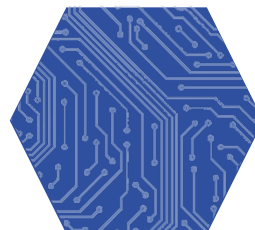
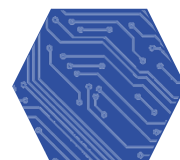




# Activity 2

Streaming Soil Moisture to the  
Internet



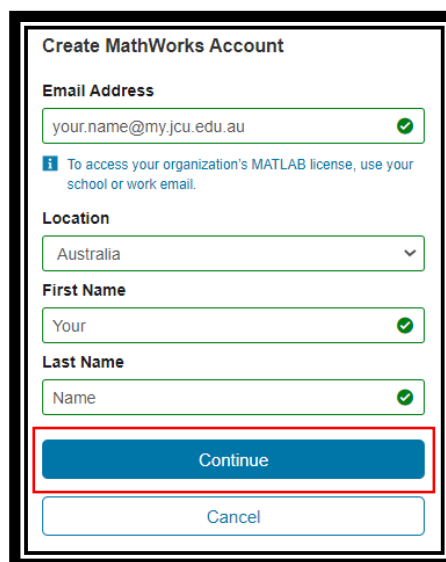
### Step 1: Create a ThingSpeak Account

ThingSpeak is an IoT application for gathering, visualising, and analysing data on the Internet. Data can be streamed to a **field** within a **channel**, and each channel can have up to 8 fields.

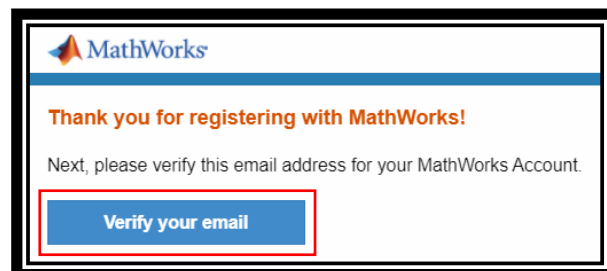
To use ThingSpeak, you will need a free account. Go to <https://thingspeak.com/login>. Click on the “No account? Create one!” link. (Note: if you’ve used ThingSpeak before you can use an existing login.)

A screenshot of the MathWorks login page. At the top is the MathWorks logo. Below it is the heading "Email". There is a text input field for an email address. Below the input field, the text "No account?" is followed by a red-outlined button labeled "Create one!". Below that, it says "By signing in you agree to our privacy policy." At the bottom right is a blue button labeled "Next".

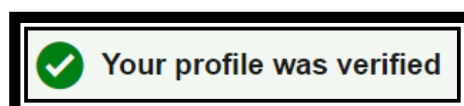
Next, fill out the required details. Make sure you use an email address that you can access today. Once you have completed your details, click “Continue”. Do not close this website.

A screenshot of the "Create MathWorks Account" form. It has several fields: "Email Address" with the value "your.name@my.jcu.edu.au" and a green checkmark; "Location" with a dropdown menu showing "Australia"; "First Name" with the value "Your" and a green checkmark; and "Last Name" with the value "Name" and a green checkmark. At the bottom, there is a red-outlined blue button labeled "Continue" and a light blue button labeled "Cancel".

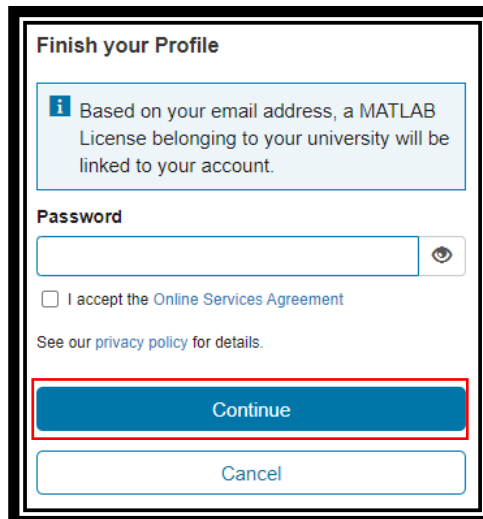
At this stage, you will receive an email from MathWorks asking you to verify your account. It will take a few minutes to come through but check your junk folders if you haven’t received it within 10 minutes. Once received, click “Verify your email”.

