



|  |  |
| --- | --- |
| GROUP 11 | OMAR RODRIGUEZ  DANIEL RODRIGUEZ  POOYA NEJAD  JOSEPH NYAMOGO |

TABLE OF CONTENTS

1. Introduction 5

1.1 Background 5

1.2 Motivation 5

1.3 Project milestones 6

1.4 Budget and finance 6

2. Executive Summary 8

2.1 Requirements and specifications 8

2.2 design overview 8

2.2.1 Block Diagram 8

2.2.2 Occupancy and Counter Sensor Nodes 9

2.2.3 Gateway and Sub Gate 11

2.2.4 Server 13

2.2.5 User Interface 14

3. Research 15

3.1 Sensor Node 15

3.1.1 Sensor 15

3.1.1.1 Inductive Loop Sensor 16

3.1.1.2 Mechanical Switch/Generator 17

3.1.1.3 Piezoelectric Coax Cable Sensor 19

3.1.1.4 Passive Infrared Sensor (PIR) 20

3.1.1.5 Ultrasonic Sensor 21

3.1.1.6 Magnetometer Sensor 23

3.1.1.7 Further Information on the Magnetometer 24

3.1.2 Transceiver 27

3.1.2.1 Bluetooth 27

3.1.2.2 Wi-Fi 28

3.1.2.3 ZigBee 28

3.1.2.4 DASH 7 30

3.1.2.5 Z-Wave 32

3.1.2.6 Microchip MiWi 32

3.1.2.7 Part Selection 33

3.1.2.8 Protocol Selection 35

3.1.3 Microcontroller 36

3.1.3.1 Texas Instruments MSP430 37

3.1.3.2 Microchip Technology 38

3.1.4 Power Source, Power Storage and Power Management 39

3.1.4.1 Liner Generator 39

3.1.4.2 Battery 41

3.1.4.3 Solar Panel/Photovoltaic Source 44

3.1.4.4 Power Management 45

3.2 Sub gate 59

3.2.1 Wall Wart 59

3.2.2 Alternative Power Management Components 60

3.2.3 Sub Gate Power Supply Decision 60

3.3 Gateway 61

3.4 Server 61

3.4.1 Hardware Requirements 61

3.4.1.1 Linux 61

3.4.1.2 Windows 62

3.4.2 Software Requirements 62

3.4.2.1 Daemon 63

3.4.2.1.1 Java 63

3.4.2.1.2 Python 64

3.4.2.1.3 C++ 64

3.4.2.2 Data Organization 65

3.4.2.2.1 MySQL 65

3.4.2.2.2 Microsoft SQL Server 65

3.4.2.3 Web 66

3.4.2.3.1 ASP 66

3.4.2.3.2 PHP 66

3.4.2.3.3 jQuery Mobile 67

3.4.2.3.4 Platform Specific SDK 68

3.4.2.3.5 Microsoft IIS 68

3.4.2.3.6 Apache Web Server 68

3.5 User Interface 69

4. Design 71

4.1 Detection Nodes 71

4.1.1 Magnetometer Sensor 71

4.1.2 Power Management System 78

4.1.2.1 Components 78

4.1.2.1.1 Photovoltaic Cell 79

4.1.2.1.2 Battery 80

4.1.2.1.3 Power Management IC 81

4.1.2.1.4 Bill of Materials for Power Management System 82

4.1.2.2 Circuitry 82

4.1.2.2.1 Energy Harvesting Source Sub System 83

4.1.2.2.2 Battery Sub System 84

4.1.2.2.3 Unregulated Output Sub System 85

4.1.2.2.4 Regulated Output Sub System 86

4.1.2.2.5 Control Sub System 87

4.1.3 Communication Circuit Design 88

4.1.3.1 Protocol Design 89

4.1.3.2 Software Design 90

4.1.3.3 Circuit Design 92

4.1.3.4 Circuitry 93

4.1.3.5 PCB Design 94

4.1.4 Microcontroller 95

4.1.4.1 Peripheral Connections 95

4.1.5 Occupancy Sensor Node Firmware 105

4.1.5.1 Data packet 105

4.1.5.2 Functions 105

4.1.5.2.1 Raw Data Acquisition and Event Detection 106

4.1.5.2.2 Main Function 107

4.1.5.3 Counter Sensor Node Firmware 109

4.2 Sub Gate 109

4.2.1 Bill of Materials for Sub Gate 110

4.2.2 Sub Gate Firmware 111

4.3 Gateway 111

4.3.1 ZENA Wireless Adapter 112

4.4 Server 112

4.5 User interface 116

5. tESTING PROCEDURES 121

5.1 Sensor node testing 121

5.2 Power Management System testing 121

5.2.1 Voltage Sensor Test Procedures 122

5.2.2 Solar Panel Test Procedures 122

5.2.3 Regulated Output Test Procedures 122

5.3 Communication System Testing 123

5.4 Graphical User Interface testing 123

6. Build, Prototype, and Evluation plan 123

7. CONCLUSION AND FUTURE FEATURES 124

APPENDIX A 126

APPENDIX B 129

APPENDIX C 133

1. **INTRODUCTION**

Our team has designed a wireless sensor network to detect available parking spots in the UCF parking garages. The data gathered from the wireless sensor network will be transmitted to a server where it will be organized and stored. A web browser user interface will access the data from the server and display which parking spots are available in the UCF parking garages. The design objective is to give the end user real time information on all available parking spots so that they can cut their parking time in half.

**1.1 BACKGROUND**

In today’s fast paced working environment, people (motorists) greatly depend on automobiles to commute to their destinations. Automobiles include: motor vehicles, motorbikes, trucks to mention but a few. The use of these automobiles has increasingly posed a demand for infrastructure to manage the parking. All around the world, parking spaces have been constructed and control points put in place. For example, in shopping malls and airports, some control points are automated whereby users can do a self-service in the use of the parking space while others are manned by control personnel. On the other hand, parking attendants have been employed in physically controlled parking bays to direct drivers where parking is empty. [12]

Managing parking areas has been improved in ensuring that motorists can easily park and leave their destination. However, as the civilization is getting bigger every day, the next generation of parking system seems like a need for our society. With today’s technology, it is expected that a new generation of parking systems will introduce a wireless parking reservation and exhibit in to the world in very close future. It is predicted that such smart parking system will play a huge role in our society. The new parking system will save us time and let us manage our time somewhere that will help in our society and future.

**1.2 MOTIVATION**

Users of automobiles spend a lot of time in the parking bays trying to locate where to park. In today’s ever busy working environment, drivers hardly have time to spend in parking bays looking for where to park. In many places, especially around shopping complexes, universities, city centers, and many other busy working environments, finding parking has been noted as one of the major causes of stress in lives of individuals who drive. The traditional method of finding parking by the naked eye has a number of irritating situations. In situations where a driver is walking towards a car or is in the car, the other drivers waiting to find parking often make signs, or whistle or try to do something intending to ask the other whether they are pulling out. Though this kind of asking might help most of the times, it leads to situations, which are often inconveniencing to other drivers. In busy towns and cities, parking management still poses a challenge that keeps growing more complex. The need for efficient parking management systems can’t be emphasized enough for such cities. This study thus seeks to provide a solution to the issues above using the latest sensing and telecommunication technology. [12]

Moreover this project dealt with an everyday common problem that is faced by the students of the University of Central Florida every day. UCF had a student population of about 55,000 students and there are just about 15,000 parking spaces in UCF. So we can pretty much judge the challenge it would be to find an empty parking spot during regular school hours. After discussing our project each one of our group members was totally into the idea of ‘Smart Parking’. First of all if we were successful in our project we would be helping the UCF parking services in implementing a system like ours at all the different parking garages and the open parking lots. Secondly we would be helping the students conserve their time looking for parking spaces. Thirdly all four of us in our group had personally experienced this big parking issue at UCF and because of that we had sometimes arrived late in class or late for an exam or a meeting etc.

* 1. **PROJECT MILESTONES**

Although we have half a year to complete this project, if we don’t keep pace we can quickly fall behind and not meet the final deadline. To keep on track with our project and to ensure that everything is submitted on time, we created a table (Table 1.3-1) of deadlines for important milestones.

|  |  |
| --- | --- |
| **Event** | **Estimated Completion Time** |
| **Senior Design 1 (Summer 2012)** | |
| Write up Specs and Requirements | May 31st |
| Divide Responsibilities | 1st week of June |
| Begin Research | ASAP |
| Begin Writing | ASAP |
| Write up of Sensor Node Research | June |
| Write up of Wireless Protocol Research | July |
| Pick Parts based on Research | July |
| Final Draft Completed | July |
| Submit Documentation | August 3rd |
| **Senior Design 2 (Summer 2012)** | |
| Finalize Parts Accusation | August |
| Review Java and C programs | 3rd and 4th week of August |
| Start coding | September |
| Implement sensor node | September |
| Implement server and UI | October |
| Test and error checking | November |
| Finishing design | November |
| Revise Documentation | December |

*Table 1.3-1 Project Milestones*

* 1. **BUDGET AND FINANCE**

The budget for this project is $550 dollars. Table 1.4-1 breaks down the budget for the sensor node network sub system; this is the only sub system that will require financing. The project will be financed by the members of our design team. Each member will be responsible for approximately $67.46

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Qty** | **Description** | **Part Number** | **Manufacturer** | **Supplier** | **Unit Price** | **Total** |
| 1 | Thin Film Micro Energy Cell | MEC201 | Infinte Power Solutions | Digi-Key | $10.44 | $10.44 |
| 2 | 12pF Capacitor | 402 | TDK Corporation | Digi-Key | 0.003 | $0.03 |
| 2 | .01uF Capacitor | 402 | Kemet | Digi-Key | 0.002 | $0.01 |
| 9 | .1uF Capacitor | 402 | Murata Electronics | Digi-Key | 0.020 | $0.04 |
| 1 | .22uF Capacitor | 402 | TDK Corporation | Digi-Key | 0.020 | $0.04 |
| 1 | 4.7uF Capacitor | 402 | TDK Corporation | Digi-Key | 0.003 | $0.01 |
| 2 | 10uF Capacitor | 805 | Kemet | Digi-Key | 0.005 | $0.01 |
| 5 | 100uF Capacitor | 1206 | Taiyo Yuden | Digi-Key | 0.065 | $0.07 |
| 1 | RED GaAs LED | LED 1 | Lite-On Inc | Digi-Key | $0.15 | $0.15 |
| 1 | GREEN GaAs LED | LED2 | Lite-On Inc | Digi-Key | $0.15 | $0.15 |
| 1 | BLUE GaAs LED | LED3 | Lite-On Inc | Digi-Key | $0.15 | $0.15 |
| 1 | Schottky Diode | BAS70KFILM | ST Microelectronics | Digi-Key | $0.07 | $0.07 |
| 1 | DPDT Slide Switch | AYZ0202 | C & K Components | Digi-Key | $0.06 | $0.06 |
| 1 | Header, 6-Pin, Right Angle | Prog Header | FCI | Digi-Key | $0.11 | $0.11 |
| 1 | Header, 6-Pin, Right Angle | UART Header | FCI | Digi-Key | $0.11 | $0.11 |
| 2 | Header, 10-Pin, Right Angle | Header 10H | TE Connectivity | Digi-Key | $0.44 | $0.88 |
| 1 | Inductor | Ferrite Bead | TE Connectivity | Digi-Key | $0.05 | $0.05 |
| 2 | Complimentary Pair MOSFETs | FDG6322C | Fairchild Semiconductor | Digi-Key | $0.10 | $0.20 |
| 2 | Complimentary Pair MOSFETs | DMG1016UDW | Diodes Inc | Digi-Key | $0.05 | $0.10 |
| 1 | 4.7MΩ RESISTOR 0402 | ERJ-2GEJ475X | Panasonic Electronic | Digi-Key | 0.002 | $0.00 |
| 5 | 1MΩ RESISTOR 0402 | ERJ-2GEJ105X | Panasonic Electronic | Digi-Key | 0.002 | $0.01 |
| 4 | 470Ω RESISTOR 0402 | ERJ-2GEJ471X | Panasonic Electronic | Digi-Key | 0.002 | $0.01 |
| 2 | 100kΩ RESISTOR 0402 | ERJ-2GEJ104X | Panasonic Electronic | Digi-Key | 0.002 | $0.00 |
| 2 | 660Ω RESISTOR 0402 | ERJ-2RKF6650X | Panasonic Electronic | Digi-Key | 0.003 | $0.01 |
| 1 | 33Ω RESISTOR 0603 | ERJ-2GEJ330X | Panasonic Electronic | Mouser | 0.002 | $0.002 |
| 6 | 10MΩ RESISTOR 0402 | ERJ-2GEJ106X | Panasonic Electronic | Digi-Key | 0.002 | $0.01 |
| 1 | RESISTOR 0402 | ERJ-2GE0R00X | Panasonic Electronic | Digi-Key | 0.002 | $0.00 |
| 2 | 1kΩ RESISTOR 0402 | ERJ-2GEJ102X | Panasonic Electronic | Digi-Key | 0.002 | $0.00 |
| 1 | 5Ω RESISTOR 0402 | ERJ-2GEJ5R1X | Panasonic Electronic | Digi-Key | 0.002 | $0.00 |
| 2 | 10kΩ RESISTOR 0402 | ERJ-2GEJ103X | Panasonic Electronic | Digi-Key | 0.099 | $0.40 |
| 4 | Pushbutton SPST NO | PTS645SK43SMTRLFS | E-Switch | Digi-Key | 0.080 | $0.16 |
| 2 | Test Point Connectors | 1249 | "Keystone Electronics |  | 0.002 | $0.004 |
| 1 | Sensor | HMC5883L | Honeywell Microelectronics | Digi-Key | $1.59 | $1.59 |
| 1 | MCU 32KB FLASH 2KB RAM 44-TQFP | PIC24F32KA304 | Microchip Technology | Digi-Key | $2.68 | $2.68 |
| 1 | 2K SPI Bus Serial EEPROM | 25AA02E48T-I/OT | Microchip Technology | Digi-Key | $0.61 | $0.61 |
| 1 | Transceiver Module | MRF24J40MA-I/RM | MICROCHIP | Newark | $7.98 | $7.98 |
| 1 | Operational Amplifier | MCP6041T-E/OT | MICROCHIP | Newark | $0.42 | $0.42 |
| 1 | PMIC | MAX17710 | Maxim | Digi-Key | $3.75 | $3.75 |
| 1 | Sollar Panel | KXOB22-01X8 | I XYZ | Digi-Key | $2.56 | $2.56 |
| 1 | Crystal Oscillator 32KHz-XTAL | ECS-.327-12.5-34B | ECS Inc | Digi-Key | $0.60 | $0.60 |
| 1 | Zena Wireless | MRF24J40 | Microchip Technology | Digi-Key | $49.95 | $49.95 |
| 1 | PicKit 3 | PG164130 | Microchip  Technology | Digi-Key | $44.95 | $44.95 |
| 1 | PCB Boards |  | 6 PCB |  | $33.00 | $1.77 |
| Total |  |  |  |  |  | $269.90 |

*Table 1.4-1 Bill of Materials*

**2. EXECUTIVE SUMMARY**

Our team is not only interested in creating a wireless sensor network that works, but we want to create a highly efficient cost effective solution to detecting parking spots. To achieve this goal our design must be confined to certain specifications. In this section well will go through the design requirements and specifications, and we will discuss the overall design of the project.

**2.1 REQUIREMENTS AND SPECIFICATIONS**

The specifications for this project are concentrated on the wireless sensor network. This sub system is the only portion of the design that accrues cost and requires a managed power supply. Below are the requirements and specifications for the wireless sensor network.

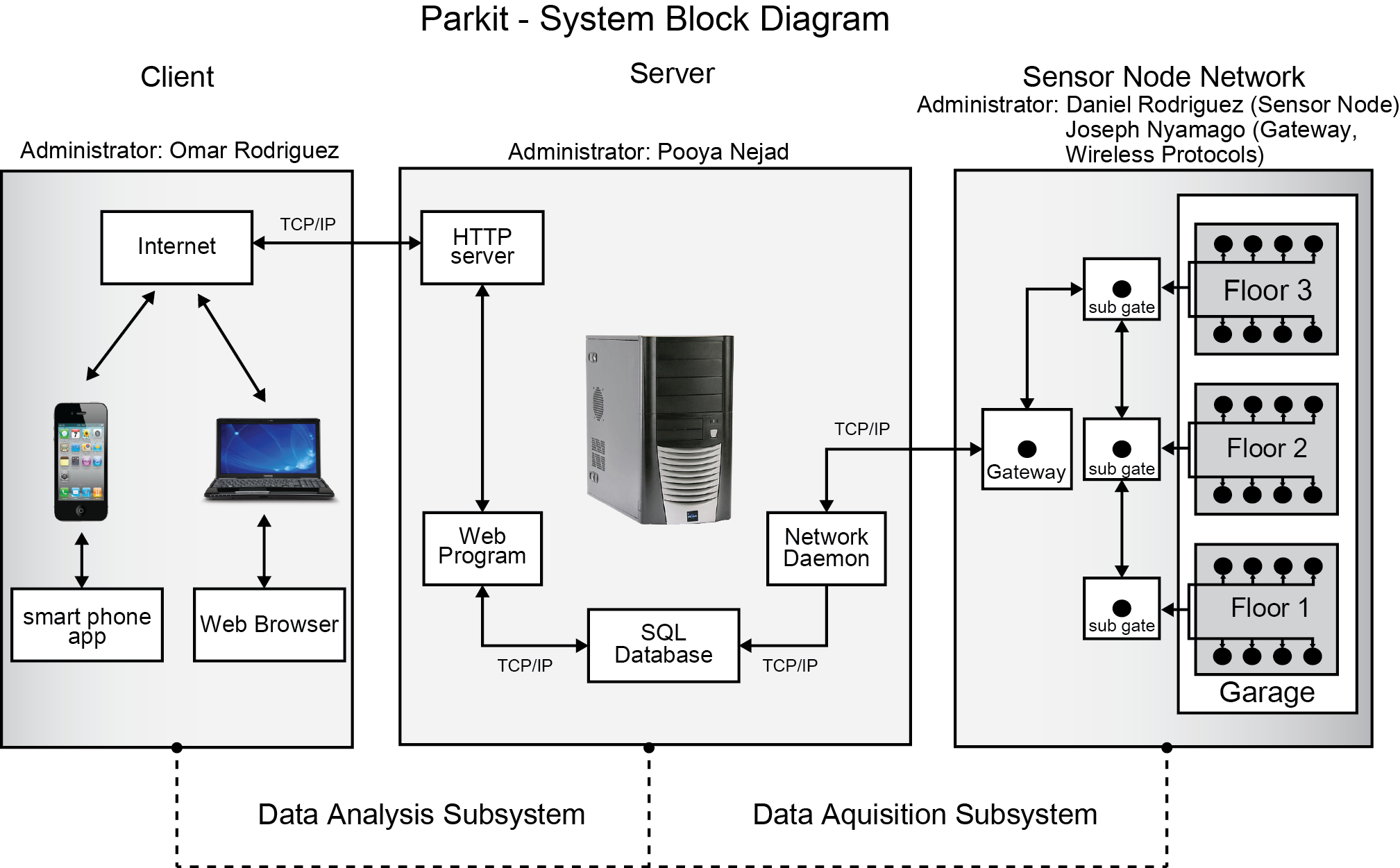
* Sensor node cost must be less than $50
* Sensor node power supply requirement, less than 5 volts
* The nodes must have an energy harvesting power source that provides perpetual power
* Wireless communication range of must reach at least 300ft for sensor nodes
* The data should have 15 seconds or less on lag time.
* 100% accuracy in car detection given proper parking conditions
* Scalable design applicable to any garage size
* Minimal maintenance ranging in the time frame of at least 10 years (e.g. replacing batteries where needed)
* Installation cost per node is not to exceed $5
* Maximum transmission current draw must be 30A
* Maximum current draw for sensor node in Idle mode bust be 10µA

**2.2 DESIGN OVERVIEW**

The Parkit design has three main sub systems, the client sub system, the server sub system, and the wireless sensor network sub system. In this section we will go over the block and state diagrams of the overall design and delve into each sub system and discuss its components in more detail.

**2.2.1 Block Diagram**

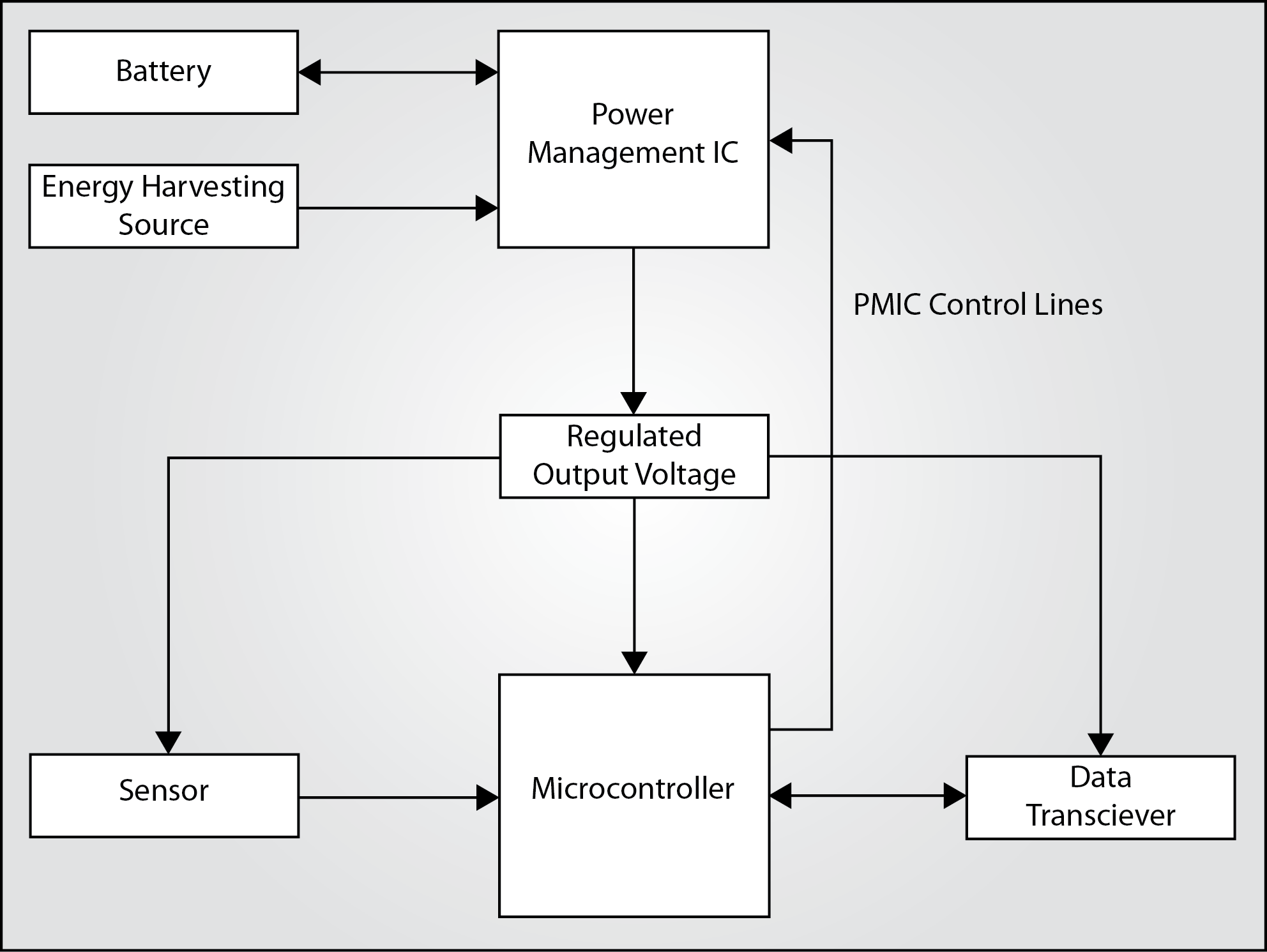
The block diagram shows a basic pictorial reference to our project. Figure 2.2.1-1 contains the block diagram for the Parkit system. On the frontend we have an iPhone application showing the user a representation of the parking lot. Behind the display will be where the server will be housed. The server will talk to the sensor network and receive information about the status of the parking spaces; it will then process the received data and feed it to the iPhone application. The wireless sensor network will branch to each floor of the garage. Each floor will have its own set of sensor nodes that detect the amount of free cars driving on the floor and also detect the occupancy of each parking spot. The data collected on each floor of the garage is sent to the sub gate. The sub gate transmits the data to the floor above it, until it reaches the gateway. The gateway then sends the data to the server to be processed.



*Figure 2.2.1-1: Complete System Block Diagram*

**2.2.2 Occupancy and Counter Sensor Nodes**

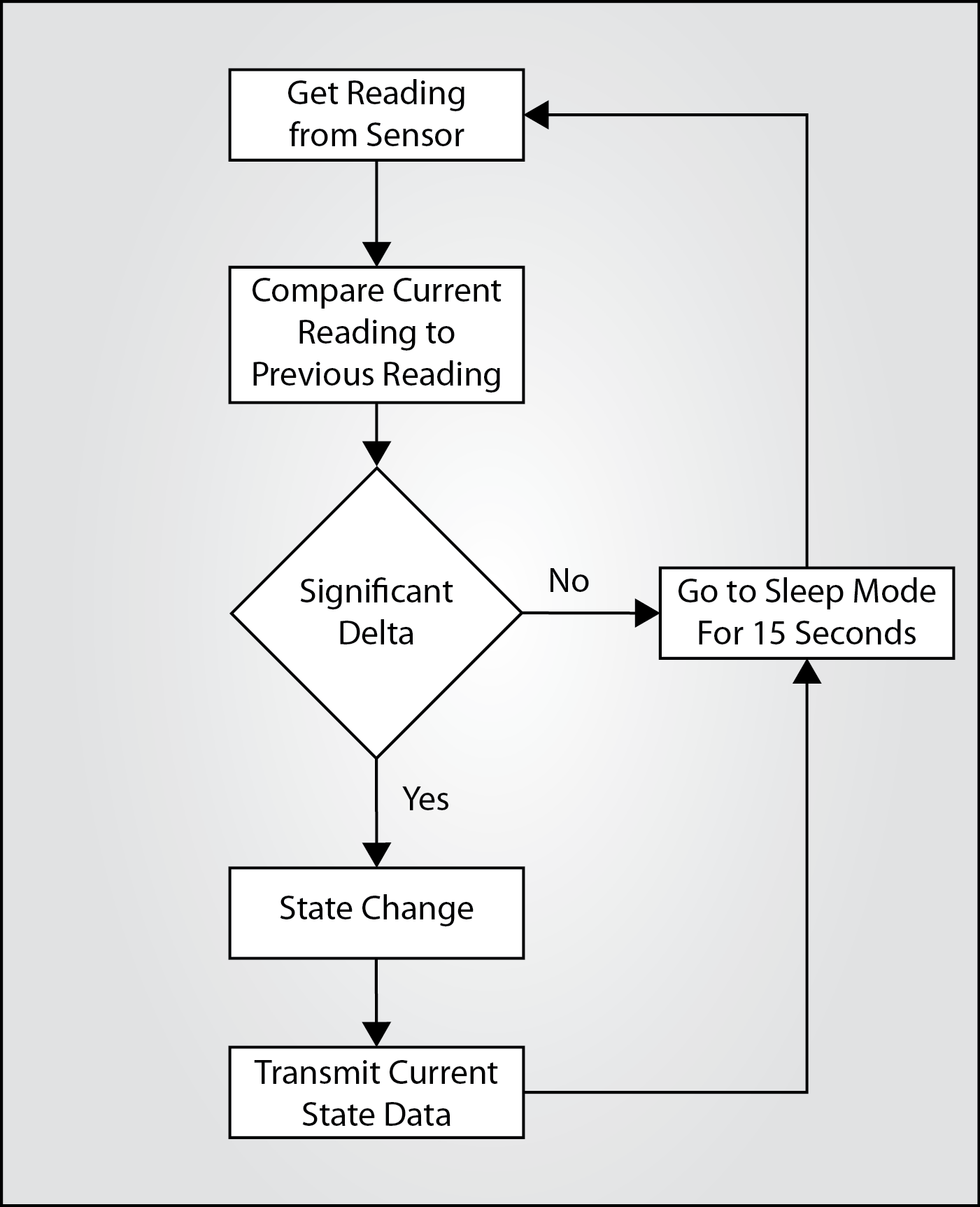
For this project we will have two main sensor nodes to detect vehicles, one will be referred to as the occupancy sensor nodes and the other will be referred to as the counter sensor nodes. A number of different sensors were considered to detect the occupancy of a parking spot. These included inductive sensors, ultrasonic sensors, pir sensors, mechanical sensors, piezoelectric sensor, and magnetometer sensors. The analysis of each sensor type can be found in section 3.1.1. The basic function of the occupancy node is to determine whether or not a space is occupied by a vehicle, whereas the counter node will be used to count the vehicles entering and exiting a garage/lot and the floor/rows. The sensors are being designed so that they will be functional and a good fit for several parking lot scenarios. Since logistics will vary, they are to work in both a garage and open lot environment. Both sensor node types will be built with the same components. The sensor node will have a battery, an energy harvesting source, a power management IC, a micro controller, a sensor, and a transceiver. Figure 2.2.2-1 contains the block diagram for the sensor node.



*Figure 2.2.2-1 Sensor Node Components Block Diagram*

The purpose of the counter nodes is to count the number of cars entering and exiting both the garages/lots and the floor/rows. The logic behind this is simple, let’s say there are three spaces available but there are six cars driving on a floor/row possibly all looking for parking. The probability of someone just entering the floor/row getting a space is lowered significantly. The same is true for an entire garage/lot; if there is 98% occupancy with more vehicles then the remaining 2% can accommodate then the probability of someone just entering the garage finding parking is also lowered significantly. The algorithm used to determine availability also considers which cars are exiting being that not all vehicles are searching for parking. The Counters will then communicate wirelessly to the sub gates to transfer their collected data.

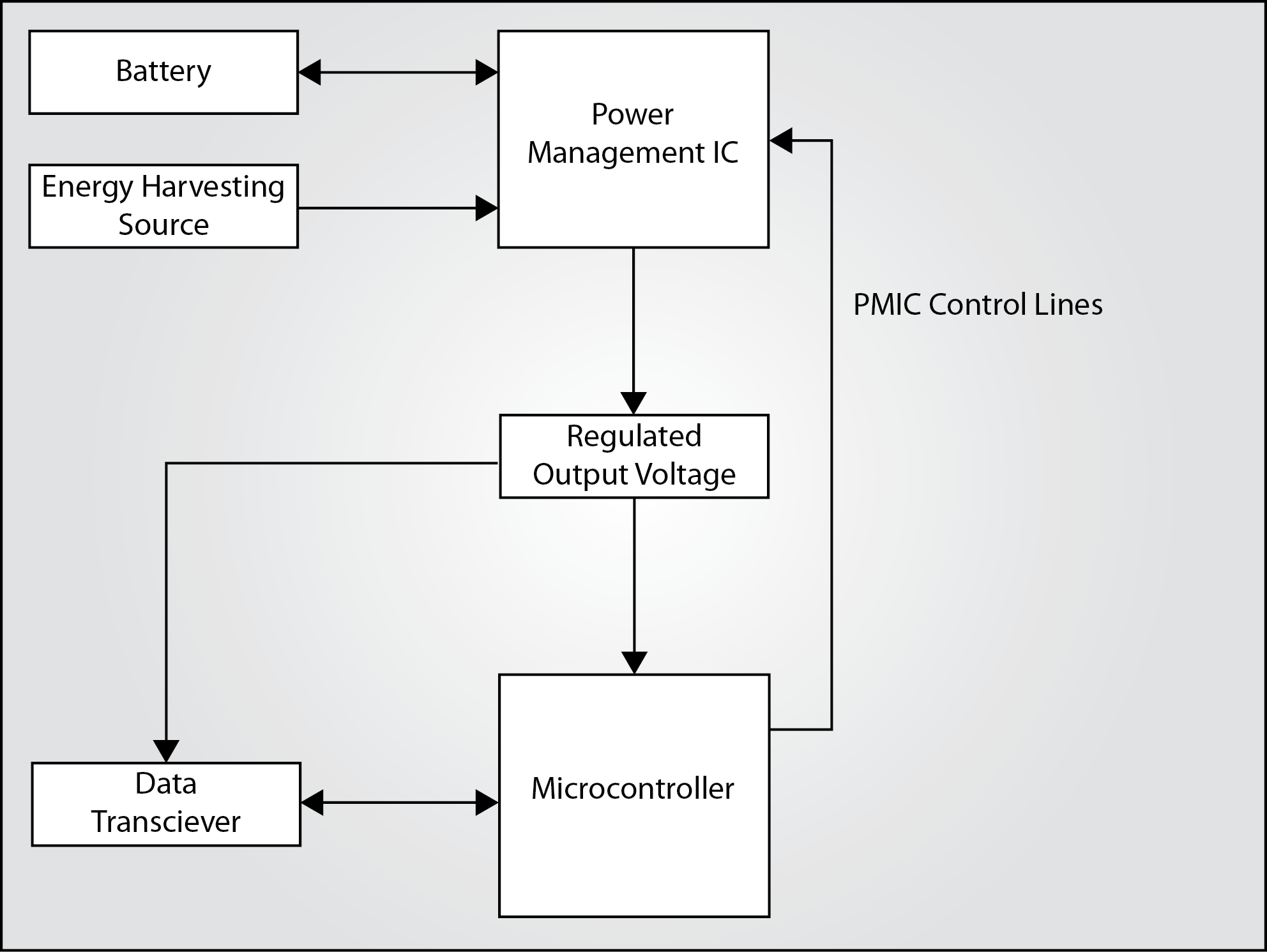
The occupancy nodes are used to determine the state of a parking space; it will be either occupied or vacant. The occupancy node samples the sensor every fifteen seconds. The data of each sample is stored in memory and compared to the previous reading. If the current reading is significantly different, the register that stores the current state of the parking spot is changed (1 for occupied, 0 for vacant), the current state of the parking spot is then transmitted to the sub gate. Figure 2.2.2-2 contains a flow chart of the occupancy sensor node. The occupancy nodes can also be pinged by commands sent by the server to enter or exit sleep mode or to transmit the current state of the parking spot



*Figure 2.2.2-2 Occupancy Sensor Flow Chart*

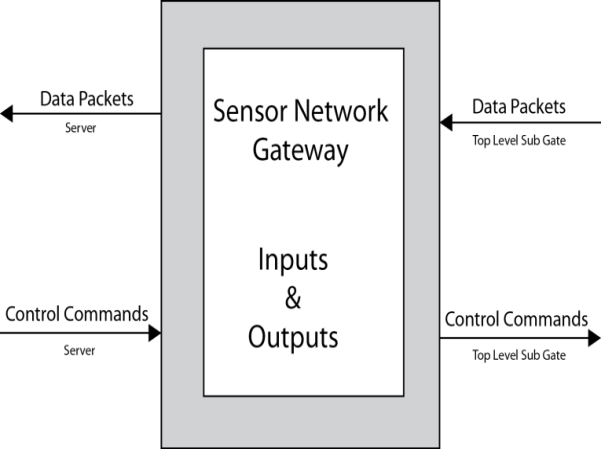
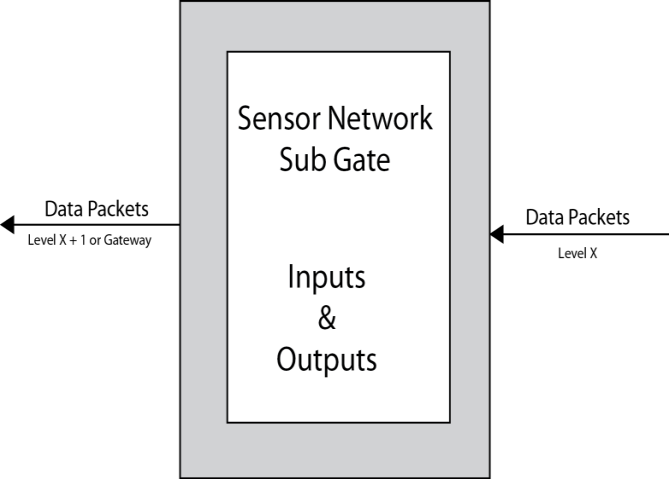
**2.2.3 Gateway and Sub Gate**

The gateway and sub gates serve as the distribution channels for the data packets collected by the sensor network and for the commands sent by the server. On each level of a garage there will be a large amount of data being collected by the network of sensor nodes. All the collected data is channeled to the floor’s singular sub gate. The sub gate has two functions, to relay the data packets, gathered from the first floor’s sensor nodes, to the sub gate located directly above it, and to relay commands from the above sub gate to all of the sensor nodes in its floor. The sub gate will contain the same hardware as the sensor nodes, minus the sensor. Figure 2.2.3-1 contains a block diagram of the sub gate.



*Figure 2.2.3-1 Sub Gate Components Block Diagram*

The gateway is the bridge that links the wireless sensor network to the server. It receives all of the commands directly from the server and channels them to the top level sub gate. It will also handle the combined data collected from all the floors of the garage and transfer them to the server. Figure 2.2.3-2 contains the block diagrams for the gateway and sub gate.



(a)

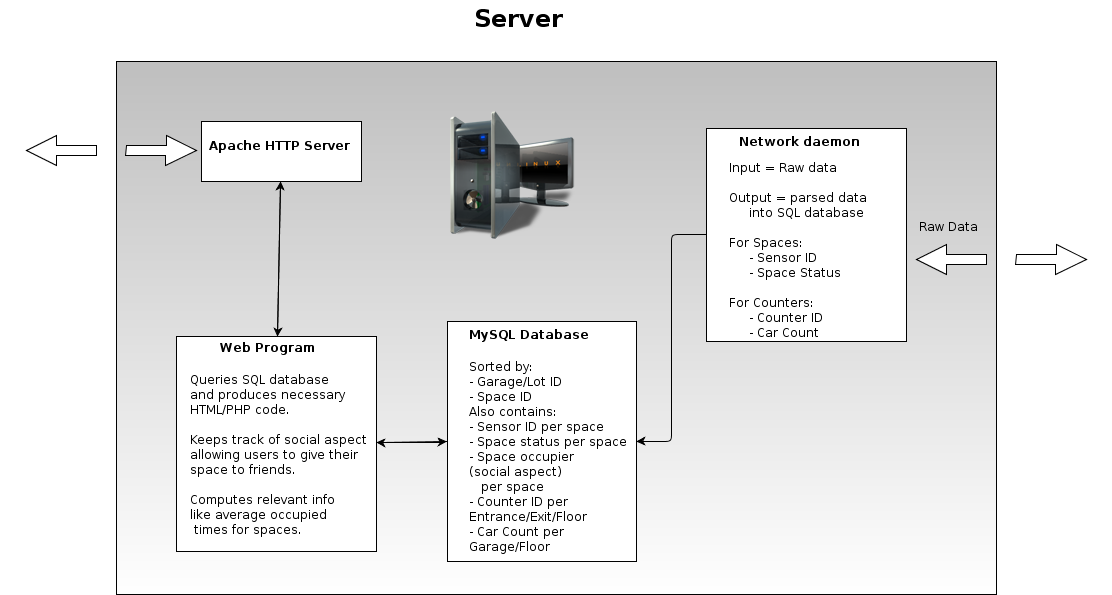
(b)

*Figure 2.2.3-2 a) Gateway Block Diagram, b) Sub Gate Block Diagram*

The gateway will be constructed differently than the sub gates. Due to the high volume of data transmissions coming from the wireless sensor network and the long distance transmission to the server, the gateway requires a larger power supply. The gateway will be plugged into the garage’s power system.

