Third Lecture

Arduino Based Application

By:

Ayub Othman Abdulrahman

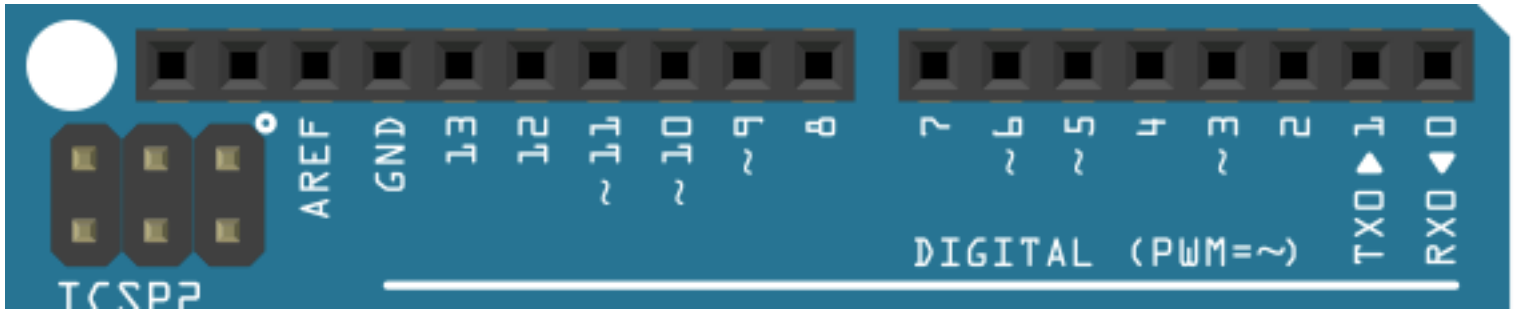
University Of Halabja

Introduction

- The Arduino microcontroller has a nearly limitless array of innovative applications for everything from robotics and lighting, to games and gardening! It's a fun way to automate everything, enabling you to control simple devices or manage complex Application.

Digital Pins

- As , it is clear in the bellow figure that we have RXD at 0 which is used for Serial receiving and then we have TXD at 1 used for Serial writing.
- So, these pins from 0 to 13 are all digital and after these digital Pins we have GND.
- Analog pins (A0 to A5) can be used as digital pin



pinMode()

- **pinMode()** : This function set the functionality of pin. Means either INPUT or OUTPUT
- INPUT use when you connect your Arduino to the sensor.
- OUTPUT use when you connect your Arduino to LED, motor etc.

digitalRead

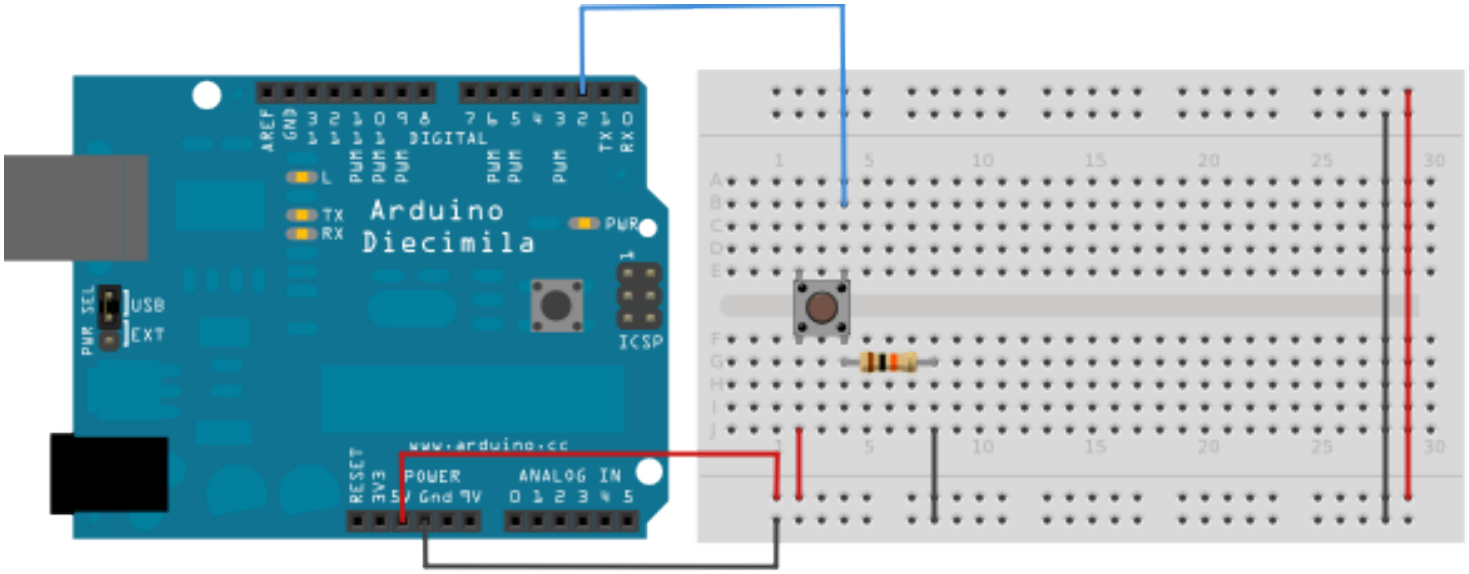
In total we have 14 digital Pins in Arduino UNO starting from 0 to 13. it will be explained them in detail and we will also have a look at How to use this digitalRead command in Arduino. So, let's get started with it:

Momentary Switch

- To use a toggle switch, the following Hardware are required.
  - Arduino Board
  - A momentary switch, or button.
  - 10k ohm resistor
  - hook-up wires
  - breadboard

Circuit of the Switch

- pin 2 is connected to one leg of the pushbutton.



How the Pushbutton is Worked

- Pushbuttons or switches connect two points in a circuit when you press them. When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground. and reads as LOW, or 0. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that the pin reads as HIGH, or 1.

Example Code for Pushbutton

```
1.  int pushButton = 2;
2.  void setup() {                  //the setup routine runs once
3.   Serial.begin(9600);       //initialize serial communication at 9600 bps:
4.  pinMode(pushButton, INPUT);  //make the pushbutton's pin an input:
5.  }
6.  void loop() {                 //the loop routine runs forever:
7.  int buttonState = digitalRead(pushButton);  //read the input pin:
8.  Serial.println(buttonState);     //print out the state of the button:
9.    delay(10);                 //delay in between reads for stability
10. }
```

digitalWrite

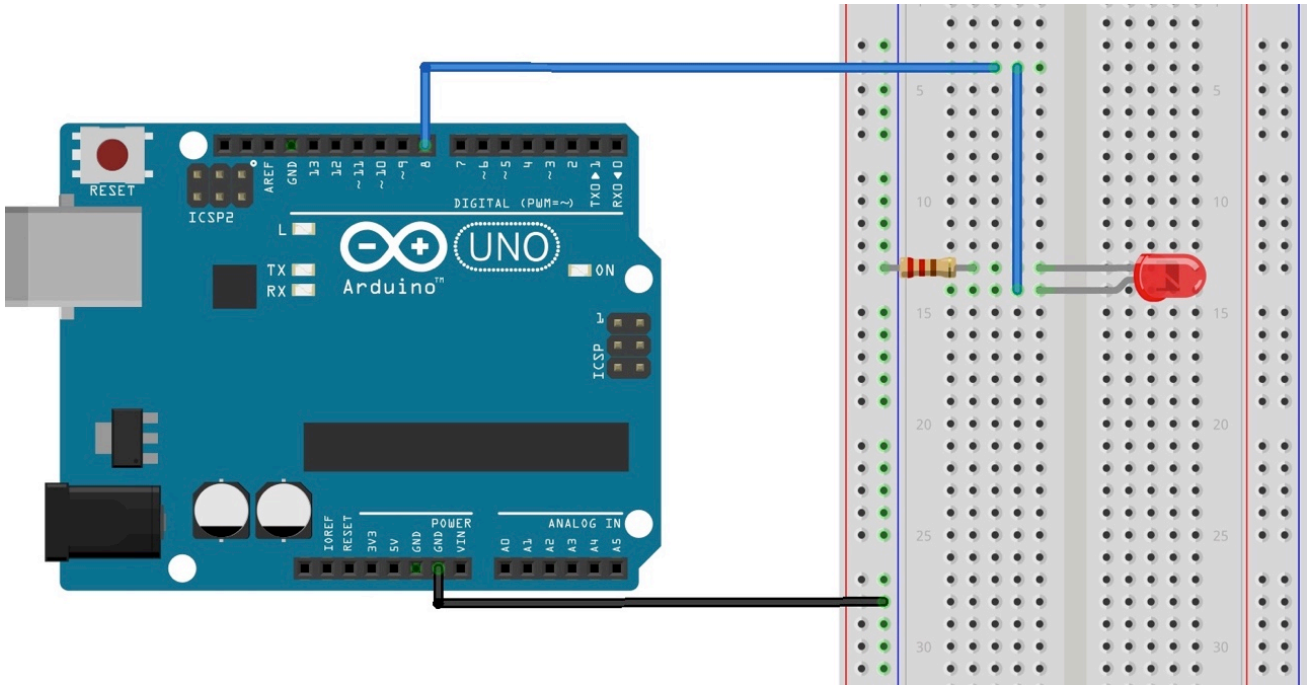
- Pins configured as OUTPUT with pinMode() are said to be in a low-impedance state. This means that they can provide a substantial amount of current to other circuits. Atmega pins can source (provide positive current) up to 40 mA (milliamps) of current to other devices/circuits. This is enough current to brightly light up an LED (don't forget the series resistor)

