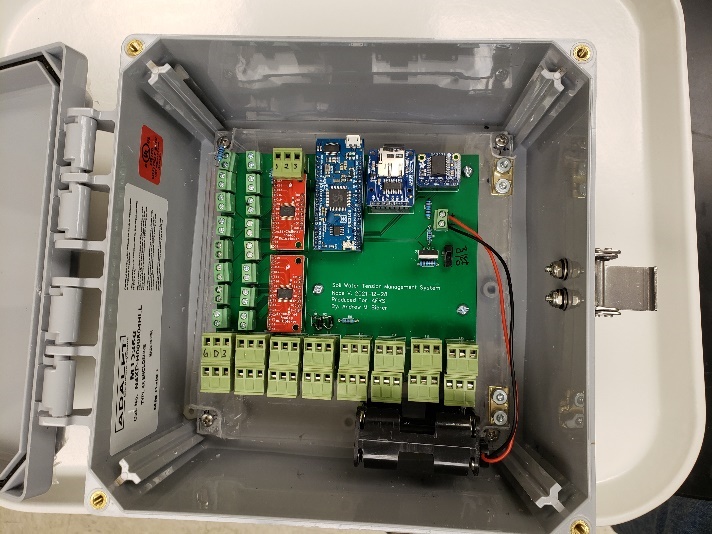
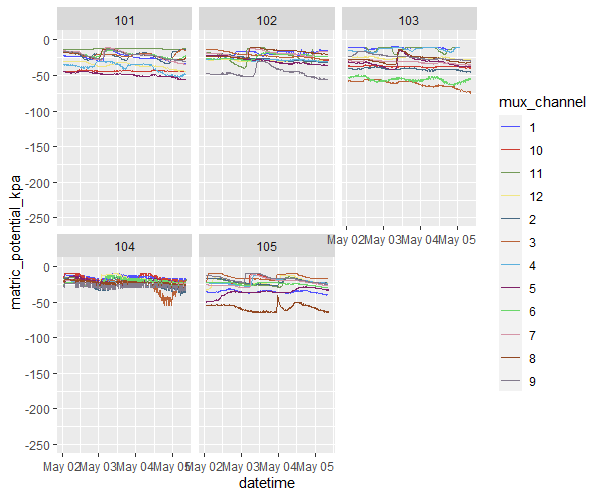
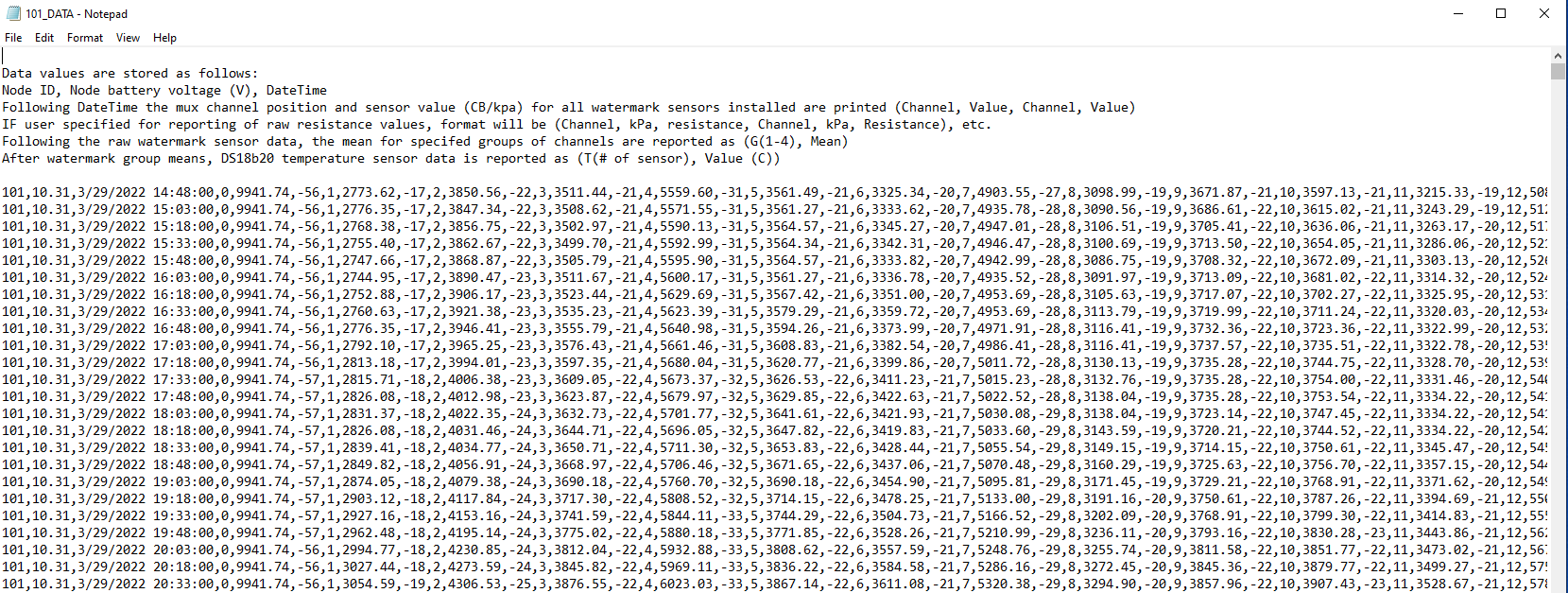
*Open\_Irr*

