2DOF Light Scan Robot

Beschreibung:

In der letzten Lehreinheit haben wir einen 1 „Degree of Freedom“ (DOF) Light Scan Roboter gebaut. In dieser Einheit soll dieser um einen weiteren Servo bzw. Freiheitsgrad erweitert werden.

Der Roboter besteht aus zwei Servos und einem Lichtsensor.

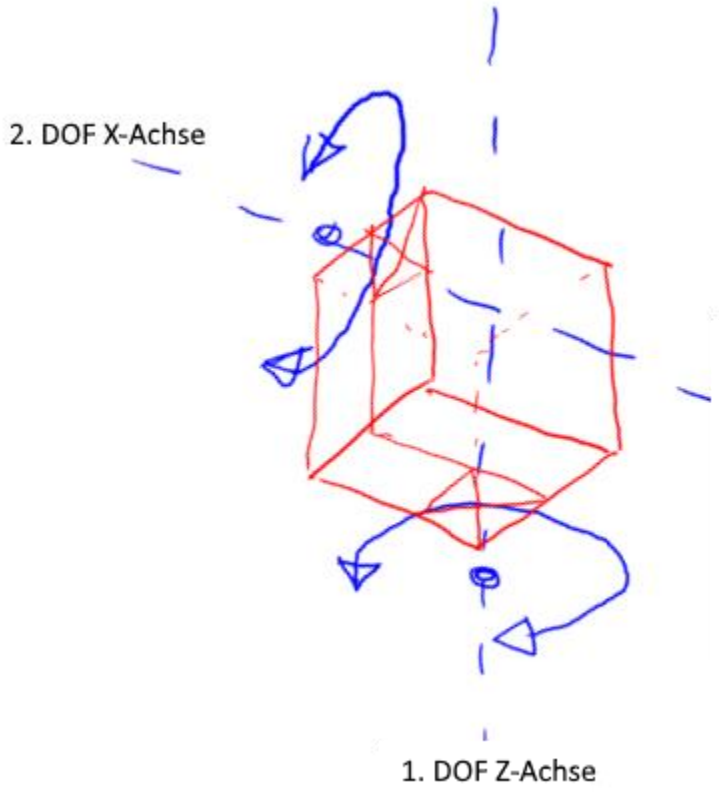


Abbildung 1 Anordnung der rotatorischen Gelenke

In der ersten Übung werden wir die beiden Servos direkt mit einem analogen Joystick steuern.

In der zweiten Übung soll der Roboter die Lichtsensorwerte selbstständig seine Freiheitsbereich/Arbeitsbereich (ab)scannen und nach dem Scannen zur hellsten Position (zurück) fahren und dort für ein paar Sekunden verweilen.

Bauanleitung:

Benötigt werden 2 Servos, 1 Lichtsensor, etwas Karton, 6x Jumperkabel (Male to Female), einen Arduino für die Steuerung und eine separate Stromversorgung für die Servos.

Die benötigten Werkzeuge sind eine Bastelschere, ein Stift, eine Heißkleberpistole mit etwas Heißkleber.

Alternativ kann die Konstruktion statt Heißkleber mit Kabelbindern, Klebeband oder Nadel und Faden fixiert werden. Dies hat den Vorteil dass man alle Teile einfacher wiederverwerten kann.

Abbildung 2 Benötigtes Material und Werkzeug

Im ersten Schritt zeichnen wir dir nötigen Teile (eine Standplatte und die Sensor-Halterung) auf Karton und schneiden diese dann aus.

Schritt 1:

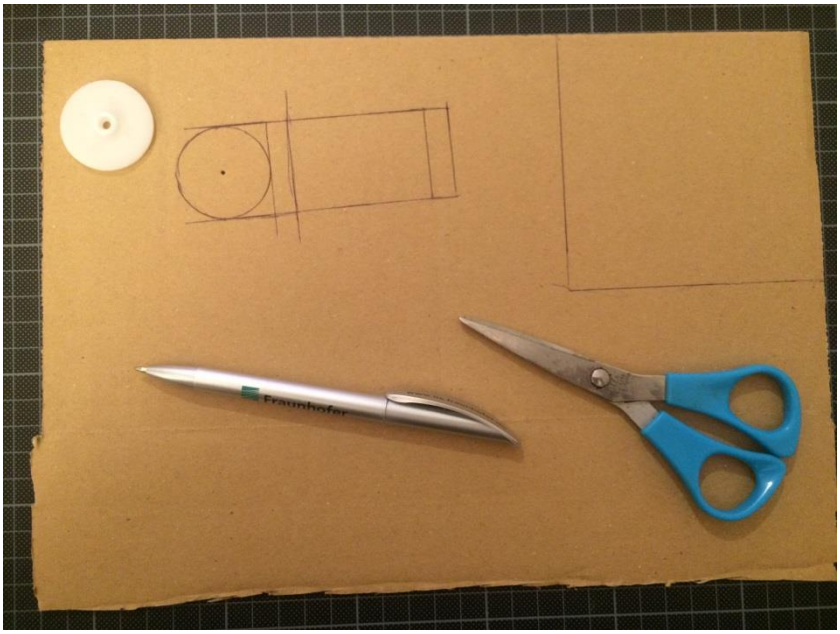


Abbildung 3 Aufzeichnen

Nehmen sie das Servo-Horn und den Servo zur Hilfe um die Größe für die Sensor-Halterung und Standplatte zu Dimensionieren. Bei der Form und Größe hat man etwas kreativen Spielraum, einzig wichtig ist das die Servos ihre Rotation möglichst voll ausnützen können ohne das die Konstruktion wo ansteht.

