University of Colorado Boulder

ECEN 5823 – IoT Embedded Firmware (Spring-2020)

Final Project Report

Project Details	Description
Created By:	**Rushi James Macwan** **Bryan Cisneros** **Atharva Nandanwar** **Abhijeet Dutt Srivastava**
Project Title:	**Greenhouse Plant Monitoring System using Bluetooth Mesh**
Platform Used:	**Silicon Labs EFR32 Blue Gecko 13**
Report Submission Date:	***April 30th, 2020***

Credits: *All content presented in this proposal is our own work and all references are duly credited.*

Please Note: *This final project report has been created based on the documentation that was originally created for the two final project updates spread out across the project development cycle.*

University of Colorado Boulder

Table of Contents

Project Overview

Abstract:

This project report focuses on a prototype built for an automated and remote monitoring enabled system for a greenhouse plant that uses Bluetooth Mesh. The project is built on the several concepts learnt throughout the ECEN 5823 course that revolve around building connected IoT devices with varying capabilities and resource constraints while minimizing human interactions at the same time. The project specifically exercises the concepts that were exposed in class on low-energy firmware design and enhancement of IoT systems that provide services on a strict energy budget and constrained deployment.

Greenhouse plants require continuous monitoring and maintenance for its satisfactory and desirable growth which is at the core of the goals within the scope of this project. The project makes use of different sensor units to interact with the environment within the greenhouse to provide care and growth services for the plants contained inside. The sensors connected with the system provide information on the essential environmental properties within the greenhouse like temperature, presence of ambient light, intensity of UV light, and moisture in the greenhouse soil. The connected network architecture offered by Bluetooth Mesh specializes the system with connected nodes that share valuable information – i.e. threshold violations for the set thresholds pertaining to the environmental properties that are being monitored on the different nodes. Upon detection of the violations for any of the set thresholds on any of the nodes will trigger an alarm within the system that notifies the central node – that acts as a bridge in the system for monitoring.

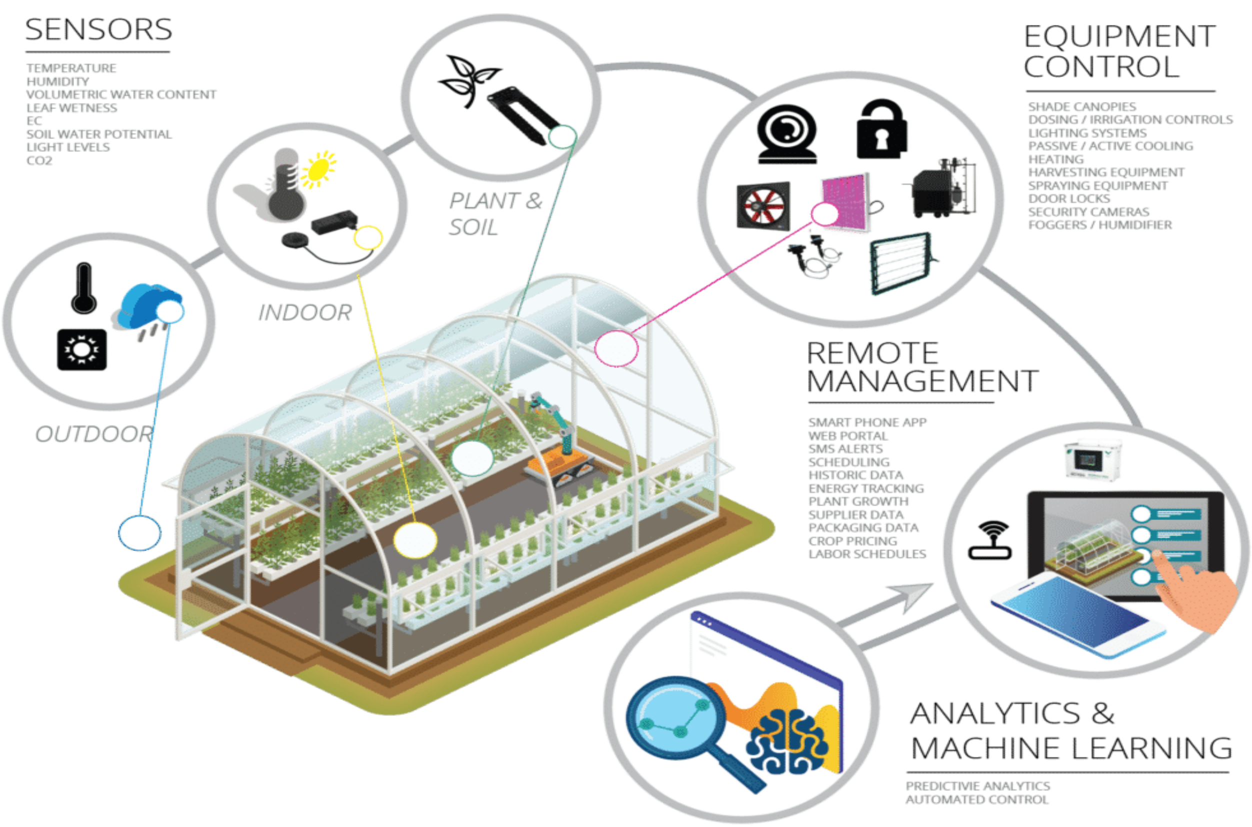
The project showcases an exhaustive implementation of Bluetooth Mesh network architecture and the concepts associated with its design – i.e. a set of Low Power Nodes and a Friend Node to support the system operations for the greenhouse plant monitoring.

