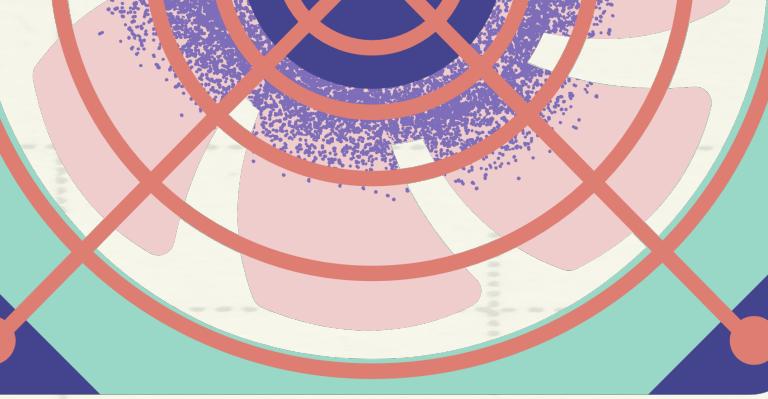
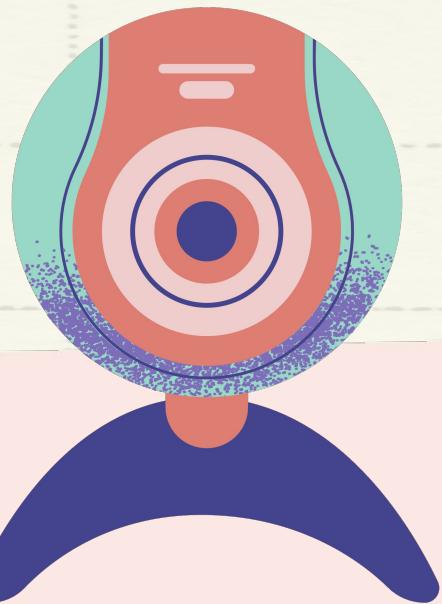
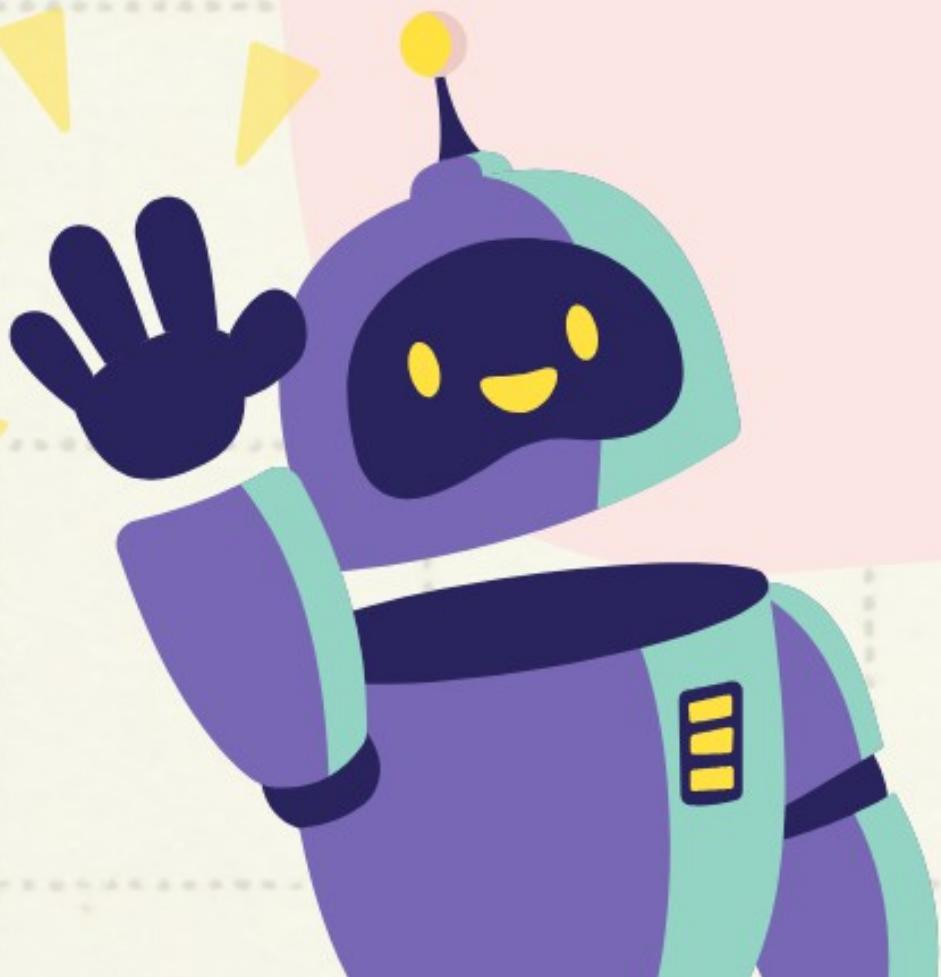


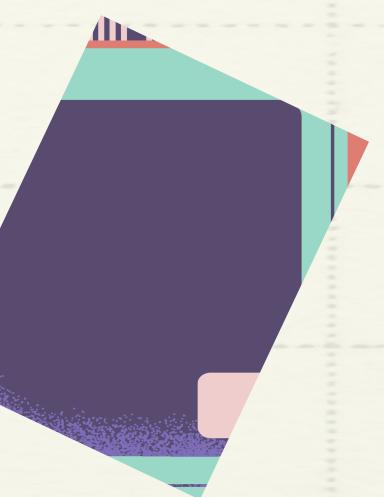
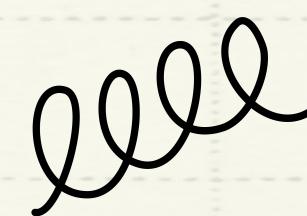
MUSIKBOX SELBER

BAUEN



DER

1. Einführung in die Arduino Programmierung
2. Elektronische Komponenten
3. Schaltplan und das zusammenstecken der
Musikbox
4. Die Logik hinter der Musikbox



DIE basics!



eee

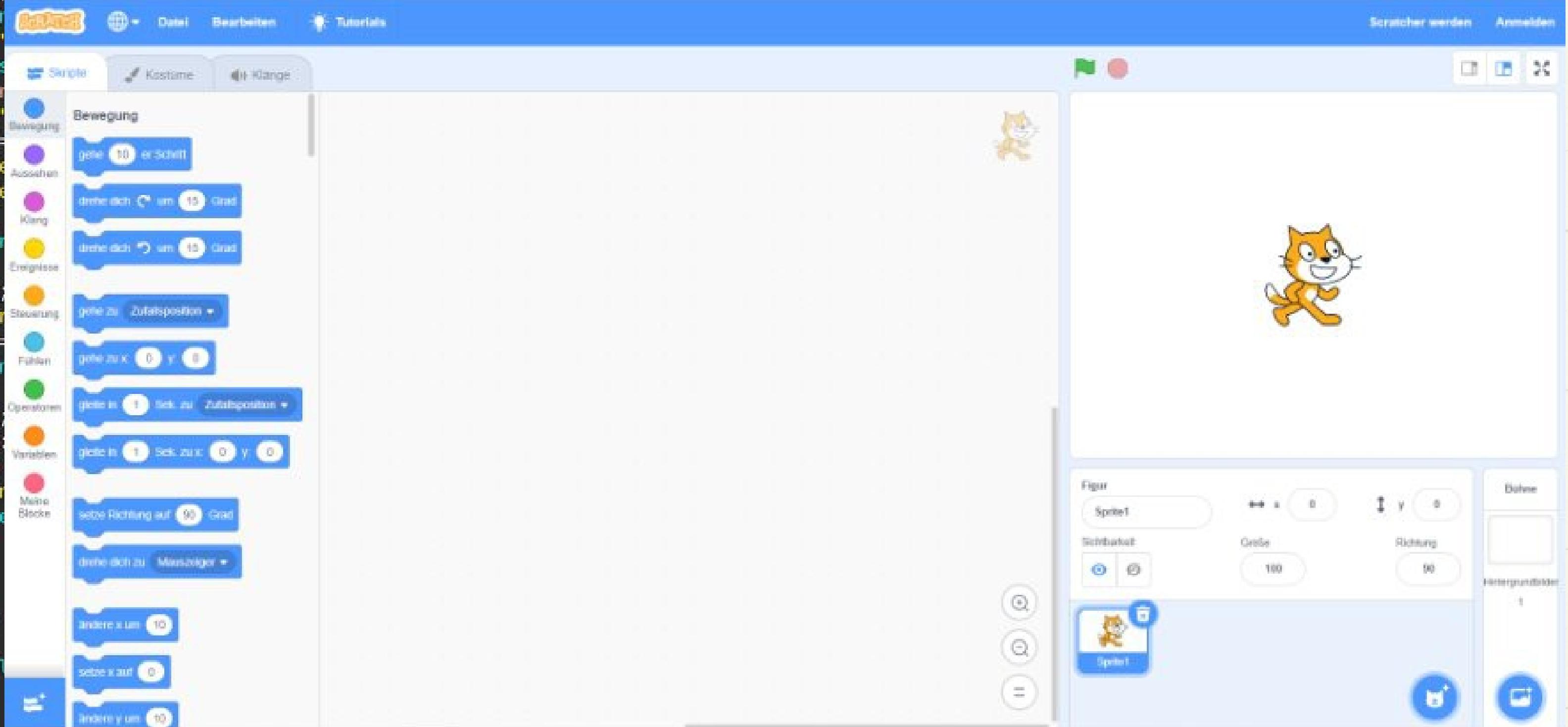
Was ist
eigentlich
Programmierung?

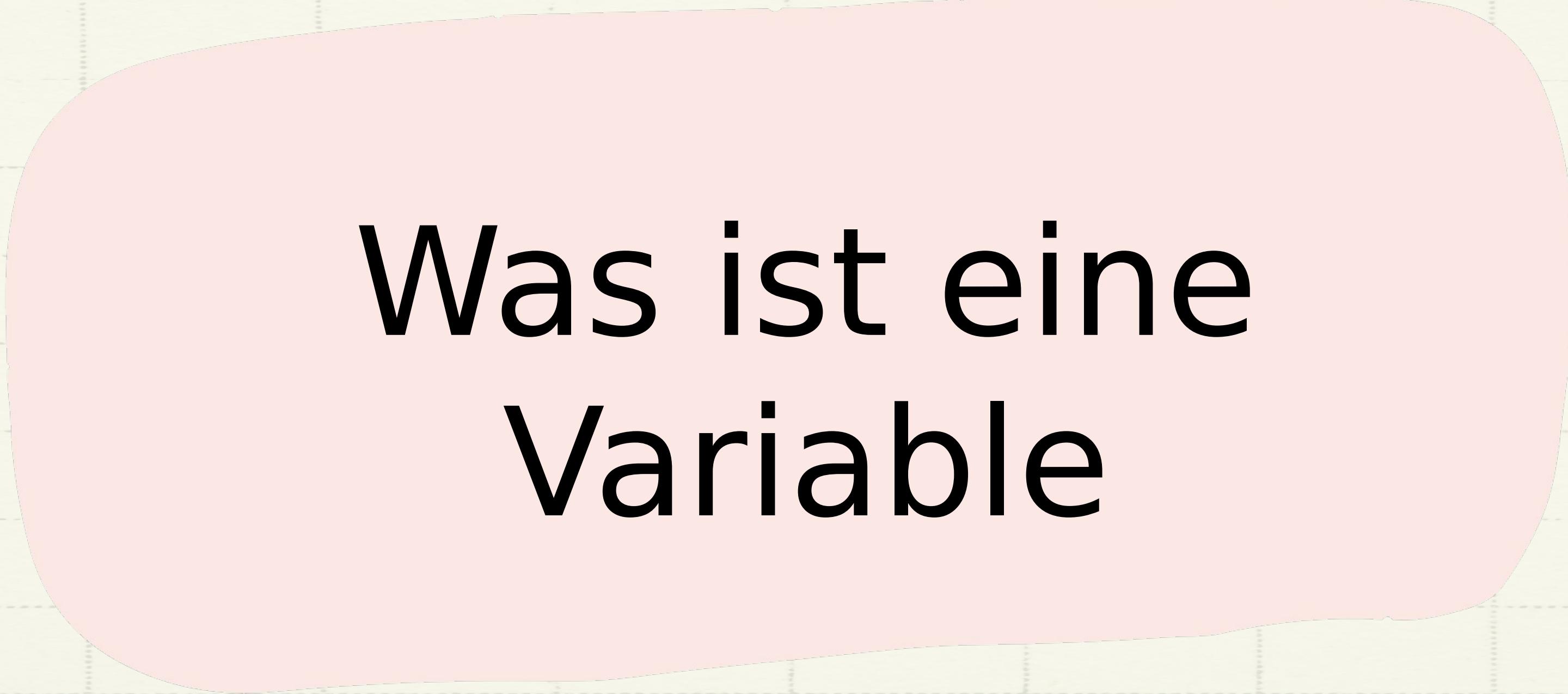
ell

```
public async Task<bool> CreateUser(UserInput input) {
    var validUserTypes = new[] { "regular", "premium", "trial" };

    User user;
    switch (input.userType)
    {
        case "regular":
            user = new RegularUser();
            break;
        case "premium":
            user = new PremiumUser();
            break;
        case "trial":
            user = new TrialUser();
            break;
        default:
            throw new ArgumentException($"Unsupported user type: {input.userType}");
    }

    bool result = await user.CreateUser(input.name);
    return result;
}
```



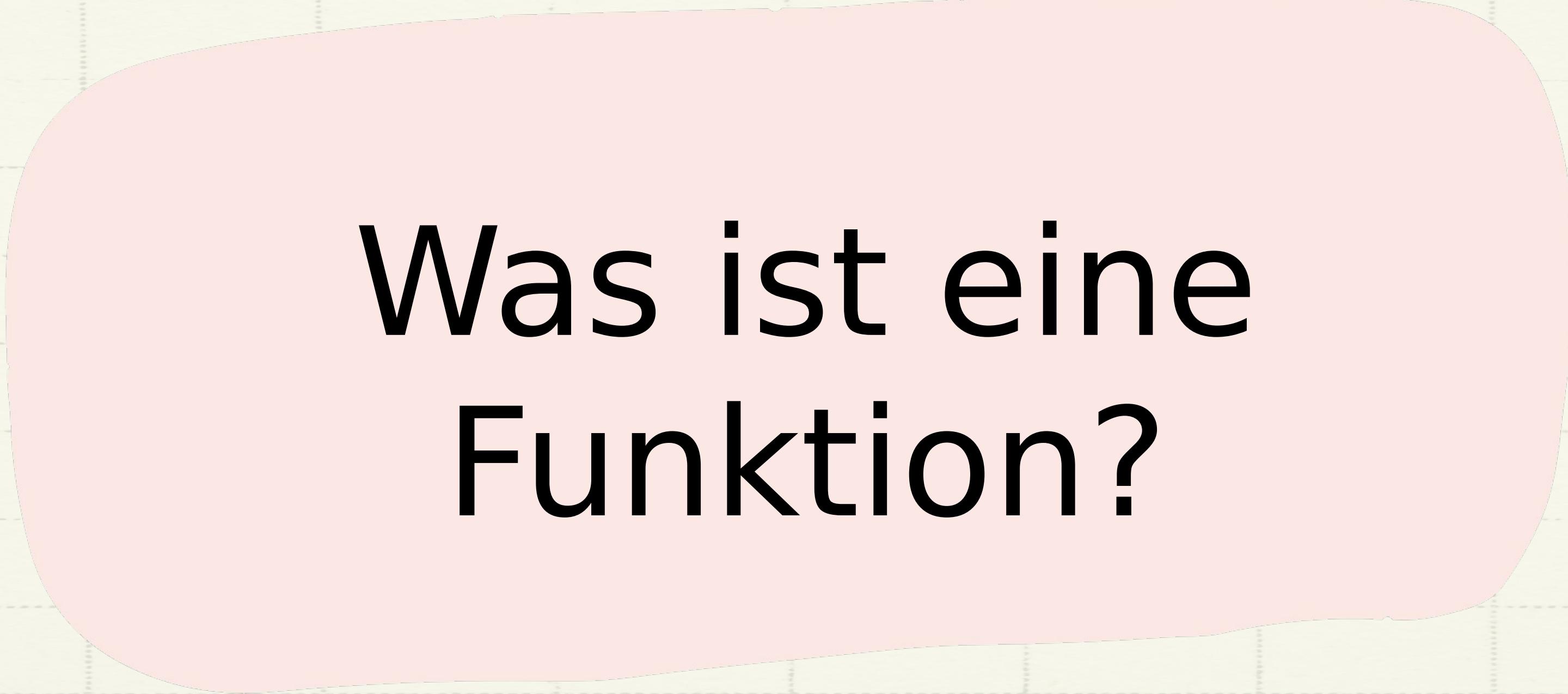


Was ist eine
Variable

Welt

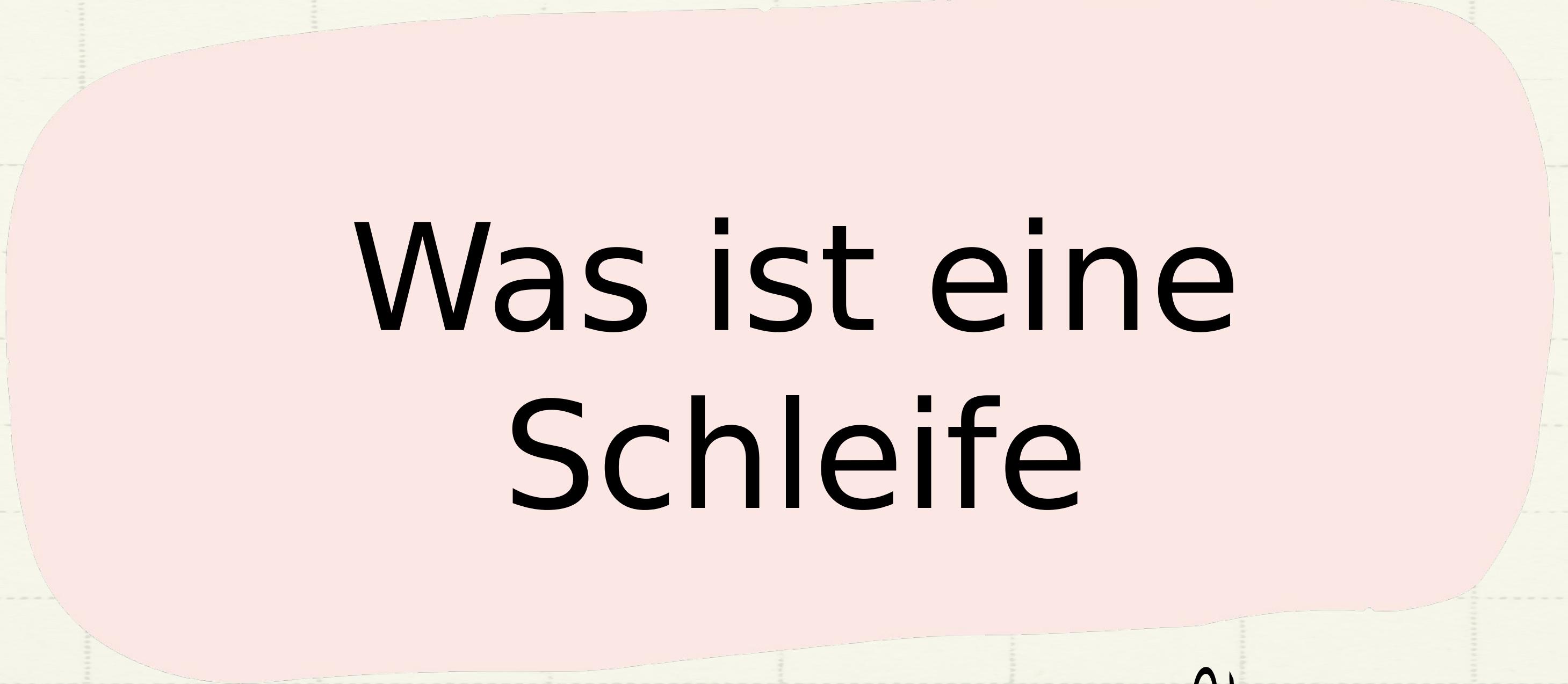
Welt

**Was ist eine
Bedingung**



**Was ist eine
Funktion?**

www



**Was ist eine
Schleife**



JULIA



Es werde Licht!

