Ex. No. 9: Home Automation Using IR Sensor, DHT11, and Ultrasonic Sensor with Raspberry Pi Pico

1. AIM:

To design and implement a home automation system using an IR sensor, DHT11 temperature and humidity sensor, and an ultrasonic sensor interfaced with Raspberry Pi Pico to automate household appliances based on sensor inputs.

2. REQUIREMENTS:

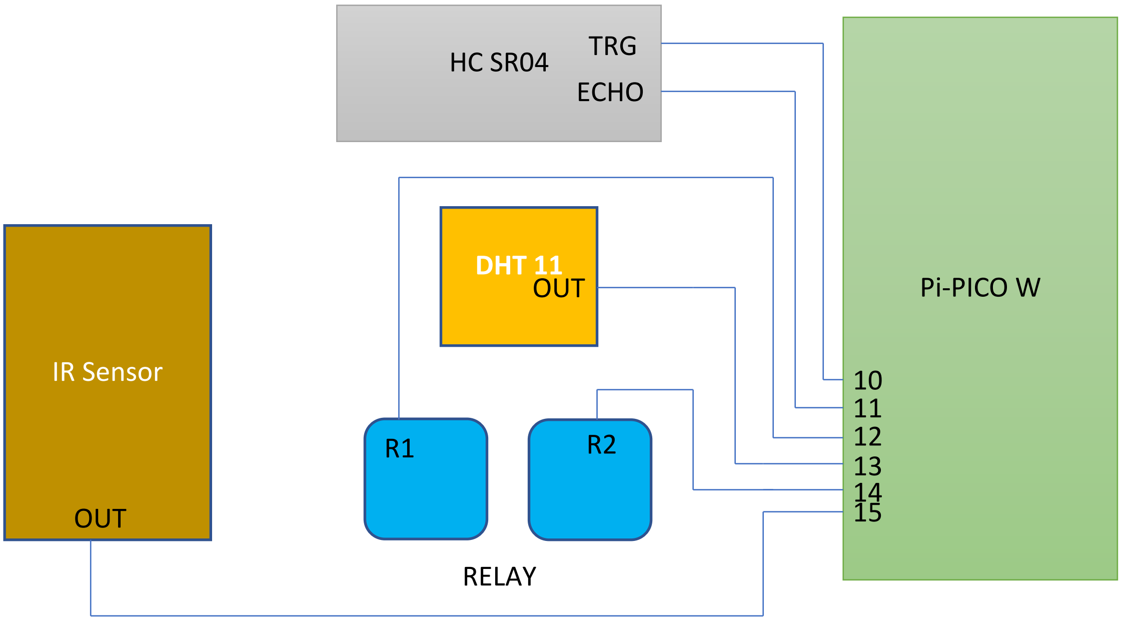
Hardware:

- Raspberry Pi Pico
- IR Sensor
- DHT11 Temperature and Humidity Sensor
- Ultrasonic Sensor (HC-SR04)
- Relay Module
- LED
- Jumper wires
- Power supply

Software:

- MicroPython
- Thonny IDE

3. CIRCUIT DIAGRAM:



- **IR Sensor** connected to GPIO 15 (Digital Input)
- **DHT11 Sensor** connected to GPIO 13
- **Ultrasonic Sensor** connected to GPIO 10 (Trigger) and GPIO 11 (Echo)
- **Relay for IR Control** connected to GPIO 14
- **Relay for Temperature Control** connected to GPIO 12
- **LED** connected to GPIO 16 (PWM)

4. PROCEDURE:

1. Connect the sensors and relays to the Raspberry Pi Pico as per the circuit diagram.
2. Write the MicroPython code to read sensor data and control relays based on sensor outputs.
3. Upload the code to the Raspberry Pi Pico using Thonny IDE.
4. Implement the following automation:
   - IR Sensor → Activates relay when an object is detected.
   - DHT11 Sensor → Turns on relay when temperature exceeds a threshold (29°C).
   - Ultrasonic Sensor → Measures distance and displays the value on the console.
5. Test the system under different conditions and record observations.

5. CODE:

```
import machine as gpio
import utime as TM
import dht

# IR Sensor and Relay Control
Sensor_In = gpio.Pin(15, gpio.Pin.IN)
Relay_IR = gpio.Pin(14, gpio.Pin.OUT)

# Temperature Sensor and Relay Control
dht11_pin = gpio.Pin(13)
dht11_sensor = dht.DHT11(dht11_pin)
Relay_Temp = gpio.Pin(12, gpio.Pin.OUT)

# Ultrasonic Sensor for Distance Measurement
trigger_pin = gpio.Pin(10, gpio.Pin.OUT)
echo_pin = gpio.Pin(11, gpio.Pin.IN)

# LED Control
LED = gpio.PWM(gpio.Pin(16))
LED.freq(1000)

def read_ir_sensor():
    if Sensor_In.value() == 1:
        Relay_IR.value(1)
        print("IR sensor detected an object. Relay ON.")
    else:
        Relay_IR.value(0)
        print("IR sensor did not detect an object. Relay OFF.")

def read_dht11_data():
    try:
        dht11_sensor.measure()
        temperature = dht11_sensor.temperature()
        humidity = dht11_sensor.humidity()
        print("Temperature: {:.2f}°C".format(temperature))
        print("Humidity: {:.2f}%".format(humidity))
        if temperature >= 29:
            Relay_Temp.value(1)
            print("Temperature exceeds 29°C. Relay ON.")
        else:
            Relay_Temp.value(0)
            print("Temperature below 29°C. Relay OFF.")
    except OSError as e:
        print("Error reading DHT11:", e)

def measure_distance():
    trigger_pin.value(1)
    TM.sleep_us(10)
    trigger_pin.value(0)

    while echo_pin.value() == 0:
        pass
    start_time = TM.ticks_us()

    while echo_pin.value() == 1:
        pass
    end_time = TM.ticks_us()

    pulse_duration = TM.ticks_diff(end_time, start_time)
    distance_cm = (pulse_duration / 2) / 29.1
    print("Distance: {:.2f} cm".format(distance_cm))
    return distance_cm

while True:
    read_ir_sensor()
    read_dht11_data()
    measure_distance()
    TM.sleep(2)
```

6. OBSERVATION:

The following observations were recorded during the execution of the system:

Sensor	**Condition Tested**	**Expected Result**	**Actual Result**
IR Sensor	Object detected	Relay ON	Relay ON
IR Sensor	No object detected	Relay OFF	Relay OFF
DHT11 Sensor	Temperature > 29°C	Relay ON	Relay ON
DHT11 Sensor	Temperature < 29°C	Relay OFF	Relay OFF
DHT11 Sensor	Humidity changes	Display correct humidity	Display correct humidity

Ultrasonic Sensor	Distance within range	Correct value displayed	Correct value displayed
Ultrasonic Sensor	No object in range	Display out of range	Display out of range

7. RESULT:

The home automation system using IR, DHT11, and ultrasonic sensors with Raspberry Pi Pico was successfully implemented. The relays were activated based on sensor inputs, and the system responded in real time to environmental changes. The project demonstrates the potential of home automation using Raspberry Pi Pico and various sensors. The system can be enhanced by adding IoT-based remote monitoring and control features for improved efficiency and user convenience.

8. APPLICATIONS:

- Smart home automation
- Automatic door opening and closing systems
- Temperature-controlled fan or AC systems
- Object detection for security systems
- Water tank level monitoring with automatic refilling