Mailbox Notifier
Technical Overview Document

Version 2; 9/15/25

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/Mailbox_Notifier/tree/main/Documents/Terms_of_Use_License_and_Disclaimer.pdf

TABLE OF CONTENTS.

INTRODUCTION.

The “Mailbox Notifier” project (aka “I’ve got mail!”) is an application of the Team Practical Projects’ Low Power LoRa[1] sensor technology to meet a specific need; see:

https://github.com/TeamPracticalProjects/LoRa_experiments

The need in question is a mailbox that is located out on a public street some distance from the residence that it serves. Mail delivery in the area is unreliable and it is helpful to be notified when the mail is delivered. The mailbox is instrumented with a magnetic reed switch door sensor and a battery-operated low power LoRa Sensor in a weatherproof enclosure:

Figure 1. Mailbox Notifier Installed on a Mailbox.

Opening the mailbox triggers the low power LoRa Sensor to send a LoRa message to a central Hub that is located inside of the residence. LoRa technology ensures reliable communication over a substantial open-air distance and then through the interior walls of the residence.

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

The Hub uses Particle[2] Cloud publish/subscribe technology in conjunction with a Google App Script to ultimately send an SMS text or a Pushover[3] notification to the user's mobile phone, indicating that the mailbox has been opened (i.e. to deliver the mail).

This project can also be used to indicate when a remote gate or door is opened, or similar applications.

MAILBOX NOTIFIER SYSTEM DESCRIPTION.

Mailbox Notifier System Technical Overview.

Figure 2 provides an overview of the components and communication flows in the Mailbox Notifier system:

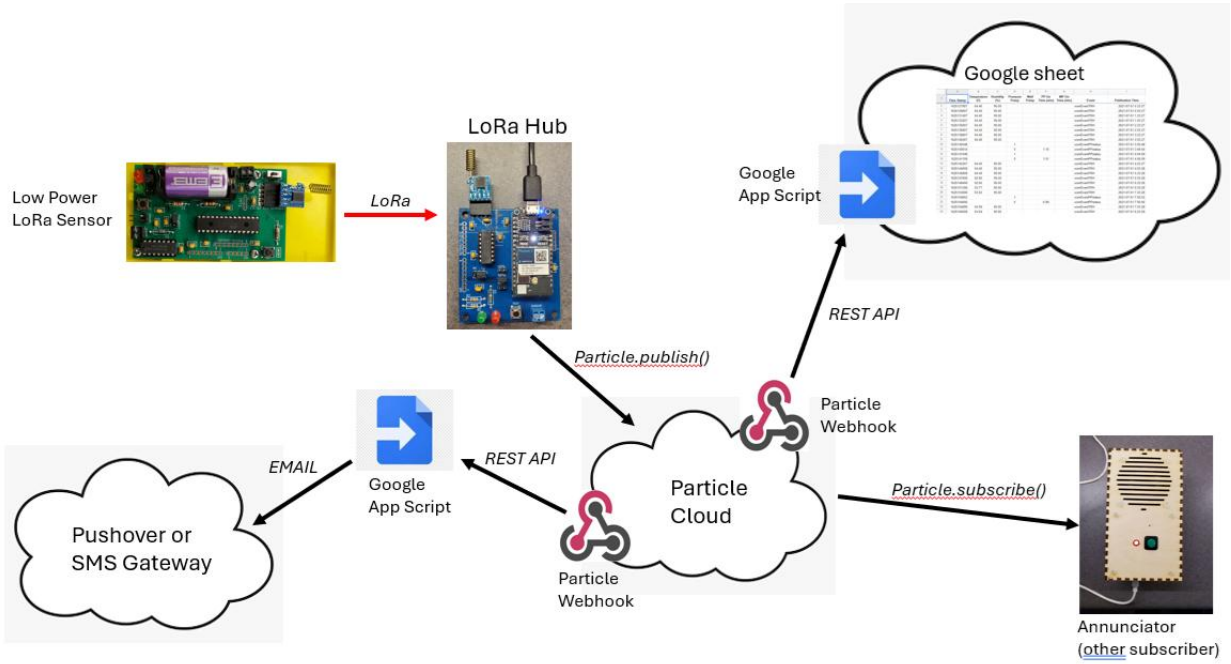

Figure 2. Mailbox Notifier System Overview.

