Ehlers, Jamison

jehlers@mtech.edu

Abstract

A quick summary of the components and code currently on the HAST system

High altitude soil testing System documentation

# Table of Contents

|  |  |
| --- | --- |
| [Necessary Libraries](#_Necessary_Libraries) | Page 2 |
| [Components in use](#_Components_in_Use) | Page 3 |
| [Main Functions](#_Main_functions) | Page 9 |
| [Secondary Functions](#_Secondary_functions) | Page 10 |
| [Power consumption model](#_Power_consumption_model) | Page 11 |
| [Final Notes](#_Final_Notes) | Page 12 |
| [APPENDIX](#_APPENDIX) | Page 14 |
| Supply List/Order Links | Page 15 |
| Code flowchart | Page 16 |
| Picture of current prototype | Page 17 |

# Necessary Libraries

The HAST soil temperature monitor uses a few custom Arduino libraries, as well as a specific board library. Below is a list of all libraries in use, as well as a description of their use and any associated URL, should you need to set up the environment. All of these libraries were found in the Arduino library manager at the time of writing. The IridiumSBD library has a GitHub page, which can be found [here](https://github.com/mikalhart/IridiumSBD).

Table 1: List of Arduino libraries

|  |  |
| --- | --- |
| **Name** | **Description** |
| Arduino Zero (Native USB port) | Board library used to program RocketScream microcontroller |
| SerialFlash | Flash memory control, turns off flash while in sleep mode |
| RTCZero | Real time clock access |
| SPI | SPI communication protocol library for SD card |
| SD | SD card read/write |
| SHT1x sensor library for ESPx (SHT1x) | Soil temperature and moisture sensor library |
| IridiumSBD | Satellite communication library |

# Components in Use

There are many components on the current iteration of the HAST Sensor board. Here, we’ll list off the parts, their functions and some overall specifications.

## RocketScream

The RocketScream is an Arduino based microcontroller, driving the board itself. It provides the libraries necessary for using the SD the soil sensors, RockBlock satellite modem and SD card reader.

Table 2: Specifications of RocketScream Microcontroller

|  |  |
| --- | --- |
| Specifications | |
| Processor | 32-bit ARM Cortex M0+ ATSAMD21G18A |
| Sleep Mode Power Draw | 20uA |
| Awake Mode Power Draw | 12mA |
| Arduino Board Option | Arduino/Genuino Zero (Native USB Port) |
| Clock | 48 MHz |
| Operating voltage | 3.3V |
| On-Chip Real Time Clock |  |
| 2x19 Pin headers |  |
| Dimensions | 25.4 mm x 55.88 mm (1.0” x 2.2”) |

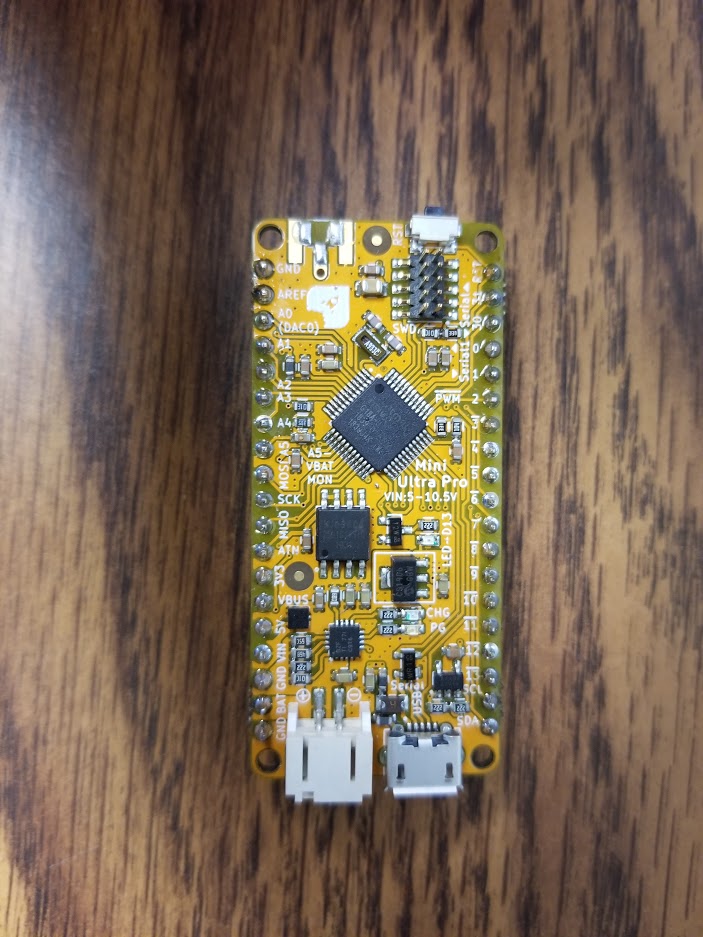


Figure 1: Picture of RocketScream on hand

IMPORTANT NOTE: When uploading new code to the RocketScream, make sure there is a delay at the beginning of the script. This prevents the microcontroller from bricking itself during any USB detachment.

