Low Power LoRa Sensor User Manual

By: Jim Schrempp and Bob Glicksman; v1, 7/14/2025

NOTICE: Use of this document is subject to the terms of use described in the document "Terms_of_Use_License_and_Disclaimer" that is included in this release package. This document can be found at:

https://github.com/TeamPracticalProjects/LoRa_experiments/tree/main/Documents/Terms_of_Use_License_and_Disclaimer.pdf

TABLE OF CONTENTS.

OVERVIEW.

This document describes a battery operated, low power LoRa sensor that is the key component of a LoRa sensor logging and notification system. The sensor uses LoRa[1] technology to send short messages over substantial distances using very low power. Note that the sensor uses LoRa in a point-point network. It does not use mesh networking technology such as LoRa WAN or Meshtastic.

LoRa communication technology is provided by a Reyax RYLR998 LoRa transceiver module. The RYLR998 data sheet is included in this folder (file: RLYR998_EN.pdf). This low cost module contains a Semtech LoRa engine and a low power microcontroller. The module uses serial asynchronous I/O to communicate with a host computer or microcontroller. Serial communication is based upon "AT" commands. These commands are documented in the LoRa AT Command Guide that is also included in this folder (file: LoRa_AT_Command_RYLR998_RYLR498_EN.pdf). The RLYR998 module can be powered down to draw less than 10 microamps of current when it is not being used to transmit or receive LoRa messages.

An ATMega328P microcontroller forms the "brains" of the sensor. This microcontroller was chosen for two reasons:

1. It can be programmed using the Arduino IDE, with a programmer made from an Arduino Uno (R3) board.
2. It can be powered down to draw very low current (< 0.35 microamps) and can be "woken up" using an external interrupt.

The sensor is triggered via a contact closing or opening. A readily available magnetic reed switch window/door sensor can trigger the sensor to wake up and send out a LoRa message whenever the contact is either closed or opened (the polarity is set via an on-board jumper). This makes the sensor useful for many projects, such as:

- Informing a user whenever a mailbox is opened.
- Informing a user when a gate or door to an out-building is opened.
- Triggering an announcement whenever a battery-powered "help button" is pressed.

The sensor uses an on-board CR123A three volt Lithium battery, albeit a terminal block is provided on the board to connect an external three volt power source to the sensor in the event that the on-board battery is not desired. This battery is inexpensive and features:

- Small size; high power density.
- Long shelf life (> 10 years).
- Flat discharge curve over most of the battery life.

[1] https://en.wikipedia.org/wiki/LoRa

- Wide ambient temperature range; suitable for outdoor deployment.

Sensors communicate over LoRa with a central Hub. The central Hub is a Particle[2] Photon 2 microprocessor. The Hub decodes received messages from sensors, acknowledges each sensor message, and publishes an event to the Particle[3] cloud for the purposes of logging, notification, and triggering of subscriber devices. Multiple sensors may be used in such a system.

The sensor board has three address jumpers that together form a binary offset (0 – 7) to a base sensor address that is defined in the microcontroller software. These jumpers allow multiple sensors to be configured into one network without having to change the software. In addition, several sensors can be configured with the same address so that they each send the same message to the Hub.

THEORY OF OPERATION.

This document relates to the “Rev B” low power sensor printed circuit board. Schematic, board layout and Eagle CAD files are included in this repository in the folder:

Hardware/LoRa Sensor Development/PCB/Rev B Board

A pdf version of the schematic is contained in the file:

LoRa_Sensor_Rev_B_schematic.pdf

[2] Particle.io
[3] Particle.io

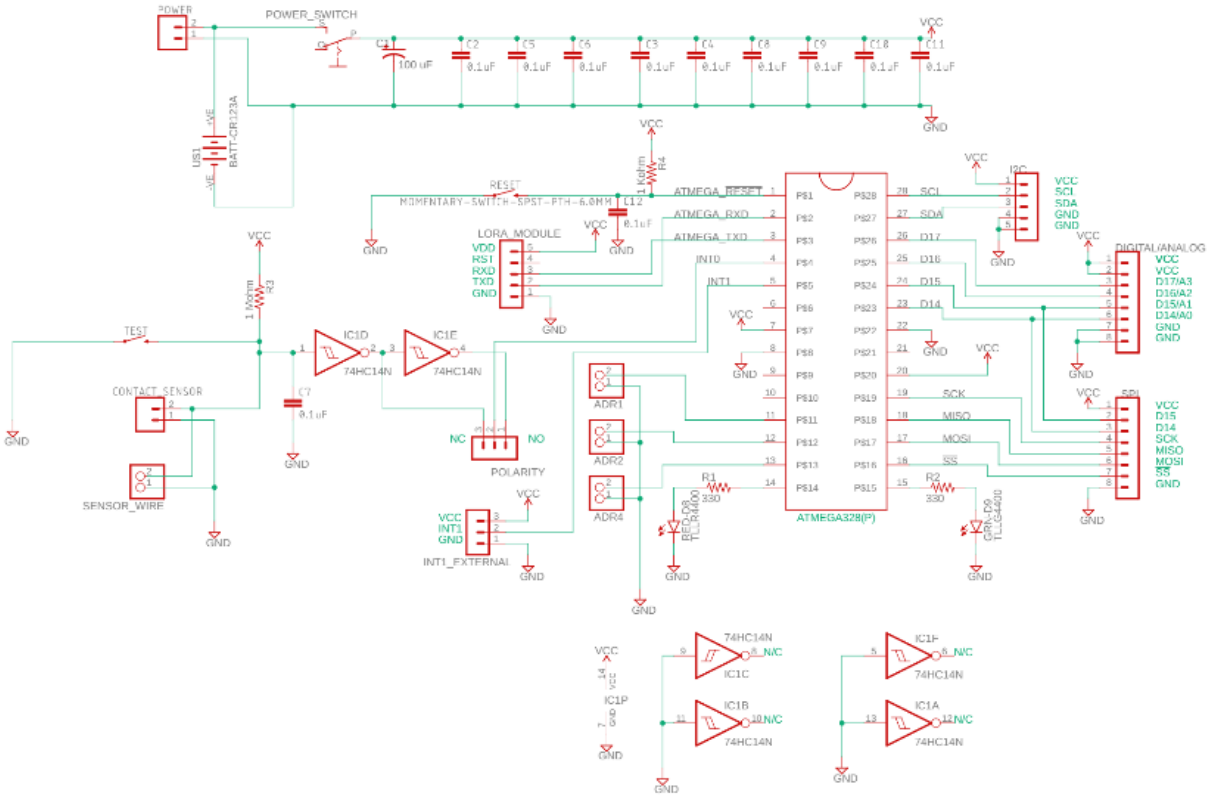

Figure 1. Low Power Sensor Schematic.

Sensor trigger circuit.

Closing and opening of a contact is sensed via a 1 megohm pull-up resistor (R3). This resistor pulls up the input of a Schmitt trigger inverter (IC1D) to Vcc. Closing the contact grounds this line. The large value of the pull-up resistor is necessary in order to reduce power consumption when the contact is closed. This large value of resistor is sensitive to ambient noise; hence a 0.1 uF capacitor (C7) is used to filter out any noise that would otherwise trigger the sensor. The slow rise time imposed by this R-C combination is mitigated using the Schmitt trigger input of the inverter.

The output of the inverter IC1D is connected to the input of a second inverter (IC1E). The outputs of IC1D and IC1E are connected to the opposite ends of the POLARITY jumper so that either a normally open (NO) or normally closed (NC) contact may be selected to interrupt the microcontroller on external interrupt `INT0` (microcontroller pin 4). A falling edge on `INT0` wakes the microcontroller up from deep sleep.

Microcontroller.

An ATMega329P microcontroller forms the “brains” of the sensor. After a reset, the microcontroller powers down the RLYX998 LoRa module and then powers itself into the lowest power state possible. The microcontroller is programmed to wake-up on a falling edge interrupt on `INT0`. When interrupted, the microcontroller:

1. Wakes itself up.
2. Wakes up the LoRa module.
3. Uses the LoRa module to transmit a “sensor tripped” message.
4. Places the LoRa module into receive mode and waits (with a 5 second timeout) for a response to its message from the Hub.
5. Powers down the LoRa module.
6. Powers itself down.

