2019

# Development of an Open-source, Custom Environmental Data Logger for Spatially Scalable Data Collection

Tim Anderson
*University Of Montana*

# DEVELOPMENT OF AN OPEN-SOURCE, CUSTOM ENVIRONMENTAL DATA LOGGER FOR SPATIALLY SCALABLE DATA COLLECTIONS

By

Tim Anderson

Bachelor of Music, The University of Montana, Missoula, MT, 2012

Thesis

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Computer Science

The University of Montana
Missoula, MT

Autumn 2018

Approved by:

Scott Whittenburg Ph.D., Dean
Graduate School

Travis Wheeler Ph.D., Chair
Computer Science

Kelsey Jencso Ph.D.
Watershed Hydrologist

Rob Smith Ph.D.
Computer Science

Anderson, Tim R., M.S., January 2019 Computer Science

Development of an Open-source, Custom Environmental Data Logger for Spatially Scalable Data Collection

Chairperson: Travis Wheeler

Characterizing the processes that lead to differences in ecosystem productivity and watershed hydrology across complex terrain remains a challenge. This difficulty can be partially attributed to the cost of installing networks of proprietary data loggers that monitor differences in the biophysical factors contributing to vegetation growth or hydrological processes. Studies that aim to compare concurrent time-series data sets across multiple locations must therefore balance the high cost of these data logger systems with the need for spatial resolution in their data. Here, we present the design, implementation, and case study for an open-source "Pinecone" data logger system, released under the GNU General Public License that can be manufactured for under $70 USD per unit. The system was designed to accommodate a wide range of generic and proprietary environmental sensors, and to be inexpensive enough to build and deploy large numbers to a study site. A case study was performed in which 54 data loggers were deployed to North Fork Elk Creek, a mountainous watershed located in Lubrecht Experimental Forest in the Garnet mountain range in Northwest Montana for a one year period. The data loggers were deployed across 6 hillsides in the watershed, representing combinations of differing elevations and aspects, at 9 study locations on each hillslope. At each of these locations we recorded air temperature, vapor pressure, soil water content, sap flow velocity, and tree basal area at 30 minute intervals. We evaluated the reliability of the systems in a case study over an 8 month period in 2016 and 4 month period in 2017. Our results suggest that open-source technologies such as the Pinecone logger can make it possible to develop dependable and spatially distributed sensor network within the confines of a typical research budget.

## ACKNOWLEDGMENTS

I would like to extend my utmost thanks to Kelsey Jencso for supporting me through the excited progress and disappointing setbacks that have lead this project to it's completion, and for being curious enough about collaborative research in the first place. I would also like to thank Zach Hoylman for deploying the data loggers and gathering the data, without him this work would still be an untested design. I am grateful to Travis Wheeler, for nudging me further and further into research while being incredibly supportive and considerate throughout. I'd like to thank Michael Cassens, whose infectious passion for computer science is the reason that I chose to explore the field in the first place. I have unbounded appreciation for my father, who was always a resource for hardware design, as well as a life-long role model. I would like to thank all my friends, for believing in me, listing to my hardware-related ramblings, and suffering through my terrible jokes. And finally, an enormous thank you to coffee, without you this research would never have been remotely possible.

# TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1 INTRODUCTION

Characterizing the processes that lead to differences in ecosystem productivity and watershed hydrology across complex terrain remains a challenge. The processes contributing to streamflow or vegetation growth and productivity (e.g. soil moisture) are usually characterized by multiple measurements at discrete locations. Ideally, the density of measurements are taken at spatial and temporal resolution sufficient to resolve all the variability that influences the hydrological and ecological processes of interest. However, due to logistical constraints, measurements are often too sparse to fully characterize the natural variability that is inherent in complex forested landscapes.

The need to quantify environmental processes has spurred the growth of an entire industry of environmental data logger technologies. Professional-level data logging platforms such as the Campbell CR1000 are often considered the canonical tool in data acquisition for environmental processes. Designed to be extensible with additional peripherals and configurable via a proprietary scripting interface, Campbell data loggers are often able to provide a holistic solution to many sensing applications. However, in projects that require spatially distributed, the cost of instrumenting many locations can make these systems financially unviable.

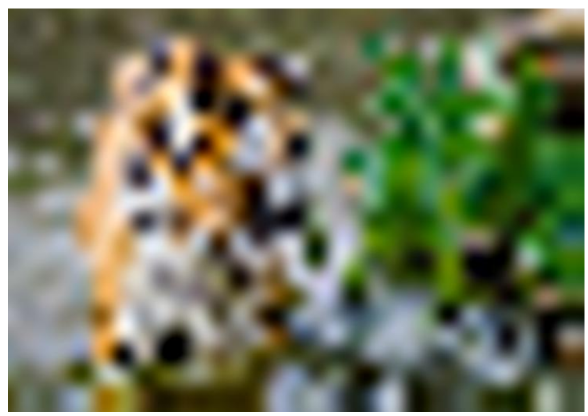
Application-specific data loggers can be a less expensive alternative, however many of these systems are, by design, limited in the sensors they support and the functions they can perform, often only supporting a single sensor. This limitation reduces the costs of the system, but make them inapplicable when multiple sensors are needed. With the recent popularity of open-source hobbyist electronics applications, many small, inexpensive data logging systems are available to an ambitious environmental scientist, including add-on shields that augment the capabilities of prototyping platforms like the Arduino[1][2] or full miniaturized computer systems.[3] By designing these systems around a user-programmable microcontroller, these programmable data loggers afford a scientist a modular data logging platform at a greatly reduced cost to professional-grade systems. Here, I describe the development of an open-source data logging platform that can interface with a wide suite of environmental sensors while inexpensive enough to instrument large, spatially distributed environmental studies.

## 1.1 Spatial Variability

Accurate modeling of hydrometeorologic processes is an increasingly important field of research. As the Earth's climate changes, understanding these processes is critical to predicting changes to natural and anthropogenic systems. At the regional scale, ecological processes are highly spatially heterogeneous, and are significantly influenced by local factors like geography, elevation, and topographic convergence and divergence of hillslope topography[4][5]. Building predictive models that account for these kinds of spatial variability requires spatially distributed data sets that can capture these spatial differences. With high enough spatial resolution in environmental data, these spatial relationships can be more accurately understood, and models can be made that take these relationships into account. Conversely, decreased spatial resolution can make identifying these spatial relationships difficult or impossible. Figure 1.1 illustrates this relationship between resolution and clarity of local heterogeneity. As the image is reduced to lower resolution, the visual structure becomes more difficult to understand. Likewise, trying to understand spatial variations in environmental data becomes more difficult as the resolution decreases.

(a) Image of a tiger at full resolution

(b) Image of a tiger at 1/4th resolution.



(c) Image of a tiger at 1/20th resolution.

Figure 1.1: (a) An image of a tiger at full resolution, (b) the same image with resolution reduced to, and (c) reduced to 1/20th the original resolution. With a small reduction in resolution, some information is lost but the original content is still recognizable. At much lower resolution, the original content is unrecognizable.

## 1.2 Overview

This thesis describes the iterative design that lead us from initial prototyping to a final system design for a cost-effective and full-featured environmental data logger. The methodologies of system development are explored, and the hardware and firmware design are explained thoroughly. Two field tests are presented, and the data produced by these tests are discussed. Finally, the efficacy of the system is discussed, and future system improvements are highlighted. The final designs, firmware source, and production files for the Pinecone data logger can be found at https://github.com/Sawwave/pinecone-datalogger, released under the GNU General Public License.

# CHAPTER 2 METHODS

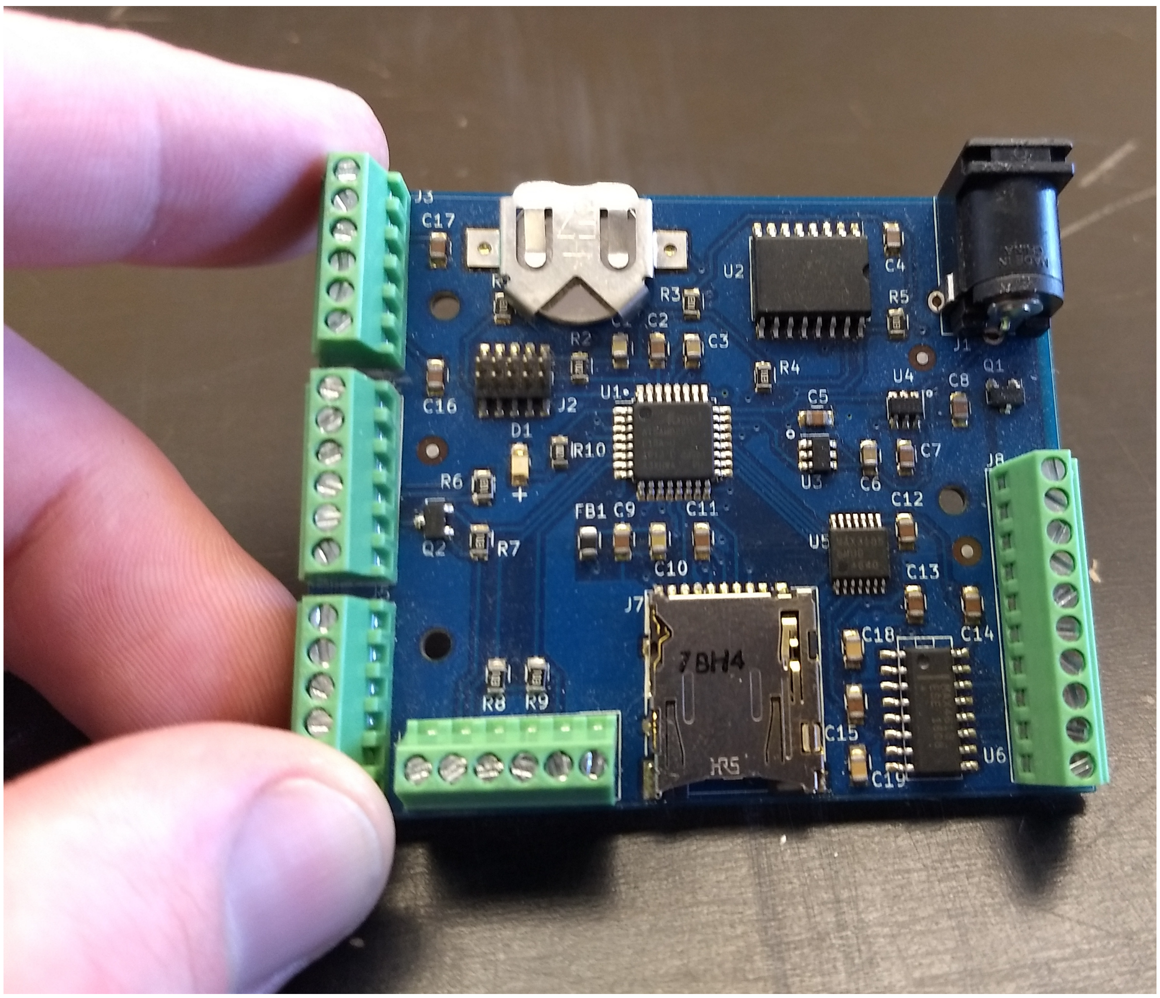



Figure 2.1: Image of the finished datalogger board

Image of a fully asssembled Pinecone datalogger. The hardware connects to the external sensors via the green screw terminals on the sides and bottom.

