Atmel START User's Guide

Atmel START User's Guide

Overview

Atmel START helps you getting started with microcontroller development. It allows you to select MCU, configure software components, drivers, middleware, and example projects to tailor your embedded application in a usable and optimized manner. Once you are done you can download the generated code project and open it in Atmel Studio or another third-party development tool.

With Atmel START you can:

- Get help selecting an MCU based on both software and hardware requirements
- Find and develop examples for your board
- Configure drivers, middleware, and example projects
- Get help with setting up a valid PINMUX layout
- Configure system clock settings

© 2018 Microchip Technology Inc.

Table of Contents

Overview....1

1. Introduction....4
   - 1.1. Atmel START, Software Content, and IDEs....4
   - 1.2. Quick Start and Workflow Overview....6
2. Requirements and Compatibility....8
   - 2.1. Supported Web Browsers....8
   - 2.2. Supported IDEs and Compilers....8
3. Getting Started: Atmel START....9
   - 3.1. Documentation Resources....9
   - 3.2. Getting Started....10
   - 3.3. How to Add and Configure Software Components....18
   - 3.4. Save and Restore Configurations....20
4. Configuration Screens....21
   - 4.1. Clock Configurator....21
   - 4.2. Event System Configurator....25
   - 4.3. Atmel START QTouch® Capacitive Sensing Library....35
   - 4.4. PINMUX Configurator....36
   - 4.5. CCL Configurator....37
5. TrustZone Support....43
   - 5.1. Creating a Secure Project....43
   - 5.2. Creating a Non-Secure Project....45
   - 5.3. Importing the Secure Project in Atmel Studio....45
   - 5.4. Importing the Non-Secure Project in Atmel Studio....47
   - 5.5. Importing the Secure Project in µVision® from Keil®....50
   - 5.6. Importing the Non-Secure Project in µVision® from Keil®....54
   - 5.7. Importing the Secure Project in IAR Embedded Workbench®....58
   - 5.8. Importing the Non-Secure Project in IAR Embedded Workbench®....58
   - 5.9. Importing the Secure Project using Makefile....58
   - 5.10. Importing Non-Secure Project using Makefile....59
6. Using Atmel START Output in External Tools....60
   - 6.1. Atmel Studio....60
   - 6.2. IAR Embedded Workbench®....67
   - 6.3. Keil® µVision®....76
   - 6.4. GNU C Makefile....81
   - 6.5. SOMNIUM DRT....82
   - 6.6. MPLAB™ X IDE....87
7. Content in Atmel START....90

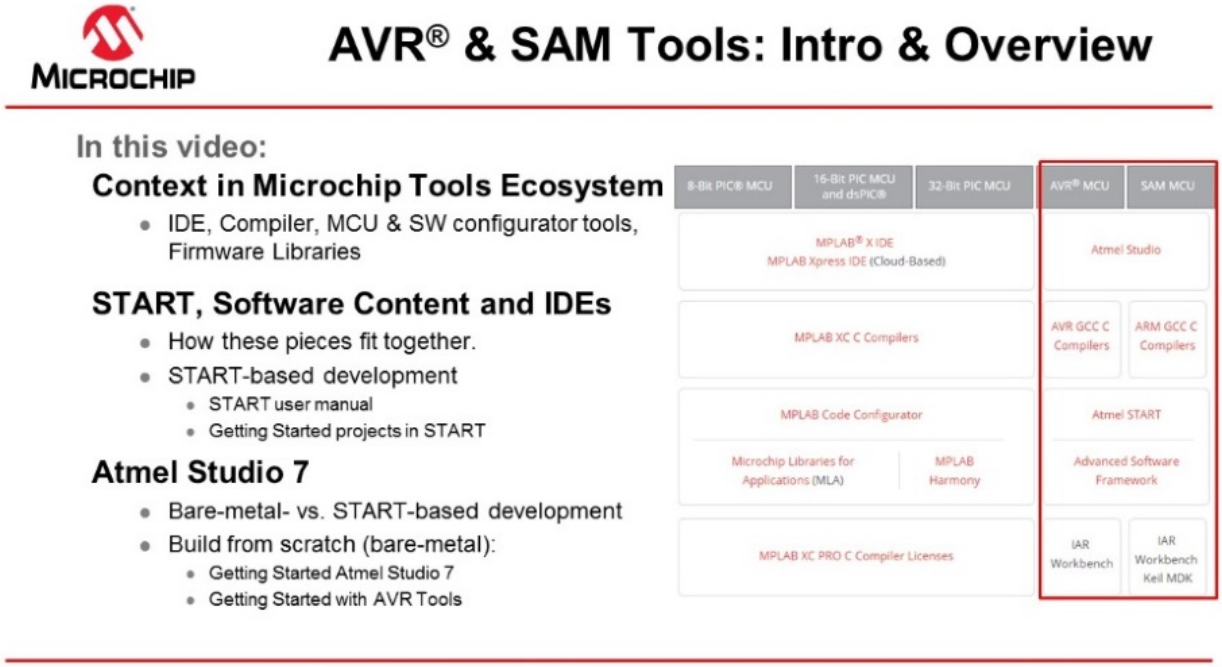
7.1. ARM Content....90
7.2. AVR® Content....90
8. Keyboard Shortcuts....92
9. Linking to Atmel START....93
10. Revision History....94
The Microchip Web Site....95
Customer Change Notification Service....95
Customer Support....95
Microchip Devices Code Protection Feature....95
Legal Notice....96
Trademarks....96
Quality Management System Certified by DNV....97
Worldwide Sales and Service....98

1. Introduction

1.1 Atmel START, Software Content, and IDEs

This section gives an overview of the various pieces in the AVR® and SAM Tools ecosystem and how they relate to each other.

Getting Started Topics



Video: AVR and SAM Tools ecosystem overview

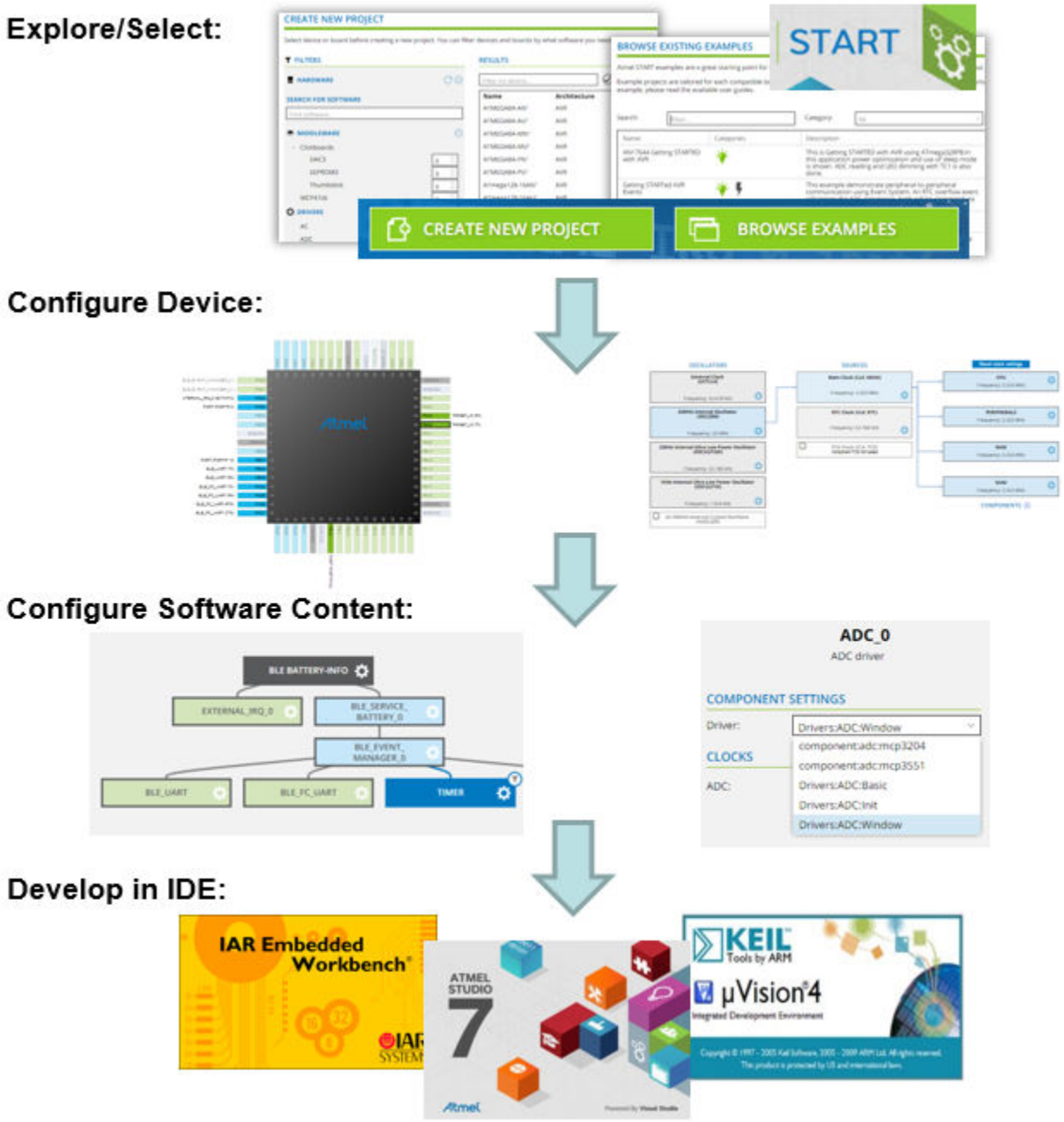
1.1.1 Atmel START

Atmel START is a web-based software configuration tool for various software frameworks, which helps you get started with MCU development. Starting from either a new project or an example project, Atmel START allows you to select and configure software components (from **ASF4** and **AVR Code**), such as drivers and middleware to tailor your embedded application in a usable and optimized manner. Once an optimized software configuration is done, you can download the generated code project and open it in the IDE of your choice, including Studio 7, IAR Embedded Workbench®, Keil® µVision®, or simply generate a makefile.

Atmel START enables you to:

- Get help with selecting an MCU, based on both software and hardware requirements
- Find and develop examples for your board
- Configure drivers, middleware, and example projects
- Get help with setting up a valid PINMUX layout
- Configure system clock settings

Figure 1-1. Relation Between START, Software Content, and IDEs



1.1.2 Software Content (Drivers and Middlewares)

Advanced Software Framework (ASF)

ASF, the Advanced Software Framework, provides a rich set of proven drivers and code modules developed by experts to reduce customer design-time. It simplifies the usage of microcontrollers by providing an abstraction to the hardware through drivers and high-value middlewares. ASF is a free and open-source code library designed to be used for evaluation, prototyping, design, and production phases.

ASF4, supporting the SAM product line, is the fourth major generation of ASF. It represents a complete re-design and -implementation of the whole framework, to improve the memory footprint, code performance, and the integration with the Atmel START web user interface. ASF4 must be used in conjunction with Atmel START, which replaces the ASF Wizard of ASF2 and 3.

microchip.com: ASF Product Page

AVR® Code
AVR Code, supporting the AVR product line, is a simple firmware framework for AVR 8-bit MCUs, equivalent to Foundation Services, which supports 8- and 16-bit **PIC®** MCUs. **AVR Code** is optimized for code-size and -speed, as well as simplicity and readability of code. AVR Code is configured by Atmel START.

1.1.3 Integrated Development Environment (IDE)

An **IDE** (Integrated Development Environment) is used to develop an application (or further develop an example application) based on the software components, such as drivers and middlewares, configured in and exported from Atmel START. Atmel START supports a range of IDEs, including Studio 7, IAR Embedded Workbench®, Keil® µVision®.

Atmel Studio 7 is the integrated development platform (IDP) for developing and debugging all AVR and SAM microcontroller applications. The Atmel Studio 7 IDP gives you a seamless and easy-to-use environment to write, build, and debug your applications written in C/C++ or assembly code. It also connects seamlessly to the debuggers, programmers, and development kits that support AVR and SAM devices. The development experience between Atmel START and Studio 7 has been optimized. Iterative development of START-based projects in Studio 7 is supported through re-configure and merge functionality.

This Getting Started training for Atmel Studio 7 will guide you through all the major features of the IDE. It is designed as a video series with accompanying hands-on. Each section starts with a video, which covers that section.

1.2 Quick Start and Workflow Overview

ASF4 is a software library consisting of peripheral drivers and example applications. The Atmel START web page (http://start.atmel.com) is used to select which software modules are needed in the user's application, and to configure these modules according to the user's needs. When using the Export Project screen, the corresponding generated C-code project can be downloaded to the user's computer and imported into an IDE of the user's choice, such as Atmel Studio 7, IAR™ Embedded Workbench, or Keil µVision. The IDE is used to modify, compile, program, and debug the project.

Installation
ASF4 is configured using the Atmel START web page (http://start.atmel.com), and the configured application is, thereafter, downloaded to the user's computer as a zip file. ASF4 does not need to install any components locally. The user will normally want to have an IDE or stand-alone toolchain installed on their computer, so the user can compile, program, and debug the downloaded code.

Workflow
The workflow is quite straightforward, starting with either the Create New Project screen or the Browse Existing Examples screen.

Starting with the **Browse Existing Examples** screen:

The Atmel START examples were designed to be used as a starting point for development. Efficient filtering mechanisms are therefore available to help developers find the projects closest to their requirements, giving them high quality, production-ready code that will work out of the box. However, these example projects are also easy to modify as the software configuration can be extended by changing the pinout (PINMUX screen) or adding additional drivers or middlewares (project **DASHBOARD**). For example, adding support for an extra timer or even adding a BLE interface. See the *Configuration Screens* section of the Atmel START user manual to understand project configuration options using Atmel START. It is also possible to create or re-configure Atmel START projects directly from Atmel Studio 7 (**File** > **New** > **Atmel Start project**).

Starting with the **Create New Project** screen:

This screen was designed to help select an MCU for your project, based on both software and hardware requirements.

1. Filter MCUs by requirements before starting a project.
2. Add components to your project, i.e. peripheral drivers and middlewares.
3. Configure each component.
4. Export the project and add it into one of the supported IDEs for further development.

The role of the IDE and Running the Code

Once the user is happy with the software configuration, the project is exported as a zip-file to the developer's IDE of choice. See *Using Atmel START Output in External Tools* in the Atmel START user manual for instructions about how to do this, as well as present a list of supported IDEs. ASF4/Atmel START does not need to install any components on your local computer.

An IDE is used to develop the code required to extend the functionality of the example project into the final product, as well as compile, program, and debug the downloaded code. A downloaded application is ready to compile. Refer to your IDE documentation for instructions on compiling the code. The behavior of the downloaded code is application-dependent:

- Configuring a "New project" generates a main()-function initializing all drivers, but no other operations
- Configuring an "Example project" generates a main()-function performing more complex operations

Related Links

4. Configuration Screens
6. Using Atmel START Output in External Tools

2. Requirements and Compatibility

2.1 Supported Web Browsers

Atmel START works best with HTML5 compatible browsers.

This includes the newest versions of:

- Chrome
- Firefox
- Internet Explorer (version 11 or higher)
- Edge
- Opera
- Safari

2.2 Supported IDEs and Compilers

The output from Atmel START can be used in a set of software tools as:

- Atmel Studio 6.2 with the Atmel START extension installed
- Atmel Studio 7.0 or later
- IAR Embedded Workbench
- Keil µVision

Also, the Atmel START output can be used with the command line GNU C compiler, utilizing the generated Makefile.

Atmel Start uses CMSIS packs for generating code. Some useful links:

- General information on CMSIS packs http://www.keil.com/pack/doc/CMSIS/Pack/html/index.html
- Device Family Packs (DFP) for Atmel Studio can be downloaded from http://packs.download.atmel.com, but best managed from Pack Manager in Atmel Studio
- Keil Device Support, see under Atmel for DFPs http://www.keil.com/dd2

Related Links

6. Using Atmel START Output in External Tools

3. Getting Started: Atmel START

Open a browser and go to http://start.atmel.com.

3.1 Documentation Resources

The major sources of documentation are **Getting Started projects**, the **Atmel START User Manual**, as well as **reference manuals** for ASF4 and Foundation Services framework content.

Getting Started projects
How to use the different pieces of the system to get them to work together, for example, how to use Atmel START efficiently with an IDE, such as Atmel Studio 7, in order to build your embedded application.

Getting Started projects have training materials like video series and/or hands-on training manuals, linked to the project user guides. The example project itself often represents the goal or end-point of the related hands-on or video training material. Training materials will give you a workflow overview of how the Atmel START and your IDE work together.

- Open Browse Examples, click the **Category** drop-down menu, and select the Getting Started category
- Links to training materials are found in the example project user-guides, which can be accessed without opening the project
- See the Atmel START User Manual

START User Manual

- What is Atmel START?
- Quick Start and Workflow Overview
  - Using Getting Started projects, finding/reconfiguring relevant example projects
  - Create New Project, by selecting an MCU based on both software and hardware project requirements
- How to use the various Atmel START configuration screens:
  - Dashboard
  - PINMUX
  - Event System Configurator
  - QTouch® Configurator
- How to export projects to various IDEs as:
  - Atmel Studio 7
  - IAR Embedded Workbench
  - Keil µVision
  - Makefile
- Content overview, software that Atmel START can configure and generate:
  - ASF4
  - Foundation Services

ASF4 API Reference Manual

- ASF4 Software Architecture

- Driver implementation and Design conventions
- API reference

AVR® Code API Reference Manual

- Foundation Services Software Architecture
- Driver implementation and Design conventions
- API reference

3.2 Getting Started

The Atmel START examples are a great starting point for embedded programmers. Example projects will work out of the box but are also easy to modify by using Atmel START.

Example projects are tailored for each compatible board and device, resulting in high quality and production-ready code with a small memory footprint. To get more information about each example, read the available user guides, which are available without even opening the project.

Getting Started projects have training materials like video series and/or hands-on training manuals linked to the project user-guides. The example project itself often represents the goal or end-point of the related hands-on or video training material. This system-level training material is designed to show how the different pieces of the system work together, i.e. how to use Atmel START efficiently with an IDE, such as Atmel Studio 7, in order to build your embedded application.

- Open Browse Examples, click the **Category** drop-down menu, and select the Getting Started category
- Links to the training material are found in the example project user guides, which you can access without even opening the project

Specifically, as an introduction to ASF4, the Getting Started Sensors project and training material are particularly relevant. In the link "video tutorials" on the Atmel | START page (http://start.atmel.com/) you will find a series of videos which demonstrates this project. In this video series, we are building a project using the analog light and digital I^2C temperature sensors on the I/O1 Xplained Pro extension board. The video series is supported by a hands-on training document. Both training resources cover the following important concepts of working with ASF4 projects, including the *How to* list below.

How to:

- Use Atmel START to add and configure the required drivers and middlewares, also configuring pins and clocks
- Export Atmel START Projects to Atmel Studio 7, where many of the latest features are used such as using *I/O View* (<F1>) links to open the data sheet in the context of that register's description
- Learn the best practice of how to use Examples folder and application level functions from `Atmel_Start.c` to efficiently build project functionally
- Easily validate each development step using the data visualizer
- Reconfigure an Atmel START project from Atmel Studio 7 to build functionality in a cyclic manner

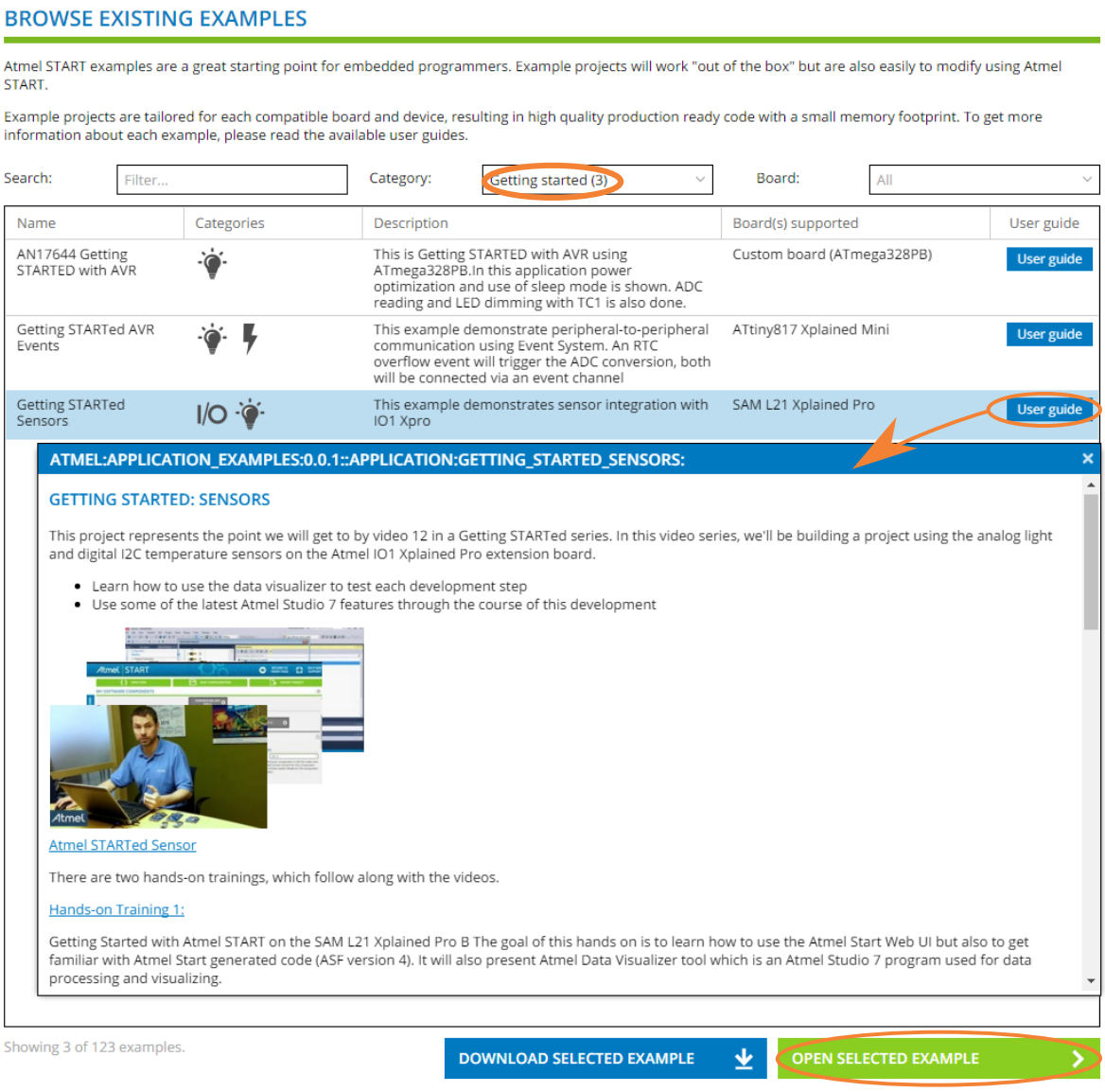
3.2.1 Getting Started: Opening an Example Project

1. Open a browser and go to http://start.atmel.com.
2. Select **BROWSE EXAMPLES**.
   A window showing all available example projects will open.

3. Select Getting Started from the Category (or type "Sensors" in the search filter) to find the **Getting Started Sensors** example project. Click **User guide** in order to get an overview of the project.

Tip: Example project user guides give a description of what the project does, required interface settings, as well as a simple list of steps to run the project. Getting Started Projects also includes links to training materials and/or videos in their user guides.

4. Once the example is useful, click on the example (makes it highlighted blue), then click **OPEN SELECTED EXAMPLE** to open it.



Tip: The example list can be filtered on search term, category, or board.
Search will filter examples on words in either the project name or description.

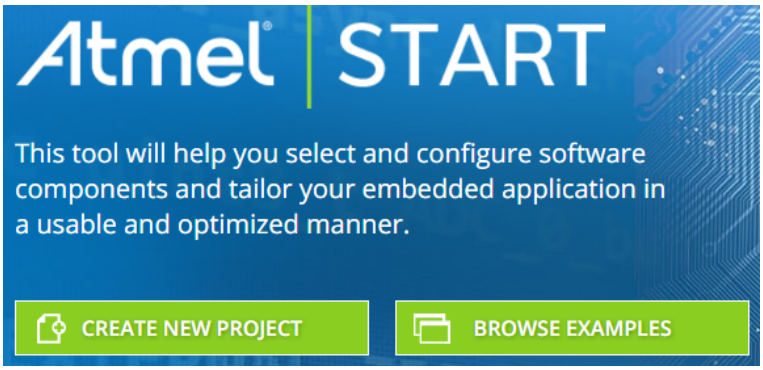
Category will filter examples on key technologies, such as BLE.

Board will filter examples on development kit names.

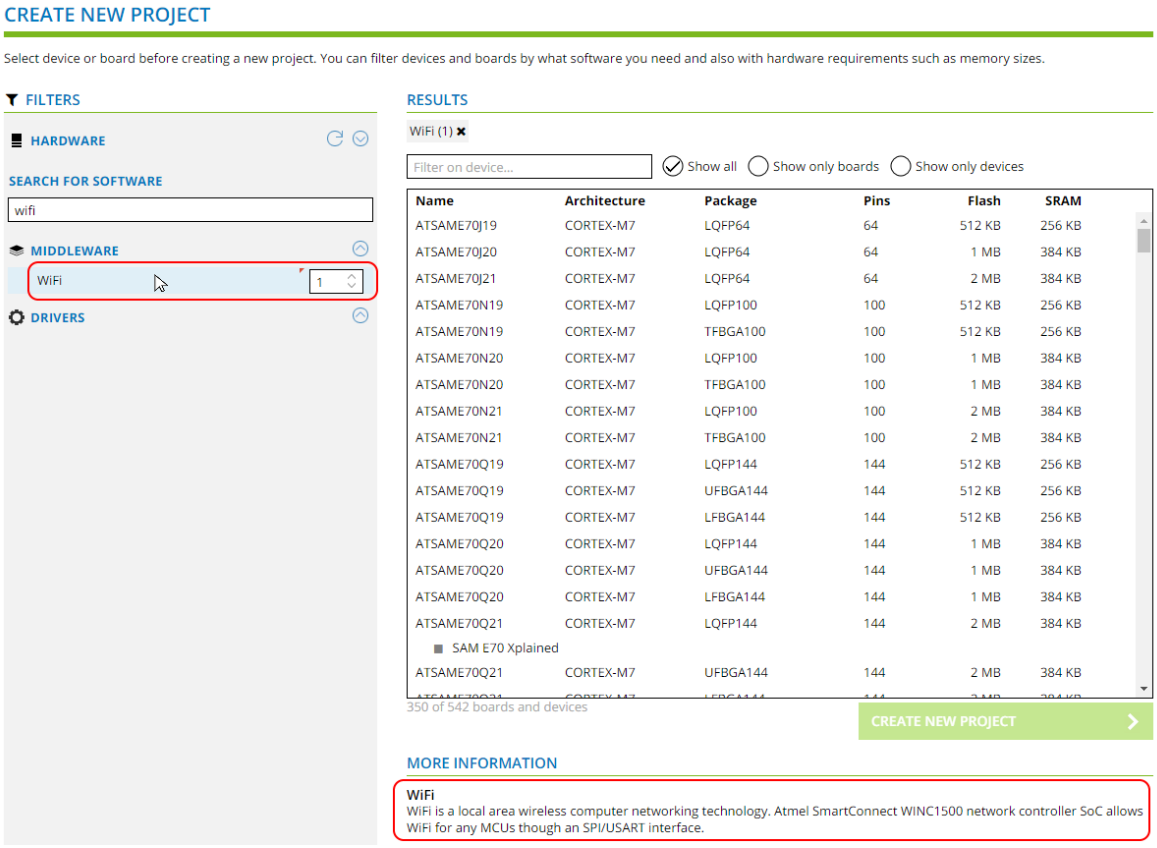
3.2.2 Getting Started: Creating a New Project

The **Create New Project** screen was designed to help select an MCU for your project, based on both software and hardware requirements. This example describes how to create a new project.

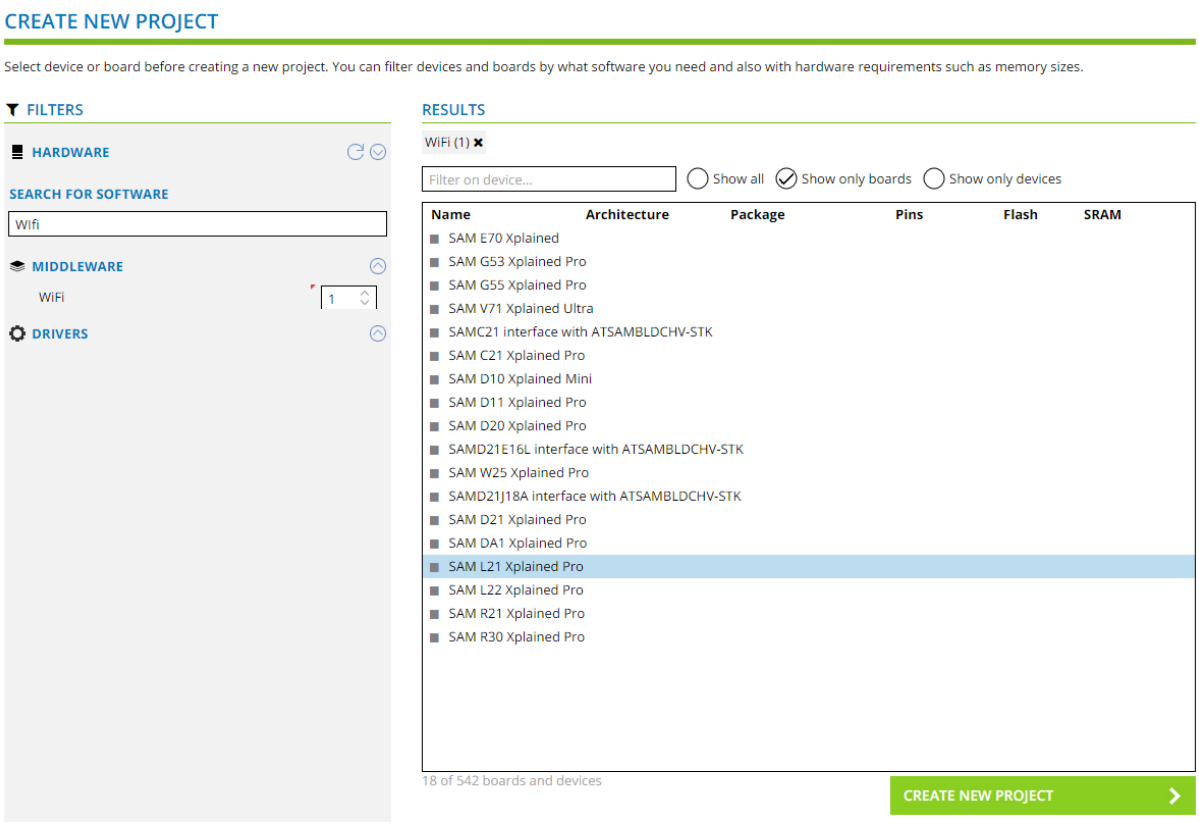
1. Open a browser and go to http://start.atmel.com.
2. Select **CREATE NEW PROJECT**.



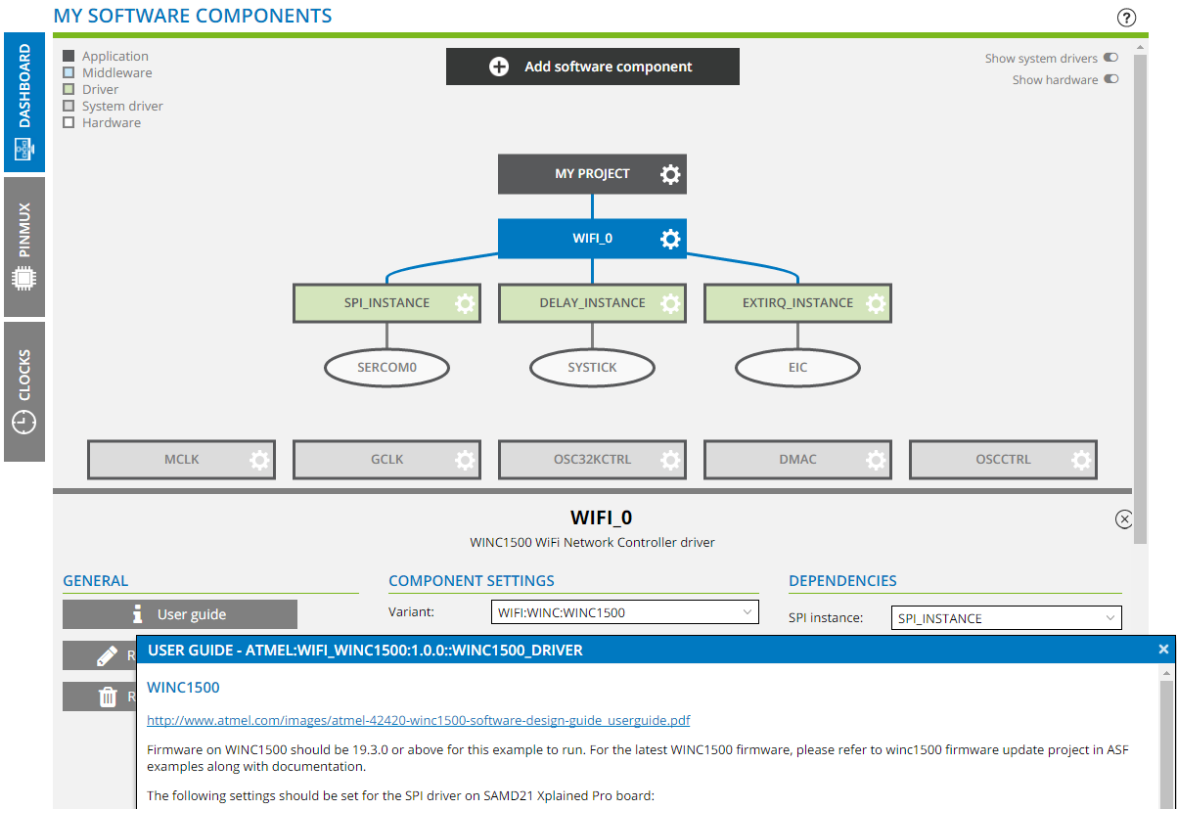
The Create New Project screen will show up as shown in the figure below. In the **FILTERS** section, when adding **MIDDLEWARE** and **DRIVERS** to the project, the list of MCUs, which meet those requirements, is narrowed down. The **SEARCH FOR SOFTWARE** searches all software components, displaying relevant results for both Middleware and Drivers. Drivers refer to both MCU *peripheral drivers* or *component drivers*, written to support external components.



