Arduino Segway

by stoppi71

Hello!

More than 3 years ago I started to tinker a homemade segway. Though it's still not really finished (I have to replace the 250W Motors by 500W-models) my children are able to drive around.

If you're going to build one you will Need the following parts:

* wheelset: In my case I use 12 1/2 x 2.75 tires usually taken for dirt bikes (f.e. www.amazon.de/HMParts-REIFEN-MIT-SCHLAUCH-12/dp/B0038X9ODA;

href="http://www.ebay.com/itm/Tire-Rear-Wheel-Mini-Pocket-Dirt-Bike-47cc-49cc-50cc-COOLSTER-QG-50-RX1-Traxxis-/182346539856?hash=item2a74b36350:g:BxUAAOSwh6xTtmwX&vxp=mtr">http://www.ebay.com/itm/Tire-Rear-Wheel-Mini-Pocket-Dirt-Bike-47cc-49cc-50cc-COOLSTER-QG-50-RX1-Traxxis-/182346539856?hash=item2a74b36350:g:BxUAAOSwh6xTtmwX&vxp=mtr)

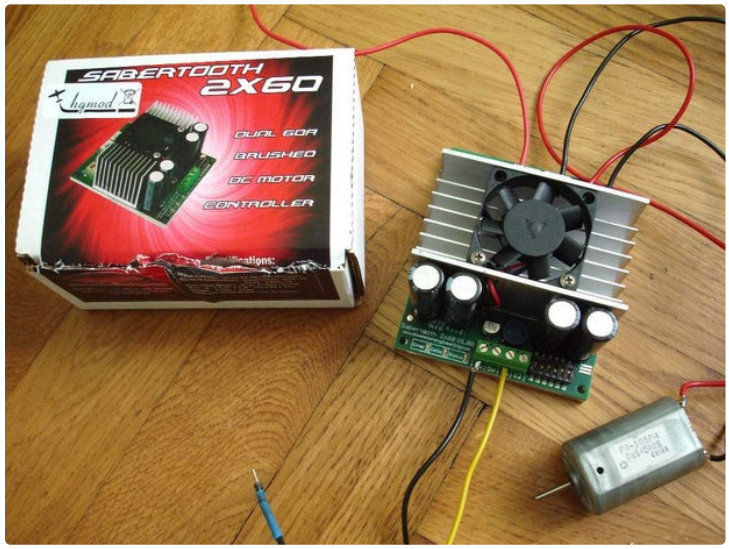
* batteries: I use two 12V/12Ah lead accumulators to get 24V for the Motors

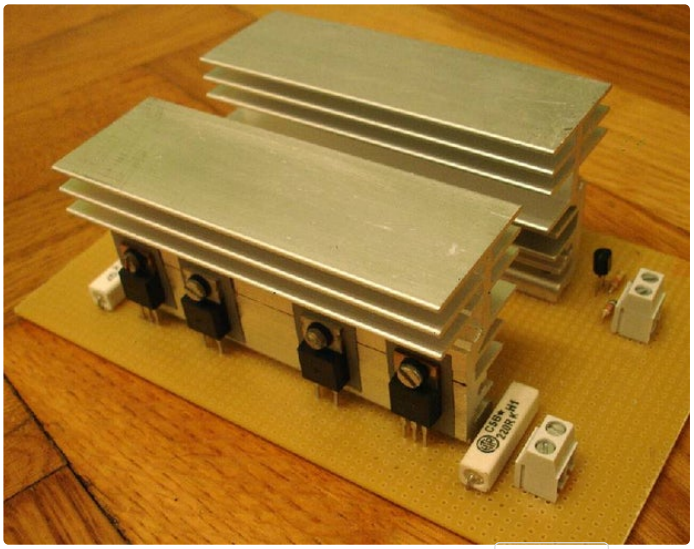
* Motors: At the moment two 24V/250W models but the

Step 1: The Motor Driver

First I tried a homemade full-bridge. To control it I used the analog-out (pwm) of the arduino. But with this variant I didn't succeed. Therefore I bought a commercial product, a 2x60A Motor Driver from sabertooth (www.dimensionengineering.com/products/sabertooth2x60).

The arduino is able to communicate with the sabertooth via a simplified serial code.





