

# Bring List

- Water
- Bring 5 kits of each color
- Bring 5 extra kits
- Extra Screws/ESPs/buttons/leds/wires
- All necessary Tools
- Wires of that guy
- Soldering Stations + Extra tips
- Solder equipment
- Ipad + send PDF to pepi
- Camera + SD card
- Laptop Charger
- “Tripod”
- Business Cards LOL
- Portable charger

# Prep before presentation:

- Open **Github**
- Open **Firebase**
- Pepi to start the zoom meeting
- Tables and soldering stations
- Set up the tools at the front table

# Prepare List

- Delete Firebase DB
- Uninstall Library

# Pepi

- Set up ESP to be in a good part of the frame

Prepare Table Layout + Chairs

Distribute materials on table

## **During Talk:**

- Distribute USB wires
- Help problem-solve

# Julia

Set up Laptop (+all windows)

Set up right camera settings + Talk to volunteer

Help setup Table/Extension Cords etc.

## **During Talk:**

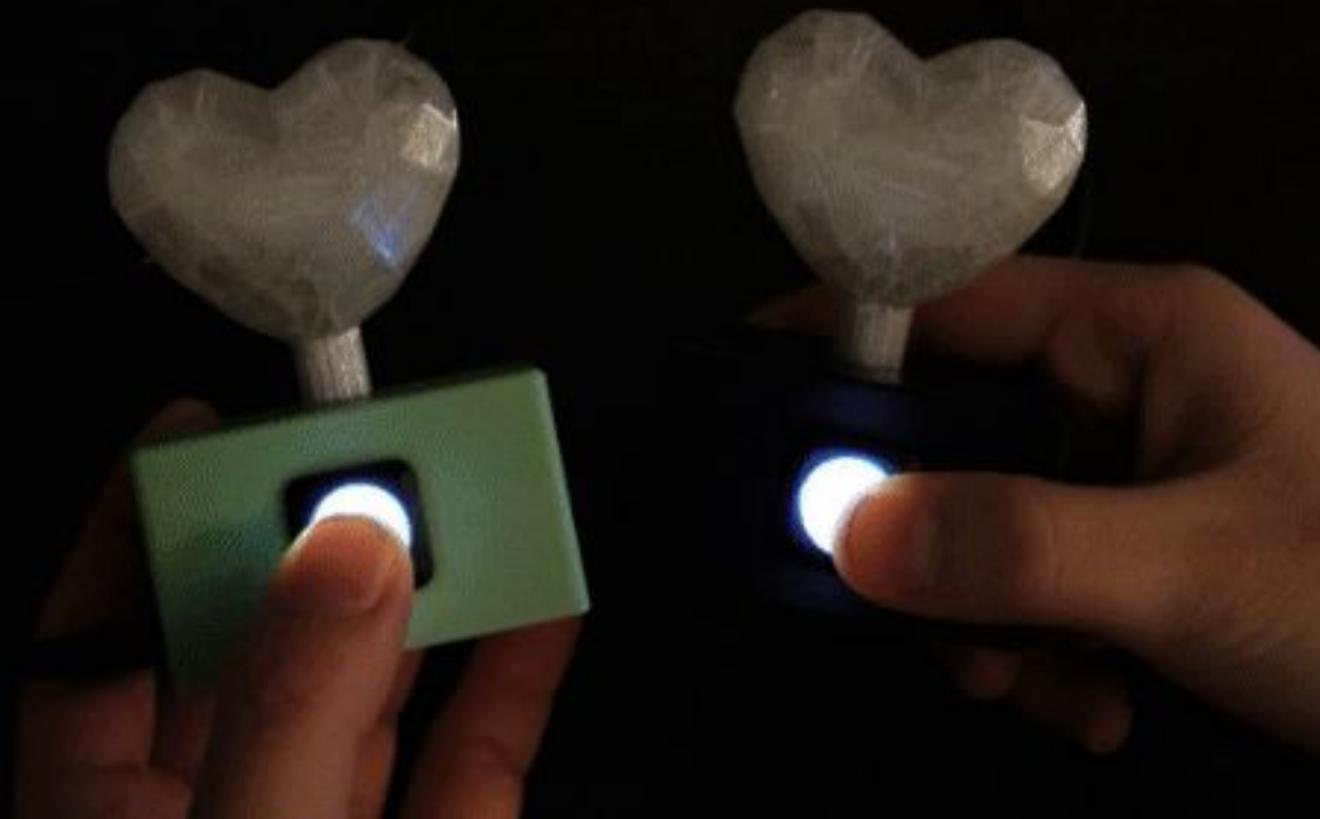
- Help with assembly
- Help with soldering

# Build Your Own Love Messengers

SIGGRAPH Experience Labs

**Today,  
3:00 - 5:00PM**

No prior experience  
needed, all materials  
provided!





Pepi Ng



Julia Daser

We are **2 artists** from **NYC**, that make **fun electronics projects** on Youtube!

@WormiCollective



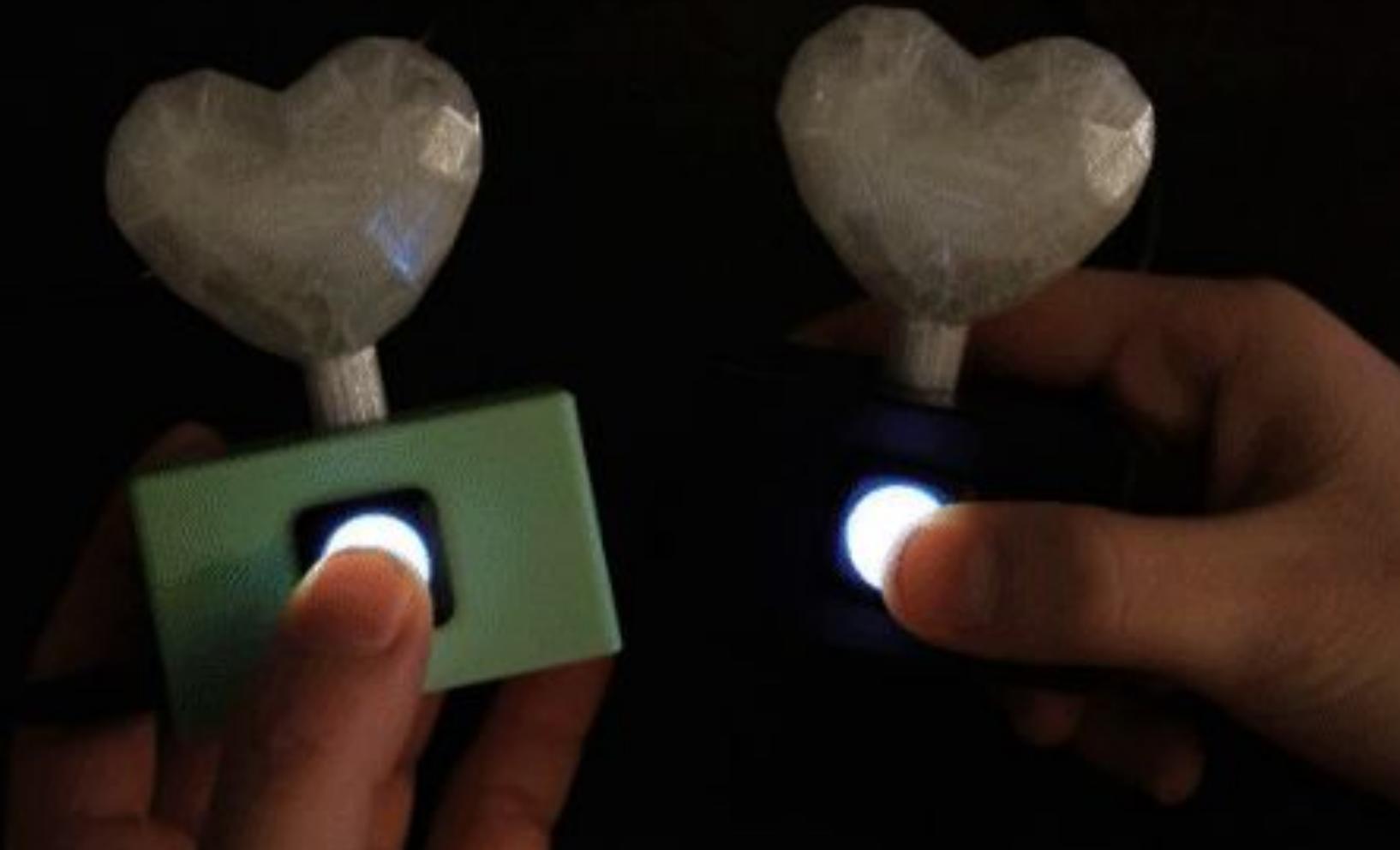
Who here has soldered before?





Who has written Arduino code before?



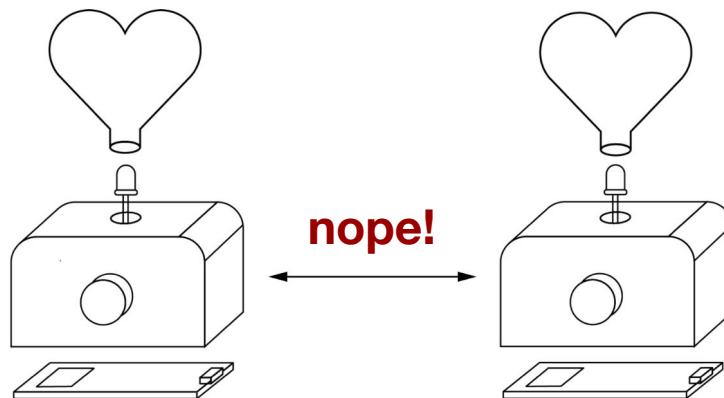


Love Messengers work over **any**  
**distance!**

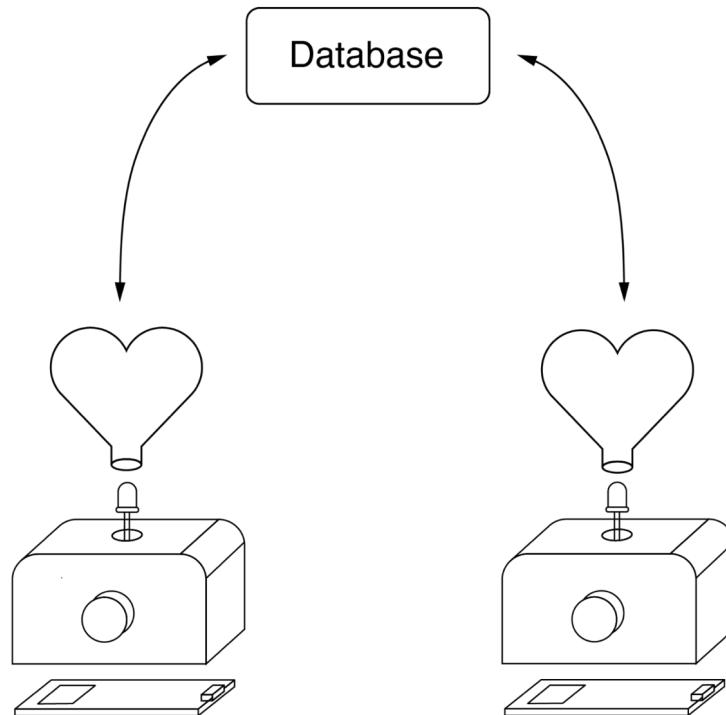
How does that work?



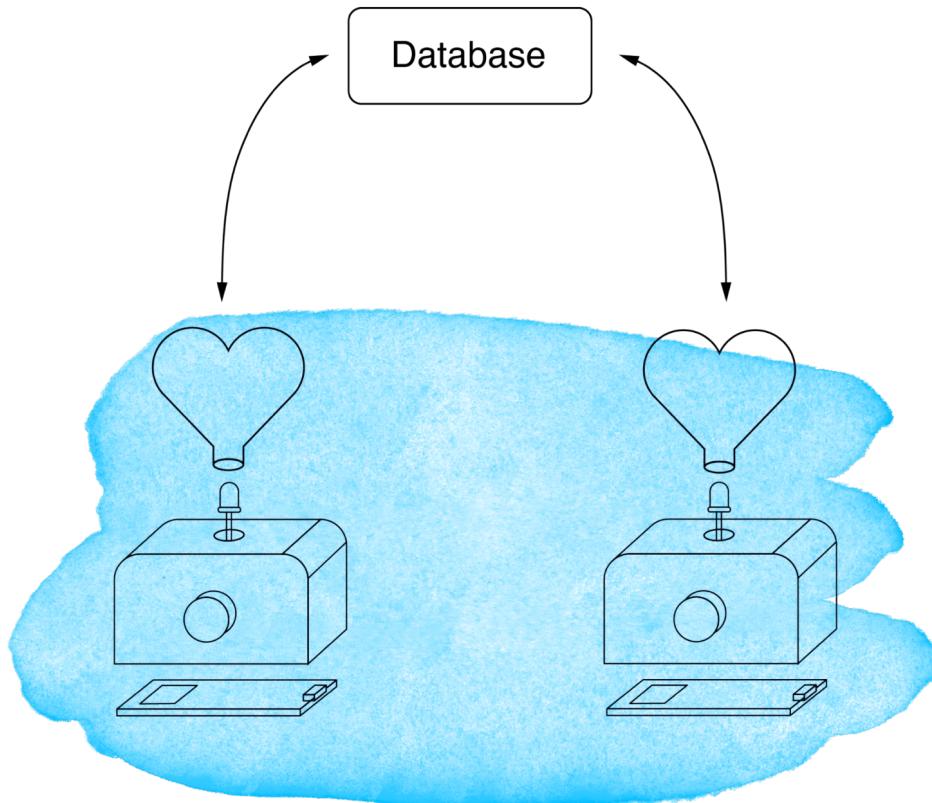
# How do the Love Messengers work?



# How do the Love Messengers work?



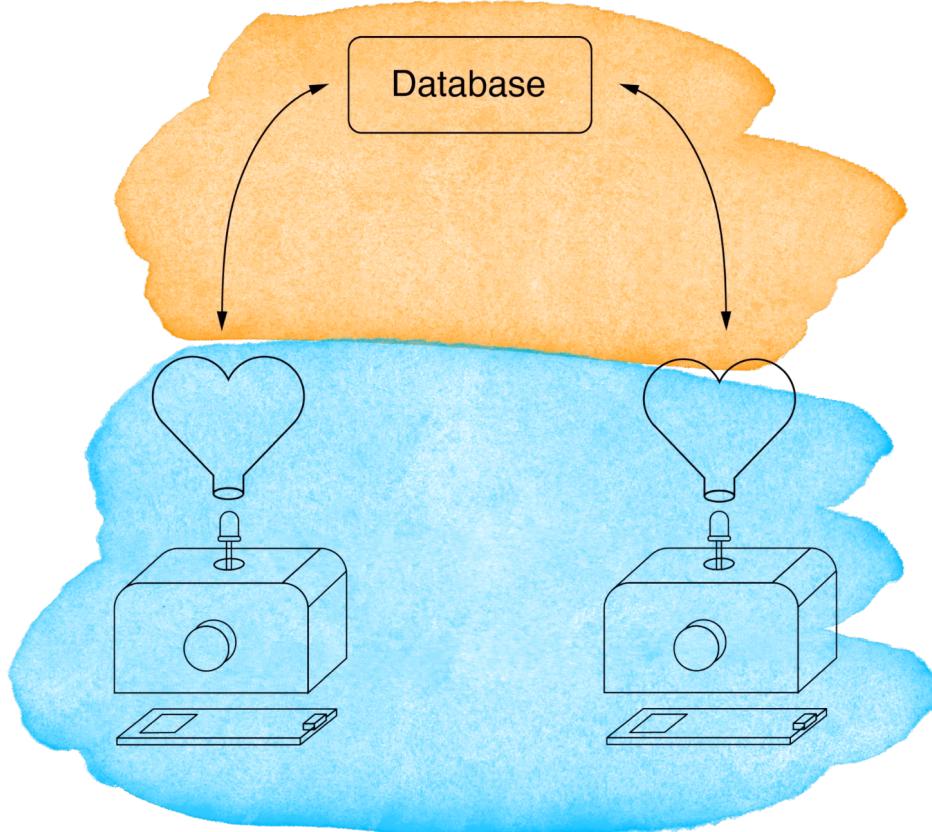
# How do the Love Messengers work?



## Physical Segment (50 mins)

- Soldering
- Assembling

# How do the Love Messengers work?



## Digital Segment (50 mins)

- Database
- Code

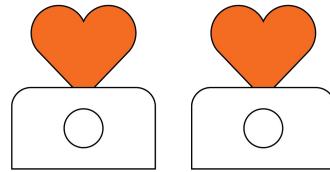
## Physical Segment (50 mins)

- Soldering
- Assembling



**Before we begin,**

We need to download the libraries needed!



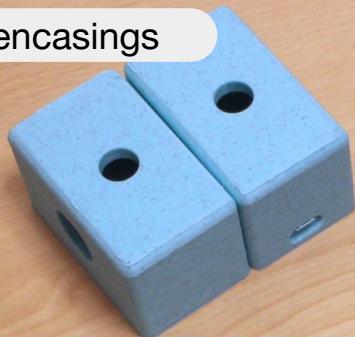
Come and collect your love messenger kits at the front

Settle at the soldering stations

# Physical Segment

# Materials provided for a pair of Love Messengers

2x box encasings



2x floors for encasings



4x heart halves



2x ESP32 microcontrollers



4x screws



2x pre-soldered  
push buttons



2x pre-soldered LEDs



# Why use ESP32 microcontrollers?



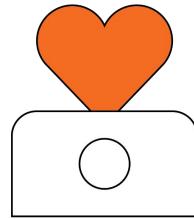
Wifi capabilities



Higher clock speed



More affordable



Let's start with just one love messenger!

# Part 1: Pre-assembly

## STEP 1

- Unscrew the button cap and metal piece

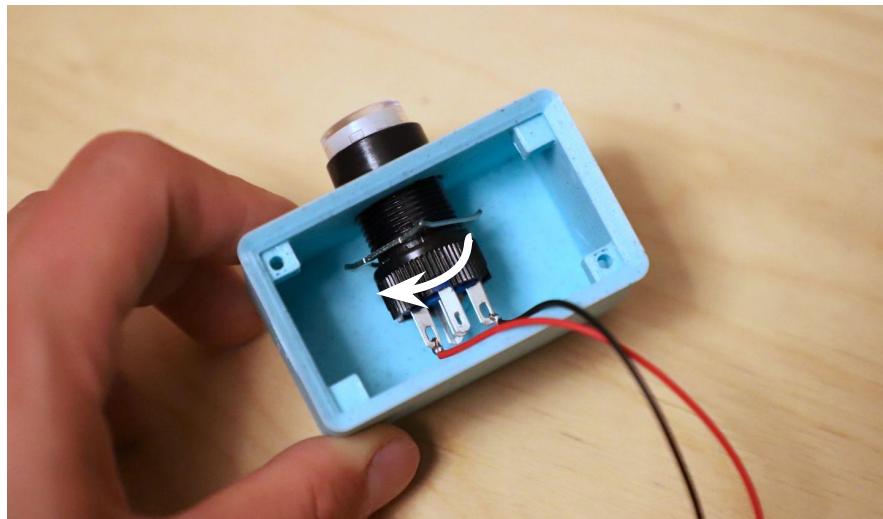
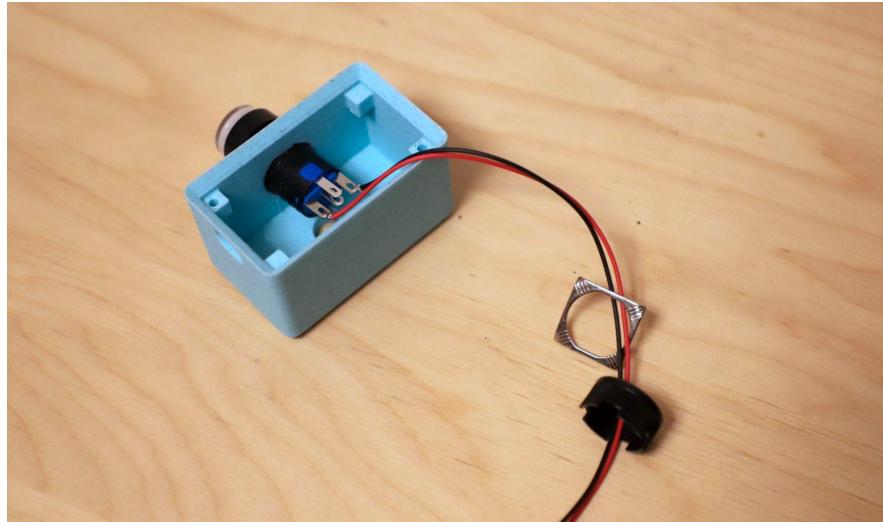


# Part 1: Pre-assembly

## STEP 2

Insert button

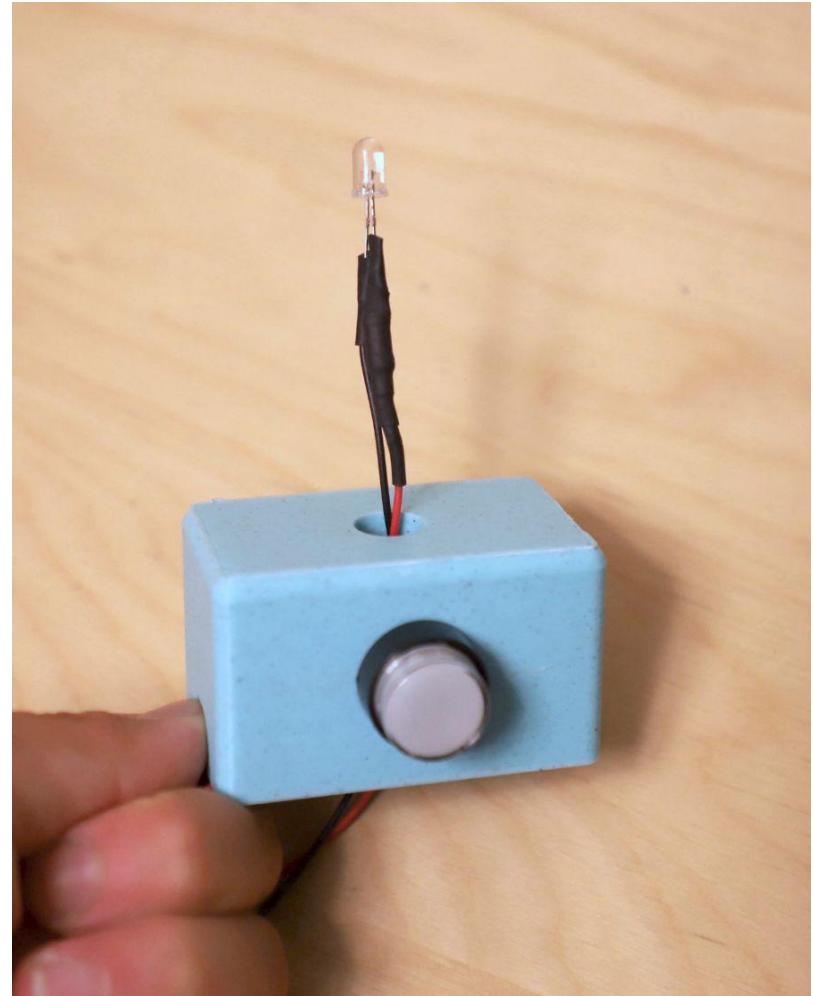
Screw button into the encasing



# Part 1: Pre-assembly

## STEP 3

Stick the pre-soldered LED through the hole at the top of the box encasing.



## Part 2: Soldering

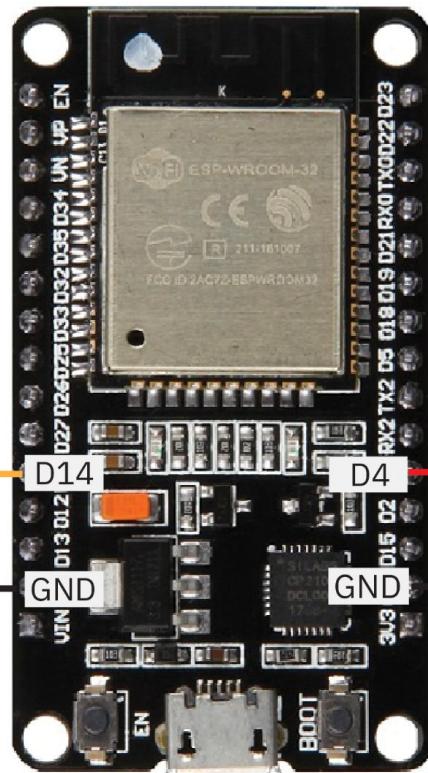
**Live demo of soldering** 



Split into groups:  
Each group is assigned to one soldering station

ESP32

## Button



Pre-soldered LED



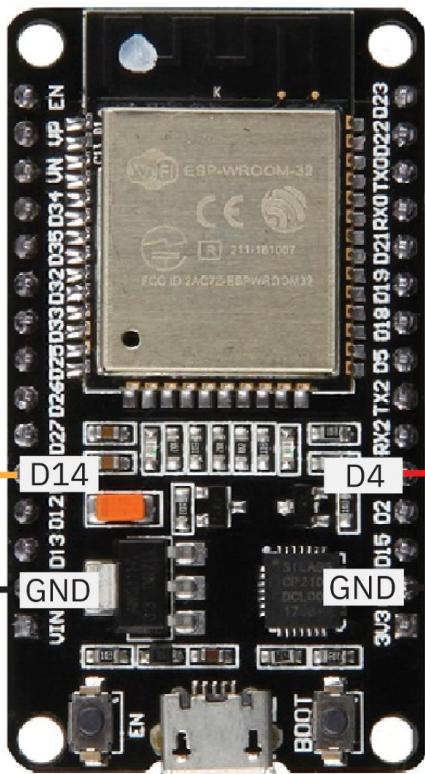
ESP32

Button



Button GND

Button Pin

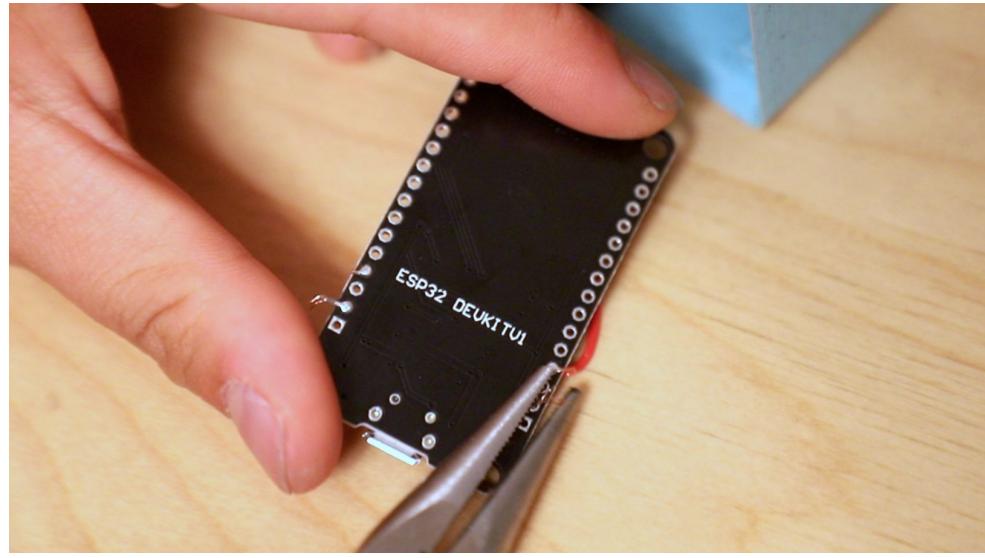
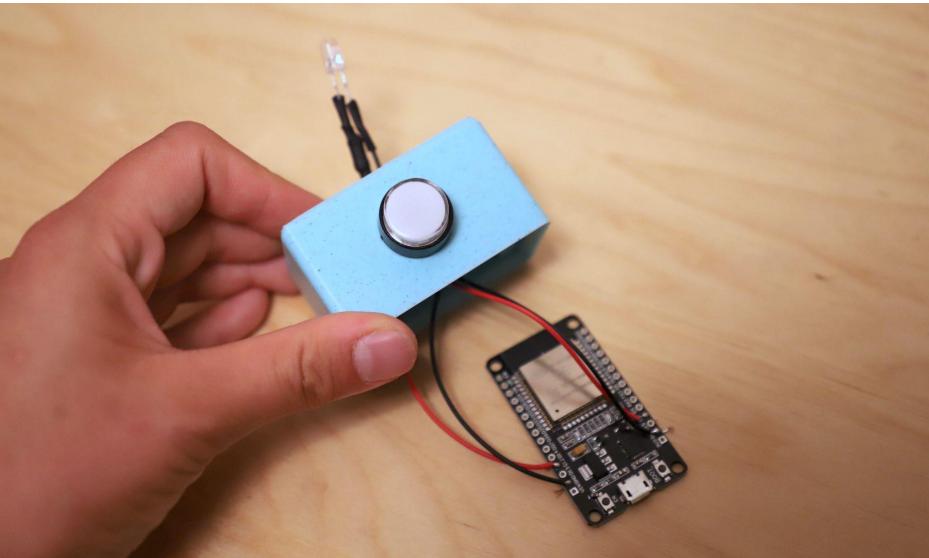


220  $\Omega$   
Resistor

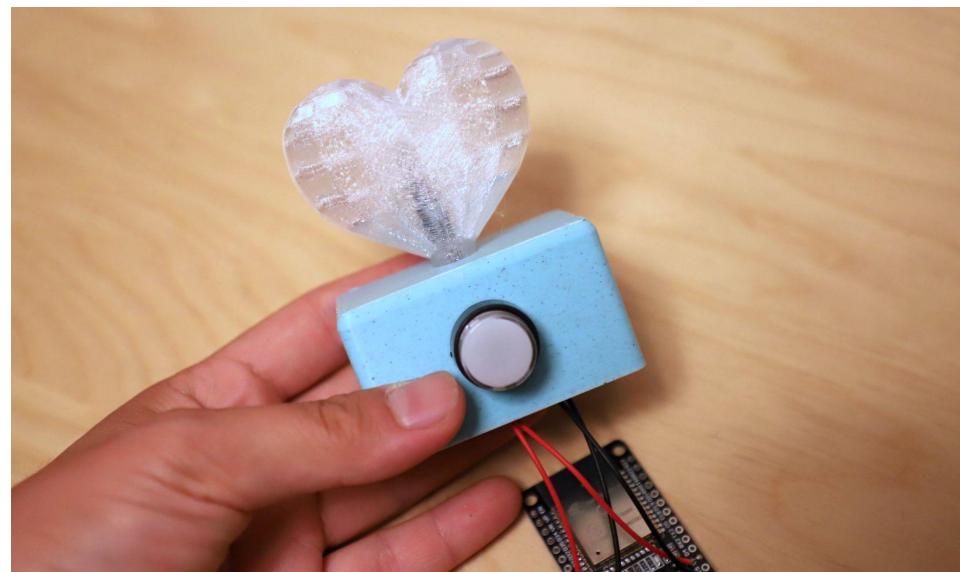
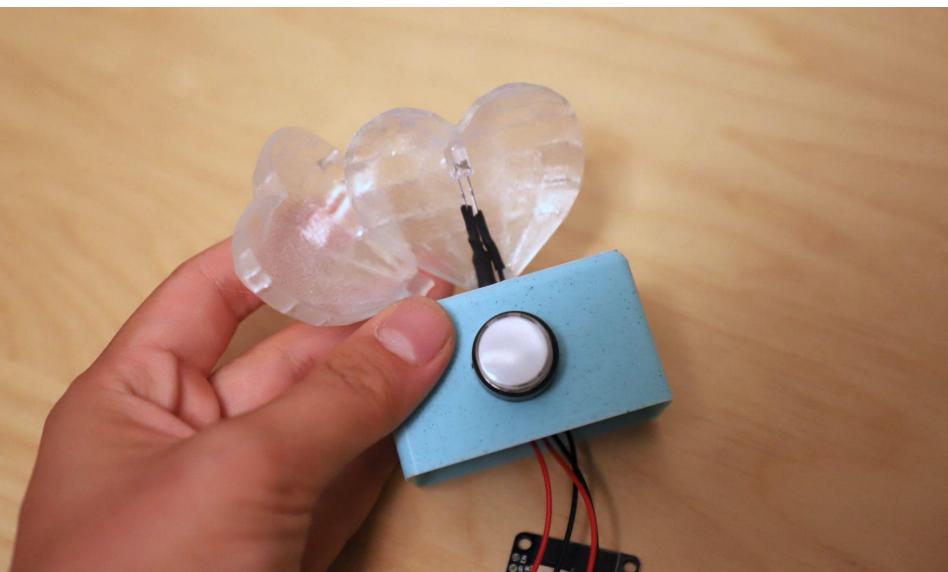
+ -

## Part 2: Soldering

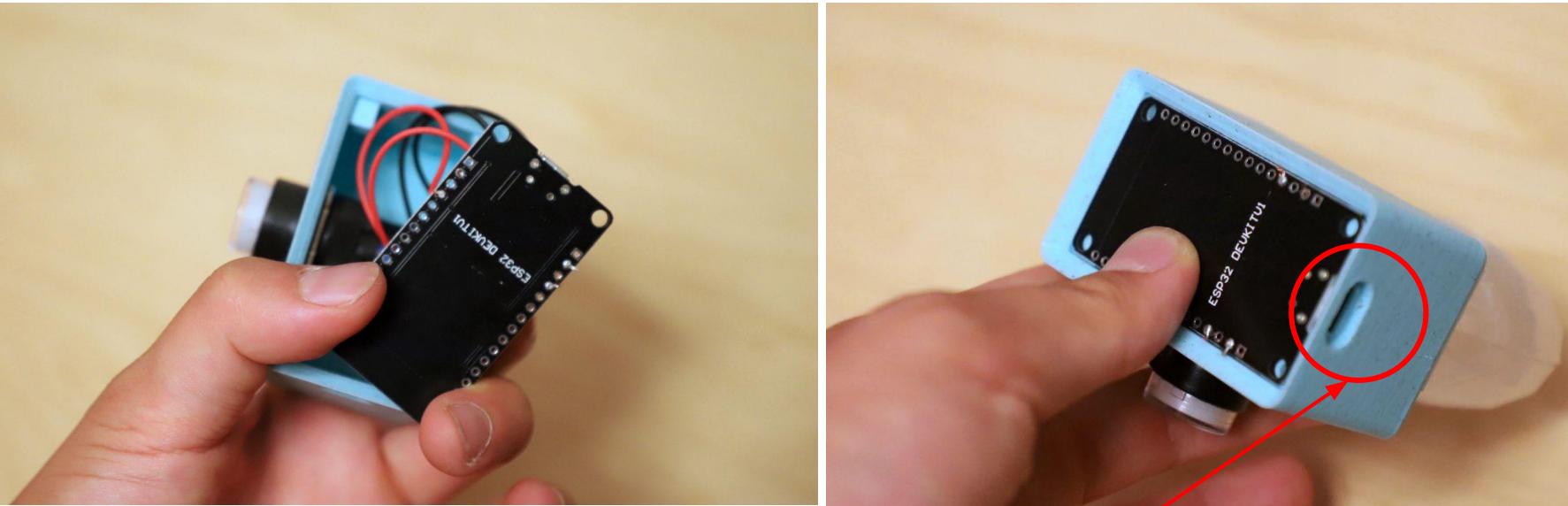
Use pliers to cut off any excess wires



# Part 3: Putting everything together! 😍

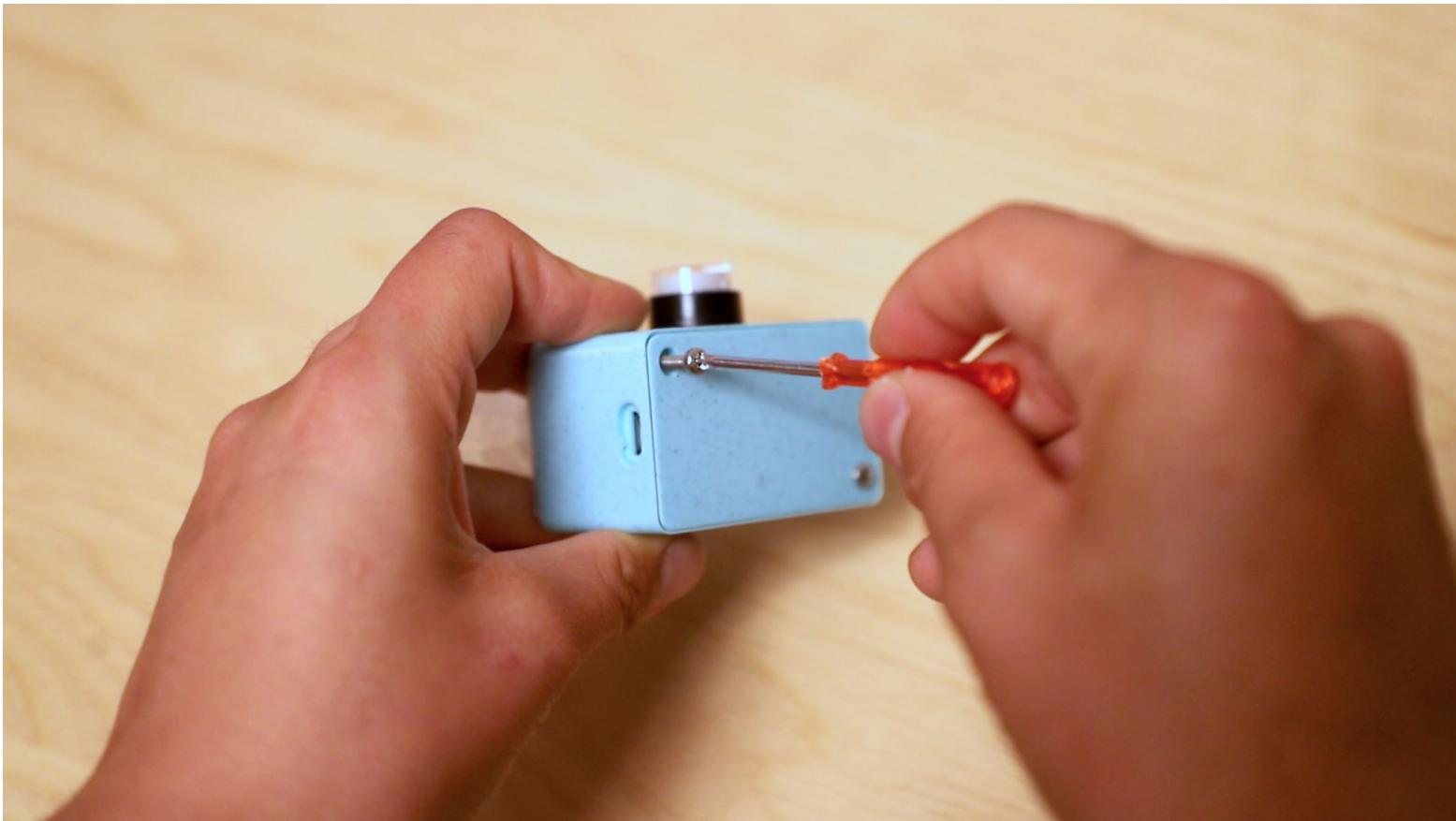


# Part 3: Putting everything together! 😊

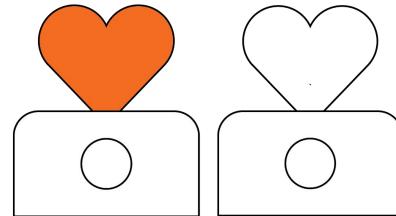


MicroUSB output of the  
microcontroller faces the hole  
on the encasing.

## Part 3: Putting everything together! 😊







You're all done with one love messenger!

Let's now make the second one!

Pre-soldered LED



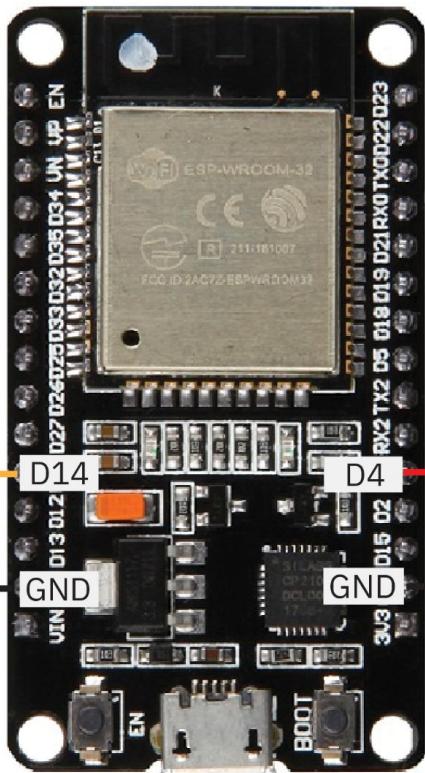
ESP32

Button



Button GND

Button Pin



220  $\Omega$   
Resistor

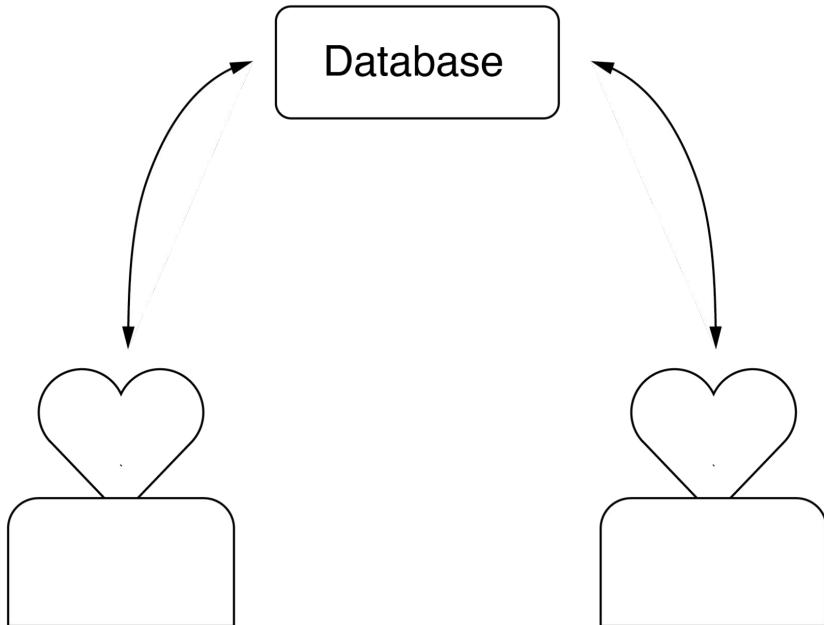
+ -

# Digital Segment

1. Deep-dive: How do the Love Messengers work?
2. Setting up our Real Time Database
3. Writing Arduino Code for the Microcontroller

1. Deep-dive: How do the Love Messengers work?
2. Setting up our Real Time Database
3. Writing Arduino Code for the Microcontroller

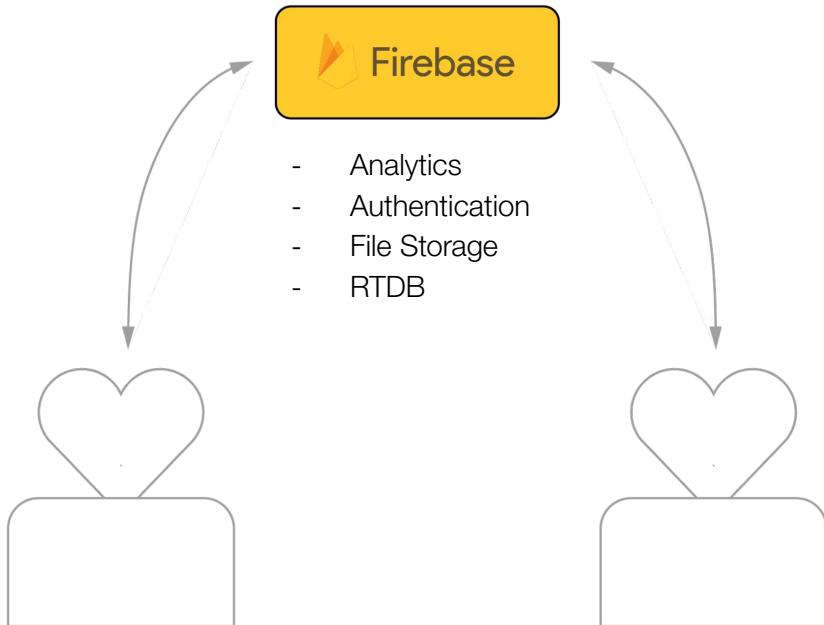
# How does it work?



## Database

Love Messengers need to know each other's button states!

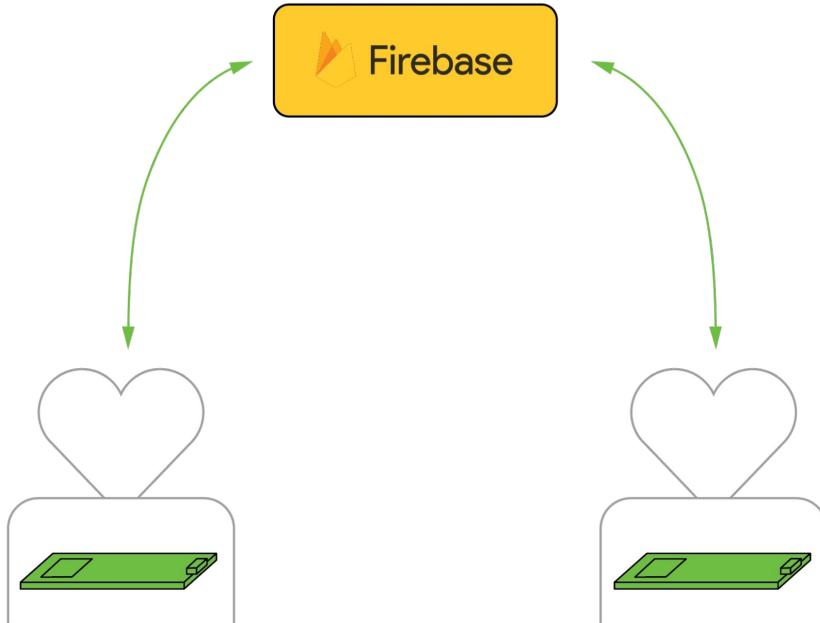
# How does it work?



## Database

1. Create Firebase Login
2. Create RTDB (= Real Time Database)

# How does it work?



## Database

1. Create Firebase Login
2. Create RTDB

## Code

3. Understand Code
4. Upload Code

1. Deep-dive: How do the Love Messengers work?
2. Setting up our Real Time Database
3. Writing Arduino Code for the Microcontroller

# Let's set up Firebase

Go to **console.firebaseio.google.com**

1. How do the Love Messengers work?
2. Setting up our Real Time Database
3. Writing Arduino Code for the Microcontroller

Pepi will also now distribute the wires

# Pseudo Code

Setup-code:

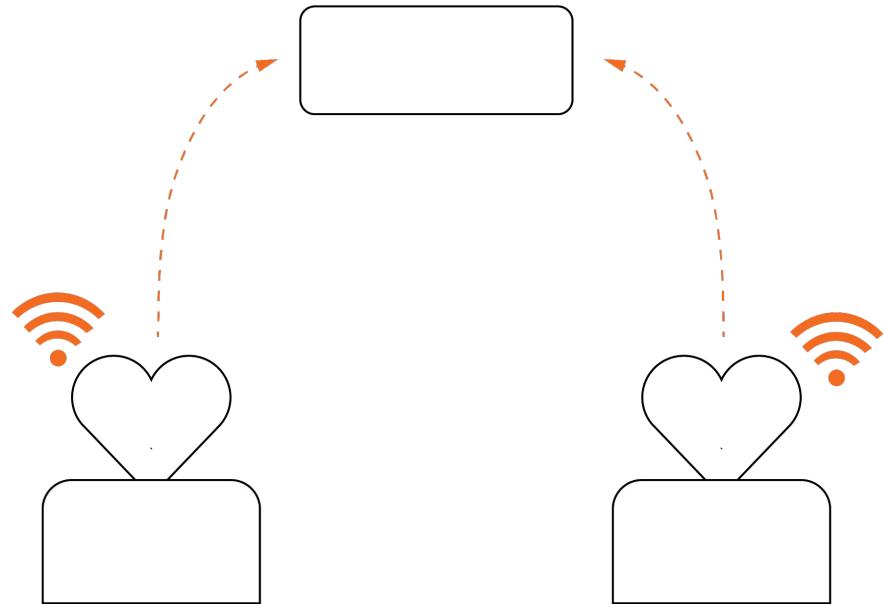
Main Code (runs in a loop):

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):



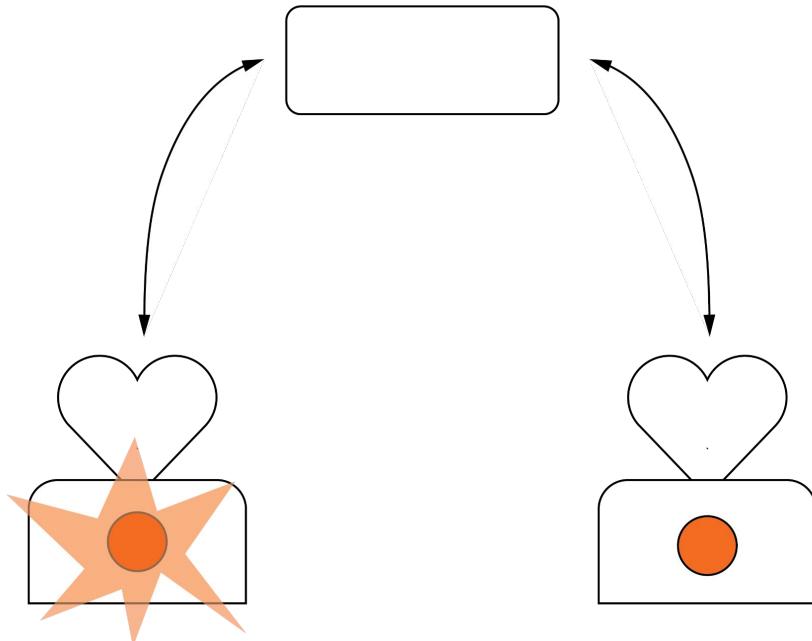
# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**



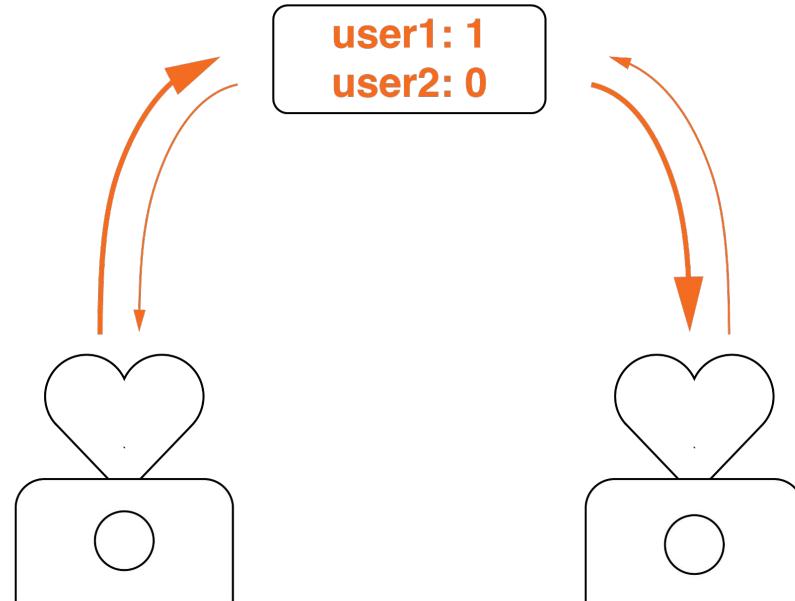
# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase



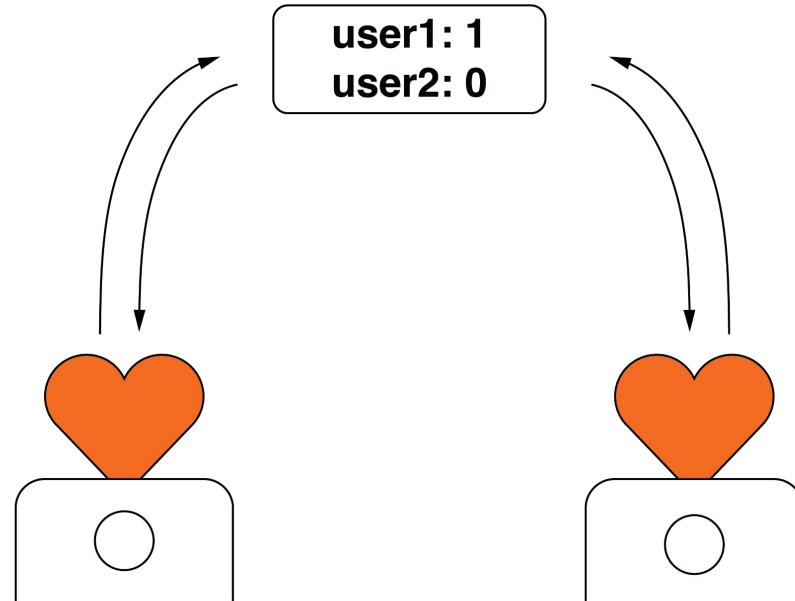
# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)



# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**



Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

# Arduino Code (C++)

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}
```

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

```
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
  
    }  
    delay(5);  
}
```

# Pseudo Code

Setup-code:

1. Connecting to **Wifi**
2. Connecting to **Firebase**

Main Code (runs in a loop):

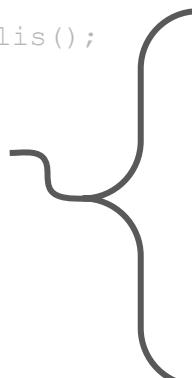
1. Reading the **Button State**
2. **Uploading** Data to Firebase
3. **Downloading** Data from Firebase
4. Manage **LED** (turn on/off)

# Arduino Code (C++)

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK && (millis() -  
lastFirebaseUpdate > 2000)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState = digitalRead(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        manageLED(buttonState, firebaseData);  
    }  
    delay(5);  
}
```

# Arduino Code

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK &&  
(millis() - lastFirebaseUpdate > 100))  
    {  
        lastFirebaseUpdate = millis();  
  
        buttonState =  
digitalRead(buttonPin);  
uploadData(buttonState);  
downloadData();  
ManageLED(buttonState,  
firebaseData);  
  
    }  
}
```



```
void uploadData(int buttonstate) {  
    if (Firebase.RTDB.setInt(&fbdo, "/user_1",  
buttonstate)) {  
        Serial.printf("Data UPLOAD successful, ");  
    } else {  
        Serial.println("Data UPLOAD failed, ");  
    }  
}
```

# Arduino Code

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK &&  
(millis() - lastFirebaseUpdate > 100))  
    {  
        lastFirebaseUpdate = millis();  
  
        buttonState =  
digitalRead(buttonPin);  
uploadData(buttonState);  
downloadData();  
ManageLED(buttonState,  
firebaseData);  
  
    }  
}
```

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

# Arduino Code

```
void setup() {  
    Serial.begin(115200);  
  
    pinMode(ledPin, OUTPUT);  
    pinMode(buttonPin, INPUT_PULLUP);  
  
    connectWiFi();  
    connectFirebase();  
}  
  
void loop() {  
    if (Firebase.ready() && signupOK &&  
        (millis() - lastFirebaseUpdate > 100)) {  
        lastFirebaseUpdate = millis();  
  
        buttonState =  
        digitalWrite(buttonPin);  
        uploadData(buttonState);  
        downloadData();  
        ManageLED(buttonState,  
        firebaseData);  
    }  
}
```

```
void manageLED(int buttonState, int  
firebaseData) {  
    if (buttonState == LOW or firebaseData ==  
LOW) {  
        digitalWrite(ledPin, HIGH);  
        delay(2000);  
        digitalWrite(ledPin, LOW);  
    } else {  
        digitalWrite(ledPin, LOW);  
    }  
}
```

# Some last bits

```
#include "addons	TokenNameHelper.h"  
#include "addons/RTDBHelper.h"  
  
#define WIFI_SSID "Insert SSID"  
#define WIFI_PASSWORD "Insert Wifi Password"  
  
#define API_KEY "Insert API Key"  
#define DATABASE_URL "Insert Database URL"  
  
const int ledPin = 2;  
const int buttonPin = 12;  
  
bool firebaseData = false;  
int buttonState = 1;  
  
FirebaseData fbdo;  
FirebaseAuth auth;  
FirebaseConfig config;  
unsigned long lastFirebaseUpdate = 0;  
int count = 0;  
bool signupOK = false;
```

**Wi-Fi credentials:**

Name and Password of current WiFi

# Some last bits

```
#include "addons	TokenNameHelper.h"
#include "addons/RTDBHelper.h"

#define WIFI_SSID "Insert SSID"
#define WIFI_PASSWORD "Insert Wifi Password"

#define API_KEY "Insert API Key"
#define DATABASE_URL "Insert Database URL"

const int ledPin = 2;
const int buttonPin = 12;

bool firebaseData = false;
int buttonState = 1;

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long lastFirebaseUpdate = 0;
int count = 0;
bool signupOK = false;
```

## Firebase Credentials:

They need to be adjusted to **your own unique API Key** and **Database URL**

# **Let's Upload the Code!**

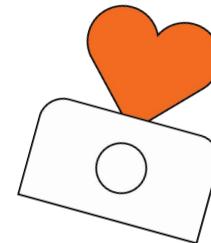
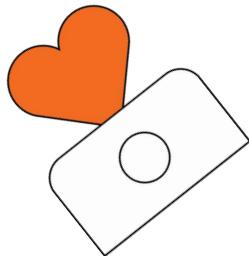


Access Code from Github:

<https://github.com/juliadaser/Siggraph-Workshop>

Time for...  
**Troubleshooting**

# Help us fill out this feedback form!



# Open to Questions!

SIGGRAPH Experience Labs

 @wormicollective

**Access the code:**

[github.com/juliadaser/Siggraph-Workshop](https://github.com/juliadaser/Siggraph-Workshop)

**Contact us:**

yiqing.ng@gmail.com