Blinking LED using delay

- In this experiment, we will be switching an LED on and off using a digital output.
- 1 x LED
- 1 x 220 ohm resistor
- 1 x Arduino UNO
- 1 x breadboard
- 2 x jumper Wires

Circuit of the LED

- Pin 8 is connected to one leg of the LED.



How the LED is Worked

- The circuit in the figure shows how to connect the LED and the 220 ohm resistor to the Arduino. As shown, the LED connects to digital I/O pin 8 of the Arduino through the resistor. The resistor controls the current through the LED.

Example Code for LED

```
1.  const int led  =  8;     //use digital I/O pin 8
2.  void setup() {
3.  pinMode(led,OUTPUT);   //set pin 8 to be an output output
4.  }
5.  void loop() {
6.    delay(1000);          //delay 1000 milliseconds
7.    digitalWrite(led,HIGH);   //set pin 8 HIGH, turning on LED
8.    delay(1000);          //delay 1000 milliseconds
9.    digitalWrite(led,LOW);    //set pin 8 LOW, turning off LED
10. }
```

Using Pushbutton to on and off the LED

- Those previous circuits can be combined and used to turn on and off the light emitting diode (LED) controlling by toggle switch.
- The switch should be initialize as input and the LED as output

Code Example of Toggle Switch and LED

```
1.  int pushButton = 2;
2.  int ledPin = 8;
3.  int x=0;
4.  void setup() {
5.   Serial.begin(9600);
6.   pinMode(pushButton, INPUT);
7.   pinMode(ledPin, OUTPUT);  }
8.  void loop() {
9.   while (x<2)  {
10.   if (digitalRead(pushButton)== HIGH)
11.   x=x+1;
12.   delay(200);
13.  switch (x){
14.  case 0:
15.  digitalWrite (ledPin, LOW);
16.  Serial.println("B");
17.  break;
18.  case 1:
19.  digitalWrite (ledPin, HIGH);
20.  Serial.println("A");
21.  break;}   }
22.  x=0;
23. }
```

