ISSN (e): **2250 – 3005** || *Volume, 08 || Issue, 5|| May – 2018 ||*

International Journal of Computational Engineering Research (IJCER)

Smart Vehicle Parking Using Iot

Pothuraju Jagadeesh

Assistant Professor, Dept. Of E.C.E, School of Engineering & Technology, SPMVV, Tirupati, A.P, India.
Corresponding Author: Pothuraju Jagadeesh

ABSTRACT

The paper aims at designing advanced smart parking system using IOT technology. The devices can be switched ON/OFF using a mobile through server (Wi-Fi). Generally people are facing problems on parking vehicles in parking slots in a city. So i design a Smart Parking System which enables the user to find the nearest parking area and gives availability of parking slots in that respective parking area.

KEYWORDS: ESP8266 Node MCU, IC voltage regulator (LM7805),Wi-Fi module, IR sensors, Arduino, Embedded "C" program.

Date of Submission: 28-04-2018 Date of acceptance: 14-05-2018

I INTRODUCTION

The paper aims at designing an advanced smart parking system using IOT technology. The devices can be switched ON/OFF using a mobile through server (Wi-Fi). Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. These had greater importance than any other technologies due to its user-friendly nature. These can be used as a replacement of the existing switches in home which produces sparks and also results in fire accidents in few situations. Considering the advantages of Wi-Fi an advanced automation system was developed to monitor the status of parking slots.

Wi-Fi (Short for **Wi**reless **Fi**delity) is a wireless technology that uses radio frequency to transmit data through the air. Wi-Fi has initial speeds of 1mbps to 2mbps. Wi-Fi transmits data in the frequency band of 2.4 GHz. It implements the concept of frequency division multiplexing technology. Range of Wi-Fi technology is 40-300 feet. The controlling device for the monitoring in the project is a Microcontroller. The data collected by the Microcontroller. Microcontroller reads the data and sends the data over Wi-Fi to the IOT web page. The Microcontroller is programmed used embedded "C" language.

II PROPOSED SYSTEM

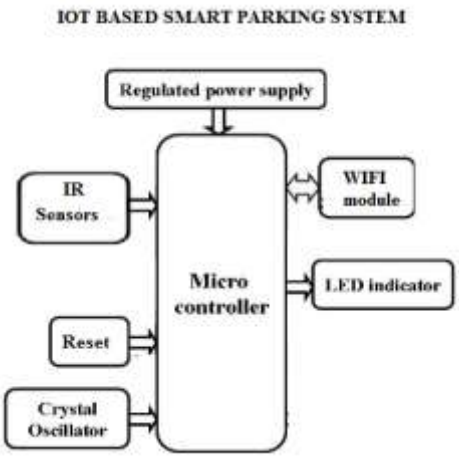
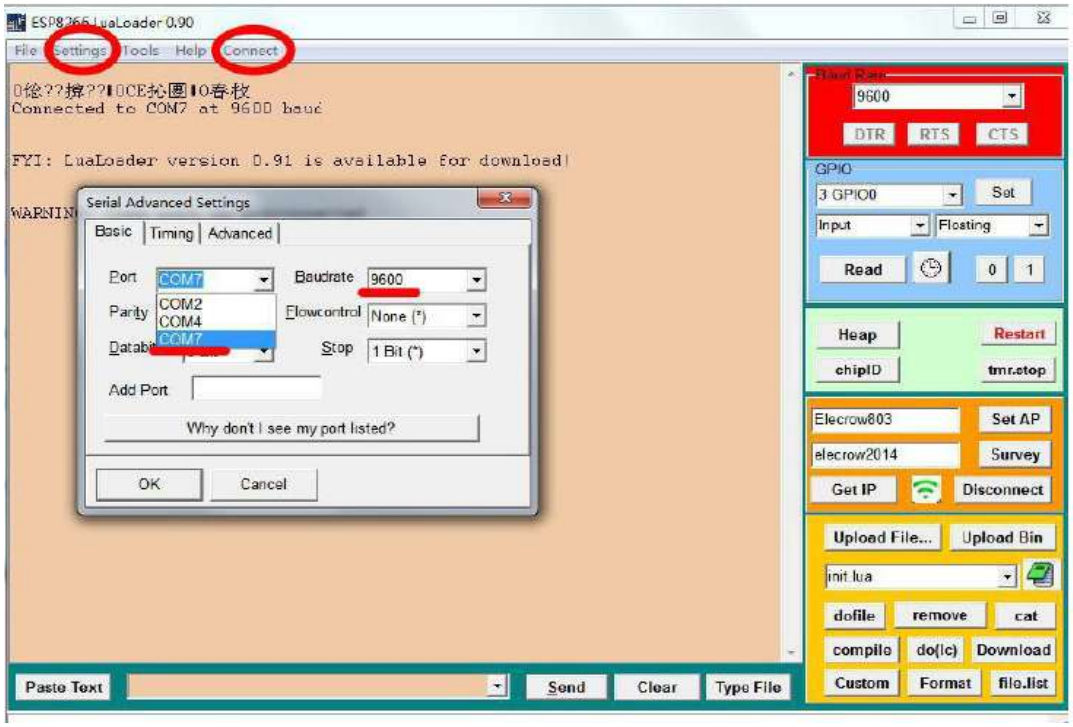