**2.2.4 Server**

The server will perform several key functions; it will execute a network daemon that will connect to specified gateway(s) in order to retrieve data from the sensor network. The daemon will need to parse data at a rate that matches the flow rate of incoming data. It will parse the data to identify: the individual sensor or counter node ID and the status/count of the nodes. Once the data is parsed it will be inserted into a MySQL database that will be sorted by garage/lot and floor/row and lastly by spaces. The sensor/counter ID’s will be linked to their respective space/entrance/exit using the front end where the web program will write the respective ID’s into the SQL database per space/counter, the daemon will then scan the SQL database to locate the relative ID where it will update the SQL entry for that space or counter based on the data received. Figure 2.2.4 contains a block diagram for the server.



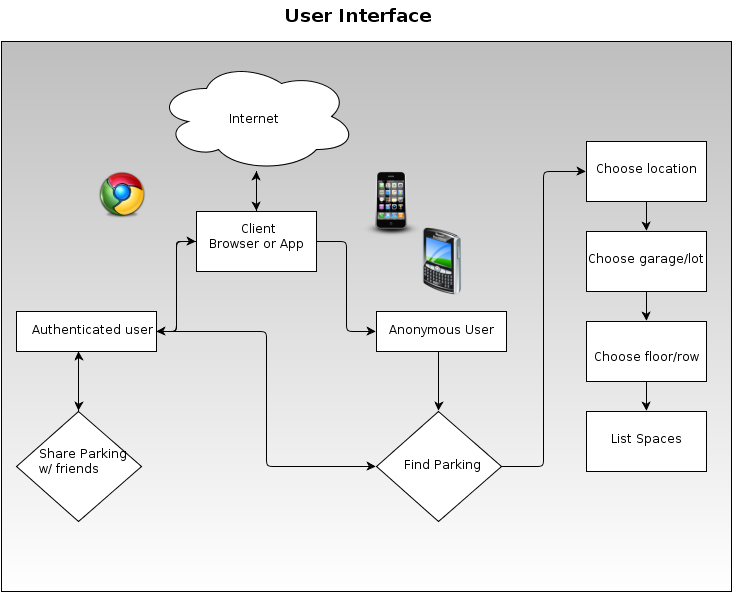
*Figure 2.2.4-1 Server Block Diagram*

The web program will be programed to retrieve necessary information from the SQL database and produce the necessary HTML to display the data to an end user. An end user does not have to be logged in to use the system however he/she will have the ability to log into the system by creating a login. Once logged into the system the end user will be able to access extra features like set an estimated time frame he/she will be using the space, friends of the end user may be able to request the space based on the time frame. For example John Doe inserts the time frame 10AM-1PM into the system for a specific space, Jane Roe then logs on and checks to see if any of her friends have a space that will be available around 1PM and finds that John will be leaving around that time. Jane can then use the front end to send John a message requesting the spot upon his departure and they arrange to meet at the spot at 1PM. However if John does not have any specific friends to offer the space to it will be available to anyone, the system can also use the estimated times of departure to estimate which garage/lot is most likely to have available parking at certain times. The web program will also have a back end where statistical data can be used to manage the system.

The web program will also check for equipment faults, it will identify which nodes have not reported a change in status/count within a pre-determined amount of time. If no change has been reported the nodes are identified and a report is generated to determine which nodes may require physical inspection, which can require maintenance or repair. Figure 2.2.4-1 contains the block diagram for the server.

**2.2.5 User Interface**

The user interface will be web based using jquery mobile, which allows for cross platform compatibility on all smart phones and tablets. The user interface will allow the end user to choose a location (UCF for example) where he/she will be parking. They will then be presented with a list of available garages/lots, after a garage/lot is chosen they will then see which floor/row has availability and finally after a floor is chosen individual parking spaces are identified. Later versions of this application might even include real depictions of the garage/lot layout and even include GPS guidance to such spots. Figure 2.2.5-1 contains a block diagram for the user interface.



*Figure 2.2.5-1 User Interface Block Diagram*

There will also be a social aspect to the user interface where the end user can log in and specify the time frame for which they will be using a space. Friends can then request the space and coordinate a time to meet at the space. If the end user decides not to limit it to friends then such information will be available for the general population using the system and his/her information will remain anonymous. Figure 2.2.5-1 contains a block diagram of the user interface.

**3. RESEARCH**

For our senior design project we are building a wireless sensor network (WSN). WSNs have a wide range of applications, such as, industrial process monitoring and control, waste water monitoring, air pollution detection, and structural monitoring to name a few. Although the application varies widely, most WSNs are built with similar structures: a sensing subsystem provides data for a server subsystem which provides data for a user interface subsystem. What makes a WSN unique is the components used to build the network. In the following sub-sections we will review the selection process for each component used for each subsystem of our WSN. The components included in this section are the (1) sensor node, (2) gateway, (3) server, and the (4) user interface.

**3.1 SENSOR NODE**

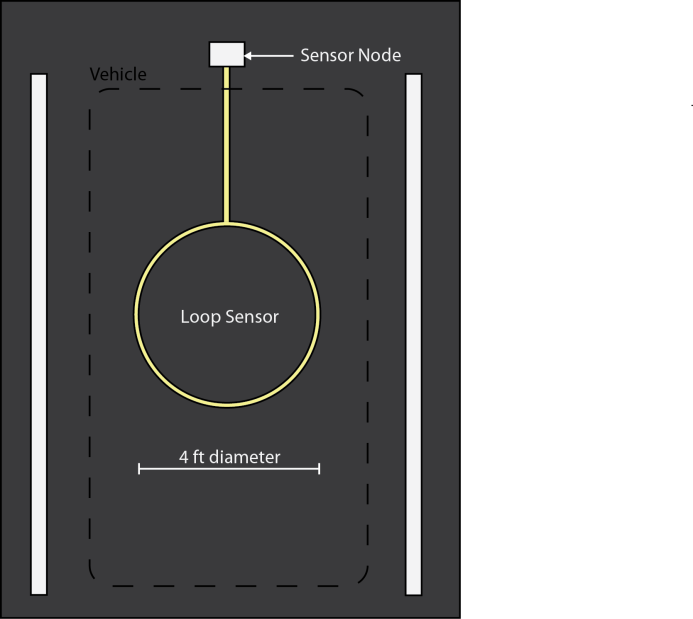
The sensor node is the most crucial component of our design. Its function is to simply monitor a parking spot. If a car is occupying a spot, the sensor in the node will recognize it and the node will then transmit that information to the gateway. Although its function is simple, due to our design specifications and requirements, it has proven to be the most complicated sub-system to research. In a real application any given garage would require thousands of nodes to monitor every parking spot available. The sheer scale of its implementation demands that each sensor node must be affordable, durable, low maintenance, energy efficient, accurate, and easy to install. This criterion was used in our research to refine the selection of parts for each sub-component of the sensor node. The sensor node can be broken into four sub-components, (1) the sensor, (2) the transceiver, (3) the microcontroller, (4) and the power supply. In this sub-section we will review how our group selected the parts which we determined to be best suited to meet our design requirements for the sensor node.

**3.1.1 Sensor**

The heart of the sensor node is the presence sensor. This component detects the presence of a vehicle in a parking spot. It sends signals to the microcontroller which indicates whether a spot is occupied or vacant. Again, the function of the sensor is simple, but due to the scale in which the end product will be implemented, the sensor must meet the criterion of our design specifications. In researching this sub-component of the node, our group took into consideration six different sensor types, which are, (1) Inductive loop, (2) Mechanical Switch/Generator, (3) Piezoelectric, (4) Infrared, (5) and Magnetometer. In the following sub-sections we will discuss why each sensor type was considered and why it was ultimately rejected or accepted as a component of the sensor node.

***3.1.1.1 Inductive Loop Sensor***

The first sensor our group considered was the inductive loop. Inductive loops are widely used to detect vehicles at traffic lights. The sensor is a coil of wire which is looped to the shape of a square or circle. When current is passed through the loop the coil produces a magnetic field. When a car is parked on top of the sensor the inductance of the circuit decreases, this is the means by which the sensor node could detect the presence of a car. Figure 3.1.1.1-1 contains a diagram of how we envisioned implementing this sensor in a parking spot.



*Figure 3.1.1.1-1 Inductive Loop Sensor: Parking Space*

There are several reasons why our group seriously considered using this sensor type for our node design. First, the circuitry of this sensor is very simple. This gives us the option of building our own sensor opposed to purchasing one. This provides us the flexibility of easily designing the loop sensor to meet our specific design requirements without being limited to the specs of a purchased product. Secondly, the cost of the sensor is within our specifications. The price would mainly be in the wire needed to create the loop. The loop requires 16 awg machine tool wire. At Grainger we could purchase 16 awg wire at 14 cents per foot. We would create a 4 foot diameter coiled loop, which would require about 12 ½ feet of wire. The loop would contain 3 turns which would produce a final cost of about $5.25, which is a competitive price compared to the other sensors types.

Another reason why we considered this sensor type is the durability of the sensor. The sesnor would be embedded in the pavement and encased in pvc conduit. This would protect it from most enviormental hazards and it would also prevent the wire from experiencing any wear and tear. Since the coiled wire is so well proctected, maintence of the sensor is reduced to nothing.

The inductive loop sensor meets three out of five design criterion. It is durable, affordable, and low maintenance. It also has the added benefit of having a simple circuitry design. In spite of this, we decided not to implement this sensor type into our final design for two reasons.

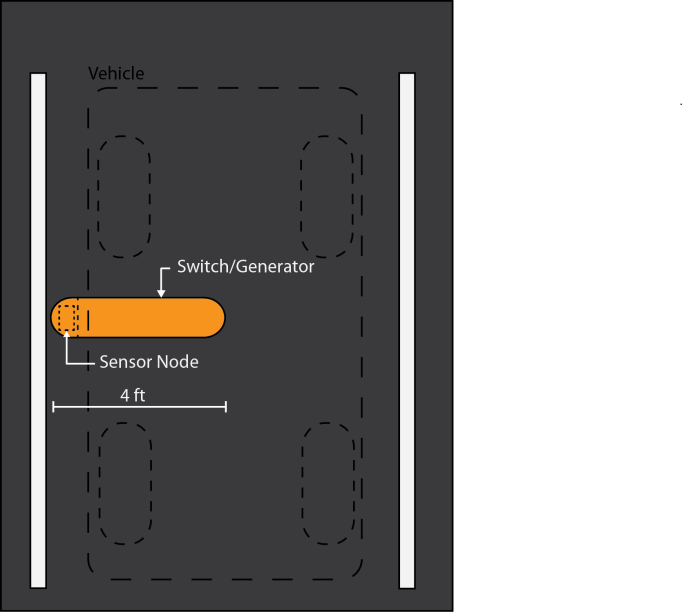
First, to generate the magnetic field required to sense the presence of a vehicle 2 feet off the ground (trucks would set the minimum sensing distance since their frames are farther from the ground), the voltage supply would have to be at least 12 volts. Since we are planning on harvesting energy from the enviornment it would not be feasible to have every parking sensor powered by 12 volts and stay within the budget alloted. Secondly, retrofitting garages for this sensor is too time consuming and expensive. Channels would have to be cut into the pavement to allow the conduit to be embedded. Also, any pre-existing wire runs through the pavement in garages would present massive work just to reroute the wires so that the sensors can be installed.

Although this sensor type meets most of our group’s design criterion, we felt that the power requirements and the obstacles presented in retrofitting this sensor into pre-existing garages, was enough to reject it for use in the final design.

***3.1.1.2 Mechanical Switch/Generator***

The next sensor our group considered was the mechanical switch/generator. This is the only sensor type we researched that was not already being used to detect vehicles and did not already exist. The sensor would be a type of “speed bump” which the vehicle would run over. The sesnor serves two puposes. It would be a switch that would be turned on when the car pulls in the parking spot and turned off when the car pulls out of the parking spot. It would also serve as a generator. Inside of the “speed bump” there would be springs, magnets, and coiled wire. When the “speed bump” is compressed the springs would cause the magnet to oscillate back and forth, creating relative motion between the magnet and coiled wire. This in turn would induce a charge which would be stored in a supercapacitor located in the sensor node. A more detailed discussion of the design of this generator can be found in section 3.1.4.1. Figure 3.1.1.2-1 contains a diagram of how we envisioned implementing this sensor in a parking spot.

There are two reasons why our group considered this sensor design. First, this is the only sensor that would not only detect the presence of a vehicle, but it also would generate power for the sensor node. Since the sensor node has a perpetual power supply, this allows us to implement a true wireless sensor network. Most of the sensors we researched required each sensor node to be wired to a common power supply unit.



*Figure 3.1.1.2-1 Mechanical Switch/Generator: Parking Space Layout*

Another benefit to this design is that the retrofit installation on existing garages would be simple. The “speed bump” would be surface mounted. Any pre-existing wire runs would not need to be rerouted. The sensor would be attached to the pavement with an adhesive (and bolts if necessary). This would greatly reduce the time and cost of installation.

The benefits of this sensor is that it is energy efficient and easy to install. Although this is the only sensor design that offered a wireless energy harvesting solution, it had several negative factors that caused it to be rejected.

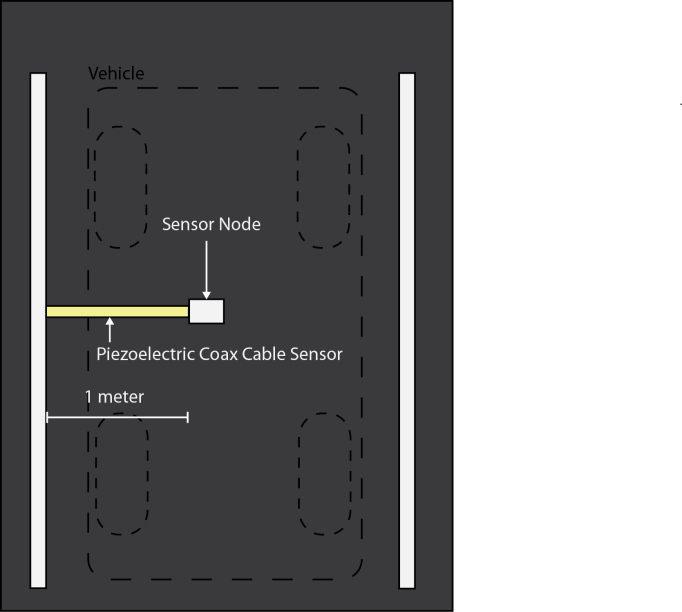
First, since the sensor requires that the vehicle runs over it, most of the components would have to be industrial grade. The external cover, springs, and internal structural support need to be sturdy enough to withstand the weight of a vehicle. Industrial parts are more expensive, which would cause our sensor budget to exceed our design specifications. Also, with the sensor constantly being compressed by the weight of the vehicles it will experience significant wear and tear with time. The life cycle of the sensor would be too short to make it cost and time effective.

Another negative with this sensor is that it is not a proven technology for detecting cars. With the other sensors we looked at, there was a significant amout of research and testing already done to prove the effectiveness of the technology. The other sensors also provide a larger body of reference designs which we could base our circuit on. With the mechanical sensor, we would have to design everything from scratch and be forced to dedicate a large portion of our time to the development of the sensor which is only one piece of a much larger design.

Although the mechanical sensor is energy efficient and easy to install, its low durability, high maintenance requirments, expensive components, and relative complex design caused our group to ultimately reject it as a viable option.

***3.1.1.3 Piezoelectric Coax Cable Sensor***

Another sensor our group considered was the piezoelectric coax cable sensor. This is a passive sensor. This sensor type was the only sensor that our group researched that did not require a power supply. This sensor is widely used for traffic monitoring applications. The piezoelectric sensor is in the form of a coax cable. When the cable is compressed it induces a voltage. The induced voltage would then trigger the circuitry in the sensor node to indicate that a vehicle has either entered or exited a parking spot. Figure 3.1.1.3-1 contains a diagram of how we envisioned implementing this sensor in a parking spot.



*Figure 3.1.1.3-1 Piezoelectric Coax Cable Sensor: Parking Space Layout*

This sensor type was considered for several reasons. First, the passive quality of this sensor made it ideal for our application. Our wireless sensor network design demands low power consumption for our sensor nodes. This sensor requires no additional power. All of our harvested energy can then be used to power the transciever and microcontroller in the sensor node. This sensor, by far, is the most energy efficient sensor our group researched.

Secondly, the coaxial design of this sensor has been tested to withstand hundreds of millions of ESAL’s. Which means that this sensor is extremely durable. In our application, this sensor would last indefinitely, the maintenance requirement would be zero. Due to the high durability and zero maintenance requirement, the installation process is tolerable. These sensors require a ¾ inch by ¾ inch channel to be sawed into the pavement,a into which the cable would be embedded and sealed with epoxy. This would make retrofitting on existing garages more time consuming and expenisve, but once installed the sensor does not need any additional care, which will save money in the long term.

This is the ideal sensor for our design, it is passive, durable, low maintenance, and relatively easy to install. Unfortunately, our group ultimately decided to reject this sensor. The price per meter for the coax cable is about $15. We would need a least a meter to ensure that the vehicle would compress the cable when entering or exiting the parking spot. There is noway we can make this sensor fit within our budget.

Although this sensor is ideal for our wireless network, it had to be rejected to stay within the budget specifications of our design. It did not qualify as a viable option for the sensor component for the sensor node.

***3.1.1.4 Passive Infrared Sensor (PIR)***

The next sensor our group considered was a short range narrow beam PIR sensor. Infrared sensors are widely used in traffic monitoring applications. The sensor detects the infrared light being emitted by an object. By using an empty parking spot as the reference infrared light measurment, we can detect vehicles when they produce a reading beyond the threshold of the reference.

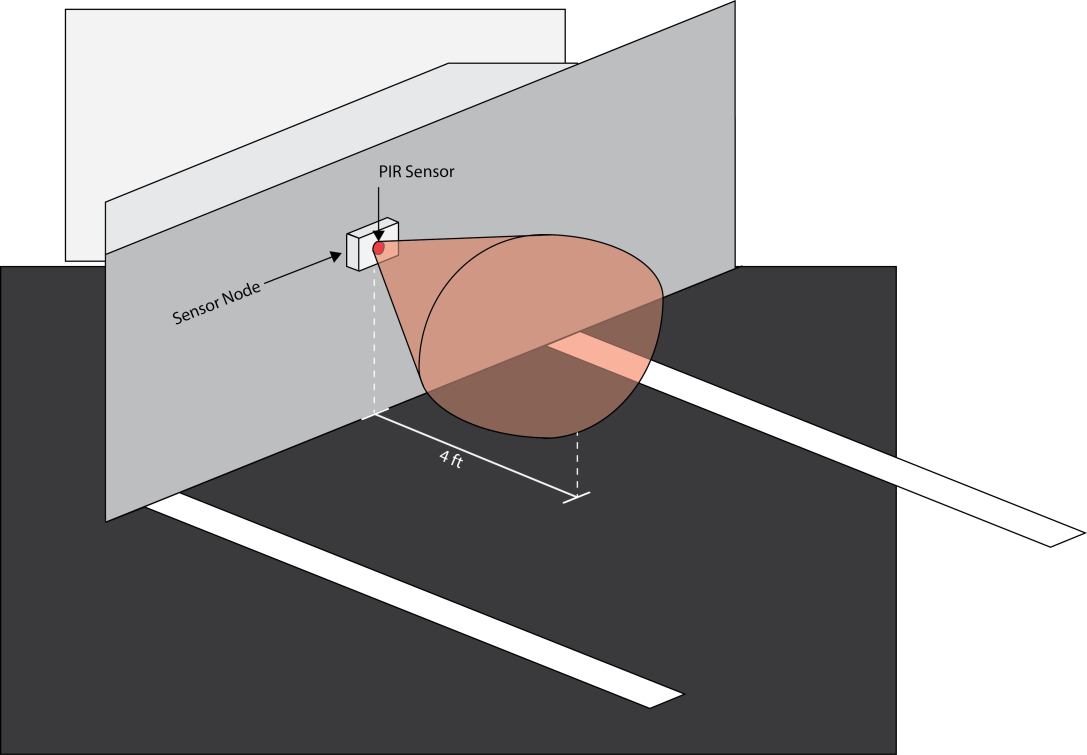
This sensor was considered as a potential option for the following reasons. First, the sensor would be durable and easy to install. The sensor would be housed in the sensor node. The node would be mounted on the wall in front of the parking space. Figure 3.1.1.4-1 illustrates the layout of the sensor in the parking space. At this location the sensor will be protected from being accidently crushed by the vehicle or walking pedestrians. If the sensor does malfunction and needs to be replaced, it is just a matter of removing a couple of screws and acessing the circutry within the sensor node. This setup woud make retrofitting the sensor network on a pre-existing garage cheap and easy.

Another reason why we considered this sensor is that it is energy efficient. The voltage supply for the sensor is 3-15V (depending on the monitoring distance). We would pulse the sensor periodically to check for occupancy. The pulse will only take a few micro seconds, which will expend minimal power.

Laslty, this sesnor is very affordable. We could purchase a sensor at Futerlec for $1.90. To protect it from the elements, we could purchase a lens for $.35 at the same reseller. The total price of $2.25 (minus supporting circuitry components) is well within our design budget for this component.

This sensor is affordable, durable, energy efficient, easy to install, and low maintenance. Although it meets most of our design specifications, we still decided to reject it due to accuracy issues. The sensor does not discriminate between vehicles and other objects. A pedestrian or animal could be standing in front of the sensor and trigger a false reading. Another problem with accuray is that the temperature of the car varies. When the car first enters the parking spot it eminates a lot of heat, after being parked for a couple of hours, the heat signature is significantly lower. This difference in heat will result in a difference of infrared light measured in the sensor which will make it more complicated to set a reference value. Due to this flaw our team decided kito reject it as a viable option for our component selection.

*Figure 3.1.1.4-1 PIR Sensor: Parking Space Layout: Source[19](permission pending)*



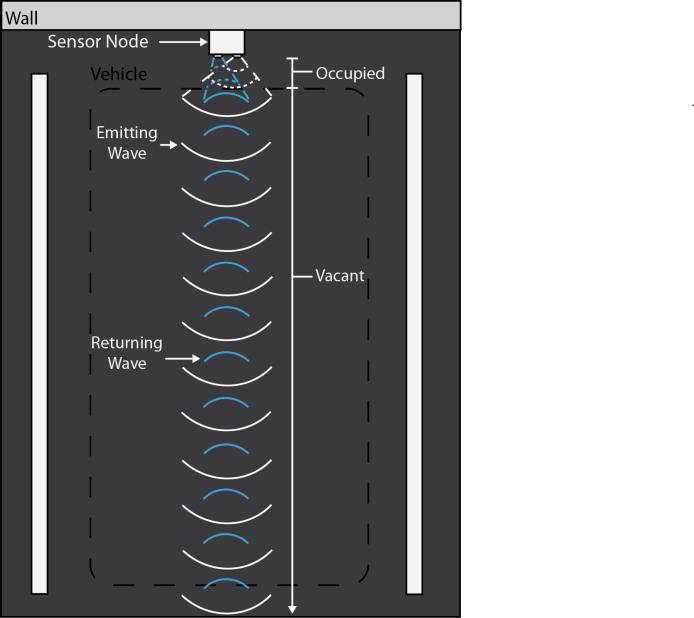
38 mm

***3.1.1.5 Ultrasonic Sensor***

Another sensor our group considered was the ultrasonic sensor. This sensor works the same as sonar. The emitter sends out an ultrasonic sound wave and the reciever measures the time it takes for the wave to return. If there is a change of time from the reference measurement (no obstruction), then an object must be in the monitored range causing the waves to bounce back faster to the reciever.

This sensor was considered as a potential option for the following reasons. First, the sensor would be durable and easy to install. Like the PIR sensor, the ultrasonice sensor would be housed in the sensor node and the node would be mounted on the wall in front of the parking space. Figure 3.1.1.5-1 illustrates the layout of the sensor in the parking space. Just like the PIR sensor this location will protect the ultrasonic sensor from being accidently crushed by the vehicle or walking pedestrians. If the sensor does malfunction and needs to be replaced, it is just a matter of removing a couple of screws and acessing the circutry within the sensor node. This setup woud make retrofitting the sensor network on a pre-existing garage cheap and easy.

The reason why this sensor would be more advantageous than the PIR sensor is that the sensor is not dependent on a changing variable such as tempearture. Which means we can monitor the space throughout the day and not have to adjust the measurements to account for enviornmental changes.



46 mm

*Figure 3.1.1.5-1 Ultrasonic Sensor: Parking Space Layout: Source[20](permission pending)*

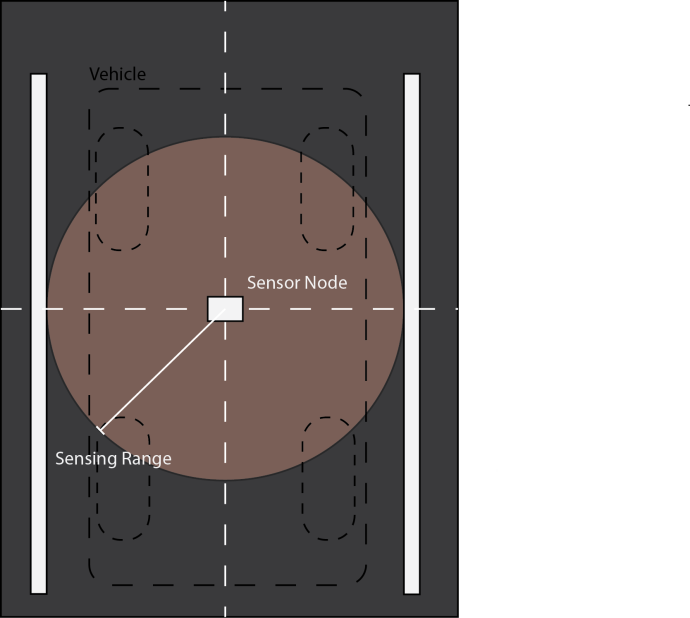
Also like the PIR sensor, the ultrasonic sensor is vulnerable to false readings. It cannot distinguish between a vehicle and any other object, including people or animals. To compensate for this, the sensor could be pulsed for a set time (1 micro second pulse, every second for 10 seconds) to check the parking spot to verify the presence of a car. Since people who trigger the sensor will most likely be walking past it, the sensor will be able to determine that it was a false reading and maintain its unoccupied state.

This sensor also meets our design’s energy requirements. It requires 5 VDC and it consumes 20 mA when in operation. Although drawing 20 mA during operation can drain our power suppy significantly, our group has designed a method to keep the power consumption low. If the sensor is pulsed at micro second intervals then the power consumtion will be low enough to be sustained by an energy harvesting power supply.

This sensor meets many of our design’s specifications and overcomes the limitations of our the previous sensors that we researched (particularly the PIR sensor). Unfortunately the sensor was rejected as a component for a sensor node for one reason only, price. The sensor itself would cost $23.99 which would make it way to expensive to ever implement on a full scale garage. For this reason we had to eliminate it as a viable option as a component to incorporate into the sensor node design.

***3.1.1.6 Magnetometer Sensor***

The sensor that we ultimately decided to utilize is the magnetometer sensor. Magnetometers are currently being used to detect vehicles in many applications, including parking and traffic monitoring systems. The sensor monitors the changes in the earth’s magnetic field. When a vehicle approaches the sensor a distortion in the earth’s magnetic field will be measured. This measurement can be used to detect the presence of the vehicle. Figure 3.1.1.6-1 illustrates how we plan to implement this sensor in the parking spot.



17.78 mm

*Figure 3.1.1.6-1 Magnetometer Sensor: Parking Space Layout: Source[21]*

The reason why we decided to select this sensor is for the following reasons. First, the sensor is less prone to false readings. Since humans don’t distort the magnetic waves of the earth the sensor will only trigger when a vehicle has come within range. This eliminates the need to add additional controls to determine if the reading was a false trigger.

Secondly, the sensor is small enough to fit within a small sturdy housing, such as a light reflector which you find on the highways. This housing will make the sensor node sturdy and durable, which will help maintain the full life cycle of the sensor node and make it less likely to be damaged by enviormental hazards. Also, fitting within this type of housing makes installation as simple as securing it to the pavement with a small amount of adhesive. Retrofitting pre-existing garages will be simple, cheap, and fast.

Another reason why this sensor was chosen is the affordability of the sensor. We can buy the sensor (HMC5883) at digikey for $3.79. This leaves us with plenty of money to buy the rest of the components of the sensor node and stay within budget.

Finally, the sensor node also meets our low energy requirments. Table 3.1.1.6-1 shows the values for the voltage supply and current draw for the HMC5883L magnetometer sensor. These values are well within the range of the energy harvesting power supply we have chosen for the sensor node (more information on the power supply can be found in section 3.1.4).

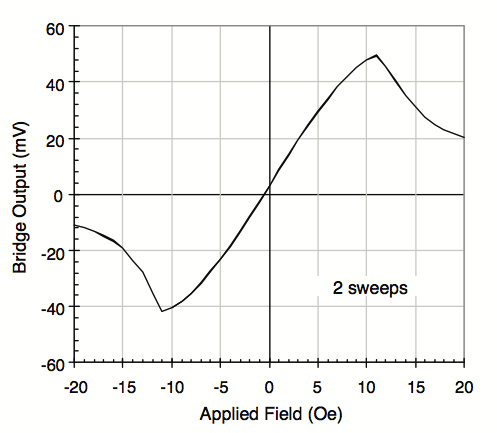
The magnetometer sensor meets all of our design specifications and requirments. It is affordable, accurate, durable, low maintenance, and energy efficient. For all these reasons we decided to choose the HMC5883 magnetometer as the sensor component for the sensor node.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Conditions | Min | Typ | Max | Units |
| Voltage Supply | AVDD Referenced to AGND  DVDD referenced to DGND | 2.5  1.6 | 1.8 | 3.3  2.0 | Volts  Volts |
| Current Draw | Sleep Mode (dual supply)  Idle Mode (dual supply)  Measurement Mode (8Hz Averaged) Dual supply (AVDD = 2.5V, DVDD = 1.8V) | -  -  - | 2.5  240  640 | -  -  - | µA  µA  µA |
|  | Sleep Mode (single supply)  Idle Mode (single supply)  Measurement Mode (8Hz Averaged)  Single supply (AVDD = 2.5V) | -  -  - | 110  340  740 | -  -  - | µA  µA  µA |

*Table 3.1.1.6-1 HMC5883L Voltage Supply and Current Draw Values Source[33]*

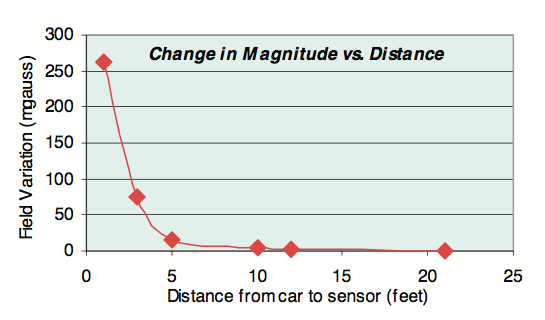
***3.1.1.7 Further Information on the Magnetometer***

The primary function of Anisotropic Magnetoresistive (AMR) sensors is to measure magnetic fields. They can be used to sense low fields (< 1 microgauss), medium fields (1 microgauss to 10 microgauss) and high fields(> 10 gauss). However, In most applications including this project, the enacting creates or modifies a magnetic field. The AMR detects the field variation, the output signal from the sensor is then processed and translated into the desired parameter value. The characteristics that will make a particular AMR sensor suitable to detect the presence of a vehicle are low cost, high sensitivity, small size noise immunity and reliability. The magneto resistive device changes resistance (delta R) when exposed to a variation in applied resistance (hence the name). This causes a corresponding change in voltage as shown in the figure below. The sensitivity is usually provided as mV/V/Gauss. The middle V refers to the bridge voltage(Vb). When the gain is set to x volts and the sensitivity is ymV/V/Gauss, then the output gain is (x\*y)mv/Oe. Through careful bridge selection, output voltages of 1 microvolt can be achieved. If the bridge output is amplified by a gain of z, then the total output sensitivity is (x\*y\*z)mV/G ~= (x\*y\*z)mV/gauss. If a range of +-i gauss is required, this corresponds to a j volt swing centered on the (x/2) volts of the bridge. The structure of the AMR is like a whetstone bridge. If the AMR detects a positive magnetic flux, the voltage at Out+ increases above Vb/2 and the voltage at Out- decreases below Vb/2. Figure 3.1.1.5-1 contains a graph of the applied field versus the bridge output.



*Figure 3.1.1.5-1 AMR output transfer curve*

During normal operation, the AMR might need to be auto-calibrated. This can be done with offset straps. Output gain variation with temperature can be reduced by using a closed loop feedback technique so that the sensor always operates in a zero filed environment. To use the AMR to detect vehicle presence, the magnetic disturbance of a large vehicle can be modeled as a composite of many dipole magnets. The dipole of these magnets causes distortions on the earth’s magnetic field which is detected by the AMR. The resulting anomaly that is unique to the shape of the car is referred to as hard iron effects. When a vehicle is present, the axis that detects magnetic field in the upward direction provides an output. This output peaks with the presence of a vehicle. Threshold levels can be established to eliminate false sensing from neighboring traffic spots. Another way to detect a vehicle would be to look at the magnitude of the magnetic variation as shown. The AMR suffers from distortions due to iron effects. The effects are hard iron and soft iron effects. Hard iron effects arise from permanent magnets and magnetized iron and steel from the vehicle being detected. These add field components along the the axes of the magnetometer having constant magnitude. Hard-iron sources have flux concentration abilities and can have remnant flux generation abilities. Flux densities is in the hundreds of gauss but most vehicles with hard-iron carry <+-2 gauss of remnant flux due to stamping of the chassis metal. Soft iron effects arise from the interaction of the earth’s magnetic field and magnetically soft material of the host. They do not have any remnant flux generated within the material. Soft-irons will concentrate the earth's magnetic flux but typically only increase the flux amplitude less than half the residual bias value at the sensor location. If the fields are concentrated in the soft-iron, they tend to de-concentrate the flux perpendicular to the field direction. The magnetic sensors will then see a few tens to hundreds of mill gauss of earth's field bias up to +-3 gauss of spikes as statistically typical vehicles come into proximity with the sensor. The likely design that will be employed for the project is to design in a +-1 dynamic range and use sudden shifts from the bias values as vehicle detection criteria.

The magnitude variation indicates the overall disturbance in the earth’s magnetic field. This magnitude produces a plot like in figure that can then be used to set the threshold levels for the AMR. Depending on the application, one might require a one, two or three axis system. The single axis system will require only one sensor, one set of sensor interface electronics and one input to digitize and place into a threshold detection algorithm. In battery applications, some AMRs can be operated on 2.7V - 3.3V. For non-portable applications, some AMRs can be operated on 4.8V-5.2V. However running some AMRs beyond 5V puts more mill watts of heat into the elements making thermal drift effects more noticeable. Also, a bridge offset voltage results for each sensor manufactured though the offset is fixed for the life of the sensor. Because the performance of some sensors can be degraded if exposed to accidental high magnetic fields, they are equipped with set/reset straps intended for pulsed currents to degauss the sensor. These fields are typically in the excess of +-10 gauss at the bridges and typically caused by magnetized hand tools, permanent magnets, portable electric motors and high currents wires. The vehicle presence is detected by using the earth’s magnetic field (about 0.5 gauss) which provides a magnetic bias point which stays substantially constant with a fixed sensor installation. Figure 3.1.1.5-2 contains a graph of the field variation versus the distance from the car to the sensor.

*Figure 3.1.1.5-2 Change of Magnitude vs Distance*

# 3.1.2 Transceiver

A transceiver is a device that transmits and receives information. Transceivers can combine transmission and reception capability on one circuit. There are basically two options for data transfer. One is wireless and the other is hard wiring. In this design we are going to use both options. Multiple sensors are hardwired to one sensor node. However, the sensor nodes will communicate wirelessly between each other and the server. There are several options for wireless communication protocols. Some of these options for the Park It system projects are Bluetooth, Wi-Fi or wireless LAN (WLAN) which operates on the 802. 11 wireless standard, ZigBee which operates on the 802. 15. 4 wireless standard, Dash 7, and Z-Wave.

***3.1.2.1 Bluetooth***

Bluetooth is a technology that allows devices to communicate with each other without cables or wires. Bluetooth low energy is designed to be very efficient at transmitting very small quantities of data at very low latencies to other devices. Bluetooth uses transceiver microchips to communicate with one another within a certain range. A typical Blue tooth device has a range of about 10 meters and can be extended to 100 meters. Some of the advantages of Bluetooth are low cost, low power consumption, and low maintenance cost which fits the basic needs of this project.

Unlike the advantages, Bluetooth has some disadvantages that prevent it from being ideal for sensor networks. There are several issues, like connection establishment delay and networking functionality that have to be solved before Bluetooth can be deployed in large sensor networks. Bluetooth architecture requires both time synchronization and frequency synchronization during its device discovery. The frequency synchronization means two devices hop to the same frequency at the same time. When both of these synchronization occur, a communication link can be established. This process becomes complicated when multiple devices exist and interfere with each other. Another problem for Bluetooth is discovery duration. It takes considerable amount of time for Bluetooth devices to find and connect to each other. The more time that it takes the more power is consumed by the devices. Both of these issues make Bluetooth a relatively poor choice for Park It project. Bluetooth is normally used indoor and inside the offices in a short range; it is not really practical to use Bluetooth for a long distances communication especially in the parking garages. Table 1 will show the specification of Bluetooth compare to ZigBee which will be discussing in details in the next section

|  |  |  |
| --- | --- | --- |
| **Feature** | **Bluetooth** | **ZigBee** |
| Battery life time | Days | Years |
| Nodes number | 7 | 64000 |
| Latency | 10 seconds | 30 ms-1s |
| Range | 1 -100 m | 1m ~ 70m |
| Extendibility | No | Yes |
| Data rate | 1Mbps | 250 Kbps |

*Table 3.1.2.1- 1 Bluetooth and ZigBee protocols. Source[31]*

## *3.1.2.2 Wi-Fi*

The popularity of wireless LANs is a testament primarily to their convenience, cost efficiency, and ease of integration with other networks and network components. The majority of computers sold to consumers today come pre-equipped with all necessary wireless LAN technology. Most modern WLANs are based on [IEEE 802.11](http://en.wikipedia.org/wiki/IEEE_802.11) standards, marketed under the [Wi-Fi](http://en.wikipedia.org/wiki/Wi-Fi) brand name. The convenience of the wireless nature of such networks allows users to access network resources from nearly any convenient location within their primary networking environment.

For a given networking situation, wireless LANs may not be desirable for a number of reasons. Most of these have to do with the inherent limitations of the technology.

The first disadvantage is the range of a common 802.11g network with standard equipment which is on the order of tens of meters. While sufficient for a typical home, it will be insufficient in a larger structure. To obtain additional range, supplementary access points will have to be purchased. It will become very expensive and costly to make these access points and it will defeat the purpose of this project.

Reliability and Security are other disadvantages of the Wi-Fi communication. Wireless networking is subject to a wide variety of interference effects that are beyond the control of the network administrator. . In a system with a potentially high number of access points, this security risk is even more of a problem.

