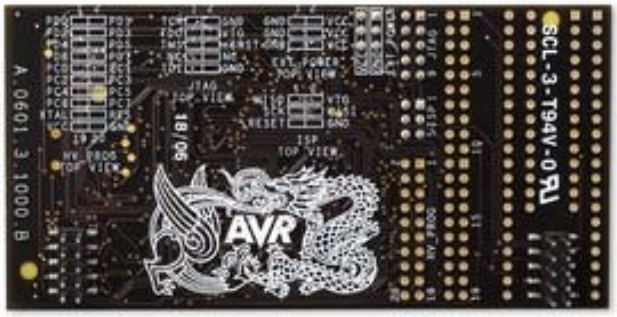
Introducing AVR Dragon

'



Front Side Back Side

With the **AVR Dragon**, Atmel has set a new standard for low cost development tools. AVR Dragon supports all programming modes for the Atmel AVR device family. It also include full emulation support for devices with 32kB or less Flash memory.
At a fraction of the price traditionally associated with this kind of featured tool, the AVR Dragon will fulfill all your programming and emulation needs. The flexible and secure firmware upgrade feature allows AVR Studio to easily upgrade the AVR Dragon to support new devices.

To see which devices are currently supported please read the Device Support page.
(New devices will be added through AVR Studio updates or Service Packs on a regular basis)

Supported Protocols

Currently the following protocols are supported:

Programming Interfaces:
- In System Programming (ISP)
- High Voltage Serial Programming (HVSP)
- Parallel Programming (PP)
- JTAG Programming (JTAG Prog)

Emulation Interfaces: (Only available for devices with 32kB Flash or less)
- JTAG (JTAG)
- debugWIRE (dW)

AVR Dragon can be used with an external target board. However, the onboard prototype area, allow simple programming and debugging without any additional hardware besides strapping cables. Please see the Using the AVR Prototype Area section for a description on how to use this.

AVR Dragon is powered by the USB cable, and can also source an external target with up to 300mA (from the VCC connector, 5V) when programming or debugging. For more information on technical details, please read the AVR Dragon Requirements section. If the target is already powered by an external power source, the AVR Dragon will adapt and level convert all signals between the target and the AVR Dragon.

Note: If the target board is powered by external power source, no connection should be made betwwen the VCC connector and the external board.

AVR Dragon if fully supported by AVR Studio. This allows the AVR Dragon firmware to be easily updated to support new devices and protocols. When connecting the AVR Dragon, AVR Studio will automatically check the firmware and prompt the user if an updated firmware is available.

Device Support

The following devices are currently supported by AVR Dragon.

	Programming				Debugging			
Device	**ISP**	**HVSP***	**PP***	**JTAG**	**JTAG**	**dW**	**Connection sheet**	**Remarks**
Tiny parts								
ATtiny11		x					SCKT3400D1	
ATtiny12	x	x					SCKT3400D1	
ATtiny13	x	x				x	SCKT3400D1	
ATtiny15	x	x					SCKT3600A1	
ATtiny2313	x	x				x	SCKT3300D3	
ATtiny24 ATtiny44 ATtiny84	x	x				x	SCKT244484	ATtiny84 HVSP programming issue
ATtiny25 ATtiny45 ATtiny85	x	x				x	SCKT3400D1	
ATtiny26	x		x				SCKT3700A1	ATtiny26 PP programming issue
ATtiny261 ATtiny461 ATtiny861	x		x			x	SCKT3700A1	
ATtiny28			x				SCKT3500D-	
ATtiny43U	x	x				x	Off Board Target	
ATtiny48 ATtiny88	x		x			x	SCKT3200A2	
ATtiny167	x		x			x	Off Board Target	
Mega parts								
ATmega8515	x		x				SCKT3000D3	
ATmega8535	x		x				SCKT3100A3	
ATmega48(P) ATmega88(P) ATmega168(P) ATmega328P	x		x			x	SCKT3200A2	
ATmega8	x		x				SCKT3200A2	
ATmega16	x		x	x	x		SCKT3100A3	
ATmega164P ATmega324P	x		x	x	x		SCKT3100A3	
ATmega644(P) ATmega1284P	x		x	x			SCKT3100A3	
ATmega162	x		x	x	x		SCKT3000D3	
ATmega32	x		x	x	x		SCKT3100A3	
ATmega32C1	x		x	x		x	Off Board Target	
ATmega32M1	x		x	x		x	Off Board Target	
ATmega32U4	x		x	x	x		Off Board Target	
ATmega64	x		x	x			Off Board Target	
ATmega128	x		x	x			Off Board Target	
ATmega640	x		x	x			Off Board Target	
ATmega1280	x		x	x			Off Board Target	
ATmega1281	x		x	x			Off Board Target	
ATmega2560	x		x	x			Off Board Target	
ATmega2561	x		x	x			Off Board Target	
ATmega165(P) ATmega169(P)	x		x	x	x		Off Board Target	
ATmega325(P) ATmega329(P) ATmega3250(P) ATmega3290(P)	x		x	x	x		Off Board Target	

ATmega645 ATmega649 ATmega6450 ATmega6490	x		x	x			Off Board Target	
ATmega406				x	x		Off Board Target	Note: Supports only debugging up to 32kB
Other							Off Board Target	
AT90CAN32	x		x	x	x		Off Board Target	
AT90CAN64 AT90CAN128	x		x	x			Off Board Target	
AT90PWM2 AT90PWM2B AT90PWM3 AT90PWM3B	x		x			x	Off Board Target	
AT90PWM216 AT90PWM316	x		x			x	Off Board Target	
AT90USB646 AT90USB647 AT90USB1286 AT90USB1287	x		x	x			Off Board Target	
AT90USB82 AT90USB162	x		x			x	Off Board Target	
ATmega16HVA	x	x				x		
ATmega32HVB	x		x			x	Off Board Target	

Note that the PP/HVSP interfaces (Parallel and High Voltage Serial Programming) are not level converted, but runs at 5V. I.e. extreme care should be taken if using these interfaces off board.

Abbreviations:
ISP: In System Programming
HVSP: High Voltage Serial Programming
PP: Parallel Programming (high voltage)
dW: debugWire

AVR Dragon does not support debugging on devices above 32kB.

New devices will be supported through updates of AVR Studio. Please visit www.atmel.com to download the latest version.

What's New

February 20th, 2008 - Sw: 0x0101 0x0200 (released with AVR Studio 4.14)

- Added full part support for ATtiny88, ATtiny167, ATmega32C1, ATmega32M1 and ATmega32U4

November 21st, 2007 - Sw: 0x0101 0x0200 (released with AVR Studio 4.13 SP2)

- Added programming support for support for ATmega1284P
- Added full part support for ATtiny43U, ATtiny48 and ATmega32HVB

July 5th, 2007 - Sw: 0x0101 0x010A (released with AVR Studio 4.13 SP1)

- Added part support for AT90USB82, AT90USB162,AT90PWM216, AT90PWM316 and ATmega16HVA
- Fixed in FW troubles with enabling debugWIRE mode in debug-projects

February 26th, 2007 - Sw: 0x0101 0x0109 (released with AVR Studio 4.13)

- Added part support for ATmega88P and ATmega168P

January 4th, 2007 - Sw: 0x0101 0x0109

- Added part support for all AVR devices. (no debugging on parts with 64kB or more Flash)
- Fixed timing problems on High Voltage Interface
- Made more robust ISP programming to handle skew on ISP lines

September 18th, 2006 - Sw: 0x0101 0x0103

- Fixed problem with programming more than 10 bytes of data to EEPROM in debugWIRE mode
- Fixed problems with reading and storing ISP programming frequence
- Fixed that PP/HVSP is automatically selected as programming interface if it was used last time.
- Fixed USB reenumeration issue, caused when disconnect/connecting the AVR Dragon
- When target voltage is below 1.8V, the AVR Dragon now reports the actual voltage, not only "Could not find target voltage"

August 3rd, 2006 - Sw: 0x0100 0x0102

- Full Support for: ATmega16, ATmega169, ATmega325P, ATmega3250P, ATmega329P and ATmega3290P

June 29, 2006 - AVR Studio 4.12 SP3: Sw: 0x0100 0x0102

- Fixed bug causing slow ISP programming

June 12, 2006 - AVR Studio 4.12 SP3: Sw: 0x0100 0x0101

- Full Support for: ATmega48/88/168, ATmega8, ATmega32, ATmega128, ATtiny13, ATtiny25/45/85 and ATtiny2313

Getting Started

Please read this section before connecting the AVR Dragon to the computer or target.

Important !	Please install AVR Studio and the USB driver **before** connecting AVR Dragon to your PC

Follow these simple steps to get started using the AVR Dragon:

1. Download AVR Studio 4.12 SP3 or later from http://www.atmel.com/avrdragon
2. Install AVR Studio and the USB driver
3. Connect AVR Dragon to the computer, and auto-install new hardware (AVR Dragon) on the computer
4. Start AVR Studio and the AVR Dragon Programming Dialog
5. Connect AVR Dragon to the target

USB Setup

In order to use the AVR Dragon it is required to install the AVR Studio and USB driver first. Please do not connect the AVR Dragon to the computer before running the USB Setup in order to follow this procedure described in Software and USB Setup.

Unpacking the AVR Dragon.

The box contains:

- AVR Dragon tool
- Internet link to Software (http://www.atmel.com/avrdragon)

There is no CD shipped with the AVR Dragon. The only way of getting the software is by downloading it directly from the Internet.

