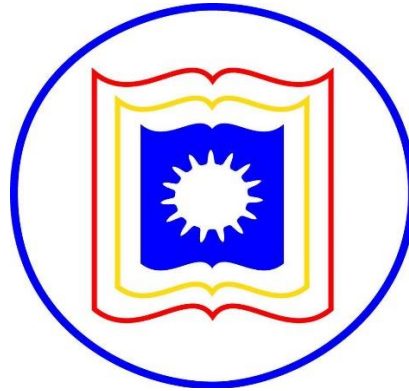


Smart Car Parking Using ARDUINO

Computer Science & Engineering
University of Rajshahi



Submitted By
Bappi Kumar
ID: 1611176112
Session: 2015-16
Year: 4th year, Odd sem.
Course ID: CSE 4152

Project Advisor
Prof. Dr. Bimal Kumar Pramanik
Dept. of CSE

Content

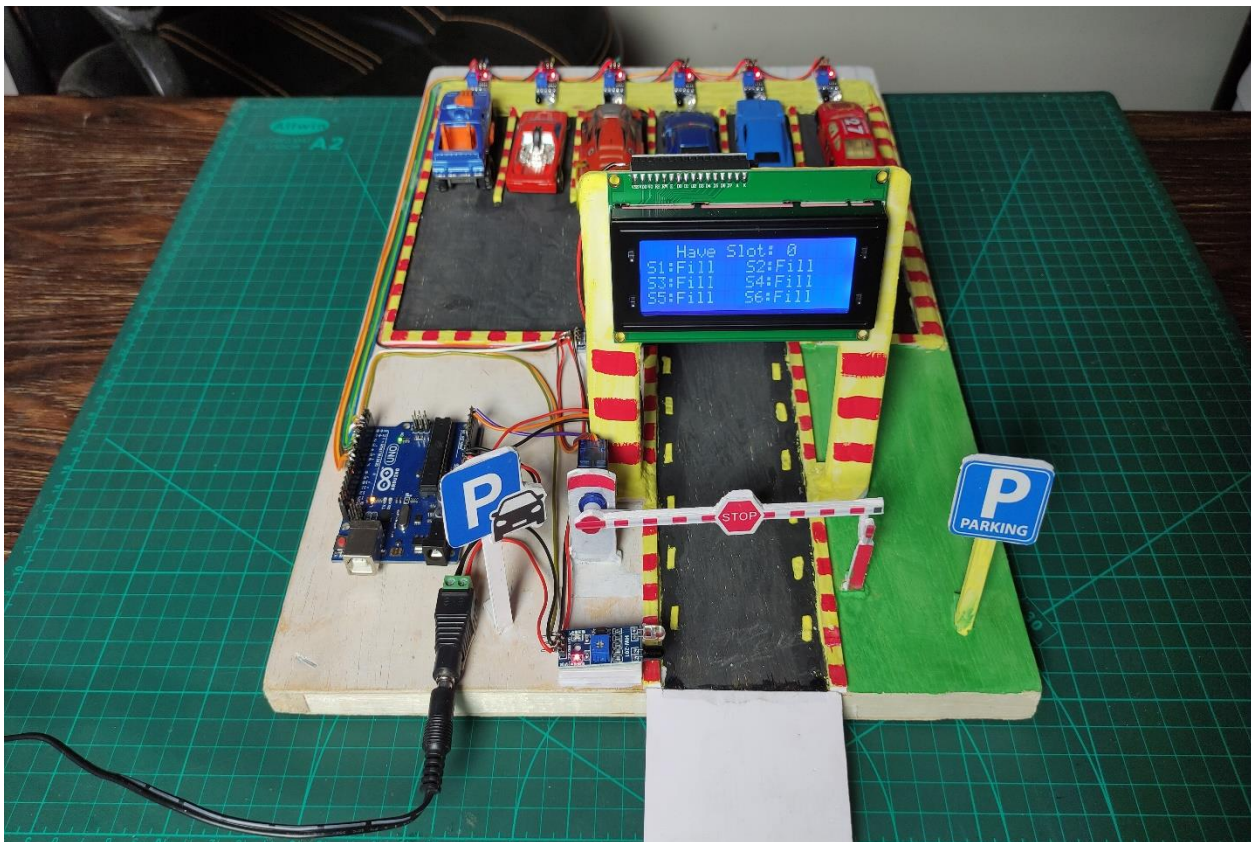
1. Abstract
2. Introduction
3. The proposed design
4. Construction and Working
5. Components
6. Block Diagram of Smart Parking System Project
7. Proteus Simulation
8. Tools Required
9. Project Background
10. Equipment Description
11. Arduino Code
12. Project Cost (Approximately)
13. Conclusion
14. References

