

After some time, I was not really happy with the force quality of the [Logitech Wingman Force Feedback 3D joystick](#) that I hacked. Basically the motor force increase at increasing airspeed is step wise, not gradually.

After reading on [FSforce forum](#) that one of the best force feedback sticks for Flightsimulator is the Microsoft Sidewinder Forcefeedback-2 stick, I decided to see if it was possible to hack this stick's hardware in similar way as I had done with Logitech Wingman Force Feedback 3D joystick

It turned out that it is quite doable, and it is actually easier to boost the stick's motor current compared to the Logitech Wingman stick.

Below is a description how to hack the Microsoft Sidewinder Forcefeedback-2 stick and modify for higher current.

Warning: This stick internals contains high voltages when connected to mains voltage.



Bought on eBay for €50,- Bottom side, and with bottom removed.
(The special screws need to be slowly turned to remove)



Remove AC/DC board. then disconnect all connectors from main PCB and unscrew main PCB

In order to trace the other electronics, I also removed the boards in the handle.



The middle picture shows the sensor that checks for presence of your hand on the stick. It is an IR LED and receiver. When IR signal is seen by receiver,

the FF power section is switched off. The IR light is blocked when your hand is on the stick, and then power section is activated.



After removing all boards, I hooked everything up separately on my desk to do some testing.

The AC/DC supply delivers 24Vdc to the main PCB. The CPU receives the USB signals, checks the position of the X, Y, rudder and Throttle potmeters, 12-switches, and controls the motor current.

The motor current control and power stages are all discrete. Motor current is around $\pm 0.7A$ max, with 8.5 ohm motors.

This stick's hardware is actually quite well done: The circuits really control the current through the motors instead of just adjusting the PWM duty-cycle.

Therefore, the motor current is precisely controlled, and not dependent on the motor internal resistance. This architecture also makes it easier to boost the current with exact ratio.

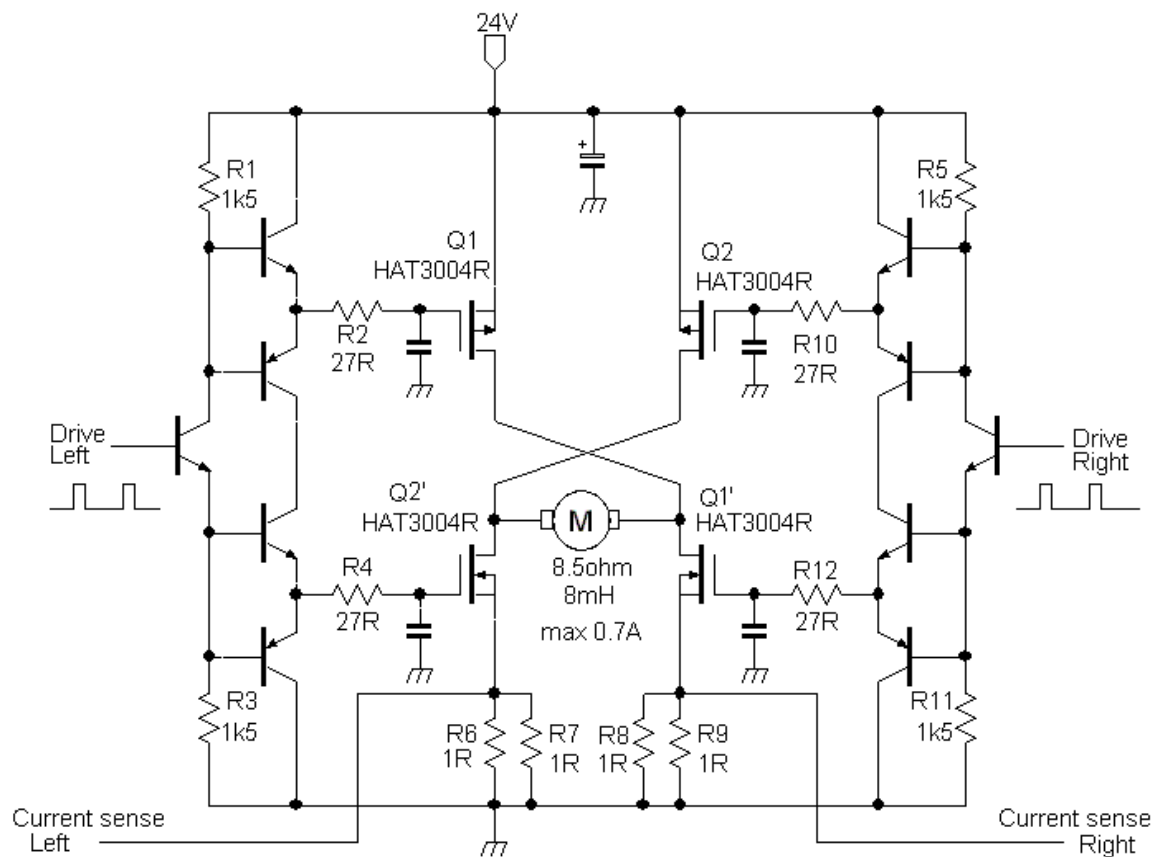
Below some hi-res pictures of the main PCB electronics. Looks a bit daunting, but for the hack we only need to modify a small part.



