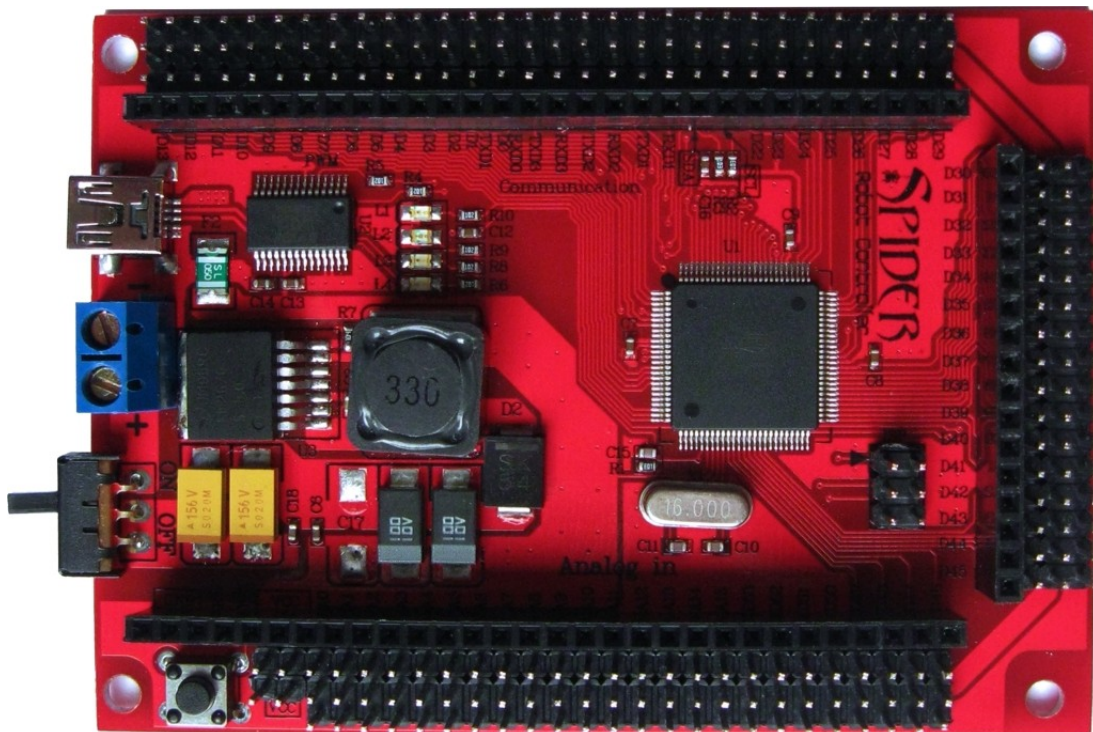


Red Back SPIDER Robot Controller

The Spider robot controllers were designed for robots requiring a large numbers of servos such as humanoid and hexapod style robots. Using an ATmega1280 CPU this controller is fully compatible with the Arduino Mega (1280).

This controller includes a 3A switch mode power supply and is capable of driving up to 48 micro servos directly from the PCB. All of the processors 70 I/O pins are terminated as both a female header and a servo compatible 3pin male header.



Features:

- ATmega1280 16MHz CPU
- 128K FLASH, 8K SRAM and 4K EEPROM
- USB interface and ISP socket
- Built in 3A switch mode power supply (7V – 30V input)
- 70 I/O pins with male and female headers
- Support for up to 48 servos
- 16x 10bit analog inputs
- 15x 8bit PWM outputs
- 1x I2C
- 6x External interrupt pins
- 4x Serial communication
- Comes with Arduino boot-loader installed



Programming the Spider controller:

The Spider controller has been designed to be 100% compatible with the Arduino Mega and comes with the Arduino boot loader installed. The boot loader allows programs written in the Arduino IDE to be uploaded via the USB interface. More information about Arduino can be found here: <http://arduino.cc/en/Guide/Introduction>

This manual assumes use of Arduino 0022 or later, which can be downloaded from here: <http://arduino.cc/en/Main/Software>.

