Photon 2 Hub and Sensor User Manual

By: Jim Schrempp and Bob Glicksman; v1, 7/28/2025

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 Particle[1] Photon 2 based circuit board that serves as either a LoRa communication Hub or a LoRa test sensor. Communication Hub software is flashed to the Photon 2 as part of a deployed LoRa system consisting of one or more sensors that send LoRa messages to the Hub when a sensor is tripped. Alternatively, Test Sensor software can be flashed onto the Photon 2 for research and experimental purposes. A low power, battery operated sensor is intended for use in deployment; see the document in this folder:

Low_Power_LoRa_Sensor_User_Manual.pdf

Test Sensor software is provided for the Photon 2 that is not low power. The USB serial port on the Photon 2 provides significant logging information for experimental use when powered via the same USB port by a laptop computer. This additional logging information is useful for experimental range testing and other research purposes.

The sensor/Hub uses LoRa[2] technology to send short messages over substantial distances using very low power. Note that LoRa technology is used here 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*).

When deployed as a Hub, the Photon 2 receives LoRa messages from one or more sensors, acknowledges each message back to the sender, and publishes an event to the Particle Cloud. Subscribers to this event can then process the event data in an application specific manner, such as:

- Logging the event data (sensor trip) to a Google spreadsheet
- Sending an email and/or notification to a user’s account for each such event
- Trigger playing of a voice clip on an Annunciator (see: https://github.com/TeamPracticalProjects/Annunciator)

A sensor is triggered via a contact closing or opening. A readily available magnetic reed switch window/door sensor can trigger the sensor to send out a LoRa message whenever the contact

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

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.

THEORY OF OPERATION.

A common printed circuit board can be used to create either a Hub or a Test Sensor. Schematic, board layout and Eagle CAD files are included in this repository in the folder:

Hardware / LoRa Photon 2 PCB / LoRa_Photon_2

Fusion 360 CAD files for a 3D printed case that protects the assembled printed circuit board is documented in:

Hardware / Hub Case

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

LoRa_Photon_2_schematic.pdf

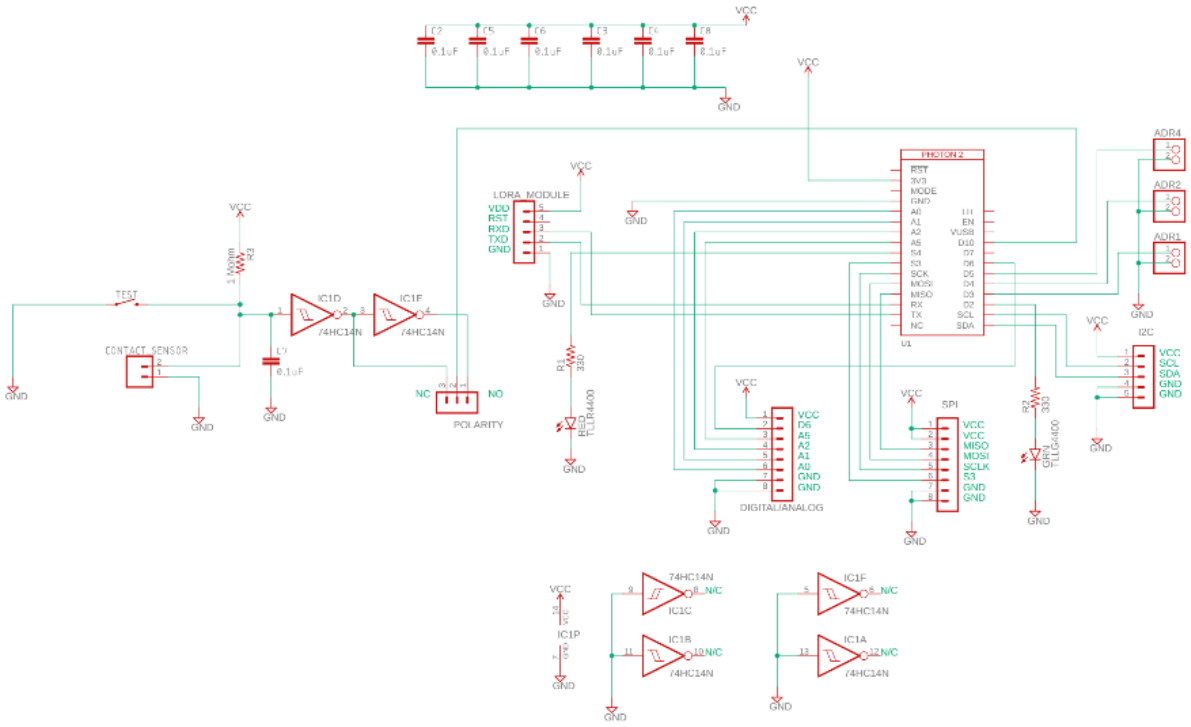

Figure 1. Hub/Test Sensor Schematic.

