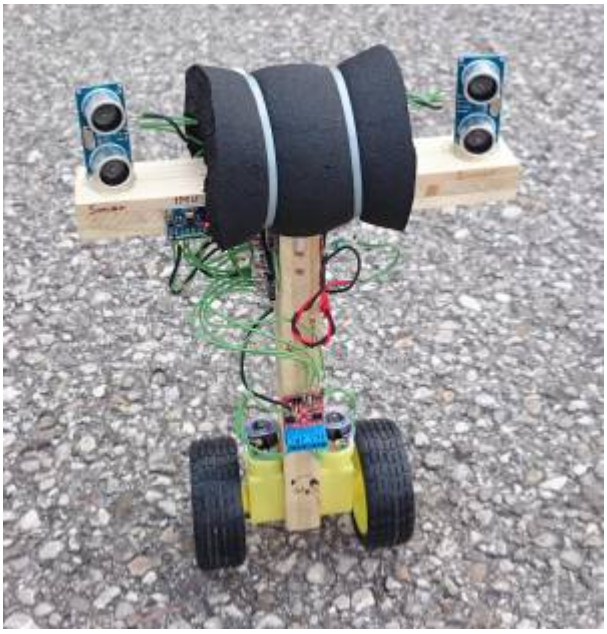
Lemming Robo



In der Kirchdorfer Talente-Woche 2015 durfte ich 20 Kindern im Alter zwischen 9 und 14 Jahren die Faszination der Robotik näher bringen. Die herausfordernde Aufgabe war, einen „Mini-Segway“ Namens Lemming-Robo nach Anleitung aufzubauen. Dabei wurde modernste Mikroelektronik verbaut und verdrahtet. Die Kinder waren mit voller Begeisterung bei der Sache und haben insgesamt sogar mehr als 8 Überstunden investiert, um endlich ihren eigenen Roboter balancieren zu sehen.

Technologie- & Innovationszentrum tiz Kirchdorf

Mechatronik Karl Zeilhofer

Videos

Hier sieht man ein wenig den Werdegang - von sehr tollpatschig bis ziemlich stabil

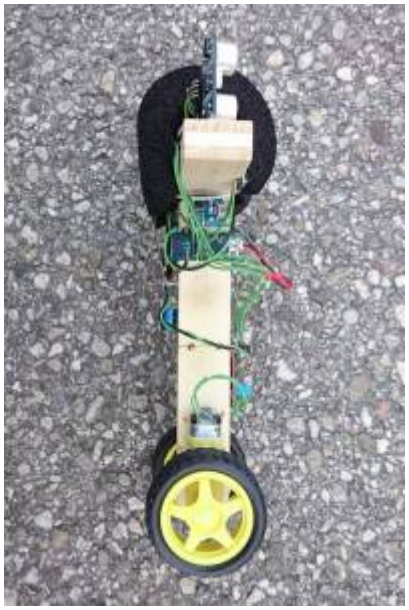
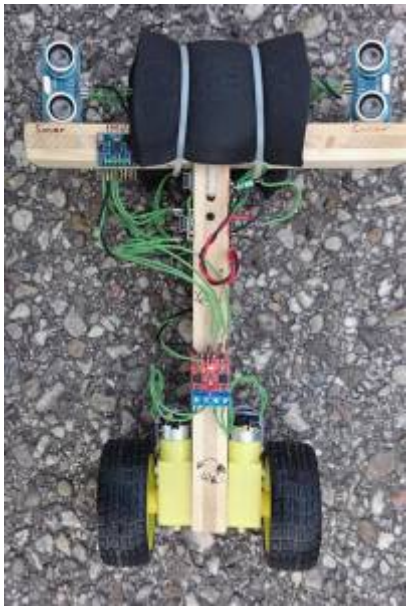
1. Erste Fahrversuche mit PD-Regler
2. Mit Hochachsenregler, d.h. er hält die Fartrichtung bei
3. Verbesserte Reglerstruktur (PDD²) zur Bedämpfung des Schwingens um die Sollposition