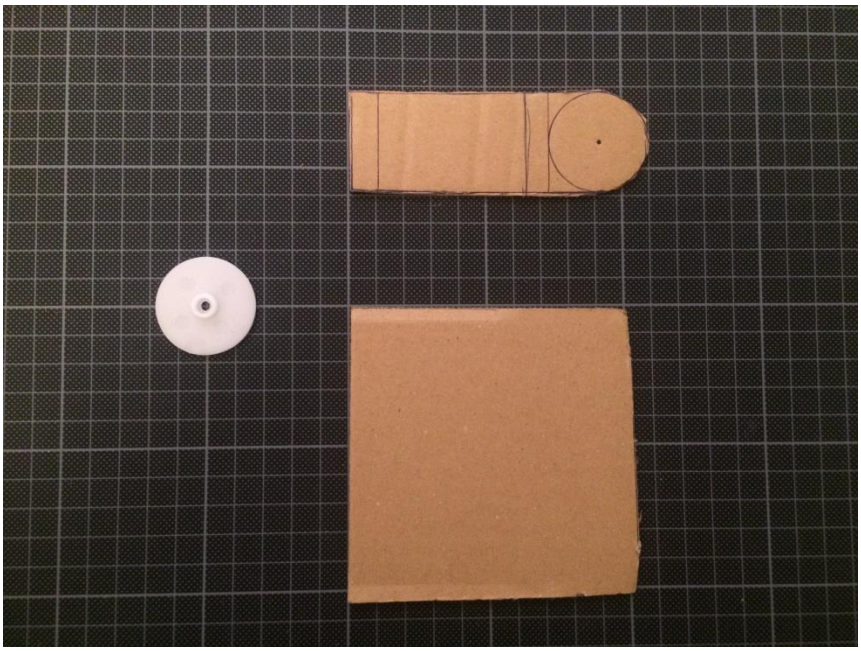
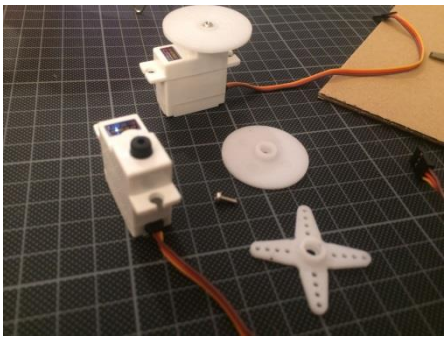


Abbildung 4 Ausschneiden

Die Kartonteile sollten ungefähr so wie in Abbildung 4 aussehen.

Schritt 2:

Im zweiten Schritt verbinden wir die zwei Servos zu einer Einheit so dass eine Servo-Drehachse zur anderen Drehachse um 90 Grad versetzt ist.



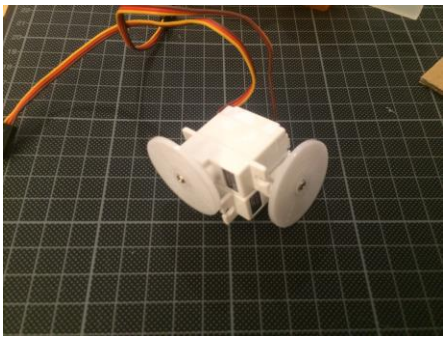
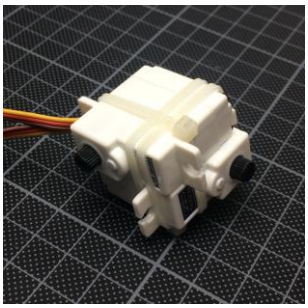
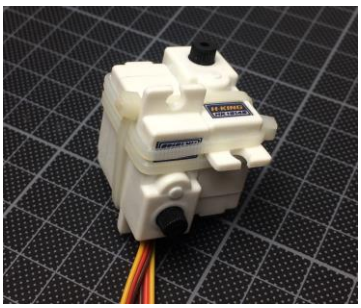


Abbildung 5 - 90 Grad versetztes Ausrichten der Servos

Achten sie darauf dass sich die Servo-Hörner frei drehen können und nicht Kontakt zum Servogehäuse haben.





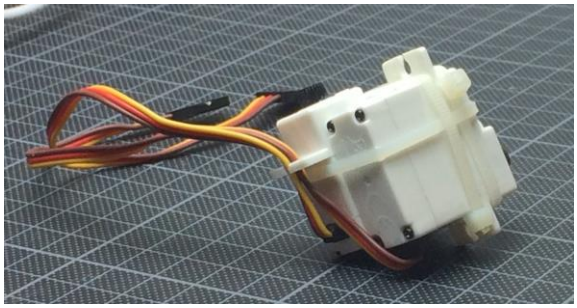


Abbildung 6 Befestigung mit 3 Kabelbinder

Sie können die Servos mit etwas Heißkleber, Klebeband oder ein paar Kalbelbindern aneinander fixieren.

Schritt 3:

Nun verbinden sie ein Servohorn mit der Sensor-Halterung und ein weiteres Servohorn mit der Bodenplatte.

