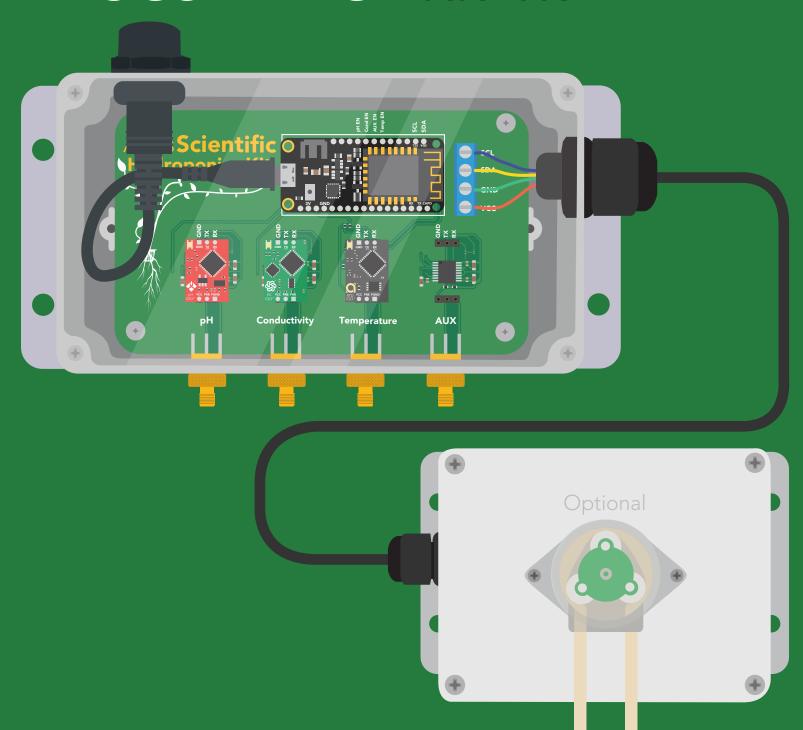


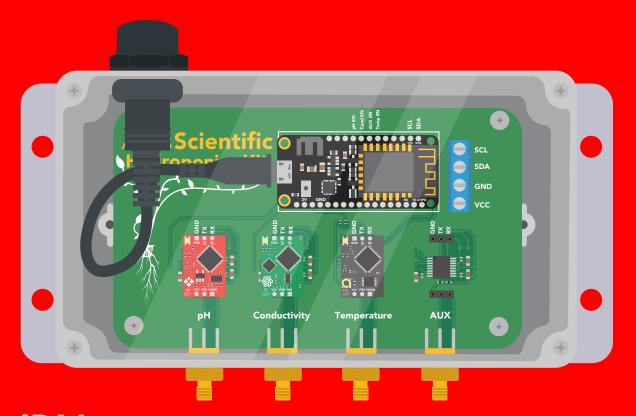
Wi-Fi Hydroponics Meter Kit Datasheet



Atlas Scientific does not make consumer electronics.

This equipment is intended for electrical engineers. If you are not familiar with electrical engineering or embedded systems programing, this product may not be for you.

This device was developed and tested using a Windows computer. It was not tested on Mac, Atlas Scientific does not know if these instructions are compatible with a Mac system.



IP64 (dust and water splash proof)

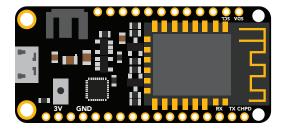
Operating principle

The Wi-Fi hydroponics kit has been designed to provide the engineer with a simple way of remotely monitoring and controlling a hydroponics system's chemistry. Sensor data is uploaded to ThingSpeak $^{\text{M}}$, a free, cloud-based data acquisition and visualization platform. The Wi-Fi hydroponics kit has also been designed to be easily modified by the engineer. Feel free to change the sensors or functionality of the device to meet your specific needs.

Overview

CPU

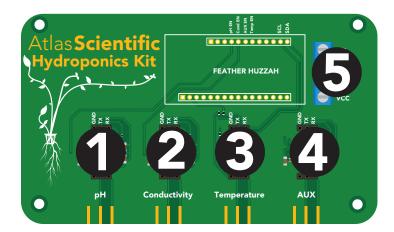
The Wi-Fi hydroponics kit is controlled using an Adafruit Feather HUZZAH as its CPU. The HUZZAH is programmed using the Arduino IDE and uses an onboard ESP8266 as its Wi-Fi transmitter. Adafruit Feather HUZZAH datasheet.



Sensor ports

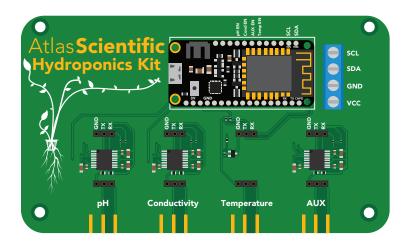
The Wi-Fi hydroponics kit PCB has 5 sensor ports. Three of the ports are electrically isolated. The isolated ports are marked pH, Conductivity, and AUX. The isolated ports are needed to take noise-free electrochemical readings. Because the sensing element of a temperature sensor is never in direct contact with the water, electrical isolation is not needed for temperature sensing.

