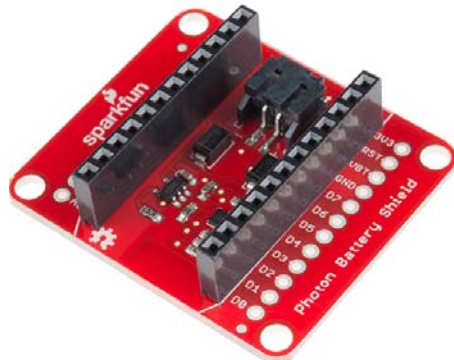




Photon Battery Shield Hookup Guide

Introduction

Free your Photon from its USB cable by powering it through the Photon Battery Shield. The Battery Shield has everything your Photon needs to run off, charge, and monitor a LiPo battery.



The Shield features two unique IC's: an MCP73831 charge controller and a MAX17043 LiPo fuel gauge. With them, you'll be able to charge your battery through USB and monitor its voltage and state-of-charge.

Please Note: All SparkFun shields for the Photon are also compatible with the Core from Particle. The WKP, DAC and VBT pins on the Photon will be labeled A7, A6 and 3V3*, respectively, on the Core, but will not alter the functionality of any of the Shields.

Covered In This Tutorial

The purpose of this hookup guide is to familiarize you with the hardware and software of the Photon Battery Shield. It's split into the following sections:

- Battery Shield Overview – A quick overview of the components and features of the Photon Battery Shield.
- Using and Charging a LiPo Battery – Some tips and tricks for running off, and charging a LiPo battery with the Shield.
- Using the MAX17043 LiPo Fuel Gauge – Example code – including a Particle library – demonstrating how to read the voltage and state-of-charge outputs, plus other features of the MAX17043 IC.

Required Materials

The Photon Battery Shield equips your Photon with just about everything it should need to use a LiPo battery – it even includes headers! Of course you'll need the Battery Shield and a Photon.



Photon Kit
 © KIT-13345

★★★★★ 9



SparkFun Photon Battery Shield

© DEV-13626

★★★★★ 1

The one thing the Battery Shield doesn't include is a battery. Our compatible LiPo batteries come in a variety of shapes and capacities – the larger the battery, the longer it will last. Any of the following will work:



Polymer Lithium Ion Battery - 2000mAh
 © PRT-08483

★★★★★ 7



Polymer Lithium Ion Battery - 400mAh
 © PRT-10718

★★★★★ 10



Polymer Lithium Ion Battery - 110mAh
 © PRT-00731

★★★★☆ 4



Polymer Lithium Ion Battery - 1000mAh
 © PRT-00339

★★★★★ 9



Polymer Lithium Ion Battery - 850mAh



Polymer Lithium Ion Battery - 6Ah
 © PRT-08484

★★★★☆ 6



Polymer Lithium Ion Battery - 40mAh

© PRT-11316

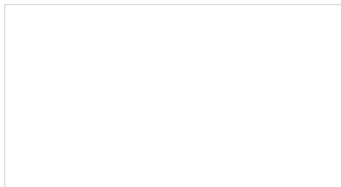
If you want to use a battery of your own, just make sure it's a **single-cell** (3.7V nominal, ~4.2V max) lithium-polymer (LiPo) or lithium-ion (Li+). Optimally, choose one that is terminated with a 2-pin PH-series JST connector, otherwise you may need to do some wire splicing.

If you don't already have one (or just want to stock up), you may also need a Micro-B USB Cable, which will come in handy when you need to recharge the battery. If you don't want to plug that cable into a computer, a USB Wall Charger is useful to have on hand.

Suggested Reading

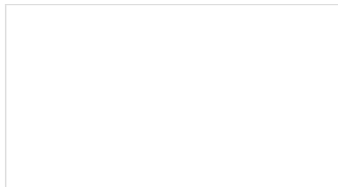
Using the Photon Battery Shield doesn't require a whole lot of pre-existing knowledge. If you want to learn more about the foundational concepts in this tutorial, here are some guides we recommend:

- [How to Power a Project](#) – Hopefully you've already figured this out, since you're reading a Battery Shield tutorial. This tutorial does have an enlightening section on battery power.
- [Battery Technologies](#) – Specifically, check out the section on Lithium Polymer batteries.
- [How Lithium Polymer Batteries are Made](#) – Learn everything you always wanted to know about what goes on behind the scenes at a battery factory.
- [I²C Communication](#) – I²C is a popular low-level, two-wire communication standard. This is what we'll use to communicate with the MAX17043 fuel gauge IC.



