Data Gator: a Provision-less Network Solution for Collecting Data from Wired and Wireless Sensors

Garrett Wells[1], Mary Everett[1], John Shovic[1]

[1]Department of Computer Science, University of Idaho, Moscow, ID, USA

A paper from the Proceedings of the
16th International Conference on Precision Agriculture
21-24 July 2024
Manhattan, Kansas, United States

Abstract.

Advances in wireless sensor technology and data collection in precision agriculture enable farmers and researchers to understand operational and environmental dynamics. These advances allow the tracking of water usage, temperature variation, soil pH, humidity, sunlight penetration, and other factors which are crucial for trend prediction and analysis. Capitalizing on this advancement, however, requires data collection infrastructure using large and varied sensor networks. Adoption and implementation of such infrastructure can be daunting to operators and researchers because it requires research, installation, monitoring, and maintenance of multiple hardware systems as well as visualization software. Currently IoT companies focused on precision agriculture develop microcontroller based IoT devices which support sensors for a single application such as climate monitoring. To promote adoption of wireless sensor networks in precision agriculture a new microcontroller hardware platform is needed. The new system should be based on open-source hardware and software so that users have complete control over the implementation and configuration of their system. Furthermore, the hardware should support the wired and wireless protocols used by common sensors so that adapting the network, to incorporate water management with climate monitoring for example, merely constitutes adapting the firmware to support required sensors. The Data Gator project models the implementation of an affordable, open source, microcontroller-based network solution with the adaptability to meet the data collection needs of most applications by supporting wired and wireless protocols and common yet high-quality sensors. Energy efficient sleep modes and solar power capability allow the Data Gator to operate sustainably year-round while logging data to local or cloud storage. A simple code-less configuration system allows modification by the end user while provided electrical design files and documentation enable contracting scalable manufacturing and assembly through third-party companies. Design validation has been conducted with twelve Data Gators installed at Laurel Grove Wine Farm since spring 2023 where they are distributed to read data from approximately 150 sensors every five minutes and

maintained remotely with Over-The-Air firmware updates. A second installation at the Sandpoint Organic Agriculture Center has been running since July 2023 and a third installation is planned for the Coeur d'Alene USDA Forest Service Nursery in spring 2024. This project demonstrates a solution and provides resources for further development of sophisticated hardware capable of integrating a variety of sensors in a variety of agricultural scenarios, including forestry, agriculture, aquaculture, viticulture, and others.

Keywords.
IoT, Wireless sensor network, Sensor Network, Over-the-Air Update, Microcontroller

Introduction

Agriculture spans an incredibly broad range of applications with differing environmental conditions and requirements for success. An increasing world population has applied pressure to the agricultural community to increase production and efficiency while adapting to global climate change. Even so, agricultural production has increased, despite the area of agricultural land in use remaining static (Taylor et al. [1]).

Thus, improvement in agricultural practices is a key driver of production increases. A key area of advancement is precision agriculture (PA), defined by the International Society for Precision Agriculture as "... a management strategy that gathers, processes and analyzes temporal, spatial and individual plant and animal data and combines it with other information to support management decisions according to estimated variability for improved resource use efficiency, productivity, quality, profitability and sustainability of agricultural production." Lowenberg-DeBoer et al. [2], has found a steady increase in usage of PA technology such as GNSS and variable rate technology (VRT), over the past two decades across various studies. An aspect of growth in PA not reflected in this study, however, is the use of Wireless Sensor Networks (WSNs) to collect data to support PA operations.

Unlike GNSS and VRT which are automation technologies integrated into key agricultural machinery such as harvesters and sprayers, to increase the granularity of management and plant care, WSNs are a method for collecting data to guide resource management. This data can be used to monitor trends over time and guide care of plants with greater granularity moving closer to the ability to provide optimal care on a per-plant basis. Use of WSNs has the ability to provide long-term data collection with minimal management by a human technician.

Setting up and configuring a WSN for operation, a process referred to as "provisioning" can be labor intensive. Provisioning usually requires the association of hardware with a unique identifier, geo-location, or the manual entry of network credentials to allow data to be logged through the network. For networks including multiple types of sensors, provisioning may also require registering individual sensors with the network as many sensors do not have unique identifiers sufficient for data association.

The combination of hardware and software design executed in the Data Gator makes it possible to reduce or eliminate much of the work involved in provisioning a WSN installation using the SCARECRO architecture. The Data Gator connects new sensors to the network using firmware to detect them and automatically begins collecting data without having to "provision". All connected sensors are associated with unique identifiers and data is timestamped to provide trend tracking.

WSN Architecture

WSNs vary between use cases, and no single set of sensors fits every application. For example, in conversations with viticulturists, WSNs provided value through mapping humidity and temperature to guide planting of new vines. For orchards, a canopy light sensor network provides insight into pruning practices and individual tree care. In aquaculture, WSN monitoring of water temperature, pH, total dissolved solids (TDAS), and dissolved oxygen (DO) has been demonstrated by Shi *et al.* [3], and Malik *et al.* [4].

Though the sensors used differ, in many instances the WSN architecture and node hardware investigated is functionally equivalent, consisting of a power efficient microcontroller(μC), wireless data transmission modules, and modules for providing connections to wired and wireless sensors. Hardware sensor support (analog, I^2C, SDI, SPI, etc.) and the backbone wireless technology used to transfer data through the network determine the key limitations of the system. These limitations can be seen as limitations of data rate (bandwidth), transmission range, and types of data available. Other characteristics of the system can be changed through software configuration.

Although many different schemes for network architecture have been proposed, Figure 1 is the

most common implementation and can be easily adapted to a broad range of applications.

