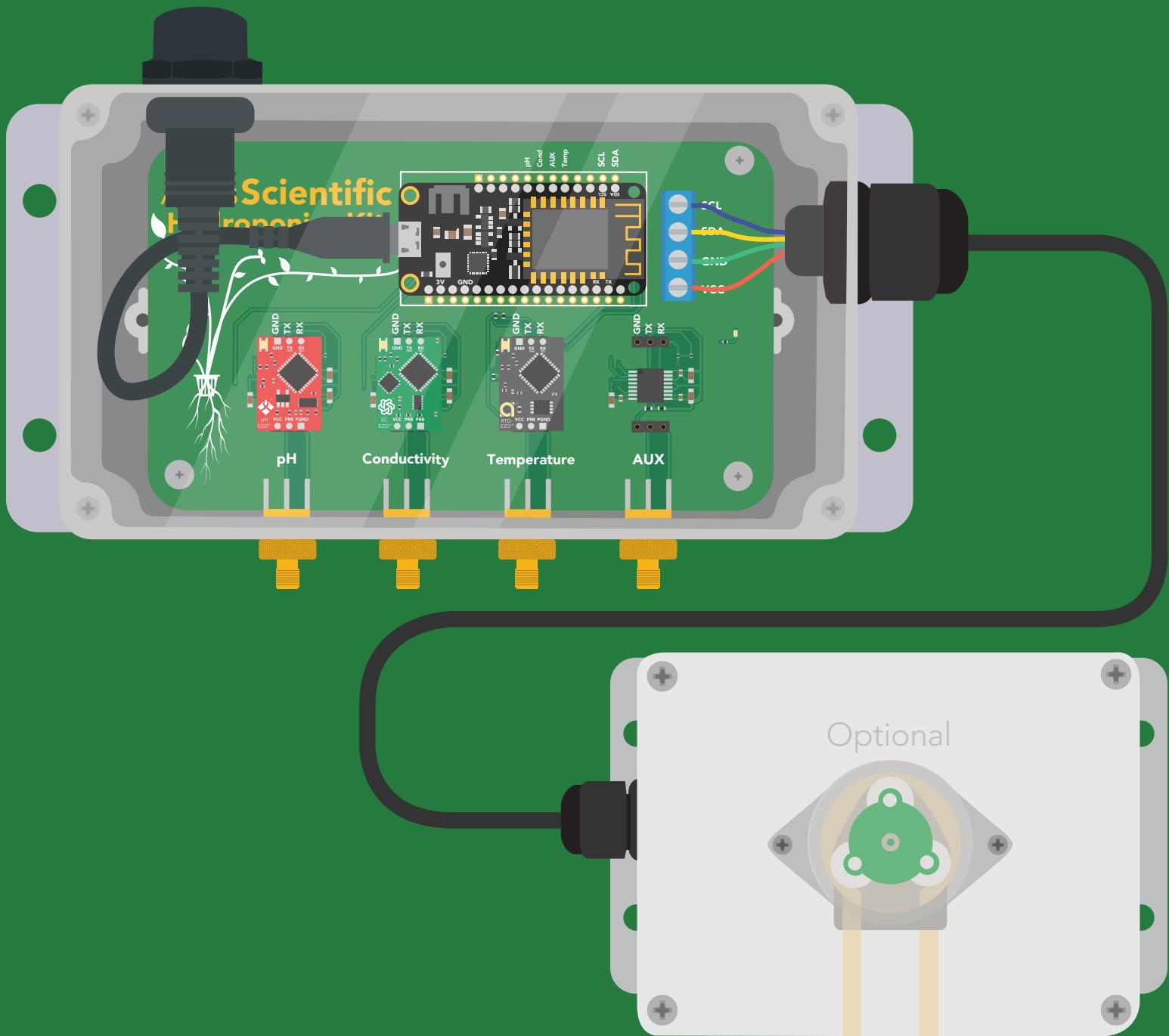


Wi-Fi Hydroponics Kit

Datasheet

V 2.0

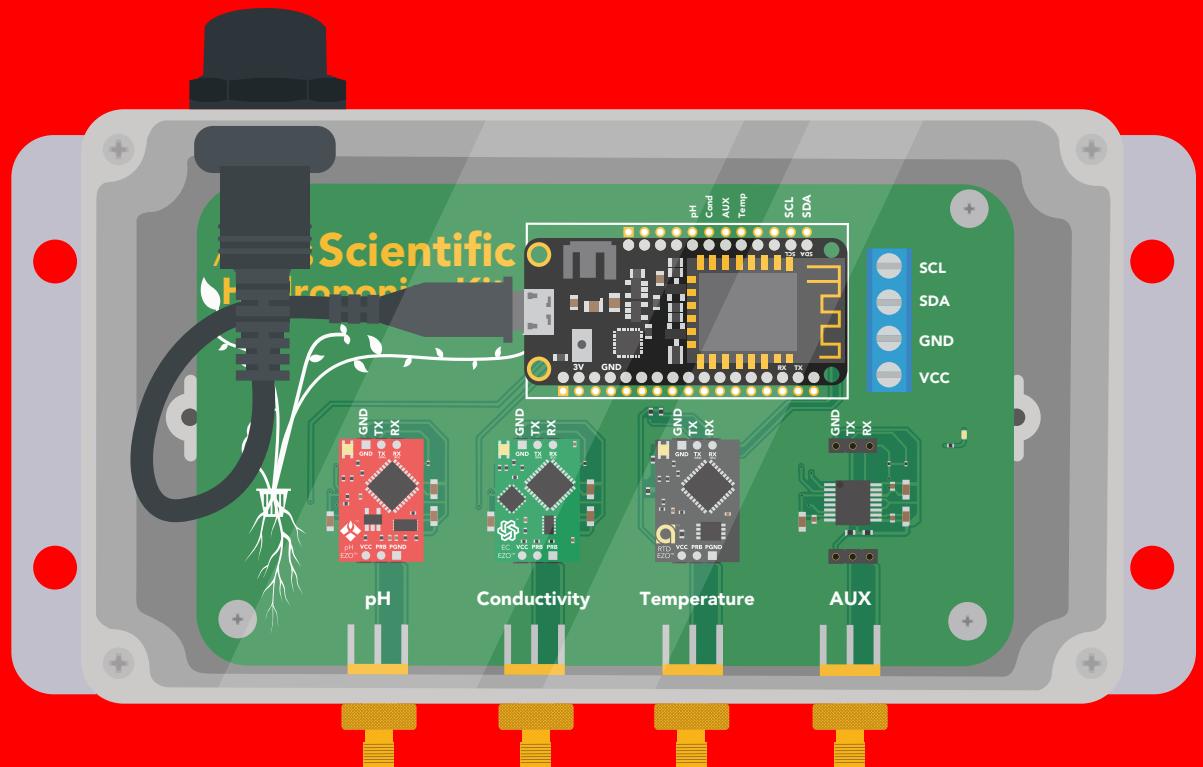


STOP

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)

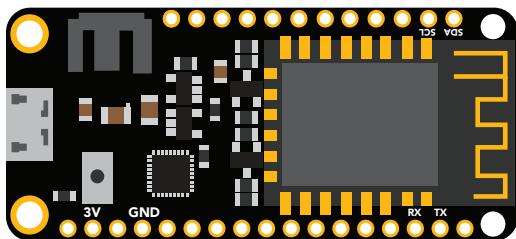
Operating principle

The Wi-Fi 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™, 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

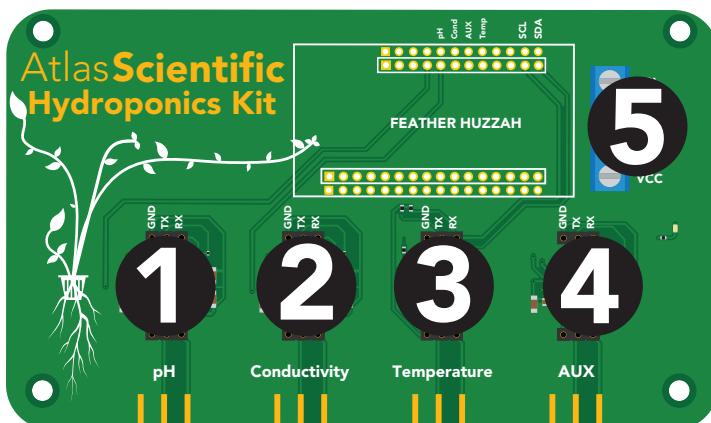
The Wi-Fi hydroponics kit is controlled using an Adafruit HUZZAH32 as its CPU. The HUZZAH is programmed using the Arduino IDE and uses an onboard ESP32 as its Wi-Fi transmitter. [Adafruit HUZZAH32 datasheet](#).



