**PROJECT REPORT ON**

# SMART PARKING USING BLUETOOTH MESH

By

SIDDHANT JAJOO
SARTHAK JAIN
SWARUPA DE
ACHYUT DAVE

University of Colorado
Boulder

**UNDER THE GUIDANCE OF:**

PROFESSOR KEITH GRAHAM,
UNIVERSITY OF COLORADO BOULDER

DATED:12/12/2018.

# TABLE OF CONTENTS

# Overview

The Smart Car Parking System aims at simplifying the process of parking in basements. It is designed to notify any incoming drivers of empty parking slots throughout the area.
In case of multi-level parking area, each level can have one friend node displaying the empty slots for that particular level. This is implemented using a Bluetooth Mesh framework of a Low Power Node(LPN) for each parking slot and a Friend node which can support as many LPNs as the application requires. For demonstration purposes, one LPN is connected to one friend node here.

# Problems Addressed

a) The first problem that drivers face in huge parking lots is that they are unaware of the status of different parking slots i.e whether they are vacant or occupied. It becomes very difficult for a driver to keep circling around the parking slot to find a vacant spot for the car. Sometimes it can be tedious if very few parking slots are available. This results in wastage of time, energy and fuel.
b) The second problem that needs to be addressed in a parking lot is the wastage of electricity due to unwanted use of lights. The parking slots do not need lights to be switched on all the time. Lights are necessarily required when a person parks/approaches the car.
c) The third problem is that of fires in an enclosed area, which aren't easily detected.

# Solutions Proposed

Our project addresses each problem by:
a) Displaying the status of different parking slots or the number of parking slots vacant in a particular area at the point of entry of vehicles on an LCD. In this way, any driver on entry can see which slot is empty and directly move to that particular slot without the headache of searching for a vacancy. To accomplish this, we are using a proximity sensor at each slot to determine whether a car is parked or not, allowing us to determine vacancy of that slot.
b) Switching on the lights (and thus saving energy) only when there is a car being parked or unparked or if we detect any motion in the respective parking slot. This is done through the use of a motion sensor at each parking slot which detects an incoming or exiting car, or even a person approaching his/her car. Also, an added feature is switching on the lights at night to allow the security cameras to continue taping. This can be done by fetching the time of the day and switching on or off the lights by checking the

time using publish and subscribe. In case of an emergency, all the lights can be switched on or off by a button press located near the LCD as well.
c) Using a fire detection system. A single fire sensor can be deployed on each friend node to continuously check for fires. If a fire is detected, a “publish and subscribe” message can be used to toggle all LEDs in the parking lot. As each friend node is common to a number of parking slots, this problem is solved as well.

# Project Summary

In summary, the Smart Car Parking System is ready to be deployed in a car park area. It has been tested for use, and detects motion immediately, resulting in a light on the Parking Slot being turned on for visibility. As soon as motion is detected, proximity data is calculated on the LPN’s side, and if below a certain threshold, data is sent to the Friend node, which displays the status on an LCD. The Friend calculates the time of day based on an RTC, and send a set message to the LPN to conserve energy by going into Low Power mode (during nighttime) or to wake up more frequently and take readings (during daytime). The Friend also monitors for any fires in the vicinity, and in case of one, sends the fire status to all the other Friends throughout the parking lot. The project functionality has been tested, and is ready for deployment. The original problems which the project aimed at addressing have been appropriately solved, and have been done so in a way so as to conserve energy as well.

