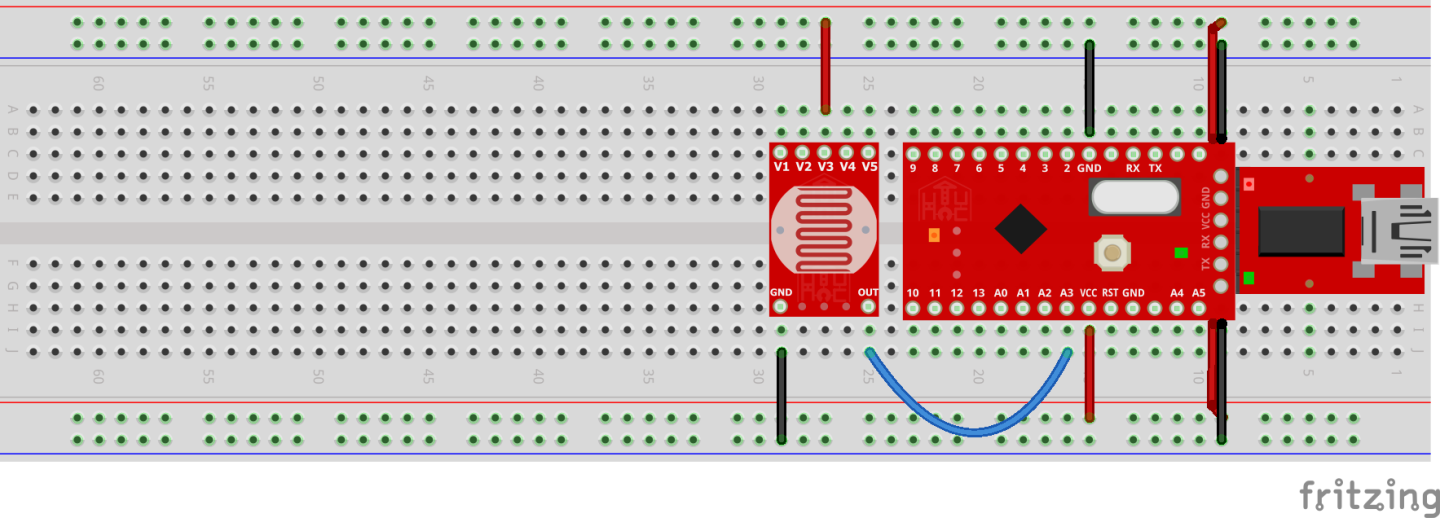
# 1. Cảm biến ánh sáng: thLight

Dùng để đo cường độ ánh sáng của môi trường:

* Giá trị cảm biến nằm trong khoảng: 0 – 1023.
* Trời càng tối thì giá trị cảm biến càng lớn và ngược lại.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với cảm biến ánh sáng:

// Arduino pin thLight pin

// A3 --- OUT

// VCC --- V3

// GND --- GND

int light\_sensor\_pin **=** A3**;**

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

pinMode**(**light\_sensor\_pin**,** INPUT**);** // cấu hình INPUT để đọc giá trị cảm biến

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

int light\_value **=** analogRead**(**light\_sensor\_pin**);** // đọc giá trị cảm biến ánh sáng

Serial**.**println**(**light\_value**);** // in kết quả ra màn hình

delay**(**250**);** // thời gian trễ giữa mỗi lần đọc

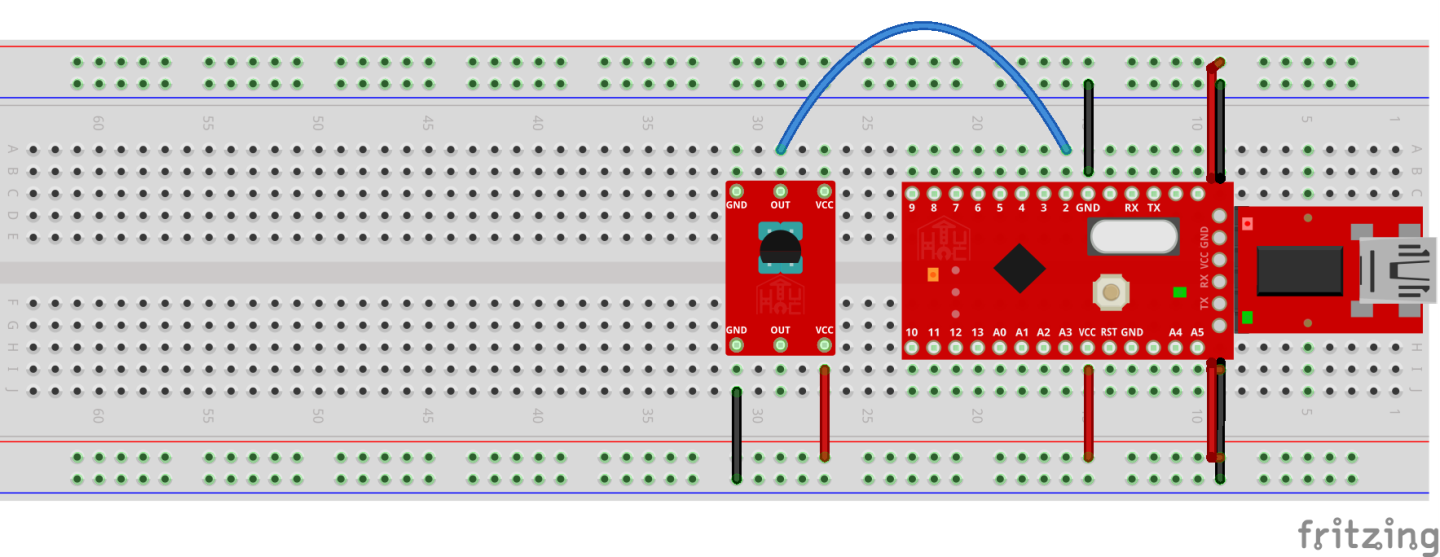
**}**

# 2. Cảm biến nhiệt độ: thTemperature

Dùng để đo nhiệt độ của môi trường:

* Dải nhiệt độ đo được: -55°C – 125°C (sai số ± 0.5°C trong khoảng -10°C – 85°C).

