



ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນລາວ
ສັນຕິພາບ ເອກະລາດ ປະຊາທິປະໄຕ ເອກະພາບ ວັດທະນາຖາວອນ



ມະຫາວິທະຍາໄລແຫ່ງຊາດ

ຄະນະວິສະວະກຳສາດ

ພາກວິຊາວິສະວະກຳຄອມພິວເຕີ

ສອນໂດຍ ປຕ ອຈ ລັດທິດາ ຄົມສອນລະສິນ

ຮຽບຮຽງ: ທ້າວ ພຸດສະຫວັນ ແຈ້ງສະຫວ່າງ

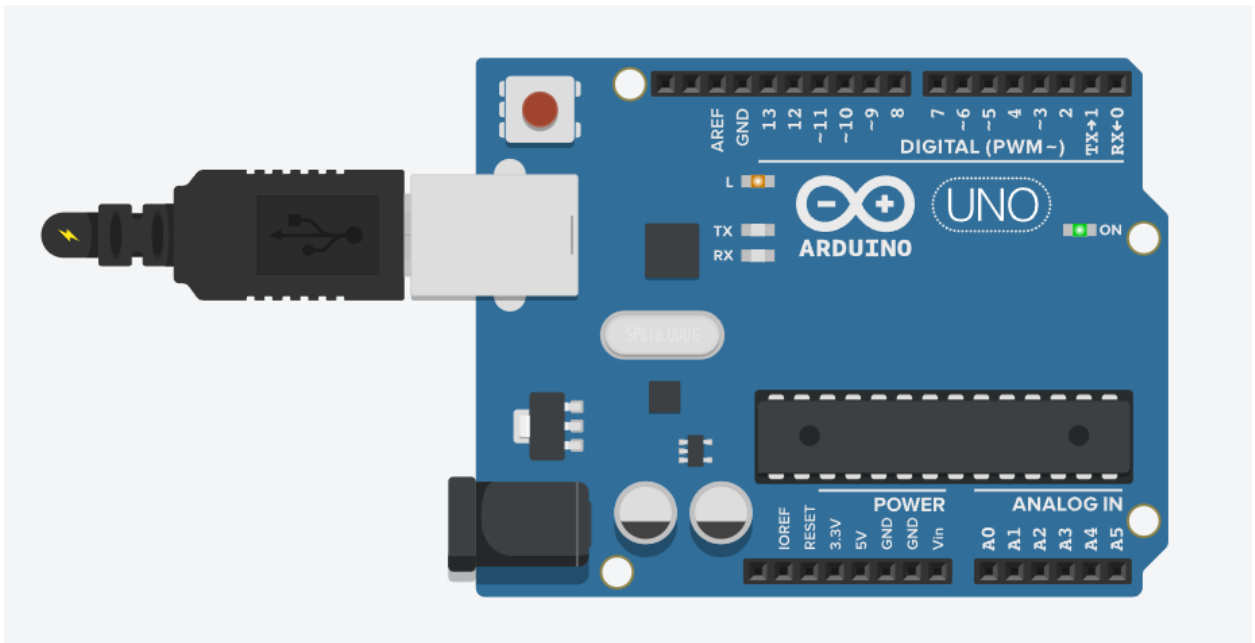
Email: nosaya2021rs@gmail.com

Number: 205804516

ບົດສະຫຼຸບວິຊາໄມໂຄໂປຣເຊັດເຊີແລະໄມໂຄຄອມພິວເຕີ.

ບົດທີ 1 Blink

1.Blink ແມ່ນຫຍັງ:



Blink ຄືໂປຣເຈັກຫຼືຊຸດຄໍາສັ່ງທີ່ໃຊ້ໃນ (tinker cad)
ເປັນ platform ການອອກແບບ 3D ແລະການຈຳລອງວົງ
ຈອນ (Circuit simulation) ໂດຍໃນສ່ວນຂອງ Blink
ໝາຍເຖິງການເຮັດໄຟ led ກະພົບ (blinking led)
ໂດຍໃຊ້ microcontroller ເຊັ່ນ Arduino ໃນ
ການຄວມຄຸມການກະພົບຂອງ led ໃນ tinker cad
ເຮົາສາມາດສ້າງ ວົງຈອນນີ້ໄດ້ຢ່າງງ່າຍດາຍ ໂດຍການໃຊ້
ເຄື່ອງມືຈຳລອງເພື່ອທົດສອບໂປຣເຈັກກ່ອນທີ່ຈະໄປໃຊ້
ງານໂຕຈິງ
ສະຫຼຸບໂປຣເຈັກ blink ໃຊ້ຫຼັກການງ່າຍໆໃນການຂຽນ
code ເຊັ່ນໃຫ້ led ຕິດແລະດັບຕາມເວລາທີ່ກຳໜົດ

ເຊິ່ງເປັນໂປຣເຈັກເລີ່ມຕົ້ນທີ່ດີສໍາຫຼັບຜູ້ທີ່ຮຽນຮູ້ການຂຽນ
ໂປຣແກຣມ microcontroller ເຊັ່ນ Arduino ຜ່ານ
tinker cad.

1.1 Blink ການເຮັດວຽກ:

1.11: ການເຂົ້າສູ່ລະບົບ Tinkercad:

ເຂົ້າສູ່ລະບົບ tinker cad ແລະເຂົ້າສູ່ລະບົບດ້ວຍ

ບັນຊີ Autodesk ຂອງເຈົ້າຂອງເອງ

1.12: ສ້າງໂປຣເຈັກໃໝ່:

ກົດທີ່ “create new design” ເພື່ອສ້າງໂປຣເຈັກ
3D ໃໝ່ຫຼືເລືອກ “circuits” ເພື່ອທຳງານກັບວົງ
ຈອນອີເລັກທຣໍນິກ .

1.13 ເພີ່ມ Arduino ແລະສ່ວນປະກອບ:

- ໃນໂໜດ “circuits” ໃຫ້ລາກ Arduino (ເຊັ່ນ
Arduino uno) ມາວາງໃນພື້ນທີ່ທຳງານ

- ເພີ່ມ led ແລະຕົວຕ້ານທານ (resistor)

ລົງໃນວົງຈອນ ໂດຍຕໍ່ led ເຂົ້າກັບຂາ digital ຂອງ

Arduino (ເຊັ່ນ ຂາ 13) ແລະຕໍ່ຕົວຕ້ານທານ

ເພື່ອຈຳກັດກະແສໄຟຟ້າ.

1.14 ຂຽນ code blink:

- ກົດທີ່ປຸ່ມ code ເພື່ອເປີດໜ້າຕ່າງຂຽນ

Code

Tinker cad ມີຕົວຢ່າງ code ພື້ນຖານສໍາຫຼັບການເຮັດໃຫ້ led ກະພົບ ເຊິ່ງເຮົາສາມາດໃຊ້ໄດ້ທັນທີຈະຂຽນ code ເອງກະໄດ້.

ຕົວຢ່າງ code:

```
void setup() {  
  pinMode(13, OUTPUT); // ຕັ້ງຄ່າຂາ 13 ເປັນ  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // ເປີດ LED  
  delay(1000);           // ຖ້າ 1 ວິນາທີ  
  digitalWrite(13, LOW);  // ປິດ LED  
  delay(1000);           // ຖ້າ 1 ວິນາທີ  
}
```

1.15 ຈຳລອງການເຮັດວຽກ:

- ຫຼັງຈາກຂຽນ code ແລ້ວ ໃຫ້ຄຣິກປຸ່ມ

start simulation ເພື່ອເລີ່ມການຈຳລອງ

ເຮົາຈະເຫັນ led ທີ່ຕໍ່ກັບຂາ 13 ຂອງ Arduino ກະຟິບ

ຕາມທີ່ຕັ້ງຄ່າໄວ້ໃນ code

1.16 ປັບແຕ່ງແລະທົດສອບ:

ເຮົາສາມາດປັບປ່ຽນ code ຫຼືວົງຈອນເພື່ອທົດສອບການ
ທຳງານທີ່ແຕກຕ່າງອອກໄປເຊັ່ນປ່ຽນຄວາມໄວການກະພົບ
ຫຼືໃຊ້ led ຫຼາຍຕົວ

ຕົວຢ່າງ Link in Tinkercad:

link:

<https://www.tinkercad.com/things/0LWIH4W>

EsVC-0-

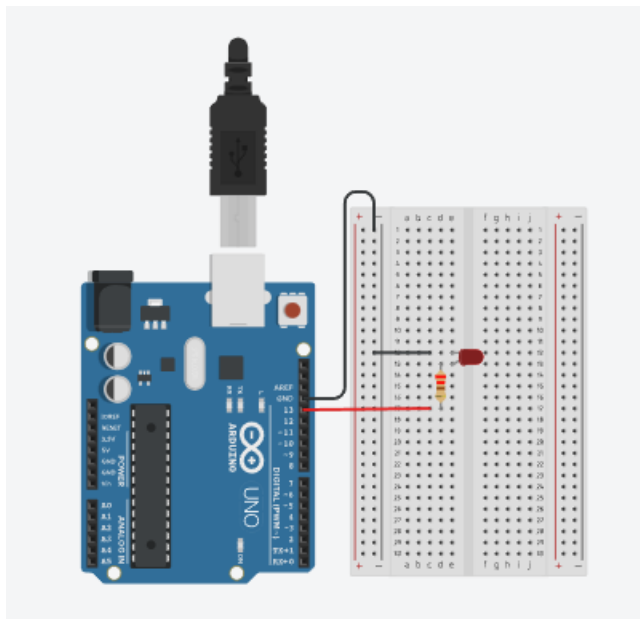
blink?sharecode=S3q8Pjr8kyig00HoM01p_

wVuP3_XJTq3WRuQ_QIWQW4



ບົດທີ2 Experiment Blink

1. Experiment Blink ແມ່ນຫຍັງ:



ຄືການທົດລອງຟື້ນຖານທີ່ເຮັດໃຫ້ led ກະຟິບ
ໂດຍໃຊ້ microcontroller ເຊັ່ນ Arduino,
raspberry pi ຫຼື board ອື່ນໆ ການທົດລອງນີ້
ມັກເປັນໜຶ່ງໃນຂັ້ນຕອນແລກສຳຫຼັບຜູ້ເລີ່ມຕົ້ນຮຽນຮູ້
ກ່ຽວກັບການຂຽນໂປຣແກຣມແລະອີເລັກທຣໍນິກ.

1.1 ການເຮັດວຽກ:

ເປັນໂຄງການຟື້ນຖານທີ່ໃຊ້ສຳຫຼັບການຮຽນຮູ້ການຂຽນ
ໂປຣແກຣມ microcontroller (Arduino) ແລະການ
ທຳງານຂອງວົງຈອນອີເລັກທຣໍນິກ ໂດຍມີຈຸດມຸ່ງໝາຍເພື່ອ
ໃຫ້ led ກະຟິບ (blink)

ຕາມໂປຣແກຣມທີ່ຂຽນ ຕໍ່ໄປນີ້ເປັນຂັ້ນຕອນແລະຫຼັກການ
ທຳງານຂອງ Experiment Blink.

1.11 ອຸປະກອນ:

- Arduino: ໃຊ້ເປັນ microcontroller ສຳຫຼັບຄວບ
ຄຸມການທຳງານຂອງວົງຈອນ
- LED: ໃຊ້ອຸປະສະແດງຜົນ ໂດຍຈະກະຜົບຕາມ
ສັນຍານທີ່ສົ່ງມາຈາກ Arduino

- Resistor: ໃຊ້ເພື່ອຈຳກັດກະແສໄຟຟ້າໃຫ້ເໝາະສົມກັບ led ປ້ອງກັນບໍ່ໃຫ້ led ເສຍຫາຍ
- ສາຍເຊື່ອມຕໍ່: ໃຊ້ສຳຫຼັບເຊື່ອມຕໍ່ລະຫວ່າງ Arduino, led, ແລະ ຕົວຕຳນິທານ

1.12 ການຕໍ່ວົງຈອນ:

1. ຕໍ່ຂາ Anode (ຂາບວກ)ຂອງ led ກັບ digital pin ຂອງ Arduino (ເຊັ່ນ pin 13).

2. ຕໍ່ຂາ Cathode (ຂາລົບ) ຂອງ led ກັບ ຕົວ
ຕ້ານທານ.

3. ຕໍ່ຕົວຕ້ານທານເຂົ້າກັບ Ground (GND)
ຂອງ Arduino.

1.13 ການຂຽນໂປຣແກຣມ:

ໂປຣແກຣມສໍາຫຼັບ Experiment Blink ຈະຂຽນດ້ວຍ
ພາສາ Arduino (C/C++)
ເຊິ່ງມີໂຄງສ້າງຫຼັກ 2 ສ່ວນຄື

Setup () ແລະ loop ():

Code ຕົວຢ່າງ:

```
1 void setup() {
2   pinMode(13, OUTPUT); // ຕັ້ງຄ່າ Pin 13 ເປັນ OUTPUT
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH); // ສົ່ງສັນຍານ HIGH (ເປີດ LED)
7   delay(1000);             // ຖ້າ 1 ວິນາທີ (1000 ມິວລິວິນາທີ )
8   digitalWrite(13, LOW);  // ສົ່ງສັນຍານ LOW (ປິດ LED)
9   delay(1000);            // ຖ້າ 1 ວິນາທີ
10 }
```


1.14 ຄໍາອະທິບາຍ Code:

Pin mode (13, OUTPUT): ຕັ້ງຄ່າ Pin 13

ເປັນໂຫມດ OUTPUT ເພື່ອສົ່ງສັນຍານອອກໄປຄວບຄຸມ
led .

digital Write (13, HIGH): ສົ່ງສັນຍານ HIGH
(ແຮງດັນ 5V) ໄປທີ່ Pin 13 ເພື່ອເປີດ led.

digital Write (13 LOW): ສົ່ງສັນຍານ LOW
(ແຮງດັນ 0V) ໄປທີ່ Pin 13 ເພື່ອເປີດ led.

Delay (1000): ຢຸດການທຳງານເປັນເວລາ

1 ວິນາທີ (1000 ມິວລິວິນາທີ).

1.15 ການທຳງານຂອງວົງຈອນ:

1. ເມື່ອໂປຣແກຣມເລີ່ມທຳງານ Arduino ຈະສົ່ງສັນຍານ HIGH ໄປທີ່ Pin 13 ເຮັດໃຫ້led ຕິດ
2. ຫຼັງຈາກນັ້ນໂປຣແກຣມຈະຢຸດທຳງານເປັນເວລາ 1 ວິນາທີ(ຕາມຄຳສັ່ງ delay (1000))

3. ຈາກນັ້ນ Arduino ຈະສົ່ງສັນຍານ low HIGH
ໄປທີ່ Pin 13 ເຮັດໃຫ້ led ດັບ
4. ໂປຣແກຣມຈະຢຸດທຳງານອີກ 1 ວິນາທີ
5. ກະບວນການນີ້ຈະວິນລູບໄປເລື້ອຍໆ
ເຮັດໃຫ້ led ກະພິບ.

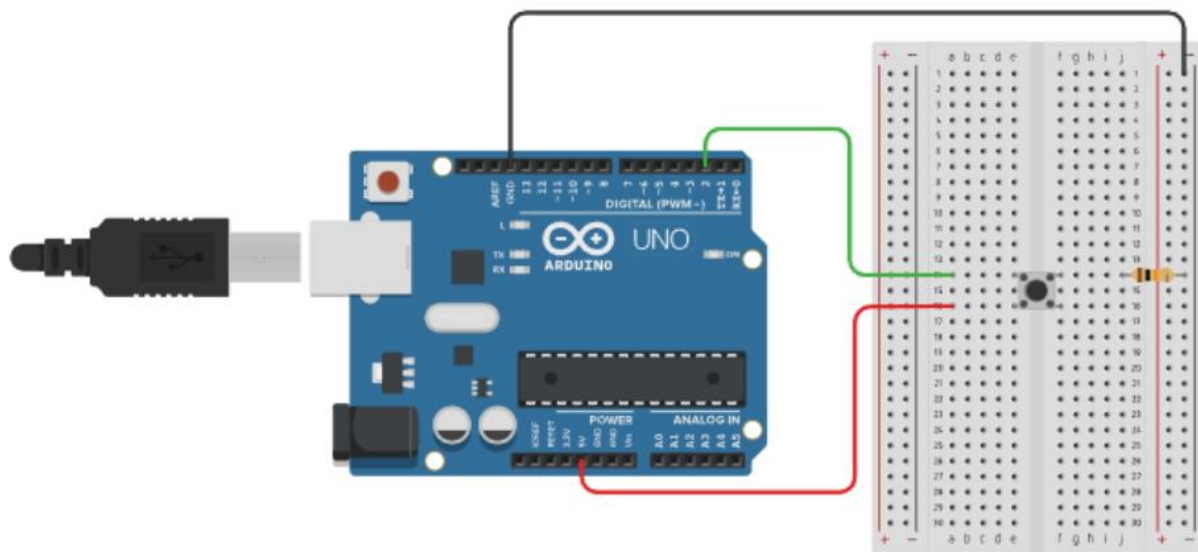
ຕົວຢ່າງ Link in Tinkercad:

https://www.tinkercad.com/things/cvyP9fFIw9a-1-experiment-blink?sharecode=IAkYurLgWOFjNI5SCZ4O_cCq7BtQ1Yri2w_S-UU-GEU