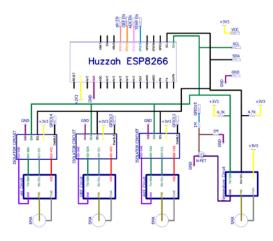
The AUX port can be used to add an additional sensor of your choice. The terminal block marked Port 5 has been designed to connect one or more dosing pumps to the device. However, the port could also be used to connect a gas sensor.



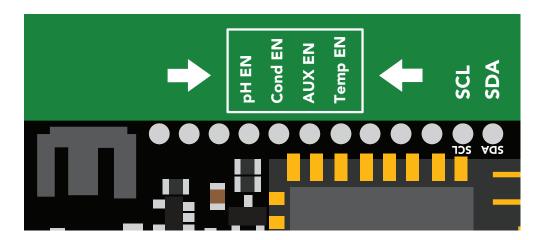
PCB

The overall design of the PCB is quite simple. The CPU is powered and programmed through the panel-mount USB connector. The CPUs onboard voltage regulator supplies the board's power bus with 3.3V at 500 mA peak. All connected sensors are running at 3.3V.





Each of the four main sensor ports have an enable pin, which must be set correctly to power the sensor. The enable pins are found here:



The first three pins (pH, Cond and Aux) must be set low to power on the sensor. The last pin (Temp) must be set high to power on the sensor.

Truth table

Pin	State	Sensor Power
рН EN	LOW	ON
Cond EN	LOW	ON
Aux EN	LOW	ON
Temp EN	HIGH	ON

Sensor port 5 (the terminal block) does not have an enable pin and can not be turned off.



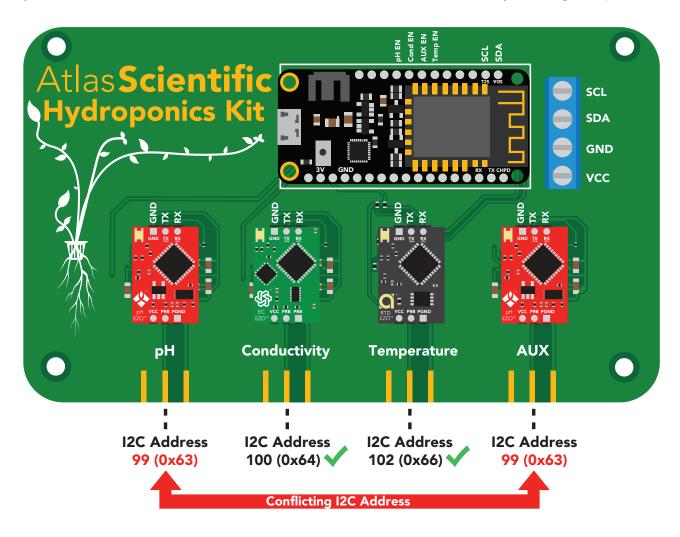
Data protocol

The CPU communicates with all peripheral sensors using the I2C data protocol. All data lines are directly connected to the CPUs I2C port. Using a different data protocol with this circuit board is not possible.

It is important to keep in mind that all Atlas Scientific components default to UART mode. When adding a new Atlas Scientific component to the kit, it must first be put into I2C mode. Refer to the component's datasheet for instructions on how to switch it over.

Adding more of the same sensor or component type

Adding additional components of the same type, such as an additional pH or conductivity sensor, is not hard to do. As mentioned above, you must set the device to I2C mode, and you must make sure that its I2C address is not the same as the already existing component.





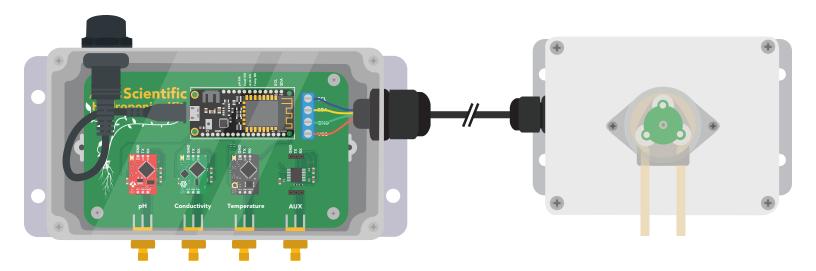
This table lists the default I2C address of components commonly added to this kit.

