

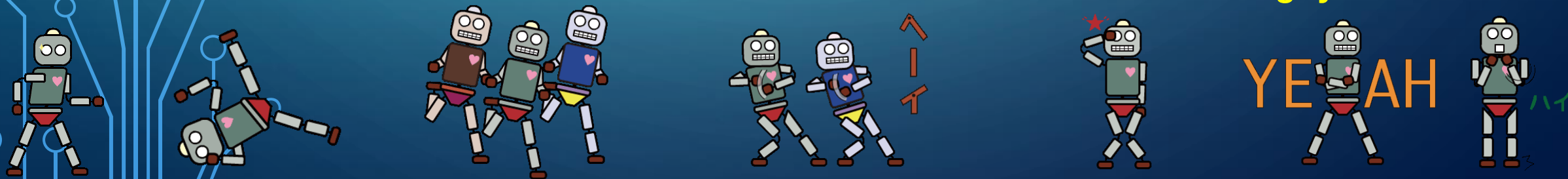


TRƯỜNG CAO ĐẲNG LÝ TỰ TRỌNG TP.HCM KHOA ĐIỆN – ĐIỆN TỬ

HỌC PHẦN

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

GV: **Trần Nguyên Bảo Trân**



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

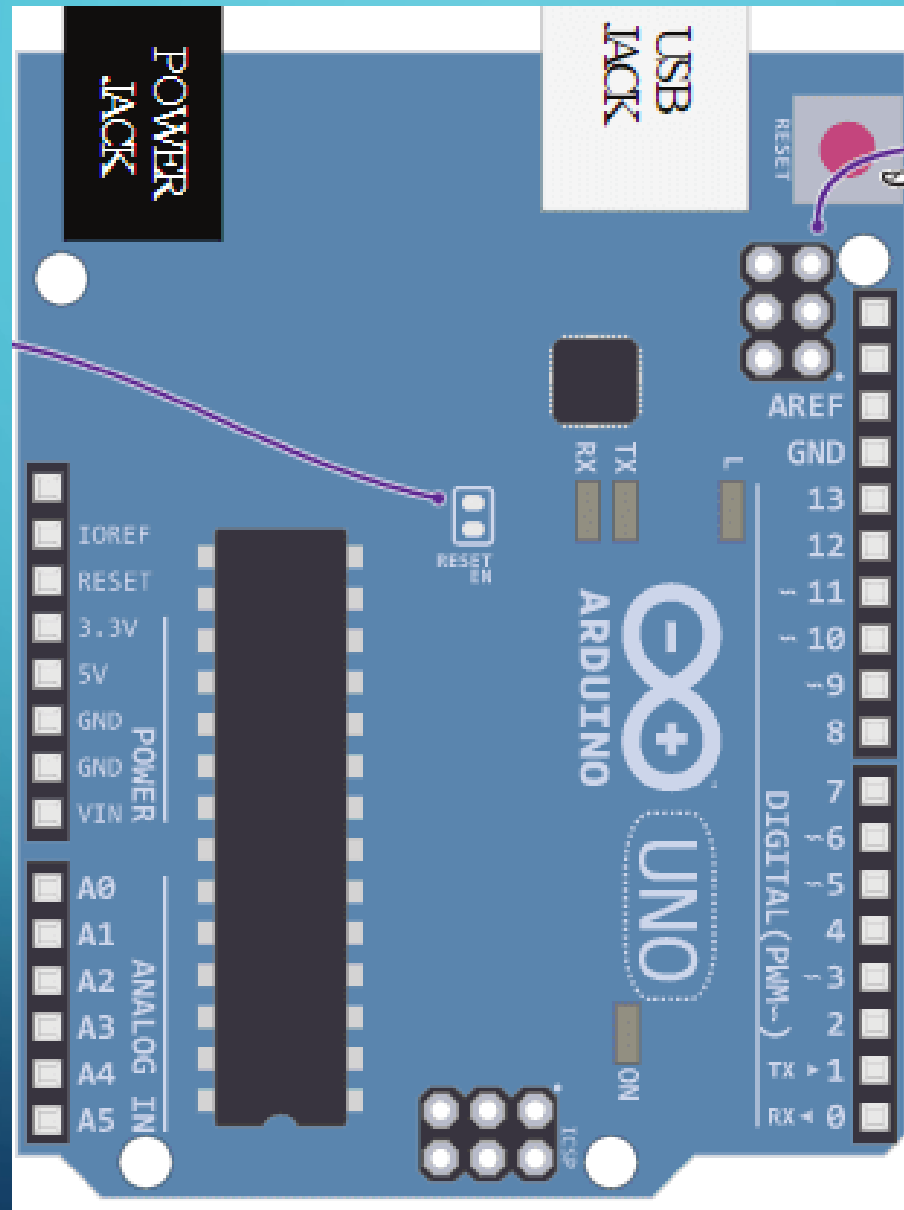
1. Khái niệm chung
2. Multi Function Shield
3. Đọc trạng thái nút nhấn
4. Đọc giá trị analog

Bài tập



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

Các chân analog input đo lường một phạm vi điện áp đặt trên chân



Các chân digital input cảm nhận mức **LOW** (không có điện áp) hay mức **HIGH** (có điện áp) trên chân

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

DIGITAL

- Dùng hàm ***pinMode(pin, INPUT)*** để thiết lập chân ***pin*** là ngõ vào số (digital input)
- Dùng hàm ***digitalRead(pin)*** để đọc trạng thái ngõ vào trên chân ***pin***
- Trên board UNO R3 có 14 chân digital (từ 0 đến 13)
trong đó, chân **0** và **1** (ký hiệu **Rx** và **Tx**) được dùng cho kết nối USB với máy tính, không dùng cho chức năng khác
- Nếu cần nhiều chân digital thêm nữa, có thể dùng các chân analog (từ 0 đến 5) như là các chân digital (từ 14 đến 19)

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

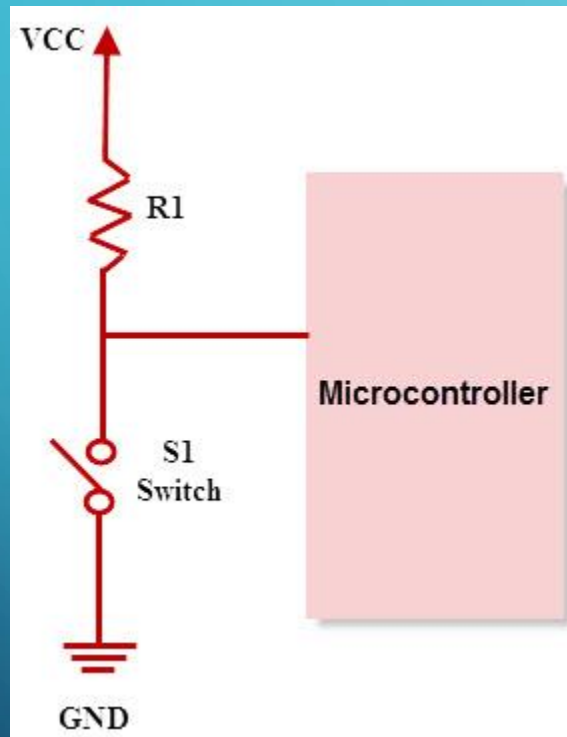
Các hằng số sử dụng trong lập trình Arduino

Constant	Pin Number
A0	Analog input 0 (Digital 14)
A1	Analog input 1 (Digital 15)
A2	Analog input (Digital 16)
A3	Analog input (Digital 17)
A4	Analog input (Digital 18)
A5	Analog input (Digital 19)

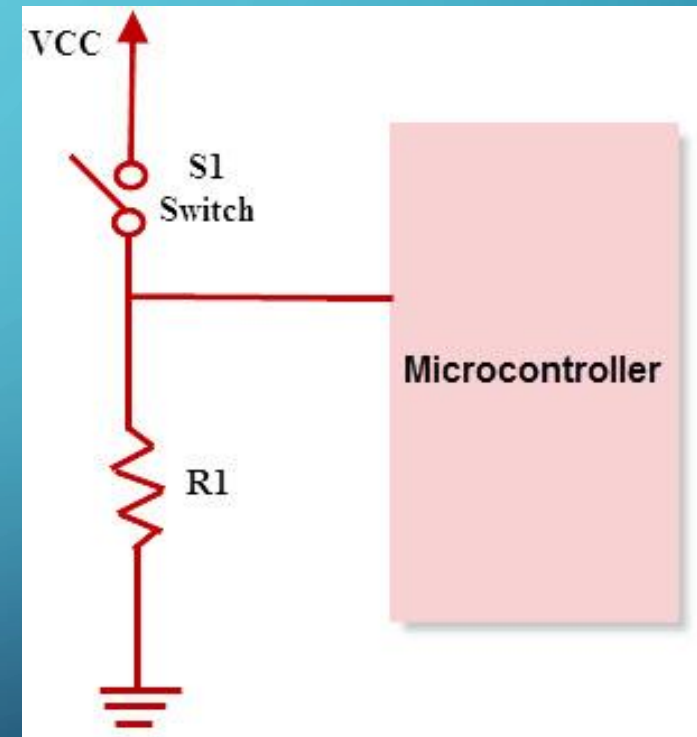
Constant	Pin Number
LED_BUILTIN	On-board LED (Digital 13)
SDA	I2C Data (Digital 18)
SCL	I2C Clock (Digital 19)
SS	SPI Select (Digital 10)
MOSI	SPI Input (Digital 11)
MISO	SPI Output (Digital 12)
SCL	SPI Clock (Digital 13)

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

Điện trở kéo lên (*pull-up resistor*)



Điện trở kéo xuống (*pull-down resistor*)



$R1 = 10K$

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

ANALOG

Các giá trị analog là giá trị liên tục, tăng giảm trong một vùng giá trị

- Dùng hàm ***analogRead(pin)*** để đọc giá trị tỉ lệ với điện áp trên chân pin
hàm ***analogRead()*** luôn trả về 1 số nguyên nằm trong khoảng từ 0 đến 1023 tương ứng với thang điện áp (mặc định) từ 0 đến 5V

có thể điều chỉnh thang điện áp này bằng hàm ***analogReference()***

Lưu ý: *Nếu chưa kết nối chân đọc điện áp, hàm ***analogRead()*** sẽ trả về một giá trị ngẫu nhiên trong khoảng từ 0 đến 1023.*

Để khắc phục điều này, phải mắc thêm một điện trở có trị số lớn (khoảng 10k ohm trở lên) hoặc một tụ điện 104 từ chân đọc điện áp xuống GND.