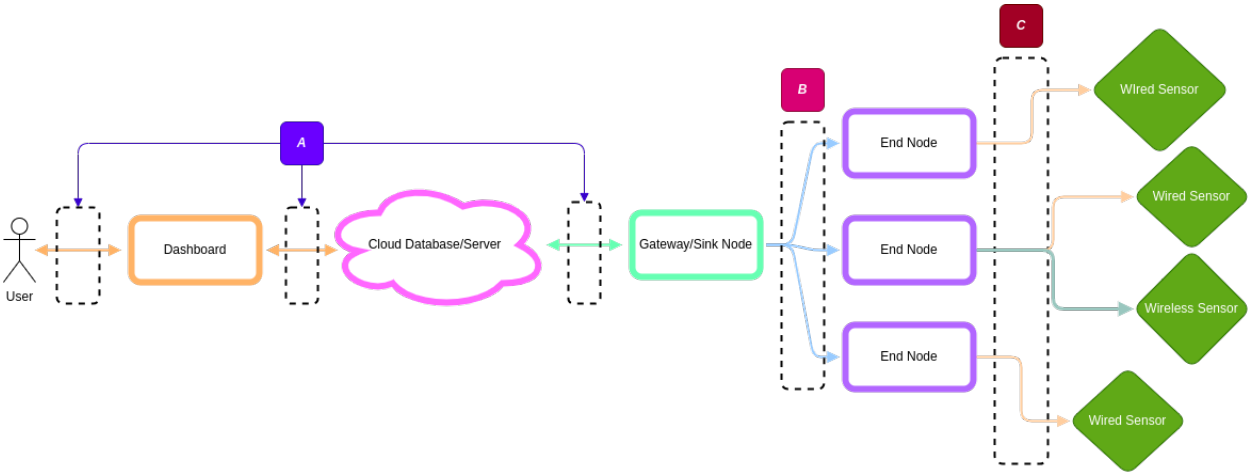

Figure 1. WSN Architecture

In Figure 1, boxes A, B, and C indicate key data transmission methods between layers of the WSN architecture. While connections between dashboard, server, and Gateway/sink node (A) are often Wi-Fi connected, the connection between sink and end node (B) may range from a high bandwidth protocol (such as Wi-Fi) to a protocol capable of long-range transmissions at a lower bandwidth (LoRa). Selections made here significantly impact reliability, power efficiency, and configuration of the network. The connection between nodes and sensors is determined by hardware availability and the technology available in the market. This is demonstrated by the increasing availability of BLE and LoRa wireless versions of common sensor types such as temperature sensors.

The foundational hardware to implement the Wi-Fi connected devices in Figure 1 has long been available to purchase off the shelf, meaning that user solutions can be tailored to meet specifications in software and that further progress requires refinement through software. Gateway (sink node) and end nodes of WSNs present a more difficult problem currently. Studies have proposed and demonstrated the viability of simple hardware designs which are power efficient and support a variety of wireless communication technologies suited to the moderately low data transmission rates prevalent in precision agriculture (see [5], [6], [7], [8], [9],[10], [11]). Common features among investigated designs include:

- Wireless Support (Wi-Fi, BLE, Zigbee, LoRa, etc.)
- Solar Power
- Offline Data Logging Capability
- Wired Sensor Support (Analog, I^2C)

Availability and adoption of such node solutions continues to be a problem, as few off-the-shelf options for node hardware platforms are available. Available solutions are often expensive, proprietary systems with limited selection of supported sensors, limited solar charging support, and limited or no option to adapt the system to offline data logging. Thus, researchers and adopters of WSN technology continue to explore custom node solutions, indicating the need for a systematic node hardware and software solution which would either provide or allow the easy incorporation of desired features.

WSN Node Modularity

Hardware

WSN node design has a large degree of variance in the detail of hardware design, yet all or some of the core features above can be identified in every node design. Since the implementation of such features in hardware and software varies from application to application, it is helpful to refer

to each of the core features above as “modules.” Figure 2 provides a diagram of these modules and their components for reference.

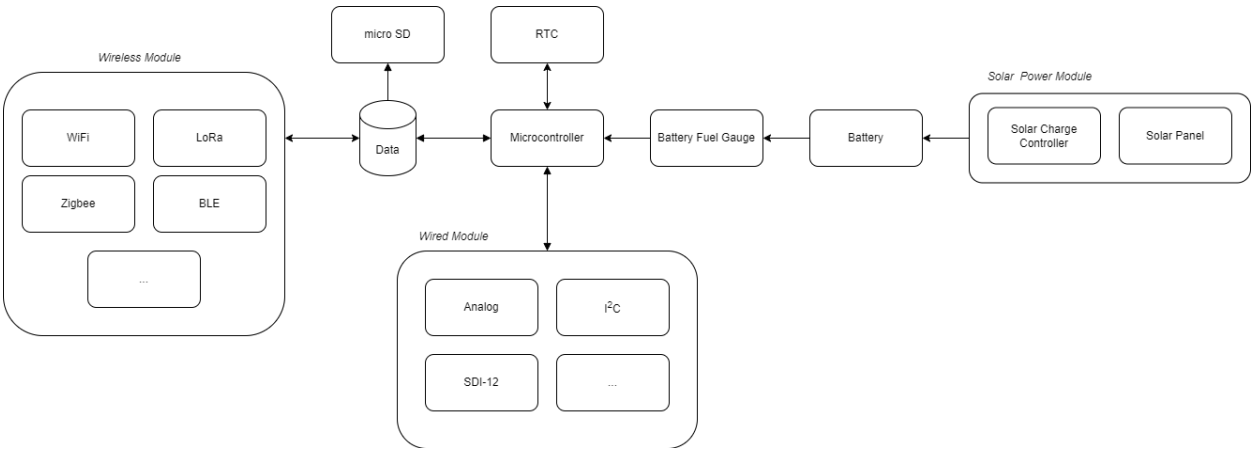

Figure 2. WSN Node Feature Modules

Modules may be implemented as part of a single PCB, or multiple connected PCBs. For example, Sophocleous *et al.* [10] are notable for their modular, stackable PCB design which allows adaptation between Wi-Fi and LoRa by combining different PCBs in a two-board stack. There are significant benefits to such modular PCB design as it: (a) allows a single “minimalist” µC node to implementation on the main PCB and (b) allows interchangeable modules (such as wireless or solar) to support various network configurations and applications. An approach centered around modular hardware would make large scale production possible as well as reduce the base power consumption of the device as high-power modules could be removed when not needed.

