Water Leak Detector
Installation and User Manual

date: 12/21/22

<u>NOTICE</u>: 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 also be found at:
https://github.com/TeamPracticalProjects/WaterLeakSensor/blob/master/Documentation/Terms_of_Use_License_and_Disclaimer.pdf

Water Leak Detector

Table of Contents

1. Overview

The primary purpose of the Water Leak Detector (WLD) is to detect leaks and to notify the user in time to contain the damage while it is still minor. To this end, the WLD provides both a local audible and visual notification and a remote notification to the user's mobile phone. The WLD is designed to support two sensors, each of which may be placed anywhere that can be cable connected to the WLD electronics.

Secondarily, the WLD contains an inexpensive temperature and humidity sensor (DHT11) that monitors ambient conditions and may provide useful augmentation to the water leak sensor data; e.g. an indication of a steam leak in a boiler in the basement of a house.

The sensors used in the WLD are actually water level sensors whose data is processed to provide an alarm condition in the event of a puddle of water less than 2 mm in height. An alarm condition exists whenever either of the sensors detects the presence of water. The alarm condition persists until both sensors are dry, in which case the WLD automatically clears out the alarm condition and re-arms for subsequent triggering of a water leak alarm.

The WLD electronics are based upon the Particle Photon module (https://www.particle.io). The Photon contains an advanced 32 bit microcontroller with a built in Wi-Fi capability. The WLD electronics requires a Wi-Fi connection to the Internet and will not work (even locally) unless the Photon can communicate with the Particle cloud over the Internet. The Particle cloud service is available to users of Particle's devices with a generous free monthly service allotment.

The WLD is designed to be accessed remotely via a smartphone. WLD uses a custom app (part of this project) to provide the user with current temperature, humidity and alarm status information, as well as to allow the user to set low and high temperature alarm limits. The app is written in MIT App Inventor 2 (http://ai2.appinventor.mit.edu/) and is currently supported on Android smartphones and tablets[1] (including Amazon Fire tablets). Water leak and temperature alarms are sent to the user's mobile device using SMS texting.

The WLD is offered to all users free of charge for non-commercial use. Any and all use of the information provided in this package is subject to the restrictions described in the document "Terms_of_Use_License_and_Disclaimer.pdf" that is included in this release package. This document can also be found at:

https://github.com/TeamPracticalProjects/WaterLeakSensor/blob/master/Documentation/Terms_of_Use_License_and_Disclaimer.pdf

The WLD is only available as a set of documentation at this time. You must purchase all of the parts yourself and you must assemble the WLD yourself, including soldering of electronic

[1] Support for iOS devices is currently in beta testing with Apple.

components and cabling, cutting and assembling of plastic material, mounting of materials on walls, setting up of accounts on cloud-based services and smartphones, and following written instructions. You will need basic hand tools (screwdrivers, files, knives, drill) and basic through hole soldering tools (soldering iron, long nose pliers, diagonal cutters, wire strippers) and the skills to use these tools properly in order to assemble the WLD.

We have endeavored to provide full, accurate, step by step instructions for each and every step in making a fully operational WLD. However, we explicitly disclaim any responsibility or liability for inaccuracies or confusion in any of our documentation. ***We are not a commercial entity and we cannot offer any guarantee of support or warranty of any kind if you are unable to successfully build and deploy a WLD.*** Therefore, *we urge you to read through all of the documentation for this project and make sure that you are comfortable with everything that you will need to do to make a working WLD before proceeding to order parts or begin the manufacture of any of the WLD components.*

2. Component Parts

An overview of the WLD can be found in the document “Water_Leak_Dectector_Concept.pdf” which is included in this package and can also be found at:

https://github.com/TeamPracticalProjects/WaterLeakSensor/blob/development/Documentation/Water_Leak_Detector_Concept.pdf.

Please read through this document first so that you are familiar with the WLD overall, before proceeding to read the details below. The WLD project consists of hardware, firmware (for the Photon), a Google Apps Script (deployed as a web app), and an app for your smartphone. These items are described in this section.

2.1. Hardware.

The WLD hardware consists of the following components:

- The water leak sensors
- The WLD electronics
- Your Wi-Fi router
- Your smartphone

2.1.1. Water Leak Sensors.

Figure 2-1 is a picture of the sensors that are used in this project. A US quarter is included in the photo to provide an indication of the size of these sensors (they are small). The sensors are intended to measure the level of water, where the bottom of the sensor is at the left of the photo and the electronics (right) are at the top.

Figure 2-1. Water Leak Sensor.

Each sensor consists of a set of parallel conductive traces arranged as interleaved “fingers”. These traces are an open circuit when the sensor is dry. However, when the sensor encounters water at some level up from the bottom, the resistance of these traces lowers and the sensor outputs a voltage that is indicative of the level of the water between the bottom (lowest voltage) and the top of the traces (supply voltage).

The WLD is only interested in the presence of water. The WLD firmware in the Photon continuously measures the voltage from each of the sensors and compares this voltage to a threshold. When this threshold is exceeded for a set number of measurements, a water leak alarm is triggered. The alarm remains triggered until both sensors are dried off and thus below the alarm threshold for a set number of measurements. We have empirically set the threshold so that an alarm is triggered by a water level that is less than 2 mm from the bottom of the sensor.

The WLD accommodates two sensors so that one WLD can monitor two different locations; e.g. by a washing machine and by a water heater in a basement or garage. The sensors themselves connect using a 3 pin 0.1" spaced header. In order to cable the sensors to the electronics neatly and securely, the WLD design includes pin header to RJ11 connector boards. A short pin header cable at the sensor end connects to a miniature circuit board that wires it to an RJ11 telephone connector. A similar board and connector are used in the electronics enclosure to transition back from RJ11 cable to the WLD electronics. RJ11 cables are available on-line and in many hardware stores. They are inexpensive, available pre-terminated in many different cable lengths and in different colors for running along baseboards and up walls. They are jacketed, relatively rugged, and the connectors have a locking mechanism so that the cable won't fall out of its mating socket. **Make sure that you use 4 wire (not 2 wire) RJ11 cable**. 2 wire RJ11 telephone cables will not work!

2.1.2. Electronics.

The WLD electronics are contained in a plastic enclosure. Figure 2-2 shows this enclosure with sensor cables, power cable and mounting bracket attached.

Figure 2-2. Assembled and Cabled Electronics Enclosure.

A plastic meter faceplate is attached to the cover of the box. The faceplate is laser cut from 3 mm plastic stock using a CAD file that is included in this project package. If you do not have access to a laser cutter, you can print out a paper copy of this faceplate from the included pdf file and carefully glue it to the cover of the box.

The faceplate mounts the "servo" meter, which indicates temperature and humidity locally. The toggle switch to the left of the servo meter is used to select whether the servo meter displays temperature or humidity. Both Fahrenheit and Centigrade scales are included on the faceplate. The humidity scale is always %RH.

The right side of the faceplate contains a backlit pushbutton switch. The backlight lights steady when the WLD is armed and ready. The backlight blinks rapidly when a water leak alarm condition is present. A piezo buzzer inside of the enclosure beeps annoyingly when a water leak condition occurs, but pressing the backlit pushbutton mutes the audible alarm until the system is automatically re-armed after the sensors are dried off.

The project documentation also includes a template for laser cutting a wall mounting bracket for the enclosure. This bracket may optionally be used both as a template for drilling mounting holes for the circuitry inside of the enclosure as well as providing holes for wall mounting the electronics enclosure after it is assembled and tested.

The two water level sensors are connected to the left side of the enclosure using RJ11 cables. The right side of the enclosure contains a USB "B" connector which is used to supply power to the enclosure. *The enclosure must be placed in a location where AC power is available*.

Figure 2-3 shows the inside of the electronics enclosure.