The magnetic reed switch affixed to the mailbox body closes when the mailbox door is opened. The closing of this switch "wakes up" the Sensor's electronics resulting in transmission of a short LoRa message to the Hub. The Hub receives the message, validates the message, and sends a reply message back to the Sensor. The Sensor then returns to its very low power "deep sleep" condition. Meanwhile, the Hub publishes a "LoRaHubLogging" event to the

[2] Particle.io
[3] Pushover.net

Particle Cloud. The data associated with this event includes the Sensor “deviceID” that triggered this process and some signal quality information for logging and analysis purposes.

Two Particle Webhooks subscribe to this event. Each Webhook triggers an https: POST to a Google App Script. These scripts are deployed as “Web Apps”:

- Logging Script: The Logging Script is bound to a Google Sheet. It adds a localized timestamp to the event data and appends this information to the Google Sheet for logging and analysis purposes. This feature is *optional*; it is used for development and testing purposes only. The details of the script and the Google Sheet are application dependent. The following document describes this process in greater detail: https://github.com/TeamPracticalProjects/Connectivity_Tools_with_Particle_Devices/blob/main/TPP_Connectivity_Tools_Technical_Description.pdf

- Notification Script: The Notification Script source code is posted here: https://github.com/TeamPracticalProjects/Mailbox_Notifier/blob/main/Software/GoogleAppScript/Google_App_Script_for_Pushover.txt. This script must be located in a Google account that has Gmail associated with it. The script sends a Gmail email message either to the user’s Pushover account or to the user’s cellular provider’s SMS gateway. The former causes a Pushover notification on the user’s mobile phone. The latter sends an SMS text message to the user’s mobile phone. Either way, the user is notified that the mailbox has been opened, usually within a few seconds of the actual event. See the following document for more information about this process: https://github.com/TeamPracticalProjects/Connectivity_Tools_with_Particle_Devices/blob/main/TPP_Connectivity_Tools_Technical_Description.pdf

A device called an “Annunciator” may *optionally* be provided with the system. The Annunciator also subscribes to the “LoRaHubLogging” event publication. The Annunciator flashes a large green LED and plays a pre-recorded audio clip whenever this event is published. The audio clip played depends upon the deviceID in the published event. The pre-recorded audio clip announces that mail has been delivered. The Annunciator is a separate component that can be used on a number of projects. It is described in its own repository:

https://github.com/TeamPracticalProjects/Annunciator

Figure 3. Annunciator.

Low Power LoRa Sensor Technology.

The LoRa Sensor is a battery operated low power device that powers up and sends out a LoRa message whenever the magnetic reed switch on the mailbox or gate closes. The LoRa message contains a “deviceID” that is configured, via jumpers, on the Sensor’s printed circuit board, to one of 8 values relative to a base device address. Eight distinct sensors may be deployed in one system by setting these jumpers accordingly. [If more than 8 distinct device IDs are needed in a system, the base address of a set of 8 IDs can be changed in the Sensor firmware]. A Sensor must be reset after changing device ID jumpers in order for the new device ID to take effect. See the following document for complete information about the low power battery operated Sensor:

https://github.com/TeamPracticalProjects/LoRa_experiments/blob/main/Documents/Low_Power_LoRa_Sensor_User_Manual.pdf

The system described here uses only one such Sensor but additional sensors can be deployed. Each Sensor will cause a “LoRaHubLogging” event to be published to the Particle Cloud by the single Hub. Downstream processing can determine which Sensor sent the LoRa message by the “deviceID” that is part of the event data string.

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

Each Sensor sends out a LoRa message when it is triggered. The LoRa messages sent out from each Sensor contains:

- The Sensor's device ID
- A very brief code indicating the device 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. These are logged and used to ensure that communication remains reliable over time.

The Sensor is mounted outdoors and exposed to the elements. A special 3D printed enclosure has been developed to keep rainwater away and to help convection cool the electronics. The enclosure is printed in PETG material. Enclosure CAD files are located here:

https://github.com/TeamPracticalProjects/LoRa_experiments/tree/main/Hardware/Sensor%20Case

The Sensor is battery operated and therefore standalone. The power draw on the CR123A Lithium battery is minimal (around 10 microamperes) when the Sensor is not activated. Modeling predicts that the battery should last several years under routine conditions.

