User Manual

Project 68: Designing Window Sensors to Advance Bird- Friendly and Energy Saving Building Design Strategies on UBC Vancouver Campus

Huawen Li
Gengran Li
Ryotaro Hokao
Benjamin Powell

Revisions

Rev.	Date (DD/MM/YYYY)	Author(s)	Change Description
00	10/05/2023	Ryotaro Hokao Benjamin Powell Huawen Li Gengran Li	Initial release.

Table of Contents

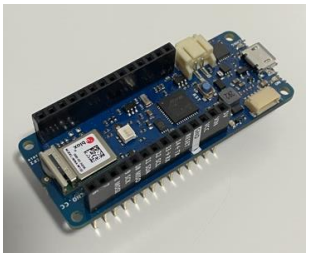
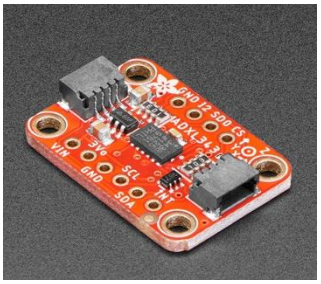
1.0 Introduction

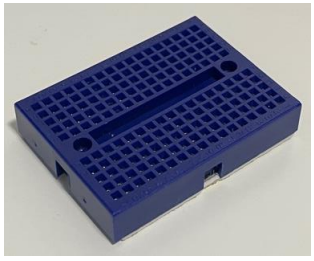
This user manual provides detailed instructions for setting up the bird collision detection system using an Arduino MKR WiFi 1010, an accelerometer, Dragino sensors, The Things Network (TTN), and ThingSpeak.

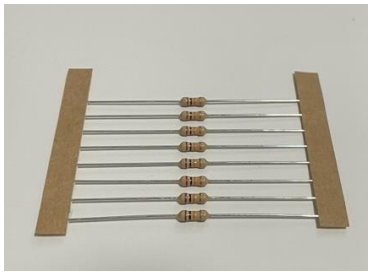
2.0 Components and Tools

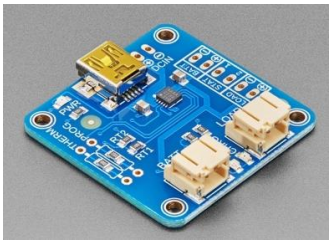
Table 1 below lists the components needed to assemble the collision detector system. See "PL-68 BOM.xlsx" for purchasing details. Table 2 lists additional tools needed to assemble or configure the collision detector system. Assembly and configuration instructions below also list which items are needed for each step.

Table 1: List of components needed to assemble the collision detector system for one window.

#	Item Name	Detailed Description	Quantity	Image
1	Arduino	Arduino MKR WiFi 1010	1	
2	Accelerometer Board	Adafruit Industries LLC, ADXL343 Evaluation Board	1	
3	Ribbon Cable	10 Position Cable Assembly Rectangular Socket to Cable 3.00' (914.40mm)	1	

4	Ribbon Cable Connector	10 Position Rectangular Socket Connector IDC Gold	1	
5	Pin Header	Connector Header Through Hole 10 position 0.100" (2.54mm)	1	
6	Duct Seal Putty	Gardner Bender Duct Seal	1	
7	Breadboard	BREADBOARD MINI (0165-40-06-15010) - BLUE	2	

8	Pushbutton	Tactile Switch SPST-NO Top Actuated Through Hole	1	
9	Wires	SILICONE HOOK UP WIRE KIT, 22AWG, 5 COLOUR (RD,BK,YW,GN,BL)	1	
10	10kΩ Resistor	RESISTORS 1/4W 10K 5% 10PCS	1	
11	Dragino Sensor	LHT65 Temperature & Humidity Sensor	2	
12	O-ring	Water- and Steam-Resistant EPDM O-Ring, 1/8 Fractional Width, Dash Number 242	1	
13	Cable Gland	Cable Gland Polyamide (PA), Nylon M16x1.5 Gray	1	

14	Locknut	M16x1.5 Locknut Nut 0.866" (22.00mm) Polyamide	1	
15	USB Cable	USB A to Micro B Cable	1	
16	Power Adapter	USB Wall Charger	1	
17	Battery	Lithium-Ion Polymer Battery - 3.7v 2500mAh	1	
18	Battery Charger	USB LiIon/LiPoly charger - v1.2	1	
19	Case	3D Printed Enclosures	1	

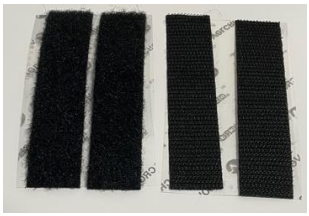
				
20	Screw	Screw M3x30MM Self-Tapping 10PCS	1	
21	Velcro Tape	VELCRO Brand Outdoor Heavy Duty Strips 4 x 1 Inch Pk of 10	1	

Table 2: List of required tools to assemble and configure the collision detector system.

#	Description
1	Computer with USB connectivity (Windows or Mac)
2	Soldering Station
3	Solder
4	Wire Stripper
5	Screwdriver (require M3 bit)
6	Electrical Tape
7	Vise

3.0 Setup Instructions for Monitoring

This section describes how to set up the already-assembled collision detector system on a window for bird-window collisions and heat flow rate monitoring. If you are purchasing components and assembling a new set of collision detector system, see section 4.0.
The assembled collision detector system should look like the following:

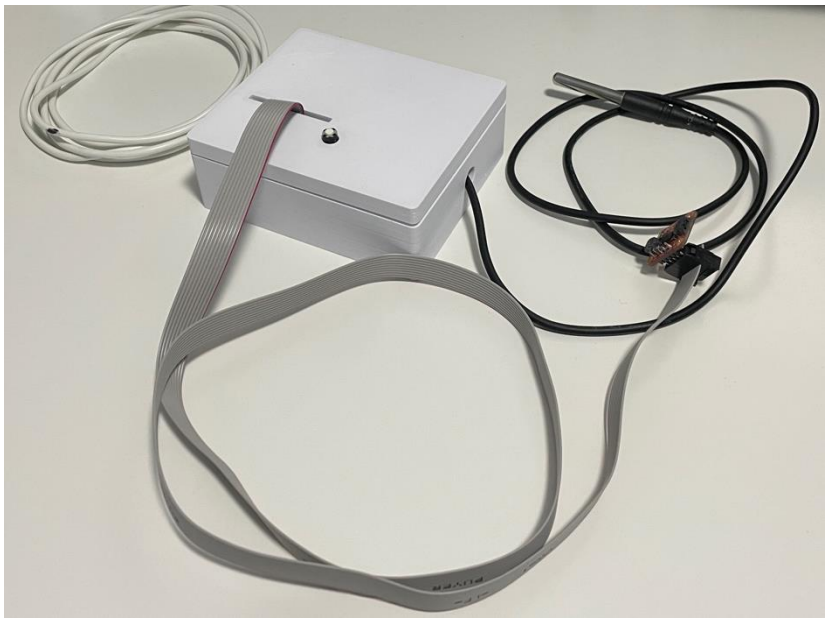
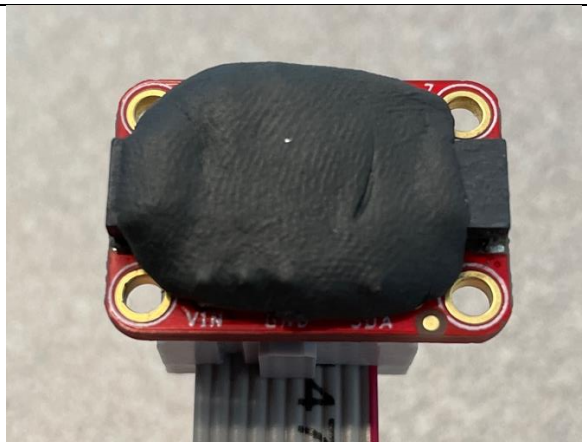
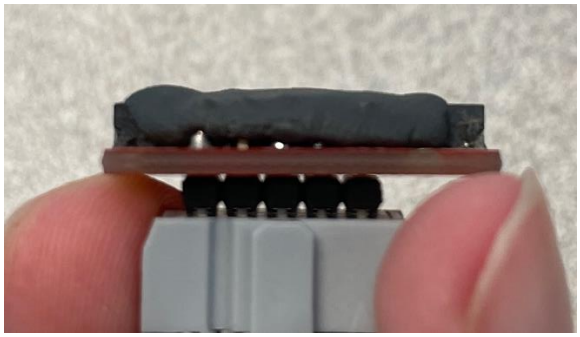


Figure 1: Assembled inside case.

Figure 2: Assembled outside case.

This step also assumes that a) the Arduino is already flashed with the firmware outlined in step 4.6, b) Dragino sensors are setup on The Things Network as outlined in step 4.7 and 4.11, and c) ThingSpeak dashboard is set up as outlined in step 4.9.

3.1 Attach the accelerometer to the window

Required Item: • Assembled Collision Detector System • Duct Seal Putty (Duct Putty) **Required Tool:** • None	
1. Use gloves and apply duct putty to the accelerometer board. Make sure the system is not powered on. **Safety Information:** Duct putty should be safe to touch (unless one has sensitive skin condition), but use of gloves is recommended. See Safety Data Sheet for more information.	 *Figure 3: Apply duct putty to cover the surface of the accelerometer board.*  *Figure 4: Putty should be evenly applied with no large bumps.*

2. Attach the accelerometer board to the window surface. Make sure the accelerometer board is as parallel to the window panel as possible.

   About the window size:
   The collision detector system is tested on a test window of 180cm by 80cm. If the system is deployed on a window larger than this, it is recommended to deploy multiple system per window.

   About the sensor location:
   Somewhere around the center of the window is recommended for the best result.

Figure 5: Attach the side with duct putty applied to the window. Push the accelerometer board against the window to securely attach it.

Figure 6: Make sure the accelerometer board is as parallel to the window panel as possible.

3.2 Attach the temperature sensors to the window

This step can be skipped if the heat flow rate measurement is not needed (i.e., if the user only needs to know the bird-window collisions).

Required Item: • Assembled Collision Detector System • Velcro Tape **Required Tool:** • Electrical Tape	
1. Attach the temperature probe coming out from the inside case to the window surface with electrical tape. Make sure the metal tip of the probe is touching the window panel.	 *Figure 7: Attach the temperature probe to the window panel with electrical tape.*

	 *Figure 8: Make sure the metal tip of the temperature probe is touching the window surface.*
2. Access the outside of the window. Attach the Velcro tape to the window sill where the outside case will be secured. Also, attach the Velcro tape to the bottom of the outside case.	

3. Attach the temperature probe coming out from the outside case to the window surface with electrical tape. Make sure the metal tip of the probe is touching the window panel.	 *Figure 9: Attach the temperature probe to the window panel with electrical tape.*

3.3 Power up the system

The inside case can be powered with a wall adapter, or with a battery. The wall power is recommended for more reliable and continuous operation.

Required Item: • Assembled Inside Case from section 4.4 • Power Adaptor • Battery **Required Tool:** • None	

Powered with wall adapter:

1. Connect the power adapter to the USB cable coming out of the inside case (if you did not attach the USB cable during assembly, see section 4.4).

Figure 10: Connect the USB cable to the power adapter.

Powered with wall adapter:

2. Plug in the power adapter to the wall outlet. Use extension cord as needed.

   Make sure the LED on the Arduino and the accelerometer board are on as shown in Figure 11 and Figure 12.

   Double check that 1) all the sensors are attached to the window correctly, 2) outside case is attached correctly, 3) inside case is in a secure location.

   Secure the inside case using Velcro tape, electrical tape, or other adhesives as needed.

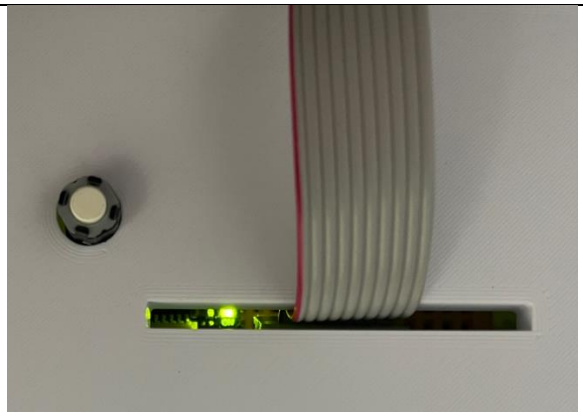


Figure 11: Check the LED on the Arduino is on through the slit on the lid.

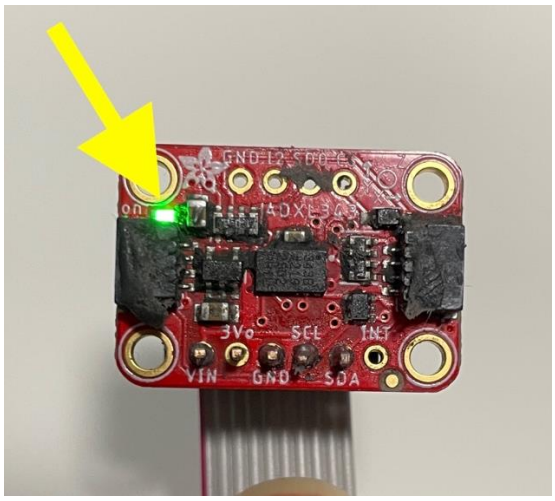


Figure 12: Check the LED on the accelerometer board.

Powered with battery:

1. Open the inside case.

   Remove the Dragino Sensor temporarily.

