Hardware Development Kit

The Atmel® Hardware Development Kit (HDK) provides all necessary information for a developer to make hardware that is compatible with Atmel Xplained Pro products, integrate it with Atmel Studio as an extension and add example firmware to Atmel Software Framework (ASF).
# Table of Contents

Hardware Development Kit ................................................................. 1

1. Introduction .............................................................................. 3
   1.1. Compatible Xplained Pro hardware ..................................... 3
   1.2. Studio integration ............................................................ 3
   1.3. Example code .................................................................. 4

2. How to Design a Xplained Pro Extension ............................... 5
   2.1. Xplained Pro ID system .................................................... 5
      2.1.1. Overview ............................................................. 5
      2.1.2. ID system implementation on extensions ....................... 5
      2.1.3. ID device data ...................................................... 6
      2.1.4. Data encoding ....................................................... 6
      2.1.5. Creating your own ID data ....................................... 7
      2.1.6. Programming the ID device ...................................... 7
   2.2. Xplained Pro extension naming convention ............................. 8
      2.2.1. Product name ....................................................... 8
      2.2.2. Silkscreen text ...................................................... 8
   2.3. Xplained Pro connectors ...................................................... 9
      2.3.1. Xplained Pro extension header .................................... 9
      2.3.2. Xplained Pro power header ..................................... 10
      2.3.3. Xplained Pro segment LCD extension connector ............ 10
      2.3.4. Xplained Pro LCD connector ................................... 11
   2.4. Power specifications ........................................................ 12
   2.5. Extension board templates ............................................... 13
      2.5.1. Designing a board with the standard extension connector 13
      2.5.2. Designing a board with the segment LCD connector ....... 25
      2.5.3. Designing a board with the LCD connector ................. 26

3. How to Integrate a Xplained Pro Extension in Atmel Studio .................................................... 27
   3.1. Xplained Pro Landing Page ................................................ 27

4. How to Integrate Code Examples for a Xplained Pro Extension in Atmel Studio ............................. 28

A. Appendix ................................................................................ 29
   A.1. Document Revision History .............................................. 29
   A.2. xpro_id Version History .................................................. 29
      A.2.1. Version 0.5 .......................................................... 29
1. Introduction

The Hardware Development Kit (HDK) describes how to integrate a Xplained Pro hardware design seamlessly into the Atmel tools and software offering. Three requirements must be fulfilled in order to accomplish this task and these are:

1. Compatible hardware
2. Atmel Studio integration
3. Example code

When all these requirements are fulfilled a good user experience is achieved because each step in the evaluation process is covered and the user has easy access to everything he needs.

1.1 Compatible Xplained Pro hardware

Xplained Pro is an evaluation platform that provides the full Atmel microcontroller experience. The platform consists of a series of Microcontroller (MCU) boards and extension boards that are integrated with Atmel Studio, have Atmel Software Framework (ASF) drivers and demo code, support data streaming and more. Xplained Pro MCU boards support a wide range of Xplained Pro extension boards that are connected through a set of standardized headers and connectors. Each extension board has an identification (ID) chip to uniquely identify which boards are mounted on a Xplained Pro MCU board.

All Xplained Pro MCU boards feature an Atmel Embedded Debugger (EDBG). The EDBG supports programming and debugging of the target MCU on the Xplained Pro MCU board and works as a gateway for a hardware identification system that lets Atmel Studio detect connected hardware. The hardware identification system covers identification of Xplained Pro MCU boards and identification of connected extension board.

This document has the information required to implement a Xplained Pro extension board with the hardware identification system.

Figure 1-1. Typical Xplained Pro kit overview

1.2 Studio integration

When Atmel Studio detects Xplained Pro compatible hardware it will search for a landing page for it and present it to the user. The landing page contains:

- Short description of the kit
- Picture of the kit
- Links to kit documentation
- Links to relevant datasheets
● Link that opens a list with relevant applications for this kit (filtered ASF examples list)
● Link to Atmel store or other places where the kit can be bought

Other information on the landing page is obtained directly from the connected hardware via the kit identification system e.g. revision, capabilities, serial number etc.
If no landing page is found the user will be requested to update the Atmel kits extension or to check for required additional extensions in the Atmel gallery.

1.3 Example code

The final step of the integration is addition of example code for the hardware and this is described in detail in the Software Development Kit (SDK).
2. How to Design a Xplained Pro Extension

2.1 Xplained Pro ID system

Identification of extension boards for the Xplained Pro platform is required in order to leverage the ease of use for Atmel products. The intention of the identification is not to protect the hardware from being copied.

Identified expansion boards are reported via the Embedded Debugger to the host PC software, which is Atmel Studio. Based on the detected hardware Atmel Studio will then provide additional information to the user such as:

- Link to user guides and relevant datasheets
- Available Atmel Software Framework (ASF) applications for the extension
- Extension revision and features

This chapter is important for all developers that want to implement the ID system in a design e.g. on extensions for Xplained Pro.

2.1.1 Overview

The Embedded Debugger (EDBG) is the central part in the overall system because it serves as a gateway between the hardware and the host PC software. The system block diagram in Figure 2-1, “ID system overview” shows how the main components of the system and how they connect to each other. Each extension connector on a Xplained MCU board has a unique ID channel which is connected to the EDBG and to an ID device on a connected extension board. Upon power up the EDBG will check all ID channels for ID devices, read out the product information and store this internally. Once a connection to the host PC software is established this information can be retrieved and presented to the user.

Figure 2-1. ID system overview

2.1.2 ID system implementation on extensions

The ID device that must be mounted on Xplained Pro extensions is the Atmel ATSHA204 in a single-wire configuration where the device is powered through the communication line. On the Atmel Xplained Pro extensions the device with the ordering code ATSHA204-TSU-T is used. Relevant features of the device are:

- Operation voltage from 2.0V to 5.5V
- Single wire interface
- 3-lead SOT23 (one wire)
- Data area with 512 bytes
- Configuration area with 88 bytes
- One time programmable (OTP) area with 64 bytes
The example in Figure 2-2, “ID device circuitry” shows the implemented ID circuitry on the Xplained Pro extension boards. The ID_DATA signal is routed to the Embedded Debugger where this signal is pulled-up. The ID chip is powered through the ID_DATA line via D100. R100 acts as a bleeding resistor to discharge C100 when the extension is unplugged, this is necessary in order to get the ID device in a safe state within a reasonable time before the board is plugged in again. The ID_DATA line is connected to a dedicated pin on the extension header your Xplained Pro board implements.

Figure 2-2. ID device circuitry

2.1.3 ID device data
The following data must be programmed into the ID device so that the most vital information can be presented to the user in Atmel Studio.

1. Manufacturer name
2. Product name
3. Product revision
4. Product serial number
5. Minimum supported voltage for the extension board
6. Maximum supported voltage for the extension board
7. Minimum current that is required to support the extension board

The product name is the key for a lookup in the available kits list in Studio and it is therefore vital that this information is unique and always present. Additional differentiation e.g. based on the manufacturer name is possible but currently not implemented. If a kit name cannot be resolved in Studio it will be suggested to the user that he should update or install the required extension and he can do this based on the available information that is read out from the ID device.

This data is placed in the OTP zone which means once it is programmed into the ID device memory it can’t be erased.

2.1.4 Data encoding
The data in the ATSHA204 is encoded in the following way. Manufacturer name, product name, product revision and serial number are stored as 0 terminated ASCII strings. This allows all the strings to have variable length. Minimum voltage, maximum voltage and required current are stored as unsigned 16 bit integer values at the last 6 bytes of the OTP memory zone. The byte ordering is big endian.

It is required to know the entire content of the OTP zone before locking it. All unused bytes in the OTP memory have to be written to a known value. All unused area of the OTP memory, meaning all bytes between the last ASCII string (terminated with the ‘\0’ character) and the 6 bytes for the max/min values is filled with 0xFF. These bytes are marked as DUMMY BYTES in the example table below.
It is also required to know the entire content of the data memory prior to locking the OTP zone, thus the entire data memory is filled with 0x00. The data zones are not locked for writing so it is possible, if desirable, to write updated information about the kit in the data memory. The table below shows an example of a preprogrammed memory for a fictional extension board called "Sensor Xplained".

Table 2-1. Exampled content for the ID device

<table>
<thead>
<tr>
<th>Data field</th>
<th>Example content</th>
<th>Data type</th>
<th>Byte position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Atmel'0'</td>
<td>ASCII string</td>
<td>OTP[0:5]</td>
</tr>
<tr>
<td>Product name</td>
<td>Sensor Xplained'0'</td>
<td>ASCII string</td>
<td>OTP[6:21]</td>
</tr>
<tr>
<td>Product revision</td>
<td>01'0'</td>
<td>ASCII string</td>
<td>OTP[22:24]</td>
</tr>
<tr>
<td>Product serial number</td>
<td>0200000002'0'</td>
<td>ASCII string</td>
<td>OTP[25:35]</td>
</tr>
<tr>
<td>DUMMY BYTES</td>
<td>0xFF, 0xFF, 0xFF...</td>
<td>Byte</td>
<td>OTP[36:57]</td>
</tr>
<tr>
<td>Minimum Voltage [mV]</td>
<td>1600</td>
<td>Unsigned 16-bit integer</td>
<td>OTP[58:59]</td>
</tr>
<tr>
<td>Maximum Voltage [mV]</td>
<td>3300</td>
<td>Unsigned 16-bit integer</td>
<td>OTP[60:61]</td>
</tr>
<tr>
<td>Required Current [mA]</td>
<td>50</td>
<td>Unsigned 16-bit integer</td>
<td>OTP[62:63]</td>
</tr>
</tbody>
</table>

Note: All ASCII strings are terminated with the value 0x00 (\'0\')

Note: 4 bytes are used for string terminations (\'0\'), 6 bytes are used for max/min values storage. That leaves 54 bytes for ASCII characters. This means that the combination of manufacturer, product name, revision and serial number cannot exceed 54 characters.

Note: The Minimum and Maximum voltage parameters may be used if the Xplained Pro boards get variable target voltage and switching of power (VCC) to the Extension ports. The Extension kit's voltage range can be read from the ID chip without applying power to the Extension module, and power can be switched on if target voltage is within the valid voltage range of the Extension module.

2.1.5 Creating your own ID data

All extensions must have a unique product name and manufacturer so that they can be associated with available documentation and firmware in Atmel Studio in the future. This means all products must be registered so that the uniqueness of the name is ensured. To register an Xplained Pro extension module id send an e-mail to edbg@atmel.com with the manufacturer name and product name.

2.1.6 Programming the ID device

The ID device can be programmed via the Embedded Debugger that is mounted on the Xplained Pro MCU boards. That means all Xplained Pro MCU boards can act as a programmer for the ID device by connecting one of the ID signals.

Atmel provides a Python module for reading and programming ID devices called xpro_id. The module is tested with python 2.7.4. The latest installer for the xpro_id package can be downloaded from the Atmel Gallery developer page.

The best way to read the documentation for the xpro_id Python module is to use pydoc.py to create a locally hosted web page. Running the following command in a command prompt will host the documentation at http://localhost:1234/xpro_id.html.

C:\Python27\python.exe C:\Python27\Lib\pydoc.py -p 1234

1 mailto:edbg@atmel.com
2 http://www.python.org/
3 The module should be compatible with other 2.x.x versions of Python as long as ctypes is installed.
4 http://gallery.atmel.com/Partner
5 Assumes that Python 2.7.x is used and installed in the default location.
The Python package is split in two classes; edbg_hid and xpro_id. edbg_hid interfaces cmsis_dap.dll to communicate with the embedded debugger and provides the required functions to read and program Xplained Pro ID devices. xpro_id provides an example of interfacing the edbg_hid class.

**Note**  
The xpro_id class is provided as an example on how to interface the edbg_hid class. The code may be altered to fit a specific manufacturing setup.

**Tip**  
Atmel uses a more direct interface to the edbg_hid class, with less checks and test on input data, in mass production of kits. Several of the input parameters are also stored in configuration files for each kit.

Questions or issues regarding Xplained Pro ID programming can be directed to edbg@atmel.com 

Version changelog for the xpro_id module is listed in “xpro_id Version History” on page 29.

### 2.2 Xplained Pro extension naming convention

##### 2.2.1 Product name

All boards of the product family are named based on the following scheme:

**[device/technology] Xplained Pro**

In addition it is possible to extend the name with a sub-part that is used to differentiate products within a product line.

- Sensors Xplained Pro Inertial
- Sensors Xplained Pro Pressure
- Security Xplained Pro Authentication

When several extensions exist with the same name and sub-naming, these can be distinguished by adding a number:

- Sensors Xplained Pro Inertial One
- OLED1 Xplained Pro
- IO1 Xplained Pro

##### 2.2.2 Silkscreen text

The board name on the PCB itself is all in capital letters, where the X in XPLAINED is the double font size than the rest of the letters. The “PRO” is attached at the end with half the font size. For example 2mm height for standard text, 4mm height for the X and 1mm height for the “PRO”. The font size that was used is Verdana with a 0.5mm inverted border.

*Figure 2-3. Extension silkscreen naming example 1*
2.3 Xplained Pro connectors

2.3.1 Xplained Pro extension header

All Xplained Pro kits have one or more dual row, 20 pin, 100mil extension headers. Xplained Pro MCU boards have male headers while Xplained Pro extensions have their female counterparts. Note that all pins are not always connected. However, all the connected pins follow the defined pin-out described in Table 2-2, “Xplained Pro extension header”. The extension headers can be used to connect a wide variety of Xplained Pro extensions to Xplained Pro MCU boards and to access the pins of the target MCU on Xplained Pro MCU board directly.

Table 2-2. Xplained Pro extension header

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Communication line to the ID chip on extension board.</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>3</td>
<td>ADC(+)</td>
<td>Analog to digital converter, alternatively positive part of differential ADC</td>
</tr>
<tr>
<td>4</td>
<td>ADC(-)</td>
<td>Analog to digital converter, alternatively negative part of differential ADC</td>
</tr>
<tr>
<td>5</td>
<td>GPIO1</td>
<td>General purpose IO</td>
</tr>
<tr>
<td>6</td>
<td>GPIO2</td>
<td>General purpose IO</td>
</tr>
<tr>
<td>7</td>
<td>PWM(+)</td>
<td>Pulse width modulation, alternatively positive part of differential PWM</td>
</tr>
<tr>
<td>8</td>
<td>PWM(-)</td>
<td>Pulse width modulation, alternatively positive part of differential PWM</td>
</tr>
<tr>
<td>9</td>
<td>IRQ/GPIO</td>
<td>Interrupt request line and/or general purpose IO.</td>
</tr>
<tr>
<td>10</td>
<td>SPI_SS_B/GPIO</td>
<td>Slave select for SPI and/or general purpose IO.</td>
</tr>
<tr>
<td>11</td>
<td>TWI_SDA</td>
<td>Data line for two wire interface. Always implemented, bus type.</td>
</tr>
<tr>
<td>12</td>
<td>TWI_SCL</td>
<td>Clock line for two wire interface. Always implemented, bus type.</td>
</tr>
<tr>
<td>13</td>
<td>USART_RX</td>
<td>Receiver line of Universal Synchronous and Asynchronous serial Receiver and Transmitter</td>
</tr>
<tr>
<td>14</td>
<td>USART_TX</td>
<td>Transmitter line of Universal Synchronous and Asynchronous serial Receiver and Transmitter</td>
</tr>
<tr>
<td>15</td>
<td>SPI_SS_A</td>
<td>Slave select for SPI. Should be unique if possible.</td>
</tr>
<tr>
<td>16</td>
<td>SPI_MOSI</td>
<td>Master out slave in line of Serial peripheral interface. Always implemented, bus type</td>
</tr>
</tbody>
</table>
### 2.3.2 Xplained Pro power header

The power header can be used to connect external power to the Xplained Pro kit. The kit will automatically detect and switch to the external power if supplied. The power header can also be used as supply for external peripherals or extension boards. Care must be taken not to exceed the total current limitation of the on-board regulator for the 3.3V regulated output. To locate the current measurement header, please refer to Figure 2-5, “Typical Xplained Pro Power Connections”.

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>SPI_MISO</td>
<td>Master in slave out line of Serial peripheral interface. Always implemented, bus type</td>
</tr>
<tr>
<td>18</td>
<td>SPI_SCK</td>
<td>Clock for Serial peripheral interface. Always implemented, bus type</td>
</tr>
<tr>
<td>19</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>20</td>
<td>VCC</td>
<td>Power for extension board</td>
</tr>
</tbody>
</table>

#### Table 2-3. Power header PWR

<table>
<thead>
<tr>
<th>Pin number PWR header</th>
<th>Pin name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VEXT_P5V0</td>
<td>External 5V input</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>3</td>
<td>VCC_P5V0</td>
<td>Unregulated 5V (output, derived from one of the input sources)</td>
</tr>
<tr>
<td>4</td>
<td>VCC_P3V3</td>
<td>Regulated 3.3V (output, used as main power for the kit)</td>
</tr>
</tbody>
</table>

**Note**

If the board is powered from a battery source it is recommended to use the PWR header. If there is a power source connected to EDBG USB, the EDBG is activated and it will consume more power.

### 2.3.3 Xplained Pro segment LCD extension connector

Xplained Pro MCU boards that have a microcontroller that supports segment LCDs contain a 51-pin segment LCD extension connector. This connector is implemented with HIROSEs DF-9 series. Xplained Pro MCU boards use the male version DF9-51P-1V(69) and Xplained Pro extension boards use the female counterpart DF9-51S-1V(69). This header has a standardized pin-out as shown in Table 2-4, “Xplained Pro segment LCD extension connector definition”.

**Note**

All pins are not connected on all Xplained Pro MCU boards. How many pins that are used depend on how many segments and common terminals the target MCU supports.

#### Table 2-4. Xplained Pro segment LCD extension connector definition

<table>
<thead>
<tr>
<th>Description</th>
<th>Function</th>
<th>Pin</th>
<th>Pin</th>
<th>Function</th>
<th>Pin</th>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common terminal 3</td>
<td>COM3</td>
<td>1</td>
<td>2</td>
<td>COM2</td>
<td>3</td>
<td>4</td>
<td>Common terminal 2</td>
</tr>
<tr>
<td>Common terminal 1</td>
<td>COM1</td>
<td>3</td>
<td>4</td>
<td>COM0</td>
<td>9</td>
<td>10</td>
<td>Common terminal 0</td>
</tr>
<tr>
<td>Segment 0</td>
<td>SEG0</td>
<td>5</td>
<td>6</td>
<td>SEG1</td>
<td>7</td>
<td>8</td>
<td>Segment 1</td>
</tr>
<tr>
<td>Segment 2</td>
<td>SEG2</td>
<td>9</td>
<td>10</td>
<td>SEG3</td>
<td>11</td>
<td>12</td>
<td>Segment 3</td>
</tr>
<tr>
<td>Segment 4</td>
<td>SEG4</td>
<td>13</td>
<td>14</td>
<td>SEG5</td>
<td>15</td>
<td>16</td>
<td>Segment 5</td>
</tr>
<tr>
<td>Segment 6</td>
<td>SEG6</td>
<td>17</td>
<td>18</td>
<td>SEG7</td>
<td>19</td>
<td>20</td>
<td>Segment 7</td>
</tr>
<tr>
<td>Segment 8</td>
<td>SEG8</td>
<td>21</td>
<td>22</td>
<td>SEG9</td>
<td>23</td>
<td>24</td>
<td>Segment 9</td>
</tr>
<tr>
<td>Segment 10</td>
<td>SEG10</td>
<td>25</td>
<td>26</td>
<td>SEG11</td>
<td>27</td>
<td>28</td>
<td>Segment 11</td>
</tr>
<tr>
<td>Segment 12</td>
<td>SEG12</td>
<td>29</td>
<td>30</td>
<td>SEG13</td>
<td>31</td>
<td>32</td>
<td>Segment 13</td>
</tr>
<tr>
<td>Segment 14</td>
<td>SEG14</td>
<td>33</td>
<td>34</td>
<td>SEG15</td>
<td>35</td>
<td>36</td>
<td>Segment 15</td>
</tr>
<tr>
<td>Segment 16</td>
<td>SEG16</td>
<td>37</td>
<td>38</td>
<td>SEG17</td>
<td>39</td>
<td>40</td>
<td>Segment 17</td>
</tr>
</tbody>
</table>
### Xplained Pro LCD connector

The LCD connector provides the ability to connect to display extensions that have a parallel interface. The connector implements signals for a MCU parallel bus interface and a LCD controller interface as well as signals for a touchcontroller. The connector pin-out definition is shown in Table 2-5, "Xplained Pro LCD connector". Note that usually only one display interface is implemented, either LCD controller or the MCU bus interface. A FPC/FFC connector with 50 pins and 0.5mm pitch is used for the LCD connector. The connector (XF2M-5015-1A) from Omron is used on several designs and can be used as a reference.

#### Table 2-5. Xplained Pro LCD connector

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Name</th>
<th>RGB interface description</th>
<th>MCU interface description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Communication line to ID chip on extension board.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>D0</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>D1</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>D2</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>D3</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>D4</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>D5</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>D6</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>D7</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>D8</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>D9</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>D10</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>D11</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>D12</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>D12</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>Pin number</td>
<td>Name</td>
<td>RGB interface description</td>
<td>MCU interface description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>20</td>
<td>D14</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>D15</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>D16</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>D17</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>D18</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>D19</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>D20</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>D21</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>D22</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>D23</td>
<td>Data line</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>PCLK / CMD_DATA_SEL</td>
<td>Pixel clock</td>
<td>Command and data select. One address line of the MCU for displays where it is possible to select either the register or the data interface.</td>
</tr>
<tr>
<td>34</td>
<td>VSYNC / CS</td>
<td>Vertical synchronization</td>
<td>Chip select</td>
</tr>
<tr>
<td>35</td>
<td>HSYNC / WE</td>
<td>Horizontal synchronization</td>
<td>Write enable signal</td>
</tr>
<tr>
<td>36</td>
<td>DATA ENABLE / RE</td>
<td>Data enable signal</td>
<td>Read enable signal</td>
</tr>
<tr>
<td>37</td>
<td>SPI SCK</td>
<td>Clock for Serial peripheral interface</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>SPI MOSI</td>
<td>Master out slave in line of Serial peripheral interface</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>SPI MISO</td>
<td>Master in slave out line of Serial peripheral interface</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>SPI SS</td>
<td>Slave select for SPI. Should be unique if possible</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ENABLE</td>
<td>Display enable signal</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>TWI SDA</td>
<td>I2C data line (maxTouch)</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>TWI SCL</td>
<td>I2C clock line (maxTouch)</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>IRQ1</td>
<td>maxTouch interrupt line</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>IRQ2</td>
<td>Interrupt line for other I2C devices</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>PWM</td>
<td>Backlight control</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>RESET</td>
<td>Reset for both display and maxTouch</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>VCC</td>
<td>3.3V power supply for extension board</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>VCC</td>
<td>3.3V power supply for extension board</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
</tbody>
</table>

### 2.4 Power specifications

The Xplained Pro kit can be powered either by USB or by an external power source through the 4-pin power header, marked PWR. This connector is described in “Xplained Pro power header” on page 10. The available power sources and specifications are listed in Table 2-6, “Power sources for Xplained Pro”.

#### Table 2-6. Power sources for Xplained Pro

<table>
<thead>
<tr>
<th>Power input</th>
<th>Voltage requirements</th>
<th>Current requirements</th>
<th>Connector marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>5V +/- 2 % (+/- 100mV) for USB host operation.</td>
<td>Recommended minimum is 1A to maxTouch</td>
<td>PWR</td>
</tr>
</tbody>
</table>
### Power input

<table>
<thead>
<tr>
<th>Power Source</th>
<th>Voltage requirements</th>
<th>Current requirements</th>
<th>Connector marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power input</td>
<td>4.3 V to 5.5 V if USB host operation is not required</td>
<td>be able to provide enough current for connected USB devices and the board itself. Recommended maximum is 2A due to the input protection maximum current specification.</td>
<td></td>
</tr>
<tr>
<td>Embedded debugger USB</td>
<td>4.4V to 5.25V (according to USB spec)</td>
<td>500 mA (according to USB spec)</td>
<td>DEBUG USB</td>
</tr>
<tr>
<td>Target USB</td>
<td>4.4V to 5.25V (according to USB spec)</td>
<td>500 mA (according to USB spec)</td>
<td>TARGET USB</td>
</tr>
</tbody>
</table>

The kit will automatically detect which power sources are available and choose which one to use according to the following priority:

1. External power
2. Embedded debugger USB
3. Target USB

**Note**

External power is required when the 500mA through the USB connector is not enough to power a connected USB device in a USB host application.

As of now all Xplained Pro MCU kits are powered by 5.0V input that is regulated to a 3.3V power supply for the EDBG, MCU and extension headers/connectors.

In the future new Xplained Pro MCU kits will get support for variable voltages to supply the target MCU and the extension headers/connectors.

**Figure 2-5. Typical Xplained Pro Power Connections**

---

### 2.5 Extension board templates

Extensions enable demonstration of MCU features that are not placed on the MCU board, e.g. sensors, displays, LEDs and push buttons. Various standard and kit specific extensions are presented in the following sections.

A future release of this document will include Altium templates for Xplained Pro extension boards.

#### 2.5.1 Designing a board with the standard extension connector

The following sections contain information about standard extension sizes used by Atmel. The holes referred to as "test jig holes" are used by Atmel during manufacturing to align the board in a test fixture, these holes can
safely be removed if they are not needed. The circles shown in the component placement circles are rubber feet placed on the bottom side of the boards. Keep in mind that following these templates will ensure that the boards will physically fit on all Xplained Pro kits.

**Extension template 1**

This is the most basic extension module and will fit all MCU boards. This means that this extension is the preferred module when starting a design.

Features:

- 30mm x 50mm
- 1 female standard extension header
- ID system
- 2 mounting holes with GND
- 2 test jig holes
- 2 rubber feet

**Figure 2-6. Standard extension 1 3D view**

**Figure 2-7. Standard extension 1 component placement**
Extension template 2

Compared to the basic default extension, this extension adds a power connector to the design. The extension type can be used if access to the power inputs/outputs of the MCU board is required.

- 45mm x 50mm
- 1 female standard expansion header
- 1 female power header
- ID system
- 2 mounting holes with GND
- 2 test jig holes
- 2 rubber feet

Figure 2-9. Standard extension 2 3D view
Figure 2-10. Standard extension 2 component placement

Connector Placement

3,80 (mm)

Power Connector

40,40 (mm)

Extension Connector

15,00 (mm)

3,80 (mm)

Figure 2-11. Standard extension 2 mechanical dimensions

Mechanical Dimensions

Ø 0,00 (mm)

3,50 (mm)

13,25 (mm)

6,50 (mm)

22,62 (mm)

2,35 (mm)

8,00 (mm)

50,00 (mm)

2,35 (mm)

42,65 (mm)

Ø 2,70 (mm)

45,00 (mm)

2,35 (mm)
Extension template 3

This extension is targeted for applications that require more signals than are available on one standard header e.g. when more than 2 ADC or PWM signals are needed. Please note that this extension might not fit all MCU boards since smaller boards will not have two extension headers on one side.

- 65.55mm x 50mm
- 2 female Standard extension headers
- ID system
- 2 mounting holes with GND
- 2 test jig holes
- 2 rubber feet

Figure 2-12. Standard extension 3 3D view
Figure 2-13. Standard extension 3 component placement

Connector Placement

Extension Connector 1

3,80 (mm)

15,00 (mm)

50,56 (mm)

Extension Connector 2

3,80 (mm)
Extension template 4

This extension is the same as extension 3 except for the additional power header.

- 80.55mm x 50mm
- 2 female standard extension headers
- 1 female power header
- ID system
- 2 mounting holes with GND
- 2 test jig holes
- 2 rubber feet
Figure 2-15. Standard extension 4 3D view
Figure 2-16. Standard extension 4 component placement

Connector Placement

- Power Connector
- Extension Connector 1
- Extension Connector 2

Dimensions:
- 3.80 mm
- 75.96 mm
- 50.56 mm
- 15.00 mm
Figure 2-17. Standard extension 4 mechanical dimensions

Extension template 5

This board size is designed to fit on all Xplained Pro MCU boards, but with an orientation for the header at the bottom of the Xplained Pro MCU board. Note that the board is too wide to be connected next to another extension board on the right hand side.

Features:

- 60mm x 60mm
- 1 female standard extension header
- ID system
- 4 mounting holes with GND
- 2 test jig holes
- 4 rubber feet
Figure 2-18. Standard extension 5 3D view
Figure 2-19. Standard extension 5 component placement

Connector Placement

30,00 (mm)

3,80 (mm)

Extension Connector
2.5.2 Designing a board with the segment LCD connector

**Board size constraint**

Xplained Pro MCU boards will support segment LCD boards up to **45mm x 30mm** with the Hirose DF9-51S-1V(69) connector in the absolute center of the board on the bottom layer. Figure 2-21, "Segment LCD board size constraint" shows an example drawing of a segment LCD board. The constraints are introduced to avoid collision with other on-board peripherals on Xplained Pro MCU boards. When the Hirose connectors on a MCU board and segment LCD board is stacked the total height is **4.3mm**, any through hole pins on a segment LCD should not be too long as they may collide with resistors/capacitors on the MCU board.
Important

Segment LCD boards should not be larger than **45mm x 30mm** with the Hirose connector located at the center of the board. Boards larger than this may collide with components on a Xplained Pro MCU board.

Figure 2-21. Segment LCD board size constraint

2.5.3 Designing a board with the LCD connector

Atmel has not designed any extension boards with the LCD connector. Please contact Atmel if you are interested in creating an extension board using this connector.
3. How to Integrate a Xplained Pro Extension in Atmel Studio

3.1 Xplained Pro Landing Page

When a Xplained Pro MCU kit is connected to a computer running Atmel Studio a landing page for the kit is shown. Figure 3-1, “Xplained Pro Landing Page in Atmel Studio” shows a landing page for SAM4S Xplained Pro with PROTO1-, IO1- and OLED1 Xplained Pro connected. The landing page provides the connected kits name, a picture of the kit, a description of the kit, links to relevant documentation/websites and all the information stored in the Xplained Pro ID chip located on the extension modules.

Figure 3-1. Xplained Pro Landing Page in Atmel Studio

A future version of the Atmel XDK will contain an API for adding kit information to the landing page in Atmel Studio.

Any developers that would like to add information about an Xplained Pro extension to Atmel Studio has to contact Atmel via e-mail: gallery@atmel.com with landing page information and a picture of the kit as shown in the figure above. The landing page uses the name of the kit stored in the ID chip as a key to display the correct information, it is therefore important that Atmel knows the exact name that will be programmed into the chip.

1 mailto:gallery@atmel.com
4. **How to Integrate Code Examples for a Xplained Pro Extension in Atmel Studio**

A future version of this document will describe how to integrate code examples for a Xplained Pro extension in Atmel Studio.
A. Appendix

A.1 Document Revision History

<table>
<thead>
<tr>
<th>Doc. Rev.</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>06/2013</td>
<td>Restructured the document. Added information about Xplained Pro ID device programming and integration to Atmel Studio</td>
</tr>
<tr>
<td>B</td>
<td>03/2013</td>
<td>Added a new chapter about Atmel Studio integration</td>
</tr>
<tr>
<td>A</td>
<td>02/2013</td>
<td>First release</td>
</tr>
</tbody>
</table>

A.2 xpro_id Version History

This chapter contains the changelog for the xpro_id python package used to program/read Xplained Pro ID devices.

A.2.1 Version 0.5

Version 0.5 is the initial released version.