

T1 Sensors

Projektteam:E21-A1

Teammitglieder :

Henrik , Sahin , Dennis



Inhaltsverzeichnis

1. Gründe der Projektwahl

2. Projektbeschreibung

3. Sensoren

4. Projektablauf

5. Erlernte Kenntnisse

6. Funktionsnachweis

7. Projektdokumente

- Arduino, Fritzing, Cura, Materialliste

8. Aufgetretene Probleme

9. Quellen

10. Fragerunde

1. Gründe für die Projektwahl

- Interesse an Sensorik
- Der tägliche Kontakt mit Sensoren im Leben (Auto, Arbeit, Privaten)
- Verschiedene Inhalte und Module kennenzulernen
- Um Kenntnisse in den Programmen zu erlangen (Arduino, CATIA, ...)

2. Projektbeschreibung

- Bewegungserkennung am T1
- Infrarotsensor für die Bewegungserkennung
- Buzzer für die akustische Alarmausgabe
- 4 Gelbe LEDs für die optische Alarmausgabe
- Umsetzen mit einen Mikrocontroller (mit Arduino Programmiert)

3. Bewegungsmelder

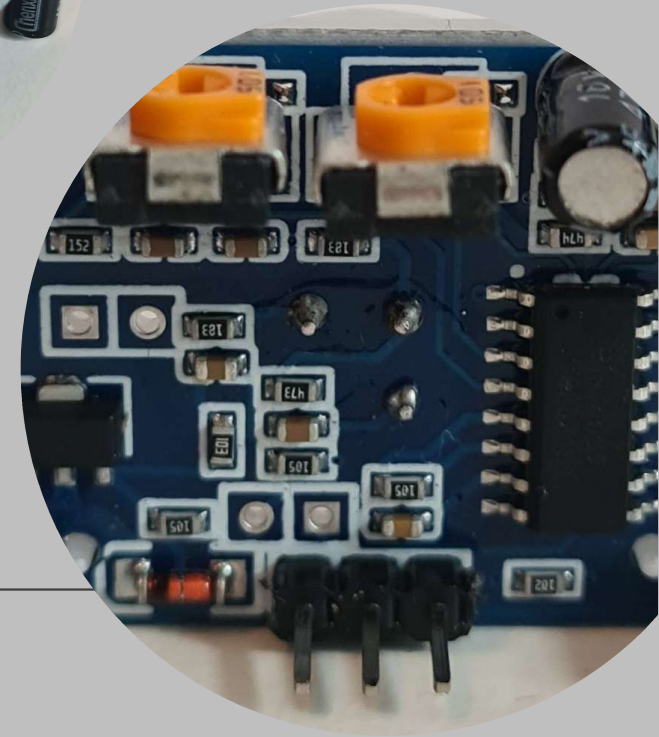
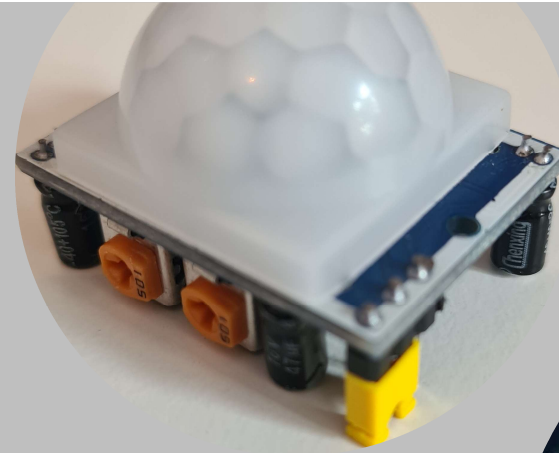
- **Beschreibung:**

- Geräte, die Bewegungen von Menschen, Tieren oder auch von Gegenständen erkennen können.
- nutzen verschiedenste Technologien die Bewegungen als Schaltsignal weitergeben.
- Anwendungsbereich im Innen- oder Außenbereich