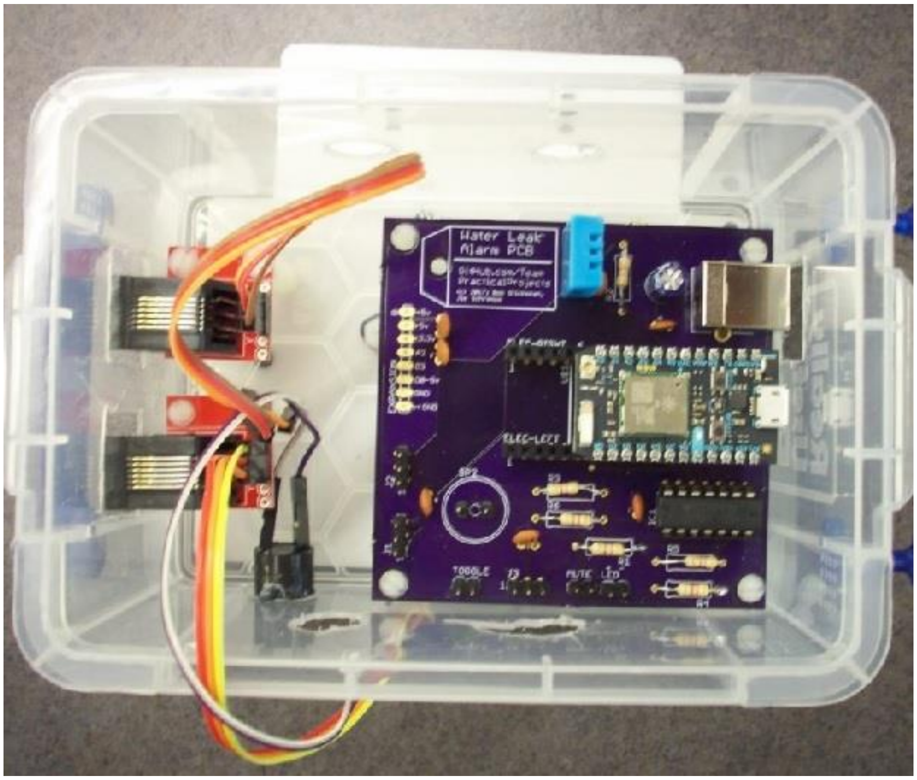

Figure 2-3. Inside View of Electronics Enclosure.

On the left side of the enclosure are two small printed circuit boards that transition RJ11 connections to pin header cables. The RJ11 jacks are accessible through holes cut through the side of the box. The main electronics printed circuit board is mounted on the right side of the enclosure with the USB "B" type power connector accessible through a hole cut through the side of the box. Figure 2-3 shows the piezo buzzer glued to the lower side of the box; however, the released version of the main printed circuit board allows the buzzer to be soldered directly to the board, obviating cutting and gluing it to the box itself.

2.1.3. Other Hardware.

The WLD project requires a Wi-Fi connection to the Internet. You must have a Wi-Fi router or gateway that is compatible with the Particle Photon – specifications can be found at:

https://www.particle.io/products/hardware/photon-wifi-dev-kit

The WLD project also requires that you have an app on a smartphone in order to receive current status information from the WLD. A suitable app for Android phones is included in this project[2]. You will need a mobile phone (not necessarily a “smart” one) to receive water leak and temperature alarms via SMS texts.

2.2. Firmware.

Firmware is what we call the software that runs on the Particle Photon microcontroller. The released WLD firmware is included in this package in the Firmware folder of this repository.

You must have an account at https://www.particle.io in order to flash the firmware to your Photon device. You will also need to have Google account and an active account with a mobile carrier in order to receive SMS alarms messages from the WLD. Complete installation instructions are included in this document.

2.3. Smart Phone App.

The WLD communicates with an app on your (Android) smartphone. The code for this app is included in the App folder of this repository. Both source code (.aia file) and an Android installer file (.apk file) are included. Installation instructions are included in this document.

2.4. Script.

The WLD requires a Google Apps Script to be installed in a Google account that is owned by the user. This script generates the SMS text alarms to the user’s mobile phone (via the user’s mobile carrier SMS gateway). Installation instructions are included in this document.

[2] The Particle Console may be used in lieu of a smartphone app (see: https://console.particle.io)

3. WLD Hardware Build, Installation and Placement.

The WLD hardware must be assembled from the parts that are listed in the "WaterLeakAlarmPartsList" document that is in the Documentation folder of this repository. The electronics are contained on a custom printed circuit board. Eagle CAD files are provided in the Hardware folder of this repository. Assembly instructions for the printed circuit board can be found in the document "PCB Assembly Instructions" in the Documentation folder of this repository.

The assembled printed circuit board becomes part of the overall WLD hardware assembly by adding an enclosure and two standalone RJ11 jack modules. There are three different options for the enclosure:

- An off the shelf plastic box with a laser cut faceplate.
- An enclosure made from 2D laser cut wood or plastic parts.
- An 3D printed plastic enclosure.

The Documentation folder of this repository contains instructions for fabricating and assembling each of these enclosure options. You may choose whichever option suits you best.

Once assembled, the Servo Meter must be calibrated. See the document "Servo_Meter_Calibration" in the Documentation folder of this repository. After calibrating the servo meter, the operating firmware needs to be flashed to the Photon microcontroller in the WLD and the WLD hardware operation should be tested using the Particle Console. See section 6.1 of this document for firmware installation instructions.

Once the WLD hardware is assembled and tested, the electronics should be mounted on a wall or otherwise accessible location within cable distance of the sensors. A wall mounting bracket is available as part of this project. The wall mounting bracket allows you to assemble and test the WLD electronics and then to mount it on a wall using two #8 screws. The wall mounting bracket also serves as a template for drilling holes in the bottom of the enclosure box for assembly of the electronic components into the box. The wall mounting bracket is provided as a CAD file with this project and you may use this CAD file to laser cut the bracket out of 3 mm plastic or wood material. If you do not have access to a laser cutter, you will have to fabricate your own means of mounting the WLD electronics to a wall.

The location of the WLD electronics must:

- <u>Be accessible to AC power</u>. A "wall wart" power supply plugs into a standard wall outlet and provides power to the WLD electronics via a standard USB A/B cable. The power cable plugs into a mating USB connector located on the right side of the enclosure.
- <u>Be physically accessible</u>. You will want to mount the WLD electronics enclosure in a location where you can see the servo meter and the backlit pushbutton switch on the

face of the enclosure. You should also be able to easily operate the toggle switch and press the backlit pushbutton at this location.

- Be located where sensor cables can be run neatly to the enclosure. RJ11 cables from the two water level sensors plug into mating RJ11 jacks located on the left side of the WLD electronics enclosure.

The two water level sensors must be mounted vertically, with the bottom of the sensor in physical contact with the floor. The back of each sensor is completely blank. We recommend that you use high strength outdoor double sided tape to secure the sensor to a wall, in the proper orientation and location.

The best location for the water level sensors are along a wall that is the lowest point in the room, near where a leak would occur (sink, washing machine, water heater, etc.). If the room has its low point at a drain near the center of the room, you might want to improvise some sort of mount for the sensor, such as a block of wood. Note, however, that the sensor must be cabled to the electronics, so you will need to take care to ensure that the sensor, its mount and the cable run do not interfere with movement through the room. A better solution might be to place a flat pan or other container underneath where leaks are expected, with one edge of the pan against a wall where the sensor wires will run. This way, the water leak can puddle in the pan and trigger a sensor that is cabled along a wall.

Each sensor attaches to a miniature printed circuit board using a 3 wire female-female header cable. The printed circuit board transitions this header cable to a 4 wire RJ11 telephone cable, as shown in figure 3-1.