A screenshot of a MathWorks confirmation page. It features the MathWorks logo at the top. Below the logo, it says "Thank you for registering with MathWorks!". Underneath, it says "Next, please verify this email address for your MathWorks Account." At the bottom, there is a red-outlined blue button labeled "Verify your email".

You should then be redirected to a webpage that confirms the verification of your account:



Return to the account creation website. After verification has been completed, click “Continue”. You will then be asked to create a password. Once you have chosen a password, click “Continue”.



**Finish your Profile**

**i** Based on your email address, a MATLAB License belonging to your university will be linked to your account.

**Password**

☐ I accept the [Online Services Agreement](#)

See our [privacy policy](#) for details.

**Continue**

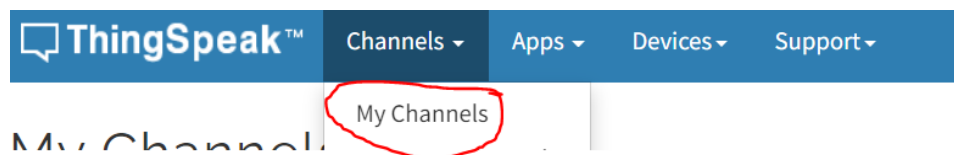
**Cancel**

At this stage, you will be returned to the login page <https://thingspeak.com/login>. Enter your newly-created account details to log in. You will then be redirected to the main page, <https://thingspeak.com/>.

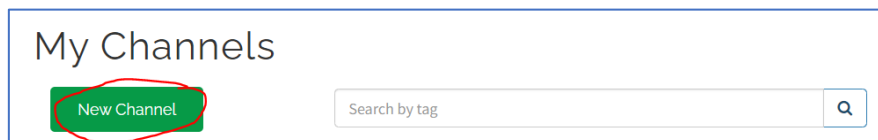
You are now ready to get started with ThingSpeak!

#### Step 2: Create your channel

Navigate to the “My Channels” screen using the top navigation bar:



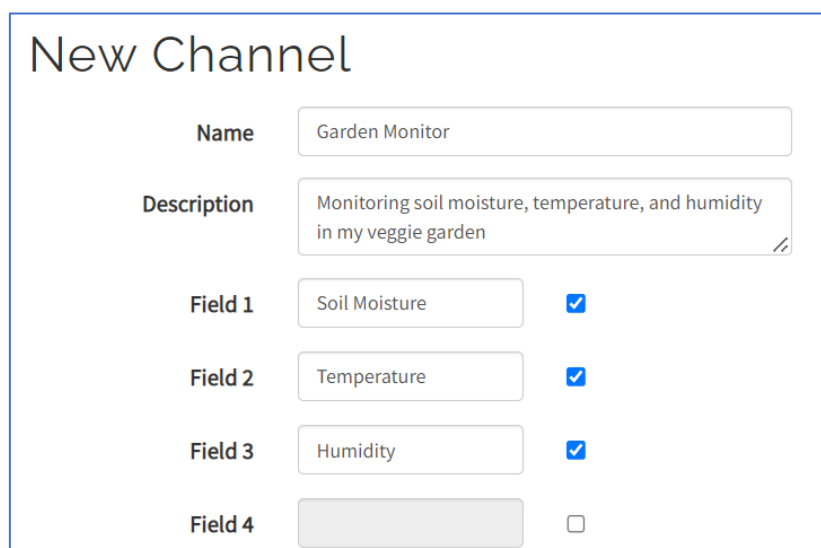
Click on “New Channel”



My Channels

**New Channel** Search by tag

Fill out the details for your new channel. If you want to implement soil moisture, temperature, and humidity, you’ll also need to create three fields.