NOTE: The ATMega328P microcontroller must be programmed using an external programmer. Therefore, it must be socketed on the sensor printed circuit board. The printed circuit board can accommodate a 28 pin ZIF socket (e.g. https://www.adafruit.com/product/382) and this socket should be used if the sensor board is being used for development and testing. However*, a ZIF socket should not be used on a deployed sensor board*, since the contacts are not gas tight and may fail due to changes in the ambient temperature. An ordinary, high quality 28 pin DIP socket should be used for deployed sensor boards.

LoRa Module.

The RLYX998 LoRa module is connected to the printed circuit board using a 5 pin, right angle, female pin header. The header is positioned on the printed circuit board so that the antenna of the LoRa module sticks out into the open air and away from signal and ground traces on the circuit board.

The RLYX998 LoRa module communicates with the ATMega328P microcontroller using the microcontroller’s on-board hardware serial port (Rx and Tx lines on microcontroller pins 2 and 3 respectively). The baud rate is set in the microcontroller software to 38,400 baud. *NOTE: this is not the default baud rate of the RYLX998. The default baud rate from the factory is usually 115,200 baud. The default board rate must be changed to 38,400 prior to using the LoRa module in the sensor. See the Software Installation and Setup instructions, below, for details.*

The ATMega328P microcontroller communicates with the RLYX998 over the serial port using “AT” commands. The RLYX998 AT command set is documented in the pdf document in this folder:

LoRa_AT_Command_RYLR998_RYLR498_EN.pdf

Connections and Status.

Two LEDs (RED and GREEN) are connected to the ATMega328P microcontroller via 330 ohm current limiting resistors (microcontroller pins 14 and 15, respectively). These LEDs are used by the software to provide operational status for debugging, testing and monitoring purposes.

An on-board battery holder is provided for the CR123A battery. A two pin terminal block (POWER) can be used to power the sensor board externally; the battery must be removed in this instance. External power between 3.0 and 3.3 volts should be used. A power on-off switch is provided to disconnect internal and external power from the rest of the circuitry, e.g. to shut off power when removing/inserting the microcontroller and/or the LoRa module.

An external dry contact sensor can be connected to the circuit board using 0.1" female Dupont pin connectors to the 90 degree male pin header on the circuit board (SENSOR_WIRE). This connector should be used for all sensor deployments as the connections are gas tight and resistant to swings in ambient temperature. A two position terminal block may alternately be used to connect a contact sensor to the circuit board (CONTACT_SENSOR). This terminal block may be more convenient for experimentation but should not be used for deployment as the contacts are prone to failure under ambient temperature changes.

Jumpers.

The POLARITY jumper position determines whether the sensor is triggered by closing a normally-open contact or by opening a normally-closed contact. *A jumper MUST be placed in one of the two positions or else the sensor will not trigger!*

Jumpers ADR4, ADR2 and ADR1 form a 3 bit offset (values 0 – 7) to the base sensor device ID in the microcontroller software. An open jumper reads as a logic 1 and an in-place jumper reads as a logic 0. Therefore, the sensor device ID is the base ID in the software plus an OFFSET, where the offset is zero with all jumpers inserted and 7 with all jumpers removed.

NOTE: the sensor address jumpers are read during setup() only. *The sensor must be reset (either with the RESET pushbutton or cycling the power) after the address jumpers are changed in order to change the sensor's address*.

Switches and Buttons.

The POWER_SWITCH slide switch connects the battery and/or external power source to the rest of the circuitry on the sensor printed circuit board.

The RESET pushbutton switch causes the ATMega329P microcontroller to reset and start running its `setup()` function before entering `loop()` and going into deep sleep.

The TEST pushbutton is wired across the sensor inputs and may be used to trigger the sensor in the absence of a contact sensor connected to either the SENSOR_WIRE connector or the CONTACT_SENSOR terminals.

HARDWARE BUILD INSTRUCTIONS.

Building the Printed Circuit Board.

