

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: [info@chipsmall.com](mailto:info@chipsmall.com) Web: [www.chipsmall.com](http://www.chipsmall.com)

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

## The Atmel-ICE Debugger



Atmel-ICE is a powerful development tool for debugging and programming ARM® Cortex®-M based Atmel® SAM and Atmel AVR® microcontrollers with On-Chip Debug capability.

It supports:

- Programming and on-chip debugging of all Atmel AVR 32-bit microcontrollers on both JTAG and aWire interfaces
- Programming and on-chip debugging of all Atmel AVR XMEGA® family devices on both JTAG and PDI 2-wire interfaces
- Programming (JTAG, SPI, UPDI) and debugging of all Atmel AVR 8-bit microcontrollers with OCD support on either JTAG, debugWIRE or UPDI interfaces
- Programming and debugging of all Atmel SAM ARM Cortex-M based microcontrollers on both SWD and JTAG interfaces
- Programming (TPI) of all Atmel tinyAVR® 8-bit microcontrollers with support for this interface

Consult the supported devices list in the Atmel Studio User Guide for a full list of devices and interfaces supported by this firmware release.

## Table of Contents

---

The Atmel-ICE Debugger.....	1
1. Introduction.....	5
1.1. Introduction to the Atmel-ICE.....	5
1.2. Atmel-ICE Features.....	5
1.3. System Requirements.....	6
2. Getting Started with the Atmel-ICE.....	7
2.1. Full Kit Contents.....	7
2.2. Basic Kit Contents.....	7
2.3. PCBA Kit Contents.....	8
2.4. Spare Parts Kits.....	8
2.5. Kit Overview.....	9
2.6. Assembling the Atmel-ICE.....	10
2.7. Opening the Atmel-ICE.....	11
2.8. Powering the Atmel-ICE.....	13
2.9. Connecting to the Host Computer.....	13
2.10. USB Driver Installation.....	13
2.10.1. Windows.....	13
3. Connecting the Atmel-ICE.....	15
3.1. Connecting to AVR and SAM Target Devices.....	15
3.2. Connecting to a JTAG Target.....	16
3.3. Connecting to an aWire Target.....	17
3.4. Connecting to a PDI Target.....	18
3.5. Connecting to a UPDI Target.....	18
3.6. Connecting to a debugWIRE Target.....	19
3.7. Connecting to an SPI Target.....	20
3.8. Connecting to a TPI Target.....	21
3.9. Connecting to an SWD Target.....	22
3.10. Connecting to Data Gateway Interface.....	23
4. On-chip Debugging.....	24
4.1. Introduction.....	24
4.2. SAM Devices with JTAG/SWD.....	24
4.2.1. ARM CoreSight Components.....	24
4.2.2. JTAG Physical Interface.....	25
4.2.3. Connecting to a JTAG Target.....	27
4.2.4. SWD Physical Interface.....	28
4.2.5. Connecting to an SWD Target.....	28
4.2.6. Special Considerations.....	29
4.3. AVR UC3 Devices with JTAG/aWire.....	29
4.3.1. Atmel AVR UC3 On-chip Debug System.....	30
4.3.2. JTAG Physical Interface.....	30
4.3.3. Connecting to a JTAG Target.....	32

4.3.4.	aWire Physical Interface.....	33
4.3.5.	Connecting to an aWire Target.....	34
4.3.6.	Special Considerations.....	35
4.3.7.	EVTI / EVTO Usage.....	35
4.4.	tinyAVR, megaAVR, and XMEGA Devices.....	35
4.4.1.	JTAG Physical Interface.....	36
4.4.2.	Connecting to a JTAG Target.....	36
4.4.3.	SPI Physical Interface.....	38
4.4.4.	Connecting to an SPI Target.....	38
4.4.5.	PDI.....	39
4.4.6.	Connecting to a PDI Target.....	39
4.4.7.	UPDI Physical Interface.....	40
4.4.8.	Connecting to a UPDI Target.....	41
4.4.9.	TPI Physical Interface.....	42
4.4.10.	Connecting to a TPI Target.....	42
4.4.11.	Advanced Debugging (AVR JTAG /debugWIRE devices).....	43
4.4.12.	megaAVR Special Considerations.....	44
4.4.13.	AVR XMEGA Special Considerations.....	45
4.4.14.	debugWIRE Special Considerations.....	46
4.4.15.	debugWIRE Software Breakpoints.....	47
4.4.16.	Understanding debugWIRE and the DWEN Fuse.....	47
4.4.17.	TinyX-OCD (UPDI) Special Considerations.....	48
<b>5.</b>	<b>Hardware Description.....</b>	<b>50</b>
5.1.	LEDs.....	50
5.2.	Rear Panel.....	50
5.3.	Bottom Panel.....	51
5.4.	Architecture Description.....	51
5.4.1.	Atmel-ICE Main Board.....	51
5.4.2.	Atmel-ICE Target Connectors.....	52
5.4.3.	Atmel-ICE Target Connectors Part Numbers.....	52
<b>6.</b>	<b>Software Integration.....</b>	<b>53</b>
6.1.	Atmel Studio.....	53
6.1.1.	Software Integration in Atmel Studio.....	53
6.1.2.	Programming Options.....	53
6.1.3.	Debug Options.....	53
6.2.	Command Line Utility.....	54
<b>7.</b>	<b>Advanced Debugging Techniques.....</b>	<b>55</b>
7.1.	Atmel AVR UC3 Targets.....	55
7.1.1.	EVTI / EVTO Usage.....	55
7.2.	debugWIRE Targets.....	55
7.2.1.	debugWIRE Software Breakpoints.....	55
<b>8.</b>	<b>Release History and Known issues.....</b>	<b>57</b>
8.1.	Firmware Release History.....	57
8.2.	Known Issues Concerning the Atmel-ICE.....	57
8.2.1.	General.....	57

