

Erfahrungsbericht zum Aufbau eines rotierenden inversen Pendels

Einleitung

Das inverse Pendel ist ein beliebtes Studienobjekt in den Ingenieurwissenschaften Regelungstechnik, Elektrotechnik und Maschinenbau. Dabei soll ein frei schwingendes Pendel senkrecht nach oben balanciert werden, ähnlich dem Balancieren eines Bleistiftes auf dem Finger. Das Pendel ist an einem beweglichen Wagen befestigt, der die nötigen Ausgleichbewegungen durchführt.

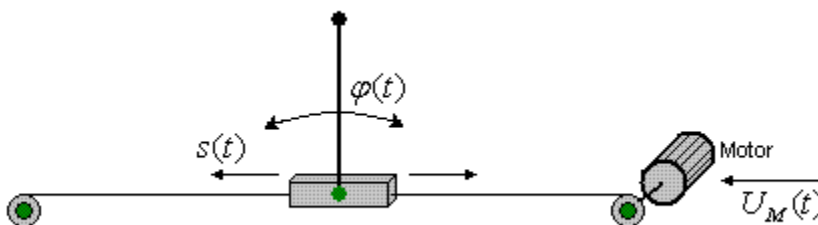


Bild 1: Grundprinzip eines inversen Pendels

Das inverse Pendel ist von seiner theoretischen Modellierung und seinen Anforderungen an eine Stabilisierung her gesehen durchaus komplex, gleichzeitig aber von seinem einfachen Aufbau und seiner Anschaulichkeit her kaum zu übertreffen. Diese Eigenschaften machen das inverse Pendel weltweit zu beliebten Lehr- und Praktikumsobjekten im Fachbereich der Regelungstechnik. Die Problemstellung beim inversen Pendel ist dementsprechend bereits gut erforscht, aber dennoch nach wie vor spannend, da hier verschiedene Forschungsbereiche wie Robotik, Kontrolltheorie und rechnergestützte Regelung kombiniert werden. Da es sich bei diesem System um ein nichtlineares, instabiles System handelt, eignet es sich hervorragend als Experimentierplattform für das Überprüfen bestehender und dem Entwickeln neuer Regelungsalgorithmen.

Die Realisierung eines geeigneten Reglers kann mittels verschiedener Entwurfsstrategien erfolgen, hier seien [PID-Regler](#), Regler mit Polvorgabe, LQR-Regler, Fuzzy Systeme oder Neuronale Netze genannt. Oft wird auch das Aufschwingen des Pendels aus der Nulllage durch Bewegungen des Wagens realisiert. Dafür sind nichtlineare Herangehensweisen wie Energieansätze erforderlich.

Aufbauvarianten des Inversen Pendels

- a) Das lineare inverse Pendel (Inverted-Pendulum): das Pendel ist dabei auf einem Schlitten montiert und der Schlitten läuft auf einer Linearführung, sowie in den Bildern 1, 2 und 3 gezeigt. Steigerungsformen sind dann das Zweifach- und Dreifachpendel, bei denen der Pendelarm nochmals unterteilt ist



Bild 3: inverses Pendel mit Linearführung

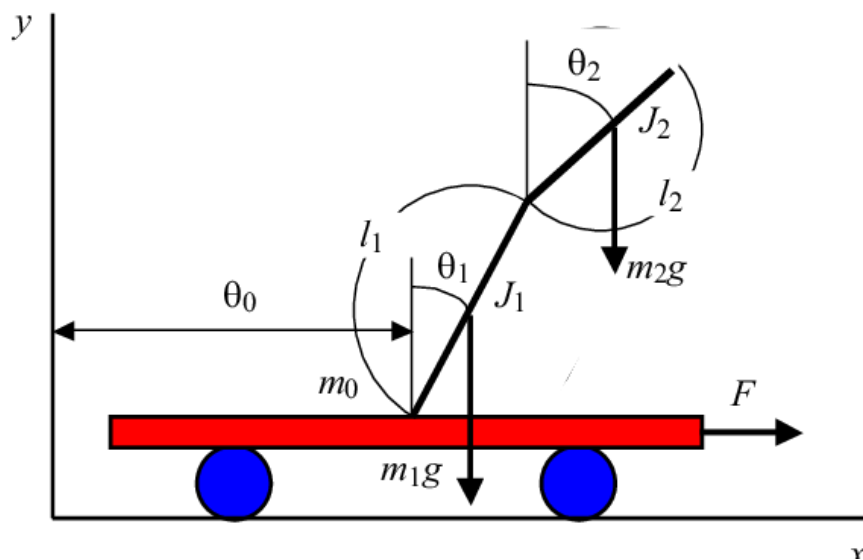


Bild 4: inverses Doppelpendel

- b) Das rotierende inverse Pendel (Rotary-Inverted-Pendulum): das Pendel ist am Ende eines rotierenden Arms befestigt. Der Arm wird von einem Motor angetrieben der die notwendige Ausgleichbewegung erzeugt. Dieser Aufbau ist sehr beliebt und mechanisch etwas einfacher zu realisieren als das lineare Pendel.

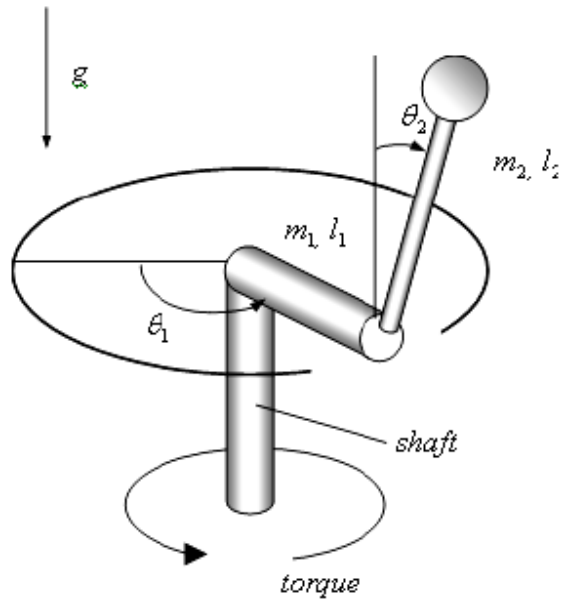


Bild 5: Grundprinzip des rotierenden, inversen Pendels



Bild 6: Aufbau eines rotierenden inversen Pendels

- c) Das räumliche, inverse Pendel (spherical inverted pendulum), das Pendel kann in alle Richtungen ausschlagen, der mechanische Aufbau und die Regelung sind besonders anspruchsvoll.



