Instructions

Uploading the Arduino Code

@ NYC Resistor

Step 1: **Download Arduino IDE** Version2.3.2 here:

https://www.arduino.cc/en/software

Arduino IDE 2.3.2

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the **Arduino IDE 2.0 documentation**.

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on **GitHub**.

DOWNLOAD OPTIONS

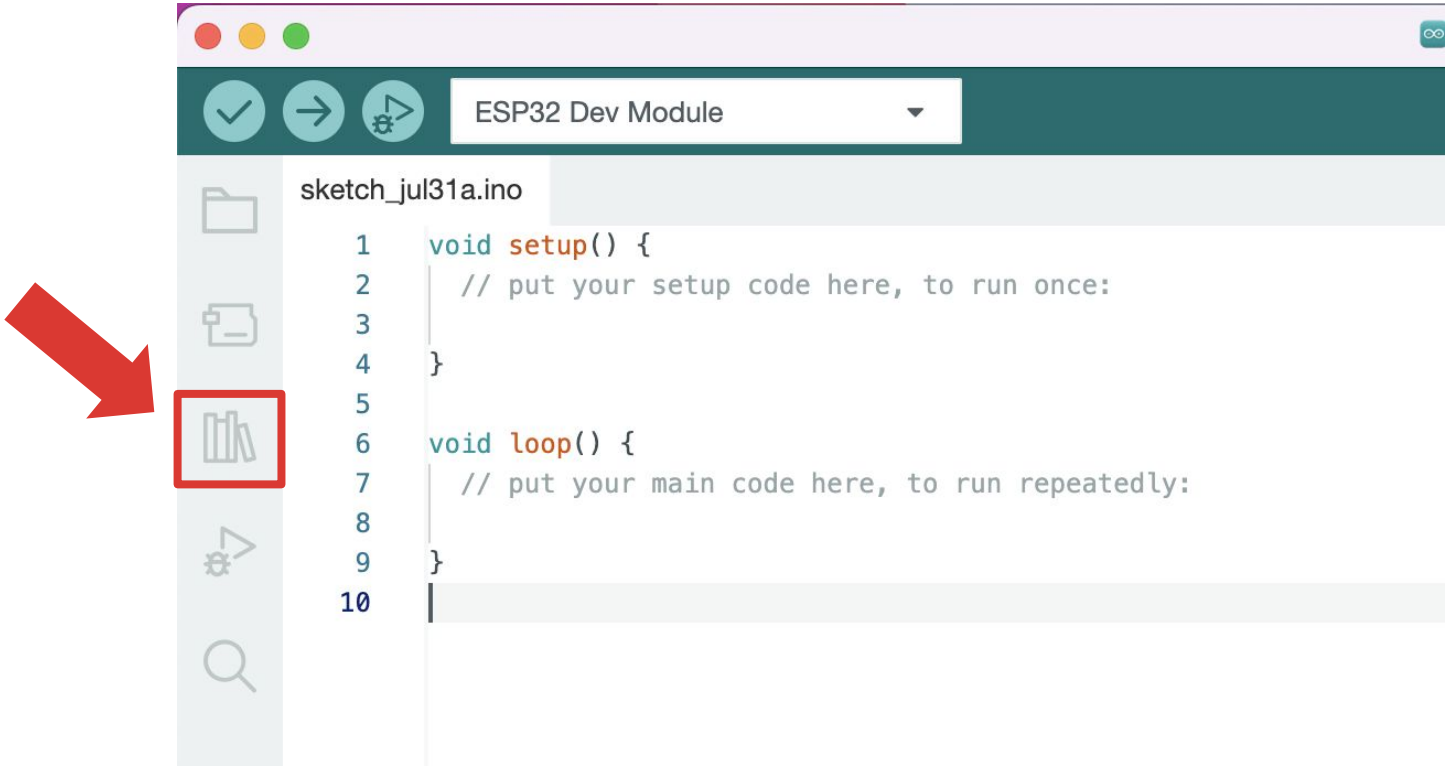
Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.15: "Catalina" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

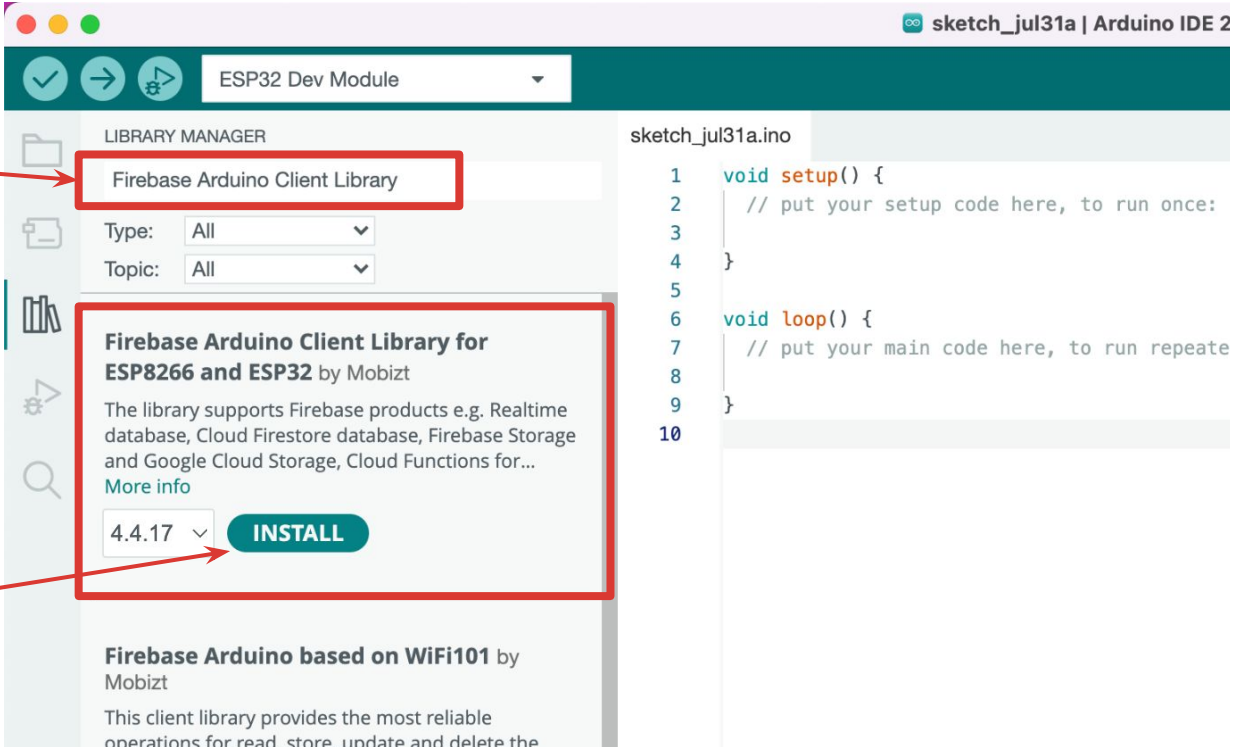
Release Notes

Step 2: Open the Arduino IDE. Click on the "**Libraries**" icon.