8.2.2.	Atmel AVR XMEGA OCD Specific Issues.....	57
8.2.3.	Atmel AVR - Device Specific Issues.....	58
9.	Product Compliance.....	59
9.1.	RoHS and WEEE.....	59
9.2.	CE and FCC.....	59
10.	Revision History.....	60

# 1. Introduction

## 1.1. Introduction to the Atmel-ICE

Atmel-ICE is a powerful development tool for debugging and programming ARM Cortex-M based Atmel SAM and Atmel AVR microcontrollers with On-Chip Debug capability.

It supports:

- Programming and on-chip debugging of all Atmel AVR UC3 microcontrollers on both JTAG and aWire interfaces
- Programming and on-chip debugging of all AVR XMEGA family devices on both JTAG and PDI 2-wire interfaces
- Programming (JTAG and SPI) and debugging of all AVR 8-bit microcontrollers with OCD support on both JTAG or debugWIRE interfaces
- Programming and debugging of all Atmel SAM ARM Cortex-M based microcontrollers on both SWD and JTAG interfaces
- Programming (TPI) of all Atmel tinyAVR 8-bit microcontrollers with support for this interface

## 1.2. Atmel-ICE Features

- Fully compatible with Atmel Studio
- Supports programming and debugging of all Atmel AVR UC3 32-bit microcontrollers
- Supports programming and debugging of all 8-bit AVR XMEGA devices
- Supports programming and debugging of all 8-bit Atmel megaAVR<sup>®</sup> and tinyAVR devices with OCD
- Supports programming and debugging of all SAM ARM Cortex-M based microcontrollers
- Target operating voltage range of 1.62V to 5.5V
- Draws less than 3mA from target VTref when using debugWIRE interface and less than 1mA for all other interfaces
- Supports JTAG clock frequencies from 32kHz to 7.5MHz
- Supports PDI clock frequencies from 32kHz to 7.5MHz
- Supports debugWIRE baud rates from 4kbit/s to 0.5Mbit/s
- Supports aWire baud rates from 7.5kbit/s to 7Mbit/s
- Supports SPI clock frequencies from 8kHz to 5MHz
- Supports UPDI baud rates from up to 750kbit/s
- Supports SWD clock frequencies from 32kHz to 10MHz
- USB 2.0 high-speed host interface
- ITM serial trace capture at up to 3MB/s
- Supports DGI SPI and USART interfaces when not debugging or programming
- Supports 10-pin 50-mil JTAG connector with both AVR and Cortex pinouts. The standard probe cable supports AVR 6-pin ISP/PDI/TPI 100-mil headers as well as 10-pin 50-mil. An adapter is available to support 6-pin 50-mil, 10-pin 100-mil, and 20-pin 100-mil headers. Several kit options are available with different cabling and adapters.

### 1.3. System Requirements

The Atmel-ICE unit requires that a front-end debugging environment Atmel Studio version 6.2 or later is installed on your computer.

The Atmel-ICE should be connected to the host computer using the USB cable provided, or a certified Micro-USB cable.

## 2. Getting Started with the Atmel-ICE

### 2.1. Full Kit Contents

The Atmel-ICE full kit contains these items:

- Atmel-ICE unit
- USB cable (1.8m, high-speed, Micro-B)
- Adapter board containing 50-mil AVR, 100-mil AVR/SAM, and 100-mil 20-pin SAM adapters
- IDC flat cable with 10-pin 50-mil connector and 6-pin 100-mil connector
- 50-mil 10-pin mini squid cable with 10 x 100-mil sockets

Figure 2-1. Atmel-ICE Full Kit Contents



### 2.2. Basic Kit Contents

The Atmel-ICE basic kit contains these items:

- Atmel-ICE unit
- USB cable (1.8m, high-speed, Micro-B)
- IDC flat cable with 10-pin 50-mil connector and 6-pin 100-mil connector

Figure 2-2. Atmel-ICE Basic Kit Contents

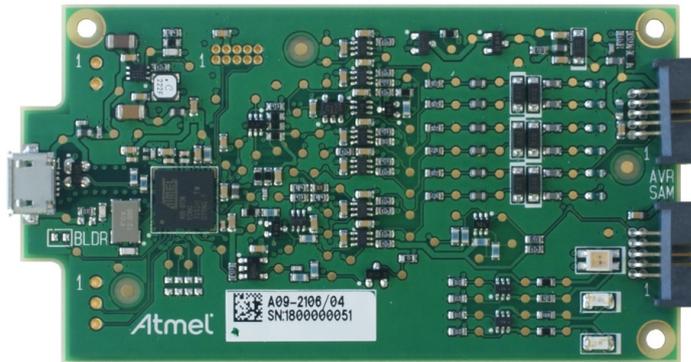


### 2.3. PCBA Kit Contents

The Atmel-ICE PCBA kit contains these items:

- Atmel-ICE unit without plastic encapsulation

Figure 2-3. Atmel-ICE PCBA Kit Contents



### 2.4. Spare Parts Kits

The following spare parts kits are available:

- Adapter kit
- Cable kit

Figure 2-4. Atmel-ICE Adapter Kit Contents



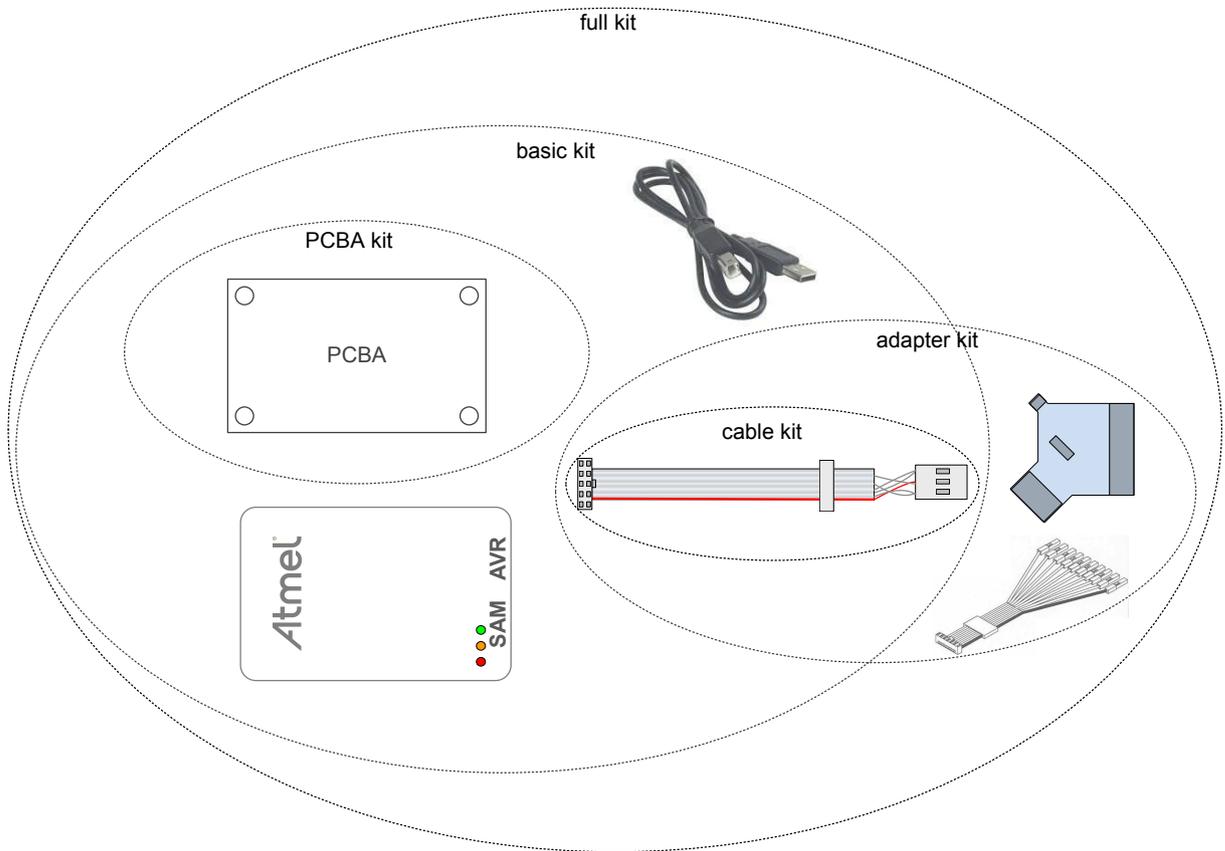
Figure 2-5. Atmel-ICE Cable Kit Contents



## 2.5. Kit Overview

The Atmel-ICE kit options are shown diagrammatically here:

Figure 2-6. Atmel-ICE Kit Overview



## 2.6. Assembling the Atmel-ICE

The Atmel-ICE unit is shipped with no cables attached. Two cable options are provided in the full kit:

- 50-mil 10-pin IDC flat cable with 6-pin ISP and 10-pin connectors
- 50-mil 10-pin mini-squid cable with 10 x 100-mil sockets

Figure 2-7. Atmel-ICE Cables



For most purposes, the 50-mil 10-pin IDC flat cable can be used, connecting either natively to its 10-pin or 6-pin connectors, or connecting via the adapter board. Three adapters are provided on one small PCBA. The following adapters are included:

- 100-mil 10-pin JTAG/SWD adapter
- 100-mil 20-pin SAM JTAG/SWD adapter
- 50-mil 6-pin SPI/debugWIRE/PDI/aWire adapter

Figure 2-8. Atmel-ICE Adapters



### Note:

A 50-mil JTAG adapter has not been provided - this is because the 50-mil 10-pin IDC cable can be used to connect directly to a 50-mil JTAG header. For the part number of the component used for the 50-mil 10-pin connector, see [Atmel-ICE Target Connectors Part Numbers](#).

The 6-pin ISP/PDI header is included as part of the 10-pin IDC cable. This termination can be cut off if it is not required.

To assemble your Atmel-ICE into its default configuration, connect the 10-pin 50-mil IDC cable to the unit as shown below. Be sure to orient the cable so that the red wire (pin 1) on the cable aligns with the triangular indicator on the blue belt of the enclosure. The cable should connect upwards from the unit. Be sure to connect to the port corresponding to the pinout of your target - AVR or SAM.