All of the circuitry shown in Figure 1 is needed when the circuit board is used as a Test Sensor. Many parts may be left off of the board when it is used as a Hub.

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 be read by the Photon 2 software on I/O pin D10.

LoRa Module.

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

The RLYX998 LoRa module communicates with the Photon 2 microcontroller using the microcontroller's second on-board hardware serial port (serial port 1; Rx and Tx pins). 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 Photon 2 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 Photon 2 microcontroller via 330 ohm current limiting resistors (microcontroller pins S4 and D2, respectively). These LEDs are used by the software to provide operational status for debugging, testing and monitoring purposes.

The circuit is powered by supplying +5 volts to the Photon 2's on-board USB connector. The Photon 2's on-board regulator supplies 3 volt power out to the rest of the circuitry on the 3V3 pin.

An external dry contact sensor can be connected to the circuit board using a two position terminal block (CONTACT_SENSOR).

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 Photon 2 must be reset after the address jumpers are changed in order to change the sensor's address*.

Switches and Buttons.

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 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:

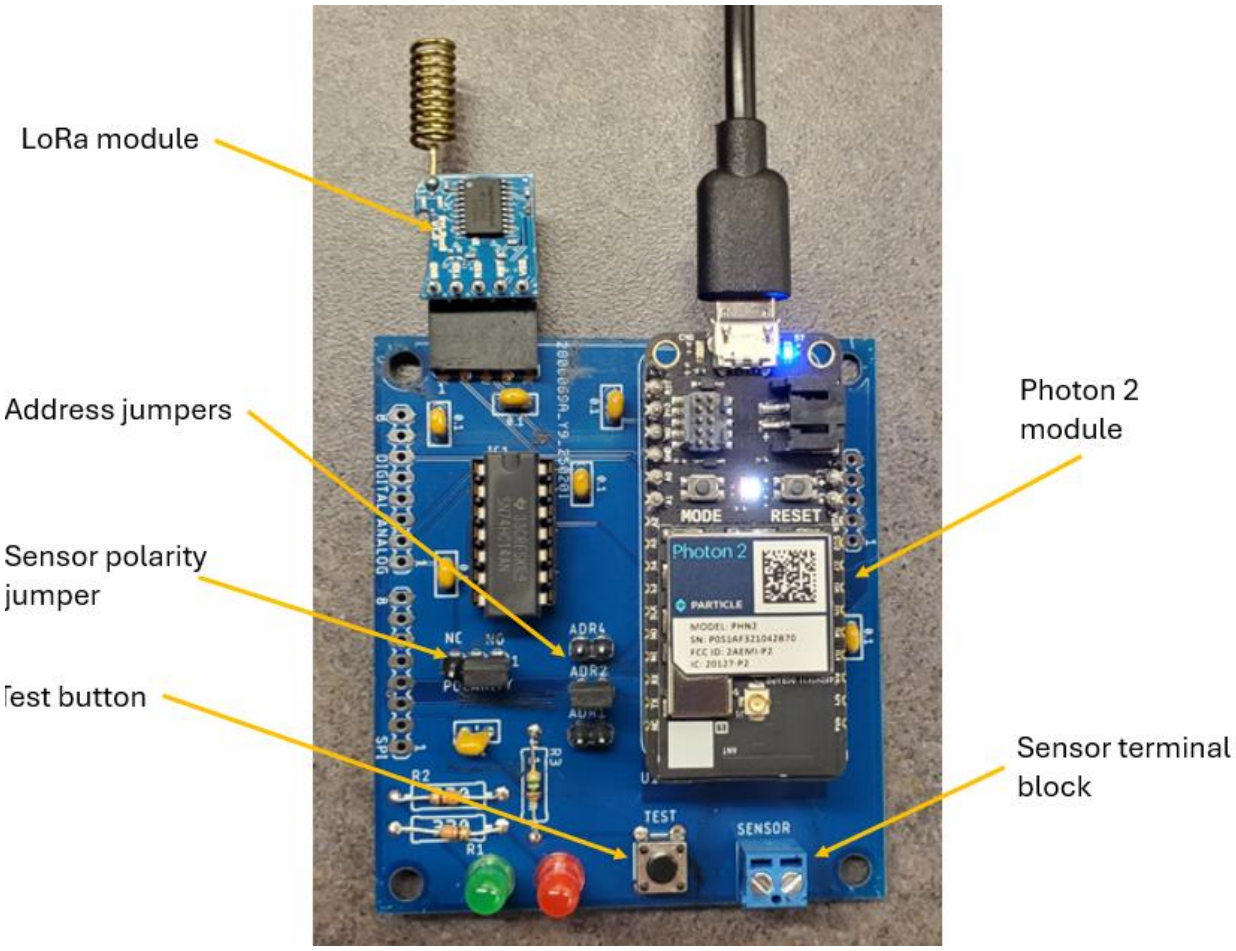

Figure 2. Assembled Hub/Test Sensor Printed Circuit Board.