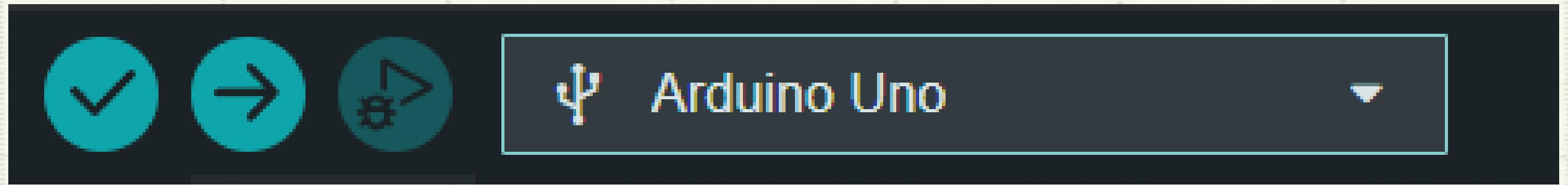
sketch_apr10a.ino

```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8  
9 }  
10
```

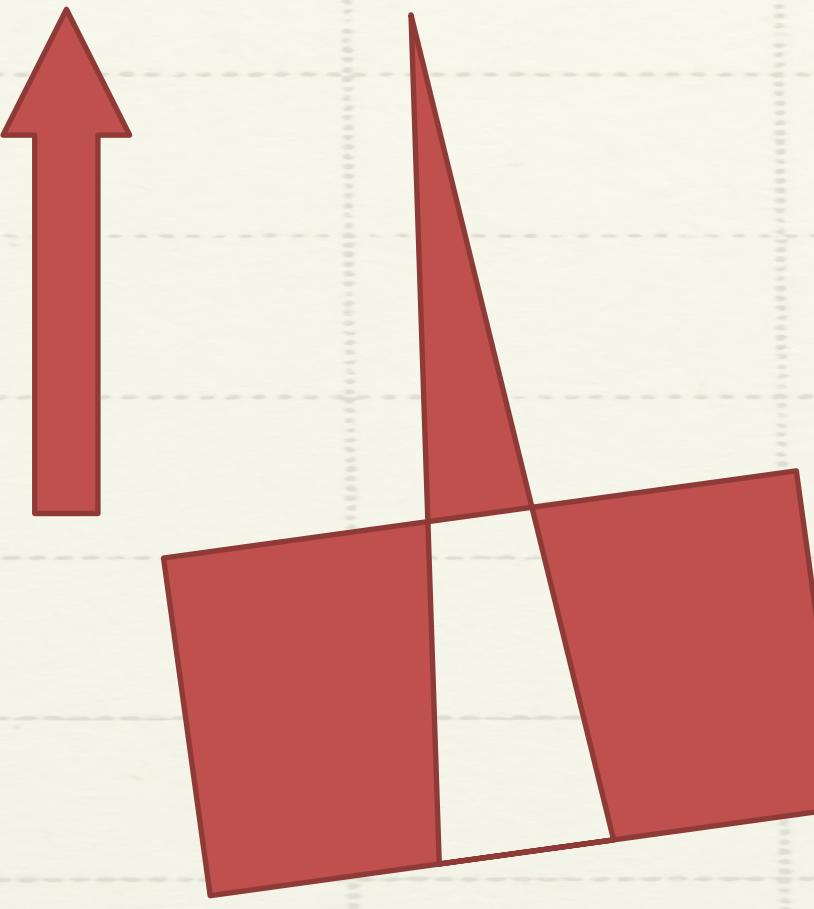
hell

Blink.ino

```
1 void setup() {
2     // Initialisiere den digitalen Pin LED_BUILTIN als Ausgang.
3     pinMode(LED_BUILTIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_BUILTIN, HIGH);    // Schalte die LED an
8     delay(1000);                      // Warte eine Sekunde
9     digitalWrite(LED_BUILTIN, LOW);    // Schalte die LED aus
10    delay(1000);                     // Warte eine Sekunde
11 }
```

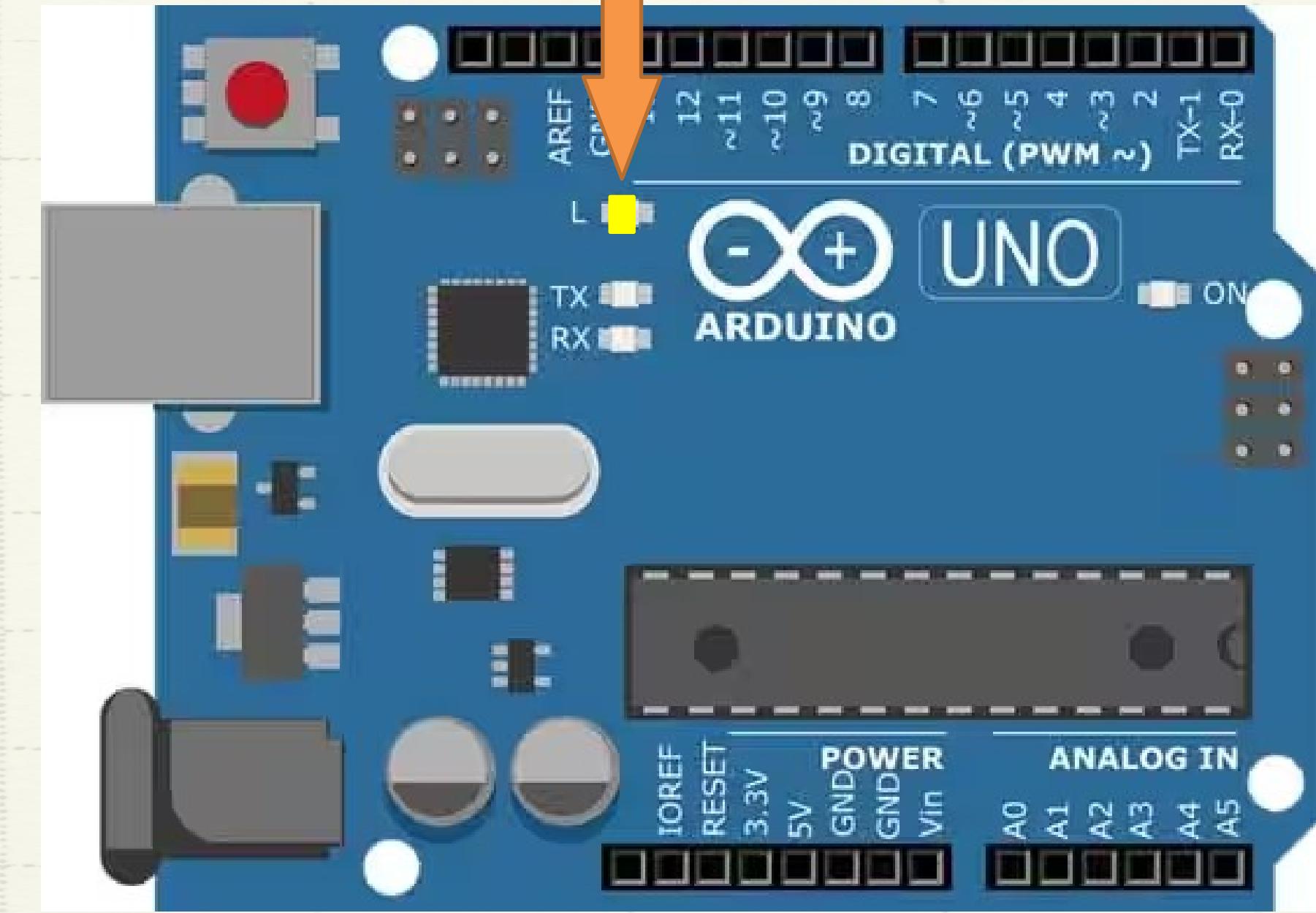


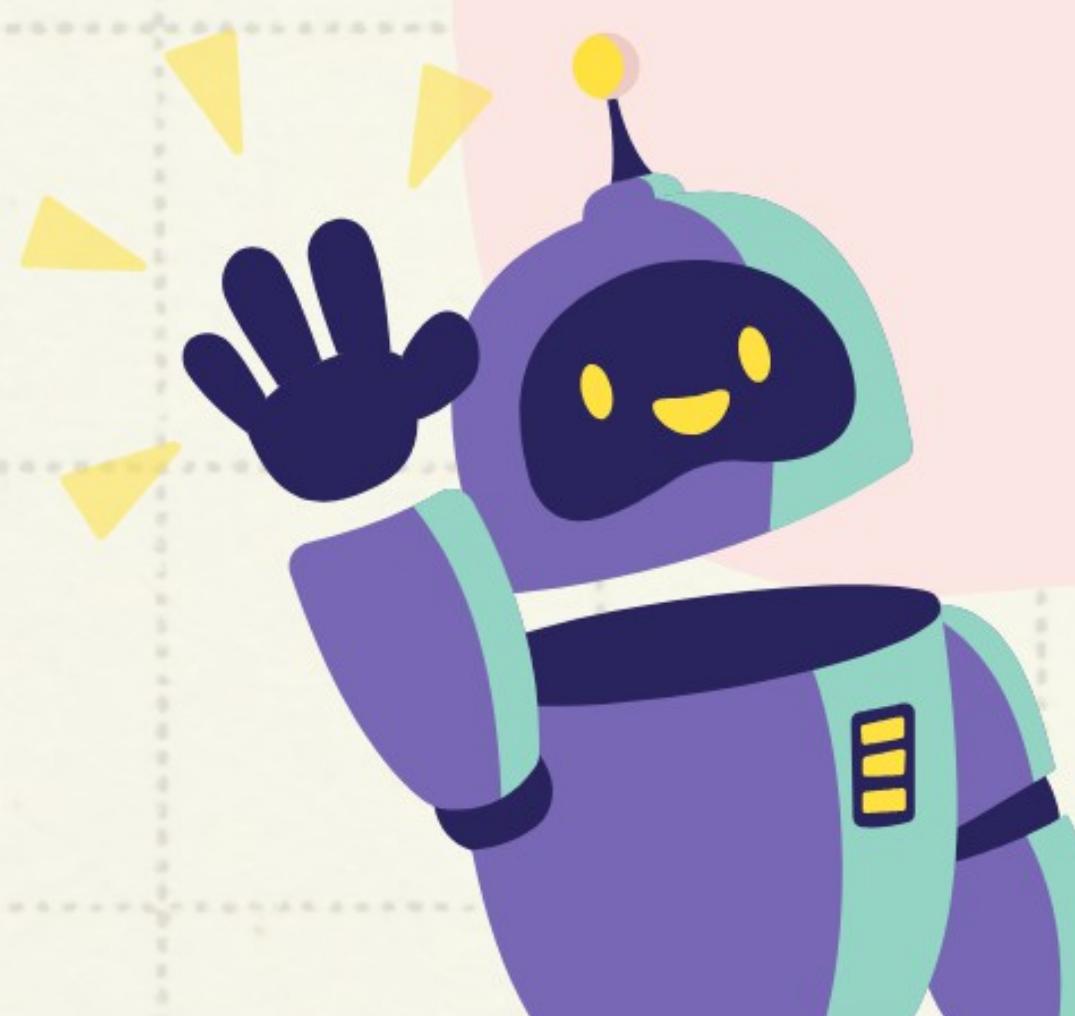
eee



Output

```
Der Sketch verwendet 924 Bytes (2%) des Programmspeicherplatzes. Das Maximum sind 32256 Bytes.  
Globale Variablen verwenden 9 Bytes (0%) des dynamischen Speichers, 2039 Bytes für lokale Variablen verbleiben. Das Maximum sind 2048 Bytes.  
"C:\Users\gceli_0blna9v\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/bin/avrdude" "-CC:C:\Users\gceli_0blna9v\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/etc/avrdude.conf" -v  
  
avrdude: Version 6.3-20190619  
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/  
Copyright (c) 2007-2014 Joerg Wunsch  
  
System wide configuration file is "C:\Users\gceli_0blna9v\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/etc/avrdude.conf"  
  
Using Port : COM6  
Using Programmer : arduino  
Overriding Baud Rate : 115200  
AVR Part : ATmega328P  
Chip Erase delay : 9000 us  
PAGEL : PD7  
BS2 : PC2  
RESET disposition : dedicated  
RETRY pulse : SCK  
serial program mode : yes  
parallel program mode : yes  
Timeout : 200  
StabDelay : 100  
CmdexeDelay : 25  
SyncLoops : 32  
ByteDelay : 0  
PollIndex : 3  
PollValue : 0x53  
Memory Detail :  
  
          Block Poll      Page      Polled  
Memory Type Mode Delay Size Indx Paged  Size  Size #Pages MinW MaxW ReadBack  
----- ----- ---- ----- ---- ----- ----- ----- ----- ----- -----  
eeprom  65  20    4    0 no    1024   4    0  3600 3600 0xff 0xff  
flash   65    6  128    0 yes   32768 128   256 4500 4500 0xff 0xff  
lfuse    0    0    0    0 no      1    0    0 4500 4500 0x00 0x00  
hfuse    0    0    0    0 no      1    0    0 4500 4500 0x00 0x00  
efuse    0    0    0    0 no      1    0    0 4500 4500 0x00 0x00  
lock     0    0    0    0 no      1    0    0 4500 4500 0x00 0x00  
calibration  0    0    0    0 no      1    0    0    0 0x00 0x00
```



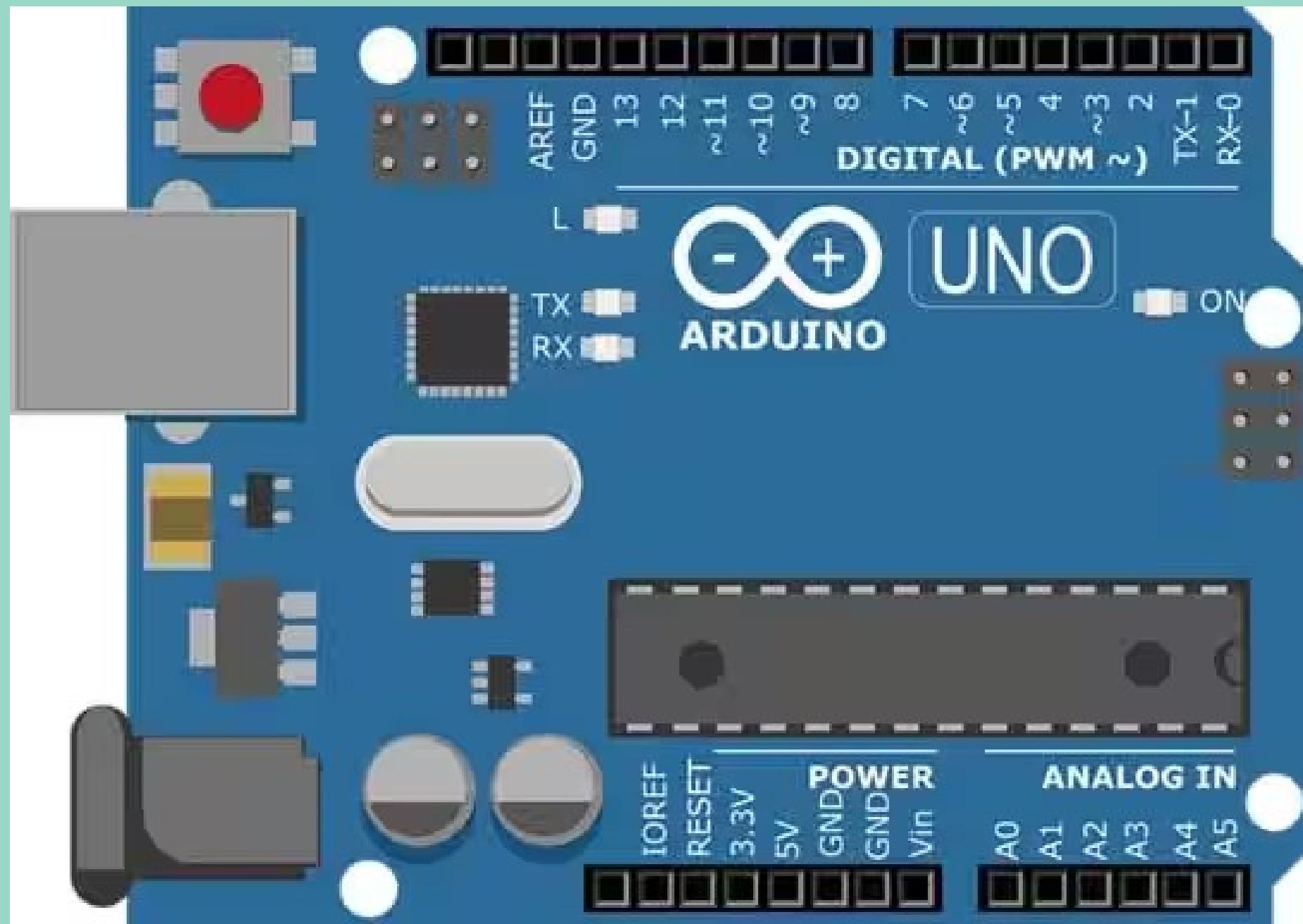


Breadboard, Kabel,
Lichtsensor...
Was ist das alles?