Device	I2C Address
EZO pH	99 (0x63)
EZO ORP	98 (0x62)
EZO DO	97 (0x61)

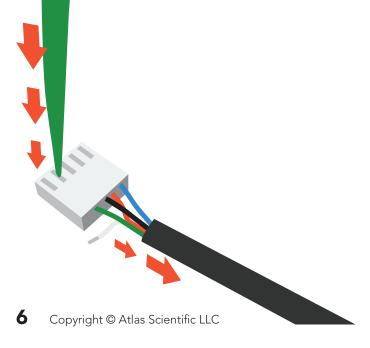
Device	I2C Address
EZO EC	100 (0x64)
EZO RTD	102 (0x66)
EZO PMP	103 (0x67)

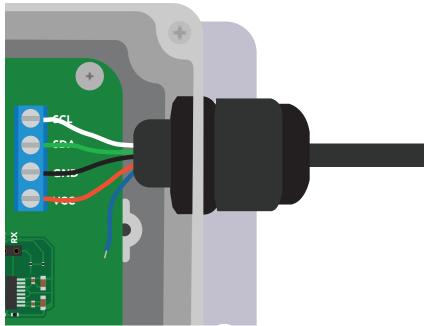
Dosing pump

An optional external dosing pump can be added to the Wi-Fi hydroponics kit. Using the <u>EZO-Expansion Pump Kit</u> is the simplest way to add on a dosing pump.



A stand-alone EZO-PMP can be used instead of the expansion pump kit; however, you must manually put the pump in I2C mode and remove the data cable connector.





Uploading sensor data to the cloud

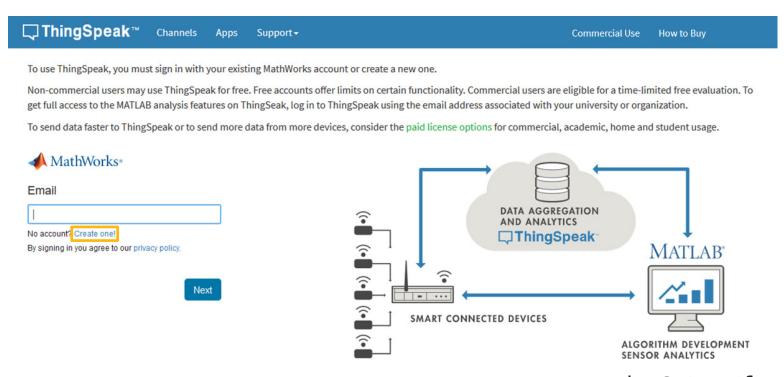
The Atlas-Scientific Wi-Fi hydroponics kit has been designed to upload sensor data to ThingSpeak[™], a free, cloud-based data acquisition and visualization platform. You will be required to set up a free account with ThingSpeak[™] to upload and visualize the data. With a free account, you can upload data once every 15 seconds. A paid account lets you upload data once per-second; look here for more info about various ThingSpeak[™] services.

Atlas Scientific has no business relationship with ThingSpeak™; we just like how it works. If you want to use a different service, modify the device as you see fit.

Setting up your Wi-Fi kit

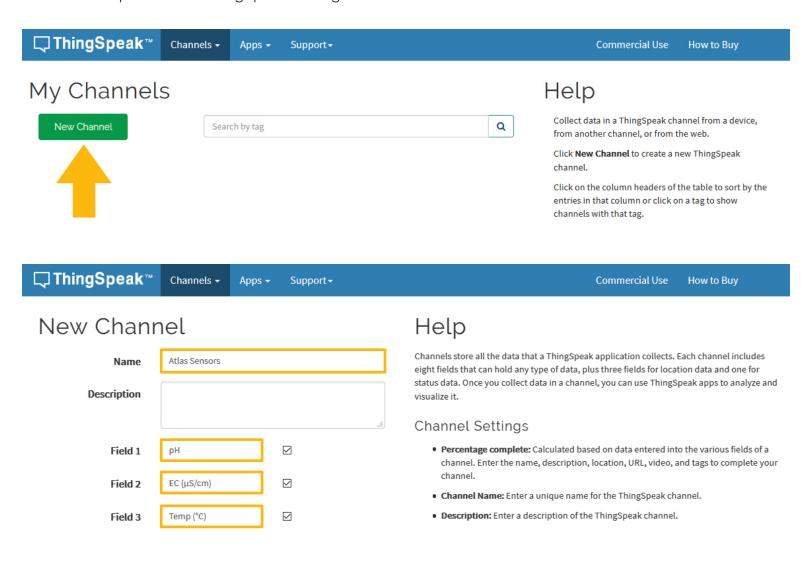
Step 1 Setup a ThingSpeak Account

Because the sensor data is stored / viewed on ThingSpeak, you will need to setup a ThingSpeak account. Create your ThingSpeak account by clicking <u>HERE</u>.



Step 2 Create a Channel

Your data is uploaded to ThingSpeak through a 'Channel.' Select New Channel



Fill out the highlighted boxes. (Be sure to click on the checkboxes to enable **field 2** and **3**) For reference, this is what we entered.