A full list of commands can be found here: <http://arduino.cc/en/Reference/HomePage>

Communications:

The Spider controller has 4x serial ports and 1x I²C interface. Serial interface 0 is used by the USB interface. Serial ports 1,2 and 3 are available for use with devices such as serial interface LCD's, Xbee, Blue Tooth and WiFi modules or even other processors. Instructions on how to use these serial interfaces can be found here:

<http://arduino.cc/en/Reference/Serial>

The I²C interface (SDA pin 20, SCL pin 21) allows the controller to communicate with devices such as external memory, real time clock (RTC), DC motor controllers and more. For the I²C interface to function properly pullup resistors must be used. Some devices will include these resistors. If necessary you can enable the internal pullup resistors on pins 20 and 21. Instructions on how to use the I²C interface can be found here:

<http://www.arduino.cc/en/Reference/Wire>

Digital I/O pins:

All 70 of the Spiders I/O pins can be used for simple digital inputs or outputs. By default, all pins except digital pin 13 have their mode set to input. When a pin is in input mode it is in a high impedance state (effectively open circuit). Digital pin 13 is set to output by the boot loader and has an LED attached. Although D13 can be used as an input the LED may interfere.

Use the pinMode() command to change a pins mode between input and output. Use digitalWrite() and digitalWrite() commands to read and write to these pins. **Note:** Analog pins A0 - A15 are digital pins D54 – D69.

Writing a "1" to a digital pin while it is configured as an output will connect that pin to Vcc(+5V). Writing a "0" to a digital pin while it is configured as an output will connect that pin to ground (0V). Each pin is capable of sinking or sourcing up to 40mA maximum but care must be taken to limit the total current to 200mA. If you wish to drive large numbers of LEDs or other devices then the output pins should be buffered.

Writing a "1" to a digital pin while it is set as an input will enable that pins internal pullup resistor. When enabled, an internal 20K resistor is connected between that pin and Vcc (+5V). Writing a "0" to that pin while configured as an input will disable the pullup resistor.



Analog pins:

Pins A0 to A15 are by default analog inputs. Each analog input has 10bit resolution and can measure the voltage on its pin. Input voltage should not exceed the analog reference voltage (default value is +5V).

The input voltage is measured using the `analogRead()` command. The reference voltage is `Vcc(+5V)` by default. The reference voltage can be changed using the `analogReference()` command.

PWM outputs:

The Spider controller is capable of generating 8bit resolution pulse width modulated outputs on digital pins D2 – D13 and D44 – D46. The output of a PWM pin is driven high and then low repeatedly with the duty cycle being adjusted to synthesize an analog output. By adding a simple RC filter to a PWM output a true analog output can be generated.

PWM outputs are generated using the `analogWrite()` command. As the internal timers of the processor are used to generate these outputs they may be disabled by other commands using the same timer. Pin assignments should be planned to avoid conflicts.

External interrupts:

The Spider has 6 interrupt pins. Interrupts allow functions to be called only when an external event occurs. These pins are useful for monitoring devices such as encoders. The interrupts and their pins are:

Interrupt 0 – D2
Interrupt 1 – D1
Interrupt 2 – D21
Interrupt 3 – D20
Interrupt 4 – D19
Interrupt 5 – D18

More information on the interrupt library can be found here:

<http://arduino.cc/en/Reference/Interrupts>

Using the EEPROM memory:

The Spider's Atmega1280 includes 4K of EEPROM memory that can be used to store information while the power is off. More information on using the EEPROM library can be found here: <http://www.arduino.cc/en/Reference/EEPROM>



Using servos:

All 70 I/O pins have a servo compatible 3 pin male header. The pin closest to the outer edge of the PCB is ground, the center pin is +5V and the inner most pin is the signal. This pin arrangement is also useful for powering sensors.

Most miniature and standard servos require between 4.8V and 6V and will work happily directly from the PCB. High-powered servos requiring 6V or more should be powered via an external power source or directly from the battery.