Bild 7: Aufbau eines sphärischen inversen Pendels

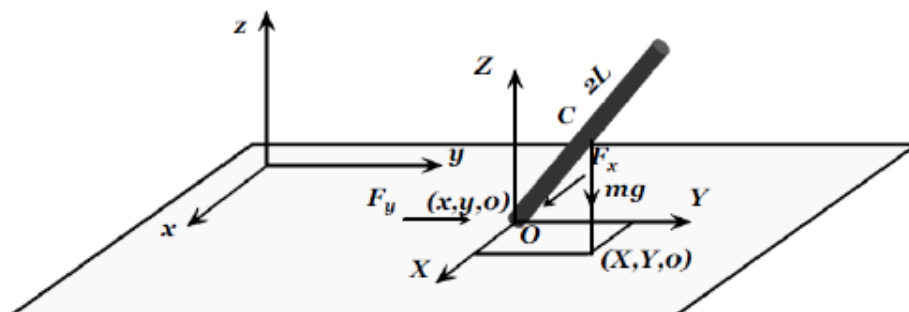


Bild 8: Prinzip eines sphärischen inversen Pendels

Quellen im Internet

Im Internet finden sich zahlreiche Beiträge und Diplomarbeiten die sich mit dem Aufbau und der Analyse des inversen Pendels beschäftigen. Auf YouTube findet man unter dem Stichwort „inverses Pendel“, „inverted pendulum“ oder „rotary inverted pendulum“ diverse Videoclips. Dabei werden neben dem Einfachpendel auch Doppel- oder Dreifachpendel vorgeführt. Einige der Arbeiten beschränken sich auf die Simulation in MATLAB. Die Arbeiten in denen der praktische Aufbau realisiert wurde, zeigen deutlich wie schwierig es ist Theorie und Praxis in Einklang zu bringen. Fazit all dieser Arbeiten ist:

- große Sorgfalt ist auf einen stabilen mechanischen Aufbau zu legen
- Auswahl geeigneter Sensoren zur Erfassung des Pendelwinkels und der Schlittenposition
- Gedanken über die Echtzeitfähigkeit des Mikrorechners für die Prozessdatenverarbeitung

Die folgende Liste zeigt eine kleine Auswahl. Viele der Arbeiten sind in englischer Sprache verfasst.

- https://de.wikipedia.org/wiki/Inverses_Pendel
- [UNI Ulm](#)
- [Diplomarbeit Tobias Migge](#)
- [Diplomarbeit TU Dresden](#)
- [Facharbeit Inverses Pendel](#)
- [Forschungsbericht Doppelpendel](#)
- [Diplomarbeit TU Chemnitz](#)
- [Robotikpraktikum](#)
- [Aufbauanleitung eines selbstbalancierenden Roboters](#)

YouTube Links

[Aufbaubericht eines inversen Pendels](#) bester Praxisbericht einschließlich kompletter Arduino Code!

[Doppel Pendel TU Wien](#)

[Dreifach Pendel TU Wien](#)

[Inverses Pendel Hochschule Merseburg](#)

[Inverses Pendel Technikum Wien](#)

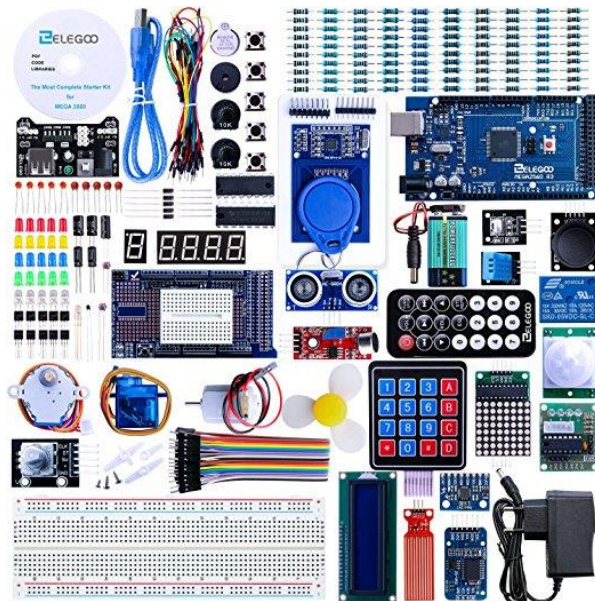
[The Cubli Balance](#)

Zielsetzung

Es soll unter den Bedingungen eines „Hobby Makers“ ein rotierendes inverses Pendel aufgebaut werden. Die Kosten der Einzelteile liegen in Summe bei 130€. Für die Montage werden keine Spezialwerkzeuge oder Maschinen wie CNC-Fräse, 3D Drucker oder Drehbank benötigt. An Werkzeugen sollte vorhanden sein: Bohrmaschine, Gewindebohrer, Feile und ein Satz Schraubendreher bzw. Bits. Die Materialien sind sämtlich über das Internet (Ebay, Amazon, usw.) bestellbar bzw. auf Baumärkten erhältlich. Als elektronische Steuerung wird ein [Arduino Megaboard](#) benutzt.



Ein paar Grundkenntnisse in der Programmierung des [Arduino](#) werden vorausgesetzt. Als Einstieg in die Welt des Arduino Mikrokontrollers sei der Baukasten von [Elegoo](#) empfohlen, der gleich das Arduino Mega Board enthält.



