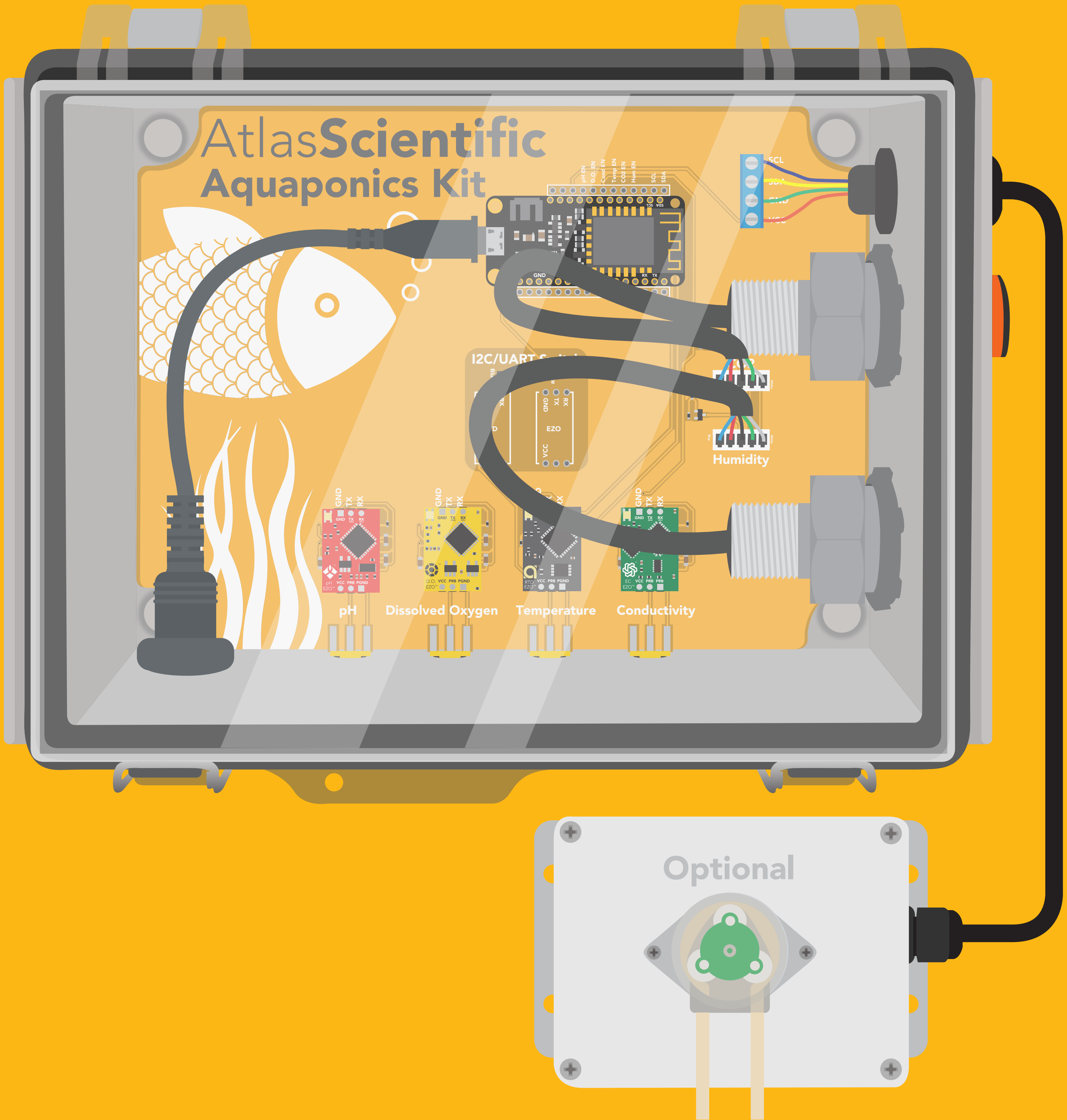
AtlasScientific™
Environmental Robotics
V 1.2

Wi-Fi Aquaponic Kit

Datasheet

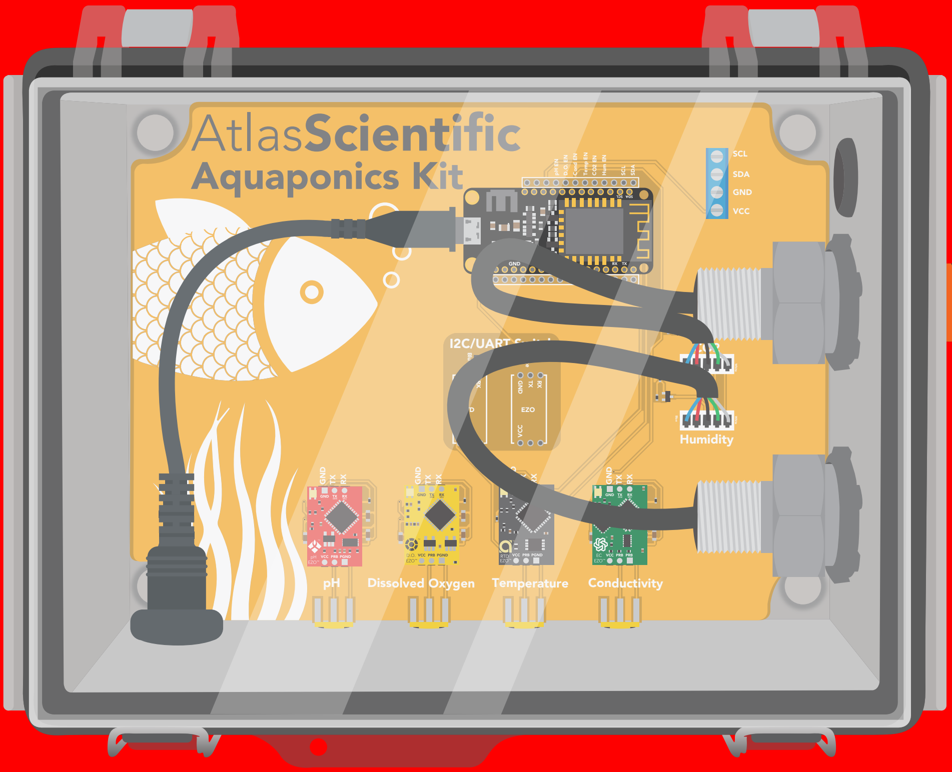


STOP

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)

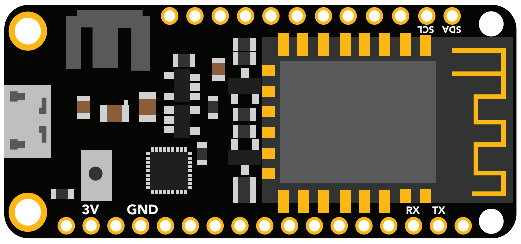
Operating principle

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

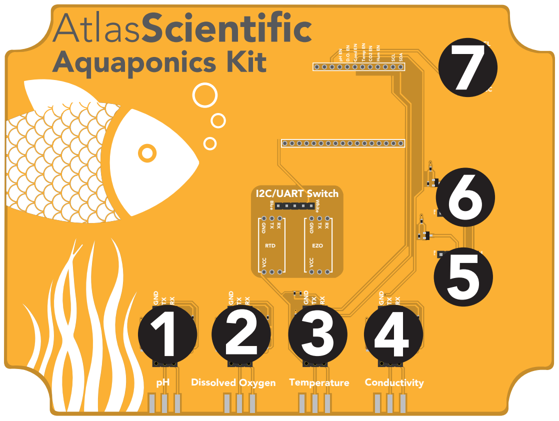
The Wi-Fi Aquaponics kit is controlled using an Adafruit HUZZAH32 as its CPU. The HUZZAH is programmed using the Arduino IDE and uses an onboard ESP32 as its Wi-Fi transmitter. Adafruit HUZZAH32 datasheet.



Sensor ports

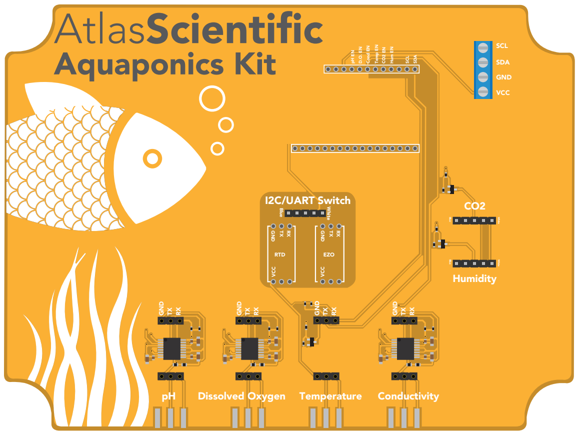
The Wi-Fi Aquaponics kit PCB has 7 sensor ports. Three of the ports are electrically isolated. The isolated ports are marked pH, Dissolved Oxygen, and Conductivity. The isolated ports are needed to take noise-free electrochemical readings. Because the sensing element of a temperature sensor is never 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.

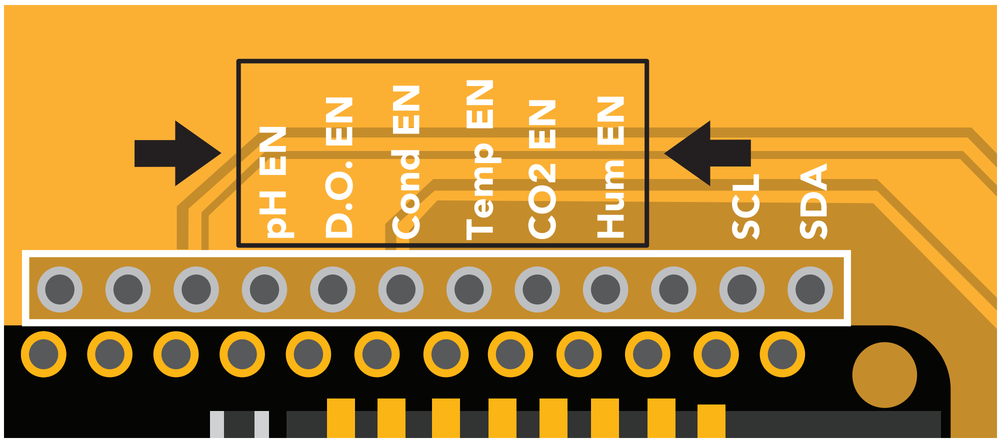


PCB

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.

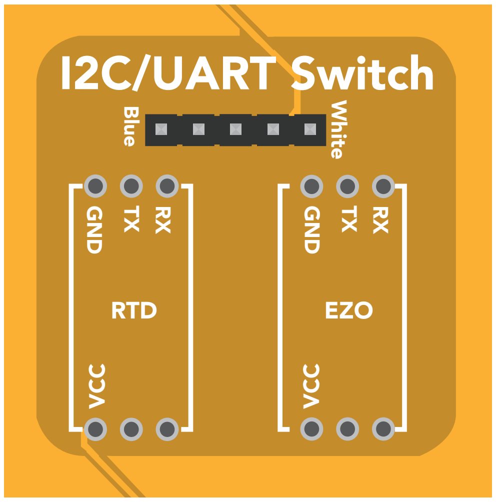
Truth table

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

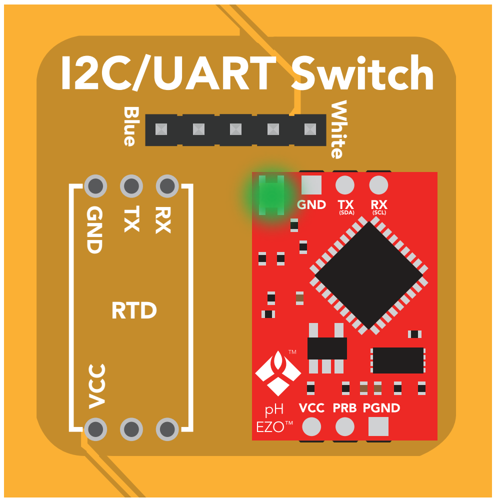
Sensor port 7 *(the terminal block)* does not have an enable pin and can not be turned off.

On Board I2C/UART Switch

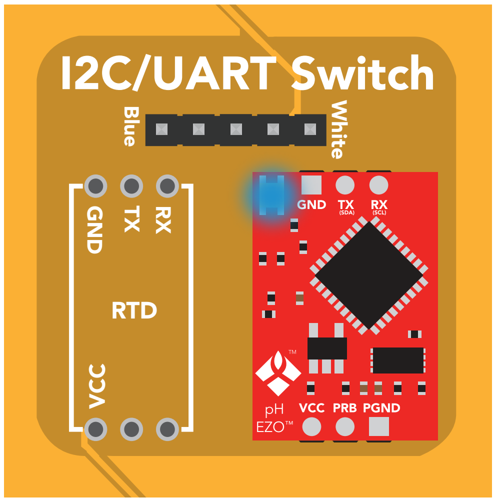
The PCB has a built in protocol toggler which can be used to switch each of the sensors between I2C/UART modes.



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



The EZO circuit in UART mode...



...is now in I2C mode.

