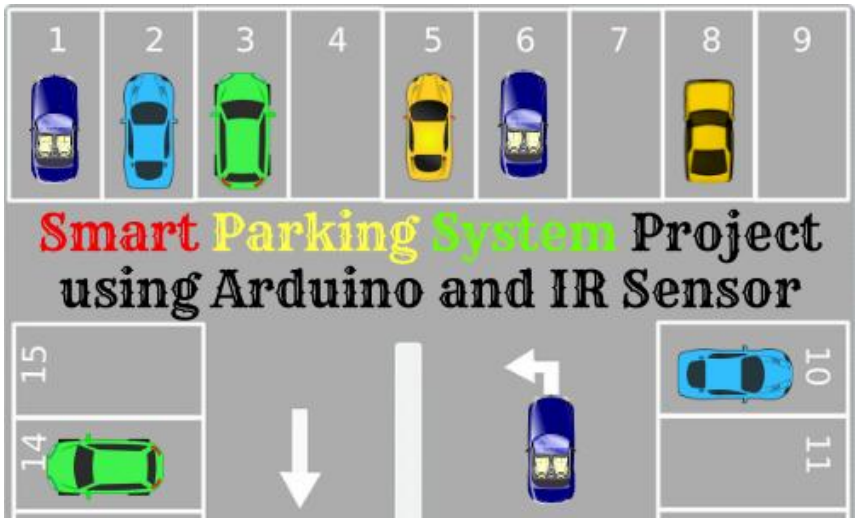
Smart Parking System

INTRODUCTION

- A car park is a designated area or building where cars and other vehicles can be parked and left temporarily.
- This project deals with an effective way of **finding empty spaces** and managing the number of vehicles moving in and out in complex multi storey parking structures by detecting a vehicle using IR sensors and thus providing a feedback.
- This automated system is used to find the vacancy in parking spaces available and navigate the driver to reach the desired space using visuals and in an effective manner, thus reducing search time.
- This system is required for malls, multistory parking structures, IT hubs and parking facilities.
- A smart car parking system gives a visual output indicating an available parking space rather than driving aimlessly. The driver looks up to the row of LED lights and their colour to deduct a result of determining the parking space availability.



Hardware / Component Requirements

➢ Servo Motor:-

A servomotor is a **rotary actuator** or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.

Features:

- Operating Voltage is +5V typically
- Rotation : 0°-180°

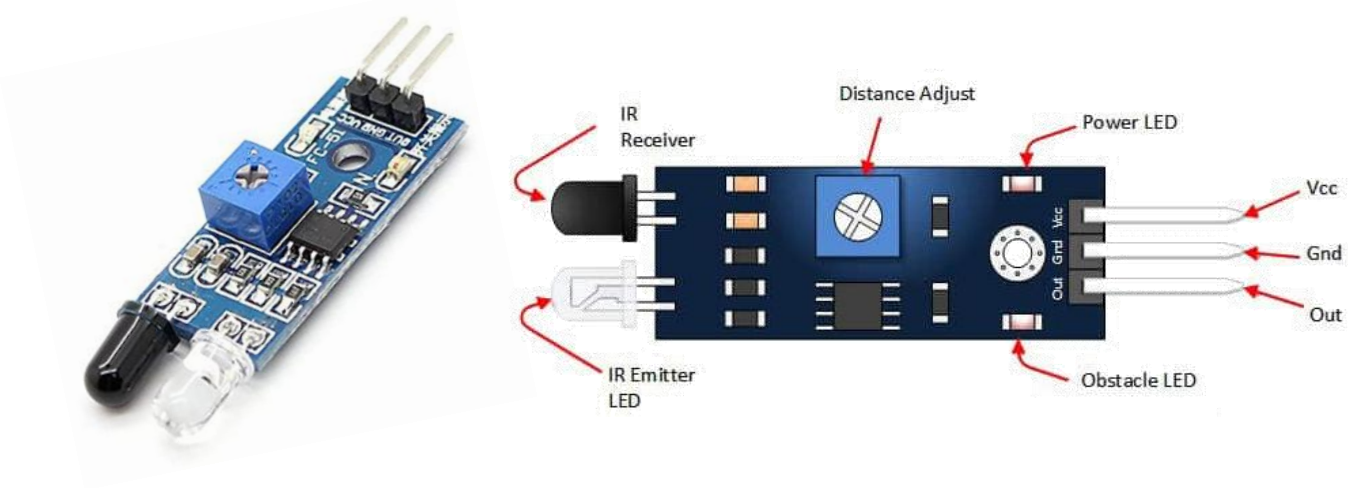
Servo Motor Wire Configuration:

Wire Number	Wire Colour	Description
1	Brown	Ground wire connected to the ground of system
2	Red	Powers the motor typically +5V is used
3	Orange	PWM signal is given in through this wire to drive the motor

➢ IR sensor:

IR, i.e., Infrared, detects the presence of an object by emitting a beam of infrared light.

It works similarly to ultrasonic sensors, though instead of using sonic waves, IR is transmitted. Infrared proximity sensors consist of an IR LED that emits, and a light detector for detection of reflection.



Features:

- Operating Voltage is 3.3V.
- Detection range is 2cm – 30cm.
- Active output level is “0” when an obstacle is detected.

IR Hardware Connections:

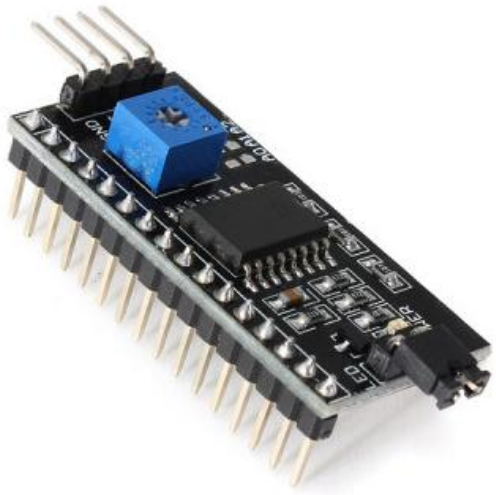
- **Vcc** to **5V** of Nodemcu.
- **GND** to **GND** of Nodemcu.
- **Out** to any **digital pin** of Nodemcu.

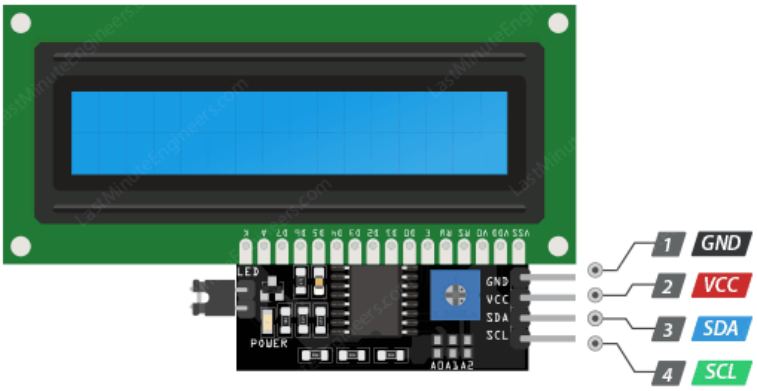
➢ LCD:

LCD (Liquid Crystal Display) is a type of flat panel display which uses liquid crystals in its primary form of operation. 16 x 2 LCD is a perfect I2C LCD display for Arduino and Raspberry Pi with high contrast and easy deployment. **16x2 means** two lines and **each line has 16 columns, 32 characters** in total.

➢ I2C Module:

I2C Module has an inbuilt PCF8574 I2C chip that converts I2C serial data to parallel data for the LCD display. These modules are currently supplied with a default I2C address of either 0x27 or 0x3F.



