Faculty of Computing

Year 1 Semester 1 (2024)

IT1140 – Fundamental of Computing Lab Sheet 08

Objectives:

This practical enables the student to learn how to use LCD display, LDR, Servo motor and the PIR sensor using Arduino.

Activity 1

a. Design the circuit diagram given in figure 8.1 in Wokwi. Use the below items to design your circuit diagram.
   - Arduino Uno
   - LCD Display (16x2)
   - Photoresistor LDR (Light-Dependent Resistor)

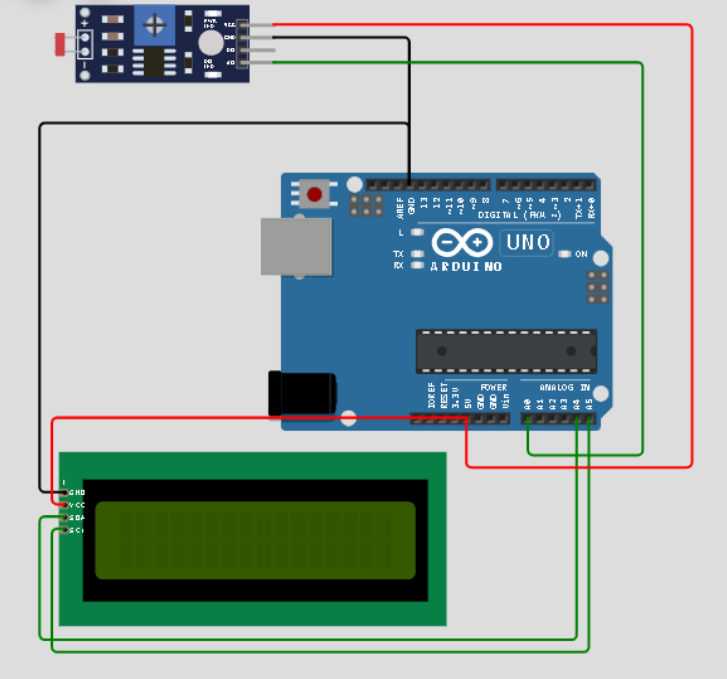

Figure 8.1

a) Connect the components correctly and design the circuit as shown in figure 8.1.

b) Add the following libraries in your code from the Library Manager.

   a. LiquidCrystal I2C

   b. LiquidCrystalWired

c) Add the following C++ code to your program to include the necessary libraries and to implement the `setup()` method.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);


void setup() {
  pinMode(A0, INPUT);
  pinMode(10, OUTPUT);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  delay(1000);
  lcd.clear();
  delay(1000);
  lcd.setCursor(0, 0);
  lcd.print("Initializing...");
  delay(1000);
}
```

d) The LDR is connected to analog pin A0 to measure the light intensity. The brightness can be changed using the Illumination (lux) value of the LDR. This value needs to be read and mapped to a percentage (0 to 100%) and then displayed on the LCD using the I2C protocol. The following loop () method is implemented to control it and to display the brightness value.

