

UNIVERSITY OF MINNESOTA—TWIN CITIES

OPEN SOURCE DATA LOGGER  
SENIOR DESIGN TEAM

---

**Design Proposal**

---

Bobby SCHULZ  
Jeff WORM  
Luke CESARZ  
David NICKEL  
Ying YANG

February 17, 2017

# 1 Introduction

In today's world, a frequently desired technology among environmental scientists is a low cost, light weight, and low maintenance solution for logging data out in the field. The current implementation of the ALogger, by Northern Widget, offers this with their logger package. However, with a few added features an improved logger design could allow for an even lower maintenance more robust data logger. The proposed features would add accessibility through a long range radio network, which would turn the on board SD card into a local backup. The network formed by these radios would support storage on a central unit to allow for real time access to the logged data.

Environmental data loggers are used for a number of reasons, whether its agriculture, environmental protection, measuring seismic activity, etc. Real time data is important for these uses. It allows time for alerts to be sent out in the case where data was captured on an activity that could pose a threat to people in the area. Also with less trips out to the logger's location, it can help scientists take more frequent measurements. The loggers can also inform the deployer that there is a problem that needs to be addressed on one of the loggers. This will lead to more efficient data collection.

## 2 Problem Statement

The current implementation of our data logger does exactly what would be expected: take measurements from sensors and store the information in an SD card. This is fine, but it can be made better by adding radio telemetry between data loggers and a centralized logger that can take all the data from the field and upload it to a server accessible through the internet. There are some closed source loggers available already provide these functionalities such as National Instruments Wireless Senson Networks or Onset HOBO Data Loggers, but they are fairly expensive and provide much more functionality than we need or have a similar cost to the new logger but do not provide some of the specific functionality needed. Not to mention we are trying to make an open source data logger, so other closed source projects will not help us much.

Adding radio telemetry comes with its own set of problems though. The hardware needed is rather power hungry. Along similar lines the data will be accessible through the internet, so the loggers will be left in the field for long periods of time with no one coming to take the data and replace the batteries. With a power hungry device and no new batteries in sight, the power supply will need to be altered. Next, telemetry is not as simple as throwing data out on radio waves and hoping it makes it to the central data server; this is not only insecure and inefficient, but also will result in lost data. In order to ensure that our loggers get all of their data to the server there will need to be a wireless network setup with specific communication protocols to pass the data along to the server. Preferably this network will be able to setup its own paths to the server because setting up a network topology is far more work than the end user should be expected to do. Lastly, the processor used in the

current design of the data logger is almost full with the current code that operates it. In order to fit the code for a network and server communication a new processor with similar peripherals and more memory will be needed on the new logger.

## 3 Proposed Design

### 3.1 Client Requirements

In this section the requirements of the client will be put forth, these are the abstract requirements which must be addressed in a technical manor within the system design

- Transmission range:  $> 1$  km  
This is in order to ensure all nodes have enough transmission range to communicate with their nearest neighbor
- Longevity:  $> 1$  year  
This means that the system must be able to remain active and functional for at least 1 year without operator interference in the field
- Internet connectivity:  
This is a requirement that there is at least one home node which is capable of connecting to the internet and depositing data to a server
- Smart Network:  
This requirement states that a network topology must be implemented which does not need to be setup manually by the user, that a group of nodes be able to setup and maintain a network on their own. This network is needed since there is only a single home node in any group which connects to the internet, as a result all data must be returned through the network to the home node which will then deliver this data to a server

### 3.2 System Requirements

This section describes the required system elements which are required to realize the abstract requirements of the client in a technical manor

- Integrated telemetry radio:  
Since the system requires wireless data transmission, a telemetry radio must be Incorporated into the hardware redesign
- Solar power electronics:  
In order to have the extended lifetime required, a solar panel and rechargeable batteries must be utilized, this requires that a maximum power point tracker (MPPT) be implemented in hardware in order to utilize the solar panel effectively

