Smart Farm

Network Documentation

Daniel Mills

3/31/2022

Table of Contents

Introduction

Design

Flowcharts

Power Consumption

Lab Testing Results

Stability Test

Data Collection Test

Field Test

3

3

4

5

7

Introduction

This document outlines the design and implementation of the Smart Farm network.

Design

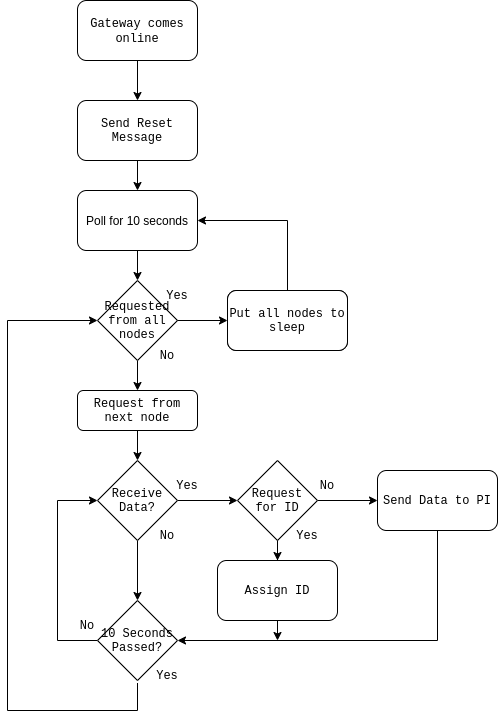
The Smart Farm network is comprised of three types of devices: sensor nodes, a gateway, and a server.

The sensor nodes are ESP-32 microcontrollers that have a wired connection to various sensors that communicate via I2C protocol. These nodes are configured to measure temperature, humidity, air pressure, altitude, and capacitance. The sensor stack that obtains these measurements is the STEMMA soil sensor and the BME280. In order to accommodate more or different sensors, a firmware update is required. These sensor nodes are designed to be low-power and low-cost, which allows the farmer to buy many of them. By virtue of being low-power, the devices do not need maintenance often.

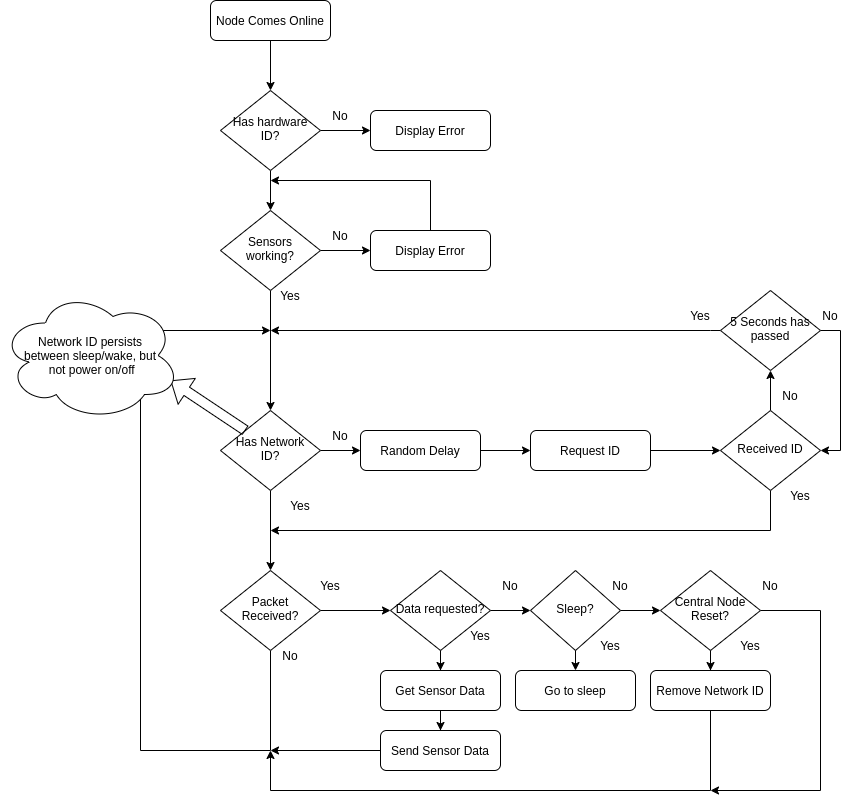
The gateway is also an ESP-32 microcontroller. Its purpose is to manage the mesh network of sensor nodes. When it collects data, it forwards the data via a physical serial connection to the server. When a sensor node comes online, it persists in requesting an ID until the gateway assigns one to it. The sensor node includes a randomly generated number in its request. When the gateway broadcasts the ID, it includes the token to ensure that the correct sensor node receives the ID. When the gateway comes online, it broadcasts a reset signal to all of the sensor nodes. This tells them to reset their network connection. The gateway further manages the network by coordinating when data is sent, when all nodes enter deep-sleep mode, and how long they sleep. It sends a packet to each node on the network requesting data from it. After it has requested data from each node, it broadcasts a packet that instructs them to fall asleep and provides the duration for their sleep. The gateway stays awake while the connected nodes sleep so that it may admit nodes that are not connected onto the network. Each time a node requests to join the network, the gateway prunes stale connections and admits it to the network.

The server powers the gateway via the serial connection between the two devices. It runs a script on startup that parses sensor data that the gateway collects into a comma separated variable file. The server hosts this data and also has a script that produces graphs of the data. The farmer can run these scripts at any time he desires; however, a more user-friendly way to visualize the data exists. The server hosts a website that can be accessed on any device with an internet connection. The website displays graphs and gives the user filter options. The website is explained in greater depth in another document. We chose a Raspberry Pi to use as our server because it has lower power demands, is more portable, and has adequate storage space.

Flowcharts



**Figure 1. Gateway Flowchart**



**Figure 2. Sensor Node Flowchart**

Power Consumption

As previously discussed, the value of the Smart Farm hinges upon its ability to function as a low-power low-cost alternative to the current solutions. Measurements outlining the power consumption of the ESP-32 sensor nodes appear in the following table. Battery life is expected to be high. The battery-drain can be offset by adding a solar panel to our current design - an option that we have considered, but did not have enough time to implement. In short, the farmer should not need to replace batteries for daily usage, only when the batteries actually fail. According to our calculations, the battery should last for approximately 195 hours, or a little more than eight days. We expect the actual result to be roughly eight days.

| Sensor Node | | Trials | | |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Board | State | 1 | 2 | 3 | Average A | Average V |
| 1 | Boot | 0.062 | 0.062 | 0.070 | 0.065 | 5.04 |
| Radio | 0.168 | 0.168 | 0.157 | 0.164 |
| Steady-state | 0.061 | 0.061 | 0.069 | 0.064 |
| Sleep | 0.008 | 0.008 | 0.008 | 0.008 |
| 2 | Boot | 0.066 | 0.080 | 0.079 | 0.075 | 5.03 |
| Radio | 0.171 | 0.173 | 0.175 | 0.173 |
| Steady-state | 0.066 | 0.061 | 0.063 | 0.063 |
| Sleep | 0.008 | 0.007 | 0.007 | 0.007 |
| 3 | Boot | 0.086 | 0.078 | 0.085 | 0.083 | 5.04 |
| Radio | 0.181 | 0.173 | 0.164 | 0.173 |
| Steady-state | 0.066 | 0.070 | 0.064 | 0.067 |
| Sleep | 0.011 | 0.010 | 0.009 | 0.010 |
| 4 | Boot | 0.080 | 0.070 | 0.080 | 0.077 | 5.05 |
| Radio | 0.167 | 0.173 | 0.176 | 0.172 |
| Steady-state | 0.063 | 0.065 | 0.069 | 0.066 |
| Sleep | 0.010 | 0.008 | 0.009 | 0.009 |
|  |  |  |  |  |  |  |
| State | Time per state | Time in S | Nodes | Average A | A Weighted S | Total Average V |
| Boot | 2 sec | 2 |  | 0.075 | 0.1496666667 | 5.04 |
| Radio | 0.5 sec | 0.5 |  | 0.171 | 0.08525 |
| Steady-state | 8 sec per node | 8 | 15 | 0.065 | 7.78 |
| Sleep | 15 min | 900 |  | 0.009 | 7.725 |
| Total |  | 1022.5 |  | 0.319 | 15.73991667 |
|  |  |  |  |  |  |  |
| Average mA Weighted by Time |  |  |  |  | 0.01539356153 |  |
| Battery Capacity | 3 | Ah |  | Battery Life | 194.8866735 |  |

**Table 1. Sensor Node Battery Life Data**

The battery life calculation for the gateway node helps us understand how much power it drains. It is not designed to enter low-power mode.

| Gateway | | Trials | | |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Board | State | 1 | 2 | 3 | Average Amps | Average Volts |
| 1 | Steady-State | 0.064 | 0.063 | 0.063 | 0.063 | 5.05 |
| Radio | 0.150 | 0.151 | 0.153 | 0.151 |
|  |  |  |  |  |  |  |
| State | Time per state |  |  | Average A | A Weighted S |  |
| Steady-state | 8 |  |  | 0.063 | 0.5066666667 |  |
| Radio | 0.5 |  |  | 0.151 | 0.07566666667 |  |
| Total | 8.5 |  |  | 0.215 | 0.5823333333 |  |
|  |  |  |  |  |  |  |
| Average mA Weighted by Time |  |  |  |  | 0.06850980392 |  |
| Battery Capacity | 3 | Ah |  | Battery Life | 43.78935318 |  |

**Table 2. Gateway Battery Life Data**

Lab Testing Results

The Smart Farm team performed many tests in the lab. This section will highlight only the most insightful and important ones.

*Stability Test:*

We connected one node to the gateway and connected the gateway to the Raspberry Pi. The sensor node was provided a continuous power source. We left the network in this state for 30 days without interference. We expected to find the sensor node still properly sending and receiving packets if the network was stable. If the network was not stable, we expected to find the node online, but not sending or receiving packets. This was before we implemented data storage, so we cannot verify whether or not the data was accurate for the entire duration of the 30 days (we performed subsequent tests for the purpose of data validation). What we found at the end of the 30 days was that the one node was connected to the network as it should be and was sending accurate data upon request. It was working just as it should after 30 days of isolation.

*Data Collection Test:*

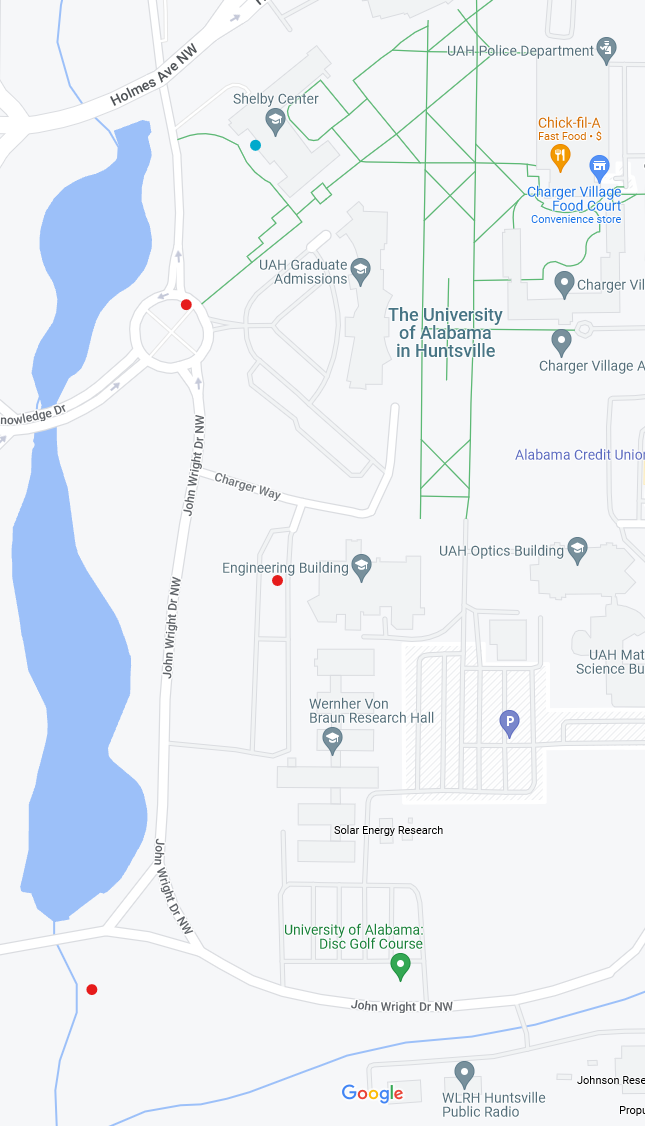
We powered on four nodes and properly secured the sensors to each node. Next, the gateway assigned a network layer ID to each node. The gateway then began collecting data from each node. Unfortunately, the first node dropped off of the network after only 20 minutes. We are still not sure why this happened, but we made updates to improve the stability of the network since then. It may have been disturbed, or its power source may have unexpectedly died. The remaining 3 nodes stayed on the network for 20 hours and the PI collected data as the gateway received it. One last thing to note is that the BME280 sensor calculates altitude based upon pressure relative to pressure at sea level. As pressure changes over time, so does the altitude for each node.

The graphed results for this test are available on github here:

<https://github.com/dandeto/Smart-Farm/tree/5458cccb586fea909f1ec651a3218c035a600b83/Pi/output/images>

*Field Test:*

We went on the UAH grounds and collected data. We placed the last node in a tree, and the wires connecting its sensors got disconnected - likely by the wind. Despite that, we got good data from the remaining 3 sensor nodes. The red dots on the following figure show where we placed each sensor node. The blue dot on the Shelby building shows where we placed the gateway.



**Figure 3. Field Test Locations**

The graphed results of this field test are available on github here: <https://github.com/dandeto/Smart-Farm/tree/main/Pi/output/images>