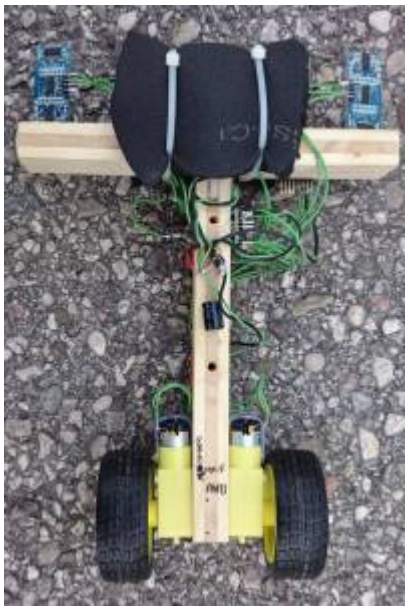
Komponenten des Robos

Holzrahmen

Der Grundkörper des Robos besteht aus 2 Stückchen 19mm 3-Schichtplatte mit je 200x40mm. Das vertikale Holz muss gebohrt werden, sodass die Befestigungsschrauben der Motoren passen, und dass

die zur Mitte gerichteten Antriebsstummel, die am Getriebe beitseitig herausgeführt sind, Platz haben.



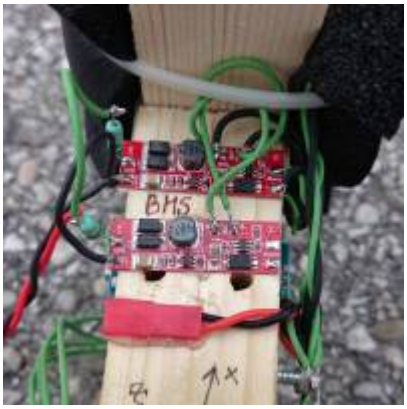
Lithium-Batterie

Als Energiequelle bzw. -speicher dienen gebrauchte 18650er-Zellen aus Notebookakkus. Davon sind 2 Stück parallel geschaltet. Deren Nennspannung ist üblicherweise 3.7V. 4.2V ist die Ladeschlussspannung und entladen werden kann die Batterie bis ca. 3V. Tiefentladung muss unbedingt verhindert werden. Hat die Zelle über längere Zeit (einige Tage) eine Spannung von weniger als 2.7V, so bilden sich metallische Kristalle. Diese führen dann zu einer Art internem Kurzschluss, wenn die Zelle wieder auf 4.2V geladen wird. Das artet dann unter Umständen in einem heftigen Feuerwerk aus.

Laderegler

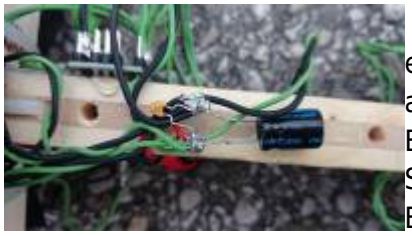
Der Laderegler sorgt für ein kontrolliertes Laden der Batterie. Hier wird ein USB-Kabel vom Typ Mini-B angeschlossen.

Batterie-Management-System (BMS)



Ein BMS sorgt dafür, dass die Batterie in ihren Betriebsgrenzen betrieben wird. Es hat Anschlüsse für den Laderegler (Charge), die Batterie (Batt) und die Last (+5V). Es schützt die Batterie vor Überladung und auch Tiefentladung.
Das besondere an den hier verwendeten Modulen ist, dass sie zusätzlich gleich auch noch einen Hochsetzsteller integriert haben, der aus der variablen Batteriespannung von 3-4.2V konstante 5V erzeugt. Ein elektronischer Hochsetzsteller ist vom Prinzip her dem mechanischen Wasserwidder sehr ähnlich. Weil die Motoren so viel Strom brauchen, werden zwei Module parallel geschaltet. Am Ausgang werden zwei 0.1Ω-Widerstände verwendet, sodass sich der Strom zwischen den beiden Modulen gleichmäßig aufteilt. Stellt das BMS einen Kurzschluss fest, wird der Ausgang deaktiviert. Eingeschaltet kann er wieder werden indem man entweder die Batterie kurz abklemmt, oder man den USB-Laderegler mit einem Kabel an eine USB-Buchse ansteckt.

Stromverteiler



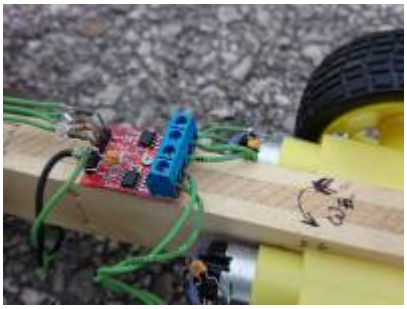
Die Stromverteilung zu all den verschiedenen Platinen erfolgt über einfache Schrauben, die als Lötstützen verwendet werden. Hier sind auch zwei Kondensatoren angebracht. Einerseits ein Aluminium-Elektrolyt-Kondensator zur Stützung der Spannungsversorgung bei Stromspitzen und außerdem noch ein kleiner Keramikkondensator zur Entstörung.

