

# Hubstützen-Projekt Informatik

## Inhaltsverzeichnis


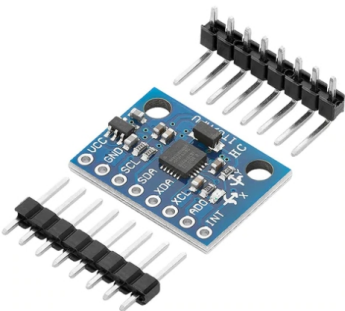
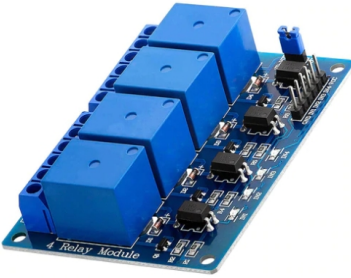
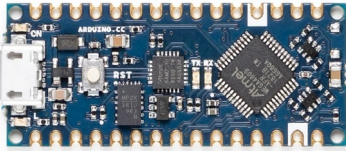
Die Idee.....	2
Die Hardware/Materialien.....	3
GY-521 MPU-6050 3-Achsen-Gyroskop und Beschleunigungssensor.....	3
4-Relais Modul 5V mit Optokoppler Low-Level-Trigger.....	3
Arduino Nano Every.....	3
Die Schaltung.....	4
Das Gyroskop.....	5
Der Code.....	6
Ende.....	7



# Die Idee

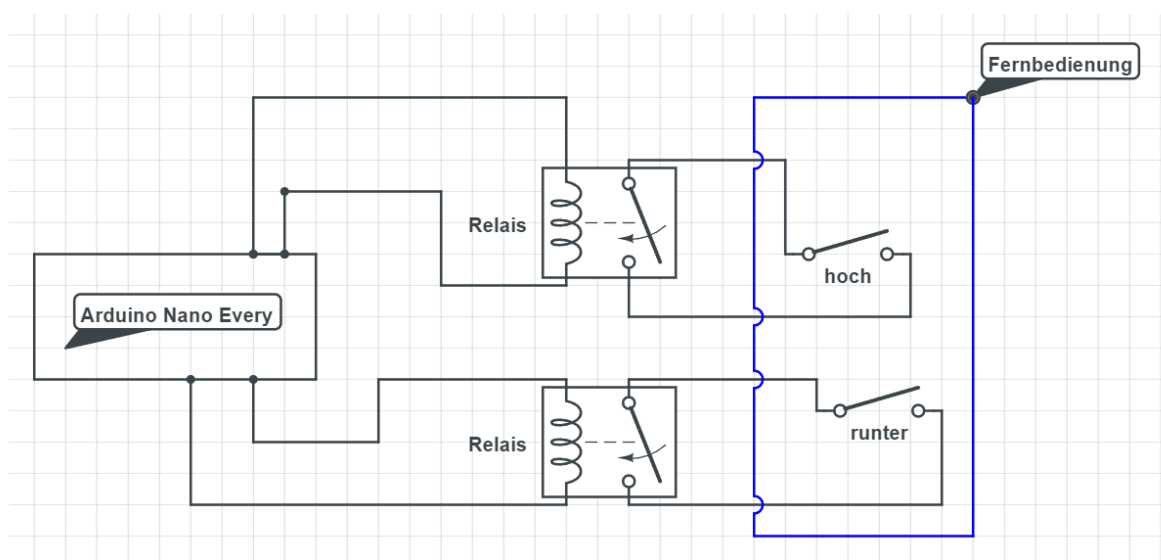
Jeder Camping-Begeisterte wird das Problem kennen: Nach einem gemütlichen Grillabend mit Freunden ist man todmüde und möchte sich entspannt ins Bett im Reisemobil legen. Der Haken an der Sache: Das Bett ist schief, weil der Boden nachgegeben hat oder weil man mit einer Wasserwaage nicht so genau manövrieren kann, dass alles perfekt gerade ist. Das hört sich jetzt halb so wild an, ist allerdings sehr nervig auf Dauer und wird vor allem bei längeren Trips zu einem großen Problem. Entweder man liegt dann mit den Füßen weiter oben als mit dem Kopf, woraufhin man mit Kopfschmerzen aufwacht oder man wird durchgehen leicht nach rechts oder links gedrückt. Natürlich kann man dieses Problem mit sogenannten Auffahrkeilen beheben, allerdings niemals genau genug um die minimalen Schräglagen ausgleichen zu können. Außerdem können die Auffahrkeile über längere Zeit je nach Untergrund leicht in den Boden einsinken und sind dann auch nicht mehr in der Lage das Reisemobil geradezustellen. Deshalb kam mir die Idee mit den Hubstützen: Könnte man nicht einen oder zwei Wagenheber nehmen, unter das Wohnmobil montieren und dann nach Belieben die Schräglage selbst beheben? Als ich also nach Wagenhebern im Internet gesucht habe, stieß ich auf diese elektrische Wagenheber, die ursprünglich dazu gedacht waren ein Auto im Fall einer Panne anzuheben um einen Reifen wechseln zu können. Sie werden benötigen eine Spannung von 12V, also das was ein handelsüblicher Zigarettenanzünder in einem Auto bereitstellen kann. Dann dachte ich mir, dass man den Prozess eigentlich noch vereinfachen könnte, indem man eine elektrische Wasserwaage nimmt, die die derzeitigen Werte der Schräglage an einen Mikrocontroller weitergibt, welcher dann die Wagenheber je nach Bedarf ansteuert und so das Wohnmobil ausrichtet. Also bestellte ich die benötigten Teile und fing an zu bauen und programmieren. Allerdings habe ich mich jetzt erstmal auf den hinteren Teil des Wohnmobils konzentriert, da dort das Bett ist und zwei Stützen mehr das Budget sprengen würden.