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với cảm biến nhiệt độ:

// Arduino pin thTemp pin

// 2 --- OUT

// VCC --- VCC

// GND --- GND

int temperature\_pin **=** 2**;**

#include <OneWire.h> // https://github.com/PaulStoffregen/OneWire/archive/master.zip

#include <DallasTemperature.h> // https://github.com/milesburton/Arduino-Temperature-Control-Library/archive/master.zip

OneWire oneWire**(**temperature\_pin**);**

DallasTemperature sensors**(&**oneWire**);**

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

sensors**.**begin**();** // cài đặt cảm biến nhiệt độ

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

sensors**.**requestTemperatures**();** // kích hoạt cảm biến nhiệt độ

float temperature **=** sensors**.**getTempCByIndex**(**0**);** // đọc giá trị nhiệt độ (C)

Serial**.**println**(**temperature**);** // in kết quả ra màn hình

delay**(**250**);** // thời gian trễ giữa mỗi lần đọc

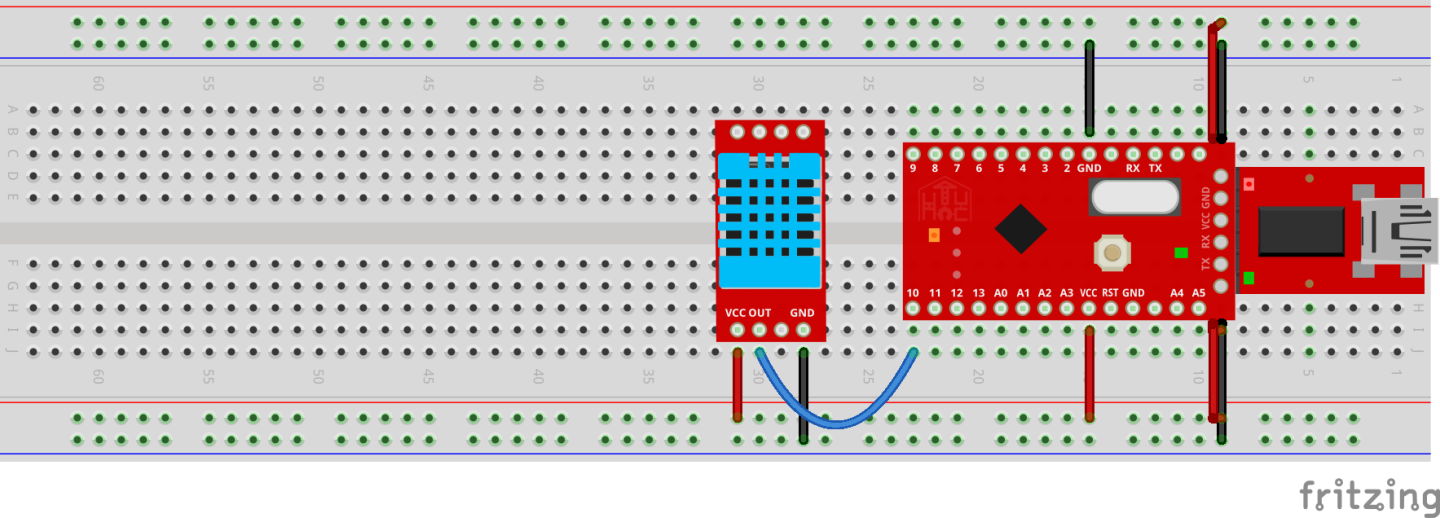
**}**

# 3. Cảm biến nhiệt độ & độ ẩm: thDHT11

Dùng để đo nhiệt độ và độ ẩm của môi trường:

* Dải nhiệt độ đo được: 0°C – 55°C (sai số ± 2°C)
* Dải độ ẩm đo được: 20 – 90% (sai số ± 5%)

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với cảm biến DHT11:

// Arduino pin DHT11 pin

// 10 --- OUT

// VCC --- VCC

// GND --- GND

int dht11\_pin **=** 10**;**

#include <dht.h> // https://arduino-info.wikispaces.com/file/view/DHT-lib.zip/545470280/DHT-lib.zip

dht DHT**;**

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

Serial**.**println**(**"Humidity (%)\tTemperature (C)"**);**

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

DHT**.**read11**(**dht11\_pin**);** // kích hoạt cảm biến DHT11

// in kết quả ra màn hình:

Serial**.**print**(**DHT**.**humidity**);** // độ ẩm (%)

Serial**.**print**(**"\t\t"**);**

Serial**.**println**(**DHT**.**temperature**);** // nhiệt độ (C)

delay**(**1000**);** // thời gian trễ giữa mỗi lần đọc

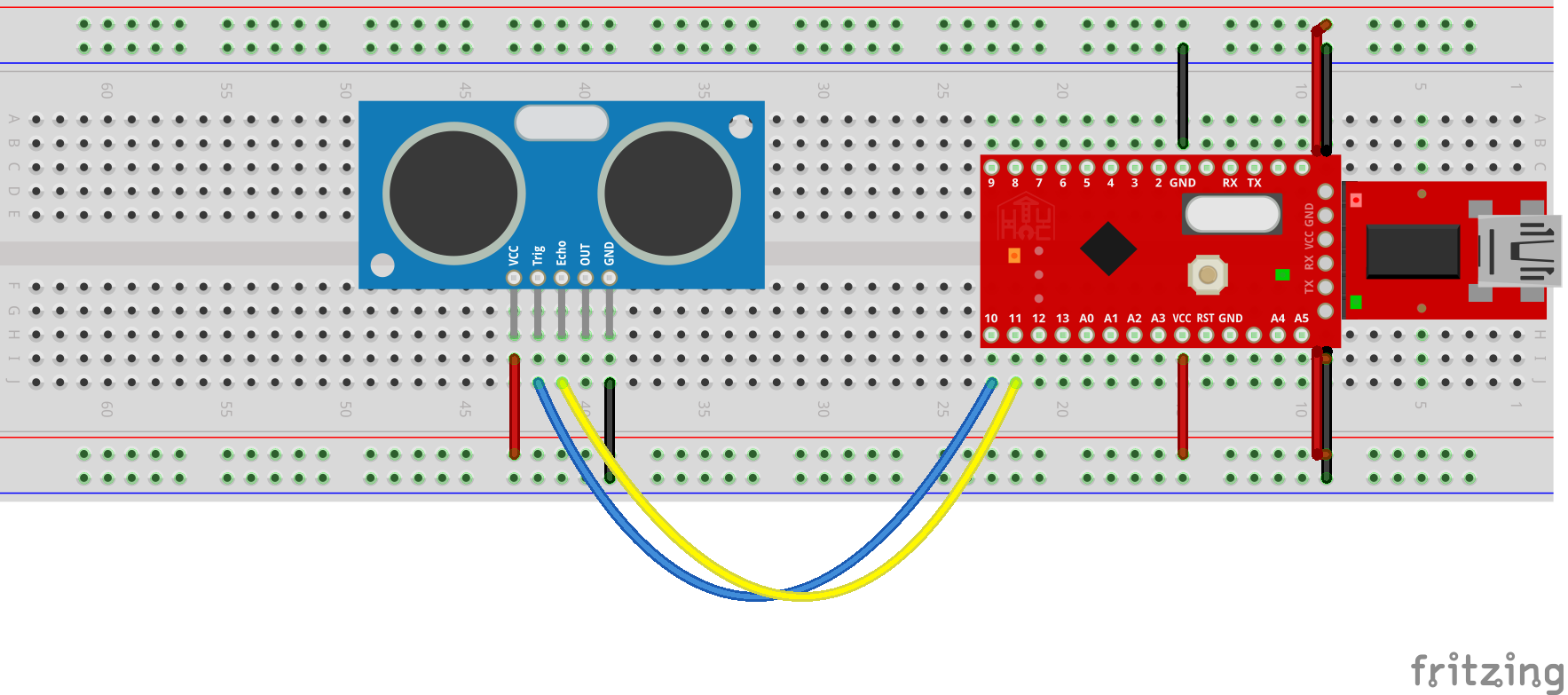
**}**

# 4. Cảm biến khoảng cách: SRF-05

Dùng để đo khoảng cách từ cảm biến tới vật thể:

* Dải khoảng cách đo được: 2 – 500 cm
* Sai số ± 2cm.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với cảm biến SRF05:

// Arduino pin SRF05 pin

// 10 --- TRIGGER

// 11 --- ECHO

// VCC --- VCC

// GND --- GND

int TRIGGER\_PIN **=** 10**;**

int ECHO\_PIN **=** 11**;**

#include <NewPing.h> // https://bitbucket.org/teckel12/arduino-new-ping/downloads/NewPing\_v1.8.zip

NewPing srf05**(**TRIGGER\_PIN**,** ECHO\_PIN**);** // Cấu hình các pin

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

int distance\_cm **=** srf05**.**ping\_cm**();** // đo khoảng cách (đơn vị cm)

Serial**.**println**(**distance\_cm**);** // in kết quả ra màn hình

delay**(**250**);** // thời gian trễ giữa mỗi lần đọc

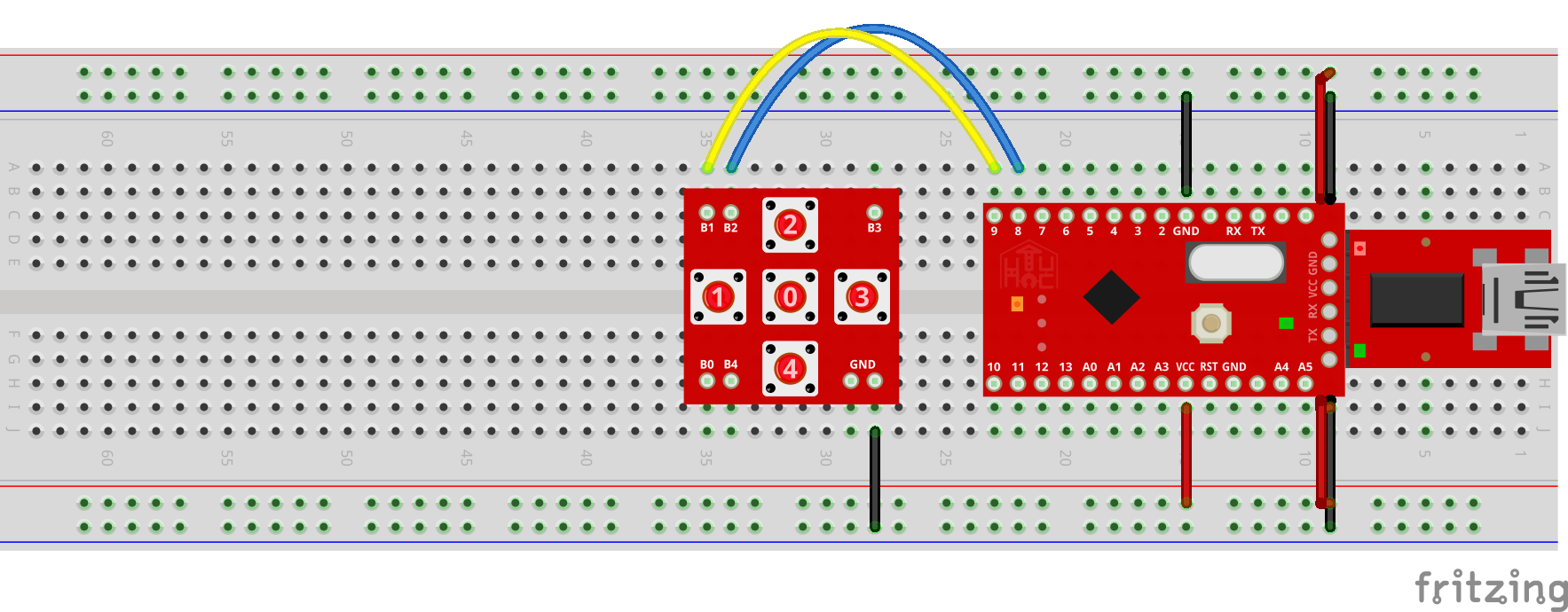
**}**

# 5. Nút bấm: thButton

Dùng để đưa tín hiệu điều khiển tới Arduino thông qua thao tác bấm:

* Gồm 5 nút bấm độc lập với nhau.
* Arduino kiểm tra trạng thái, nếu phát hiện nút được bấm thì thực hiện các nhiệm vụ tương ứng.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với nút bấm:

// Arduino pin thButton pin

// 9 --- B1

// 8 --- B2

// GND --- GND

int button1\_pin **=** 9**;**

int button2\_pin **=** 8**;**

#include <thButton.h>

thButton button1**(**button1\_pin**);**

thButton button2**(**button2\_pin**);**

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

**if** **(**button1**.**onPress**())** // nếu button1 được bấm

**{**

Serial**.**println**();**

Serial**.**println**(**"button 1 onPress"**);**

**if** **(**button2**.**isPress**())**

Serial**.**println**(**"button 2 isPress"**);**

**else**

Serial**.**println**(**"button 2 isRelease"**);**

**}**

**if** **(**button2**.**onPress**())** // nếu button2 được bấm

**{**

Serial**.**println**();**

Serial**.**println**(**"button 2 onPress"**);**

**if** **(**button1**.**isPress**())**

Serial**.**println**(**"button 1 isPress"**);**

**else**

Serial**.**println**(**"button 1 isRelease"**);**

**}**

delay**(**50**);** // thời gian trễ giữa mỗi lần đọc

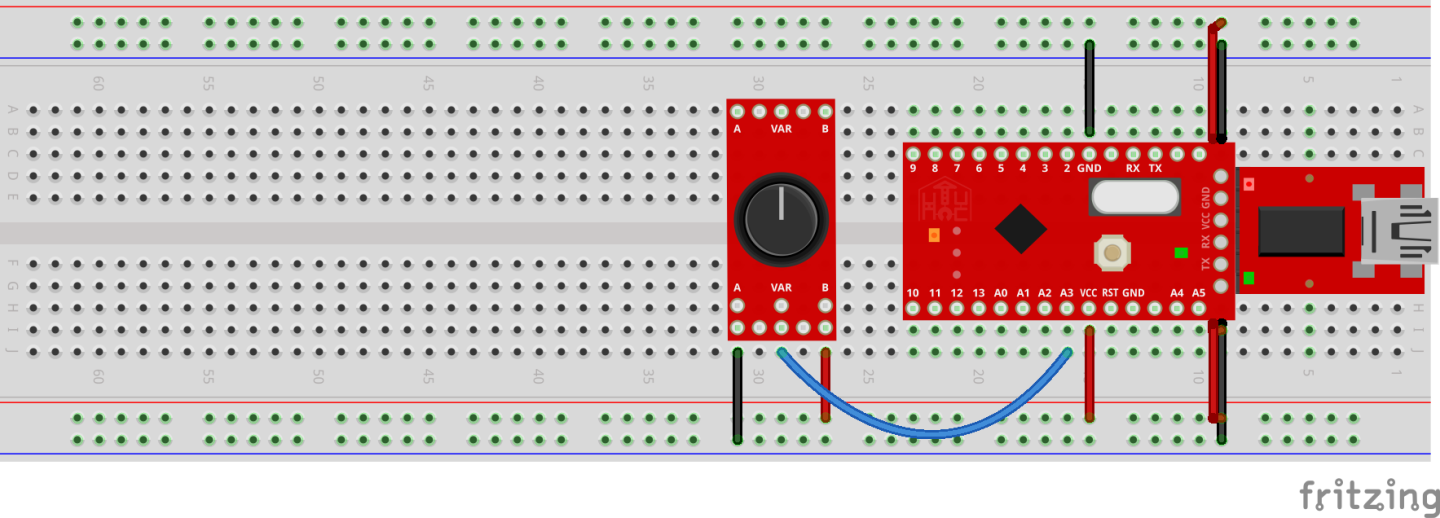
**}**

# 6. Biến trở: thKnob

Điều chỉnh giá trị điện trở bằng cách xoay núm vặn:

* Arduino kiểm tra giá trị của biến trở để đưa ra các lệnh điều khiển.
* Giá trị trả về nằm trong khoảng: 0 – 1023.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với biến trở:

// Arduino pin thKnob pin

// A3 --- VAR

// VCC --- A

// GND --- B

int knob\_sensor\_pin **=** A3**;**

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

pinMode**(**knob\_sensor\_pin**,** INPUT**);** // cấu hình INPUT để đọc giá trị cảm biến

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

int knob\_value **=** analogRead**(**knob\_sensor\_pin**);** // đọc giá trị cảm biến

Serial**.**println**(**knob\_value**);** // in kết quả ra màn hình

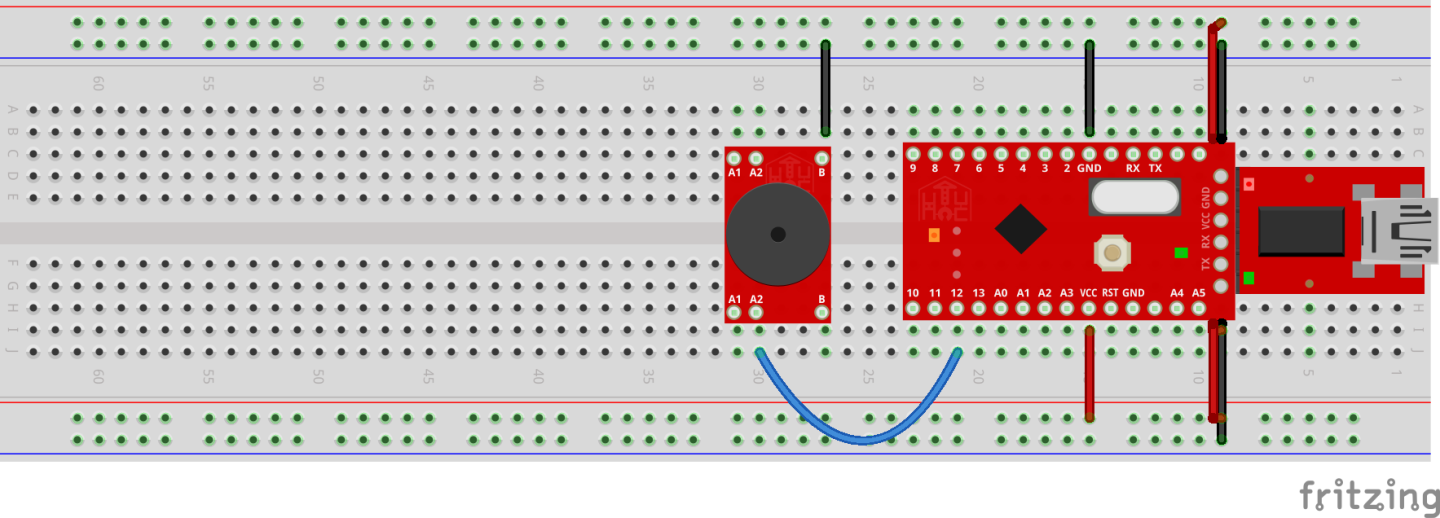
delay**(**250**);** // thời gian trễ giữa mỗi lần đọc

**}**

# 7. Loa: thBuzzer

Dùng để phát ra các âm thanh với các tần số khác nhau.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thBuzzer:

// Arduino pin thBuzzer pin

// 12 --- A2 (nối với chân A1 sẽ kêu to hơn)

// GND --- B

int buzzer\_pin **=** 12**;**

// Hàm setup() chạy một lần:

void setup**()**

**{**

pinMode**(**buzzer\_pin**,** OUTPUT**);** // cài đặt OUTPUT điều khiển buzzer

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

tone**(**buzzer\_pin**,** 440**);** // phát tần số 440Hz trong 1000ms

delay**(**1000**);** // thời gian phát

noTone**(**buzzer\_pin**);** // dừng phát

delay**(**1500**);** // thời gian trễ

tone**(**buzzer\_pin**,** 4000**);** // phát tần số 4000Hz trong 500ms

delay**(**500**);** // thời gian phát

noTone**(**buzzer\_pin**);** // dừng phát

delay**(**1000**);** // thời gian trễ

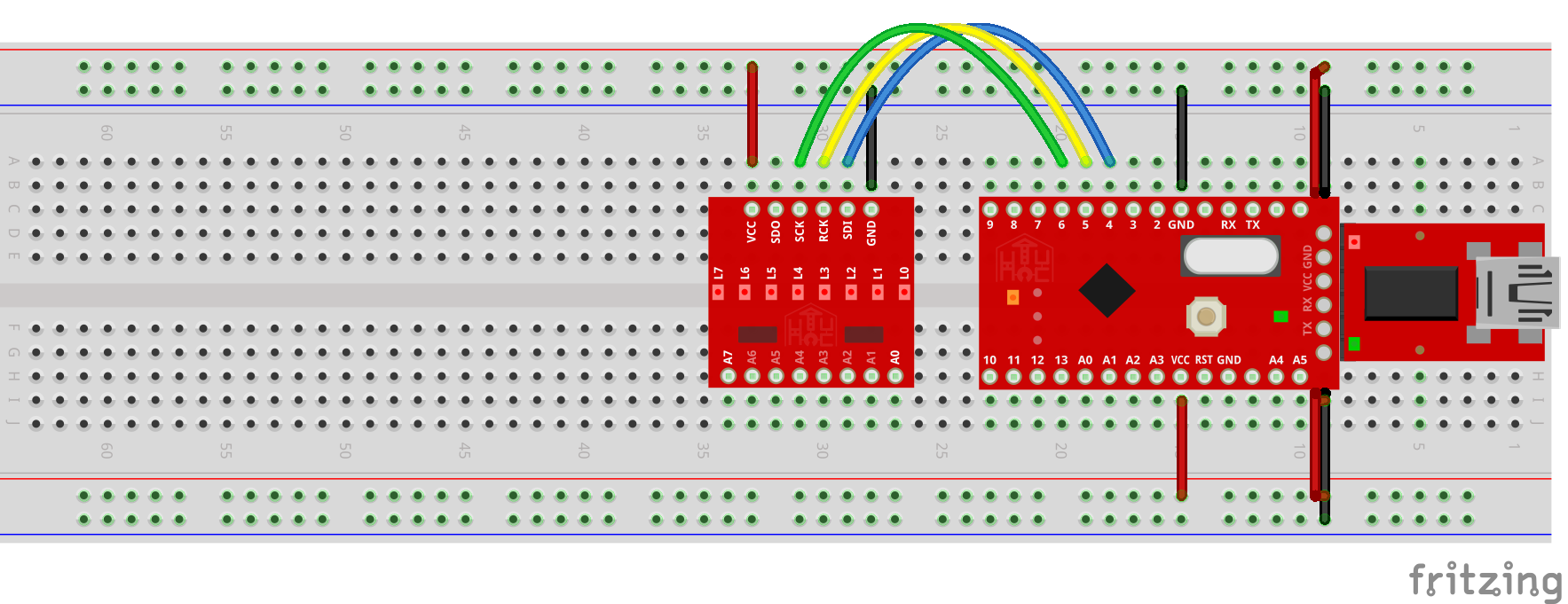
**}**

# 8. Đèn led đơn: thLed

Gồm 8 đèn led độc lập nhau:

* Có thể sử dụng để điều khiển bật / tắt hoặc tạo ra các hiệu ứng.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thLed:

// Arduino pin thLed pin

// 4 --- SDI

// 5 --- RCK

// 6 --- SCK

// VCC --- V3

// GND --- GND

int thLED\_SDI\_pin **=** 4**;**

int thLED\_RCK\_pin **=** 5**;**

int thLED\_SCK\_pin **=** 6**;**

#include <thLed.h>

// Hàm setup() chạy một lần:

void setup**()**

**{**

thLed**.**begin**(**thLED\_SDI\_pin**,** thLED\_SCK\_pin**,** thLED\_RCK\_pin**);** // cài đặt thLed

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

thLed**.**setLed**(**2**,** LED\_ON**);** // bật Led 2

thLed**.**setLed**(**7**,** LED\_ON**);** // bật Led 7

delay**(**500**);** // thời gian trễ

thLed**.**setLed**(**2**,** LED\_OFF**);** // tắt Led 2

thLed**.**setLed**(**7**,** LED\_OFF**);** // tắt Led 7

delay**(**1500**);** // thời gian trễ

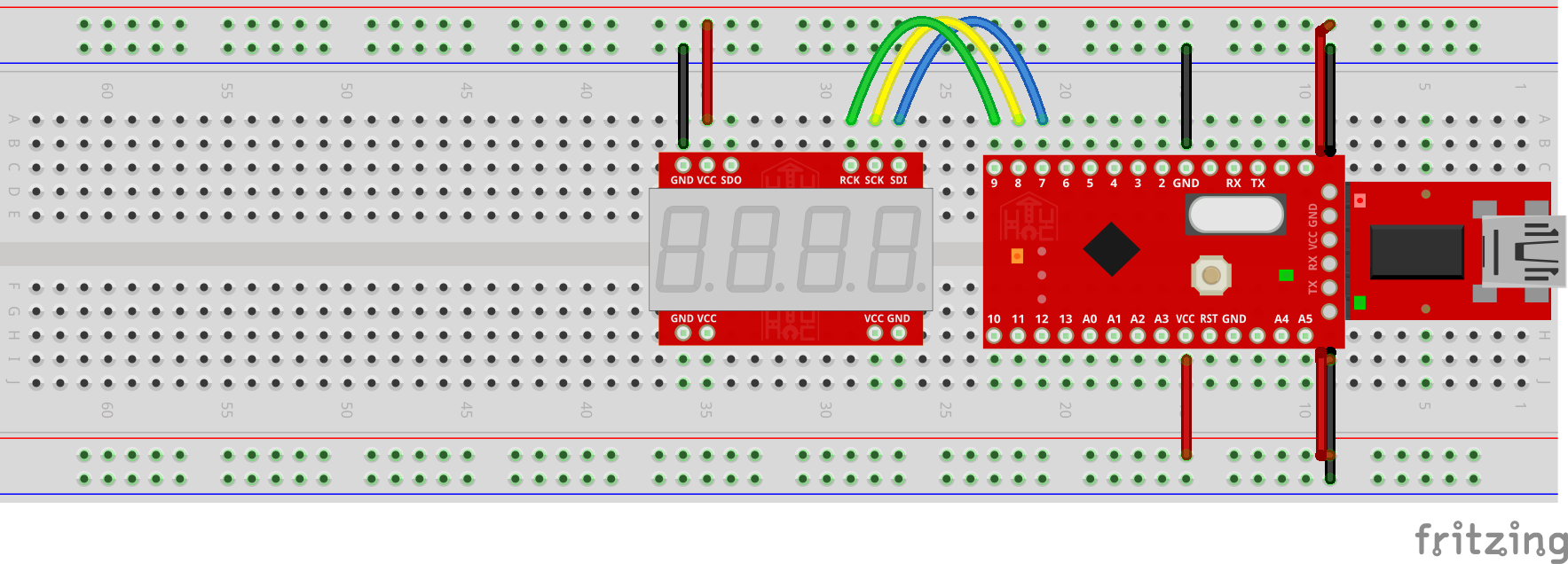
**}**

# 9. Đèn led hiển thị số: thNumber

Dùng để hiển thị các thông tin dạng số như: thời gian, nhiệt độ, độ ẩm, …

* Hiển thị được 4 chữ số: từ 0000 – 9999.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thNumber:

// Arduino pin thNumber pin

// 7 --- SDI

// 8 --- SCK

// 9 --- RCK

// VCC --- VCC

// GND --- GND

int thNumber\_SDI\_pin **=** 7**;**

int thNumber\_SCK\_pin **=** 8**;**

int thNumber\_RCK\_pin **=** 9**;**

#include <thNumber.h>

// Hàm setup() chạy một lần:

void setup**()**

**{**

thNumber**.**begin**(**thNumber\_SDI\_pin**,** thNumber\_SCK\_pin**,** thNumber\_RCK\_pin**);** // cài đặt thNumber

