



Andhra Pradesh State Skill Development Corporation



INTERNET OF THINGS (IoT)

LED BLINKING & BUTTON INTERFACING TO THE ARDUINO



1LED Blinking

Aim: To Blink a led using Arduino.

Software: Arduino IDE

Components Required:

1. System -1
2. Arduino Uno Board -1
3. Arduino dumping cable -1
4. LED -1
5. Resistor 330 ohms-1
6. Breadboard-1
7. Connecting Wires -Required

Theory: LED: A light-emitting diode (LED) is a semiconductor device that emits visible light when an electric current passes through it. The light is not particularly bright, but in most LEDs it is monochromatic, occurring at a single wavelength. The output from an LED can range from red (at a wavelength of approximately 700 nanometers) to blue-violet (about 400 nanometers). Some LEDs emit infrared (IR) energy (830 nanometers or longer); such a device is known as an infrared-emitting diode (IRED).

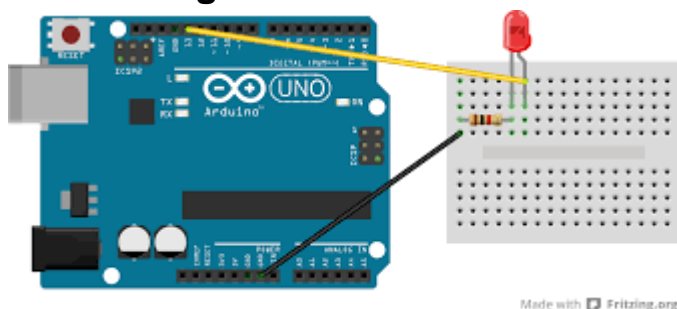
Advantages of LED's.

- Low power requirement: Most types can be operated with battery power supplies.
- High efficiency: Most of the power supplied to an LED or IRED is converted into radiation in the desired form, with minimal heat production.
- Long life: When properly installed, an LED or IRED can function for decades.

Applications:

- Indicator lights: These can be two-state (i.e., on/off), bar-graph, or alphabetic-numeric readouts.
- LCD panel backlighting: Specialized white LEDs are used in flat-panel computer displays.
- Fiber optic data transmission: Ease of modulation allows wide communications bandwidth with minimal noise, resulting in high speed and accuracy.
- Remote control: Most home-entertainment "remotes" use IREDs to transmit data to the main unit.
- Optoisolator: Stages in an electronic system can be connected together without unwanted interaction.

Circuit Diagram:-



Procedure:

1. Open Arduino IDE.
2. Write the code in the text editor.
3. Save the sketch with .ino extension.
4. Connect the hardware circuit and Connect your Arduino Board to the USB port of your computer.
5. Select the serial device of the Arduino board from the Tools | Serial Port menu.
6. Compile the file by clicking on the verify button.
7. If successful, the message "Done Compiling." will appear in the status bar.
8. If there is any errors it will show them in the Transcript window, rectify those errors and compile it again.
9. Push the reset button on the board then click the Upload button in the IDE. Wait a few seconds. If successful, the message "Done uploading." will appear in the status bar.

Code:

```
void setup( )
{
  pinMode(13,OUTPUT);
}
void loop( )
{
  digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(13,LOW);
  delay(1000);
}
```

Result: LED Blink continuously.



BUTTON INTERFACING WITH ARDUINO

AIM: To interface the button with Arduino

Software: Arduino IDE

Components Required:

1. System
2. Arduino Uno board-1
3. Arduino dumping cable-1
4. button
5. 10K ohm resistor
6. Bread board.
7. LED-1
8. connecting wires

THEORY:

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.

You can also wire this circuit the opposite way, with a pullup resistor keeping the input HIGH, and going LOW when the button is pressed. If so, the behavior of the sketch will be reversed, with the LED normally on and turning off when you press the button.

If you disconnect the digital I/O pin from everything, the LED may blink erratically. This is because the input is "floating" - that is, it will randomly return either HIGH or LOW. That's why you need a pull-up or pull-down resistor in the circuit.

Procedure:

1. Open Arduino IDE.
2. Write the code in text editor.
3. Save the sketch with .ino extension.
4. Connect the hardware circuit and Connect your Arduino Board to the USB port of Your Computer.
5. Select the serial device of the Arduino board from the Tools | Serial Port menu.
6. Compile the file by clicking on the verify button.
7. If successful, the message "Done Compiling." will appear in the status bar.



8. If there are any errors it will show them in the Transcript window, rectify those errors and compile it again.

9. Push the reset button on the board then click the Upload button in the IDE. Wait a few seconds. If successful, the message "Done uploading." will appear in the status bar.

Code:

```
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);
  }
}
```

Result: when you press the button the LED gets ON else, the LED gets OFF.