## 2.1 Prototyping

Development of the Pinecone data logger system began with initial prototyping on an Arduino with a shield attachment that added an SD card for non-volatile storage of logged sensor data. Then, work began on supporting sensors with the SDI-12 protocol, half-duplex asynchronous serial communication protocol primarily used for interfacing with hydrographic and meterorological sensors[6]. An Arduino library for SDI-12 support was used to interface with two Decagon 5TM sensors for soil temperature and volumetric water content[7], and TRS sockets were added to a connected breadboard to interface with the SDI-12 sensors. After successfully supporting the two 5TM sensors, a D1302 real time clock (RTC) was added to the prototype in order to provide accurate timestamps for the sensor logs.

Once assembled, it was determined that a design involving an Arduino prototyping board would not be a scalable solution because of the reliance on multiple hardware shields and a breadboard to support all necessary components; implementing the data logger with an Arduino and multiple extra hardware addons would have been expensive, bulky, electrically inefficient, and potentially unreliable. After determining that this implementation would not be sufficient, work began on designing a standalone printed circuit board (PCB) hardware system. A circuit schematic was created for a data logger that implemented the Arduino Uno's ATmega328p microcontroller unit (MCU), SD card socket, DS1302 RTC, voltage regulators, TRS sockets for interfacing with the SDI-12 sensors, and necessary components to shift logic levels between the 5 volt logic of the ATmega328p and the 3.3 volts of the SD card. The circuit schematic was then used to generate a PCB design in the program Fritzing[8]. Once the design was finalized, the design was exported to the industry-standard gerber format[9], and printed by the PCB manufacturing company Advanced Circuits[10].

The firmware was created with the Arduino development environment, and the data loggers' MCUs were programmed by slotting the ATmega328p chip into an Arduino Uno's DIP socket and using the Arduino toolchain to load the firmware. 10 Assembled boards were tested at the Lubrecht Experimental Forest over a 3 month period.

This initial system fabrication and test case highlighted a number of design issues to be addressed in the next version. The design used both through-hole and surface mount device (SMD) components; soldering the through-hole components was time consuming, and using a soldering iron on the SMD parts was difficult to do correctly, and resulted in approximately 20% of boards failing to pass quality control checks. A pull-down resistor on the SDI-12 data pin was accidentally excluded from the design. This was solved by soldering a resistor to the back of the board between a data pin on the MCU and ground on the power socket. During the case study, multiple systems failed due to battery failure. A contributing factor to the battery failure was found to be the delay function used to wait between logging intervals not properly putting the MCU into a low-power state. By using a sleep-mode library[11], MCU power usage between readings was reduced 300-fold from 1.2mA to 4.2$\mu$A. We determined that power usage and system cost could be reduced further by replacing some of the bulkier through-hole components with surface mount components that were smaller and less expensive.

After the first test study, work began to design an improved data logging system that could accommodate a wider range of environmental sensors, and improve on design of the prototype. We drafted a list of requirements to describe the primary capabilities and behaviors of the new design. Table 2.1 describes the primary capabilities and described in the updated design (Pinecone version 1.0).

A complete system that attempted to address all major requirements was produced after the initial prototyping phase. Interface terminals for two DHT-22 sensors

| Requirement | Description |
|---|---|
| Soil Water Content | Support up to 5 Decagon 5TM sensors running on an SDI-12 interface bus. |
| Tree Basal Area Increment | Support for up to 2 recording dendrometers. |
| Sap Flux | Support for collection of sap flow measurements via the Heat Ratio Method as in Burgess, et.al, 2007. System must implement 4 differential thermocouples and a controllable external heater capable of generating 9 watts of heat for pulses of at least 3 seconds. |
| Relative Humidity | Support for 2 external relative humidity sensors. |
| Air Temperature | Support for a minimum of 1 external ambient air temperature sensor. Sensor must include an enclosure. |
| Data Storage | System must be able to store at least 1 year of collected data at a logging interval of 30 minutes in non-volatile memory. Data must be retrievable in CSV format. |
| Time Stamping | System must be able to accurately time stamp all sensor readings. System must be able to timestamp correctly after losing power. |
| Runtime Configuration | System must be able to configured through text files loaded onto an SD card. System configuration must include a configurable wait time between readings. |
| Firmware In-Service Reprogramming | System must support firmware changes after deployment |

Table 2.1: List of significant functional requirements that must be addressed in a successful implementation of the outline data logging system.

(relative humidity and air temperature), two point dendrometers, and support for hardware sap flow sensors were added to the system. A solar panel and solar regulator were included to address power consumption issues. The firmware was adapted to support these new components, and the MCU was switched into a low power state when idle. Wireless communication support was considered as a requirement, but ultimately rejected due to power usage of wireless transceivers, system cost, and poor

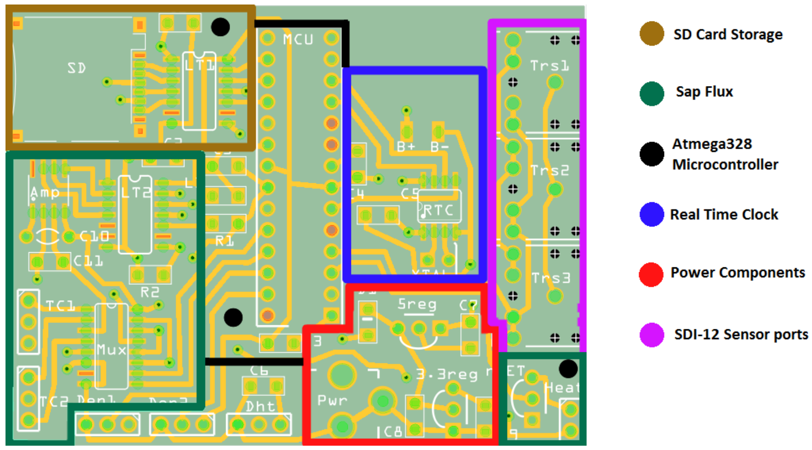



Figure 2.2: Diagram of the Pinecone version 1.0 circuit board. Individual subsystems are colored and labled.

signal reception in the densely forested case-study location. This iteration of the data logger design was named the Pinecone version 1.0.

54 logger units were deployed in Lubrecht Forest from April 2016 to October 2016. The 2016 case study highlighted a few issues that still needed to be addressed with the logger design. When the data generated from the sap flow thermocouple system was analyzed, the data product was noisy and had low resolution. This was determined to be caused by poor signal to noise ratio and low thermal resolution of the thermocouple amplifier hardware (Figure 2.3). Additionally, the new design had significantly improved power usage, but some sites still had power failures, resulting in temporal gaps in the data logs. In order to support sap flow and reduce power usage, another board design was created, and boards were fabricated and populated to replace the previous systems. This final data logger iteration was named the Pinecone version 2.0.

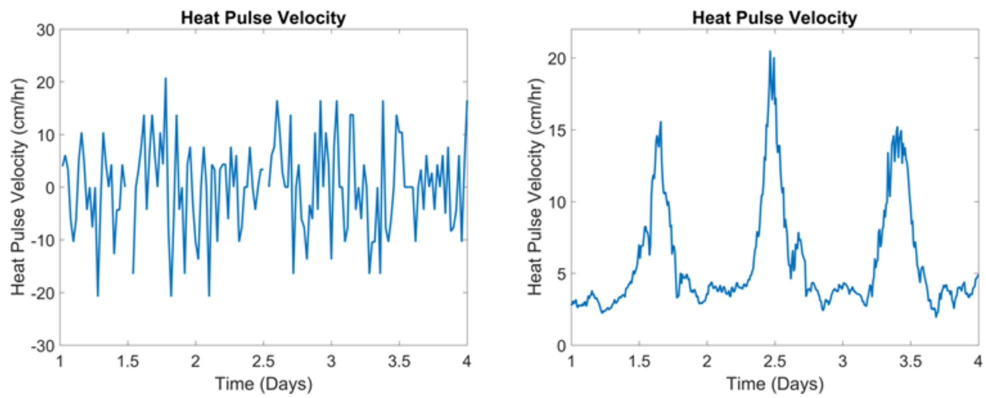



Figure 2.3: Time series comparison of heat pulse velocity measurements for quantifying sap flow, generated by the MAX31855T thermocouple amplifier used in the Pinecone version 1.0 (left) and the MAX31856 thermocouple amplifier used in version 2.0 (right). The diurnal signal of sap flow can be easily seen in the data from the MAX31856, but the low resolution and high noise from the MAX31855T fails to accurately capture the signal.

## 2.2 Hardware

### 2.2.1 Microcontroller

The Atmel SAMD20E was chosen as the microcontroller for the version 2.0 design to address shortcomings of the ATmega328P used in the previous two iterations. Table 2.3 compares some significant specifications about the two microcontrollers. This change was motivated by the desire for a more powerful and more electrically efficient microcontroller, and to transition away from using the Arduino framework. As a result, we were able to create larger firmware programs than would be able to fit onto the ATmega328P, and the increased SRAM size allowed for the program to be more easily written without running out of memory. A 10-pin cortex debug connector was included to initially program the firmware and allow for firmware updates after deployment. Figure 2.4 shows the schematic for the SAM D20E MCU and ancillary components.

| Requirement | Part | Interface/ SDK | Price |
|---|---|---|---|
| Microcontroller | SAM D20E | Advanced Software Framework | \$2 |
| Point Dendrometers | Linear Potentiometer | Onboard ADC | \$23 |
| Air Temp/ Humidity | DHT-22 | Software One-wire | \$10 |
| Sap Flow | MAX31856, MAX4618 | Hardware SPI, GPIO Mux Select | \$10 |
| Soil Temp/ Water Content | Meter 5-TM Sensors | Software SDI-12 | \$170 |
| Data Storage | Micro-SD card | Hardware SPI, FATFS | \$10 |
| Real Time Clock | DS3231 | Hardware I2C | \$8 |

Table 2.2: Table of major hardware components in the Pinecone version 2.0 circuit board, or the sensors they use. All prices are listed per item, as of January 2019. Note that while all SDI-12 compatible sensors are supported, the ones used in our field tests are the most expensive components.

