University of Mauritius

IOT Workshop

Table of Content

IoT Lab

Practical Session 1

Objectives:

Participant should be able to:

1. Program basic LED Diode On and Off (Blink) using an Arduino.
2. Program a Light Dependent resistor LDR to monitor light intensity.
3. Switch On and Off a Relay.
4. Find IOT solution to everyday problems.

Note: This lab uses the AUTODESK TINKERCAD circuit web platform to simulate electronic prototype.

1. Make sure you sign in using an account on https://www.tinkercad.com/
2. Create new Circuit.
3. Check out our Github repository: https://github.com/MRabill/IOT_Workshop.git

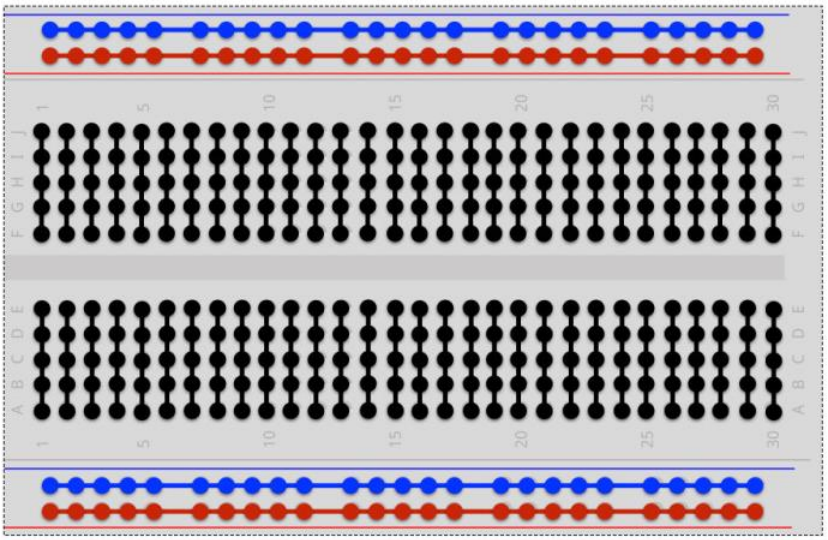
For the first practice, we will make a LED turn ON and OFF after some delay

Firstly, we need to know what a breadboard is and how it works

A **breadboard** is a solderless device for temporary prototype with electronics and test circuit designs.

The way it works is that the holes are connected to each other in a way that allows you to connect components easily by inserting the components pin into the hole.

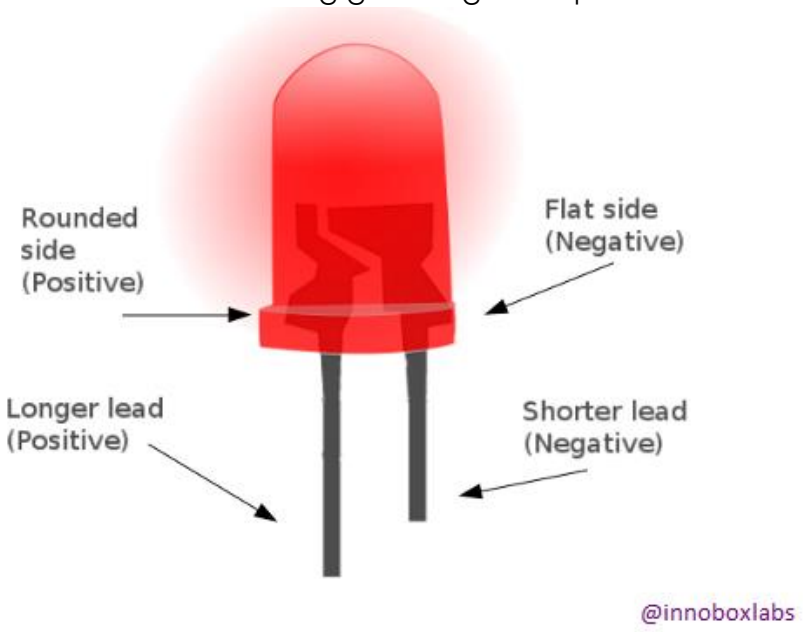
But we must know how the connections are made inside the breadboard. The diagram below shows how the holes are connected.