Name Atlas Sensors

Field 1 pH

Field 2 EC (µS/cm)

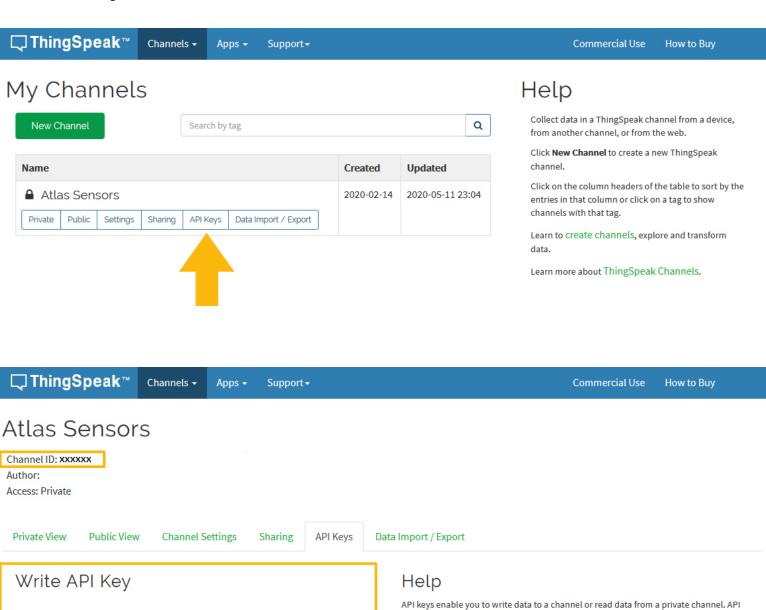
Field 3 Temp (°C)

Scroll to the bottom of the page and click Save Channel.



Step 3 Get ThingSpeak API keys

After you saved your channel settings, you will be redirected to your channel page. Click on **API keys**.



keys are auto-generated when you create a new channel.

been compromised, click Generate New Write API Kev.

. Write API Key: Use this key to write data to a channel. If you feel your key has

• Read API Keys: Use this key to allow other people to view your private channel

API Keys Settings

Be sure to save your **Channel ID** and **Write API Key** we are going to need these, in the next few steps.



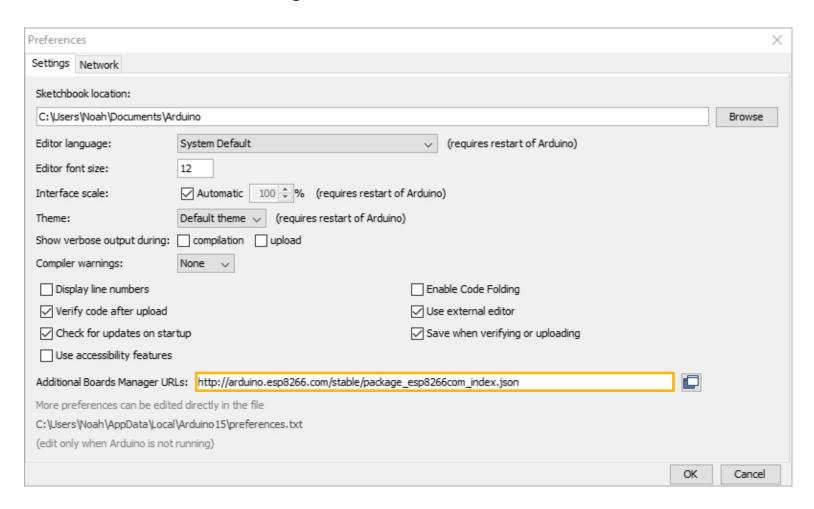
Key

XXXXXXXXXXXXXXX

Step 4 Make sure your Arduino IDE libraries are up to date

A Make sure you have the correct path for the Esp8266 Library

In the IDE, go to File > Preferences Locate the Additional Boards Manager URLS text box.

