A Project Report

on

IOT BASED PARKING SYSTEM

For the award of
ONTARIO GRADUATE CERTIFICATE

Submitted by
Francis Vijith A J [C0742980]
Surinder Singh [C0]
Shivam Jassal [C0730509]

Under the guidance of
Dr. Mike Aleshams

ESE-4009 Embedded System Design Project
(Sept 2018 - Jan 2020)

Abstract

The goal of this project was to develop an IoT based Parking System using a PIC 16877A microcontroller. There is a uncontrollable increment in the number of infrastructures and automobiles everywhere. The conventional method of valet parking must be reduced, and technology must be incorporated, such as IoT, which is used in this project. Through this, the effort of searching for a vacant space can be limited as the vacant spots can be observed through Internet. It saves a lot of time and fuel for the users.

IoT, basically comprises of connecting devices to Internet, through which the data can be analyzed, stored, extracted, and imported by users from anywhere. In this project, the primary focus is to connect the PIC 16F877A to a ThingSpeak channel, which is an open source web platform, through which the users can observe the current status of the parking spot from anywhere.

The project hardware was finalized, developed, and presented in a timely manner. Each hardware was tested for full functionality and then each was interfaced to the PIC 16F877A microcontroller. The final project was tested for data collection from the peripheral devices and sending it to the web platform. An user guide is also provided for using and installing the hardware.

Acknowledgments

We would like to express out deepest appreciation to everyone at Embedded Systems Design Engineering Class in Lambton College in Toronto who provided us the possibility to complete this project. A special gratitude to project guide, Dr. Mike Aleshams whose engagement and contribution help us coordinate our project and filing final project report.

We would also like to extend our appreciation to Dr. Takis Zourntos, Course Coordinator, Embedded Systems Design Engineering for providing necessary cooperation during troubleshooting times.

Lambton College in Toronto is a great place to study and work, largely because of supporting faculty members. Thanks to all my colleagues in college for supporting and encouraging our idea.

Francis Vijith A J
Surinder Singh
Shivam Jassal

TABLE OF CONTENTS

List of Figures

Chapter I
Introduction

Overview

This project is about the development of an IOT based parking system, which enables the user to have a live feedback of the parking spots in an area. A PIC 16f877a microcontroller is employed to collect and send data from the peripheral devices like IR sensor, 16x2 LCD display and a stepper motor. It is also enabled with a Wi-Fi module which connects it to the nearest Wi-Fi network to make it an IOT product. The projects make use of the Thingspeak server, to collect data at real time and display to the user from anywhere. The average cost of the project comes around 100$.

Problem Statement

As the cities are expanding and there are more and more vehicles on the road everyday, combining technology with the building infrastructure is vital. In the parking lots, if the conventional valet parking is still continued, then there would be more congested areas due to inefficient parking schemes. And for common users of primate vehicles, they experience a great delay to park their vehicles as they keep searching for a vacant spot. The inclusion of IoT service for parking of vehicles make the users easy to find a spot, even before they arrive at their destination.

Goal and Objectives

- **Goal**
  Develop an IOT based Parking System powered by a PIC16f877a microcontroller by the end of Winter 2020 academic term.

- **Objective**
  - Programming the PIC16877a microcontroller to interface with stepper motor, 16x2 LCD display, IR sensor and ESP8266-01 Wifi module.
  - Test for working of each device with PIC 16877a
  - Configuring the Wi-fi properties of the project and connecting it with the ThingSpeak channel

The Scope of the Project

- **Deliverable**
  - An IOT based parking system enabled by a PIC16f877a microcontroller
  - Wi-Fi and live feedback capabilities to Thinspeak server.

- **Milestones**
    - February 02,2020 – Hardware requirements finalized.
    - February 20,2020 – Testing each hardware parts.
    - February 23,2020 – Designing circuit in software.
    - March 10, 2020 – Interfacing LCD, IR sensor with PIC16f877a
    - March 19, 2020 – Interfacing LED, stepper motor with PIC16f877a
    - March 21, 2020 – Interfacing ESP8266 with PIC16f877a
    - April 04, 2020 – Interfacing the ThingSpeak channel with PIC17f877a
    - April 10, 2020 – PCB designing

- **Limitations**
    - There is a lack of user interface at setting up cinnection to internet.
    - Delay provided to update the Thingspeak in considerably lengthy.
    - The ESP chip draws more current at initial stage, which can damage the on board connections.

Outcomes and Benefits

- Significant low price of the whole system, when compared to other similar IOT products.
- Usage of an open source IOT platform provides free usage of user end applications.
- Easy to set up regardless of factors regarding the location or geography.

Facilities and Resources

- **Laboratory**
    - Embedded Systems Lab.
    - Oscilloscope, multimeter, logic analyzers and PC's in the lab.
- **Intellectual Resources**
    - PIC16F877A MCU datasheet from Microchip.
    - ESP8266-01 Wi-Fi module datasheet from Espressif Systems
    - MPLAB X IDE Users guide

Procedure and Methodology

I. Test each hardware, and its power consumption in every condition.
II. Initiate the interfacing of each hardware to the PIC 16F877A MCU.
III. Evaluate the power consumption of the ESP8266-01 module and adopt an optimum power source for the module.
IV. Develop an IoT web platform, which can indicate the live status of the parking spot and has a cloud storage.
V. Deploy the final code after connecting the hardware with the IoT web platform.

Chapter II
Literature Review

Recently, there have been many proposals related to a smart parking system. According to "IOT Based Smart Parking System" (G. Abhijith, 2019), the project can be implemented using raspberry Pi, but it welcomes the use of an additional module called NodeMCU for controlling the power to the Raspberry PI, which makes more bulky and also the proposal does not provide a Web Application or a server for storing and retrieving the data for future use. It only has a mobile app version which is not flexible for everyone and not safe for the drivers. A scholar paper done on

"A Novel Smart Car Parking System Using IoT for Smart Cities" (Asha, 2020), uses ATtiny 2313 Microcontroller and Ultrasonic sensor. It handles with an Ultrasonic sensor which proved to be not efficient during working as it does not have a wide range of detection compared with IR sensor.

"Intelligent smart Parking System" (Kanteti, Srikar, & Ramesh, 2017) deals with smart parking system developed with many features like automated, rotary parking and sensor technology. This project operates according to an algorithm to make an effective parking system by taking a few information into account. But in this project, CMOS sensors are used for number plate detection and this may lead to incorrect reading of plate numbers as CMOS sensors work with light and the surrounding condition may affect the transmission of light.