Antrieb

Den Antrieb übernehmen zwei Getriebemotoren, mit einer Untersetzung von 48:1. Nominell können sie mit 6V betrieben werden, hier werden sie mit 5V angesteuert. Der Elektromotor muss vor dem verbauen aus dem Getriebegehäuse genommen werden, und umgedreht werden. Dadurch schauen die elektrischen Kontakte dann zum Reifen, und sind daher leicht zugänglich. Dies ist notwendig, weil ansonsten der Reifen auf jene Halb-Antriebswelle gesteckt werden würde, die mit der anderen Hälfte nur durch eine Steckverbindung gefügt ist. Nach längerem Betrieb lockert sich diese Verbindung, und rutscht durch. Die andere Hälfte bildet ein Spritzgussteil mit dem Zahnrad im Inneren des Getriebes, ein durchrutschen ist daher ausgeschlossen.
Es empfiehlt sich, die Reifen auf der Getriebewelle mit einem Tupfen Superkleber zu fixieren.

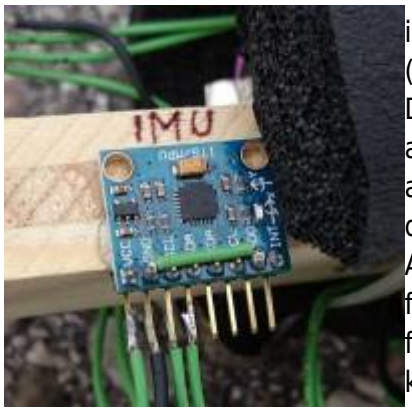
Beim Prototypen hat sich herausgestellt, dass die Kombination aus den Hochsetzstellern und den Motoren starke elektromagnetische Störungen verursacht, sodass sich der Mikrocontroller aufhängt, und nur mehr durch einen Reset wieder zur Vernunft gebracht werden kann. Als Abhilfe wurden mehrere Maßnahmen getroffen. Unter anderem wurden Entstörkondensatoren direkt an den Gleichstrommotoren angebracht, je zwei Stück pro Motor.

Motortreiber



Um die Motoren in beide Richtungen drehen lassen zu können, wird ein Doppel-H-Brücken-Modul verbaut. Es wird einerseits direkt mit 5V versorgt, und andererseits hat es 4 Eingangspins, mit denen jede der 4 Halbbrücken angesteuert wird. Der Arduino steuert diese mit einer Pulsweitenmodulation mit einer Frequenz von 1kHz an. Dadurch können die Motoren nicht nur schnell nach vorne und schnell nach hinten, sondern quasi stufenlos jede Geschwindigkeit fahren. Auch hier wurde eine Entstörmaßnahme verbaut: Die Ansteuerleitungen haben je einen Reihenwiderstand von 1kOhm integriert. Die H-Brücken sind nicht gerade besonders Rückwirkungsfrei, d.h. energiereiche Störungen von der Motorseite gelangen zurück auf die Ansteuerpins.

Beschleunigungs- und Drehratensensor

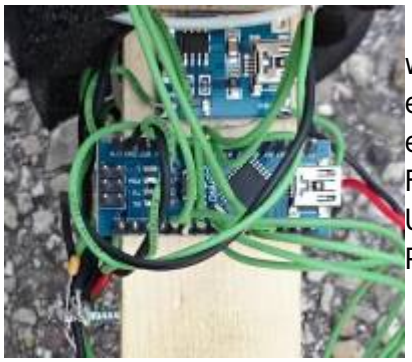


Dieses Sensormodul hat in einem Mikrochip gesamt 6 Sensoren integriert. Es kann in je 3 Achsen die Beschleunigung (Veränderungsrate der Geschwindigkeit, F=m*a oder a=dv/dt) und die Drehgeschwindigkeit messen. Es wird benötigt, um einerseits den aktuellen Kippwinkel zu berechnen, und andererseits um den Roboter auch geradeausfahren lassen zu können. Denn lenken tut der Robo ja durch unterschiedliche Drehgeschwindigkeiten der beiden Antriebsräder. Die laufen aber von hausaus schon ungleich, was dazu führt, dass der Robo ständig leichte Kurven fährt, wenn er geradeaus fahren soll. Ein einfacher Regelalgorithmus sorgt dafür, dass dies kompensiert wird, sodass die Drehgeschwindigkeit um die Hochachse ca. Null ist. 100 mal pro Sekunde werden die Messwerte erfasst und in der Software dann so weiterverarbeitet, dass der Robo aufrecht stehen bleibt, und auf Lenkbefehle richtig reagiert.

