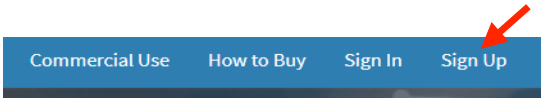
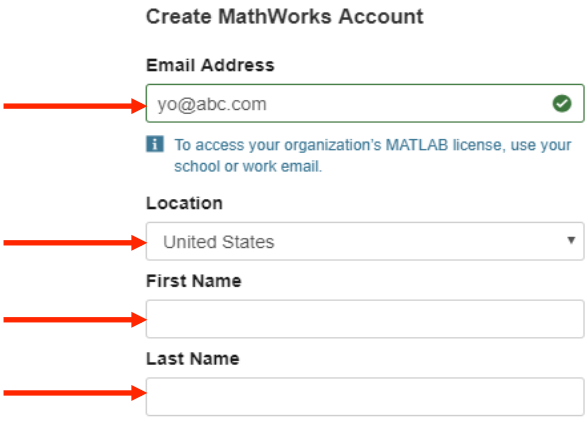
Detailed Instructions for Setting up ThingSpeak Channel for Data Logging with logIT.

I. Required Software - Download the Arduino code provided with this manuscript at the journal website. Also, the "Adafruit_FONA.h" library must be downloaded from the internet and installed in the proper directory on your computer. Details and further instructions are found at https://learn.adafruit.com/adafruit-fona-mini-gsm-gprs-cellular-phone-module/arduino-test.

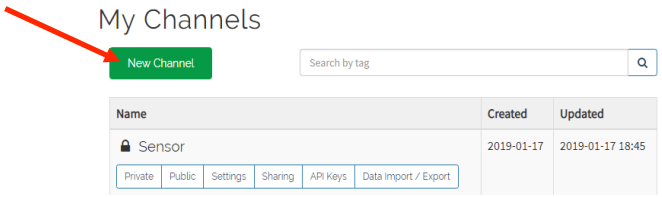
II. Make a free account on the thingspeak website (https://thingspeak.com/). Sign into the new account. If you experience problems with this step, try a different web browser.



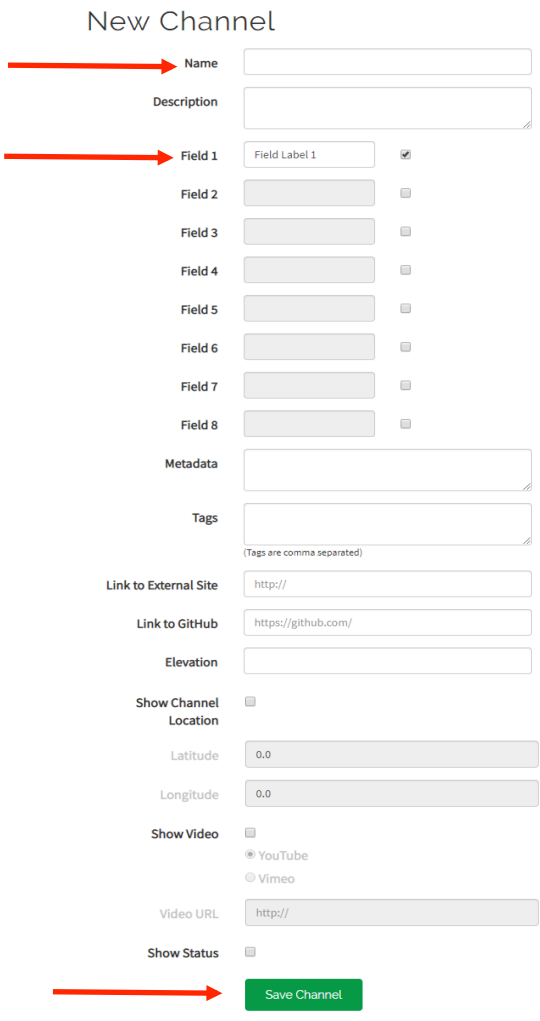


III. Sign in to your MathWorks Account. Click on "New Channel" and define all the necessary fields according to your requirements. (It is always good to define a descriptive name and field name. Examples for fields - Longitude, Latitude, Sensor 1 etc.). The user may define up to 8 channels. Linking or including videos are optional. Then, save the channel.