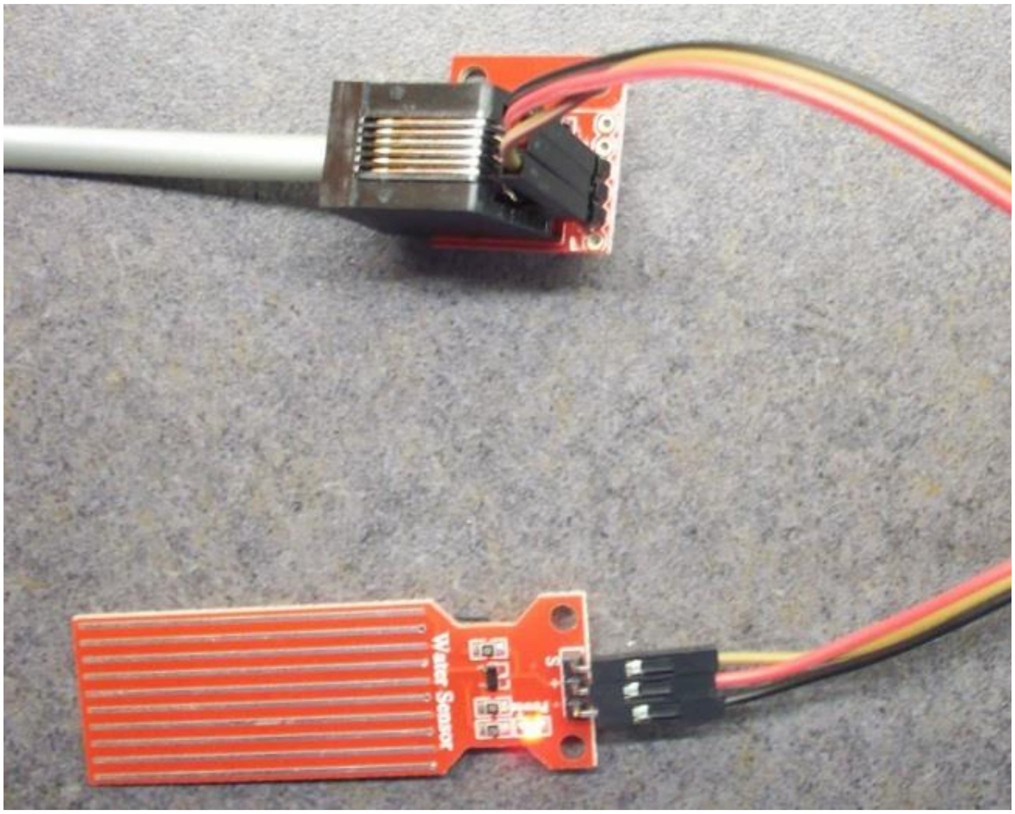

Figure 3-1. Water Level Sensor Transition to RJ11 Telephone Cable.

When locating the water level sensor along a wall or mounting block, provision must be made for similarly mounting this miniature printed circuit board nearby and for neatly routing the telephone cable from the miniature printed circuit board up to the electronics enclosure. We have tested connection of water level sensors to the WLD electronics enclosure using 10 foot long telephone cables. In theory, longer cables (25 feet or even 50 feet) should work fine; however, we have not tested longer cable connections.

Figure 3-2 is a schematic diagram for the wiring of the water level sensors through transitions to and from RJ11 jacks and cable and up to the WLD electronics printed circuit board on the enclosure.

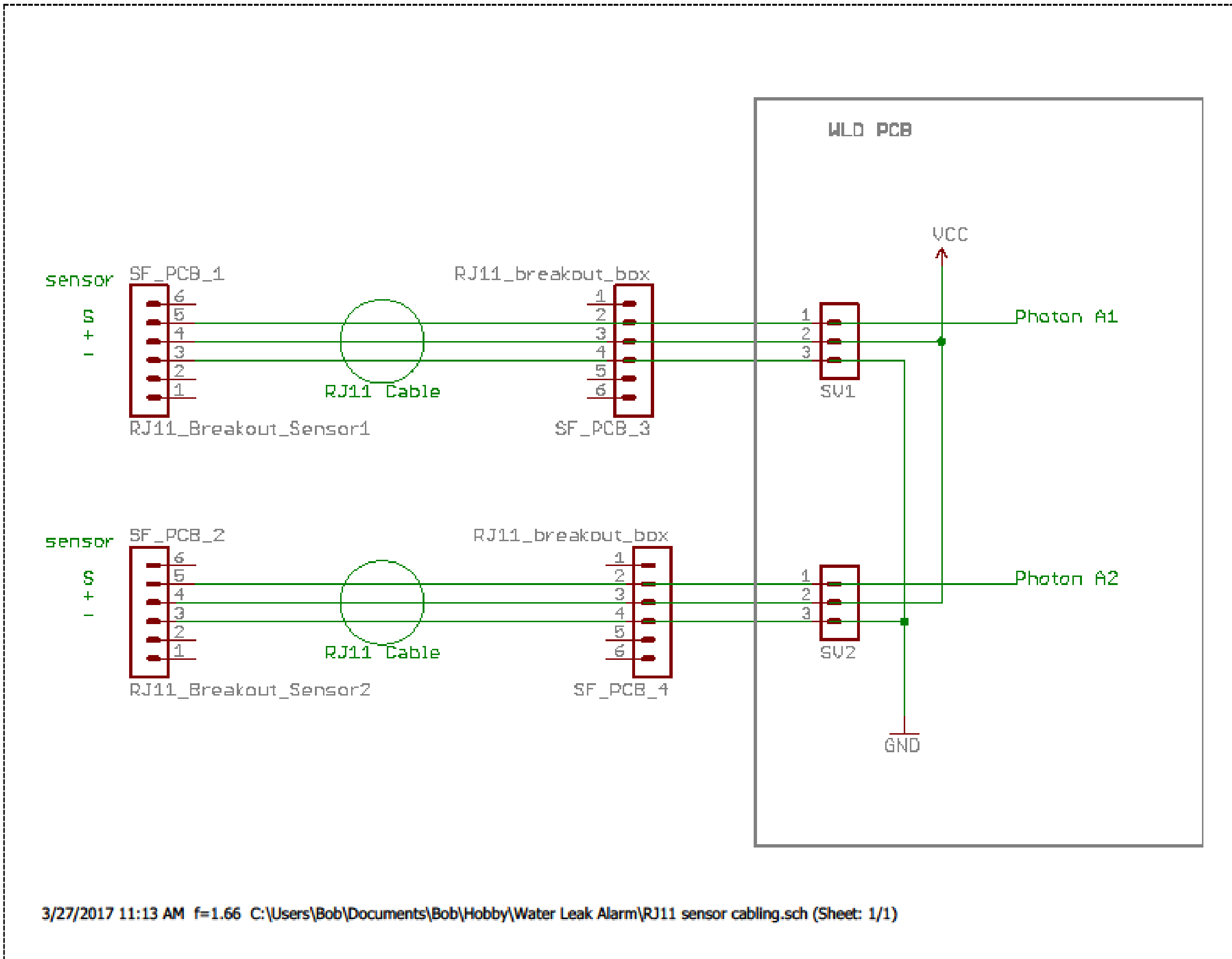

Figure 3-2. Water Level Sensor Cabling to WLD Electronics.

Note that the sensors connect to pins 5, 4 and 3 on the miniature PCB by the sensor, but the corresponding pins at the miniature PCB within the WLD electronics enclosure are pins 2, 3, and 4 respectively. ***Be careful to use the correct pins when wiring sensors to the miniature PCB and the corresponding miniature PCB to the main electronics PCB***. Further details can be found in the document “WLD_Enclosure_Build_Instructions.pdf” which is included in the Documentation folder of this repository.

The WLD supports two water level sensors, so that two different locations can be monitored for leaks (e.g. a washing machine and a water heater in a basement or garage). It is best if you use both sensors, even if the sensors are monitoring two locations nearby the same appliance. If you wish to use only one sensor, you must ground the signal input pin at the 3 pin header connector for that sensor on the WLD electronics main PCB. This can be achieved by placing a female-female pin header jumper between pin 1 and pin 3 on whichever of the sensor connectors are not used. *If you do not install either this jumper or a sensor to one of the sensor input connectors, ambient noise will trigger water leak alarms.*

4. Local Operation.

The WLD has been designed for both local and remote operation. Local operation is described in this section.