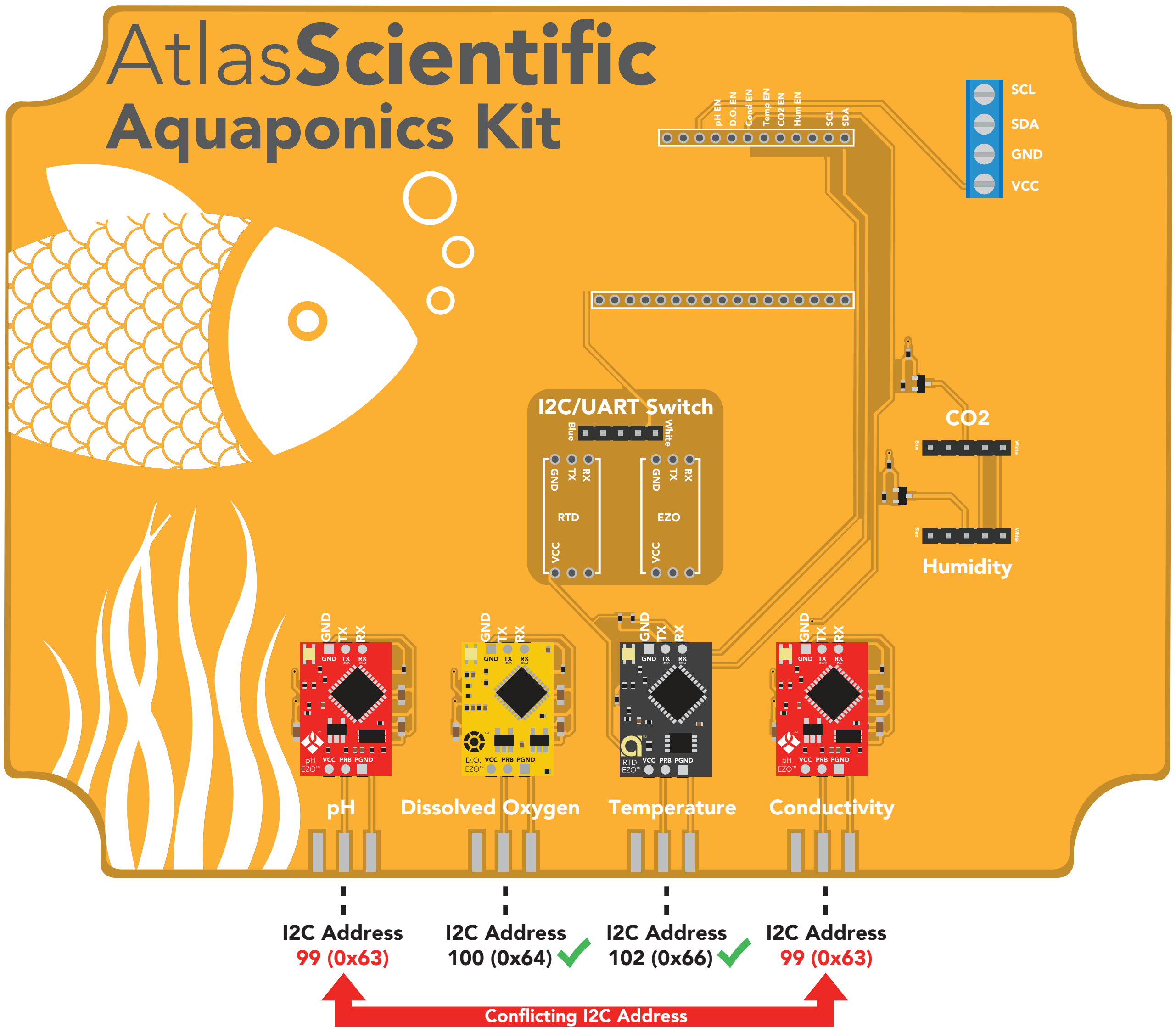
Data protocol

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.

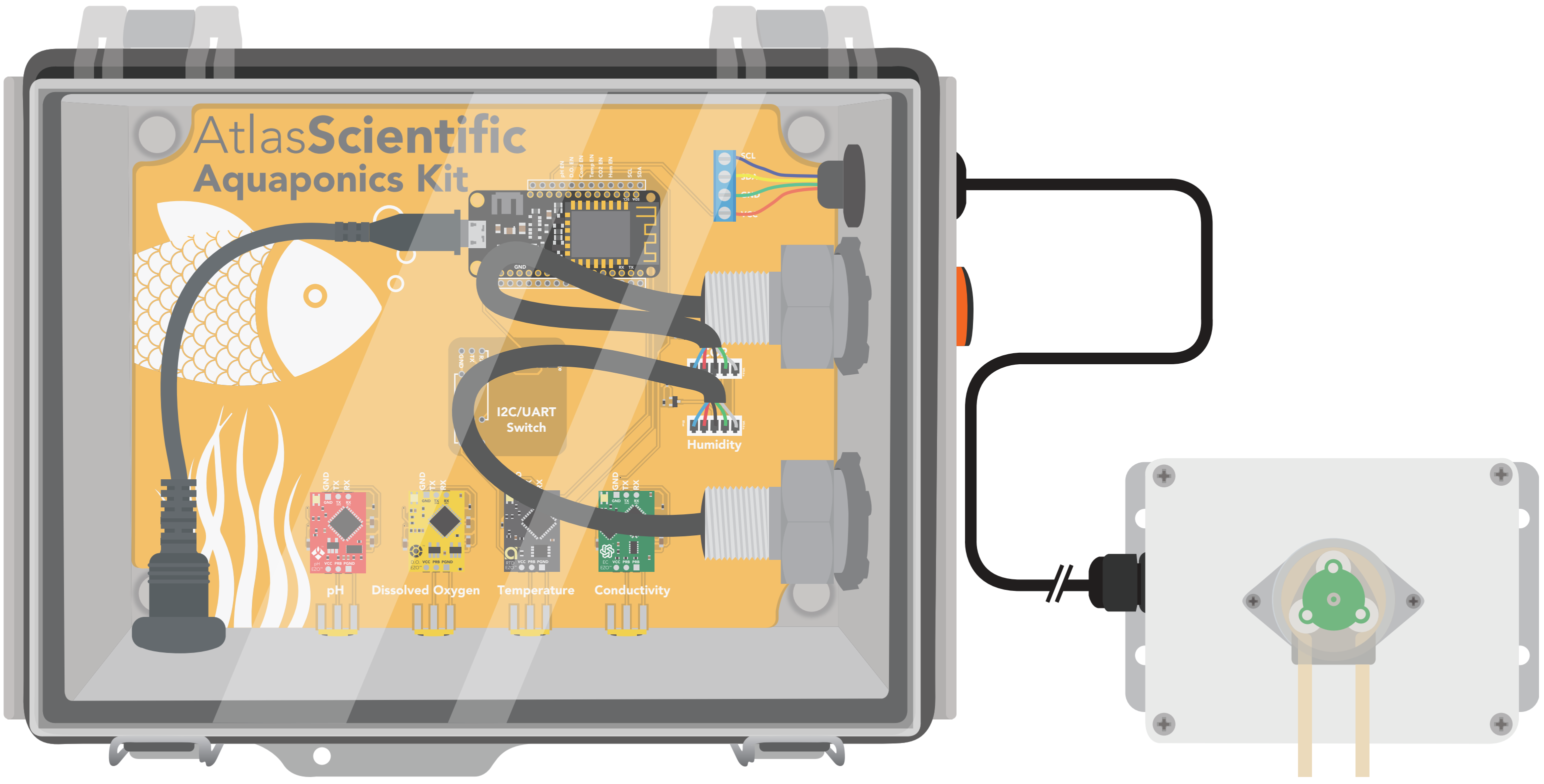


This table lists the default I2C address of components commonly added to this kit.

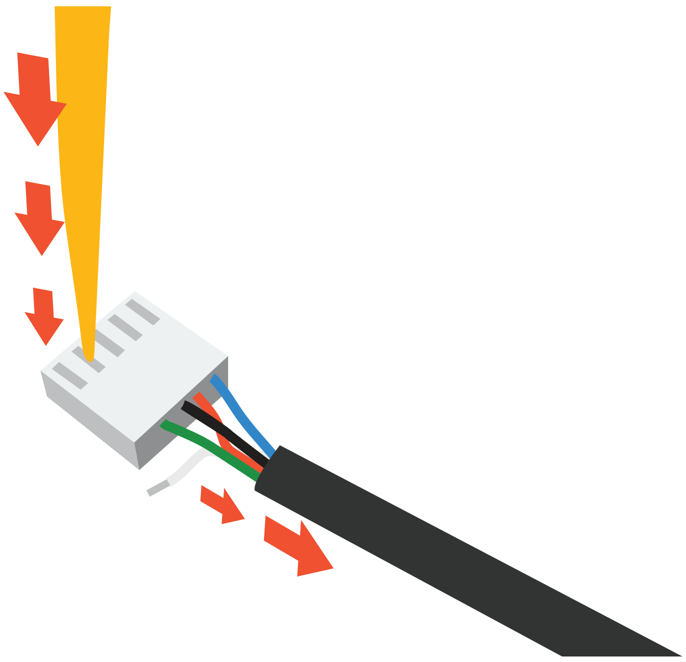
Device	I2C Address	Device	I2C Address	Device	I2C Address
EZO pH	99 (0x63)	EZO EC	100 (0x64)	EZO CO2	105 (0x69)
EZO ORP	98 (0x62)	EZO RTD	102 (0x66)	EZO HUM	111 (0x6F)
EZO DO	97 (0x61)	EZO PMP	103 (0x67)	EZO PMP-L	109 (0x6D)

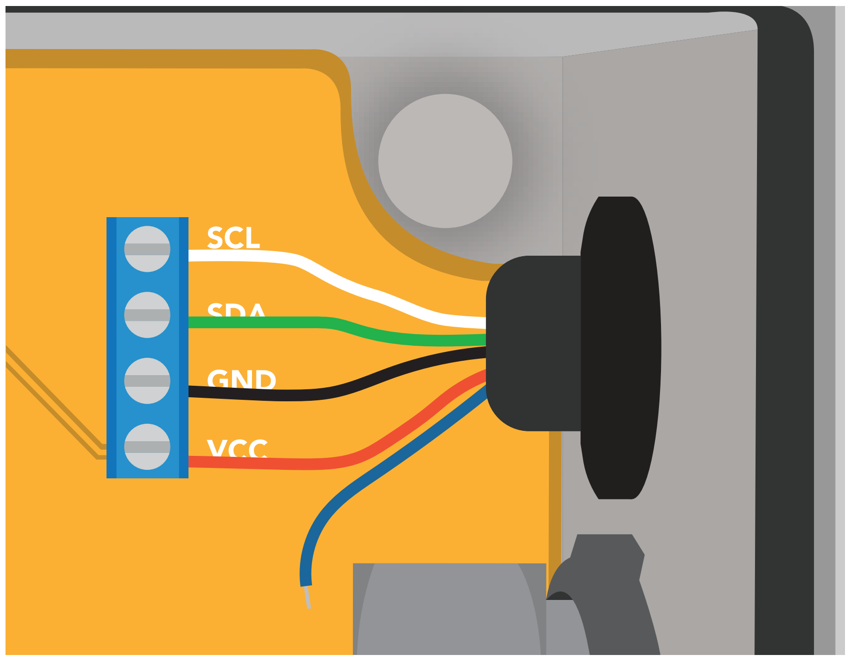
Dosing pump

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

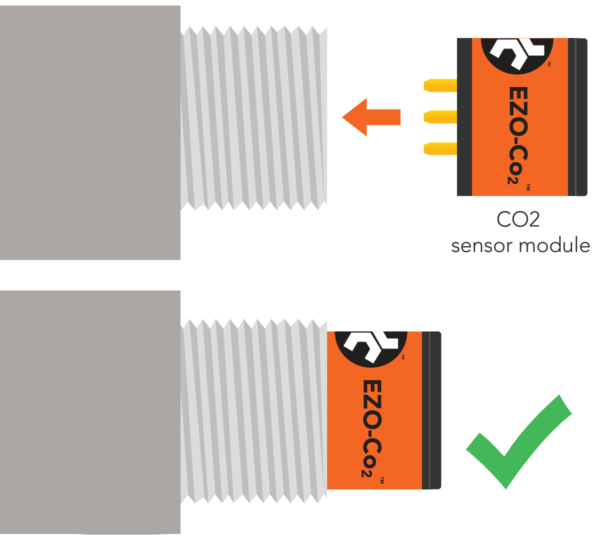
The Atlas-Scientific Wi-Fi hydroponics kit has been designed to upload sensor data to ThingSpeak™, a free, cloud-based data acquisition and visualization platform. You will be required to set up a free account with ThingSpeak™ to upload and visualize the data. With a free account, you can upload data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