eee

Arduino Uno

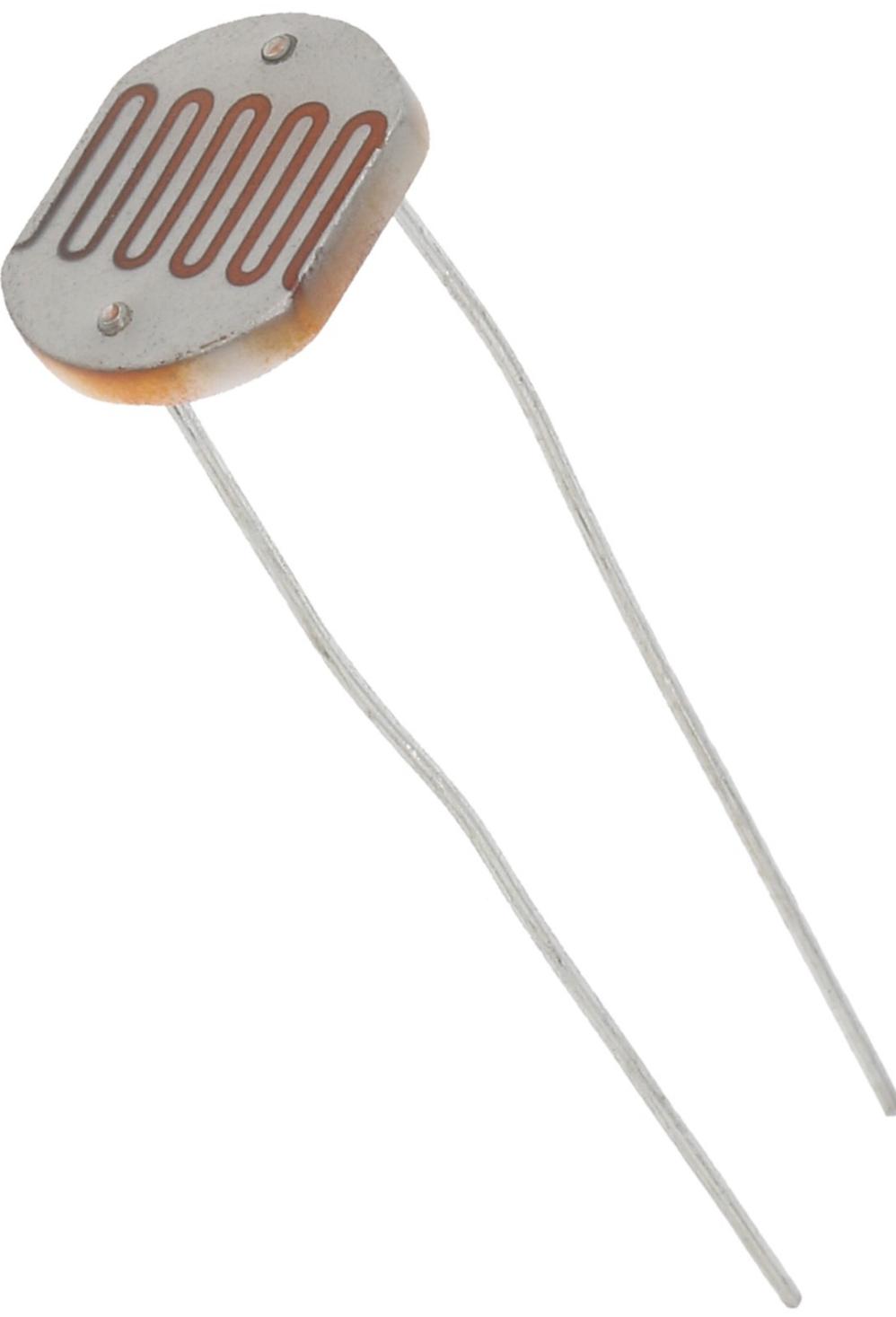




eee

Lichtsensor



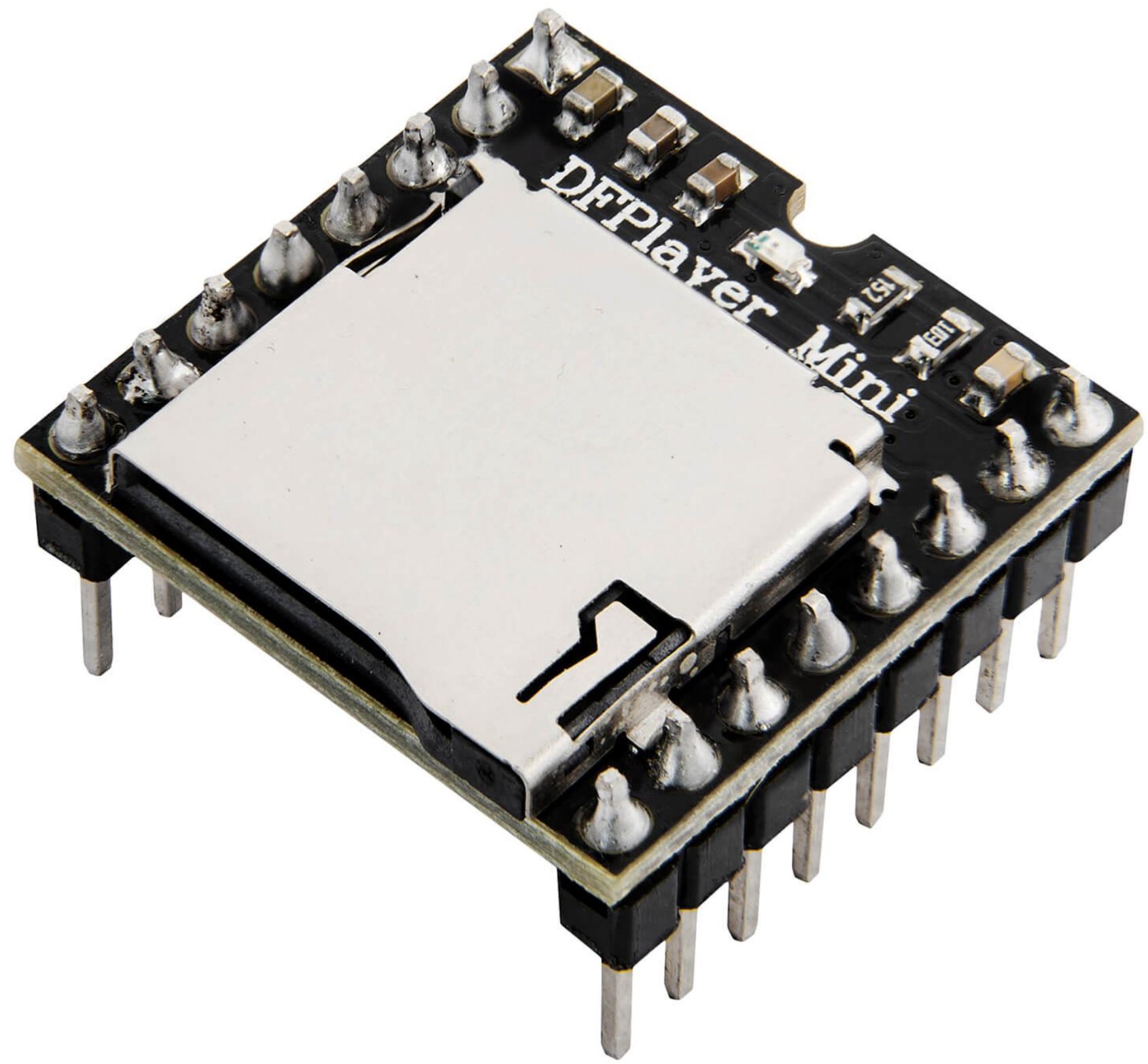


ell

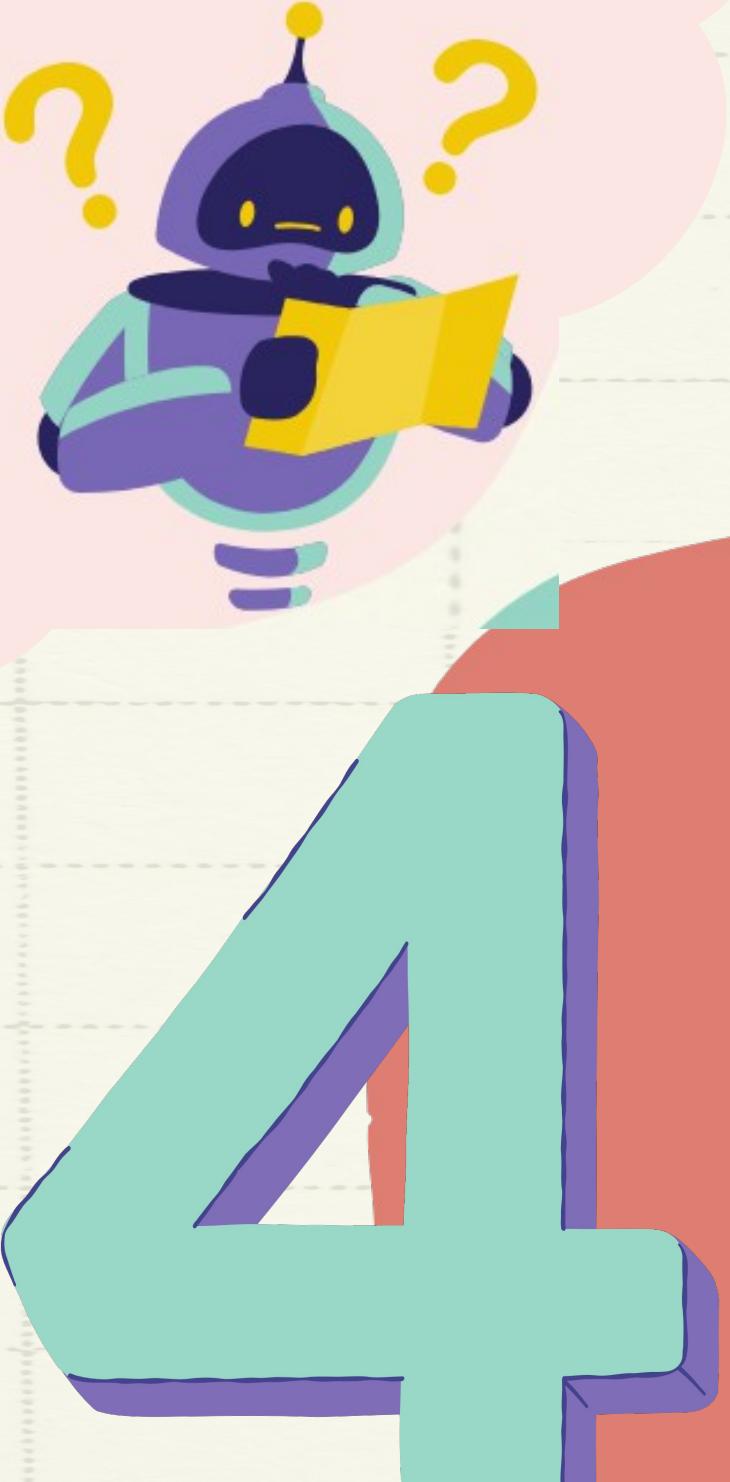


DF Player Mini Aka MP3 Player

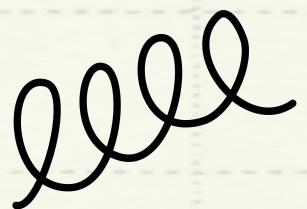
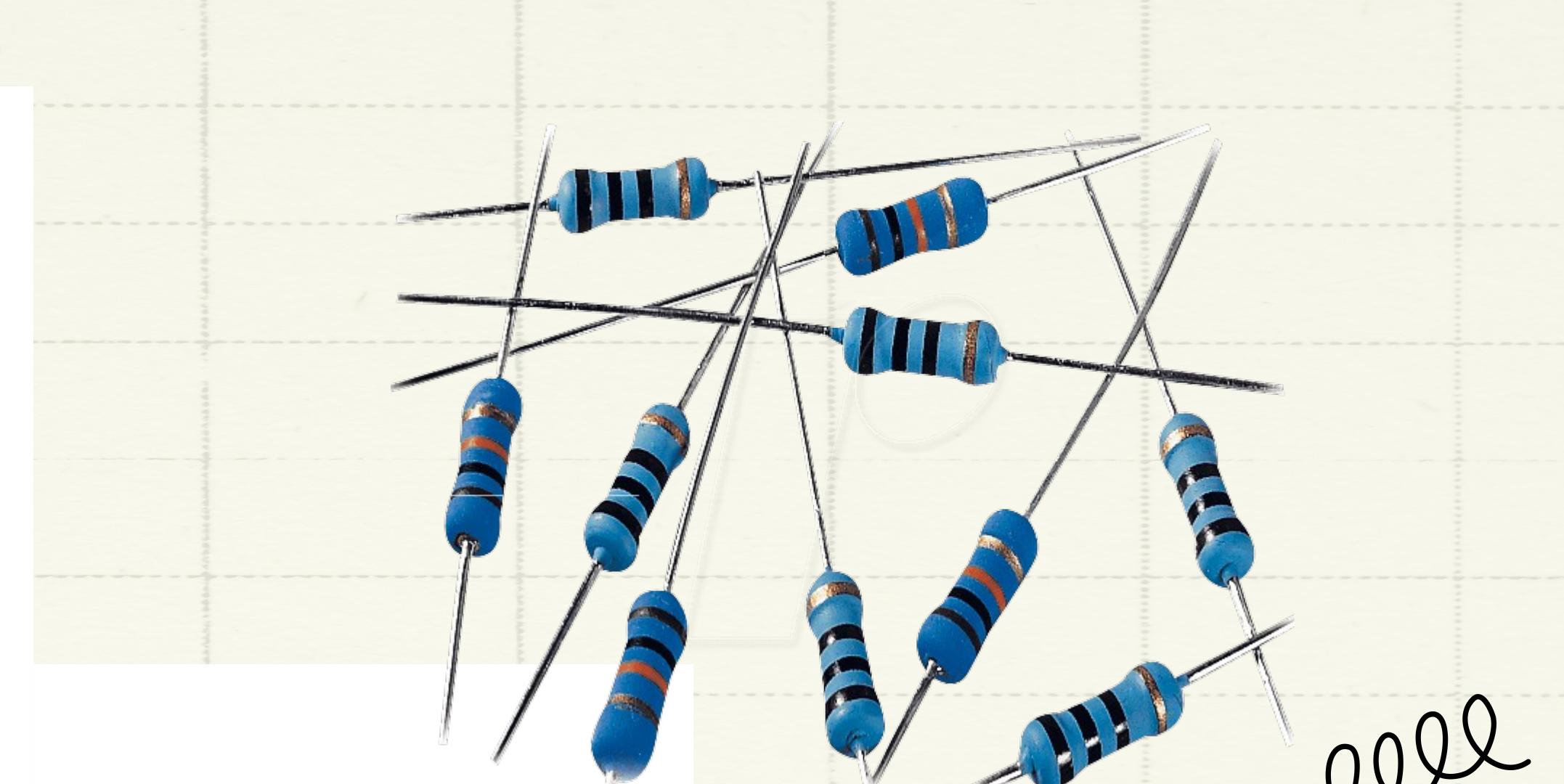