Figure 2, below, shows the major components on a fully assembled sensor printed circuit board:

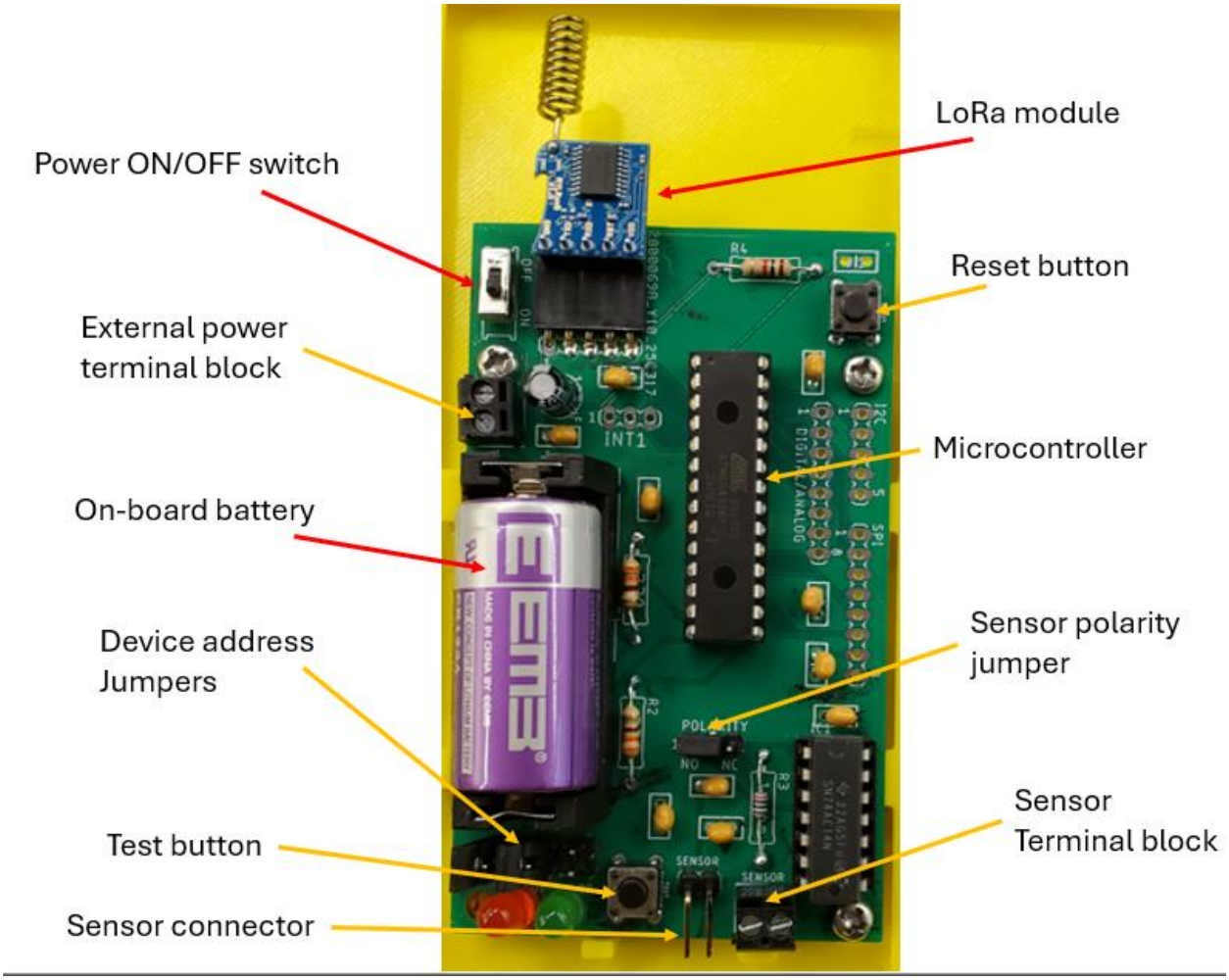

Figure 2. Assembled Sensor Printed Circuit Board.

A parts list is included in this repository:

Hardware/LoRa Sensor Development PCB/Rev B Board/LoRa_Sensor_RevB_parts_list.pdf

Cadsoft Eagle files are provided in this repository. The manufacturing files in:

LoRa_Sensor_RevB_2025-03-15.zip

have been used to order boards from JLCBCB.com, but should work with most online PCB houses.

