**3. Design files**

Design files summary.

| **Design file name** | **File type** | **Open source license** | **Location of the file** |
| --- | --- | --- | --- |
| eGH\_Sensor\_Package | .ino |  |  |
| config | .h |  |  |
| eGreenhouseJSON | .cpp |  |  |
| eGreenhouseJSON | .h |  |  |
| GUI | .pde |  |  |
| Timer | .pde |  |  |
| Hub\_Receive | .ino |  |  |
| config | .h |  |  |
| eGreenhouseJSON | .cpp |  |  |
| eGreenhouseJSON | .h |  |  |
| Hub\_Transmit | .ino |  |  |
| config | .h |  |  |
| hyperJSON | .cpp |  |  |
| hyperJSON | .h |  |  |
| Hyper | .ino |  |  |
| config | .h |  |  |
| hyperJSON | .cpp |  |  |
| hyperJSON | .h |  |  |
| HyperRail\_Driver | .cpp |  |  |
| HyperRail\_Driver | .h |  |  |
| eGreenhousePCBpowerboost | .brd |  |  |
| eGH Sensor Package222 | .png |  |  |
| eGH Sensor Packagev5 | .png |  |  |
|  |  |  |  |
|  |  |  |  |

**Hub\_Transmit**: INO file used for gathering User Input from GUI.pde then send to HyperDrive.

**config:** H file used for the Hub\_Transmit.ino file for enabling hardware features such as radio communication.

**hyperJSON**: CPP file and h file used for communication between Hyper and Hub\_Transmit.

**Hyper:** INO file used for moving the HyperRail from User Input and send values and indication to start measure to the eGH\_Sensor\_Package.

**config:** H file used for Hyper.ino file for enabling hardware features such as radio communication.

**hyperJSON**: CPP file and h file used for communication between Hyper and Hub\_Transmit.

**HyperRail\_Driver**: CPP file and h file used for adding functions of the movement of the HyperRail.

**eGH\_Sensor\_Package:** INO file used for measuring K30, SHT31-D, and TSL2591 then log to SD card and send values to the Hub\_Receive.

**config:** H file used for eGH\_Sensor\_Package.ino file for enabling hardware features such as sensors and radio communication.

**eGreenhouseJSON:** CPP file and H file for communication between eGH\_Sensor\_Package.ino and Hub\_Receive.ino.

**Hub\_Receive**: INO file used for receiving sensor values and coordinates, then publish to GoogleSheets.

**config:** H file used for Hub\_Receive.ino file for enabling hardware features such as address for GoogleSheets and radio communication.

**eGreenhouseJSON:** CPP file and H file for communication between eGH\_Sensor\_Package.ino and Hub\_Receive.ino.

**GUI**: PDE file that is used for the GUI that take User Input

**Timer**: PDE file that enable a built-in timer for the GUI file.

**eGreenhousePCBpowerboost:** BRD file used for layout of the sensor location.

**eGH Sensor Package222:** PNG file used for the top view of the eGreenHouse\_Sensor\_Collector.

**eGH Sensor Packagev5:** PNG file used for the side view of the eGreenHouse\_Sensor\_Collector.

[The Design files for the 1D HyperRail]

**4. Bill of materials**

Materials for eGH\_Sensor\_Package

| **Designator** | | **Component** | **Number** | **Cost per unit** | **Total cost** | **Source of materials** | **Material** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Development  Board | | Adafruit Feather M0 with RFM95 LoRa Radio - 900MHz - RadioFruit | 1 | $34.95 | $34.95 | Adafruit | Non-specific |
| Antenna Kit | 900Mhz Antenna Kit - For LoPy, LoRa, etc | | 1 | $12.75 | $12.75 | Adafruit | Non-specific |
| Antenna Connector | uFL SMT Antenna Connector | | 1 | $0.75 | $0.75 | Adafruit | Non-specific |
|  | |  |  |  |  |  |  |
| Data Logger  with RTC Board | | Adalogger FeatherWing - RTC + SD Add-on For All Feather Boards | 1 | $8.95 | $8.95 | Adafruit | Non-specific |
| Micro SD Card | | 16 GB MicroSD Card | 1 | $7.00 | $7.00 | Amazon | Non-specific |
| CO2 Sensor | | K30 10,000ppm CO2 Sensor | 1 | $85.00 \*\*\* | $85.00 | CO2Meter | Non-specific |
| Light Sensor | | Adafruit TSL2591 High Dynamic Range Digital Light Sensor | 1 | $6.95 | $6.95 | Adafruit | Non-specific |
| Temperature  and Humidity  Sensor | | Adafruit Sensirion SHT31-D - Temperature & Humidity Sensor | 1 | $13.95 | $13.95 | Adafruit | Non-specific |
| Sensor PCB | | Custom Component | 1 |  |  |  |  |
| Coin Cell Battery | | 1220, Button Cell Battery, Lithium, 3VDC, Diameter 0.472", Depth 0.078" | 1 | $0.68 | $0.68 | Grainger | Non-specific |
| Board Headers | | Stacking Headers for Feather - 12-pin and 16-pin female headers | 1 | $1.25 | $1.25 | Adafruit | Non-specific |
| Board Headers | | Short Feather Male Headers - 12-pin and 16-pin Male Header Set | 1 | $0.50 | $0.50 | Adafruit | Non-specific |

Pricing Notes

*\* price will vary with source*

*\*\* optional*

*\*\*\* price will vary with capacity/Option*

Additional equipment that may be needed:

SLA 3D printer, such as Form 2, to print the sensor adapter

Soldering station for soldering sensor and other parts to PCB

Materials for Hub\_Transmit and Hub\_Receive

| **Designator** | **Component** | **Number** | **Cost per unit** | **Total cost** | **Source of materials** | **Material** |
| --- | --- | --- | --- | --- | --- | --- |
| Development  Board | Adafruit Feather M0 with RFM95 LoRa Radio - 900MHz - RadioFruit | 2 | $34.95 | $34.95 | Adafruit | Non-specific |
| Antenna Kit | 900Mhz Antenna Kit - For LoPy, LoRa, etc | 2 | $12.75 | $12.75 | Adafruit | Non-specific |
| Antenna Connector | uFL SMT Antenna Connector | 2 | $0.75 | $0.75 | Adafruit | Non-specific |
| Ethernet Connector | Adafruit Ethernet FeatherWing | 1 | $19.95 | $19.95 | Adafruit | Non-specific |
| Battery | Lithium Ion Battery Pack - 3.7V 6600mAh | 1 | $29.50 | $29.50 | Adafruit | Ion |
| Board Headers | Short Headers Kit for Feather - 12-pin + 16-pin Stacking Headers | 1 | $1.50 | $1.50 | Adafruit | Non-specific |
| Board Headers | Short Feather Male Headers - 12-pin and 16-pin Male Header Set | 2 | $0.50 | $0.50 | Adafruit | Non-specific |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Pricing Notes

*\* price will vary with source*

*\*\* optional*

*\*\*\* price will vary with capacity*

Additional equipment that may be needed:

SLA 3D printer, such as Form 2, to print the sensor adapter

Soldering station for soldering sensor and other parts to PCB

[The bill of material for the 1D HyperRail]

## 5. Build instructions

## [Need to add the guide for the HyperRail Build Instruction]

### 5.1. eGreenhouse Sensor Package Hardware Setup

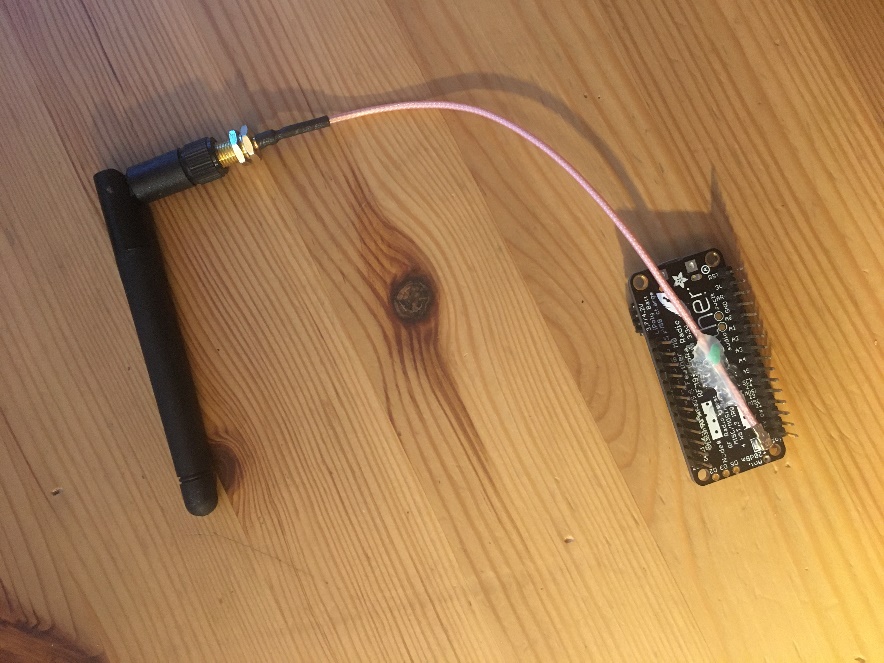
#### **5.1.1. Development Board**

For this part, you need the development board, antenna kit, antenna connector, and both 16-pin and 12-pin male headers. First solder the both 16-pin and 12-pin headers to the development board. Once that is complete, then solder the antenna connector where it says Ant. +20dBm on the back of the development board. The yellow arrow in figure 1 will indicate where the location is.



(Figure 1: The back side of the Development Board where the yellow is showing where to solder the antenna connector)

Once that is complete, then connect the antenna cable to the antenna connector and connect the antenna to the antenna cable. Once that is complete, you are set with the development board set up. The result will be in figure 2.



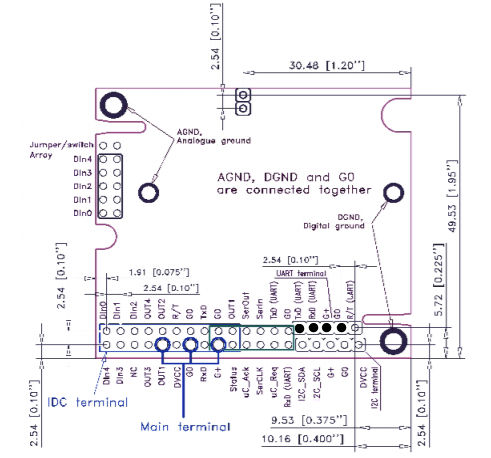
(Figure 2: The complete look for the Development Board)

#### **5.1.2. Data Logger with RTC Board**

For this step, you need Data Logger with RTC Board, Micro SD Card, Coin Cell Battery, and stacking 16-pin and 12-pin headers. First, solder the stacking 16-pin and 12-pin headers to the Data Logger with RTC board. Once that is complete, then insert the coin cell battery into the coin battery slot and insert the Micro SD Card to the Micro SD Card slot. Once that is complete, you are set with the Data Logger with RTC.

#### **5.1.3. CO2 Sensor**

For this step, you need 4-pin male headers and the CO2 sensor. First solder the 4-pin male headers to the location that is colored in black in figure 3.

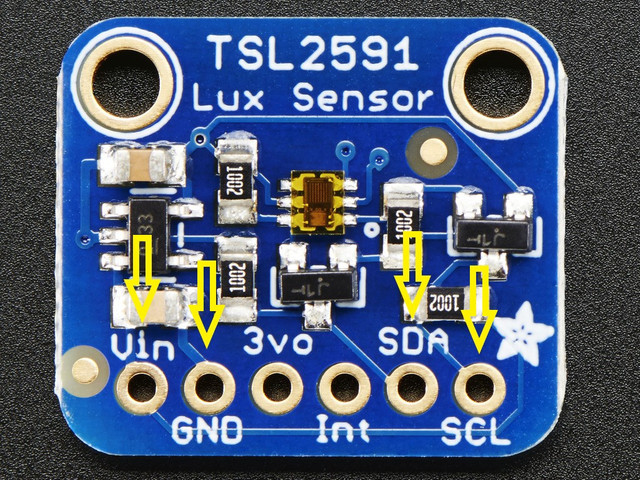


(Figure 3: The Layout of the CO2 K30 Sensor)

Once that is complete, then you are done with CO2 sensor setup.

#### **5.1.4. Light Sensor**

For this step, you need 6-pin male headers and the light sensor. Solder the 6-pin headers to the designated area as you find in figure 4. For now, you can ignore the yellow arrows.



(Figure 4: The Layout of the Light TSL2591 Sensor with four yellow arrows pointing some pin location)

Once that is complete, then you are done with setting up with the light sensor.

#### **5.1.5. Humanity and Temperature Sensor**

For this step, you need 7-pin male headers and humanity and temperature sensor. Solder the 7-pin headers to the designated area as you find in figure 5. For now, you can ignore the yellow arrows.



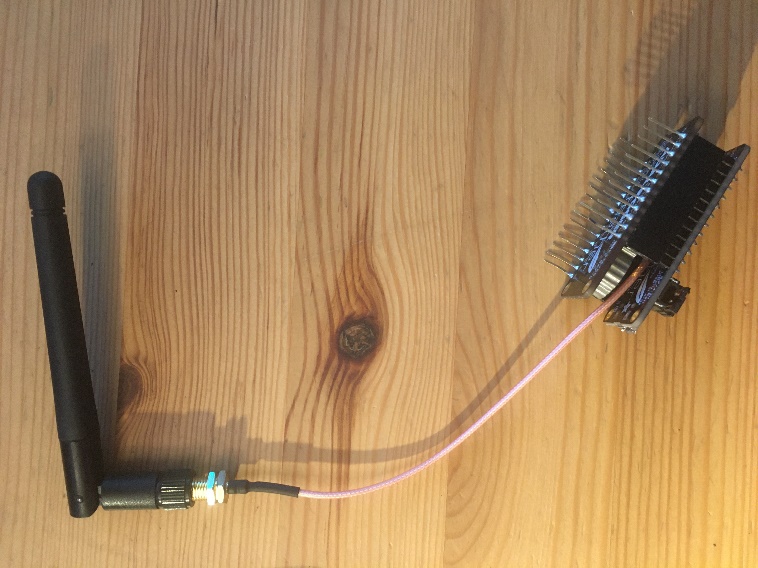
(Figure 5: The Layout of the Humanity and Temperature SFHT31-D Sensor with four yellow arrows pointing some pin location)

Once that is complete, then you are done with setting up with the humanity and temperature sensor.

### 5.2. eGreenhouse Sensor Package Setup

#### **5.2.1. Connect Development Board to the Data Logger with RTC Board**

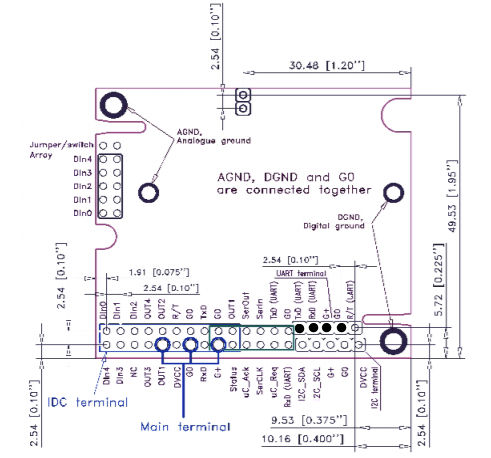
For this step, you need the development board that is soldered with the 16-pin and 12-pin male headers and Data Logger with RTC board that is soldered with the 16-pin and 12-pin stacking headers. As you can see, you can insert the male headers to the stacking headers. Put the development board on top of the Data Logger wit RTC board. Once that is complete, it will look in figure 6 with a completion.



