Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

Atmel MCUs

Xplained Pro Hardware Development Kit (HDK)

USER GUIDE

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 and add example firmware.

Table of Contents

1. Introduction

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

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 needed.

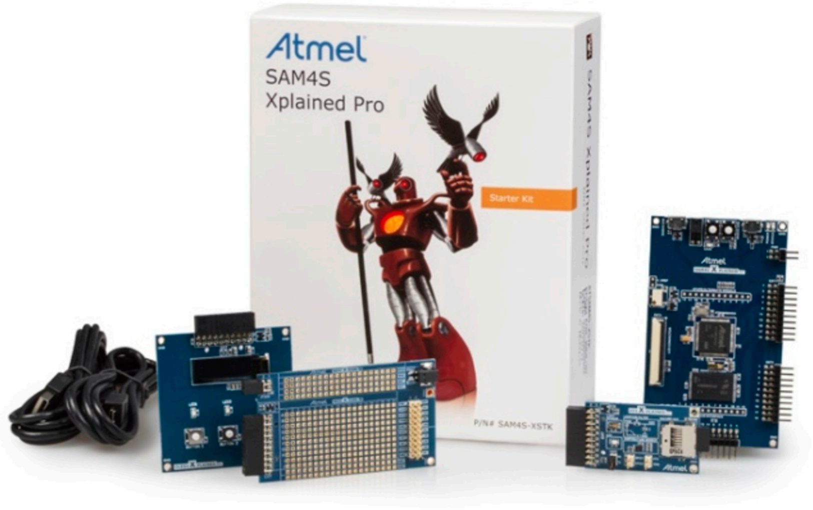
1.1. Compatible Xplained Pro Hardware

The Xplained Pro platform consists of several standardized building blocks that need to work together for the system to work. Otherwise interoperability issues may occur where extensions cannot be connected due to incompatible pinout or the hardware identification system does not work. Typical hardware building blocks on the Xplained Pro platform are:

- Standardized pinout and position for extension headers
- Standardized board sizes
- Embedded Debugger (EDBG)
- Hardware identification system
- Xplained Pro Analog Module (XAM)

The Hardware Development Kit provides all information that is necessary to create an Xplained Pro product that is compatible with the Xplained Pro platform. The above building blocks are described in this document.

Figure 1-1 Typical Xplained Pro Hardware



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 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 from the Atmel Gallery.

Related Links

Xplained Pro Landing Page on page 63

1.3. Example Code

The final step of the integration is addition of example code for the hardware. This is described in detail in the Software Development Kit (SDK). The SDK is available at the Atmel Gallery Partner site.

2. Xplained Pro Hardware Platform

2.1. Naming Convention

2.1.1. Product Hierarchy

The Xplained Pro platform consists of several boards, kits, and bundles. It is important to be accurate and consistent in all documentation when describing a physical Xplained Pro product.

The assembled PCB with components is an Xplained Pro board where:

- The microcontroller (MCU) board can be referred to as an Xplained Pro MCU board or Xplained Pro main board.
- The extension can be referred to as an Xplained Pro extension or Xplained Pro extension board

A (cardboard) box containing one Xplained Pro board is called an Xplained Pro kit. A kit always contains at least one Xplained Pro board, and may also contain additional components such as cables, storage media, or a display module. There are three types of Xplained Pro kits:

- A kit containing one MCU Xplained Pro board should be referred to as an Xplained Pro Evaluation Kit
- A kit containing one Xplained Pro extension should be referred to as an Xplained Pro Extension kit
- A kit containing several kits including cables and everything a new user needs to get started is called an Xplained Pro Starter Kit

Examples:

- SAM4L Xplained Pro MCU board
- SAM4L Xplained Pro Evaluation Kit
- SAM4L Xplained Pro Starter Kit
- OLED1 Xplained Pro Extension
- OLED1 Xplained Pro Extension Kit

2.1.2. Xplained Pro Main Board Naming Convention

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

[device_series_name] Xplained Pro

Examples:

- UC3 L Xplained Pro
- SAM4L Xplained Pro
- XMEGA® A1U Xplained Pro

The above suggestions only work if only one product for the MCU family exists. When sub family products are made it is required to add the sub-series part of the MCU name (or the memory size indicator for kits that have a new memory size derivate).

Examples:

- UC3 A3 Xplained Pro
- SAM4LC Xplained Pro
- SAM4L8 Xplained Pro

2.1.3. Xplained Pro Extension Naming Convention

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
- I/O1 Xplained Pro

2.1.4. 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 used in the below example is Verdana with a 0.5mm inverted border.

