

ECROS Technology

www.ecrostech.com

Dragon Rider User's Guide

Preliminary, Revision D

March 2, 2008

This document is Copyright © 2008, ECROS Technology, All Rights Reserved.
Atmel and AVR are registered trademarks of Atmel Corporation. All other products
and/or service names mentioned in this user manual may be trademarks of the
companies with which they are associated.

Table of Contents

Introduction.....	1
General Description and Overview.....	1
Cautions (Please Read This Section).....	2
Parts and Assembly.....	3
Sourcing the Parts.....	3
General Assembly Tips.....	4
40-Pin Target Microcontrollers.....	6
Crystal Oscillator and Reset.....	7
Analog Supply and Reference Voltage.....	7
ISP (In-System Programming).....	8
JTAG Programming and Debugging.....	8
28-Pin Target Microcontrollers.....	9
Crystal Oscillator and Reset.....	9
Analog Supply and Reference Voltage.....	9
ISP (In-System Programming).....	10
debugWIRE.....	10
20-Pin Target Microcontrollers.....	10
Crystal Oscillator and Reset.....	11
ISP (In-System Programming).....	11
debugWIRE.....	12
8-Pin Target Microcontrollers.....	12
Crystal Oscillator and Reset.....	12
ISP (In-System Programming).....	13
debugWIRE.....	13
High-Voltage Programming.....	13
Add-on Circuits.....	15
Auxiliary Power Supply Add-on Circuit.....	15
Serial Interface Add-on Circuit.....	16
2-Line by 16-Character LCD Add-on Circuit.....	18
LED and Switch Add-on Circuit.....	18

Introduction

Thank you for your purchase of an ECROS Technology Dragon Rider.

The purpose of this User's Guide is to make sure that you fully understand the Dragon Rider and how it interacts with the AVR Dragon and the target microcontroller so that you get maximum benefit from owning this combination of products.

General Description and Overview



Photo 1.- The Dragon Rider 500, Assembled, with all Add-ons

Photo 1 shows the Dragon Rider 500, fully assembled, and with all Add-on circuits installed. The AVR Dragon is clearly visible underneath (the USB connector faces to the left). Although it may appear to be hanging, the pillars supporting the Dragon Rider and the feet stuck to the underside of the AVR Dragon securely hold both in position on a flat working surface.

Four IC sockets are provided for 40-, 28-, 20 and 8-pin target microcontrollers. As with the Atmel STK500, only one can be used at a time. Later sections in this Guide go into detail on each type of target. The General Purpose Input/Output (GPIO) ports of the target devices are connected to four 10-pin headers along the front edge of the Dragon Rider. These have the same pin-out as the GPIO headers of the STK500 and can be linked by 10-way ribbon cables to external circuits in exactly the same way. The microcontroller can use its internal oscillator or a crystal, which can be seen above and in front of the Dragon's USB jack. Behind this is the reset push-button. Target power is taken from the USB port through the AVR Dragon via a slide switch near the front in the middle. The analog supply, for targets with A-to-D converters, is decoupled from the digital supply and a precision voltage reference can be fitted.

Unlike the STK500, ISP programming connections are made by placing jumpers onto headers, rather than by a 6-way ribbon cable. For the targets supporting this, JTAG programming and debugging connections are also made with jumpers. High-Voltage Programming requires ribbon cables, like the STK500.

28-, 20 and 8-pin target devices share pins between GPIO and the special functions of reset and the crystal oscillator. On the STK500, these are hard-wired to the special functions, but, on the Dragon Rider, you can choose what you do with these pins by placing jumpers on headers.

Three peripheral functions of the STK500 also appear on the Dragon Rider as "Add-on" circuits. A bank of 6 LEDs is provided (not 8, but are 8 really necessary?) and five push buttons (again, not 8, but arranged in a Left/Right Up/Down diamond plus Enter). These can be linked to GPIO ports by ribbon cables. The third function is a serial port for connection of the target to a terminal, PC or other system. In addition, the Dragon Rider can be equipped with a 2-line by 16-character LCD, also linked to a GPIO port by a ribbon cable. Finally, an auxiliary power supply allows the target to be powered from an external power pack independent of the AVR Dragon.

Cautions (*Please Read This Section*)

You probably won't want to read right to the end of this document before you put your Dragon Rider together and start using it. Having "Cautions" in the first section of the User's Guide may not present a product in the most attractive light, but some words of advice are gathered here so that you won't regret your enthusiasm. Please bear these following points in mind as you work with your Dragon Rider.

If you are experienced at electronic assembly, you will not need to read the section on assembling the Dragon Rider. However, note that the four sockets that mate with the headers on the AVR Dragon need to go on the bottom of the board. This means that they must be soldered on the top. If you leave them until last, *you may find you don't have much room to work*. Consider assembling these sockets before the JTAG, ISP and High Voltage headers.

Where power enters the Dragon Rider from the AVR Dragon, a footprint has been placed for a regulator device, U6. In most cases, you will *install a wire link* between the top two holes (closer to U4) to directly connect the incoming power to the switch SW2. The purpose of the regulator is allow the target microcontroller to be powered from a lower voltage than the nominal 5 volts coming in from the Dragon. Should you need this feature, details are given below.

The auxiliary power supply add-on circuit can add a lot of flexibility to the Dragon Rider. You can run your target circuit while the Dragon is off doing something else, use more power than the Dragon can supply or even use a separate supply for relays or circuits requiring a different voltage. However, *the target area supply should not be connected to the Dragon and the auxiliary power supply at the same time*. Do not, therefore, ever close SW1 and SW2 at the same time (at least, when the Dragon and the external power pack are connected). If you think that this might happen, take a look at the section on the auxiliary power supply add-on circuit, below, for some suggestions, such as using a schottky diode in series with power from the Dragon.

R3 and U5 implement a precision analog voltage reference for use with the Analog-to-Digital converters in the ATmega48 family, ATmega16, etc. (details below). The accuracy is much better than the references in the microcontrollers themselves, but once these components are in place, the internal voltage reference can no longer be used and *must not be selected by software*. This is because the reference voltage appears at the AREF pin and will be in conflict with the externally applied voltage. The decoupling capacitor, C8, however, should always be installed.

When the AVR Dragon is tucked away under the Dragon Rider, it is easy to forget

that it is there. The jumpers make it very easy to connect the Dragon to your target, but you must still make sure that the connections are appropriate to your use of the target I/O ports. Don't allow the Dragon to get into conflict with something else you have connected to a port or *damage to the Dragon may result*. This is no different from using the Dragon with ribbon cables as an external programmer /debugger, except that it is hidden underneath. Atmel does not publish much about the Dragon's design, so it is hard to know what might damage it or how it might be protected without interfering with its function. Read and understand the Atmel documentation for the AVR Dragon. Remove all connections to the I/O ports used for programming or debugging before using them, or be sure that the connection is safe. For example, the LED Add-on circuit can remain connected to a port used by the Dragon for programming because of the 33k Ω resistors that limit the current that can be drawn. But, the LCD Add-on should be disconnected from a port used by the Dragon as it is able to drive signals onto the port bits and may conflict with signals driven by the Dragon. The sections, below, on each type of target microcontroller identify the port bits used by the Dragon for each programming / debugging interface type.

Parts and Assembly

Sourcing the Parts

Kits of parts for all Dragon Rider functions are available, but in most cases you will also be able to procure the parts yourself or find them in your spares drawer.

It is by no means necessary to install all the circuitry shown in Photo 1. You can, for example, use the Dragon Rider just for ISP programming and debugWIRE debugging of the 28-pin ATmega88 microcontroller at first. Other target sockets, High Voltage Programming and the Add-on circuits can be added later. This Guide aims to help you understand the Dragon Rider sufficiently to adapt and use it as best suits you.