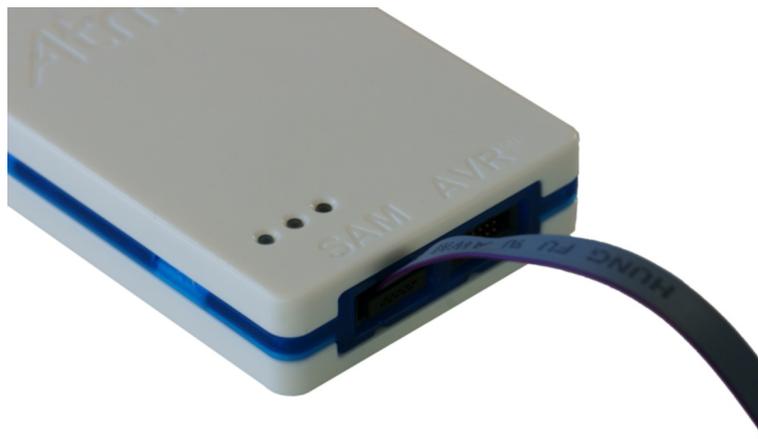
Figure 2-9. Atmel-ICE Cable Connection



Figure 2-10. Atmel-ICE AVR Probe Connection



Figure 2-11. Atmel-ICE SAM Probe Connection



## 2.7. Opening the Atmel-ICE

### Note:

For normal operation, the Atmel-ICE unit must not be opened. Opening the unit is done at your own risk. Anti-static precautions should be taken.

The Atmel-ICE enclosure consists of three separate plastic components - top cover, bottom cover, and blue belt - which are snapped together during assembly. To open the unit, simply insert a large flat

screwdriver into the openings in the blue belt, apply some inward pressure and twist gently. Repeat the process on the other snapper holes, and the top cover will pop off.

**Figure 2-12. Opening the Atmel-ICE (1)**



**Figure 2-13. Opening the Atmel-ICE (2)**



**Figure 2-14. Opening the Atmel-ICE(3)**



To close the unit again, simply align the top and bottom covers correctly, and press firmly together.

## **2.8. Powering the Atmel-ICE**

The Atmel-ICE is powered by the USB bus voltage. It requires less than 100mA to operate, and can therefore be powered through a USB hub. The power LED will illuminate when the unit is plugged in. When not connected in an active programming or debugging session, the unit will enter low-power consumption mode to preserve your computer's battery. The Atmel-ICE cannot be powered down - it should be unplugged when not in use.

## **2.9. Connecting to the Host Computer**

The Atmel-ICE communicates primarily using a standard HID interface, and does not require a special driver on the host computer. To use the advanced Data Gateway functionality of the Atmel-ICE, be sure to install the USB driver on the host computer. This is done automatically when installing the front-end software provided free by Atmel. See [www.atmel.com](http://www.atmel.com) for further information or to download the latest front-end software.

The Atmel-ICE must be connected to an available USB port on the host computer using the USB cable provided, or suitable USB certified micro cable. The Atmel-ICE contains a USB 2.0 compliant controller, and can operate in both full-speed and high-speed modes. For best results, connect the Atmel-ICE directly to a USB 2.0 compliant high-speed hub on the host computer using the cable provided.

## **2.10. USB Driver Installation**

### **2.10.1. Windows**

When installing the Atmel-ICE on a computer running Microsoft® Windows®, the USB driver is loaded when the Atmel-ICE is first plugged in.

**Note:**

Be sure to install the front-end software packages before plugging the unit in for the first time.

Once successfully installed, the Atmel-ICE will appear in the device manager as a "Human Interface Device".

