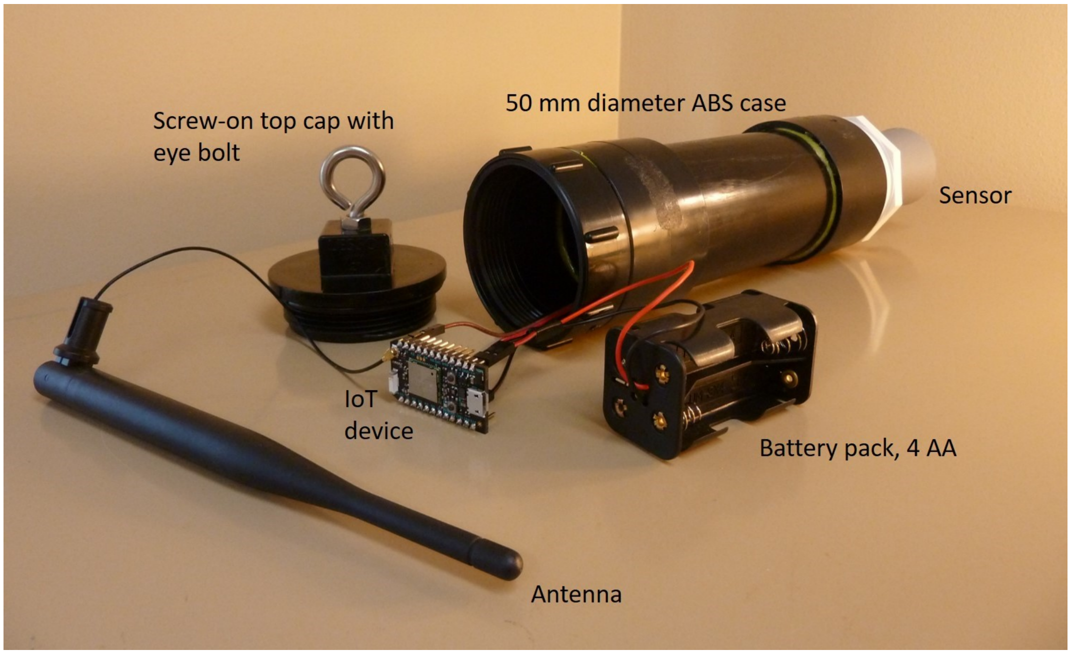
GEOSCIENCE AND MINES BRANCH

Instructions for Building a Real-Time Well Water Level Meter

J. Drage

Open File Report ME 2019-003



NOVA SCOTIA

Energy and Mines

Geological Survey

Halifax, Nova Scotia
November 2019

Instructions for Building a Real-Time Well Water Level Meter

J. Drage

Introduction

This report provides instructions for building a low-cost, real-time water level meter for use in dug wells. The meter is designed to hang inside a dug well, measure the water level once a day, and send the data by WiFi to a webpage for immediate viewing and downloading. The cost for the parts to build the meter is approximately Can$200. The water meter is shown in Figures 1 and 2.

The meter uses an ultrasonic sensor to measure the depth to water in the well. The sensor is attached to an Internet-of-Things (IoT) device that connects to a WiFi network and sends the water level data to a web service to be graphed. The web service used in this project is ThingSpeak.com (ThingSpeak, 2019), which is free to use for non-commercial small projects (fewer than 8,200 messages/day). In order to work, the meter must be located close to a WiFi network. Domestic water wells often meet this condition because they are located close to a house with WiFi. The meter does not include a data logger, rather it sends the water level data via WiFi to ThingSpeak where they are stored in the cloud. Therefore, if there is a WiFi connection problem (e.g. during an Internet outage) the water level data for that day are not transmitted and are permanently lost.

The meter was designed and tested for large-diameter (0.9 m inside diameter) dug wells with shallow water depths (less than 10 m below ground surface). However, it could potentially be used for measuring water levels in other settings, such as environmental monitoring wells, drilled wells, and surface water bodies.

The meter design presented here was modified after a meter that was made for measuring water levels in a domestic water tank and reporting the water level via Twitter (Ousley, 2019). The main differences between the original design and the design presented here are the ability to operate the meter on AA batteries instead of a wired power adaptor, the ability to view the data in a time-series graph instead of a Twitter message, and the use of an ultrasonic sensor that is specifically designed for measuring water levels.

Parts List

Table 1 lists the parts needed to build the meter. The electronic parts can be ordered from online retailers. The plumbing and hardware parts for the meter case can be found at building supply stores. The only parts that have been specifically identified with a manufacturer and model number are the ultrasonic sensor (MaxBotix Inc., 2019) and the IoT device (Particle, 2019) because the instructions and code presented here are unique to these parts. However, any similar sensor and IoT device could potentially be substituted for the ones used here, with

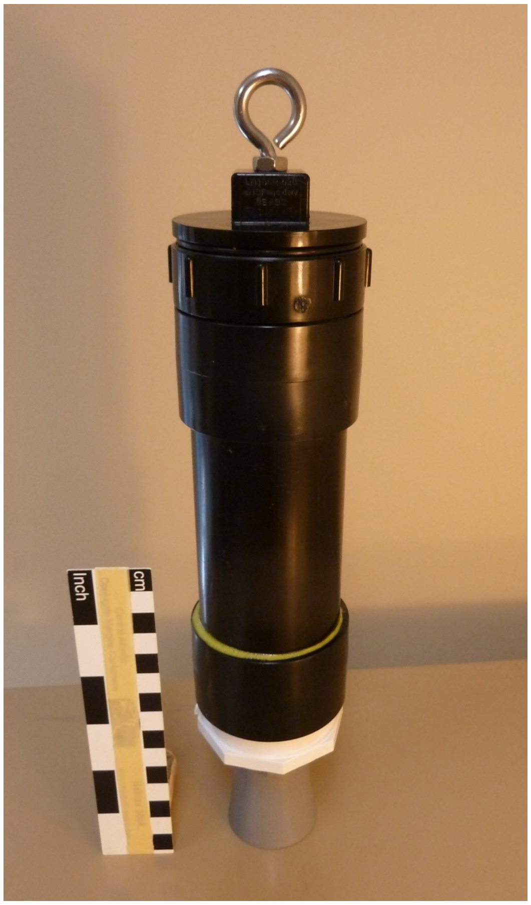


Figure 1. External view of meter. Dimensions: 320 mm long (tip to tip), 50 mm diameter.

