປັບປຸງຂອງຄວາມຮັກນ (Infrared Radiation) ຈາກຮ່າງກາຍຂອງຄົນ ຫຼື ສັດ.

- **ໜັກການ:** ເມື່ອມີການເຄື່ອນໄຫວໃນຂອບເຂດ, ສັນຍານ Output ຈະປັບປຸງຈາກ **LOW** ເປັນ **HIGH** ແຊ່ງເປັນ ສັນຍານ ດີຈິຕອລ ທີ່ສາມາດອ່ານໄດ້ໂດຍກົງຈາກ Arduino.
- ມັນເປັນອຸປະກອນ Input ທີ່ໃຫ້ຜົນລັບພຽງສອງສະຖານະຄື: ກວດພິບ (**HIGH**) ຫຼື ບໍ່ກວດພິບ (**LOW**).

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

- **Security Alarm:** ສັ່ງການ Buzzer ເຕືອນເມື່ອກວດພິບການເຄື່ອນໄຫວ.
- **Automatic Lighting:** ເປີດໄຟ LED ເມື່ອຄືນເຂົ້າໄປໃນຫ້ອງ ແລະ ປິດເມື່ອອອກໄປ.

## 25. Ultrasonic module (HC-SR04)

ການໃຊ້ງານຂອງອຸປະກອນ:

ໃຊ້ໃນການວັດແທກ ໄລຍະທາງ ໄປຫາວັດຖຸໂດຍໃຊ້ຄືນສຽງ Ultrasonic.

- **ຫຼັກການ:** Arduino ສັ່ງສັນຍາ HIGH ໄປຫາຂາ Trig ເປັນເວລາ  $10\mu s$  ເພື່ອໃຫ້ເຊັນເຊີປ່ອຍຄືນສຽງ. ເມື່ອຄືນສຽງກະທົບວັດຖຸ ແລະ ສະຫຼອນກັບ, ຂາ Echo ຈະເປັນ HIGH. Arduino ວັດແທກໄລຍະເວລາທີ່ Echo ເປັນ HIGH ( $t$ ).
- **ການຄຳນວນໄລຍະທາງ:** ໄລຍະທາງ (Distance) =  $\frac{\text{Speed of sound} \times t}{2}$ .

ຮູບພາບອຸປະກອນຕົວຈິງ:

## ULTRASONIC SENSOR



ຕົວຢ່າງການນຳໃຊ້:

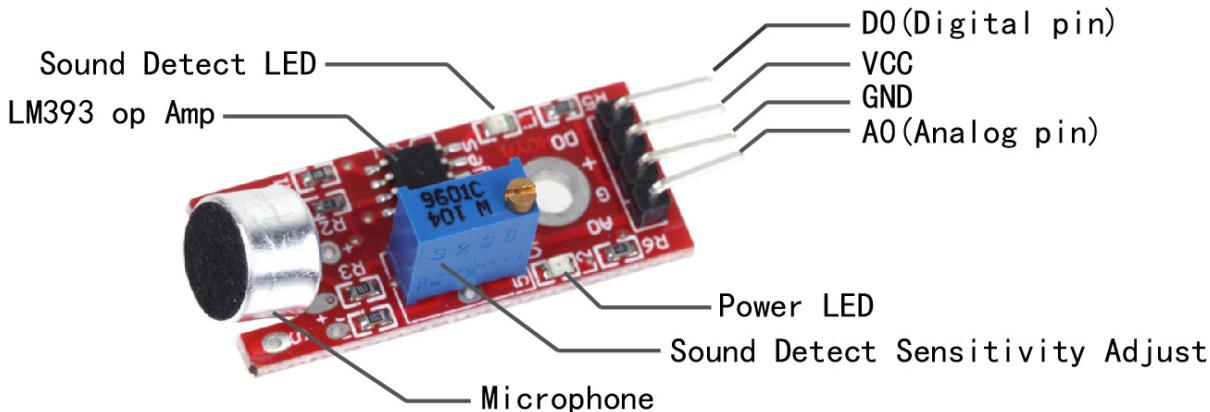
- **Obstacle Avoidance Robot:** ຫຼຸ່ມຍິນໃຊ້ກວດຈັບສິ່ງກີດຂວາງ.
- **Parking Assist Sensor:** ລະບົບຊ່ວຍຈອດລືດແບບບ່າຍໆ.
- **Water Level Measurement:** ວັດແທກລະດັບນ້ຳໃນຖຸໄດ່ບໍ່ຕ້ອງສໍາຜັດກັບນ້ຳ.

### 26. Sound sensor module

ການໃຊ້ງານຂອງອຸປະກອນ: ໃຊ້ສໍາລັບກວດຈັບການມີຢູ່ ຫຼື ວັດແທກລະດັບຄວາມດັງຂອງສຽງອ້ອມຂ້າງ.

- **AOUT:** ໃຊ້ວັດແທກຄວາມດັງສຽງແບບຕໍ່ເນື້ອງ (ເຊັ່ນ: ເພື່ອເຮັດໃຫ້ໄຟ LED ສະຫວັນຂຶ້ນຕາມຄວາມດັງຂອງເພິງ). ເຊື່ອມຕໍ່ກັບຂາ Analog.
- **DOUT:** ໃຊ້ສໍາລັບການກວດຈັບສຽງທີ່ມີຄວາມດັງສູງ (ເຊັ່ນ: ສຽງຕືບມື). ເຊື່ອມຕໍ່ກັບຂາ Digital.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

- **Clap Switch:** ເປີດ/ປິດໄຟດ້ວຍການຕືບມື.

