SMART PARKING

Control the HC-SR04 Ultrasonic sensor with ESP32

Introduction

- The **HC-SR04** is an ultrasonic distance sensor that can be used to measure the distance to an object by sending out a sound wave and measuring the time it takes for the sound wave to bounce back.
- The **ESP32** is a microcontroller with built-in WiFi and Bluetooth capabilities, which can be programmed using the Micropython. The **SSD1306** is a monochrome OLED (Organic Light Emitting Diode) display that can be used to display text and graphics.

Necessary components

ESP32

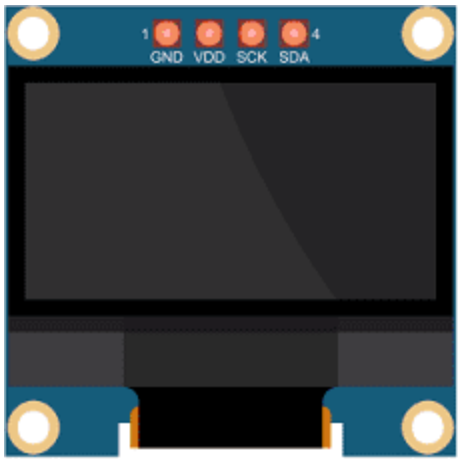
- The **ESP32** is a low-cost, low-power microcontroller with built-in Wi-Fi and Bluetooth capabilities.
- It is a popular choice for IoT projects and is commonly used for a variety of applications such as home automation, wireless control, and sensor data logging.
- The **ESP32** features a dual-core processor, a rich set of peripherals, and support for a wide range of protocols.
- It can be programmed using the Arduino IDE and various other programming languages such as C, C++, and MicroPython.
- Additionally, the **ESP32** has a wide range of features including:

HC-SR04 sensor:

- The **HC-SR04** is an ultrasonic distance sensor that can be used to measure the distance to an object by sending out a sound wave and measuring the time it takes for the sound wave to bounce back.
- It uses the principle of SONAR (Sound Navigation and Ranging) to detect the distance of an object.
- The **HC-SR04 sensor** has four pins: Vcc, Trig, Echo, and GND. The Vcc pin is used to provide power to the sensor, typically 5V.

- The Trig pin is used to trigger the sensor to send out a sound wave. The Echo pin is used to receive the sound wave that bounces back from the object. The GND pin is used to ground the circuit.

SSD1306 display:

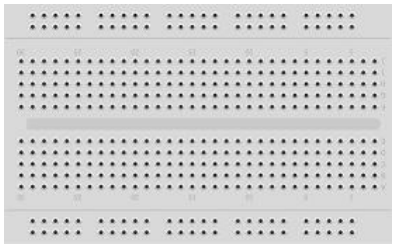


- The **SSD1306** is a monochrome OLED (Organic Light Emitting Diode) display that can be used to display text and graphics.
- It is a small, low-power display that can be easily integrated into a wide variety of projects.
- The **SSD1306** display is controlled using an I2C or SPI interface, which allows it to communicate with a microcontroller or microprocessor.
- It typically requires only a few connections to the microcontroller, such as Vcc, GND, SDA and SCL for I2C or SCK, MOSI, CS, DC and RST for SPI.
- The **SSD1306** display has a resolution of 128×64 pixels, and it can display text and graphics using a monochrome pixel format.
- This allows for low power consumption and makes it suitable for battery-powered projects.

Jumper wires:

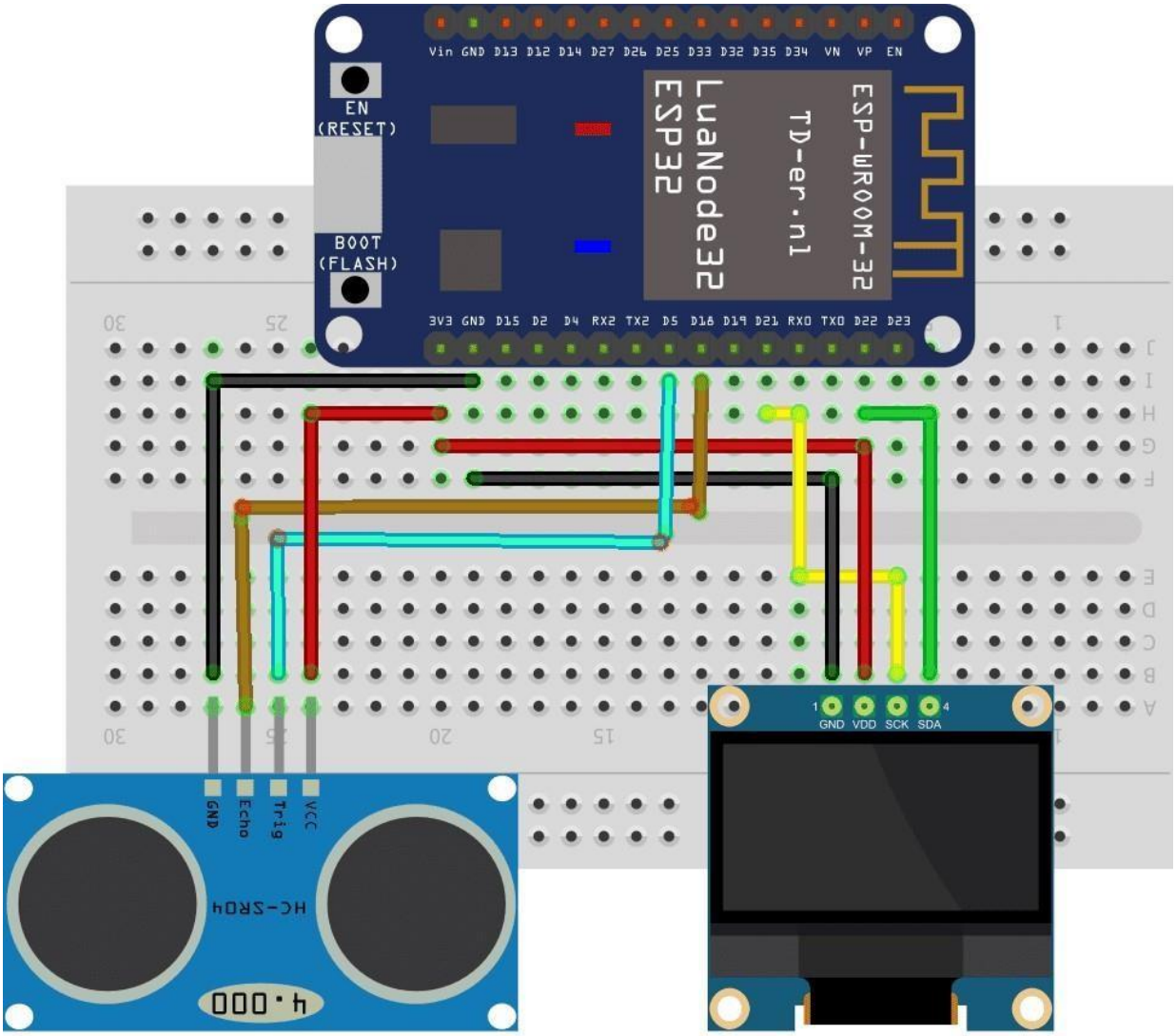
- **Connecting wires** refers to the process of physically **connecting wires** or cables to a device or circuit in order to establish an electrical connection.
- This can be done by using various connectors such as plugs, sockets, or terminal blocks. The wires are typically color-coded to indicate their function, such as red for power, black for ground, and yellow for signals.

Bread board:

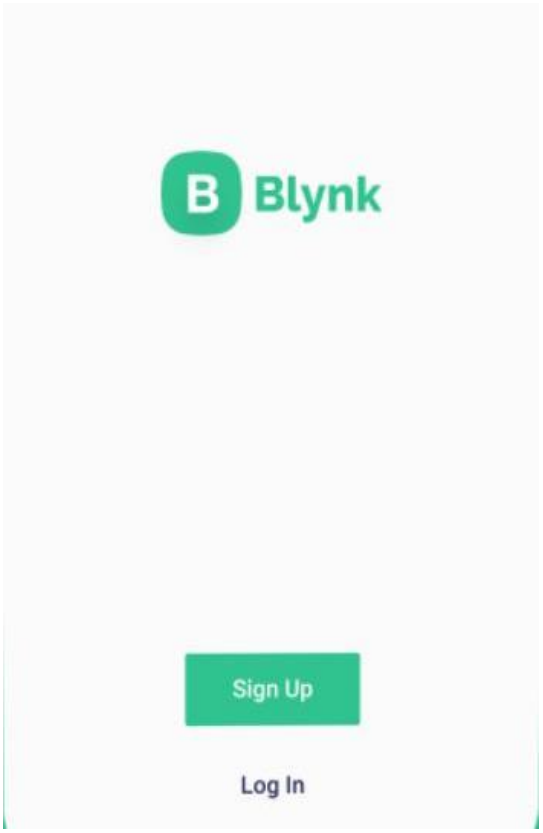


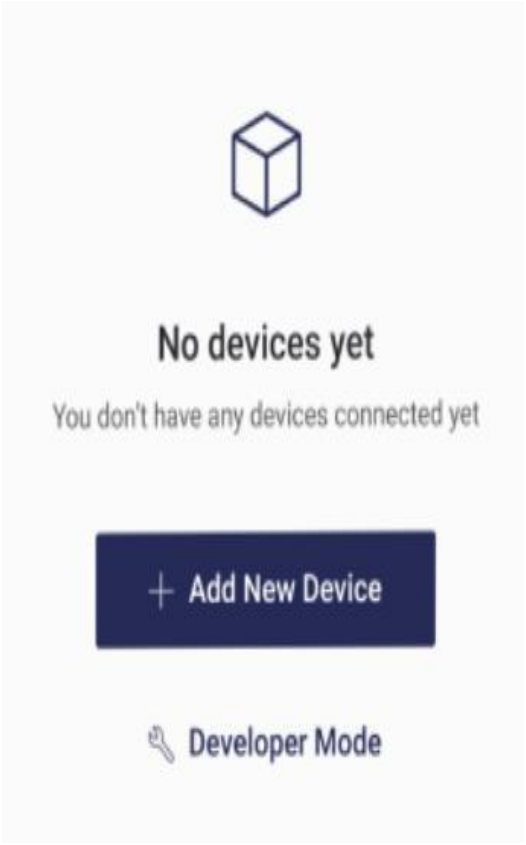
- A **bread board** is a type of circuit board that is used to test electronic components. It typically consists of a flat board made of a non-conductive material, such as plastic or fiberglass, with a number of holes or pads that are used to connect electronic components.
- The **bread board** allows you to connect electronic components and test them easily.

Circuit diagram:



Cloud setup





- Download Blynk application from play store and create an account in it
- After creation of account click add devices and select Esp32 and set the module and download the libraries from it
- Once the package is installed then import it in your code.
- By this open source cloud app blynk will used to give the output and status of the car parking system .
- Add your ssid and password of blynk in your code before extracting it.

Mounting:

The assembly is disconcertingly simple:

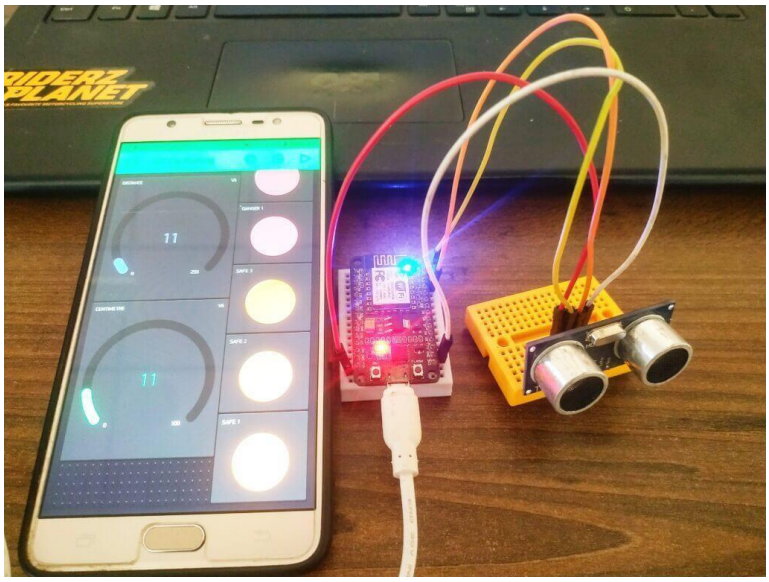
For HC-SR04 sensor :

- The 3.3V power supply from the ESP32 board goes to the VCC pin of the sensor.
- The GND pin of the ESP32 board goes to the GND pin of the sensor.
- Pin D5 of the ESP32 board goes to the TRIGGER pin of the sensor.
- Pin D18 of the ESP32 board goes to the ECHO pin of the sensor.

For SSD1306 display:

- the SCL pin to pin D22 of the ESP32 board
- the VCC pin to the 3.3V pin of the ESP32 board
- the GND pin to the GND pin of the ESP32 board
- the SDA pin the D21 pin of the ESP32 card

Model:



Program:

the program that displays the distance in cm that separates the ultrasonic sensor from an obstacle.

```
from hcsr04 import HCSR04
from machine import Pin,I2C
import ssd1306,time
import blynk
i2c = I2C(scl=Pin(22), sda=Pin(21)) #Init i2c
oled=ssd1306.SSD1306_I2C(128,64,i2c,0x3c)
sens
or =
HCSR04(trigger_pin=5,echo_pin=18,echo_timeout_us=1000000)

while True:
 distance = sensor.distance_cm() print(distance,' cm')
 time.sleep_ms(100)
 oled.fill(0) oled.text("Distance:",30,20)
 oled.text(str(distance),30,40) oled.text("cm",30,50)
 oled.show() #display the distance between the sensor and a
detected obstacle
```

Output

```
The distance from object is 236.327 cm
The distance from object is 238.5051 cm
The distance from object is 241.1976 cm
The distance from object is 243.4442 cm
The distance from object is 245.5194 cm
The distance from object is 247.6974 cm
The distance from object is 249.8926 cm
```

```
The distance from object is 89.16286 cm
The distance from object is 91.44381 cm
The distance from object is 93.50181 cm
The distance from object is 95.6627 cm
The distance from object is 97.84076 cm
The distance from object is 100.156 cm
The distance from object is 102.3169 cm
```

Output from blynk