It is strongly recommended to include all components shown in figure 2 when constructing a sensor PCB. Note that a sensor PCB that is used for software development should use a ZIF socket (e.g. https://www.adafruit.com/product/382) so that the ATMega328P microcontroller can be easily inserted into and removed from the circuit board for re-programming and testing of new/revised code.

Components not included on the assembled printed circuit board.

Figure 3 shows components that are not connected for a normal, deployed sensor board:

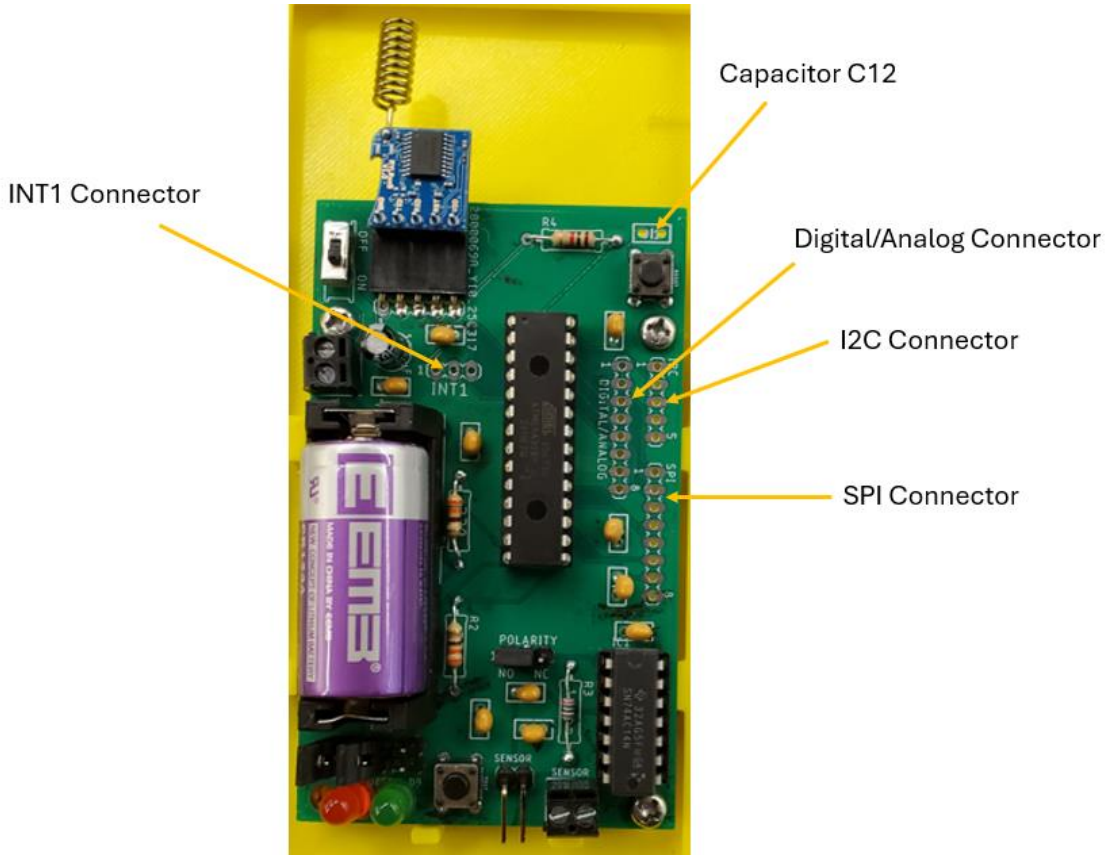

Figure 3. Components that are not normally connected.

