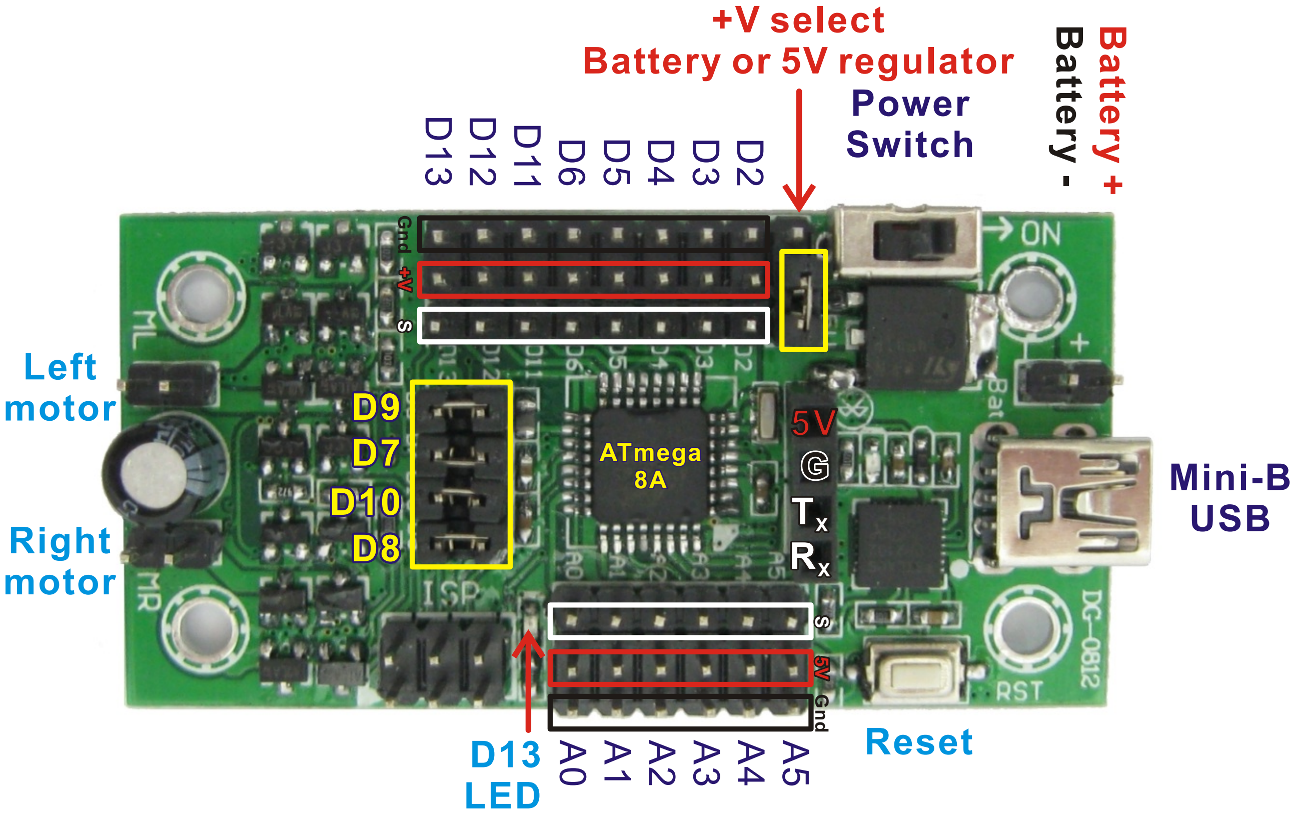
Mini Driver

Instruction Manual



Introduction:
The Mini Driver is a low cost controller that is software compatible with the Arduino NG. With it's built in dual FET "H" bridge and the ability to drive servos directly from the battery it makes a great motor or servo driver. Using code built into the Arduino IDE, the Mini Driver can also be used as an ISP programmer for re-burning bootloaders or bypassing the bootloader completely.