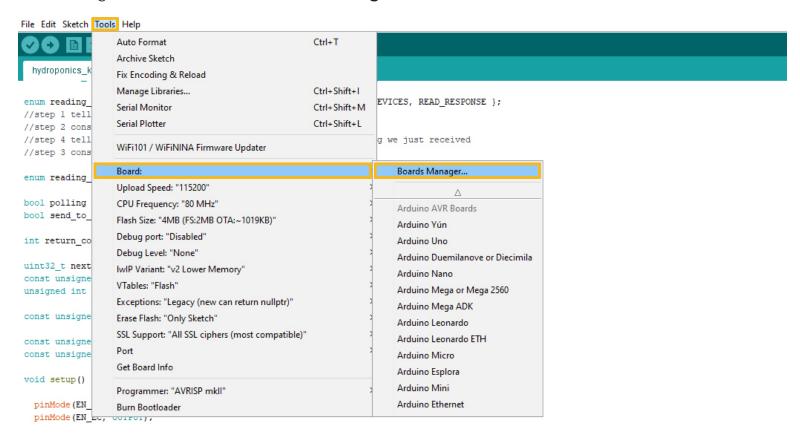


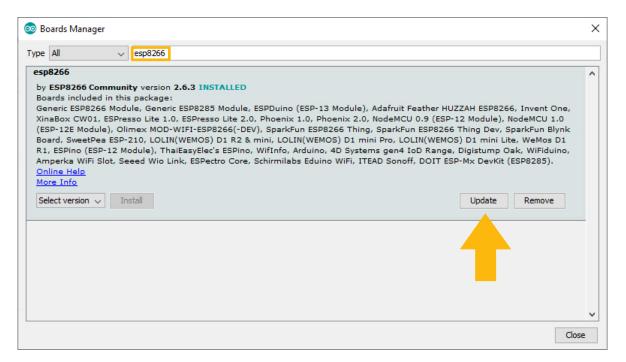
Make sure this URL is in the textbox http://arduino.esp8266.com/stable/package_esp8266com_index.json Click OK.



Update the esp8266 board

In the IDE, go to Tools > Board > Boards Manager





In the search bar of the Boards Manager, lookup esp8266. Update to the most recent version if you don't already have it.

(Version 2.6.3 in not the most recent version)



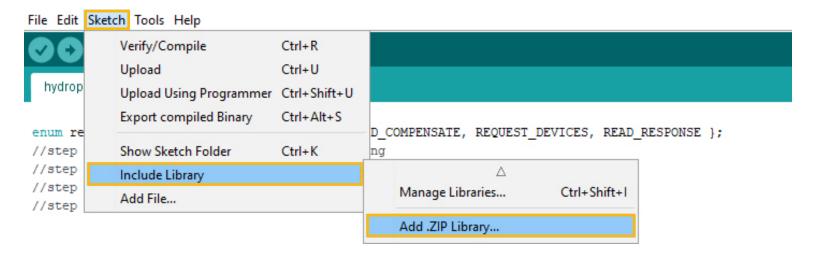
Download the ThingSpeak library for Arduino

Click HERE to download the latest version of the ThingSpeak library.

Don't unzip it!

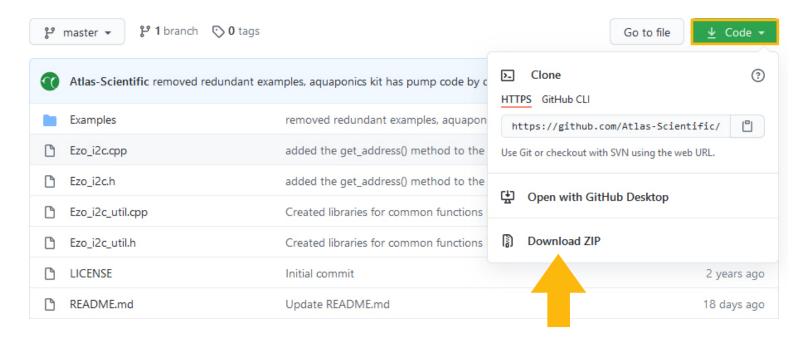
Import the .ZIP file into your Arduino IDE.

To import the .ZIP file go to **Sketch > Include Library > Add .ZIP Library**



Add the EZO I2C Library

To download the Ezo_I2c library file, click <u>HERE</u>.



Don't unzip it!

Import the .ZIP file to your Arduino IDE.

To import the .ZIP file go to Sketch > Include Library > Add .ZIP Library



Step 5 Flash the Hydroponics meter with the correct code

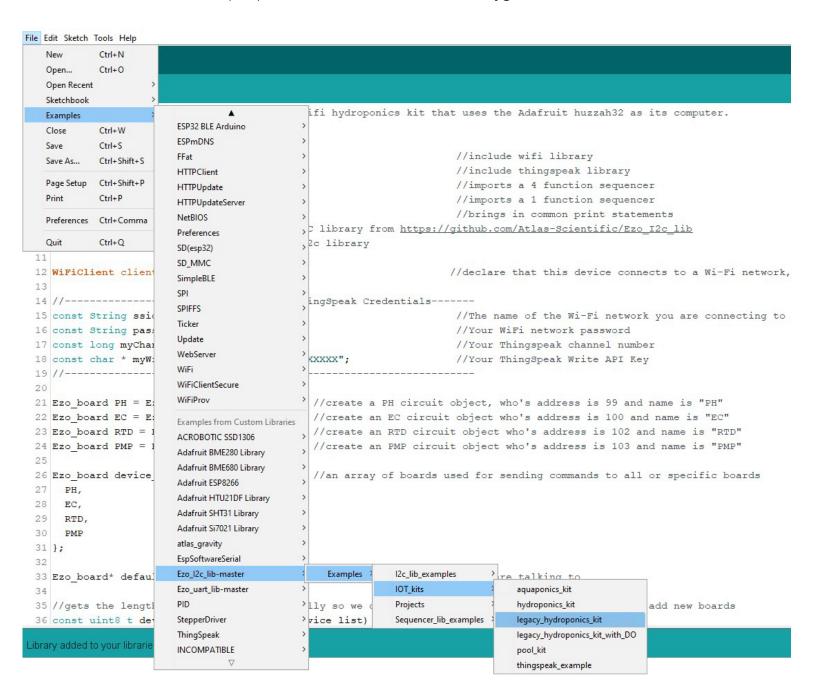
A Select, open and adjust the code you want to use for your Wi-Fi Kit