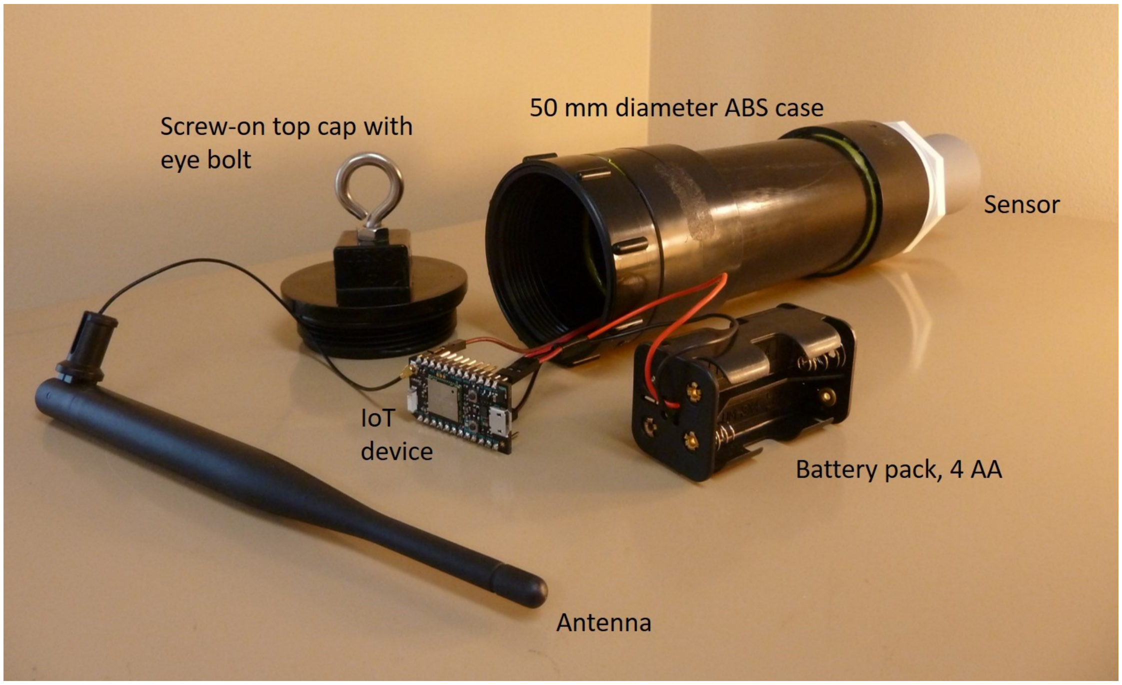


Figure 2. View of meter with top cap removed to show internal parts.

Table 1. Parts list.

Electronic parts	**Cost***
Sensor – MaxBotix MB7389 (5m range) (https://www.maxbotix.com/Ultrasonic_Sensors/MB7389.htm)	$132
IoT device - Particle Photon with headers (https://store.particle.io/collections/wifi)	$30
Antenna – 2.4 GHz, 6dBi, IPEX or u.FL connector, 170 mm long	$7
Battery pack – 4 X AA	$1
Wire – jumper wire with push on connectors (300 mm length)	$2
Batteries – 4 X AA	$5
Plumbing and hardware parts	
Pipe - ABS, 50 mm (2 inch) diameter, 125 mm long	$1
Top cap, ABS, 50 mm (2 inch), threaded with gasket to make a watertight seal	$5
Bottom cap, PVC, 50 mm (2 inch) with ¾ inch female NPT thread to fit sensor	$4
2 Pipe couplers, ABS, 50 mm (2 inch) to connect top and bottom cap to ABS pipe	$2
Eye bolt and 2 nuts, stainless steel (¼ inch) to make hanger on the top cap	$2
Other materials: electrical tape, Teflon tape, solder, silicone, glue for assembling case	-
TOTAL	**$191**

*Costs are in Canadian dollars and do not include tax and shipping.

appropriate assembly and code modifications. The exact choice of wire, antenna, and battery holder is flexible, keeping in mind that the antenna and battery holder must fit inside the meter case. Note that the sensor listed in Table 1 has a 5 m range. If a 10 m range is needed, it can be substituted with MaxBotix sensor model MB7388. For additional information about using a 10 m range sensor model, see the section below entitled "Meter Construction and Operation Tips".

Water Level Meter Construction

Step-by-step instructions for constructing the water level meter are provided below. It is recommended that the builder read through all construction steps and the section below entitled "Meter Construction and Operation Tips" before starting the meter construction process. The IoT device used in this project is a Particle Photon, and therefore in the following sections the terms "IoT device" and "Photon" are used interchangeably.

Step 1 – Assemble the Meter Case

Assemble the meter case as shown in Figures 1 and 2. The total length of the assembled meter, tip to tip including the sensor and eye bolt, is approximately 320 mm. The 50 mm diameter ABS pipe used to make the meter case should be cut to approximately 125 mm in length. This allows sufficient space inside the case to house the IoT device, battery pack, and antenna.

Seal all joints with either silicon or ABS glue to make the case watertight. This is very important, otherwise moisture can get inside the case and destroy the internal components. A small desiccant pack can be placed inside the case to absorb moisture.

Install an eye bolt in the top cap by drilling a hole and inserting the eye bolt and nut. A nut should be used on both the inside and the outside of the case to secure the eye bolt. Silicon the inside of the cap at the bolt hole to make it water tight.

Step 2 – Attach Wires to Sensor

Three wires (Fig. 3a) must be soldered to the sensor in order to attach it to the Photon (GND, V+, and Pin 3). Soldering the wires to the sensor can be challenging because the connection holes on the sensor are small and close together. It is very important that the wires are properly soldered to the sensor so there is a good, strong physical and electrical connection and no solder arcs between adjacent wires. Good lighting and a magnifying lens helps with the soldering process. For those who do not have previous soldering experience, some practice soldering is recommended prior to soldering the wires to the sensor.

After the wires are soldered to the sensor, any excess bare wire that sticks out from the sensor can be trimmed off with wire cutters to approximately 2 mm length. It is recommended that the solder joints be covered with a thick bead of silicon. This gives the connections more strength and reduces the chance of corrosion and electrical problems at the sensor connections if moisture gets into the meter case. Electrical tape can also be wrapped around the three wires at the sensor connection to provide additional protection and strain relief, reducing the chance that the wires will break at the solder joints.

