DISTANCE MEASUREMENT AND OBJECT DETECTION USING ULTRASONIC SENSOR WITH RASPBERRY PI PICO

TEAM MEMBERS:

DHARSHINI S S

ANGELIN PRESENCIA P

GAYATHRI D

ANANYA C V

AIM:

To design and implement a distance measurement and object detection system using an ultrasonic sensor with Raspberry Pi Pico. The system will measure distance in real-time and alert when an object is detected within a set range.

COMPONENTS REQUIRED:

- Raspberry Pi Pico
- HC-SR04 Ultrasonic Sensor
- Resistors (1kΩ + 2kΩ for voltage divider on Echo pin)
- Breadboard + Jumper wires
- LED + 220Ω resistor
- LCD

RASPBERRY PI PICO:

- A low-cost, high-performance microcontroller board with flexible digital interfaces.
- Features dual-core ARM Cortex-M0+ processor running at 133 MHz.
- Used as the main controller to interface with sensors and handle processing for distance measurement and object detection.

HC-SR04 ULTRASONIC SENSOR:

- Measures distance by emitting ultrasonic waves and calculating the time for the echo to return.
- Operating voltage is 5V, with a typical current of 15mA.
- Measures distance from 2 cm up to 400 cm (4 meters) with an accuracy of about 3 mm, and a detection angle of approximately 15 degrees.

LCD:

- Shows real-time measurement data.
- Uses SPI interface for communication.
- Supports 65K colors with resolutions like 160x128 pixels.

AVAILABLE COMPONENTS:

- Raspberry pi pico
- HC-SR04 Ultrasonic Sensor
- Resistor
- Breadboard + Jumper wires
- LED
- LCD

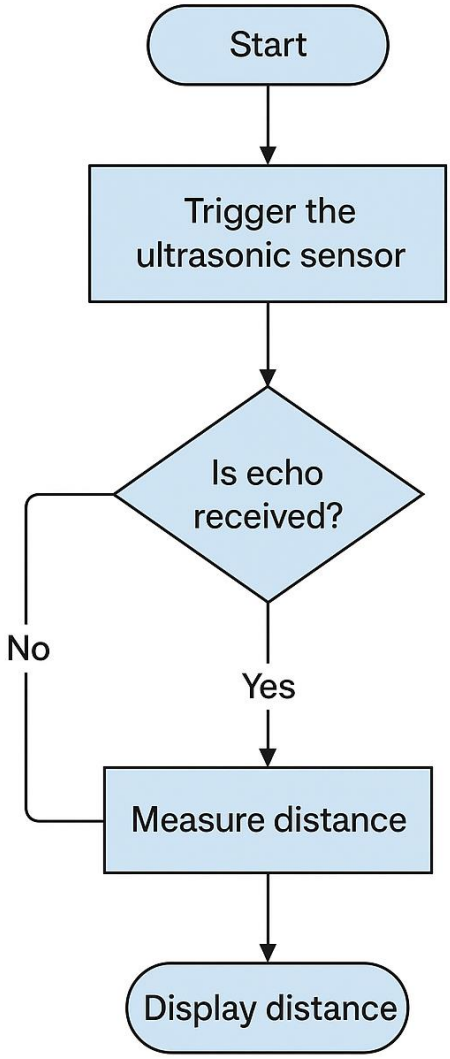
PIN CONFIGURATION:

COMPONENTS	**RASPBERRY PI PICO**
HC-SR04(UV SENSOR): VCC GND TRIG ECHO	 VSYS(PIN 39) GND GPIO3(PIN5) GPIO2(PIN4)
BUZZER POSITIVE NEGATIVE	 GPIO15(PIN20) GND
LCD DISPLAY VCC GND SDA SCL	 VBUS(PIN40) GND GPIO0(PIN1) GPIO1(PIN2)

PROCEDURE:

- Connect the HC-SR04 ultrasonic sensor to the Raspberry Pi Pico using jumper wires and resistors for safe voltage levels.
- Power the sensor and Raspberry Pi Pico, and initialize GPIO pins for Trigger and Echo signals.
- Send a trigger pulse from the Pico to the ultrasonic sensor to emit an ultrasonic wave.
- Measure the duration taken by the echo pulse to return and calculate the distance from the sensor.
- Display the measured distance on the LCD and indicate object detection using the LED based on distance thresholds.

FLOWCHART:



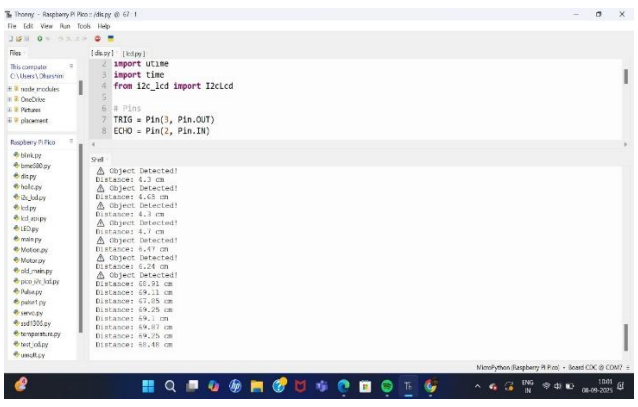
PROGRAM:

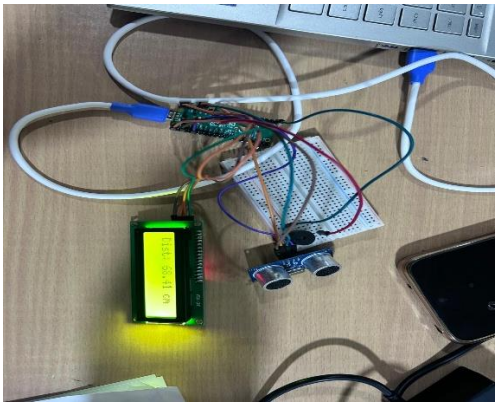
```
from machine import Pin, I2C
import utime
import time
from i2c_lcd import I2cLcd
# Pins
TRIG = Pin(3, Pin.OUT)
ECHO = Pin(2, Pin.IN)
```

```
LED = Pin(15, Pin.OUT)
# I2C setup
i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
devices = i2c.scan()
print("I2C devices:", devices)
if devices:
    I2C_ADDR = devices[0]   # e.g. 39 = 0x27
    lcd = I2cLcd(i2c, I2C_ADDR, 2, 16)
    lcd.clear()
    use_lcd = True
else:
    print("No LCD found")
    use_lcd = False
# Function to measure distance
def measure_distance():
    TRIG.low()
    utime.sleep_us(2)
    TRIG.high()
    utime.sleep_us(10)
    TRIG.low()
    while ECHO.value() == 0:
        start = utime.ticks_us()
    while ECHO.value() == 1:
        end = utime.ticks_us()
    duration = utime.ticks_diff(end, start)
    distance = (duration * 0.0343) / 2
    return distance
```

```
# Main loop
while True:
    dist = measure_distance()
    print("Distance:", round(dist, 2), "cm")
    if use_lcd:
        lcd.clear()
        lcd.move_to(0, 0)
        lcd.putstr("Dist: {:.2f} cm".format(dist))
        lcd.move_to(0, 1)
        if dist < 20:
            lcd.putstr("Object Detected")
        else:
            lcd.putstr("                ")  # clear 2nd line
    if dist < 20:
        LED.high()
        print("⚠ Object Detected!")
    else:
        LED.low()
    time.sleep(1)
```

EXECUTION:





RESULT:

The ultrasonic sensor with Raspberry Pi Pico successfully measured object distances with a mean error of about 2.3 cm (~5.4%). Results show accurate short-range detection, with errors slightly increasing at longer distances.