File> Examples> EZO_I2C_lib-master> Examples> IOT_kits>

You have two choices of code:

Legacy Hydroponics_kit or Legacy Hydroponics_kit_with_DO

(The "with DO" version is for people who want to add Dissolved oxygen)





Fill in your Wi-Fi / ThingSpeak credentials

Fill in your Wi-Fi name and Password, along with the Channel ID and Write API Key to the code. (see step 3)

hydroponics_kit | Arduino 1.8.13 File Edit Sketch Tools Help

```
hydroponics_kit
 1 #include <iot cmd.h>
 2 #include <ESP8266WiFi.h>
                                                                    //include esp8266 wifi library
 3 #include "ThingSpeak.h"
                                                                    //include thingspeak library
 4 #include <sequencer4.h>
                                                                    //imports a 4 function sequencer
 5 #include <sequencer1.h>
                                                                    //imports a 1 function sequencer
 6 #include <Ezo i2c util.h>
                                                                    //brings in common print statements
 7 #include <Ezo_i2c.h> //include the EZO I2C library from https://github.com/Atlas-Scientific/Ezo_I2c_
 8 #include <Wire.h> //include arduinos i2c library
10 WiFiClient client;
                                                                   //declare that this device connects
12 //----Fill in your Wi-Fi / ThingSpeak Credentials-----
13 const String ssid = "Wifi Name";
                                                                    //The name of the Wi-Fi network you
14 const String pass = "Wifi Password";
                                                                    //Your WiFi network password
15 const long myChannelNumber = 1234566;
                                                                    //Your Thingspeak channel number
16 const char * myWriteAPIKey = "XXXXXXXXXXXXXXXXXX";
                                                                    //Your ThingSpeak Write API Key
```

Setting up your pump

If you do not have a pump attached, you can just skip this part. The code is rather self explanatory. You set what parameters will trigger the pump to engage.

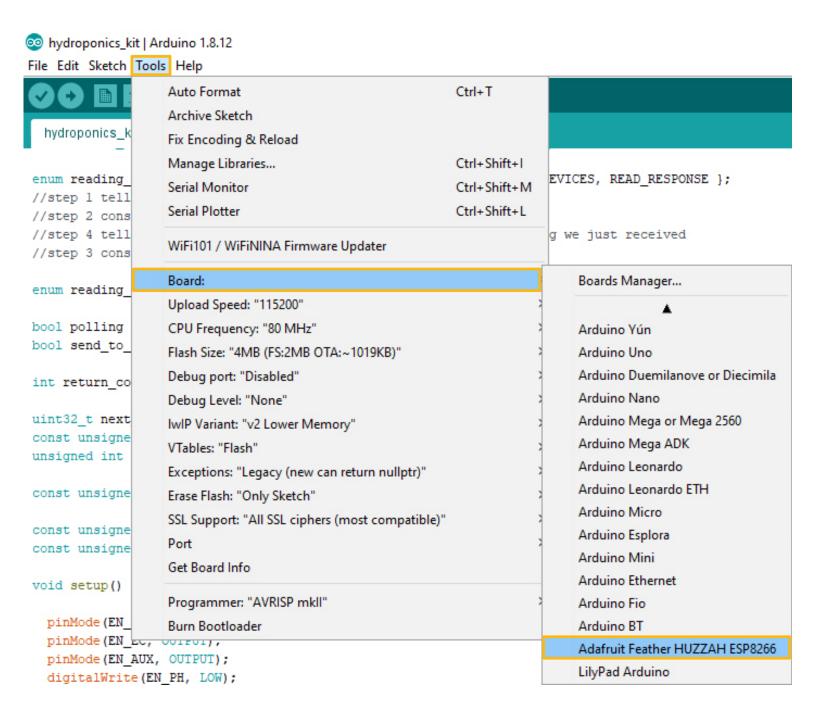
```
48 //parameters for setting the pump output
49 #define PUMP_BOARD PMP //the pump that will do the output (if theres more than one)
                                   //the dose that the pump will dispense
50 #define PUMP DOSE
                          -0.5
                                  //the circuit that will be the target of comparison
51 #define EZO BOARD
                          EC
                                  //true means the circuit's reading has to be greater than the comparison value,
52 #define IS_GREATER_THAN true
53 #define COMPARISON_VALUE 1000
                                  //the threshold above or below which the pump is activated
```