Example use cases:

1. ***Smart Greenhouse Remote Monitoring System*** *(source)*
2. ***Greenhouse Remote Monitoring System*** *(source)*
3. ***Remote Monitoring System for Agriculture Greenhouses*** *(source)*
4. ***Indoor Agriculture: Smart Greenhouse Environmental Monitoring and Control*** *(source)*
5. ***Greenhouse Sensor Systems for Real Time Monitoring and Control*** *(source)*

The above use cases found from sources online share common goals that provide a basic example of a use case for this project. The sources above demonstrate a mixed use of sensors, equipment control and management, with enhanced remote management and which ultimately provides analytical services. The solutions provides an example use case of how plants within the greenhouses are protected from environmental extremes that can significantly hurt the plant growth and development. Besides reducing the threat models for the plants within the greenhouses, the services provided by these use cases reduce operational costs by reducing human interactions and involvements for the management of their greenhouses using connected IoT technologies that use a breath of sensors, actuators and smart algorithms that run the state machines behind these systems. Moreover, the use of these connected systems provides a multitude of data points for logging and reporting which can provide a great deal of feedback in the greenhouse operations which can ultimately improve the economy and throughput of these greenhouses.

The below image showcases an example use-case from a high-level view:



(Courtesy – image source)

Besides the technical advances and benefits of these use cases for greenhouses, the implemented systems also maximize revenues and profits, saves time and resources while increasing the yield, provides easy regulatory compliances by reliable remote monitoring and also surfaces low-cost and easy installation.

From a solid point of view, the most important advances observed in these use cases are depicted in the below image which provides the influential characteristics that have attracted investors in this area:

(Courtesy – image source)

From a practical point of view, this greenhouse plant monitoring project prototype can resource certain kinds of maintained environment for the growth of plants which may not be always achieved in the natural environment depending on the location of the plantation/farming area. For example, crops that include growing warm season vegetables (e.g. peppers) or tropical plants (e.g. cacti) in frigid regions on the planet can be achieved through automated and real-time managed greenhouses that provide desirable throughput.

Credits: *Please note, the above example use-cases are derived from the referenced sources and are duly credited for their original work.*

High Level Requirements

List of major requirements:

This project involves the following major elements that have gone into its design and development:

1. The project primarily makes use of the Silicon Labs EFR32BG13 (Blue Gecko) development kits that support implementing low-energy firmware for applications that are resource constrained as the one under focus of this project.

2. The system requires a coin cell (CR2032) for its operation of the Low-Power Nodes (LPNs) that would be located at different locations physically within the greenhouse plant that allows a Bluetooth Mesh (BTM) connection. However, for test/development purpose of this report, all interactions with the development boards were carried out with the help of the USB connection with the computer.

3. The Friend Node (FN) that acts a central node for interactions carried out with the LPNs, it requires a balanced power source – which means it is not resource constrained. This is because the FN operates at a much higher frequency than the LPNs in practical implementations. Within the scope of this project, the FN is continuously listening for the information arriving from the LPNs and that implies that it cannot be resource constrained as the transceiver would be under continuous use.

4. For all interactions with the external environment / physical properties, this project makes use of some sensors. The sensors are required in the project to measure the external properties. Based on the measurements, the system would compare the acquired information with the user-specified thresholds. If the thresholds are violated, alarm signals are created and sent through the respective LPN to the central node (FN). The sensors that are used in the implementation of this project are namely as given below:

    - Temperature Sensor
    - Moisture Sensor
    - Ambient Light Sensor
    - UV Light Sensor

5. The project makes use of persistent data that requires flash storage capabilities on the utilized hardware platform. Within the scope of this project, the Blue Gecko development kit provides a flash memory support for storing persistent data and special information that correspond to the provisioned node information to sustain its status within the BTM network. The persistent data is implemented for storing the thresholds for the sensor measurements on the LPNs while it is used to hold the alarm statuses on the FN in case of power resets.

List of requirements that has not been implemented:

1. This project would commercially have real-time monitoring and tracking systems in the form of smartphone applications. However, we have not been able to provide this support in our prototype because of constraints of time although it is commercially much of an integral part of the system.

2. This project would require intense protection for the electronic boards and circuits in the commercial setting for its deployment. This requirement comes for the safety of these electrical circuits from the presence of hazardous environmental conditions for these electrical systems within the greenhouses. For an example, a continuous presence of high temperature, exposure to sunlight and/or presence of moisture within the greenhouses is detrimental to the electrical systems deployed within the greenhouses and effective protection is commercially required, although is not researched within the scope of this project.

3. For a commercial / product deployment, custom electrical devices are designed to scale down the deployment costs and or regulate licensing costs. However, this is not at the focus of this project development since the development has been focussed on best firmware design practices.

4. For an industrial IoT product development, a continuous real-time monitoring, data-logging and machine-learning system is often deployed to improve the efficiency, reduce the operational costs and to increase the overall throughput of the greenhouses. However, this commercial setting requirement will has not been implemented in this project.

5. Although BTM networks inherently implement MITM protection using compulsory encryption of data exchanges. However, practical/industrial IoT devices/products would have an exhaustive software/hardware security interface with necessary firewalls. This area has not been focussed upon within the scope of this project.

6. Although testability of our system is exhaustively discovered, industrial products/systems would see a massive testing environment and large-scale test data that is analysed for enhancing software/hardware responses. This has not been researched within the scope of this project.

High Level Design

Description of proposed solution:

The proposed solution makes use of 4 Bluetooth Mesh nodes that perform various tasks ranging from monitoring the different environmental properties within the greenhouse as well as sending/receiving alarms based on threshold violations as set on each node with respect to the data that it is acquiring.

The 4 Bluetooth Mesh nodes required for realizing this proposed solution can be broken down into two types: 3 Low-Power Nodes (LPNs) and 1 Friend Node (FN). A Bluetooth Mesh provisioner will provision the nodes after which the system operations will come alive. The LPNs will collect the data points from the environmental properties (e.g. temperature, level of ambient light, etc.) and will compare it with the set thresholds stored as persistent data on the respective nodes. Upon threshold violations, the LPNs will send alarm notifications to the FN. The LPNs will also simultaneously display the information on the on-board LCDs for informing the plant operators of the critical information in real-time.

The FN will receive alarm statuses from the LPNs and display the "No Alarm / Alarm" status on its LCD (using the Generic On/Off Model). The FN will also receive the sensor readings (using the Generic Level Model) and it will display the same on its LCD. While using the Generic Level Model, the FN will acquire identify alarm set signals arriving from the LPNs using the flag 0xFFFF and would identify alarm clear signals using 0x7FFF. The LPNs will be configured to provide notifications at their wakeup instances to the FN when the newly acquired data points on the LPNs through its connected sensors falls outside the specified thresholds stored as persistent data on the FN. Such alarming conditions will trigger the FN to display its corresponding LCD row specifying the new status of the particular LPN sending the alarm.

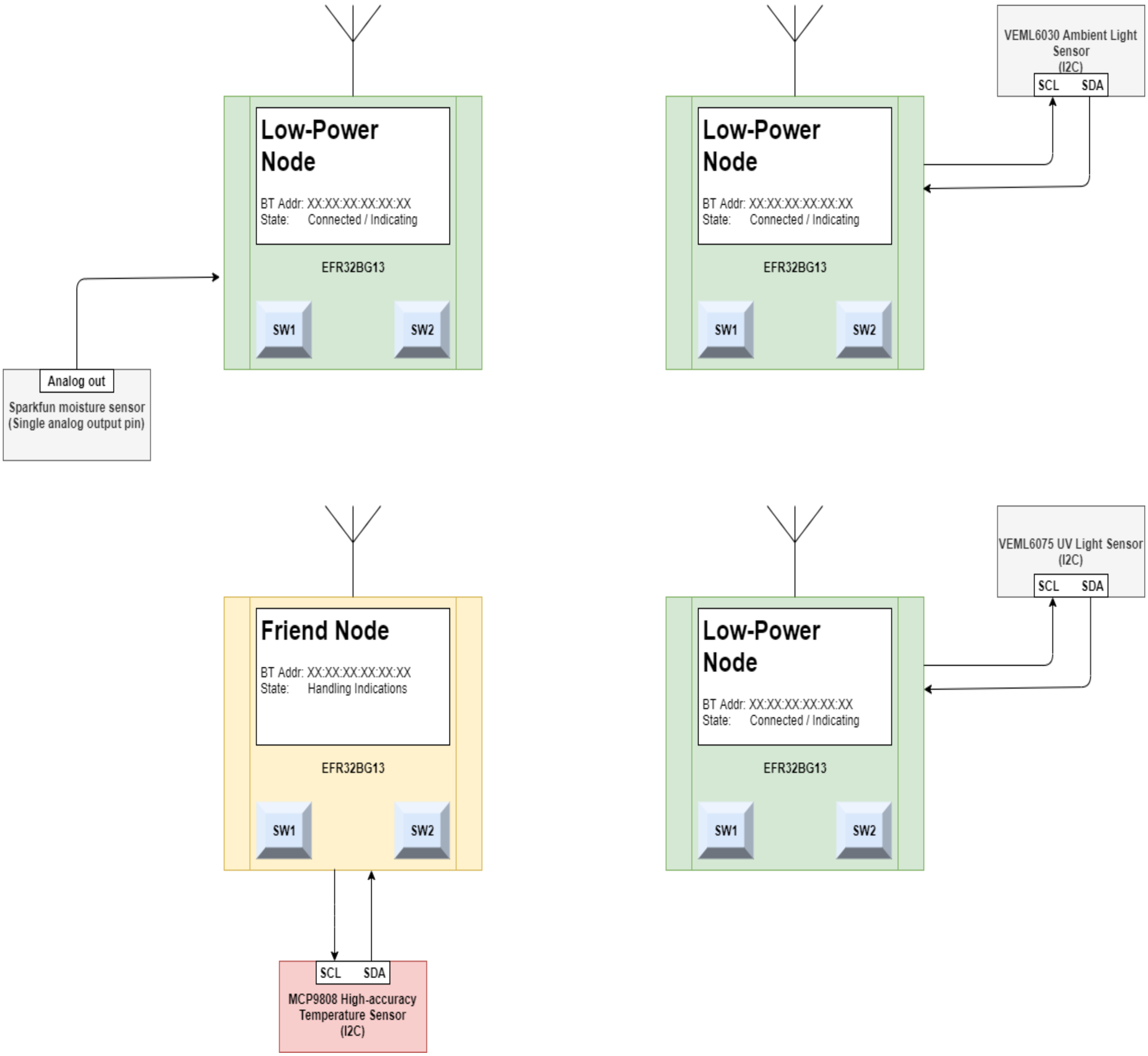
Above and beyond this basic functionality, the FN will store persistent data. As persistent data, the FN will store the alarm statuses that were obtained during its last reception from the LPNs and before it was last powered off. That means, if there is an alarm and if the FN runs out of power, the moment it returns to power, it will hold the alarm statuses as it had been before it ran out of power. This implies the persistent data helps the FN remain consistent with the LPNs. On the other hand, if the LPN runs out of power, and if there had been an alarm, the alarm would not be cleared as long as the FN receives a new alarm status from that particular LPN or it is factory reset. This is because the persistent data on the FN will hold the alarm status.