"Applying smart parking system with Internet of things" (Mufaqih, 2020), works on a wide area parking lot, where it uses a Raspberry PI to connect it to a PC, to send the data to an AWS cloud server. It was implemented in shopping mall in Jakarta. They had a survey with the shopping mall managers to establish a better way to control the crowded parking lots of the city, to make it more organized. The findings were cross matched with their literature reviews and they had to come with a plan to make the users more flexible by adding reservation services of the parking spot.

Chapter III
Requirements / Analysis

Hardware

The project basically requires a core microprocessor/microcontroller capable of processing different types of operations simultaneously. Therefore, an 8-bit PIC 16F877A was selected for the job. The team decided to move forward with designing a PCB to incorporate the circuit to make the project uncomplicated and robust.

The hardware requirements of this project are:

- Microchip Technology PIC 16F877A 40 DIP microcontroller
- 16*2 Character LCD display module
- 4 phase stepper motor
- ULN2003 motor driver module
- IR sensor module (5 nos.)
- ESP8266-01s series Wi-fi module
- Breadboard and jumper wires
- General use PCB and soldering unit
- Green light LEDs.
- Arduino UNO board for testing
- Digital multimeter
- Crystal oscillator, capacitor, and resistors.

Software

The whole project is made in a Windows operating system, over a Linux machine as this project doesn't need a beagleboard device nor any terminal features of Linux.

Following is a list of required software for this project:

- EasyEDA – An online open source schematic and PCB design tool.
- Proteus 8 professional – used for simulating the circuit.
- Microchip technology MPLAB X IDE – for programming the PIC16877A in embedded C language.
- Thingspeak IOT webserver
- Microsoft office – for project presentation and documentation of final report.

As it is evident, two separate software are used, where EasyEDA is used for schematic design and Proteus is used for simulating the circuit. The reason is that EasyEDA made it easy to design the schematic and much more user friendly as it had the ESP8266 module in its online forum. But both the software lacked a ESP8266 module which could be simulated. Thus, a Proteus 8 professional suite was used and instead of the ESP8266-01, a virtual terminal was used which enabled the team to test the UART serial communication between PIC16877a and ESP8266 Wi-Fi module.

Power Requirement

The ESP8266-01 module works on 3.3V while the PIC 16877a MUC and other peripheral devices works on 5V. So, there are two different power requirement for the whole project. This difference in power was solved by using a 3.3V/5V Dual voltage supply.

