BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

INTERFACING TECHNIQUES

ENCS4380

Task #2

Arduino basics and interrupts

Prepared by:

Mohammad Fareed **ID: 1212387**

Ahmad Hamdan **ID: 1210241**

Qossay Rida **ID: 1211553**

Mohammed Abed Alkareem **ID: 1210708**

Instructor: **Dr. Wasel Ghanem**

Teaching assistant: **Mr. Loai Shehadeh**

Date: **Monday, May 27, 2024**

Abstract

This task involves building a distance measurement system with detection motion and emergency action with push button using Arduino,. The ultrasonic sensor will measure the distance between the system and an object, the LEDs will visually represent this emergency, the IR will detect the motion of the object and the LCD will display the information.

Table of Contents

Table of Figures

1. Theory

The HC-SR04 ultrasonic module operates based on the principles of SONAR and RADAR systems. It comprises an ultrasonic transmitter, receiver, and control circuit integrated onto a single board, with only four pins: Vcc, Gnd, Trig, and Echo.

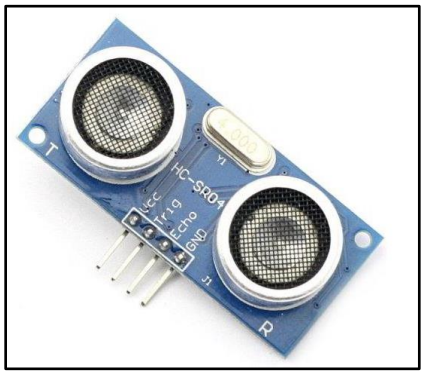

Figure 1-1 Ultrasonic HC-SR04 Module

When a pulse of 10 microseconds or longer is applied to the Trig pin, the module generates eight pulses of 40 kHz. Subsequently, the control circuit in the module sets the Echo pin high.

The Echo pin remains high until it receives the echo signal of the transmitted pulses back. The duration for which the Echo pin stays high, representing the width of the Echo pulse, indicates the time taken for the ultrasonic sound to travel towards the object and return.

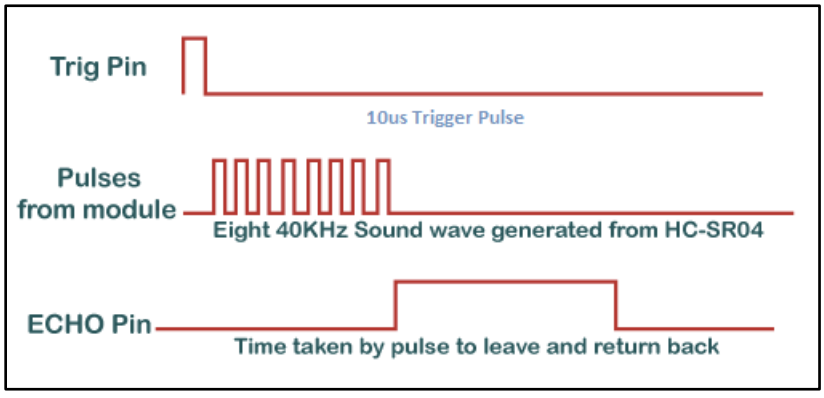

Figure 1-2 ultrasonic Signals

By utilizing this time along with the speed of sound in air, it's possible to calculate the distance to the object using a straightforward formula relating speed, time, and distance.

```
distance = time x 0.034/2
```

The Keyestudio KY-032 IR sensor It operates based on the reflection of infrared light and can detect objects within a certain range with only four pins: Vcc, Gnd, EN, and out.

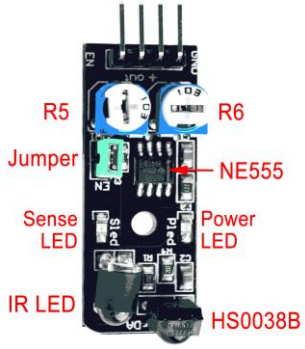

Figure 1-2 Keyestudio KY-032

The IR led sent infrared light when there is no detection the IR light doesn't reflect, when an object enters the sensor's detection range, the emitted IR light reflects off the object and is received by the IR sense led, than the sensor's output pin changes state.