The speed on most wireless networks is typically from 1-54 Mbps. This speed is far slower than even the slowest common wired networks which are from 100Mbps up to several Gbps. These disadvantages clearly show that Wi-Fi is not suitable for this project. Table below will show Wi-Fi protocols overview.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Protocol** | **Release** | **Frequency** | **Modulation** | **Max data rate** | **Inner range** |
| 802.11a | 1999 | 5 GHz | OFDM | 54 Mbps | 35 m |
| 802.11b | 1999 | 2.4 GHz | DSSS | 11 Mbps | 35 m |
| 802.11g | 2003 | 2.4 GHz | OFDM/DSSS | 54 Mbps | 38 m |
| 802.11n | 2009 | 2.4/ 5 GHz | OFDM | 150 Mbps | 70 m |

*Table 3.1.2.2-1: Wi-Fi protocols overview Source [16]*

## *3.1.2.3 ZigBee*

ZigBee is designed as a low-cost, low-power, low-data rate wireless mesh technology. ZigBee wireless mesh technology makes wireless sensor and control network applications practical. ZigBee looks so much simpler than Bluetooth and the operation range of ZigBee is 10-75 m compared to 10 m for Bluetooth. ZigBee device is essentially efficient in terms of battery performance. The battery lifetime is expected to be from a few months to several years when the system is in power saving modes. . ZigBee can also wake up from sleep mode in 15 msec whereas a Bluetooth device would take around 3 seconds to wake up and respond. Therefore latency will affect battery life. Basically anytime the latency of the device is increased, the battery life of the client nodes is improved. ZigBee wireless communication is also very flexible, reliable, and self-healing network. Installation and provisioning of devices can occur rapidly and without significant costs or physical construction. Installation without physical wiring will avoid the $50 to$100 cost per foot of wire (includes labor costs). Placement of sensors in optimal locations allows a network to be adaptable and reconfigurable. In addition, monitoring a large number of inexpensive sensors will offer improved control information, and capabilities to prevent failure and avoid system downtime.

It is very important to provide the sensor network with acceptable security to avoid the data from being stolen. ZigBee’s security is based on a 128-bit AES algorithm which adds on the security model provided by IEEE 802.15.4. ZigBee uses three types of keys to manage security. Theses keys are Master, Network and Link. The master keys are used as a shared secret between two devices when they perform the Key Establishment Procedure to generate link keys. The Network key performs security Network on a ZigBee network. All devices on a ZigBee network share the same key. The Link key is basically optional and maintained by the devices themselves. Link keys are used in a high security mode. These options let the user to pick and choose between the security keys that are important for their sensor network.

ZigBee depends on the basic 802.15.4 standard to establish radio performance. ZigBee is a short-range wireless and does not try to compete with a high-powered transmitter. On the other hand, it has an ultra-long battery life and low transmitter power. The standard specifies transmitter output power is at 0.5 mW with the regulator that is used on the sensor. At 0.5 mW output, single-node ranges from10 m to more than 100 m, depending on the environment, antenna, and operating frequency band. ZigBee uses the basic 802.15.4 simple transmitter with an extensible network function that allows multi-node operation. This operation can create networks hat use several nodes with cumulative ranges in to hundreds and thousands of meters. There are different modules that can be used in our project. One of these modules is Digi’s XBee Pro module that transmits data up to 1200 meters which is more than enough for the sake of this project.

|  |  |  |
| --- | --- | --- |
|  | **XBEE PRO** | **XBEE** |
| Range | 100 ft | 300 ft |
| Voltage | 3.3 V | 3.3 V |
| Transmit Current | 250 mA | 45 mA |
| Power | 710 mW | 165 mW |
| Frequency | 2.4 GHz | 2.4 GHz |
| Price | $32 | $19 |

## *Table 3.1.2.3-1 ZigBee modules Source [27]*

As shown in table 3.1.2.3, ZigBee has different modules that we can choose from for our project. The different modules are XBee and XBee Pro. Modules can cost anywhere between $20 to $200. Some of the modules are bundled with the development kit that seems like it would work for our project. Price of the development kits can be up to $300. However, having one sensor node connected to several amount of sensors will worth spending that money since we will save so much time designing the module into the design.

As mentioned before, ZigBee uses embedded applications with low data rates and low power consumption. . ZigBee’s current focus is to deﬁne a general-purpose, inexpensive, self-organizing mesh network that can be used for industrial control, embedded sensing, medical data collection, smoke and intruder warning, building automation, home automation, etc. [2].

Another value of ZigBee module is the software that is designed to be easy to develop on small and cheap microprocessors. ZigBee has been using radio design that has few analog stages and uses digital circuits when necessary. ZigBee has multiple software designs that we could use in our project which it would probably make ZigBee our number one choice for transmitting data.

## *3.1.2.4 DASH7*

Dash 7 technology is a fairly innovative method to communicate wirelessly in two major applications - Wireless Sensor Networks (WSNs) and Mobile Phones with Near Field Communications (NFC). Industry and users are very interested to make Dash 7 as a reliable method of communication that consumes low power with extended transmission range. Figure 3.1.2.4 contains a chart that compares the different capabilities of different wireless networks.



*Figure 3.1.2.4-1 Capabilities of different sensor networks* : *Source [9](permission Granted)*

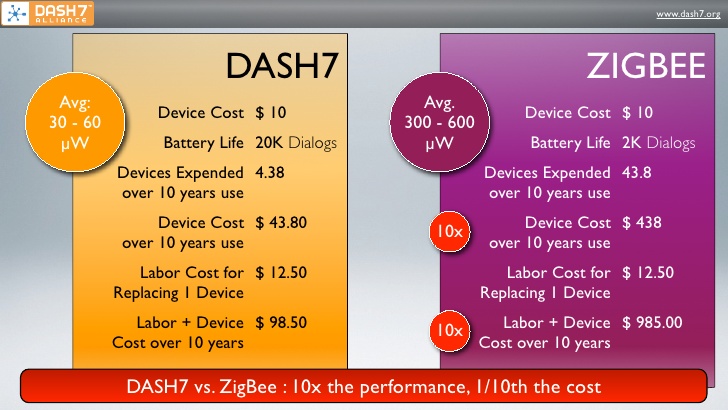
Dash 7 has been used in military for the last couple years. It was designed to provide a better communication method in terms of battery life, long range, and low latency compare to ZigBee. Dash 7 is used for devices with low power so the battery can last for years, and is suitable for large networks and for communication with things on the move.

It operates in the 433.92 MHZ which does not interfere with any other frequencies. Dash 7 is available all over the world and uses License-free spectrum. DASH7 has a longer range than ZigBee and is able to penetrate through obstacles, so also suitable for indoors. Dash 7 has a range of 1km.

Dash 7 used RFID to track and communicate with other sensor nodes in its network. The sensor network consists of a very small computer chips made of silicon with an antenna, which can send and receive data wirelessly.

Dash 7 can be used in numerous applications. As mentioned before, the U. S. Department of Defense and other militaries have used Dash 7 to improve their current infrastructure for tracking and communication. Dash 7 has also used for mobile advertising, billboards, and posters. It is also used indoors because the sensor signals are able to penetrate walls.

ZigBee and Dash 7 are both suitable for transmitting data in our project. Dash 7 has multiple advantages that are very helpful for our project such as transmitting data in a long range and also the ability to penetrate the concrete of the parking garages. However, the amount of time it takes for DASH7 device to wake up from standby mode is 2.5 to 5 seconds which is more than ZigBee devices (100 ms). Diagram below shows the cost and performance difference between ZigBee and DASH7.



*Figure 3.1.2.4-2 Differences s Between DASH7 and ZIGBEE: Source [17](permission granted)*

## *3.1.2.5 Z-Wave*

Z-Wave is a wireless communications protocol that controls home, offices, and educational facilities. This technology uses a low power radio frequency in entertainment systems and household appliances. Z-Wave makes an integrated wireless network that does not need any complicated programing and no cables to connect to the devices.

Z-Wave is a low-power wireless technology designed for remote control applications. Z-Wave operates at 908.42 MHz which is far from 2.4 GHz frequency. Therefore it will not interfere with other wireless electronics that operates in this range. Because of low power consumption and low cost, Z-Wave is used in battery operated devices such as remote controls. Its mesh network covers all areas of the home, as the radio waves travel easily through walls, floors and furniture, making connectivity 100% reliable.

Another advantage of Z-Wave is stability. Z-Wave forwards data packets between nodes that are not able to communicate directly over the air. This will increase the range of a Z-Wave network and improves stability. The range of a Z-Wave device is influenced by a number of factors, first being the presence of walls in the area. Typical reported ranges are around 30 meters (90 feet) indoors and 100 meters (300 feet) in the open air. Extending the normal range of these products is possible simply by adding more Z-Wave devices to the network. Because all Z-Wave devices are repeaters, the signal is passed along from one to the next and each time it is repeated, another 30 meters (approximately) of range is gained. Up to three additional devices (hops) can be used to extend the signal before the protocol terminates the signal.

The Z-Wave wireless mesh networking technology enables any node to talk to other adjacent nodes directly or indirectly through available relays. A master controller node controls any additional nodes. The nodes communicate directly with one another if they’re within range. If two nodes that want to communicate aren’t within range, they can link with another node that both can access and exchange information. A Z-Wave network can have up to 232 nodes. Multiple controllers can be set up to partition a network as required for different functions.

Comparing Z-Wave to ZigBee will show us that Z-wave is not as power efficient. The reason is that Z-Wave cannot act as a repeater if it is in its sleep mode; therefore, Z-Wave devices always have to be powered which will defeat the purpose of power efficiency for this project. Table below shows the differences between ZigBee and Z-wave.

|  |  |  |  |
| --- | --- | --- | --- |
| **Technology** | **Frequency** | **Data Rate** | **Range** |
| ZigBee | 2.4 to 2.483 GHz | 250 Kb/s | 10 m |
| Z-Wave | 908.42 MHz | 9.6/40 Kb/s | 30 m |

## *Table 3.1.2.5-1 ZigBee and Z-wave Protocol: Source [28]*

## *3.1.2.6 Microchip MiWi*

MiWi is Microchip’s proprietary protocols that develop wireless applications. These protocols support short range wireless networking applications. MiWi has the same principal as ZigBee in a smaller and non-secure environment. Microchip has a free source code for use of their microcontroller and transceiver. Microchip MiWi is based on IEEE 802.15.4 standard that is designed for low cost, short range, and low data rate applications. MiWi is only compatible with microchip; therefore, we can only use transceivers that are made by microchip. MiWi would fit the specification of our project because it will make a small and inexpensive network. Also, microchip offers the ZENA wireless adapter which connects USB-equipped devices with microchip wireless products for development or application users. The ZENA wireless adapter is preprogrammed with a MiWi wireless protocol application. The MiWi protocol defines three types of MiWi protocol devices, based on their functions in the network: PAN Coordinator, Coordinator and End Device. The MiWi Wireless Networking Protocol Stack functionality helps to determine the type of IEEE functionality that the device requires.

One of the modules that we can use is MRF24J40 transceiver. MRF24J40 has low power consumption usually about 18mA in RX mode, 22mA in TX mode, and 2 μA in sleep mode. This transceiver has the advantage of using MiWi and ZigBee protocol. Because ZigBee has become more complex and costly, MiWi seems like an easy solution for smaller projects like ours. A MiWi protocol system does not require any certification when a microchip’s microcontroller and MRF24J40 are used. The MRF24J40 is the first high frequency module produced by microchip. It only needs few external components to connect to the microcontroller and will provide better performance than the other IEEE 802.15.4 standard.

There are two specific terms that are considered when describing the MiWi protocol. The first term is Cluster, which refers to a grouping of nodes that form a network. A MiWi protocol cluster has 3 deep nodes that are controlled by a cluster head. The cluster head is the same as the PAN coordinator. The second term is Socket, which refers to as connector between two devices. Instead of having a hard-wired connection between devices, all the devices will communicate using common method to associate devices. We will explain more about the terminology of the protocols in the network section. Table below shows another comparison of the protocols that we are researching.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Frequency Used** | **Range** | **Average Power** | **Multi Hop Capabilities** |
| Dash 7 | 443 MHz | 1 Km | 30-60 µW | Yes |
| Zigbee | 2.4 MHz | 30-500 m | 125-400 µW | Yes |
| MiWi | 2.4 MHz | 30-500 m | 95-115 µW | No |

*Table 3.1.2.6-1 Protocol comparison: Source [29]*

## *3.1.2.7 Part Selection*

In this section we are going to choose our option to be considered to use in the final design of this project. After comparing different protocols and transceivers, we decided to pick microchip’s MRF24J40MA transceiver. This transceiver will simply connect to the microchip microcontroller to transfer data from the sensor node to the gateway.

MRF24J40MA is a 2.4 GHz transceiver module that is certified to IEEE 802.15.4 standard. It has integrated antenna and supports ZigBee, MiWi and MiWi P2P protocols. This module comes with 4-wire SPI interface and it connects to PIC24 thru UEXT connector. . It is available in a 40-pin QFN package (6x6 millimeters), has a serial interface (SPI), and supports the Media Access Control (MAC) and the Advanced Encryption Standard (AES) with hardware coding [6].



*Figure 3.1.2.7-1 Microchip MRF24J40MA module: Source [7](Permission Pending)*

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Features:** |  |  * 2.4 GHz IEEE 802.15.4 Transceiver Module * Supports ZigBee™, MiWi™, and MiWi P2P protocols FCC (USA), IC (Canada), ETSI (Europe) Certified * Integrated PCB Antenna with Simple four-wire SPI interface to PIC microcontroller * Integrated 20 MHz and 32.768 kHz oscillator drive * Low current consumption - Tx 19mA - Rx 23mA - Sleep 2uA * Hardware CSMA-CA Mechanism * Automatic ACK response * Hardware security engine (AES-128) * Automatic packet retransmit capable * Surface Mountable |

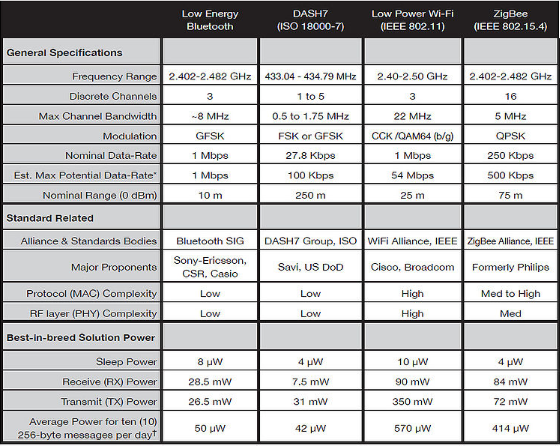
The MRF24J40MA PICtail Plus 2.4GHz Radio Frequency Card is built with the NEW agency-certified MRF24J40MA IEEE 802.15.4 RF transceiver module. The module is attached to a PICTail Plus carrier board allowing the RF Board to be used with an Explorer 16 development kit. Targeted for ZigBee, MiWi, and MiWi P2P wireless protocol applications, the MRF24J40MA is a perfect solution for low-cost, low-power complete embedded wireless PAN applications. The price of this module is $18.95 which is cheaper than the XBee module.

## *3.1.2.8 Protocol Selection*

## 

After comparing different types of protocols, we decided to choose Microchip MiWi wireless networking protocol because it provides a low cost and easy to use development with our Microchip microcontroller. MiWi protocol provides features that help implementing our wireless network easier. A network using the MiWi™ protocol is capable of having a maximum of 1024 nodes on a network. Each coordinator is capable of having 127 children, with a maximum of 8 coordinators in a network. Packets can travel a maximum of 4 hops in the network and 2 hops maximum from the PAN coordinator. With these specifications MiWi works perfectly for our design which is less than 1024 nodes. For implementing a bigger design for the whole parking lot at UCF, we need a more complex networking solution like ZigBee.

Figure 3.1.2.8-1 contains a table that shows the differences between some of the protocols that we had researched.



*Figure 3.1.2.8-1 Protocol Selection: Source [8](Permission Granted)*

The MiWi P2P protocol modifies the IEEE 802.15.4 specification’s Media Access Control (MAC) layer by adding commands that really simplifies using this protocol. However, there are some application specific issues such as when to perform energy detect scan or when to jump channels that are not defined in the protocol. The MiWi P2P wireless protocol has the following features

• Operates on Microchip PIC18, PIC24, dsPIC33 and PIC32 platforms

• Supports Microchip C18, C30 and C32 compilers

• Functions as a state machine (not RTOS-dependent)

• Supports a sleeping device at the end of the communication

• Enables Energy Detect (ED) scanning to operate on the least-noisy channel

• Provides active scan for detecting existing connections

• Enables frequency agility (channel hopping)

# 3.1.3 Microcontroller

Another component that needs to be researched for our sensor node is microcontroller. The main reason of the microcontroller is to control electronic devices. Microcontrollers are designed for small applications; however, microprocessors are used in PCs and other high performance applications. A microcontroller can consist of a single chip, or it can be composed of several components collected together on a circuit board to function as a single unit. When a microcontroller is mounted on a circuit board with other components, it is called a module or a microcontroller board. Most microcontroller modules consist of the following: a central processing unit, Input/Output (I/O) peripherals, random access memory (RAM), ROM, EPROM, EEPROM, Flash Memory, and clock. Microcontrollers used to be programmed with assembly language, now microcontrollers use high level language that makes it so much easier for the user. Computers are used to program microcontrollers, and the program is transferred to the microcontroller using serial, parallel, or USB ports.

In our project, microcontroller’s job is to receiver data from the sensors, analyze it, and transmit that information to the transceiver. One of the conditions that we have to consider in the microcontroller is the processor speed. Clock speed determines how much processing can be accomplished in a given amount of time by the microcontroller. Therefore the specific clock frequency is chosen to generate better speed.

Microcontrollers can be categorized in to internal bus width, architecture, memory and instruction set. When the ALU performs arithmetic and logical operations on a byte (8-bits) at an instruction, the microcontroller is an 8-bit microcontroller. When the ALU performs arithmetic and logical operation on a double word (32 bits) at an instruction, it is a 32-bit microcontroller. Almost all of today’s microcontrollers are based on the CISC (Complex Instruction Set Computer) concept. When a microcontroller has an instruction set that supports many addressing modes for the arithmetic and logical instruction, data transfer and memory accesses instructions, the microcontroller is said to be of CISC architecture. There is also another architecture called RISC. When a microcontroller has an instruction set that supports fewer addressing modes for the arithmetic and logical instructions and for the data transfer instructions, the microcontroller is said to be RISC architecture. [5]

There are some factors for choosing the microcontroller for our project. The goal is to find a low cost, low power, and reliable microcontroller. One of the big differences for microcontroller is how much power they consume and how much information they can store. Power consumption is probably the key feature for our design. We need to find a microcontroller that has a sleep mode where the system is on standby mode and does not consume as much power compare to when the system is on. There are some manufactures that are known as making a low power microcontrollers like TI and Microchips.

Another big factor in choosing a microcontroller is a price. In terms of price ranges they are very close to each other. MSP430 and PIC18 are around the same price of $3. The price of the microcontroller is represented by volume of the package and the number of pins that it has. Microcontroller models vary in data size from 4 to 32 bits. The 4-bits are manufactured in huge volumes for very simple applications, and 8-bit units are the most useful. 16 and 32-bit units are used in high speed control and signal processing applications.

Another important factor of the microcontroller is simplicity. We are looking for a microcontroller that can be programmed in a higher level language like C or C++. Choosing a microcontroller depends on the type of sensor and transceiver that we use in the project. In the next section different types of microcontroller is discussed and we will choose the one that best fits the design of our project.

## *3.1.3.1 Texas Instruments MSP430*

The first microcontroller manufacture that is known for low price and low power is Texas Instrument. MSP430 is the number one choice for our project. MSP430 has been around for a long time and there are lots of applications that have been done with this microcontroller. MSP430 family of processors are low power 16 bit devices. The MSP430 can be used for low powered embedded devices. The current drawn in standby mode is less than 1μA. The top CPU speed is 25 MHz the CPU speed can control the power consumption.MSP430 has the ability to go to sleep mode since the microcontroller utilizes six different low power modes that can disable unneeded clocks and CPU. While the microcontroller is on sleep the other peripherals continue to work without energy. MSP430 minimizes its average current consumption by allowing the microcontroller to stay in sleep mode for a longer period of time. MSP430 is capable of wake-up times below 1 μs.

MSP430 meets all of our design requirements in order to implement our project. There are different variety of MSP430 available which gives us lots of flexibility and advantage over the other microcontrollers.

Key features of the MSP430x1xx family include:

* Ultralow – power architecture extends battery life
* 0.1 μA RAM retention
* 0.8 μA real-time clock mode
* 250 μA / MIPS active
* 12 – bit or 10 – bit ADC --- 200 ksps, temperature sensor

Key features of the MSP430x 1xx family continued:

* Comparator – gated timers for measuring resistive elements
* 16 – bit RISC CPU enables new applications at a fraction of the code size.
* Compact core design reduces power consumption and cost
* Only 27 core instructions and seven addressing modes
* Extensive vectored – interrupt capability
* In – system programmable Flash permits flexible code changes, field upgrades and data logging.

**3.1.3.2 Microchip Technology**

Another type of microcontrollers is called PIC. There are many companies who they use PIC family microcontrollers for most of their devices. There are couple microcontrollers that we are considering to use in our project. After researching different microcontrollers from microchip website and looking at different projects, we were thinking about choosing the PIC16F877. This microcontroller has been used in variety of projects before; therefore, there is lots of information in terms of troubleshooting. This microcontroller comes with 28 pins. These chips also included transmit and receive pins which can be used for serial communication with a computer. This microcontroller is expensive compare to the other types of microcontroller that we discussed. Some of the main features of the PIC16F877 are shown below.

PIC16F877 Main Features:

* Operating speed: DC - 20 MHz clock input
* DC - 200 ns instruction cycle
* Up to 8K x 14 words of FLASH Program Memory,
* Up to 368 x 8 bytes of Data Memory (RAM) Up to 256 x 8 bytes of EEPROM Data Memory
* Interrupt capability (up to 14 sources)
* Wide operating voltage range: 2.0V to 5.5V

As continuing our research, we found fairly newer PIC microcontroller that has the capability to have a low sleep currents down to 20 nA which will be suitable for our project. Microchip Technology Inc. has expanded its 16-bit extreme low power microcontroller by adding an on-chip 12-bit ADC, EEPROM, intelligent mTouch capacitive sensing, and the capability to run from a 3.6-V supply. PIC24F32KA304 provides more support for wireless-communication protocol because it has twice as much Flash program memory and 30 percent more RAM. In this chip the number of timers and pulse-width modulators are expanded and the pin count increased to 44 pins. The MCUs’ intelligent touch sensing module includes a charge time measurement unit (CTMU) that performs automated scan in sleep mode, enabling extremely low-power capacitive sensing. This new CTMU is asserted to affectedly reduce current, thereby conserving even more battery power.

Some of the features for a Nano Watt PIC24F32A304 microcontroller are:

PIC24F32A304 Main Features:

* PIC24F products support 1.8V-3.6V operation
* 20nA Deep Sleep mode
* 25nA Sleep mode (RAM retention)
* 500nA Real Time Clock & Calendar operation in Sleep mode
* 400nA Watch Dog Timer operation in Sleep modes
* 1uS wake-up from Sleep

PIC24F32A304 Main Features continued:

* 50nA I/O port leakage
* 200uA at 1MHz Run mode
* Power Modes: Run, Doze, Idle, Sleep, Deep Sleep
* System Supervisors: Low Power BOR, WDT, INT0 and RTCC
* Internal oscillator support - 31 kHz to 8 MHz, up to 32 MHz with 4X PLL
* Fail-Safe Clock Monitor – allows safe shutdown if clock fails

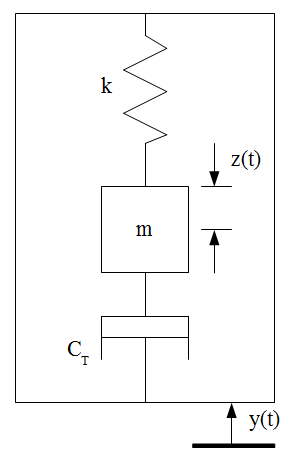
The chances that we are going to pick this microcontroller are very high. This microcontroller fits all of our ideal specifications, which is low cost, low power, and reliable.

**3.1.4 Power Source, Power Storage and Power Management**

One of the objectives of this project is low power and efficient use of power. To power the detection nodes, many power sources were considered. They are ambient RF power, linear electromagnetic power and the ubiquitous solar panel. Also, research into the various methods of power storage was considered. Finally, to get the power to the storage device, proper and efficient management of the power is required. The function of transferring power from source to the storage is performed by the charge controller. To perform this function, the charge controller will need the appropriate hardware and software. These components of the charge controller will be discussed.

***3.1.4.1 Linear Generator***

The linear generator would be placed underneath the bar that the vehicle would drive over. The electromagnetic linear generator(LG) can be designed without contacts between parts. The LG is composed of two parts. The first is the scheme to convert vibration energy to electrical energy. A power management circuit to maintain the largest dc current to charge a super capacitor. Four connected magnets move relatively to two coils under the effects of two springs. The simple model of the LG can be modeled as a mass-spring-damper system as shown in the figure below.



*Fig. 3.1.4.1.1 Linear Generator model*

For the LG, the mass of the vibration source is assumed to be much greater than the mass of the generator and the vibration source an infinite source of power. Also, the movement of the source needs to be unaffected by the energy extracted by the LG. The differential equation that describes the movement of the mass with respect to the platform can be described with



Where, z = relative displacement mass(m), k = spring constant(N/m), cm = mechanical damping coefficient due to friction and cogging force, m = mass(kg) and y(t) = the input frequency. Writing the equation above in frequency domain we get,



and for the transfer function we get



As the slides move back and forth, the magnetic field changes position and consequently, the flux linkage of the copper coils varies according to the flux linkage equation. This in turn induces a voltage according to Faraday’s voltage law. If a simple resistive load is attached to the LG, the electrical system is a first order electrical circuit with the coils inductance L in series with load resistance Rl and coil resistance r. The induced current in the coils is now given by



The total electromechanical force on the coil is



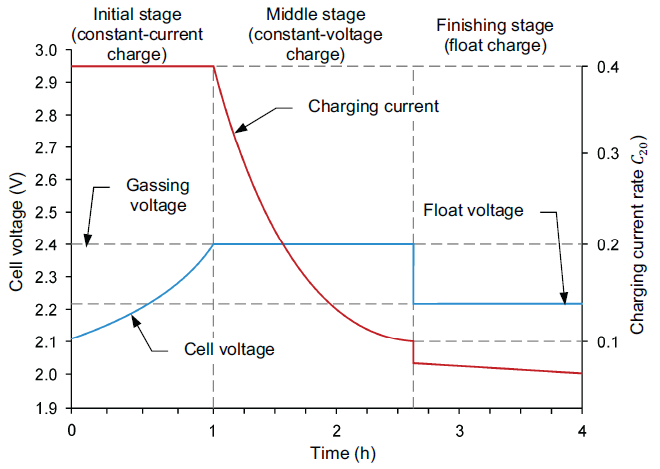
Where Fs is electromechanical force generated by the induced current on one side of the coil. This force opposes the relative motion between the magnet and the coils. The vibration frequency of the platform will range from 2 to 5Hz, the inductive impedance is much lower than the resistive impedance and can be neglected. For the LG to be able to be provide enough power for the detection unit, it has to have the power output optimized. The output power is affected by the spring constant, load resistance, induced voltage and harmonics. To find the maximum converted electrical power one can compute the power consumed by the electrical damping force in the second order system. The average power is then computed. For a low damping factor, there is a peak in the average power around the natural frequency. When the vibration frequencies are concentrated around the natural frequency, a low damping factor increases the power generated at that point. When the main vibration frequency varies over time, the peak caused by the low damping factor is narrow. The damping factor should therefore be increased to get broadband power. For a lightly damped LG, the potential for high power output is increased. If there is a small difference between the natural frequency and the vibration frequency, the power output drops significantly. To prevent this problem, it is necessary to match the vibration frequency of the platform. One way to achieve this is to actively tune the natural frequency of the LG by changing the spring constant. A type of spring with a varying spring constant is a progressive spring. The flexibility of tuning an LG makes it outperform other converters (electrostatic, piezo electric) in terms of maximum power conversion. Harmonics can deform the voltage waveform by introducing local maxima. A process should be used to pick out the frequency which carries the largest power among the spectrum and set k to resonant at this frequency.

***3.1.4.2 Battery***

The batteries are the primary storage for the sensor node. Selecting the appropriate type of battery results in longer life for the sensor node. It will also affect the ability of the sensors to be properly powered and send the proper signals back to the MCU. The three key factors that will affect the type of battery chosen for the project are: Energy density (size and weight), the charge/discharge cycles(life cycle) and capacity(operational capacity without recharge). The various rechargeable batteries that will be considered for this project are listed below. Nickel cadmium(NiCd), nickel metal hydride(NiMH), sealed lead-acid(SLA) and lithium-ion(Li-On).

*Alkaline*: These batteries are not rechargeable but are used because they have the advantage being common and are readily useable. The rechargeable alkaline types are not practical since they degrade quickly after a few charge cycles.

*SLA(Sealed Lead Acid)*:Lead-acid are the most common due to their wide availability in many sizes, low cost and well understood performance characteristics. They also have an advantage of low self discharge. Their main disadvantage is having a low energy density and low discharge and charge cycles. They are also not environmentally friendly. There are different types of Lead-Acid battery classifications that will be considered. The first is starting, lighting and ignition (SLI). They are primarily useful for shallow cycle service. They are useful in small PV systems where the average daily depth discharge is limited to 10-20% and the maximum allowable depth of discharge is limited to 40-60%. Lead-Antimony batteries have the advantage of greater mechanical strength, excellent deep discharge and high discharge rate performance. They have a good lifetime when operated at high temperatures. Their disadvantages are high self-discharge rate and require frequent water additions. The other type of battery is a Lead-Calcium. It’s advantages are, mechanical strength, low self discharge and reduced gassing. The disadvantages are poor charge acceptance after deep discharge and shortened life at higher operating temperatures if discharged to greater than 25% depth of discharge repeatedly. Captive Electrolyte Lead-Acid batteries have the advantages of being portable and spill proof making them ideal for remote applications where maintenance is infrequent or unavailable, less susceptible to freezing. The disadvantage is that it is intolerant to overcharging, affected by high temperature and requires precise voltage equation. The recommended charging algorithm is constant voltage. The first point to consider is that SLA batteries self-discharge at approximately 5% per month. It is recommended therefore that the batteries should be recharged when they reach 70% of their capacity. Also, the ambient temperature affects the battery charging conditions. As the temperature rises, the electrochemical activity in the battery increases and so the charging voltage should be reduced to prevent overcharge. On the contrary, as the temperature decreases, the charging voltage is increased to prevent undercharging. After considering the various charging methods, the best approach is to utilize the modified constant-voltage charging method (fast charging method). This charging method utilizes three stages. In the first stage (initial stage), the charging begins with constant current until a certain voltage (gassing voltage) is reached. The charger then maintains this voltage until the current decreases to a value equal to 0.1\*the charging current. At this point, the voltage is reduced to a lower voltage (float voltage) that is completes and maintains the battery charge. This process can take a long time so to reduce the charging time, higher initial current and voltage is used. This process is depicted in the diagram below. The fast charge current is a function of temperature and therefore should be reduced according to the typical temperature coefficient of 0.3% per °C. The maximum temperature recommended for fast charging is 50° (122°F). Maintenance charging however, can generally proceed above that temperature. Since incorrectly charging a battery could result in damaging the battery or an explosion, it is important to understand the battery’s profile. The battery profile that will be used is that of the SLA.



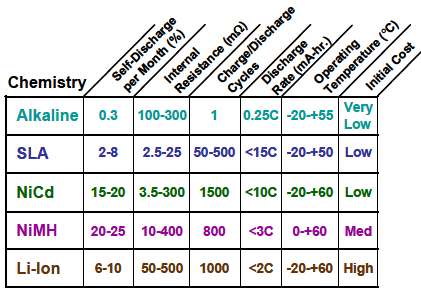
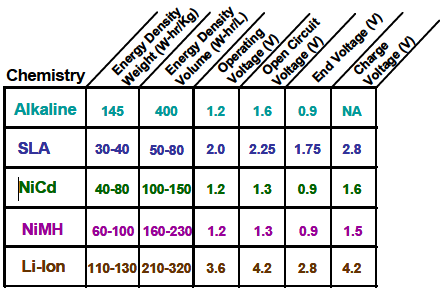
*Fig 3.1.4.2.1: Modified constant-voltage charging profile*

*NiCd(Nickel-Cadmium)*: Among the rechargeable batteries, NiCd have the best charge and discharge cycle. They provide good substitute for alkaline batteries since they have the same profile. They are good for a medium cost project since it is a technology that matured. The NiCd battery has some advantages over the lead-acid types discussed above including long life, low maintenance, able to tolerate excessive discharges, excellent low temperature capacity retention and non-critical voltage regulation requirements. One of their disadvantages is that they require to be charged and discharged regularly due to the memory effect. They also have a high self discharge rate and are high cost compared to the SLA. They are used in a few critical low temperature applications but have a high initial startup cost.

*NiMH(Nickel-Metal Hydride)*: An improvement over the NiCd chemistry above is the NiMH. It is made of environmentally friendly material,it has higher energy density and shares the same charge algorithm as the NiCd so no modification is required to a battery charger that is designed for a NiCd. However when being charged, NiMH produce a lot of heat and therefore require a complex charging algorithm. It is also difficult to detect the change in voltage.

*Li-Ion(Lithium Ion)*: Has the advantage of higher volt per cell, low rate of self discharge, low maintenance and high energy density. It has the disadvantage of aging with time even if rarely used. It also needs circuit protection for overvoltage when charging and under-voltage when discharging. Shown below are tables tht compare the different battery chemistries that can be used to power the sensor node.

Therefore to determine the right type of battery, the performance characteristics of the battery, as shown below, will be considered. Ampere-Hour, Capacity, cut off voltage, cycle, discharge, charge, rate of charge/discharge, negative (-), positive (+), open circuit voltage. The other factor that will be considered is the procedure used to charge the battery. These are: bulk or normal charge, float or finishing charge, equalizing charge. Another factor that will be considered is battery discharging which consists of: depth of discharge (DOD) (which further consists of allowable DOD and average daily DOD), state of charge (SOC), autonomy, self discharge, battery lifetime, temperature effects, effects of discharge rates, corrosion, Type of system and mode of operation, charging characteristics; internal resistance, required days of storage (autonomy),Amount and variability of discharge current, Maximum allowable depth of discharge, Daily depth of discharge requirements, Temperature and environmental conditions, Energy storage density, Size and weight, Gassing characteristics, Availability of auxiliary hardware, Terminal configuration, Reputation of manufacturer, Cost and warranty.



*Fig 3.1.4.2.2: Battery comparisons*

***3.1.4.3 Solar Panel/Photovoltaic(PV) Source***

Photovoltaic energy source is suitable for the following reasons. It is a sustainable source, it is environmentally friendly, suitable for remotely powered devices, for low power devices it can provide more than enough power. A PV source however, has some drawbacks that need to be taken into consideration when utilizing this source to provide power. First of all, the power that is generated from the PV source varies due to the following factors: solar irradiation, temperature, type of silicon used in the PV. The are three types of PV panels that result in different factors that need to be taken into consideration. The three types are armophous, polycrystalline and monocrystalline.

For the photovoltaic cell, the following illuminated values are considered when selecting the PV cell: Open Circuit Voltage (VOC), Short Circuit Current (ISC), Maximum Power (PMAX), Current at PMAX (IMP), Voltage at PMAX (V MP), Fill Factor (FF), Shunt Resistance (RSH), [Series Resistance (RS)](http://zone.ni.com/devzone/cda/tut/p/id/7230/_Toc194890976), Maximum Efficiency (ηMAX). The following no-light values are considered: shunt resistance (RSH), series resistance(RS). Other factors to consider are as listed below.

*Cost*: First and foremost cost is one the most important factors to consider since it directly affects the objectives of the project. It is important to have a low cost of PV panels to have an overall low cost of the project. Amorphous panels are the least expensive followed by polycrystalline then followed by monocrystalline panels.

*Size and Weight*: These two parameters affect the outcome of the project because they will affect how the entire hardware will be mounted and where it can be mounted.

*Voltage output*: higher voltage output means that smaller diameter wires can be used. This reduces the cost of the cable since the price of copper has gone up.

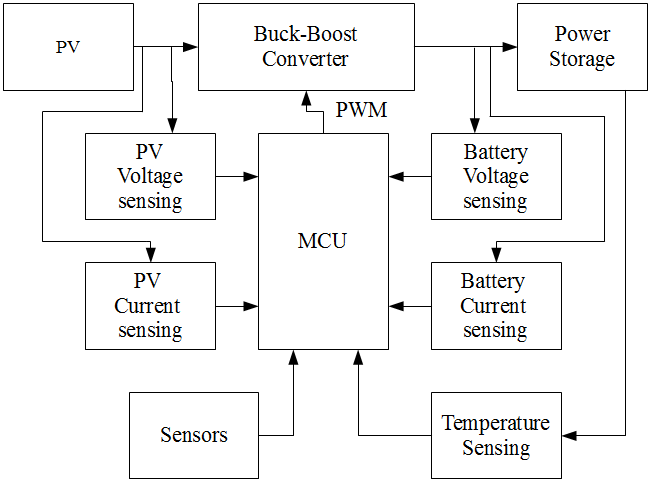
*The power output of the solar panel:* This factor will affect the price and output of the panel. Monocrystalline panels are the most efficient. Polycrystalline panels are less efficient than monocrystalline panels. Amorphous solar panels are the least expensive and less efficient than the other two types. This means that more area and hence size for the same output power.

*Temperature:* The solar panels need sunlight to operate but heat reduces the efficiency of solar panels. The panels will therefore have to be cooled if they will be in the presence of heat for too long.

*Position*: This is affects the solar panel in that the direction its facing will affect its illumination which will affect its output power. The position is also important when insolation(amount of sunlight energy delivered to a square meter in a day) is factored.

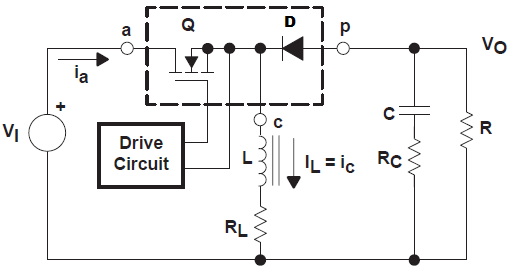
***3.1.4.4 Power Management***

For the power requirements of the sensor node, an MPPT charge controller will be designed. The hardware requirements will compose of a buck-boost DC-DC converter, voltage and current measuring circuits on the PV and battery side of the controller. The buck-boost converter will be used because it is smaller, lighter, more efficient and is the required method for use with Maximum Power Point Tracking (MPPT) control. The MPPT will be performed with the MCU that will control the converter using PWM. Before the actual implementation of the charge controller, a generic diagram of the charge controller is required. The PV voltage sense and current sense will be required for the MPPT algorithm and the battery voltage and current sense will be required for the battery charging cycle. The figure is depicted in Fig. 3.1.4.4.1 below.



