UPCX 530 Power Board

Purpose of the Project

This board is a more powerful board for the UPCX530 motion controller. For details on the controller and its firmware please refer to my website at www.vbesmens.de.
This board supports up to 80V supply voltage for the motor and up to 12 A motor current per phase. If you are using a 70µm copper layer on the PCB, it should be suitable for up to 16A motor current.
It supports sensored BLDC motors as well as DC servomotors.

V. Besmens April 2014

Included in the Project

- This PDF document
- Schematic, Layout (Altium designer 14.2 and PDF files) and Extended-Gerber including GC-Preview (*.gwk) files for a single control board (100mm by 50mm in size). Also included are paneled gerber files for one control board together with 2 pairs of RS422 receivers / transmitters (100mm by 83mm). Their description can also be found on my website.
- A CAD drawing of the heatsink.

Note:
I did not supply a BOM because the circuit is relatively simple.
If you select the shunt resistor (either THD or SMD) make sure that it has low inductance and that the value fits the power consumption of your specific motor.

Technical Data

- The maximum supply voltage for the motors with the current components used is 80V regulated. The maximum motor current is limited by the PCB trace width and should not exceed 12A per motor phase. If you are using a 70um copper layer, the board should be able to drive 16A motor phase current.
- The maximum step frequency I saw was around 120KHz for 1 encoder position per step with 2000 impulse per revolution encoders and a 65W BLDC motor of an unknown brand (this results in 3600rpm). The maximum step frequency depends on the motor type (BLDC or DC), motor specifications, encoder type, cables and motor supply voltage.

Firmware and UPCX Controller

For details on the UPCX530 controller and its firmware refer to the UPCX530 project description on my website.

What you need to build the PCB yourself

- A PCB-manufacturer that accepts Extended Gerber files or GC-Preview files to create the dual layer PCB's. The PCB's need 6mil / 6mil (0.15mm / 0.15mm) technology for structures and gaps with a minimum hole size of 0.4mm.You can use either 35µm copper or, if higher current is needed 70µm copper.
- The ability to solder SMD parts with 0805 size.
- The ability to build a heatsink for the bridge FET's.
- A sensored BLDC or DC motor with TTL-Level A/B encoder
- A drill press or mill and M3 tapping tool to tap the mounting holes for the FETs.

What you need to program the Controller by yourself

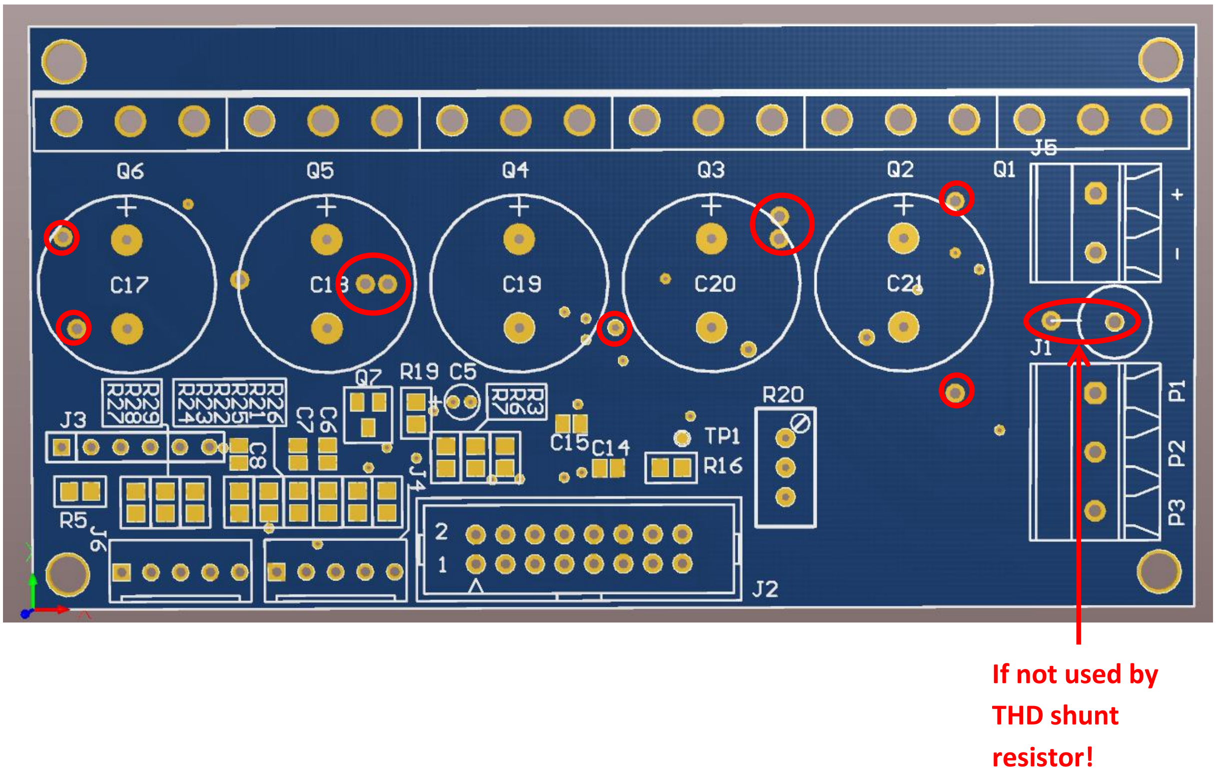
- A PIC programmer that is capable of programming the dsPIC 30F4012-30 (e.g. Microchip PICkit 3). It is connected to J3.

