Assignment: 5

Title: Write a program using Arduino to control LED(One or more ON/OFF) or Blinking.

Objective: Connectivity and configuration of Raspberry-pi/ Beagal board/ Arduino circuit with basic peripherals like LEDS.

Hardware Requirement: Arduino, LED, 220 ohm resistor etc

Software Requirement: Arduino IDE

Theory:

This example shows the simplest thing you can do with an Arduino to see physical output: it blinks the on-board LED.

This example uses the built-in LED that most Arduino boards have. This LED is connected to a digital pin and its number may vary from board type to board type. To make your life easier, we have a constant that is specified in every board descriptor file. This constant is LED_BUILTIN and allows you to control the built-in LED easily. Here is the correspondence between the constant and the digital pin.

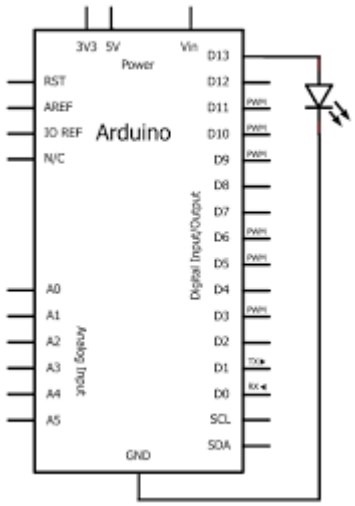
- D13 - 101
- D13 - Due
- D1 - Gemma
- D13 - Intel Edison
- D13 - Intel Galileo Gen2
- D13 - Leonardo and Micro
- D13 - LilyPad
- D13 - LilyPad USB
- D13 - MEGA2560
- D13 - Mini
- D6 - MKR1000
- D13 - Nano
- D13 - Pro
- D13 - Pro Mini
- D13 - UNO
- D13 - Yún
- D13 – Zero

If you want to lit an external LED with this sketch, you need to build this circuit, where you connect one end of the resistor to the digital pin correspondent to the LED_BUILTIN constant. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the GND. In the diagram below we show an UNO board that has D13 as the LED_BUILTIN value.

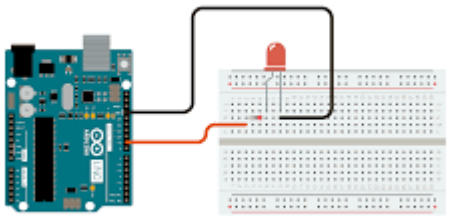
The value of the resistor in series with the LED may be of a different value than 220 ohm; the LED will lit up also with values up to 1K ohm.

Diagram:

Pin Diagram:



Schematic :



Apparatus: Arduino Uno board, Micro-IoT sensor actuator board, Power adaptor.

Interface:

LED	Arduino Pin
LED 1	0
LED 2	1
LED 3	2
LED 4	4

Procedure:

Step 1: Connect the Arduino board to the Micro-IoT Sensor board using the FRC cable provided with the board.

Step 2: Connect the Power supply adaptor and power on the circuit.

Step 3: Open Arduino IDE and create a new sketch (program) for LED blinking using the above pins.

Step 4: In the Arduino IDE go to tools□Port and select the appropriate COM port.

Step 5: In the Arduino IDE click on the upload button () to compile and download the code into the Arduino UNO. When successfully downloaded the code will start running and you can observe the LED's blinking on the board.

Code:

```
void setup() {
 // set the mode of the pins a s GPIO and direction as OUTPUT
pinMode(0, OUTPUT);
 pinMode(1, OUTPUT);
 pinMode(2, OUTPUT);
 pinMode(4, OUTPUT);
 }

void loop() {
 // put your main code here, to run repeatedly:
//Make LED ON
 digitalWrite(0, HIGH);
 delay(100);
 digitalWrite(1, HIGH);
 delay(100);
 digitalWrite(2, HIGH);
 delay(100);
 digitalWrite(4, HIGH);
 delay(200);
 //Make LED OFF
 digitalWrite(0, LOW);
 delay(100);
 digitalWrite(1, LOW);
 delay(100);
 digitalWrite(2, LOW);
 delay(100);
 digitalWrite(4, LOW);
 delay(200);
 }
```

Code:

After you build the circuit plug your Arduino board into your computer, start the Arduino Software (IDE) and enter the code below. You may also load it from the menu File/Examples/01.Basics/Blink . The first thing you do is to initialize LED_BUILTIN pin as an output pin with the line

pinMode(LED_BUILTIN, OUTPUT);
In the main loop, you turn the LED on with the line:
digitalWrite(LED_BUILTIN, HIGH);
This supplies 5 volts to the LED anode. That creates a voltage difference across the pins of the LED, and lights it up. Then you turn it off with the line:

digitalWrite(LED_BUILTIN, LOW);
That takes the LED_BUILTIN pin back to 0 volts, and turns the LED off. In between the on and the off, you want enough time for a person to see the change, so the **delay()** commands tell the board to do nothing for 1000 milliseconds, or one second. When you use the **delay()** command, nothing else happens for that amount of time. Once you've understood the basic examples, check out the BlinkWithoutDelay example to learn how to create a delay while doing other things. Once you've understood this example, check out the DigitalReadSerial example to learn how read a switch connected to the board.

Observation:

You can observe the LED's turning ON and OFF. You can also change the delay and see the changes.

Conclusion:LED blinking is controlled by connectiong to Aurdino board