Machen sie vor dem Fixieren der Servohörner ein Loch an der Stelle der Servo-Drehachsen in den Karton.

Mit etwas Heißkleber, Klebeband oder mit Faden und Nadel lassen sich die Servohörner einfach am Karton fixieren.

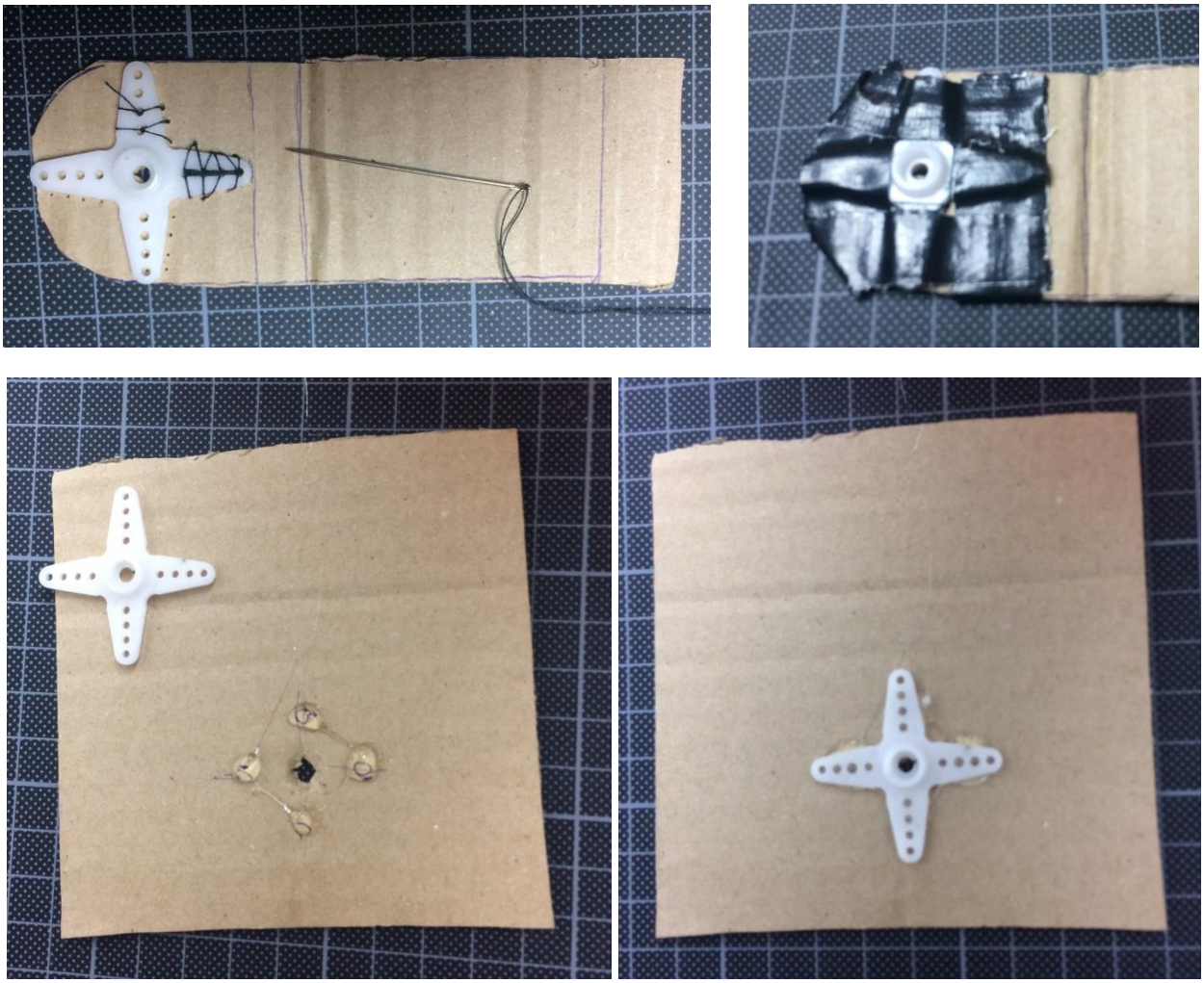


Abbildung 7 Befestigung von Servohörner mit Sensor-Halterung und Bodenplatte

Schritt 4:

Schneiden sie ein kleines rechtwinkliges Stück aus dem übrig gebliebene Karton. Dies wird zur Verstärkung der Sensorhalterung dienen.

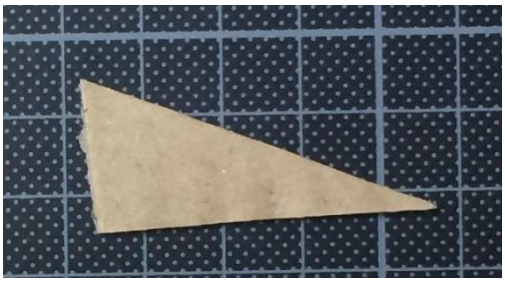


Abbildung 8 Verstärkungs-Winkel

Biegen/Knicken sie nun die Sensorhalterung zu einem Winkel mit 90 Grad und befestigen sie den Verstärkungswinkel mit etwas Heißkleber in der Ecke der Sensorhalterung.