   Unplug the USB cable from the Arduino. Then, move the breadboard as shown in Figure 13 to make space to connect the battery.

Figure 13: Remove the Dragino Sensor, unplug the ribbon cable, unplug the USB cable, and move the breadboard temporarily.

2. Plug in the battery to the connector on Arduino as shown in Figure 14.

   Do not plug in the connector all the way in, which can make it difficult to unplug later.

   Make sure the LED on the Arduino and the accelerometer board are on as shown in Figure 12 and Figure 14.

   Leave the lid off when powered with battery.

   Double check that 1) all the sensors are attached correctly, 2) outside case is attached correctly, 3) inside case is in a secure location.

   Secure the inside case using Velcro tape, electrical tape, or other adhesives as needed.

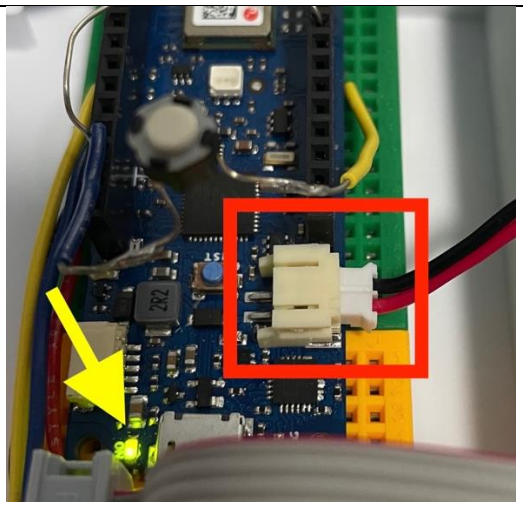
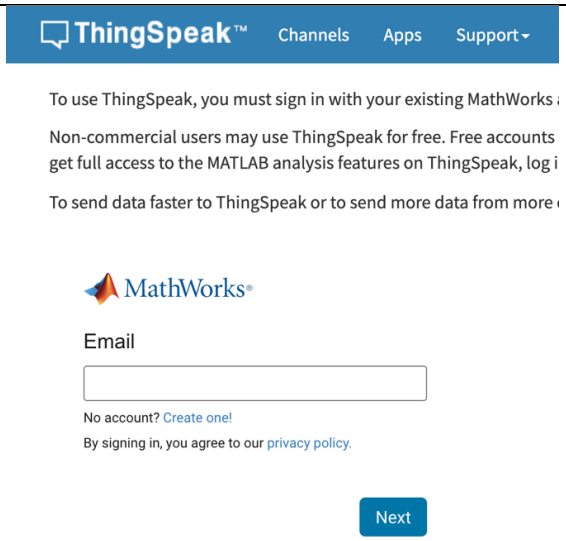
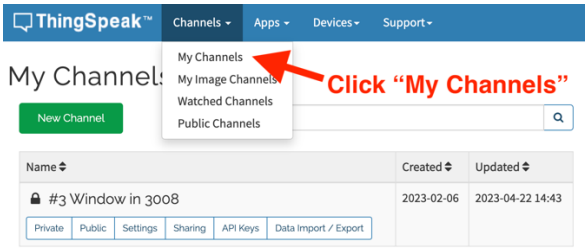


Figure 14: Plug in the battery to the connector on the Arduino. Do not plug the connector all the way in. Also, check the LED on the Arduino.

Figure 15: Leave the lid off when powered with battery.

3.4 Monitor from ThingSpeak

As mentioned earlier, this section assumes that ThingSpeak dashboard is set up as outlined in step 4.9.

Required Item: • None **Required Tool:** • Computer	
1. Log in to ThingSpeak with the provided email and password. Click “Channels” > “My Channels”, then select the appropriate dashboard to monitor the collision data and heat flow rate measurement (if applicable). If you have not configured the ThingSpeak dashboard, see section 4.9.	 *Figure 16: ThingSpeak log in page.*  *Figure 17: Click "Channels", then "My Channels" to see a list of dashboards.*

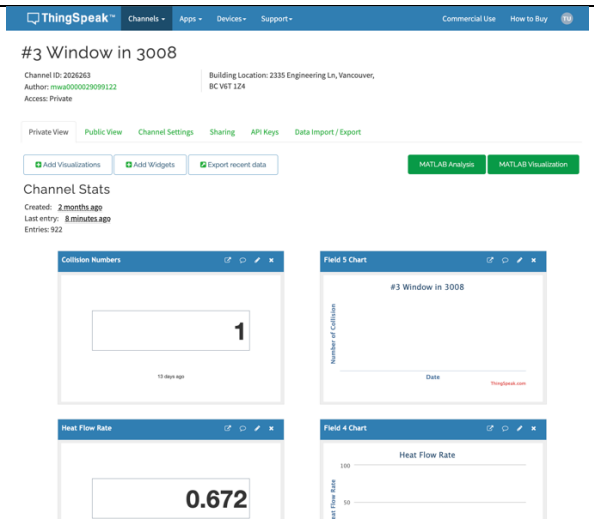


Figure 18: Configured ThingSpeak dashboard.

2. Click “Channel Settings” as shown in Figure 19.

   Modify “Name” and “Description” fields as needed (e.g., where the device is deployed). See Figure 20.

   Scroll down and click “Save Channel” to save changes as shown in Figure 21.

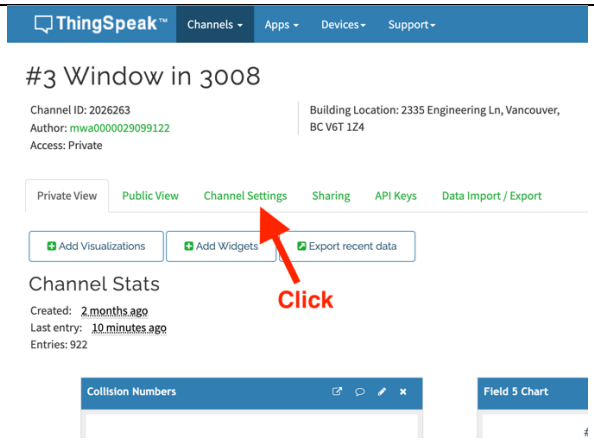


Figure 19: Click "Channel Settings" to change the dashboard name.

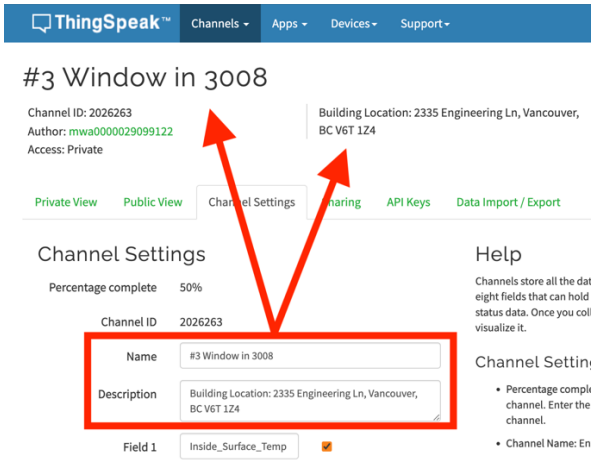


Figure 20: Change "Name" and "Description" fields as needed. They are displayed at the top of the dashboard.

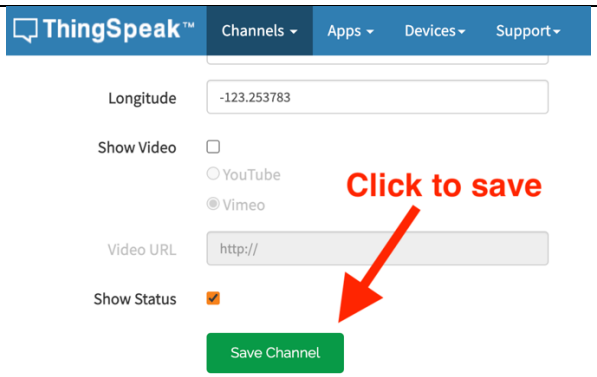


Figure 21: Scroll down and click "Save Channel" to save changes.

3. To change the U value and the window dimension needed for heat flow rate measurement, click “Apps” at the top, then “MATLAB Analysis” as shown in Figure 22.

   Open “Heat Flow Rate Calculation” script.

   Change the U value and the window height and width as shown in Figure 23.

   Scroll down and make sure to click “Save and Run”.

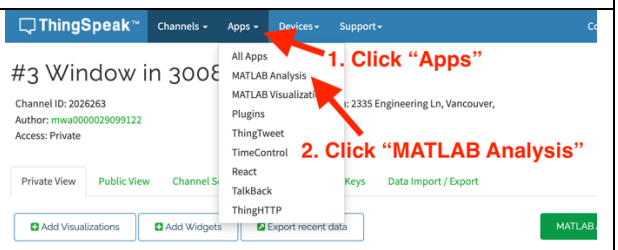


Figure 22: Click "Apps" at the top, and click "MATLAB Analysis" to modify the U value and window dimension.

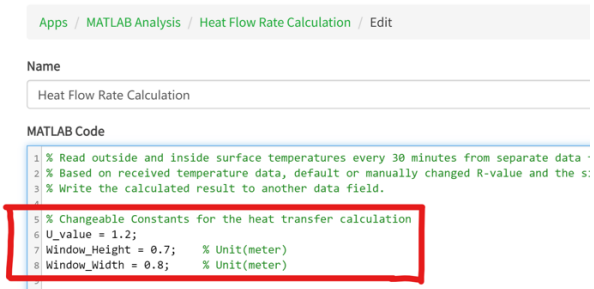


Figure 23: Change the U value and the window dimension to calculate the heat flow rate.

4.0 Assembly Instruction

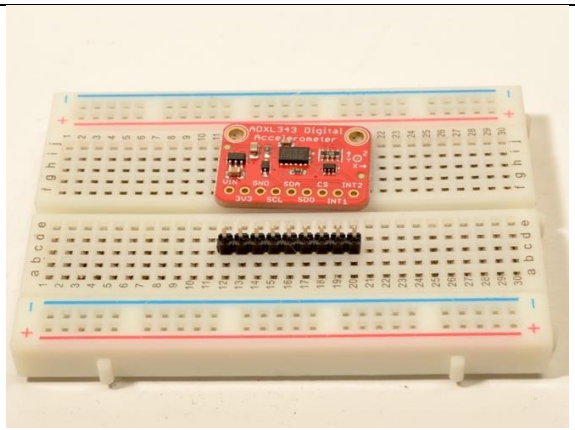
This section describes how to assemble the components into a collision detector system. If you already have an assembled system, see section 3.0 for how to deploy it on a window for monitoring.

****Note to the user:**

Assembly steps involve soldering, connecting parts on a breadboard, uploading firmware code to Arduino, etc. Many steps are documented in a way that non-engineer users can assemble the system by following the detailed instructions. However, for safety reasons, it is strongly recommended to have someone with soldering experience perform the soldering outlined in section 4.1 below.

4.1 Solder accelerometer board to 10-pin header

Manufacturer's tutorial page also has instructions on soldering, and additional information on the accelerometer board: https://learn.adafruit.com/adxl343-breakout-learning-guide/assembly

Required Item: • Accelerometer Board • 5-Pin Header **Required Tool:** • Soldering station • Solder	
1. Insert pin header into a breadboard. Cut the pin header strip to 5 pins as needed.	 *Figure 24: Use breadboard to help secure the accelerometer board while soldering.*

2. Put the accelerometer board over the pins. Solder the following pins:
   - VIN
   - GND
   - SCL
   - SDA

   Cut the excess pin headers as shown in Figure 27.

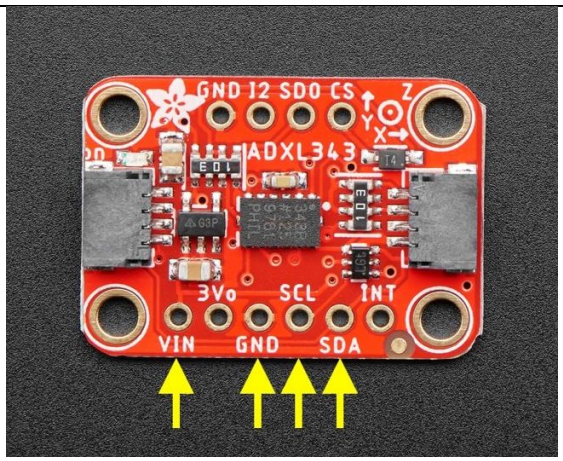


Figure 25: Solder the four pins needed for I2C communication.

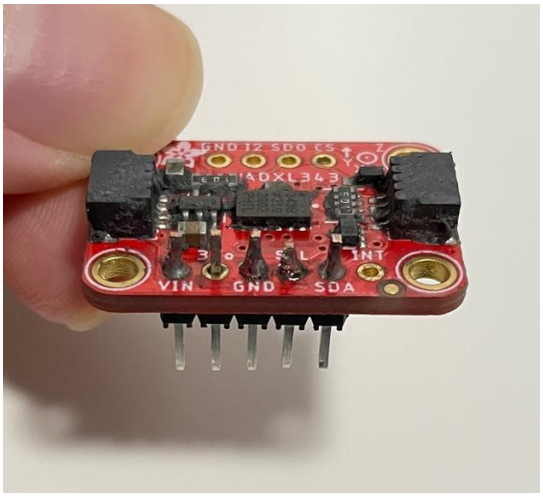


Figure 26: Soldered accelerometer board. Unused pins like "3Vo" pin can be left unsoldered.