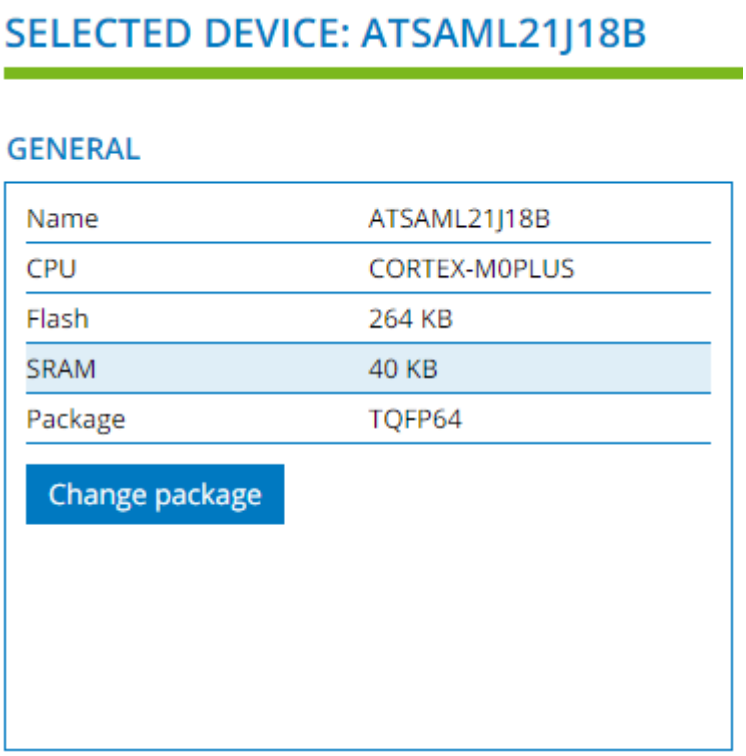
3. For example, by typing "WiFi" in the **SEARCH FOR SOFTWARE** bar and clicking on the Wi-Fi middleware, found by the search, you will see a description of the selected software component under **MORE INFORMATION**. Here we can see that this is a library supporting the *ATWINC1500 network controller SoC*, through an *SPI/USART* interface.
4. Incrementing the Wi-Fi middleware count to 1, now filters devices, showing only those MCUs and boards which are able to work with the ATWINC1500 Wi-Fi SoC. To see which kits support a Wi-Fi extension, mark “Show only boards” in the RESULTS window. The image below shows the result at this stage. Click on **CREATE NEW PROJECT**.



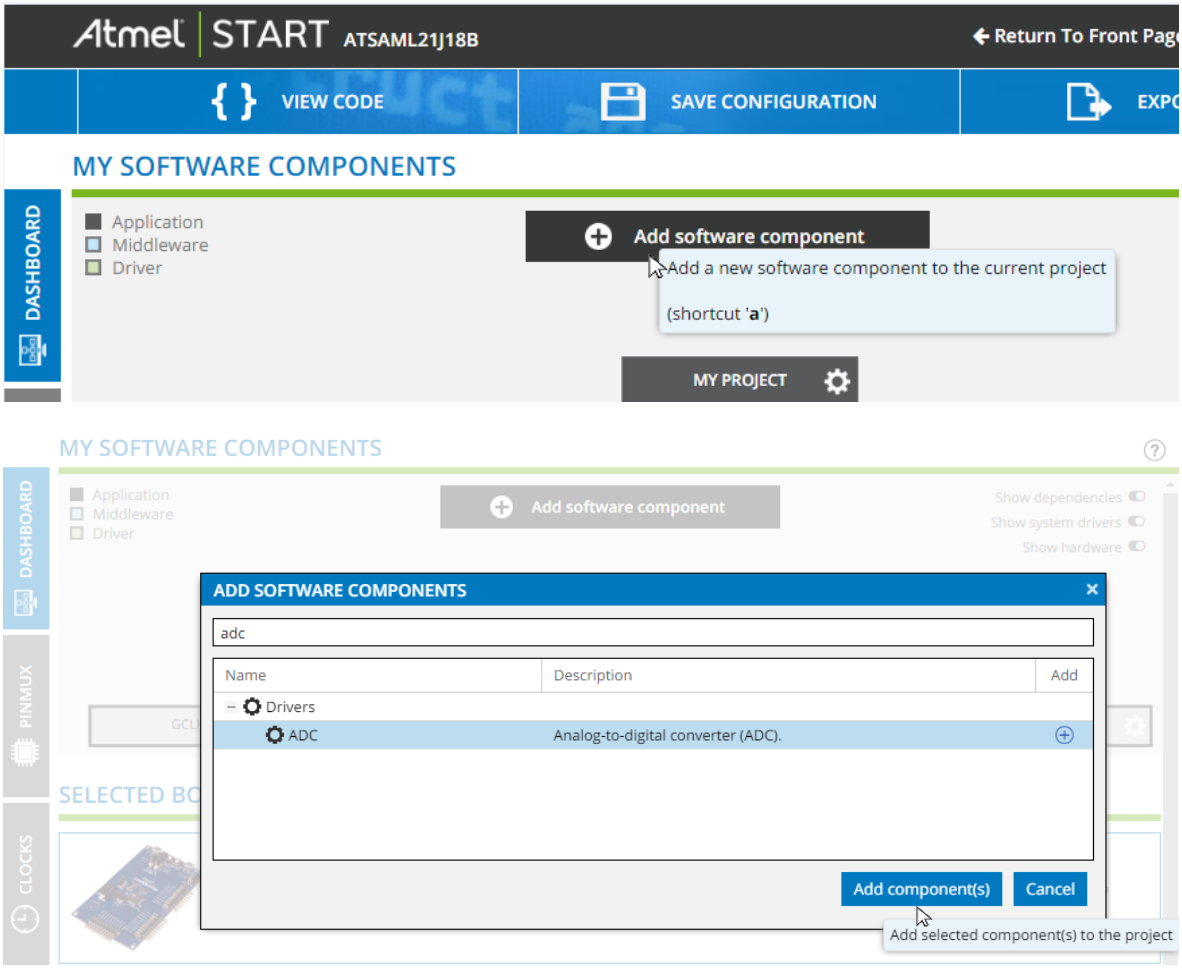
The project is created in Atmel START and opens in the **DASHBOARD** tab, as shown below. Note that the Wi-Fi middleware library for the ATWINC1500 module has already been added to the project, along with related driver dependencies. For more information on the Wi-Fi middleware, click on **User Guide**. Here you will find a link to the primary programming reference for the ATWINC1500, the *ATWINC1500 Software Design Guide*.



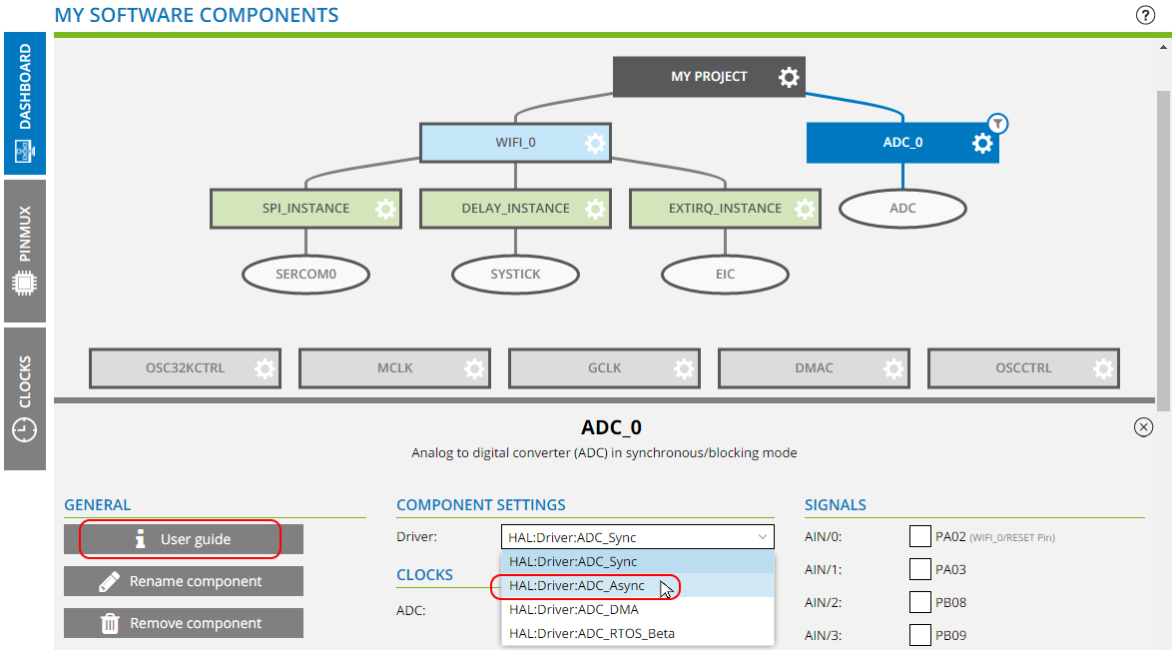
Note: You can check that choosing the SAM L21 Xplained PRO board automatically selects the ATSAML21J18B (TQFP64) as device, which is the one mounted on it.



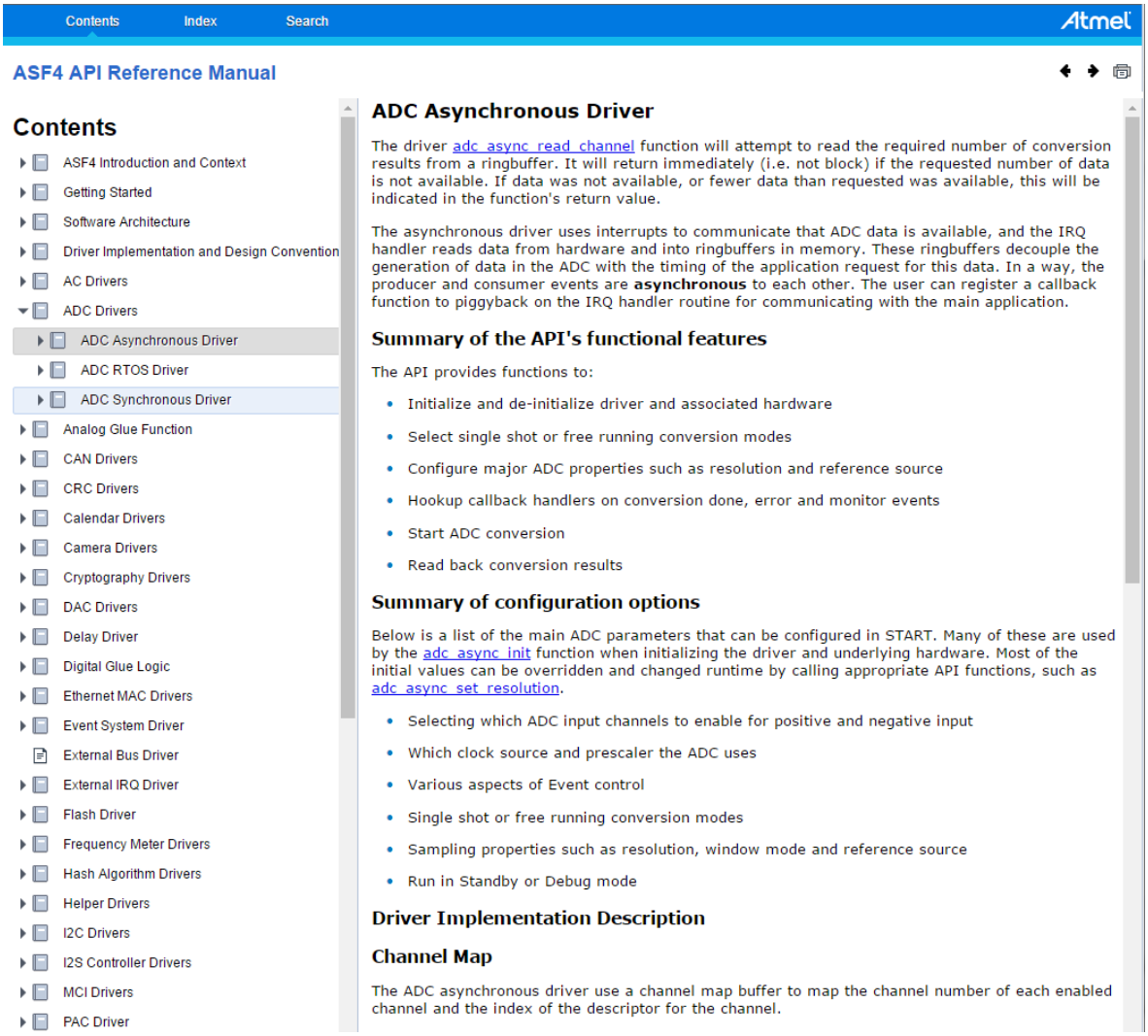
5. Now, click on **Add software component** and search for "ADC". Then, select the *ADC driver* from the list and click on **Add component(s)**.



6. The ADC driver or *Wi-Fi middleware* library is now added to the project dashboard. The Driver drop-down menu lets you select the driver type. Click on the ADC_Async option.



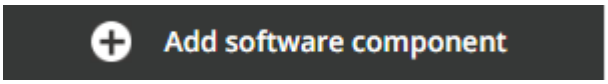
7. Clicking on **User guide** will now open an HTML version of the ASF4 API Reference Manual, in context of the ADC_Async API reference, as shown below.



3.3 How to Add and Configure Software Components

3.3.1 Adding a Software Component

1. In an Atmel START project, select **DASHBOARD** and click on **ADD SOFTWARE COMPONENT**:



2. Add a USART driver by typing *USART* in the "Filter" field, then add it to your project by clicking the plus sign, or by double-clicking on the line.

Tip: You can also look for it in the Drivers list.

You will have it displayed in the SELECTED COMPONENTS view:

3. You can now complete the addition of the USART driver by clicking on the **Add component(s)** button.
   The USART driver is added to the application.

3.3.2 How to Configure a Software Component

Software components can be added and removed from the Dashboard after a project is created.

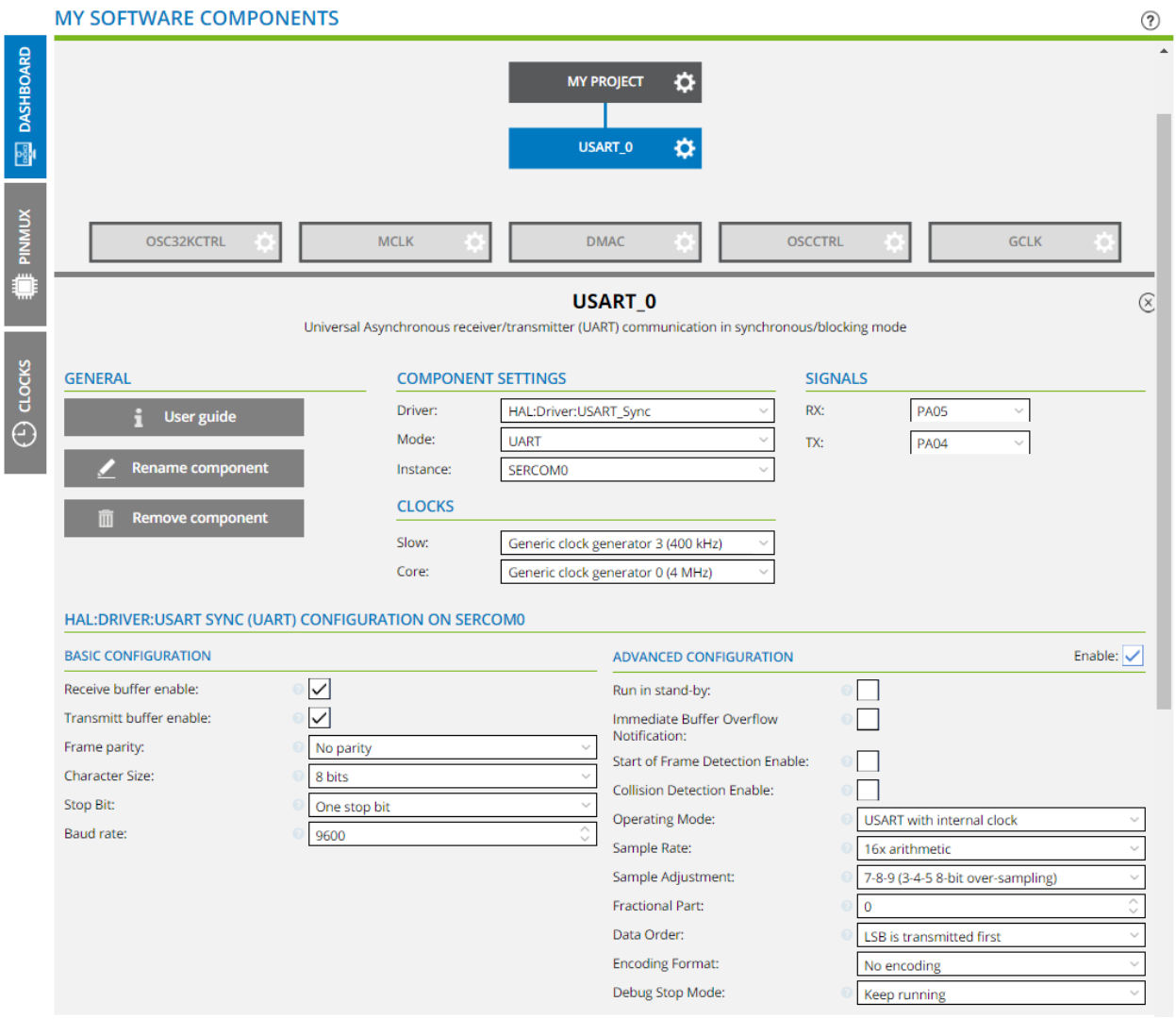
To configure a software component, go to **MY SOFTWARE COMPONENTS** in the **DASHBOARD** tab. Hovering over a component will give an indication of dependencies to other software components. Clicking on a component will open the software component's editor. From here you can:

- Remove or rename the component
- Open a user guide if available
- Select drivers, mode, and instance if available
- Select signals (if available) and resolve possible conflicts with other components
- Configure parameters specific for the selected software component

Basic configuration is common functionality or parameters that are available on any implementation of the peripheral, for example, common to all ADC hardware peripherals or USART hardware peripherals.

Basic mode: Using field names, drop-down options, and hover-over text, an embedded developer should be able to intuitively select an appropriate configuration.

Advanced configuration exposes hardware-specific configuration parameters, with unique or differentiated functionality. Using this functionality means that more work may be required to move the code to a different architecture. On common MCU platforms, such as the Microchip M0+, even the advanced configuration may be portable between D21 and L21.



3.4 Save and Restore Configurations

You can save your configuration by clicking on **Save Configuration** from the **DASHBOARD**. From here you can download a configuration file, which you can upload by using the restore project option on the front page. A configuration file is included in all packs that are generated and downloaded. You can also upload the entire pack by using the same restore project option.

1. Create a new project and configure it.
2. Go to **Save Configuration**. Give the file an appropriate name and click **Download Configuration**.

At a later stage, when you want to reconfigure your session:

1. Go to the Atmel START front page.
2. In the **Load Existing Project** section on the front page, click the **Load Project From File** button.
3. Click **Browse** and select your saved configuration. Click **Open Selected File**.

The project should now load and you can pick up from where you left off.

4. Configuration Screens

Atmel START offers a flexible array of configuration screens, in order to set up various aspects of your design.

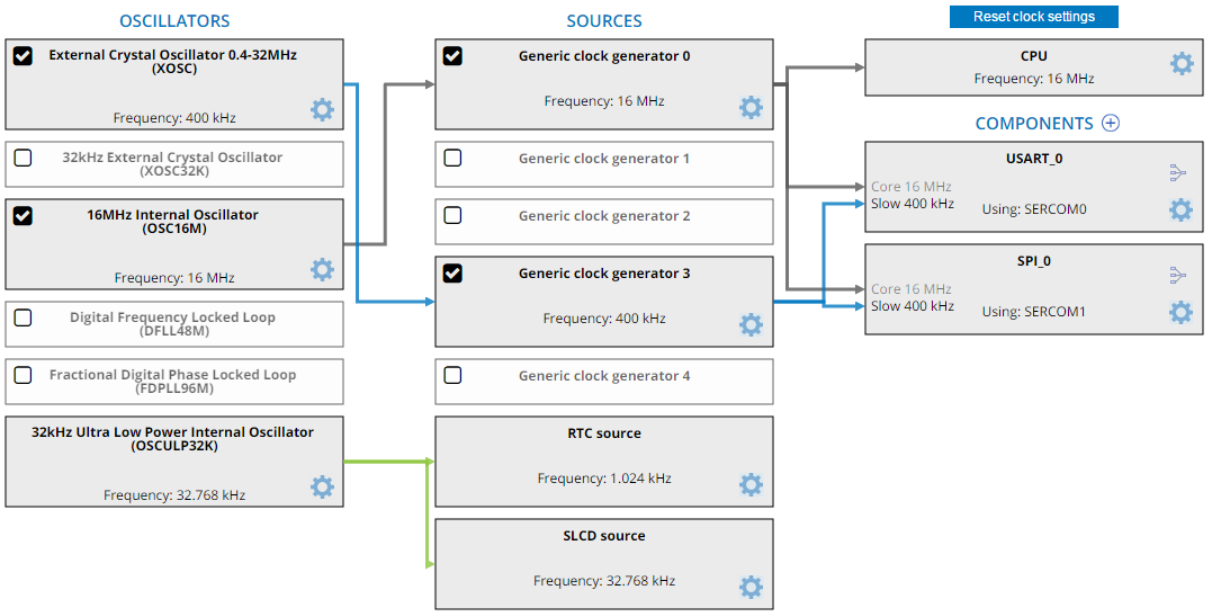
- PINMUX Configurator: The PINMUX Configurator presents an overview of assigned project pins, as well as configure GPIO pins.
- Clock Configurator: The clock system consists of oscillators and sources of different types. By using the Clock Configuration you can configure each source and oscillator, set prescalers and see the calculated output frequency.
- Event System Configurator: The Event System Configurator provides simple drag-and-drop functionality to connect event generators, channels, and event users.
- QTouch Configurator: QTouch composer functionality, is now brought into Atmel START, for configuration of QTouch buttons. Sliders and wheels will soon be available.

4.1 Clock Configurator

The clock system consists of oscillators and sources of different types. By using the Clock Configurator you can configure each source and oscillator, and see the calculated output frequency.

The view also lets you select the correct clock source for the selected components.

- The OSCILLATORS section displays the oscillators available for your selected device. Most clock sources will have a fixed frequency, while others may be specified either directly or by selecting dependent input signal. Open the settings dialog to configure the oscillator parameters.
- The SOURCE section lets you configure clock frequency by selecting input signal and changing multiplier and/or divider, if available
- The COMPONENTS section will list module instances added to your configuration. This table allows you to select the input signal and see the resulting frequency.
- The connection from disabled clocks are displayed by using dotted lines



Not all devices have check boxes to enable/disable Sources. These must be set by accessing the Settings Dialog. Oscillators with no check box are always enabled.

How to use the Clock Configurator

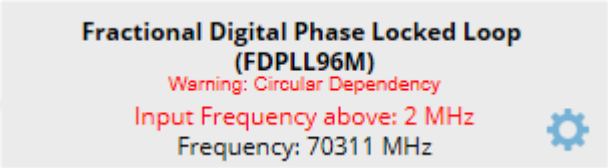
- A default configuration is provided. Most components use Generic clock generator 0.
- Set up connections by drag and drop between boxes or open settings dialog to select the desired input
- Access the settings dialog by clicking on the gear icon on each box. The clock must first be enabled.
- Display individual clock paths by clicking on the box
- Enable disabled clocks and clock paths by clicking on the disabled icon. Tooltips on icon provide more info.
- Reset all clock settings

Enabling and Disabling

When creating a project many of the oscillators will initially be disabled. These have an empty check box in the lower left corner. Sources and components using a disabled oscillator will have a warning sign. To enable or disable a Source, use the check box if available or open the settings menu and check the appropriate parameter.

Oscillator Source Input

Some oscillators may have a Source Generator as input. These will display a warning if the frequency is outside the defined limits. Hover over the text to display the tooltip showing source and offending frequency. A warning is also displayed if an oscillator is an input to a Source that is input to the same oscillator, creating a circular dependency.

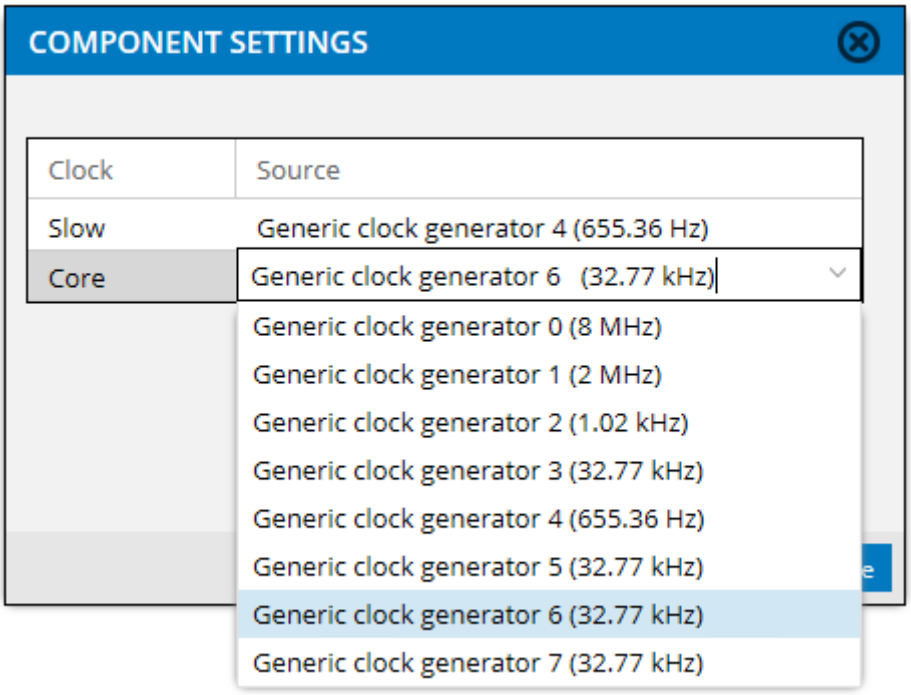


Multiple Clocks

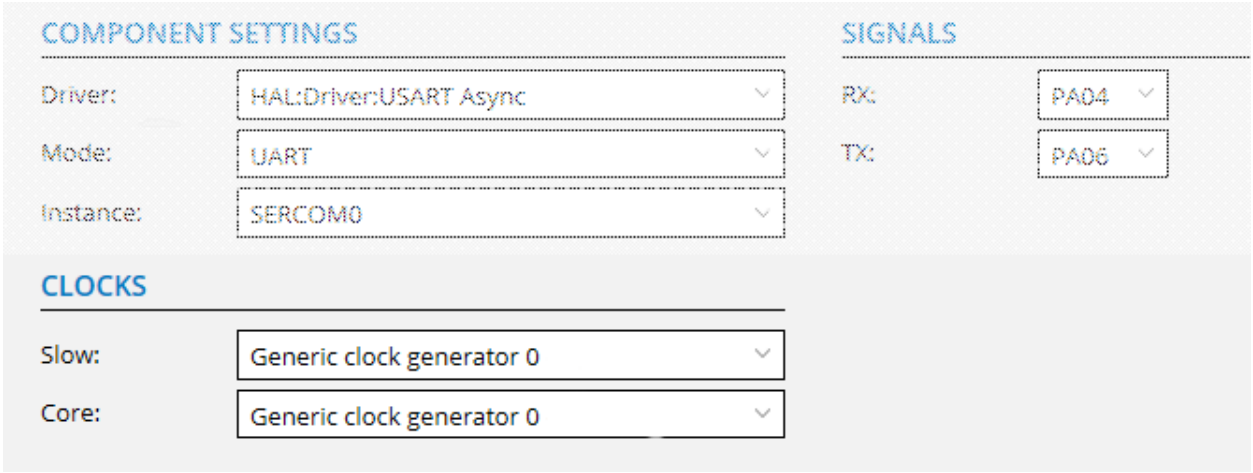
Some components have more than one clock. If this is the case, the clock used for drag and drop has a darker font. To switch, click on the name of the clock that should be used in drag and drop.

Selecting Component Clock Sources

Clock Sources can be set from the Component Settings Dialog:

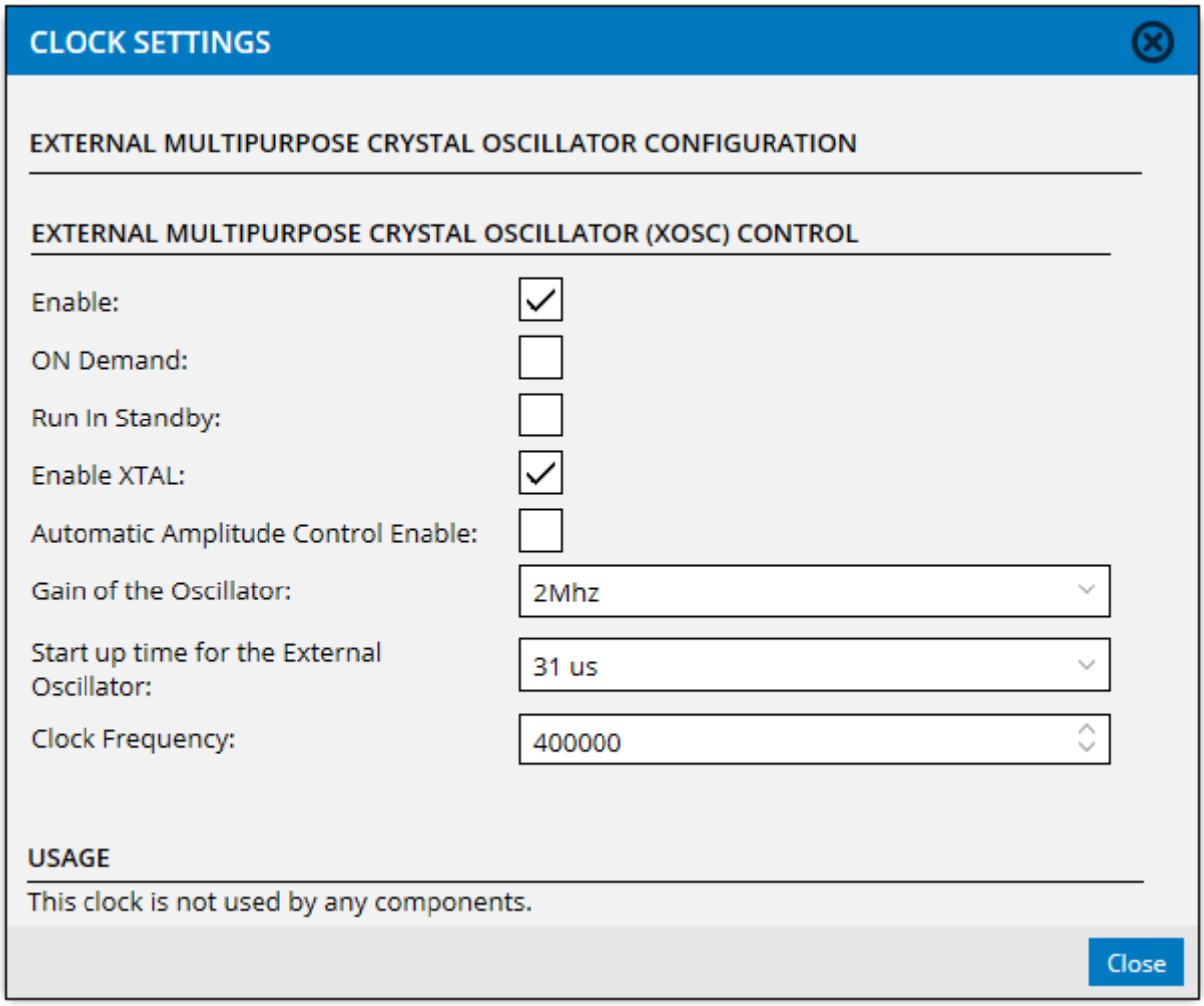


Clock Sources can also be selected from the Component Editor:



Clock Settings

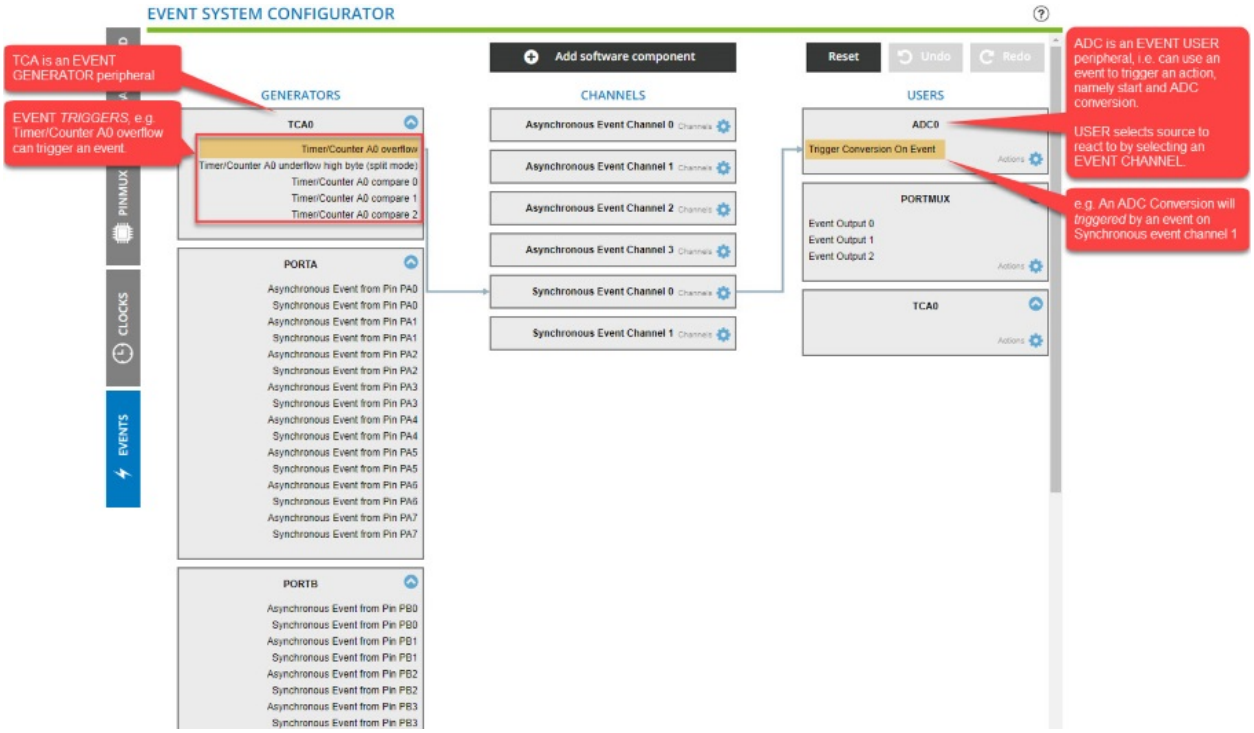
The Oscillator and Source Settings Dialog will display settings defined by the item selected. A tooltip is provided to help with the options. Check the "Enable" box in the upper corner to enable editing of options.



