**Memo**

To: Professor Pisano, Professor Hirsch, Professor Osama

From: Ben Livney, Maxine Loebs, Sergio Pareja, Emanuel Perez, Noah Spahn

Team: 27 - Plants are Neat

Date: 11/19/20

Subject: First Prototype Test Plan

1. **Abstract**

For our test, we have decided to show our progress in a few key areas of our project, namely node communication, power, and data display for the user. We will show that our nodes are able to communicate with one another and transmit data, that our solar panel is able to charge our battery, and both the solar panel and battery can deliver power to a load via a power path management IC, and that we are able to display data to the user for interpretation.

**2.0 Required Software and Hardware Components**

2.1 Hardware

* 2 Adafruit Feather M0 microcontrollers
* 1 BQ24074 module
* 1 3.5W 6V solar panel
* 1 3.7V Li-Ion battery pack, not fully charged
* 1 I2C sensor
  1. Software
* Arduino IDE
* MatLab IoT ThingSpeak
* RadioHead.h
* RH\_RF95.h

**3.0 Communication Between Nodes**

3.1 Network Procedure

1. Mother Node sends poll request for a specific node to other nodes
2. If node is not the one requested, then the node sends the poll request to other nodes as well
3. Once the correct node is found, it sends data to the one asking for it
4. Data is passed around the nodes till it reaches the mother node

3.2 Network Setup

* One outer node and one mother node
* Network Protocol
* Transmission device (LoRa)

3.3 Network Criteria

* Poll is successfully sent to other nodes
* Poll successfully reaches the requested node
* Requested node successfully sends back it’s sensor data to the poll sender
* Mother node receives correct sensor data from the requested node
* Able to repeat the process to find all the nodes in the network
* If node is not requested but receives the poll, the node sends resends the poll to other nodes (closest one)

**4.0 Solar Charging and Power Path Management**

4.1 Solar Charging Procedure

* Connect the solar panel DC output to the DC input of the BQ24074 using the barrel connector.
* Connect the LiPo output of the BQ24074 to the positive terminal of the battery pack.
* Measure the battery voltage.
* Connect the GND output of the BQ24074 to the negative terminal of the battery pack.

4.2 Solar Charging Measurable Criteria

* When the solar panel is powering the BQ24074, a PGOOD LED should turn on.
* When charging is enabled, a charging LED indicator should turn on and a positive charging current should be measured.

4.3 Power Path Management Procedure

* Disconnect the battery pack from the BQ24074.
* Connect an LED and resistor in series to the load output of the BQ24074.
* Remove the solar panel and reconnect the battery.

4.4 Power Path Management Measurable Criteria

* When powered by the solar panel, the LED should turn on.
* When powered by the battery pack, the LED should also turn on.

**5.0 Data Analysis and Visualization**

5.1 MatLab Iot ThingSpeak

* Iot ThingSpeak tool used to connect the uplink node, database, analytical functionality and visualisation into a html output stream.

5.2 Data Representation

* Data is written and read as a csv file with the unique channel ID for each console uploading and doing analytics. This allows the systems to be scaled as more than one system can use the same procedures with different labeled data as long as more space is allocated on the server.

5.3 Software:

* MatLab 3 Scripts:
  + Read Data From Data Channel
  + Analyse and Generate Visualisation of Data
  + Write Data To Data Channel
* Html 1 Script:
  + Stream Visualisation to Html Webpage

5.4 Pre-Testing Procedure:

* Initial step of the Pre-Testing Procedure was to load previous data into each html page.
* Ensure any past Stream Visualisation analysis and visual is visible on the page.
* Ensure any past data points are noted before the test.

5.5 Testing Procedure:

* The html block is loaded and refreshed with an old data graph.
* Run MatLab channel write to add data to the channel.
* After a few seconds the html block will refresh with new sample data.
* Again Run MatLab channel write to add another data point to the channel.
* After a few seconds the html block will refresh with new sample data.

5.6 Measurable Criteria:

* Before Upload Data should stay constant.
* After Upload the graph should update to include new data points at the current timestamp.
* The visualisation should update without the whole page restarting.

5.7 Measurement Results:

* The graph refreshes itself without a page refresh.
* The graph updates data points after upload including only the additional data points added. including a timestamp with when the data was taken.

**6.0 Using Sensor and Collecting Data**

6.1 Setup

* Upload code onto board
* Connect the sensors

6.2 Collecting data

* Let the code run, get data readings on temperature and pressure
* Data will be displayed in serial monitor

6.3 Measurable criteria

* Readings are coming back accurately
* Demonstrate that readings change in response to change in temperature (can’t really vary pressure)

**7.0 Conclusion**

After the conclusion of our tests, we determined that we are able to successfully transmit data between sensor nodes, implement a sleep wake system that bounces data between nodes ,charge the battery we built with with the solar charger and solar panel, upload data and generate visualisation with collected sensor data. Our next task for our group is to continue working on the mesh network development so that it is better suited for scale as well as connecting all our components together that we were able to test out in the prototype. The wake protocole test will be connected to the data transfer test enabling us to wake and sleep nodes for data collection and transmission. The console node will then be connected to the uplink test using a wifi shield as our bridge between the board and the internet. This will enable the upload and updated visualisations using the most real time data available. Given we complete these tasks we would like to fully integrate the system into a single board allowing for a cleaner and more dependable solution.