- **Kenndaten:** Ausgang: 3,3 V (digital)
Reichweite: 3 bis 7 Meter (einstellbar)
Erfassungsbereich: 140°
Betriebstemperatur: -15 bis 70 °C
Empfindlichkeit, Haltezeit und Impulsrate ist einstellbar

Anwendungsbeispiel:

- automatische Lichtansteuerung (Energieeinsparung, Sicherheitsfunktion)
- für Sicherheitsaufgaben (z.B. Alarmanlage)



3. Ultraschallsensor



Beschreibung:

-kann die Entfernung zu einem Objekt bestimmen

Ultraschallwellen werden gesendet (Trig), vom Objekt reflektiert und vom Sensor empfangen (Echo)

-vergangene Zeit wird gemessen und lässt auf Entfernung zurück schließen

Kenndaten:Anschlüsse:

VCC (Spannungsversorgung 5 V)

GND (Spannungsversorgung

Ground)

Trig (Sender)

Echo (Empfänger)

Wichtige Informationen:

Nicht verpolen!

Messbereich liegt zwischen ca. 2 cm und 3 m

4. Projektablauf

1. Aufbau des T1-Buses
2. Kennenlernen der Sensoren
3. Erstellung und Zusammensetzung des Schaltplanes
4. Programmieren des ESP32 in Arduino IDE
5. Funktionsprüfung
6. Herstellung des Sensorhalters (3D-Druck)
7. Montage der Komponenten in den T1
8. Erstellung Projektdokumentation

5. Erlernte Kenntnisse

1. Sensoren kennengelernt
2. Schaltpläne erstellen in Fritzing
3. Arduino IDE Programmierungen
4. Konstruktion mit Catia V5
5. 3D-Druck
6. Schaltungen auf dem Breadboard aufbauen
7. T1 verkabeln und Anbauteile montieren
8. Projektdokumentation erstellen
9. Lösungswege ermitteln und auf Probleme reagieren
10. Slicing Programme erstellen für den Ausdruck
11. Arbeitsaufteilungen Planen
12. Halten einer Präsentation

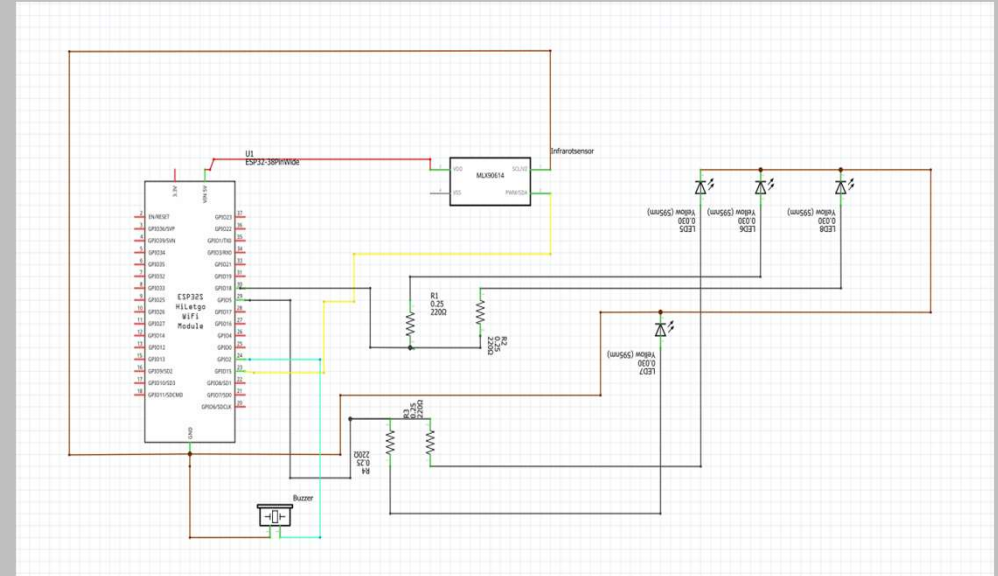
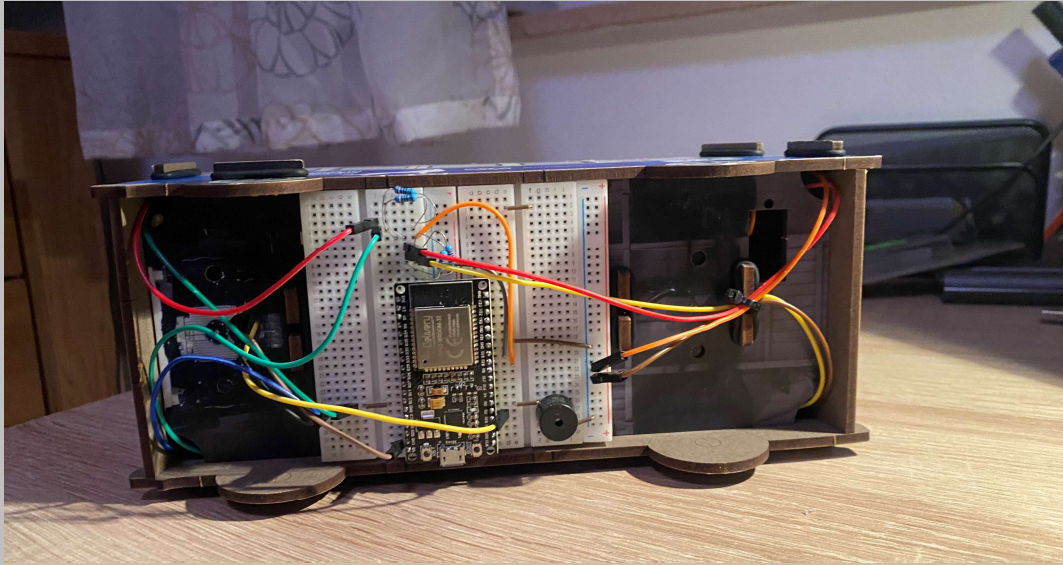
6. Funktionsnachweis