ບົດທີ3 Switch

1. Switch ແມ່ນຫຍັງ:



ເປັນສ່ວນປະກອບອີເລັກທຣ໌ນິກທີ່ໃຊ້ໃນການຄວມຄຸມວົງ
ຈອນໄຟຟ້າ ໂດຍການເປີດຫຼືປິດວົງຈອນເພື່ອໃຫ້ກະແສ
ໄຟຟ້າໄຫຼຫຼືຢຸດໄຫຼ

Switch ເປັນອຸປະກອນພື້ນຖານທີ່ໃຊ້ໃນໂຄງການອີ
ເລັກທຣ໌ນິກຫຼາຍປະເທດແລະສາມາດນຳມາໃຊ້ໃນ
tinkercad ເພື່ອຈຳລອງການທຳງານຂອງວົງຈອນໄດ້

1.1 ປະເພດຂອງ Switch:

1. Push Button Switch

ສະວິດ ທີ່ທຳງານເມື່ອກົດປຸ່ມແລະຈະກັບສູ່ສະຖານະເດີມ
ເມື່ອປ່ອຍ

ມັກໃຊ້ໃນໂຄງການທີ່ຕ້ອງການຄວບຄຸມຊົ່ວຄາວ ເຊັ່ນ
ການເປີດ/ປິດ led ຊົ່ວຄາວ

2. Toggle Switch

- ສະວິດທີ່ສາມາດລ່ອກສະຖານະໄດ້
ເຊັ່ນເປີດຫຼືປິດ
- ເໝາະສຳລັບການຄວບຄຸມທີ່ຕ້ອງການໃຫ້ຄົງສະຖານະໄວ້
ຈົນກວ່າຈະສະລັບອີກຄັ້ງ

3. Slide Switch

ສະວິດທີ່ທຳງານໂດຍການເລື່ອນສະວິດໄປມາເພື່ອເປີດຫຼືປິດ
ວົງຈອນ.

1.3 ການໃຊ້ສະວິດໃນ Tinkercad

ສະວິດ Tinkercad ສາມາດນຳມາໃຊ້ໃນວົງຈອນເພື່ອ
ຄວບຄຸມການທຳງານຂອງອຸປະກອນອື່ນໆ

ເຊັ່ນ led ,ມໍເຕີ,ຫຼືເຊັ່ນເຊີ ຕໍ່ໄປນີ້ເປັນຕົວຢ່າງການໃຊ້ງານ
ສະວິດໃນວົງຈອນ Arduino:

1.31 ສ່ວນປະກອບທີ່ໃຊ້:

Arduino (ເຊັ່ນ Arduino Uno)

ສະວິດ (Push Button)

LED

ຕົວຕ້ານທານ(Resistor)

ສາຍເຊື່ອມຕໍ່

1.32 ການຕໍ່ວົງຈອນ:

ຕໍ່ຂາໜຶ່ງຂອງສະວິດເຂົ້າກັບ Digital Pin ຂອງ

Arduino (ເຊັ່ນ Pin 2)

ຕໍ່ຂາອີກເບື້ອງຂອງສະວິດເຂົ້າກັບ

Ground (GND)

ຕໍ່ LED ກັບ Digital Pin ອື່ນ (ເຊັ່ນ Pin 13) ຜ່ານ
ຕົວຕ້ານທານ

ຕໍ່ Ground ຂອງ LED ເຂົ້າກັບ Ground ຂອງ
Arduino

1.4 ຂຽນໂປຣແກຣມ:

```
1  const int buttonPin = 2; // ຕັ້ງຄ່າ Pin ທີ່ຕໍ່ກັບສະວິດ
2  const int ledPin = 13;   // ຕັ້ງຄ່າ Pin ທີ່ຕໍ່ກັບ LED
3
4  void setup() {
5      pinMode(buttonPin, INPUT); // ຕັ້ງຄ່າ Pin 2 ເປັນ INPUT
6      pinMode(ledPin, OUTPUT);   // ຕັ້ງຄ່າ Pin 13 ເປັນ OUTPUT
7  }
8
9  void loop() {
10     int buttonState = digitalRead(buttonPin); // ອ່ານສະຖານະຂອງສະວິດ
11     if (buttonState == HIGH) {                // ຖ້າສະວິດຖືກກົດ (HIGH)
12         digitalWrite(ledPin, HIGH);           // ເປີດ LED
13     } else {
14         digitalWrite(ledPin, LOW);            // ປິດ LED
15     }
16 }
```

1.5 ການທຳງານ:

ເມື່ອກົດສະວິດ Arduino ຈະອ່ານສັນຍານຈາກສະວິດ
ແລະສົ່ງສັນຍານHIGH ໄປທີ່ LED ເຮັດໃຫ້ LED ຕິດ.

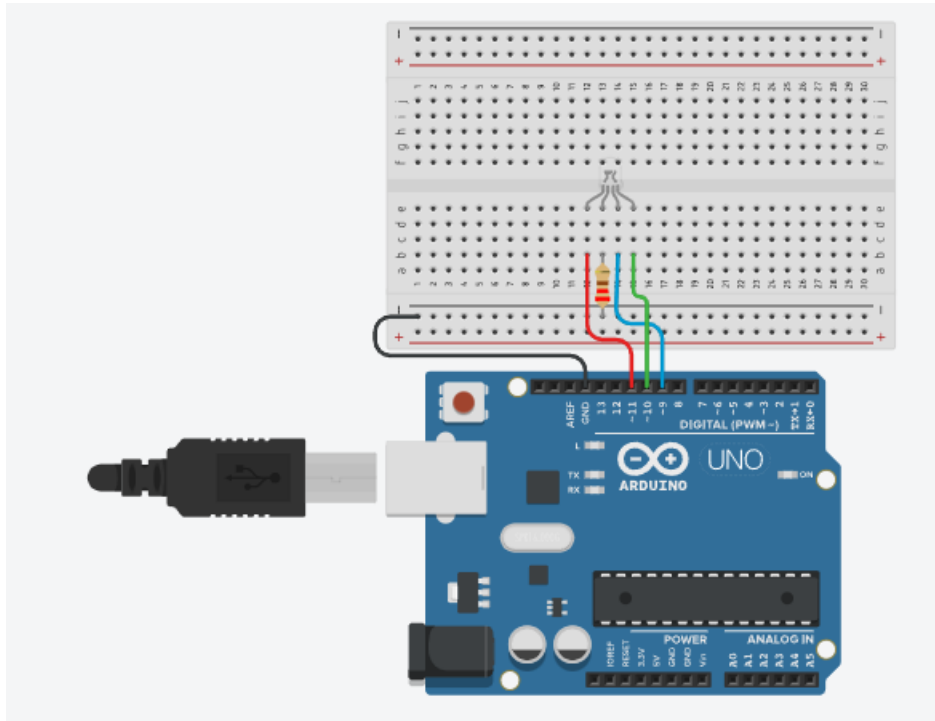
ຕົວຢ່າງ Link in Tinkercad:

https://www.tinkercad.com/things/dMgCvs3qKHW-3-rgb-or-led-with-serial?sharecode=YVUB_XQ-frLhMYtVyexzW3WWZ7HG-SA0G4nDgigX9xQ



ບົດທີ4 RGB

1. RGB or LED with Serial ແມ່ນຫຍັງ:



ທີ່ສາມາດມີແສງໄດ້ຫຼາຍສີ ໂດຍປະສົມສີແດງ (Red),
ຂຽວ (Green), ແລະສີຟ້າ (Blue)