Figure 1: IOT Based Smart Vehicle Parking Proposed Model

III ESP8266 NODE MCU

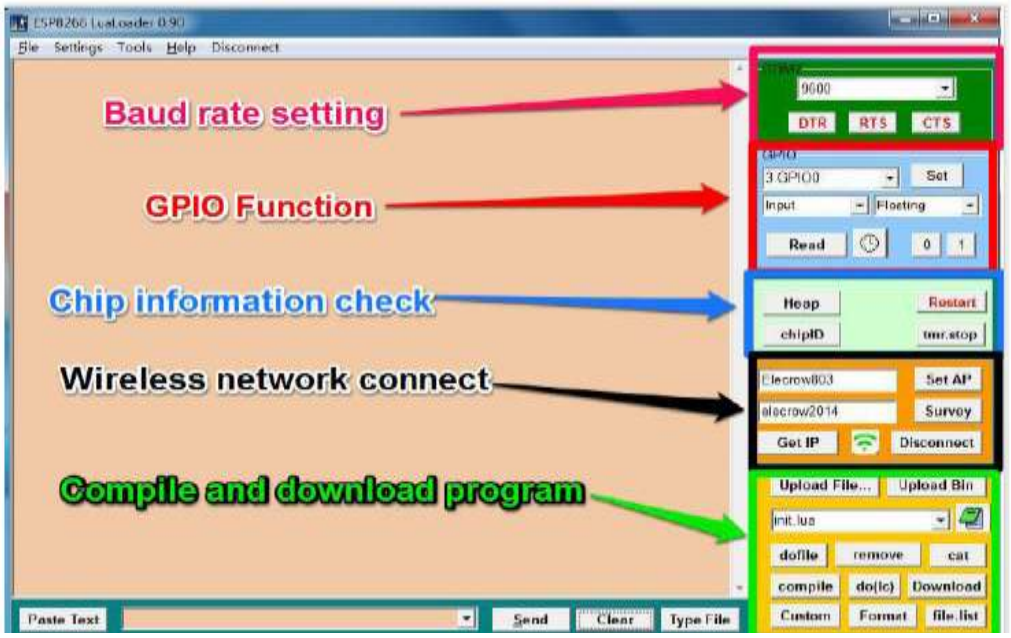
How to use ESP8266 Node MCU?

1. Before you use the micro USB cable to connect the ESP8266 NodeMCU with your computer, you need to install the cp2012 USB driver.
2. After succeed connection, we need to choose some development tools such as LuaLoader and luaEditor, we will provide these software. You can open them from Tools folder. The luaEditor is used to programming and debugging, finally it will generate .lua file. The LuaLoader is used to download and compile the .lua file, and it provides some other features. The next will introduce the usage of LuaLoader.
3. open the LuaLoader, click the menu "Setting", choose the "Comm port Settings", and it will popular "Serial Advance Setting", set up the right port here then click the "Connect" button on the top of the menu.
4.



When the information Connected to COMX at 9600baud" display in the debug window, it means connecting is successful, and you can input command statement into the bellow edit box. As send command " print("Hello world") ", then the result will display in the debug window.

Also, you can use some of the function which display on the right side of Lualoader interface.