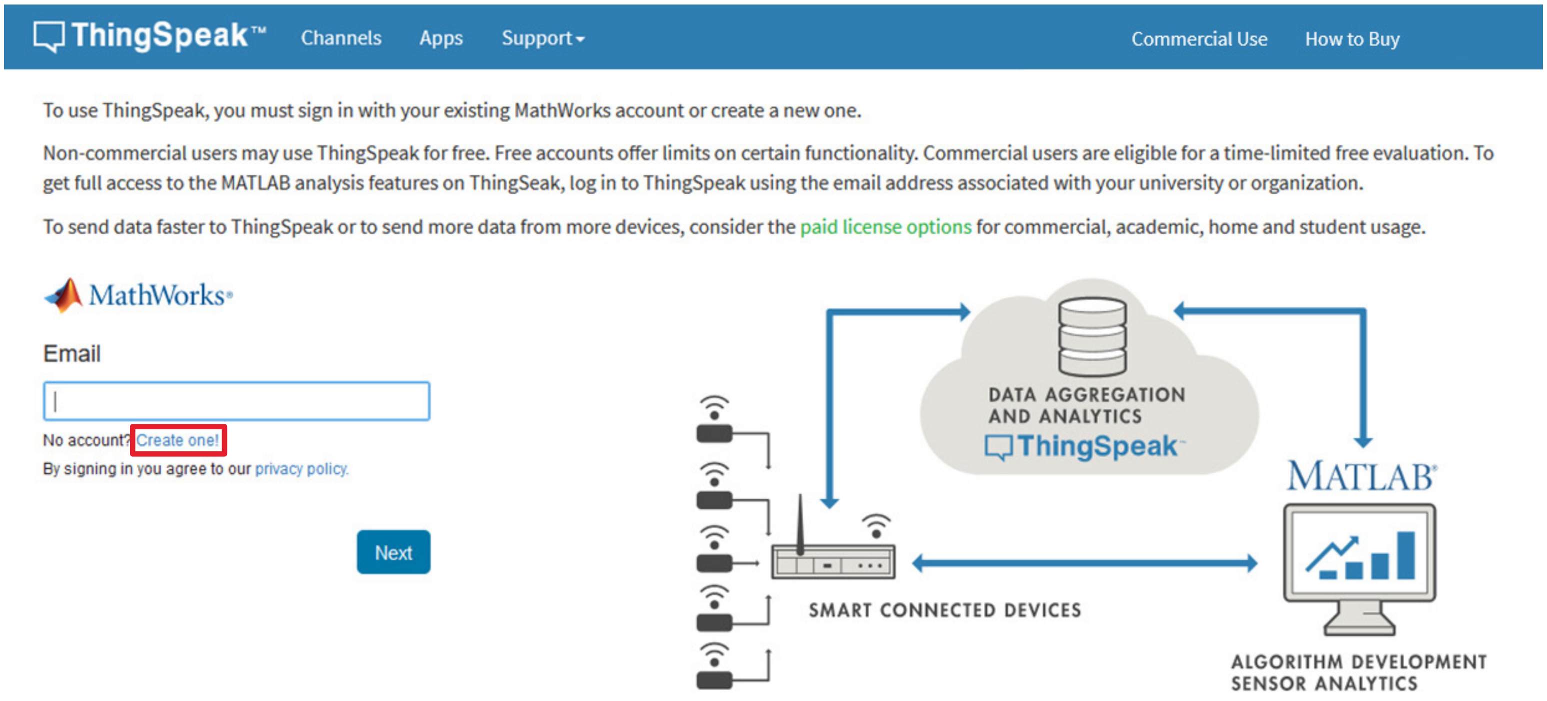
The CO2 sensor module has been removed to protect it during transport. Before you begin setting up the aquaponics kit, plug it back into position.



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.



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the paid license options for commercial, academic, home and student usage.

MathWorks

Email

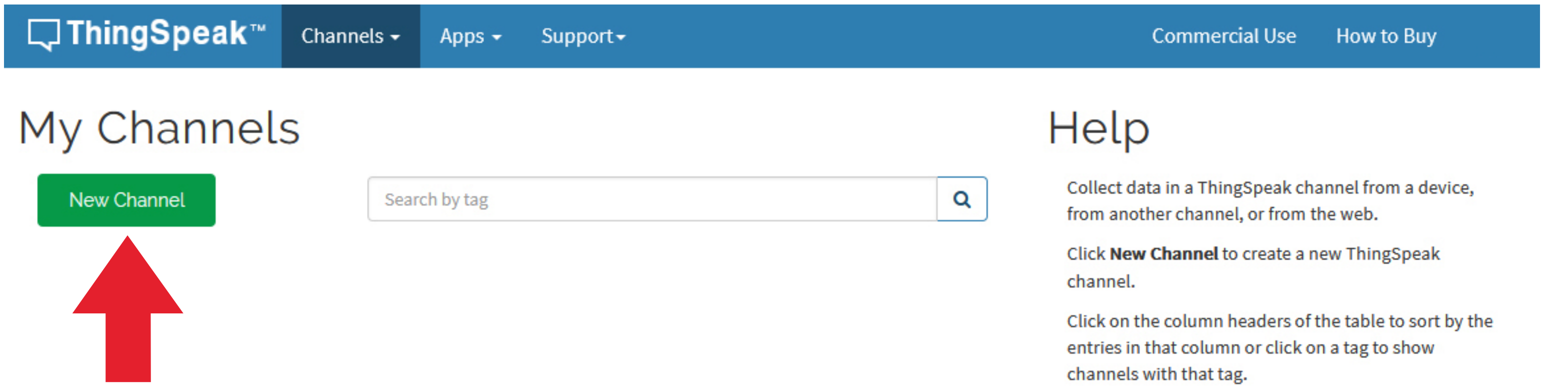
No account? Create one!

By signing in you agree to our privacy policy.

Next

Step 3 Create a Channel

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

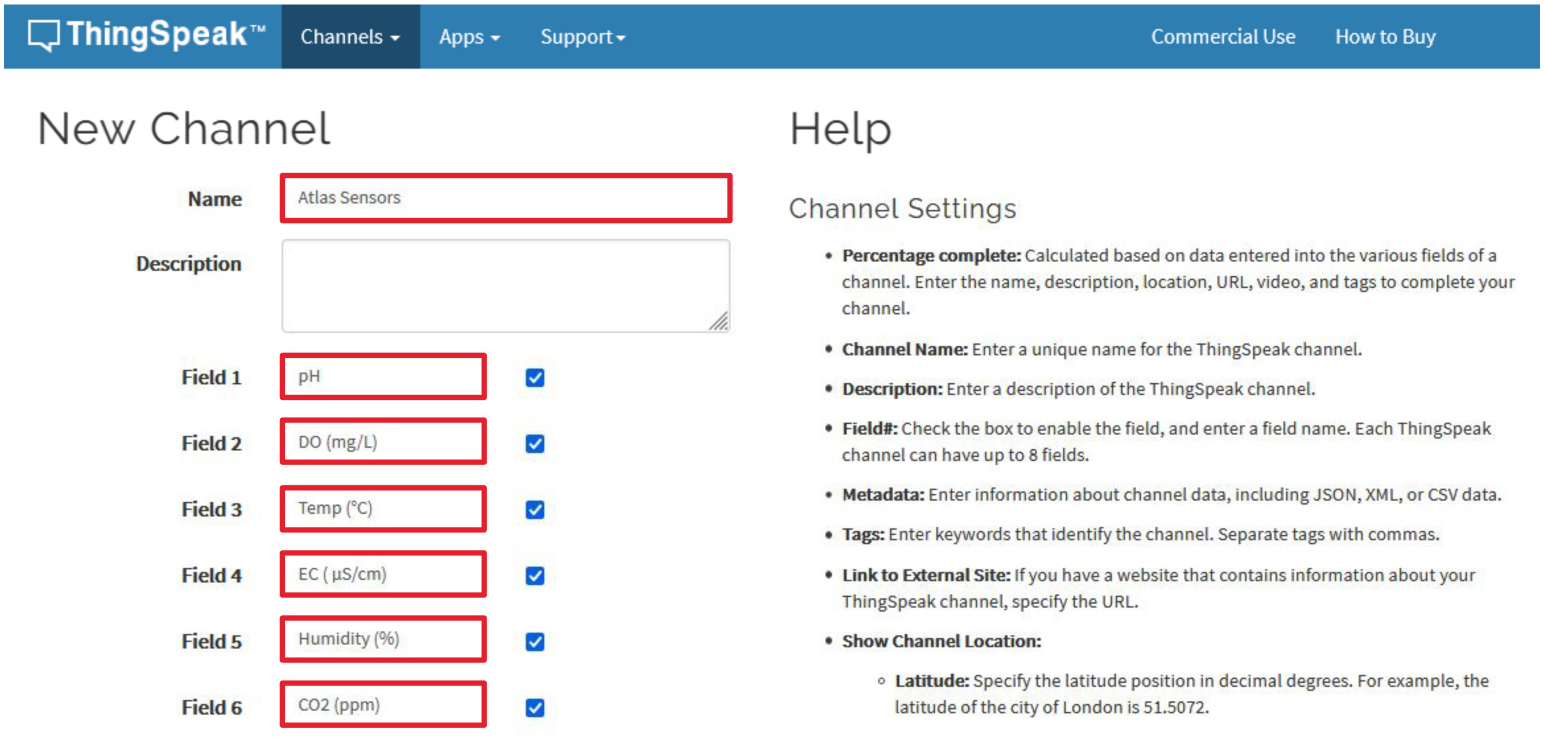


Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.



New Channel

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

Help

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.

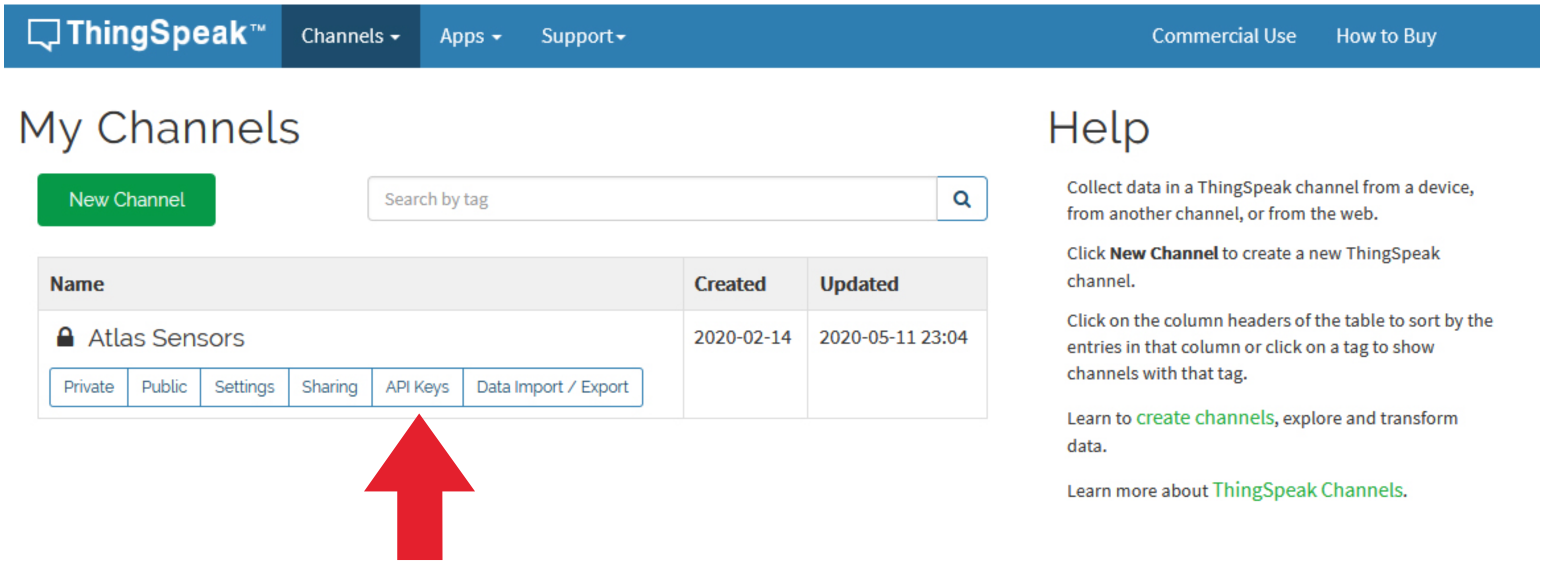
Fill out the highlighted boxes. (Be sure to click on the checkboxes to enable **fields 2 – 6**)
For reference, this is what we entered.

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**.