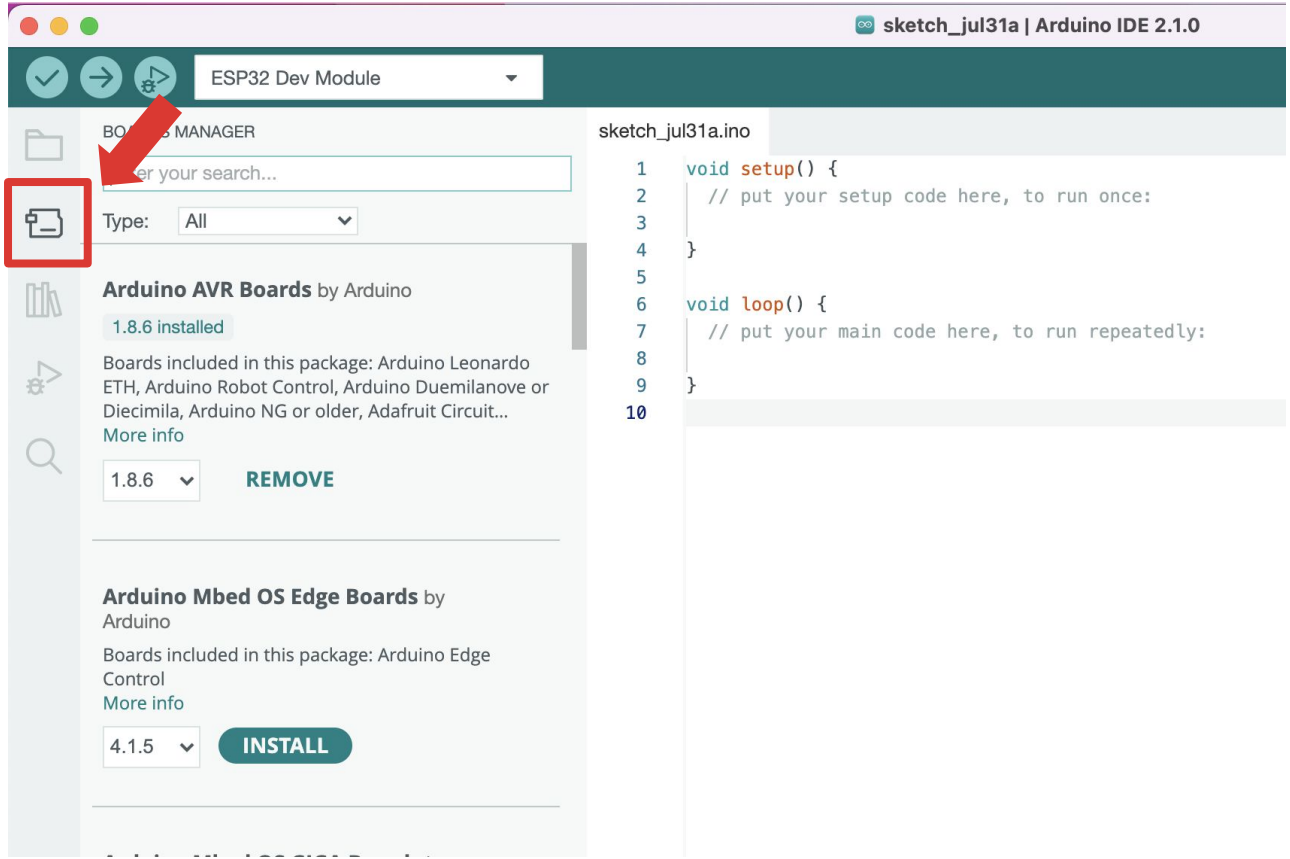


Step 3: In the search bar, type in: "**Firebase Arduino Client Library**"

Step 4: **Install** "Firebase Arduino Client Library for ESP8266 and ESP32" by Mobizt

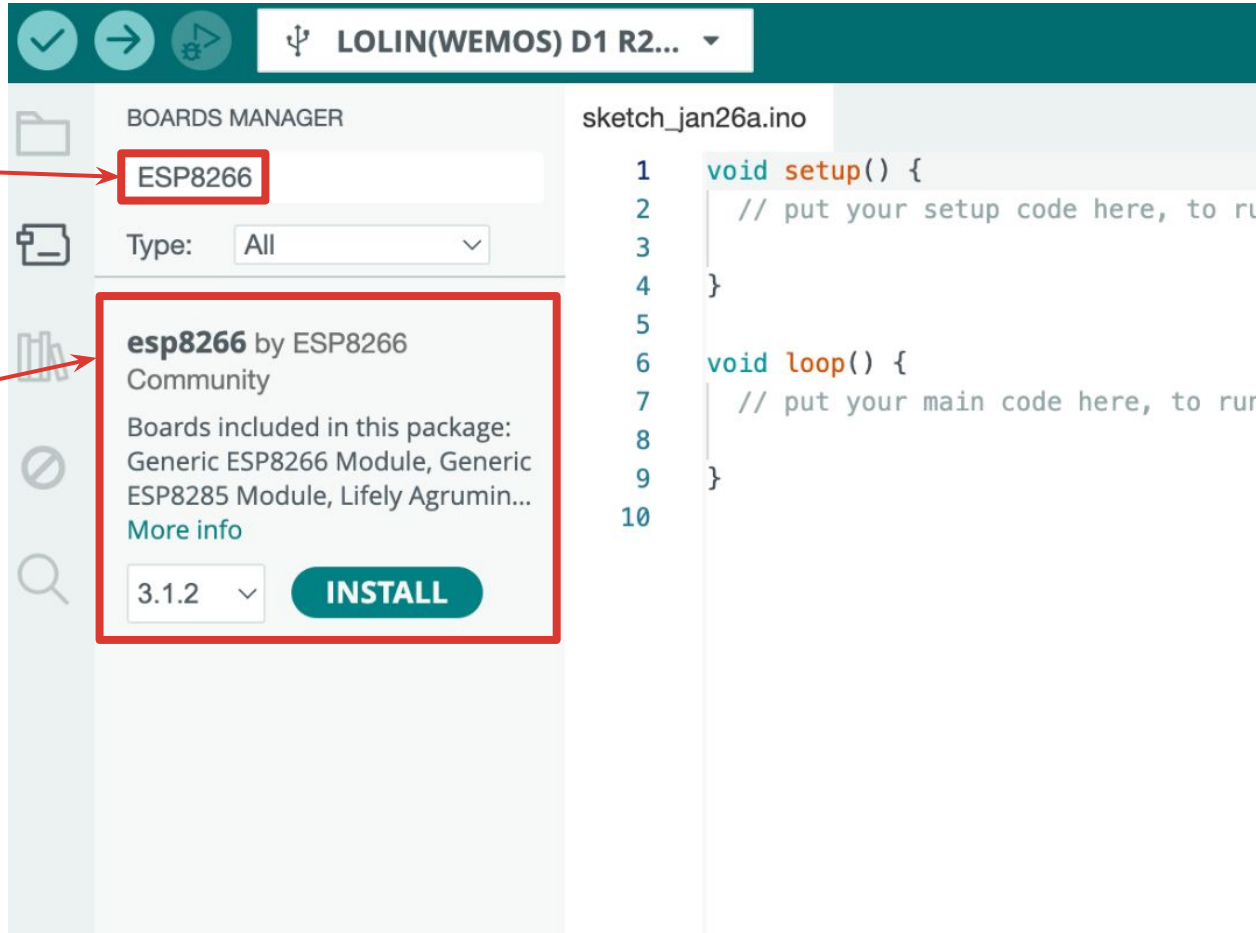


Step 5: Next, click on the Boards Manager icon.

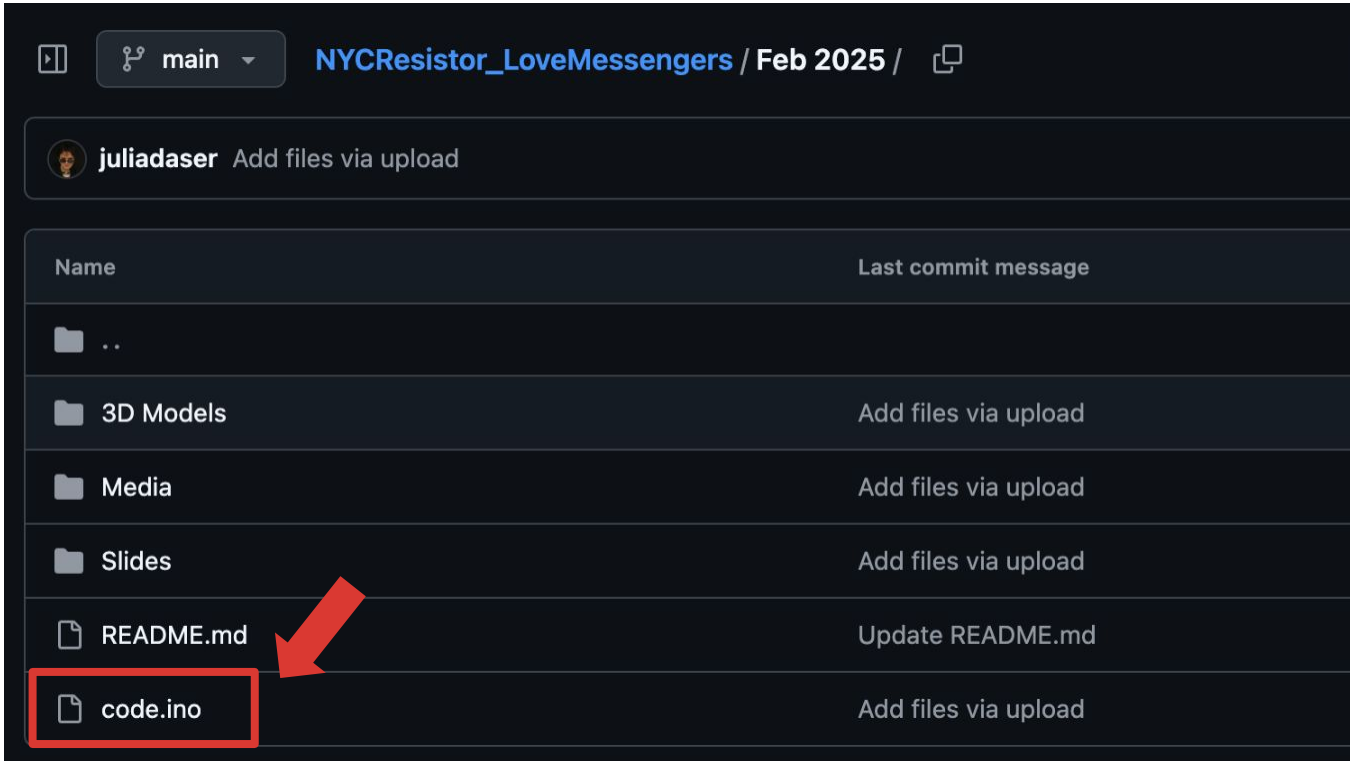


Step 6: In the search bar, search "**ESP8266**"

Step 7: Install "esp8266" library by ESP8266 Community. This could take some time.



Step 8: Head to our GitHub repository: https://github.com/pepzicles/NYCResistor_LoveMessengers/tree/main/Feb%202025 and go into the "**Code.ino**" file.

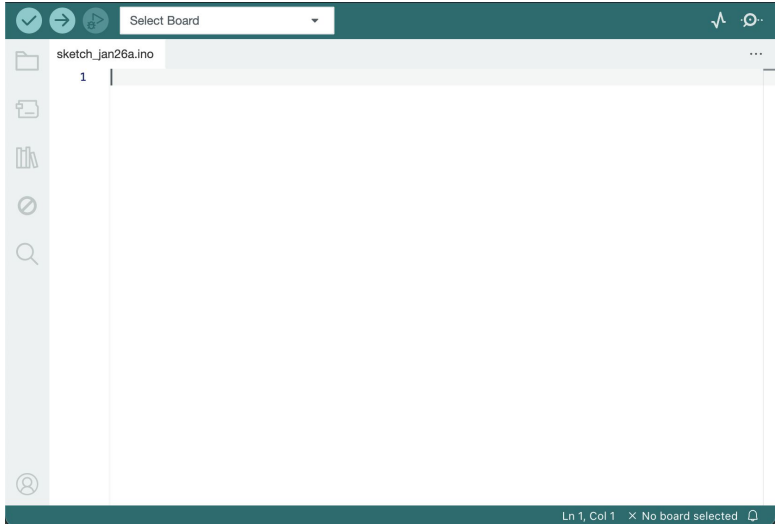


Step 10: Hit this button to copy all the code.

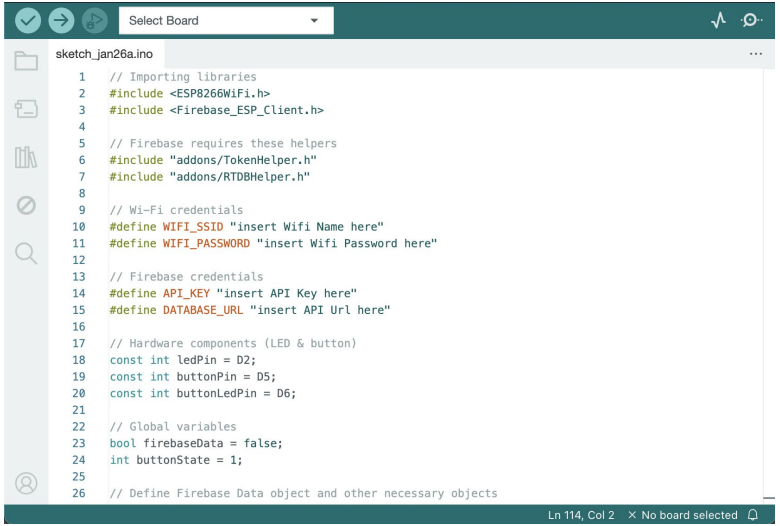
Code Blame Raw

```
// Importing libraries
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>

// Firebase requires these helpers
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// Wi-Fi credentials
#define WIFI_SSID "insert Wifi Name here"
#define WIFI_PASSWORD "insert Wifi Password here"

// Firebase credentials
#define API_KEY "insert API Key here"
#define DATABASE_URL "insert API Url here"

// Hardware components (LED & button)
const int ledPin = D2;
const int buttonPin = D5;
const int buttonLedPin = D6;

// Global variables
bool firebaseData = false;
int buttonState = 1;

// Define Firebase Data object and other necessary objects
FirebaseData fbdo;
FirebaseAuth auth;
```

Step 11: Delete all the code that is currently in your Arduino IDE.



Step 12: Paste all the code you have copied from the GitHub repository into here.



Step 12: Insert your home WIFI Name and Password.

Select Board

sketch_jan26a.ino

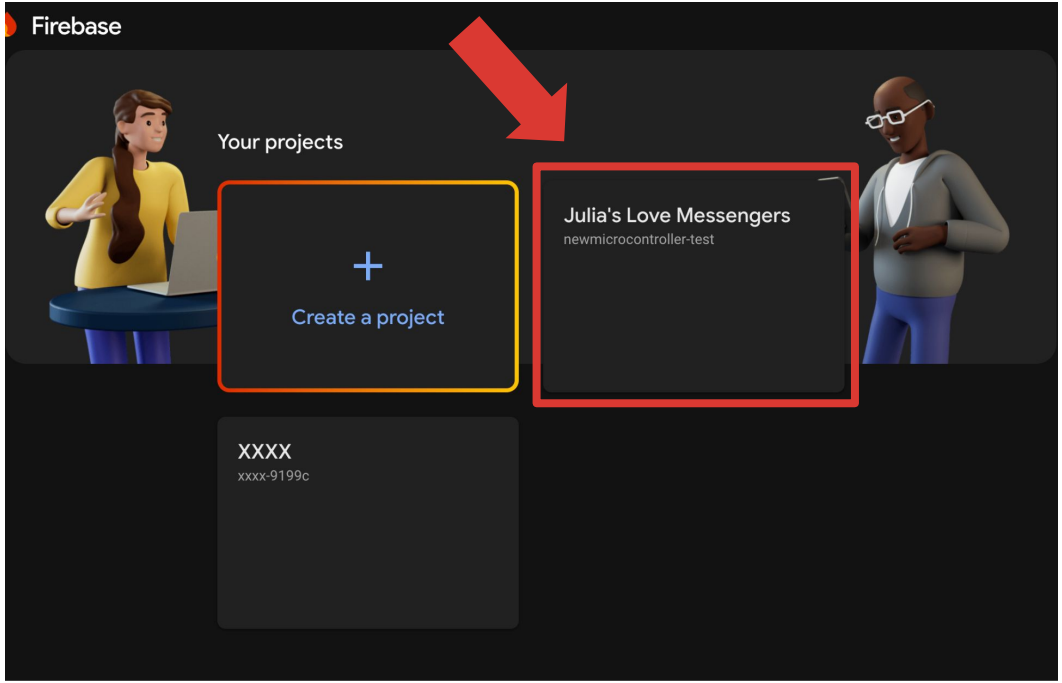
```
// Importing libraries
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>

// Firebase requires these helpers
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// Wi-Fi credentials
#define WIFI_SSID "insert Wifi Name here"
#define WIFI_PASSWORD "insert Wifi Password here"

// Firebase credentials
#define API_KEY "insert API Key here"
#define DATABASE_URL "insert API Url here"

// Hardware components (LED & button)
const int ledPin = D2;
const int buttonPin = D5;
const int buttonLedPin = D6;

// Global variables
bool firebaseData = false;
int buttonState = 1;

// Define Firebase Data object and other necessary objects
```

Ln 114, Col 2 × No board selected

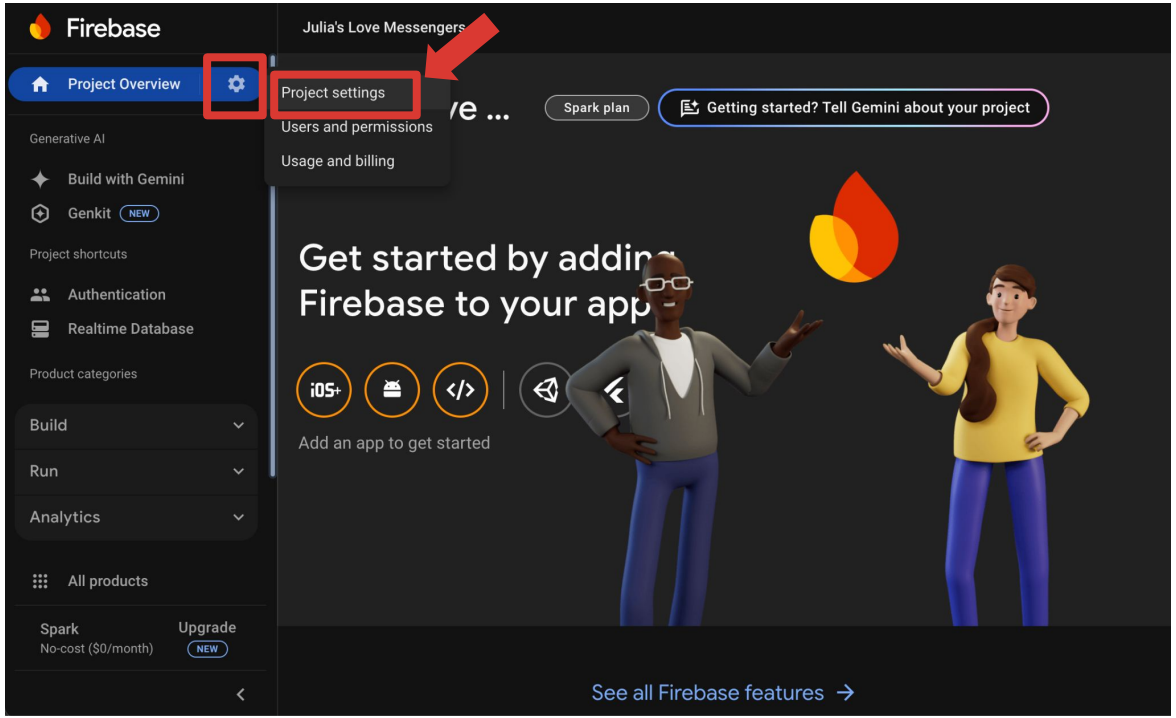
For example:

```
// Wi-Fi credentials
#define WIFI_SSID "Verizon-R500L6"
#define WIFI_PASSWORD "juliaswifipassword"
```

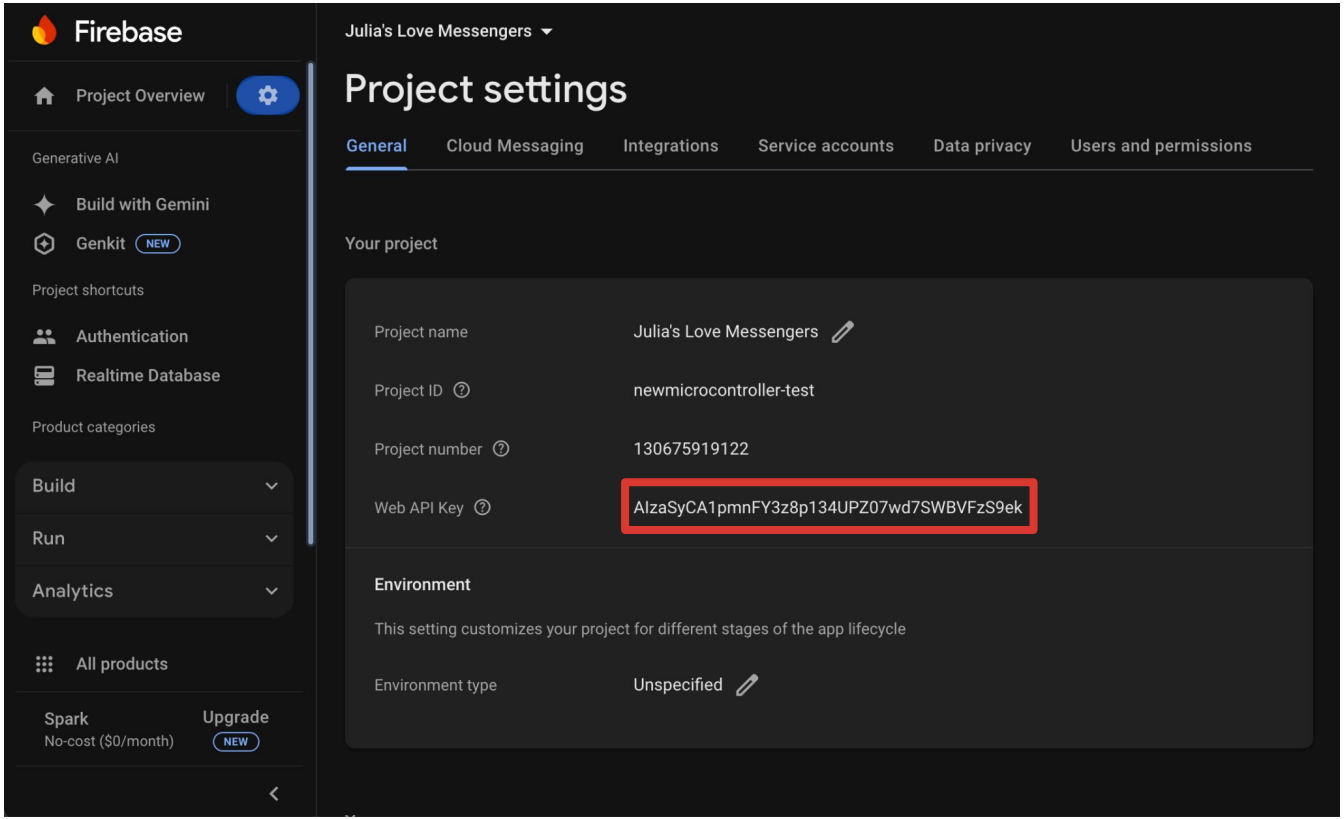
Step 13: Next, you insert the unique API Key of your Database in the code.To find it, head to console.firebase.google.com, and select your Firebase Project. If you have not yet created a Firebase Project, head to our “CreateDatabase.pdf” file in the Slides folder of our Github.



Step 14: Inside your Firebase Console, select the little **gear icon**, and go to "**Project Settings**"



Step 15: Inside your Project Settings, you will find your API Key. Copy the API Key.



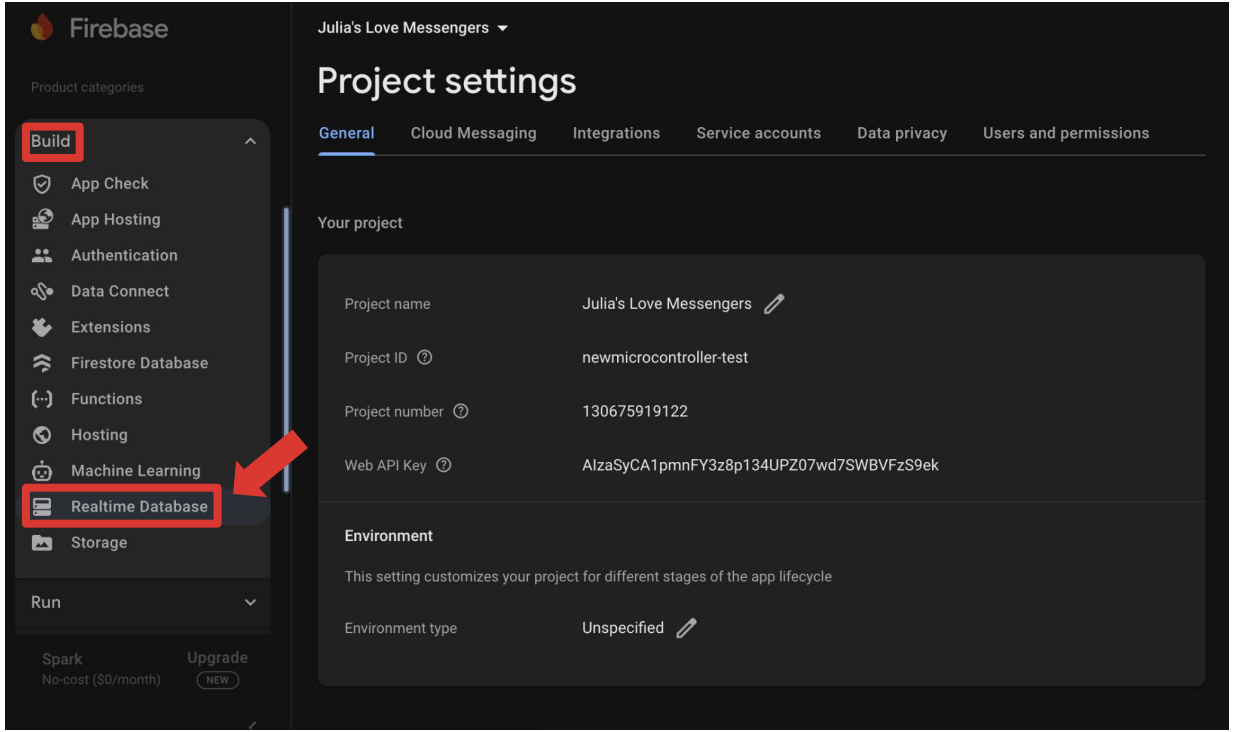
Step 16: Head back to the Arduino code, and insert the API Key inside the double quotes after “#define API_KEY”

Select Board

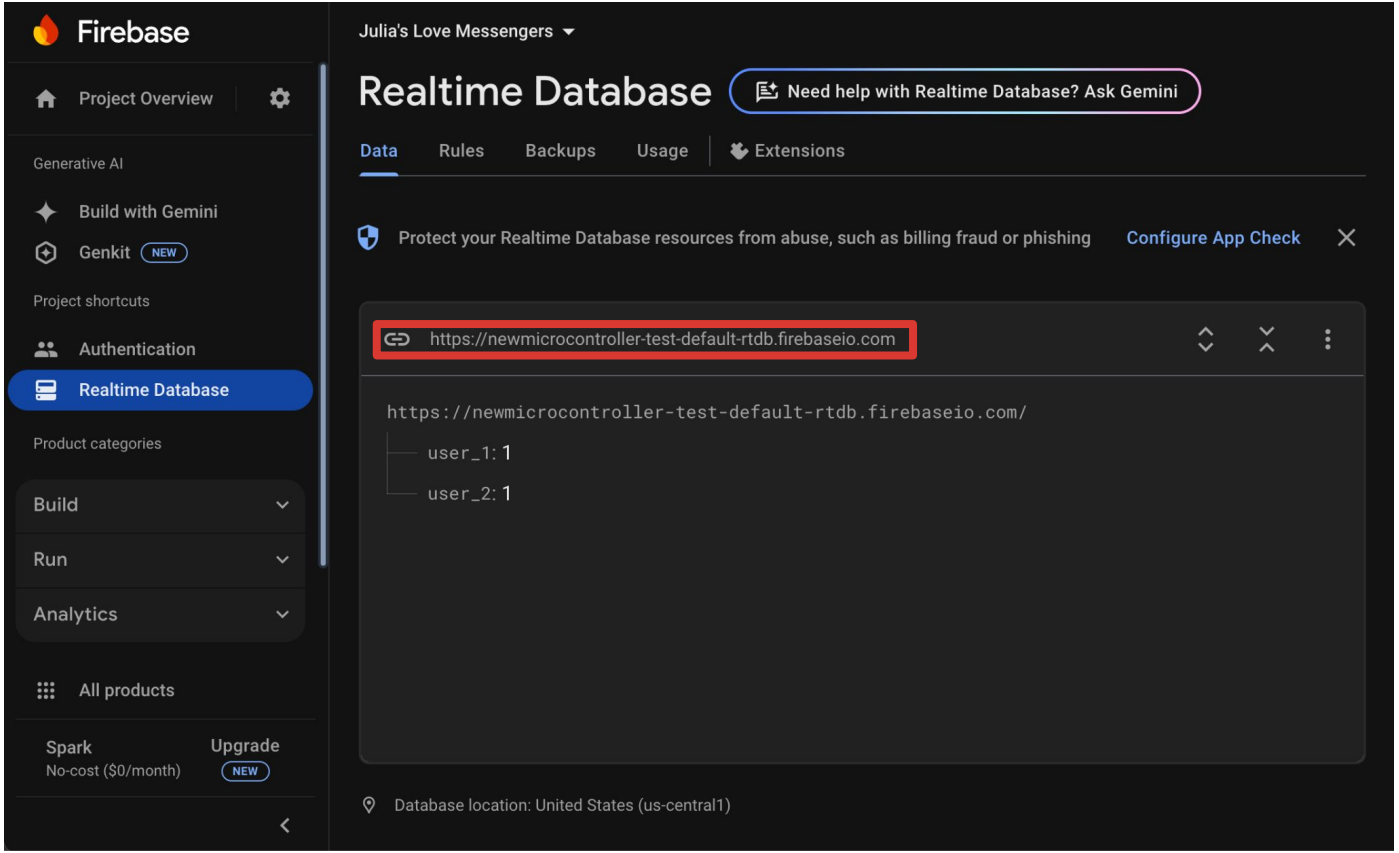
sketch_jan26a.ino

```
// Importing libraries
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>

// Firebase requires these helpers
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// Wi-Fi credentials
#define WIFI_SSID "insert Wifi Name here"
#define WIFI_PASSWORD "insert Wifi Password here"

// Firebase credentials
#define API_KEY "AIzaSyCA1pmnFY3z8p134UPZ07wd7SWBVFzS9ek"
#define DATABASE_URL "insert API Url here"

// Hardware components (LED & button)
const int ledPin = D2;
const int buttonPin = D5;
const int buttonLedPin = D6;

// Global variables
bool firebaseData = false;
int buttonState = 1;

// Define Firebase Data object and other necessary objects
```

Step 17: Next, we will find your Database- URL. On Firebase, head to "**Build**" and select "**Realtime Database**"



Step 18: Copy your Database URL.



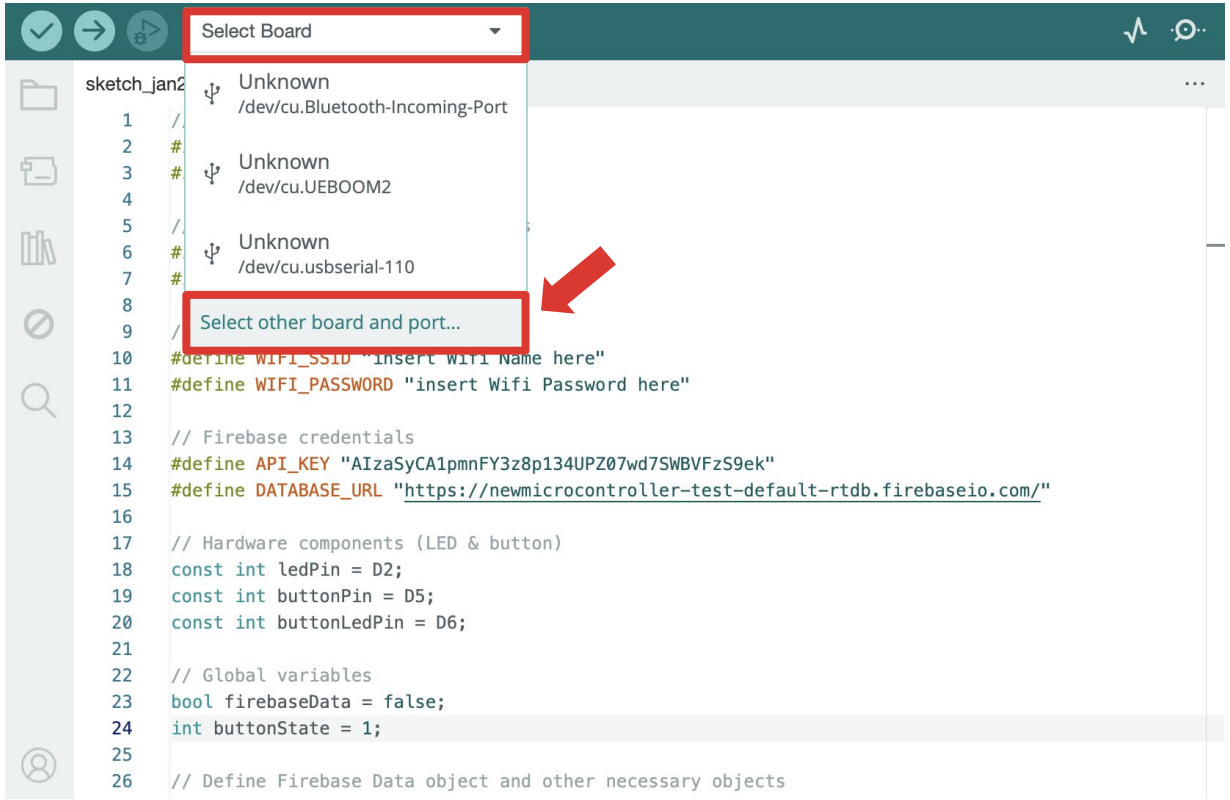
Step 19: Paste the Database URL into the Arduino Code into the double quotes after “#define DATABASE_URL”

Select Board

sketch_jan26a.ino

```
// Importing libraries
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>

// Firebase requires these helpers
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// Wi-Fi credentials
#define WIFI_SSID "insert Wifi Name here"
#define WIFI_PASSWORD "insert Wifi Password here"

// Firebase credentials
#define API_KEY "AIzaSyCA1pmnFY3z8p134UPZ07wd7SWBVFzS9ek"
#define DATABASE_URL "https://newmicrocontroller-test-default-rtdb.firebaseio.com/"

// Hardware components (LED & button)
const int ledPin = D2;
const int buttonPin = D5;
const int buttonLedPin = D6;

// Global variables
bool firebaseData = false;
int buttonState = 1;

// Define Firebase Data object and other necessary objects
```

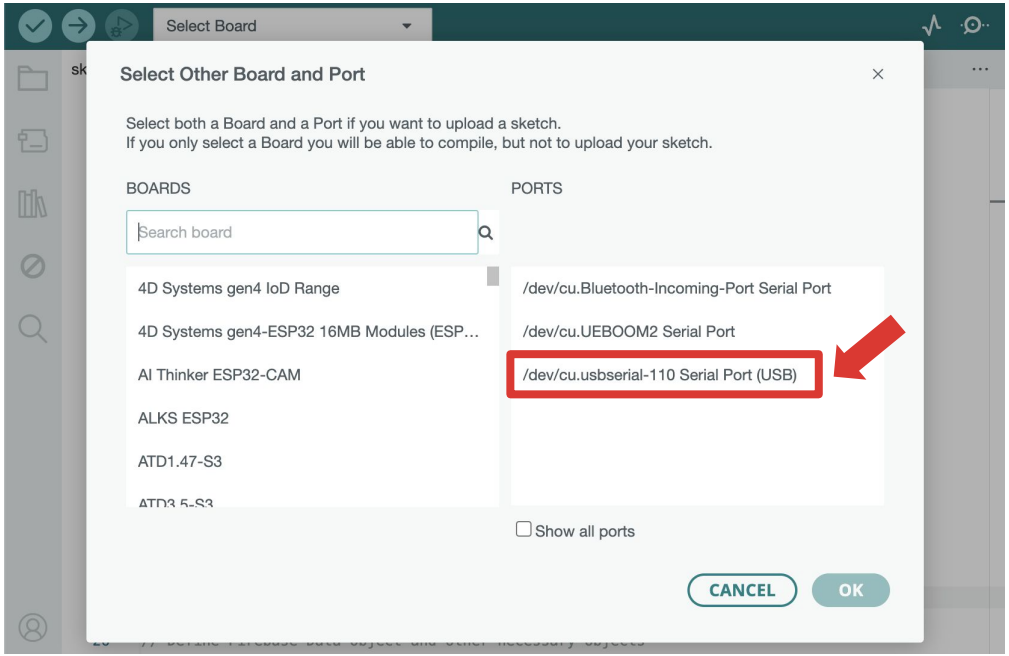
Step 20: The code is now ready to be uploaded to the first Love Messenger. Go to "**Select Board**", and go to "**Select other board and port…**"



Step 21: Plug in your first Love Messenger to your Computer using a data-transfer USB cable.

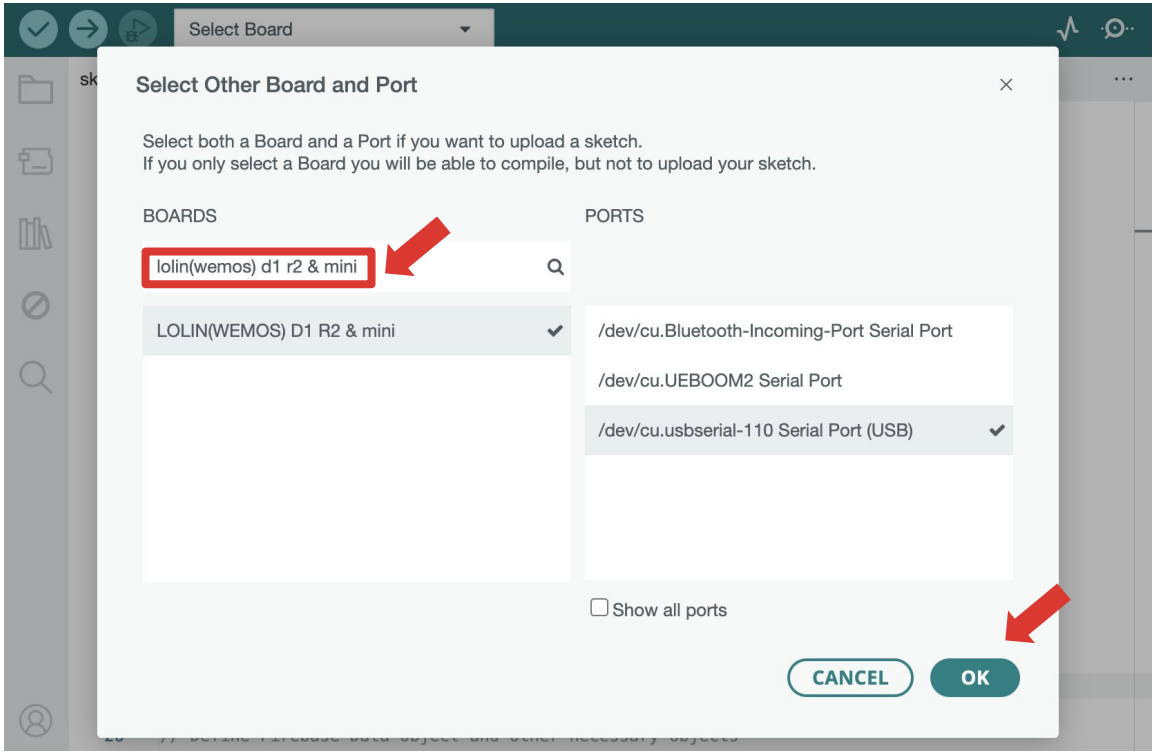
Once the Love Messenger is plugged in, a new port should appear in the “Ports” section. Depending on if you are using a Mac or PC, the port names can vary (“COM” for PC, and “usbserial” for Mac).

Select the new port.

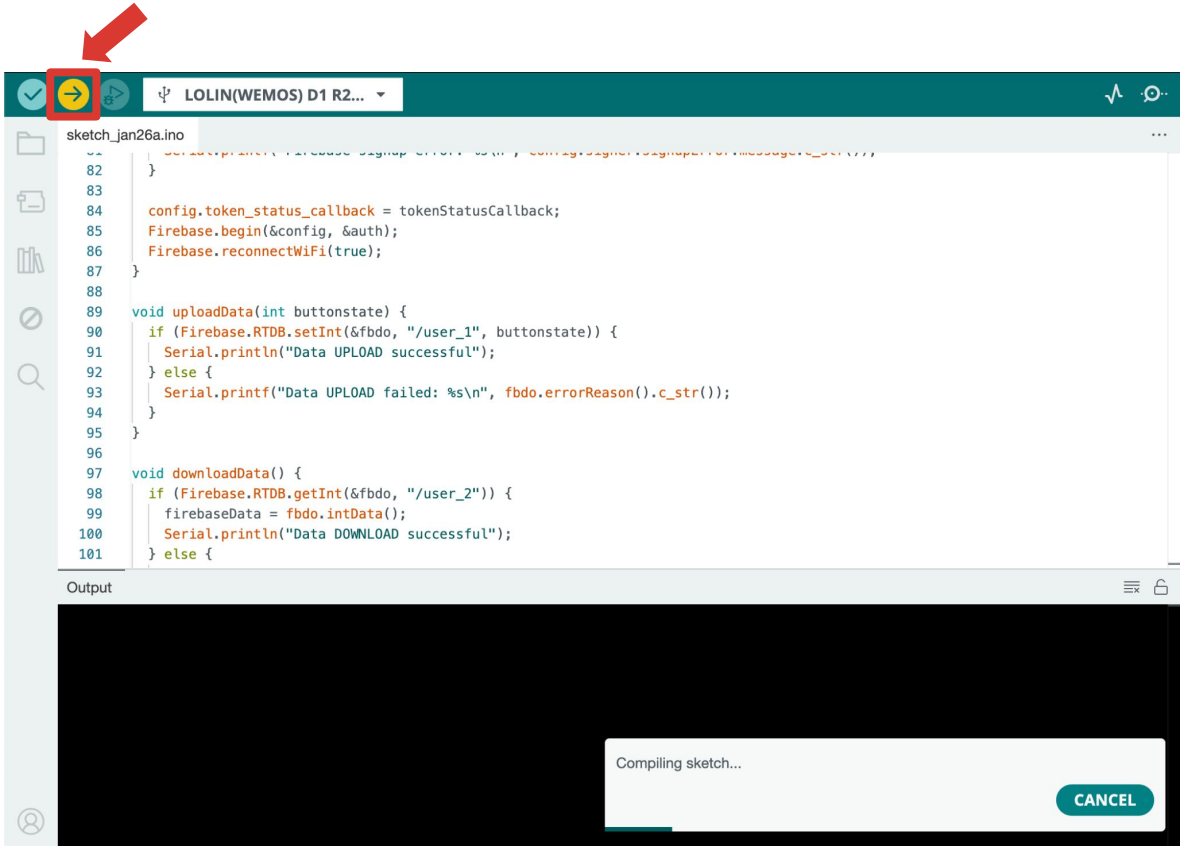


**If your port does not show up, you might be using a charge-only USB wire that doesn’t transmit data. You need to use a USB wire that also can do data-transfer. We have the ones we used in the workshop linked on Github.*

Step 22: In the “**Boards**” section, search for “**lolin(wemos) d1 r2 & mini**”, and select it. Then Press OK

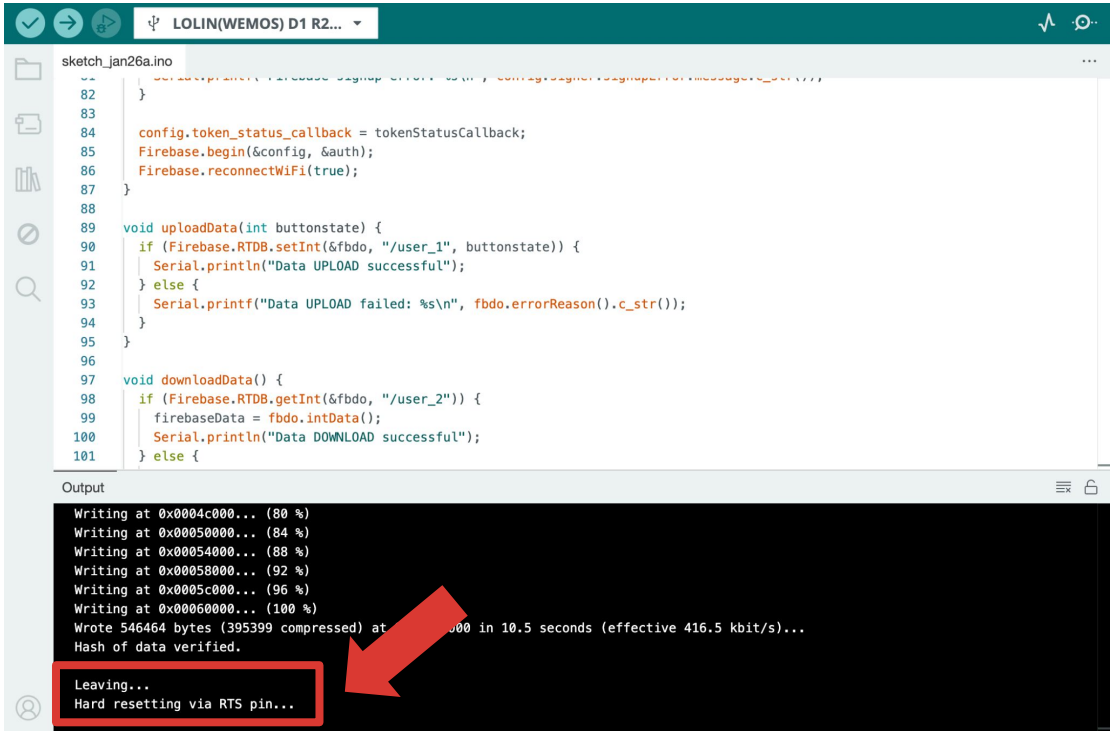


Step 24: The code is now ready to be uploaded. Hit the little arrow icon. This will compile the code (takes a few minutes), and will upload it to your Love Messenger.



Step 25: The Sketch will first compile and then upload.

The code is uploaded once the "**Output**" window reads "**Hard resetting via RTS pin…**"



Step 26: To check if our code works, open the Serial Monitor.

LOLIN(WEMOS) D1 R2...

sketch_jan26a.ino

```
  }

  config.token_status_callback = tokenStatusCallback;
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}

void uploadData(int buttonstate) {
  if (Firebase.RTDB.setInt(&fbdo, "/user_1", buttonstate)) {
    Serial.println("Data UPLOAD successful");
  } else {
    Serial.printf("Data UPLOAD failed: %s\n", fbdo.errorReason().c_str());
  }
}

void downloadData() {
  if (Firebase.RTDB.getInt(&fbdo, "/user_2")) {
    firebaseData = fbdo.intData();
    Serial.println("Data DOWNLOAD successful");
  } else {
```

Output

```
Writing at 0x0004c000... (80 %)
Writing at 0x00050000... (84 %)
Writing at 0x00054000... (88 %)
Writing at 0x00058000... (92 %)
Writing at 0x0005c000... (96 %)
Writing at 0x00060000... (100 %)
Wrote 546464 bytes (395399 compressed) at 0x00000000 in 10.5 seconds (effective 416.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Step 27: Change the baud rate to **115200 baud**.

LOLIN(WEMOS) D1 R2...

sketch_jan26a.ino

```
}

void downloadData() {
  if (Firebase.RTDB.getInt(&fbdo, "/user_2")) {
    firebaseData = fbdo.intData();
    Serial.println("Data DOWNLOAD successful");
  } else {
    Serial.printf("Data DOWNLOAD failed: %s\n", fbdo.errorReason().c_str());
  }
}

void manageLED(int buttonState, int firebaseData) {
  if (buttonState == HIGH || firebaseData == HIGH) {
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Output Serial Monitor ×

Message (Enter to send message to 'LOLIN(WEMOS) D1 R2 & mini' on '/dev/cu.usbserial-110') New Line 115200 baud

```
1
Data UPLOAD successful
Data DOWNLOAD successful
1
Data UPLOAD successful
Data DOWNLOAD successful
1
Data UPLOAD successful
Data DOWNLOAD successful
```

Step 28: In the serial Monitor, you should see these messages;

The Love Messenger will first connect to Wifi. (If it doesnt connect, double check your wifi strength, and if you correctly entered your name and password.)

The Wifi is connected

Once the Serial Monitor prints this, we can move on to the second Love Messenger. Everything is set up.

Output Serial Monitor ×

Message (Enter to send message to 'ESP32 Dev Module' on '/dev/cu.usbserial-0001')

```
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.470 -> Connecting to Wifi...
18:10:33.774 -> Connected with IP: 172.22.52.167
18:10:34.894 -> Firebase successful
18:10:35.518 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:40.008 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:43.701 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:47.495 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:50.266 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:53.112 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:56.809 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:11:00.107 -> Data UPLOAD successful,  Data DOWNLOAD failed
```

Step 29: Now, unplug your first Love Messenger, and plug in your second Love Messenger.

Step 30: In your code, in the **`uploadData`** function, change "/user_**1**" to "/user_**2**"

```
89  void uploadData(int buttonstate) {
90    if (Firebase.RTDB.setInt(&fbdo, "/user_1", buttonstate)) {
91      Serial.println("Data UPLOAD successful");
92    } else {
93      Serial.printf("Data UPLOAD failed: %s\n", fbdo.errorReason().c_str());
94    }
95  }
```

Change to

```
89  void uploadData(int buttonstate) {
90    if (Firebase.RTDB.setInt(&fbdo, "/user_2", buttonstate)) {
91      Serial.println("Data UPLOAD successful");
92    } else {
93      Serial.printf("Data UPLOAD failed: %s\n", fbdo.errorReason().c_str());
94    }
95  }
```

Step 31: Similarly, in your code, in the **`download data`** function, change "/user_**2**" to "/user_**1**"

```
97  void downloadData() {
98    if (Firebase.RTDB.getInt(&fbdo, "/user_2")) {
99      firebaseData = fbdo.intData();
100     Serial.println("Data DOWNLOAD successful");
101   } else {
102     Serial.printf("Data DOWNLOAD failed: %s\n", fbdo.errorReason().c_str());
103   }
104 }
```

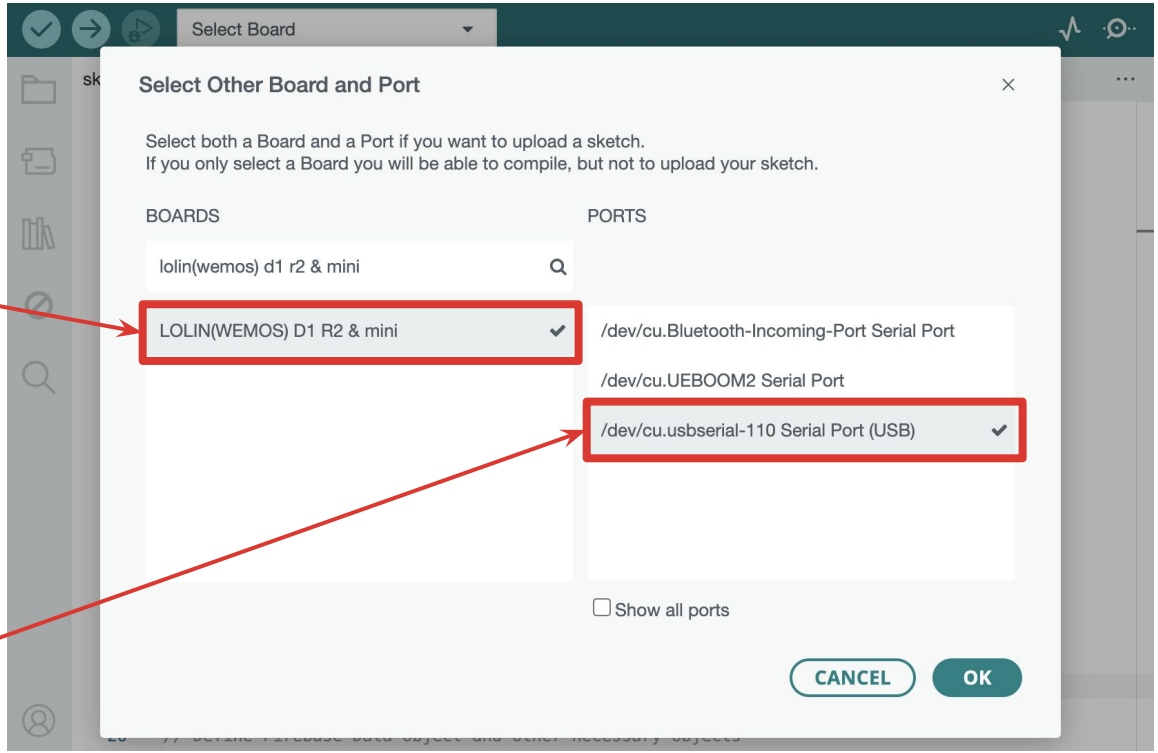
Change to

```
97  void downloadData() {
98    if (Firebase.RTDB.getInt(&fbdo, "/user_1")) {
99      firebaseData = fbdo.intData();
100     Serial.println("Data DOWNLOAD successful");
101   } else {
102     Serial.printf("Data DOWNLOAD failed: %s\n", fbdo.errorReason().c_str());
103   }
104 }
```

Step 32:

Once again, under “Select Board”, you should have “**lolin(wemos) d1 r2 & mini**” selected under BOARDS.

Check that your second Love Messenger is properly connected to a port.



**If your Love Messenger is not showing up in the PORTS, try disconnecting the wire from your Love Messenger and connecting it again.*

Step 33: Now, upload the code to the second Love Messenger!

Once again, be patient! Your code might take awhile to compile and upload.

Step 34: Once it has successfully uploaded, check the serial monitor again.

Once again, it needs to go through these steps in order to successfully connect to Firebase

```
18:20:03.343 -> Connecting to Wifi...
18:20:03.638 -> Connected with IP: 172.22.52.101
18:20:04.963 -> Firebase successful
18:20:05.490 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:07.597 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:09.801 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:11.912 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:14.156 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:16.420 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:18.511 -> Data UPLOAD successful,  Data DOWNLOAD successful
18:20:20.758 -> Data UPLOAD successful,  Data DOWNLOAD successful
```

Step 35: Once you see **`“Data UPLOAD successful, Data DOWNLOAD successful”`**, this means that your second Love Messenger has successfully connected to Firebase, together with your first Love Messenger!

Step 36: Now, plug both Love Messengers into your laptop/ any power supply.

Wait a few moments for both Love Messengers to connect to Wifi.

If everything is successful, you should see both Love Messengers working!

Tips:

1. Be patient: press the button for a few seconds until you see your Love Messenger light up
2. Unplugging and replugging the Love Messenger can help too!

Step 37: While the Love Messengers are working, you should be able to see Firebase updating in real time.

Default Firebase states

https://newmicrocontroller-test-default-rtdb.firebaseio.com

```
https://newmicrocontroller-test-default-rtdb.firebaseio.com/
├── user_1: 0
└── user_2: 0
```

When user_1's button is successfully pressed- both Love Messengers should light up!

https://newmicrocontroller-test-default-rtdb.firebaseio.com

```
https://newmicrocontroller-test-default-rtdb.firebaseio.com/
├── user_1: 1
└── user_2: 0
```

Troubleshooting!

1. If only one Love Messenger is working…

Unplug both Love Messengers, and **plug only one** into your laptop again.

Check the Serial Monitor: Make sure that its connection to Firebase is successful.

Then, **plug in your second Love Messenger again**, and make sure it is also successfully connected to Wifi and Firebase.

Output Serial Monitor ×

Message (Enter to send message to 'ESP32 Dev Module' on '/dev/cu.usbserial-0001')

```
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.172 -> Connecting to Wifi...
18:10:33.470 -> Connecting to Wifi...
18:10:33.774 -> Connected with IP: 172.22.52.167
18:10:34.894 -> Firebase successful
18:10:35.518 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:40.008 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:43.701 -> Data UPLOAD successful,  Data DOWNLOAD failed
18:10:47.495 -> Data UPLOAD successful,  Data DOWNLOAD failed
```

Troubleshooting!

2. If you are getting a “Token error” in your serial monitor:

Upload the code again to the respective Love Messenger.

Troubleshooting!

There are many other things that can go wrong with delicate code

Reach out to us if you have any more questions:

Email: yiqing.ng@gmail.com

Instagram: @julia.daser

We are more than happy to help!❤️