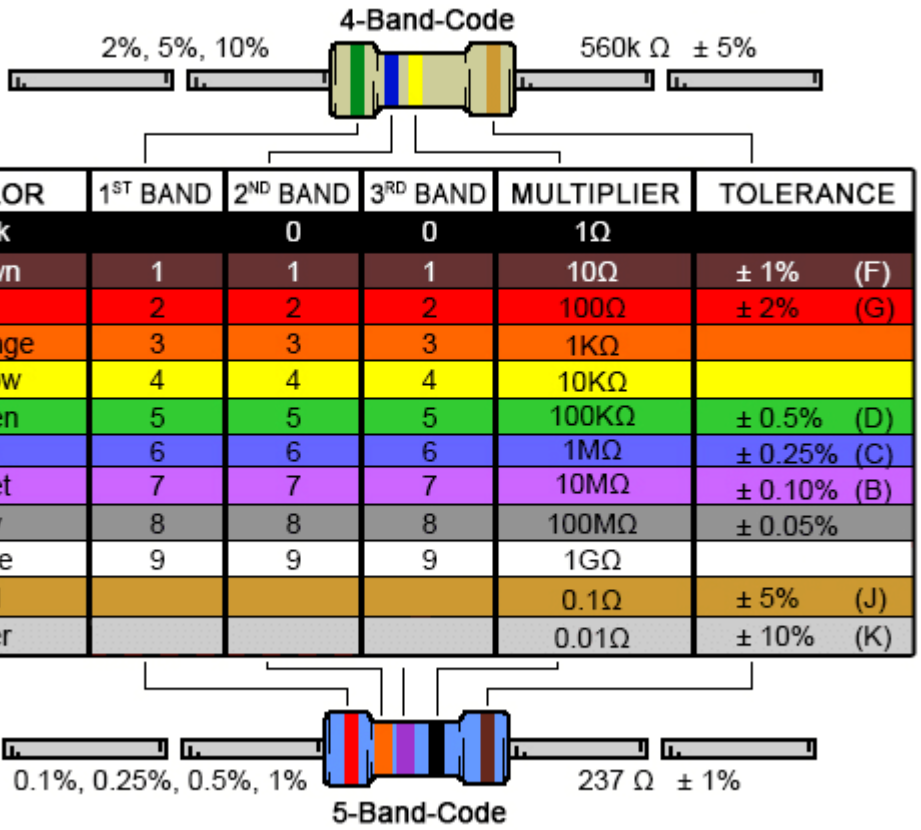
Resistor

- There are many different types of Resistor available which can be used in both electrical and electronic circuits to control the flow of current or to produce a voltage drop in many different ways. Resistors are available in a range of different resistance values from fractions of an Ohm (Ω) to millions of Ohms.

Resistors Colour Code.

- The resistance value, tolerance, and wattage rating are generally printed onto the body of the resistor as numbers or letters when the resistors body is big enough to read the print.
- But small resistors use coloured painted bands to indicate both their resistive value and their tolerance with the physical size of the resistor. This generally known as a Resistors Colour Code.

Resistors Colour Code.



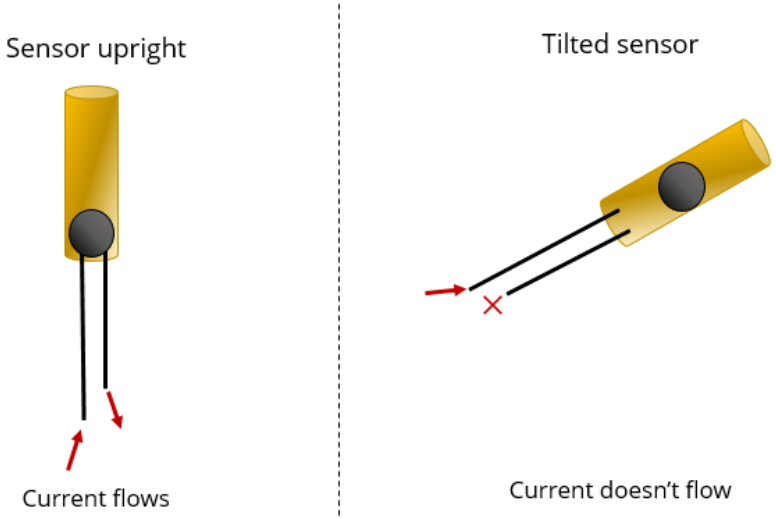
COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE
Black		0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8	100MΩ	± 0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)

Roll Ball Switch (Tilt Sensor)

- The tilt sensor is a component that can detect the tilting of an object. However it is only the equivalent to a pushbutton activated through a different physical mechanism. This type of sensor is the environmental-friendly version of a mercury-switch.