You will also need: (not included)

- PC with free USB connector or a USB HUB capable of delivering 500mA
- USB Cable
- AVR Studio 4.12 with Service Pack 3 or later (Link: http://www.atmel.com/avrdragon)
- 6/10 pin Header Connector (or similar cables to connect the AVR Dragon to the target board)

System Requirements

The minimum hardware and software requirements are:

1. Pentium (Pentium II and above is recommended)
2. Windows® 98, Windows ME, Windows® 2000 or Windows® XP
3. 128 MB RAM
4. AVR Studio 4.12 with Service Pack 3
5. USB port, self-powered (500mA required)
6. Internet Connection for Software download

Note: Windows 95 and Windows NT does not support USB, hence cannot be used with AVR Dragon

Hints

· Always power up the AVR Dragon first before connecting to or powering up the target

· AVR Dragon needs to sense the target voltage at pin 2 on the ISP header or pin 4 on the JTAG header.
-This also applies when using High Voltage interface.

· The High Voltage interface are set to 5V. Make sure the target board are running at 5V before using this interface off board.

· VCC header are set to 5V, and can source max 300mA.

· If AVR Dragon is used for programming/debugging targets on the STK500, the RESET jumper on the STK500 must be removed.

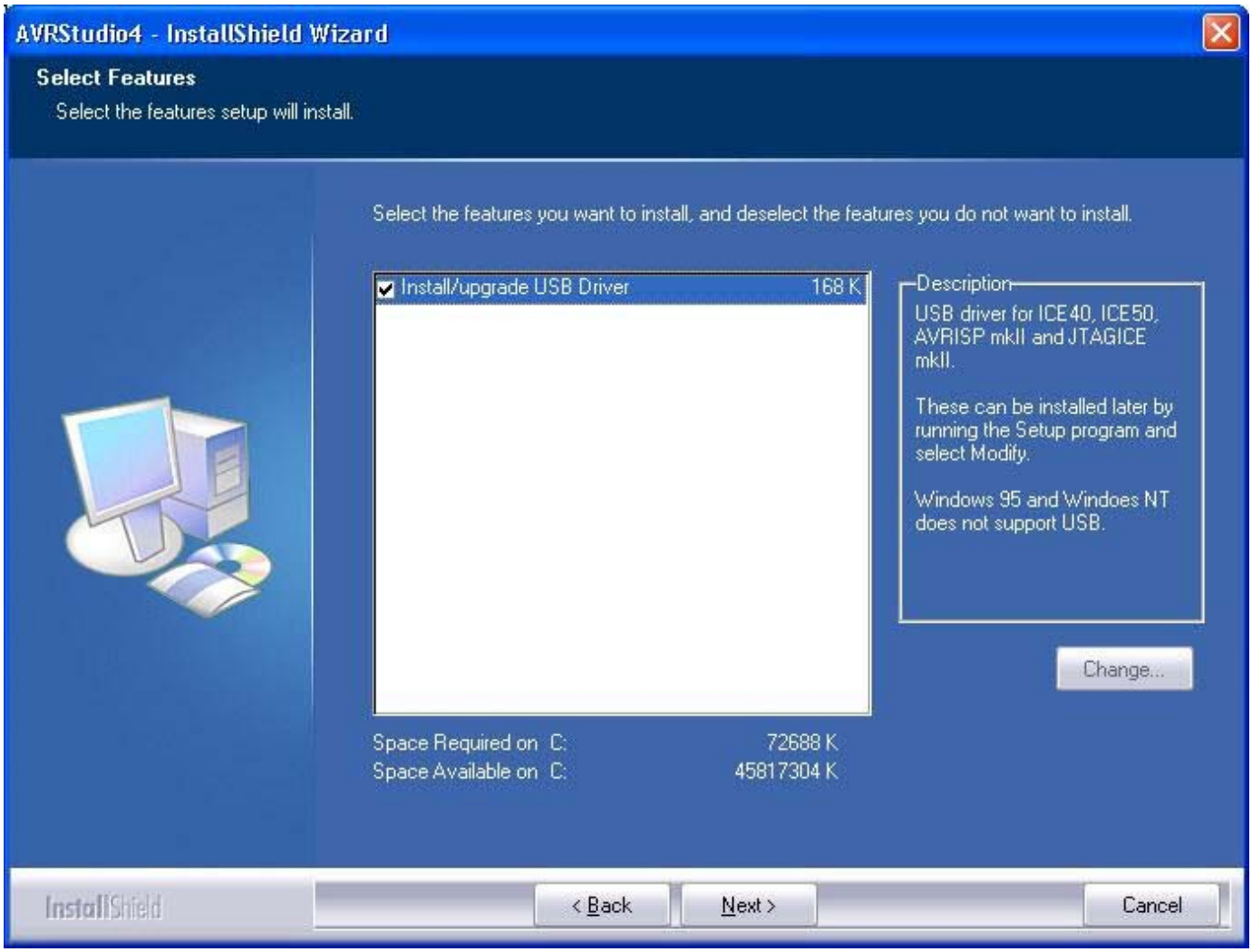
Software and USB Setup

Software and USB Setup

In order to use the AVR Dragon it is required to install the USB driver. Please do not connect the AVR Dragon to the computer before running the USB Setup. USB driver installation is done during the AVR Studio installation.

Note: AVR Dragon requires AVR Studio 4.12 with Service Pack 3 or later. Latest version of the AVR Studio can be found at: www.atmel.com/products/AVR/

Start the AVR Studio installation. During this installation the dialog box in the figure below will be presented to the user.



To install the USB driver, check the Install/Upgrade USB Driver checkbox, and the USB Driver will automatically be installed.

Install new hardware on the computer

When AVR Studio and USB driver installation is finished, please attach the USB cable to both PC and AVR Dragon. (The AVR Dragon is powered from the USB). If it is the first time the AVR Dragon is connected to the computer, the box below will appear:

If running Windows XP you need to click "Next" a couple of times. Please wait until the installation process completes by itself. It may take from a few seconds up to a few minutes depending on the computer and operating system.

If the USB driver is correctly installed and AVR Dragon is connect to the PC, the green LED inside the encasing next to the USB connector will be lit.

If the AVR Studio for some reason can't detect the AVR Dragon after the USB setup, try to restart the computer in order to get the driver properly loaded.

Install USB driver after AVR Studio is installed

The USB driver can be installed even after AVR Studio have been installed by following these steps:

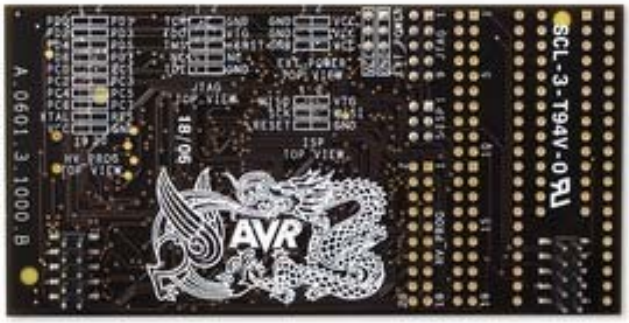
1. Open "Control Panel" on the PC (Windows 95 and Windows NT does not support USB)
2. Select "Add or Remove Programs"
3. Select "AVRStudio4" in the list of programs
4. Click on the "Change" button
5. Select "Modify"
6. Select "Install/upgrade USB Driver"

The USB driver is now properly installed on the PC

Note: The AVR Dragon requires a USB port that can deliver 500mA (self-powered USB hub).

Board Description / Headers

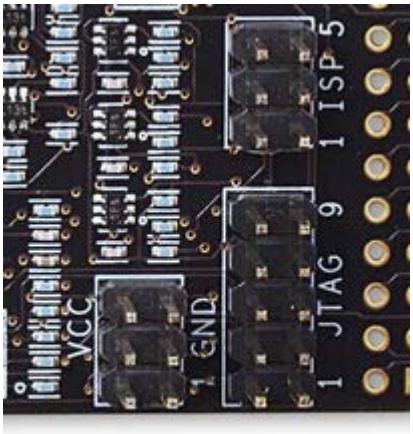
AVR Dragon Board:



Headers:

Out of the box, the AVR Dragon has the following 3 header connectors mounted:

- ISP Header - Used for ISP programming and debugWIRE OCD
- JTAG Header - Used for JTAG programming and JTAG OCD.
- VCC Header - Used for powering Devices placed in the prototype area, or to power external target boards (max 300mA). Set to fixed 5V



The following header are not mounted

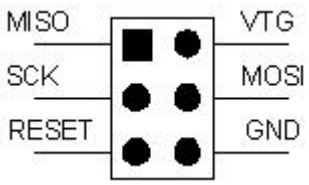
- HV_PROG Header
- EXPAND Header
- 40-pin DIP socket
- 28-pin DIP socket

ISP Header (mounted):

This 6-pin header uses the standard AVR ISP pinout for easy connection to external targets. The signals are level-converted to allow

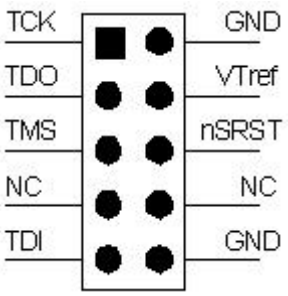
communication with targets running at any voltage between 1.8 and 5.5V

Note that the target voltage must be applied to pin 2 on the ISP header for the AVR Dragons level converters.