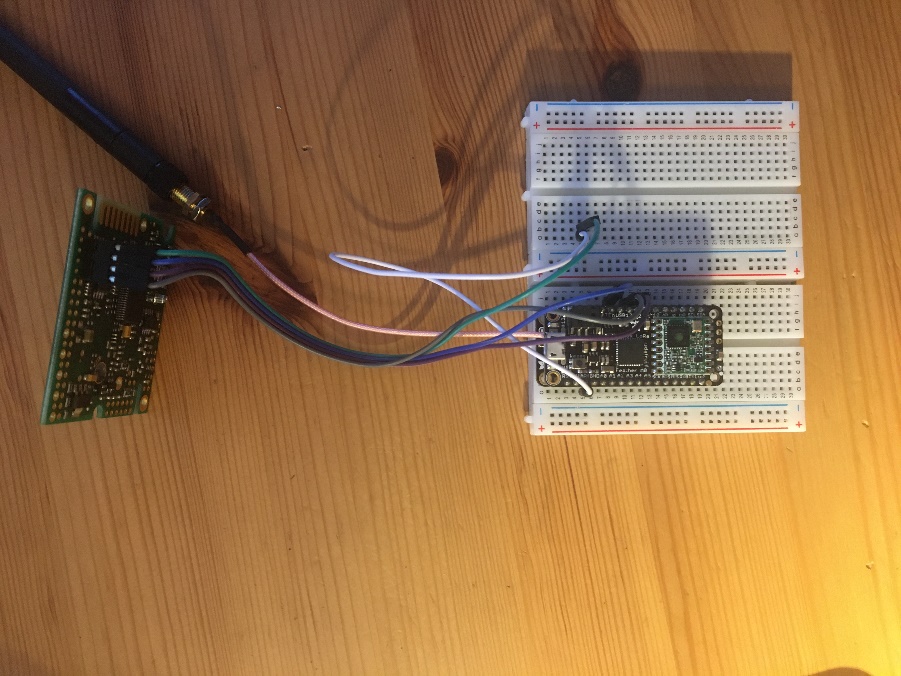
(Figure 6: The complete look for connecting the Development Board and Data Logger with RTC Board)

#### **5.2.2. Connect CO2 Sensor to the Development Board**

For this step, you need the complete soldered CO2 sensor, the development board that is connected to the Data Logger with RTC board, 4 jumper wires that are male-to-female, and breadboard. First insert the board into the breadboard. Once you inserted, according to the position of figure 7, the sensor of TxD will be connected to the D12 pin on the board, the sensor of RxD will be connected to the D11 pin on the board, the sensor of G+ will be connected to the USB on the board, and the sensor of G0 will be connected to the GND pin on the board. The four black dots in figure 7 will indicate the location for each pin. Once you complete, it will look like figure 8.



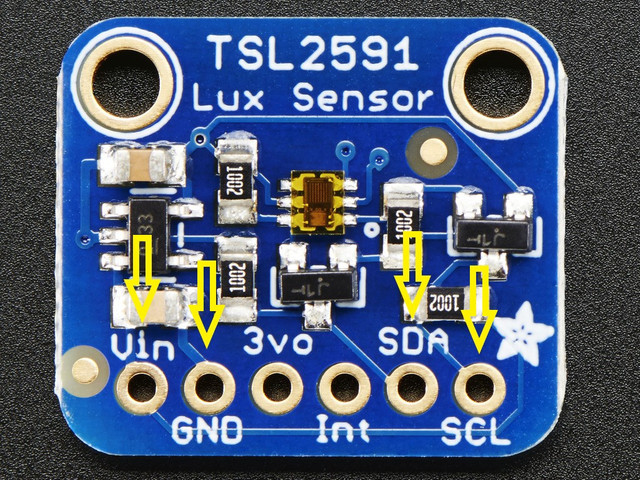
(Figure 7: The Layout of the CO2 K30 Sensor)



(Figure 8: The complete look of connecting the CO2 Sensor to the Development Board)

#### **5.2.3. Connect Light Sensor and Humanity and temperature sensor to the Development Board**

For this step, you need the complete soldered light sensor, the humanity and temperature sensor, and 8 jumper wires that are male-to-female. First, connect 4 jumper wires to both sensors that are label as Vin, GND, SCL, and SDA. Once that is complete, connect the Vin pin to the 3V on the development board, GND pin to the GND on the development board, SCL pin to the SCL on the development board, and SDA pin to the SDA on the development board. For reference figure 9 will be the light sensor and figure 10 will be the humanity and temperature sensor pin locations. The yellow arrows will show the location of it.

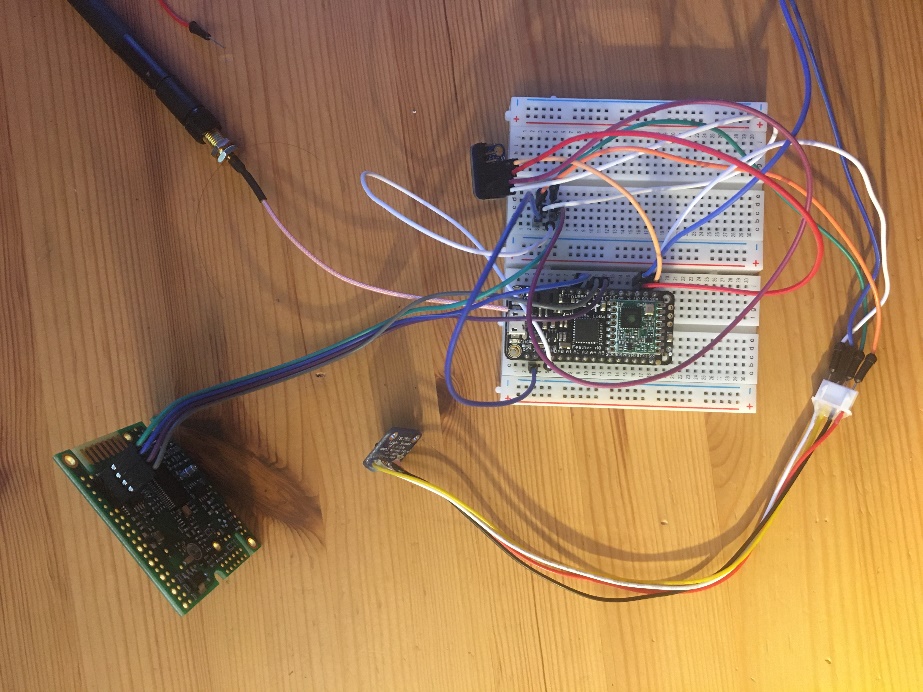


(Figure 9: The Layout of the Light TSL2591 Sensor with four yellow arrows pointing some pin location)



(Figure 10: The Layout of the Humanity and Temperature SFHT31-D Sensor with four yellow arrows pointing some pin location)

Once that step is complete, it should look like figure 11 with a completion of setting up the eGreenhouse Sensor Package.



(Figure 11: The complete look for the eGreenhouse Sensor Package)

### 5.3. Hub\_Transmit Hardware Setup

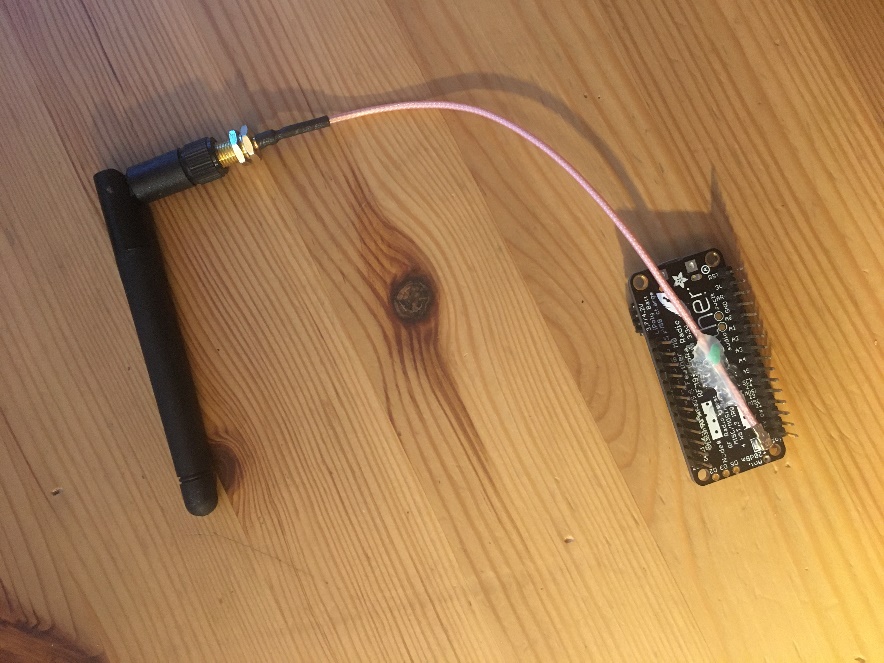
#### **5.3.1. Development Board**

For this part, you need the development board, antenna kit, antenna connector, and both 16-pin and 12-pin male headers. First solder the both 16-pin and 12-pin headers to the development board. Once that is complete, then solder the antenna connector where it says Ant. +20dBm on the back of the development board. The yellow arrow in figure 12 will indicate where the location is.



(Figure 12: The back side of the Development Board where the yellow is showing where to solder the antenna connector)

Once that is complete, then connect the antenna cable to the antenna connector and connect the antenna to the antenna cable. Once that is complete, you are set with the development board set up. The result will be in figure 13.



(Figure 13: The complete look for the Development Board)

### 5.4. Hub\_Receive Setup

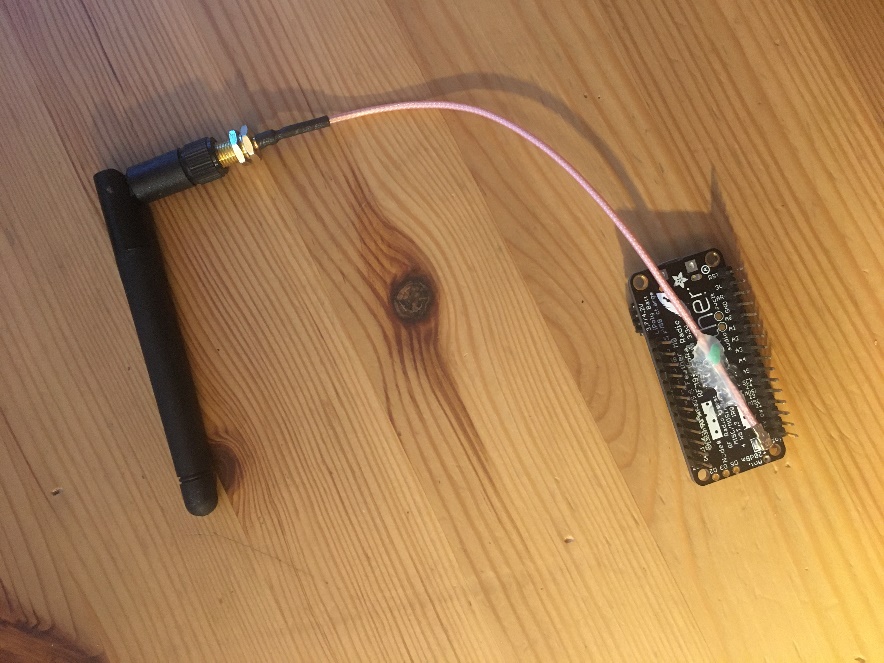
#### **5.4.1. Development Board**

For this part, you need the development board, antenna kit, antenna connector, and both 16-pin and 12-pin stacking headers. First solder the both 16-pin and 12-pin headers to the development board. Once that is complete, then solder the antenna connector where it says Ant. +20dBm on the back of the development board. The yellow arrow in figure 14 will indicate where the location is.



(Figure 14: The back side of the Development Board where the yellow is showing where to solder the antenna connector)

Once that is complete, then connect the antenna cable to the antenna connector and connect the antenna to the antenna cable. Once that is complete, you are set with the development board set up. The result will be in figure 15.



(Figure 15: The complete look for the Development Board)

#### **5.4.2. Ethernet Connector**

For this step, you need Ethernet Connector Board and male 16-pin and 12-pin headers. Solder the male 16-pin and 12-pin headers to the Ethernet Connector board. Once that is complete, you complete this step.

#### **5.4.3. Connect Ethernet Connector to Development Board**

For this step, you need the development board that is soldered with the 16-pin and 12-pin stacking headers and Ethernet Connector board that is soldered with the 16-pin and 12-pin male headers. As you can see, you can insert the male headers to the stacking headers. Put the Ethernet Connector board on top of the Development board. Once you complete, you are set with the Hub\_Receive.

I don’t know how to do the citation of the 1D HyperRail paper for this. However, it should be the same.

## 6. Operation instructions

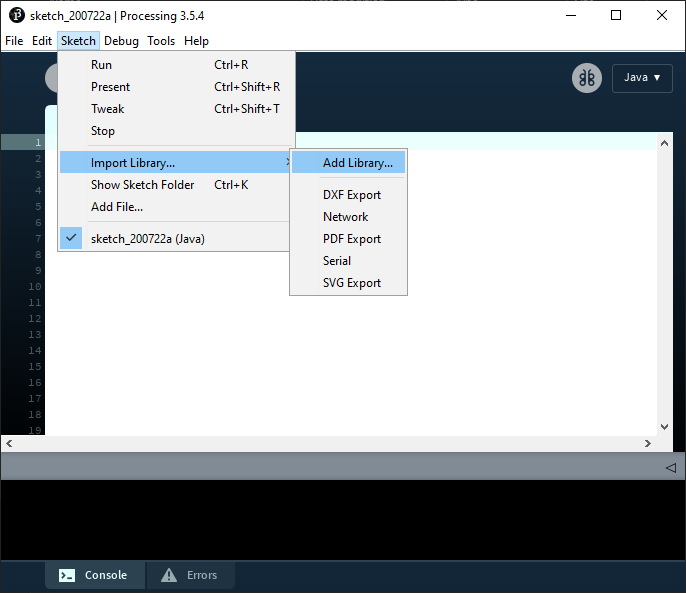
### 6.1. Software Download

#### **6.1.1. Windows**

Download Git Bash Shell, Arduino IDE, and Processing. Once you download Git Bash Shell, open it and enter the following command. This should install all the dependencies and libraries to operate properly.