Below is the schematic of one channel. It is an H-bridge with P-MOSFETs and N-MOSFETs.

(each HAT3004R contains a P/N MOSFET pair)

The drive has push-pull stages for each MOSFET and one transistor that can switch-on both P- and N-MOSFETs together. (Note the crossed H-bridge connection, so no short will occur)



In this type of drive with all MOSFETs switching, shoot-through (conduction from 24V - GND via 2 MOSFETs) can easily occur if there is a small delay in one of the MOSFET drives. The designer has therefore chosen to have only one side of the bridge active at the time for one motor current direction, the other side will be fully switched off, and the inductive current will flow through the other side MOSFET's body diodes. Drawback of this method is that the power dissipation in the non-active MOSFET's is actually quite high, due to the higher voltage drop across the body diodes.

You can also see that there are current sense resistors in each side of the bridge. The voltages across the current sense resistors are processed with OPAMPs and serve as the feedback of the force current loop. By reducing the current sense resistors R7 ~ R9, the system will increase the voltage across the motors to keep the same feedback voltage, thus the current through the motors will increase. So 2x higher motor current, decrease the current sense resistors to half their original value.

But the higher current will increase the power dissipation in the MOSFETs. Above 1.4A, MOSFETs get really hot. I wanted to increase current 4x, which required using better MOSFETs and adding external Schottky diodes. I also improved the supply filtering, to avoid noise coupling into sensitive circuits.

Motor considerations: When current increases, the voltage drop across the motors will also increase.

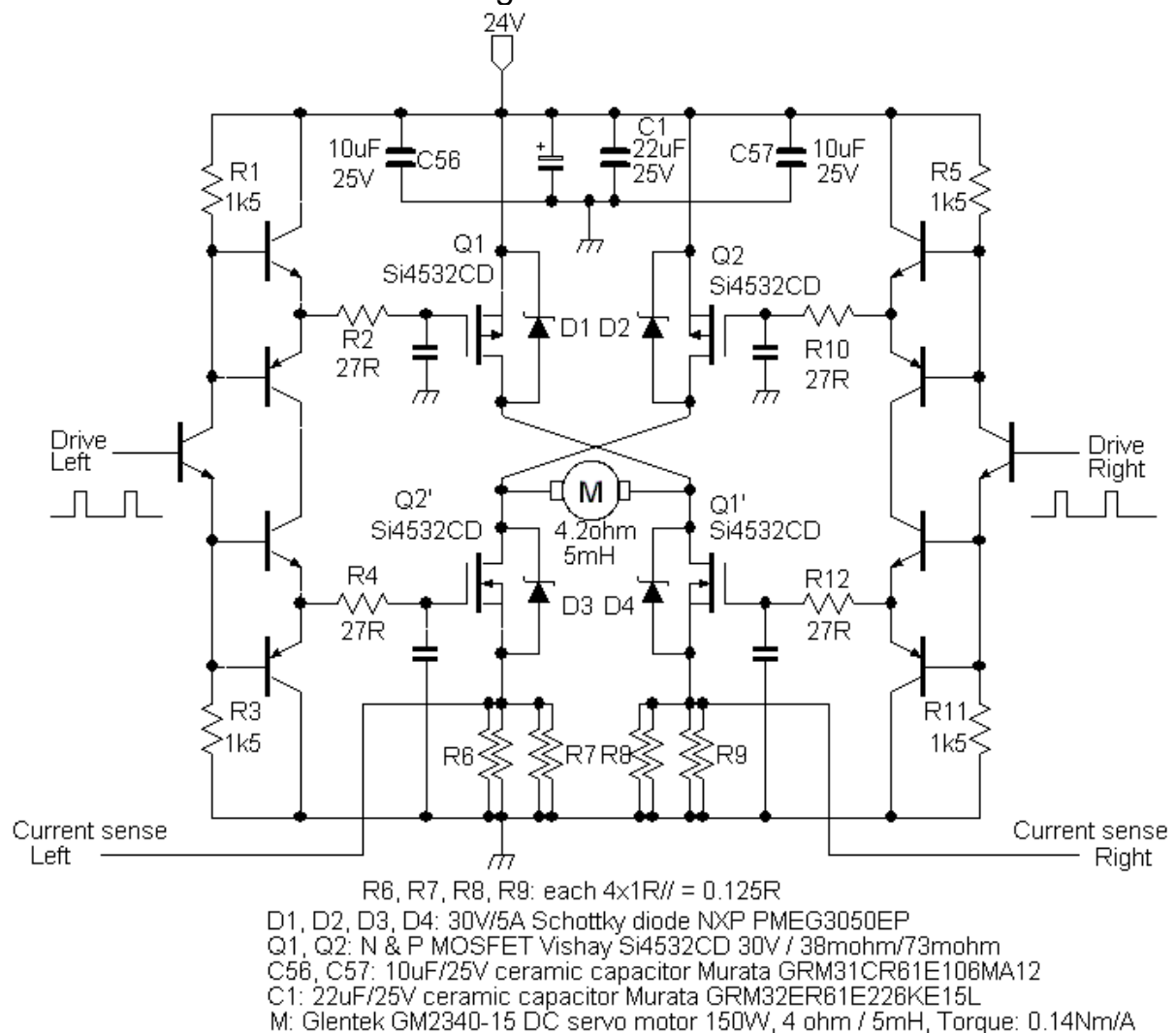
The original motors in this FF stick have 8.5 ohm resistance. you could