# Different Nodes used in the Network

## 1. LPN

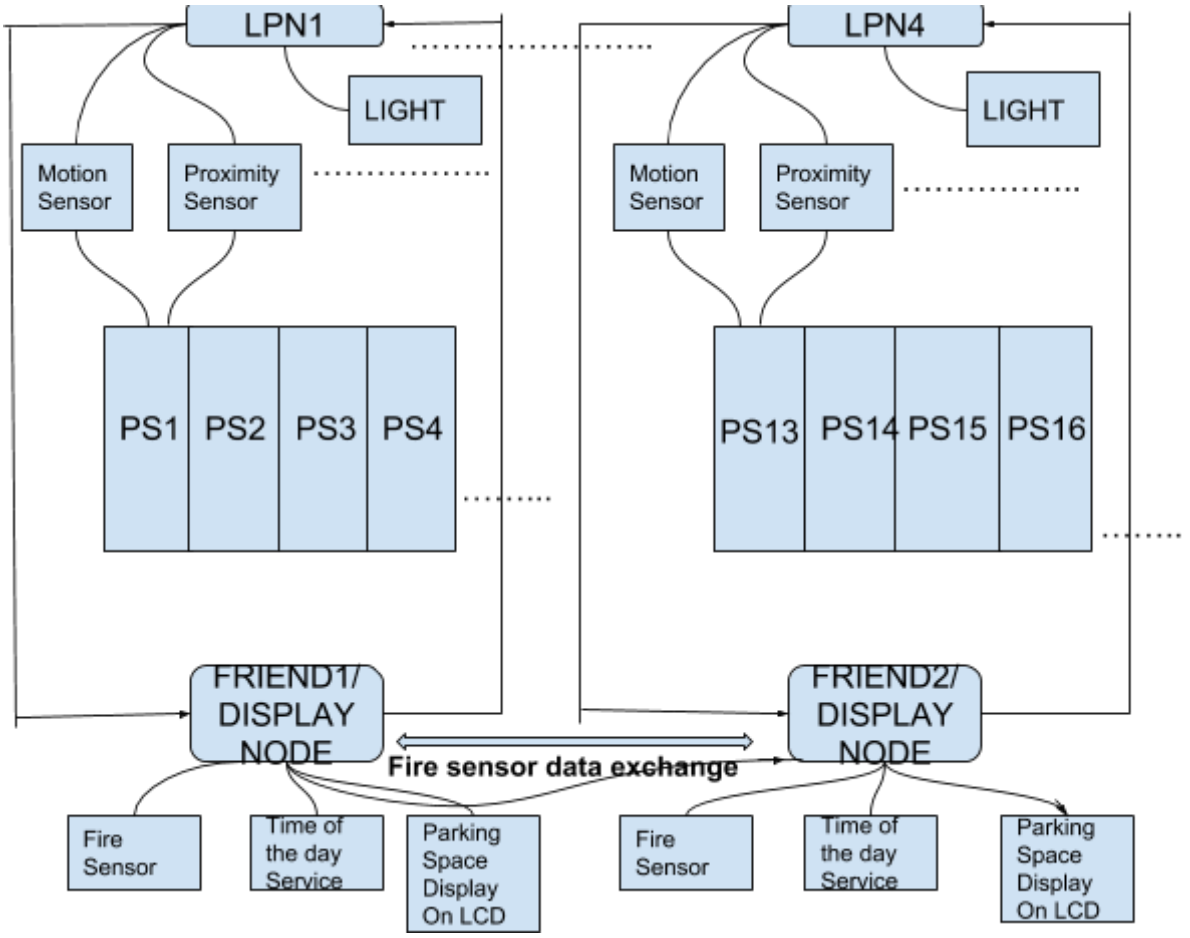
- This is a Low Power Node (Blue Gecko) with Server Model.
- Server Model uses the Generic ONOFF Mesh Model.
- Server Model which stores the state of the Parking Slot value depending upon the Motion Sensor and the Proximity Sensor and turns on the light(LEDs) accordingly.
- The sensors used are:
    1. Sparkfun Distance Sensor Breakout - RFD77402.(Proximity Sensor) (Quantity-1), cost - $ 12.71
    2. Sparkfun Open PIR - $ 14.95 (Quantity -1) (Could not find the p/n number on the website )

## 2. FRIEND

- This is a Friend Node(Blue Gecko) with Control Model.
- Server Model which stores the CO2 value using the Air Quality Sensor.
- Client Model which can control the light(LEDs) depending on time of day and by using the push buttons.

- The Sensor used is:
    1. Adafruit Air Quality sensor CCS811
- The Service used is:
    1. Real Time Service using on-board RTCC

# Functional Block Diagram



# Description of Individual Functional Projects

## LPN

The first two nodes will be Low Power Nodes(LPN). For our proposed project scenario we have one Low power node for every 4 parking slots. The Low power nodes are equipped with a motion and proximity sensor each. The motion sensor is used to detect if a person is approaching or leaving the parking slot and the lights are switched on/off correspondingly. The proximity sensor is used to detect an empty slot. This data is then sent to the Friend Node/Display Node which displays the status of the Parking Slots on the LCD screen.

## FRIEND

There will also be two Friend nodes implemented, each being connected to the required number of LPNs, and each of which stores the Time of Day obtained from the Real Time service using RTCC as the message in the Friend Queue.

The friend node also has a fire sensor which can be used to detect fires in the parking lot and send out an alarm to all other friend nodes in the network. The LPN sends the proximity sensor data to the friend and the friend displays the parking slot status on the LCD. The friend has an added control functionality of publishing to all LPNs, a message setting their lights on or off via a button push, giving a user control of all lights in the parking lot.

