



Timo Feuerstein Filmon Russom Noah luc
Lashley

24. Mai 2022

Myclimate ZABU Projekt

Projekt: Myclimate



Inhaltsverzeichnis

Idee 2

Planung..... 2

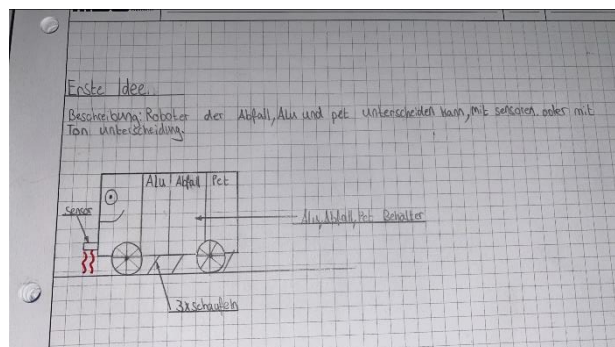
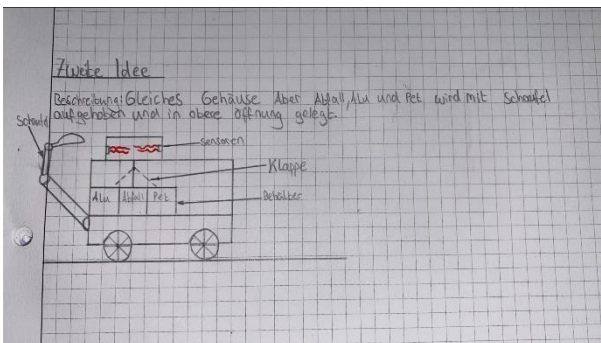
Realisierung..... 2

Programmierung..... 4

MSW W.A.L.L.E

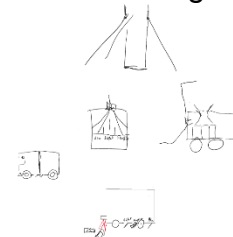
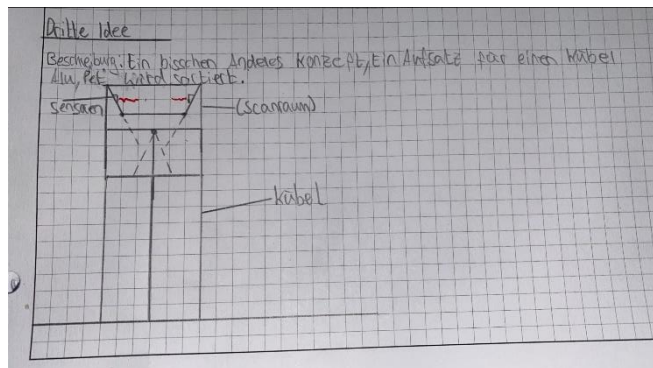
Idee

Wir haben uns zusammengesetzt und begonnen uns zu überlegen was wir als Myclimate Projekt machen könnten. Zuerst hatten wir die Idee einen fahrenden Müllroboter zu planen und zu bauen, welcher mit einem Greifarm den Müll vom Boden aufnehmen kann ihn danach scannt und ihn in Alu oder PET aufteilen kann. Wir haben uns überlegt ob wir dieses Projekt überhaupt durchführen können und Filmon war sehr davon überzeugt das wir es verwirklichen können so haben wir begonnen.



Planung

Zu Beginn waren wir noch sehr überzeugt das wir einen Roboter mit Greifarm machen könnten, welcher sich auch noch bewegt. Doch nach den ersten paar Stunden wurde uns klar, dass ein Roboter mit Greifarm und einer KI gesteuerten Bewegung unsere Fähigkeiten überschreiten würde. So haben wir uns noch einmal zusammengesetzt und uns überlegt was wir weglassen könnten das es für unsere Fähigkeiten immer noch möglich wäre aber auch nicht allzu viel von unserer Grund Idee abweicht. So sind wir dann auf die Idee gekommen einen Aufsatz für einen Mülleimer zu konstruieren welcher Alu und PET voneinander trennen kann, da die Menschen oft zu faul sind allen Abfall voneinander zu trennen und man so nur einen Abfalleimer Eingang bräuchte, um beides zu entsorgen und so hoffentlich mehr Alu und PET richtig zu trennen und so die Umwelt ein wenig zu schützen.