To meet all the above objectives, the project uses both the BTM Generic On/Off Model and Generic Level Model (separate firmware for both the Model implementations). The LPNs will demonstrate a Client Model whereas the FN will demonstrate a Server Model. The LPNs will be publishing the information available from the sensors interfaced to each of them, the FN will use a Subscriber model whereas the LPNs will be using the Publisher model. Besides that, the nodes will have a Configuration Model in order to support provisioning using a BTM provisioner in form of a proxy GATT device (e.g. smartphone) or an EFR32BG13 board that runs the provisioner firmware.

As for security, the project is dependent on the basic Man-In-The-Middle Attack security (MITM) and encryption security features provided by the BTM architecture.

Functional block diagram:

The high-level functional block diagram for the project is provided below:



Please Note: *Please refer to individual reports for a detailed overview of each subsystem block diagram and it's working.*

Description of sensors, data types & rates, wireless connections:

From a systems perspective, the project showcases the following list of sensors and actuators that will allow it to interact and influence the external environmental properties (e.g. temperature, level of ambient light, etc.).

Sensors/Actuators	Connections	Data Rates	Data Types
Ambient Light Sensor	I2C	1 reading per minute	Uint16
Moisture Sensor	Single Pin Analog	1 reading per minute	ADC Value (based on precision level)
UV Light Sensor	I2C	1 reading every 3 seconds	Uint8
Temperature Sensor	I2C	1 reading every second	Uint16

Friend Node – supports MCP9808 Temperature Sensor:

Developed by: Rushi James Macwan

The FN has a high-accuracy temperature sensor that will monitor the temperature variations within the greenhouse. The sensor – a MCP9808 I2C High-accuracy temperature sensor, operates at 3.3V/5V and will be powered using the EFR32BG13 supply. The FN will not power-constrained, however in a practical setting, the FN can also showcase an implementation of load-power management where a GPIO pin on the development kit is toggled that is fed to the power supply pin on the sensor to perform load power management. However, this feature has not been showcased in this project.

The FN is interfaced with the external sensor using the I2C0 lines. The I2C temperature sensor will provide temperature readings with an I2C maximum operational frequency of 400 kHz, and with a default resolution of +0.0625 degree Celsius. *(Courtesy – MCP9808 Datasheet)*

Low Power Node – supports VEML6075 UV light sensor:

Developed by: Bryan Cisneros

This LPN is connected to a UV light sensor which monitors the level of UVA and UVB light. The sensor (part number VEML6075) operates at 3.3V and communicates with the EFR32BG13 over I2C. The node will monitor the light levels and send periodic measurements over to the FN. In addition, if the node detects an unacceptable level of UV light, it will send an alert message to the FM, and it will clear the alert signal once the levels return to an acceptable level.

This node is also optimized as much as possible for low energy consumption. It is set up in the firmware as a low power node, and as such, the Bluetooth stack sleeps as much as possible. In addition, the node uses LPM to reduce the energy consumption of the UV light sensor as much as possible. When not in use, power to the sensor is turned off so that no energy is consumed by the sensor. With these techniques, the node uses the minimum energy possible to maintain connectivity to the mesh network and only uses energy for the sensor while a measurement is in progress.

Low Power Node – supports VEML 6030 ambient light sensor:

Developed by: Atharva Nandanwar

This Low Power Node uses I2C Ambient Light Sensor which will notify the Friend Node about the current value of light. The Ambient Light Sensor VEML6030 operates at 3.3V, and detects Ambient Light values based on Configuration Register. Setting up the configuration register such that it provides adequate power saving and accurate readings.