My Channels

Name	Created	Updated
Atlas Sensors Private · Public · Settings · Sharing · API Keys · Data Import / Export	2020-02-14	2020-05-11 23:04

Help

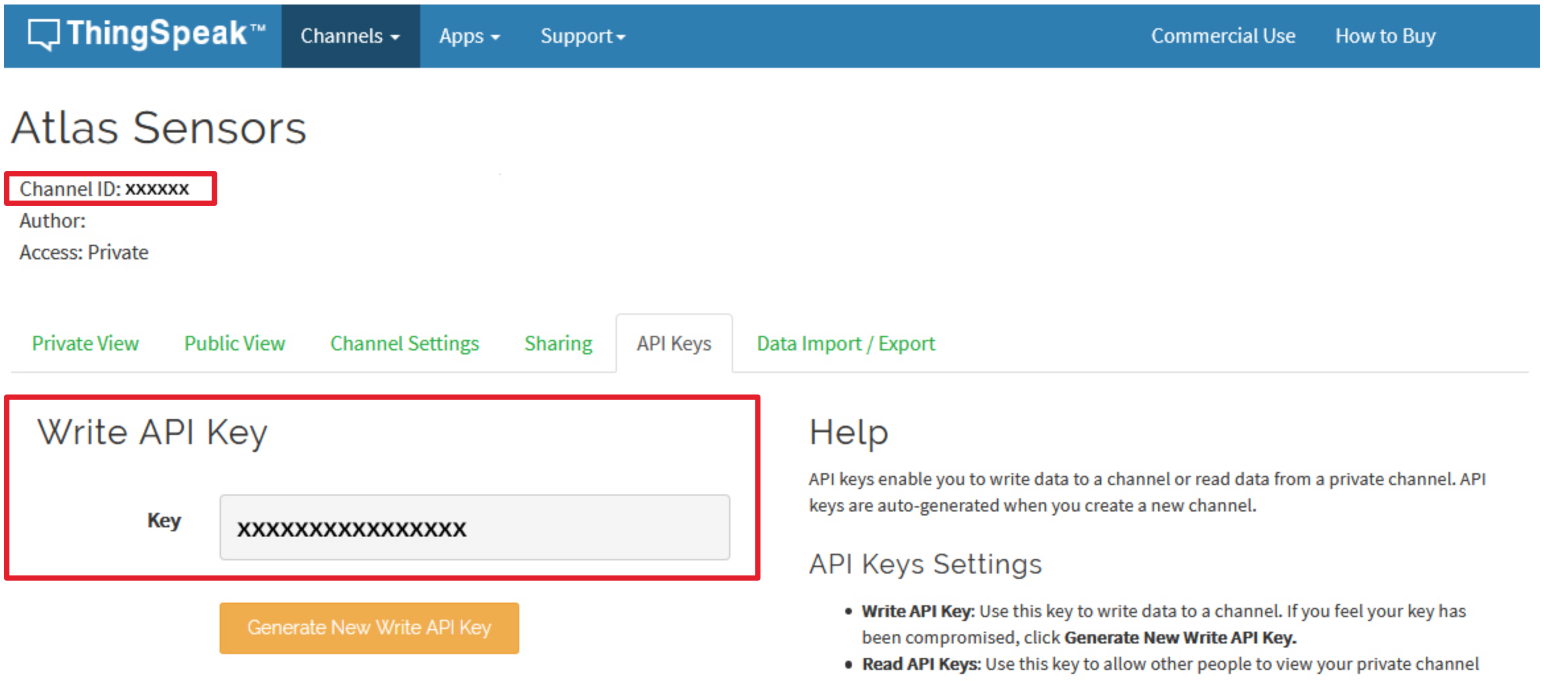
Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.



Atlas Sensors

Channel ID: **xxxxxx**
Author:
Access: Private

Write API Key

Key XXXXXXXXXXXXXXXX

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

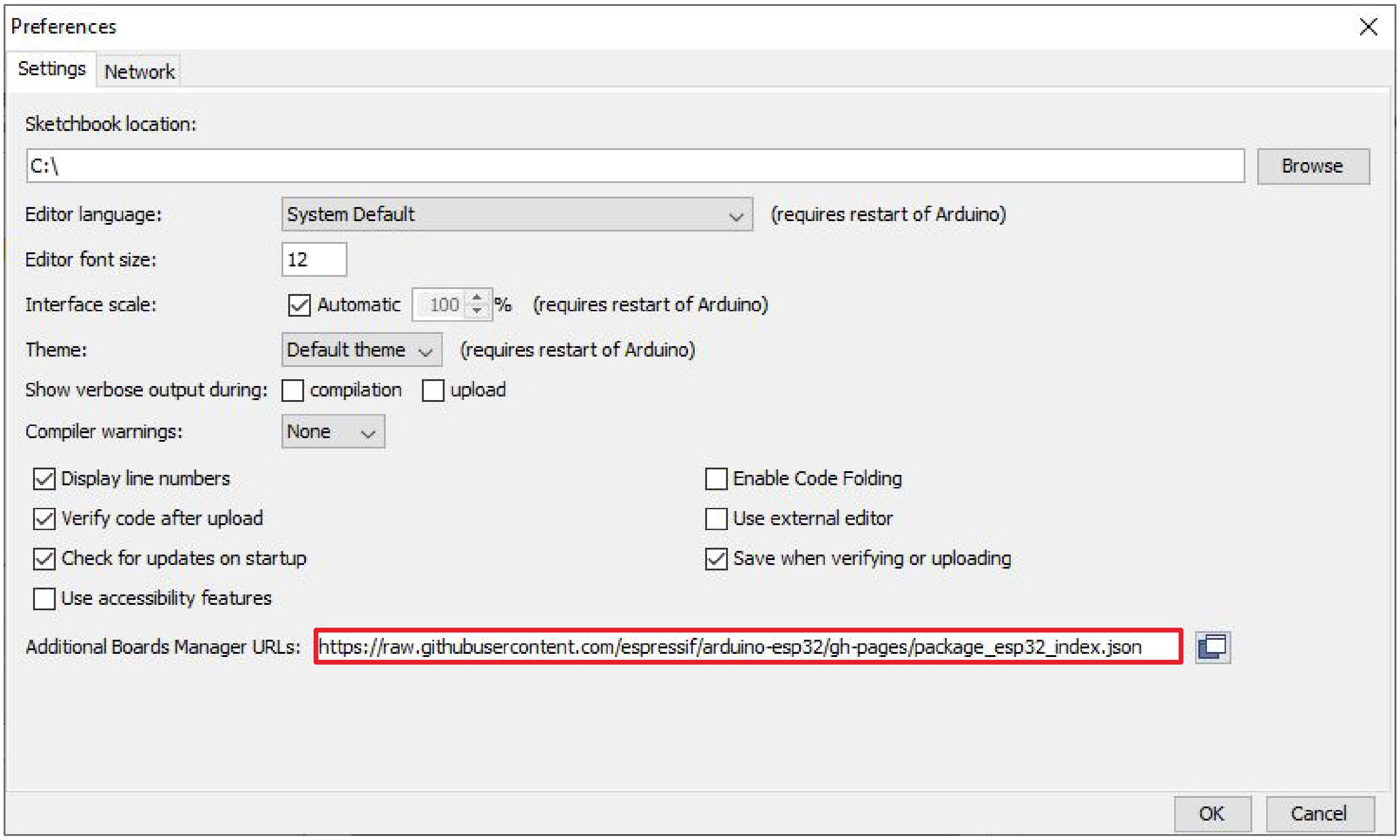
- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys:** Use this key to allow other people to view your private channel

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.



Preferences

Sketchbook location: C:\

Editor language: System Default (requires restart of Arduino)
Editor font size: 12
Interface scale: [✓ Automatic] 100 % (requires restart of Arduino)
Theme: Default theme (requires restart of Arduino)
Show verbose output during: ☐ compilation ☐ upload
Compiler warnings: None

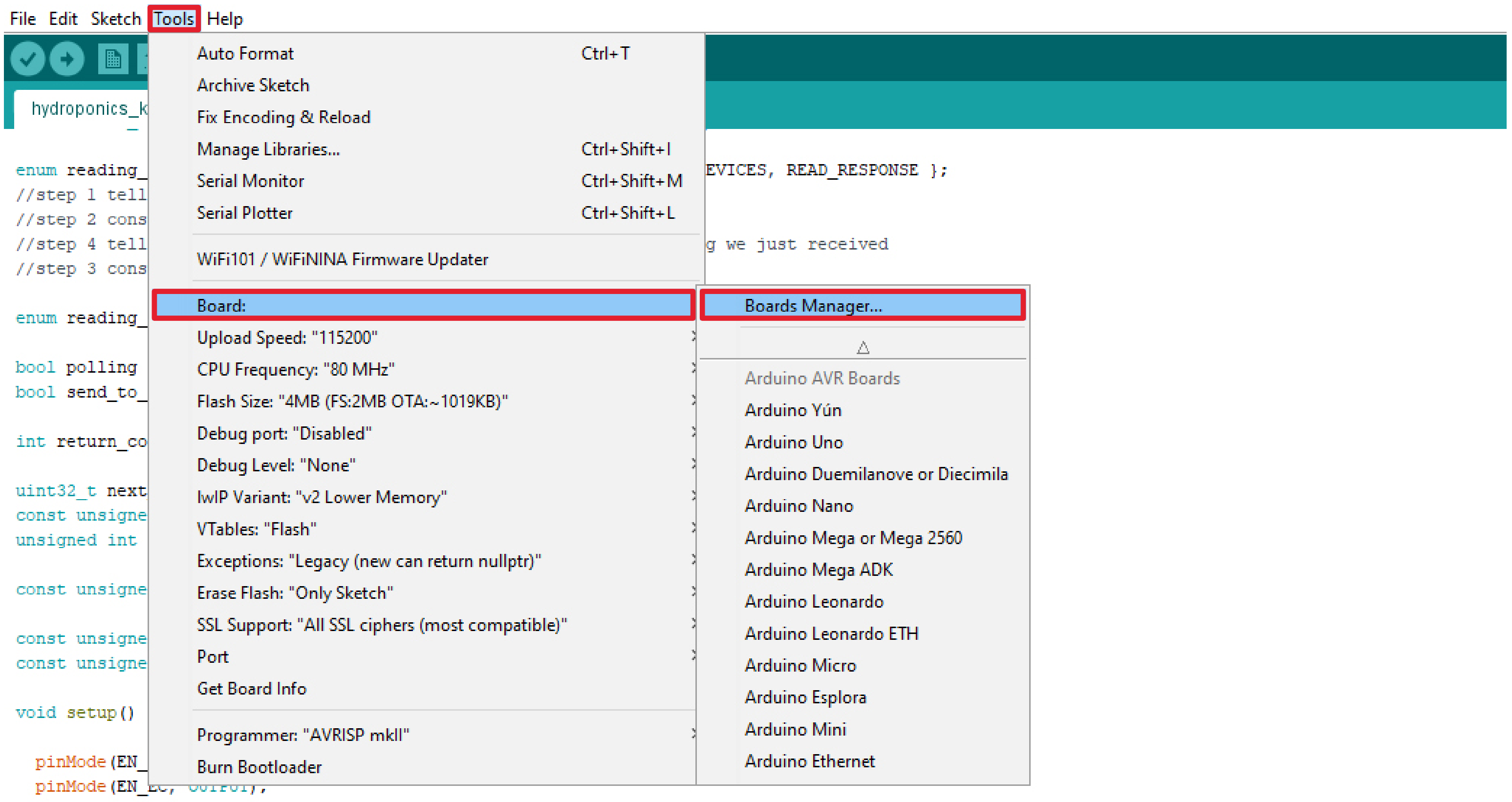
- ✓ Display line numbers
- ✓ Verify code after upload
- ✓ Check for updates on startup
- ☐ Use accessibility features
- ☐ Enable Code Folding
- ☐ Use external editor
- ✓ Save when verifying or uploading

Additional Boards Manager URLs: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

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

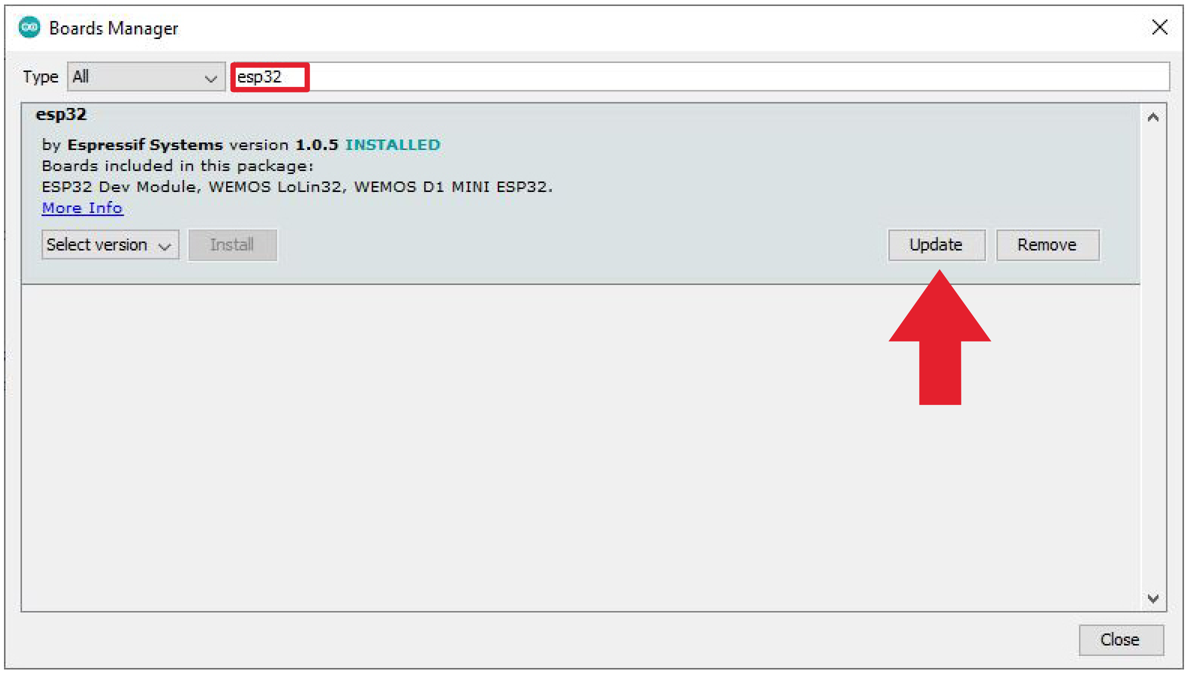
B Update the Esp32 board

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



Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Disabled"
Debug Level: "None"
IwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Legacy (new can return nullptr)"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Port
Programmer: "AVRISP mkII"

```
enum reading_                    EVICES, READ_RESPONSE };
//step 1 tell
//step 2 cons
//step 4 tell                    g we just received
//step 3 cons

enum reading_

bool polling
bool send_to_

int return_co

uint32_t next
const unsigne
unsigned int

const unsigne

const unsigne
const unsigne

void setup()

  pinMode(EN_
  pinMode(EN_EC, OUTPUT);
```



esp32
by **Espressif Systems** version **1.0.5** INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

