3/31/2012

4X4 Driver Shield Manual

High current, high side switching for Arduino

Logos Electromechanical

4X4 Driver Shield Manual

High current, high side switching for Arduino

Introduction

The Logos Electromechanical ARD-SRG-IPS4X4 4X4 Driver Shield is designed to enable users to switch DC loads up to 5A at up to 30V with no heat-sinking. It uses four International Rectifier IPS6044 four channel fully-protected high side MOSFET switch ICs. Each includes over-temp, over-current, and under-voltage protection on each switch. The high-side topology is safer in many applications than the more common low-side topology because a short to ground cannot energize the circuit while the switch is turned off, as it can with a low-side switch. Each four-channel chip has an independent power supply to allow the user maximum flexibility in configuring this board for their application.

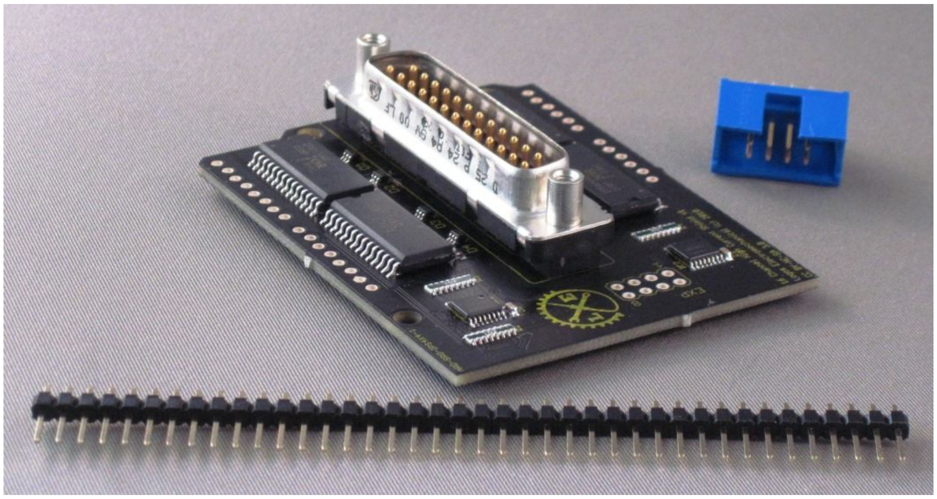

Figure 1: Kit Contents

The Arduino interface pins are arranged so that the shield can be driven with the hardware SPI bus. This allows a much higher rates of data transfer to the shield or a string of shield than is possible by bit-banging the data. It’s fast enough to do fast, high-resolution PWM on large numbers of channels.

The PCB is made with generous traces to handle the current with minimal voltage loss and heating on the board. A pair of 74AHCT595 8-bit shift registers allow the control of sixteen high current channels from only four Arduino pins using the shiftOut() function. The use of shift registers makes it possible to daisy-chain as many as 25 boards off a single Arduino, for a total of up to 400 high current channels.[i] In addition, it can be daisy-chained with all other Logos output boards with a shift register input, i.e. those with a part number of the form *-SRG-*.

This board is shipped as a kit, with none of the connectors soldered to the board. This gives the user the flexibility to install the desired connectors for the application.

Board Overview

Board Layout

The switches are laid out around a vertical DB-25 to carry all of the high current signals . The DB-25 was chosen due to its relatively high current capacity (5A/pin), long history, and the enormous range of types and grades the mating connectors are available in.

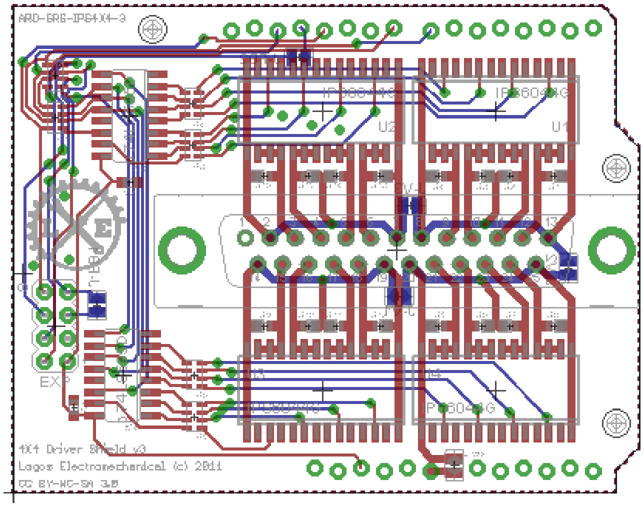

Figure 2 Board Layout

Each of the four IPS6044 chips houses four channels. Each IPS6044 draws its power supply from a pair of pins on the main connector and switches it to four output pins on the same connector. This arrangement allows the switching loads with different power requirements on a single shield. However, each group must be arrange so that it draws no more than 10A continuous in order to avoid overloading the connector. An anti-parallel diode protects each switch, allowing fast switching of large inductive loads without damage to the switches.