## RockBlock

The RockBlock Mk2 is a satellite uplink, configured to send short “Twitter length” emails via satellite to a specified email address. In order to use the RockBlock you must pay a line rental fee, as well as credits for every 50 bytes. The RockBlock itself can send messages to the specified email address of up to 340 bytes. The log-in information is supplied at the end of this document. Handle this component with care, as the patch antenna at the top of the device is very static sensitive.

Table 3: Specifications for RockBlock Module

|  |  |
| --- | --- |
| Specifications | |
| Operating Voltage | 5V |
| Estimated active current draw | 100 mA minimum, 400 mA maximum |
| Dimensions | 76.0 x 51.5 x 19.0 mm |
| Communication protocol | UART |
| Built in patch antenna |  |
| Weight | 76 g |
| Line rental fee/month | $16.00 |
| 1x6 and 1x8 pinheaders |  |



Figure 2: RockBlock Satellite Uplink with patch antenna outlined

## SD Card Adapter

The SD card adapter is for easy access to storage as well as connections to the board. Currently, to attach the SD card adapter to the board, you must de-solder the connections existing on the adapter to solder onto main project board. It is recommended, however that should another board be designed that this be integrated into the main board itself or design a daughter board for later applications.

Table 4: SD Card Adapter Specifications

|  |  |
| --- | --- |
| Specifications | |
| Typical operating voltage | 4.8V |
| Estimated on-current draw | ~100-300 mA (When using SD protocol, not SPI, according to documentation) |
| SD Card Size | 8 GB |
| Dimensions | 41.8 x 23.5 x 1.56 mm |

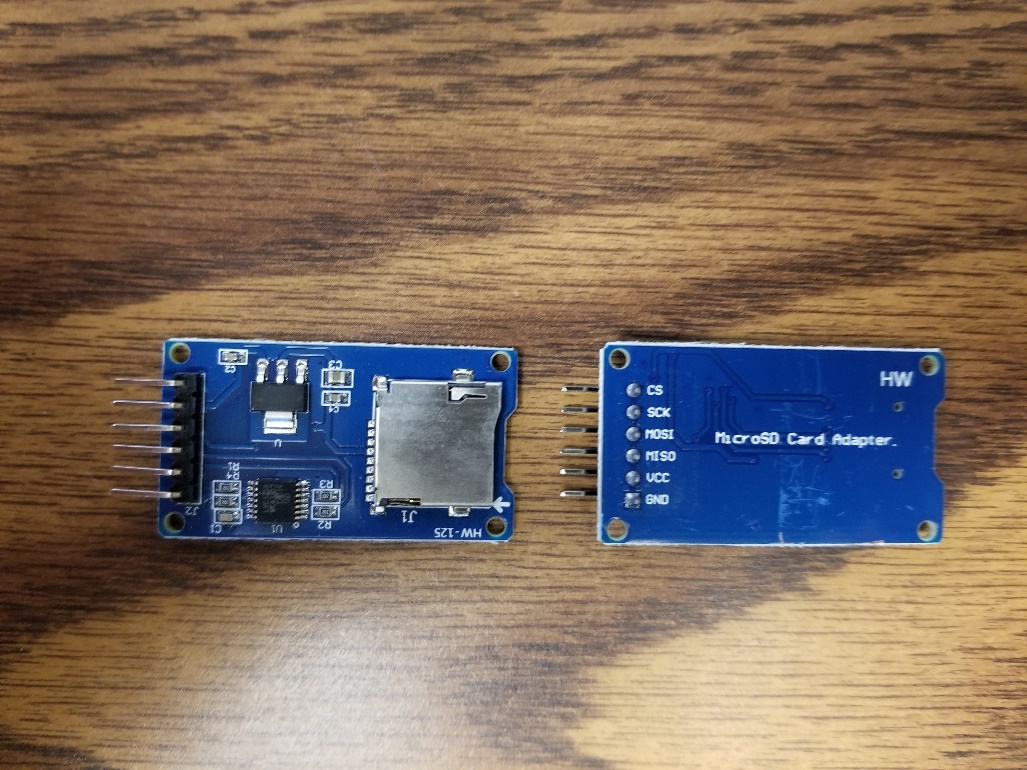


Figure 3: SD Card Adapter

## Lithium Thionyl Batteries

Three 3.6V D-Cell sized Lithium Thionyl Batteries are used on the board. Two of them are configured in series providing 7.2 V to most of the system, while one is used to power just the RocketScream. Each are rated for 19000 milliamp-hours.