There is one Hub for any given system. The Hub operates on LoRa Channel 18, and all Sensors must use this same channel (and the same LoRa settings) in order to communicate with the Hub. The Hub is configured (in the Hub firmware) with a specific LoRa device address. The Hub device address must be different from any Sensor's LoRa device address and must be unique for any given system. Note that LoRa device address is configured in the firmware and is distinct from the Sensor's device ID, which is configured by jumpers on the Sensor board. The following document contains further information about the Hub:

https://github.com/TeamPracticalProjects/LoRa_experiments/blob/main/Documents/Photon_2_Hub_and_Sensor_User_Manual.pdf

When the Hub receives a LoRa message, the Hub firmware 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 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.

A 3D printed mount for the Hub is documented here:

https://github.com/TeamPracticalProjects/LoRa_experiments/tree/main/Hardware/Hub%20Case

This is an open frame mount with protection for the LoRa module and its antenna.

LoRa Technology.

LoRa communication is via Reyax RYLR998 LoRa modules. These modules are low cost (approximately $12 each) and contain a LoRa transceiver and a low power microcontroller. The Reyax RYLR998 module acts as a “LoRa Modem” and uses serial asynchronous “AT” commands to communicate with a host computer or microcontroller.

Figure 4. Reyax RYLR998 LoRa Module.

The “AT” commands are documented in the repository:

https://github.com/TeamPracticalProjects/LoRa_experiments/blob/main/Documents/LoRa_AT_Command_RYLR998_RYLR498_EN.pdf

BUILDING AND DEPLOYING THE MAILBOX SENSOR.

The Mailbox Sensor consists of the following parts:

- LoRa Low Power Sensor electronic printed circuit board.
- 3D printed weatherproof enclosure.
- Magnetic reed switch.
- Waterproof pigtail connector.
- Hookup wire.

Figure 5 identifies these components:

Figure 5. Mailbox Sensor Components.

Fusion 360 CAD files for the 3D Printed enclosure can be found here:

https://github.com/TeamPracticalProjects/LoRa_experiments/tree/main/Hardware/Sensor%20Case

The enclosure consists of three parts:

- Mounting bracket (bracket.stl): the mounting bracket is affixed to the mailbox at a convenient location, using either outdoor double sided tape or screws.
- Enclosure body (Bottom square.stl): the enclosure "box" with tabs that engage with the Mounting bracket.
- Enclosure top (Top.stl): the enclosure cover that also mounts the electronic printed circuit board.

Any standard alarm type magnetic reed switch can be used to sense when the mailbox door is opened. We used this one:

https://www.amazon.com/dp/B094XPMFGK?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1

This particular reed switch is held open when the magnet is close by and closes when the magnet is separated from the switch. This configuration (normally open) is desirable but not mandatory. If a normally closed reed switch is used, the polarity jumper on the Sensor printed circuit board should be moved from "NO" to "NC". Many commercial reed switches have both NO and NC contacts. The most important characteristic of the magnetic reed switch is that it will be mounted outdoors and should therefore be as waterproof as possible.

The magnet is mounted on the mailbox door and the reed switch is mounted on the mailbox body immediately adjacent to the magnet. These components can be mounted using outdoor rated double sided tape or else using screws.

In order to facilitate easy removal of the electronics from the mailbox (for servicing, such as battery replacement), a waterproof connector is used in between the Sensor electronics and the magnetic reed switch contacts. We used the following part:

https://www.adafruit.com/product/743

Ordinary hookup wire (#22 - #24 awg) is used to connect the reed switch to the pigtail connector body and from the pigtail connector plug to the Sensor electronics. Stranded wire is recommended for flexibility. The pigtail leads are soldered to hookup wire and waterproofed using heat shrinkable tubing and electrical tape. The connections to the reed switch and the Sensor electronics are not polarized.