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

ANALOG

hàm ***analogReference()*** có nhiệm vụ đặt lại mức (điện áp) tối đa khi đọc tín hiệu `analogRead`

Cú pháp ***analogReference(type)***

type: một trong các giá trị: DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V56, hoặc EXTERNAL

Xem thêm tại: <http://arduino.vn/reference/analogreference>

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.2 Multi Function Shield

4 LED đơn ở PIN 13, 12, 11, 10

4 LED 7 đoạn giao tiếp SPI (dùng IC Driver 74HC595D)

Chân cắm cảm biến nhiệt độ DS18B20 hoặc LM35 (Pin A4)

Chân cắm bộ thu hồng ngoại (Pin 2)

Chân cắm kết nối điều khiển động cơ Servo (Pin 5, 9, A5)

Chân cắm kết nối module Bluetooth nhận dạng giọng nói APC220 (Pin 6)

Tích hợp **biến trở** (Pin A0)

4 Button : 1 Reset Button, 3 Button để test (Pin A1, A2, A3)

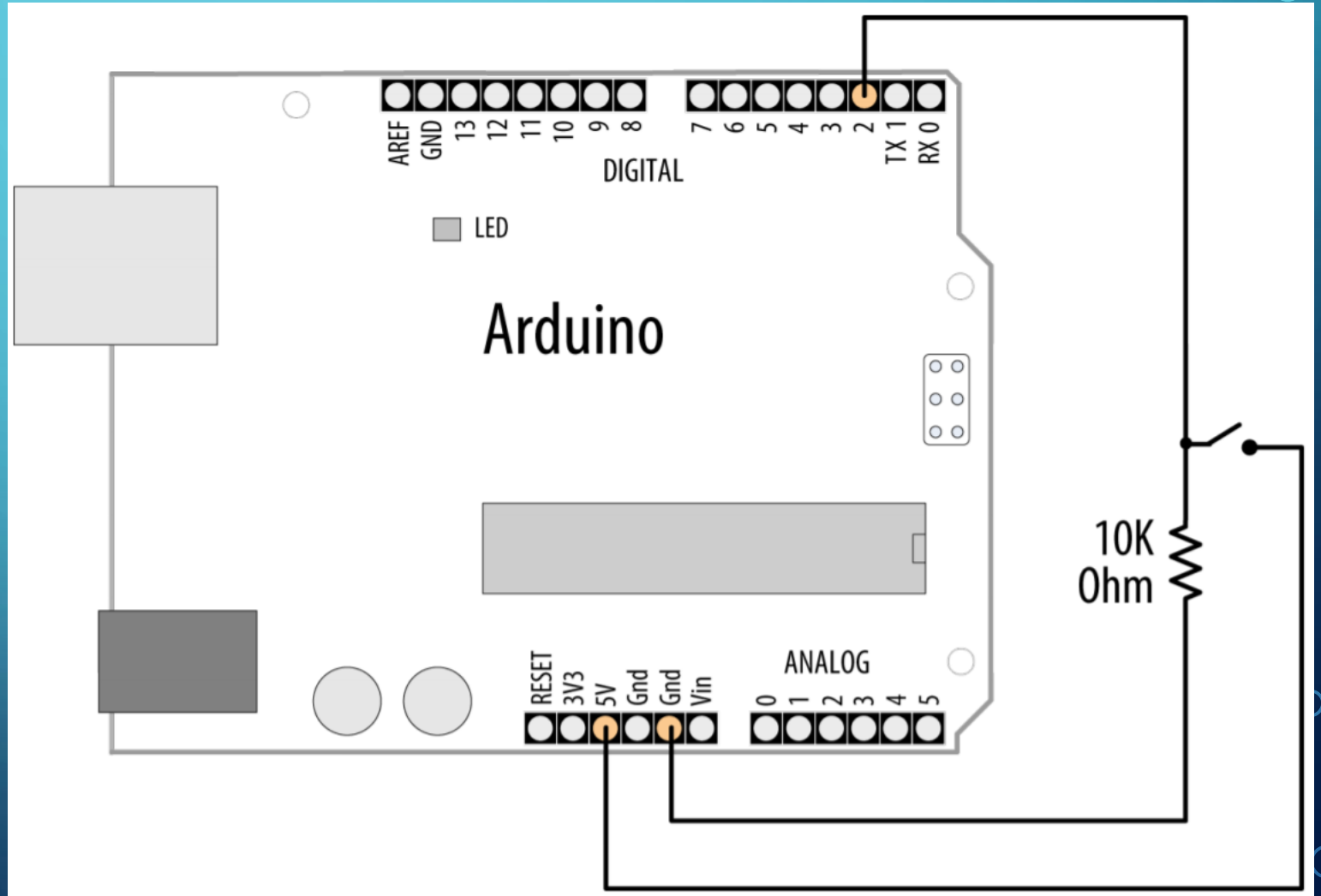
1 còi Buzzer (Pin 3)



