

ECEN 5823 IoT Embedded Firmware

Title: Smart Farming using Bluetooth Mesh

Hardik Senjaliya

Guided by Prof Keith Graham

# Table of Contents

[Table of Contents 2](#_Toc532506435)

[1. The Problem addressed by this part 3](#_Toc532506436)

[2. The Solution to the problem 4](#_Toc532506437)

[3. Bill of Materials (BOM)/Sensors used by this part 5](#_Toc532506438)

[4. Functional Block Diagram 6](#_Toc532506439)

[4.1 Models used 6](#_Toc532506440)

[5. Flowchart of Friend and Proxy Node 7](#_Toc532506441)

[6. Development Schedule 8](#_Toc532506442)

[7. Validation Plan 9](#_Toc532506443)

[8. Summary 11](#_Toc532506444)

[8.1 Energy Optimization 11](#_Toc532506445)

[8.2 Application Security 11](#_Toc532506446)

[8.3 Lessons Learnt from the Project 11](#_Toc532506447)

[8.4 Diffuculties encountered during the Project 12](#_Toc532506448)

[8.5 Final Stauts of the Project 12](#_Toc532506449)

[**Figure 1 Functional Block Diagram** 6](#_Toc528529478)

# The Problem addressed by this part

In traditional farming the farmer must do every task manually, be it switching on/off the motor or switching on/off the sprinkler. Moreover, one must visit every part of the farm to check the current condition and then take necessary actions. This manual task can be done automatic by the solution implemented in this part and overall project.

# The Solution to the problem

So, the solution is to implement Friend Node and Proxy Node of the Bluetooth Mesh technology. As a friend node can connect multiple *Low Power Nodes (LPN),* the friend node in this part will connect to the *LPN*. The *friend node* will be interfaced with the *Soil Moisture sensor* and continuously monitors the sensor data. Therefore, whenever the *LPN* will ask for the sensor data from the *friend queue* Friend will send the data. The *LPN* then check the data and depending on the current soil moisture level, it will turn on/off the sprinkler system. Thus, by implementing the *Friend node* one can automatically switch on/off the sprinkler system based on the current requirement of the moisture level for the soil.

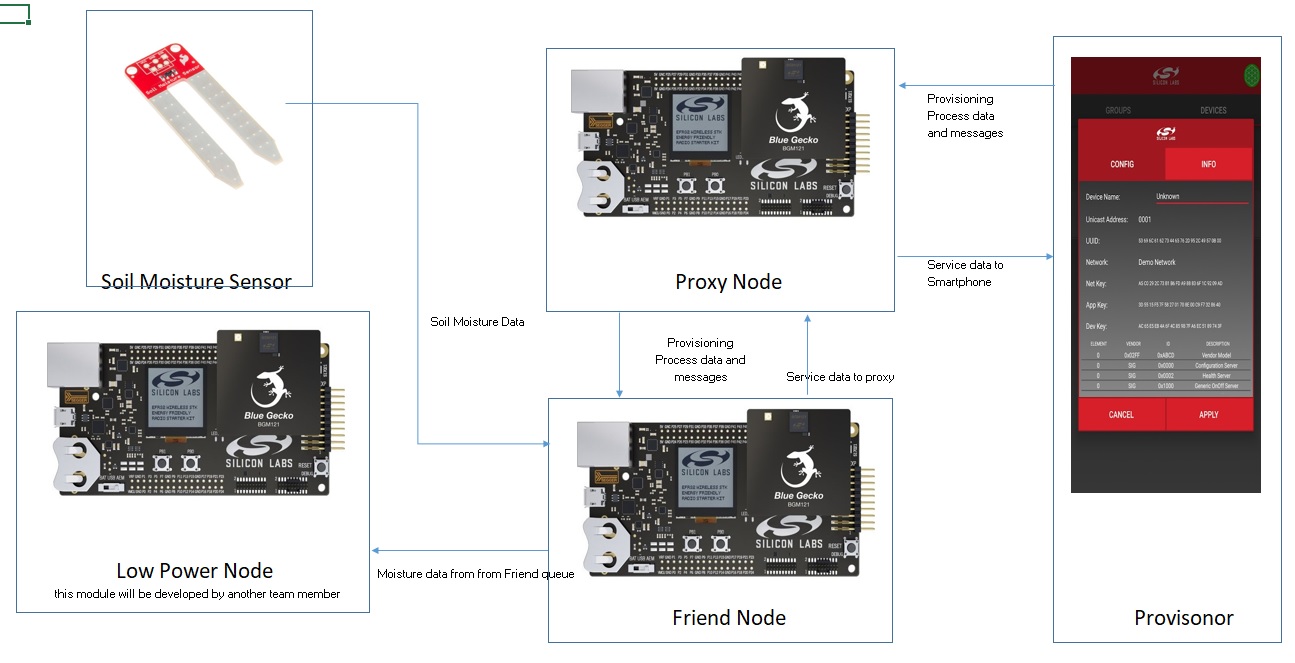
Secondly, by implementing the *Proxy* node which enables the connection between a smartphone and a Bluetooth Mesh network, one can monitor the soil moisture level remotely on the smartphone.