The hookup wire is fed through a hole in the side of the enclosure and knotted for strain relief. Standard Dupont female straight headers are crimped to the ends of the hookup wires and connected to the right-angle male pin headers on the Sensor electronic printed circuit board that is mounted inside of the enclosure Top. Figure 6 shows the major components of the Sensor electronics printed circuit board:

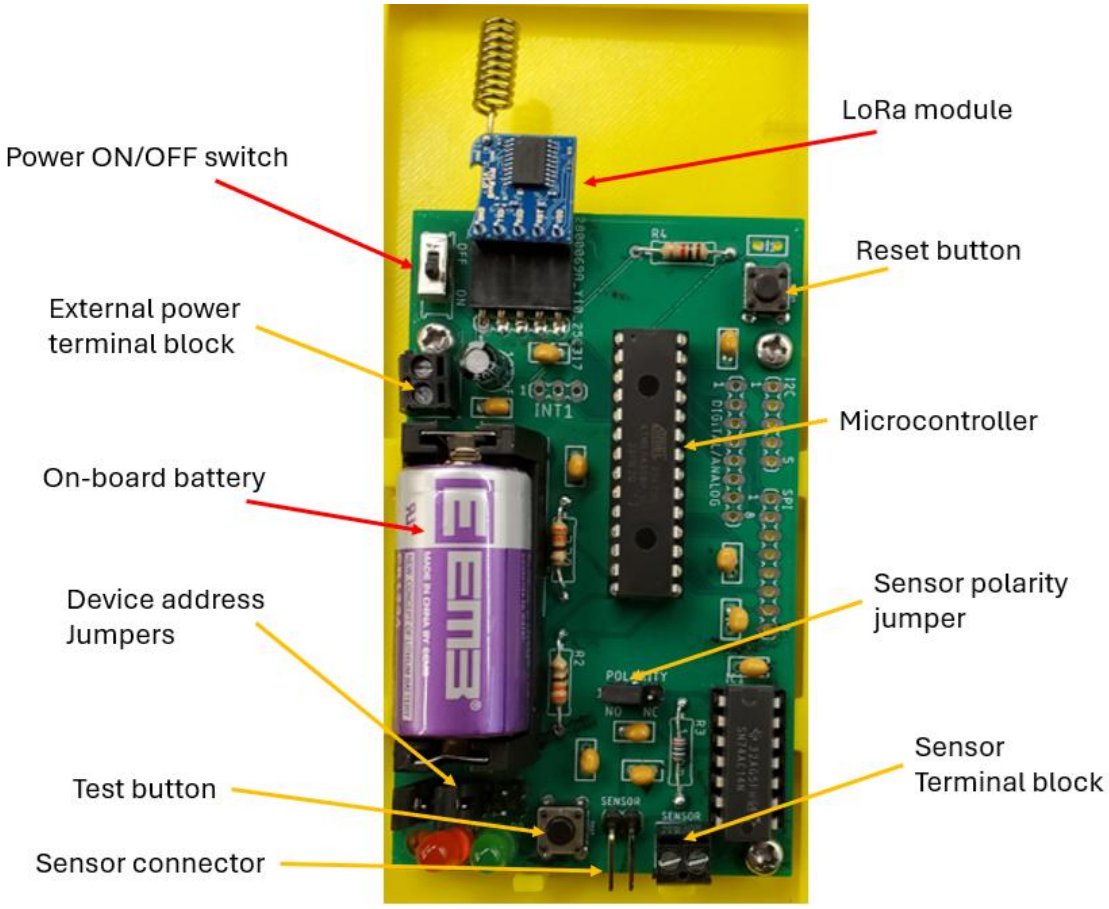

Figure 6. Sensor Electronic Printed Circuit Board.

Note in Figure 6 the location of the Sensor connector at the bottom of the board. The Sensor Terminal Block should not be used for deployment; it is provided for testing purposes only.

Complete instructions for building and configuring the Sensor electronic printed circuit board are in the following document:

https://github.com/TeamPracticalProjects/LoRa_experiments/blob/main/Documents/Low_Power_LoRa_Sensor_User_Manual.pdf

A jumper must be placed between the center and either the "NO" or "NC" position of the Sensor polarity jumper field or else the Sensor will not trigger! The jumper should be in the "NO" position for the reed switch referenced above.