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

**for** **(**int value **=** 0**;** value **<** 100**;** value**++)**

**{**

thNumber**.**setNum4**(**0**,** value**);** // hiển thị giá trị value lên thNumber

thNumber**.**showDot**(**0**);** // bật hiển thị dấu chấm ở vị trí 0

delay**(**500**);**

thNumber**.**hideDot**(**0**);** // tắt hiển thị dấu chấm ở vị trí 0

delay**(**500**);**

**}**

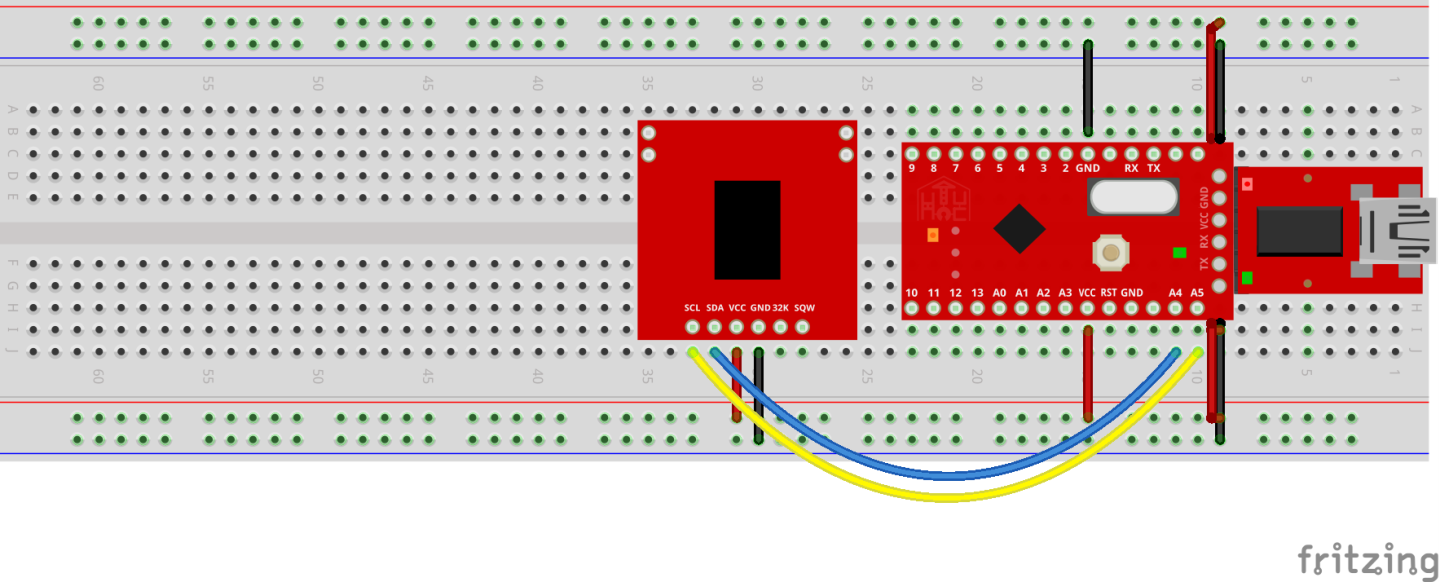
**}**

# 10. Đồng hồ thời gian thực: thRTC

Lưu thông tin về thời gian hiện tại: ngày, tháng, năm, giờ, phút, giây.

* Arduino đọc dữ liệu thời gian từ thRTC để hiển thị hoặc hẹn giờ thiết bị…
* Có thể cài đặt lại được thời gian.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thRTC:

// Arduino pin thRTC pin

// A4 --- SDA

// A5 --- SCL

// VCC --- VCC

// GND --- GND

#include <Wire.h>

#include <TimeLib.h> // https://github.com/PaulStoffregen/Time/archive/master.zip

#include <DS1307RTC.h> // https://github.com/PaulStoffregen/DS1307RTC/archive/master.zip

// Hàm setup() chạy một lần:

void setup**()**

**{**

Serial**.**begin**(**9600**);** // khởi tạo kết nối tới máy tính với tốc độ 9600

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

tmElements\_t currentTime**;**

RTC**.**read**(**currentTime**);** // đọc dữ liệu từ RTC

// hiển thị: ngày / tháng / năm

Serial**.**print**(**currentTime**.**Day**);**

Serial**.**print**(**"/"**);**

Serial**.**print**(**currentTime**.**Month**);**

Serial**.**print**(**"/"**);**

Serial**.**print**(**tmYearToCalendar**(**currentTime**.**Year**));**

// hiển thị: giờ : phút : giây

Serial**.**print**(**", "**);**

print2digits**(**currentTime**.**Hour**);**

Serial**.**print**(**":"**);**

print2digits**(**currentTime**.**Minute**);**

Serial**.**print**(**":"**);**

print2digits**(**currentTime**.**Second**);**

Serial**.**println**();**

delay**(**1000**);** // cách 1s đọc một lần

**}**

void print2digits**(**int number**)**

**{**

**if** **(**number **>=** 0 **&&** number **<** 10**)**

Serial**.**write**(**'0'**);**

Serial**.**print**(**number**);**

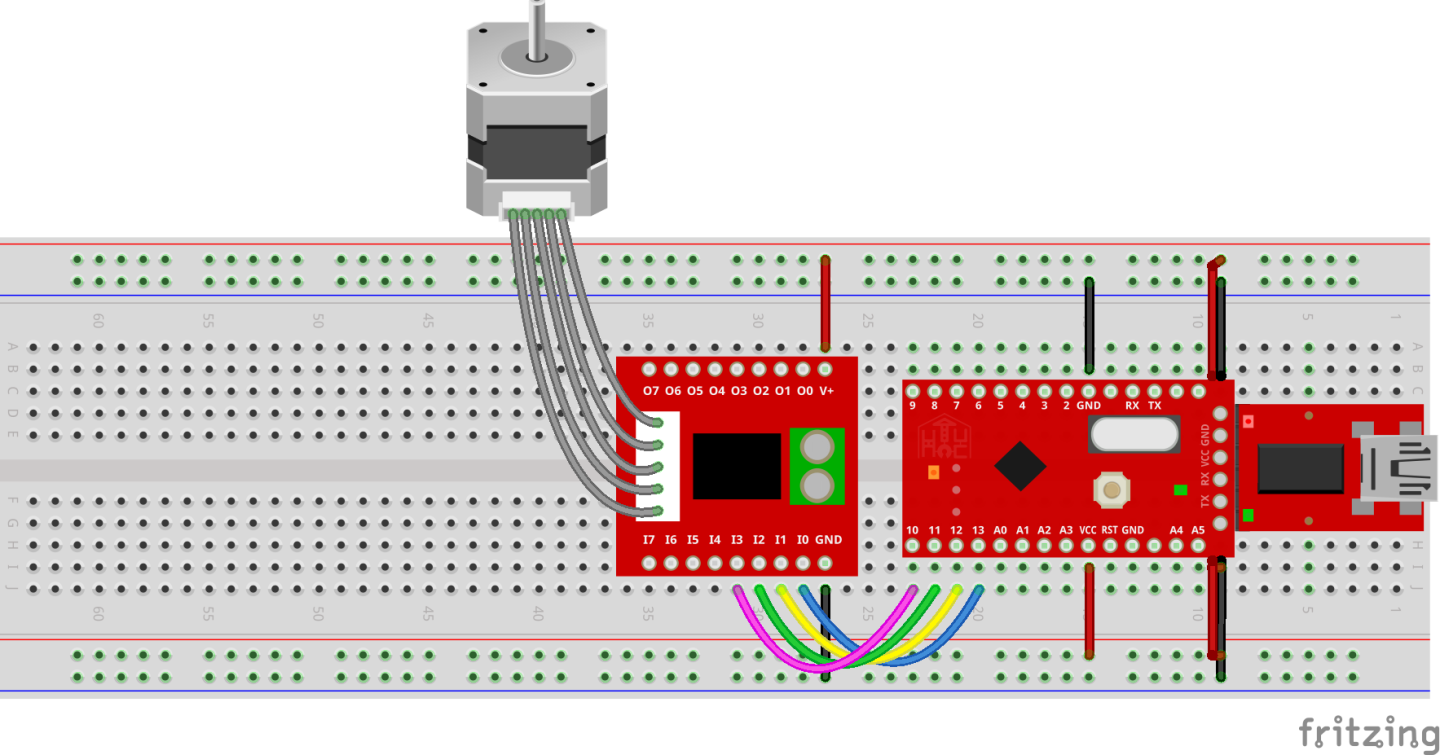
**}**

# 11. Động cơ bước: thStepper

Thường được sử dụng để điều khiển: rèm, cửa cuốn, …

* Có thể thay đổi được tốc độ và chiều quay của động cơ.
* Có thể điều khiển chính xác số vòng quay mong muốn.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thStepper:

// Arduino pin thStepper pin

// 10 --- I3

// 11 --- I2

// 12 --- I1

// 13 --- I0

// VCC --- V+

// GND --- GND

int I0\_pin **=** 13**;**

int I1\_pin **=** 12**;**

int I2\_pin **=** 11**;**

int I3\_pin **=** 10**;**

#include <Stepper.h>

Stepper myStepper**(**64**,** I0\_pin**,** I2\_pin**,** I1\_pin**,** I3\_pin**);**

int stepsPerRevolution **=** 2048**;** // số bước để động cơ quay đủ 1 vòng

// Hàm setup() chạy một lần:

void setup**()**

**{**

**}**

// Hàm loop() chạy lặp lại vô hạn:

void loop**()**

**{**

int mySpeed **=** 300**;**

myStepper**.**setSpeed**(**mySpeed**);** // đặt tốc độ = 300

myStepper**.**step**(**stepsPerRevolution**);** // quay thuận 1 vòng

delay**(**1000**);** // dừng 1s

mySpeed **=** 150**;**

myStepper**.**setSpeed**(**mySpeed**);** // đặt tốc độ = 150

myStepper**.**step**(-**stepsPerRevolution **/** 2**);** // quay ngược 1/2 vòng

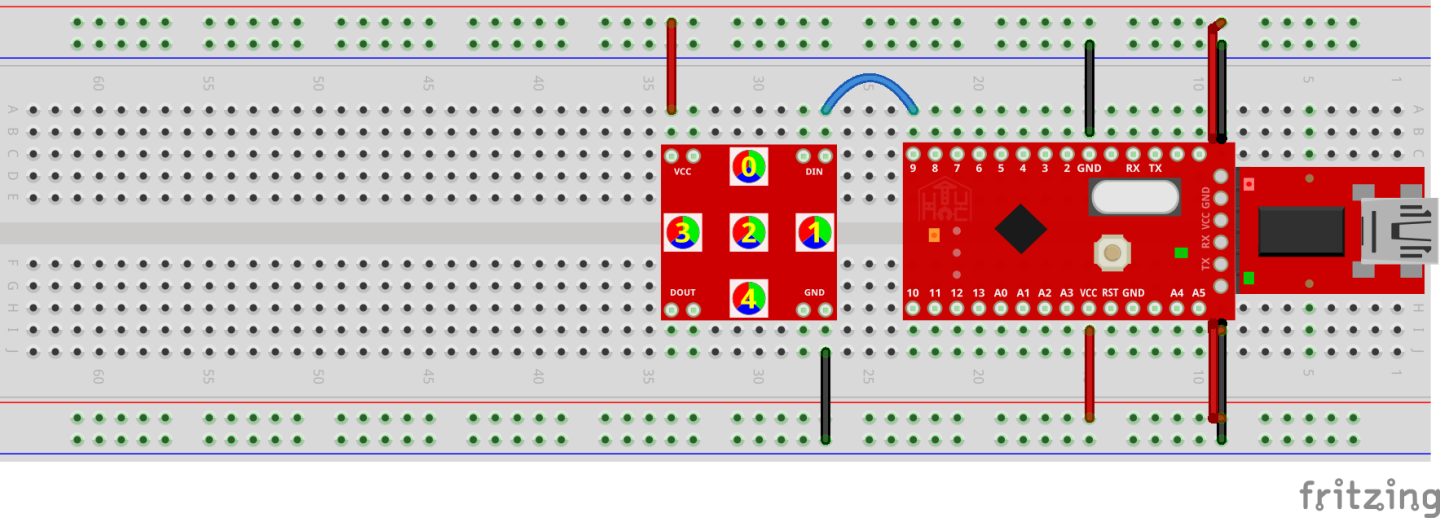
delay**(**1000**);** // dừng 1s

**}**

# 12. Led nhiều màu: thColorLed

* Gồm 5 led: có thể điều chỉnh màu sắc, độ sáng của từng Led.
* Có thể ghép nối tiếp nhiều module với nhau.

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thColorLed:

// Arduino pin thColorLed pin

// 9 --- DIN

// VCC --- VCC

// GND --- GND

#include <FastLED.h> // FastLED library - http://fastled.io

#define LED\_PIN 9

#define NUM\_LEDS (5\*1) // total number of LEDs

#define GLOBAL\_BRIGHTNESS 15 // Max brightness = 255 (very bright)

#define LED\_UP 0 // Vị trí các LED trên module thColorLed

#define LED\_RIGHT 1

#define LED\_CENTER 2

#define LED\_LEFT 3

#define LED\_DOWN 4

CRGB leds**[**NUM\_LEDS**];**

void setup**()**

**{**

FastLED**.**addLeds**<**WS2812B**,** LED\_PIN**,** GRB**>(**leds**,** NUM\_LEDS**);**

FastLED**.**setBrightness**(**GLOBAL\_BRIGHTNESS**);**

**}**

void loop**()**

**{**

leds**[**LED\_UP**]** **=** CRGB**::**Red**;**

leds**[**LED\_LEFT**]** **=** CRGB**::**Green**;**

leds**[**LED\_DOWN**]** **=** CRGB**::**Blue**;**

leds**[**LED\_RIGHT**]** **=** CRGB**::**Yellow**;**

leds**[**LED\_CENTER**]** **=** CRGB**(**255**,**255**,**255**);** // or CRGB::White

