

Experiment No. 4

Aim: Control Structure writing programs for if else, for and while.

A programming language's control structures allow you to take action based on certain conditions. The Arduino control structures are very similar to the control structure of the C language and include:

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Conditional Statements

If statements and *while* statements are called *conditional statements*. They evaluate a particular condition, and execute a series of statements in brackets if the condition is true { }. Here's an example:

```
if (pin_b == LOW)
{
digitalWrite(ledPin, HIGH); // This tells the LED to turn on
}
```

“if” statement consists of word "if" followed by a condition in parentheses. If condition is true, then statements between brackets that follow will be executed. In this case, if digital value of pin_b is LOW, then program will set value of ledPin to HIGH. Statements between brackets are executed only once. While loop is similar to if statement, but in this case, statements between brackets are executed over and over until condition becomes false.

Here's an example:

```
while (pin_0 == LOW)
{
digitalWrite(ledPin, HIGH);
delay(100);
digitalWrite(ledPin, LOW);
delay(100);
switchValue = digitalRead(pin_0); // This stores a new value into the variable switchValue
}
```

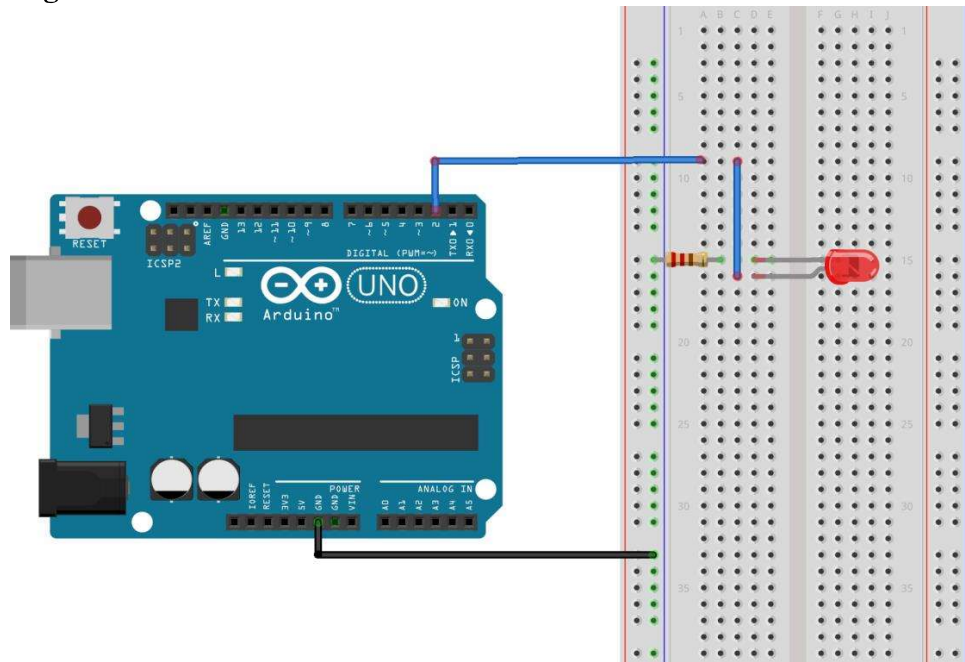
In this example, the program will toggle the state the LED (i.e. blink) until a user presses a switch causing the digital value at pin_0 to go HIGH.

EXPERIMENT 1: Using conditional statements to change brightness of an LED. Assume LED to be connected to pin 2 of Arduino.

Hardware Required

- | | |
|-------------------|------------------------|
| • 1 x Arduino UNO | • 1 X 10k ohm resistor |
| • 1 x breadboard | • Jumper wires |
| • 1 x LED | |

Wiring Diagram



1. Using For LOOP

```
const int ledPin = 2;
void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(ledPin, brightness);
    delay(10);
  }

  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
}
```