Block Diagram and Product Development

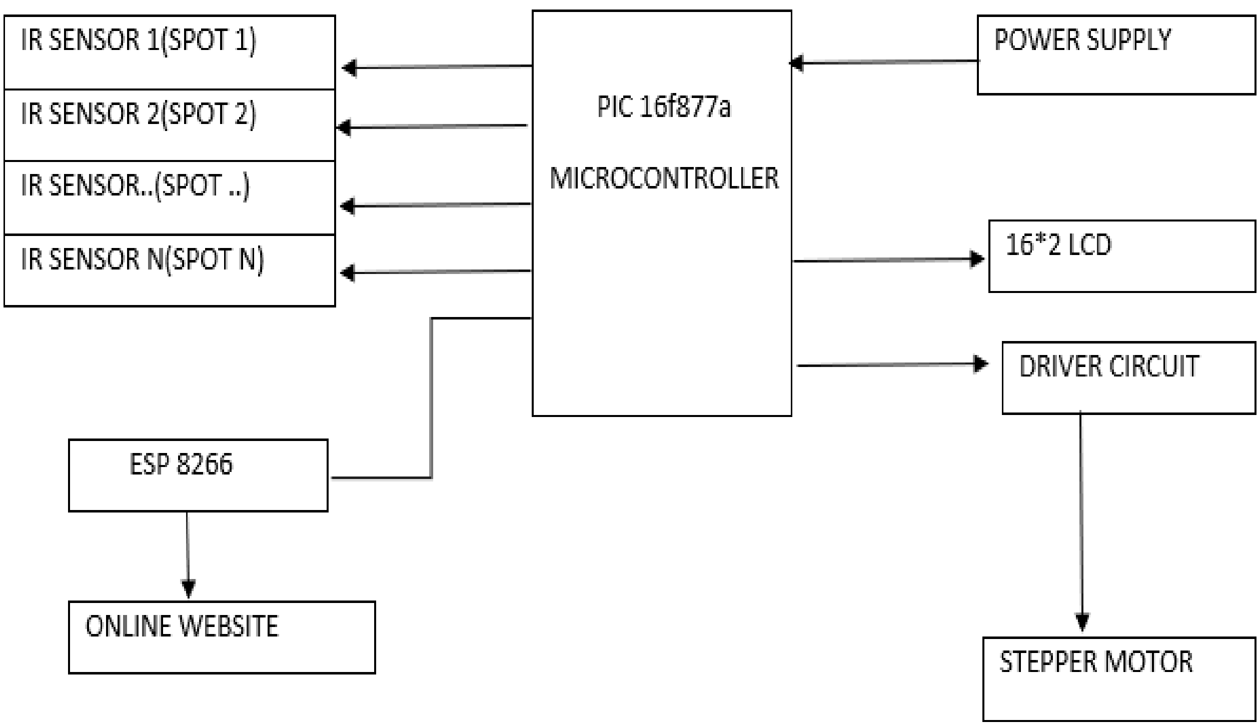

Fig 3.1: *Block Diagram of the Project*

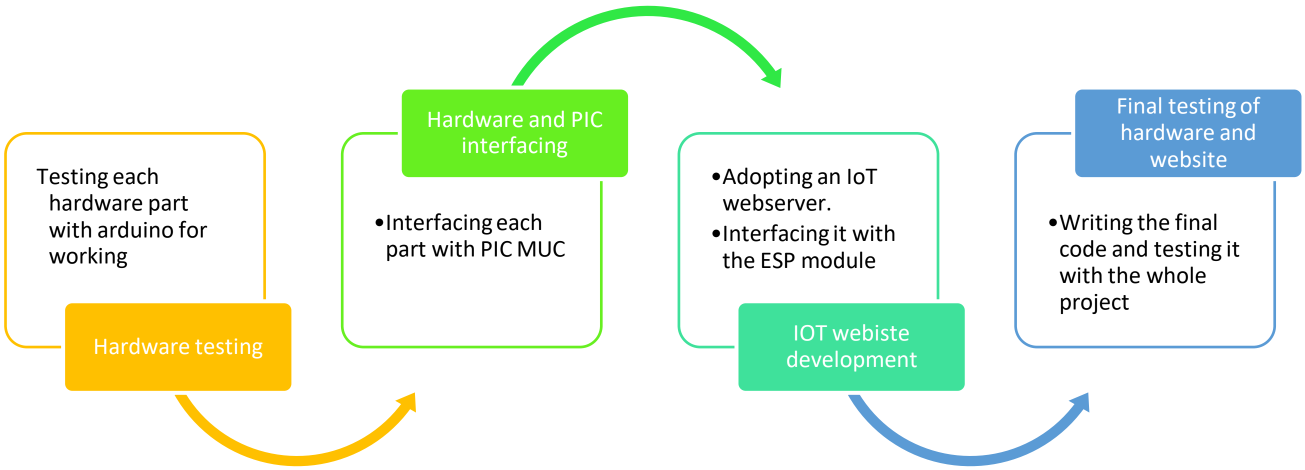

Fig 3.2 *Block Diagram of Product Development*

Schematic Layout

As mentioned in the previous section, EasyEDA, which is an open source online platform is used to design the schematic circuit diagram for the whole project. In the Fig:4.1, all the parts are interfaced with the PIC, such as the LCD display, IR sensor, Stepper motor via ULN2003 motor driver and the ESP8266-01 Wi-fi module. Single power source of 12V DC is given through a voltage regulator circuit for ESP8266-01 module and for the PIC MUC and its peripherals.

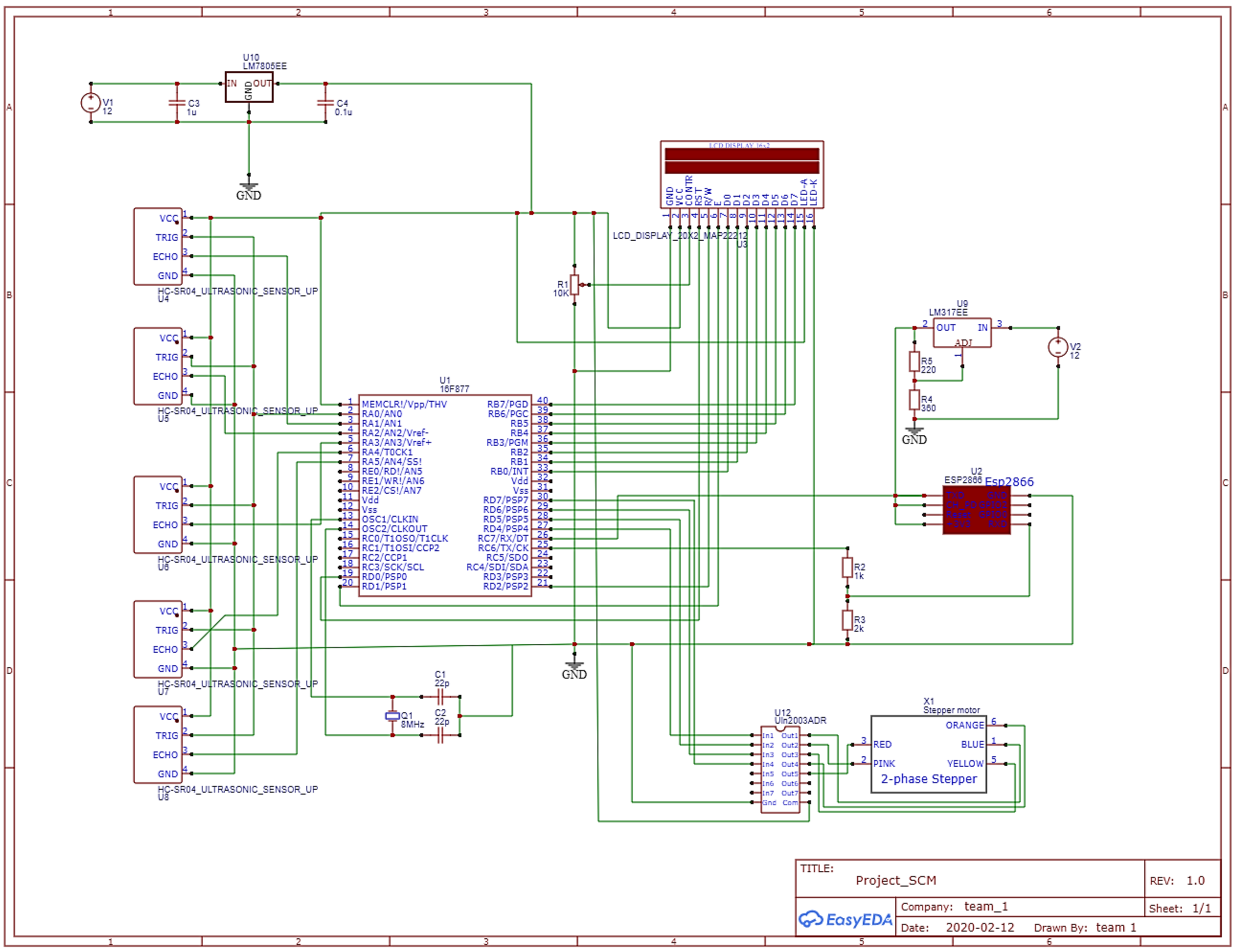

Fig 4.1: *Schematic Layout of the Project*

PIC 16F877A Microcontroller

As mentioned in the previous sections, PIC 16F877A 40 DIP MUC acts as the central processing unit of the whole project. As per the Schematic layout, the interfacing of all the peripheral devices was effortless except for the ESP8266-01 Wi-Fi module. As the Wi-Fi module has different power rating, it was a bit challenging to interface to the PIC MUC and due to several other reasons. The below shown Fig 4.2 is a 40 DIP MUC chip and the pinout diagram is shown in Fig 4.3.

Fig 4.2: *Microchip PIC16F877A 40 DIP MUC* (Microchip Technology, n.d.)