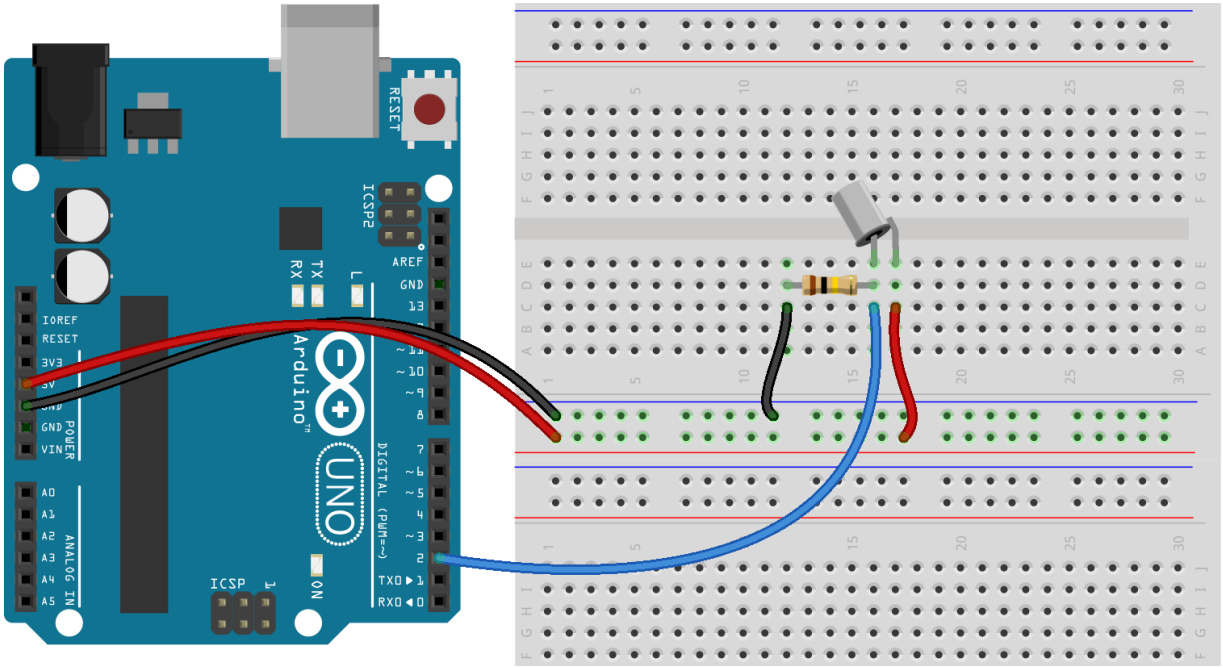
Roll Ball Switch (Tilt Sensor)

- It contains a metallic ball inside that will commute the two pins of the device from on to off and vice versa if the sensor reaches a certain angle.



Tilt Sensor (Circuit Diagram)

- The following circuit diagram is sample diagram of roll ball switch



Tilt Sensor (Code Example)

```
1.  const int tilt = 4;            // the number of the tilt pin
2.  const int ledPin =  13;                  // the number of the LED pin
3.  int buttonState = 0;        // variable for reading the pushbutton status
4.  void setup() {
5.  pinMode(ledPin, OUTPUT); // initialize the LED pin as an output:
6.  pinMode(tilt, INPUT);      // initialize the tilt pin as an input:
7.  }
8.  void loop() {
9.  buttonState = digitalRead(tilt); // read the state of the pushbutton value:
10. if (buttonState == HIGH) {
11. digitalWrite(ledPin, HIGH); // turn LED on:
12.   } else {
13. digitalWrite(ledPin, LOW); // turn LED off:
14.   }}
```