- Capacitor C12: A provision has been made to connect a filter capacitor between the microcontroller reset pin and ground, in the event that the sensor is deployed in a noisy environment extreme enough to trigger the microcontroller reset even with the strong 1K ohm pullup (R4). We have not found a need to include a filter capacitor in any of our deployments and we do not recommend soldering a capacitor into this position unless testing deems it necessary.
- INT1 connector field: The ATMega328P microcontroller provides 2 external interrupts – INT0 and INT1. INT0 is used to wake up the microcontroller on a sensor trip. INT1 is not currently used. This connector field is provided for future development and experimentation.
- Analog/Digital connector field: These are spare analog and digital pins on the ATMega328P microcontroller. They are not used in the current design but are provided for future development and experimentation.
- I2C connector field: There are spare IC2 bus/digital I/O pins on the ATMega328P microcontroller. They are not used in the current design but are provided for future development and experimentation.
- SPI connector field: There are spare SPI bus/digital I/O pins on the ATMega328P microcontroller. They are not used in the current design but are provided for future development and experimentation.

Optional Components on the Sensor PCB.

We STRONGLY recommend including all components on the sensor printed circuit board that are shown in figure 2. However, some components may be left off of the board if they are not needed for a specific deployment scenario:

- Battery holder/battery: If an external power source will be used to power the sensor board, then the battery holder and battery may be left off of the board. The POWER terminal block must be soldered on the board and used to supply power externally to the sensor board.
- External Power Terminal Block: If the on-board battery will always be used to power the sensor, then the POWER terminal block need not be soldered onto the printed circuit board.
- SENSOR connector and SENSOR terminal block: These components are wired in parallel and only one is used to connect an external contact sensor to the sensor circuit board. Either one may be left off of the board if it is not intended to be used. We STRONLY recommend using the 0.1” Dupont male terminals to connect an external sensor to the board, since these provide a gas-tight connection that is resistant to changes in ambient temperature. The terminal block should only be used for experimentation and not for deployment. It is provided for experimental purposes since it is easier to connect devices to for bench testing.

- RESET pushbutton: We STRONLY suggest including this button on the assembled printed circuit board in order to easily reset the microcontroller. However, cycling the power will also reset the microcontroller.
- TEST pushbutton: This pushbutton should be included to make it easy to test the sensor – no external contact is needed. It can be omitted from the circuit board if the board is to be deployed in a location where access to the TEST button is difficult. Note that the TEST button is useful for bench testing after building the sensor and/or when replacing the battery or making changes to the sensor code.
- RED and GREEN LEDs: These LEDs and their associated 330 ohm current limit resistors are provided for testing and debugging the microcontroller code. These components may be omitted from the circuit board if the LEDs are not visible when the sensor is deployed. Note, however, these LEDs are still useful on the bench, i.e. after replacing the battery.

SOFTWARE INSTALLATION AND SETUP INSTRUCTIONS.

After the sensor printed circuit board is assembled, two operations need to be completed in order to have a working sensor:

1. The microcontroller software must be flashed to the microcontroller chip.
2. The RYLX998 LoRa module must have its baud rate and LoRa parameters set properly for deployment.

Programming the Microcontroller.

The ATMega328P microcontroller must be removed from the sensor printed circuit board and placed into a dedicated programmer in order to flash software onto it. A dedicated programmer can be constructed from an Arduino Uno (R3), a solderless breadboard, a ZIF socket, and some jumper wires. Complete instructions for building and operating this programmer are contained in the following document in this folder:

Arduino Uno as ICSP Programmer for Bare Bones ATmega328.pdf

Please be sure to follow the instructions in this document carefully and completely in order to successfully flash code to the microcontroller. This programmer uses the Arduino IDE to compile and flash code to the bare-bones ATMega328P chip.

The actual code that you will flash to the ATMega328P chip is contained in the following folder in this repository:

/ arduinoSketchbook / RangeTestSensor / RangeTestSensor.ino

After connecting the programmer to your computer and installing the Arduino IDE to your computer (from the Arduino.cc website), open up the RangeTestSensor.ino file from this location. *NOTE: there are included library files in this RangeTestSensor folder. All of the files are necessary for the code to compile.*

Once the code is compiled, place the ATMega328P chip into the ZIF socket of the programmer and follow the instructions in the programmer document to flash the compiled code onto the chip.