You can see that on the sides we have 2 rows one blue and the other one red. These are basically used for Power connection. (RED for Positive and BLUE for negative). The middle holes are used to connect components and sensors.

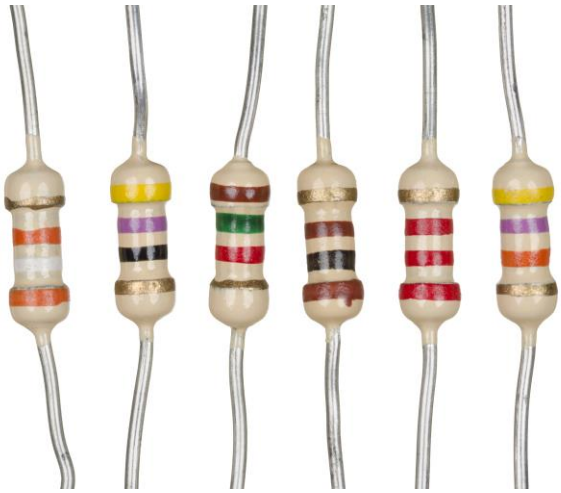
So now we can easily connect our Led without any soldering.

LEDs have polarity, which means they will only light up if you orient the legs properly. The long leg is typically positive and should connect to a digital pin on the Arduino board. The short leg goes to ground pin GND.



As the LED works with Low power, we need a resistor to reduce the 5V power supply from the pin 13 to a value that make the LED to lights up without getting overloaded.

If the LED get overheated, it will burn out. So, for this we will need a component call a **Resistor**. A **Resistor** is a passive two-terminal electrical component that implements electrical resistance as a circuit element.



For our project we will use a 1k Ohms resistor

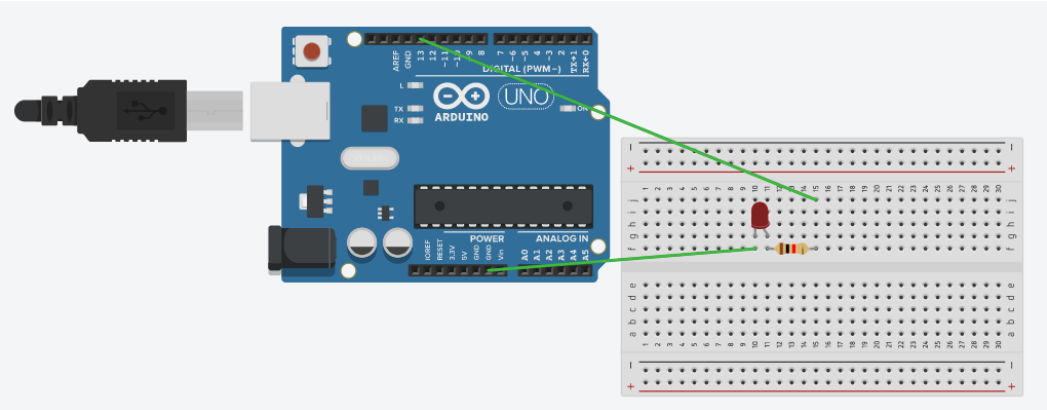
Step 1

Drag and drop these following components from the component library on the right side of your screen

- Arduino UNO
- Breadboard Small
- Resistor 1K Ohms
- LED

Step 2

Connect the component as the below diagram



Step 3

Code

To blink the LED the first thing we do is define a variable that will hold the number of the pin that the LED is connected to. We will use an integer variable (called an int) and pin 13 on the Arduino

```
int ledPin = 13;
```

