# Section 1

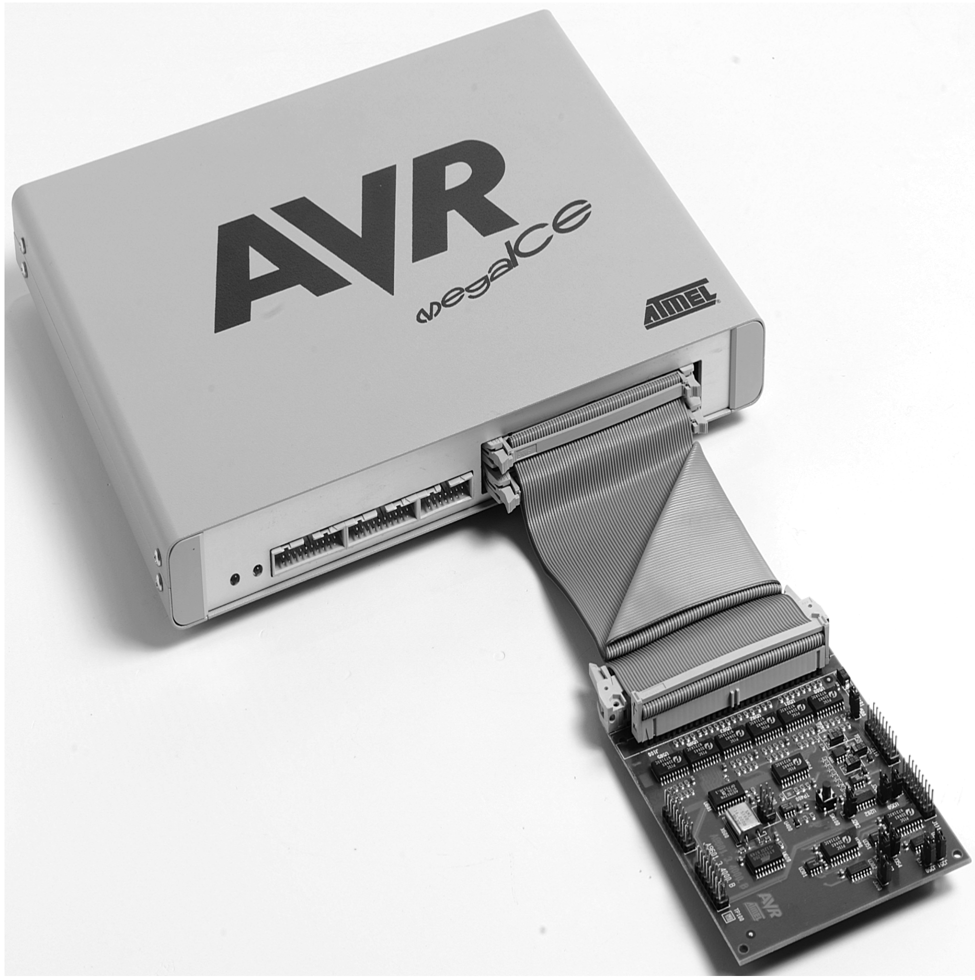
# Introduction

The ATmegaICE is a real time In-Circuit Emulator (ICE) for all ATmega devices. It can be upgraded to support future ATmega parts. It is controlled by AVR® Studio, which is a professional front-end for both high-level and assembly level debugging. AVR Studio is described in Section 1 in the Development Tools User Guide and must be studied carefully in order to take full advantage of this product.

The ATmega ICE system consists of the following elements:

- AVR ICE unit
- POD card ATmegaPOD
- Pod cable
- ATmega103/ATmega603 probe with TQFP64 adapter including cables
- RS-232 cable
- Power supply unit
- Documentation

***Figure 1-1.*** ATmegaICE with Pod Connected



The AVR ICE unit is the main part of the AVR ATmegaICE system. AVR Studio, which runs on a host PC, controls the ICE unit. The ICE unit must be connected to an available RS-232 port on the host PC with the supplied RS-232 cable. AVR Studio automatically detects if there is an emulator present on one of the PC's serial ports. Note that if no ICE is found, AVR Studio will show "Simulator" in the lower right corner of the main window. If the ICE is found, "AVR Emulator" will be indicated and your connections are correct. AVR Studio will issue a warning if a program previously run in emulator mode is started in simulator mode.