4.2 Event System Configurator

The event system enables direct peripheral-to-peripheral communication and signaling. The event system consists of generators, channels, and users. A change in the peripheral's state is referred to as an event.

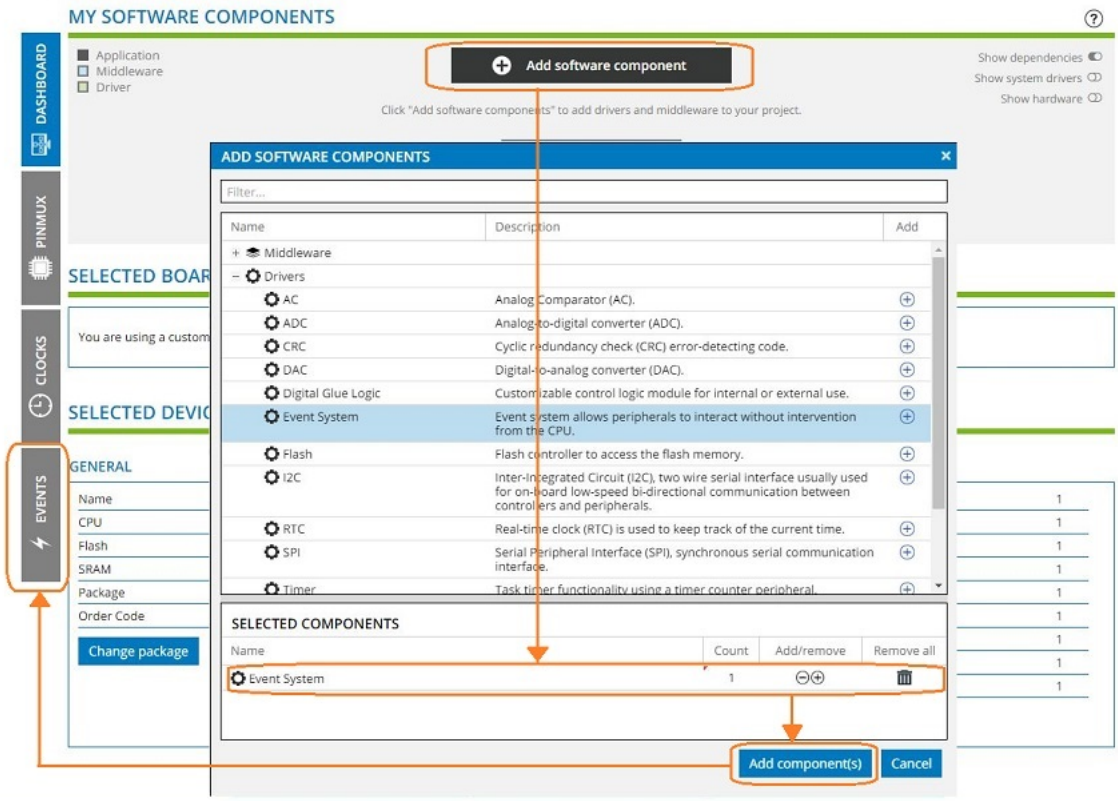
An event that takes place in one peripheral (say source) can be used to trigger another event in a different peripheral (destination). The term generator refers to the event of the source peripheral and the term user refers to the event of the destination peripheral. The generator and the source are connected via an event channel. The channel section lets you configure available channels for a particular event generator source. There are two types of event channels; Synchronous and Asynchronous. Asynchronous event control enables the event input to qualify the output directly. For more details on event channels, refer the hardware manual. This document explains the event system with an example of a timer event (event generator) triggering the ADC conversion (event user) through a synchronous channel.



4.2.1 Concepts and Principles

A typical example is to have a timer (overflow or compare match) event to start an ADC conversion. In this example, the timer (TCA) is the source and ADC is the action. Both will be connected via an event channel.

1. Create a new project using the ATtiny817.
2. Click on **Add software component** to add the event system driver.
   A tab named **EVENTS** appears on the left side of the screen.

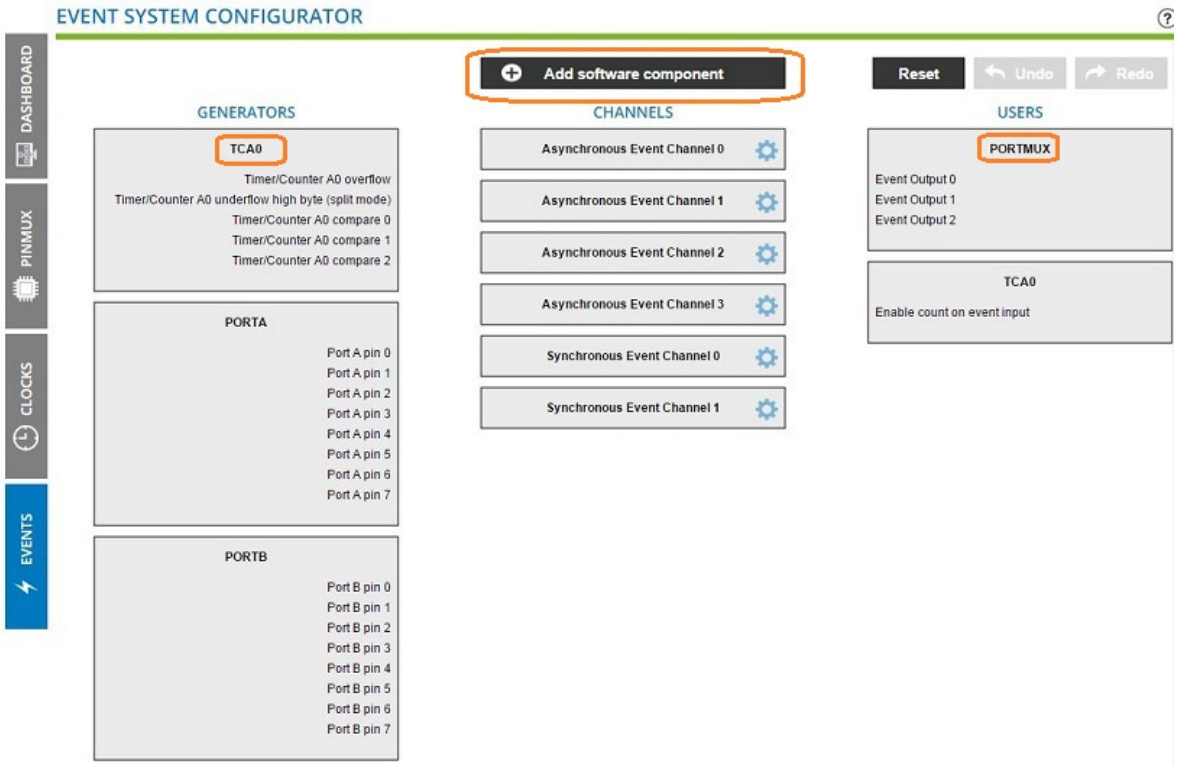


3. Click on the **EVENTS** tab.
   The GENERATORS and the USERS from the system drivers will appear by default, such as Port and PORTMUX.

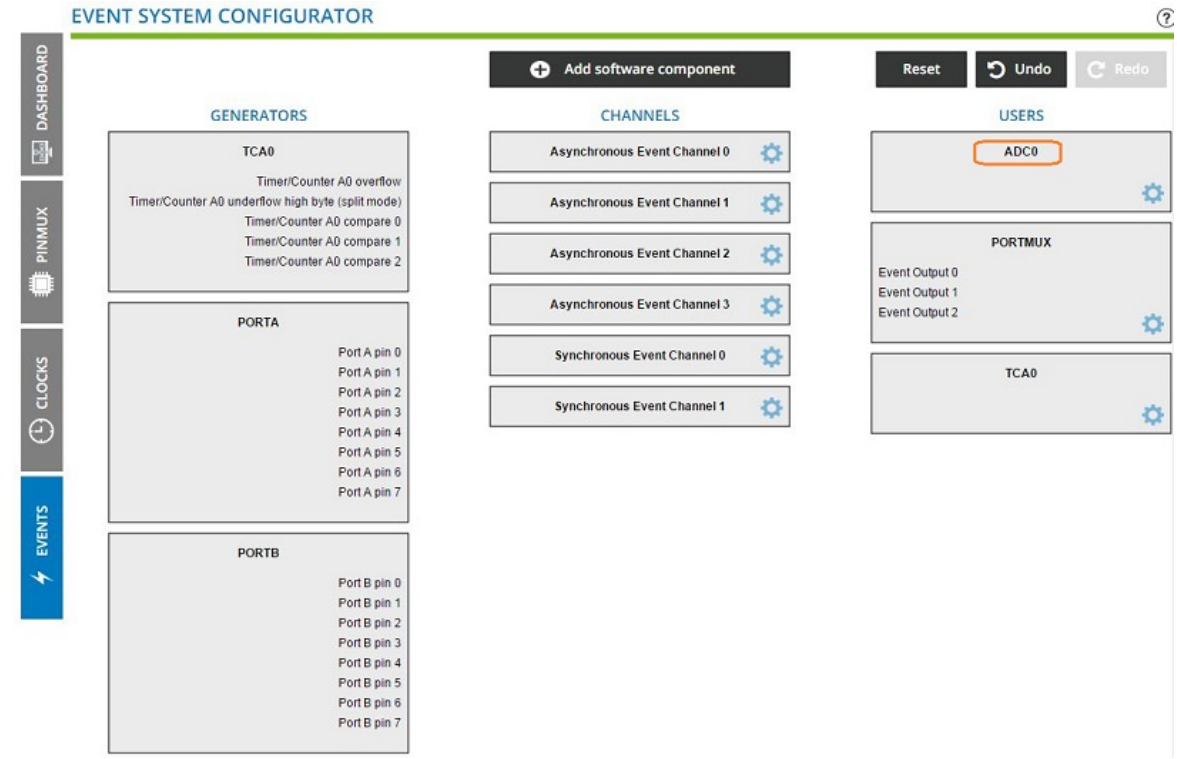
4.2.2 Overflow Timer Event Starts an ADC Conversion

In this example, we have assumed that only the event system driver was selected earlier.

1. Click on **Add software component** to add the Timer driver.
   The TCA0 is added as a generator and a user. The overflow event is listed as a generator.



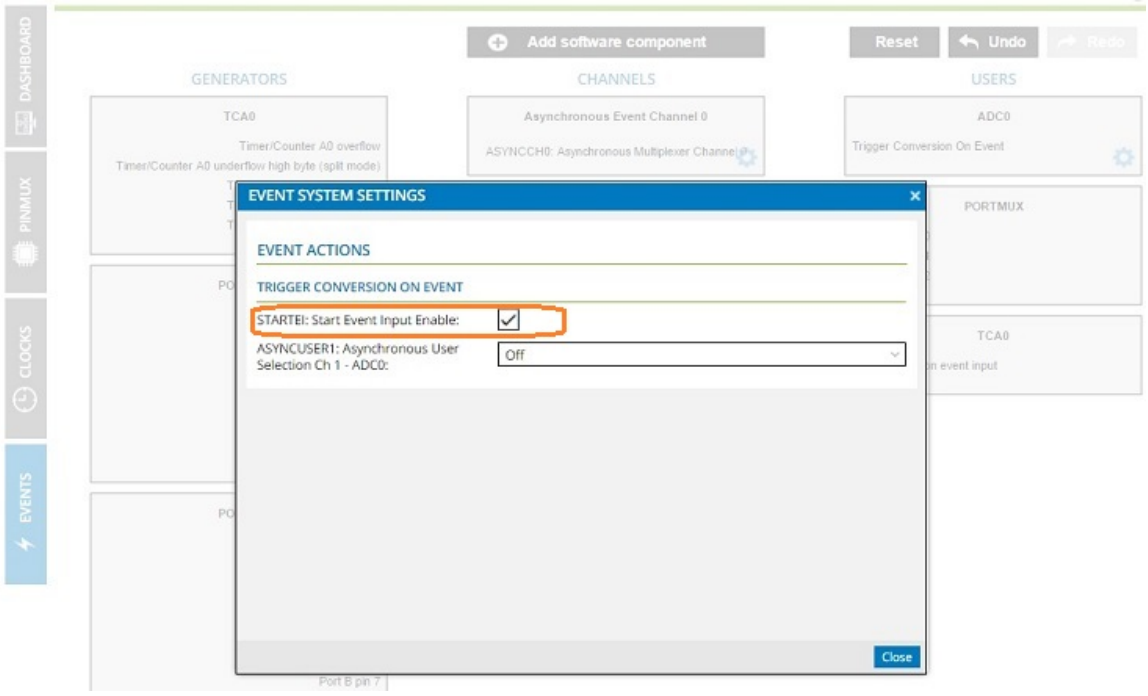
2. Click on **Add software component** to add the ADC driver.



The ADC is added as a user.

3. Start by dragging the TCA event generator (overflow) onto an Asynchronous Event Channel 0, then try dragging this event channel onto the ADC. This does not work because we have not yet enabled

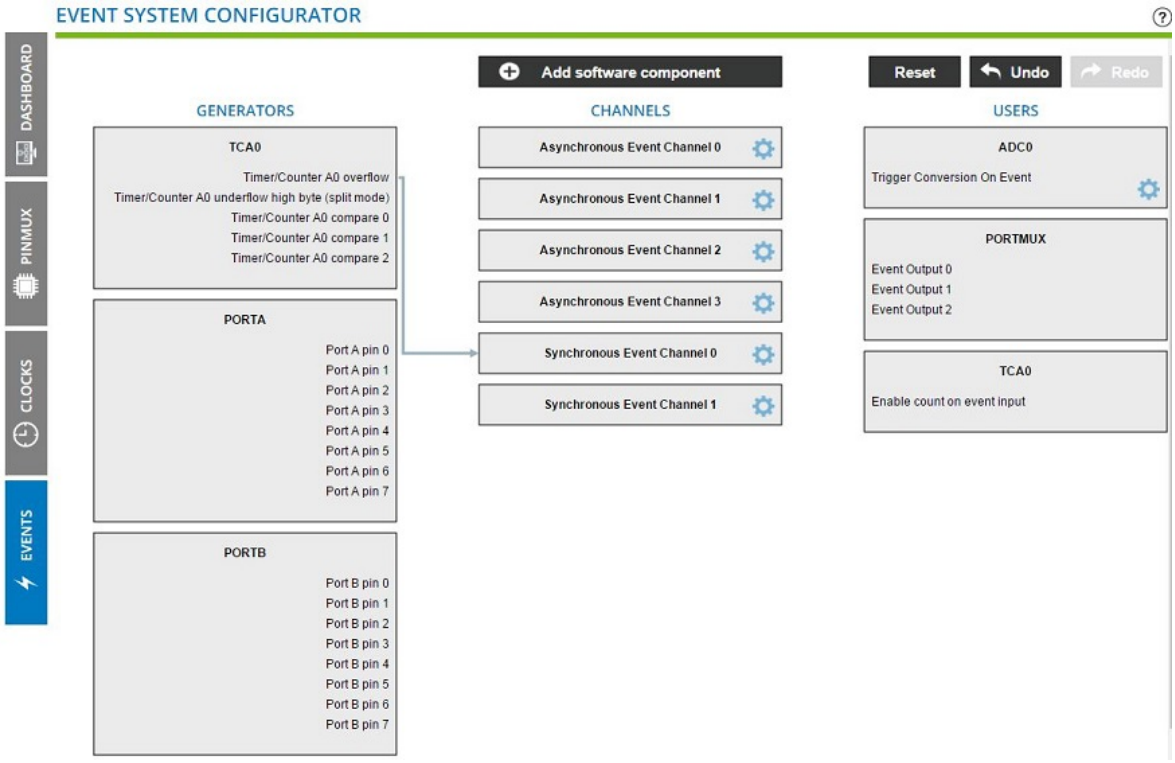
STARTEI (START Event Input Enable) for the ADC. Click on the ADC0 configuration wheel, then check "STARTEI" and close. You will now see that “Trigger Conversion on Event” is present in the ADC0 as a specific event user. Now you can drag the event channel onto this event (note: ADC0 should be grayed out when no specific event users have been enabled).



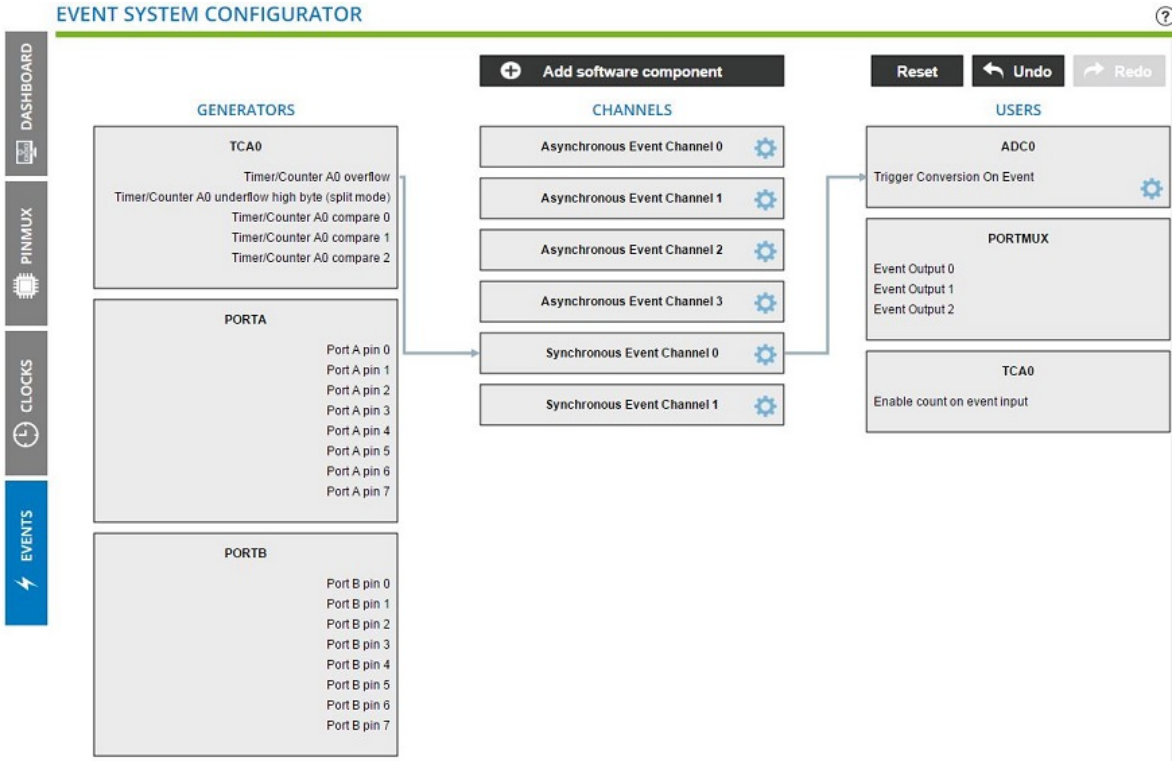
4. Next, connect the TCA event with the ADC: Click and drag the **Timer/Counter A0 overflow** to the **Synchronous Event Channel 0** channel.

   During drag and drop, only supported channels are highlighted.

   A connection is made between TCA and the event channel.



5. Click and drag the **Synchronous Event Channel 0** to the ADC0.
   A connection is made between the event channel and the Trigger Conversion On Event.



This will result in the following code:

```
EVSYS.ASYNCUSER1 = EVSYS_ASYNCUSER1_SYNCCH0_gc /* Synchronous Event Channel 0 */ ;
EVSYS.SYNCCH0 = EVSYS_SYNCCH0_TCA0_OVF_LUNF_gc /* Timer/Counter A0 overflow */;
```

The event input is enabled in `adc.c`.

```
ADC0.EVCTRL = 1 << ADC_STARTEI_bp /* Start Event Input Enable: enabled */;
```

4.2.3 Event System Configurator

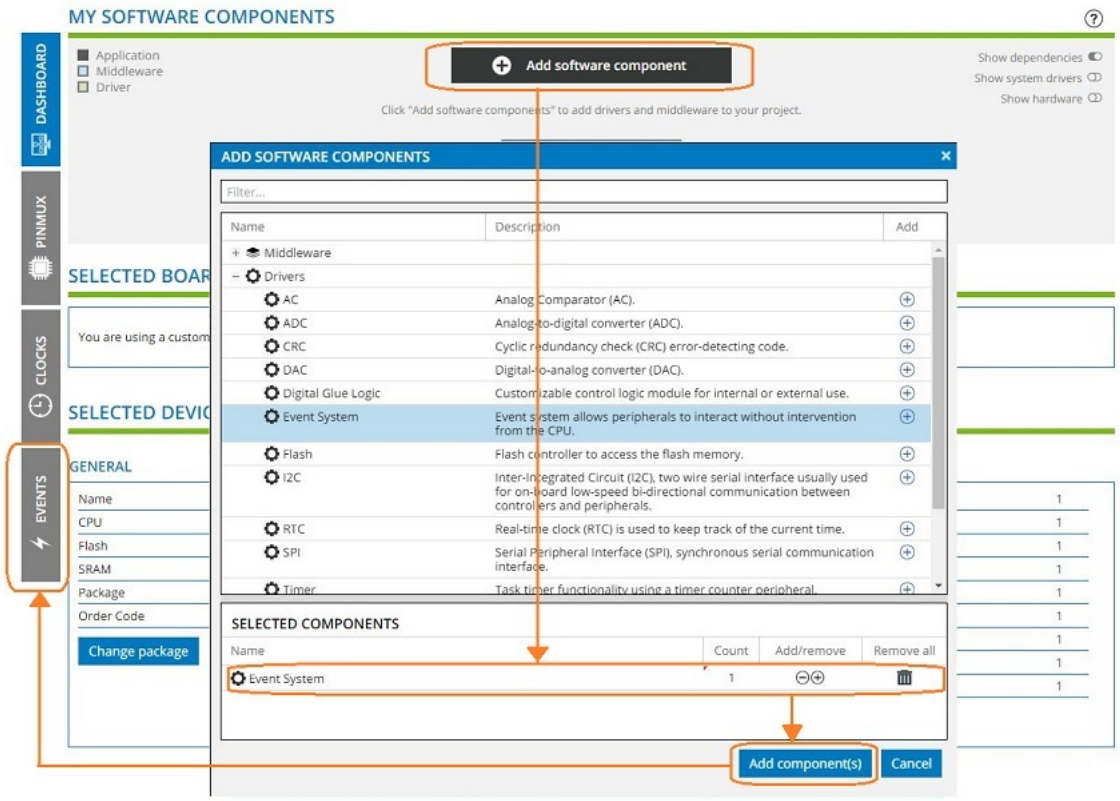
The event system enables direct peripheral-to-peripheral communication and signaling. The event system consists of generators, channels, and users. A change in the peripheral's state is referred to as an event.

An event that takes place in one peripheral (say source) can be used to trigger another event in a different peripheral (destination). The term generator refers to the event of the source peripheral and the term user refers to the event of the destination peripheral. The generator and the source are connected via an event channel. The channel section lets you configure available channels for a particular event generator source. There are two types of event channels; Synchronous and Asynchronous. Asynchronous event control enables the event input to qualify the output directly. For more details on event channels, refer the hardware manual. This document explains the event system with an example of a timer event (event generator) triggering the ADC conversion (event user) through a synchronous channel.

4.2.3.1 How to use the Event System Configurator

A typical example is to have a timer (overflow or compare match) event to start an ADC conversion. In this example, the timer (TCA) is the source, and ADC is the action. Both will be connected via an event channel.

1. Create a new project using the ATtiny817.
2. Click on **Add software component** to add the event system driver.
   A tab named **EVENTS** appears on the left side of the screen.

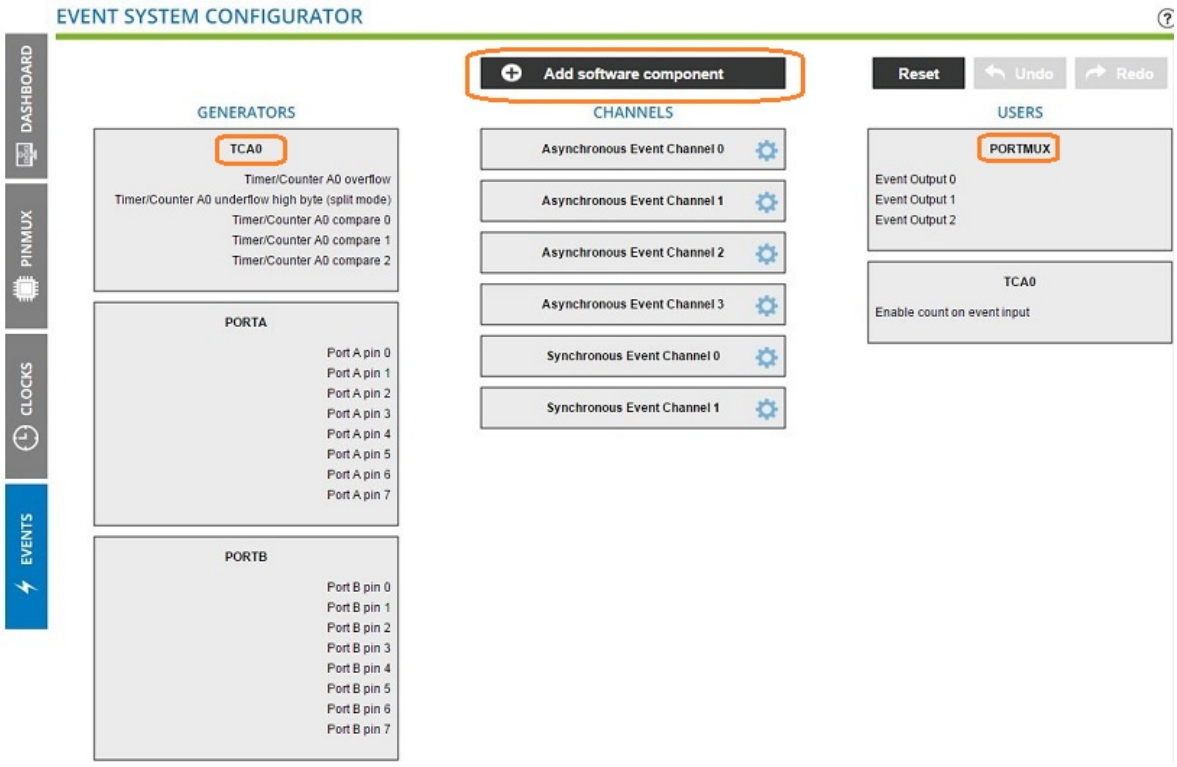


3. Click on the **EVENTS** tab.
   The GENERATORS and the USERS from the system drivers will appear by default, such as Port and PORTMUX.

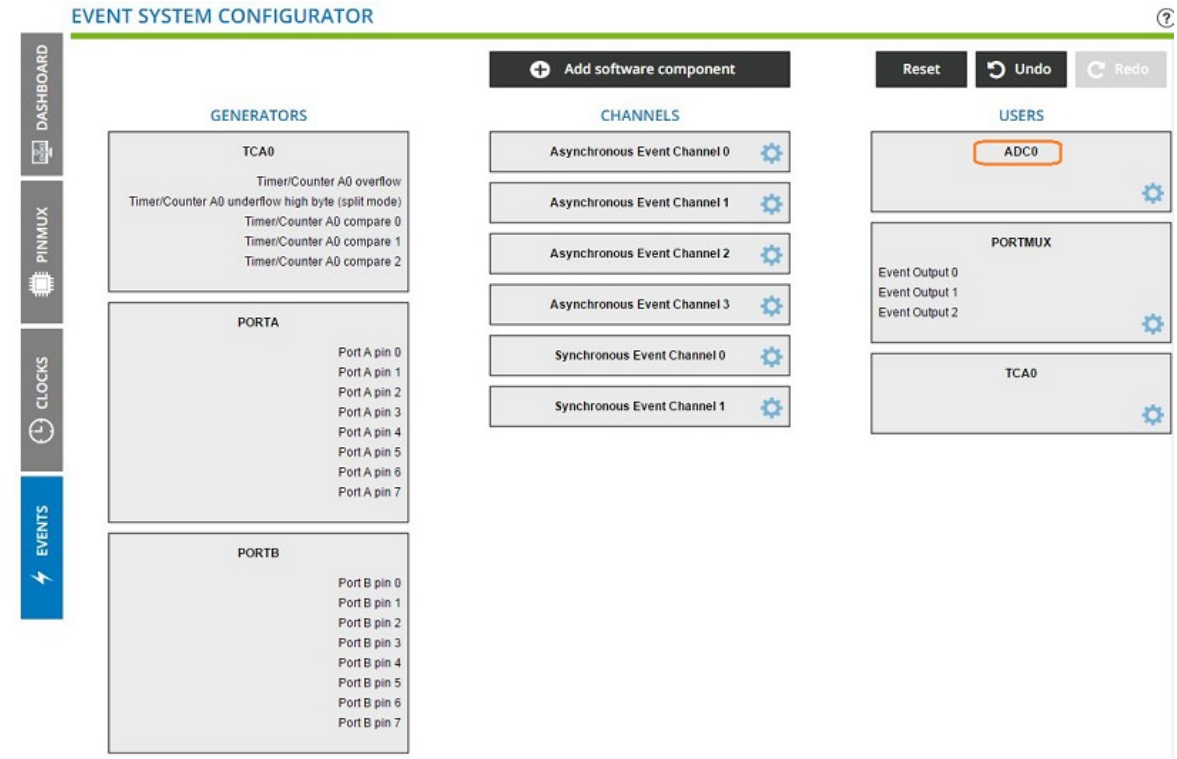
4.2.3.2 Overflow Timer Event Starts an ADC Conversion

In this example, we have assumed that only the event system driver was selected earlier.

1. Click on **Add software component** to add the Timer driver.
   The TCA0 is added as a generator and a user. The overflow event is listed as a generator.



