AVR® ICE 200

User Guide

Table of Contents

Section 1

Preface – Read this First

1.1 About this Manual

This user guide serves as a reference manual for the Atmel AVR® ICE200 In-Circuit Emulator. This user guide is an easy introduction on how to use the ICE200, and a detailed reference for advanced users.

The user should install the latest version of the AVR Studio® available on the Atmel web site.

1.2 Helpful Information

This manual contains helpful information to improve the reliability, performance, and longevity of the ICE200 and the target system.

NOTICE!

This is a Notice...

Please follow the instructions in a NOTICE carefully.

1.3 Tips

Some sections contain useful tips for using the ICE200. All the tips are emphasized as shown in the example below.

Tip!

This is a tip!

1.4 Checklists

When the detailed descriptions in the *Connecting to the Target Application* and in the *Configuration* sections have been used and you are beginning to feel comfortable with the use of the ICE200, you can use the checklists at the end of these sections for fast setup of a new project. The checklists are of great help for getting the debugging system online without problems. However, novice users should also check that the operating conditions of the target system are compliant to the requirements of ICE200. This is described in Section 4: “Using the ICE200”.

1.5 Related Documentation

The AVR Technical Library CD-ROM contains various documentation relating to the use of AVR microcontrollers and of the debugging tools including AVR Studio User Guide, AVR Assembler User Guide and complete microcontroller data sheets.

Section 2

Introduction

The ICE200 In-Circuit Emulator provides an easy way of debugging embedded systems that utilizes the Atmel AVR microcontroller. It emulates eleven different devices of the AVR and the tinyAVR™ families.

The philosophy of the ICE200 is to provide an easy-to-use debugging platform, with a minimum of differences between the emulator and the actual processor it is emulating. The AVR emulator chip used by the ICE200 is produced in the same process technology as the microcontroller it is emulating. This provides identical electrical characteristics. On-board debugging resources ensure non-intrusive software emulation. The ICE200 hardware also includes an automatic configuration system that makes the process of connecting the target to the emulator an easy task.

Figure 2-1. The ICE200 Components

When used with the AVR Studio debugging environment, the ICE200 gives the user full run time control, unlimited number of Break Points, symbolic debugging and full Memory and Register visibility. Multiple ICE200 emulators can be used by AVR Studio at the same time, only limited by the number of serial ports available, giving a high degree of flexibility.

2.1 Additional Components

The ICE200 Surface Mount Kit that can be purchased seperately for design using the SMD versions of the AVR family microcontrollers. See Section 7 for details.

2.2 ICE200 Features

- Devices Supported:

  ATtiny12, AT90S2313, AT90S2333/4433, AT90S4414/8515, AT90S4434/8535, ATtiny10/11 (using ATtiny12 adapter), AT90S1200 (using AT90S2313 adapter).
- Supports 8 MHz (+4.0V to +6.0V) AVR Emulator Chip (varies between devices being emulated).
- Wide Target Voltage Range (+2.7V to +5.5V).
- Emulator Chip Provides Excellent AC Characteristics.
- Non-intrusive.
- Target Voltage Sensing Ensures Secure Operation.
- Personality Adapter for Each of the Supported Processors.
- 32-bits Cycle Counter.
- I/O Continues to Operate in Halt State After a Break or Break Point.
- Single-stepping or Continuous Timer Operation while Single-stepping Code. Utilizes the AVR Studio Debugging Environment that adds: Full Run Time Control: run, break, trace into, step over, step out, run-to-cursor, reset, autostep and multistep.
- Unlimited Number of Break Points.
- Symbolic Debugging Support.
- Full Visibility of and Access to Register File, SP, PC, and Memories.
- Access to all I/O Registers – See Section 5: “Special Considerations”.
- Auto Log Points – Non-real Time Logging/Watches.

2.3 ICE200 Contents

The ICE200 contains the following items:

- ICE200 main board, pod and two flexible printed circuit cables.
- Personality adapters for:

  ATtiny12 (8-pin DIP), AT90S2313 (20-pin DIP), AT90S2333/4433 (28-pin DIP), AT90S4414/8515 (40-pin DIP), and AT90S4434/8535 (40-pin DIP)
- 9-pin RS-232C cable
- AVR Technical Library CD-ROM containing:

  AVR Data Sheets

  Application notes

  AVR Studio

  AVR Assembler
- ICE200 User Guide (this document)
- Power cable
- Diagnostic adapter for test purposes

2.4 System Requirements

2.4.1 Hardware Requirements

A personal computer with the following specifications is recommended:

- 64M Bytes RAM, or more
- 15M Bytes of free hard disk space
- CD-ROM or Internet access (for software and data sheets)
- SVGA monitor, or better
- 16650 Compatible Serial Port (COM port)

2.4.2 Software Requirements

The following operating systems are currently supported by AVR Studio:

- AVR Studio v2.00 or later installed. See the Atmel web site (www.atmel.com) for latest version.
- Microsoft® Windows® NT® 4.0
- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows 2000
- Microsoft Windows XP

Note: AVR Studio will be updated to comply with new versions of these operating systems. See AVR Studio User Guide for latest information.

2.4.3 Operating Conditions

- Operation Temperature: 0°C - 70°C.
- Operating Humidity: 10 - 90% RH (non-condensing).
- Supply Voltage: +9.0V DC.
- Supply Current: 400 mA.

NOTICE!

Violating the recommended operating conditions for the ICE200 might cause incorrect operation and damage the emulator.

2.4.4 Host Interface

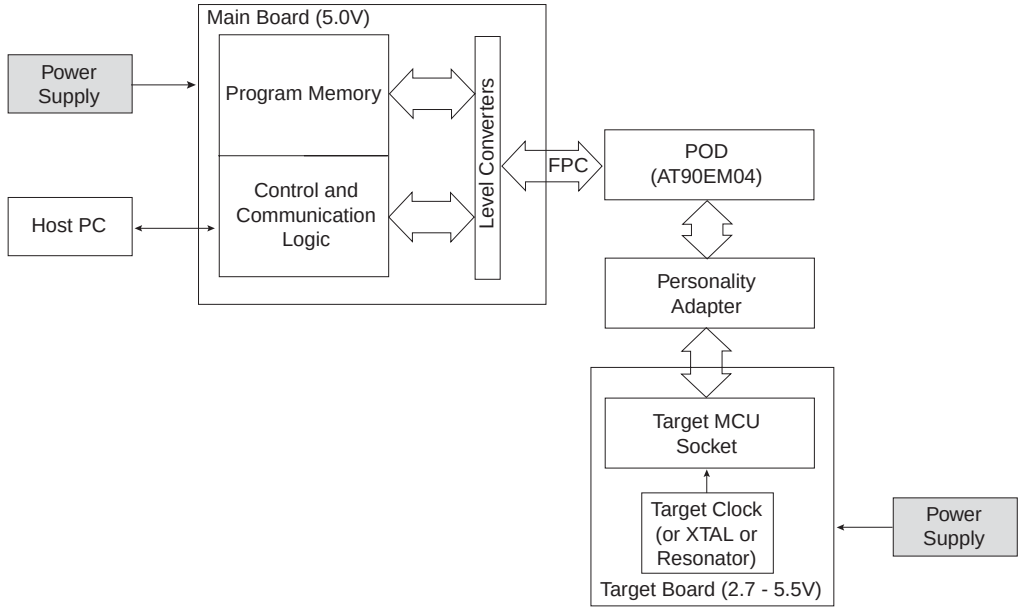
RS-232C @ 19200 bps, 1 start-, 8 data- and 1 stop-bit, no parity. 9-pin female connector.

Section 3

General Description

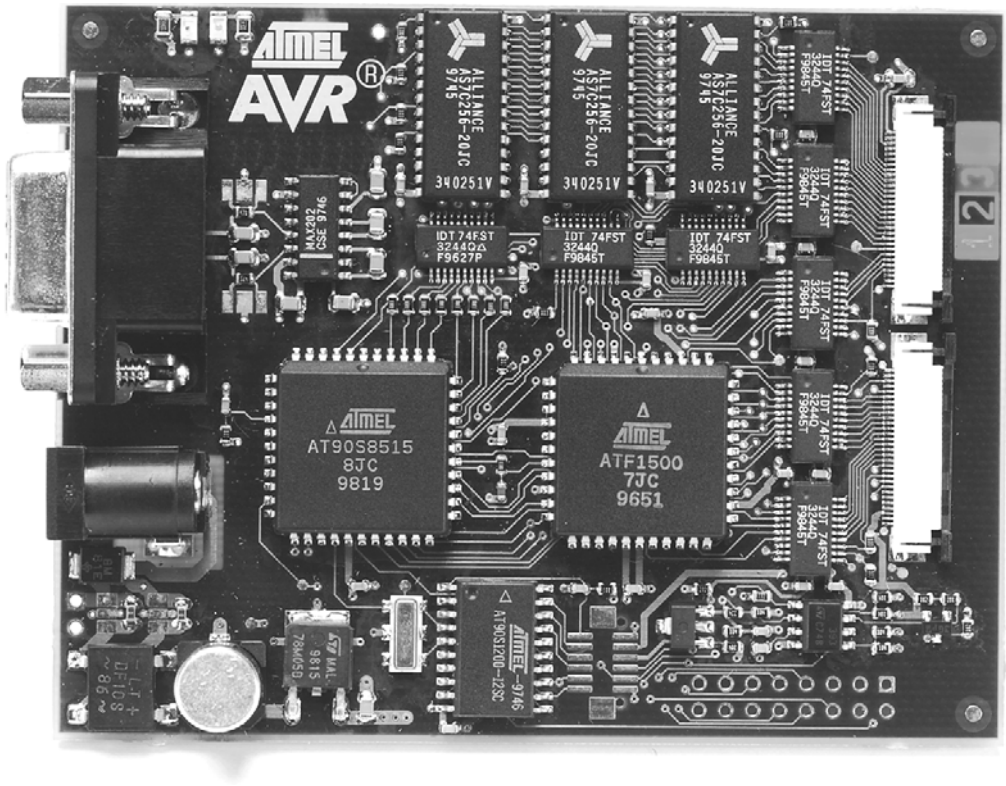
Figure 3-1 shows a simplified block diagram of the ICE200 connected to a target board (the application). Power supplies and a host PC are also shown.

Figure 3-1. ICE200 – Simplified Block Diagram



The main board (Figure 3-2) contains the Program memory (overlay memory) which holds the application code that is being emulated. The main board also contains logic for communicating with the host PC, and the Break Point logic. The level converters allow the target to operate at a different supply voltage from that of the emulator. The level converters also protect the emulator and the target from being damaged if only one of them is powered. Due to this feature, a strict Power-up sequence is not required.

Figure 3-2. ICE200 – Main Board



The FPC or Flexible Printed Cable (Figure 3-3) connects the main board to the ICE200 Pod. The actual appearance of the FPC may differ from the figure.

Figure 3-3. ICE200 – FPC

NOTICE!

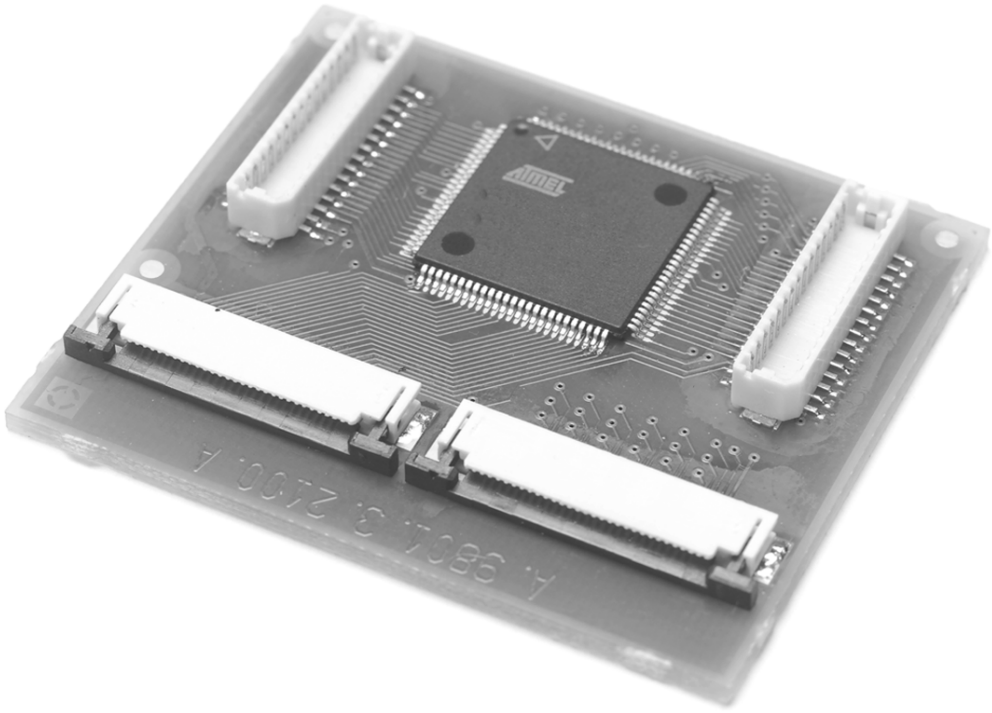
The Flexible Printed Cable must not be folded.

NOTICE!

Do not disassemble the Flexible Printed Cable from the pod or ICE200 main board.

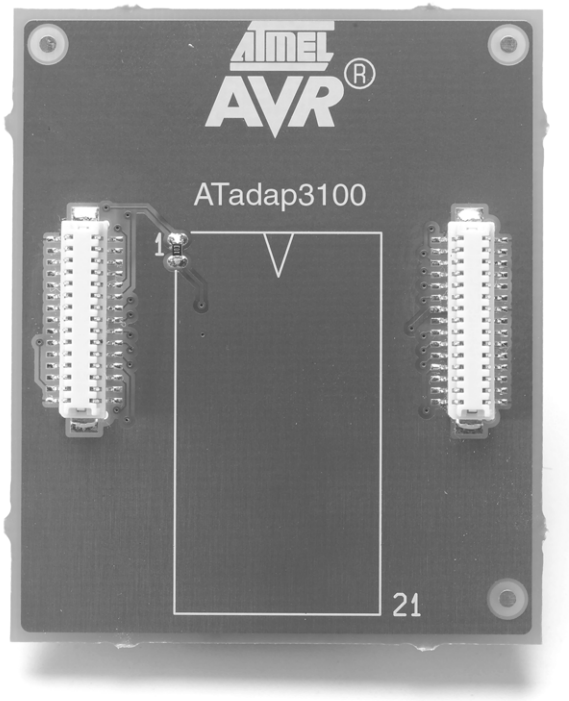
The pod (Figure 3-4) contains the AVR emulator chip. Note that the AVR emulator chip must be supplied with power and a clock source, i.e., a crystal, resonator, Oscillator or any other clock generator, from the target.

Figure 3-4. The ICE200 Pod



The personality adapters (Figure 3-5) map the pinout from ICE200 Pod to each microcontroller it supports. The adapters include an identification code that AVR Studio uses for automatic device type detection. The ICE200 kit contains five different personality adapters for dual-in-line package devices.

Figure 3-5. The ICE200 – Personality Adapter for the AT90S8535 – ATadap3100



Section 4

Using the ICE200

Before opening the ESD protection bag, take precaution to eliminate electrostatic discharge. Always use ESD protected tools and clothing when using the ICE200. Grounded wrist-band and static-dissipative work surface provides the most efficient ESD protection. The ICE200 should be handled with the same care as any CMOS component.

NOTICE!

ESD (Electrostatic Discharge) SENSITIVE DEVICE. Do not use the ICE200 outside an ESD protected environment.

A discharge may result in permanent damage or performance degradation.

4.1 Target Hardware Requirements

The target application hardware must include both power supply and a clock source. The ICE200 can not function unless these conditions are met. Note that the emulator also supports the internal RC Oscillator option to the ATtiny12 device.

Tip!

You can use an AVR development board (ATSTK500 or ATSTK200) for using the ICE200 as a standalone emulator platform.

Please follow the recommended operating conditions listed in the next two sections. These conditions also apply for the standard AVR microcontrollers.

4.2 Power and Signal Operating Conditions

Table 4-1. Recommended Operating Conditions, Power and Signals
(T_A = 0°C to 70°C, V_{CC} = +2.7V to +5.5V, GND = 0V)

Symbol	Min	Max
V_{CC}[1]	2.7V	5.5V
AVCC	V_{CC}	V_{CC} ± 0.3V
AGND	GND	GND
AREF	AGND	AVCC
V_{SI} (Signal Input Voltage)	-0.5V	V_{CC} + 0.5V
V_{RESET} (RESET pin input Voltage)	GND	V_{CC}[2]

Notes: 1. When V_{CC} < 2.4V, the AVR emulator chip is reset and the program memory disconnected.
2. The ICE200 does not support +12V RESET pin voltage that is used for Parallel Programming.

4.3 Clock Driver Requirements

AVR microcontrollers are fully static designs. The processor clock can be stopped externally. The AVR emulator chip needs a clock to communicate with the main board. Without a clock source, the host PC gets a serial communication time-out when reading status or variables from the emulator. Please refer to the data sheet for information about clock Oscillator options.

Table 4-2. Recommended Operating Conditions, Clock Drive
(+4.0V to +5.5V, T_A = 0°C to 70°C, V_{CC} = +4.0V to +5.5V, GND = 0V)

Symbol	Min (kHz)	Max (MHz)
f_{osc} (ATtiny12)	32.768	8
f_{osc} (AT90S2313)	32.768	10
f_{osc} (AT90S4433/2333)	32.768	10
f_{osc} (AT90S8515/4414)	32.768	8
f_{osc} (AT90S8535/4434)	32.768	8

Table 4-3. Recommended Operating Conditions, Clock Drive
(Low-voltage +2.7V to +4.0V, T_A = 0°C to 70°C, V_{CC} = +2.7V to +4.0V, GND = 0V)

Symbol	Min (kHz)	Max (MHz)
f_{osc} (ATtiny12/)	32.768	4
f_{osc} (AT90S2313)	32.768	4
f_{osc} (AT90S4433/2333)	32.768	4
f_{osc} (AT90S8515/4414)	32.768	4
f_{osc} (AT90S8535/4434)	32.768	4

NOTICE!

Using the ICE200 outside the recommended operating conditions will cause incorrect operation and can damage the emulator.

4.4 Personality Adapters

The ICE200 is supplied with five different personality adapters. Each adapter makes the pinout mapping for one or more AVR microcontrollers.

Tip!

Mounting a DIP socket on the personality adapter reduces the risk of breaking pins on the adapter, thus extending adapter lifetime.

Tip!

You can mount additional DIP sockets to the adapter to increase the space between the adapter and the target. However, the number of extra sockets should be kept at a minimum.

If you are utilizing Surface Mount Device (SMD) versions of the supported AVR microcontrollers, you need to obtain an SMD adapter that converts from DIP to the appropriate socket.

Figure 4-1. Personality Adapter for ATtiny12 – ATadap3400

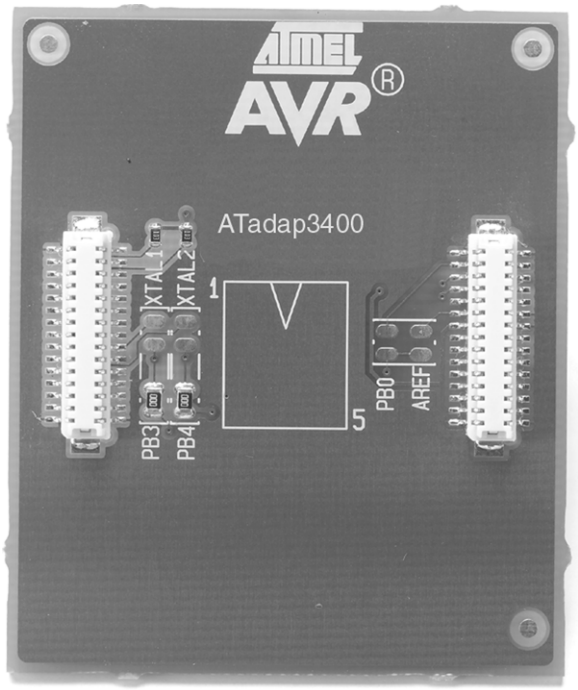


Figure 4-2. Personality Adapter for AT90S2313 – ATadap3300

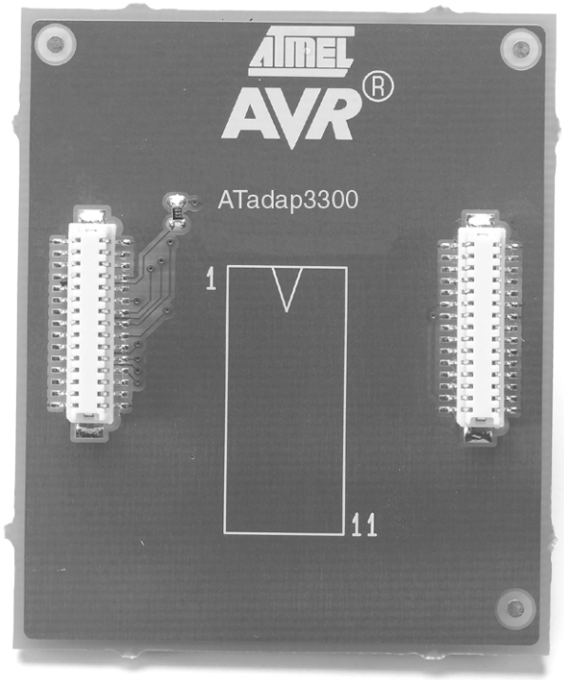


Figure 4-3. Personality Adapter for AT90S4433/2333 – ATadap3200

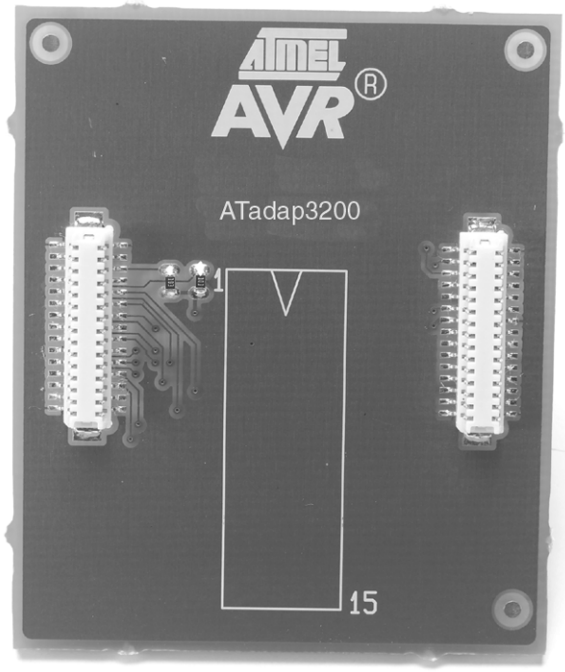


Figure 4-4. Personality Adapter for AT90S8515/4414 – ATadap3000

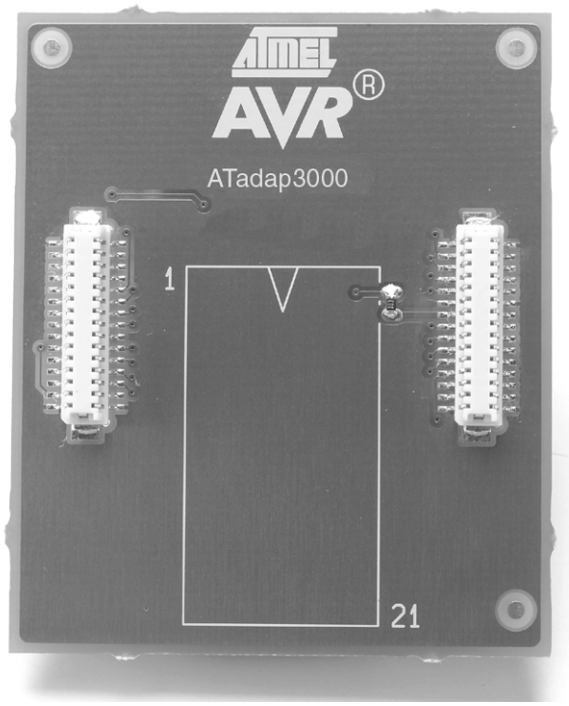
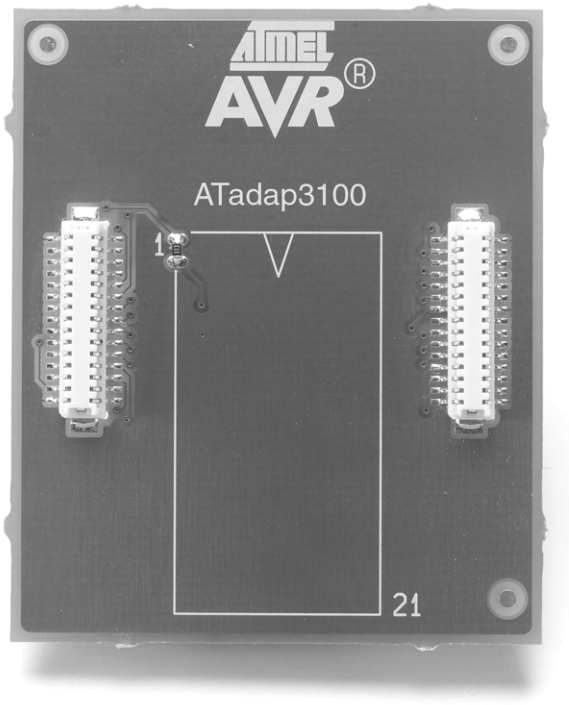


Figure 4-5. Personality Adapter for AT90S8535/4434 – ATadap3100



NOTICE!

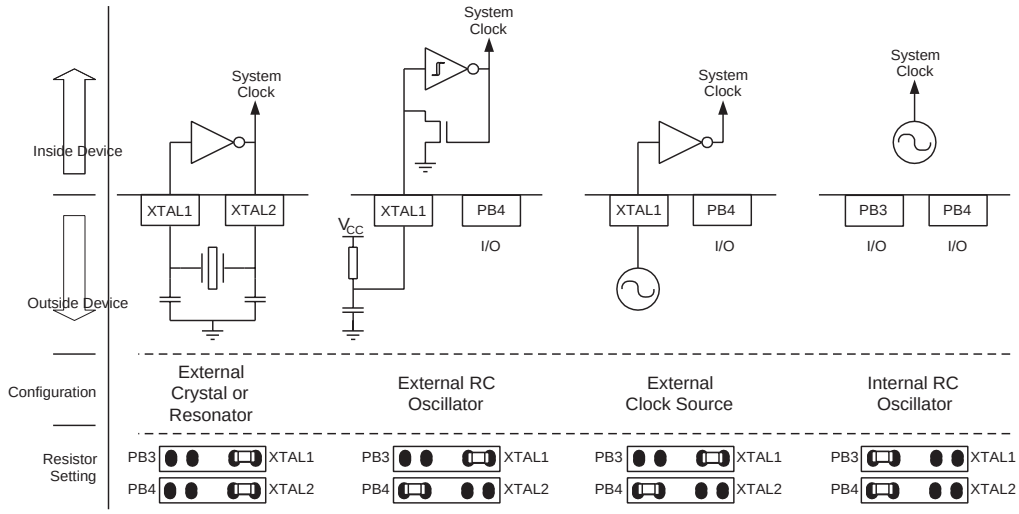
Do not change personality adapter without turning power-off on both the emulator and the target.

4.5 Special ATtiny12 Personality Adapter Settings

The AVR ATtiny12 microcontroller includes some special Oscillator pin features that could not be implemented in the AVR emulator chip due to its multiple device support. However, the options are supported on the personality adapter by changing the position of two 0 Ω resistors that configure the port and the oscillator pin mapping. The setting of the resistors is shown in Figure 4-6.

The resistor setting depends on the desired clock configuration. Select the resistor setting based on the figure below. Default factory resistor setting is internal RC Oscillator.

Figure 4-6. Settings of Resistors on ATadap3400



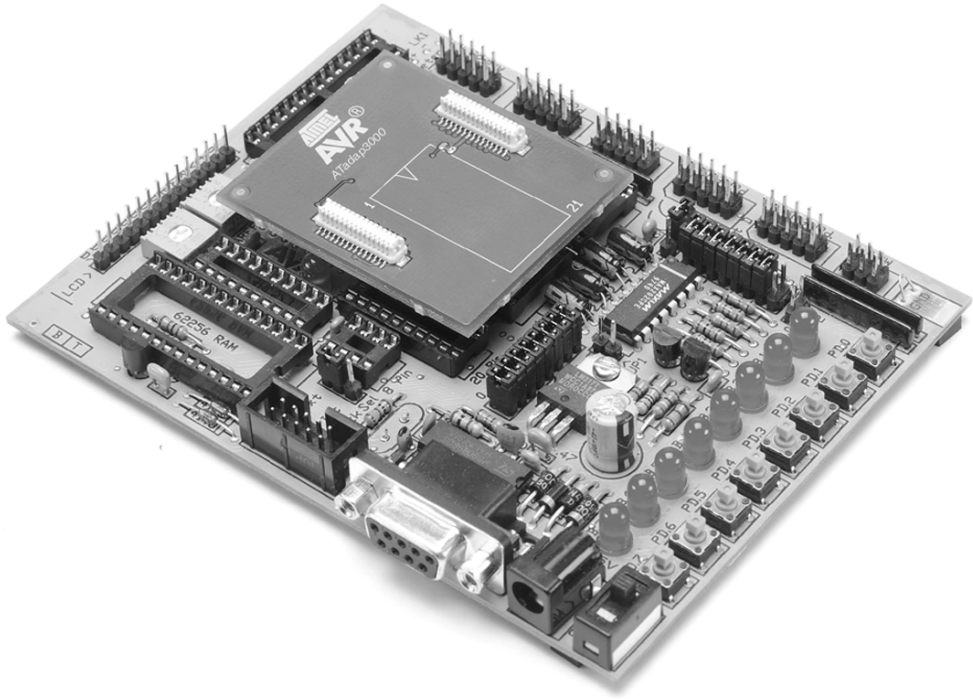
The personality adapter board has four additional resistors. These are used as identification codes for the automatic configuration and for production test purposes. Do not remove these resistors.

4.6 Connecting to the Target Application

The procedure in this section must be followed when setting-up the emulator. The end of this section contains a useful checklist for ensuring fast and safe installation of the system.

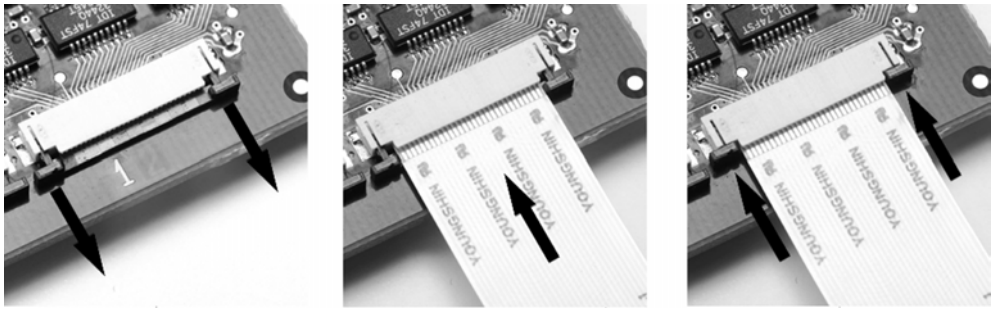
1. Before connecting the ICE200 to the target application, make sure that the ICE200 and the target application are not powered. This also applies when the ICE200 is removed from the target. When connecting or disconnecting the ICE200 from the host PC, make sure that neither the ICE200 nor the target application is powered.
2. Start inserting a personality adapter (see Figure 4-7). Make sure that pin 1 on the personality adapter corresponds with pin 1 on the target socket.

Figure 4-7. Inserting a Personality Adapter Into the Target Hardware



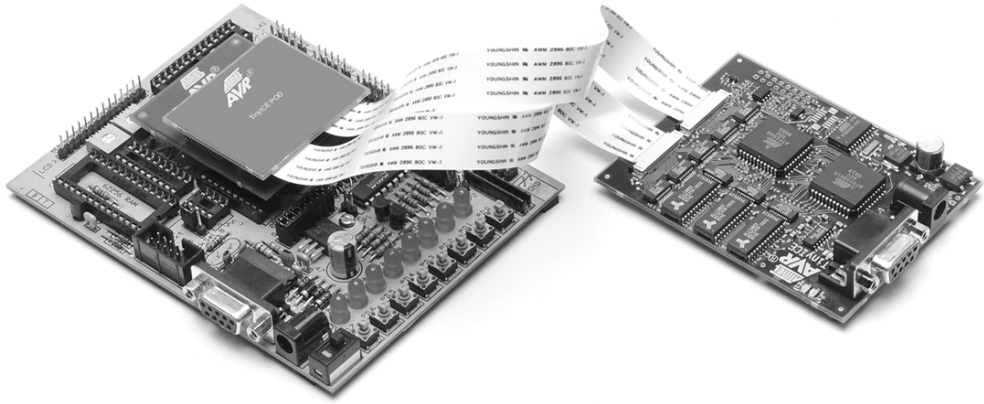
3. If the Flexible Printed circuit Cable (FPC) is not already connected to the pod and the main board, connect it as shown in Figure 4-8. Note that the FPC lead prints must face up.

Figure 4-8. Mounting the FPC to the pod and the Main Board



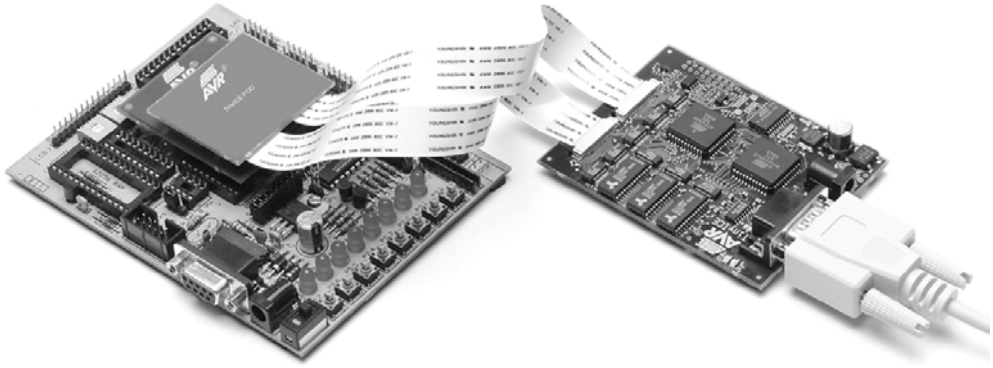
4. Mount the pod onto the personality adapter as shown in Figure 4-9. Do not use force since the pod only fits one way into the personality adapter.

Figure 4-9. Mounting the Pod Onto the Personality Adapter



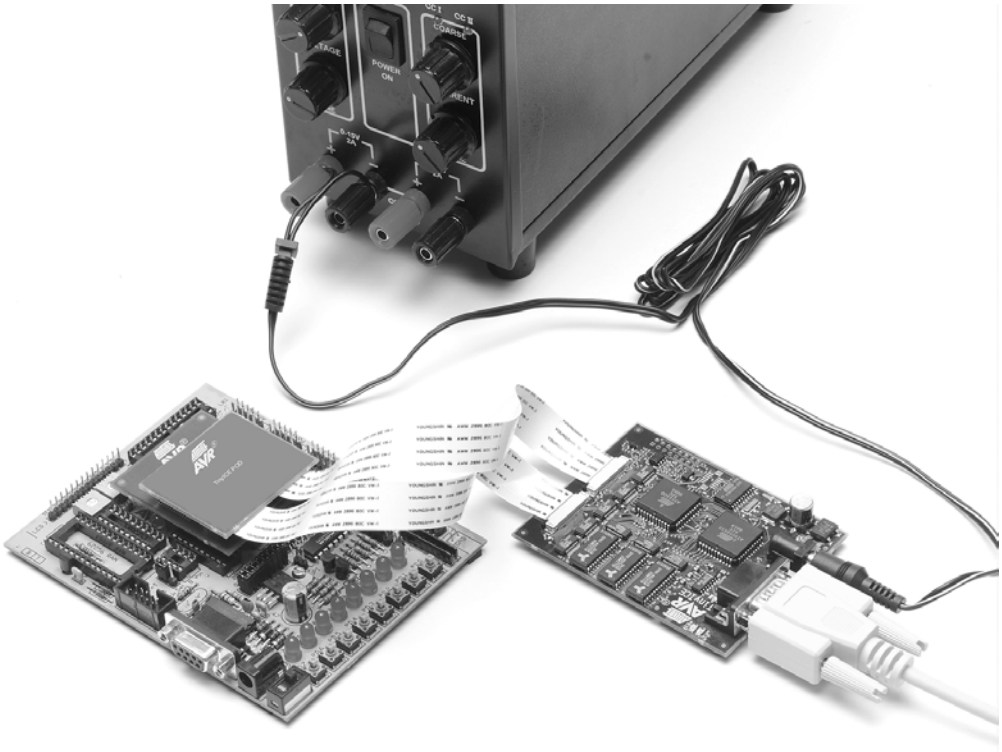
5. Connect the ICE200 to the host PC. Use the 9-pin RS-232C cable that is shipped with the ICE200. Connect the male cable connector to the ICE200 and the female cable connector to the host.

Figure 4-10. Connecting the ICE200 to a Host PC



6. AVR Studio can now be started. However, do not open any files before connecting power supply to the ICE200. If the ICE200 is not powered then AVR Studio cannot detect the emulator and therefore enters Simulation mode.
7. The ICE200 has no power switch. Just connect the power supply cable shipped with the ICE200 to a 9V DC power supply, and then connect the power cable to the ICE200 (see Figure 4-11). A battery eliminator is a good alternative to the laboratory power supply shown in the figure.

Figure 4-11. Connecting Power Supply to ICE200



8. Enable the target power supply. The red LED will now be lit, telling that power is present, but no connection to the host PC has been established.

The hardware is now ready for use. Use the checklist below to ensure that the setup was done correctly, and then proceed to the next section “Configuration”.

4.6.1 Checklist

1. ☐ Turn-off power on all units.
2. ☐ Insert personality adapter.
3. ☐ Pins on adapter agree with pins on target.
4. ☐ Mount pod.
5. ☐ Mount FPC (first time only, do not disassemble).
6. ☐ Connect RS-232C cable.
7. ☐ Turn-on power on all units.

4.7 Configuration

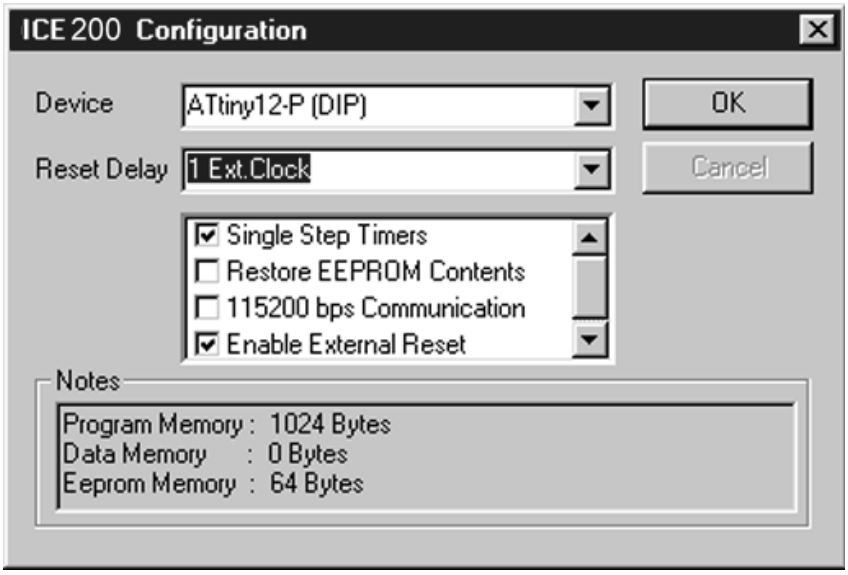
When the ICE200 is connected to the target application, the next step is to set the device configuration of the part you are using. This is required when an application code project is opened for the first time, but can later be changed in the emulator options menu. The configuration is stored in a separate file, [project name].avd, in the same directory as the application code.

4.8 Quick Start

Follow the procedure described below to configure the ICE200.

1. Connect the ICE200 and start AVR Studio as described in previous section.
2. Open the object file. If the file is opened in AVR Studio for the first time, the “Emulator Options” window automatically appears, see Figure 4-12. The red LED will now be turned off and the green will be turned on (if it is not already on), indicating that the connection between the ICE200 and the host PC is established.

Figure 4-12. ICE200 Configuration Dialog Box



3. Set up the desired configuration. A detailed description is found in the following sections; However, the default settings are sufficient in most cases due to the automatic personality adapter detection.
4. Press the “OK” button.
5. The ICE200 is now ready for use!

Tip!

To change configuration for the current project, select the Option – Emulator Options menu.

For advanced users, Section 4.9 describes, in more detail, the possible settings in the Options menu.

The software is now ready for use. Use the checklist below to ensure that the setup was done correctly. The ICE200 is now ready for debugging.

Tip!

If AVR Studio cannot connect to the emulator, make sure no other programs are using the serial port.

4.8.1 Checklist

1. ☐ The personality adapter selected corresponds to the device that is to be used.
2. ☐ Target application is correctly connected to the emulator (as described in previous section).
3. ☐ AVR Studio detects that the emulator is present when opening the application code.
4. ☐ The Options menu for ICE200 was shown when opening the source file (only first time).
5. ☐ The device in the Options menu correspond to the chip configuration you want to use.
6. ☐ The text “AVR Emulator” appears in the status bar of AVR Studio found in the lower right corner of the window.

4.9 Emulator Options Settings

4.9.1 Device Settings

Some devices have identical pinout and functionality. An example is the AT90S8515 and the AT90S4414 devices, there is no need for a personality adapter for each of them. The automatic personality adapter detection ensures correct pinout configuration of the AVR emulator chip. When selecting a specific device enter, the emulator options menu and select the device from the device list.

4.9.2 Clock Selection Settings

Many AVR microcontrollers have Fuse bits for selecting the Reset Delay time. The Reset Delay is necessary for the clock Oscillator to stabilize. The time it takes for the Oscillator to stabilize depends on the crystal or the resonator. If an external clock source is used then the Reset Delay can be set to only a few clocks. The Reset Delay Fuse bits (CKSEL/FSTRT depending on device) can be set or cleared by a Parallel or Serial Programmer in an actual device. In the emulator, they are set in the options menu.

For more information about the Reset Delay Fuses, please refer to the data sheets.

4.9.3 Single-step Timers Setting

This setting allows single-stepping of the timers if checked. If cleared the timers continue to count (if enabled) even after the program execution is stopped by a user break or Break Point. All other peripherals (SPI/UART/EEPROM/PORTs) continue to operate when the program execution is stopped.

This feature allows cycle-by-cycle debugging of the counter value, which is useful for event timing. However, in many cases, stopping the counter operation while debugging might not be desired. One example is when the timer is used in PWM mode. Stopping the timer in this case might damage the equipment that is being controlled by the PWM output.

Note that eventual Timer Interrupts will not be handled before execution is resumed.

4.9.4 EEPROM Restore Setting

Some AVR devices have On-chip EEPROM Data Memory. The ICE200 emulates the EEPROM by using an SRAM replacement inside the AVR emulator chip. This is done to eliminate problems with EEPROM write endurance. However, by doing so, a new problem is introduced since a power loss on the target will result in loss of the data stored in the SRAM that emulates the EEPROM.

A split solution handles the power loss situation. First select the EEPROM restore option. Before removing the power, take a snapshot of the EEPROM contents by pressing the EEPROM snapshot button (Figure 4-13). This will tell the ICE200 main board to read all EEPROM data into a buffer. When power is switched off and then on again, the ICE200 will restore the contents of the buffer to the SRAM before starting code execution.

Figure 4-13. EEPROM Snapshot Button

4.9.5 Communication Speed Setting

The default communication speed between AVR Studio and the ICE200 is 19,200 bps. If the option “115,200 bps Communication” is selected, the communication speed is changed to 115,200 bps. Some systems cannot handle this speed, so if AVR Studio looses contact with the emulator after enabling this option, it should be disabled.

4.9.6 Reset Pin Setting (ATtiny12 Only)

When selecting ATtiny12 an additional option selection appears in the emulator options dialog box. The ATtiny12 device has a programmable Fuse that lets the RESET pin function as an Input pin (PB5). When this option is selected, PB5 functions as a Normal Reset pin. When this option is not selected, PB5 functions as an Input pin. The device is then only Reset at Power-on or when giving a reset command from AVR Studio. Refer to the ATtiny12 data sheet for more information.

Section 5

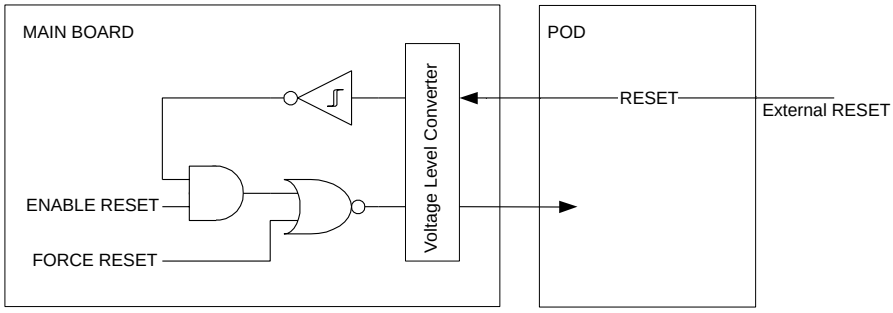
Special Considerations

The ICE200 accurately emulates most AVR features. However, there are some differences worth noting. Most of the exceptions apply to controlling the program flow, i.e., single-stepping and so on. Program flow control is an extension to the normal functions of the microcontroller that allows the user to do the debugging. This extension must not interfere with the normal program execution (Run mode). If the program execution is stopped (Stopped mode) and then restarted, or the program is executed line-by-line (single-stepping), the program functionality can, in some cases, be affected.

5.1 External RESET

The ICE200 main board has to be able to control the Reset pin on the AVR emulator chip. An External Reset Source must therefore go via the Control Logic as shown on Figure 5-1. This is handled automatically by the pod and main board

Figure 5-1. ICE200 – External RESET Circuit



The main board is working with 5V supply and the pod uses the target voltage. Therefore, a level converter is inserted between the two systems. The extra logic and the level converters introduce a small, and for most of the time, negligible delay. Note that the voltage converters do not handle a +12V input voltage on the RESET pin which is used for enabling the parallel programming on standard parts.

After a Power-up, the reset is forced active while configuring the AVR emulator chip, introducing a 1 - 10 ms delay.

NOTICE!

The internal pullup featured in the AVR devices is not present in ICE200. Therefore an external pullup must be present on the Reset pin to ensure correct behavior of the emulator.

5.2 SLEEP Instruction

If a SLEEP instruction is executed and sleep is enabled the AVR emulator chip will enter one of the AVR Low-power modes (Power-down, Power-save or Idle). The power consumption will be slightly higher compared to the real chip due to the higher complexity of the emulator chip. In room temperature (25°C) this is not significant.

Tip!

When debugging your application, you can replace the SLEEP instruction with a NOP by using macros.

IAR C example:

```
#ifdef   SLEEPEMU
#define  SLEEP() _SLEEP()
#else
#define  SLEEP() _NOP()
#endif
```

5.3 Watchdog Timer (WDT)

The Watchdog Timer operates asynchronously (it has its own clock) to the rest of the system. The Watchdog Timer Reset instruction is therefore repeatedly issued in Stopped mode to avoid false Resets if the Watchdog Timer is enabled. Debugging the exact timing of the WDT behavior while doing single-stepping or when stopping the program execution, is therefore not supported by the ICE200.

Disabling WDT is secured by the WDTTOE (WDT Turn-Off Enable) bit. Following is an assembly program example that shows a WDT disable sequence:

disableWDT:

```
  ( cli )                          ;(only needed if any interrupts are in use)
ldi     r16, (1<<WDTTOE)+(1<<WDE)  ;Set the WDT turn-off enable bit and the
                                   ;WDE bit
  out   WDTCR, r16
  ldi   r16, (0<<WDE)              ;Within four cycles, clear the WDEbit
  out   WDTCR, r16
done:
  ( sei )                          ;(only needed if any interrupts are in use)
```

The four cycle Time-out for the WDT disabling is not supported by the AVR emulator chip when in Stopped mode. This means that single-stepping this sequence will not disable the WDT.

Tip!

You can use the run to cursor to the instruction after the WDT Enable bit is cleared (labeled done in the example above), instead of doing single-stepping. Or use the following macro (for AVR assembler only).

MACRO:

```
.macro disableWDT
    ( cli )
  ldi     r16, (1<<WDTTOE)+(1<<WDE)
  out     WDTCR, r16
  ldi     r16, (0<<WDE)
  out     WDTCR, r16
  ( sei )
.endmacro
```

USAGE:

```
disableWDT
```

5.4 EEPROM

When writing to the on-board EEPROM on an AVR device, a special sequence must be used to ensure the integrity of the EEPROM data. Following is an assembly program example that shows an EEPROM write sequence:

```
writeEEPROM:
  ( cli )                   ;(only needed if any interrupts are in use)
  sbi     EECR, EEMWE
  sbi     EECR, EEWE
done:
  ( sei )                   ;(only needed if any interrupts are in use)
```

The four cycle Time-out for the EEPROM write is not supported by the AVR emulator chip when in Stopped mode. This means that single-stepping this sequence will not strobe an EEPROM write.

Tip!

You can use the run to cursor to the instruction after the EEPROM Write Enable bit is set (labeled done in the example above), instead of doing single-stepping. Or use the macro defined below.

MACRO:

```
.macro writeEEPROM
  ( cli )
  sbi   EECR, EEMWE
  sbi     EECR, EEWE
  ( sei )
.endmacro
```

USAGE:

```
writeEEPROM
```

5.5 I/O Port Access

A special situation occurs when single-stepping a change of the PORT value as shown in the following example:

```
write_with_readback:
  ldi     r16, 0xFF          ; Set all pins as output
  out     DDRx,r16           ; -"-
  out     PORTx, r16         ; Set the PORTx values
  in      r16,PINx           ; Read the PINx values
  in      r17,PINx           ; Read the PINx values
```

When running this example program at full-speed in the ICE200 or in a real chip, the value read back in r16 will not end up being the value written at the first line, but will contain the value the port pins had the cycle before the port was written. This is the correct behavior. The PINx value must be synchronized, and therefore it is delayed one cycle to avoid erratic port behavior caused by metastability. r17 will therefore contain the value written to the port. However, when the program is single-stepped, the value of the PINx will change immediately after the single-step and the value of r16 will contain the same value as before.

Changing pin values on an I/O port from AVR Studio when the ICE200 emulator is stopped does not represent any problem. However, note that, as for the single-stepping case, the pin values are changed immediately.

Clearly this is not a real problem, but it is important to be aware of the effects of the two cases described above. If not, an incorrect program might seem to work in the emulator, but will not work in the real chip.

5.6 16-bit I/O Access (Timer1 and A/D Converter)

Reading or writing 16-bit values directly from AVR Studio can cause some problems. To read, for example, the counter value from Timer1, a 16-bit value, one of the bytes must be stored in a temporary register. This temporary register will be corrupted if the 16-bit value is read when the program execution is stopped.

Tip!

Using the following macros (for AVR assembler only) will solve the 16-bit access problem when using symbolic debugging.

MACROS:

```
.macro outw
  (cli)
  out     @2, @0
  out     @2-1, @1
  (sei)
.endmacro

.macro inw
  (cli)
  in      @2, @0-1
  in      @1, @0
  (sei)
.endmacro
```

USAGE:

```
inw      r17, r16, TCNT1H        ; Reads the counter value
outw     TCNT1H, r17, r16        ; Writes the counter value
```

When using symbolic debugging in C, the entire C line is executed for each set. Therefore the 16-bit read or write problem will not occur in this situation.

5.7 UART Data Register

Reading the UART Data Register cleans the RXC bit in the UART Control Register. Hence, the monitor program does not attempt to read the UART Data Register. Therefore, the value displayed by AVR Studio for this register does not reflect the real value of this register.

5.8 ATtiny12

When emulating ATtiny12 there are some differences compared to the ATtiny12 device:

In ICE200 the internal RC Oscillator is running at a nominal speed of 2 MHz. This is twice the frequency of the ATtiny12 device.

In ICE200 PortB pin5 can only be used as Reset or general input. The alternative function as open drain output is not supported.

The Pull-up Disable (PUD, MCUCR, bit6) bit in ATtiny12 is not supported.

Tip!

Disable pull-ups individually by clearing the appropriate bits in the PortB Register.

5.9 Timer Interrupt Flags

Normally writing a “1” to a flag will clear it, so manually setting of flags is impossible. But when single stepping the timers the Timer Interrupt Flags can be set manually using the AVR Studio I/O view.

Note: Writing a “1” to the flags in software can’t set them, even when single stepping. Doing this will clear the flags as normal. The manual setting of Timer Interrupt Flags can only be done from AVR Studio’s I/O view, and only in Timer Single Stepping mode.

5.10 Clear Timer/Counter1 on Compare Match

When the Clear Timer/Counter1 on Compare Match control bit is set and the Timer is single stepped the count sequence is as follows:

… | C-2 | C-1 | 0 | … i.e., the Timer is cleared before reaching the compare value.

When in Run mode the counter behaves normally, i.e., the sequence is
… | C-2 | C-1 | C | 0 | …

Note: If the program executes a 2-cycle (or more) instruction during the Compare Match the count sequence will be the same as in Run mode.

Tip!

When reaching the compare value use the run to cursor feature or a Break Point to ensure that the emulator is in Run mode during Compare Match.

5.11 Timer/Counter1 Output Compare A Match Interrupt

Single stepping Timer1 when the Clear Timer/Counter1 on Compare Match control bit is set will not set the Output Compare Flag 1A and the interrupt routine will not be executed. This occurs only in Single Step Timers mode. In Run mode the Timer behaves normally, i.e., the Output Compare Flag 1A in TIFR is set regardless of the Clear Timer/Counter1 on Compare Match control bit's value.

Note: If the program executes a 2-cycle (or more) instruction during the Compare Match, the Output Compare Flag 1A will be set, and the associated interrupt routine will be executed.

Tip!

Place a Break Point at the first instruction of the Timer1 Compare A Interrupt Handler Routine. When the Timer/Counter1 value is close to the Compare A value run the program at full speed. The interrupt routine will be executed, and a program break will occur at the Break Point. If the actual jump to the Interrupt Vector needs to be verified the Output Compare Flag 1A can be set manually using AVR Studio's I/O view.

5.12 Power-down Mode

When entering Power-down mode the clock will continue to run on all devices but ATtiny12.

Section 6

Appendix

6.1 Emulating AT90S1200 and ATtiny10/11

6.1.1 Using the Include Files

Always use include files for the I/O Registers addresses and for bit definitions in your source code files. This will ease the process of porting code from one microcontroller to another. The files can be found on the CD-ROM which is included in the ICE200 kit. Copy the include file to your project directory, and include it in the top of the program code as shown below:

(AVR Assembler example)

```
.include "1200def.inc"
```

Then, when writing a value to a I/O Register, use the following notation:

(AVR Assembler example)

```
ldi     r16, (1<<DDD1) + (1<<DDD4)    ; Set port D pin 1 and 4 as output and
                                      ; the rest as input.
out     DDRD, r16
```

Note the use of the bit definitions. DDD1 and DDD4 are pin definitions in form of bit position, and therefore they must be shifted this number of bits to the left to make a correct mask.

Interrupt Vector locations might differ from part-to-part. This is easily handled by using the vector definitions found in the include files.

(AVR Assembler example):

```
.include "1200def.inc"
.org     0
  rjmp   RESET_Handler
.org     INT0addr
  rjmp   INT0_Handler
.org     OVF0addr
  rjmp   OVF0_Handler
.org     ACIaddr
  rjmp   ACI_Handler
... ( program code starts here )
```

6.1.2 Using the ATtiny12 Adapter for Emulating the ATtiny10/11

The ATtiny10 and ATtiny11 are both subsets of ATtiny12. Therefore, it is possible to select the ATtiny12 device when configuring the ICE200 to support either ATtiny10 or ATtiny11. These devices all have the same pinout, but ATtiny10/11 does not have the following features:

- **Brown-out Detection (BOD)**
- **Calibration of the RC Oscillator**
- **Reset Source Register**
- **Band-gap Reference on the Comparator**
- **EEPROM Interrupt**

Also note that the startup times differ slightly between the devices. Please refer to the data sheets for more detailed information.

6.1.3 Using the AT90S2313 Adapter for Emulating the AT90S1200

The AT90S1200 can be defined as a subset of AT90S2313. They have the same pinout, but AT90S1200 does not have the following features:

- **UART**
- **SRAM**
- **Memory Access Instructions (ld/st/lds/sts/ldd/std/lpm)**
- **16-bit Arithmetic Instructions (adiw/sbiw)**
- **INT1**
- **Timer/Counter 1 and Input Capture**
- **Stack Pointer to SRAM (AT90S1200 has a Three-level Hardware Stack)**

Avoiding the use of these features and using only half the program and EEPROM memories allows the AT90S2313 to be used when emulating AT90S1200.

IMPORTANT!

Since the AT90S2313 uses a Stack Pointer, this has to be initialized. The simplest way is include the following lines at the top of the program code:

(AVR Assembler example):

```
ldi     r16, 0x65        ; Set the stack Pointer to point at the address to
                         ; give a three level deep stack
out     0x3D, r16
```

The AT90S2313 has no RC Oscillator, so this feature found on the AT90S1200 can not be supported.

Since the AT90S2313 features the EEMWE bit for writing data to the EEPROM Memory, this must also be added to the AT90S1200 code if the EEPROM is used.

Include the AT90S2313 file when emulating AT90S1200 to get the interrupts placed on the right locations, see Section 6.1.1.

6.2 AVR Emulator Chip Errata

Latest errata is found on the Atmel web site: www.atmel.com.

6.3 Troubleshooting

If you experience problems when installing AVR Studio, connecting the emulator or configuring the emulator, first of all use the checklists in the previous sections to confirm that you have done installation and the setup of the emulator correctly.

6.3.1 Feedback and Support

To get correct answers to your problems, please include the following details in your request.

ICE200/AVR Studio:

- Details of which release of ICE200 you are using.
- Details of the platform on which you are running (OS, amount of memory, etc.).
- A small stand-alone sample of code which reproduces the problem.
- A clear explanation of what you expected to happen, and what actually happened.
- The commands or menu selections you used.
- Sample output illustrating the problem.
- The information shown in the About dialog box in AVR Studio (version numbers).
- The emulated device.

Documentation:

- The user guide title and revision.
- The page number(s).
- A concise explanation of the problem.

General suggestions for additions and improvements are also welcome.

6.4 Contact Information

For technical support, please contact your distributor, Atmel sales representative or local Atmel sales office. Atmel sales offices and distributors are listed on the Atmel web site: www.atmel.com.

Section 7

ICE200 Surface Mount Adapter Kit User Guide

The ICE200 Surface Mount Adapter Kit is used together with an ICE200. It is a low-cost solution to In-Circuit Emulation on designs using SMD versions of the AVR family of microcontrollers.

The ICE200 Surface Mount Adapter Kit is available from any of Atmel's franchised distributors. The ordering code is ATAVRSMD.

7.1 Supported Devices

The following devices are supported by the ICE200 Surface Mount Kit:

ATtiny12, AT90S2313, AT90S4414/8515, AT90S4434/8535

ATtiny11 (using ATtiny12 adapters)

AT90S1200 (using AT90S2313 adapters)

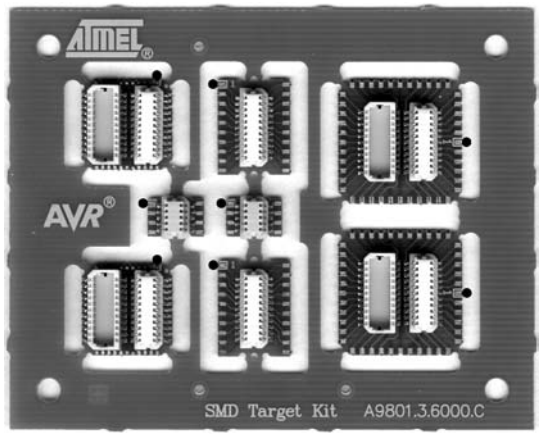
The 20-pin SSOP package (AT90S1200-xYx) and the 32-pin TQFP packages (AT90S2333/4433-xAx) are not supported by this kit due to size constraints.

7.2 ICE200 Surface Mount Kit Contents

The ICE200 Surface Mount Adapter Kit consists of two different types of adapters: The SMD personality adapters and the SMD target adapters. The personality adapters determines which device is emulated, while the target adapters makes it possible to connect the personality adapters to the correct footprint.

The target adapters are delivered as one single board, the SMD Target Kit, as shown in Figure 7-1. The kit contains two 44-pin TQFP adapters, two 8-pin SOIC adapters, two 20-pin SOIC adapters, and two 44-pin PLCC adapters. The four personality adapters are also delivered as one board as shown in Figure 7-2. Note that R104 and R105 on ATadap3410 are not mounted.

Figure 7-1. SMD Target Kit.



Note: The black dots indicate pin1 of each adapter.

Figure 7-2. SMD Personality Adapter Kit

7.3 Using the ICE200 Surface Mount Adapter Kit

The combination of one personality adapter and one target adapter replaces one of the PDIP personality adapters that are delivered with the ICE200. Table 7-1 shows which adapter to use for different devices. Please note that the target adapters are not designed to be reused on different boards. Unsoldering a target adapter is possible, but difficult, and the target adapter may be damaged in the process. The kit therefore includes two of each target adapter.

Table 7-1. Adapter Combinations Used for Different Devices and Packages

Device	Package Type	SMD Personality Adapter	SMD Target Adapter
AT90S1200-xSx	SOIC-20	ATadap3310	SOIC-20
AT90S2313-xSx	SOIC-20	ATadap3310	SOIC-20
AT90S4414-xJx	PLCC-44	ATadap3010	PLCC-44
AT90S4414-xAx	TQFP-44	ATadap3010	TQFP-44
AT90S8515-xJx	PLCC-44	ATadap3010	PLCC-44
AT90S8515-xAx	TQFP-44	ATadap3010	TQFP-44
AT90S4434-xJx	PLCC-44	ATadap3110	PLCC-44
AT90S4434-xAx	TQFP-44	ATadap3110	TQFP-44
AT90S8535-xJx	PLCC-44	ATadap3110	PLCC-44
AT90S8535-xAx	TQFP-44	ATadap3110	TQFP-44
ATtiny11-xSx	SOIC-8	ATadap3410	SOIC-8
ATtiny12-xSx	SOIC-8	ATadap3410	SOIC-8

The size of the personality adapters is 31.5 x 21.5 mm. They will therefore cover an area larger than the SMD devices they emulate. The distance between the personality adapter and the target board will be 5 mm when the adapters are mounted. Components larger than this will have to be removed or moved in the prototype setup. Similarly, the distance between the target board and the ICE200 Pod is 10 mm. Figure 7-3 shows a TQFP adapter mounted on a PCB. The electrolytic capacitor in the lower right corner is more than 5 mm high and must be removed. As this is a through hole component, it can soldered from the other side of the board or just bent slightly as shown in Figure 7-4.

Figure 7-3. TQFP44 Target Adapter Mounted on PCB

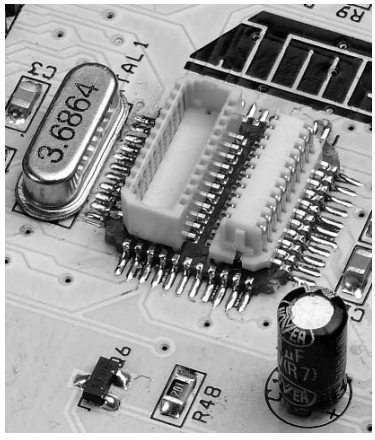


Figure 7-4. Personality Adapter Mounted on Target Adapter

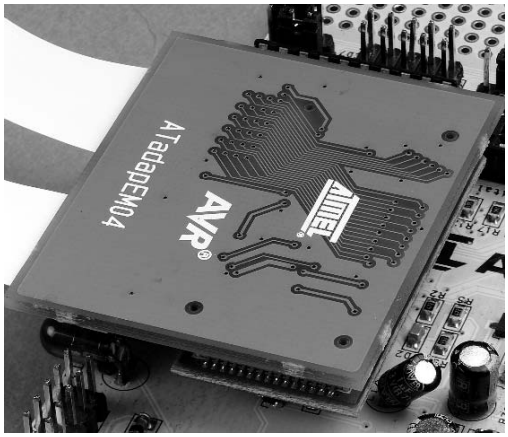
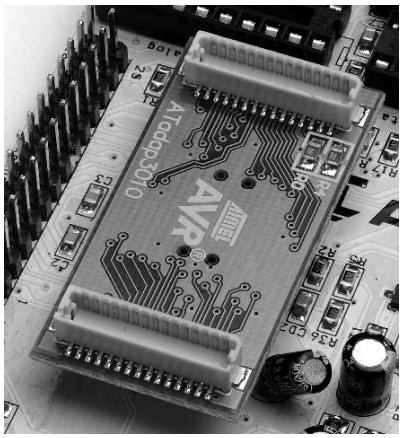


Figure 7-5. ICE200 Pod Mounted on Top of Personality Adapter



7.3.1 POD Dimensions

Figure 7-6 and Table 7-2 shows the POD dimensions for adapting TQFP44 to the ICE 200.

Figure 7-6. POD Illustration

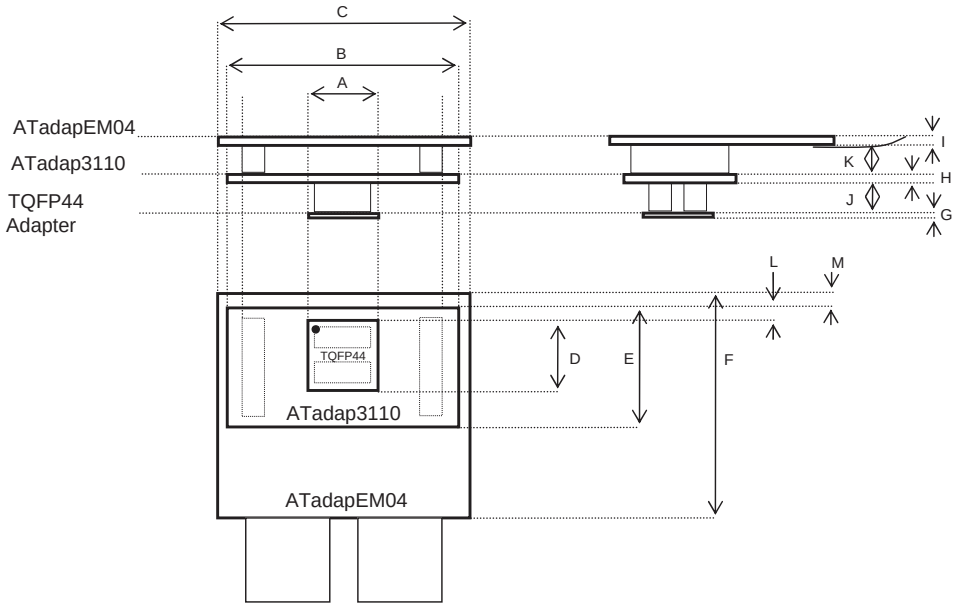


Table 7-2. POD Dimensions

Units	Label	Inches	mm
TQFP44 Adapter Width	A	0.47	12
ATadap3110 Width	B	1.64	42
ATadapEM04 Width	C	1.8	46
TQFP44 Adapter Length	D	0.47	12
ATadap3110 Length	E	0.83	21
ATadapEM04 Length	F	1.6	40
TQFP44 Adapter Height	G	0.0325	0.8
ATadap3110 Height	H	0.065	1.6
ATadapEM04 Height	I	0.065	1.6
TQFP44-ATadap3110 Spacing	J	0.2	5
ATadap3110-ATadapEM04 Spacing	K	0.2	5
TQFP44-ATadap3110 Offset 1	L	0.16	4
ATadap3110-ATadapEM04 Offset 1	M	0.05	1.3

7.4 Soldering

The best way to solder an SMD adapter onto a PCB is to use an SMD rework station. For most types of rework stations, mounting the adapters should be similar to mounting a regular SMD component on the PCB. For users who are used to soldering SMD components by hand, it should not cause much problems to solder the SMD adapters.

Required tools include a good soldering iron, thin solder (made for SMD-soldering), cleaning fluid (alcohol or some other flux solvent), flux, a side-cutter, a magnifying glass (8x or stronger is recommended), unsolder braid and a multimeter.

Make sure there is plenty of light at the workplace.

Prepare the target board, verify that there is sufficient room for the personality adapter and the ICE200. If it is necessary to move components, these can be connected using wires or mounted in a position where they will not be in contact with the personality adapter or the ICE200. Hole mounted components can often be mounted on the opposite side of the board.

If you plan to unsolder the target adapter later, please read the unsoldering section as well before you start soldering an adapter onto the target board. Unsoldering is easier if a small piece of tape, paper or similar non-conductive material is placed between the SMD adapter and the target board. This will form a small gap between the adapter and the board, making it easier to remove the solder between them. A larger gap will make unsoldering easier, but it will also make the soldering more difficult.

Remove the SMD-adapter you intend to use from the SMD target kit using a pair of side-cutters. If necessary, cut away the remaining fiberglass supports on the corners of the TQFP44 and PLCC44 adapters and on the ends of the SOIC8/SOIC20 adapters.

The adapters have been tested for short circuits and broken wires during production. However, there may be some small pieces of copper left on the edges due to the production process. Inspect the adapters with a magnifying glass. This is particularly important on the TQFP-adapters, as the lead pitch here is only 0.8 mm. It is recommended to rub the edges of the adapters a bit using fine sandpaper. The connection between the top and bottom layers of the adapters is in the bottom of the grooves. Sanding the edges carefully will therefore not affect the operation of the adapter as long as the groove remains. If in doubt, check the adapter for short circuits using a multimeter.

Use some cleaning fluid to clean he adapter and the area of the target PCB where you intend to mount the adapter.

Add flux to the target adapter and to the footprint on the PCB. This makes it a lot easier to solder in the adapter and reduces the chances for short circuits due to solder bridges dramatically. The flux in the solder's core is not always sufficient, especially if the solder joint is reheated several times. If a corrosive flux is used, it should be cleaned off afterwards. If a “no-clean” flux is used, this is not necessary.

Place the adapter on the footprint on the PCB. This is the trickiest and may be the most important part of the job. Shift the adapter around until it matches the footprint perfectly. Make sure to place the adapter in the right direction. Pin1 is marked with white paint on each adapter. The black dots in Figure 7-1 indicate pin1 of each adapter.

When the adapter fits perfectly, carefully solder one pin on one of the corners. Then recheck the position. If the adapter has moved, reheat the solder on the pin and adjust the adapter until it matches perfectly again. Then solder a pin on the opposite corner. If the adapter still matches the footprint perfectly, double-check that the orientation of the adapter is correct before you solder all the remaining pins. It requires a lot of time and skill to unsolder an incorrectly mounted adapter!

The adapters can be soldered in two ways: The first option is to solder pin by pin, using very thin solder and a thin soldering iron. For boards with solder masks, the other option is possible: Solder all pins on one side, using excessive solder and flux, and then remove the extra solder using the unsolder braid. This will leave exactly the necessary amount of solder between the adapter and the PCB. This will also be the way to remove solder bridges.

Clean the remaining flux from the PCB. Inspect the solder with both a magnifying glass and a multimeter to ensure there are no errors. Check for short circuits and for pads that are not connected.

Break apart the personality adapters, and mount the correct adapter on top of the target adapter as shown in Figure 7-4. The adapters can only be mounted one way. Finally, connect the ICE200 pod as shown in Figure 7-5. Alternatively, connect the personality adapter to the ICE200 before both are connected to the target adapter.

7.4.1 Checklist for Mounting SMD Target Adapters

- ☐ Remove the SMD target adapter from the frame.
- ☐ Check it for short circuits. Sand the edge if needed.
- ☐ Clean the adapter and the PCB.
- ☐ Add flux to the adapter and PCB.
- ☐ Place the adapter on the PCB. Make sure the orientation is correct.
- ☐ Solder two pads on opposite corners. Adjust the position of the target adapter if necessary.
- ☐ Recheck the orientation of the adapter.
- ☐ Solder the remaining pads.
- ☐ Remove all excessive flux using some cleaning fluid.
- ☐ Check for short circuits with a magnifying glass and multimeter.
- ☐ Test.

7.5 Unsoldering

As mentioned before, the target adapters are not designed to be reused on different boards. However, testing has shown that unsoldering a target adapter is possible, although it requires extensive soldering skills.

It may not be possible to reuse the adapters afterwards. Unsoldering is primarily done to be able to solder an AVR device onto the target board instead of the SMD adapter.

Similar to soldering, special equipment makes unsoldering much easier. The best options are an SMD rework station or a special soldering iron made for unsoldering SMD parts. It is then easy to melt all solder joints at the same time and remove the adapter from the target PCB.

It is more difficult to unsolder an adapter using only a soldering iron. The methods used for unsoldering SMD ICs are difficult to use, as the trick here is usually to unsolder pin by pin or to cut the pins away. The SMD target adapters are rigid, so it is necessary to heat all pins at the same time.

Required tools include a good soldering iron (50W or better) and all the other tools listed under the soldering section. In addition to this, a piece of copper wire (AWG 20-24/0.5-0.8 mm, single cord, not insulated) is needed.

The copper wire conducts heat and stores thermal energy, and it will be used as a special unsoldering iron.

First, remove as much solder as possible using the unsolder braid. If the adapter was mounted with a piece of tape or paper between the adapter and the target PCB, it may

be possible to remove all the solder this way. This is particularly true if the gap is wide enough to fit the unsolder braid between the adapter and the PCB. If this does not work, it is necessary to use the copper wire as described below.

Solder the wire along one side of the adapter. Bend it around to the next side, solder it and continue until you have wire all around the adapter. It is important to place the wire as tightly as possible around the adapter. The next step is to heat all the solder around the adapter and remove the adapter. This can be done in one of two ways:

1. Heat up two sides at the time. (One side at the time for SOIC adapters.) Move the soldering iron along the wire, add solder until it conducts heat to all the pins and pads. When all pins are loose, lift the corner carefully. Then move on to the next corner and so on until the whole adapter has been lifted up 1 - 2 mm. Use unsolder braid to remove the solder. After this, the adapter should be loose from the PCB. This method does not heat up nearby components too much. You can take pauses to let the PCB and nearby components cool down. However, there is a higher chance of damaging the tracks on the PCB than if you are using the second method.
2. Heat up all the sides at the same time. Move the soldering iron along the wire and add solder until it conducts heat to all the pins and pads. Using two soldering irons is quite effective, but not necessary. The clue is to keep adding heat until all the solder has melted and then simply remove the adapter. This method has less risk of damaging the PCB, but the massive heat required may overheat some of the nearby components on the PCB.

Remove all excessive solder from the footprint on the target PCB using the unsolder braid. Be careful not to damage the tracks or the footprint. Finally, clean off all the remaining flux on the PCB.

7.5.1 Unsoldering Checklist (Method 1)

- ☐ Remove as much solder as possible.
- ☐ Solder in heat conductor (copper cable).
- ☐ Massive heat up (two sides at the time).
- ☐ Carefully lift the adapter, one corner at the time.
- ☐ Repeat step 3 and 4 until the adapter has been lifted 1 - 2 mm up from the PCB.
- ☐ Remove the copper-cable on one side at the time.
- ☐ Remove excessive solder using an unsolder braid.
- ☐ Remove the now unsoldered adapter.
- ☐ Clean the footprint for solder and flux.

7.5.2 Unsoldering Checklist (Method 2)

- ☐ Remove as much solder as possible.
- ☐ Solder in heat-conductor (copper wire).
- ☐ Massive heat up on all sides at the same time.
- ☐ Remove the adapter when all the solder has melted.
- ☐ Clean the footprint for solder and flux.

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

1413D–AVR–02/03 /0M