Programming:
The Mini Driver can be programmed using the Arduino IDE with the board type set for "Arduino NG or older w/ ATmega8". The built in USB interface allows the controller to be programmed using a standard "A" to mini "B" USB cable. Drivers for the CP2102 interface can be downloaded from here:
http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx

Wireless Communications:
The serial I/O pins D0 and D1 are broken out into a 4 pin female header with +5V and Gnd. This allows wireless transceivers such as an Xbee or Bluetooth module to be connected using a voltage translation interface. Deviced connected to D0 and D1 must be removed for a new program to be uploaded using the USB interface.

Power:
The Mini Driver is a low voltage device. Input voltage should be between 5V and 9V. Input voltages higher than 9V may damage the controller. A reverse polarity protection circuit will prevent damage if the battery is connected the wrong way however short circuit currents higher than 5.8A can cause permanent damage. Maximum current from the 5V regulator should be limited to 500mA.

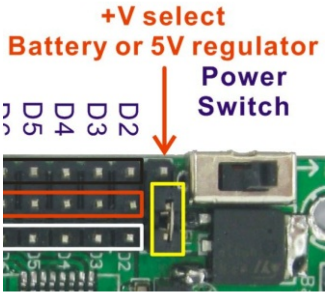
Dual FET "H" Bridge:
A dual "H" bridge allows the Mini Driver to control two D.C. brushed motors with a stall current as high as 2.5A per motor. When not required, the dual "H" bridge can be isolated from the controller by removing 4 jumpers. This then makes digital pins D7, D8, D9 and D10 available for other uses.

Connect your left motor to the two pin male header ML and the right motor to MR.The motors are always powered directly from the battery. D7 controls the direction of the left motor. D9 is the PWM pin for the left motor. D8 controls the direction for the right motor, D10 is the PWM pin for the right motor. If your motor turns in the wrong direction then swap the connections on the ML or MR headers.

Digital Pins and Servos:
Digital pins D2, D3, D4, D5, D6, D11, D12 and D13 are terminiated with servo compatible 3 pin male headers. The V+ rail for these pins can be set to +5V or +Battery.

When V+ is set to +5V, digital sensors such as ultrasonic range finders can be powered from the 5V regulator. Small servos may also be powered from the 5V regulator as long as they do not draw too much current.

When V+ is set to +Bat, servos can be powered directly from the battery. If you need to power +5V devices as well as servos then power the +5V device using the power pins next to the analog inputs.



Analog Pins:
Analog pins A0 to A5 are also terminated with a 3 pin header for powering sensors. The +V rail is always at +5V for powering sensors. A6 is terminated as a small solder pad on the back of the PCB while A7 is hard wired into a voltage monitor. Battery voltages as high as 10V can be read from A7.

Analog Pins as Digital Pins:
Analog pins A0 - A7 can also be used as digital pins D14 to D21 although D21 should not be used as it is hardwired as a voltage monitor. When used as digital pins, servos can be connected to D14-D19 if necessary.

Pullup Resistors:
All Arduino controllers using ATmega processors have internal 20K pullup resistors on the pins which can be activated by writing a digital 1 (HIGH) to that pin while the pin is configured as an input pin. This is ideal for reading switches connected between the input pins and Gnd.

Special Function Pins:
As with all Arduino controllers, some of the Mini Driver pins have dual functions which can be read about more in the Arduino reference text.

- D0 & D1 are the serial port RX and TX pins.
- D2 & D3 are the external interrupt pins.
- D9, D10 & D11 are PWM capable pins (dissabled by the Servo library).
- D10, D11, D12 and D13 connect to the ISP socket. D13 is output by default.
- A4 & A5 are the SDA and SCL pins for the I2C interface.

Using the Mini Driver as a Programmer:
A tutorial explaining how to use an Arduino as a programmer for burning or bypassing bootloaders can be found here:
http://letsmakerobots.com/node/35649

Please note the tutorial was written using Arduino 1.02 but may not work in later versions. The Arduino IDE is not backward compatible.