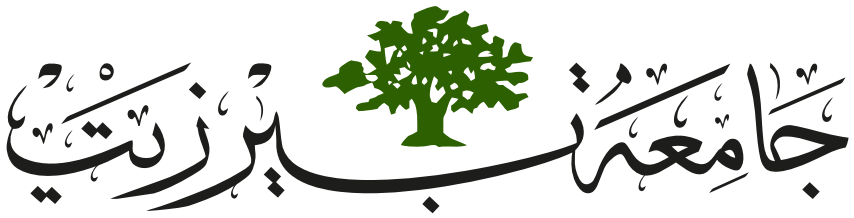

BIRZEIT UNIVERSITY

Department of Electrical & Computer Engineering
ENCS4380 - INTERFACING TECHNIQUES

IR sensor

Prepared by:
Zakiya AbuMurra-1191636
Ibraheem Duhaidi – 1190283
Eyab Ghifari – 1190999
Tariq Odeh – 1190699
Date: May 25,2023

Contents

List of Figures

1 Theory

The aim of the task is to build a system that utilizes a servo motor and a Keyes IR sensor. A 20x20 cm space with a gap in one of the walls can hold the system. Both automated and manual modes should be available, and an LCD module and serial port should be used to display information about openings.The system contain IR sensor, servo motor and control bit using push button and LCD.

1.1 IR sensor

A small device called a Keyes IR sensor is capable of detecting infrared light. It operates by discharging infrared light and subsequently observing its reflection. The infrared light is reflected back to the sensor when an item is in front of it. This reflection can be detected by the sensor, which can then send out a signal indicating the presence of an object. Projects like remote controls and obstacle detection systems frequently employ it. How it Works ?

1. A sensor that detects infrared light.

2. The transmitted light reflects back when it comes into contact with an item.

3. The reflected infrared light is picked up by the IR sensor.

4. The intensity of the light that is reflected reveals the existence or absence of an obstruction.

5. The sensor turns the light that it has detected into an electrical signal. The signal is contrasted with a threshold value in step 6 to assess the presence or absence of obstacles.

6. The performance of a sensor can be impacted by variables like distance, object surface, and ambient lighting.

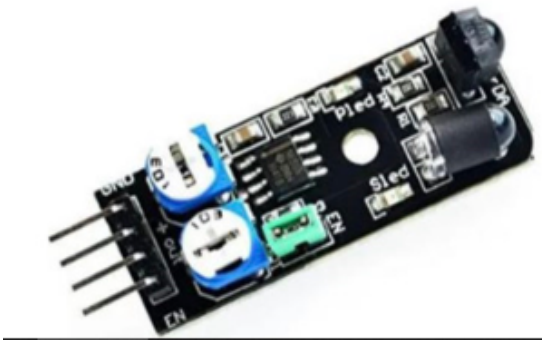

Figure 1.1: IR sensor

1.2 Servo motor

Servo motors are unique motors with extremely accurate motion capabilities. It has a sensor inside that helps it determine its precise location. Sending the servo motor signals in the form of pulses allows you to control it. Where the motor should move depends on the pulse length.

A steady power source, typically between 4.8V and 6V, is required for servo motors. They are often used in robotics for things like grippers and movable robot arms. To control their movements, they are also applied to remote-controlled vehicles including cars, planes, and cameras.

You can use a library or module that makes controlling servo motors simple to use. These libraries offer straightforward functionality for adjusting the position and speed of the motor.

Figure 1.2: Servo motor

1.3 Automatic Mode

In Automatic Mode, the servo motor rotates the IR sensor to scan the environment like a rotating radar.then the IR sensor senses infrared reflections to determine the presence of objects or openings. The system notifies the user of the location of the opening (front, back, left, or right) by a serial port message sent every 20 seconds. finally, on the LCD module, the outcomes are also displayed.

1.4 Manual Mode

The control bit allows the user to select manual mode.to check a certain direction (forward, back, left, or right), the user issues commands using switches,to determine if an opening exists or not, the IR sensor scans that direction. The system notifies the user over the serial port whether or not the given direction has an opening.Finally , on LCD the result is displayed,

2 Procedure

In k3ksh Lab , we build the circuit to achieve the aim of the task.

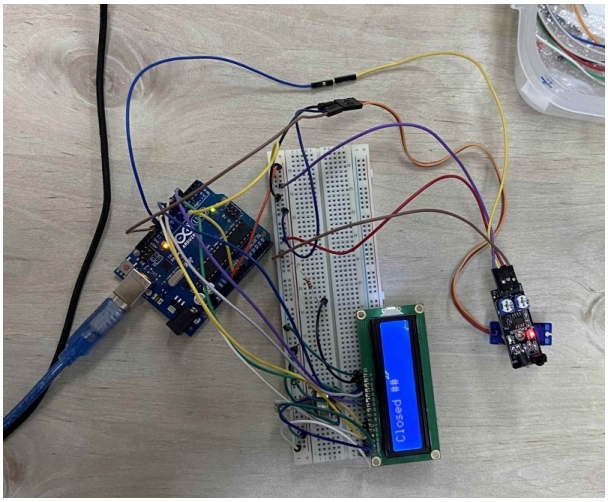

Figure 2.1: Build the circuit in LAB

Initially, we attempted to accomplish the task using three push buttons, but unfortunately, it didn't yield the desired results. We faced difficulties and realized that a simpler approach was needed. As a result, we decided to reduce the complexity by utilizing only one "push" button. This adjustment proved to be more successful. We were finally able to make the necessary adjustments to the code and gain full control over the process. Through perseverance and trial and error, we achieved our goal and successfully harnessed the functionality we were aiming for.