Some of it Manufacturer's Specifications [3]:

- Working voltage: 3.3V to 5VDC
- Working current: ≥ 20mA
- Operating temperature: -10°C to +50°C
- Detection distance: 2 to 40cm
- IO Interface: 4-pin (EN /out / + / GND)
- Output signal: TTL level
    - LOW level if obstacle detected
    - HIGH if no obstacle detected
- Adjustment: two single-turn variable resistors
- Effective angle: ±35°
- Size: 28mm × 23mm
- Weight: 9g

HD44780 LCDs are famous types of LCDS used for displaying, it has theses pins V0,VSS,VDD,RS,RW,E,D0-D7,A,K and it has two operating mode using 8 bits D0-D7 or 4 bits D4-D7.

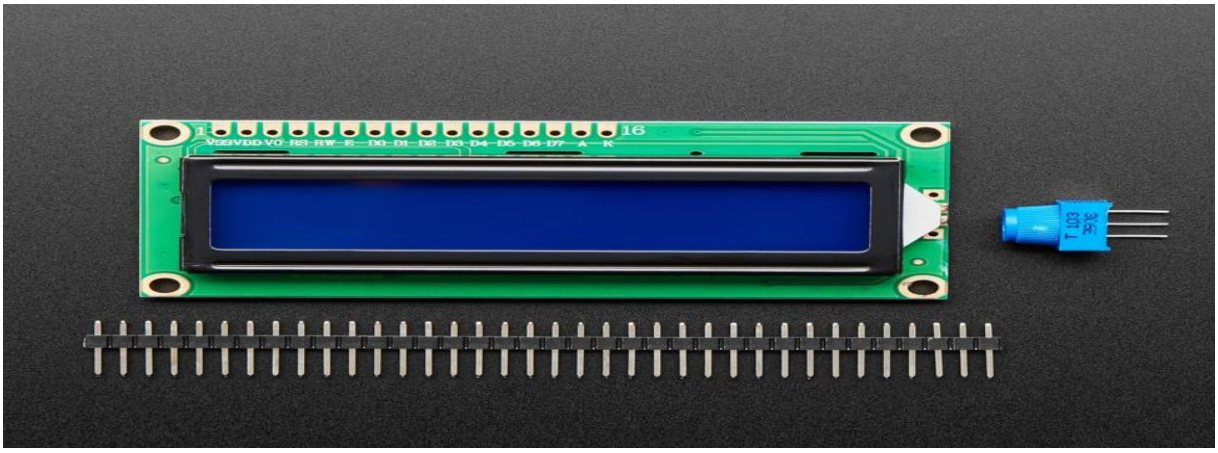

Figure 1-3 LCD

The way of connection as follow:

- VSS with GND
- VDD with Vcc
- V0 with potentiometer
- RS with pin(0 data treated as command 1 data treated as text)
- RW with GND since we want the LCD to write
- E to a pin (it should be zero when sending data)
- D0-D4 with pins
- A with Vcc
- K with GND

In 4 pins push button pin 1 is always internally connected to pin 2 and same goes for pin 3 and pin 4 and to use push button:

Pin 1 – 3 or Pin 1 – 4 or Pin 2 – 3 or Pin 2 – 4

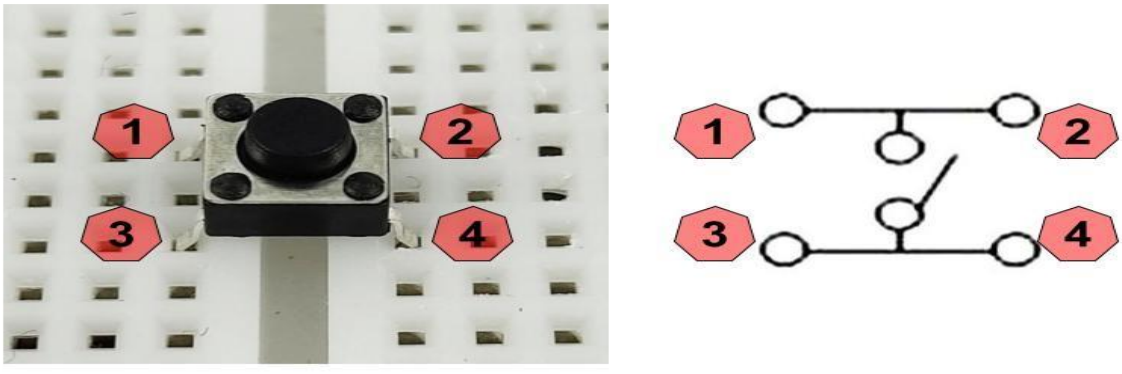

Figure 1-4 push buttons

2. Procedure

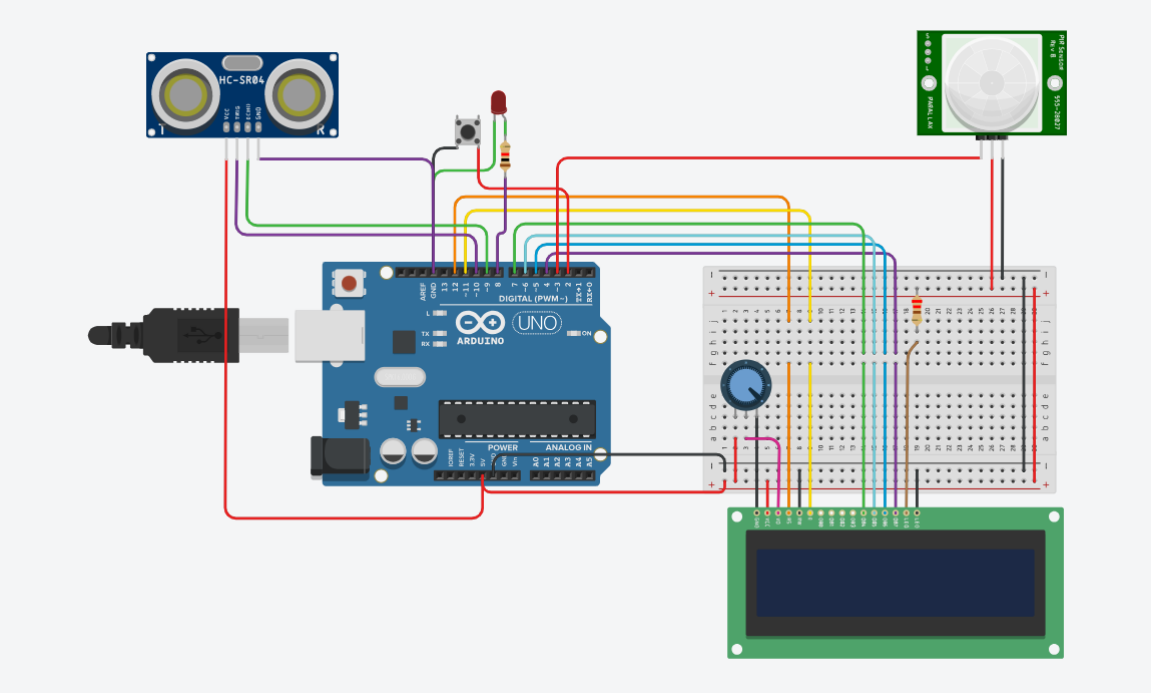

Figure 2-1 circuit

The circuit was connected as in fig(2-1)

The microcontroller was programmed on Arduino IDE

Code Clarification

The code can be found in the appendix

Initialization:

The pins for the ultrasonic sensor (trigPin and echoPin) ,the LED (led1Pin), IR(irPin)and button(buttonPin) were defined.

A volatile Boolean ledOn used for toggle the led, volatile detectionTime for store that last time when an object is detected and constant debounceDelay used to eliminate false signals that can occur when a mechanical button is pressed.

Setup:

Serial communication for debugging purposes was initialized.