Sensor ports

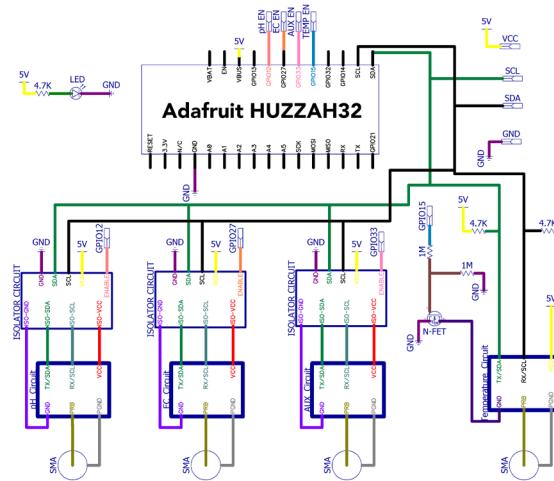
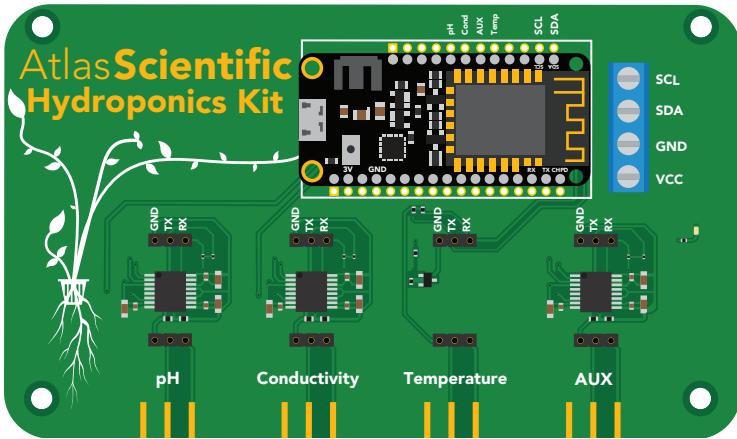
The Wi-Fi hydroponics kit PCB has 5 sensor ports. Three of the ports are electrically isolated. The isolated ports are marked pH, Conductivity, and AUX. The isolated ports are needed to take noise-free electrochemical readings. Because the sensing element of a temperature sensor is never in direct contact 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.

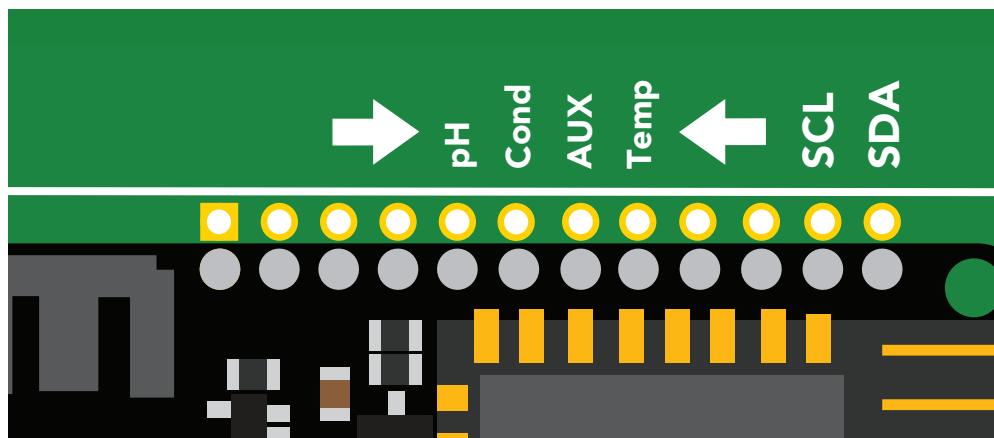


PCB

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



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	Adafruit Huzzah32 GPIO	State	Sensor Power
pH	12	LOW	ON
Cond	27	LOW	ON
Aux	33	LOW	ON
Temp	15	HIGH	ON

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

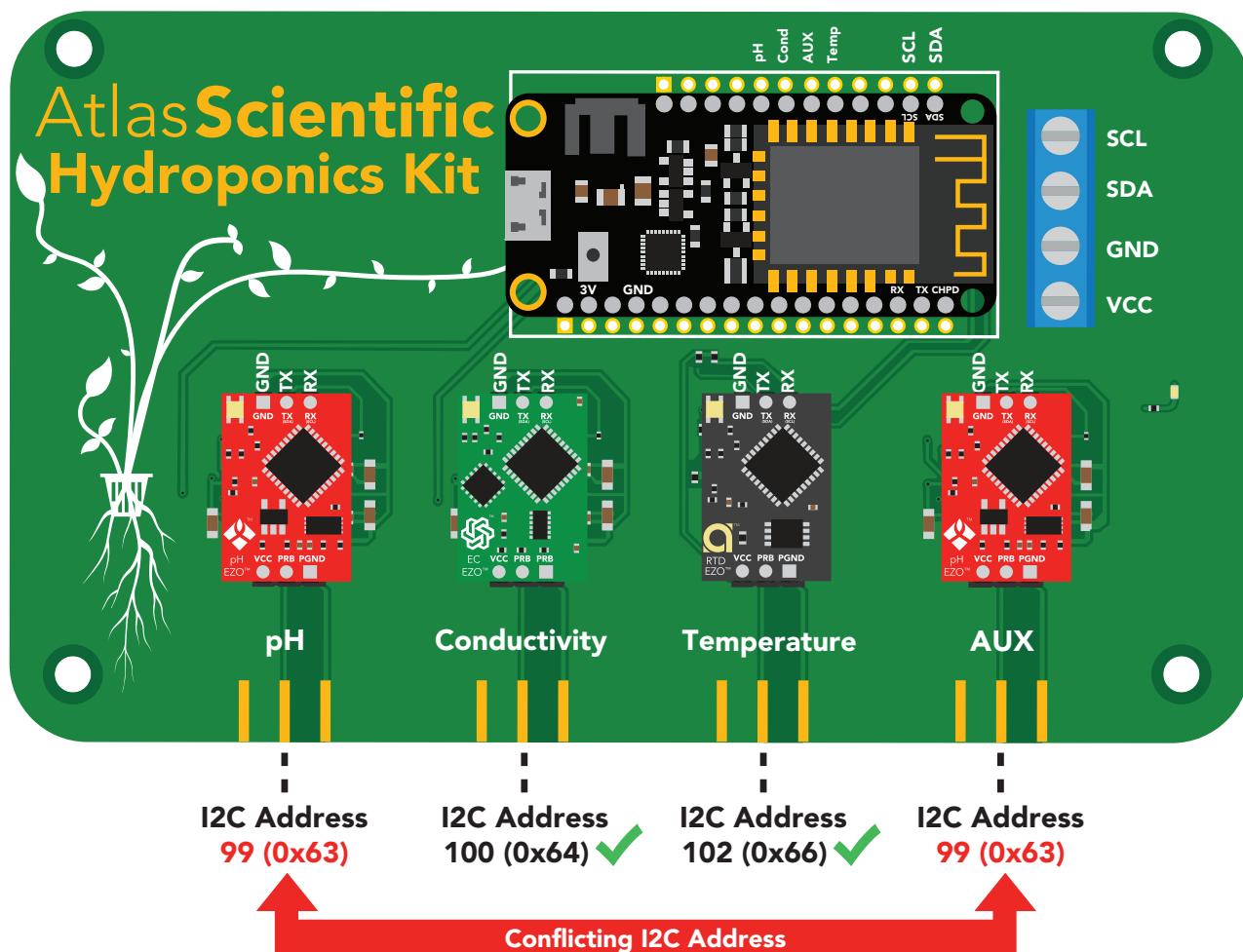
Data protocol

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.