ECROS Technology Parts Kits

For your convenience, all the parts needed to assemble the Dragon Rider are available from ECROS Technology in Parts Kits. To keep the cost down, the basic functions of the Dragon Rider are available using the components in the Basic Parts Kit. For the more advanced functions, that not everyone will need, purchase the Advanced Parts Kit in addition. The least expensive way to get up-and-running with the exploration of the AVR Dragon's capabilities is with the Quick Start Parts Kit, which includes everything you need for 28-pin targets, including the microcontroller itself, plus the LED and Switch Add-on Parts Kit. For more information about the Parts Kits, see the Web site at <http://www.ecrostech.com/AtmelAvr/DragonRider/Assembly.htm>

Using Your Own Parts

Wherever feasible, the Dragon Rider has been designed without the use of esoteric or hard-to-find components. Headers, sockets, resistors, capacitors, crystal, switches, etc. are all the sorts of things you may have in your spares drawer or will be able to buy from your favorite electronic parts supplier. This section offers some advice on the few exceptions where you might encounter difficulties.

The Dragon Rider and the AVR Dragon are supported on a work surface at the right spacing by support pillars (for the Dragon Rider) and “bumpon” feet (for the AVR Dragon). Obviously, the height of the pillars and feet have been carefully chosen and

not just any old parts will do. Also, a light pipe allows you to see the Dragon's Status LED from the top of the Dragon Rider. These parts are not totally necessary and you could get by without them. If you want to use them and have difficulty finding them, consider purchasing a Basic or Quick Start parts kit. You can use your own parts for the additional functionality of the advanced kit and add-on circuits.

The LCD module is of an unusually compact design and may be difficult to procure. In addition to this, the LCD Add-on parts kit contains the pillars, washers, nuts and screws to securely support the module above the PCB.

General Assembly Tips

This section contains some tips on assembling the Dragon Rider, either from the parts in the ECROS Technology kits or from your own components. If you are experienced at electronic assembly work, you will want to skip or just skim over this section, but take note of the first paragraph of "Planning Ahead", below. If you are a beginner at this sort of thing, please take the time to do a little reading-up first and practice on a few expendable spare parts before you start. What follows is not intended to take you from the point of never having soldered before. Dragon Rider printed circuit boards are made using the same processes as commercial and industrial electronic goods and won't fall apart on you like cut-rate hobbyist PCBs, but to squeeze all the features into a modest area the widths of traces and the sizes of pads are smaller than the typical "starter" electronic project. The gold finish (yes, it's real gold) prevents oxidation, so you can store the board and not run into difficulty, but it is very slightly less willing to take the solder than a tin/lead finish and requires proper soldering technique.

Equipment Needed

The Dragon Rider requires only moderate skill to assemble. You should use a good-quality soldering iron with a fine tip and thin, flux-cored solder specifically designed for electronic work. You will also need a small pair of "flush" bladed side-cutters, also designed for electronic work. The "flush" means that the cutting edge is all the way to one side of the blade, so that, by laying the blade against a workpiece, a wire or pin can be cut off entirely level with the surface.

Planning Ahead

There is one difficulty to be addressed, which is mounting the header sockets on the underside of the board while there is room to solder their pins on the topside. Do not leave the header sockets to last; instead mount them before the headers for High Voltage, ISP and JTAG programming.

The suggested order of assembly, below, is simply to mount parts in order of height from the board. This allows you to turn the board upside-down to hold components in place for soldering. If you have the Advanced Parts Kit, don't put it to one side and assemble the Basic parts first; assemble both kits at the same time. Be sure to protect your work surface from heat and abrasion; a piece of card will do this nicely.

Resistors and Capacitors

Begin with the resistors R1 to R4 and the axial capacitors C1, C2 and C5 to C9. Bend the leads at right angles to match the pitch of the holes in the board and pass them through. On the back side, bend them sharply at 45 degrees to hold the part in place while you repeat this process for the other components. Turn the board over and

solder the leads. Snip off the excess flush with the board. Holding the blades of your cutters flat to the board, rather than facing the tip sharply down, will avoid damaging the printed circuit traces.

The Reset Switch

If you are using the ECROS Technology Parts Kit, the reset switch, SW0, is shorter than the IC sockets and can be assembled next. However, as it is a snap fit and holds itself in place, it can actually be assembled at any time. If you are using your own parts, don't assemble the reset switch at this point if it is taller than the IC sockets.

IC Sockets

The four target IC sockets, U1 to U4 will be next. Press down on the back of the board while you solder two pins in opposite corners first. Check that the socket is sitting flush against the board and re-melt the solder if necessary while closing any gap. Then proceed to solder the remaining pins and clip off the extra length of the pins.

Crystal and Load Capacitors

If you intend to use a crystal on the Dragon Rider, this is probably a good point at which to think about this. The crystal oscillator circuit consists of the load capacitors, C3 and C4, and the crystal or, preferably, a crystal socket. It isn't hard to add these parts even after everything else is assembled. Note that if you solder a crystal directly to the board, you may have problems with High Voltage programming, when the AVR Dragon wants to supply the target clock signal. Soldering down the crystal is completely safe, however, if you will only use ISP, JTAG and/or debugWIRE. Of course, if you want to be able to change the crystal frequency, a socket is necessary. It may be cheaper to make one by cutting down a machined-pin IC socket than to buy a purpose-designed crystal socket. This is actually what you will find in the Advanced Parts Kit.

Headers and Header Sockets

Next, assemble the headers on the left half of the board, J1 to J13. Do not at this time mount headers J17 to J28 or it may be difficult to solder the header sockets, which mount on the bottom of the board.

Before finishing the topside headers, mount the header sockets that connect to the AVR Dragon onto the bottom of the board. These are not identified on the silkscreen, but it should be completely obvious where they go if you look at your AVR Dragon. There is a ten-pin socket (2 by 5) for JTAG, two six-pin sockets (2 by 3) for power and ISP and a twenty-pin socket (2 by 10) for High Voltage. Support the board so that the sockets, inserted from underneath, lie flush to the underside. Solder opposite corner pins, check and reposition, solder all pins and clip the leads. To be sure that everything will line up, consider sliding the header sockets onto your AVR Dragon's headers and placing the Dragon Rider on top. To solder pins while they are connected to your valuable AVR Dragon, you should be confident of the quality of your soldering iron, or you can unplug it and ground it while you do this. If you plan to use High Voltage programming and don't have the 20-pin header on your AVR Dragon, you will, of course, have to add it.

Precision Reference

The precision reference supplied in the Advanced Parts Kit as U5 will give you a far more accurate analog reference voltage than is available in the microcontroller. It is rated at 2.5 volts nominal with a maximum room temperature error of only 0.4% and a maximum drift of 90 parts per million per °C (13 typical). R3 sets the current to 2.5 mA (the minimum for full performance is 1 mA). Note that *once you have installed R3 and U5, you must not select an internal reference source in your software* as it will be shorted to the external reference.

Power Switch and Sub-Regulator

The power switch, SW2, being tall, should be one of the last parts to go on to the board. Just to the right of this switch, there is a footprint for the sub-regulator, U6, that few people will use. If you are not using a sub-regulator, that is if your target is powered directly from the AVR Dragon, install a wire jumper between the top two holes of the U6 footprint (closer to the IC U4). The bottom hole, closer to the Port B header, J2, is ground, so don't put the wire link in there! If you do wish to use the target power sub-regulator, it deserves some special attention. The footprint for U6 is squeezed onto the PCB in between the AVR Dragon power connector and the switch, SW2. The heatsink tab is to the left, as marked. But, if you actually put it here, it will almost surely be in the way. Most people are not expected to need sub-regulation (if you have the Advanced Parts Kit, put the regulator in your spares drawer), but if you do, consider mounting U6 *underneath* the Dragon Rider board. With the tab facing down and the pins bent up, it will be out of the way to the left of the 6-pin power connector.