In normal circumstances, the WLD is powered up and the green backlit pushbutton is lit continuously. This indicates that the WLD is powered up, connected to the Particle cloud via the Internet, reading temperature and humidity from the DHT11 sensor, and that the water leak detection is armed.

If either of the water level sensors detects a leak, the LED in the backlit pushbutton begins to blink and the piezo buzzer makes an annoying loud sound. This provides a local user with both an audible and visual alert to the leak condition. When the local user responds to the alarm, s/he can mute the buzzer sound by pressing the pushbutton. The backlight of the pushbutton will continue to blink until the alarm condition is cleared. The alarm condition is cleared whenever both water level sensors are dry. Clearing the alarm also automatically rearms the system for subsequent leak detection.

The servo meter on the front of the WLD enclosure continuously displays either the ambient temperature or the ambient humidity, depending upon the position of the toggle switch. When the toggle switch is in the Temperature position, the servo meter indicates the temperature which can be read off of either the Fahrenheit or Centigrade scales on the meter face; whichever scale is desired. When the toggle switch is in the Humidity position, the %RH can be read off of the humidity scale. Flipping the toggle switch position instantly changes the servo meter from temperature to humidity or visa-versa.

If the light on the backlit pushbutton is off (neither flashing nor lit constantly), then the WLD is not up and running properly. This may be due to one of the following conditions:

- Power is off.
- Photon is not connected to the Internet through Wi-Fi.
- Photon is connected to the Internet but not able to connect to the Particle cloud.
- Photon is cloud connected to Particle but the DHT11 temperature and humidity sensor is not responding.

If the light on the backlit pushbutton is off, check that the power supply is plugged into a working AC outlet and that the USB cable is connected to the power supply and to the WLD electronics. If power is connected, unplug the power supply or the power cable for 5 seconds and then plug it back in. The Photon should reset and reestablish all needed connections, if possible. This may take 30 seconds or more. If everything is working, the backlit pushbutton should light continuously. Otherwise, something did not connect properly.

In order to further diagnose a problem, it is necessary to observe the multicolor LED on the Photon module and also the blue “D7” LED on the Photon. These can be seen without removing the faceplate by looking in through the transparent right side of the WLD electronics enclosure. After unplugging and re-plugging the power supply, the following observations can be made:

- Power is on in the electronics. It may take a few seconds, but the Photon’s multicolor LED should start flashing different colors after power is applied. If not, there is a power problem.
- Photon acquisition. The Photon should blink green for a while, perhaps blink some other colors (but not red), and finally “breath cyan”. Breathing cyan means that the Photon is connected to the Particle cloud via Wi-Fi to the Internet. If the breathing cyan state is not reached after 30 seconds of power, consult the Particle documentation (https://docs.particle.io/guide/getting-started/modes/photon) for further definition of the problem.
- Setup. After the Photon starts breathing cyan, it enters setup mode. In setup mode, the Photon is connecting to the DHT11 sensor. This normally takes no more than 3 seconds. If setup mode completes correctly, the Photon enters normal operation with the backlit pushbutton LED on constantly and the servo meter reading the temperature or humidity, as selected by the toggle switch. When in normal operation, the Photon’s D7 LED flashes on and off at 4 second intervals. If the D7 LED is not flashing, then the Photon has not exited setup mode and there was a problem with the DHT11 communication. Make sure that the DHT11 is plugged into its socket correctly and solidly. If the D7 LED is flashing on and off but the backlit pushbutton is not lighting, then there is a problem with the wiring to the pushbutton (perhaps a wire has broken off).

5. Remote Operation (App, SMS Alarms).

Remote access to the WLD is made possible through a smartphone app. Android installation package for this app (“WaterLeakDetectorApp.apk) is provided in the App folder on this repository. MIT App Inventor 2 source code for this app is also provided: “WaterLeakDetectorApp.aia”. See section 6.2, below, for further installation information.

The smartphone app provides real-time status from the WLD:

- Current temperature at the WLD (to one decimal place).
- Current RH at the WLD (to one decimal place).
- Status of the three alarms: low temperature alarm, high temperature alarm, leak alarm.

In addition, the app displays the current low and high temperature alarm thresholds and provides a button to set new alarm thresholds.

5.1. SMS Alarms.

Three alarm conditions are processed by the WLD:

- Temperature is at or below the low temperature alarm limit.
- Temperature is at or above the high temperature alarm limit.
- A water leak is detected at either of the water leak sensors.

When an alarm condition is met, an SMS text message will be sent to the user’s mobile phone. An overview of this process is contained in the document “Water_Leak_Detector_Concept_V2” in the Documentation folder of this repository.

A low or high temperature alarm message will contain the nature of the alarm as well as the actual temperature measured when the alarm was triggered. A water leak alarm will contain the alarm message.

Once an alarm message is sent, that alarm message will be held off for one day. If the alarm condition persists for a full day, a new alarm message will be sent, followed by another one-day alarm holdoff. In other words, a persistent alarm condition will generate an SMS textual alarm message once per day. However, each type of alarm is re-activated after the condition that causes the alarm is cleared; hence, a new alarm message will be sent immediately upon the alarm condition being met once more.

Note that alarm messages are immediate and independent of any smartphone app. In fact, alarm messages may be received by any mobile device that supports SMS text messaging. Alarm text messages are initiated by the Google Apps Script (part of this project) sending an email to the SMS gateway of the user’s cellular phone carrier. SMS text alarms may be sent to more than one device – simply add more lines of “EMAILapp.sendMail()” of the script. More information about this process can be found in the document:

https://github.com/TeamPracticalProjects/Connectivity_Tools_with_Particle_Devices/blob/main/TPP_Connectivity_Tools_Technical_Description.pdf

5.2. Smart Phone App.

This project includes a smartphone App. The App is currently available only for Android phones and tablets. The App allows the user to monitor and status the WLD in real-time. Figure 5-1 is an annotated screenshot of the App. See section 6 of this document for App installation instructions.