Minimal Node Description

For the purposes of this paper, a minimal node is a µC on a printed circuit board (PCB) with µSD holder, Analog-to-Digital Converter (ADC) with ports, I^2C ports, capacity to support (or integrated) wireless module, and capacity to connect to a solar charging system. Ports refer to any method of connecting a wired sensor or module to the node, but it is suggested that these connections: (a) have crimped wire connectors to allow easy connection of new sensors and (b) some form of lock or latching mechanism to ensure secure connection in rough environments.

The node design should also provide power switching capability to minimize passive power drainage when not actively taking readings from sensors. This is a key feature as individual wired sensors may passively draw 10 mA or more unless disconnected from the power supply.

Software

Software and firmware play a vital role in node functionality. Despite this, discussion and study of these topics is not common in literature. Mesas-Carrascosa *et al.* [7] detail an implementation based on the Arduino hardware and software (C++) platform in combination with an Android application developed in Java. Sophocleous *et al.* [10], provide even less software description, merely noting that the system was developed for an ESP32 µC. Morais *et al.* [9] emphasize the use of Arduino based systems with a lower number of µCs from other manufacturers. John *et al.* [12] outlines a simple 8 step algorithm describing node software execution for mesh network operation. However, further documentation of the software implemented is excluded.

Though the required software and firmware for a functional node can be quite simple, often not meriting discussion, intentional software *architecture* can reduce the effort required to add support for new hardware platforms and therefore becomes a tool for accelerating technology adoption. One of the motivations of this project is to provide a modular software implementation improving flexibility and configuration for the user.

Despite the potential to leverage software to reduce the barriers to adopt innovative technology

and accelerate development, such software complexity has the potential to drastically increase node cost through software development time. Therefore, an open-source architecture and design platform has the potential to mitigate cost increase while advancing the capability of such node designs. Systematic, modular hardware platforms paralleled by open-source modular software and firmware have the potential to provide a holistic node system. With sufficient support and documentation such a system could allow users to build WSNs based on a common node design standard or “language” much like those used in industrial automation. A systematic approach such as this could provide better access to PA methods across the breadth of agricultural use cases and beyond.

Research Objective

Here we present an open-source design which demonstrates the ability to create a modular, economic, and scalable solution through thoughtful hardware and software design. It is hoped that the creation and publication of this system will serve as a guide for the refinement of systematic WSN node creation within PA application, increasing opportunities for adoption while lowering the required level of technical expertise. Through publishing documentation, guides building the node, a software tool chain, and all source design files it is hoped that this project may become an example of both node design and effective management for remote applications.

Design and Implementation

Discussion of the PCB design methods is divided into six sections starting with an overview of the design, discussion of the fabrication and assembly methods, budget, documentation, support, and finally areas for future development.

Design

The core components of the Data Gator main PCB include a Watch Dog timer, power cut-off switches, a µSD holder, ESP32 µC with Wi-Fi and Bluetooth module, and 12-bit ADC (see Figure 3). These components provide the basis for minimal node functionality and provide the hardware interfaces to support the key functional modules for WSN nodes.

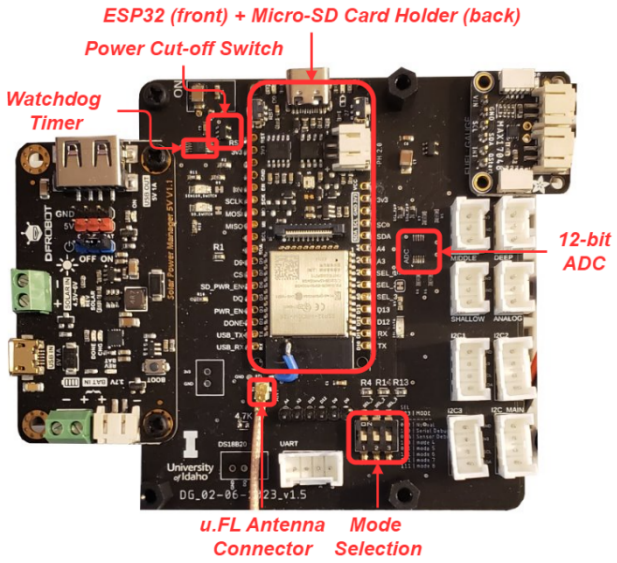

Figure 3. Main PCB Components

In this case, the wireless module is integrated with the µC, which is one of the strengths of the Espressif ESP32 µC platform. The Data Gator main board augments the wireless module by providing a u.FL connector so that an external antenna may be attached with a simple modification to increase signal strength and range. This feature helps mitigate the impact of placing the Data Gator in a waterproof enclosure, which may interfere with signal transmission.

The Data Gator board supports several common wired interfaces as shown in Figure 4, including four 3.3 V analog sensor ports, four I^2C ports, and a serial interface port. Solar charging support is provided through the additional Solar Charger PCB and Fuel Gauge. These boards are available for purchase from DFRobot and Adafruit, third-party PCB vendors.