Baud rate setting: set the baud rate that you need ,normally set at 9600.
GPIO Function: Choose different GPIO port, set its mode(Input, Output or Interrupt),set GPIO Pull up resistor (Floating, Pull up or Pull down), and "Read" or "Write" operation to the GPIO 3 port.
Chip information check: check the information about the chip or restart the Node MCU.
Wireless network connect: Enter the wireless network account and password, you can connect it and get the IP.
Compile and download program: First, click "Upload File…"upload the .lua file that you want to download. Secondly click "compile" to compile the .lua file, through the compile can click "Download" to download the program.

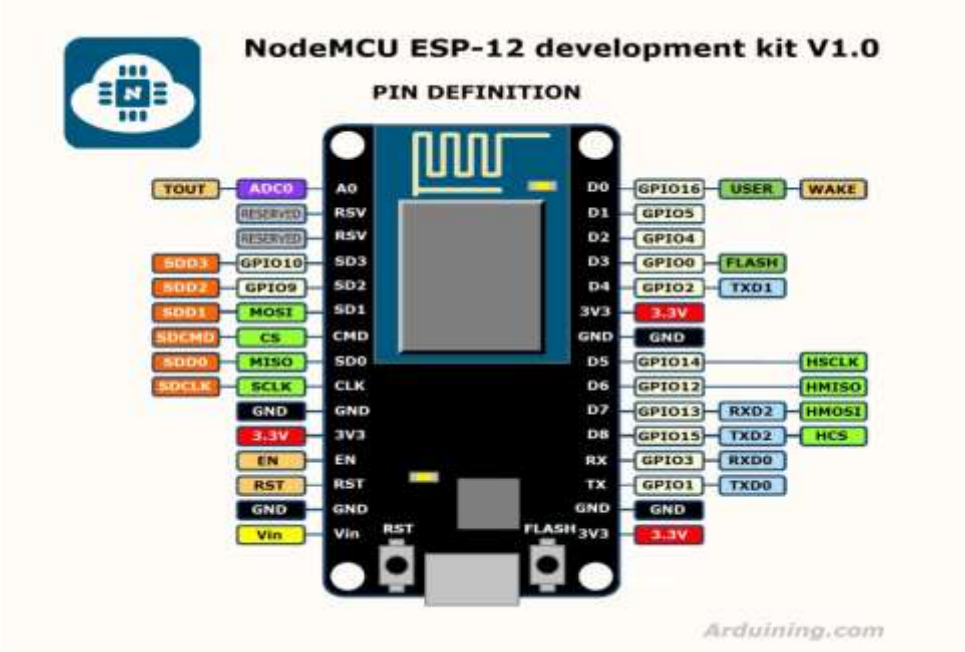


Figure 2: Node MCU ESP-12 development kit V1.0

- Node MCU is a open source IOT platform . It includes firmware which runs on the ESP8266 Wi-Fi SOC from Espressif systems and hardware which is based on the ESP-12 module. The term node MCU by default refers to the firmware rather than the dev kits.
- The node MCU provides access to GPIO (general purpose input/output)and for developing purpose .
- GPIO pins of node MCU are of 3.3v level logic, cannot tolerate 5v all pins are bread board friendly.
- The pin AO (ADCO PIN) it is only one analog pin accepts maximum voltage 1v.

On board there is a 3.3v regulator , so safely provide 5v at Vin pin.

- EN (CH-PD) it is a chip enable pin , which is pulled high internally.
- RST pin - when pulled low resets ESP8226.
- GPIO0 – when pulled low sets ESP into boot loader mode.
- GPIO2- has a pull-up resistor , which is connected to blue LED on ESP chip (near Wi-Fi antenna) used to detect boot mode and as an output to control blue LED.
- GPIO15- has a pull-down resistor can be used as output only, do not pull this during power up.
- GPIO16- used to wakeup the module from the deep sleep mode. Advisible to connect this to RST.
- On board CP2102IC-provides USB to TTL functionality .Previously CS340IC is used ,but CP2102IC is advanced one , works upto windows 10.

Digital IO:

- Pin numbers in Arduino correspond directly to the ESP8266 GPIO pin numbers. pinMode, digital Read, and
- digital Write functions work as usual, so to read GPIO2, call digitalRead(2).
- Digital pins 0—15 can be INPUT, OUTPUT, or INPUT_PULLUP. Pin 16 can be INPUT, OUTPUT or
- INPUT_PULLDOWN_16. At startup, pins are configured as INPUT.
- Pins may also serve other functions, like Serial, I2C, SPI. These functions are normally activated by the corresponding library.
- The diagram below shows pin mapping for the popularESP-12 module.