2. Using If-else

```
int ledPin = 2;
int brightness = 0;
int fadeAmount = 5;

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

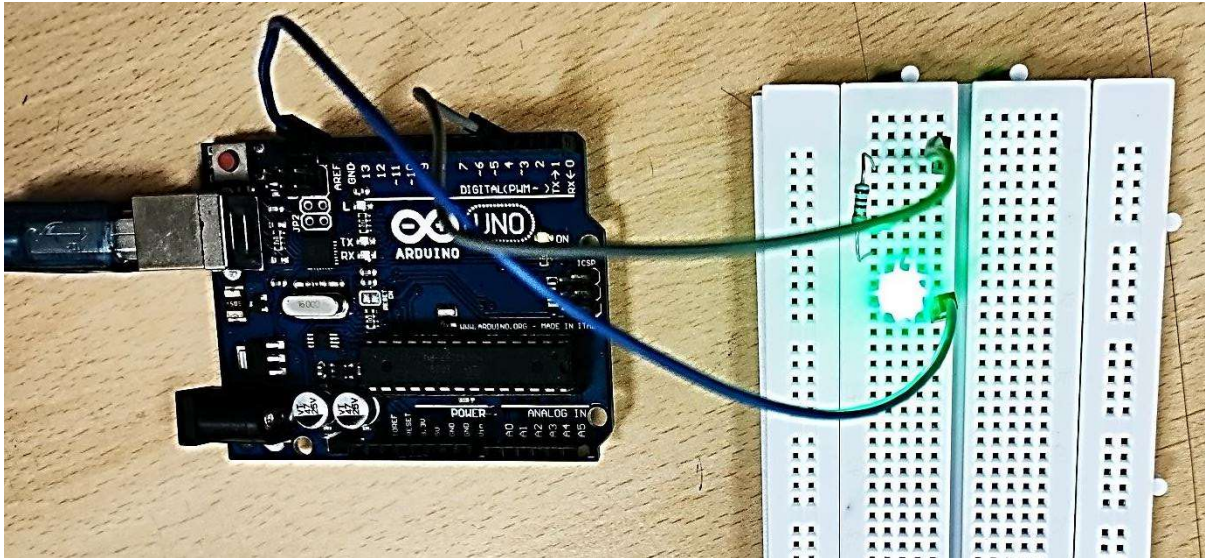
void loop() {
  analogWrite(ledPin, brightness);
```

```
    if (brightness <= 0) {  
        fadeAmount = 5;  
    } else if (brightness >= 255) {  
        fadeAmount = -5;  
    }  
  
    brightness += fadeAmount;  
    delay(30);  
}
```

3. Using Do-while loop

```
const int ledPin = 2;  
int brightness = 0;  
int fadeAmount = 5;  
  
void setup() {  
    pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
    do {  
        analogWrite(ledPin, brightness);  
        brightness += fadeAmount;  
        delay(50);  
    } while (brightness <= 255);  
  
    do {  
        analogWrite(ledPin, brightness);  
        brightness -= fadeAmount;  
        delay(50);  
    } while (brightness >= 0);  
}
```

Wiring Diagram



Result

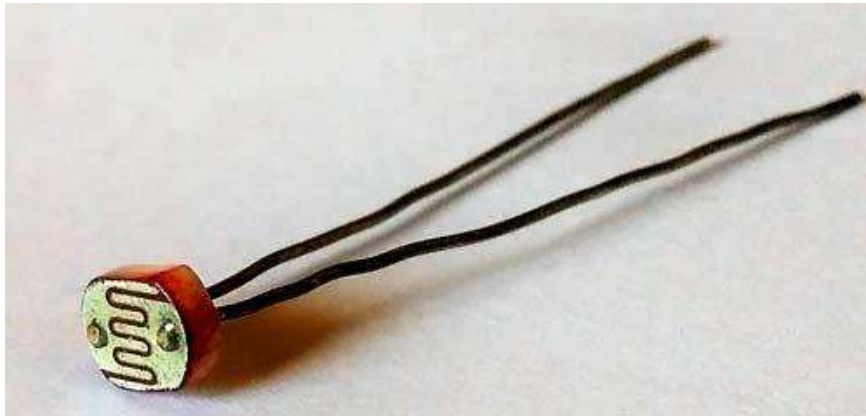
PASTE YOUR SCREENSHOT OF THE SUCCESSFUL RUN OF PROGRAM INCLUDING HARDWARE CONNECTION AND ARDUINO IDE

Experiment No. 5

Aim: a. Measuring light with Lux and a photo resistor demonstration
b. Reading and writing digital and analog values. Digital and analog read/write demonstration.