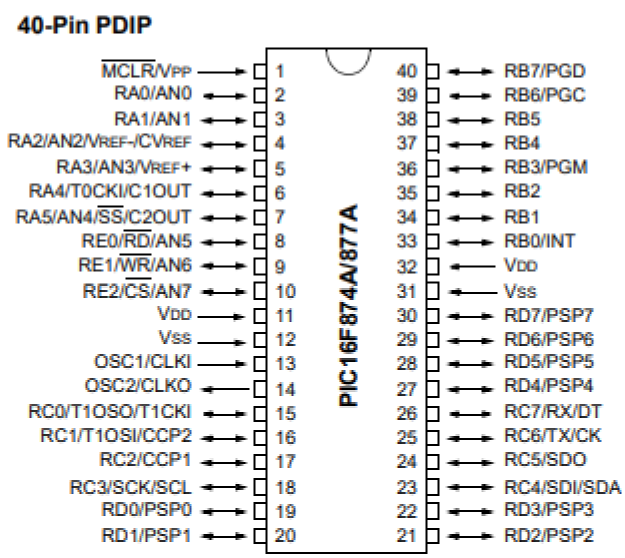

Fig 4.3: *Pinout of PIC16F877A MUC* (Microchip, 2013)

16x2 Character LCD Display

The Newhaven Display, as given in the fig 4.4 supports 2 lines with 16 alphanumeric characters with a built-in controller. The power supply for this display 5 V, with yellow/green LED backlight and is also RoHS compliant. 4-bit simple character display technique is used here where DB4-DB7 pins of LCD is connected to RB4-RB7 of PIC MCU. The pinout connections are given in the table 4.1.

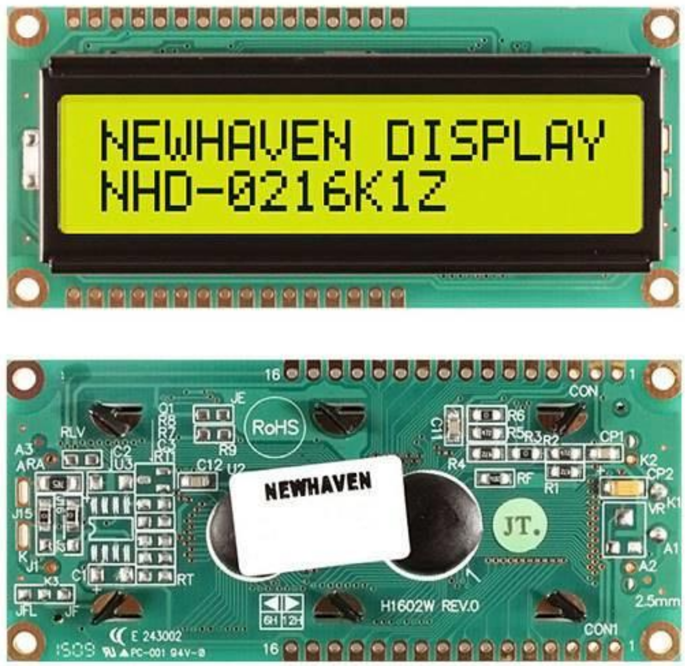

Fig 4.4: *16x2 Character display* (Digikey, 2017)

16x2 LCD Display	PIC MCU
RS	RB2 (35)
EN	RB3 (36)
D4	RB4 (37)
D5	RB5 (38)
D6	RB6 (39)
D7	RB7 (40)

Table 4.1: *Pin connections between LCD Display and PIC MCU*

Stepper Motor with ULN 2003

A 4 – phase stepper motor, as shown in the fig 4.5, is used in this project to resemble a gate lever of a parking spot. It is a unipolar stepper motor with a ULN2003 driver circuit provided with a 5V DC supply. Signals are given to inputs of the ULN2003 driver circuit, shown in the fig 4.5, and it channels it to the stepper motor by enabling it with high current signals that the MCU is not capable of. The pin connection between the PIC MCU, ULN 2003 and the stepper motor is given in table 4.2.

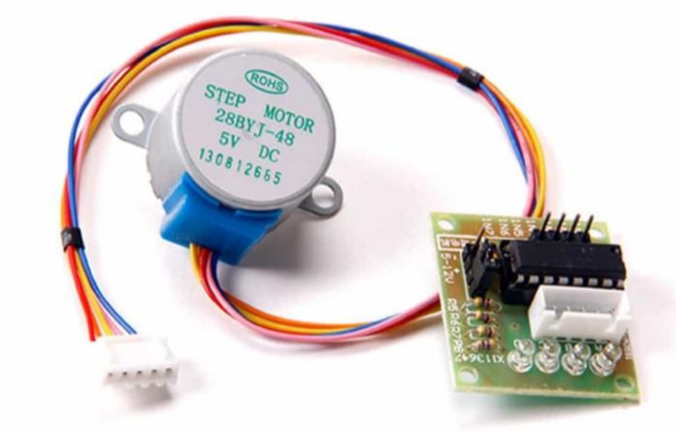

Fig 4.5: *Unipolar Stepper motor with ULN2003* (Bot Shop, n.d.)

STEPPER MOTOR	ULN2003 O/P & I/P	PIC MCU
BLUE	1	RC0 (15)
PINK	2	RC1 (16)
YELLOW	3	RC2 (17)
ORANGE	4	RC3 (18)

Table 4.2: *Pin Connections Between Stepper Motor, ULN 2003 Driver and PIC MCU*

IR Sensor Module

An IR sensor, as shown in the fig 4.6, is used to detect the obstacle in a parking spot in this project. At the beginning, for this purpose, an ultrasonic sensor was determined to detect the obstacle. But it was not detecting an obstacle within a wide range and transmitting and receiving signals was comparatively slow. So, it was replaced with an IR sensor, which proved to be working efficiently than the former. And interfacing it with PIC MCU was easier than a US sensor. The working of IR sensor is simple, that it has an IR signal transmitter and a receiver, and the output will be high if the IR receiver detects any signal within the signal range which can be changed using an onboard potentiometer.

The output of each IR sensor is connected to these pins of PIC MCU – 19 (RD0), 20(RD1), 21(RD2), 22(RD3) & 27(RD4).

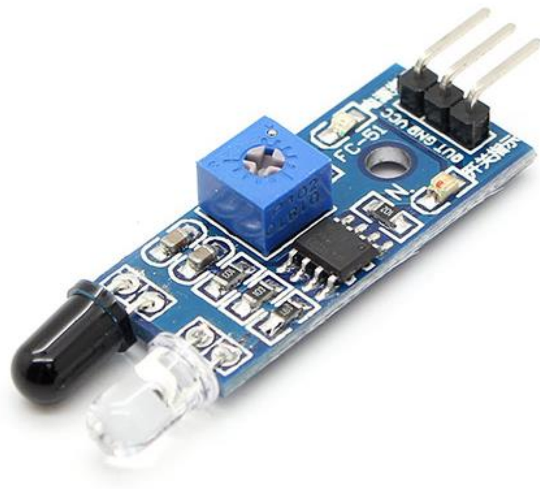

Fig 4.6: *IR sensor module* (Teczium, n.d.)