NOTES:

- It is necessary to install a hardware description file for the bare bones ATMega328P into the Arduino IDE, as detailed in the programmer instruction document. The IDE must be closed and re-opened after installing this file.
- It is necessary to burn the bootloader to a brand new ATMega328P chip in order to set the fuses properly. The bootloader will subsequently be overwritten as it is not needed to flash code to the chip using the documented procedure. This step may be skipped when re-flashing code to a previously-programmed chip.
- The programmer document describes a small modification to the hardware description file if you want to program an ATMega328 (non-P version). These chips will work, albeit they power down to a slightly higher quiescent power draw than do the P version of these chips.

Once the code is successfully flashed to the chip (as indicated by the status at the bottom of the Arduino IDE), the chip may be removed from the programmer and inserted into the socket on the sensor printed circuit board.

Configuring the LoRa Module.

The RYLR998 LoRa module must be configured properly in order to be used in the sensor. There are two aspects to this configuration:

1. The baud rate for the module must be set to 38,400.
2. The many LoRa parameters must be set to the exact same values as the module used on the Hub to which the sensor will communicate.

Setting the Baud Rate on the LoRa module.

The RYLR998 LoRa module usually comes factory pre-set to a baud rate of 115,200. The baud rate must be changed to 38,400 in order for the sensor microcontroller to communicate with the

module. The best way to set the baud rate is to use an “FTDI” module that plugs into a computer’s USB port and provides 3.3 volts, ground, Tx and Rx signals. A module that works well can be found at:

https://www.amazon.com/dp/B00LODGRV8?ref_=ppx_hzsearch_conn_dt_b_fed_asin_title_1

In order to use this module to set the LoRa board rate, jumper wires are used to connect:

- FTDI +3.3 volts to LoRa module Vcc
- FTDI ground to LoRa module ground.
- FTDI Tx to LoRa Rx
- FTDI Rx to LoRa Tx

Next, plug the FTDI module into a USB port on your computer and note the COM port/TTY port where it is installed.

Open up any serial monitor program on the computer. If you are using the Arduino IDE on this computer, you can use its serial monitor.

Set the serial monitor to the COM/TTY port of the FTDI module. Set the serial monitor baud rate to 115,200, and set the end of line to both CR and LF. You should be able to type “AT <enter>” on the serial monitor and have the LoRa module respond with “OK”.

Now, type the following on the serial monitor: “AT+IPR=38400<enter>”. It should respond with OK. Now, change the baud rate on the serial monitor to 38,400 and type “AT<enter>”. It should respond with “OK”. The baud rate on the RYLR998 LoRa module is now set to the proper 38,400 baud.

Setting the LoRa Parameters on the RYLR998 Module.

The LoRa parameters may be set individually using the FTDI module / serial monitor procedure described above for setting the baud rate. There are many parameters to set and these parameters must exactly match the LoRa module setting on the Hub in order for the sensor and the Hub to communicate. The commands for setting the LoRa parameters are documented in the following manual in this folder:

LoRa_AT_Command_RYLR998_RYLR498_EN.pdf

The Hub software has been designed to set the desired LoRa parameters into the RLYR998 module during `setup()`. These parameters are stored in flash memory on the LoRa module, so once they have been set, the module can be moved and the parameters will stay valid. The best way to ensure that the sensor LoRa module and the Hub LoRa module have the same LoRa parameters is to plug the sensor LoRa module into the Hub and power up the Hub. After

the Hub code runs `setup()`, the LoRa parameters used in the sensor/Hub system will have been flashed, and the LoRa module may be removed from the Hub and placed into its socket on the sensor.

Battery Life.

Battery life is dependent on environmental conditions, quiescent power draw of the sensor, and the number of times that the sensor is tripped. Battery life for a fresh CR123A battery has been modeled on a spreadsheet and is optimistically estimated at 4.5 years. This model assumes 3 sensor trips per day. The model does not take into account the self-discharge of the battery nor the impact of ambient temperature on the battery, so the actual life of the battery will probably be somewhat shorter. Basic measurements have been taken on a real sensor board:

- Quiescent current draw: 9 microamps (sensor open); 12 microamps (sensor closed).
- Current draw when transmitting a message: > 150 milliamps.
- Current draw when receiving a message: < 35 milliamps.

It is recommended that the battery be pro-actively replaced once per year to ensure that the sensor will continue working at its designed performance level.