Low Power LoRa Sensor User Manual

By: Jim Schrempp and Bob Glicksman; v1, 7/25/2025

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 provides an overview of the contents of this repository:

TeamPracticalProjects/LoRa_experiments

The LoRa experiments were undertaken to evaluate low power, low cost LoRa technology for possible applications involving battery powered sensors that need to communicate short messages over substantial distances. Applications of this technology include:

- Sensing when a gate, mailbox, or outbuilding door is opened.
- Sending a message whenever a battery powered “help button” is pressed.

These experiments have resulted in the creation of hardware and software that has been deployed in a number of such situations. This re-usable hardware and software is fully documented in this repository. In addition, non-reusable and developmental software is memorialized here, as are the results of LoRa range tests performed in various urban and suburban settings.

TECHNOLOGY.

Technology Overview.

The LoRa experiments were conducted using Reyax RYLR998 LoRa modules. These modules are low cost (approximately $12 each) and contain a LoRa transceiver and a low power microcontroller.

Figure 1. Reyax RYLR998 LoRa Module.

A host computer can communicate with the RYLR998 using serial I/O “AT” commands. The commands are documented in this repository:

> TeamPracticalProjects/LoRa_experiments/Documents/
> LoRa_AT_Command_RYLR998_RYLR498_EN.pdf

The RYLR998 communicates using LoRa[1] technology. Communication that was investigated here was point to point messaging between a battery powered Sensor and a centrally located, Internet enabled Hub. Mesh network overlays to LoRa point to point, such as LoRaWan and Mestastic, were not investigated here.

Hardware.

Two versions of Sensors were developed and deployed as part of this program. A battery operated Sensor using an ATMega328P microcontroller is powered down to an extremely low ambient current draw[2] until a contact is opened or closed. Contact closure or opening wakes up the microcontroller, which in turn powers up the LoRa transceiver and sends out a message to the Hub. The Sensor waits for receipt of an acknowledgement from the Hub and then powers itself back down. The Low Power Sensor is further documented in this repository:

> TeamPracticalProjects/LoRa_experiments/Documents/Low_Power_LoRa_Sensor_User_Manual.pdf

The second type of Sensor that was developed is a non-low power version of the same sensor based upon a Particle Photon 2 microcontroller. The hardware for this latter Sensor doubles as the hardware for the Hub, and is documented in this repository:

> TeamPracticalProjects/LoRa_experiments/Documents/Photon_2_Hub_and_Sensor_User_Manual.pdf

The Photon 2 based Sensor is not low power and does not contain an on-board battery. It must be powered via the on-board USB connector. The Photon 2 based Sensor software is very similar to the low power sensor software; however it has additional logging of messages and events via the on-board USB port. We used this additional logging capability as part of our experiments in LoRa communication range under various conditions of orientation and of LoRa configuration parameters.

The Photon 2 based Sensor hardware also serves as the Hardware platform for the Hub. The same hardware may be used for both, but the Photon 2 software is different for the Hub and for the Sensor. See the User Manual for details.

---

[1] https://en.wikipedia.org/wiki/LoRa
[2] Typically 9 – 12 microamperes from a 3 volt power source

The Sensor and the Hub hardware are contained on printed circuit boards that have been designed with deployment in mind. Various 3D printed enclosures have also been developed. Source files for PCBs in Eagle and 3D CAD files in Fusion 360 are included in the ***Hardware*** folder in this repository.

Software.

This project includes deployable software for the Sensors and the Hub. It also includes other software related items such as Particle Webhooks, Google Apps Scripts, an MIT App Inventor 2 app, and a Google spreadsheet for logging. The deployable Sensor and Hub software has been designed to provide sufficient LoRa communication for experimentation and development, but is also robust enough for deployment on projects. See the User Manuals and the section on ***Contents of this Repository*** in this document for further information.

This repository also memorializes some very simple microcontroller software that was used for basic hardware capability testing of LoRa communication and for low power operation of an ATMega328P microcontroller. We have preserved this software for future development that might want to follow a similar development path but with different LoRa or microcontroller hardware. See the section on ***Contents of this Repository*** in this document for further information.

System Overview.

Figure 2 contains an overview of the system that we used for experimentation, as well as the basis for deployed projects.