Thus, by implementing *friend* and *proxy* node, we are trying to eliminate manual tasks required to switch on/off the sprinkler systems by the farmer and it will result in reduced use of water and reduced efforts from the farmer.

# Bill of Materials (BOM)/Sensors used by this part

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Part Number** | **Price** | **Link** |
| Spark fun Soil Moisture Sensor | SEN-13322 ROHS | $4.95 | https://www.sparkfun.com/products/13322 |

# Functional Block Diagram



**Figure 1 Functional Block Diagram**

\*Images are used from respective manufacturer’s website

## 4.1 Models used

1. Friend Node Module: Client Model

* Client model: sends moisture data and with on/off command from the sensor to the LPN node and to the smartphone via proxy node.
* Soil Moisture Measurement – Moisture Level state

1. Proxy Node Module: Server Model

* Server model: receives moisture data from the Friend node and transmits to the smartphone.
* Sprinkler State: On or Off state

# Flowchart of Friend and Proxy Node

*Double click on the pdf icon to open the file.*

1. Friend Flowchart



1. Proxy Flowchart



*Please refer to the files submitted with this report if link does’t work.*

# Development Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Milestones | Expected Deadline | Completed  On | Stauts |
| Understanding Bluetooth Mesh | 10/31/2018 | 07/11/2018 | Completed |
| Developing persistent memory routine | 11/01/2018 | 11/01/2018 | Completed |
| Interfacing sensor to the Blue Gecko board | 11/09/2018 | 11/11/2018 | Completed |
| Load power management of the sensor | 11/10/2018 | 11/28/2018 | As it is a friend node not required (still tried to achieve by turning off ADC and LETIMER0 when rain is detected) |
| Integrating sensor to application code | 11/13/2018 | 11/13/2018 | Completed |
| Integrating LCD to application code | 11/14/2018 | 11/14/2018 | Completed |
| Developing Bluetooth Mesh code for server and client | 11/20/2018 | 11/23/2018 | Completed |
| Developing Bluetooth Mesh Friend node | 11/26/2018 | 11/30/2018 | Completed |
| Developing Bluetooth Mesh Proxy node | 11/28/2018 | 11/08/2018 | Completed |
| Integrating team member project modules | 11/30/2018 | 12/09/2018 | Completed |
| Validation and test of the project | 12/07/2018 | 12/10/2018 | Completed |