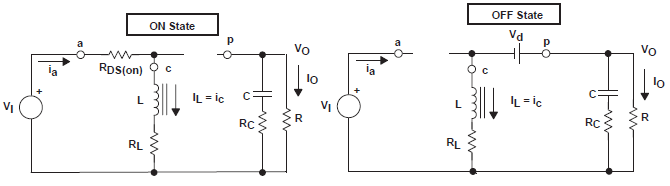
*Fig. 3.1.4.4.1: Block diagram of the PV battery charger(permission requested).*

The design approach is accomplished by first starting with the buck boost portion of the controller. The buck-boost DC-DC converter is composed of an NMOS Q and an output diode D. An inductor L and a capacitor C. The inductor series resistance RL and the capacitor equivalent series resistance RC have been included. The design is shown in Fig. 3.1.4.4.2 below.

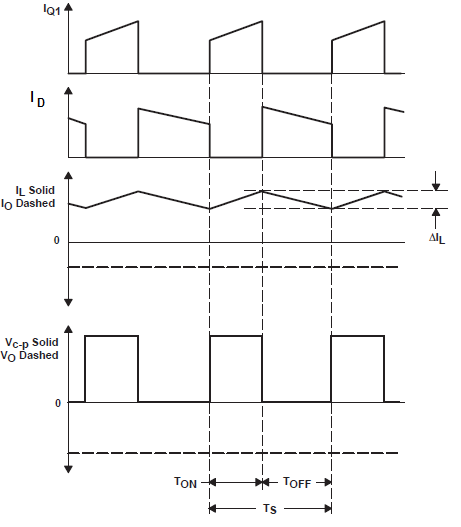


*Fig. 3.1.4.4.2 Buck-Boost converter stage of the charge controller(permission requested).*

The BBC might have to operate in one of two modes depending on the power requirements of the system. The first mode is the continuous conduction mode (CCM) where the current in the inductor flows without going to a zero value during the switching cycle. The second is the discontinuous conduction mode (DCM) where the current in the inductor goes to zero for part of the switching cycle. The two modes will be discussed in greater detail. Since the power stage frequency response of the inductor changes significantly with a change between the two modes, it is desired that the BBC stays in one mode over its expected operating conditions. The BBC utilizes a MOSFET is used with a positive voltage VGS(ON) applied from the gate to the source terminal. An NMOS or PMOS could be used however, using an NMOS has the advantage of having a lower RDS(on). The description of the BBC in CCM will be given to show the derivation of the voltage conversion relationship. The importance of this analysis is that it shows how the output voltage depends on the duty cycle and input voltage and also how the duty cycle can be calculated based on the input and output voltage. This analysis will be performed in steady-state. This implies that the input voltage, output voltage, output load current are fixed. In CCM, the BBC has two states. In the off cycle, Q is on and D is off and in the on cycle Q is off and D is on. The diagrams for these two states are shown in Fig. 3.1.4.4.3 below. Two linear circuits are used to represent the two states. The ON state duration is given by D×TS = TON. D is the duty cycle and TS is the switching cycle. D is a ratio of TS. The duration of the OFF cycle is TOFF. There are only two state in CCM so TOFF equals (1-D) × TS . The waveforms of these two states are shown in the figure below.

**

*Fig. 3.1.4.4.3: The ON and OFF states of the BBC(permission requested).*

**

*Fig. 3.1.4.4.4 CMM Power stage waveforms(permission requested)*

In the ON state, the low resistance of Q from drain to source is RDS(on). The small voltage drop is VDS= IL× RDS(on). Also, the voltage drop across the inductor is then IL× RL. Therefore the voltage applied across the inductor is VI minus (VDS + IL× RL). During this time D is OFF since its reversed biased. During the ON state, the inductor current increases due to the applied voltage. Also, since the input voltage is constant, the inductor current increases linearly. The amount by which the inductor current increases can be calculated as shown below.



The quantity ΔL is referred to as the inductor ripple current. During this time, all the output load current is provided by the capacitor C. During the OFF state, the drain to source impedance of Q is high. The current IL cannot change instantaneously and hence switches from Q to D. As the current IL decreases, the voltage VL reverses polarity, the diode D becomes forward biased and turns ON. The voltage across L is constant and equal to (VO –Vd - IL× RL) where Vd is the forward voltage drop of D. The current IL now flows through the capacitor C, load resistor R and diode D combination. The direction of current IL and the orientation of D means that the current through C and R results in the output voltage VO being opposite in polarity to VI. Since the inductor voltage during TOFF is opposite in polarity to VI (during the TON time), hence the current IL decreases linearly. This current is given by



IL(-) is also an the inductor ripple current. During steady state conditions, IL(+) and IL(-) must be equal otherwise there is a net increase or decrease from cycle to cycle which is not a steady state condition. For equal IL(+) and IL(-), the two currents can be equated and solved for VO. The result CCM buck-boost voltage relationship as shown below.



Setting the two values of ΔIL equal to each other is equivalent to balancing the volt-seconds on the inductor. This is the product of the voltage applied and the time it was applied. This is good way to compute unknown values such as VO and D in terms of known circuit parameters.

The volt-second balance on the inductor should be well understood as much as ohms law. In the equations for ΔIL(+) and ΔIL (-), output voltage was assumed to be constant and has no AC ripple voltage during TON and TOFF. This is a simplification that is common and it involves two separate effects. The first is that the output capacitor is assumed to be large enough that its output voltage is negligible. The second is that the voltage that is across the capacitance ESR is also assumed to be negligible. These are valid assumptions because the ripple voltage of the AC is designed to be much less than the DC component of the output voltage. In the relationship above, it can be observed that VO can be adjusted by adjusting the duty cycle D. VO approaches zero as D approaches zero and increases without bound as D approaches 1. We can simplify the above equation by assuming that VDS, Vd and RL are small enough to ignore and hence set them to zero. This gives the simplified form below which shows the circuit operation.

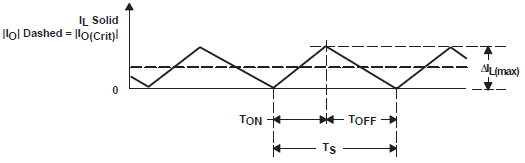


A qualitative way to visualize the operation of the circuit is to view the inductor as an energy storage element. During TON, energy is delivered to the inductor. During TOFF, energy is delivered to the output capacitor and load. The output voltage can then be controlled by controlling the on time of Q. If the time Q is on is increased, more energy is delivered to the inductor and when Q is off more energy is delivered from the inductor to the load and hence the output voltage increases. Unlike in the buck power stage, the average of the inductor current does not equal the output current. The inductor delivers current to the load only during TOFF. The average current during a complete switching cycle is then equal to the output current. To relate the inductor current to the output current it is known that the average current in the output capacitor is equal to zero. For the CCM for the buck-boost power stage, the relationship between the average inductor current and output current is given by

or

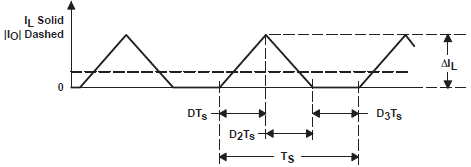
It is also important to note the following, that the average inductor current is proportional to the output current, inductor ripple current is independent of the output load current and the minimum and maximum inductor current values exactly track the output current. For example, if the average value of the inductor current decreases by 1A due to a decrease in the load current then the values of the minimum and maximum of the inductor current decrease by 1A. The derivations above were performed for the buck-boost power stage in CCM.

We now turn our attention to making similar derivations for the buck-boost steady-state (s-s) in discontinuous conduction mode (DCM). We have seen that in CCM, the average of the inductor current tracks the output current and also the maximum and minimum of the inductor current exactly follows the average current of the inductor. If the current of the output load goes below a certain critical current level, then the output of the inductor current will be zero for part of the switching cycle. This condition is evident from the waveforms given above since the ripple current does not change as the current of the load changes. If the inductor current tries to go below zero, it won’t be able to since the diode D prevents this. This is known as the DCM. In a power stage operating in discontinuous mode, there are three unique stages during each switching cycle. This is different from the two stages that are associated with the CCM. There is a condition where the inductor current is at the boundary between the CCM and the DCM. This is where the current in the inductor has fallen to zero and the next cycle begins. The IO and IO(crit.) are given in absolute values since IO and IL have opposite polarities. This condition is shown in Fig: 3.1.4.4.5 below.

**

*Fig.3.1.4.4.5: Boundary between CCM and DCM (permission requested).*

A continued reduction in the output current of the load puts the power stage in DCM as shown in figure 3.1.4.4.6 below.

**

*Fig. 3.1.4.4.6: Discontinuous Conduction Mode.*

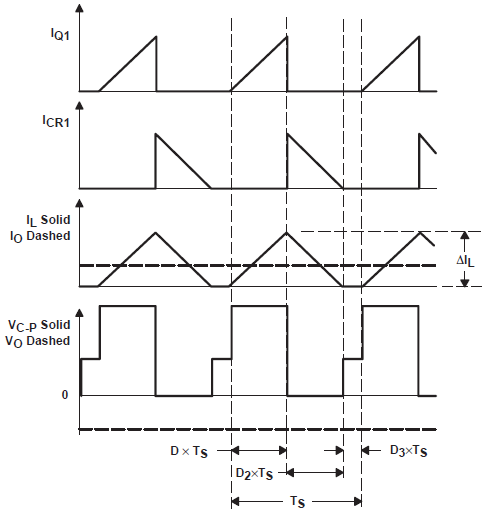
The frequency response of the DCM is different from that of the CCM. Also, the input output relationship differs. The derivation of the DCM is begun by understanding the three unique states that the converter assumes when in DCM. The first is the ON state where Q is on and D is off, the OFF state where Q is OFF and D is on and the IDLE state where both Q and D are off. The first two states are identical to their counterparts in CCM and the circuits are also similar except that TOFF≠(1-D)TS×TS. The third switching cycle is the IDLE state. The following voltages are considered small enough to be negligible: The DC resistance of the output inductor, the forward voltage drop of the output diode, the ON-state voltage drop of the MOSFET. The ON state is defines as TON = D × TS and the OFF state is defined as TOFF = D2×TS. The IDLE state is defined as TS – TON – TOFF = D3 × TS. The waveforms that depict the given states are shown in Fig. 3.1.4.x below. The increase in the inductor current during TON is given by:



The magnitude of the ripple current ΔIL(+) of the inductor is also the peak current IPK since in DCM the current begins from zero for every cycle. During TOFF, the current decrease of the inductor is given by:



We observe that like in CCM, the increase in current, ΔIL(+) during TON and decrease in current, ΔIL(-) during TOFF are equal. The two equations can therefore be equated and solved for VO which results in one of two equations that will be used in the voltage conversion ratio.

**

*Fig. 3.1.4.4.7: Discontinuous Mode Waveforms*



The output current can be computed using VO and load resistance R. This is the average current during one switching cycle for the duration that Q is conducting.



The relationship above is substituted for IPK which results in



The following two questions now exist. The first is for VO and the second is for IO. Both equations are in terms of VI, D and D2. Both equations can be solved for D2 and the two equations set equal to each other. The result is an expression for the DCM buck-boost voltage conversion, VO,which is defined as:

 where K is 

It can be observed from this result the difference between the two conduction modes. For DCM, the voltage conversion relationship depends on VI, duty cycle, inductance of the power stage, switching frequency and load resistance of the output. For CCM, the conversion relationship is only a function of the input voltage and duty cycle. For typical applications, the BBC is design to be operated in one mode. In the next section, the inductance parameters that allow the BBC to operate in CCM are explored.

A BBC design can operate in CCM for load currents above a certain value (5% to 10%) of full load. Usually, the range of the input voltage, the output voltage and the load current are defined by the specification of the converter. The inductor value therefore is the parameter that is left with the duty to maintain continuous conduction mode. There is a minimum value of the inductor to maintain CCM. This value is determined using the following procedure. First, the critical current, (IOCrit), required to maintain CCM is defined. The value being design for is a minimum inductor value. It is therefore more straightforward to perform the derivation by using the inductor current. So, the minimum average value of the inductor current required to maintain CCM is defined as:



The second step is to calculate L to satisfy the relationship above. This can be accomplished by using ΔIL(+) or ΔIL(-). It should be observed that either relationship of ΔIL is independent of the output current level. From the derivations, the worst case conditions that would give the largest Lmin is given by the maximum input voltage since this gives the maximum ΔIL. Substituting and solving for Lmin:



The equation above can be simplified and rewritten as



Using the value of an inductor computed by the above expression will guarantee the BBC to operate in CCM for output load currents that are above the IO(crit). After obtainging the necessary equations of the BBC, the process of selecting each component is discussed. The applied stresses and electrical requirements are given for each component.

*Output Capacitance:* The capacitor stores energy in an electrical field. The function of the capacitor is therefore to attempt to maintain a constant voltage. The value of this capacitor will be selected to minimize the output voltage ripple to a value required by the specification. The output voltage ripple is determined by the series impedance of the capacitor and the output current. The impedance (and hence the output voltage ripple) of the capacitor are a function of three elements. The equivalent series resistance (ESR), the equivalent series inductance (ESL), and the capacitance (C). The output capacitor is selected using the following guidelines. For CCM, the value of capacitance needed is a function of the following parameters: IO, switching frequency fS, and desired output voltage ripple ΔVO. This capacitance value is computed using the equation below. This is assuming that the output voltage ripple is only affected by the capacitance due to the fact that the output capacitor supplies the entire output load current during the TON stage.



Where IO(Max) is the maximum output current and D(Max) is the maximum duty cycle. For DCM operation, the following equation is used and once gain the ripple of the output voltage is assumed to be due to the value of the capacitor



In a lot of practical designs, to achieve the required ESR, a capacitance value that is much greater than the required value is used. For CCM operation and assuming that the capacitance value is such that the ripple due to the capacitance can be ignored, the ESR needed to limit the ripple to ΔVO VP-P is defined as:



For DCM operation and again assuming that the capacitance value is such that the ripple due to the capacitance can be ignored, the ESR needed to limit the ripple to ΔVO VP-P is defined as:



The ripple current flowing through a capacitor results in a power dissipation in the capacitor. This power dissipation causes a rise in the capacitor internal temperature. Excessive temperature can shorten the life of a capacitor. The ripple current ratings of capacitors are dependent on ambient temperature and should not be exceeded. The output capacitor ripple current is the current of the inductor IL, less the output current IO. Therefore the RMS value of the ripple current flowing in the output capacitance in CCM is given by:



ESL is problem in low megahertz region by causing ringing. It can be controlled by choosing low ESL capacitors, by limiting the length of the leads and by replacing one large device with several smaller devices in parallel. The three capacitor technologies that are suitable for low-cost commercial applications are: low-impedance aluminum, organic semiconductor and solid tantalum. The lowest cost that offers high capacitance in small packages is low-impedance aluminum. However, its ESR is higher than the other two. For power supplies, organic semiconductor have become popular in recent years. They offer high capacitance in a small package and a low ESR that is stable over the temperature range. If a surface mount device (SMD) is an absolute must then solid-tantalum capacitors are the best choice. They offer a relatively stable low ESR over the temperature range, high capacitance to volume ratio, high surge-current testing, low ESL and high ripple-current capability.

*Output Inductance:* In the BBC, The function of the inductor is to store energy. Since this energy is stored in their magnetic field due to current flowing, they serve the purpose of attempting to maintain a constant current or limiting the rate of change of current. The output inductance value is usually selected to limit the peak-to-peak current that flows in it. In this way, the mode of operation, CCM or DCM is determined. Inductor ripple current is proportional to the applied voltage and the instant the voltage is applied and is inversely proportional to its inductance. The following discussion talks about the process of selecting an inductor for a design are discussed below. Apart from inductance, the other factors to be considered in inductor selection are the maximum value of the dc or peak current and maximum operating frequency. The inductor should be used within its dc current rating so that it does not overheat and saturate. If the inductor is operated at less than its maximum frequency rating, the maximum core loss will not be exceeded and hence it will not overheat or saturate. Inductors are available in many types. The common core material ones are: Powdered iron and ferrites. Bobbin or rod-core inductors are common and inexpensive but cautioned must be observed when using them since they are more prone to causing noise than other constructions. Current that flows through an inductor results in power dissipation due to the dc resistance of the inductor. This value can be easily computed. Also, the flux swing by the ac voltage results in power dissipation in the inductors core. This value is rarely provided by the manufacturer. On occasion, the maximum operating frequency of the inductor and/or the volt-seconds ratings that is applied will guide the designer with regard to the core loss. The power dissipation causes the inductor temperature to rise. Excessive temperature can degrade the insulation of the winding and increase the core loss. Therefore, all inductor maximum ratings should not be exceeded. The loss in the inductor is defined as:



Where, RCu is the winding

*Power Transistor:* The power transistor has the function of controlling the flow of power from the input power source to the output power storage. In the BBC, transistor Q connects the input power to the inductor during TON and disconnects it during TOFF. The transistor must allow the current to flow to the inductor during TON and block the flow of current during TOFF. Also the transistor must transition quickly from one state to the next to avoid excessive power dissipation during transitions. The transistor utilized for this BBC is a power MOSFET. The MOSFET is the best option in terms of cost and performance considering drive circuits. The n-channel and p-channel are the two types of MOSFETS that can be used for the BBC. PMOSFET is the popular type of transistor for a BBC because the gate drive is simpler than the gate drive for the NMOSFET. The power dissipated by Q is given by:



Where, tr and tf are the MOSFET turn-on and turn-off switching times respectively. QGate is the MOSFET gate-to-source capacitance. Other than selecting the NMOSFET or the PMOSFET, other factors that need to be taken into consideration when selecting the appropriate MOSFET are the maximum drain current ID(Max) and the maximum drain-to-source breakdown voltage, V(BR)DSS. The MOSFET should have a rating of V(BR)DSS greater than the maximum difference between the input and output voltage with some margin added for transients and spikes. Also, the MOSFET should have a rating of ID(Max) that is at least two times the maximum inductor current. Because the margin is insufficient, the junction temperature is also considered and care is taken that it is not exceeded. This junction temperature is estimated as by the equation below.



Where, TA is the ambient/heat-sink temperature, RΘJA is the thermal resistance from the MOSFET chip to the ambient air or heat-sink.

*Output Diode:* The output diode conducts in the TON duration and provides an path for the current. The important criteria for selecting this diode to minimize power dissipation are: appropriate packaging, low-forward voltage drop, current rating, breakdown voltage, and fast switching. If the application does not require the complexity and extra expense of a synchronous diode, the best approach for a low-voltage output is a Schottky diode. The breakdown voltage of the diode must be greater than the maximum voltage of the input and output voltages with extra margin added for spikes and transients. The current rating should be at least twice the maximum output current. Under normal conditions the current rating will be much higher than the output current because limitations due to the power and junction temperature will dominate the selection of the device. The major losses in the diode is primarily a result of the voltage drop across the diode. Power dissipated by the diode can be calculated as product of the forward voltage drop and output load current. Switching losses that occur during the transition from TON to TOFF are negligible relative to conduction losses and as such usually ignored. Power dissipated by the output diode is defined as:



Where, VD is the forward voltage drop of the diode.

The junction temperature can be approximated as shown below

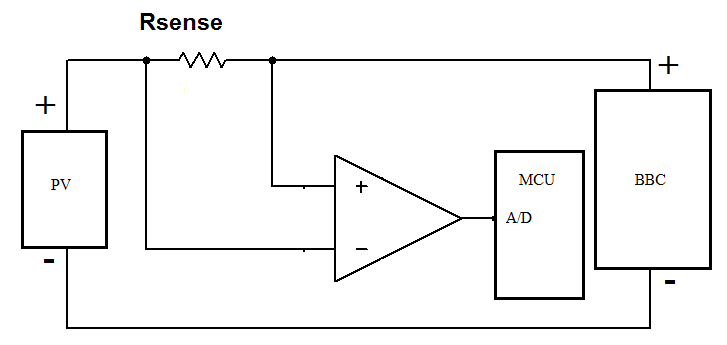


*Voltage and Current Measurement:* In order for the MPPT controller to measure the voltage provided by the PV, two resistors, R1 and R2, are employed in parallel with the PV to act as a voltage divider. The voltage across R2 in the voltage divider is fed into an analog-to-digital converter (ADC) driver circuit (op-amp in a voltage follower configuration that feeds into a low-pass filter) before being delivered to the ADCINA0 channel of the MCU. By choosing the values of R1 and R2 appropriately, the maximum amount of current diverted from the load, I2, is small enough, even in a worst-case scenario, to be considered negligible. The allowable voltage range for each ADC channel of the MCU should be observed. Therefore, the voltage across R2 (which serves as a scaled-down representation of the solar panel's voltage) should not exceed the allowable voltage of the ADC channel. Based on the chosen value of R2 , the maximum voltage, V(R2,max), sent to the ADC driver circuit (and thus ADC channel ADCINA0) can be calculated. The Fig below shows the configuration.

**

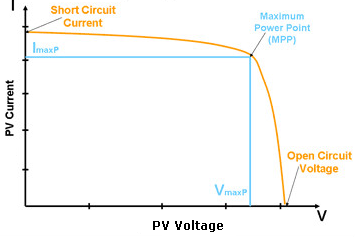
*Fig.3.1.4.4.8: Voltage Sensing Configuration.*

In order for the MCU to measure the current provided by the solar panel, a single resistor (Rsense) is placed in series between the PV and the BBC. The voltage across Rsense is fed into an Op-Amp current sensor whose output voltage is then fed into the ADC driver circuit (op-amp in a voltage follower configuration that feeds into a low-pass filter) before being delivered to the ADCINA1 channel of the MCU. By choosing the value of Rsense appropriately, the maximum voltage drop across Rsense, VRsense, is small enough, even in a worst-case scenario, to be considered to be negligible



*Fig.3.1.4.4.9 Current Sensing Configuration.*

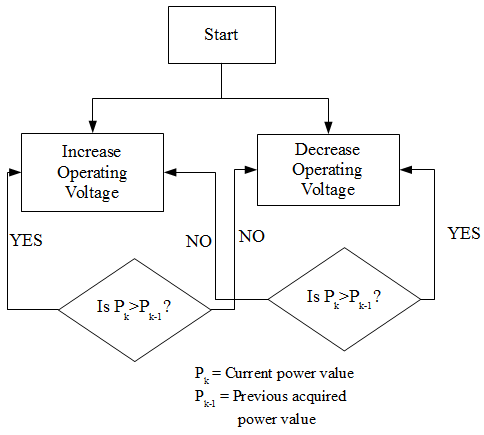
*Maximum Power Point Tracking:* Now that we have covered the hardware for the charge controller, let us cover the software. First, the amount of power that a photovoltaic system generates is a function of solar irradiance(the solar energy per unit area of the surface of the solar panel), ambient temperature, cloud cover among other things. There is a certain value of current and voltage at which the photovoltaic(PV) cell provides the maximum power. These values are not known in advance. The point at which the PV outputs the maximum power is known as the maximum power point(MPP). This is shown in the I-V characteristic of a PV below.

**

*Fig. 3.1.4.4.10: Maximum Power Point of a PV*

To efficiently deliver maximum power from the PV to the power storage, it is necessary to track the MPP. This process is achieved by utilizing algorithms in a process known as maximum power point tracking (MPPT). To choose the right algorithm, the following factors are considered. Convergence speed; this is the time taken by the CC to arrive at the MPP. This time should be small. Cost; the type of algorithm used will affect the microcontroller and number of sensors used. This in turn affects the cost. Ability to detect local Maxima; the irradiance level varies from point to point on the PV’s surface. The result is a local maxima. An efficient algorithm has to be able to detect the MPPT since at local maxima maximum power cannot bed extracted. Sensors utilized: The number and type of sensors depends on the MPPT algorithm. For a more precise MPPT, more sensors will be required. The number of sensors in turn will affect the cost of the CC. The two most common methods of implementing MPPT are discussed below.

*Pertub and Observe:* In this method, the process involves modifying the operating voltage or current of the PV until maximum power is obtained. As an example, the PV operating voltage can be increased which results in an increase in the power. The operating voltage is increased further until the power begins to decrease. At this point the voltage is decreased to get back to the maximum power. The output power value oscillates in the neighbor hood of the MPP until its stable. This method is common due to its implementation. The process is commonly preferred because it’s simple to implement as depicted in the Fig. below. A major drawback of this method is that the obtained power oscillates about the MPP in s-s conditions. Also, under rapidly varying irradiance levels, the algorithm can track in the wrong direction.

**

*Fig. 3.1.4.4-11 Perturb and Observe Flowchart*

Incremental Conductance: This algorithm takes advantage of the fact that the slope of the power-voltage curve is greater than zero at the left of the MPP, less than zero at the right of the MPP and equal to zero at the MPP. These manipulations are shown below



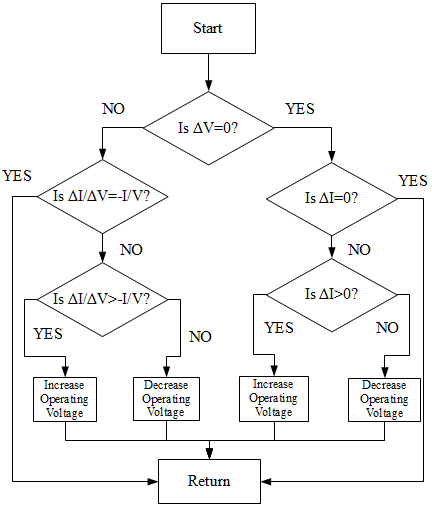
Since



The equation above can be written as



The MPP can then be tracked by comparing the instantaneous conductance (I/V) to the incremental conductance (ΔI/ΔV) as shown in the flow chart below.

**

*Fig. 3.1.4.4.12: Flowchart of Incremental Conductance*

**3.2 SUB GATE**

The sub gate is going to contain many of the same components as the sensor node, there will only be one modification made. The sensor will be eliminated since the sub gate serves only as a data transceiver and not as a vehicle detection device. When researching for the sub gate our group considered the higher power consumption and discussed modifying the power management system (which includes the battery, energy harvesting source, and the power management IC). Our group researched two options for the new power management system. The first option was to replace the power management circuit with a wall wart circuit. The second was to upgrade the components for the sensor node power management system to meet the higher power requirements for the sub gate. In the following sections we will discuss the pros and cons of each option and reveal which one was ultimately selected as the power supply option for the sub gate.

**3.2.1 Wall Wart**

The first substitute power supply our group considered for the sub gate was the wall wart. A wall wart is an external power supply that plugs into a wall outlet. The wall wart our design requires would need to be an AC to DC converter and the power needs to be regulated to 3.3V. We can purchase a wall wart within the specifications that our circuit needs for approximately $5.30 at Digikey.

There are two big reasons why this component was considered. First is the price. By eliminating the power management system, which includes the battery, the power management IC, and the solar power energy harvester, we would be eliminating $16.00 from the price of the sub gate. The total price of the sub gate would be around $15. The sub gate would cost about $25 if the power management system remains in the circuit. This is a significant savings which makes this choice very appealing.

The second reason why we considered this component is that it is a permanent solution to the power requirements for the sub gate. A solar power energy source combined with a micro energy battery cell can definitely supply the power requirements for the sub gate device, but no matter how well the battery is built, it will eventually die. The wall wart solution can provide power to the sub gate circuit indefinitely.

Although the wall wart provides significant savings and a perpetual power supply, our team ultimately decided to eliminate it as a viable option as the power supply for the sub gate. The deciding factor was its hardwired design. Although the benefits caused us to seriously consider this component, we couldn’t sacrifice our wireless design for the convenience of a wired power supply. Therefore our team made the decision to eliminate the wall wart as a viable option for the power supply of the sub gate.

**3.2.2 Alternative Power Management Components**

The power management system for the sensor node is designed to handle the energy requirements for five or less transmission events per hour. The sub gate will have up to 128 children sensor nodes, which means that it needs to handle the energy requirements for potentially 640 transmission events. Each transmission event draws 19mA while receiving the data packet from the sensor node and 23mA while transmitting the data packet to the gateway. This power consumption becomes manageable when considering that the average transmission time will be approximately 400µs. With the energy requirement for the sub gate being 128 times greater than that of the sensor node, our group initially believed that modifications needed to be made to compensate for the additional power. We proposed making two changes to the power management system. Upgrade the energy harvesting source and battery.

The solar panel used for the sensor node outputs typically 18µA and 3V (read sections 4.1.2.1.1 for further details). These values do not take advantage of the quick charge time potential of the battery. The battery can be recharged to 90% charge state in 15 minutes if it receives a constant 4.1V at 25mA. We can purchase a solar panel to meet those specs for about $11.95 at Flex Solar Cells. This should keep the battery charged enough to supply the power needs for the data transmission of the 128 children sensor nodes.

The battery was also considered for an upgrade to a higher capacity value. Since there will be a much higher volume of sensor nodes, the battery capacity was scaled down to maximize the savings per node. Since there will be no more than three nodes per floor, we can afford a higher capacity battery which will be more than sufficient to supply the power needs of the sub gate.

**3.2.3 Sub Gate Power Supply Decision**

After considering two different replacement options for the power supply for the sub gate, our team took a closer look at the power consumption numbers. At 640 transmission events per hour, with each event lasting 400µs and drawing 23mA, the power necessary to support the sub gate’s transmission is 5.88mAs. This equates to 1.64µAh. The MEC201-7S has a capacity of .7mAh. This means that the battery can supply approximately 427 transmission hours on one charge, or about 17 days. We can half that time to take into consideration the receiving transmission energy and we still have 8.5 days of transmission support from one charge of battery. Since the power management system has a solar power that harvest energy to charge the battery, the battery should last at least 20 years. When taking this into consideration, our group decided that the solar panel and battery combination used on the sensor node circuits were more than sufficient for the sub gate circuit.

**3.3 GATEWAY**

The gateway is going to be the direct link between the wireless sensor network and the server. In a real life application, the gateway will require additional hardware to connect to a local area network where the data can then be transferred to the server via the internet. To support the additional hardware the gateway would be hardwired to the power grid of the garage. For our project we will be bypassing the local area network connection by connecting the gateway to the server via a wireless network adapter with a USB connection.

**3.4 SERVER**

All of the data extracted from the wireless sensor network will be stored in a database on the server. The database will be sorted and organized in such a way to make use of that data for a user interface application as simple as possible. To ensure that the server gathers, processes, organizes, and provides access to the data effectively, our team carefully researched the best hardware and software combinations that would produce the results our design requires. In the following sub sections, we will discuss the hardware and software considerations that were made for the server.

**3.4.1 Hardware Requirements**

Two main options will be considered as operating systems for the server, Windows and Linux. The server will serve to both communicate with the wireless sensor network through a RS232 or USB interface and the web through http. In order to meet the hardware requirements needed for the scope of this project the equipment must be capable of running one of the two platforms mentioned above with the available interfaces. Below are outlines of the pros and cons of each operating system and the respective hardware requirements.

***3.4.1.1 Linux***

There are several open source Linux distributions available at no cost, this is one of the main reasons a Linux solution is attractive. There are large user based support communities for all the major distributions with the most popular having commercial support as well. This leads to active development of the Linux platform however sometimes leads to somewhat unorthodox upgrade paths due to the wide array of existing distributions and lack of official support and management. However stable Linux distributions are available and are in use especially in server environments, with a wide array of support options. The main distribution being considered for this is Ubuntu, mainly because of it being free and having the largest support community resulting in a wide array of available software and tools. Ubuntu Server 12.04 is capable of running all of the software requirements other then Microsoft SQL server; there is a driver available that allows Linux to communicate with a remote MS SQL database but this would require additional hardware and effort. Below are the hardware requirements for Ubuntu Server 12.04.

|  |  |
| --- | --- |
| Ubuntu Current Minimum Requirements | |
| Processor(x86) with i686 instruction set | 300 MHz |
| Memory RAM | 128MB |
| Hard Drive | 1GB Free Space |
| Cost | Free (opensource) |

*Figure 3.4.1.1 Current Minimum hardware requirements for Ubuntu Server 12.04*

***3.4.1.2 Windows***

The availability of software development tools, its compatibility with a wide array of hardware and software make Windows an ideal choice. There are two versions of Windows that are able to run the required software for the scope of our projects, one other OS that was considered is XP but due to the fact that as of April 8, 2014 it will no longer supported ruled it out. Windows Server 2008(R2) is able to and in many respects better fit for the requirements of our project however after researching its cost1 it will not be considered. This leaves Windows 7 as the OS to act as the server if the Windows platform is chosen, it is more then capable of running all of the software requirements but a drawback is its cost compared to Ubuntu Server and its higher hardware requirements. There are several versions of Windows 7 available however due to the nature of the use of the server Windows 7 Professional is the best choice mainly due to its advanced backup features and a few other added features. Below are the current minimum hardware requirements for Windows 7 Professional.

|  |  |
| --- | --- |
| Windows 7 Current Minimum Requirements | |
| Processor(x86) | 1GHz |
| Memory RAM | 1024MB(1GB) |
| Hard Drive | 16GB Free Space |
| Cost (Professional edition) | $199($0 our cost)1 |

*Figure 3.4.1.2 Current Minimum hardware requirements for Windows 7*

After researching the above two operating systems and given the scope, budget and requirements of this project Ubuntu server is a better option. This would eliminate MS SQL as being an option for the database however this is not an issue an there are other options that are just as prominent in the database software.

***3.4.2 Software Requirements***

The software requirements for the server are as follows, data packets will need to be accepted via a serial connection. The data packets will then be processed and inserted in a SQL database. A daemon will need to be created for this purpose, it will need to accept the data parse it and insert it into the database. The daemon will also need to identify when the data is from a counter node, it will then first fetch the respective data from the database then increment/decrement the value as needed then insert the new value into the database. Adding a unique id to identify node types, which can be part of the individual node IDs will distinguish counter data. Once the data is in the database a server-side scripting language like ASP, PHP, JavaScript, etc. or a combination of languages can then process it and create the necessary html mark up to display to the end user. The algorithms for determining vacancy percentages and parking probability will be written in this scripting language so it has to be flexible enough to support such logic. The following sections cover different software languages and suites being considered for use as components of the server.

***3.4.2.1 Daemon***

As explained the daemon will accept the data packets via a serial connection between the server and a gateway, the gateway will serve as a bridge to the wireless network. The data packets will contain a series of binary numbers used to identify each nodes’ ID, location and status. The daemon will need to parse the data and convert it to the required format for insertion into the database. The considered software languages are in the following sections.

***3.4.2.1.1 Java***

Java being an object-oriented language would be a good fit for our application, it is cross platform and available in Linux. The Java communications API (javax.comm) is an available extension that facilitates developing platform independent communication applications. It provides access to RS-232 hardware which is a probably solution for our given application. Other extensions like jUSB and JSR-80 also offer options for connecting via USB, which is an attractive solution due to the lost cost and availability of USB devices. Even with the available extensions programming serial connections to hardware can be difficult due to Java cross platform nature and design. A drawback to Java given our application is its automatic enforcement of garbage collection, which can make writing a real time solution like ours difficult. In contrast C/C++ permits garbage collection but does not require it, memory management can be dealt based on application. In order to connect to a MySQL database Java has an API call the MySQL Connector/J. Most major IDEs support Java development, i.e. Netbeans, Eclipse etc. Given Javas memory management and our given real time application it is not an ideal solution for us and will not be used for the project.

|  |  |  |
| --- | --- | --- |
| **Java Minimum System Requirements & Compatibility** | | |
|  | Windows | Linux |
| Processor | 1GHz | 1GHz |
| Disk space | 98MB | 58MB |
| Memory RAM | 128mb(avail.) | 64mb(avail.) |
| Browsers | Chrome, IE7-9, Firefox 3.6+, Firefox 4 | Chrome, Firefox 3.6+ |
| Cost | Open Source(free) | Open Source(free) |
| MySQL Compatible & Lib/Ext | Yes/MySQL Connector/J | Yes/MySQL Connector/J |
| RS232 Compatible & Lib/Ext | Yes/javax.comm | Yes/javax.comm |
| USB Compatible & Lib/Ext | Yes/jUSB or JSR-80 | Yes/jUSB |

*Figure 3.4.2.1.1 Current system requirements for Java*

***3.4.2.1.2 Python***

Python has extensions for use for both a serial RS232 and a USB connection, the modules being pySerial and pyUSB respectively. It has the MySQLdb interface available in order to read and write to a MySQL database for web purposes. The syntax and ease of use of Python makes it an attractive programming language for use in our application. It is cross platform and can be used given our software environment for the task.

|  |  |  |
| --- | --- | --- |
| Python Suggested System Requirements | | |
|  | Windows | Linux |
| Processor | 1GHz | 1GHz |
| Disk space | 5GB+(free space) | 5GB+(free space) |
| Memory RAM | 2GB | 512MB |
| Cost | Open Source(free) | Open Source(free) |
| MySQL Compatible & Library/Extension | Yes/MySQLdb | Yes/MySQLdb |
| RS232 Compatible & Library/Extension | Yes/pySerial | Yes/pySerial |
| USB Compatible & Library/Extension | Yes/pyUSB | Yes/pyUSB |

*Figure 3.4.2.1.2 Current system requirements for Java*

***3.4.2.1.3 C++***

As far as benchmark testing C++ scores amongst the highest if not the highest of the languages being considered for this project. It is a lower level language than the others however offers more in the sense of performance where it lacks in ease of use. It is ideal to write real time software is written in C/C++ due to its performance. There is an available library that makes USB development easier known as libusb, and a few random libraries exist for serial RS232 as well. MySQL has extensive documentation in developing C++ applications that access databases, it is known as MySQL Connector/C++.

|  |  |  |
| --- | --- | --- |
| **C++ Suggested System Requirements** | | |
|  | Windows | Linux |
| Processor | Program Dependent | Program Dependent |
| Disk space | Program Dependent | Program Dependent |
| Memory RAM | Windows Suggested | Linux Suggested |
| Cost | Open Source(free) | Open Source(free) |
| MySQL Compatible & Library/Extension | Yes/ MySQL Connector/C++ | Yes/ MySQL Connector/C++ |
| RS232 Compatible & Library/Extension | Partial/No mainstream solution | Partial/No mainstream solution |
| USB Compatible & Library/Extension | Yes/libusb | Yes/libusb |

*Figure 3.4.2.1.3 Current system requirements for C++*

***3.4.2.2 Data Organization***

The data from the wireless sensor network will be stored into a SQL database; a web program will then query the database for needed data. The database application has to work on the Linux platform since it was chosen for the project, but for the sake of compatibility this aspect of the SQL database will be researched. However due to the use of the Linux platform MySQL with be used due to it’s open source and cross platform nature. A table structure will be created and designed in later portions of this document outlining the hierarchy of the parking system.