# Die Hardware/Materialien

Bild	Beschreibung	Link
	<p>Elektrischer Wagenheber, der eine Spannung von 12V benötigt um maximal 2T heben zu können. Dabei ist eine Fernbedienung mit den Knöpfen „hoch“ und „runter“.</p>	<p><a href="https://bit.ly/3HH2qyk">https://bit.ly/3HH2qyk</a></p>
	<p><b>GY-521 MPU-6050 3-Achsen-Gyroskop und Beschleunigungssensor</b></p> <p>Dient uns als elektrische Wasserwaage um die Schräglage zu ermitteln</p>	<p><a href="https://bit.ly/3JccEpF">https://bit.ly/3JccEpF</a></p>
	<p><b>4-Relais Modul 5V mit Optokoppler Low-Level-Trigger</b></p> <p>Vier Relais auf einer kompakten Platine</p>	<p><a href="https://bit.ly/3GDw851">https://bit.ly/3GDw851</a></p>
	<p><b>Arduino Nano Every</b></p> <p>Ein kleiner, kompakter Mikrocontroller, der etwas mehr Leistung hat als sein Vorgänger</p>	<p><a href="https://bit.ly/3GC6c9M">https://bit.ly/3GC6c9M</a></p>

# Die Schaltung

Das erste mit dem ich mich beschäftigen musste war natürlich wie ich die Wagenheber ansteuern sollte, denn einfach nur Servomotoren an die Fernbedienungen zu schrauben, welche dann die Knöpfe drücken schien mir etwas zu primitiv. Also habe ich die Fernbedienungen aufgeschraubt und was mich dort erwartet hat war erstmal eine Katastrophe. Die Lötstellen da drin waren ein einziger Alptraum, das hat auch den Wackelkontakt erklärt, den die Fernbedienungen hatten als ich die Stützen getestet habe. Aber was will man auch für das Geld erwarten, das wichtigste war einfach, dass die Verbindungen in der Platine funktionieren und das tun sie.



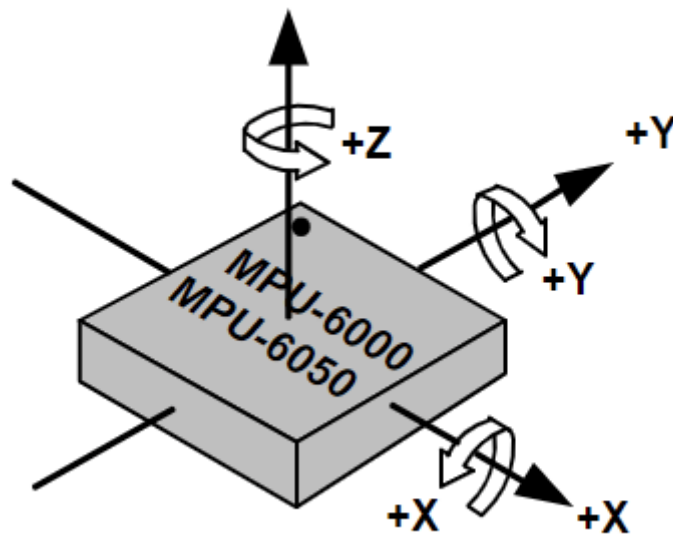
*Schaubild 1: Schematische Darstellung der Überbrückung der Taster, die eigentlichen Leitungen der Knöpfe wurden außer Acht gelassen*

Auf jeder Fernbedienung war jeweils ein „hoch“ und ein „runter“ Knopf. Ein Knopf, genauer gesagt ein Taster, macht ja nichts anderes als zwei Kontakte miteinander zu verbinden, das heißt ich kann die Taster einfach mit Relais überbrücken, welche man dann über den Mikrocontroller ansteuern kann.