ESP8266-01 Wi-Fi Module

This ESP8266-01 Wi-Fi module microchip, shown in the fig 4.7, incorporates an ESP8266 microchip manufactured by Espressif Systems. It provides a wide range of applications to embed Wi-Fi capabilities into the system. The main advantage of this module is that it is of low cost and can also be programmed to function as a separate microchip. But in this project, it serves the function of a Wi-Fi adapter to the whole project through connecting the module to PIC through serial UART.

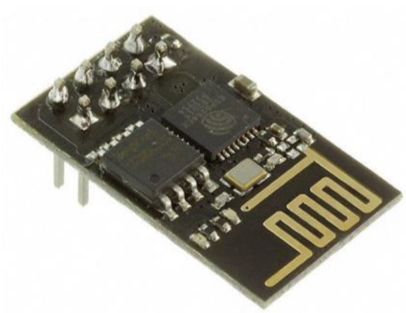

Fig 4.7: *ESP8266-01 Wi-Fi Module* (Digikey, n.d.)

The fig 4.8 shows the pin configuration of the ESP8266-01 module and the Table 4.3, given below shows the pin connections between PIC MCU and ESP8226-01 module.

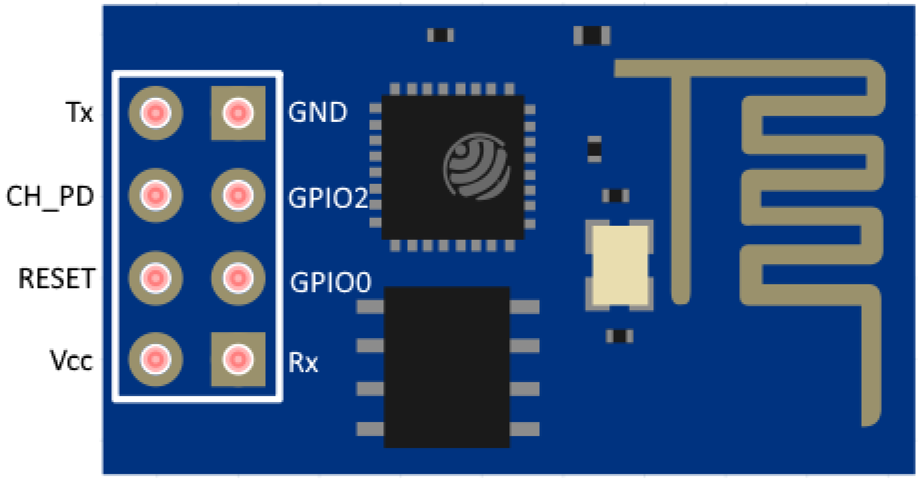

Fig 4.8: *ESP8266-01 Wi-Fi module Pinout* (Jean-Matthieu, 2016)

ESP8266-01 MODULE	Connected to
Vcc	3.3 V
CH_PD/EN	3.3 V
TX	RX pin of PIC MUC
RX	TX pin of PIC MUC
GND	GND

Table: ESP8266-01 connections

As the ESP8266-01 module operates in a voltage range of 3.3V -3.6V, all the pins should be provided only up to 3.3V, or else it will damage the module. As, the TX pin of PIC MUC has a 5V level, a voltage divider circuit is used as shown in the fig: between the TX pin of PIC MUC and RX pin of ESP8266-01 module.

The ESP8266-01 module is configured with several AT commands which is send through UART from PIC MUC. The ESP8266-01 module will respond correctly, if not, it means that the module is faulty or there is some error with the baud rate of PIC and ESP. The following table shows the basic AT commands used in the project.

AT COMMAND	RESPONSE	FUNCTION
AT	OK	Test if the system is working correctly
AT+RST	OK	Reset the module
AT+GMR	*version, OK	Print the firmware version
AT+CWMODE=*mode	OK	Set mode of operation
AT+CWJAP=*ssid, *password	OK	Commands ESP8266 to connect to a SSID with password
AT+CIPSTART=*type, *addr, *port	OK	Start a connection as client
AT+UART_CUR=*baud rate, *data bits, *stop bits, *parity bit	OK	To change the baud rate of ESP8266-01 module permanently.

Table 4.4: *AT Command List of ESP8266-01*

*version - Firmware version

*mode - 1, sets the ESP8266-01 as client

*ssid, *password – name of the SSID connection and its password

*type – string “TCP”

*addr – IP address

*port – remote port

*baud rate, *data bits, *stop bits, *parity bit – parameters to set the new baud rate of ESP8266-01

ThingSpeak for IOT Projects

It is an IOT platform, where it allows the IOT system to accumulate, visualize, analyze and store the data (ThingSpeak, n.d.). It uses MATLAB code to execute the data collected to visualize it in live stream. This platform enables the project to get the live status of the parking spot from anywhere. The basic working of this platform is by providing a channel with set of unique secret write and read API key to the client. The write key is used to write data to channel from the ESP8266-01 and the read key is used to fetch data from the client using MATLAB codes to visualize the data.

As shown in the fig 4.9, it provides a channel to visualize the data fetched from the IOT system, which here is called PARKING SPOT with its Channel ID, Author and Access details. A short description is given along with some search tags. ThingSpeak provides both a private view for the client and a public view open for the users.

The command for writing data to ThingSpeak from the IOT system is :

```
GET https://api.thingspeak.com/update?api_key=HIGI5NSCT2TNP1GG&field1=0
```

and the MATLAB function thingSpeakRead is used to read data send by the IOT system.