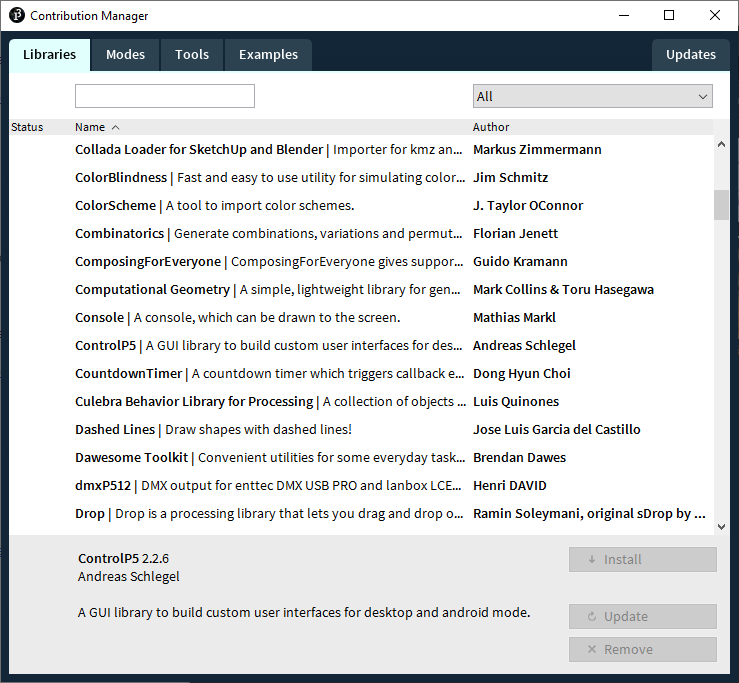
curl -fsSL https://raw.githubusercontent.com/OPEnSLab-OSU/Loom/master/setup/setup-windows.sh | sh

Once it says it complete, then open Processing. Then go to Sketch -> Import Library… -> Add Library… Look at Figure 16 for location of the option.



(Figure 16: Processing Application where to select to add libraries to the board)

Once you select Add Library, you will get a pop up like in Figure 17. On the top left, search both Control5P and Arduino(Firmate) and install both.

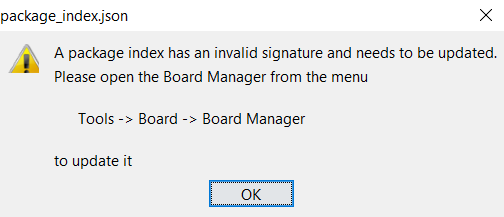


(Figure 17: The Processing Contribution Manager for installing new Libraries for the Processing)

Once you are complete, then go to Documents -> Arduino -> libraries. In that folder, add the dependencies that is in the Design file.

Once you complete, then you are set up with the software.

Note: when you open the Arduino IDE, you will get the following message in figure 18. You can simply ignore that.



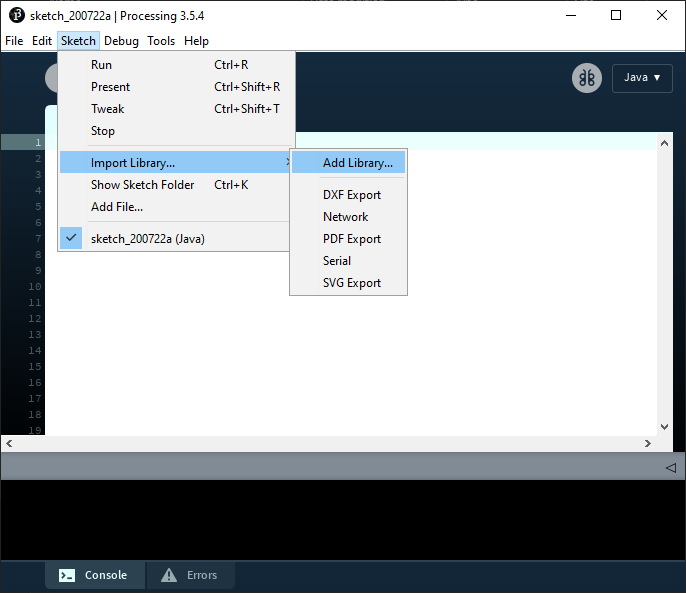
(Figure 18: Warring Message after installing the Arduino and the Command line, but simply can be ignored)

#### **6.1.2. Mac**

Unlike Windows, Mac users can open their terminal to install Arduino and its dependencies and libraries by entering the following command line.

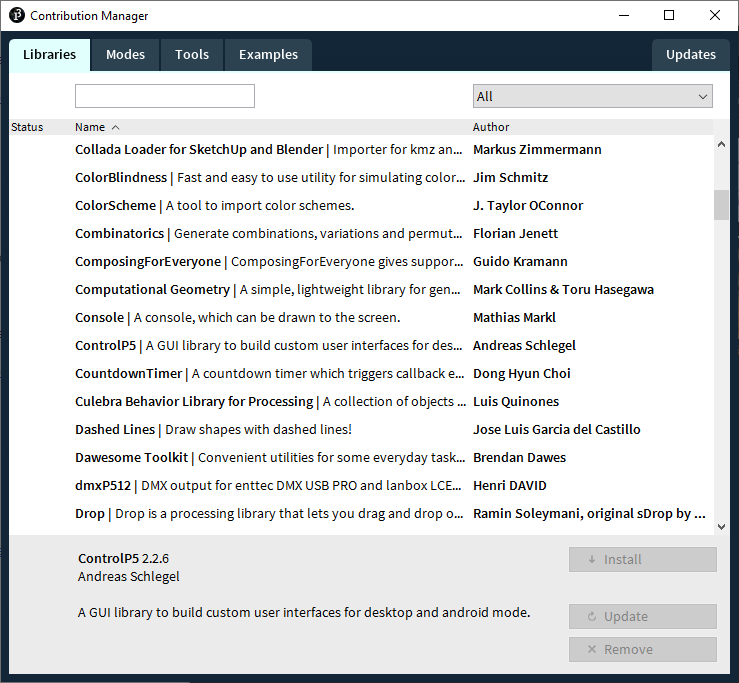
curl -fsSL https://raw.githubusercontent.com/OPEnSLab-OSU/Loom/master/setup/setup-mac.sh | sh

Once that is complete, download Processing and open it. Then go to Sketch -> Import Library… -> Add Library… Look at Figure 19 for location of the option.



(Figure 19: Processing Application where to select to add libraries to the board)

Once you select Add Library, you will get a pop up like in Figure 20. On the top left, search both Control5P and Arduino(Firmate) and install both.

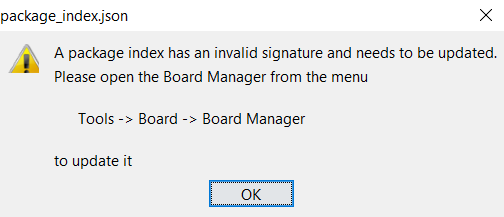


(Figure 20: The Processing Contribution Manager for installing new Libraries for the Processing)

Once you are complete, go to Documents -> Arduino -> libraries. In that folder, add the dependencies that is in the Design file.

Once you complete, then you are done with setting up the software.

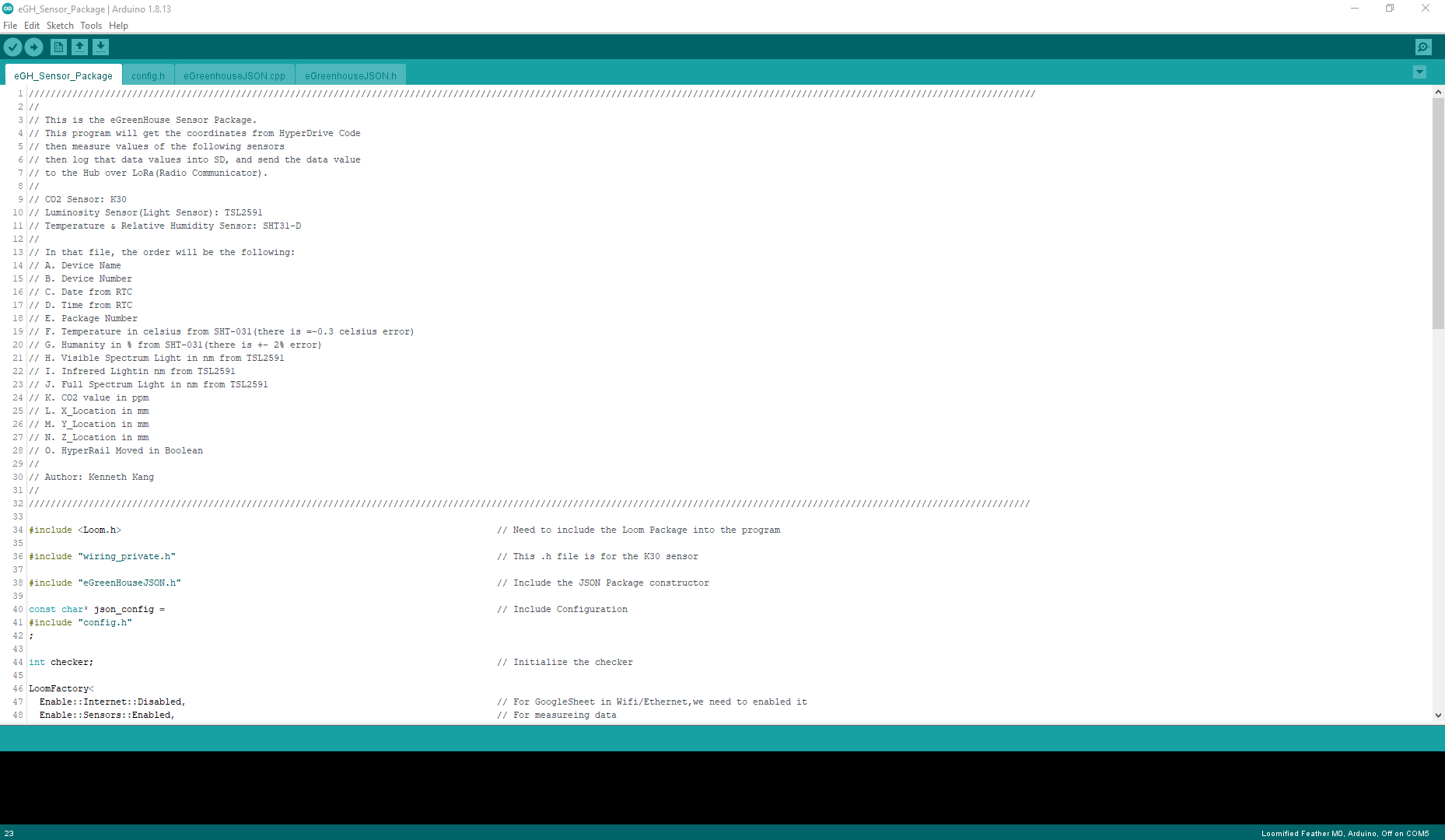
Note: when you open the Arduino IDE, you will get the following message in figure 21. You can simply ignore that.



(Figure 21: Warring Message after installing the Arduino and the Command line, but simply can be ignored)

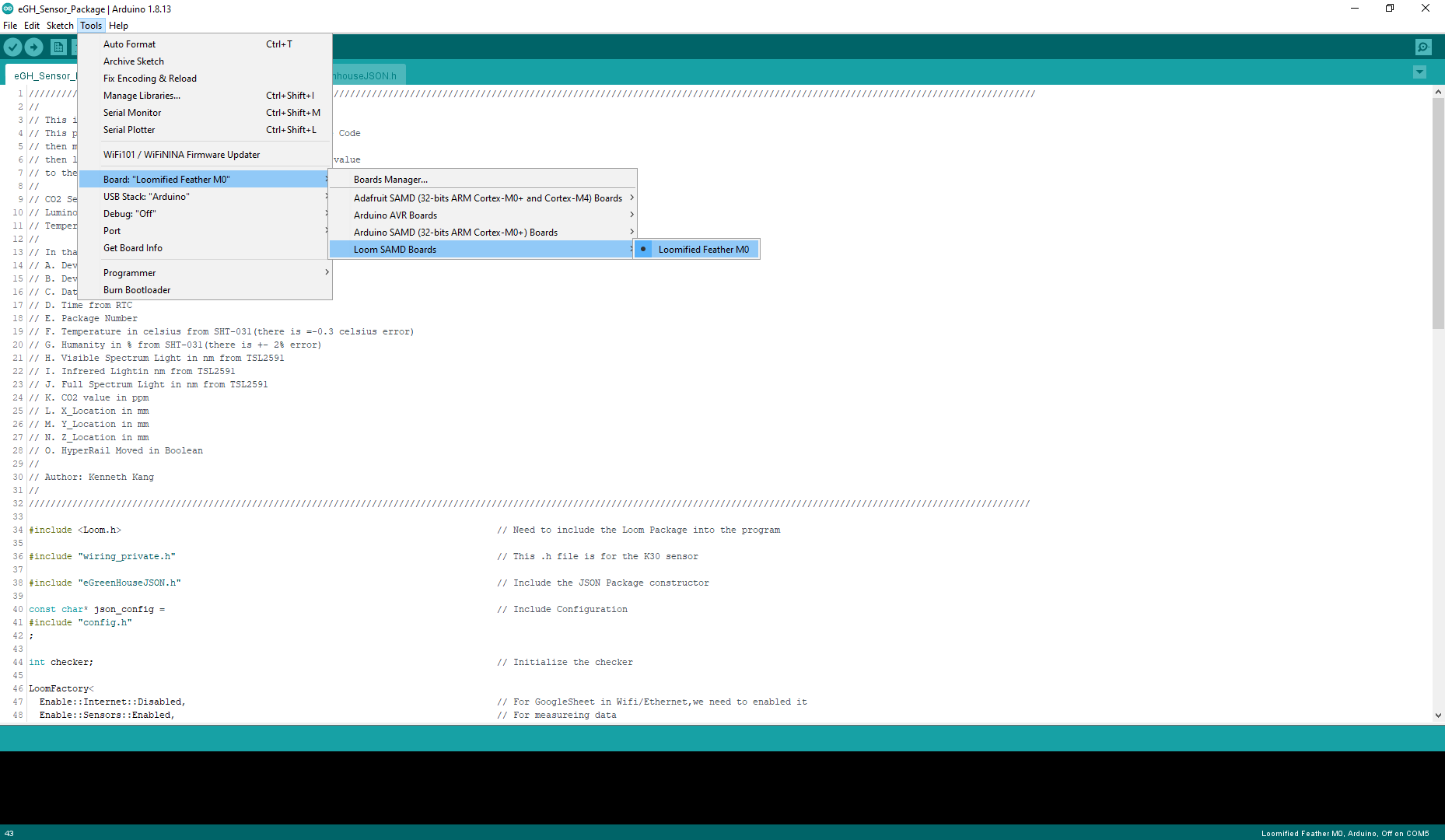
### 6.2. eGreenhouse Sensor Package

In the code file, open eGH\_Sensor\_Package folder. In that folder, only open eGH\_Sensor\_Package.ino. Once you open it, you will get screen in figure 22.