- Battery charge controller:  
Since rechargeable batteries are to be used, a battery charge controller must be added in order to facilitate proper charging
- 

## 4 Technical Approach

In order to address the problem, a two tiered design approach would be implemented, firstly a new hardware architecture would be developed which integrates the existing features of the system with enhanced processing capabilities, advanced power management systems, and wireless telemetry hardware. On top of this hardware framework, a software wireless network would be developed using a low level software and firmware integration which would allow for streamlined operation

### 4.1 Hardware Improvement

The proposed hardware development would not be starting anew, since there is an already existing hardware architecture that was developed by the client. This architecture would include all features and abilities available through the current architecture, but with the significant additions of a  $915MHz$  radio system, as well as more advanced power management in the form of a solar charge controller and switch-mode power supplies, and an overall improvement in processing capabilities with the use of a more advanced micro-controller which will serve to alleviate the programming bottlenecks currently being experienced by the client.

As noted in Sec. 3, the primary goal of this project is to provide radio telemetry to the existing data logger system. In order to meet the various specifications of cost, transmission range, and power consumption specified by the client, a RFM69  $915MHz$  radio system would be used. This unit provides a long transmission range, a reasonable form factor, and above all, low cost relative to comparable solutions. Another significant benefit of this radio system is that there are already software libraries developed in order to perform many of the required low level operations for network formation, which significantly lowers the cost to implementation.

As noted in Sec. 2, one of the major problems with adding additional power consumptive hardware (such as the proposed telemetry radio) is that the unit is required to be in the field for extended periods without interaction (ie. changing of batteries or charging of the system). Previously the client had used disposable alkaline primary cells to power the system, however it was calculated that the power required to operate the telemetry radio would be too great to be sustained for the required time ( 1 year) with these type of cells alone. As a result, it is proposed to switch to a rechargeable system using high capacity Li-Ion batteries as a storage device and utilizing a photovoltaic array (PV array, or solar array) for constant and sustainable energy harvesting. In order to accomplish this, a solar charge controller will

be introduced into the hardware design. In a similar desire for high power efficiency, the low efficiency power electronics currently utilized by the client will be replaced with high efficiency switch-mode power supplies.

Finally, in order to facilitate the current software requirements and add the additional network layer to the system, a more powerful micro-controller with a larger program memory is required. As a result it is proposed to switch from the current ATMEGA328P micro-controller to the more advanced ARM core based SAM21 micro-controller. This transition would provide for all required network overhead, while also providing for future system development.

## **4.2 Network Implementation**

The proposed self discovery network will need to be written entirely from scratch because all of the software currently associated with the data loggers only handles local storage. Each node in the purposed network will be assigned one three roles: coordinator, router, or leaf node. The coordinator is the node that all other nodes send their data, and it is responsible for uploading the data to the server. A router node is between the coordinator and the leaf nodes; they are responsible for getting the data collected by the leaf nodes and themselves to the coordinator node. Leaf nodes are the furthest nodes form the coordinator, and their sole purpose is collecting data and passing it to a router node or the coordinator node.

The network will be a self discovering network. When setting up the network, each node will go through a block of logic and decide what its role is and perform all necessary tasks. To start, the coordinator, who knows its role from server connection, will broadcast that it is the coordinator. The nodes close enough to hear that broadcast will take note of this, tell the coordinator that they plan to pass data directly to it, and then broadcast that they have coordintor connection. The nodes that hear this new broadcast will under go a similar process. Nodes that get told they will have data sent to them will be router nodes and conversely the nodes that do not receive this message will be leaf nodes. Through this setup process all the nodes in the network will have a dedicated path to the coordinator node.

# **5 Project Management**

## **5.1 Time-line**

Hardware and software groups are going to work in parallel.