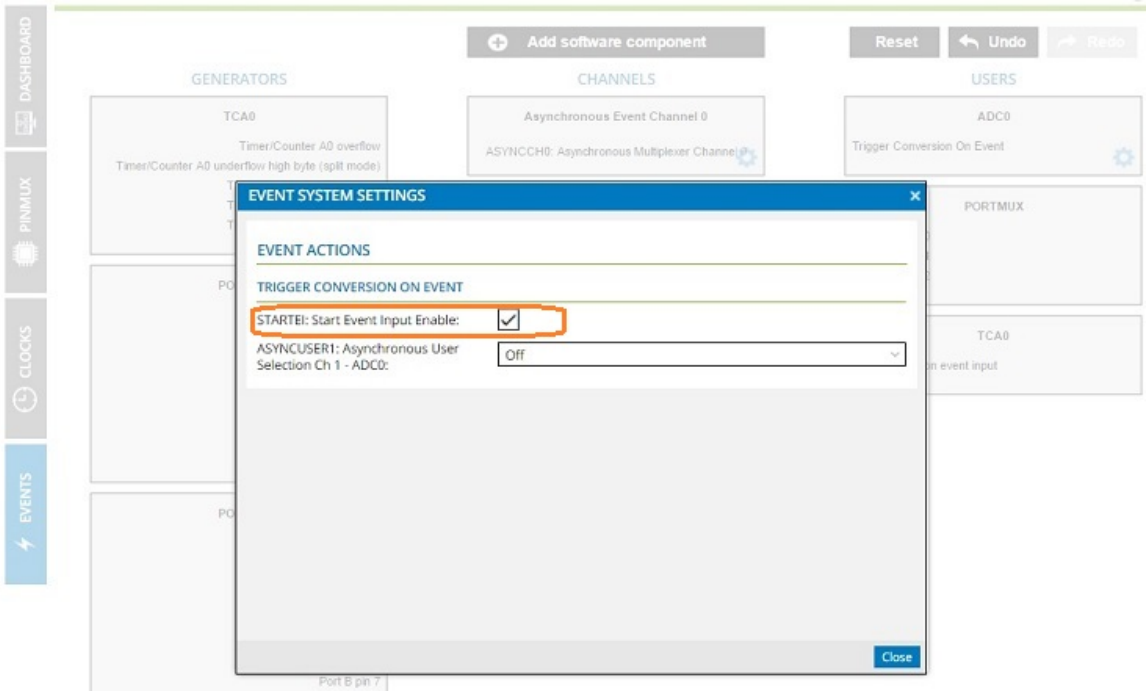
2. Click on **Add software component** to add the ADC driver.



The ADC is added as a user.

3. Start by dragging the TCA event generator (overflow) onto an Asynchronous Event Channel 0, then try dragging this event channel onto the ADC. This does not work because we have not yet enabled

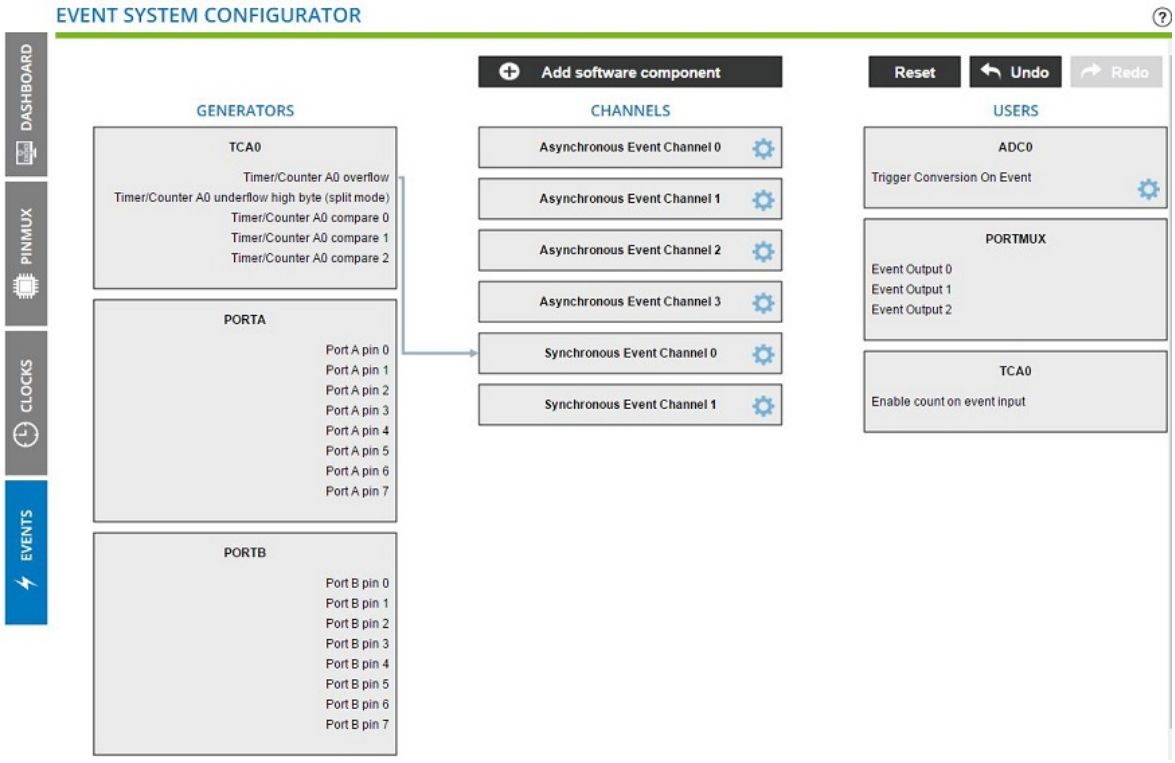
STARTEI (START Event Input Enable) for the ADC. Click on the ADC0 configuration wheel, then check "STARTEI" and close. You will now see that “Trigger Conversion on Event” is present in the ADC0 as a specific event user. Now you can drag the event channel onto this event (note: ADC0 should be grayed out when no specific event users have been enabled).



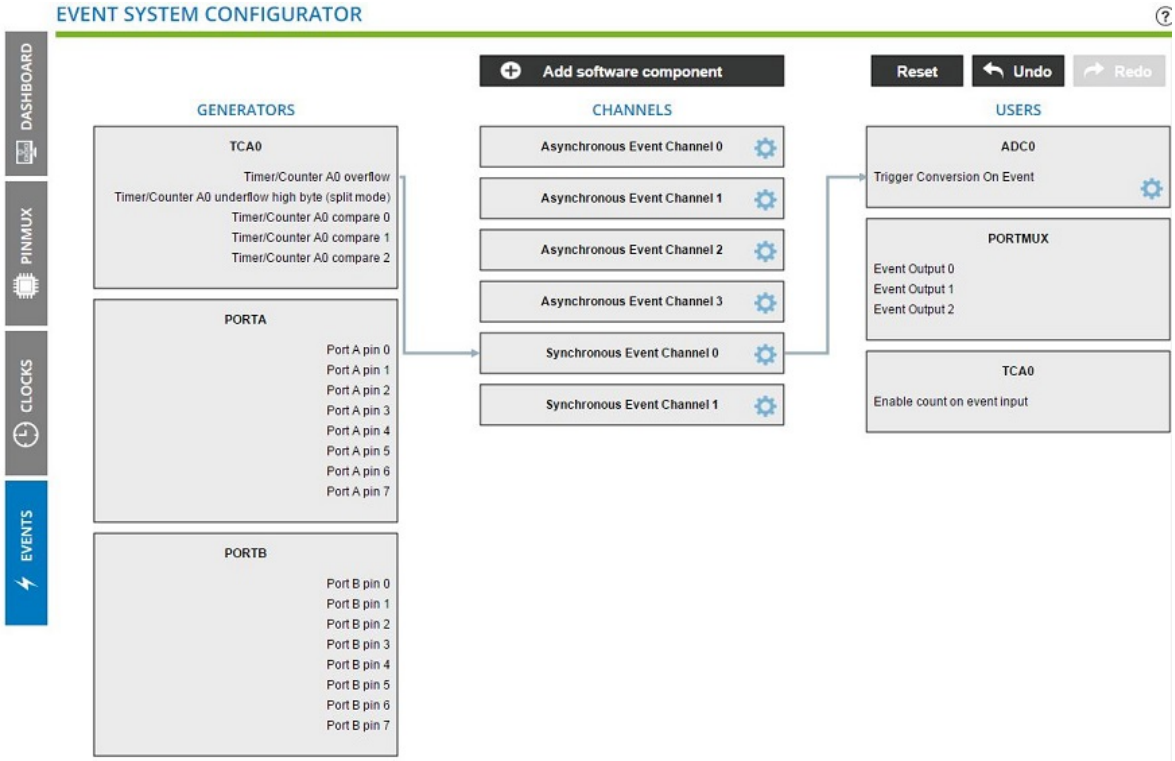
4. Next, connect the TCA event with the ADC: Click and drag the **Timer/Counter A0 overflow** to the **Synchronous Event Channel 0** channel.

   During drag and drop, only supported channels are highlighted.

   A connection is made between TCA and the event channel.



5. Click and drag the **Synchronous Event Channel 0** to the ADC0.
   A connection is made between the event channel and the **Trigger Conversion On Event**.



This will result in the following code:

```
EVSYS.ASYNCUSER1 = EVSYS_ASYNCUSER1_SYNCCH0_gc /* Synchronous Event Channel 0 */ ;
EVSYS.SYNCCH0 = EVSYS_SYNCCH0_TCA0_OVF_LUNF_gc /* Timer/Counter A0 overflow */;
```

The event input is enabled in `adc.c`.

```
ADC0.EVCTRL = 1 << ADC_STARTEI_bp /* Start Event Input Enable: enabled */;
```

4.3 Atmel START QTouch® Capacitive Sensing Library

The Peripheral Touch Controller (PTC) is a capacitive touch sense module that supports autonomous sensing of 1 to 256 channels. The PTC supports both self and mutual capacitance sensor layouts in the same application, which provide great flexibility for the system designer. Due to the autonomous operation, the PTC uses very little CPU resources and power, even for high key count designs.

Key Features

- Supports buttons, sliders, and wheels. Up to 256 touch channels depending on the package.
- Supports mutual and self-capacitive touch, even at the same time no external components are needed
- Self-calibrating, no tuning needed
- Autonomous operation for low CPU load
- Very low power consumption
- Excellent conducted immunity (CI) for operation in harsh environments
- Built-in water tolerance support
- Data Streaming of Surface data
- Lump and low-power support

QTouch® Library is a royalty-free software library for developing touch applications using AVR® and SAM devices with Peripheral Touch Controller. QTouch Configurator is an Atmel START-based touch example project builder, while Data Visualizer allows you to visualize sensor debug data and tune QTouch Library parameter settings. The following pages of this tutorial cover the steps needed to use QTouch Configurator and Data Visualizer tools when working with a QTouch Library touch example project.

4.3.1 QTouch® Configurator

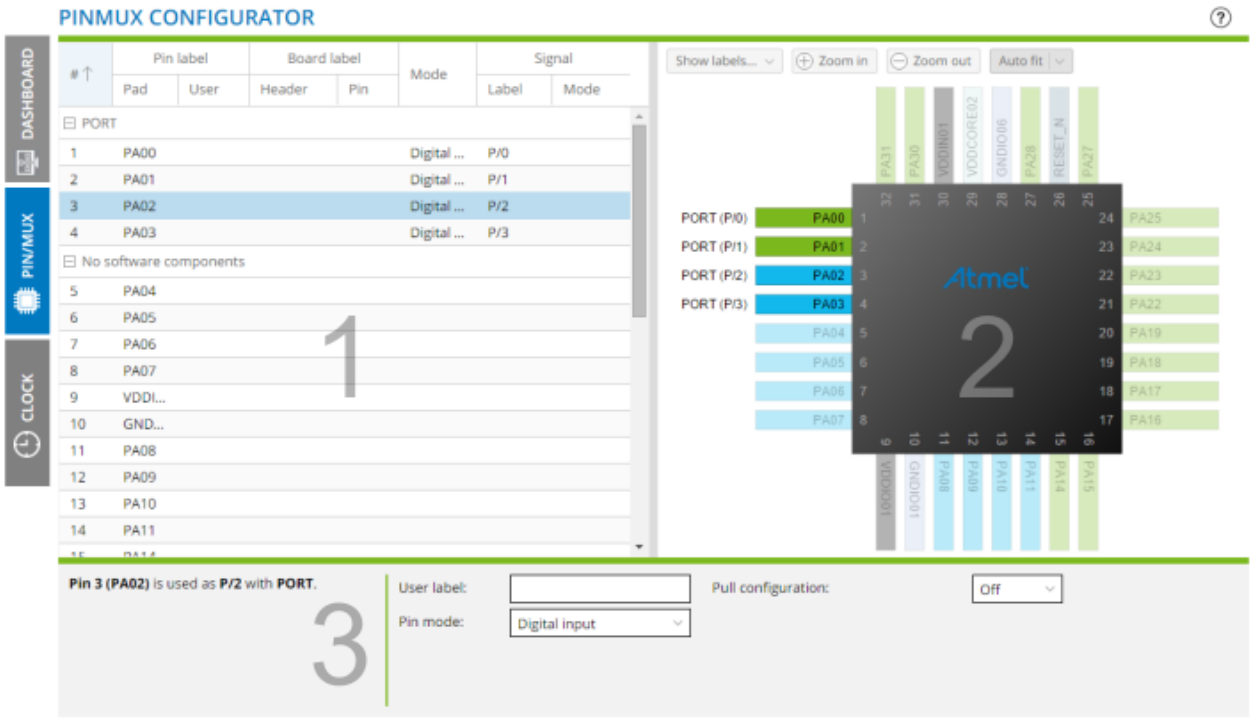
The QTouch Configurator provides the option to generate a QTouch project for custom user boards, as well as to download a QTouch project for standard evaluation boards.

- Steps to Generate User Board Touch Project
- Steps to Generate 2D Touch Surface and Gesture Project
- Steps to Download Microchip Board Project

4.4 PINMUX Configurator

The PINMUX Configurator presents an overview of all the configured pins.

Figure 4-1. PINMUX Configurator



1. Pin Assignment. 2. Device Layout. 3. Pin Details Editor.

From the **Pin Assignment** table you can:

- List pin number, pad, software component, mode, and signals
- Click on the column header to change the sort order
- Add columns by clicking on the header and select Columns
- Select a pin by clicking on the row, select more by holding the <SHIFT> or <CTRL> keys when clicking

From the **Device Layout** you can:

- Hover over a pin to highlight the corresponding pad in the Pin Assignment table
- Click on a pin to open the corresponding pad for editing in the Pin Assignment table
- Click in the center of the device to access web-based information for the device

The Pin Details Editor is used for setting properties for GPIO pins. Select one or more GPIO pins in the Pin Assignment table, and the Pin Details Editor will open at the bottom of the screen. Here you can assign a user label, select a pin mode, and configure the selected pin mode.

Tip:
You can select multiple pins at once by pressing the <SHIFT+CTRL> keyboard keys while clicking on the list. This allows you to configure multiple pins at once.

If the pins selected are of different types, they may not support the same pin modes. You will only be able to configure settings that are common to the selected pins.

4.5 CCL Configurator

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB and logic operations can be performed using Programmable Lookup Tables (LUT) and sequential logic without CPU intervention.

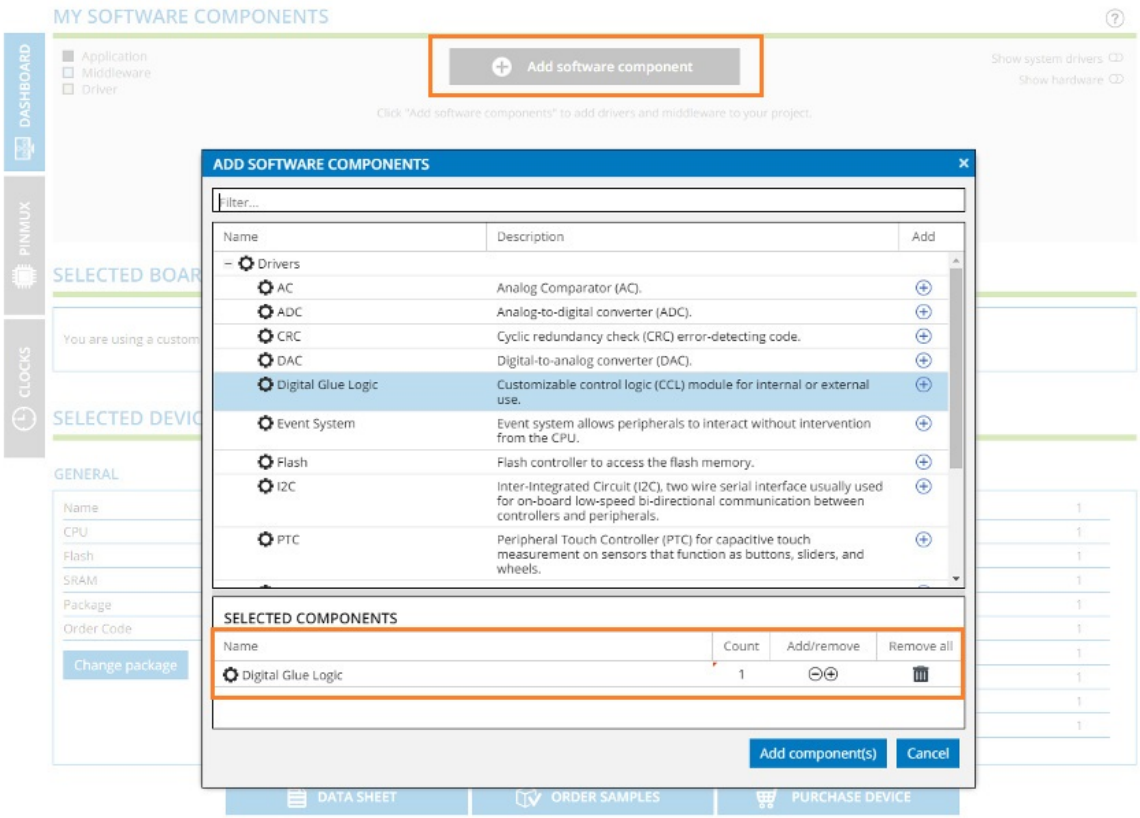
Each LUT consists of three individually configurable inputs and a truth table, in addition to the optional filter, synchronizer, and edge detector. Inputs can either be I/O pins, internal feedback, peripherals, or events from the Event System. Each LUT can generate an output as a user programmable logic expression from the three inputs. Two adjacent LUTs are connected to a sequential logic block, forming a unit. The optional sequential logic can be enabled for complex waveform generation such as JK flip-flop, RS latch, D latch, or gated D flip-flop. The output can be routed to I/O pins, peripherals, event system, or as feedback to the corresponding unit.

CCL Configurator supports end-to-end configuration of the CCL module graphically.

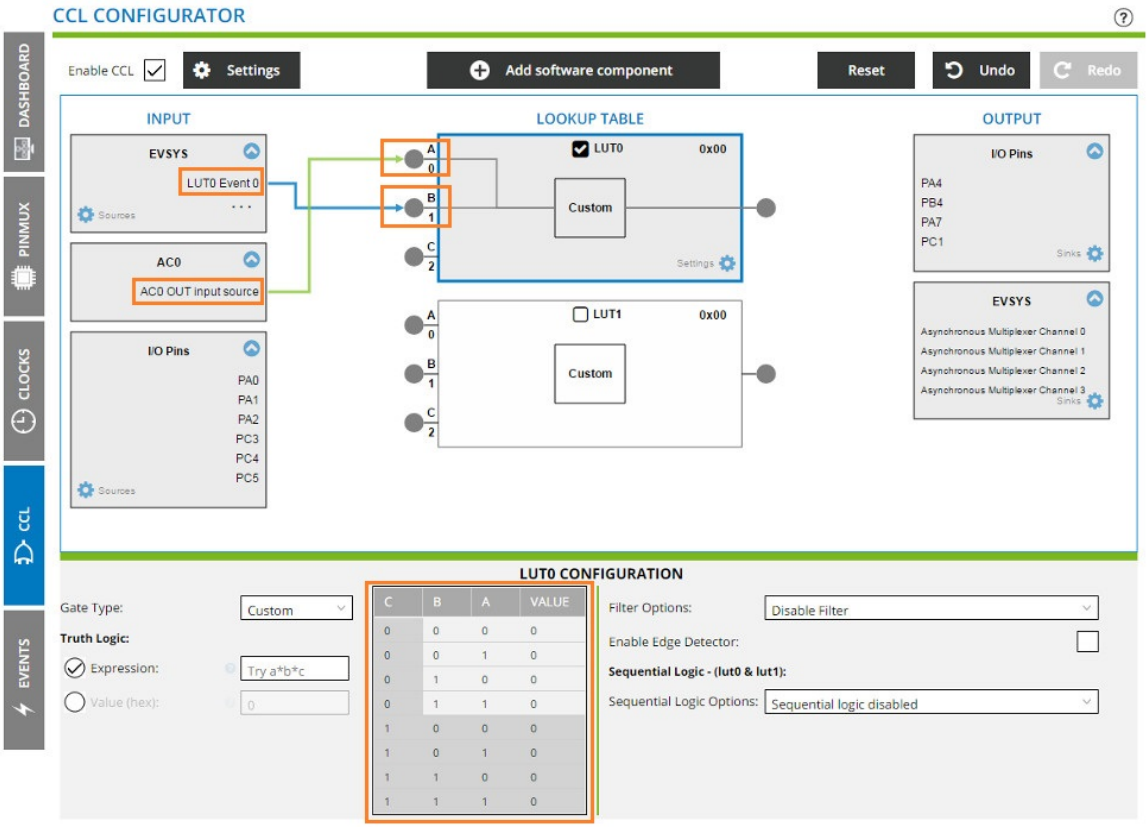
4.5.1 How to use CCL Configurator

The following application demonstrates the use of internal peripherals and I/O pins as inputs to the CCL. The CCL uses the LUTs and sequential modules to generate the output based on the inputs. The example application below detects motion in low-light condition and provides the output signal to a pin. The signal will be available in that pin until a switch is pressed to turn off the signal.

1. Create a new project using ATtiny817.
2. Click on **Add software component** and add the Digital Glue Logic driver.
   Once the **Digital Glue Logic driver** has been added, a tab named **CCL** appears on the left side of the screen.



3. Click again on **Add software component** and add AC and Event System drivers.
4. Configure the Analog Comparator module to provide the desired output *(light or dark)* based on its inputs. Use the Dashboard view to configure the AC module.
5. Configure the Event System (EVSYS) module to provide the desired output *(motion detected)* based on its inputs.
6. Go to the **CCL** tab and enable the first LUT (LUT 0). Drag and drop **AC0 OUT input source** to the first Input (A) of LUT0. Then drag and drop **LUT0 event 0** to the second input (B) of LUT0. Leave the third input disconnected.
   Truth table rows and columns are updated based on the selected inputs and invalid rows are disabled in the figure below.



7. We can configure the truth table by one of the following methods:
   1. Selecting one of the predefined logic gate types (`AND`, `NAND`, `OR`, `NOR`, `XOR`, `XNOR`)
   2. Entering an expression

      `AND` is represented using '`*`'. If an operator is not specified, `AND (*)` is assumed

      `OR` is represented using '`+`'

      `XOR` is represented using '`^`'

      `NOT` is represented using '`!`'

      `NAND` can be created by using the expression - `!(a*b)`. If an operator is not specified, `NAND !(*)` is assumed

      `NOR` can be created by using the expression - `!(a+b)`

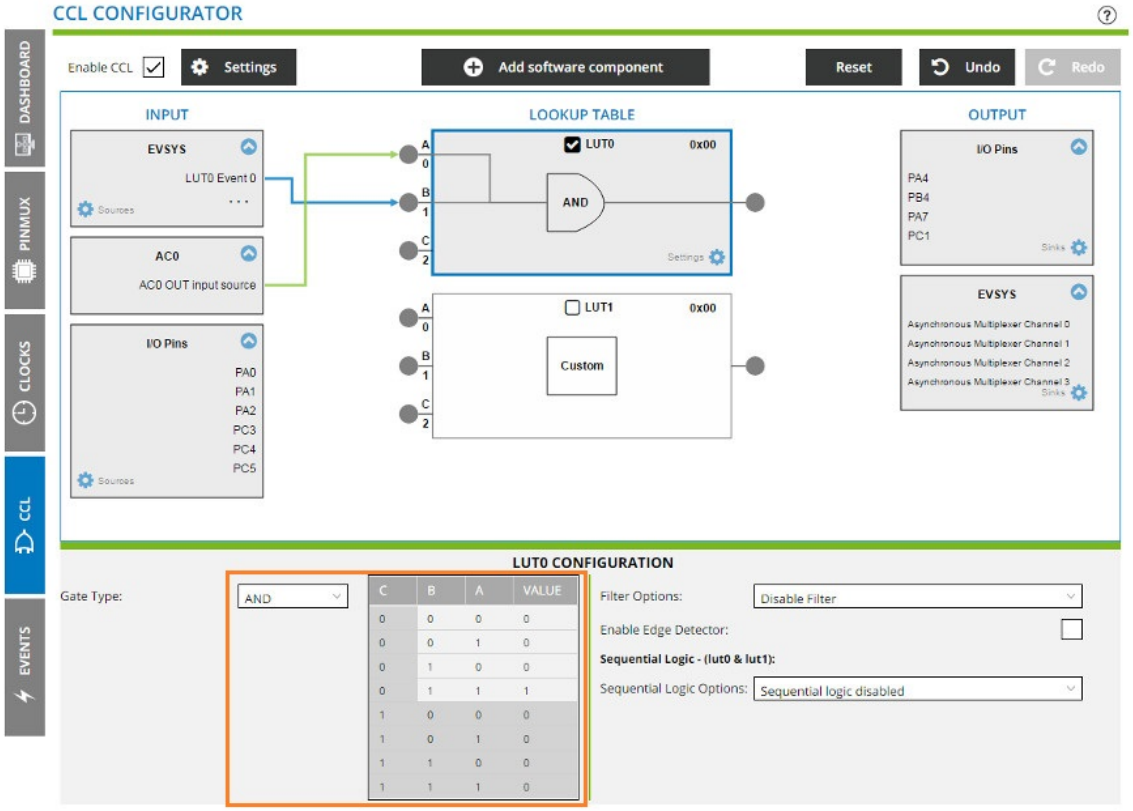
      `XNOR` can be created by using the expression - `!(a^b)`

      Complex combinational logic can be created using an expression that involves the operators listed above. Brackets can be used to control the precedence of operators by grouping the part of an expression. E.g. - `a+((b^c)*(!c))`
   3. Entering Value of truth logic

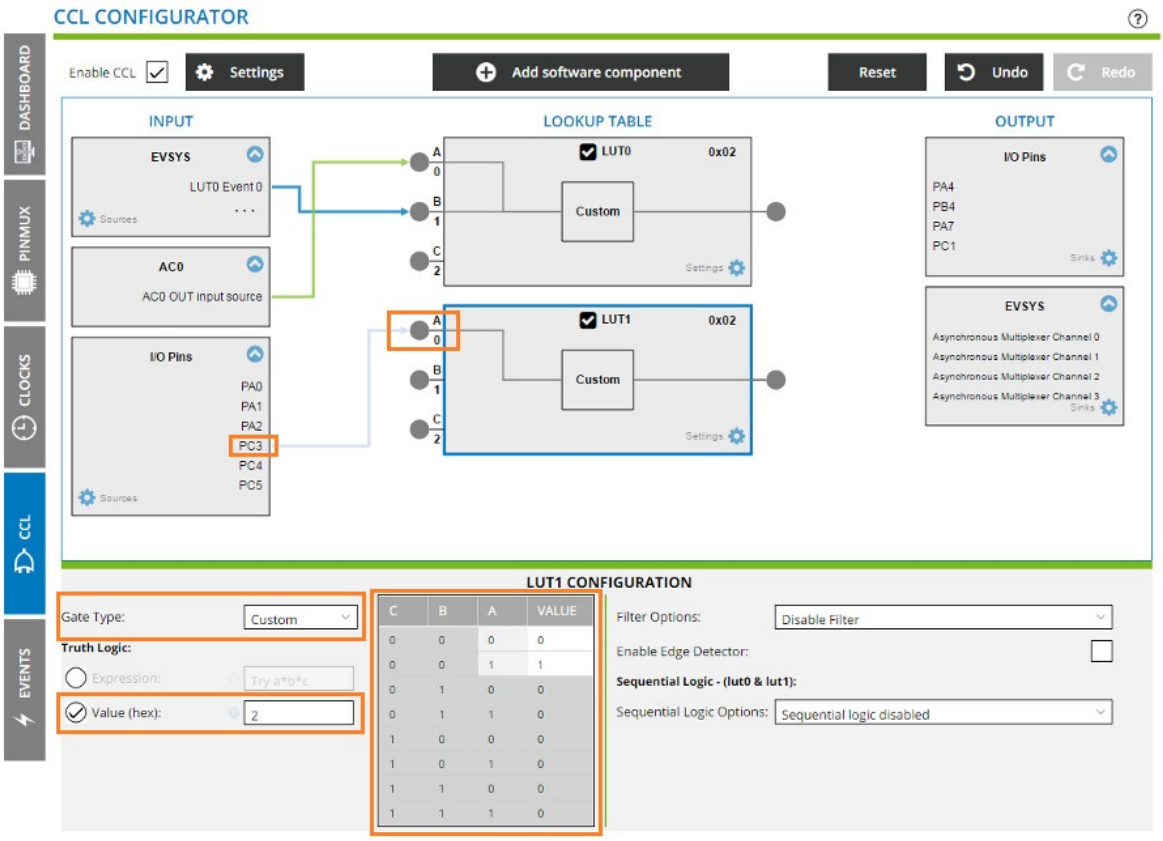
      The VALUE column in the truth table will be enabled to set individual bits.

      "Custom" should be elected in gate type for using option 7.2 and 7.3. The Truth table will automatically be updated based on the option selected.

Based on the example that is being considered here, "LUT0 output" represents "motion in low light condition". In order to configure that, select the AND gate from the drop-down box as shown in the figure below. Note that the truth table is updated to reflect the AND gate.



8. Configure the PC3 pin as Digital input using the PINMUX view. Assume that the appropriate external signal (e.g., from a push button) is connected to the PC3 pin.
9. Select and enable the second LUT (LUT1) and drag and drop **PC3** to the first Input (A) of "LUT1". The signal in PC3 should be made available "LUT1 output" to give to sequential circuit. Select "Custom" in Gate type Drop-down list and select value option under Truth Logic. Toggle the values in the truth table grid or directly enter the value in the Value-field as shown in the figure below.

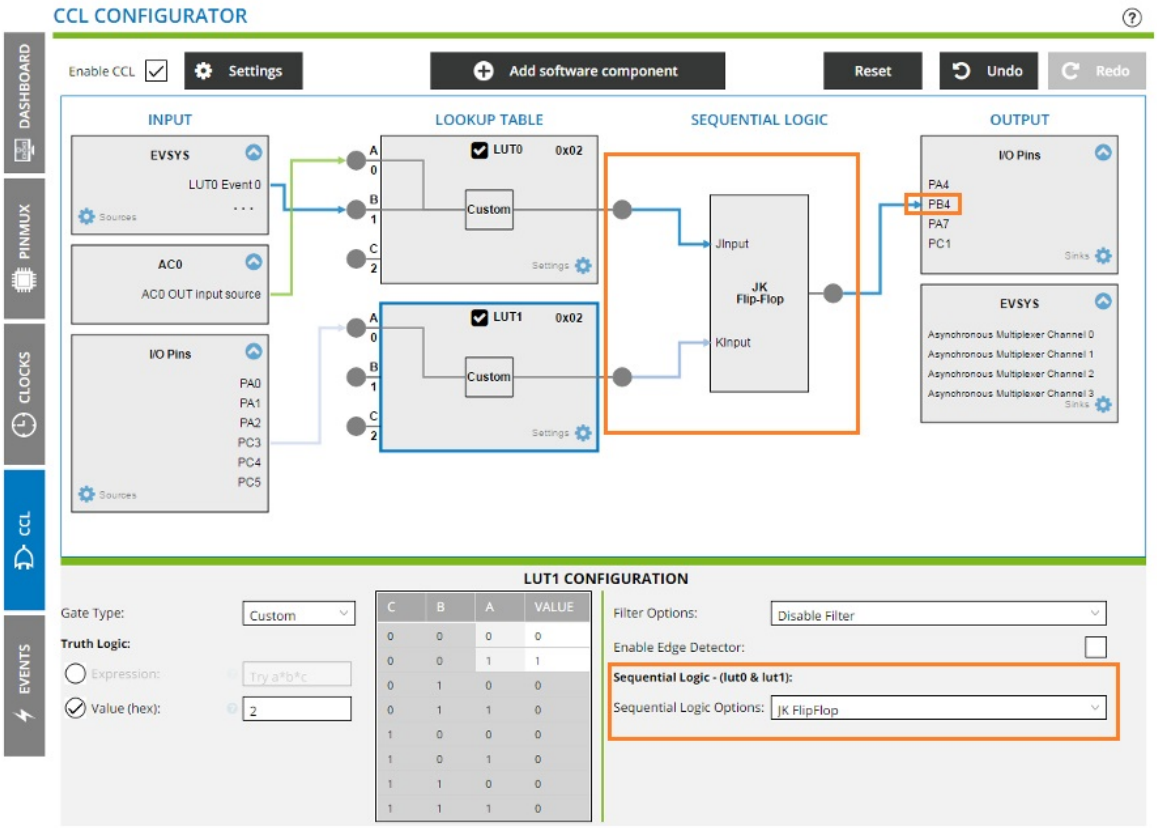


10. Select the JK flip-flop. “LUT0 output” is hardwired to the J input and “LUT1 output” is hardwired to the K input.

    The Configurator will be updated automatically to show these connections.

    Redirect the JK flip-flop output to pin PB4.

    A logic high will be available in PB4 whenever motion is detected under the low-light condition and the signal will be held available in PB4 until a switch is pressed (**PC3**).