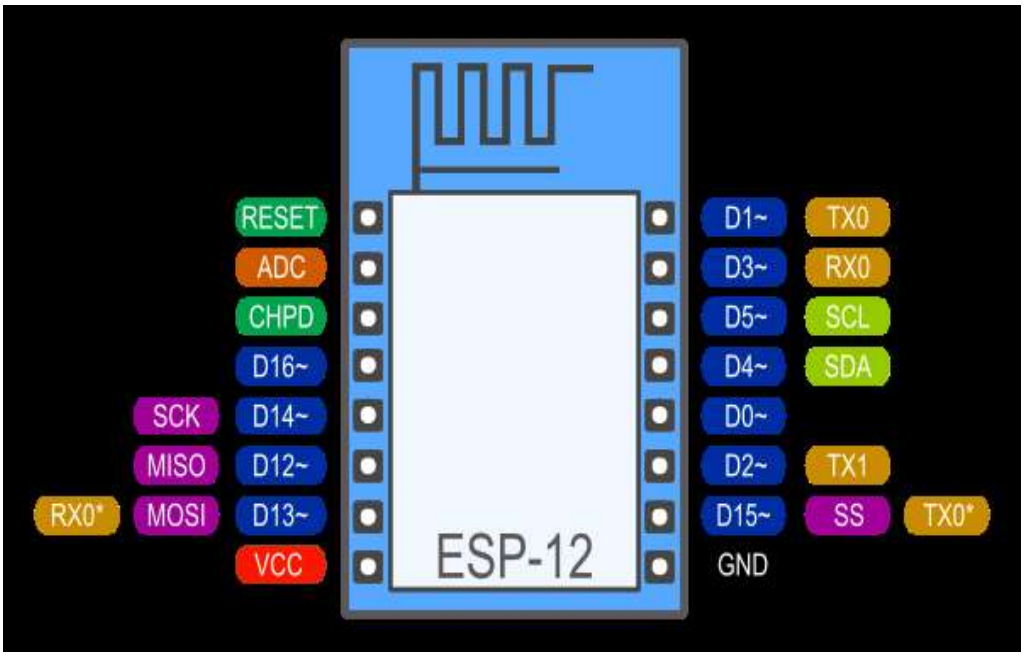


Figure 3: ESP-12 Pin Functions

- Digital pins 6—11 are not shown on this diagram because they are used to connect flash memory chip on most modules. Trying to use these pins as IOs will likely cause the program to crash.
- Note that some boards and modules (ESP-12ED, NodeMCU 1.0) also break out pins 9 and 11. These may be used as IO if flash chip works in DIO mode (as opposed to QIO, which is the default one).

Pin interrupts are supported through attachInterrupt, detachInterrupt functions. Interrupts may be attached to any GPIO pin, except GPIO16. Standard Arduino interrupt types are supported: CHANGE, RISING,FALLING.

Analog input:

- ESP8266 has a single ADC channel available to users. It may be used either to read voltage at ADC pin, or to read module supply voltage (VCC).
- To read external voltage applied to ADC pin, use analog Read(A0). Input voltage range is 0 — 1.0V.
- To read VCC voltage, use ESP. Get Vcc() and ADC pin must be kept unconnected. Additionally, the following line has to be added to the sketch:

Analog output:

- analogWrite(pin, value) enables software PWM on the given pin. PWM may be used on pins 0 to 16. Call analogWrite(pin, 0) to disable PWM on the pin. value may be in range from 0 to PWMRANGE, which is equal to 1023 by default. PWM range may be changed by calling analogWriteRange(new_range).

- PWM frequency is 1kHz by default. Call analogWriteFreq(new_frequency) to change the frequency.

Timing and delays:

- millis() and micros() return the number of milliseconds and microseconds elapsed after reset, respectively.
- delay(ms) pauses the sketch for a given number of milliseconds and allows WiFi and TCP/IP tasks to run.delay(ms) Microseconds(us) pauses for a given number of microseconds.
- Remember that there is a lot of code that needs to run on the chip besides the sketch when WiFi is connected. WiFi and TCP/IP libraries get a chance to handle any pending events each time the loop() function completes, OR when delay is called. If you have a loop somewhere in your sketch that takes a lot of time (>50ms) without calling delay,you might consider adding a call to delay function to keep the WiFi stack running smoothly.
- There is also a yield() function which is equivalent to delay(0). The delayMicroseconds function, on the
- other hand, does not yield() to other tasks, so using it for delays more than 20 milliseconds is not recommended.