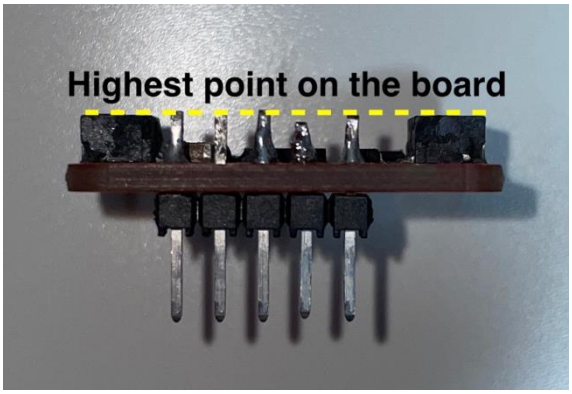


Figure 27: Cut the pins that are longer than the highest point of the accelerometer board.

4.2 Connect pushbutton and pin header to Arduino

Required Item:

- Arduino
- Breadboard x2
- 5-Pin Header
- Pushbutton
- 10kΩ Resistor
- Wires

Required Tool:

- Wire Stripper

1. Connect two breadboards.

   Insert Arduino and pin header to breadboard as shown. Cut the pin header strip to 5 pins as needed.

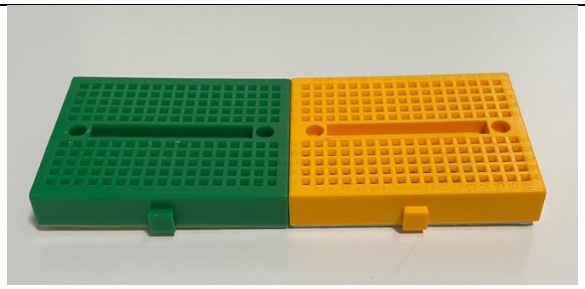


Figure 28: Two breadboards connected together.

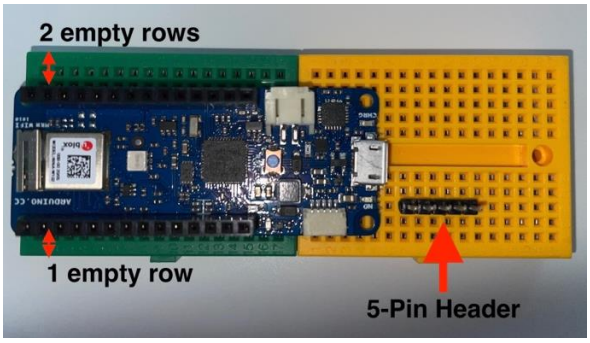


Figure 29: Position of Arduino and 5-pin header.

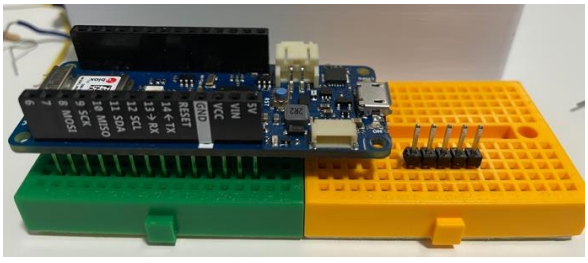


Figure 30: Placement of Arduino pins for reference. Pins need to be pushed down so that the pins are inserted in the breadboard.

2. Cut four wires with the following colours and lengths:
   - Red: 5cm
   - Black: 6cm
   - Blue: 7.5cm
   - Yellow: 7.5cm

   Strip 1cm from both ends of each wire.

   Colours of the wires are arbitrary, but it is recommended to keep them consistent for easy troubleshooting.

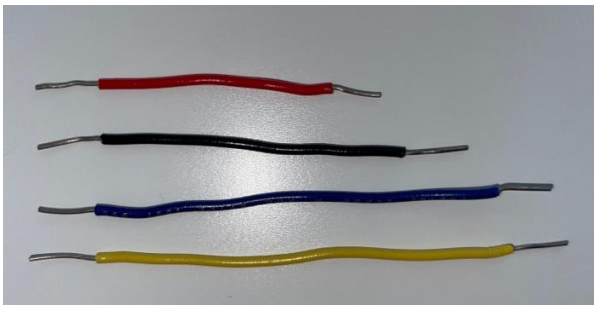


Figure 31: four wires used to connect the Arduino and the pin header.

3. Connect the following pins of Arduino to the pin header using wires you cut in the previous step as shown.
   - VCC
   - GND
   - SCL
   - SDA

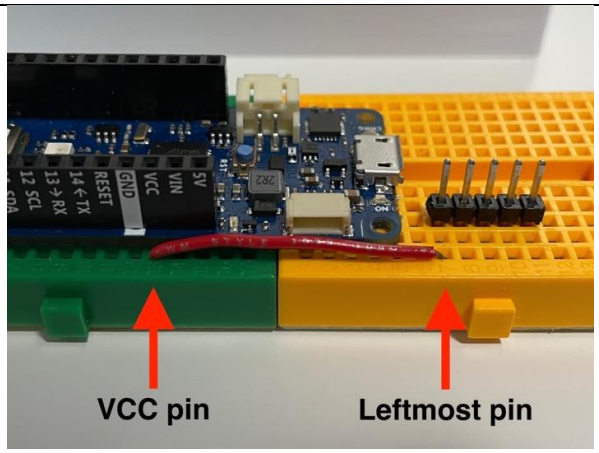


Figure 32: Connect VCC pin to the leftmost pin using red wire.

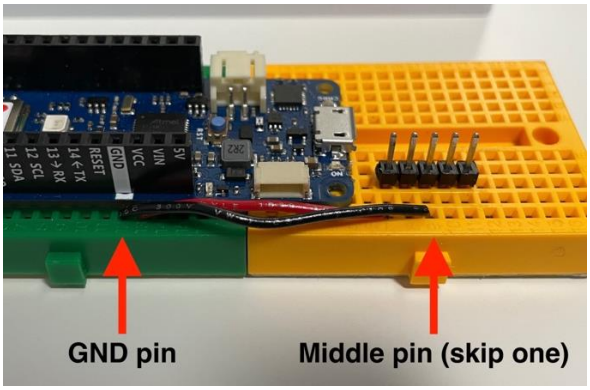


Figure 33: Connect GND pin to the middle pin using black wire.

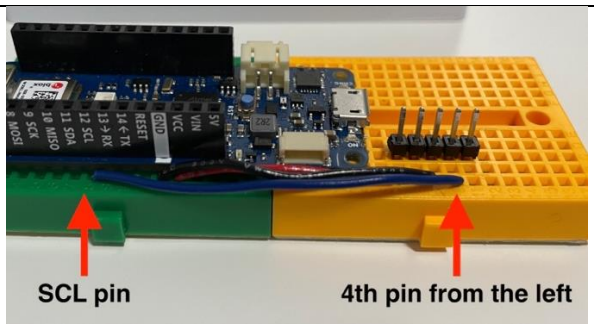


Figure 34: Connect SCL pin to the 4th pin from the left using blue wire.

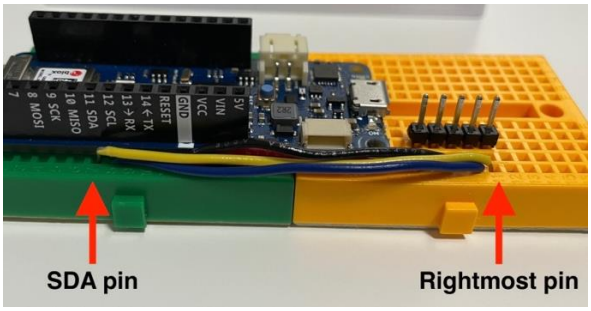


Figure 35: Connect SDA pin to the rightmost pin using yellow wire.

4. Cut two wires with the following colours and lengths:
   - Blue: 3.5cm
   - Yellow: 4.5cm

   Strip 1cm from both ends of each wire.

   Solder legs of the pushbutton to two wires as shown.

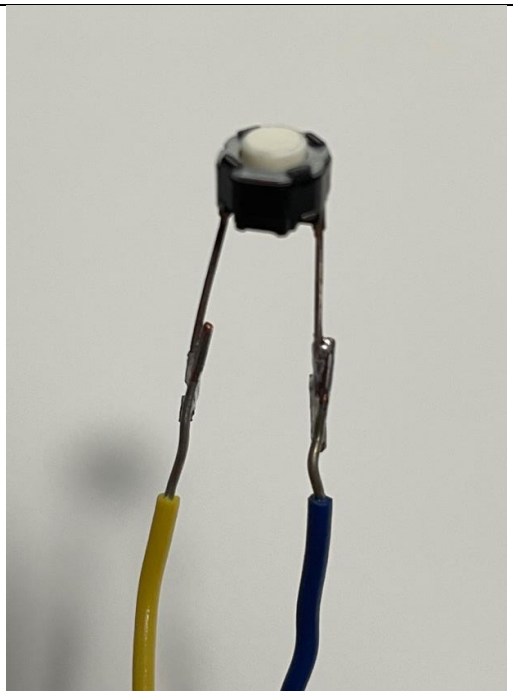


Figure 36: Soldered pushbutton.

5. Connect one leg of the resistor to VCC pin on the Arduino as shown in Figure 37.

   Connect the other leg of the resistor to pin 0 located on the other side of the Arduino as shown in Figure 38.

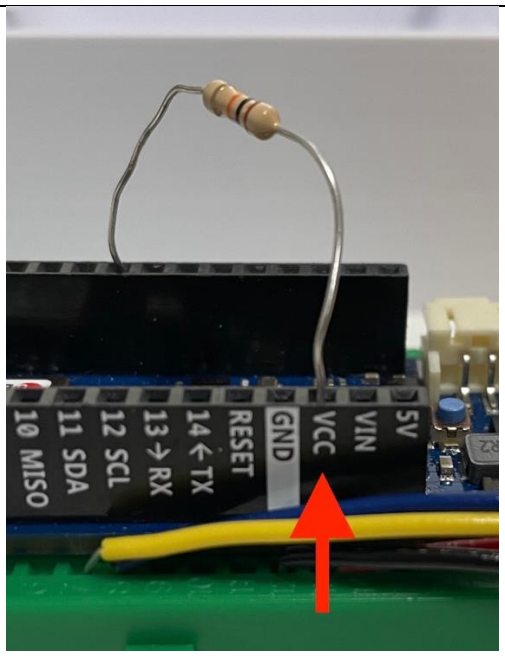


Figure 37: Connect one leg of the resistor to VCC pin.

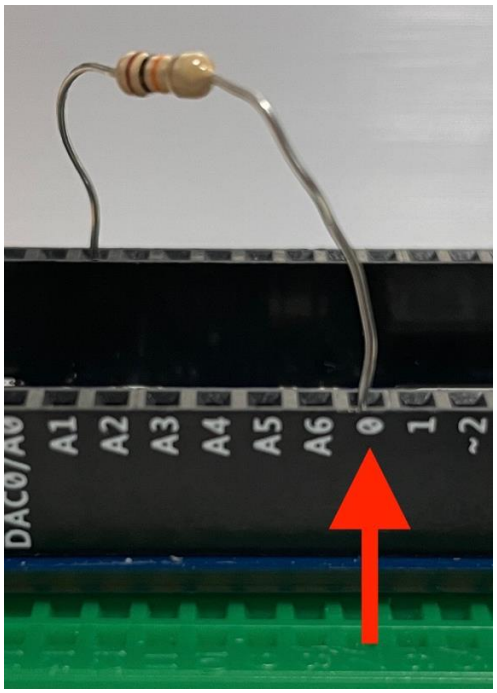


Figure 38: Connect the other leg of the resistor to pin 0.

6. Connect blue end of the soldered pushbutton to GND pin on the Arduino as shown in Figure 39.

   Connect the yellow end of the soldered pushbutton to pin 0 through breadboard as shown in Figure 40.

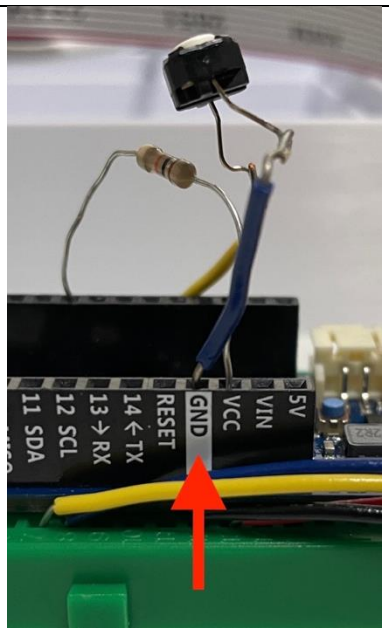


Figure 39: Connect the leg of the pushbutton with blue wire to GND pin.