eee



Jumperkabel, Widerstände und Breadboard



Stromquelle und die BOX

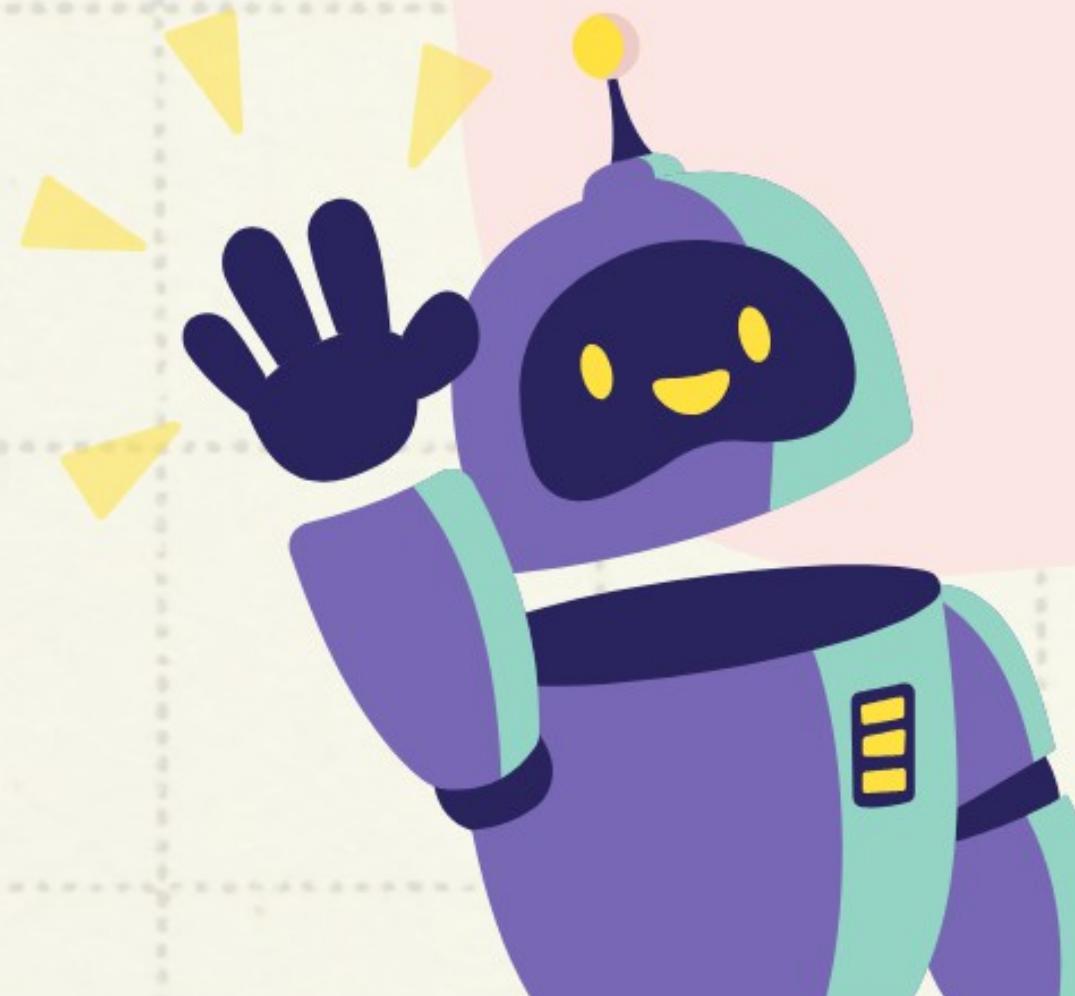




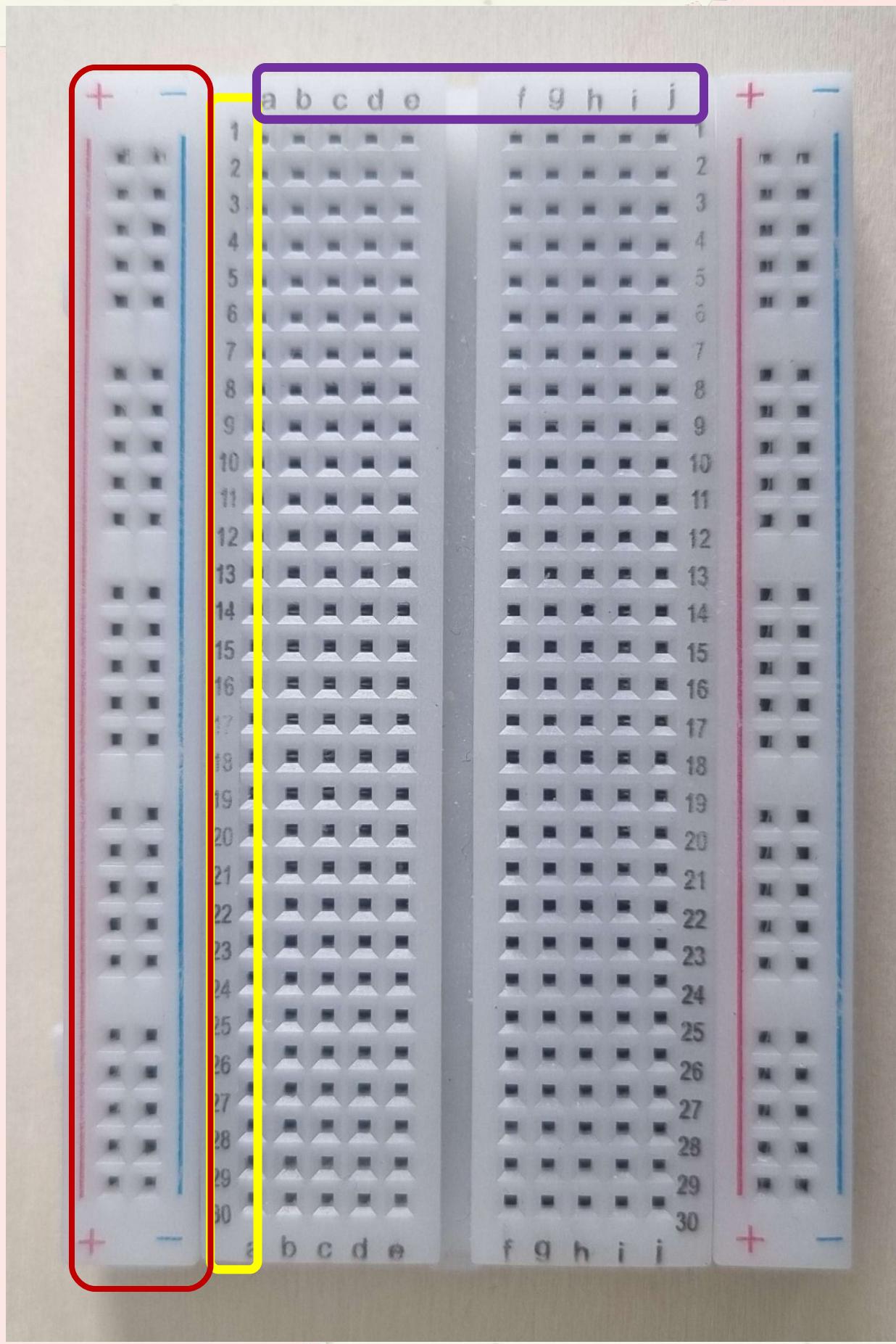
eee

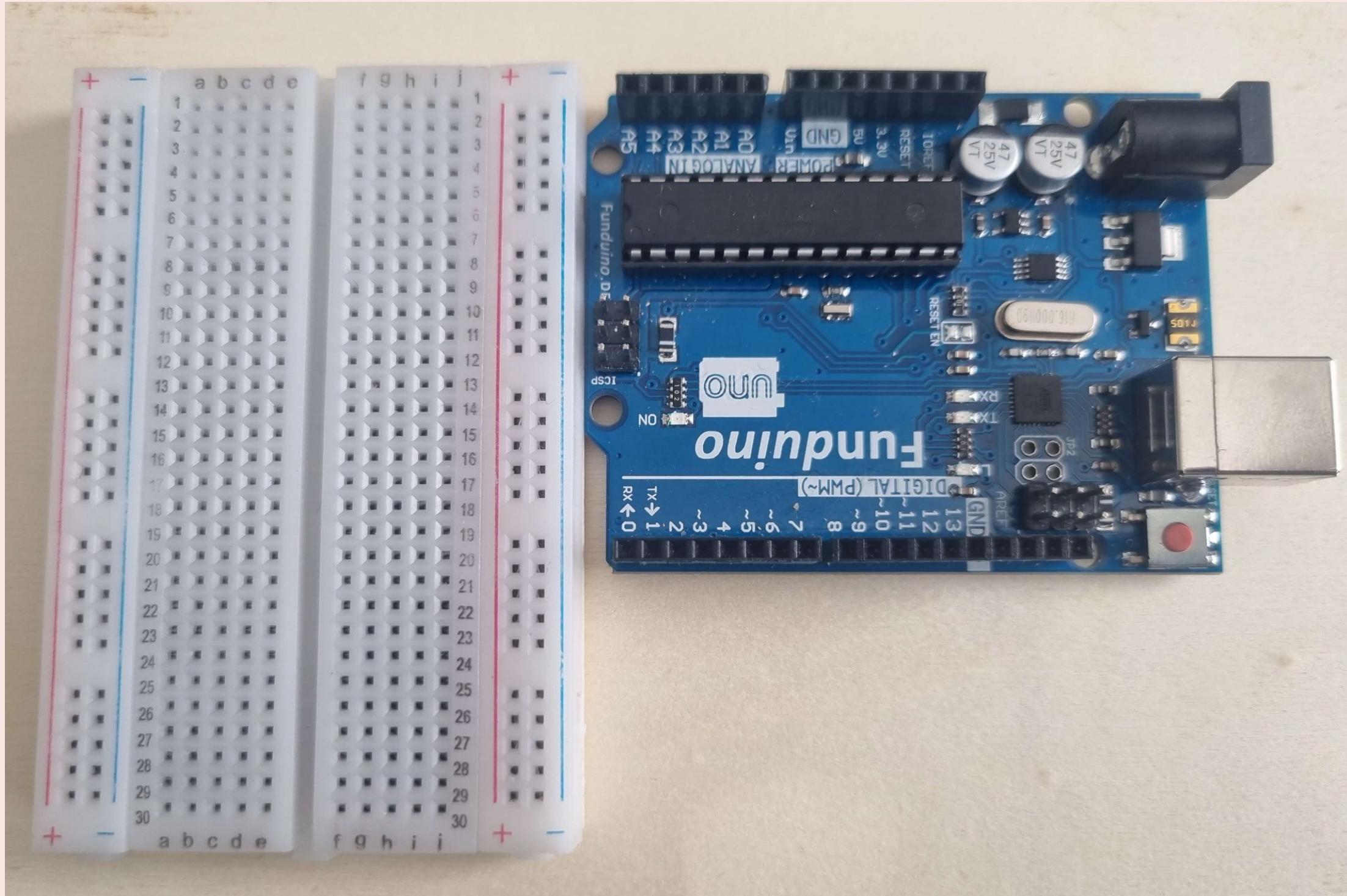


Schaltplan und Steckanleitung

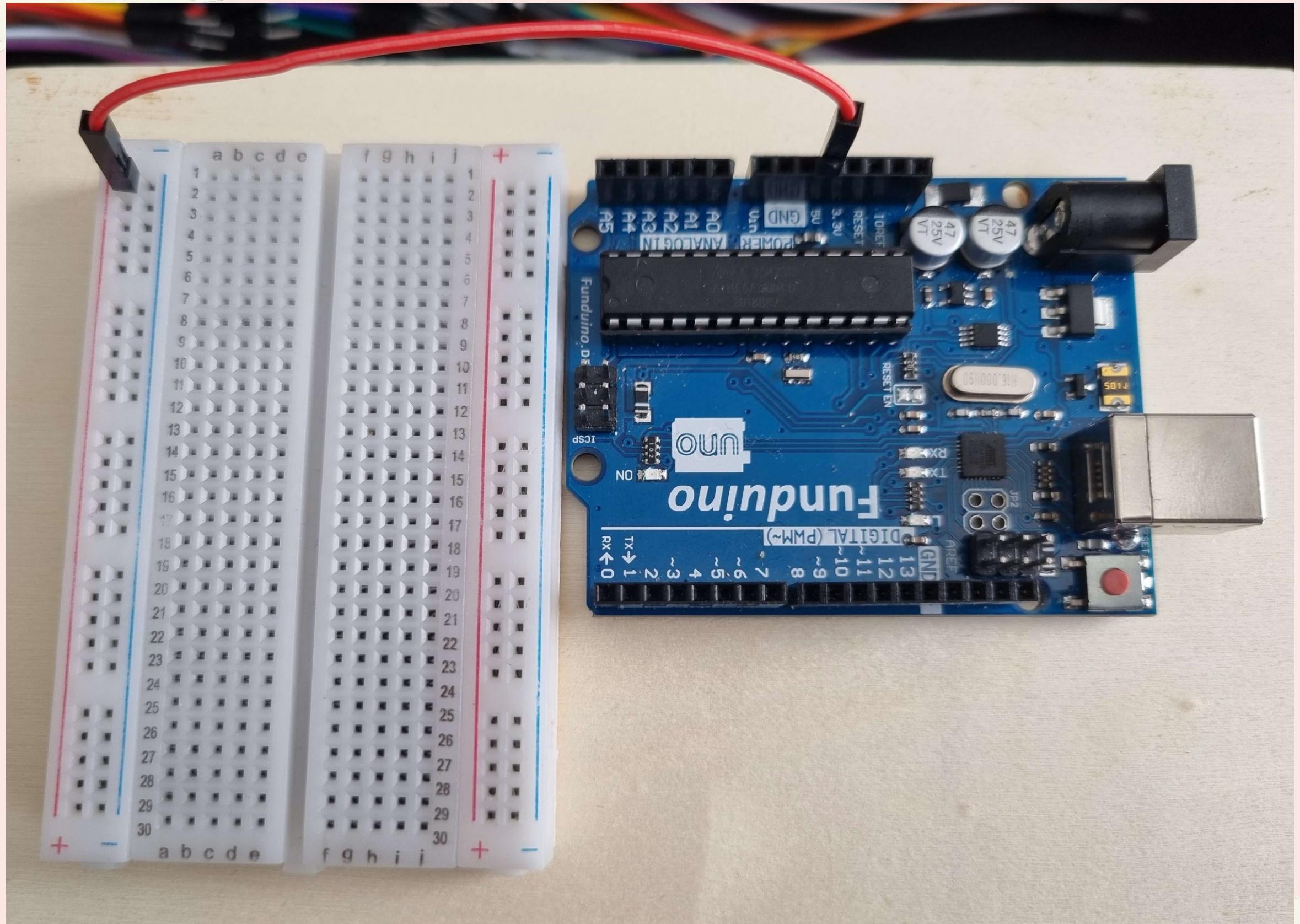


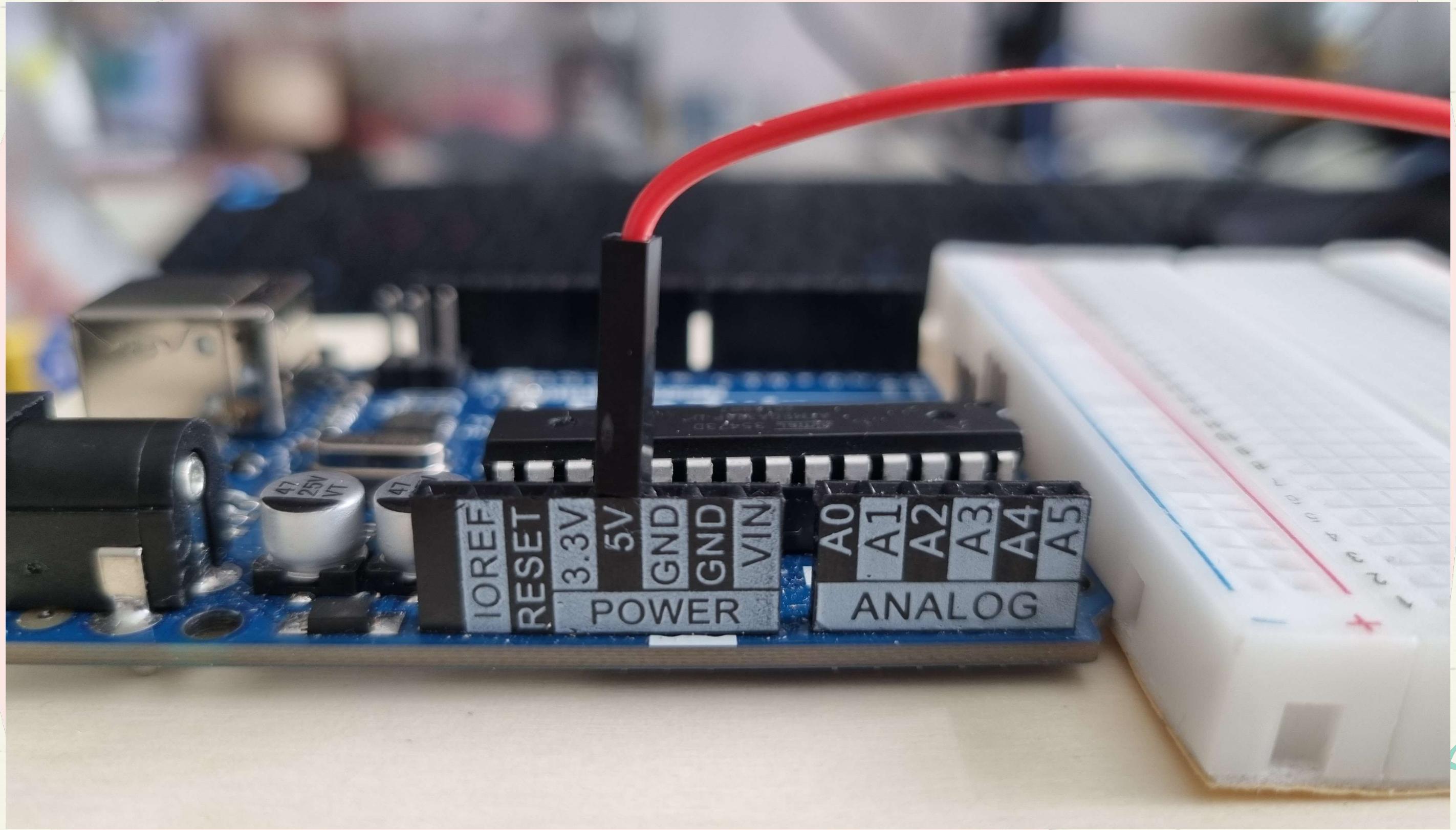
eee



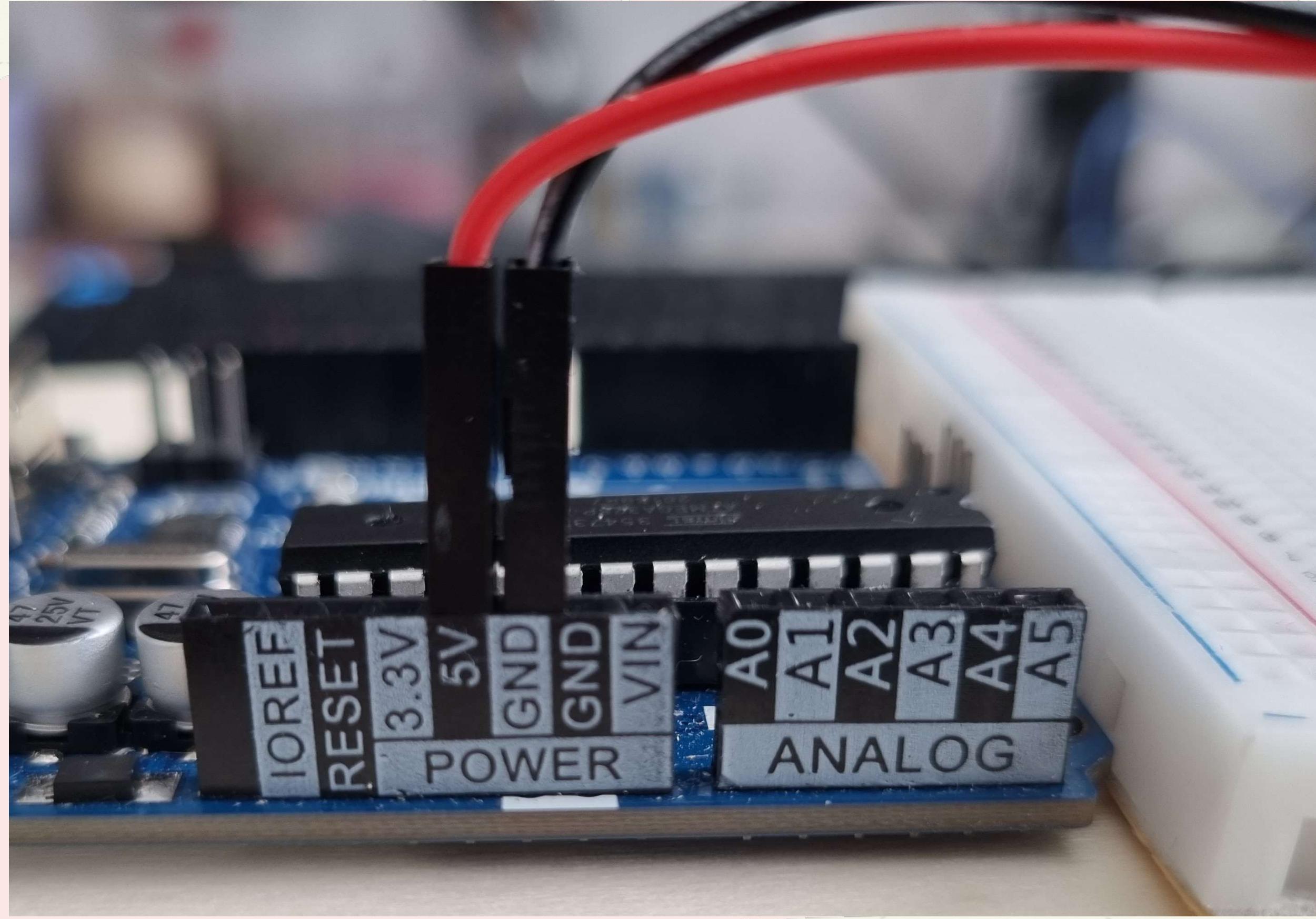


Well

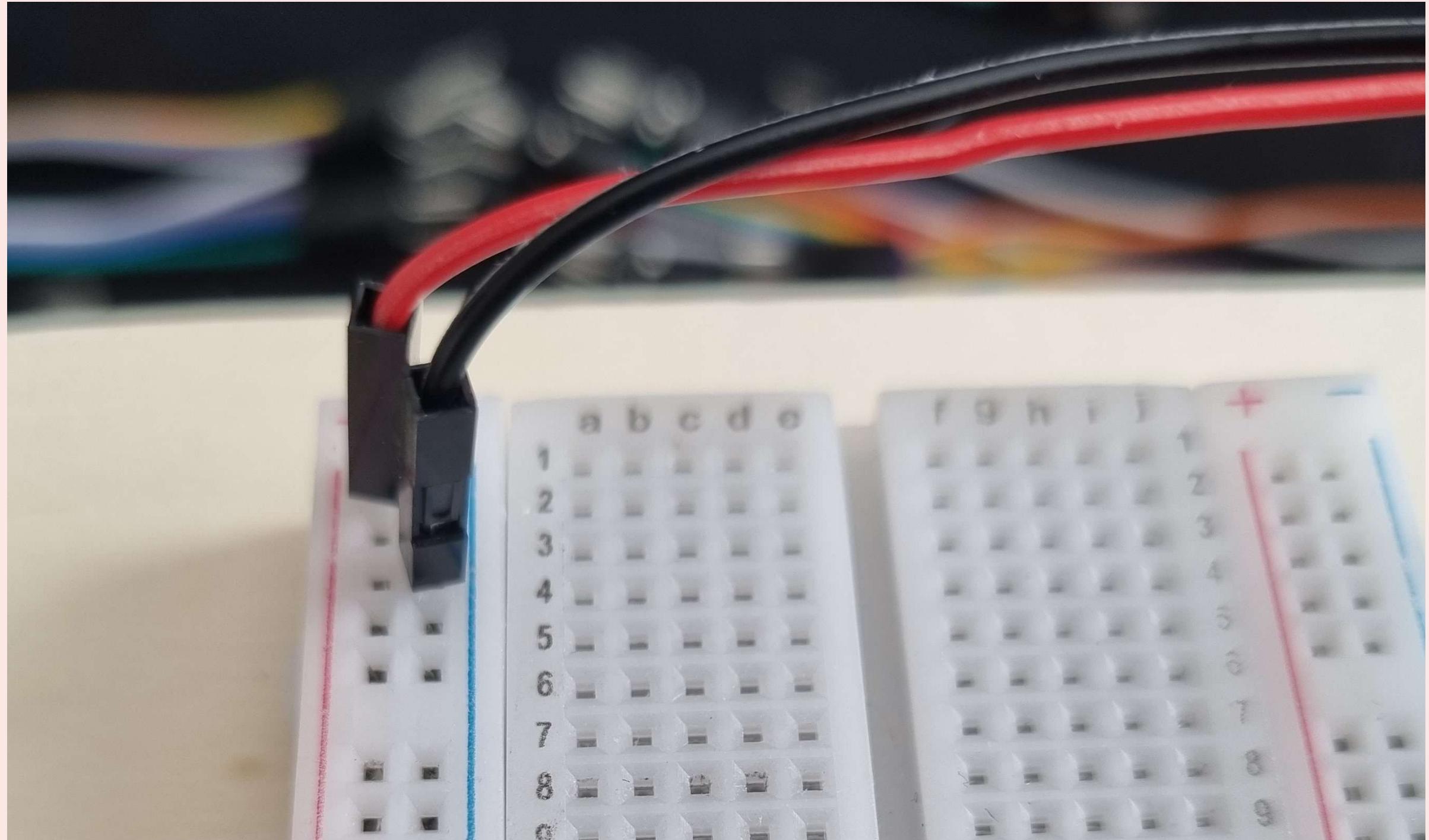




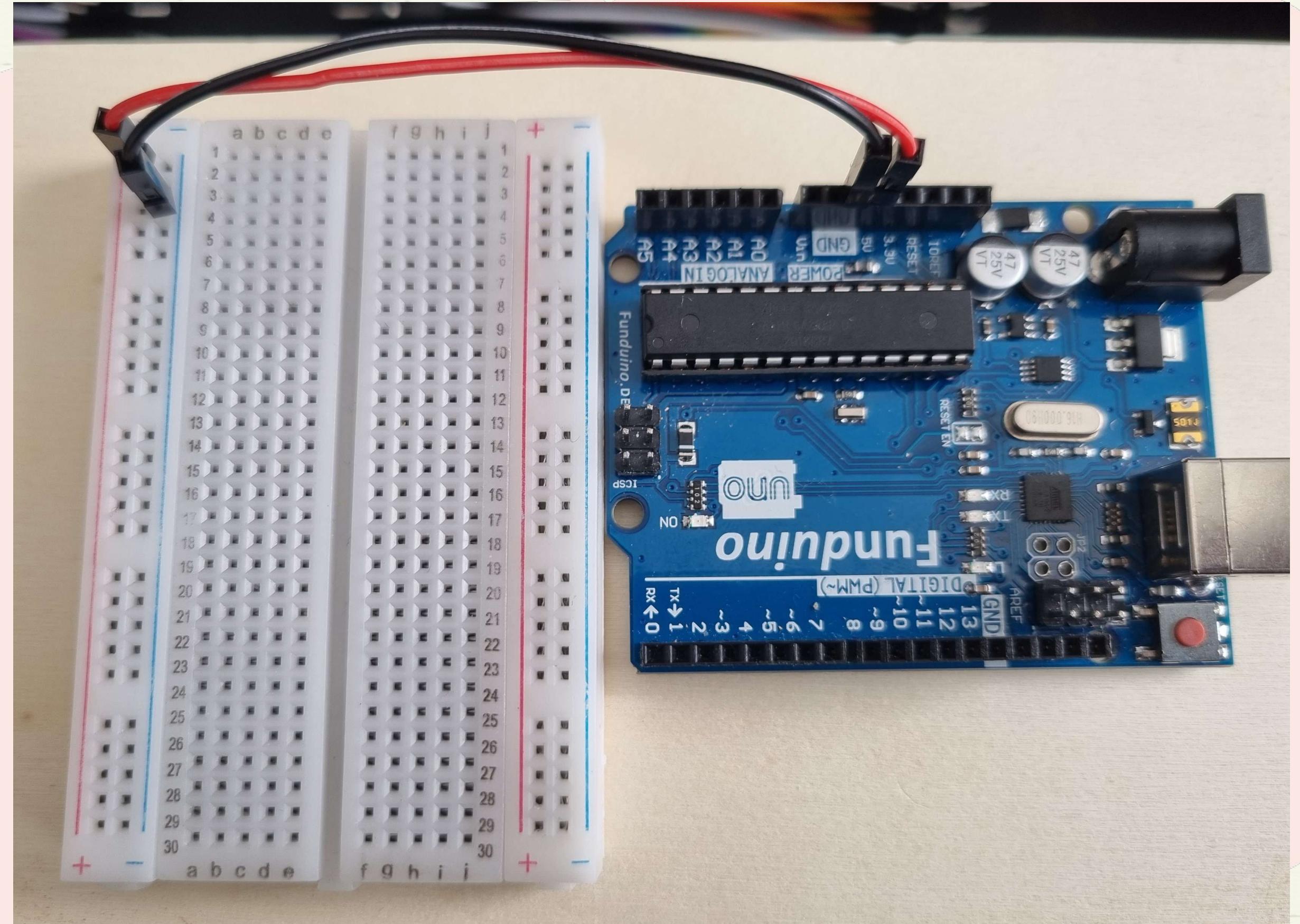
eee



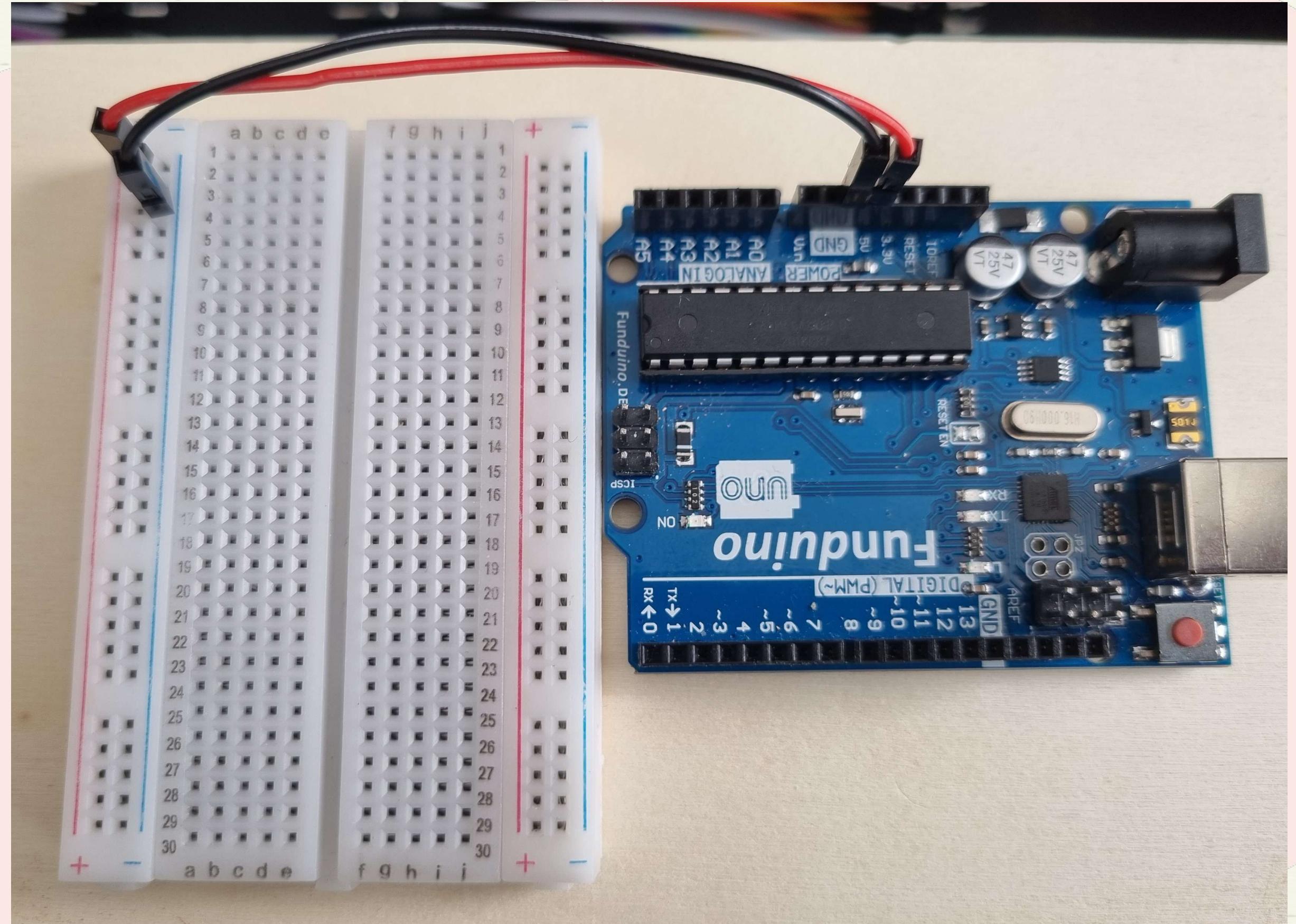
eee



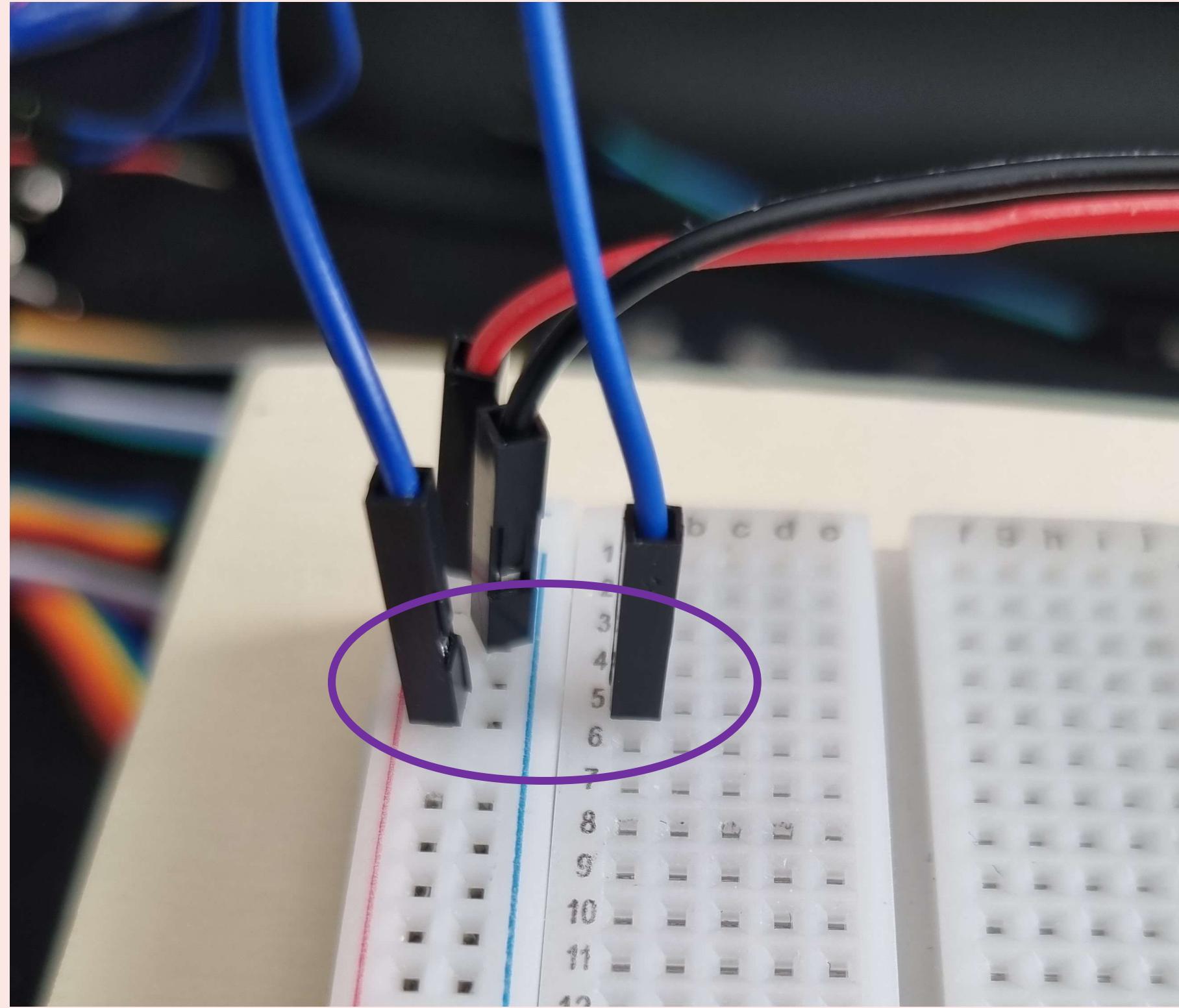
eee