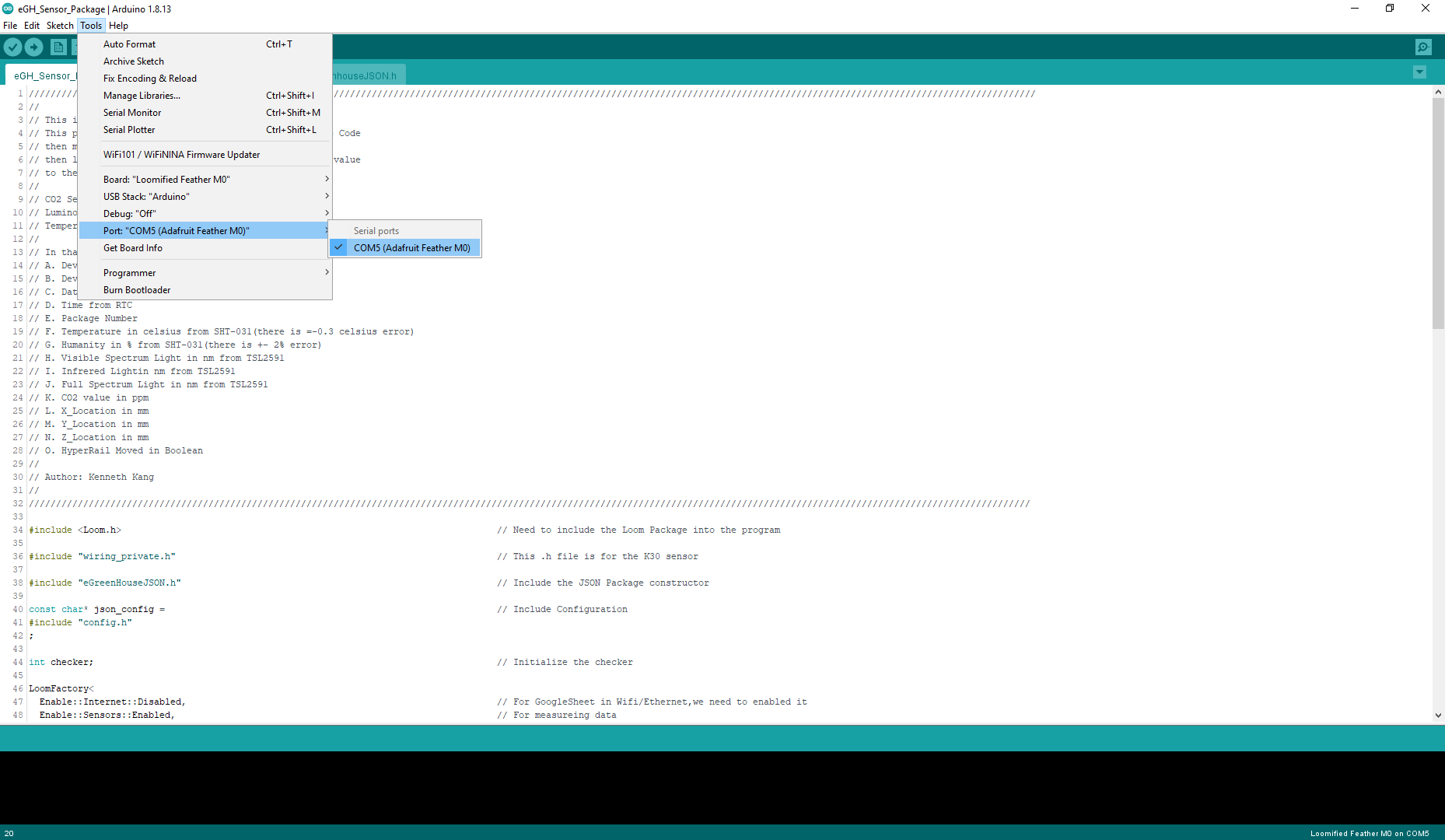
(Figure 22: The eGreenhouse Sensor Package Code Screen from the Arduino IDE)

In that screen, select Tools -> Board -> Loom SAMD Boards -> Loomfied Feather M0 like figure 23.



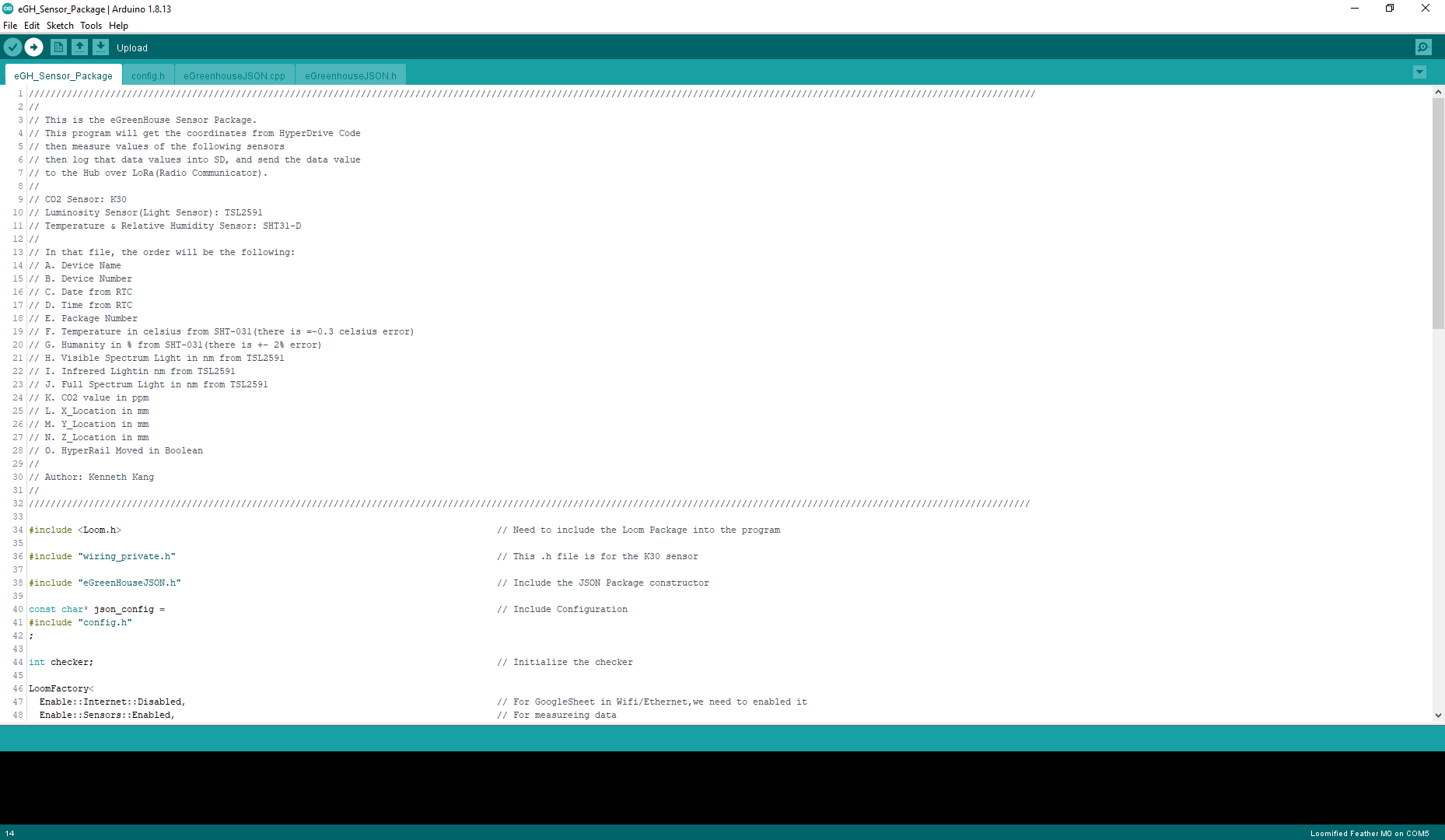
(Figure 23: Selecting the Board in the Arduino IDE)

Once that is complete, then connect the eGreenhouse Sensor Package to the computer with Micro USB Cable. It should recognize the board. Then go to Tools -> Port and see if there an option to select. If there is no option, try either a different cable or reconnect it. It should be some like in Figure 24.



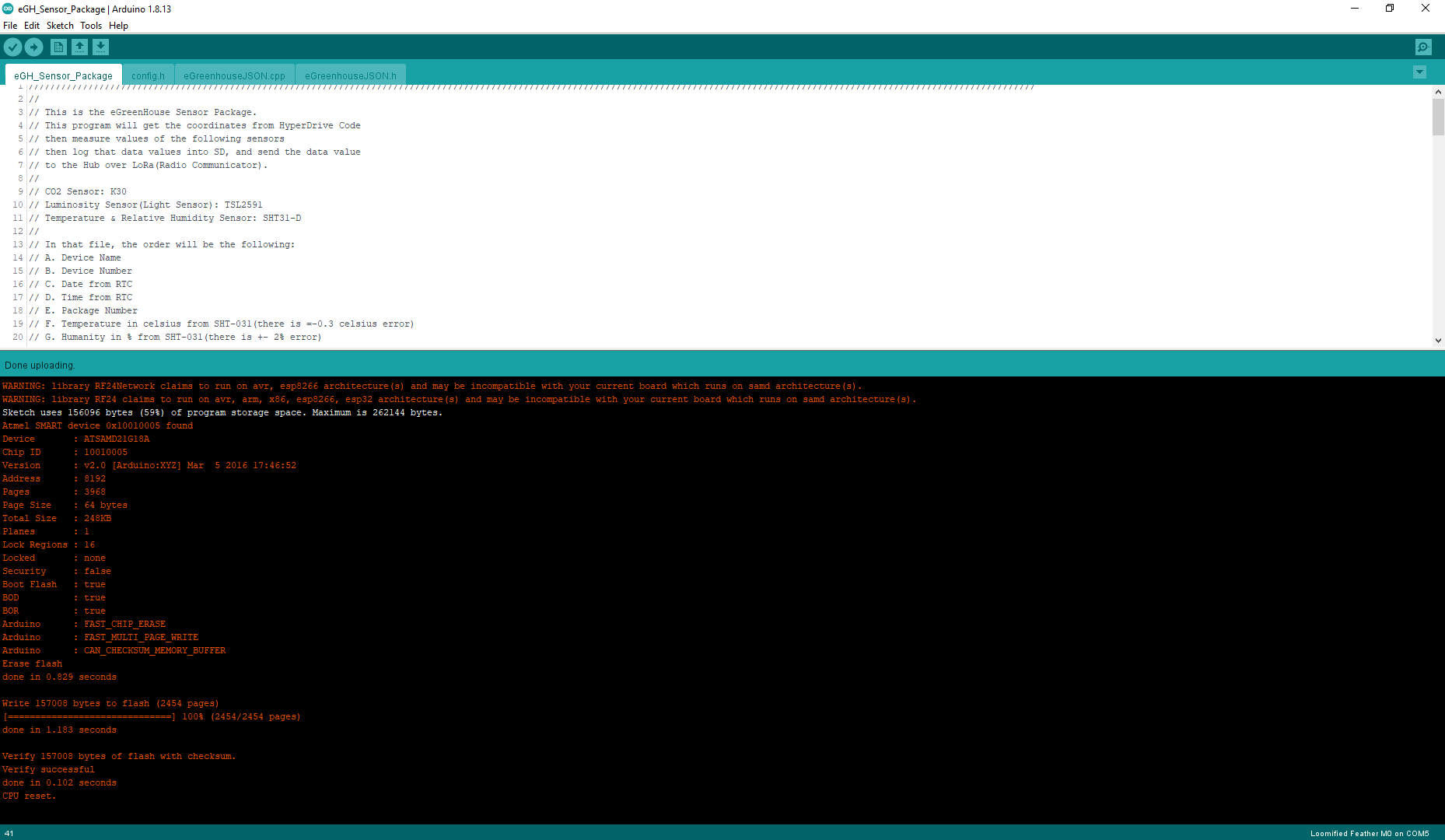
(Figure 24: The proper look when you connect the eGreenhouse Sensor Package. Note that the port number may be different)

Once the computer recognizes the board, then click upload where it looks like an arrow pointing to the right. Look at Figure 25.



(Figure 25: Upload button on the Arduino IDE)

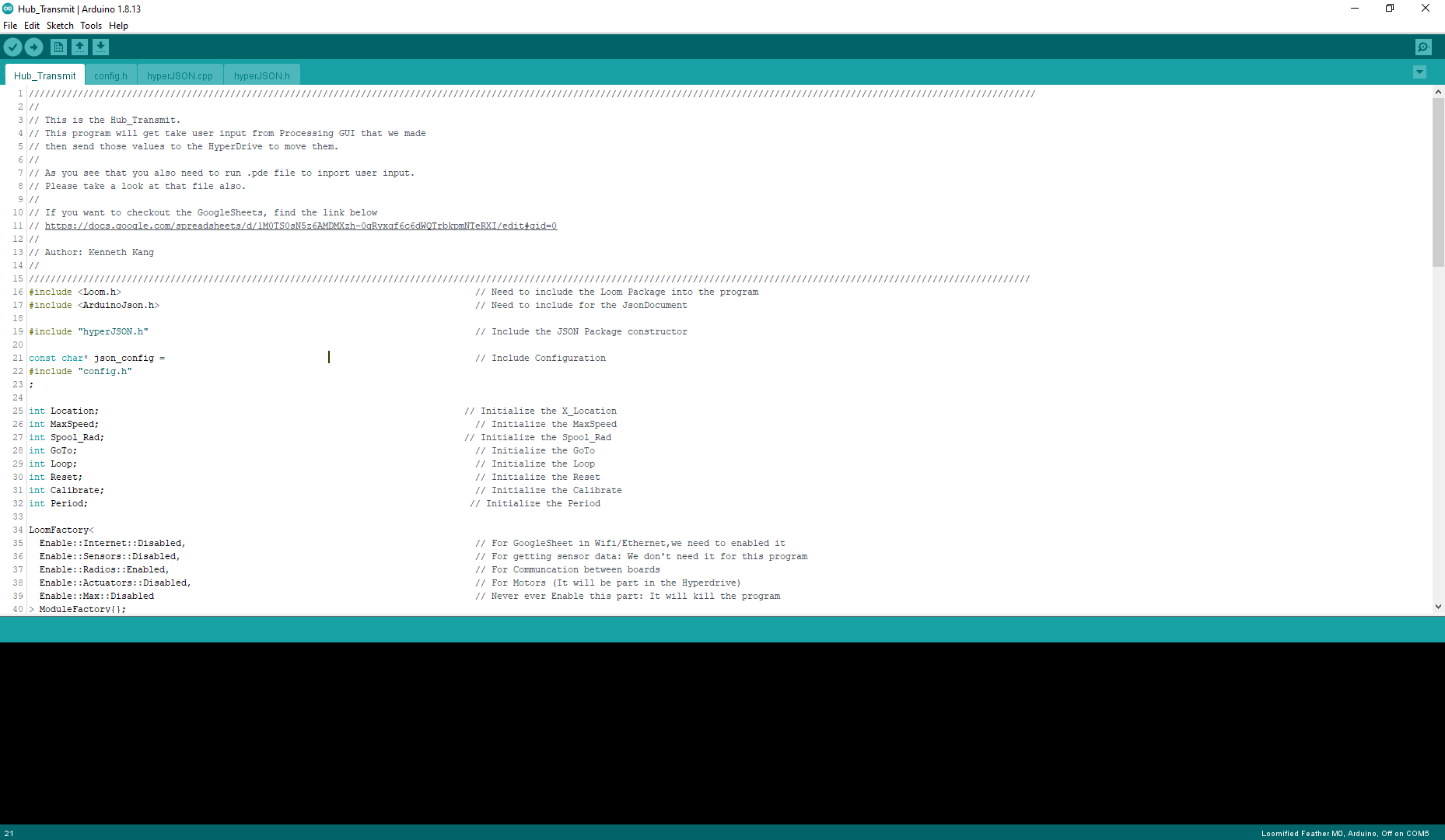
Wait until it gets the message like figure 26. Once you get this message as “CPU Reset”, then it uploaded properly and ready to use.