Figure 4: Three Lithium-Thionyl Batteries

## Soil sensors

Currently, the soil sensors in use on the board are SHT-10 soil temperature and humidity sensors. They can read the temperature in as either Farenheight or Celsius, and the humidity is in relation to pure water, ie zero percent is bone-dry, while one hundred percent is in water.

Table 5: SHT-10 soil sensor specifications

|  |  |
| --- | --- |
| Specifications | |
| Operating voltage | 3.3V |
| Estimated power consumption | 2.45 uA |
| Temperature Range | -40°C - 120°C |
| Cable length | ~1m |
| Body Dimensions | 14 mm diameter, 50 mm long |

Red – Vcc (3.3V)

Black – Ground

Green - Data

Yellow – Clock

Figure 5: Wiring diagram for SHT-10 soil sensors

## Electronics

This section is dedicated to all secondary electronics on the board, such as the resistors, capacitors, voltage regulators and optoisolators.

**Resistors**

**All resistors used on the board are 0805 SMT package. They have values of 3.3kΩ and 330Ω. Below is a table of which resistors are which value.**

Table 6: Resistor Table

|  |  |
| --- | --- |
| **PULLUP1 – PULLUP4** | **3.3kΩ** |
| **CLK\_PULL** | **3.3kΩ** |
| **PULLUP5-PULLUP8** | **330Ω** |

**Capacitors**

**All capacitors used on the board are 0805 SMT and 20uF.**

**Regulators**

**On the board, there are two 3.3V LDO regulators and two 5V LDO regulators. The 3V regulators are used to power the soil sensors and the optocouplers. The 5V regulators are used for powering the SD card adapter and the RockBlock itself.**

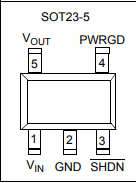
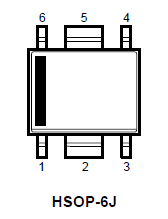
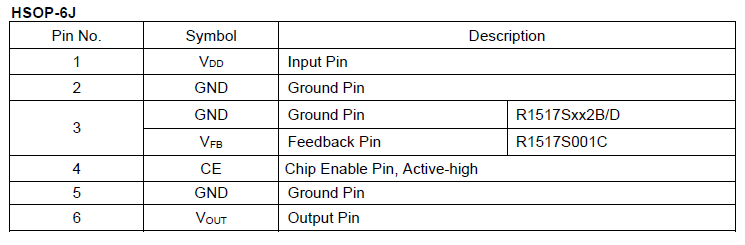
 

Figure 6: 5V LDO with pinout diagram Figure 7: 3.3V LDO Pinout Diagram

Table 7: 5V LDO Pinout Information



**Optocouplers**

The optocouplers are phototransistors, used to isolate the RocketScream from the RockBlock. This prevents the RockBlock leaking current into the RocketScream, and vice versa. Specifications can be found in the datasheet.

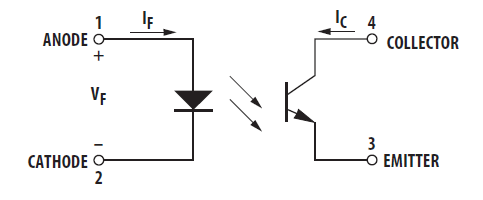


Figure 8: Pinout diagram for optocoupler

## **Case components**

**NEMA Case**

The case used is a ML-47F NEMA case, with IP68 rating.

Table 8: NEMA case specifications

|  |  |
| --- | --- |
| **Specifications** | |
| NEMA Case ### | ML-47F |
| IP Rating | IP68 |
| Dimensions | 7.63 x 4.63 x 3.09 in  193.80 x 117.60 x 78.49 mm |

**Cable glands**

The cable glands used on the system are PG7 cable glands, for 3.1-6.6 mm cable

|  |  |
| --- | --- |
| ***Specifications*** | |
| IP Rating | IP68 |
| Outer diameter | 12 mm |
| Inner diameter | 3.1 - 6.6 mm |

## **Other Components**

The final components are the screw-down terminals, power switch and battery cell holders. The screw-down terminals allow for easy changing of sensors and for security in the field, so the sensors do not disconnect from the board. The pins have a 2.54 mm pitch. The power switch controls input power from the multiple cell setup and the battery cell holders connect the batteries to the board from underneath.