PIR Motion Sensor

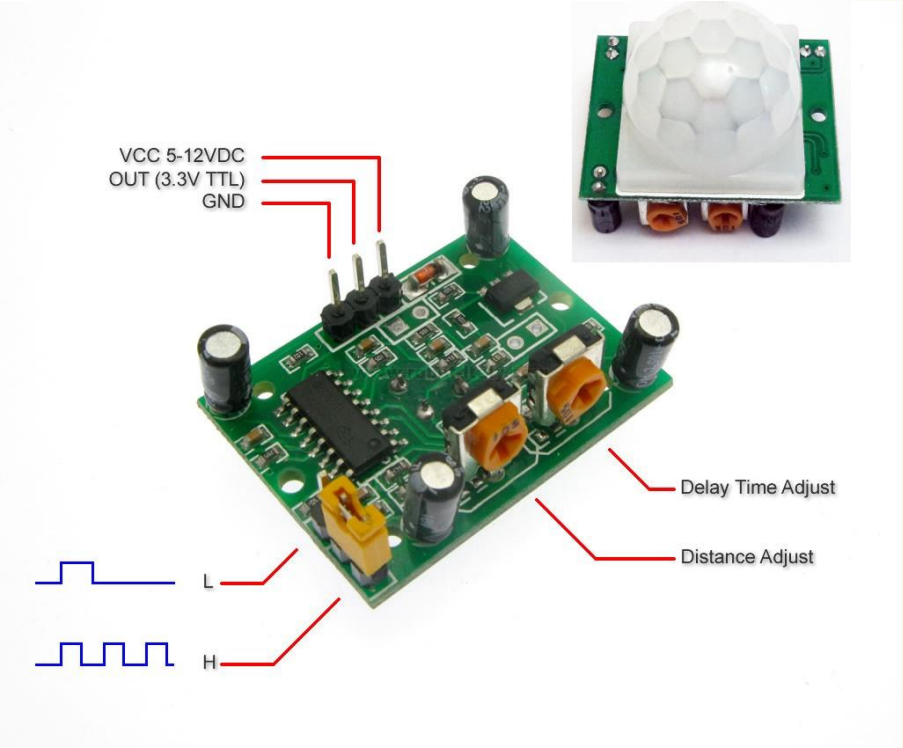
- So, it can detect motion based on changes in infrared light in the environment. It is ideal to detect if a human has moved in or out of the sensor range.

PIR Motion Sensor

- The PIR motion sensor is ideal to detect movement. PIR stand for “Passive Infrared”. Basically, the PIR motion sensor measures infrared light from objects in its field of view.

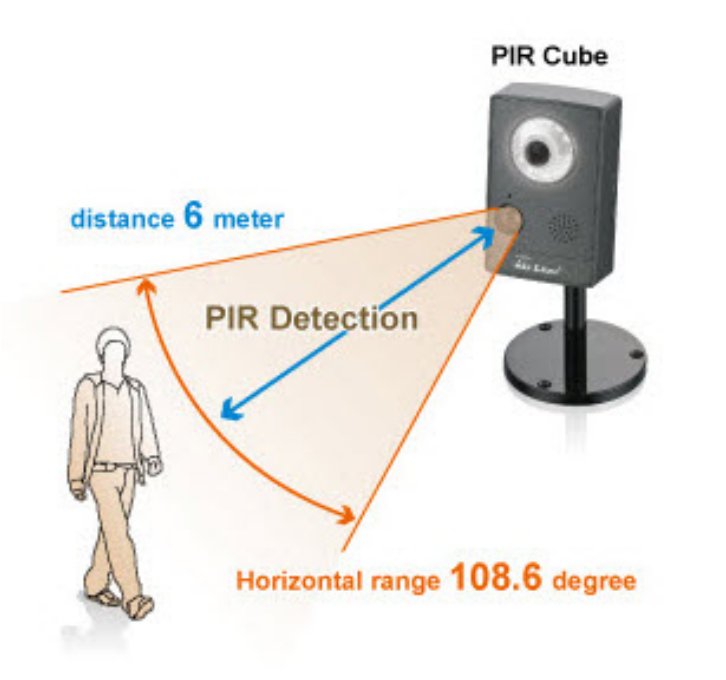
PIR Motion Sensor

- The sensor has two built-in potentiometers to adjust the delay time (the potentiometer at the left) and the sensitivity (the potentiometer at the right).

