

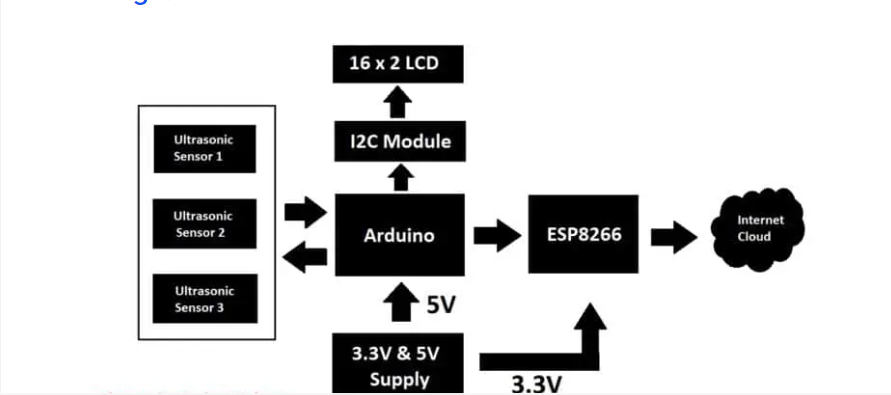
|  |  |  |  |
| --- | --- | --- | --- |
| **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  **Domain Name : Internet of things (IOT)**  **Project Title : smart car parking System** | | | |
| **1.** | **Name of the Student (s)** |  |
|  | |  |  |  |  | | --- | --- | --- | --- | | **S.No** | **Name of the Student** | **E-Mail ID** | **Phone No.** | | **1** | Muthuannammalai.SV | Muthusivaraman04@gmail.com | 7904988251 | | **2** | Gowtham.S | Gowthamsel235@gmail.com | 7810045220 | | **3** | Thejesh bhagavanth.G | thejeshbhagavanth@gmail.com | 9626609830 | | 4  5 | Sasinath .D  Sibi.S | sasinathengineer@gmail.com  Sibilali17@gmail.com | 8825782192  9047774669 | | 6 | Hemasundar. | spideysundar2004@gmail.com | 6381543849 | | | |
| **2.** | **Name of the Guide** | : Ms.Suganya |
|  | **Department/ Designation** | : CSE/AP |
|  | **Institutional Address** | : Chettinad College of Engineering  and Tecnology  NH-67, Karur-Trichy Highway,  Puliyur CF, Karur |
|  | **Phone No** | : 6374527207 |

**PROBLEM DEFINITION :**

The project involves integrating IoT sensors into public transportation vehicles to monitor ridership, track locations, and predict arrival times. The goal is to provide real-time transit information to the public through a public platform, enhancing the efficiency and quality of public transportation services. This project includes defining objectives, designing the IoT sensor system, developing the real-time transit information platform, and integrating them using IoT technology and Python.

.

**DESIGN THINKING :**

****

**System Model:**

The circuit we are going to build will be based on the above architecture. An inexpensive Arduino board is going to be the brain of the project.

 A 16 x 2 LCD is utilized for displaying the number of vacant spots locally (without internet). An I2C module is utilized for driving the LCD with just four wires so that GPIO pins can be saved for interfacing the sensors and other modules.

There are three ultrasonic sensors for detecting 3 cars / vehicles on the parking spot, we are using ultrasonic sensors instead of IR based sensors because if the parking lot is situated outdoors, infrared light from sunlight may interfere with IR sensors and may give incorrect detection of the vehicle, whereas ultrasonic sensor acts like a mini radar and environmental factors affecting its functionality is minimal.

An ESP8266 Wi-Fi module is used for internet connectivity which sends the parking lot’s data to a cloud server where general public can view the data in real time. A power supply module is utilized which provides 5V and 3.3V for Arduino, ultrasonic sensors and ESP8266 Wi-Fi Module.

The internet cloud service we are going to use is called “Thingspeak” where the parking lot’s data to be sent, stored and displayed in real time. This concludes the block diagram.

**Power supply unit:**

A power supply module takes 9V to 12V DC from a wall adapter and converts in to 5V and 3.3V outputs, the 5V output from power supply module is directly connected to 5V pin of Arduino and GND of power supply is connected to GND of Arduino.

Similarly 3.3V from the power supply unit is connected to 3.3V Vcc of ESP8266 (it operates strictly on 3.3V and 5V will kill the module), the ground of power supply is connected to ground of ESP8266.

Make sure you that have inserted the voltage select jumpers correctly.

**Generic ESP8266 Wi-Fi module:**

This project utilizes a generic ESP8266 Wi-Fi module for internet connectivity. The ESP8266 is actually a miniature microcontroller board and just like Arduino the ESP8266 need a program code to perform its intended function.

It uses UART protocol to communicate with Arduino board; the baud rate we are going to set for UART is 115200 bits per second.

**Ultrasonic Sensor HC – SR04:**

The sensor we are going to use for detecting a parked vehicle on its parking spot is called HC – SR04 which is an ultrasonic sensor module.

The ultrasonic sensor module generates ultrasonic sound at around 40 KHz, these sound waves are inaudible to human beings and propagate through air and if the ultrasonic sound wave hits an obstacle, it reflects back to sensor just like radars.

If a car or any vehicle is parked, the ultrasonic sound waves hit the parked vehicle and the sensor module detects the reflection and thus existence of a vehicle on a parking spot is detected.

The ultrasonic sensor module has four pins, Vcc, GND, trigger and echo. The Vcc is connected to 5V supply and GND is connected to GND of the supply.  When we apply “HIGH” signal to trigger pin for 10 microseconds, the module generates ultrasonic sound from one of the transducers, when the sound wave hit back the other transducer, the echo pin gets “HIGH” and this signal is detected by Arduino.

The time taken between generating and detecting the sound wave is calculated and thus a parked vehicle is detected.

**LCD display module 16 x 2:**

In this project we are using a 16 x 2 LCD display for displaying parking lot’s data locally without the need for internet. The LCD is driven by an I2C adapter module to reduce the number of wires to four; otherwise you need to connect up to 16 wires to Arduino just to drive the display. If the LCD occupies most of the I/O pins, then there won’t be any pins left for the sensors.

The I2C module has 16 pins at the output and just four at the input: Vcc, GND, SDA and SCL. The SDA and SCL are I2C bus pins which are connected to A4 and A5 pins of Arduino respectively and it operates on 5V.

You can control the contrast of the display by adjusting the trim pot on the I2C adapter module. This concludes about the circuit diagram.

**How to setup your Thingspeak account?**

We are using a (free) cloud service called Thingspeak where we will send parking lot’s data to share it with public.

**How to operate the machine?**

* After completing the circuit setup and Thingspeak account setup, power the circuit ON.
* Here we have three parking spots and each parking spot’s status is displayed along with the number of free parking spots.
* When you bring an obstacle like a junk box near to the ultrasonic sensor, the machine will count it as an occupied parking spot and this information will be updated to Thing speak as well as on the display.
* “OK” status signifies that a car / vehicle is occupied and “NO” signifies the parking spot is empty.

**Thing speak public view on smart phone:**

The URL of the public view page is converted into a QR and this can be done using any online URL to QR code converter tool. This QR code can be placed at the parking spot or anywhere else, so that the users can scan and bookmark the URL and when they are visiting the parking spot, users can open the link and see how many vacant spaces exist and if this parking lot is full they can move to another.

Future:

**Website Access**: Additionally, display the QR code on website, so anyone visiting the website can access the QR code for scanning. This provides an alternative way for users to access parking information.