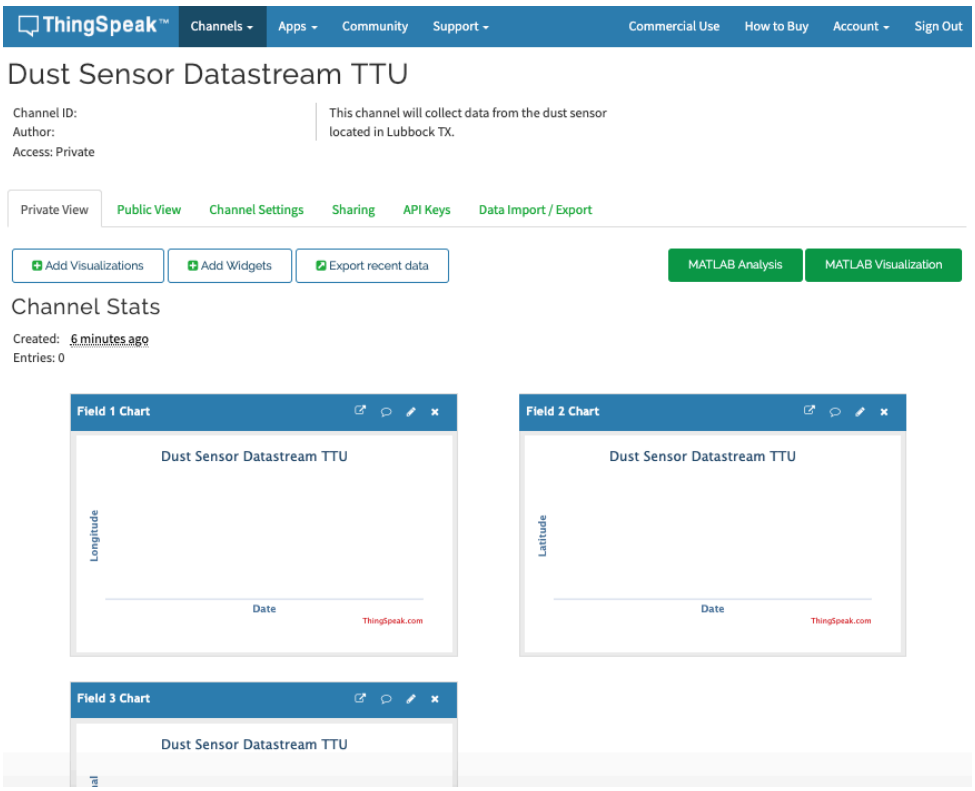
A. Creating a new channel.



B. Filling the fields

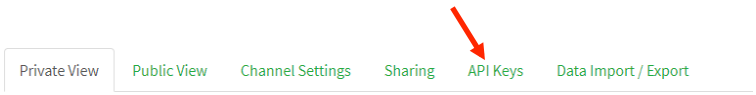


C. After completion. The results should look as indicated.

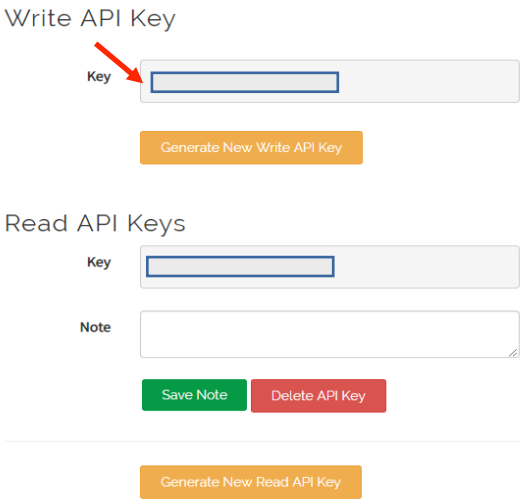


IV. After creating a new channel, check for your own unique API by clicking on "API Keys." For this application you need to record the write API key and reference it in your Arduino code. Note the API mentioned under the Write API. (It is important for reading the data on the Thingspeak channel).

A.



B.



V. Changes to Arduino Code for Customization.

A. One thing that will need to possibly be changed in the Arduino sketch file is the APN. The APN is used by your network to recognize the SIM, you will need to change the APN if not using TING network. Check with your network provider for the proper APN. For TING, the APN is 'WHOLESALE'

```
Serial.println(("AT+CSTT=\"WHOLESALE\""));          // Use the APN of service provider
fonaSS.println(("AT+CSTT=\"WHOLESALE\""));           // APN for TING is wholesale

Serial.println("AT+SAPBR=3,1,\"APN\",\"wholesale\""); // Make changes in APN
fonaSS.println("AT+SAPBR=3,1,\"APN\",\"wholesale\""); // Make changes in APN
```

B. Next, after taking note of your API at the Thingspeak website, specify the correct write API within the Arduino code. Locate the section of code which reads the following and then enter the correct API:

```
Serial.print("GET /update?key=Enter the API&field1=" + String( Parameter 1) + "&field2="+String(Parameter 2)) + "&field3="+String(Parameter 3);

fonaSS.print("GET /update?key=Enter the API&field1=" + String( Parameter 1) + "&field2="+String(Parameter 2)) + "&field3="+String(Parameter 3);
```

The values for Parameter 1, Parameter 2, and Parameter 3 are the variables in the Arduino sketch the user defines for measurements. These also must be changed to reflect your code. Here, Parameter 1, Parameter 2, and Parameter 3 are the names for the variables the user defines in the Arduino sketch. In our code, we name these 'measured', the second 'latitude', and the third 'longitude.' If your variable names differ, you will need to update the sketch.

C. Users may need to alter the void loop( ) section to coordinate with their specific sensor(s). This is needed to collect data from each sensor you employ and store the current value into a user-defined variable. Remember to define a variable to hold the sensor data, and reference that variable in step 5B above for logging into Thingspeak.

VI.

Connect the Arduino / FONA 808 module to the computer with a USB cable. Upload / Run the code and observe the changes in the field values. On occasion, it may require some time to acquire a GPS fix. If communications is a problem or if the data logging gets hung up, try hitting the reset button on the FONA 808 board. Usually, this corrects any issue and logging resumes within a few minutes.