An Hand zahlreicher Beispiele wird die Programmierung Schritt für Schritt erläutert. Man erhält einen guten Einblick in die Welt der Sensoren (Temperatur, Entfernung, Beschleunigung usw.). Weiterhin findet man für den Arduino zahlreiche Beiträge und Programme bzw.

Programmibliotheken im Internet sowie diverse Bücher. Der Autor hat mit dem [Arduino Kochbuch](#) sehr gute Erfahrungen gemacht.

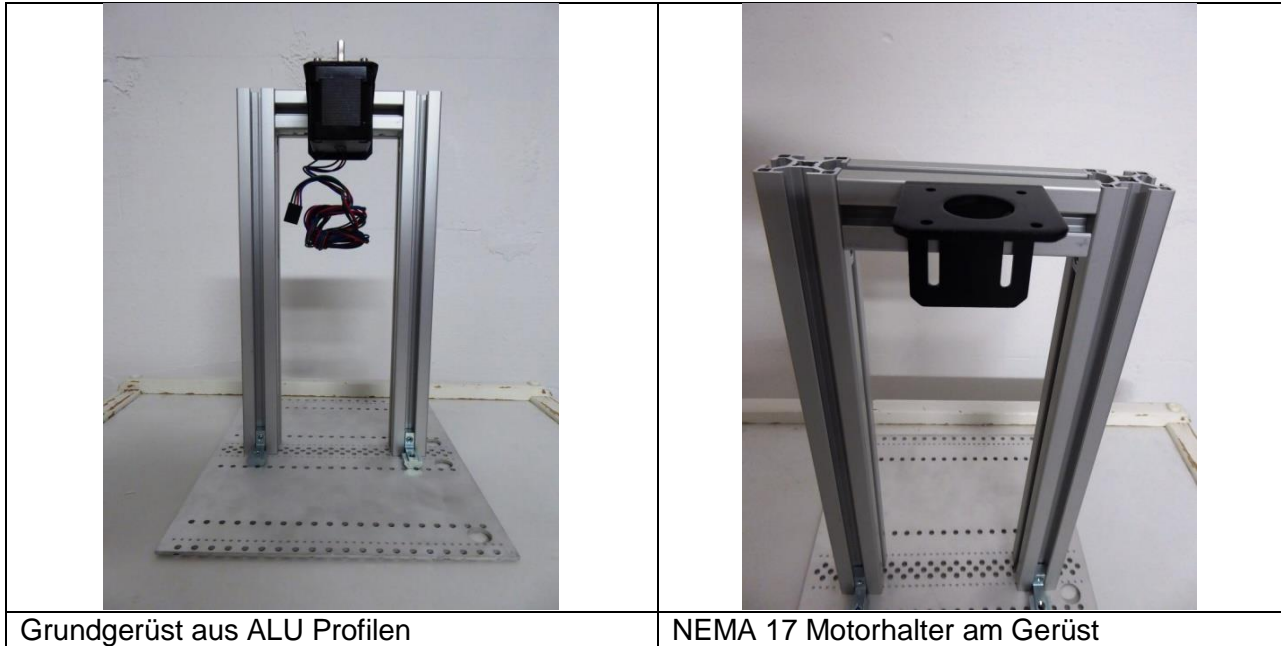
Das mechanische Grundgerüst



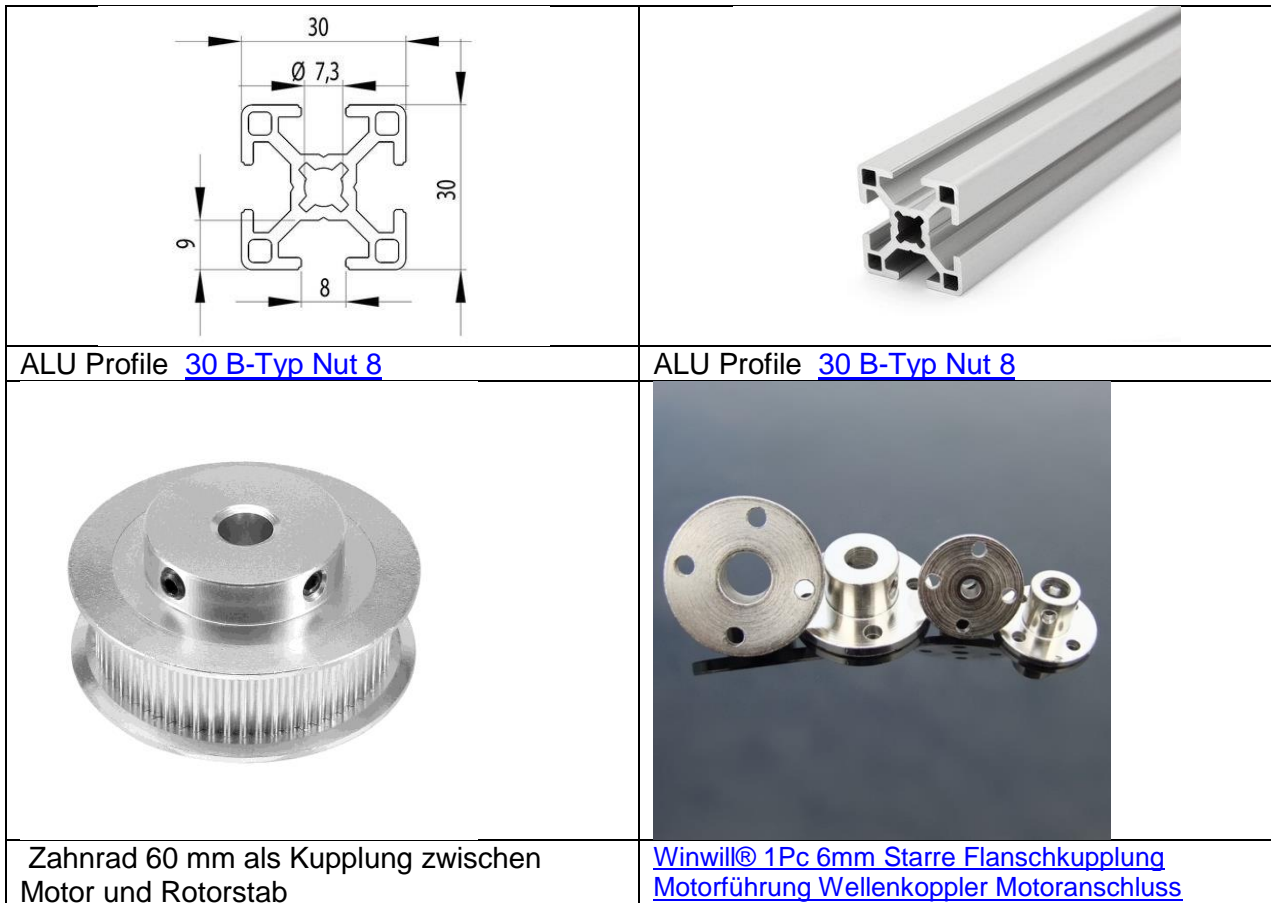
Labora Aufbau des rotierenden inversen Pendels