***3.4.2.2.1 MySQL***

Being the world’s most popular open source database software there exist an extensive amount of tools that can be used to connect from several programming languages including all considered for this project. Being open source there is no licensing fee and is able to run on both Windows and Linux. As far as features are concerned MySQL lacks features compared to MS SQL Server and other database software, however it meets all the required functions for the scope of this project.

|  |  |  |
| --- | --- | --- |
| **MySQL System Requirements** | | |
|  | Windows | Linux |
| Disk space | 5GB | 5GB |
| Memory RAM | Windows Suggested | Linux Suggested |
| Cost | Open Source(free) | Open Source(free) |

*Figure 3.4.2.2.1 Current system requirements for MySQL*

***3.4.2.2.2 Microsoft SQL Server***

This feature rich SQL Server is a great fir for this project with the exception of two aspects, those being it is Windows dependent and there’s a cost for the software. The fact that Linux was chosen as the OS of choice eliminates MS SQL for this project, however is was noteworthy and for platform dependent build of this project will strongly be considered.

|  |  |  |
| --- | --- | --- |
| **Microsoft SQL Server System Requirements** | | |
|  | Windows | Linux |
| Disk space | 6GB | N/A |
| Memory RAM | 4GB | N/A |
| Cost | License Dependent | N/A |

*Figure 3.4.2.2.2 Current system requirements for Microsoft SQL Server*

***3.4.2.3 Web***

In order to display the data to the end user via a web browser or a mobile device we will first need to process the data retrieved from the database and create the necessary html to be transferred via http. Several server side languages a re being considered for this project; those being ASP & PHP. These languages will be used in unison with jQuery mobile to provide cross platform compatibility for mobile devices and also a browser version of the site. Another consideration is the development of platform specific applications for Android and iOS, however due to the implementation of this project a bias exist towards developing anything platform specific. Two web servers have been considered Apache and IIS, however due to the choice of using Linux IIS was eliminated but discussed here for noteworthy reasons.

***3.4.2.3.1 ASP***

Resources for developing ASP applications are abundant; use of ASP is abundant however it is close to impossible to find a professional developer who prefers it over the alternatives. Furthermore its performance has been tested to be lower then that of PHP and its unique syntax make it harder to develop in. It will only work in a Microsoft server environment (IIS), eliminating this language for this project due to the choice of going with Linux.

|  |  |  |
| --- | --- | --- |
| ASP Compatibility | | |
|  | Windows | Linux |
| System | Yes | No |
| Database | Microsoft SQL (recommended)/MySQL | N/A |
| Server | IIS/ (Not Apache compatible) | N/A |
| Cost | Need a Windows Server License to run IIS | N/A |

*Figure 3.4.2.3.1 ASP Compatibility*

***3.4.2.3.2 PHP***

This is going to be the language of choice for developing the server side software for creating HTMl markup. Being similar to C/C++ makes it ideal for programming in this language, not mentioning the all but infinite amount of documentation. Connecting to a MySQL database using PHP is simple by establishing connections without the use or any additional libraries or extensions. PHP is cross platform compatible and has an open source licensing; several options exist for developing PHP applications. Due to the well rounded nature of this language it will be chosen for interpreting the SQL data into HTML markup.

|  |  |  |
| --- | --- | --- |
| PHP Compatibility | | |
|  | Windows | Linux |
| System | Yes | Yes |
| Database | Microsoft SQL/MySQL | MySQL |
| Server | IIS/ Apache | Apache |
| Cost | Open Source(free) | Open Source(free) |

*Figure 3.4.2.3.2 PHP Compatibility*

***3.4.2.3.3 jQuery mobile***

This extension to the jQuery language allows for the rapid cross platform development of mobile apps via http. It will serve as the framework for formatting the HTML generated for compatibility mainly with mobile devices (i.e. Smartphone, tablet). It provides default classes for displaying content in mobile devices, allowing for a visually pleasing user interface without all the hassles of developing platform specific applications. Content will be dynamically generated with PHP where tags and classes will be added, jQuery then styles according to the tag & class. Most importantly is the wide array of compatible browsers found below.

*jQuery Mobile 1.1.0 Supported Platforms:*

jQuery Mobile has broad support for the vast majority of all modern desktop, smartphone, tablet, and e-reader platforms.

A-grade – Full enhanced experience with Ajax-based animated page transitions.

* Apple iOS 3.2-5.0 - Tested on the original iPad (4.3 / 5.0), iPad 2 (4.3), original iPhone (3.1), iPhone 3 (3.2), 3GS (4.3), 4 (4.3 / 5.0), and 4S (5.0)
* Android 2.1-2.3 – Tested on the HTC Incredible (2.2), original Droid (2.2), HTC Aria (2.1), Google Nexus S (2.3). Functional on 1.5 & 1.6 but performance may be sluggish, tested on Google G1 (1.5)
* Android 3.1 (Honeycomb)  – Tested on the Samsung Galaxy Tab 10.1 and Motorola XOOM
* Android 4.0 (ICS)  – Tested on a Galaxy Nexus S. Note: transition performance can be poor on *upgraded* devices
* Windows Phone 7-7.5 – Tested on the HTC Surround (7.0) HTC Trophy (7.5), LG-E900 (7.5), Nokia Lumia 800
* Blackberry 6.0 – Tested on the Torch 9800 and Style 9670
* Blackberry 7 – Tested on BlackBerry® Torch 9810
* Blackberry Playbook (1.0-2.0) – Tested on PlayBook
* Palm WebOS (1.4-2.0) – Tested on the Palm Pixi (1.4), Pre (1.4), Pre 2 (2.0)
* Palm WebOS 3.0 – Tested on HP TouchPad
* Firebox Mobile (10 Beta) – Tested on Android 2.3 device
* Chrome for Android (Beta) – Tested on Android 4.0 device
* Skyfire 4.1 - Tested on Android 2.3 device
* Opera Mobile 11.5: Tested on Android 2.3
* Meego 1.2 – Tested on Nokia 950 and N9
* Samsung bada 2.0 – Tested on a Samsung Wave 3, Dolphin browser
* UC Browser – Tested on Android 2.3 device
* Kindle 3 and Fire - Tested on the built-in WebKit browser for each
* Nook Color 1.4.1 – Tested on original Nook Color, not Nook Tablet
* Chrome Desktop 11-17 - Tested on OS X 10.7 and Windows 7
* Safari Desktop 4-5 - Tested on OS X 10.7 and Windows 7
* Firefox Desktop 4-9 – Tested on OS X 10.7 and Windows 7
* Internet Explorer 7-9 – Tested on Windows XP, Vista and 7
* Opera Desktop 10-11 - Tested on OS X 10.7 and Windows 7

B-grade – Enhanced experience except without Ajax navigation features.

* Blackberry 5.0: Tested on the Storm 2 9550, Bold 9770
* Opera Mini (5.0-6.5) - Tested on iOS 3.2/4.3 and Android 2.3
* Nokia Symbian^3 - Tested on Nokia N8 (Symbian^3), C7 (Symbian^3), also works on N97 (Symbian^1)

C-grade – Basic, non-enhanced HTML experience that is still functional

* Blackberry 4.x - Tested on the Curve 8330
* Windows Mobile - Tested on the HTC Leo (WinMo 5.2)
* All older smartphone platforms and featurephones – Any device that doesn’t support media queries will receive the basic, C grade experience

***3.4.2.3.4 Platform Specific SDK***

Software developments kits were considered as an option for building the mobile version of the user interface for mobile devices. Several reasons were responsible for the elimination of this as an option for this project. The fact that this route is the most time consuming and given the limited timeframe of the development cycle of this project, this option was eliminated. Different environments use different languages to program in and each have their own respective SDK, for example Googles’ SDK involves programming in Java where iOS uses C++/Objective C. Though applications can be constructed to tailor fit devices and platforms this route will not be an option for the scope of this project, a more standard cross platform approach will be taken.

***3.4.2.3.5 Microsoft IIS (Internet Information Services)***

This is Microsoft’s proprietary server technology. It is integrated into Microsoft’s Server edition of their OS. In order to run this server environment a Server license must first be acquired. IIS also aims to integrate well with other Microsoft server products such as MS SQL Server and Exchange. These extra features can be attractive if later versions of this project are developed, however for the scope of this project its cost and complexity is undesirable. It is not cross platform compatible and will not operate on Linux. Below is an outline of IISs compatibility.

|  |  |  |
| --- | --- | --- |
| IIS Compatibility | | |
|  | Windows | Linux |
| System | Yes | NO |
| Database | Microsoft SQL/MySQL | N/A |
| Language | ASP/PHP | N/A |
| Cost | Requires Windows Server license | N/A |

*Figure 3.4.2.3.5 IIS Compatibility*

***3.4.2.3.6 Apache Web Server***

This being the most popular and openly developed web server in the world makes it the most desirable server software for this project. Its cross platform nature make it ideal for this project, it will allow the use of developed software to be used on either platform. Furthermore if a Windows environment is desired an apache web server can be setup on any version of windows and a Server version is not required, making the attribute4d cost far less then an IIS environment. It allows for virtual host and integrates with BIND9 for DNS management. Apache will be used for the project and will provide http services for displaying data to the end users. Below is an outline of Apaches compatibility.

|  |  |  |
| --- | --- | --- |
| Apache Compatibility | | |
|  | Windows | Linux |
| System | Yes | Yes |
| Database | Microsoft SQL/MySQL | MySQL |
| Language | ASP/PHP | PHP/ASP(requires port) |
| Cost | Opensource(free) | Opensource(free) |

*Figure 3.4.2.3.6 Apache Compatibility*

**3.5 USER INTERFACE**

Given our application the user interface designed must be intuitive and easy to follow. The routine phases will be used when designing the interface and are outlined below.

*Functionality requirements* – User must be able to choose a Garage/Lot based on easy to see information (i.e. % vacant), then identify a floor/row based on the same info then finally find an individual space. Furthermore the option to share a space with a friend must be intuitive fast and easy, in other words they need to post the space availability and time in a routine effective manner. Design will emphasize on ease of use and transgression, the primary role of this application is to inform.

*User analysis* – We will deal with a wide range of users with different backgrounds, language support may be an option. Users will expect an efficient process in order to make this a plausible solution; it will need to be both visual and textual to accommodate different learning styles. Other considerations are how would the application fit into the users daily parking routine, what devices will be used and what preferences will be chosen.

*Information Architecture* **–** Provide a hierarchy/tree system to outline content in the system, this provides a streamlined view of available information.

*Graphic interface design* - The look and feel of the application, for branding purposes this should be unique as to allow a user to easily identify content provided by the system. Consistent user interfaces have a shorter learning curve, because users will recognize parts of the system and be able to fall back on prior experience.

This part of the design will be subject to change depending on user feedback, once a graphical interface has been designed as a sketch or rough draft testing can be performed in order to grasp the effectiveness of the graphic interface design. When designing a graphic interface coloring is not only important for branding purposes studies have shown that different colors induce different psychological effects on people, it is our aim to induce an environment free of confusion and stress which is vital when operating a vehicle.

*Prototyping and Usability testing* - Once the general idea of how the application is to function and appear is worked out testing is needed to verify theoretical findings.

The ergonomics standard ISO 9241 establishes a framework of “ergonomic” principles for the dialogue techniques with high-level definitions and illustrative applications and examples of the principles. The following seven of the ten principles that apply to this project are outlined below.

* Suitability for the task: the dialogue is suitable for a task when it supports the user in the effective and efficient completion of the task.
* Self-descriptiveness: the dialogue is self-descriptive when each dialogue step is immediately comprehensible through feedback from the system or is explained to the user on request.
* Controllability: the dialogue is controllable when the user is able to initiate and control the direction and pace of the interaction until the point at which the goal has been met.
* Conformity with user expectations: the dialogue conforms with user expectations when it is consistent and corresponds to the user characteristics, such as task knowledge, education, experience, and to commonly accepted conventions.
* Error tolerance: the dialogue is error tolerant if despite evident errors in input, the intended result may be achieved with either no or minimal action by the user.
* Suitability for individualization: the dialogue is capable of individualization when the interface software can be modified to suit the task needs, individual preferences, and skills of the user.
* Suitability for learning: the dialogue is suitable for learning when it supports and guides the user in learning to use the system.

The concept of usability is outlined in part 11 of the ISO 9241 standard and states:

* Usability is measured by the extent to which the intended goals of use of the overall system are achieved (effectiveness).
* The resources that have to be expended to achieve the intended goals (efficiency).
* The extent to which the user finds the overall system acceptable (satisfaction).

Part 12 of the standard outlines the standard for organizing and presenting information. There are seven presentation attributes to the standard listed below.

* Clarity: the information content is conveyed quickly and accurately.
* Discriminability: the displayed information can be distinguished accurately.
* Conciseness: users are not overloaded with extraneous information.
* Consistency: a unique design, conformity with user’s expectation.
* Detectability: the user’s attention is directed towards information required.
* Legibility: information is easy to read.
* Comprehensibility: the meaning is clearly understandable, unambiguous, interpretable, and recognizable.

The user guidance in Part 13 of the ISO 9241 standard describes that the user guidance information should be readily distinguishable from other displayed information and should be specific for the current context of use. User guidance can be given by the following five means:

* Prompts indicating explicitly (specific prompts) or implicitly (generic prompts) that the system is available for input.
* Feedback informing about the user’s input timely, perceptible, and non-intrusive.
* Status information indicating the continuing state of the application, the system’s hardware and software components, and the user’s activities.
* Error management including error prevention, error correction, user support for error management, and error messages.
* On-line help for system-initiated and user initiated requests with specific information for the current context of use.

**4. DESIGN**

After researching which technologies would best suite our design needs, our group selected the specific parts to be used in the final design of our Parkit system. In the following sections we will discuss which parts were selected for the sensor node subsystem, the server sub system, and the user interface subsystem. We will also go into detail on how these parts are integrated together to construct the circuits and programs for each subsystem.

**4.1 DETECTION NODES**

The sensor node is the most complex component, in terms of circuitry design, of the entire Parkit system. More research time was spent on this component, than on every other component combined. This section covers the circuitry (for the occupancy and counter nodes) and software design for the sensor node.

**4.1.1 Magnetometer Sensor**

Detecting the presence of a vehicle is performed by the three magnetic sensors one for the x, y, and z axis measurement. The sensors are multiplexed and connected to a 12-bit ADC. The device is configured and controlled by registers that are on the chip and will be described in the section that follows. The registers are shown in Table 4.1.1-1 below.

|  |  |  |
| --- | --- | --- |
| Address Location | Name | Access |
| 00 | Configuration Register A | Read/Write |
| 01 | Configuration Register B | Read/Write |
| 02 | Mode Register | Read/Write |
| 03 | Data Output X MSB Register | Read |
| 04 | Data Output X LSB Register | Read |
| 05 | Data Output Z MSB Register | Read |
| 06 | Data Output Z LSB Register | Read |
| 07 | Data Output Y MSB Register | Read |
| 08 | Data Output Y LSB Register | Read |
| 09 | Status Register | Read |
| 10 | Identification Register A | Read |
| 11 | Identification Register B | Read |
| 12 | Identification Register C | Read |

*Table 4.1.1-1 Register List of the HMC5883*

There are two configuration registers. The first register, configuration register A is used to set the rate of the data output and the set the configuration of measurement. It has eight bits. Bits five to seven have to be cleared for correct operation. Bits two to four set the rate which the data will be written to all of the three data output registers. Bits zero and one configure the flow of measurement. They specifically indicate whether or not an applied bias to the sensor should be incorporated into the measurement. The selectable rates of the data measurement are shown in table Table 4.1.1-2 below. The second configuration register is used to set the gain of the device. Bit zero through bit seven is used to indicate the location of the bits. Bits five to seven is used to configure the gain of the device. This gain is common for all channels. Bits zero to four have to be cleared for correct operation. The nominal gain settings are shown in table Table 4.1.1-3 below. The next register is the mode register. The primary purpose of the operating modes is power management. This is an 8-bit register which data can be read from or can be written to. It is used to select the mode in which the device will operate. Bits 0 through 7 indicate the location of the bits with bit 7 being the first bit of the data stream. Bits 2 to 7 have to be cleared for correct operation and bits 0 and 1 select the operation mode of the device.

|  |  |  |  |
| --- | --- | --- | --- |
| DO2 | DO1 | DO0 | Typical Data Output Rate (Hz) |
| 0 | 0 | 0 | 0.75 |
| 0 | 0 | 1 | 1.5 |
| 0 | 1 | 0 | 3 |
| 0 | 1 | 1 | 7.5 |
| 1 | 0 | 0 | 15 (default) |
| 1 | 0 | 1 | 30 |
| 1 | 1 | 0 | 75 |
| 1 | 1 | 1 | Not used |

*Table 4.1.-.2 Data output rates*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GN2 | GN1 | GN0 | Sensor Input Field Range: | Gain  (counts/Gauss) | Output Range |
| 0 | 0 | 0 | ± 0.9 Ga | 1280 | 0xF800–0x07FF (-2048–2047) |
| 0 | 0 | 1 | ± 1.2 Ga | 1024 (default) | 0xF800–0x07FF (-2048–2047) |
| 0 | 1 | 0 | ± 1.9 Ga | 768 | 0xF800–0x07FF (-2048–2047) |
| 0 | 1 | 1 | ± 2.5 Ga | 614 | 0xF800–0x07FF (-2048–2047) |
| 1 | 0 | 0 | ± 4.0 Ga | 415 | 0xF800–0x07FF (-2048–2047) |
| 1 | 0 | 1 | ± 4.6 Ga | 361 | 0xF800–0x07FF (-2048–2047) |
| 1 | 1 | 0 | ± 5.5 Ga | 307 | 0xF800–0x07FF (-2048–2047) |
| 1 | 1 | 1 | ± 7.9 Ga | 219 | 0xF800–0x07FF |

*Table 4.1.-3 Nominal Gain Settings*

*Continuous measurement mode:* In this mode, the device performs measurement continuously with the results placed in the data register. In continuous measurement mode, the configuration register settings affect the rate of the data output, the configuration of the measurement and the gain. For conservation of current between measurements, the device can be placed in a state that is similar to but not idle mode. In this state, the mode register bits are unchanged. If necessary, the data can be re-read from the output registers. The master must ensure that the data is accessed before the next measurement is completed or the previous measurement may be lost. While in continuous measurement mode, all the registers maintain their values and the I2C bus is enabled for use by other networked devices.

*Single measurement mode:* This is the mode that the device defaults to when powered up with a single power supply. With a dual power supply this mode is the default when AVDD is high. In this mode, the device makes a single measurement that is placed in the data output registers. In this mode, the configuration register settings affect the measurement configuration and the gain. After measurements are taken, the output register is updated, the device goes into sleep mode and the mode register is changed to reflect the mode of the device. Also, all the registers maintain their values and the I2C bus is enabled for use by other devices on the network.

*Idle mode:* In this mode, the device can be accessed by the I2C bus. However, the components that consume lots of power are disabled. These include, ADC, amplifier, SVDD pin and bias current of the sensor. All the registers maintain their values and the I2C bus is enabled to be used by other devices on the network.

*Sleep mode:* This is the default mode when the device powers up and has dual power supplies and DVDD goes high while AVDD remains low. In this mode the device is only limited to listening to the I2C bus. Also, the internal clock is not running and the values in registers are not maintained. The only function that the device is able to perform is recognize and execute any instructions specific to it and it doesn’t change from sleep mode due to the presence of other traffic on the network. The network is enabled so that other devices may use it. There are two differences between this mode and the idle mode. The first is since the clock is disabled, the device created less noise on the system and second is, because the clock is disabled, the device consumes lower current.

*Off mode:* In this mode the device is off and no functionality exists. The AVDD and DVDD are low, the bus is enabled so that other networked devices can utilize it and the I2C pins are in a high impedance state. A table that shows the mode register is shown in Table 4.1.1-4 below. A table with the designation of the bits of the mode register is shown in Table 4.1.1-5 below and a table that shows the operating modes of the device is shown in Table 4.1.1-6 below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MR7 | MR6 | MR5 | MR4 | MR3 | MR2 | MR1 | MR0 |
| (0) | (0) | (0) | (0) | (0) | (0) | MD1 (0) | MD0 (1) |

*Table 4.1.1-4 Mode Register*

|  |  |  |
| --- | --- | --- |
| Location | Name | Description |
| MR7 to MR2 | 0 | These bits must be cleared for the device to operate correctly |
| MR1 to MR0 | MD1 to MD0 | These bits are used to select the mode |

*Table 4.1.1-5 Mode Register Designation bits*

|  |  |  |
| --- | --- | --- |
| MD1 | MD0 | Mode |
| 0 | 0 | Continuous-Measurement Mode. |
| 0 | 1 | Single-Measurement Mode (default). |
| 1 | 0 | Idle Mode. |
| 1 | 1 | Sleep Mode. |

*Table 4.1.1-6 Operating Modes*

The data output registers are two 8-bit registers, register A and register B for the X channel. The MSB is contained in register A and the LSB is contained in register B. The value in these two registers is a 16-bit value that is in 2-complement form. The values range from 0xF800 to 0x07FF. The default value is 0. If the ADC reading overflows, underflows or the math overflows during the bias measurement, the registers will contain the value -4096 in 2’s complement form that will clear when the next valid measurement is made. The same registers exist for the Y and Z channel. The output register operation is such that when one or more of the register are read, new data cannot be placed in any of the output registers until all six are read. The DRDY and the RDY bit can also not be read until all the data is placed in the output registers. To indicate the status of the device, a status register (SR) is provided. This is an 8-bit read only register with bit 7 denoting the first bit of the data stream. The register bits are designated as follows. Bits 3 to 7 must be cleared for proper operation. Bit 2 is the regulator enabled bit which is set when the internal voltage regulator is enabled and cleared when the regulator is disabled. Bit 1 is set when some but not all of the six data output registers have not been read. In the set condition, new data cannot be written to the registers until one of four conditions have been met. Condition one, all six data output registers have been written or the mode has been changed, condition two, a POR has been issued. Condition three, the mode is changed. Condition four, the measurement is changed. Bit 0 is the ready bit which is set when data has been written to all the six output registers and is cleared when the device has initiated a write to the data output registers when in off mode and one or more of the data output registers has been written to. When the ready bit is cleared, it remains so for 5 μs. As an alternative, the DRDY can be used to monitor the device for measurement data. Table 4.1.1-7 below shows the status register and Table 4.1.1-8 below shows the status register bit designations. Three identification registers are used to identify the device with bit 7 being the first bit of the data stream. These are identification register A which contains ASCII value H, identification register B which contains ASCII value 4 and identification register C which contains ASCII value 3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SR7 | SR6 | SR5 | SR4 | SR3 | SR2 | SR1 | SR0 |
| (0) | (0) | (0) | (0) | (0) | REN (0) | LOCK (0) | RDY(0) |

*Table 4.1.1-7 Status Register*

|  |  |  |
| --- | --- | --- |
| Location | Name | Description |
| SR7 to SR3 | 0 | Bits that have to be cleared for the device to operate properly. |
| SR2 | REN | Regulator Enabled Bit. |
| SR1 | LOCK | Data output register lock. |
| SR0 | RDY | Ready Bit. |

*Table 4.1.1-8 Status Register Bit Designation*

*Component Selection:* After researching about the various ways that a vehicle could be detected, the group decided that it was going to utilize the HMC5883L, a 3-axis digital magnetometer. The features and benefits the component that meets the design requirements are listed below.

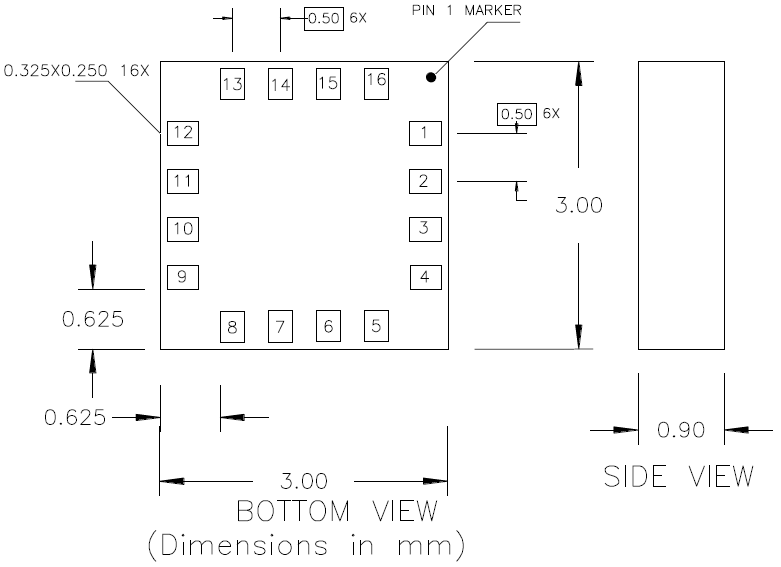
*Self test:* This feature is useful in providing a way to scale the sensitivity of the individual sensing components so that they have similar outputs.

*Internal power management:* The power management feature is useful in managing the power used to power the internal components. This reduces the amount of specific hardware that has to be bought for the magnetometer and therefore reduces the overall project cost.

Power on Reset: The magnetometer should have a default power on reset that sets the default state of the various components like the MUX, the ADC and the various currents and voltages.

*3-Axis Magnetoresistive Sensors*

* 12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 5 milli-Gauss Resolution in ±8 Gauss Fields
* Low Voltage Operations (1.6 to 3.3V)
* Built-In Strap Drive Circuits
* I2C Digital Interface
* Lead Free Package Construction
* Software and Algorithm Support Available
* Fast 116 Hz Maximum Output Rate
* Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors
* Designed for High Volume, Cost Sensitive OEM Designs
* Easy to Assemble & Compatible with High Speed SMT Assembly
* Compatible for Battery Powered Applications
* Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
* Popular Two-Wire Serial Data Interface for Consumer Electronics
* Sensors Can Be Used in Strong Magnetic Field Environments
* The device package is shown below in Fig. 4.1.1.1.1



*Fig. 4.1.1.1.1 Device dimensions*

The table 4.1.1.1.8: below shows the pin configurations

|  |  |  |
| --- | --- | --- |
| Pin | Name | Description |
| 1 | SCL | Serial Clock – I2C Master/Slave Clock |
| 2 | AVDD | Analog Positive Supply |
| 3 | NC | Not to be Connected |
| 4 | NC | No Connection |
| 5 | NC | No Connection |
| 6 | NC | No Connection |
| 7 | NC | No Connection |
| 8 | SETP | Set/Reset Strap Positive – S/R Capacitor (C2) Connection |
| 9 | GND | Supply Ground/Return |
| 10 | C1 | Reservoir Capacitor (C1) Connection |
| 11 | GND | Supply Ground/Return |
| 12 | SETC | S/R Capacitor (C2) Connection |
| 13 | DVDD | Digital Positive Supply |
| 14 | VREN | Voltage Regulator Enable, (GND = Dual Supply, AVDD = Single Supply) |
| 15 | DRDY | Data Ready, Interrupt Pin. Nominally high. Low for 5usec when data is placed in the output registers |
| 16 | SDA | Serial Data – I2C Master/Slave Data |

*Table 4.1.1.1.8: Pin Configurations*

The specifications of the device that will be of importance to the design of the project are listed in the table below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Characteristics | Conditions\* | Min | | Typ | | Max | | Units |
| Power Supply | | | | | | | | |
| Supply Voltage | AVDD Referenced to AGND  DVDD Referenced to DGND | 2.5  1.6 | | 1.8 | | 3.3  2.0 | | V  V |
| Current Draw | Sleep Mode (dual supply)  Idle Mode (dual supply)  Measurement Mode (8Hz averaged)  Dual supply (AVDD = 2.5V, DVDD  = 1.8V) | -  -  - | | 2.5  240  640 | | -  -  - | | μA  μA  μA |
| Sleep Mode (single supply)  Idle Mode (single supply)  Measurement Mode (8Hz averaged)  Single supply (AVDD = 2.5V) | -  -  - | | 110  340  740 | | -  -  - | | μA  μA  μA |
| Performance | | | | | | | | |
| Resolution | AVDD=3.0V, GN2 | |  | 10 |  | | milli-gauss | |
| Linearity | ±2.0 gauss input range | |  |  | 0.1 | | ±% FS | |
| Disturbing Field | Sensitivity starts to degrade. Use S/R pulse to restore sensitivity. | |  |  | 20 | | gauss | |
| Output Rate | Refer to Configuration Register A section | |  | 15 | 116 | | Hz | |
| Measurement Period | From receiving command to data ready | |  | 8.3 |  | | msec | |
| Turn-on Time |  | |  | 200 |  | | μs | |
| I2C Address | 7-bit address  8-bit read address  8-bit write address | |  | 0x1E 0x3D 0x3C |  | | hex hex hex | |
| I2C Rate | Controlled by I2C Master | |  |  | 400 | | kHz | |
| I2C Hysteresis | Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (DVDD=1.8V)  Rise (DVDD=1.8V) | |  | 0.603 1.108 |  | | V  V | |
| Self Test | X & Y Axes  Z Axis | |  | ±0.64 ±0.59 |  | | gauss | |
| General |  | |  |  |  | |  | |
| Operating Temperature | Ambient | | -30 |  | 85 | | °C | |

*Table 4.1.1.1.8: Device Specification*

*Software Design:* The device communicates using the I2C communication protocol. It uses 0X3C for the write operations and 0x3D for the read operations. In single measurement mode, it takes 8.3 milli-seconds to fill the six data register with a valid first measurement. The measurement can be changed to continuous measurement mode by sending the three bytes: 0x3C 0x02 0x00. This writes 00 into the mode register to convert from single measurement to continuous. A wait is required and that depends on the data rate. To clock out the new data, the byte 0x3D is sent and the data in registers 3 through 8 is clocked out. All the six registers must be read properly before new data can be placed in them. The device then automatically re-points to register 3 for the next 0x3D byte. The flow chart in Fig X2.below shows the program flow.



*Fig. 4.1.1.1.2: Program flow*

**4.1.2 Power Management System**

One of the design specifications for our project is that the sensor nodes are energy-harvesting, with the power supply lasting at least twenty years. To achieve this, we designed a power management subsystem that produces and maintains the energy the sensor node requires. This subsystem harvests, charges and stores energy, and powers the entire node system. In the following sections we will discuss the components used to build the subsystem and how they are integrated together in the circuit to meet the design specifications.

***4.1.2.1 Components***

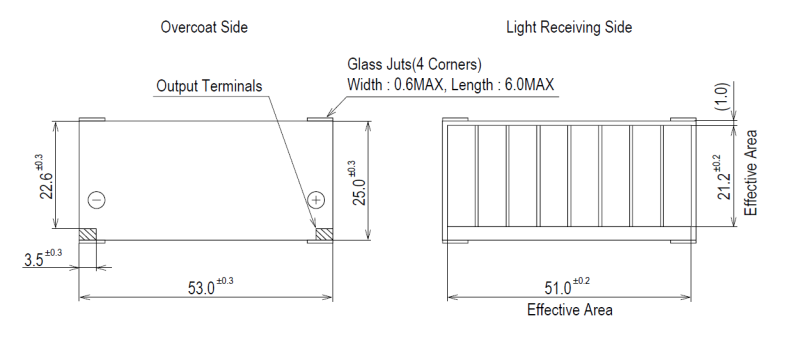
There are three main components that make up the power management system. These three components are the photovoltaic cell which serves as the means for energy harvesting, the battery that stores the energy harvested and an IC that manages the battery charge, protects the circuit from over charge and over discharge, and provides the regulated voltage for the sensor node circuit. In the following sub sections we will discuss the specs for each component and provide a bill of materials for the components used to build the power management system.

***4.1.2.1.1 Photovoltaic Cell***

Our group chose the AM-1801 photovoltaic cell for our energy harvesting medium. It is compact enough (53 x 25 x 1.3 mm) to fit within the housing of our sensor node. It outputs 3V and 18.5µA at 200 lux. During the day the cell will receive enough lux from the daylight to harvest energy to keep the battery adequately charged during usage. At night, the garage lights offer enough lux to continue charging the battery, and since the sensor nodes will be at sleep mode after the garage closes for the night, the power draw will be minimal (nW). Figure 4.1.2.1.1-1 shows the full dimensions of the photovoltaic cell and Table 4.1.2.1.1-1 contains the specs for the AM-1801 photovoltaic cell.

|  |  |
| --- | --- |
| Specs | AM-1801 |
| Full Solar Panel | |
| Open Circuit Voltage @ 200 lux, 25 C | 4.9V |
| Short Circuit Current @ 200 lux, 25 C | 20µA |
| Operating Voltage @ 200 lux, 25 C  @ 50 lux, 25 C | 3V  2.6V |
| Operating Current @ 200 lux, 25 C  @ 50 lux, 25C | 18.5µA  4.6µA |
| Operating Temperature | -5C – 40C |
| Individual Cell | |
| Open Circuit Voltage @ 200 lux, 25 C | 0.63V |
| Short Circuit Current @ 200 lux, 25 C | 17µA/cm^2 |
| Maximum Output @ 200 lux, 25 C | 0.7µW/cm^2 |
| Price | $3.38 |

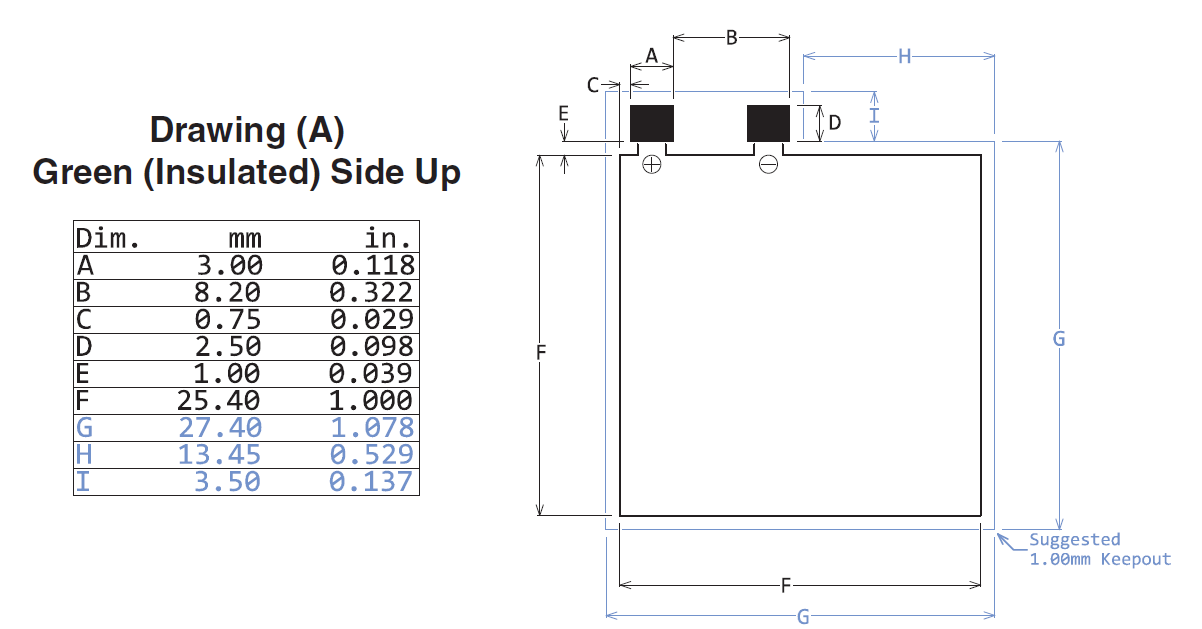
*Table 4.1.2.1.1-1 AM-1801 Specifications: Source[24]*



*Figure 4.1.2.1.1-1 AM-1801 Dimensions: Source[24](permission pending)*

***4.1.2.1.2 Battery***

Our group selected the MEC201-7S battery. It supplies 4.0 - 4.1V for 80-100% charge state and it supplies 3.9V for 0-80% charge state. The voltage decay is 1% per year, after 40 years the voltage will be at 2.46V which is enough to power all of the components of the sensor node. This doubles the minimum life requirement of 20 years for the battery that has been established in the design specifications. The battery has a maximum continuous discharge current of 30mA, this is enough amps to run the entire sensor node circuit, even when the transceiver is transmitting and receiving data packets. The typical discharge current leakage is less than 10nA, which is about 1000 times better than standard ion battery leakage values. The battery has a capacity of 1mAh. Although this value is low, the majority of the current the circuit draws is in micro amps while in active mode and nano amps while in idle mode. The only significant draw is from the transceiver, which draws 19 to 23 amps. To extend the battery charge cycle, the sampling rate will be lowered to 1 or 2 samples per minute. Figure 4.1.2.1.2-1 shows the dimensions of the battery cell and Table 4.1.2.1.2-1 contains the specs for the MEC201-10S.



*Figure 4.1.2.1.2-1 MEC201 Package Dimensions: Source[22](permission granted)*

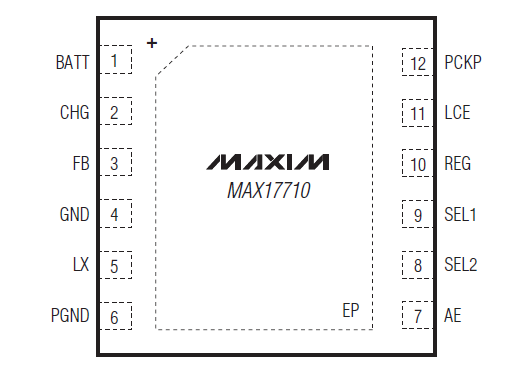
|  |  |  |
| --- | --- | --- |
| Specs | MEC201 | Conditions |
| Capacity | .7mAh | .35 mA Discharge Rate @25C |
| Operating Temperature | -40C-85C |  |
| Charge Time (90%) | 20 min | 4.1V/min peak current 10mA |
| Max Discharge Current | 30mA | 25C |
| Annual Capacity Loss | 1%,3%,6% | 25C,45C, and 65C respectively |
| Annual Self-Discharge Rate | 1%,3%,6% | 25C,45C, and 65C respectively |
| Price | $10.44 |  |

*Table 4.1.2.1.2-1 MEC201 Specifications: Source[22]*

***4.1.2.1.3 Power Management IC (PMIC)***

Our group selected the MAX17710 as the IC for the power management system. Its dimensions are 3mm x 3mm x 0.5mm. This particular PMIC performs two functions. One, it charges a low capacity cell battery with overcharge protection and two, it is a low dropout regulator output with over discharge protection. The PMIC can manage an energy harvest source that outputs as little as 1µW. This means that even if there are poor lighting conditions, the solar panel will produce enough power to trickle charge the battery through the PMIC. The PMIC also has a shunt regulator. If the voltage charge exceeds 5.7V the shunt regulator will redirect the excess power to be dissipated.

The PMIC provides three regulated output voltages, which are, 1.8V, 2.4V, and 3.3V. If there is not enough charge in the battery to supply the load, the PMIC will shut off the regulated voltage output until the harvested energy source charges the battery to the 2.3V threshold. The PMIC also supports energy buffering. On pin 12, external capacitors can be connected to store energy to be used in current pulsing. Figure 4.1.2.1.3-1 shows the pin configuration for the PMIC and Table 4.1.2.1.3-1 contains the specs for the MAX17710.