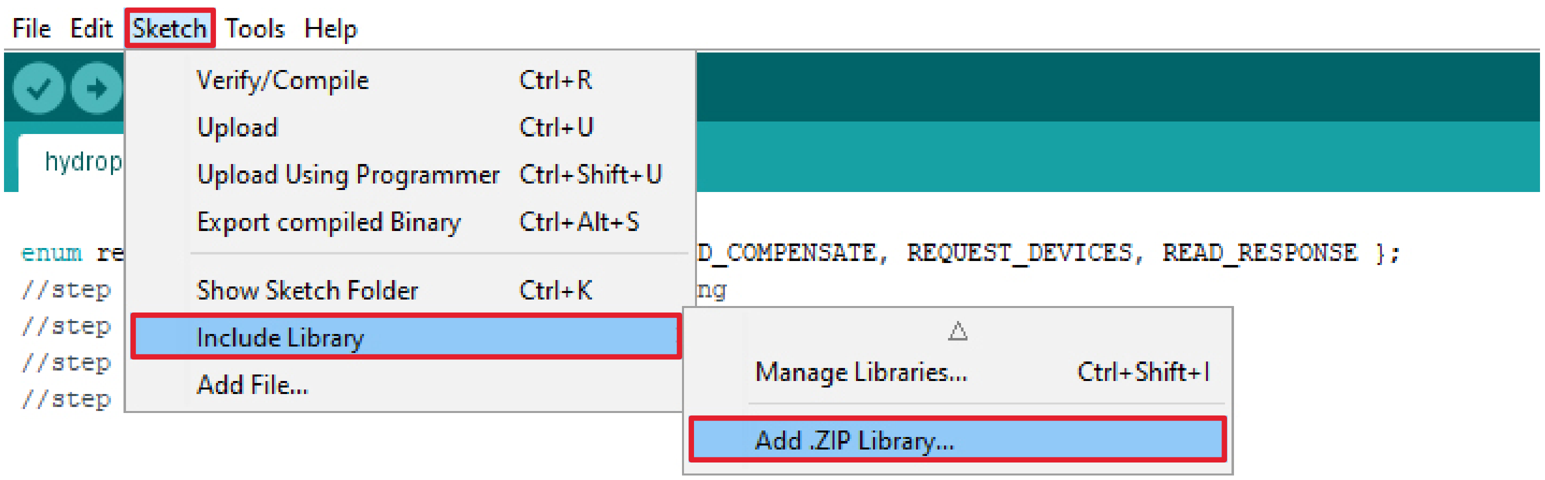
In the search bar of the Boards Manager, lookup **esp32.**
Update to the most recent version if you don't already have it.

(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.

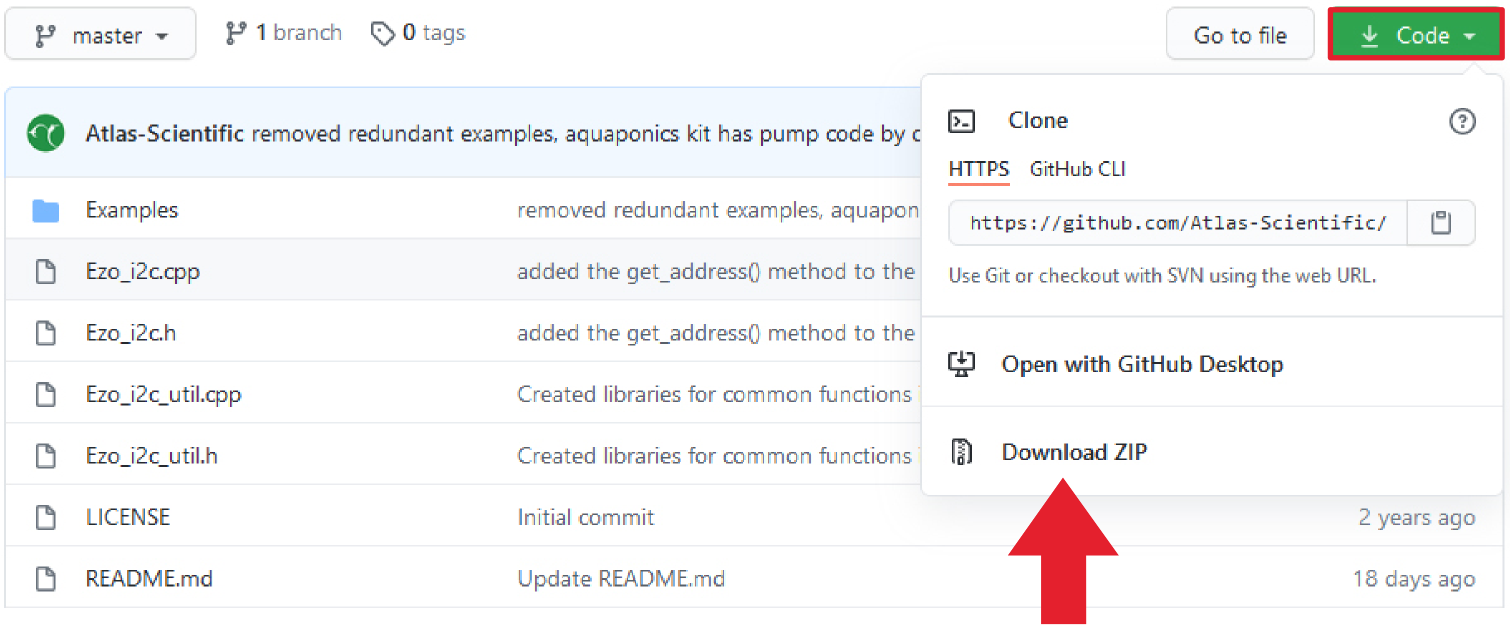
Don't unzip it!
Import the .ZIP file into your Arduino IDE.
To import the .ZIP file go to **Sketch > Include Library > Add .ZIP Library**


File
Edit
Sketch
Tools
Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
hydrop

```
enum re                     D_COMPENSATE, REQUEST_DEVICES, READ_RESPONSE };
//step                      ng
//step
//step
//step
```

D Add the EZO I2C Library

To download the Ezo_I2c library file, click HERE.



Atlas-Scientific removed redundant examples, aquaponics kit has pump code by c

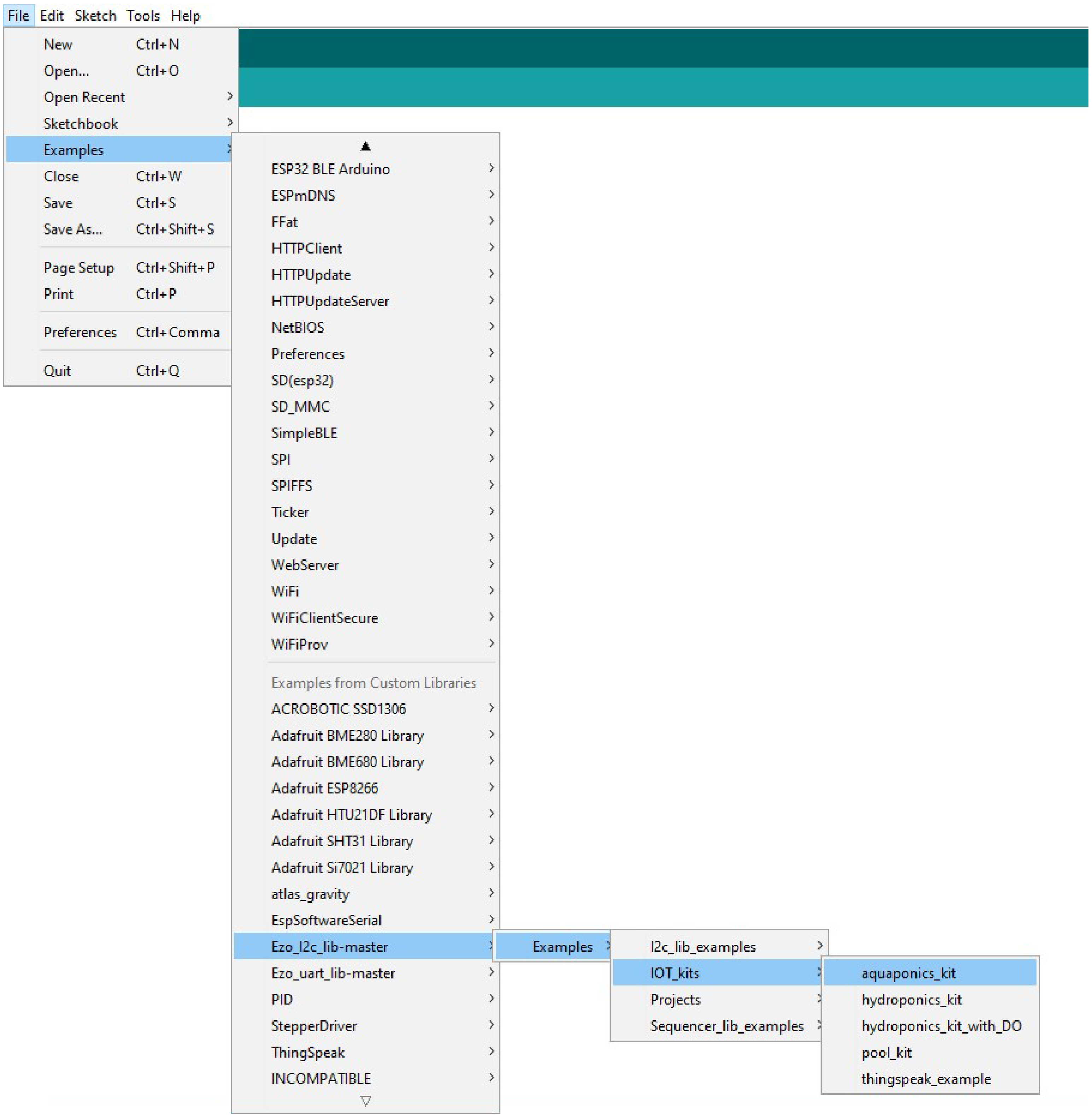
Examples	removed redundant examples, aquapon	
Ezo_i2c.cpp	added the get_address() method to the	
Ezo_i2c.h	added the get_address() method to the	
Ezo_i2c_util.cpp	Created libraries for common functions	
Ezo_i2c_util.h	Created libraries for common functions	
LICENSE	Initial commit	2 years ago
README.md	Update README.md	18 days ago

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

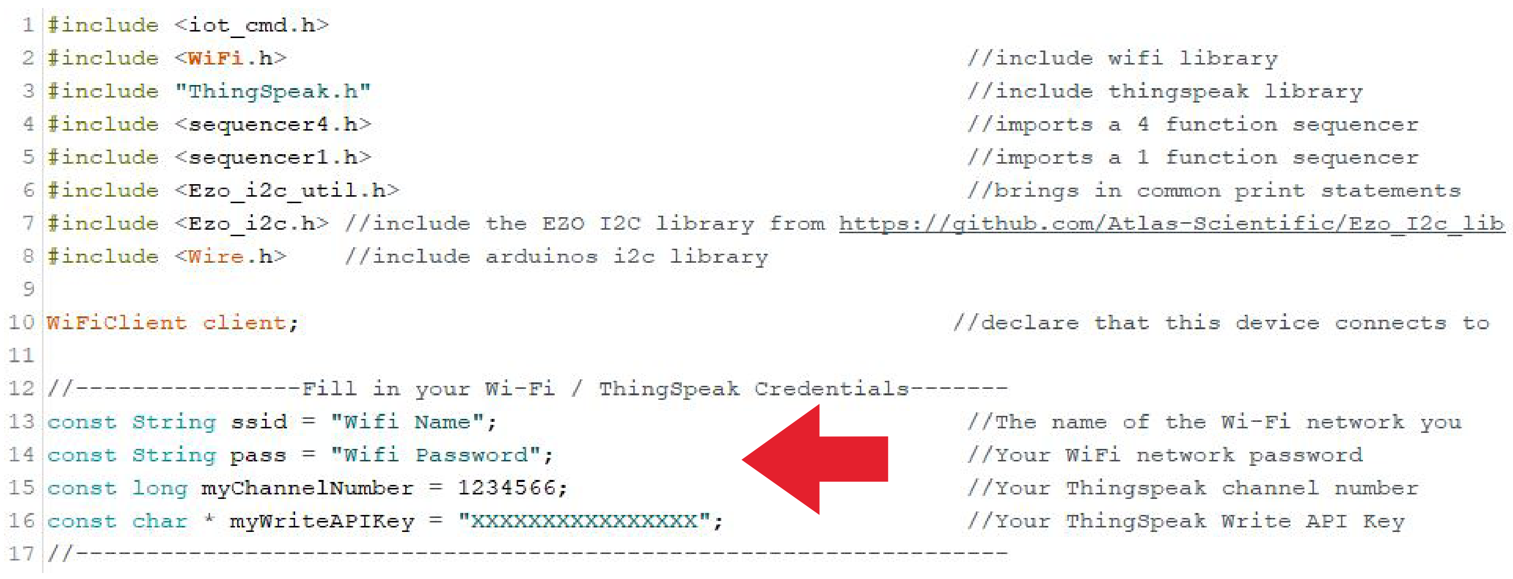
A Select, open and adjust the code you want to use for your Wi-Fi Kit