Light Pipe and Mechanical Support

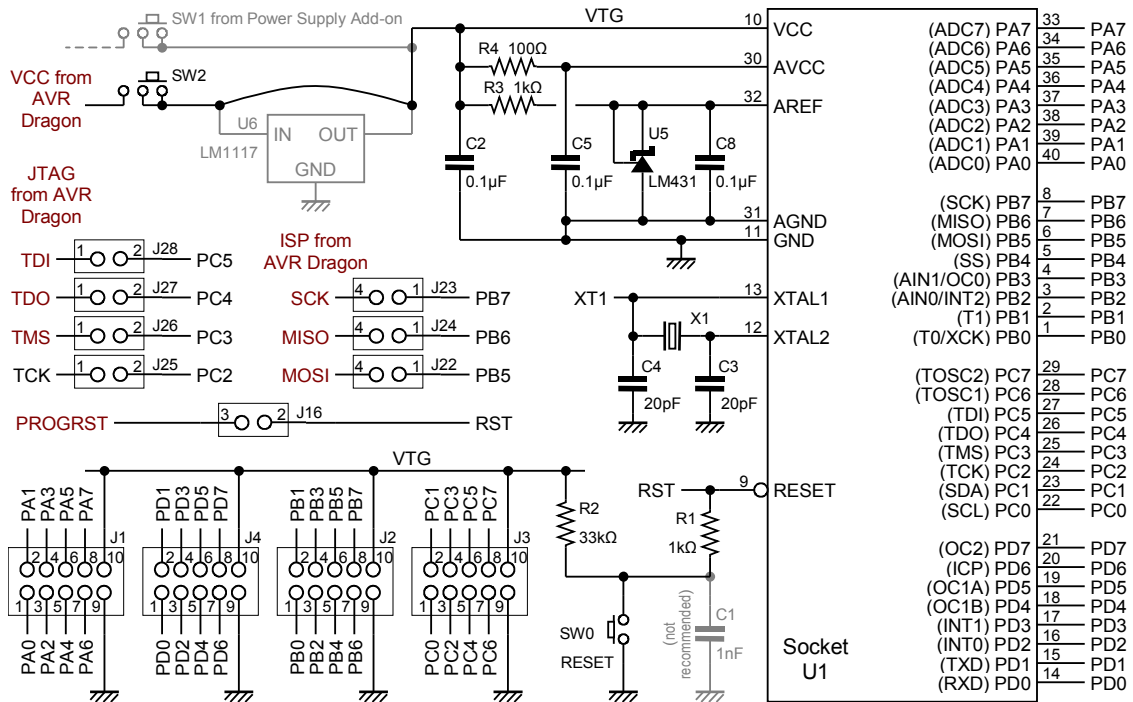
The finishing touches are the non-electrical parts. Push the light pipe through its hole (near the reset switch) from the top of the board and slide the ring over it on the bottom to hold it in place. Assemble the four pillars by placing a nylon washer over the threaded end, inserting this end through one of the four corner holes from the bottom of the board and securing it with a nylon nut (not too tight, it's only nylon). Place the bumpon feet on the bottom of your AVR Dragon in the four corners. When assembled to the Dragon Rider, both the AVR Dragon and the Dragon Rider will be supported on a flat work surface by the bumpons and pillars, respectively, when the headers are fully engaged in the sockets. The AVR Dragon's status LED will be visible from the top of the assembly through the light pipe. If you are using your own headers and sockets and they are of unusual height, you may have to make adjustments in the height of the pillars and/or the feet applied to the AVR Dragon.

The corner pillars for the Dragon Rider have 4-40 tapped holes in the bottom. So, if you want a semi-permanent assembly, you can hold everything down to a base of some sort (such as Plexiglas) with four 4-40 machine screws. Even without such measures, the headers and sockets supplied in the Parts Kits hold the AVR Dragon to the Dragon Rider very securely and you can pick up and even shake the assembly without fear of the Dragon taking wing and flying away.

40-Pin Target Microcontrollers

This section focuses on working with 40-pin microcontrollers, such as the ATmega16 and ATmega32, in socket U1 of the Dragon Rider 500. The schematic below shows

how the 40-pin socket is interconnected, omitting details relevant only to the other target sockets. The pin functions shown refer to the ATmega16/32 and may differ for other 40-pin devices.



Crystal Oscillator and Reset

40-pin target devices have pins dedicated to the crystal oscillator and reset functions.

The crystal, X1, and load capacitors, C3 and C4, are directly wired to pins 12 and 13. It is recommended that a crystal socket be installed, rather than directly soldering in a crystal, if High Voltage Programming will be used. The AVR Dragon will need to drive the XT1 line and this is best done without a crystal. The load capacitor, C4, will increase the capacitance of this line, but this has not been found to cause problems.

The RESET switch, SW0, is wired to the reset line via a 1kΩ resistor, R1. The purpose of this resistor is to avoid damage to the AVR Dragon if the RESET switch is pressed while the Dragon is driving the RST line. When the RESET switch is not pressed, the input to R1 is pulled up to VTG via a 33kΩ resistor, R2. A footprint is provided on the Dragon Rider PCB for a capacitor, C1, to suppress glitches on the reset line. This has not been found necessary and it is not recommended that a capacitor be installed.

Analog Supply and Reference Voltage

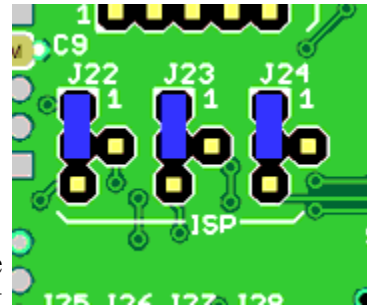
The analog supply voltage, AVCC at pin 30, is supplied from the target supply, VTG, via a 100Ω resistor, R4, and is decoupled with a 0.1μF capacitor, C5, to the common ground plane. For additional immunity from noise on VTG, install an inductor in place of R4 and wire a tantalum electrolytic capacitor in parallel with C5 (observing the correct polarity).

The analog reference voltage at pin 32 is decoupled with a 0.1μF capacitor, C8, to the common ground plane. This is effective when either AVCC or the internal reference of the target device is selected by software. An external precision voltage reference,

U5, can be installed to provide a more accurate voltage reference. The footprint on the PCB is for an LM431-type device in a TO-92 package, but other types may also be suitable. Resistor R3 provides bias current for this reference. A value of $1\text{k}\Omega$ will supply a bias of 2.5 mA to a 2.5 volt reference when VTG is 5 volts. The TL1431 supplied in the Advanced Parts Kit requires a minimum of 1 mA and has an accuracy of 0.4% ($\pm 10\text{ mV}$). For other situations, you may need to recalculate the value of this resistor. For example, if VTG is sub-regulated to 3.3 volts, the voltage across R3 is only 0.8 volts and a lower value, such as 470Ω , is needed to provide at least 1mA.

ISP (In-System Programming)

The AVR Dragon's ISP feature requires connections to the target microcontroller's RESET, SCK, MISO and MOSI lines. Make the RESET connection by installing the jumper on J16 between pins 2 and 3. Headers J22, J23 and J24 offer three possible connections of the SCK, MISO and MOSI lines to individual bits of Port B. The ATmega16 and ATmega32 microcontrollers require the jumpers on these headers to be between pin 4 (the middle pin) and pin 1 (the top pin) as shown in the figure at right.



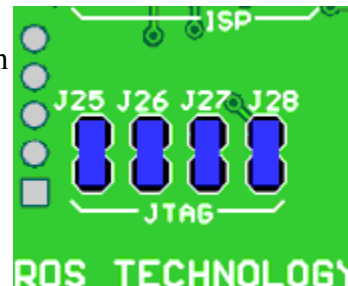
ISP Jumpers

When using ISP, remove the JTAG jumpers from J25, J26, J27 and J28. Make sure you do not have anything connected to the pins that are used for ISP that might result in a logic conflict with the AVR Dragon. For 40-pin targets, these pins are shared with GPIO port bits PB5, PB6 and PB7, so be careful what you connect to these bits of Port B at J2. If in doubt, disconnect.