### 2.2.2 SDI-12

Screw headers were included to allow up to 4 SDI-12 compatible sensors to interface with the datalogger. Since all SDI-12 sensors share the same data bus line, more sensors may be supported by externally connecting their data lines to the onboard data bus. While the SDI-12 specifications allow up to 62 sensors on a single bus, version 2.0 currently limits the number of sensors to 24, due to memory constraints in the firmware. While many SDI-12 compatible sensors support logic at 3.3v, the specification standard only guarantees functionality with 5v logic. In order to communicate with the 3.3v logic MCU, a bi-directional level shifter circuit was implemented on the datalogger with an N-channel MOSFET and two pull-up resistors (Figure 2.5). The 5v source for the sensors is disabled by an onboard load switch when the sensors are not in use in order to reduce power usage.

| | Atmega328P | SAMD20E |
|---|---|---|
| **Program Memory** | 32 kB | 256 kB |
| **Maximum Clock Speed** | 16 MHz | 48 MHz |
| **SRAM** | 2 kB | 32 kB |
| **General Purpose IO Pins** | 23 | 26 |
| **ADC bit depth** | 10 bit | 12 bit |
| **Power Supply Current (Active)** | 5.2 mA (@ 5V) | 2.33 mA (@ 3.3V) |
| **Power Supply Current (Idle)** | 1.2 mA (@ 5V) | 0.78 mA (@ 3.3V) |
| **Power Supply Current (Power Down)** | 4.2 $\mu$A | 3.8 $\mu$A |

Table 2.3: Typical Electrical characteristics and capabilities of the MCUs used in the prototype system (Atmega328P) and the final design (SAM D20E)

### 2.2.3 Non-Volatile Storage

A socket for a mini-SD card was included on the datalogger board (Figure 2.6). An SD card was our preferred storage medium because of their cost efficiency, reliability, and because they are easy to remove and read on a personal computer in the field. Any SDHC memory card with a FAT file system is supported.

### 2.2.4 Real Time Clock

A DS3231 real time clock (RTC) was integrated into the system to allow data to be accurately timestamped (Figure 2.7). The choice for this IC was motivated by its stability over a wide range of extreme temperatures due to its internal temperature controlled crystal oscillator (TCXO). The MCU also uses the alarm functionality of the RTC to create external interrupts that wake the MCU from the low power standby state between readings. A 3 volt coin cell and coin cell retainer were included to allow the RTC to function when the system is unpowered. A retainer was chosen over a

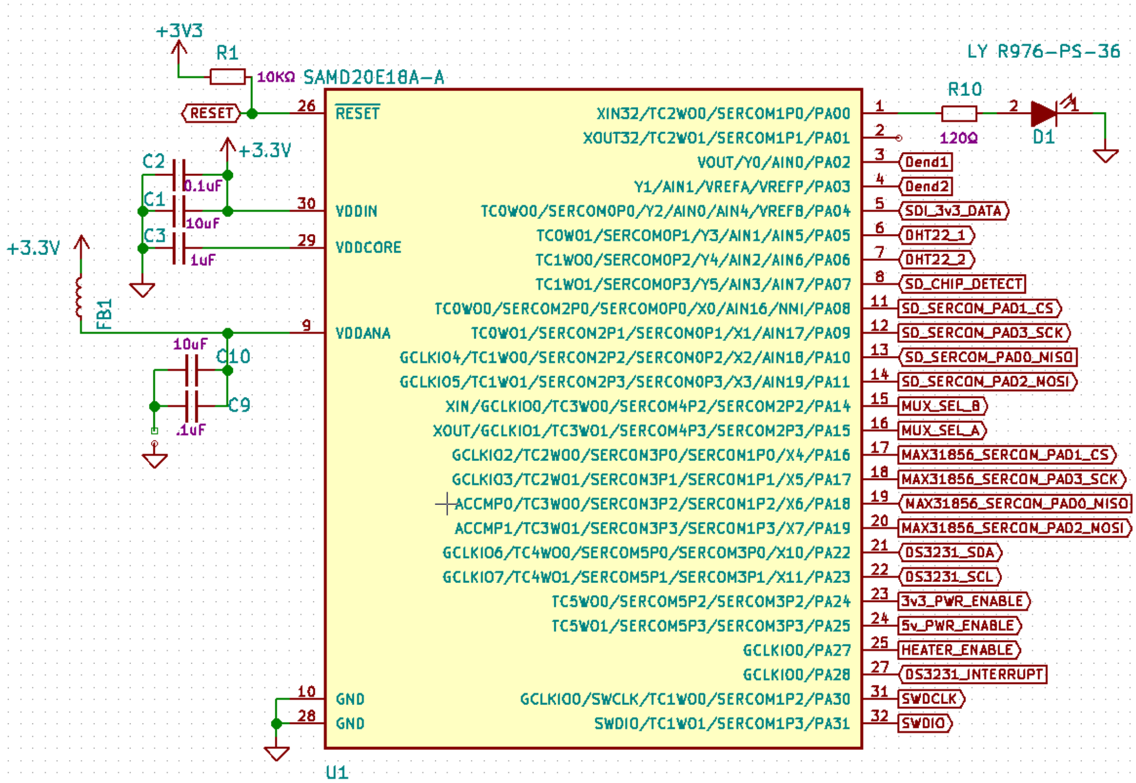



Figure 2.4: Schematic of the SAM D20 microcontroller and related hardware components. The included LED allows the board to signal simple error codes and show successful operation.

soldered battery to allow for ease of battery replacement.

### 2.2.5 Sap Flow Sensing

In order to quantify sap flow in tree xylem tissue, the Pinecone datalogger system includes hardware to perform the Heat Ratio Method (HRM) as outlined in Burgess et. al, 2001[12]. In this method, four thermocouples are inserted into the xylem tissue inside syringe needles, two above and two below a 16Ω needle heating element. To perform sap flow measurement, temperature readings are taken with each thermocouple, and a 3 second heat pulse is generated from the heating element with a 12 volt source. The temperature readings are then taken 60 seconds later, and the 8 readings

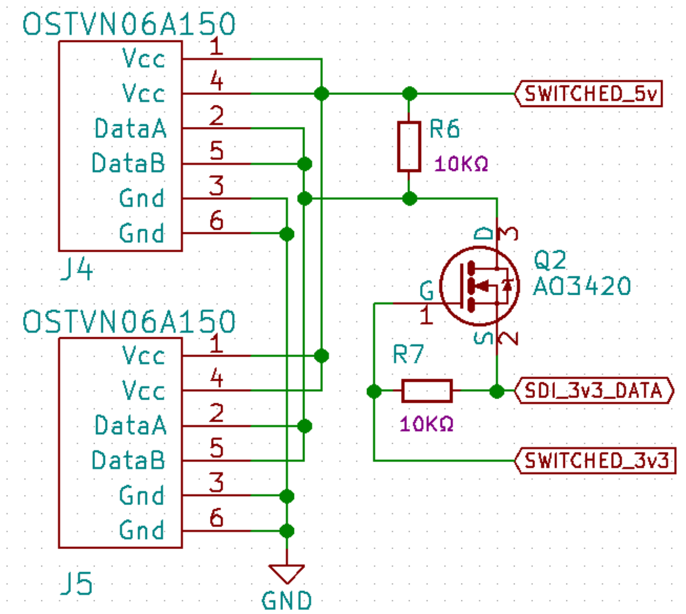



Figure 2.5: Schematic for the SDI-12 hardware bus. The included MOSFET, along with the pull-up resistors, bi-directionally translates between the 5v logic of the SDI-12 sensors, and the MCU's 3.3v logic.

are used as inputs to calculate the sap flow, once corrected for factors like xylem thermal conductivity and tissue wounding correction. A MAX31856 thermocouple amplifier and serial interface IC was included to measure the thermocouple voltages for our sap flow measurements (Figure 2.8). As the sap flow measurements use small changes in temperature, Pinecone version 2.0 incorporated a thermocouple interface with a resolution of 0.008C to accurately measure these changes. A MAX4618 two circuit, 4-channel multiplexer was included to multiplex the 4 thermocouples into the single amplifier. This IC was chosen for its low on-state resistance (8Ω) as to not distort the thermocouple signal voltages. Screw terminals are provided to interface with the 4 thermocouples that are inserted into the xylem tissue of a tree above and below the heating element. An N-channel MOSFET was used to allow the MCU to control the heat pulses. The heater consumes the most energy of the entire system,

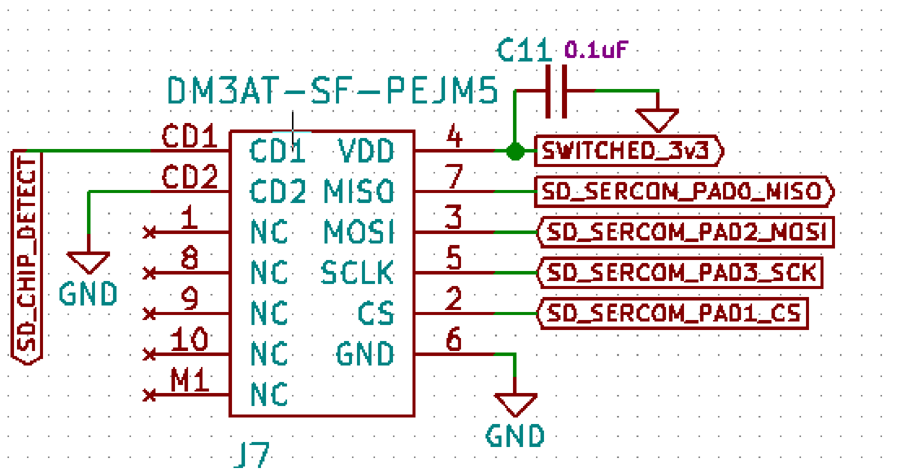



Figure 2.6: Schematic of the SD card holder. The MCU communicates with the memory card via a SPI communication bus.

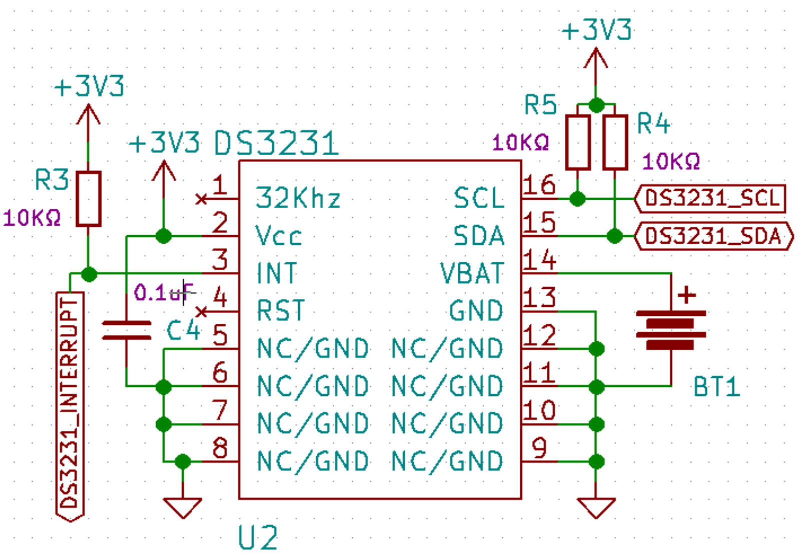



Figure 2.7: Schematic for the DS3231 Real Time Clock. The included coin cell battery allows the device to keep time when the board isn't powered.

consuming 750mA for a duration of 3 seconds.