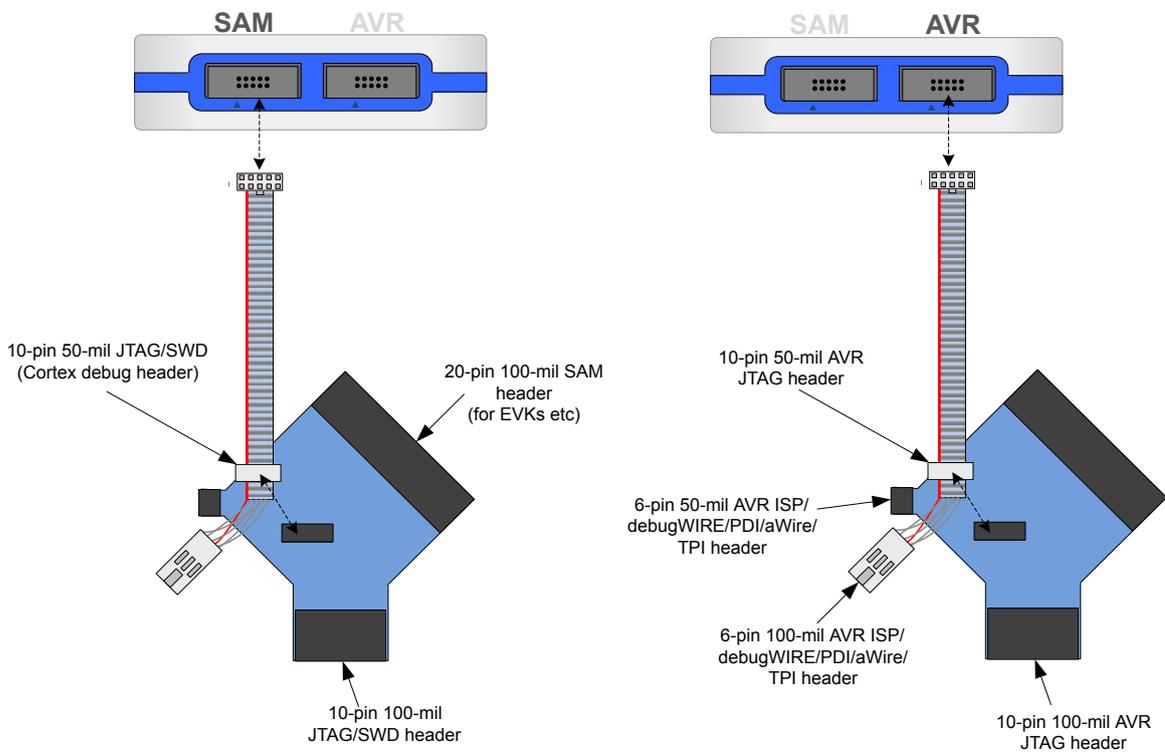
### 3. Connecting the Atmel-ICE

#### 3.1. Connecting to AVR and SAM Target Devices

The Atmel-ICE is equipped with two 50-mil 10-pin JTAG connectors. Both connectors are directly electrically connected, but conform to two different pinouts; the AVR JTAG header and the ARM Cortex Debug header. The connector should be selected based on the pinout of the target board, and not the target MCU type - for example a SAM device mounted in an AVR STK<sup>®</sup>600 stack should use the AVR header.

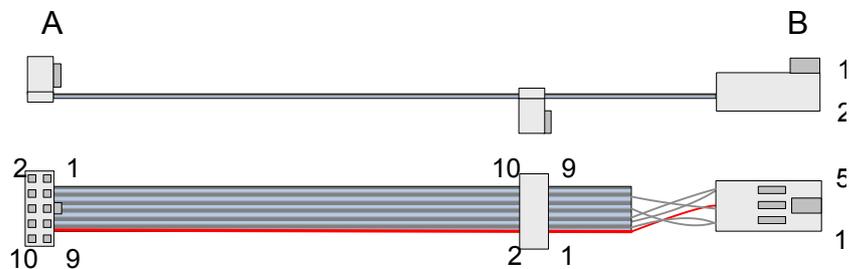
Various cabling and adapters are available in the different Atmel-ICE kits. An overview of connection options is shown.

Figure 3-1. Atmel-ICE Connection Options



The red wire marks pin 1 of the 10-pin 50-mil connector. Pin 1 of the 6-pin 100-mil connector is placed to the right of the keying when the connector is seen from the cable. Pin 1 of each connector on the adapter is marked with a white dot. The figure below shows the pinout of the debug cable. The connector marked A plugs into the debugger while the B side plugs into the target board.

Figure 3-2. Debug Cable Pinout



## 3.2. Connecting to a JTAG Target

The Atmel-ICE is equipped with two 50-mil 10-pin JTAG connectors. Both connectors are directly electrically connected, but conform to two different pinouts; the AVR JTAG header and the ARM Cortex Debug header. The connector should be selected based on the pinout of the target board, and not the target MCU type - for example a SAM device mounted in an AVR STK600 stack should use the AVR header.

The recommended pinout for the 10-pin AVR JTAG connector is shown in [Figure 4-6](#).

The recommended pinout for the 10-pin ARM Cortex Debug connector is shown in [Figure 4-2](#).

### Direct connection to a standard 10-pin 50-mil header

Use the 50-mil 10-pin flat cable (included in some kits) to connect directly to a board supporting this header type. Use the AVR connector port on the Atmel-ICE for headers with the AVR pinout, and the SAM connector port for headers complying with the ARM Cortex Debug header pinout.

The pinouts for both 10-pin connector ports are shown below.

### Connection to a standard 10-pin 100-mil header

Use a standard 50-mil to 100-mil adapter to connect to 100-mil headers. An adapter board (included in some kits) can be used for this purpose, or alternatively the JTAGICE3 adapter can be used for AVR targets.



#### Important:

The JTAGICE3 100-mil adapter cannot be used with the SAM connector port, since pins 2 and 10 (AVR GND) on the adapter are connected.

### Connection to a custom 100-mil header

If your target board does not have a compliant 10-pin JTAG header in 50- or 100-mil, you can map to a custom pinout using the 10-pin "mini-squid" cable (included in some kits), which gives access to ten individual 100-mil sockets.

### Connection to a 20-pin 100-mil header

Use the adapter board (included in some kits) to connect to targets with a 20-pin 100-mil header.

**Table 3-1. Atmel-ICE JTAG Pin Description**

Name	AVR port pin	SAM port pin	Description
TCK	1	4	Test Clock (clock signal from the Atmel-ICE into the target device).
TMS	5	2	Test Mode Select (control signal from the Atmel-ICE into the target device).
TDI	9	8	Test Data In (data transmitted from the Atmel-ICE into the target device).
TDO	3	6	Test Data Out (data transmitted from the target device into the Atmel-ICE).
nTRST	8	-	Test Reset (optional, only on some AVR devices). Used to reset the JTAG TAP controller.

Name	AVR port pin	SAM port pin	Description
nSRST	6	10	Reset (optional). Used to reset the target device. Connecting this pin is recommended since it allows the Atmel-ICE to hold the target device in a reset state, which can be essential to debugging in certain scenarios.
VTG	4	1	Target voltage reference. The Atmel-ICE samples the target voltage on this pin in order to power the level converters correctly. The Atmel-ICE draws less than 3mA from this pin in debugWIRE mode and less than 1mA in other modes.
GND	2, 10	3, 5, 9	Ground. All must be connected to ensure that the Atmel-ICE and the target device share the same ground reference.

### 3.3. Connecting to an aWire Target

The aWire interface requires only one data line in addition to  $V_{CC}$  and GND. On the target this line is the nRESET line, although the debugger uses the JTAG TDO line as the data line.

The recommended pinout for the 6-pin aWire connector is shown in [Figure 4-8](#).

#### Connection to a 6-pin 100-mil aWire header

Use the 6-pin 100-mil tap on the flat cable (included in some kits) to connect to a standard 100-mil aWire header.