ISP allows you to set and clear device fuses and program and erase the Flash and EEPROM memories. It is essential that the device's main clock be running. Do not program the clock fuses in a manner inconsistent with what is connected to the XTAL1 and XTAL2 pins.

JTAG Programming and Debugging

To use JTAG programming and debugging with 40-pin devices that support this feature, install jumpers on two-pin headers J25, J26, J27 and J28 as shown in the figure at right. Also, install the reset jumper on J16 between pins 2 and 3.



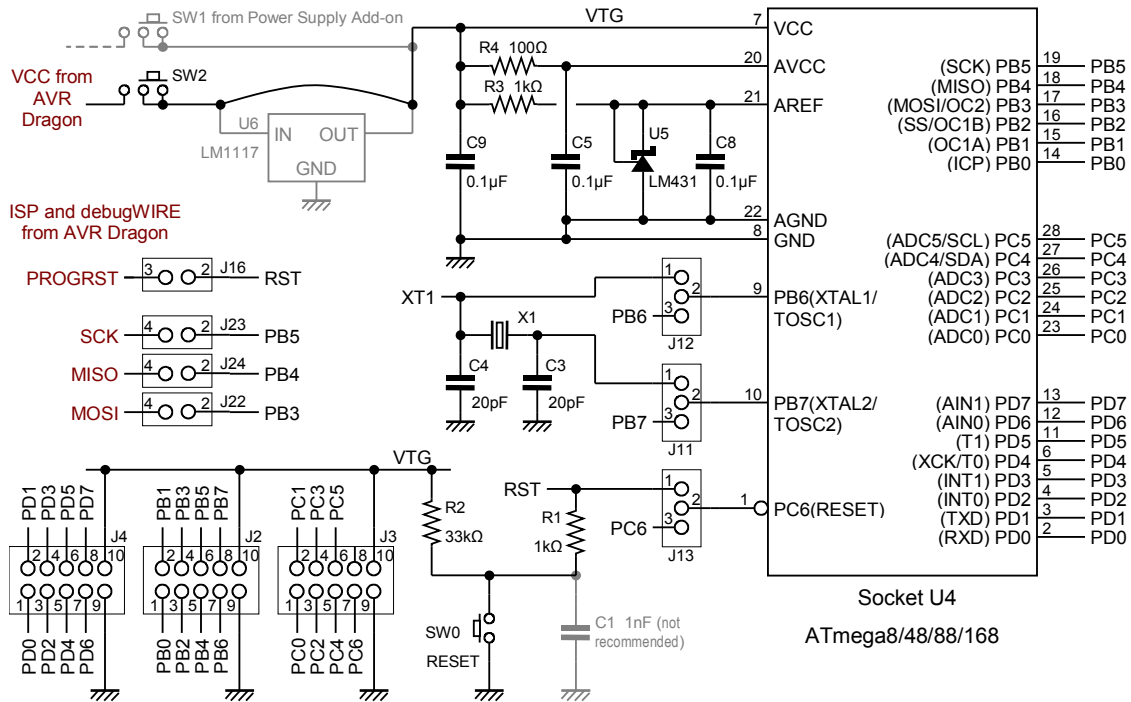
JTAG Jumpers

Make sure you do not have anything connected to the pins that are used for JTAG that might result in a logic conflict with the AVR Dragon. These pins are shared with GPIO port bits PC2, PC3, PC4 and PC5, so be careful what you connect to these bits of Port C at J3. If in doubt, disconnect.

Using JTAG, you can set and clear device fuses, program and erase the Flash and EEPROM memories and use on-chip debug features. There is no need to switch between different interfaces to perform these functions as is necessary with ISP and debugWIRE. In addition, to set fuses and program memories, it is not necessary for the microcontroller clock to be running. The JTAG interface provides its own clock. So, it is not possible to lock yourself out of your microcontroller by inappropriately setting the clock fuses, as is common with beginners using ISP.

28-Pin Target Microcontrollers

This section focuses on working with 28-pin microcontrollers, such as the ATmega8 and the ATmega48 family, in socket U4 of the Dragon Rider 500. The schematic below shows how the 28-pin socket is interconnected, omitting details relevant only to the other target sockets. The pin functions shown refer to the ATmega48 family and may differ for other 28-pin devices.



When using 28-pin targets, remove the JTAG jumpers from J25, J26, J27 and J28.

Crystal Oscillator and Reset

28-pin target devices have pins that are shared between the crystal oscillator and reset functions and normal input/output port functions.

To connect pins 9 and 10 to the crystal oscillator, place jumpers between pins 1 and 2 of both J12 and J11. A jumper in this position on J12 is also necessary to connect pin 9 to the XTAL1 output of the AVR Dragon for High Voltage Programming. To connect pin 9 to input/output port line PB6 at pin 7 of J2, place a jumper between pins 2 and 3 of J12. Similarly, to connect pin 10 to PB7 at pin 8 of J2, place a jumper between pins 2 and 3 of J11.

To connect pin 1 to the reset line, place a jumper between pins 1 and 2 of J13. This must be done for any of the AVR Dragon programming or debugging methods to be used (ISP, debugWIRE or High Voltage Programming). If you are not programming or debugging with the AVR Dragon, you can connect pin 1 to input/output port line PC6 at pin 7 of J3 by placing a jumper between pins 2 and 3 of J13.

Analog Supply and Reference Voltage

The analog supply voltage, AVCC at pin 20, is supplied from the target supply, VTG, via a 100Ω resistor, R4, and is decoupled with a 0.1μF capacitor, C5, to the common

ground plane. For additional immunity from noise on VTG, install an inductor in place of R4 and wire a tantalum electrolytic capacitor in parallel with C5 (observing the correct polarity).

The analog reference voltage at pin 21 is decoupled with a 0.1 μ F capacitor, C8, to the common ground plane. This is effective when either AVCC or the internal reference of the target device is selected by software. An external precision voltage reference, U5, can be installed to provide a more accurate voltage reference. For additional information on this option, please refer to the section on 40-pin microcontrollers.

ISP (In-System Programming)

The AVR Dragon's ISP feature requires connections to the target microcontroller's RESET, SCK, MISO and MOSI lines. Make the RESET connection by installing a jumper on J16 between pins 2 and 3. It is also necessary to install the reset jumper between pins 1 and 2 of J13, as described above, to connect the RESET line to pin 1. Headers J22, J23 and J24 connect the SCK, MISO and MOSI lines to individual bits of Port B. The ATmega8 and Atmega48 family of microcontrollers require the jumpers on these headers to be between pin 4 (the middle pin) and pin 2 (the rightmost pin) as shown in the figure at right.

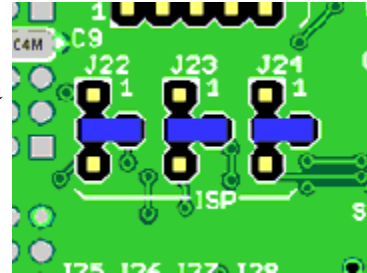


Photo 1 - ISP Jumpers.

Make sure you do not have anything connected to the pins that are used for ISP that might result in a logic conflict with the AVR Dragon. For 28-pin targets, these pins are shared with GPIO port bits PB3, PB4 and PB5, so be careful what you connect to these bits of Port B at J2. If in doubt, disconnect.

ISP allows you to set and clear device fuses and program and erase the Flash and EEPROM memories. It is essential that the device's main clock be running. Do not program the clock fuses in a manner inconsistent with what is connected to the XTAL1 and XTAL2 pins.