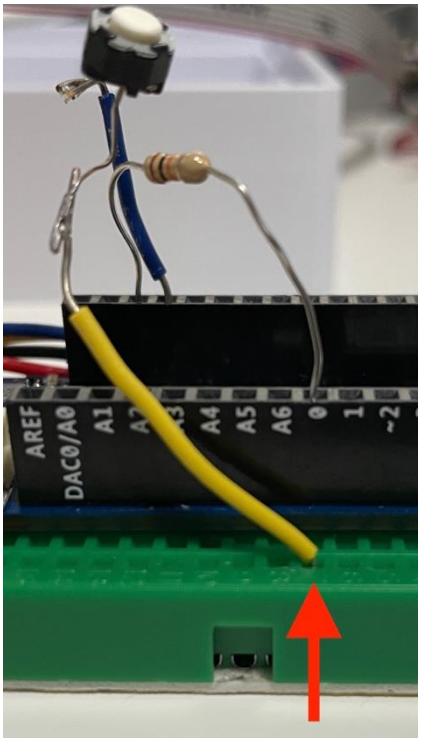
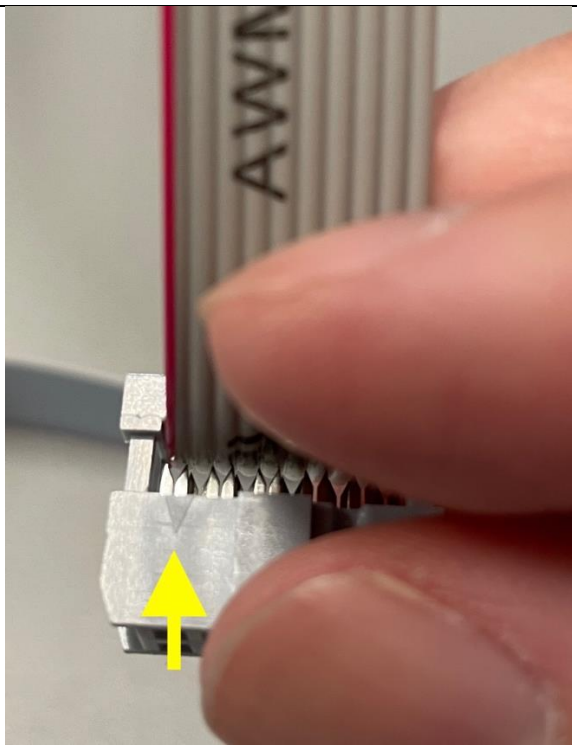
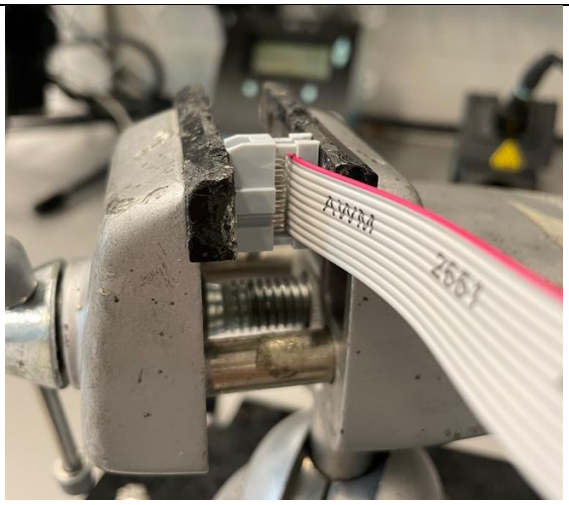


Figure 40: Connect the leg of the pushbutton with yellow wire to pin 0 through breadboard.

4.3 Put ribbon cable through lid

Required Item: • Inside Case (Lid) • Ribbon Cable • Ribbon Cable Connector **Required Tool:** • Vise	
1. Pass the ribbon cable through the slit of the lid as shown. Pay attention to the orientation.	 *Figure 41: Pass the ribbon cable through the slit on the lid.*

2. Pass through the end of the ribbon cable with no connector through the ribbon cable connector as shown. Align the side of the cable marked red with the triangle on the connector as shown. Check the orientation as shown in .	 *Figure 42: Align the triangle on the connector and the side of the ribbon cable marked red.*
3. Clamp the ribbon cable connector using the vise.	 *Figure 43: The connector can be clamped using a vise.*

4.4 Put Arduino assembly and Dragino sensor in the inside case

Required Item:

- Arduino Assembly from section 4.2.
- Lid Assembly from section 4.3.
- USB Cable
- Dragino Sensor
- Inside Case

Required Tool:

- Electrical Tape

1. Pass through the temperature probe from the inside of the case as shown in Figure 44.

   Connect the temperature probe to the Dragino sensor.

   Then, place the Dragino sensor inside the case. Align it as shown in Figure 46.

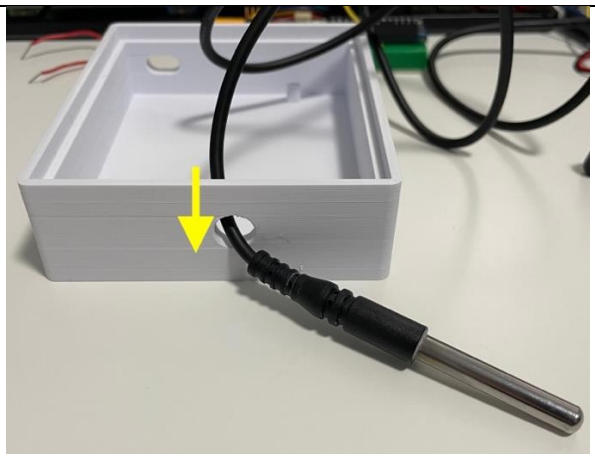
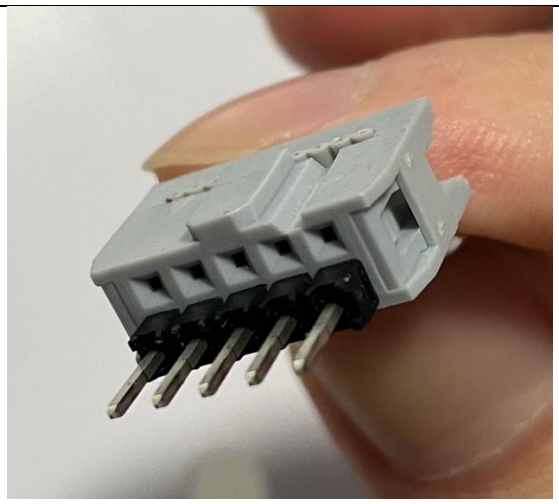
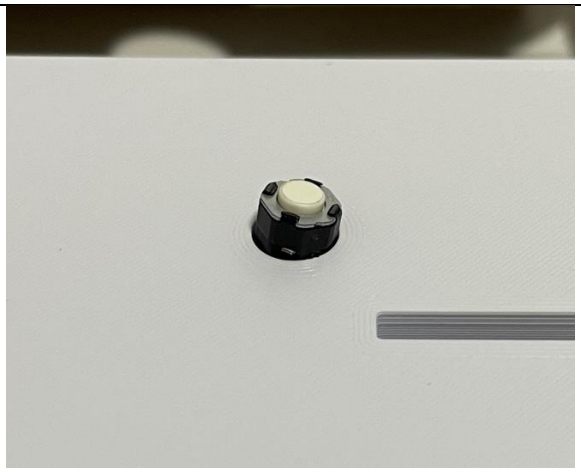
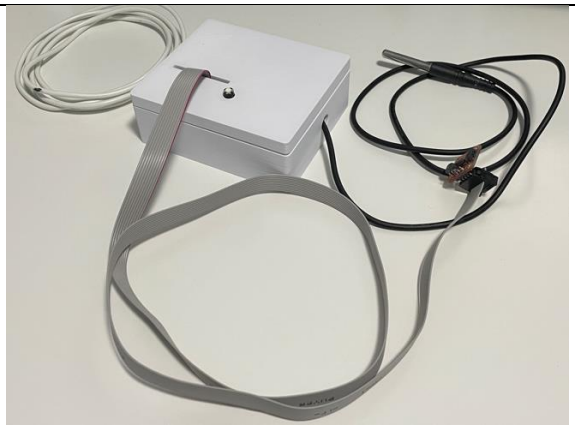


Figure 44: Pass the temperature probe through the hole on the side of the case.

Figure 45: Dragino sensor placed inside the case.

	 *Figure 46: Align the Dragino sensor with the "hook".*
2. Pass the micro USB cable through the other hole on the case from the outside of the case.	 *Figure 47: Pass through the micro USB cable from the other hole on the side of the case.*

3. Place the Arduino assembly in the case. Pay attention to the orientation as shown. Connect the micro USB cable to the Arduino.	 *Figure 48: Arduino assembly placed inside the case.*
4. Connect the ribbon cable connector to the pin header on breadboard as shown.	 *Figure 49: Connect the ribbon cable connector to the pin header on breadboard.*

	 *Figure 50: Ribbon cable connector is connected to the pin header like this.*
5. Align the pushbutton such that it pokes out of the hole in the lid. Secure the pushbutton with electrical tape if needed.	 *Figure 51: Align the pushbutton such that it pokes out of the hole in the lid.*
6. Double check all the connections. Close the lid.	 *Figure 52: Assembled inside case.*

4.5 Put Dragino sensor in the outside case

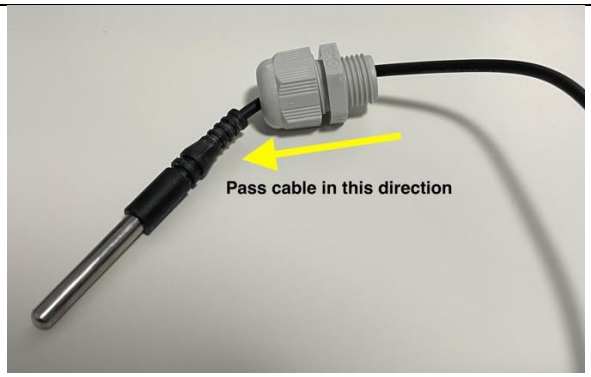
<table>
<tr><td>Required Item:
Dragino Sensor
Outside Case
O-ring
Cable Gland
Locknut
Screw
Required Tool:
Screwdriver with M3 bit</td><td></td></tr>
<tr><td>1. Pass the temperature probe through the cable gland as shown.</td><td> 

Figure 53: Pass the temperature probe in the direction indicated above.</td></tr>
<tr><td>2. Pass the temperature probe and cable gland through the hole on the case.

Secure the cable gland from the inside of the case using the locknut as shown in Figure 55.</td><td> 
Figure 54: Insert cable gland in the hole on the side of the case.</td></tr>
</table>

	 *Figure 55: Put the locknut from the other side of the hole and tighten to secure the cable gland.*
3. Connect the temperature probe to the Dragino sensor. Place the Dragino sensor in the case.	 *Figure 56: Connect the end of the temperature probe cable to the Dragino sensor.*

Figure 57: Place the Dragino sensor inside the case.

4. Tighten the cable gland by turning the outside piece shown in Figure 58.

   Keep turning if you can see the other side as shown in Figure 59.

   Stop when you cannot see the other side anymore, and if the cable is clamped as shown in Figure 60.

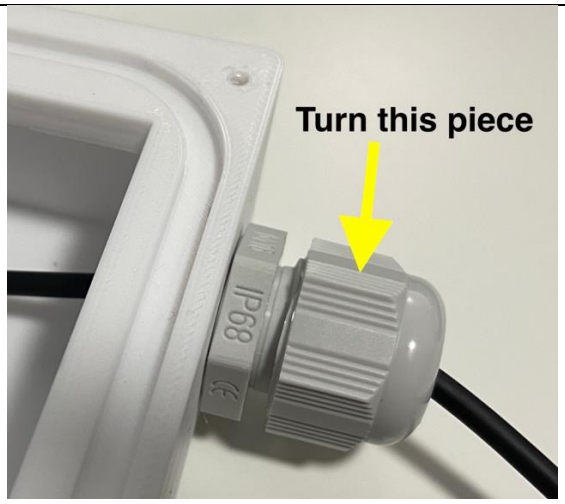


Figure 58: Tighten the cable gland by turning this piece.

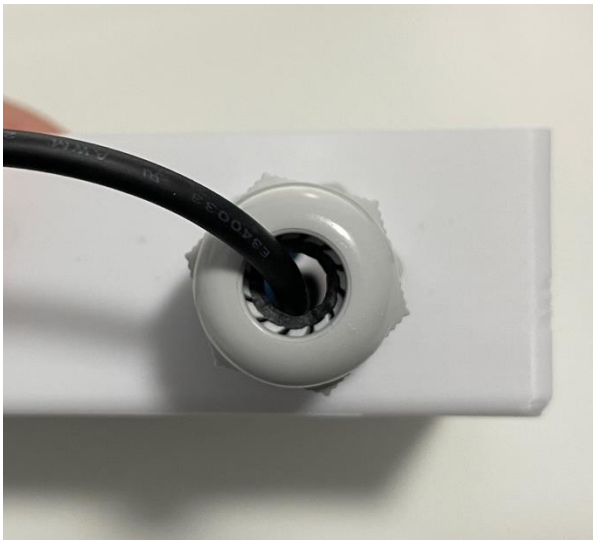
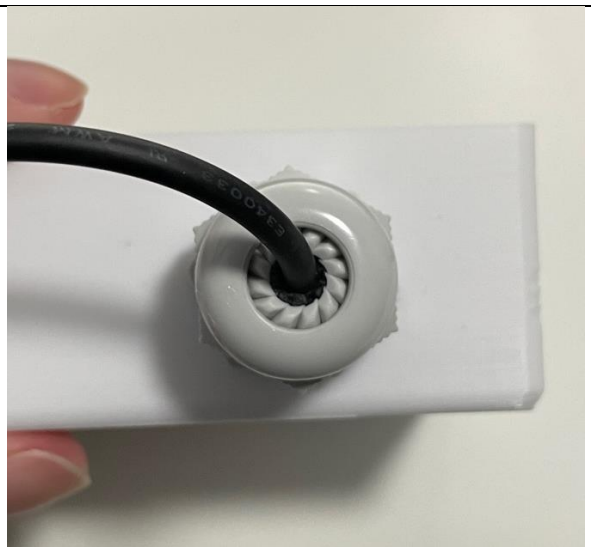
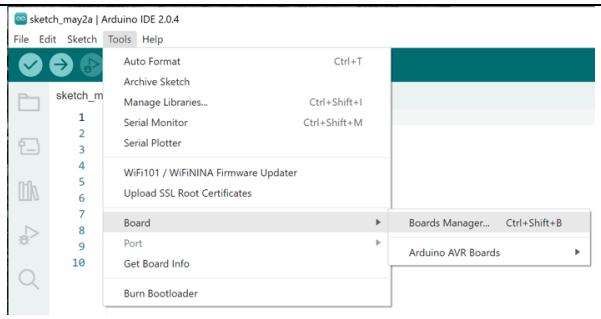


Figure 59: Cable gland before tightening.

