Arduino Dual L6470 Stepper Motor Shield

Data Sheet

Adaptive Design ltd

V1.0

20th November 2012

General Description

The Arduino stepper motor shield is based on L6470 microstepping driver and is capable of driving two motors or four motors when two shields are stacked together, allowing the user to build complex robotic hardware, by the means of integrated motion hardware. All communication with the drivers is done over the SPI interface using just one CS (chip select) line, the drivers being cascaded on the bus, two digital lines for BUSY and FLAG (errors or warnings) and an optional SCLK (step clock) line for enabling coordinated movement driven externally. The drivers have integrated home-ing mechanism and they handle the home switch in hardware simplifying the software development. More, the drivers can handle programmable velocity profiles, with acceleration and deceleration handled in hardware, and many other interesting features. For a complete description of features please read the L6470 Datasheet from ST Microelectronics Website.

Power

The stepper motor shield is powered externally via a terminal block with DC (Please see the figures for polarity) and the recommended voltage is from 12V to 24V max. The voltage is dictated by the motors connected to the shield, and is open to experimentation when the parameters are tuned for a particular motor/application.

The shield is not loading the Arduino board regulators, so it's irrelevant how the Arduino board is powered, USB or external.

The power consumption is approx. 10-20 mA (with the motors disabled) and is dependent on the status leds. After power on reset the RED leds (FLAG) are lit, for all drivers, signalling UVLO (under voltage lockout) – this is the normal state after reset, and is cleared after the first status read from software.

We are recommending to use a stabilized power supply with current limit and display for easy debug in prototyping stage, please see the video on our website.

The shield is not protected for power reversal so please mind the polarity when connecting the power

CLK source

The Driver chips can be clocked locally by using the internal 16MHz oscillator or using the 32MHz Cristal oscillator fitted on the PCB. We do recommend using the XO oscillator for application where precision is required, or using just one internal 16MHz and forwarding the clock to the next chip, will ensure both chips will run using the same clock. When using two shields stacked together we

recommend using the 32MHz as single option. All clock setting are available via register programming from software.

External Signals

The main communication interface is the SPI interface, with a max speed of 5MHz (please see the example software on our website)

The CS line is programmable via a jumper and a legend is printed on the PCB silkscreen.

Jumper 11	CS
Position 1-2 closed	Arduino 9 (PB1)
Position 2-3 closed	Arduino 10 (SS)

The shield can be configured as single (just one shield) or Master / Slave when two shields are stacked together. When one shield is used we are going to have 2 devices chained in the SPI bus, and when we have two shields we are having 4 devices. The jumper setting are as follows

Setup	J12	J13
Single (default)	1-2 (closed)	2-3 (closed)
Master	1-2 (closed)	1-2 (closed)
Slave	2-3 (closed)	2-3 (closed)

Note: When using two shields stacked you need to free up Arduino P3 PD3_3 (it need to be an input) because this signal is forwarding the SPI bus from one shield to another.

FLAG – signal

The flag signals from drivers are driving RED leds on the pcb and then are OR-ed together in one signal Arduino pin 6 (PD6) to minimise the pin count. This signal can be used as am interrupt for Arduino, but the software needs to interrogate all devices on the bus to determine which one has raised the flag.

BUSY – signal

The busy signal is driving GREEN led for each channel and then is OR-ed with all busy signals and forwarded to Arduino pin 7 (PD7) – this signal is provided to poll the driver during execution of a command.

Both FLAG and BUSY signals are accessible via registers as well.

STCK1, STCK2 signals

Step Clock signals are accessible on two pins Arduino pin 4 & Arduino pin 8 (PD4 & PD0 respectively) for driving the step movement directly from Arduino. For using this feature the proper STCK mode has to be programmed and the Direction of the movement via config register (is not enabled by default)

Note: When using 2 stacked shields the STCK1 signal is shared between the first drivers on each shield, and STCK2 is shared between the second drivers on each shield. Please keep this in mind when writing the software.

RESET DRIVER signal

A hardware reset of the drivers (all drivers) can be done by driving this line LOW, this line is attached to Arduino pin 5 (PD5) – this line is pulled high, with a 10K resistor.