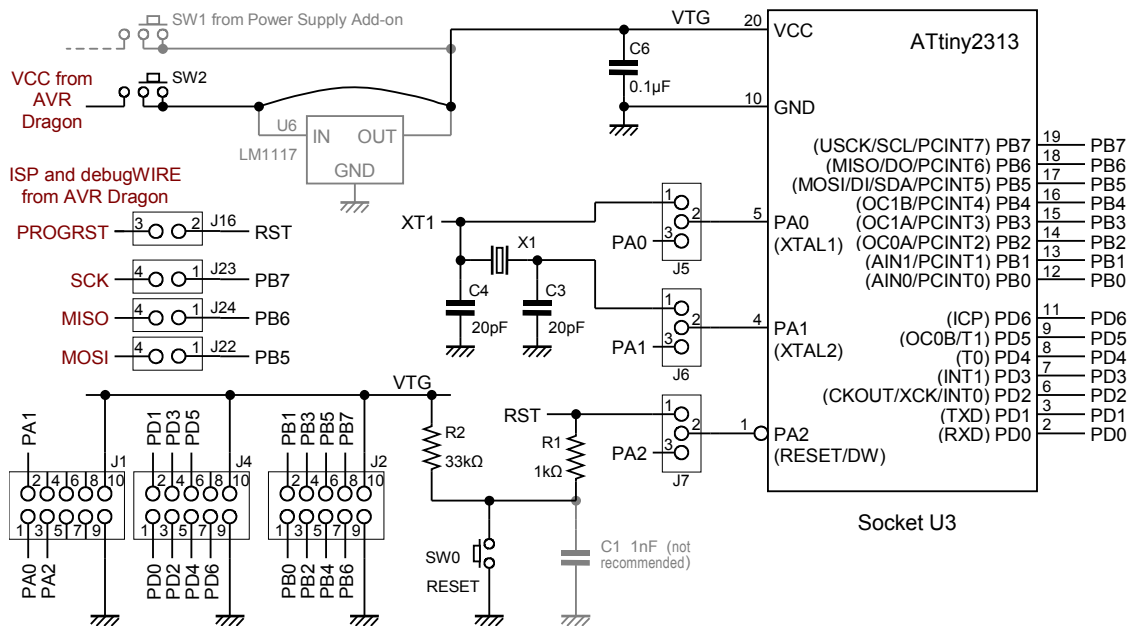
debugWIRE

The AVR Dragon's debugWIRE feature requires only a connection to the target microcontroller's RESET line. Make this connection by installing a jumper on J16 between pins 2 and 3. It is also necessary to install the reset jumper between pins 1 and 2 of J13, as described above, to connect the RESET line to pin 1.

debugWIRE allows you to erase and program the device's Flash memory and to debug program execution. You cannot set and clear device fuses with this interface; you must switch over to ISP to do this. If you are not using port bits PB3, PB4 or PB5 for any other purpose, you can place jumpers on J22, J23 and J24 as shown above to have the ISP interface available. Otherwise, you can remove the ISP jumpers.

20-Pin Target Microcontrollers

This section focuses on working with compatible 20-pin microcontrollers, such as the ATtiny2313, in socket U3 of the Dragon Rider 500. The schematic below shows how the 20-pin socket is interconnected, omitting details relevant only to the other target sockets. The pin functions shown refer to the ATtiny2313 and may differ for other 20-pin devices.

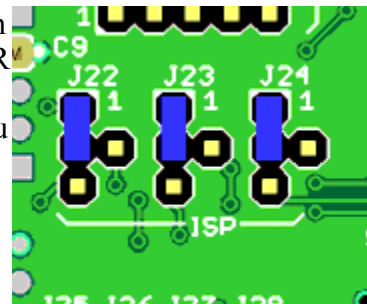


Crystal Oscillator and Reset

20-pin target devices have pins that are shared between the crystal oscillator and reset functions and normal input/output port functions.

To connect pins 5 and 4 to the crystal oscillator, place jumpers between pins 1 and 2 of both J5 and J6. A jumper in this position on J5 is also necessary to connect pin 5 to the XTAL1 output of the AVR Dragon for High Voltage Programming. To connect pin 5 to input/output port line PA0 at pin 1 of J1, place a jumper between pins 2 and 3 of J5. Similarly, to connect pin 4 to PA1 at pin 2 of J1, place a jumper between pins 2 and 3 of J6.

To connect pin 1 to the reset line, place a jumper between pins 1 and 2 of J7. This must be done for any of the AVR Dragon programming or debugging methods to be used (ISP, debugWIRE or High Voltage Programming). If you are not programming or debugging with the AVR Dragon, you can connect pin 1 to input/output port line PA2 at pin 3 of J1 by placing a jumper between pins 2 and 3 of J7.



ATtiny2313 ISP jumpers.

ISP (In-System Programming)

The AVR Dragon's ISP feature requires connections to the target microcontroller's RESET, SCK, MISO and MOSI lines. Make the RESET connection by installing the jumper on J16 between pins 2 and 3. It is also necessary to install the reset jumper between pins 1 and 2 of J7, as described above, to connect the RESET line to pin 1. Headers J22, J23 and J24 offer three possible connections of the SCK, MISO and MOSI lines to individual bits of Port B. The ATtiny2313 microcontroller requires the jumpers on these headers to be between pin 4 (the middle pin) and pin 1 (the top pin) as shown in the figure.

Make sure you do not have anything connected to the pins that are used for ISP that might result in a logic conflict with the AVR Dragon. For 20-pin targets, these pins are shared with GPIO port bits PB5, PB6 and PB7, so be careful what you connect to

these bits of Port B at J2. If in doubt, disconnect.

ISP allows you to set and clear device fuses and program and erase the Flash and EEPROM memories. It is essential that the device's main clock be running. Do not program the clock fuses in a manner inconsistent with what is connected to the XTAL1 and XTAL2 pins.

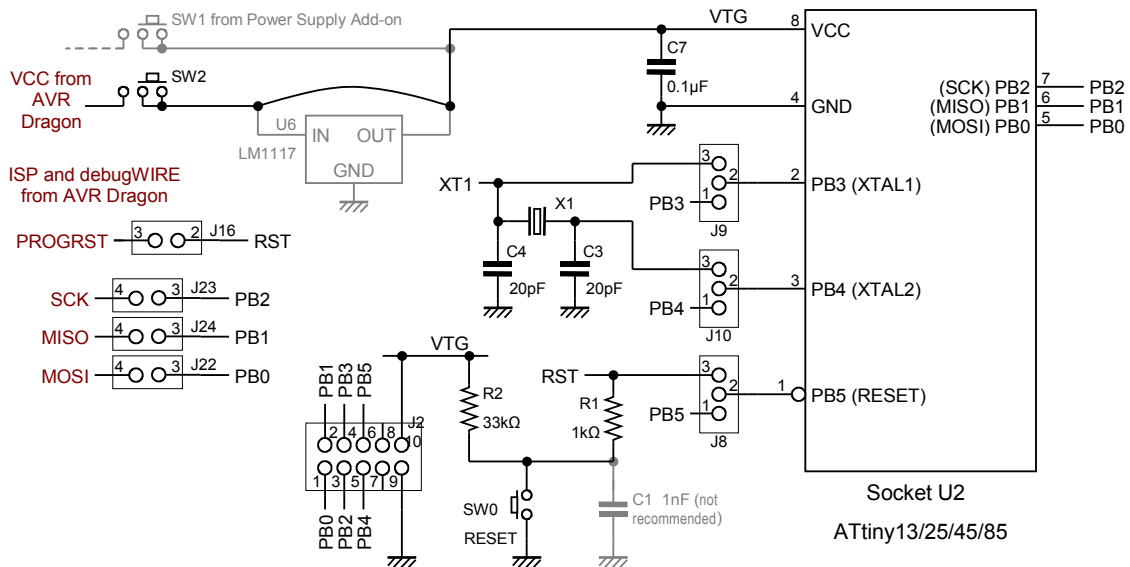
debugWIRE

The AVR Dragon's debugWIRE feature requires only a connection to the target microcontroller's RESET line. Make this connection by installing a jumper on J16 between pins 2 and 3. It is also necessary to install the reset jumper between pins 1 and 2 of J7, as described above, to connect the RESET line to pin 1.

debugWIRE allows you to erase and program the device's Flash memory and to debug program execution. You cannot set and clear device fuses with this interface; you must switch over to ISP to do this. If you are not using port bits PB5, PB6 or PB7 for any other purpose, you can place jumpers on J22, J23 and J24 as shown above to have the ISP interface available. Otherwise, you can remove the ISP jumpers.

