APPLICATION NOTE

AT12615: Getting Started with ATmega168PB

Atmel AVR 8-bit Microcontrollers

Objective

- Get started with the ATmega168PB (Atmel® megaAVR® series device).
- Find the ATmega168PB device related links in the Atmel website (like documents, tools etc).
- Get started with Atmel Studio 6.2 (project creation and demo program (GPIO input / output configuration) development on a step by step basis). All these steps are also described in Atmel Studio help link.

Pre-requisites

- Atmel ATmega168PB Xplained Mini Kit
- Atmel Studio 6.2 or later

Table of Contents

Objective 1

Table of Contents 2

1 Getting Started with ATmega168PB 3

1.1 ATmega168PB Key Features 3
1.2 ATmega168PB Block Diagram 3
1.3 ATmega168PB Device Related Website Links 4
1.4 ATmega168PB Xplained Mini Kit 6
1.5 Atmel Studio 7
1.5.1 Atmel Studio Webpage 7
1.5.2 Atmel Studio Microsite 7
1.6 Connecting the ATmega168PB Xplained Mini kit 8
1.6.1 Auto board identification of ATmega168PB Xplained Mini Kit 8
1.6.2 Connect the ATmega168PB Xplained Mini UART to the mEDBG COM Port 9
1.7 Programming and Debugging 9
1.7.1 Programming the ATmega168PB Xplained Mini using mEDBG 9
1.7.2 Debugging the ATmega168PB Xplained Mini using mEDBG 9

2 Creating an Example Application in Atmel Studio 10

3 What's Next? 14

4 Revision History 15

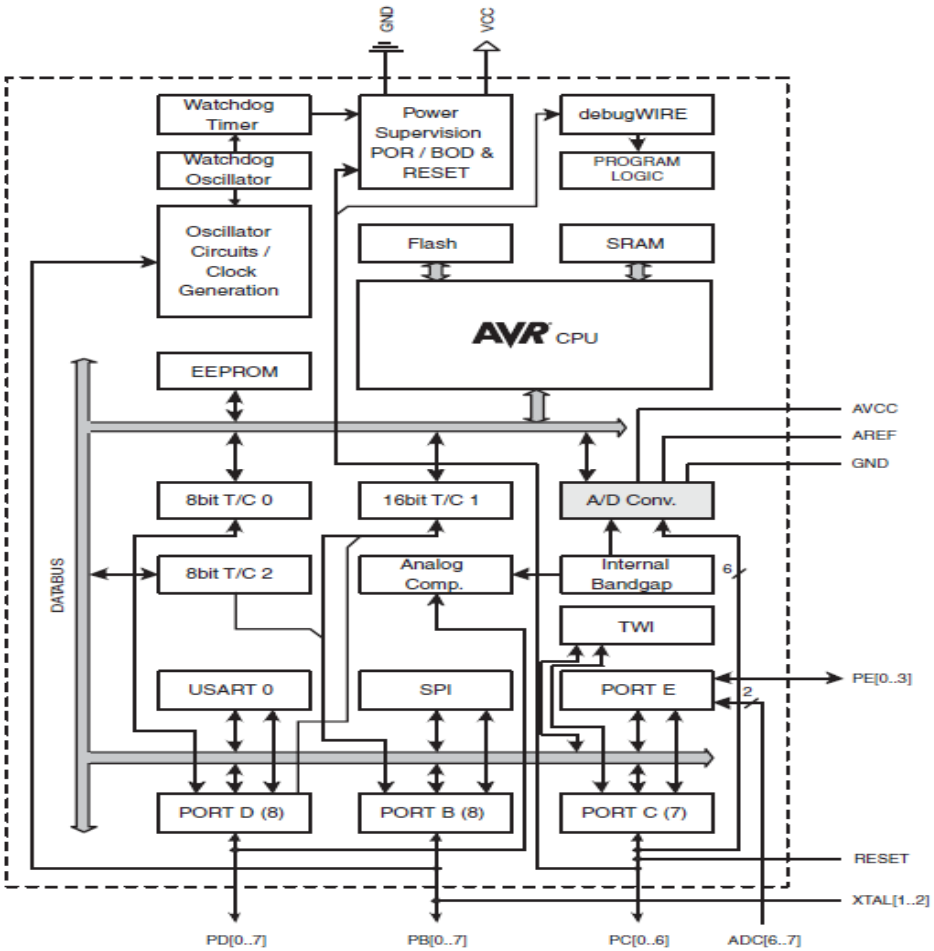
1 Getting Started with ATmega168PB

1.1 ATmega168PB Key Features

- AVR (Harvard) Architecture
- 20 MIPS @ 20 MHz
- High Code Density (Advanced RISC Instruction Set)
- On-Chip Hardware Multiplier
- Short Interrupt Latency 4 Clock Cycles
- Factory Calibrated Internal RC Oscillator
- Security with Fuses and Lock Bits
- Atmel QTouch® Library support
- Compatibility between devices (Portability)

Note: For detailed information (like flash size, number of pins, operating voltage range, number of peripheral channels, module description etc) refer to the ATmega168PB datasheet.