***Figure 1-2.*** ATmegaICE Back Panel


***Figure 1-3.*** ATmegaICE Front Panel

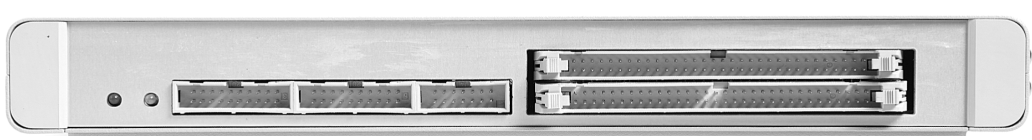
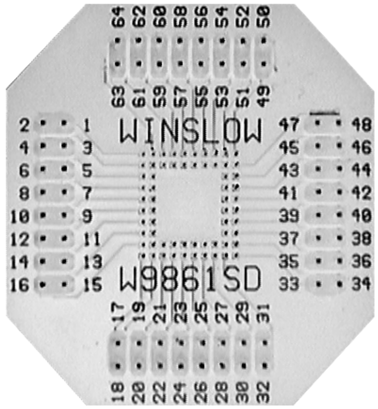


***Figure 1-4.*** ATmega103/ATmega603 Probe



# Section 2

# Preparing the ATmegaICE System for Use

Complete the following procedure in order to start using the ATmegaICE.

Before connecting the probe cable to the user application:

1. Connect the pod card to the AVR ICE unit with the pod cables. Do not disassemble the pod cables.
2. Connect the probe cables to the pod. See Section 4 for details.
3. Connect the RS-232 cable between the AVR ICE unit and an available serial port on the host PC.
4. Connect a power supply (9 - 15 VDC, negative center) to the ATmegaICE unit. The power supply delivered with your ATmegaICE is correctly dimensioned.
5. Turn the power on and check that the red LED marked “POWER” is lit.
6. After a short time (<10 s) the green LED marked “READY” will be lit and the ATmegaICE system is ready.
7. Turn the power off.
8. Solder the TQFP64 adapter PCB part of the probe connector to the target PCB.
9. Connect the probe into the application, and pay attention to connect it correctly. If it is not connected correctly, your ATmegaICE system may be damaged.
10. When the probe is correctly connected to the application, check the jumper marked “J500” (Ext.Power) on the ATmegaPOD (see Figure 3-1). If the application is to be powered from the ATmegaICE unit, the jumper should be mounted. If the application is to use its own power supply, the jumper must be removed.
11. Make sure that the voltage regulation, external clock and Timer oscillator are set according to the target system requirements (see Section 3).
12. When all jumpers are set, turn the power on. When the green LED is lit, the AVR ATmegaICE system is ready for use. Start AVR Studio, and begin your AVR emulation session.

It is possible to use the ATmegaICE without the ATmega probe and the TQFP64 adapter if four 8 x 2 pin headers are added to the target application PCB.

Tables 4-1, 4-2, 4-3 and 4-4 give details on how to use these header connectors. Connect the pin headers on the target application to the In-System connectors on ATmega-POD by using the probe cable included in your kit.