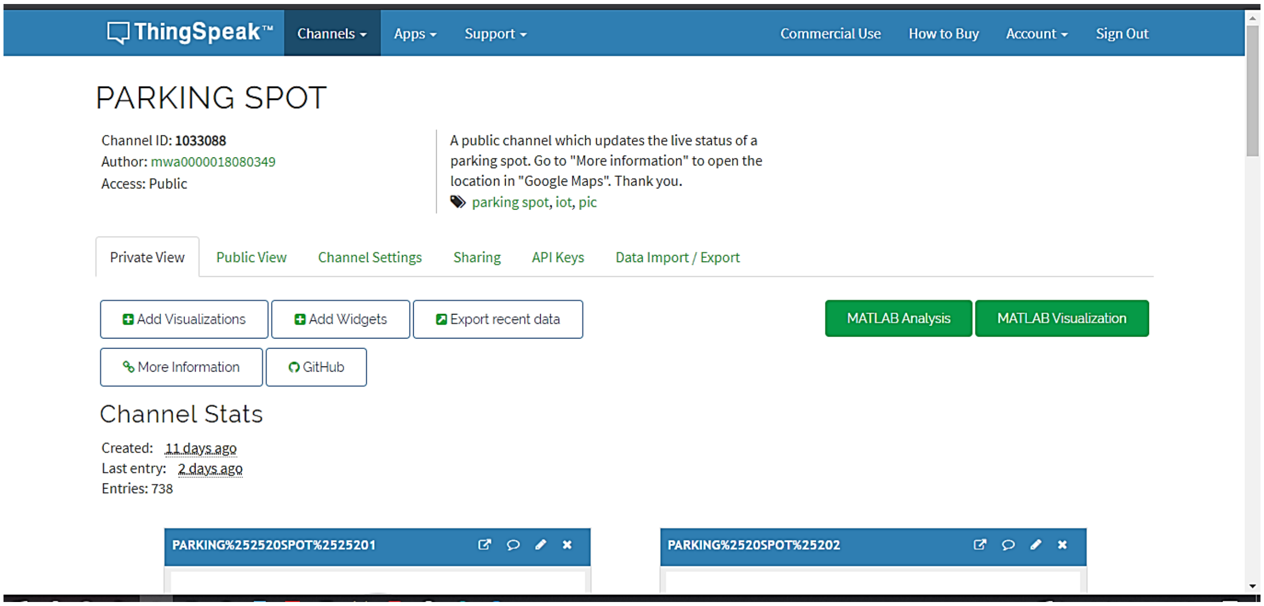

Fig 4.9: *ThingSpeak Parking Spot Channel*

Working Code

.All the working code of the project is given in the link mentioned below-
https://github.com/surindersingh51/IoT-based-automatic-parking-system-using-pic16f877a-

Chapter V
Implementation and Test

The PIC MCU and the Peripherals

As discussed earlier, the whole project is powered up by a 12 V DC adapter through a 5V/3.3V dual voltage supply. The peripheral devices such as the LCD display, stepper motor and the IR sensors are powered from the 5V supply whereas the ESP8266-01 module is by the 3.3V pin. As the PIC MUC does not need a bootup time, on powering up, it starts to detect the IR sensor and the working can be observed from the adjacent LEDs and also from the LCD display. The system is designed in such a way that, a lit LED represents a vacant parking spot, and vice versa. And the status can be viewed from the LCD display, which is also available from the ThingSpeak channel. As given from the fig 5.1, the lit LEDs represent the vacant spots and the off LEDs represents the taken spots.

Fig 5.1: *Initiating the system*

The following fig 5.2, shows the detection of obstacles by the IR sensor and displaying the result. And the status of the vacant LEDs is also visible. The status of the ThingSpeak channel is given in the following sections.

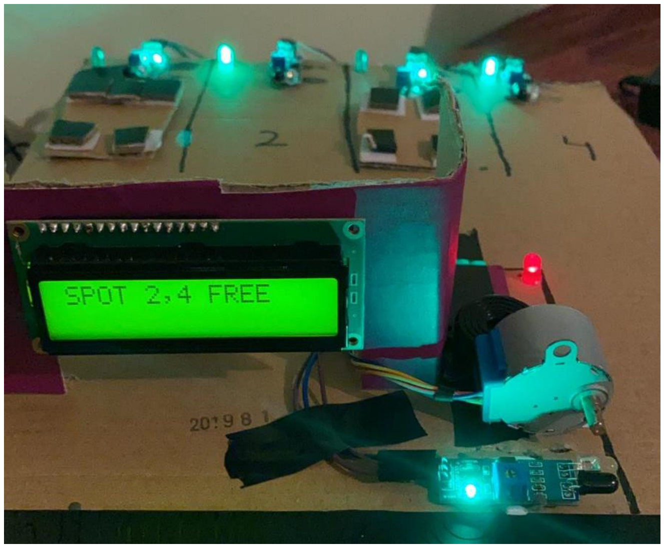

Fig 5.2: *Working of the system*

PIC MCU and ESP8266-01 Wi-Fi Module

When the module is powered up, the red on-board LED will be lit, and the blue on-board LED will be blinking whenever the TX pin is activated due to data transmission. The response of the ESP8266-01 module was tested at the beginning phase, using the serial monitor of the Arduino IDE and Arduino UNO board as shown in fig 5.3.