Step 6 Setting up the HUZZAH board

A Set the target CPU to flash

Tools> Board> Adafruit Feather HUZZAH ESP8266

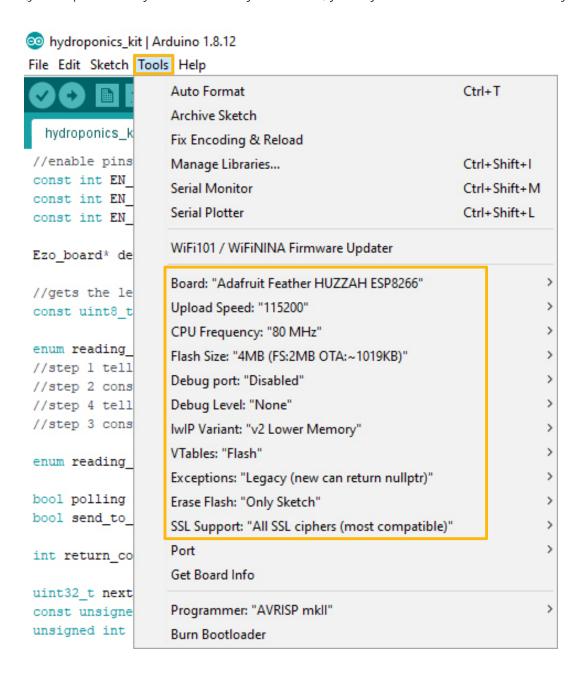




Adjust CPU Settings

Make sure the CPU settings on the Adafruit Feather HUZZAH ESP8266 are correct. To adjust the CPU settings, click **Tools**.

For reference, this is what Atlas Scientific set the CPU settings to. (your options may not be exactly the same, just try and match them as closely as possible.)

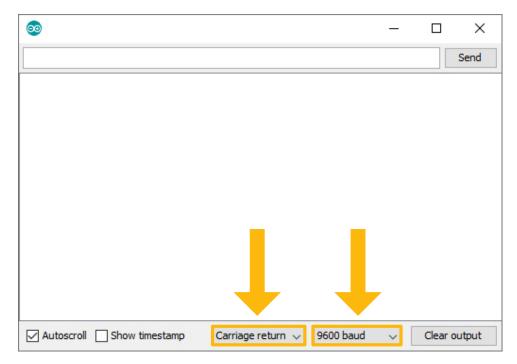




Step 7 See the readings

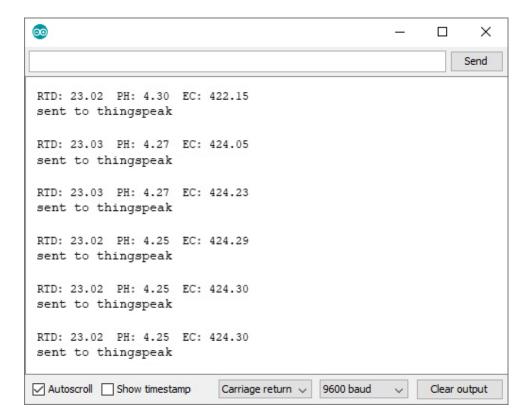
Open your Arduino serial monitor.

(You must have the serial monitor set to the comport from the Adafruit Feather HUZZAH)



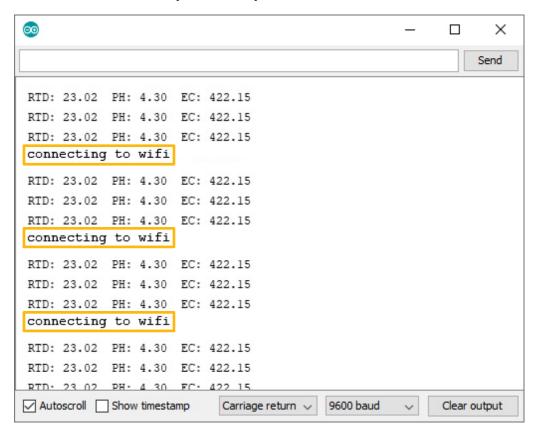
Set to carriage return and 9600 baud.

The Wi-Fi Hydroponics Meter will always attempt to connect to ThingSpeak on bootup.





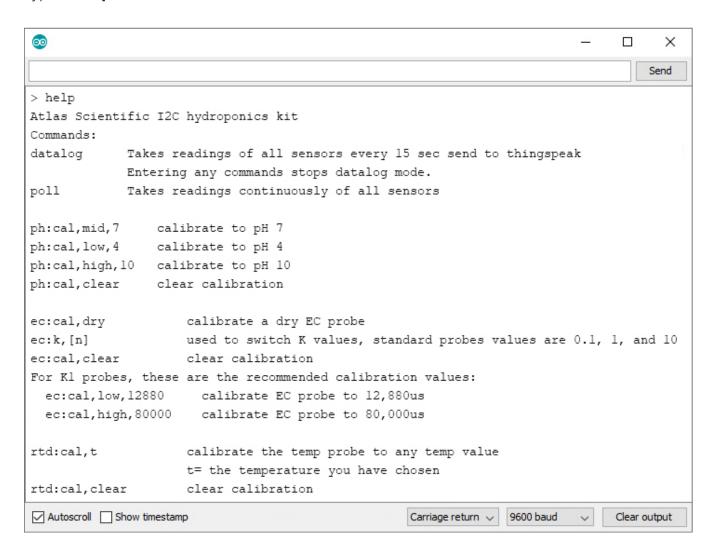
If it cannot connect to your Wi-Fi you will see this:



Entering the poll command will stop the Wi-Fi Hydroponics Meter from uploading the readings to thingspeak, while you debug your Wifi problems.

Step 8 Sensor Calibration

Atlas Scientific created a list of calibration commands that are built into the library. Type in **help** to see a list of commands.



The poll command

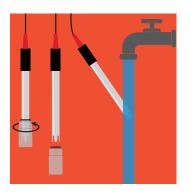
Send the command **poll**; This will let you see the readings once per second and it will stop uploading to ThingSpeak while you calibrate.



Calibrate pH

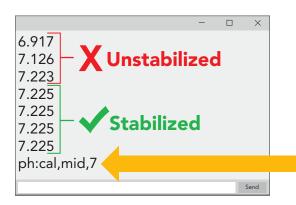
When calibrating pH, you must always calibrate to pH 7 first.

Remove the soaker bottle and rinse off the pH probe. Remove the top of the pH 7.00 calibration solution pouch. Place the pH probe inside the pouch and let the probe sit in the calibration solution until the readings stabilize. This will take about 1-2 mins.









Once the readings have stabilized, issue the Mid point calibration command. ph:cal,mid,7

Rinse off the probe and repeat this process for both **pH 4.00** and **pH 10.00**.

C Calibrate Conductivity

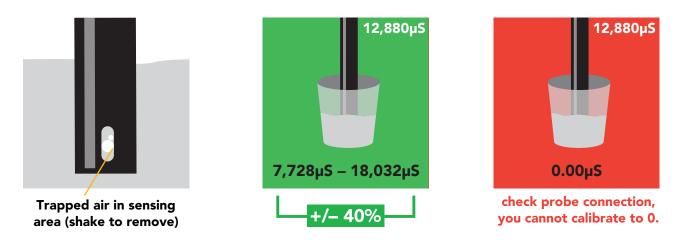
When calibrating Conductivity, you must always calibrate a dry probe first.

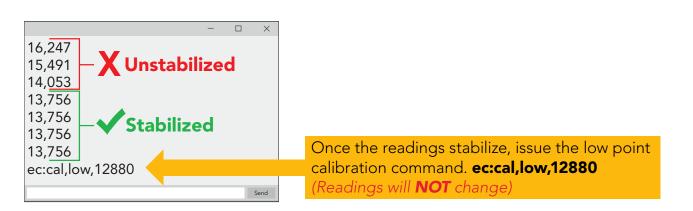


Make sure that the probe is dry before issuing this command, ec:cal,dry

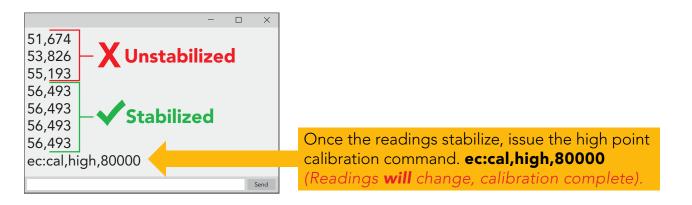


Once the dry calibration has been completed, place the probe into a small cup of the 12,880µS calibration solution. Shake the probe to make sure you do not have trapped air bubbles in the sensing area. You should see readings that are off by 1 - 40% from the stated value of the calibration solution. Wait for readings to stabilize.





Rinse off the probe before calibrating to the high point. Pour a small amount of the 80,000µS calibration solution into a cup. Shake the probe to remove trapped air. Again, the readings may be off by 1 - 40% Wait for readings to stabilize.

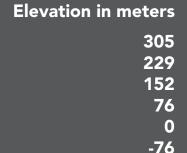




Calibrate Temperature

Calibrating the PT-1000 temperature probe is not required. However, if you want to, a simple method to calibrate the probe is to place the PT-1000 into boiling water. Then issue command rtd:cal,t

100 °C



-152

Boiling point °C 98.9 °C 99.2 °C 99.5 99.7 °C 100 °C 100.3 °C 100.5 °C

Calibration Complete



Step 9 Almost done!

Once you are finished with calibration, issue the datalog command to resume taking a reading every 15 seconds and uploading it to thingspeak.

To see the data on your phone, download the ThingSpeak app.



Setup Complete!