CCL CONFIGURATOR
DASHBOARD
PINMUX
CLOCKS
CCL
EVENTS
Enable CCL
Settings
Add software component
Reset
Undo
Redo
INPUT
LOOKUP TABLE
SEQUENTIAL LOGIC
OUTPUT
EVSYS
LUT0 Event 0
Sources
AC0
AC0 OUT Input source
I/O Pins
PA0
PA1
PA2
PC3
PC4
PC5
LUT0
0x02
Custom
Settings
LUT1
0x02
Custom
Settings
JInput
JK Flip-Flop
KInput
I/O Pins
PA4
PB4
PA7
PC1
Sinks
EVSYS
Asynchronous Multiplexer Channel 0
Asynchronous Multiplexer Channel 1
Asynchronous Multiplexer Channel 2
Asynchronous Multiplexer Channel 3
Sinks
LUT1 CONFIGURATION
Gate Type:
Custom
Truth Logic:
Expression:
Try a*b*c
Value (hex):
2
C B A VALUE
0 0 0 0
0 0 1 1
0 1 0 0
0 1 1 0
1 0 0 0
1 0 1 0
1 1 0 0
1 1 1 0
Filter Options:
Disable Filter
Enable Edge Detector:
Sequential Logic - (lut0 & lut1):
Sequential Logic Options:
JK FlipFlop

5. TrustZone Support

TrustZone Projects

TrustZone for the ARMv8-M technology is enabled in Microchip SAM L11 Microcontrollers and provides hardware-enforced isolation between the trusted and the untrusted resources on the MCU device.

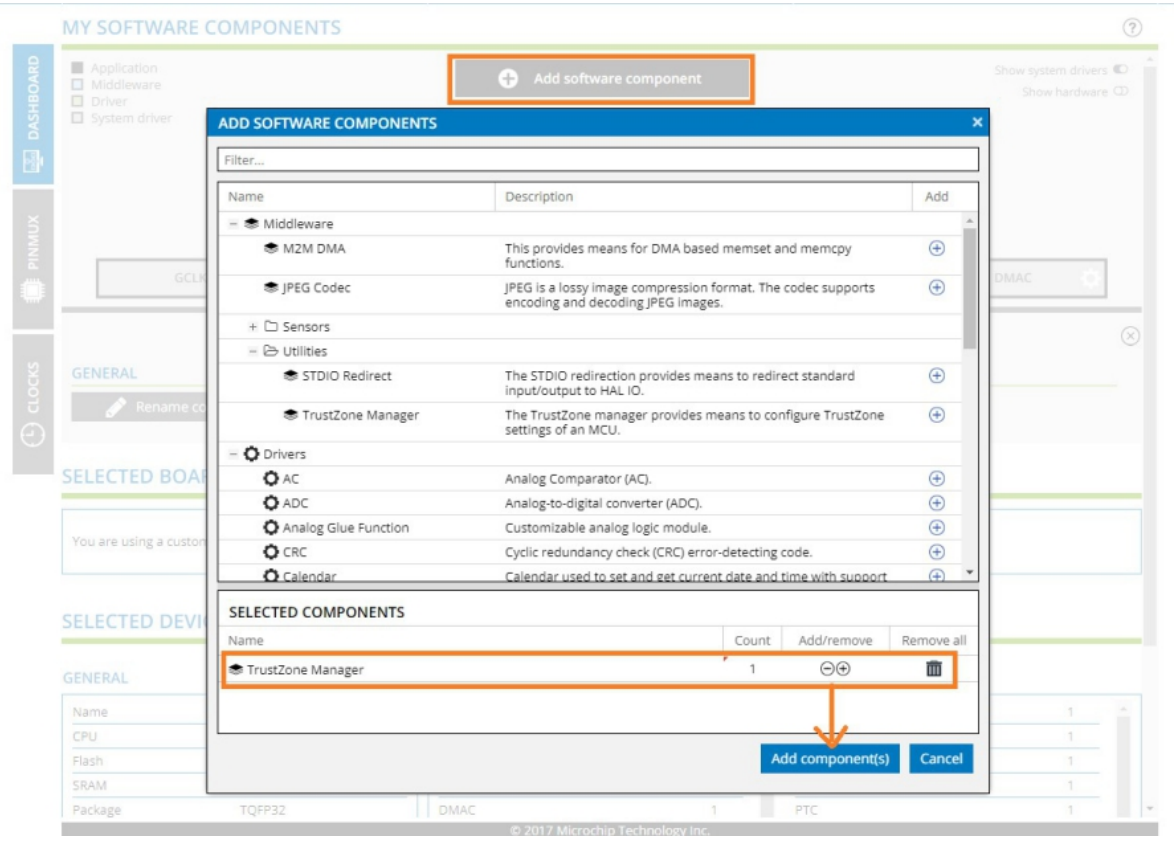
The TrustZone Manager component provides means to configure global TrustZone settings of a SAM L11 MCU, which is based on the Cortex-M23 core.

There are two types of TrustZone for ARMv8-M based projects:

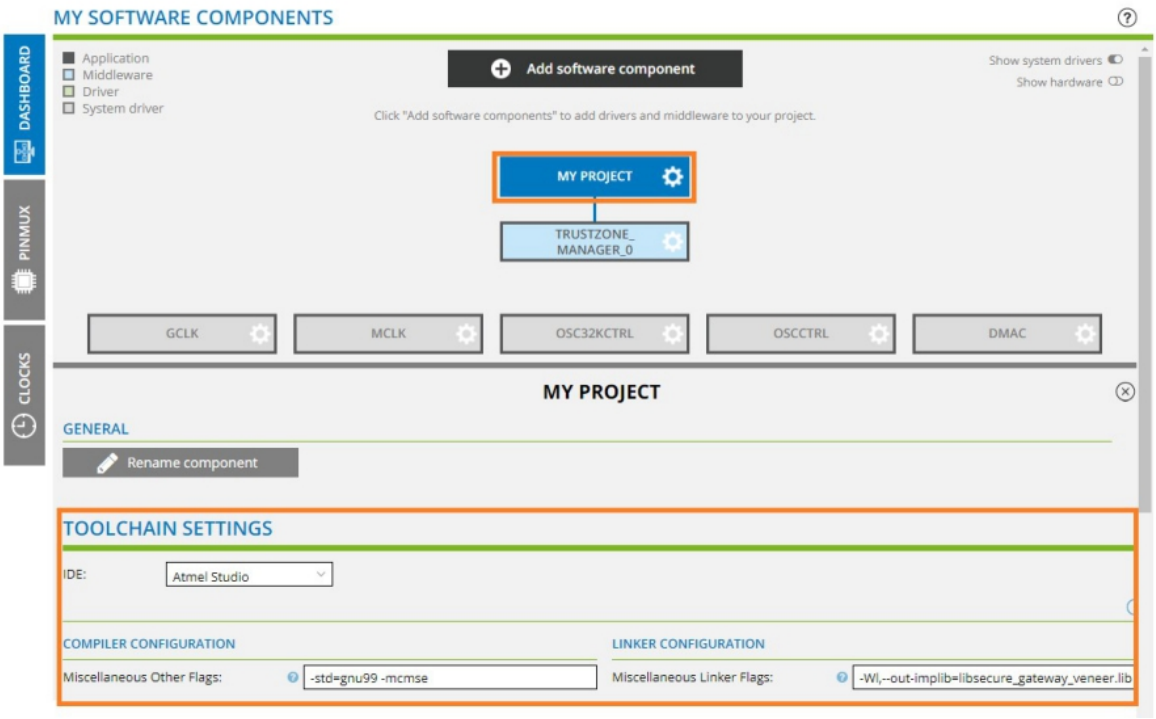
- **Secure project**: TrustZone Manager component MUST be added into the project to get access to the global TrustZone settings.
- **Non-Secure project**: Do not add TrustZone Manager component into the project as the Non-Secure project will rely on a Secure project for TrustZone global settings.

5.1 Creating a Secure Project

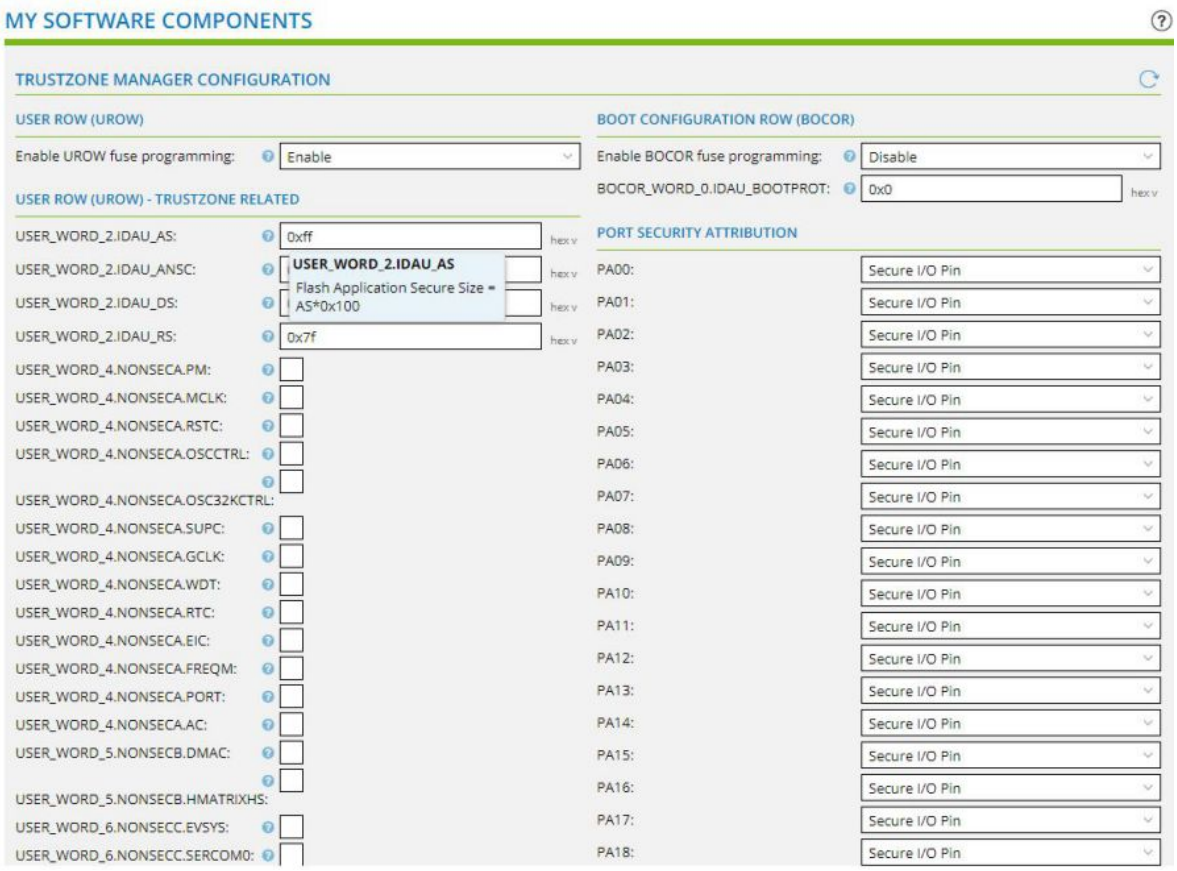
1. Create a new project with SAM L11 device.
2. Click on **Add software component** to add TrustZone Manager middleware.



3. Once the middleware is added, go to **DASHBOARD** and select the Project component. The toolchain settings for Atmel Studio will be displayed under the Project component. The toolchain settings for IAR and Keil are not provided in the UI; the user needs to manually select those settings in the respective IDE.



4. Compiler miscellaneous flags are populated with '`-std=gnu99 -mcmse`' settings. The '`-mcmse`' flag is used to enable the compiler for creating CMSE secure executable files.
5. Linker miscellaneous flags are populated with '`-Wl,--out-implib=libsecure_gateway_veneer.lib -Wl,--cmse-implib`' settings. '`-Wl,--cmse-implib`' is for generating a secure gateway veneer and a symbol in an import library. If you do not need to generate a secure gateway library, remove this setting and keep the text box empty.
6. Select the TrustZone Manager Middleware software component and configure the settings like the different memory secure and non-secure sizes or the peripherals security attribution.
7. Hover over the question mark near each setting to know more about those settings.



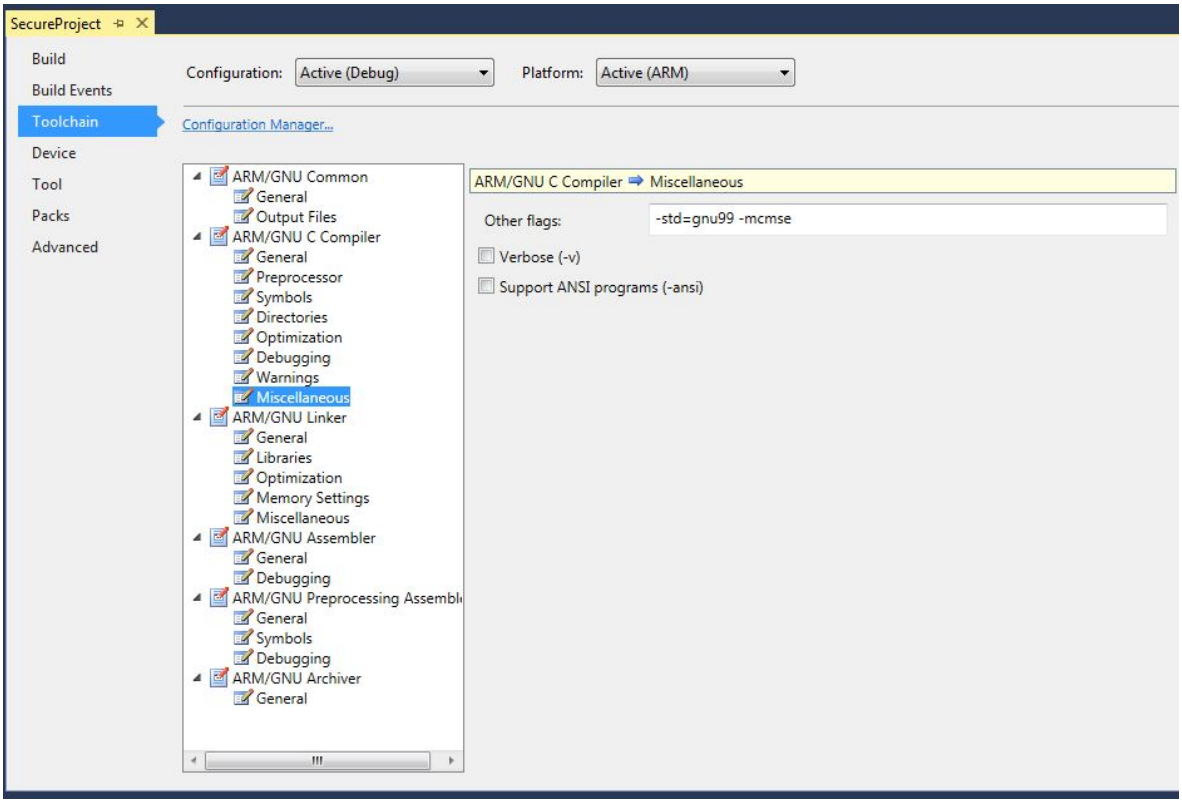
8. The Secure project will create two linker script files based on the above configuration; one for using in the secure project and another for using in the non-secure project that will be created later on.

5.2 Creating a Non-Secure Project

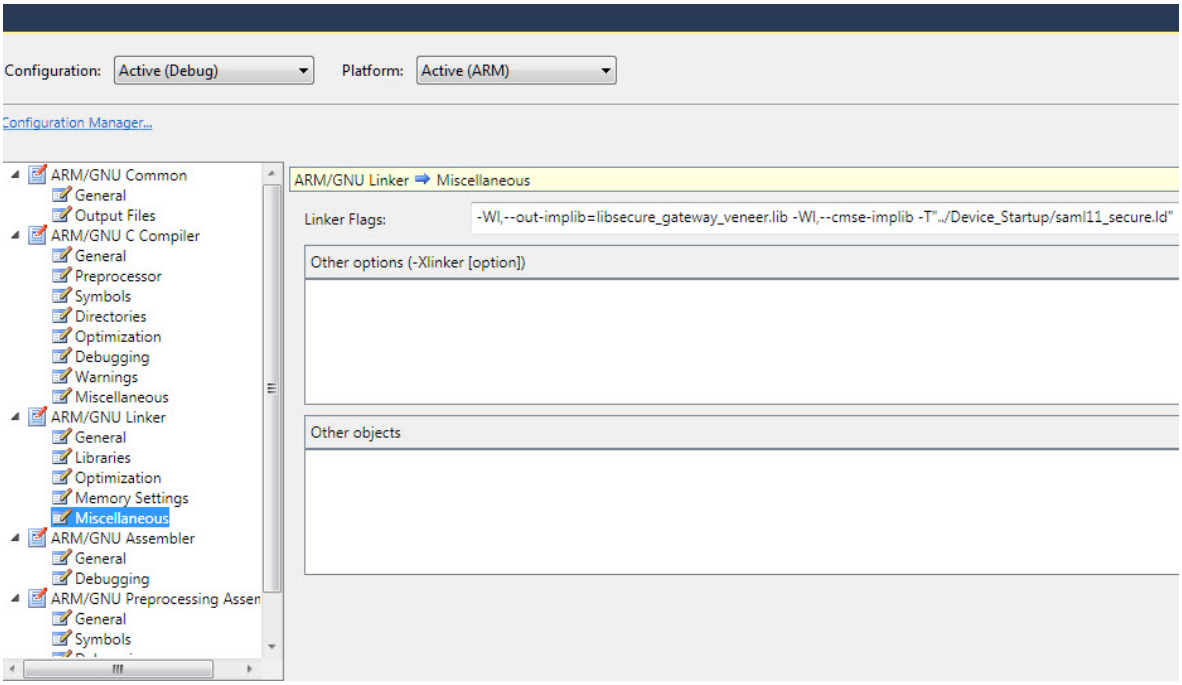
1. Create a project using SAM L11 device without adding the TrustZone Manager Middleware. Toolchain settings will not be displayed for non-secure projects.

5.3 Importing the Secure Project in Atmel Studio

1. Import the secure project ATZIP in the Atmel Studio IDE using the procedure described here - 6.1.2 Steps to Import the Atmel START Output into Atmel Studio.
2. Go to Project Properties by right clicking the project name in the solution explorer and select Toolchain. Under the Compiler Configuration, go to Miscellaneous settings and see the compiler settings from the START have been updated.

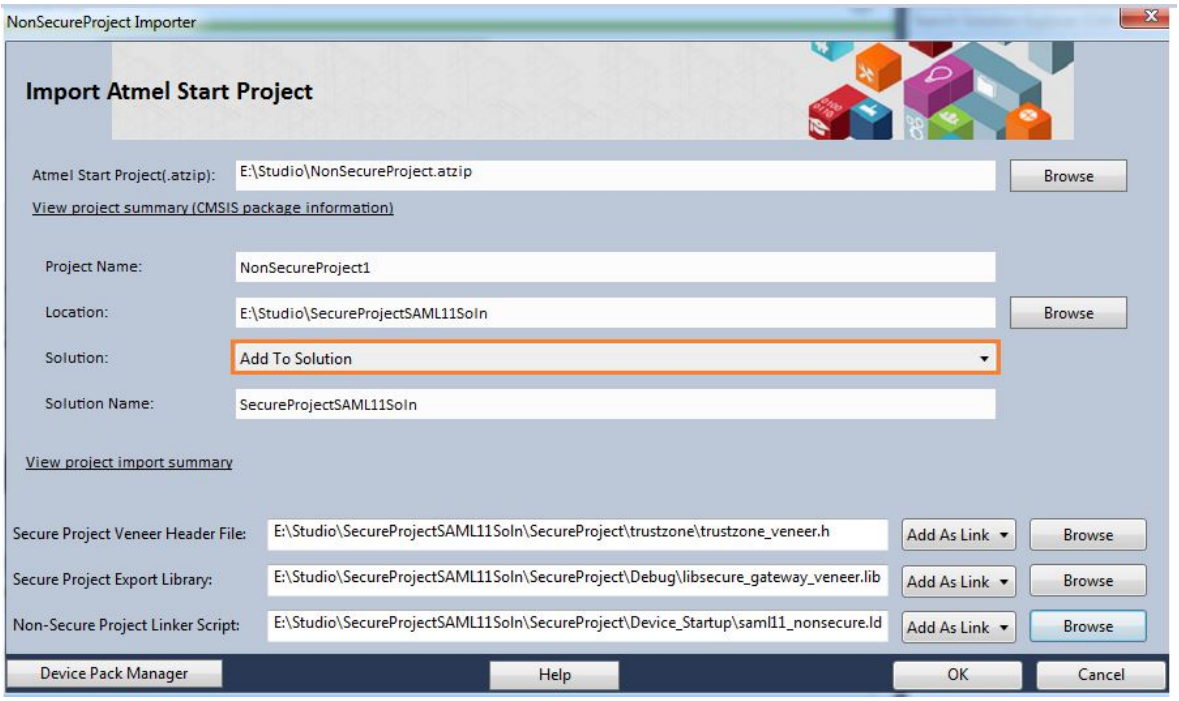


3. Under the Linker configuration, go to Miscellaneous settings and see the Linker configuration from the START has been updated along with the secure linker script path.

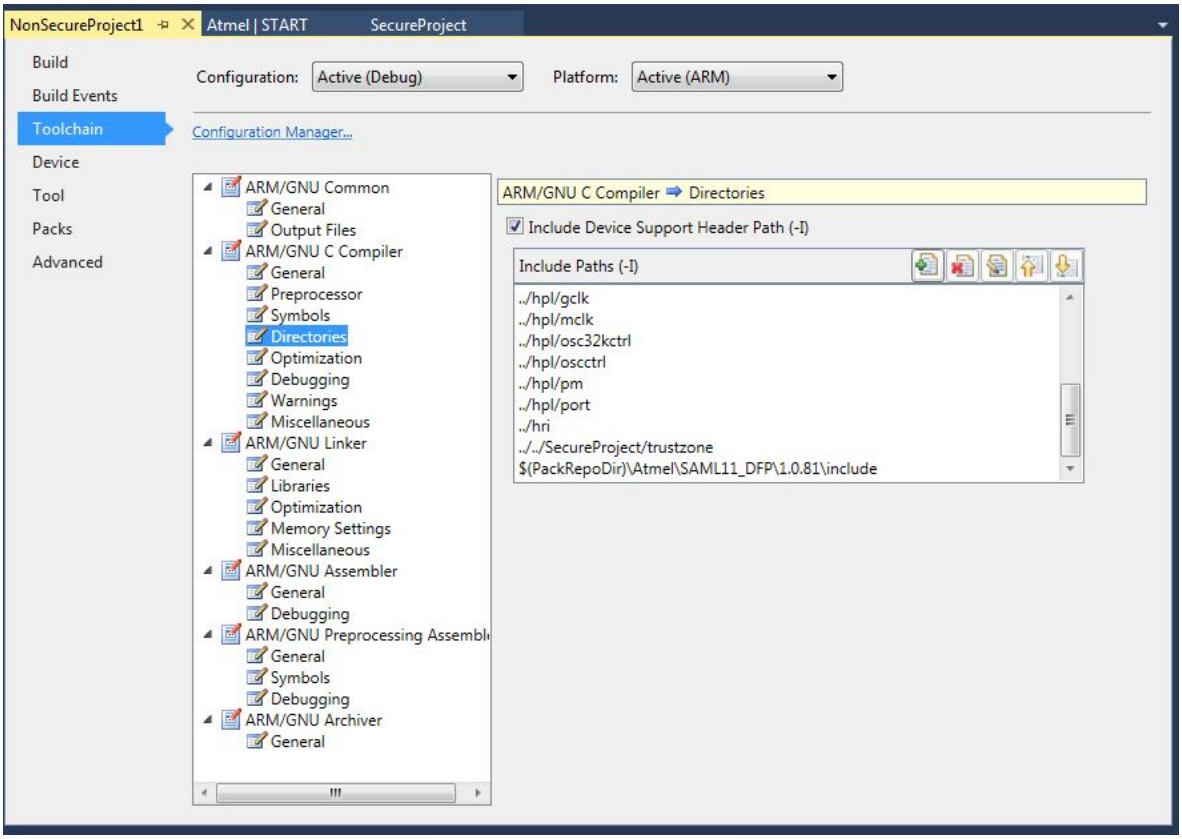


5.4 Importing the Non-Secure Project in Atmel Studio

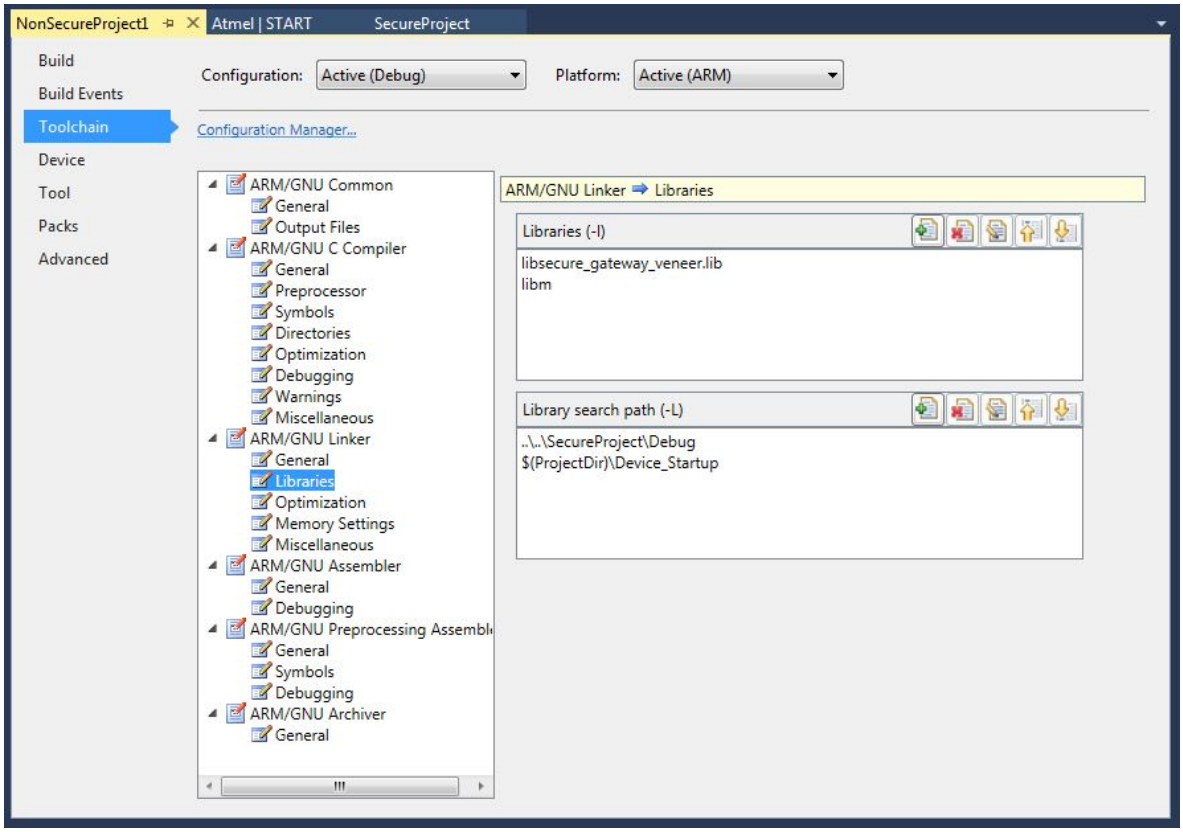
1. The non-secure project can be imported either by creating a new solution or import it directly to a secure project solution if one exists.
2. For importing directly to a secure project solution, open the secure project solution and import the non-secure project into the solution as described here - 6.1.2 Steps to Import the Atmel START Output into Atmel Studio. Make sure that "Add to Solution" is selected as shown in the figure below.
3. While importing the non-secure project into Studio, the user will be asked to browse the veneer header file, Secure Project Export Library, and Linker Script File to complete the import process. You can get these files from a pre-compiled secure project.
4. While selecting the files, the user has two options:
   - **Add as a link**: Add link to the path where the file is browsed instead of copying the file to the project. Usually chosen if you embed both secure and non-secure projects in a single solution.
   - **Add as a file**: Copy and add the files to the Project. Normally used if the above-required files are being given by another user source and no secure project is available on the user side.
5. Browse the files and click **OK** to complete the import.



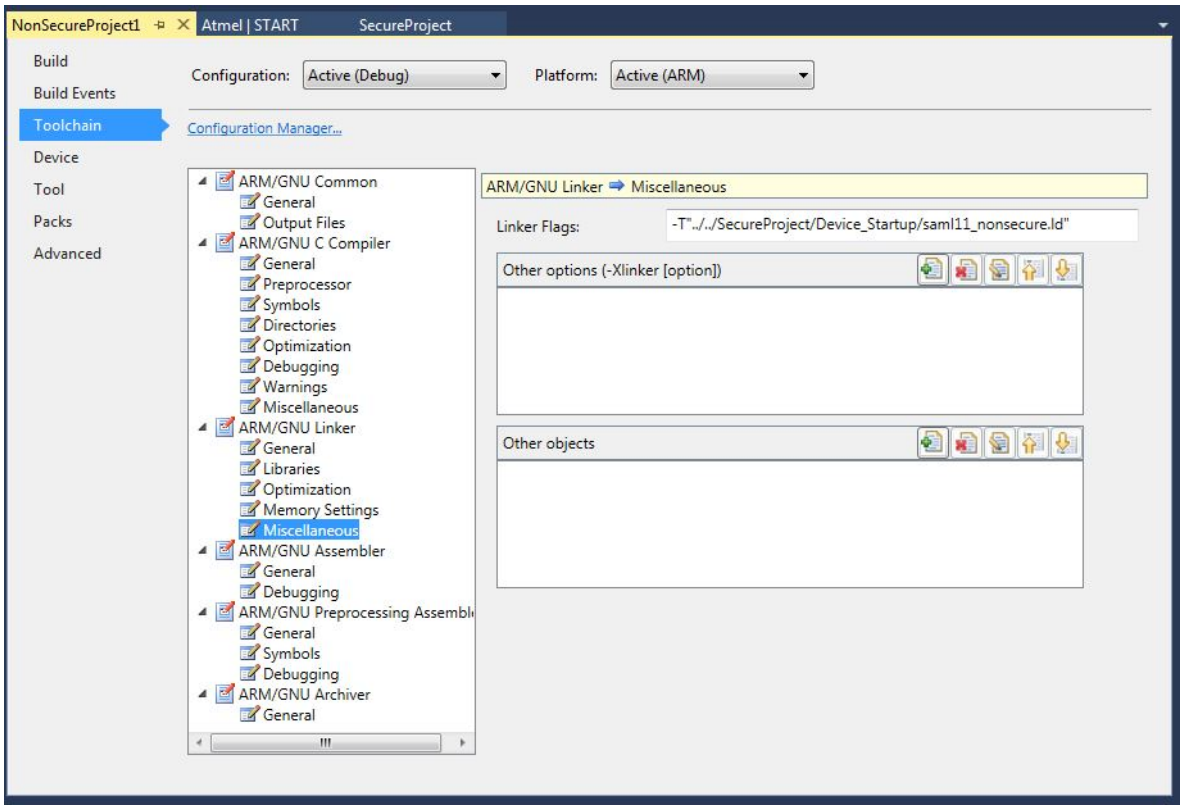
6. Go to Project Properties by right clicking the project name in the solution explorer and select Toolchain. Under the Compiler Configuration, go to Directories settings and see the include path has been updated.



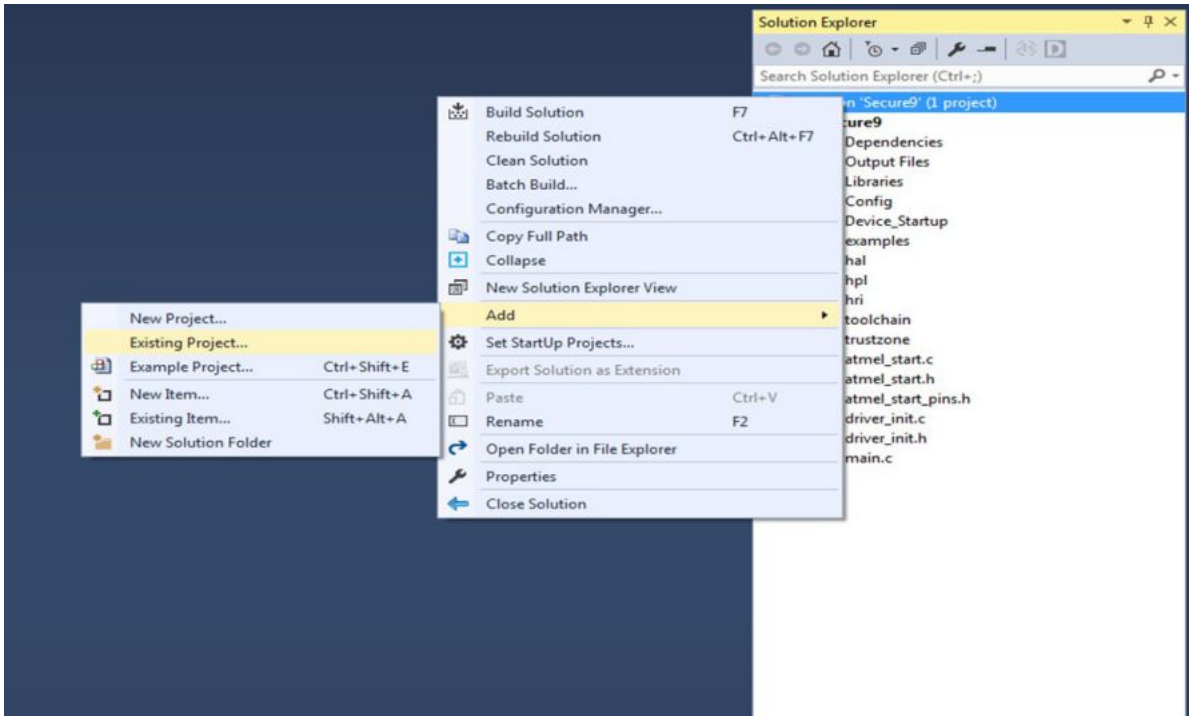
7. Under the Linker configuration, go to Libraries and see the Browsed Library and its path have been updated.