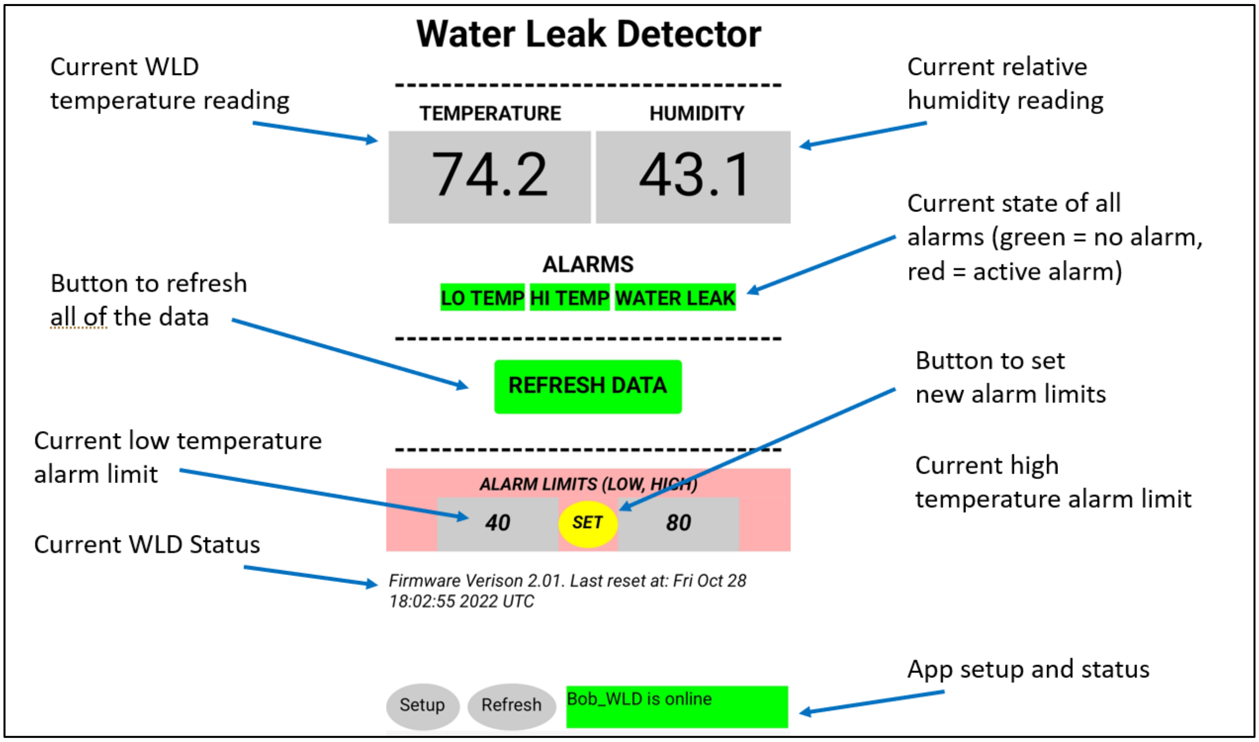

Figure 5-1. Smartphone App Screen.

The following is an explanation of the features and functions of the App, from top to bottom:

- TEMPERATURE. Displays the current temperature as measured by the WLD in degrees F (to one decimal place). The reading is updated each time that the REFRESH DATA button is tapped.

- HUMIDITY. Displays the current relative humidity as measured by the WLD in % RH (to one decimal place). The reading is updated each tome that the REFRESH DATA button is tapped.

- ALARMS. The current status of three alarms is displayed in the following order: LOW TEMP(erature), HIGH TEMP(erature), WATER LEAK. The alarm text is always present, regardless of the alarm status. The alarm background color is: Green for no alarm, Red for a current alarm. The alarm status' are updated each time that the REFRESH DATA button is tapped.

- REFRESH DATA. This button is tapped to refresh all of the data. This includes TEMPERAURE, HUMIDITY, and all three alarms. The current alarm limits (below) are also refreshed to show the latest information from the WLD.

- ALARM LIMITS. There are three parts to the ALARM LIMITS field. The leftmost part is the current low temperature alarm limit, in degrees F, as an integer. The rightmost part is the current high temperature alarm limit, in degrees F, as an integer. These fields are *editable*, so the user can change the alarm limits in the WLD by changing these values and tapping the SET button in the middle of this field. After tapping SET, the contents of the low temperature alarm limit and the high temperature alarm limit fields are stored into non-volatile memory inside the WLD and then the contents of both of these fields will be updated to show the current status of the WLD alarm limits.

At the very bottom of the App screen are SETUP and REFRESH buttons and a STATUS field. These elements are part of our standard "Particle App Template", see:

https://github.com/TeamPracticalProjects/Particle_App_Template

Briefly, tapping the SETUP button takes the user to the App's setup screen, where the user can log into their Particle account and select the Particle device that the App communicates with. The REFRESH button re-connects the App with the currently selected Particle device and also refreshes all of the App screen data (i.e. similar to REFRESH DATA). The status field shows which device the App is connected to and the connection status (Green denotes "connected OK", Red denotes "not connected", and Yellow denotes "awaiting response"). This field must be green in order for the App to work.

A note about the field and button colors and contents: After the REFRESH DATA or the SET button is pressed, the button color turns gray and stays gray until the operation is complete. After the button operation is complete, the button returns to its normal color (shown in figure 5-1). Temperature, humidity and temperature alarm limit fields turn blank (no displayed text) when the data is being refreshed. Only the newest data is displayed.

6. Software/Firmware Installation.

6.1. Firmware Installation and Servo Calibration.

The *Firmware* folder in this repository contains the Particle firmware that you need to flash to your WLD Photon. There are two subfolders:

- *ServoCal/src*: contains one file: "ServoCal.ino". This is the source code that you need to flash to the WLD Photon in order to calibrate the servo that is used as a temperature and humidity meter.

- *WaterLeakDetector*: contains all source code and library files that are needed to flash the WLD operating firmware to the WLD Photon.

You must first build and test the WLD hardware before you flash firmware to the Photon. We strongly recommend that you thoroughly familiarize yourself with the WLD overall by reading this manual and the document "Water_Leak_Detector.pdf" before proceeding to build anything. Next, you will need to build the WLD printed circuit board, per the instructions in the document "PCB Assembly Instructions.pdf". Finally, you will need to build one of the enclosures: either (1) "Enclosure Build Instructions Plastic Case.pdf", or (2) "Enclosure Build Instructions 2D Box.pdf", or (3) "Enclosure Build Instructions 3D Case.pdf". The choice of which enclosure to use for your WLD is up to you. You will also need to claim a new Photon to your Particle account; see Particle's documentation at:

https://docs.particle.io/quickstart/photon/

You should also familiarize yourself with the Particle Workbench IDE:

https://docs.particle.io/getting-started/developer-tools/workbench/

Once you have assembled all of the hardware, you will need to calibrate the servo meter. In order to do so, you need to flash the firmware "ServoCal.ino" to the WLD Photon and follow the instructions in the document "Servo_Meter_Calibration.pdf" in the *Documentation* folder in this repository. NOTE: you may use Particle's WEB IDE or the Particle Workbench to flash this code to your WLD Photon.

After calibrating the servo meter, you will need to re-flash the WLD Photon with the WLD firmware. This firmware is found in the *WaterLeakDetector* subfolder in the *Firmware* folder in this repository. NOTE: There are multiple subfolders and firmware source code files in the *WaterLeakDetector* subfolder. We strongly recommend that you use the Particle Workbench to compile and flash this firmware to your WLD Photon. You must compile this firmware using the Particle Workbench and then you can use the Workbench to flash the compiled code to your

WLD Photon. We suggest compiling against Particle OS version 3.3 released. As of this writing, this is the latest version of the Particle OS that supports the Photon. You may use either the cloud compile and cloud flash options in the Workbench to do this, or else the local compile and local flash, if you prefer. Cloud compiling and cloud flashing is probably the fastest and easiest to use, while local compiling and flashing provides more information about the build progress.

6.2. App Installation.

You will need to install the smartphone app on an Android mobile phone or tablet if you want to status your WLD at any time. *You do not need to install the app in order to get SMS text alarms from your WLD*.

The smartphone app files are in the *App* folder in this repository. Two files are included:

- "*WaterLeakDetectorApp.aia*": This is the source code file, written in MIT App Inventor 2 (http://ai2.appinventor.mit.edu). You do not need this source code in order to install and use the App. The source code is provided for those users who wish to examine the code and perhaps modify the app to meet their unique needs.

- "*WaterLeakDetectorApp.apk*": This is the installation package. You need to load this package onto your Android smartphone or tablet in order to install the App.

In order to install the app on your Android device, you need to get the file "*WaterLeakDetector.apk*" loaded on to this device. There are several ways to do this:

- Download the .apk file directly to your Android device from this repository. Make note of where this file is downloaded to in the Android device's folder hierarchy.

- Sideload the .apk file: Connect a USB cable from your development computer (where you downloaded this file) to your Android device. The Android device should appear as a memory device on your development computer. You can then drag and drop the .apk file somewhere on the Android device, e.g. in the *Downloads* folder. It doesn't matter where you sideload this file to; just make note of where it is.

- Email the apk file from your development computer to yourself. Then read the email on your Android device and store the attachment (the .apk file) on the Android device, making note of where the file is stored.