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.3 Đọc trạng thái nút nhấn

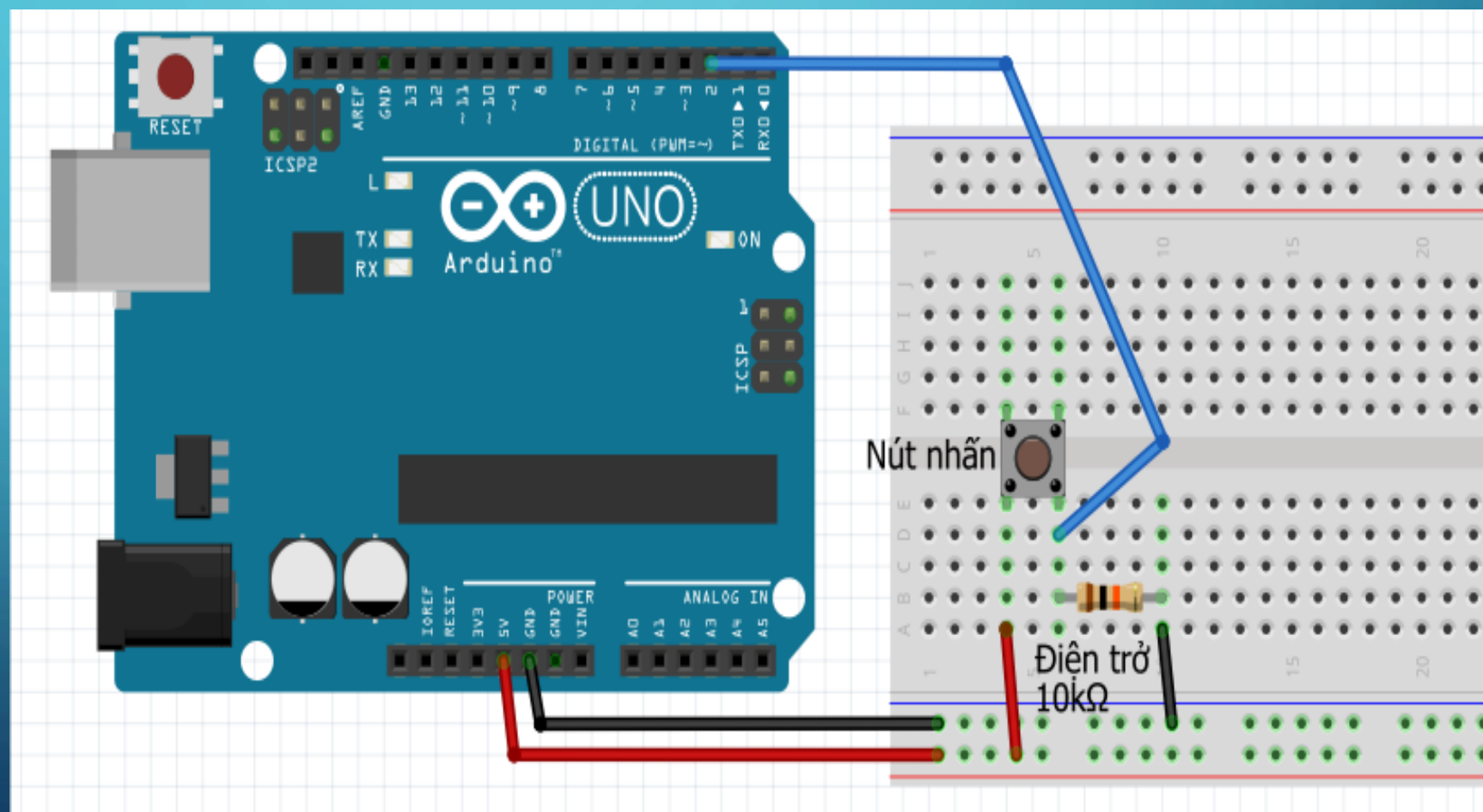
Sơ đồ mạch nguyên lý



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.3 Đọc trạng thái nút nhấn

Sơ đồ kết nối mạch



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TỰ ĐỘNG TỰ

3.3 Đọc trạng thái nút nhấn

Chương trình

```
int button = 2;

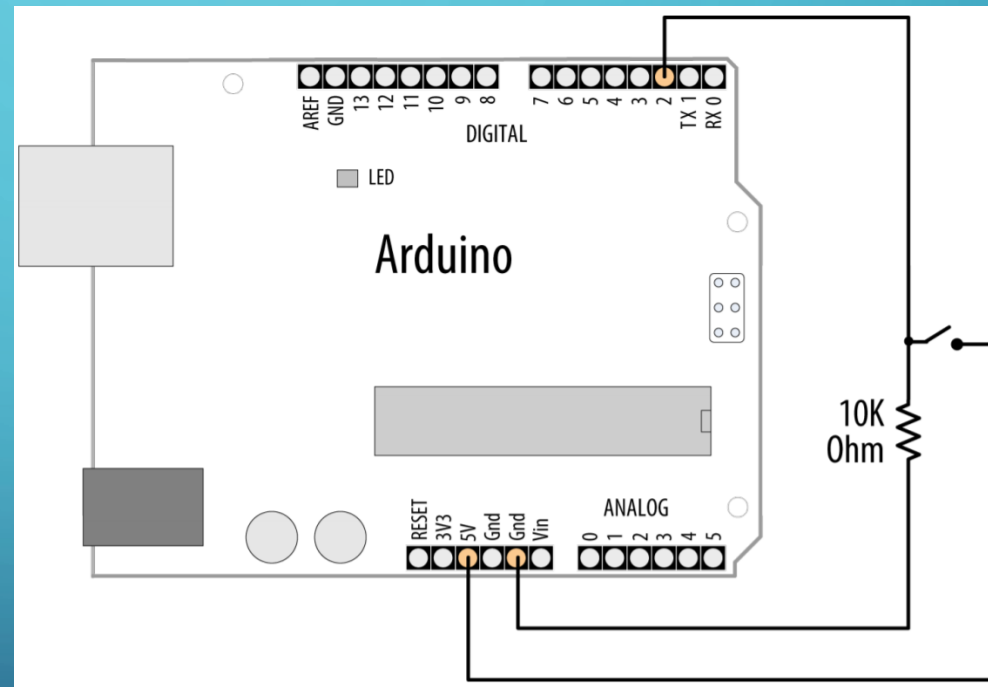
void setup() {
  Serial.begin(9600);
  pinMode(button, INPUT);           //Cài đặt chân D2 ở trạng thái đọc dữ liệu
}

void loop() {
  int buttonStatus = digitalRead(button); //Đọc trạng thái button
  Serial.println(buttonStatus);           //Xuất trạng thái button
  delay(200);                             //Chờ 200ms
}
```


Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.3 Đọc trạng thái nút nhấn

Bài tập 1: Cho sơ đồ mạch nguyên lý sau đây



Viết chương trình điều khiển sao cho khi nhấn nút thì đèn LED ở chân 13 sáng, khi nhả nút thì đèn LED tắt.

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.3 Đọc trạng thái nút nhấn

Chương trình

```
const int ledPin = 13;           // choose the pin for the LED
const int inputPin = 2;         // choose the input pin (for a pushbutton)
void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare pushbutton as input
}
void loop(){
  int val = digitalRead(inputPin); // read input value
  if (val == HIGH)                // check if the input is HIGH
  {
    digitalWrite(ledPin, HIGH);   // turn LED on if switch is pressed
  }
  else
  {
    digitalWrite(ledPin, LOW);    // turn LED off
  }
}
```

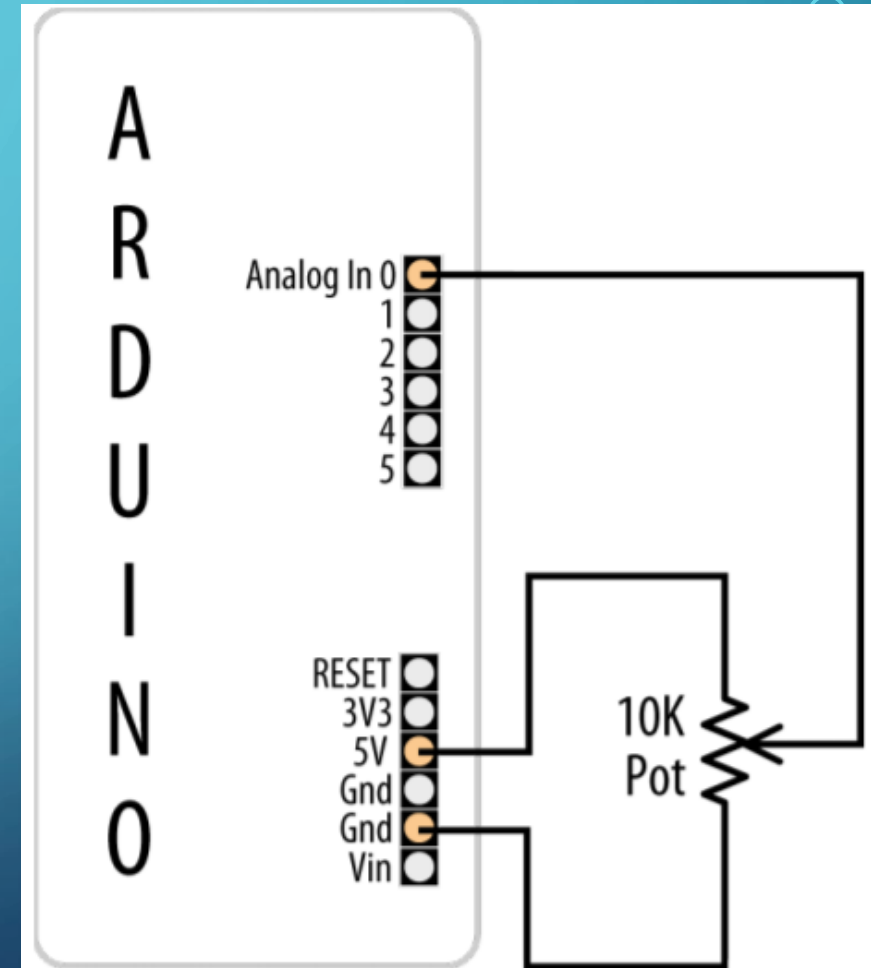
Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.4 Đọc giá trị analog

Sơ đồ mạch nguyên lý

Bài tập 2:

Viết chương trình điều khiển tốc độ nháy của đèn LED ở chân 13 sang bằng biến trở



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.4 Đọc giá trị analog

Chương trình

```
const int potPin = 0;           // biến trở gắn vào chân 0
const int ledPin = 13;         // chân gắn LED
int val = 0;                   // biến sẽ lưu giá trị đọc từ biến trở
void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop() {
  val = analogRead(potPin);     // đọc điện áp trên biến trở
  digitalWrite(ledPin, HIGH);   // bật LED sáng
  delay(val);                   // tạo trễ theo giá trị đọc được từ biến trở
  digitalWrite(ledPin, LOW);    // tắt LED
  delay(val);
}
```

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

map()

là hàm dùng để chuyển một giá trị từ thang đo này sang một giá trị ở thang đo khác.

Giá trị trả về của hàm **map()** luôn là một số nguyên.

Cú pháp **map(val,A1,A2,B1,B2);**

trong đó: **val** là giá trị cần chuyển đổi
 A1, A2 là giới hạn trên và dưới của thang đo hiện tại
 B1,B2 là giới hạn trên và dưới của thang đo cần chuyển tới

Ví dụ

//Chuyển đổi 37 độ C sang độ F

int C_deg = 37;

int F_deg = map(37,0,100,32,212);

➔ //F_deg = 98

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.4 Đọc giá trị analog

Bài tập 3: Viết chương trình đọc giá trị từ 8 biến trở

Arduino UNO chỉ có 6 ngõ vào Analog

→ Dùng IC multiplexer để đọc từng ngõ vào analog thông qua các chân điều khiển digital

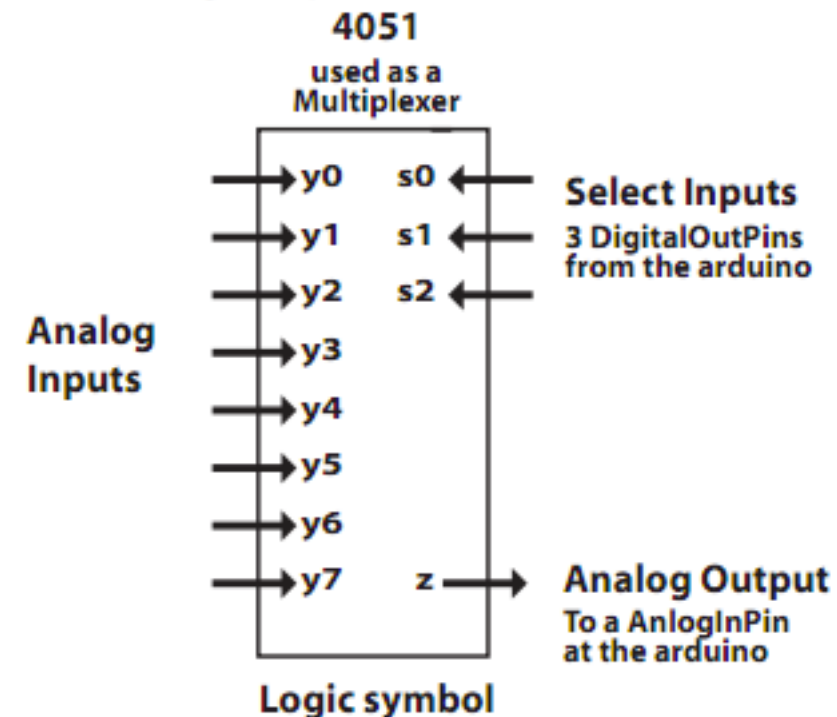
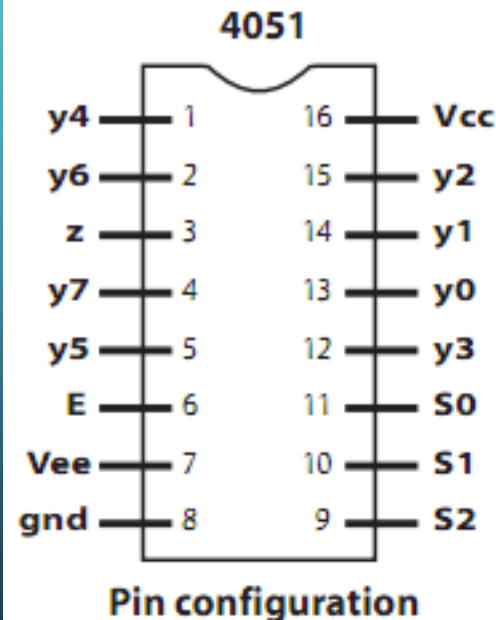
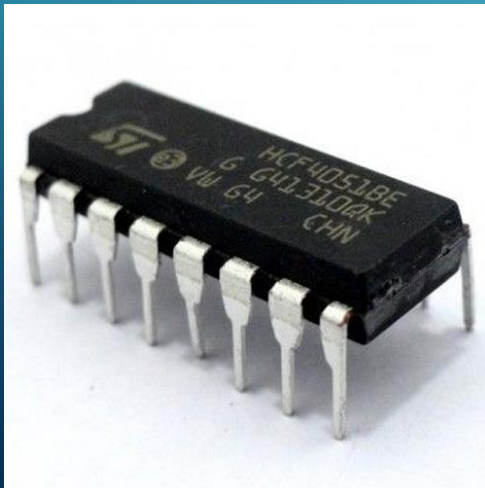
Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.4 Đọc giá trị analog