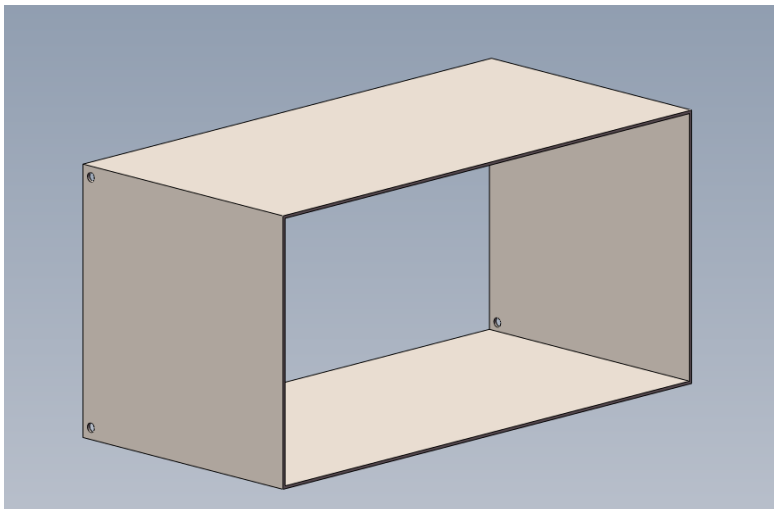


Realisierung

Wir haben uns in zwei Gruppen aufgeteilt, Filmon hat die Programmierung von Sensor übernommen und Ich und Noah haben die Planung und Konstruktion übernommen. Zuerst mussten Noah und ich die groben Maße

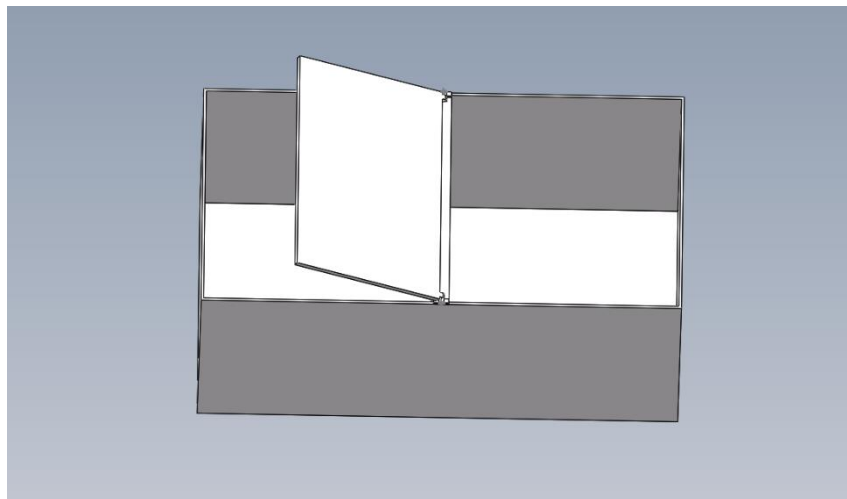
bestimmen und uns überlegen, wie wir alles herstellen sollten und aus welchem Material alles gebaut werden soll. Wir haben uns entschieden das wir unseren kleinen Roboter aus Aluminium bauen wollen, so haben wir begonnen auf dem Konstruier Programm unsere Teile zu konstruieren, Auszumessen und zu 3D Modellen zu machen.

Als wir alle Teile fertig konstruiert hatten, haben wir die Zeichnung erstellt, was wir alles im gleichen Programm machen konnten. Mit diesen Zeichnungen sind wir in die Werkstatt gegangen, um unsere Werkstattlehrer zu fragen ob das alles so möglich wäre herzustellen. Wir haben zuerst verschiedene Antworten bekommen doch am ende hatten wir uns auf eine Vorgehensweise geeinigt. Wir werden unsere Teile mit Aluplatten herstellen, indem wir sie biegen und dann zusammen Nieten. So können wir das außen Gerüst machen.



Um den Abfall dann zu trennen, müssen wir eine

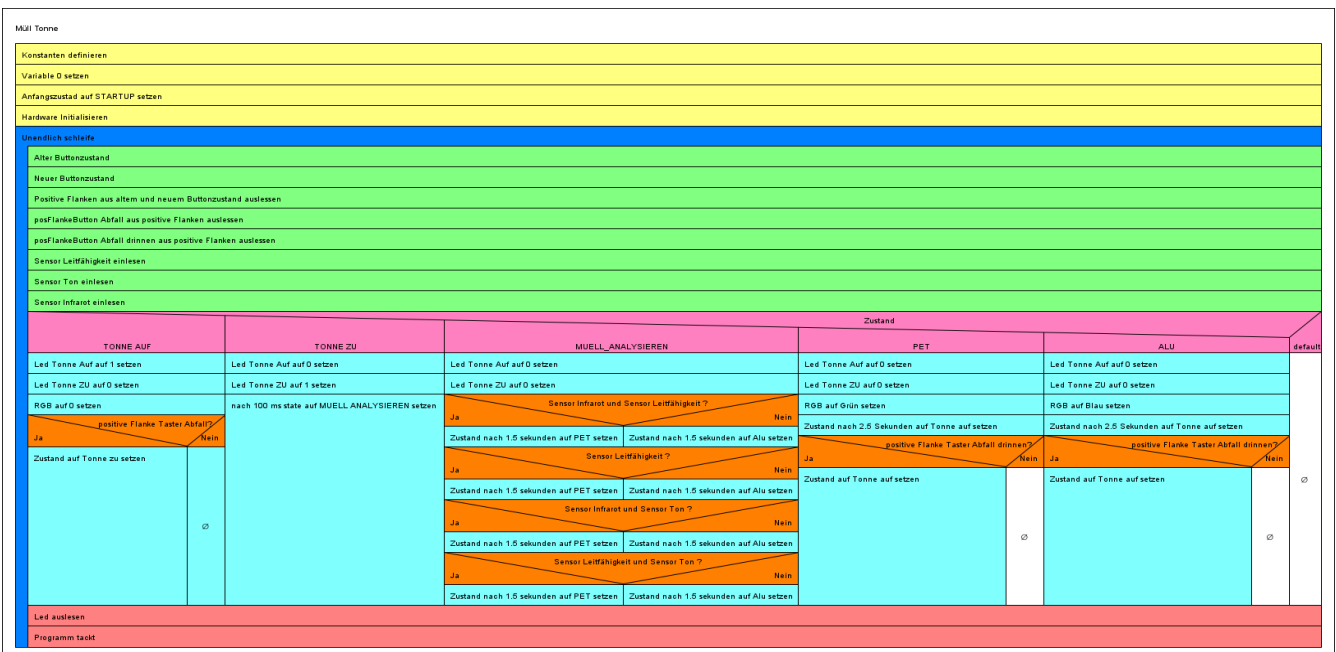
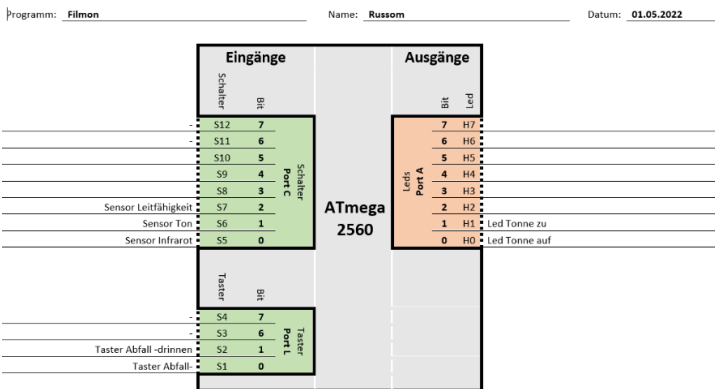
Klappe bauen, welche von links nach rechts und wieder zurückklappen kann. So kann der Abfall zuerst gescannt und danach in die richtige Klappe fallen.



Programmierung

Ich war für das Programmieren zuständig. Ich habe mir überlegt wie ich PET und Alu voneinander trennen kann, also habe ich mir überlegt was eigentlich die Eigenschaften von PET und Alu sind, dann bin ich auf die Idee gekommen das PET durchsichtig ist also kann ich mit Infrarotlicht den PET scannen und wenn das Licht auf der anderen Seite rauskommt, dann ist es PET und wenn nicht ist es Alu. Es war zu unsicher denn wenn es blöd kommt und das infrarot Licht auf der Etiketle landet und das Licht nicht durchkann, dann wird es auch als Alu erkannt. So habe ich noch nach anderen Möglichkeiten gesucht und bin auf die Idee gekommen ich könnte es durch Töne unterscheiden denn PET und Alu machen verschiedene Töne, wenn sie fallen und eine andere Möglichkeit wäre da Alu leitfähig ist könnte ich testen ob es leitet, wenn es leitet, ist es Alu und wenn nicht dann ist es PET. Da ich leider nicht die Möglichkeit hatte diese Variante zu verwirklichen habe ich was ich alles in der Lehre gelernt habe eingesetzt ich habe zuerst ein Struktogramm erstellt, das ist eine Struktur der Code beschreibt das Programm einfacher, danach habe ich noch eine Hardwarestruktur gemacht das ist dafür da bei meinen Mikrokontroller zu sehen welcher Schalter für was zuständig ist. Nachdem ich das alles gemacht hatte, habe ich mit dem Programmieren angefangen, ich habe mit Microchip Studio c programmiert als erstes habe ich die Schalter und LED's definiert und eingelesen

danach habe denn Programm geschrieben danach habe ich es auf meinen Mikrokontroller hochgeladen und getestet.





```

/*****\
*
* MMMMMMMMMMMM  SSSSSSSSSSS  WW  WW  WW  MECHATRONIK
* MM  MM  MM  SS  WW  WW  WW  SCHULE
* MM  MM  MM  SSSSSSSSSSS  WW  WW  WW  WINTERTHUR
* MM  MM  MM  SS  WW  WW  WW
* MM  MM  MM  SSSSSSSSSSS  WWWWWWWWWWWW  www.msw.ch
*
*
* Dateiname: main.c
*
* Projekt  : MSW W.A.L.L.E
* Hardware : uC-Board, ATmega2560v von Atmel
*
* Copyright: MSW, E2
*
* Beschreibung:
* =====
* Siehe Prüfungsaufgabe
*
* Portbelegung:
* =====
* Port C:  Schalter
* Port A:  LEDs
* für genauere Beschreibung siehe Hardwarestruktur
*
* Verlauf:
* =====
* Datum:      Autor:      Version  Grund der Änderung:
* 24.05.2022  F. Russom    V1.0    Neuerstellung
*
\*****/

//uC-Board-Treiber hinzufügen
#include "ucBoardDriver.h"
#define TASTER_ABFALL                (0b00000001)
#define TASTER_ABFALL_DRINNEN      (0b00000010)

#define SCHALTER_INFRAROT          (0b00000001)
#define SCHALTER_TON                (0b00000010)
#define SCHALTER_LEITFEAHIGKEIT    (0b00000100)
#define LED_TONNE_AUF              (0x01)
#define LED_TONNE_ZU                (0x02)

#define SYSTEMTACKT_MS              10

typedef enum Zustand_t {TONNE_ZU, TONNE_AUF, PET, ALU, MUELL_ANALYSIEREN, MUELL}
zustand_t;
//Hauptprogramm
int main(void)

```



```
{
//Variablen
uint8_t inTaster=0;
uint8_t inTasterAlt=0;
uint8_t posFlanken=0;

uint16_t schalter_infrarot =0;
uint16_t schalter_ton =0;
uint16_t schalter_leitfaehigkeit =0;
uint16_t taster_Abfall =0;
uint16_t taster_Abfall_drinnen =0;

uint16_t led_Tonne_auf =0;
uint16_t led_Tonne_zu =0;
uint16_t rot = 0;
uint16_t blau = 0;
uint16_t gruen = 0;

zustand_t state = TONNE_AUF;

//Initialisieren
initBoard(1);

//Unendlichschlaufe
while(1)
{
//Eingabe-----
inTasterAlt = inTaster; //Alter Buttonzustand
inTaster = buttonReadAllPL(); //Neuer Buttonzustand
posFlanken = (inTaster ^ inTasterAlt) & inTaster;
taster_Abfall = posFlanken & TASTER_ABFALL;
taster_Abfall_drinnen = posFlanken & TASTER_ABFALL_DRINNEN;
schalter_infrarot = switchReadAll() & SCHALTER_INFRAROT;
schalter_leitfaehigkeit = switchReadAll() & SCHALTER_LEITFEAHIGKEIT;
schalter_ton = switchReadAll() & SCHALTER_TON;
//Verarbeitung-----
switch (state)
{
case TONNE_AUF:
led_Tonne_auf = TONNE_AUF;
led_Tonne_zu = 0;
rot = 0;
gruen = 0;
blau = 0;
lcdLog("Tonne Auf");
if (taster_Abfall)
{
state = TONNE_ZU; //Zustand auf Tonne zu setzen;
}
}
}
}
```



```
break;
case TONNE_ZU:
led_Tonne_zu = TONNE_ZU;
led_Tonne_auf = 0;
lcdLog("Muell drinnen");
_delay_ms(1000);
lcdLog("Tonne Zu");
_delay_ms(100);
state = MUELL_ANALYSIEREN;
break;
case MUELL_ANALYSIEREN:
led_Tonne_zu = 0;
led_Tonne_auf = 0;
lcdLog("Muell analysieren");

if (schalter_infrarot && !schalter_leitfaehigkeit)
{
    _delay_ms(1500);
    state = PET; // auf PET setzen;
}
else if (!schalter_leitfaehigkeit)
{
    _delay_ms(1500);
    state = PET; //PET setzen;
}

else if (!schalter_leitfaehigkeit && schalter_ton)
{
    _delay_ms(1500);
    state = PET; //auf PET setzen;
}
else if (schalter_infrarot && schalter_ton)
{
    _delay_ms(1500);
    state = PET; //auf PET setzen;
}
else
{
    _delay_ms(1500);
    state = ALU; //auf Alu setzen;
}
break;
case PET:
led_Tonne_zu = 0;
led_Tonne_auf = 0;
rot = 0;
gruen = 250;
blau = 0;
rgbWrite(rot, gruen, blau);
lcdLog("PET erkannt");
```




```
    _delay_ms(2500);
    state = TONNE_AUF;
    if (taster_Abfall_drinnen)
    {
        state = TONNE_AUF; //auf Tonne auf setzen;

    }
    break;
    case ALU:
    rot = 0;
    gruen = 0;
    blau = 250;
    led_Tonne_zu = 0;
    led_Tonne_auf = 0;
    rgbWrite(rot, gruen, blau);
    lcdLog("Alu erkannt");
    _delay_ms(2500);
    state = TONNE_AUF;
    if (taster_Abfall_drinnen)
    {
        state = TONNE_AUF; //Zustand auf Tonne auf setzen;

    }
    break;
    default:
    ;
}
//Ausgabe-----
ledWriteAll( led_Tonne_auf | led_Tonne_zu );
rgbWrite(rot, gruen, blau);
_delay_ms(SYSTEMTACKT_MS);
}
}
```