Abstract

Smart car parking project aims at providing a confusion free and easy parking. This project helps the drivers of the cars to park their vehicles with minimum wastage of time with accurate information of the availability of the space to park over Android app. The operator also can collect parking fees efficiently and the drivers can book and pay for their parking space over Android app. It includes an Arduino Uno and Arduino Mega as the microcontroller unit to which the servo motors, LCD, object counter using IC 555 and IC 4026, ultrasonic sensors (HC-05) and IR sensors SR21-IC are interfaced. The LCD displays the availability of the space, the counter keeps the check of the number of cars entering and exiting the parking space, the servo motor helps as gate for the entry and exit of the cars. The ultrasonic sensors detect the availability of the parking space.

Introduction/Project Summary

Smart car parking is an automated, flexible, user friendly and highly efficient technology as the booking of parking slot for the driver's vehicle is made possible using an Android app. Also, the operator can easily keep the track of vehicles entering and exiting the parking space and parking fees collected.



The proposed design

As mentioned above, the proposed smart parking lot circuit will be equipped with several sensors, inexpensive microcontrollers and Wi-Fi module using which a car / any vehicle owner can check if there is a vacant space in a parking lot using his / her phone or tablet or even on computer.

The number of vacant spaces in the smart parking lot can be viewed from anywhere in the world using a URL link or the user can scan a QR code. The scanned / shared URL can be browsed on any web browser to know how many empty parking spot exist in real time.

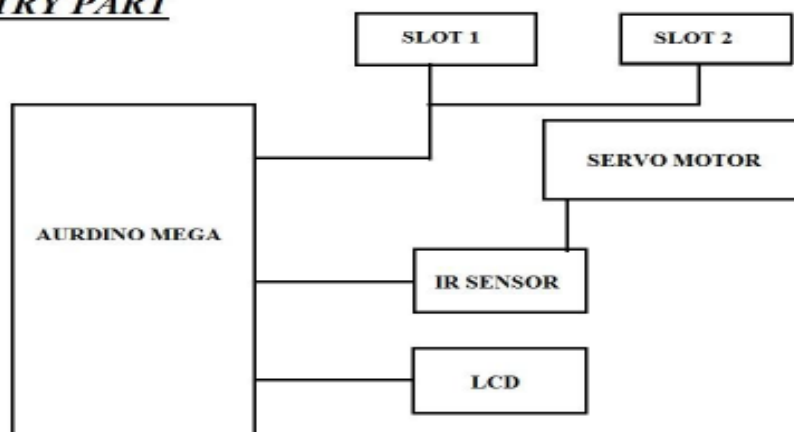
Construction and Working

The construction and working of the project is divided into two parts:

1. Entry Part

The Entry Part of the project consists of Arduino Mega microcontroller to which a servo motor, IR sensor, LCD and ultrasonic sensor are interfaced. The servo motor acts as a gate at the entrance and it opens and closes when the IR sensor detects presence of car. The LCD displays the parking slots which are empty for the car drivers. The ultrasonic sensors detect the presence and absence of car in each parking slot.

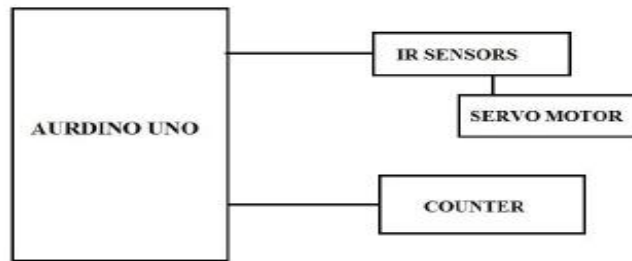
ENTRY PART



2. Exit Part

The Exit Part of the project consists of Arduino Uno to which a servo motor, IR sensor and the object counter are interfaced. The servo motor acts as a gate at the entrance and it opens and closes when the IR sensor detects presence of car. The object counter circuit is designed using IC 555 and IC 4026 to count the number of cars exiting the parking space. This will help the operator to calculate the amount collected.