Bài tập 3: Viết chương trình đọc giá trị từ 8 biến trở

→ dùng IC Multiplexer 4051

IC **4051** là một bộ 8-kênh analog multiplexer/demultiplexer với 3 đầu vào địa chỉ



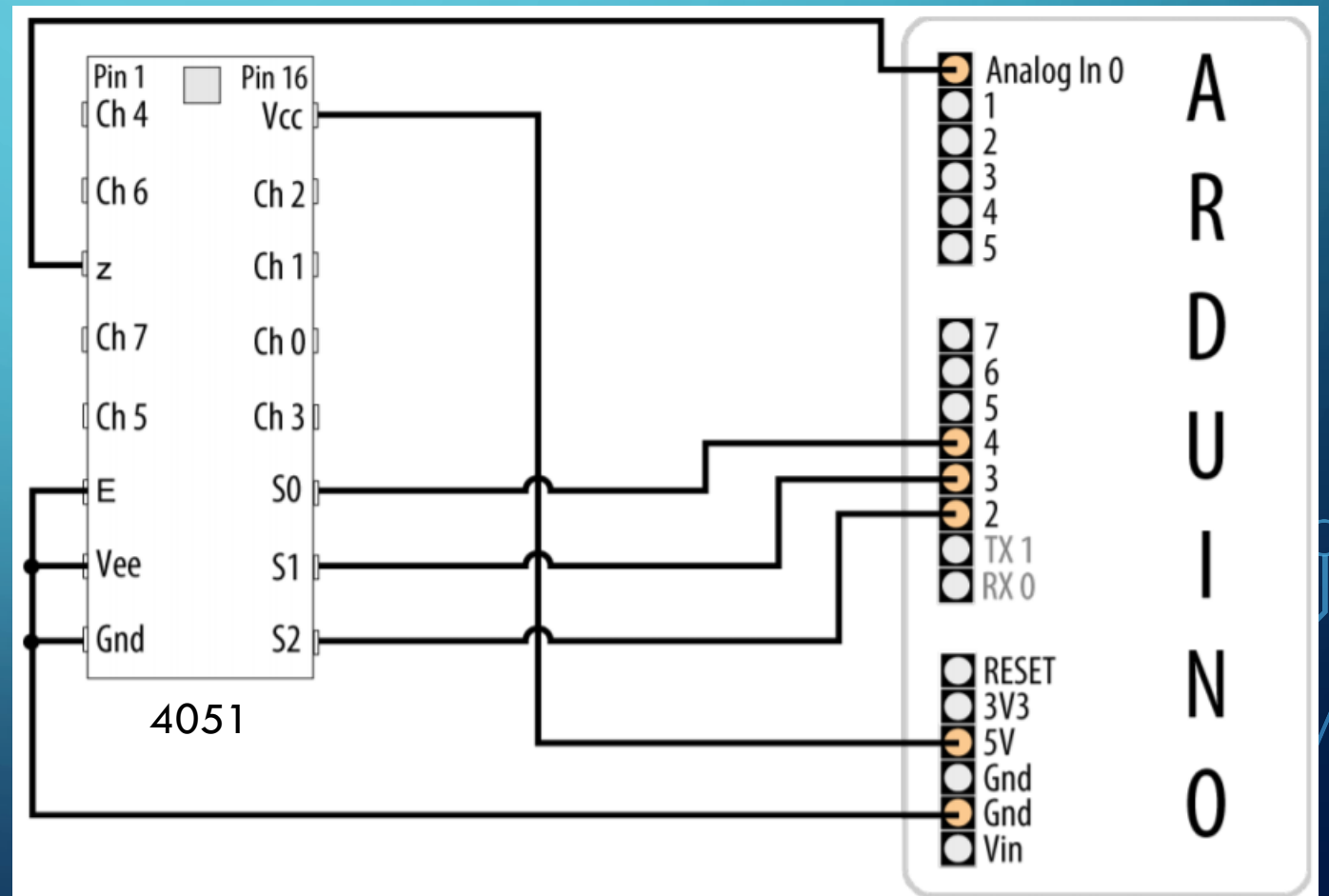
Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

3.4 Đọc giá trị analog

Bài tập 3: Viết chương trình đọc giá trị từ 8 biến trở

Sơ đồ mạch nguyên lý

Selector pins			Selected input
S2	S1	S0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

Bài tập 3: Viết chương trình đọc giá trị từ 8 biến trở

Chương trình

```
// array of pins used to select 1 of 8 inputs on multiplexer
const int select[] = {2,3,4};      // pins connected to the 4051 input select lines
const int analogPin = 0;           // the analog pin connected to multiplexer output
                                   // this function returns the analog value for the given channel
int getValue( int channel)
{
    // set the selector pins HIGH and LOW to match the binary value of channel
    for(int bit = 0; bit < 3; bit++)
    {
        int pin = select[bit];      // the pin wired to the multiplexer select bit

        int isBitSet = bitRead(channel, bit); // true if given bit set in channel
        digitalWrite(pin, isBitSet);
    }
    return analogRead(analogPin);
}
```

Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

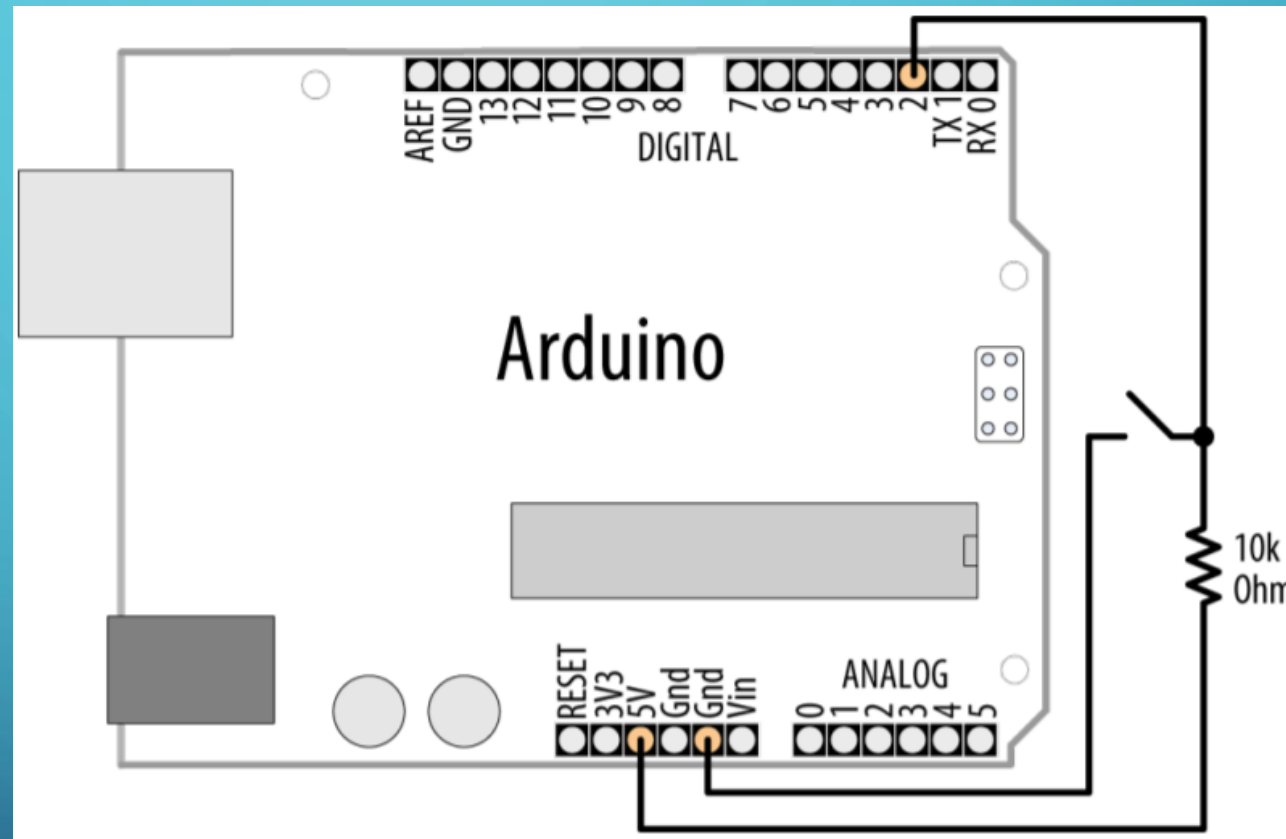
Bài tập 3: Viết chương trình đọc giá trị từ 8 biến trở

Chương trình

```
void setup()
{
    for(int bit = 0; bit < 3; bit++)
        pinMode(select[bit], OUTPUT);           // set the three select pins to output
    Serial.begin(9600);
}
void loop () {
    // print the values for each channel once per second
    for(int channel = 0; channel < 8; channel++)
    {
        int value = getValue(channel);
        Serial.print("Channel ");
        Serial.print(channel);
        Serial.print(" = ");
        Serial.println(value);
    }
    delay (1000);
}
```