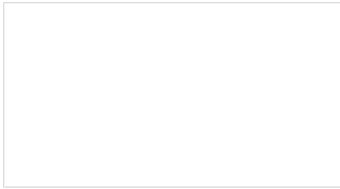
Battery Technologies

The basics behind the batteries used in portable electronic devices: LiPo, NiMH, coin cells, and alkaline.



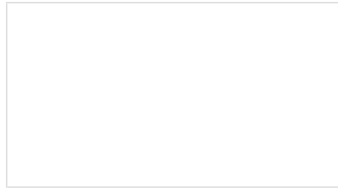
How to Power a Project

A tutorial to help figure out the power requirements of your project.



I2C

An introduction to I2C, one of the main embedded communications protocols in use today.

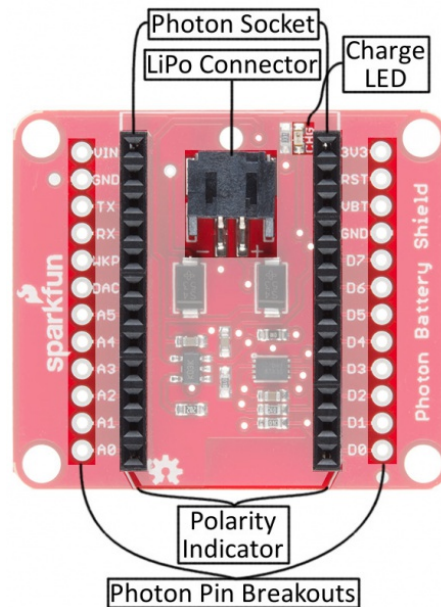


How Lithium Polymer Batteries are Made

We got the opportunity to tour the Great Power Battery factory. Checkout how LiPos are made!

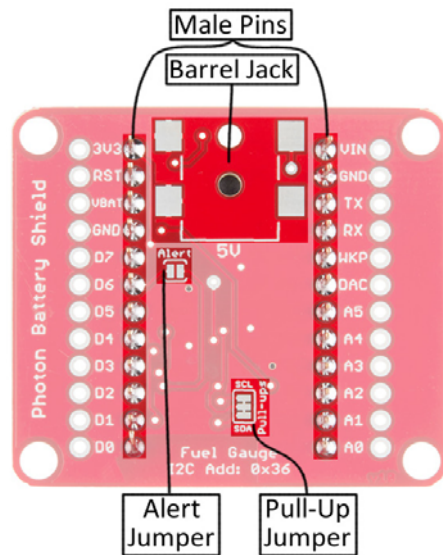
Battery Shield Overview

Let's do a quick rundown of the Photon Battery Shield's hardware features. Here are some highlights of the board's main components:



- **LiPo Connector (PH-Series JST)** – The battery goes here. This connector is polarized, so it'll only plug in the right way.
- **Photon Socket** – Each Photon comes pre-soldered with male headers, so all of our shields come pre-populated with female connectors on their top-side.
- **Photon Pin Breakouts** – Need more access to the Photon's I/O? These pin break-outs have that covered. These break outs are in the same place on each of our Photon Shields – should you need to interface them together that way.
- **Photon Polarity Indicator** – Nothing's going to work if your Photon isn't plugged in correctly. These angled edges should match up to those on the Photon's PCB.
- **Charge LED Indicator** – This red LED will light up whenever the LiPo battery is being charged.