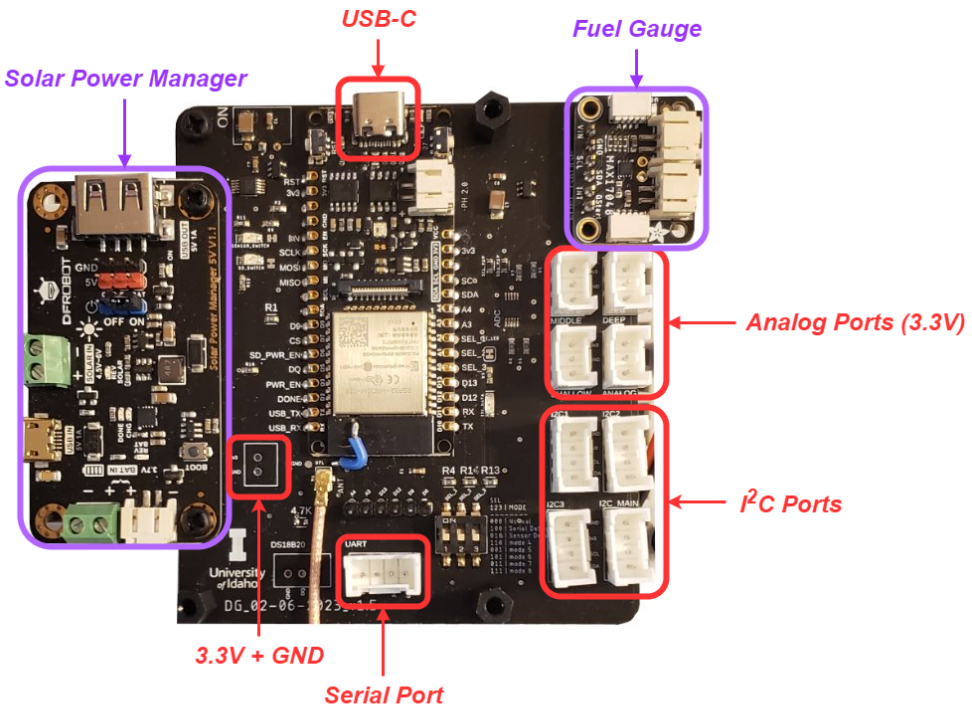

Figure 4. Data Gator Ports and Daughter Boards

Fabrication and Assembly

Fabrication and assembly of embedded devices such as the Data Gator can represent a significant challenge, requiring considerable time, knowledge, and skill to complete. Fabrication refers to the process of creating the multi-layered PCB which components will later be soldered to during assembly. This is completed by a fabrication house with specialized equipment, meaning that the designer must submit their PCB design files to the fabricator, have them vetted, and then wait for the boards to be made. Most fabrication firms are only willing to produce large runs of hundreds or thousands of PCBs, though a smaller subset of these companies provide small scale prototyping services, allowing customers to purchase PCBs in scalable batches with a minimum run size of 1 to 5 boards.

The next step is assembly. During assembly components are attached to the PCB and may be offered as an add-on service by the fabrication house. This service is extremely valuable for customers who do not have sufficient equipment to assemble components themselves but may increase the cost of each board and constrain component selection. The emergence of limited run prototyping services with quick delivery times is potentially significant to researchers seeking to create and install a custom WSNs as it is now possible to have a node manufactured at scale, reducing time and capital investment for budget constrained research projects. For

other technology adopters in agriculture, it also becomes possible to purchase manufacturing services for open-source hardware projects with limited hardware experience, reducing the expertise required to take advantage of open-source resources. To elaborate on the benefit of such a service, we will compare the process and equipment of assembling the Data Gator through a 3rd party as opposed to self-assembly.

Self-Assembly

It is difficult to give an accurate representation of the exact equipment required for assembling a PCB as there are design considerations which may reduce the equipment needed. For example, a PCB designed using only through-hole technology (THT) requires a mere soldering iron (with appropriate tips), vise to hold the PCB, solder, and a minimal amount of rework equipment. This equipment may be purchased for around $100, depending on the quality of equipment selected and is therefore the most economical option.

THT components are unfortunately quite large, and a board designed around them will be quite large. Increasing size increases cost. Everything from the cost of manufacturing the PCB to the cost of a weatherproof enclosure for the system. To reduce size, and cost, using surface mount technology (SMT) is imperative. SMT components are smaller and more affordable than their THT equivalents and allow components to be placed on both sides of a PCB. The equipment required to use SMT is much more complex and expensive. Assembly of the Data Gator required the following equipment:

- Reflow Oven
- Microscope
- Hot Air Gun
- Soldering Iron
- PCB Printing Screen
- PCB Mask
- Tweezers, Solder Pump, etc.

This is not an exhaustive list, as there are various other consumables such as solder paste, solder wick, and solder flux yet it shows that the investment both monetary and experiential required for an efficient assembly system is substantial. Assembly of an individual Data Gator required 90 minutes of work. To a limited degree, several boards can be built at the same time, making the process more efficient, but the time investment is still substantial for anyone building multiple boards.

Assembly through the fabrication house is far more time efficient as fabrication houses offer assembly of all components on the board except for through-hole components. During this project, three companies were used to produce boards for testing and deployment at the test sites. Of the companies, JLCPCB and PCBWay are Chinese companies while the third, OSH Park, is located in the USA. Each company offers fabrication services, but of the three, OSH Park is the only one which does *not* provide an add-on assembly service for surface-mount components.

Use of the add-on assembly service provides several benefits:

- Reduced equipment required,
- Man-hours (expensive), can be focused on development and testing,
- Devices are assembled using components from the service provider's inventory.

It is extremely difficult to minimize cost for each device when performing self-assembly and invest the further effort and time required to plan and maintain inventory of components and supplies required for board assembly.

Budget

The Data Gator is a low-cost WSN node option; the hardware cost to produce one device is shown in the table below.