# Validation Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr.No | To be verified | Definition of Passing | Date test performed | Tested By | Measured Result | Passed |
| 1 | Developing software architecture (Proxy) | Debug using LEDs inside each event and verifying the sequence | 11/10/2018 | Hardik & Kiyasul | Commands and Events are successfully understood | Yes |
| 2 | Developing software architecture (Friend) | Debug using LEDs inside each event and verifying the sequence | 11/10/2018 | Hardik | Commands and Events are successfully understood | Yes |
| 3 | Persistent Memory Routine | Display the stored value on the LCD | 13/11/2018 | Hardik & Kiyasul | Loaded generic on/off server value on the display | Yes |
| 4 | Adding OOB authentication feature to the Proxy Node | A random number displayed on the LCD, so the app can authenticate | 16/11/2018 | Hardik & Kiyasul | Displayed a single digit random number(OOB) and successfully authenticated | Yes |
| 5 | Interface (Soil Moisture) sensor to Friend Node | ADC module should read the correct moisture level | 19/11/2018 | Hardik | Displayed sensor value on the Friend and varified in different moisture conditions | Yes |
| 6 | Integrating both sensors to application software | Bug free software with both the sensor working | 23/11/2018 | Hardik & Kiyasul | Integrated with Friend and LPN architecture and displayed the sensor values, verified with values measured in number 7 and 8 of Project Proposal | Yes |
| 7 | Sprinkler Off/On command from Silabs Mesh App | LED state should change on each module | 25/11/2018 | Hardik & Kiyasul | Basic test by toggling LED on each node | Yes |
| 8 | Friend sending messages from queue | Display the current queue message on LCD | 12/03/2018 | Hardik | By displaying moisture level on the LPN display and turning on/off the led based on the sent on/off command | Yes |
| 9 | Test Proxy Node relay feature | Add an extra relay node in the network and check if the data is traversing to LPN | 12/09/2018 | Hardik & Kiyasul | Tested by adding a node in the network and confirming the data received at the LPN side | Yes |
| 10 | Testing the entire system | Rigorous testing all corner cases on the system | 12/10/2018 | Hardik & Kiyasul | Tested all the scenarios based on the readings from the moisture and rain sensor | Yes  (Though we had a problem of intermittent loss of connection between Friend and LPN node) |

# Summary

## Energy Optimization

As friend node’s functionality is to always remain in the power on state to cache the data for LPN node, energy optimization could not be achieved at the higher level. However, I have tried to achieve the energy optimization as mentioned below.

The functionality of the Friend node is to read the Moisture level and based on the level publish on/off command to turn on/off the sprinkler. Because the LPN node is having the Rain Sensor to detect the rain, whenever it is raining, LPN node does not need the Moisture level.

Having this fact in the mind, I turned off LETIMER (which is used to periodically read and publish the Moisture level) and ADC (used for reading the Moisture level) which in turn stops publishing messages into the network. This is how energy optimization is achieved in this part of the project.

I could have achieved more energy optimization by using DMA controller to read the moisture level from the ADC thus decreasing the CPU usage.

## Application Security

To provide the security, I have implemented Out of Band (OOB) security feature of the Bluetooth Protocol to provision the nodes in the network.

While provisioning the node into the network, the user must enter the correct random number generated by the node. If entered wrong number, the node can not be a part of the network.

## Lessons Learnt from the Project

I learnt following lessons while working on the project.

* Learnt how to implement Bluetooth Mesh.
* Learnt how to develop ADC drivers and learnt about new sensor called Soil Moisture sensor
* Learnt about DMA controller and driver development
* Learnt how publishing and subscribing works in the Bluetooth mesh network.
* Learnt how unicast addressing works.
* Learnt how Out of Band authentication works.

## Diffuculties encountered during the Project

I encountered following difficulties while working on the project.

* While working on the Out of Band authentication, I couldn’t figure out what parameters I should use to achieve the desired functionality. The reason behind the problem is the contrast in the documents on which parameters should be used to achieve output OOB method.

(In the Silicon Labs API reference and Mesh Profile 1.0)

* We had a problem of intermittent loss of connection between Friend and LPN node. After lots of trial and error, we found the solution by changing the Poll time of the LPN node and for an hour we had a constant connection. However, the same solution did not work when we gave the demo.
* I could not implement Control model because of the Smartphone app limitation.

## Final Status of the Project

I have achieved all the desired and proposed functionalities of the project. However, we had a problem of intermittent loss of connection between Friend and LPN. I implemented functionalities using just client model. If smartphone app had more features, I could have achieved the same by implementing the Control model.