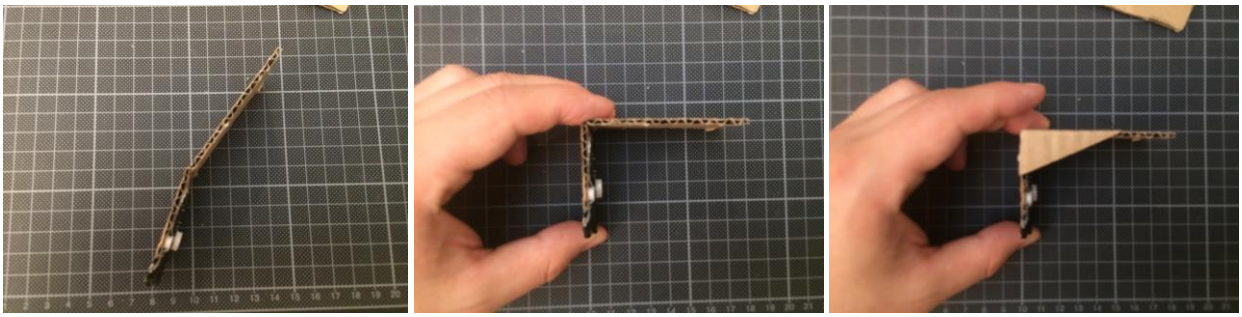


Abbildung 9 Sensorhalterung

Bei Bedarf können sie die Innen-Kante der 90Grad-Ecke mit Heißkleber verstärken.

Damit sollten alle nötigen Kartonteile gefertigt sein.

Schritt 5:

Nun können sie die Servohörner der Bodenplatte und der Sensorhalterung an die Servoachsen stecken.