Table 1. Data Gator Wired PCB Interfaces

Item	Cost
PCB w/ Components	$16.78
ESP32 (DFRobot Firebeetle2)	$8.90
Micro-SD Card	$6.00
Connectors	$3.00
Fuel Gauge	$5.95
Total (minimal)	**$40.72**
Battery	$24.50
Total (minimal usable)	**$65.22**
Solar Charge Manager & Panel	$25.90
Total	**$91.12**

The Data Gator is still in the development stage and the costs of certain components above reflect that. For example, although the µC used is a variant of an ESP32 module, the team chose to integrate a DFRobot Firebeetle2 micro-controller PCB which is represented on the ESP32 line above. This decision utilizes the castellated vias of the Firebeetle2 to provide GPIO access while removing the need to provide a USB bus interface, simplifying the design. This choice unfortunately increases the cost of the Data Gator and updating the hardware design to use a cheaper ESP32 module and USB interface is a potential method for reducing board cost. Doing so, however, would increase reliance on 3rd-party assembly services.

Documentation and Support

Documentation for Data Gator PCB is hosted through a GitHub organization (https://github.com/Data-Gator/Hardware). All design files required to order a Data Gator from JLCPCB or PCBWay are provided, including complete lists of components used, guides for ordering from both companies, and additional guides for purchasing and connecting supported sensors.

Areas for Future Development

Future development for the Data Gator should focus on three areas: support for additional wired sensor protocols, support for wireless protocols as modules, the addition of an RTC, and transition from using the DFRobot Firebeetle2 ESP32 board to an ESP32 module. Adding an RTC module would allow accurate time keeping between resets while switching to an ESP32 module would reduce the µC cost per Data Gator by nearly 50%. Most important is adding support for the wired protocol SDI-12 and support for LoRa as a low-cost, low-power option for data logging.

Software Design

Software for the Data Gator was written in C++ using the PlatformIO embedded software development platform. PlatformIO provides a build platform used as a key part of the provisioning of the Data Gator before installation. The Arduino C++ framework was chosen for optimal software compatibility, as this framework provides a common API supporting a broad range of µCs from various sources.

A modular architecture was used to divide functionality into logical units which can be modified to

customize execution easily with minimal modification of unrelated code. Key modules of the software are highlighted in Figure 5. Code-less provisioning is accomplished through configuration profiles written in Python's human readable INI file format. A Python automation script integrated into PlatformIO's build system then generates `config.hpp` from a user selected configuration profile. This header file is then compiled to produce a binary image which can be flashed to every Data Gator in the WSN, allowing users to provision a WSN with the creation of a single configuration file, and later easily modify sensor polling rates, network credentials, and other parameters.

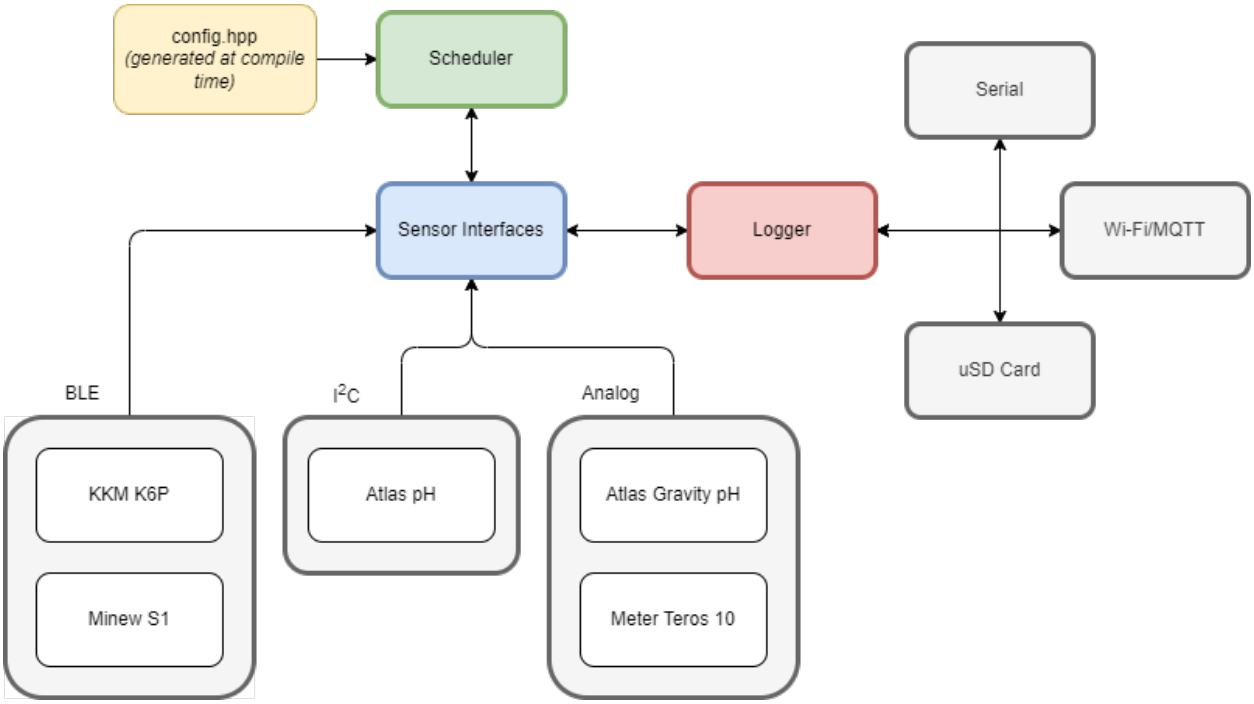

Figure 5. Data Gator Firmware Modules

The pattern of execution is shown in Figure 6.

```
// on wake from sleep
increment_relative_clock();
detect_connected_peripherals();
initialize_logging_module();
read_config_params();
update_scheduler();
execute_scheduler_marked_tasks();
log_data();
sleep(64s);
```

Figure 6. Data Gator Firmware Execution Pattern

The Data Gator lacks a hardware real-time-clock (RTC) which would allow the device to track time while asleep, thus scheduled tasks, such as reading from the VWC sensors are tracked using a relative timing method based on the watch-dog timer (WDT) which triggers a device reset every 64 seconds if not fed. Thus, maximum resolution on sensor measurements is set to approximately one minute, which is appropriate for many use cases as applications seeking higher resolution must provide for large power consumption and are not likely to sleep between sensor readings.