Soil Water Tension Management System

Operation Manual V. 2022.05.06





Open\_Irr Node diagram………………………………2

Open\_Irr Gateway diagram…………………………3

Specifications………………………………………………4

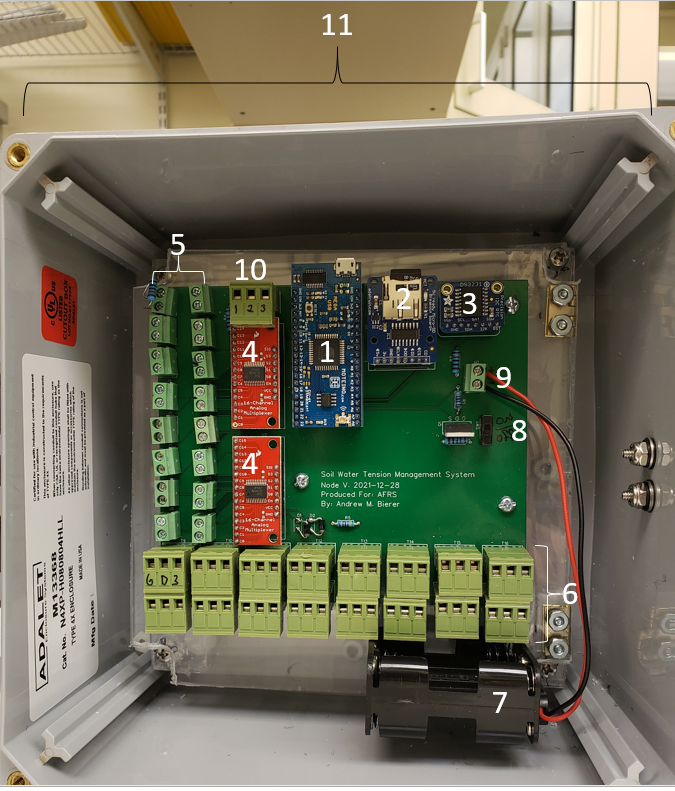
Warnings and Cautions………………………………..4

Node Operation……………………………………………5

Gateway Operation……………………………………...10

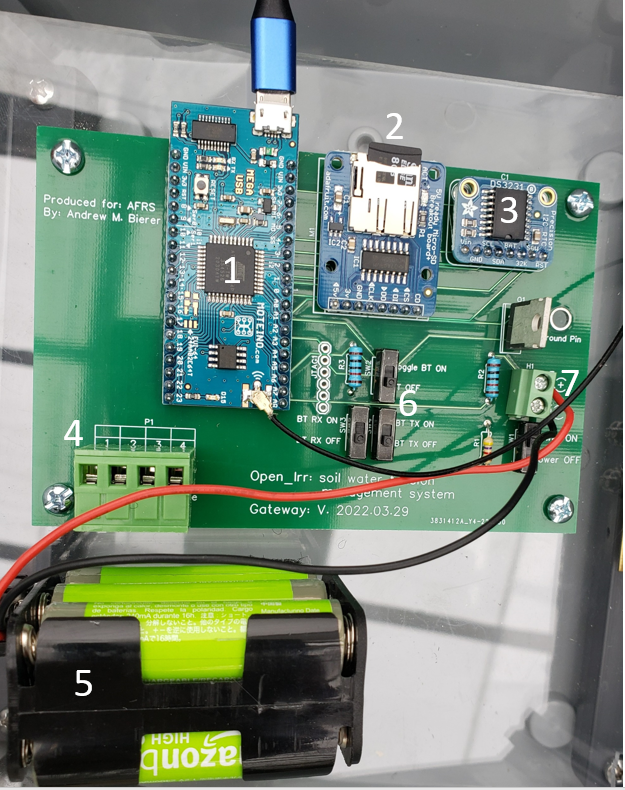
Maintenance…………………………………………………12

Troubleshooting…………………………………………….12

Open\_Irr Node Diagram

1. Moteino-Mega usb microcontroller core
2. Micro-sd card breakout module
3. Real Time Clock (RTC) breakout module
4. 16 channel Multiplexor(s)
5. 2-Pin terminal block connectors for WaterMark sensors
6. 3-Pin Male-Female terminal block connectors for Dallas 1-wire DS18B20 temperature sensors