Download

https://www.instructables.com/ORIG/FLA/UBAB/IW6PNEUD/FLAUBABIW6PNEUD.pdf

Download

https://www.instructables.com/ORIG/FBQ/AVM8/IW6PNEUA/FBQAVM8IW6PNEUA.zip

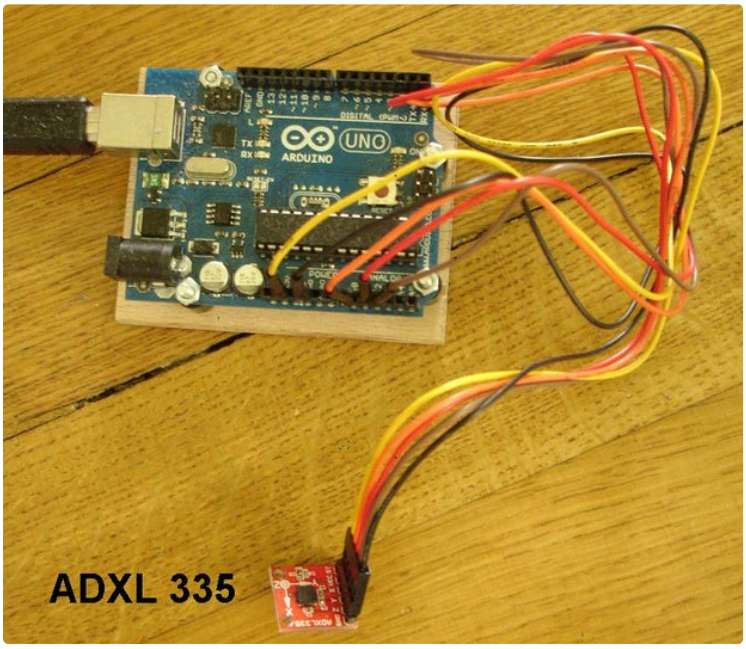
Step 2: The Angle Determination

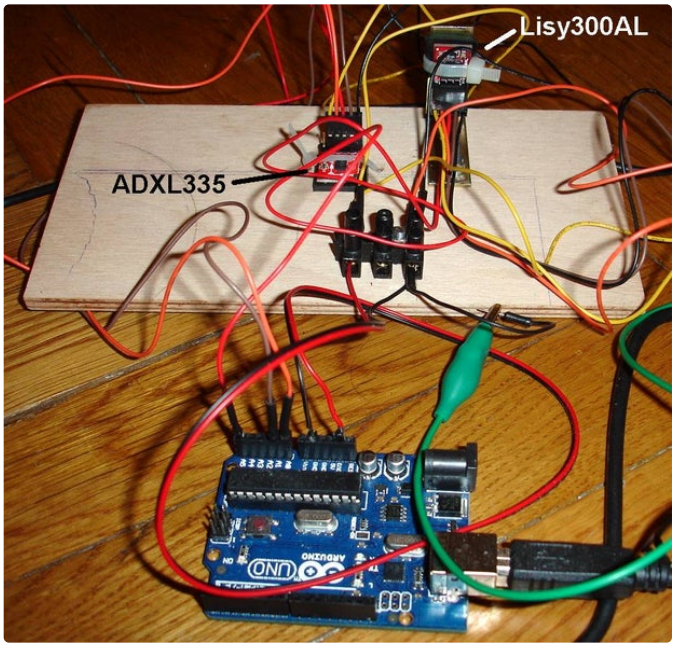
To determinate the current angle I first used a separate accelerometer (ADXL335) and gyroscope (LY530AL). Last week I changed this Setup and the two boards have been replaced by the MPU6050.sensor, which combines the two elements.