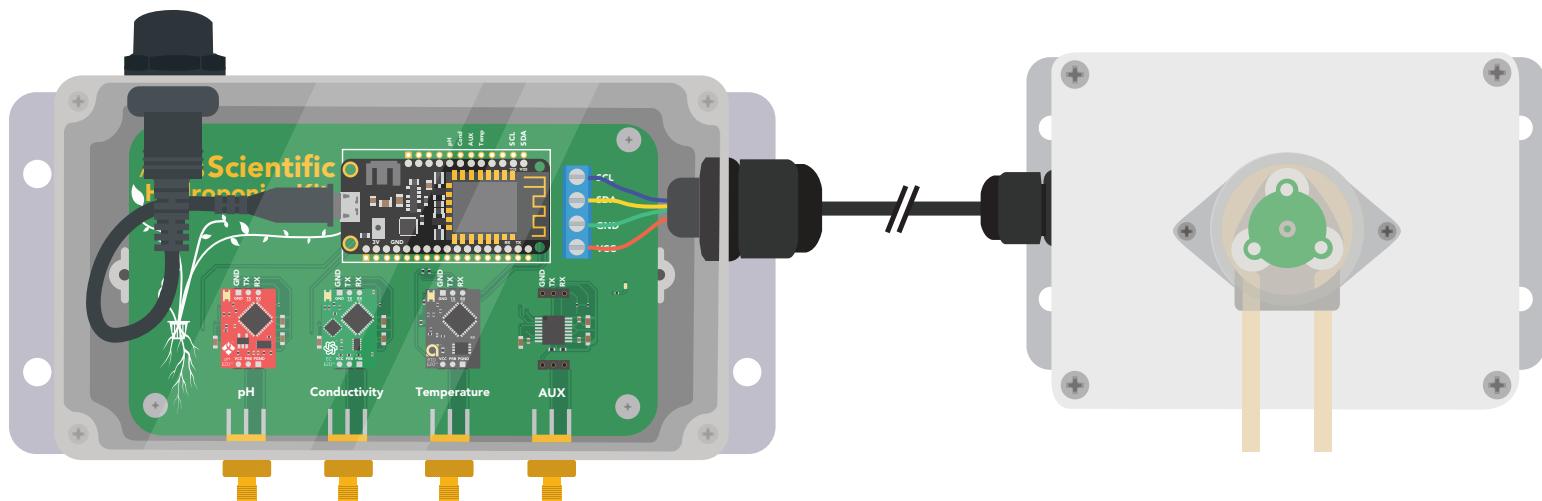


This table lists the default I2C address of components commonly added to this kit.

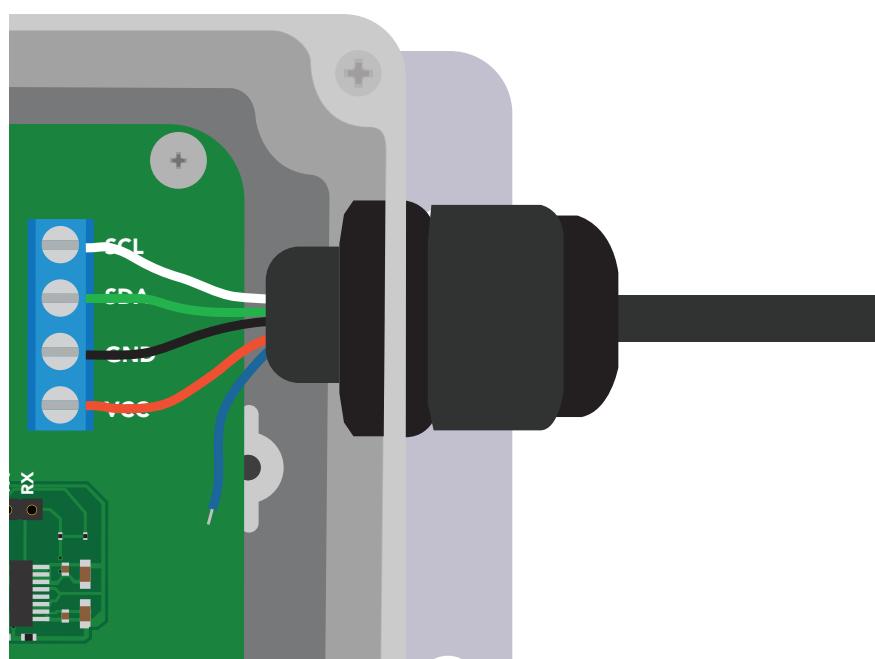
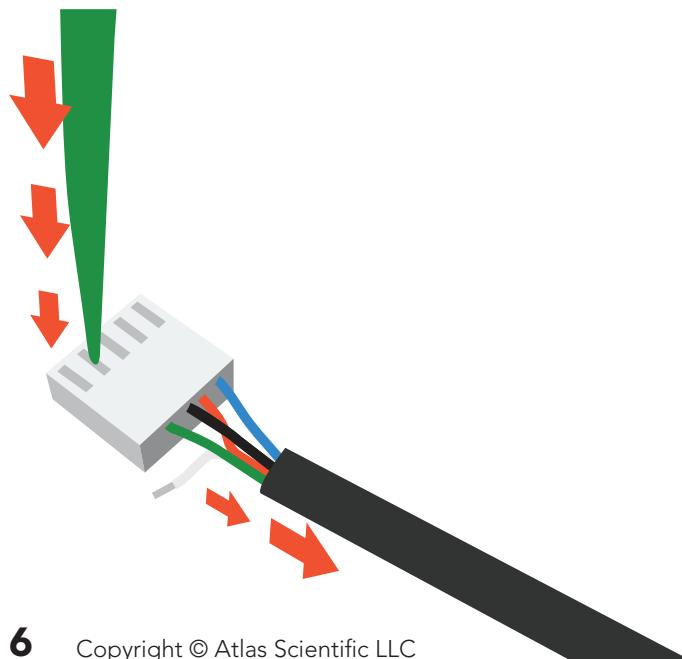
Device	I2C Address	Device	I2C Address
EZO pH	99 (0x63)	EZO EC	100 (0x64)
EZO ORP	98 (0x62)	EZO RTD	102 (0x66)
EZO DO	97 (0x61)	EZO PMP	103 (0x67)

Dosing pump

An optional external dosing pump can be added to the Wi-Fi hydroponics 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 Setup a ThingSpeak Account

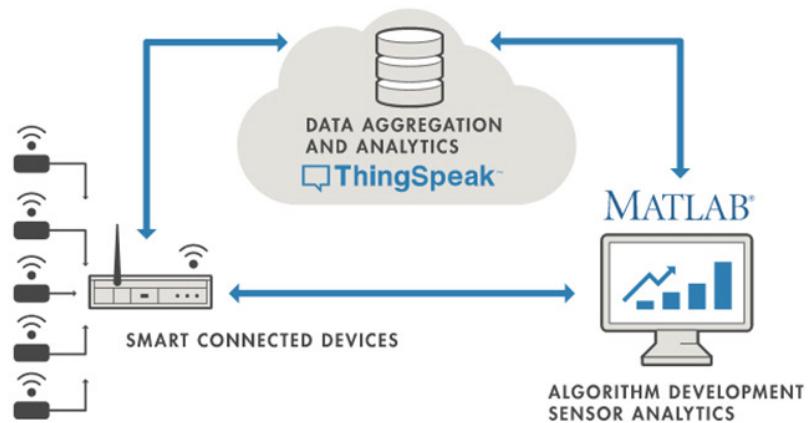
Because the sensor data is stored / viewed on ThingSpeak, you will need to setup a ThingSpeak account. Create your ThingSpeak account by clicking [HERE](#).



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 ThingSpeak, 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.

A screenshot of a MathWorks sign-in form. It has fields for 'Email' and a 'Create one!' button. Below the form, there's a note about agreeing to privacy policy and a 'Next' button.

Step 2 Create a Channel

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

The screenshot shows the ThingSpeak interface. At the top, there's a navigation bar with links for 'Channels', 'Apps', 'Support', 'Commercial Use', and 'How to Buy'. Below the navigation bar, the main title 'My Channels' is displayed. On the left, a green button labeled 'New Channel' is highlighted with a yellow arrow pointing towards it. In the center, there's a search bar with the placeholder 'Search by tag' and a magnifying glass icon. To the right, a 'Help' section is visible, containing text about collecting data and creating new channels.

The screenshot shows the 'New Channel' creation page. The 'Name' field contains 'Atlas Sensors' and is highlighted with a yellow box. The 'Description' field is empty. Below, there are three 'Field' sections: 'Field 1' with 'pH', 'Field 2' with 'EC (µS/cm)', and 'Field 3' with 'Temp (°C)'. Each field has a corresponding input box and a checked checkbox. To the right, a 'Help' section provides information about channels and channel settings, including descriptions of 'Percentage complete', 'Channel Name', and 'Description'.

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**.