8. Go to Miscellaneous settings and see the browsed linker script path has been updated.

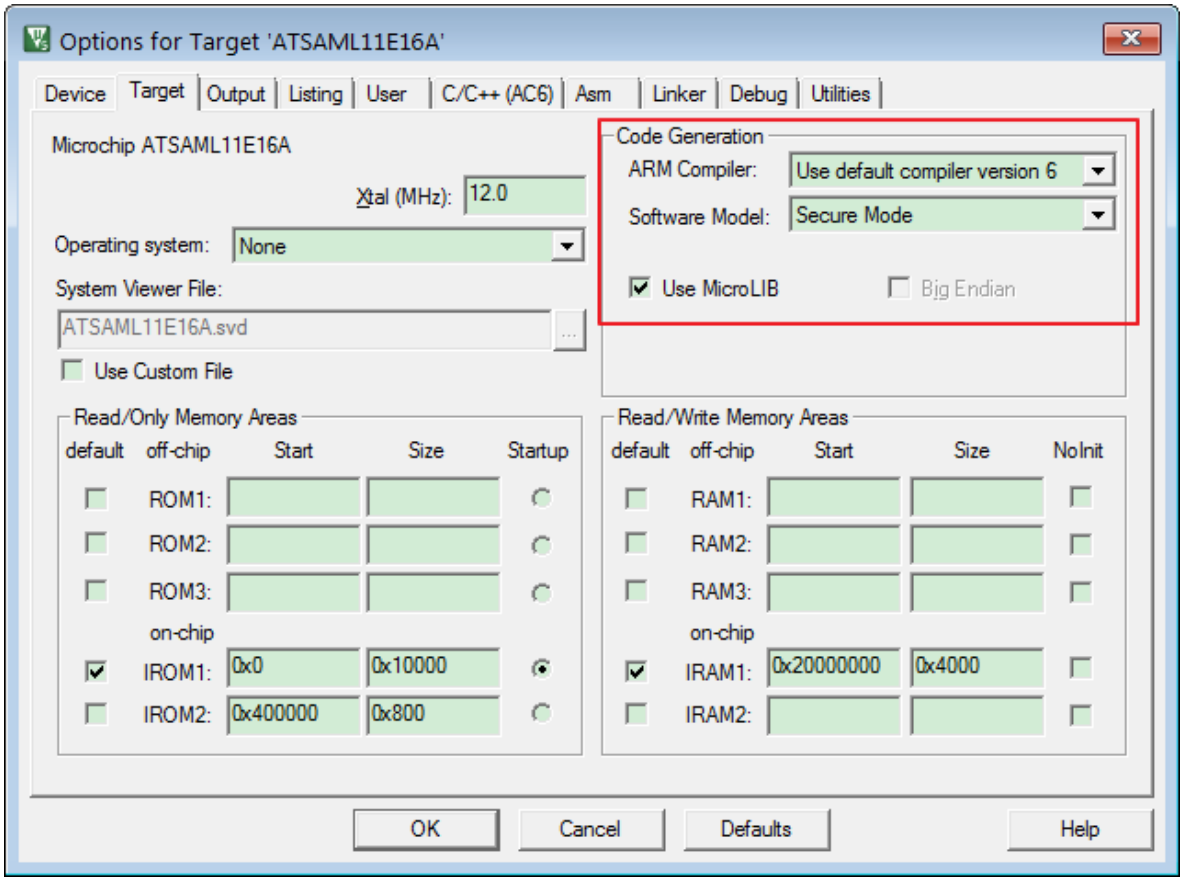


Tip: The Non-Secure Project can also be imported by creating a new solution. Once the solution is created it can be added into a secure solution using **Add** > **Existing Project**, as shown below:

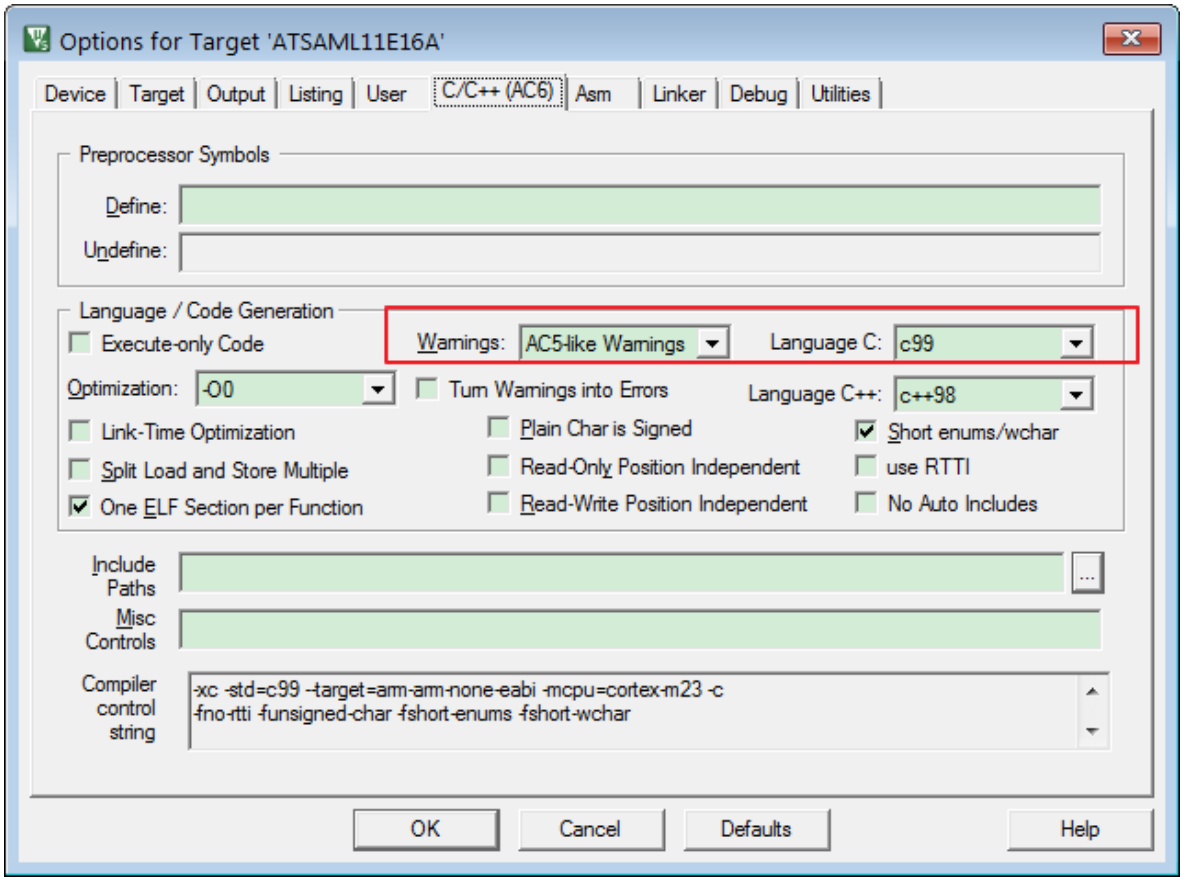


5.5 Importing the Secure Project in µVision® from Keil®

1. For importing both secure and non-secure projects in the same workspace:
   - Create a folder and unzip the secure and non-secure project in the same folder
   - Rename the *gpdsc files in both folders to avoid confusion (e.g: AtmelStart_S.gpdsc and AtmelStart_NS.gpdsc)
   - Create a Keil project file by selecting these *.gpdsc files separately:
     3.1. File -> Open -> Secure project folder -> select the AtmelStart_S.gpdsc file (after selecting Show All File Types) -> a project will be created. Save and close the project.
     3.2. Repeat the above step for the non-secure project
   - Create a Multi Project workspace in Keil MDK and add both the project files (.uvprojx file for secure and non-secure)
   - Configure the secure project as described below
2. Go to Options by right-clicking the project name in the workspace. Under the Target tab, go to the Code Generation section and make sure the ARM Compiler is “Use default compiler version 6”, change the software model to “Secure Mode” for a secure project, and select the check box “Use MicroLIB”.

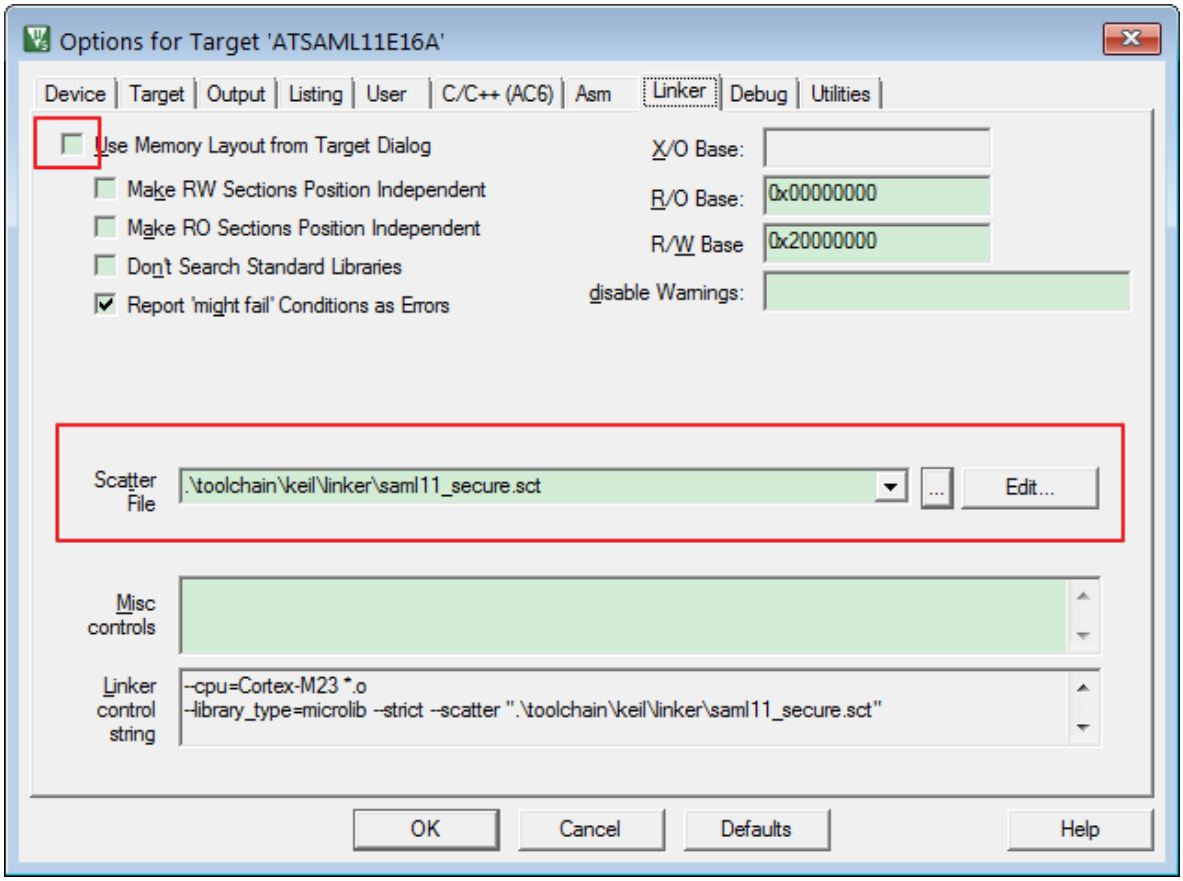


3. Go to the C/C++ tab. Under the Language/Code Generation section change the language to “C99”.

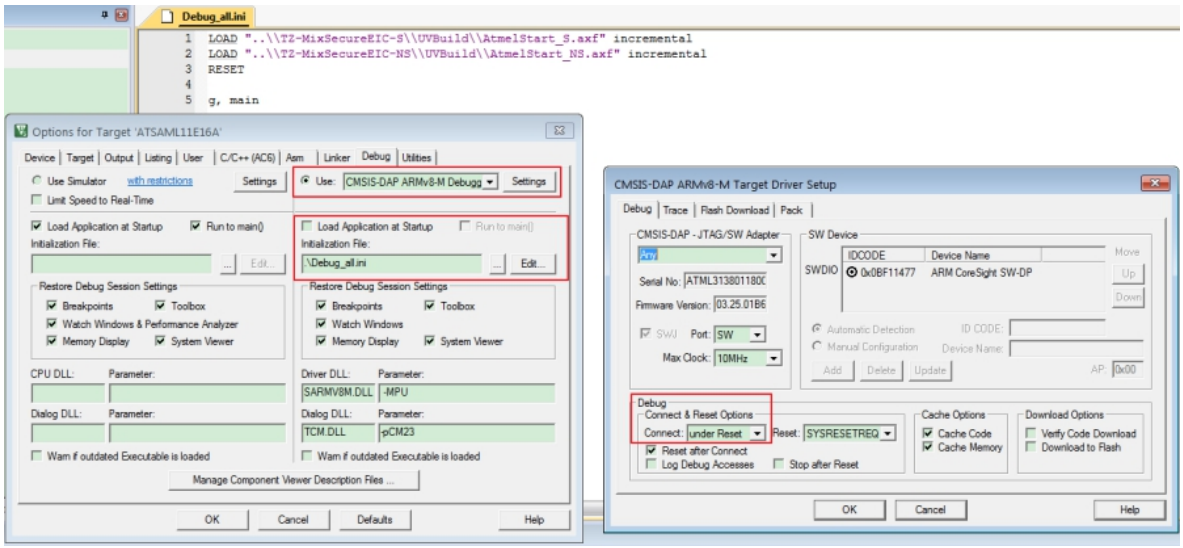


Note: Selecting the value for Warnings as “AC-5 like Warnings” is optional.

4. Under the Linker tab, uncheck the “Use Memory Layout from Target Dialog Box” and browse the Scatter file from the secure project.



5. Under the Debug tab, select the value CMSIS-DAP for the “Use” radio button. Click on Settings and go to the Pack tab. Change the default dbgconfig file with the new UROW settings, which will be available in Toolchain\keil\DebugConfig\SAML11.dbgconf.



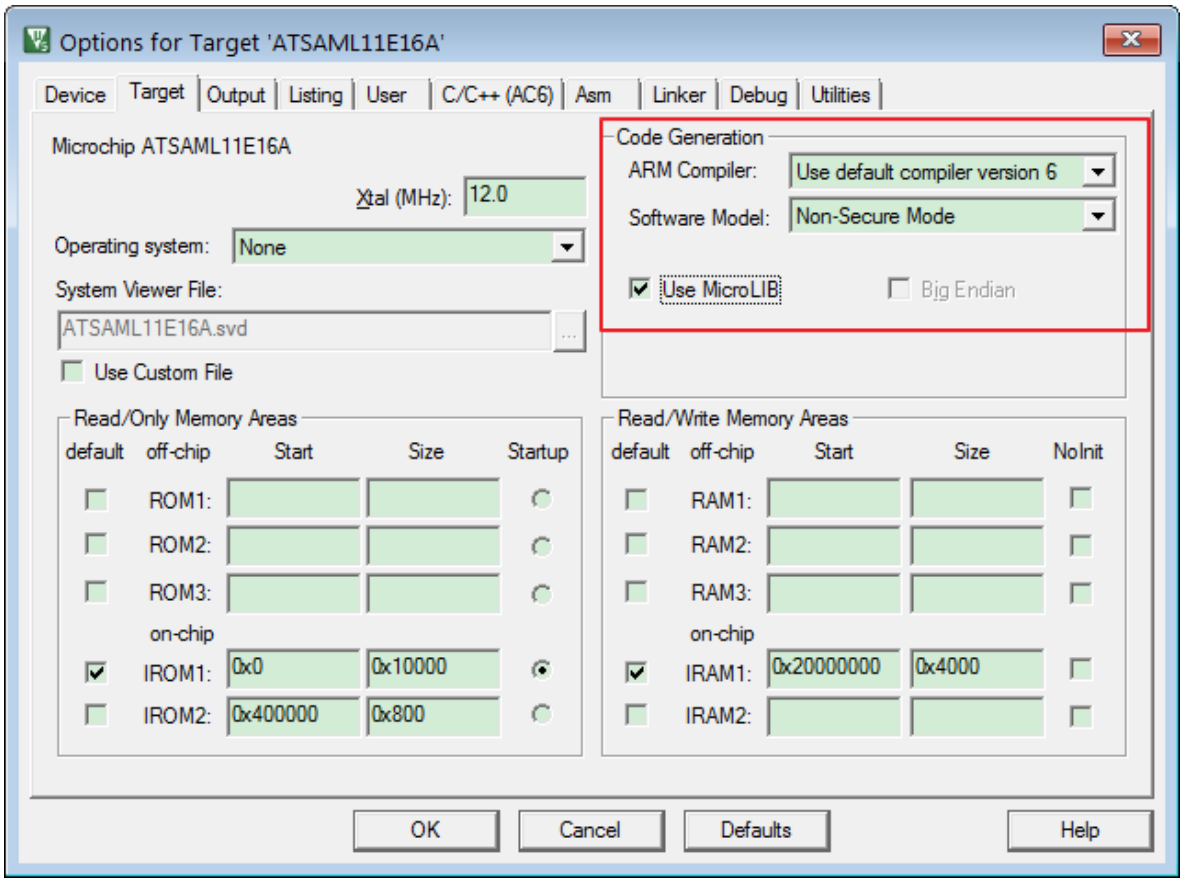
6. In addition, go to the Debug tab in Options for a secure project and edit the content of the initialization file as follows:
   - LOAD “Specify the Path of .axf file of non-secure project" incremental
   - LOAD “Specify the Path of .axf file of secure project” incremental

- RESET
- g, main

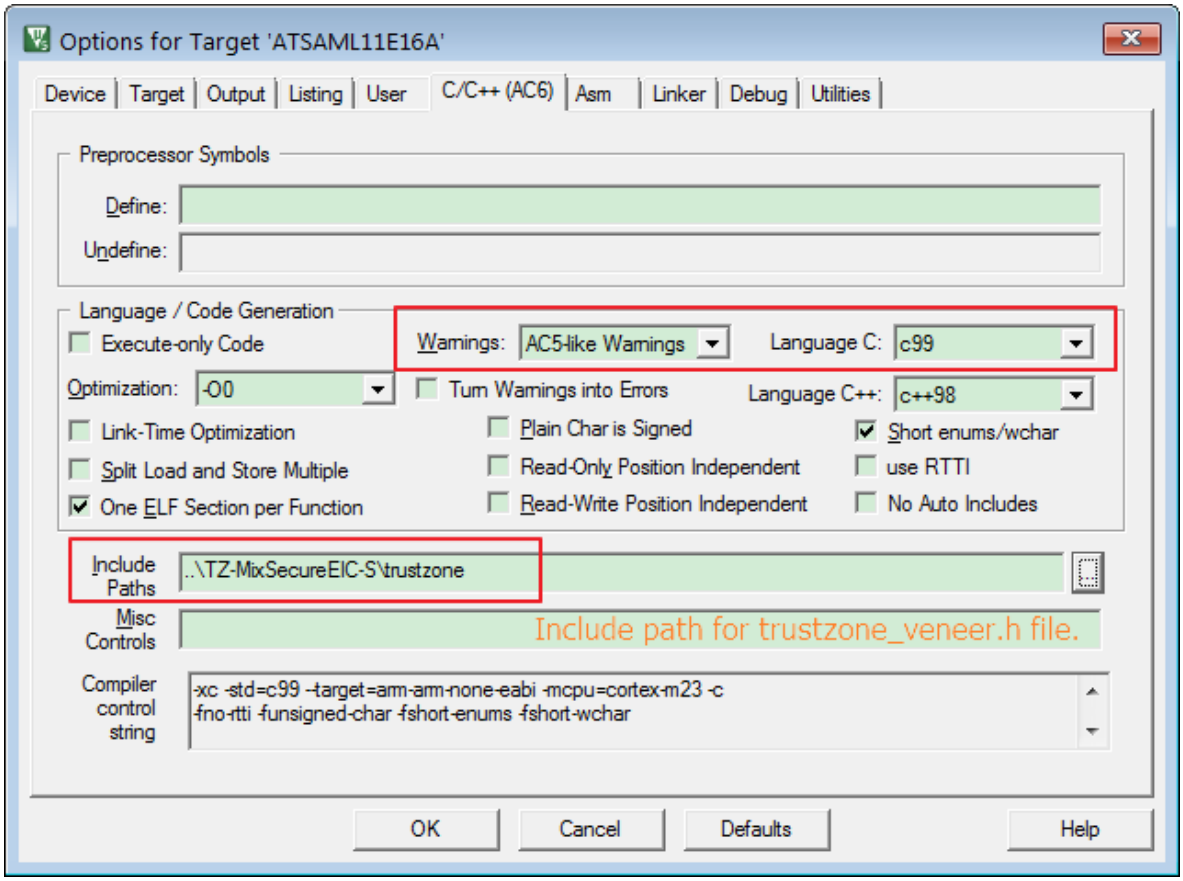
7. Under the Utilities tab, go to the Configuration Flash menu command and browse the Init file as Load_ns.ini under “User Target Driver for Flash Programming”. Edit the content of the Init file as follows:
   - LOAD “..\\Specify Non-secure project path\\UVBuild\AtmelStart_NS.axf” incremental
8. Compile the secure project.
9. Configure the non-secure project as described here – Configure Non-Secure Project.

5.6 Importing the Non-Secure Project in µVision® from Keil®

1. Create a Multi Project workspace as described here - Importing the Secure Project in µVision® from Keil®.
2. Go to the Options by right-clicking the project name in the workspace. Under the Target tab, go to the Code Generation section and make sure the ARM Compiler is “Use default compiler version 6”, change the software model to “Non-Secure Mode” for a secure project, and select the check box “Use MicroLIB”.

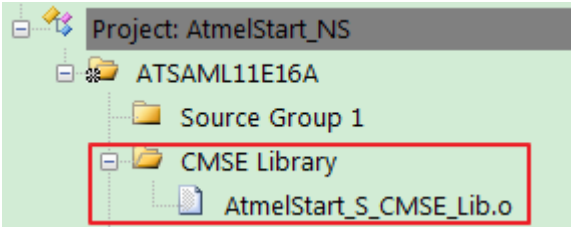


3. Go to the C/C++ tab. Under the Language/Code Generation section change the language to “C99”. Browse the path of Trustzone_veneer.h file in Include Paths.

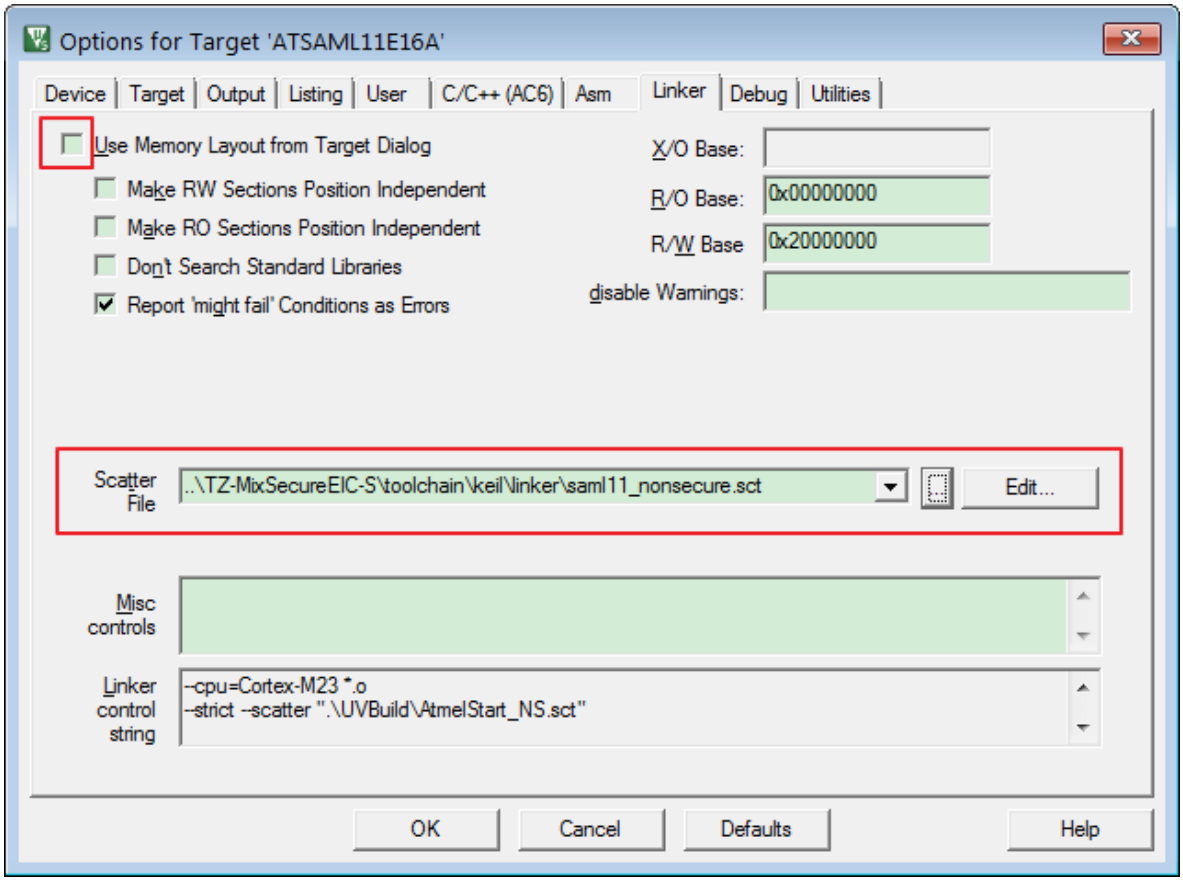


Note: Selecting the value for Warnings as “AC-5 like Warnings” is optional.

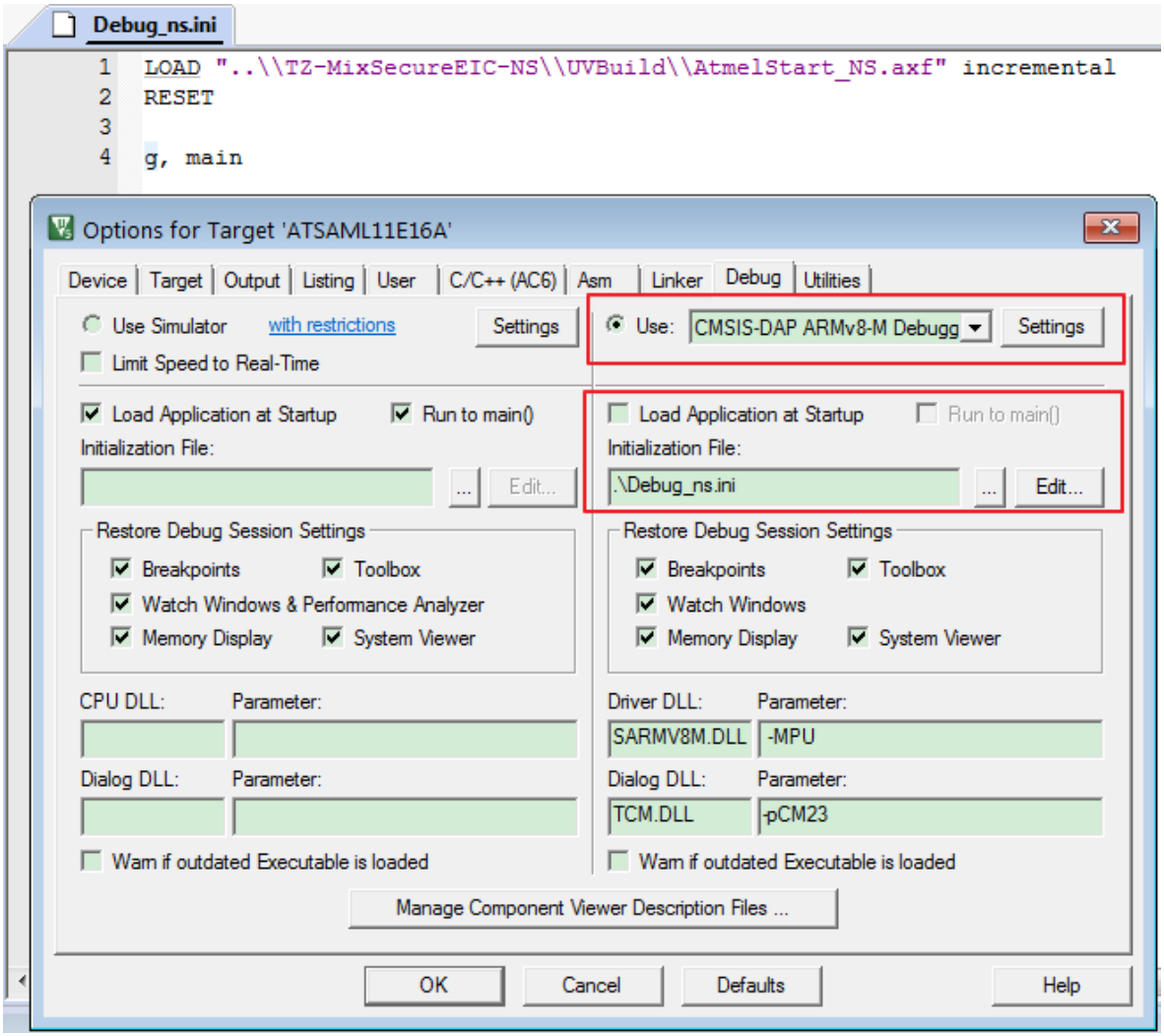
4. Add a secure veneer library file (Workspace \UVBuild\AtmelStart_S_CMSE_Lib.o) in a non-secure project.



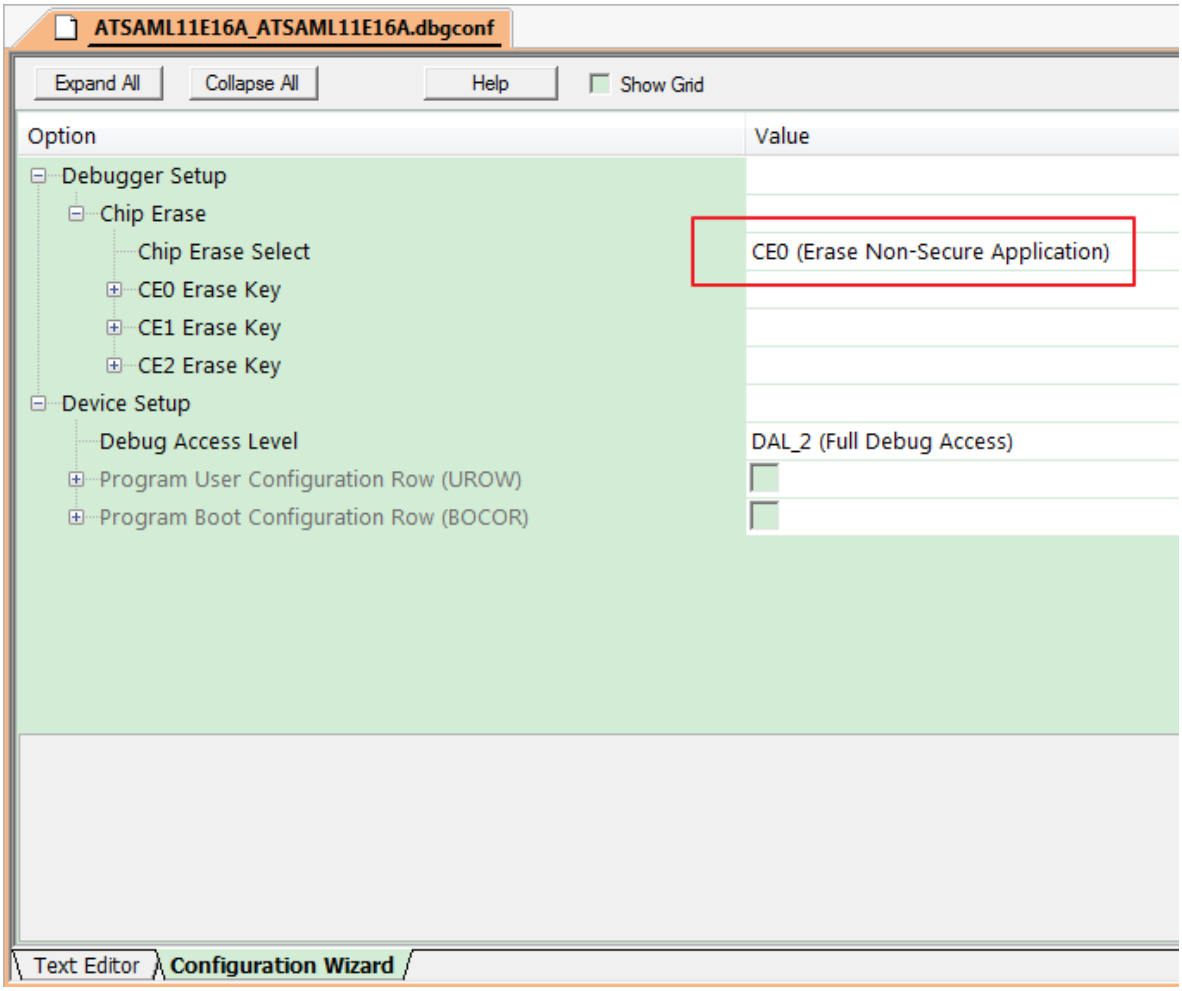
5. Under the Linker tab, uncheck the “Use Memory Layout from Target Dialog Box” and browse the Scatter file from the non-secure project.