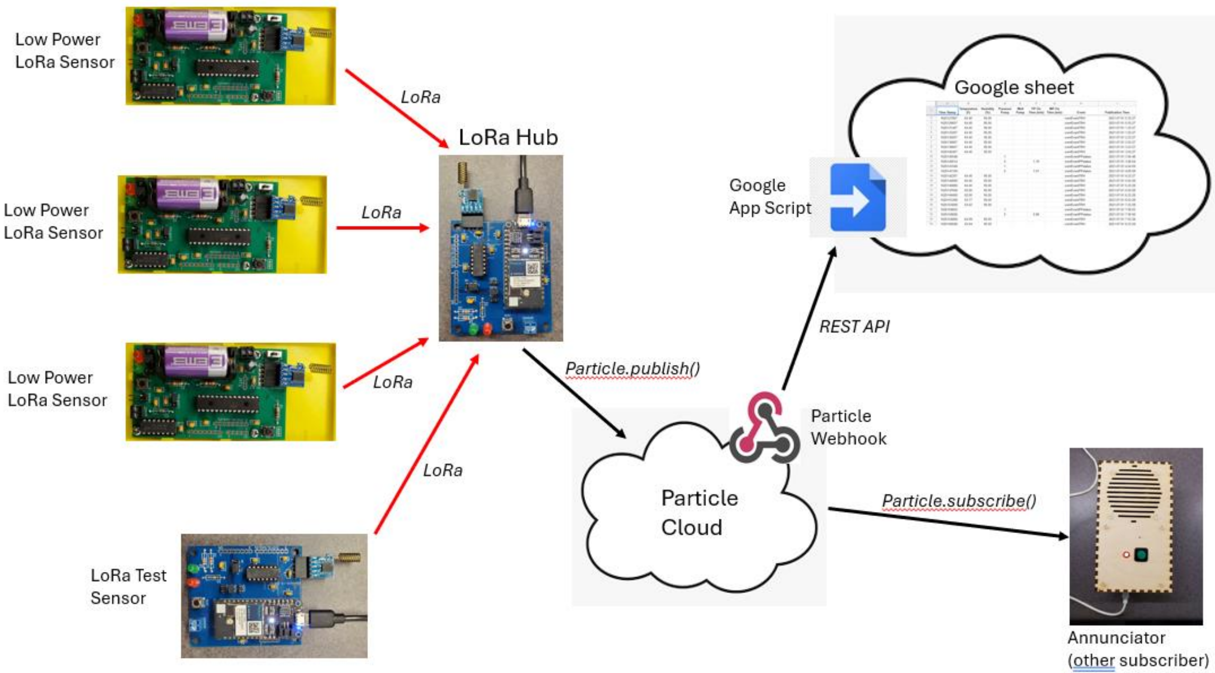

Figure 2. LoRa Experiments System Overview.

A mixture of Low Power Sensors and Photon 2 based Test Sensors can all be part of one system. Both types of Sensors have address jumpers that can be set to create up to 8 different device IDs for each Sensor. Multiple Sensors with the same device ID are allowed. If more than 8 distinct Sensor device IDs are needed in a system, the base address of a set of 8 IDs can be changed in the Sensor software. A Sensor must be reset after changing device ID jumpers in order for the new Sensor device ID to take effect.

All Sensors contain low power circuitry to detect a change (opening or closing) of a dry contact. The polarity of the action (opening or closing) that triggers the sensor to send out a LoRa message can be set via an on-board jumper.

Each Sensor sends out a LoRa message when it is triggered. The LoRa messages that are part of the deployable software are a compromise between brevity (needed for low power operation) and the need to obtain and log sufficient information for experimental purposes. The LoRa messages sent out from each Sensor contain:

- The Sensor's device ID
- A very brief code indicating the Sensor type.
- A one-up message count number that starts with zero (after a reset) and increments thereafter.
- RSSI and SNR readings for the previous response message from the Hub.

There is only one Hub for any given system. The Hub operates on LoRa Channel 18, and all Sensors must communicate on this same channel (and with the same LoRa settings) in order to communicate with the Hub. The Hub is configured (in the Hub software) with a specific LoRa device ID. The Hub device ID must be different from each Sensor's device ID and must be unique for any given system.

When the Hub receives a LoRa message, the Hub software decodes the message, checks the message for validity, responds to the sending device with a response message, and publishes an event to the Particle Cloud. The published event contains information about the received message, including:

- The device ID of the sensor sending of the message.
- The message type and message count information from the received message.
- The Hub's recorded RSSI and SNR values for the received message.

In addition, the Hub contains a Particle Cloud function that can be called (e.g. from the Particle Console) to simulate receipt of a message from a Sensor without the need for an actual LoRa Sensor to have sent a message to the Hub. This capability is useful for testing, as well for activation of simulated Sensors from an App. MIT App Inventor 2 code for such an App is included in this repository.