ເຂົ້າດ້ວຍກັນ ເຊິ່ງເປັນຫຼັກການດຽວກັບລະບົບສີ RGB ທີ່
ໃຊ້ໃນຈໍພາບຫຼືອຸປະກອນສະແດງຜິນຕ່າງໆ RGB LED
ມີຂາເຊື່ອມຕໍ່ 4 ຂາ ໄດ້ແກ່ ຂາ (Anode) ຫຼື ຂາ
(Cathode) ຮ່ວມແລະຂາສໍາຫຼັບແຕ່ລະສີ (Red,
Green, Blue) ໂດຍການຄວບຄຸມຄວາມສະຫວ່າງຂອງ
ແຕ່ລະສີຈະເຮັດໃຫ້ໄດ້ສີທີ່ແຕກຕ່າງກັນອອກໄປ

Serial Communication

ຄືການສື່ສານຂໍ້ມູນລະຫວ່າງອຸປະກອນອີເລັກທຣໍນິກຜ່ານ
(Serial Port) ເຊິ່ງໃນທີ່ນີ້ໝາຍເຖິງການສົ່ງຂໍ້ມູນຈາກ
ຄອມພິວເຕີຫຼື Microcontroller (ເຊັ່ນ Arduino) ໄປທີ່
RGB LED ເພື່ອຄວບຄຸມການສະແດງຜິນຕ່າງໆ

1.1 RGB or LED with Serial ການທຳງານ:

ການທຳງານຂອງ RGB LED with Serial ໃນ

Tinkercad ເປັນຂະບວນການທີ່ໃຊ້

Microcontroller (ເຊັ່ນ Arduino) ຄວບຄຸມການສະ

ແດງຜົນຂອງ RGB LED ຜ່ານການສື່ສານແບບ

(Serial Communication) ໂດຍຜູ້ໃຊ້ສາມາດສົ່ງຄຳ

ສັ່ງຈາກຄອມພິວເຕີໄປທີ່Arduino

ເພື່ອປ່ຽນສີຂອງ RGB LED ໄດ້ຕາມຕ້ອງການ

1.11 ຕໍ່ວົງຈອນໃນ Tinkercad:

- ຂາ (Anode) ຕໍ່ກັບຂາ 5V ຫຼື GND
(ຂຶ້ນກັບປະເພດຂອງ RGB LED)
- ຂາສີແດງ (Red) ຕໍ່ກັບຂາ PWM ຂອງ Arduino
(ເຊັ່ນ ຂາ 9)
- ຂາສີຂຽວ (Green) ຕໍ່ກັບຂາ PWM ຂອງ
Arduino (ເຊັ່ນ ຂາ 10)

- ຂາສີຟ້າ (Blue) ຕໍ່ກັບຂາ PWM ຂອງ Arduino
(ເຊັ່ນ ຂາ 11)

- ຕົວຕ້ານທານ (Resistor) ທີ່ຂາແຕ່ລະສີຂອງ
RGB LED ເພື່ອປ້ອງກັນບໍ່ໃຫ້ LED ເສຍຫາຍ
ຈາກກະແສເກີນ

1.12 ຂຽນໂປຣແກຣມໃນ Arduino:

ໂປຣແກຣມຈະຮັບຄ່າສີ RGB ຈາກ Serial Monitor
ແລະສົ່ງສັນຍານ PWM ໄປຍັງຂາແຕ່

ລະສີຂອງ RGB LED ເພື່ອຄວບຄຸມຄວາມສະຫວ່າງຂອງ
ແຕ່ລະສີ

ຕົວຢ່າງ Code:

```
1  int redPin = 9;
2  int greenPin = 10;
3  int bluePin = 11;
4
5  void setup() {
6      Serial.begin(9600);
7      pinMode(redPin, OUTPUT);
8      pinMode(greenPin, OUTPUT);
9      pinMode(bluePin, OUTPUT);
10 }
11
12 void loop() {
13     if (Serial.available() > 0) {
14         int red = Serial.parseInt();
15         int green = Serial.parseInt();
16         int blue = Serial.parseInt();
17
18         analogWrite(redPin, red);
19         analogWrite(greenPin, green);
20         analogWrite(bluePin, blue);
21     }
22 }
```

1.13 ທົດສອບໃນ Tinkercad:

- ອັບໂຫຼດ Code ໄປທີ່ Arduino ໃນ Tinkercad

- ເປີດ Serial Monitor ແລະປ້ອນຄ່າສີ

RGB ໃນຮູບແບບ R G B

(ເຊັ່ນ 255 0 ສໍາຫຼັບສີແດງ)

- RGB LED ຈະປ່ຽນສີຕາມຄ່າທີ່ປ້ອນເຂົ້າໄປ

1.14 ຕົວຢ່າງການປ້ອນຄ່າໃນ Serial Monitor:

ສີແດງ: 255 0 0

ສີຂຽວ: 0 255 0

ສີນ້ຳຝາ: 0 0 255

ສີເຫຼືອງ: 255 255 0

ສີມ່ວງ: 255 0 255

ສີຟ້າ: 0 255 255

ສີຂາວ: 255 255 255

ສະຫຼຸບການທຳງານ:

1 ຜູ້ໃຊ້ປ້ອນຄ່າສີ RGB ຜ່ານ Serial Monitor

2 Arduino ຮັບຄ່າສີແລະສົ່ງສັນຍານ PWM ໄປທີ່ຂາ
ແຕ່ລະສີຂອງ RGB LED

3 RGB LED ປ່ຽນສີຕາມຄ່າທີ່ປ້ອນເຂົ້າໄປ.

ຕົວຢ່າງ Link in Tinkercad:

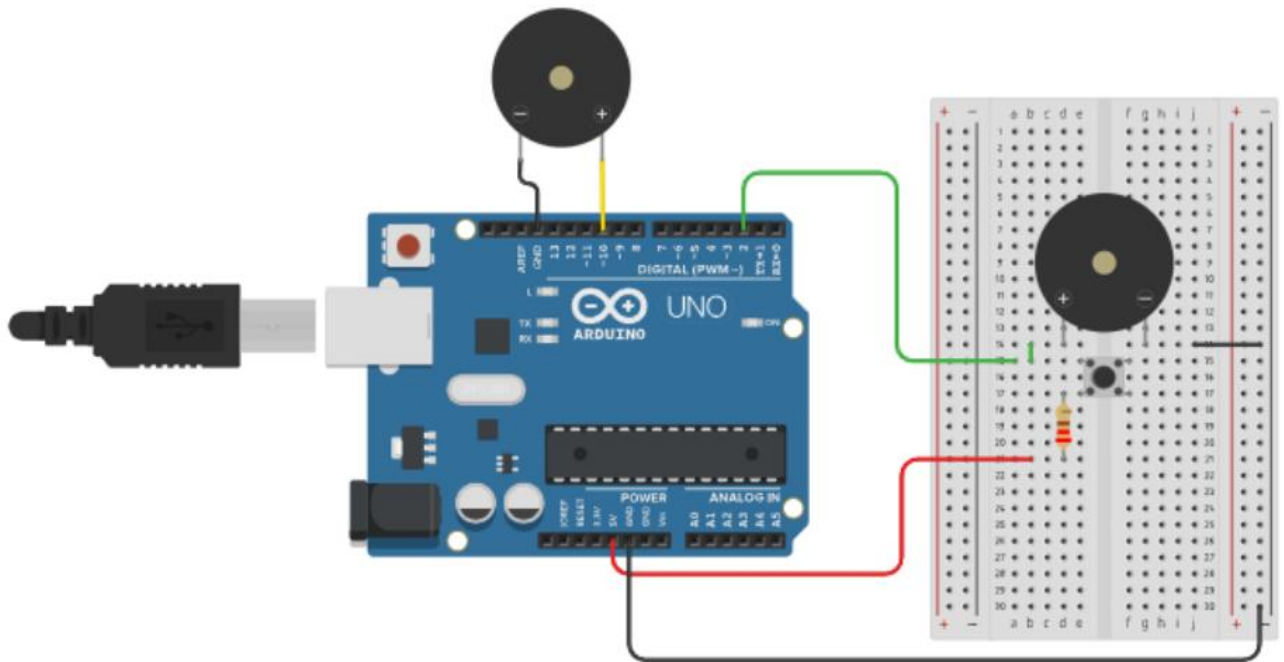
https://www.tinkercad.com/things/dMgCvs3qKHW-3-rgb-or-led-with-serial?sharecode=YVUB_XQ-frLhMYtVyexzW3WWZ7HG-SA0G4nDgigX9xQ



