

Projektbericht „LABS for CHIPS“ – Unser Tracking-Fan

1. Projektteilnehmer (Dr. Jäckel)

Entwickler der Projektidee und ihrer Beschreibung ist der Oberstufenschüler Jakob Seifert. Jakob und seine 31 Mitschüler (2 Informatikkurse des grundlegenden Anforderungsniveaus der 11. Jahrgangsstufe) verwirklichten gemeinsam mit ihrer Lehrperson im Schuljahr 2021/22 das Projekt „Tracking – Fan“ im für sie neuen Lernbereich der „Technischen Informatik“.

Für die Umsetzung des Projekts verfügte das Staatliche Gymnasium „Dr. Konrad Duden“ außerhalb der Peripherie der gut ausgestatteten PC-Kabinette jedoch über keinerlei Materialien.

2. Projektbeschreibung (Dr. Jäckel)

Der Tracking-Fan ist ein Ventilator, der die Bewegung einer Person nachverfolgt und somit immer die ideale Position für die beste Erfrischung ganz von selbst ermittelt. Spezialität des Tracking-Fans ist die automatische Anpassung der Drehgeschwindigkeit des Lüfters, je nachdem, wie weit man von ihm entfernt sitzt. Daraus resultierten viele Vorteile gegenüber einem handelsüblichen Ventilator. Zusätzlich kann der Tracking-Fan über ein USB-Kabel mit Strom versorgt werden, sodass für die Inbetriebnahme eine Powerbank oder den PC, an dem man gerade arbeitet, ausreichend ist.

Laut Vorstellung der Lernenden kann der Ventilator im Sommer bei den Hausaufgaben helfen, wenn man an einem Schreibtisch sitzt und sich ein wenig bewegt oder im Zimmer kleine Arbeiten erledigt. Somit fungiert er primär als kleines und selbst entwickeltes cooles Gadget.

Darüber hinaus bauten die Lernenden selbst ein Informatiksystem, welches sie aus ihrer Lebenswelt bereits kennen, nach und optimierten es sogar. Hierzu mussten sie sich aktiv und intensiv mit dem Lerngegenstand auseinandersetzen, ihn zunächst analysieren, in seiner Funktionalität neu entwerfen, die Hardware hierfür erstellen und zusammenbauen und die Software implementieren.

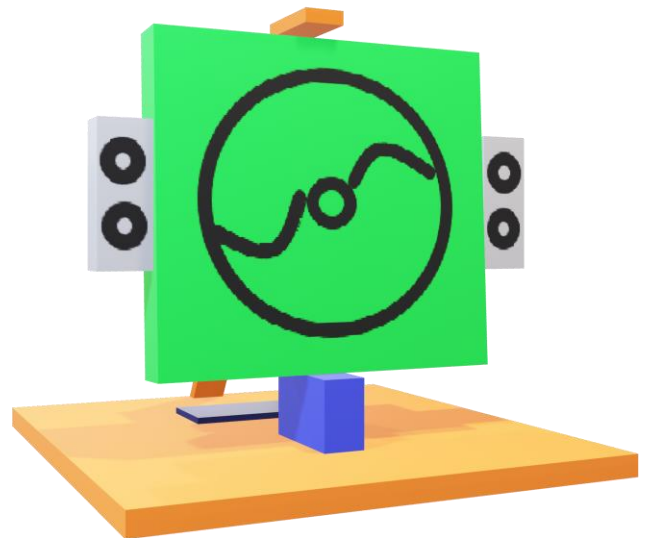


Abbildung 1 3D-Grafik als Projektskizze (entwickelt von Jakob Seifert)

3. Liste der Materialien und Bauteile (Juliane Pätz)

Für die Umsetzung unseres Projekts beantragen wir beim Wettbewerb „Invent a Chip“ eine Unterstützung in Höhe von 2.000 Euro, für welche wir die folgenden Materialien und Bauteile anschaffen konnten:

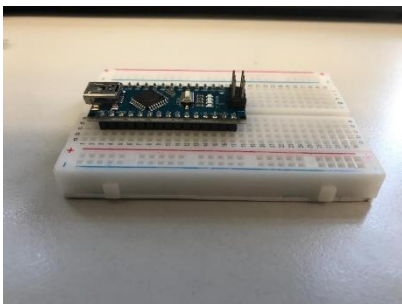
Stückzahl	Bauteil
1	Grundplatte (Holz)
1	Servo
1	Lüfter („be quiet“)
2	Stützen (Holz)
1	Breadboard
1	Arduino Nano (im weiteren als „Arduino“ bezeichnet)
1	Voltage Booster
1	Mosfet
1	Diode
1	220Ω Widerstand
1	10 kΩ Widerstand
2	Ultraschallsensoren
2	Jumperwires (à 4 Kabelstränge)
3 bzw. 11*	Jumperwires (einzeln)
1	Reißzwecke

*) Entscheidend für die Anzahl (also ob 3 oder 11), sind die Kabelenden der Jumperwires, die mit den Ultraschallsensoren verbunden werden.

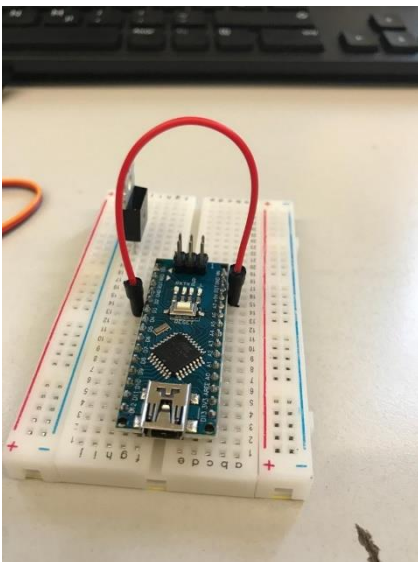
4. Bau des Tracking Fans (Anton Hüttner)



Zuerst wird das Plastikteil aufgesetzt. Im Anschluss wird das bewegliche Plastikteil in die Mittelstellung gebracht. Zuletzt wird die Schraube zur Befestigung des Plastikteils angebracht.

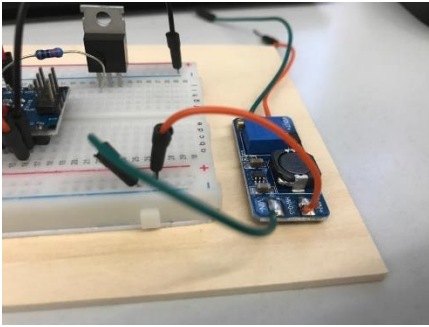


Danach befestigt man den Arduino Nano auf dem Breadboard. Damit können nun die verschiedenen Jumperwires am richtigen Pin für den Arduino gesteckt werden, da am Breadboard, die Leiter senkrecht zum Arduino für die Pins stehen.

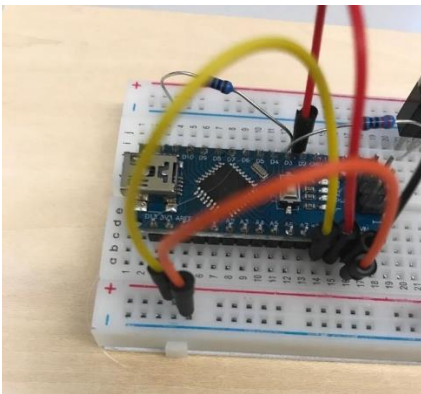


Mosfet auf Breadboard anbringen.

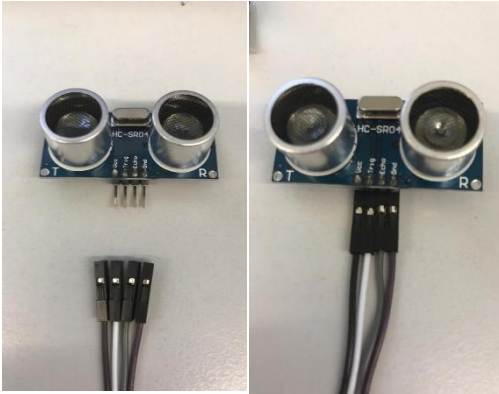
Der Reset wird mit einem Jumperwire auf den Pin2 gelegt.