# Project Verification Plan

| No. | To be Verified | Definition of passing | Date Test Performed | Tested By | Measured result | Passed |
|---|---|---|---|---|---|---|
| 1. | All Devices provisioned with the help of OOB | Displayed successfully as provisioned devices on the silicon labs app by entering the passcode | 11/7/2018<br>11/8/2018<br>11/8/2018<br>11/9/2018 | Siddhant<br>Achyut<br>Swarupa<br>Sarthak | Provisioned using app and check the result of the api command | YES |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2. | Sensors Interfaced with the board | Receiving output values correctly as viewed on display LCD | 11/9/2018<br>11/10/2018<br>11/19/2018 | Achyut<br>Siddhant<br>Sarthak<br>Swarupa | Motion Sensor interfaced by turning on led if motion detected.<br>Working on Proximity Sensor using i2c.<br>Both air Quality Sensor interfaced successfully. | YES |
| 3. | Persistent Data routine implemented as expected | Sensor data, threshold values and state information retained on system boot id | 11/20/2018 | Siddhant<br>Swarupa<br>Sarthak | Checking the data by resetting the device | YES |
| 4 | Handle on Power up Behaviour | The states should retain the last values on Reset. For example -<br>Lights On or Off | 11/21/2018 | Siddhant | By checking the state values on terminal | YES |
| 5. | Friendship established between LPN and friend | Making sure it enters the friendship_established on both nodes by printing on serial port | 11/11/2018<br>11/8/2018<br>11/10/2018<br>11/9/2018 | Siddhant<br>Achyut<br>Sarthak<br>Swarupa | Checking the result of the api commands and viewing on terminal | YES |
| 6. | LPN correctly detects the empty slot using proximity sensor and sends the data to friend | The client_server_status is generated and the friend successfully receives the correct data. | 11/16/2018 | Siddhant | Car detected and sending proximity data | YES |
| 7. | Both friend nodes display the empty slot number. | The Slots are correctly displayed as empty or occupied | 12/5/2018<br>12/3/2018 | Swarupa<br>Sarthak | Slots numbers and Status displayed correctly | YES |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8. | The LPN nodes detect human presence accurately and turns on light accordingly. | Testing the sensor and making sure the corresponding LEDs are turning on. | 11/10/2018<br>11/10/2018 | Siddhant<br>Achyut | Human presence detected but turning on light pending | |
| 9. | The friend node correctly determines time of day and sends a on/off signal to the LPN every 30 Seconds | Intermittently displays time of day data on friend node's side.<br>Sends the on/off signal to the LPN and is able to turn on/off the LPN's light | 12/6/2018<br>12/5/2018 | Swarupa<br>Sarthak | The LED is turned on/off with every 30 seconds by client set message sent by the friend | YES |
| 10. | Friend Queue Implemented | The LPN is made to go into sleep mode for 15 seconds. It still receives the data sent by another node in the middle of those 15 seconds | 12/5/2018<br>12/5/2018 | Siddhant<br>Sarthak<br>Swarupa | The LPN is sent a message from another node while the LPN is sleeping. The LPN receives the message after a slight delay of around 8-10 seconds.. | YES |
| 11. | The friend node is able to detect fire correctly and sends an alarm | Fire Detected is displayed and the LED is toggled in a random pattern | 12/7/2018<br>12/7/2018 | Swarupa<br>Sarthak | LED toggling and Message Displayed | YES |
| 12 | Implementing on/off and on/off control models | States of the Nodes can be changed accurately and data can be sent accurately | 12/2/2018<br>12/3/2018<br>12/7/2018 | Siddhant<br>Sarthak<br>Swarupa | All three nodes can send/ receive data successfully and change their states according to the messages received. | YES |

| | | | | | | |
|---|---|---|---|---|---|---|
| 13 | Implementation of unicast and group addressing mode | Accurate data is received by the nodes specified by the different addressing modes | 12/3/2018<br>12/3/2018<br>12/7/2018 | Siddhant<br>Sarthak<br>Swarupa | Unicast messages sent via Client set commands by both friend nodes. The server receives the correct messages sent.<br>Server Publish and Client publish messages used to implement group addressing.<br>All other nodes successfully receive the message. | YES |
| 14 | Implementing mesh hopping | Data is transferred from a friend node to LPN via another friend node. | 12/8/2018<br>12/8/2018<br>12/8/2018 | Siddhant<br>Swarupa<br>Sarthak | One friend node is placed at such a distance from the LPN, where the LPN cannot receive messages anymore if there isn't another node between them.<br>Hopping is tested by placing another node in the middle and making sure the LPN is receiving data | YES |

| | | | | | | |
|---|---|---|---|---|---|---|
| 15 | Energy saving implemented on all low-power nodes | Monitoring energy profiler readings for LPN for an entire unit of time, eg. a day | 12/3/2018 | Siddhant | By checking if the LPN goes into EM3 | YES |

# Project Team members

1)Achyut Dave
2)Sarthak Jain
3)Siddhant Jajoo
4)Swarupa De

# References


- Silicon Labs' bluetooth Mesh Light and Switch Examples
- OOB Authentication - https://www.silabs.com/community/wireless/bluetooth/forum.topic.html/bluetooth_mesh_provi-PF3f
- Bluetooth Mesh Model and Profile Specification by Mesh Working Group
- Sparkfun Proximity Sensor - https://www.sparkfun.com/products/14539
- Sparkfun Motion Sensor - https://www.sparkfun.com/products/13285
- Adafruit Air Quality sensor CCS811 software library - https://adafruit.github.io/Adafruit_CCS811/html/class_adafruit___c_c_s811.html