Serial:

- Serial object works much the same way as on a regular Arduino. Apart from hardware FIFO (128 bytes for TX and RX) HardwareSerial has additional 256-byte TX and RX buffers. Both transmit and receive is interrupt-driven. Write and read functions only block the sketch execution when the respective FIFO/buffers are full/empty.
- Serial uses UART0, which is mapped to pins GPIO1 (TX) and GPIO3 (RX). Serial may be remapped to GPIO15 (TX) and GPIO13 (RX) by calling Serial.swap() after Serial.begin. Calling swap again maps UART0 back to GPIO1 and GPIO3.
- Serial1 uses UART1, TX pin is GPIO2. UART1 cannot be used to receive data because normally it's RX pin is occupied for flash chip connection. To use Serial1, call Serial1.begin(baudrate).
- If Serial1 is not used and Serial is not swapped - TX for UART0 can be mapped to GPIO2 instead by calling Serial.set_tx(2) after Serial.begin or directly with Serial.begin(baud, config, mode, 2).By default the diagnostic output from WiFi libraries is disabled when you call Serial.begin. To enable debug output again, call Serial set Debug Output(true). To redirect debug output to Serial1 instead, call Serial1.setDebugOutput(true).
- You also need to use Serial.setDebugOutput(true) to enable output from printf() function.Both Serial and Serial1 objects support 5, 6, 7, 8 data bits, odd (O), even (E), and no (N) parity, and 1 or 2 stop bits. To set the desired mode, call Serial.begin(baudrate, SERIAL_8N1), Serial. begin(baudrate, SERIAL_6E2), etc.

IV WORKING MODEL

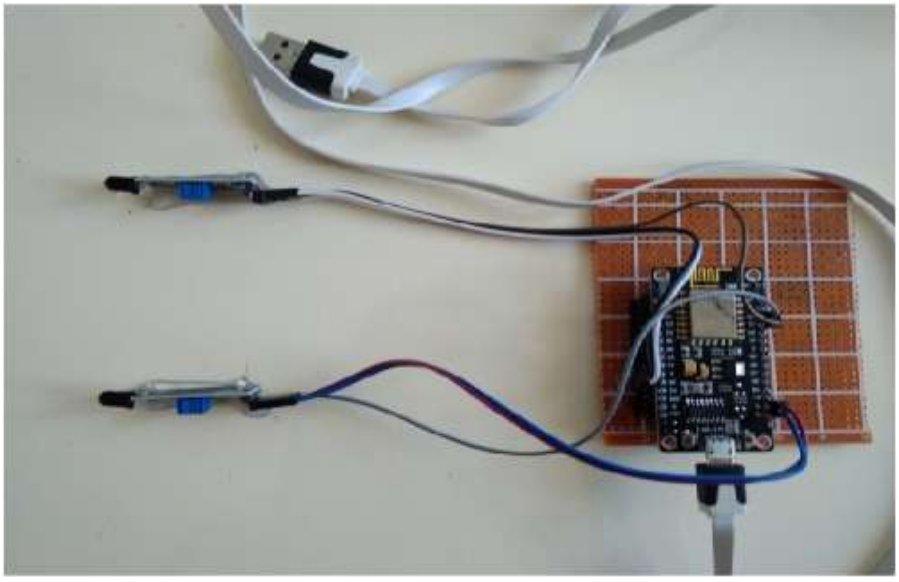


Figure 4: Working Model of IOT Based Smart Parking System

- "IOT based smart parking system" is mainly intended to monitor the status of devices through server (Wi-Fi).
- The controlling device of the whole system is a microcontroller. The power supply of this is provided by IC voltage regulator (LM7805).