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



B Fill in your Wi-Fi / ThingSpeak credentials

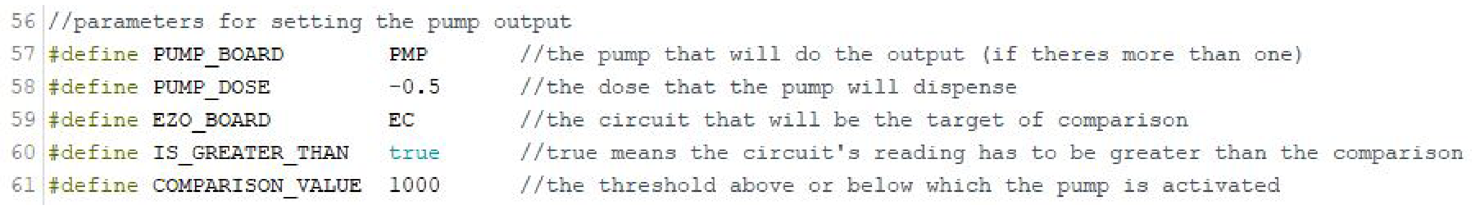
Fill in your Wi-Fi name and Password, along with the Channel ID and Write API Key to the code. *(see step 3)*



```
#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.

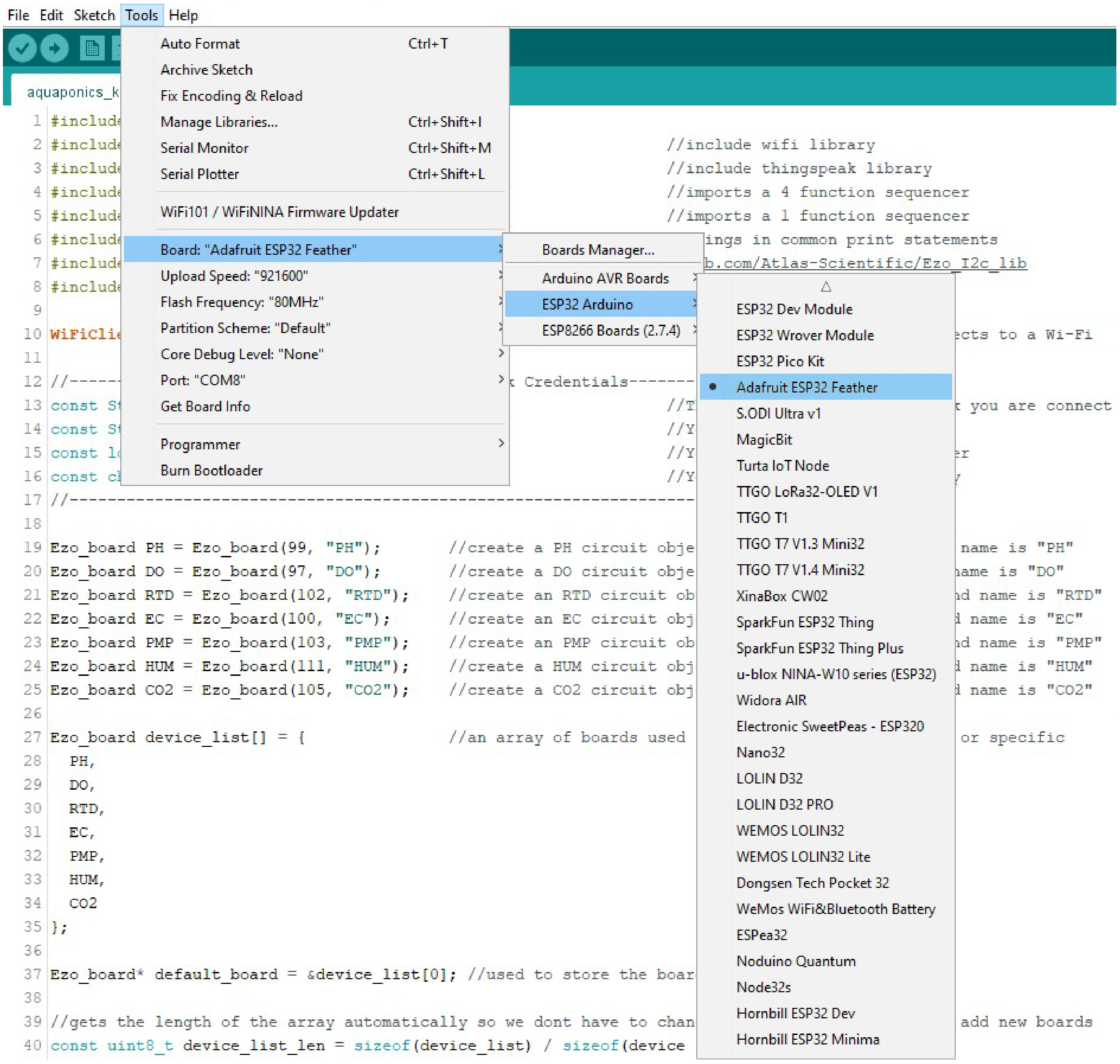


```
//parameters for setting the pump output
#define PUMP_BOARD        PMP      //the pump that will do the output (if theres more than one)
#define PUMP_DOSE         -0.5     //the dose that the pump will dispense
#define EZO_BOARD         EC       //the circuit that will be the target of comparison
#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



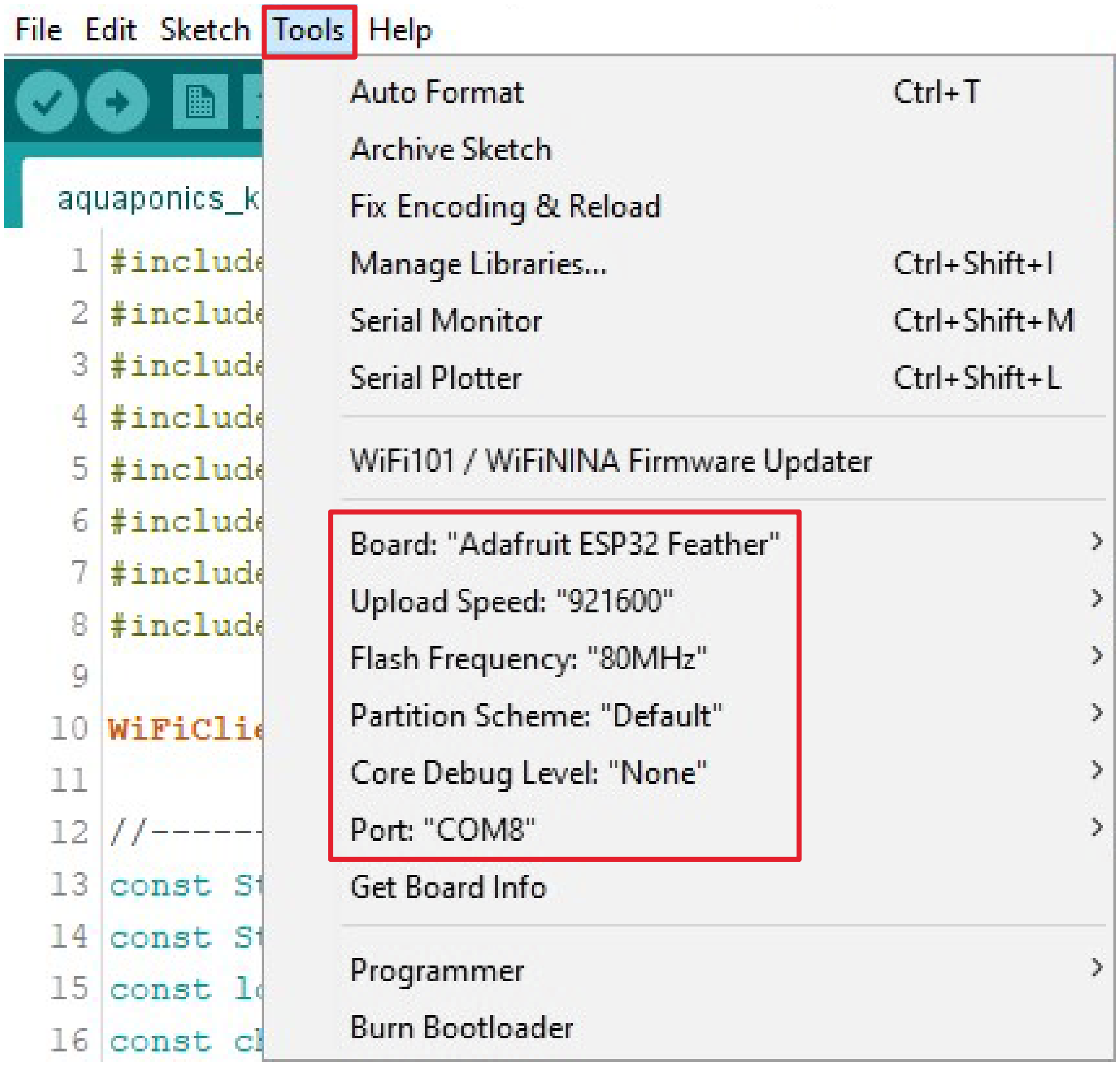
Board: "Adafruit ESP32 Feather"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Partition Scheme: "Default"
Core Debug Level: "None"
Port: "COM8"

```
#include
#include                                          //include wifi library
#include                                          //include thingspeak library
#include                                          //imports a 4 function sequencer
#include                                          //imports a 1 function sequencer
#include                                          ings in common print statements
#include                                          b.com/Atlas-Scientific/Ezo_I2c_lib
#include

WiFiClie                                                                    ects to a Wi-Fi

//-----                      k Credentials-------
const St                                          //T                       t you are connect
const St                                          //Y
const l                                           //Y                       er
const c                                           //Y                       y
//------------------------------------------------------------------

Ezo_board PH = Ezo_board(99, "PH");         //create a PH circuit obje         name is "PH"
Ezo_board DO = Ezo_board(97, "DO");         //create a DO circuit obje         ame is "DO"
Ezo_board RTD = Ezo_board(102, "RTD");      //create an RTD circuit ob         nd name is "RTD"
Ezo_board EC = Ezo_board(100, "EC");        //create an EC circuit obj         d name is "EC"
Ezo_board PMP = Ezo_board(103, "PMP");      //create an PMP circuit ob         nd name is "PMP"
Ezo_board HUM = Ezo_board(111, "HUM");      //create a HUM circuit obj         d name is "HUM"
Ezo_board CO2 = Ezo_board(105, "CO2");      //create a CO2 circuit obj         d name is "CO2"

Ezo_board device_list[] = {                 //an array of boards used          or specific
  PH,
  DO,
  RTD,
  EC,
  PMP,
  HUM,
  CO2
};

Ezo_board* default_board = &device_list[0]; //used to store the boar

//gets the length of the array automatically so we dont have to chan           add new boards
const uint8_t device_list_len = sizeof(device_list) / sizeof(device
```

B Adjust CPU Settings

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

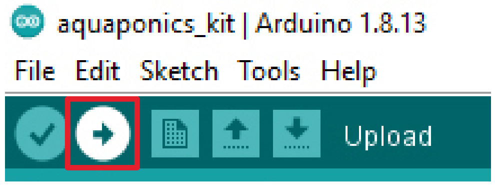
For reference, this is what Atlas Scientific set the CPU settings to.
(your options may not be exactly the same, just try and match them as closely as possible. Don't forget to set the correct com port for your device.)



Board: "Adafruit ESP32 Feather"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Partition Scheme: "Default"
Core Debug Level: "None"
Port: "COM8"

```
#include
#include
#include
#include
#include
#include
#include
#include

WiFiClie

//------
const St
const St
const lo
const ch
```

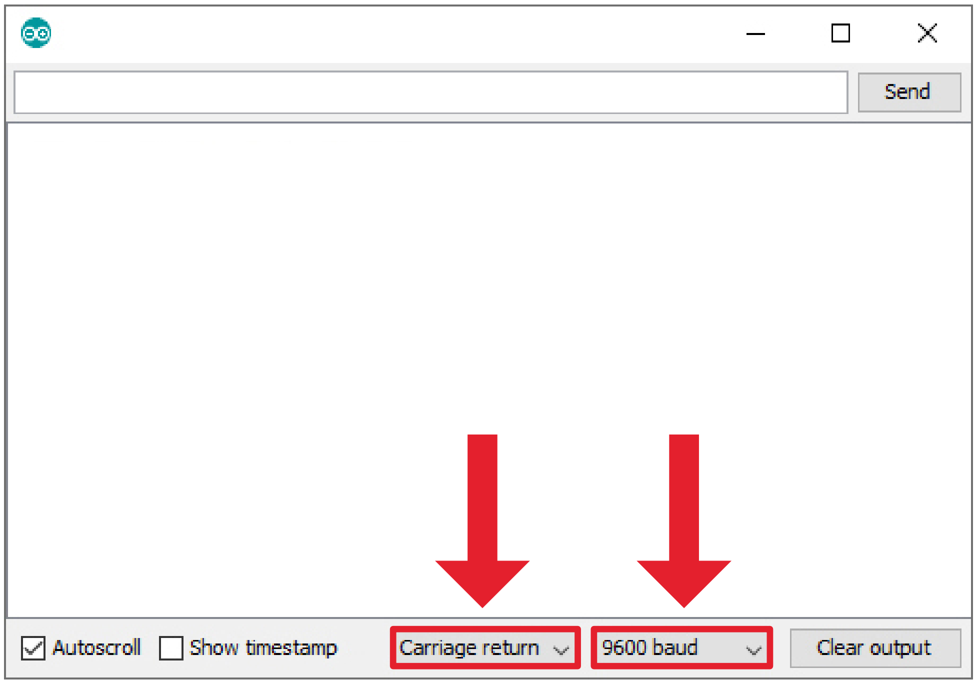
C Compile and upload



Compile and upload the code.