Für das mechanische Grundgerüst eignen sich ALU Profile [30 B-Typ Nut 8](#) der Firma Dold Mechatronik. Dazu werden zwei Profilstangen der Länge 300mm senkrecht auf einer Grundplatte 300x200x2.5mm aus Aluminium oder Schichtpressstoff montiert. Die beiden senkrecht stehenden Profile werden oben mit einem 200 mm Profil quer verbunden (siehe Skizze). An diesem Querverbinder wird ein NEMA 17 Motorwinkel befestigt der den DC-Motor oder Schrittmotor aufnimmt

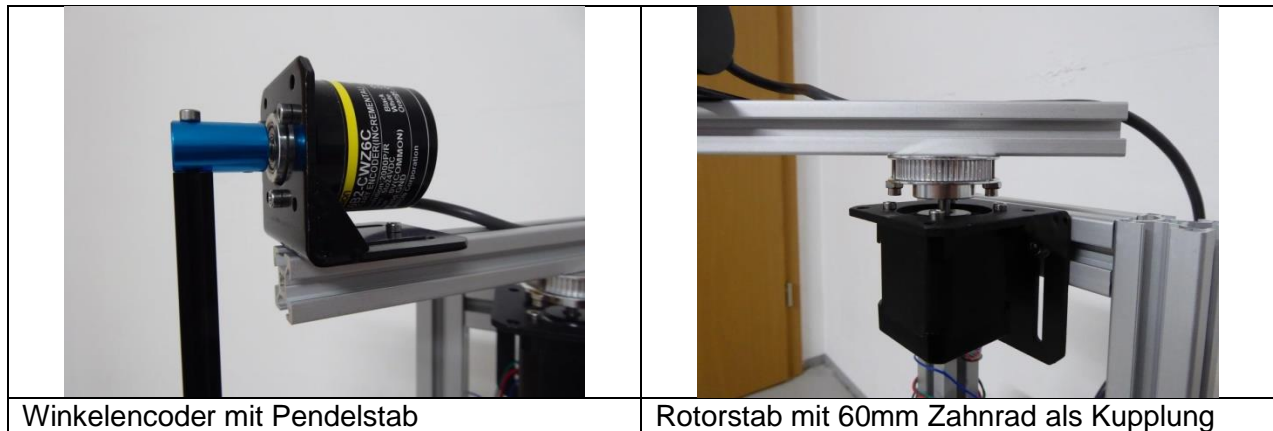
Aufbau eines rotierenden inversen Pendels



Auf der Welle des Motors wird eine Flanschkupplung oder ein 60mm Zahnrad montiert. An der Flanschkupplung bzw. dem Zahnrad wird der Rotorstab befestigt.



Als Rotorstab eignet sich ein 200mm AluProfil vom Typ 20 B-Typ Nut 6. Am Ende des Rotors wird ein Nema17 Winkel befestigt der den Winkelencoder aufnimmt. Am Winkelencoder ist der Pendelstab befestigt, ein Maker Beam Profil 10x10x300mm.



Der Winkelgeber (Encoder)

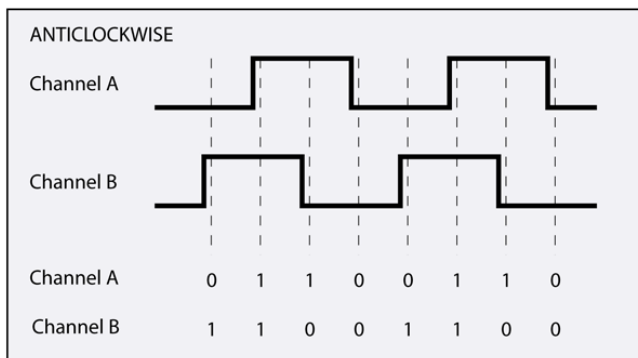
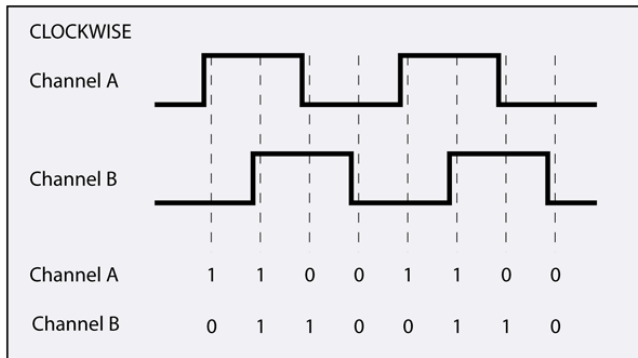
Um das Pendel in der aufrechten Position zu halten, muss fortlaufend der Winkel des Pendels gemessen werden (genauer gesagt die Abweichung zur Senkrechten). Für unsere Zwecke sind optische Encoder mit einer Auflösung von 1200 ... 2000 Impulsen je Umdrehung zweckmäßig. Die untere Grenze ist eine Auflösung von 600 Impulsen je Umdrehung. Bei Ebay findet man preiswerte Winkel Encoder Modelle im Bereich € 13 bis € 30 (z.B. Firma Omron aus chinesischer Fertigung). Der Encoder wird mit einem [NEMA 17 Motorwinkel](#) am Ende des Rotorstabs befestigt.