(Figure 26: The console log that it has upload successfully)

6.3. Hub\_Transmit

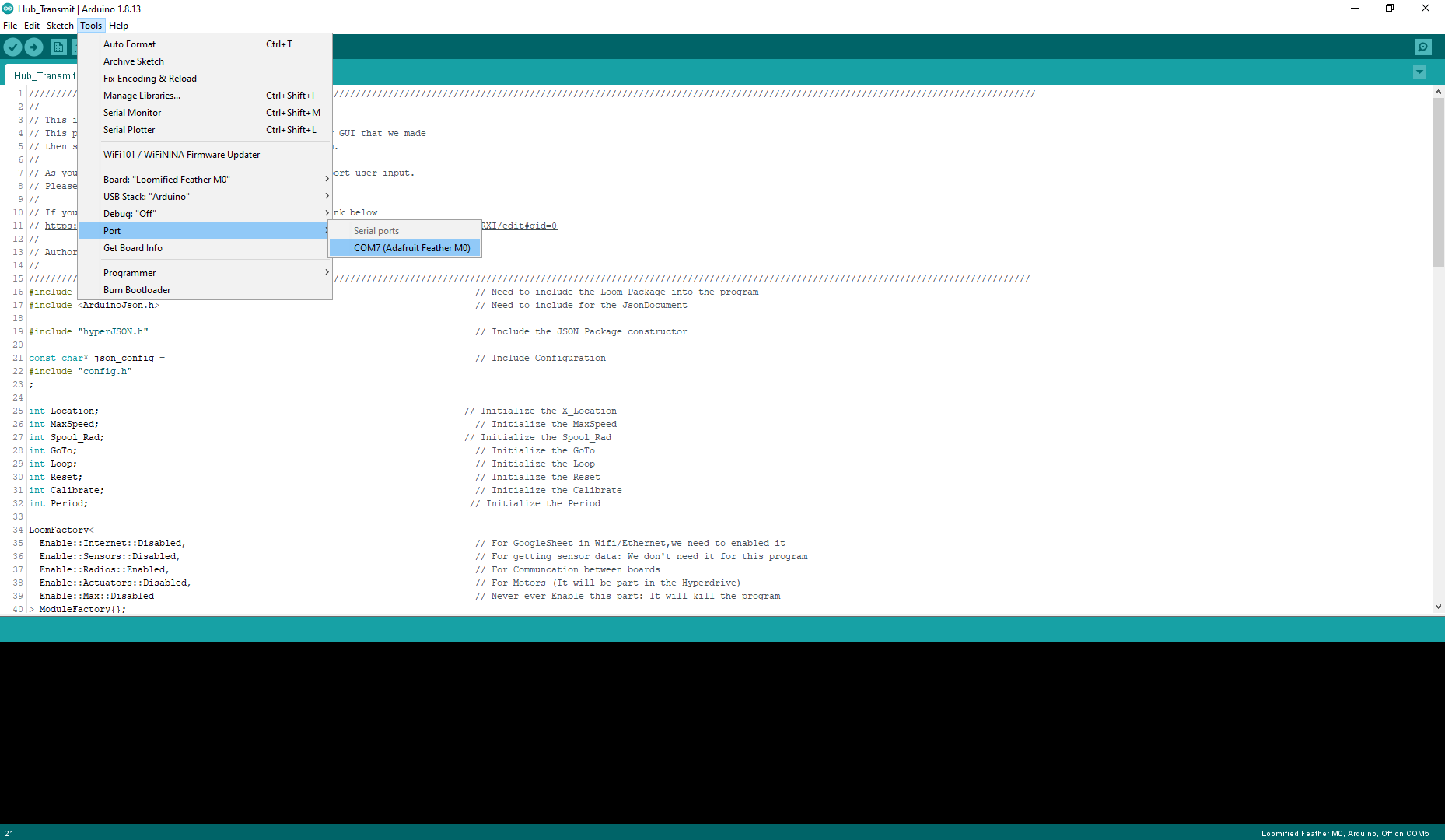
In the code file, open Hub\_Transmit folder. In that folder, only open Hub\_Transmit.ino. Once you open it, you will get screen in figure 27.



(Figure 27: The Hub\_Transmit Code Screen from the Arduino IDE)

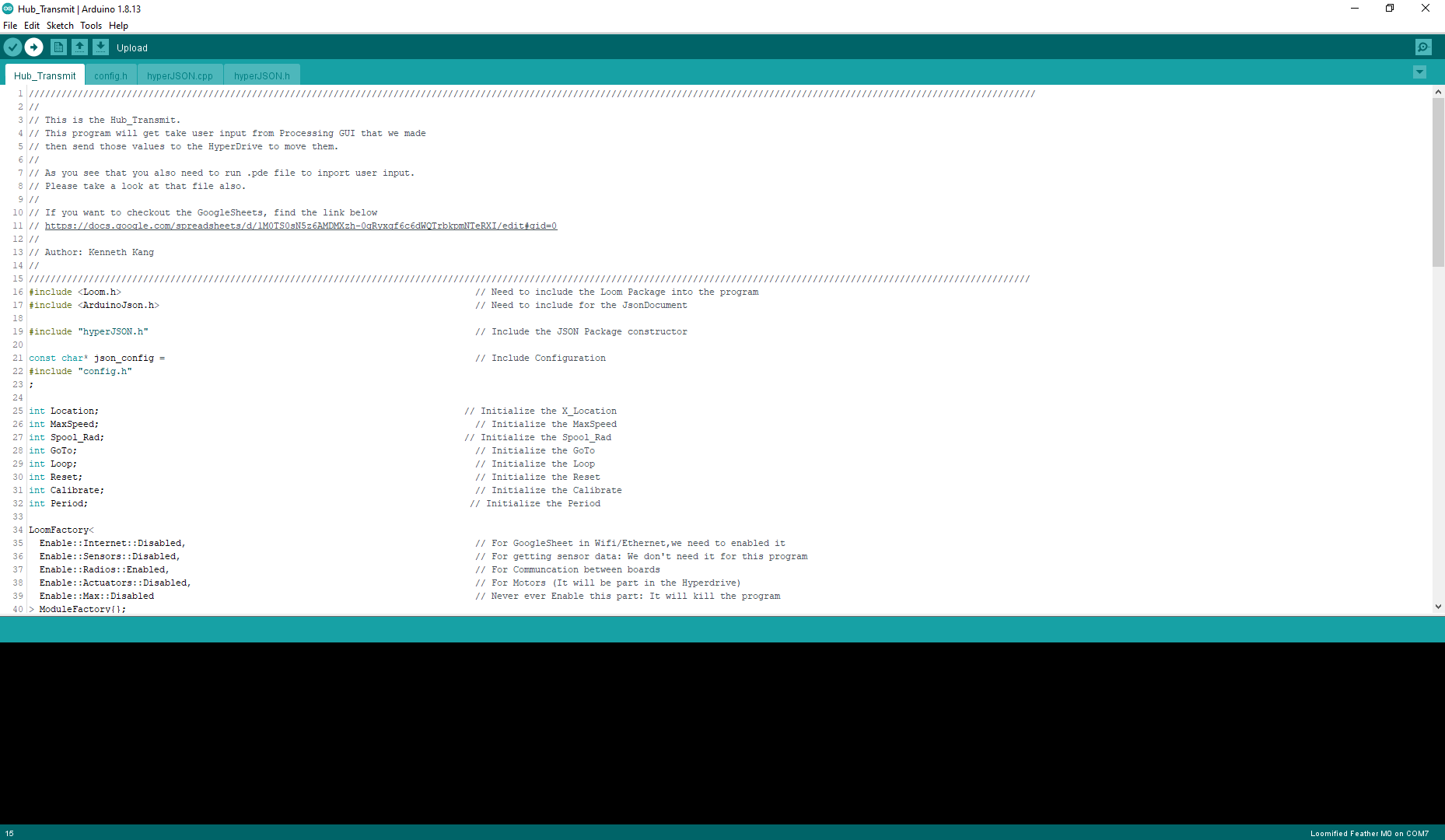
If you have previously selected the board before, it should set as default. If not, then change the board to Loomified Feather M0.

Once that is complete, then connect the Hub Transmit to the computer with Micro USB Cable. It should recognize the board. Then go to Tools -> Port and see if there an option to select. If there is no option, try either a different cable or reconnect it. It should be some like in Figure 28. Also, writing the port number just for the Hub Transmit will be useful for the later.



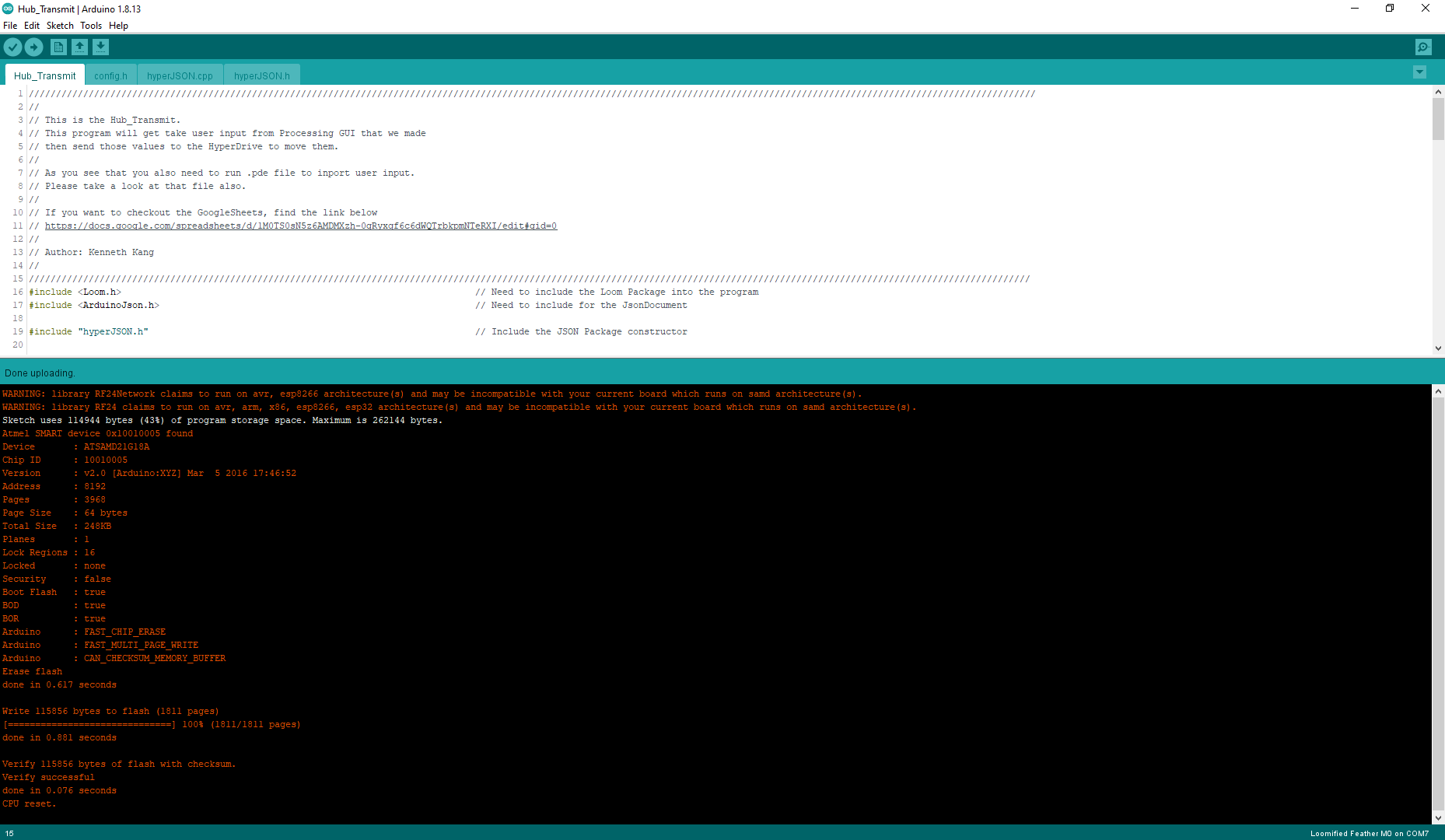
(Figure 28: The proper look when you connect the Hub Transmit. Note that the port number may be different)

Once the computer recognizes the board, then click upload where it looks like an arrow pointing to the right. Look at Figure 29.