JTAG Header(mounted):

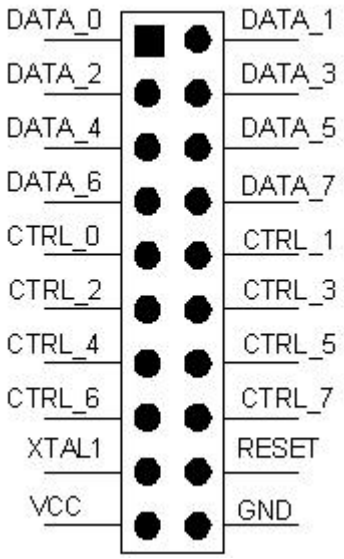
The 10-pin JTAG header is a standard pinout JTAG connector. When connecting the AVR Dragon JTAG header to an external target, the signals are level converted to match the target board voltage. This is done automatically. Please note that the AVR Dragon will not power the target through the JTAG interface. The target needs to be powered through a dedicated powersupply. (or by powering it using the VCC connector (5.0V max 300mA). AVR Dragon needs to sense the target voltage on pin 4 on the JTAG connector.



Pin	Signal	I/O	Description
1	TCK	Output	Test Clock, clock signal from AVR Dragon to target JTAG port
2	GND	-	Ground
3	TDO	Input	Test Data Output, data signal from target JTAG port to AVR Dragon
4	VTref	Input	Target reference voltage. VDD from target used to control level-converter
5	TMS	Output	Test Mode Select, mode select signal from AVR Dragon to target JTAG port
6	nSRST	In/Out-put	Open collector output from adapter to the target system reset. This pin is also an input to the adapter so that the reset initiated on the target may be reported to the AVR Dragon
7	-	-	Not connected
8	-	-	Not Connected
9	TDI	Output	Test Data Input, data signal from AVR Dragol to target JTAG port
10	GND	-	Ground

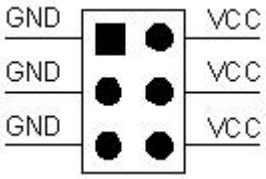
HV_PROG Header (not mounted):

The HV_PROG connector contains all signal required to do HVSP or PP programming. The signals on this connector is not level-converted, and should only be connected to the EXPAND connector on the AVR Dragon. You could damage both your target and the AVR Dragon if you try to do HVSP or PP on an external target board. The signal levels on the HV_PROG header are 5V.



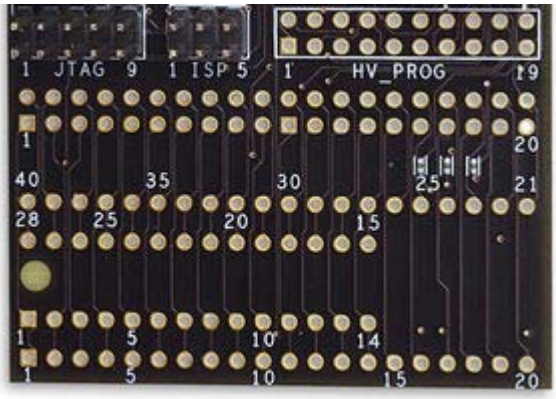
The figure above shows the general pinout of the HV_PROG header. The High Voltage programming pin mapping are not the same for all parts. See the HVSP Description or PP Description chapters for more information on the pinout of this header.

VCC Header (mounted):



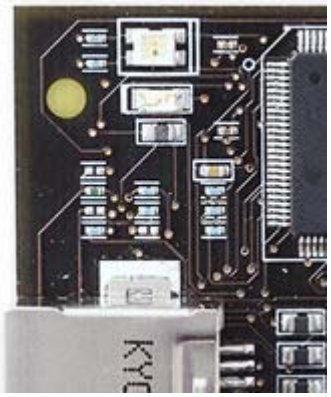
The VCC Header contains 5.0 Volt VCC and GND that must be used to power the target device placed in the prototype area of the AVR Dragon board. The voltage can also be used to power an external target board, but it is important that the current consumption is less than 300mA. Please note that the AVR Dragon current sourcing capabilities are also limited by the amount of current the Host USB controller can deliver.

EXPAND Header (not mounted):



The expand connector is directly mapped to the 28 and 40-pin DIP sockets. Pin 1 on the connector - is pin one on both the 28 and the 40pin DIP socket. When doing either programming or emulation on-board, the appropriate signals should be routed from the ISP, JTAG, VCC and HV_PROG headers to the correct pins on the EXPAND connector. Please read the "Using the AVR Dragon Prototype Area" section for more information on how to use this function.

Status LEDs



Two LEDs show the status of the AVR Dragon. Check the Troubleshooting Guide to check for solutions if there are any errors.

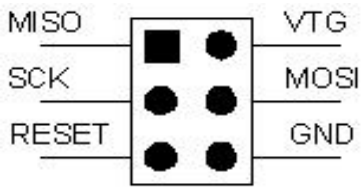
LED #	Color	Description
2	Green	Indicates USB traffic
1	Red	Idle, not connected to AVR Studio
	Dark	Idle, connected to AVR Studio
	Green	Data Transfere
	Yellow	Firmware Upgrade or Initialization

In System Programming

In System Programming is well suited for programming devices soldered onto external target boards. This section explains how to connect the AVR Dragon to ISP program an external target. The ISP lines are equipped with level converters that automatically will level shift the AVR Dragon to the target board voltage.

It is recommended that a 6-pin header connector with 2.54mm (100 MIL) spacing is placed on the target board to allow easy access to the ISP programming interface. The following pinout should be used.

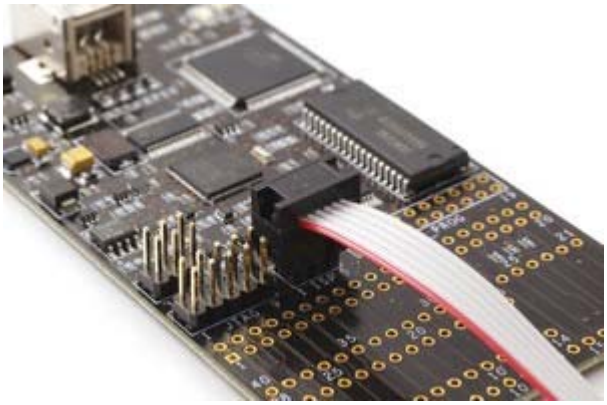
Figure: 6pin Header Connector with 2.54mm (100 MIL) spacing

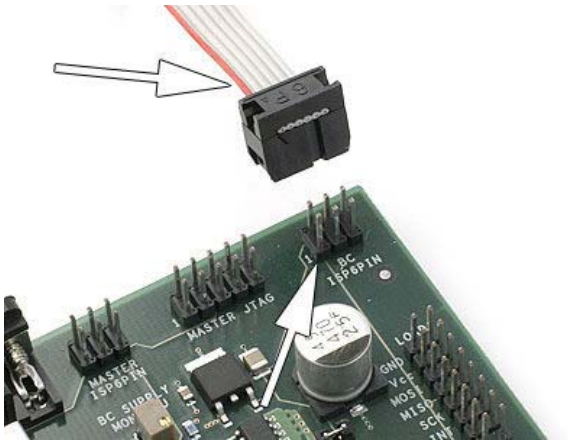


Note: When connecting the AVR Dragon to the target, connect MISO to MISO pin on the target device, MOSI to MOSI and so on.

Note: AVR Dragon must sense the target voltage at pin 2 on the ISP header in order to set up the level-converter. For on-board targets, the voltage must be supplied from pin 2,4,6 on the VCC header (5V) into pin 2 (VTG) on the ISP header. When using off-board targets there should be no connection between VCC header and pin 2 of the ISP header.

Connect the 6pin cable from the AVR Dragon to the external target as shown in these pictures:





debugWIRE OCD interface is also accessed through this ISP header.

See the AVR Studio user guide for information about using the Programming Dialog to program the target device.

High Voltage Serial Programming Description

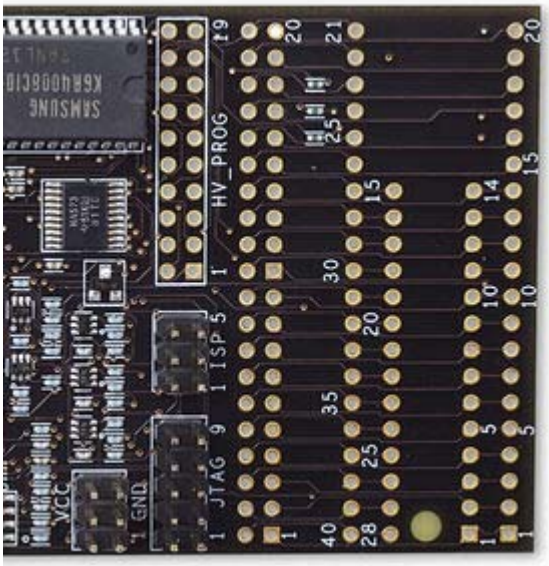
Low pin count AVR devices do not have enough IO pins to support the full Parallel Programming interface. These devices instead use HVSP programming, which is a serial version of the Parallel Programming interface.

Important!
Extreme care should be taken if using HVSP mode to program a AVR device on an external target. The HVSP lines do not have level converters, so it is important that the target board is powered by the AVR Dragon VCC header, and **not** using another power supply. In addition the AVR Dragon will apply 12V to the reset pin, so it is important that the target board is designed to handle 12V on this line.