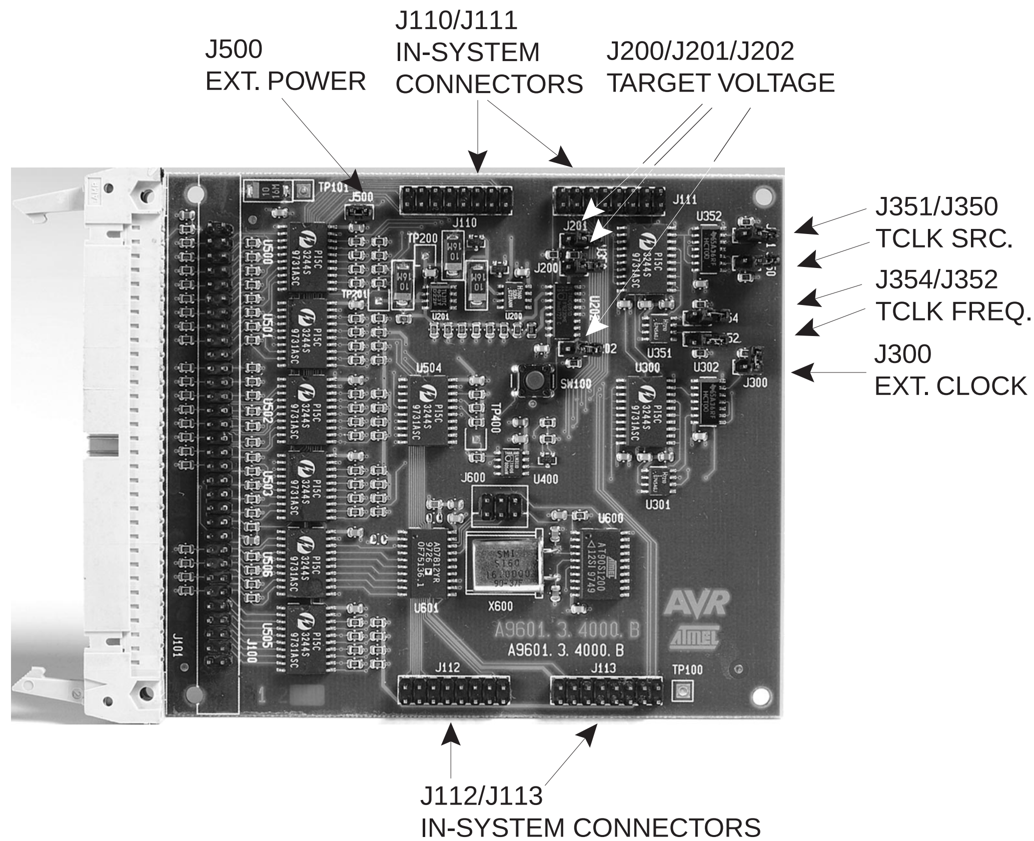
Section 3

# ATmegaPOD

ATmegaPOD is the pod card used for the microcontrollers ATmega103/603.

The pod contains several jumpers which must be set to achieve the desired operation. The jumpers involved in the configuration are indicated in Figure 3-1.

***Figure 3-1.*** Pod Jumper Placements



## 3.1 Voltage Regulation for Port Pins

If the target system uses its own power supply, the output voltage must be set to the voltage in the target system. The jumpers J200-J202 are used for this purpose. The interpretation of the different settings of these jumpers is shown in Figure 3-2.

***Figure 3-2.*** Jumper Setting

| Target VCC | J202 | J201 | J200 |
|---|---|---|---|
| 2.7 - 2.9 V | ▮ | ▮ | ▮ |
| 3.0 - 3.3 V | ▮ | ▮ | : |
| 3.4 - 3.7 V | ▮ | : | ▮ |
| 3.8 - 4.1 V | ▮ | : | : |
| 4.2 - 4.5 V | : | ▮ | ▮ |
| 4.6 - 4.8 V | : | ▮ | : |
| 4.9 - 5.1 V | : | : | ▮ |
| 5.2 - 5.5 V | : | : | : |

## 3.2 Clock Setting

The external clock system can be set to use either an external crystal or an external clock signal.

To use an external crystal, select “External Oscillator” in the “Emulator Options” menu in AVR Studio, and select an appropriate frequency range for the mounted crystal. Note that the selected frequency range may not correspond to the crystal frequency due to load capacitors and/or stray capacitors in the system. J300 must be open when using an external crystal.

If an external clock signal is used (on pin XTAL1), jumper J300 must be mounted. This setting also requires the user to select External Oscillator in the Emulator Options menu in AVR Studio.

ATmegaICE also has a very accurate programmable internal clock which can be used for emulation. Select this clock by choosing Internal Oscillator in the Emulator Options menu in AVR Studio. Setting of J300 on the ATmegaPOD and crystals and clock signals in the user application are insignificant when the internal oscillator is selected.