7. Battery pack holding 6 AA batteries
8. Power switch, UP = ON | DOWN = OFF
9. 2-Pin terminal block connector for battery pack
10. 3 or 4-Pin Male-Female terminal block connector for Low-level (3.3V) output / irrigation automation
11. Environmental enclosure housing

Open\_Irr Gateway Diagram

1.  Moteino-Mega usb microcontroller core
2. Micro-sd card breakout module
3. Real Time Clock (RTC) breakout module
4. 4-Pin Male-Female terminal block connector for Low-level (3.3V) output / irrigation automation
5. Battery pack holding 6 AA batteries
6. Bluetooth operation switches, not operational at time of writing
7. 2-Pin terminal block connector for battery pack

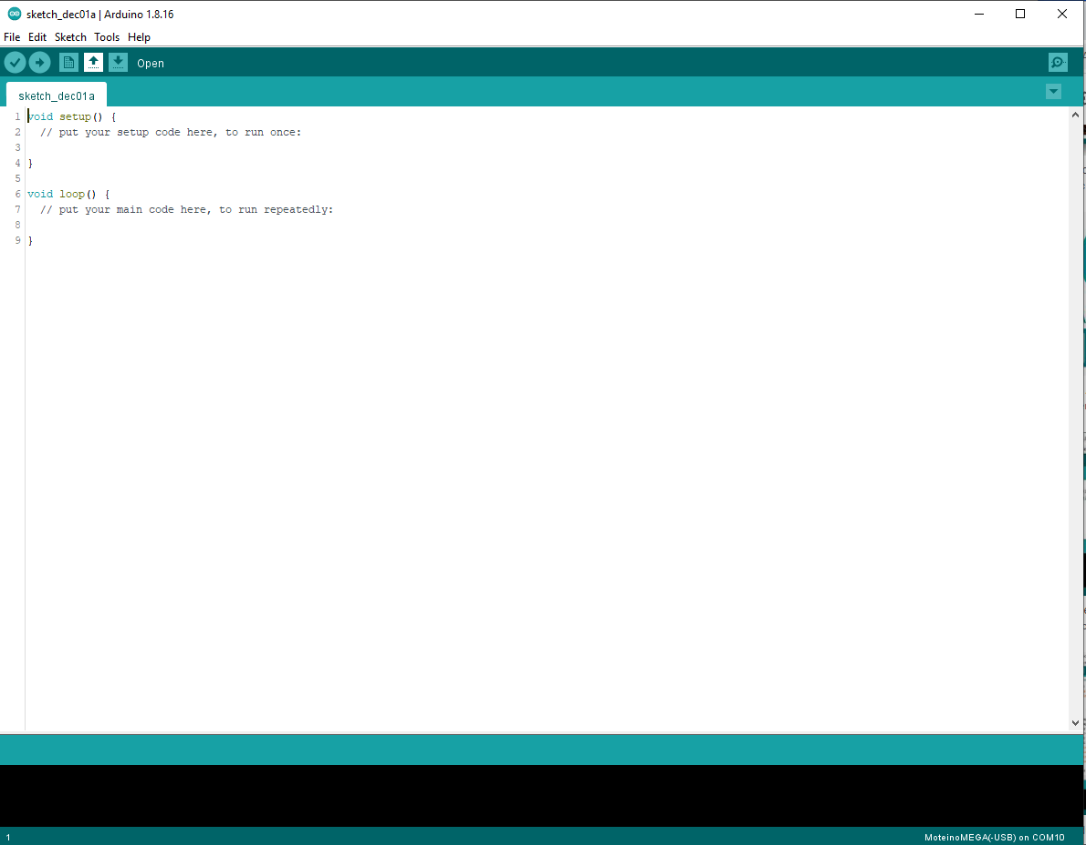
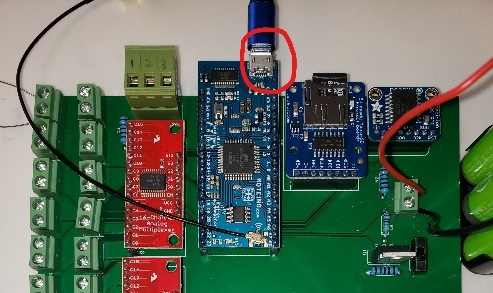
Open\_Irr Specifications