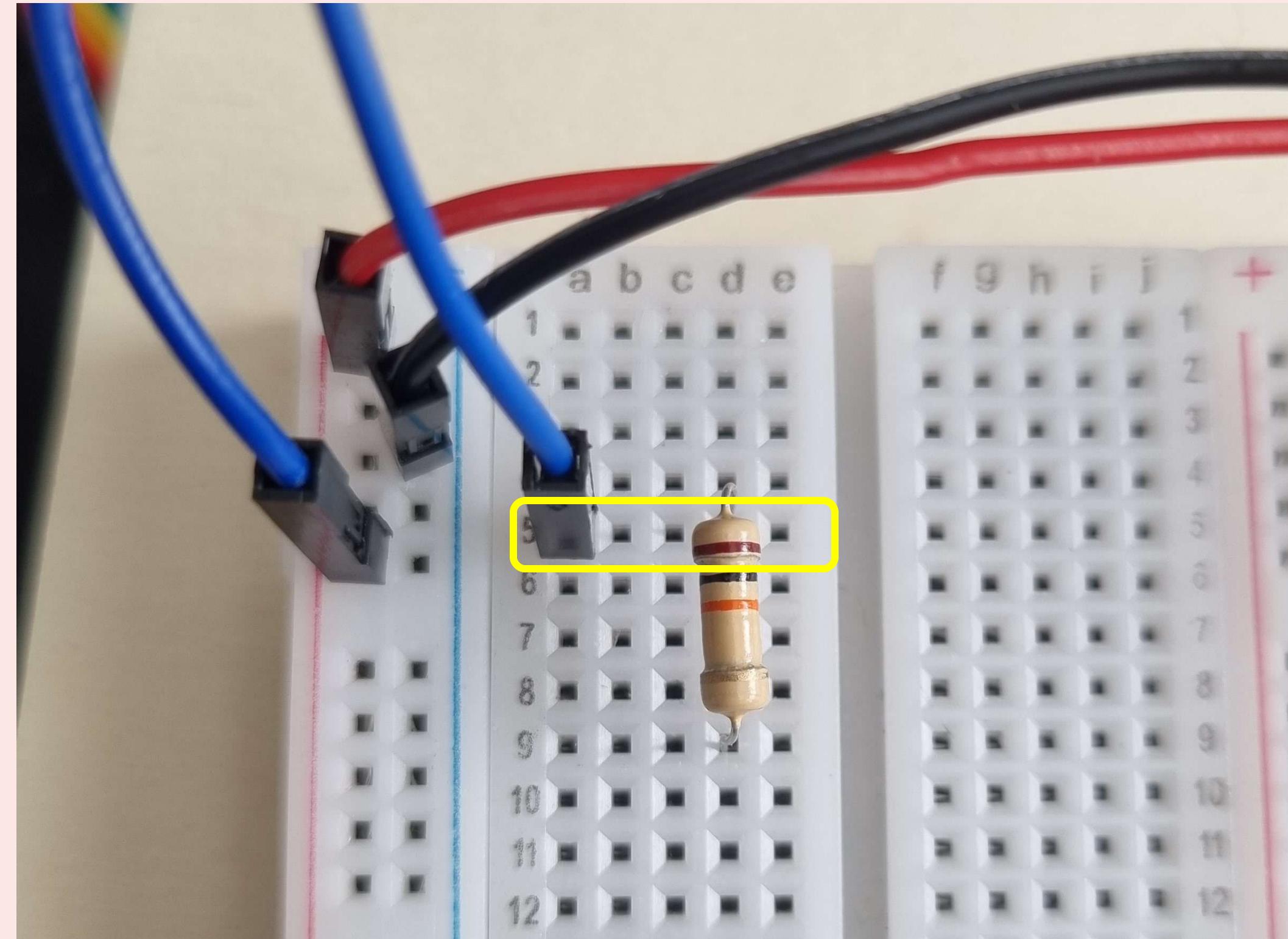
hell

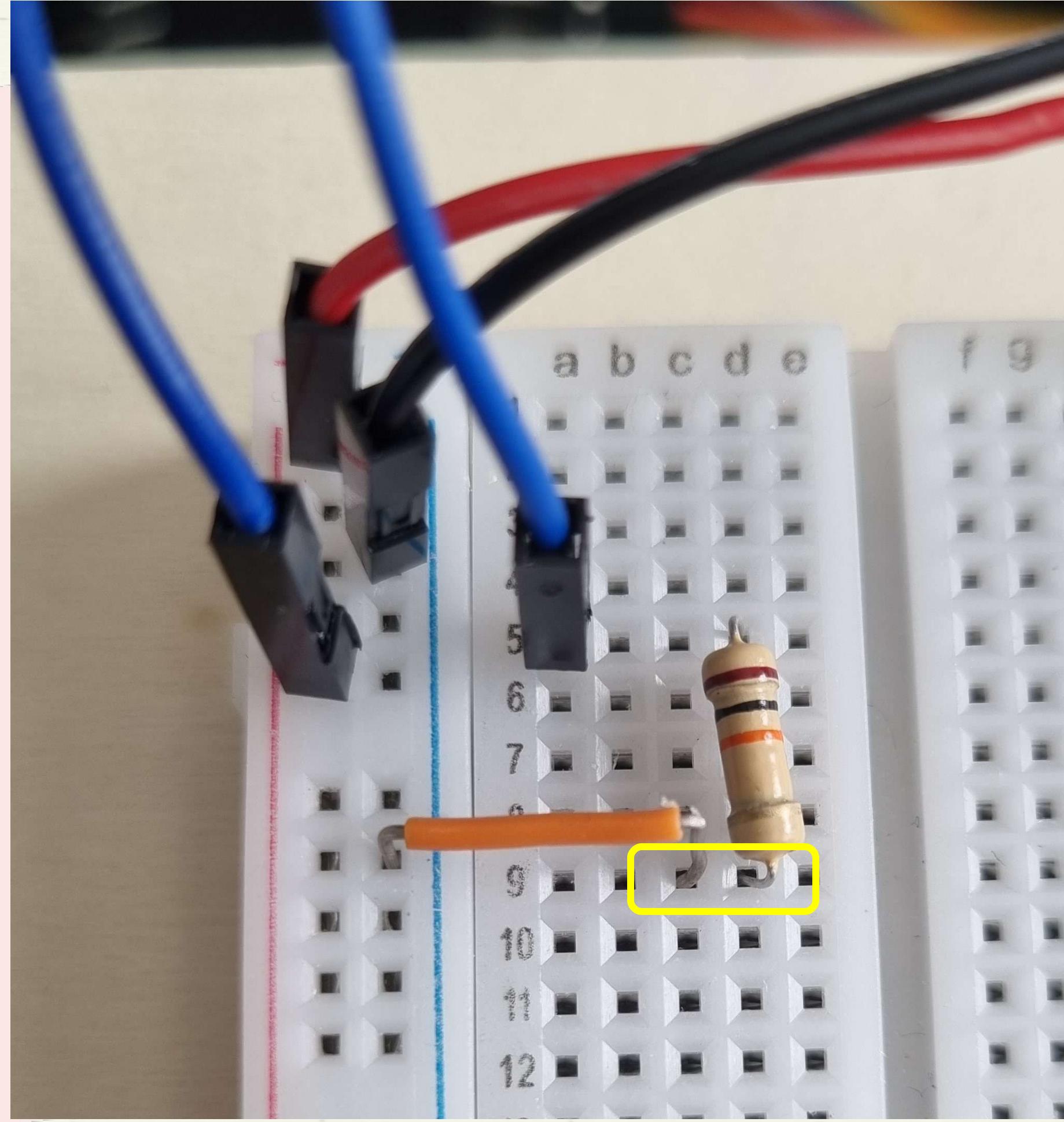


hell

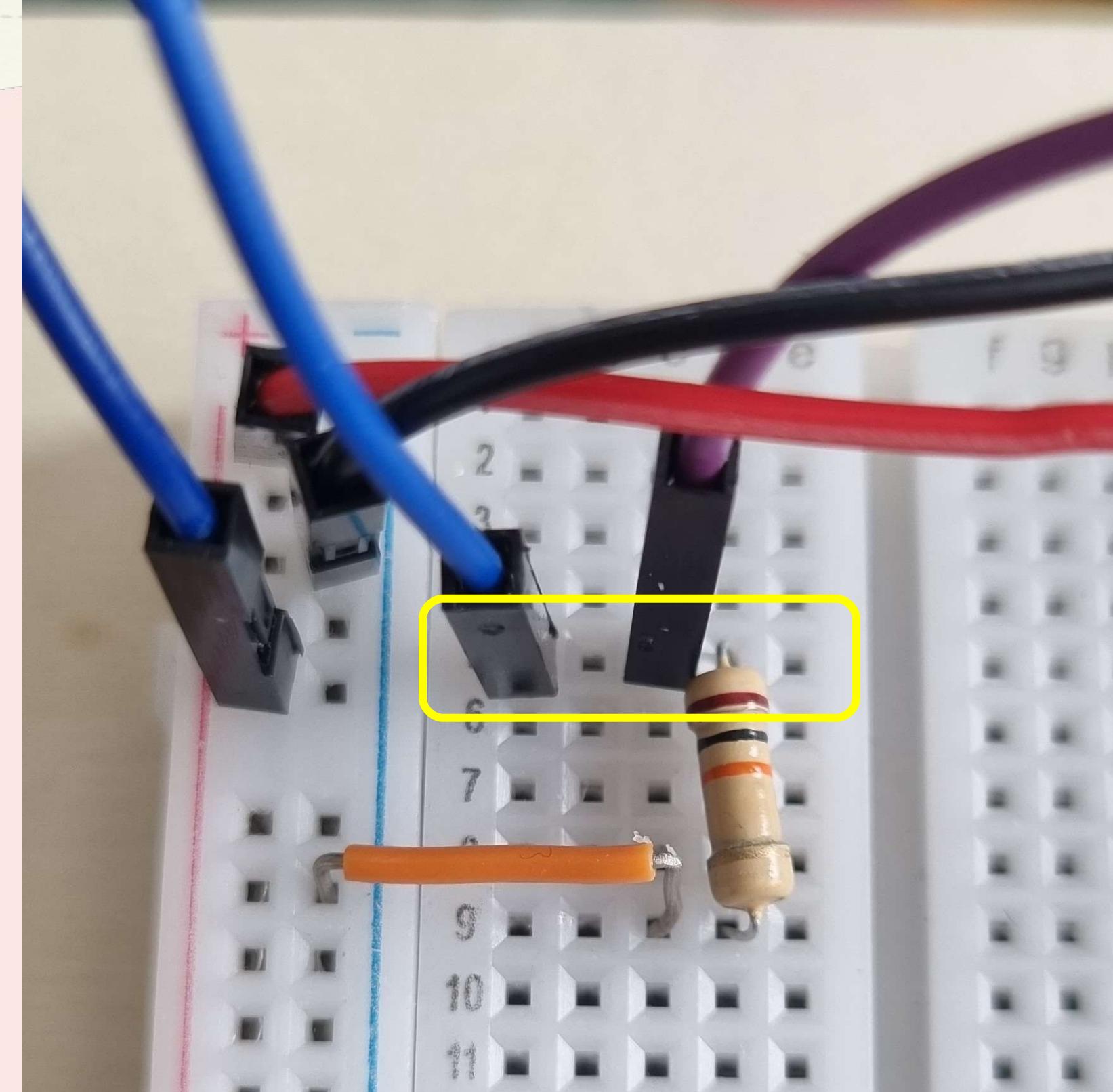


eee

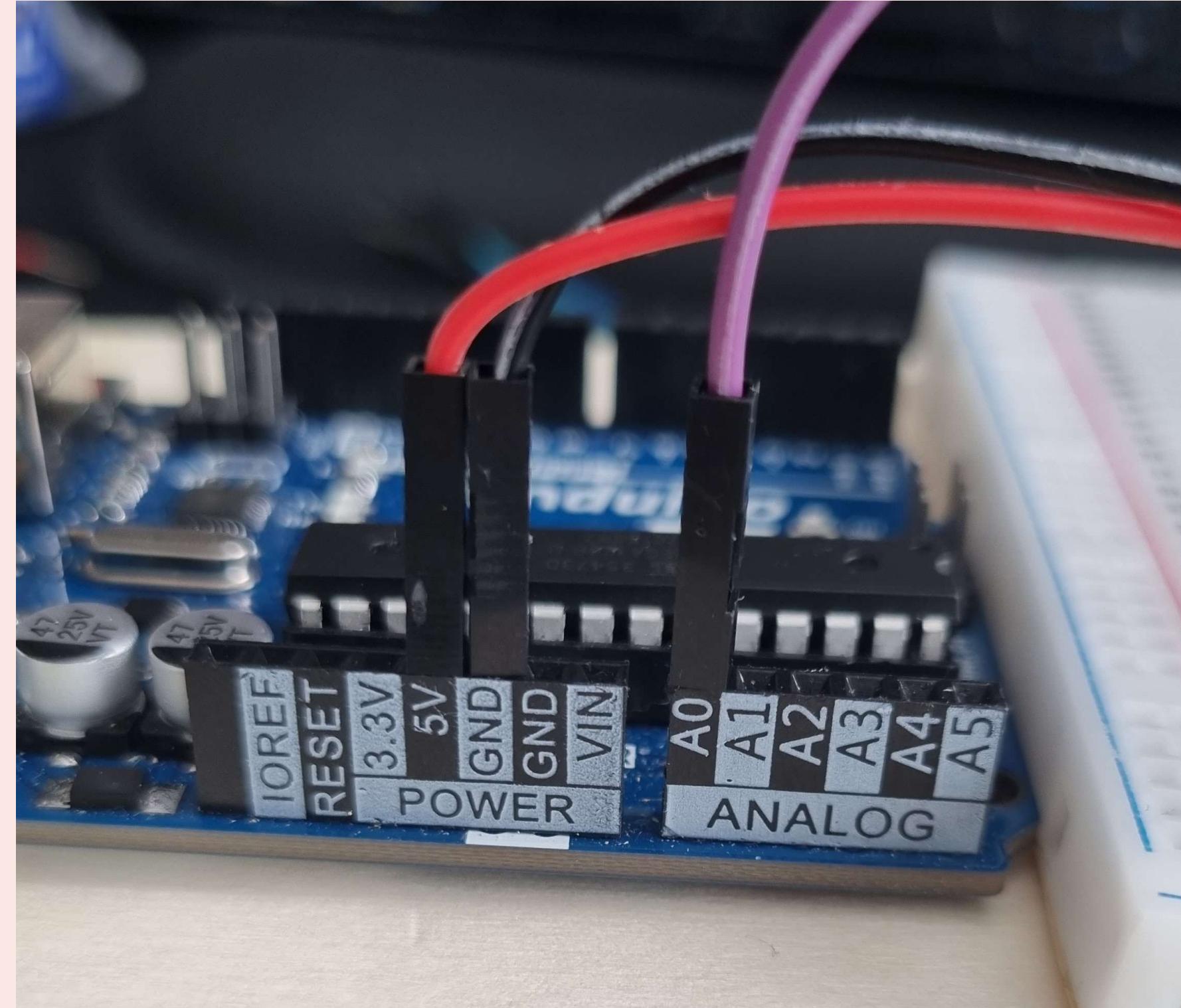




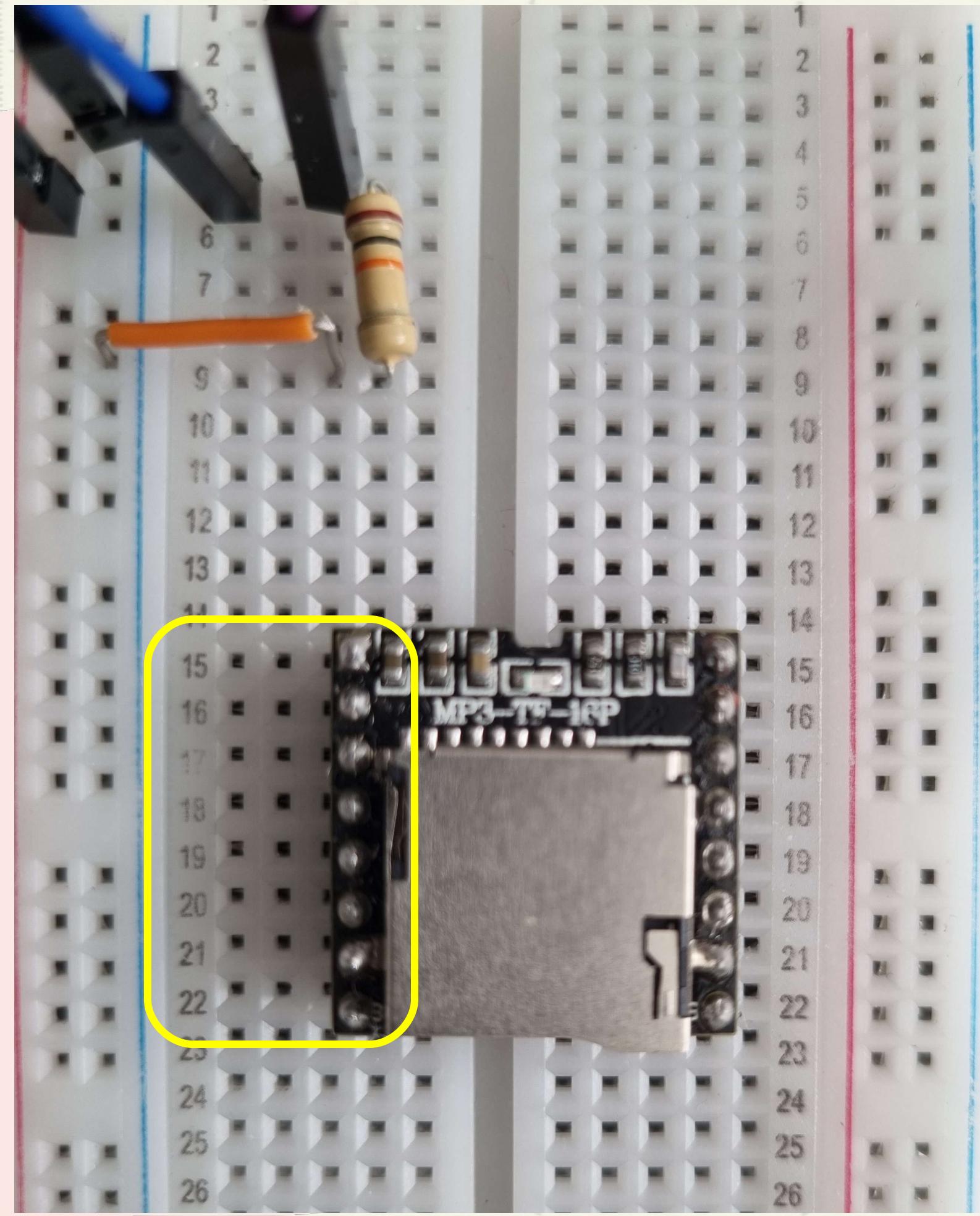
eee



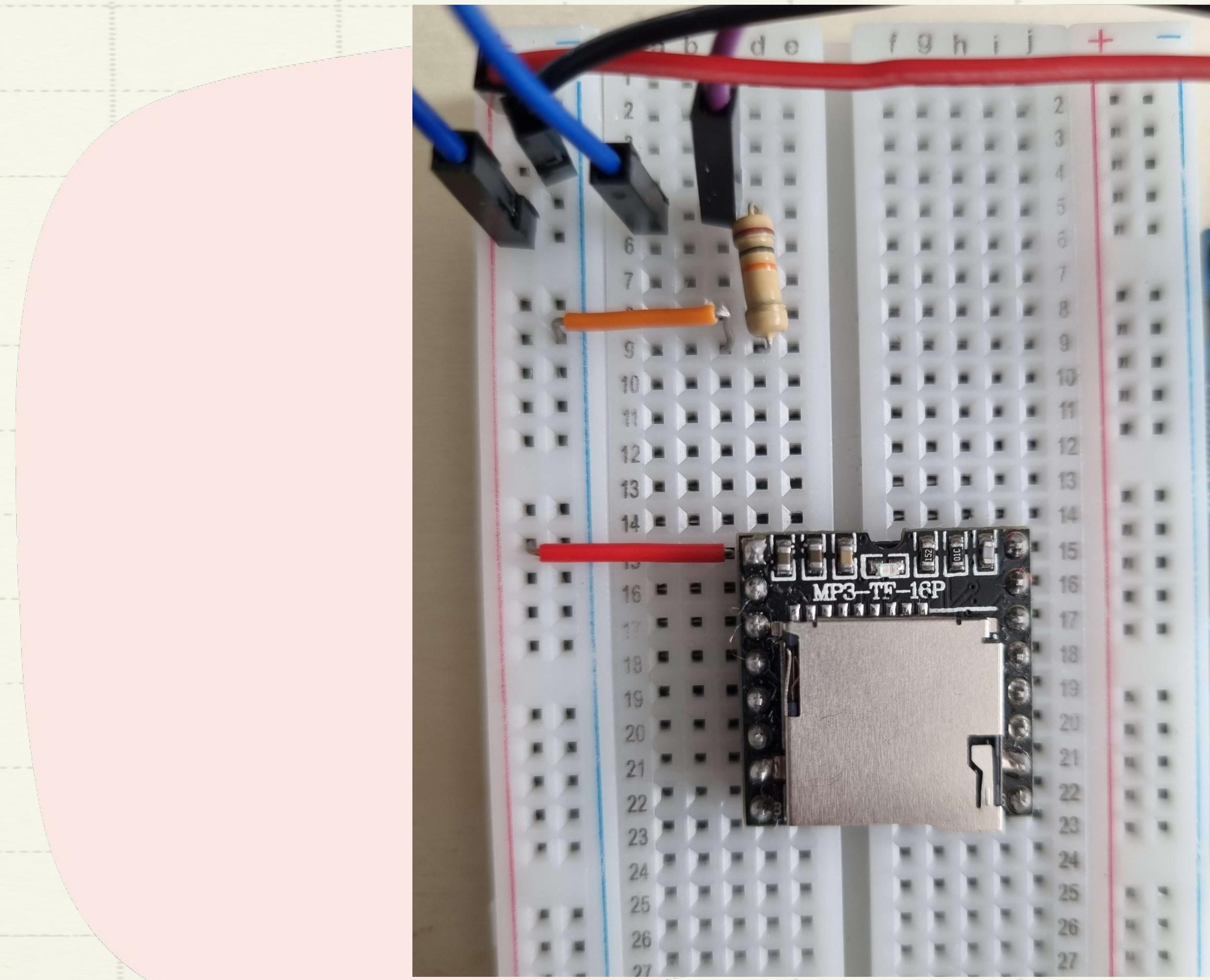
eee



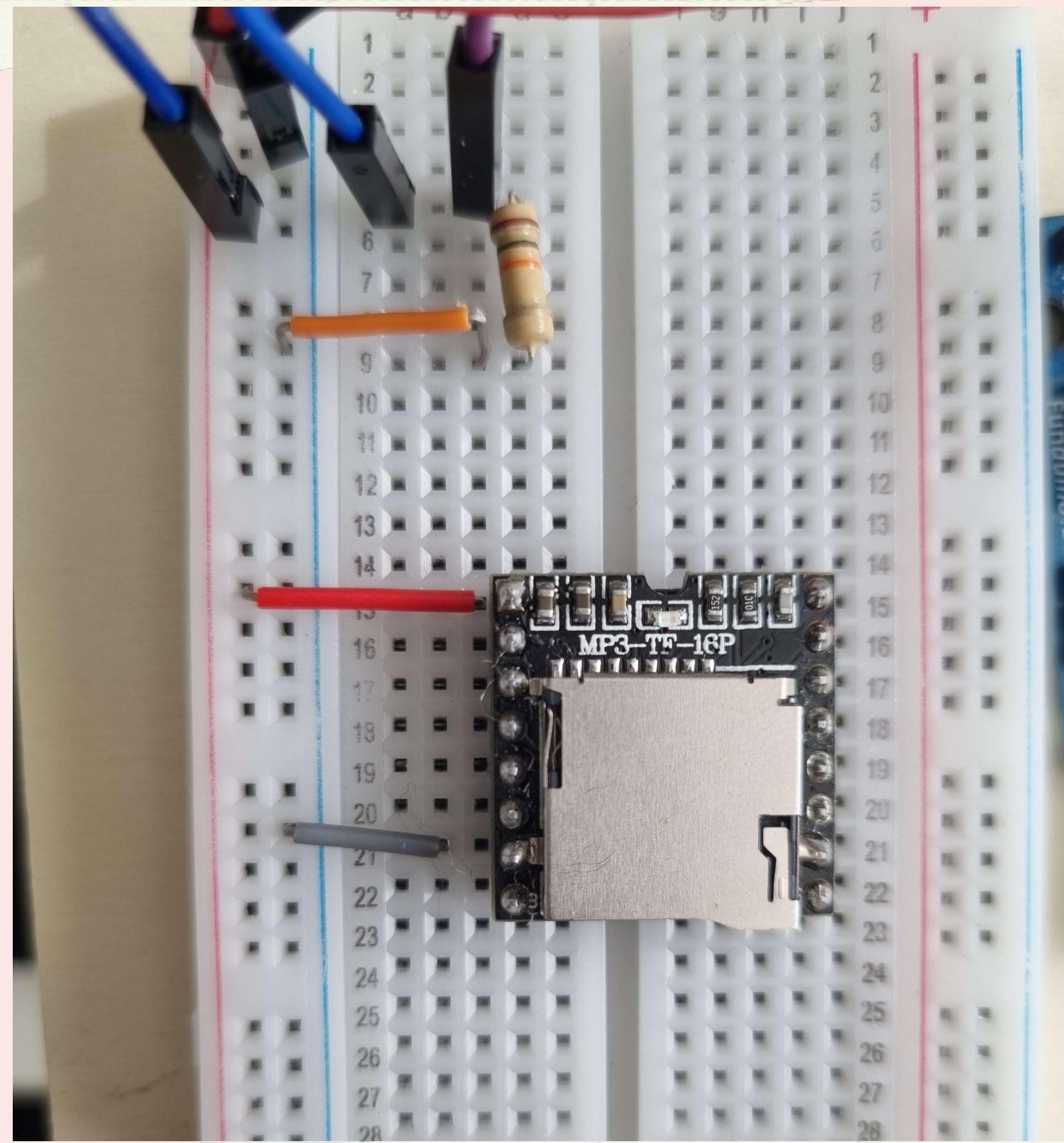
eee



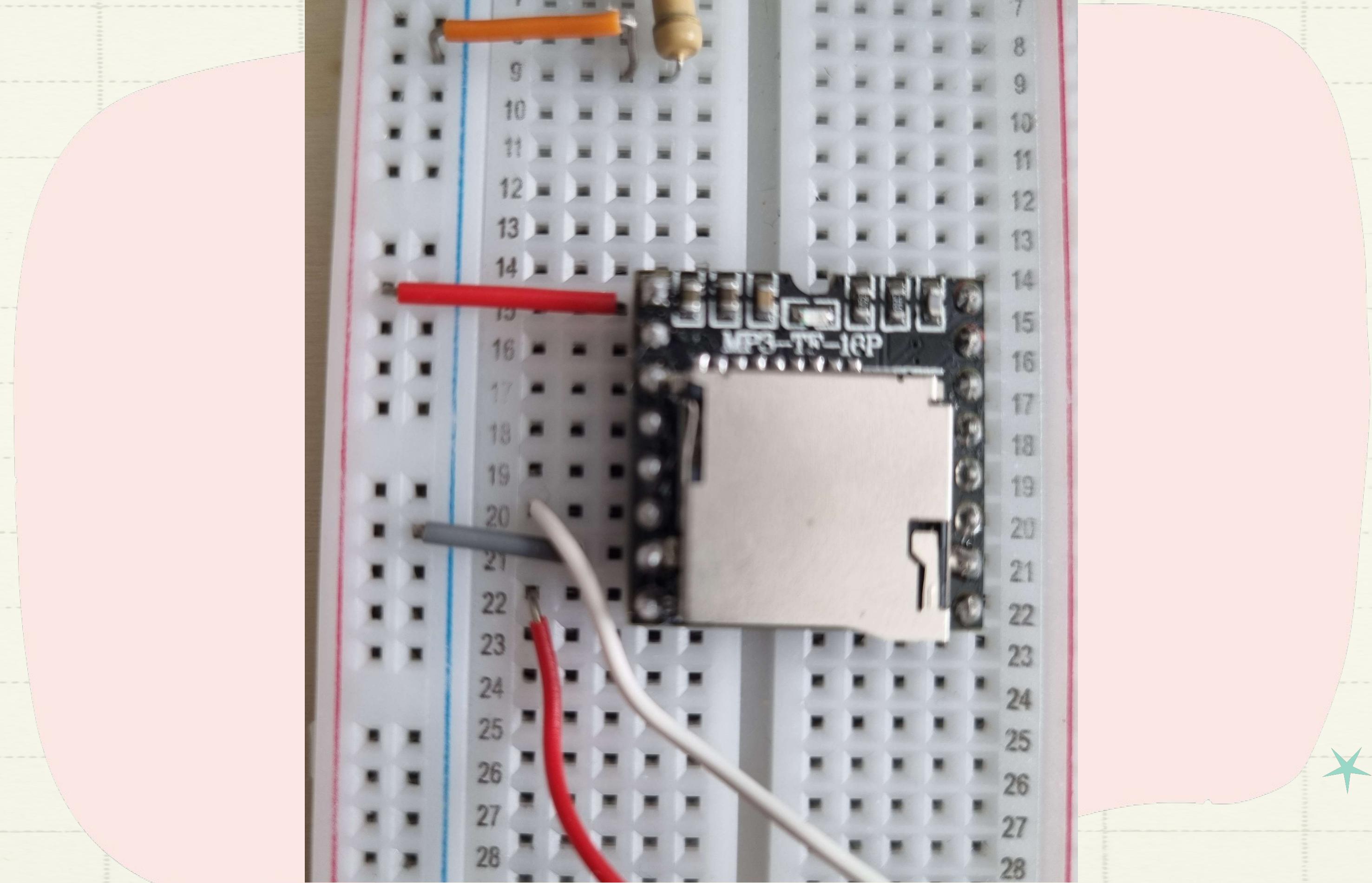
hell



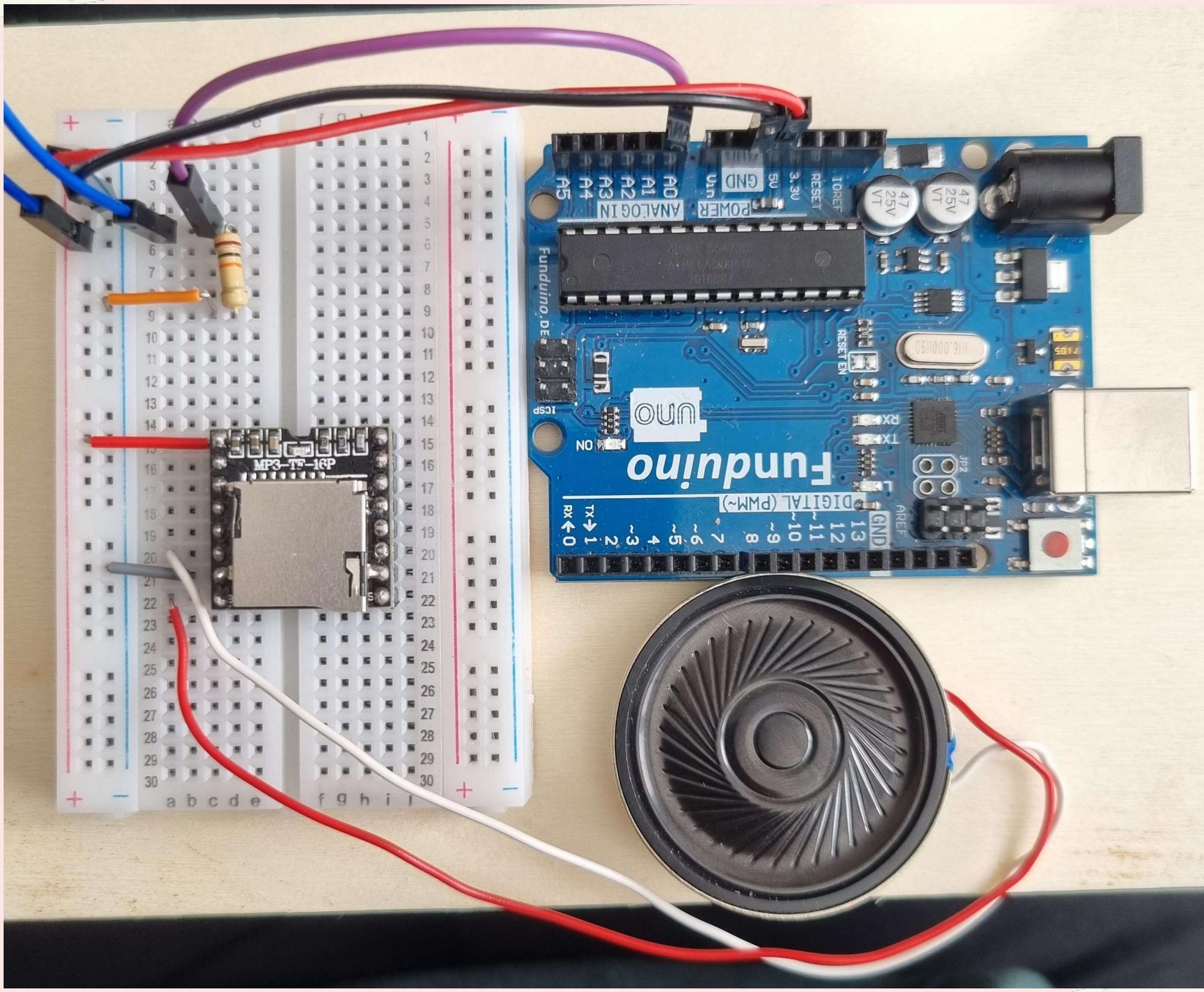
hell



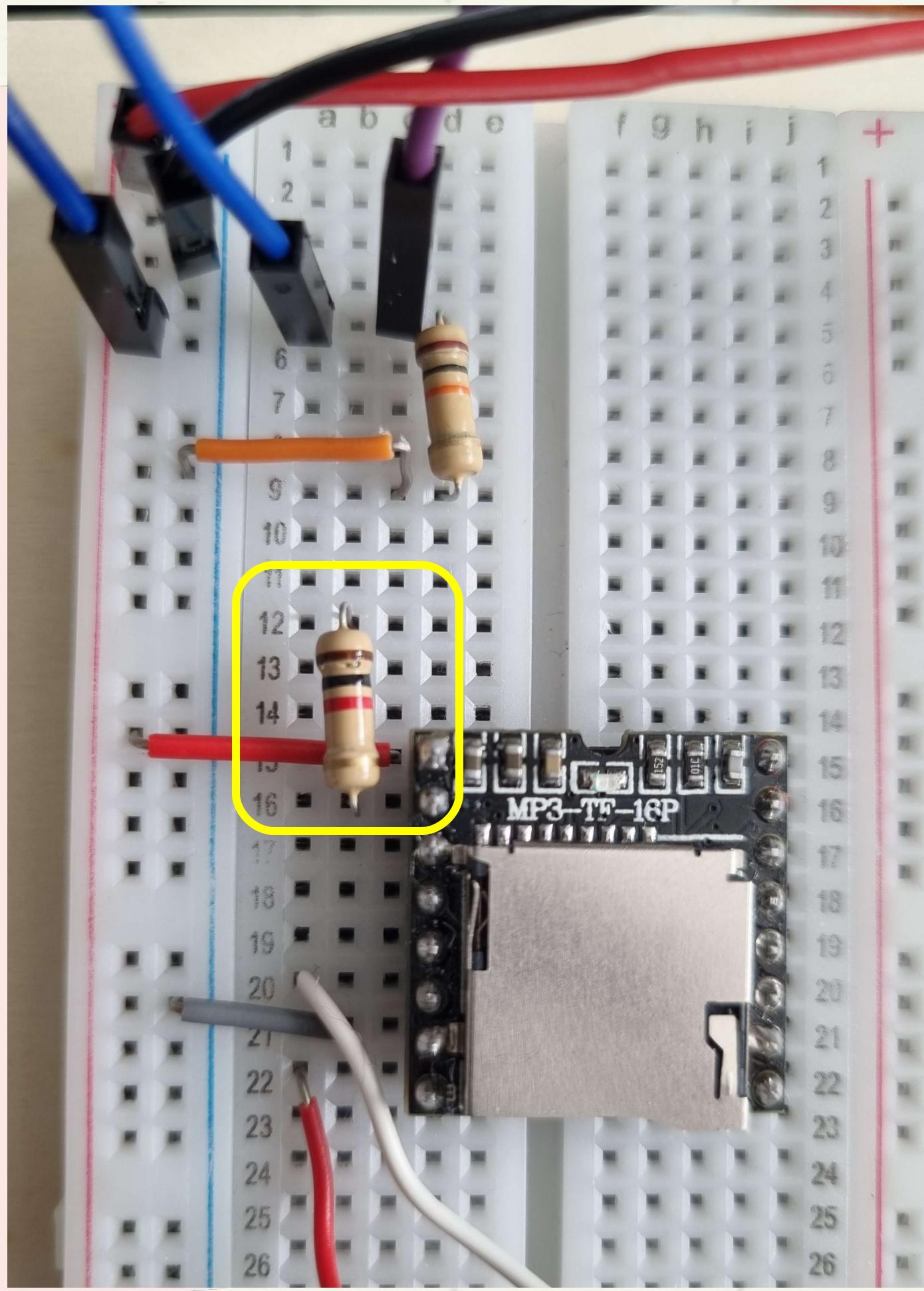
hell



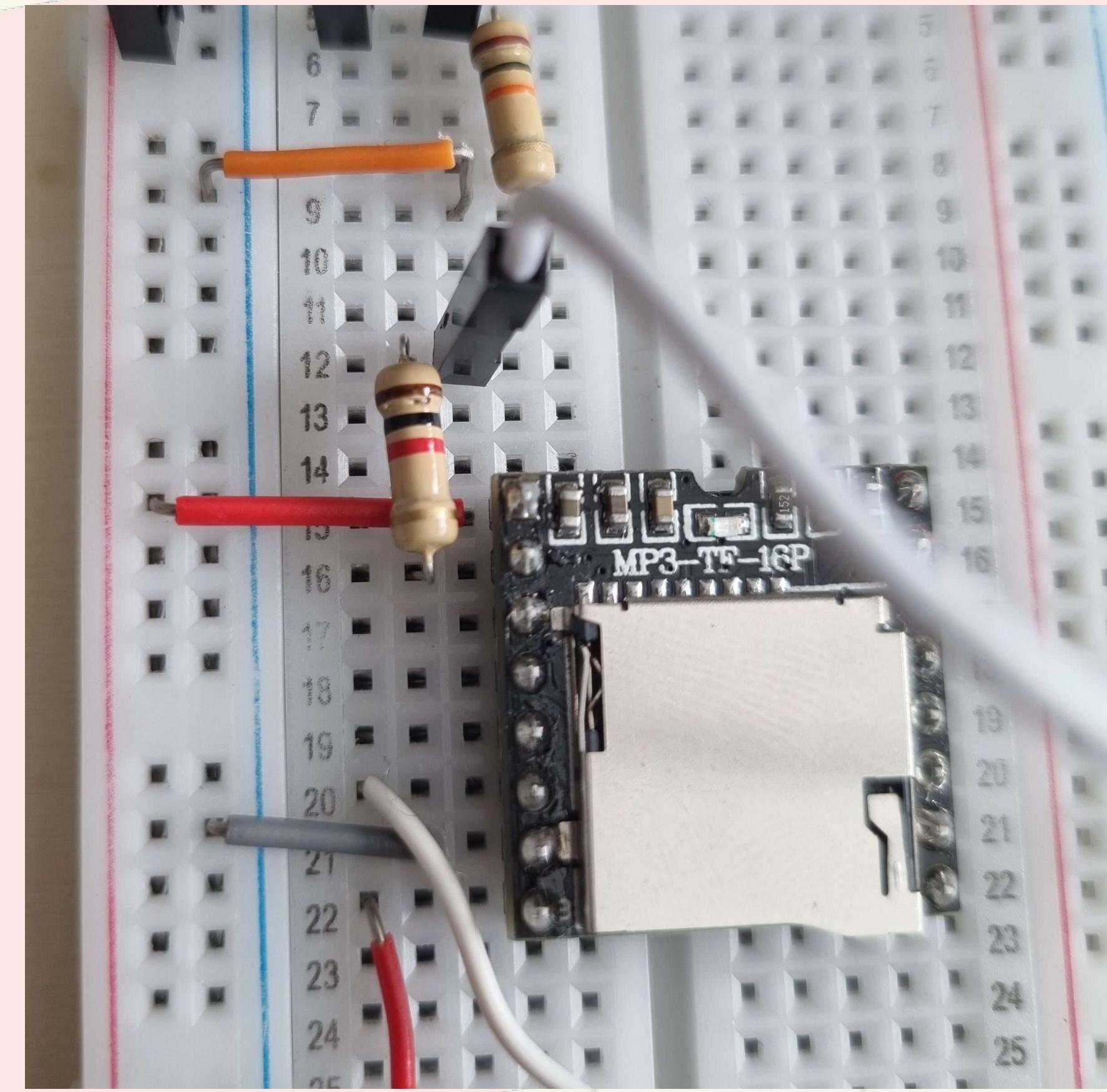
Well



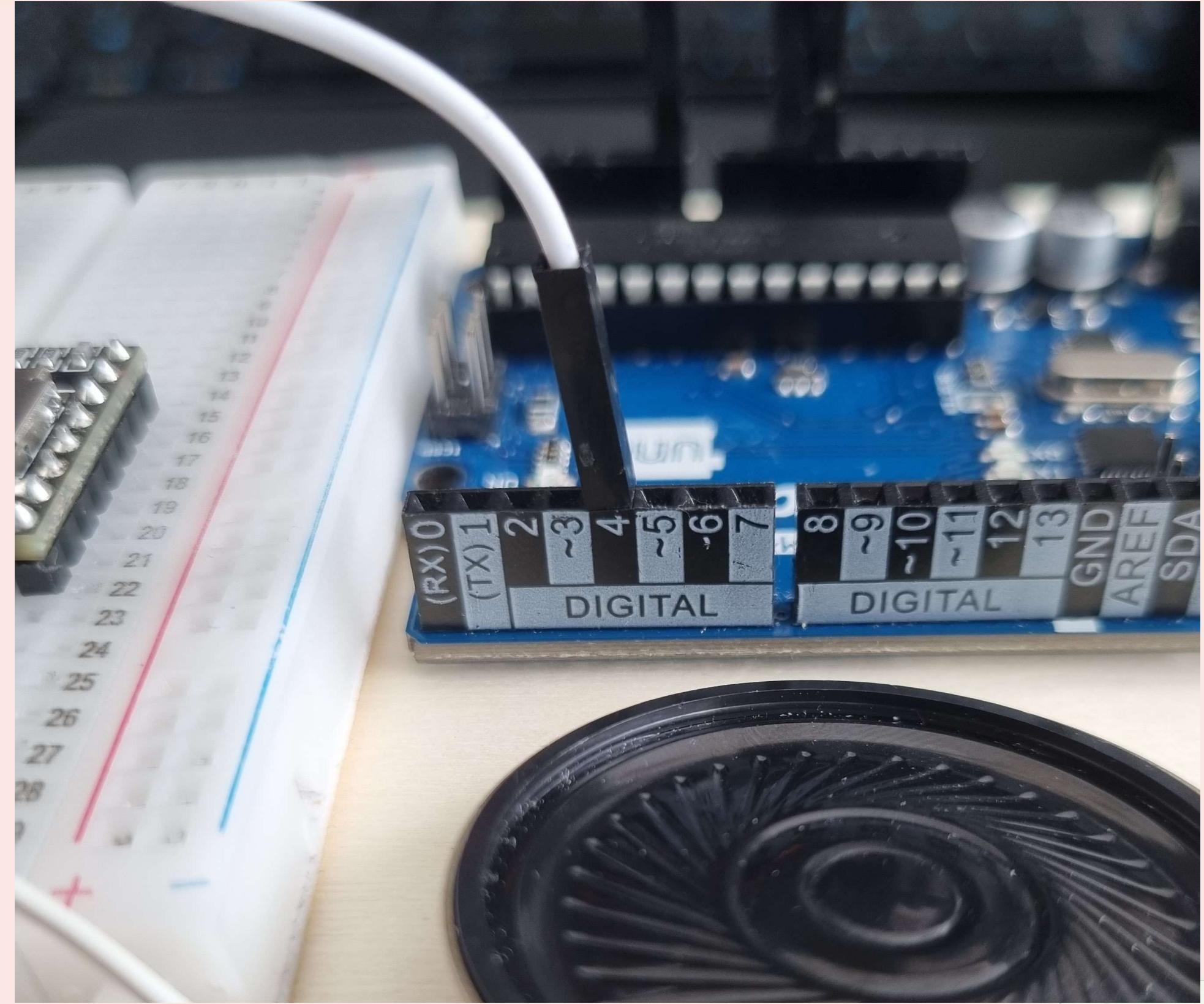
www