- Wi-Fi module, IR sensors are interfaced to the microcontroller. IR sensors are fed as input to the microcontroller through digital pins (D2, D3). In this project we used only two sensors, but there is a possibility to connect up to eight sensors through (D0-D7). The microcontroller processes this data and transmits over Wi-Fi, which will be received from mobile.
- Flash cache hardware on the ESP8266 only allows mapping 1MB of code into the CPU address space at any given time.
- File system is stored on the same flash chip as the program, programming new sketch will not modify file system contents. This allows to use file system to store sketch data, configuration files, or content for Web server.
- Reset is also named RSBT or REST (adding PullUp improves the stability of the module).
- GPIO pins of node MCU are of 3.3v level logic, cannot tolerate 5v all pins are bread board friendly.
- 3v pin and ground pins are connected to the IR sensors. The power supply from 3v is given to the IR sensors.
- In this three 3v pins are present and connected only three IR sensors. So only use three slots.
- For one slot of IR sensor use one digital pin, one 3v pin(Vcc) and ground pin.
- Range of ESP8266 powered Display Modules by 4D Systems.
- In achieving the task controller is loaded with a program written using embedded c language.
- The user who wants to park the vehicle is connected to the Wi-Fi network of that particular parking slot through the password.
- The IR sensors send the status to the microcontroller, where the data processing is done.
- The microcontroller sends information to the user using IOT. This way the user can easily find a parking slot without any congestion and in less time.
- The project was designed such that the status of parking slots can be known from anywhere in the users web page. This is achieved through Wi-Fi communication.
- In this system the user has to be connected to the Wi-Fi network of that particular parking, area through which user is given access to the web page and can known about the status of the parking slot.