EXIT PART

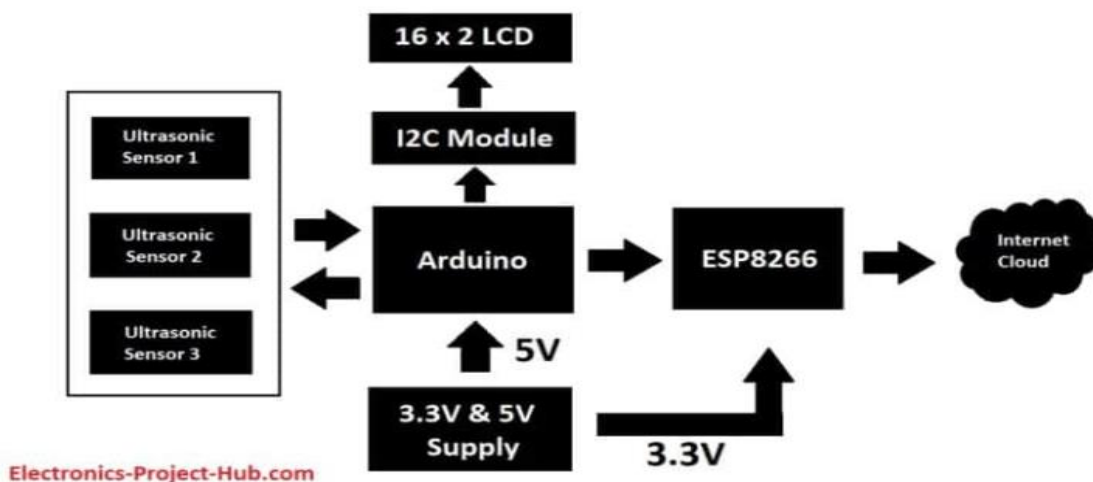


Components

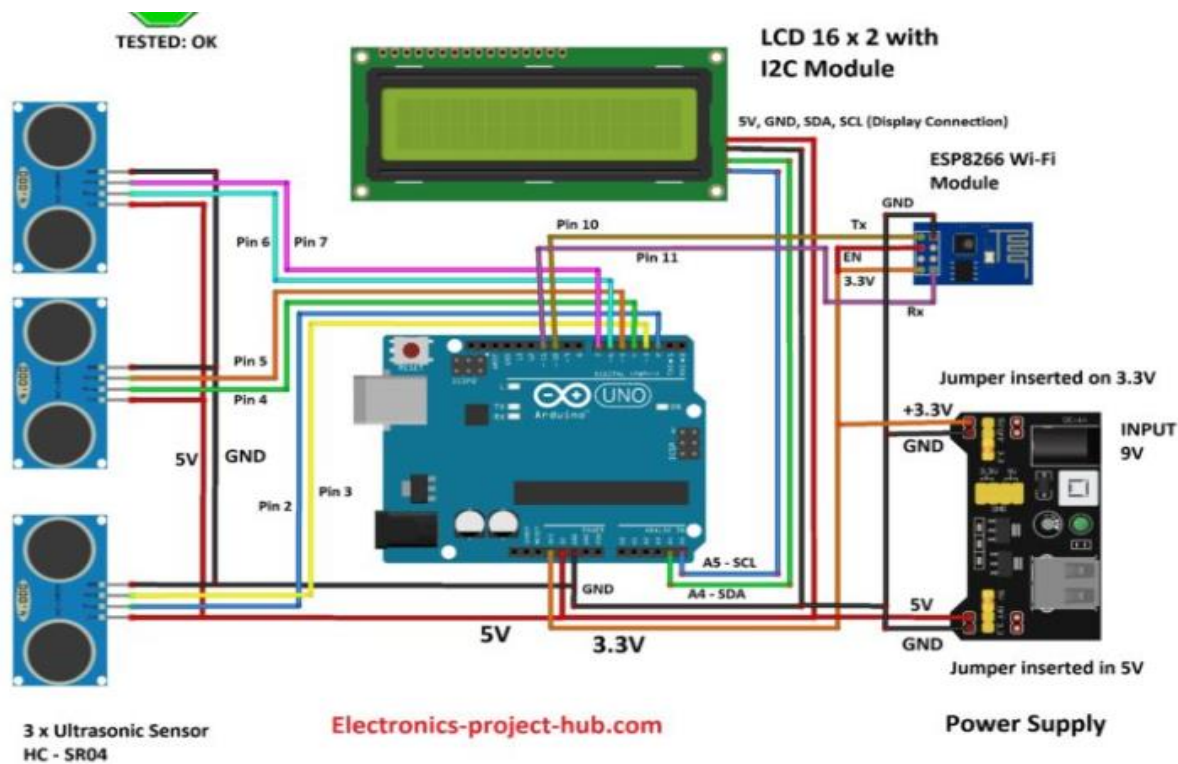
The components used in the project are:-