The sensor wires can have push-on-type connectors (Fig. 3b) at the one end to attach to the Photon. Using push-on connectors makes it easier to assemble and disassemble the meter. The sensor wires should be at least 270 mm long so they can extend the entire length of the meter case. This length will allow the Photon to be connected from the top end of the case with the sensor in place at the bottom end of the case. Note that this recommended wire length assumes that the ABS pipe used to make the meter case is cut to a length of 125 mm. Confirm in advance of cutting and soldering the wires to the sensor that a wire length of 270 mm is sufficient to extend beyond the top of the meter case so that the

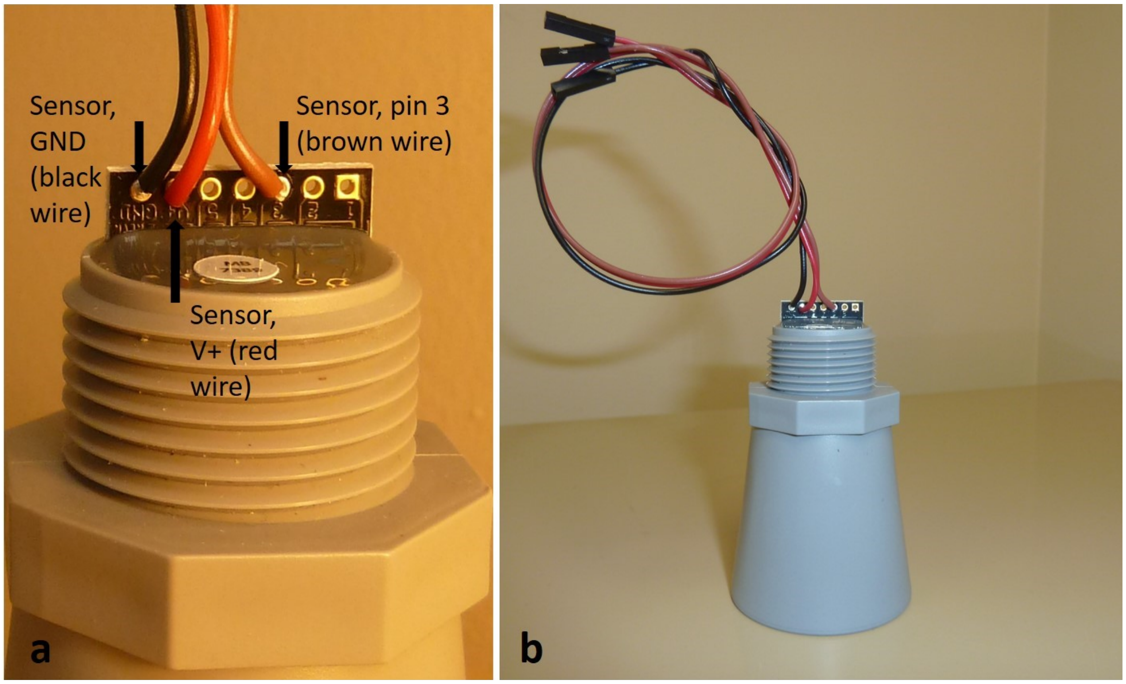


Figure 3. Ultrasonic sensor: a) close-up view showing wire connections; b) full view of sensor showing wires with push-on-type connectors at one end for attaching to the IoT device.

Photon can be connected after the case has been assembled and the sensor is permanently attached to the case.

The sensor can now be attached to the meter case. It should be screwed tightly into the bottom cap, using Teflon tape to ensure a watertight seal.

Step 3 – Attach Sensor, Battery Pack, and Antenna to IoT Device

Attach the sensor, battery pack, and antenna to the Photon (Fig. 4), and insert all parts into the meter case. Table 2 provides a list of the pin connections indicated in Figure 4. The sensor and battery pack wires can be attached by soldering directly to the Photon or with push-on-type connectors that attach to the header pins on the underside of the Photon (Fig. 2). Using push-on connectors makes it easier to disassemble the meter or replace the Photon if it fails. The antenna connection on the Photon requires a u.FL type connector (Fig. 4) and needs to be very firmly pushed onto the Photon to make the connection. Do not install the batteries into the battery pack until the meter is ready to be tested or installed in a well. There is no on/off switch included in this design, so the meter is turned on and off by installing and removing the batteries.

Step 4 – Software Setup

Four main steps are needed to set up the software for the meter:

1. Create a Particle account that will provide an online interface with the Photon. To do this, download the Particle mobile app to a smartphone (https://docs.particle.io/quickstart/photon/). After installing the app, create a Particle account and follow the online instructions to add the Photon to the account. Note that any additional Photons can be added to the same account without the need to download the Particle app and create an account again.