|  |  |
| --- | --- |
| Soil Moisture Sensor Type | Analog WaterMark® matric potential |
| Temperature Sensor Type | Dallas 1-wire (DS18B20) |
| Soil Moisture Sensor Range | Resolution | 0 to -239 kPa | 1kPa |
| Temperature Sensor Range | Resolution | Accuracy | -55 to 125°C | 9 to 12 bit, 9 bit default | ± 0.5°C from -10 to 85°C |
| Input | DC 3.3V to 16V |
| Output | DC 3.3V |
| Frequency | 915 MHz |
| Transmission Power | 0 to 20 dBm, 20dBm default |
|  |  |
|  |  |
|  |  |
|  |  |

Warnings and Cautions

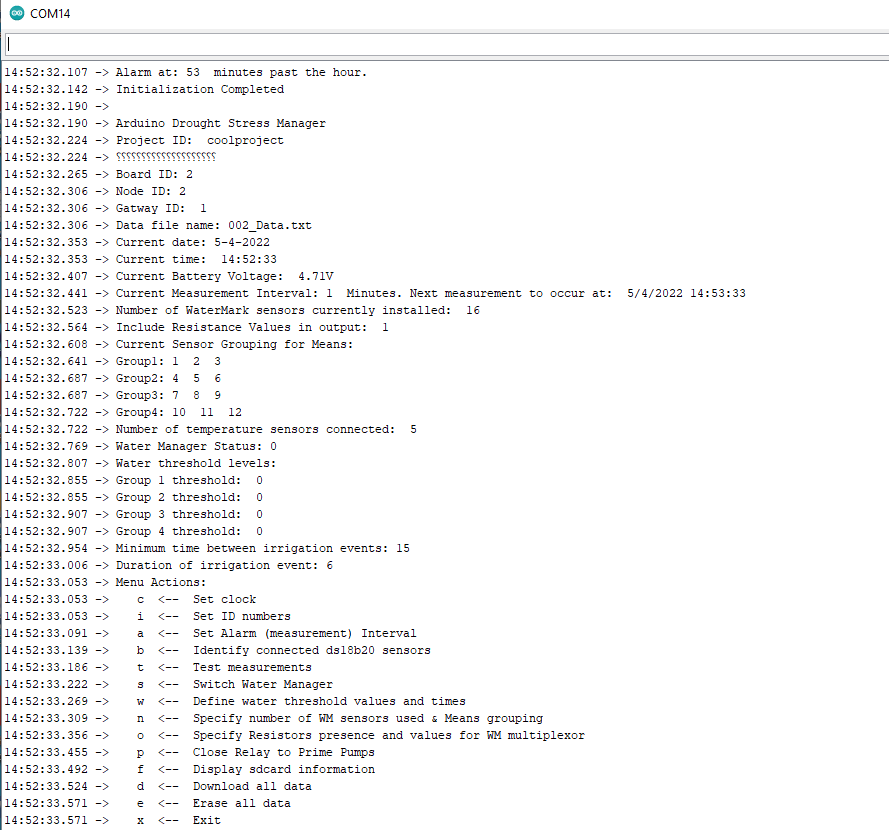
* The Open\_Irr system may operate on battery voltages from 3.3V to 16V, exceeding this will damage the unit.
* Do not operate the unit without an antenna connected – irreversible damage to the radio transceiver may occur.
* Do not bridge exposed pins, damage may occur.
* In the event of catastrophic component failure, component heating and smoke are possible. **IF witnessed, remove power from the device (unplug micro-usb from power source or disconnect battery pack)**
* Avoid electrostatic discharge to the unit.
* Do not attempt to interface higher-voltage systems with the automated irrigation routine if unfamiliar in AC / DC wiring.