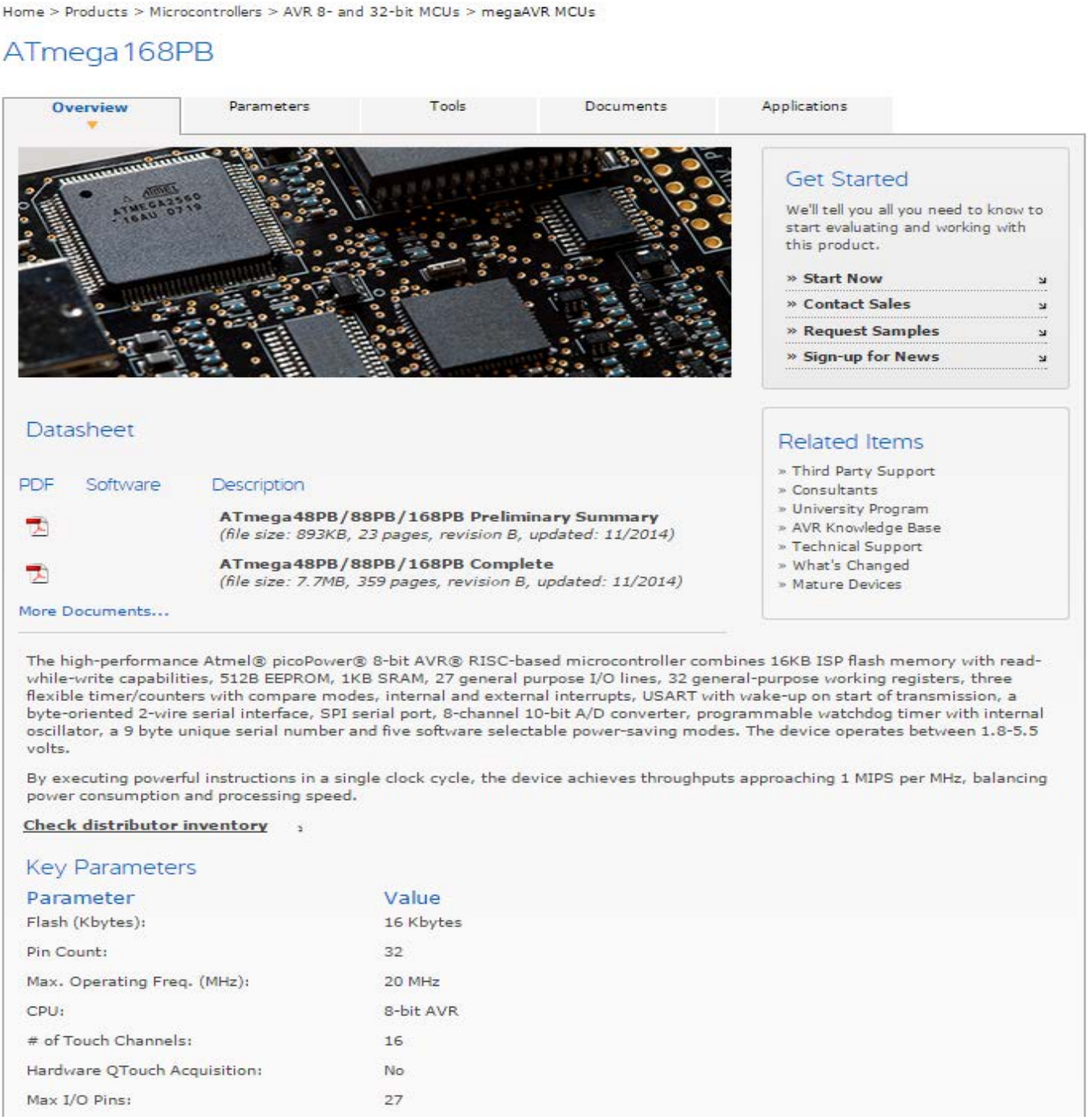
1.2 ATmega168PB Block Diagram



Note: For detailed information (like module description) refer to the ATmega168PB datasheet.

1.3 ATmega168PB Device Related Website Links

The ATmega168PB product overview webpage (as shown below) is available at the following link
http://www.atmel.com/devices/ATMEGA168PB.aspx



In ‘Parameters’ tab, we can find the configuration parameter details (like flash size, number of peripheral channels etc) for this device.

http://www.atmel.com/devices/ATMEGA168PB.aspx?tab=parameters

In ‘Documents’ tab, we can find all the related documents (like datasheet, application notes) for this device

http://www.atmel.com/devices/ATMEGA168PB.aspx?tab=documents

In the above link under ‘datasheet’ section there are two documents

- Complete version (includes all peripheral descriptions and electrical characteristics)
- Summary version (includes Ordering Information, pin out, and Packaging Information)

The device related application notes (like hardware design considerations) and its associated firmware (if any) is also available in the above specified link.

In ‘Applications’ tab, we can find the recommended application areas for (not limited to) for this device

http://www.atmel.com/devices/ATMEGA168PB.aspx?tab=applications

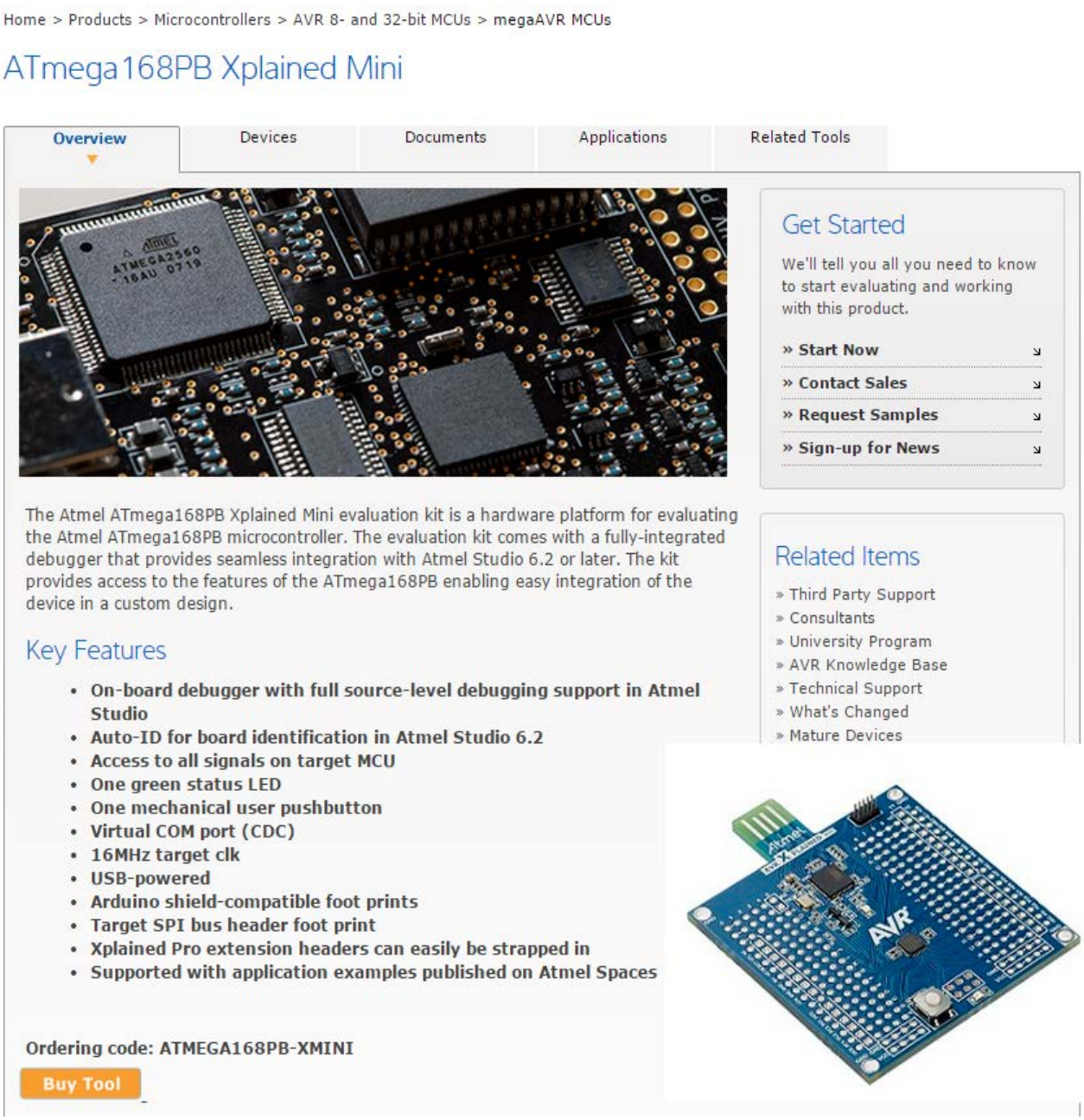
In ‘Tools’ tab, we can find all the related tools (like IDE, programmer, debugger, evaluation kits, BSDL files) for this device

http://www.atmel.com/devices/ATMEGA168PB.aspx?tab=tools

If we click ‘ATmega168PB Xplained Mini’ from the above link then we can view the ATmega168PB Xplained Mini kit’s overview webpage (as shown below)

http://www.atmel.com/tools/MEGA168PB-XMINI.aspx?tab=overview

1.4 ATmega168PB Xplained Mini Kit



If we click on 'Buy Tool' from the above link then we can purchase the same kit from our Atmel Store

http://store.atmel.com/PartDetail.aspx?q=p:10500404

In 'documents' tab from the above link we can find the kit related documents like (schematic, user guide).

http://www.atmel.com/tools/MEGA168PB-XMINI.aspx?tab=documents

Note: For detailed information like header and connections refer to the ATmega168PB Xplained Mini User Guide which is available at the above specified link.

1.5 Atmel Studio

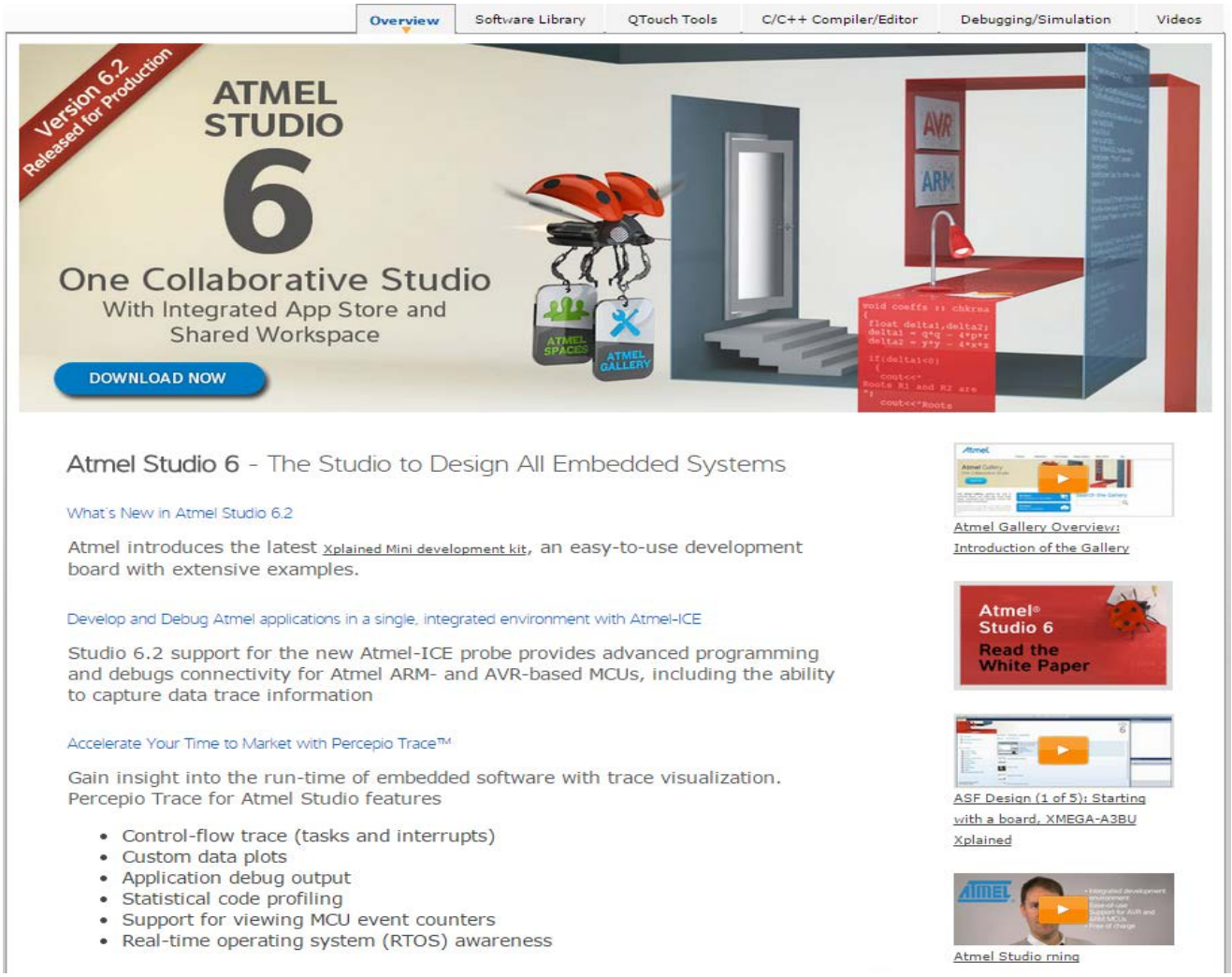
1.5.1 Atmel Studio Webpage

The Atmel Studio installer (free IDP) is available at the following (as shown below) link

http://www.atmel.com/tools/ATMELSTUDIO.aspx

1.5.2 Atmel Studio Microsite

To learn more about Atmel Studio refer to the following micro site link
http://www.atmel.com/microsite/atmel_studio6/



In 'Videos' tab we can find the getting started videos (like editor, creating a new C (gcc) project, debugging AVR applications, debugging ARM® applications etc)

http://www.atmel.com/microsite/atmel_studio6/videos.aspx

1.6 Connecting the ATmega168PB Xplained Mini kit

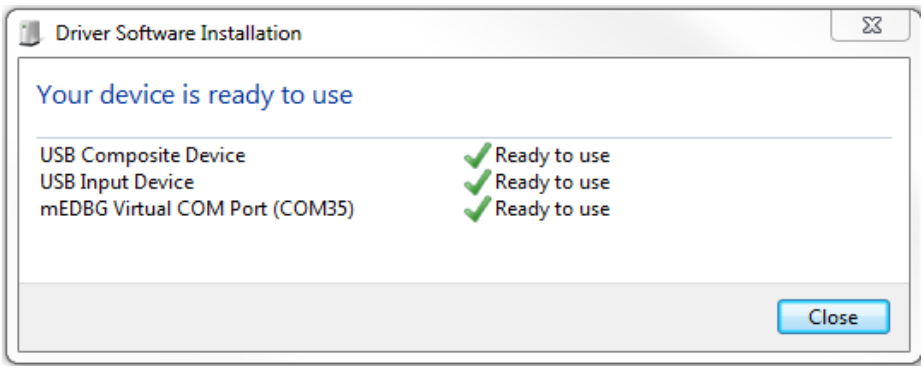
This section helps user to connect the ATmega168PB Xplained Mini with Atmel Studio 6.2

1. Download and install Atmel Studio version 6.2 or later.
2. Launch Atmel Studio.
3. Connect the ATmega168PB Xplained Mini to the USB port and it will be visible in Atmel Studio

1.6.1 Auto board identification of ATmega168PB Xplained Mini Kit

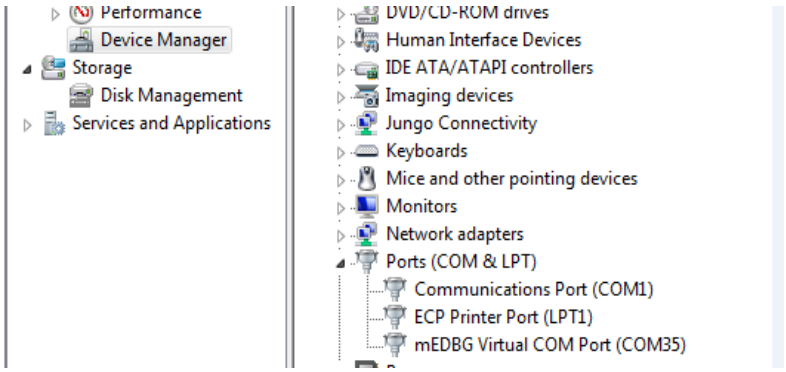
1. Once the ATmega168PB Xplained Mini kit is connected to the PC, the Windows® Task bar will pop-up a message as shown in figure.

Figure: ATmega168PB Xplained Mini Driver Installation



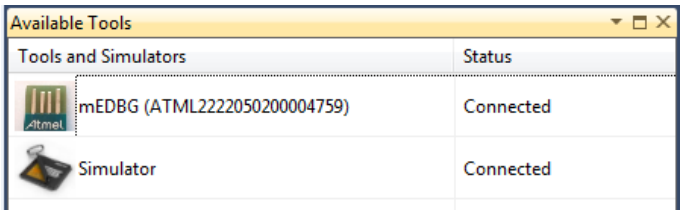
2. If the driver installation is proper, EDBG will be listed in the Device Manager as shown in figure.

Figure: Successful mEDBG Driver Installation



3. Open Atmel Studio 6.2, Go to ‘View’ -> ‘Available Atmel Tools’. The EDBG should get listed in the tools as "EDBG" and the tool status should display as "Connected". This indicates that the tool is communicating properly with Atmel Studio.

Figure: mEDBG under Available Atmel Tools



1.6.2 Connect the ATmega168PB Xplained Mini UART to the mEDBG COM Port

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 9600 baud N 8 1. The COM port settings can be changed using the Device Manager.

1.7 Programming and Debugging

This section helps to program and debug the ATmega168PB Xplained Mini kit using mEDBG.

1.7.1 Programming the ATmega168PB Xplained Mini using mEDBG

1. Connect the mEDBG USB to the PC.
2. Go to Atmel Studio: Click Tools, select Device Programming, and select the connected mEDBG as Tool with Device = ATmega168PB and Interface = ISP, click Apply.
3. Select "Memories" and locate the source hex or elf file and then click Program.
4. If the source contains fuse settings go to "Production file" and upload the elf file and program the fuses.

Note: If ISP programming fails it could be because debugWIRE is enabled. See debugging section on how to disable debugWIRE mode.

1.7.2 Debugging the ATmega168PB Xplained Mini using mEDBG

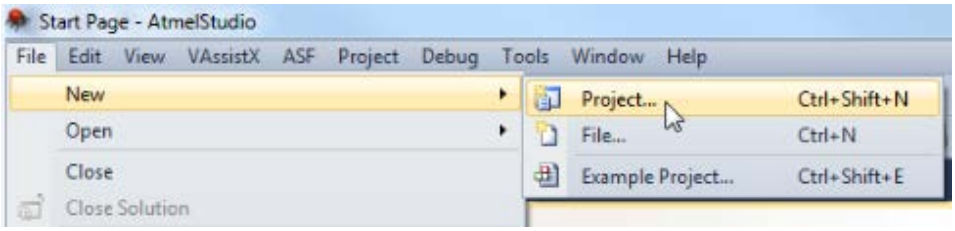
1. Start Atmel Studio.
2. Connect the mEDBG USB to the PC.
3. Open your project.
4. In the Project menu select the project properties page, select the Tools tab and select mEDBG as debugger and debugWIRE as interface.
5. In the Debug menu click Start Debugging and Break.
6. Atmel Studio will display an error message if the DWEN fuse in the ATmega168PB 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.

Note: If not exiting debug mode by selecting "Disable debugWIRE and Close" in the Debug menu, the DWEN fuse will be enabled and the target will still be in debug mode, i.e. it will not be possible to program the target using the SPI (ISP) interface.

2 Creating an Example Application in Atmel Studio

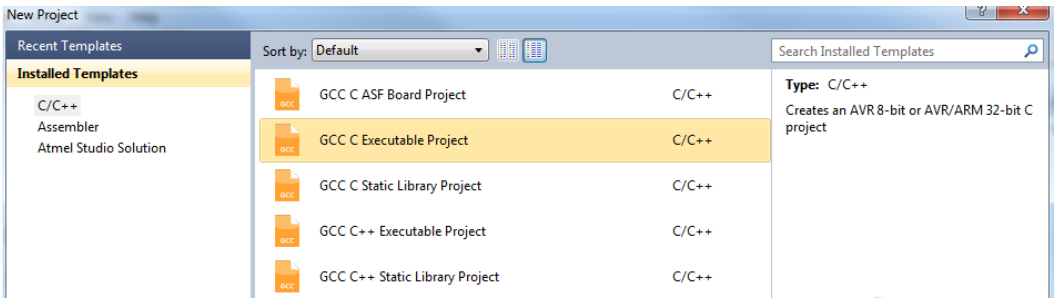
1. After connecting the board, to create a new project in Atmel Studio go to ‘File’ -> ‘New’ and click on ‘Project…’. (as shown in figure 2-1)

Figure 2-1. Creating New Project in Atmel Studio



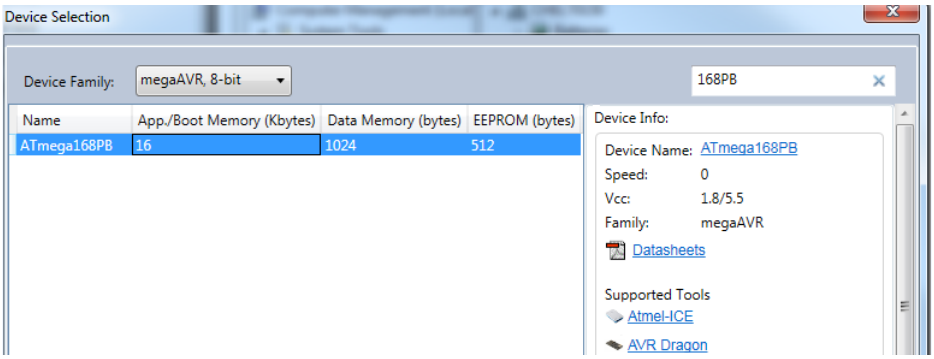
2. The New Project wizard will display as shown in figure 2-2 and select “GCC C executable Project” template and name the project and click “OK” to get the device selection wizard.

Figure 2-2. New Project Wizard



3. The Device Selection wizard will display as shown in figure 2-3 and select the ATmega168PB device from megaAVR device family then click OK.

Figure 2-3. Device Selection Wizard



4. The new project and .c file will be created as shown in figure 2.4.

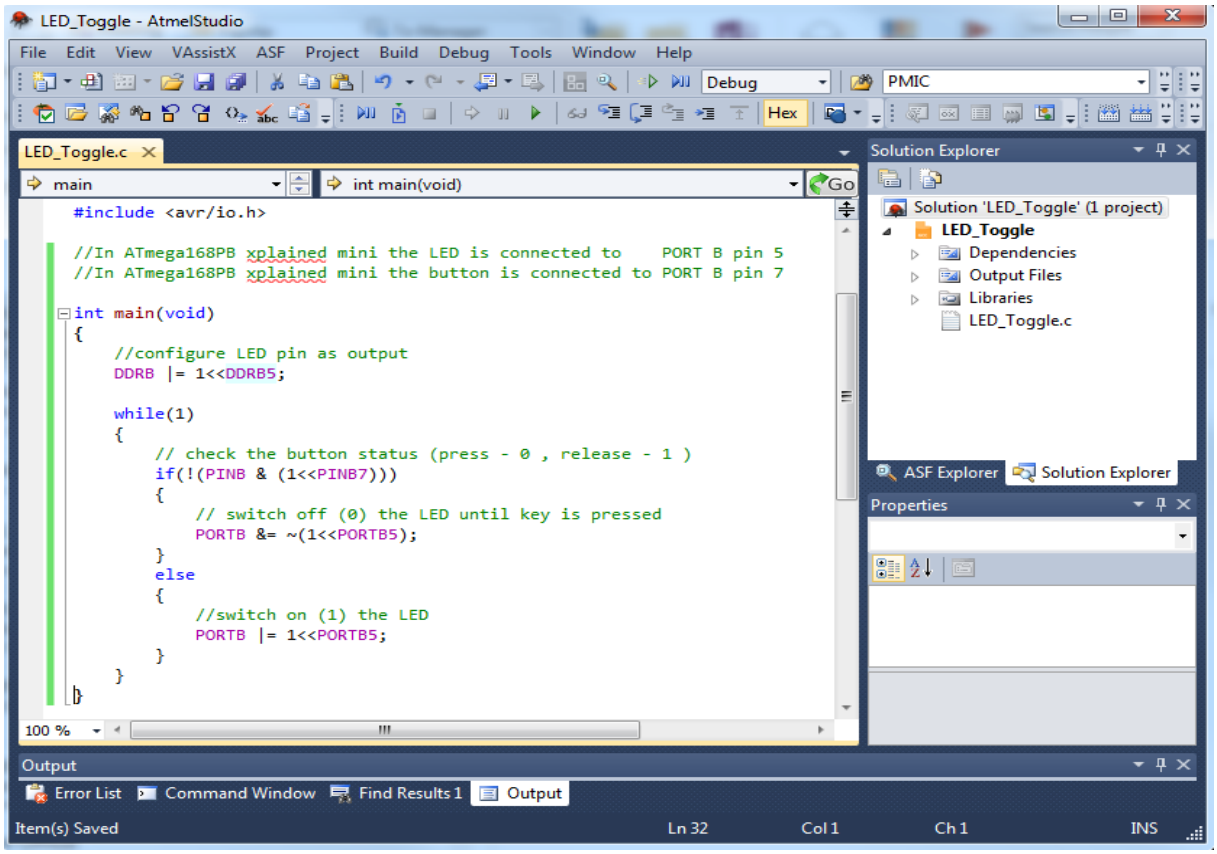
Add the following code snippet (LED control using push button) in .c file.

```
int main(void)
{
	//configure LED pin as output
	DDRB |= 1<<DDRB5;

	while(1)
	{
		/* check the button status (press - 0 , release - 1 ) */
		if(!(PINB & (1<<PINB7)))
		{
			/* switch off (0) the LED until key is pressed */
			PORTB &= ~(1<<PORTB5);
		}

		else
		{
			/* switch on (1) the LED*/
			PORTB |= 1<<PORTB5;
		}
	}
}
```

5. Code Explanation:

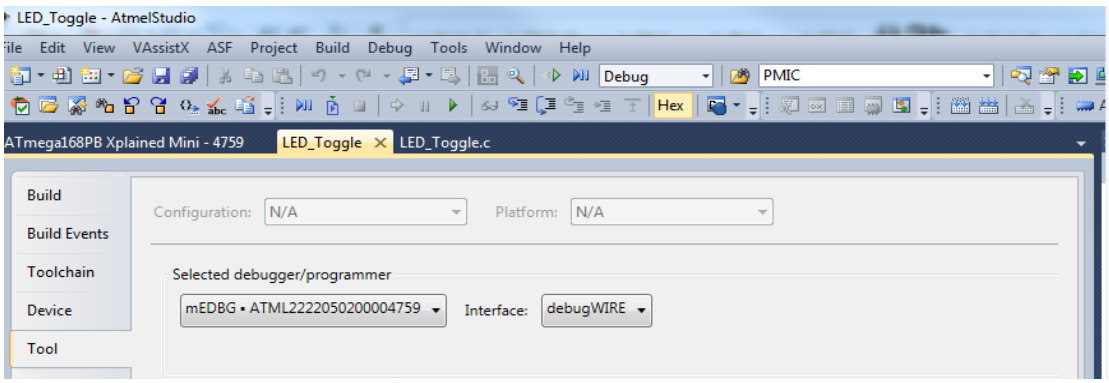
- Each PORT has three registers DDRx, PORTx, and PINx.
- The DDRx register is used to configure the port pin direction. 1 - Output ; 0 - Input.
- If one pin is configured as output pin and if the respective bit in PORTx is written logic one then the respective port pin is driven high. If the same bit is written logic zero then the pin will be driven low.
- The PINx register is used to return the logic level available on port pin.
- In the above example code we use PB7 Button as input and PB5 LED0 as output.
- We are controlling the LED0 based on the push button status.
- As long the button is in pressed state (0) then the LED0 will not glow (0).
- If we release the button (1) then the LED0 will glow (1 - default).

Figure 2-4. Project Window



6. In order to debug this project, configure the Tool and Interface in the Project properties. To open the project properties, go to ‘Project’ menu -> ‘Properties’. In the project properties, go to "Tool" tab -> Under the Selected Debugger/Programmer, select the tool as "mEDBG" and interface as "debugWIRE" as shown in figure 2-5 Tool and Interface Settings.

Figure 2-5. Tool and Interface Settings



7. To program and execute the application, we have two options.

    a. We can start a debug session on the board, where we will be able to program and debug.
    b. We can program the generated hex file into the controller and execute the application.

    In this case we will program the code with no debugging, so we select the green arrow for "Start without Debugging".

    Both these options can be done on ATmega168PB Xplained Mini as shown in figure 2-6 and figure 2-7.

Note: If not exiting debug mode by selecting "Disable debugWIRE and Close" in the Debug menu, the DWEN fuse will be enabled and the target will still be in debug mode, i.e. it will not be possible to program the target using the SPI.

Figure 2-6. Start without Debugging

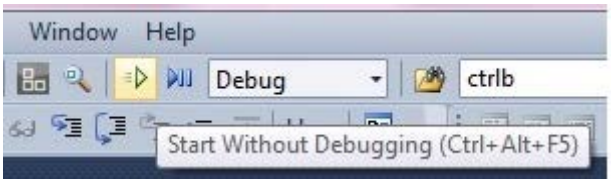


Figure 2-7. Start Debugging and Break

3 What's Next?

- Atmel Studio videos
  http://www.atmel.com/microsite/atmel_studio6/videos.aspx

- Atmel Studio online help
  http://www.atmel.com/webdoc/atmelstudio/

- Atmel Studio offline help (After installing Atmel Studio)

  In Atmel Studio
  Help -> View Help (Ctrl+F1) -> Atmel Studio

- ASF (Atmel Software framework) Getting Started and ASF Reference manual
  http://www.atmel.com/tools/AVRSOFTWAREFRAMEWORK.aspx?tab=documents

- ASF online documentation
  http://asf.atmel.com/docs/latest/

- Technical documentation for various products
  http://www.atmel.com/webdoc/

- Atmel Gallery
  https://gallery.atmel.com/

- Production Selection Guide
  Atmel MCU Selector on http://www.atmel.com/

- Ordering Samples
  https://secure.atmel.com/forms/secure/verifylogin.aspx?target_url=samplestore

- Buying evaluation board & kits
  http://store.atmel.com/

- Technical Documentation
  http://www.atmel.com/design-support/documentation/default.aspx

- Knowledge Base and Technical Support/Design Support
  http://www.atmel.com/design-support/

- Collaborative workspace
  http://spaces.atmel.com

- AVR Freaks community
  http://www.avrfreaks.net/

4 Revision History

Doc. Rev.	Date	Comments
42422A	03/2015	Initial document release

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

© 2015 Atmel Corporation. / Rev.:Atmel-42422A-MCU-Getting-Started-with-ATmega168PB-Application-Note_032015.

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, AVR®, megaAVR® and others are registered trademarks or trademarks of Atmel Corporation in U.S. and other countries. ARM® is a registered trademark of ARM Ltd. Windows® is a registered trademark of Microsoft Corporation in U.S. and or other countries. Other terms and product names may be trademarks of others.

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.