

[SUCCEEDED] CAPSTONE WASTE CAN DEVELOPMENT
JUNE 26

After a week of trying to do serial communication between Arduino Uno and ESP 32 CAM, it finally worked.

Huge Help: <https://www.aranacorp.com/en/communicate-with-arduino/#:~:text=It%20is%20possible%20to%20communicate,The%20SoftwareSerial.>

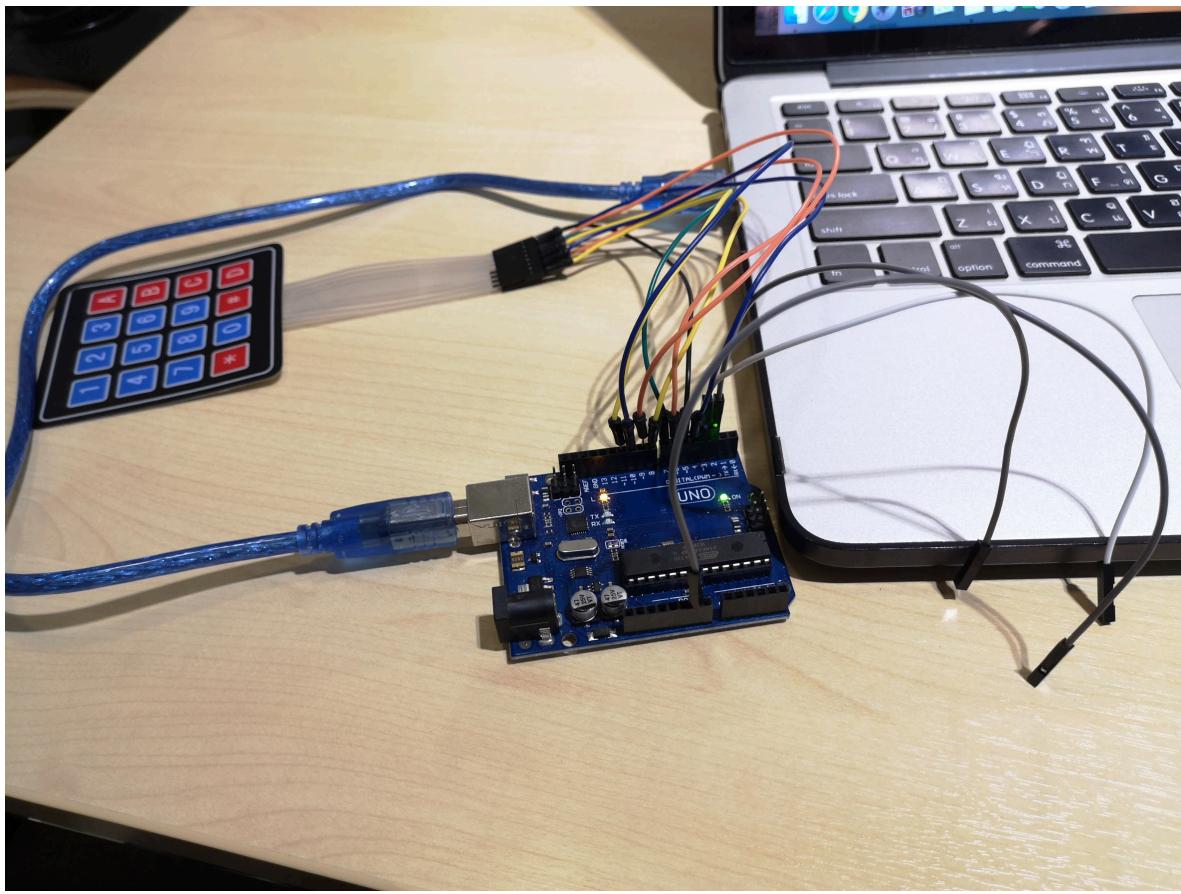
1. Connected keypad to Arduino and uploaded the code

Schematic:

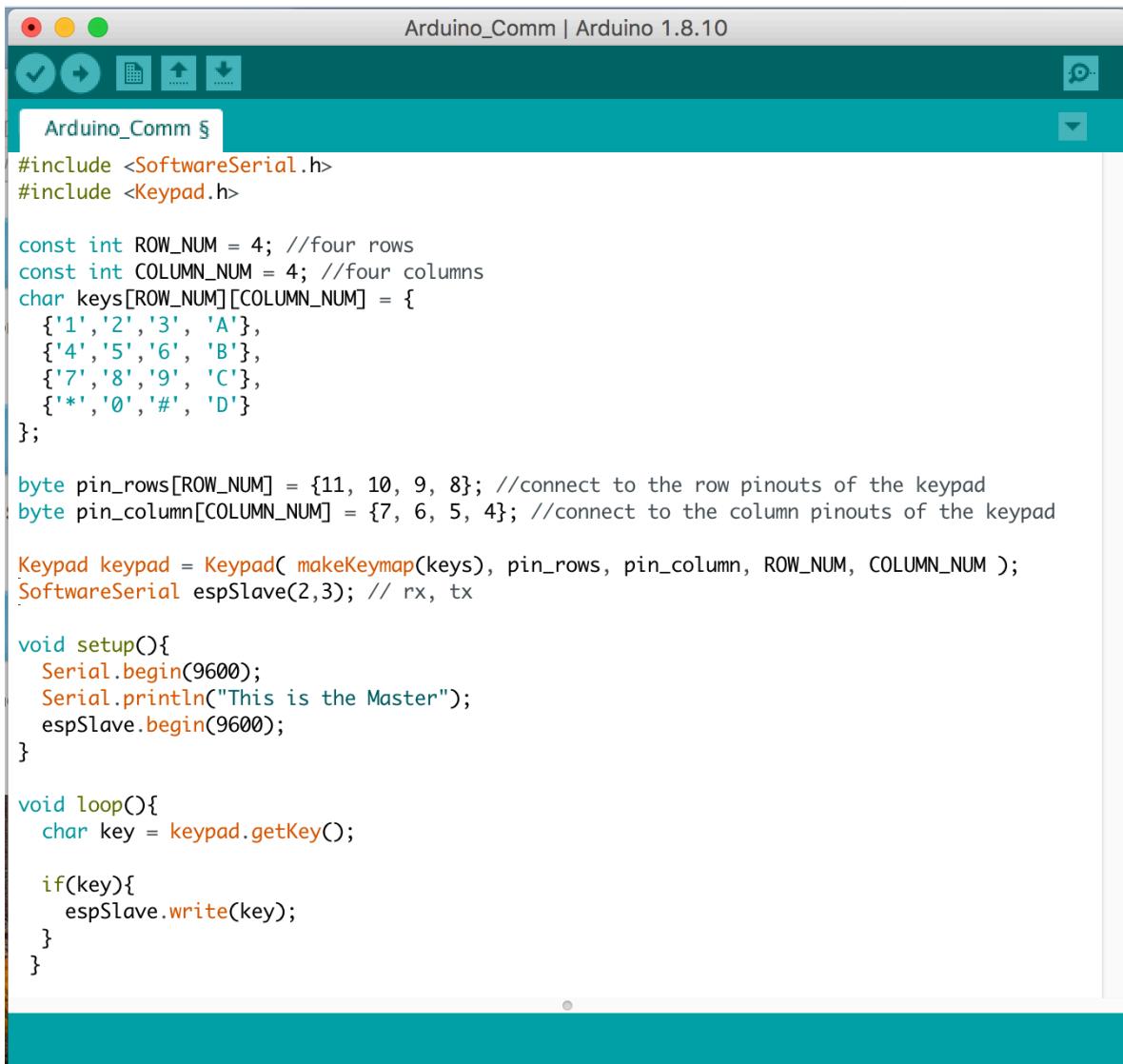


R1 to C4 matches with Arduino pins 11 - 4, in order (ONLY THING CONNECTED WHILE UPLOADING)

When uploading code to Arduino, pins 2 and 3 (rx and tx, respectively) and GND are disconnected from ESP32 CAM



Code: (Remember to select the right board and port!)



The screenshot shows the Arduino IDE interface with the title bar "Arduino_Comm | Arduino 1.8.10". The main window displays the following C++ code:

```
#include <SoftwareSerial.h>
#include <Keypad.h>

const int ROW_NUM = 4; //four rows
const int COLUMN_NUM = 4; //four columns
char keys[ROW_NUM][COLUMN_NUM] = {
    {'1','2','3', 'A'},
    {'4','5','6', 'B'},
    {'7','8','9', 'C'},
    {'*','0','#', 'D'}
};

byte pin_rows[ROW_NUM] = {11, 10, 9, 8}; //connect to the row pinouts of the keypad
byte pin_column[COLUMN_NUM] = {7, 6, 5, 4}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM );
SoftwareSerial espSlave(2,3); // rx, tx

void setup(){
    Serial.begin(9600);
    Serial.println("This is the Master");
    espSlave.begin(9600);
}

void loop(){
    char key = keypad.getKey();

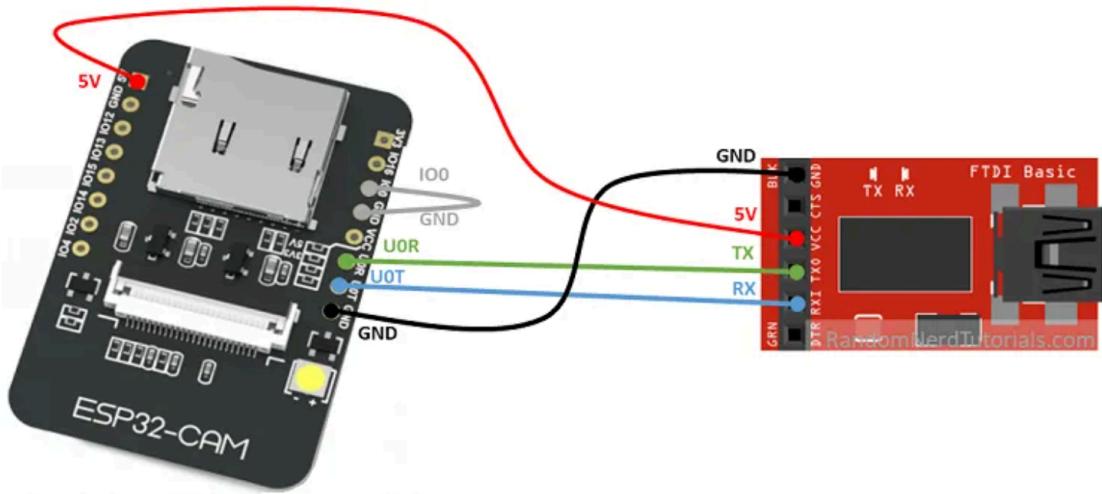
    if(key){
        espSlave.write(key);
    }
}
```

When any key in the keypad is pressed, data will be written (sent) to the ESP32 Cam (the espSlave is the object where serial is communicated to through Arduino pins 2 and 3).

2. Connect ESP32 CAM to FTDI Programmer and LED

Schematic:

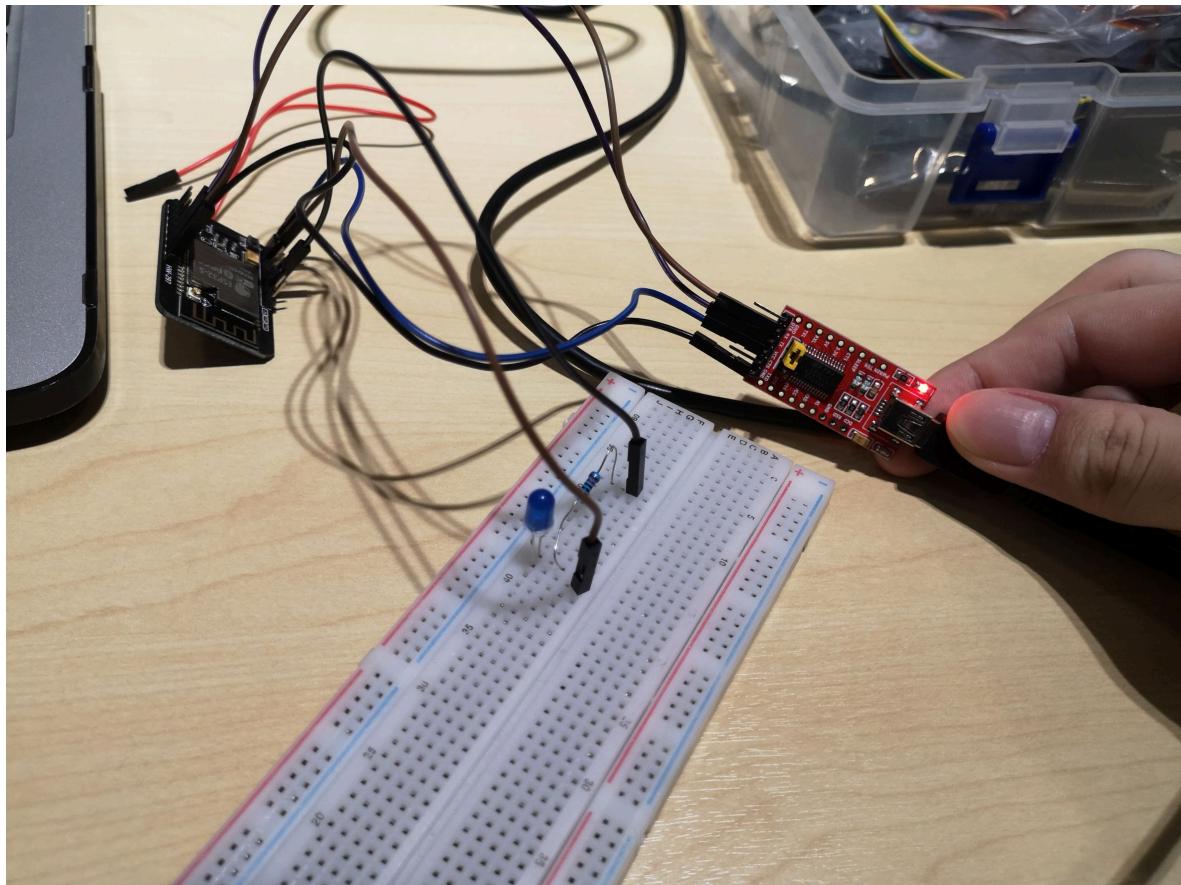
ESP32 to FTDI Programmer:



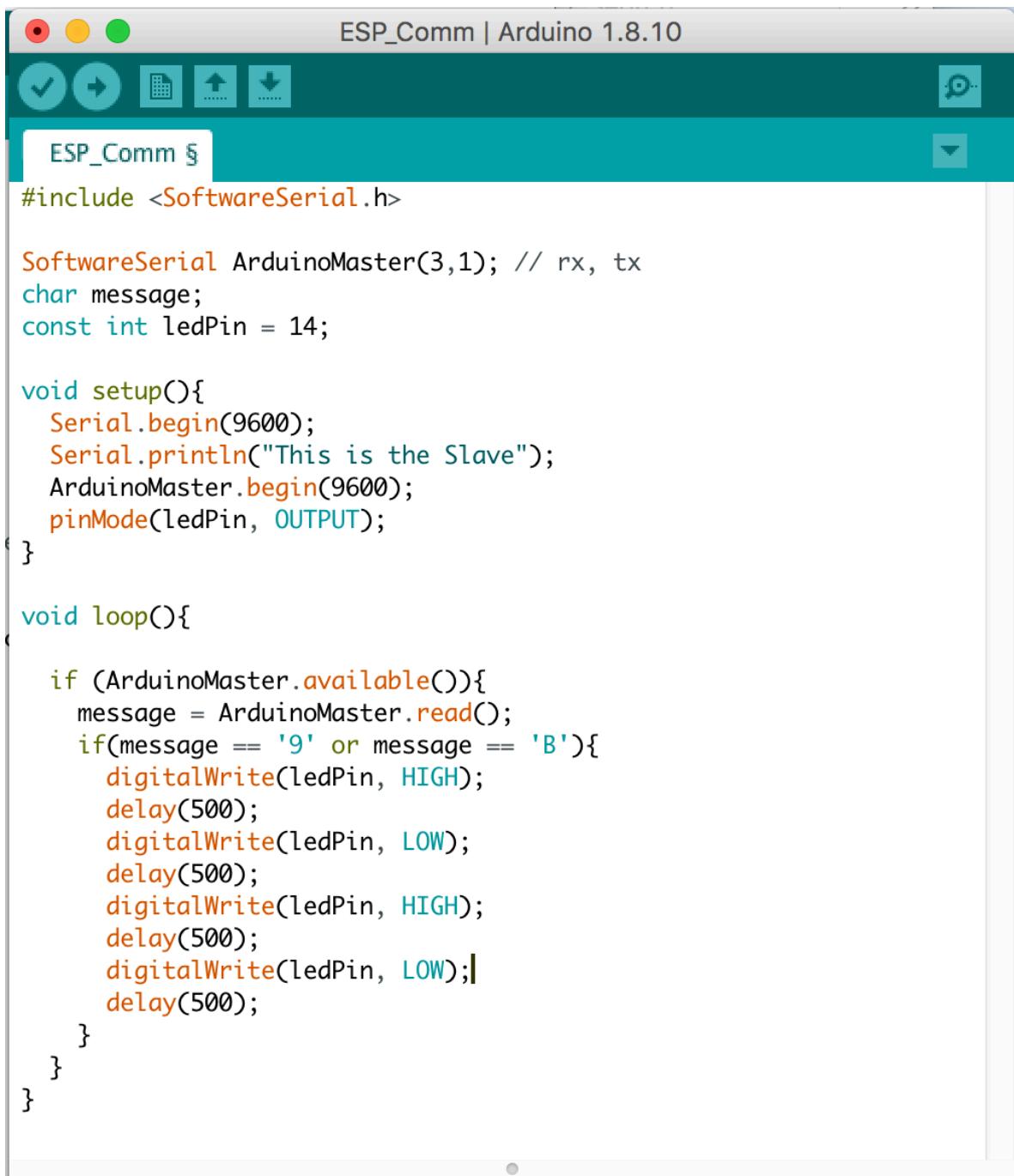
ESP 32 to LED:

- + added **GND** (ESP32 CAM) to negative pin of LED and **GPIO 14** (ESP32 CAM) to positive pin of LED

(NOTHING CONNECTED TO ARDUINO WHILE UPLOADING CODE!)



Code:



The screenshot shows the Arduino IDE interface with the title bar "ESP_Comm | Arduino 1.8.10". The toolbar contains standard icons for file operations and a refresh button. The code editor window displays the following sketch:

```
#include <SoftwareSerial.h>

SoftwareSerial ArduinoMaster(3,1); // rx, tx
char message;
const int ledPin = 14;

void setup(){
  Serial.begin(9600);
  Serial.println("This is the Slave");
  ArduinoMaster.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop(){
  if (ArduinoMaster.available()){
    message = ArduinoMaster.read();
    if(message == '9' or message == 'B'){
      digitalWrite(ledPin, HIGH);
      delay(500);
      digitalWrite(ledPin, LOW);
      delay(500);
      digitalWrite(ledPin, HIGH);
      delay(500);
      digitalWrite(ledPin, LOW);
      delay(500);
    }
  }
}
```

When data is sent from Arduino (data is .available), the serial reads the message from the Arduino and stores it in message. The light blinks twice if the right key is pressed.

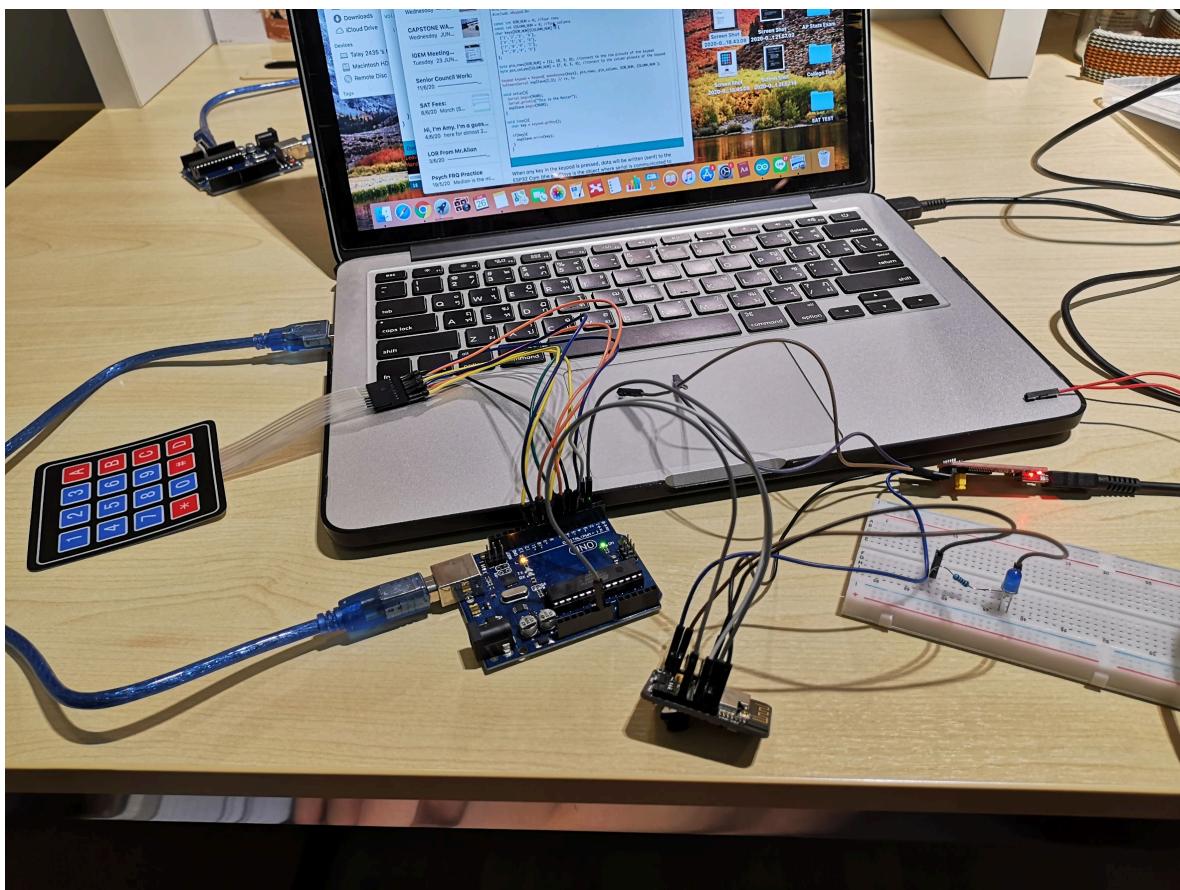
(While uploading code through FTDI, when it is connecting press reset on the ESP32 CAM.

When it is done uploading, remove GPIO0 and GND wire and press reset.)

3. Connecting ESP32 to Arduino to Initiate Communication

Once code is uploaded to both boards:

**Disconnect Tx and Rx from FTDI programmer.
Connect Tx and Rx of ESP32 to Rx and Tx of Arduino, respectively.
Connect GND (ESP32) to GND (Arduino).**



4. Works now, communication succeeds!