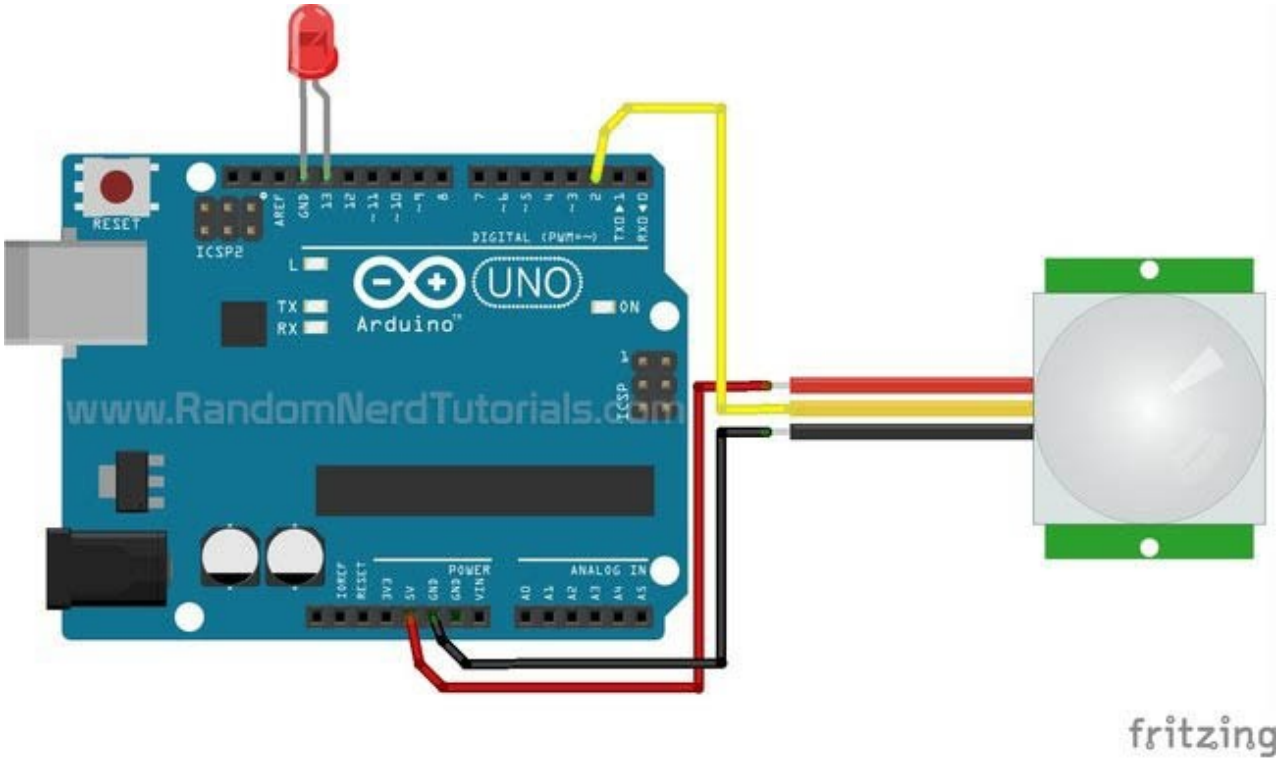


Sensitively Range

- The sensor outputs a 5V signal as soon as it detects the presence of a person. It offers a tentative range of detection of about 6–7m and is highly sensitive.



Circuit Diagram



Code Example

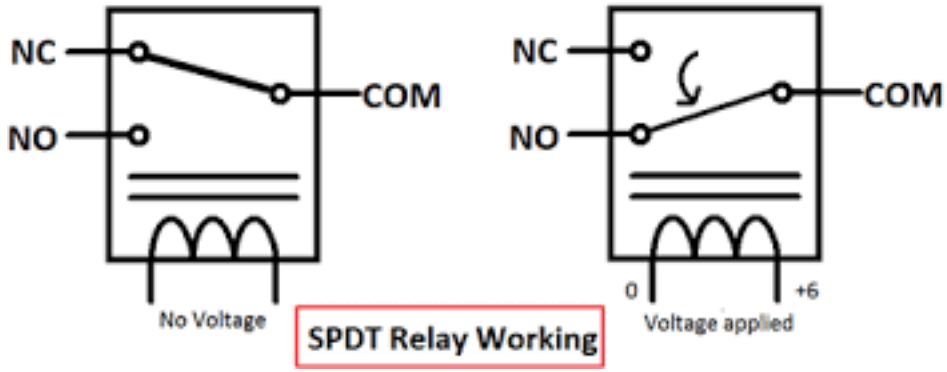
```
1.  int led = 13;              // the pin that the LED is atteched to
2.  int sensor = 2;            // the pin that the sensor is atteched to
3.  int val = 0;               // variable to store the sensor status (value)

4.  void setup() {
5.    pinMode(led, OUTPUT);     // initalize LED as an output
6.    pinMode(sensor, INPUT);   // initialize sensor as an input
7.    Serial.begin(9600); }     // initialize serial

8.  void loop(){
9.    val = digitalRead(sensor);   // read sensor value
10.   if (val == HIGH) {           // check if the sensor is HIGH
11.     digitalWrite(led, HIGH);   // turn LED ON
12.     delay(100);                // delay 100 milliseconds
13.     Serial.println("Motion detected!");
14.     } else {
15.   digitalWrite(led, LOW); // turn LED OFF
16.     delay(200);            // delay 200 milliseconds
17.     Serial.println("Motion stopped!"); }}
```