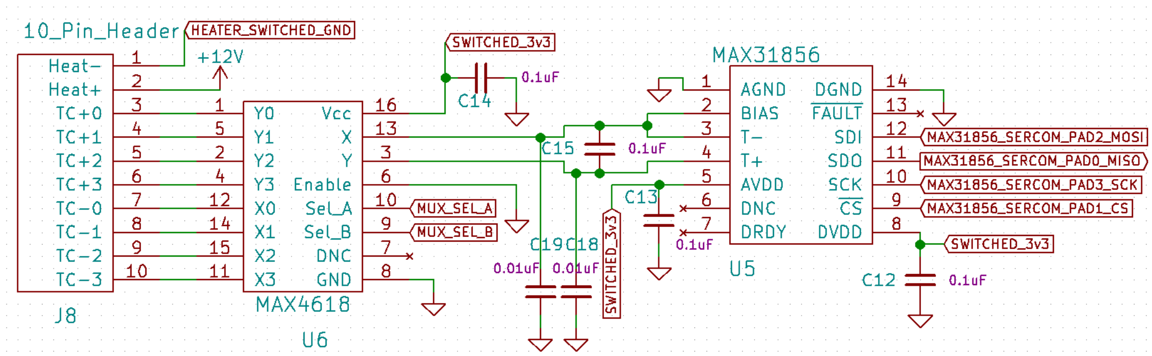



Figure 2.8: Schematic for the thermocouple system. The MAX4618 IC multiplexes the 4 thermocouples into the MAX31856 thermocouple amplifier. The MCU can then communicate with the amplifier via the SPI bus.

### 2.2.6 Temperature, RH, and Dendrometers

Screw terminals were included to interface with point dendrometers and two DHT22 relative humidity and temperature sensors. The point dendrometers are implemented as linear potentiometers that are mounted onto a tree on opposite sides to accurately quantify tree basal area[13]. The linear potentiometers are used as voltage dividers between 3.3v and ground, and the resulting signal is read by the MCU's internal 12-bit analog to digital converter (ADC). As the potentiometers used in our case study have a travel distance of 12.7mm, the resulting data has a resolution of 3.1 microns. Two DHT22 sensors are supported, and collect temperature with an accuracy of ±0.2C and resolution of 0.1C, and relative humidity with an accuracy of ±2% and resolution of 1%. Because of the minimal weather protection on these sensors, plastic weather shrouds were fabricated following (Holden, 2013)[14]. Four ports are provided for interfacing with SDI-12 interface compatible sensors. The SDI-12 sensor bus is enabled when needed, and sensors are queried by their known addresses.

### 2.2.7 Power

A MIC5211 linear voltage regulator was included to generate regulated 3.3 and 5 volt sources from a 12 volt battery. (Figure 2.9)To reduce system power usage, an AP2281 load switch IC was included to reduce power usage by cutting power to the 3.3v sensors when the system goes into the power-down between readings. The lead acid battery, solar panel, and solar regulator were included in the case studies, but are not part of the data logger design.

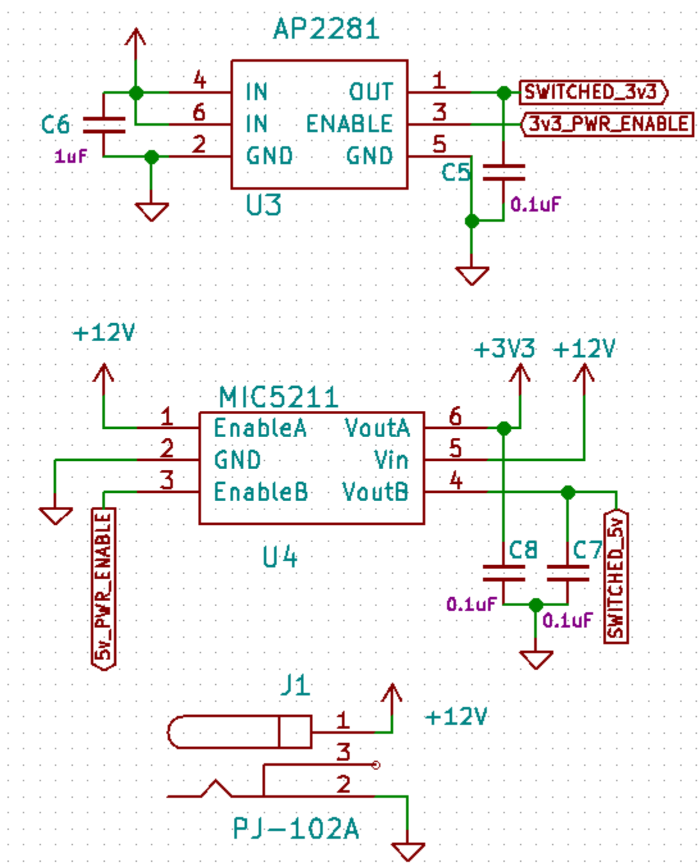



Figure 2.9: Schematic of the power control hardware. the MIC5211 provides the regulated 3.3v and 5v power sources, and uses the EnableB pin to keep the SDI-12 subsystem powered off when not in use. The AP2281 gates the power to non-critical 3.3v systems when not in use.

### 2.2.8 Hardware Production

The printed circuit board (PCB) design was performed in the electronic design automation suite KICAD[15]. KICAD was chosen for its ability to design PCBs with 4 or more copper layers and for being open-source with no purchase cost. The printed circuit board schematic was designed and used to generate PCB trace routing between components. A four layer design was created with 2 signal layers, a power layer, and a ground layer. This design significantly simplified trace routing and reduced signal noise. The PCBs were fabricated by the company Seeed[16], and a PCB solder stencil was purchased with the boards. Once PCB was fabricated, the loggers were assembled on-site. To facilitate applying the solder paste with the solder mask, a board retainer was created by taping old PCB prototypes to a plywood board. To apply the solder paste, a PCB was slotted into the retainer, and the solder mask was taped over the board into the correct position. Once the solder paste was applied with the mask, the surface mount components were placed by hand with tweezers. Once all surface mount parts were placed, the populated boards were baked in a small toaster oven (Figure 2.10) controlled by a reflow oven controller from RocketScream[17]. Once surface mount soldering was complete, all through-hole headers were soldered by hand. Silicone spray conformal coating was applied to the finished board to prevent condensation-related electrical shorting.

### 2.2.9 Firmware Design

The system firmware was written using Microchip's Advanced Software Framework (ASF)[18] development kit, and is comprised of 2,205 lines of C99 code, not including external ASF libraries. To upload the firmware to the device, an Atmel ICE (In Circuit Emulator and debugger) is connected to the datalogger's 10-pin cortex debug


Figure 2.10: Surface mount soldering reflow oven made from a standard toaster oven. In order to achieve a high enough temperature to properly solder the components, a toaster oven was chosen that could reach 500 degrees Fahrenheit.

connector. When the system is powered on, the firmware begins by detecting if the source voltage is below the brown out level ( 1.7V)[19]. If a brown out is detected, the system is put into a sleep state for 10 minutes to avoid losing power during a critical logging operation. Once the firmware has assured a sufficient source voltage, it begins initializing the necessary hardware registers, disables the 5V subsystem for the SDI-12 bus and checks to make sure that an SD card is connected and functioning correctly. After initialization, the datalogger attempts to find the file “time.txt” on the root directory of the SD card. If this file is found, it parses the file’s contents into a datetime format and uses it to set the real-time clock. After setting the datetime, it deletes this file so that it doesn’t set the clock to this time on the next system boot. The datalogger then looks for a configuration file “lgr.cfg” in the root directory of the SD card. This file contains usage configurations that allow the user to set the interval between readings, force the logger to begin logging at the start of the next hour, set

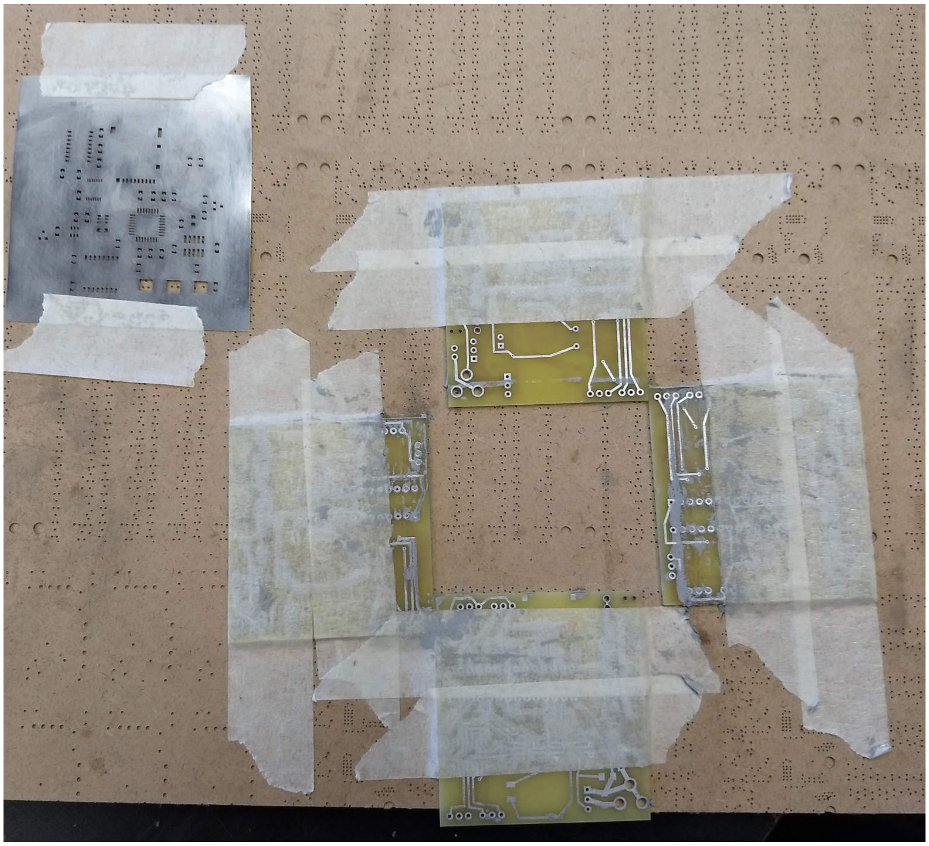


Figure 2.11: Board retainer used for applying the solder stencil. Once a board was firmly placed in the board retainer, the metal solder stencil was taped into position, solder was applied to the stencil, and a metal squeegee was dragged across the stencil to apply the solder to the board.