probably increase the current up to 1.4A and still keep the same motors. But at higher currents, the voltage drop across the motors will be too high, and switching duty-cycle goes to 100%, for which this system was not designed. Also the original motors become quite hot) . In this design, the supply voltage must be kept at around 24V for all circuits to operate normally. So when choosing motors, best to choose a type that has ~ 12V drop at max current.

Note on DC supply: The original AC/DC 24V supply is quite weak, it cannot deliver much more than the power for original drive. So for higher motor currents, you have to use a 24V supply that can deliver more current.

Description of the Hack:

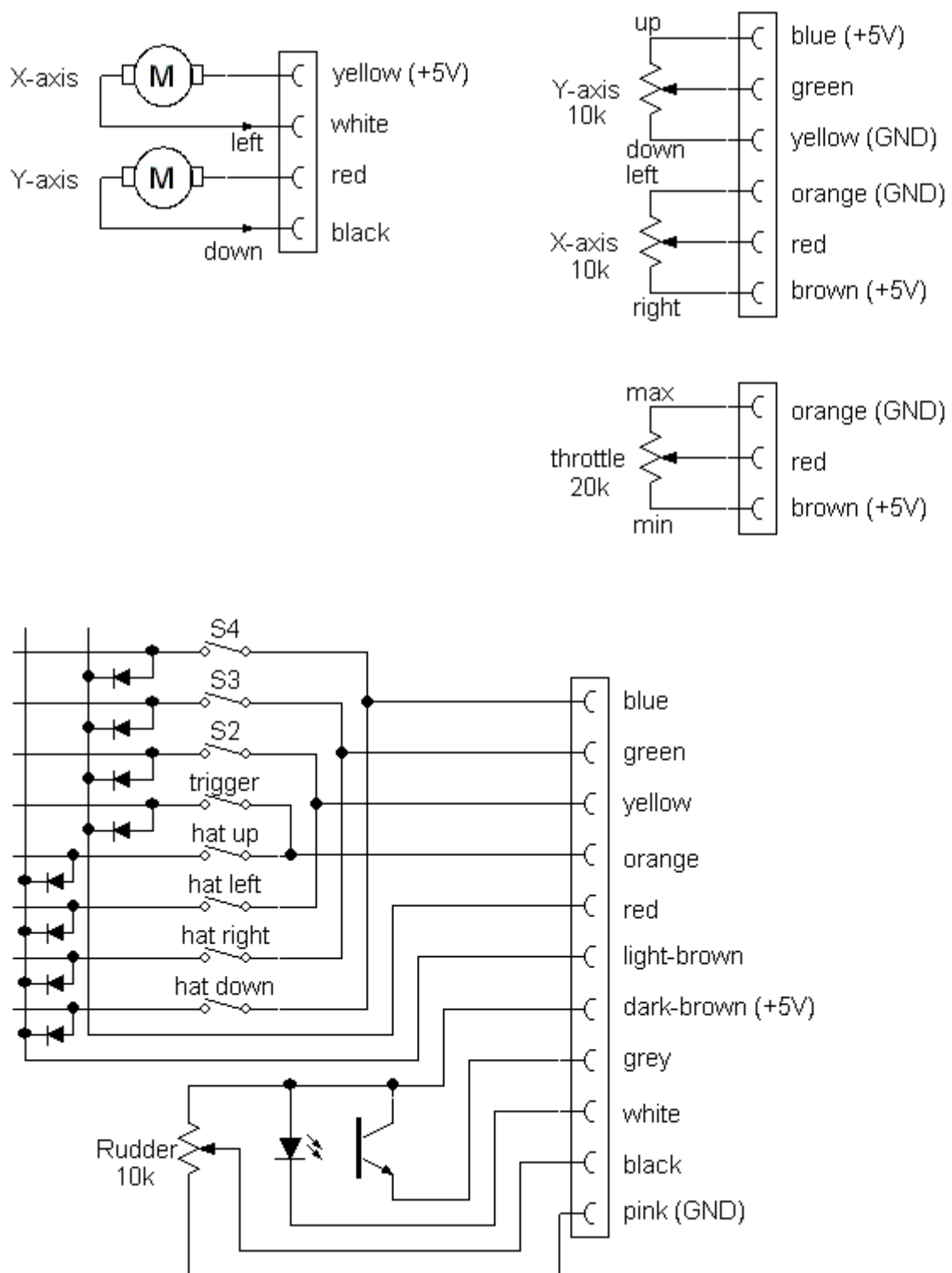
Below is the modified circuit diagram.