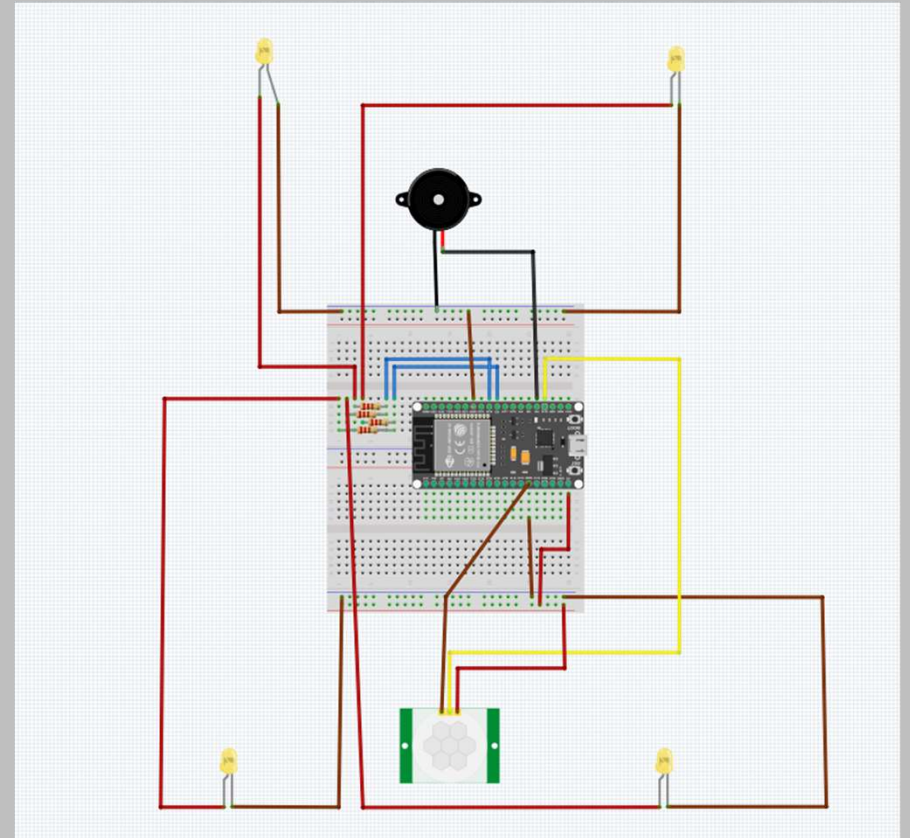
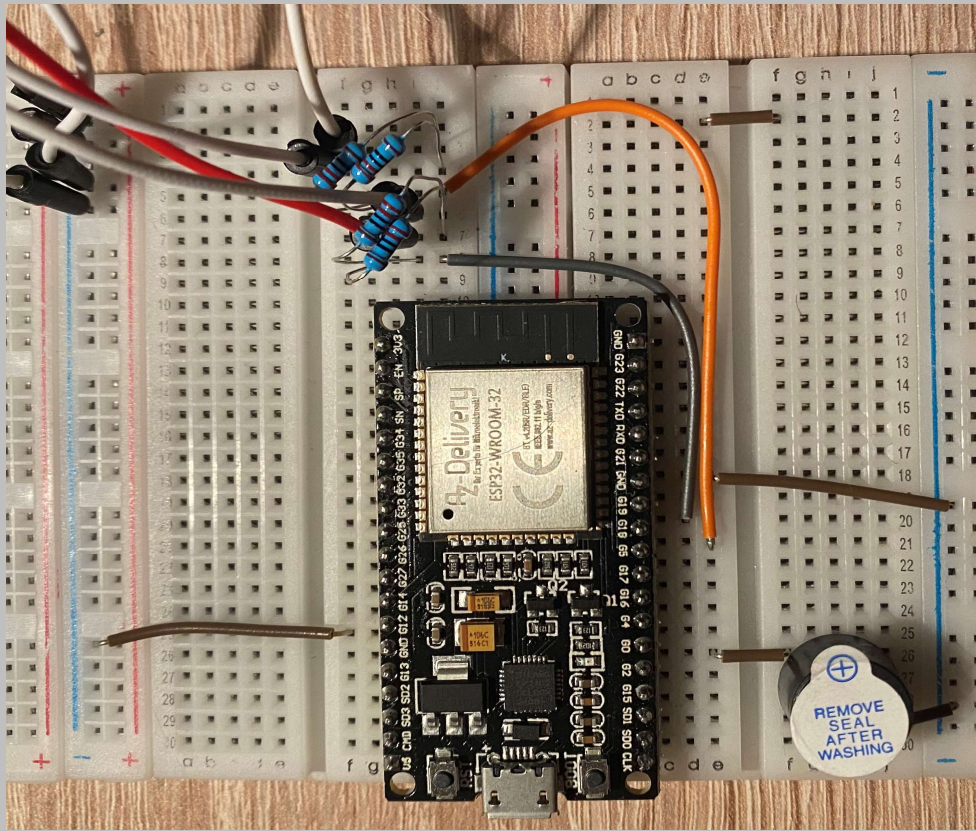
Bewegungssensor

```
1 int bewegungssensor = 15;           // Hier wird die Pinnr. vom ESP 32 auf Bewegungsstatus definiert.
2 int bewegungsStatus = 0;
3 int alarmLED = 5;                   // Hier wird die Pinnr. vom ESP32 auf Alarm LED definiert.
4 int alarmLED2 = 18;                 // Hier wird die Pinnr. vom ESP32 auf Alarm LED2 definiert.
5 int alarmBuzzer = 2;                // Hier wird die Pinnr. vom ESP32 auf Buzzer definiert.
6
7
8 void setup()
9 {
10  pinMode(alarmLED2, OUTPUT);        // Hier wird der Pin alarmLED2(18) des ESP 32 als Ausgangssignal definiert.
11  pinMode(alarmLED, OUTPUT);         // Hier wird der Pin alarmLED(5) des ESP 32 als Ausgangssignal definiert.
12  pinMode(alarmBuzzer, OUTPUT);      // Hier wird der Pin alarmBuzzer (2) des ESP 32 als Ausgangssignal definiert.
13  pinMode(bewegungssensor, INPUT);   // Hier wird der Pin bewegungssensor (15) des ESP 32 al Eingangssignal definiert.
14  Serial.begin(9600);                // Initialisiert die serielle Kommunikation mit der Geschwindigkeit in Baud.
15
16 }
17
18 void loop()
19 {
20  alarm();
21 }
22
23 void alarm()
24 {
25  bewegungsStatus = digitalRead(bewegungssensor);
26
27
28  if (bewegungsStatus == HIGH)
29  {
30    Serial.print("Alarm");
31    for (int i = 0; i < 3; i++)
32    {
33      digitalWrite(alarmLED, HIGH);
34      digitalWrite(alarmLED2, HIGH);
35      digitalWrite(alarmBuzzer, HIGH);
```