The expansion connector, shift registers, and associated resistors are located on the left side of the shield as seen in Figure 2. The shift registers are powered from the +5V logic supply of the host Arduino. The board is shipped with the Arduino and expansion connectors provided but not installed in order to permit the user to to install the connectors most appropriate to their application.

Pin Descriptions

Table 1: Arduino Pins

Arduino Pin	Name	Function
4	DAT	Data Return
7	LAT	Register Clock (positive edge)
8	MR	Master Reset (active low)
11	SIN	Serial Input
13	SCK	Serial Clock (positive edge)
+5V	5V	Logic Supply
GND	GND	Logic Ground (Shorted to power ground)

- **DAT:** Data return. This allows boards on a daisy chain to return data to the controlling Arduino. Each shift register compatible shield has a jumper that allows this pin to be connected to the SOUT pin (J-FB1) or digital pin 4 (J-FB2). The last board in a chain must have this pin jumpered to SOUT in order to enable data return and the board mounted to the host Arduino must have this pin jumpered to digital 4 in order to allow the host to read the data.
- **LAT:** A positive edge on this pin latches the current contents of all of the the shift register to the outputs.
- **MR:** Pulling the Master Reset low clears all of the shift registers.
- **SCK:** A positive edge on this pin move the current value of the SIN pin to the least significant bit of the first shift register (IC1), and shifts every current bit in each shift register one bit up. This is connected to the SCLK line of the hardware SPI bus on an Arduino Uno.
- **SIN:** The value of this pin is shifted in to the least significant bit of the first shift register on each rising clock edge. This is connected to the MOSI line of the hardware SPI bus on the Arduino Uno.

Table 2: Expansion Connector Pinout

Input Pin	Name	Function
1	+5V	+5V power for shift registers
2	SCK	Serial Clock (positive edge)
3	GND	Ground
4	MR	Master Reset (active low)
5	DAT	Data Return

Input Pin	Name	Function
6	LAT	Register Clock (positive edge)
7	SOUT	Serial Data Out (to next board)
8	SIN	Serial Data In (from last board/host)

- **SCK, SIN, LAT, MR, DAT:** These pins work as described above.
- **SOUT:** When a bit is shifted past the end of the second shift register (IC2) it appears on this pin. This is what allows multiple shields to be daisy chained.

Table 3: Power Connector Pinout

Power Pin	Name	Function
1	GND	Common Ground Reference
2	VCC2	Power supple for switches 0 - 3
3	OUT3	Switch 3
4	OUT2	Switch 2
5	OUT1	Switch 1
6	OUT0	Switch 0
7	VCC2	Power supple for switches 0 - 3
8	VCC1	Power supple for switches 4 - 7
9	OUT7	Switch 7
10	OUT6	Switch 6
11	OUT5	Switch 5
12	OUT4	Switch 4
13	VCC1	Power supple for switches 4 - 7
14	VCC3	Power supple for switches 8 - 11

Power Pin	Name	Function
15	OUT8	Switch 8
16	OUT9	Switch 9
17	OUT10	Switch 10
18	OUT11	Switch 11
19	VCC3	Power supple for switches 8 - 11
20	VCC4	Power supple for switches 12 - 15
21	OUT12	Switch 12
22	OUT13	Switch 13
23	OUT14	Switch 14
24	OUT15	Switch 15
25	VCC4	Power supple for switches 12 - 15

Electrical Characteristics

Table 4: Switch Electrical Characteristics

Symbol	Parameter	Max	Typ	Min	Units
R_{ds}(on)	Resistance, on state	0.13	0.12	0.11	Ω
V_{clamp}	Clamping Voltage	39	39	37	V
I_{lim}	Current Limit	10	7	4	A
V_{cc}Op	Recommended Operating Voltage Range	28	-	6	V
Td_{on}	Turn-on Delay Time	-	5	15	µs

Symbol	**Parameter**	**Max**	**Typ**	**Min**	**Units**
T_r(90%)	Rise Time to 90% of V_{CC}	-	4	20	

Board Usage

Electrical Connection

The ideal connector to use is a high-grade DB-25 with high-quality crimp-style socket terminals. Solder-style are also acceptable, but less durable and more labor-intensive to assemble correctly. In both cases, appropriate strain-relief is crucial to reliability and durability. A good quality backshell for the connector will come with appropriate strain relief components. Figure 3 shows a schematic of a typical connection.