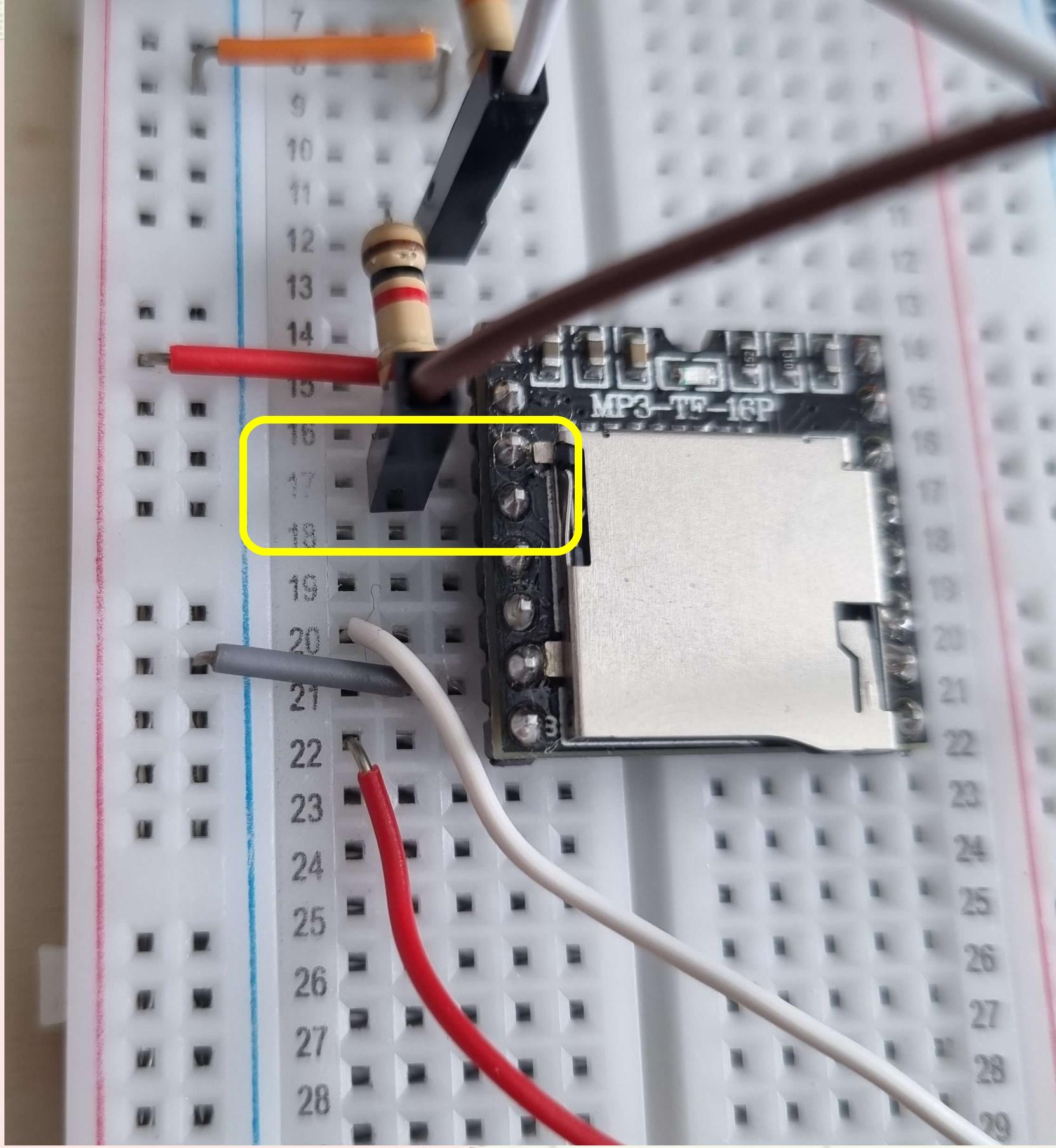
hell



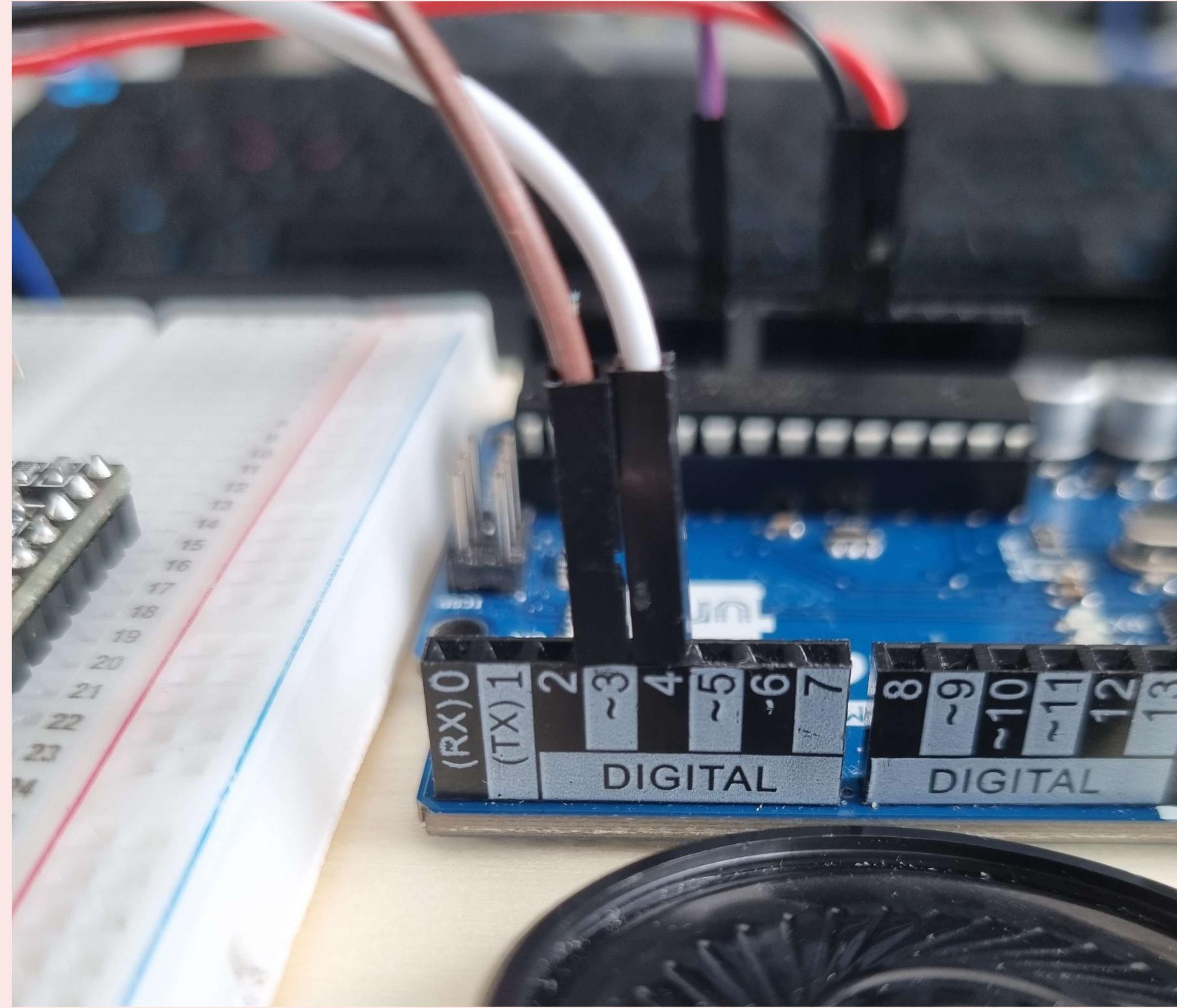
hell



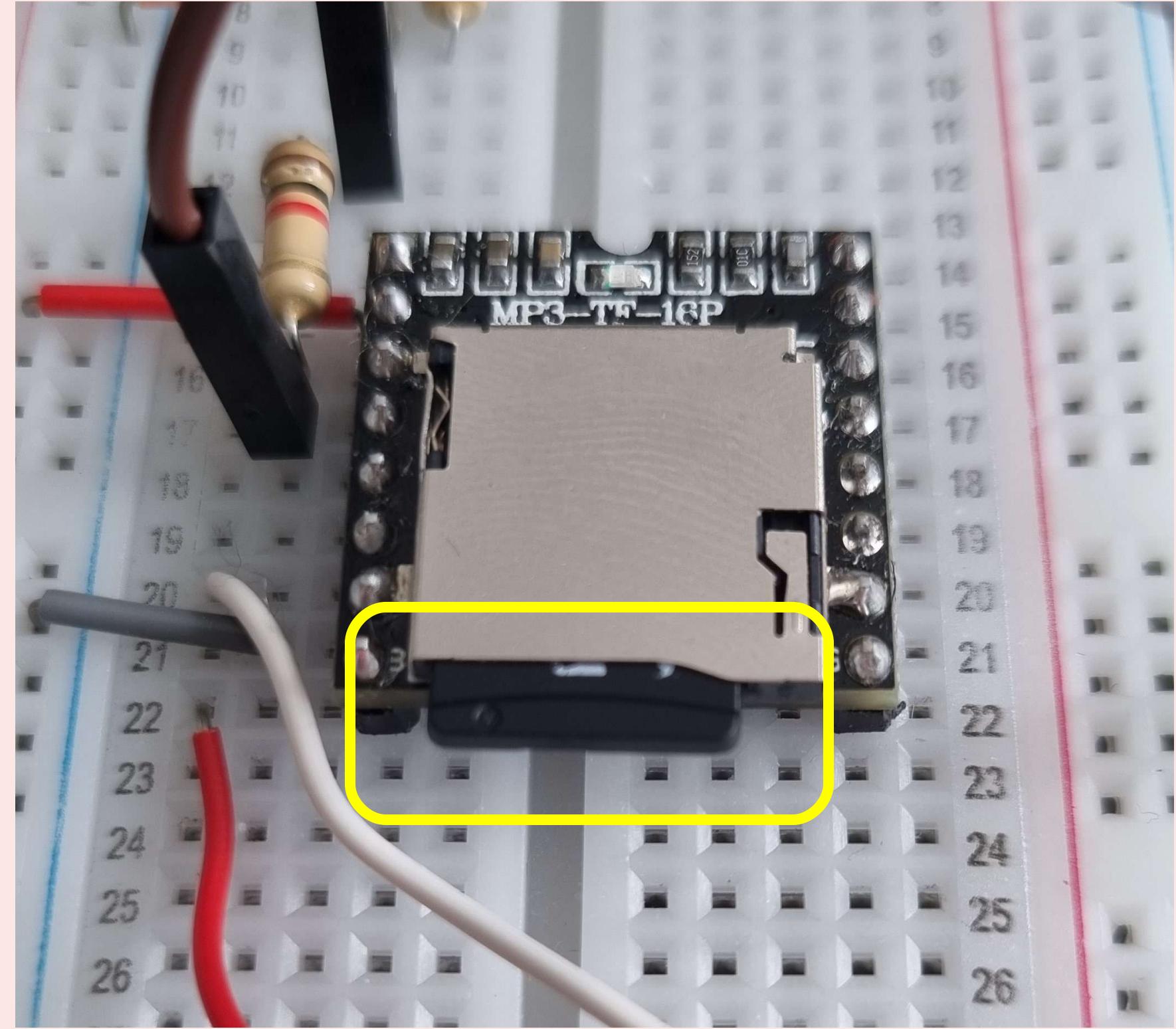
eee



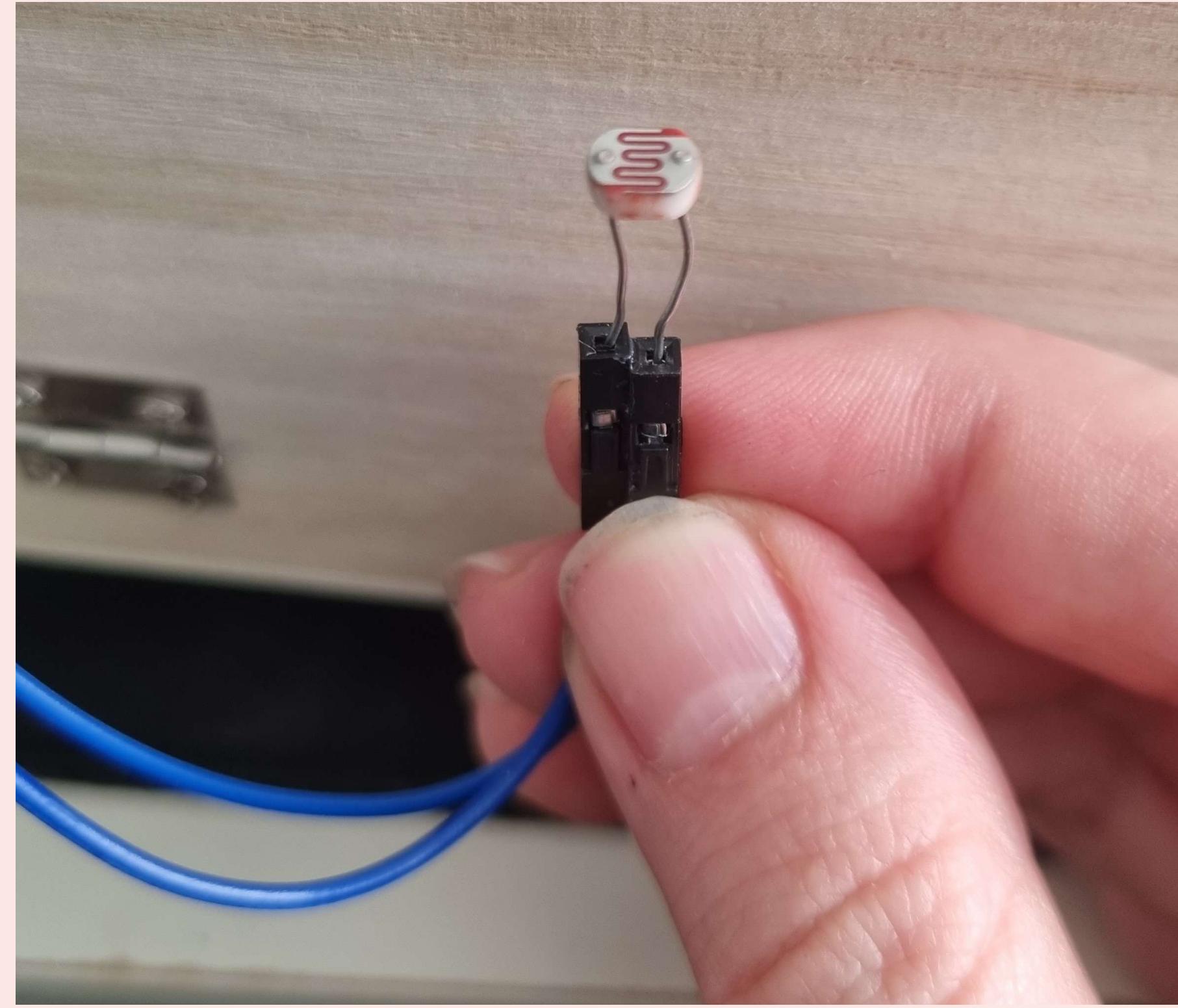
eee



eee



eee



eee



Die Logik hinter der
Musikbox!

eee

GirlsDayMusikBox.ino

```
1 #include "SoftwareSerial.h"  
2 #include "DFRobotDFPPlayerMini.h"
```

eee

```
4 int Receive = 3;  
5 int Transmit = 4;  
6 int ZustandVorher =0; //1 Hell 0 Dunkel  
7 int LichtSensor = A0;  
8 int Zustand_LichtSensor = 0;
```

hell



```
10 SoftwareSerial softwareSerial(Receive, Transmit);  
