DISTANCE MEASUREMENT AND OBJECT DETECTION USING ULTRASONIC SENSORS WITH RASPBERRY PI PICO W

TEAM MEMBERS:

Hilary Royson CB
Nitish Kumar J
Allan christ B
Ahash J

Aim:

To measure the distance and detect objects using ultrasonic sensors with Raspberry Pi Pico W.

Tools / Hardware Required:

- Raspberry Pi Pico W
- Ultrasonic Sensor HC-SR04
- LED
- Buzzer
- Resistor
- OLED
- Bread Board
- Battery 5V

Theory:

A distance measuring system is used to measure the distance between two objects by transmitting high-frequency waves and calculating the time taken for the echo to return. In this project,

- **Raspberry Pi**
  - It is a low-cost, high-Performance micro Controller board.
- **Ultrasonic Sensor (e.g., HC-SR04)**
  - A sensor that uses high-frequency sound waves to measure distance.
  - Can detect objects and calculate their distance from the sensor.
  - The sensor has TRIG and ECHO pins: TRIG sends the signal, and ECHO receives the reflected pulse.
  - Range varies between 2cm – 4m.

To calculate the distance, we use the following formula:

Distance = Time x 0.034/2 = 0.017m/s

The system can detect objects within a certain range and alert the user if an obstacle is too close.

Applications include obstacle avoidance, smart parking systems, and automation.

LED:

- A light emitting diode is a semiconductor diode which glows when a voltage is applied.\

Resistor:

- A resistor is an electronic Components that limits Components that limits or Controls The flow of electric Converting electrical energy into heat.

OLED:

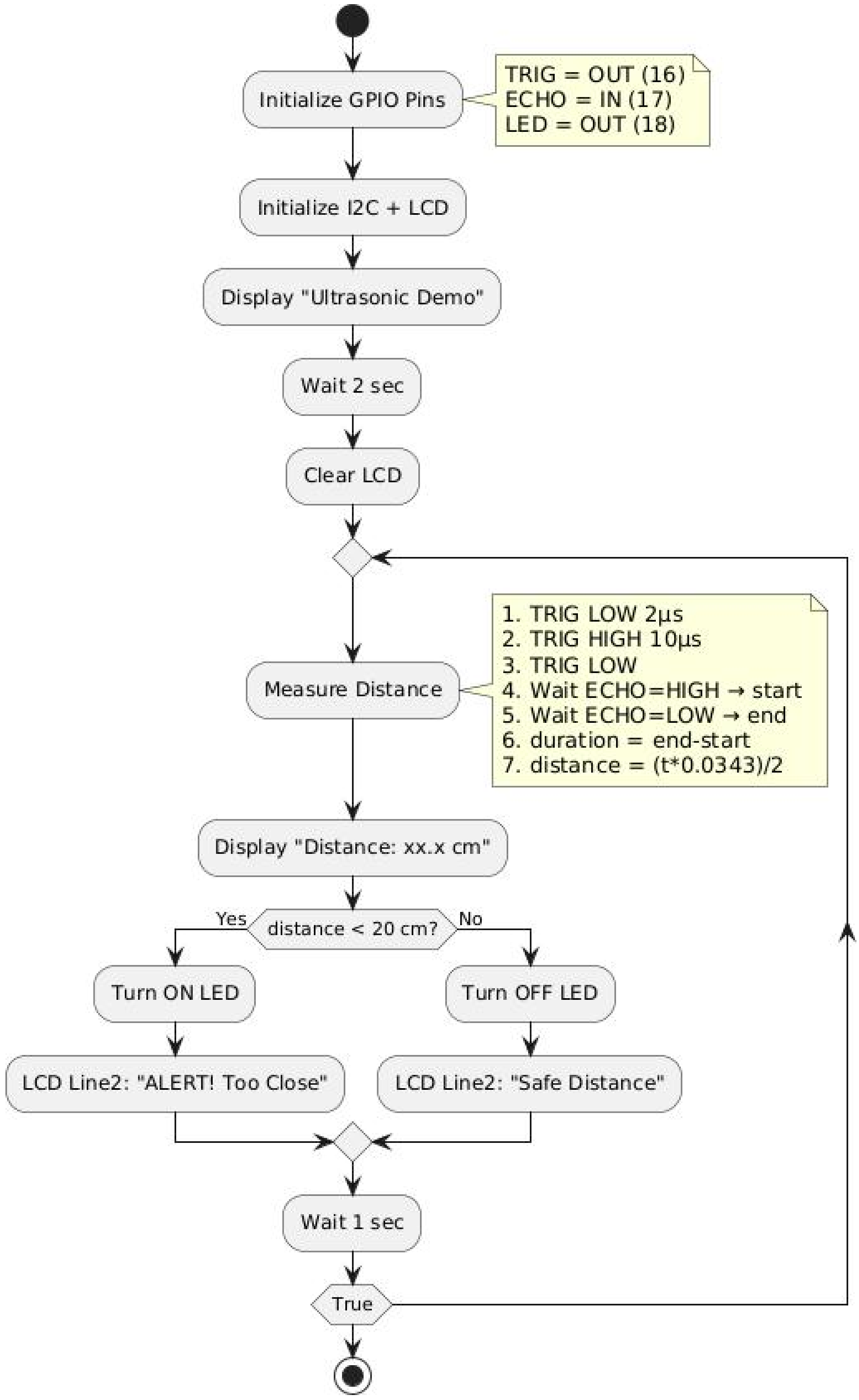
- OLED(Organic Light-emitting Diode) is a display technology that uses a series of organic (carbon-based) thin films to produce light. unlike Traditional LED-backlit LCD Screens, which require a separate light Sources, OLED pixels emit their Own light when emissive nature allows for exceptional Picture quality in high and devices