Open\_Irr Node Operation

The Open\_Irr system is microcontroller based datalogger programmed with a derivative of the C++ language, Arduino, and can be programed through the Arduino Individual Development Environment (IDE). The Open\_Irr system was developed in 2021 by the laboratory of Dr. Andrew Bierer at the Appalachian Fruit Research Laboratory in Kearneysville, WV. Please refer to the laboratories Github <https://github.com/andrewbierer/Open\_Irr > to check for the latest Open\_Irr firmware for both the Node and Gateway.

To aid operation of the system, some tasks have been saved to the devices EEPROM (electrically erasable programable read-only memory). This allows creation of a crude “menu” system where the end user can change some program functions. To interface with Open\_Irr, plug a phone or laptop into the micro-usb port on the Moteino board. The phone or laptop used requires a serial port data monitoring program be installed. For PC, the Arduino IDE has a serial monitor you can use. For Iphone/Android, pick a reputable serial monitoring application.

When plugged into micro-usb port, the Open\_Irr system will be powered ON and begin displaying information in the serial monitoring program on the connected device, note that you have to select the serial port in the monitoring program. You will be greeted by the start-up screen which can be referred to as the Menu. **Note: while at the menu, the Open\_Irr system will wait for a user command for 10 seconds before exiting the menu automatically and beginning normal operation.** To return to the menu, simply close and re-open the serial monitoring window of the application in use. You may also un-plug/re-plug into the device or type “menu”, hit enter, and wait for the system to receive the command to return to the start-up menu.



Open\_Irr, Node start up menu

The Menu contains the commands listed below (at the time this guide was updated) which are accessed by typing a single command character and hitting enter. A synopsis of each menu option is given below.

c <-- Set clock

Guides user through setting the time in the Real Time Clock (RTC) module. Will sequentially ask for Month, Day, Year, Hour, Minute, and Second. Enter each of these as 2-digit numbers, i.e. for 2022 type “22”. The Hour must be specified using the 24-hour clock.

i <-- Set ID numbers

Guides user through specifying a project ID (name), the integer values attributed to the Node/Boards ID, and the Gateway to which it will communicate. The project ID may contain characters up to 32 characters in length. The node and gateway ID may be any integer. Typically, we reserve 001,002,003 etc. for connected gateways and start numbering Nodes at 101, 102, 103 etc.

a <-- Set Alarm (measurement) Interval

Allows user to define the sampling interval for the Open\_Irr system. System will wake and perform 1 cycle through its main loop before returning to sleep for the defined interval. Specified in Minutes.

b <-- Identify connected ds18b20 sensors

Guides user through iterative identification of connected DS18B20 temperature sensors. The sensors have unique “addresses” which are called when temperature readings are requested. Upon first set-up or adding/removing sensors, user should re-identify the connected sensors; start by unplugging ALL connected DS18B20 sensors. The system will ask you if you want to continue – if so type “1”, any other key will return to start-up menu. The system will ask how many DS18B20 sensors are being connected, enter an integer for this number (0-16). The user will then plug in individual sensors one-by-one and the system will save that sensor’s address to its eeprom memory. The order in which the sensors are plugged in is the “number” of the sensor in the program. When the user specified number of DS18B20 sensors has been identified the system will return to the start up menu.The use of the dallas 1-wire sensors in a “star” wiring topology comes with some downsides. Namely, the pull-up resistor connected to the data line may need changed (lowered) as the total length of DS18B20 wire increases as this adds to the resistance on the data line. If having connection issues, try replacing the pull-up resistor with a lower value resistor. Future iterations of Open\_Irr may utilize a potentiometer or terminal block to ease changing pull-up resistance or use a “bus” wiring topology to improve DS18B20 performance.

t <-- Test measurements

If selected, the system will run through 1 cycle of the main program loop except for saving or transmitting the data. Used primarily as a check on the values for connected sensors.

s <-- Switch Water Manager

If selected, will ask the user if they want to enable the water manager subroutine for sending a low-level (3.3V) signal to the output terminal block lines (1 to 4) based on sensor readings. If not connected to irrigation infrastructure (purely used as data recording device), keep water management routine OFF. Type “1” to enable the water management subroutine. The water manager routine works as follows. The system reads the soil matric potential from all connected WaterMark sensors and then calculates the mean for each group (1 to 4) of sensors that the user has defined in menu function “n”. The raw mean of a group is calculated first, then each individual sensor reading is compared to the group’s raw mean. If an individual sensor is ≥ 20% different () from the raw mean, it is flagged as a *possible* outlier. If an individual sensor is ≥ 20% different from the raw mean **AND** ≥ ± 10 kPa from the raw group mean **AND** ≥ ± 10 kPa from the user defined water threshold (menu option “w”), then that individual sensor is identified as an outlier and is dropped from the calculation of the “buffered” group mean that is saved to the data string and used in sending the low-level signal to the output terminal blocks (1 to 4). Note that no data is “lost” as raw readings are reported for each individual sensor and the raw mean can be recalculated. Reasons for anomalous readings include: poor sensor to substrate contact, preferential flow paths to or away from the sensor, unequal irrigation of the sensors within a group, variation in substrate a sensor is deployed in, and an aging sensor has lesser ability to compensate for elevated salinity in any given substrate. The “buffered mean” is compared against the user defined threshold values in menu option “w”. If the matric potential is lower (more negative, drier) than the threshold value for that group, the low-level signal is sent for the defined period in menu option “w”. The low-level signal may not be sent in the case that the minimum time between irrigation events defined in menu option “w” has not been exceeded. Future iterations of Open\_Irr may have other time exclusion options based on day and time of day.

w <-- Define water threshold values and times

Allows user to specify threshold matric potential levels for each group sequentially (1 to 4), and a common minimum time between irrigation events (in minutes). Future iterations of Open\_Irr may allow specification of time between irrigation events for each group and may also be set based on day and time of day. Then, the system will ask if you would like to include raw resistance values ( in Ohms) in the data string that is saved and transmitted – Type 1 to include or 0 to exclude the raw resistance values.

n <-- Specify number of WM sensors used & Means grouping

Allows user to specify number of connected Watermark sensors and instructions for grouping connected sensors – if you want to continue type ”1”, any other key will return to startup menu. Enter the number of installed Watermark sensors (0-16), the sensors must be installed in sequential order (i.e. sensor 1 is connected to 2-pin terminal block number 1). Typically, 2-pin terminal block 0 is reserved to hold a fixed resistor to calibrate the equation converting resistance to matric potential. The Open\_Irr system uses the calculation specified by the Watermark Manufacturer <https://www.irrometer.com/200ss.html> and does take into account temperature of the substrate (via DS18B20 sensors) in its calculation. The system will then ask the user for group averaging instructions for each group one-by-one (1 to 4) by first asking for the number of sensors to include in the group – enter “0” for no averaging routine; any number of sensors can be included in each group. After this is specified, the user must enter one-by-one the sensor numbers to include, up until the number of sensors in the group has been specified. The routine will return to the start up menu, be sure to check the grouping information was entered as intended.

o <-- Specify Resistors presence and values for WM multiplexor

Allows user to specify: first, whether a calibration resistor is included (1 = true | 0 = false); second, the calibration resistors position (0 to 15) among 2-pin terminal blocks; third, the value of the calibration resistor in Ohms (typically will be 10000 unless changed); and forth, the value of the FIXED pull-up resistor the analog reading is routed through before reading (should also be 10000).