7.1 Arduino Sketch



7.2 Fritzing Schaltplan

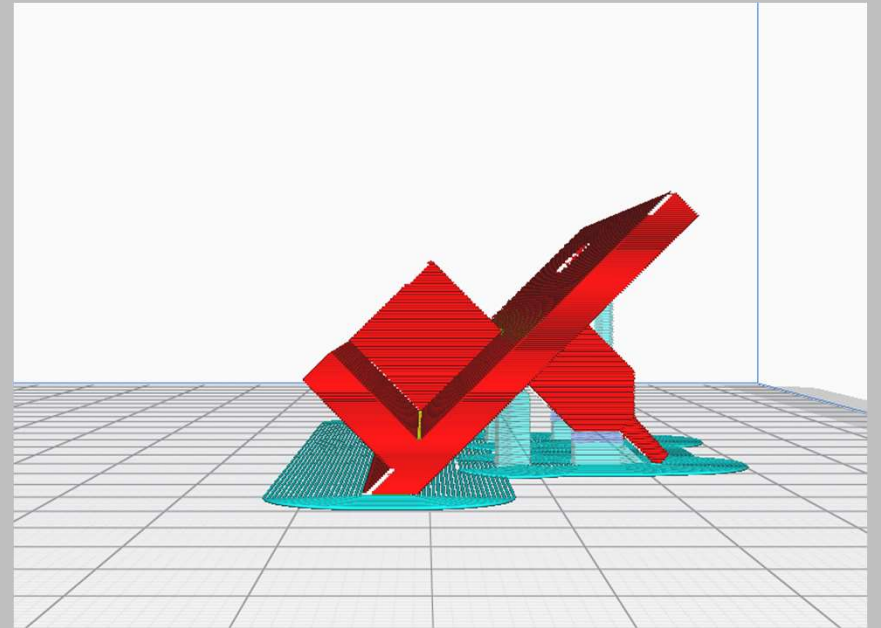
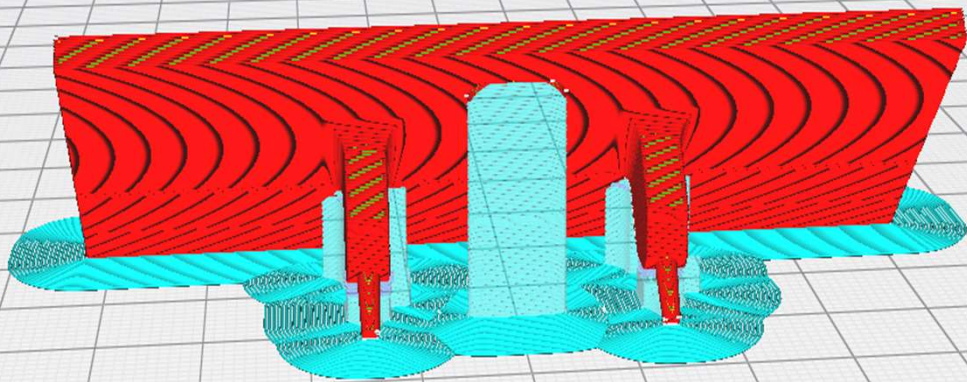