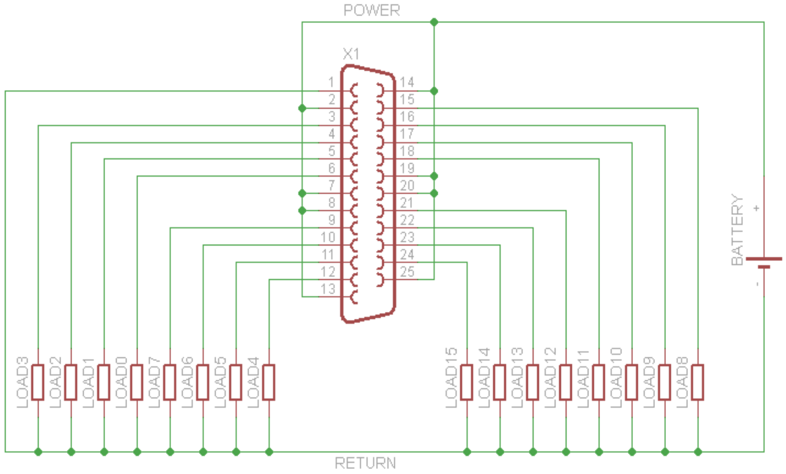

Figure 3: Typical power connection

Software Interface

This board is designed to work with the `shiftOut()` library function from the Arduino library. Since this board includes two shift registers, two calls are required in order to operate all of the switches. The following code will write the contents of `lowByte` and `highByte` to S0-S7 and S8-S15, respectively.

```
digitalWrite(7, LOW);   // Prepares latch
digitalWrite(8, HIGH);  // Deactivates master reset
shiftOut(11, 13, MSBFIRST, highByte);     // shift data for OUT8-OUT15
shiftOut(11, 13, MSBFIRST, lowByte);      // shift data for OUT0-OUT7
```

```
     digitalWrite(7, HIGH);  // latch data
```

On the last line, all of the switches will switch at once. The most significant bit of `highByte` will control OUT15 and the least significant bit will control OUT8 and likewise for `lowByte`, OUT7 and OUT0.

Daisy Chaining

This shield is designed to be daisy-chained in order to get greatly expanded numbers of high current outputs. This requires a cable of the type shown in Figure 4. All of the pins are carried over, with the exception of SIN & SOUT (pins 8 & 7, respectively). SOUT of the source shield must be connected to SIN of the next shield, and so on down the chain.

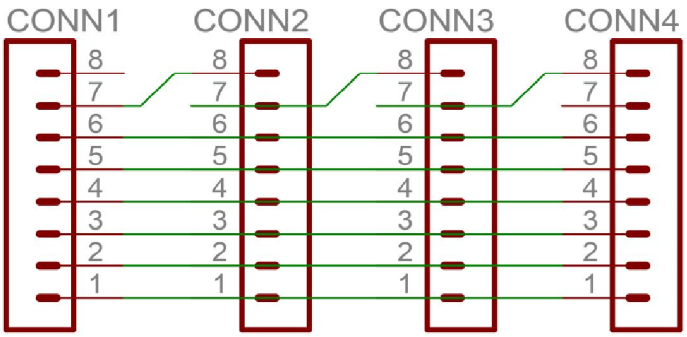

Figure 4: Daisy chain cable schematic

If you're using IDC connectors with ribbon cable to make your daisy chain cable, this is straightforward. Assemble the first connector normally. At the second connector, separate and swap conductors seven and eight. After assembling the connector, cut conductor seven (now plugged into pin eight of the second connector) downstream of the second connector and conductor eight (now plugged into pin seven of the second connector) upstream of the connector. Assemble the third connector normally, but after you have assembled it, trim conductor eight downstream and conductor seven upstream. Assemble the fourth connector as you did the second and the fifth as you did the third, alternating for any additional connectors. See Figure 5 for how it should look. Connector 1 is on the left end of the cable.