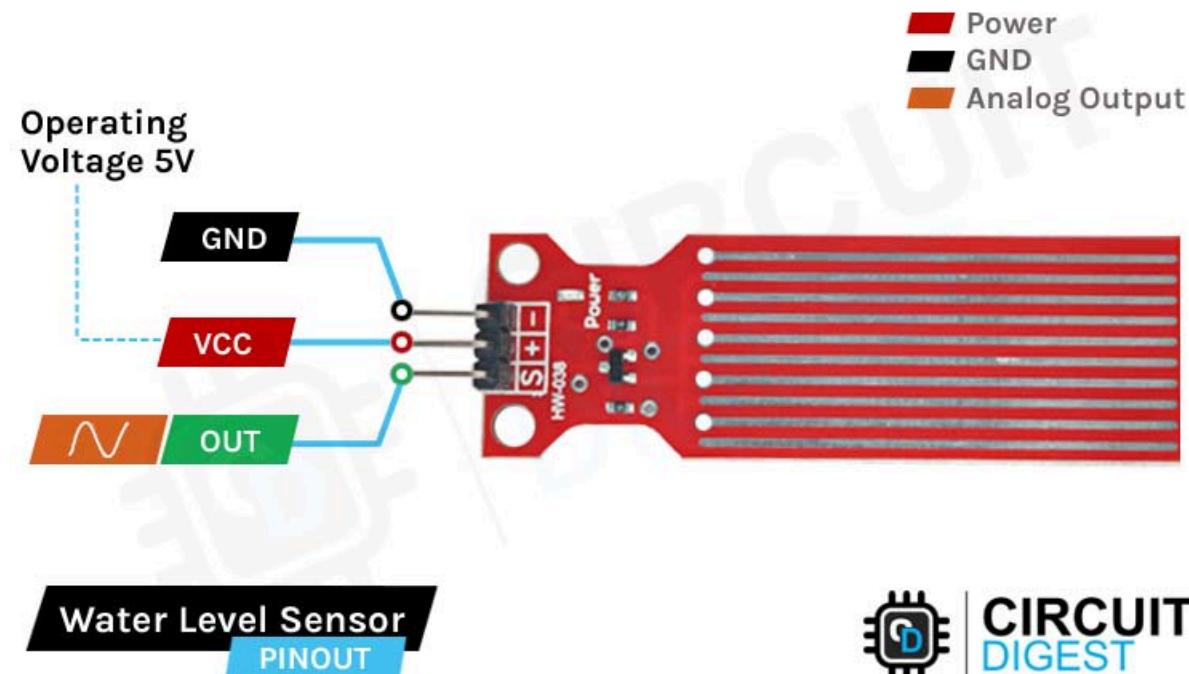
- **Noise Monitoring:** ກວດສອບວ່າສຽງໃນຫ້ອງເກີນຂີດຈຳກັດທີ່ກຳນົດໄວ້ຫຼືບໍ່.

## 27. Water sensor module

**ການໃຊ້ງານຂອງອຸປະກອນ:** ໃຊ້ສໍາລັບກວດຈັບລະດັບນ້ຳ (ເມື່ອຈຸ່ມລົງໃນນ້ຳ) ຫຼື ກວດຈັບນ້ຳຜົນ (ເມື່ອຕິດຕັ້ງໄວ້ຂ້າງນອກ).

- **ຫຼັກການ:** ນ້ຳເຮັດໜ້າທີ່ເປັນຕົວນໍາໄຟຟ້າ (Conductivity), ເຊື່ອມຕໍ່ຮ່ອງໂລຫະຕ່າງໆໃນແຜ່ນເຊັນເຊີ້ ເຮັດໃຫ້ຄວາມຕ້ານຫານຫຼູດລົງ ແລະ ປຽນຄ່າແຮງດັນອອກ.
- **ການເຊື່ອມຕໍ່:** ໃຊ້ AOUT ສໍາລັບການວັດແທກລະດັບ/ປະລິມານ, ແລະ DOUT ສໍາລັບການກວດຈັບວ່າມີນ້ຳແມ່ນອນບໍ່.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

- **Rain Alarm:** ສ້າງການ Buzzer ເຕືອນເມື່ອກວດພົບນ້ຳຜົນ.
- **Automatic Watering System:** ກວດສອບລະດັບນ້ຳໃນຖຸງ.

## 28. Flame sensor module

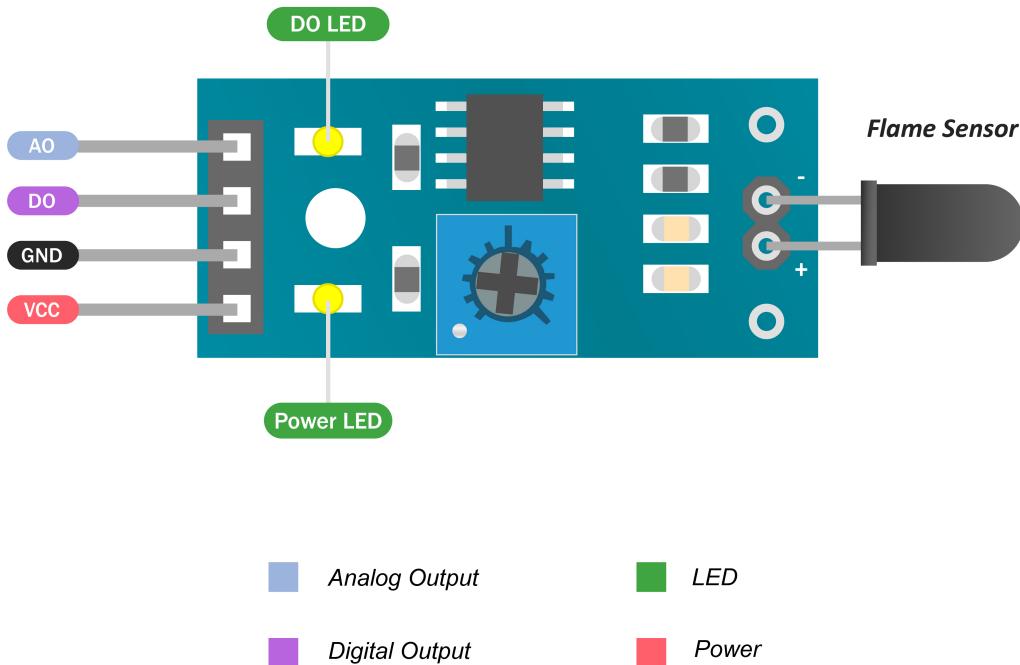
**ການໃຊ້ງານຂອງອຸປະກອນ:**

ໃຊ້ສໍາລັບກວດຈັບແປວໄຟ ຫຼື ແຫ່ງຄວາມຮັອນອື່ນໆທີ່ປ່ອຍລັງສີ Infrared (ຄວາມຍາວຄື້ນປະມານ 760nm ຫາ 1100nm).

- **ຫຼັກການ:** ແປວໄຟປ່ອຍລັງສີ IR ຢ່າງເຂັ້ມຊັ້ນ, ເຊິ່ງຖືກກວດຈັບໂດຍ Photodiode.
- **ການເຊື່ອມຕໍ່:** ໃຊ້ DOUT ເພື່ອເປີດ/ປິດສັນຍາເຕືອນໄຟແບບງ່າຍໆ. ໃຊ້ AOUT ເພື່ອວັດແທກຄວາມເຂັ້ມຂອງແປວໄຟ (ເຊັ່ນ: ເພື່ອກຳນົດໄລຍະຫ່າງ).

ຮູບພາບອຸປະກອນຕົວຈິງ:

## ADIY Flame Sensor Module



ຕົວຢ່າງການນຳໃຊ້:

- Fire Alarm System:** ສັ່ງການ Buzzer ເຕືອນໄພໃໝ່.
- Flame Following Robot:** ຫຼຸ່ມຍິນທີເຄືອນທີ່ໄປຫາແຫຼ່ງກຳເນີດໄຟ.