## 3.3 Timer Oscillator Setting

In order to enable the Timer oscillator, jumper J351 must be mounted. When J351 is mounted, the setting of J350 determines whether to use an external clock signal or the local oscillator. If J350 is mounted, the external clock signal is used (on pin TOSC1).

If J350 is not mounted, an external crystal is expected to be mounted between the TOSC1 and TOSC2 pins. When an external crystal is used (J350 not mounted), jumpers J352 and J354 should be used to indicate the frequency range of the oscillator. The interpretation of the different settings for these jumpers is given in Table 3-1.

***Table 3-1.*** OSC frequency range jumper settings

| J354 | J352 | Frequency Range |
|---|---|---|
| - | - | 10 kHz - 100 kHz |
| - | Mounted | 100 kHz - 1 MHz |
| Mounted | - | 1 MHz - 5 MHz |
| Mounted | Mounted | 5 MHz - 10 MHz |

As noted above, the selected frequency range may not correspond to the crystal frequency due to load capacitors and/or stray capacitors.

Due to long probe leads, it may be difficult to emulate the 32 kHz oscillator in many designs. A way to work around this is to lift pin 2 and 3 on U351 from the PCB and solder a 32 kHz crystal directly to the pins, without neither the pins nor the crystal touching the PCB.

## 3.4 Power to Target PCB

The ATmegaICE can power the target PCB with 5.0V. To enable this option, J500 should be mounted. Note that the absolute maximum supply current is 1.0A, so the target should use less than 5W of power to use this option.

If the target PCB uses voltages other than 5.0V, J500 must be removed.

# Section 4

# Connecting to the System

The device's pins are placed on the four in-system connectors, J110, J111, J112 and J113. The pinout of these connectors is shown in Tables 4-1, 4-2, 4-3 and 4-4. The column "#" shows the pin number on each header. The column "&" shows the corresponding pin number on an ATmega103/ATmega603 device. The numbering of the pins on each header is shown in Figure 4-1.

***Table 4-1.*** Pinout for Header J110

| J110 | | | | | | |
|---|---|---|---|---|---|---|
| # | | & | | # | | & |
| 1 | PE0 (PDI/RXD) | 2 | | 2 | $\overline{\text{PEN}}$ | 1 |
| 3 | PE2 (AC+) | 4 | | 4 | PE1 (PDO/TXD) | 3 |
| 5 | PE4 (INT4) | 6 | | 6 | PE3 (AC-) | 5 |
| 7 | PE6 (INT6) | 8 | | 8 | PE5 (INT5) | 7 |
| 9 | PB0 ($\overline{\text{SS}}$) | 10 | | 10 | PE7 (INT7) | 9 |
| 11 | PB2 (MOSI) | 12 | | 12 | PB1 (SCK) | 11 |
| 13 | PB4 (OC0/PWM0) | 14 | | 14 | PB3 (MISO) | 13 |
| 15 | PB6 (OC1B/PWM1B) | 16 | | 16 | PB5 (OC1A/PWM1A) | 15 |

***Table 4-2.*** Pinout for Header J111

| J111 | | | | | | |
|---|---|---|---|---|---|---|
| # | | & | | # | | & |
| 1 | TOSC2 | 18 | | 2 | PB7 (OC2/PWM2) | 17 |
| 3 | $\overline{\text{RESET}}$ | 20 | | 4 | TOSC1 | 19 |
| 5 | GND | 22 | | 6 | VCC | 21 |
| 7 | XTAL1 | 24 | | 8 | XTAL2 | 23 |
| 9 | $\overline{\text{INT1}}$ | 26 | | 10 | $\overline{\text{INT0}}$ | 25 |
| 11 | $\overline{\text{INT3}}$ | 28 | | 12 | $\overline{\text{INT2}}$ | 27 |
| 13 | PD5 | 30 | | 14 | PD4 (IC1) | 29 |
| 15 | PD7 (T2) | 32 | | 16 | PD6 (T1) | 31 |