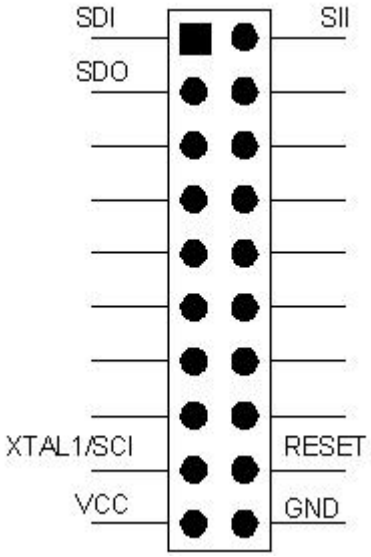
Note that the target voltage, i.e. the 5V from the VCC header must be applied to either pin 2 on the ISP header or pin 4 on the JTAG header. This because the AVR Dragon must read the target voltage.

To avoid damaging the Target Board, the AVR Dragon or both, it is recommended to **only** use HVSP mode on devices placed in the 28/40 pin DIP socket on the AVR Prototyp area on the AVR Dragon.

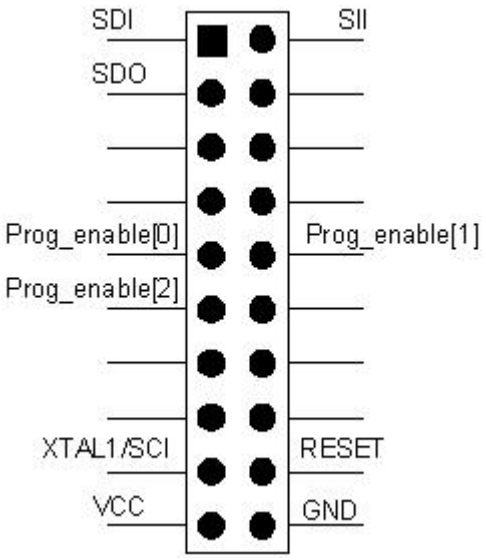
Figure: Prototype Area



The High Voltage Serial Programming (HVSP) interface is found on the HV_PROG header. Figure below shows the pinout of the HV_PROG header when used for HVSP for all parts except ATtiny24/44/84 :



The ATtiny24/44/84 uses separate pins for entering programming mode. See the pinout of the HV_PROG header when used for these parts below:



Please see the "Device Connection Sheet" section for information on how to connect AVR Dragon for HVSP programming for the different supported devices.

See the AVR Studio user guide for information about using the Programming Dialog to program the target device.

Parallel Programming Description

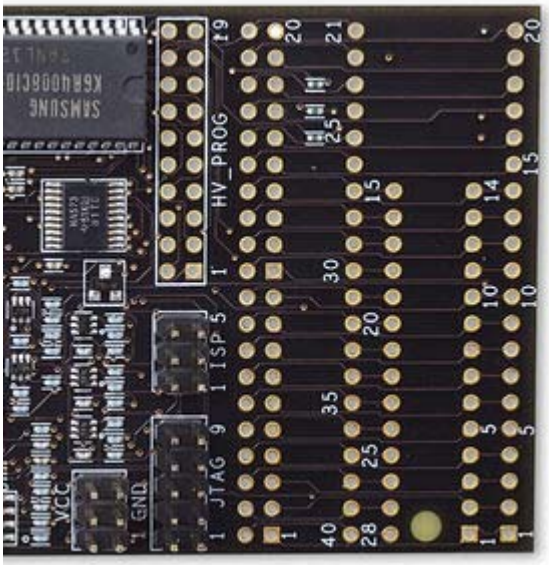
High pin count AVR devices support the full Parallel Programming (PP) interface. This interface offer high speed programming, and also support programming all fuse and lock bits in the AVR Device.

Important!
Extreme care should be taken if using PP mode to program a AVR device on an external target. The PP lines do not have level converters, so it is important that the target board then is powered by the AVR Dragon VCC header, and not using its own power supply. In addition the AVR Dragon will apply 12V to the reset pin, so it is important that the target board is designed to handle 12V on this line.

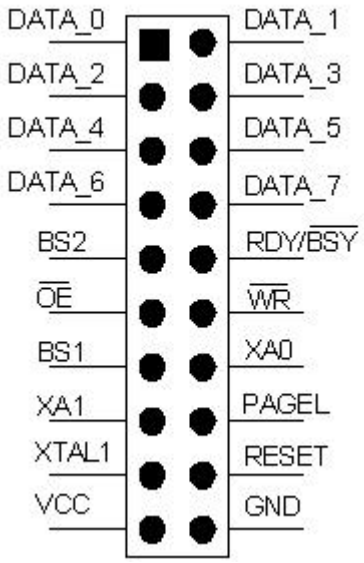
Note that the target voltage, i.e. the 5V from the VCC header must be applied to either pin 2 on the ISP header or pin 4 on the JTAG header. This because the AVR Dragon must read the target voltage.

To avoid damaging the Target Board, the AVR Dragon or both, it is recommended to **only** use PP mode on devices placed in the 28/40 pin DIP socket on the AVR Prototyp area on the AVR Dragon.

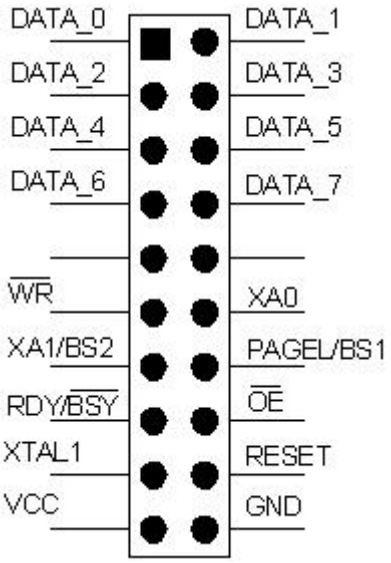
Figure: Prototype Area



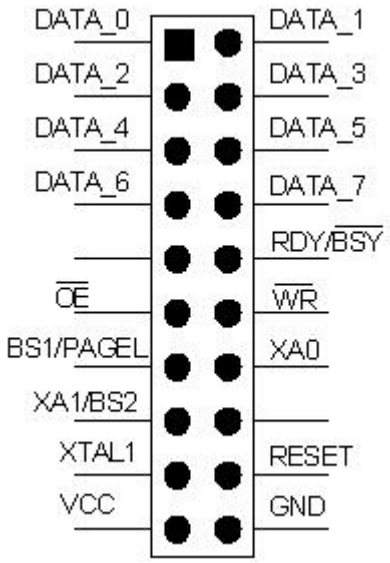
The HV_PROG header pinout is listed below. This is the standard pinout for about all AVR parts. However the pinout on the HV_PROG header are slightly different for some parts, these exceptions are listed further down in this page.



For ATtiny26/261/461/861 the HV_PROG header will have this pinout:



For ATtiny2313 the HV_PROG header will have this pinout:



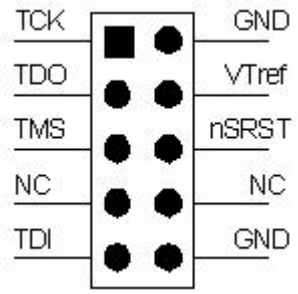
Please see the "Device Connection Sheet" section for information on how to connect AVR Dragon for PP programming.

See the AVR Studio user guide for information about using the Programming Dialog to program the target device.

JTAG Programming Description

AVR devices with JTAG interface also support programming through this interface. The connection for JTAG programming is the same same as the JTAG debug interface. Please see section "Connecting to target through the JTAG Interface " for information how to connect the AVR Dragon to your external target board.

It is also possible to do JTAG Programming on a device placed on the Prototype Area of the AVR Dragon. Please see the "Device Connection Sheets" for information on how to connect the different AVR devices.



See the AVR Studio user guide for information about using the Programming Dialog to program the target device.

Command Line Software

The DOS command line of the programming interface is the same as for the STK500 starter kit. It is named "AVRDragon.exe" and allows simple batch files for automatic programming. The program is located on the disk at: ..\Program Files\Atmel\AVR Tools\AVRDragon\

Typing "avrdragon -h" will display all the commands, options and the parts currently supported by the AVR Dragon.

```
Commands:
 -h, -?          Print this help.
 -he             Print exit codes.
 -e              Erase device.
 -f <fusebytes> Write high/low fuse bytes. The LSB of fusebytes will be
                 written to the low fuse byte, and the MSB will be written
                 to the high fuse byte.
 -j <ID>         Read JTAG ID string. If an optional ID is specified, the
                 program will terminate if the read ID differs.
 -l <lockbyte>   Set the lock byte to the specified value.
 -p[f|e|b]       Program FLASH (-pf), EEPROM (-pe) or both (-pb) memories.
                 Requires input files specified with the -if (FLASH)
                 and/or -ie (EEPROM) options.
 -q              Read fuse bytes. If the selected device has extended fuse,
                 it will be also be read.
 -r[f|e|b]       Read FLASH (-rf), EEPROM (-re) or both (-rb) memories.
                 Requires output files specified with the -of (FLASH) and/or
                 -oe (EEPROM) options.
 -s              Read signature bytes.
 -v[f|e|b]       Verify FLASH (-vf), EEPROM (-ve) or both (vb) memories.
                 Requires input files specified with the -if (FLASH) and/or
                 -ie (EEPROM) options.
 -y              Read lock byte.
 -E <ext.fuse>   Write extended fuse byte.
 -F <fusebytes> Verify high/low fuse bytes. The LSB of fusebytes will be
                 compared to the low fuse byte, and the MSB will be compared
                 to the high fuse byte.
 -L <lockbyte>   Verify the lock byte.
 -G <ext.fuse>   Verify the extended fuse byte against the given value.
 -O <address>    Read oscillator calibration byte. Address is location of byte
                 as specified in the device's data sheet.
 -S[f|e] <address>
                 Write oscillator calibration byte read with -O or -Z to given
                 location in FLASH (-Sf) or EEPROM (-Se).
 -W              Disable debugWIRE interface, enabling ISP mode. Only available
                 for debugWIRE devices.
 -Y              Perform the oscillator calibration sequence. See appnote
                 AVR053 for more information.
 -Z <address>    Load value from address in EEPROM prior to erasing the device.
                 This value can be written to FLASH or EEPROM again with
                 the -S[f|e] command.
Options:
 -ae <start>,<end>
                 Specify address range for reading EEPROM memory. Default is
                 entire memory.
 -af <start>,<end>
                 Specify address range for reading FLASH memory. Default is
                 entire memory.
 -d <device>     [required] Specify the type of device.
 -ie <file>      Specify input file for programming EEPROM memory.
 -if <file>      Specify input file for programming FLASH memory.
 -m[i|j|p]       Choose programming mode: ISP (-mi), parallel/high-voltage (-mp)
                 or JTAG (-mj) (default).
 -oe <file>      Specify output file for reading EEPROM memory.
 -of <file>      Specify output file for reading FLASH memory.
 -z              No progress indicator. Useful when redirecting output to file.
 -D <devsbefore,devsafter,bitsbefore,bitsafter>
                 Set daisy chain settings. Only available in JTAG mode.
 -I <speed>      Specify ISP frequency (Hz). Only available for ISP mode.
 -J <speed>      Specify device speed (Hz). Only available for JTAG mode.
 -R              Apply external reset while programming. Only available in
                 JTAG mode.

Example usage: avrdragon.exe -dATmega128 -e -pf -if -mj c:\tmp\test.hex
```