6. Under the Debug tab, select the value as CMSIS-DAP for the “Use” radio button. Click on Settings and go to the Pack tab. Change the default dbgconfig file with the new UROW settings which will be available in Toolchain\keil\DebugConfig\SAMl11.dbgconf.



7. In the configuration wizard, change the Chip Erase value to CE0 for a non-secure project.



8. In the Debug tab, browse the initialization file as Debug_ns.ini and edit the file by adding the following content:
   - LOAD “Specify the Path of .axf file of non-secure project" incremental
   - RESET
   - g, main
9. Compile the non-secure project.

5.7 Importing the Secure Project in IAR Embedded Workbench®

IAR Embedded Workbench support will be added soon.

5.8 Importing the Non-Secure Project in IAR Embedded Workbench®

IAR Embedded Workbench support will be added soon.

5.9 Importing the Secure Project using Makefile

Build the project using Makefile, following the procedure described here - 6.4.2 Executing a GNU C Makefile.

5.10 Importing Non-Secure Project using Makefile

1. Build the project using Makefile, by following the procedure described here - 6.4.2 Executing a GNU C Makefile.
2. It displays the following warning messages:
   2.1. The SAML11 non-secure project needs input files from the secure project. Here are the example steps:
      2.1.1. Get the '`Device_Startup/saml11_nonsecure.ld`', '`trustzone/trustzone_veneer.h`' and '`gcc/libsecure_gateway_veneer.lib`' files from the secure project.
      2.1.2. Copy the '`saml11_nonsecure.ld into gcc/gcc`' folder in the non-secure project.
      2.1.3. Copy '`trustzone_veneer.h`' and '`libsecure_gateway_veneer.lib`' into the root folder, in the non-secure project.
      2.1.4. Update Makefile to use the new linker script: `-T"../gcc/gcc/saml11_nonsecure.ld"`.
      2.1.5. Update Makefile to include the secure gateway lib: `-Wl,--start-group -lm -l:libsecure_gateway_veneer.lib -L"../" -Wl,--end-group`.
3. Follow the procedure described in the warning message.

6. Using Atmel START Output in External Tools

When your project is configured in Atmel START, you must export it in order to use it in a C compiler.

Related Links

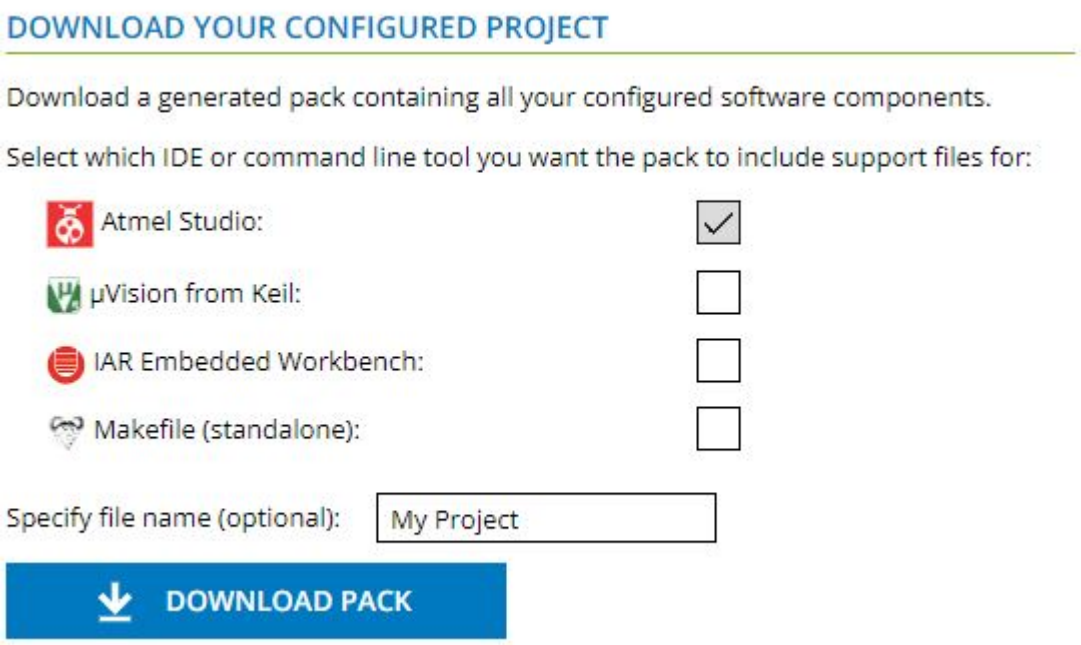
2.2 Supported IDEs and Compilers

6.1 Atmel Studio

The following steps describe how to use the output from Atmel START in Atmel Studio.

6.1.1 Exporting the Project from Atmel START

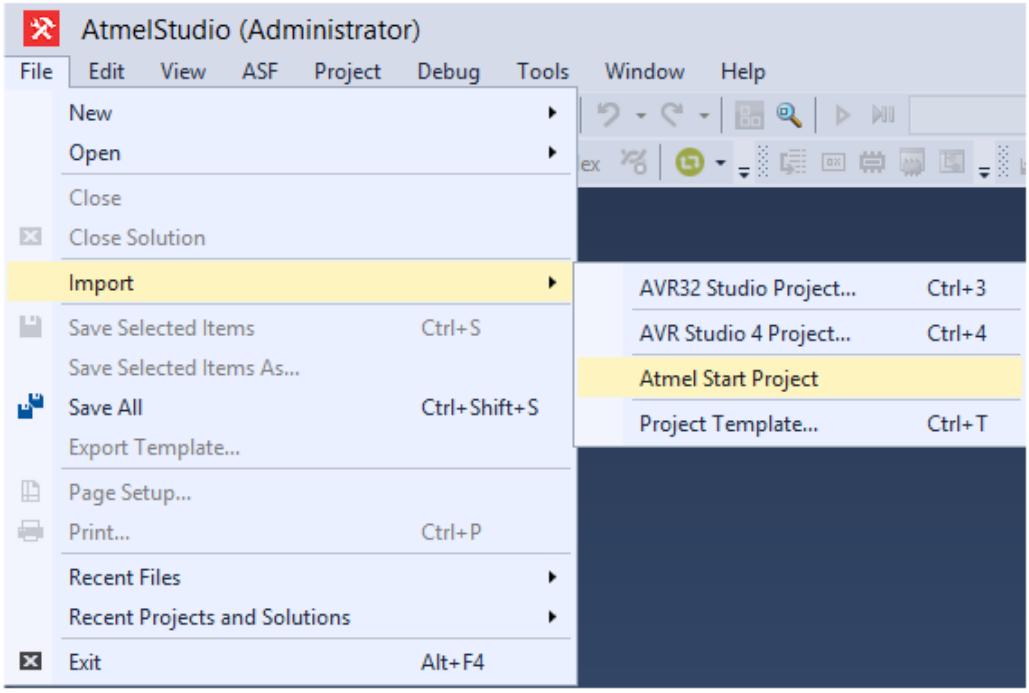
1. On the Atmel START website, create a new project (Example or Board).
2. Click on the **Export Software Component** button. For exporting project for Atmel Studio and Somonium IDE, select Atmel Studio.



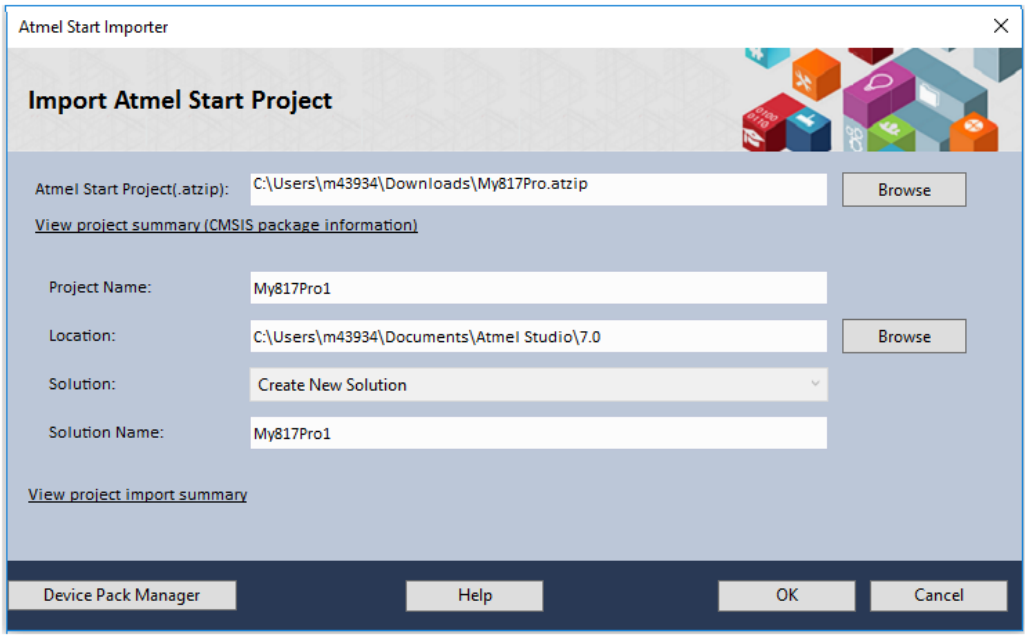
3. Click on **DOWNLOAD PACK**.
   An '`atmelstart.atzip`' pack file will be downloaded.

6.1.2 Steps to Import the Atmel START Output into Atmel Studio

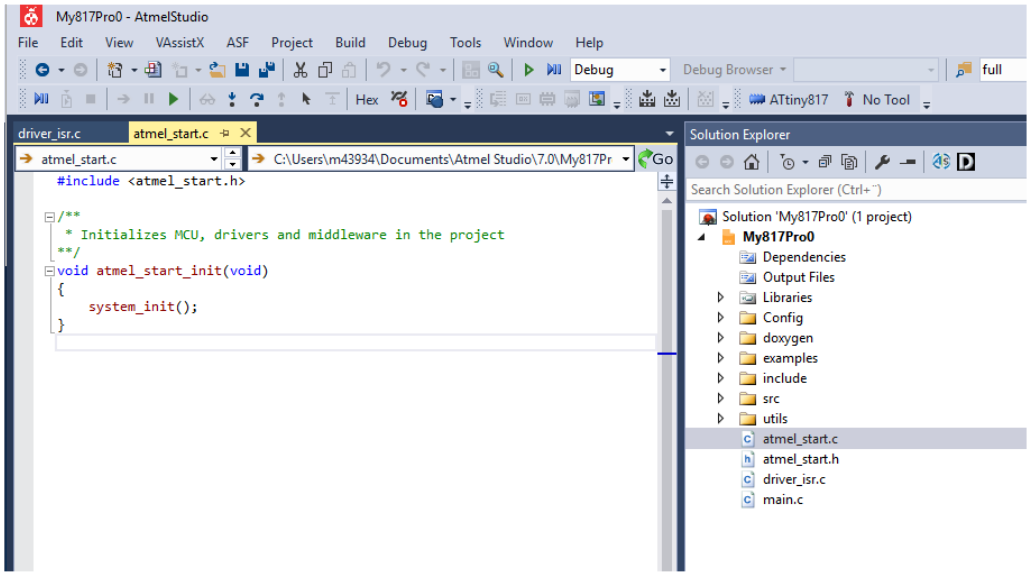
1. Launch Atmel Studio.
2. Select **File** > **Import** > **Atmel Start Project**.



3. Browse and select the downloaded '`atmelstart.atzip`' file.
4. The "Atmel Start Importer" dialog box will open. Enter the project details **Project name** > **Location** > **Solution name**. Click **OK**.



A new Atmel Studio project will be created, and the files will be imported.



6.1.3 Generate Doxygen Output

Some projects contain documentation formatted for Doxygen.

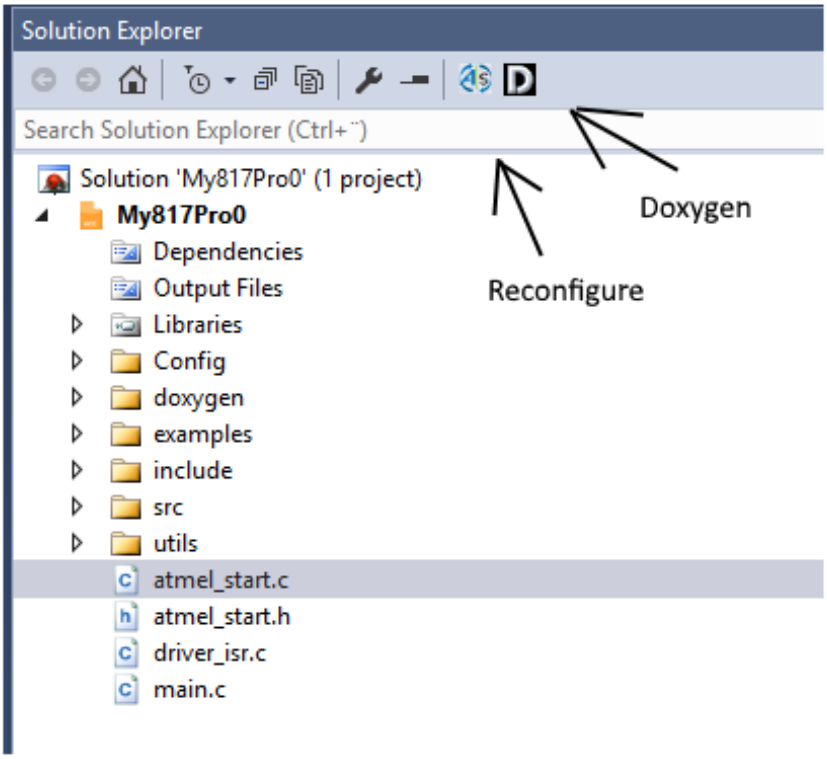
1. **Note:** Doxygen must be installed; download it from http://www.doxygen.org. You will be asked to configure Studio to locate Doxygen executable; this defaults to '`C:\Program Files\doxygen\bin\doxygen.exe`'.

   Click on the **Doxygen** button to generate documentation.
   Doxygen will run, and the generated documentation will open in a window.

Tip: The generated files can be found under the folder '`doxygen/generator/html`' and '`doxygen/generator/xml`'. These are not added to the project. There is a separate Doxygen extension for Studio that can be used if more control is needed. It can be found at https://gallery.atmel.com

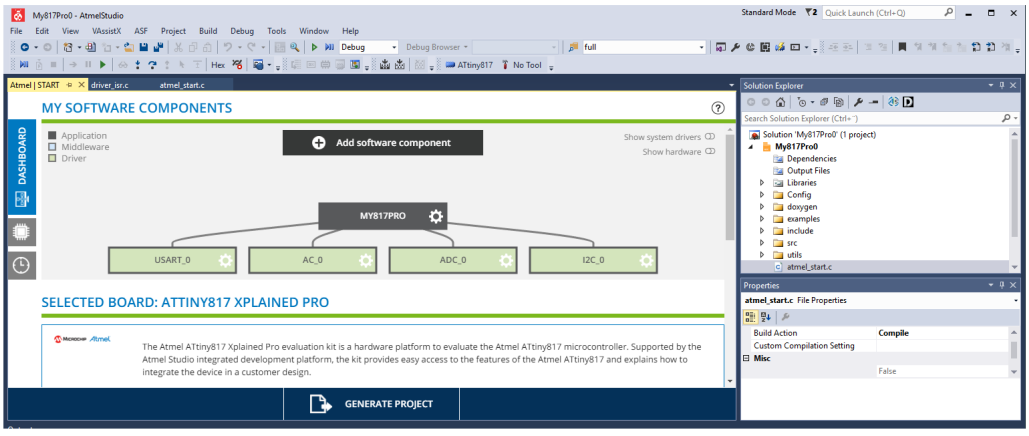
6.1.4 Steps to Re-Configure the Project using Atmel START

1. Click on the **Re-Configure** button or right click on the project node in the Solution Explorer window, and from the menu, select Re-configure Atmel Start Project.



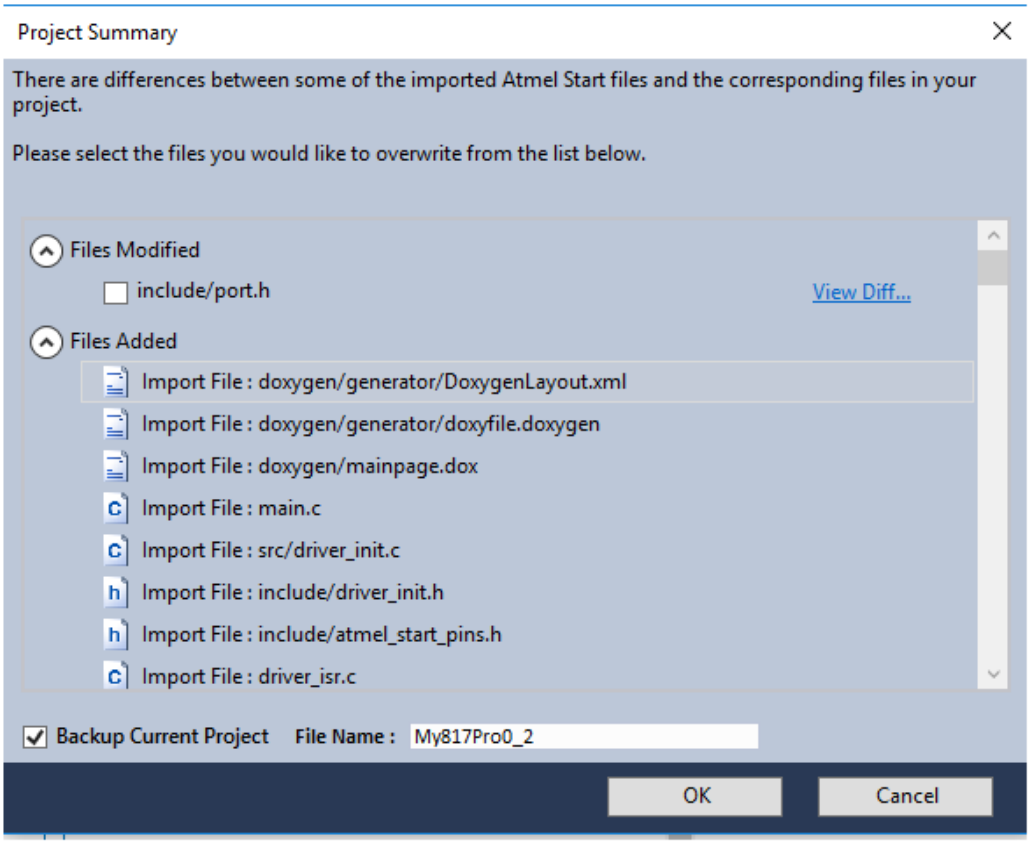
Atmel START will open up in a window inside Atmel Studio.

2. Do changes to the project; click the **GENERATE PROJECT** button at the bottom of the Atmel START window.



The old configuration will be removed and the new configuration applied.

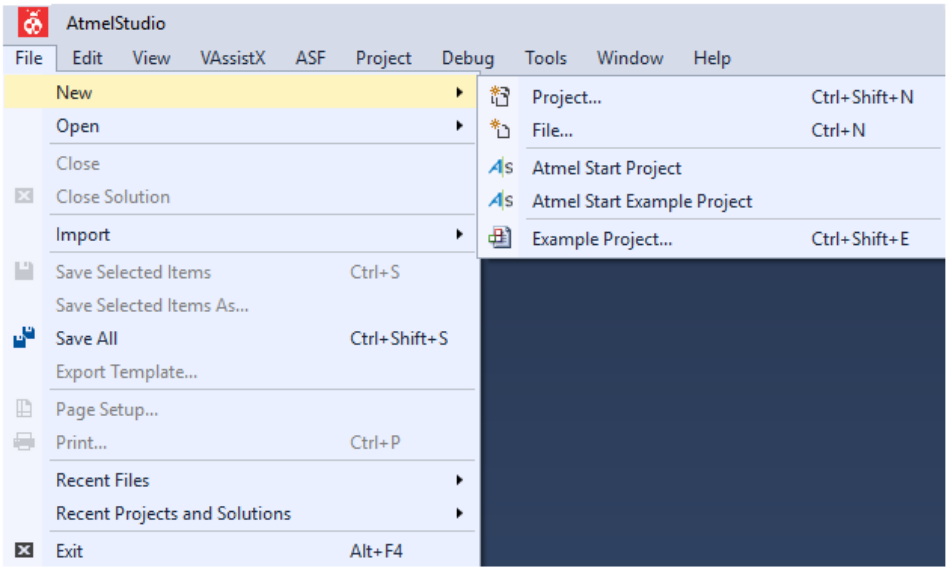
3. If there are any modified files, you will get a window with a list of the modified files. Select which files should be overwritten. Optionally, select to backup the overwritten files.



6.1.5 Creating New Project using Start

Start can be opened in a window inside Studio to generate a project or load an example project.

1. Select Atmel Start Project or Atmel Start Example Project from the **File** > **New** menu.

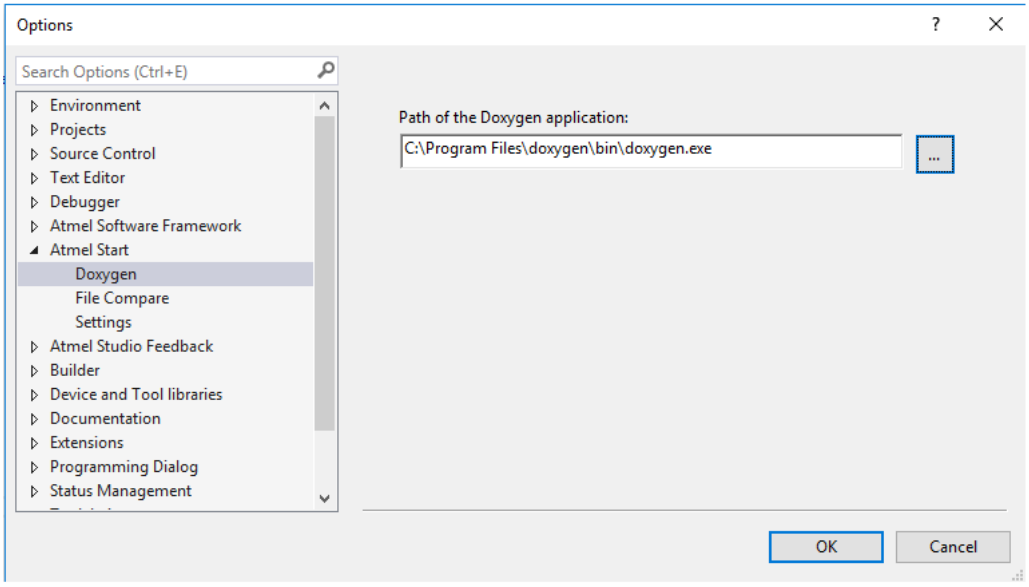


A new project is created.

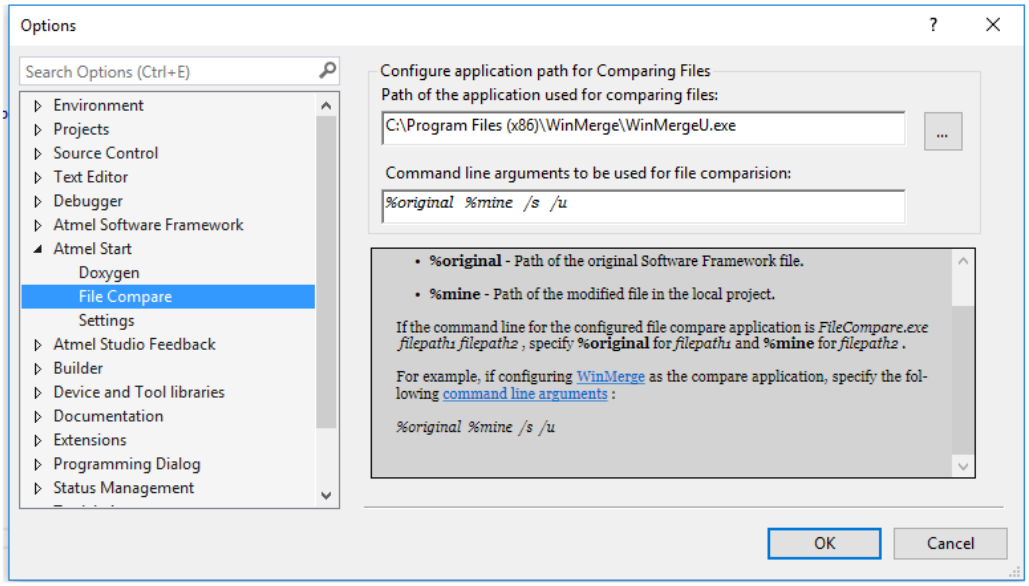
6.1.6 Settings

Settings are accessed from menu **Tools** > **Options** > **Atmel Start**.

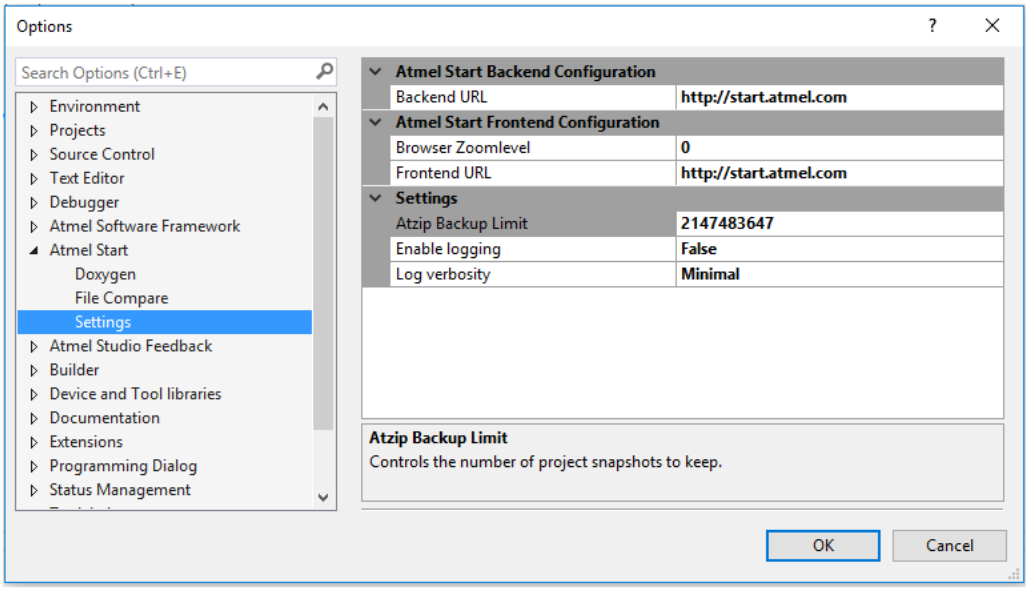
1. Set location of Doxygen executable.



2. Define tool for file comparison.



3. General settings.



6.1.7 Restore Project from Backup

A backup of the project is taken when reconfiguring with Atmel START. Backups are stored in the folder '`.atmel-start-backup`' under the project folder.

1. Backups of type '`.atzip`' are opened in Atmel Studio by using the Atmel Start Importer.
   Note: These backups have a bug: If the user has added or renamed files and folders to the backed-up project, these will be removed if the restored backup is reconfigured in Atmel START. If this happens, import the project again and avoid using Atmel START to reconfigure.

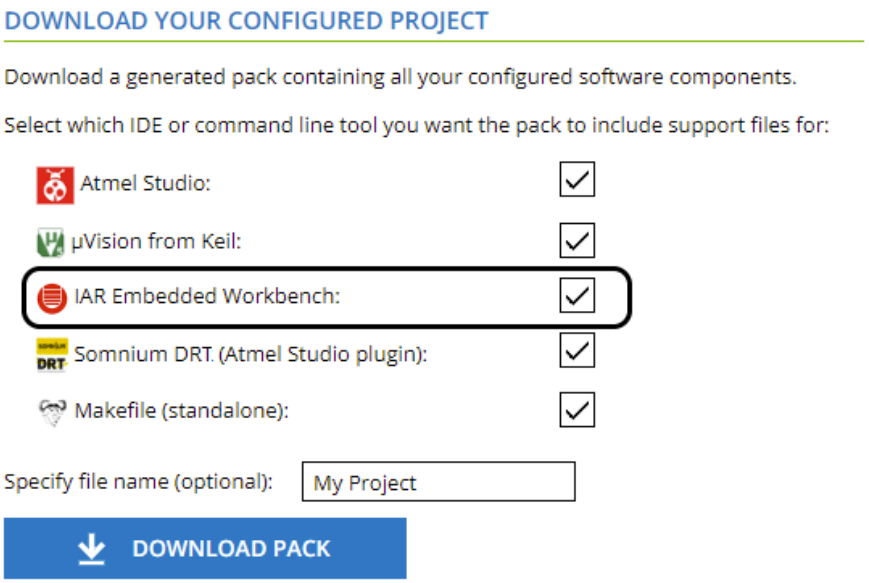
2. Backup of type '`.zip`'.
    1. Unzip to a folder.
    2. Create a new project in Atmel Studio of type **Atmel Studio Solution** > **Blank Solution** (**File** > **New** > **Project**).
    3. In Solution Explorer on the right side, select the **Solution**, right click and select **Add Existing Project**, browse to the unzipped folder and select the Atmel Studio project file.

6.2 IAR Embedded Workbench®

The following steps describe how to use the output from Atmel START in IAR Embedded Workbench. Make sure you have the latest updates of IAR Embedded Workbench installed.

6.2.1 Exporting the Project from Atmel START

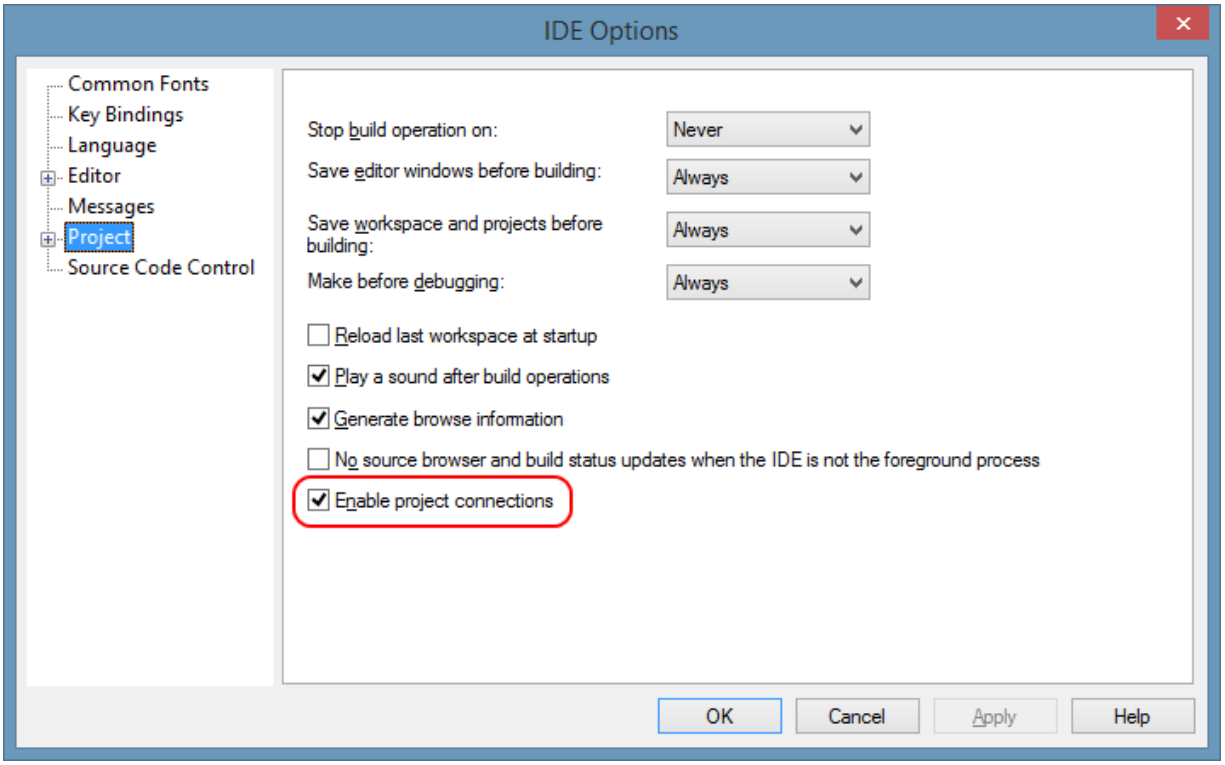
1. On the Atmel START website, create a new project (Example or Board).
2. Click on the **Export Software Component** button.
3. Make sure the "IAR Embedded Workbench" check box is checked.