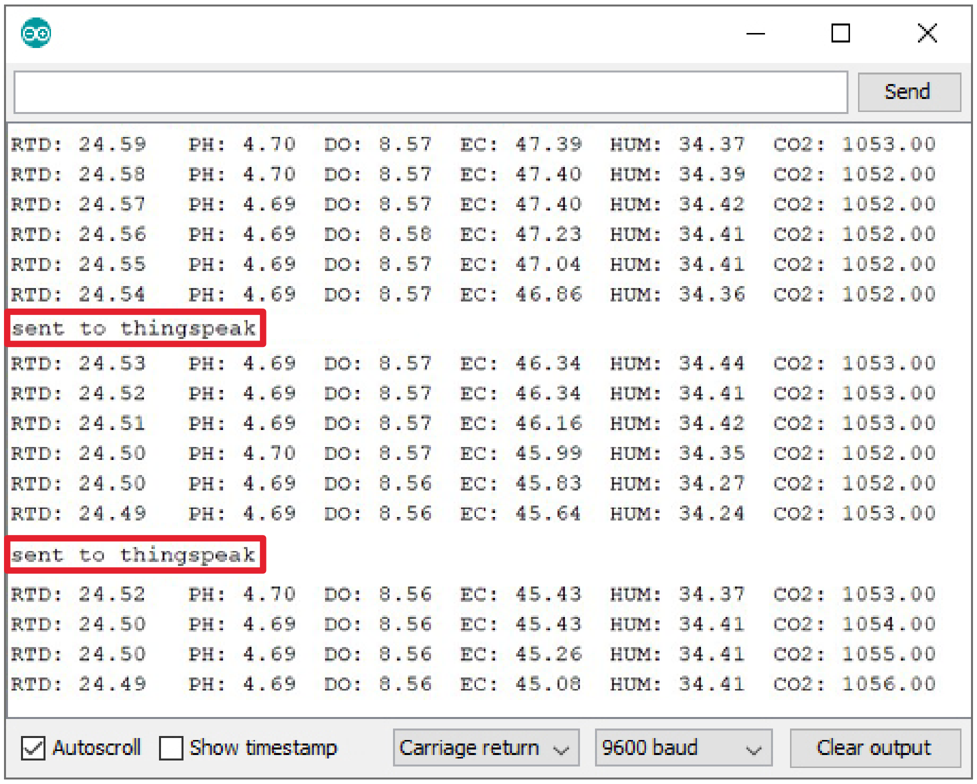
Step 8 See the readings

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.**

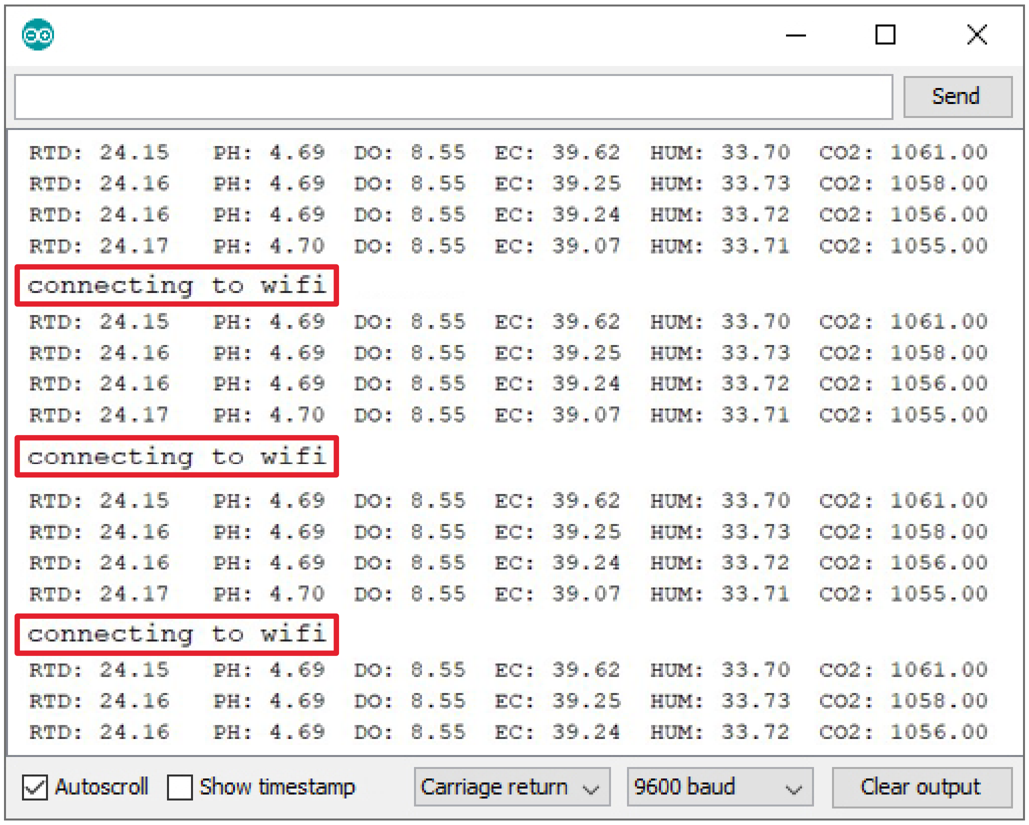
The Wi-Fi Aquaponics Meter will always attempt to connect to ThingSpeak on bootup.



```
RTD: 24.59   PH: 4.70  DO: 8.57  EC: 47.39  HUM: 34.37  CO2: 1053.00
RTD: 24.58   PH: 4.70  DO: 8.57  EC: 47.40  HUM: 34.39  CO2: 1052.00
RTD: 24.57   PH: 4.69  DO: 8.57  EC: 47.40  HUM: 34.42  CO2: 1052.00
RTD: 24.56   PH: 4.69  DO: 8.58  EC: 47.23  HUM: 34.41  CO2: 1052.00
RTD: 24.55   PH: 4.69  DO: 8.57  EC: 47.04  HUM: 34.41  CO2: 1052.00
RTD: 24.54   PH: 4.69  DO: 8.57  EC: 46.86  HUM: 34.36  CO2: 1052.00
sent to thingspeak
RTD: 24.53   PH: 4.69  DO: 8.57  EC: 46.34  HUM: 34.44  CO2: 1053.00
RTD: 24.52   PH: 4.69  DO: 8.57  EC: 46.34  HUM: 34.41  CO2: 1053.00
RTD: 24.51   PH: 4.69  DO: 8.57  EC: 46.16  HUM: 34.42  CO2: 1053.00
RTD: 24.50   PH: 4.70  DO: 8.57  EC: 45.99  HUM: 34.35  CO2: 1052.00
RTD: 24.50   PH: 4.69  DO: 8.56  EC: 45.83  HUM: 34.27  CO2: 1052.00
RTD: 24.49   PH: 4.69  DO: 8.56  EC: 45.64  HUM: 34.24  CO2: 1053.00
sent to thingspeak
RTD: 24.52   PH: 4.70  DO: 8.56  EC: 45.43  HUM: 34.37  CO2: 1053.00
RTD: 24.50   PH: 4.69  DO: 8.56  EC: 45.43  HUM: 34.41  CO2: 1054.00
RTD: 24.50   PH: 4.69  DO: 8.56  EC: 45.26  HUM: 34.41  CO2: 1055.00
RTD: 24.49   PH: 4.69  DO: 8.56  EC: 45.08  HUM: 34.41  CO2: 1056.00
```

AtlasScientific Environmental Robotics

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

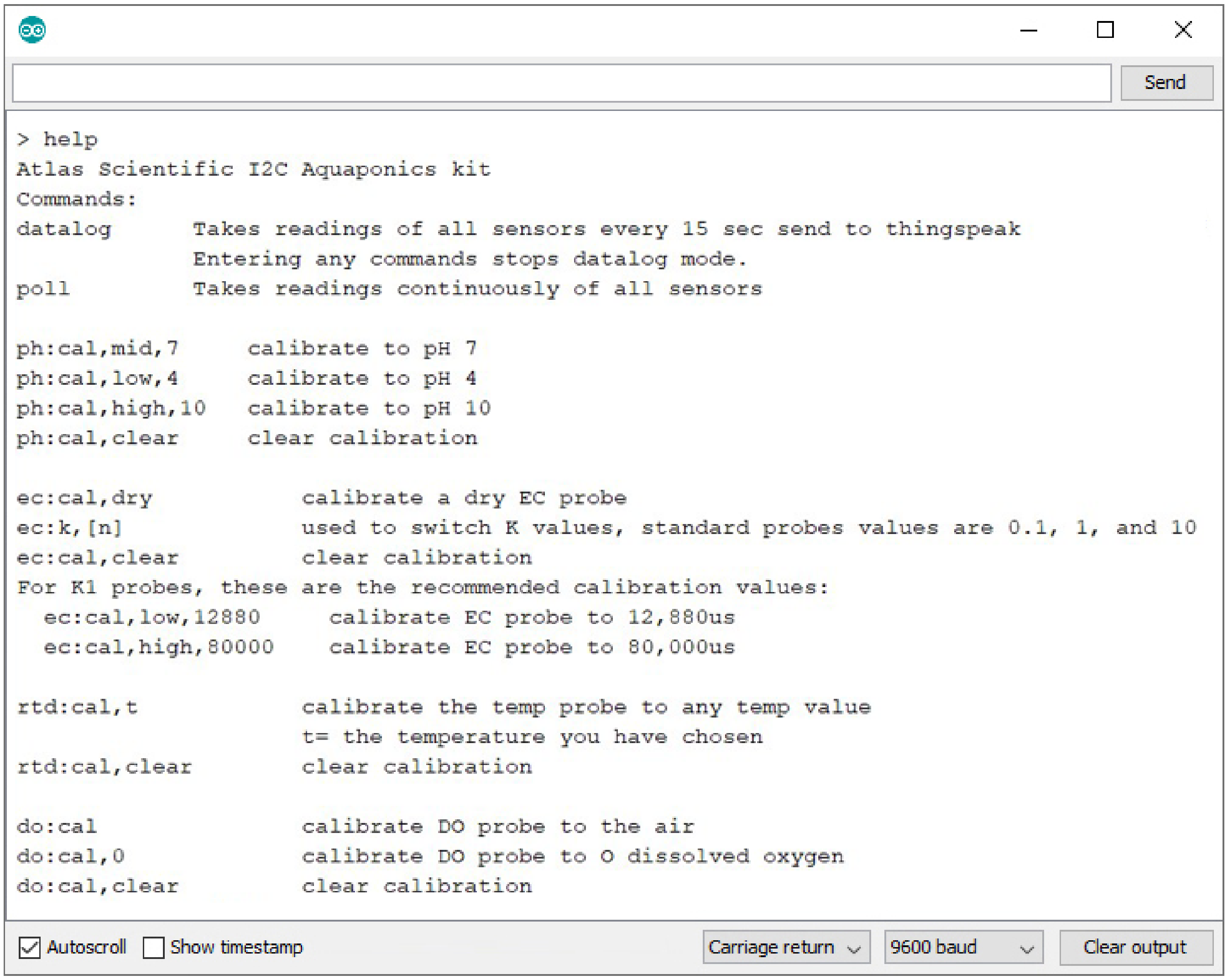


```
RTD: 24.15   PH: 4.69  DO: 8.55  EC: 39.62  HUM: 33.70  CO2: 1061.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.25  HUM: 33.73  CO2: 1058.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.24  HUM: 33.72  CO2: 1056.00
RTD: 24.17   PH: 4.70  DO: 8.55  EC: 39.07  HUM: 33.71  CO2: 1055.00
connecting to wifi
RTD: 24.15   PH: 4.69  DO: 8.55  EC: 39.62  HUM: 33.70  CO2: 1061.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.25  HUM: 33.73  CO2: 1058.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.24  HUM: 33.72  CO2: 1056.00
RTD: 24.17   PH: 4.70  DO: 8.55  EC: 39.07  HUM: 33.71  CO2: 1055.00
connecting to wifi
RTD: 24.15   PH: 4.69  DO: 8.55  EC: 39.62  HUM: 33.70  CO2: 1061.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.25  HUM: 33.73  CO2: 1058.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.24  HUM: 33.72  CO2: 1056.00
RTD: 24.17   PH: 4.70  DO: 8.55  EC: 39.07  HUM: 33.71  CO2: 1055.00
connecting to wifi
RTD: 24.15   PH: 4.69  DO: 8.55  EC: 39.62  HUM: 33.70  CO2: 1061.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.25  HUM: 33.73  CO2: 1058.00
RTD: 24.16   PH: 4.69  DO: 8.55  EC: 39.24  HUM: 33.72  CO2: 1056.00
```

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.



```
> help
Atlas Scientific I2C Aquaponics kit
Commands:
datalog      Takes readings of all sensors every 15 sec send to thingspeak
             Entering any commands stops datalog mode.
poll         Takes readings continuously of all sensors

ph:cal,mid,7     calibrate to pH 7
ph:cal,low,4     calibrate to pH 4
ph:cal,high,10   calibrate to pH 10
ph:cal,clear     clear calibration

ec:cal,dry           calibrate a dry EC probe
ec:k,[n]             used to switch K values, standard probes values are 0.1, 1, and 10
ec:cal,clear         clear calibration
For K1 probes, these are the recommended calibration values:
  ec:cal,low,12880     calibrate EC probe to 12,880us
  ec:cal,high,80000    calibrate EC probe to 80,000us

rtd:cal,t            calibrate the temp probe to any temp value
                     t= the temperature you have chosen
rtd:cal,clear        clear calibration

do:cal               calibrate DO probe to the air
do:cal,0             calibrate DO probe to 0 dissolved oxygen
do:cal,clear         clear calibration
```

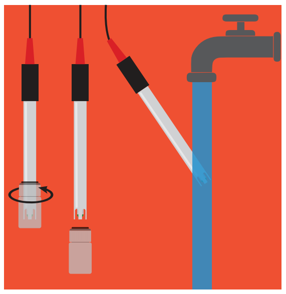
A The poll command

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.