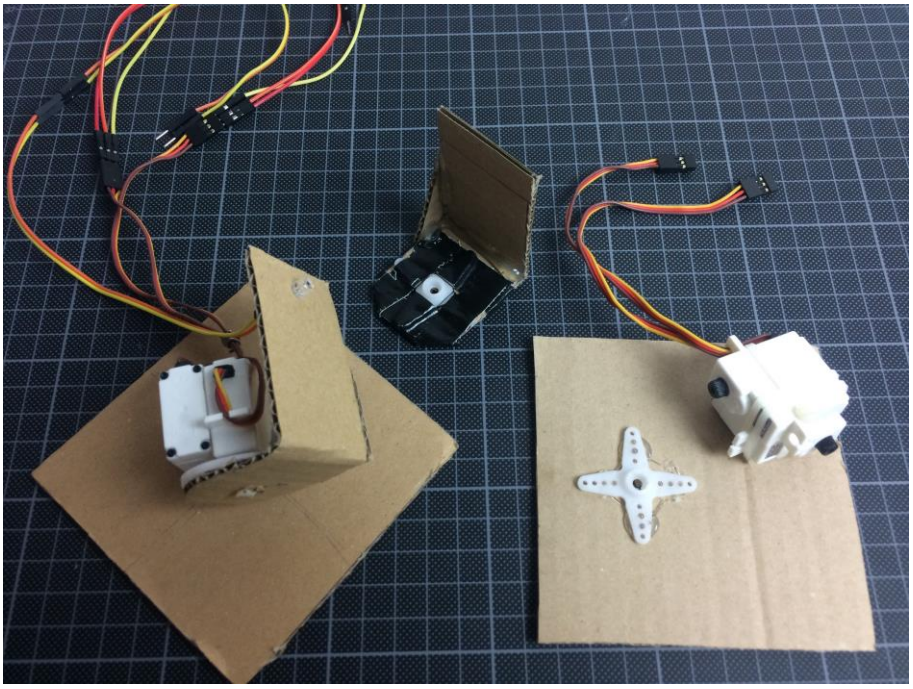


Abbildung 10 Sensorhalterung und Bodenplatte

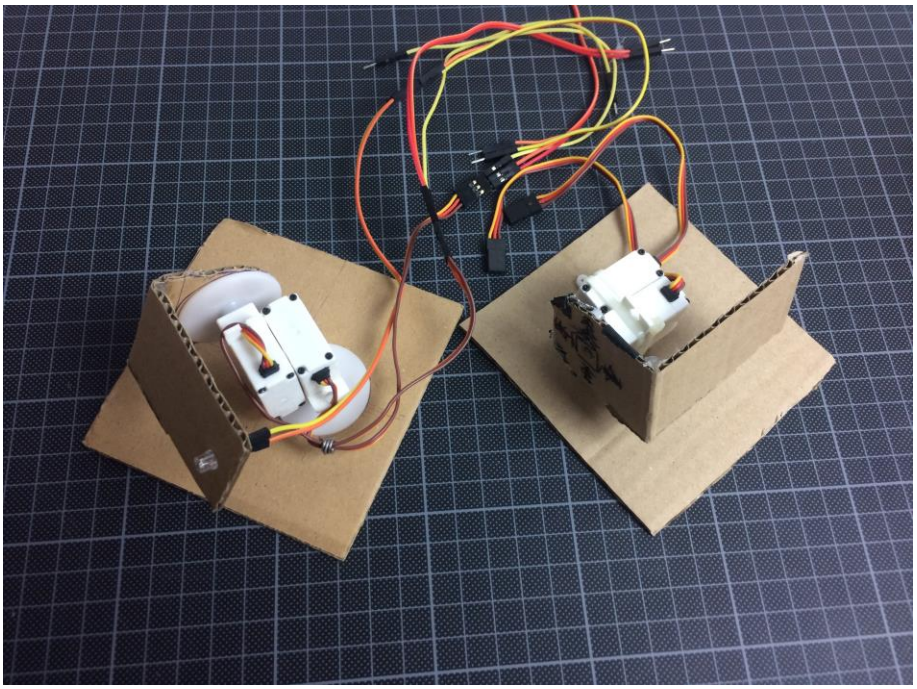


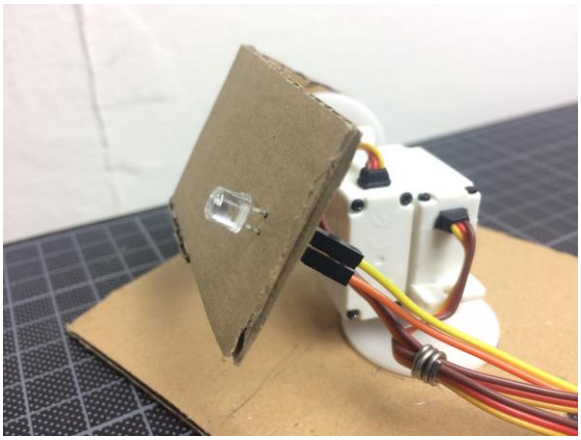
Abbildung 11 Fertig Assembliert

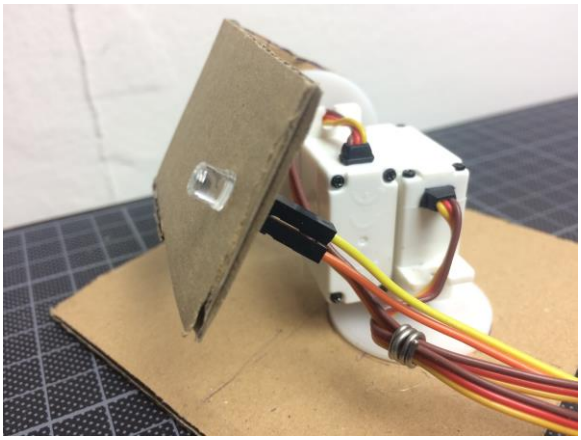
Positionieren sie die Sensorhalterung so dass der linke Anschlag und der rechte Anschlag des Servos nicht an der Bodenplatte anstehen.

Schritt 6:

Zum Abschluss bringen sie den Lichtsensor am Rand der Sensorhalterung an.

Stecken sie hierfür die (Draht-) Beine des Sensors einfach durch den Karton und stecken sie zwei Female-to-Male Jumperkabel an den Sensor an.