However you choose to get the .apk file on to your Android device, you next need to use the Android device's file explorer to locate the .apk file. Then tap on the file to install it. The Android OS will install the file on the Android device. NOTE: The installation process will warn you about the Android resources that the App requires – you need to accept this to install the App. Sometimes the Android installer will issue you a warning that the App is not trusted. This simply means that you didn't download it from the Google Play Store. Go ahead and tap on the option to install it anyway.

Once the App is installed on your Android device, you may tap to open it. You can identify the App by the little leaky pipe icon that it presents on the Android device's App selector screen.

The first time that you open the App after installation, the App's home screen will show a red ERROR! Message at the bottom right of the screen. This means that a Particle device has not been selected for the App to communicate with. Tap on the "Setup" button at the lower left of the home screen to get to the setup page of the App. On the Setup page, you need to type in your Particle login and Password where indicated and tap "Request Device List". The gray button below that said "No Devices" will change to "Select device". Tap on "Select device" to get a pick list of all of the Particle devices in your Particle account. Then tap on the device used for your WLD and the tap "Exit Setup" at the bottom of the setup screen to return to the App home screen. The status field at the bottom right of the home screen should turn yellow briefly and then turn green, indicating that the WLD Photon is on-line and the App can be used. If this field turns red, the selected Particle device is not on-line to the Particle cloud. Make sure that the Photon in your WLD is powered up and that the multicolor LED is breathing cyan, indicating that it is connected to the Particle cloud.

6.3. Script and Webhook Installation.

The Google Apps Script should be created and deployed first. The Script has a URL which needs to be copied and pasted into the webhook.

6.3.1. Script Installation.

Open an "Incognito" window on a web browser. Then log in to your Google account and open the Google "drive" app. Select "New" in the upper left hand corner and then find and select "Google Apps Script"; see figure 6-1. If you don't see Google Apps Script in your Google account, you can add it by clicking on "+ Connect more apps".

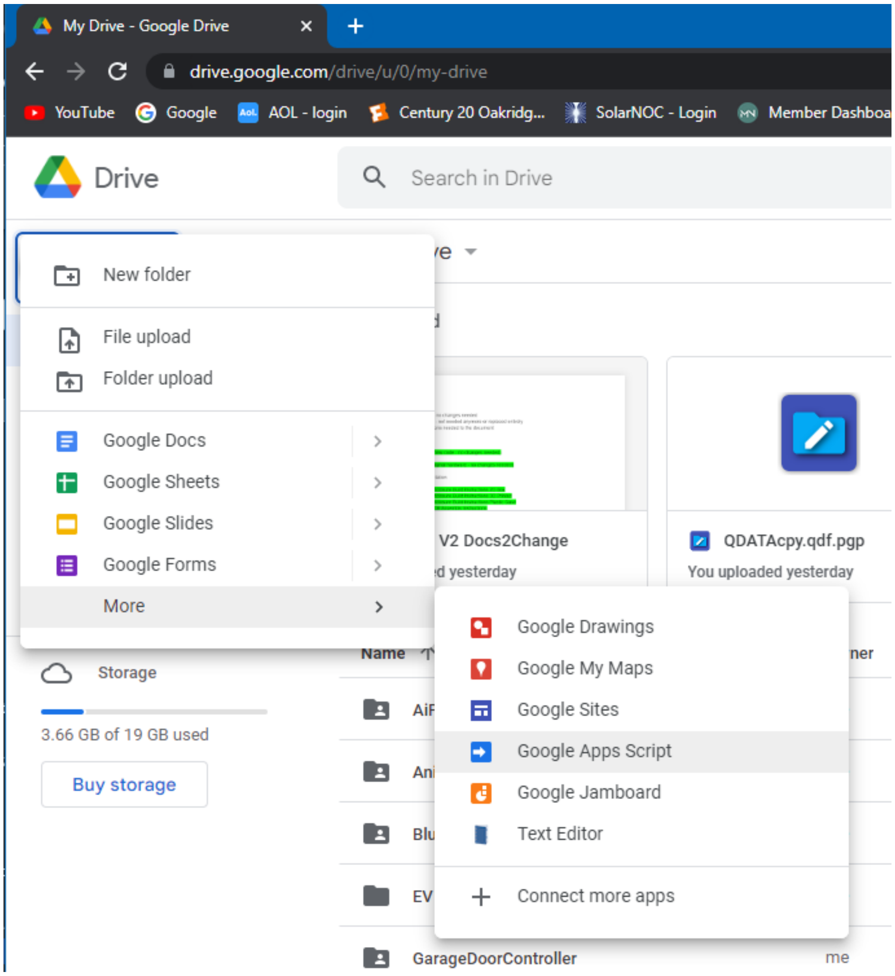

Figure 6-1. Open New Google Apps Script.

The Google Script editor will open with a default function template; see figure 6-2.

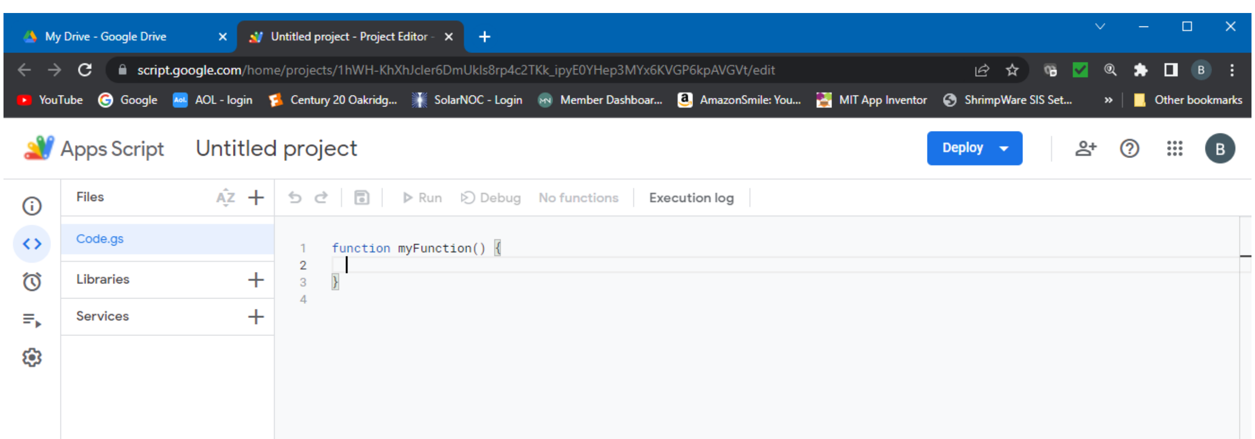

Figure 6-2. Blank Script Editor Screen.

Delete the function template on the script editor screen so that nothing at all is in the editor window. Also, click on Untitled project at the top of the editor screen and give you project a name, such as “WLDAlarmScript”.

In the GitHub repository for this project, open the “Scripts” folder and open “Alarm_script.txt”. Select every line in this document and copy it to your computer’s clipboard. Make sure that you copy all lines (1-50). Then go back to the scripts editor and paste the script code in the editor window. Locate the line (line 10):

```
const targetBob = "<email for sms gateway goes here>";
```

Replace `<email for sms gateway goes here>` with the email address of your sms gateway account with your cellular carrier. The email address begins with your 10 digit cell phone number, @ sign, and then the address of the gateway. Gateway addresses for all cellular carriers in the United States can be found here:

https://jagerpro.com/help-center/sms-and-mms-gateway-list/

For example, mobile carrier “AT&T former Cingular” would be formatted:

111-222-3456@att.txt.net

If you are not in the US Pacific time zone, go to the script line 47 (begins with “var formattedDate = “) and replace the parameter “America/Los_Angeles” with the time zone that you want alarms reported in. Now save your script using the little disk icon near the top of the editor screen.