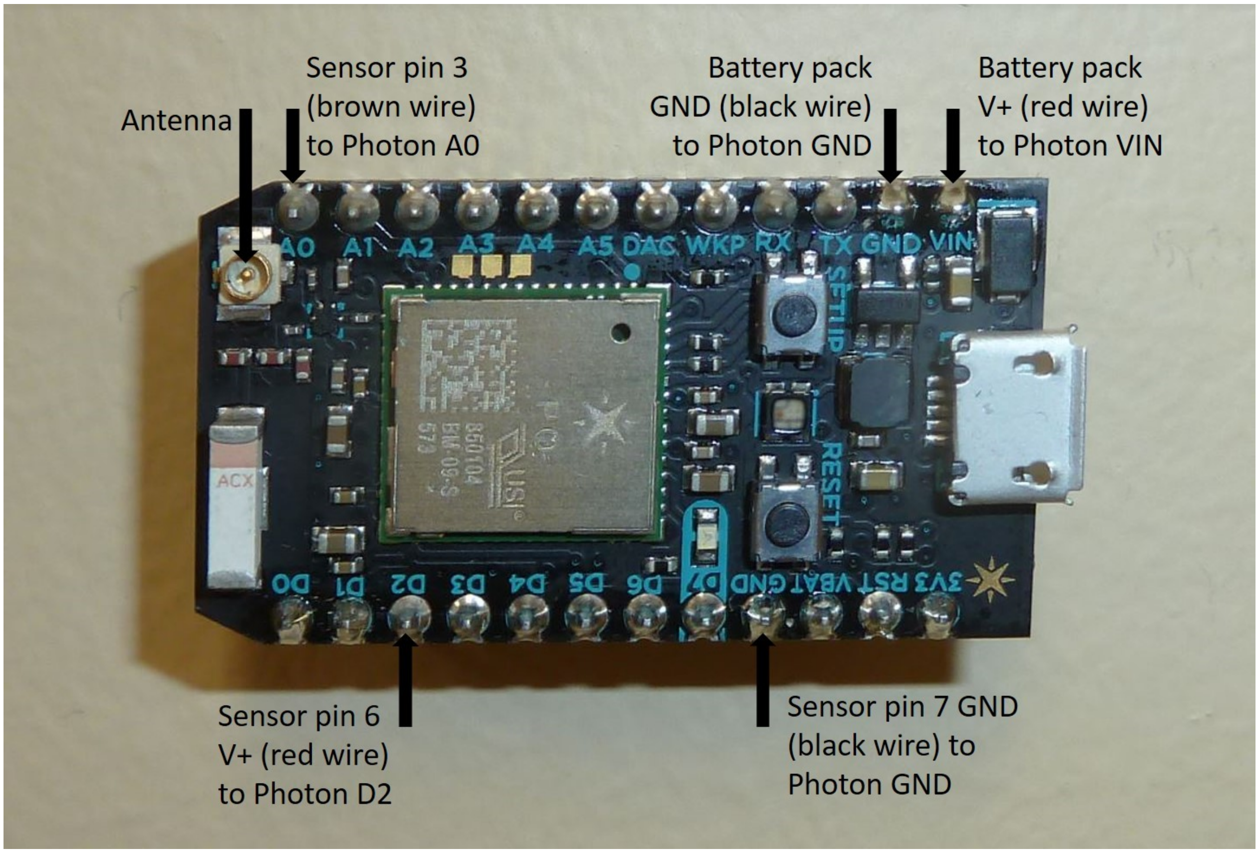

Antenna
Sensor pin 3 (brown wire) to Photon A0
Battery pack GND (black wire) to Photon GND
Battery pack V+ (red wire) to Photon VIN
A0 A1 A2 A3 A4 A5 DAC WKP RX TX GND VIN
SETUP
RESET
D0 D1 D2 D3 D4 D5 D6 D7 GND VBAT RST 3V3
Sensor pin 6 V+ (red wire) to Photon D2
Sensor pin 7 GND (black wire) to Photon GND

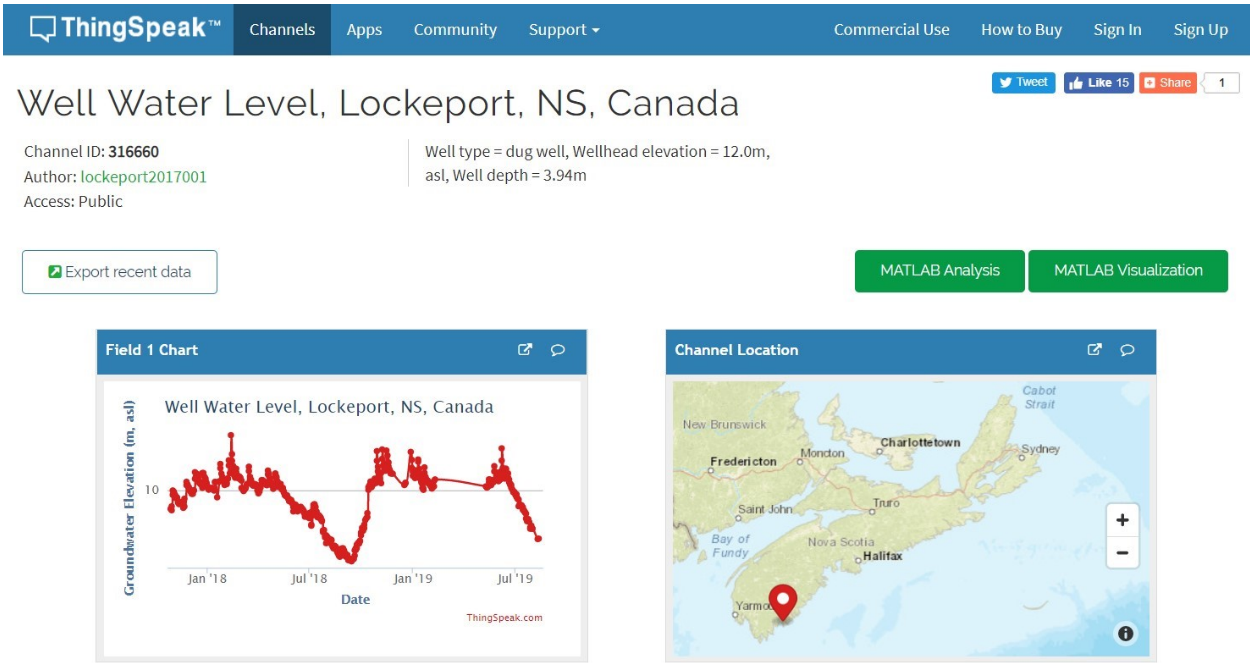
Figure 4. Pin connections on the IoT device (Particle Photon).

Table 2. List of pin connections on the IoT device (Particle Photon).

IoT Device Pin Name	Attachments
A0	Sensor pin 3, Analog voltage output (brown wire)
D2	Sensor pin 6, V+ (red wire)
GND	Sensor pin 7, GND (black wire)
VIN	Battery pack, V+ (red wire)
GND	Battery pack, GND (black wire)
u.FL	Antenna

2. Create a ThingSpeak account (https://thingspeak.com/users/sign_up) and set up a new channel to display the water level data. An example of a ThingSpeak webpage for a water meter is shown in Figure 5. Instructions for setting up a ThingSpeak channel are provided at https://docs.particle.io/tutorials/device-cloud/webhooks/. Note that additional channels for other Photons can be added to the same account without the need to create another ThingSpeak account.

3. A “webhook” is required in order to pass data from the Photon to the ThingSpeak channel. Instructions for setting up a webhook are provided at https://docs.particle.io/tutorials/device-cloud/webhooks/. If more than one water meter is being built, a new webhook with a unique name must be created for each additional Photon.



Well Water Level, Lockeport, NS, Canada

Channel ID: 316660
Author: lockeport2017001
Access: Public

Well type = dug well, Wellhead elevation = 12.0m, asl, Well depth = 3.94m

Figure 5. ThingSpeak webpage for a dug well located in Lockeport, Nova Scotia.