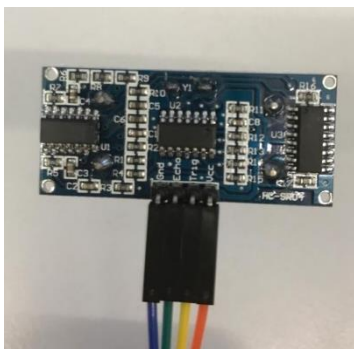
Das angelötete Jumpwire des Vin- Anschluss musste auf die Minuspol-Reihe gelegt werden und das Jumpwire des Vin+ musste auf die Pluspol-Reihe des Arduino gelegt werden.



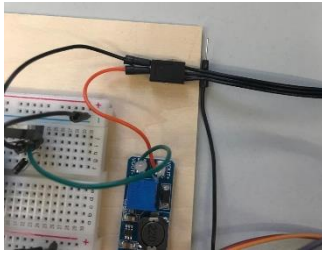
Der Ground-Pin wurde mithilfe eines Jumpwires zur Minuspol-Reihe gelegt und der +5V-Pin, welcher die Stromversorgung des Arduino darstellt, wurde mithilfe eines Jumpwires zur Pluspol-Reihe gelegt.



Als nächstes mussten die 4-Pin-Kabel mit den Anschlüssen der beiden Ultraschallsensoren verbunden werden.



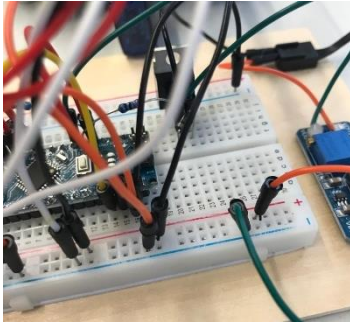
Wichtig war zudem bei beiden Ultraschallsensoren die Anschlüsse samt Kabeln um 90 Grad zu biegen.



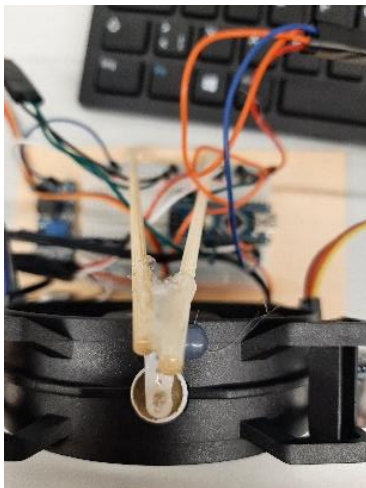
Lüfteranschluss wurde auf Grundplatte geklebt.



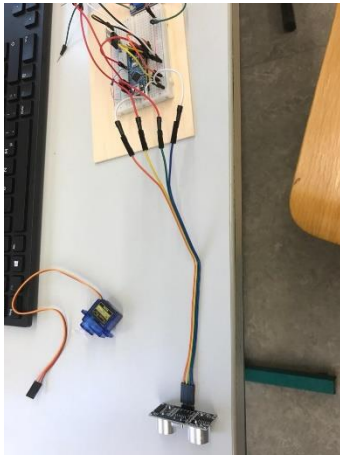
Ultraschallsensoren wurden an Lüfter geklebt.



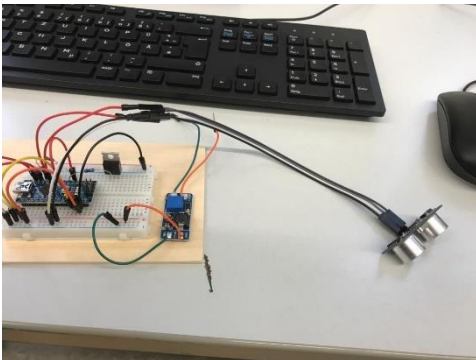
Signal-Pin des Servos wurde mit Pin 4 des Arduino verbunden und der VCC-Pin des Servos wurde mit der Pluspol-Reihe des Arduino verbunden. Außerdem wurde der Ground-Pin des Servos mit der Minuspol-Reihe des Arduino verbunden.



Zuletzt wird auf der Oberseite des Lüfters, welcher außerdem auf die Grundplatte geklebt wurde, eine Reißzwecke angeklebt. An ihr und der Grundplatte werden mit Heißleim zwei Holzstäbe befestigt, um zu verhindern, dass der Lüfter umkippen kann.