the type of thermocouples being used for sap flux measurements, disable any sensors that are not currently being used, and set the addresses of the SDI-12 sensors that the logger will communicate with. Each sensor address is followed by the number of values expected to be received from that sensor. While this requires user knowledge about the sensors being used, it ensures that all recorded logs have the same number of values, even in the case of communication failure with one of the SDI-12 sensors. Error codes are shown to the user via a small LED on the board. Successful setup will result in 3 quick, 100ms blinks to notify the user that the system is powered on and correctly operational. Slower, 800ms blinks are used to show the following error states: A single slow blink to indicate the SD card not being found, 3 slow blinks to indicate the configuration file missing, and 4 slow blinks to indicate that the real-time clock is not configured, but the time file was not found. When the datalogger is ready to record readings, it records the current date time, and sets an alarm in the real-time clock to generate an external interrupt when the next readings are scheduled. The system then enables the 5V power subsystem, and iterates through each of the sensors. If a sensor is enabled from the configuration file, the logger attempts to read from the sensor. To read from the dendrometers, the MCU simply queries the ADC for each of the two IO pins the dendrometers are connected to. To get the temperature readings from the thermocouples, the MCU communicates with the MAX31856 thermocouple amplifier via it's Serial Peripheral Interface (SPI) hardware bus. Thermocouple inputs are multiplexed into the amplifier by setting or clearing the multiplexer's select pins. The heater is then enabled and disabled by controlling the gate of a MOSFET. Communication with the DHT22 sensors is performed with software rather than dedicated hardware, since it's protocol is not widely supported. Likewise, there are few hardware options for implementing the SDI-12 interface, so the interface is implemented in software as well. Once all sensors have been read, the the data are appended to the end of a log file on the SD card. Any sensor readings that were disabled or had a communication error are replaced with the text "NAN". The datalogger then disables power to the 5V subsystem and the 3.3v load switch, and it sets a watchdog timer for one minute greater than the logging interval to act as a backup alarm in case of a failure in the real-time clock. Once this timer has been set, the MCU enters a standby power state to conserve energy until it wakes up for the next logging event.

## 2.3 Field Testing

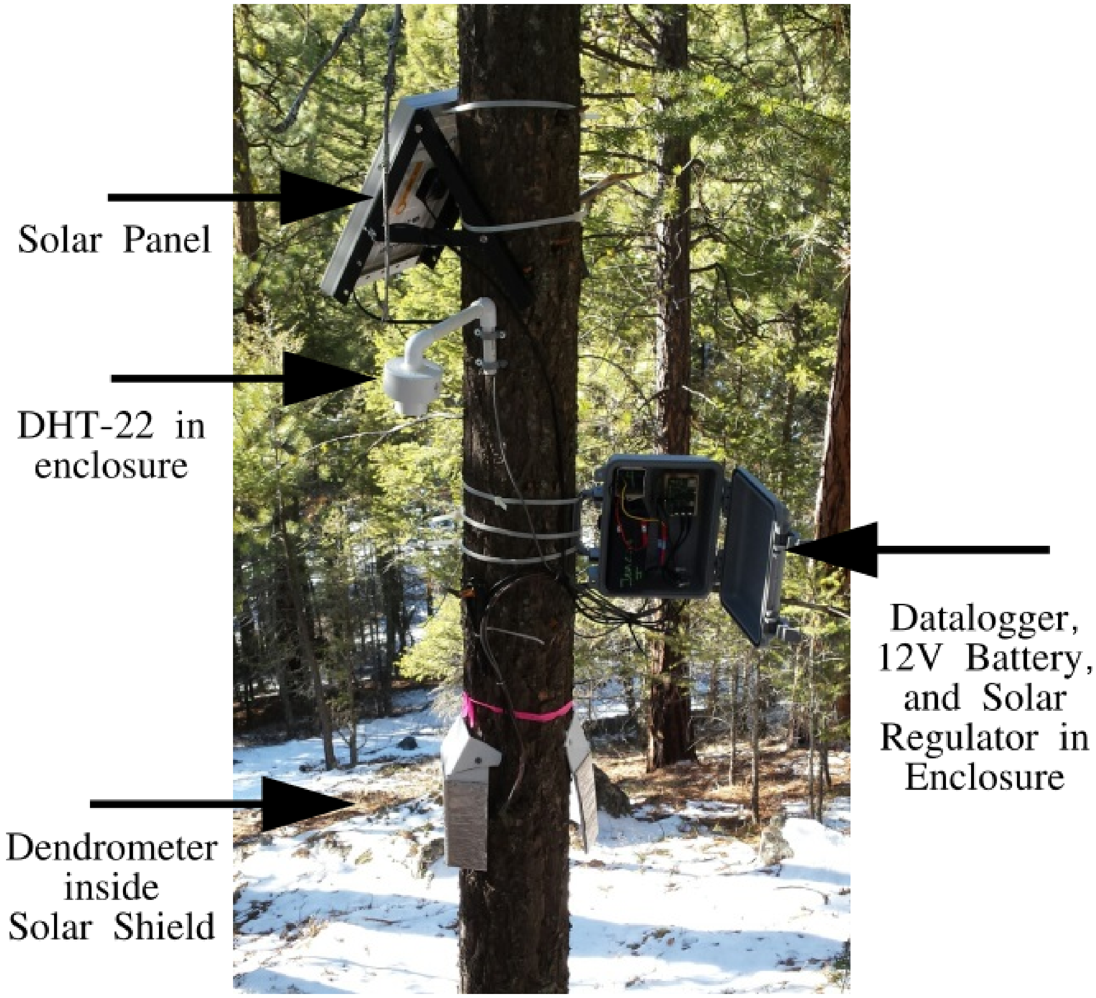



Figure 2.12: The Pinecone datalogger system fully installed on a tree in Lubrecht Experimental Forest. From top to bottom, a solar panel facing away from the camera, to recharge the system battery, A DHT-22 sensor inside a protective enclosure made from PVC piping, the opened case that holds the datalogger board, battery, and solar panel regulator, and shiny thermal protectors to shield the dendrometers from heating up in direct sunlight.

To study the reliability of the final data logger systems, we performed a case study in the North Fork Elk Creek (NFEC), a semiarid headwater catchment in the Lubrecht Experimental Forest (LEF), located in the Garnet Mountain Range in Western

Montana (47°N, 113°W). Fifty-four data loggers were installed in the NFEC watershed over six hillslopes representing nine locations across north and south aspects of low, mid, and high elevation slopes. At each location, a data logger system was attached to a tree, and connected to two dendrometers, two Decagon 5-TM Volumetric Water Content (VWC) and soil temperature sensors measuring at depths of 5cm and 50cm, and two DHT-22 relative humidity and temperature sensors installed at 15cm and 200cm above the ground floor. The Pinecone version 2.0 fully implements sap flow measurement, but was not used in the field tests due to concerns about the heating element's power usage. The system was powered by 12 volt lead-acid battery, and recharged with a 10 watt solar panel mounted to the tree trunk at a steep angle such that snow would not accumulate on the panel's face. A solar panel regulator was used to regulate the charging voltage of the battery. From April, 2016 until October 2016, each site was instrumented with a prototype of the datalogger system. The next year, each site was replaced with the final version of the Pinecone datalogger, and collected data from May 2017 until July 2017. Figure 2.12 shows one of the data loggers fully installed with a full compliment of sensors.

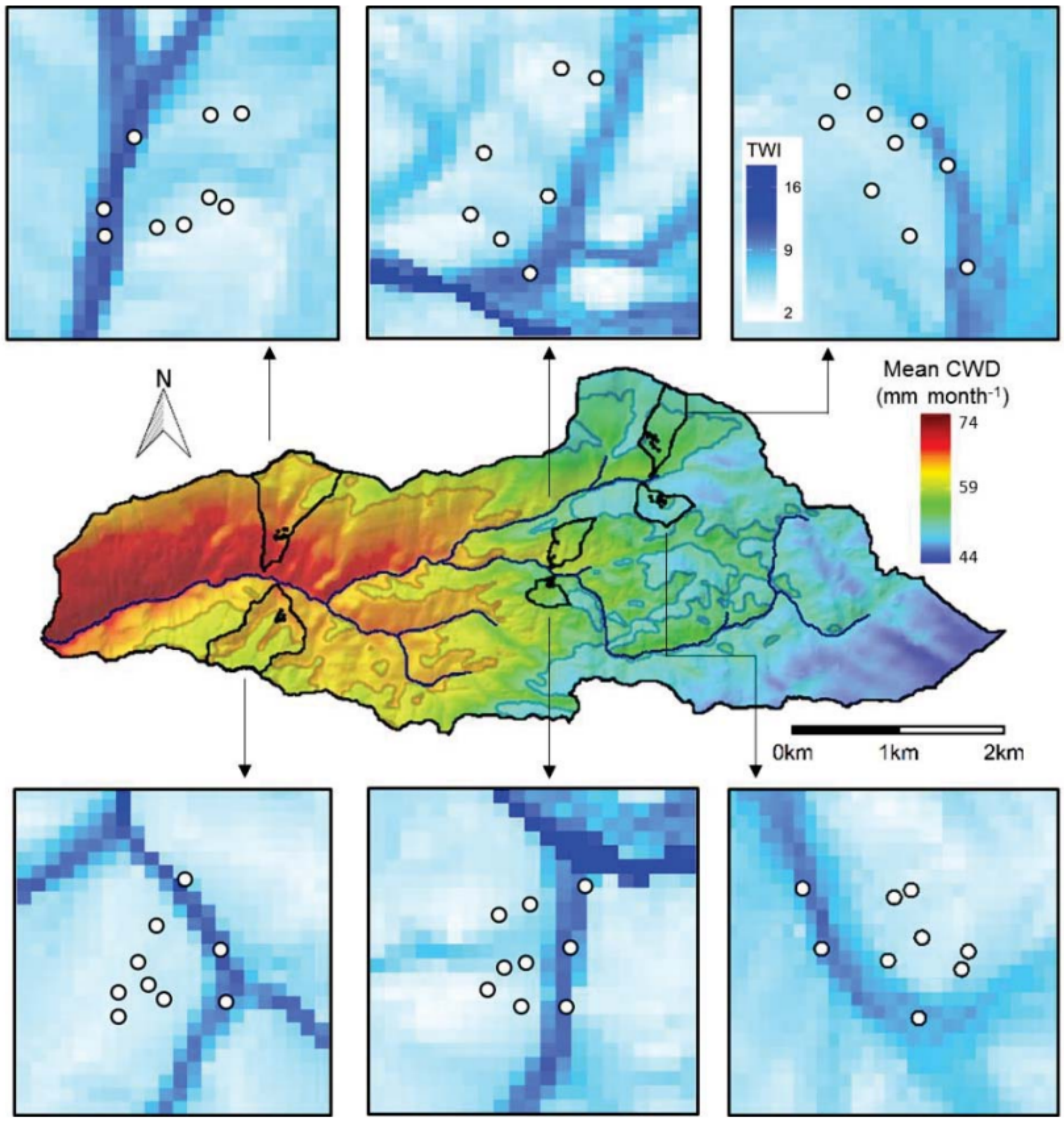



Figure 2.13: Map showing the location of the North Fork Elk Creek (NFEC) catchment within Montana, the climatic water deficit across the NFEC catchment and delineated boundaries of each study catchment. White circles, squares and triangles in call out plots represent hollow, sideslope and upslope hillslope positions respectively. Dashed boxes show regions represented by call out plots.[20]