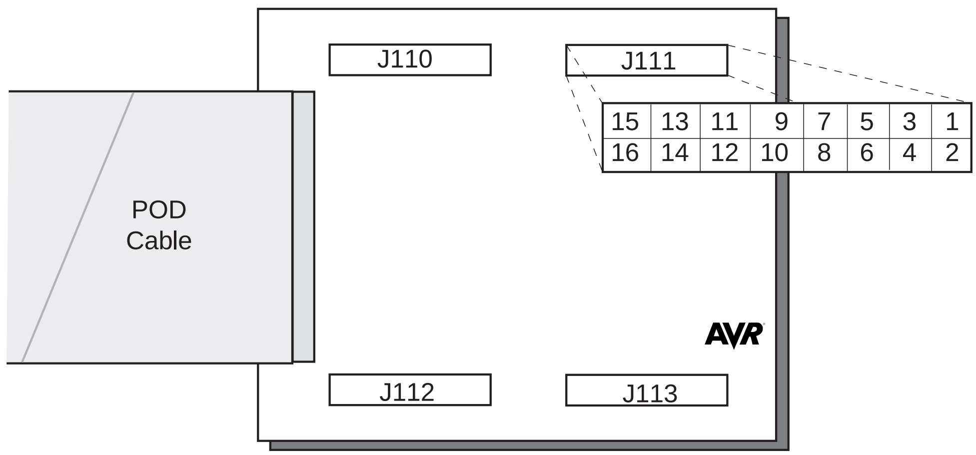
***Table 4-3.*** Pinout for Header J112

| J112 | | | | | | |
|---|---|---|---|---|---|---|
| # | | & | | # | | & |
| 1 | $\overline{\text{RD}}$ | 34 | | 2 | $\overline{\text{WR}}$ | 33 |
| 3 | PC1 (A9) | 36 | | 4 | PC0 (A8) | 35 |
| 5 | PC3 (A11) | 38 | | 6 | PC2 (A10) | 37 |
| 7 | PC5 (A13) | 40 | | 8 | PC4 (A12) | 39 |
| 9 | PC7 (A15) | 42 | | 10 | PC6 (A14) | 41 |
| 11 | PA7 (AD7) | 44 | | 12 | ALE | 43 |
| 13 | PA5 (AD5) | 46 | | 14 | PA6 (AD6) | 45 |
| 15 | PA3 (AD3) | 48 | | 16 | PA4 (AD4) | 47 |

***Table 4-4.*** Pinout for Header J113

| **J113** | | | | | | |
|---|---|---|---|---|---|---|
| # | | & | | # | | & |
| 1 | PA1 (AD1) | 50 | | 2 | PA2 (AD2) | 49 |
| 3 | VCC | 52 | | 4 | PA0 (AD0) | 51 |
| 5 | PF7 (ADC7) | 54 | | 6 | GND | 53 |
| 7 | PF5 (ADC5) | 56 | | 8 | PF6 (ADC6) | 55 |
| 9 | PF3 (ADC3) | 58 | | 10 | PF4 (ADC4) | 57 |
| 11 | PF1 (ADC1) | 60 | | 12 | PF2 (ADC2) | 59 |
| 13 | AREF | 62 | | 14 | PF0 (ADC0) | 61 |
| 15 | AVCC | 64 | | 16 | AGND | 63 |

***Figure 4-1.*** Pin Numbering



# Section 5

# Trace Buffer

The AVR ATmegaICE has a 32K x 96-bit trace buffer which stores information about program execution for every clock cycle. When the emulator is stopped, this trace buffer can be examined to extract information about the history of the emulated program. The details on which data are stored and how to retrieve them are described in the AVR Studio User Guide. When the trace buffer is full, it will wrap around and start overwriting the oldest entries.

The trace buffer can be turned on or off at any program line. This makes it possible to skip tracing delay loops and other subroutines which would otherwise fill the trace memory with unnecessary data. The trace buffer is inactive by default. To trace an entire program, a "Trace on" marker should be placed on the first line of the program.

# Section 6

# External Triggers

The AVR ATmegaICE has five external trigger inputs and five trigger outputs, all located on the AUX connector next to the pod connector.

