USER GUIDE

ATmega168 Xplained Mini User Guide

Introduction

This user guide describes how to get started with the Atmel® ATmega168 Xplained Mini board.

The ATmega168 Xplained Mini evalutation kit is a hardware platform to evaluate the Atmel ATmega168 microcontroller. The evaluation kit comes with a fully integrated debugger that provides seamless integration with Atmel Studio 6.2 (and later version). The kit provides access to the features of the ATmega168 enabling easy integration of the device in a custom design

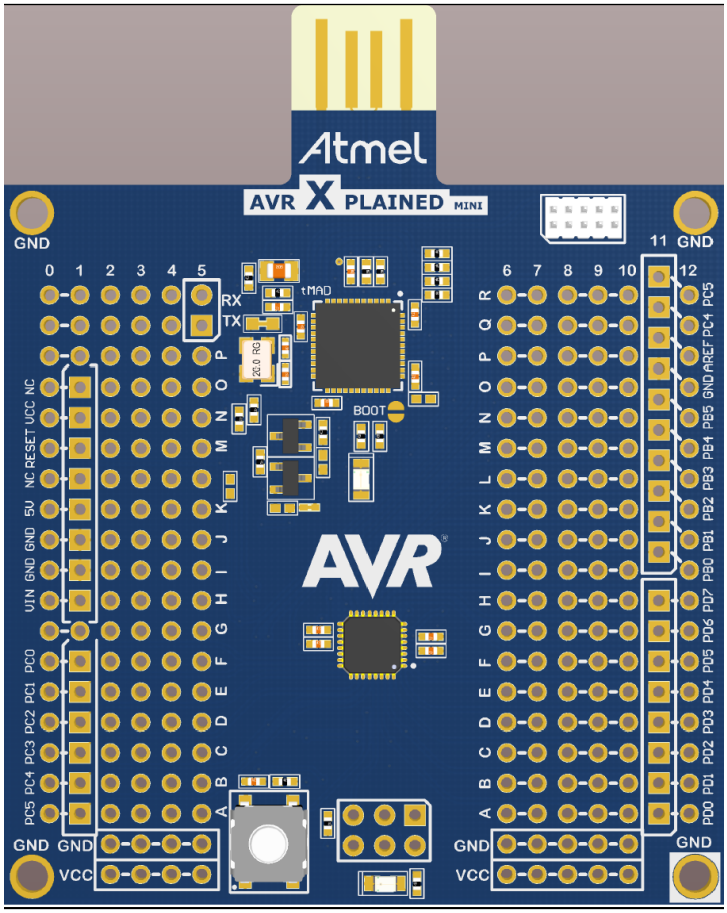


Table of Contents

1. Getting Started

1.1 Features

The ATmega168 Xplained Mini evaluation board provides a development platform for the Atmel ATmega168.

1.2 Design Documentation and Related Links

The most relevant documents and software for the evaluation board.[1]

1.3 Board Assembly

The Xplained Mini board is very flexible and can be used in a number of ways. E.g. as your own prototype for SW development and HW verification.

1.3.1 In Customer Development Assembly

The ATmega168 Xplained Mini board can be wired into the customer prototype assembly by using the on-board connector grid, where most target signals are available.

1.3.2 Connecting an Arduino Shield

By assembling receptacles in the marked positions Arduino® shields can be mounted.

1.3.3 Standalone Node

The ATmega168 Xplained Mini board can be used as a standalone node - use the 4xAAA battery pack available in Atmel store to provide power.

1.4 Connecting the Kit

How to connect the evaluation board.

1.4.1 Atmel Studio

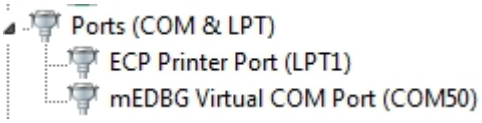
How to connect the ATmega168 Xplained Mini board assembly to Atmel Studio.

1. Download and install Atmel Studio[2] version 6.2 or newer.
2. Launch Atmel Studio.
3. Connect the board to the USB port and it will be visible in Atmel Studio.

1.4.2 Connect the ATmega168 UART to the mEBDG COM Port

All Xplained Mini boards have an embedded debugger (mEBDG) with a number of features, among them a CDC/COM port which enables the user to connect the ATmega168 UART to the PC.

1. Connect the mEDBG USB to the PC.
2. Use the Device Manager to find the COM port number.
3. Default COM port settings are 9600baud N81. The COM port settings can be changed using the Device Manager.



1.5 Programming and Debugging

How to program and debug the Xplained Mini board.

1.5.1 Programming the Target Using mEDBG

Using the Embedded Debugger on the Xplained Mini board to program the ATmeag168 via the SPI bus.

[1] http://www.atmel.com/tools/XplainedMini.aspx
[2] http://www.atmel.com/tools/atmelstudio.aspx

1. Connect the mEDBG USB to the PC.
2. Go to Atmel Studio: Tools/Device Programming, and select the connected mEDBG as Tool with Device = ATmega168PA and Interface = ISP, click Apply. Note that if ISP programming fails it could be because debugWIRE is enabled. See debugging chapter on how to disable debugWIRE mode: “Debugging the Target Using mEDBG” on page 4.
3. Select "Memories" and locate the source hex or elf file and click Program.
4. If the source contains fuse settings go to "Production file" and upload the elf file and program the fuses.
5. To set fuses manually click Fuses and select the setting.

   Recommended fuse setting:

   BOOTSZ = 1024W_1C00, BOOTRST = [ ], RSTDISBL = [ ], DWEN = [ ], SPIEN = [X], WDTON = [ ], EESAVE = [ ], BODLEVEL = DISABLE, CKDIV8 = [ ], CKOUT = [ ], SUT_CKSEL = EXTCLK_6CK_14CK_65MS

Important If any other cpu clk than the external clk supplied by the mEDBG is used the debugWIRE is not guaranteed to work.

The mEDBG will prevent writing certain fuse combinations in order to protect your kit.

1.5.2 Debugging the Target Using mEDBG

Using the Embedded Debugger on the Xplained Mini board to debug the ATmega168 via debugWIRE.

1. Start Atmel Studio.
2. Connect the mEDBG USB to the PC.
3. Open your project.
4. In the Project menu select the project properties page, select the Tools tab and select mEDBG as debugger and debugWIRE as interface.
5. In the Debug menu click Start Debugging and Break.
6. Atmel Studio will display an error message if the DWEN fuse in the ATmega168 is not enabled, click YES to make Studio set the fuse using the ISP interface.
7. A debug session is started with a break in main, debugging can start.
8. When exiting debug mode select "Disable debugWIRE and Close" in the Debug menu, this will disable the DWEN fuse.

Important If any other cpu clk than the external clk supplied by the mEDBG is used the debugWIRE is not guaranteed to work.

1.5.3 Programming the Target Using an External Programmer

How to program the target ATmega168 using the AVR® JTAGICE mkII, JTAGICE3, or other Atmel Programmers.

1. Connect the External Programmer to the PC.
2. Connect the External Programme to the evaluation board connector (J204) (Need the 6-pin 100mil adapter connected to the JTAGICE).
3. Go to Atmel Studio: Tools/Device Programming, and select the External Programmer connected as Tool, Select Device = ATmega168PA, Interface = ISP and click Apply.
4. Select "Memories" and locate the source hex or elf file and click Program.
5. If the source contains fuse settings go to "Production file" and upload the elf file and program the fuses.

Recommended fuse setting:

BOOTSZ = 1024W_1C00, BOOTRST = [ ], RSTDISBL = [ ], DWEN = [ ], SPIEN = [X], WDTON = [ ], EESAVE = [ ], BODLEVEL = DISABLE, CKDIV8 = [ ], CKOUT = [ ], SUT_CKSEL = EXTCLK_6CK_14CK_65MS

1.5.4 Programming the ATmega32U4 Using an External Programmer

How to program the ATmega32U4 using the AVR® JTAGICE mkII, JTAGICE3, or other Atmel Programmers.

To restore the mEDBG FW use the /tools/mEDBG/mEDBG_fw.zip from the Studio installation.

1. Connect the External Programme to the PC.
2. Connect the External Programme to the board connector (J100).
3. Go to Atmel Studio: Tools/Device Programming, and select the External Programme connected as Tool, select Device = ATmega32U4, Interface = JTAG and click Apply.
4. Select "Memories" and locate the source hex or elf file and click Program.
5. If the source contain fuse settings go to "Production file" and upload the elf file and program the fuses.

   Recommended fuse setting:

   BODLEVEL = 3V5

   HWBE = [X]

   OCDEN = [ ]

   JTAGEN = [X]

   SPIEN = [X]

   WDTON = [ ]

   EESAVE = [X]

   BOOTSZ = 2048W_3800

   BOOTRST = [ ]

   CKDIV8 = [ ]

   CKOUT = [X]

   SUT_CKSEL = EXTXOSC_8MHZ_XX_258CK_65MS

Important CKOUT must be enabled the provide clock to the target.

1.5.5 Bootloader

This section describes how to use the bootloader to program the ATmega32U4.

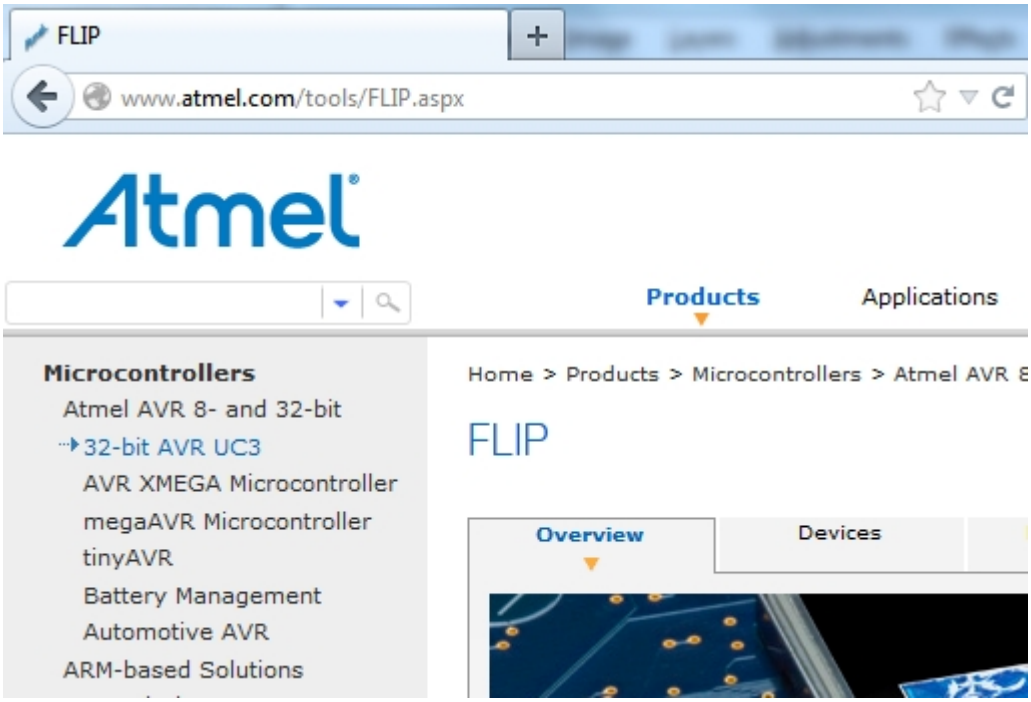
1. Install the Bootloader interface on the PC as described in “How to Install the "Bootloader PC tool"” on page 6.
2. Start the Bootloader PC GUI "FLIP".
3. Short strap J102.

   Insert the USB stick in the PC.
4. Select Device = ATmega32U4 (Device - Select).
5. Select USB communication (**Ctrl+U**).
6. Select memory area to program (Use the toggle memory button bellow the Atmel logo).
7. Select Load Hex file (**Ctrl+L**).
8. Select Programming Options.
9. Click "Run", observe status in status field.

1.5.6 How to Install the "Bootloader PC tool"

How to install the Bootloader PC GUI tool,

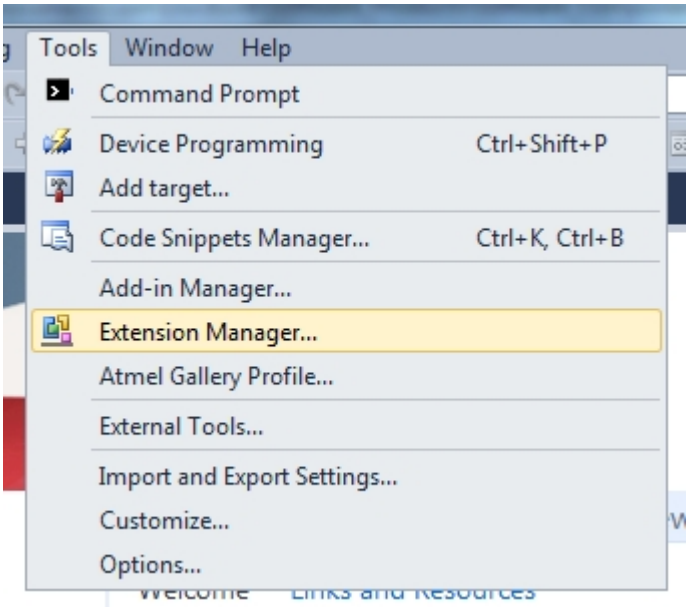
1. Download the Flip "in system programming tool" installer from http://www.atmel.com/tools/FLIP.aspx[3].

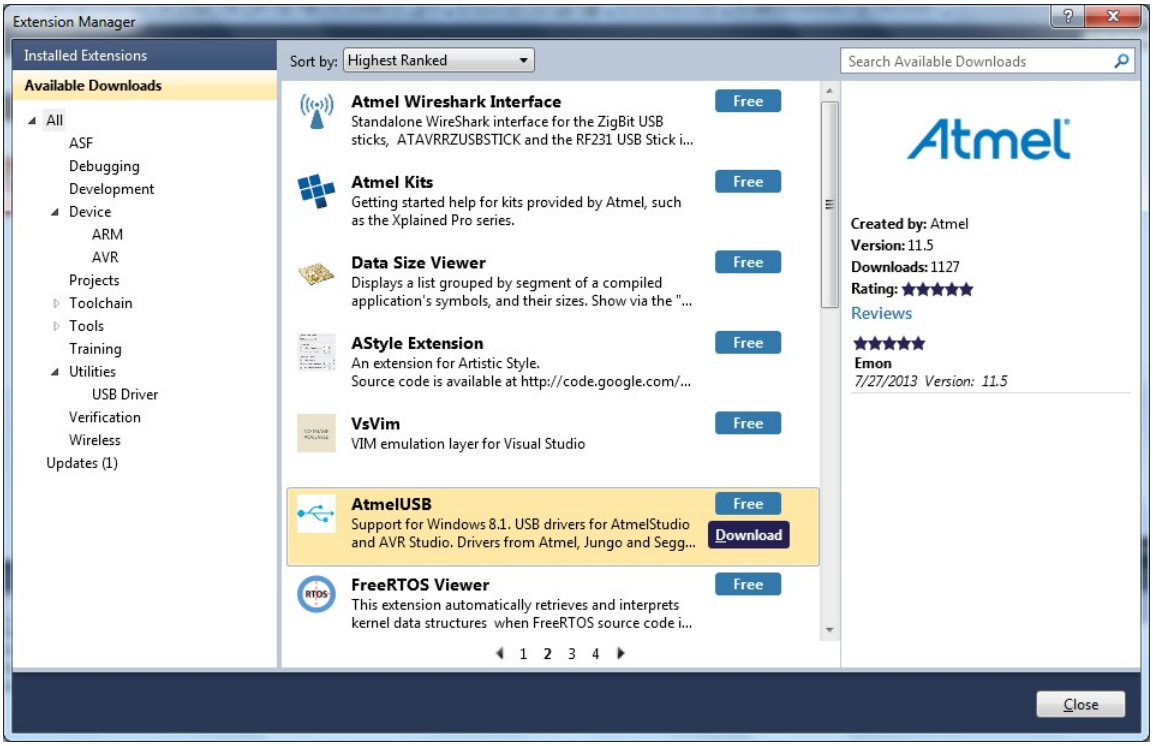


2. Run the Flip Installer.

[3] http://www.atmel.com/tools/FLIP.aspx

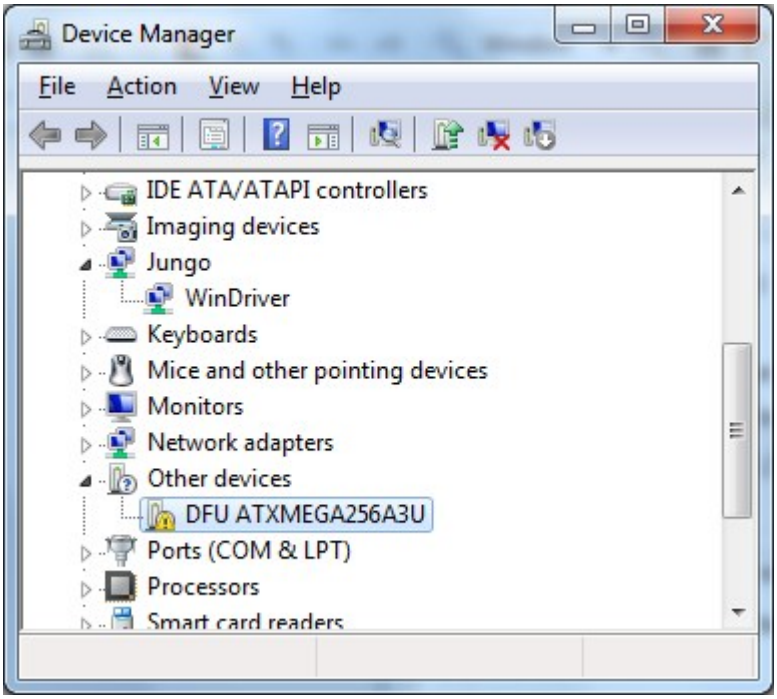
3. Download the Atmel USB extension and run the installer.

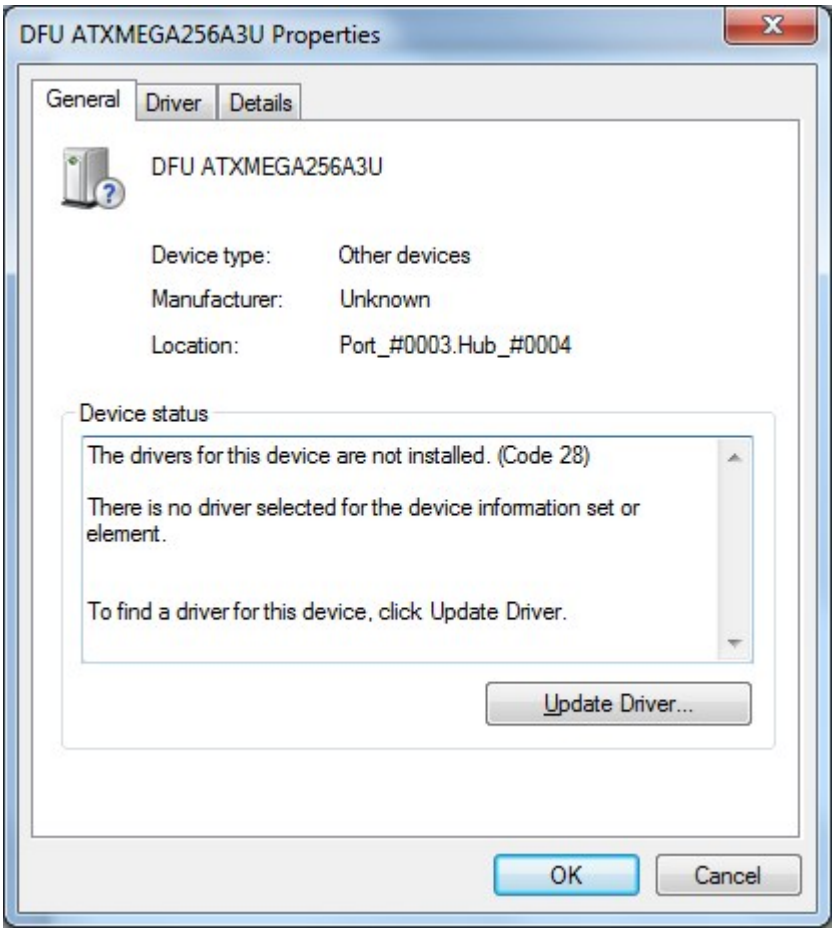




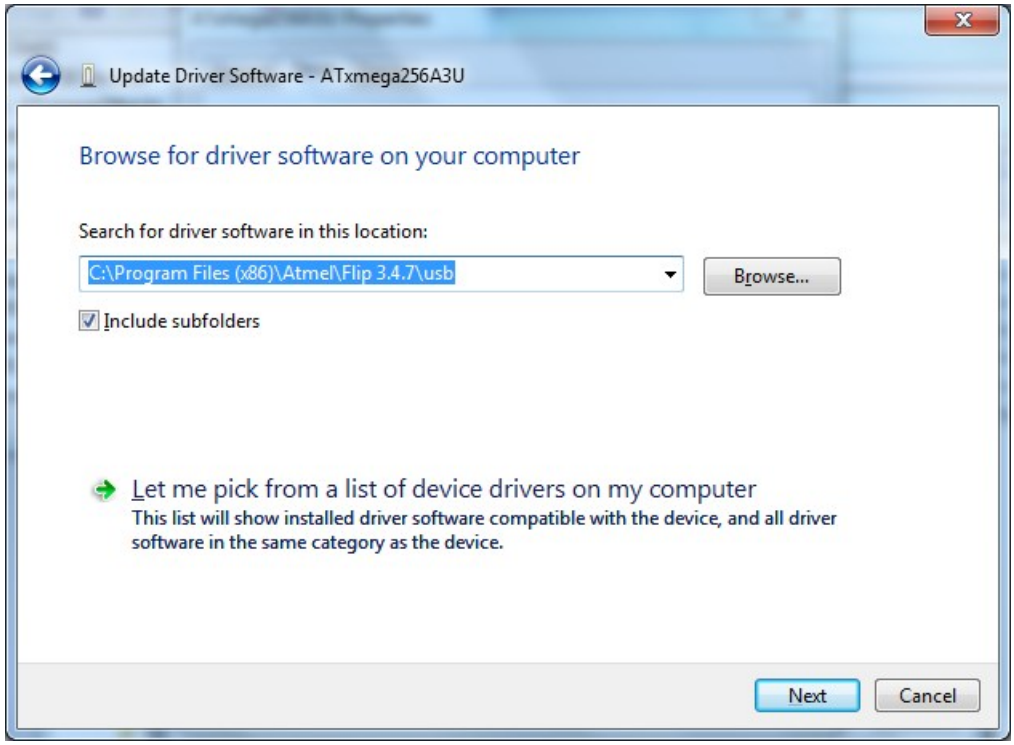
4. Start Flip - if the USB stick is not recognized continue with step 5.

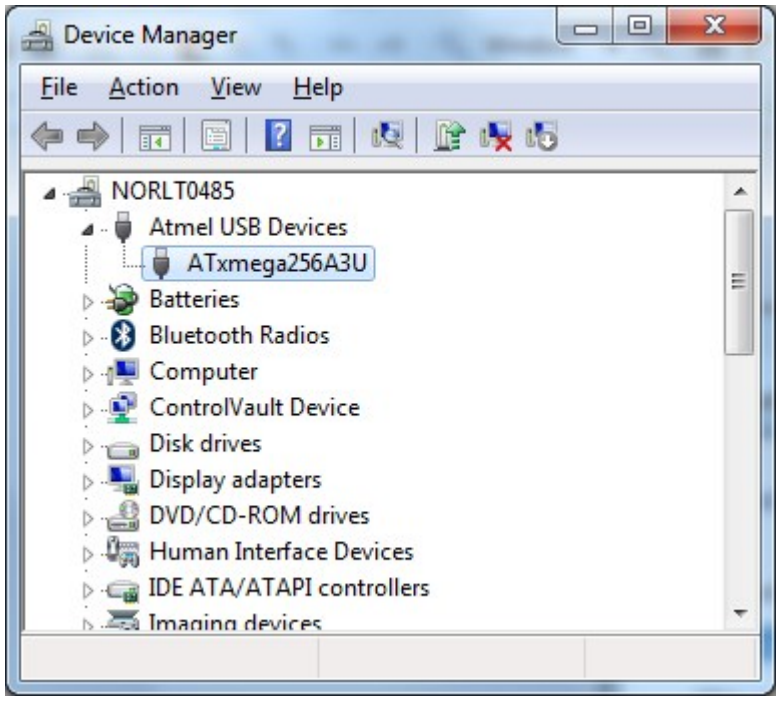
5. Update the USB DFU driver.





Clik the Update Driver button and select "Browse my computer-..."





1.6 Available Example Code

The ATmega168 is preprogrammed with a demo program, ReMorse. Source code is available in Atmel Spaces[4].

When the CDC COM port is connected to a terminal window, the text you write will be transmitted via the LED in Morse code.

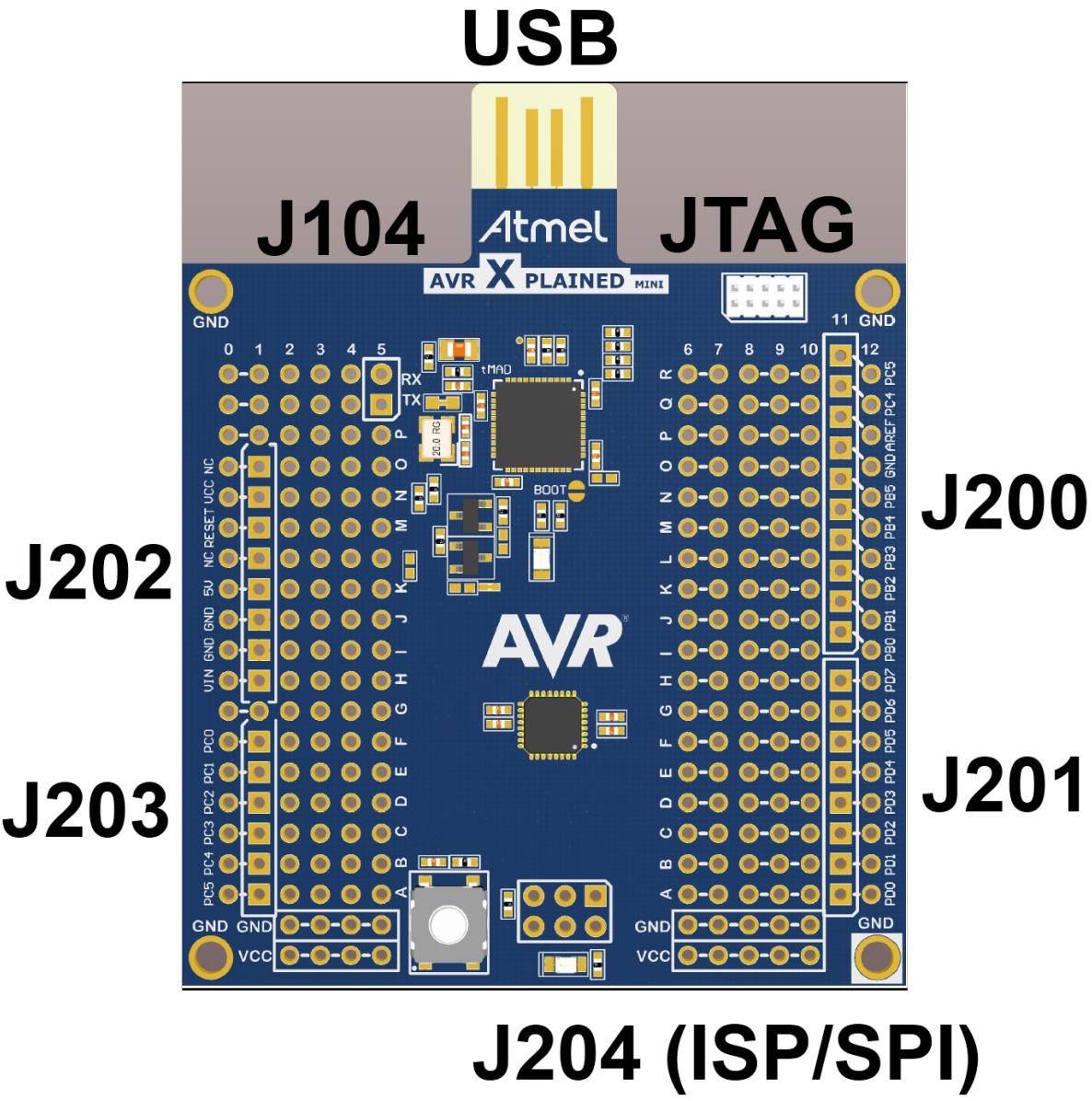
Any Morse code transmitted by using the switch will be displayed as text in the terminal window.

[4] http://spaces.atmel.com/gf/

2. Hardware User Guide

2.1 Board Overview

Figure 2-1. ATmega168XMINIoverview.png



2.2 Clock Distribution

The ATmega32U4 (mEDBG) has an external 16MHz XTAL.

The ATmega32U4 provides an external 16MHz clock to the ATmega168 (target).

2.3 Headers and Connectors

The board connectors.

2.3.1 JTAG (J100)

J100 is the JTAG programming header typically used by the JTAGICE for programming of the ATmega32U4 (mEDBG).

Table 2-1. JTAG

J100 pin	Signal function
1	JTAG_TCK

J100 pin	Signal function
2	GND
3	JTAG_TDO
4	VCC (5V0)
5	JTAG_TMS
6	RESET
7	NC
8	NC
9	JTAG_TDI
10	GND

2.3.2 USB (J101)

J101 is a USB2.0 A PCB connector connecting the board to the PC.

Table 2-2. USB

J101 pin	Function
1	VBUS
2	D-
3	D+
4	GND

2.3.3 USART (J102)

The ATmega32U4 UASRT signals are available on J102.

Table 2-3. J102 USART

J102 pin	ATmega32U4	ATmega168	Function
1 - UART TXD	PD3	PD1	TxD from ATmega32U4
2 - UART RXD	PD2	PD0	TxD from ATmega168

2.3.4 ATmega168 Digital I/O (J200 and J201)

J200 and J201 provide access to ATmega168 digital I/O pins.

Table 2-4. J200

J200 pin & location	ATmega168 pin
J200-1/ I11	PB0
J200-2	PB1
J200-3	PB2
J200-4	PB3
J200-5	PB4
J200-6	PB5
J200-7	GND
J200-8	AREF
J200-9	PC4
J200-10	PC5

Table 2-5. J201

J201 pin	ATmega168 pin
J201-1	PD0

J201 pin	ATmega168 pin
J201-2	PD1
J201-3	PD2
J201-4	PD3
J201-5	PD4
J201-6	PD5
J201-7	PD6
J201-8	PD7

2.3.5 ATmega168 Analogue I/O (J203)

The ATmega168 analogue I/O pins are available in J203.

Table 2-6. J200

J200 pin & location	ATmega168 pin
J200-1/ I11	PC0
J200-2	PC1
J200-3	PC2
J200-4	PC3
J200-5	PC4
J200-6	PC5

2.3.6 Power (J202)

The ATmega168 VCC and RESET is available in J202.

Table 2-7. J100 Xplained Pro Extension Header

J202 pin	Description
1	NC.
2	ATmega168 Vcc
3	RESET
4	NC
5	ATmega168 Vcc.
6	GND
7	GND
8	ATmega168 Vcc

2.3.7 ATmega168 ISP (J204)

J204 enable direct connection to ISP for programming of the ATmega168 or to use the SPI bus to connect external equipment.

Table 2-8. J204

J204 pin	Function
1	MISO
2	VCC target (ATmega168)
3	SCK
4	MOSI
5	RESET
6	GND

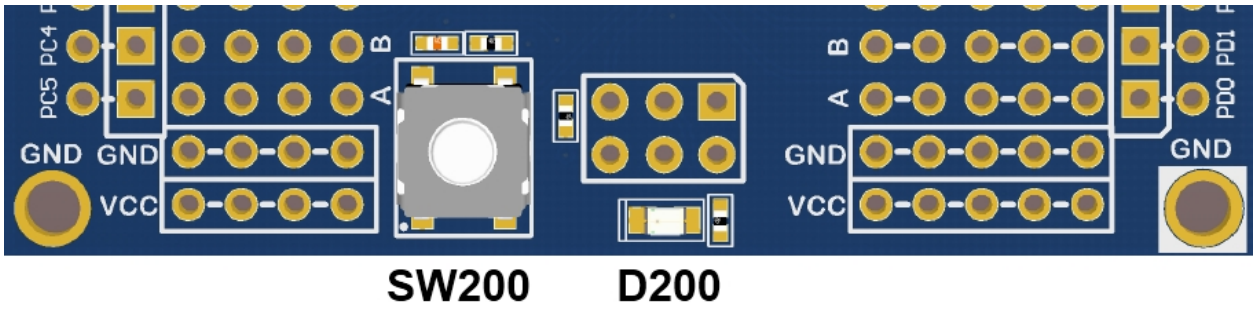
2.4 Board GUI

2.4.1 LEDs

There are one LED available for use by application SW and one for the mEDBG.

Table 2-9. LEDs

LED	Function
D100 - Green	mEDBG, will light during enumeration.
D200 - Yellow	ATmega168 pin 17 - PB5, also connected to mEDBG SCK for ISP programming, in 3-state when not used.



2.4.2 Button

A button is available for general use by application SW.

Table 2-10. Button

Button	Function	ATmega168 pin
SW200	User defined high signal, press to ground (negate).	8 - PB7

2.5 Factory Programmed Data

The ATmega168 Xplained Mini board comes with a demo program preprogrammed in the ATmega168 FLASH using the external clock provided by the ATmega32U4.

The ATmega32U4 is preprogrammed with the mEDBG.

3. Document Revision History

Document revision	Date	Comment
42250A	01/2014	Initial document release

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | www.atmel.com