BÀI TẬP CHƯƠNG 3

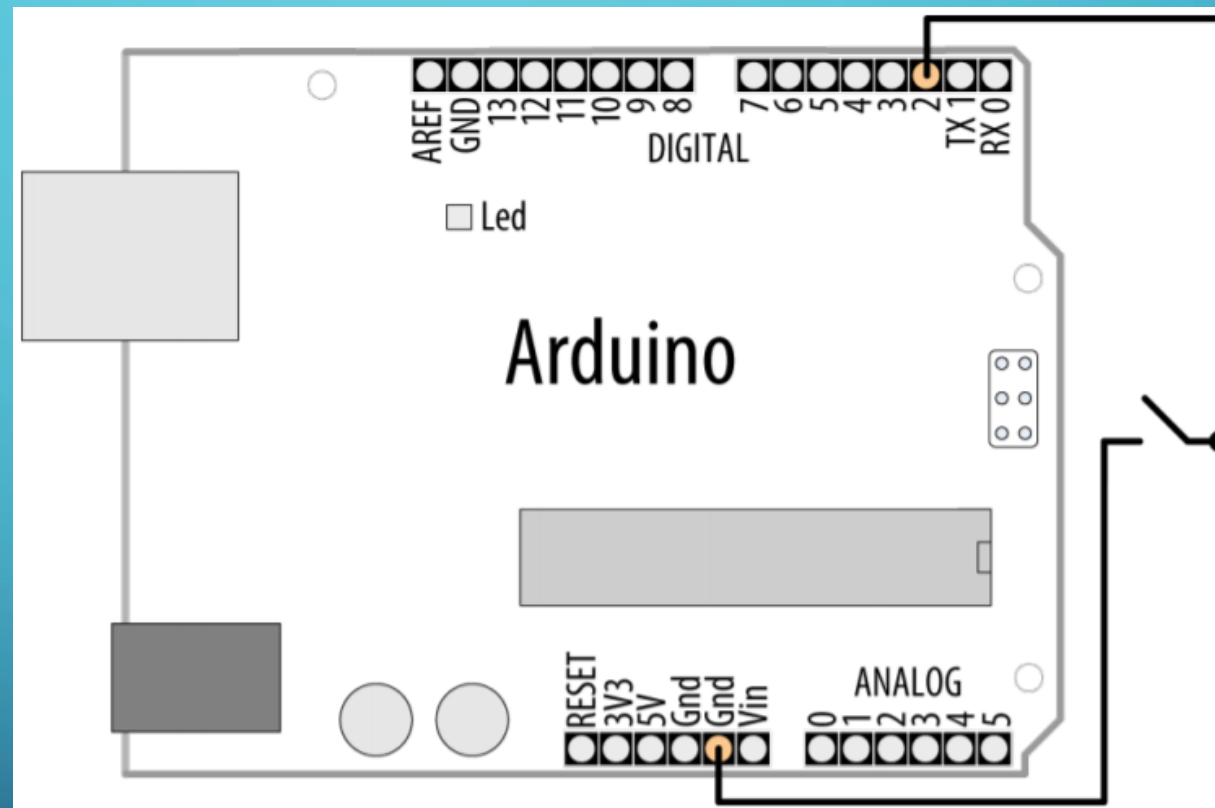
1. Cho sơ đồ mạch nguyên lý sau đây



Viết chương trình điều khiển sao cho khi nhấn nút thì đèn LED ở chân 13 sáng, khi nhả nút thì đèn LED tắt.

BÀI TẬP CHƯƠNG 3

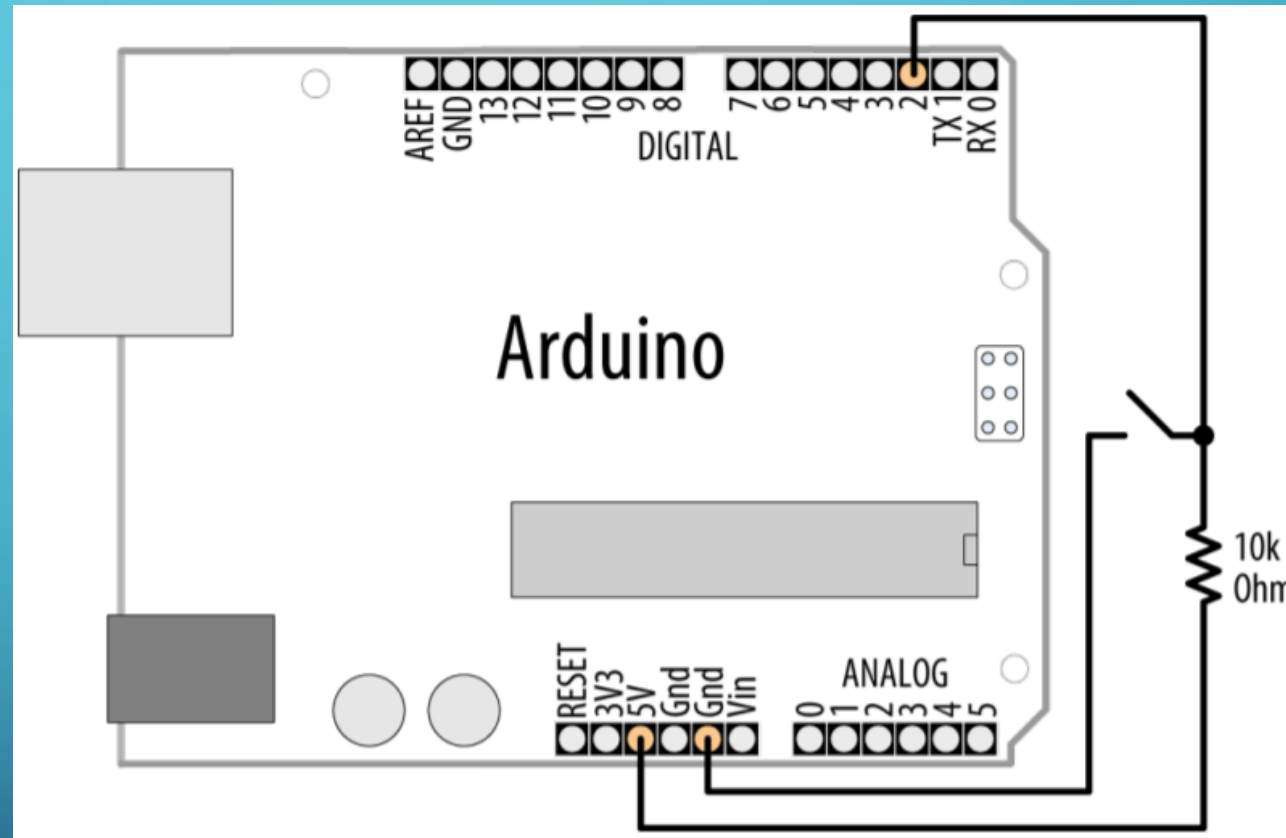
2. Cho sơ đồ mạch nguyên lý sau đây



Viết chương trình điều khiển sao cho khi nhấn nút thì đèn LED ở chân 13 sáng, khi nhả nút thì đèn LED tắt.

BÀI TẬP CHƯƠNG 3

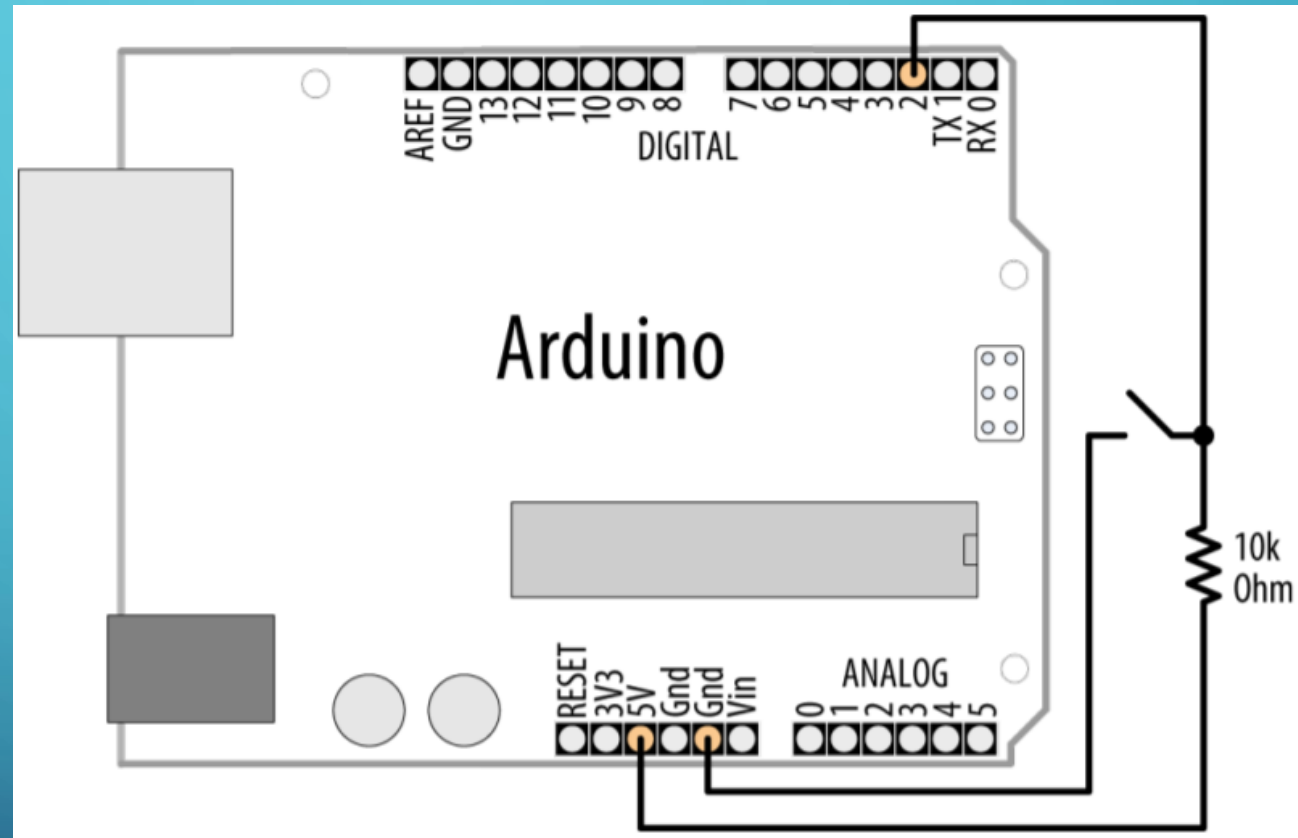
3. Cho sơ đồ mạch nguyên lý sau đây



Viết chương trình điều khiển sao cho khi nhấn nút thì đèn LED ở chân 13 sáng, khi nhả nút thì đèn LED tắt có sử dụng **chương trình con chống dội phím**