4. Finally, the webhook that was created in the above step must be inserted into the code in the Appendix. On a computer, go to the Particle webpage (https://login.particle.io/login?redirect=https://console.particle.io/devices), login to the Particle account, and navigate to the Particle app interface. Copy the code in the Appendix and use it to create a new app. Modify the code by inserting the name of the webhook created above into line 86 of the code. Line numbers are not shown in the Appendix, but they are visible within the Particle app interface. The code can now be verified, saved, and installed onto the Photon. Note that the code is stored in and installed onto the Photon from the cloud. This code will be used to operate the water meter when it is in the water well. During the field installation, some changes will need to be made to the code to set the reporting frequency to once a day and add information about the water well, as described below in the section entitled "Installing the Meter in a Water Well".

Step 5 – Test the Meter

The meter construction and software setup are now complete. At this point it is recommended that the meter be tested. Two tests should be completed. The first test is used to confirm that the meter can measure water levels correctly and send the data to ThingSpeak. The second test is used to confirm that the power consumption of the Photon is within the expected range. This second test is useful because the batteries will fail sooner than expected if the Photon is using too much power.

For testing purposes, the code in the Appendix is already set to measure and report water levels every two minutes. This is a practical time period to wait between measurements while the meter is being tested. If a different measurement frequency is desired, go to lines 63, 73, 78 and 91 of the code in the Appendix and change the number 120 to the desired measurement frequency. The measurement frequency is entered in seconds (i.e. 120 seconds equals two minutes).

The first test can be done in the office by hanging the meter above the floor, turning it on, and checking that the ThingSpeak channel accurately reports the distance between the sensor and the floor. In this testing scenario the ultrasonic pulse reflects off the floor, which is used to simulate the water surface in the well.

For the second test, the electrical current between the battery pack and the Photon should be measured to confirm that it matches the specifications in the Photon datasheet (Particle, 2019). Experience has shown that this test helps identify defective IoT devices before they are deployed in the field. Measure the current by placing a current meter between the positive V+ wire (red wire) on the battery pack and the VIN pin on the Photon. The current should be measured in both operating mode and deep sleep mode by running the code in the Appendix, which will turn the Photon on in operating mode (as indicated by the LED on the Photon turning a cyan colour) for approximately 20 seconds and then automatically put the Photon into deep sleep mode for two minutes (as indicated by the LED on the Photon turning off). The operating current between the battery pack and Photon should be between 80 and 100 mA, and the deep sleep current should be between 80 and 100 µA. If the current is higher than these values, the Photon should be replaced.

Meter Construction and Operation Tips

It is very important that the meter case is water tight. The IoT device can be destroyed if any moisture enters the case. As such, it is important to seal any holes and ABS pipe joints with silicon or glue. If an external antenna is added to the meter, it is recommended that the antenna cable be inserted through the bottom of the case and thoroughly sealed with silicon to prevent moisture entry (Fig. 6).

The MaxBotix sensor listed in Table 1 (model MB7389) has a 5 m range. If a 10 m range is needed, use the MaxBotix sensor model MB7388. The software code presented in these instructions applies to the 5 m range model. When using the 10 m range model, please adjust the code according to the instructions for Pin 3 (Analog Voltage Output), as described in the MaxBotix datasheet (MaxBotix Inc., 2019).

The field installation instructions provided in the section below describe how to set the code to measure water levels once a day, but a different measurement frequency can be used if desired. To change the measurement frequency, go to lines 63, 73, 78 and 91 of the code in the Appendix and enter the desired measurement frequency in seconds (i.e. 120 seconds equals two minutes and 86400 seconds equals one day). Changing the measurement frequency will affect the battery life of the meter. The battery life is expected to be approximately 1 to 2 years with a water level measurement frequency of once a day. At the time of writing, the longest observed battery life in the field has been 19 months. Using a higher measurement frequency will reduce the battery life.

The IoT device used in this project includes 18 input/output options. In the design presented here, only two of these input/outputs are used to connect to the water level sensor. Therefore, several inputs are potentially available for other sensors (e.g. water temperature, electrical conductivity).

Installing the Meter in a Water Well

Before Going to the Field

Check the local regulatory requirements regarding applicable qualifications and procedures for opening a water well (such as well disinfection requirements after removing a well cap) and take precautions not to introduce contamination into the well while installing the well meter.

Prior to going into the field, and while the Photon is still turned on and connected to a WiFi network, install the Tinker app from the Particle interface onto the Photon. The Tinker app will keep the Photon awake so a connection to the well owner's WiFi network can be made. Note that the code in the Appendix cannot be used to make the initial WiFi connection because, as an energy-saving feature, it is programmed to shutoff the Photon if no WiFi signal is found within 25 seconds. This limit of 25 seconds in not enough time to complete the initial WiFi configuration in the field. After the WiFi is configured on the Photon, the operating code in the Appendix can be re-installed.