FastLED**.**show**();** // Cập nhật hiển thị các Led

delay**(**2500**);**

leds**[**LED\_UP**]** **=** CRGB**::**Green**;**

leds**[**LED\_LEFT**]** **=** CRGB**::**Blue**;**

leds**[**LED\_RIGHT**]** **=** CRGB**::**Red**;**

int hsv\_saturation **=** 255**;**

int hsv\_value **=** 255**;**

**for** **(**int hsv\_hue **=** 0**;** hsv\_hue **<** 256**;** hsv\_hue**++)**

**{**

leds**[**LED\_CENTER**]** **=** CHSV**(**hsv\_hue**,** hsv\_saturation**,** hsv\_value**);**

leds**[**LED\_DOWN**].**setHue**(**hsv\_hue **+** 128**);**

FastLED**.**show**();** // Cập nhật hiển thị các Led

delay**(**20**);**

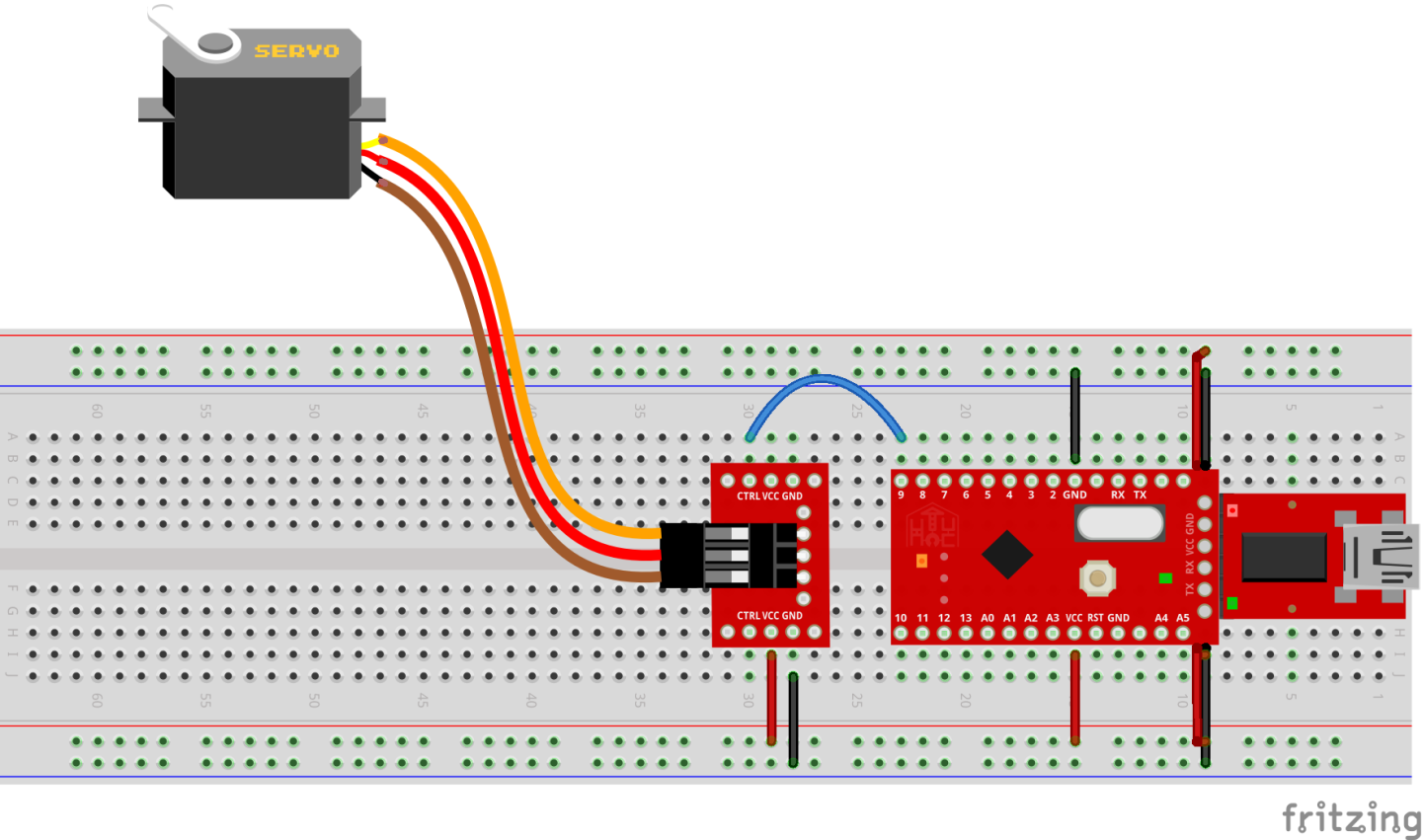
**}**

**}**

# 13. Động cơ servo: thServo

* Servo là một loại động cơ đặc biệt, chỉ quay được các góc bất kì nằm trong khoảng từ 0° - 180°.
* Thường dùng trong các ứng dụng đặc biệt đòi hỏi độ chính xác cao, ví dụ: điều chỉnh camera, điều khiển cánh tay robot, rada địa hình,…

## a) Sơ đồ kết nối mạch:



## b) Code mẫu:

// Kết nối Arduino với thServo

// Arduino pin thServo pin

// 9 --- CTRL (orange)

// VCC --- VCC (red)

// GND --- GND (brown)

int myServo\_pin **=** 9**;**

#include <Servo.h> // sử dụng thư viện Servo

Servo myServo**;** // khai báo myServo

// Hàm setup() chạy một lần

void setup**()**

**{**

int SG90\_minPulse **=** 700**;**

int SG90\_maxPulse **=** 2300**;**

myServo**.**attach**(**myServo\_pin**,** SG90\_minPulse**,** SG90\_maxPulse**);** // khởi tạo myServo

**}**

// Hàm loop() chạy lặp lại vô hạn

void loop**()**

**{**

int waittime **=** 15**;** // thời gian đợi servo quay (ms)

int pos **=** 90**;** // vị trí servo

**for** **(;** pos **<=** 180**;** pos **+=** 1**)** // quay servo từ 90 đến 180 độ

**{**

myServo**.**write**(**pos**);** // quay myServo đến vị trí 'pos'

delay**(**waittime**);** // đợi servo quay

**}**

**for** **(;** pos **>=** 90**;** pos **-=** 1**)** // quay servo từ 180 đến 90 độ

**{**

myServo**.**write**(**pos**);** // quay myServo đến vị trí 'pos'

delay**(**waittime**);**

**}**

delay**(**2000**);** // dừng 2 giây

**for** **(;** pos **>=** 0**;** pos **-=** 1**)** // quay servo từ 90 đến 0 độ

**{**

myServo**.**write**(**pos**);**

delay**(**waittime**);**

**}**

**for** **(;** pos **<=** 90**;** pos **+=** 1**)** // quay servo từ 0 đến 90 độ

**{**

myServo**.**write**(**pos**);**

delay**(**waittime**);**

**}**

delay**(**2000**);** // dừng 2 giây

**}**