Flipping the Shield over reveals a few, less commonly-used features of the board:

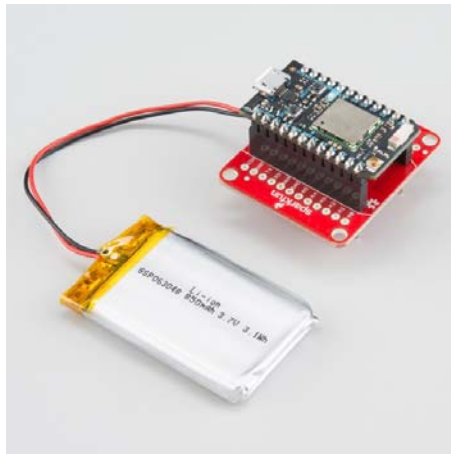


- **“Stackable” Male Pins** – Whether you want to plug the battery shield into more Photon shields, or use it in a breadboard, these male pins should do the job. If you don’t need this feature, the pins can all be quickly beheaded by some flush cutters.
- **SMD Barrel Jack footprint** – This footprint allows you to add an SMD Barrel Jack to the Battery Shield’s underside. This’ll allow you to charge the battery with a 5V Wall Adapter, or even a Small Solar Panel. Just make sure your power source isn’t higher than **5.5V**.
- **I²C Pull-up Resistor Enable/Disable** – The Photon Battery Shield includes a pair of 10kΩ pull-up resistors on the SDA and SCL lines (D0 and D1). If you’ve got more I²C devices to connect, you may need to disable those. These jumpers are closed by default, but can be opened with a couple slices of a hobby knife.
- **Alert Interrupt Jumper** – The MAX17043 features a programmable interrupt output, which can fire whenever your battery gets below a set percentage of charge. This pin can be connected to the Photon’s **D6 pin** by adding a blob of solder between these two pads. Note that this jumper is open by default, as other shields make use of D6.

Using and Charging a LiPo Battery

Before plugging your battery in, grab your Photon, and plug it into the battery shield. Make sure you **line up the Photon’s angled edges** with the matching white silkscreen on the shield. The Photon’s USB connector should be pointing out the same direction as the black JST connector.

Next, grab your LiPo battery and mate its white JST connector with the black one on the shield. These connectors are polarized, so you can only plug it in the *right* way.



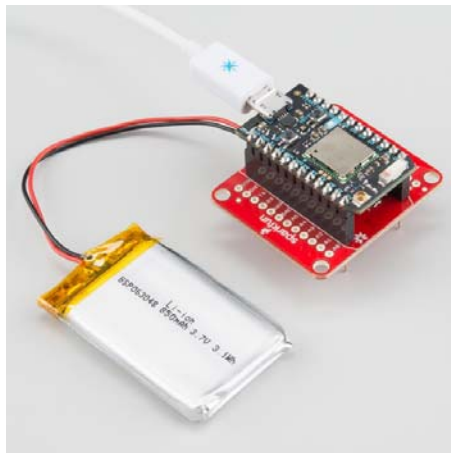
Running your Photon off a LiPo is as simple as plugging in a battery and the Photon.

If the battery had any charge, your Photon should turn on, and its RGB LED should begin being colorful. If it was already commissioned, your Photon should connect to your WiFi network. You can even load up the Particle Build IDE and truly start loading code over-the-air.

Charging the Battery through USB

Eventually your LiPo's power capacity will be drained, and it'll be time to charge it back up. To avoid any extra USB connectors, the Photon Battery Shield was designed to use the **Photon's USB connector** as a charge-source.

With the LiPo and Photon still connected to the Battery Shield, simply plug a Micro-B USB cable into your Photon (the other end of the USB cable can be plugged into a computer or USB wall adapter).



To charge the LiPo, just plug a USB cable into the Photon's USB port.

Once USB is attached, the red charge LED indicator should illuminate – it'll remain lit up until the battery is fully charged.

(Optional) Charging the Battery through Barrel Jack

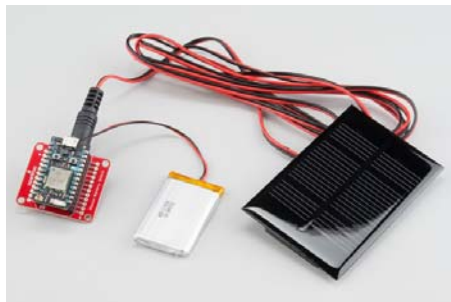
The Battery Shield provides additional options for charging your LiPo in an unpopulated barrel jack footprint. If you need to charge the LiPo sans-Photon, this may be the best option for you.

To add this feature, you'll need our Surface-Mount Barrel Jack, and some soldering tools. If you're a novice solderer, don't be scared off by the "SMD" soldering – these joints are about as easy as it gets. Check out our soldering tutorial for some guidance.