p <-- Close Relay to Prime Pumps

For when external irrigation infrastructure is intended to be toggled by Open\_Irr through low-level (3.3V) output on the 4-pin terminal block. This allows the user to prime the lines by sequentially (1 to 4) sending low-level output to each pin of the 4-pin terminal block. Use to prime pumps or check irrigation infrastructure. In the past, we have toggled irrigation events using normally closed solenoids <https://www.adafruit.com/product/997?gclid=CjwKCAjw682TBhATEiwA9crl3yLJRAsjktwoK_7m6ltjDu48MROgKBlYWyEo5EGiTPZja9C7iAh95xoC5rIQAvD_BwE>.

f <-- Display sdcard information

This option will display information regarding the allocated and open space on the connected micro-sd card. Note at time of writing this function does not correctly indicate this information – this problem is known and listed on the “to-do” list.

d <-- Download all data

This option “prints” all data stored on the micro-sd card under the current working filename to the serial monitor of the connected phone or laptop. The routine will hang up for about 10 seconds to allow the user to copy and paste the data to another file etc., before returning to the startup menu. The filename will be XXX\_Data.txt where X’s are replaced with the Node/Board ID or the GatewayID in the case of the Gateway unit. **NOTE:** **This can be an extremely long process if there is a lot of data stored on the micro-sd card.** In effect, please consider the specified measurement interval and the length of time the unit has been ON before running the subroutine – it may be easier to turn the unit OFF, remove the micro-sd card and copy it to your PC through an external card reader.

e <-- Erase all data

Allows the user to erase the data on the micro-sd card saved under the current filename. User will have to type “YES” to confirm deletion of the file is desired. There is no option to recover deleted data, please save to an external source before deletion when in doubt.

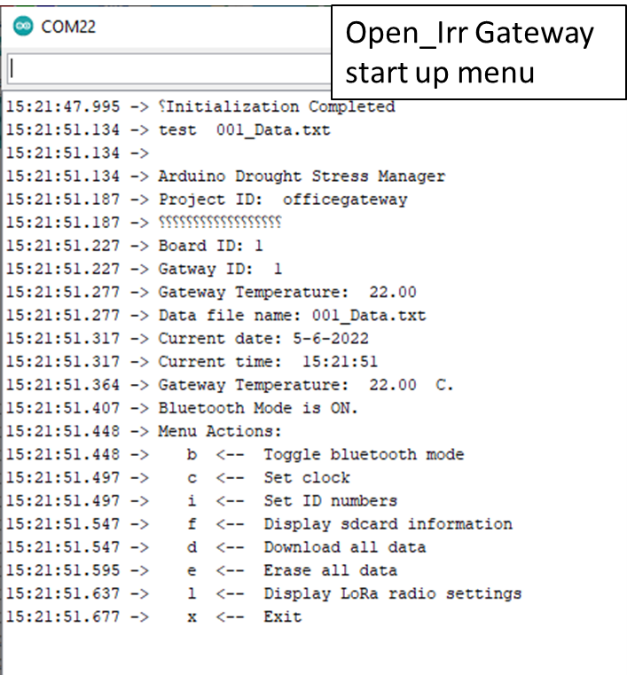
x <-- Exit

This option will exit the startup menu and proceed with the typical operating program. **Note: The startup menu will automatically “timeout” and exit from the startup menu if no actions are taken by the user in 10 seconds.**

Open\_Irr Gateway Operation

The Open\_Irr system is deployable as a single datalogging “Node” without need for radio transmissions, or as a network of one or more measurement “Node(s)” which transmit sensor data at 915 MHz to a connected “Gateway” for compilation. Specifying Node and Gateway ID’s is done through the startup menu. The Open\_Irr Gateway is constructed using the same components as Node units except that it lacks the connections for sensors. The advantage of using the Gateway is that multiple Node units can be connected to the same gateway – one can keep track of measurements made from multiple nodes spread out over a relatively large area from reading the micro-sd card at a single location. Future iterations of Open\_Irr may consider allowing the gateway itself to trigger irrigation events on received Node data as well as cellular, Wifi, and Bluetooth capability.