The second thing we need to do is configure as an output the pin connected to the LED. We do this with a call to the pinMode() function, inside of the sketch's setup() function:

```
void setup()
{
  pinMode(ledPin, OUTPUT);
}
```

Finally, we have to turn the LED on and off with the sketch's loop() function. We do this with two calls to the digitalWrite() function, one with parameter HIGH to turn the LED on and one with LOW to turn the LED off. If we simply alternated calls to these two functions, the LED would turn on and off too quickly for us to see, so we add two calls to delay() to slow things down. The delay function works with milliseconds, so we use 1000ms to pause for a second.

```
void loop()
```

```
{
digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Step 4

Full code

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

Click the Start Simulation button on the top right corner to run the Arduino simulation.

The LED should turn On and OFF with a 1 second delay

Sensor

The sensor we will use is a LDR (Light dependent resistor) which is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface

You will need to capture the data from the sensor and display it in the serial monitor

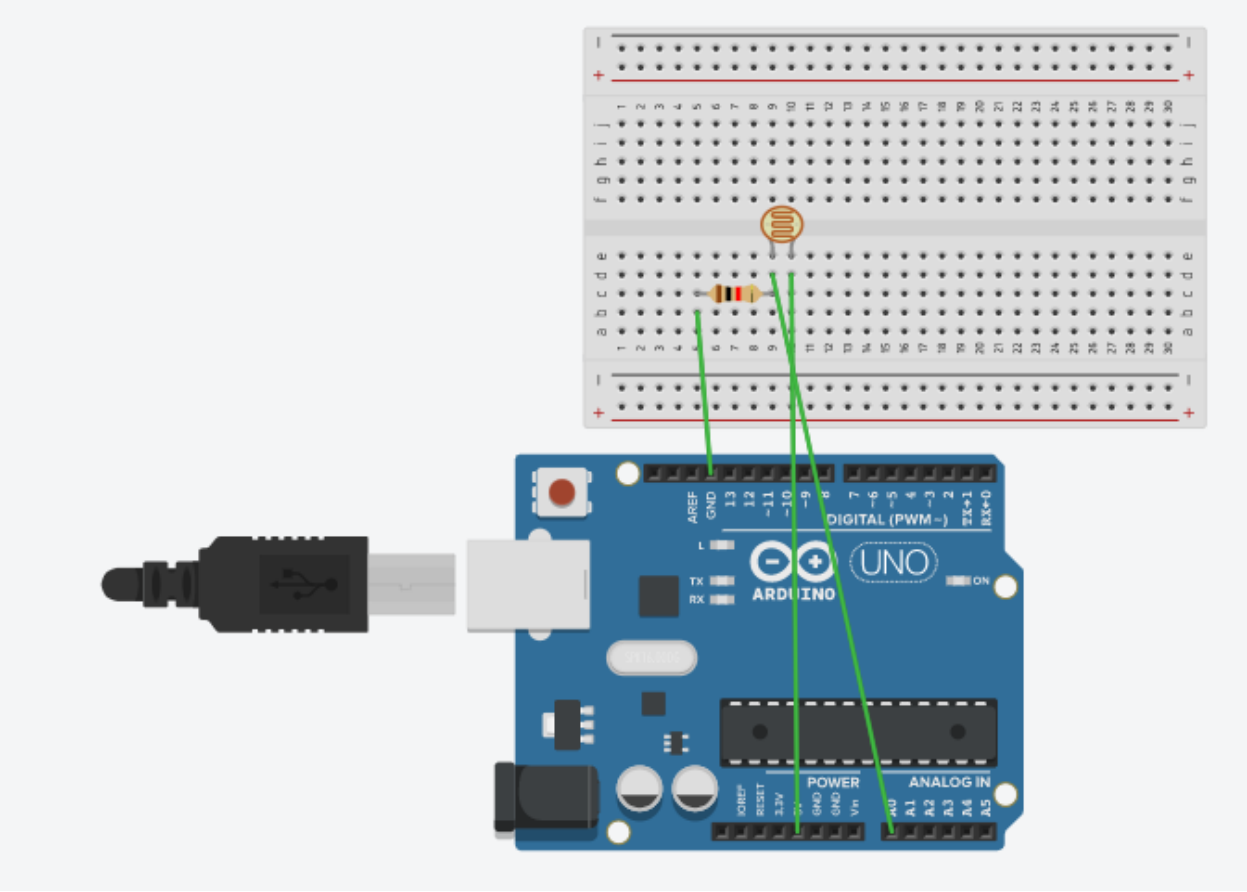
Step 1

Drag and drop these following components from the component library on the right side of your screen

- Arduino UNO
- Breadboard Small
- Resistor 1K Ohms
- LDR

Step 2

Connect the component as the below diagram



Step 3

Code

To store the sensor value, we will need a sensor value variable (sensorValue) and declare the sensor Pin to be A0 which is Analog Pin 0