Die Hauptkomponente, das Gyroskop, muss jetzt nur noch an den Arduino angeschlossen werden, damit dieser weiß, welche Relais er öffnen und schließen muss um die Schräglage auszugleichen.

# Das Gyroskop

Der Sensor den ich nutze um die Schräglage zu ermitteln ist ein Gyroskop- und Beschleunigungssensor, das heißt er gibt mir sowohl die derzeitige Beschleunigung (auch Erdbeschleunigung) in  $\text{m/s}^2$ , als auch die Rotation um die Achsen in  $\text{rad/s}$  aus. Auch wenn es erst widersprüchlich klingt, nutzen wir den Beschleunigungssensor und nicht das Gyroskop um die Schräglage zu ermitteln, da das Gyroskop die Rotation pro Sekunde misst und somit 0 ausgibt, wenn der Chip ruhig liegt, auch wenn er schräg liegt. Der Beschleunigungssensor misst allerdings die Beschleunigung entlang der Achse und nicht um die Achse, das heißt wenn der Chip unter Idealbedingungen perfekt gerade liegt misst er  $x=0$ ,  $y=0$  und  $z=9,81$ . Diese Idealbedingungen werden allerdings nie erreicht, da die Gravitation nicht überall auf der Erde gleich ist, somit können wir den z-Wert ignorieren. Wenn der Chip allerdings z.B. leicht nach links gedreht ist bekommen wir einen x-Wert  $< 0$  und wissen somit, dass der Chip auf der y-Achse nicht perfekt gerade ist. Um das verbildlichen zu können, hier eine kleine Grafik:



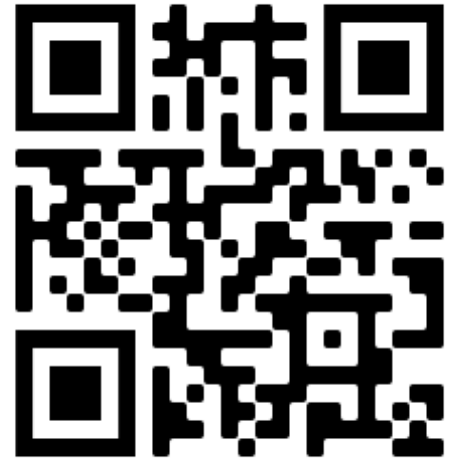
## WICHTIG:

Der Unterschied zwischen Beschleunigungssensor und Gyroskop ist, dass der Beschleunigungssensor die Beschleunigung in Richtung der x-, y- und z-Achse detektiert, wohingegen das Gyroskop die Bewegung um die Achsen detektiert. Wenn das Modul in Ruhe ist, liefert das Gyroskop für x, y und z den Wert Null. Der Beschleunigungssensor hingegen detektiert die Erdbeschleunigung auch im Ruhezustand (in z-Richtung, wenn das Modul flach liegt).

# Der Code

Um den Code möglichst übersichtlich zu gestalten, habe ich mehrere Funktionen geschrieben, die durch Befehle über die serielle Schnittstelle aufgerufen werden können. Die Eingabe der Befehle über die serielle Schnittstelle kann deshalb auch sehr einfach ausgetauscht werden z.B. durch Taster oder eine App auf dem Handy, das ganze ist also nach Belieben modifizierbar.

Da der ganze Code zu lang wäre um ihn hier zu erklären, habe ich die einzelnen Codezeilen direkt in der .ino-Datei erklärt, der auf GitHub über den QR-Code oder den Link darunter zur Verfügung gestellt wird. Ich erkläre hier nur die einzelnen Befehle und ihre Funktionen:



<https://bit.ly/3uCelc3>

Funktion	Befehl	Beschreibung
balance()	balance	Die Hauptfunktion des Projekts; fragt den aktuellen x-Wert des Gyroskops ab und steuert dementsprechend die Relais an, die für die Korrektur der Schräglage benötigt werden
pullup()	end	Zieht alle stützen nach oben um z.B. das Reisemobil abfahrtbereit zu machen
stopall()	stop	Stoppt sofort alle Stützen; schaltet alle Relais ab; kann als Notaus genutzt werden
expandall()	expand	Lässt alle stützen bis zum Maximum fahren
measuresensor()	-	Frägt den aktuellen x-Wert des Sensors ab und speichert diesen unter der globalen Variable „measuredresult“
checkserial	-	Frägt die letzte serielle Eingabe ab und speichert diese unter der globalen Variable „command“

## Ende

Ich hoffe Ihnen hat mein Projekt gefallen und meine Dokumentation war zumindest halbwegs gut erklärt. Falls Sie Interesse an meinem Prototypen-Code haben, können Sie diesen gerne in dem GitHub-Repository anschauen, in dem auch meine finale Codeversion liegt. Erreichbar ist der Code jederzeit über: <https://bit.ly/3uCelc3> oder diesen QR-Code:

