
IT-Projekt Lageregelung

August 2020

Inhaltsverzeichnis

IT-Projekt.....	1
Inhaltsverzeichnis.....	2
Ziele.....	3
Servoantrieb.....	4
Programm zum Betrieb des Servoantriebs.....	6
Lagemessung.....	8
Programmbeispiel zur Lagemessung.....	9
Regelung.....	11
Dokumentation.....	13
Doxygen.....	13
SVN.....	13
Fritzing.....	13
Gnuplot.....	14

Ziele

Hier sind ein Lagesensor und ein Servoantrieb so zu verschalten und durch ein Programm mit einer Regelung zu bedienen, dass eine Plattform (Servo) stabilisiert bzw. balanciert wird.

Zunächst ist die geplante, grundlegende Arbeitsweise von Sensor und Servo zu ermitteln. Neben den vorgegebenen Unterlagen ist auch eine Recherche im Internet sinnvoll. Daraus folgt eine Definition der erforderlichen Arbeitsschritte und es kann ein Pflichtenheft über die einzelnen Funktionen der Regelung erstellt werden.

Danach sollen ein oder mehrere geeignete Programme in C/C++ für ein Arduino-Board erstellt werden.

Die erreichte Funktionalität ist zu dokumentieren. Ziel ist eine Beschreibung des Programms und seiner Funktionen. Hierzu sollten auch Zeitverläufe von ausgewählten Vorgängen oder Teilvorgängen aufgezeichnet werden.

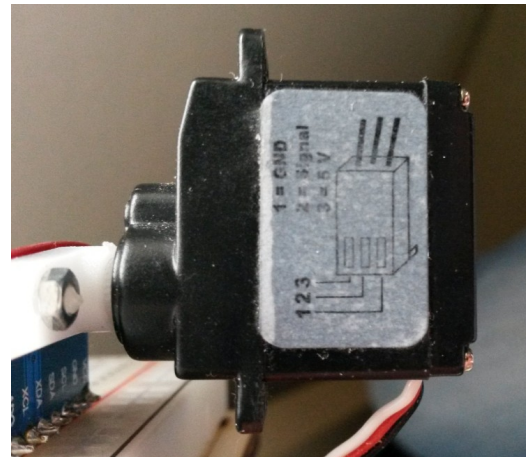
Die Lösung dieser Grundaufgabe soll passend erweitert werden. Hier können eigene Ideen verwirklicht werden. Ein Beispiel wäre eine Neigungsanzeige auf einem Display oder Smartphone.

Ebenfalls möglich wären numerische Untersuchungen etwa betreffend der Linearität, evtl. mit weiteren Programmen speziell für diese Untersuchungen.

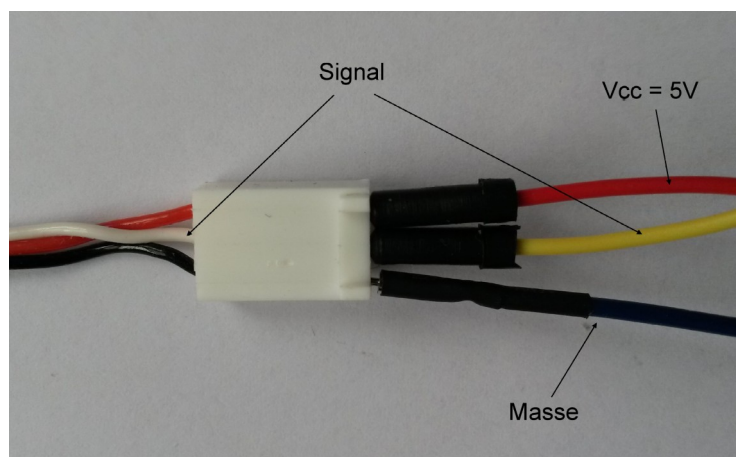
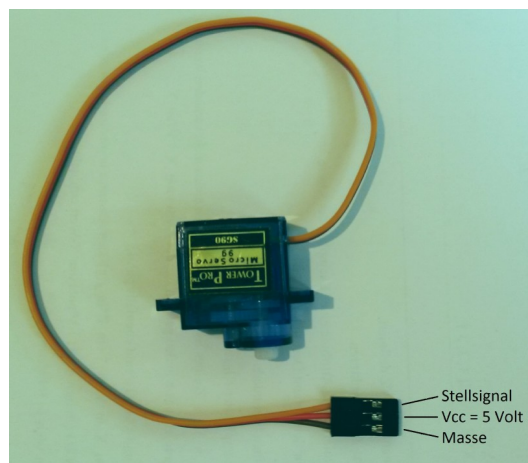
Das IT-Projekt kann auch durch Teile ergänzt werden, die nicht unmittelbar in die Programmierung einfließen. Eine Software-Dokumentation mittels Doxygen oder ein Versionskontrollsystem auf der Basis von SVN sind solche Ergänzungen.

Servoantrieb

Als Antrieb dient ein miniaturisierter Servo-Antrieb, wie er beispielsweise in Spielzeug und Flug-Schiffs- oder Auto-Modellen verwendet wird.



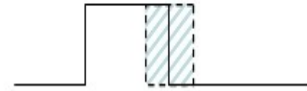
Derartige Servos arbeiten üblicherweise mit 5 Volt Betriebsspannung oder wenig mehr.



Als Stellsignal wird ein pulsbreiten-moduliertes Signal (PCM) verwendet. Normalerweise mit einer Taktzeit von ca 20 msec und einer Pulsbreite von 1 bis 2 msec.

Digital-Proportional-Verfahren

Pulslängenvariation 1,5 ms +/- 0,5 ms, z.B.



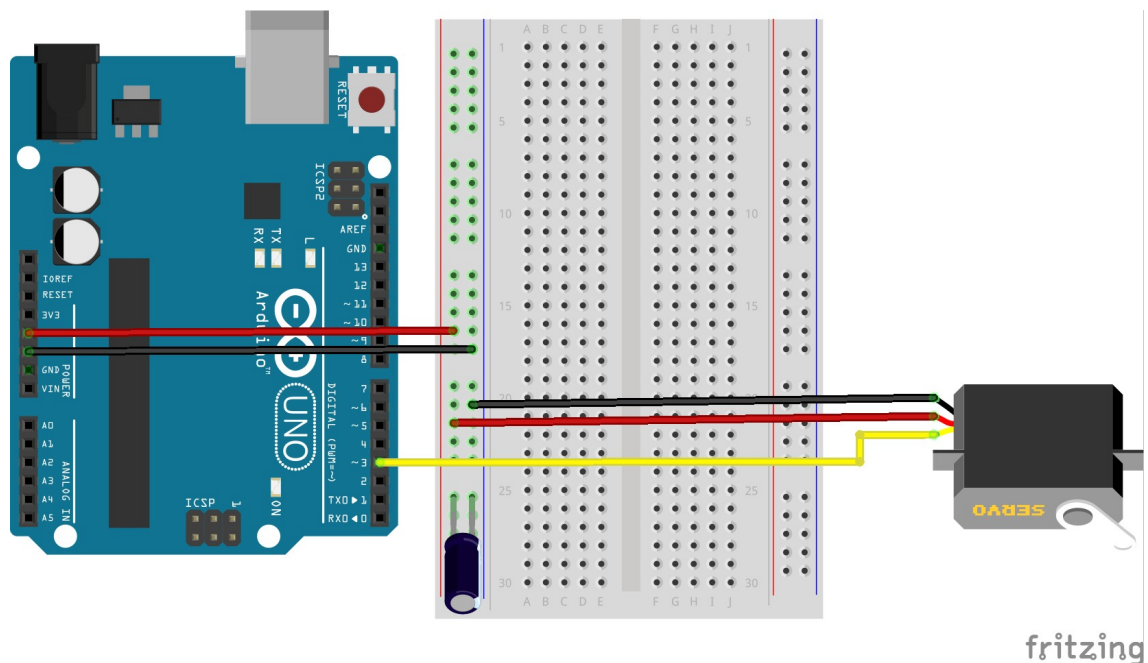
Die Pulsbreite codiert hierbei den zu übertragenden Wert, z.B. 1 msec = 0% und 2 msec = 100%.

ACHTUNG, die Anschlüsse sind bei manchen Servos vertauscht. Prüfen!