	
<p>Kabelstecker 5-polig Gerade Amphenol, T 3300 001</p>	<p>Gerätedose 5-polig, Amphenol, T 3363 000</p>

Die Encoder werden in der Regel ohne Anschlussstecker geliefert. Wenn man später den Arduino und das Netzteil in einem geschlossenen Gehäuse unterbringen möchte, empfiehlt sich die Verwendung von Rundsteckern und einer endsprechenden Kabeldose im Gehäuse (siehe Beispiel von der Fa. Amphenol).

Der Encoder liefert an den zwei Ausgängen (A, B) eine um 90 Grad phasenverschobene Impulsfolge (in der Literatur auch als Quadratur Encoder bekannt). Aus der Auswertung beider Signale kann der Drehwinkel und die Drehrichtung erfasst werden.



Ausgangssignale A, B des Encoders



Für den Arduino gibt es eine fertige Programm-bibliothek [encoder.h](#), zur Auswertung der Signale. Dort wird erläutert, dass die beiden Encodersignale (A,B) an Pins mit Interrupt Fähigkeit gelegt werden sollten. Die folgende Tabelle zeigt für die verschiedenen Arduino Modelle die möglichen Pins. Der Autor hat für den Encoder des Pendelwinkels die Pins 20, 21

bei Arduino Megaboard benutzt. Später, in der zweiten Ausbaustufe, wurde der Encoder zur Erfassung der Linearposition an die Pins 18 und 19 angeschlossen.

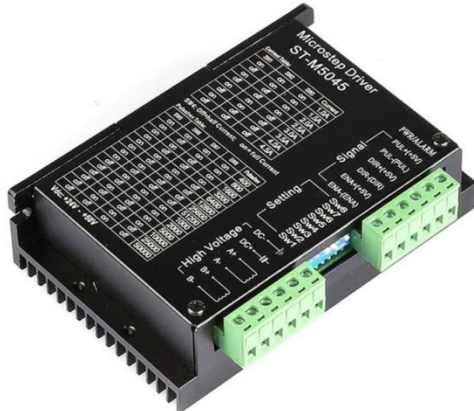

Board	Interrupt Pins	LED Pin (do not use)
Teensy 3.0 - 3.6	All Digital Pins	13
Teensy LC	2 - 12, 14, 15, 20 - 23	13
Teensy 2.0	5, 6, 7, 8	11
Teensy 1.0	0, 1, 2, 3, 4, 6, 7, 16	
Teensy++ 2.0	0, 1, 2, 3, 18, 19, 36, 37	6
Teensy++ 1.0	0, 1, 2, 3, 18, 19, 36, 37	
Arduino Due	All Digital Pins	13
Arduino Uno	2, 3	13
Arduino Leonardo	0, 1, 2, 3	13
Arduino Mega	2, 3, 18, 19, 20, 21	13
Sanguino	2, 10, 11	0

Der Pendelstab

Für den Pendelstab eignen sich die bei Bastlern beliebten Maker Beam ALU Profilstangen mit einer Länge von 200mm ... 300mm und einem Querschnitt von 10x10mm. Die größeren Profile mit 15mm x 15mm oder 20x20mm sind zu schwer. Als Befestigung an der Encoderachse hat sich die folgende Flanschkupplung für 6mm Wellen bewährt.

	
<p>Winwill® 1Pc 6mm Starre Flanschkupplung Motorführung Wellenkoppler Motoranschluss</p>	<p>Maker Beam ALU Profil oder Dold ALU Profile oder My Hobby CNC</p>

Der motorische Antrieb

	
<p>SainSmart CNC-Mikroschrittmotor Treiber 2M542 (30.-€)</p>	<p>Quimat Nema 17 Schrittmotor (€13.70)</p>

Beim rotierenden inversen Pendel benötigt man zu Steuerung eine sehr kleine Drehzahl. Die üblichen DC Motoren mit 3000 rpm wären dazu nicht geeignet. Die Verwendung eines Schrittmotors in Verbindung einer Microsteppdriver Stufe hat sich als ideal erwiesen. Beim Kauf des Microstepp Drivers rät der Autor von Billigprodukten wie dem [Modul TB6600](#) ab. Diese erzeugen Resonanzen im Schrittmotor die sich über den Rotorstab auf das Pendel übertragen. Ein solides Produkt ist [TM542](#), das auch in der Lage ist größere Schrittmotoren wie den NEMA 23 zu steuern. Beim Einsatz von Schrittmotoren ist ein 24DC Netzteil besser geeignet. Für die Befestigung des Schrittmotors an den ALU Profilen gibt es geeignet Winkel (siehe Stückliste).