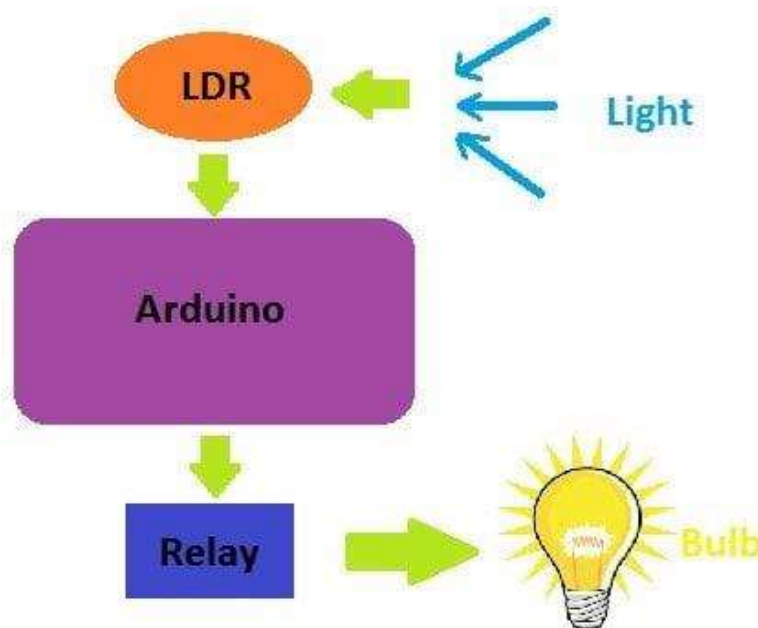
Theory

LDR stands for Light Dependent Resistor. LDR is a type of resistor that changes resistance as light on its surface changes i.e., less light or more darkness on LDR surface causes its resistance to increase. LDR is connected in series with a 10k resistor. LDR is used to detect light levels and switch an LED on or off.



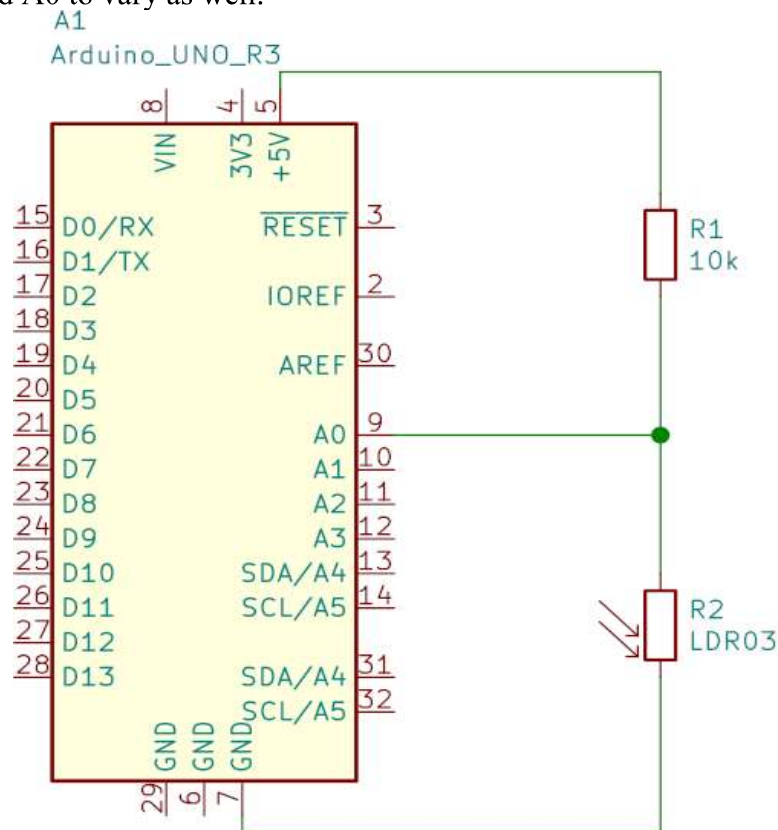
Working of LDR with Arduino Uno

- LDR gives out analog voltage when connected to VCC (5V), which varies with input light intensity.
- i.e., greater the intensity of light, greater will be corresponding voltage from LDR.
- LDR gives out an analog voltage, as it is connected to analog input pin on Arduino.
- Arduino, with its built-in ADC (analog-to-digital converter), converts analog voltage (from 0-5V) into a digital value in range of (0-1023).
- When there is sufficient light in its environment or on its surface, converted digital values read from LDR through Arduino will be in range of 800-1023.