The screenshot shows the ThingSpeak interface. At the top, there's a navigation bar with links for 'Channels', 'Apps', 'Support', 'Commercial Use', and 'How to Buy'. Below the navigation bar is a search bar labeled 'Search by tag' with a magnifying glass icon. On the left, there's a green button labeled 'New Channel'. The main area is titled 'My Channels' and displays a table with one row for the 'Atlas Sensors' channel. The table columns are 'Name', 'Created', and 'Updated'. The 'Name' column contains 'Atlas Sensors' with a lock icon. The 'Created' column shows '2020-02-14' and the 'Updated' column shows '2020-05-11 23:04'. Below the table are buttons for 'Private', 'Public', 'Settings', 'Sharing', 'API Keys', and 'Data Import / Export'. To the right of the table is a 'Help' section with several paragraphs of text and links.

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](#).

The screenshot shows the 'Atlas Sensors' channel page. At the top, there's a navigation bar with links for 'Channels', 'Apps', 'Support', 'Commercial Use', and 'How to Buy'. Below the navigation bar is a box containing 'Channel ID: xxxxxx', 'Author:', and 'Access: Private'. The main area has tabs for 'Private View', 'Public View', 'Channel Settings', 'Sharing', 'API Keys' (which is highlighted with a yellow box), and 'Data Import / Export'. Below these tabs is a section titled 'Write API Key' with a 'Key' input field containing 'xxxxxxxxxxxxxxxxxxxx'. A yellow box highlights this input field. At the bottom of this section is a button labeled 'Generate New Write API Key'. To the right of this section is a 'Help' section with text about API keys and their uses, and a 'API Keys Settings' section with a bulleted list of instructions.

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

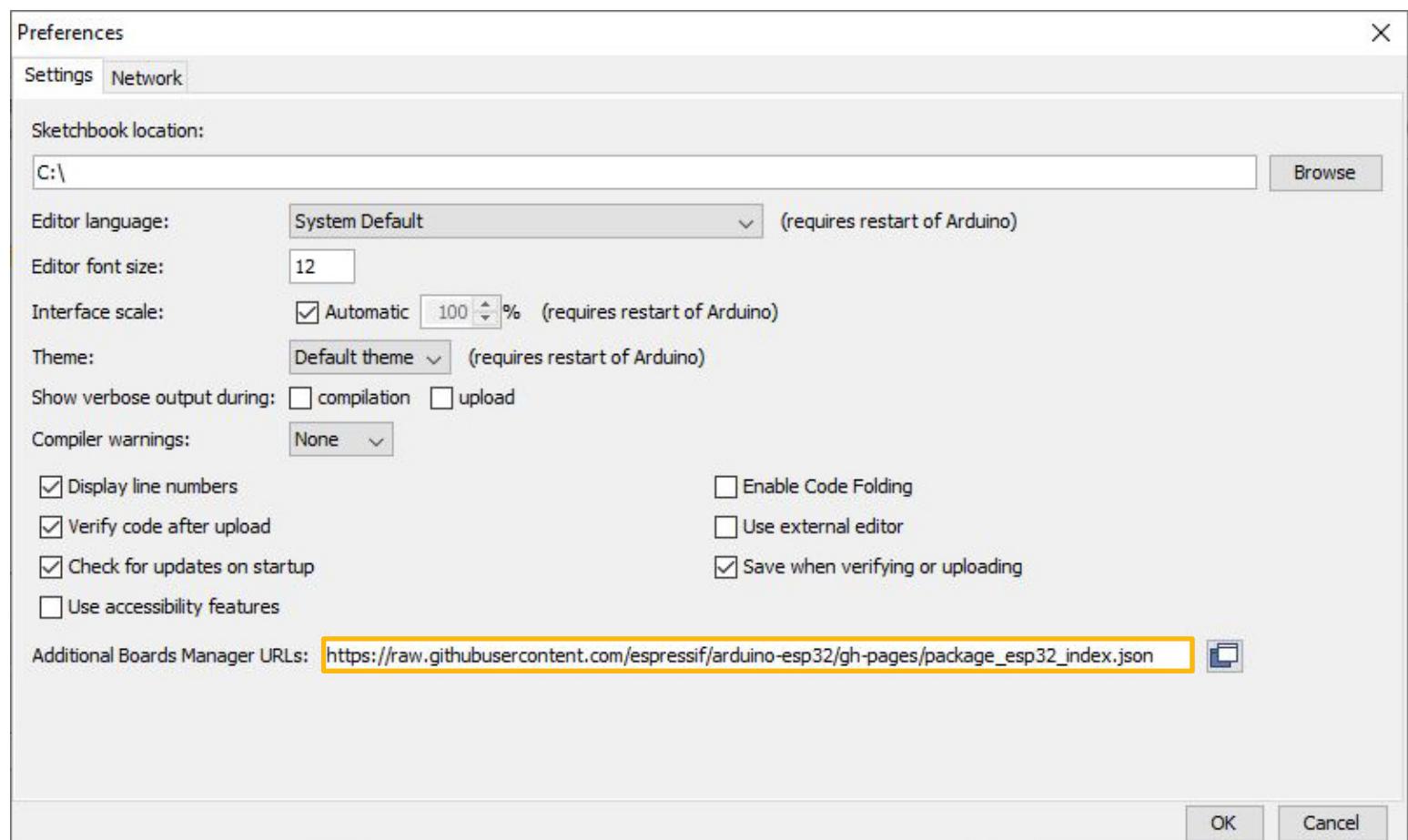
Be sure to save your **Channel ID** and **Write API Key** we are going to need these, in the next few steps.

Step 4 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.



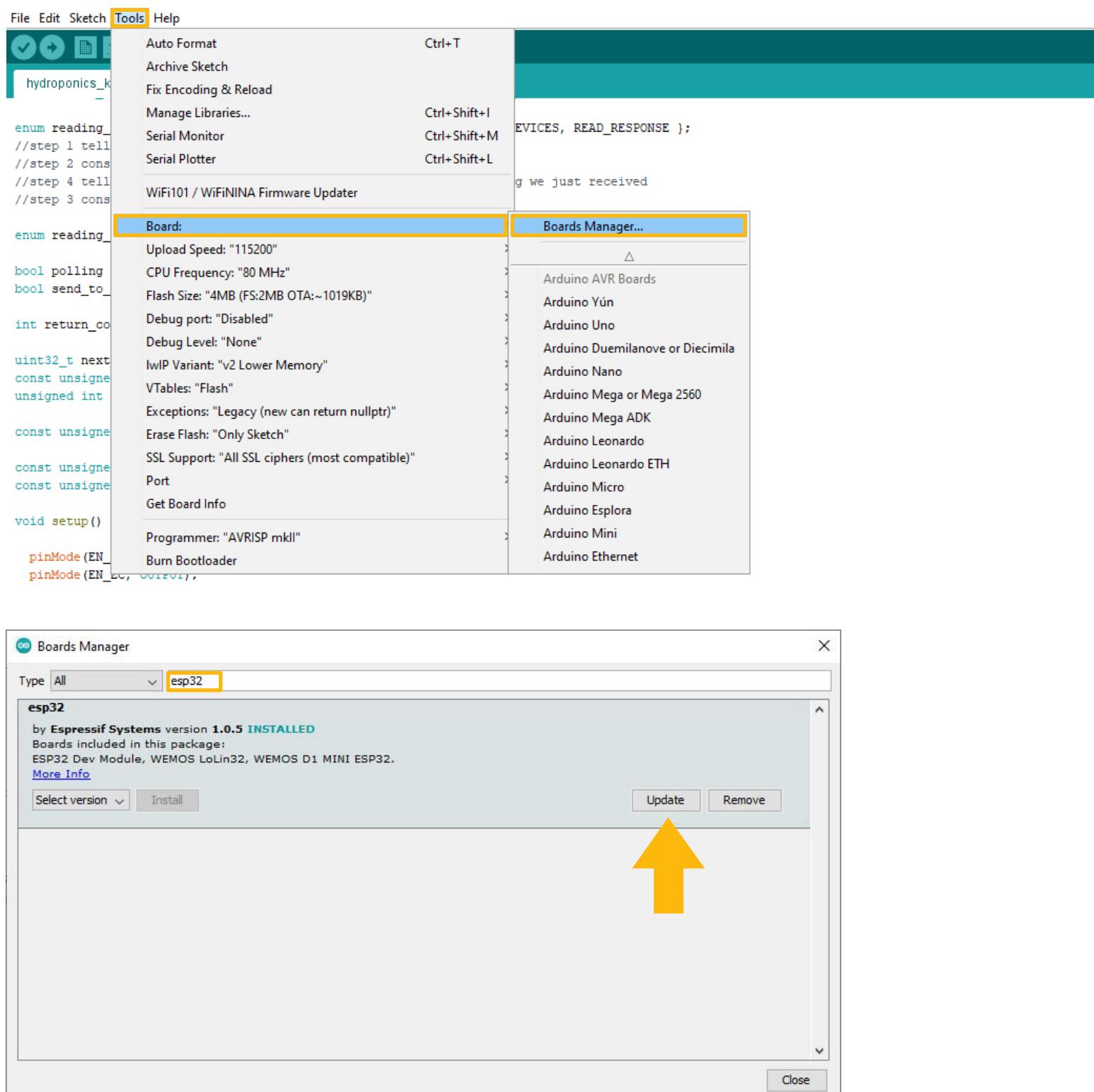
Make sure this URL is in the textbox

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 is not the most recent version)