Stepper Libray zum Betrieb des Schrittmotors

Die Geschwindigkeit und Drehrichtung wird über je einen digitalen Impulsausgang des Arduino gesteuert. Im Zuge der CNC-Fräsen und 3D-Drucker wurden für den Arduino eine Reihe von Bibliotheken zur Ansteuerung von Schrittmotoren geschrieben. Nicht alle diese Bibliotheken sind für die Regelung des Pendels geeignet, da sie teilweise zu langsam sind. Der Autor hat gute Erfahrungen mit dem Einsatz der Bibliothek [SteppDriver](#) gemacht. Ein Auszug aus dem Programmcode sieht wie folgt aus:

```
#include <PID_v1.h>
#include <Encoder.h>
#include "BasicStepperDriver.h"

// Motor steps per revolution. Most steppers are 200 steps or 1.8
degrees/step
#define MOTOR_STEPS 400

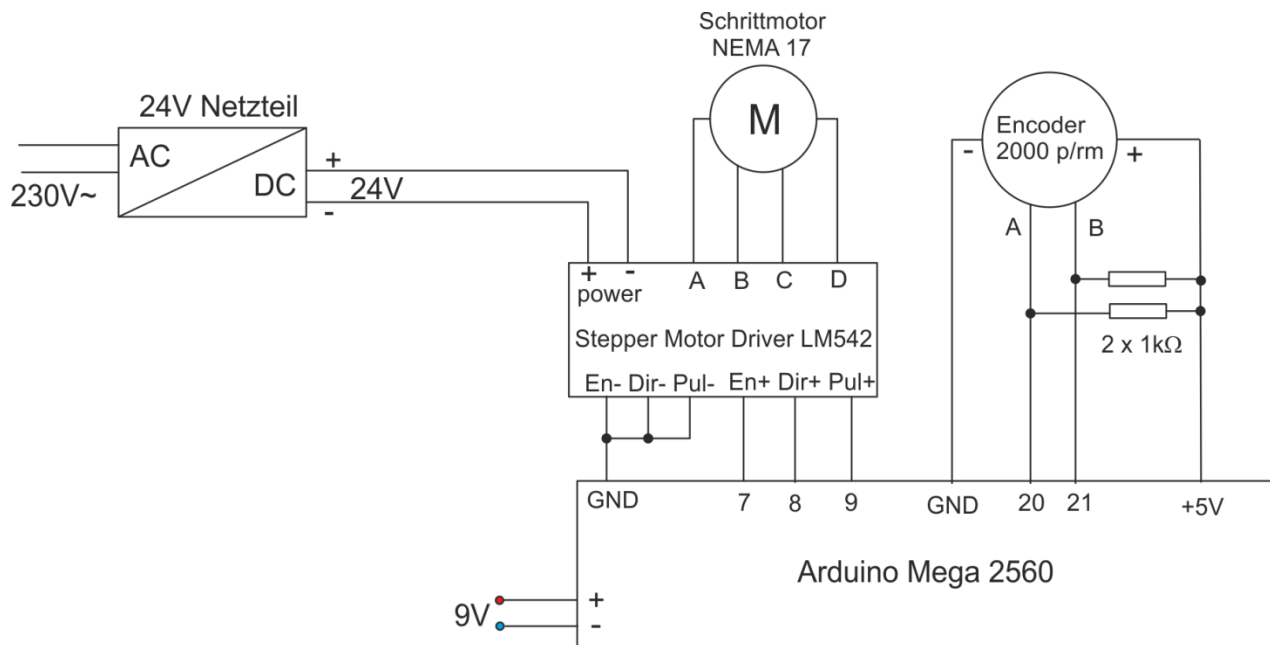
// All the wires needed for full functionality
```

```
#define ENBL 7
#define DIR 8
#define STEP 9

// Since microstepping is set externally, make sure this matches the selected
mode
// 1=full step, 2=half step etc.
#define MICROSTEPS 1
// 2-wire basic config, microstepping is hardwired on the driver
BasicStepperDriver stepper(MOTOR_STEPS, DIR, STEP);
```

Der Stromlaufplan

Für die Verdrahtung zwischen Encoder und Arduino genügt ein kleines Breadbord, um die 1 kOhm Abschlusswiderstände zu stecken. Die vier Kabel des Schrittmotors werden direkt mit dem Microstepdriver verbunden. Man könnte für den Schrittmotor NEMA 17 und dem Arduino auch ein gemeinsames 12 Volt Netzteil verwenden.



Auswahl des Regler Algorithmus

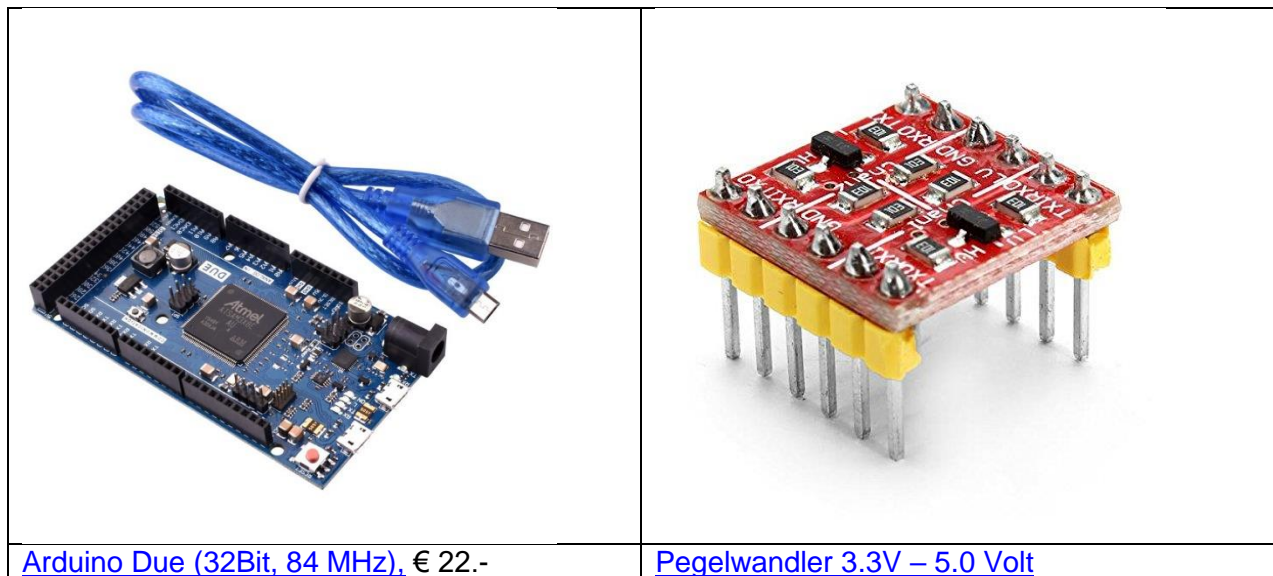
Bei der Auswahl und Implementierung des Regelalgorithmus spielen folgende Aspekte eine wichtige Rolle:

- verfügbare Rechenleistung des Mikrokontrollers
- erforderliche Abtastrate
- Kenntnistand auf dem Gebiet der Regelungstechnik und Mathematik

- Wie genau können die Zustandsgrößen (Weg, Geschwindigkeit, Pendelwinkel, Winkelgeschwindigkeit) und Modellparameter (z.B. Reibung) erfasst werden?
- Genauigkeit der Sensoren

Im Internet finden sich unter dem Suchwort *Inverses Pendel* bzw. *invert pendulum* zahlreiche Bachelor- und Masterarbeiten (siehe Links auf Seite 2). Dort werden die gängigen Konzepte wie PID Regler oder Zustandsregelung umgesetzt. In Ausnahmen kommen Fuzzycontroller oder neuronale Netze zum Einsatz. Bei der Zustandsrückführung ist eine tiefer gehende, mathematische Betrachtung notwendig. Auf YouTube findet man unter dem Suchwort Control Bootcamp eine gut verständliche Vortragserie zu modernen Regelungskonzepten.