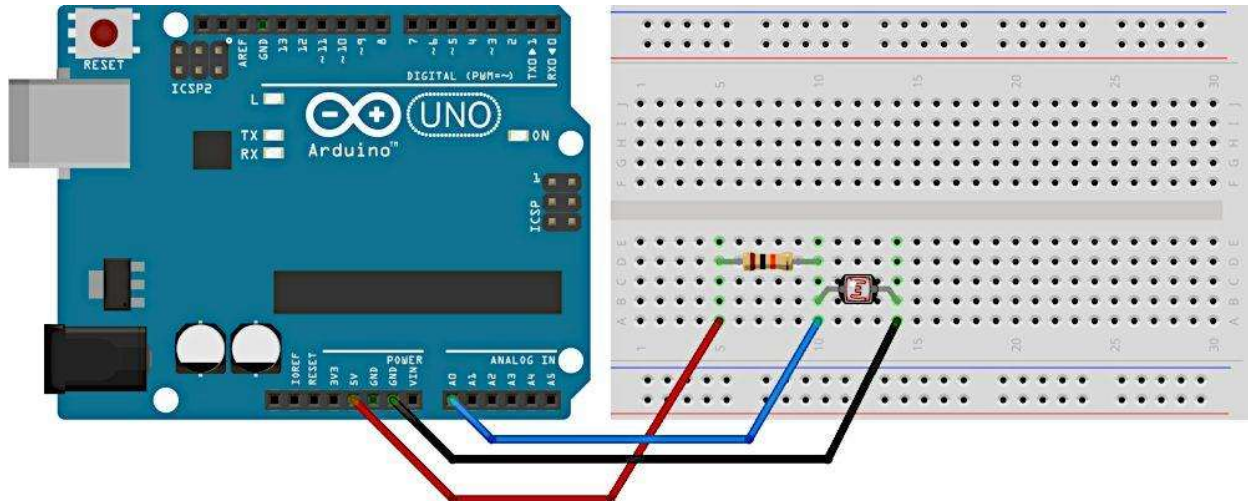


Connection Diagram

- Here, 10k resistor R1 and LDR R2 form a voltage divider.
- i.e, voltage at junction of R1 and R2 is divided voltage from 5V that is across them.
- As light varies on LDR surface, so does its resistance which causes voltage between GND and A0 to vary as well.



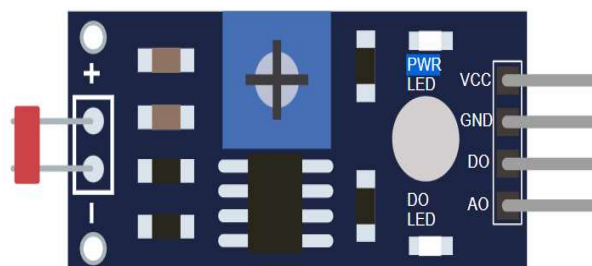
Arduino Uno LDR Breadboard Circuit



LDR Module Pin Description

Pins on sensor

- VCC - Positive power supply,
- GND - Ground
- Do - Digital output
- Ao - Analog output



Code 1: Working of LDR with Arduino Uno

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

void loop()
{
  int sensorValue = analogRead(A0);
  if (sensorValue > 700)
  {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else
  {
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(10);
}
```

Result

PASTE YOUR SCREENSHOT OF THE SUCCESSFUL RUN OF PROGRAM INCLUDING HARDWARE CONNECTION AND ARDUINO IDE

Question 1: Working of LDR using While Statement to blink two LEDs for two different conditions

Result

PASTE YOUR SCREENSHOT OF THE SUCCESSFUL RUN OF PROGRAM INCLUDING HARDWARE CONNECTION AND ARDUINO IDE