```
int sensorValue;

int sensorPin = A0;
```

The second thing we need to do is configure as an Input the pin connected to the LDR. We do this with a call to the pinMode() function, inside of the sketch's setup() function. We also need to create a serial communication to the Arduino to be able to display the value (Serial.begin(9600))

```
void setup() {

Serial.begin(9600);

pinMode(sensorPin, INPUT);

}
```

Now we need to program the code for reading the sensor and display it in the serial monitor. To read the sensor value we use the analogRead function and as parameter the sensor Pin. We assign this value to the SensorValue variable, and we proceed by serial print the value to the screen.

To make things a little bit slower we can add a delay of 500ms

```
void loop() {

 sensorValue= analogRead (sensorPin);

 Serial.println(sensorValue);

 delay (500);

}
```

Step 4

Full code

```
int sensorValue;
int sensorPin = A0;
void setup() {
Serial.begin(9600);
pinMode(sensorPin, INPUT);
}
void loop() {
 sensorValue= analogRead (sensorPin);
 Serial.println(sensorValue);
 delay (500);
}
```

Click the Start Simulation button on the top right corner to run the Arduino simulation.

Click the Serial Monitor icon below your code to view the serial monitor. You can adjust the LDR light intensity using the slider above the LDR to see your code in action

Relay

A Relay is a switch that open and close circuits electronically. It uses 5V or 12V to operate the switch and can be used to switch high voltage for example your house light.

In this lab session we will turn on and off a relay in TinkerCad which is connected to a 9V DC motor.

Step 1

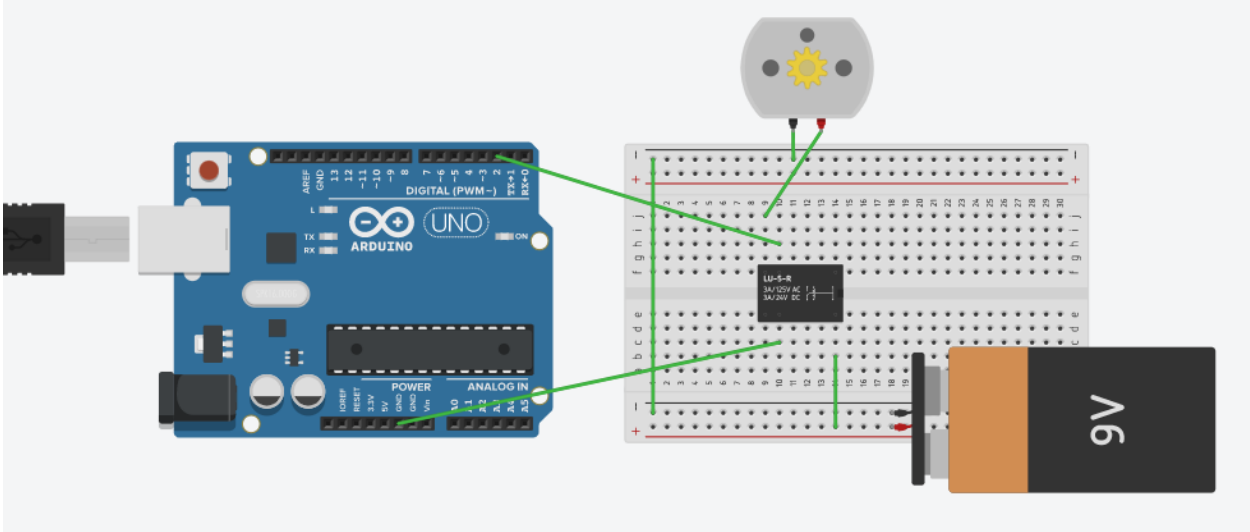
Drag and drop these following components from the component library on the right side of your screen

- Arduino UNO
- Breadboard Small
- Relay SPDT
- DC Motor

- 9V battery

Step 2

Connect the component as the below diagram



Step 3

Code

We first start declaring the relay pin to be at pin 2 on the Arduino.