The Open\_Irr gateway is currently powered through a micro-usb connection of 3.3-16 DCV input, a battery pack containing 6 AA batteries is provided to bridge lapses in micro-usb power. At this time, the Gateway runs continuously and cannot fully rely on battery power for periods >24hrs. Future iterations of Open\_Irr will synchronize the Gateway with deployed Nodes so that a power-saving sleep routine can be incorporated in the Gateway to extend battery life.

When plugged into micro-usb port, the Open\_Irr Gateway will be powered ON and begin displaying information in a serial monitoring program on the connected device, note that you have to select the correct serial port to monitor. You will be greeted by the start-up screen which can be referred to as the Menu which is very similar to the Node menu. **Note: while at the menu, the Open\_Irr system will wait for a user command for 10 seconds before exiting the menu automatically and beginning normal operation.** To return to the menu, simply close and re-open the serial monitoring window of the application in use. You may also un-plug/re-plug into the device or type “menu”, hit enter, and wait for the system to receive the command to return to the start-up menu.

The Menu contains the commands listed below (at the time this guide was updated) which are accessed by typing a single command character and hitting enter. A synopsis of each menu option is given below.

b <-- Toggle bluetooth mode

This function is not functional at this time.

c <-- Set clock

Guides user through setting the time in the Real Time Clock (RTC) module. Will sequentially ask for Month, Day, Year, Hour, Minute, and Second. Enter each of these as 2-digit numbers, i.e. for 2022 type “22”. The Hour must be specified using the 24-hour clock.

i <-- Set ID numbers

Guides user through specifying a project ID (name), the integer values attributed to the Boards ID, and the Gateway ID to which Nodes will need addressed to. The project ID may contain characters up to 32 characters in length. The gateway ID may be any integer. Typically, we reserve 001,002,003 etc. for connected gateways and start numbering Nodes at 101, 102, 103 etc.

f <-- Display sdcard information

This option will display information regarding the allocated and open space on the connected micro-sd card. Note at time of writing this function does not correctly indicate this information – this problem is known and listed on the “to-do” list.

d <-- Download all data

This option “prints” all data stored on the micro-sd card under the current working filename to the serial monitor of the connected phone or laptop. The routine will hang up for about 10 seconds to allow the user to copy and paste the data to another file etc., before returning to the startup menu. The filename will be XXX\_Data.txt where X’s are replaced with the Node/Board ID or the GatewayID in the case of the Gateway unit. **NOTE:** **This can be an extremely long process if there is a lot of data stored on the micro-sd card.** In effect, please consider the specified measurement interval and the length of time the unit has been ON before running the subroutine – it may be easier to turn the unit OFF, remove the micro-sd card and copy it to your PC through an external card reader.

e <-- Erase all data

Allows the user to erase the data on the micro-sd card saved under the current filename. User will have to type “YES” to confirm deletion of the file is desired. There is no option to recover deleted data, please save to an external source before deletion when in doubt.

l <-- Display LoRa radio settings

This prints the current radio details to the serial monitor, this is printed in register numbers which need additional interpretation.

x <-- Exit

This option will exit the startup menu and proceed with the typical operating program. **Note: The startup menu will automatically “timeout” and exit from the startup menu if no actions are taken by the user in 10 seconds.**

Open\_Irr Maintanence

* It is recommened to use and regularly change dessicant packs inside the Open\_Irr environmental enclosure.
* Regularly check and replace depleted battery packs.
* After a long deployment and while disconnected from any power source, wipe down terminal blocks and environmental enclosure.
* The RTC breakout module is powered by a CR1220 coin cell battery to maintain datetime while not externally powered from usb or battery pack, it may need changed if recurring loss of datetime is encountered.

Open\_Irr Troubleshooting

* To find the Arduino IDE follow this link <https://www.arduino.cc/en/software>
* To install drivers for the Moteino microcontroller used in Open\_Irr follow the guide here <https://lowpowerlab.com/guide/moteino/programming-libraries/> . You will also need to find the patched version of the Radiohead library here <https://lowpowerlab.com/guide/moteino/lora-support/> . Also remember supporting documentation can be found at the Open\_Irr Github page <https://github.com/andrewbierer/Open_Irr> .
* For contributing to Open\_Irr, please submit pull requests and upload a merge request with a description of the change or improvement to the projects Github page.
* A handful of R scripts for visualizing datafiles from Open\_Irr are available on the projects Github page.