Programm zum Betrieb des Servoantriebs

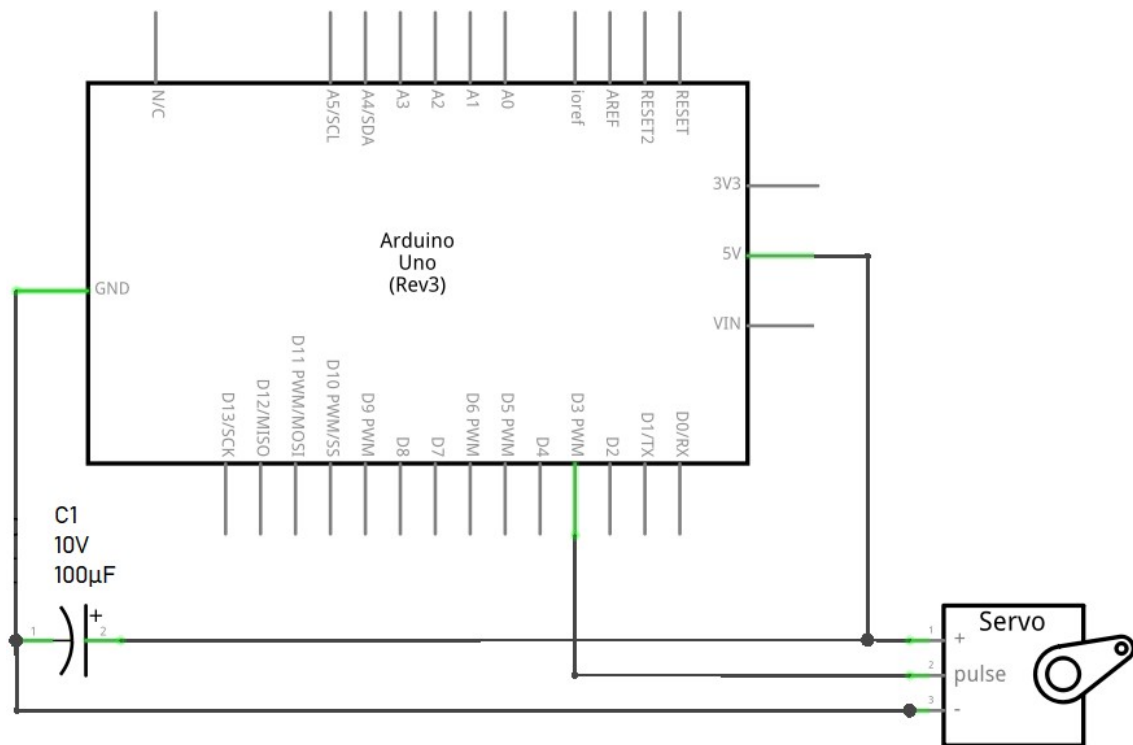
Für erste Experimente kann ein einfaches Programm zur gezielten Bewegung des Servoantriebs erstellt werden. Eine passende Grundlage hierzu findet man in der Servo-Bibliothek des Arduinos – vgl. <https://www.arduino.cc>.

Als erstes Beispiel könnte der Servo kontinuierlich zwischen den beiden Endstellungen hin- und hergefahren werden. Eine Alternative wäre eine Einstellung auf Werte, die über die serielle Schnittstelle (USB in der Arduino IDE) interaktiv vorgegeben werden.



Sinnvoll wären beispielsweise auch Überlegungen, ob in Prozentwerten oder in Winkeln gerechnet werden soll. Evtl. skalieren.

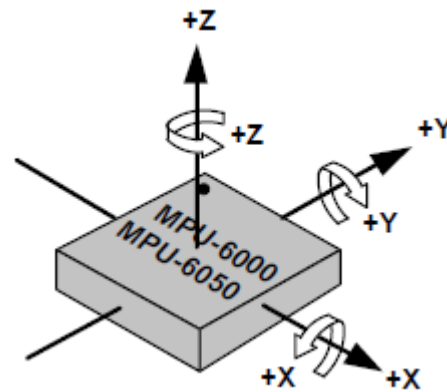
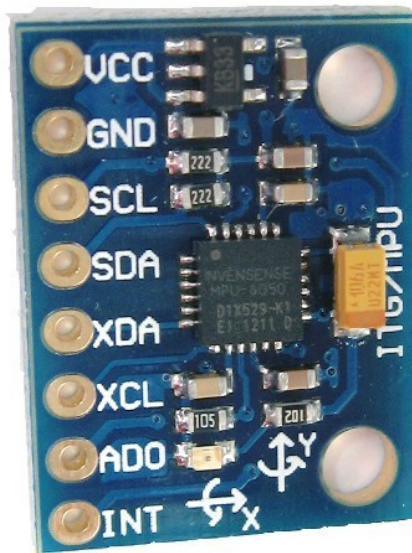
Hier gezeigt das Anschlussschema.