Connecting to target through the debugWIRE Interface

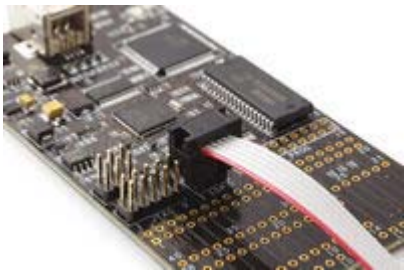
A minimum of 3 wires is required for communication between AVR Dragon and the target board with the debugWIRE interface. These Signals are RESET, VTref and GND.

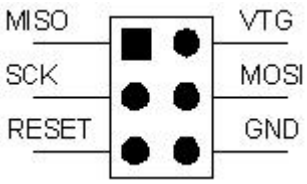
Important!
This interface uses only 1 pin, (RESET pin) for communication with the target. To enable the debugWIRE interface on an AVR Device, the debugWIRE Enable fuse (DWEN) must be programmed, (DWEN=0). AVR devices with debugWIRE interface are shipped with the DWEN fuse un-programmed from the factory. The debugWIRE interface itself cannot enable this fuse. The DWEN fuse can be programmed through ISP mode, which requires connection to a 6-pin header. For this reason it is recommended to place the full 6-pin ISP connector on your target board to simplify emulation and programming.

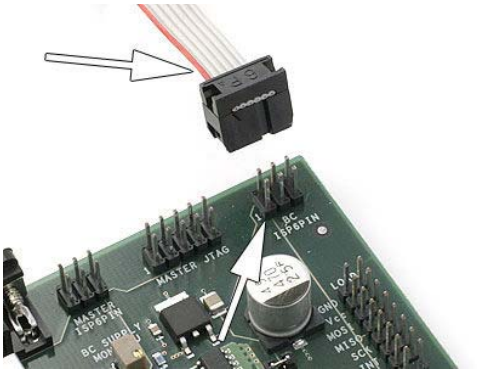
NOTE: When the DWEN fuse is set, the ISP Interface normal functionality is disabled. This because the debugWIRE must have control over the RESET pin. When DWEN is set it is no longer possible to use ISP. Use debugWIRE or High Voltage programming to disable the DWEN fuse.

Note
If using this connection from AVR Dragon on a STK500, be sure to de-attach the RESET jumper on the STK500. And connect to the correct ISP header for the actual AVR device, guided by the colour code in the STK500 silk-print.

AVR Dragon debugWIRE connector







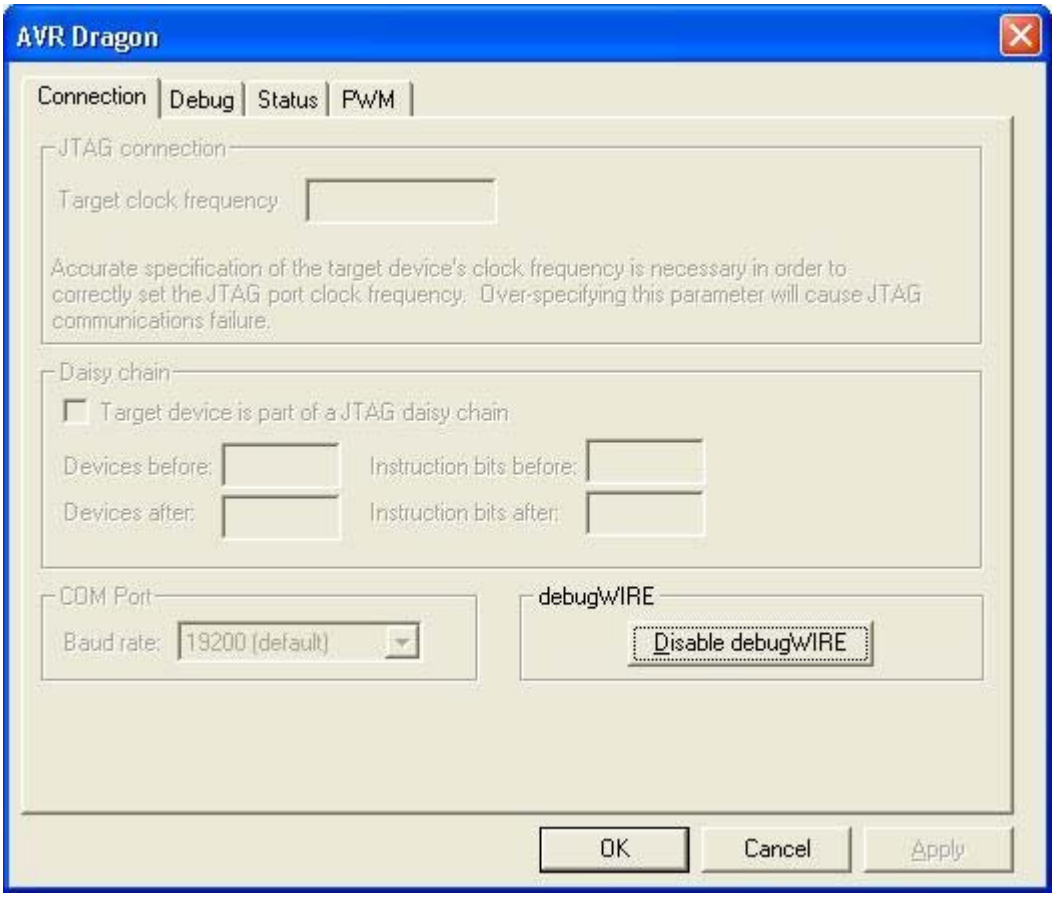
Connecting AVR Dragon probe to 6-pins ISP header using a 6-pin cable

When DWEN fuse is programmed, there is only need for GND, VTref and RESET line for using the debugWIRE interface. However to ease the task of changing between ISP mode and debugWIRE mode, it is recommended to do debugWIRE with all six lines connected. The AVR Dragon will automatically tristate all unused ISP pins when running debugWIRE.

Re-enabling the ISP interface

By following the description below, the ISP Interface is re-enabled.

1. Connect the AVR Dragon to the target with ISP (6-pin connection) as described above.
2. Load a project and Start a debug session using the "Start Debugging" command (found under the Debug pulldown-menu in AVR Studio)
3. Open "AVR Dragon Options" (found under the Debug pulldown-menu in AVR Studio) and select the "Connections" tab.
4. Click on the button "disable DebugWIRE" as shown in the figure below. And click "Yes" in the next message box.



Note: Some precautions regarding the RESET line must be taken to ensure proper communication over the debugWIRE interface. If there is a pull-up on the RESET line, this resistor must be larger than 10Kohm, and there should be no capacitive load. The pull-up resistor is not required for debugWIRE functionality. Other logic connected to the RESET line should be removed.

Note
It's not possible to use the debugWIRE Interface if the lockbits on the target AVR are programmed. Always be sure that the lockbits are cleared before programming the DWEN fuse and never set the lockbits while the DWEN fuse is programmed. If both the debugWIRE enable fuse (DWEN) and lockbits are set, one can use High Voltage Programming to do a chip erase, hence clear the lockbits. When the lockbits are cleared the debugWIRE Interface will be re-enabled.

Note
The ISP Interface is only capable of reading fuses, signature and do a chip erase when the DWEN fuse is unprogrammed.