11 DFRobotDFPlayerMini mp3;
```

```
13 void setup() {  
14  
15   Serial.begin(9600);  
16   softwareSerial.begin(9600);  
17  
18  
19   if (mp3.begin(softwareSerial)) {  
20     Serial.println("OK");  
21     // Lautstärke festlegen (0 bis 30).  
22     mp3.volume(24);  
23   } else {  
24     Serial.println("Connecting to DFPlayer Mini failed!");  
25   }  
26 }
```

hell

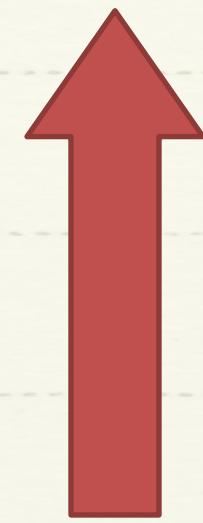
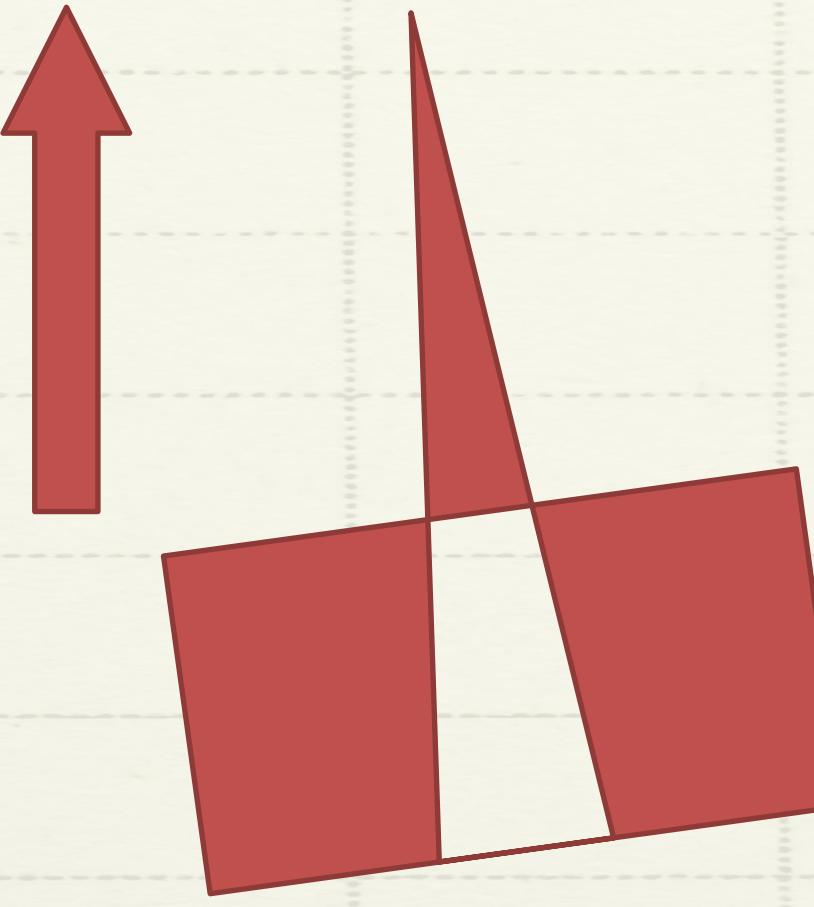
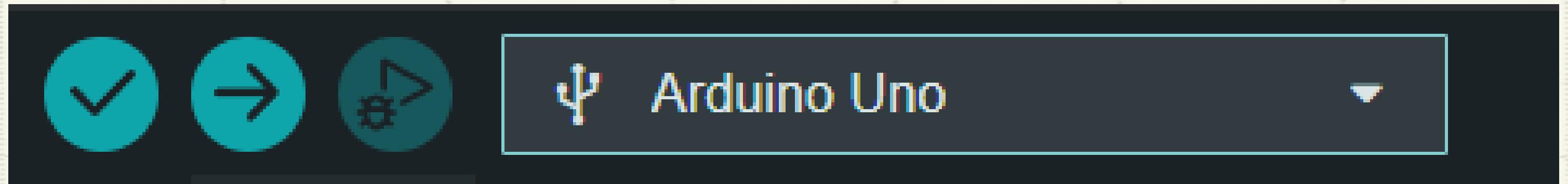
```
28 void EsIstHell(){
29     ZustandVorher =1;
30     mp3.play(1);