- The trigger inputs can act as break signals to the emulator, and/or they can be logged in the trace buffer. If an input is enabled for breaking the emulator, a rising edge on that input is required to break the emulator.
- The trigger outputs may be set as trigger points on any instruction in the code window in AVR Studio. If enabled on an instruction, the output(s) will remain high for one AVR clock cycle when the marked instruction is executed. This can be used to trigger a logic analyzer or an oscilloscope.

The details on how to enable and setup triggers and mask registers are presented in the AVR Studio User Guide.

There are three global mask registers that are used to control the behavior of the triggers:

- The *Trigger Output Global Mask Register* controls which of the output pins are allowed to be controlled by the trigger settings in the code. An output pin which is disabled will remain low even if a trigger point for that particular pin is set in the code.
- The *Trigger Input Global Mask Register* controls which of the input pins are allowed to break the emulator. If more than one line is enabled, the emulator will break on either one, but will not store any information about which input caused the event. Note that unconnected inputs are pulled high by internal pull-up resistors. Unused lines must either be pulled low or disabled in the Trigger Input Global Mask Register. Otherwise, the emulator may break on the first line (on start up, all lines are disabled, so this only applies if input triggers are enabled by the user).
- The *External Trace Mask Register* controls which of the input pins will be stored in the trace memory. Input pins that are not enabled in this register will be stored as zero in the trace memory. To be traced, input signals must be valid and stable at the rising edge of the AVR clock and for 50ns thereafter. It is also necessary that the trace buffer is enabled and turned on.

The trigger input and the external trace are two independent functions acting on the same input pins. Note that the trigger logic is asynchronous and edge driven, whereas the trace logic is clocked on the AVR clock. The emulator may therefore break on a glitch signal which is too narrow to be traced.

Section 7

# Troubleshooting Procedures

If your AVR ATmegaICE system malfunctions, try the following actions:

- Push the Stop button in AVR Studio, then the reset button, and then try again.
- Push the “ICE RESET” button on the back panel and restart AVR Studio.
- Turn the power OFF, and then ON again and restart AVR Studio.
- Turn off power and disconnect your target application. Then try again. If everything is OK now, this indicates that something in your target application is disturbing normal operation.
- Turn off power and disconnect your pod card. Then try again. If everything is OK now, this indicates that your settings on the pod card are wrong and disturbs normal operation.

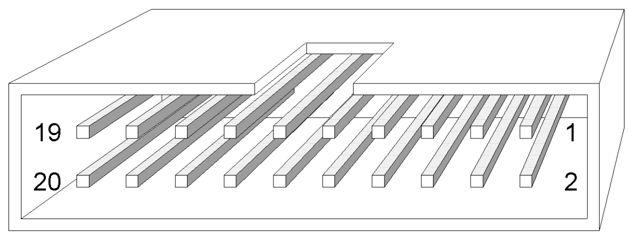
If your ATmegaICE system still operates abnormally, please contact your nearest Atmel sales office or AVR support at avr@atmel.com.

# Section 8

# AVR ATmegaICE Connector Description

## 8.1 Logic Analyzer 1 Connector Description

***Figure 8-1.*** Logic Analyzer 1 Connector



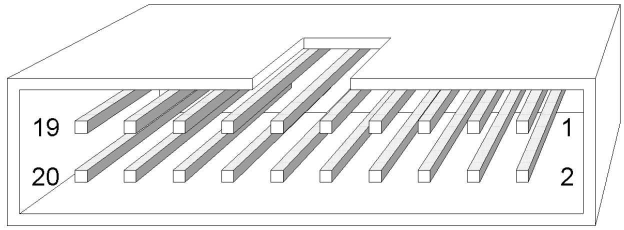

The connector marked “Logic Analyzer 1” on the back panel of the AVR ICE unit has the following pin-out with signals from the instruction address bus.