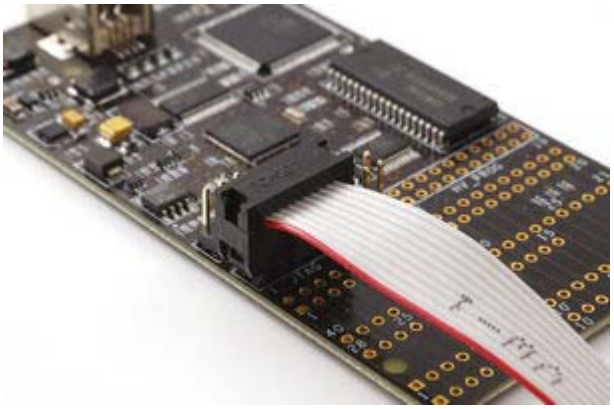
Connecting to the target through the JTAG Interface

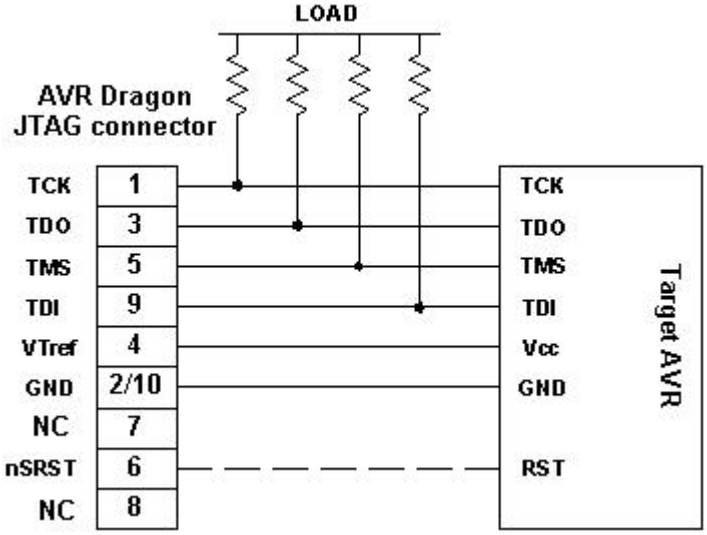
A minimum of 6 wires is required to connect AVR Dragon to the target board. These Signals are TCK, TDO, TDI, TMS, VTref and GND.

Optional line is the nSRST. The nTRST signal is not used, and is reserved for compatibility with other equipment.
nSRST is used to control and monitor the target reset line. This is however not necessary for correct emulation. But if the application code sets the JTD bit in the MCUCSR, the JTAG Interface will be disabled. For the AVR Dragon to reprogram the target AVR, it will need to have control of the Reset Pin.

The figure below shows which JTAG lines should be connected to the target AVR to ensure correct operation. To avoid drive contention on the lines it is recommended that series resistors are placed between the JTAG lines and external circuitry. The value of the resistor should be chosen so that the external circuitry and the AVR do not exceed their maximum ratings (i.e. sinks or sources to much current).

Connecting AVR Dragon to Target Board





Connecting AVR Dragon to several devices placed in a JTAG Chain

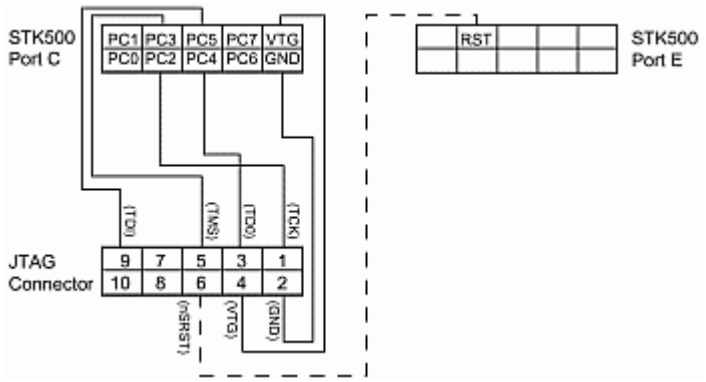
AVR Dragon support emulation of devices placed in a JTAG Chain. When connecting N devices in a JTAG scan chain all devices should connect to TMS and TCK in parallel. The first device should connects it's TDI to the emulator while the TDO should be wired to TDI of the next device up to device N. The last device should connects it's TDO to the emulator.

Connecting AVR Dragon to STK500

STK500 does not have a dedicated JTAG interface connector. To connect the AVR Dragon to the STK500 board, the JTAG Probe must be strapped to the appropriate JTAG Port Pins of the target device using a squid cable. Check the target device datasheet for the location of the JTAG pins on the appropriate device. Figure below shows an example on how the pins should be connected for an ATmega32 on the STK500. Remember to remove the reset jumper on the STK500 if the reset pin is going to be controlled from the AVR Dragon.

Note: Add-on cards for the STK500 like e.g. STK501/502 may have a dedicated JTAG connector.

Example: Connecting AVR Dragon to STK500 with ATmega32



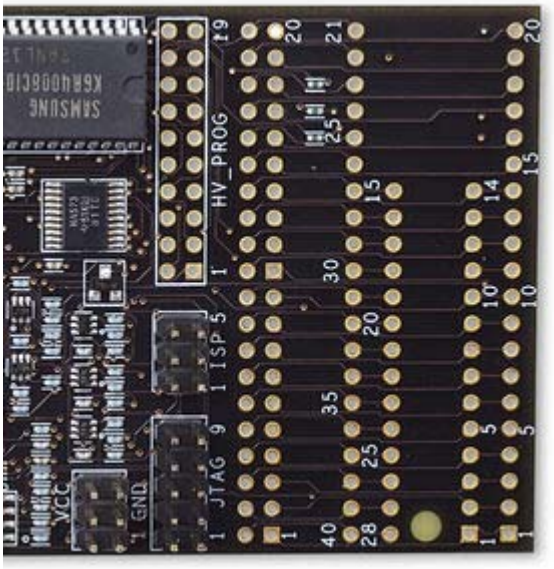
STK500 JTAG Adapter

The STK500 JTAG Adapter, that comes with the JTAGICE mkII, can be used to simplify the connection to the STK500 for AVR devices with JTAG that mates with socket SCKT3100A3 and SCKT3000D3 on the STK500.

Enabling the JTAG Enable fuse

If the JTAGEN fuse (JTAG Enable) in the target device is un-programmed, the JTAG Interface will be disabled. This fuse cannot be programmed through the JTAG Interface and must therefore be programmed through the ISP Interface or High Voltage Serial or Parallel Interface.

Using the Onboard Prototype Area

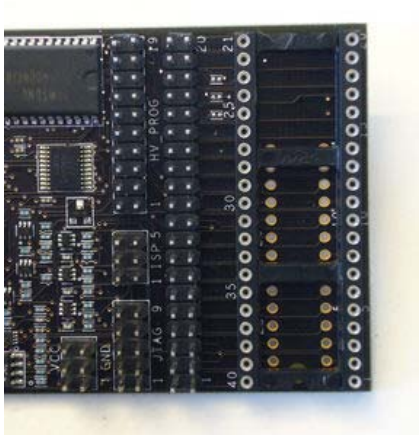


The AVR Dragon has layout for a 40-pin and a 28-pin PDIP socket. The DIP socket pins are connected directly to the 40-pin Header connector. By strapping the ISP, JTAG, HV_PROG and VCC header signals to the 40-pin header connector programming or emulation can be preformed without the need for an external target board.

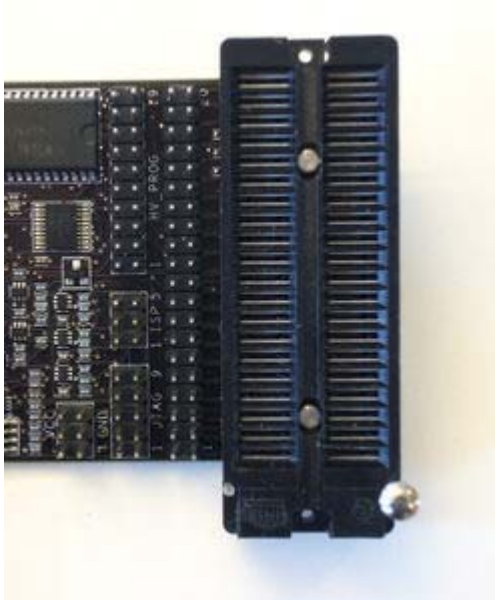
This section shows how to strap the AVR Dragon for different operation modes. Each supported AVR device has its own Device Connection Sheet containing all information required.

There is a number of ways to utilize the prototype area. If only one device type / programming mode is to be used, the easiest and cheapest way is to just solder wires directly from the HV_PROG, ISP, JTAG and VCC headers to the EXPAND header. Howerver, to make the board more flexible header connectors can be soldered in to allow using cables to be connected without soldering.

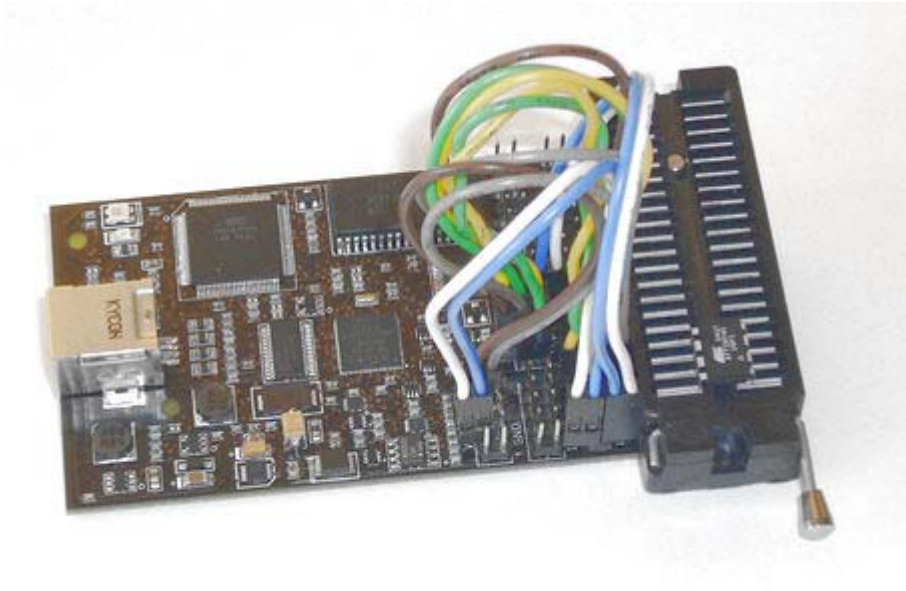
Here is a suggestion how to modify the AVR Dragon board to make it flexible and able to use all DIP socket AVR devices.