When the Data Gator has access to a Wi-Fi network, it will automatically acquire an “absolute” time stamp using a network time protocol (NTP) server. In cases when an NTP server cannot be reached, the last absolute timestamp is used, and the number of device resets is added to the last absolute timestamp to provide an approximation of the time before the data is logged to the µSD card. Thus, when the device is reconnected to the network, or a human user is able to access the device the data may be retrieved.

The default method for logging data is over Wi-Fi using the MQTT protocol, using the topic message field to record node ID and the sensor type. The message body contains fields such as the sensor brand, model, raw sensor reading (such as analog voltage), scaled sensor reading, and other sensor parameters if available. Sensor readings are logged in JSON text files to the µSD card and can be retrieved physically or re-transmitted in MQTT messages as needed.

Tasks are scheduled by the hardware module they are connected to, meaning that all sensors connected to that interface are read at the same time. The rate of polling, in minutes, may be set by the user through the text configuration file before the firmware image is compiled and uploaded to the device. This selection is based on several factors, though the primary reason is power. To read a single sensor from the I^2C bus requires that power be supplied to *all* sensors on the bus, as power switching is provided at an interface level, not an individual sensor level. Therefore, it is most efficient to read data from all sensors at the same time. This is especially true for cases when the sensors require a "warm-up" period while the sensor initializes and prepares to take a reading.

Remote maintenance of the Data Gator is provided through over-the-air (OTA) updates which allow new firmware images to be flashed to individual devices over a Wi-Fi connection. This feature was used with great success to manage the test installation at Laurel Grove remotely, allowing new sensors to be tested and integrated into the firmware before the sensors were added in the field by the client.

Areas for Future Development

The primary area for feature development is the addition of configuration loading from the µSD card. The current configuration method only allows setting parameters at compile time. Although this is preferable for the efficient installation of a network of nodes all operating with similar sensors and polling rates, it complicates modifying individual node operation post installation. Allowing µSD stored configuration options to override compiled options would open the possibility for many user-friendly features.

Documentation

Much attention was given to the development of documentation for this project to help make it accessible and usable. An overview of this aspect of the Data Gator is provided here to show the development of resources for other projects and how they can be used to support open-source projects.

GitHub

GitHub is a version control solution which also hosts projects and provides a variety of other free services to open-source projects. These same features are available for proprietary development, albeit behind a paid service tier. GitHub is used by this project to host source code, documentation, and design files.

Visitors to the page (https://github.com/Data-Gator) will find links to documentation for users, developers, and an overview of the SCARECRO architecture which the Data Gator is designed to be incorporated into. User documentation focuses on describing how to use PlatformIO to compile a fresh binary and upload it to a Data Gator board. Developer documentation documents workflows and processes for developing new firmware modules to support new sensors. Additionally, documentation for the code-less configuration system is provided to support provisioning at compilation.

The final component of the GitHub documentation provided is the hardware documentation. This outlines how to connect wired sensors to the Data Gator assuming a sensor straight from the manufacturer without crimped connectors. Additionally, the component lists, board design files, and automation files required for 3rd party assembly are provided with user guides for buying boards from several manufacturers.

Doxygen

All project firmware comments and source documentation are written according to the Doxygen style guides. The HTML documentation generated by Doxygen from the source code is also hosted as part of the GitHub documentation, giving access to detailed, searchable documentation of all software deployed on the Data Gator.

Evaluation

The Data Gator has been tested with a simple benchtop system as well as several test installations. Here, a summary of what was learned from the various tests is described along with a description of the process used to conduct the testing. Results from several test locations will be examined including an installation at a future vineyard and an organic orchard.

Benchtop Testing

The Data Gator was evaluated first using a benchtop testing setup consisting of a RaspberryPi using an INA219 integrated circuit chip to measure the power supplied to the Data Gator through a USB port on the RaspberryPi. The current draw was measured over approximately 45 minutes of normal operation with three analog Meter Teros 10 VWC sensors connected, each drawing approximately 20 mA. The measured system consumption is listed in Table 2.

Table 2. Data Gator Power Draw With 5V Supply

Measurement	Current (mA)
Max (data transmission)	200
Average	14
Est. Battery Life (6600 mAh)	~19 days

This evaluation allows a simple estimation of battery life which can be used to design a solar charging system for the Data Gator. This is of limited usefulness, however, as this measurement is highly dependent on a variety of factors including: the number (and type) of connected sensors, and the configured polling frequency. As shown in the Laurel Grove test installation, even robust solar systems are prone to occasional outages, though quick recovery by the system is usually possible without intervention.

Laurel Grove

Laurel Grove is a future vineyard located in Winchester, Virginia. Twelve Data Gators were installed, each connected to three volumetric water content (VWC) sensors, and a variety of BLE temperature and humidity sensors. A small number of the Data Gators were also connected to a soil pH sensor. Each device was powered by a solar system consisting of a 2 W solar panel and 6600 mAh lithium-ion battery. These are the components reported in Figure 3 and Table 2 for cost and performance estimates. Real world battery charge levels were recorded and can be seen in Figure 7 for comparison with the estimates. Solar system components were chosen with the intention that the device would be able to operate on an average of 3 hours of full sunlight each day and have enough battery capacity to maintain operation during cloudy periods of at least a week. These parameters were chosen based on the average number of sunny days per year for the region using estimates from internet sources and assuming around 3 sunny days per week.