```
void loop() {
  int ldr = analogRead(A0);
  int brightness = map(ldr, 0, 1023, 100, 0);
  Line A
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Light: ");
  Line B
  lcd.print("%");
  delay(3000);
}
```

i. Complete **Line A** to display the brightness value in the serial monitor.

ii. Complete **Line B** to display the brightness value in the LCD screen.

iii. What is the purpose of **`lcd.clear()`** in the above code?

e) Simulate your circuit and observe the behaviour of the voltage value appearing in LCD screen by changing the brightness (lux) value.

f) Briefly explain your observations.

g) Modify the above circuit by connecting a LED bulb to the Aduino.

h) Control your circuit by adding a C++ code to turn ON the LED when the brightness of the environment is low and, turn OFF the LED when the brightness is high. Use the below threshold values. (*Note: Consider the pin no used to connect the LED to write the code and include it to the loop () method.*)

Brightness	**Action**
Below 25%	Turn ON LED
Equal or above 25%	Turn OFF LED

Activity 2:

b. The circuit in figure 8.2 automatically controls the door (Servo motor) with PIR Motion Sensor. Design the circuit using Wokwi using the below items.
    - Arduino Uno
    - PIR sensor
    - Servo motor

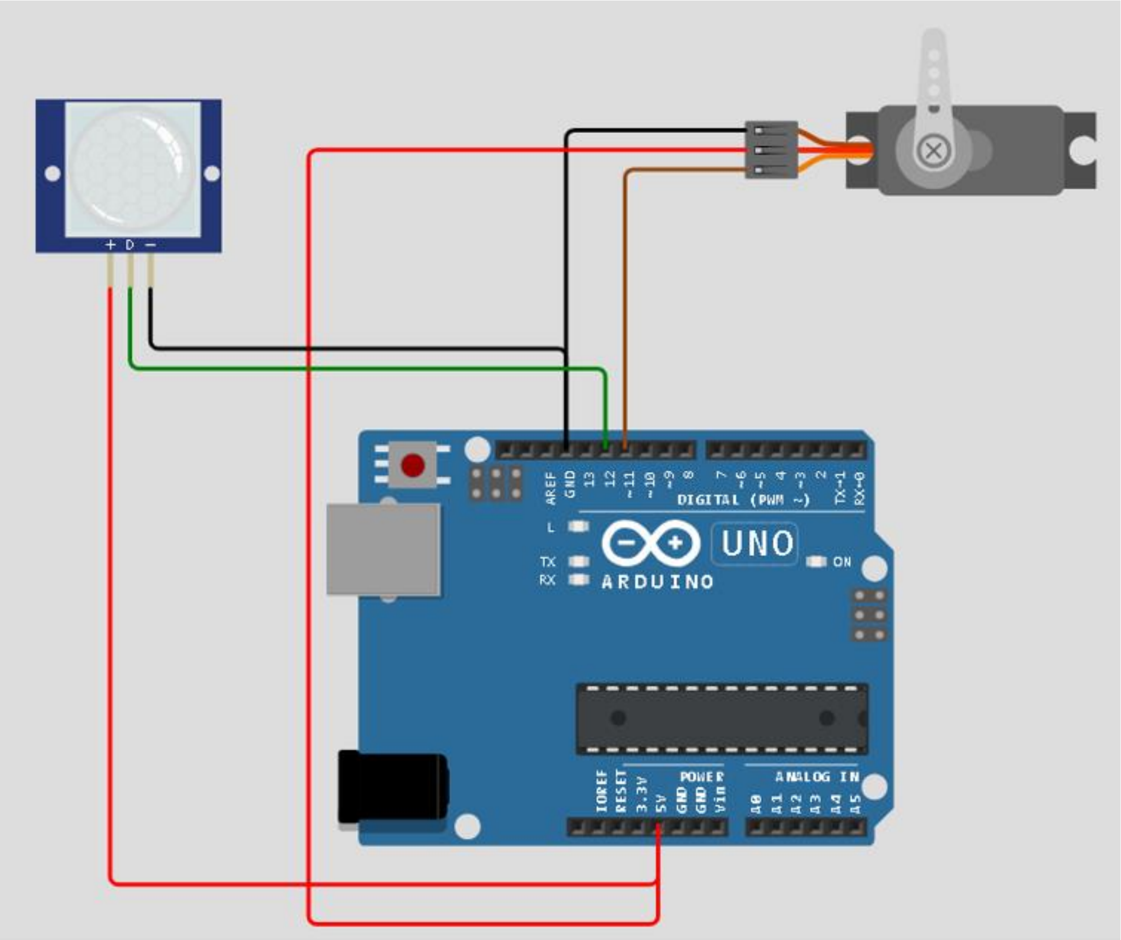

Figure 8.2

Include the below code in your project. Note the following connections.

PIR motion sensor

- VCC (Red wire) connected to 5V on the Arduino.
- GND (Black wire) connected to GND on the Arduino.
- Output (Green wire) connected to a digital pin on the Arduino pin 12.

Servo motor

- Power (Red wire) connected to 5V on the Arduino.
- Ground (Black wire) connected to GND on the Arduino.
- Signal (Brown wire) connected to a PWM (Pulse Width Modulation) pin 11.

```
#include <Servo.h>

int ser_pin = 11;
int pos = 0;
Servo servo1;

void setup()
{
    pinMode(12, INPUT);
    servo1.attach(ser_pin);
}

void loop() {
    int motion = digitalRead(12);

    if ( ) {  // Line A

        for ( ) { // Line B
            servo1.write(pos);
            delay(30);
        }

        delay(3000);


        for ( ) {  // Line C
            servo1.write(pos);
            delay(30);
        }

        delay(30);
    }
}
```

Do the following.

a) Include the Servo.h library in the code.

b) Declare and initialize `ser_pin` and `pos` variables as in the code.

c) Create a Servo object to control the servo motor as in the code.

d) Implement the `Setup ( )` method.

e) Complete Line A, Line B and Line C in the `loop ( )` method as per the below instructions.

   i. Line A: Check whether a motion is detected. If there is a motion it indicates HIGH.

   ii. Line B: Implement the `for` loop to control the movement of the servo from 0 to 180 degrees.

   iii. Line C: Implement the `for` loop to control the movement of the servo back from 180 to 0 degrees.