*Figure4.1.2.1.3-1 MAX17710 Pin Configuration: Source[23](permission granted)*

|  |  |
| --- | --- |
| Specs | MAX17710 |
| Charge Input Maximum Voltage | 5.7V |
| Charge Input Maximum Current | 100mA |
| Charge to Battery Dropout Voltage  (Vcharge = 4V, Ibattery = -40mA) | 100mV |
| Charge Quiescent Current | 625nA |
| Battery Regulator Volatage | 4.125V |
| Regulated Output Voltage | 1.8V  2.3V  3.3V |
| Battery Under Voltage Threshold | 2.3V |
| Regulator startup time | 5.3ms |
| Charge Input Shunt Limit | 50mA |
| Charge Shunt Delay | 5µs |

*Table 4.1.2.3-1 MAX17710 Specifications: Source[23]*

***4.1.2.1.4 Bill of Materials for Power Management System***

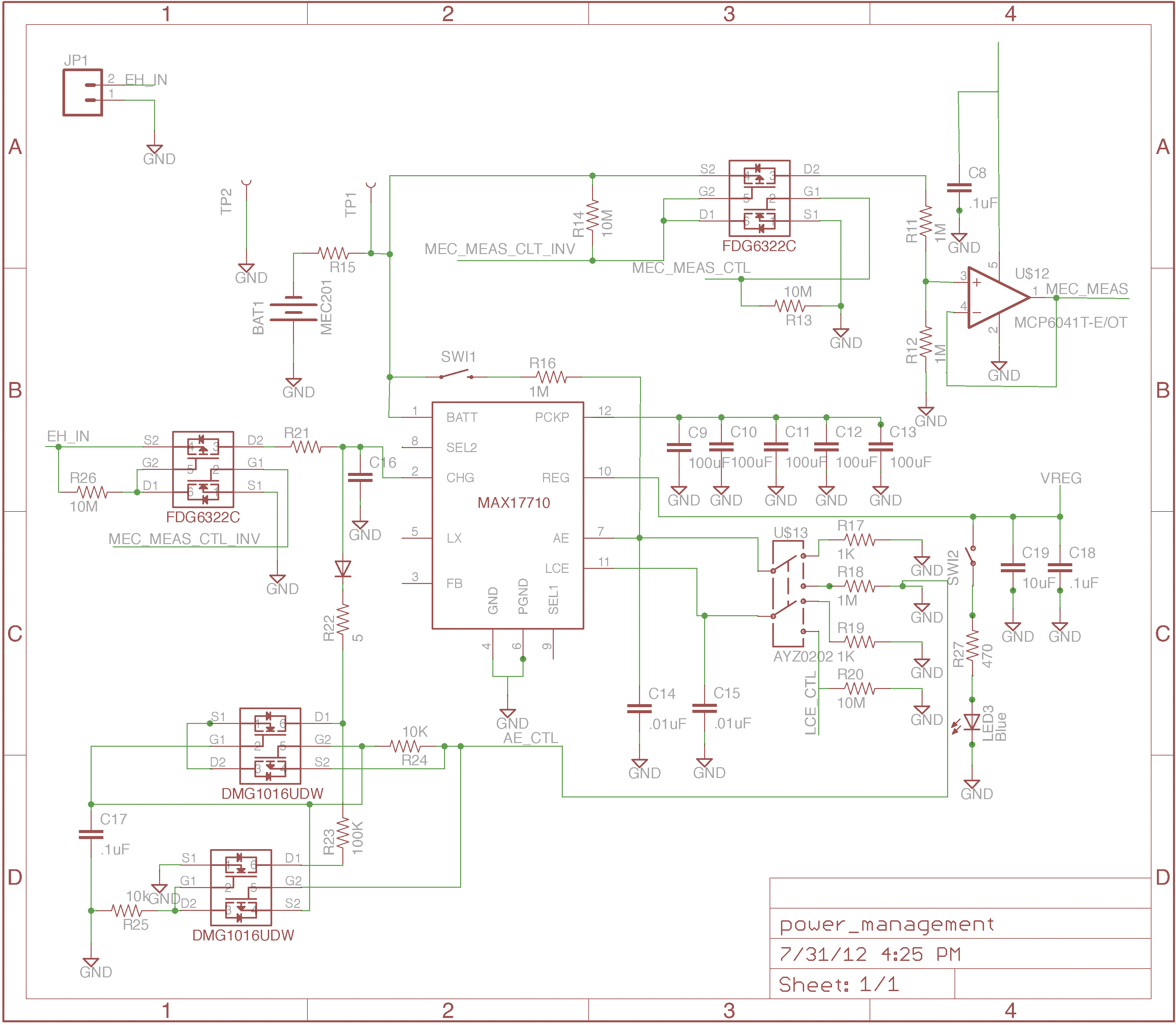
The three main components, the battery, the energy harvesting source, and the IC are integrated together to form the power management system. The system requires supporting components to produce the desired operational values. Table 4.1.2.1.4-1 lists of all the components used to build the power management circuit.

|  |  |  |
| --- | --- | --- |
| Description | Item/Value | Qty |
| IC | MAX17710 | 1 |
| Micro Energy Cell | MEC201-7S | 1 |
| Photovoltaic Cell | AM-1801 | 2 |
| Resistor | 5Ω  470Ω  1kΩ  2kΩ  10kΩ  100kΩ  1MΩ  3.01MΩ  10MΩ | 1  1  3  1  2  1  4  1  4 |
| Capacitor | .01µF  .1µF  .7µF  10µF  100µF | 2  4  1  2  5 |
| Complimentary Paired  MOSFETs | DMG1016UDW  FDG6322C | 2  2 |
| 600 nA, Rail-to-Rail Input/Output Operational Amplifier, 5-Pin SOT-23, Extended Temperature, | MCP6041T-E/OT | 1 |
| Schottky Barrier Diode | BAS70KFILM (NP) | 1 |

*Table 4.1.2.1.4-1 Bill of Materials for Power Management System*

***4.1.2.2 Circuitry***

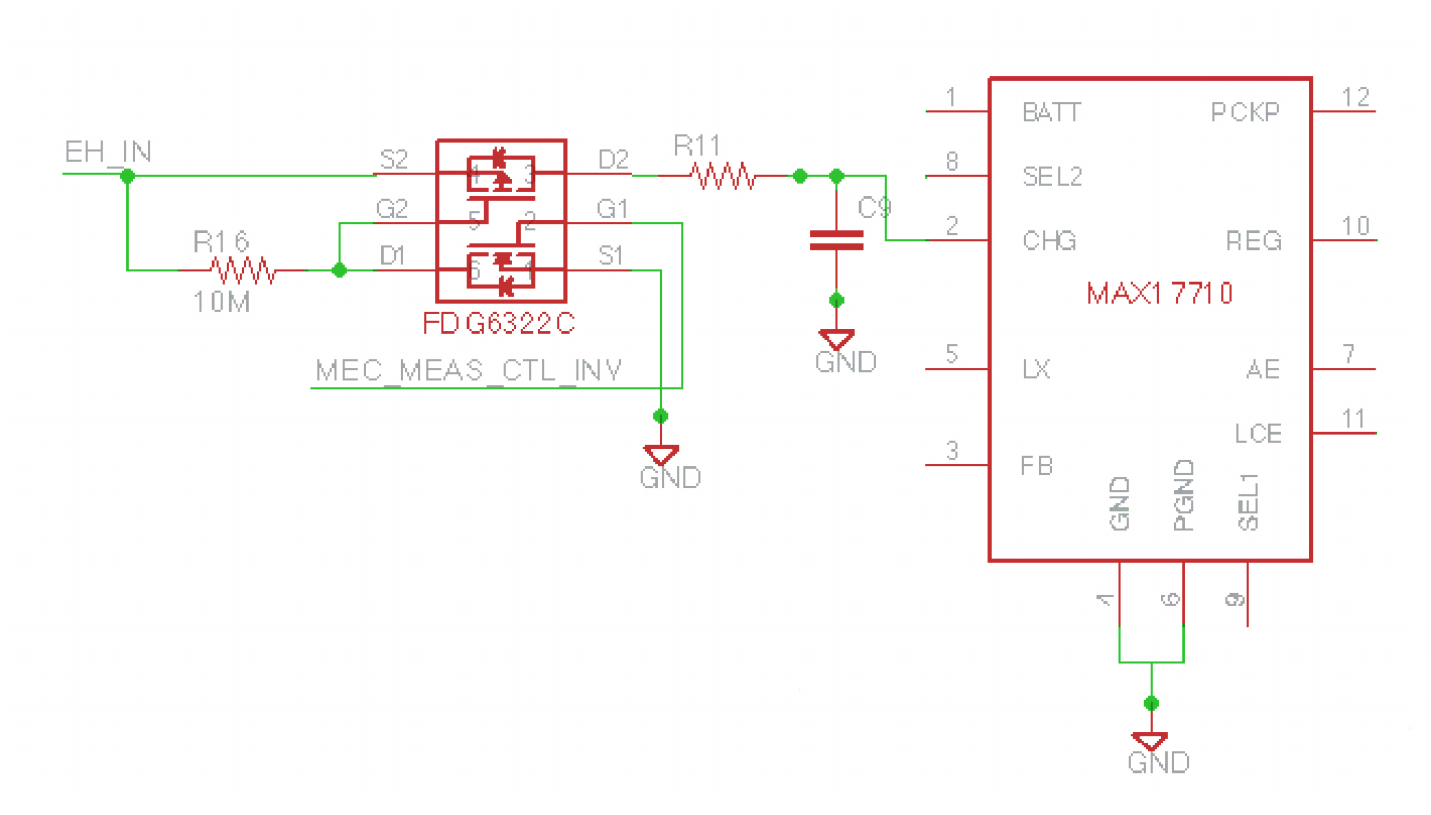
The power management system revolves around the PMIC. All other components are used as inputs or outputs to the PMIC. Figure 4.1.2.2-1 contains a schematic of the complete power management system. The system can be divided into smaller subsystems, which include the charging system, the battery cell system, the unregulated output system, the regulated output system, and the control system. In the following sub-sections we will go over each subsystem and its function.



*Figure 4.1.2.2-1 Power Management System Schematic*

***4.1.2.2.1 Energy Harvesting Source Sub System***

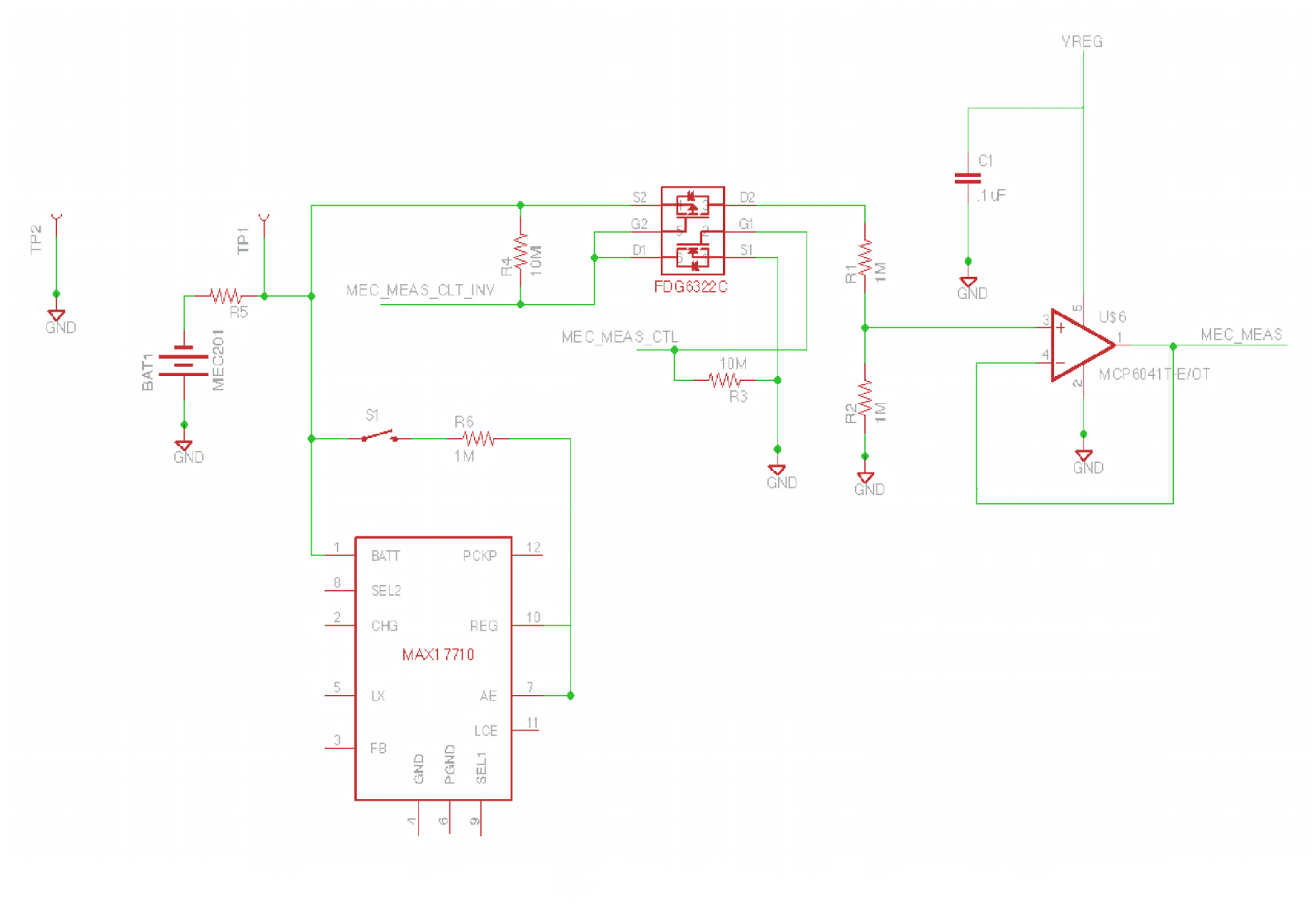
The charging circuitry contains a complimentary pair of MOSFETS, 2 resistors and a capacitor. Figure 4.1.2.2.1-1 is a schematic of the charging circuit. The Capacitor, C7, is used to stabilize the shunt regulator within the PMIC. EH In is the voltage charge from the solar panel which is connected to pin 1 of header J1. In normal operation the voltage signal from EH In passes through the MOSFET switch to pin 2 on the MAX17710. When a voltage reading needs to be made from the battery, the voltage charge from the solar panel is grounded so that the micro controller can obtain more accurate readings. The micro controller sends a signal through the MEC\_Meas\_Ctl\_Inv (pin 27 on the micro controller) that disconnects the charging voltage from the PMIC. When the reading is completed the voltage charge is reconnected to the PMIC. If the voltage of the solar panel is more than the voltage in the battery, the current is passed directly to the battery. When the voltage of the solar panel rises above 4.15V the input linear regulator within the PMIC turns on to limit the charging voltage to 4.125V.



*Figure 4.1.2.2.1-1 Charging Source Circuit*

***4.1.2.2.2 Battery Sub System***

The battery sub system contains 1 MEC201-7S battery, 5 resistors, 2 capacitors, 1 operational Amplifier, and 1 push button. Figure 4.1.2.2.2-1 is a schematic of the charging circuit. The push button is used to turn on the regulated output at the initial startup of the device. Under regular operations, the sub system is designed to supply a low capacity energy storage device, such as the MEC201-7S, with a constant charge from the charging source circuit. The battery subsystem also supplies power for the regulated output on pin 10 and the unregulated output at pin 12.

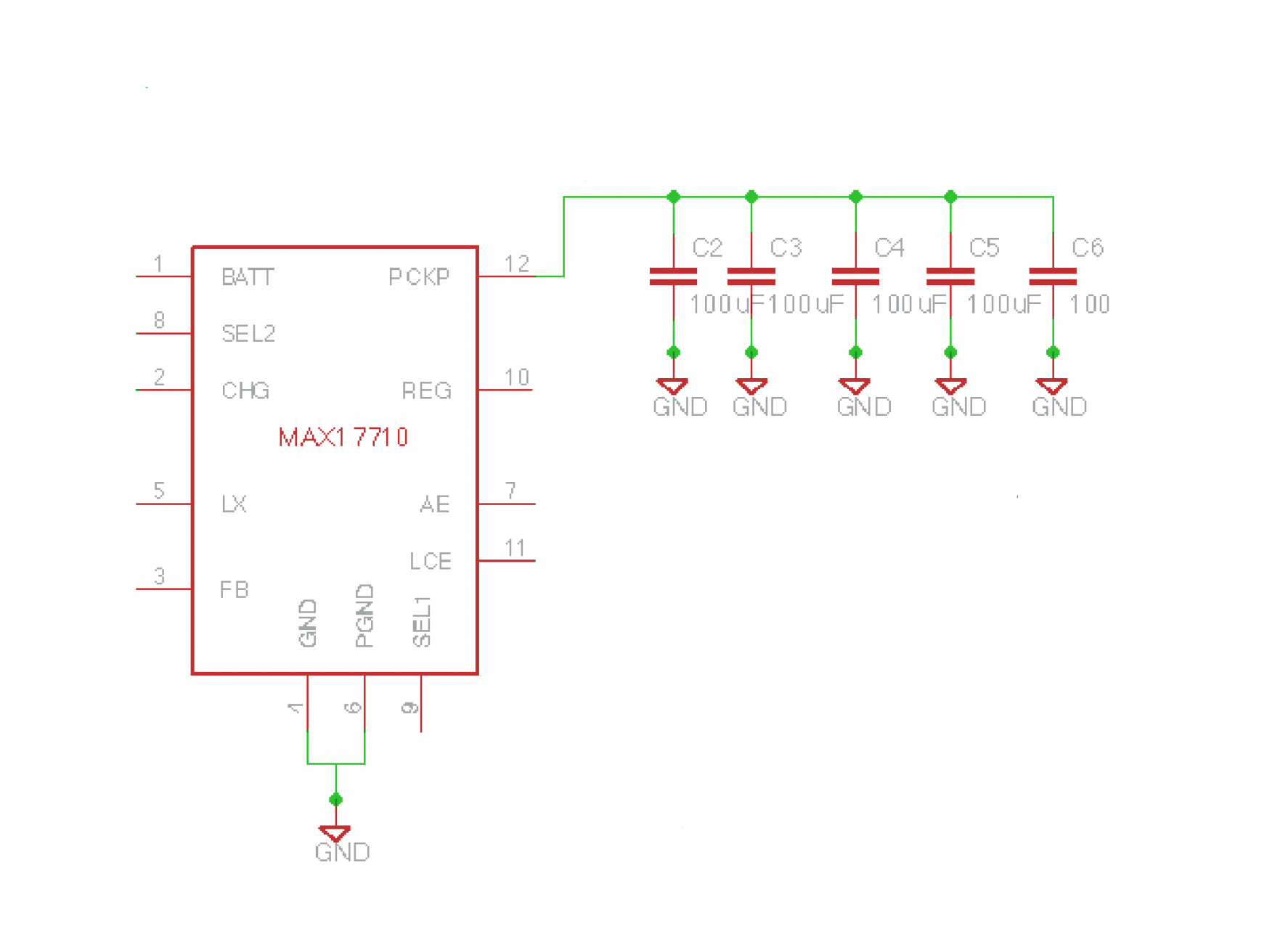


*Figure 4.1.2.2.2-1 Battery Sub System Schematic*

The battery circuit also contains a voltage sensor. The voltage sensor is not necessary for the PMIC to function correctly. It is included in the circuit so that the voltage level on the battery can be transmitted to the server. This data allows the end user to monitor the battery levels of each node in the wireless sensor network. When the micro controller sends a signal through the MEC\_Meas\_Ctrl wire (pin 27 on the micro controller), the MOSFET switch connects the voltage sensor to the battery and a reading is made from the output terminal of the operational amplifier. After the reading, the MEC\_Meas\_Ctrl signal goes low and the MOSFET disconnects the voltage from the battery circuit again.

***4.1.2.2.3 Unregulated Output Sub System***

The unregulated output sub system contains 5 capacitors. Figure 4.1.2.2.3-1 is a schematic of the unregulated output circuit. The battery charges the five capacitors on pin 12 until the voltage on pin 12 reaches 3.7V. This will store 1.85As of energy in the capacitors to be used in the regulated output for high current pulses on the load. By using the capacitors to power the load, the battery cell is protected from over discharge. If the amps required for the load are beyond the means of the battery and capacitors to supply, the voltage on pin 12 will collapse and the regulated output will be turned off. This is not a concern for the sensor node circuit. The 1.85As of capacity will be more than enough to handle the full load of the circuit. The maximum current draw will be approximately 20 mA which is pulled by the transceiver when a state change is detected by the sensor. The power stored in the capacitor will be more than sufficient to handle multiple transmission events without needing to draw power form the battery cell.



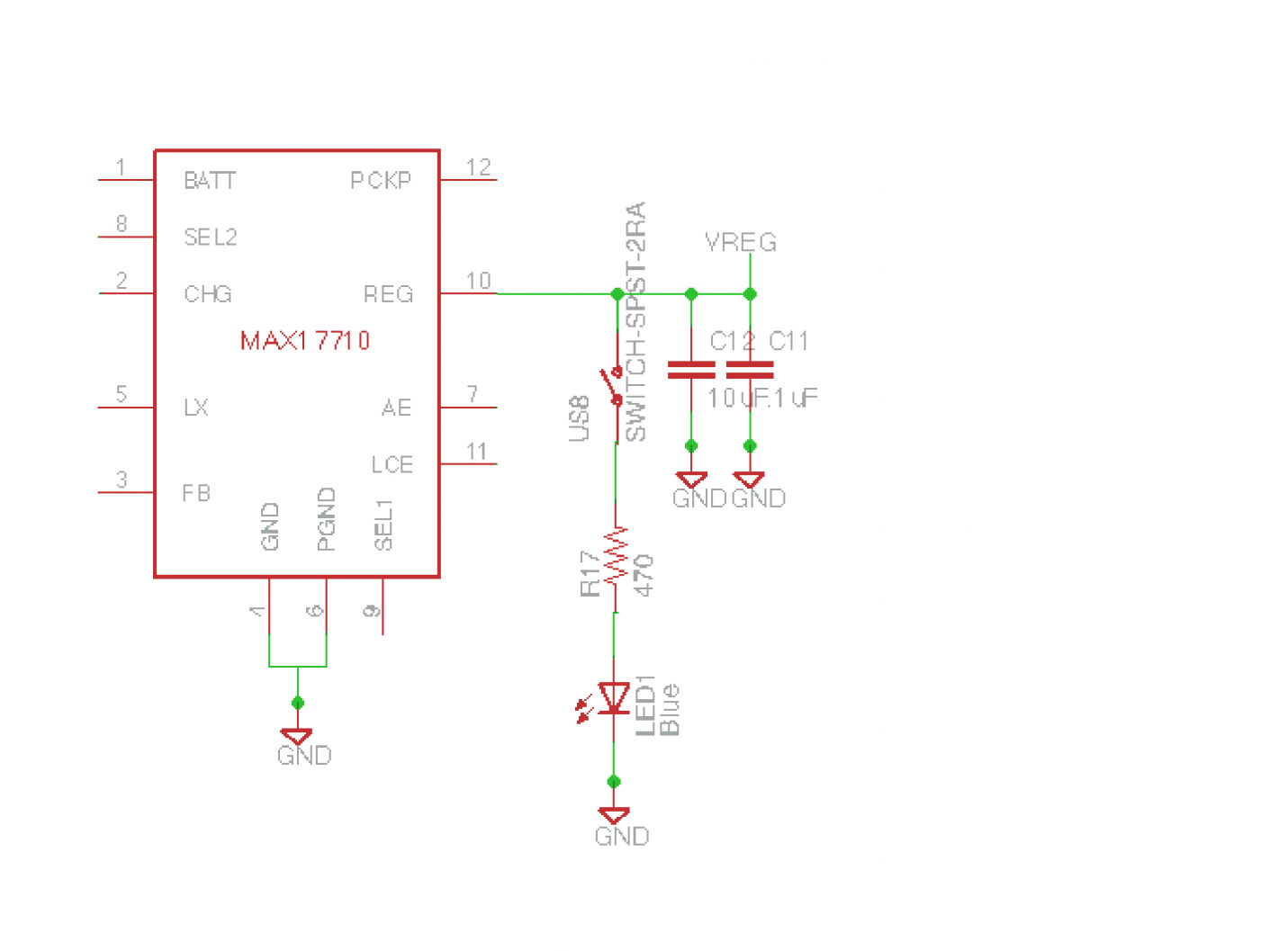
*Figure 4.1.2.2.3-1 Unregulated Output Sub System Schematic*

***4.1.2.2.4 Regulated Output Sub System***

The regulated output sub system contains 1 resister, 2 capacitors, 1 push button, and 1 blue led light. Figure 4.1.2.2.4-1 is a schematic of the unregulated output circuit. The pins that are part of the sub system are pins 9 and 10. Pin 9 is used to select one of the three regulated voltages (1.8V, 2.3V, 3.3V). To select 1.8V, the battery voltage is connected to pin 9 (high voltage). To select 2.3V, pin 9 needs to be tied to ground (low voltage). To select 3.3V, pin 9 is set to three-state (high impedance).

The circuit contains a push button that serves as a tester for the output voltage. When the button is pushed the led will turn on if the regulated voltage is available. The regulated voltage is then tied into the rest of the sensor node circuit to supply the power for all of the components.

When the circuit is first turned on, the regulated output is not available. The PMIC first charges the unregulated output pin where the capacitors are being used as an energy buffer. Only when the charge on the unregulated voltage rises to 3.7V does the regulated output become available. This is to protect the battery cell from an excessively high startup discharge.

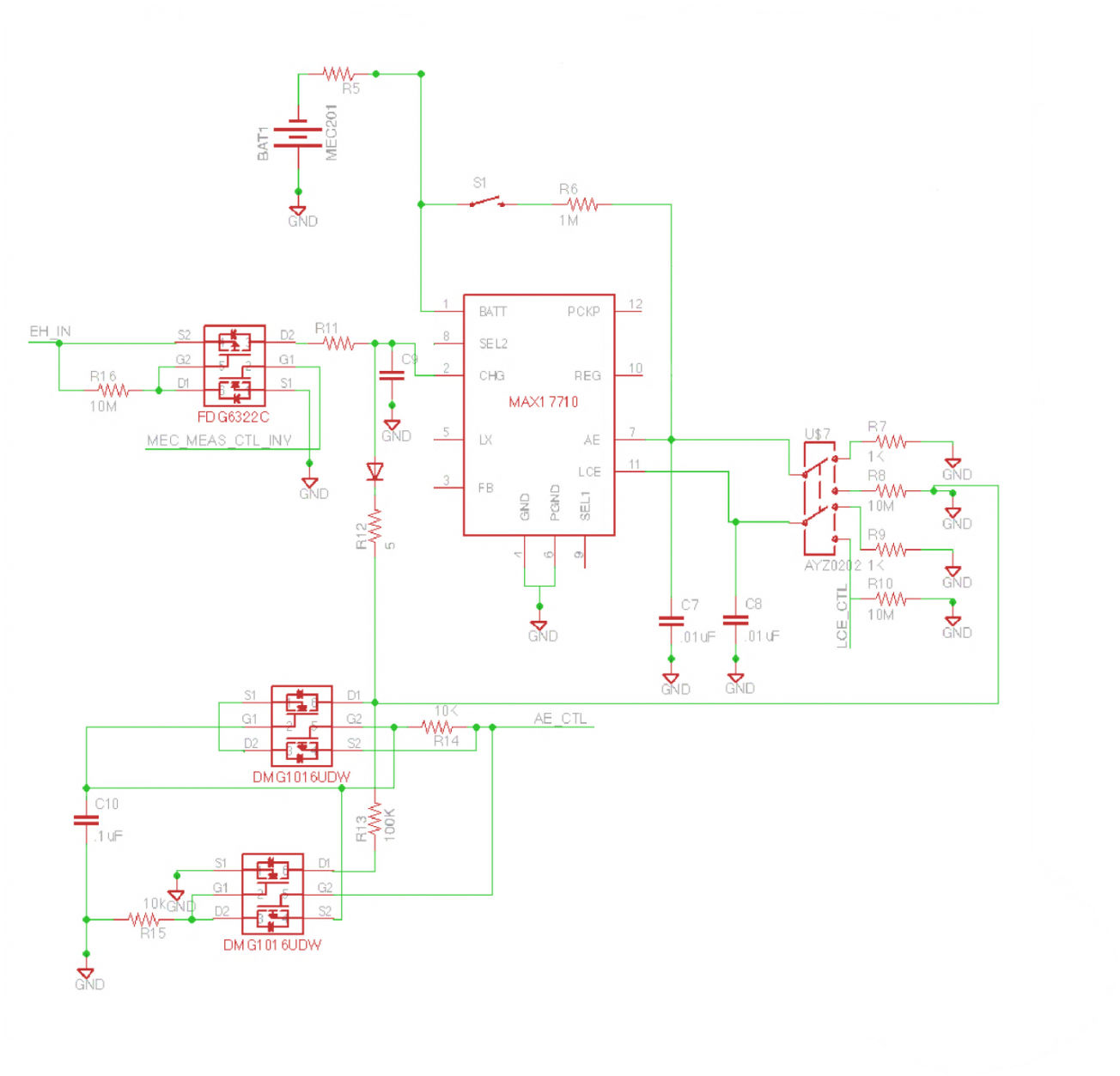
****

*Figure 4.1.2.2.4-1 Regulated Output Sub System Schematic*

***4.1.2.2.5 Control Sub System***

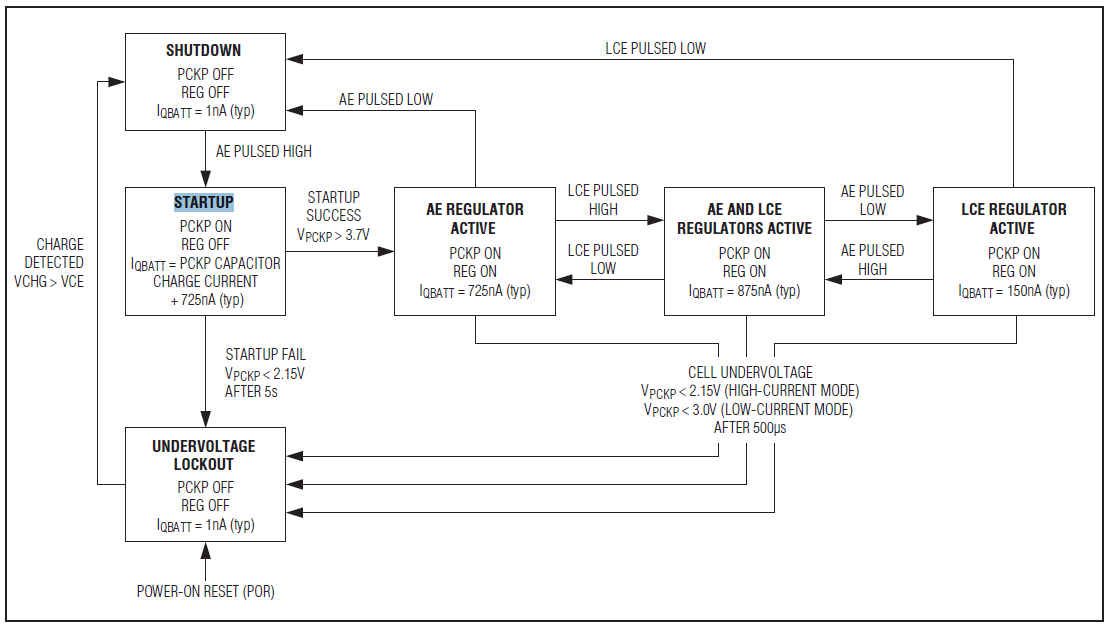
The control sub system contains 1 slide switch, 9 resistors, 3 capacitors, 1 Schottky barrier diode, 1 push button, and 2 complimentary MOSFET pairs. Figure 4.1.2.2.5-1 is a schematic of the control sub system. There are two control lines, the LCE control line (pin 42 on the micro controller) and the AE control line (pin 41 on the micro controller). The AE control line controls the high power regulator output. When AE is pulsed high, the stored energy in the capacitors is available at high-current use. In this mode the PMIC can supply up to 75mA to the sensor node circuit. When AE is pulsed low the PMIC shuts down unless LCE has been pulsed high.

The LCE pin sets the PMIC to a low-current mode. In this mode the PMIC will only supply a maximum of 50µA to the load. The sensor node circuit will be in sleep mode while the PMIC is set to low current mode. To set the PMIC to low-current mode, AE must first be pulsed high, then LCE is pulsed high and finally AE is pulsed low. If LCE is pulsed low while in low-current mode the PMIC will shut down. To return to high-current mode from low-current mode, AE is first pulsed high and then LCE is pulsed low. Figure 4.1.2.2.5-2 contains a state diagram for the regulated output.



*Figure 4.1.2.2.5-1 Control Sub System Schematic*

The PMIC also has a built in voltage lockout regulator. If the voltage on the capacitors on pin 12 is less than 2.5V in high current mode, or if the voltage on the capacitors on pin 12 is less than 3V in low current mode, then the PMIC is set to low voltage lockout mode. The PMIC needs to be reset before it is released from the lockout mode.



*Figure 4.1.2.2.5-2 Regulator Output State Diagram: Source[23](permission granted)*

**4.1.3 Communication circuit Design**

In our design the microcontroller needs to be able to connect to internet server so the user will be able to check their cellphone at any time and find an empty spot in the parking garage. One of the solutions is having a transceiver that can transmit and receive data from the sensor board to the server. Looking at Microchip wireless solutions, the MRF24J40MA radio frequency transceiver module fits many of our requirements. MRF24J40MA is a 2.4 GHZ IEEE std.802.15.4 compliant, surface mount module with matching circuitry, integrated crystal, internal voltage regulator and PCB antenna. The MRF24J40MA module is compatible with Microchip’s ZigBee, MiWi and MiWi P2P software stacks. This module interfaces with most of the Microchip PIC microcontrollers with connecting of a 4-wire serial SPI interface, interrupt, wake, Reset, Power and ground. The MRF24J40MA module is based on Microchip MRF24J40 transceiver IC. The serial I/O (SCK, SDI, SDO and CS), RESET, WAKE, INT and CLKOUT pins are brought out to the module pins.

***4.1.3.1 Protocol design***

This module can be programmed with different protocols within 802.15.4 specifications. As we mentioned before Microchip offers three protocol options. ZigBee has 100k memory use app and has unlimited hops, however, MiWi which is a lighter version of ZigBee is divided into two parts.

* MiWi Mesh, Star/Mesh topology uses 20k memory.
* MiWi P2P, Star/P2P topology uses 10k memory.

There are different factors that consider in the amount of memory being used. Memory usage depends on protocol structure, optimizations, security and device type. MiWi P2P was chosen to be used mostly because of simplicity over ZigBee.

There are three topologies that MiWi protocol can support. The first one is Star topology that has one Personal Area Network coordinator that initiates communications and accepts connections from other devices. The connection for the End devices is only possible when they have connections with the PAN coordinator. The second one is Peer to Peer topology that permits additional coordinators to be added to the network. This topology also has one PAN coordinator that starts communication from end devices. The last one is a mesh topology which is very similar to peer to peer.

Coordinator

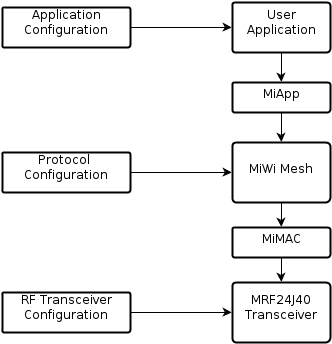
Coordinator

*Figure 4.1.3.1-1 Peer to Peer topology: Source [14]*

From the functional perspective the PAN coordinator is a Full Function Device (FFD) that is always on and powered from the main source and the End devices can be FFD or RFD. RFD is powered from the battery and can be turned off. In this topology end devices that are FFDs can have multiple connections, however, the end devices that are RFDs can only connect to one FFD and cannot connect to another RFD.

MiWi P2P protocol only supports one-hop communication; therefore the data transmits through extended Organizationally Unique Identifier (EUI) or long address. For our transceiver the unique address length is 2 to 8 bytes. The short addressing can be used when transmitting of data is done from stack.

The basic function of wireless communication protocol is to transmit and receive data between two nodes. Transmitting and receiving data functions are provided by Media Access Controller (MAC) layer and Physical (PHY) layer. Microchip MAC layer, known as MiMAC, is used for communication protocols and transceivers supported by Microchip for short range, low data rate, and low power wireless applications. MiMAC interface is used to bind the application level interface with hardware receivers. The following block diagram explains the stack and program interface. The application configuration consists of MiApp interface which is common for all Microchip protocols. MiApp is used to choose the protocol and MiMac is to choose the RF Transceiver.



*Figure 4.1.3.1-2 Block diagram of MiWi: Source [13]*

***4.1.3.2 Software Design***

The next subject that we need to consider in our design is standardized software programing interface between MiMAC and Microchip. There are some basic functions that are used to configure the Mac layer in a MiWi P2P protocol. In this section we are going to explain four of these functions.

• **MiMAC\_Init**

This function call initializes the behavior of the MiMAC layer. This function is called to enables the transceiver to send data packets when the Repeat bit is set. It also defines the length of the addresses for the transceiver. The length can be defined between 2 to 8 bytes. Using PAddress, this function set the pointers to point to the permanent addresses of the transceiver.

BOOL MiMAC\_Init (MACINIT\_PARAM initValue);

**• MiMAC\_SetPower**

This function sets the output power of the transceiver. The output power represented by one byte. The transceiver’s job is to interpret the input value and set the proper output power. This function is set in the PHY layer and the MiMAC layer is just an intermediate layer to pass the setting to the PHY layer of the transceiver.

BOOL MiMAC\_SetPower (BYTE outputPower);

**• MiMAC\_SetChannel**

This function sets the frequency needed for the RF transceiver. This function has two inputs. Channel input is defined from 0 to 31 and finds the center frequency that would make the transceiver work. Offset input is used as an extra configuration to set the center frequency at any frequency not defined by the channel. The offset frequency is always less than the frequency gap between adjacent channels. We are not going to use all the channels for our transceiver. Setting the right input channel is important because if the input parameter channel is not properly supported by the RF transceiver, the return value will be false.

BOOL MiMAC\_SetChannel (BYTE channel, BYTE offsetFreq);

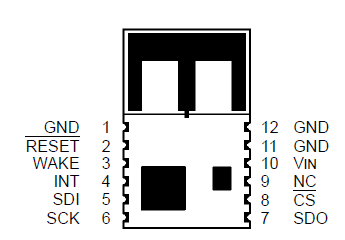
**• MiMAC\_SetAltAddress**

This function sets the alternative network address after the wireless node joins the network. This function has two inputs. One is the pointer that points to the address. This input is required to support the network addresses to identify the sensor node in the network. The second input is PANID that is used to identify the network itself.

BOOL MiMAC\_SetAltAddress (BYTE \*Address, BYTE \*PANID);

***4.1.3.3 Circuit Design***

MRF24J40MA has many features that work well with our system and there are sample codes form microchip to interface with our microcontroller. In order to better understand the MRF24J40MA, we need to recognize what each pin is stand for and capable of doing for our circuit design. MRF24J0MA module requires three additional pins excluding the standard four SPI pins. These pins are RESET, INT, and WAKE. The memory that is needed for this module is as low as 3kb. This specification makes it possible to use this module with any Microchip microcontroller.



*Figure 4.1.3.3-1 Pin Diagram: Source [32](Permission Pending)*

The Input/Output from the microcontroller will connect with the SPI Chip Select (SC) input and clock (SCK), while the data will exchange through the SPI data in and out (SDI,SDO) pins. All data communications with the module are through the SPI interface.

*Power and Ground Pins (Vin/ GND)*

The power and ground pins are very self-explanatory. The operating voltage for the transceiver module is 3.3V. This power is produced from the rechargeable MEC201 battery attached to the solar panel daughter card.

*Serial Peripheral Interface (SPI)*

The MRF24J40MA communicates with our microcontroller via a 4-wire SPI port as a slave device. The MRF24J40MA supports SPI mode 0, which means that SCK idles in a low state because the CS pin must be held low while communicating with the transceiver module. Data is received by the transceiver module with the SDI pin and is clocked in on the rising edge of SCK. The data is transmits with the SDO pin and is clocked out by the falling edge of SCK.