It is hoped that this system would be capable of operating in most locations in the USA as well as other comparable climates. Each Data Gator was mounted on a pole, approximately four feet above the ground in a weather sealed box with the solar panel mounted on top of the box facing south at approximately a 45° angle from the horizontal to provide optimal sun exposure. Despite the repeatable mounting configuration, not all devices received identical sun exposure, and this is thought to play a role in the performance recorded.

Several observations should be made of the data provided in **Error! Reference source not found.**. The first is that though the data spans a period of more than a year, there is a period during the fall of 2023 when the Data Gators were removed from the field and no data was collected. Data collection resumed in December and continued through the date of this paper's writing in May of 2024. The second observation is that the charge percentage of the batteries is not perfectly calibrated, and some batteries reported maximum charge levels of over 100%. This is of marginal importance, as the greater value in this data is the ability to track the relative change in charge level over time and how the system recovers from a low battery state. Therefore, the third observation is that although approximately half of the devices fully depleted their battery, or "crashed", they all recharged within one day without assistance and did not continue to experience further crashes.

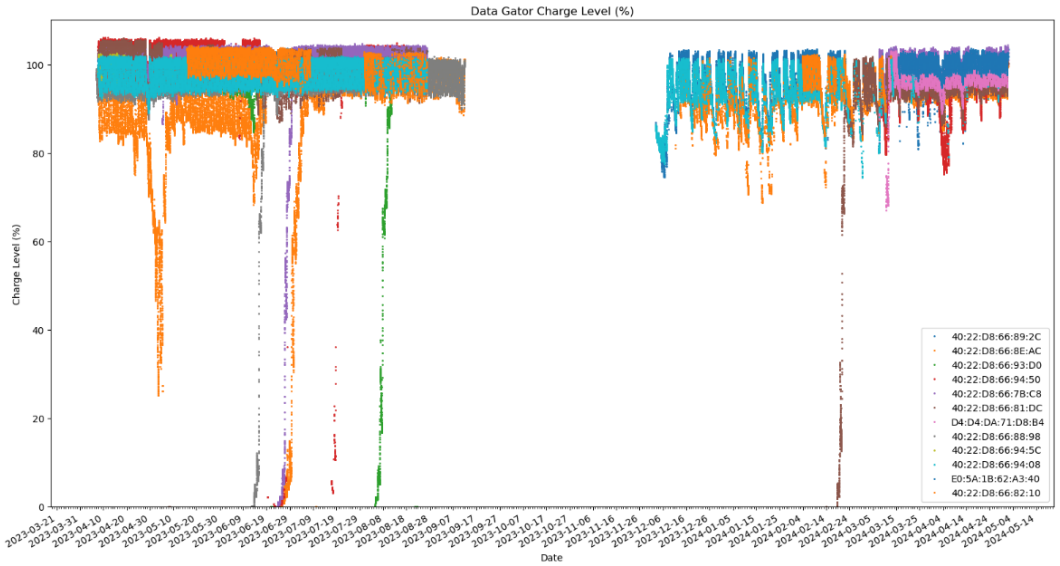

Figure 7. Laurel Grove Data Gator Battery Charge Level

The data shown in **Error! Reference source not found.** shows that in general the charge level of each device varies by 10% per day, allowing an estimated battery life of about 10 days without recharging. During the early spring months, the charge level variance noticeably increased and periodically failed to recover with the same degree of consistency, taking a day or two to fully recharge. Fewer days with full sun exposure may explain the change in behavior, but the true reason is not known. Despite this, fewer complete battery discharges were observed.

SOAC

A second test installation was located at the Sandpoint Organic Agricultural Center (SOAC) in Sandpoint, Idaho. This installation was much smaller and consisted of one Data Gator which operated at the location for approximately three months. Figure 8 shows that the test device operated reliably, but for an unknown reason, began to lose charge slowly but continuously from September onwards.