8-Pin Target Microcontrollers

This section focuses on working with 8-pin microcontrollers, such as the ATtiny13 and the ATtiny25 family, in socket U2 of the Dragon Rider 500. The schematic below shows how the 8-pin socket is interconnected, omitting details relevant only to the other target sockets. The pin functions shown refer to the ATtiny25 and may differ for other 8-pin devices.



Crystal Oscillator and Reset

8-pin target devices have pins that are shared between the crystal oscillator and reset functions and normal input/output port functions.

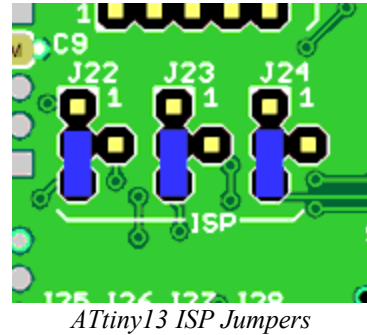
To connect pins 2 and 3 to the crystal oscillator, place jumpers between pins 2 and 3 of both J9 and J10. A jumper in this position on J9 is also necessary to connect pin 2 to the XTAL1 output of the AVR Dragon for High Voltage Programming. To connect pin 2 to input/output port line PB3 at pin 4 of J2, place a jumper between

pins 1 and 2 of J9. Similarly, to connect pin 3 to PB4 at pin 5 of J2, place a jumper between pins 1 and 2 of J10.

To connect pin 1 to the reset line, place a jumper between pins 2 and 3 of J8. This must be done for any of the AVR Dragon programming or debugging methods to be used (ISP, debugWIRE or High Voltage Programming). If you are not programming or debugging with the AVR Dragon, you can connect pin 1 to input/output port line PB5 at pin 6 of J2, by placing a jumper between pins 1 and 2 of J8.

ISP (In-System Programming)

The AVR Dragon's ISP feature requires connections to the target microcontroller's RESET, SCK, MISO and MOSI lines. Make the RESET connection by installing the jumper on J16 between pins 2 and 3. It is also necessary to install the reset jumper between pins 2 and 3 of J8, as described above, to connect the RESET line to pin 1. Headers J22, J23 and J24 offer three possible connections of the SCK, MISO and MOSI lines to individual bits of Port B. The ATtiny13 and ATtiny25 family microcontrollers require the jumpers on these headers to be between pin 4 (the middle pin) and pin 3 (the bottom pin) as shown.



Make sure you do not have anything connected to the pins that are used for ISP that might result in a logic conflict with the AVR Dragon. For 8-pin targets, these pins are shared with GPIO port bits PB0, PB1 and PB2, so be careful what you connect to these bits of Port B at J2. If in doubt, disconnect.

ISP allows you to set and clear device fuses and program and erase the Flash and EEPROM memories. It is essential that the device's main clock be running. Do not program the clock fuses in a manner inconsistent with what is connected to the XTAL1 and XTAL2 pins.

debugWIRE

The AVR Dragon's debugWIRE feature requires only a connection to the target microcontroller's RESET line. Make this connection by installing a jumper on J16 between pins 2 and 3. It is also necessary to install the reset jumper between pins 2 and 3 of J8, as described above, to connect the RESET line to pin 1.

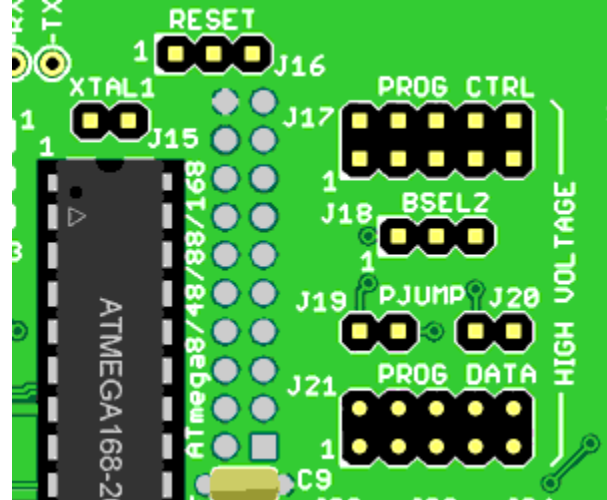
debugWIRE allows you to erase and program the device's Flash memory and to debug program execution. You cannot set and clear device fuses with this interface; you must switch over to ISP to do this. If you are not using port bits PB0, PB1 or PB2 for any other purpose, you can place jumpers on J22, J23 and J24 as shown above to have the ISP interface available. Otherwise, you can remove the ISP jumpers.

High-Voltage Programming

Using the High-Voltage Programming (HVP) interface to Atmel microcontrollers is in most cases less convenient than ISP or JTAG. The target supply voltage is much more important and more device pins must be under control of the AVR Dragon. There is one advantage of HVP over ISP, and that is that the device clock does not need to be running because an externally applied clock is used. So, if the clock fuses are set incorrectly, for example to expect an external clock, then the device can be

“rescued” using HVP to repair the fuse settings. The JTAG interface also applies its own clock and so can more conveniently be used to rescue devices that have JTAG interfaces.

The Dragon Rider adapts the AVR Dragon's 10-pin HVP header to the pair of 10-pin headers used on the Atmel STK500. This allows 10-way ribbon cables to be used to connect up the appropriate port bits. The RESET and XTAL1 headers of the STK500 that divert these signals for HVP are also provided. Finally, the BSEL2 and PJUMP headers route the HVP signals to microcontrollers that do not have full-width ports for the data and control signals. As a result, using the Dragon Rider and AVR Dragon for HVP is similar to using the STK500. The figure here shows the location of the HVP headers on the Dragon Rider.



High-Voltage Programming Header Area

Add-on Circuits

Auxiliary Power Supply Add-on Circuit

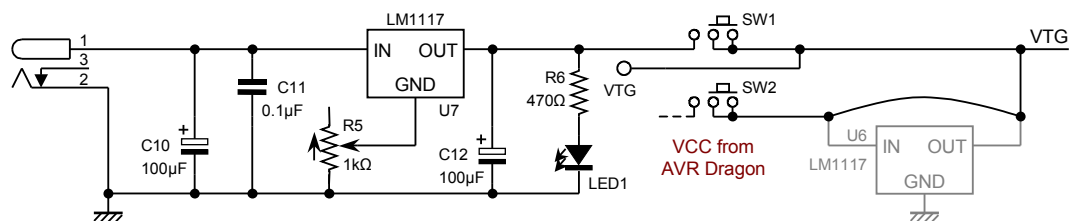
The Dragon Rider PCB includes an area for a power supply circuit that allows some or all of the circuitry on the board to be powered independently of the AVR Dragon.

Important – Please Read This Paragraph

While the auxiliary power supply has several benefits, it also presents certain hazards that you must be careful to avoid. First, never power the target microcontroller from both the AVR Dragon and the auxiliary power supply together. In other words, never have both SW1 and SW2 on at the same time if the AVR Dragon is attached. If you think that this might happen, you can protect the AVR Dragon by installing a schottky diode in the footprint of U6 instead of the usual wire link. (If you have installed U6 here as a sub-regulator, it will act to protect the AVR Dragon.) Remember that you will lose a small voltage, about 0.3 volts, across the diode. Second, if you install R5 to make the supply adjustable, be careful that you do not use it to power the target microcontroller when it is set to a higher voltage than the target device is rated for. Check the output voltage at pin 1 of the header J34 (which is normally not installed) when you are making adjustments before closing SW1.