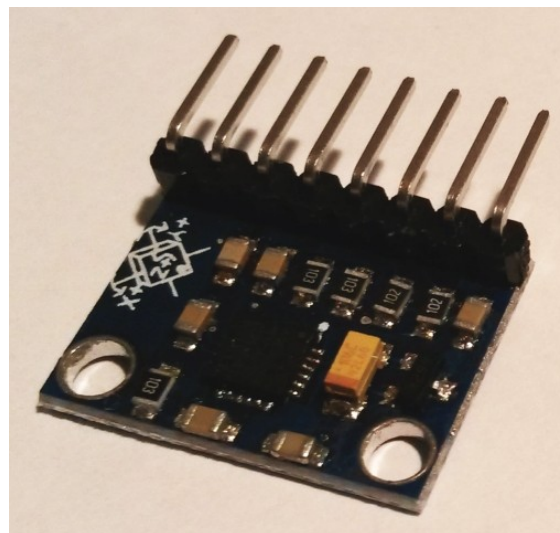
Der Elektrolyt-Kondensator (links unten) gleicht Spannungsschwankungen aus, die der Servo gelegentlich, vor allem bei Anlauf erzeugt. 100 μF sind dazu ausreichend.

Lagemessung

Zur Lagemessung dient ein 6-Achsen Gyro-Baustein vom Typ MPU-6050 wie er in ähnlicher Form auch in Smartphones Verwendung findet.



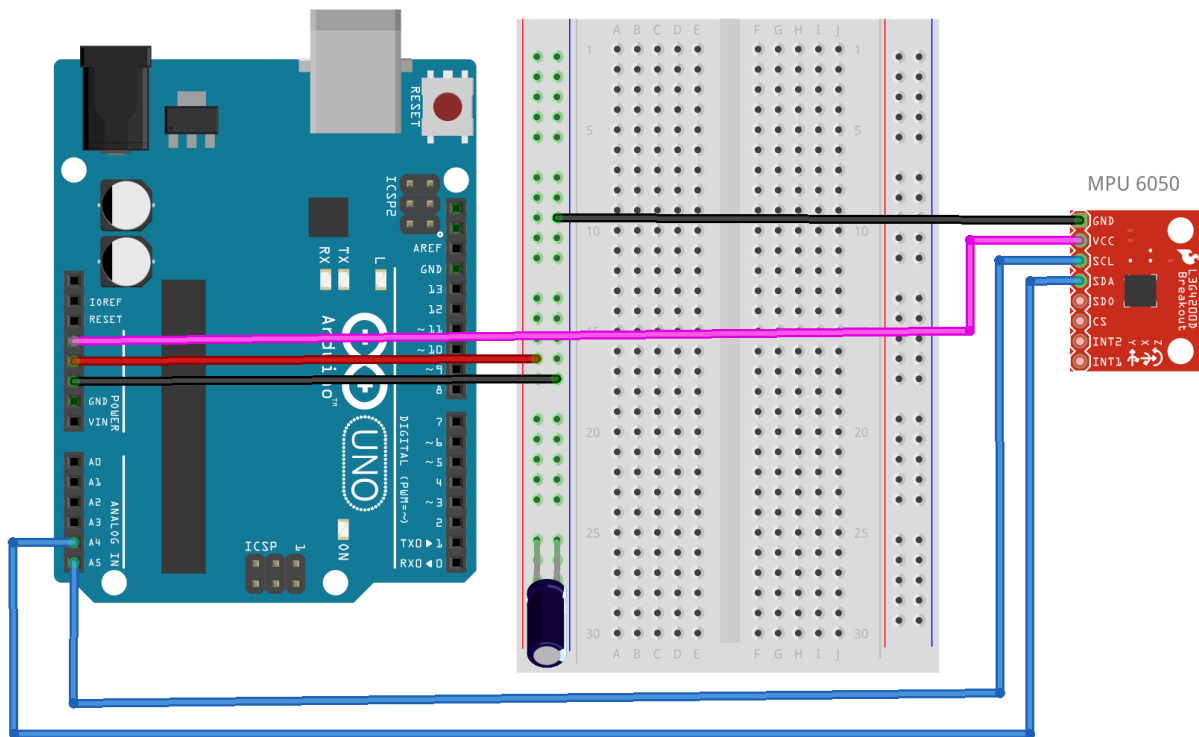
Zur besseren Handhabbarkeit ist dieser auf einer kleinen steckbaren Platine angebracht.