(Figure 29: Upload button on the Arduino IDE)

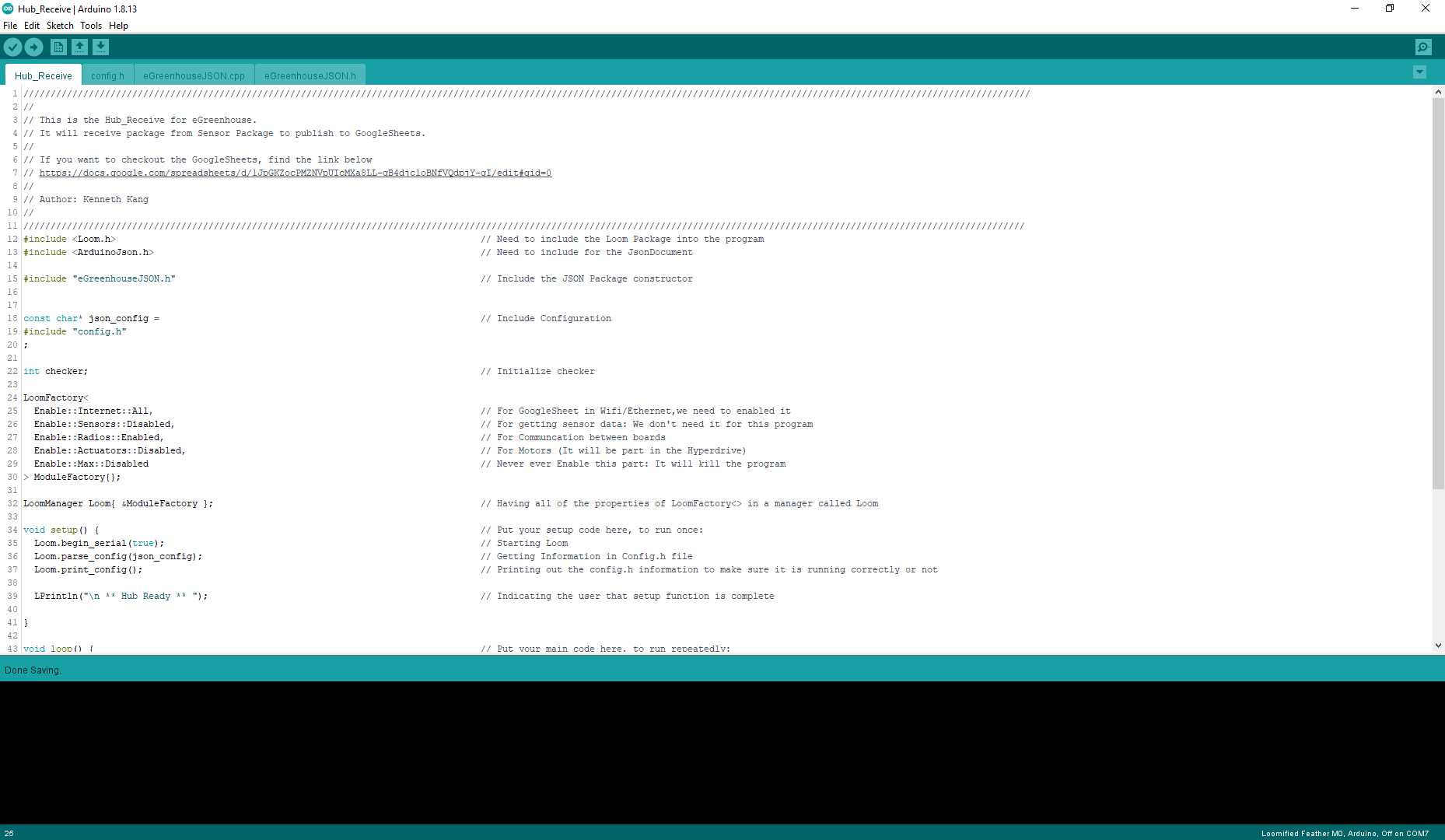
Wait until it gets the message like figure 30. Once you get this message as “CPU Reset”, then it uploaded properly and ready to use.



(Figure 30: The console log that it has upload successfully)

6.4. Hub\_Receive

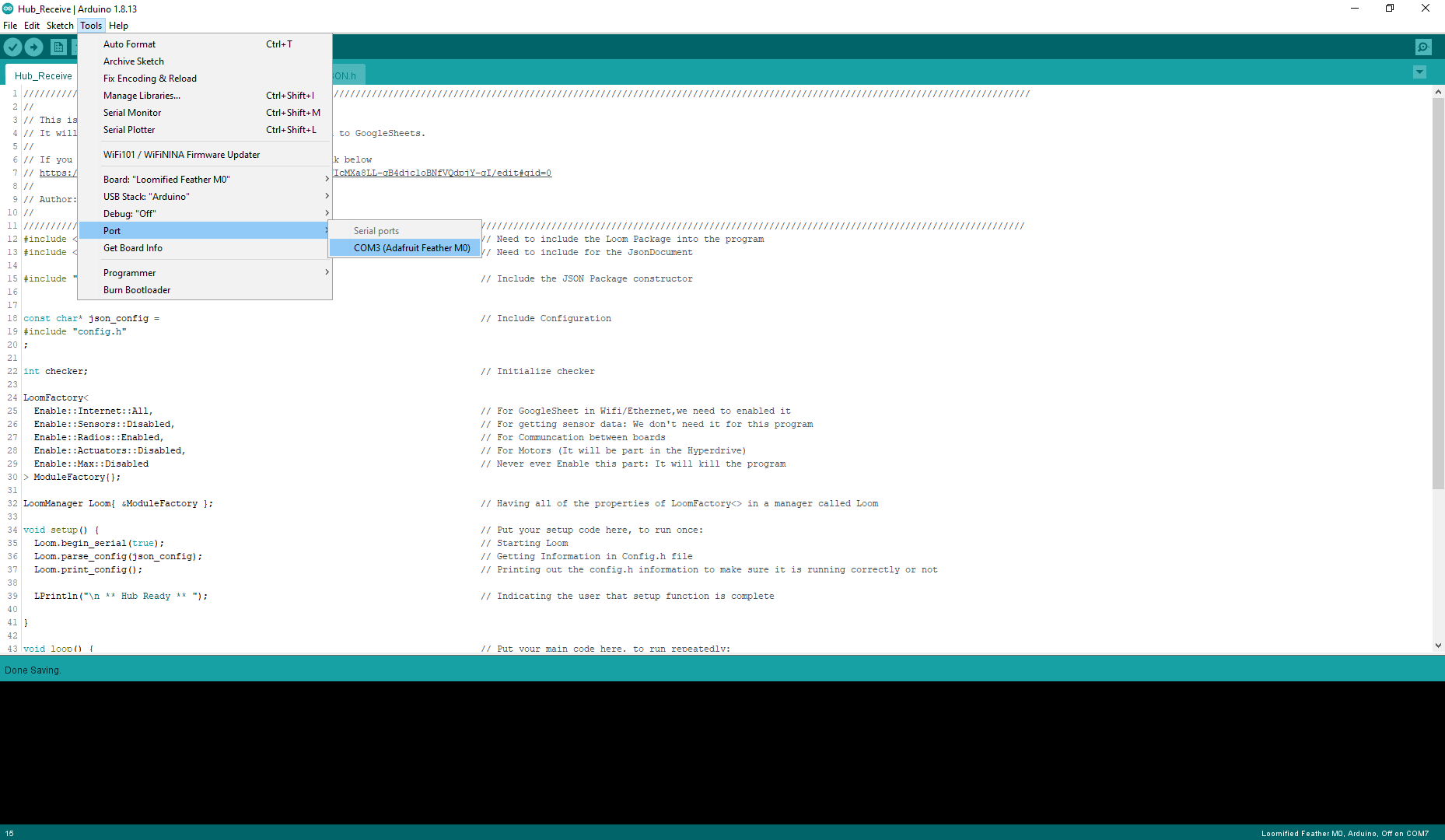
In the code file, open Hub\_Receive folder. In that folder, only open Hub\_Receive.ino. Once you open it, you will get screen in figure 31.



(Figure 31: The Hub\_Receive Code Screen from the Arduino IDE)

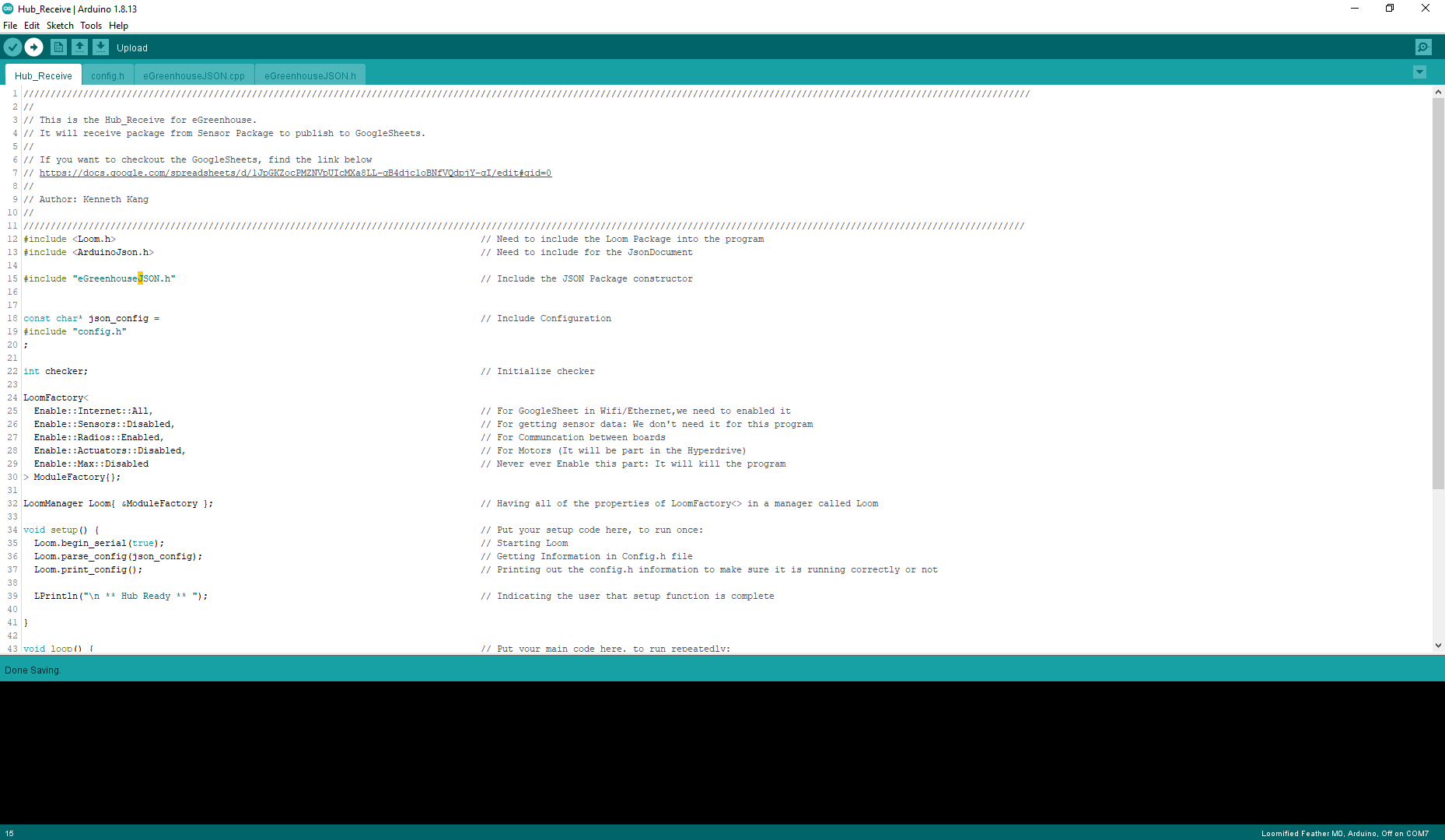
If you have previously selected the board before, it should set as default. If not, then change the board to Loomified Feather M0.

Once that is complete, then connect the Hub Receive to the computer with Micro USB Cable. It should recognize the board. Then go to Tools -> Port and see if there an option to select. If there is no option, try either a different cable or reconnect it. It should be some like in Figure 32.



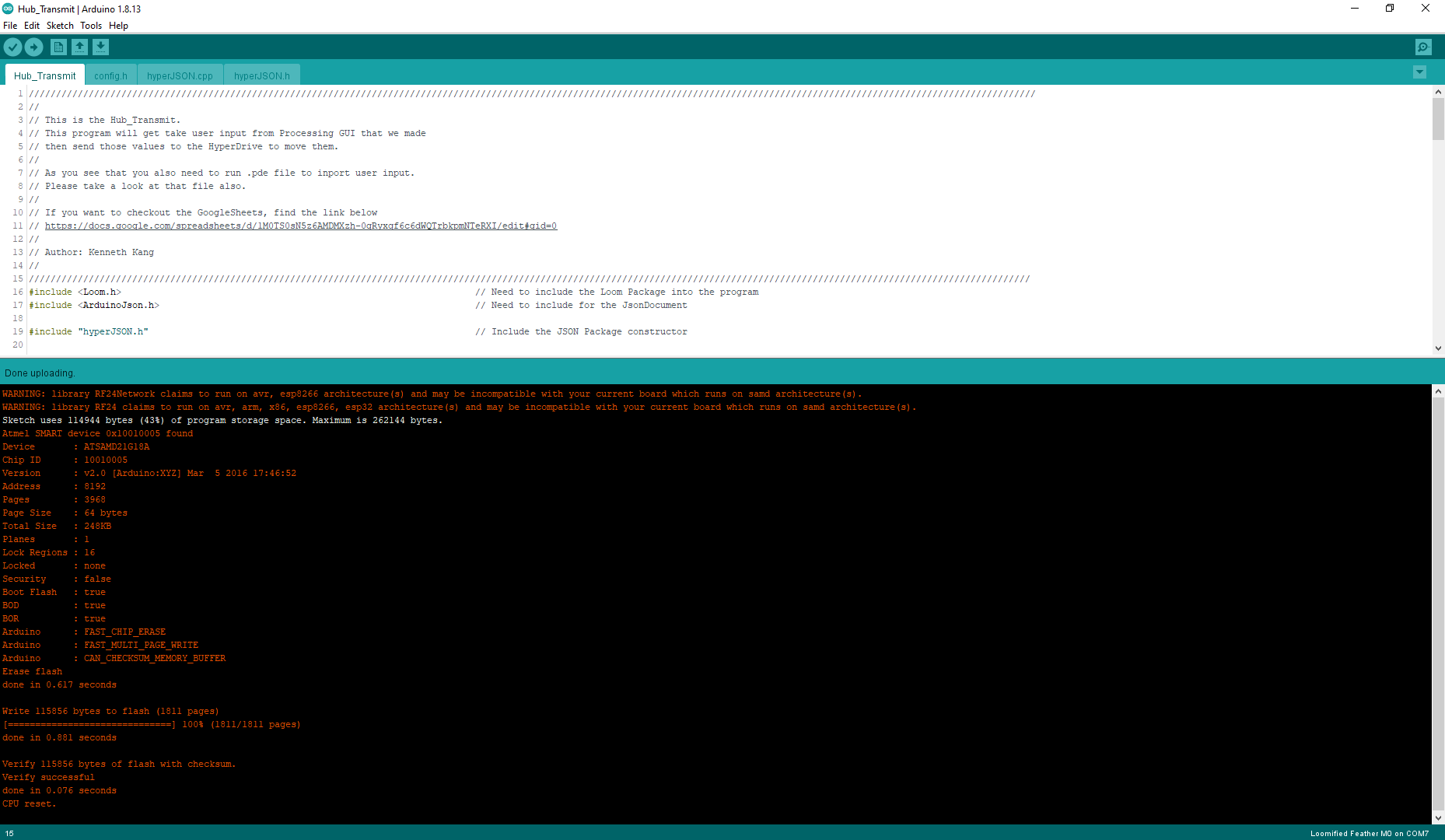
(Figure 32: The proper look when you connect the Hub Receive. Note that the port number may be different)

Once the computer recognizes the board, then click upload where it looks like an arrow pointing to the right. Look at Figure 33.