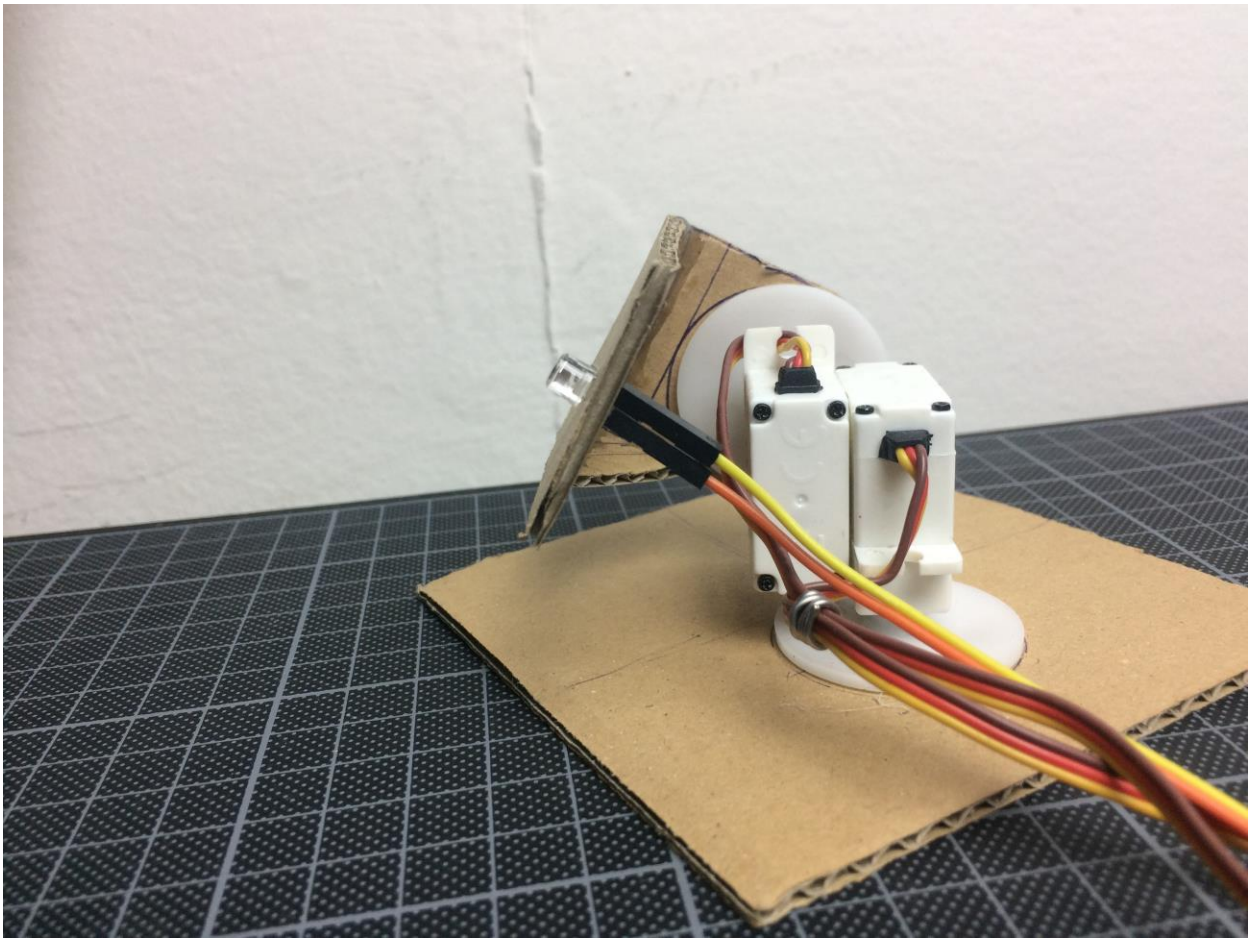


Abbildung 12 - 2-DOF Licht Scann Roboter

Verkabelung:

Verbinden sie Servos, Lichtsensor und optional einen analogen Joystick mit dem Arduino und der Stromversorgung. Benutzen sie dazu Male-to-Male und Female-to-Male Jumper-Kabel.

Komponente	**Verbindung**
1. Servo (vertikale)	Servo Power- (Braun/Schwarz) -> Ground am Steckbrett Servo Power+(Orange/Rot) -> +5Volt am Steckbrett Servo Signal (Gelb/Weiss) -> Arduino Pin 5
2. Servo (horizontal)	Servo Power- (Braun/Schwarz) -> Ground am Steckbrett Servo Power+(Orange/Rot) -> +5Volt am Steckbrett Servo Signal (Gelb/Weiss) -> Arduino Pin 6
Arduino Ground	Arduino Ground Pin -> Ground am Steckbrett
Lichtsensor/Photozelle	Lichtsensor Cathode (kurzes Bein) -> Arduino Ground Lichtsensor Anode (langes Bein) -> Arduino analog Pin A0
Optionaler Joystick	Joystick-GND -> Arduino Ground Joystick-+5V -> Arduino +5Volt Joystick-VRx -> Arduino analog Pin A1 Joystick-VRy -> Arduino analog Pin A2
Stromversorgung (5Volt)	Stromversorgung minus (Leitung) -> Ground am Steckbrett Stromversorgung plus (Leitung) -> +5V am Steckbrett

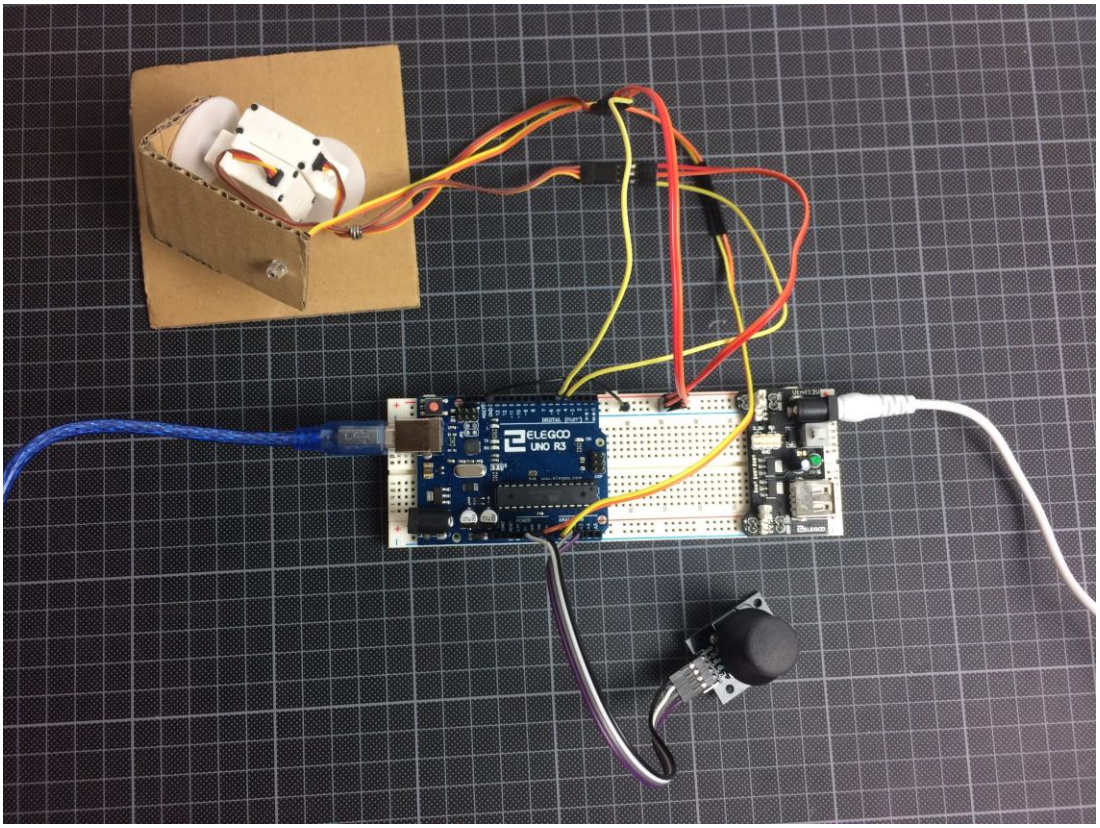


Abbildung 13 Steckbrett-Verkabelung von Arduino mit 2x Servos, Lichtsensor, einem Analoger-Joystick und einer Stromversorgung

Software:

Um dem Roboter Leben einzuhauchen müssen wir nur noch die Software schreiben.