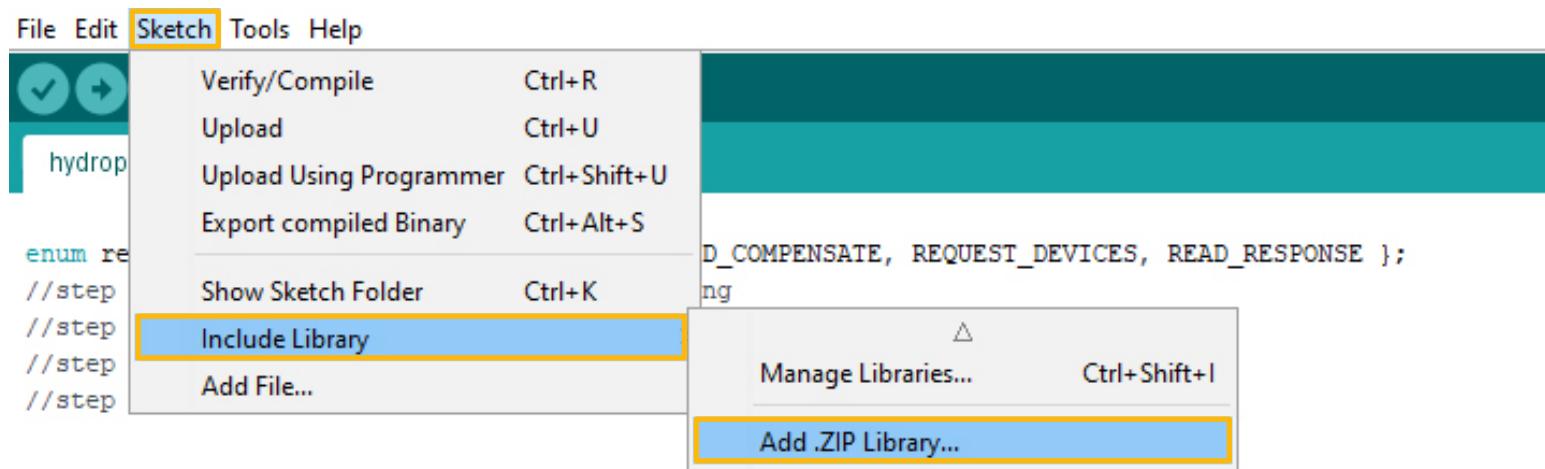
C Download the ThingSpeak library for Arduino

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](#).

A screenshot of a GitHub repository page for 'Atlas-Scientific/Ezo_I2c'. The page shows basic repository statistics: 'master' branch, '1 branch', '0 tags'. On the right, there are download and cloning options: 'Clone' via HTTPS or GitHub CLI, 'Open with GitHub Desktop', and a large yellow arrow pointing to the 'Download ZIP' button. The main area lists files and their descriptions: Examples (removed redundant examples, aquaponics), 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), and README.md (Update README.md). The last update was 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 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> hydroponics_kit

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The left sidebar shows the file structure:

- File
- Edit
- Sketch
- Tools
- Help
- New Ctrl+N
- Open... Ctrl+O
- Open Recent >
- Sketchbook >
- Examples** > **EZO_I2C_lib-master** > **IOT_kits** > **hydroponics_kit**
- LittleFS
- NetBIOS
- Preferences
- SD
- SD_MMC
- SimpleBLE
- SPI
- SPIFFS
- Ticker
- Update
- USB
- WebServer
- WiFi
- WiFiClientSecure
- WiFiProv
- Examples from Custom Libraries
- ACROBOTIC SSD1306
- Adafruit BME280 Library
- Adafruit BME680 Library
- Adafruit BusIO
- Adafruit ESP8266
- Adafruit GFX Library
- Adafruit HTU21DF Library
- Adafruit LiquidCrystal
- Adafruit SHT31 Library
- Adafruit Si7021 Library
- atlas_gravity
- EspSoftwareSerial
- EZO_I2C_lib-master** > **Examples** > **I2c_lib_examples** > **IOT_kits** > **hydroponics_kit**
- Ezo_uart_lib-master
- MCUFRIEND_kbv
- PID
- StepperDriver
- ThingSpeak
- INCOMPATIBLE

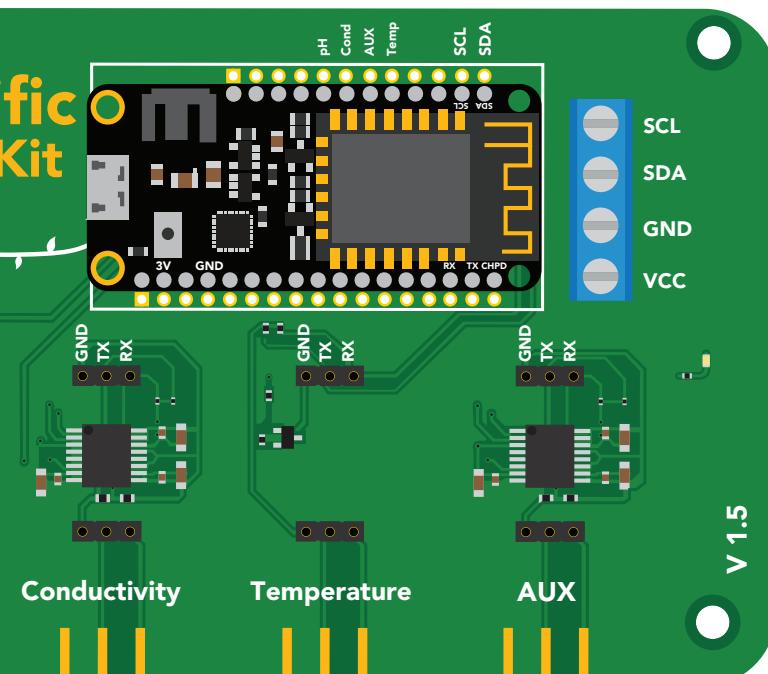
The main code area displays the `hydroponics_kit` sketch, which includes comments for WiFi and Thingspeak configuration, and definitions for various sensor objects (PH, EC, RTD, PMP) and a board array.

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)

```
3 #include <iot_cmd.h>                                //include wifi library
4 #include <WiFi.h>                                    //include thingspeak library
5 #include "ThingSpeak.h"                               //imports a 4 function sequencer
6 #include <sequencer4.h>                             //imports a 1 function sequencer
7 #include <sequencer1.h>                             //brings in common print statements
8 #include <Ezo_i2c_util.h>                           //include arduinos i2c library
9 #include <Ezo_i2c.h> //include the EZO I2C library from https://github.com/Atlas-Scientific/Ezo\_I2c\_lib
10 #include <Wire.h>        //include arduinos i2c library
11
12 WiFiClient client;                                //declare that this device connects to a Wi-Fi network,c
13
14 //-----Fill in your Wi-Fi / ThingSpeak Credentials-----
15 const String ssid = "Wifi Name";                  //The name of the Wi-Fi network you are connecting to
16 const String pass = "Wifi Password";              //Your WiFi network password
17 const long myChannelNumber = 1234566;            //Your Thingspeak channel number
18 const char * myWriteAPIKey = "XXXXXXXXXXXXXXXXXX"; //Your ThingSpeak Write API Key
19 //-----
```

C Choose enable pins



```
38 //-----For version 1.4 use these enable pins for each circuit-----
39 //const int EN_PH = 13;
40 //const int EN_EC = 12;
41 //const int EN_RTD = 33;
42 //const int EN_AUX = 27;
43 //-----
44
45 //-----For version 1.5 use these enable pins for each circuit-----
46 const int EN_PH = 12;
47 const int EN_EC = 27;
48 const int EN_RTD = 15;
49 const int EN_AUX = 33;
50 //-----
```

If version 1.4
use these enable pins.