V RESULTS

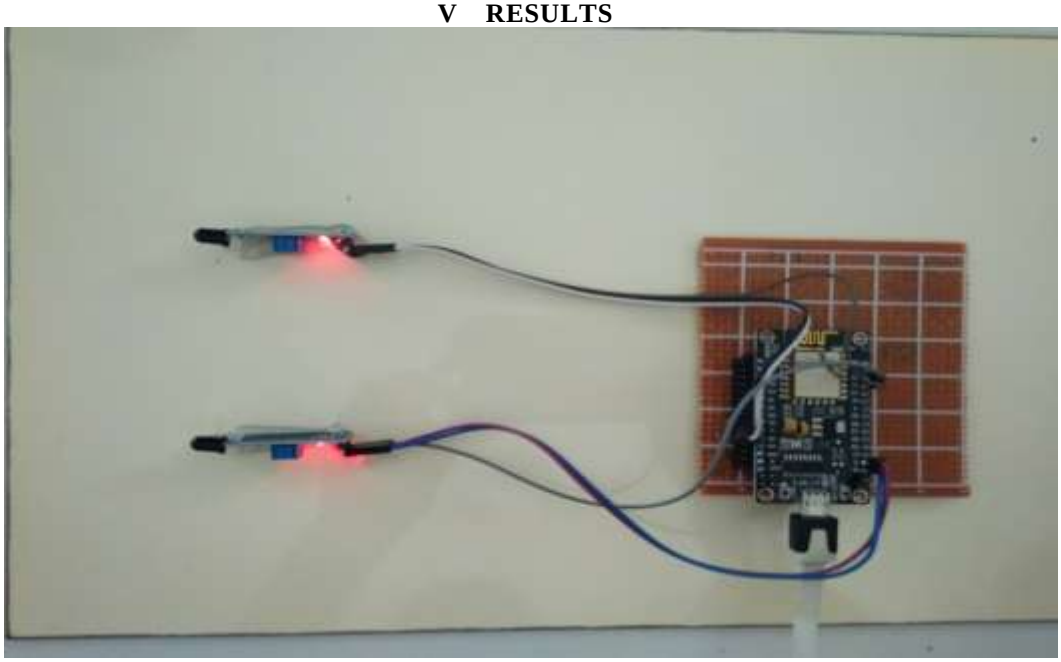


Fig 5: When slots (both) are empty

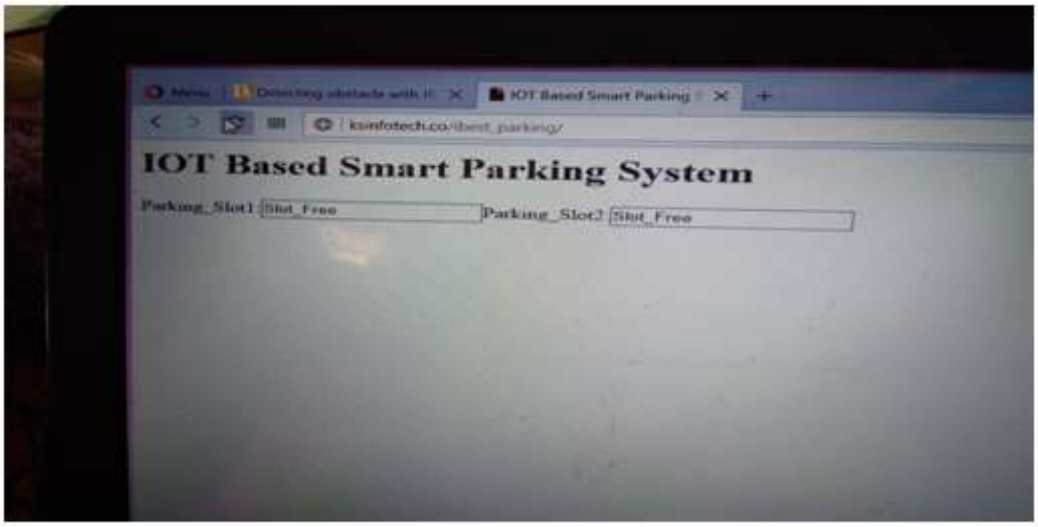


Fig 6: Display the availability of slots

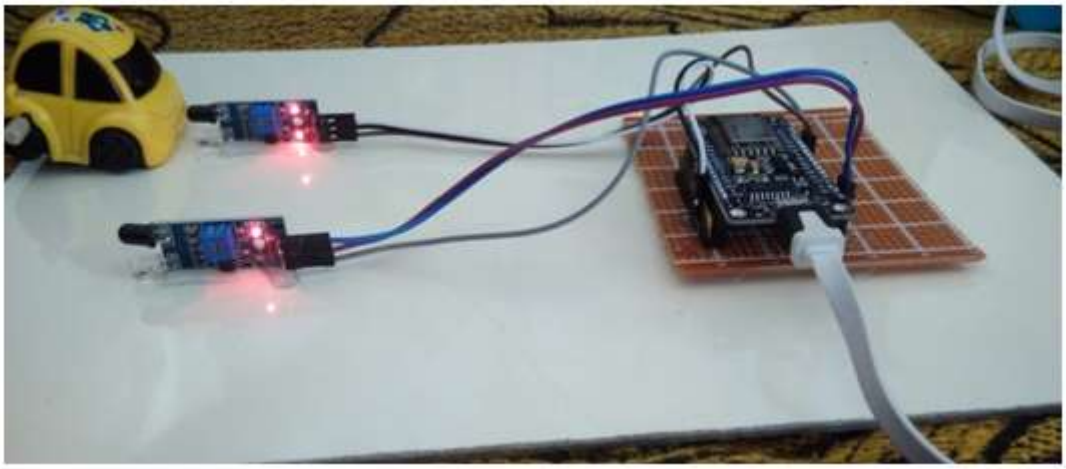


Fig 7: When one slot is occupied by vehicle

Fig 8: Display of that particular slot

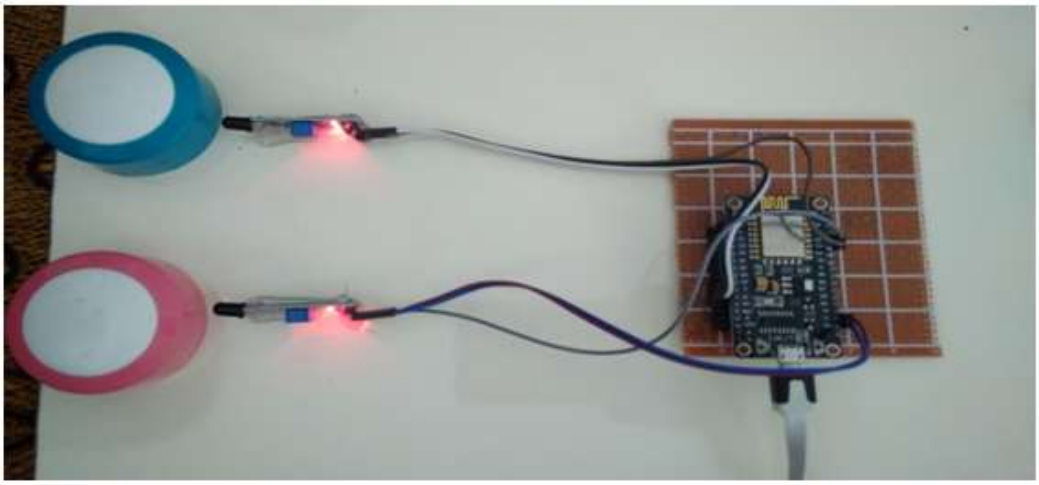


Fig 9: When slots (both) are full

Fig 10: Display of the status of both slots

VI ADVANTAGES

- Reduce time.
- Reduce fuel consumption.
- Reduce carbon foot prints.