BÀI TẬP CHƯƠNG 3

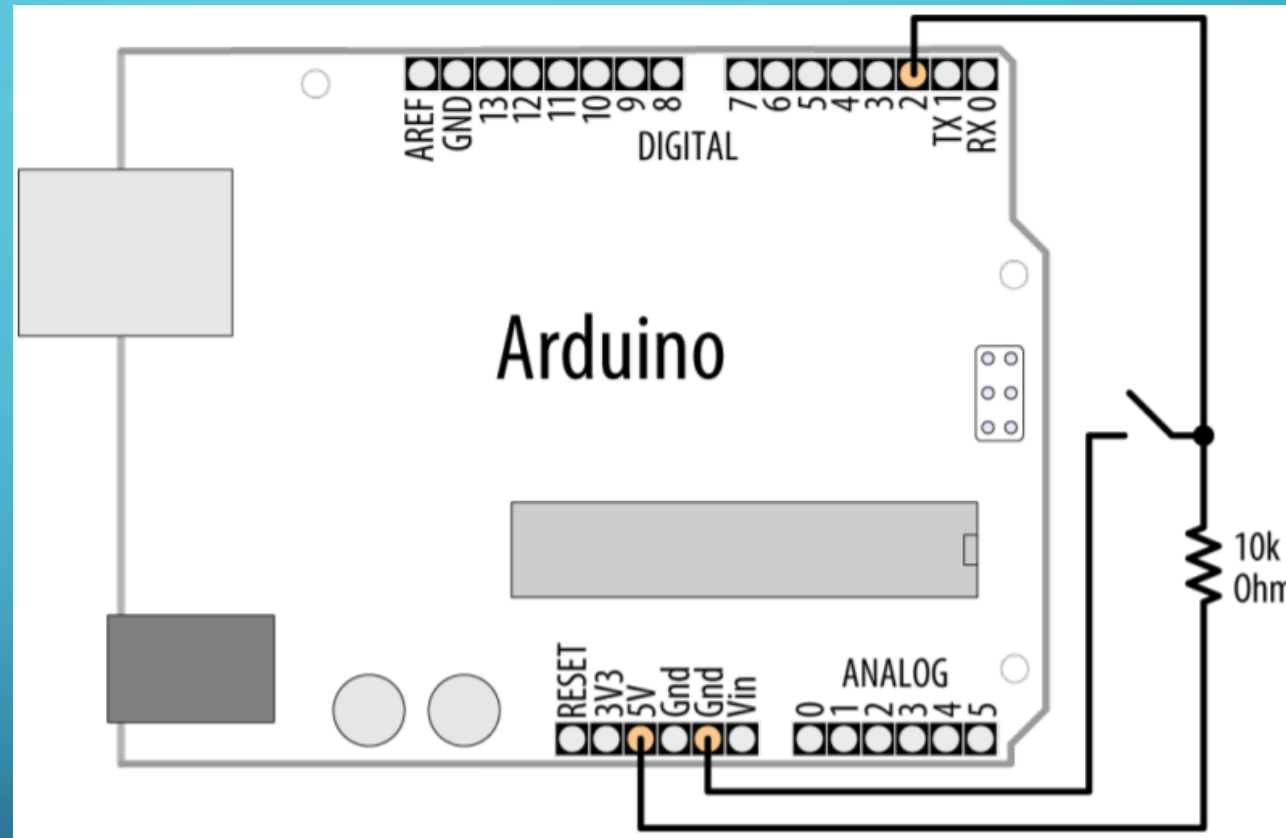
4. Cho sơ đồ mạch nguyên lý sau đây



Viết chương trình đếm số lần nhấn nút

BÀI TẬP CHƯƠNG 3

5. Cho sơ đồ mạch nguyên lý sau đây

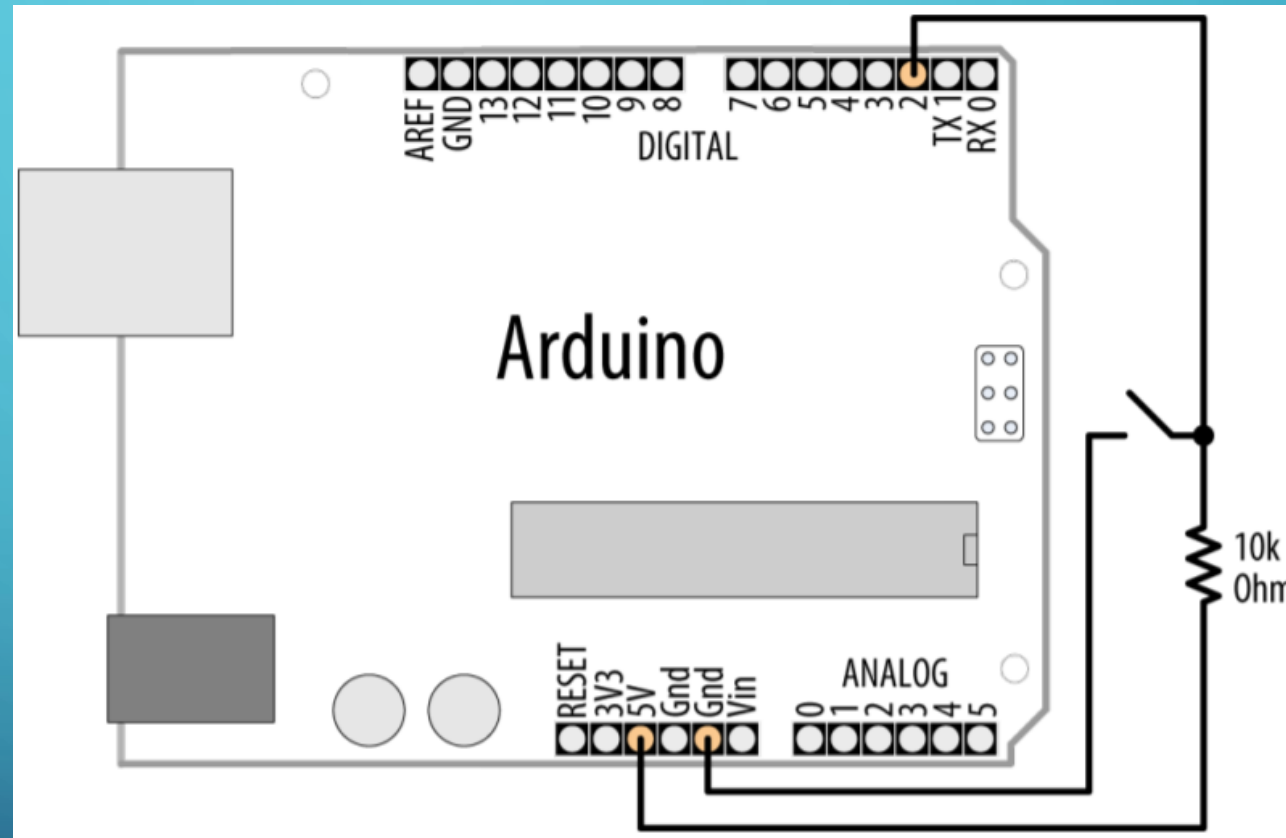


Viết chương trình đếm số lần nhấn nút.

Khi số lần nhấn nút bằng 5 đến 10 thì LED ở chân 13 sáng, còn lại thì LED tắt

BÀI TẬP CHƯƠNG 3

6. Cho sơ đồ mạch nguyên lý sau đây



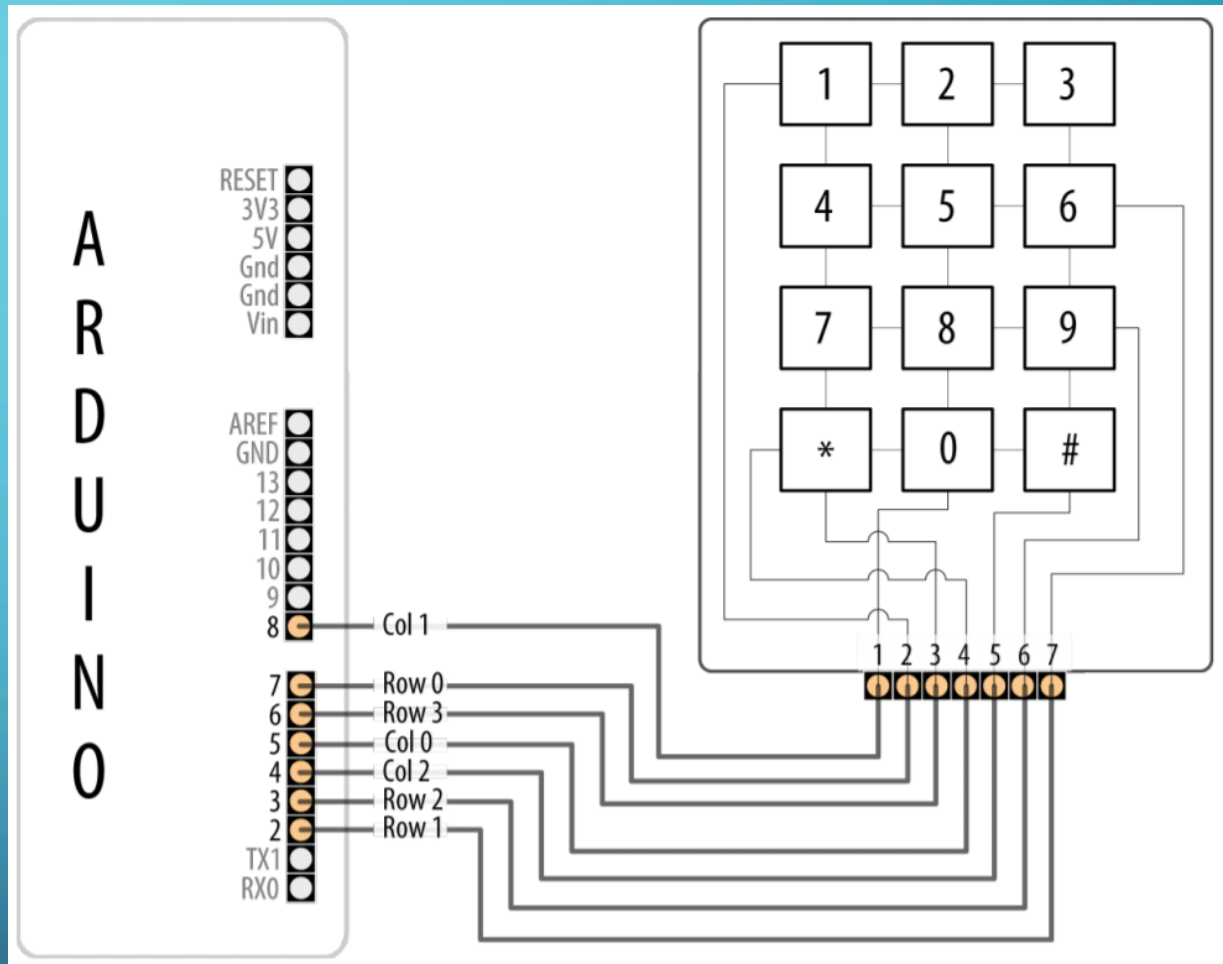
Viết chương trình đếm thời gian nút được nhấn.