In der ersten Übung steuern wir die Rotation/Position der beiden Servos direkt durch den analog Werte eines Joystick.

In der zweiten Übung soll sich der Roboter selbstständig bewegen und nach dem hellsten Lichtpunkt scannen.

Übung 1: Joystick-(direkt)-steuerung

Schreiben sie ein Programm das direkt die analogen Joystick-Werte (0-1023) für die Servo-Positionssteuerung (0-180) verwendet. Joystick-Wert X für den ersten Servo und Joystick-Wert Y für den zweiten Servo.

Plotten sie dabei ihren Joystickwert und den Lichtsensor Wert mittels Serial Plotter.

Eine hilfreiche Funktion um die gelesenen analogen Werte auf eine Servo Positionen umzurechnen ist die Arduino-Funktion map(...).

zB: Ihr Joystick liefert Werte von minimal 0 bis maximal 1023 dann können sie die Werte mit folgendem Statement umrechnen.

```
// Deklariere neue int Variable servoMappedPosX und weise ihr den umgerechneten Wert von
// Variable joystick_val_x zu. Umrechnung des Wertebereichs 0 bis 1023 auf 0 bis 180.

int servoMappedPosX = map( joystick_val_x , 0 , 1023 , 0 , 180 );// Map to  0 – 180
```

Ihre setup Funktion könnte so aussehen:

```
// Inkludieren der Servo.h library fuer die Servo-Steuerung
#include<Servo.h>

// Deklarieren der Servos als globale Objekte/Variablen mit Namen myServoX und myServoY
Servo myServoX;
Servo myServoY;

// Arduino Setup Funktion (wird einmal bei Programm-Start aufgerufen)
void setup() {

	// Setzen der Servo-Pins
	myServoX.attach(5); // Attach servo PWM signal to pint 5
	myServoY.attach(6); // Attach servo PWM signal to pint 6

	// Setzen der analogen Arduino Pins A0, A1 und A3 als Eingang/INPUT
	pinMode(A0, INPUT); // Light Sensor Pin
	pinMode(A1, INPUT); // Joystick X Pin
	pinMode(A2, INPUT); // Joystick Y Pin

	// Oeffnen einer Serielen Schnittstelle
	Serial.begin(115200); //Serial for SerialPlotter with 115200 bits per seconds
}
```

Ihre loop Funktion könnte so aussehen:

Vervollständigen sie den fehlenden Programm-Code [*] und laden sie das Programm auf ihren Arduino.

Plotten sie dabei den Lichtsensorwert du die Joystickwerte mit Arduino's SerialPlotter.

```
// Arduino loop Funktion
void loop() {

// 1.)
// Einlesen der analogen Spannungs-Werte des Lichtsensors und der Joystick-Achsen x und y
       int sensor_val     = analogRead(A0); // Neue Variable mit Lichtsensorwert
       int joystick_X_val = analogRead(A1); // Neue Variable mit Joystickwert x
       int joystick_Y_val = analogRead(A2); // Neue Variable mit Joystickwert y

// 2.)
// Daten Ouput fuer SerialPotter
       Serial.print(sensor_val); // Lichtsensor
       Serial.print(" "); // Seperator
       Serial.print(joystick_X_val);// Joystick x
       Serial.print(" ");// Seperator
       Serial.println(joystick_Y_val);// Joystick y (mit println fuer das Zeilenende)

// 3.)
/* [*] Deklarieren sie zwei Integer (int) Variablen mit Namen servoMappedPos_X und
servoMappedPos_Y und weisen ihnen die umgerechneten Werte der Variablen joystick_X_val
und joystick_Y_val zu. Der Wertebereich von 0 bis 1023 soll dabei auf den Bereich 0 bis
180 umgerechnet werden. */

/* Code hier */

// 4.)
// Positionieren der Servo
       myServoX.write(servoMappedPos_X);
       myServoY.write(servoMappedPos_Y);

// 5.)
// Ein kurze Pause von 1ms jeden Loop
       delay(1);
}
```

Übung 2: 2-DOF Lichtwert scannen

Schreiben sie eine Funktion die eine Servo in 1 Grad-Schritten von 0 – 180 fahren lässt. Am Ende jeder Fahrt von 0 - 180 oder 180 - 0 Grad soll der zweite Servo (ausgehend von der Position 0) seine Position um 5 Grad erhöhen um anschließend mit den ersten Servo in 1 Grad-Schritten wieder in die andere Richtung zu fahren.

Lesen sie nach jedem 1Grad Schritt des ersten Servos einen Lichtsensor wert und merken sie sich die Positionen der beiden Servos und den Lichtwert falls dieser ein (neuer) maximaler Wert ist. Die Funktion sollte die Position mit dem höchsten Lichtwert in globalen Variablen speichern.

Wenn der volle Bereich der beiden Servos gescannt wurde ist die Funktion fertig und kann ‚returnen'.

