

Microcontroller Experiments using Arduino or MSP430:

- a. Button
- b. IR emitter

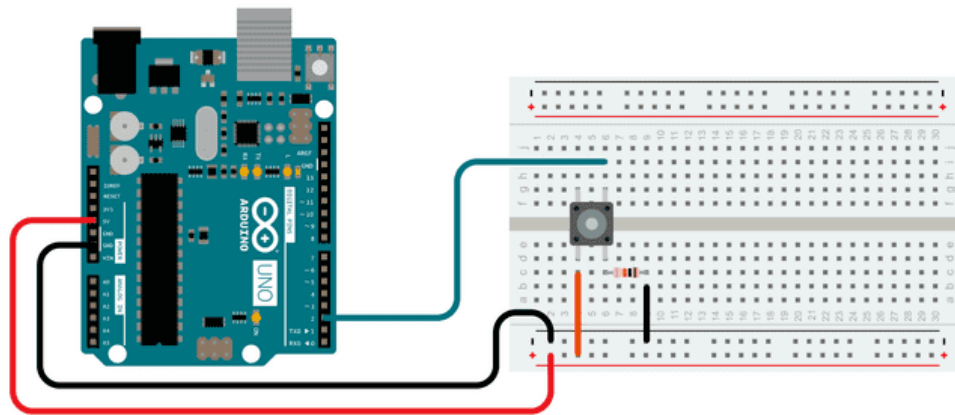
a. Button

Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Components Required:

- Arduino Board
- Momentary button or Switch
- 10K ohm resistor
- hook-up wires
- breadboard

Circuit Diagram:



Connect three wires to the board.

- The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground.
- The third wire goes from digital pin 2 to one leg of the pushbutton. That same leg of the button connects through a pull-down resistor (here 10K ohm) to ground.
- The other leg of the button connects to the 5 volt supply.

When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and we read a LOW. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that we read a HIGH.

Program:

```
/*
Button
Turns on and off a light emitting diode(LED) connected to digital pin 13,
when pressing a pushbutton attached to pin 2.
The circuit:
- LED attached from pin 13 to ground through 220 ohm resistor
- pushbutton attached to pin 2 from +5V
- 10K resistor attached to pin 2 from ground
- Note: on most Arduinos there is already an LED on the board
  attached to pin 13.*/

// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13;  // the number of the LED pin
```

```
// variables will change:
int buttonState = 0; // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

b. IR emitter

The Arduino UNO reads the state of three buttons. When a button is pressed, the Arduino uses the IR remote library to transmit a unique hexadecimal code encoded as an IR signal through the IR LED. Different codes can be assigned to each button press, allowing for communication of specific instructions to an IR receiver.

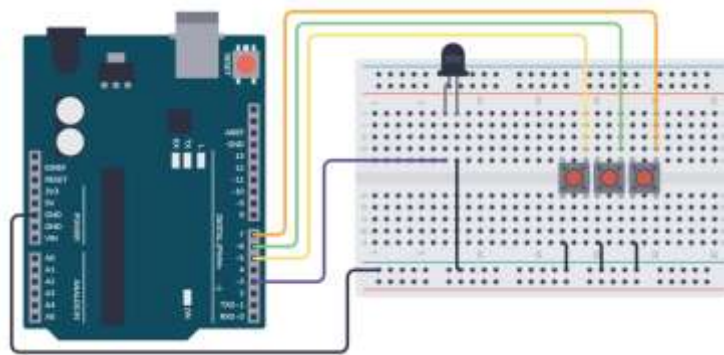
Components Required:

- Arduino
- IR LED (Light-Emitting Diode)
- 3 x Pushbuttons
- Breadboard

Building the Circuit

Connect each push-button switch to its respective pin (5, 6, or 7) on the Arduino board using jumper wires. Connect one side of each switch to the Arduino digital input pins and the other side to ground.

Next, connect the anode of the IR LED to digital pin 3 of the Arduino and the cathode to the ground. An IR LED is connected similarly to a regular LED, but it operates with invisible infrared light.



Program

```
#include <IRremote.h>
```

```
IRsend irsend;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(5, INPUT_PULLUP); // SW1 connected to pin 2  
  pinMode(6, INPUT_PULLUP); // SW2 connected to pin 3  
  pinMode(7, INPUT_PULLUP); // SW3 connected to pin 4  
  // The IR LED is connected to pin 3 (PWM ~) on the Arduino  
}
```

```
void loop() {  
  
  if (digitalRead(5) == LOW) { // When SW1 is pressed  
    irsend.sendNEC(0x34895725, 32); // Replace with your own unique code  
    Serial.println("Code sent!");  
    delay(30);  
  }  
  
  else if (digitalRead(6) == LOW) { // When SW2 is pressed  
    irsend.sendNEC(0x56874159, 32); // Replace with your own unique code
```

```
    Serial.println("Code sent!");
    delay(30);
}

else if (digitalRead(7) == LOW) { // When SW3 is pressed
    irsend.sendNEC(0x15467823, 32); // Replace with your own unique code
    Serial.println("Code sent!");
    delay(30);
}

else {
    Serial.println("Nothing to send");
    delay(30);
}

delay(100);
}
```