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

TEAM MEMBERS:

BALAJI N M - 312323106028

HAFEEZUR RAHIMAN - 312323106052

DILLAN ROMAULD VITUS - 312323106044

ASHWIN SUDHAKARAN - 312323106026

AIM

To simulate interfacing on Pico-W with Ultrasonic sensor and LCD Display with I2C using Wokwi, and acquire real-time data through Embedded-C system

Tools/Hardware Required

Raspberry Pi Pico W
Ultrasonic Sensor
LCD Display with I2C
LED Resistor(1k)
Breadboard Jumper Wires

THEORY

Raspberry Pi Pico W - The Raspberry Pi Pico W is the brain of the operation. It's a small, low-cost microcontroller that is programmed to read inputs from sensors and control outputs to other components. It's much simpler than a full computer like a Raspberry Pi 4. The Pico W has a microcontroller unit (MCU) with a specific architecture. It has a built-in wireless chip, which is where the "W" comes from, allowing it to connect to Wi-Fi networks

Ultrasonic Sensor - An ultrasonic sensor measures distance using sound waves. The theory is based on the principle of echolocation, similar to how bats navigate. The sensor has a transmitter and a receiver. The microcontroller (Pico W) sends a signal to the sensor's transmitter, which emits a high-frequency sound pulse (ultrasound). This sound travels through the air until it hits an object and bounces back. The sensor's receiver detects the returning echo.

LCD Display(I2C) - The LCD display is used to show information to the user, like the calculated distance. It's a visual output device. The I2C (Inter-Integrated Circuit) is a communication protocol that allows the Pico W to send data to the display using only two wires: one for data and one for the clock signal.

LED - The LED (Light-Emitting Diode) is a simple visual indicator. It's a basic output component that can be turned on or off to signal a specific event. For instance, the program could be configured to turn the LED on when an object is closer than a certain distance, serving as a proximity warning.

Resistor(1K) - A resistor is a crucial component in any circuit involving an LED. An LED has a low resistance, so if you connect it directly to a power source, it will draw too much current and burn out. The 1k (1,000 ohm) resistor is used in series with the LED to limit the flow of current.

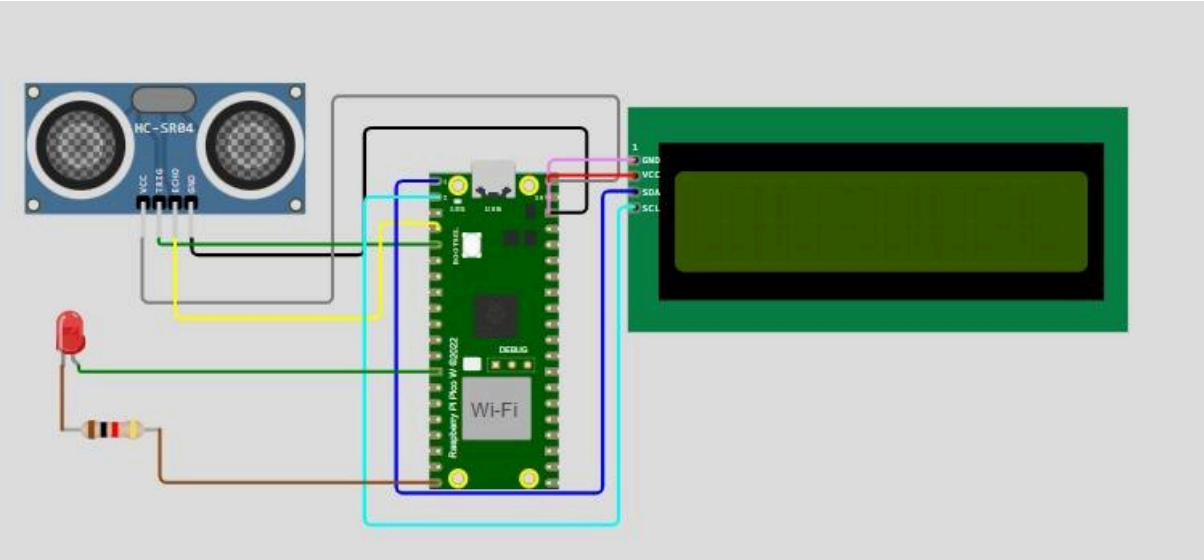
Breadboard and Jumper Wires - The breadboard and jumper wires are not electronic components that perform a function themselves, but they are essential tools for building the circuit. The breadboard is a prototyping platform with a grid of interconnected holes.