	 *Figure 60: Cable gland after tightening.*
5. Inspect the O-ring for any damage (e.g., cut, tear). Discard the O-ring if damaged. Gently place O-ring in the groove. Make sure the O-ring does not get scratched in the process.	 *Figure 61: O-ring in the groove.*
6. Place the lid over the case. Secure the case with 4 screws.	 *Figure 62: Assembled outside case.*

4.6 Upload firmware to Arduino

Required Item: • Assembled Inside Case from section 4.4. **Required Tool:** • Computer with USB port	
1. On the computer, install Arduino IDE by following the instructions on: https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing. (On a work computer managed by the organization, you might need to contact IT for support.)	
2. Open Arduino IDE. Navigate to: "Tools" > "Board" > "Boards Manager" Search for "Arduino SAMD Boards" and click install as shown in Figure 64.	 *Figure 63: Navigate to "Board Manager".*

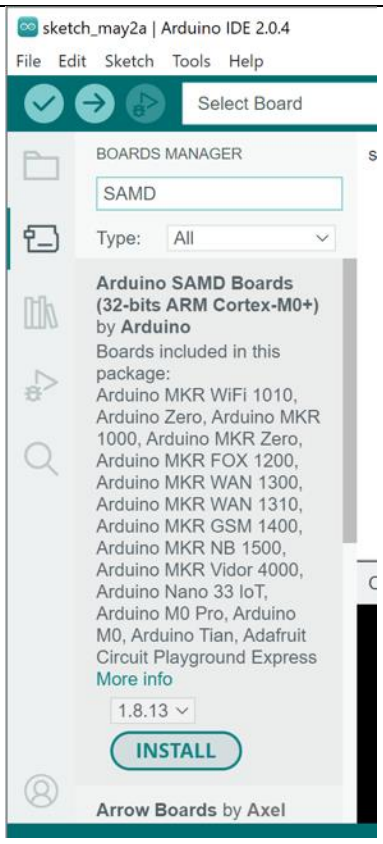


Figure 64: Search "SAMD", and install "Arduino SAMD Boards".

3. Go to [GitHub repository](#).

   See the README files for details on the contents of the repository.

   Click “Code”, then “Download ZIP” as shown in Figure 65.

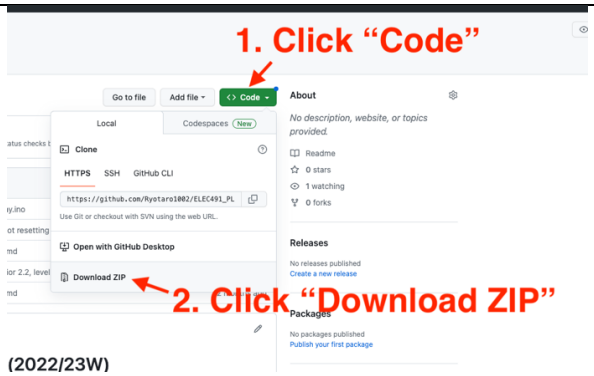


Figure 65: Steps to download the necessary files.

4. Unzip the zip file. Copy the contents to the folder generated by Arduino IDE:

   On Windows:
   "C:\Users\username\Documents\Arduino"

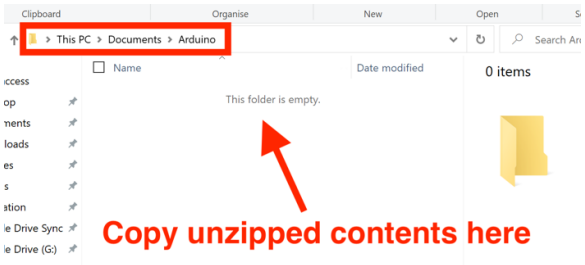


Figure 66: Copy unzipped files from GitHub to the Arduino folder.

5. Open one of the two INO files in "Integrated Sketch" folder.

   "Once_Per_Day.ino": Updates collision data once a day at midnight.

   "Every_Five_Minutes.ino": Updates collision data every 5 minutes. This version does not support battery-powered operation.

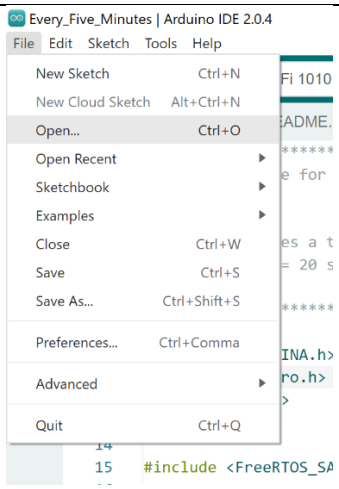


Figure 67: Click "File" then "Open" to open a file on Arduino IDE.

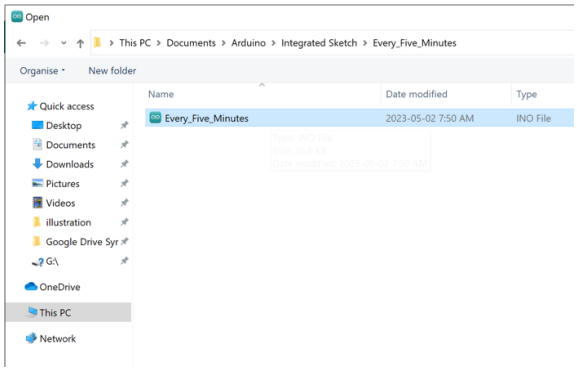


Figure 68: Open one of the INO file inside "Integrated Sketch" folder.

6. Change the “tempChannelNumber”, “myWriteAPIKey”, “writeFieldNumber”, and “FailFieldNumber” to values from ThingSpeak channel/dashboard. See 4.10.

```
61    //  Thingspeak libraries.
62    #include "ThingSpeak.h" // always include thingspeak hea
63    WiFiClient  client;
64
65    //Read/Send data from thingspeak
66    unsigned long tempChannelNumber = 2026263;
67    const char * myWriteAPIKey = "WN6UZKY58ABMJW89";
68    unsigned int wirteFieldNumber = 6;
69    unsigned int FailFieldNumber = 7;
70    /*-------------------------------------------------------
71    //*******************************************************
72    //static float32_t output[BLOCK_SIZE];
```

Figure 69: Update the Channel ID, Write API Key, and Field Numbers to communicate to the correct ThingSpeak dashboard.

7. If UBC visitor WiFi is not available, change “ssid[]”, and “pass[]” variables to change the WiFi network used by the Arduino.

```
30    int senddata_counter = 0;
31    // Include WiFiNINA libraries.
32
33    char ssid[] = "ubcvisitor";    // your network SSID (name)
34    char pass[] = "";    // your network password (use for WPA, or use as key for WEP)
35
36    int status = WL_IDLE_STATUS;  // the Wifi radio's status
37
```

Figure 70: Change SSID and the password to change the WiFi network used.

8. Connect the Arduino to the computer using the USB cable coming out of the case.

   Make sure the green LED on the accelerometer board is on as shown in Figure 71.

   If not, try changing the row on the ribbon cable connector as shown in Figure 72.

   If the LED still does not turn on, try switching the orientation of the accelerometer board as shown in Figure 73 and Figure 74.

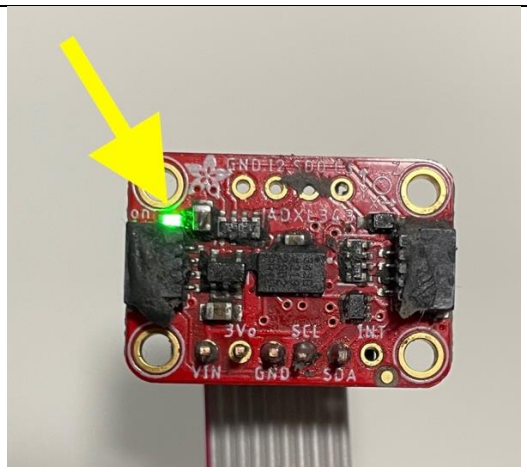


Figure 71: Make sure the LED on the accelerometer board turns on.

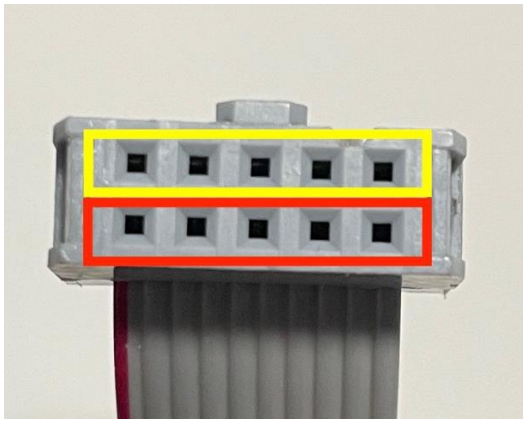


Figure 72: If the LED does not turn on, try switching the row on the connector.

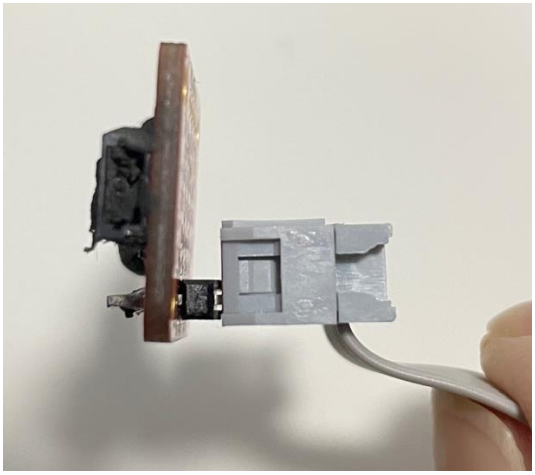


Figure 73: Intended orientation of the accelerometer board.

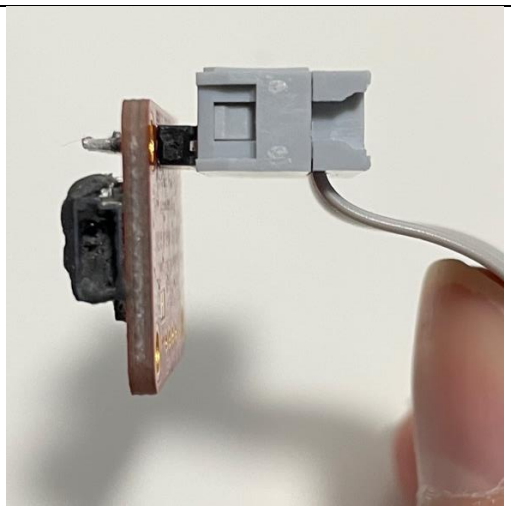


Figure 74: If the wiring is done differently, the LED may turn on with this orientation, which is okay.

9. Select the Arduino as shown in Figure 75. Make sure the device name appears as bold as shown in Figure 76.

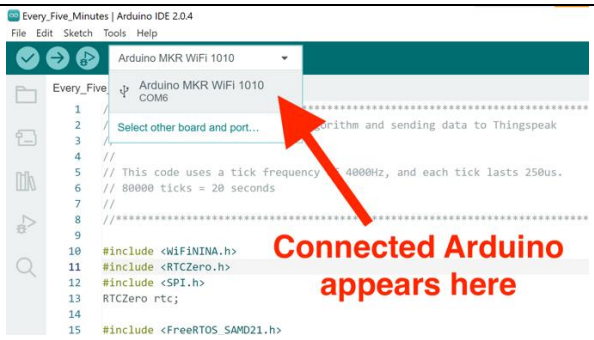


Figure 75: Arduino appears here once it is connected to the computer. Click to select.

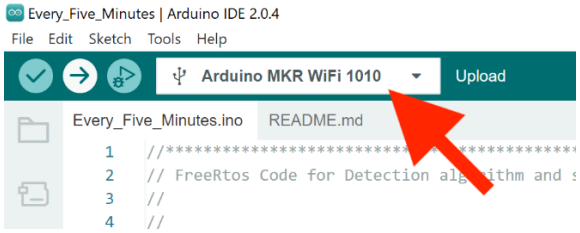


Figure 76: Once the Arduino is selected, the device name appears as bold.

10. Click the “Upload” button as shown in Figure 77 to upload the firmware code to the Arduino.

The “Output” window at the bottom of the window displays the status.

Once uploaded, the “Output” window displays something like shown in Figure 78.

See https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-uploading-a-sketch for more detail.