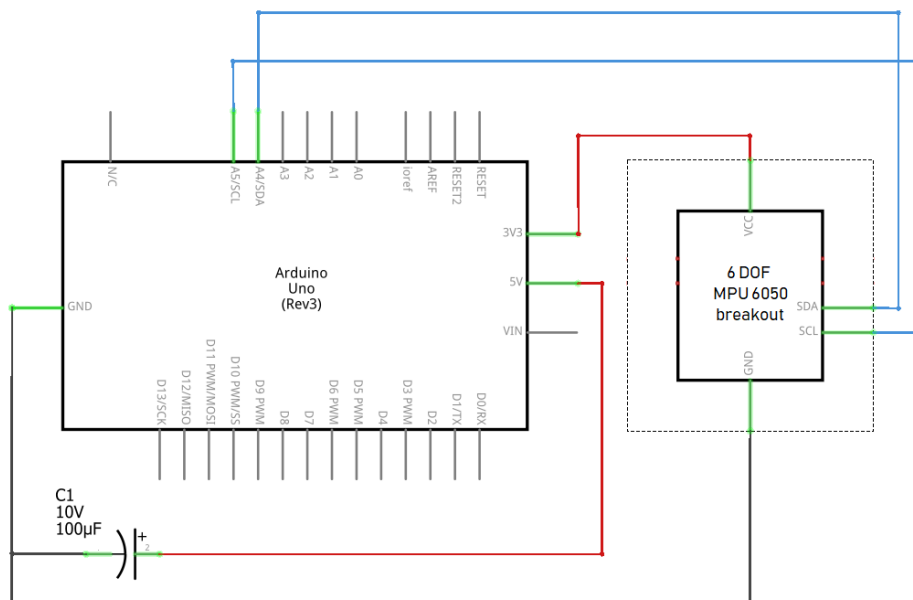
ACHTUNG, der Vcc- Anschluss muss an 3.3 Volt angeschlossen werden.

Programmbeispiel zur Lagemessung

In der Datei Read_MPU_6050.ino liegt ein Programm zur Messung der Winkel-Koordinaten (und erforderlichenfalls Winkel-Beschleunigungen) vor. Dieses Programm dient lediglich für erste Experimente mit dem Sensorbaustein.



Hier zeigt das Anschlussschema.



Das Programm liest kontinuierlich die Werte aus dem Sensor und zeigt sie an. Für die Lageregelung mittels des Servos sind nicht alle angezeigten Werte erforderlich. Hier gibt es Optimierungsmöglichkeiten. Allerdings sollten Überlegungen zu dem 3-dimensionalen Messvektoren gemacht werden. Betrachtungen zu den geometrischen Verhältnissen sind evtl. nötig.

Der Sensor ist mit dem Arduino über eine I²C (TWI) Busverbindung angeschlossen.