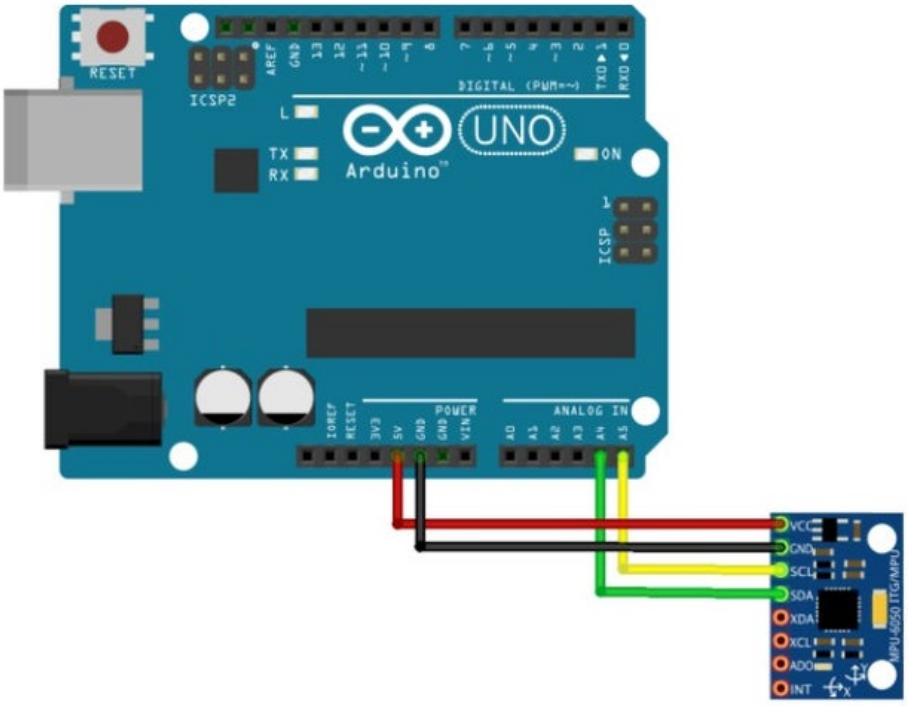
The advantage of the gyroscope is, that accelerations don't influence the measurements. The disadvantage is, that the gyro has a drift and after a while the angle goes higher and higher.

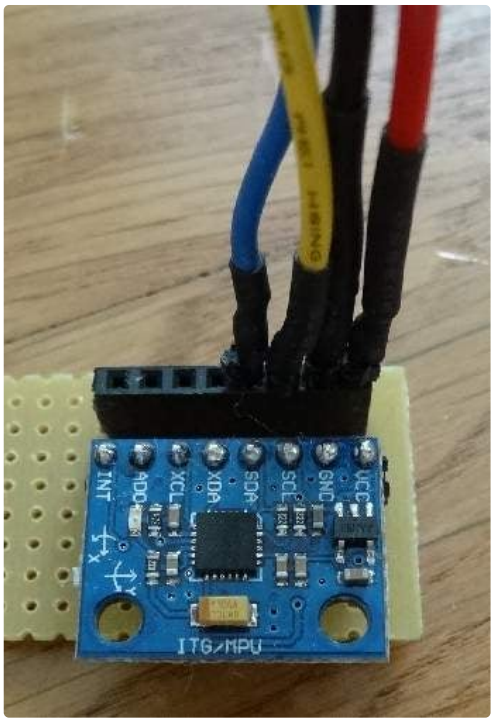
The Advantage of the accelerometer is, that it doesn't show any drift like the gyro. But it's being influenced by accelerations, which change the Output.

A filter combines both pro's so the calculated angle is near the real value. This filter can be a simple complementary-filter (like in the Picture) or the more complicated kalman-filter. The Input of the kalman-filter is the actual angle measured by the accelerometer, the angular Speed measured by the gyroscope and the time-step between two measurements. With these Parameters the kalman-filter is able to calculate an angle, which follows the gyro in times of high accelerations and in times of less accelerations it approaches the value of the accelerometer. In the Picture you can see the drift of the gyro and the good Job of the filter.









Komplementärfilter

$$\varphi_{neu} = (1-k)\cdot(\varphi_{alt} + \omega_{Gyro}\cdot\Delta t) + k\cdot\varphi_{Acc}$$

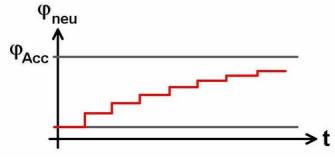
Ergebnis: φ_{neu} **nähert sich** φ_{Acc} **an**

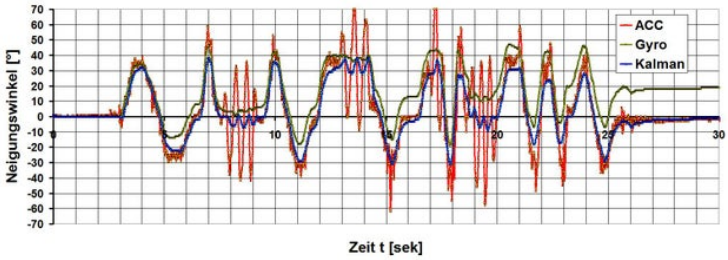
Angenommen $\omega_{Gyro} = 0$**:**

$$\varphi_{neu} = (1-k)\cdot(\varphi_{alt} + \omega_{Gyro}\cdot\Delta t) + k\cdot\varphi_{Acc}$$

$$\varphi_{neu} = (1-k)\cdot(\varphi_{alt}) + k\cdot\varphi_{Acc}$$

unterschiedlich gewichtete Mittelwertsbildung!





Step 3: The PID-controller

The segway tries to get the angle down to 0°. If the angle is positive, the segway brakes or accelerates backwards. If it's negative the segway accelerates forward.

But which signal get the motors? This is exactly the task of the pid-Controller.

The **p**roportional-part: The angle should be 0°. So the greater the angle, the higher the speed of the motors.

Mathematically you can write:

angle_error = angle_should - angle_is

p-Motor-value = k_p * angle_error

The **i**ntegral-part: When the current angle isn't zero, the Controller tries to get the segway horizontally again. But this approach shouldn't be slow and the smaller the angle-values, the slower the changes. Therefore the Controller allows a faster approach with the disadvantage, that the movement doesn't stop at 0° and goes a bit to the other side. This behaviour is guaranteed by the integral-part. All the angles are added (integrated) and this integral over time should be Zero. So if the angle is f.e. positive for a while, the Controller tries to get the seqway into the negative area. This guaranteeds faster approaches...

Mathematically you can write:

error_sum_new = error_sum_old + error

i-motor-value = k_i * (error_sum_new)

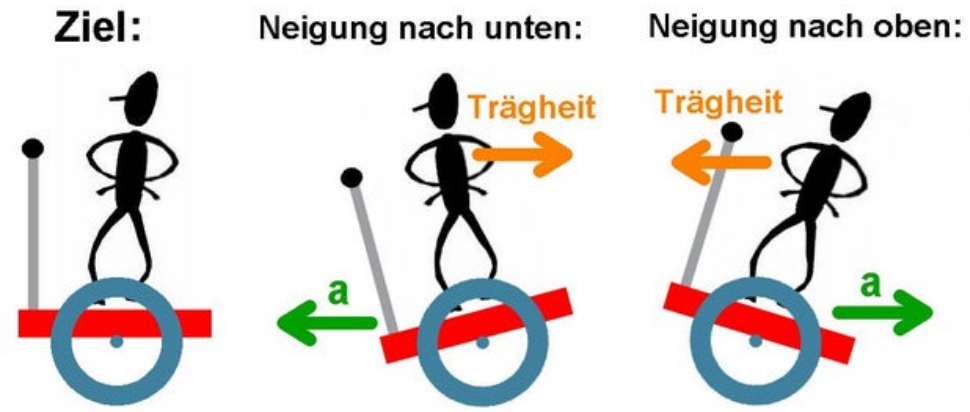
error_sum_old = error_sum_new

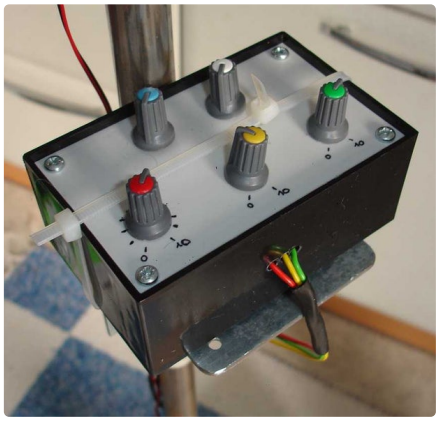
The **d**erivative-part: Not just the current angle is important but also the changes of the angle. Let me explain: If the angle is f.e. 10° and it doesn't Change a lot, the Motors don't need to correct this shift very fast. But if the angle is 10° and it changes very fast (high derivative), than the motors have to correct this also very fast.

