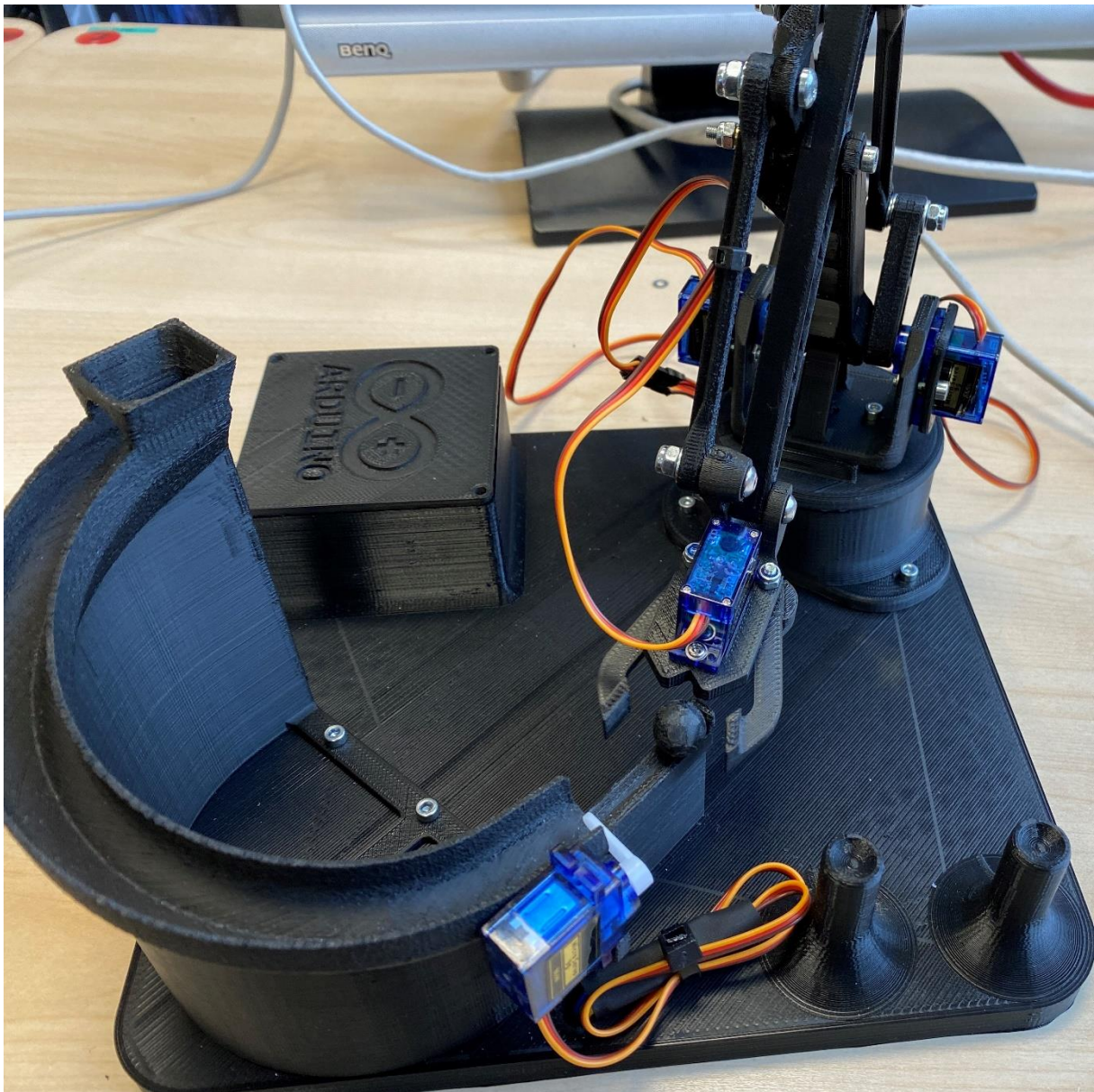


Zusatzqualifikation Programmierung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano



	Prüflingsteilnehmer	Ausbildungsbetrieb
Name	-----	Audi AG
Anschrift	----- -----	Hindemithstr. 27 85057 Ingolstadt
Prüflingsnummer	----	

Zusatzqualifikation Programmierung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

1. Bezeichnung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

2. Auftragsbeschreibung und Zielsetzung

In der Zusatzqualifikation „Programmierung“ wird ein Roboter aus Einzelteilen, welche im 3D-Drucker gefertigt wurden, zusammengebaut, die verbauten Motoren an den Arduino Nano angeschlossen und in der Programmiersprache C programmiert.

Geplant war zuerst das Schreiben eines einzelnen Programmes, doch nach Rücksprache mit dem Trainer haben wir uns dazu entschieden ein Alternativprogramm noch anzufertigen. Das erste Programm, welches der Hauptteil der Arbeit war und das korrekte Anfahren sicherstellen sollte, lässt den Roboter von dem rechten Sockel einen Ball greifen, welchen er zur obersten Stelle der Rampe bringt und ihn dort loslässt, sodass dieser runterrollt. Liegt der Ball unten an der Rampe öffnet die Schranke, welche am Ende verbaut ist und die Kugel rollt dann bis zum Anschlag weiter. Ist der Ball nun zum Stillstand gekommen nimmt sich der Roboterarm den Ball und bringt ihn wieder hoch, um den Ablauf erneut zu starten. Die Bewegung des Greifens am Ende der Rampe und das Loslassen am Anfang der Rampe wird dann dauerhaft erneut ausgeführt bis der Roboter von der Spannungsversorgung getrennt wird. Falls möglich soll das Erkennen vom Ball an der Schranke „intelligent“ funktionieren, dies wäre durch eine Lichtschranke zu realisieren.

Das Alternativprogramm ist ähnlich wie das erste, aber er wiederholt den Vorgang vom Bringen des Balls vom Ende der Rampe zum Anfang nur vier Mal und bringt dann den Ball wieder zurück zum Lager, legt ihn aber auf dem linken Sockel ab.

Zusatzqualifikation Programmierung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

3. Information und Arbeitsplanung

Vor der Durchführung sind die benötigten Arbeitsschritte und die benötigte Zeit zu planen. Zudem müssen alle benötigten Werkzeuge und Bauteile besorgt und vorbereitet werden. Zusätzlich muss das Programm „Arduino IDE“ und die Treiber des Arduino Nanos heruntergeladen und installiert werden.

4. Auftragsdurchführung

Am Anfang werden alle Bauteile des Roboters mithilfe von M4 und M3 Schrauben verschraubt. Ist der Roboterarm fertig zusammengebaut wird dieser an der Platte befestigt und die Motoren mit dem Arduino Nano verbunden. Dann wird der Mikrocontroller mit dem Laptop verbunden und das Programm „Arduino IDE“ geöffnet.

Dort stelle ich erst ein welchen Mikrocontroller ich verwende und auf welchem Anschluss dieser sitzt. Danach füge ich die Bibliothek Servo.h ein, um das Steuern der Servomotoren zu ermöglichen. Dann gebe ich den 5 verbauten Motoren jeweils einen Namen.

```
#include <Servo.h>

Servo greifer;
Servo base;
Servo senkrecht;
Servo waagerecht;
Servo rampe;
```

In der Funktion void setup, welche nur einmal beim Verbinden des Roboters mit Spannung ausgeführt wird, deklariere ich welcher Pin am Arduino Nano mit welchem Motor verbunden ist.

```
void setup() {
  Serial.begin(9600);
  greifer.attach(6);
  senkrecht.attach(8);
  waagerecht.attach(13);
  base.attach(9);
  rampe.attach(11);
}
```

Zusatzqualifikation Programmierung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

In der Funktion void loop, welche immer wieder ausgeführt wird solange eine Spannungsversorgung vorhanden ist, fährt der Arm erst in Grundstellung. Die Motoren brauchen dafür einen Winkel, welchen ich im Programm angebe, den sie dann auch anfahren. Danach verfährt er Richtung Sockel und vor, um von da den Ball zu greifen.

```
//Greifen
base.write(180);
greifer.write(80);
delay(200);
for (rechts = 0; rechts <= 46; rechts += 1){
  waagerecht.write(rechts);
  delay(15);
}
delay(200);
for (links = 0; links <= 133; links += 1) {
  senkrecht.write(links);
  delay(15);
}
for (pos=80;pos>=30;pos-=1) {
  greifer.write(pos);
  delay(15);
}
```

Damit der Roboterarm nicht mit voller Geschwindigkeit vorfährt, den Ball wegstößt und um die Motoren zu schonen benutze ich zum Regeln des Tempos die for Schleife, welche es mir ermöglicht dem Motor einzelne Punkte und eine Pause zwischen den Bewegungen anzugeben.

Wenn er den Ball dann oben auf der Rampe losgelassen hat und ihn dann unten wieder aufheben soll, soll er im ersten Programm in eine Schleife fallen. Um diese Schleife zu realisieren benutze ich ein while-Statement.

```
//Schleife für Rampe
while(true){
  for (pos=120;pos<=146;pos+=1){
    base.write(pos);
    delay(15);
  }
  delay(500);
  for (pos=120;pos>=90;pos-=1){
    senkrecht.write(pos);
    delay(15);
  }
  for (pos=120;pos>=33;pos-=1){
    waagerecht.write(pos);
    delay(15);
  }
  for (pos=90;pos<=119;pos+=1){
    senkrecht.write(pos);
    delay(15);
  }
  for (pos=80;pos>=30;pos-=1) {
    greifer.write(pos);
    delay(15);
  }
  delay(200);
  for (rechts=33; rechts<=40; rechts += 1) {
    waagerecht.write(rechts);
    delay(15);
  }

  for (pos=119; pos>=90;pos-=1) {
    senkrecht.write(pos);
    delay(15);
  }
  for (rechts=40; rechts<=120; rechts += 1) {
    waagerecht.write(rechts);
```

Dadurch, dass die while-Schleife auf true gesetzt ist läuft diese im Programm ununterbrochen weiter.

Die Schranke am Ende der Rampe soll erst öffnen, wenn der Ball auch unten anliegt. Um das zu realisieren habe ich die Variablen „rechts“ und „links“ eingebaut, diese dienen dazu, um den Motor für die Schranke zu bewegen, wenn der Roboterarm auf einer bestimmten Position ist und die Zeit abgelaufen ist, welche der Ball wahrscheinlich braucht, um unten anzuliegen.

Zusatzqualifikation Programmierung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

Die Befehle für die Schranke werden erst dann ausgeführt, wenn „links“ größer gleich 115 und „rechts“ größer gleich 100 ist. „Links“ steht hier für die waagrechte Achse und „Rechts“ steht für die senkrechte Achse des Roboters

```
//rampe
if (links>=115 && rechts>=100){
  delay(1000);
  rampe.write(180);
  delay(1000);
  rampe.write(55);
}
```

```
//Schleife für Rampe
while(ablaeufer<4){
  ablaeufer++;
  for (pos=120;pos<=146;pos+=1){
    base.write(pos);
    delay(15);
  }
  delay(500);
  for (pos=120;pos>=90;pos-=1){
    senkrecht.write(pos);
    delay(15);
  }
  for (pos=120;pos>=30;pos-=1){
    waagerecht.write(pos);
    delay(15);
  }
  for (pos=90;pos<=118;pos+=1){
    senkrecht.write(pos);
    delay(15);
  }
  for (pos=80;pos>=30;pos-=1){
    greifer.write(pos);
    delay(15);
  }
}
```

Im zweiten Programm deklarieren ich im while Statement, dass die Schleife nur vier Mal ablaufen soll bevor der Ball auf dem anderen Sockel ablegt werden soll.

Dafür habe ich eine Variable angelegt, die „ablaeufer“ benannt und für jeden Schleifendurchlauf wird die Variable um eins hochgezählt und wenn diese den Wert vier annimmt ist das while Statement falsch und wird nicht mehr abgearbeitet.

Am Ende wird ein if-Statement aktiv, welches den Roboterarm den Ball greifen lässt und auf den linken Sockel platzieren lässt.

```
//Ablegen
if (ablaeufer<=4){
  for (pos=120;pos<=146;pos+=1){
    base.write(pos);
    delay(15);
  }
  delay(500);
  for (pos=120;pos>=90;pos-=1){
    senkrecht.write(pos);
    delay(15);
  }
  for (pos=120;pos>=30;pos-=1){
    waagerecht.write(pos);
    delay(15);
  }
  for (pos=90;pos<=118;pos+=1){
    senkrecht.write(pos);
    delay(15);
  }
  for (pos=80;pos>=30;pos-=1){
    greifer.write(pos);
    delay(15);
  }
}
```

Eigentlich war für beide Programme geplant eine Lichtschranke am Ende der Rampe zu verbauen und das Verfahren des Roboters von den Werten des Sensors abhängig zu machen. Dieser konnte jedoch nicht verbaut werden, weil er den Ball nur erkennt, wenn die Kugel direkt davor liegt, da aber die Position

Zusatzqualifikation Programmierung

Erstellung und Erprobung eines Roboterschulungsmodells mithilfe eines Arduino Nano

des Roboterarms zum Greifen des Objektes nah an der Rampe ist würde die Lichtschranke den Roboter behindern. Als Lösung habe ich das Programm Zeitabhängig geschrieben, statt Werte des Sensors auszulesen. Eine weitere Lösung wäre eine Lichtschranke zu besorgen, welche Objekte aus Kunststoff aus etwas weiterer Distanz erkennen kann oder den Ball aus einem anderen Material zu fertigen, welches der jetzige Sensor von einem bestimmten Abstand erkennen kann.

4. Fazit

Beide fertige Programme wurden einzeln auf den Arduino Nano geladen und getestet. Der Roboter hat alle Punkte richtig angefahren und den Ball auf seine Position gebracht.

5. Zeitplanung

Nr.	Tätigkeit	Soll-Std.	Ist-Std.
	Information und Auftragsplanung		
1	Arbeitsschritte planen, Zeitplanung erstellen	0,35 Std.	0,35 Std.
2	Herunterladen des Programmes und der Treiber	0,1 Std.	0,1 Std.
3	Werkzeug und Material auswählen und beschaffen	0,25 Std.	0,25 Std.
	Auftragsdurchführung		
4	Zusammenbauen des Roboterarms	2 Std.	2,5 Std.
5	Schreiben der Programme	4 Std.	4 Std.
	Auftragskontrolle		
6	Kontrolle der Positionen des Roboterarms	0,5 Std.	0,5 Std.
	Geplante Zeit	7,7 Std.	
	Benötigte Zeit	7,2 Std.	
	Abweichung	+0,5 Std.	
	Grund der Abweichung	Ausbessern der Halterungen für die Motoren des 3D-Drucks, weil diese zu klein gedruckt wurden.	