```
int relayPin = 2;
```

The second thing we need to do is configure as an Output the pin connected to the Relay. We do this with a call to the pinMode() function, inside of the sketch's setup() function.

```
void setup()
{
  pinMode(relayPin, OUTPUT);
}
```

Now we need to program the code for turning ON and OFF the relay.

Code as previous blink code

```
void loop()
{
  digitalWrite(relayPin, HIGH);
  delay(1000);
  digitalWrite(relayPin, LOW);
  delay(1000);
}
```

Step 4

Full code

```
int relayPin = 2;
void setup()
{
  pinMode(relayPin, OUTPUT);
}
void loop()
{
  digitalWrite(relayPin, HIGH);
  delay(1000);
  digitalWrite(relayPin, LOW);
  delay(1000);
}
```

Click the Start Simulation button on the top right corner to run the Arduino simulation.

You should see the motor turning on and off after 1s delay

Case Study

Problem: Mr. Sam grows some vegetables in his garden. But as he is a very busy man, he often forgets to water his plants. But when he remembers, he waters his plant after the sunrise and before the sunset. He knows that you are an IOT enthusiast and ask you to develop a simple and small system to water his plant automatically. He asks you to use his watering system that is connected to a 9v pump.

Develop a prototype on TinkerCad and test it.

Step 1

Gather requirements for the case study above

1. Water the plants after sunrise and before sunset
2. Use the 9V pump watering system

Find Solution

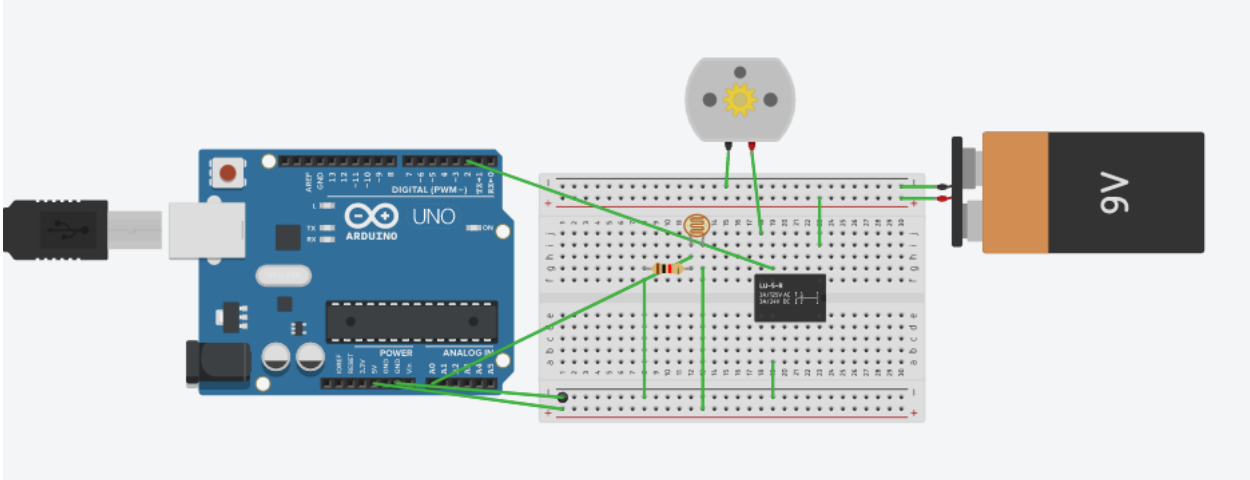
A possible solution can be the use of an Arduino with a LDR sensor to monitor the light intensity which determine day and night. This can trigger a relay that will turn ON and OFF the 9V pump.

Drag and drop these following components from the component library on the right side of your screen

- Arduino UNO
- Breadboard Small
- Relay SPDT
- DC Motor (act as the pump)
- Resistor 1k
- LDR
- 9V battery

Step 2

Connect the component as the below diagram



Step 3

Code

We first start declaring the pin on the Arduino. We have the motorPin which is connected to the relay and pin 2 on the Arduino, A sensor (LDR) at pin A0, A variable to store the Light intensity (sensorRead) and a Boolean to know if we need to irrigate or not