7.3 Fritzing Steckplatine

7.4 Materialliste

- Mikrocontroller (z.B. ESP32, Arduino)
- Breadboard
- Jumper Kabel
- 4 gelbe LED mit passendem Vorwiderstand
- Infrarotsensor (HC-SR501)
- Sensorhalterung (Erstellung mit 3D-Druck, siehe "Sensorhalterung")
- Lautsprecher
- Micro-USB Kabel zur Spannungsversorgung und Programmierung des Mikrocontrollers

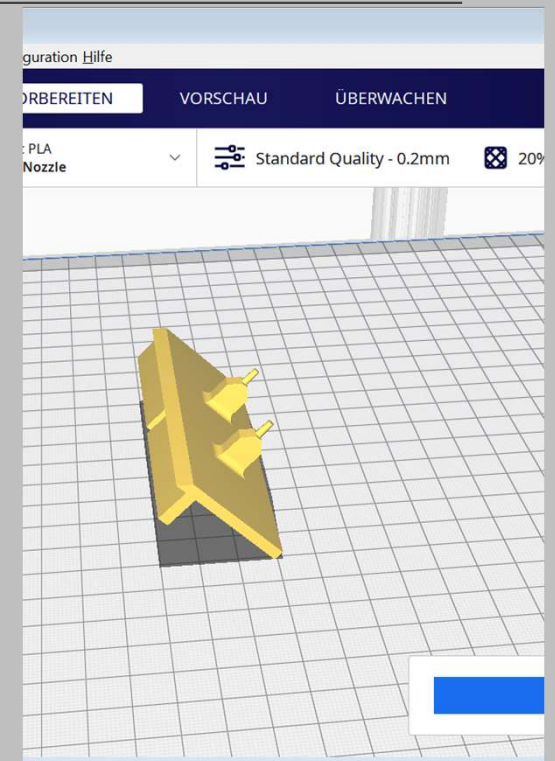
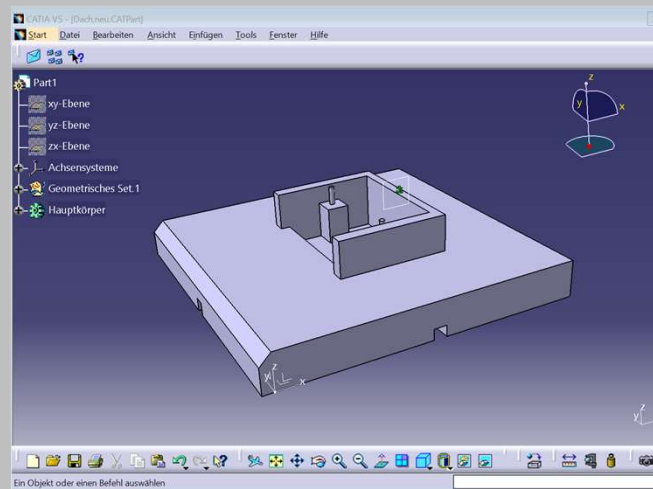
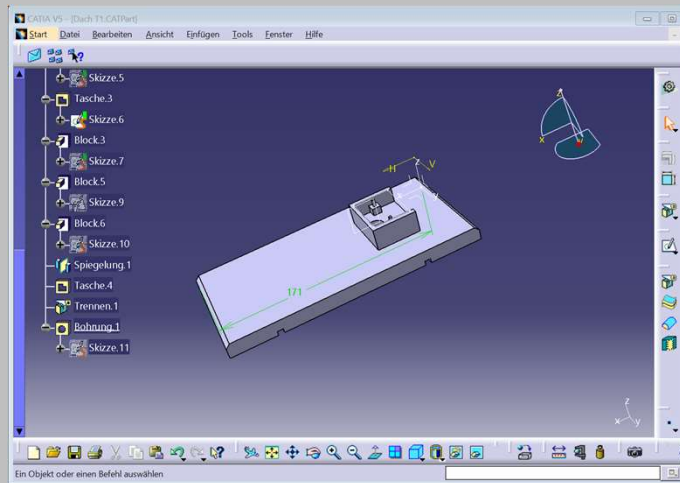
7.5 Slicing Program