#### Connection to a 6-pin 50-mil aWire header

Use the adapter board (included in some kits) to connect to a standard 50-mil aWire header.

#### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR connector port and the target board. Three connections are required, as described in the table below.

**Table 3-2. Atmel-ICE aWire Pin Mapping**

Atmel-ICE AVR port pins	Target pins	Mini-squid pin	aWire pinout
Pin 1 (TCK)		1	
Pin 2 (GND)	GND	2	6
Pin 3 (TDO)	DATA	3	1
Pin 4 (VTG)	VTG	4	2
Pin 5 (TMS)		5	
Pin 6 (nSRST)		6	
Pin 7 (Not connected)		7	
Pin 8 (nTRST)		8	
Pin 9 (TDI)		9	
Pin 10 (GND)		0	

### 3.4. Connecting to a PDI Target

The recommended pinout for the 6-pin PDI connector is shown in [Figure 4-11](#).

#### Connection to a 6-pin 100-mil PDI header

Use the 6-pin 100-mil tap on the flat cable (included in some kits) to connect to a standard 100-mil PDI header.

#### Connection to a 6-pin 50-mil PDI header

Use the adapter board (included in some kits) to connect to a standard 50-mil PDI header.

#### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR connector port and the target board. Four connections are required, as described in the table below.



#### Important:

The pinout required is different from the JTAGICE mkII JTAG probe, where PDI\_DATA is connected to pin 9. The Atmel-ICE is compatible with the pinout used by the Atmel-ICE, JTAGICE3, AVR ONE!, and AVR Dragon™ products.

**Table 3-3. Atmel-ICE PDI Pin Mapping**

Atmel-ICE AVR port pin	Target pins	Mini-squid pin	Atmel STK600 PDI pinout
Pin 1 (TCK)		1	
Pin 2 (GND)	GND	2	6
Pin 3 (TDO)	PDI_DATA	3	1
Pin 4 (VTG)	VTG	4	2
Pin 5 (TMS)		5	
Pin 6 (nSRST)	PDI_CLK	6	5
Pin 7 (not connected)		7	
Pin 8 (nTRST)		8	
Pin 9 (TDI)		9	
Pin 10 (GND)		0	

### 3.5. Connecting to a UPDI Target

The recommended pinout for the 6-pin UPDI connector is shown in [Figure 4-12](#).

#### Connection to a 6-pin 100-mil UPDI header

Use the 6-pin 100-mil tap on the flat cable (included in some kits) to connect to a standard 100-mil UPDI header.

#### Connection to a 6-pin 50-mil UPDI header

Use the adapter board (included in some kits) to connect to a standard 50-mil UPDI header.

### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR connector port and the target board. Three connections are required, as described in the table below.

**Table 3-4. Atmel-ICE UPDI Pin Mapping**

Atmel-ICE AVR port pin	Target pins	Mini-squid pin	Atmel STK600 UPDI pinout
Pin 1 (TCK)		1	
Pin 2 (GND)	GND	2	6
Pin 3 (TDO)	UPDI_DATA	3	1
Pin 4 (VTG)	VTG	4	2
Pin 5 (TMS)		5	
Pin 6 (nSRST)	[/RESET sense]	6	5
Pin 7 (Not connected)		7	
Pin 8 (nTRST)		8	
Pin 9 (TDI)		9	
Pin 10 (GND)		0	

## 3.6. Connecting to a debugWIRE Target

The recommended pinout for the 6-pin debugWIRE (SPI) connector is shown in [Table 3-6](#).

### Connection to a 6-pin 100-mil SPI header

Use the 6-pin 100-mil tap on the flat cable (included in some kits) to connect to a standard 100-mil SPI header.

### Connection to a 6-pin 50-mil SPI header

Use the adapter board (included in some kits) to connect to a standard 50-mil SPI header.

### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR connector port and the target board. Three connections are required, as described in [Table 3-5](#).

Although the debugWIRE interface only requires one signal line (RESET),  $V_{CC}$  and GND to operate correctly, it is advised to have access to the full SPI connector so that the debugWIRE interface can be enabled and disabled using SPI programming.

When the DWEN fuse is enabled the SPI interface is overridden internally in order for the OCD module to have control over the RESET pin. The debugWIRE OCD is capable of disabling itself temporarily (using the button on the debugging tab in the properties dialog in Atmel Studio), thus releasing control of the RESET line. The SPI interface is then available again (only if the SPIEN fuse is programmed), allowing the DWEN fuse to be un-programmed using the SPI interface. If power is toggled before the DWEN fuse is un-programmed, the debugWIRE module will again take control of the RESET pin.

#### Note:

It is highly advised to simply let Atmel Studio handle setting and clearing of the DWEN fuse.

It is not possible to use the debugWIRE interface if the lockbits on the target AVR device 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, and thus clear the lockbits. When the lockbits are cleared the debugWIRE interface will be re-enabled. The SPI Interface is only capable of reading fuses, reading signature and performing a chip erase when the DWEN fuse is un-programmed.

**Table 3-5. Atmel-ICE debugWIRE Pin Mapping**

Atmel-ICE AVR port pin	Target pins	Mini-squid pin
Pin 1 (TCK)		1
Pin 2 (GND)	GND	2
Pin 3 (TDO)		3
Pin 4 (VTG)	VTG	4
Pin 5 (TMS)		5
Pin 6 (nSRST)	RESET	6
Pin 7 (Not connected)		7
Pin 8 (nTRST)		8
Pin 9 (TDI)		9
Pin 10 (GND)		0

### 3.7. Connecting to an SPI Target

The recommended pinout for the 6-pin SPI connector is shown in [Figure 4-10](#).