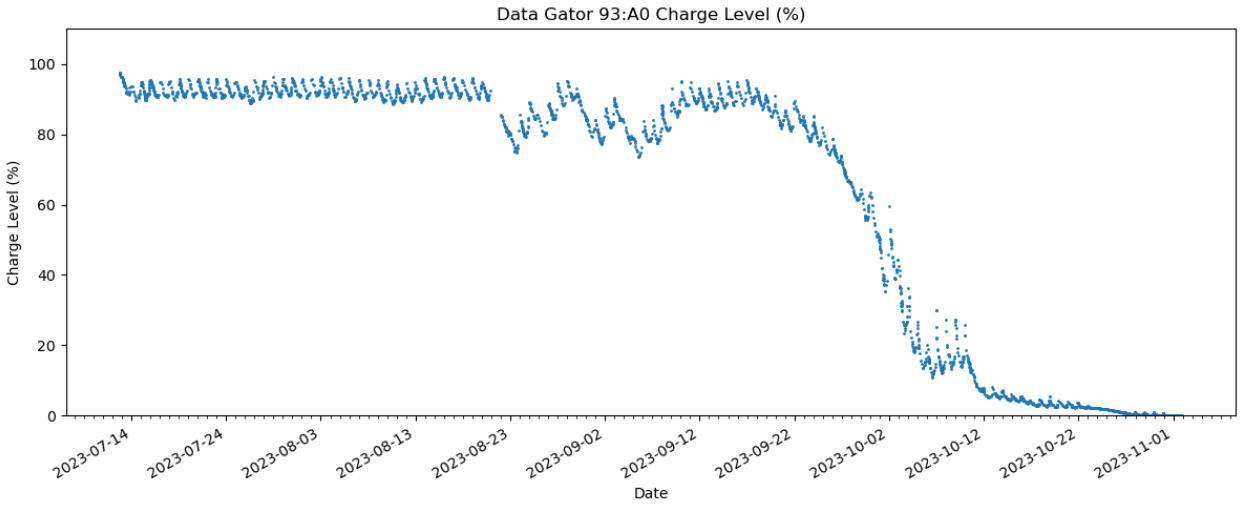

Figure 8. SOAC Test Installation

Although the charge level continued to decrease, it did so over the period of almost two months and continued to operate at levels below 20% for about a month. This is interesting since it deviates from the behavior of the Laurel Grove devices, which discharged at a much faster rate, yet also recovered unassisted. Laurel Grove devices were polling from more sensors, wired and wireless, which may partially explain a faster drain rate. Another possible explanation is a decrease in battery health and maximum recharge capacity, for the SOAC device, leading to an inability to fully recharge the battery. Unfortunately, without another device to compare against or more data to help explain the reason for this failure, it is impossible to draw a conclusion as to the significance of the data.

Conclusion

The evolving landscape of PA reveals new perspectives and potential approaches for resource management. Effective management relies on information to understand current needs and predict critical action steps for future operation. WSNs provide a valuable way to acquire data which makes efficient management of increasingly large agricultural systems possible.

Access to power and economical internet connection in agrarian communities makes the adoption of WSNs for data collection possible. Adoption of the technology is not likely, however, unless the hurdles currently facing users are not lowered or removed. Among these hurdles is the difficulty of provisioning WSNs, which the system proposed in this project addresses using a combination of hardware and software design choices. The system makes it possible to connect the supported sensors and begin recording data by simply providing a firmware image with network SSID and password. Supplied open-source hardware and software documentation as well as firmware build system seek to make the technology accessible for researchers and other users.

Real-world testing of the Data Gator over a year of operation across several locations has shown that the Data Gator provides robust and reliable data collection. The experiences provided through this test period show that the Data Gator can be installed outdoors for long periods of time with a high degree of confidence that the device will run for a reasonable amount of time while relying on solar power. Thus, the proposed design meets the general hardware requirements for a WSN node while effectively demonstrating several strategies for reducing the hurdles faced by those attempting to use the technology in PA applications. This is highly encouraging and demonstrates several strategies for open-source projects to produce embedded systems which are easily usable and effective.

Acknowledgments

Thanks to Dustin and Jaclyn Mommen of Laurel Grove Wine Farm for supporting the research conducted in this paper.

References

[1] C. A. Taylor and J. Rising, "Tipping point dynamics in global land use," *Environ. Res. Lett.*, vol. 16, no. 12, p. 125012, Dec. 2021, doi: 10.1088/1748-9326/ac3c6d.

[2] J. Lowenberg-DeBoer and B. Erickson, "Setting the Record Straight on Precision Agriculture Adoption," *Agron. J.*, vol. 111, no. 4, pp. 1552–1569, 2019, doi: 10.2134/agronj2018.12.0779.

[3] B. Shi, V. Sreeram, D. Zhao, S. Duan, and J. Jiang, "A wireless sensor network-based monitoring system for freshwater fishpond aquaculture," *Biosyst. Eng.*, vol. 172, pp. 57–66, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.016.

[4] P. K. Malik, P. Malik, G. R. Kumar, Sneha, R. Abraham, and R. Singh, "Design and Implementation of a LoRa-Based Water Quality Monitoring System," in *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, Nov. 2023, pp. 120–124. doi: 10.1109/AECE59614.2023.10428618.

[5] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail, "Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review," *Sensors*, vol. 17, no. 8, Art. no. 8, Aug. 2017, doi: 10.3390/s17081781.

[6] R. K. Math and N. V. Dharwadkar, "A wireless sensor network based low cost and energy efficient frame work for precision agriculture," in *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, Jan. 2017, pp. 1–6. doi: 10.1109/ICNTE.2017.7947883.

[7] F. J. Mesas-Carrascosa, D. Verdú Santano, J. E. Meroño, M. Sánchez de la Orden, and A. García-Ferrer, "Open source hardware to monitor environmental parameters in precision agriculture," *Biosyst. Eng.*, vol. 137, pp. 73–83, Sep. 2015, doi: 10.1016/j.biosystemseng.2015.07.005.

[8] A. Mittal, K. P. Chetan, S. Jayaraman, B. G. Jagyasi, A. Pande, and P. Balamuralidhar, "mKRISHI wireless sensor network platform for precision agriculture," in *2012 Sixth International Conference on Sensing Technology (ICST)*, Dec. 2012, pp. 623–629. doi: 10.1109/ICSensT.2012.6461755.

[9] R. Morais, J. Mendes, R. Silva, N. Silva, J. J. Sousa, and E. Peres, "A Versatile, Low-Power and Low-Cost IoT Device for Field Data Gathering in Precision Agriculture Practices," *Agriculture*, vol. 11, no. 7, Art. no. 7, Jul. 2021, doi: 10.3390/agriculture11070619.

[10] M. Sophocleous, A. Karkotis, and J. Georgiou, "A Versatile, Stand-Alone, In-Field Sensor Node for Implementation in Precision Agriculture," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 11, no. 3, pp. 449–457, Sep. 2021, doi: 10.1109/JETCAS.2021.3099112.

[11] M. M. Maha, S. Bhuiyan, and M. Masuduzzaman, "Smart Board for Precision Farming Using Wireless Sensor Network," in *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, Jan. 2019, pp. 445–450. doi: 10.1109/ICREST.2019.8644215.

[12] G. E. John, "A low cost wireless sensor network for precision agriculture," in *2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*, Dec. 2016, pp. 24–27. doi: 10.1109/ISED.2016.7977048.