**ECEN5823 Final Project Team Proposal**

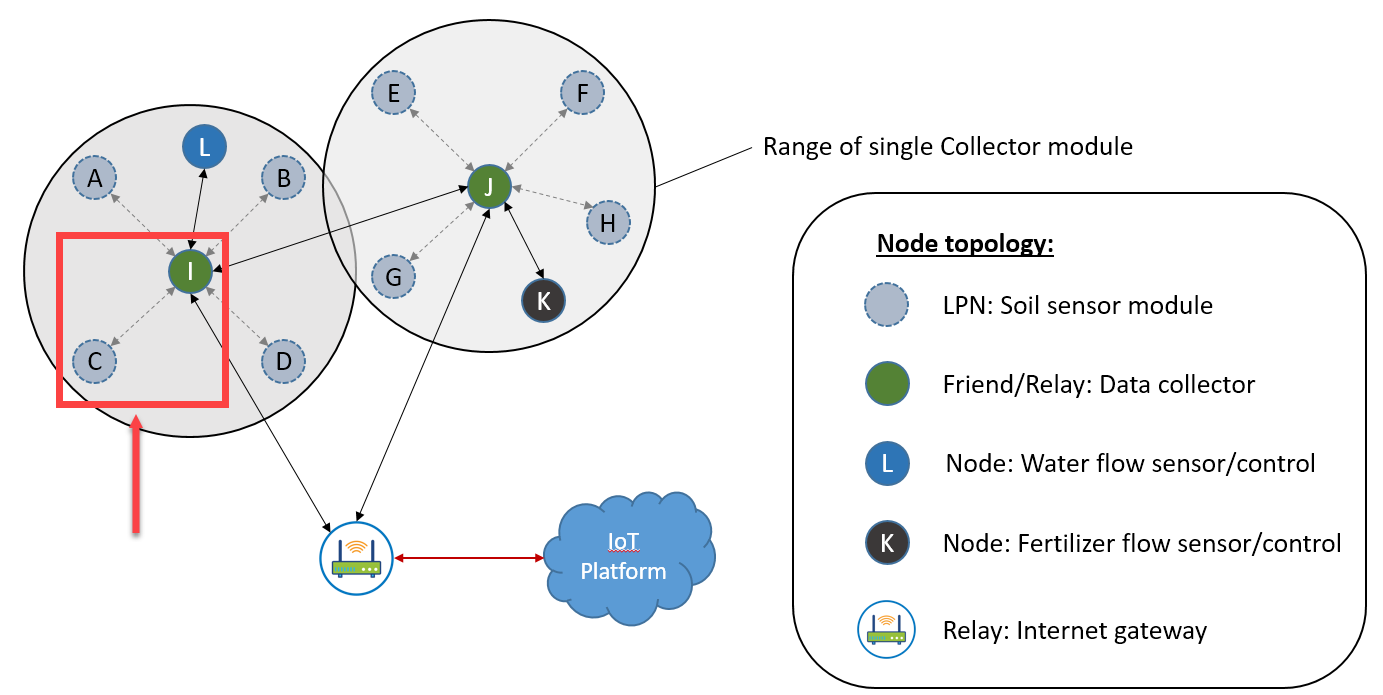
**Project Name:** Intelligent Agriculture Control System

**Members:** Victor Kronberg

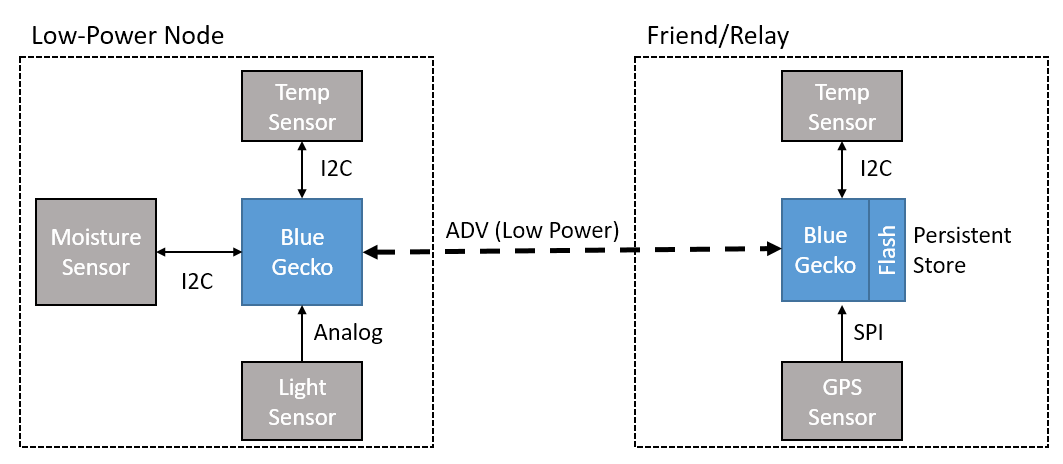
1. **Project Status**
   1. Thus far, the sensor breakout boards have been acquired from SparkFun, header pins have been soldered on and they are connected to a breadboard for testing and firmware development. The initial skeleton codebase has been checked out (btmesh) and functionality such as timer, gpio, and display have been fully ported over from other projects. Initial ADC initialization code has been implemented, but not fully tested. Pins have been selected for ADC input.
   2. Challenges so far have been finding time. I have not hit the meat of the technical portion, so no challenges there so far.
2. **Subsystem Overview**
   1. With a rapidly increasing population and limited fresh water resources for agriculture, utilizing our resources as efficiently as possible become ever more critical in the coming years. On top of that, high water prices and global agricultural competition mean that U.S. farmers need every advantage that they can get in order to stay financially competitive. Without detailed understanding of how the environment is impacting the soil and how the soil responds to treatments (water/chemical), it becomes ever more challenging to stay competitive.
3. **Subsystem High Level Design**
   1. The idea behind this project is to provide a modular, scalable wireless solution for monitoring and controlling soil conditions in an agricultural setting in order to optimize watering and fertilization to give the farmer the greatest yield per dollar spent on resources. This includes soil moisture level, soil temperature, sun exposure, and mineral content. There would also be sensors for monitoring water flow for irrigation and mineral content in fertilizer applications. This information would be fed into an intelligent platform that utilizes AI to help a farmer to optimize their water usage and soil treatments based on the soils’ response to watering and fertilizer applications.

Subsystem proof-of-concept will implement a low power node and friend node. This subsystem is responsible for sensing the properties of the soil and transmitting that information back to a “collector”. The “collector” stores the sensor data in persistent memory in case of network outage or system/power failure. In a full deployment, the “collector” would also transmit and relay sensor data towards the Internet gateway.

* 1. Subsystem Block Diagram:



**Figure 1: Subsystem of network to implement**



**Figure 2: Block diagram of subsystem and sensors to implement**

* 1. Sensors for Low Power Node:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sensor** | **Data type** | **Property Measured** | **Sparkfun P/N** |
| Moisture sensor | Analog (ADC) | Soil moisture content | <https://www.sparkfun.com/products/13637> |
| Ambient light sensor | Analog (ADC) | Ambient light | <https://www.sparkfun.com/products/8688> |
| Humidity & Temperature | I2C | Soil/surface temperature | Si7021 – Included on Blue Gecko |

Sensors for Friend/Relay Node:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sensor** | **Data type** | **Property Measured** | **Sparkfun P/N** |
| GPS | SPI | GPS Location | <https://www.sparkfun.com/products/15193> |
| Humidity & Temperature | I2C | Soil/surface temperature | Si7021 – Included on Blue Gecko |

* 1. *Sparkfun part numbers above*
  2. Profiles and exposed Services to implement:
     1. **Environment Sensing Profile**
        1. Irradiance Service
        2. Humidity Service
        3. Temperature Service
        4. (Custom) – Soil moisture content Service
     2. (optional) **Location and Navigation** **Profile**
        1. Location Service
  3. **Implementation Plan:**
     1. Moisture sensor
        1. ADC Input: Expansion pin 14
           1. Port A, Pin 1
        2. Power (3.3V): Expansion pin 20
        3. Ground (GND): Expansion pin 19
     2. Ambient Light sensor
        1. ADC Input: Expansion pin 12
           1. Port A, Pin 0
        2. Power (3.3V): Expansion pin 20
        3. Ground (GND): Expansion pin 19
     3. Temp/Humidity sensor
        1. Utilizing Si7021 code from previous assignments for temperature
        2. Add code for also reading off humidity measurements
     4. GPS
        1. SPI
           1. SPI\_CS: Expansion Pin 10

Port C, Pin 9

* + - * 1. SPI\_SCK: Expansion Pin 8

Port C, Pin 8

* + - * 1. SPI\_MISO: Expansion Pin 6

Port C, Pin 7

* + - * 1. SPI\_MOSI: Expansion Pin 4

Port C, Pin 6

* + - 1. Power (3.3V): Expansion pin 20
      2. Ground (GND): Expansion pin 19
  1. *Command table in “****documents****” folder*
  2. **Persistent data**
     1. Persistent data will be stored in a struct in the server. Struct will contain client state (temp/humidity, soil moisture level, and ambient light level) and server state (temp/humidity). GPS is assumed to be static, so there should not be a need to store that as well – if there is space, then it could be. Each struct will be less than 56 bytes. Persistent data will store up to the previous 128 entries – each entry taken when LPN publishes its state.
  3. **User Interface**
     1. Client/Low-Power Node
        1. Display device name: “Low-power node”
        2. Display device BT address
        3. Display connection status: “Unprovisioned”, “Provisioned”, “Low power”? TBD
        4. Other status?
     2. Server/Friend & Relay
        1. Display device name: “Friend node”
        2. Display device BT address
        3. Display connection status: “Unprovisioned”, “Provisioned”, “Low power”? TBD
        4. Display LPN state (might need to cycle through?):
           1. Temp/humidity
           2. Soil moisture level
           3. Ambient light
        5. Display own state:
           1. Temp/humidity
           2. GPS location

1. **Proposed Development Schedule (task – target implementation date)**
   1. Interface software to moisture and light sensor (ADC) – 11/14
      1. Develop ADC initialization routine – *initial attempt, complete*
   2. Interface software to temp/humidity sensor (I2C) – 11/16
   3. Integrating sensors to application code – 11/19
   4. Integrating LCD to application code – 11/20
   5. Load Power Management of sensors – 11/22
   6. Developing BLE Service / Client code – 11/26
   7. Develop persistent memory routine – 11/27
   8. Firmware Update (DFU) – 11/30
   9. Validation of the project – 12/1-12/10