#### Connection to a 6-pin 100-mil SPI header

Use the 6-pin 100-mil tap on the flat cable (included in some kits) to connect to a standard 100-mil SPI header.

#### Connection to a 6-pin 50-mil SPI header

Use the adapter board (included in some kits) to connect to a standard 50-mil SPI header.

#### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR connector port and the target board. Six connections are required, as described in the table below.



#### Important:

The SPI interface is effectively disabled when the debugWIRE enable fuse (DWEN) is programmed, even if SPIEN fuse is also programmed. To re-enable the SPI interface, the 'disable debugWIRE' command must be issued while in a debugWIRE debugging session. Disabling debugWIRE in this manner requires that the SPIEN fuse is already programmed. If Atmel Studio fails to disable debugWIRE, it is probable because the SPIEN fuse is NOT programmed. If this is the case, it is necessary to use a high-voltage programming interface to program the SPIEN fuse.

**Info:**

The SPI interface is often referred to as "ISP", since it was the first In System Programming interface on Atmel AVR products. Other interfaces are now available for In System Programming.

**Table 3-6. Atmel-ICE SPI Pin Mapping**

Atmel-ICE AVR port pins	Target pins	Mini-squid pin	SPI pinout
Pin 1 (TCK)	SCK	1	3
Pin 2 (GND)	GND	2	6
Pin 3 (TDO)	MISO	3	1
Pin 4 (VTG)	VTG	4	2
Pin 5 (TMS)		5	
Pin 6 (nSRST)	/RESET	6	5
Pin 7 (not connected)		7	
Pin 8 (nTRST)		8	
Pin 9 (TDI)	MOSI	9	4
Pin 10 (GND)		0	

### 3.8. Connecting to a TPI Target

The recommended pinout for the 6-pin TPI connector is shown in [Figure 4-13](#).

#### Connection to a 6-pin 100-mil TPI header

Use the 6-pin 100-mil tap on the flat cable (included in some kits) to connect to a standard 100-mil TPI header.

#### Connection to a 6-pin 50-mil TPI header

Use the adapter board (included in some kits) to connect to a standard 50-mil TPI header.

#### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR connector port and the target board. Six connections are required, as described in the table below.

**Table 3-7. Atmel-ICE TPI Pin Mapping**

Atmel-ICE AVR port pins	Target pins	Mini-squid pin	TPI pinout
Pin 1 (TCK)	CLOCK	1	3
Pin 2 (GND)	GND	2	6
Pin 3 (TDO)	DATA	3	1
Pin 4 (VTG)	VTG	4	2
Pin 5 (TMS)		5	

Atmel-ICE AVR port pins	Target pins	Mini-squid pin	TPI pinout
Pin 6 (nSRST)	/RESET	6	5
Pin 7 (not connected)		7	
Pin 8 (nTRST)		8	
Pin 9 (TDI)		9	
Pin 10 (GND)		0	

### 3.9. Connecting to an SWD Target

The ARM SWD interface is a subset of the JTAG interface, making use of the TCK and TMS pins, which means that when connecting to an SWD device, the 10-pin JTAG connector can technically be used. The ARM JTAG and AVR JTAG connectors are, however, not pin-compatible, so this depends upon the layout of the target board in use. When using an STK600 or a board making use of the AVR JTAG pinout, the AVR connector port on the Atmel-ICE must be used. When connecting to a board, which makes use of the ARM JTAG pinout, the SAM connector port on the Atmel-ICE must be used.

The recommended pinout for the 10-pin Cortex Debug connector is shown in [Figure 4-4](#).

#### Connection to a 10-pin 50-mil Cortex header

Use the flat cable (included in some kits) to connect to a standard 50-mil Cortex header.

#### Connection to a 10-pin 100-mil Cortex-layout header

Use the adapter board (included in some kits) to connect to a 100-mil Cortex-pinout header.

#### Connection to a 20-pin 100-mil SAM header

Use the adapter board (included in some kits) to connect to a 20-pin 100-mil SAM header.

#### Connection to a custom 100-mil header

The 10-pin mini-squid cable should be used to connect between the Atmel-ICE AVR or SAM connector port and the target board. Six connections are required, as described in the table below.

**Table 3-8. Atmel-ICE SWD Pin Mapping**

Name	AVR port pin	SAM port pin	Description
SWDC LK	1	4	Serial Wire Debug Clock.
SWDIO	5	2	Serial Wire Debug Data Input/Output.
SWO	3	6	Serial Wire Output (optional- not implemented on all devices).
nSRST	6	10	Reset.
VTG	4	1	Target voltage reference.
GND	2, 10	3, 5, 9	Ground.

### 3.10. Connecting to Data Gateway Interface

The Atmel-ICE supports a limited Data Gateway Interface (DGI) when debugging and programming is not in use. Functionality is identical to that found on Atmel Xplained Pro kits powered by the Atmel EDBG device.

The Data Gateway Interface is an interface for streaming data from the target device to a computer. This is meant as an aid in application debugging as well as for demonstration of features in the application running on the target device.