# CHAPTER 3 RESULTS

The data logs were gathered from the 2017 field test study and analyzed to gauge reliability of the data logger systems. In the data presented below, high, mid, and low elevation hillslopes are denoted with a prefix HE, ME, and SE respectively, and north and south aspects are denoted with suffixes NA and SA. 9 sites were chosen for data logger installation on each hillslope, generally following the expected path of water down the hillslope. Of these 54 sites, two were left uninstrumented due to difficulty accessing the site or prior issues with the location: HESA #5 and MESA #5. Each hillslope was allowed to collect data over a different range of dates, as outlined in Table 3.1.

| Hillslope | Start Date | End Date | Length of Time Series |
|---|---|---|---|
| **HENA** | May 5th, 2017 | July 26th, 2017 | 82 Days |
| **HESA** | May 6th, 2017 | July 26th, 2017 | 50 Days |
| **MENA** | April 12th, 2017 | July 26th, 2017 | 105 Days |
| **MESA** | May 1st, 2017 | July 26th, 2017 | 86 Days |
| **LENA** | June 2nd, 2017 | July 26th, 2017 | 54 Days |
| **LESA** | June 2nd, 2017 | July 26th, 2017 | 54 Days |

Table 3.1: Ranges of dates for data logger deployments in the 2017 case study. Hillslope locations are identified with abbreviations for high, mid, or low elevation (HE, ME, LE), and north or south aspect (NA, SA)

## 3.1 Performance Analysis

Of the 52 sites instrumented, three sites experienced failures such that they stopped collecting data. HENA 3 and MESA 9 were found to have completely drained batteries. MENA 5, however, was partially torn off of the tree it was attached to, and sensors were destroyed. Since the damage to the system was similar to logger systems previously damaged by bears, we attribute this failure to interference by a large animal as well.

| Hillslope | Site Index | Time Series Length | Days Before Failure |
|---|---|---|---|
| **HENA** | 3 | 82 Days | 40 Days |
| **MENA** | 5 | 105 Days | 87 Days |
| **MESA** | 9 | 86 Days | 82 Days |

Table 3.2: System failures that lead to time series ceasing to be gathered. Failures at HENA 3 and MESA 9 were battery-related, while failure at MENA 5 was due to a large animal destroying the system.

The power failures in the 2016 study were the major motivating factor for the final system redesign. Compared to version 1.0 used in the 2016 field study, version 2.0 consumed significantly less power, from 22 mA current during all operation to 3.5 mA and 1.4 mA during active logging and in power saving mode respectively. Three system failures during the 2017 field study are listed in Table 3.2. The power failures at HENA 3 and MESA 9 were similar to a number of failures over the 2016 field test. When the systems were visited to retrieve the data logs, the lead acid batteries were found with voltages so low that the battery was considered destroyed (e.g. 4 volts, instead of 12 volts). This efficiency helped to reduce the number of power failures, but did not eliminate them entirely. These two power failures are primarily attributed to the solar panel system used to recharge the lead acid battery. Because the 10 watt solar panels we used[21] were mounted to a tree at a steep angle, the panel's power generation was reduced. At many of our installation sites, tree branches also partially blocked direct sunlight to the panel. This kind of "hard shading" results in a reduction in output voltage relative to the ratio of shaded to unshaded cells on the panel.[22]. As our solar panels output a maximum of 17V, if a large section of cells are shaded on a panel it may output less voltage than the 12.7V required to charge as 12V lead acid battery. The topography of the study location also compounded these issues, since all the installation sites were in mountainous terrain that further reduced the hours of direct sunlight. The solar panel regulator was also a contributing factor to the power usage, rated to self-consume 6mA at all times[23]. The combination of poor solar energy collection and the constant draw from the regulator also explains some of the power failures from the 2016 field study; many sites had no power failures, while some sites with poor direct sunlight experienced multiple power failures during the 8 month study, even after replacing both the lead acid battery and data logger hardware.

Of the 169,774 data records generated over the 2017 field test, communication with the SDI-12 experienced errors such that at least one variable from the sensors was not logged in 208 records, or 0.1%. Our best explanation for these communication failures is due to slight variations in timing in the asynchronous software implementation of the SDI-12 protocol. This

14 data loggers experienced a real-time clock reset before installation. In these data files, the time series starts either at January 1st, 2000, or a similar but illegal timestamp (e.g. 0/0/2000). The default behavior of the DS3231 RTC is to begin keeping time when initialized, so each of these data logs keeps time, starting from the moment it is powered on. Therefore, since the time of installation was recorded for each logger, it is trivial to fix the log files with approximate timestamps. Since

all the loggers had their RTC time states set and quality control checked as part of production, losing RTC time configuration suggests that the times were lost or corrupted between production and installation. Our best guess is that the data logger units were not properly packed to reduce movement in transit, and the RTC backup battery became dislodged from the battery retainer during deployment. The decision to use a battery retainer rather than soldering the battery to the board was motivated by the desire to make the batteries easily replaceable, but we did not anticipate this issue when choosing retainers. Other than resets on the stored RTC dates, one record was unable to be timestamped due to communication error with the RTC. Of the 52 data loggers deployed during 2017, 4 experienced gaps in data greater than one hour, not including gaps due to complete power failures. LENA 7 had one gap of 12 hours, MESA 2 had a 1 hour gap and a 2 hour gap, and MENA 9 had 8 data gaps between 2 hours and 9 hours. MENA 5 had 19 data gaps, however since this was one site where the battery power source failed, data gaps are not surprising.

# CHAPTER 4 DISCUSSION

While the field study data products show some problems, these problems mostly fall under usability, rather than major design flaws. Environmental data loggers should be designed to be hard to use incorrectly, and fail only in exceptional cases. While the data logger design has some work needed before it fully meets these standards, the Pinecone data logger has allowed us to generate a high quality, spatially distributed data set that can help refine ecohydrological and climatological modeling techniques with spatial resolution that was infeasible with more expensive, commercial systems[20]. An example of the data created from the 2016 field study can be found in Figure 4.1, where precipitation events have demonstrably different impact on dryness at varying topographical positions.

The data analysis using these data have shown that, across a watershed, hillslope position has a strong impact on hydrometorologic dryness in especially arid locations like lower elevations and southern aspect slopes. Conversely, hillslope position was shown to have less of an impact on dryness in wetter, high elevations and north aspects. The volumetric water content data showed that subsurface water was almost entirely absent from arid regions of the watershed, even in areas with large convergent drainage areas, but persistently occurred across both convergent and upslope positions in wetter regions of the watershed. Finally, data from these loggers showed a strong spatial organization of shallow subsurface water flow as a function local hydrometeorology. This relationship suggest that local moisture conditions play a

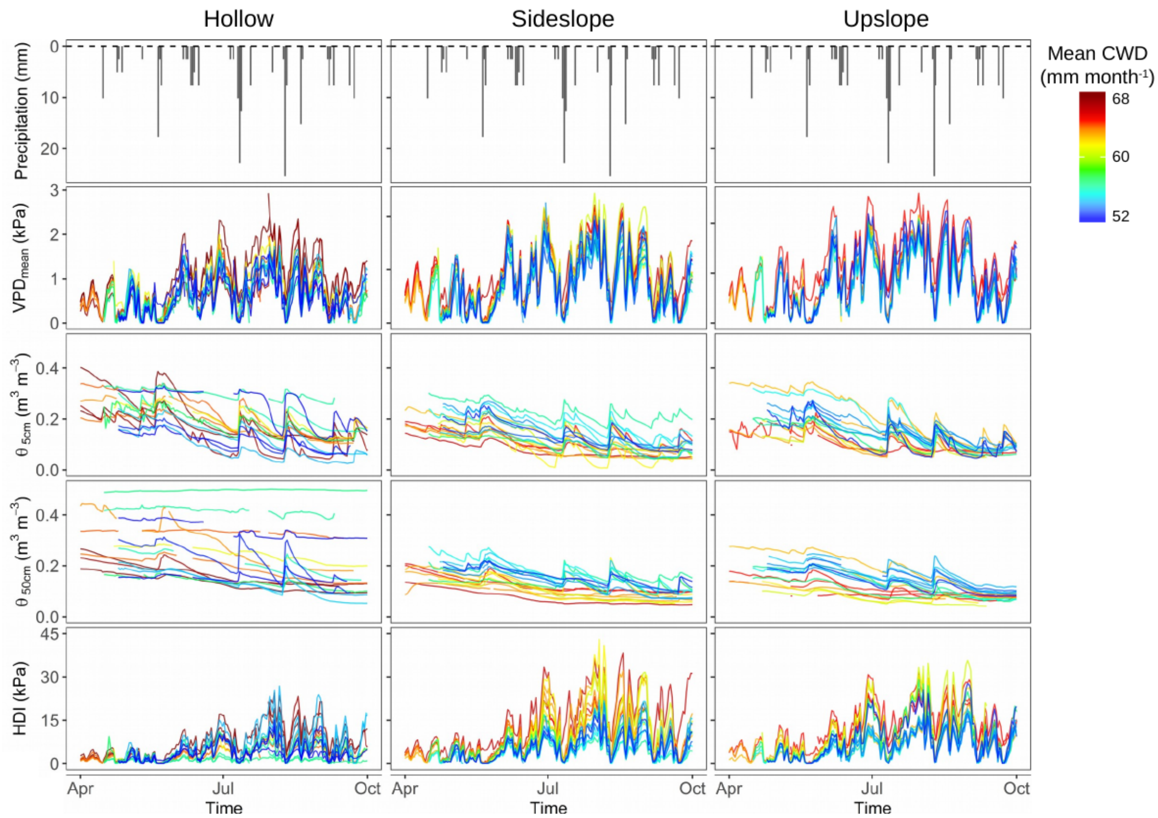



Figure 4.1: Data and data products created from 2016 data logger records.[20] Vapor Pressure Deficit (VPD), and Volumetric Water Content (VWC) at two depths (5cm and 50 cm) are plotted over various Cumulative Water Deficit (CWD) environments. From these data, hydrometeorologic dryness index (HDI) is modeled and plotted to show how precipitation events affect dryness at various water deficits.

dominant role in driving hydrologic connectivity.