Khi nút được nhấn giữ lớn hơn 1 giây thì LED ở chân 13 chớp tắt, còn lại thì LED tắt

BÀI TẬP CHƯƠNG 3

7. Cho sơ đồ mạch nguyên lý sau đây

Viết chương trình đọc bàn phím ma trận Keypad



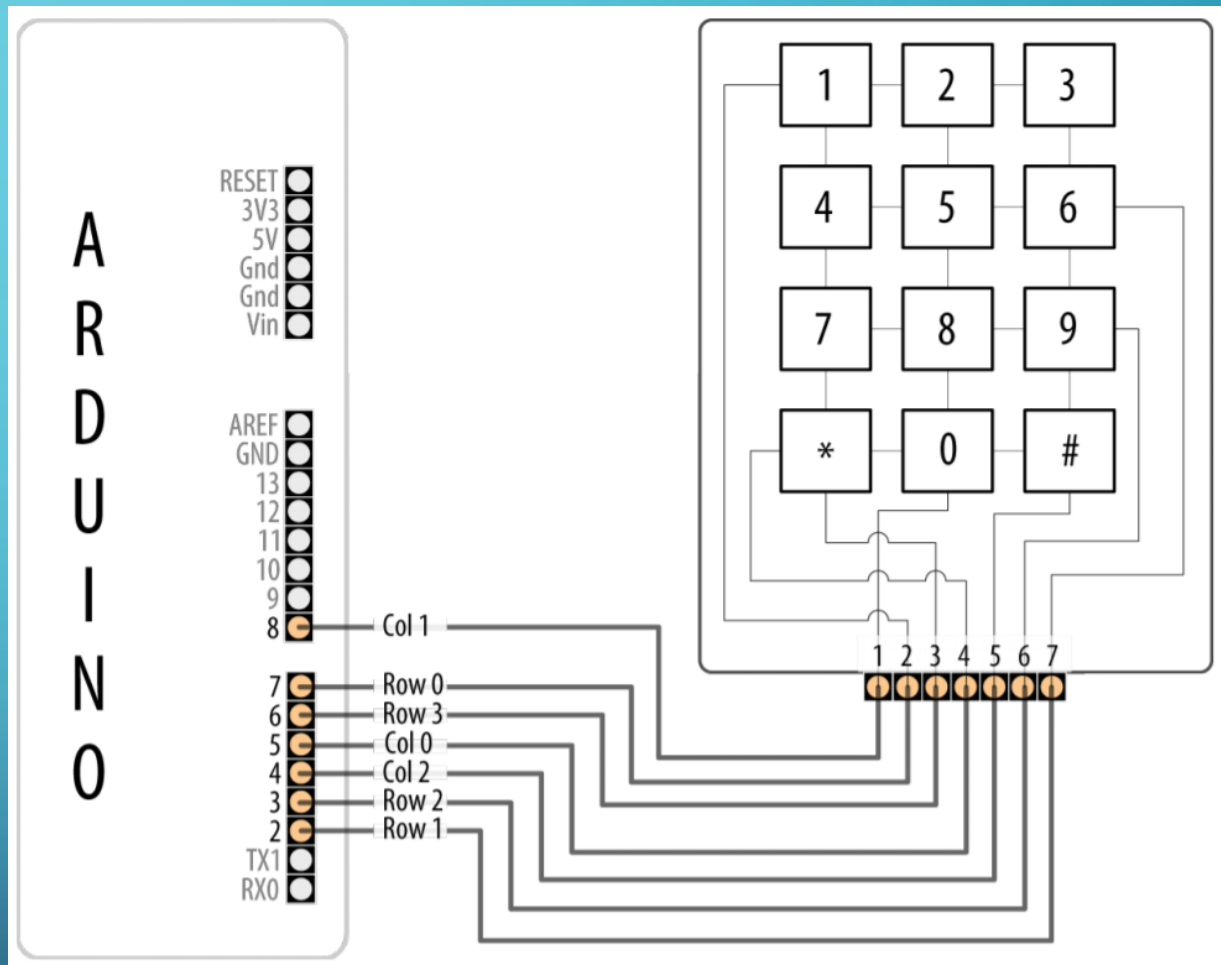
BÀI TẬP CHƯƠNG 3

8. Cho sơ đồ mạch nguyên lý sau đây

Viết chương trình khóa số: nếu tổ hợp 3 phím được nhấn trùng với tổ hợp 3 giá trị đã đặt trước thì LED ở chân 13 sẽ chớp tắt.

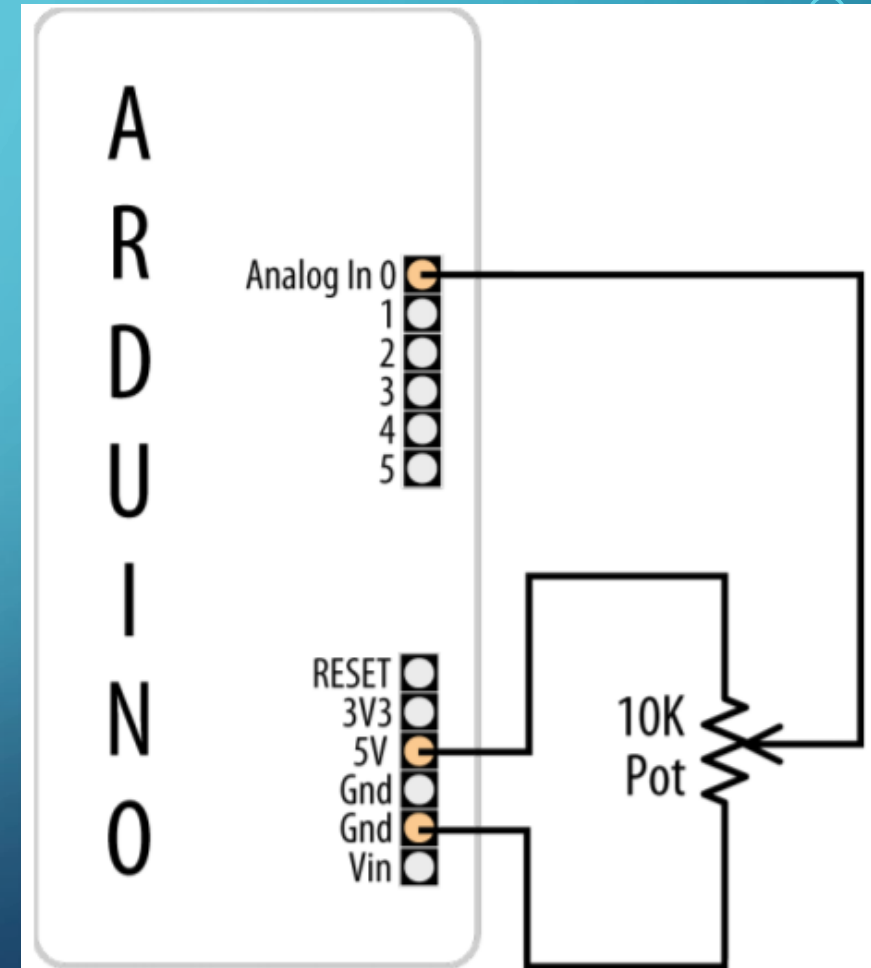
Nút # tương đương nút ENTER

Nút * tương đương nút CLEAR



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

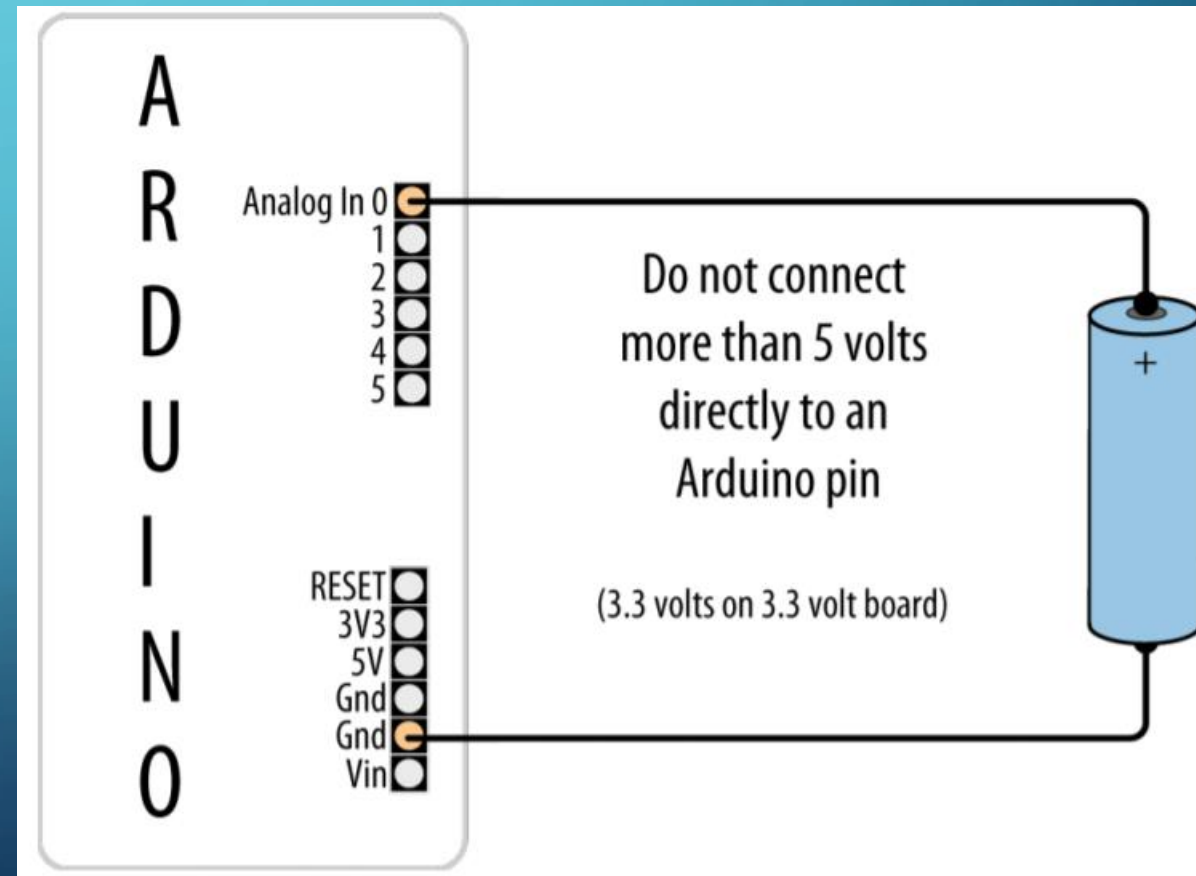
9. Cho sơ đồ nguyên lý như hình bên.
Viết chương trình điều khiển tốc độ nháy của đèn LED ở chân 13 sang bằng biến trở tính theo phần trăm (%)