*Reset Pin (RESET)*

An external hardware Reset can be performed by asserting the RESET pin 2 low. MRF24J40MA will be released from Reset approximately 250 us after the RESET pin is released. The Reset pin has an internal weak pull-up resistor.

*Interrupt Pin (INT)*

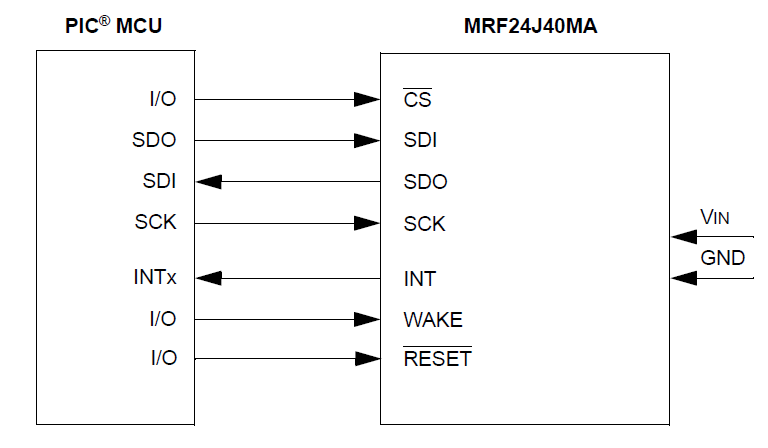
The Interrupt (INT) pin 4 provides an interrupt signal to the microcontroller from the transceiver module. Interrupts have to be enabled and unmasked before the INT pin is active. When an enabled interrupt occurs, the interrupt pin will remain at its interrupt state, as determined by the INTEDGE bit, until all of the flags are cleared by the host controller.

*Wake pin (WAKE)*

The WAKE pin 4 provides an external wake-up signal to the MRF24J40MA from the microcontroller. The WAKE and Sleep mode are used in combination. It is one of microcontroller duty to put the transceiver to sleep and wakes it up. We can put the transceiver module in to immediate Wake up mode by setting IMMWAKE bit to 1 and put in to sleep mode by setting SLPACK bit to 1. Both modes will be cleared by hardware to 0 automatically.

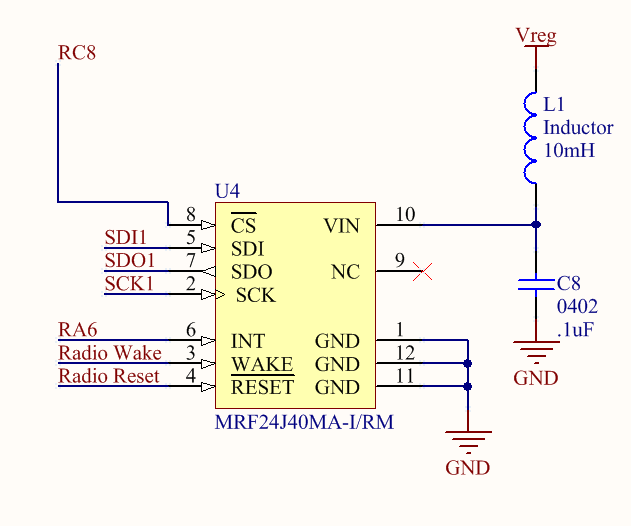
***4.1.3.4 Circuitry***

The figure below shows the schematic design for our MRF24J40MA transceiver module. This part of the schematic is just for the transceiver module. The complete section of the circuitry is discussed in the microcontroller section of this document.



*Figure4.1.3.4-1 MRF24J40 interface to microcontroller Source [32]*

*(Permission Pending)*



*Figure 4.1.3.4-2 MRF24J40MA module schematic*

***4.1.3.5 PCB Design***

The MRF24J40MA is a surface mount module that has an integrated PCB antenna. It is recommended from data sheet to mount the module on the edge of the host PCB and the area around the antenna has to be kept clear of metal objects for approximately 1.2 inch. Moreover, a host PCB ground plane around the module acts as a counterpoise to the PCB antenna; therefore, the ground plane needs to be extended at least 0.4 inch around the module.



*Figure4.1.3.5-1 PCB Layout: Source [32] (Permission Pending)*

**4.1.4 Microcontroller**

The microcontroller in the sensor node will coordinate all activity of all the circuit peripherals as well as identify an event (i.e. car leaving/entering spot or garage). In order to reduce the amount of power consumed by the sensor node the microcontroller will determine what peripherals will be on, which will be in sleep mode and when these respective states are to occur. Algorithms will be created to determine the sampling rates based on activity and to detect an event, based on this gathered and computed information it will then initiate a communication routine to report its findings back to the server through supporting infrastructure (i.e. sub-gate -> gateway -> server). The following sections explain the functions and design of the microcontroller in more detail.

***4.1.4.1 Peripheral Connections***

For the sensor node thirty-three of the available forty-four pins will be needed. The microcontroller will be connected to the four main sub-systems of the sensor node; those being Power Management, Transceiver, Sensor and supporting components. The transceiver will use a 4 wire SPI serial connection to communicate with the microcontroller. The sensor will use a 2 wire I2C serial connection and the power management circuit will use 3 general-purpose pins and 1 analog pin. The supporting components will include the programming header, the UART header and the LED indicator lights. The following sections go into more detail involving the individual tasks that will be handled by the microcontroller.

**Power Management –** The Microcontroller will coordinate the Power Management (PM) circuit using three general-purpose pins and one analog pin available on the microcontroller. The first two outlined below are meant to measure and control the charge on the battery, the latter two are meant to control the PMIC high-power and low-current regulator enables. Pins 15 and 27 will be connected to MEC\_MEAS and MEC\_MEAS\_CTL respectively whose use and characteristics are outlined in sections 4.1.2.2.1-4.1.2.2.2. Pins 41 and 42 will be connected to AE\_CTL and LCE\_CTL respectively; the characteristics of these control lines are described in section 4.1.2.2.5. The connections are shown below in figure 4.1.4.1-1. Functions and declarations will need to be created to perform the following tasks.

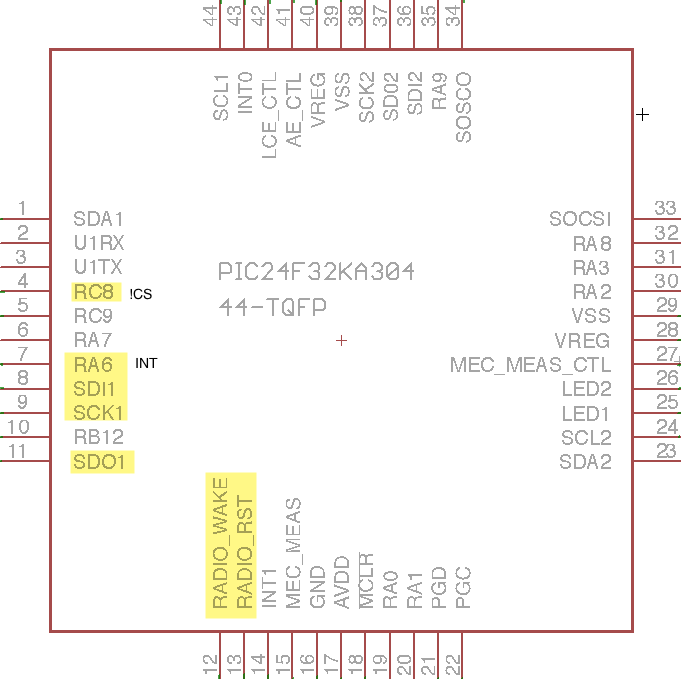
* Set control signals for the PMIC (AE\_CTL, LCE\_CTL and MEC\_MEAS\_CTL)
* Measure the battery charge after appropriate signals have been sent (i.e. pulse MEC\_MEAS\_CTL low to disconnect EH, MEC\_MEAS\_CTL\_INV goes high and battery can be accurately read). The ADC can then be used to measure the battery voltage
* Put circuit peripherals to sleep if voltage crosses a certain threshold, this can be a software interrupt that puts all circuit components not required for basic functionality to sleep until the required voltage is detected.

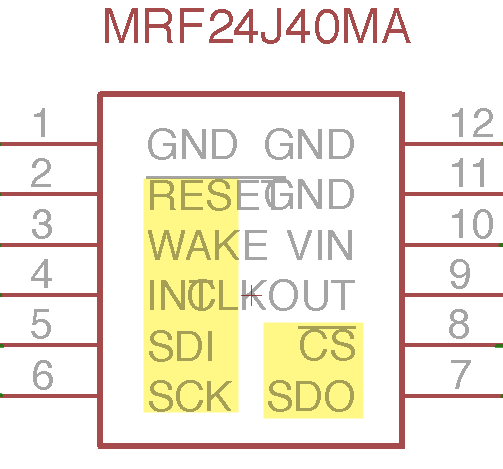
A class will be created for sensor data where the measured battery voltage will be stored. Finally de-coupling capacitors are placed in parallel with every pair of power supply pins.

|  |  |
| --- | --- |
|  |  |
| *Figure 4.1.4.1-1**Microcontroller and Power Management* | |

**Transceiver & EEPROM -** The transceiver will use a four wire serial SPI connection to the microcontroller with the standard lines connected to pins 4,8,9 and 11 respectively. Three additional connections are required to operate the transceiver; these control and are connected to the interrupt (INT), Wake and pins on the transceiver and are connected to pins 7, 12 and 13 respectively. The EEPROM is also connected to the SPI bus and is used to store the nodes EUI-48 MAC address, the characteristics and operation of the transceiver and EEPROM are outlined in section [fill in the blank]. The transceiver will use the MiWi protocol, which is Microchips proprietary wireless protocol. The code necessary to successfully transmit data is vast but can be broken down into three main categories.

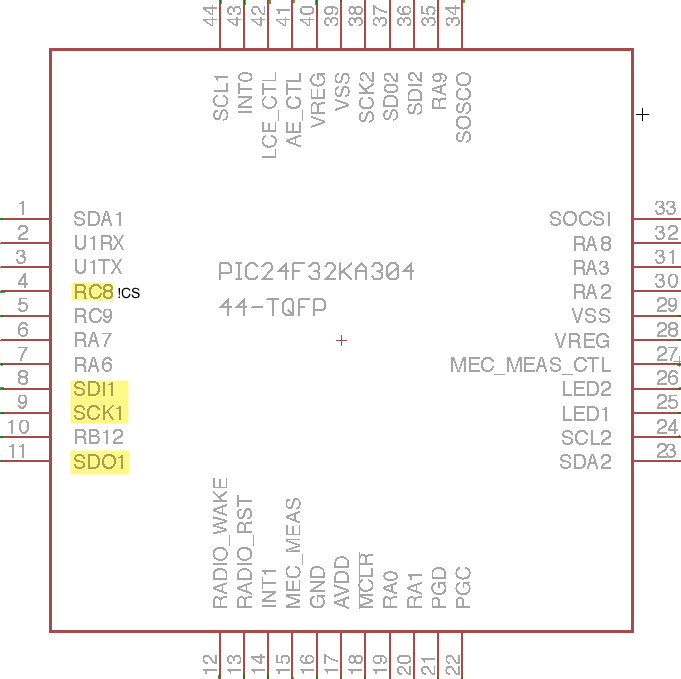
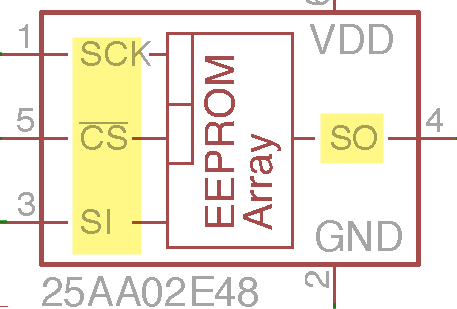
* Addressing – This involves the reading/writing the EUI-48/EUI-64 MAC addresses to the EEPROM, once prepared the addresses are transmitted accordingly.
* Transmission – Sending and receiving data/address packets according to the MiWi peer-to-peer topology.
* Management – This includes clocking, power management and other operational routines.





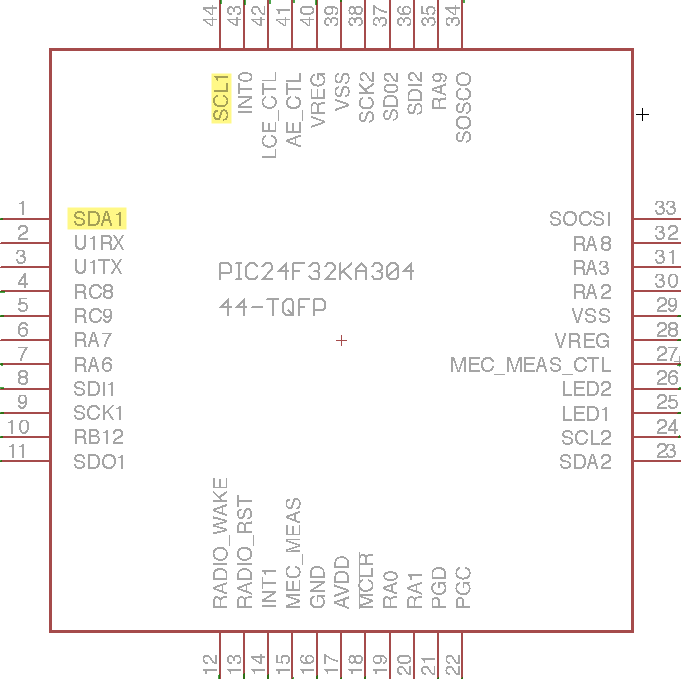
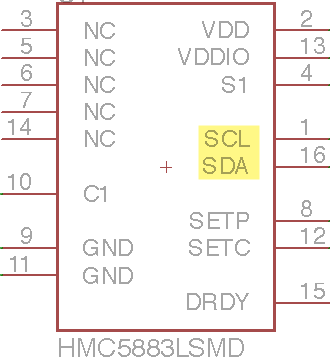
*Figure 4.1.4.1-2 Microcontroller and transceiver*

Below in figures 4.1.4.1-2 and 4.1.4.1-3 the circuit connections are shown.

*Figure 4.1.4.1-3 Microcontroller and EEPROM*

**Sensor –** The microcontroller has two available I2C buses, we will use the first available bus where SDA1 is located on pin number 1 and SCL1 on pin number 44. Refer to figure 4.1.4.1-4. They will be connected to pin numbers sixteen and one respectively on the HMC5883L sensor.

*Figure 4.1.4.1-4 Sensor and Microcontroller connections*

*Configuration* **-** First the HMC5883L sensor as described in section 4.1.1.1 consist of thirteen registers three of which will need to be written to in order to configure the sensor for our application. Configuration register A is used to select the number of samples averaged per measurement output, set the rate at which measurements are written to the data output registers and set the measurement mode. By default these are set to 8, 15Hz and normal measurement mode respectively; we will leave the defaults in place and will not require any configuration of register A. Configuration register B is used to set the gain of the sensor and is set to +/- 1.3 Ga, we will leave the default in place for initial testing but will need to change this value in order to fine tune the sensors. A function will be required in order to rewrite Configuration Register B.

*Writing Data -* The procedure used to write using the I2C protocol is the same regardless of which register is being written to, so it is logical to write one routine to write to the registers and call it when necessary. This function can then be called from other functions used to write to the sensor registers, in order to configure the various settings. It first initiates the I2c transmission; it then sends the write address and data and finally closes the I2C transmission.

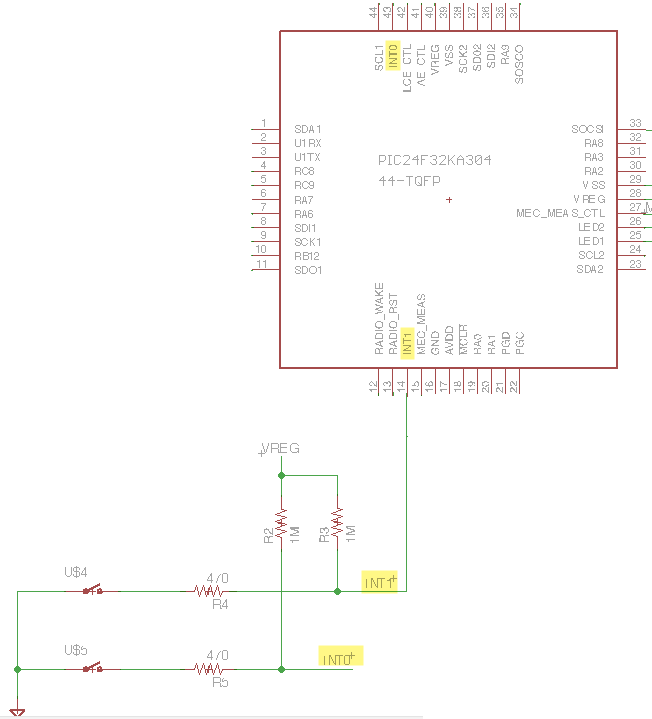
With the write function in place a function can then be passed the gauss value as an argument, based on the passed value a relative register value is then chosen and passed to the Write function along with the register address. The register address will be set globally as to allow using it in various functions. One important fact to note is that the settings are stored in the most significant three bits so the register value must be shifted before calling the Write function.

The final register required to configure the sensor is the Mode register, this register determines whether the sensor is in continuous mode or in single measurement mode. The mode is set to single measurement mode by default, we will need it to be in continuous mode for our application so a simple function can be written that takes in the 8bit hexadecimal value for the mode as an argument and passes it to the Write function. The above three functions outlines are all that is necessary to configure the HMC5883L sensor given our application. The following sections cover the routines needed to read the sensor data into the microcontroller.

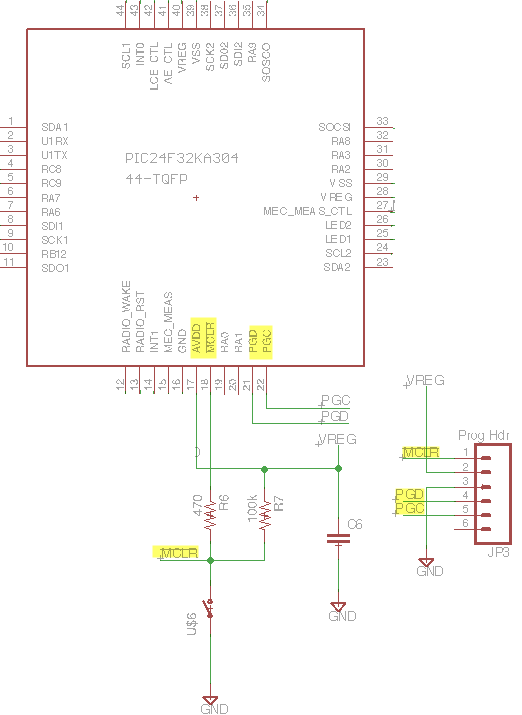
*Reading Data* **-** Similar to writing to the sensor we will need a function that reads data from the sensor that can be called by other functions. The Read function will take in 48 bits of data store in in a buffer array. This array can then be parsed by other functions to either provide raw or scaled data, for this project the raw data is sufficient for detecting events. To read from the sensor a I2C transmission must first be initiated to write the beginning address of the registers that are going to be read, this address will be stored in a variable for semantic purposes and is equal to 0x0316 (i.e. the first data register). In the following examples both the read and parse concepts are portrayed in pseudo code. Once this data has been retrieved it can then be used to perform calculations that will determine if an event has occurred. A function will be written that is passed the starting address of the register to be read and the data length to be read (i.e. equal to six for reading the X Y Z axis each two bytes long so a six byte read length). It then initiates a I2c transmission and data is then stored into a buffer array where each array element is a byte, the function then returns the buffer array. With this buffer now returning the required data one last function can be written that will take in the data into a another buffer array using the Read function; it then parses the data by shifting the MSB eight positions to the left and performs a logical OR and stores the result to provide a 16bit data output for computing events.

**Supporting Circuitry –** The microcontrollerwill be setup to have two interrupts INT0 and INT1 connected to pins 43 and 14 respectively that are to be used to reset certain functions on the microcontroller. The interrupt switches are momentary switches that are coupled with current limiting resistors connecting to Vreg, the interrupts are active low and function when the switches are closed (i.e. pulling the pins low activates the interrupt). In figure 4.1.4.1-5 the circuit is depicted.

*Master Clear -* A similar circuit is used to implement (Master Clear), which will be used to reset the device and is necessary for the in-circuit-debugger and programmer. The device programmer and debugger will drive the pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. The circuit serves to isolate from the capacitor C6 during debugging and programming. All components associated with the pin will be placed within 0.25 inch (6 mm) of the pin. In order for the master clear to be used bit 7 (FPOR<7>) in the Reset Configuration (FPOR<7:0>) register needs to be set high. If the momentary switch is pressed bit 7 (RCON<7>) in the Reset Control (RCON<15:0>) Register goes high and the device is reset. Also when the microcontroller needs a hard reset to exit deep sleep mode a event can be used. The UART connection will be used to connect to a PC for debugging purposes.



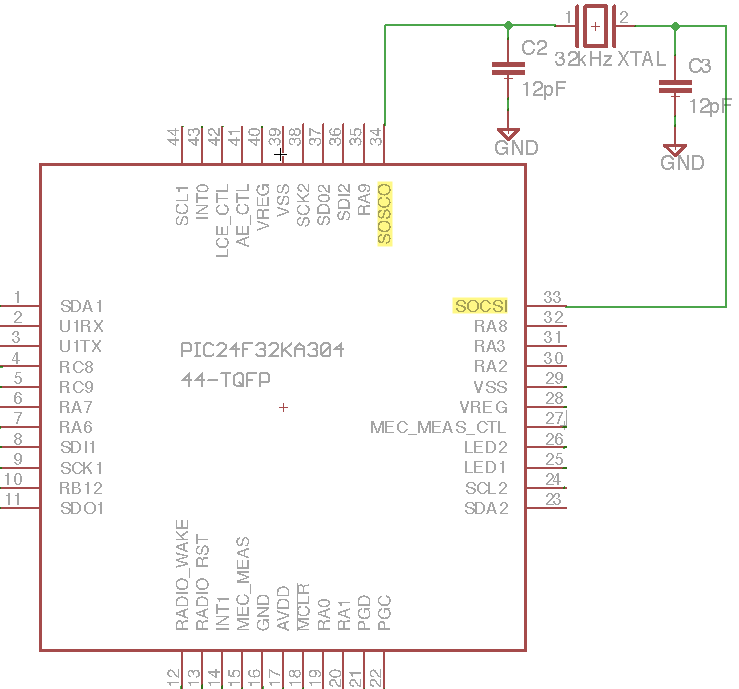
*Figure 4.1.4.1-5**Microcontroller and External interrupt circuit.*

****

*Figure 4.1.4.1-6 Microcontroller Master Clear circuit and Program Header*

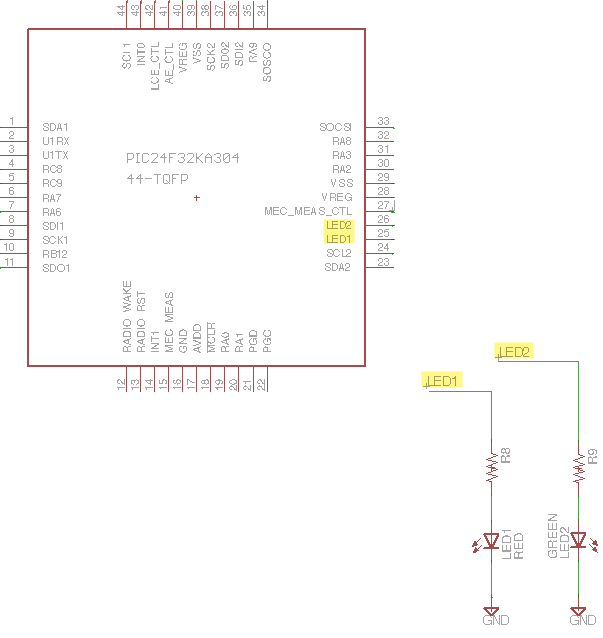
*Secondary Oscillator -* We will also use a secondary 32kHz secondary oscillator in our design. The secondary oscillator will be used to set as the source clock rate when the microcontroller goes into sleep mode. When in sleep mode the clock speed will have a 1:256 ratio to that of the internal 8MHz clock. In order to create the sleep delay a function will be created that takes in the sleep delay in seconds and a pointer to an exit flag as arguments. This function will configure the Timer1 Control Register (T1CON<15:0>), it will write 0x0032 to T1CON. This will temporarily disable Timer1 by setting bit 15 (T1CON<15>) the Timer1 On Bit (TON) to 0, which stops the 16-bit Timer1. It will also set bits 5-4 (T1CON<5:4>) which are the Timer1 Input Clock Pre-scale bits (TCKPS<1:0>), we set it to 112 which chooses our desired 1:256 ratio. Writing 0x0032 to T1CON also sets bit 1 which is the Timer1 Clock Source Select bit (TCS) which allows Timer1 clock to be selected by bits 9-8 of T1CON which are the Timer1 Extended Clock select Bits (T1ECS<1:0>), we will set them to 00 which sets Timer1 to use the Secondary Oscillator (SOSC) as the clock source. The function will then clear the Timer1 counter and interrupt flag; it will then also re-enable the Timer1 interrupt. It sets up the count by multiplying the clock speed by the number of seconds passed as an argument and storing it in an unsigned variable of type long, it will then divide count variable by a factor of 256 to account for the clock ratio (i.e. shift 8 to the right) and use it to set the Timer1 period. Once the period has been set and Timer1 re-enabled a while loop can be used to check to see if the timer is done and the exit flag has been received, if conditions are not met the microcontroller stays in sleep mode. Once the conditions are met the microcontroller will exit sleep mode and disable the Timer1 interrupt and then disable Timer1.

The crystal oscillator will be connected in parallel with a capacitor between both it’s input and output, which will be connected to pin numbers 33 and 34 respectively. The secondary oscillator circuit is depicted below in figure 4.1.4.1-7.



*Figure 4.1.4.1-7 Microcontroller and Secondary oscillator (SOSC) circuit*

*LED Circuit –* Two LEDs will be added to the circuit to serve as indicators, one green and one red. When ever the microcontroller is in normal operating mode the green LED will be lit, if an error state occurs the red LED will be turned on. The LED will also be used to indicate successful completion of routines where either LED can be used by turning them on in preset patterns and durations. The LED circuit is simple; the LEDs are connected to pins 25 and 26 where red is connected to pin 25 and green to 26. There is a current limiting resistor in series with the LED in between the microcontroller and the LED. For either LED to turn on all that is required is to set pins 25 or 26 to high. Below the LED circuit is depicted in figure 4.1.4.1-8.



*Figure 4.1.4.1-8 Microcontroller and LED circuit*

**Microcontroller Summary –** This section outlines all the microcontroller connections and pin assignments for a quick and easy to access information.Below in table [fill in the blank] all the microcontroller pin assignments and operating modes are outlined. In figure [fill in the blank] the complete PCB circuit schematic is shown.

|  |  |
| --- | --- |
| **PIC24FXXKA304** | |
| **PIN** | **PIN Connection and Feature usage** |
| 1 | SDA1 connected to Magnetometer Sensor / SDA1 |
| 2 | U1RX connected to UART header / U1RX |
| 3 | U1TX connected to UART header / U1TX |
| 4 | connected to Transceiver / RC8 |
| 5 | connected to EEPROM / RC9 |
| 6 | NC |
| 7 | INT connected to Transceiver / RA6 |
| 8 | SDI1 connected to Transceiver/EEPROM / SDI1 |
| 9 | SCK1 connected to Transceiver/EEPROM / SCK1 |
| 10 | Connected to GND |
| 11 | SDO1 connected to Transceiver/EEPROM / SDO1 |
| 12 | RADIO\_WAKE connected to Transceiver / RA10 |
| 13 | RADIO\_RST connected to Transceiver / RA11 |
| 14 | INT1 connected to momentary switch / INT1 |
| 15 | MEC\_MEAS connected to PM circuit / AN9 |
| 16 | VSS connected to GND / VSS |
| 17 | Vreg connected circuit / VDD |
| 18 | connected to push button and Vreg / |
| 19 | NC |
| 20 | NC |
| 21 | PGD connected to program header / PGED1 |
| 22 | PGC connected to program header / PGEC1 |
| 23 | NC |
| 24 | NC |
| 25 | LED1 connected to LED circuit / RC0 |
| 26 | LED2 connected to LED circuit / RC1 |
| 27 | MEC\_MEAS\_CTL connected to PM circuit / RC2 |
| 28 | Vreg connected to de-coupling capacitor / VDD |
| 29 | VSS conncted to GND / VSS |
| 30 | NC |
| 31 | NC |
| 32 | NC |
| 33 | Secondary Oscillator connected to 32kHz oscillator input / SOSCI |
| 34 | Secondary Oscillator connected to 32kHz oscillator output / SOSCO |
| 35 | NC |
| 36 | NC |
| 37 | NC |
| 38 | NC |
| 39 | VSS connected to GND / VSS |
| 40 | Vreg connected to GND / VDD |
| 41 | AE\_CTL connected to PM circuit / RB5 |
| 42 | LCE\_CTL connected to PM circuit / RB6 |
| 43 | INTO connected to momentary switch / INT0 |
| 44 | SCL1 connected to Magnetometer sensor / SCL1 |

*Table 4.1.4.1-1 Microcontroller Pin Assignments and Features*

**4.1.5 Occupancy Sensor Node Firmware**

The firmware for the occupancy sensor node will be written in the C programming language. The firmware has to address two main challenges. The first is to ensure that the occupancy data collected is up to date, meaning that the sensor is sampled at a high enough rate to be able to detect a change of state event as close to the actual event taking place as possible (i.e. a car parks at t = 0s, the sensor node detects a parked car at t = 15s). The second is to ensure that the sensing routine minimizes the power consumption to as low a value as possible. To overcome these two challenges a routine has been developed to sample the sensor at a high enough rate while still consuming the least amount of power. In the following sections we will go over the contents of the data packet, the supporting function required to execute the routine and the logic flow of the main function.

***4.1.5.1 Data packet***

The occupancy sensor node will be gathering, processing, and transmitting data almost constantly. Each data packet transmitted will help build the database on the server which will not only contain the available parking spots, but it will also contain information about the power supply as well. The data packet will contain 2 bytes for the voltage reading collected from the battery, one byte that will indicate the current state of the parking spot being monitored, and an integer that will contain the rate at which the sensor is being sampled. The rest of the data packet will contain the information required to complete a successful transmission using the MiWi protocol (further information on the functions of the MiWi protocol can be found in section 4.1.3).

With the complete set of data gathered from all of the sensor nodes that make up the garage’s wireless sensor network, the administrator of the Parkit system will be able to determine which floors are experiencing the most traffic and which nodes are consuming the most power. With this information algorithms can be made to optimize the power efficiency of the wireless sensor networks.

***4.1.5.2 Functions***

In the final firmware design there will be many functions that will handle the backend communication protocols, component processing and activation, and data gathering, just to name a few. Instead of focusing on every miniscule function in this section, we will discuss the major ones which control the overall logic of the firmware. The other functions are discussed in greater detail in the other design sections for the sensor node. The sensor node routine will have two major functions. The functions we will discuss perform the following tasks:

* Detects the magnetic field reading of a parking space
* Handles a state change event
* Prepares the data packet to be transferred
* Establishes a wireless connection with the sub gate and transmits the data to it

In the following sub sections, we will take a look at the function that acquires the raw data from the magnetometer and detects event changes. We will also review the main function and look at its logic flow.

***4.1.5.2.1 Raw Data Acquisition and Event Detection***

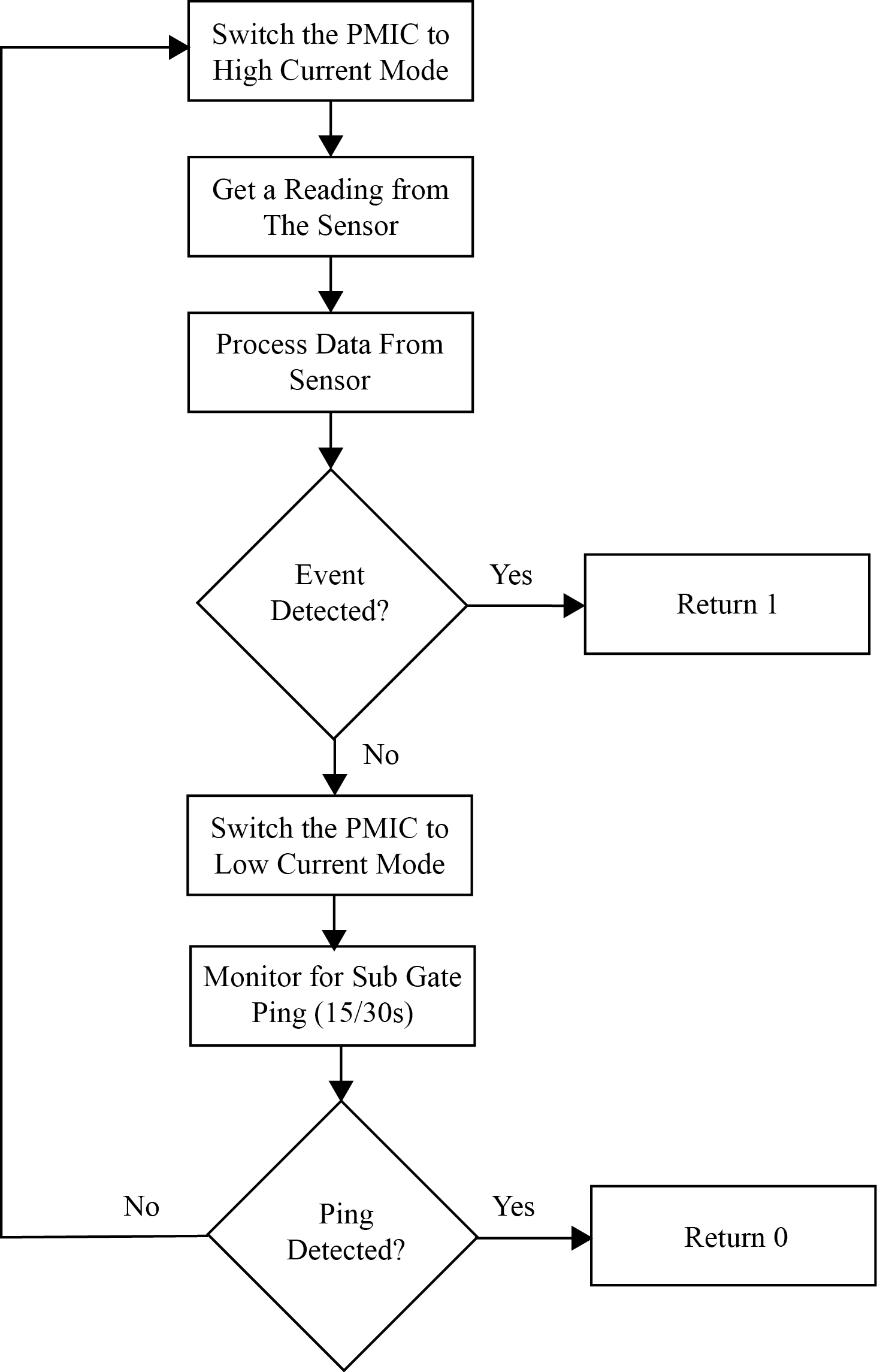
This function will get data readings from the magnetometer and return an integer when it has processed an event change or when it has received a ping from the sub gate. The prototype for this function will be as follows:

**int get\_data( int rate)**

The function’s first job is to change the power mode of the PMIC from low-current mode to high current mode. Once there is enough power to take measurements from the sensor, the function will retrieve a sample reading. The HMC5883L sensor draws 100µA while measuring the magnetic field. The sensor has two modes for measurements, single measurement and continuous measurement. The MEC207-7S battery could sustain seven hours of continuous measurement, which will not be sufficient for our application. In single measurement mode, the magnetic field can be sampled in 6ms (including startup time from idle mode). By sampling the magnetic field every 15s, the total sampling time per day is reduced to 34.56s. The amount of energy required to supply the measurements is 3.456mAs, which is only a fraction of the total energy capacity of the battery.

The integer parameter in the function prototype passes the rate at which to sample the data. Although sampling the data at 15s will maintain the power consumption to a manageable level, the rate is reduced to 1 measurement every 30s at night to compensate for the power consumption on the sub gate (for further details go to section 4.2). Also, the rate variable can vary from floor to floor or even from node to node depending on the location of the node and its traffic history. For example if the first floor experiences a high level of traffic, its rate variable may be set to a higher frequency to render more accurate readings for the motorists looking for available spots at the cost of a higher power consumption. On the other hand, the fourth floor may experience a very low level of traffic, which would make even a standard 15s sampling rate wasteful on energy. To make the energy more efficient the sampling rate can be lowered to 30s or less.

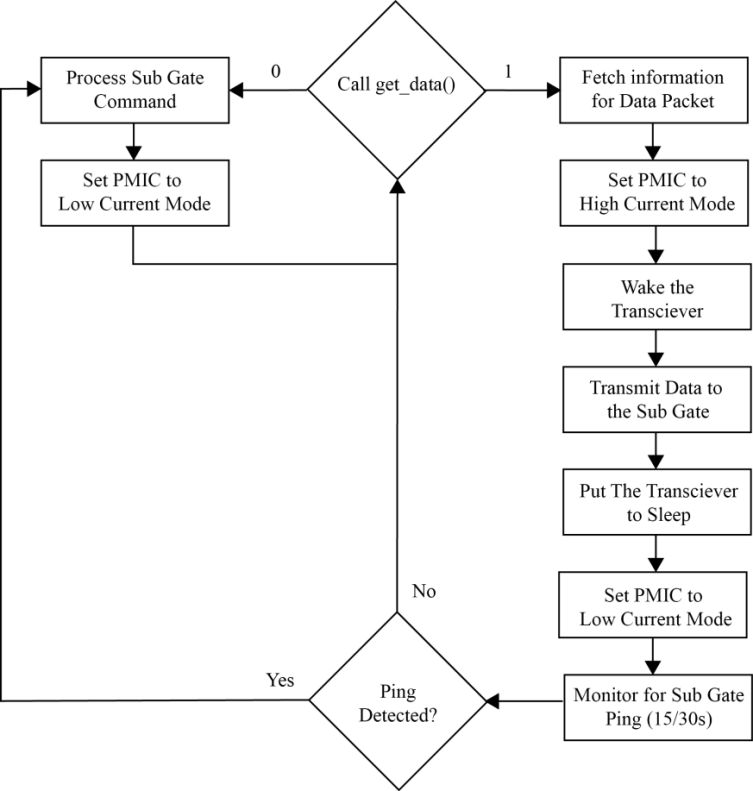
Once the sensor has been sampled for a reading, the function will process the sampled data. The HMC5883L provides readings for the x, y, and z axis. The function will calculate the magnitude of the reading and store it in a variable. This value is compared to the previous reading, if there is a significant change in the two readings then an event is triggered and the function returns an integer value of one. If there is not a significant difference between the two readings, then the function signals the PMIC to enter low-current mode. It will then monitor to see if the sub gate is trying to ping the sensor node for the amount of time passed by the rate parameter. If the function receives a ping from the sub gate it will return an integer value of 0. The function will loop until it detects an event change or it receives a ping from the sub gate. Figure 4.1.5.1.1-1 contains a flow chart for this function.