31 }
32 void EsIstDunkel(){
33     ZustandVorher =0;
34     mp3.stop();
35 }
```

hell

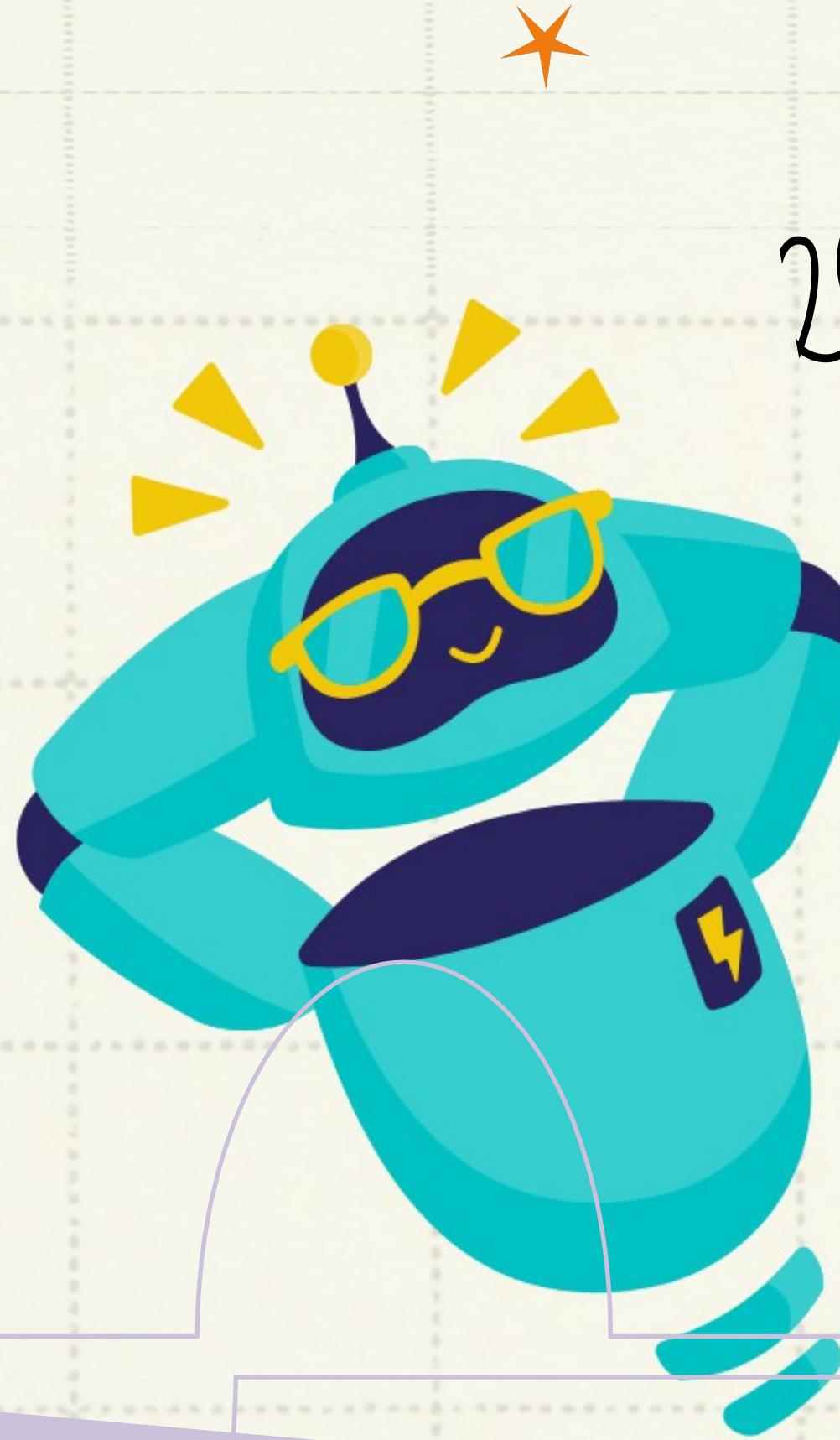


```
37 void loop() {  
38     Zustand_LichtSensor = analogRead(LichtSensor);  
39  
40     Serial.println(Zustand_LichtSensor);  
41     delay(500);  
42  
43     if(Zustand_LichtSensor > 100 && ZustandVorher == 0){  
44         EsIstHell();  
45     }  
46     if(Zustand_LichtSensor < 99 && ZustandVorher ==1){  
47         EsIstDunkel();  
48     }  
49 }
```

hell



eee



well

Ende

**Viel Spaß mit deiner eigenen
Musikbox!!**



well