***Table 8-1.*** Logic Analyzer Connector 1

| SIGNAL | Logic Analyzer 1 | | SIGNAL |
|---|---|---|---|
| AVRCLK | Pin 1 | Pin 2 | Low |
| Low | Pin 3 | Pin 4 | A15 |
| A14 | Pin 5 | Pin 6 | A13 |
| A12 | Pin 7 | Pin 8 | A11 |
| A10 | Pin 9 | Pin 10 | A9 |
| A8 | Pin 11 | Pin 12 | A7 |
| A6 | Pin 13 | Pin 14 | A5 |
| A4 | Pin 15 | Pin 16 | A3 |
| A2 | Pin 17 | Pin 18 | A1 |
| A0 | Pin 19 | Pin 20 | GND |

## 8.2 Logic Analyzer 2 Connector Description

***Figure 8-2.*** Logic Analyzer 2 Connector



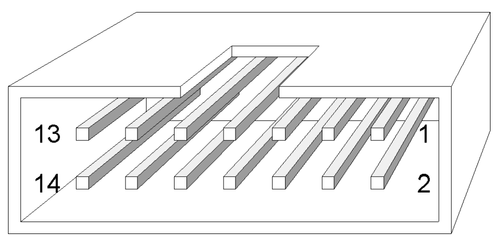

The connector marked "Logic Analyzer 2" on the back panel of the AVR ICE unit has the following pin-out with signals from the instruction data bus.

***Table 8-2.*** Logic Analyzer Connector 2

| SIGNAL | Logic Analyzer 2 | | SIGNAL |
|---|---|---|---|
| AVRCLK | Pin 1 | Pin 2 | Low |
| Low | Pin 3 | Pin 4 | D15 |
| D14 | Pin 5 | Pin 6 | D13 |
| D12 | Pin 7 | Pin 8 | D11 |
| D10 | Pin 9 | Pin 10 | D9 |
| D8 | Pin 11 | Pin 12 | D7 |
| D6 | Pin 13 | Pin 14 | D5 |
| D4 | Pin 15 | Pin 16 | D3 |
| D2 | Pin 17 | Pin 18 | D1 |
| D0 | Pin 19 | Pin 20 | GND |

## 8.3 AUX Connector Description

***Figure 8-3.*** AUX Connector




The connector marked "AUX" on the back panel of the AVR ICE unit is used for external triggers and has the following pinout:

***Table 8-3.*** AUX Connector

| SIGNAL | AUX | | SIGNAL |
|---|---|---|---|
| GND | Pin 1 | Pin 2 | GND |
| Input 0 | Pin 3 | Pin 4 | Output 0 |
| Input 1 | Pin 5 | Pin 6 | Output 1 |
| Input 2 | Pin 7 | Pin 8 | Output 2 |
| Input 3 | Pin 9 | Pin 10 | Output 3 |
| Input 4 | Pin 11 | Pin 12 | Output 4 |
| GND | Pin 13 | Pin 14 | GND |

# Section 9

# Technical Specifications

**System Unit**

| | |
|---|---|
| Physical Dimensions | (H x W x D) 32.4 x 277.1 x 218.6 mm / 1.3 x 10.8 x 8.5 in |
| Weight | 400g / 0.88 lbs |
| Power Voltage Requirements | 9 - 15 VDC |
| Power Consumption | < 20 W |
| ICE Power Consumption | < 15 W |
| Max. Application Power Consumption | 5 W |
| Ambient Temperature | 0 - +70°C (Operating) |
| | -55 - +95°C (Non-Operating) |
| Relative Humidity (Non-condensing) | 10 - 90% (Operating) |
| | 5 - 95% (Non-Operating) |
| Shock | 20 g, 11ms half sine |
| Vibration | 5 g |

**Connections**

**Power**

| | |
|---|---|
| Connector | 5.5 mm OD/ 2.1mm ID Center Negative |

**Host**

| | |
|---|---|
| Serial Connector (RS-232) | 9-pin D-SUB Female |
| Serial Communications Speed | 19200 bits/s |