**New Channel**

**Name** Garden Monitor

**Description** Monitoring soil moisture, temperature, and humidity in my veggie garden

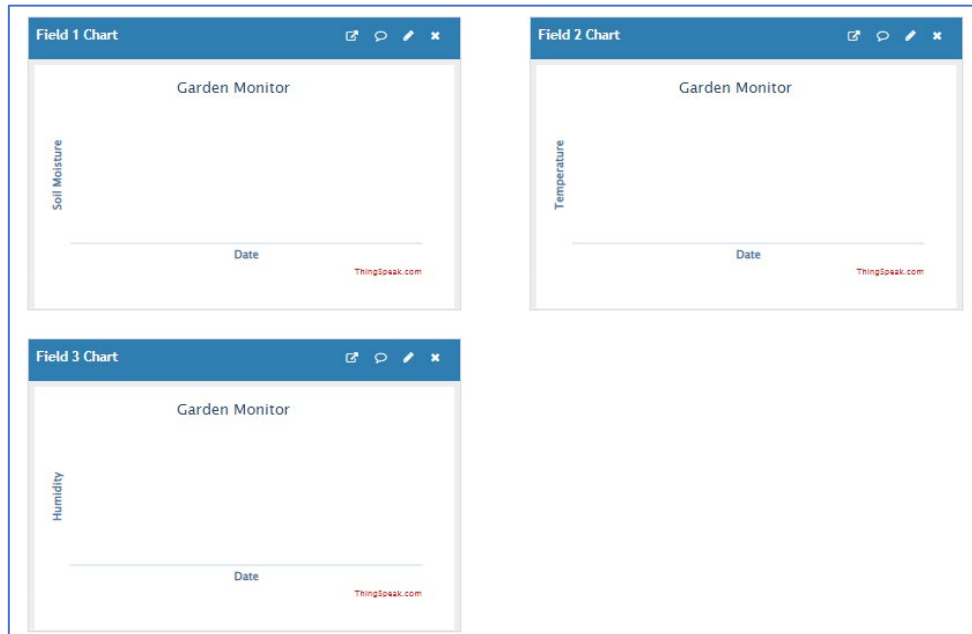
**Field 1** Soil Moisture ☒

**Field 2** Temperature ☒

**Field 3** Humidity ☒

**Field 4**  ☐

When you have filled out the above details, you can skip the rest - scroll to the bottom and click “Save Channel”. You should now have a blank channel that looks something like this:



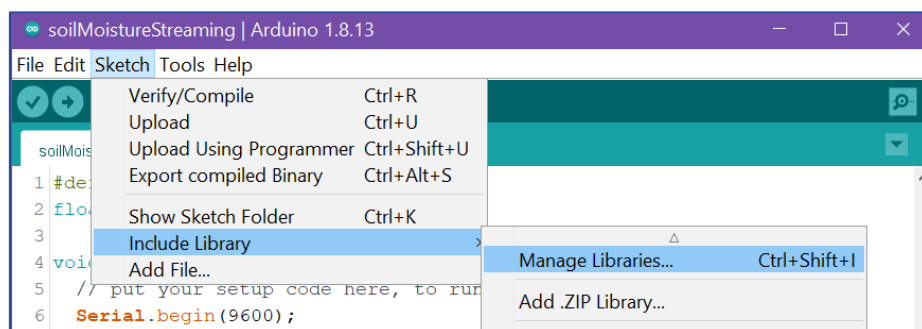
This is where your data will appear once you have it streaming!