1. Arduino Uno
2. Arduino Mega
3. I2C LCD Module
4. Ultrasonic Sensor
5. IR Sensor
6. Mini Servo Motor SG-90
7. Object Counter
8. Male Header
9. Female Header
10. Female DC Power Jack
11. 5v 2Amp Power Adapter
12. 20×4 LCD Display

Block Diagram of Smart Parking System Project



Proteus Simulation



Tools Required

- 1.Soldering Iron
- 2.Soldering wire
- 3.Soldering flux
- 4.Soldering stand
- 4.Multimeter
- 5.De-soldering pump
- 6.Wire-cutter

Project Background

After assembling all components according to the circuit diagram and uploading the code to the Arduino board. Now place the sensors and servo motor at accurate positions.

There are four parking slots in this project, IR-Sensor-3, 4, 5, and 6 are placed at slot-1, 2, 3, and 4 respectively. IR sensor-1 and 2 are placed at the entry and exit gate respectively and a servo motor is used to operate the common single entry and exit gate. The LCD display is placed near the entry gate.

The system used IR sensor-3, 4, 5, and 6 to detect whether the parking slot is empty or not and IR sensor-1, and 2 for detecting vehicles arriving or not at the gate.

In the beginning, when all parking slots are empty, then the LCD display shows all slots are empty. When a vehicle arrives at the gate of the parking area then the IR sensor-1 detects the vehicle and the system allowed to enter that vehicle by opening the servo barrier. After entering into the parking

area when that vehicle occupies a slot then the LED display shows that the slot is full. In this way, this system automatically allows 4 vehicles.

In case the parking is full, the system blocked the entrance gate by closing the servo barrier. And the LED display shows that slot-1, 2, 3, and 4 all are full.

When a vehicle leaves a slot and arrives at the gate of the parking area then the IR sensor-2 detects that vehicle and the system open the servo barrier. Then the LED display shows that the slot is empty. Again the system will allow entering a new vehicle.

Equipment Description

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.

Ultrasonic Sensor

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.

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.

LCD display module

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.

Arduino Code

```
#include <Servo.h> //includes the servo library
#include <Wire.h>
#include <LiquidCrystal_I2C.h> //includes LiquidCrystal_I2C library
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

```
Servo myservo;
```

```
#define ir_enter 2
#define ir_back 4
```

```
#define ir_car1 5
#define ir_car2 6
#define ir_car3 7
#define ir_car4 8
```

```
int S1=0, S2=0, S3=0, S4=0 ;
int flag1=0, flag2=0;
int slot = 6;
```

```
void setup(){
  Serial.begin(9600);
  // initialize digital pins as input.
  pinMode(ir_car1, INPUT);
  pinMode(ir_car2, INPUT);
  pinMode(ir_car3, INPUT);
  pinMode(ir_car4, INPUT);
```

```
  pinMode(ir_enter, INPUT);
  pinMode(ir_back, INPUT);
```

```
  myservo.attach(9); // Servo motor pin connected to D9
  myservo.write(90); // sets the servo at 0 degree position
```

```
// Print text on display
  lcd.begin(20, 4);
  lcd.setCursor (0,1);
```



```
lcd.print(" Smart Car ");
lcd.setCursor (0,2);
lcd.print(" Parking System ");
delay (2000);
lcd.clear();
```