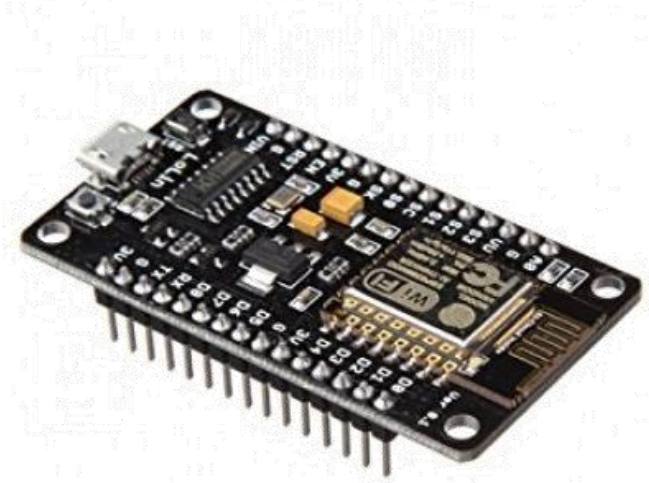


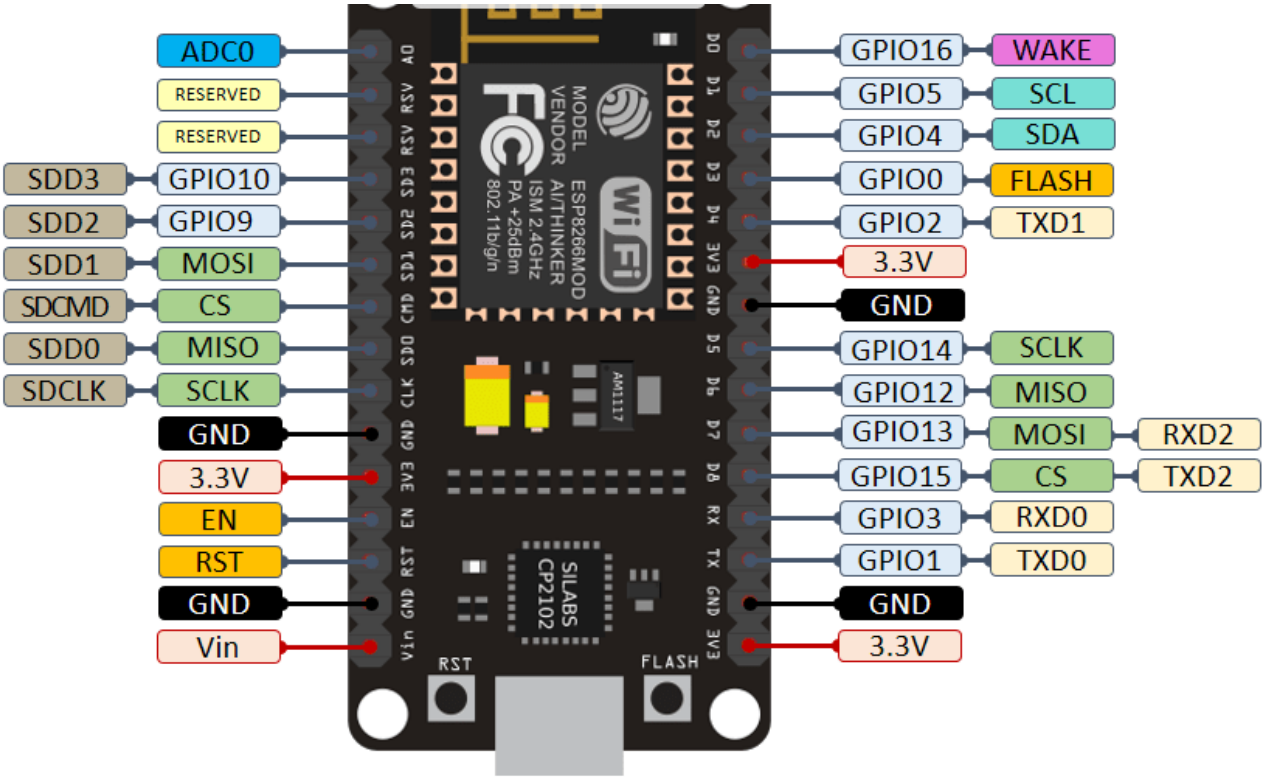
I2C Module Hardware Connections:

- **Vcc** to **5V** of Nodemcu.
- **GND** to **GND** of Nodemcu.
- **SDA** to Serial Data.
- **SCL** to Serial Clock.

➢ NODEMCU – ESP8266:

The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The firmware uses the **Lua scripting language** and development board specially targeted for IoT based Applications.