The sensor offsets the current usage by 1 mA. This current is being used by I2C pull-up resistors along with the sensor itself. Alerts are being generated by the Low Power Node based on raw sensor data. The reason GPIO based alerts were not included was that it would increase the complexity of the program. For low power operation, I am turning on I2C peripheral and peripheral clock only during I2C Transactions.

Low Power Node – supports moisture sensor:

Developed by: Abhijeet Dutt Srivastava

This LPN will be responsible for measuring the soil moisture levels using the soil moisture sensor. The sensor is rated at 3.3 V and can be easily powered using the on-board power supply of the EFR32BG13 development board.

The soil moisture sensor has a single pin analog pin the value of which depends upon the moisture level in the soil. The precision of this analog value depends upon the Vcc value used and the resolution of the ADC (The maximum resolution of the on-board ADC is 12 bits at up to 1 Msps).

Description of items/subsystems that would be included in a commercial setting:

Given below is an exhaustive list of the common items/subsystems pertaining to this project that would go into a commercial product which however has not implemented in this project.

- In a commercial setting, such a system would allow remote monitoring through proxy support for accessing/modifying the data and properties on the nodes. This project however does not fully enable that feature (although it provides the ability to provision the network of nodes using a GATT device – e.g. Android smartphone).
- A commercial system would allow the proxy support to send commands to the system remotely. These functions could be to power on/off the device, or modify network properties remotely. However, this project does not provide the same functionality.
- This project does not display the real-time sensor information (besides the alarm statuses) on the FN from the LPNs. Although a practical system would normally display sensor information levels on the central node, this project does not include this feature.
- A commercial system can often have actuators assisting the sensor network (of nodes). For example, if the temperature rises, the FN can implement a fan service that tries to manipulate the external environment that would lead to a reduction in the temperature. However, this project does not showcase any actuator.
- This project makes use of low-energy firmware design principles that showcases load-power management and other power-saving features like turning off display unless prompted by a user using the pushbuttons available on the nodes. However, in a general setting, other smart features may be available that do not require the user to manually/physically trigger the systems to display / interact with the user. The nodes can have haptic / motion support to detect someone from a distance to increase usability. This project does not showcase these features.

Summary of Individual Subsystems

Friend Node (Temperature Sensor):

Developed by: Rushi James Macwan

- **Architecture Model:** The FN is responsible for interacting with the LPNs and for acquiring the Generic On/Off Model based alarm notifications from the LPNs at regular intervals based on its wakeup cycles.
- **Alarms:** The FN will receive an alarm notifications when the model state information on the LPNs crosses a set threshold provided as a persistent data on the LPNs.
- **Persistent Data:** The FN will store the alarm status as persistent data every time it receives an update to the alarm status (i.e. a notification from a LPN).
- **LCD Integration:** The FN will respond to the regular notifications arriving from the LPNs and display the same information in a user-friendly format on the on-board LCD (e.g. displaying "Alarm/No Alarm" statuses for the LPNs that are connected, temperature readings from the sensor connected to FN, and other critical information like FN BTM information and node type – i.e. Friend Node).

Low-Power Node (UV light sensor):

Developed by: Byran Cisneros

Low Power Node for monitoring UV light levels in the greenhouse environment:

- **UV light monitoring:** The LPN will monitor UVA and UVB light intensity using the VEML6075 UV light sensor and notify of any light levels that fall outside the set thresholds.
- **Low Energy:** Will implement Low Energy Features to reduce the energy consumption and increase the battery life of that LPN. Sleeping into lowest possible energy modes, waking up only for I2C transactions, sensor measurements, and publishing data to the FN.
- **Persistent Data:** This node will store configuration data that will persist across reboots. This configuration data will include 3 parameters: frequency of measurements, alert threshold, and alert clear threshold.
- **LCD Integration:** LCD will be utilized to display the sensor data measured by the sensor, but only when requested by the user (by pressing PB0).