VII FUTURE SCOPE

This technology can be extended by adding an application of booking the parking slot before reaching the destination. This can be achieved by using GSM and RFID communication.

VIII CONCLUSION

The hassle in searching for available parking slots has been completely eliminated by reserving the slots via IOT system. The security feature of the system is enhanced with the password requirements upon entrance of the parking slot. The designed system could be applied everywhere due to its easy of usage and effectiveness.

REFERENCES

[1]. M.S. Joshi and Deepali V. Mahajan ,"Arm 7 Based Theft Control, Accident Detection and Vehicle Positioning System'', International Journal of Innovative Technology and Exploring Engineering , vol. 4, no. 2, July 2014.

[2]. Ms.S.S.Pethakar, Prof. N. Srivastava, Ms.S.D.Suryawanshi ,"RFID, GPS and GSM Based Vehicle Tracing and Employee Security System '', International Journal of Advanced Research in Computer Science and Electronics Engineering, vol. 1, no. 10, Dec 2012.

[3]. Pravada P. Wankhade and Prof. S.O. Dahad , "Real Time Vehicle Locking and Tracking System using GSM and GPS Technology-An Anti-theft System'',International Journal of Technology And Engineering System, vol. 2, no.3, Jan –March 2011.

[4]. Abid khan and Ravi Mishra, "GPS – GSM Based Tracking System'', International Journal of Engineering Trends and Technology, vol. 3, no.2, 2012.

[5]. Prashantkumar R., Sagar V. C., Santosh S., Siddharth Nambiar ,"Two Wheeler Vehicle Security System'', International Journal of Engineering Sciences & Emerging Technologies, vol. 6, no. 3, Dec 2013, pp: 324-334.
[6]. Hu Jian-ming, Li Jie and Li Guang-Hui ,"Automobile Anti-theft System Based on GSM and GPS Module'', Fifth International Conference on Intelligent Networks and Intelligent Systems, 2012.
[7]. Fleischer, P.B.; Nelson, Atso Yao; Sowah Ribert Adjectey, Bremang, Appah," Design and Development of GPS/GSM Based Vehicle Tracking and Alert System for Commercial Inter-City Buses'', Adaptive Science and Technology, IEEE 4th International Conference , Oct 2012 .
[8]. Thuong T. Le-Tien, V. Phung-The, "Routing and Tracking System for Mobile Vehicles in Large Area", Fifth IEEE International Symposium on Electronic Design, Test & Applications, pp. 297-300, 2010.
[9]. Nagaraja, B. G.; Rayappa, R.; Mahesh, M.; Patil, C.M.; Manjunath, T. C., "Design and Development of a GSM Based Vehicle Theft Control System", Advanced Computer Control, 2009. ICACC '09. International Conference on, vol., no., pp.148, 152, 22-24 Jan 2009.
[10]. M. A. Al Rashed, Ousmane Abdoulaye Oumar, Damanjit Singh, "A real time GSM/GPS based tracking system based on GSM mobile phone'', IEEE Journal on Signals and Telecommunication, vol. 3, no.1, March 2014, pp. 33-39.

AUTHOR'S INFORMATION

Mr. Pothuraju Jagadeesh obtained his B.Tech (Electronics and Communication Engineering) Degree from Sree Vidyanikethan Engineering College, A.Rangampet, Tirupati, affiliated to JNTU Anantapur in 2010. He obtained his M.Tech (Electronics) Degree from Pondicherry University (A Central University) in 2012. He worked as an Assistant Professor in Vemu Institute of Technology, Chittoor during a period from 2012 -2013. Presently he is working as an Assistant Professor in the Dept.of ECE at School of Engineering And Technology, Sri Padmavati Mahila Visvavidyalayam, Tirupati.

Pothuraju Jagadeesh. "Smart Vehicle Parking Using Iot." International Journal of Computational Engineering Research (IJCER), vol. 08, no. 05, 2018, pp. 24-32.