The barrel jack's input should face out towards the edge of the board. Make sure you solder all four pads:



To charge a LiPo through this jack, you'll need a **5V power source** – our 5V Wall Adapter should do the trick. Or, if you really want to avoid wires, you can use our small, 0.45W solar panel, which will source up to 5V when it's nice and sunny.



If you don't want to solder the barrel jack on, but still have a 5V power source available, you can supply that voltage to the "VIN" and "GND" pins on the shield's header. This is a more advanced charge technique, and only recommended for more experienced users.

Using the MAX17043 LiPo Fuel Gauge

One of the most unique features of the Photon Battery Shield is the integrated MAX17043 LiPo fuel gauge. The MAX17043 sits between the battery and your Photon – it uses a calibrated ADC to measure the battery voltage. Comparing that measured voltage against their "ModelGauge" algorithm, the IC can produce a state-of-charge (SOC) estimate, to give you an idea of what percentage remains of the battery charge.

Loading the SparkFunMAX17043 Library

The MAX17043 communicates over I²C, so coding an interface between the Photon and the fuel gauge can be a little tricky. Luckily, we've done the hard work for you! We've written a Particle library for the MAX17043. You can load it up in the Particle IDE by going to the "Libraries" tab, and searching for **SparkFunMAX17043**.



Search the “Libraries” tab for “SparkFunMAX17043” to find the *Fuel Gauge* library.

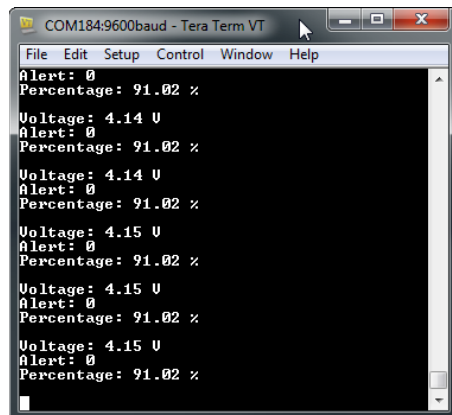
If you’re using the desktop version of the IDE, Particle Dev, and still want to use the library, you can grab the latest version from our GitHub repository.

Running the MAX17043_Simple Example

After finding the library, navigate to “MAX17043_Simple.cpp” and click **Use This Example** – the Build IDE will create a clone of this sketch in your “Code” tab.

You shouldn’t have to change anything, just make sure your Photon is selected in the “Devices” tab, and click “Flash”.

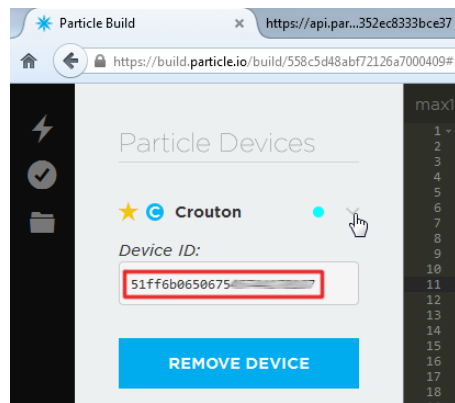
Once your Photon is running the code, there are two ways to view its data. One is to open a serial terminal (see our Serial Terminal Basics tutorial if you need help with this). Find your Photon’s serial port number (“COM#” on Windows “/dev/tty.usbmodemXXXX” on Mac) and set the baud rate to 9600. The battery’s voltage, state-of-charge, and alert status will stream by.



Using TeraTerm to view the Photon’s *Serial.print* debug output.

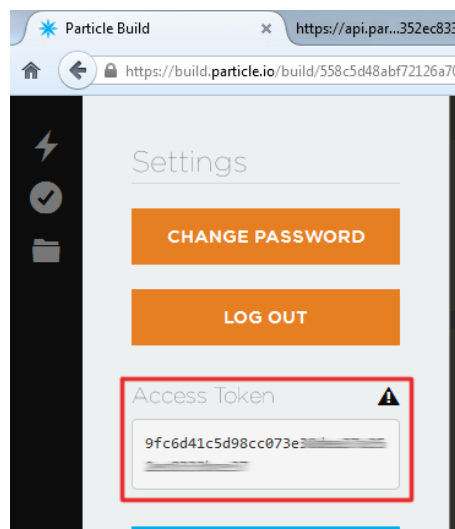
Since USB is plugged in, the voltage and percentage should steadily increase as the battery charges.