Figure 2-1 MCU Board Silkscreen Naming Example 1

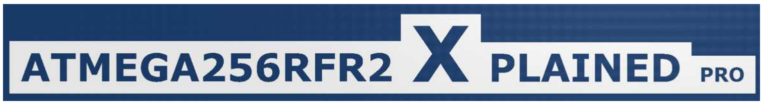


Figure 2-2 MCU Board Silkscreen Naming Example 2

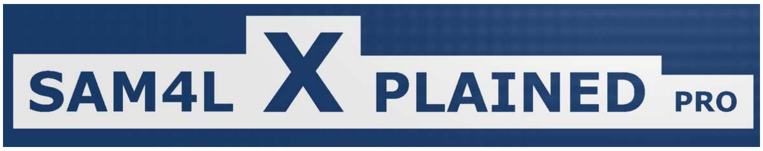


Figure 2-3 Extension Silkscreen Naming Example 1

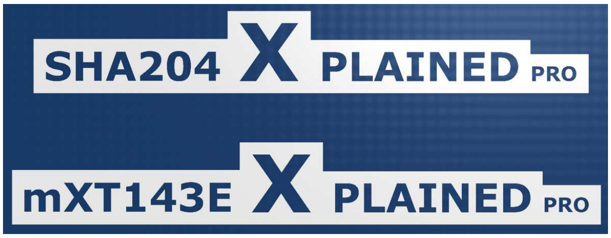
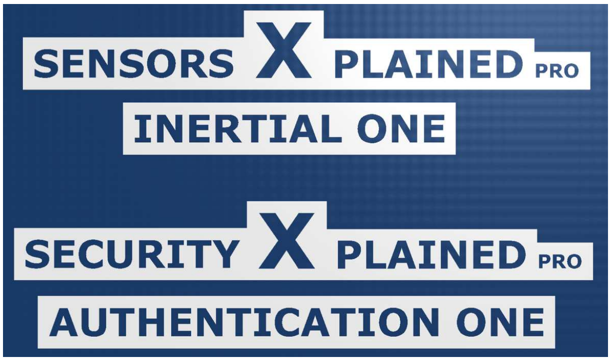


Figure 2-4 Extension Silkscreen Naming Example 2



2.2. Embedded Debugger

The Xplained Pro contains the Atmel Embedded Debugger (EDBG) for on-board debugging. The EDBG is a composite USB device of three interfaces; a debugger, Virtual COM Port, and a Data Gateway Interface (DGI).

Together with Atmel Studio, the EDBG debugger interface can program and debug the target device. On Xplained Pro, the programming interface is connected between the EDBG and the target device.

The Virtual COM Port is connected to a UART on the target device and provides an easy way to communicate with the target application through terminal software. It offers variable baud rate, parity, and stop bit settings. Note that the settings on the target device must match the settings given in the terminal software.

Info: If not set automatically, data terminal ready (DTR) must be set in the terminal software.

The DGI consists of several physical interfaces for communication with the host computer. Communication over the interfaces is bidirectional. It can be used to send events and values from the target device or as a generic printf-style data channel. Traffic over the interfaces can be timestamped on the EDBG for more accurate tracing of events. Note that timestamping imposes an overhead that reduces maximal throughput. Atmel Data Visualizer is used to send and receive data through DGI.

The EDBG controls two LEDs on Xplained Pro; a power LED and a status LED. Table 2-1 EDBG LED Control on page 8 shows how the LEDs are controlled in different operation modes.

Table 2-1 EDBG LED Control

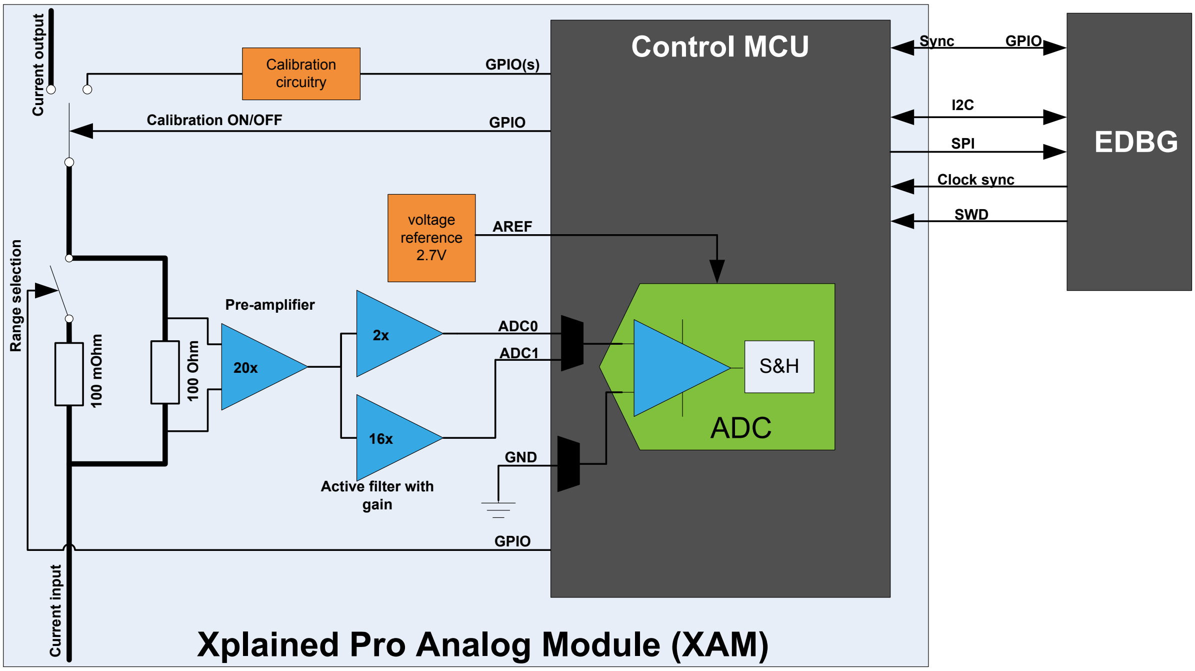
Operation mode	Power LED	Status LED
Normal operation	Power LED is lit when power is applied to the board.	Activity indicator, LED flashes when any communication happens to the EDBG.
Bootloader mode (idle)	The power LED and the status LED blinks simultaneously.	
Bootloader mode (firmware upgrade)	The power LED and the status LED blinks in an alternating pattern.	

For further documentation on the EDBG, see the EDBG User Guide.

2.3. Xplained Pro Analog Module (XAM)

2.3.1. Overview

The Xplained Pro Analog Module (XAM) extends the embedded debugger with high dynamic range current measurement. This enables power profiling of the target system.