A Particle Webhook is configured to subscribe to received sensor message events. This webhook does an https: POST to a Google apps Script that is bound to a Google sheet. A configurable Google sheet is used to log messages received by the Hub.

Additional subscribers to the Hub-published events can be used to make the system more flexible. Figure 2 shows the addition of an "Annunciator". The Annunciator is not part of this repository, as it does not use LoRa communication in any way. It is a general purpose device that can subscribe to Particle event publications and can play an appropriate MP3 clip associated with the published event data. The Annunciator is documented in its own GitHub repository:

TeamPracticalProjects/Annunciator

SUMMARY OF LORA EXPERIMENTS RESULTS.

Tests of the Reyax RYLR998 LoRa modules have been well documented on the Internet, specifically on YouTube. Most of these tests were performed at significant elevation of the communicating transceivers and over open air, unobstructed by buildings, terrain or vegetation. These tests demonstrate considerable communication range (several miles) under these conditions. However, our interest was in using this technology under some real-world conditions, including:

- Communication over some open-air distance but also through some walls, specifically walls of wood frame structures.

- Communication where the Sensors and Hub are close to the ground; e.g. table height.
- Communication where direct line-of-sight between Sensor and Hub is lost but where LoRa signals may still make it from Sensor to Hub.

We were also interested in the effects of antenna orientation and of varying LoRa parameters such as signal rate, signal coding, bandwidth and spreading factors.

We performed our experiments mostly in urban/suburban residential environments and in one light industrial environment. All of our tests used the Reyax RYLR998 LoRa modules as is; i.e. using the small coil antenna that is soldered on the module. A summary of our findings follows.

Range testing.

Tests performed in a suburban residential neighborhood, with Sensor and Hub at table height level showed:

- Successful communication over 700 feet of open air and through about 50 feet of wood frame building interior walls.
- Successful communication over 1000 feet of open air and through one exterior wall of a wood frame building.
- Successful communication over approximately one quarter mile of open air and through an earthen berm, an exterior concrete wall, and interior glass/metal frame wall and about 15 feet of interior space in a quiet industrial park setting.
- Successful communication between a Sensor and Hub through an open alleyway but where direct line of sight between Sensor and Hub was blocked by light industrial buildings.
- Partial (3 successes out of 10 tries) successful communication out ¼ mile “as the crow flies” between Sensor and Hub, but direct line of sight blocked by light industrial buildings, but opportunity for signals to be reflected off of the buildings’ exteriors.
- Successful communication of over ¼ mile in a suburban residential neighborhood with near line of sight between Hub and Sensor but communication lost when cars passed between the Hub and Sensor.

These range tests verified that LoRa technology using the Reyax RYLR998 LoRa modules was very useful in several applications that we had in mind.

Height and Antenna Orientation Testing.

Most of our tests were performed with the LoRa modules at “table top” height off of the ground. At this height, LoRa communication range can be compromised by the “Fresnel effect” whereby reflection of the transmitted signal off of the ground interferes with the line of sight signal between Sensor and Hub and thereby limits the range. We performed an additional set of tests where the Hub was placed near a 2nd story window but the Sensor was at “table top” height. The range was a little bit longer than where the Hub was at ground level; however, the range extension was not significant. We concluded that the “Fresnel effect” was still limiting the range because the Sensor (transmitter) was still near the ground.

We also performed tests with various combinations of antenna position:

- Both Sensor and Hub antennas were vertical, with respect to the ground.
- Both Sensor and Hub antennas were horizontal, with respect to the ground.
- The Hub antenna was vertical with respect to the ground but the Sensor antenna was horizontal with respect to the ground.

We found that the best range was achieved where both Sensor and Hub antennas were vertical, with respect to the ground. The range was slightly degraded where both Sensor and Hub antennas were horizontal, with respect to the ground. The range was further degraded when the Hub antenna was vertical with respect to the ground but the Sensor antenna was horizontal with respect to the ground. All in all, however, the range loss in the latter two cases was small compared to the overall range achieved in the testing.

LoRa Parameter Testing.

The Reyax RYLR998 LoRa modules come with a default set of LoRa parameters. These are documented in the Reyax document in this repository:

TeamPracticalProjects/LoRa_experiments/Documents/
LoRa_AT_Command_RYLR998_RYLR498_EN.pdf

- Band: 915 MHz (for the US)
- Spreading Factor: 9 (range 5 – 12)
- Bandwidth: 125 KHz (options are 250 KHz and 500 KHz)
- Coding Rate: 1 (options are 1 – 4)
- Programmed Preamble: 12
- RF output power: 22 (range 0 – 22)

We tried changing the Spreading factor to 12, and found a small improvement in range over the default. We tried increasing the bandwidth but didn’t find much difference in range. We did not change the other parameters. We did change the RF power and found, unsurprisingly, that the RF output power from the transmitter directly affected the range.

We were not able to measure the impact on battery power draw of changing the LoRa parameters, but theory tells us that the higher the RF output power and the longer that the message transmission takes, the greater the draw on the battery.

All in all, we decided to leave the output power at its maximum and use the default LoRa parameters in a deployed environment.

As a side note: LoRa module parameters are set using the AT commands in the referenced manual. They are stored in flash memory on the module; once set, they will persist (even through power removals) until they are changed via AT commands.

Most of the LoRa parameters must be the same for the Sensors and the Hub in order for successful communication. Our Hub software configures all of these parameters for the LoRa module that is installed on the Hub when setup() is run. The Sensor software assumes that the installed LoRa module is properly configured. Therefore, if the factory defaults are not used in a system, all Sensor LoRa modules should be first installed on the Hub and setup() run in order to ensure that the Sensor LoRa modules have matching parameters to the Hub.

CONTENTS OF THIS REPOSITORY.

This section provides a complete description of all folders and files in this Repository.

TeamPracticalProjects/LoRa-experiments.

This is the root directory for this project. It contains the following files of significance:

- **README.md**: The GitHub home page readme document.
- **Terms_of_Use_Licence_and_Disclaimer**.pdf: This document details the terms of use and licensing for the contents of this repository. You must agree to the contents of this document in order to have license to use any of the materials or information herein.
- **LoRa_Experiments_Overview_Document**.pdf: This document. It contains an overview of the repository, a description of the technology and a guide to the repository's contents.

Documents folder.

This folder contains documentation relevant to this repository:

- **Low_Power_LoRa_Sensor_User_Manual**.pdf: A detailed document describing the Low Power LoRa sensor, with references to the Eagle CAD files for the PCB and to the software source files.
- **Photon_2_Hub_and_Sensor_User_Manual**.pdf: A detailed document describing the Photon 2 version of the Sensor and the Hub, with references to the Eagle CAD files for the PCB and to the software source files.
- **RYLR998_EN**.pdf: Reyax data sheet for the RYLR998 LoRa module.
- **LoRa_AT_Command_RYLR998_RYLR498_EN**.pdf: Reyax manual detailing the AT command set for the RYLR998 module.
- **Arduino Uno as ICSP Programmer for Bare Bones ATmega328**.pdf: a detailed manual describing how to build a programmer for the ATMega328P chip and how to use the programmer, in conjunction with the Arduino IDE, to compile and flash code to the chip.

Hardware folder.

This folder contains electronic CAD (Eagle) files for the Sensor and Hub printed circuit boards and Fusion 360 CAD files for various 3D printed enclosures that can be used on deployed projects.

- **Photon-2-datasheet.**pdf: A screen capture of the Photon 2 data sheet from the Particle.io website.
- **Lora Sensor Development PCB** (folder): Contains 3 subfolders:
  - **Rev B Board:** contains Eagle source files, a zip folder with Eagle manufacturing files (works with JLCPCB, among other PCB vendors), pdf versions of the schematic and board layout, and a pdf version of a parts list for this board. *This is the current and released version of this board and is suitable for both development and deployment.*
  - **Rev A Board:** contains Eagle source files, a zip folder with Eagle manufacturing files (works with JLCPCB, among other PCB vendors), and pdf versions of the schematic and board layout. *This is an earlier version of the PCB. It has the same functionality of the Rev B PCB but does not have provision for an on-board battery, an external pullup/capacitor on the ATMega328P reset line, or a right angle pin header connection for a dray contact sensor. The rev B PCB supersedes this version and rev B should be used going forward.*
  - **Original.** contains Eagle source files, a zip folder with Eagle manufacturing files (works with JLCPCB, among other PCB vendors), and pdf versions of the schematic and board layout. *This is the original version of the PCB. It lacks sensor address jumpers and other features that were added for the rev B version of the board. The rev B PCB supersedes this version and rev B should be used going forward.*
- **LoRa Photon 2 PCB / LoRa_Photon2** (folder): contains Eagle source files, a zip folder with Eagle manufacturing files (works with JLCPCB, among other PCB vendors), pdf