*Figure 4.1.5.2.1-1 Data Function Flow Chart*

***4.1.5.2.2 Main Function***

The main function will handle the overall flow of the firmware. Figure 4.1.5.1.2 contains a flow chart for the main function. The function will start off by calling the get\_data() function and setting a condition for the returned value. If a 0 is returned then the function will process the incoming command (change rate variable, transmit current state, reset node), set the PMIC to low-current mode and loop back to the call for the get\_data() function.



*Figure 4.1.5.1.2-1 Main Function Flow Chart*

If the get\_data() function returns a 1, then the function begins the process of transmitting an event change. The process begins by changing the state of the parking spot. There will be a variable that holds either a 1 or 0. If the variable is a 1 it indicates that the parking spot being monitored is occupied, if it is 0 it indicates that the parking spot is available. When get\_data() returns a 1 it means that the parking spot has switched states either from occupied to available or from available to occupied. So when the get\_data() function returns a 1 the variable changes from its current state (0 turns to 1 or 1 turns to 0).

After changing the state of the parking spot the function continues gathering the required information for the data packet by signaling the PMIC to connect the voltage sensor to the battery circuit and to disconnect the solar panel from the charging circuit. Once this is done an accurate reading can be sampled from the battery and stored in the data packet. The function then sends another signal to the PMIC to disconnect the voltage sensor from the battery circuit and to reconnect the solar panel to the charging circuit. The function will then gather the rest of the contents of the data packet, including the node address (which is stored in the EEPROM), the coordinator address, sampling rate, etc…

Once all the data is gathered the function sends a signal to the PMIC to enter into high-current mode and then initiates the appropriate MiWi protocols to transmit the data to the sub gate. After transmitting the data the function monitors for a ping from the sub gate for the amount of time saved in the sampling rate variable. If no pings are detected, then the function loops back and calls the get\_data() function again. If a ping is detected, then the function loops back to the routine to process the incoming command and then loops back to the beginning of the function.

***4.1.5.3 Counter Node Firmware***

The counter node is responsible for keeping track of how many vehicles are in the garage. It also counts the number of cars entering and exiting each floor. This is important data. With it, the user interface will be able to give the end user an idea of what his or her chances are for acquiring a given parking spot. For instance, if there are 20 parking spots available in a garage, but there are 30 cars that have entered the garage (presumably looking for a parking spot), then the chances of acquiring the parking spot is greatly reduced. On the other hand, if there are 20 parking spots and only 10 cars have entered the garage, then chances of acquiring the parking spot is very high.

To keep the accuracy of the data high, the counter nodes will be sampled once every second. If the sensor was sampled at a longer rate, it risks the chance of missing a vehicle entering or exiting a floor. This is the only difference between the occupancy sensor nodes and the counter occupancy nodes. The occupancy counter nodes will sample the sensor reading once every 15 seconds or longer. Its sampling rate will vary, depending on the time of day and the traffic history of the spot it is monitoring. The counter node will have a fixed sampling rate of reading every 1s.

Since the sampling rate will be controlled by the server, the firmware of the counter node will be exactly the same as the firmware for the occupancy node. This means that any node in the garage can potentially serve as counter or an occupancy node. The location of the node will ultimately determine its function. For more information on the firmware, please read section 4.1.5.2.

**4.2 SUB GATE**

The sub gate serves as a coordinator node in our MiWi network. All the sensor nodes in a floor will be transmitting their data to the sub gate. The sub gate will then relay that data to the gateway. Although the function of the sub gate is different than that of the sensor node, the hardware components are basically the same. The only difference between the two circuits is that the sensor node has the magnetometer and the sub gate does not. Our group considered at one point to include a sensor in the sub gate to serve as a reference node. Since the magnetic field of the earth drifts, we thought that it would be necessary to have a sensor take measurements in a location where there are no cars to serve as a reference to compare all the other readings to. After researching the sensor we discovered that we could just compare two sample readings to each other and determine if there has been a state change. This eliminated the need of a reference node in the sub gate. Other than the sensor the circuitry for the sub gate is identical to the sensor node (A full discussion on the sub gate/sensor node circuitry can be found in sections 4.1.2-4.1.4). The only real difference between the two devices is the firmware. In the following sub section, we will discuss the firmware for the sub gate in more detail.

***4.2.1 Bill of Materials for Sub Gate***

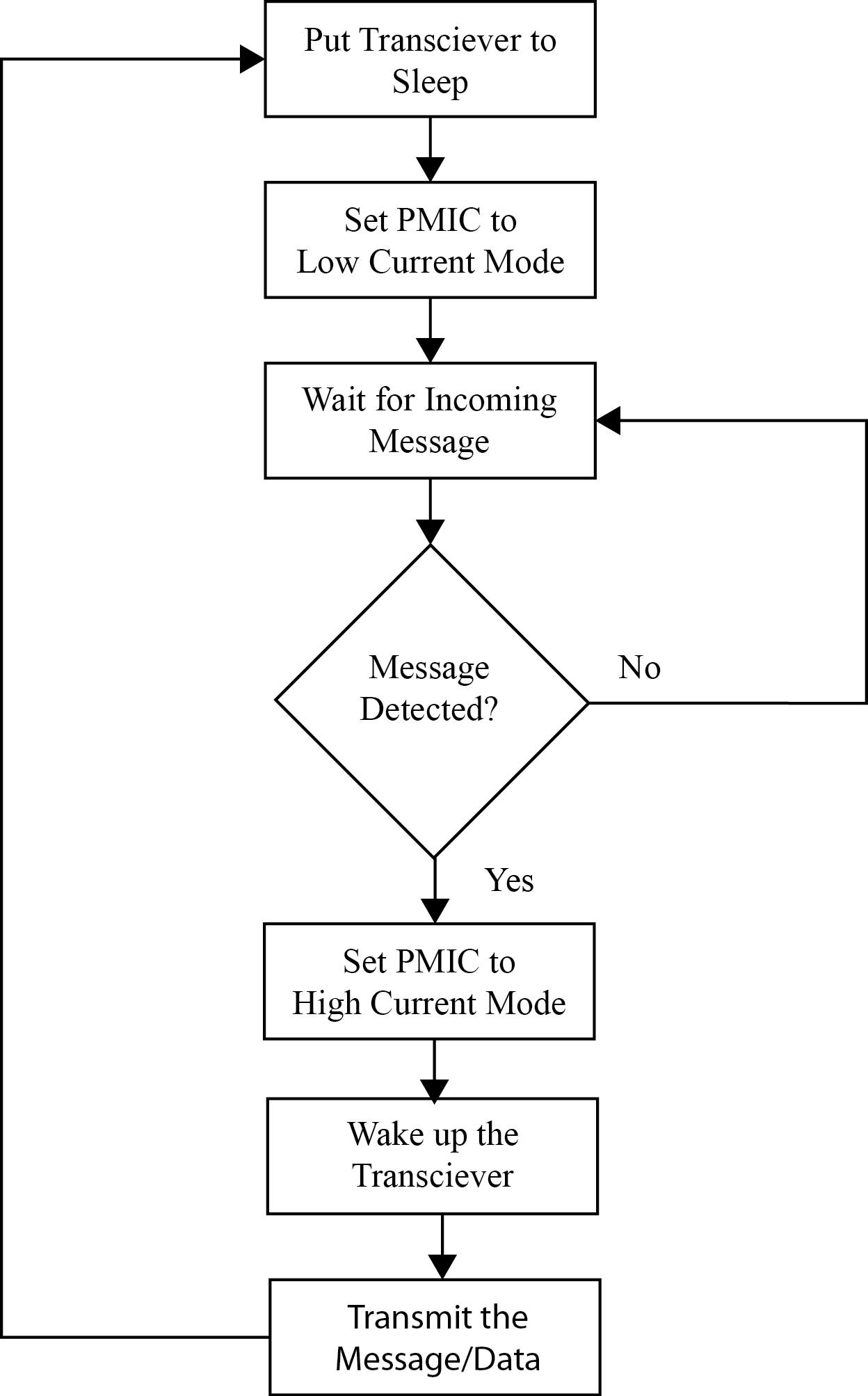
This section contains the bill of materials for the sub gate circuitry in Table 4.3-1.

|  |  |  |
| --- | --- | --- |
| Description | Item/Value | Qty |
| Microcontroller | PIC24F32KA304-I/PT | 1 |
| Transceiver | MRF24J40MA | 1 |
| IC | MAX17710 | 1 |
| Micro Energy Cell | MEC201-7S | 1 |
| Photovoltaic Cell | AM-1801 | 2 |
| Resistor | 5Ω  20Ω  470Ω  1kΩ  2kΩ  4.7kµ  10kΩ  100kΩ  1MΩ  3.01MΩ  4.7MΩ  10MΩ | 1  1  6  5  1  2  3  2  6  1  1  11 |
| Capacitor | 12pF  .01µF  .1µF  .7µF  10µF  100µF | 2  2  11  1  2  5 |
| Complimentary Paired  MOSFETs | DMG1016UDW  FDG6322C | 2  5 |
| 600 nA, Rail-to-Rail Input/Output Operational Amplifier, 5-Pin SOT-23, Extended Temperature, | MCP6041T-E/OT | 1 |
| Schottky Barrier Diode | BAS70KFILM (NP) | 1 |
| Test Point | Connector-1249 | 2 |
| Push Button | TL1015BF160QG | 4 |
| Slide Switch | AYZ0202 | 2 |
| Headers | 10-pin, right angle  6-pin, right angle  2-pin, right angle | 1  3  1 |
| Inductor | Ferrite Bead | 1 |
| LED | Typical Infrared GaAs | 3 |

*Table 4.3-1 Bill of Materials for the Sub Gate*

**4.2.2 Sub Gate Firmware**

The sub gate has a very simple function. All it does is relay messages. It will either receive a command from the gateway or transmit it to the sensor nodes, or it will receive a data packet from the sensor node and transmit it to the gateway. Most of its functions will part of the MiWi function library to initiate and execute data transmissions (for further information on the MiWi functions, refer to section 4.1.3). Figure 4.2.1-1 contains a flow chart of the main function for the sub gate.



*Figure 4.2.1-1 Sub Gate Firmware Flow Chart*

**4.3 GATEWAY**

The gateway is going to be the direct link between the wireless sensor network and the server. In a real life application, the gateway will require additional hardware to connect to a local area network where the data can then be transferred to the server via the internet. To support the additional hardware the gateway would be hardwired to the power grid of the garage. For our project we will be bypassing the local area network connection by connecting the gateway to the server via a wireless network adapter with a USB connection. The adapter our group selected is the ZENA wireless adapter 2.4GHz MRF24J40. In the following sub section we will take a closer look at the wireless adapter.

**4.3.1 ZENA Wireless Adapter**

The ZENA Wireless Adapter 2.4GHz MRF24J40 is a USB device that connects to a laptop or desktop and provides diagnostic support and protocol analyzing for MiWi networks. It can also be used to connect the host device to the wireless sensor network. The transceiver in the adapter is the same transceiver being used by the sensor nodes and gateway.

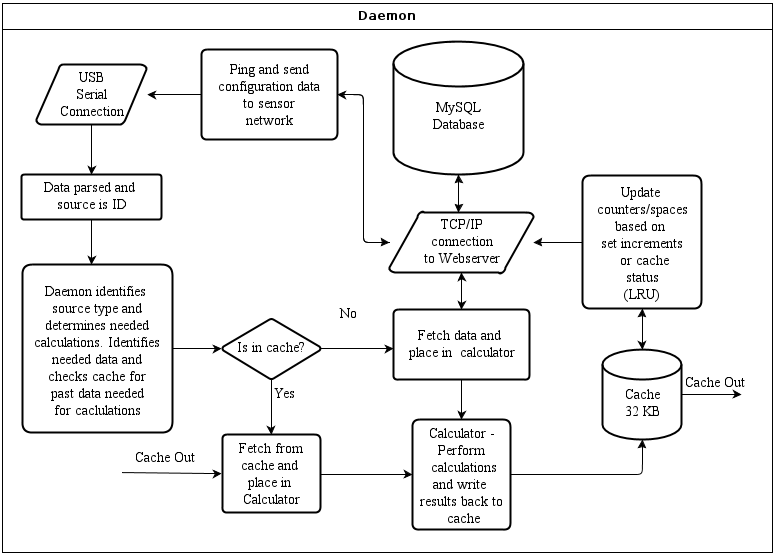
The adapter comes with a driver and a software program (Wireless Development Studio) by which functions can be accessed to sniff the MiWi network for data packets or it can turn the host device into a node in the wireless network. The latter is what we propose to use the device for. We will connect the ZENA to a laptop which will be gathering the data from the wireless network and loading it to a program within the laptop. Though the laptop will serve as a node in the wireless network, it will also serve as the server where the database will be populated and organized. The program that parses the data and populates the MySQL database is discussed in detail in section 4.4.

**4.4 SERVER**

The server will perform three main functions; it will take in serial data from the sensor network and process it, it will serve as a medium to store data using a MySQL database and it will access that data using an array of web technologies to allow HTTP access to an end user. For the initial design a modular approach is going to be taken as to allow for easy development amongst the team members and in general. The daemon that will be used to accept the serial data via the Zena USB wireless module will be developed natively using C/C++ and then complied according the host operating system. The daemon will connect to the MySQL database remotely to read or write data, this design will most likely have both the daemon and webserver on the same machine however the design will allow for scalability being that a gateway to the sensor network need not be on the same machine as the server. The webserver and MySQL database will be virtualized using VirtualBox, section [fill in the blank] below describes this implementation in further depth. There are multiple gateways can be designed based on logistics and other factors for different operating environments, for this project the Zena wireless module will be used to emulate the gateway and will serve as a proof of concept and will be connected directly to the server via USB.

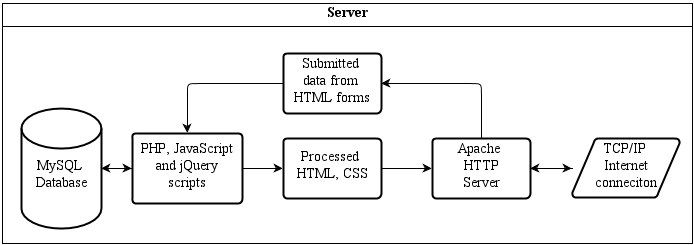
*Daemon –* The daemon will run as a background process that constantly monitors and communicates via a serial connection to the wireless sensor network. The daemon will be written in C/C++ using various tools and compiled directly in the host operating systems. It will connect to the MySQL server using the TCP/IP protocol using default procedures outlined in the MySQL documentation. It will acquire its data to send to the databases via a serial connection to the Zena wireless module. The Zena will communicate with our application using the MPLabComm drivers provided by Microchip. The data will need to be parsed to identify the source of the transmission and the data being sent. Once the ID of the source and data is read the daemon will check if the source is a counter or a standard detector, based on this information it will identify which calculations need to be performed on the data. It will check if the data is the cache that is updated using the least recently used (LRU) cache scheme, if the data is found it will be sent to the calculation functions and if it is not then it will be first be retrieved from the MySQL database and sent to appropriate calculation functions.

The calculation functions will increment and decrement the total amount of cars per floor and per garage based on the counter data, total number of spaces and occupied spaces. An algorithm will be created to calculate the percentage of vacant parking spaces per floor and per garage based on the collected data. That is it will consider the amount of available spaces compared to the amount of free roaming vehicles per floor/garage when creating a percentage, it will also consider whether or not the vehicles are simply entering and exiting a floor/garage or looking for parking. Due to this slight delay in processing parking data by space will be available almost instantaneously while percentages will be updated with a slight delay, since for example a percentage change on Floor A will effect the percentage change in the entire Garage A and as expected the same holds true for the entire Location. Below in figure 4.4-1 a flowchart for the daemon is depicted.



*Figure 4.4-1 Daemon Flowchart*

*Operating System & Software –* As mentioned above the web server and MySQL database will be run in a virtual machine. This is mainly to allow the developed environment the option of portability amongst several operating systems and environments during development and prototyping. We will use Ubuntu Server 12.04 LTS in the virtualized environment; the networked device will be bridged as to allow the virtual machine full network access. To serve HTTP request Apache 2.4.2 will be installed and configured to operate on a local network for prototyping and proof of concept purposes meant for the scope of this project. To store the data MySQL 5.5 will be installed and configured to communicate to the Apache webserver through a server side scripting languages locally and communicate with the daemon via a TCP/IP connection. The server side scripting languages that will be used are PHP and JavaScript, these will serve to update and generate HTML and CSS to be displayed to the end user. All the software listed above will be installed and configured using standard settings found in documentation cited in section [fill in the blank] of Appendix [fill in the blank]. Below is a flowchart of the server process.



*Figure 4.4.2**Server Flowchart*

*MySQL Database –* The database will serve to store and organize all the data collected from the sensor network and any data needed for the user interface. A database will be created named ParkIt in which tables will be created for each type of data entry. For storing and organizing the sensor network data The Adjacency List model will be used to organize and store the sensor network data, this is mainly due to the fact that for every location the categories will be the same (i.e. Locations > Garages/Lots > Floors/Rows, Spaces). Knowing that at most there will be three levels makes the only limitation of the Adjacency Model irrelevant. An example MySQL table for our application will be built using the following model.

CREATE TABLE nodes(

node\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(20) NOT NULL,

status INT DEFAULT NULL,

parent INT DEFAULT NULL );

*Figure 4.4-3**Prototype MySQL create command*

Looking at the previous code it is shown a table is created for all sensor data named Node; this table will have the fields for the name of the node (i.e. UCF if location. Garage UCF-X if garage, Floor UCFX-Y if floor and Space UCFXY-Z if space), it will have the node status which will be either a Boolean true or false for the spaces represented by a 1 or 0 or a percentage is the node represents any element above the bottom space tier. Parent will represent which level the node resides on, the following example MySQL query demonstrates the hierarchy.

Using the model below it is possible to query database specifically based on any of the criteria (i.e. only print out data where the garage is equal to UCF-B) , this will be exploited to dynamically generate content throughout the user interface. Once the data is stored into the database it is ready to be accessed via PHP or JavaScript to be included into the HTML and displayed to an end user.

SELECT \* FROM parkit ORDER BY parkit\_id;

+-------------+----------------------+--------+

| category\_id | name | parent |

+-------------+----------------------+--------+

| 1 | UCF | NULL | //LOCATION

| 2 | UCF-A | 1 | //GARAGES

| 3 | UCF-B | 1 |

| 4 | UCF-N | 1 |

| 5 | UCF-A-1 | 2 | //FLOORS

| 6 | UCF-B-1 | 3 |

| 7 | UCF-N-1 | 4 |

| 8 | UCF-A-1-101 | 5 | //SPACES

| 9 | UCF-A-2-101 | 6 |

| 10 | UCF-A-3-101 | 7 |

+-------------+----------------------+--------+

10 rows in set (0.00 sec)

*Figure 4.4-4**Example MySQL query results*

*Apache HTTP Server –* We will use a standard install of apache and access the application directly using the servers IP address. In more developed versions of this project the servers IP would be binded with a registered domain name and other security and standard routines would be performed in the process of setting up the HTTP server. But for this proof of concept a standard install will be utilized to simply server the HTTP request with no extra considerations.

*Server Side Scripting –* As explained both PHP and JavaScript will be used to dynamically generate content to be displayed to the end user, it will also perform minor calculations. Due to performance benchmarks of PHP and JavaScript most of the heavy lifting as far as computing is concerned will be handled by the daemon, the server side scripting will be minimized as much as possible to improve performance and limited to basic calculations that revolve around displaying data. For example the type of calculations performed using the server side scripting languages would involve generating the refresh rates at which the data is attained from the database and displayed to the end user. Any and all calculations involving the network data will be processed by the daemon and placed in the database where the server side languages can then extract it and display it accordingly. Standard compliant HTML and CSS will be generated using default PHP and JavaScript functions and classes. PHP will serve as the framework and JavaScript will mostly be used for aesthetic purposes. The PHP will have two main operating layers; the first of which will solely deal with reading and writing data to the MySQL database and will be accesses via an include, the second layer would be throughout the html dynamically generating content into placeholders.

**4.5 USER INTERFACE**

The user interface will be designed with user friendliness in mind. Utilizing our system, users are going to choose a parking space with the provided information from our designed web application. There are some features that we are going to consider to make our design somewhat different than other applications. Some of these features are “Share Space with Friends” and “Where Did I Park?” We are going to explain these features in detail and how they are going to be incorporated into our design..

*Storage –* All content will be stored into the MySQL database, in the root of our directory that will hold the site source files there will only exist templates for displaying content. The content itself will be dynamically generated and retrieved fro the MySQL database when needed. Standard MySQL query commands will be used for this purpose using PHP and the content will be stored into variables that will be printed via the page templates that use HTML and CSS to display the content to the user.

This project will use jQuery Mobile as a scripting layer that will handle the classes used for our design. The JQuery Mobile set of plug-ins will provide an API for creating a mobile web application supported by multiple platforms. Our code will consists of four types of files: a JavaScript, CSS, HTML and graphic files. Basically using jQuery Mobile will we give the user a standard layout, user interface widgets, color scheme and over workflow that will be uniform across all platforms.

Standard jQuery libraries will be used to accomplish multiple tasks such as creating widgets, applying event listeners, and creating an API. This will be used in unison with Cascading Style Sheets to create a layout and the graphical view of our pages in a consistent easy to use fashion.

The data-attributes library of jQuery Mobile will help us to transform our simple markup into a complex user interface with interactive elements. The figure below shows basic HTML5 code with jQuery and jQuery Mobile libraries. This code is a prototype of how the code is to be inserted into the head of our HTML document.

<head>

<title>Example head with included jQuery Links</title>

<link rel="stylesheet" href="link to jQuery Mobile standard css"/>

<link rel="stylesheet" href="link to our custom css" />

<style>

</style>

<script src="link to jQuery library">

</script>

<script src="link to jQuery Mobile library">

</script>

<script src="link to our custom javascript">

</script>

</head>

*Figure 4.5-1**HTML head prototype*

We are going to display all relevant content thorough our application. All of these codes will be in the body section using basic HTML; the document will be of type PHP to allow PHP tags to dynamically generate content. The first step is to create page templates based on the type of data being displayed (i.e. location, garage or floor) and templates for elements to be used throughout the design (i.e. navbar, listviews and forms). Every page will be unique in the sense of the data it will display however using template files we will have a manner to scale a site without having to add individual pages (HTML/PHP files) to our design.



*Figure 4.5-2 Mobile Application*

As shown in figure 4.4-1 our home page consists of a header and image for branding purposes, a list-view of available locations and a navigation bar at the bottom of the page. The styles will be applied by adding the jQuery mobile *data-role* attribute to the HTML tags. It is a simple layout as to allow easy and efficient identification of destination.

A header and image are self-explanatory. In this page *data-role=”listview”* is applied to the list. The list-view elements have no interactive properties. The design is meant to be intuitive and go towards the users tendency to associate words with images and shapes, most data is also presented in the form of an interactive element like link buttons for example. The locations labeled UCF and Valencia are used to depict the way locations will be listed to the user. Each one of these list-views is going to be generated recursively based on data from the database and periodically updating the vacancy percentage of each location based on a configurable refresh rate. The list-views in jQuery Mobile are nested, in other words the user can drill down from the top-level list to display the sub-list by following the hierarchy structure until a space is found. The last element found on this page and on every page is the main navigation bar, which uses the *tab* data-role in jQuery. Below are prototypes for the data-role classes needed for jQuery to

<!—the header data-role is a Div element -->

<div data-role="header" data-position="fixed" data-theme="b">

<!—the listview data-role is a ul element -->

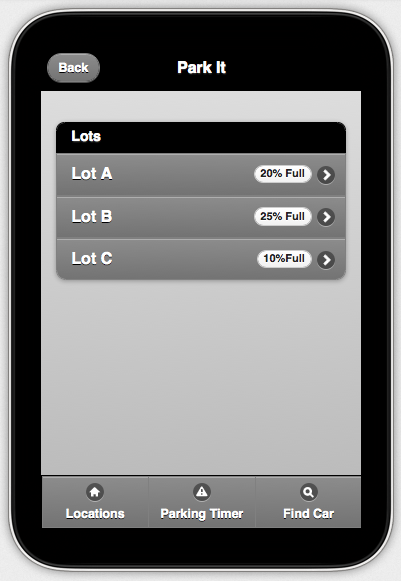
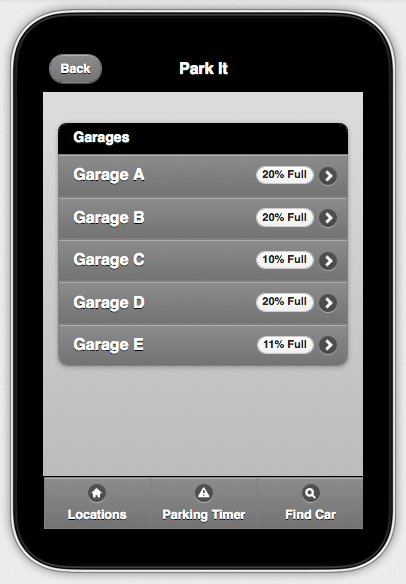
<ul data-role="listview" data-inset="true" data-divider-theme="b">

<!—the tabbar (i.e. Main Nav bar) data-role is a Div element -->

<div data-role="tabbar" data-theme="b" data-iconpos="top">

*Figure 4.5-3**HTML head prototype*

When a location is chosen the user will present with a list of available garages or lots. Each garage list will also be generated dynamically by using PHP to pull data from the database. To stay consistent with a users urge to look for a back button at the top left corner of the UI will add the back button to the top left corner of each page other then the home page. As shown in the figures below, every garage will dynamically get the vacancy data from the database and provide that information to the user every 15 to 30 seconds and is presented in a list-view, the idea is to make the user familiar with each interface. The items in each list will be generated using recursive methods as to allow for clean and manageable code. Below are concepts of what the Garage/Lot interfaces will look like.



*Figure 4.4-2 Garage Concept Figure 4.4-3 Lot Concept*

Next in the workflow the user will be presented will available Floors or Lanes. As we mentioned before some of the functions of these pages are going to be redundant. The functionality of floor page and garage page are exactly the same as to create familiarity and ease of use. The idea is that the user will eventually be able to find a parking spot fast and efficiently based on the visual data presented to them.

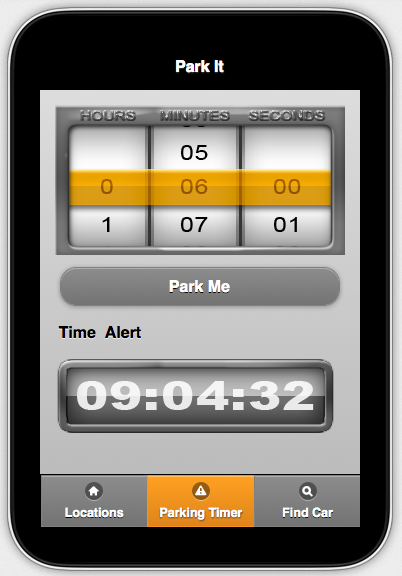
More information is given in the Server design section regarding how all the information is gathered for the graphical user interface.

The user will follow a similar workflow all the way down until he/she is presented with a 2D representation of the floor/lot with labeled spaces they can choose from. The user will pick the available parking space that they want to use and press *Park Now* button. By pressing the Park Now Button, the system will log which parking space the user has parked at and initiate the timers. In addition to the timer that counts up a timer can be configured to time down where a metered parking environment is being used.

|  |  |
| --- | --- |
|  |  |
| *Figure 4.4-4 Floors for every Garage* | *Figure 4.4-5 Available Spaces* |

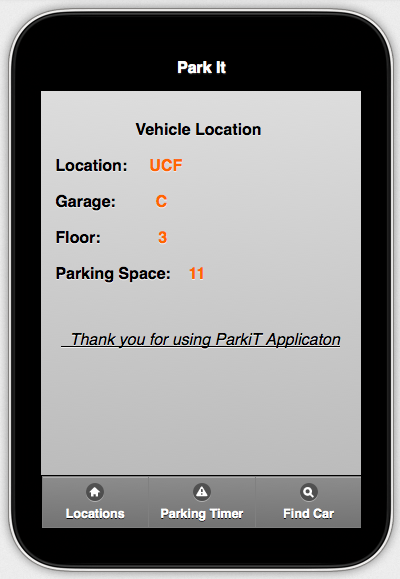
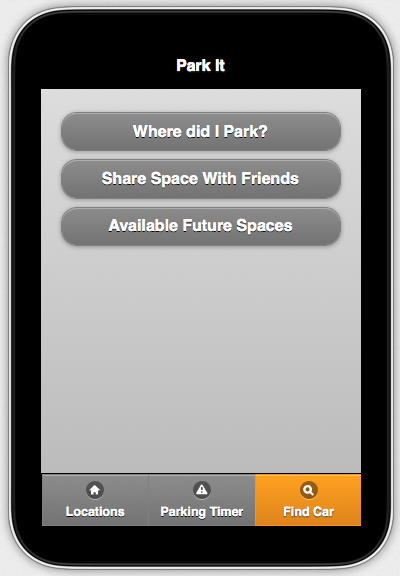
The design of this page is differs from the other pages. The challenging part implementation of the design is creating the different layouts for the floors/lanes as garages can vary greatly in one location alone. Above in Figure 4.4-5 shows a conceptual design layout of a garage floor. This page will contain jQuery form elements as to allow the user to his/her parameters. Nested lists will be used to represent all the parking spots in the code and using JavaScript we will create the visual display.

As explained before once the user presses *Park Now*, he/she will be directed to the *Parking Timer* page. The user will also have the option to put in the amount of time he/she is planning to park their vehicle and they press *Park Me, which* initiates the timers. If the time set is about to expire the *Time Alert* will pop up a message to alert the user. This function is better suited for metered environments or in applications where a user in on a schedule and needs to leave at a set time. This feature makes our application more global so it will have room for future improvements tailored for different operating environments. There ill also be an aspect of the application that shows estimated available spaces based on peer inputted departure times. This section of the data is just informational and does not interfere with sensing logic and algorithms, as the estimate is going to be based on user inputted data. If the sensor detects that there is no car in the parking space, then the timer will reset itself.



*Figure 4.4-6 Parking Timer*

The last feature in the application is found under the label *Find Car*. This will redirect the user to the *Find Car* page where the user is presented with a few options. Amongst the options the user will have the ability to show the entered information regarding where he/she parked. Below are conceptual depictions of the *Find Car* and *Where did I park?* pages.



*Figure 4.4-7 Vehicle Location*

The second feature is the “*Share Space With Friends”.* This section will let the user to share their parking space information with their friends. The timer and space information is relayed to all friends in the app.

These features are going to make our design unique in terms of web application design. There are some more upgrades that can be done that we will talk about more in the future section.

Now that we have all the pages, we can put all of these pages together to build an actual mobile application with jQuery Mobile. Once we generate the pages we can use *juery.ajax ()* method to keep them updated with minimum strain on the server and client alike. This method basically makes a cross browser AJAX implementation. Using jQuery functions makes our job easier to build an easy to use interactive and intuitive mobile application, using PHP allows for server side scripting to dynamically generate all the content.

**5. TESTING PROCEDURES**

I group is very confident in the design of wireless sensor network, server, and user interface. Each component was carefully chosen and thoroughly refined. Although we believe our project will be able to function as desired, we still have planned several testing procedures to ensure that each component is operating within the specifications set forth by our design. In the following sub sections we will discuss the testing procedures for the sensor, power management system, and the communication system.

**5.1 SENSOR NODE TESTING**

As part of its features, the sensor has a self-test procedure which will be utilized to ensure the sensor is operating properly. This simplifies the test method since a test plan does not necessarily have to be developed for the sensor. The first step is to change the LSB of the configuration register A to 01. This biases the X and Y axes with a positive voltage and the Z axis with a negative voltage. Next, MR is placed into single measurement mode by changing the value of MR0 to 1 and MR1 to 0. Now data from each of the axes elements is acquired twice. The first data acquisition, a set pulse is given and immediately the external field is measured. In the second data acquisition, the offset straps are excited by a current of approximately 5.5 mA. The X and Y axes are biased in positive mode and the Z axis is biased in negative mode. The data from the first acquisition is subtracted from the data in the second acquisition and the result is placed in the data output register. With the configuration register B containing a value of 0x02, a value of approximately +655 ADC counts (0.64 Ga×1024 counts/Ga) is placed in the X and Y data output registers and a value of approximately -604 is placed in the Z data output register.

**5.2 POWER MANAGEMENT SYSTEM TESTING**

There are three components in the power management system that needs to be tested to ensure that the circuit is functioning properly. These three components are the voltage sensor, the solar panel, and the regulated output. The following are the procedures our group will take to test if each component is functioning properly.

**5.2.1 Voltage Sensor Test Procedures**

The voltage sensor circuit in the power management system will report the voltage value of the MEC201-7S battery. This data will then be used to monitor the health of the batteries for each sensor node in the wireless network. To determine if the value sampled by the voltage sensor is accurate, our group will follow these procedures:

* Turn on the sensor node
  + Slide the power control switch to “ON”
  + Push the regulated output button to supply power to the circuit
* Connect a volt meter to test point 1 and 2
* Send a command to the node to report the voltage
* Compare the value returned to the reading on the volt meter

If the values do not match, measure the voltage output of the unity gain buffer amplifier in the voltage sensor circuit. If the voltage output of the op amp matches the battery voltage, then the error is in the firmware code. Trace the code and determine the error. Repeat the test procedures until the correct values are being reported.

If the voltage output of the op amp does not match the battery voltage, then the op amp is faulty and must be replaced.

**5.2.2 Solar Panel Test Procedures**

The solar panels will be used to charge the battery cells of each sensor node. The solar panel that our team selected is the AM-1801. To determine if the solar panel can harvest enough power from the ambient lighting of a typical garage, our team will take a solar panel and sample the open circuit voltage and short circuit current at different times during the day to ensure that enough power can be generated to efficiently charge the battery cell. If the solar panel cannot generate power with the ambient light of the garage, the solar panel must be upgraded.

**5.2.3 Regulated Output Test Procedures**

The power management system has a voltage regulator that outputs 3.3V to the sensor node circuit. If this voltage is not outputting correctly the sensor won’t be able to make accurate measurements and the transceiver will not be able to communicate with the sub gate. To test the output regulator, a blue LED has been included in the circuit. If the PMIC is outputting the 3.3V the LED will light up, if not it will remain off.

**5.3 COMMUNICATION SYSTEM TESTING**

There are multiple communications testing procedure that we need to involve in our design. One is the communication in the sensor node, another one is the communication between the sensor nodes and sub gate and the other one is the communication between the sub gate to the gate way.

In the sensor node itself, we are going to test the communication between the microcontroller and the transceiver by programming the device so they can perform properly. This is the most important step for our design to be able to get the data from the sensor node. The data that is sampled from the sensor node has to be at a high enough rates to be able to detect a change of state event as close as the actual event so this will affect our power consumption. For this reason one of the tests for the sensor node will be making sure that the sleep function is working properly. The test will check to make sure that the transceiver will go to sleep mode and the microcontroller will be able to wake the transceiver up.

Another test is the communication between the sensor nodes and sub gate. There will be another test using MiWi protocol to make sure that all of the data packets initiate transmission to the right sub gate. In another word, making sure that sensor nodes in each floor communicate to the right sub gate in that floor. We are going to use the microchip software program for this test.

The last test that we will perform is the communication between the sub gate to the gate way. First we have to test a firmware for the sub gate to make sure the function is different than a sensor node. The gateway is going to be a direct link between the wireless sensor network and the server. We are going to test the ZENA adaptor software to make sure that the functions can be accessed to sniff the MiWi network for the data packets. We also can use the ZENA as a device into a node in the wireless network to test the sub gates.

Some other testing procedure that we will be doing is the distance between the nodes to check the communication range between the transceivers.

We already ordered the parts for our design, so we can start testing the device before next semester. The sooner we start testing, the smoother our project is going to be next semester.

**5.5 GRAPHICAL USER INTERFACE TESTING**

Our team is going to test the GUI using jQuery Mobile to design a test case to make sure that the application is easy to use for everyone. We are going to check the application with different group of people to try the application and then they are going to fill out the survey form to write their opinion and Ideas.

**6. BUILD, PROTOTYPE, AND EVALUATION PLAN**

*Build & Prototype:*

1. The first step in the build process will be acquisition of parts, before any PCB design is sent to the board house we will make sure are parts are 100% obtainable.
2. Once the parts have been acquired we will proceed to send our PCB designs to Advanced Circuits to have our PCBs made.
3. When the PCBs have been received we will use the equipment available at the UCF Radio club to mount all the parts.
4. All power circuits will be tested using test points.
5. Program Microcontroller with firmware.
6. Debug any issues using UART header.
7. Build and perform preliminary test on webserver and MySQL database (this will occur simultaneously with the previous steps for the most part).
8. Test microcontroller and sensor network communication.

*Evaluation Plan:*

1. Connect the network in a controlled environment that most closely resembles a real world scenario.
2. Tweak until events can be accurately detected and displayed via the UI.
3. Test in a real world scenario and log data to test for accuracy, effectiveness and specification requirements.
4. Make any adjustments if possible and re-deploy in a real world scenario.
5. Make a final evaluation of our design and implementation.

**7. Conclusion and Future Features**

The *Parkit* project has been very helpful for all of our team members. The team clearly understands the concept of the project and what each component is doing. Our group had several meeting a week during the semester to make sure all the ideas are shared between team members and everyone is in the same page. The team divided up the project into four sections of Sensor node, Power Management, Server, and User Interface. Everyone would participate on the first section with research and design. As soon as one section was done, then we all moved on to the next section. This method helped us learn more about the project as a team and challenge ourselves to learn new materials about our project.

One of our goals for this semester was to be a head of the schedule as much as possible. Therefore, we had already ordered some of the parts that are required for our design. Also, we are going to order our PCB boards before next semester. This way we won’t have any pressure waiting for the materials to be ready for our design. We have also contacted some manufactures to get some free samples for some of the components. The budget for the project is more than what we were aiming for at the start of the semester, however, with the energy harvesting system that we are designing our system will last for 20 to 30 years. This is one of the advantages that our design has compared to similar designs.

One of the most valuable features that our project offers is the availability to be expanded. There are lots of ideas and features that our group wanted to implement for this project but due to over complicating, time constraints and budget limitations we chose not to implement some of those ideas.

*GPS:*

One of those ideas is having a GPS device in our mobile application. GPS will give a user the direction to the empty parking space. This application will direct the user to the exact position of the parking space which ultimately saves lots of time and hassle.

*Voice Activated System:*

This system is the attachment to the GPS. Voice activated system will talk to the user asking for the destination. Then it will find the empty parking space and will direct the user to that exact final destination. This way there is no need to look at the device for direction while driving, Voice activated system will guide the user to the exact location.

*Lighting Indicator:*

Lighting indicator is another feature that we could add to our project. The way this lighting indicator works is that it is going to be green and red lights for every parking space to indicate vacancy of the parking space. Green light shows that the parking space is empty and the red light shows that the parking space is full. Lighting indicators are located in every lane of each floor and also for every parking space. This feature will help the users to find their parking spot without any problems.

*RF ID to monitor decal:*

Another feature that we are adding to the future features is RF ID to monitor decals for school or any places that use any kind of tags. Every decal consists of a paper thin coiled antenna of foil printed onto the back. These decals do not need any power supply and they get enough power form receiving a signal. This feature will help school identify the expired decals. Using this method, each decal will be specified to the person like their vehicle tag number. As soon as the decal is going to expire, it will notify the user and they can renew or purchase a new decal.