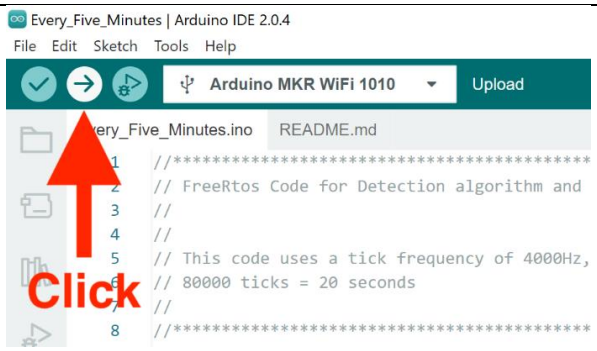


Figure 77: Click the "Upload" button to upload the code to the Arduino.

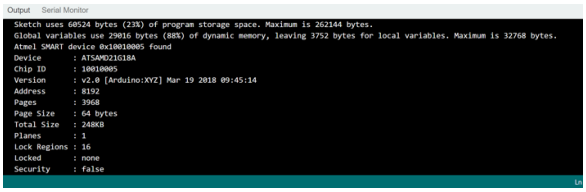


Figure 78: “Output” window displays something like this once compiled and uploaded the code.

Figure 79: It will say "Done uploading." once the code is successfully uploaded to the Arduino.

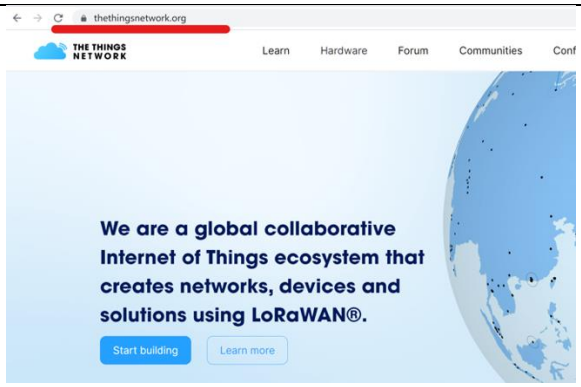
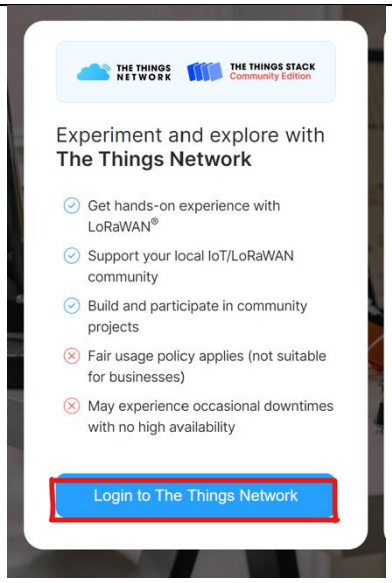
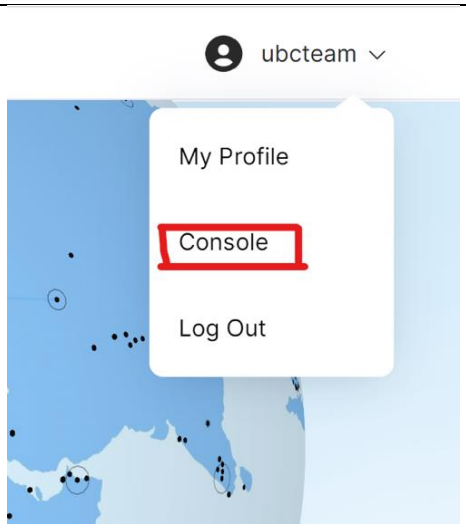
4.7 Set up Dragino sensors with The Things Network

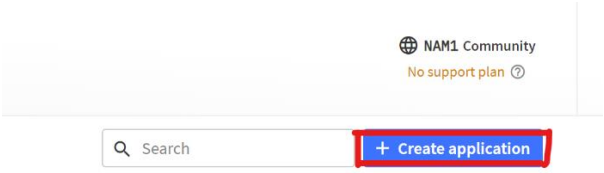
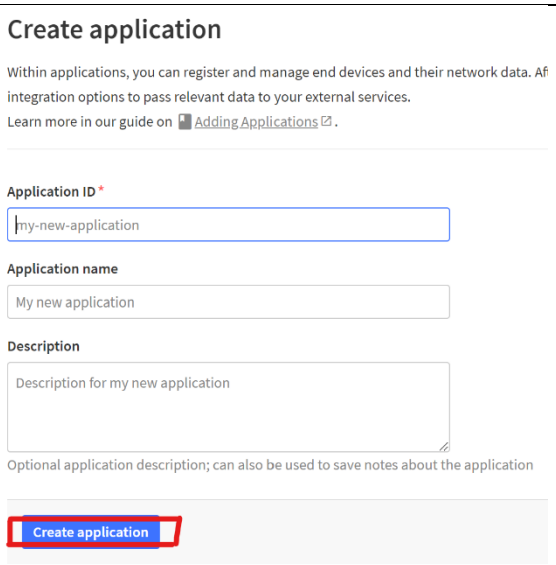
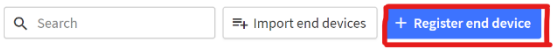
Required Item:

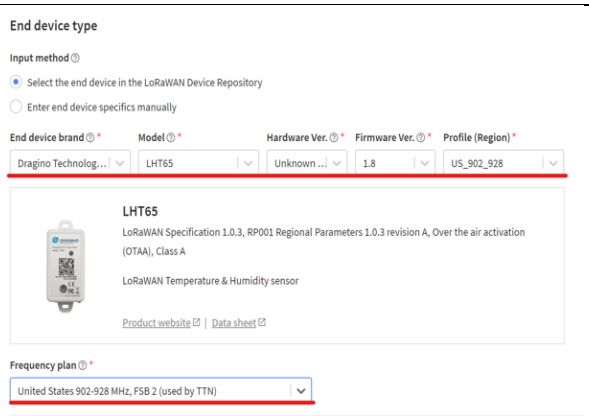
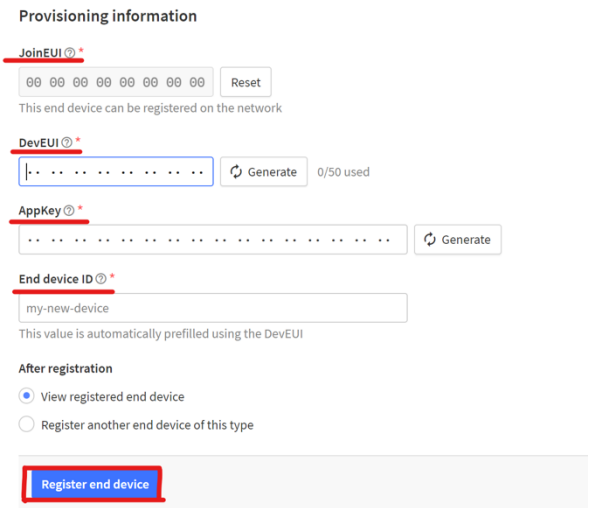
- Dragino Sensor x2

Required Tool:

- Computer

1. On the computer, navigate to The Things Network (TTN) website (https://www.thethingsnetwork.org/)	 *Figure 80: Go to The Things Network on a web browser.*
2. Click "Login" on the TTN website and enter your credentials.	 *Figure 81: Log in to the TTN account.*
3. After logging in, click on "Console" in the top-right corner of the page.	 *Figure 82: Click "Console" to access the TTN Console.*

4. On the TTN Console, go to "Applications" and click "Create application".	 *Figure 83: Click "Create application".*
5. Configure the application. Create an "Application ID". Click the "Create application" to create the application.	
6. Click "Register End Device" to register Dragino sensors.	 *Figure 84: Click "Register end device" to register a Dragino sensor.*

7. Fill in the information about the type of Dragino sensor to be added as shown. **End device brand:** Dragino Technology Co., Limited **Model:** LHT65 **Hardware Ver.:** Unknown ver. **Firmware Ver.:** 1.8 **Profile (Region):** US_902_928 **Frequency plan:** … FSB2 (Used by TTN)	 *Figure 85: Fill in the device information.*
8. Configure the Dragino sensor: Fill in the "JoinEUI" (aka App EUI), "DevEUI", and "AppKey". They are labeled on the Dragino Sensor, or are on the box the Dragino Sensor came in. Leave other settings as default. Click "Register end device" to complete the device registration. Repeat step 6-8 in the same application to register another Dragino Sensor.	 *Figure 86: Fill in the "JoinEUI", "DevEUI", and "AppKey" provided by the manufacturer.*
9. Once configured, go through step 4.11 to change the update interval to 30 minutes from the default interval.	

10. Connect the Dragino sensor to TTN Gateway:

Push the blue button on the Dragino Sensor (labeled ACT) till the green light blink 5 times, to power on the sensor.

Make sure the sensor is in range of a TTN Gateway.

The sensor will automatically connect to the Gateway and start sending data to TTN.

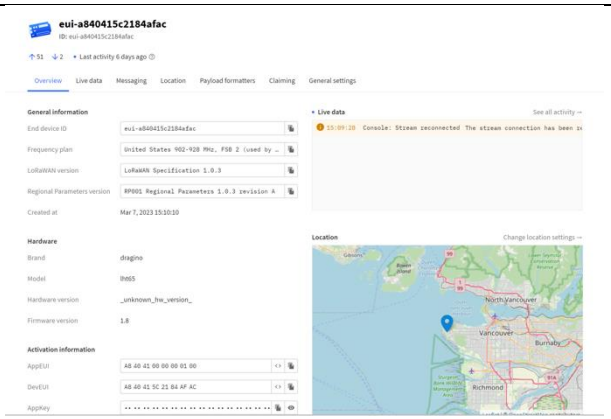


Figure 87: The end device page on TTN once registered.

Behavior on ACT	Function	Action
Pressing ACT between 1s < time < 3s	Test uplink status	If LHT65 is already Joined to LoRaWAN network, LHT65 will send an uplink packet, if LHT65 has external sensor connected, blue led will blink once. If LHT65 has not external sensor, red led will blink once.
Pressing ACT for more than 3s	Active Device	green led will fast blink 5 times, device will enter working mode and start to JOIN LoRaWAN network. green led will solid turn on for 5 seconds after joined in network.
Fast press ACT 5 times.	Deactivate Device	red led will solid on for 5 seconds. Means LHT65 are in Deep Sleep Mode.

Figure 88: Functionalities of the blue button (ACT button) at the bottom of the Dragino Sensor.

11. Verify data transmission:
Click the registered device to open its details page.

Under the "Live data" tab, you should see the received data from the Dragino sensor.

See the user manual for the Dragino Sensor for more detail.

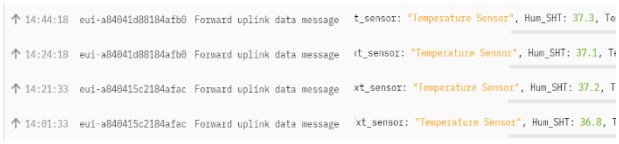


Figure 89: Example of data received from the Dragino Sensor on TTN.

4.8 Connect TTN to ThingSpeak

Required Item:

- None

Required Tool:

- Computer

1. On the computer, open the TTN console page as described in step 4.7.

   Click “Application” and enter your application. Select one of the end devices.

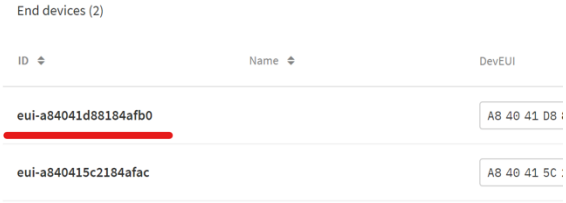


Figure 90: Click one of the registered end devices first.

2. On your end device page, select “Playload formatters”. Make sure you in the “Uplink” tab.

   Change the Formatter type to be “Custom Javascript formatter”.

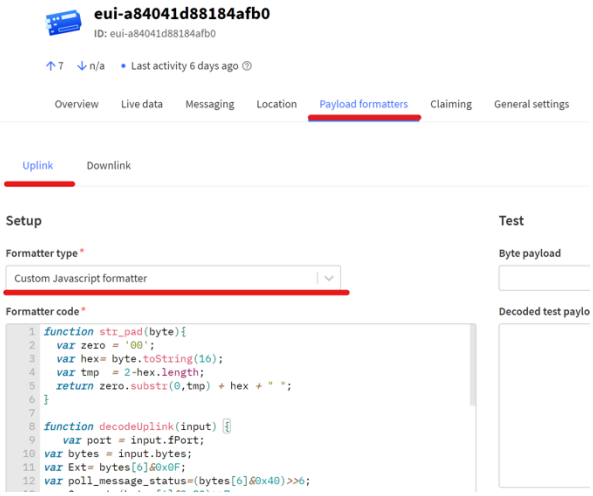


Figure 91: Go to "Payload formatters" and select "Custom Javascript formatter" under "Uplink" tab.

3. Go to the GitHub repository.

   Go to the “Dragino Payload” folder, and open “TTN_Uplink_Payload.js”. Copy the code.

   Overwrite the formatter code on TTN with the copied code.

   Click “Save changes”.

   Repeat step 1-3 for the other end device.

Figure 92: Once payload code is update. Click "Save changes".

4. Back to the TTN console page. Click “Application” and choose the application.

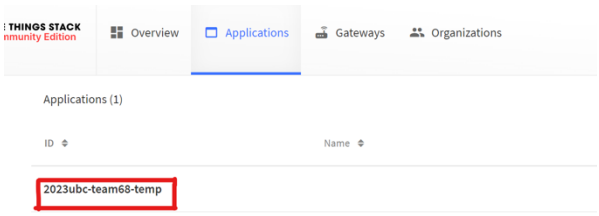


Figure 93: Go to application.