The XAM consists of:

- Calibration circuitry
- Voltage reference
- Analog frontend
  - Shunt resistors with a range selection switch
  - Pre-amplifier
  - Two active filters with gain
- Control MCU
  - Analog to digital converter
  - Signal processing
  - Control/communication interface to the EDBG

The current measurement frontend is a high side shunt measurement with a pre-amplifier and a second active filter stage with gain. The wide dynamic range is achieved by four measurement ranges which are defined by two shunts and the two parallel second stage active filters with gain.

2.3.2. EDBG Interface

The Xplained Pro Analog Module (XAM) is connected to the EDBG with the following interfaces:

- I^{2}C: This is used to control and configure the XAM
- SPI: Current measurement data is streamed to the EDBG via this interface. This is a one-way data transfer channel from the XAM to the EDBG
- SWD: The MCU in the XAM is programmed via SWD from the EDBG
- GPIO: At least one GPIO that is connected to the EDBG from the target MCU is also connected to the current measurement unit to enable the user to sync current measurements with his application
- Clock sync: Synchronization signal to synchronize ADC measurements with EDBG
- Reference clock: Reference clock for the XAM

2.3.3. Sample Rate

The raw sampling rate of the Xplained Pro analog module (XAM) is up to 250kHz and with the default averaging configuration (average of 16 samples) the actual output of the XAM is 16.67kSPS (note that the XAM output sample rate is not an integer fraction of the raw sampling).

2.3.4. Measurement Ranges and Accuracy

The Xplained Pro analog module has four measurement ranges. These are defined by two shunt resistors and two gain stages.

Measurement range	Hardware	Resolution	Accuracy	Comments
Range 1	Low current shunt and high gain stage	20nA	1 LSB ±1%	Below 1µA the error will increase. Typical error for 300nA is 1 LSB ± 10%
Range 2	Low current shunt and low gain stage	150nA	1 LSB ±1%	
Range 3	High current shunt and high gain stage	10µA	1 LSB ±1%	
Range 4	High current shunt and low gain stage	100µA	1 LSB ±1%	Above 100mA the error will increase to 1 LSB ±5% at 400mA. Maximum current is 400mA

The ranges are switched automatically by the XAM to achieve best measurement results and the currently active range is visualized in the Atmel Data Visualizer frontend tool. The maximum voltage drop over the shunt resistor is 100mV and the XAM will switch the range automatically before this limit is reached.

2.4. Xplained Pro ID System

2.4.1. Overview

Identification of extensions 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.

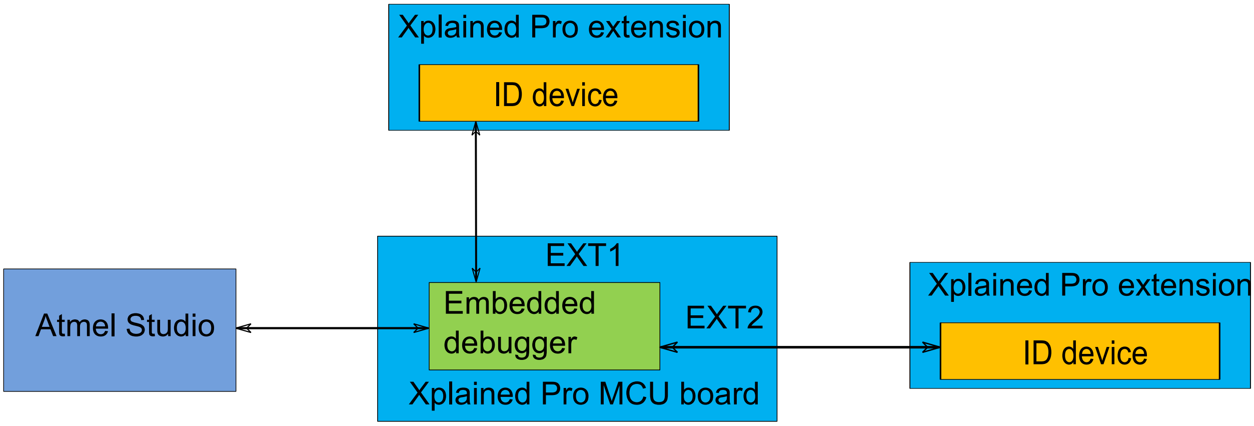
Identified extensions are reported through the Embedded Debugger to the host PC software, which is Atmel Studio. Based on the detected hardware Atmel Studio will then provide additional information to the user such as:

- 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.

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

Figure 2-5 ID System Overview



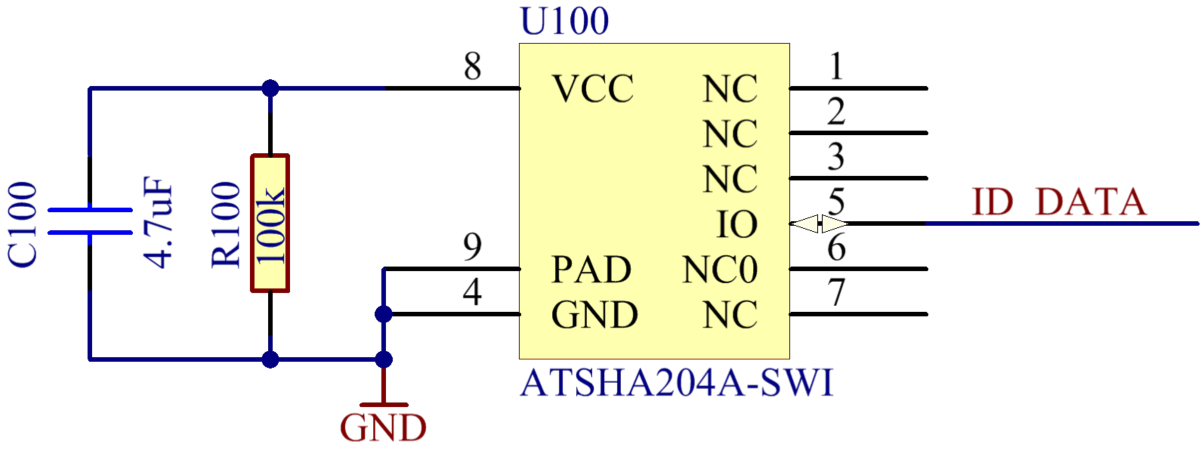
2.4.2. ID System Implementation on Extensions