An Adjustable Voltage Supply

The schematic below shows all available part footprints of the auxiliary power supply. If fully assembled, R5 adjusts the output voltage. This is connected to VTG, which is the Dragon Rider target area supply, when switch SW1 is closed. LED1 lights when the external power source is attached to J32, whether or not SW1 is closed, and can serve as a warning that SW2 should not be closed if the AVR Dragon is attached.



A typical choice of regulator device for U7 would be the LM1117-3.3V, which is the device supplied in the ECROS Technology parts kit. With component values shown, the voltage is adjustable from 3.3 volts to about 7 volts. It is important not to set the voltage above 5.5 volts when SW1 is closed to power the target area. Other devices in the LM1117 family are also suitable for this circuit and can be chosen to allow the voltage to be set lower for Atmel microcontrollers that run below 3.3 volts.

A 3.3 or 5 Volt Fixed Supply

If you want a fixed voltage auxiliary power supply, it may be better not to install R5 so as to remove the possibility of setting the voltage too high. For a 3.3 volt supply, simply put a wire link between the wiper of R5 and ground. On the Dragon Rider PCB, the wiper has a trace on the underside running to the regulator, U7. The ground pin is square. The third pin is not connected. For a 5 volt supply, you can use the 5 V device from the LM1117 family. Alternatively, the 3.3 V device can provide close to 5 V if the wiper of R5 is linked with a wire to the junction of R6 and LED1. This

works because the voltage across LED1 is about 1.7 volts and $1.7 + 3.3 = 5.0$.

Using the Auxiliary Supply

With SW1 installed, the target area of the Dragon Rider and any connected circuits can be powered either from the AVR Dragon or the auxiliary supply by closing either SW2 or SW1, respectively. Never close both when the AVR Dragon is connected. This allows you to power up and use your project without the Dragon, freeing it for use elsewhere. Alternatively, with the Dragon connected, you can power your project separately so that you can use a different supply voltage from the Dragon or you can draw more current than the Dragon is able to source. If you always want power to come from the auxiliary supply, you can remove or not install SW2. Note, however, that during High-Voltage Programming, the Dragon expects to be powering the target microcontroller.

If you do not install SW1, the output of the auxiliary supply can be used to power parts of the project other than the microcontroller. For example, if you are switching relays it is often a good idea to provide separate power for the relay coils to avoid the introduction of noise into the logic supply. The voltage of the auxiliary supply may be different from the AVR Dragon supply to the microcontroller, assuming that the circuit is properly designed for this. Perhaps the relays need a 6 volt supply. You can pick up power from the auxiliary supply at the topmost hole of the footprint of SW1, near the word "ON" on the silkscreen.

Selecting an External Power Source

Using an LM1117 family regulator, the auxiliary power supply is designed to work with an unregulated DC input of no more than 15 volts. Other regulators, such as the LM317 family, allow higher input voltages, but adjustments elsewhere may be needed to use such a device. The footprint of J32 fits a standard coaxial DC jack that will mate with plugs typically found on "wall-wart" type power packs. It is important that the plug be wired "center positive", i.e. the hole in the middle of the plug has positive polarity and the sleeve around the outside negative. The jack supplied in the ECROS Technology parts kit has a 2.1 mm center pin. This is suitable for many wall-warts. In fact, wall-warts designed for a 2.5 mm center pin seem to work just fine. You can of course wire the external power source to the Dragon Rider PCB without a plug and socket using the holes into which J32 would otherwise fit.

An LM1117 family regulator has a "drop-out" voltage of 1.2 volts. The external power source therefore needs to produce this much voltage above your intended output voltage. For example, when set to output 5 volts, the regulator should have an input of at least 6.2 volts. A wall-wart with a nominal output voltage of 7.5 volts would be a good choice, although one rated for 6 volts would work if used below its maximum current. A 9 volt wall-wart would also work. You should check a 12 volt wall-wart to make sure that it does not produce more than 15 volts at the load you are applying.

Serial Interface Add-on Circuit

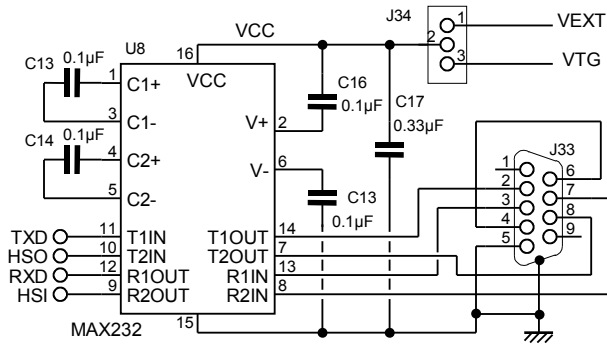
The Dragon Rider PCB includes an area for a serial interface, consisting of a level converter IC and a board-mounted, female DB9 connector. Using this, the target microcontroller's built-in UART can be linked to a PC "COM" port, terminal, modem or to other systems for control, monitoring and logging. A no-compromise design allows data rates up to the highest supported by the target and the cable or data

transmission system.

The figure at right is the circuit diagram of the serial interface. The serial interface circuit is independent of all other circuitry on the Dragon Rider, having only the ground connection in common, until you connect it up. Connections are made directly with wire links.

Power to the level converter IC (U8)

is supplied by pin 2 (the middle pin) of J34. Normally, you will link this to target system power at pin 3. Exceptionally, you may want to power the converter from the external power supply add-on circuit, perhaps to avoid any possibility of noise being injected from the charge-pumps in the level converter. You can do this by linking pins 1 and 2 of J34. Although no header and shunt are supplied in the kit, you can install a 3-pin header as J34 and keep your options open.



Transmit Data (TXD) and Receive Data (RXD) connections to the level converter are available at labeled pads near the bottom left corner of the LCD footprint. TXD passes through a transmit channel of the converter and appears at pin 2 of the DB9, which is RXD of the connected COM port device. TXD of the connected device arrives at pin 3 of the DB9, passes through a receive channel of the converter and appears at the RXD pad. Thus, a straight-through cable should be used between the Dragon Rider and the male COM port connector of a PC.

To connect TXD and RXD of the serial interface circuit to Port D bits 1 and 0, respectively, of the target microcontroller, simply solder wire links between the pads above and below the TXD and RXD labels on the Dragon Rider silk-screen. This turns out to be the connection needed to use the built-in UART of many of the compatible AVR devices. To use the interface with a device that does not have a UART at these port bits, or to use a software UART, you will need to figure out your own way to wire up TXD and RXD.

Hardware flow control or "hand-shaking" signals are also available at pads labeled HSO (hand-shake out) and HSI (hand-shake in). HSO passes through a transmit channel of the converter and appears at pin 8 of the DB9, which is CTS (Clear To Send) of the connected COM port device. RTS (Request To Send) of the connected device arrives at pin 7 of the DB9, passes through a receive channel of the converter and appears at the HSI pad. If you want to implement hardware flow control, wire these to general purpose port bits of your choice and drive them appropriately in software. If you read RTS as inactive at HSI, the connected device does not want to receive data from you. Similarly, if your software is not ready to receive data from the device, make CTS inactive at HSO. Note that you cannot expect flow control to take effect immediately, so be ready to receive a few more data bytes after you drop HSO. For this to work, you will need to enable hardware flow control at the connected device. If the device implements hardware flow control but you do not wire up HSO and HSI as above, you can link them together and data will flow (but without control). If the device does not implement hardware flow control or you disable it, then it is safe to leave HSO and HSI unconnected. If all this is confusing to you, just link HSO and HSI together but if you're using a high data rate be on the lookout for missed data that is sent by one side when the other side is not ready to

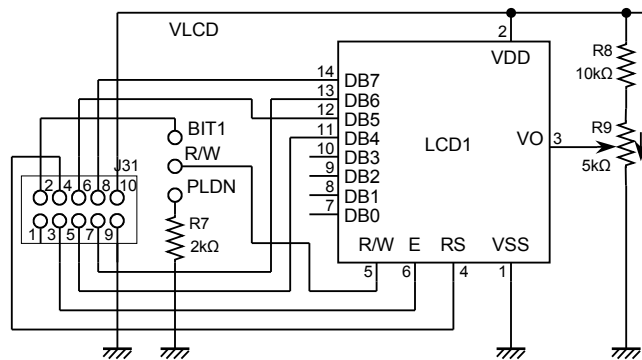
receive it. In your microcontroller, you can watch out for this using the Data OverRun flag of the UART.