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

Hardware / LoRa Photon 2 PCB / LoRa_Photon_2_2025-01-31.zip

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

Optional Components on the Hub PCB.

We recommend including all components on the Hub printed circuit board that are shown in figure 2. This way, the assembled printed circuit board may be used as either a Hub or a Test Sensor simply by flashing the relevant software to the Photon 2. However, the Hub software only requires the following components to operate:

- Photon 2
- LoRa module

- Decoupling capacitors C2, C3, C4, C5, C6, C8

All other components may be left off of the assembled printed circuit board *if the board is only intended to be used as a Hub*.

SOFTWARE INSTALLATION AND SETUP INSTRUCTIONS.

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

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

Programming the Microcontroller.

Programming the Hub software.

The code that makes this device into a Hub is located in this repository at:

/ Range_Testing / Range_Test_Hub / LoRaRangeTestHub / src / LoRaRangeTestHub.ino

This code file has a number of #include statements to include the following library files:

- tpp_LoRaGlobals.h
- tpp_LoRa.h
- tpp_LoRa.cpp

Cloud compile and cloud flash this code to the Photon 2 using either the Particle Workbench or the Particle Web IDE.

There are several other files in the / src / folder:

- *Webhook.txt*: This is the json representation of a webhook that can be located in the Particle Cloud. This webhook subscribes to the events published by the Hub and sends an https: POST to a Google Apps Script that logs the event information to a Google Sheet.
- *GoogleAppsScript.js*: source code for a Google Apps Script that is bound to a Google Sheet and appends event message data to the Google Sheet.

Use of the webhook and script is application dependent, but logging to a Google Sheet is generally useful for experimentation and development.

Programming the Test Sensor software.

The code that makes this device into a Test Sensor is located in this repository at:

/ Range_Testing / Range_Test_Sensor / RangeTestSensor / src / RangeTestSensor.ino

This code file has a number of #include statements to include the following library files:

- tpp_LoRaGlobals.h
- tpp_LoRa.h
- tpp_LoRa.cpp

Cloud compile and cloud flash this code to the Photon 2 using either the Particle Workbench or the Particle Web IDE.

Configuring the LoRa Module.

The RYLR998 LoRa module must be configured properly in order to be used in either a Hub or a Test 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 a 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 a 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 a 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. If the LoRa module is for the Hub, it is simply left in place on the Hub.

HUB CLOUD FUNCTIONS.

The Hub software provides a Particle Cloud Variable and a Particle Cloud Function. These are accessible via the Particle Console. Log in to the Particle Console with the account where the Hub Photon 2 is claimed. Select the Hub Photon 2 device from the list of devices shown in the "My devices" page in the Console. You will see the following in the bottom right portion of the screen:

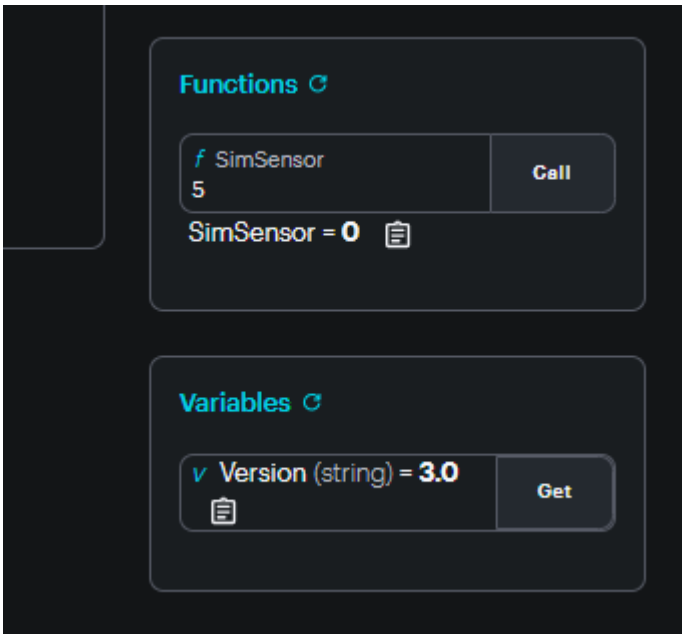

Figure 3. Particle Variable and Function.

There is one Cloud Variable called "*Version*". Clicking on "Get" will display the Hub code internal version number (3.0 in the figure above).

There is one Cloud Function called "*SimSensor*". Entering a number (5, in the figure above) and clicking on "Call" will cause the Hub to publish a sensor trip event to the Particle Cloud as if the event was triggered via receipt of a LoRa message from an actual LoRa sensor with this device ID. The "*SimSensor*" Cloud function is useful for testing subscribers to the published events. Additionally, an app can be written to make a REST API call to this function in the Particle Cloud to trigger the Hub to publish events for simulated sensors. An example of such an app is in:

/ Range_Testing / Range_Test_Hub / Clip Play App

The use of such an app is application dependent.