version of the schematic and a .png version of the board layout. This is the version of the PCB that is needed for both the Hub and for the Photon 2 Sensor.
- **Hub Case** (folder): Contains Fusion 360 design files for a 3D printed enclosure for the Photon 2 Sensor / Hub. This is an open enclosure that provides a stable mounting for the PCB and protection for the vertically mounted LoRa module.
- **Sensor Case** (folder): Contains Fusion 360 design files for a 3D printed enclosure that is suitable for outdoor deployment of the low power, battery operated sensor. This design utilizes an innovative rainwater management system that keeps the electronics dry without requiring use of a rubber/silicone gasket.
- **Help Button Case** (folder): Contains design files for a 2 part, 3D printed enclosure for the low power, battery operated sensor. This case is simpler than the Sensor case above, but is intended for indoor use only.

Range_Testing folder.

This folder contains material relevant to the fully deployable version of the Sensor software that includes features for performing range testing as well as supporting full sensor functionality.

- **LoRa Module Range Testing Report**.pdf: a document memorializing the prototype hardware originally used for range testing experiments. The prototype hardware was subsequently replaced by low power sensor PCB (rev B) and Photon 2 Sensor / Hub PCB.
- **Range_Test_Sensor/RangeTestSensor** (folder): Particle source code for the deployable version of the Photon 2 Test Sensor.
- **Range_Test_Hub** (folder): Contains 2 sub-folders:
    - **Clip Play App** (folder): contains source code and Android installation file for an MIT App Inventor 2 app that simulates sensors on the Hub. This particular app was deployed at Maker Nexus to allow staff members to play pre-recorded administrative messages over connected Annunciators.
    - **LoRaRangeTestHub** (folder): Particle source code for the Photon 2 Hub.

arduinoSketchbook folder.

This folder contains deployed and earlier development code for the ATMega328P Low Power Sensor.

- **RangeTestSensor** (folder): contains Arduino source code for the low power, battery operated Sensor. *This is the deployable software that contains the full capability for project use as well as for development purposes*. It contains the main program file (.ino) and well as supporting library files.
- **Low_Power_Testing_Tester** (folder): contains Particle software for a hardware tester that was used to stress test the low power, battery operated sensor. Instructions for the

hardware platform and use of this software are in the comments at the front of the .ino file. *This tester does not need to be part of a LoRa Sensor system. It is memorialized here in the event it is needed for some future development effort.*

- **Interrupt_Test2_Low_Power** (folder): contains Arduino software that was used to investigate powering down the ATMega329P and using an external interrupt to power it up. *This code does not need to be part of a LoRa Sensor system; the necessary code is embedded in the deployable low power sensor software. It is memorialized here in the event it is needed for some future development effort.*
- **atmega_sensor_button** (folder): Early Arduino/Particle testing software that was used for range testing. This software is not low power and can be used with any Arduino or Particle device. *This code does not need to be part of a LoRa Sensor system; the necessary code is embedded in the deployable low power sensor software. It is memorialized here in the event it is needed for some future development effort.*

LoRa_config / config_set / config_set_rangeTest / src (folder).

This folder contains source code for a Particle Photon / Photon 2 that was used as a jig for configuring RLYR988 modules with a compatible set of LoRa parameters. The LoRa modules have a substantial set of LoRa parameters that can be configured using “AT” commands. Sensors and the Hub must be configured identically in order to communicate. *This code was originally used to configure Test Sensor and Hub LoRa modules to ensure that they would communicate. This functionality has now been built into the deployed Hub software and this code is not needed for a LoRa system. It is memorialized here in the event it is needed for some future development effort.*

Low_Power_Testing (folder):

This folder contains legacy material that was used for testing the low power modes of:

- The RLYR988 LoRa module.
- The ATMega328P microcontroller.

Individual tests were performed in order to understand how to power down these devices and to measure the power draw in the powered-down state. The functionality derived from these tests has been embedded in the deployed software for the low power battery operated sensor and none of the material in this folder is needed for a LoRa system. *It is memorialized here in the event it is needed for some future development effort.*

Integration_Testing (folder):

This folder contains legacy material that was used for testing the integration of the RLYR988 LoRa module with an ATMega328P microcontroller. Both basic (not low power) and powered down low power operation was developed and tested for operation and reliability. The functionality derived from these tests has been embedded in the deployed software for the low power battery operated sensor and none of the material in this folder is needed for a LoRa system. *It is memorialized here in the event it is needed for some future development effort.*