Alternatively, if you want to monitor the battery *discharge*, you can use the Photon’s Internet-connectivity to view the battery voltage and charge status. First, though, you’ll need to identify your Photon’s **device ID** as well as your account’s **access token**. The device ID can be found in Particle Build by clicking the ‘>’ next to your device name.



Find your Device ID under the “Devices” tab, by clicking the carrot next to your Photon.

Your access token can be found under the “Settings” tab.



Find your access token under the “Settings” tab.

Armed with those long strings of hex characters, open a new browser tab and navigate to:

```
https://api.particle.io/v1/devices/DEVICE_ID/voltage?access_token=ACCESS_TOKEN
```

Make sure to sub in the proper values for `DEVICE_ID` and `ACCESS_TOKEN`. If everything was entered correctly, you should see something like this:



The Spark variable responds with a JSON string including a “result” key and value.

Among the data in that JSON response is a “result” key, which shows the current voltage reading. You can also view the state-of-charge, and alert status by navigating to:

```
https://api.particle.io/v1/devices/DEVICE_ID/soc?access_token=ACCESS_TOKEN
```

and

```
https://api.particle.io/v1/devices/DEVICE_ID/alert?access_token=ACCESS_TOKEN
```

Yay Spark Variables!

Using the SparkFunMAX17043 Library

The SparkFunMAX17043 library is simple. There's some initialization required. Make sure to include the library, and in your `setup()` call `lipo.begin()`.

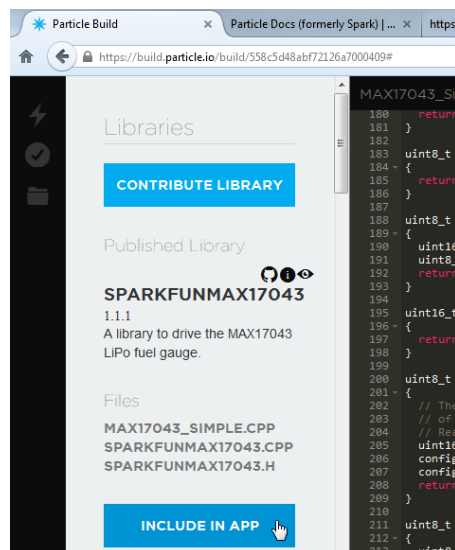
```
#include "SparkFunMAX17043/SparkFunMAX17043.h" // Include the
SparkFun MAX17043 library

void setup()
{
  // Set up the MAX17043 LiPo fuel gauge:
  lipo.begin(); // Initialize the MAX17043 LiPo fuel gauge

  // Quick start restarts the MAX17043 in hopes of getting
  a more accurate
  // guess for the SOC.
  lipo.quickStart();
}
```

Calling `lipo.quickStart()` will re-calibrate the MAX17043's ADC, and usually results in more accurate readings.

If you're using the library in Particle Build, make sure you go to the SPARKFUNMAX17043 library in the "Libraries" tab, select **INCLUDE IN APP**, and add it to your desired code file.



Any sketch using the SparkFunMAX17043 library needs to have it "included" first.

Reading Voltage and SoC

Two functions are used to read the MAX17043's voltage and state-of-charge values:

```
// lipo.getVoltage() returns a voltage value (e.g. 3.93)
voltage = lipo.getVoltage();
// lipo.getSOC() returns the estimated state of charge (e.g. 79%)
soc = lipo.getSOC();
```

Both functions return a `float` variable. `lipo.getVoltage()` should usually be between 0.0 and about 4.2. `lipo.getSOC()` should be somewhere between 0.0 and 100.0.

Using the Alert Interrupt

One of the MAX17043's nifty features is a programmable alert interrupt. You can tell it to trigger a flag whenever the state-of-charge falls below a certain threshold.

To set up the alert, use `lipo.setThreshold([percentage])`. For example:

```
lipo.setThreshold(15); // Set Alert threshold to 15%
```

...will set the alert threshold to 15%.

The alert status can be read in both software and hardware. To get the alert status in software, call `lipo.getAlert()`. This function will return `0` if the alert is not triggered, and `1` if it is.