## 29. . RFID Module (RC522)

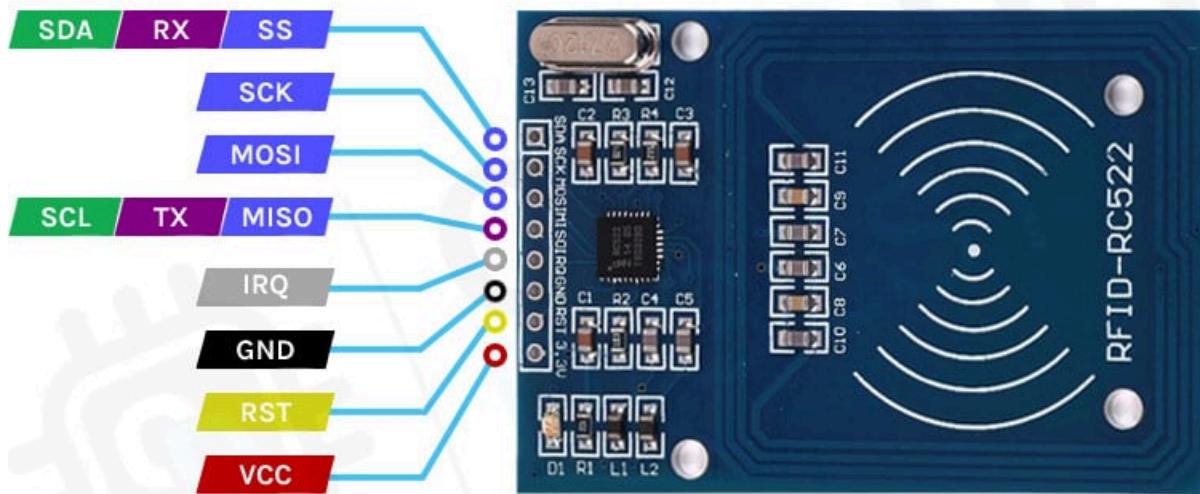
ການໃຊ້ງານຂອງອຸປະກອນ:

ໃຊ້ໃນການອ່ານ ແລະ ຂຽນຂໍ້ມູນໃສບັດ ຫຼື Tag RFID (ຄວາມຖື 13.56MHz).

- ຫຼັກການ:** ໂມດຸນສ້າງສະໜາມແມ່ເຫຼັກໄຟຟ້າ (Electromagnetic Field) ແຊ່ງຈະເປັນຕົວກະຕຸນໃຫ້ RFID Tag (ບັດ/ກະແຈ) ເຮັດວຽກ. ມັນຈະອ່ານຂໍ້ມູນລະຫັດສະເພາະ (UID) ຈາກ Tag ມັນ.
- ຂໍຄວນລະວັງ:** RC522 ໃຊ້ໄຟ 3.3V. ຖ້າເຊື່ອມຕໍ່ກັບ 5V ໂດຍກົງອາດເຮັດໃຫ້ໂມດຸນເສຍຫາຍໄດ້.

## ຮູບພາບອຸປະກອນຕົວຈິງ:

- I2C Communication Pins
- SPI Communication Pins
- UART Communication Pins
- Interrupt Pin
- GND
- VCC
- Reset pin



RC522 RFID Module  
PINOUT



## ຕົວຢ່າງການນຳໃຊ້:

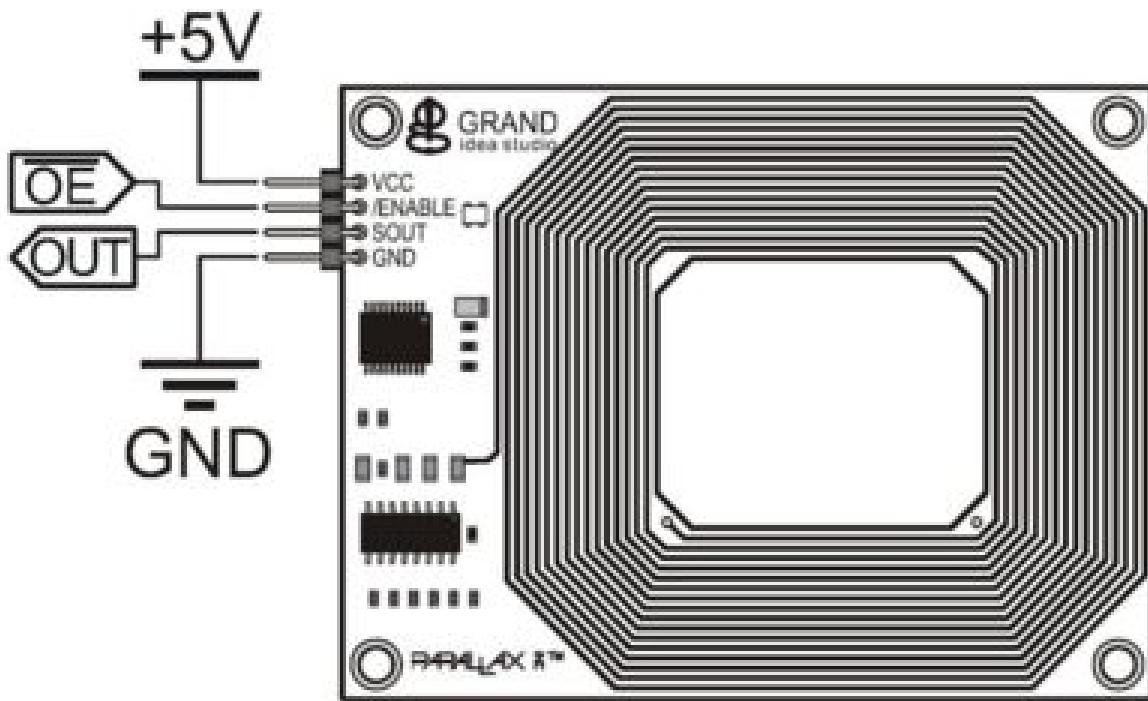
- Access Control:** ລະບົບເປີດ/ປິດປະຕຸດ້ວຍບັດ.
- Asset Tracking:** ລະບົບຕິດຕາມຂັບສິນ.

## 30. RFID Tag (Card/Key Fob)

**ການໃຊ້ງານຂອງອຸປະກອນ:** RFID Tag ແມ່ນອົງປະກອບທີ່ຖືກຕິດຕັ້ງໃສ່ວັດຖຸ ຫຼື ຖືກໃຊ້ໂດຍບຸກຄົນ ເພື່ອໃຫ້ RFID Reader ສາມາດລະບຸຕົວຕົນ ຫຼື ຕິດຕາມໄດ້.

- ຫຼັກການ (Passive Tag):** ບໍ່ມີແບັດເຕີຣີໃນຕົວ. ມັນໄດ້ຮັບພະລັງງານຜ່ານການເໝັງນຳ (Inductive Coupling) ຈາກສະໜາມແມ່ນໜີ້ໄຟຟ້າທີ່ປ້ອຍອອກມາຈາກ RFID Reader (ເຊັ່ນ: RC522). ເມື່ອໄດ້ຮັບພະລັງງານພຽງຟ້ວ, ມັນຈະເປີດໃຊ້ງານ Chip ແລະ ສິ່ງຂໍ້ມູນທີ່ເກັບໄວ້ (ເຊັ່ນ: UID) ກັບຄົນໄປຫາ Reader ຜ່ານຄື່ນວິທະຍຸ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



#### ປະເພດຂອງ Tag:

1. **Passive Tag (Tags ທີ່ບໍ່ມີແບດເຕີຣີ):** ຖືກໃຊ້ຫຼາຍທີ່ສຸດໃນໂຄງການ Arduino (ເຊັ່ນ: ບັດ MIFARE S50). ໃຊ້ພະລັງງານຈາກ Reader ເທົ່ານັ້ນ.
2. **Active Tag (Tags ທີ່ມີແບດເຕີຣີ):** ສາມາດສົ່ງສັນຍານໄດ້ໄກກວ່າ ແລະ ເປັນໄລຍະ.

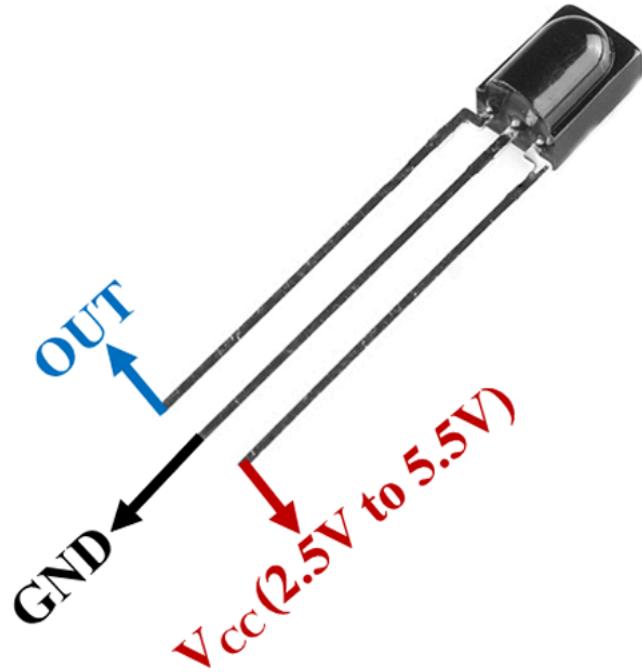
#### 31. Infrared receiver

##### ການໃຊ້ງານຂອງອຸປະກອນ:

ໃຊ້ເພື່ອ ຮັບ ແລະ ຖອດລະຫັດ (Decode) ສັນຍານ infrared ທີ່ສົ່ງມາຈາກ IR Remote Control (ເຊັ່ນ: ອີໂມດໂທລະພາບ ຫຼື ອີໂມດທີ່ມາພ້ອມຊຸດ Kit).

- **ໜັກການ:** IR Receiver (ເຊັ່ນ: VS1838B ຫຼື TSOP Series) ຈະຖືກປັບໃຫ້ຮັບຄື່ນ infrared ຄວາມຖື່ສະເພາະ (ປົກກະຕິ 38kHz). ເມື່ອມັນໄດ້ຮັບຄື່ນທີ່ມີ ລະຫັດ Pulse ຈາກຮີໂມດ, ມັນຈະຖອດລະຫັດນັ້ນເປັນສັນຍານ HIGH/LOW ດີຈິຕອລທີ່ສາມາດສົ່ງໄປໃຫ້ Arduino ອ່ານໄດ້.
- **Arduino Library:** ປົກກະຕິແລ້ວຕ້ອງໃຊ້ **IR remote Library** ເພື່ອໃຫ້ Arduino ສາມາດແປຄ່າລະຫັດ hexadecimal ທີ່ໄດ້ຮັບຈາກແຕ່ລະບຸມກົດເທິງຮີໂມດ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

- **Home Automation:** ຄວບຄຸມໄຟ, ພັດລິມ, ຫຼື ມີເຕີດວຍຮີໂມດໂທລະພາບເກົ່າ.
- **Robotics:** ຄວບຄຸມຫຸ້ນຍິນຈາກໄລຍະໄກ.

## 🎮 Remote & Control

### 32. Infrared Remote Control (IR)

ການໃຊ້ງານຂອງອຸປະກອນ: ໃຊ້ໃນການສິ່ງຄໍາສິ່ງແບບໄຮສາຍໃນໄລຍະໄກ (Line-of-Sight) ໂດຍໃຊ້ລັງສີ Infrared (IR).

- **ຫຼັກການ:** ອີໂມດ (Transmitter) ຈະສິ່ງສັນຍານໂດຍການປ້ອຍແສງ IR ທີ່ມີ ລະຫັດ Pulse ສະເພາະຂອງປຸ່ມທີ່ກິດ. ໂມດຸນຮັບ (Receiver) ເຊັ່ນ VS1838B ຈະແປລະຫັດນັ້ນ ແລະ ສິ່ງຄໍາດິຈິຕອລທີ່ສອດດ່ອງກັນໄປໃຫ້ Arduino.

## ຮູບພາບອຸປະກອນຕົວຈິງ:



### ຕົວຢ່າງການນຳໃຊ້:

- ຄວບຄຸມໄຟ, ພັດລົມ, ຫຼື ມໍຕີຈາກໄລຍະໄກ.

### 33. Joystick Module

#### ການໃຊ້ງານຂອງອຸປະກອນ:

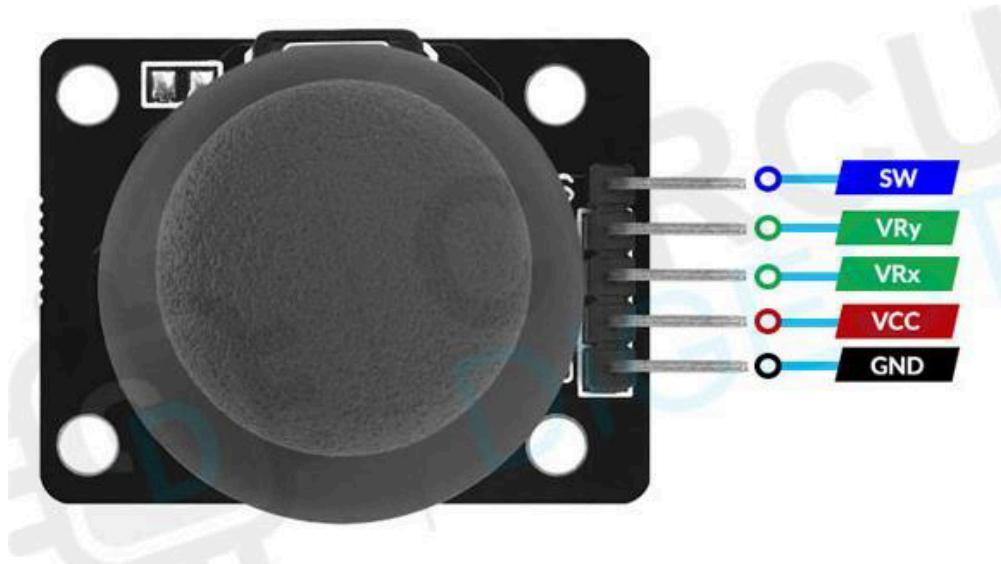
ໃຊ້ໃນການຮັບຄ່າ Input ທີ່ເປັນການຄວບຄຸມແບບ 2 ມິຕີ (2D) ແລະ ບຸ່ມກິດ.

- ຫຼັກການ:** ແຕ່ລະແກນ (X ແລະ Y) ໃຊ້ Potentiometer ເພື່ອປ່ຽນຕຳແໜ່ງເປັນຄ່າແຮງດັນ Analog. Arduino ອ່ານຄ່າເຫຼົ່ານີ້ດ້ວຍ analogRead() ໂດຍຄ່າຈະຢູ່ໃນຊ່ວງ 0 (ສຸດທາງໜຶ່ງ) ຫາ 1023 (ອີກສຸດທາງໜຶ່ງ), ແລະ ປະມານ 512 ຢູ່ຈຸດໃຈກາງ.
- ບຸ່ມກິດ (SW):** ເຮັດວຽກເປັນບຸ່ມກິດດີຈີຕອລທຳມະດາ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



ຕົວຢ່າງການນຳໃຊ້:

- **Robot Control:** ຄວບຄຸມການເຄື່ອນທີ່ ແລະ ຫົດທາງຂອງຫຼຸນຍິນ.
- **Game Controller:** ໃຊ້ເປັນເຄື່ອງຄວບຄຸມໃນເກມແບບຍ່າຍ່າ.
- **Servo Control:** ຄວບຄຸມແຂນກິນ.

#### 34. 4x4 Matrix Keyboard Module

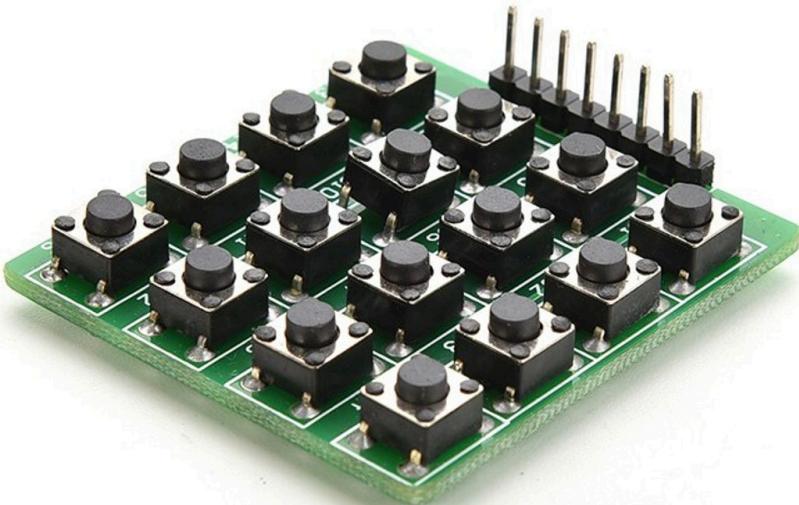
ການໃຊ້ງານຂອງອຸປະກອນ:

ໃຊ້ເປັນສ່ວນຕິດຕໍ່ກັບຜູ້ໃຊ້ (User Input Interface) ເພື່ອໃຫ້ຜູ້ໃຊ້ສາມາດປ້ອນຕົວເລກ, ຕົວອັກສອນ, ຫຼື ຄຳສັ່ງເຂົ້າໄປໃນໂຄງການ Arduino ຂອງທ່ານ.

- **ຫຼັກການ (Scanning/Matrixing):** ແກນທີ່ຈະໃຊ້ 16 Pins ສໍາລັບປຸ່ມ 16 ບຸ່ມ, Keypad ມີຈັດປຸ່ມເປັນຕາຕະລາງ 4 ແຖວ (Rows) ແລະ 4 ຖັນ (Columns), ແຊ່ງຕ້ອງການພຽງ  $4 + 4 = 8$  Pins ເທົ່ານັ້ນ.

- **ການອ່ານປຸ່ມ:** Arduino ຈະ **ບ້ອຍສັນຍານ HIGH** (ຫຼື **LOW**) ໄປຕາມແຖວເທົ່ອລະແຖວ (**Scanning**). ໃນຂະນະດຽວກັນ, ມັນຈະ ກວດສອບຄໍາທີ່ Pin ຂອງທັນ ເພື່ອເບິ່ງວ່າສັນຍານນັ້ນຖືກສຶ່ງຜ່ານປຸ່ມໃດປຸ່ມໜຶ່ງກັບຄືນມາຫຼືບໍ່.
- **ຕົວຢ່າງ:** ຖ້າ Arduino ສຶ່ງ **HIGH** ໄປທີ່ Row 2, ແລະ ດີເຮັບ **HIGH** ຢູ່ Column 3, ມັນຈະຮູ້ວ່າປຸ່ມທີ່ຕັດກັນຢູ່ (Row 2, Column 3) ທີກກິດ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



## 4x4 membrane keypad



## ពិវឌ្ឍន៍ការងារម៉ាទ្រី:

- **Security System:** ລະບົບປ້ອນລະຫັດຜ່ານ (Password/PIN) ເພື່ອເປີດປະຕຸ.
  - **Menu Navigation:** ການເລືອກຕົວເລືອກໃນຈຳສະແດງຜົນ.
  - **Data Input:** ປ້ອນຄໍາຕົວເລກສໍາລັບການຕັ້ງຄ່າຕ່າງໆ.

## 35. Relay Module

**ການໃຊ້ງານຂອງອປກອມ:** ໃຊ້ເພື່ອ ຄວບຄຸມໄຟຟ້າກະແສສະຫຼັບ (AC) ຫຼື ໄຟຟ້າກະແສກົງທີ່ມີແຮງດັນ/ກະແສສູງ ໂດຍໃຊ້ສັນຍານແຮງດັນຕໍ່ຈາກ Arduino.

- **ຫຼັກການ:** Relay ເຮດວຽກຄືກັບ ສະວິດໄຟຟ້າ (Electromagnetic Switch). Arduino ສິ່ງສັນຍານໄປກະຕຸນ Coil ຂອງ Relay, ເຊິ່ງຈະສ້າງສະໜາມແມ່ເຫຼັກເພື່ອດຶງໜີບສະວິດ, ເຮດໃຫ້ວົງຈອນໄຟຟ້າແຮງສູງ (ເຊັ່ນ:ໄຟເຮັດວຽກ) ເປົ້າ ຫຼື ປິດໄດ້ຢ່າງປອດໄພ.

## ຮູບພາບອຸປະກອນຕົວຈິງ:



## Single Channel Relay



# Motors & Drivers

## 36. Servo Motor

ການໃຊ້ງານຂອງອປະກອນ:

ໃຊ້ໃນການໜູນແກນໄປຢັງ ມູມທີ່ກໍານົດໄວ້ ຢ່າງຊັດເຈນ (ເຊັ່ນ:  $0^\circ$  ຫາ  $180^\circ$ ) ແລະ ຮັກສາມຸມນັ້ນໄວ້.

- **ຫຼັກການ:** ຄວບຄຸມດ້ວຍສັນຍານ **PWM (Pulse Width Modulation)** ທີ່ມີຄວາມຖື່ກິງທີ່ ( $50\text{Hz}$ ). ມູມທີ່ມີເຕີ ພູນແມ່ນຖືກກໍານົດໄດ້ ຄວາມກວ້າງຂອງ **Pulse (Pulse Width)**. ມັກໃຊ້ Servo Library (`<Servo.h>`) ຂອງ Arduino.
  - Pulse Width 1.0ms ຫາຍເຖິງ ມູມ  $0^\circ$  (ໂດຍປະມານ).
  - Pulse Width 2.0ms ຫາຍເຖິງ ມູມ  $180^\circ$  (ໂດຍປະມານ).

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

- **Robotic Arm:** ຄວບຄຸມຂຶ້ຕໍ່ຂອງແຂນກົນ.
- **Camera Gimbal:** ປັບມຸມກ້ອງ.
- **Actuator:** ເປີດ/ປິດປະຕຸ ຫຼື ປອງຢ້ຽມອັດຕະໂນມັດ.

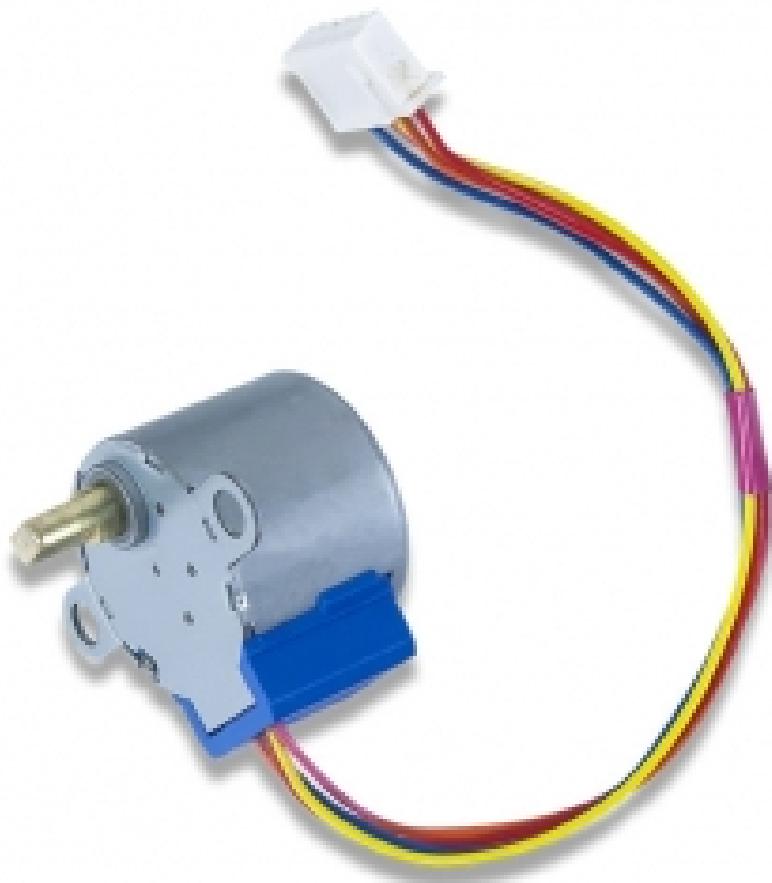
## 37. Stepper Motor

ການໃຊ້ງານຂອງອຸປະກອນ:

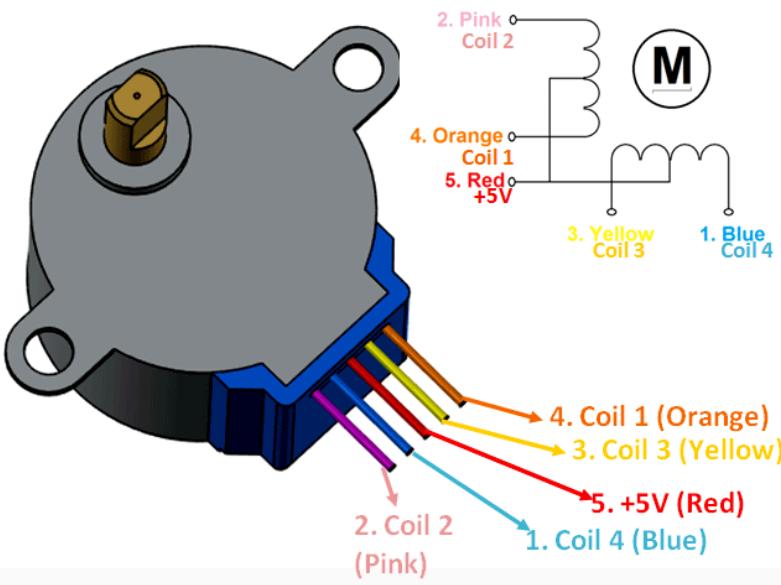
ຕ້ອງໃຊ້ຄູ່ກັບ Driver Board (ເຊັ່ນ: ULN2003) ເພື່ອສະໜອງກະແສໄຟຟ້າທີ່ສູງພຽງພໍໃຫ້ກັບ Coil ຂອງມີເຕີ.

- **ຫຼັກການ:** Arduino ຈະສົ່ງສັນຍານ Digital ໄປຫາ Driver Board ຕາມ **ລຳດັບ Pulse (Sequencing)** ທີ່ ກໍານົດໄວ້. ລຳດັບນີ້ຈະກະຕຸ້ນ Coil ຂອງມີເຕີເຫື່ອລະ Coil, ເຊິ່ງເຮັດໃຫ້ແກນມີເຕີ "ກ້າວ" ໄປໃນມູມທີ່ກໍານົດໄວ້ ຢ່າງຊັດເຈນ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຮູບອຸປະກອນແບບ Schematic Diagram:



ຕົວຢ່າງການນຳໃຊ້:

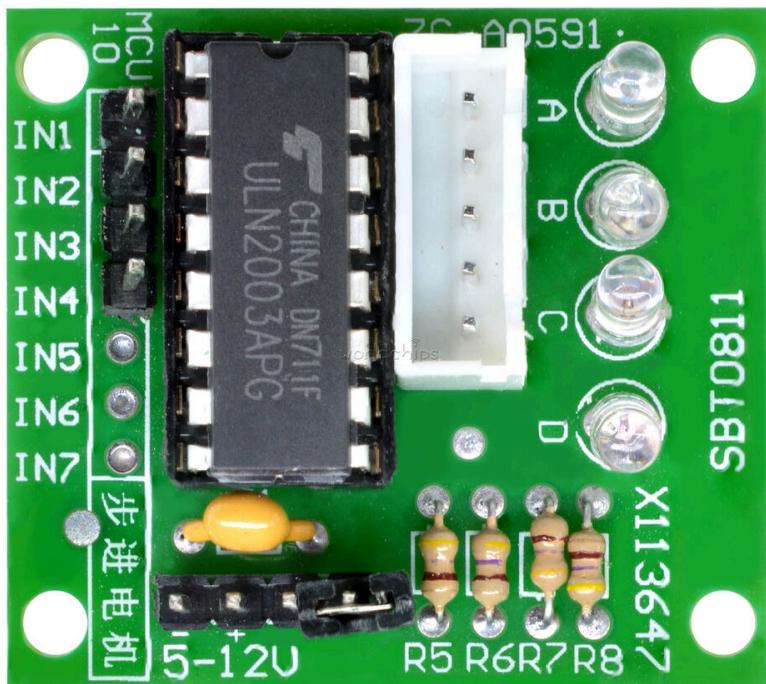
- **3D Printers & CNC:** ການເຄື່ອນຫຼືຂອງແກນ X, Y, Z ທີ່ຕ້ອງການຄວາມຊັດເຈນສູງ.
- **Camera Platforms:** ຄວບຄຸມການຫັນມູມຂອງກ້ອງຖ່າຍຮູບ.
- **Automation:** ຄວບຄຸມຕຳແໜ່ງຂອງວາວ ຫຼື ກິນໄກໃນໂຮງງານ.

## ការងារខ្លួនរបស់ខ្លួន:

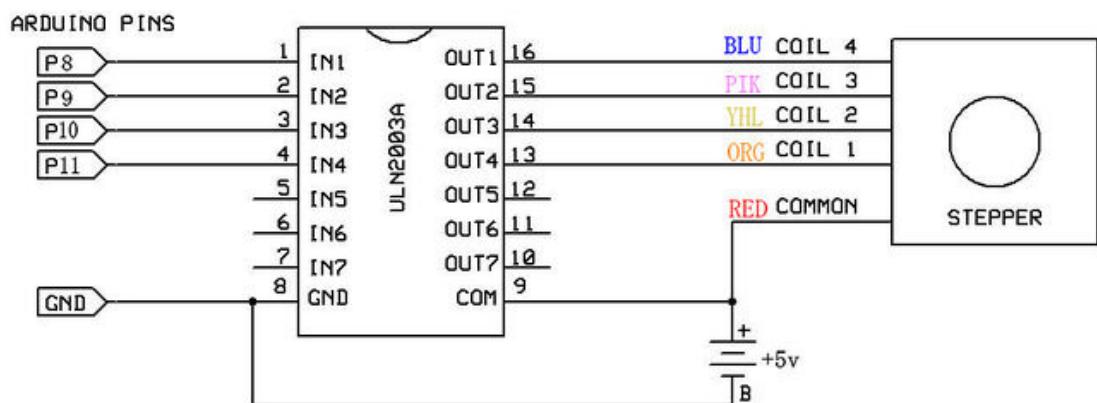
ធ្វើដោយ ខ្សែតែ លម្អិត Arduino និង Stepper Motor. ទាំងអស់ពីរមានចំណាំខាងក្រោម:

- ខ្សែតែរបស់ខ្លួន: ដើម្បីបង្កើតសមាថុមុន្តៃ, យោង Arduino ប៉ុន្មាននៅក្នុងការងារខ្លួន។
- រួមចាប់ផ្តើមពី: បិកប៉ែង Digital Pins ខ្លួន Arduino ទាំងអស់ពីរនៅក្នុងការងារខ្លួន។

## រូបរាងរបស់ខ្លួន:



## រូបរាងរបស់ខ្លួន Schematic Diagram:



## ពិវិះំរាយរបស់ខ្លួន:

គុណភាពរបស់ខ្លួន គឺជាផ្លូវការរបស់ខ្លួន។

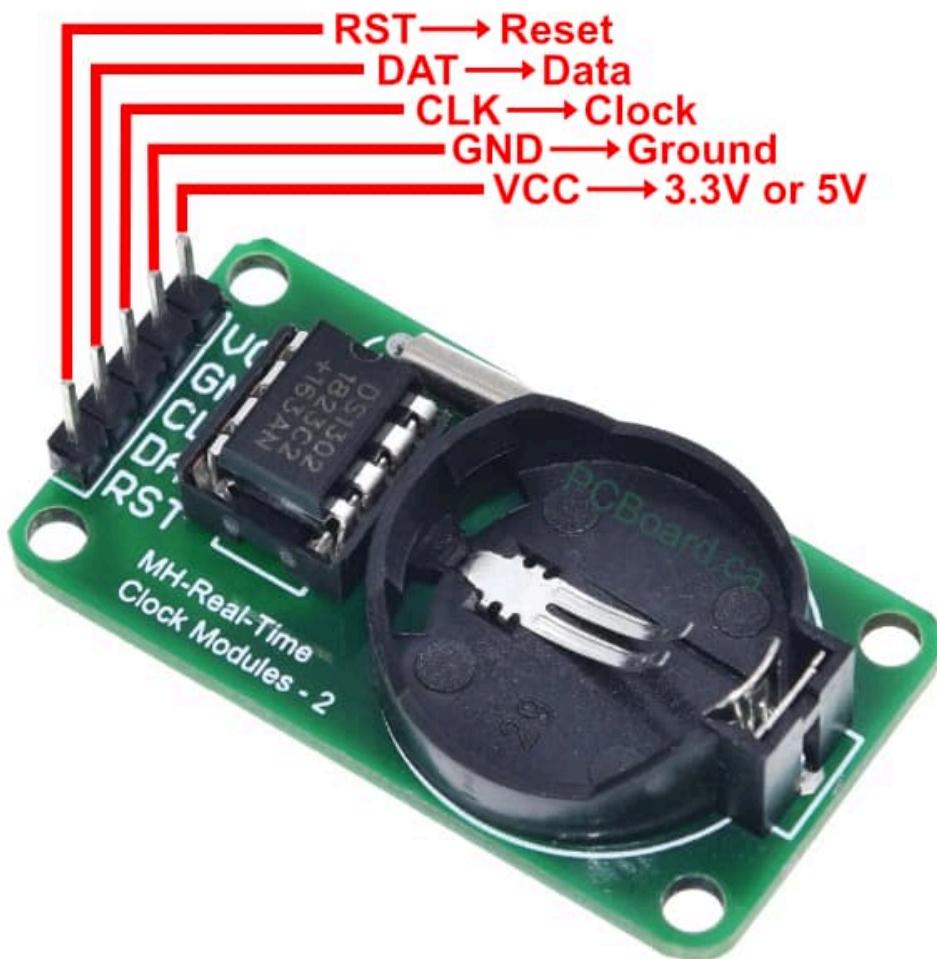
## ICs & Modules

### 39. Real-time Clock Module DS1302

ການໃຊ້ງານຂອງອຸປະກອນ: ໃຫ້ຂໍ້ມູນ ວັນທີ ແລະ ເວລາ ຫີ້ຖືກຕ້ອງແກ່ໂຄງການ Arduino ຂອງທ່ານ.

- **ຫຼັກການ:** ມັນມີວິຈຈອນ Oscillator ແລະ Counter ພາຍໃນທີ່ຮັກສາເວລາໄວ້ເຕິງແມ່ນວ່າ Arduino ຈະຖືກປິດ. ມັນເກັບກຳຂໍ້ມູນ ເຊັ່ນ: ວັນທີ, ນາທີ, ຊົ່ວໂມງ, ມື້, ວັນທີ, ເດືອນ, ແລະ ປີ. Arduino ສື່ສານກັບມັນຜ່ານການເຊື່ອມຕໍ່ແບບ 3 ສາຍ ເພື່ອອ່ານ ແລະ ຕັ້ງຄ່າເວລາ.

ຮູບພາບອຸປະກອນຕົວຈິງ:



ຕົວຢ່າງການນຳໃຊ້:

- **Data Logging:** ບັນທຶກເວລາຂອງເຫດການຕ່າງໆ.
- **Automatic Timer:** ຕັ້ງໂມງຈັບເວລາໃຫ້ລະບົບເປີດ/ປິດອັດຕະໂນມັດຕາມເວລາທີ່ກໍານົດ.

### 40. 74HC595 Shift Register Chip

ການໃຊ້ງານຂອງອຸປະກອນ: 74HC595 ເປັນ Shift Register ທີ່ໃຊ້ເພື່ອ **ຂະຫຍາຍຈຳນວນ Output Digital Pins** ຂອງ Arduino ຂອງທ່ານ.

- **បង្កការ:** មันសາມາດប្រើប្រាស់មុន Serial ទីសំឡែងរវាង Arduino (ដើម្បី 3 Pins) ទៅប្រើប្រាស់មុន Parallel 8 Bits. នឹងមាយគុណភាពថា ពីរសាមាតភ័យគឺជាក្នុង Output ត្រួតពិនិត្យ 8 ពិន្ទុ (ខ្លះ: ឯក LED 8 ឈូក) ដោយដើម្បី 3 Digital Pins នៃ Arduino. ផ្ទាល់ខ្លួន 2 Chips 74HC595 ចំណេះ (Cascading), ពីន្ទុនវាបានបាន Output 16 ពិន្ទុ, ដើម្បី 3 Pins គិត។

## 🔌 PINOUT:

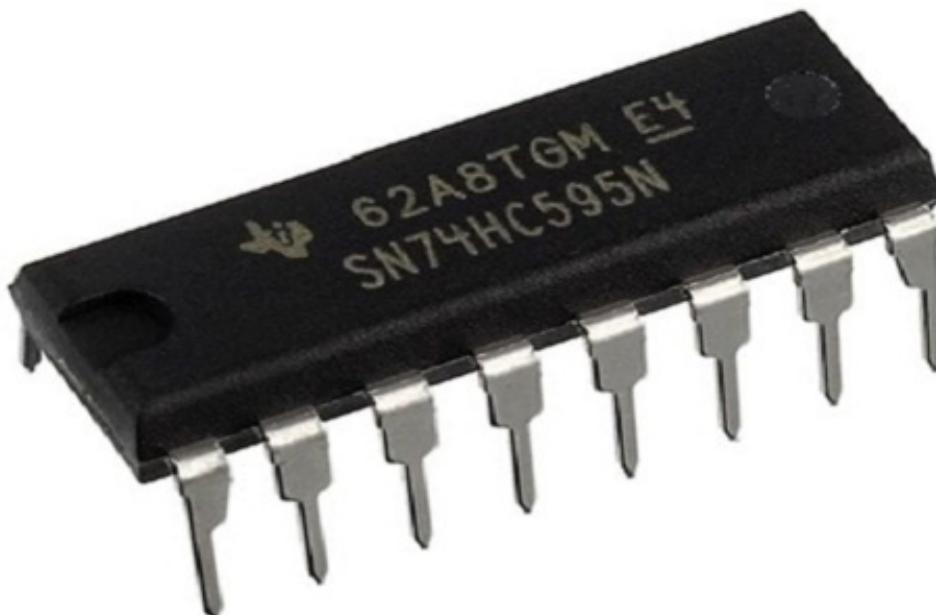
**DS (Data Serial)** ຂໍເຊື້ອມເຂົ້າ Serial ຈາກ Arduino.

**SHCP (Shift Register Clock)** ຂໍ Clock ທີ່ໃຊ້ໃນການຍ້າຍຂໍມູນເຂົ້າໃນ Register.

**STCP (Storage Register Clock)** 係 Clock ที่ใช้ในการส่งค่า Output.

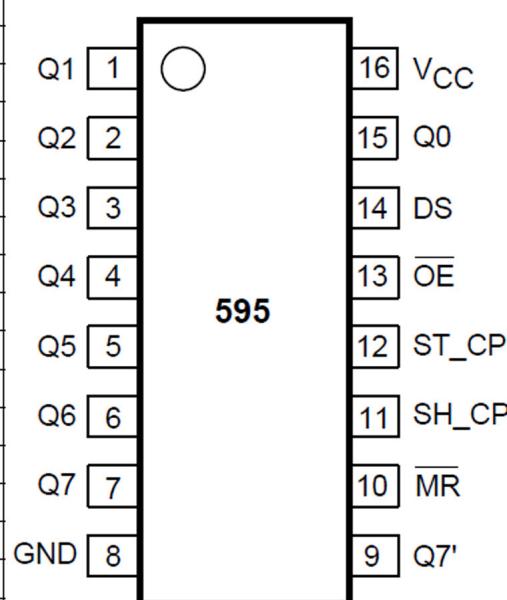
**Q0 - Q7** Output Pins (8 ตัว).

## ຮູບພາບອຸປະກອນຕົວຈິງ:



### ຮບອປະກອນແບບ Schematic Diagram:

PIN	SYMBOL	DESCRIPTION
1	Q1	parallel data output
2	Q2	parallel data output
3	Q3	parallel data output
4	Q4	parallel data output
5	Q5	parallel data output
6	Q6	parallel data output
7	Q7	parallel data output
8	GND	ground (0 V)
9	Q7'	serial data output
10	$\overline{MR}$	master reset (active LOW)
11	SH_CP	shift register clock input
12	ST_CP	storage register clock input
13	$\overline{OE}$	output enable (active LOW)
14	DS	serial data input
15	Q0	parallel data output
16	$V_{cc}$	positive supply voltage



## ຕົວຢ່າງການນຳໃຊ້:

- **LED Matrix/Bar:** ຄວບຄຸມໄຟ LED ຈຳນວນຫຼາຍ (16, 24, 32 ດອກ ຫຼື້ຫຼາຍກວ່ານັ້ນ) ດ້ວຍຈຳນວນ Pin ທີ່ຈໍາກັດ.
- **7-Segment Display:** ຄວບຄຸມ 7-Segment Display ຫຼາຍຕົວພ້ອມກັນ.