On the hardware side, basic functionalities of the radio will be tested with simple Arduino code. New parts used in the solar/battery management system (BMS) are to be selected based on customer's requirements, especially the solar charge controller and switching mode converter. Prototype board of the power section will be designed and built for testing. And if all functionalities are proved to be working smoothly, the PCB will be updated with the new parts.

Meanwhile the software group will work on network implementation. Priorities of various network communication features are determined at first, and then pseudo-code is written to provide an outline of the entire project. Since we are switching to a different micro-controller in the new version of data logger, firmware potentially needs to be changed. And software can be written and tested following the structure shown in the pseudo-code. Finally, the newly implemented code will be integrated with current working code, in order to allow a single set of code working for both versions of data logger.

Timelines for both groups are shown in the tables below. Rearrangements may be made according to unexpected design changes or other limitations.

| <b>Task</b>                         | <b>Start Date</b> | <b>Days to Complete</b> |
|-------------------------------------|-------------------|-------------------------|
| Background Research                 | 1-Feb             | 5                       |
| Radio Testing                       | 6-Feb             | 3                       |
| BMS/Solar Design                    | 6-Feb             | 7                       |
| Spec Parts                          | 13-Feb            | 2                       |
| Prototype BMS/Solar                 | 20-Feb            | 4                       |
| PCB Preliminary Design              | 24-Feb            | 7                       |
| Order Parts                         | 15-Feb            | 1                       |
| Order Preliminary Board             | 1-Mar             | 1                       |
| Populate and Test Preliminary Board | 14-Mar            | 5                       |
| Final PCB Design                    | 9-Apr             | 5                       |
| Final Design Testing                | 24-Apr            | 3                       |
| Final Presentation                  | 5-May             | 1                       |

Table 1: Hardware Timeline

| <b>Task</b>                  | <b>Start Date</b> | <b>Days to Complete</b> |
|------------------------------|-------------------|-------------------------|
| Network requirements         | Completed         | N/A                     |
| Network Priorities           | 1-Feb             | 5                       |
| Network pseudo-code          | 7-Feb             | 5                       |
| Firmware Porting and Testing | 13-Feb            | 3                       |
| Network Code                 | 20-Feb            | 10                      |
| Network Testing              | 7-Mar             | 10                      |
| Code Integration             | 25-Mar            | 5                       |
| Integrated Testing           | 5-Apr             | 5                       |
| Final Presentation           | 5-May             | 1                       |

Table 2: Software Timeline

## 5.2 Budget

| Item Name                           | Purpose     | Cost [\$]    |
|-------------------------------------|-------------|--------------|
| Arduino Zero                        | Prototyping | \$50         |
| Development Parts                   | Prototyping | \$350        |
| PCB Development                     | Prototyping | \$150        |
| Solar Cells                         | Prototyping | \$60         |
| Batteries                           | Prototyping | \$40         |
| <b>Development Total</b>            |             | <b>\$650</b> |
| PCB                                 | Production  | \$50*        |
| Generic Parts                       | Production  | \$30         |
| Radio (RFM69)                       | Production  | \$10         |
| Batteries                           | Production  | \$40         |
| Solar Pannel                        | Production  | \$24         |
| <b>Production Total (per board)</b> |             | <b>\$154</b> |

Table 3: Itemized Budget

## 6 Conclusion

Between more frequent measurement feedback, being able to read those measurements from far away, and not having to disturb the logger and the area around it to get your measurements, adding a radio network to these loggers would be beneficial on all accounts. The data would also be much more accessible to enthusiasts and professions outside of the field of enviromental science.

Our team is more than capable of developing new radio network functionality to the existing design, as well as optimizing the board layout to match the inclusion to keep good battery life and power effeciency. Alternate options for a data logger either fail to provide vital functions or are excessive in price. Our option provides a robust, inexpensive option that efficiently gathers and transmits the data straight to the end user, with little interaction with the physical system beyond setup. This project is a fantastic opportunity to create a practical, automated logging system that can be used to boost the information output without sacrificing a lower price tag.