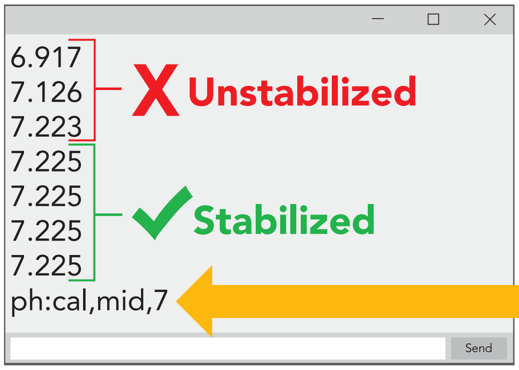
Remove the soaker bottle and rinse off the pH probe. Remove the top of the pH 7.00 calibration solution pouch. Place the pH probe inside the pouch and let the probe sit in the calibration solution until the readings stabilize. This will take about 1 – 2 mins.



AtlasScientific
pH 7.00
pH Calibration Solution

pH 7.00
pH Calibration Solution
1 – 2 mins

pH 7.00
After Calibration



```
6.917
7.126
7.223
7.225
7.225
7.225
7.225
ph:cal,mid,7
```

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.

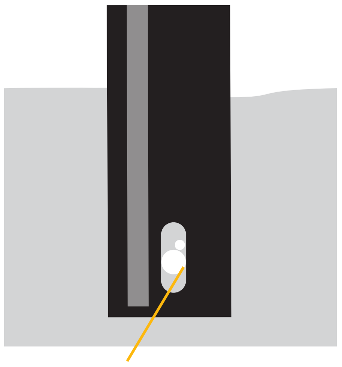
Rinse off the probe and repeat this process for both **pH 4.00** and **pH 10.00**.

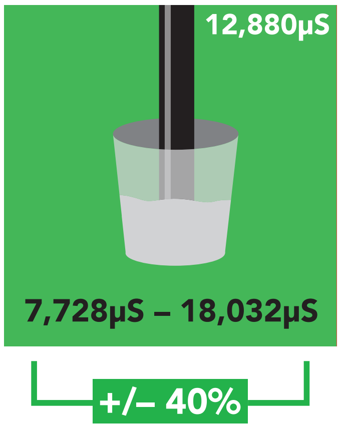
C Calibrate Conductivity

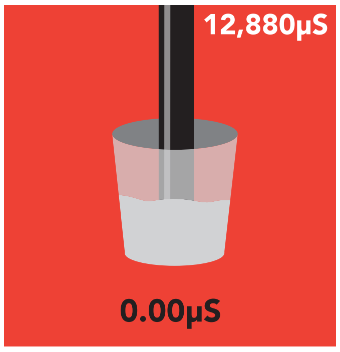
When calibrating Conductivity, you must always calibrate a dry probe first.

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

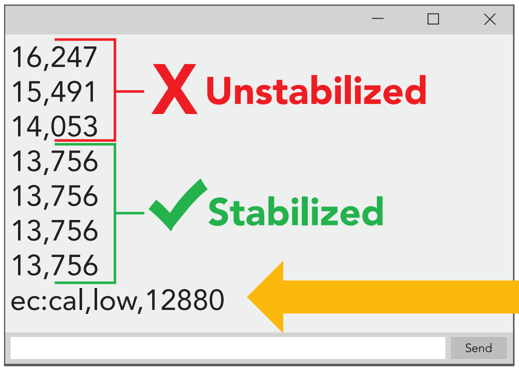
Once the dry calibration has been completed, place the probe into a small cup of the 12,880μS calibration solution. Shake the probe to make sure you do not have trapped air bubbles in the sensing area. You should see readings that are off by **1 – 40%** from the stated value of the calibration solution.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


Trapped air in sensing area (shake to remove)



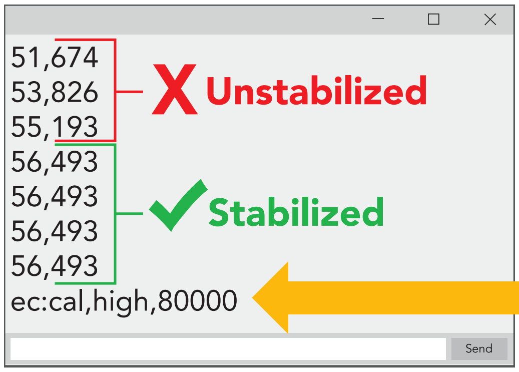


check probe connection, you cannot calibrate to 0.


```
16,247
15,491
14,053
13,756
13,756
13,756
13,756
ec:cal,low,12880
```

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

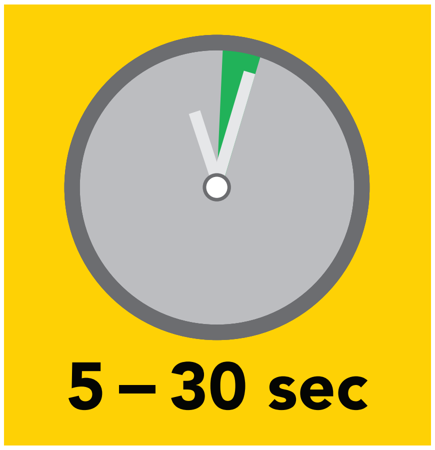
Rinse off the probe before calibrating to the high point. Pour a small amount of the 80,000μS calibration solution into a cup. Shake the probe to remove trapped air. Again, the readings may be off by **1 – 40%** Wait for readings to stabilize.

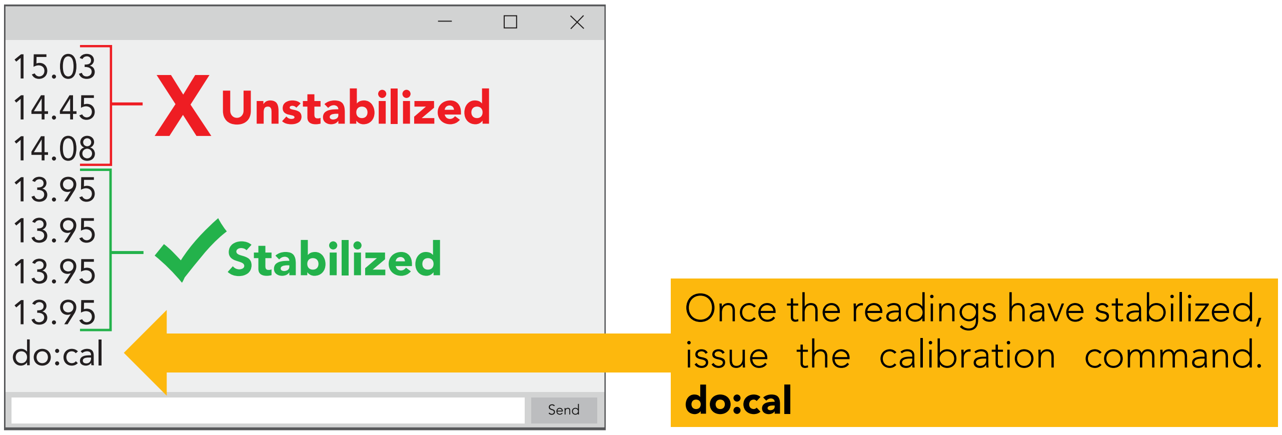

```
51,674
53,826
55,193
56,493
56,493
56,493
56,493
ec:cal,high,80000
```

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).*





```
15.03
14.45
14.08
13.95
13.95
13.95
13.95
do:cal
```

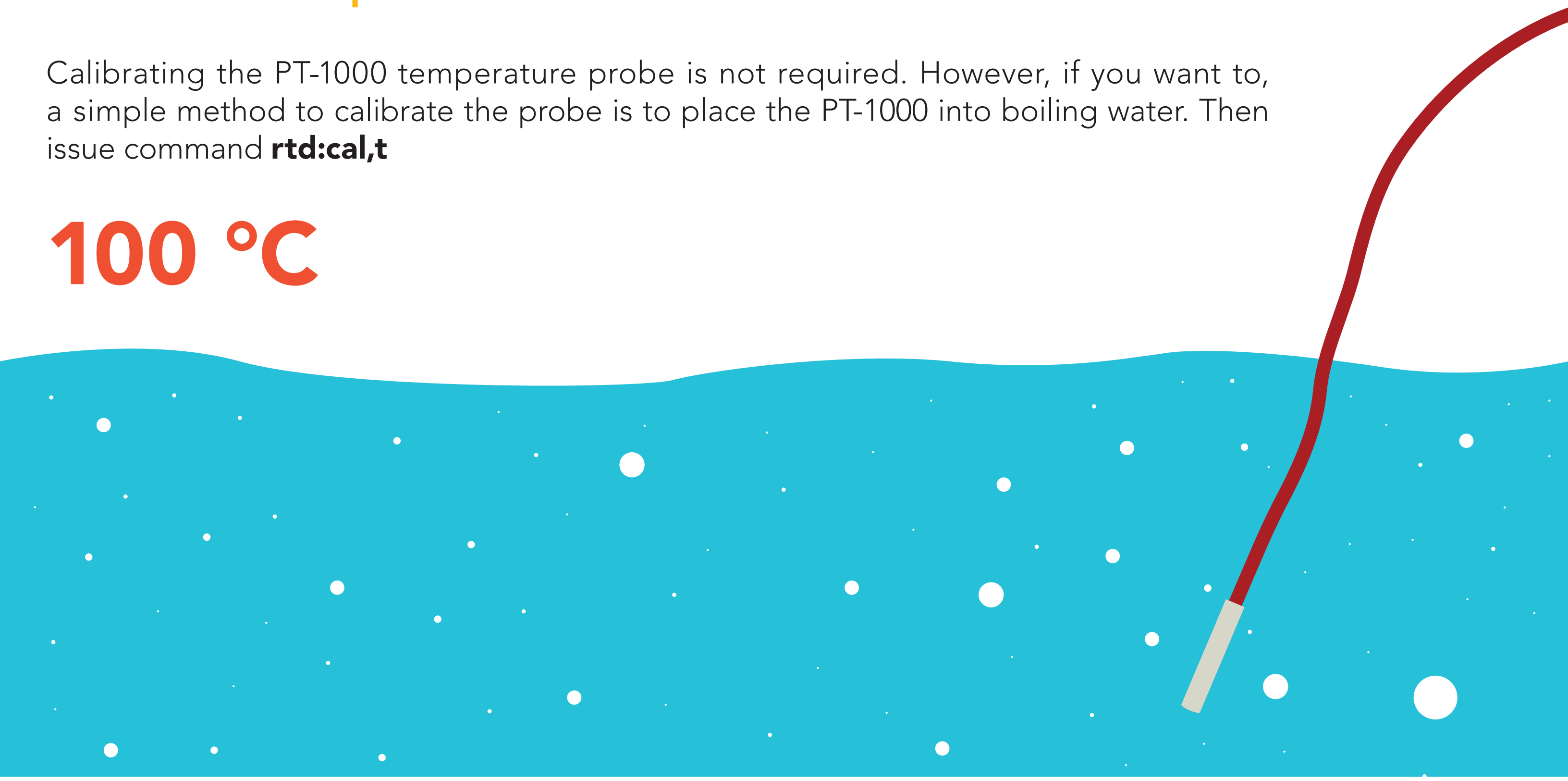
Once the readings have stabilized, issue the calibration command. **do:cal**

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)

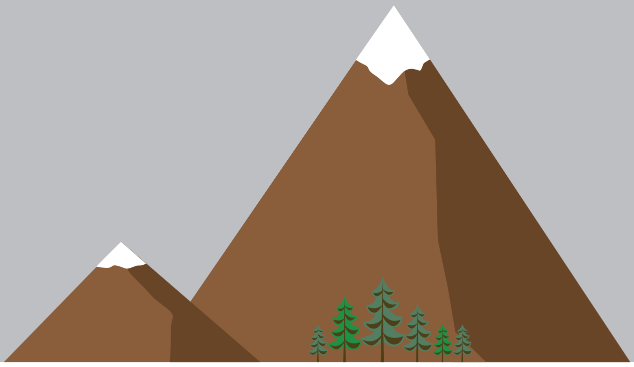
E Calibrate Temperature

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!