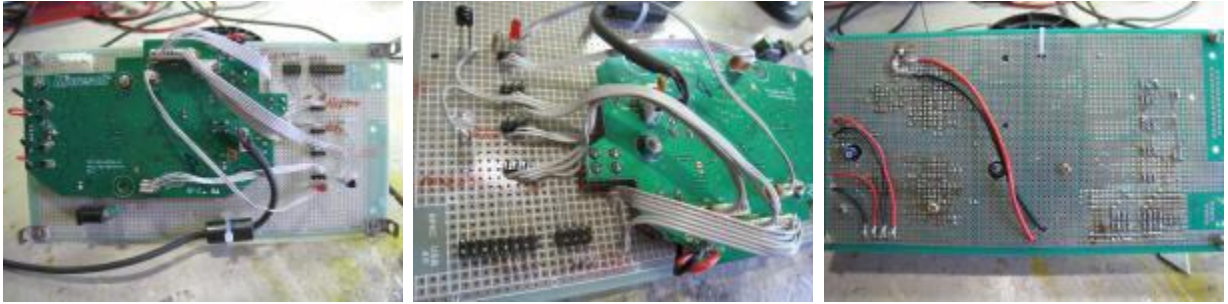
- * I soldered 3 more 1R resistors on each 1R current sense resistor.
- * I changed the MOSFETs from HAT3004R to Vishay Si4532CD MOSFETs which have similar gate drive characteristics, but lower $R_{ds(on)}$
- * I added an NXP PMEG3050EP 30V/5A Schottky diode across each MOSFET to reduce the power dissipation in the MOSFETs
- * I added 1 ceramic filter capacitor next to main supply electrolytic capacitor,

Adding the Schottky diodes is a bit tricky, only 4 can be added on the top side, the other 4 have to be mounted on bottom side, and this requires some cutting in the bottom ground plane.

Other circuits of the Microsoft Sidewinder Forcefeedback-2 stick :



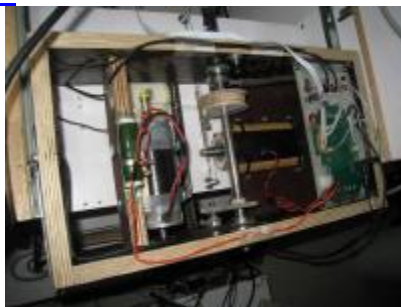
As I wanted to use the hacked board in my Force Feedback Yoke II , I had re-wired the above circuits on a separate board, see below pictures.



I kept the IR sensor link intact, as it can be used during testing to switch-on/off the motor drive.

For the 24V supply, I used a 24V / 3.5A power supply "brick".

This new hacked FF electronics has been build into my [Force feedback yoke II](#) construction.



For the force feedback software link to FS2004, I have used [FSForce](#), which is a great improvement over the standard force feedback from FS2004. No more delays, more realistic elevator trimming, and extremely tweak-able settings for many different effects.

For more info on the mechanical build, see [Force Feedback Yoke II](#) page.