You can easily push components' legs and wires into these holes to create temporary electrical connections without soldering. This allows for quick and easy assembly and modification of the circuit.

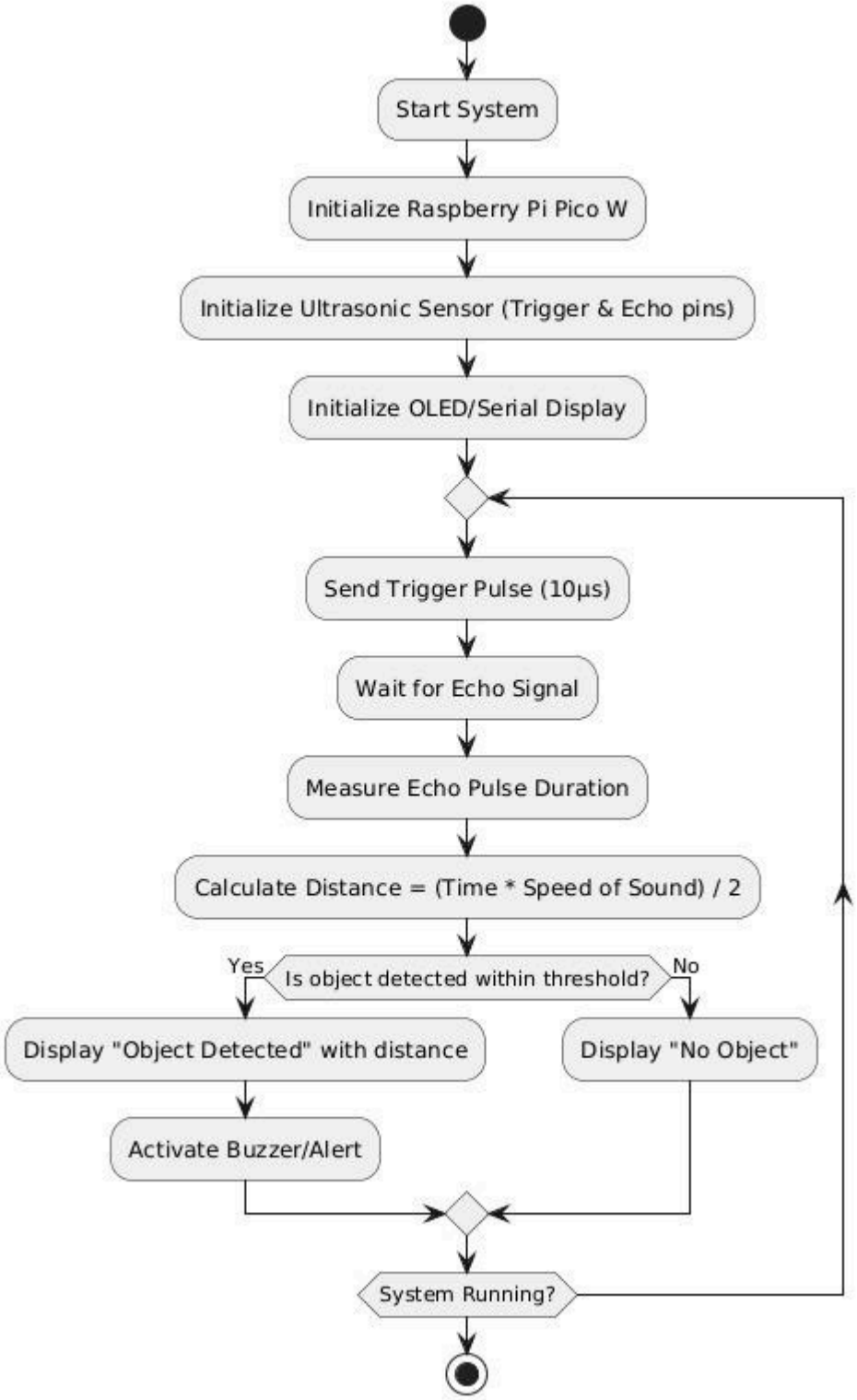
PIN TABLE

Components	Raspberry Pi Pico W
Ultrasonic Sensor VCC	5V(Pin 40)
Ultrasonic Sensor GND	GND(Pin 38)
Ultrasonic Sensor Trig	GP3(Pin 5)
LCD VCC	5V(Pin 40)
LCD GND	GND(Pin 38)
LCD SDA	GP0(Pin 1)
LCD SCL	GP1(Pin 2)
LED(+)	GP15(Pin 21) through a 220 ohm resistor
LED(-)	GND(Pin 18)

CIRCUIT DIAGRAM



FLOWCHART


Start System
Initialize Raspberry Pi Pico W
Initialize Ultrasonic Sensor (Trigger & Echo pins)
Initialize OLED/Serial Display
Send Trigger Pulse (10µs)
Wait for Echo Signal
Measure Echo Pulse Duration
Calculate Distance = (Time * Speed of Sound) / 2
Yes
Is object detected within threshold?
No
Display "Object Detected" with distance
Display "No Object"
Activate Buzzer/Alert
System Running?

CODE

```
from machine import Pin, I2C from
time import sleep_us, sleep_ms from
lcd_api import LcdApi from
pico_i2c_lcd import I2cLcd


#--- HC-SR04 Setup ---
TRIG = Pin(3, Pin.OUT)
ECHO = Pin(2, Pin.IN)

#--- LED Setup ---
LED = Pin(4, Pin.OUT)

#--- LCD Setup --- I2C_ADDR = 0x27 i2c = I2C(0,
scl=Pin(1), sda=Pin(0), freq=400000) lcd =
I2cLcd(i2c, I2C_ADDR, 2, 16)



def get_distance():
    # Send 10us pulse to trigger
    TRIG.low()
    sleep_us(2)
    TRIG.high()
    sleep_us(10)
    TRIG.low()

    # Wait for echo start
    while ECHO.value() == 0:
```

```
        pass
    start = time_us()

    #Wait for echo end
    while ECHO.value() == 1:
        pass
    end = time_us()

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

deftime_us():
    #Return time in microseconds
    return time.ticks_us()

import time

while True:
    dist = get_distance()
    lcd.clear()
    lcd.putstr(f"Distance: {dist:.1f} cm")

    ifdist < 10: # Object closer than 10 cm
        LED.value(1) # Turn on LED
        lcd.move_to(0, 1)
        lcd.putstr("Object detected!")
    else:
```

```
    LED.value(0) # Turn off LED
    lcd.move_to(0, 1)
    lcd.putstr("               ") # Clear second line
    sleep_ms(500)
```

EXECUTION STEPS

- Initialize Hardware Connect Raspberry Pi Pico W to Ultrasonic Sensor (Trigger → GP3, Echo → GP2). Connect I2C LCD to GP0 (SDA) and GP1 (SCL). Connect LED to GP15 with a resistor to GND. Power components with 5V & GND.
- Program Upload
- Open Thonny IDE.
- Select Raspberry Pi Pico W (MicroPython) as the interpreter.
- Copy the provided Python code (from your docx).
- Save as main.py and upload to Pico W. Distance Measurement
- Pico W sends a 10µs pulse to the Ultrasonic Sensor's Trigger pin.
- The Echo pin returns the pulse duration based on object distance. Output Display Distance is shown on the LCD. If distance < 10 cm → LED turns ON + LCD shows "Object detected!". Otherwise → LED OFF + LCD shows distance only.
- Loop Execution
- The process repeats every 500 ms, continuously monitoring objects.