# **Main functions**

### Initialization

Here, we define all our objects needed for the rest of the script, import all our needed libraries, as well as initialize our SD card files and serial UART line with the RockBlock.

|  |  |
| --- | --- |
| **Variable Name** | **Function** |
| DATAX\_PIN | Pins used as data inputs from soil sensors |
| CLK\_PIN | Pin used as the shared clock between all of the soil sensors |
| SENSE\_PWR | Chip select for regulator that supplies sensors |
| RB\_PWR | Chip select for regulator that supplies RockBlock |
| SD\_PWR | Chip select for regulator that supplies SD card adapter |
| COMM\_PWR | Chip select for regulator that supplies communication scheme |
| dataFile | .txt file that contains measurements |
| posFile | .txt file that contains current position on where to read measurements from when sending message |
| READINGS\_PER\_MSG | How many readings to take before sending message |
| MSG\_END | Delimiter for RockBlock message, initially set as ‘;’ |
| CS\_PIN | Chip select for SD card adapter |
| SERIAL\_BAUD | Baud rate for serial line to host computer (used for debugging) |
| RB\_BAUD | Baud rate for serial UART line to RockBlock (needs to be 19200) |
| SCANS\_PER\_DAY | Number of scans to take per day |
| SHT1 – SHT4 | Individual objects for each sensor |
| SENSORS[] | List of sensor objects |
| NUM\_SENSE | Number of sensors in use |
| rtc | Real-Time clock object |
| SEC, MIN, HOUR | Initial time on startup |
| DAY, MON, YEAR | Initial date on startup |
| PIN\_NUMBER | Current number on which pin being disabled for sleep mode |
| ALARM\_VAR | Allows code to get into main loop for operation |
| SCANS\_TAKEN | Number of scans taken since last message sent |
| SEND\_ATTEMPTS | Number of attempts to send a message |
| message | Current message attempting to be sent |

### Setup

In setup, we initialize the real time clock, and detach the USB from the host PC.

|  |  |
| --- | --- |
| **Function Name** | **Purpose** |
| rtc.begin() | Starts real time clock |
| rtc.setTime(HOUR,MIN,SEC) | Sets real time clock with current initial time |
| rtc.setDate(DAY,MON,YEAR) | Sets real time clock with current initial date |
| rtc.setAlarmSeconds() | Sets alarm to go off in so many seconds |
| rtc.enableAlarm(rtc.MATCH\_MMSS) | Enables alarm to go off at set time |
| rtc.attachInterrupt(alarmMatch) | Attaches interrupt to function alarmMatch |
| USBDevice.detach() | Detaches USB from host device |
| rtc.standbyeMode() | Puts real time clock in standbye mode |

### Loop

Main loop, only does anything if ALARM\_VAR = 1. Reattaches USB to host device, scans the soil, initializes the SD card, writes SD card, and sees if a message needs to be sent.

|  |  |
| --- | --- |
| **Function Name** | **Purpose** |
| scan() | Takes scan of soil temperatures and moistures |
| initSD() | Initializes SD card for write and read |
| writeSD() | Writes data to SD card file |
| initRB() | Initializes RockBlock for message send |

## Secondary functions

### pinOff

Sets each pin as an input minus the few we need for operation to meet power requirements.

### Scan

Takes a scan of soil temperature and moisture. The for loop goes through each sensor and takes individual readings, stores the readings as strings into the message string.

### DateTime

Inputs time stamp into message as string

### makeItNice

Changes time stamp into regular length

### initSD

Initializes and starts up SD card for write and read

readSD

Reads off data from SD card, not needed in final script.

### writeSD

Writes soil measurement to SD card data file.

### getPos

Returns current position on SD card to avoid overwriting

### setPos

Sets the current position for the SD card write after sending a message.

### initRB

initializes and wakes up RockBlock to return signal strength and decide if we send or not.

### sendMessage

Attempt to send message

### sendData

Refreshes message to send with current position and most recent reading.

# Power consumption model

To help save on costs and give and approximate time frame to the project, a model for power consumption was created in Python 3 using the Simpy framework. You’ll find the Python file on the GitHub for the project.

A few problems with the model were that the snow accumulation is not accurate. It randomly adds snow to a container object in Simpy, and randomly decreases the amount of snow in the same container. This starts to look more like an exponential curve, rather than something that looks similar to a sine wave. One suggested fix was to search for SNOTEL data for Goat Flat and use that as a reference for the snowfall in some way.

Another problem was that when the capacity of one of the battery containers reaches zero, or tries to use more power than that is in the container, the system sits and waits and doesn’t actually go to zero. To fix this, when the code attempts to grab more than that is in the container, just set the container equal to zero.

# Final Notes

## Hardware

As for final results, when I was testing the current draw when the system was in sleep mode, I noticed that something other than the RocketScream was drawing power, around 1.9 mA. I theorize this to be a combination of the voltage regulators and the pullup resistors for each of the data lines.

The quiescent current for the 5V regulators is 18 uA, while the 3.3V regulators have a quiescent current of 100 uA, with two of each regulator, this comes together for a total of 208uA. Now, if you calculate the current through each of the 10 kΩ pull-up resistors (A total of 5 on the board) and sum them with the previous total, the current comes to around 1.8 mA.

Should these components be the source of the current draw, I recommend the following changes to the board.

1. Add a switching circuit to the general power side of the board, allowing for complete isolation of the components from their source, controlled by the RocketScream.
2. Implement an isolating circuit for each of the data lines to the sensors, isolating them until needed.
3. Replacing the current 3.3V regulator to one with a lower quiescent current or instead of power coming directly from the battery, tier the power supplies, to where the 5V regulator for the SD card also powers the regulators for the optocouplers and the sensors themselves.

## Software

The string of data being sent to the RockBlock needs to be converted into a more compressed format. Initially, this was to be done with the scheme described in the Data Encoding section of this document. However, I think I may have found a better way to encode the data, while being able to encode more.

The method I found is called SBUS encoding/decoding. SBUS is a form of inverted UART communication. The way it works is by using the equations on the next page to encode data into bytes, essentially what RockBlock is suggesting on their webpage [here](https://docs.rockblock.rock7.com/docs/compressing-your-data). To encode the data, first you use equation 1, where the “channel” is the data you are encoding, while chan1BS and chan2BS are your bitshift values for where you are moving the data. For the implementation I am purposing, they would be multiples of three. Chan1 and chan2BM are your bitmask values, effectively selecting which data in the message string you are encoding. Then, after all the “channels” have been encoded into “bytes”, you would then feed the encoded char array into the sendSBDText function. Then, on the receiving side, you would set up an algorithm to decode the values. The algorithms are found in Equations 1 and 2 respectively. However, in SBUS encoding, there is a endian swap, from little endian to big endian. This is for hardware implementation, so we wouldn’t need the endian swap.

Equation 1: SBUS Encoding algorithm

|  |
| --- |
|  |

Equation 2: SBUS Decoding algorithm

|  |
| --- |
|  |

# Data Encoding

In order to maximize the data use when sending the readings over the satellite connection, the data string needs to be encoded. Currently, a switch case has ben set up to encode every two characters of the message string into a new string. This switch case is not finished yet. Currently, the switch case does not save characters, but

The switch case currently is very similar to hex. Below is an example table of the number inputted into the switch case, then the hex number representation, then what the encoding takes it as.

|  |  |  |
| --- | --- | --- |
| Decimal Number | Hex Number | Encoding Number |
| 9 | 0x09 | 09 |
| 10 | 0x0A | 0a |
| 11 | 0x0B | 0b |
| 12 | 0x0C | 0c |
| 13 | 0x0D | 0d |
| 14 | 0x0E | 0e |
| 15 | 0x0F | 0f |
| 16 | 0x10 | 0g |
| 17 | 0x11 | 0h |
| 18 | 0x12 | 0i |
| 19 | 0x13 | 0j |
| 20 | 0x14 | 0k |
| 21 | 0x15 | 0l |
| 22 | 0x16 | 0m |
| 23 | 0x17 | 0n |
| 24 | 0x18 | 0o |
| 25 | 0x19 | 0p |
| 26 | 0x1A | 0q |
| 27 | 0x1B | 0r |
| 28 | 0x1C | 0s |
| 29 | 0x1D | 0t |
| 30 | 0x1E | 0u |
| 31 | 0x1F | 0v |
| 32 | 0x20 | 0w |
| 33 | 0x21 | 0x |
| 34 | 0x22 | 0y |
| 35 | 0x23 | 0z |
| 36 | 0x24 | 0A |
| 37 | 0x25 | 0B |
| 38 | 0x26 | 0C |
| 39 | 0x27 | 0D |
| 40 | 0x28 | 0E |
| 41 | 0x29 | 0F |
| 42 | 0x30 | 0G |
| 43 | 0x31 | 0H |
| 44 | 0x32 | 0I |