If version 1.5
use these enable pins.

D Setting up your pump

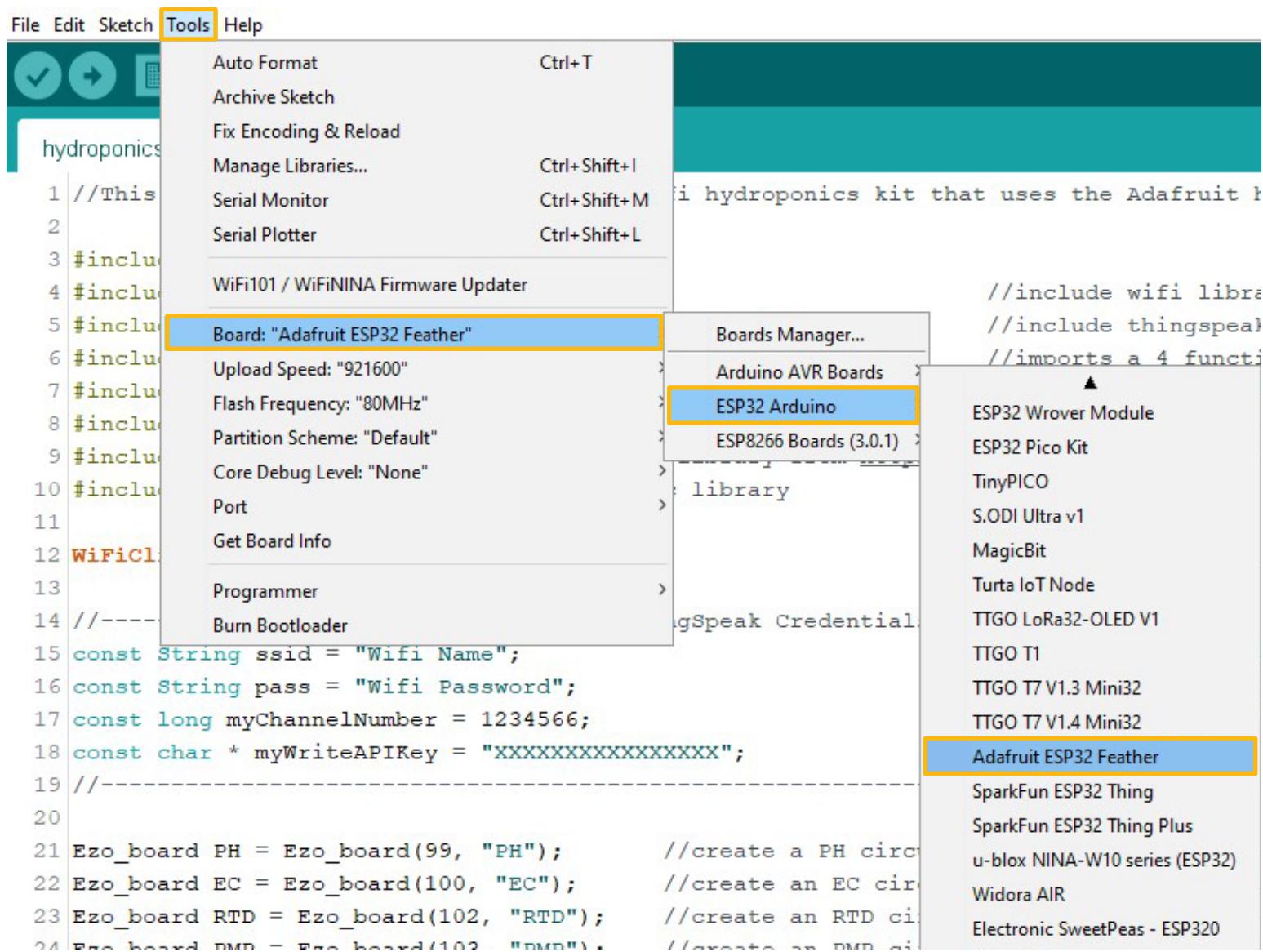
If you do not have a pump attached, you can just skip this part. The code is rather self explanatory. You set what parameters will trigger the pump to engage.

```
57 //parameters for setting the pump output
58 #define PUMP_BOARD      PMP      //the pump that will do the output (if theres more than one)
59 #define PUMP_DOSE        -0.5    //the dose that the pump will dispense in milliliters
60 #define EZO_BOARD        EC       //the circuit that will be the target of comparison
61 #define IS_GREATER_THAN  true    //true means the circuit's reading has to be greater than the comparison value,
62 #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

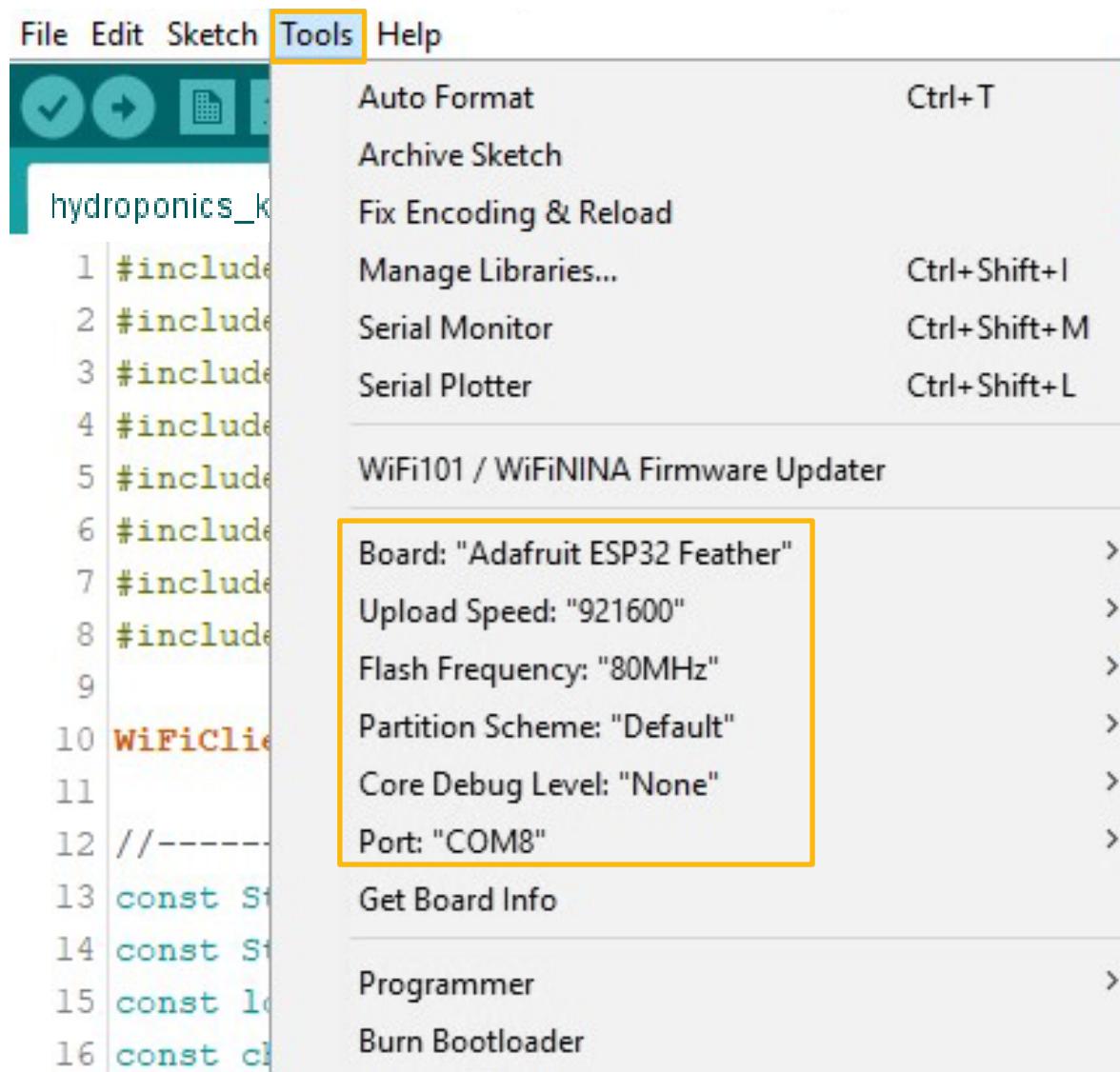
Tools > Board > ESP32 Arduino > Adafruit ESP32 Feather



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.
Don't forget to set the correct com port for your device.)



C Compile and upload

hydroponics_kit | Arduino 1.8.13

File Edit Sketch Tools Help



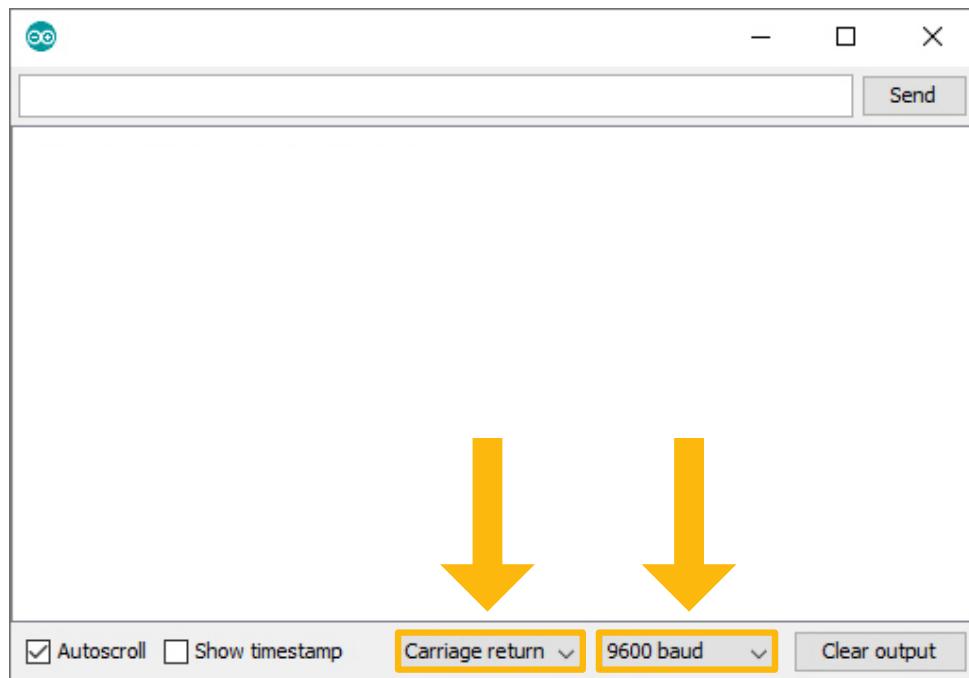