```
Read_Sensor();
```

```
int total = S1+S2+S3+S4;
slot = slot-total;
}
```

```
void loop()
{
```

```
Read_Sensor();
```

```
lcd.setCursor (0,0);
lcd.print(" Have Slot: ");
lcd.print(slot);
lcd.print(" ");
```

```
lcd.setCursor (0,1);
if(S1==1)
{
  lcd.print("S1:Fill ");
}
else
{
  lcd.print("S1:Empty");
}
```

```
lcd.setCursor (10,1);
if(S2==1)
{
  lcd.print("S2:Fill ");
}
else
{
  lcd.print("S2:Empty");
}
```

```
lcd.setCursor (0,2);
if(S3==1)
```

```

{
  lcd.print("S3:Fill ");
}
else
{
  lcd.print("S3:Empty");
}

```

```

lcd.setCursor (10,2);
if(S4==1)
{
  lcd.print("S4:Fill ");
}
else
{
  lcd.print("S4:Empty");
}

```

/* Servo Motor Control

*******/**

```

if(digitalRead (ir_enter) == 0 && flag1==0) // read digital data from IR sensor1
{
  if(slot>0)
  {
    flag1=1;
    if(flag2==0)
    {
      myservo.write(180);
      slot = slot-1;
    }
  }
  else
  {
    lcd.setCursor (0,0);
    lcd.print(" Sorry Parking Full ");
    delay(1500);
  }
}

```

```

if(digitalRead (ir_back) == 0 && flag2==0) // read digital data from IR sensor2
{
  flag2=1;
  if(flag1==0)
  {

```

```

    myservo.write(180); // sets the servo at 180 degree position
    slot = slot+1;
  }
}

if(flag1==1 && flag2==1)
{
  delay (1000);
  myservo.write(90); // sets the servo at 90 degree position
  flag1=0, flag2=0;
}
delay(1);
}

void Read_Sensor()
{
  S1=0, S2=0, S3=0, S4=0;
  if(digitalRead(ir_car1) == 0){S1=1;} // read digital data from IR sensor3
  if(digitalRead(ir_car2) == 0){S2=1;} // read digital data from IR sensor4
  if(digitalRead(ir_car3) == 0){S3=1;} // read digital data from IR sensor5
  if(digitalRead(ir_car4) == 0){S4=1;} // read digital data from IR sensor6
}

```

Project Cost (Approximately)

1. Arduino Uno (989 Tk)
2. Arduino Mega (2200 Tk)
3. I2C LCD Module (430 Tk)
4. Ultrasonic Sensor (110 Tk)
5. Generic ESP8266 Wi-Fi module (200 Tk)
6. Mini Servo Motor SG-90 (355 Tk)
7. 5v 2Amp Power Adapter (100 Tk)
8. Male Header (20 Tk)
9. Female Header (20 Tk)
10. Female DC Power Jack (15 Tk)

Conclusion

Our project detects the empty slots and helps the drivers to find parking space in unfamiliar city. The average waiting time of users for parking their vehicles is effectively reduced in this system. The optimal solution is provided by the proposed system, where most of the vehicles find a free parking space successfully. Our preliminary test results show that the performance of the Arduino UNO based system can effectively satisfy the needs and requirements of existing car parking hassles thereby minimizing the time consumed to find vacant parking lot and real time information rendering. This smart parking system provides better performance, low cost and efficient large

scale parking system. When car enters the parking area, the driver will park the car in the nearest empty slot when slot is occupied the LED light glows and when slot is empty LED lights are turned off automatically indicating that the parking slot is empty to be occupied. It also eliminates unnecessary travelling of vehicles across the filled parking slots in a city.

References

1. Thanh Nam Pham¹, Ming-Fong Tsai¹, Duc Bing Nguyen¹, Chyi-Ren Dow¹ and Der-Jiunn Deng². "A Cloud- Based Smart-Parking System Based on Internet-of-Things Technologies". IEEE Access, volume 3, pp. 1581 – 1591, september 2015.
2. International Journal of Computer Applications (0975 – 8887) Volume 169 – No.1, July 2017
3. <https://electronics-project-hub.com>
4. <https://store.roboticsbd.com>
5. International Journal of New Technology and Research (IJNTR)