



G2Shield r2 – external drivers

Using TinyG2/G2core on an Arduino Due, connect up to 6 external stepper controllers with optocoupler inputs and limit switches for both XYZ and ABC direction. Onboard interface for common spindles and coolant relay.

List of components

SMD:

Qty	Value	Device	Parts
15	100n	C-1206	C1-C14, C81
2	47n	C-1206	C15, C16
17	10k	R-1206	R1-R6, R26-R30, R40-R45
4	560	R-1206	R16, R17, R18, R19
2	47k	R-1206	R20, R23
2	100k	R-1206	R22, R25
20	2k2	R-1206	R7-R15, R21, R24, R31-R39
1	TS912	SOIC-8	IC1
2	ILD217	SOIC-8	OK1, OK2
1	BSS138	SOT23	Q1
18	BC850	SOT23	T1-T18

Through-hole:

Qty	Value	Device	Parts
3	2-pin	Con-KK	COOLANT, SPINDLE_DIR, SPINDLE_EN
1	3-pin	Con-KK	SUPPLY
1	4-pin	Con-KK	SPINDLE_SPEED
6	6-pin	Con-ML6	MOTOR1-6
2	2-pin	jumper	VOUT
1	2x4	pinheader	Control switches
1	1x10,5x1x8,2x18	pinheader	DUEG2
2	2x6	pinheader	Limit switches ABC/XYZ (inputs 1-12)
2	47uF	C-elec	C17,C18

Introduction

Thanks for buying this G2Shield kit for external drivers! SMD components have been premounted for your convenience, so only the remaining through-hole parts need to be mounted. Make sure you read the complete instructions before you start mounting. Assembly can be done by an experienced hobbyist in about half an hour.

List of components

Please check if the list of components is complete. If desired, you can adapt them to your needs.

Tools

- ✓ Soldering iron and solder
- ✓ Multi-meter (voltage and resistance)
- ✓ Side-cutting pliers

General instructions

Mounting

With the SMD components premounted, mounting the remaining through-hole connectors is straightforward. Start with the Arduino connection by connecting the pinheaders to the Arduino and solder on the top side of the PCB. This way everything is nicely aligned. Then remove the Arduino and solder the bottom side by placing components with the lowest height in steps first and proceed up to the highest height.

External components

Connect external components according to the instructions shown in the schematic:

- Stepper motors
 - For external stepper drivers with optocoupler inputs
 - Enable, direction, step/pulse interface
- Limit switches
 - 12 general purpose inputs, labeled as Xmin/max to Cmin/max on the pcb. G2core allows them to be freely configured, e.g. you can configure one of them as a probe input
 - max 9 inputs supported currently by G2core
 - Both NC and NO supported by G2core
- Control switches
 - not yet supported by G2core
- Coolant relay
 - For external 5V relays
- Spindle control
 - For external common VFD spindle controllers with direction, enable and voltage input for speed control
 - Direction, enable, speed interface
 - Onboard optocouplers
 - Two voltage outputs for controlling spindle speed (only first one supported by G2core) using a PWM to analog output voltage conversion.
 - Jumper 3+4 need to be left open if an external supply voltage (commonly 10V) from the spindle is used. The jumpers can be put on to use the internal +5V instead. Never use the jumpers in combination with an external supply as that might fry the g2shield and arduino.

G2core software

For a thorough introduction to the G2core software, please check the Synthetos website. A quick introduction on how to get G2core and use it in combination with the G2Shield:

<https://www.djuke.nl/en/support/18-cnc/46-using-g2-core>

Testing

Do not connect external components yet until below voltages are checked!

Connect the G2Shield on top of a Arduino Due with G2core software. Power the Arduino Due using USB or external supply.

- The voltage between corresponding pins (1-2, 3-4 etc) of the limit switch inputs is 3.3V
- The voltage on pin 2,4,6 of the Motor connectors is 5V
- The voltage on COOLANT-1 is 5V
- The voltage on SUPPLY-1 is 3.3V
- The voltage on SUPPLY-3 is 5V

Now, the external components can be connected and tested from G2core. The easiest way to do so is to clone the GIT repository from <https://github.com/Djuke-DIYAudio/g2> and build the firmware with settings_Djuke_test.h

Connect to the Arduino Due using USB and a terminal program.

Useful test commands:

Command	Description
\$in	Shows input values
\$clear	Clears alarm state
M3 S100	Enables spindle clockwise with 100% PWM output
M4 S50	Enables spindle anti-clockwise with 50% PWM output
M5	Disable spindle
M7	Enable coolant relay
M9	Disable coolant relay
G0 x#	Move x-axis to position # (similar for y/z/a/b/c axis)

Schematic