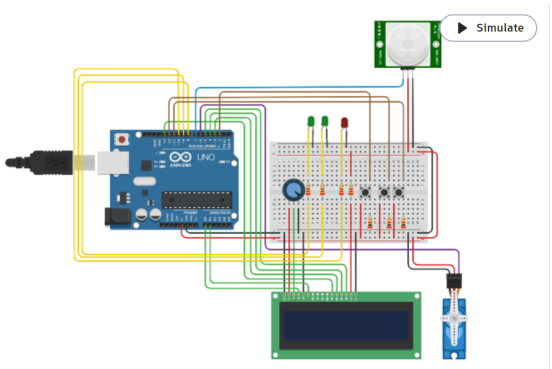

Figure 2.2: Build the circuit with 3 push buttons using tinkercad

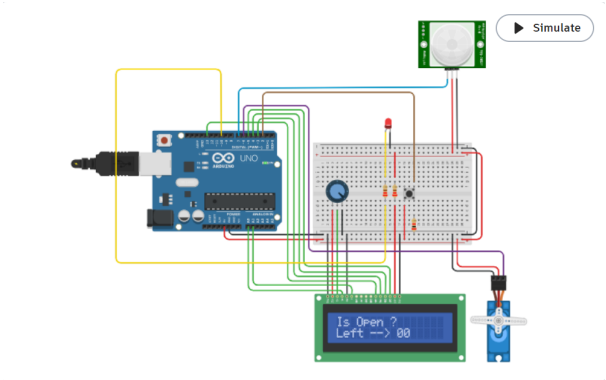

Figure 2.3: Build the circuit with one push button using tinkercad

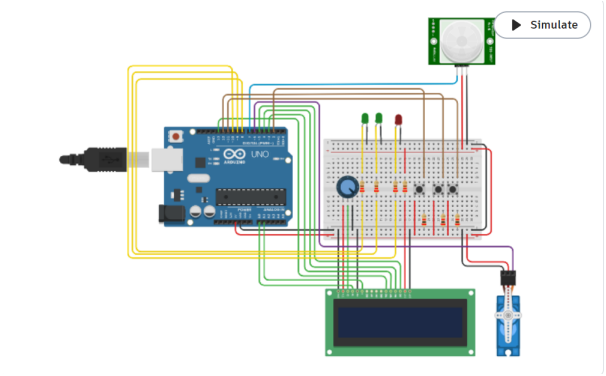

Figure 2.4: final circuit using tinkercad

2.1 Links

1. Circuit with three push buttons
2. Circuit with one push button
3. Circuit without push buttons which is the final circuit

3 Appendix

```
#include <Servo.h>
#include <LiquidCrystal.h>

const int rs = 9, en = 8;
const int d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int servoPin = 12;                  // Pin connected to the servo
↪ motor
int pirPin = 11;                    // Pin connected to the PIR
↪ sensor output
Servo myServo;
bool automaticMode = false;

void setup()
{
    Serial.begin(9600 );
           lcd.begin(16, 2);
    lcd.print("<<<Hello>>>");
    pinMode(pirPin, INPUT);                 // Set PIR sensor pin as
    ↪ input
    myServo.attach(servoPin); // Attach the servo to the specified
    ↪ pin
    myServo.write(90);       // Move the servo to the initial
    ↪ position (90 degrees)
}

void loop()
{
    if (Serial.available())
    {
        char input = Serial.read();

        if (input == '4')
        {
            automaticMode = true;
            automaticServoControl();
        }
        else if (input >= '0' && input <= '2')
        {
            automaticMode = false;
            manualServoControl(input - '0');
        }
    }
}
```

```
void automaticServoControl()
{
  for(int i=0 ; i<3 ; i++){
    manualServoControl(i);
    delay(1000);
  }
  delay(2000);
  myServo.write(90);
}

void manualServoControl(int direction)
{
    int pirState = digitalRead(pirPin);
    switch (direction)
    {
    case 0: // Front
        Serial.println("Front");
        myServo.write(0);
        delay(800);
        pirState = digitalRead(pirPin);
        if (pirState == LOW){
            lcd.clear();
            lcd.print("CLOSED!");
        delay(500);

        }
        else{
          lcd.clear();
          lcd.print("OPEN ##");
        }


        break;
    case 1: // Left
        Serial.println("Left");
        myServo.write(90);
        delay(800);
        pirState = digitalRead(pirPin);
        if (pirState == LOW){
          lcd.clear();
          lcd.print("Closed!");

        }
        else{
          lcd.clear();
            lcd.print("Open!");

        }
```

```
            break;

    }
    }
```

References