ບົດທີ 5 Passive Buzzer and Active Buzzer

1. Passive Buzzer and Active Buzzer

ແມ່ນຫຍັງ:



ເປັນອຸປະກອນສ້າງສຽງທີ່ໃຊ້ໃນວົງຈອນອີເລັກທຣ໌ນິກ
ໂດຍທັງສອງປະເພດມີຫຼັກການທຳງານແລະຄຸນສົມບັດທີ່
ແຕກຕ່າງກັນ ດັ່ງນີ້:

1.1 Passive Buzzer (ປັດເຊີແບບແຜຟ)

ຫຼັກການທຳງານ: Passive Buzzer ບໍ່ມີວົງຈອນ
ກຳເນີດສຍງພາຍໃນ ຕ້ອງໃຊ້ສັນຍານໄຟຟ້າພາຍນອກ
(ເຊັ່ນ ສັນຍານ PWM) ເພື່ອຄວບຄຸມຄວາມຖີ່ແລະຮູບ
ແບບຂອງສຽງ

ສຽງ: ສາມາດສ້າງສຽງໄດ້ຫຼາກຫຼາຍຄວາມຖີ່ແລະຮູບແບບ
ຂຶ້ນຢູ່ກັບສັນຍານທີ່ປ້ອນເຂົ້າ

ການໃຊ້ງານ: ຕ້ອງໃຊ້ Microcontroller ຫຼືວົງຈອນ
ສ້າງສັນຍານເພື່ອຄວບຄຸມສຽງ

ຕົວຢ່າງການໃຊ້ງານ: ໃຊ້ໃນອຸປະກອນທີ່ຕ້ອງການສຽງ
ຫຼາກຫຼາຍເຊັ່ນດົນຕີອີເລັກທຣອນິກ, ລະບົບແຈ້ງເຕືອນທີ່
ຊັບຊ້ອນ

1.2 Active Buzzer (ບັດເຊີແອກທິບ)

- ຫຼັກການທຳງານ: Active Buzzer ມີວົງຈອນກຳເນີດ
ສຽງພາຍໃນຕົວ ເຮັດໃຫ້ສາມາດສ້າງສຽງໄດ້ທັນທີເມື່ອ
ຈ່າຍໄຟໃຫ້ (ມັກເປັນໄຟ DC) ໂດຍບໍ່ຕ້ອງໃຊ້ສັນຍານ
ພາຍນອກ

- ສຽງ: ມັກສ້າງສຽງດຽວ(ເຊັ່ນ "beep") ທີ່ຄວາມຖີ່ຄົງທີ່
- ການໃຊ້ງານ: ໃຊ້ງານງ່າຍໝຽນຈ່າຍໄຟຟ້າຈະເກີດສຽງທັນທີ
- ຕົວຢ່າງການໃຊ້ງານ: ໃຊ້ໃນອຸປະກອນແຈ້ງເຕືອນງ່າຍໆເຊັ່ນ ໂມງປຸກ, ອຸປະກອນແຈ້ງເຕືອນ
- ສະຫຼຸບຄວາມແຕກຕ່າງ
- Active Buzzer: ໃຊ້ງານງ່າຍແຕ່ສ້າງສຽງໄດ້ຈຳກັດ

- Passive Buzzer: ຕ້ອງໃຊ້ສັນຍານຄວບຄຸມແຕ່
ສ້າງສຽງໄດ້ຫຼາກຫຼາຍ

ການເລືອກໃຊ້ຂຶ້ນຢູ່ກັບຄວາມຕ້ອງການຂອງໂຄງການຫາກ
ຕ້ອງການສຽງງ່າຍໆໃຫ້ເລືອກ Active Buzzer

ແຕ່ຫາກຕ້ອງການຄວບຄຸມສຽງໄດ້ລະອຽດໃຫ້ເລືອກ
Passive Buzzer.

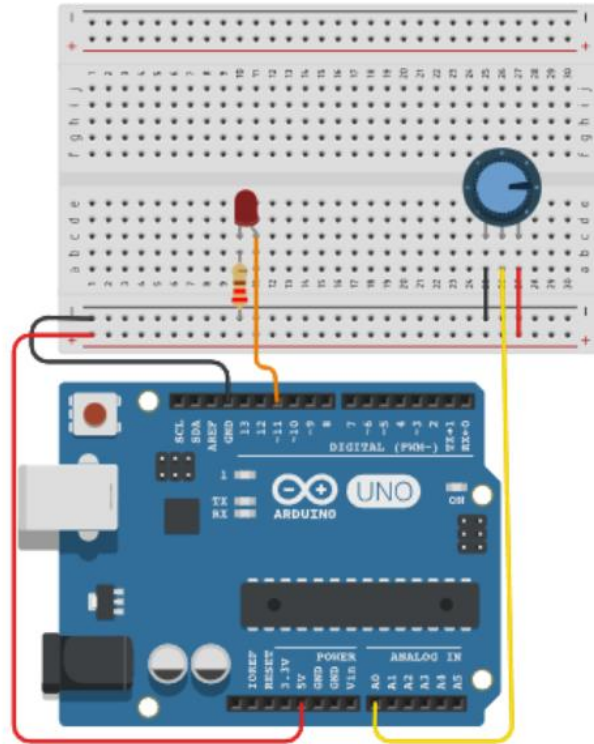
2. ຕົວຢ່າງ Link in Tinkercad:

<https://www.tinkercad.com/things/9oIOJalejpU-4-passive-buzzer-and-active-buzzer?sharecode=2Jto0pc2lrtL-1mG7yl6ffl26tMx8TCiFNaglGG099Q>



ບົດທີ 6 Potentiometer

1. Potentiometer ແມ່ນຫຍັງ:



ເປັນອຸປະກອນອີເລັກທຣ໌ນິກຊະນິດໜຶ່ງທີ່ເຮັດໜ້າທີ່ເປັນ
ຕົວຄວບຄຸມຫຼືປັບຄ່າຄວາມຕ້ານທານໄຟຟ້າໄດ້ໂດຍມັກ
ເອີ້ນສັ້ນໆວ່າ "Pot" ຫຼື "ຕົວປັບຄ່າ" ມີລັກສະນະເປັນ
ອຸປະກອນ 3 ຂາ ແລະສາມາດປັບຄ່າຄວາມຕ້ານທານ
ລະຫວ່າງຂາສອງຂາໄດ້ດ້ວຍການໝູນຫຼືເລື່ອນຕົວຄວບຄຸມ

- ສ່ວນປະກອບແລະໂຄງສ້າງຂອງ Potentiometer

1.1 ຕົວຖັງແລະແກນໝູນ: ມີແກນໝູນຫຼືສະໄລສຳຫຼັບປັບ
ຄ່າ

1.2 ຂົ້ວຕໍ່ ມີ 3 ຂາໄດ້ແກ່:

- ຂາທີ 1 ແລະຂາທີ 3: ເປັນຂາສຸດຂອງຄວາມ

ຕ້ານທານ(ຄ່າຄວາມຕ້ານທານລວມ)

- ຂາທີ 2 ເປັນຂາສຳຜັດທີ່ເລື່ອນໄດ້ (Wiper) ເຊິ່ງຈະປ່ຽນຄ່າຄວາມຕ້ານທານເມື່ອໝູນຫຼືເລື່ອນ

1.3 ແຖບຄວາມຕ້ານທານ: ພາຍໃນມີແຖບຄວາມຕ້ານທານ (Resistive Strip) ທີ່ເຮັດຈາກວັດສະດຸເຊັ່ນຄາບອນຫຼືລວດຟັນ

2.1 ຫຼັກການທຳງານ:

- ເມື່ອໝູນແກນຫຼືເລື່ອນຕົວຄວບຄຸມ ຂາສຳຜັດ (Wiper) ຈະເຄື່ອນໄປຕາມແຖບຄວາມຕ້ານທານ
- ຄ່າຄວາມຕ້ານທານລະຫວ່າງຂາສຳຜັດ (ຂາທີ 2) ແລະ ຂາອື່ນໆ ຈະປ່ຽນແປງຕາມຕຳແໜ່ງຂອງ Wiper
- ຖ້າ Wiper ຢູ່ໃກ້ຂາທີ 1 ຄ່າຄວາມຕ້ານທານ ລະຫວ່າງຂາທີ 1 ແລະຂາທີ 2 ຈະນ້ອຍລົງ

- ຖ້າ Wiper ຢູ່ໃກ້ຂາທີ 3 ຫາທີ ຄ່າຄວາມຕ້ານທານລະຫວ່າງ 1 ແລະຂາທີ 2 ຈະເພີ່ມຂຶ້ນ

2.1 ປະເພດຂອງ Potentiometer:

1. Rotary Potentiometer: ປັບຄ່າດ້ວຍການໝູນແກນ (ມັກໃຊ້ໃນອຸປະກອນປັບສຽງ ເຊັ່ນ ວິທະຍຸ)
2. Linear Potentiometer: ປັບຄ່າດ້ວຍການເລື່ອນຕົວຄວບຄຸມເປັນເສັ້ນຕົງ (ມັກໃຊ້ໃນເຄື່ອງມືວັດ)
3. Digital Potentiometer: ໃຊ້ສັນຍານ digital ໃນການຄວບຄຸມຄ່າຄວາມຕ້ານທານ (ມັກໃຊ້ໃນວົງຈອນອີເລັກທຣອນິກສະໄໝໃໝ່).

2.2 ການໃຊ້ງານ Potentiometer:

1. ປັບລະດັບສຽງ: ໃນເຄື່ອງສຽງ ເຊັ່ນ ວິທະຍຸ, ເຄື່ອງຂະຫຍາຍສຽງ
2. ຄວບຄຸມຄວາມສະຫວ່າງ: ໃນອຸປະກອນແສງສະຫວ່າງ ເຊັ່ນ ໄຟປັບຄວາມສະຫວ່າງໄດ້
3. ຄວບຄຸມຄວາມໄວ: ໃນມໍເຕີຫຼືຟັດລົມ
4. ເຊັ່ນເຊີວັດຕໍາແໜ່ງ: ໃນລະບົບຄວບຄຸມຫຼືຫຸ່ນຍົນ
5. ປັບຄ່າໃນວົງຈອນອີເລັກທຣໍນິກ: ເຊັ່ນ ຕັ້ງຄ່າຄວາມໄວຂອງວົງຈອນ.

2.3 ຕົວຢ່າງການຕໍ່ວົງຈອນ:

ຖ້າໃຊ້ Potentiometer ເປັນຕົວແບ່ງແຮງດັນ
(Voltage Divider):

ຂາທີ 1 ຕໍ່ກັບແຫຼ່ງຈ່າຍໄຟ (VCC)

ຂາທີ 2 ຕໍ່ກັບກຣາວ (GND)

ຂາທີ 3 ຈະໃຫ້ແຮງດັນອອກທີ່ປ່ຽນແປງໄດ້ຕາມຕຳແໜ່ງ
Wiper.

2.4 ສະຫຼຸບ:

Potentiometer ເປັນອຸປະກອນທີ່ໃຊ້ປັບຄ່າຄວາມ
ຕ້ານທານໄດ້ຢ່າງຕໍ່ເນື່ອງ ມີທັງແບບໝຸນແລະແບບ

ເລື່ອນ ໃຊ້ງານໃນວົງຈອນອິເລັກທຣ໌ນິກແລະອຸປະກອນ
ຕ່າງໆເພື່ອຄວບຄຸມຫຼືປັບປ່ຽນຄ່າ ເຊັ່ນ ລະດັບສຽງ,
ຄວາມສະຫວ່າງ, ຫຼືຕຳແໜ່ງ.

ຕົວຢ່າງ Link in tinkercad:

<https://www.tinkercad.com/things/hb9Z5N>

Waxlr-5-potentiometer-

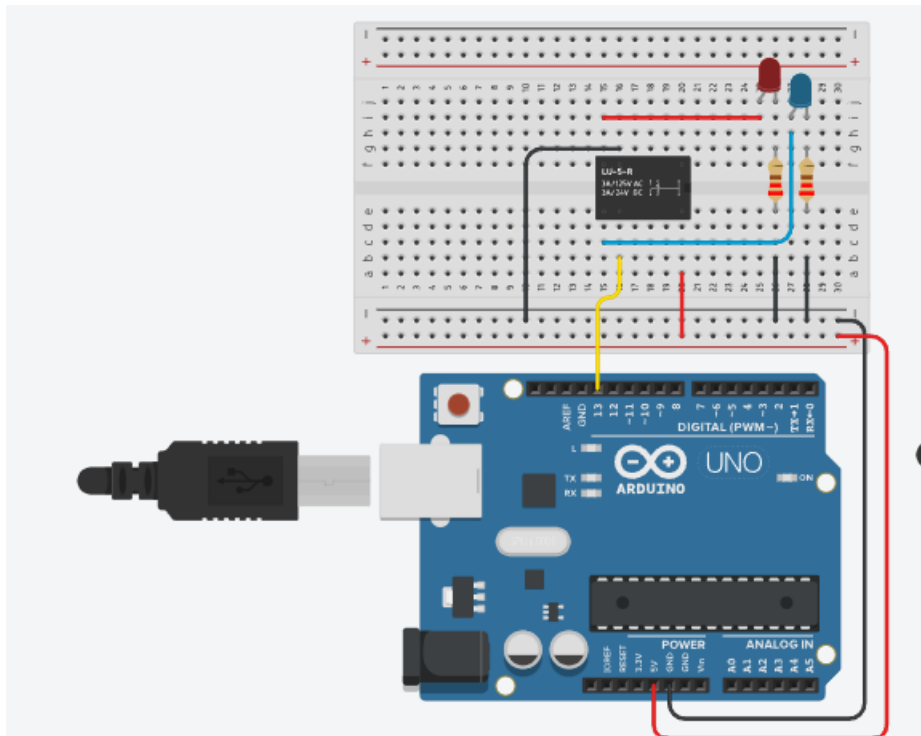
volume?sharecode=KqPM7CuKWvKno6x

AgQZkPywSxzLAscl4U_4F-26q5RU



ບົດທີ 7 Relay with 12 V

1. Relay with 12 V ແມ່ນຫຍັງ



ເປັນອຸປະກອນສະວິດໄຟຟ້າທີ່ໃຊ້ສັນຍານໄຟຟ້ານ້ອຍເພື່ອ
ຄວບຄຸມວົງຈອນໄຟຟ້າທີ່ມີກຳລັງສູງກວ່າ ໂດຍທົ່ວໄປ
Relay ປະກອບດ້ວຍມ້ວນແມ່ເຫຼັກໄຟຟ້າແລະຊຸດໜ້າ
ສຳຜັດ ເມື່ອຈ່າຍໄຟໃຫ້ມ້ວນ ຈະເກີດສະນາມແມ່ເຫຼັກ
ທີ່ດຶງໜ້າສຳຜັດໃຫ້ເປີດຫຼືປິດເຮັດໃຫ້ສາມາດຄວບຄຸມ
ອຸປະກອນອື່ນໆໄດ້.