Mathematically you can write:

d-motor-value = k_d * angular Speed

Finally you get: motor-value = p-motor-value + i-motor-value + d-motor-value





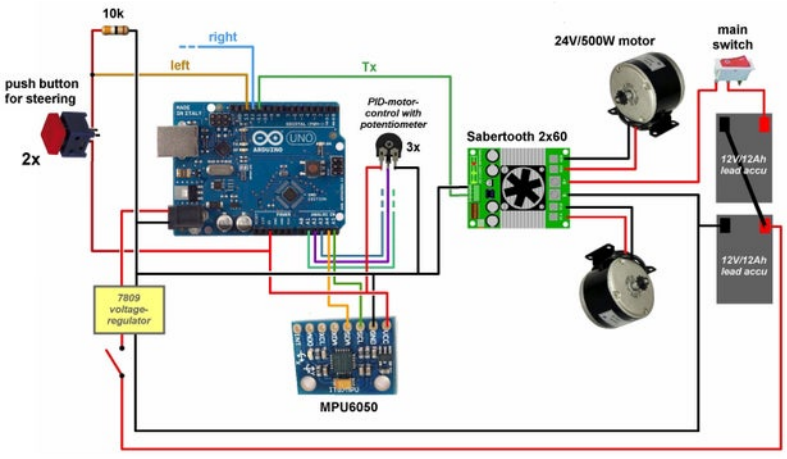
Step 4: The Complete Setup

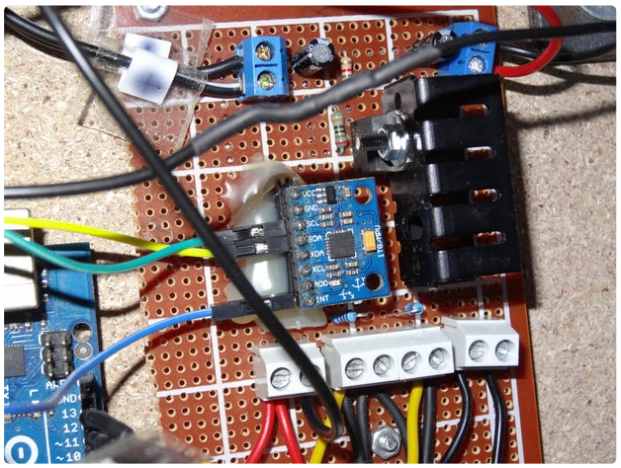
Now it's time to start the segway. To drive not only straight ahead you have to mount two buttons. When you push f.e. the right one, the left Motor increases Speed and the right Motor slows down. To ensure smooth transitions the added and deducted value starts at zero and increases step by step (+0.5 every loop) until he reaches the maximum value (f.e. 30).

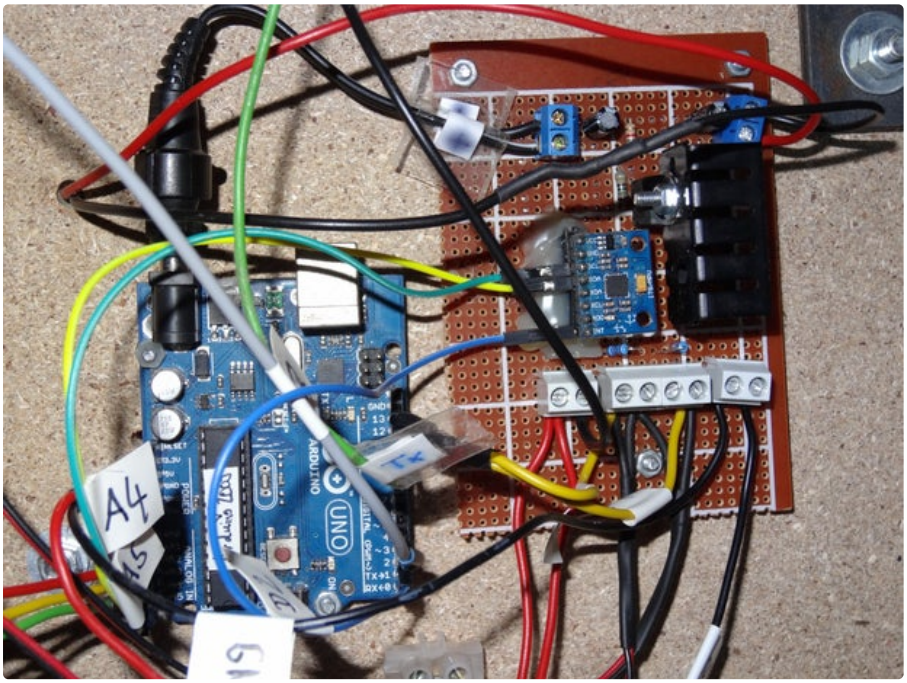
To try out your own segway I 've attached the simple program for you. Good luck and don't follow the segway-inventor James Heselden down a cliff ;-)