What you need to run the Controller

- You will need to know how to tune a PID loop.
- A power supply with enough power to run the motor(s).
- A regulated power supply with 10 to 12 Volts and 100mA current per control board.
- Another regulated power supply with 5 Volts and 300mA current per control board.
- For first tests of the circuit use a laboratory supply where you can limit the motor current to find the right hall-sensor, motor-phase and encoder-phase connections. This is not always obvious from the motor datasheet and will usually end in a try and error procedure with 3 Hall + 3 Phase + 2 Encoder cables.
- Although not absolutely necessary a TTL frequency generator for step signal generation to test the circuit and the tuning.
- At least one programming adaptor to adjust the PID settings. This is a simple RS232 to TTL level transceiver or a USB-Serial adaptor that delivers a 5V TTL-level output and can handle a 5V TTL level input.
- A PC running Microsoft Windows XP or higher with one serial port (DB9). USB-Serial adaptors are also fine if the driver is installed correctly and you know the correct port number.
- The adjustment software (part of this package)

Notes on Building the PCB

- The minimum component size is 0805.
- The board is populated on both sides.
- There are some components that are not necessary when you want to build a DC controller only. Please refer to the schematics for details.
- You can use either THD or SMD components for the current sensing resistor. You will have to calculate its value depending on the motor type used.
- Make sure that you follow proper ground connection rules. The ground levels of the 3 operating voltages must be kept at the same level all time to protect the FET-drivers.
- The FETs must be isolated from the heatsink and from each other using a silicone PAD.
  The heatsink must also be isolated from the PCB using a stable and nonconductive foil. The soldermask of the PCB is not suitable for isolation!
- Connect the heatsink to Earth ground to avoid discharge sparking to the FETs.
- Adjust the overcurrent protection by measuring the voltage at TP2. Please refer to schematic for details. Adjust it for to 1.5 to 2 times the rated motor current.
- It is a good idea to switch on the motor power after the 5V and 12V supply have stabilized.
- You can use different shunt resistor types and values. Select them to fit your needs.
- Fill the marked vias with solder before mounting any parts onto the PCB:



Notes on Building the Programming Dongle

You can use the programming dongle from the UPCX530 demo board or build one from a MAX232 and some additional capacitors.

Selecting the Motors and Encoders

Refer to the Technical Data section above for maximum voltage and current.
The encoders must generate a TTL-level (5V) A/B signal with an optional Index pulse (for example Avago HEDS 5400 series encoders). The PCB is intended to be used near the motors. If you need some distance between the controller and the motor use RS422 transceivers for the encoder and hall generators that are part of the PCB panel.
A BLDC servo must contain TTL-level (5V) hall sensors that generate 6 states per electrical circle. The three phases for the motor windings must be connected using "star" wiring. I did not test "delta" wiring.

Notes on Connecting a DC motor

Connecting a DC servo motor is easy. Use Pin1 and Pin2 of J1 to connect the motor leads. After adding some P-Value in the adjustment software, you will either find the motor trying to keep its position or you will find the motor running out of its position when you rotate the motor axis and the software reporting "Max error exceeded". In the second case simply swap the motor leads or the A/B leads of the encoder.