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

10. Cho sơ đồ nguyên lý như hình bên.

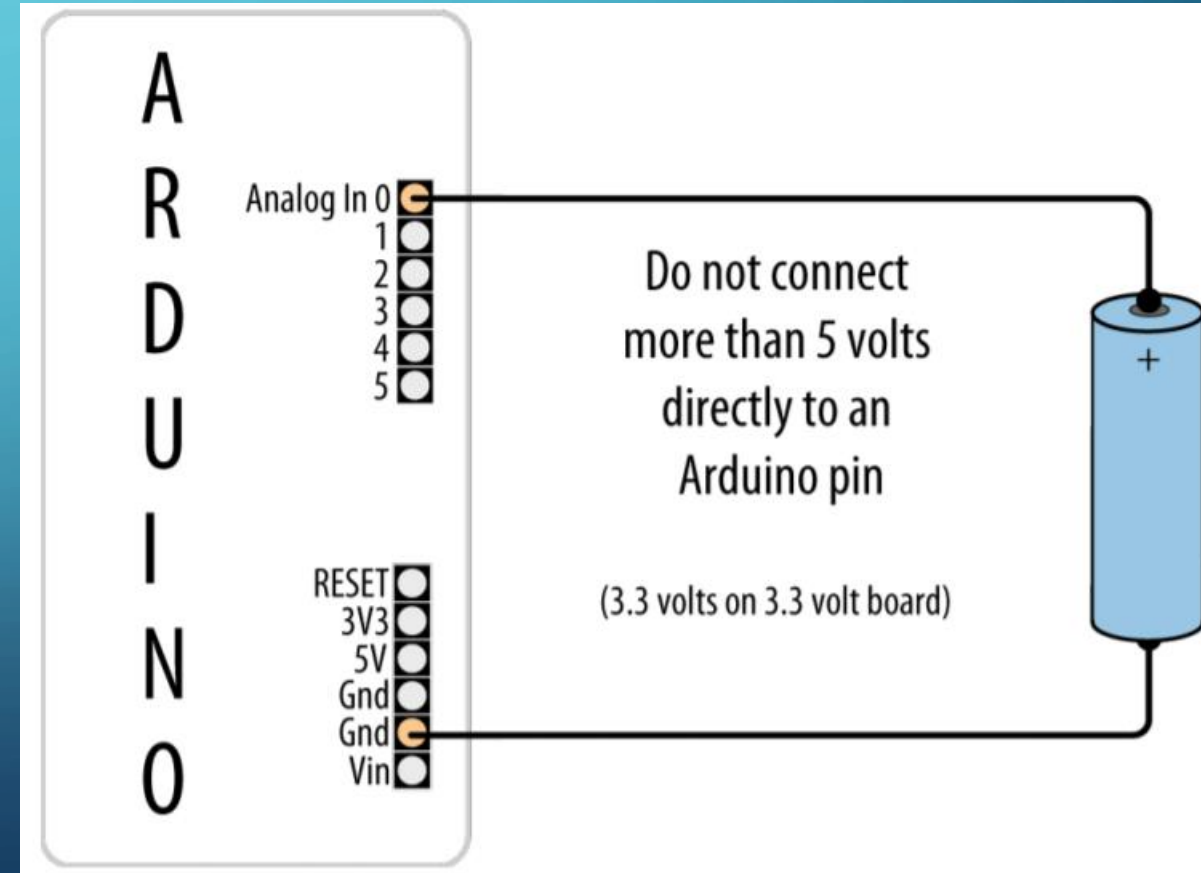
Viết chương trình đọc và hiển thị trên Serial Monitor giá trị điện áp của pin.



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

11. Cho sơ đồ nguyên lý như hình bên.

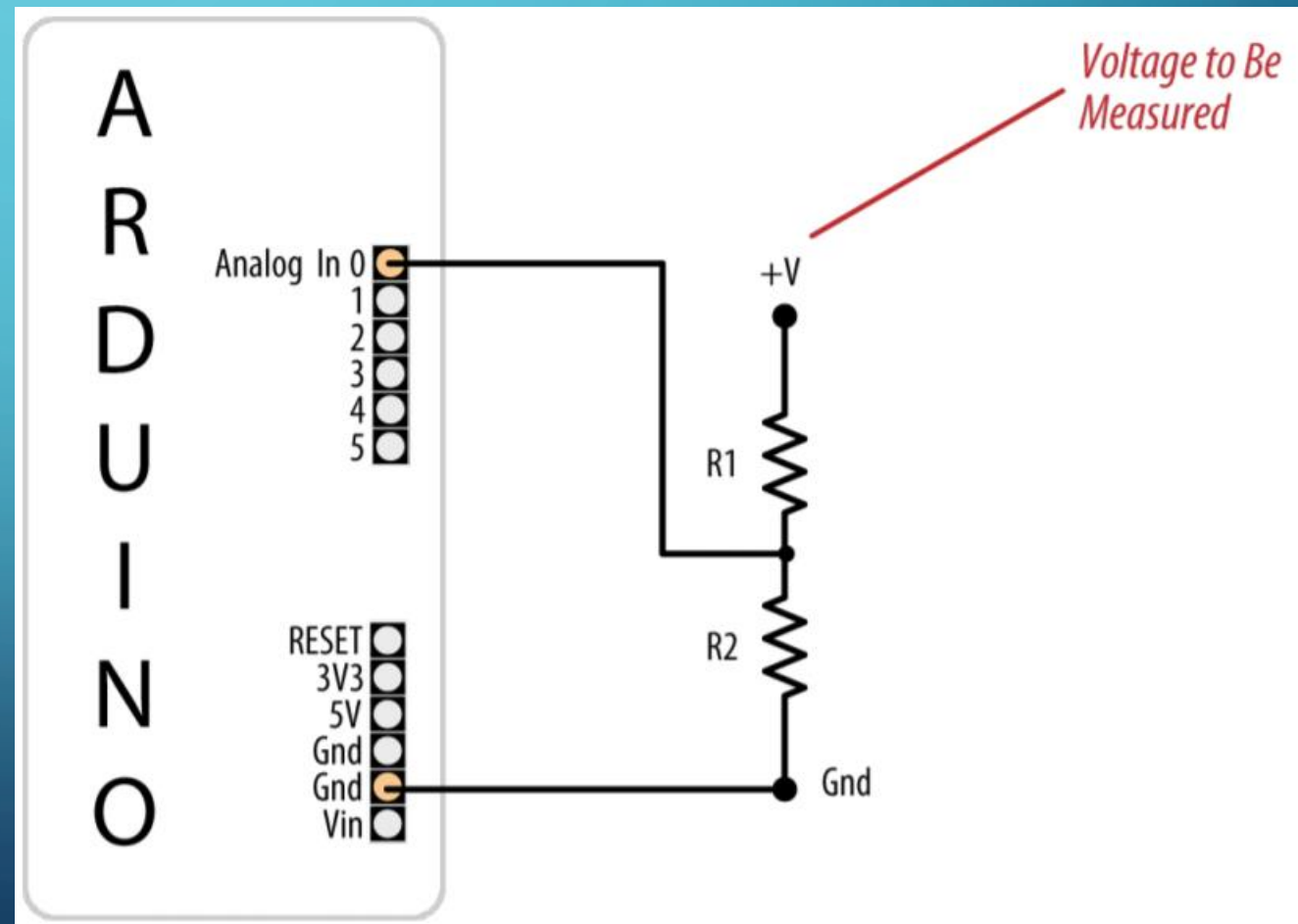
Viết chương trình đọc và hiển thị trên Serial Monitor giá trị điện áp pin đo được bằng đơn vị là miliVolt



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

12. Cho sơ đồ nguyên lý như hình bên.

Viết chương trình đọc và hiển thị trên Serial Monitor giá trị điện áp đo khi điện áp đưa vào đo lớn hơn 5V



Chương 3: XUẤT NHẬP DỮ LIỆU SỐ - TƯƠNG TỰ

12. Cho sơ đồ nguyên lý như hình bên.

Viết chương trình đọc và hiển thị trên Serial Monitor giá trị điện áp đo khi điện áp đưa vào đo lớn hơn 5V

Gợi ý: bảng giá trị điện trở sử dụng và mức chia

			Calculation	
Max voltage	R1	R2	$R2/(R1 + R2)$	value of resistorFactor
5	Short (+V connected to analog pin)	None (Gnd connected to Gnd)	None	1023
10	1K	1K	$1(1 + 1)$	511
15	2K	1K	$1(2 + 1)$	341
20	3K	1K	$1(3 + 1)$	255
30	4K (3.9K)	1K	$1(4 + 1)$	170