Check that the script is complete and correctly edited. Now click on the “Deploy” button near the top right of the screen. You will get a window similar to figure 6-3, below:

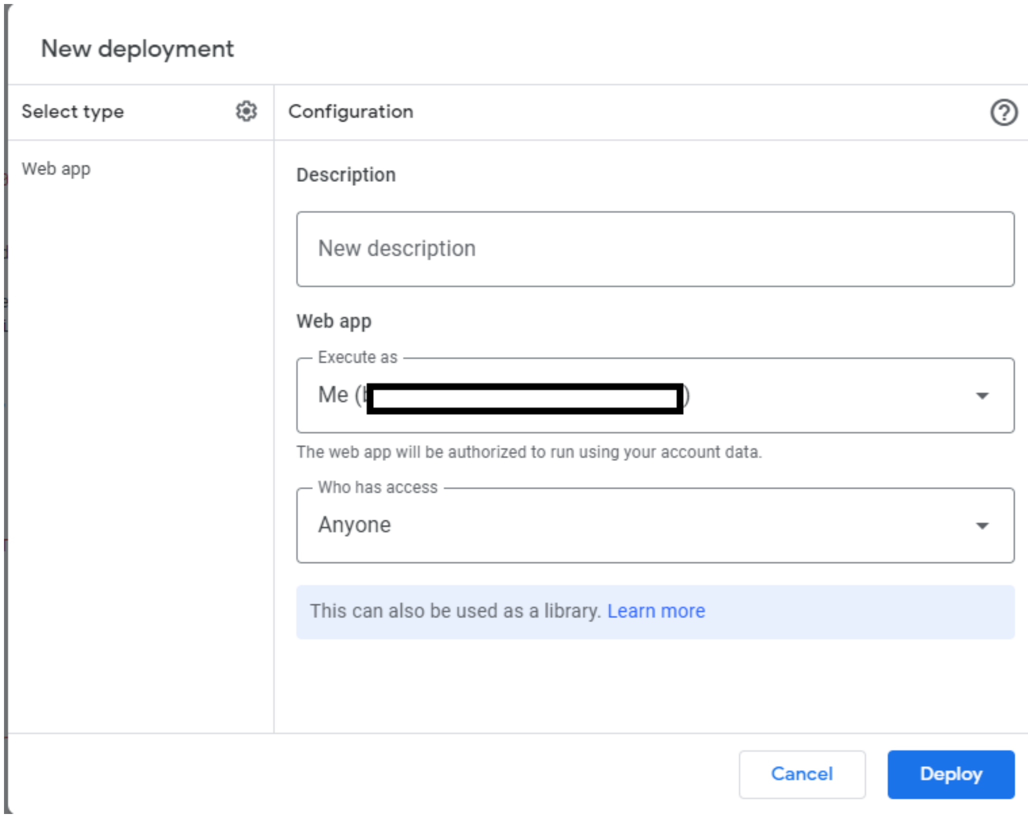

Figure 6-3. Script deployment.

Make sure to deploy the script as a “Web app”. Put something in the Description field, e.g. “Initial release” or “Version 1”. In “Execute as”, select the “Me” option. In “Who has access”, select “Anyone”. Double check your settings and click “Deploy”. You will see a popup with the App ID and the App URL. Click “Copy link” for the App URL to copy this link to your computer’s clipboard. If you accidently copy something else over this when you need it for the webhook, you can return to the Script editor “Manage deployments” and find and select it again.

6.3.2. Webhook Installation.

You create a new webhook by going to “console.particle.io” in your web browser and logging in to your Particle account. Select “Integrations” (the little star like icon in the left hand panel of the screen). Click on “NEW INTEGRATION” and click on “Webhook”. You will be presented with a form that has a number of fields to fill out. Fill them out as follows:

- Event Name: enter **WLDAlarm** (exactly as shown here – case sensitive).
- URL: paste in the script URL that you saved to your computer’s clipboard in the previous section.
- Request Type: POST

- Request Format: Web Form
- Device: Any
- Status: Enabled

You do not need to access the Advanced Settings. Just click on Create Webhook. You should see the webhook as in figure 6-4, below.

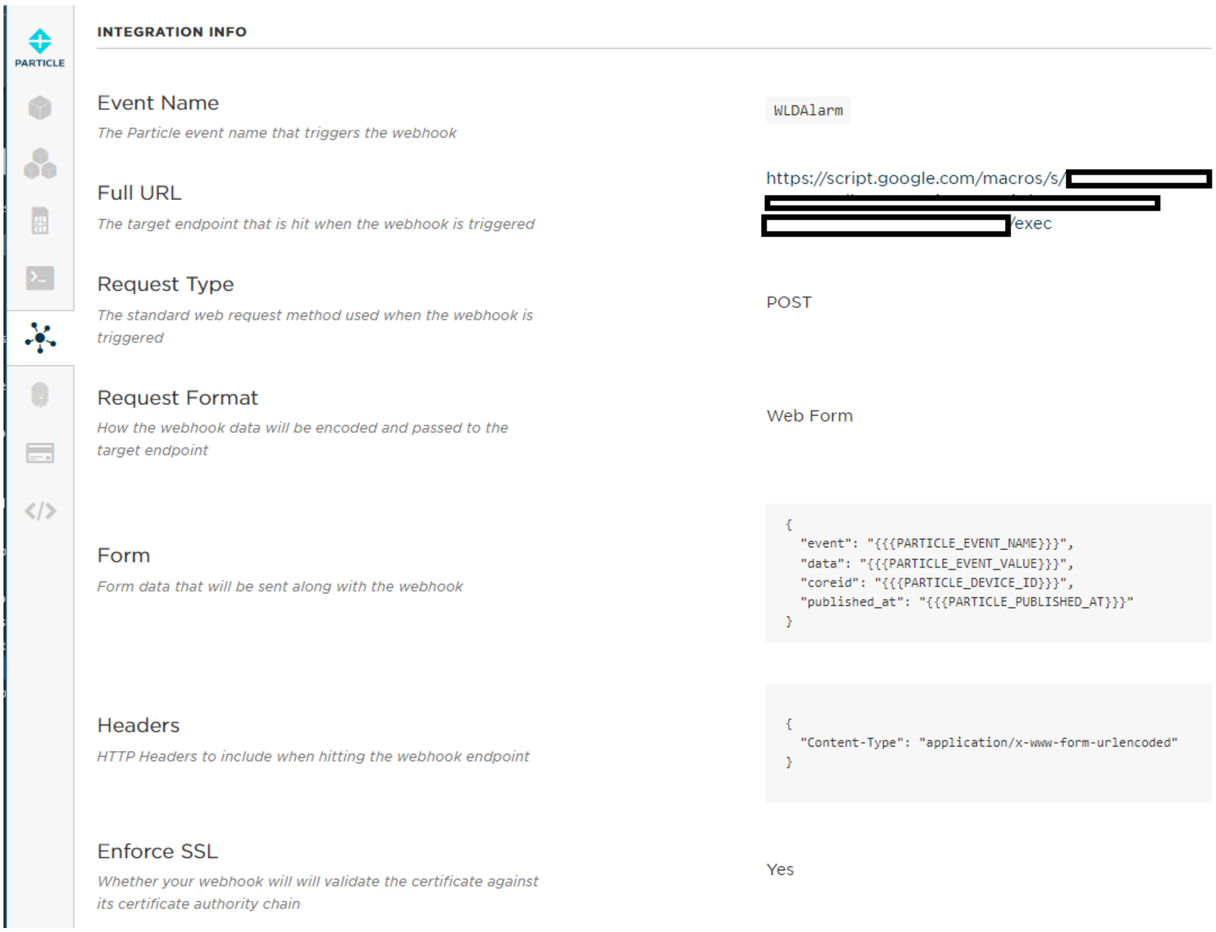

Figure 6-4. Webhook Setup.

The webhook is ready to go.