# APPENDIX

Table 9: Component List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | **Manufacturor #** | **Distributor #** | **Distributor** | **# Per Unit** |
| [1 x 19 Pinheaders](https://www.digikey.com/product-detail/en/sullins-connector-solutions/PPTC191LFBN-RC/S7017-ND/810157) | PPTC191LFBN-RC | S7017-ND | Digi-Key | 2.74 |
| [3.3V LDO](https://www.mouser.com/ProductDetail/Microchip-Technology/MCP1755T-3302E-OT?qs=OCR3PNzn1cdSAPBsg9LlMg%3D%3D) | MCP1755T-3302E/OT | 579-MCP1755T-3302EOT | Mouser | 2 |
| [3.6V Battery](https://www.batteryjunction.com/xl-205f.html) | N/A | XENO-XL-205F-OPT | Battery Junction | 3 |
| [5V LDO](https://www.mouser.com/ProductDetail/Ricoh-Electronic-Devices-Company/R1517S502B-E2-KE?qs=55YtniHzbhCGtUA09e4RAQ%3D%3D) | R1517S502B-E2-KE | 848-R1517S502BE2KE | Mouser | 2 |
| [8 GB Industrial SD Card](https://www.amazon.com/Kingston-Digital-8GB-Industrial-SDCIT/dp/B01DOFCPNW/ref=sr_1_6?dchild=1&keywords=Industrial+SD+Cards&qid=1607016348&sr=8-6) | N/A | N/A | Amazon | 1 |
| [Battery Cell Holder](https://www.digikey.com/en/products/detail/mpd-memory-protection-devices/BHD-2/1640078) | BHD-2 | BHD-2-ND | Digi-Key | 3 |
| [PG7 Cable Glands](https://www.amazon.com/mxuteuk-lengthening-Connectors-Protectors-Waterproof/dp/B083NP966N/ref=sr_1_12?dchild=1&keywords=pg7+cable+gland+waterproof&qid=1591221043&s=hi&sr=1-12) | N/A | N/A | Amazon | 4 |
| [Capacitor (10uF)](https://www.digikey.com/products/en?keywords=399-4925-2-ND) | C0805C106K8PACTU | 399-4925-2-ND | Digi-Key | 4 |
| Mk 3 PCB | N/A | N/A | JLCPCB | 1 |
| [Resistor (150)](https://www.digikey.com/product-detail/en/panasonic-electronic-components/ERA-6AEB151V/P150DATR-ND/1465728) | ERA-6AEB151V | P150DATR-ND | Digi-Key | 4 |
| [Resistor (1k)](https://www.digikey.com/product-detail/en/panasonic-electronic-components/ERA-6AEB102V/P1-0KDATR-ND/1465748) | ERA-6AEB102V | P1.0KDATR-ND | Digi-Key | 5 |
| [Resistor (3.3k)](https://www.digikey.com/product-detail/en/panasonic-electronic-components/ERA-6AEB332V/P3-3KDATR-ND/1465761) | ERA-6AEB332V | P3.3KDATR-ND | Digi-Key | 5 |
| [Resistor (330)](https://www.digikey.com/product-detail/en/panasonic-electronic-components/ERJ-PB6D3300V/P20962TR-ND/6213745) | ERJ-PB6D3300V | P20962TR-ND | Digi-Key | 4 |
| [Rock Block Mk2](https://www.rock7.com/shop-product-detail?productId=46) | N/A | N/A | Rock Block | 1 |
| [Mini Ultra Pro V3 w/out Radio](https://www.rocketscream.com/blog/product/mini-ultra-pro-v3-without-radio/) | N/A | N/A | Rocket Scream | 1 |
| [SD Card Reader](https://www.amazon.com/HiLetgo-Adater-Interface-Conversion-Arduino/dp/B07BJ2P6X6/ref=sr_1_1_sspa?dchild=1&keywords=arduino+sd+card+module&qid=1594656884&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzUjg1U05ZMjE1Q0NLJmVuY3J5cHRlZElkPUEwODM0NTQ5MTU2UjVNQkc5VDdQViZlbmNyeXB0ZWRBZElkPUEwNTQ0Mzg1MU02S1JOTEMyTFZOSyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=) | N/A | N/A | Amazon | 1 |
| [SHT-10](https://www.digikey.com/en/products/detail/adafruit-industries-llc/1298/7241454?s=N4IgjCBcoLQdIDGUAuAnArgUwDQgPZQDa4ArAEwAcIAugL517nHjkCc19QA) | 1298 | 1528-2209-ND | Digi-Key | 4 |
| [Broadcom Optoisolator](https://www.digikey.com/product-detail/en/broadcom-limited/HCPL-181-00BE/516-1787-2-ND/1234888) | HCPL-181-00BE | 516-1787-2-ND | Digi-Key | 2 |
| [Terminal Blocks](https://www.mouser.com/ProductDetail/Phoenix-Contact/1725672?qs=Ul7CXFMnlWXX93F9bOV2pQ%3D%3D) | 1725672 | 651-1725672 | Mouser | 4 |
| [TS Optoisolator](https://www.digikey.com/product-detail/en/taiwan-semiconductor-corporation/TPC817D-C9G/TPC817DC9G-ND/7359671) | TPC817D C9G | TPC817DC9G-ND | Digi-Key | 0 |
| [SPST DIP switch](https://www.digikey.com/product-detail/en/cts-electrocomponents/206-2/CT2062-ND/20733) | CTS Electrocomponents | CT2062-ND | Digi-Key | 1 |
| [NEMA Case](https://www.polycase.com/ml-47f) | polycase | ML-47F\*1508 | polycase | 1 |