Ultraschallsensoren

Diese Sensoren haben einen Lautsprecher und ein Mikrophon. Es wird ein Piepton, den man nicht hören kann, weil dessen Frquenz höher ist, als das menschliche Ohr wahrnehmen kann, ausgesendet, und nach einer gewissen Zeit wieder am Mikrophon empfangen. Durch die Schallgeschwindigkeit in der Luft kann man daraus dann die Entfernung des reflektierenden Gegenstandes berechnen. In dieser Anwendung reagiert der Robo z.B. auf eine vorgehaltene Hand auf einer Seite mit einer Drehbewegung, und kann somit gelenkt werden.

Arduino Nano



Als Herzstück, das all die Berechnungen und Steuerungen übernimmt, wird ein Arduino Nano verwendet. Man kann sich vorstellen, dass das ein ganzer Computer in klein ist. Er hat Ein- und Ausgabefähigkeiten, eine CPU (16MHz), Arbeitsspeicher (RAM, 2kByte) und quasi eine Festplatte (Flash-Speicher, 32kByte). Programmiert wird er über die USB-Schnittstelle. Auf dem PC braucht man das Arduino-Studio, um das Programm runterspielen zu können.

Der Zusammenbau

Foto-Galerie für den Zusammenbau (V3.3)

Werkzeugliste

Seitenschneider	für Draht, Widerstände, Kondensatoren, Schrumpfschlauch
Abisolierzange	es sind viele Drahtstücke zu verlöten

Kombizange	ist zusammen mit einem starken Gummiring eine dritte Hand
Lötkolben	min. 20 Watt, Meißelspitze mit 2-3mm Breite
Lötkolbenständer	am besten fixierbar am Tisch und mit Lötschwamm
Schraubendreher, PH1	zum fixieren der Motoren am Holz
Schraubendreher, flach 3mm	für die Klemmen am Motortreiber
Heißklebepistole	zum fixieren der Module auf dem Holz
Messer	zum Aufschneiden der Rohrisolierung
Ständerbohrmaschine	für das vertikale Holzstück für die Motoraufnahmen
Bohrer	2.5mm für M3-Schrauben und einen 7.5mm für Auslässe
Heißluft	200°C für Schrumpfschläuche
Computer	mit Arduino IDE für die Programmierung

Verbrauchsmaterialien

Heißklebesticks	10mm, ca. eine viertel Stange
Lötzinn	1mm
Isolierband	evt. für die 0.1Ω-Widerstände
Kabelbinder	2x 350mm, 2x 200mm
Drähte	Radox 125, Litzendraht 0.25mm² grün, 0.5mm² schwarz Radox 125 ist ein recht temperaturbeständiger Draht, sodass die Isolierung nicht schmilzt beim verlöten