Pins 4 (DTR) and 6 (DSR) of the DB9 are wired together on the Dragon Rider PCB. So, if the PC or terminal is looking at Data Set Ready, it will go active as soon as the PC raises its own Data Terminal Ready line.

2-Line by 16-Character LCD Add-on Circuit

The Dragon Rider PCB includes an area for a Lumex LCM-S01602DTR/M 2-line by 16-character LCD display module and support circuitry. A 10-pin header allows the LCD to be linked to a target microcontroller port with a 10-way ribbon cable of the type supplied with the Atmel STK500. The module uses the very common HD44780-type controller, wired in 4-bit mode to conserve port bits. Firm support is provided by standoffs and the display contrast can be adjusted with a trimmer potentiometer.

The figure at right is the circuit diagram of the LCD. The LCD circuit is independent of all other circuitry on the Dragon Rider, having only the ground connection in common. Power is taken from pin 10 of header J31. Therefore, the LCD can be connected to and used with other systems, for example the Atmel STK500. This is possible whether or not the Dragon Rider is powered or in use with the Dragon.



The upper four bits of the LCD module's HD44780-type controller are connected to pins 5 to 8 of J31, so when connected to a port header (such as J4 for Port D) they appear at the corresponding bits, i.e. 4 to 7. The register select (RS) line is connected to pin 4 and so appears at bit 3. The enable (E) line is connected to pin 3 and so appears at bit 2. The read/write (R/W) line is brought to a pad and can be linked with a wire jumper to pin 2, so appearing at bit 1 of the port, or can be linked to a pull-down resistor, R7, so that the controller is operated in write-only mode. Not being able to read from the controller means implementing timed, worst-case delays in software (rather than polling a status bit), but conserves a port bit.

The trimmer potentiometer, R9, allows the "contrast" of the LCD to be adjusted for best appearance. The specified LCD module needs a nominal 5 volt supply, so this circuit should not be used when the target voltage is below 4.5 volts. A direct connection to power supplied by the AVR Dragon is fine.

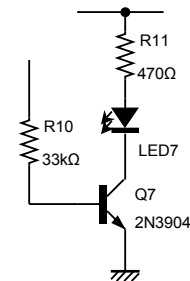
LED and Switch Add-on Circuit

The Dragon Rider PCB includes an area for six LEDs and five push-button switches with associated interface circuitry. The LED driver inputs and the switch outputs can each be independently connected to any of the available 8-bit input/output ports using 10-way ribbon cables. Additionally, with a suitable choice of component values, it is possible to connect both the LED drivers and the switches to the same port. Using a standard ribbon cable, that connects like-numbered pins at each end, the six LEDs and five switches are mapped to the most significant bits of the 8-bit port. Power is also obtained from the port header through the ribbon cable. Except for ground, both the

LED and switch circuits are completely isolated from the rest of the Dragon Rider and from each other. Power is taken from pin 10 of header J29 for the LEDs and pin 10 of header J30 for the switches. So, if desired, they can be connected to other systems using the Atmel 10-pin header standard, such as the STK500 or AVR Butterfly. This is possible whether or not the Dragon Rider is powered or in use with the Dragon.

LEDs and Transistor Drivers

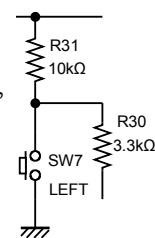
The figure at right is the circuit diagram of an LED channel (channel 7 is shown). There are six channels, numbered 2 to 7 from right to left on the PCB silk-screen and connected to pins 3 to 8, respectively, of J29. As a result, if J29 is connected, pin-for-pin, to one of the port headers of the Dragon Rider (for example, J4 for Port D), then LED 2 will be controlled by bit 2, LED 3 by bit 3, etc. All LEDs will pick up power from pin 10 of the port header. Pin 9 is connected to ground. Bits 0 and 1 of the port will not be connected and can be used for something else.



LED current is set to about 6 mA (for yellow or green LEDs) by the 470 Ω resistor (R11), assuming a 5 volt supply. The 33 k Ω base resistor (R10) provides a bias of 120 μ A to the transistor (again, assuming a 5 volt supply). This is enough to turn it fully on if the current gain is 50 or more. The specified 2N3904 transistor has sufficient gain. If you substitute a lower gain transistor, or use smaller collector resistors to get more current in the LEDs, you should re-calculate the necessary base resistors. The fairly high values in the parts kit were chosen so that the LED circuit can be connected to a port being used for something else without drawing much current and disturbing the logic levels.

Switches, Pull-Ups and Current Limit Resistors

The figure at right is the circuit diagram of a switch channel (LEFT channel shown). There are five channels, marked LEFT, RIGHT, UP, DOWN and ENTER on the PCB silk-screen and connected to pins 8, 5, 7, 6 and 4, respectively, of J30. As a result, if J30 is connected, pin-for-pin, to one of the port headers of the Dragon Rider (for example, J4 for Port D), then pressing the LEFT, RIGHT, UP, DOWN and ENTER switches will change the input level at bits 7, 4, 6, 5 and 3, respectively. Switch numbers are also shown on the silk-screen (for example, LEFT is SW7), and these numbers correspond directly with the port bits (LEFT; SW7; bit 7). All switches will pick up power from pin 10 of the port header. Pin 9 is connected to ground. Bits 0, 1 and 2 of the port will not be connected and can be used for something else.



Switch current (when closed) is set to about 0.5 mA by the 10 k Ω pull-up resistor (R31), assuming a 5 volt supply. The 3.3 k Ω resistor (R30) provides current limiting to about 1.5 mA in the event that the connected port bit is driving high and the switch is closed. Thus, it is not only safe to allow this, the port will also be able to drive a valid logic level. If you use different resistor values here, take into account whether you want to allow the port bits used to read the switches to be also used as outputs to drive something else.

Connecting LEDs and Switches to the Same Port

With the resistor values shown and supplied in the parts kit, it is perfectly OK to

connect the LEDs and switches (J29 and J30) to the same port, for example Port D at J4. Your software should normally configure the port bits in use (bits 2 to 7 if you're using all six LEDs) as outputs and set their states to high for the LED to be ON or low for the LED to be OFF. Periodically, perhaps every 5 to 20 ms, your software should change the configurations of the port bits used with switches (bits 3 to 7 for them all) to make them inputs, read the input values and then restore the configuration to outputs. The blink in the LEDs will not be noticeable, but now you have a snapshot of the states of the switches. This should be de-bounced in the normal way before recognizing a change of state.

If you like this idea a lot, you might want to make a permanent connection between J29 and J30 so that you only need to run a ribbon cable to one of them (J30 is near the front edge of the board). One way to do this would be to solder wires underneath the board to the ends of the header pins that poke out from the bottom. Otherwise, you will probably need a ribbon cable with several sockets to "daisy-chain" the headers together and some careful folding will be needed to keep it out of the way.

Connecting the LCD to the Same Port As Well

In the same way that your software can take a break from holding the LEDs ON or OFF to sample the switches, it can also take time out to write something to the LCD without any noticeable flicker. So, the LCD (J31) can also be connected to the same microcontroller port. To write to the LCD, your software needs to put a pulse on the enable (E) line, which is connected to pin 3 (bit 2). So, you will have to give up the independent use of LED2 to make this work. Note that there is no switch on bit 2 of the port, so you won't be accidentally writing things to the LCD by pressing a switch.