Compile and upload the code.

Step 7

See the readings

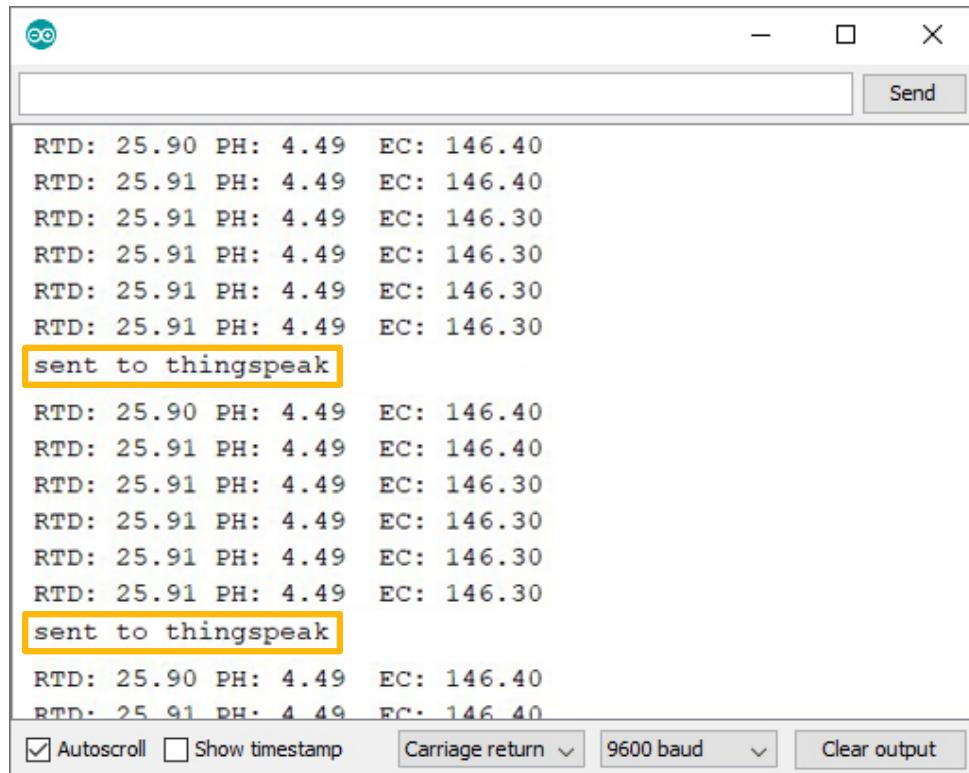
Open your Arduino serial monitor.

(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 Hydroponics Meter will always attempt to connect to ThingSpeak on bootup.



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

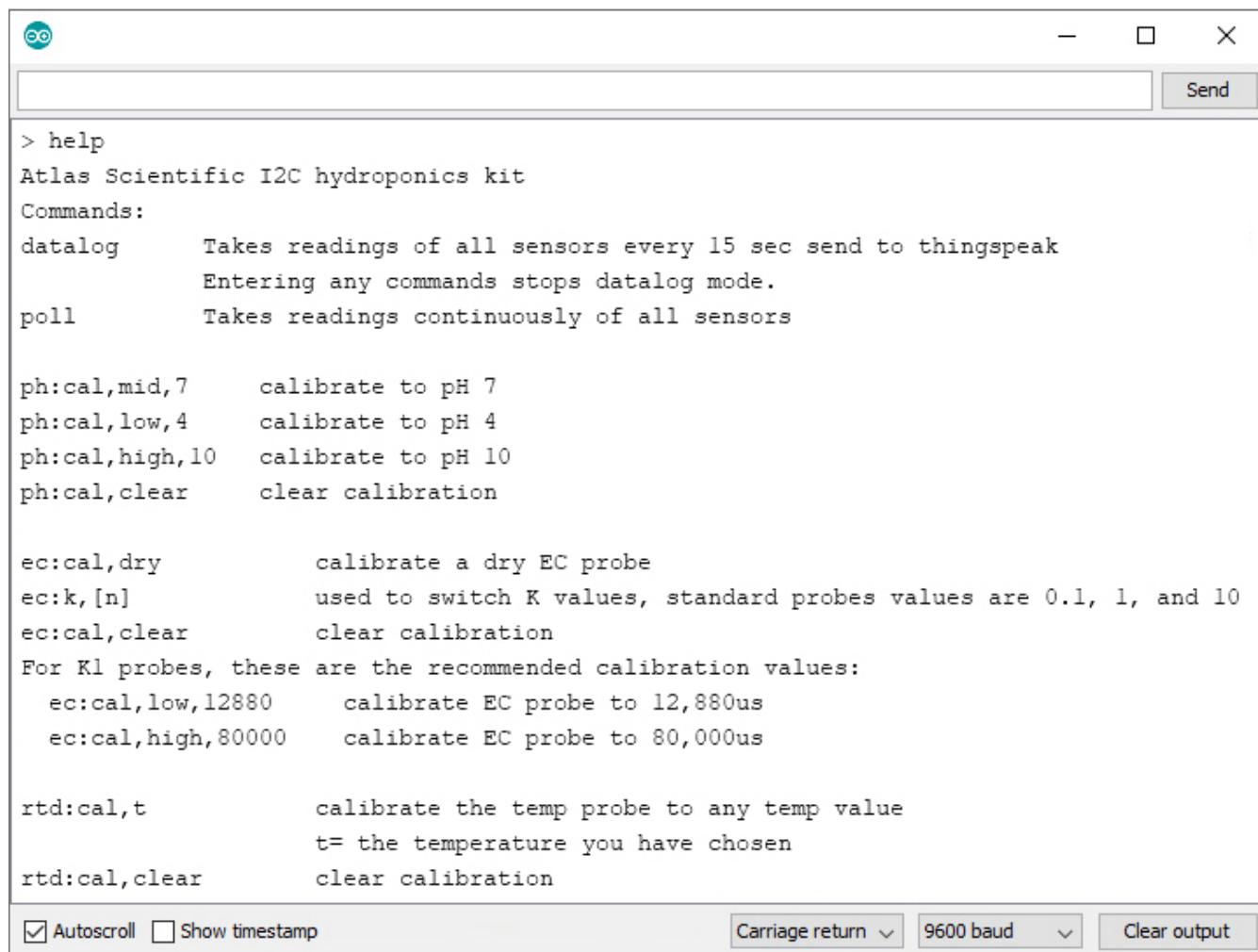
```
RTD: 25.76 PH: 4.49 EC: 143.20
RTD: 25.76 PH: 4.49 EC: 143.20
RTD: 25.76 PH: 4.49 EC: 143.20
RTD: 25.76 PH: 4.49 EC: 143.00
connecting to wifi
RTD: 25.76 PH: 4.49 EC: 143.00
RTD: 25.76 PH: 4.49 EC: 143.00
RTD: 25.76 PH: 4.49 EC: 143.00
RTD: 25.76 PH: 4.49 EC: 142.80
connecting to wifi
RTD: 25.76 PH: 4.49 EC: 143.20
RTD: 25.76 PH: 4.49 EC: 143.20
RTD: 25.76 PH: 4.49 EC: 143.20
RTD: 25.76 PH: 4.49 EC: 143.00
connecting to wifi
RTD: 25.76 PH: 4.49 EC: 143.00
RTD: 25.76 PH: 4.49 EC: 143.00
RTD: 25.76 PH: 4.49 EC: 143.00
RTD: 25.76 PH: 4.49 EC: 142.80
```

Autoscroll Show timestamp

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 screenshot shows a terminal window with the following content:

```
> help
Atlas Scientific I2C hydroponics 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

 Autoscroll  Show timestamp   
```

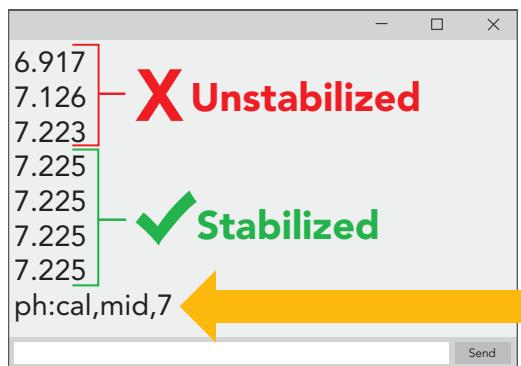
A The poll command

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

Setting the Conductivity probe type

If your probe ≠ K 1.0 (default), then set the probe type by using the **ec:k,n** command. (where n = K value of your probe)

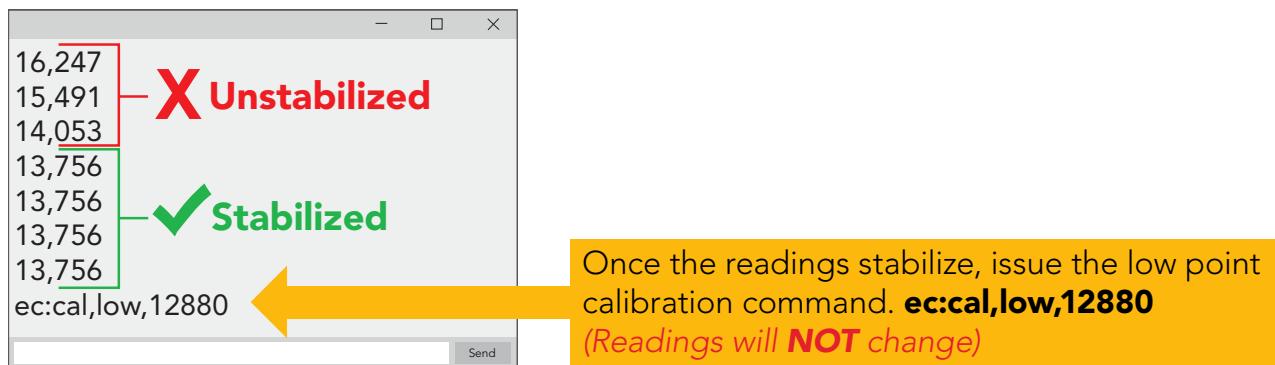
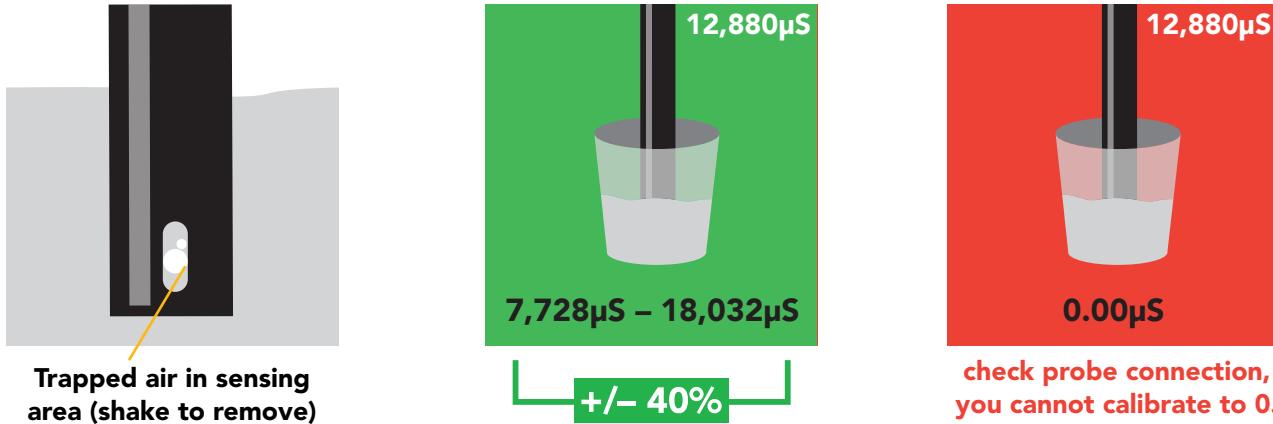
Example, if you have a K 0.1 conductivity probe issue the command **ec:k,0.1**

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

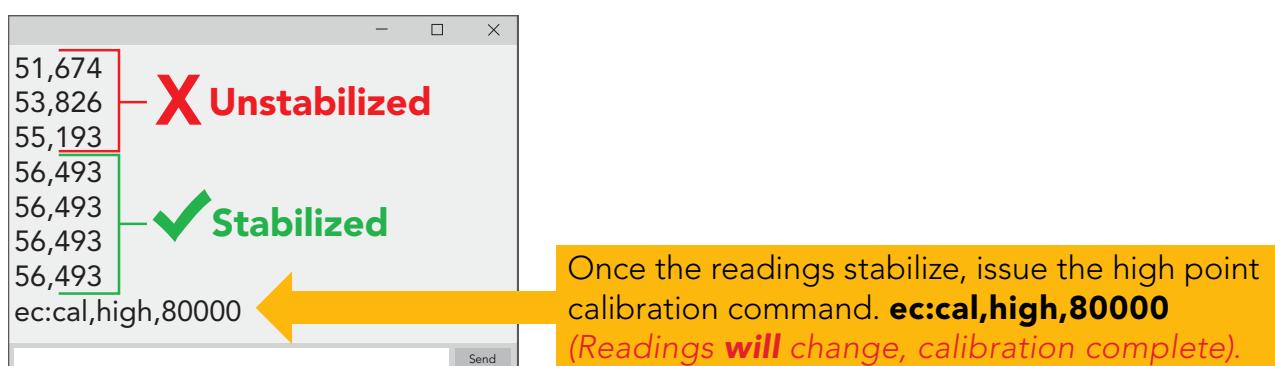


Make sure that the probe is dry before issuing this command, **ec:cal,dry**

Once the dry calibration has been completed, place the probe into a small cup of the low point 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 high point 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.



D 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**

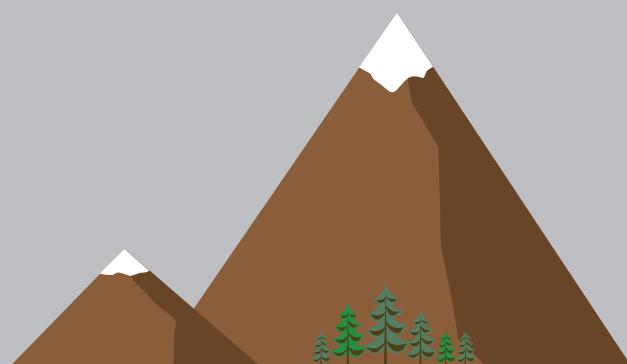
100 °C

Elevation in meters

305
229
152
76
0
-76
-152

Boiling point

98.9 °C
99.2 °C
99.5 °C
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!