The ID device that must be mounted on Xplained Pro extensions is the Atmel ATSHA204A 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 ATSHA204A-MAHCZ-T is used. Relevant features of the device are:

- Operation voltage from 2.0V to 5.5V
- Single wire interface
- 8-lead UDFN (one wire)
- Data area with 512 bytes
- Configuration area with 88 bytes
- One time programmable (OTP) area with 64 bytes

The example in Figure 2-6 ID Device Circuitry on page 12 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 through an internal diode between pin 5 and pin 8. *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 2-6 ID Device Circuitry



2.4.3. 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 [mV].
6. Maximum supported voltage for the extension board [mV].
7. Minimum current that is required to support the extension board [mA].

The product name is the key for a lookup in the available kits list in Atmel Studio and it is therefore vital that this information is unique and always present. If a kit name cannot be resolved in Atmel Studio it will be suggested to the user that he or she should update or install the required Atmel Studio extension for the extension kit. All Atmel extension kits will be identified by the Atmel Studio extension Atmel Kits.

The above data is placed in the OTP (One Time Program) zone, which means once it is programmed into the ID device memory it can’t be erased or re-written.

2.4.4. 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 six 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 six bytes for the max./min. values are 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 2-2 Exampled Content for the ID Device

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

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

Info:
Four bytes are used for string terminations ('\0'), six 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.

Info:
The minimum and maximum voltage parameters is used if the Xplained Pro boards supports other target voltages than 3.3V and switching of power (VCC) to the Extension connectors. The Extension kits voltage range can be read from the ID chip without applying power to the Extension kit, if the target voltage is within the valid voltage range of the Extension kit power will be switched on.

2.4.5. 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.

2.4.6. Programming the ID Device

The ID device can be programmed via the Embedded Debugger that is mounted on 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® CLI for reading and programming ID devices called id_tool. The CLI is tested with `Python 2.7.10`[1]. The latest version of the id_tool package can be downloaded from the Atmel Gallery developer page.

The Python CLI is distributed as source and split in two files; edbg_driver and id_tool. edbg_driver interfaces cmsis_dap.dll to communicate with an embedded debugger and provides the required functions to read and program Xplained Pro ID devices. id_tool contains the CLI and that interfaces the edbg_driver. To get started, run the following command:

```
C:\Python27\python.exe id_tool.py -h
```

Info:
The id_tool is provided as a CLI that can be used to read and program Xplained Pro ID devices. The code may be altered to fit a specific manufacturing setup.

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

Related Links

id_tool Version History on page 65

2.5. Xplained Pro Connectors

2.5.1. Extension Header Numbering

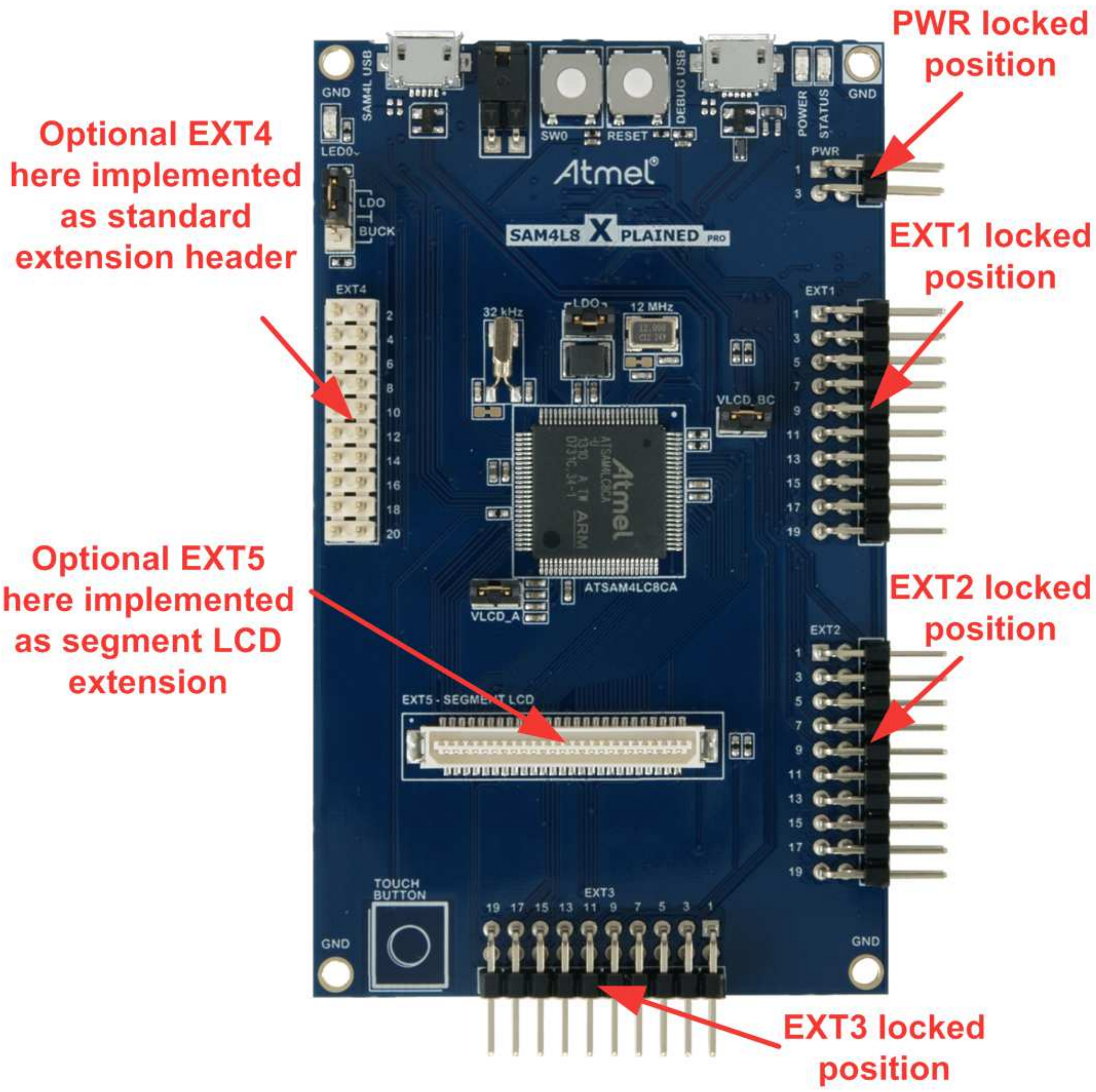
The extension headers are given names EXTn where n ϵ [1…7], n is determined by which ID pin is connected to the embedded debugger. A header with ID7 signal from the embedded debugger connected should be called EXT7. PWR, EXT1, EXT2 and EXT3 are standard extension headers that have a predfined position according to the list below:

- PWR is right angled at the top right hand side of the board. This header must always be implemented.
- EXT1 is right angled at the top right hand side of the board, located below the PWR header. This header must always be present.
- EXT2 is right angled and at the bottom right hand side of the board. This header is mandatory for medium and large boards and should not be implemented on small boards.
- EXT3 is right angled pointing downwards

All MCU boards have to implement at least PWR, EXT1, EXT2 (on medium and large boards), and EXT3. EXT4 to EXT7 can be placed differently depending on the board design. EXT4 to EXT7 can either be standard extension headers or application specific headers.

[1] The module should be compatible with other 2.7.x versions of Python too.

Figure 2-7 Example Extension Header Numbering and Placement



PWR locked position

Optional EXT4 here implemented as standard extension header

EXT1 locked position

Optional EXT5 here implemented as segment LCD extension

EXT2 locked position

EXT3 locked position

2.5.2. Xplained Pro Standard Extension Header

All Xplained Pro kits have one or more dual row, 20-pin, 100mil extension header. Xplained Pro MCU boards have male headers, while Xplained Pro extensions have their female counterparts. Note that all pins are not always connected. All connected pins follow the defined pin-out description in Table 2-3 Xplained Pro Standard Extension Header on page 16.

The extension headers can be used to connect a variety of Xplained Pro extensions to Xplained Pro MCU boards or to access the pins of the target MCU on Xplained Pro MCU boards directly.

Table 2-3 Xplained Pro Standard Extension Header

Pin number	Name	Description
1	ID	Communication line to the ID chip on an extension board
2	GND	Ground
3	ADC(+)	Analog to digital converter, alternatively positive part of differential ADC
4	ADC(-)	Analog to digital converter, alternatively negative part of differential ADC
5	GPIO1	General purpose I/O
6	GPIO2	General purpose I/O
7	PWM(+)	Pulse width modulation, alternatively positive part of differential PWM
8	PWM(-)	Pulse width modulation, alternatively negative part of differential PWM
9	IRQ/GPIO	Interrupt request line and/or general purpose I/O
10	SPI_SS_B/ GPIO	Slave select for SPI and/or general purpose I/O
11	I^2C_SDA	Data line for I^2C interface. Always implemented, bus type.
12	I^2C_SCL	Clock line for I^2C interface. Always implemented, bus type.
13	UART_RX	Receiver line of target device UART
14	UART_TX	Transmitter line of target device UART
15	SPI_SS_A	Slave select for SPI. Should preferably be unique.
16	SPI_MOSI	Master out slave in line of serial peripheral interface. Always implemented, bus type.
17	SPI_MISO	Master in slave out line of serial peripheral interface. Always implemented, bus type.
18	SPI_SCK	Clock for serial peripheral interface. Always implemented, bus type.
19	GND	Ground
20	VCC	Power for extension board

2.5.2.1. Populating Extension Headers

The number of extension headers and how they are populated will greatly affect the Xplained Pro platform as a whole. Since there are 20 pins per standard extension header some pins from the target MCU will be shared between several extension headers. If the pin-out is not done thoroughly there will be many compatibility issues when several extension modules are connected at the same time. There has to be some sharing of MCU pins on the extension headers, if not too few headers will be available for the user.

EXT1 should be a golden extension header meaning this header should not share any of its MCU pins with other extension headers. Demo code for a specific extension module should always run on EXT1 without the need to modify the code. For low pin-count devices it might be difficult to achieve this but in all cases this should be at least attempted.