Pin Connections:

Component	Pico W Pin
HC-SR04 VCC	VBus

HC-SR04 GND	GND
HC-SR04 ECHO	Pin 16
HC-SR04 TRIG	Pin 17
LCD SDA	GPIO 0
LCD SCL	GPIO 1

Flowchart:


Initialize GPIO Pins
TRIG = OUT (16)
ECHO = IN (17)
LED = OUT (18)
Initialize I2C + LCD
Display "Ultrasonic Demo"
Wait 2 sec
Clear LCD
Measure Distance
1. TRIG LOW 2µs
2. TRIG HIGH 10µs
3. TRIG LOW
4. Wait ECHO=HIGH → start
5. Wait ECHO=LOW → end
6. duration = end-start
7. distance = (t*0.0343)/2
Display "Distance: xx.x cm"
distance < 20 cm?
Yes
No
Turn ON LED
Turn OFF LED
LCD Line2: "ALERT! Too Close"
LCD Line2: "Safe Distance"
Wait 1 sec
True

Program:

```
from machine import Pin, I2C
from time import sleep, sleep_us, ticks_us
import sys
from pico_i2c_lcd import I2cLcd

TRIG = Pin(16, Pin.OUT)
ECHO = Pin(17, Pin.IN)
LED = Pin(18, Pin.OUT)

i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
devices = i2c.scan()

if len(devices) == 0:
    print("No I2C device found!")
    sys.exit()
else:
    I2C_ADDR = devices[0]  # Auto-detect first device (commonly 0x27 or 0x3F)
    print("I2C device found at address:", hex(I2C_ADDR))

lcd = I2cLcd(i2c, I2C_ADDR, 2, 16)

def measure_distance():
    TRIG.low() sleep_us(2)
    TRIG.high()
    sleep_us(10)
    TRIG.low()

    while ECHO.value() == 0:
        start = ticks_us()
    while ECHO.value() == 1:
        end = ticks_us()

    duration = end - start
    distance = (duration * 0.0343) / 2  # cm
    return distance

threshold = 20  # cm

lcd.putstr("Ultrasonic Demo\nStarting...")
sleep(2)
lcd.clear()

while True:
    try:
        dist = measure_distance()
        lcd.clear()
        lcd.putstr("Distance:{:.1f}cm".format(dist))
        if dist < threshold:
            LED.high()
            lcd.move_to(0, 1)
            lcd.putstr("ALERT! Too Close")
        else:
            LED.low()
            lcd.move_to(0, 1)
            lcd.putstr("Safe Distance")

        sleep(1)

    except KeyboardInterrupt:
        lcd.clear()
        lcd.putstr("Program Stopped")
        sys.exit()
```

Result:

Thus using Raspberry Pi Pico W, the distance is measured and the object is detected.