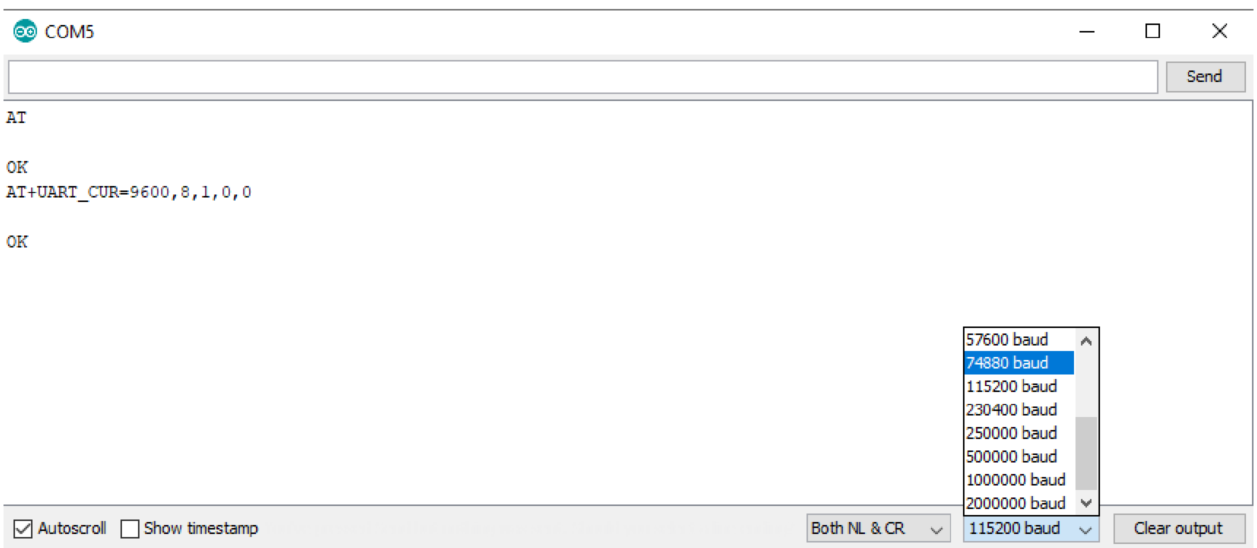

Fig 5.3: *Testing ESP8266-01 for response to AT commands*

PIC MCU, Peripherals and ThingSpeak Channel

As shown in the fig 5.4, the status of the parking spots as given in the previous was updated with a delay of 15 -20 seconds. This delay is inbuilt in the ThingSpeak application as a false obstacle like a person or stray animal give a false detection of obstacle other than a car. The indicator will only get activated if the obstacle is consistently stationary.

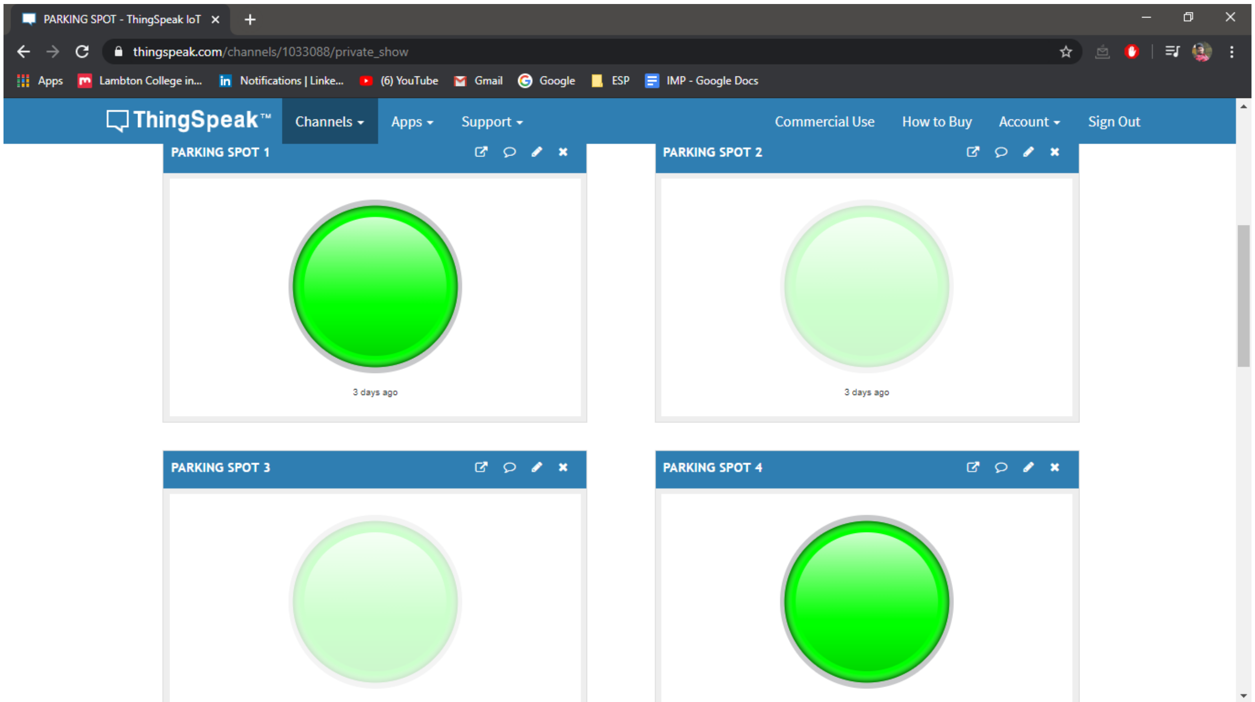

Fig 5.4: *Status of the Parking Spots*

Software Architecture

The hardware part of the project was discussed in the previous chapters I and chapter IV, where the primary control unit of the project is the PIC16F877A and the peripheral devices are the IR sensors, LCD display, stepper motor and the ESP8266-01 Wi-Fi Module. The device drivers that is used in this project are the PICKIT3 programming kit, as shown in fig 5.5. It is a family of programmers of PIC Microcontrollers made by Microchip Technology.

Fig 5.5: *PicKit3 programming kit* (Pickit 3, n.d.)

The Software part of this project comprises of MPLAB X IDE v5.35 for programming the PIC16F877A MCU. EasyEDA and Protues 8 was used for developing the schematic and simulating the whole circuit. Several Libraries are used to develop the working of the hardware parts, such as libraries for 16x2 LCD Display and PIC16F77A libraries.

In the application level of the project, the user input if processed by the IR sensor, which is send to the cloud based platform of ThingSpeak by the PIC16F877A MCU through ESP8266-01 Wifi module.

Chapter VI
Evaluation

Introduction

This section of report includes the examination of the final product for its complete working and the approaches adopted for troubleshooting throughout the product development. Some problems were prevalent related with powering up the ESP8266-01 module. Then there was a software issue with the MPLAB X IDE and with connecting the PICKIT 3 with host computer.

Minimum Requirements

The minimum requirements of the project were to develop a sensing unit using IR sensors for obstacle and sending the information to an IOT web platform (ThingSpeak) through an ESP8266-01 Wi-Fi module with a data storage facility in a user-friendly format.

Troubleshooting

The most difficult issue to solve was powering up the ESP8266-01 Wi-Fi module with the right amount of current consumption. As the team worked with this module, it was concluded that it was a sensitive chip with narrow range for operating current as the chip was replaced thrice after each one was short-circuited due to fluctuating current values.

At first, the ESP8266-01 module was tested by connecting it to Arduino UNO 3.3V source. It worked perfectly at the start but stopped responding to AT commands and was overheating. Assuming the ESP8266-01 module was drawing more current from Arduino, it was replaced with a new module. It was supplied power from a 5V DC adapter through a dual voltage supply divider. But it was also observed overheating, assuming due to the same reason as before. Without waiting for it to blow up, the third ESP8266-01 module was bought along with a 12 V adapter to supply both with power rating of 25W and output current of 2.1A.

With the 12V adapter and a 5V/3.3V dual voltage supply circuit, the ESP8266-01 module started working fine without the problem of overheating. So, it was decided to use a 12V DC adapter to supply for the whole project as it did not cause any overheating problems with ESP8266-01 Wi-Fi module.

The stepper motor which was interfaced with the PIC 16F877A stopped working at the end of the project. It is assumed that the stepper motor was faulty, as the voltage and current consumption of the motor was tested and the circuit was tested by replacing it with a buzzer. It gave output at the circuit output of the stepper motor.

Chapter VII
Conclusion

The key objective of this project was to develop an IOT based parking system. The primary focus was to establish a connection between the IOT physical system and the web platform. The user can observe the live status of the parking spot from the public PARKING SPOT channel in the Thingspeak platform.

Although there were some issues related with the power consumption of ESP8266-01 Wi-Fi module, it was mentioned in the previous section that it was solved by using a 12V DC adapter for the whole project. And the issues with connecting the PICKIT 3 to host PC was solved by installing the latest update of the software and the XC8 compiler.

Working in this project gave all the team members to have exposure to the IoT world. As all the members had previous experience in basic interfacing techniques with a microcontroller, this particular project gave the opportunity to learn about programming a PIC microcontroller in Embedded C, various functions of a popular Wi-Fi module called the ESP8266-01, which is widely used in household IoT applications and how to use a web based IoT platform like ThingSpeak to collect, analyze and store the data in cloud.

Future Work

Although the project is fully functional, its future scope is never hazed. The following lists the modification or upgrade that can be introduced to the system:

1) A user experience at the client end, so that it becomes easy to connect to the local SSID and password, by incorporating an android platform and a software application for different OS.
2) An advanced MUC with a high clock speed or a single board computer like Raspberry Pi can be used to integrate more functions.
3) A live visual can be streamed by connecting the project to any remote camera, so that the user can spot the vacant positions more accurately. There is a provision to add a live video steam to the Thingspeak platform.
4) A reservation system of parking spots in the system can also be included in this project.

Chapter VIII
User's Guide

Introduction

Thanks for choosing the Team one's IOT based Parking System, which provides a reliable parking system, by which the drivers can monitor the live status of a parking spot. Please go through the manual to install and operate the system properly.

Features

- 5 IR Sensors for 4 parking spots and at the entrance.
- 16x2 LCD display for driver's assistance.
- Barrier gate operated by a stepper motor.
- ThingSpeak channel – a web platform for live update of parking spots with information of location.

The parking system detects a car at the entrance near to the barrier gate and displays the current status of filled spots and the vacant spots on the LCD display. The same status can be viewed from anywhere from the ThingSpeak web channel mentioned above. The SSID name of the connection and the password is already set in the hardware. It will automatically connect to the Wi-Fi on powering up.

The information about the included components and the technical specifications of the system is given below in Table 8.1 and Table 8.2.

COMPONENT	FUNCTION
LCD DISPLAY	For displaying the current status of the parking lot to the drivers at the entrance
BARRIER GATE	To wait the drivers at the entrance, which is attached to a DC stepper motor.
IR SENSOR MODULES	For detecting the car in the slot
CONTROL UNIT	To collect and send the data to the ThingSpeak channel

Table 8.1: *Included Components*

COMPONENT	FUNCTION
Rated operating voltage	12 V DC
Power requirements	25 W
Maximum current	2.1 A
Working Temperature	-20℃ - 70℃
Sensor Distance Detection	2cm to 30 cm
Sensor Angle Detection	35 Degree

Table 8.2: *Technical Specifications*

Important Installation tips

- **LCD Display** – The LCD display is ought to be mounted at the entrance where it is clearly visible for the drivers. It can be mounted by drilling through the holes at the corners of the LCD module.
- **Barrier Gate –** The Barrier Gate must be placed at a 1m to 1.5m height from the ground.
- **Sensors –** Due to the chances for bogus alarms**,** the sensor modules must be mounted at height of 19” to 34” inches from the ground and the sensors should face horizontally with 90 degrees angle to the ground.

Connecting to the web platform

- The link to the ThingSpeak account is : https://thingspeak.com/channels/1033088
- This link can be shared with the drivers for them to obtain the status of the parking spots in live.
- The features of the website are:
  - Live status of the parking spots
  - Status can be obtained in a graph format also
  - Location of the spot is assigned in the channel, also available in google maps.
  - A cloud storage facility of the data and can be extracted.
  - A provision for adding live video to the channel

Chapter IX
References

- Asha, M. S. (2020). A Novel Smart Car Parking System Using IoT.
- Bot Shop. (n.d.). Retrieved from https://www.botshop.co.za/product/5v-stepper-motor-uln2003-driver-board-set/
- Digikey. (2017, December 04). Digikey. Retrieved from Kickstarter: https://www.kickstarter.com/projects/519743/sweam-smart-projector?ref=nav_search&result=project&term=sweam
- Digikey. (n.d.). *WRL-13678*. Retrieved from https://www.digikey.ca/product-detail/en/sparkfun-electronics/WRL-13678/1568-1235-ND/5725944
- G. Abhijith, H. S. (2019). In L. M. N. R. Shetty, *Emerging Research in Computing, Information, Communication and Applications, Volume 2* (pp. 465-472).
- Jean-Matthieu. (2016, February 28). *Flashing ESP8266 ESP-01*. Retrieved from IoT Experiments: https://www.iot-experiments.com/flashing-esp8266-esp01/
- Kanteti, D., Srikar, D. V., & Ramesh, T. K. (2017). Intelligent smart parking algorithm. *SmartTechCon*.
- Microchip. (2013). *PIC16877A*. Retrieved from Microchip: http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf
- Microchip Technology. (n.d.). *PIC16F877A*. Retrieved from Microchip: https://www.microchip.com/wwwproducts/en/PIC16F877A
- Mufaqih, M. S. (2020). Applying smart parking system with internet of things (IoT) design. *IOP Conference Series: Materials Science and Engineering*.
- Teczium. (n.d.). Retrieved from Teczium: https://teczium.net/product/ir-sensor/
- ThingSpeak. (n.d.). *Learn More About ThingSpeak*. Retrieved from https://thingspeak.com/pages/learn_more
- MPLAB IDE User's Guide, Microchip Technology 2006.
- PIC16F87XA Datasheet, Microchip Technology, 2013
- Espressif Smart Connectivity Platform: ESP8266-01 Datasheet, 2013