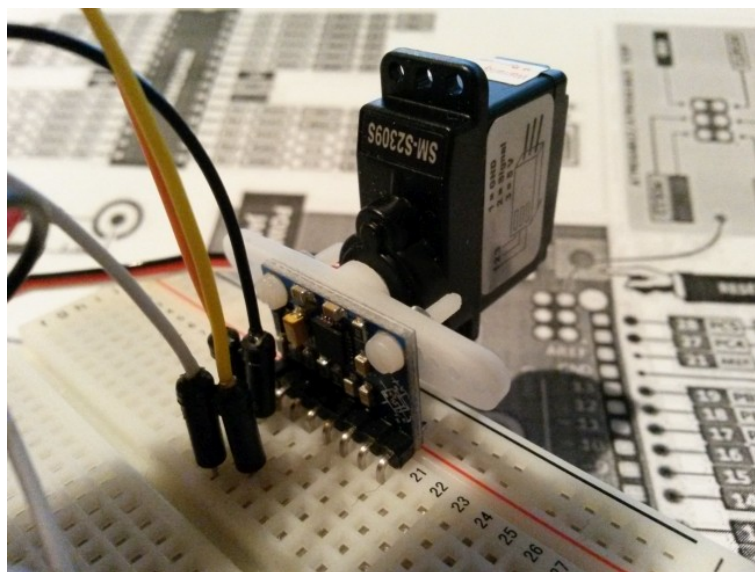
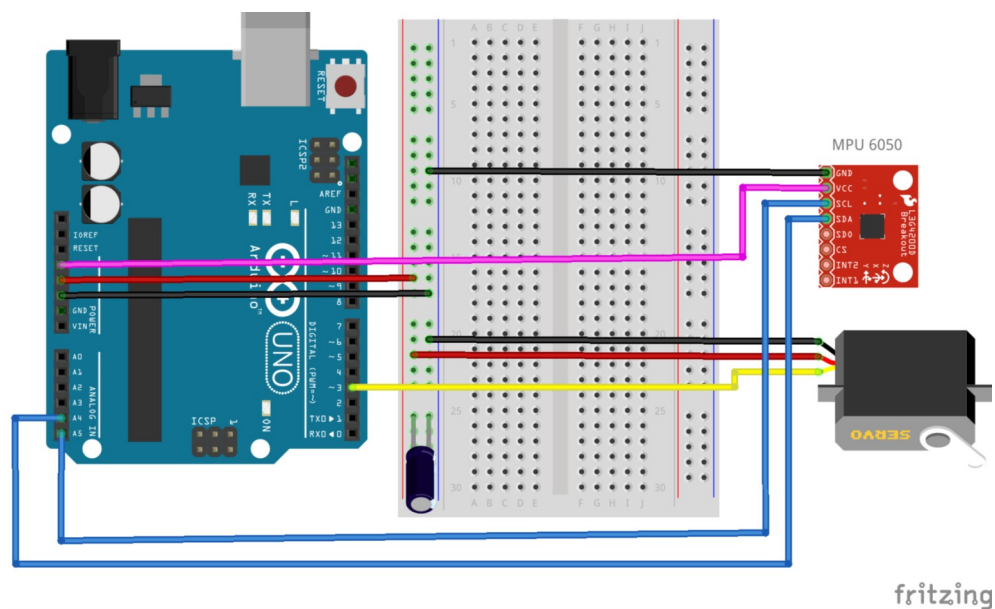
Im Internet gibt es vielfach verbesserte Versionen von Beispielen und Bibliotheken – vgl. etwa <http://playground.arduino.cc/> und <https://playground.arduino.cc/Main/MPU-6050>. Hiermit kann die Genauigkeit erhöht werden.

Regelung

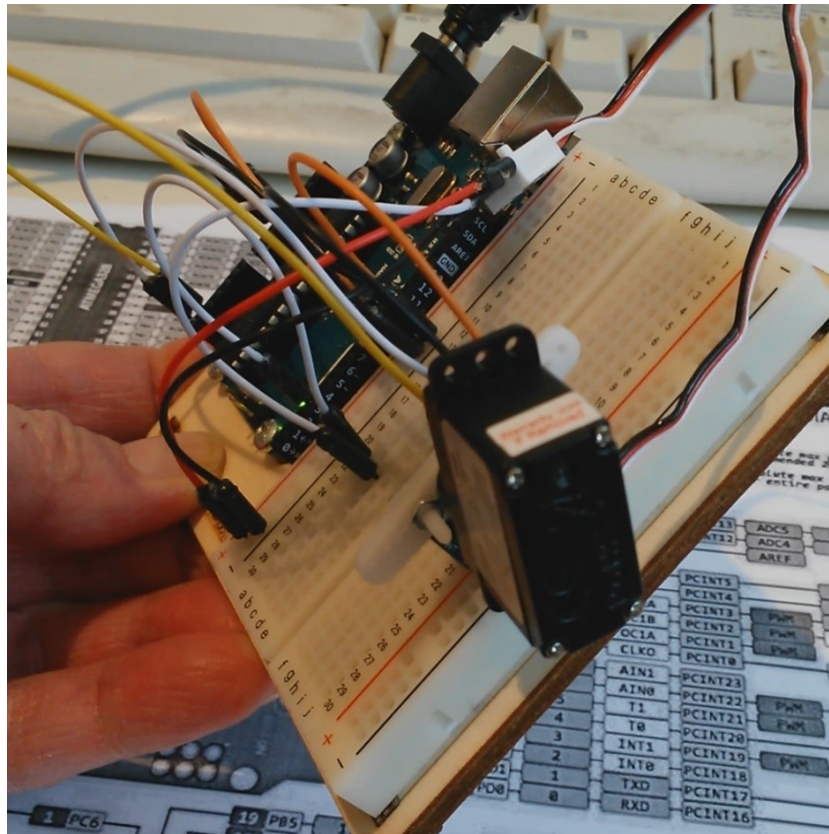
Im Versuchsaufbau sind Lagesensor und Servo fest miteinander und mit einer Grundplatte verbunden. Das Ziel einer Regelungen ist

der Servo aufrecht stehen bleibt, wenn die Grundplatte gekippt wird (balancieren).