**Pod**

| | |
|---|---|
| Connectors | two 2 x 32 Male Header |

**External Trigger Inputs / Outputs**

| | |
|---|---|
| Connector | 2 x 7 Male Header |

**Logic Analyzer Interface**

| | |
|---|---|
| Connectors | two 2 x 10 Male Headers |

**Clock specification**

**Internal Clock:**

| | |
|---|---|
| Minimum frequency | 400 kHz |
| Maximum frequency | 20.0 MHz |

**External Crystal:**

| | |
|---|---|
| Minimum frequency | 32.768 kHz |
| Maximum frequency | 10.0 MHz |

**Internal Watchdog RC oscillator:**

Running frequency . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1.0 MHz ± 30%

**Operation**

Minimum running speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .32.768 kHz

Maximum running speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6.0 MHz

Minimum Single-step speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .32.768 kHz

Maximum Single-step speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6.0 MHz

Minimum breakpoint speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .32.768 kHz

Maximum breakpoint speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6.0 MHz

**Memory specification**

Program Memory. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .128K bytes

Event Memory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .128K bytes

EEPROM Memory. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .64K bytes

SRAM Memory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .64K bytes

Register file . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .32 bytes

I/O area . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .64 bytes

Trace buffer memory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .32K x 12 bytes

**I/O pins:**

Output level . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .CMOS (VCC: 2.7 - 5.5 $V_{DC}$)

Maximum sink current . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24 mA

Maximum source current. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10 mA

Permanent Pull-Up . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .1.0 MΩ

# Section 10

# Appendix

## 10.1 megaICE Warranty

Atmel warrants that the megaICE in-circuit emulation system delivered hereunder shall conform to the applicable specifications and shall be free from defects in material and workmanship under normal use and service for a period of **1 year** from the applicable date of invoice. The megaICE pod and cables included with the emulation system are warranted for a period of **30 days** from the applicable date of invoice. Products which are “samples”, “design verification units”, and/or “prototypes” are sold “AS IS,” “WITH ALL FAULTS,” and with no warranty whatsoever.

If, during such warranty period, (i) Atmel is notified promptly in writing upon discovery of any defect in the goods, including a detailed description of such defect; (ii) such goods are returned to Atmel, DDP Atmel’s facility accompanied by Atmel’s Returned Material Authorization form; and (iii) Atmel’s examination of such goods discloses to Atmel’s satisfaction that such goods are defective and such defects are not caused by accident, abuse, misuse, neglect, alteration, improper installation, repair, improper testing, or use contrary to any instructions issued by Atmel, Atmel shall (at its sole option) either repair, replace, or credit Buyer the purchase price of such goods. No goods may be returned to Atmel without Atmel’s Returned Material Authorization form.

Prior to any return of goods by Buyer pursuant to this Section, Buyer shall afford Atmel the opportunity to inspect such goods at Buyer’s location, and any such goods so inspected shall not be returned to Atmel without its prior written consent.

Atmel shall return any goods repaired or replaced under this warranty to Buyer transportation prepaid, and reimburse Buyer for the transportation charges paid by Buyer for such goods. The performance of this warranty does not extend the warranty period for any goods beyond that period applicable to the goods originally delivered.

THE FOREGOING WARRANTY CONSTITUTES ATMEL'S EXCLUSIVE LIABILITY, AND THE EXCLUSIVE REMEDY OF BUYER, FOR ANY BREACH OF ANY WARRANTY OR OTHER NONCONFORMITY OF THE GOODS COVERED BY THIS AGREEMENT. THIS WARRANTY IS EXCLUSIVE, AND IN LIEU OF ALL OTHER WARRANTIES. ATMEL MAKES NO OTHER WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, including WITHOUT LIMITATION ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE SOLE AND EXCLUSIVE REMEDY FOR ANY BREACH OF THIS WARRANTY SHALL BE AS EXPRESSLY PROVIDED HEREIN.

## 10.2 Limitation on Liability

Notwithstanding anything to the contrary contained herein, Atmel shall not, under any circumstances, be liable to Buyer or any third parties for consequential, incidental, indirect, exemplary, special, or other damages. Atmel's total liability shall not exceed the total amount paid by Buyer to Atmel hereunder. Atmel shall not under any circumstances be liable for excess costs of reprocurement.