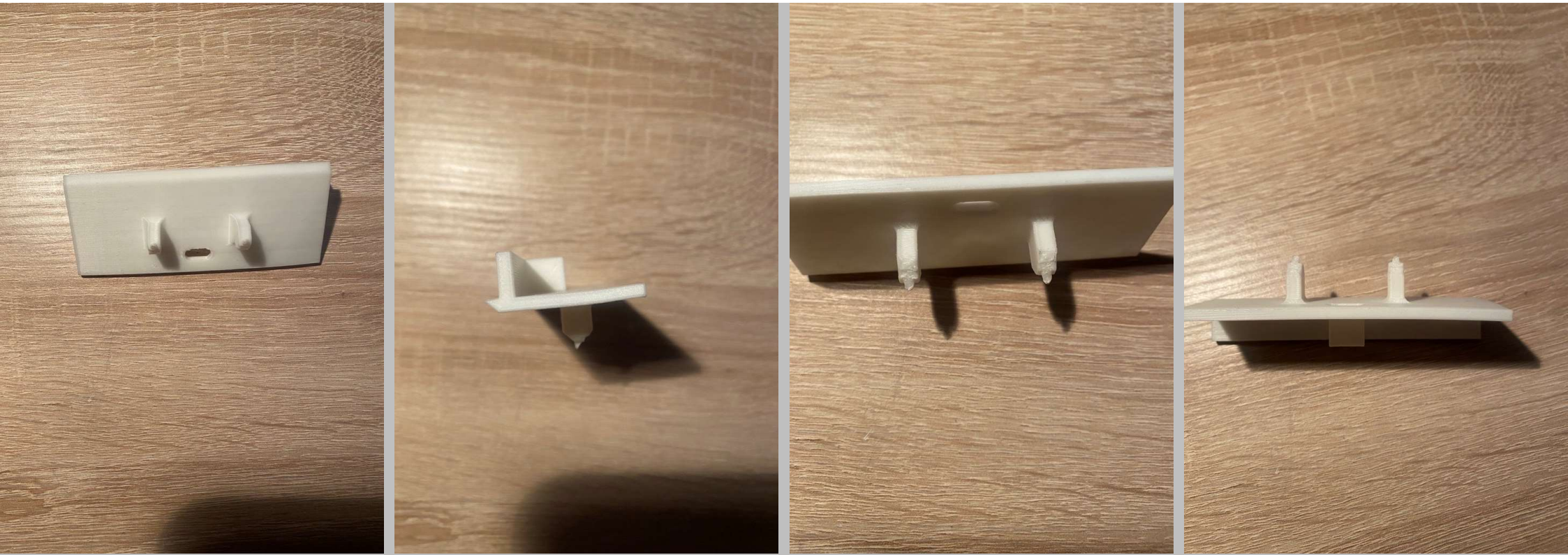


8. Aufgetretene Probleme

- anfänglich durch mangelhafte Kompetenzen im Umgang mit den Programmen (Arduino,Catia...)
- Sensorhalterung wurde angepasst (Änderung des Sensors seitens des Herstellers)