```
int motorPin = 2;

int sensorRead;

int sensor = A0;

bool irrigation = false;
```

The second thing we need to do is configure as an Output the pin connected to the Relay and Input the LDR Pin. We do this with a call to the pinMode() function, inside of the sketch's setup() function.

```
void setup()

{

  pinMode(motorPin, OUTPUT);

  pinMode(sensor, INPUT);

}
```

Now we need to program the code for turning ON and OFF the motor once after sunrise and before sunset.

We first read the value of the LDR

If the value of the LDR is greater than a threshold (after sunrise) and irrigation is false, then we turn on the motor for test amount of 10 seconds.

But to prevent the code from running again we should assign the irrigation to true.

Then if the value of the LDR is less than a threshold (before sunset) and the irrigation is true then we turn on the motor for test amount of 10 seconds.

And set the irrigation variable to false again

```
void loop()
{
  sensorRead = analogRead(sensor);
  if ( sensorRead > 635 && irrigation == false)
  {
      digitalWrite(2, HIGH);
      delay(10000);
      digitalWrite(2, LOW);
    irrigation = true;
  }
  if ( sensorRead < 150 && irrigation == true)
  {
      digitalWrite(2, HIGH);
      delay(10000);
      digitalWrite(2, LOW);
    irrigation = false;
  }
}
```

Step 4

Full code

```
int motorPin = 2;
int sensorRead;
int sensor = A0;
bool irrigation = false;

void setup()
{
  pinMode(motorPin, OUTPUT);
  pinMode(sensor, INPUT);
}

void loop()
{
  sensorRead = analogRead(sensor);

  if ( sensorRead > 635 && irrigation == false)
  {
          digitalWrite(2, HIGH);
          delay(10000);
          digitalWrite(2, LOW);
    irrigation = true;
  }

  if ( sensorRead < 150 && irrigation == true)
  {
          digitalWrite(2, HIGH);
          delay(10000);
```

```
        digitalWrite(2, LOW);
    irrigation = false;
  }
}
```

Click the Start Simulation button on the top right corner to run the Arduino simulation.

Adjust the LDR to full and you should see the motor turning on for 10 seconds and turn off

Then if you adjust the LDR to dark you should again see the motor turning on for 10 seconds and turn off.

These get in a loop and run each day

References

Images:

https://www.pololu.com/product/4000

https://magpi.raspberrypi.org/articles/breadboard-tutorial

https://owlcircuits.com/resources/how-to-determine-led-polarity/

https://en.wikipedia.org/wiki/Resistor

Workshop Design by:

Madarboccus Rabill

Student at University of Mauritius