Hier gezeigt das komplette Anschlussschema.



Balancieren



Die Art der Regelung dürfen Sie selbst wählen. Schon mit einer einfachen P-Regelung können gute Ergebnisse erzielt werden. Kümmern Sie sich auch um “Schmutzeffekte” - z.B. wenn das Servo nicht rechtwinklig über der Grundplatte steht, ist ein passender Offset in der Winkelberechnung nötig. Gegebenenfalls sind auch die erreichbaren Endstellungen zu berücksichtigen.

Eventuell ist es angenehmer als Spannungsquelle eine 9V-Batterie zu verwenden.

Dokumentation

Ein wesentlicher Teil des IT-Projektes ist eine Dokumentation. Diese sollte aus einer funktionalen Beschreibung der Regelung und der erstellten Software hierfür bestehen.

Die Form der Dokumentation ist freigestellt. Typisch und häufig verwendet wird hierzu Word. Auch Fotografien, ggf. Video sind inzwischen als erläuternder Teil in Dokumentationen gängig.

Es ist aber auch möglich (und vielleicht sogar handlicher und leistungsfähiger) die Dokumentation direkt im Quellcode einzupflegen und mit Doxygen zu erstellen.

Doxygen

Doxygen ist eine weitverbreitete Software zur Dokumentation von Software innerhalb ihres Quellcodes. Gesteuert von speziellen Markierungen in den Quelldateien kann sie Dokumentationen in verschiedenen Formaten erstellen. Hier im IT-Projekt ist eine Dokumentation in Form von HTML-Seiten und kompiliertem HTML etwas vorbereitet. Die HTML-Seiten werden mit einem HTML-Compiler in CHM-Dateien umgesetzt und komprimiert.

Siehe auch den Quellcode der Regelung.ino, Regelung.CHM und Regelung.doxy.

SVN

Software soll meist über den Tag hinaus bestehen bleiben und muss reproduzierbar sein. Da Software meist weiterentwickelt wird, sind zur Fehlersuche häufig auch die vorangegangenen Versionen wichtig. Als erstes Backup und zur Versionsverwaltung werden verschiedene Software eingesetzt.

Eine solche, weit verbreitete und moderne Versionsverwaltung ist SVN (Apache Subversion). Die Software des IT-Projektes soll auf einem SVN-Server hinterlegt werden.

Fritzing

Fritzing ist eine Entwurfs-Software für Arduino und andere derartige Boards. Sie kann beispielsweise zur Darstellung von Steckplatine und Schaltplan verwendet werden.

Eine Alternative aber deutlich komplexer ist KiCAD.

Gnuplot

Gnuplot ist eine viel verwendete, freie Software zur Anzeige von datenbasierten Graphiken. Mit ihr kann u.a. das zeitliche Verhalten der Anlage in Form von y-t-Diagrammen dargestellt werden.

Im einfachsten Fall wird am USB/COM-Anschluß des Arduino Boards ein speicherndes Terminal angeschlossen (z.B. <https://sourceforge.net/projects/realterm/>) und die Ausgaben bzw. Messdaten in eine Textdatei protokolliert.

Erforderliche Eingaben im Gnuplot Kommando-Fenster wenn die Messdaten in der Datei C:\irgendwo\Balancing.dat vorliegen.

```
set title("Winkel");
set xlabel("sec");
set ylabel("");
set grid;
plot "Balancing.dat" using 1:2 with lines
```

Auch eine kontinuierliche Darstellung während des Betriebes ist möglich.

```
set title("Winkel");
set xlabel("sec");
set ylabel("");
set grid;
plot "Balancing.dat" using 1:2 with lines
pause 2
reread
```

Alternativ kann auch mit dem Plotter der Arduino IDE gearbeitet werden – vgl. ArduinoPlotter.