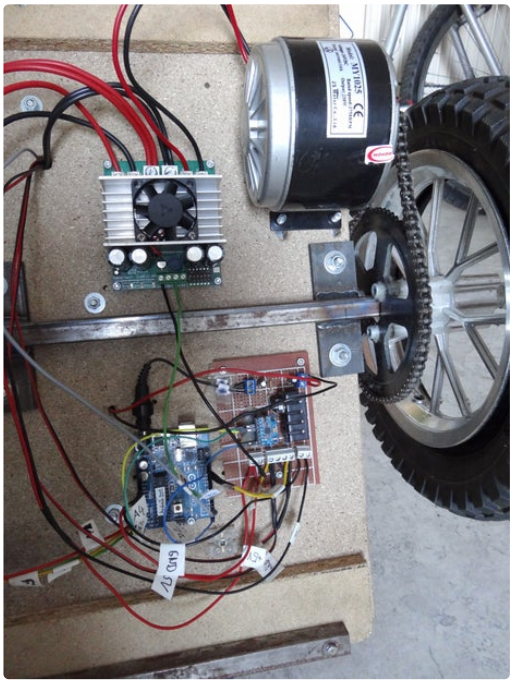
Maybe you want to take a look at my YouTube-channel: www.youtube.com/user/stopperl16/Videos

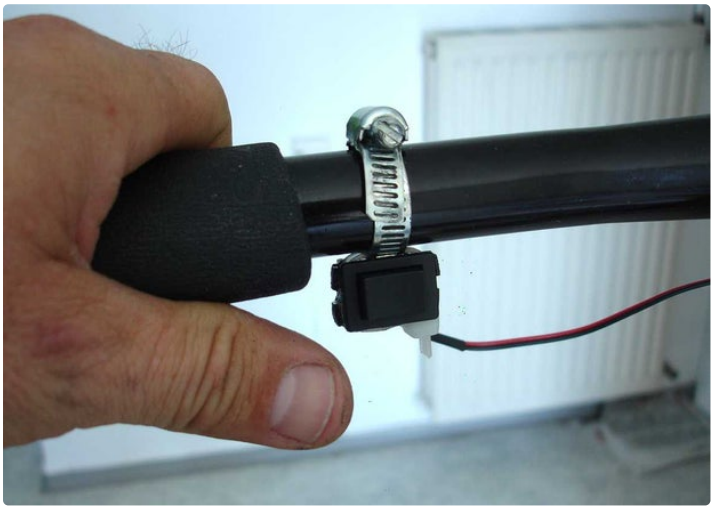
more physics projects: https://stoppi-homemade-physics.de/











https://www.youtube.com/watch?v=JhQ4s2_pG1o

Download

https://www.instructables.com/ORIG/FIG/W4SU/KJYAJMV6/FIGW4SUKJYAJMV6.ino

Yes, you'll need a second button (voltage-divider with the 10 kOhm-resistor)... Pin 12 and 13 are low, if the button isn't pushed. If you push the button for steering, the signal goes high...

Therefore I ask in the program: if (steering-button-right or left) = high then ...

I've changed the Picture of the circuit, because in the first version the status was low if the button was pressed ;-)

can u explain briefly about it. Also can we use potentiometer for steering the direction?

Can i get full pdf of segways construction

Can i use another motor driver? i would like to use your project as reference in my project but the driver that you used is not available in our country and the cost was too high heheh... can you give me some advice about the motor driver? thanks ^_^

thank you for sharing, you have making the best project for me ..you're actually a professional engineer.!

very good, am making a robot with that code, and its working!

Way too advanced for me but i know some people in stockholm that made one at maker space. Was very impressed!