Placement of the Device address jumpers is optional, but it is best to insert all three jumpers for the first Sensor in the system. Additional Sensors, if used, must be jumpered to unique addresses.

Please be sure to follow the complete instructions for building the printed circuit board, including the SOFTWARE INSTALLATION AND SETUP INSTRUCTIONS.

BUILDING AND DEPLOYING THE HUB.

The Mailbox Notifier Hub consists of a printed circuit electronic board and an optional 3D printed mount. Complete instructions for assembling and programming the printed circuit board can be found in the document:

https://github.com/TeamPracticalProjects/LoRa_experiments/blob/main/Documents/Photon_2_Hub_and_Sensor_User_Manual.pdf

An optional 3D printed mount holds the printed circuit board and protects the LoRa module that is plugged into it. 3D CAD files for this mount can be found at:

https://github.com/TeamPracticalProjects/LoRa_experiments/tree/main/Hardware/Hub%20Case

Please be sure to follow the complete instructions for building the printed circuit board, including the SOFTWARE INSTALLATION AND SETUP INSTRUCTIONS.

NOTE: The Hub Photon 2 device and all Annunciator Photon devices must all be claimed into the same Particle account in order for the Publish/Subscribe mechanism to work.

BUILDING AND DEPLOYING ANNUNCIATORS.

Instructions for building, programming, operating and testing Annunciators can be found at:

https://github.com/TeamPracticalProjects/Annunciator/blob/main/Documentation/Annunciator_User_Manual.pdf

Annunciators are housed in a laser cut enclosure. 2D CAD files for this enclosure can be found at:

https://github.com/TeamPracticalProjects/Annunciator/tree/main/Hardware/LasercCutEnclosure

NOTE: The Hub Photon 2 device and all Annunciator Photon devices must all be claimed into the same Particle account in order for the Publish/Subscribe mechanism to work.

LOGGING.

Logging of Mailbox Notifier events to a Google sheet may be accomplished as follows:

- Configure a Particle Webhook that subscribes to the “LoRaHubLogging” event publications (by the Hub). The Webhook should POST to a Google App Script that appends the event data to a Google sheet.
- A Google Sheet cloud spreadsheet is created to house the logged event data.
- A Google App Script is created with a doPost(e) method that parses the event data out of the event object “e” that is the argument to the doPost() function.
- The script may append local date/time and otherwise process and parse the event data in order to format it for logging to the Google sheet.
- The url of the Google App Script is configured into the url field of the Particle Webhook.
- The url of the Google Sheet spreadsheet and the name of the sheet that the data is to be logged to are configured into the Google App Script.

The details of the script depend upon what information is to be logged and in what format. More information about this process can be found in:

https://github.com/TeamPracticalProjects/Connectivity_Tools_with_Particle_Devices

NOTIFICATION.

The Mailbox Notifier user is notified of mailbox opening events using either Pushover event notification or an SMS text. The Pushover app must be installed on the user's mobile phone to use Pushover notifications; see:

https://pushover.net/

Note that there is a one-time in-app charge of $5 in order to use this app. An alternative to Pushover notifications is to send an SMS text to the user's mobile phone via the user's cellular carrier's SMS email gateway. Not all carriers provide an SMS email gateway.

In either case, a notification or SMS text is triggered by sending a Gmail email to the Pushover gateway or an SMS email gateway. Pushover provides the user with a unique email address for this purpose. Each cellular carrier that provides an SMS email gateway has its own email destination system for users. The following document provides more information about this type of integration:

https://github.com/TeamPracticalProjects/Connectivity_Tools_with_Particle_Devices

A Google Apps Script that sends a Gmail email to a configured Pushover or SMS gateway email address is contained here:

https://github.com/TeamPracticalProjects/Mailbox_Notifier/blob/main/Software/GoogleAppScript/Google_App_Script_for_Pushover.txt

This script must be deployed as a "Web App" and must be in a Google account that has a Gmail email associated with it.

Lastly, a Particle Webhook must be created that subscribes to the "LoRaHubLogging" event. It performs an https: POST to the url of the deployed Google Apps Script. The published script does not process the event data because there is only one Sensor in the Mailbox Notifier System. If additional Sensors are used in a system, the script should parse the "deviceID" out of the event data string and format an appropriate message in the email body.