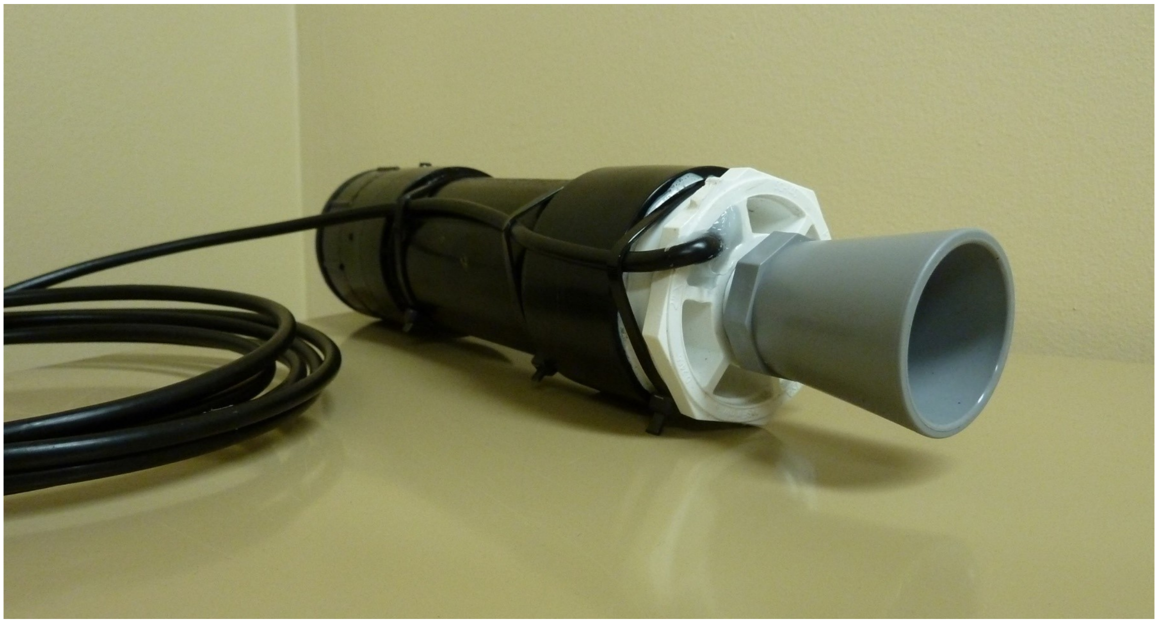


Figure 6. External antenna cable inserted through case bottom and sealed with silicon.

To install the Tinker app, use a computer to go to the Particle webpage (https://login.particle.io/login?redirect=https://console.particle.io/devices), login to the Particle account, and navigate to the Particle app interface (Particle Web IDE). The Tinker app is available at the bottom of the webpage under example apps. Select the Tinker app and install it onto the Photon. The Tinker app can also be installed in the field without using the Internet by connecting the Photon directly to a computer with a USB connection. This is done by placing the Photon in DFU mode and using the Particle CLI program to install the Tinker app over the USB connection. For instructions, please see the section on “Firmware Reset” at https://docs.particle.io/tutorials/device-os/led/photon/.

Field Setup Procedure

Upon arrival at the well, check that a smartphone or computer can successfully connect to the well owner’s WiFi network while standing at the wellhead. Please see the section below entitled “Field Installation Tips” if the smartphone is unable to connect to the WiFi network at the well.

Open the water well and use a manual water level tape (or common carpenter’s tape measure) to measure the depth to water from the top of the well casing. This will be used to calculate an offset value that will be added to the code so it will correctly report the depth to the water. Install a hanger inside the well for the water meter to hang from.

Turn the water meter on by installing batteries in the battery pack. The Photon should now be running the Tinker app, as discussed in the section above. Use the Particle mobile app on a smartphone to add a new device to the Particle account using the well owner’s WiFi network name and password. Once the Photon is successfully connected to the WiFi network, insert the Photon into the meter case and screw the top cap tightly onto the case to make a watertight seal.

Hang the meter in the well (Fig. 7) and close the well cover. Next, use the Particle App interface on a laptop computer to install the modified code from the Appendix onto the Photon. The modified code that should be used is the version that was created in Step 4 above (Software Setup) with the webhook name inserted in line 86. This code has a two-minute reporting frequency, which is temporarily used to complete the meter setup and will be set to a frequency of once a day at the end of the setup process.

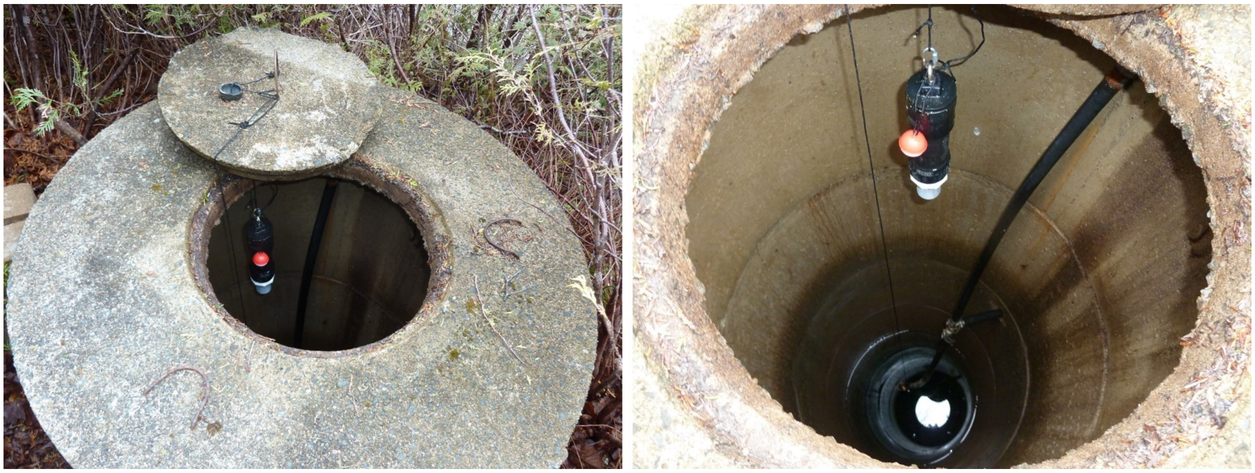


Figure 7. Water level meter installed in a dug well.

Confirm that the meter is now successfully connecting to the Internet while inside the water well and reporting the water level to the ThingSpeak channel every two minutes.

Use the water level reported by the meter on ThingSpeak to calculate the meter hang-down or offset distance and enter this value into line 26 of the modified code in the Appendix for the variable named “HangDown”. This value is used to adjust the meter reading so it reports the distance from the datum to the water level. To calculate the HangDown value, subtract the water level reported by the meter from the water level measured manually earlier. For example, if the water level measured by water level tape is 3.0 m below the top of the casing and the meter reports the distance to water as 2.5 m, then the HangDown value is 0.5 m (i.e. 3.0 m - 2.5 m = 0.5 m).

Enter the elevation of the datum (i.e. top of casing elevation) with respect to mean sea level in line 27 of the modified code for the variable named “Datum”. This is used to convert the depth to water measurement to a groundwater level elevation. Save the code with the new HangDown and Datum values.

Now change the monitoring frequency in the code to once a day by changing the number 120 to 86400 in lines 63, 73, 78, and 91 of the code. Verify and save the code. The final code is now ready to be installed onto the Photon so that it will read the water level once a day and report the groundwater level elevation to ThingSpeak. At this point the Photon is still turning on and measuring water levels every two minutes. The next time the Photon turns on and comes online, install the newly updated code onto the Photon. The installation process is now complete.