(Figure 33: Upload button on the Arduino IDE)

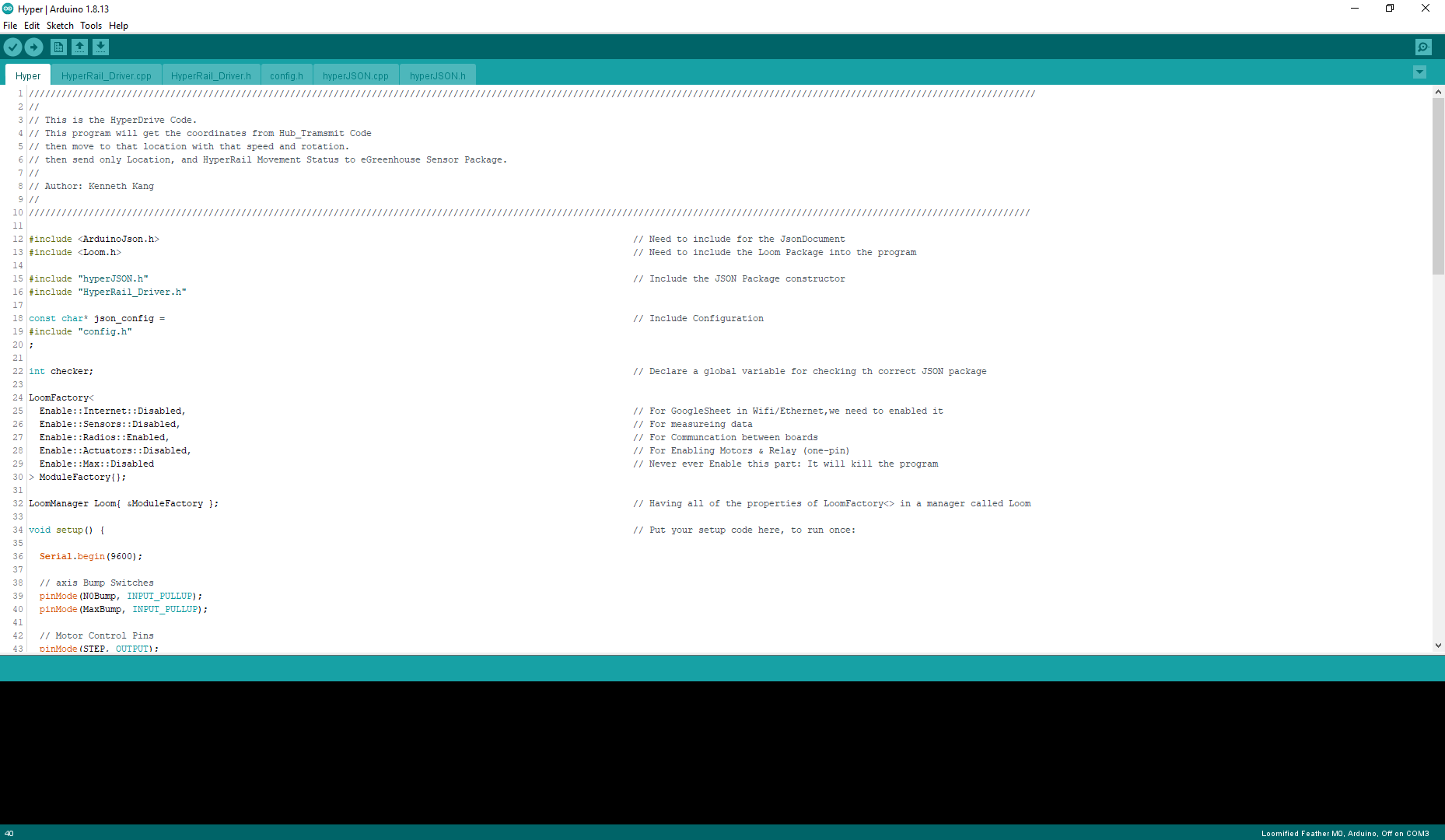
Wait until it gets the message like figure 34. Once you get this message as “CPU Reset”, then it uploaded properly and ready to use.



(Figure 34: The console log that it has upload successfully)

6.5. HyperRail

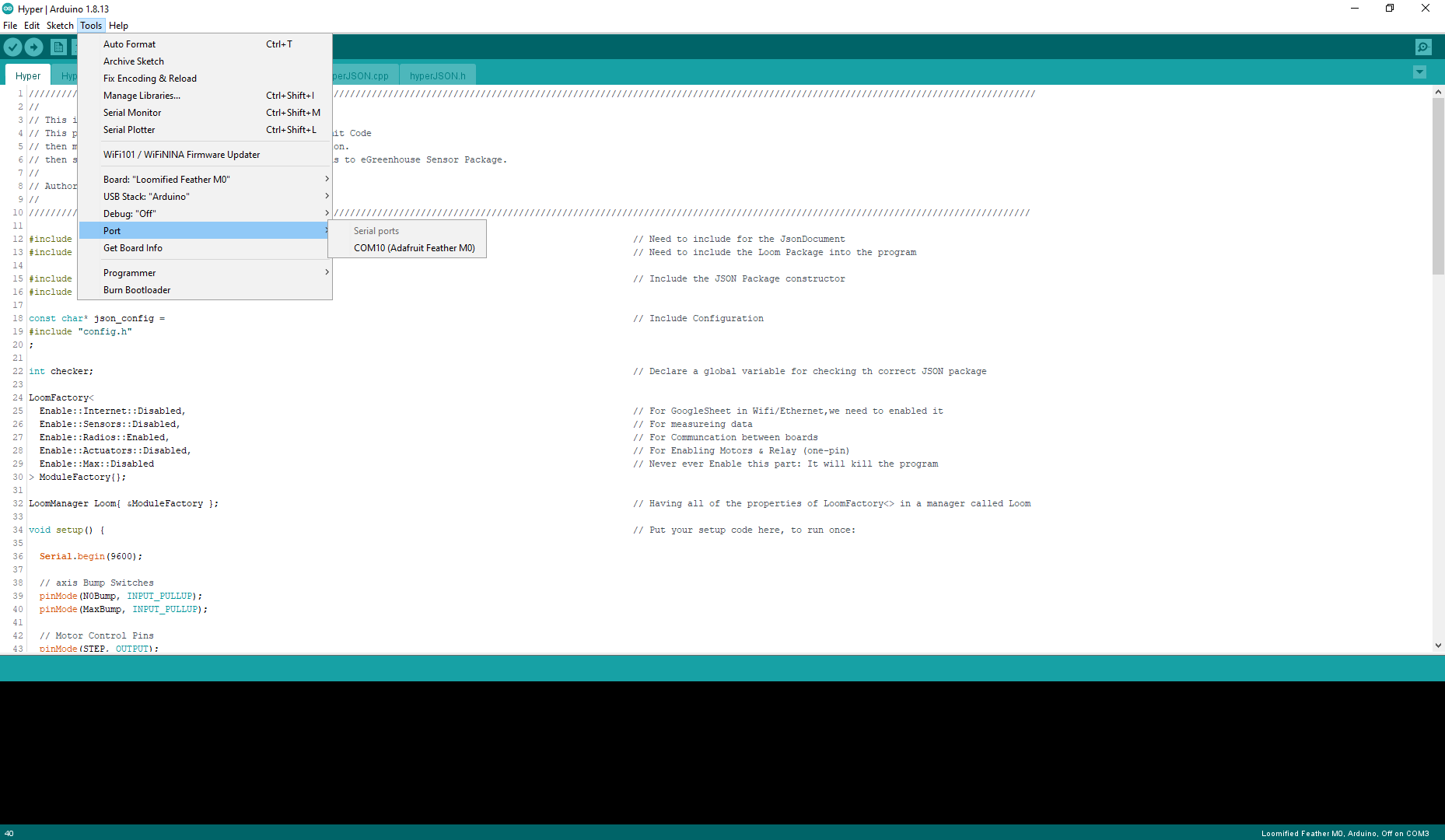
In the code file, open Hyper folder. In that folder, only open Hyper.ino. Once you open it, you will get screen in figure 35.



(Figure 35: The Hyper Code Screen from the Arduino IDE)

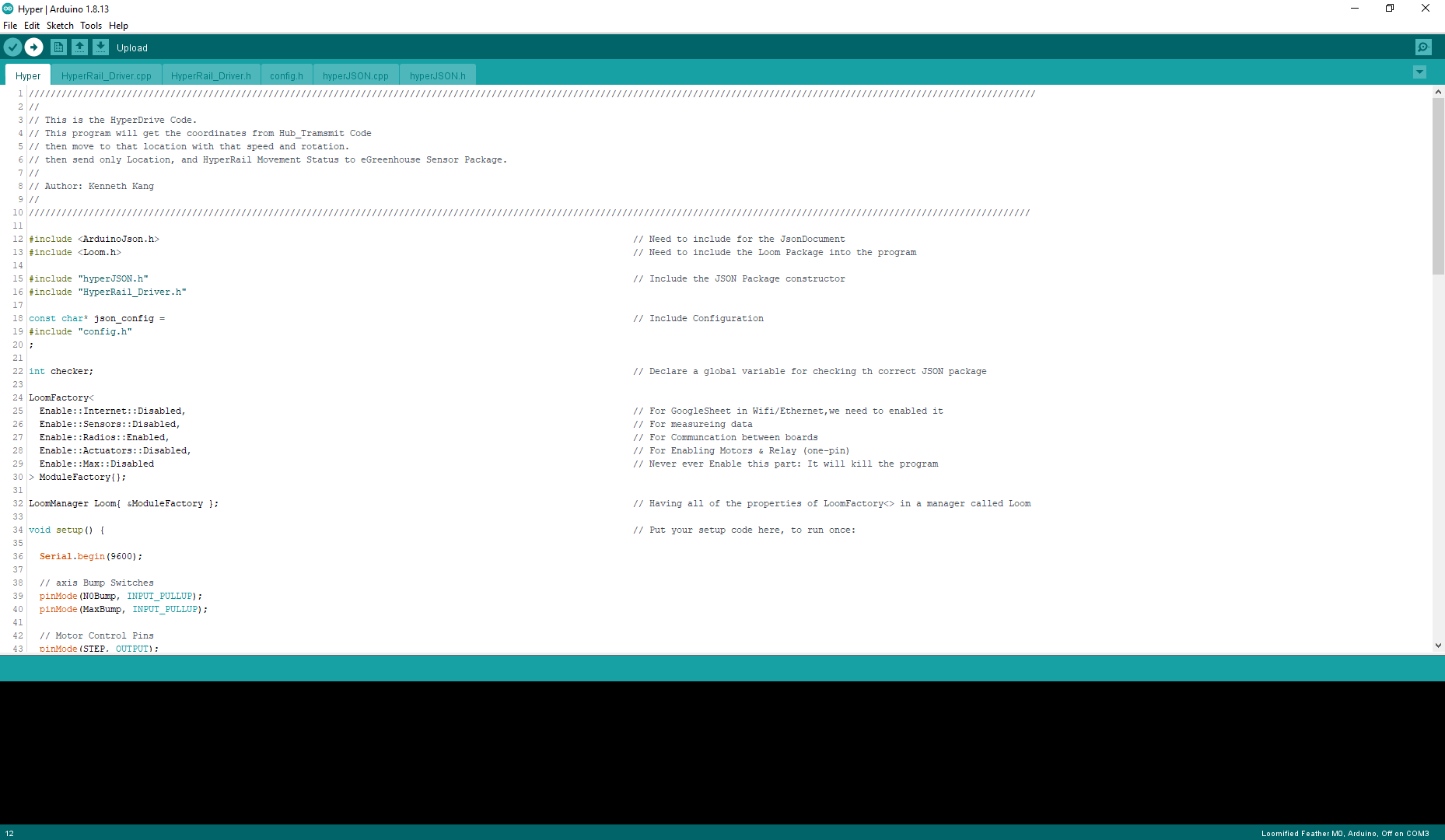
If you have previously selected the board before, it should set as default. If not, then change the board to Loomified Feather M0.

Once that is complete, then connect the Hub Transmit to the computer with Micro USB Cable. It should recognize the board. Then go to Tools -> Port and see if there an option to select. If there is no option, try either a different cable or reconnect it. It should be some like in Figure 28.



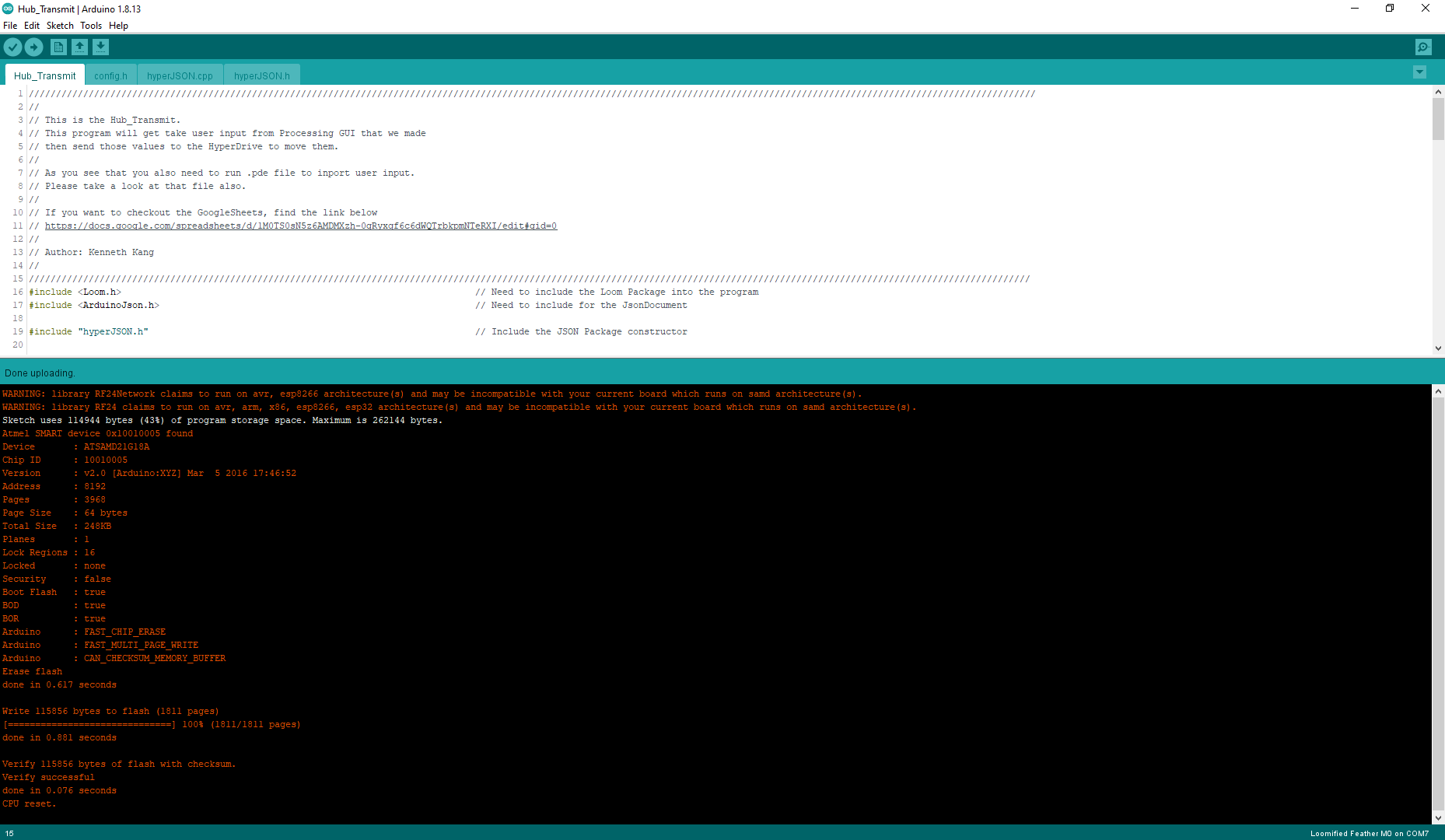
(Figure 36: The proper look when you connect the Hub Transmit. Note that the port number may be different)

Once the computer recognizes the board, then click upload where it looks like an arrow pointing to the right. Look at Figure 37.