8. Konstruktion Sensorhalterung

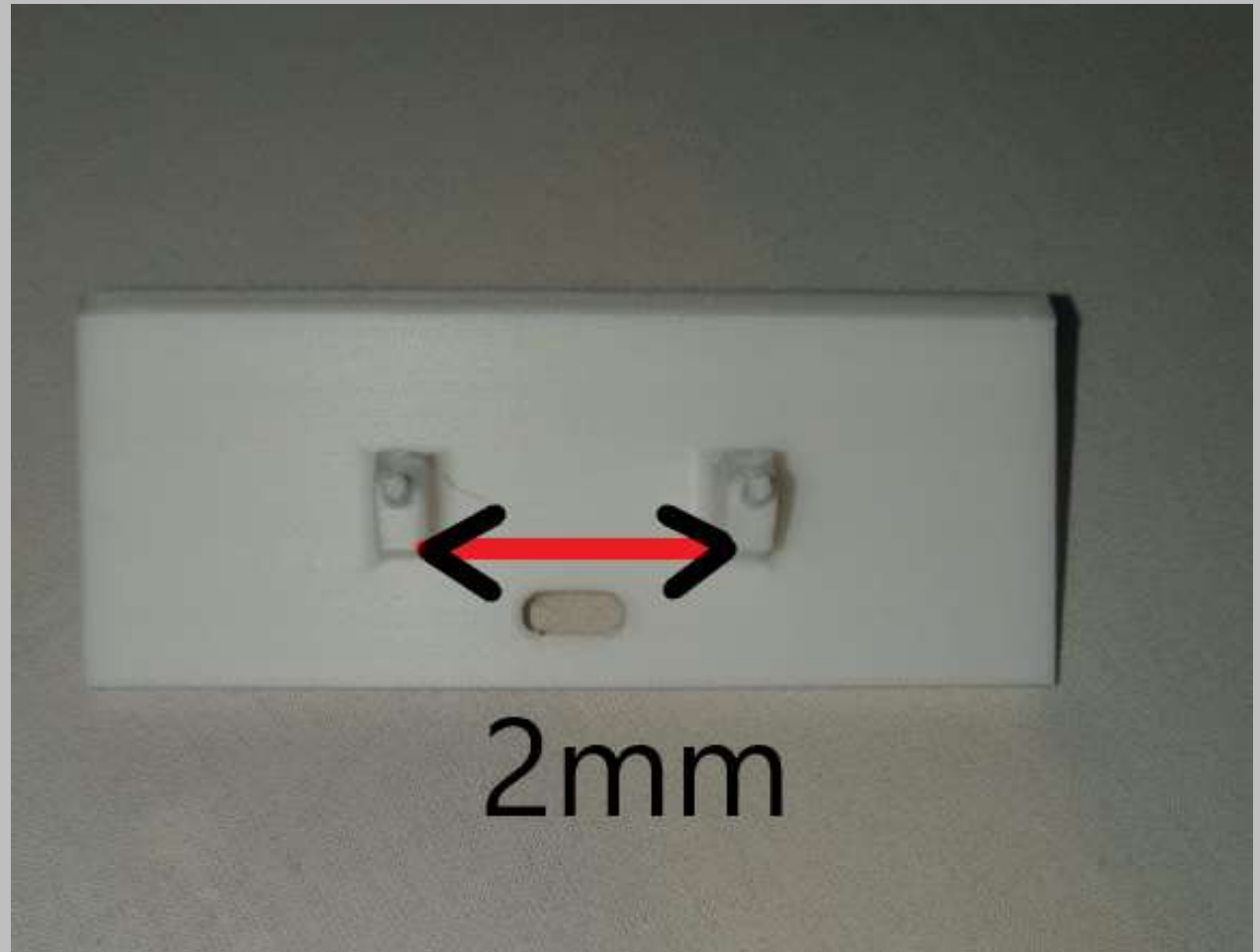




8. Fehlerhafter Sensorhalter

8. Geänderter Sensorhalter

- Skalierung wurde geändert
- 2 Versuche mit 1mm und 2mm



9. Quellen/ Datei Anhang

Projekt 3 : T1-Sensors

<https://www.xplore-dna.net/course/view.php?id=163>

- Fotodokumentation
- Geänderter Sensorhalter
- Arduino Sketch
- Fritizing Schaltplan
- Projektstatusberichte
- Projektannahmen

-Alle Dateien befinden sich in dem Ordner Schüler-Sammelunterlagen unter Projektteam A1

Link: <https://bbs2wob.de/iserv/file/-/Groups/FST21/Sch%C3%BCler-%20Sammelunterlagen/Projektarbeit%20Modul%2002%20%28T1%29/A1.%20Henrik%2CSahin%2CDeannis>

10. Fragerunde



"Dieses Foto" von Unbekannter Autor ist lizenziert gemäß [CC BY-NC-ND](#)