Field Installation Tips

A common problem in the field is a weak WiFi signal that prevents the meter from connecting to the Internet. The WiFi signal strength (RSSI) should be checked at the wellhead on a smartphone or computer. The Photon also reports WiFi signal strength as part of its online diagnostic tools. Signal strengths are presented as negative values and the closer to zero, the stronger the signal. For example, -60 dBm is a stronger signal strength than -70 dBm. WiFi signal strengths of less than -80 dBm can be unreliable. In these cases, a WiFi repeater can be added to boost the WiFi signal strength and/or an external antenna can be attached to the water meter. The external antenna can be connected with a long (3 m) cable that allows the antenna to be attached to the outside of the well at the wellhead (Fig. 8). It is recommended that the antenna cable be inserted through the bottom of the case and thoroughly sealed with silicon to prevent moisture entry (Fig 6). A good-quality, waterproof, outdoor coaxial extension cable is recommended.

In cold climates, condensation and frost can form on the sensor face when outdoor air temperatures drop below freezing. This can block the ultrasonic beam and cause inaccurate water level readings. The problem can usually be solved by lowering the meter into the well by 1 to 2 m where temperatures are warmer. This approach has been tested in the field and found to solve most condensation and frost problems. An external antenna must be added in this case because the WiFi signal will likely be too weak when the meter is lowered deeper into the well.

Field testing has shown that the meter works well in large-diameter dug wells with smooth interior walls (i.e. concrete or plastic casing). However, problems with inaccurate water level readings have been encountered in older dug wells that are constructed with rocks instead of smooth-walled concrete or plastic casing. The presence of irregularly shaped rocks inside the well may cause the ultrasonic beam to reflect off the rocks rather than the water surface. This can be corrected by placing a small-diameter (75 mm or greater) PVC sleeve in the well that extends from the wellhead to below the water level and placing the water meter inside the PVC sleeve. This prevents the rocks from influencing the ultrasonic beam. It should be noted that the MaxBotix sensor model used here is specifically designed to ignore small acoustic targets and report the distance to the most likely target (i.e. the water surface). However, if there are significant obstacles in the well above the water level, such as pipes and wires, these obstacles may interfere with the ultrasonic beam and cause inaccurate water level readings, like the rocks discussed above. A PVC sleeve can also be installed in the well to solve this problem.

How to Make a Cellular Version of the Meter

A cellular version of the water meter can be built by making modifications to the above parts list and instructions. The cellular version does not require WiFi because it connects to the Internet via a cellular

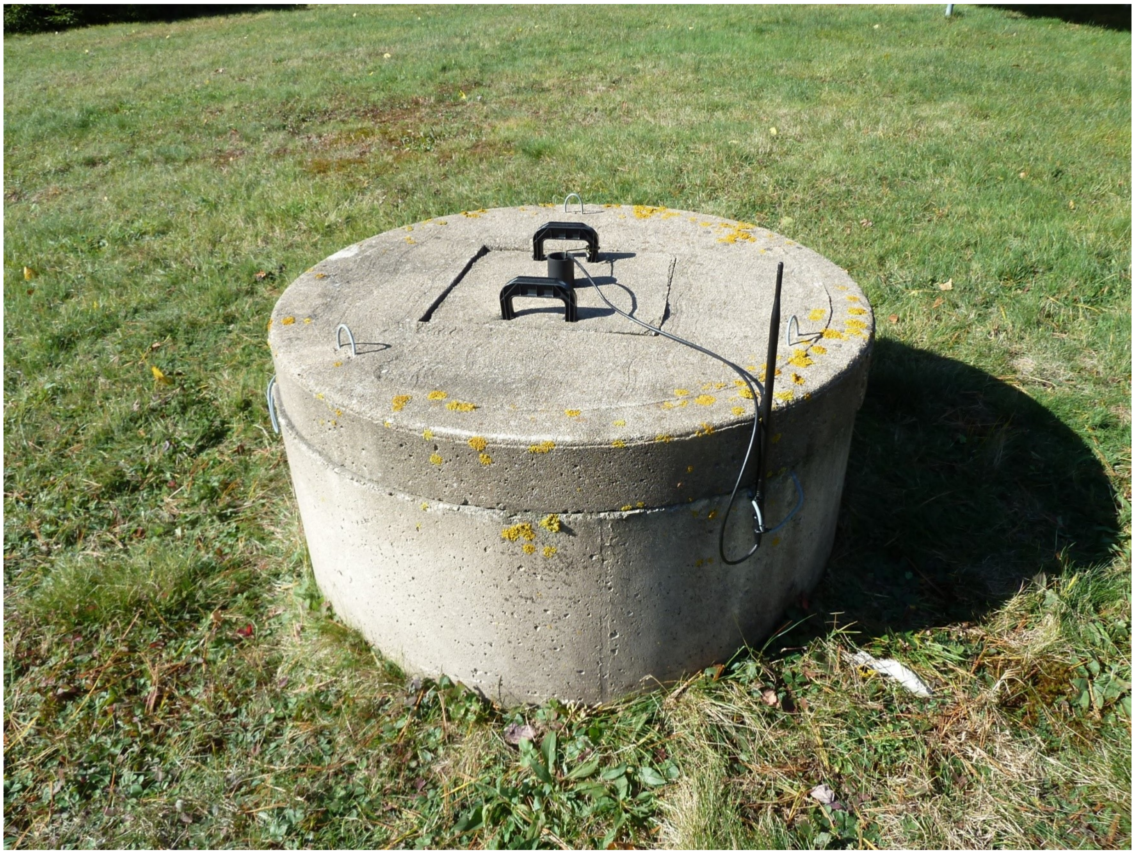


Figure 8. External antenna mounted on side of well with cable leading to meter inside well.

signal. The cost of the parts to build the cellular version of the meter is approximately Can$300 (excluding taxes and shipping), plus approximately Can$4 per month for the cellular data plan that comes with the cellular IoT device. Please note that at the time of writing, field testing of the cellular version has been underway for three months and is ongoing. The field unit is working as expected; however, more field testing is required to confirm battery life.

The cellular meter uses the same parts listed in Table 1 and construction steps listed above with the following modifications:

- Substitute the WiFi IoT device (Particle Photon) for a cellular IoT device (Particle Electron) (https://store.particle.io/collections/cellular). The cost for this cellular IoT device is approximately Can$110 for a 3G version with a 3.7 V Li-Po battery included. When constructing the meter, use the same pin connections described above for the WiFi meter in Step 3.