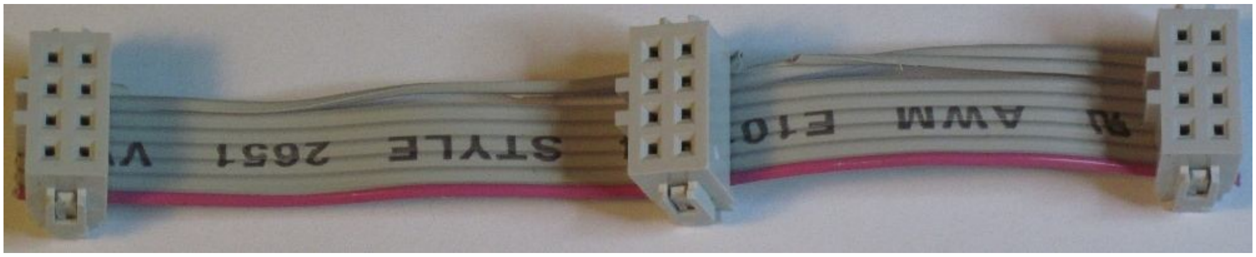

Figure 5: Daisy chain cable

Certain other Logos Electromechanical parts also use shift registers compatible with this shield – see the product page for which ones. They will have expansion connectors compatible with the eight pin connector on this shield.

The daisy chain also contains the option to pass information from the daisy chain back to the controlling host. This does nothing on this board except wrap the transmitted value back to the host, but is included for compatibility with future planned boards.

There are two jumpers on the board that control this function – one between the SOUT and DAT lines on the expansion connector and one between the DAT line and digital pin 4 on the Arduino. In a project where no wrap-around or feedback is desired, both jumpers should be left open. If feedback is desired, the SOUT-DAT jumper should be made on the last board in a chain, in order to wrap the serial stream back to the return line. The DAT-D4 jumper should be made on the board connected to the host Arduino, in order to allow the host to receive the returned serial stream.

Stepper Motor Control

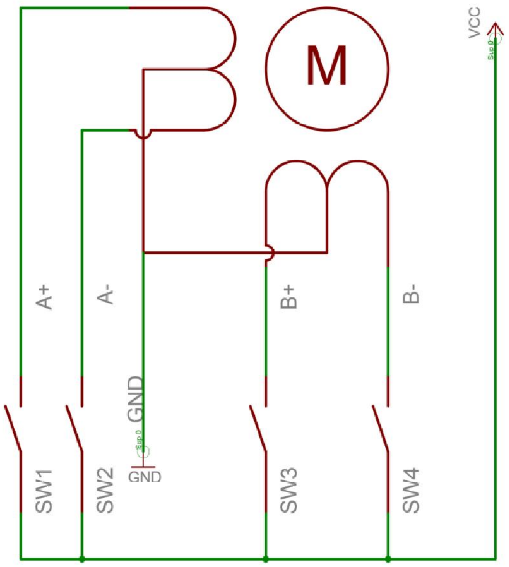

Figure 6: Unipolar steppper wiring schematic

This shield can be used, with some caution, to control up to four unipolar stepper motors. However, the connection is a little different than the usual for conventional unipolar stepper motors. Typically, one connects the common lead(s) of the stepper motor to a power source and then switches the four coil terminals to ground in a pattern that moves the stepper motor as desired.

However, the switches on this board are all high-side switches. That means that you have to flip the polarity of all of the coils, and wire the common terminal to ground. While this works with most unipolar stepper motors, it cannot be guaranteed to work with any particular motor.

References

Datasheet for 74HC595 shift registers http://datasheet.octopart.com/MM74HC595MTC-Fairchild-datasheet-12735.pdf

Datasheet for IPS6044GPBF http://www.irf.com/product-info/datasheets/data/ips6044g.pdf

Arduino Language Reference http://arduino.cc/en/Reference/HomePage

[i] The limiting factor on the number of boards that can be daisy-chained off a single Arduino is the fan-out of the pins that drive the three inputs common to all devices in the chain (SCK, MR, and LAT). This can be expanded by adding a non-inverting buffer to drive these pins out farther along the chain.