2. ວົງຈອນ Relay 12 V ໃນ tinkercad

ໃນ tinkercad ເຈົ້າສາມາດສ້າງວົງຈອນ

Relay 12 V ໄດ້ໂດຍໃຊ້ Components

ຕ່າງໆເຊັ່ນ:

1 Relay 12 V: ອຸປະກອນ Relay ທີ່ທຳງານທີ່
ແຮງດັນ 12 V

2 ແຫຼ່ງຈ່າຍໄຟ 12 V: ສຳຫຼັບຈ່າຍໄຟໃຫ້ Relay

ສະວິດຫຼື Microcontroller:

3 ສຳຫຼັບຄວບຄຸມ Relay

4 ໂຫຼດ: ເຊັ່ນ ຫຼອດໄຟຫຼືມໍເຕີ ທີ່ Relay

ຈະຄວບ ຄຸມ

3. ຂັ້ນຕອນການສ້າງໃນ Tinkercad

- ເພີ່ມ Components: ເລືອກ Relay 12 V,

ແຫຼ່ງຈ່າຍໄຟ, ສະວິດແລະໂຫຼດ

- ຕໍ່ວົງຈອນ: ຕໍ່ຂົ້ວມ້ວນ Relay ເຂົ້າກັບ

ແຫຼ່ງຈ່າຍໄຟ 12 ແລະສະວິດ ຕໍ່ໜ້າ

ສຳຜັດ Relay ເຂົ້າກັບໂຫຼດ

- ຈຳລອງວົງຈອນ: ໃຊ້ເຄື່ອງມືຈຳລອງເພື່ອທົດສອບ
ການທຳງານຂອງ Relay

4. ຕົວຢ່າງການໃຊ້ງານ

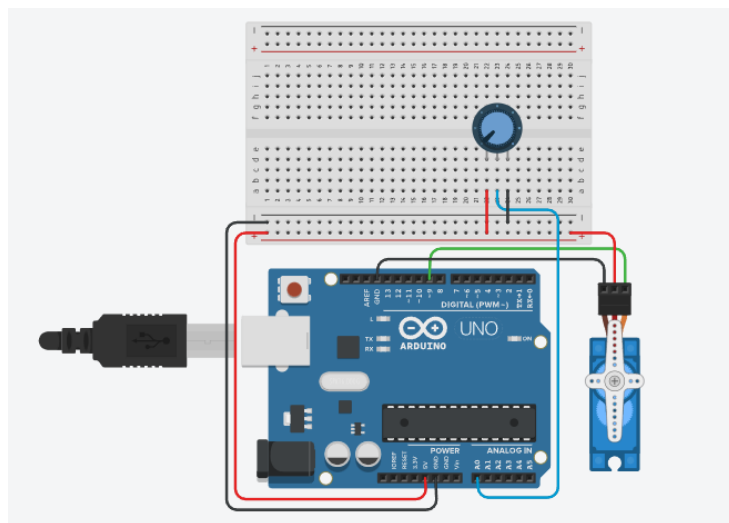
- ຄວບຄຸມຫຼອດໄຟ: ໃຊ້ Relay ເປີດປິດຫຼອດໄຟ
- ຄວບຄຸມມໍເຕີ: ໃຊ້ Relay ຄວບຄຸມມໍເຕີ

5. ສະຫຼຸບ:

"Relay with 12 V in Tinkercad" ຄືການຈຳລອງ
ວົງຈອນ Relay 12 V ໃນ Tinkercad ເຊິ່ງຊ່ວຍໃຫ້
ເຮົາເຂົ້າໃຈການທຳງານຂອງ Relay ແລະທົດສອບວົງ
ຈອນກ່ອນສ້າງຈິງ

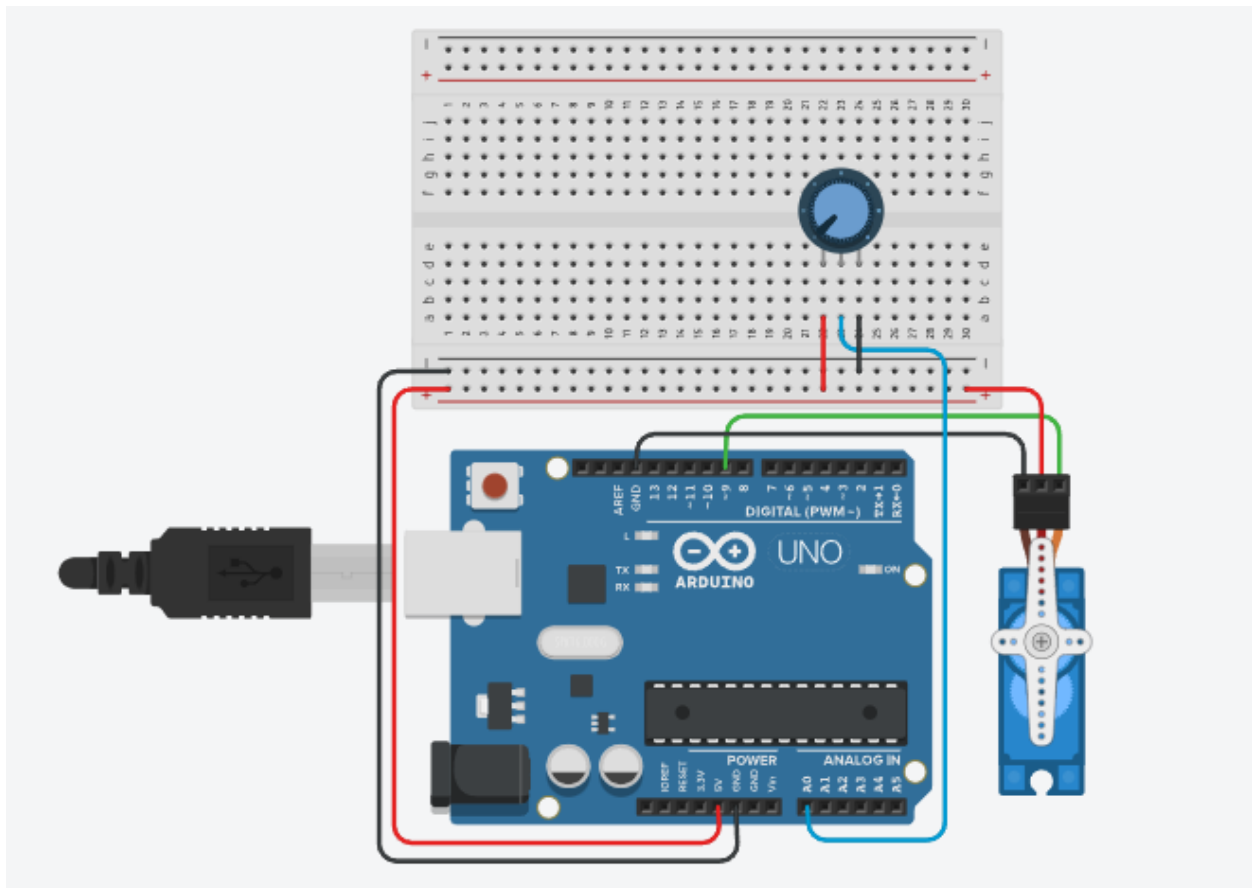
ຕົວຢ່າງ Link in tinkercad:

[https://www.tinkercad.com/things/5kojtkV7Va7-6-sweep-and-knob-servo-motor-
?sharecode=A0980kTiAqv9xDDiH55xKSm
Lm124xRyOTAMe6bWdnn8](https://www.tinkercad.com/things/5kojtkV7Va7-6-sweep-and-knob-servo-motor-?sharecode=A0980kTiAqv9xDDiH55xKSmLm124xRyOTAMe6bWdnn8)



បំពាក់ 8 servo

1. Sweep and Knob Servo Motor:



ເປັນຕົວຢ່າງການຄວບຄຸມເຊີໂວມ໌ເຕີ

(Servo Motor) ທີ່ມັກໃຊ້ໃນການຮຽນຮູ້

ແລະການທົດລອງ ໂດຍສະເພາະກັບ Arduino

ແລະໂປຣແກຣມຈຳລອງເຊັ່ນ Tinkercad.

2.ເຊີໂວມ໌ເຕີ (Servo Motor)

ເປັນມໍເຕີທີ່ສາມາດຄວບຄຸມຕຳແໜ່ງການໝູນໄດ້ຢ່າງ

ແມ່ນຍໍາ ໂດຍທົ່ວໄປສາມາດໝູນໄດ້ປະມານ 0 ຫາ 180

ອົງສາ ມັນມັກໃຊ້ໃນງານທີ່ຕ້ອງການ ການເຄື່ອນໄຫວທີ່
ແມ່ນຍຳ ເຊັ່ນ ຫຸ່ນຍົນ, ລະບົບຄວບຄຸມແລະອື່ນໆ.

3. Sweep Servo Motor

ໝາຍເຖິງການເຮັດໃຫ້ເຊີໂວມໍເຕີໝຸນໄປມາໃນມຸມທີ່
ກຳນົດ ເຊັ່ນ ຈາກ 0 ຫາ 180 ອົງສາ ແລະກັບມາທີ່
0 ອົງສາ ຕົວຢ່າງ Code Arduino ສຳຫຼັບການເຮັດ
Sweep:

ตัวอย่าง Code:

```
#include <Servo.h>

Servo myservo;  // สร้างออบเจ็กต์ Servo

int potpin = A0;  // ต่อโพเทนชิโอมิเตอร์ที่ขา A0
int val;          // ตัวแปรเก็บค่าจากโพเทนชิโอมิเตอร์

void setup() {
  myservo.attach(9);  // ต่อเซอร์โวมอเตอร์ที่ขา 9
}

void loop() {
  val = analogRead(potpin);           // อ่านค่าจากโพเทนชิโอมิเตอร์
  val = map(val, 0, 1023, 0, 180);    // แปลงค่าจาก 0-1023 เป็น 0-180
  myservo.write(val);                 // สั่งให้เซอร์โวมอเตอร์หมุนไปยังตำแหน่ง val
  delay(15);                          // รอเวลา 15 มิลลิวินาที
}
```

4.ການໃຊ້ໃນ Tinkercad

ໃນ Tinkercad ເຮົາສາມາດສ້າງວົງຈອນແລະຂຽນ
Code ເພື່ອຄວບຄຸມເຊີໂວມໂຕ້ໄດ້ຢ່າງງ່າຍດາຍ:

1.ເພີ່ມ Components: ເລືອກເຊີໂວມໂຕ້, Arduino,
ແລະ Potentiometer (ສໍາຫຼັບ Knob)

2.ຕໍ່ວົງຈອນ: ຕໍ່ເຊີໂວມໂຕ້ແລະ Potentiometer ເຂົ້າ
ກັບ Arduino

3.ຂຽນ Code: ໃຊ້ຕົວຢ່າງ Code ດ້ານເທິງຫຼືຂຽນ
Code ຂອງເຮົາເອງ

4.ຈຳລອງວົງຈອນ: ໃຊ້ເຄື່ອງມືຈຳລອງເພື່ອທົດສອບການ
ທຳງານ.

5.ສະຫຼຸບ:

- **Sweep Servo Motor:** ເຮັດໃຫ້ເຊີໂວມໂຕໜຶ່ງໄປມາໃນມຸມທີ່ກຳໜົດ
- **Knob Servo Motor:** ຄວບຄຸມເຊີໂວມໂຕດ້ວຍ Potentiometer.

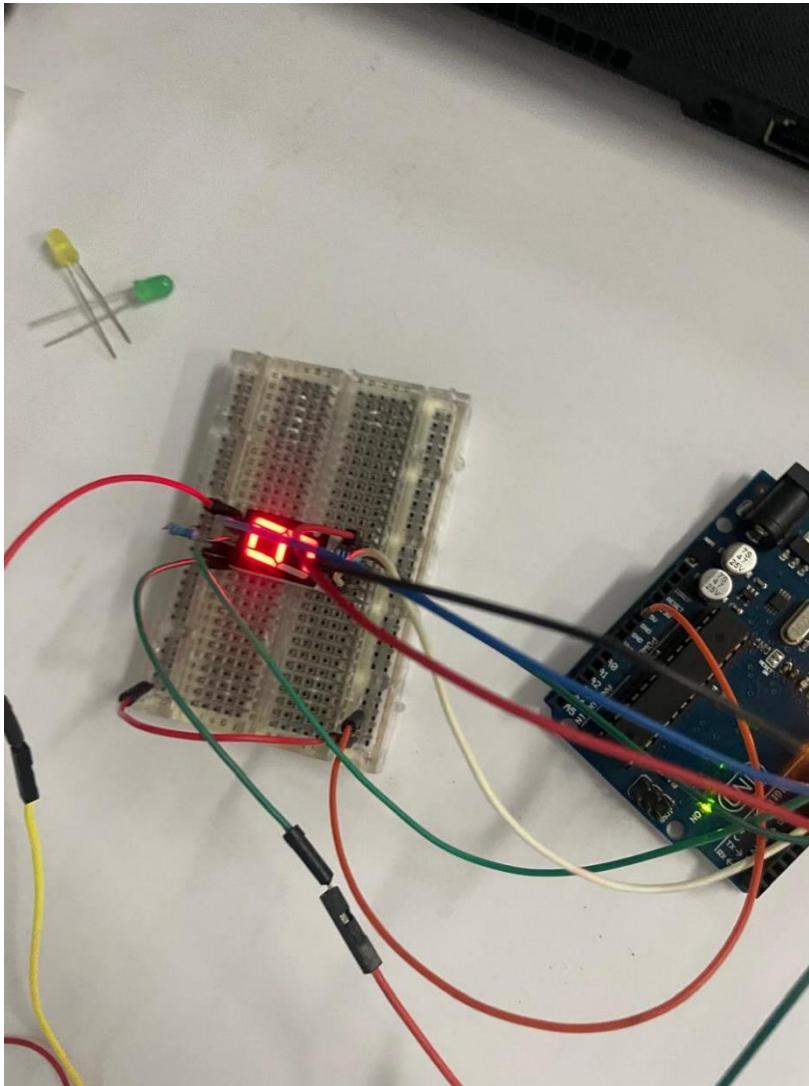
ຕົວຢ່າງ Link in tinkercad:

[https://www.tinkercad.com/things/5kojtkV7Va7-6-sweep-and-knob-servo-motor-
?sharecode=A0980kTiAqv9xDDiH55xKSm
Lm124xRyOTAMe6bWdnn8](https://www.tinkercad.com/things/5kojtkV7Va7-6-sweep-and-knob-servo-motor-?sharecode=A0980kTiAqv9xDDiH55xKSmLm124xRyOTAMe6bWdnn8)



ບົດທີ 9 seven segment

1. seven segment ແມ່ນຫຍັງ:



ເປັນອຸປະກອນສະແດງຜົນແບບດິຈິຕໍລ໌ທີ່ໃຊ້ສະແດງ
ຕົວເລກແລະບາງຕົວອັກສອນ ໂດຍປະກອບດ້ວຍ
LED (Light Emitting Diode) 7 ສ່ວນທີ່ເອີ້ນວ່າ
"segment" ເຊິ່ງແຕ່ສ່ວນສາມາດເປີດຫຼືປິດໄດ້ເພື່ອສ້າງ
ຮູບຕົວເລກຫຼືອັກສອນ.

2.ສ່ວນປະກອບຂອງ **Seven Segment**:

1/ Segment ທັງ 7 ສ່ວນ

- Segment ແຕ່ລະສ່ວນມີຊື່ເອີ້ນເປັນອັກສອນ

A ຫາ G

- ແຕ່ລະສ່ວນສາມາດເປີດຫຼືປິດໄດ້ເພື່ອສ້າງຮູບຕົວ
ເລກຫຼືອັກສອນ

2/ ຈຸດທົດນິຍົມ (Decimal Point):

- ບາງລຸ່ນມີຈຸດທົດນິຍົມ (DP) ເພື່ອສະແດງຈຸດທົດນິຍົມໃນຕົວເລກ

3.ການສະແດງຜົນ:

- ຕົວເລກ 0-9: ໂດຍການເປີດ - ປິດ Segment ທີ່ໝາະສົມ
- ຕົວອັກສອນບາງຕົວ: ເຊັ່ນ A, B, C, D, E, F

4.ປະເພດຂອງ Seven Segment:

Common Anode: ຂົ້ວບວກຂອງ ທັງໝົດເຊື່ອມຕໍ່ກັນ

Common Cathode: ຂົ້ວລົບຂອງ LED ທັງໝົດ
ເຊື່ອມຕໍ່ກັນ

5.ການໃຊ້ງານ

ເຄື່ອງຄິດເລກ

ໂມງດິຈິຕຣ໌

ເຄື່ອງວັດ

ປ້າຍສະແດງຜົນ

6.ຂໍ້ຜິດ

- ສະແດງຜົນໄດ້ຈຳກັດ

(ສະເພາະຕົວເລກແລະຕົວອັກສອນບາງຕົວ)

- ຕ້ອງການການຄວບຄຸມຫຼາຍຂາ

7.ສະຫຼຸບ:

ເປັນອຸປະກອນສະແດງຜົນທີ່ໃຊ້ກັນທົ່ວໄປໃນອຸປະກອນອີ
ເລັກທຣ໌ເນື່ອງຈາກໃຊ້ງານງ່າຍແລະລາຄາຖືກ.

ຕົວຢ່າງ Link in tinkercad:

<https://www.tinkercad.com/things/cBtpbeDxncA-ldr-counter-using-arduino>