The Spider can drive up to 48 servos simultaneously using the Servo library:

<http://www.arduino.cc/en/Reference/Servo>

The Servo command uses a timer for each 12 servos used starting with Timer 5. As these timers are also used for commands like PWM you need to plan your pin assignments to avoid conflicts.

1 to 12 servos use timer 5 disabling PWM on pins 44,45 and 46.
13 to 24 servos use timers 1&5 disabling PWM on pins 11,12,44,45 and 46.
25 to 36 servos use timers 1,4&5 disabling PWM on pins 6,7,8,11,12,44,45 and 46.
37 to 48 servos use timers 1,3,4&5 disabling PWM on pins 2,3,5,6,7,8,11,12,44,45 and 46.

Servos can be assigned to any digital pin from D0 - D53.
Analog pins A0 - A9 (D54 – D63) can also be used if required.

A short tutorial including sample code for driving 48 servos can be found here:

<http://letsmakerobots.com/node/25923>

Shields:

The female headers on the Spider controller are spaced so that a shield can be easily made from standard prototype PCB. Shields allow you add custom circuitry to your controller.

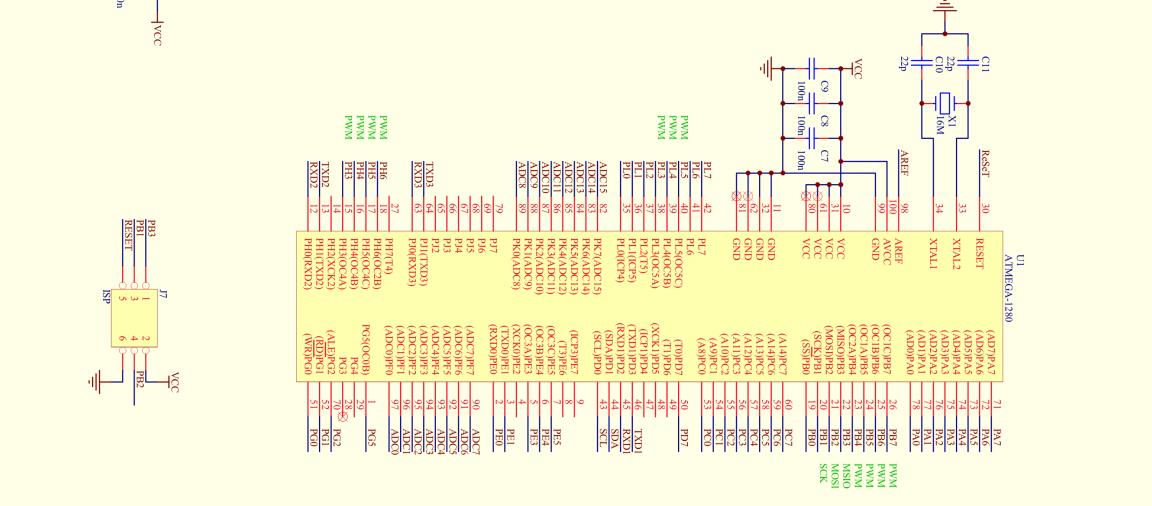
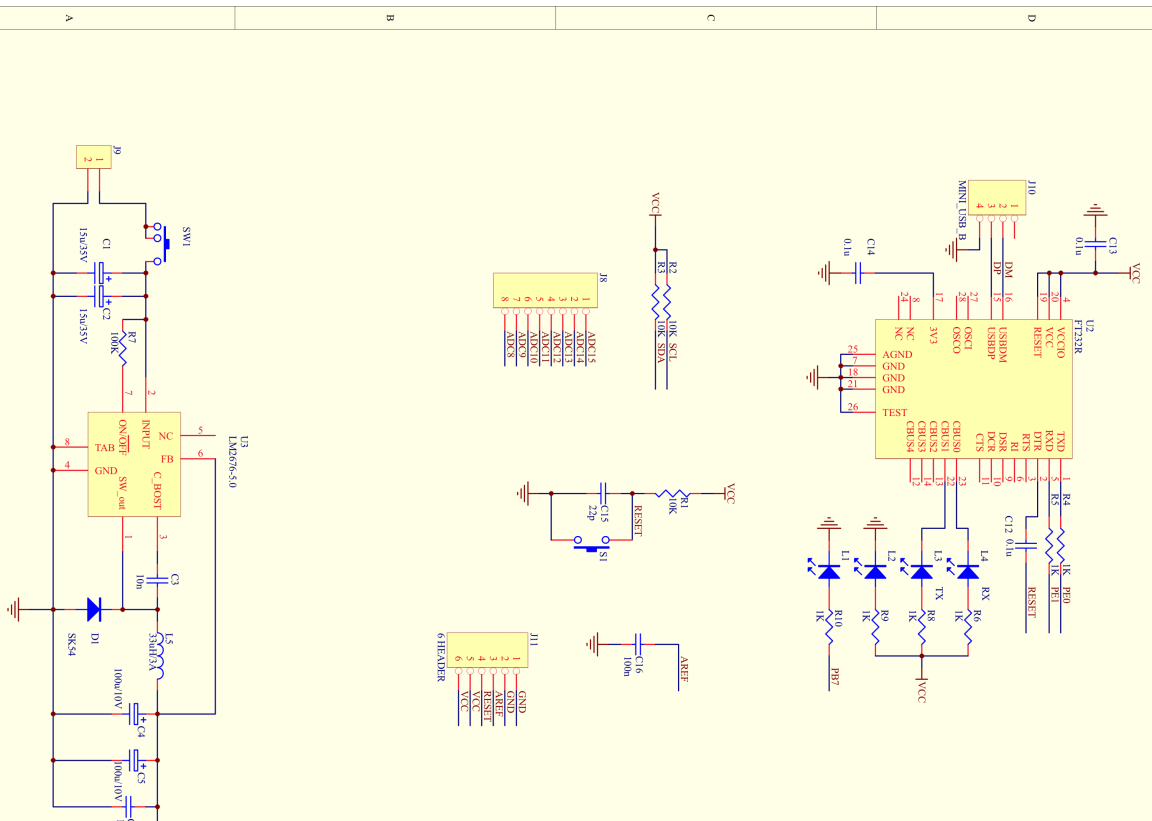
Alternative programming methods:

Experienced users may choose to use WinAVR or AVR studio to program their controller directly via the ISP socket. The bootloader can be removed which will free up an additional 4K of memory.

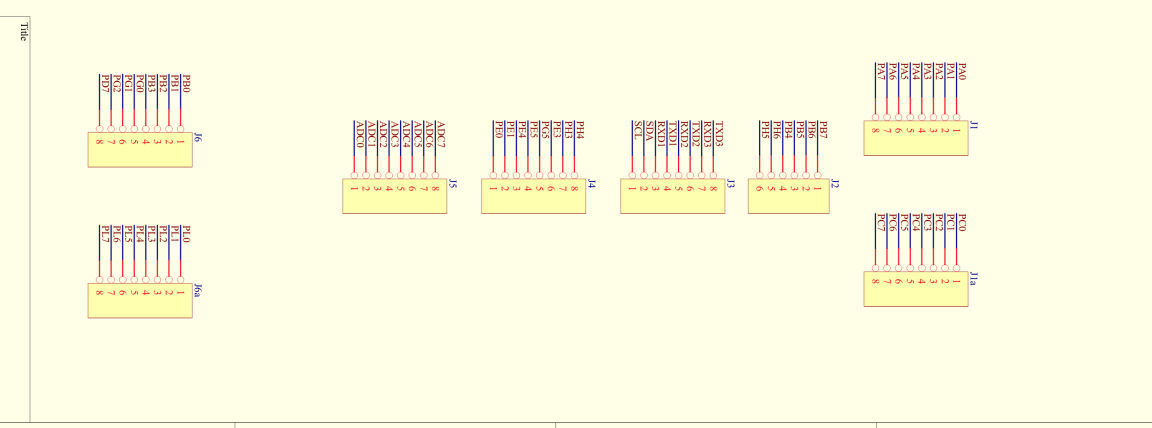
WinAVR: <http://winavr.sourceforge.net/>