Attach interrupt for button and IR.

Begin the LCD.

The pins were set as either INPUT or OUTPUT as required.

Loop:

If the last detection passes 3 second the LCD is cleared.

a short pulse to the ultrasonic sensor was sent to trigger a measurement.

the duration of the echo pulse was measured using pulseIn() function and the distance based on the time it took for the pulse to return was calculated and convert it to string and print it on the LCD and Serial Monitor.

Button Interrupt:

A static variable is declared to save the last interrupt time, with a condition checks if the time difference between the current interrupt and the last interrupt is greater than the debounceDelay the led is toggled.

IR_ISR Interrupt:

The detection time is updated and "object detected "is displayed on the LCD.

Based on the requirements:

- Display the distance.

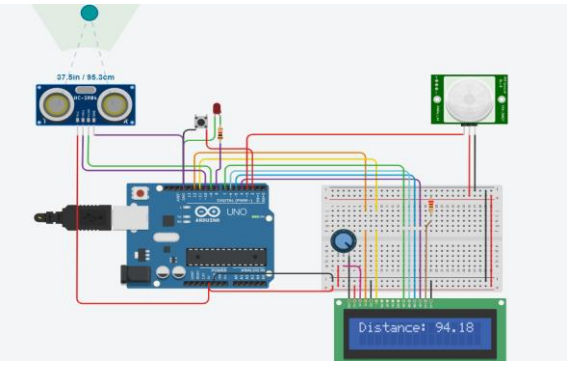

Figure 2-2 Display distance

- Display object detected and clear the LCD after 3 seconds.

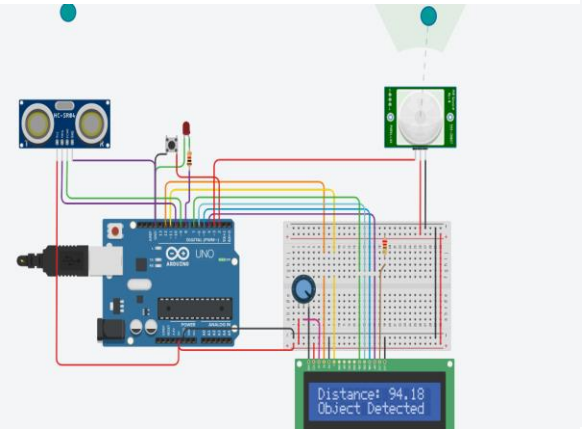

Figure 2-3 Detect object

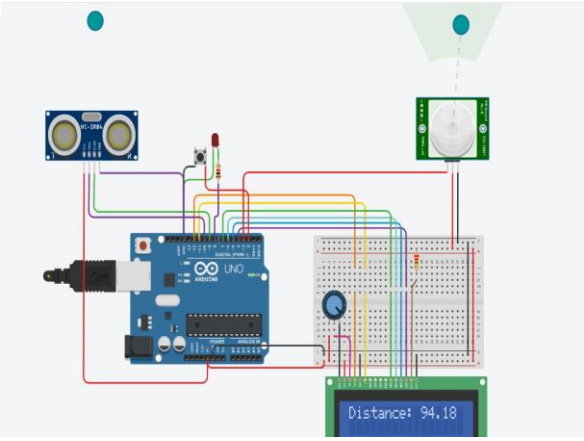

Figure 2-4 Clear LCD after 3 seconds

- Button interrupt and toggle the LED

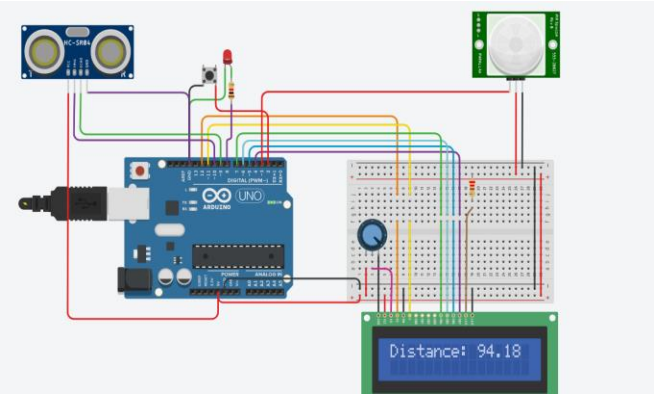

Figure 2-5 First time click the button

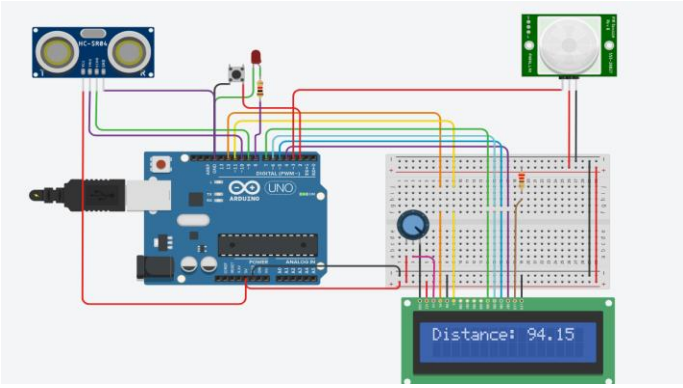

Figure 2-6 Second time click the button

Conclusion

To sum up, this project uses Arduino, ultrasonic sensors, LED, IR sensor and push button to measure distance, detect object motion and toggle the led states when button pressed.it improve our skill with dealing with sensors and using interrupts.

References

[1] Avr Atmega Ultrasonic Module Hc Sr04 Interfacing With Atmega1632 ... (electronicwings.com)

[2] Arduino Ultrasonic distance sensor - JavaTpoint

[3] http://irsensor.wizecode.com/

Appendix

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 7, 6, 5, 4); // LCD pins


const int buttonPin = 2;
const int irPin = 3;
const int ledPin = 8;
const int echoPin = 9;
const int triggerPin = 10;


volatile bool ledOn = false;
volatile unsigned long detectionTime = 0;
const unsigned long debounceDelay = 50;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  digitalWrite(buttonPin, HIGH);      // Enable pull-down resistor
  pinMode(irPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);


  attachInterrupt(digitalPinToInterrupt(buttonPin), buttonInterrupt, FALLING);
// Attach interrupt on button pin
  attachInterrupt(digitalPinToInterrupt(irPin), IR_ISR, FALLING);
  lcd.begin(16, 2);
  Serial.begin(9600);
}

void loop() {
  if (millis()-detectionTime>3000){
    lcd.setCursor(0, 1);
      lcd.print("                ");
  }

  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
```

```
  // Measure the duration of the pulse on echoPin
  long duration = pulseIn(echoPin, HIGH);

  // Calculate distance in centimeters (you can use inches by changing the
conversion factor)
  float distance_cm = duration * 0.034 / 2;

  // Convert the float to a string
  char buffer[10]; // Adjust the buffer size as needed
  dtostrf(distance_cm, 5, 2, buffer); // 5 is the minimum width, 2 is the
number of decimal places
  Serial.println(buffer);
  Serial.println(distance_cm);

  // Print the distance to the serial monitor
  lcd.setCursor(0, 0);
  lcd.print("                ");
  lcd.setCursor(0, 0); // Set cursor to the first row
  lcd.print("Distance: ");
  lcd.print(buffer);
  // Wait a bit before taking the next measurement
  delay(3000);

}
void buttonInterrupt() {
  static unsigned long lastInterruptTime = 0;
  unsigned long interruptTime = millis();

  // If interrupts come faster than debounceDelay, assume it's a bounce and
ignore
  if (interruptTime - lastInterruptTime > debounceDelay) {
    if (ledOn) {
      ledOn = false;
      digitalWrite(ledPin, LOW);
    } else {
      ledOn = true;
      digitalWrite(ledPin, HIGH);
    }
  }
  lastInterruptTime = interruptTime;
}


void IR_ISR() {
  detectionTime = millis();
```

```
  lcd.setCursor(0,1);
  lcd.print("Object Detected");
}
```