In this picture one 20-pin header connector, a 40-pin header connector and a 40-pin DIP socket has been soldered onto the AVR Dragon.



To make it even more flexible and allow for narrow DIP packages, a ZIF (Zero Insertion Force) DIP socket has been added in the picture above. Additional sockets can be bought from third party vendors to support MLF/QFN, TQFP, SOIC etc packages. (Link: http://www.atmel.com/products/AVR/thirdparty.asp#adapters)



- And finally the complete setup for debugWIRE and ISP programming of the ATtiny45. For details on how this is connected please have a look at the ATtiny45 Device connection sheet

Devicesheet: SCKT3000D3

Supported Programming Modes: **ISP**, **Parallel Programming**, **JTAG Prog**
Supported Emulation Modes: **JTAG** for devices with less than 64kB Flash
Applicable for the following devices :

- **ATmega8515** Note: JTAG programming and emulation not supported by ATmega8515
- **ATmega162**

In System Programming:

VCC

6	**B**	nc	5
4	**G**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

21	nc	**F**	20
22	nc	nc	19
23	nc	nc	18
24	nc	nc	17
25	nc	nc	16
26	nc	nc	15
27	nc	nc	14
28	nc	nc	13
29	nc	nc	12
30	nc	nc	11
31	nc	nc	10
32	nc	**E**	9
33	nc	**C**	8
34	nc	**A**	7
35	nc	**D**	6
36	nc	nc	5
37	nc	nc	4
38	nc	nc	3
39	nc	nc	2
40	**G**	nc	1

JTAG Programming and JTAG Emulation

VCC

6	**D**	**H**	5
4	**I**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

JTAG

10	**H**	**G**	9
8	nc	nc	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

DEVICE

21	nc	**B**	20
22	nc	nc	19
23	nc	nc	18
24	nc	nc	17
25	**A**	nc	16
26	**E**	nc	15
27	**C**	nc	14
28	**G**	nc	13
29	nc	nc	12
30	nc	nc	11
31	nc	nc	10
32	nc	**F**	9
33	nc	nc	8
34	nc	nc	7
35	nc	nc	6
36	nc	nc	5
37	nc	nc	4
38	nc	nc	3
39	nc	nc	2
40	**I**	nc	1

Parallel Programming

VCC

6	**U**	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	**T**	**S**	19
18	**R**	**Q**	17
16	**P**	**O**	15
14	**N**	**M**	13
12	**L**	**K**	11
10	**J**	**I**	9
8	**H**	**G**	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**U**	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

21	nc	**T**	20
22	nc	**Q**	19
23	nc	nc	18
24	nc	**P**	17
25	nc	**O**	16
26	nc	**N**	15
27	nc	**M**	14
28	nc	**L**	13
29	nc	**K**	12
30	nc	**J**	11
31	nc	nc	10
32	nc	**R**	9
33	nc	**H**	8
34	nc	**G**	7
35	nc	**F**	6
36	nc	**E**	5
37	nc	**D**	4
38	nc	**C**	3
39	**I**	**B**	2
40	**S**	**A**	1

Devicesheet: SCKT3100A3

Supported Programming Modes: **ISP**, **Parallel Programming**, **JTAG Prog**
Supported Emulation Modes: **JTAG** for devices with less than 64kB Flash
Applicable for the following devices :

- **ATmega16**
- **ATmega32**
- **ATmega164P**
- **ATmega324P**
- **ATmega644**
- **ATmega644P**
- **ATmega1284P**
- **ATmega8535** Note: JTAG programming and emulation not supported on ATmega8535

In System Programming:

VCC

6	**B**	nc	5
4	**G**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

21	nc	nc	20
22	nc	nc	19
23	nc	nc	18
24	nc	nc	17
25	nc	nc	16
26	nc	nc	15
27	nc	nc	14
28	nc	nc	13
29	nc	nc	12
30	nc	**F**	11
31	nc	**G**	10
32	nc	**E**	9
33	nc	**C**	8
34	nc	**A**	7
35	nc	**D**	6
36	nc	nc	5
37	nc	nc	4
38	nc	nc	3
39	nc	nc	2
40	nc	nc	1

JTAG Programming and JTAG Emulation

VCC

6	**D**	**H**	5
4	**I**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	nc	nc	5
4	nc	nc	3

DEVICE

21	nc	nc	20
22	nc	nc	19
23	nc	nc	18
24	**A**	nc	17
25	**E**	nc	16
26	**C**	nc	15
27	**G**	nc	14
28	nc	nc	13
29	nc	nc	12
30	nc	**B**	11
31	nc	**I**	10
32	nc	**F**	9
33	nc	nc	8

2	nc	nc	1

JTAG

10	**H**	**G**	9
8	nc	nc	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

34	nc	nc	7
35	nc	nc	6
36	nc	nc	5
37	nc	nc	4
38	nc	nc	3
39	nc	nc	2
40	nc	nc	1

Parallel Programming

HV_PROG

20	**T**	**S**	19
18	**R**	**Q**	17
16	**P**	**O**	15
14	**N**	**M**	13
12	**L**	**K**	11
10	**J**	**I**	9
8	**H**	**G**	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**U**	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

VCC

6	**U**	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

21	**P**	**O**	20
22	nc	**N**	19
23	nc	**M**	18
24	nc	**L**	17
25	nc	**K**	16
26	nc	**J**	15
27	nc	nc	14
28	nc	**Q**	13
29	nc	nc	12
30	nc	**T**	11
31	nc	**S**	10
32	nc	**R**	9
33	nc	**H**	8
34	nc	**G**	7
35	nc	**F**	6
36	nc	**E**	5
37	nc	**D**	4
38	nc	**C**	3
39	nc	**B**	2
40	**I**	**A**	1

Devicesheet: SCKT3200A2

Supported Programming Modes: **ISP**, **Parallel Programming**
Supported Emulation modes: **debugWIRE**
Applicable for the following devices :

- **ATmega8** - Note: debugWIRE not supported by ATmega8
- **ATmega48**
- **ATmega48P**
- **ATmega88**
- **ATmega168**
- **ATtiny48**

In System Programming and debugWIRE emulation

VCC

6	**B**	nc	5
4	**G**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
15	nc	nc	14
16	nc	nc	13
17	**D**	nc	12
18	**A**	nc	11
19	**C**	nc	10
20	nc	nc	9
21	nc	**F**	8
22	nc	**G**	7
23	nc	nc	6
24	nc	nc	5
25	nc	nc	4
26	nc	nc	3
27	nc	nc	2
28	nc	**E**	1

Parallel Programming

VCC

6	**U**	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	**T**	**S**	19
18	**R**	**Q**	17
16	**P**	**O**	15
14	**N**	**M**	13
12	**L**	**K**	11
10	**J**	**I**	9
8	**H**	**G**	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**U**	nc	1

JTAG

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
15	**B**	**A**	14
16	**C**	**P**	13
17	**D**	**O**	12
18	**E**	**N**	11
19	**F**	nc	10
20	nc	**Q**	9
21	nc	**T**	8
22	nc	**S**	7
23	**G**	**M**	6

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

24	**H**	**L**	5
25	**I**	**K**	4
26	nc	**J**	3
27	nc	nc	2
28	nc	**R**	1

Devicesheet: SCKT3300D3

Supported Programming Modes: **ISP**, **Parallel Programming**
Supported Emulation Modes: **debugWIRE**
Applicable for the following devices :

- **ATtiny2313**

In System Programming and debugWIRE emulation

VCC

6	B	nc	5
4	G	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	F	E	5
4	D	C	3
2	B	A	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
11	nc	F	10
12	nc	nc	9
13	nc	nc	8
14	nc	nc	7
15	nc	nc	6
16	nc	nc	5
17	D	nc	4
18	A	nc	3
19	C	nc	2
20	G	E	1

Parallel Programming

VCC

6	U	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	T	S	19
18	R	Q	17
16	nc	O	15
14	N	M	13
12	L	K	11
10	J	nc	9
8	H	G	7
6	F	E	5
4	D	C	3
2	B	A	1

ISP

6	nc	nc	5
4	nc	nc	3
2	U	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
11	O	T	10
12	A	N	9
13	B	M	8
14	C	L	7
15	D	K	6
16	E	Q	5
17	F	nc	4
18	G	J	3
19	H	nc	2
20	S	R	1

Devicesheet: SCKT3400D1

Supported programming modes: **ISP**, **High Voltage Serial Programming**
Supported emulation modes: **debugWIRE**
Applicable for the following devices :

- **ATtiny11** - Note: ISP and debugWIRE not supported by ATtiny11
- **ATtiny12** - Note: debugWIRE not supoported by ATtiny12
- **ATtiny13**
- **ATtiny25**
- **ATtiny45**
- **ATtiny85**

High Voltage Serial Programming

VCC

6	A	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	H	G	19
18	F	E	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	D	3
2	C	B	1

ISP

6	nc	nc	5
4	nc	nc	3
2	A	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
5	B	H	4
6	C	nc	3
7	D	E	2
8	G	F	1

In System Programming and debugWIRE emulation:

VCC

6	A	nc	5
4	G	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	E	F	5
4	B	D	3
2	A	C	1

JTAG

DEVICE

nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc
nc	nc

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

	nc	nc	
5	**B**	**E**	4
6	**C**	nc	3
7	**D**	nc	2
8	**G**	**F**	1

Devicesheet: SCKT3500D-

Supported Programming Modes: **Parallel Programming**
Supported Emulation modes: **none**
Applicable for the following devices :

- **ATtiny28** Note: Only High Voltage Programming available for ATtiny28

Parallel Programming

VCC

6	**U**	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	**T**	**S**	19
18	**R**	**Q**	17
16	nc	**O**	15
14	**N**	**M**	13
12	**L**	**K**	11
10	**J**	nc	9
8	**H**	**G**	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**U**	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
15	**B**	**A**	14
16	**C**	nc	13
17	**D**	**O**	12
18	**E**	**N**	11
19	**F**	nc	10
20	nc	**Q**	9
21	nc	nc	8
22	**T**	**S**	7
23	**G**	**M**	6
24	**H**	**L**	5
25	nc	**K**	4
26	nc	**J**	3
27	nc	nc	2
28	nc	**R**	1

Devicesheet: SCKT3600A1

Supported Programming Modes: **ISP**, **High Voltage Serial Programming**
Supported Emulation modes: **None**
Applicable for the following devices :

- **ATtiny15** Note: Emulation not supported by ATtiny15

In System Programming

VCC

6	**B**	nc	5
4	**G**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
5	**D**	**F**	4
6	**A**	nc	3
7	**C**	nc	2
8	**G**	**E**	1

High Voltage Serial Programming

VCC

6	**H**	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	**G**	**F**	19
18	**E**	**D**	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**H**	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
5	**A**	**G**	4
6	**B**	**D**	3
7	**C**	nc	2
8	**F**	**E**	1

Devicesheet: SCKT3700A1

Supported Programming Modes: **ISP**, **Parallel Programming**
Supported Emulation modes: **debugWIRE**
Applicable for the following devices :

- **ATtiny26** - Note: debugWIRE not supported by ATtiny26, PP - prog issue, see known issues section
- **ATtiny261**
- **ATtiny461**
- **ATtiny861**

In System Programming and debugWIRE emulation

VCC

6	**B**	nc	5
4	**G**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
11	nc	**E**	10
12	nc	nc	9
13	nc	nc	8
14	nc	nc	7
15	nc	F	6
16	nc	**G**	5
17	nc	nc	4
18	nc	**C**	3
19	nc	**A**	2
20	nc	**D**	1

Parallel Programming

VCC

6	**S**	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	**R**	**Q**	19
18	**P**	**O**	17
16	**N**	**M**	15
14	**L**	**K**	13
12	**J**	**I**	11
10	nc	nc	9
8	**H**	**G**	7
6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**S**	nc	1

JTAG

10	nc	nc	9

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
11	**H**	**P**	10
12	**G**	**M**	9
13	**F**	**N**	8
14	**E**	**O**	7
15	nc	**R**	6
16	nc	**Q**	5

8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

17	**D**	**L**	4
18	**C**	**K**	3
19	**B**	**J**	2
20	**A**	**I**	1

Devicesheet: SCKT244484

Supported Programming Modes: **ISP**, **High Voltage Serial Programming**
Supported Emulation modes: **debugWIRE**
Applicable for the following devices :

- **ATtiny24**
- **ATtiny44**
- **ATtiny84**

In System Programming and debugWIRE emulation

VCC

6	**B**	nc	5
4	**G**	nc	3
2	nc	nc	1

HV_PROG

20	nc	nc	19
18	nc	nc	17
16	nc	nc	15
14	nc	nc	13
12	nc	nc	11
10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

ISP

6	**F**	**E**	5
4	**D**	**C**	3
2	**B**	**A**	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3
2	nc	nc	1

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
8	**A**	**D**	7
9	**C**	nc	6
10	nc	nc	5
11	nc	**E**	4
12	nc	nc	3
13	nc	nc	2
14	**F**	**G**	1

High Voltage Serial Programming

VCC

6	**U**	nc	5
4	nc	nc	3
2	nc	nc	1

HV_PROG

20	**J**	**I**	19
18	**H**	**G**	17
16	nc	nc	15
14	nc	nc	13
12	nc	**F**	11
10	**E**	**D**	9
8	nc	nc	7
6	nc	nc	5
4	nc	**C**	3
2	**B**	**A**	1

ISP

6	nc	nc	5
4	nc	nc	3
2	**U**	nc	1

JTAG

10	nc	nc	9
8	nc	nc	7
6	nc	nc	5
4	nc	nc	3

DEVICE

	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
	nc	nc	
8	**B**	**A**	7
9	**C**	nc	6
10	nc	nc	5
11	**F**	**H**	4
12	**E**	nc	3
13	**D**	**G**	2

2	nc	nc	1	14	**J**	**I**	1

Devicesheet: off board targets

Supported Programming Modes: **ISP**, **Parallel Programming**, **JTAG Prog**
Supported Emulation Modes: **debugWIRE**, **JTAG**

This Device is not available in PDIP package and can therefor not directly be connected to the AVR Dragon Prototype area. See the applicable datasheet for information on how to connect for ISP, PP and JTAG

Troubleshooting

Problem	Reason	Solution
Signature Bytes read as 0x00 0x00 0x00	ISP Frequency is to high.	Lower ISP freq under the board settings
Not able to communicate with device through debugWIRE	RESET pullup resistor to small	Remove or increase the pull-up value to 10K ohm or more.
Not able to communicate with device through debugWIRE	Decoupling Capasitor destroys communication on RESET line	Remove Decoupling capasitor on reset line during debugWIRE emulation
debugWIRE communication fails when using STK500	RESET line strongly tied to VCC	Remove RESET jumper on STK500 to allow AVR Dragon to control the line.
After successfully enabling the DWEN fuse, AVR Dragon is not able to enter debug mode	RESET line strongly tied to VCC	Remove RESET jumper on STK500 to allow AVR Dragon to control the line.
Target voltage is read as 0V for **on-board** targets	AVR Dragon get no reference voltage to the target voltage sensing. Target voltage is sensed from either pin 2 on ISP header or pin 4 on the JTAG header.	In order to get reference voltage to the level converters of AVR Dragon, connect Pin 2 or 4 or 6 on the VCC header to pin 2 on ISP header or pin 4 on the JTAG header. **Note that VCC pins on AVR Dragon are set to 5V and this procedure should therefore not be used for off-board targets.**
Target voltage is read as 0V for **off-board** targets.	AVR Dragon get no reference voltage to the target voltage sensing. Target voltage is sensed from either pin 2 on ISP header or pin 4 on the JTAG header.	Connect the target voltage from the target board to pin 2 on ISP header or pin 4 on the JTAG header.
Not able to set ISP frequency	AVR Dragon is not reading any target voltage.	See above
ISP/PP/HVSP programming fails	AVR Dragon is not reading any target voltage	See above
Not able to connect to AVR Dragon, and status led is yellow	AVR Dragon firmware upgrade failed, or firmware is corrupt.	Go to the menu "Tools->Upgrade AVR Dragon" to perform a manual upgrade.

Known issues

- JTAG communication with packages in PDIP which have the CKOUT fuse enabled and running above 3.5V may be unstable if there is long wiring from the AVR Dragon to the PDIP AVR.

· ATtiny43U Programming issue:
Parallel Programming may fail if the Internal RC Osc fuse is set with the shortest startup time.

· ATtiny84 Programming issue:
Parallel Programming may fail on the ATtiny84 if both the DWEN fuse and any of the external clock fuses at enabled at the same time.
Workaround is to use the STK500 platform for Parallel Programming of this part.

· ATtiny26 Programming issue:
Parallel Programming on ATtiny26 is unstable with the AVR Dragon
Workaround is to use the STK500 platform for Parallel Programming of this part.

· An error message will be triggered when entering the Board tab in the programming dialog, if there is no target voltage present.

· In order to set ISP frequency, AVR Dragon needs to sense target voltage. See trouble shooting guide

AVR Dragon Requirements

System Unit
Physical Dimensions..... (H x W x D) 53 x 105 x 15 mm
Power Voltage Requirements..... 5.0V USB powered
AVR Dragon Current Consumption..... 150mA
Max Current Source Capability (to target)..... 300mA
Ambient Temperature.....0-70'C

Operation
Target Voltage Range.....1.8 - 5.5 V

I/O Pins
Maximum Pull-up on ISP/JTAG header..... 1K
Maximum Pull-down on ISP/JTAG header..... 10K

Maximum Source Current VCC header..... up to total 300mA.

Note that the AVR Dragon requires a USB port that can deliver up to 500mA. (self-powered USB hub)

Technical Support

Before requesting techical support make sure you have the latest available AVR Studio, tool firmware installed. The latest AVR Studio version can be downloaded from www.atmel.com/avrstudio, and contains the latest firmware version for all Atmel AVR tools. When connecting your tool, AVR Studio will automatically check the firmware version and request an update if needed.

For technical support please contact avr@atmel.com. When requesting technical support for AVR Dragon please include the following information:

- Version number of AVR Studio. This can be found in AVR studio menu "Help->About"
- PC processor type and speed
- PC operating system and version
- What target AVR device is used (Complete part number)
- Fuse settings on the AVR
- Target clock frequency
- If CLKPR (Clock Prescaler Register) is used (for AVRs with this feature)
- Target voltage
- Programming speed
- A detailed description of the problem, and how to recreate it.
- Any error or waring information generated by AVR Studio when the error occurred.