➢ Jumper Wires:

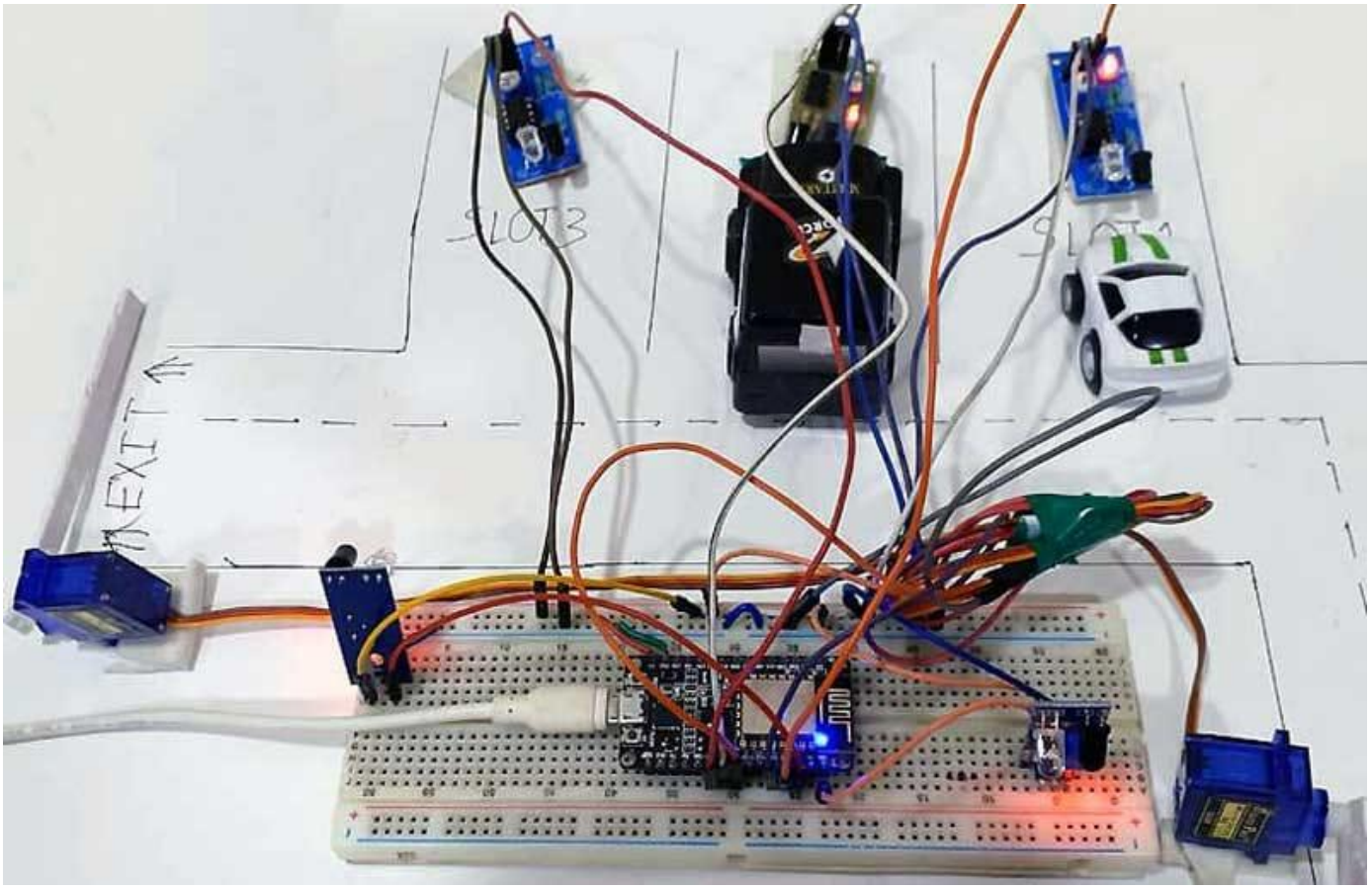
A jump wire (also known as jumper, jumper wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Male to Female Jumper Wires