7. Testing Using the Particle Console.

The Particle console may be used to test out the hardware/firmware, and also to generate a test alarm to test that the webhook and Google Apps Script are working properly. Go to “console.particle.io” in your web browser and log in to your Particle account. You will see a list of the Particle devices in your account. Make sure that the WLD device is powered up and that the multicolored LED is breathing Cyan. Select the Photon device used in your Water Leak Detector in the Console. You will see the Console screen for this device. The Console screen has many helpful items, including device status, device health, etc. When using the Console for testing, you will use two sections of the Console screen:

- Events (on the right side of the screen).
- Functions and Variables (on the left side of the screen).

Figure 7-1 focuses in on the Functions and Variables. The WLD firmware has two cloud functions:

- “*SetTempAlarmLimits*”: used to set the low and high temperature alarm limits. Enter the data as “lowTempAlarmLimit,highTempAlarmLimit”; e.g. 40,85 as shown in figure 7-1. Clicking on “CALL” will set these limits (low temp alarm limit of 40 degrees and high temp alarm limit as 85 degrees) into the Photon’s non-volatile memory.

- “*Send a test alarm*”: clicking on “CALL” with no argument data will send a test alarm message to your configured SMS gateway and, from there, to your cell phone. You will also see the published alarm event in the Event window on the right side of the Console screen, as in figure 7-2. This shows that the Photon firmware sent out the test alarm event to the Particle cloud. If you don’t get an SMS text message within a few seconds, this means that either your webhook or your script was improperly set up. You can test that your mobile carrier SMS text gateway is working by sending an email to the gateway address that you configured in your Google Apps Script on the line:

    const targetBob = ……

Below the Functions on the Console screen are the Cloud Variables:

- *Info (string):* Clicking on GET will return a string of information about the firmware – the version number and the time of the last reset (in UTC – not timezone adjusted).

- *Temperature (string):* Clicking on GET will return the currently measured temperature, in degrees F, rounded to one decimal digit.

- *Humidity (string):* Clicking on GET will return the currently measured relative humidity, in %RH, rounded to one decimal digit.

- *Alarms (string):* Clicking on GET will return a comma delimited string indicating the state of three alarms, in the order of: low temperature alarm, high temperature alarm, water leak alarm. A “0” indicates no alarm, a “1” indicates a current alarm condition. Note: this is the current status of the alarms which may be different from the alarm status when an SMS text was sent.

- *LowTempAlarmLimit (string):* Clicking on GET will return the low temperature alarm limit that is currently being used by the Photon firmware, rounded to the nearest integer value.

- *HighTempAlarmLimit (string):* Clicking on GET will return the high temperature alarm limit that is currently being used by the Photon firmware, rounded to the nearest integer value.

Note that you can force a low or high temperature alarm by setting the low temperature alarm limit higher than the current temperature, or setting the high temperature alarm limit lower than the current temperature. You will see the Alarm event publications in the Console Event panel, as depicted in figure 7-2.

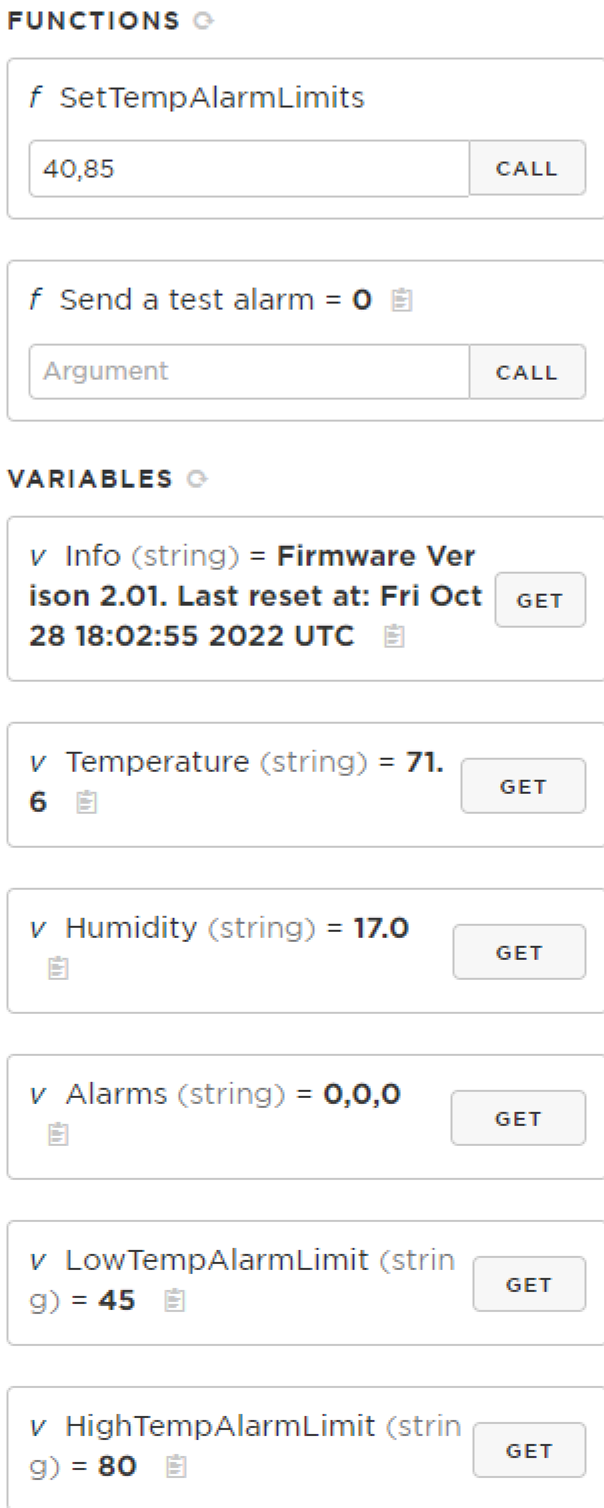

Figure 7-1. Variables and Functions.

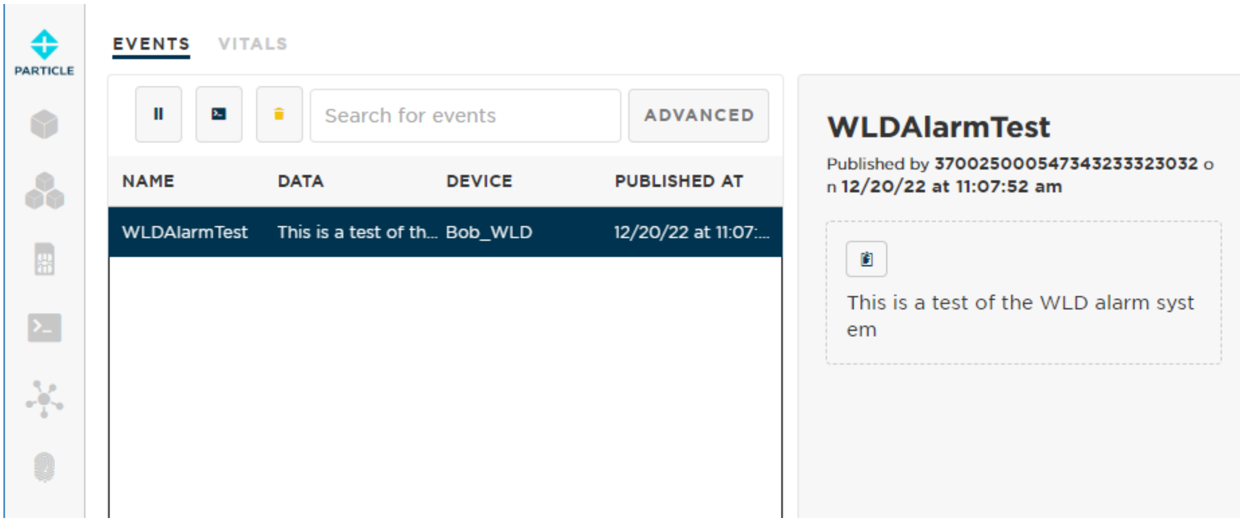

Figure 7-2. Test Alarm.