DGI consists of multiple channels for data streaming. The Atmel-ICE supports the following modes:

- USART
- SPI

**Table 3-9. Atmel-ICE DGI USART Pinout**

AVR port	SAM port	DGI USART pin	Description
3	6	TX	Transmit pin from Atmel-ICE to the target device
4	1	VTG	Target voltage (reference voltage)
8	7	RX	Receive pin from the target device to Atmel-ICE
9	8	CLK	USART clock
2, 10	3, 5, 9	GND	Ground

**Table 3-10. Atmel-ICE DGI SPI Pinout**

AVR port	SAM port	DGI SPI pin	Description
1	4	SCK	SPI clock
3	6	MISO	Master In Slave Out
4	1	VTG	Target voltage (reference voltage)
5	2	nCS	Chip select active low
9	8	MOSI	Master Out Slave In
2, 10	3, 5, 9	GND	Ground



**Important:** SPI and USART interfaces can not be used simultaneously.



**Important:** DGI and programming or debugging cannot be used simultaneously.

## 4. On-chip Debugging

### 4.1. Introduction

#### On-chip Debugging

An on-chip debug module is a system allowing a developer to monitor and control execution on a device from an external development platform, usually through a device known as a *debugger* or *debug adapter*.

With an OCD system the application can be executed whilst maintaining exact electrical and timing characteristics in the target system, while being able to stop execution conditionally or manually and inspect program flow and memory.

#### Run Mode

When in Run mode, the execution of code is completely independent of the Atmel-ICE. The Atmel-ICE will continuously monitor the target device to see if a break condition has occurred. When this happens the OCD system will interrogate the device through its debug interface, allowing the user to view the internal state of the device.

#### Stopped Mode

When a breakpoint is reached, the program execution is halted, but some I/O may continue to run as if no breakpoint had occurred. For example, assume that a USART transmit has just been initiated when a breakpoint is reached. In this case the USART continues to run at full speed completing the transmission, even though the core is in stopped mode.

#### Hardware Breakpoints

The target OCD module contains a number of program counter comparators implemented in the hardware. When the program counter matches the value stored in one of the comparator registers, the OCD enters stopped mode. Since hardware breakpoints require dedicated hardware on the OCD module, the number of breakpoints available depends upon the size of the OCD module implemented on the target. Usually one such hardware comparator is 'reserved' by the debugger for internal use.

#### Software Breakpoints

A software breakpoint is a BREAK instruction placed in program memory on the target device. When this instruction is loaded, program execution will break and the OCD enters stopped mode. To continue execution a "start" command has to be given from the OCD. Not all Atmel devices have OCD modules supporting the BREAK instruction.

### 4.2. SAM Devices with JTAG/SWD

All SAM devices feature the SWD interface for programming and debugging. In addition, some SAM devices feature a JTAG interface with identical functionality. Check the device datasheet for supported interfaces of that device.

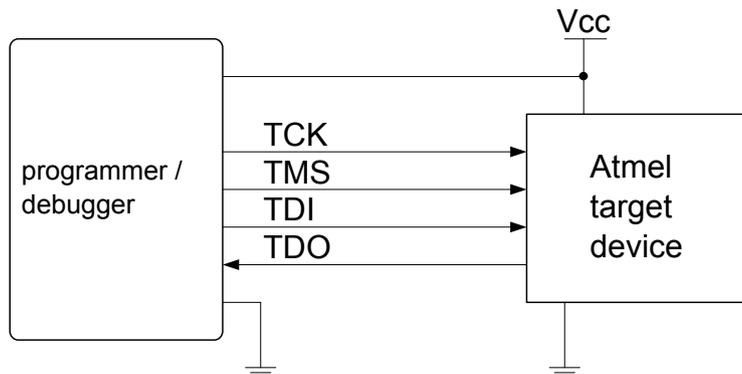
#### 4.2.1. ARM CoreSight Components

Atmel ARM Cortex-M based microcontrollers implement CoreSight compliant OCD components. The features of these components can vary from device to device. For further information consult the device's datasheet as well as CoreSight documentation provided by ARM.

## 4.2.2. JTAG Physical Interface

The JTAG interface consists of a 4-wire Test Access Port (TAP) controller that is compliant with the IEEE® 1149.1 standard. The IEEE standard was developed to provide an industry-standard way to efficiently test circuit board connectivity (Boundary Scan). Atmel AVR and SAM devices have extended this functionality to include full Programming and On-chip Debugging support.

Figure 4-1. JTAG Interface Basics



### 4.2.2.1. SAM JTAG Pinout (Cortex-M debug connector)

When designing an application PCB which includes an Atmel SAM with the JTAG interface, it is recommended to use the pinout as shown in the figure below. Both 100-mil and 50-mil variants of this pinout are supported, depending on the cabling and adapters included with the particular kit.

Figure 4-2. SAM JTAG Header Pinout

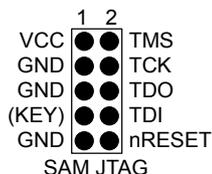


Table 4-1. SAM JTAG Pin Description

Name	Pin	Description
TCK	4	Test Clock (clock signal from the Atmel-ICE into the target device).
TMS	2	Test Mode Select (control signal from the Atmel-ICE into the target device).
TDI	8	Test Data In (data transmitted from the Atmel-ICE into the target device).
TDO	6	Test Data Out (data transmitted from the target device into the Atmel-ICE).
nRESET	10	Reset (optional). Used to reset the target device. Connecting this pin is recommended since it allows the Atmel-ICE to hold the target device in a reset state, which can be essential to debugging in certain scenarios.
VTG	1	Target voltage reference. The Atmel-ICE samples the target voltage on this pin in order to power the level converters correctly. The Atmel-ICE draws less than 1mA from this pin in this mode.
GND	3, 5, 9	Ground. All must be connected to ensure that the Atmel-ICE and the target device share the same ground reference.
KEY	7	Connected internally to the TRST pin on the AVR connector. Recommended as not connected.