### Step 3: Modifying your Arduino code to stream to the Internet

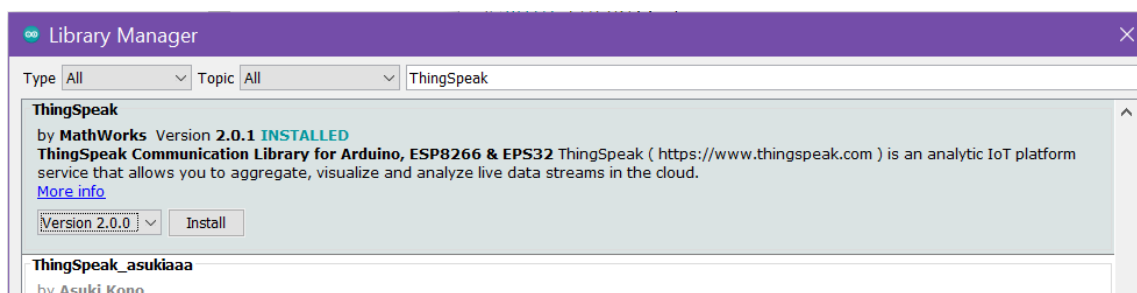
There are many useful libraries that can be downloaded for Arduino and Arduino-compatible boards that make programming easier. Two libraries that are helpful for this task are **ESP8266 WiFi** and **ThingSpeak**. The ESP8266 WiFi library provides a lot of useful functions (code commands) for creating a Wi-Fi connection, while ThingSpeak gives us functions that makes sending data to our ThingSpeak Channel easy.

The ESP8266 WiFi library is installed by default when you add the ESP8266 chip to Arduino (which you did in Activity 1). However, you will need to manually install the ThingSpeak library.

To do so, navigate to Sketch > Include Library > Manage Libraries (or alternatively Tools > Manage Libraries)



This will open a new window with a search bar at the top. Type in “ThingSpeak” and this will find the ThingSpeak library for you. Then, click “Install”. It will take a few moments to install, and then you can close the Library Manager.



Now that we have the libraries installed, we need to tell the D1 mini that we want to use them!

```
1 // Include the necessary libraries
2 #include <ESP8266WiFi.h>
3 #include "ThingSpeak.h"
```

Now it's time to connect the Wi-Fi! In the Ideas Lab, you will have access to the 'elequent' network. Add the following code to tell the D1 mini what network it will need to use:

```
6 // Set up WiFi information
7 #define WIFI_NETWORK_NAME "elequent"
8 #define WIFI_PASSWORD "elequent137"
9 WiFiClient client;
```

If you're using it at home, you can replace the network name and password with your home Wi-Fi's details!

Next, head to the setup loop to start the WiFi up:

```
21 void setup() {
22   // Start serial port
23   Serial.begin(9600);
24
25   // Start WiFi
26   WiFi.mode(WIFI_STA); ←
27 }
```

The starter code you were given includes some commented-out lines for managing the Wi-Fi connection. You can uncomment these now by removing the `//` from the start of each line. Hint: a shortcut for doing this is to highlight the relevant lines and press CTRL and `/` at the same time.

```
24 void loop() {
25   // WiFi connection - uncomment the following 10 lines when ready to connect!
26   // if (WiFi.status() != WL_CONNECTED) {
27   //   Serial.print("Attempting to connect to WIFI_NETWORK_NAME: ");
28   //   Serial.println(WIFI_NETWORK_NAME);
29   //   WiFi.begin(WIFI_NETWORK_NAME, WIFI_PASSWORD);
30   //   while(WiFi.status() != WL_CONNECTED) {
31   //     Serial.print("..");
32   //     delay(5000);
33   //   }
34   //   Serial.println("\nConnected.");
35   // }
```



```
24 void loop() {
25   // WiFi connection - uncomment the following 10 lines when ready to connect!
26   if (WiFi.status() != WL_CONNECTED) {
27     Serial.print("Attempting to connect to WIFI_NETWORK_NAME: ");
28     Serial.println(WIFI_NETWORK_NAME);
29     WiFi.begin(WIFI_NETWORK_NAME, WIFI_PASSWORD);
30     while(WiFi.status() != WL_CONNECTED) {
31       Serial.print("..");
32       delay(5000);
33     }
34     Serial.println("\nConnected.");
35   }
```

Next up, you will need to establish a connection to the ThingSpeak channel. Head up to your definition section, and start with this syntax:

```
11 // Set up ThingSpeak streaming
12 #define THINGSPEAK_CHANNEL_NUM 0123456
13 #define THINGSPEAK_WRITE_KEY "MYWRITEKEY000"
```

However, you will need to replace the values! Your channel has a unique number and write key. Both bits of information that you need can be found on the “API Key” screen:

Garden Monitor

Channel ID: 1785492  
Author: mwa0000020473525  
Access: Private

Monitoring soil moisture, temperature, and humidity in my veggie garden

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key: Q1ZX38P4J3TRJJ9M

Generate New Write API Key

Help

API keys enable you to write data to a channel. Keys are auto-generated for each channel.