Low-Power Node (Ambient light sensor):

Developed by: Atharva Nandanwar

Ambient light monitoring will be implemented by gathering data from Ambient Light Sensor periodically. This data will provide us time of the day, and will be utilized to decisions or build models on time-of-day and growth of plants.

- **Low Energy:** Implementation of Low Energy Features to reduce the energy consumption and increase the battery life of that LPN. Sleeping into lowest possible energy modes, making I2C transactions, and servicing data to FN.

- **LCD Integration:** LCD will be utilized to display the sensor data actively as the sensor data is updated at regular intervals.
- **Persistent Data:** Persistent data was used to store the threshold values. Pertaining to initial plan, giving the user of greenhouse monitoring system an option to change certain parameters like threshold values, and update time. Currently, Low Power Node has provisions to store threshold high and threshold low values.
- **Real-time Display:** Shows real time status of the connection status, and ambient light data.
- **Flexible Client Models:** Currently, the code is designed in such a way that a compile-time switch can change the publishing model from Level Client to On/Off Client.

Low-Power Node (Moisture sensor):

Developed by: Abhijeet Dutt Srivastava

Low Power Node for measuring soil moisture levels will have the following features:

- Soil moisture measurement: The LPN will be integrated with a soil moisture sensor by using an ADC.
- Expose custom services: To transmit the soil moisture level measurements, two custom services will be exposed. The LPN will send fixed-interval notifications to a Friend Node (FN) over a Bluetooth Mesh Network.
- Alarm Notifications: The LPN will monitor the sensor at fixed interval and compare the values with a threshold condition stored as persistent data. In case, values cross the threshold an alarm notification will be sent to the FN.
- Integration with switch: The switches PB1 and PB0 are used to easily factory reset the device
- Sleep and low power: Energy modes will be implemented to ensure that the LPN stays in lowest energy states possible.
- LCD integration: LCD will be integrated to display the soil moisture levels and ambient light levels

Project Team Members

The following students are involved as a team for designing and developing this project:

1. Rushi James Macwan
2. Bryan Cisneros
3. Atharva Nandanwar
4. Abhijeet Dutt Srivastava

Provisioner Repository – please visit here.

Team Project Validation Plan

Please find the project validation plan here.

Final Project Status

The final project status for the project development is COMPLETED.

The final project includes a complete list of features that were proposed in the original project proposal – excluding a few. An exhaustive list of the features that have been implemented in the final project submission has been provided below:

- The Friend Node allows both Generic On/Off Server Model application as well as Generic Level On/Off Server Model application for the Low Power Node connections that also both of these models.
- The repository contains separate firmware support for both these models and have been tested and demonstrated for its features and usability.
- The nodes implement persistent data to retain critical information – like thresholds for the sensor data acquisition on the LPNs and status of the alarm buffer on the FN.
- The nodes provide a provisioning and factory reset feature at runtime using both a GATT device that can be used for provisioning the nodes or using the embedded provisioner.
- The nodes provide full display support to report critical information for the alarm statuses on the FN and sensor readings available on the LPNs in addition to providing other secondary information such as name of the node and its address.
- The LPNs provide load power management functionality to reduce power consumption and extend battery time for practical deployment of this project.
- The FN supports interactive UI with the help of pushbuttons and the LCD display to either turn on/off the LCD display and to clear the alarms generated on the FN for overriding the alarm notifications and/or refreshing/clearing persistent data.
- The LPNs provide generic on/off and generic level readings (separately when implemented with the two different firmware codes) and the level readings are readily reported on the FN.
- The requirement of interfacing one external sensor on each node has been met. More information on the interfacing of sensors and related hardware/firmware designs can be discovered in the individual project reports as well as the sections provided in this report dedicated to the subsystem information.
- The nodes provide sufficient error logging and error checking for the BT Mesh stack events that occur and the gecko commands that are executed within those events.
- Adequate error logging and checking has been also provided for other peripheral and interfaces (e.g. I2C transactions) to demonstrate clean, robust and reliable firmware.

*Please see the glossary below.

LPN – Low Power Node
FN – Friend Node
BTM – Bluetooth Mesh Network