### 4.0.1 Future Improvements

Field testing the Pinecone data logger highlighted a few improvements that can be made to the design, especially in the field of user experience. A number of data loggers had issues with the RTC configuration due to the battery being jostled out of the retainer. To solve this, the board can be modified to use a retainer that more firmly holds the battery in place, or the logger can be provided to the user with a custom plastic housing that would hold the battery in the existing retainer. The current configuration file format is easy to read at the software level, but is not very human readable; this could be improved by changing the configuration file to a more readable format like JSON. This quality-of-life change wouldn't have been possible with the memory limitations of the ATmega328P, but the current MCU could support it easily

While sap flow sensing hardware was an important feature of the Pinecone's design, it is likely that other studies using this data logger system would not include it. Since the sap flow hardware contains some of the more expensive parts for building the system, a future version of the data logger may move the sap flow hardware components to a connectable add-on board, and reduce system cost for users who don't need to measure it.

While SDI-12 sensor recording errors occurred in only 0.02% percentage of recordings, a revision to the protocol implementation may reduce these errors. If it is assumed that these were the result of communication protocol implementation, there are a few approaches that may improve the communication reliability. One approach would involve adding hardware to the data logger board to enable the SDI-12 protocol to be performed with the MCU's Universal Asynchronous Receiver/Transmitter (UART) communication bus. Since the SDI-12 protocol uses inverted logic compared to UART (i.e., SDI-12 treats 0v as logical 1, and 5v as logical 0), a hardware implementation must bidirectionally logic level shift between 3.3v logic and 5v logic, as well as bidirectionally invert the logic. Circuits to perform this conversion for SDI-12 have been created, however this implementation requires additional components and uses 3 of the MCU's IO pins instead of 1.[24]

Wireless communication is a popular technology in environmental sensing. The ability to automate data collection for data logger units can significantly save research time, especially if there are many data logger units spread out spatially over complex terrain, as in our field study. Wireless communication ability would be a large improvement to the current system, however it may not be appropriate for every application. Ultra-High Frequency (UHF) technologies like LoRa have been used widely in environmental data loggers, but complex terrains and dense foliage can significantly attenuate the signals, reducing the communication distance.[25] In turn, this can increase the energy usage necessary to communicate. Since wireless connectivity can be a significant energy drain on data logger systems and can increase system cost significantly, it will not be appropriate for all applications. Because of this, it may be more prudent to implement wireless connectivity via an optional add-on board.

Near Field Communication (NFC) to the data loggers would also be an easy addition to improve system usability. Instead of manually pulling SD cards and saving the logs to a computer, a cell phone could download the data logs with the press of a button on the logger. Since the NFC would be essentially off except when in use, adding NFC capabilities would not significantly change the power draw of the data logger.

## 4.1 Conclusions

As this is a completely open-source project, collaboration, modification, and recreation are highly encouraged. While this manuscript can act as a blueprint for creating similar systems from scratch, interested researchers can obtain all the necessary files for reproduction can be found at github.com/Sawwave/pinecone-datalogger. The Extended Gerber PCB design files are ready to give directly to a PCB manufacturer, and the bill of materials list all hardware components needed to assemble the board.

# APPENDIX A Full Circuit Schematic

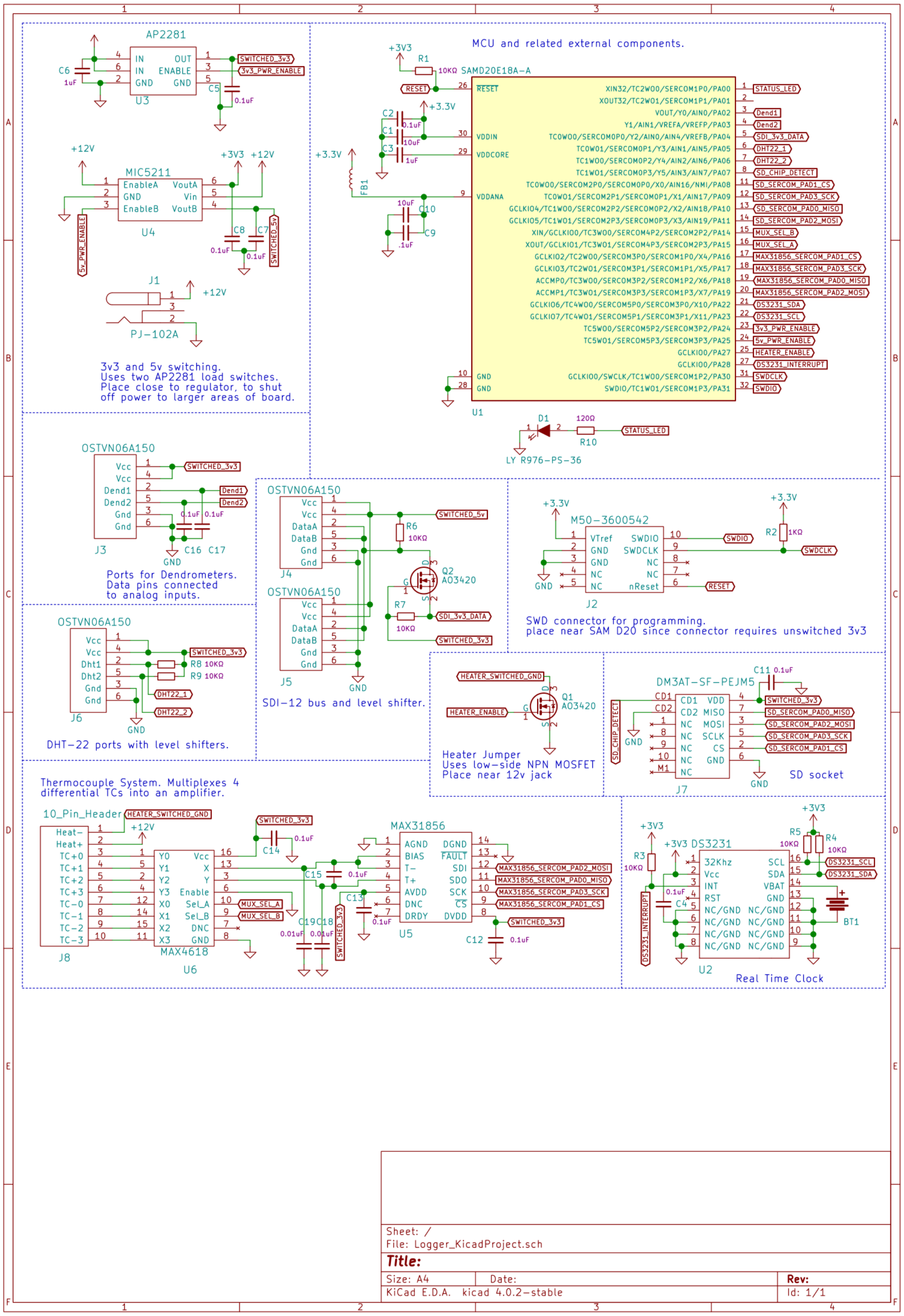



Figure A.1: Full schematic for the datalogger board.

# APPENDIX B 2016 Full Data Plots

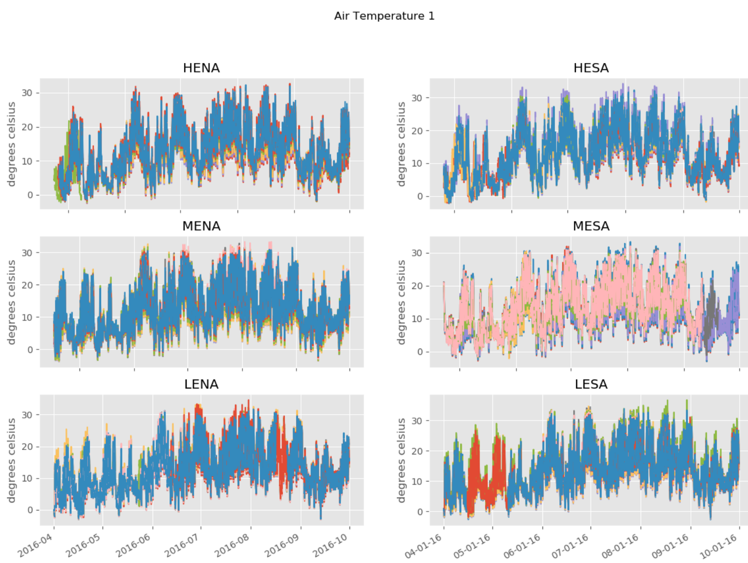



Figure B.1: Plots of air temperature from sensor 15cm above the ground surface during the 2016 field study.

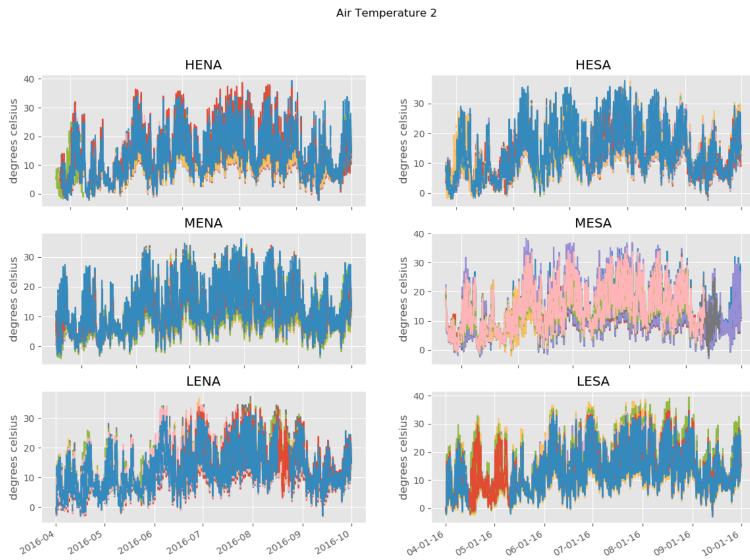



Figure B.2: Plots of air temperature from sensor 200cm above the ground surface during the 2016 field study.

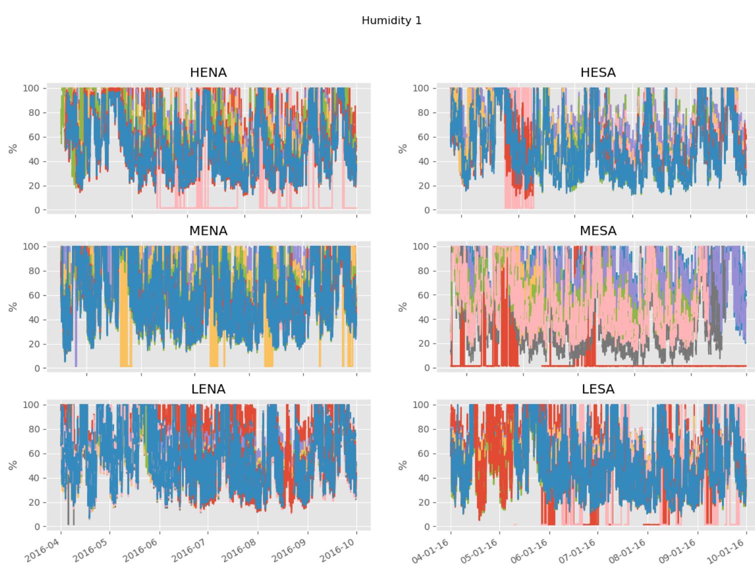



Figure B.3: Plots of relative humidity from sensor 15cm above the ground surface during the 2016 field study.