Cool project and nicely documented!

How much weight does the current setup (250W motors) carry around?
and do you have some kind of encoders or odometry for the wheels?

You definately have my vote.

Thanks for this :)

I've to thank ;-)

My question, too: why did the Arduino PWM O/P and a bridge cct not work? What does the commercial controller do that a basic DIY set-up not do?

Take a look at my YouTube-Video "segway - first Version" ;-)

Hi. What problems did you encounter with Arduino PWM control and H-bridge?

Hi!

Take a look at my YouTube-Video: Segway-first Version...

Then you'll clearly see the Problems I had with the first Setup ;-)

Do you have a cost estimate? What are its safety features? I'm a senior, hoping to move to a walkable community where I still might have a need for speed. This looks better than an electric bike. Could a double kick-stand be added to the front edge of the platform to make it stand upright while parked, if it doesn't do that already?

Concerning the costs:

* Motors 100 Euro

* Sabertooth motor driver 150 Euro

* complete wheelset: 120 Euro

* chain, pinion: 30 Euro

* arduino, MPU6050 sensor and other electronics: 15 Euro

* lead batteries (used): 30 Euro

The safety Features are, that in the program the maximum-angle is limited with 30 degrees. But I don't have a redundant-System like in the original segways. So it's just a bit for stunt-Drivers ;-)

If you let the segway alone, it keeps standing horizontally. If you incline it Forwards or backwards the Motors will stop.

Did you already try the 500W motors ? I think that maybe your 250W motors aren't driven properly, or maybe they are old and have lost some power. 500W is about 0.7 hp, which is one fourth of the average power of a 50cm3 gas engine. Such an engine can carry a load of 200kg at a speed of 100km/h (62 mph) !

It s because their children are in the nitro circus team. They dont have enough power to pass the triple backflip yet ;)

lol

I could be wrong, but it seems to me that even a couple of 50W motors would be more than sufficient for this application.

I don't think so. 100 W is 1/7 horse power and you need a lot of power, especially at low angles. But that is difficult to manage, maybe with a higher threshold, from which the Motors start at low angles...

To be fair an actual Segway has 1.5KW per side if I'm reading correctly!? "Maximum power is 2 horsepower (1.5 kW) per servo motor." Thats wikipedia not the manuf, so could be wrong - 3000w seems a little excessive....

As I mentioned above the Problem is the low torque and power for tiny angles. This can be managed with a threshold and/or more powerful Motors. As you can see ín my Video the segway is using the 2 x 250W Motors a bit shaky. I'll try both improvements and maybe I fasten the Loop (at the Moment 100 Hz)...

...and servo Motors have much higher torque at lower Speed than brush-motors at low voltage. That's a huge Problem....

A big problem with specifying the motors is the starting loads are so much greater than the running load. Worse, the motors are starting, stopping and changing speed every few seconds while adjusting the direction to stay on course. The torque output will be changing constantly.

Having motors with plenty of capacity may help their lifespan.

I totally agree. I'll try to improve the code and use a threshold so the Motors don't start at 0 if the angle is very low. This helps to increase the motor-power for tiny angles...

I've finished my segway with the 2 x 500W Motors not yet, but I'm trying to finish it soon (too many other Projects ;-)).

I don't thin that Motors get really old and loose power. Maybe my batteries (they were used when I bought them) were a bit weak.

I'll try to improve the code too. At the Moment the measure-frequency is 100 Hz. I'll take a look at the loop-velocity and maybe I can reduce the delay to get a faster Loop.

Yeah !

The Sabertooth provides 5V @ 1A from a switching regulator. Is there a reason you used the 7809 regulator for the Arduino and sensors? I am using a Sabertooth 2X25A for a 50kg self balancing robot and finding it quite erratic. Maybe I have power supply problems?

Hi!

I'm using the 7809-regulator because I like to have a bit voltage-reserves...

Two times 25A can be enough but it depends on the Motors you use ;-)

I am a little confused about the buttons for steering. Only one button is shown in the complete setup diagram but a second blue wire is there for right turns. Would that wire connect to a second switch in the same way as the left? Also, as the diagram is drawn, it appears that the 5V pin and pin 13 are always connected whether the switch is pressed or not? Is that correct?