5. Select "Integrations" from the left menu, and choose "Webhooks" from the dropdown menu.

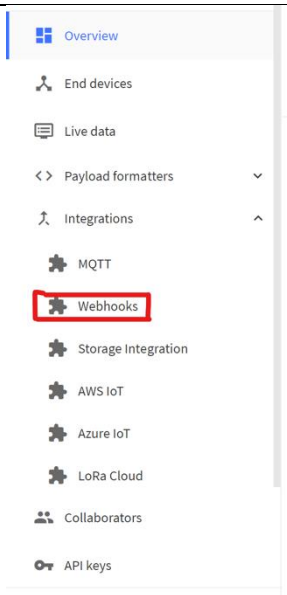


Figure 94: Select "Webhooks" from the "Integration" menu.

6. Click “Add webhook” on the top right of the screen.

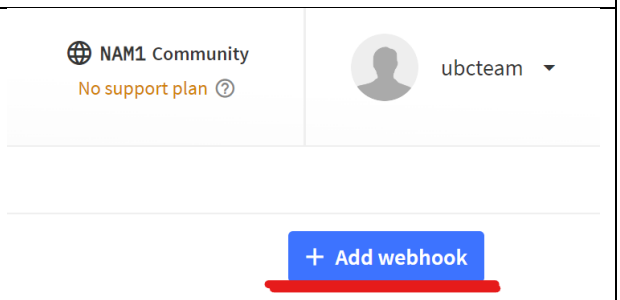


Figure 95: Click "Add webhook".

7. Locate and click “ThingSpeak” from the list.

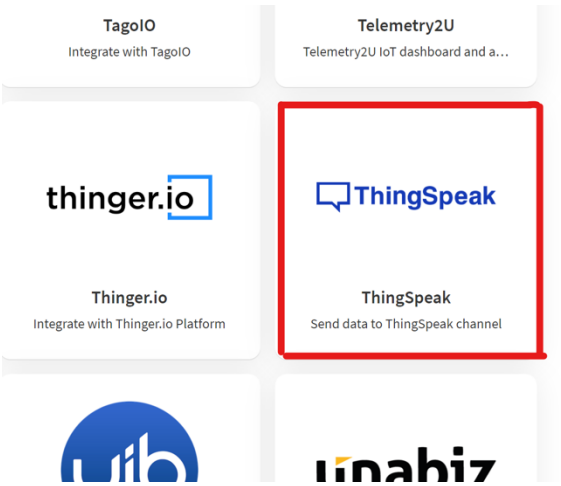


Figure 96: Locate and click "ThingSpeak" from the list.

8. Fill in the "Webhook ID"(Name of the channel), "Channel ID", and "API Key" from the ThingSpeak channel. See step 4.9 and 4.10.

   Click "Create ThingSpeak webhook" to complete the webhook registration.

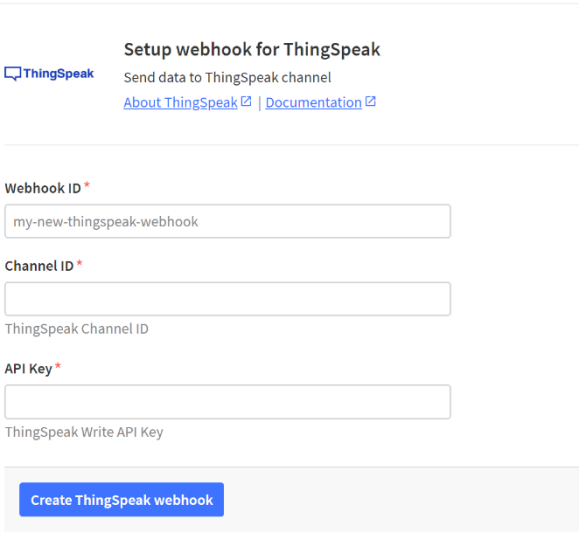


Figure 97: Fill in the "Webhook ID", "Channel ID", and "API Key" to configure the webhook.

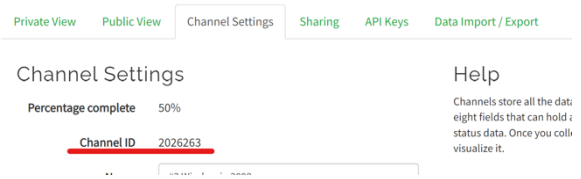
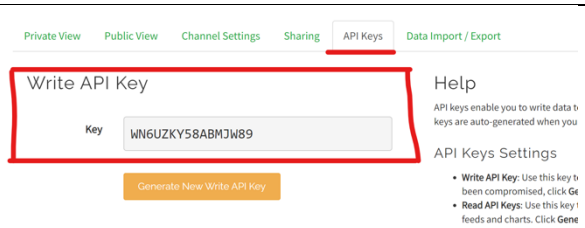
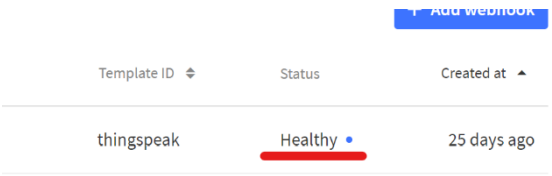


Figure 98: Find channel ID on ThingSpeak.

	 *Figure 99: Find "API Key" on ThingSpeak.*
9. Wait for 10 minutes and then check if the connection status is “Healthy” on the Webhooks page on TTN.	 *Figure 100: Wait for 10 minutes and check if the connection status is "Healthy" on TTN.*

4.9 Set up ThingSpeak Dashboard

Required Item: • None **Required Tool:** • Computer	

1. Go to the ThingSpeak website (https://thingspeak.com/) and log in with the provided email and password.

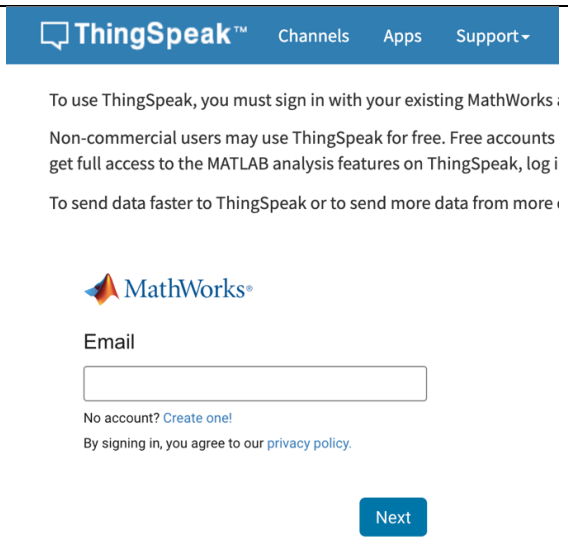


Figure 101: Log in to ThingSpeak.

2. Once logged in, click "Channels" from the top menu bar and select "My Channels."

   Click "New Channel" to create a new dashboard.

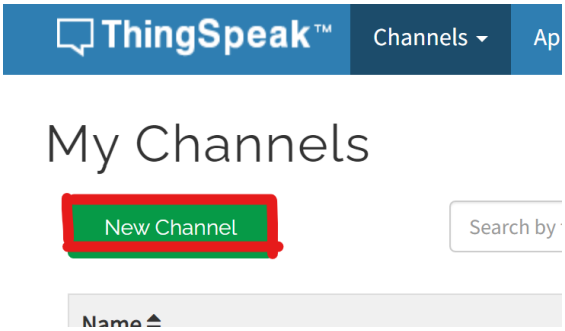


Figure 102: Click "New Channel" to create a new dashboard.

3. Configure the channel:

   Fill in the "Name" and "Description" for the channel as needed. Also see section 3.4.

   Enable up to Field 7 by clicking the checkboxes.

   Fill in the field name as follows:
   Field 1: Inside_Surface_Temp
   Field 2: Outside_Surface_Temp
   Field 3: Outside_Air_Temp
   Field 4: Heat Flow Rate
   Field 5: Number of Collision
   Field 6: Collision Data from Arduino
   Field 7: Fall Detection

   Click "Save Channel" to create the channel.

   Field names and orders are arbitrary, but it is recommended to keep them consistent for easy troubleshooting.

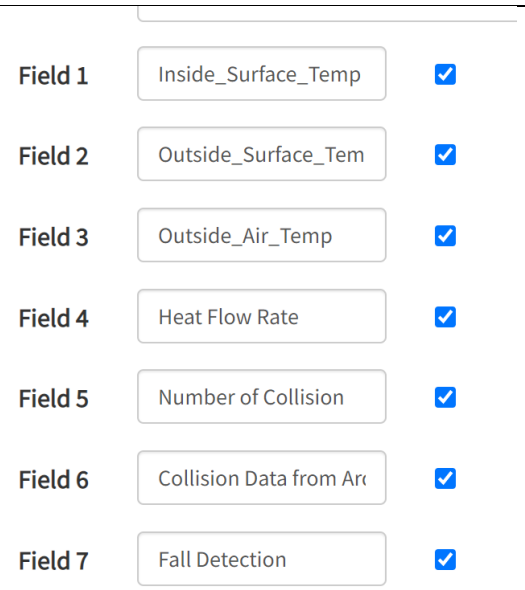


Figure 103: Enable seven fields and fill in their names.

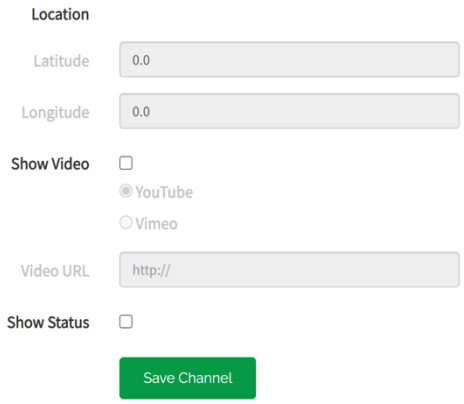


Figure 104: Click "Save Channel".

4. In the channel created, click "Add Widgets".

   Select the widget type from the options shown in Figure 106.

   Configure the widget as shown in Figure 107. Click “Create” to create a widget.

   Create 6 Numeric Display widgets as shown in Table 3 below.

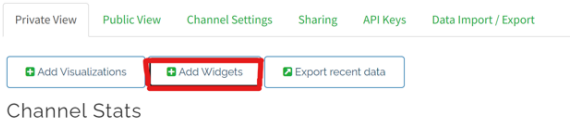


Figure 105: Click "Add Widgets" to create widgets where data are displayed.

Table 3: Numeric Display widgets configurations.

Field #	Update Interval	Unit
1	1800	Celsius
2	1800	Celsius
3	1800	Celsius
4	15	Watts
5	15	
7	15	

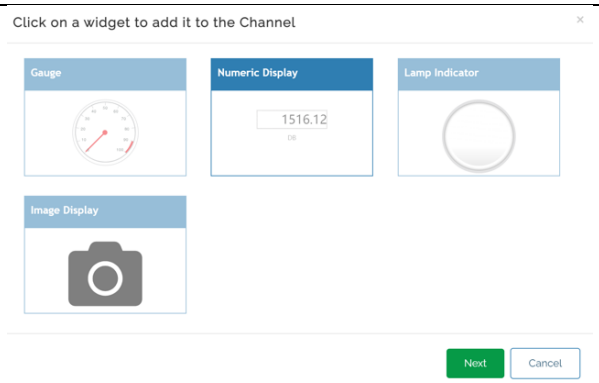


Figure 106: Select widget type.

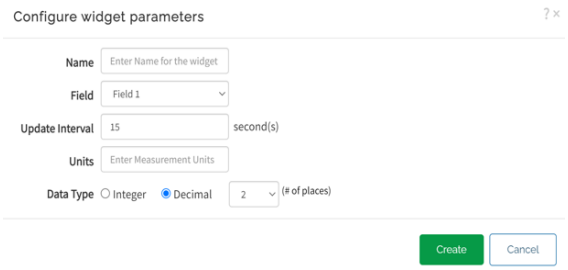


Figure 107: Configure a widget by selecting a channel created.

5. Click “Add Visualizations”. Select “Field X Chart”, then click “Save” to add a chart to the dashboard.

   Click the pencil icon to configure the chart as shown in Figure 108.

   Click “Type” to change the graph type. Then click “Save” to save the chart.

   Create 6 charts as shown in Table 4 below.

Table 4: Charts configurations.

Field #	Type	Results
1	line	60
2	line	60
3	line	60
4	line	60
5	step	60
7	column	60

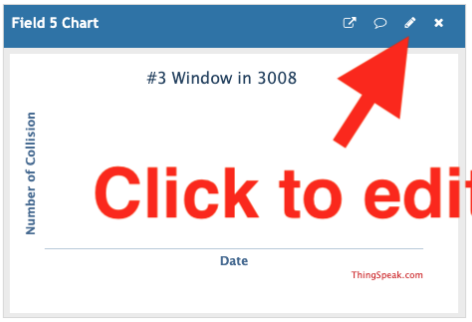


Figure 108: Click the pencil icon to configure the chart.

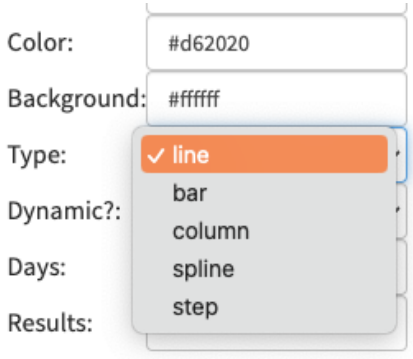


Figure 109: Click "Type" to change the graph type.