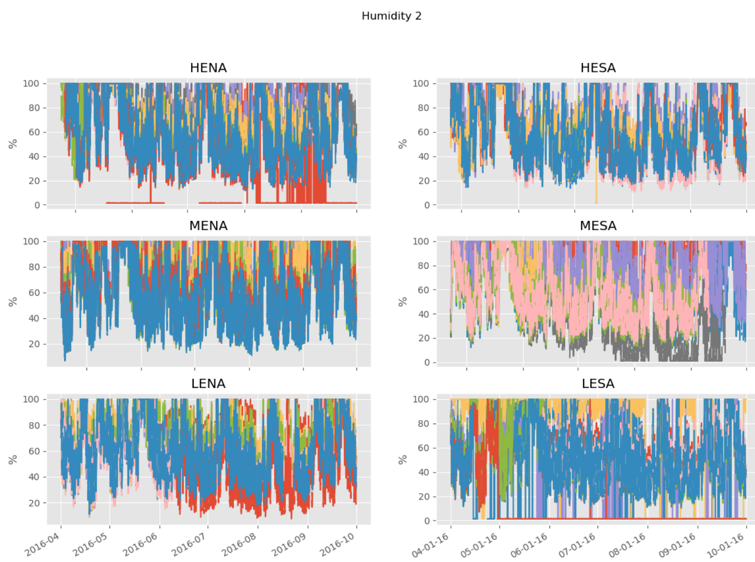



Figure B.4: Plots of relative humidity from sensor 200cm above the ground surface during the 2016 field study.

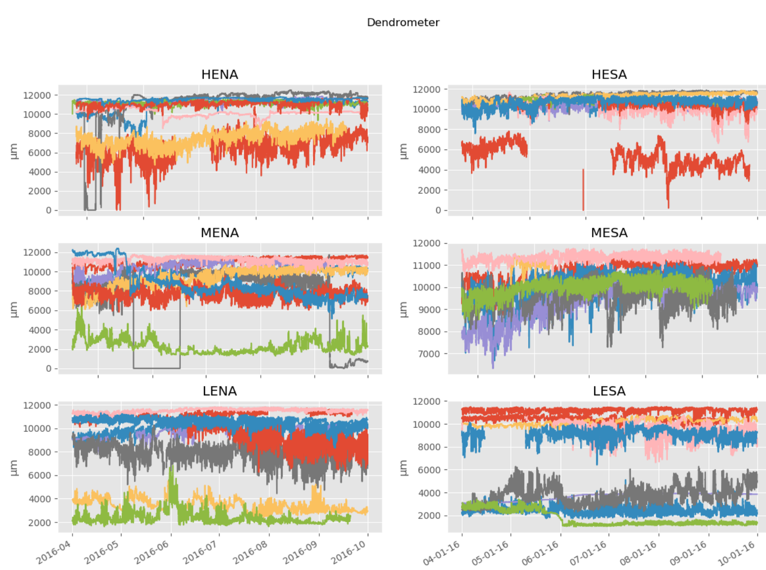



Figure B.5: Plots dendrometer readings during the 2016 field study. Values reflect the mean value of the two point dendrometers.

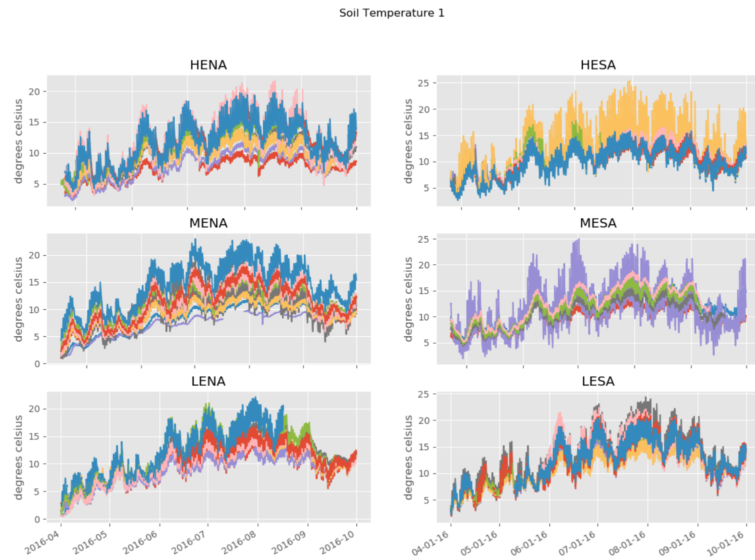



Figure B.6: Plots of subsurface soil temperature from sensor at depth of 5cm during the 2016 field study.

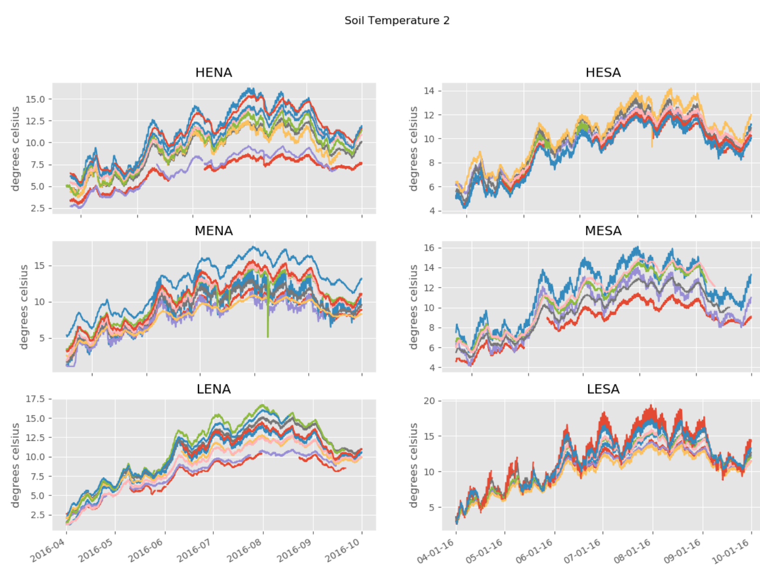



Figure B.7: Plots of subsurface soil temperature from sensor at depth of 50cm during the 2016 field study.

# BIBLIOGRAPHY

[1] "Pinoccio." [Online]. Available: https://www.crowdsupply.com/pinoccio/mesh-sensor-network

[2] H. R. Barnard, M. C. Findley, and J. Csavina, "Parduino: a simple and inexpensive device for logging photosynthetically active radiation," *Tree Physiology*, vol. 34, no. 6, p. 640–645, Jan 2014.

[3] "Sdi-12 data logging on beagle bone black," Jun 2017. [Online]. Available: https://liudr.wordpress.com/2017/06/30/sdi-12-data-logging-on-beagle-bone-black/

[4] G.-R. Walther, E. Post, P. Convey, A. Menzel, C. Parmesan, T. J C Beebee, J.-M. Fromentin, O. Hoegh-Guldberg, and F. Bairlein, "Ecological responses to recent climate change," *Nature*, vol. 416, pp. 389–95, 04 2002.

[5] A. Mishra, R. Singh, N. Raghuwanshi, C. Chatterjee, and J. Froebrich, "Spatial variability of climate change impacts on yield of rice and wheat in the indian ganga basin," *Science of The Total Environment*, vol. 468-469, 2013.

[6] S.-. S. Group. (2017, December) Sdi-12 a serial-digital interface standard for microprocessor-based sensors. [Online]. Available: http://www.sdi-12.org/current_specification/SDI-12_version-1_4-Dec-1-2017.pdf

[7] EnviroDIY, "Envirodiy/arduino-sdi-12," Oct 2018. [Online]. Available: https://github.com/EnviroDIY/Arduino-SDI-12

[8] "Fritzing." [Online]. Available: http://fritzing.org/home

[9] "Gerber format - ucamco - file formats." [Online]. Available: https://www.ucamco.com/en/file-formats/gerber

[10] "Advanced circuits." [Online]. Available: https://www.4pcb.com/

[11] brabl2, "brabl2/narcoleptic," Dec 2018. [Online]. Available: https://github.com/brabl2/narcoleptic

[12] S. S. O. Burgess, M. A. Adams, N. C. Turner, C. R. Beverly, C. K. Ong, A. A. H. Khan, and T. M. Bleby, "An improved heat pulse method to measure low and reverse rates of sap flow in woody plants," *Tree Physiology*, vol. 21, no. 9, p. 589–598, Jan 2001.

[13] F. H. Bormann and T. T. Kozlowski, "Measurements of tree growth with dial gage dendrometers and vernier tree ring bands," *Ecology*, vol. 43, no. 2, p. 289, 1962.

[14] Z. A. Holden, A. E. Klene, R. F. Keefe, and G. G. Moisen, "Design and evaluation of an inexpensive radiation shield for monitoring surface air temperatures," *Agricultural and Forest Meteorology*, vol. 180, p. 281–286, 2013.

[15] "Kicad eda," Dec 2018. [Online]. Available: http://kicad-pcb.org/

[16] "Seeed pcb and pcb assembly." [Online]. Available: https://www.seeedstudio.com/fusion.html

[17] "Reflow oven controller shield (arduino compatible)." [Online]. Available: http://www.rocketscream.com/shop/reflow-oven-controller-shield-arduino-compatible

[18] "Advanced software framework." [Online]. Available: http://asf.atmel.com/docs/latest/

[19] [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/60001504B.pdf

[20] J. Martin, N. Looker, Z. Hoylman, K. Jencso, and J. Hu, "Differential use of winter precipitation by upper and lower elevation douglas fir in the northern rockies," *Global Change Biology*, vol. 24, no. 12, p. 5607–5621, Jul 2018.

[21] "Slp010-12u datasheet." [Online]. Available: http://www.solarlandusa.com/product/11972/12040/77659.html

[22] M. R. Maghami, H. Hizam, C. Gomes, M. A. Mohd Radzi, S. Rezadad, and S. Hajighorbani, "Power loss due to soiling on solar panel: A review," *Renewable & Sustainable Energy Reviews*, vol. 59, pp. 1307–1316, 01 2016.

[23] "Sunguard." [Online]. Available: https://www.morningstarcorp.com/products/sunguard/

[24] A. Saari, S. A. Hinzey, J. R. Frigo, M. C. Proicou, and L. Borges, "Using sdi-12 with st microelectronics mcu's," 9 2015.

[25] K. A. Ahmad, M. S. Salleh, J. D. Segaran, and F. R. Hashim, "Impact of foliage on lora 433mhz propagation in tropical environment," *American Institute of Physics Conference Proceedings*, vol. 1930, no. 1, 2018.