Fahren sie nach dem Scannen an die Position des höchsten Lichtwertes und verweilen sie ein paar Sekunden an dieser Position bevor sie den nächsten vollen Scan durchführen.

Plotten sie dabei den Lichtsensorwert du die Servo-Positionen mit Arduino's SerialPlotter.

Ihre Funktion könnte so aussehen:

Vervollständigen sie den fehlenden Programm-Code `[*]` und laden sie das Programm auf ihren Arduino.

```
// Globale Variablen zum Speichern der Servo Positionen
int posX_Max = 90;
int posY_Max = 90;

// Funktion die vollständig Scannt
int Scan_FullXY()
{
	//1.) Deklarieren der (temporären) Variablen
	int tmp_ValueMax = -1000; // Initialer maximaler Lichtwert (-1000)
	int tmp_PosXMax = 0;      // Temporäre Variable der Servo Position @max Light
	int tmp_PosYMax = 0;      // Temporäre Variable der Servo Position @max Light

	//2.) Ausgangs-Positionieren beider Servos
	myServoX.write(10);
	myServoY.write(10);
	delay(500); // Dem positionieren etwas Zeit geben (500ms)

	// Hilfs-Variablen fuer die innere for Schleife
	int tmp_StartY = 0;
	int tmp_EndY = 180;
	int tmp_Update = 1;

	//3.) Scanne vollen Bereich der vertikalen achse (in 5 Grad Schritten)
	for (int x = 0; x <= 180; x += 5)
	{
		// Neue Servo Position aktualisieren
		myServoX.write(x);
```

```
                // 4.) Scanne vollen Bereich der horizontalen achse (in 1 Grad Schritten)
                for (int y = tmp_StartY; y <= tmp_EndY; y += tmp_Update)
                {
                        // Neue Servo Position aktualisieren
                        myServoY.write(y);
                        delay(3); // Etwas warten

/* [*] Deklarieren sie ein Integer (int) Variablen mit Namen lightValue und weisen ihr
den gemessenen Lichtmesswert zu. */

                        /* Code hier */

                        int lightValue = ReadSensorOversampling(A0, 32);

/* [*] Geben sie die Servo Position und Lichtmess wird fuer Serial Plot aus */

                        /* Code hier */

                        // 6.) Wenn neuer Hoechst-Wert dann Position speichern
                        if (tmp_ValueMax < lightValue)
                        {
                                tmp_PosXMax = x;
                                tmp_PosYMax = y;
                                tmp_ValueMax = lightValue;
                        }
                }

                // 7.) Ändern der Fahr-Richtung nach jeder (inneren) Fahrt.
                if (tmp_Update == 1)
                {
                        tmp_StartY = 180;
                        tmp_EndY = 0;
                        tmp_Update = -1;
                }
                else
                {
                        tmp_StartY = 0;
                        tmp_EndY = 180;
                        tmp_Update = 1;
                }
        }
        // Speichern der Positionen mit dem maximalen Lichtwert
        posX_Max = tmp_PosXMax;
        posY_Max = tmp_PosYMax;

        return tmp_ValueMax;
}

// Neue Loop Funktion
void loop() {
        // Voller Scann
        Scan_FullXY();
        // Fahre an die maximale Licht-Position
        myServoX.write(posX_Max);
        myServoY.write(posY_Max);
        // Und 6 Sekunden warten
        delay(6000);
}
```

Optionale Übung 3: Verbessern sie das Sensorsignal

Schreiben sie eine Mess-Funktion um den Lichtsensor-Wert zu Verbessern.

Oft hat man es mit nicht idealen Sensorwerten. Rauschen oder zu geringe Auflösung erschweren das Arbeiten mit den Werten. Um dies zu verbessern gibt es viele Möglichkeiten. (Filtern, Mehrfach-Samplen, Offset rausrechnen, anpassen der analogen Messreferenz etc...)

Wichtig dafür ist es sich die Charakteristik der Werte und des Sensor anzusehen um nicht am Ziel vorbei Zuschießen oder es gar zu verschlechtern.

Pseudo-Code:

Filtern:

Die Signal Vergangenheit mit einbeziehen um keine zu schnellen Sprünge zu machen.

```
valueLast = (valueLast * 2 + valNow) / 3;
```

Mehrfachsampling:

Die Summe aus mehrfachen Sensor Werten bilden und danach dividieren.

```
for (int i = 0; i < oversampling; i++)
{
	int tmp = analogRead(pin);

	value += tmp;

}

value = value  / oversampling; // Der Summen Wert muss nicht durch die volle Anzahl
                               // Samples dividiert werden. Zb: (oversampling – 2)
```

Minimale und Maximale Samples verwerfen:

Sie können auch den kleinsten und größten gelesen Wert nicht in ihre oversampling Summe einbeziehen

```
int minVal = 10000;
int maxVal = -10000;
// Collect all values
for (int i = 0; i < oversampling; i++)
{
	int tmp = analogRead(pin);
	value += tmp;
	// store the min and max value
	if (tmp < minVal) minVal = tmp;
	if (tmp > maxVal) maxVal = tmp;
}

value -= minVal + manVal;

value = value  / (oversampling - 2); // Der Summen Wert muss nicht durch die volle Anzahl
                                     // Samples dividiert werden. Zb: (oversampling – 4)
```