Notes on Connecting a BLDC-Motor

Connecting a BLDC motor is somewhat more complicated. You will have to find the correct relationship of the 3 motor phases to the 3 hall signals after adding some P-Value as well as finding out if the A/B phases of the encoder are in correct order. There will be different combinations that will work. I used the try and error method for hall sensor / motor phase connection for some motor types:

Connector	**Nanotec DB57 series***	**Motionstep 57mm series****
J4 Pin1 (VCC)	Orange (VCC)	Red (VCC)
J4 Pin2 (GND)	Black (GND)	Black (GND)
J4 Pin3 (HALL1)	Green (H3)	Blue (Hall Phase C)
J4 Pin4 (HALL2)	Gray (H2)	Yellow (Hall Phase B)
J4 Pin5 (HALL3)	Yellow (H1)	Green (Hall Phase A)
J1 Pin1 (P1)	Brown (U or Phase A)	Green (Phase A)
J1 Pin2 (P2)	Blue (W or Phase C)	Yellow (Phase B)
J1 Pin3 (P3)	White (V or Phase B)	Blue (Phase C)

* www.nanotec.de

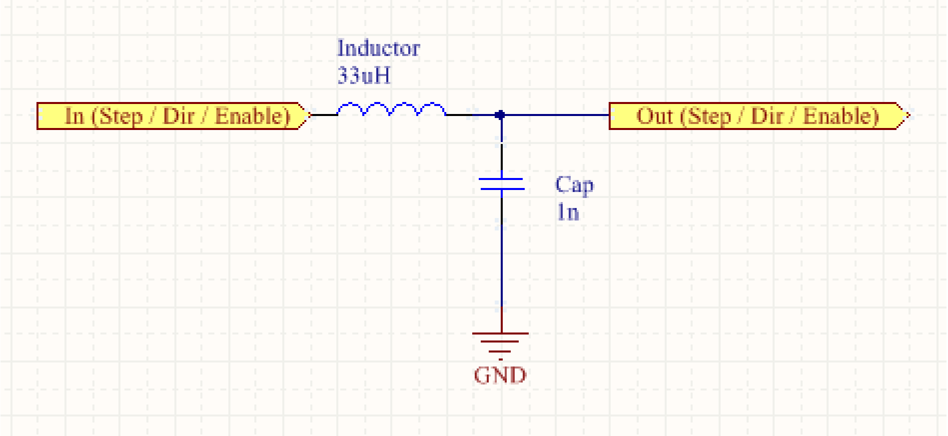
** The company "Motionstep" does no longer exist!

Notes on Control Signals and Power Sequence

There is a special power sequence suggested. VMOT should be the last voltage to be switched on.
The unit is controlled from J1.

Pin (J1)	Signal	Type	Description
1	12V	Power	Regulated 10 to 12V 100mA
2	12V	Power	
3	Step	In (TTL)	Step input on HL transition when Enable is low
4	Dir	In (TTL)	Direction input
5	Enable	In (TTL)	Device is enabled when low
6	Error	Out (OC)	Open collector error output, low on error, positioning errors can be reset by pulling enable high and low again
7	Index	Out (TTL)	Index signal from encoder if available on encoder
8	NC		No connection
9	GND	Power	GND
10	GND	Power	
11	VCC	Power	Regulated 5V 200mA
12	VCC	Power	
13	TX	Out	PID programming TX (57600 Baud 8N1)
14	RX	In	PID programming RX (57600 Baud 8N1)
15	GND	Power	GND
16	GND	Power	GND

The controller generates noise on control (Step/Dir/Enable) signals and is also very volatile to noise on those signals. If you find the unit working as expected using the build-in oscillator but not working stable with external control signals, you can reduce the value of R3, R6, R7 to 1K and / or place an L-C filter into to signal lines as shown below:



You will probably have to change the values for the inductor and / or the capacitor.

You can also try T-Filter components for noise reduction.

Adjusting the PID and BLDC Parameters / Using the Software

Please refer to the UPCX530 / PP4 description and project documentation on my website.

References

Lawrence Glaister's website: http://members.shaw.ca/swstuff/
Maximilien Mousset's website is no longer available (April 2014)
Microchip dsPIC BLDC SVM example (AN957): www.microchip.com
Graphicode GC-Preview to review gerber files: http://www.graphicode.com
My website: http://www.vbesmens.de