Bei Verwendung genauer Sensorik (hochauflösende Encoder, stabile Mechanik) und Einsatz des Arduino **Due** Boards (84 MHz) könnte man eine Zustandsregelung mit Beobachter und Kalmanfilter implementieren. Das Arduino Due Board besitzt die gleichen Anschlüsse wie das Arduino Mega Board. Unbedingt zu beachten ist, dass der Arduino Due nur Ein- bzw. Ausgangsspannungen bis 3.3 Volt verträgt, der Arduino Mega hingegen mit 5.0 Volt arbeitet. Viele Encoder besitzen als Ausgangsstufe einen offenen Kollektor. In diesem Fall verbindet man den Arbeitswiderstand (1 kOhm) mit dem 3.3V Pin des Due. Andernfalls müssen Pegelwandler 3.3 V – 5.0 V zum Einsatz kommen.



Programmierung des PID Reglers für den Arduino

Für die Auswertung der Encoder Signale wird die encoder.h Library benutzt:

<https://github.com/PaulStoffregen/Encoder>

Eine genaue Erläuterung zur Funktion der Encoder Bibliothek findet man unter:

https://www.pjrc.com/teensy/td_libs_Encoder.html

Für den PID Algorithmus gibt es eine fertige Bibliothek für den Arduino, die sich als recht stabil bewährt hat: <https://playground.arduino.cc/Code/PIDLibrary>

```
#include <PID_v1.h>
#include <Encoder.h>
#include "BasicStepperDriver.h"

// Motor steps per revolution. Most steppers are 200 steps or 1.8
degrees/step
#define MOTOR_STEPS 400

// All the wires needed for full functionality
#define ENBL 7
#define DIR 8
#define STEP 9

// Since microstepping is set externally, make sure this matches the selected
mode
// 1=full step, 2=half step etc.
#define MICROSTEPS 1
// 2-wire basic config, microstepping is hardwired on the driver
BasicStepperDriver stepper(MOTOR_STEPS, DIR, STEP);

// Definition der Encoderanschlüsse
#define rot_Encoder_A 20 // rotary Encoderpin A an Pin 20
#define rot_Encoder_B 21 // rotary Encoderpin B an Pin 21
Encoder rotEnc(rot_Encoder_A, rot_Encoder_B);

// angle Limits for Controller
double limit_angle= 350; //limit at which it becomes impossible to catch
the pendulum again if exeeded,
byte output_deadband= 5; //Deadband Variables

//Miscellaneous Variables
double angle, y=0;
double a = 0.80; // Filterkonstante zur Dämpfung

//PID Setup
double rot_kp= 2.5;
double rot_ki= 55.0;
double rot_kd= 0.0;
double Output= 0.0;
double rot_Setpoint = 0.0;

//PID Controller initialisation
PID
rot_Controller(&angle, &Output, &rot_Setpoint, rot_kp, rot_ki, rot_kd, DIRECT);

void setup() {
  pinMode(ENBL, OUTPUT);
  digitalWrite(ENBL,1); //stepper aus
  Serial.begin(115200);
  //Initialize Encoder counts
  //full pendulum rotation = 8000 Encoder counts
  rotEnc.write(-4000); //set when pendulum is hanging straight down due to
gravity
```

```
//PID Settings
  rot_Controller.SetMode(AUTOMATIC);
  rot_Controller.SetOutputLimits(-180,180);
  rot_Controller.SetSampleTime(5); //2
  rot_Controller.SetTunings(rot_kp,rot_ki,rot_kd);
}

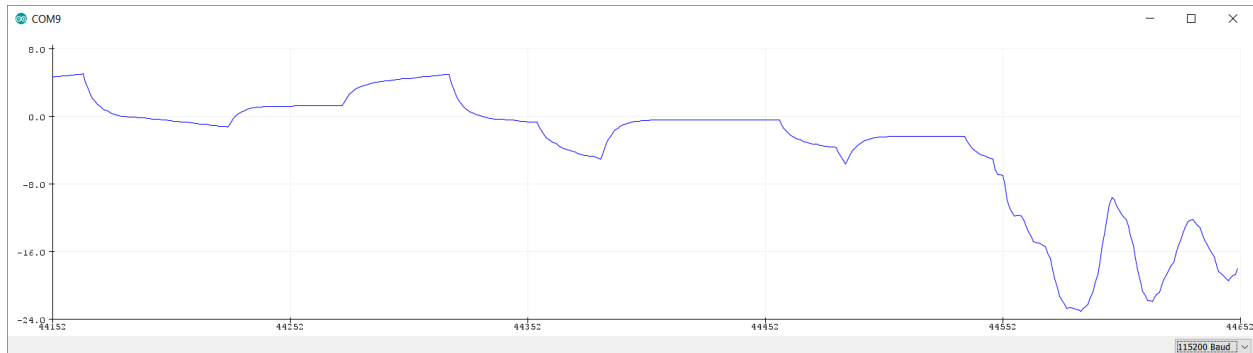
void loop() {
  angle = rotEnc.read();
  if (angle < limit_angle && angle > (limit_angle*-1)){
  //perform as long as pendulum is within the angular limits
    //Execute angular PID controller
    rot_Controller.Compute();
    y = a * y + (1.0 - a) * Output;
    Serial.println(y); // Ausgabe der Stellgröße auf den Monitor
    digitalWrite(ENBL,0); // Motor aktivieren
    if(y > output_deadband){
      stepper.begin(y, MICROSTEPS);
      stepper.move(-MICROSTEPS);
    }
    else if(y < -output_deadband){
      stepper.begin(abs(y), MICROSTEPS);
      stepper.move(MICROSTEPS);
    }
  } //stop motor if pendulum has tipped out of the angular limits
  else {
    digitalWrite(ENBL,1);
  }
} //End Loop
```