Relay

- The new KEYES 5V Relay Module is perfectly made for Arduino application. It has three pins, the VCC, GND and Signal. It can act as switch if the circuit and the load circuit have different supply voltage. It is commonly use if the load circuit is AC.

How The 5V Relay Works

- The 05VDC- relay has three high voltage terminals (NC, C, and NO) which connect to the device you want to control. The other side has three low voltage pins (Ground, Vcc, and Signal) which connect to the Arduino.

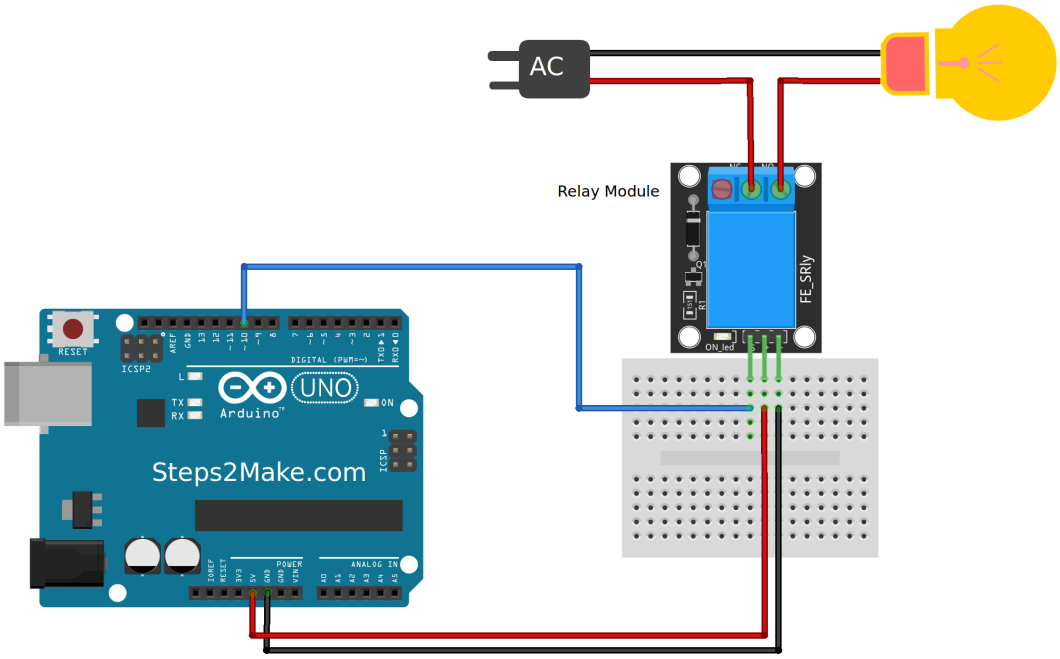


How The 5V Relay Works

- NC: Normally closed 120-240V terminal
- NO: Normally open 120-240V terminal
- C: Common terminal
- Ground: Connects to the ground pin on the Arduino
- 5V Vcc: Connects the Arduino's 5V pin
- Signal: Carries the trigger signal from the Arduino that activates the relay

Circuit of the LED

- Pin 10 is connected to signal pins of the Arduino.



Relay (Code_Example)

```
1.  int RELAY1= 10;
2.  void setup() {
3.  Serial.begin(9600);
4.  pinMode(RELAY1, OUTPUT);  }
5.    void loop() {
6.     digitalWrite(RELAY1,0);          // Turns ON Relays 1
7.     Serial.println("Light ON");
8.     delay(2000);                              // Wait 2 seconds
9.     digitalWrite(RELAY1,1);          // Turns Relay Off
10.    Serial.println("Light OFF");
11.    delay(2000);
12.  }
```

End

Any Questions?