As mentioned in the Battery Shield Overview section, a jumper on the bottom of the board can optionally be closed to connect the MAX17043's alert pin to Photon pin D6. This alert is **active-low** – meaning it'll be HIGH when it's not triggered and LOW when the SoC has fallen below threshold.

Once an alert has been triggered, you'll need to clear it before it can fire again. That's what `lipo.clearAlert()` will do for you. Even if the SoC rises above the threshold, you'll still need to manually clear the alert.

Resources & Going Further

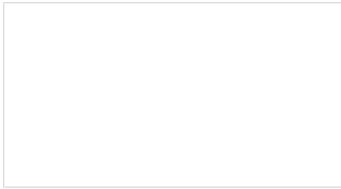
If you need any resources detailing the Photon Battery Shield, here are some schematics, datasheets, and other links you may find handy:

- Photon Battery Shield GitHub Repo
- Photon Battery Shield Schematic
- Particle Documentation Pages – go here to set up and configure your Photon (or other Particle devices)
- Particle Community Forum – anything that you couldn't find in the docs should be easily found in the community forum. If you are having trouble, search this forum first, as many of the answers are there already.
- MAX17043 LiPo Fuel Gauge Datasheet
- MCP73831 Charge Controller Datasheet

If you are interested in putting your Photon into sleep mode to conserve more battery power, check out the Particle Docs links below:

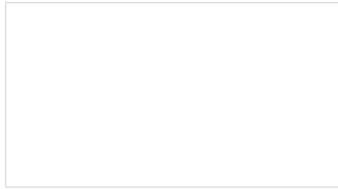
- Photon System Modes - Playing with different modes allows you to have control over when the Photon is and isn't connected to the web.
- Photon System Sleep Function - The sleep function puts the Photon to sleep while it isn't performing any actions to conserve power.

Now that you've become familiarized with the Photon Battery Shield, what project are you going to make with it? Need some inspiration? Check out the below SparkFun tutorials for some ideas:



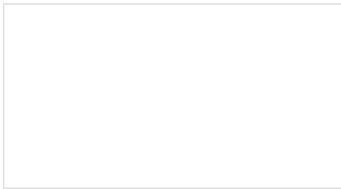
HID Control of a Web Page

Learn how to move a slider on a webpage and make a motor spin. We connect HTML and HID to read sensors and interface with the physical world.



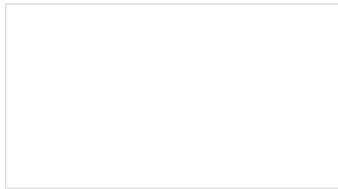
Are You Okay? Widget

Use an Electric Imp and accelerometer to create an "Are You OK" widget. A cozy piece of technology your friend or loved one can nudge to let you know they're OK from half-a-world away.



Pushing Data to Data.SparkFun.com

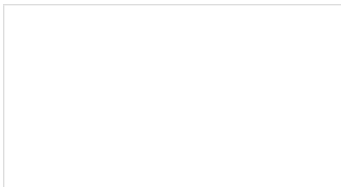
A grab bag of examples to show off the variety of routes your data can take on its way to a Data.SparkFun.com stream.



ESP8266 Thing Hookup Guide

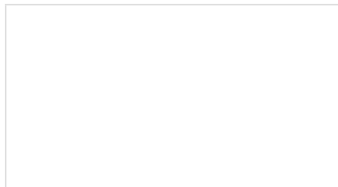
An overview of SparkFun's ESP8266 Thing - a development board for the Internet of...Things. This tutorial explains the circuitry driving the board, and how to get it up and running in an Arduino environment.

The Battery Shield pairs very well with any of our other Photon Shields; check out our hookup guides for those shields:



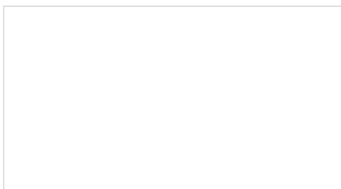
Photon Wearable Shield Hookup Guide

Learn how to use the Photon Wearable Shield for your next projects!



Photon OLED Shield Hookup Guide

The Photon OLED Shield has everything you need to add a small yet crisp OLED screen to your Photon projects. This hookup guide will show you how to get started.



Photon IMU Shield Hookup Guide

Learn how to use the SparkFun Photon IMU Shield for your Photon device which houses an on-board LSM9DS1 system-in-a-chip that houses a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer.