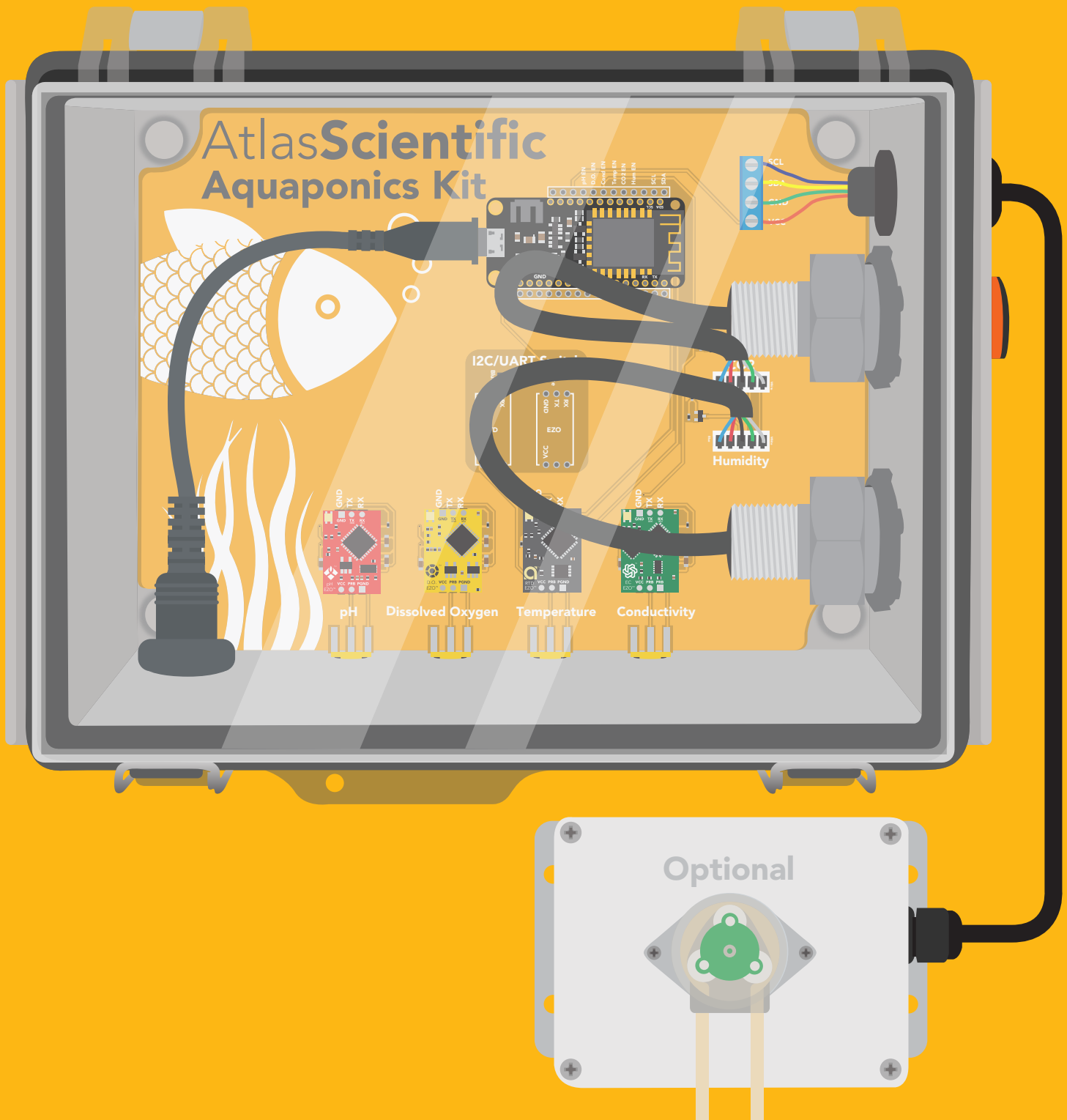
Atlas**Scientific**™
Environmental Robotics
V 1.1

Wi-Fi Aquaponic Kit

Datasheet

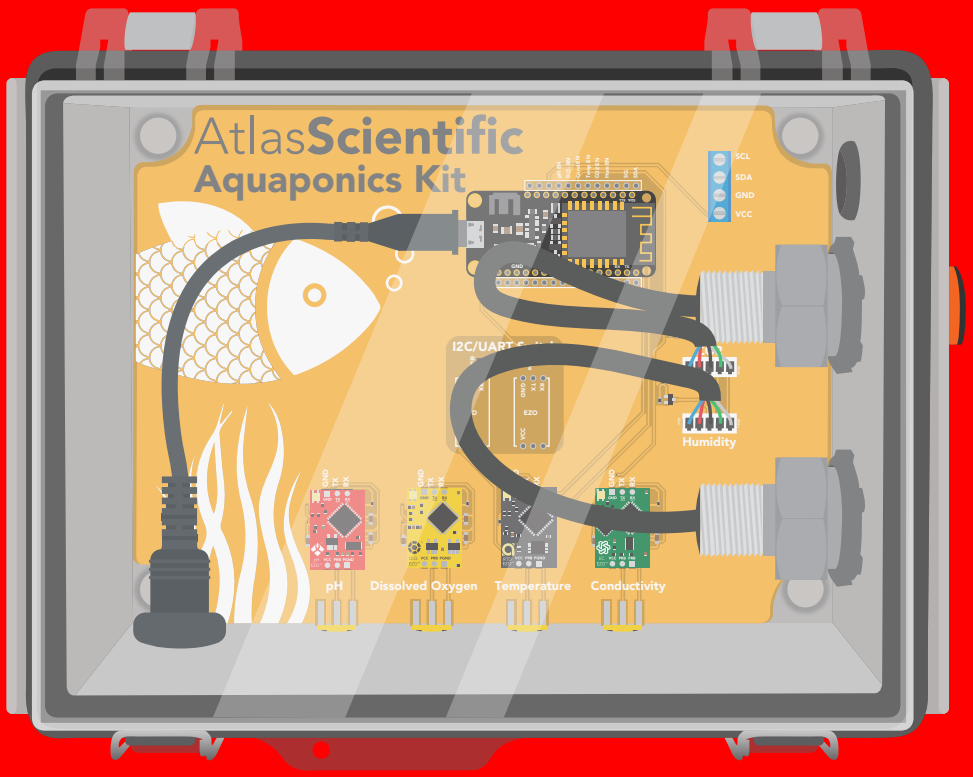


STOP

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)

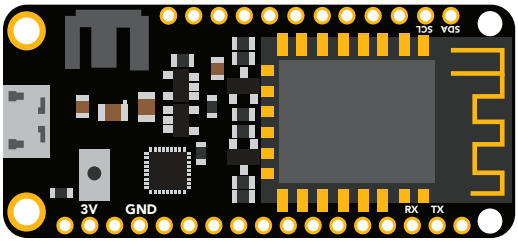
Operating principle

The Wi-Fi Aquaponics kit has been designed to provide the engineer with a simple way of remotely monitoring and controlling an aquaponics system's chemistry. Sensor data is uploaded to ThingSpeak ™, a free, cloud-based data acquisition and visualization platform. The Wi-Fi Aquaponics 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

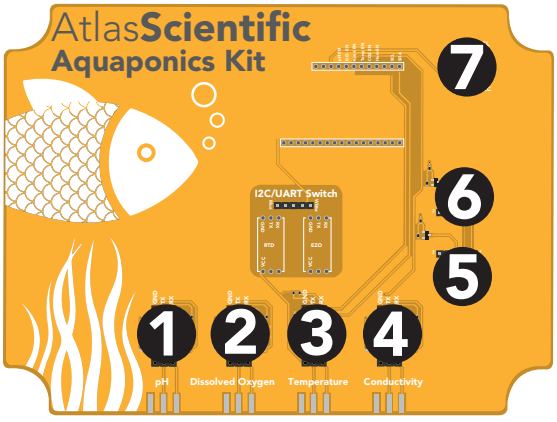
The Wi-Fi Aquaponics kit is controlled using an Adafruit HUZZAH32 as its CPU. The HUZZAH is programmed using the Arduino IDE and uses an onboard ESP32 as its Wi-Fi transmitter. Adafruit HUZZAH32 datasheet.



Sensor ports

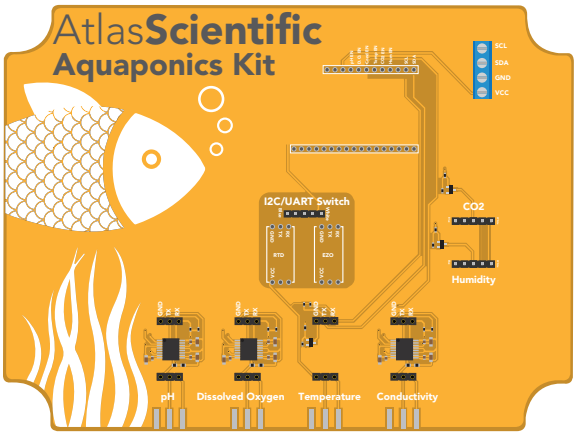
The Wi-Fi Aquaponics kit PCB has 7 sensor ports. Three of the ports are electrically isolated. The isolated ports are marked pH, Dissolved Oxygen, and Conductivity. The isolated ports are needed to take noise-free electrochemical readings. Because the sensing element of a temperature sensor is never in direct contact with the water, electrical isolation is not needed for temperature sensing.

Port 5 and 6 are marked Humidity and CO2. The terminal block marked Port 7 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.

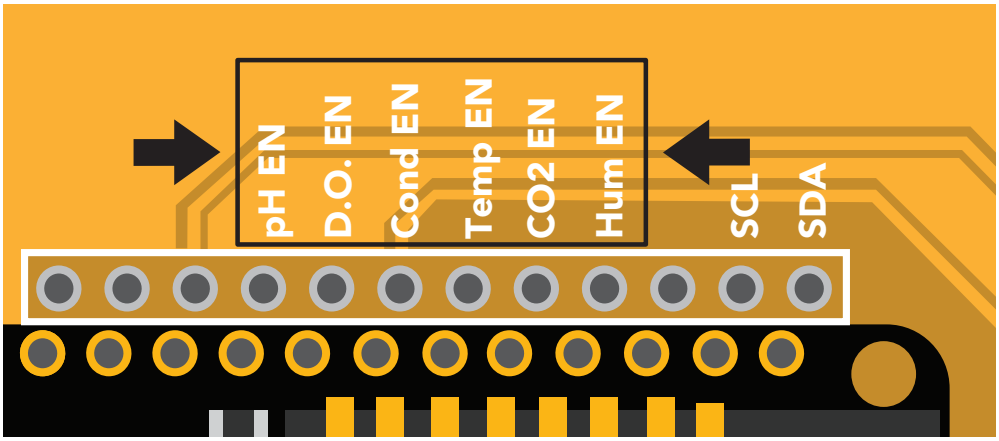


PCB

The overall design of the PCB is quite simple. The CPU is powered and programmed through the panel-mount USB connector. The CPUs USB pin supplies the board's power bus with 5V.



Each of the six 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, D.O and Cond) must be set low to power on the sensors. While the Temp, CO2 and Hum pins must be set high to power on the sensors.

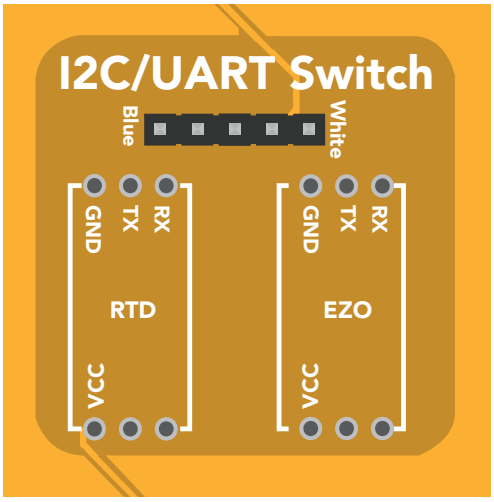
Truth table

Pin	State	Sensor Power
pH EN	LOW	ON
D.O. EN	LOW	ON
Cond EN	LOW	ON
Temp EN	HIGH	ON
CO2 EN	HIGH	ON
Hum EN	HIGH	ON

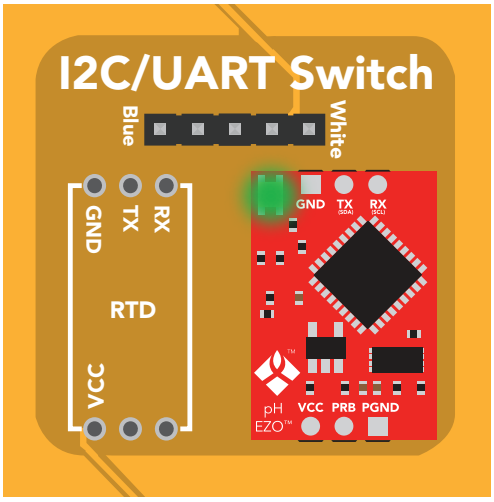
Sensor port 7 *(the terminal block)* does not have an enable pin and can not be turned off.

On Board I2C/UART Switch

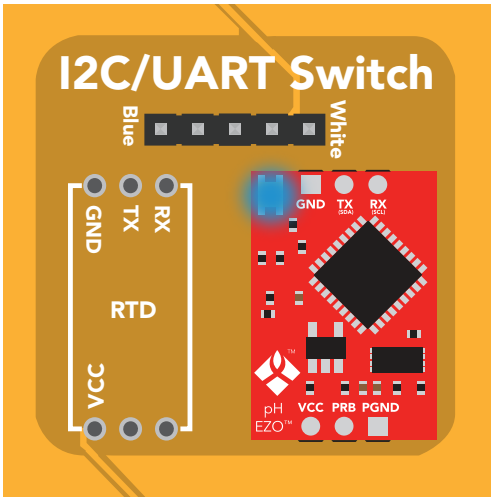
The PCB has a built in protocol toggler which can be used to switch each of the sensors between I2C/UART modes.



Place an EZO circuit onto the I2C/UART switch pins.



The EZO circuit in UART mode...



...is now in I2C mode.

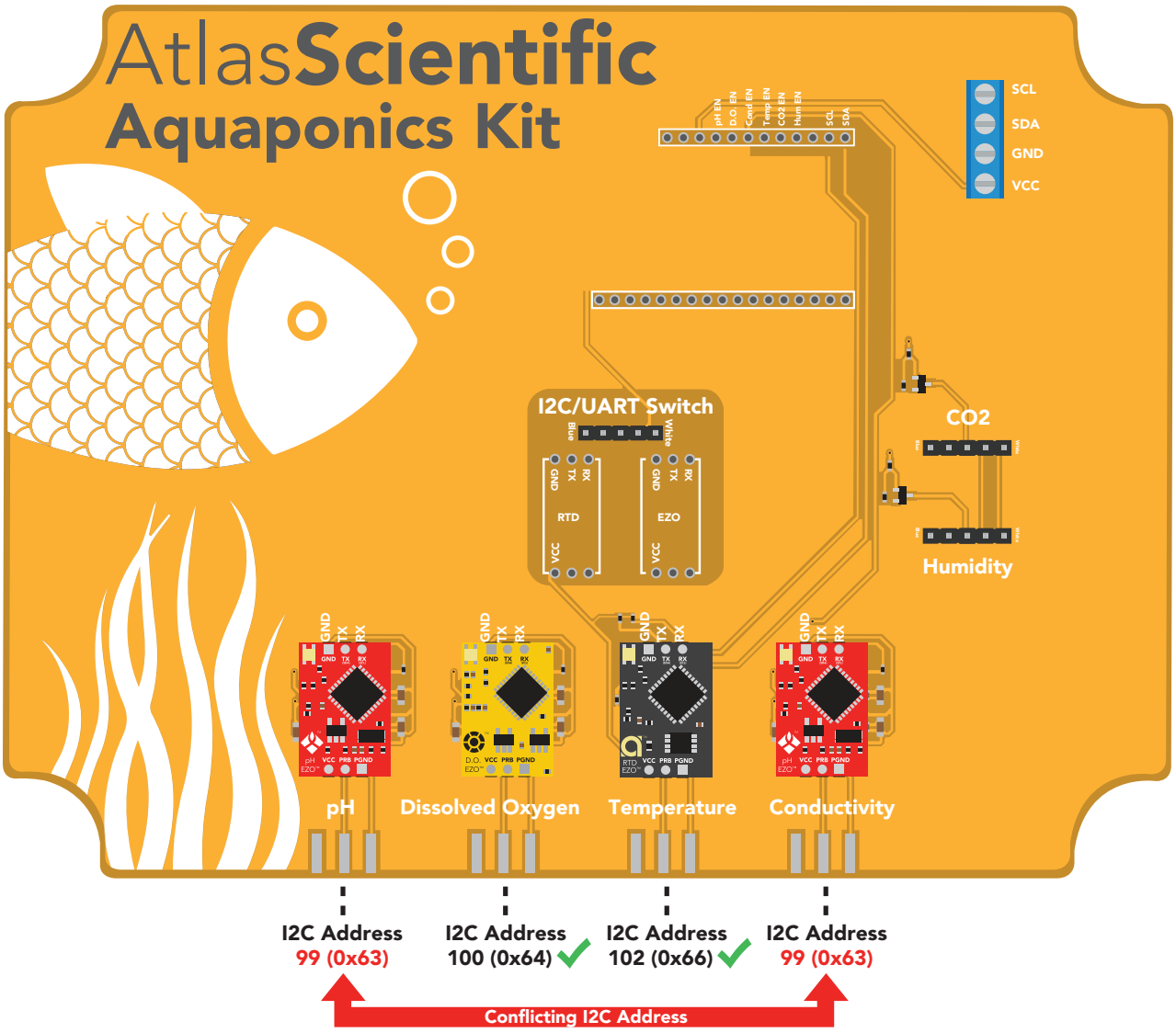
Data protocol

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, or use the on board protocol toggler mentioned above.

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.

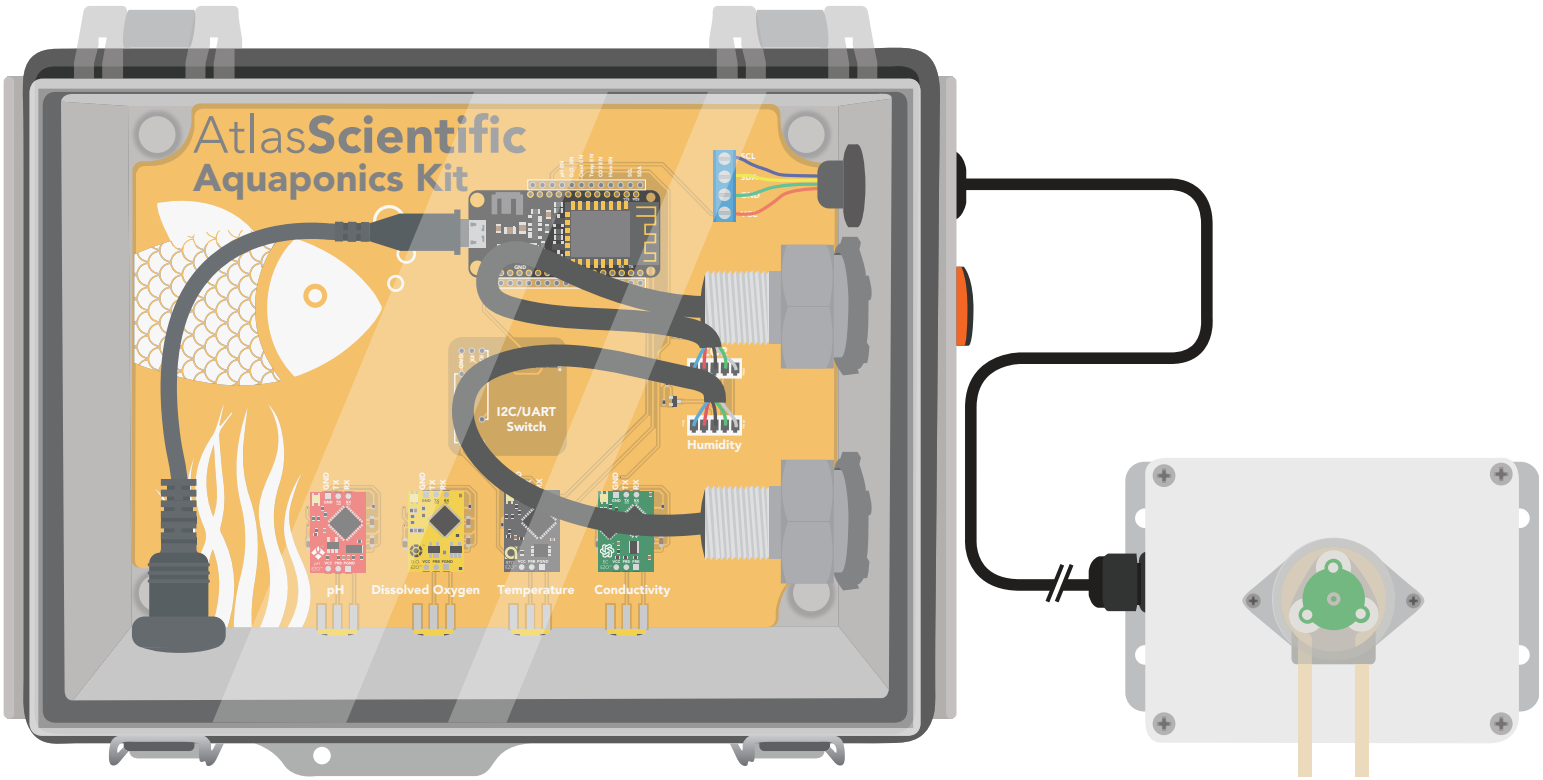
Device	I2C Address
EZO pH	99 (0x63)
EZO ORP	98 (0x62)
EZO DO	97 (0x61)

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

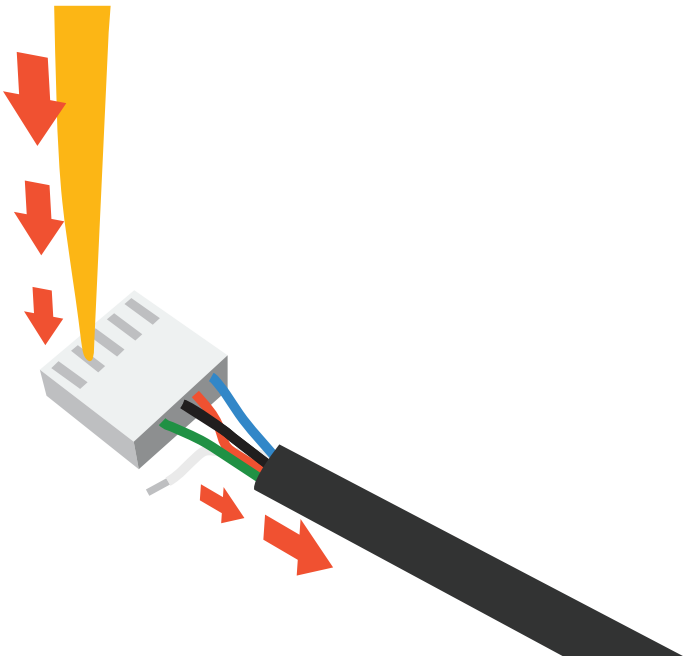
Device	I2C Address
EZO CO2	105 (0x69)
EZO HUM	111 (0x6F)
EZO PMP-L	109 (0x6D)

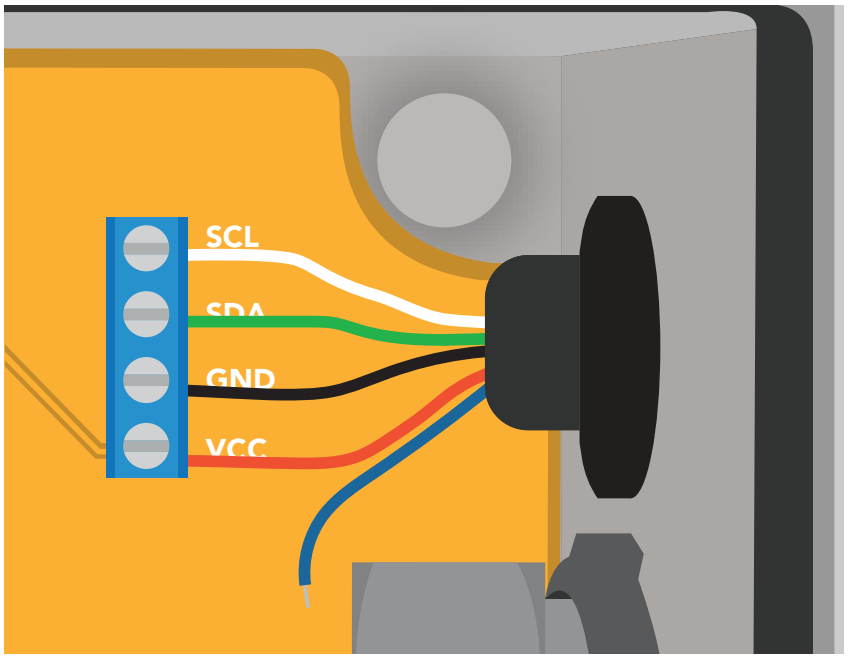
Dosing pump

An optional external dosing pump can be added to the Wi-Fi Aquaponics kit. Using the SGL-PMP-BX 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

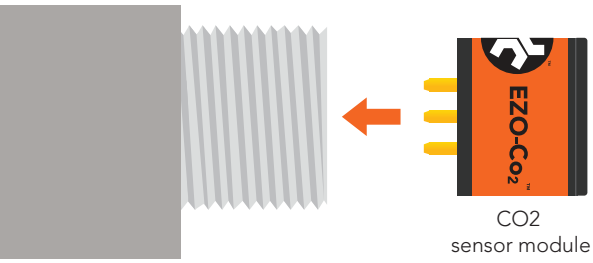
The Atlas-Scientific Wi-Fi hydroponics kit has been designed to upload sensor data to ThingSpeak™, a free, cloud-based data acquisition and visualization platform. You will be required to set up a free account with ThingSpeak™ to upload and visualize the data. With a free account, you can upload data 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 Connect Co2 sensor module

The CO2 sensor module has been removed to protect it during transport. Before you begin setting up the aquaponics kit, plug it back into position.

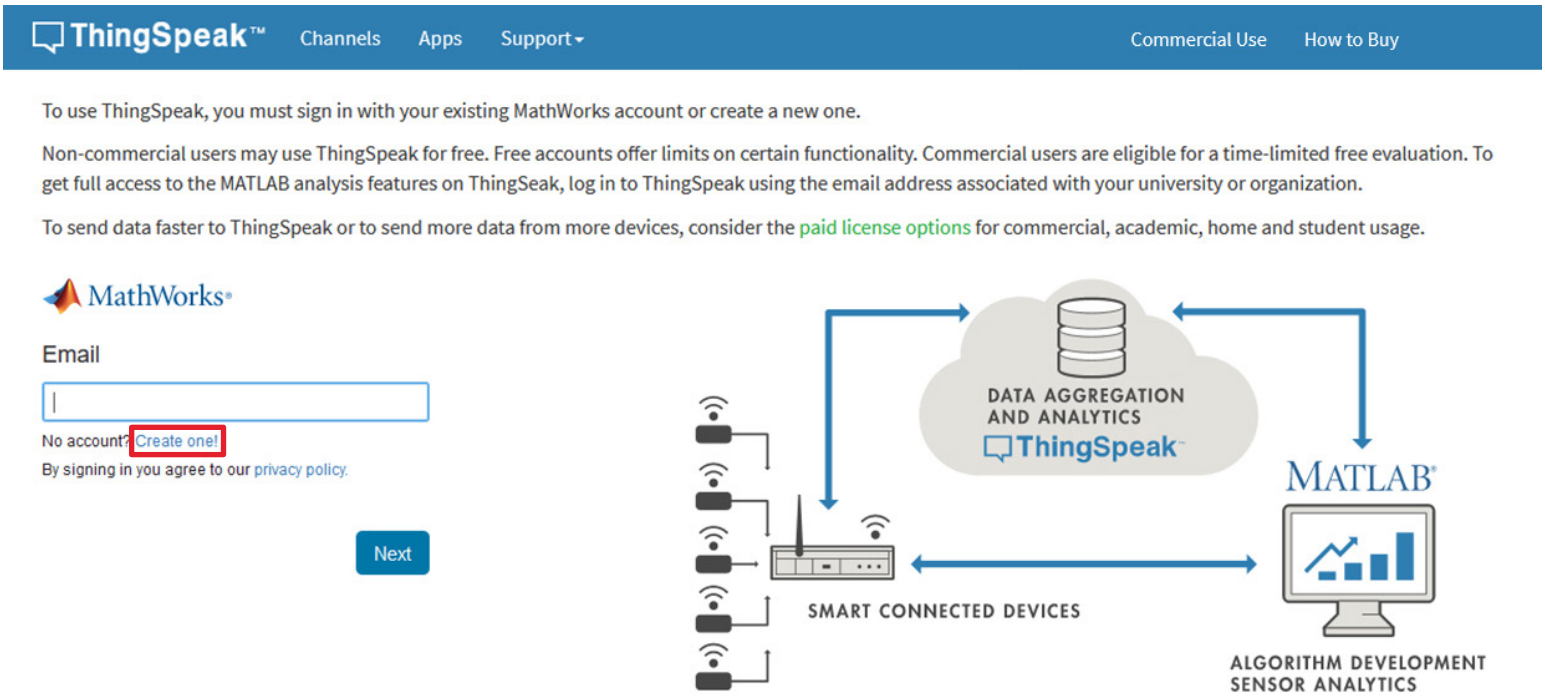


CO2
sensor module

Warning:
Do not mix up sensor modules as each is calibrated to a specific EZO-CO2 device.

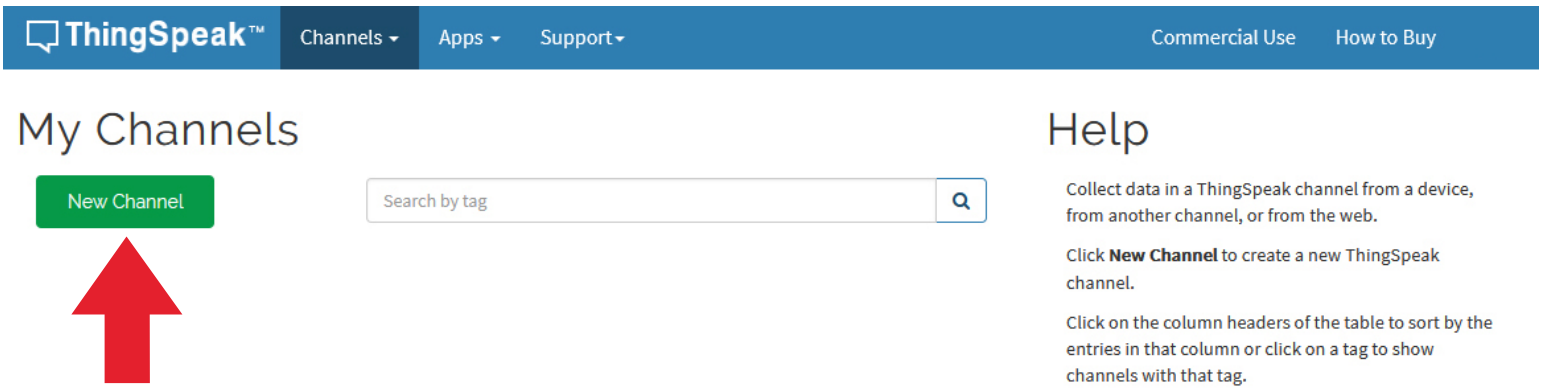
Step 2 Setup a ThingSpeak Account

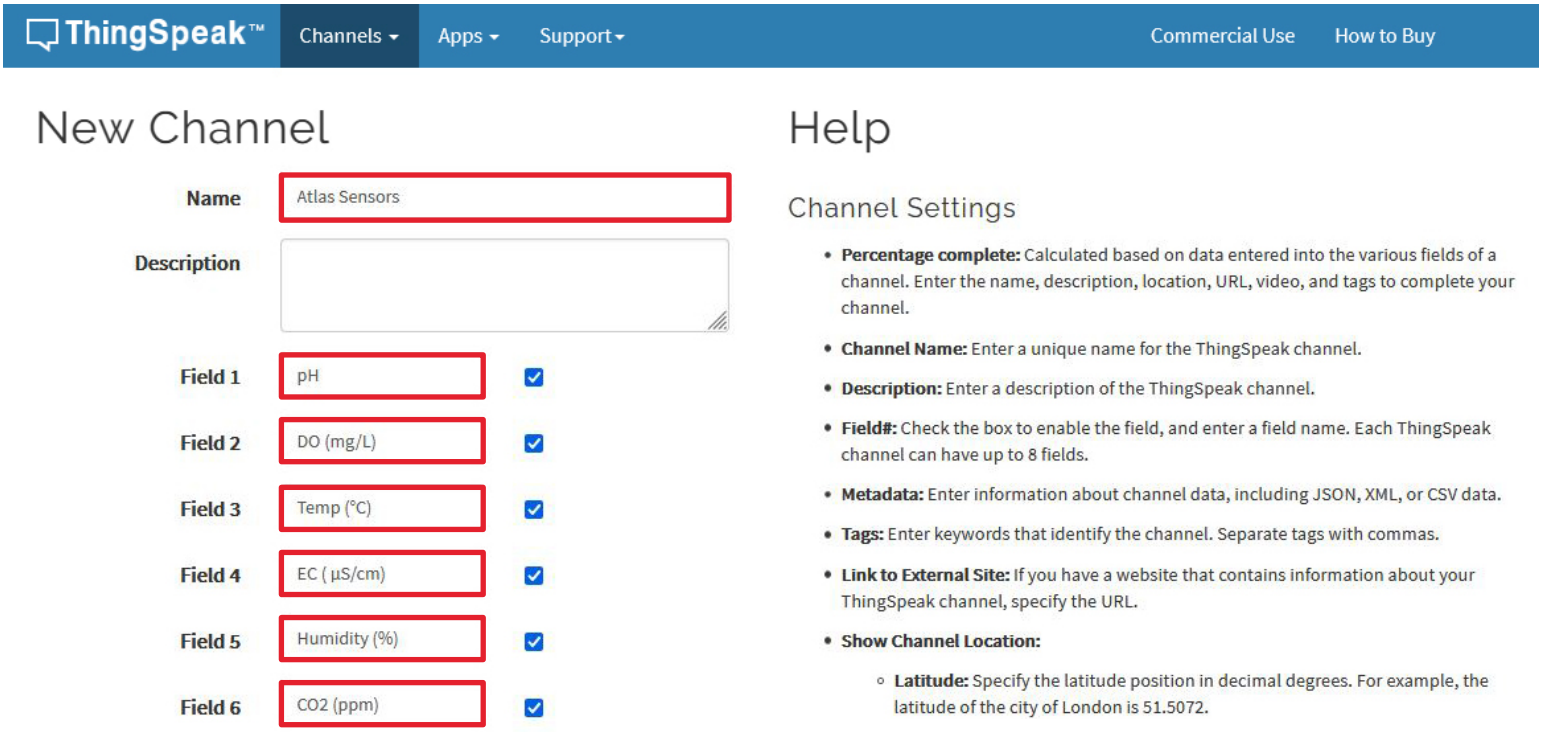
Because the sensor data is stored / viewed on ThingSpeak, you will need to setup a ThingSpeak account. Create your ThingSpeak account by clicking HERE.



Step 3 Create a Channel

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





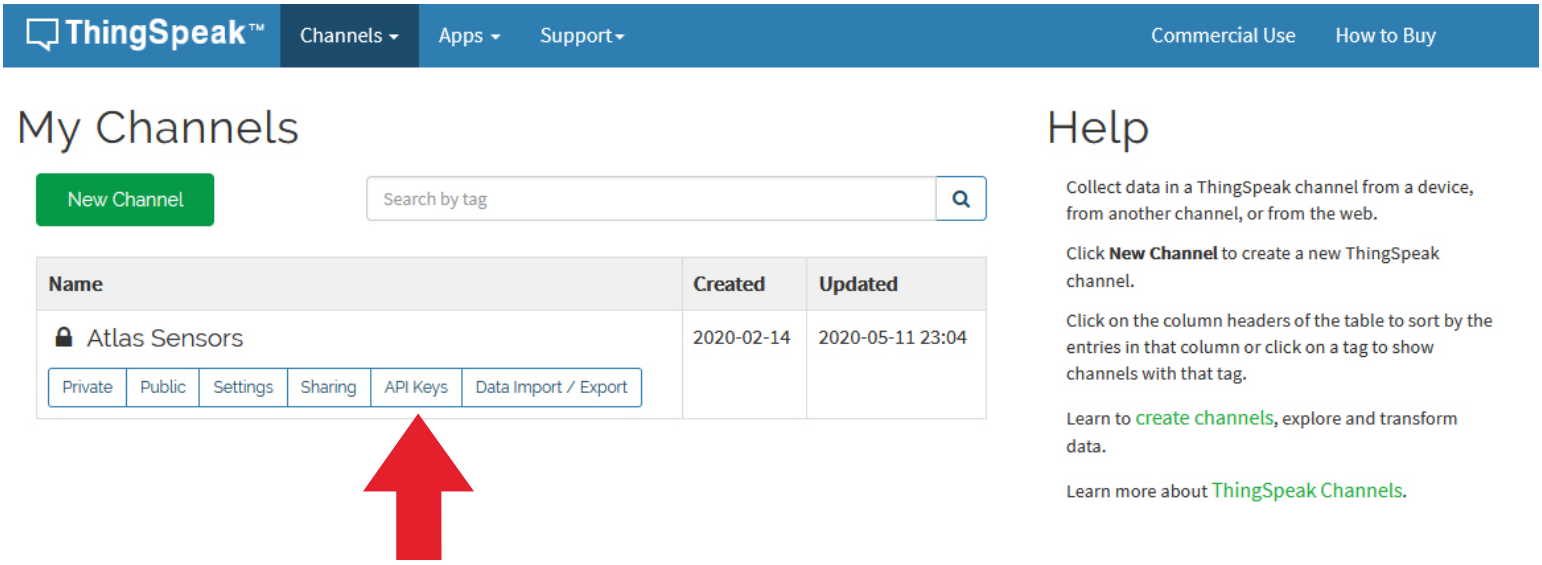
Fill out the highlighted boxes. (Be sure to click on the checkboxes to enable **fields 2 – 6**)
For reference, this is what we entered.

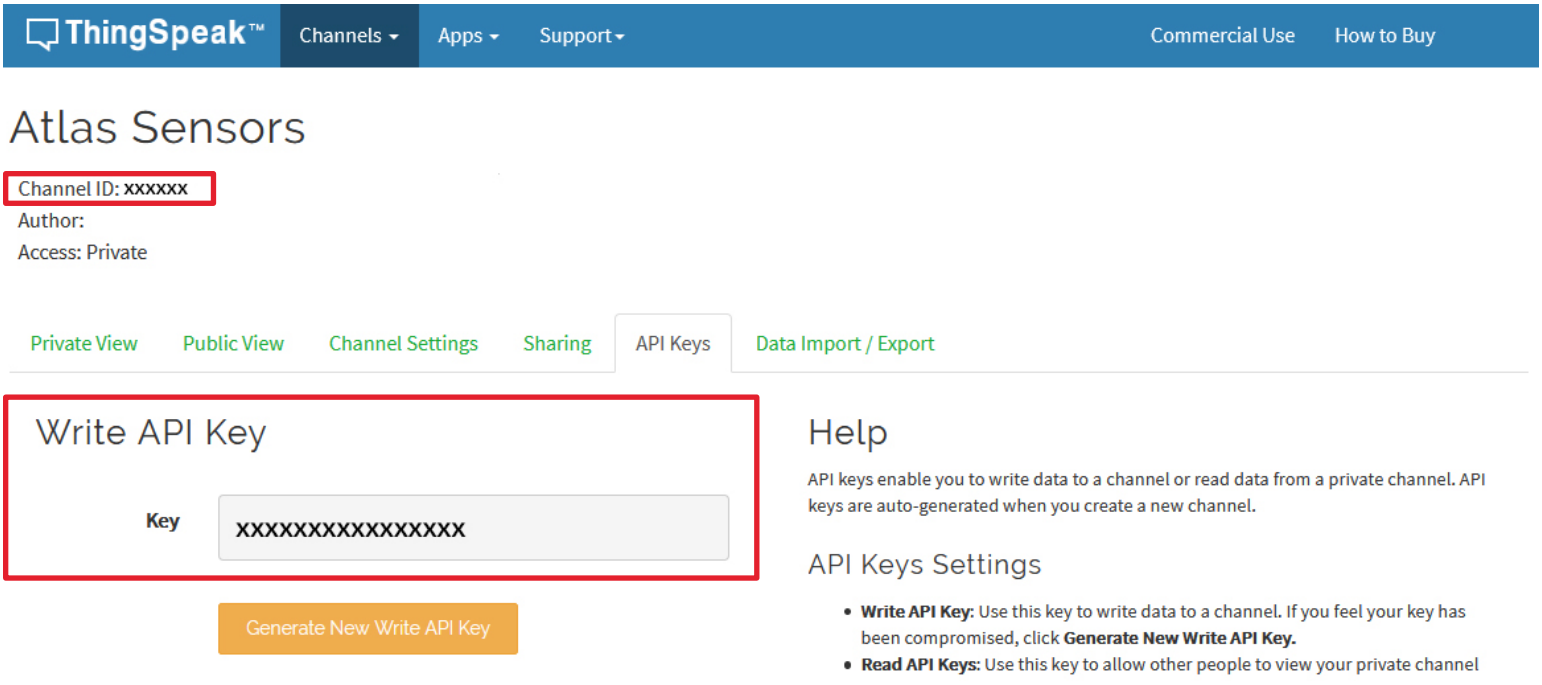
Name **Atlas Sensors**
Field 1 **pH**
Field 2 **DO (mg/L)**
Field 3 **Temp (°C)**
Field 4 **EC (µS/cm)**
Field 5 **Humidity (%)**
Field 6 **CO2 (ppm)**

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

Step 4 Get ThingSpeak API keys

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



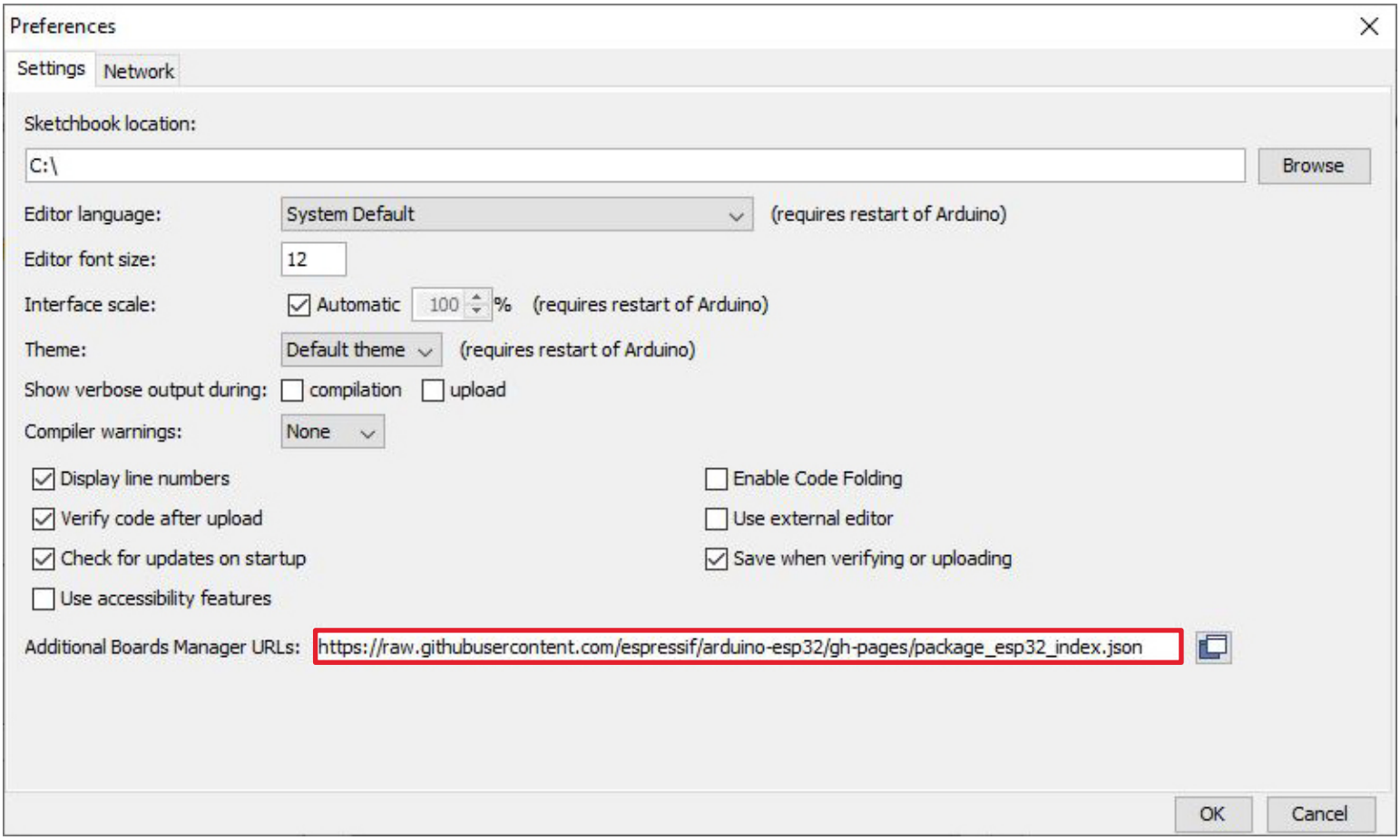


Be sure to save your **Channel ID** and **Write API Key** we are going to need these, in the next few steps.

Step 5 Make sure your Arduino IDE libraries are up to date

A Make sure you have the correct path for the Esp32 Library

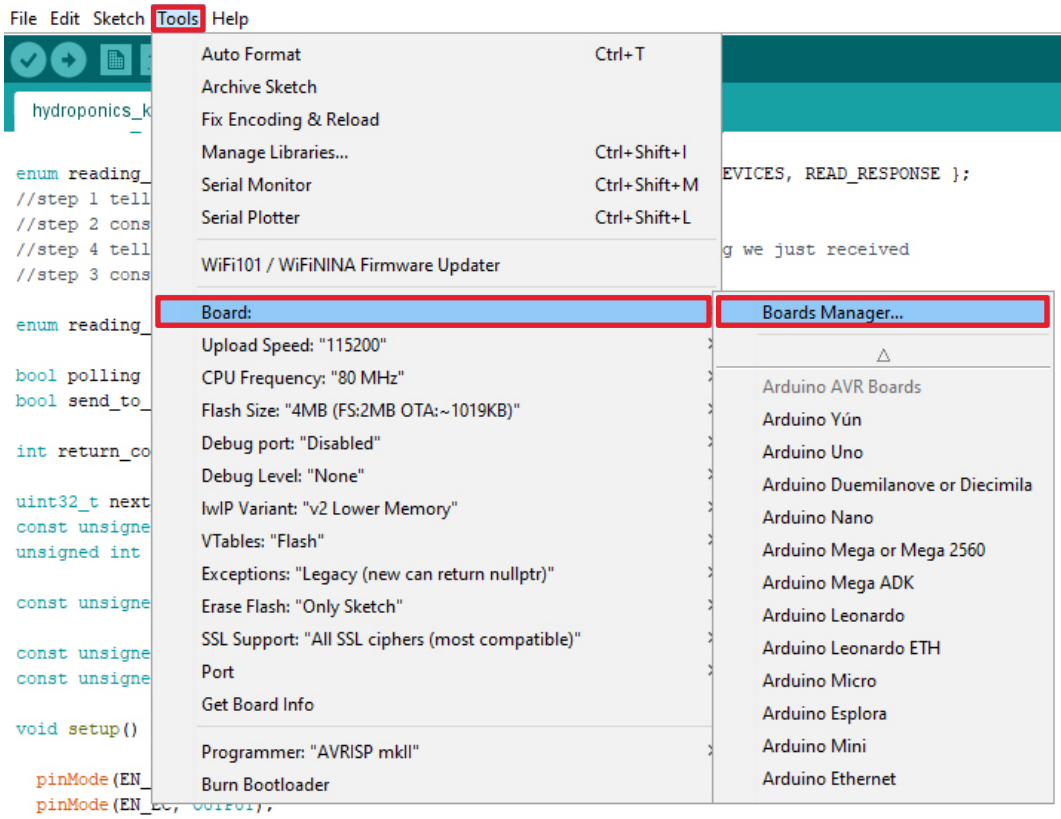
In the IDE, go to **File > Preferences**
Locate the **Additional Boards Manager URLS** text box.

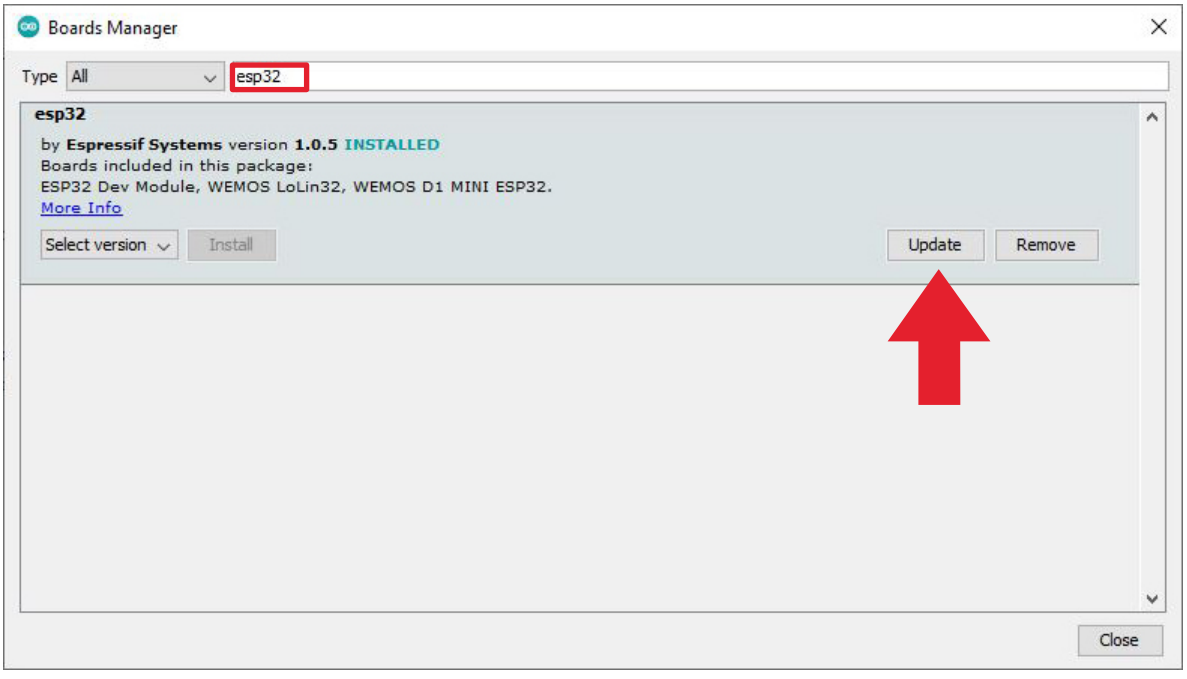


Make sure this URL is in the textbox
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
Click **OK**.

B Update the Esp32 board

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





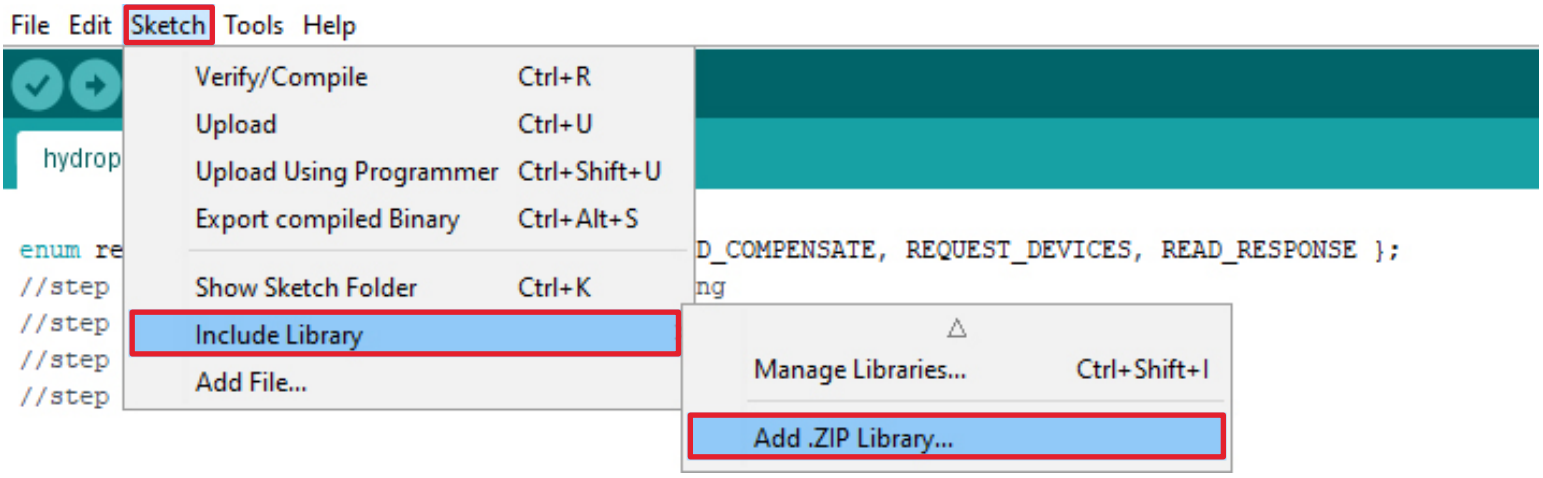
In the search bar of the Boards Manager, lookup **esp32.**
Update to the most recent version if you don't already have it.

(Version 1.0.5 in not the most recent version)

C Download the ThingSpeak library for Arduino

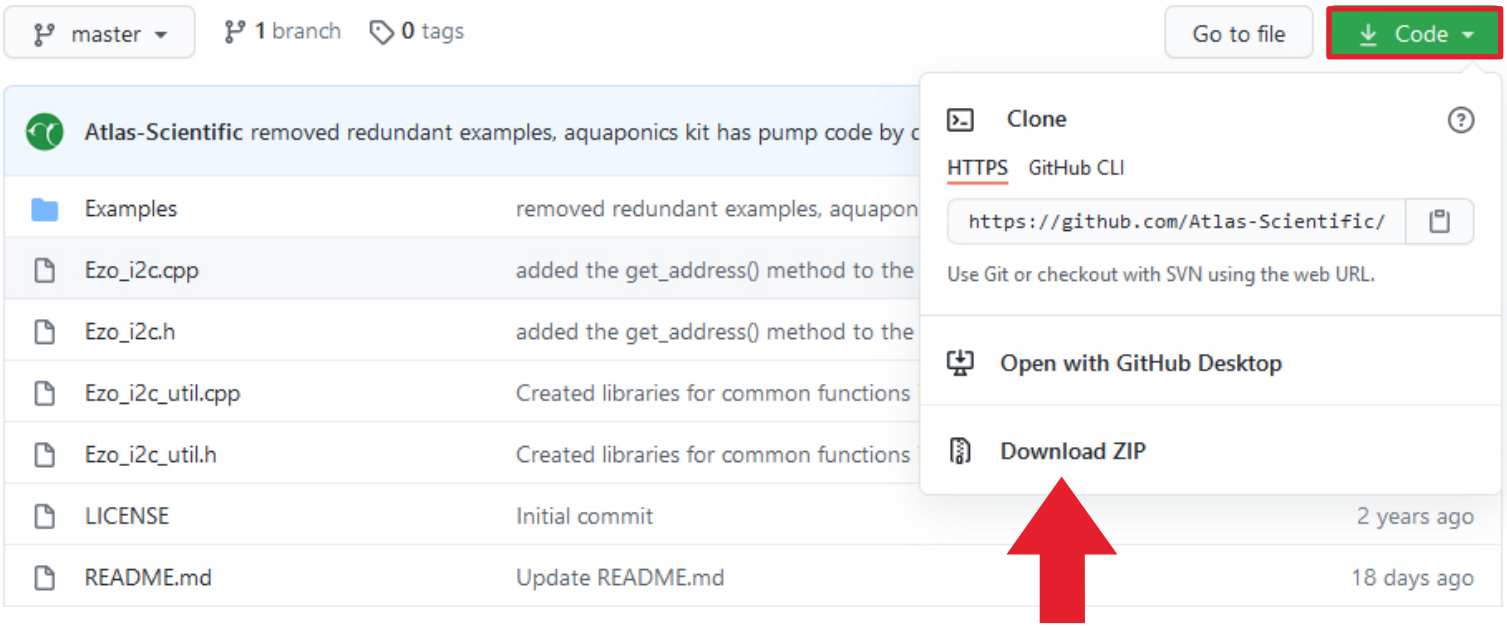
Click HERE to download the latest version of the ThingSpeak library.

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**



D Add the EZO I2C Library

To download the Ezo_I2c library file, click HERE.

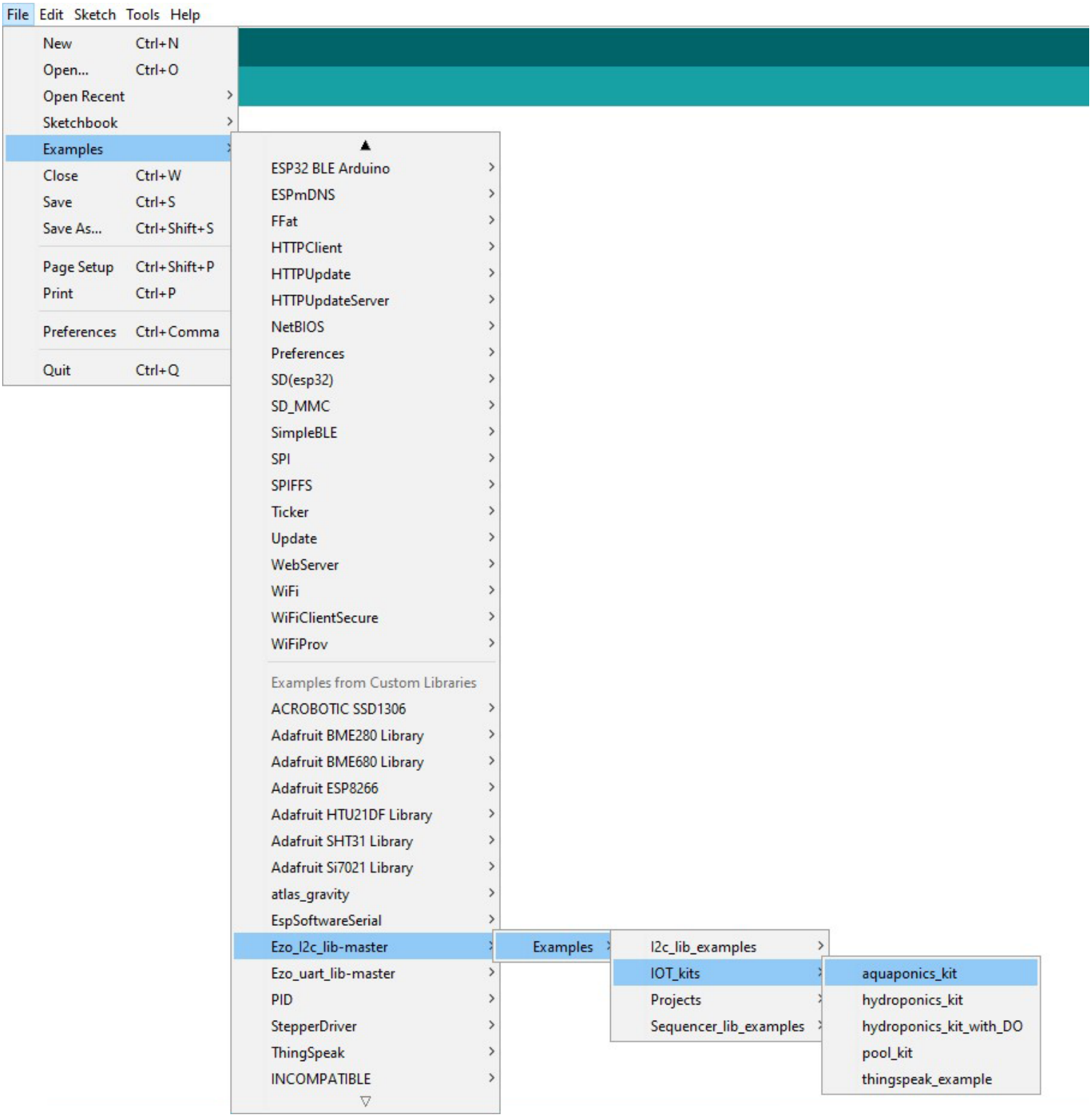


Don't unzip it!
Import the .ZIP file to your Arduino IDE.
To import the .ZIP file go to **Sketch > Include Library > Add .ZIP Library**

Step 6 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> aquaponics_kit



AtlasScientific Environmental Robotics

B 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)*

```
#include <iot_cmd.h>
#include <WiFi.h>                                          //include wifi library
#include "ThingSpeak.h"                                    //include thingspeak library
#include <sequencer4.h>                                    //imports a 4 function sequencer
#include <sequencer1.h>                                    //imports a 1 function sequencer
#include <Ezo_i2c_util.h>                                  //brings in common print statements
#include <Ezo_i2c.h> //include the EZO I2C library from https://github.com/Atlas-Scientific/Ezo_I2c_lib
#include <Wire.h>    //include arduinos i2c library

WiFiClient client;                                        //declare that this device connects to

//----------------Fill in your Wi-Fi / ThingSpeak Credentials-------
const String ssid = "Wifi Name";                           //The name of the Wi-Fi network you
const String pass = "Wifi Password";                       //Your WiFi network password
const long myChannelNumber = 1234566;                      //Your Thingspeak channel number
const char * myWriteAPIKey = "XXXXXXXXXXXXXXXX";          //Your ThingSpeak Write API Key
//------------------------------------------------------------------
```

C Setting up your pump

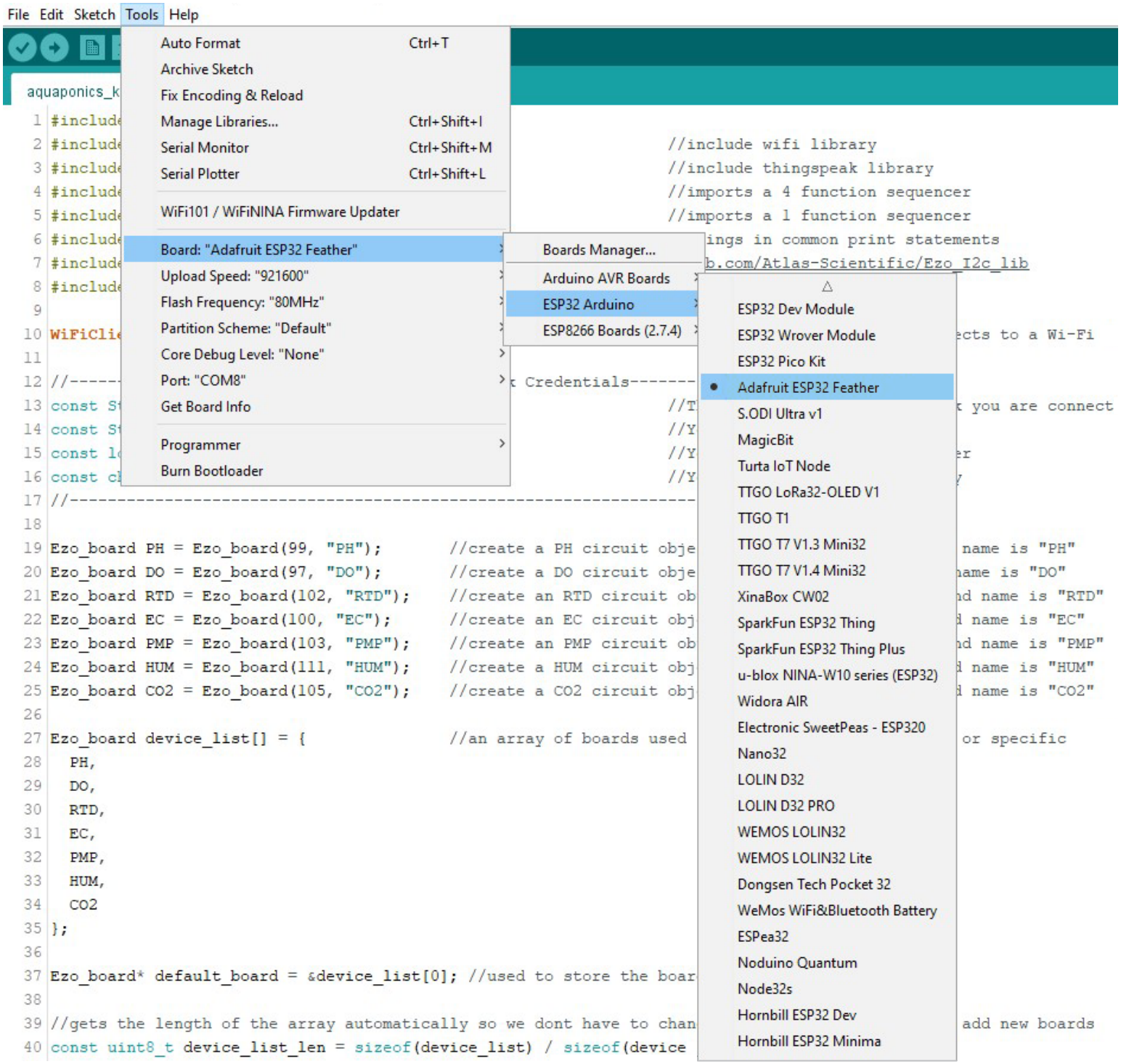
If you do not have a pump attached, you can just skip this part. The code is rather self explanatory. You set what parameters will trigger the pump to engage.

```
//parameters for setting the pump output
#define PUMP_BOARD        PMP       //the pump that will do the output (if theres more than one)
#define PUMP_DOSE         -0.5      //the dose that the pump will dispense
#define EZO_BOARD         EC        //the circuit that will be the target of comparison
#define IS_GREATER_THAN   true      //true means the circuit's reading has to be greater than the comparison
#define COMPARISON_VALUE  1000      //the threshold above or below which the pump is activated
```


Step 7 Setting up the HUZZAH board

A Set the target CPU to flash

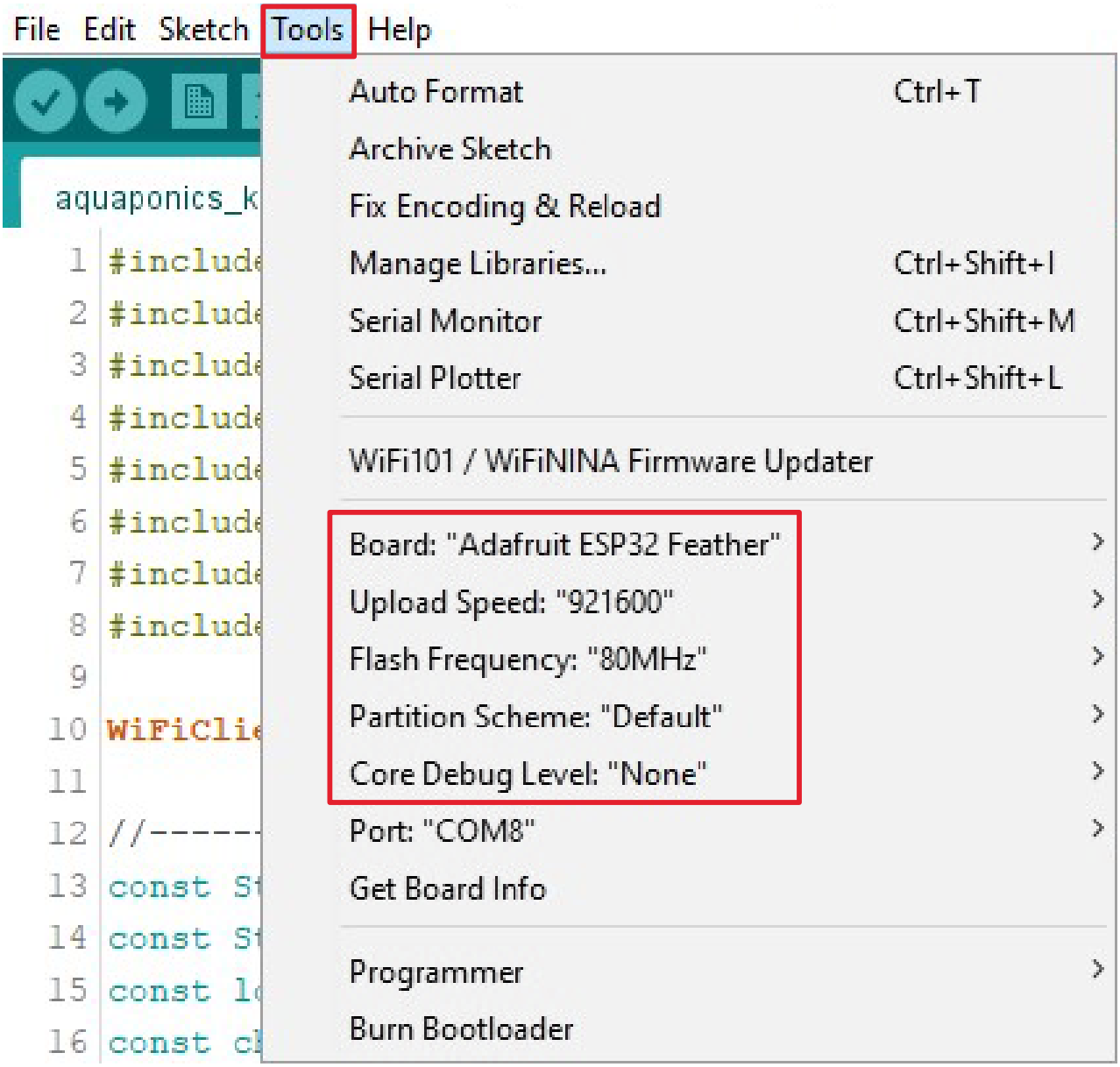
Tools> Board> ESP32 Arduino > Adafruit Esp32 Feather



B Adjust CPU Settings

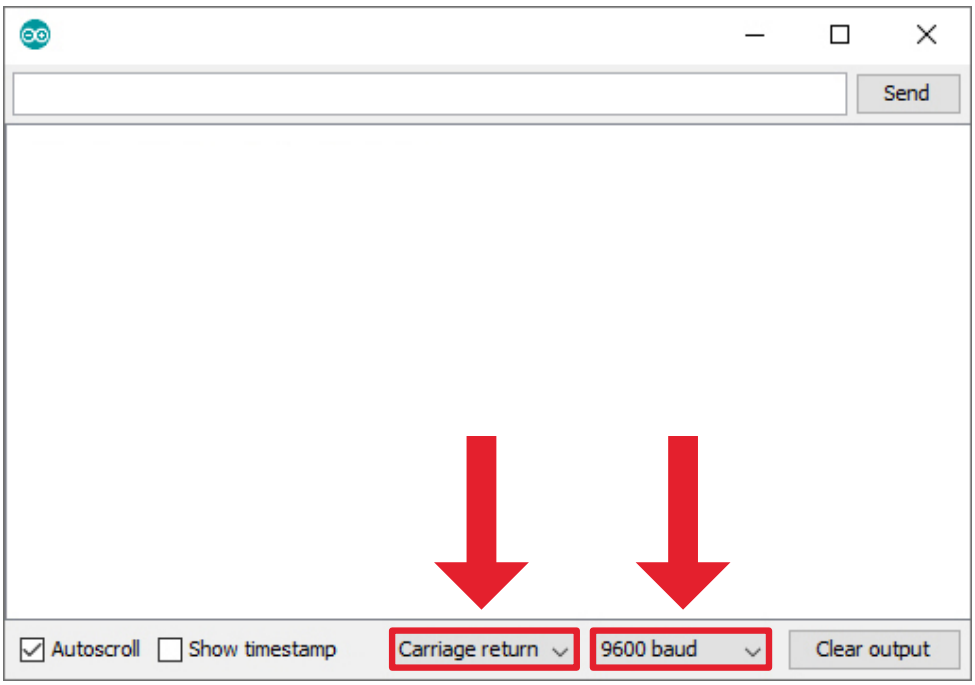
Make sure the CPU settings on the Adafruit HUZZAH32 are correct.
To adjust the CPU settings, click **Tools**.

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.)



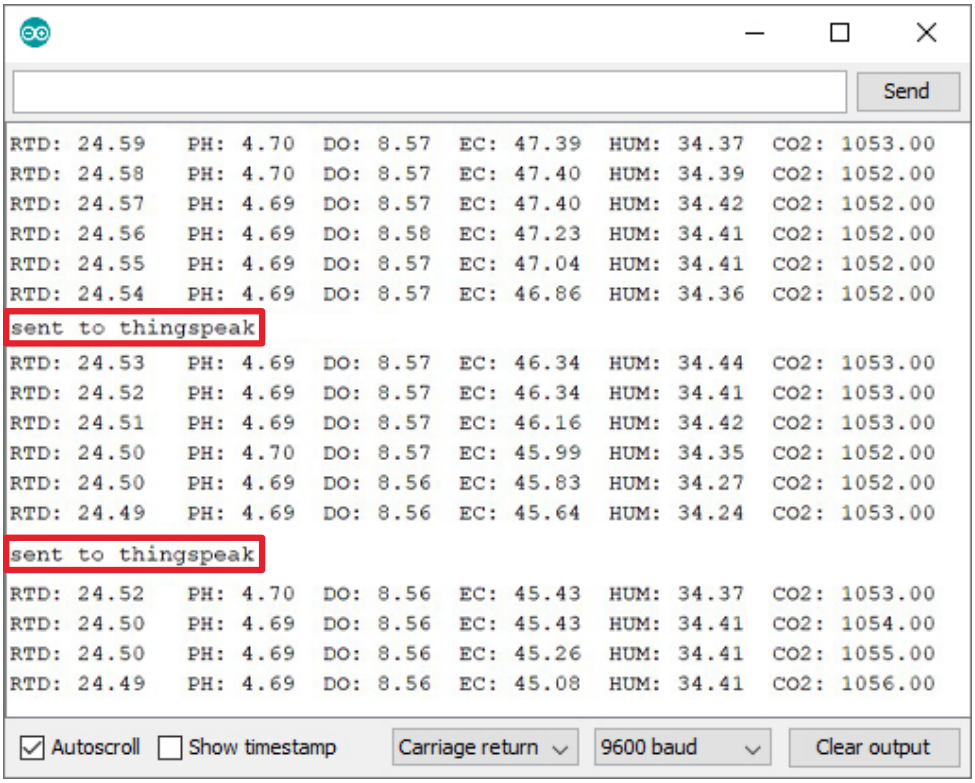
Step 8 See the readings

Open your Arduino serial monitor.
(You must have the serial monitor set to the com port from the Adafruit HUZZAH32)

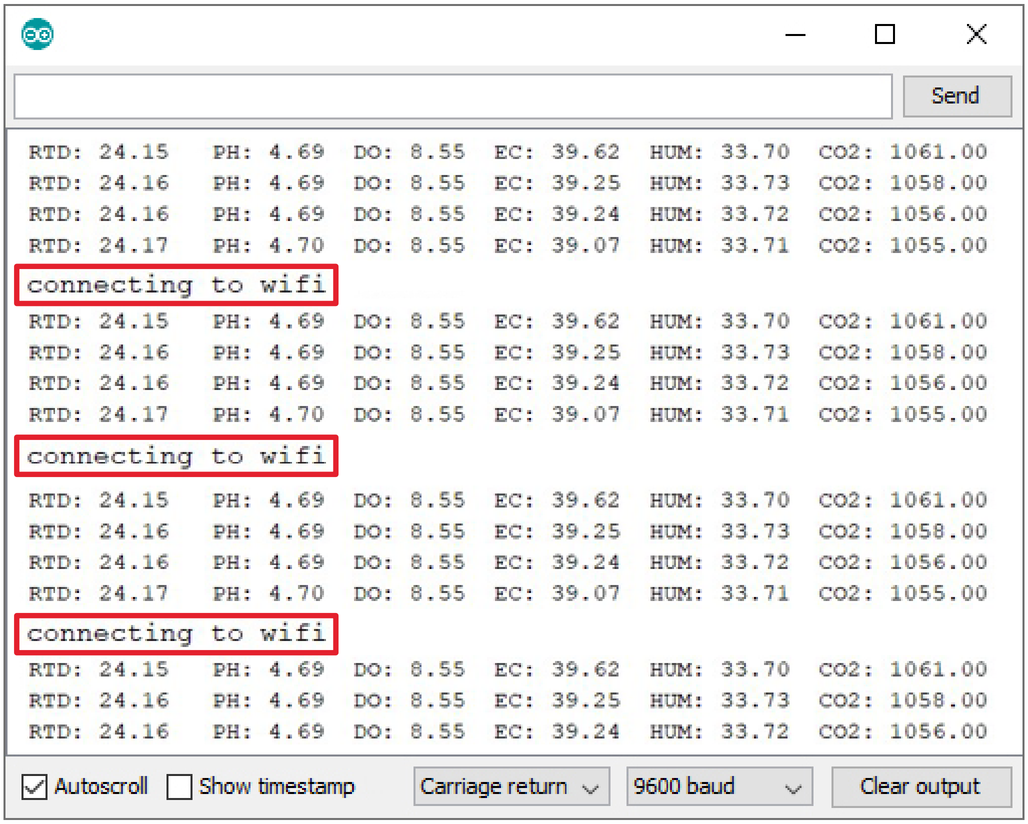


Set to **carriage return** and **9600 baud.**

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



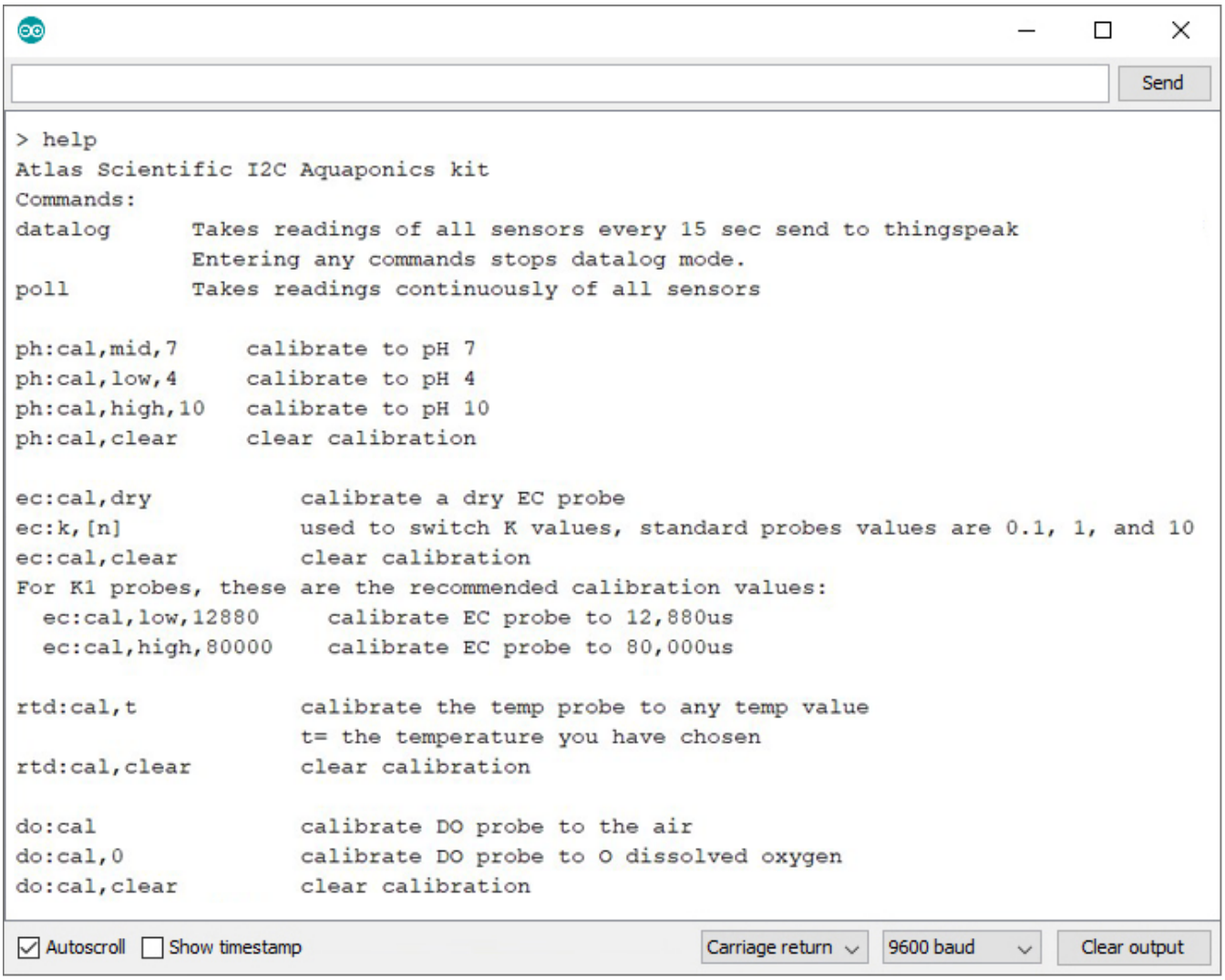
If it cannot connect to your Wi-Fi you will see this:



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

Step 9 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.



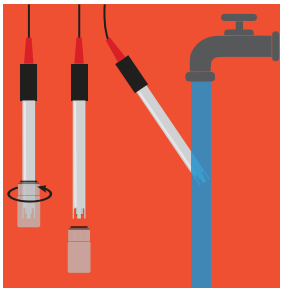
A The poll command

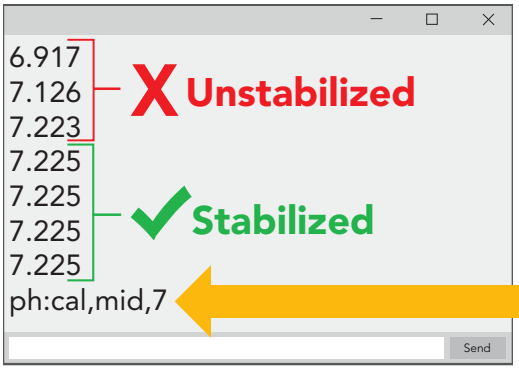
Send the command **poll**; This will let you see the readings once per second and it will stop uploading to ThingSpeak while you calibrate.

B 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**

After 20 mins, the calibration solution inside an open pouch is no longer considered accurate.

Dispose of the unused solution, after calibration.

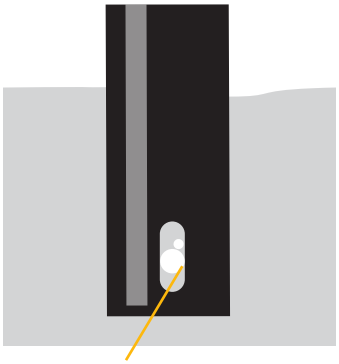
Rinse off the probe and repeat this process for both **pH 4.00** and **pH 10.00**.

C Calibrate Conductivity

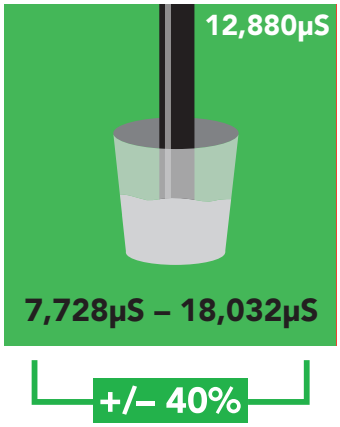
When calibrating Conductivity, you must always calibrate a dry probe first.

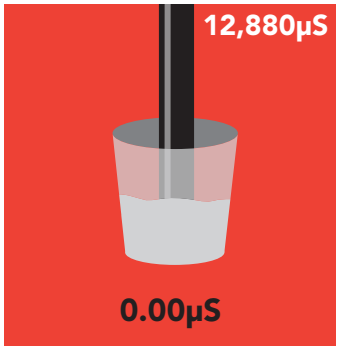
Make sure that the probe is dry before issuing this command, **ec:cal,dry**

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.

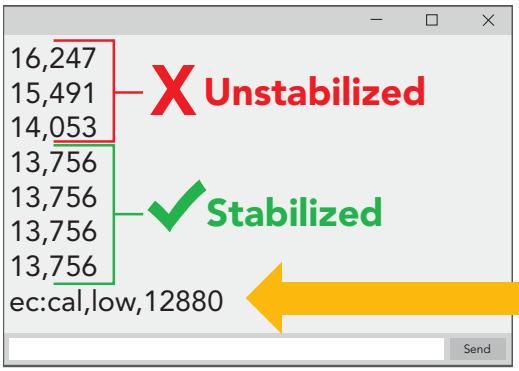


Trapped air in sensing area (shake to remove)



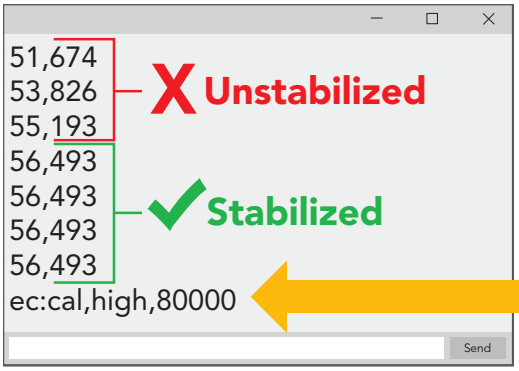


check probe connection, you cannot calibrate to 0.



Once the readings stabilize, issue the low point calibration command. **ec:cal,low,12880**
*(Readings will **NOT** change)*

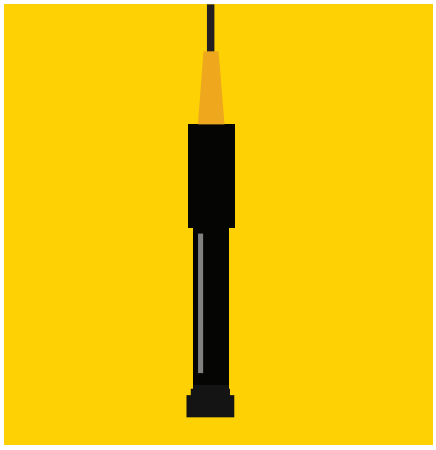
Rinse off the probe before calibrating to the high point. Pour a small amount of the 80,000µS calibration solution into a cup. Shake the probe to remove trapped air. Again, the readings may be off by **1 – 40%** Wait for readings to stabilize.

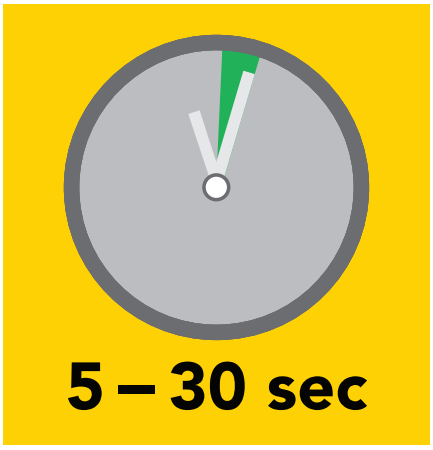


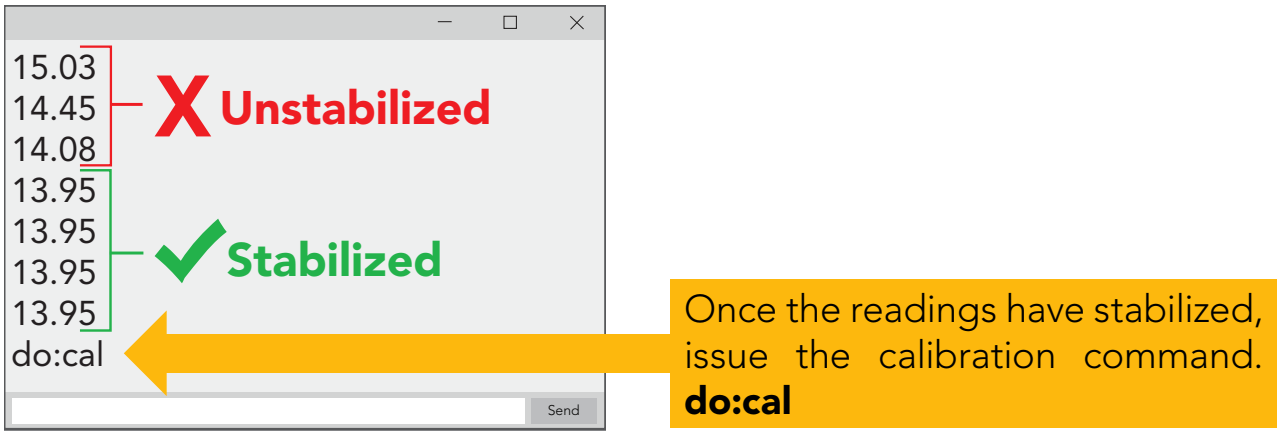
Once the readings stabilize, issue the high point calibration command. **ec:cal,high,80000**
*(Readings **will** change, calibration complete).*

D Calibrate Dissolved Oxygen

Let the Dissolved Oxygen probe sit, exposed to air until the readings stabilize. *(small movement from one reading to the next is normal).*





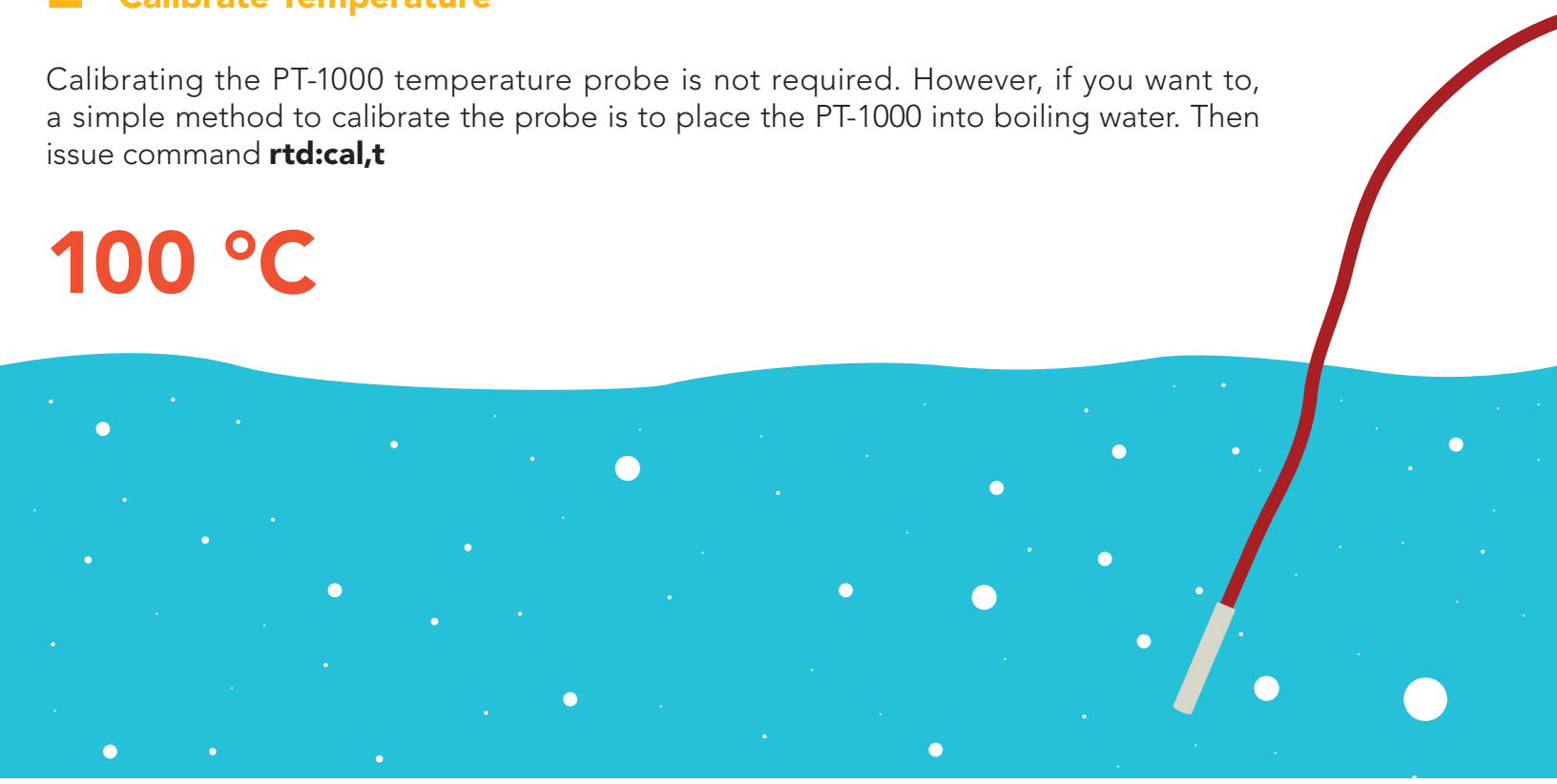


After calibration is complete, you should see readings between **9.09 – 9.1X mg/L.**
(only if temperature, salinity and pressure compensation are at default values)

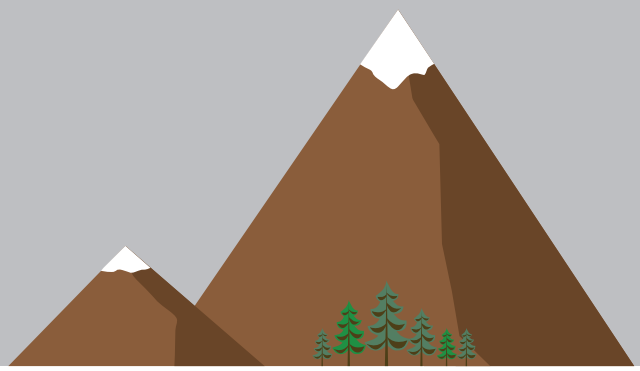
E Calibrate Temperature

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



Elevation in meters	Boiling point
305	98.9 °C
229	99.2 °C
152	99.5 °C
76	99.7 °C
0	100 °C
-76	100.3 °C
-152	100.5 °C



Calibration Complete

Step 10 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!