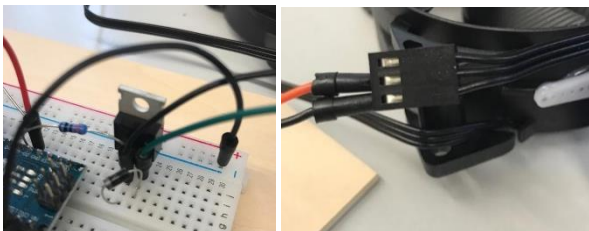
Nun wurde Ultraschallsensor 1 angeschlossen, dabei wurde der Triggerpin durch ein Jumperwire mit Pin 11 am Arduino verbunden. Anschließend wurde der Echopin durch ein Jumperwire mit Pin 11 am Arduino verbunden. Weiter wurde der VCC-Pin mit der Pluspol-Reihe verbunden und zuletzt noch der Ground-Pin des Ultraschallsensors mit der Minuspol-Reihe des Arduino.



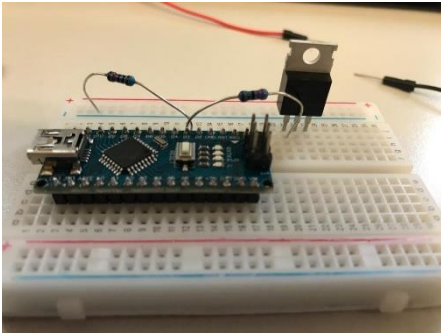
Nun wurde Ultraschallsensor 2 angeschlossen, dabei wurde der Triggerpin durch ein Jumperwire mit Pin 9 am Arduino verbunden. Anschließend wurde der Echopin durch ein Jumperwire mit Pin 8 am Arduino verbunden. Weiter wurde der VCC-Pin mit der Pluspol-Reihe verbunden und zuletzt noch der Ground-Pin des Ultraschallsensors mit der Minuspol-Reihe des Arduino.



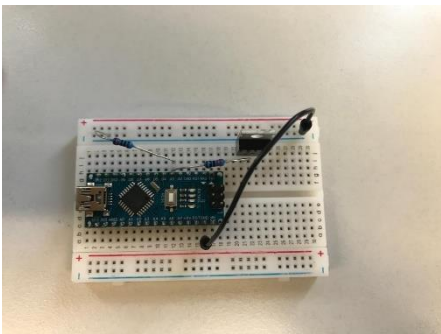
Der Servo wurde an der Unterseite des Lüfters mit Heißleim verklebt.



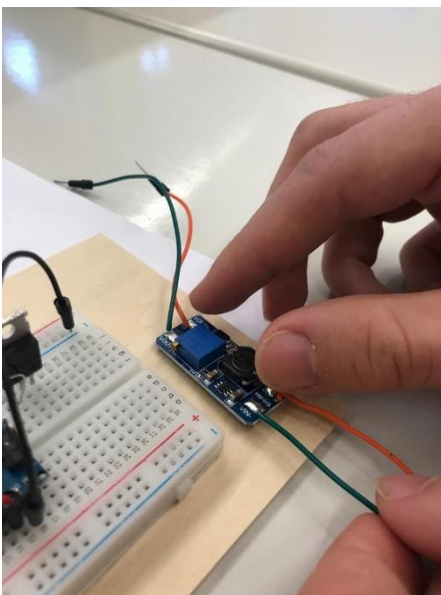
Nun wurde zuerst einmal der VOUT- -Pin mit dem Source-Pin des Mosfet durch ein Jumperwire verbunden. Der VOUT+-Pin des Voltageboosters wurde durch ein Jumperwire mit dem Pluspol-Anschluss des Lüfters verbunden. Weiter wurde der Drain-Pin des Mosfet durch ein Jumperwire mit dem Minuspol-Anschluss des Lüfters verbunden. Als letztes wurde eine Diode vom Source-Pin zum Drain-Pin des Mosfet auf dem Breadboard angebracht.



Nun wird ein 220Ω Widerstand vom Ground-Pin des Mosfet zum Pin D3 gelegt. Zusätzlich wird ein 10kΩ Widerstand vom Pin D3 zur Minuspolreihe gelegt.



Mithilfe eines Jumperwires wird der Ground-Pin zur Minuspol-Reihe gelegt.



Bevor der Voltage-Booster auf das Brett geklebt wurde, wurde das Breadboard samt Arduino, Widerständen, Jumpwires und Mosfet auf das Brett geklebt.

5. Kommentierter Quellcode für den Arduino-Nano (Jakob Seifert, Mathis Wetzel)

```
#include<Servo.h>// Bibliothekimport für Servo  
  
#include <NewPing.h> // Bibliothekimport für Ultraschallsensoren  
  
NewPing sonar_r(11, 10, 400); //(Trigger rechter Ultraschallsensor, Echo  
rechter Ultraschallsensor,  
Reichweite in cm) → Belegung des Pins
```

```

NewPing sonar_l(9, 8, 400); //(Trigger linker Ultraschallsensor, Echo
linker Ultraschallsensor,

Reichweite in cm) → Belegung des Pins

Servo achse; // Servo wird „achse“ genannt

int pos =0; // int=Variable vom Typ Integer, die die Position angibt

void setup() { // ab hier beginnt der Setup-Teil → Ausführung einmalig

  achse.attach(4); //Definiere, dass Servo an dem digitalen Pin 4
angeschlossen wird

  pinMode (3,OUTPUT); // definiere Pin 3 als Output, um Lüfter darüber
anzusteuern

  Serial.begin(9600); // Beginn der Seriellen Schnittstelle mit Arduino
(Baudrate = 9600)

  delay(50); // 0,05 Sekunden Pause

  pos = 87; // Position des Servomotors wird auf Mittelstellung gesetzt
(87°)

  achse.write(pos); // die gesetzte Position jetzt auch umsetzen → erst
jetzt bewegt sich der Servo
}

void loop() { // Beginn des Loop- Teils → stetige Wiederholung

  Serial.print("Abstand l: "); // über den Seriellen Monitor das Wort
„Abstand l: “ ausgeben

  Serial.println(sonar_l.ping_cm()); // nach dem ausgegebenen Wort den
entsprechenden Wert des Sensors

  ausgeben

  Serial.print("Abstand r: "); // ebenso für den rechten Ultraschallsensor

  Serial.println(sonar_r.ping_cm()); // ebenso für den rechten
Ultraschallsensor

  Serial.print("pos: "); // über den Seriellen Monitor das Wort „pos: “
ausgeben (pos = Position)

  Serial.println(pos); // hinter dem ausgegebenen Wort den entsprechenden
Wert des Servos ausgeben

  int abstand = (sonar_l.ping_cm() + sonar_r.ping_cm()) / 2; //den
Durchschnittsabstand der beiden Sensoren

  errechnen und in der Variablen „abstand“ speichern

  abstand = map(abstand, 0, 40, 0, 255); //das Verhältnis des Abstandes (0cm
bis 40cm) auf einen Wert von 0 bis

  255 übertragen

  analogWrite(3, abstand); // dieser Wert entspricht der Drehgeschwindigkeit
des Lüfters und wird an diesen

  durch den Pin 3 übermittelt → Anpassung der Drehgeschwindigkeit an
Abstand zum Lüfter

// jetzt fangen verschiedene Fälle an, die für die eigentliche
Bewegungserfassung benötigt werden

  if (pos > 124){ // es wird verhindert, dass der Lüfter an die
Holzstäbchen stößt

```

```

pos = 124; // → wenn die Position den Winkel von 124° überschreiten
würde, dann wird die Position wieder auf

    124° zurückgesetzt

achse.write(pos); // der Winkel wird auf den Servo übertragen
//delay(100); //optionale Pause für 0,1 Sekunden
}

else if (pos < 51){ // es wird verhindert, dass der Lüfter an die
Holzstäbchen stößt

    pos = 51; // wenn die Position den Winkel von 51° unterschreiten würde,
dann wird die Position wieder auf

        51° zurückgesetzt

achse.write(pos); // der Winkel wird auf den Servo übertragen

// delay(100); // optionale Pause für 0,1 Sekunden
}

else if ((sonar_l.ping_cm() ) > (sonar_r.ping_cm() )){ //Rechtsdrehung:
wenn die ermittelte Entfernung des linken

Ultraschallsensors größer ist, als der ermittelte Abstand des rechten
Ultraschallsensors, dann folgendes:

        pos = pos + 5; // die Variable der Position wird um 5° vergrößert →
Rechtsdrehung um 5°

achse.write(pos); // Übertragung der Variable „pos“ auf den Servo

        delay(10); // Pause für 0,01 Sekunden

    }

else if ((sonar_l.ping_cm() + 5) < (sonar_r.ping_cm() + 5)){
//Linksdrehung: wenn die ermittelte Entfernung des

linken Ultraschallsensors kleiner ist, als der ermittelte Abstand des
rechten Ultraschallsensors, dann folgendes:

        pos = pos - 5; // die Variable der Position wird um 5° verkleinert →
Linksdrehung um 5°

achse.write(pos); // Übertragung der Variable „pos“ auf den Servo

        delay(10); // Pause für 0,01 Sekunden

    }

else if (sonar_l.ping_cm() == 0 && sonar_r.ping_cm() == 0) { //wenn die
ermittelte Entfernung der Sensoren zum

Objekt gleich groß ist, dann folgendes:

        delay(10); // → nur eine Pause von 0,01 Sekunden, mit dem Ziel in der
Stellung zu bleiben

    }

        delay(100); // im kompletten Vorgang der Überprüfung der verschiedenen
Fälle eine Pause für 0,1 Sekunden

}

```


6. Kritische Betrachtung und Feedback (Noah Pahlke)

a) Mögliche Erweiterungen

Man könnte die Stärke des Lüfters so an den Abstand zum Objekt koppeln, dass dieses immer einen gleichstarken Luftzug erfährt. Dazu wird der mittlere Abstand der beiden Sensoren zum nächsten Objekt ermittelt und dann einem proportional dazu wachsenden Wert für die Windstärke zugeordnet. Außerdem könnte man beispielsweise eine LED an einen Pin so anschließen, dass diese immer leuchtet, wenn der Tracking Fan in der perfekten Ausrichtung ist. Dazu wird ein Signal weitergegeben, welches die Stromversorgung der LED ermöglicht, wenn die Werte der Sensoren sich annähern, und dann die Stromversorgung wieder kappt, wenn die Werte auseinandergehen.

b) Kritische Reflexion

Der Tracking Fan ist weitgehend zuverlässig, da er in seinen gegebenen Grenzen die Aufgaben erfüllt, d.h. lüften und ausrichten, und Software und Hardware gut zusammenarbeiten. Wenn die Sensoren zwei unterschiedliche, aber gleich nahe Objekte messen, kann sich die Software nicht auf eines festlegen und der Servo bewegt den Lüfter zwischen den beiden hin und her. Wenn ein Kabel einen Wackelkontakt hat oder nur lose im Breadboard oder anderen Bauteilen steckt, führt dies zum vollständigen Verlust der Funktionalität des Tracking Fans. Die Funktionalität des Tracking Fans kann auch beeinflusst werden, wenn Bauteile z.B. Kabel oder Stützen „im Weg“ für die Drehung des Lüfters sind. In dem Fall verringert sich der Winkel, mit dem der Lüfter lüften kann.

Die Nutzerfreundlichkeit des Tracking Fans ist auf jeden Fall gegeben, da die Software konstant auf dem Arduino gespeichert ist und er diese sofort ausführt, wenn der Tracking Fan mit Strom versorgt wird. Allerdings ist der Lüfter zwar leise, aber die Drehungen des Servos und der Minicomputer selbst können Störgeräusche verursachen.

c) Feedback

Ich fand das Projekt sehr abwechslungsreich im Vergleich zum sonstigen Unterricht und vor allem den Einblick in die technische Seite, mit der ich mich so noch nie befasst hatte, sehr aufschlussreich.

7. Dank an unsere Förderer

Ein sehr herzliches Dankeschön gilt dem gesamten Invent a Chip-Team und seinen Sponsoren für die Organisation und Ausrichtung des Wettbewerbs. Hierdurch konnten die Elftklässler der Informatikkurse des Staatlichen Gymnasiums „Dr. Konrad Duden“ Schleiz einen sehr vielfältigen und lebenspraktischen Einblick in die Grundlagen des Inhaltsbereichs „Technische Informatik“ erlangen.

Dr. Stefanie Jäckel,
Fachschaftsleiterin Informatik,

im Namen der Informatikkurse des Staatlichen Gymnasiums „Dr. Konrad Duden“ Schleiz