API Keys Section

- Write API Key: Used to write data to a channel. It has been compromised.
- Read API Key: Used to read data from a channel. It has been compromised.

Add your channel ID and write key to your code in the format shown above.

Now head back to your ‘setup’ block and add the following code after your Wi-Fi code. This lets the Arduino know that you’re planning on using ThingSpeak.

```
28 // Start up ThingSpeak
29 ThingSpeak.begin(client);
```

#### Step 4: Write to ThingSpeak!

Now you’re ready to send data to ThingSpeak. Head back to the ‘loop’ loop and find where you are reading the soil moisture data. After your code for reading the soil data, add the following code:

```
48 // Write data to ThingSpeak
49 ThingSpeak.setField(1, soilValue);
50 int x = ThingSpeak.writeFields(THINGSPEAK_CHANNEL_NUM, THINGSPEAK_WRITE_KEY);
```

In the ThingSpeak.setField code, the first value is the position of the field that you are writing to. If you made soil moisture in ‘Field 1’, then this number should be 1. If you put it somewhere else, you can change this number. The second value, ‘soilValue’, is the variable that has the soil moisture data. Feel free to change this if you used a different variable name when reading the soil moisture.

At this point, it would be helpful to increase your delay. At this stage, try 10 seconds. Once you have tested your ThingSpeak functionality, you will likely want to increase this further – soil moisture doesn’t tend to change too fast!

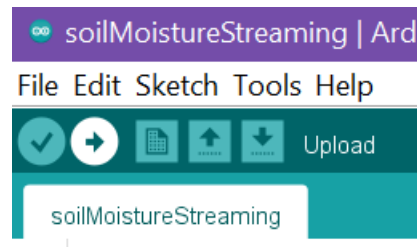
```
52 // Delay between updates
53 delay(10000)
```

As a little advanced functionality, you can the following block of code in between your write and delay code:

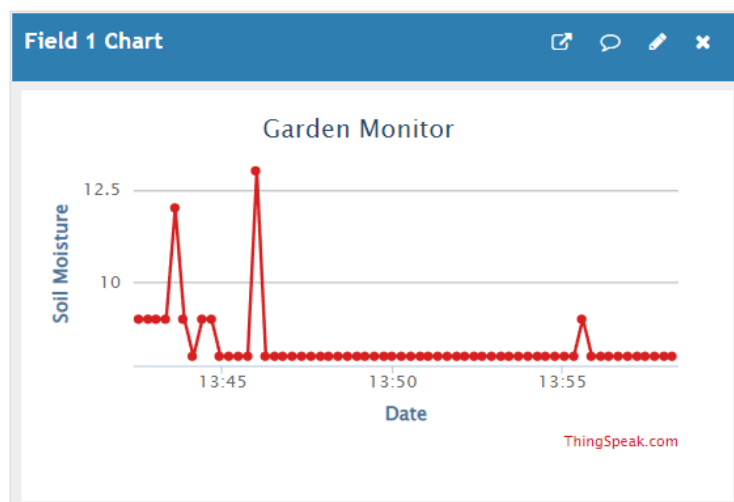
```
52 // Check for success
53 if (x == 200) {
54   Serial.println("Channel update successful.");
55 } else {
56   Serial.println("Problem updating channel. Error code: " + String(x));
57 }
```

This checks whether the write operation has been successful; a successful write returns the value '200', whereas any other number indicates an error of some kind. This can help you to debug your code if anything strange is happening.

Next, upload your code:



Once it has finished uploading, head over to your ThingSpeak channel and wait. After a few seconds, you should see data start to arrive on your channel. A new data point should appear roughly every 10 seconds (as this was the delay that we set).



Once you see data arriving on your channel (and it matches the data being shown in your Serial Monitor), you are finished with this activity! In Activity 3, you will learn how to add temperature and humidity sensing to your garden monitoring system.