Table 10: Login information for RockBlock management (see additional RockBlock documentation for more info), Outlook account, and GitHub account

|  |  |  |  |
| --- | --- | --- | --- |
| **Website** | **Username** | **Password** | **URL** |
| RockBlock Manager | [bhill@mtech.edu](mailto:bhill@mtech.edu) | TechDigger2020 | <https://rockblock.rock7.com/Operations> |

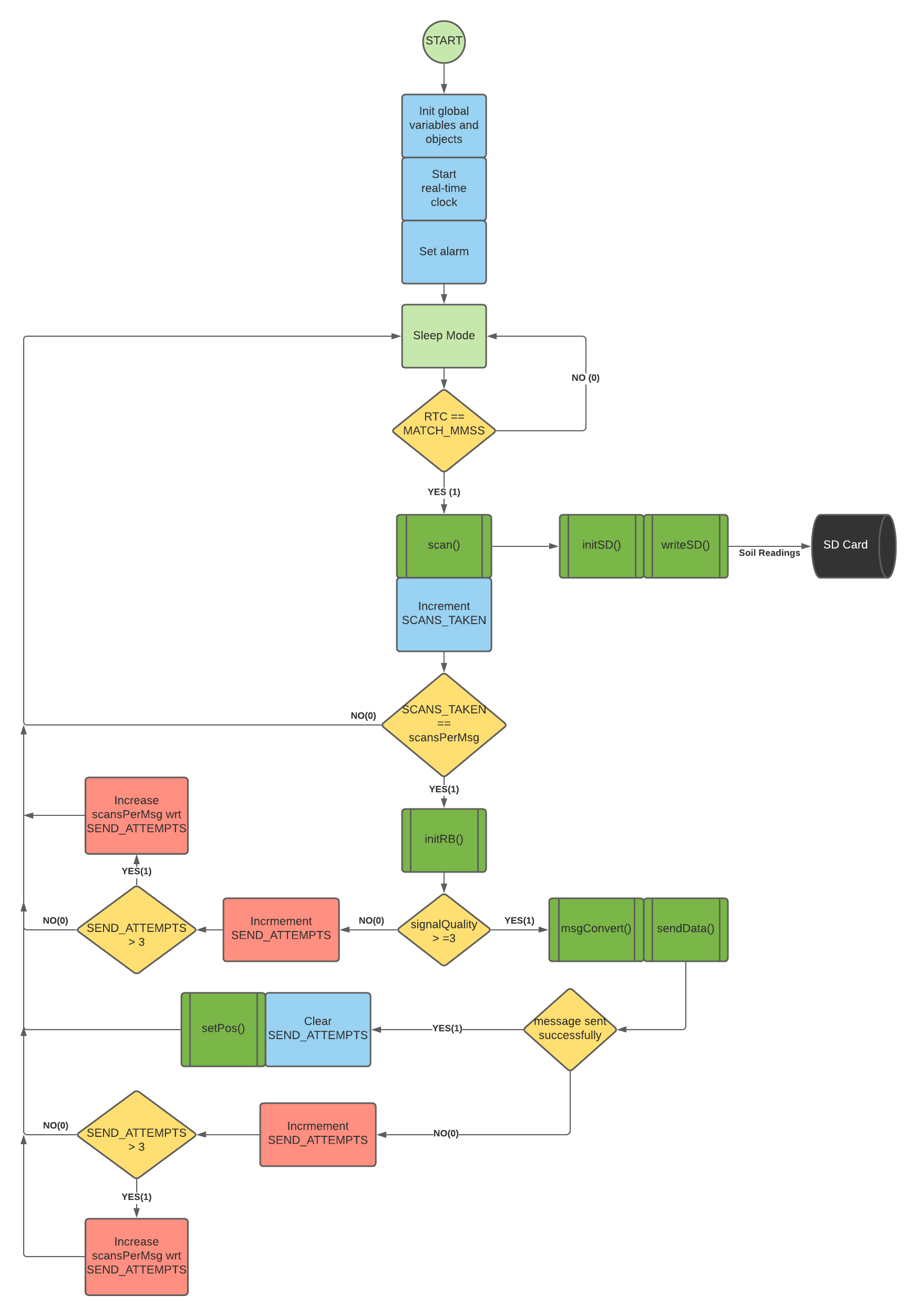


Figure 9: Code flowchart

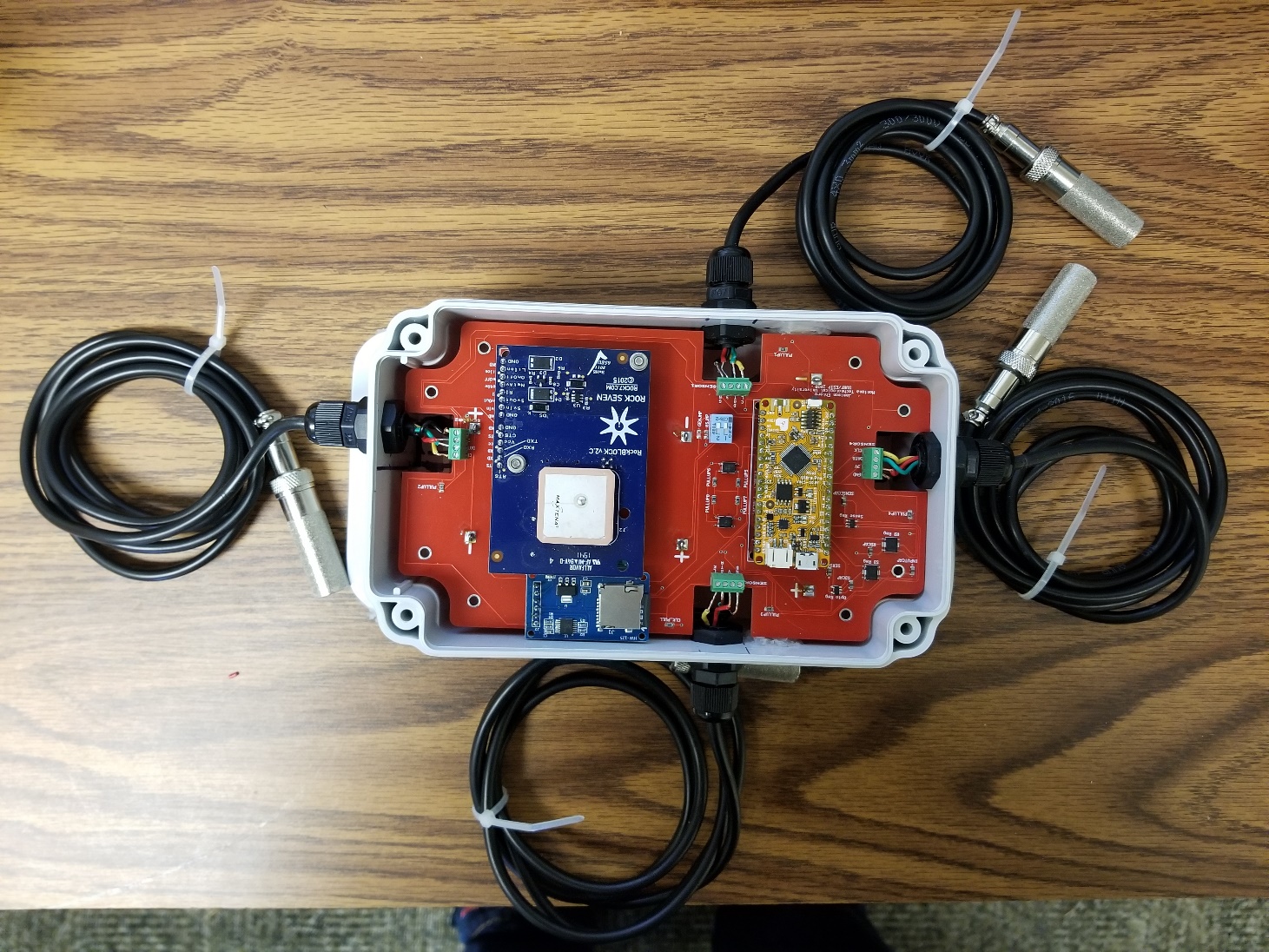
**

Figure 10: System prototype