AVR Studio 5: http://www.atmel.com/microsite/avr_studio_5/





Pin	Function	Pin	Function
1	AVCC	51	RESET
2	AREF	52	RESET
3	ADSC	53	RESET
4	ADIF	54	RESET
5	ADYGO	55	RESET
6	ADYDF	56	RESET
7	ADYDF	57	RESET
8	ADYDF	58	RESET
9	ADYDF	59	RESET
10	ADYDF	60	RESET
11	ADYDF	61	RESET
12	ADYDF	62	RESET
13	ADYDF	63	RESET
14	ADYDF	64	RESET
15	ADYDF	65	RESET
16	ADYDF	66	RESET
17	ADYDF	67	RESET
18	ADYDF	68	RESET
19	ADYDF	69	RESET
20	ADYDF	70	RESET
21	ADYDF	71	RESET
22	ADYDF	72	RESET
23	ADYDF	73	RESET
24	ADYDF	74	RESET
25	ADYDF	75	RESET
26	ADYDF	76	RESET
27	ADYDF	77	RESET
28	ADYDF	78	RESET
29	ADYDF	79	RESET
30	ADYDF	80	RESET
31	ADYDF	81	RESET
32	ADYDF	82	RESET
33	ADYDF	83	RESET
34	ADYDF	84	RESET
35	ADYDF	85	RESET
36	ADYDF	86	RESET
37	ADYDF	87	RESET
38	ADYDF	88	RESET
39	ADYDF	89	RESET
40	ADYDF	90	RESET
41	ADYDF	91	RESET
42	ADYDF	92	RESET
43	ADYDF	93	RESET
44	ADYDF	94	RESET
45	ADYDF	95	RESET
46	ADYDF	96	RESET
47	ADYDF	97	RESET
48	ADYDF	98	RESET
49	ADYDF	99	RESET
50	ADYDF	100	RESET



Pin	Function	Pin	Function
1	AVCC	51	RESET
2	AREF	52	RESET
3	ADSC	53	RESET
4	ADIF	54	RESET
5	ADYGO	55	RESET
6	ADYDF	56	RESET
7	ADYDF	57	RESET
8	ADYDF	58	RESET
9	ADYDF	59	RESET
10	ADYDF	60	RESET
11	ADYDF	61	RESET
12	ADYDF	62	RESET
13	ADYDF	63	RESET
14	ADYDF	64	RESET
15	ADYDF	65	RESET
16	ADYDF	66	RESET
17	ADYDF	67	RESET
18	ADYDF	68	RESET
19	ADYDF	69	RESET
20	ADYDF	70	RESET
21	ADYDF	71	RESET
22	ADYDF	72	RESET
23	ADYDF	73	RESET
24	ADYDF	74	RESET
25	ADYDF	75	RESET
26	ADYDF	76	RESET
27	ADYDF	77	RESET
28	ADYDF	78	RESET
29	ADYDF	79	RESET
30	ADYDF	80	RESET
31	ADYDF	81	RESET
32	ADYDF	82	RESET
33	ADYDF	83	RESET
34	ADYDF	84	RESET
35	ADYDF	85	RESET
36	ADYDF	86	RESET
37	ADYDF	87	RESET
38	ADYDF	88	RESET
39	ADYDF	89	RESET
40	ADYDF	90	RESET
41	ADYDF	91	RESET
42	ADYDF	92	RESET
43	ADYDF	93	RESET
44	ADYDF	94	RESET
45	ADYDF	95	RESET
46	ADYDF	96	RESET
47	ADYDF	97	RESET
48	ADYDF	98	RESET
49	ADYDF	99	RESET
50	ADYDF	100	RESET