Downloads

Schaltplan

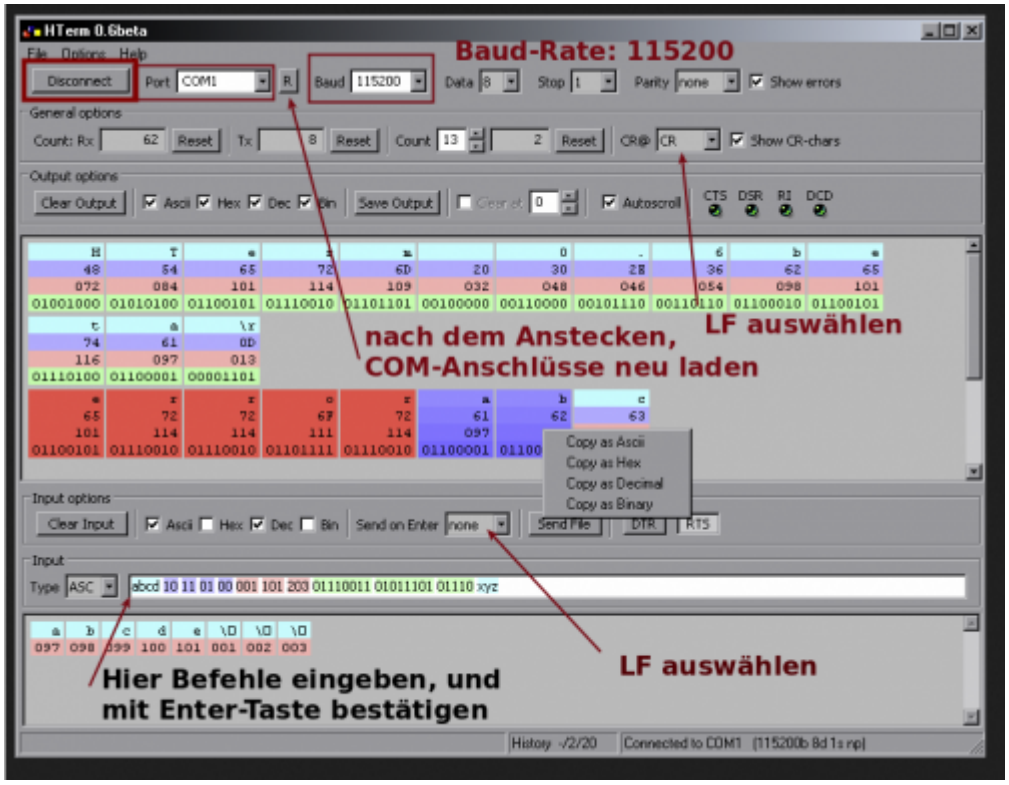
V3.0 V3.1 V3.3

Quellcode

NEU: die Firmware ist nun auf GitLab zu finden: https://gitlab.com/KarlZeilhofer/LemmingRobo
alt: Arduino Quellcode
Zum programmieren braucht man nur noch die kostenlose Entwicklungsumgebung Arduino IDE

Paramterjustierung

Der Lemming Robo speichert seine Parameter im EEPROM des Arduinos ab.
Diese können über ein Terminal-Programm (Text-basierte Ein-/Ausgabe) verändert werden.
Für Windows kann ich hterm sehr empfehlen.



1. Stromversorgung des Robos einstecken.
2. USB-Kabel an Lemming-Robo (am Arduino, nicht am Laderegler) und PC anstecken
3. hterm starten
4. „richtigen“ COM-Port auswählen, das ist evt. eine Probiererei
   hier muss man evt. auch auf **R** klicken, um die Leste der COM-Ports zu aktualisieren.
5. Baudrate auf 115200 einstellen
6. LF (line feed) oben und unten wie im Screenshot zu sehen auswählen.
7. Auf „Connect“ links oben klicken.

Nun können Befehle gesendet werden.

- **L**
  Listet alle Parameter auf. Parameter die mit einem Unterstrich beginnen, können nich verändert werden, und dienen nur der Information. Hier eine beispielhafte Ausgabe:

```
[<] dtMin=9.95e-3 s
[r] phiNull=175.00 deg
[t] _phiSteuer=6.25 deg
[z] phiLimit=8.00 deg
[u] _phiKalman=92.29 deg
[i] _phiFehler=-55.70 deg
[q] phiReg_P=285.16e-3 pwm/deg
[w] phiReg_I=3.84e-6 pwm/deg.s
[e] phiReg_D=937.51e-3 pwm/ deg/s
[o] Qangle=999.46e-6
[p] Qbias=2990.76e-6
[f] Rmeasure=29.91e-3
[g] _omegaKalman=-100.91e-3 deg/s
[h] _pwmMotor=-1000.00e-3 pwm
[j] pwmFilter_ta=3.13 s
[k] kMotor=636.72e-3 m/s /pwm
```

```
[l] _vMotor=-455.25e-3 m/s
[y] _pos=-1577.41e-3 m
[x] _posSoll=0.00e-9 m
[c] _posFehler=1577.41e-3 m
[a] posReg_P=1312.53e-3 deg/m
[s] posReg_I=500.00e-3 deg/m.s
[d] posReg_D=6.19 deg/ m/s
[D] posReg_DD=1464.10e-3 deg/ (m/s)
[v] motorEin=1000.00e-3 binaer
[b] _phiBeschl=92.57 deg
[n] _omegaGyro=-126.11e-3 deg/s
[m] _phiGyro=92.27 deg
[#] _phiSelekt=107.95 deg
```

- Mit den Buchstaben in eckigen Klammern kann man den jeweiligen Parameter auswählen. Mit **r** kann man z.B. phiNull auswählen.
- nun kann man mit **+** oder **-** den Wert des aktuell ausgewählten Parameters verändern. In unserem Beispiel stellt sich nun ein neuer Sollwinkel des Robos ein. Der Winkel wird in 0.2° Schritten verändert. Man kann auch z.B. 5 Plus-Zeichen in einer Zeile eingeben, und erst dann mit Enter bestätigen.
- Passen die Parameter, müssen diese mit dem Befehl **S** gespeichert werden.
- Nun USB-Kabel trennen, und den Robo resetten.