EXT3 is intended for user interface boards containing displays, buttons etc, and these boards will most likely be used a lot. For all other standard extension connectors a lower number should have priority over a connector with a higher number, e.g. EXT2 > EXT4.

Table 2-4 Recommended Priority for the Standard Xplained Pro Extension Header Population

Pin number	Name	Recommendation
1	ID	Always connect to a unique ID line on the embedded debugger
2	GND	
3	ADC(+)	First priority for ADC
4	ADC(-)	Second priority for ADC
5	GPIO1	First priority for GPIO (devices that have USART RTS hardware support should route this function to this pin)
6	GPIO2	Second priority GPIO (for devices that have USART CTS hardware support should route this function to this pin)
7	PWM(+)	First priority for PWM
8	PWM(-)	Second priority for PWM
9	IRQ/GPIO	Should be unique if possible, priority 2
10	SPI_SS_B/GPIO	Should be unique if possible, priority 3
11	TWI_SDA	Always implemented
12	TWI_SCL	Always implemented
13	USART_RX	
14	USART_TX	
15	SPI_SS_A	Should be unique if possible, priority 1
16	SPI_MOSI	Always implemented
17	SPI_MISO	Always implemented
18	SPI_SCK	Always implemented
19	GND	
20	VCC	

Signals with unique priority should be populated first e.g. SPI_SS_A, then IRQ/GPIO, and lastly SPI_SS_B/GPIO. After Unique priorities have been filled other signals with the lowest number have the highest priority e.g. PWM(+), GPIO1 and ADC(+) must be populated before ADC(-), GPIO2 and PWM(-).

SPI, TWI, and UART should always be implemented. SPI and TWI can always be implemented because they are bus types.

It is not allowed to connect two (or more) MCU signals to one extension connector pin, as this will make the not used MCU pin unavailable to the designer for alternative use. It is however an option to connect one MCU signal to two or more extension connector pins, to ensure that we can make standard functionality available on all extension connectors.

For devices with hardware flow control for the UART it is beneficial to route these signals to the EXT1 header GPIO pins because these would most likely also be used for this purpose on extension boards that require these signals. RTS should be routed to GPIO0 and CTS should be routed to GPIO1. Connecting the flow control signals is not a requirement but it is obvious that following this suggestion will offer better support for this functionality.

All I/O pins of the target device that are not connected to extension headers should be made easily accessible. Exceptions for this rule are:

- USB differential data signals to improve signal integrity and to prevent the users from directly touching the signals which can lead to data loss. We have seen that this can lead to a bad user experience if the USB connection is suddenly reset due to a touch of the user e.g. when he holds the board in his hand to show a demo.
- Crystal oscillator connections: The reason behind this is a possible disturbance of the oscillation and additional load capacitance.
- Any critical signal that would have degraded performance by this additional routing e.g. clock lines

2.5.2.2. QTouch on MCU and Extension Boards

Xplained boards with devices containing a build-in PTC module should try to overload the Table 2-3 Xplained Pro Standard Extension Header on page 16 functions with QTouch® lines to make it compatible with the QTouch extension boards.

QTouch Design of Small Boards

For small extensions with only one header (EXT1), four Y- and four X-lines are the maximum of touchlines that can be expected to be available for compatibility for the series of small MCU boards. Both the MCU and the extension board should be designed with this rule set.

For compatibility with some of the first extension boards Y-lines noted in parentheses might optionally be implemented on the MCU board, but this should only be done with XY-capable QTouch lines for (Y-line 5 and 6).

Table 2-5 Recommended Priority for PTC Signals on One Header Boards (EXT1)

Pin number EXT1	MCU board	Extension board	Recommendation
3	**Y-line 1**	Y-line 1	1st priority for QTouch Y-lines
4	**Y-line 2**	Y-line 2	2nd priority for QTouch Y-lines
5	**Y-line 3**	Y-line 3	3rd priority for QTouch Y-lines
6	**Y-line 4**	Y-line 4	4th priority for QTouch Y-lines
7	**X-line 1**	X-line 1	1st priority for QTouch X-lines
8	**X-line 2**	X-line 2	2nd priority for QTouch X-lines
9	**X-line 3** (Y-line 5)	X-line 3	3rd priority for QTouch X-lines
10	**X-line 4** (Y-line 6)	X-line 4	4th priority for QTouch X-lines
11			
12			

Pin number EXT1	MCU board	Extension board	Recommendation
13			
14			
15			
16	(Y-line 7)		
17			
18	(Y-line 8)		

QTouch Design of Medium and Large Boards
For medium QTouch extension boards with header EXT1 and EXT2, only the four first lines on each header should be used to ensure compatibility with most MCU boards.

If more than four Y-lines are required on the extension kit, use the ones marked in the table with the loss of compatibility with one or more MCU board. If four or less Y-lines are required, duplicate the four first X-lines from the EXT2 header on the EXT1 header to make it compatible with small MCU boards.
MCU boards should be designed with the minimum of four Y-lines on EXT1 and four X-lines on EXT2. For compatibility with the QT1 Xplained Pro board, Y-lines 5-8 can optionally be designed in. For compatibility to small extension boards, XY-lines should be designed in on pin 7-10.

Table 2-6 Recommended Priority for PTC Signals on Two Header Boards (EXT1)

Pin number on EXT1	MCU board	Extension board	Recommendation
3	**Y-line 1**	Y-line 1	1st priority for QTouch Y-lines
4	**Y-line 2**	Y-line 2	2nd priority for QTouch Y-lines
5	**Y-line 3**	Y-line 3	3rd priority for QTouch Y-lines
6	**Y-line 4**	Y-line 4	4th priority for QTouch Y-lines
7	(X-line 5)	X-line 1 duplicated	5th priority for QTouch X-lines
8	(X-line 6)	X-line 2 duplicated	6th priority for QTouch X-lines
9	Y-line 5 (X-line 7)	Y-line 5 or X-line 3 duplicated	5th priority for QTouch Y-lines (Use XY capable lines if possible for supporting one header boards)
10	Y-line 6 (X-line 8)	Y-line 6 or X-line 4 duplicated	6th priority for QTouch Y-lines (Use XY capable lines if possible for supporting one header boards)
11			
12			
13			
14			

Pin number on EXT1	MCU board	Extension board	Recommendation
15			
16	Y-line 7	Y-line 7	7th priority for QTouch Y-lines
17			
18	Y-line 8	Y-line 8	8th priority for QTouch X-lines

Table 2-7 Recommended Priority for PTC Signals on Two Header Boards (EXT2)

Pin number on EXT2	MCU board	Extension board	Recommendation
3	**X-line 1**	X-line 1	1st priority for QTouch X-lines
4	**X-line 2**	X-line 2	2nd priority for QTouch X-lines
5	**X-line 3**	X-line 3	3rd priority for QTouch X-lines
6	**X-line 4**	X-line 4	4th priority for QTouch X-lines
7	X-line 9	X-line 5	9th priority for QTouch X-lines
8	X-line 10	X-line 6	10th priority for QTouch X-lines
9	X-line 11	X-line 7	11th priority for QTouch X-lines
10	X-line 12	X-line 8	12th priority for QTouch X-lines
11			
12			
13			
14			
15			
16			
17			
18			

2.5.3. 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 any 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 when using the target voltage pin.

Table 2-8 Xplained Pro Power Header

Pin number	Pin name	Description
1	VEXT_P5V0	External 5V input
2	GND	Ground
3	VCC_P5V0	Unregulated 5V (output, derived from one of the input sources)
4	Target VTG	Regulated target voltage (output, used as main power supply for the kit)

2.5.4. Current Measurement Header

All Xplained Pro MCU boards feature an angled 1x2, 100mil pin-header marked with MCU current measurement that is located at the upper edge of the board. All power to the target device is routed through this header. The header is populated with a jumper cap. The purpose of this header is to enable power consumption measurements of the target device with external equipment.

2.5.5. Xplained Pro Segment LCD Connector

Xplained Pro MCU boards that have a microcontroller that supports segment LCDs can implement a 51-pin segment LCD extension connector. This connector is implemented with HIROSE 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). The connector has a standardized pin-out as shown in Table 2-9 Xplained Pro Segment LCD Connector on page 21.

Info:
All pins are not connected on all Xplained Pro MCU boards, it depends on how many segments and common terminals the target MCU supports.

Pin 37, 38, 39, 40, 41 and 42 can alternatively be used for QTouch signals. When they are used for touch they should not be used for display segments.

Table 2-9 Xplained Pro Segment LCD Connector

Description	Function	Pin	Pin	Function	Description
Common terminal 3	COM3	1	2	COM2	Common terminal 2
Common terminal 1	COM1	3	4	COM0	Common terminal 0
Segment 0	SEG0	5	6	SEG1	Segment 1
Segment 2	SEG2	7	8	SEG3	Segment 3
Segment 4	SEG4	9	10	SEG5	Segment 5
Segment 6	SEG6	11	12	SEG7	Segment 7
Segment 8	SEG8	13	14	SEG9	Segment 9
Segment 10	SEG10	15	16	SEG11	Segment 11
Segment 12	SEG12	17	18	SEG13	Segment 13
Segment 14	SEG14	19	20	SEG15	Segment 15
Segment 16	SEG16	21	22	SEG17	Segment 17

Description	Function	Pin	Pin	Function	Description
Segment 18	SEG18	23	24	SEG19	Segment 19
Segment 20	SEG20	25	26	SEG21	Segment 21
Segment 22	SEG22	27	28	SEG23	Segment 23
Segment 24	SEG24	29	30	SEG25	Segment 25
Segment 26	SEG26	31	32	SEG27	Segment 27
Segment 28	SEG28	33	34	SEG29	Segment 29
Segment 30	SEG30	35	36	SEG31	Segment 31
Segment 32 / QTouch X-line 2	SEG32 / QT_X2	37	38	SEG33 / QT_Y2	Segment 33 / QTouch Y-line 2
Segment 34 / QTouch X-line 1	SEG34 / QT_X1	39	40	SEG35 / QT_Y1	Segment 35 / QTouch Y-line 1
Segment 36 / QTouch X-line 0	SEG36 / QT_X0	41	42	SEG37 / QT_Y0	Segment 37 / QTouch Y-line 0
Common terminal 4	COM4	43	44	COM5	Common terminal 5
Common terminal 6	COM6	45	46	COM7	Common terminal 6
Backlight anode	Backlight V+	47	48	Backlight V-	Backlight Cathode
Backlight control	Backlight CTRL	49	50	ID	Xplained Pro ID
Ground	GND	51			

2.5.6. Xplained Pro LCD Extension 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 touch controller. The connector pin-out definition is shown in Table 2-10 Xplained Pro LCD Connector on page 22. Note that usually only one display interface is implemented, either the 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 Xplained Pro designs and can be used as a reference.

Table 2-10 Xplained Pro LCD Connector

Pin number	Name	RGB interface description	MCU interface description
1	ID	Communication line to the ID chip on an extension board	
2	GND	Ground	
3	D0	Data line	
4	D1	Data line	
5	D2	Data line	
6	D3	Data line	
7	GND	Ground	

Pin number	Name	RGB interface description	MCU interface description
8	D4	Data line	
9	D5	Data line	
10	D6	Data line	
11	D7	Data line	
12	GND	Ground	
13	D8	Data line	
14	D9	Data line	
15	D10	Data line	
16	D11	Data line	
17	GND	Ground	
18	D12	Data line	
19	D13	Data line	
20	D14	Data line	
21	D15	Data line	
22	GND	Ground	
23	D16	Data line	
24	D17	Data line	
25	D18	Data line	
26	D19	Data line	
27	GND	Ground	
28	D20	Data line	
29	D21	Data line	
30	D22	Data line	
31	D23	Data line	
32	GND	Ground	
33	PCLK / CMD DATA SEL	Pixel clock	Display RAM select. One address line of the MCU for displays where it is possible to select either register or data interface.
34	VSYNC / CS	Vertical Synchronization	Chip select
35	HSYNC / WE	Horizontal Synchronization	Write enable signal
36	DATA ENABLE / RE	Data enable signal	Read enable signal

Pin number	Name	RGB interface description	MCU interface description
37	SPI SCK	Clock for serial peripheral interface	
38	SPI MOSI	Master out slave in of serial peripheral interface	
39	SPI MISO	Master in slave out of serial peripheral interface	
40	SPI SS	Slave select for serial peripheral interface. Preferably a dedicated pin.	
41	ENABLE	Display enable	
42	I^{2}C SDA	I^{2}C data	
43	I^{2}C SCL	I^{2}C clock	
44	IRQ1	Interrupt 1	
45	IRQ2	Interrupt 2	
46	PWM	Backlight control	
47	RESET	Extension reset	
48	VCC	3.3V power supply for extension board	
49	VCC	3.3V power supply for extension board	
50	GND	Ground	

2.6. 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. The available power sources and specifications are listed in the table below.

Table 2-11 Power Sources for Xplained Pro

Power input	Voltage requirements	Current requirements	Connector marking
External power	5V ±2% (±100mV) for USB host operation. 4.3V to 5.5V if USB host operation is not required	Recommended minimum is 1A to 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.	PWR
Embedded debugger USB	4.4V to 5.25V (according to USB spec.)	500mA (according to USB spec.)	DEBUG USB
Target USB	4.4V to 5.25V (according to USB spec.)	500mA (according to USB spec.)	TARGET USB

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.

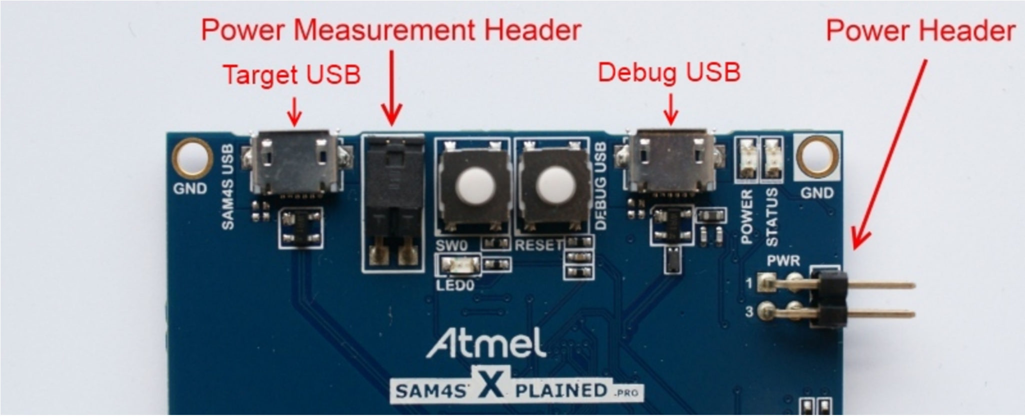
Info:
External power is required when the 500mA through the USB connector is not enough to power a connected USB device in a USB host application.

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

Some Xplained Pro MCU kits implements a separate 3.3V regulator for the EDBG.

Some Xplained Pro MCU kits have support for 5.0V for the target MCU, and extension headers/connector. These kits implements full level shift between the EDBG and target MCU.

Figure 2-8 Typical Xplained Pro Power Connections

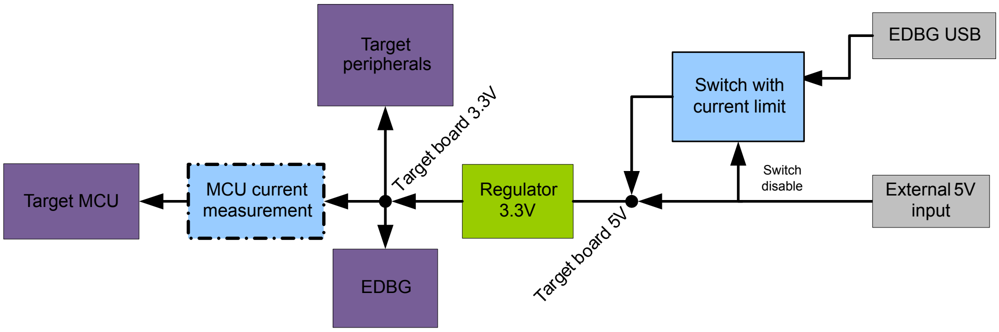


2.6.1. Typical Power Supply Implementations

2.6.1.1. No Target USB

When the target MCU does not offer a USB interface the power supply system can be reduced to the configuration shown below.

Figure 2-9 Power Supply Block Diagram for Boards with no Target USB



2.6.1.2. Target USB

When a target MCU offers a USB device interface it is mandatory to implement this interface and hook it up to the power supply system.