(Figure 37: Upload button on the Arduino IDE)

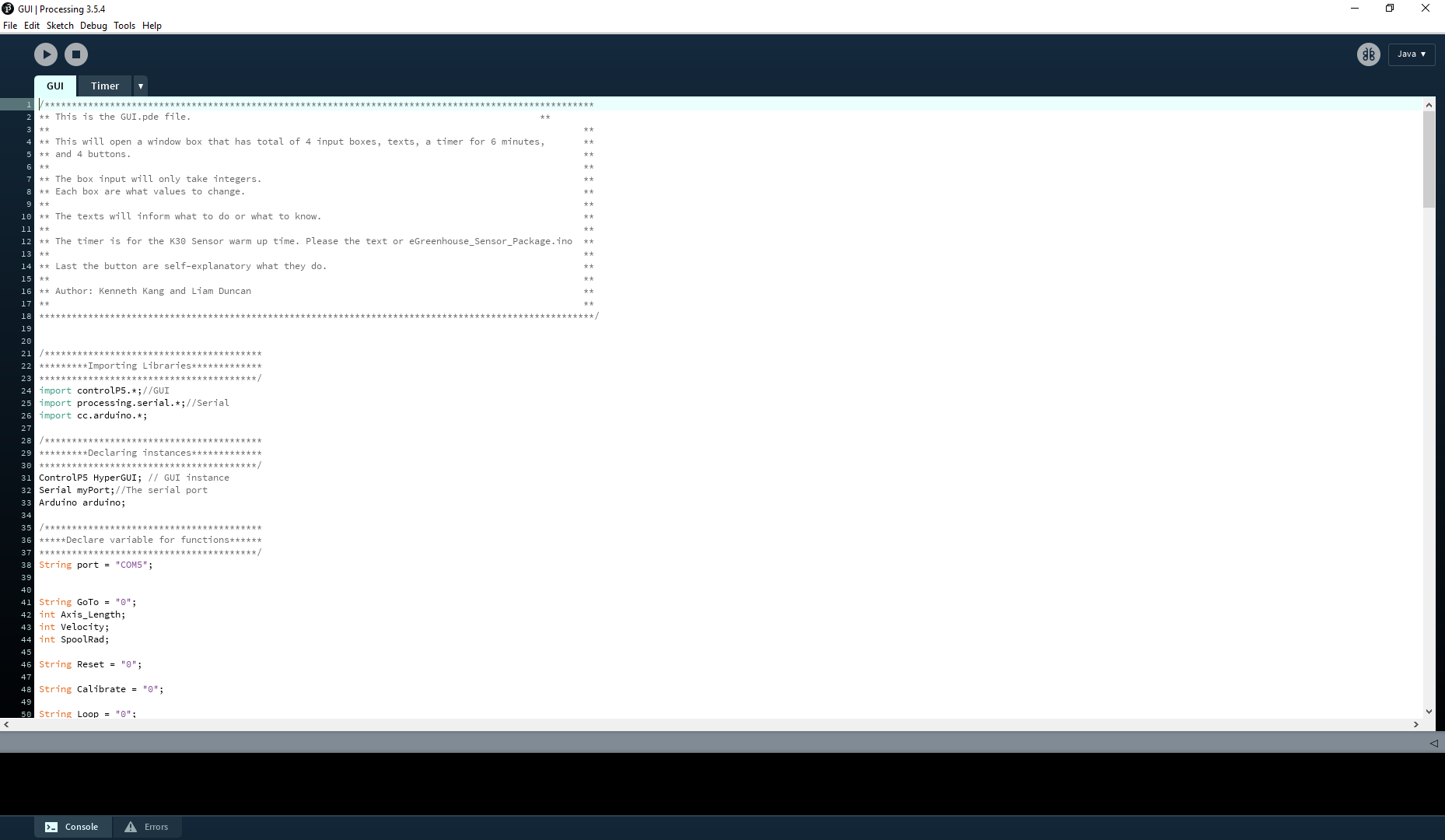
Wait until it gets the message like figure 38. Once you get this message as “CPU Reset”, then it uploaded properly and ready to use.



(Figure 38: The console log that it has upload successfully)

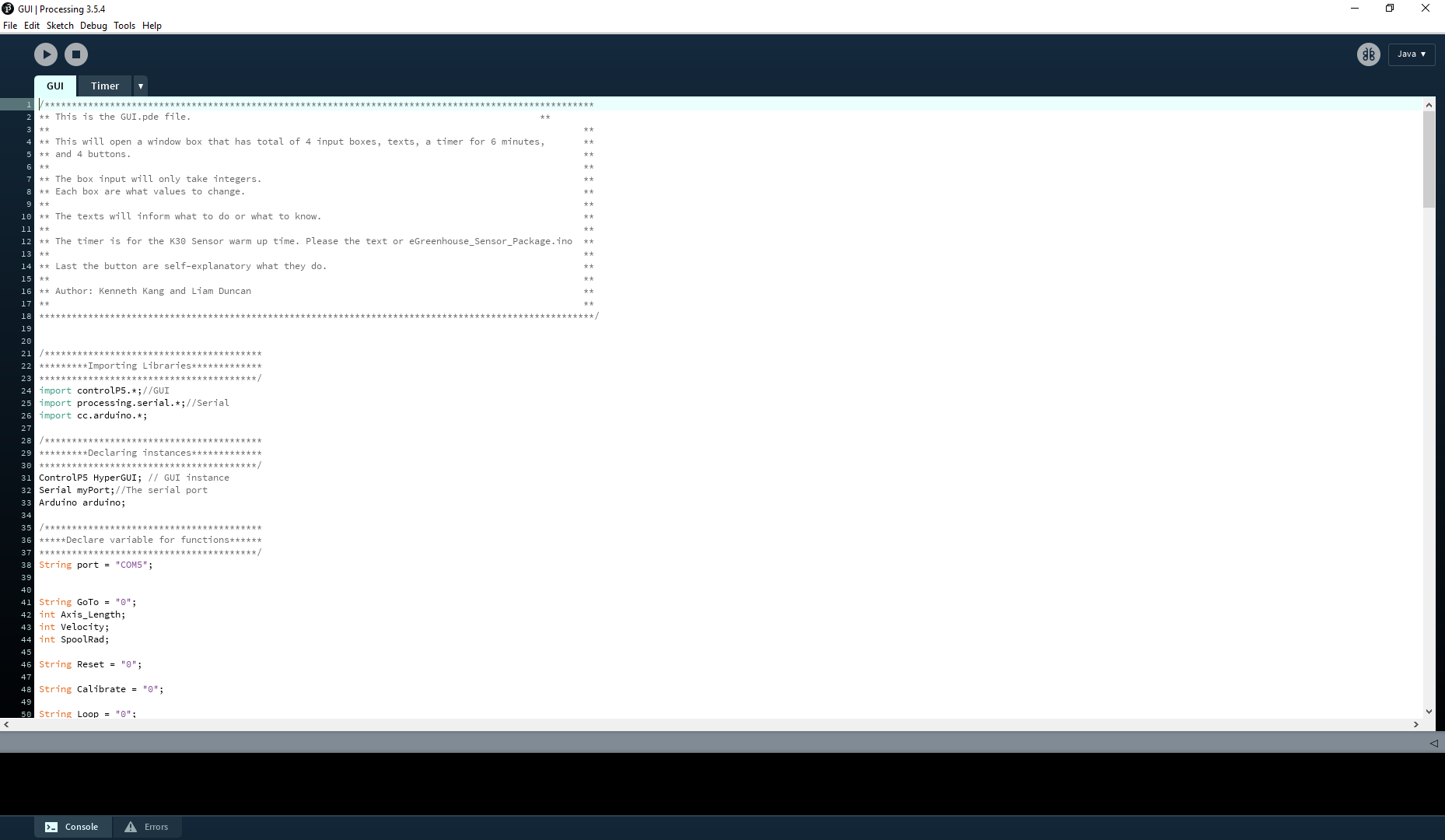
6.6. Loading firmware on microcontroller

In the design file, open the GUI folder. Inside that folder, only open the GUI.pde file. Once you open it, you will get a screen in figure 39.



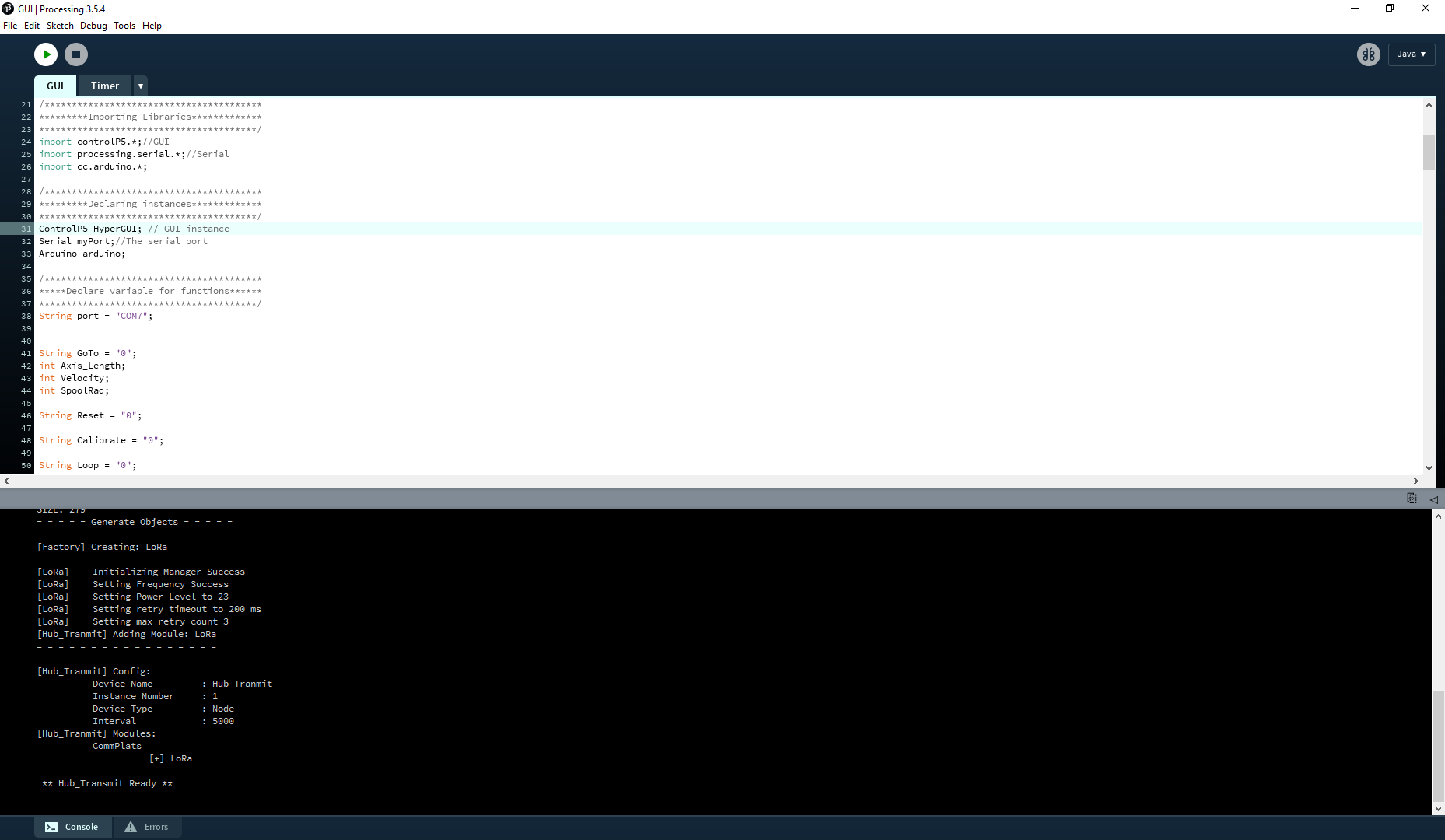
(Figure 39: Processing Application for the GUI)

In the code, change the port number for the Hub Transmit Board where it says, “String port = “COM5””. You just need to change the number. Look at Figure 40 for where it is located.



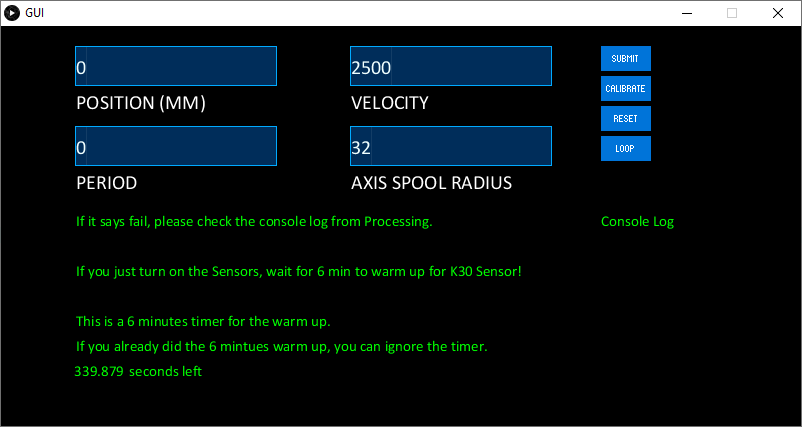
(Figure 40: The location of the port number where to change)

Once that is set, connect the Hub Transmit board to the computer and click the green arrow button on the top left. Reference in Figure 41.



(Figure 41: The location to run the Processing Application)

Then you will get this Application look like Figure 42.



(Figure 42: The look of the GUI)

In the application, you can move the position of the HyperRail, the velocity, the radius, and the period of number of loops. All the text inputs only take integers. As you see, there is a timer for the CO2 sensor warm up time. This timer exists that the CO2 sensor needs at least 6 minutes warm up time to get accurate measurements. If you know the eGreenhouse Sensor Package was running more than 6 minutes, then you can simply ignore it. On the bottom right, you can see the console log. If the user selects one of the four buttons, it will print out if it sends to the HyperRail board or not. If not, it will be said failed. Once you are getting this, you are set to control the HyperRail and measure the eGreenhouse Sensor Package if both boards are on.

6.7. Publishing to GoogleSheets

This [link](https://docs.google.com/spreadsheets/d/1JpGKZocPMZNVpUIcMXa8LL-qB4djc1oBNfVQdpjY-qI/edit#gid=0) will lead to the GoogleSheets that will be uploading data if the Hub\_Receive is connected to an Ethernet Cable. Make sure to power on the Hub\_Receive and connect the Ethernet cable to the Ethernet port on the board and the port that is active. Once that is setup, then it will automatically publish data to GoogleSheets once it gets values from the eGreenhouse Sensor Package.