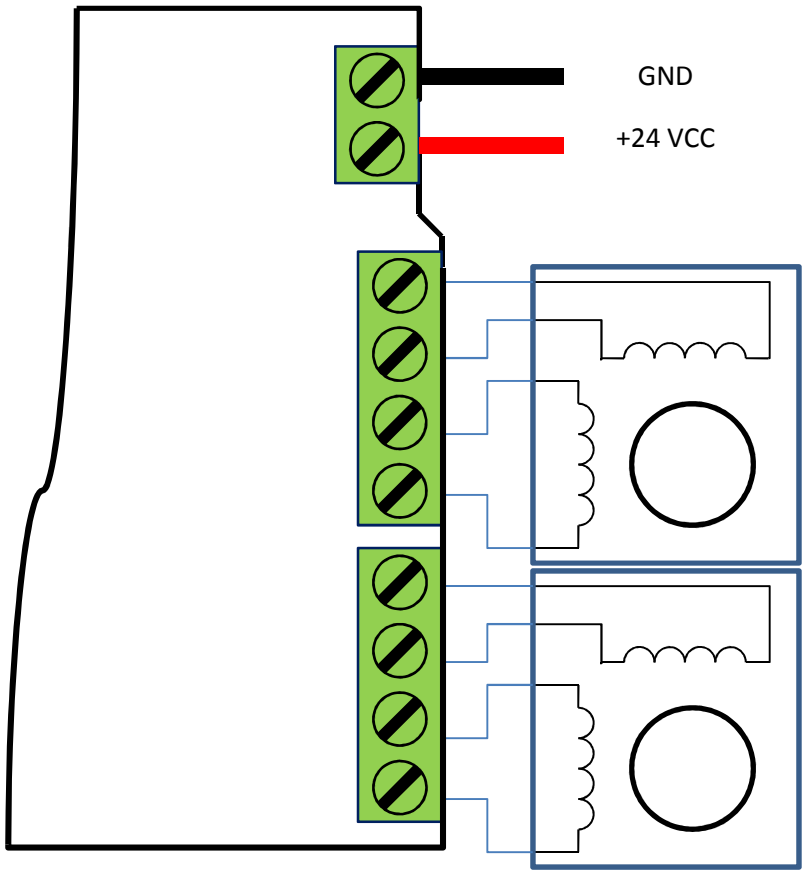
Summary of all signals used

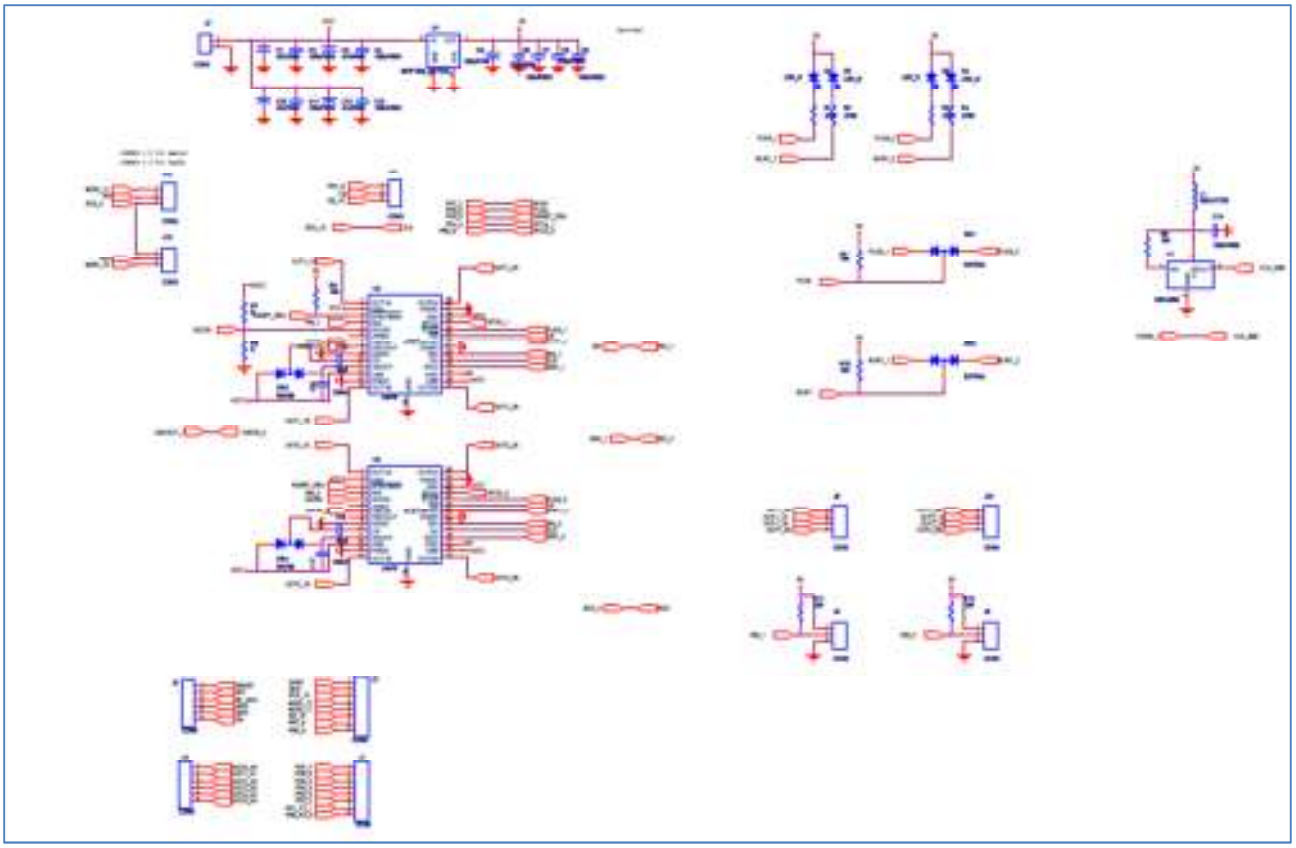
Name of signal	Arduino pin	Note
SPI MISO	pin 12	
SPI MOSI	pin 11	
SPI SCLK	pin 13	
CS	pin 10 or pin 9	Depending on J11 position
FLAG	pin 6	
BUSY	pin 7	
STCLK1	pin 4	Shared when using two shields
STCLK2	pin 8	Shared when using two shields
RESET DRV	pin 5	
Reserved	pin 3 – input mode	When using two shields

Motor Connection

The shield is accepting bipolar stepper motors (4 wire) with 2 phases. Please make sure the motor is connected correctly otherwise the driver output bridge will be irreversible damaged.

Each motor has his own terminal block with four terminals.





Shield Schematics

Document History

Date	Notes	Version
20 Nov 2012	Initial Draft	V1.0