Female to Female Jumper Wires

Male to Male Jumper Wires

CIRCUIT DIAGRAM



CODE

```
#include <Servo.h>                    // Library for Servo motors
#include <Wire.h>                     // Library for I2C protocol
#include <LiquidCrystal_I2C.h>        // Library for I2C LCD

Servo myservo;                        //servo as exit gate
Servo myservos;                       //servo as entry gate
LiquidCrystal_I2C lcd(0x3f,16,2);     // set the LCD Address

int carEnter  = D0;                   // entry sensor
int carExited = D6;                   //exit sensor
int slot1     = D7;
int slot2     = D3;
int count     = 0;
int CLOSE_ANGLE = 100;                // The closing angle of the servo motor arm
int OPEN_ANGLE = 0;                   // The opening angle of the servo motor arm
int s1=0, s2=0;
int slot = 2;
```

```
boolean entrysensor, exitsensor;
void setup()
{
  Serial.begin (9600);
  Wire.begin(D2, D1);                    // i2c start
  myservo.attach(D8);                    // servo pin to D4
  myservos.attach(D5);                   // servo pin to D5 FOR ENTRY
  pinMode(carEnter, INPUT);              // ir as input
  pinMode(carExited, INPUT);             // ir as input
  pinMode(slot1, INPUT);
  pinMode(slot2, INPUT);
  lcd.begin(16,2);                       //begin lcd
  lcd.backlight();
  lcd.setCursor (0,0);
  lcd.print("  SMART PARKING   ");
  lcd.setCursor (0,1);
  lcd.print("     SYSTEM ");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  Have Slot: ");
  lcd.print(slot);
  delay(3000);
```

```
  lcd.clear();
}
void loop()
{
  entrysensor= !digitalRead(carEnter);
  exitsensor = !digitalRead(carExited);
  s1 = digitalRead(slot1);
  s2 = digitalRead(slot2);

   if (entrysensor == 1 && count<2)
   {
    count=  count+1;
    lcd.setCursor(1,0);
    lcd.print(count);
    lcd.print(" car entered");
    myservos.write(OPEN_ANGLE);
    delay(3000);
    myservos.write(CLOSE_ANGLE);
    lcd.clear();
    slot=slot - 1;
    lcd.setCursor(0,0);
    lcd.print("  Have Slot: ");
    lcd.print(slot);
```

```
 delay(3000);
 lcd.clear();
 if(count>=2)
 {
  delay(200);
  lcd.setCursor(1,0);
  lcd.print("Space Full");
  myservos.write(CLOSE_ANGLE);
 }
}
if (exitsensor == 1 && count>0)
{
 count= count-1;
if(count==1)
{
 lcd.setCursor(1,0);
 lcd.clear();
 lcd.print(" 1 Spcae");
 lcd.setCursor(1,1);
 lcd.print("Available");
 myservo.write(OPEN_ANGLE);
 delay(2500);
 myservo.write(CLOSE_ANGLE);
```

```
 lcd.clear();
 slot=slot + 1;
 lcd.setCursor(0,0);
 lcd.print("  Have Slot: ");
 lcd.print(slot);
 delay(3000);
 lcd.clear();
}

if(count==0)
{
 lcd.setCursor(1,0);
 lcd.print("2 Spcae");
 lcd.setCursor(1,1);
 lcd.print("Available");
 myservo.write(OPEN_ANGLE);
 delay(2500);
 myservo.write(CLOSE_ANGLE);
 lcd.clear();
 slot = slot + 1;
 lcd.setCursor(0,0);
 lcd.print("  Have Slot: ");
 lcd.print(slot);
```

```
    delay(3000);
    lcd.clear();
   }
 }
}
```

ABOUT THE TEAM

- We all team members have worked together towards a common goal and all share his/her responsibility to achieve specific outcomes of their project.
- My team fellow Suraj and Bhavya have their contribution as to collect all the project related materials likes Buying Sensors & IoT devices, connection established between Nordmcu and Sensors that are required for our project.
- **I have given my contribution to help my team as to make connection with Nodemcu and write the code to run the project.**
- We all team members enjoyed this project very well and we were thankful to our mentor because they give us the right direction and also help in the completion of the project.