Inbetriebnahme des Pendels

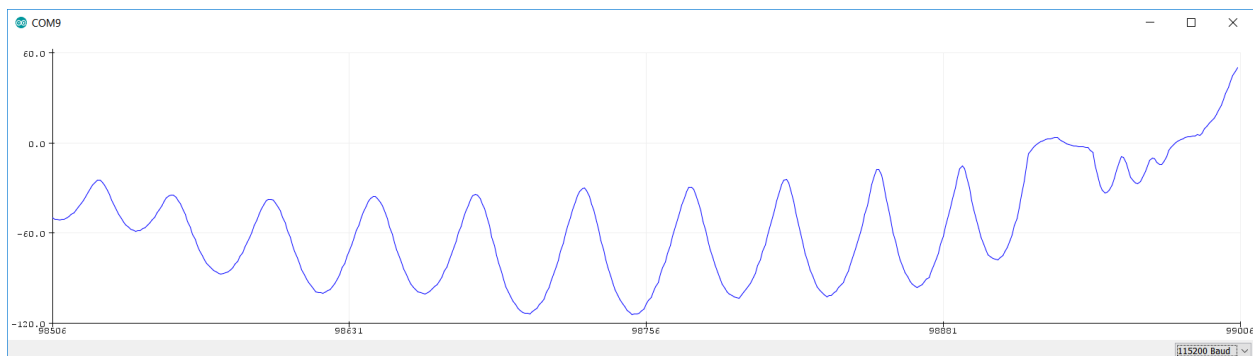
Nach dem Laden des Programms und einschalten der 24V Stromversorgung für den Schrittmotor, muss das Pendel absolut ruhig und senkrecht nach unten hängen. Jetzt wird kurz die rote Resettaste auf dem Arduino Board betätigt. Damit wird der Encodercounter mit dem Referenzwert -4000 geladen (eine volle Umdrehung sind 8000 Impulse, da die Encoderbibliothek die 2000 Impulse vervierfacht (Auswertung der steigenden und fallenden Signalfanken).

Danach bringt man das Pendel im Uhrzeigersinn in die aufrechte Position. Der Regler beginnt zu arbeiten und sollte das Pendel jetzt aufrecht halten. Sollte der Rotarm stets in die falsche Richtung steuern, müssen die Schrittmotoranschlüsse am Microstepdrivermodul getauscht werden. Weiterhin müssen die PI Reglerparameter Kp und Ki abgeglichen werden (leicht absenken oder erhöhen). Der Rotor sollte um einen kleinen Drehwinkel von +/- 15 Grad hin- und herschwingen. Eventuell muss die Grundplatte des Pendels waagrecht ausgerichtet

werden, falls das Pendel immer nach einer bestimmten Seite hin abdriftet. Zur Kontrolle der Reglerausgabe wird der serielle Plotter in der Arduino Box am PC geöffnet. Das Signal sollte innerhalb von ± 40 verbleiben (siehe Graphik). Ist der Kp Wert zu hoch eingestellt, schwingt sich der Regler auf und die maximale Stellgröße von ± 180 auf und das Pendel kippt ab.



Kp und Ki sind richtig eingestellt



Kp und Ki sind zu hoch gewählt, das Pendel schwingt sich langsam auf

Aufschwingen des Pendels aus der Ruhelage

Eine weitere Herausforderung besteht darin, dass Pendel aus der Ruhelage in eine annähernd aufrechte Position aufzuschwingen. Im Internet findet man unter dem Suchwort *Swing Up Strategy* einige Konzepte. Eine mathematisch fundierte Betrachtung findet man in der Arbeit von [K. J. Åström](#).

Energetisch am günstigsten ist es, dass Pendel im Nulldurchgang (Pendelwinkel = 0) periodisch anzustoßen. Ein weitere Möglichkeit besteht darin im jeweils linken bzw. rechten Umkehrpunkt das Pendel anzustoßen (Winkelgeschwindigkeit = 0).

Stückliste

Lfd. Nr.	Bezeichnung	Anzahl	Quelle	Preis
1	Arduino Mega 2560	1	Amazon	13.-
2	NEMA 17 Schrittmotor	1	Amazon	17.-
3	M542 MircoStep Driver	1	Amazon	28.-
4	Schaltnetzteil 240W, DC 24V, 10A	1	Amazon	18.-
5	OMRON Rotary Encoder E6B2-CWZ6C 2000P/R 5V-24V	1	ebay	27.-
6	UEETEK NEMA 17 Schrittmotor Halterung w/M3 Schrauben - 2 Stück (als Befestigung für den Motor und Encoder)	1	Amazon	7.60€
7	ALU Profile 30 B-Typ Nut 8 300mm lang	2	Dold Mechatronic	3.90€
8	ALU Profil 30 B-Typ Nut 8 100mm lang	1	Dold Mechatronic	1.30€
9	ALU Profil Typ 20 B-Typ Nut 6, 200mm lang	1	Dold Mechatronic	0.85€
10	Innenwinkel Zinkdruckguss 30 B-Typ Nut 8	2	Dold Mechatronic	2.88€
11	Winwill® 1Pc 6mm Starre Flanschkupplung Motorführung Wellenkoppler Motoranschluss	1	Amazon	1.50€
12	Maker Beam Alu Profil 10x10, 300mm lang (als Pendelstab)	1	myhobby-cnc shop	2.50€
13	MagiDeal Aluminium Schrittmotor Flexible Wellenkupplung	1	Amazon	1.50€