6. Click “Apps” at the top, then “MATLAB Analysis”.

   Click “New”, select “Custom (no starter code”, then click “Create”.

   Go to the [GitHub repository](#).

   Go to the “ThingSpeak MATLAB” folder, and open “Setup.m”.

   Copy and paste the code to ThingSpeak.

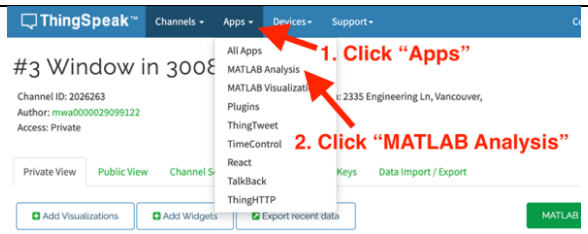


Figure 110: Click "Apps", then "MATLAB Analysis".

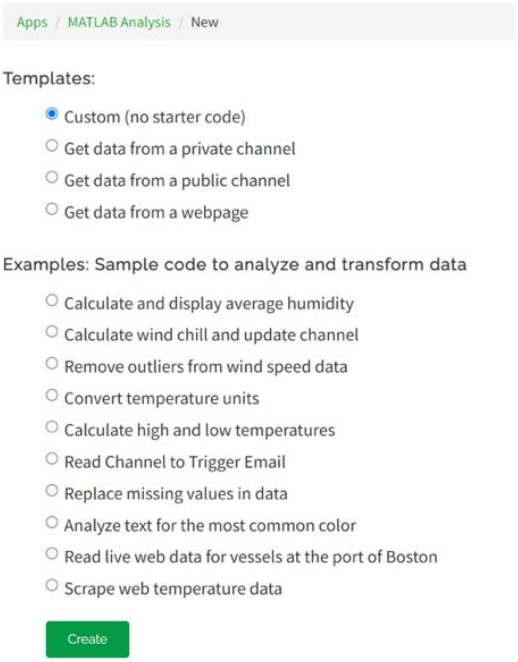


Figure 111: Click "New", then select "Custom (no starter code)". Then click "Create".

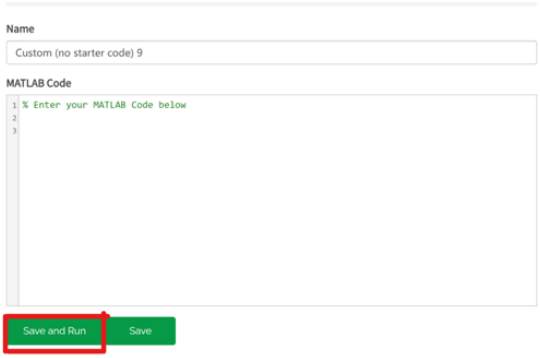


Figure 112: Copy and paste the code from GitHub, and click "Save and Run".

7. Update the channel ID, "write" and/or "read" API Key at the top of each code block, to match the values on the right-hand side.

   Click “Save and Run” to save code.

   Repeat step 6-7 for other files on GitHub:
   “Outside_Air_Temp.m”
   “Detection_Once_Per_Day.m”
   “Detection_Every_Five_Minutes.m”
   “Fall_Detection.m”
   “Heat_Flow_Rate.m”

Figure 113: Make sure the channel ID, "write" and/or "read" API Key in the code match with the values on the right-hand side.

8. Adding Time Controls:

   Click “Apps” at the top, then “Time Control”. Then click “New Time Control”.

   Select “Recurring”. Select one of the code from step 6 from the drop-down list at the bottom of the page.

   Configure the following Recurrence:
   “Outside_Air_Temp.m”: 30 min
   “Detection_Once_Per_Day.m”: 11:59pm
   “Detection_Every_Five_Minutes.m”: 5 min
   “Fall_Detection.m”: 5 min
   “Heat_Flow_Rate.m”: 30 min

   Then Click “Save TimeControl”.

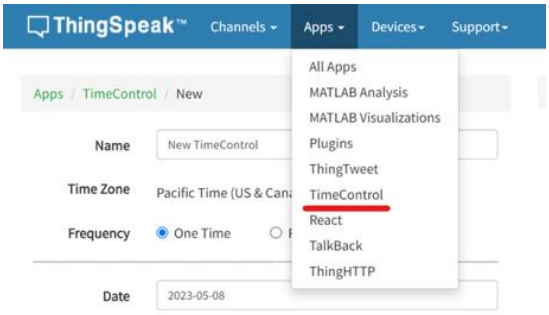


Figure 114: Click "Apps", then "TimeControl".

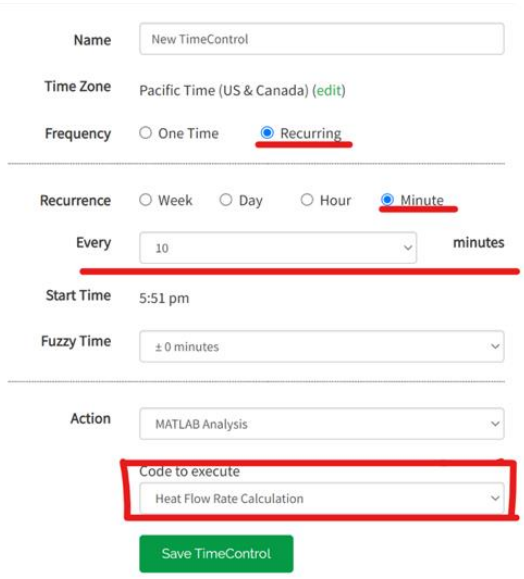
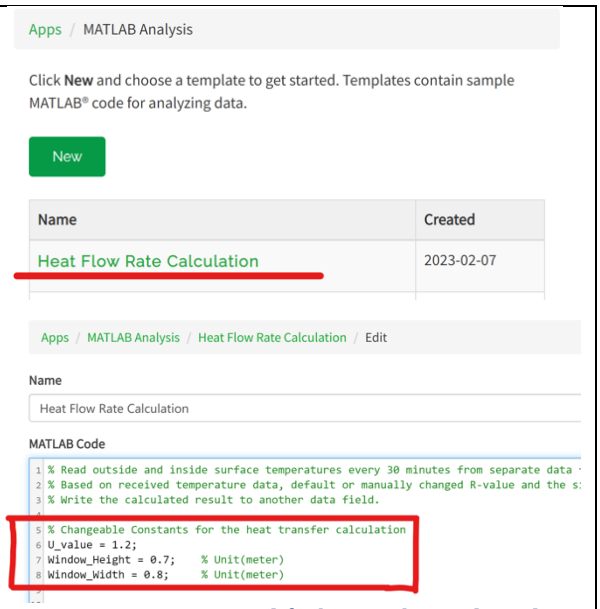
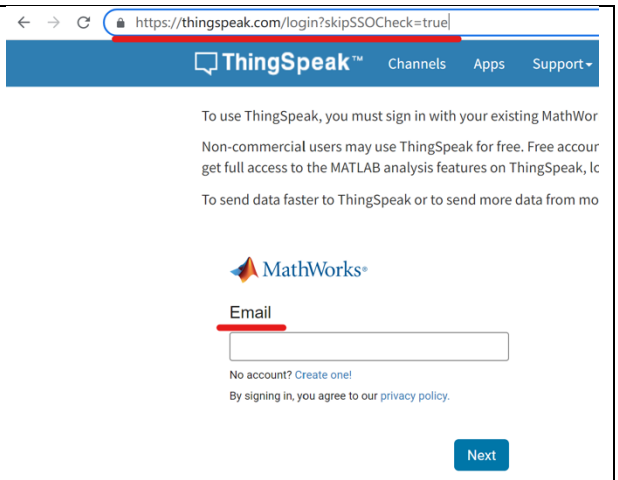
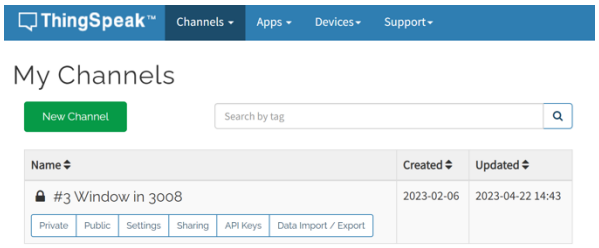
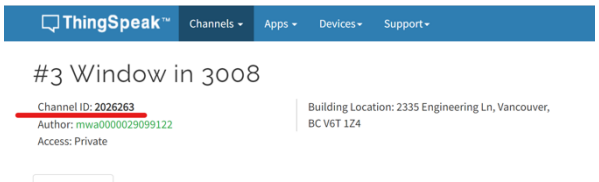


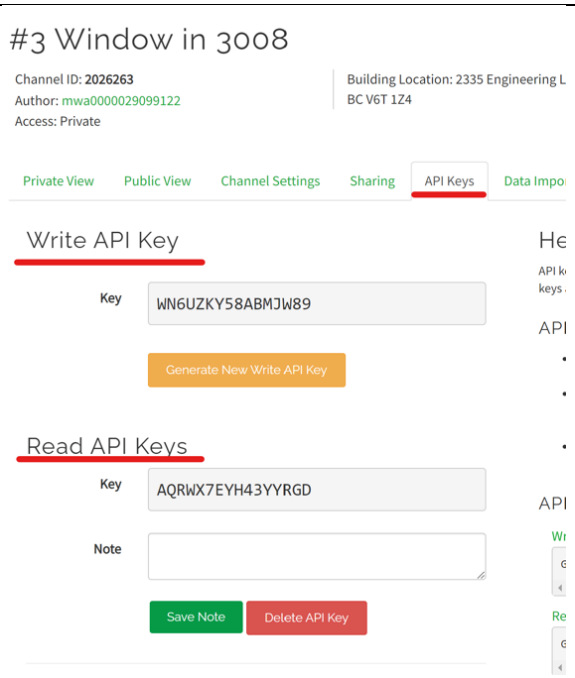
Figure 115: Configure TimeControl for each code.

9. **Adjust R-value and window size:** Click “Apps” at the top, then “MATLAB Analysis”. Then select “Heat Flow Rate Calculation” MATLAB code. You can modify the U-value and window size by changing the number. Make sure to click “Save and Run”.	 *Figure 116: You can modify the U value and window dimension.*

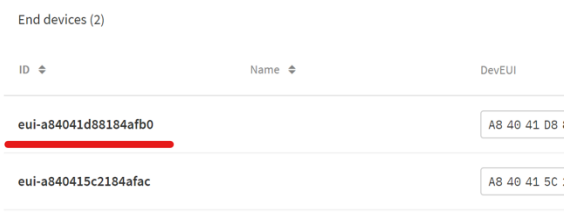
4.10 Finding Channel ID and API key

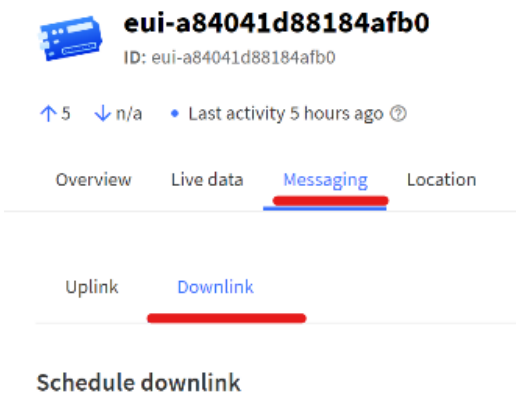
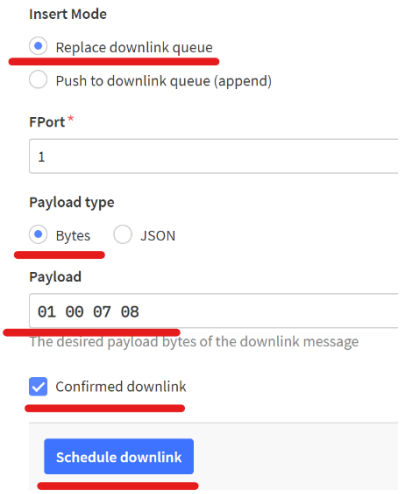
Required Item: • None **Required Tool:** • Computer	

1. Navigate to the ThingSpeak website (https://thingspeak.com/)	 *Figure 117: Log in to ThingSpeak.*
2. Enter the channel you want to check.	 *Figure 118: Select the appropriate channel.*
3. On the top of the channel page, you can find the channel ID there.	 *Figure 119: You can find Chanel ID at the top of the dashboard.*

4. Click “API Key” to find the API keys	 *Figure 120: Click "API Keys" to find Write/Read API keys.*

4.11 Change the Updating frequency of Dragino Sensor

Required Item: • None **Required Tool:** • Computer	
1. On the computer, open the TTN console page as described in step 4.7. Click “Application” and enter your application. Select one of the end devices.	 *Figure 121: Click one of the registered end devices first.*

2. On the end device page, select "Messaging". Make sure the "Downlink" tab is selected.	 *Figure 122: Select the appropriate channel.*
3. Fill in the following information to change the sensor uploading interval to be 30 minutes. **Payload type:** "Bytes" **Payload:** "01 00 07 08" (The "…7 08" is the Hex number of 1800 seconds which is 30 minutes.) Click Schedule downlink to confirm change. Repeat Step 1-3 for another sensor.	 *Figure 123: You can find Change the updating time by sending a downlink playload to Dragino sensor.*