- The cellular IoT device uses more power than the WiFi version, and therefore two battery sources are needed: a 3.7V Li-Po battery, which comes with the IoT device, and the battery pack (4 AA batteries) listed in Table 1. The 3.7V LiPo battery attaches directly to the IoT device with the connectors provided. The AA battery pack is attached to the IoT device the same way as described above for the WiFi meter in Step 3. An alternative to using both the AA battery pack and the included 2000 mAh 3.7 V Li-Po battery is to use one 3.7V Li-Po battery with a higher capacity (e.g. 4000 or 5000 mAh).

- An external antenna must be attached to the meter. The same antenna listed in Table 1 can be used, but ensure it is rated for the frequency used by the cellular service provider where the water meter will be used. The antenna that comes with the cellular IoT device is not suitable for outdoor use. The external antenna can be connected with a long (3 m) cable that allows the antenna to be attached to the outside of the well at the wellhead. It is recommended that the antenna cable be inserted through the bottom of the case and thoroughly sealed with silicon to prevent moisture entry. A good-quality, waterproof, outdoor coaxial extension cable is recommended.

- The cellular IoT device runs on the same code provided in the Appendix, but three modifications must be made:

  1) Remove the following from line 7 of the code:

  ```
  STARTUP(WiFi.selectAntenna(ANT_EXTERNAL))
  ```

  This modification is needed because the cellular IoT device does not have an option to select an internal antenna, and therefore the code will not compile if this line is left in place.

  2) Insert the following into line 36 of the code, which is currently an empty line:

  ```
  PMIC().disableCharging();
  ```

  This modification is needed to prevent the AA battery pack from charging the 3.7V LiPo battery.

  3) Change the value from 25000 to 90000 in the following line 61 of code:

  ```
  if ( !waitFor(Particle.connected, 25000) )
  ```

  This modification is needed because the cellular IoT device can take longer to connect to the Internet than the WiFi version, and therefore the longer time-out value of 90 seconds (i.e. 90000 milliseconds) is required.

Acknowledgments

The author would like to thank Julia Cantelon and Nicole LeRoux at Dalhousie University for testing these instructions by building a water meter and providing review comments. The instructions were substantially improved by their suggestions.

References

MaxBotix Inc., 2019. HRXL-MaxSonar®-WR™ series. MaxBotix Inc., 22 p. <https://www.maxbotix.com/documents/HRXL-MaxSonar-WR_Datasheet.pdf> [Accessed June 3, 2019]

Ousley, T., 2019. Wi-Fi twitter water level sensor; instructables circuits, Autodesk Inc. <https://www.instructables.com/id/Wi-Fi-Twitter-Water-Level-Sensor/> [Accessed October 20, 2016]

Particle, 2019. Photon datasheet (V016), model number: PHOTONH, PHOTONNOH; Particle <https://docs.particle.io/datasheets/wi-fi/photon-datasheet/> [Accessed June 12, 2019]

ThingSpeak, 2019. ThingSpeak for IoT projects: Data collection in the cloud with advanced data analysis using MATLAB; The MathWorks Inc. <https://thingspeak.com/> [Accessed June 3, 2019]

Appendix – Code

```
#include <math.h>

// Coded by J.Drage, 17-Jun-2019

STARTUP(System.enableFeature(FEATURE_RETAINED_MEMORY));
SYSTEM_THREAD(ENABLED);
STARTUP(WiFi.selectAntenna(ANT_EXTERNAL)); // selects the u.FL antenna

//------------------------------------------------------------------------
// Photon pins:
//  MAX_AI_PIN        Pin A0, input from the sensor
//  MAX_EN_PIN         Pin D2, powers the sensor on
//------------------------------------------------------------------------

const int MAX_EN_PIN = D2;
const int MAX_AI_PIN = A0;
const float V_PER_ADC_CODE = 805.86e-6; //3.3V/4095codes=0.00080586;  scale factor
const float CM_PER_V = 155.152; // 512cm/3.3V=155.152cm/V; scale factor

//------------------------------------------------------------------------
// Variables
//------------------------------------------------------------------------

double range_m=0; //Measured range from the sensor (metres)
double GWelevation=0; // Water table elevation (metres)
double HangDown=0.0; // Distance from datum to sensor, need to enter this for each well (metres)
double Datum=0.0; //Elevation of datum, typically top of casing is used for the datum (metres)

//------------------------------------------------------------------------

void setup() {
   //----------------------------------------------------------------------
   // Configure pins on the Photon
   //----------------------------------------------------------------------

   digitalWrite(MAX_EN_PIN, HIGH);
   pinMode(MAX_EN_PIN, OUTPUT);
   digitalWrite(MAX_EN_PIN, HIGH);

   Serial.begin(9600);
}

void loop() {

   //----------------------------------------------------------------------
   // Read the range from the sensor
   //----------------------------------------------------------------------

   delay(1000);
   int adcReading = analogRead(MAX_AI_PIN);
   range_m = adcReading*V_PER_ADC_CODE*CM_PER_V/100;
   GWelevation = (Datum-HangDown-range_m);

   //----------------------------------------------------------------------
   // Check to see if Photon is connected to Wifi, if not then go to sleep to preserve the batteries
   //----------------------------------------------------------------------

   if( !Particle.connected() ) {
      Particle.connect();
      if ( !waitFor(Particle.connected, 25000) ) {
         digitalWrite(MAX_EN_PIN, LOW);
         System.sleep(SLEEP_MODE_DEEP, 120);
      }
   }

   //----------------------------------------------------------------------
   // Check to see if the water depth is out of sensor range and if so, go to sleep without reporting the
result
   //----------------------------------------------------------------------

   if (range_m<0.31) {
     delay(20000);
     System.sleep(SLEEP_MODE_DEEP, 120);
   }

    if (range_m>4.93) {
     delay(20000);
     System.sleep(SLEEP_MODE_DEEP, 120);
   }

   //----------------------------------------------------------------------
   // Send the water level to real-time chart at ThingSpeak.com
   //----------------------------------------------------------------------

   String level = String(GWelevation);
```

```
  Particle.publish("delete this text and insert webhook name inside these parentheses", level, PRIVATE);

  // Wait 20 seconds
  delay(20000);
  digitalWrite(MAX_EN_PIN, LOW);
  System.sleep(SLEEP_MODE_DEEP, 120);
}
```