4. Click on **DOWNLOAD PACK**.
   An '`atmelstart.atzip`' pack file will be downloaded.
5. Rename the downloaded file's extension to '`.zip`' (change from '`.atzip`' to '`.zip`').
6. Unzip the resulting '`atmelstart.zip`' file into a folder of your choice, where you would like to create your IAR project.

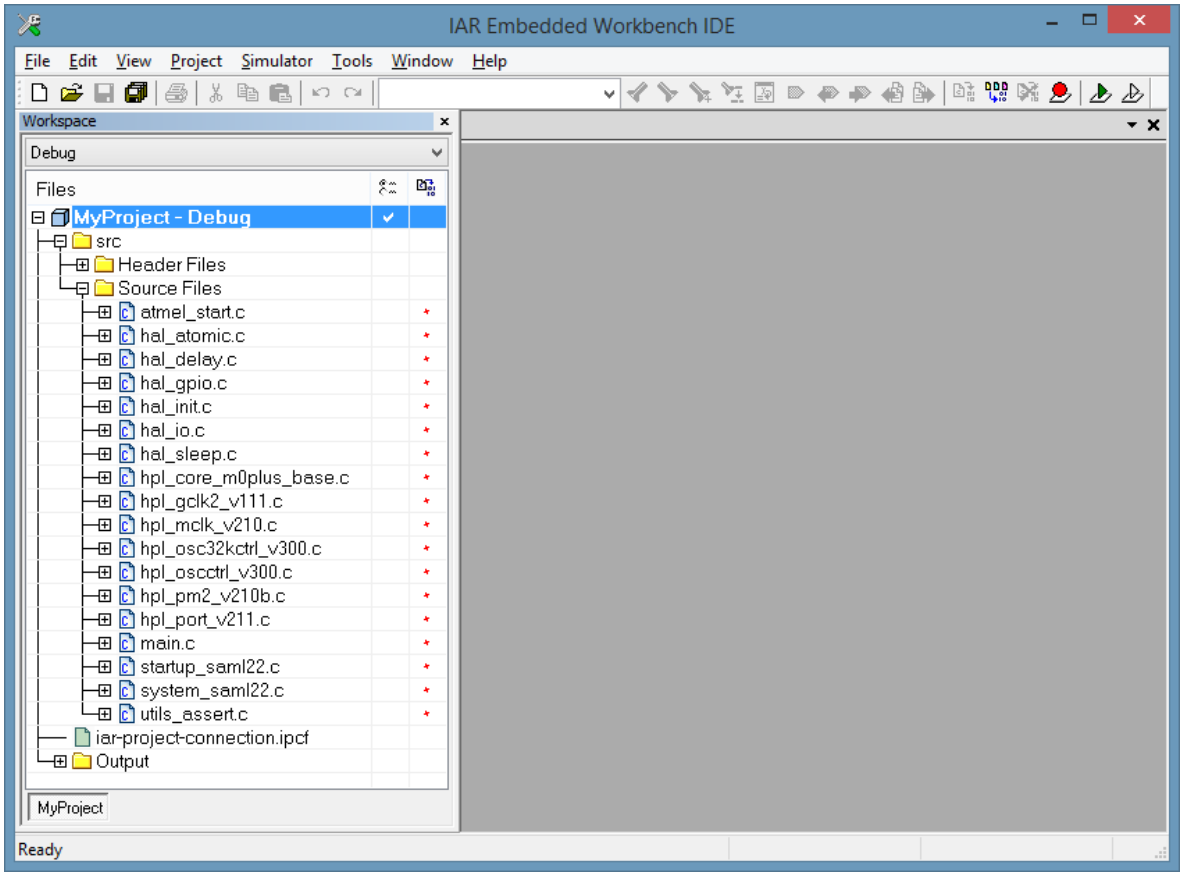
6.2.2 Open the Project in IAR Embedded Workbench®

Before importing the project, make sure that the project connections are enabled in IAR Embedded Workbench: Go to **Tools** > **Options** > **Project**, where you check the "Enable project connections" box.

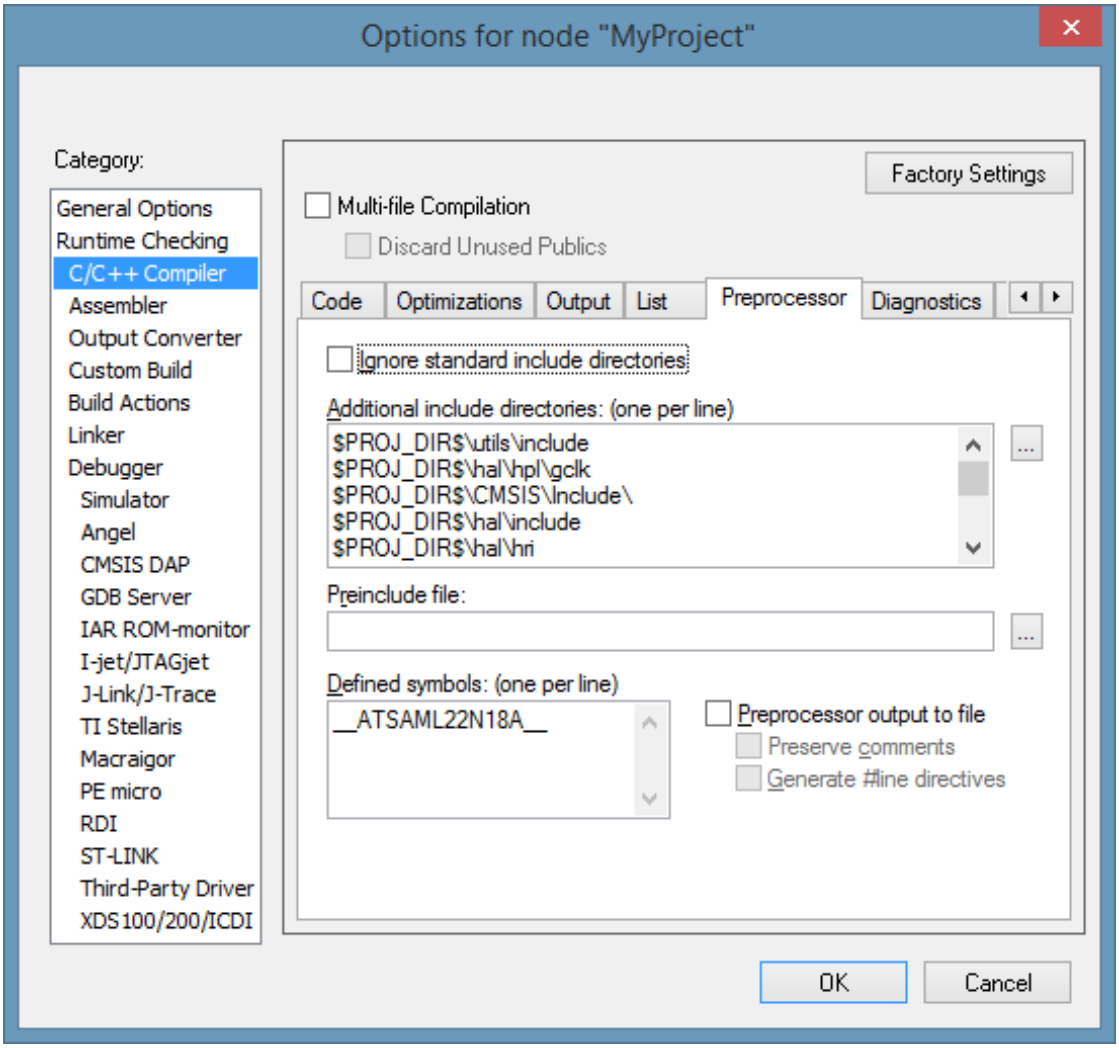


1. To start using project connections, you need to have an open project, which can be empty.
1. To create a new, empty project, go to **Project** > **Create New Project...** > **Empty Project**.

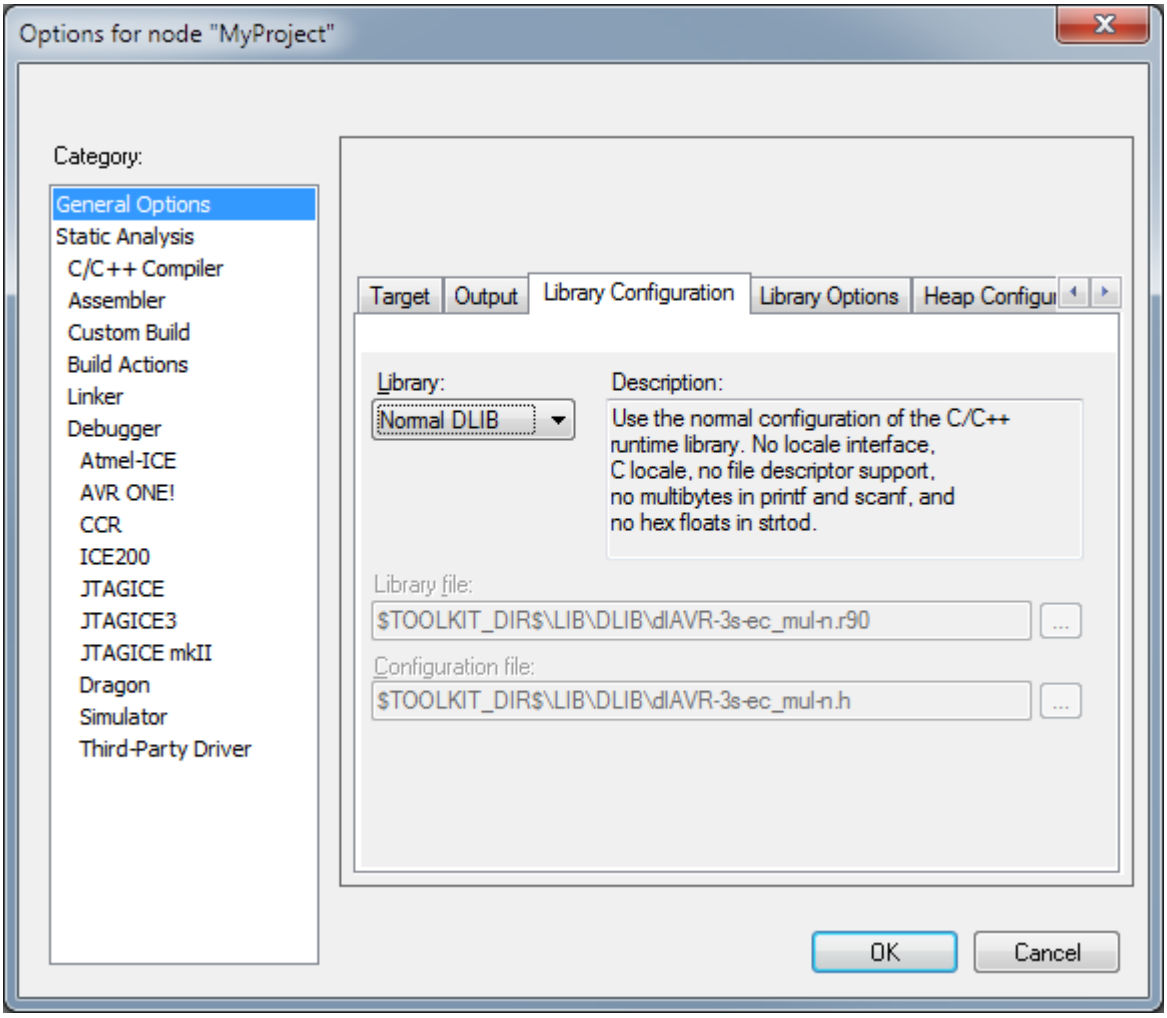
   Note: The IAR project (the .ewp file) must be created in the same folder as the downloaded `atmelstart.zip` was extracted.
2. When you have an open IAR project, you can connect it to your Atmel START project.
2. From the Project menu, you will be able to select Add Project Connection. This opens a dialog box where you can choose the external tool you want to set up a connection with. Choose IAR Project Connection and click **OK**.
3. A browser window opens. Select the `iar-project-connection.ipcf` file in the extracted folder created as in section 6.2.1 Exporting the Project from Atmel START.
   After the import of the .ipcf file, all the generated code is automatically added to the IAR Embedded Workbench project.



The required include directories (**Options** > **C/C++ Compiler** > **Preprocessor**) are added to the project as well.



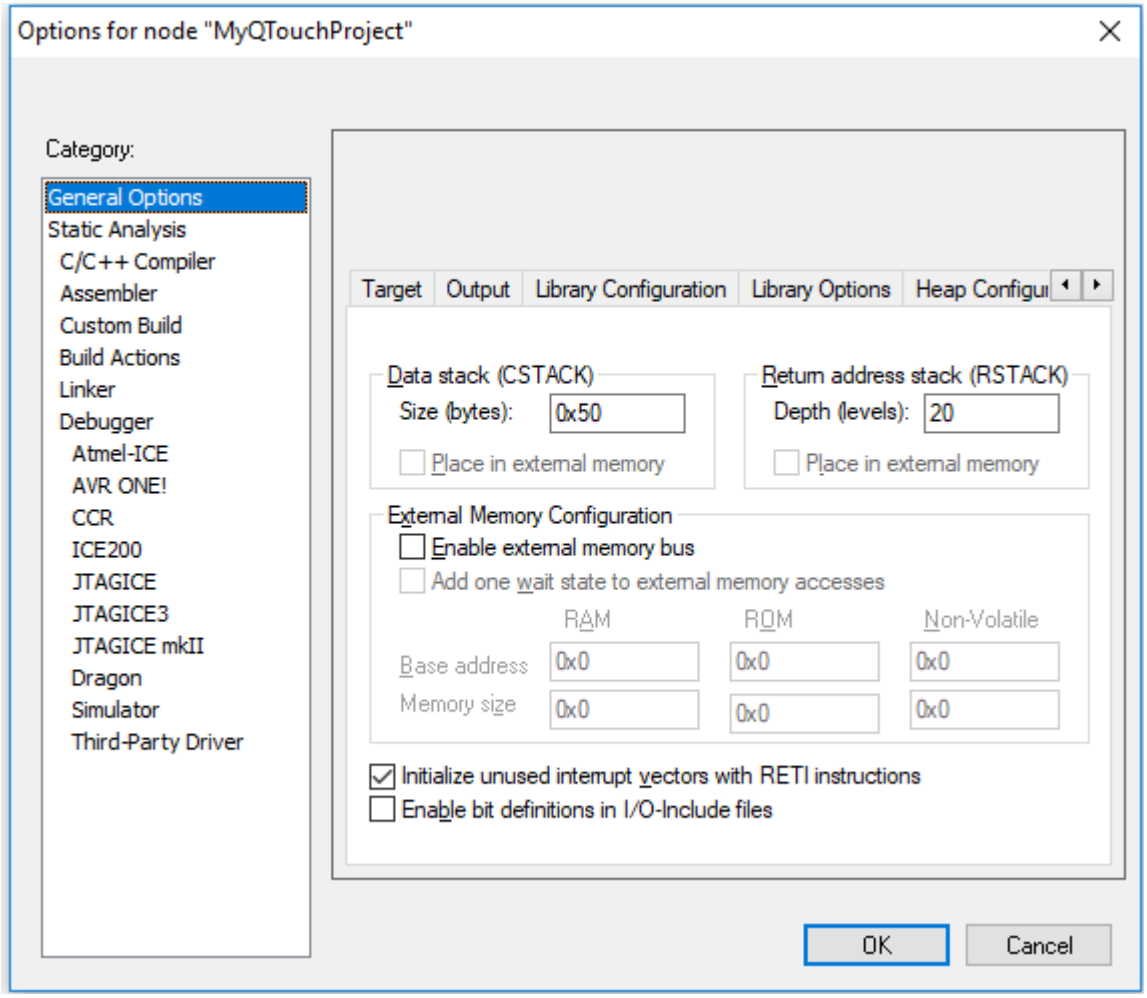
4. If the project uses an AVR 8-bit controller, the C-library must be changed. Select **Project** > **Options** > **General Options** > **Library Configuration**. Change "Library" to "Normal DLIB".



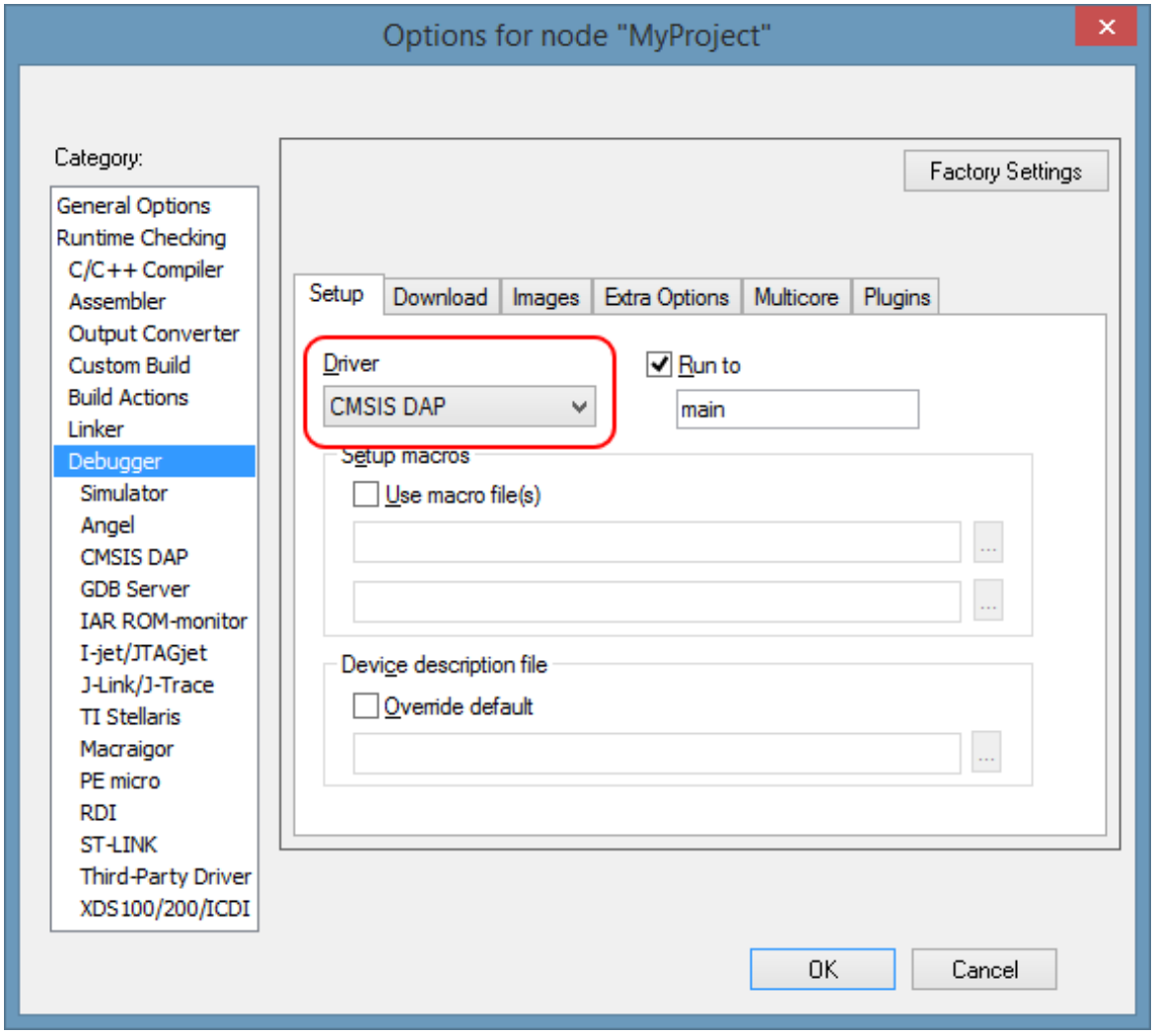
3. **Note:** Adding a project connection for a project for an AVR device may generate warnings about that IAR IDE cannot enable linker map and log files. These messages are notifications about the use of the default linker files and can be safely ignored. It is recommended to disable these notifications by checking the "Don't show again" check-box.

   Since the IAR IPCF files do not support debugger options, they must be selected manually before debugging. For Microchip Xplained boards, "CMSIS DAP" should be used to download and debug the projects correctly.

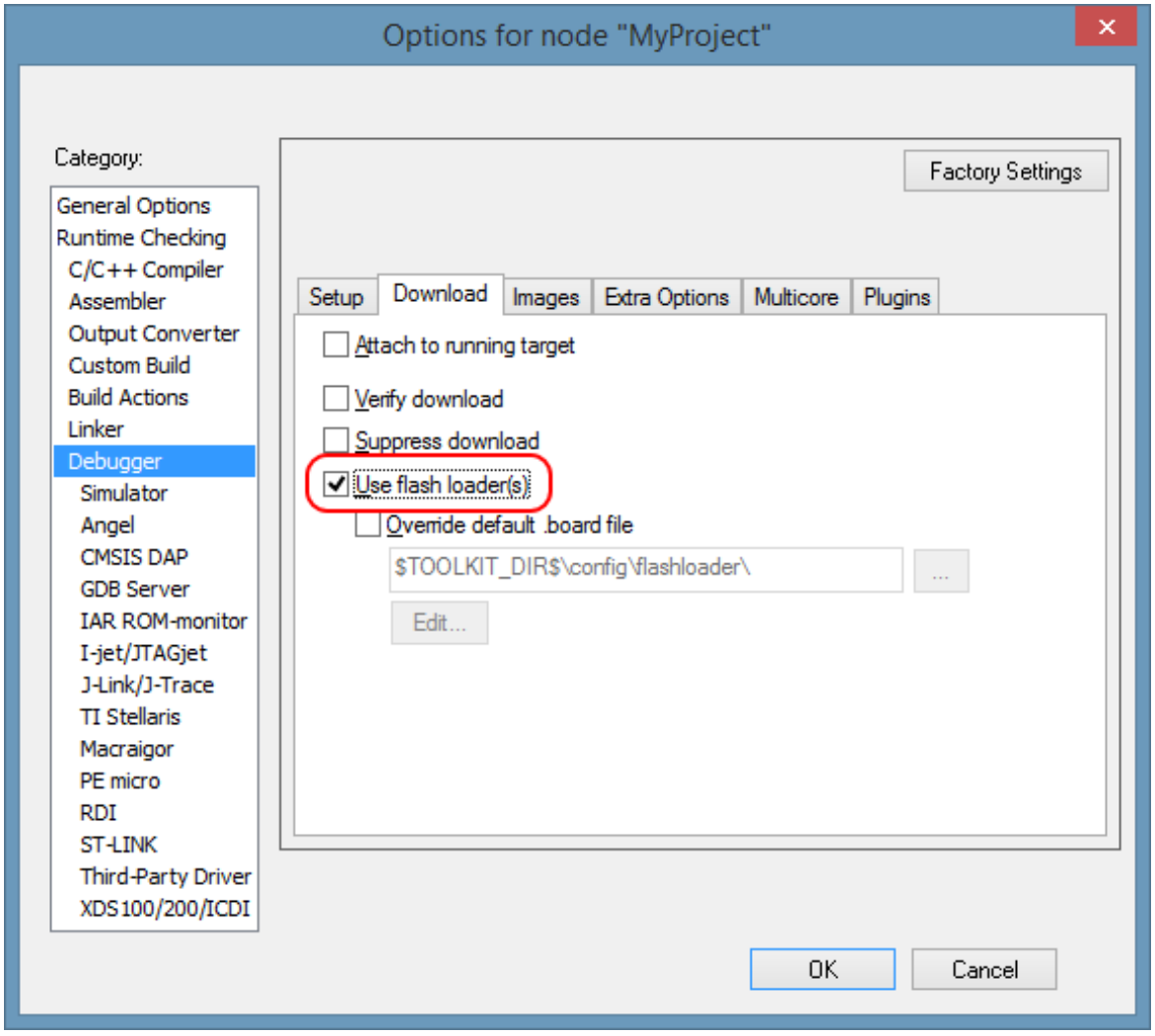
5. If the project uses **QTouch Library**, the stack memory set by default is not sufficient, change CSTACK and RSTACK.



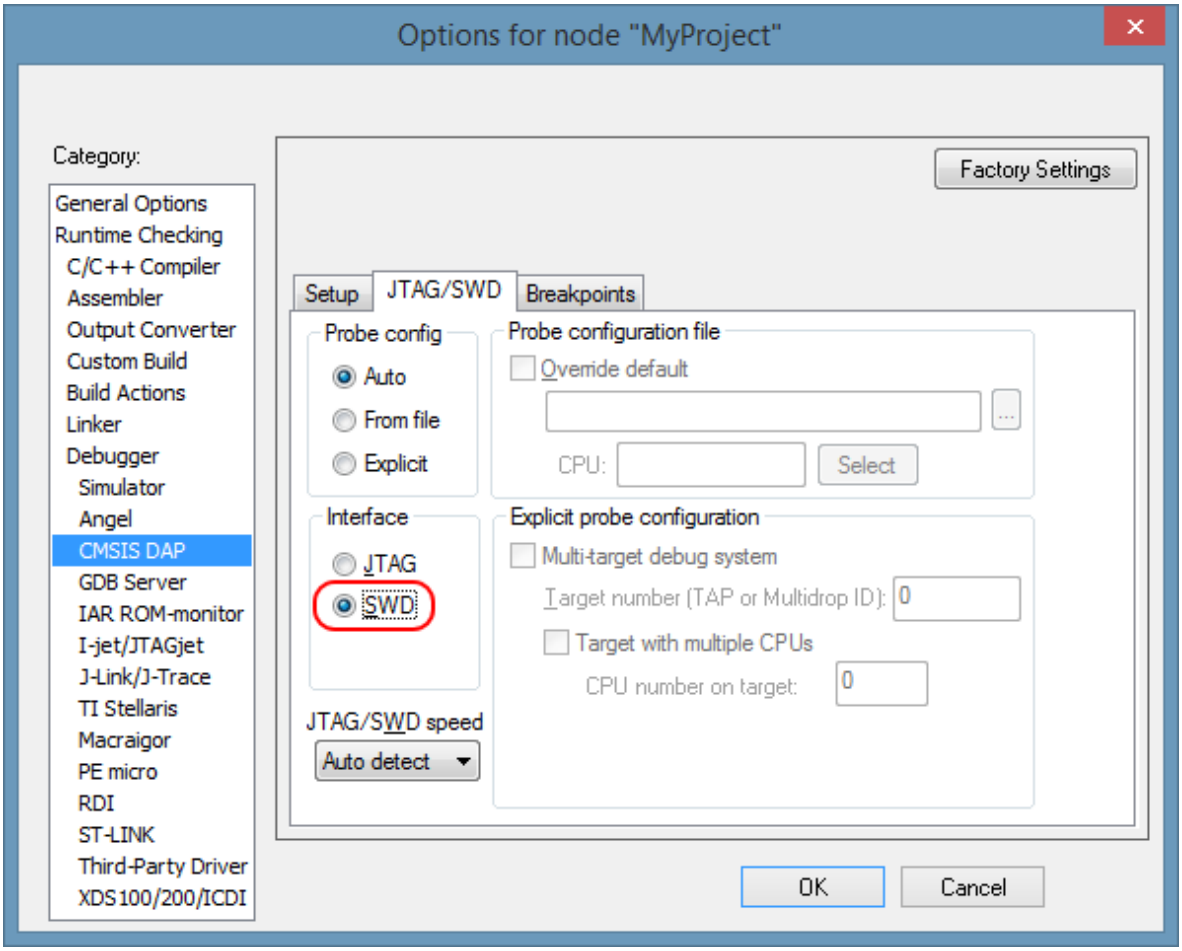
6. Select **Project** > **Options** > **Debugger** > **Setup**. Change "Driver" to "CMSIS DAP".



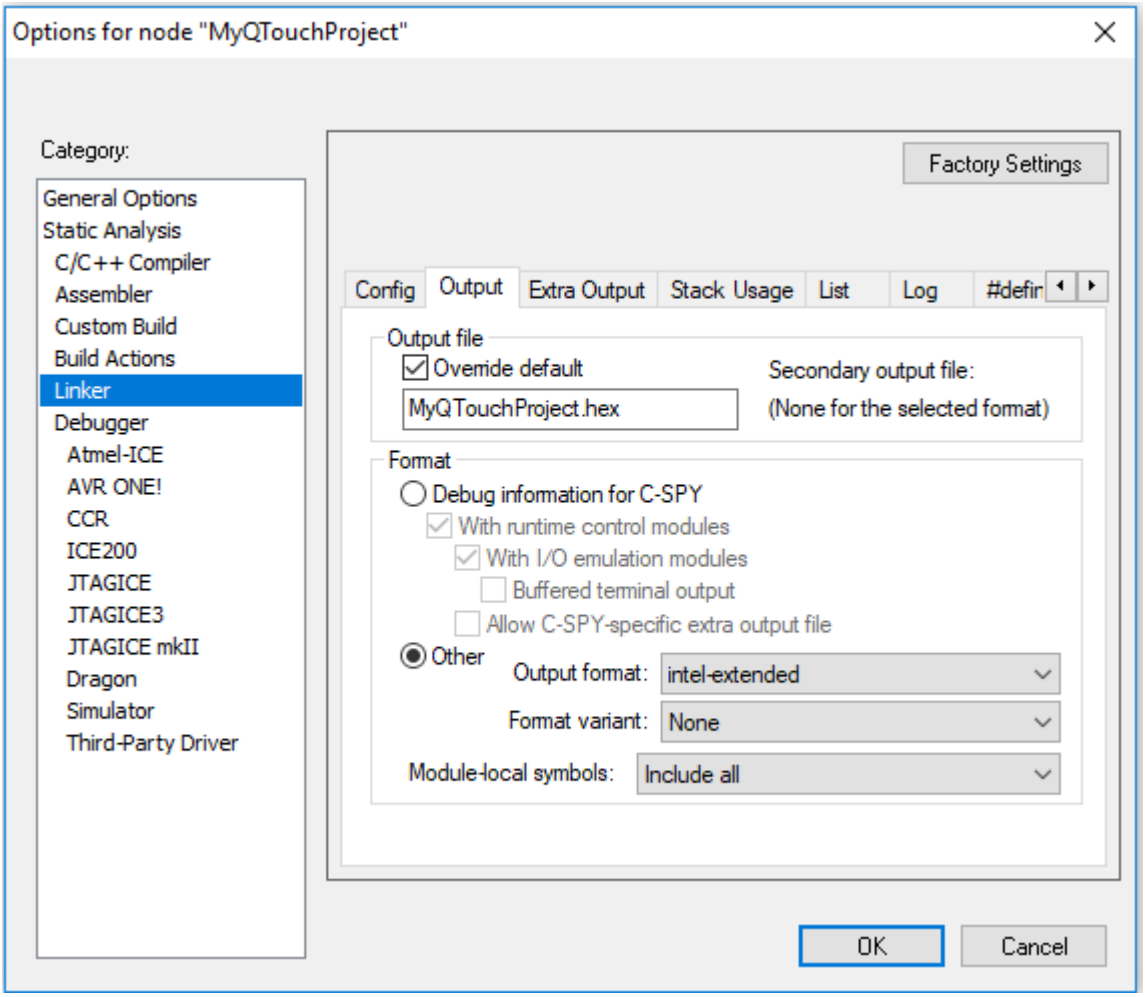
7. Click on the **Download** tab. Make sure the "Use flash loader(s)" is checked.



8. Select CMSIS-DAP category, go to the **JTAG/SWD** tab, and set Interface to "SWD". Click **OK**.



9. If the project uses **QTouch Library**, the linker output format must be set to Intel hex. **Project** > **Options** > **Linker** > **Output**. Click on the "Override default" box and change the output file extension to '`.hex`'. Under Format, select Others and select '`intel-extended`' from the drop-down menu.

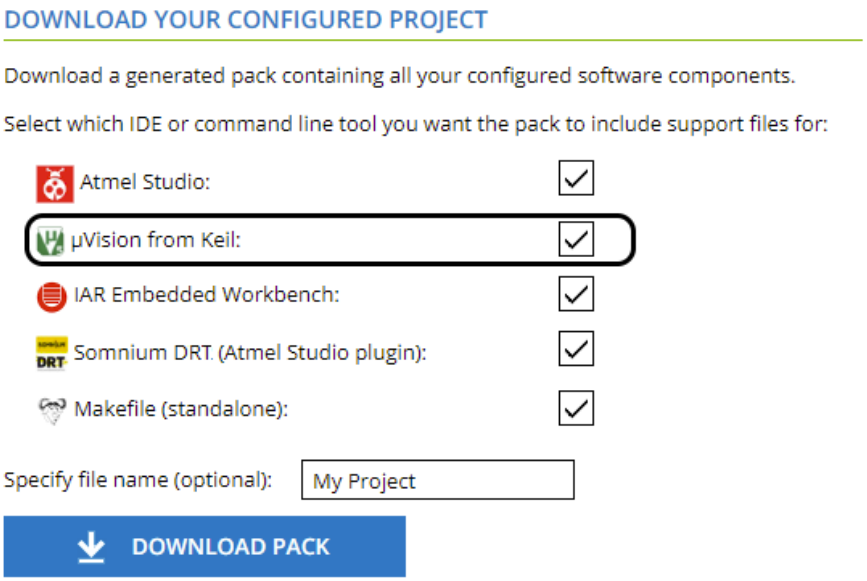


6.3 Keil® µVision®

The following steps describe how to use the output from Atmel START in Keil µVision.

6.3.1 Exporting the Project from Atmel START

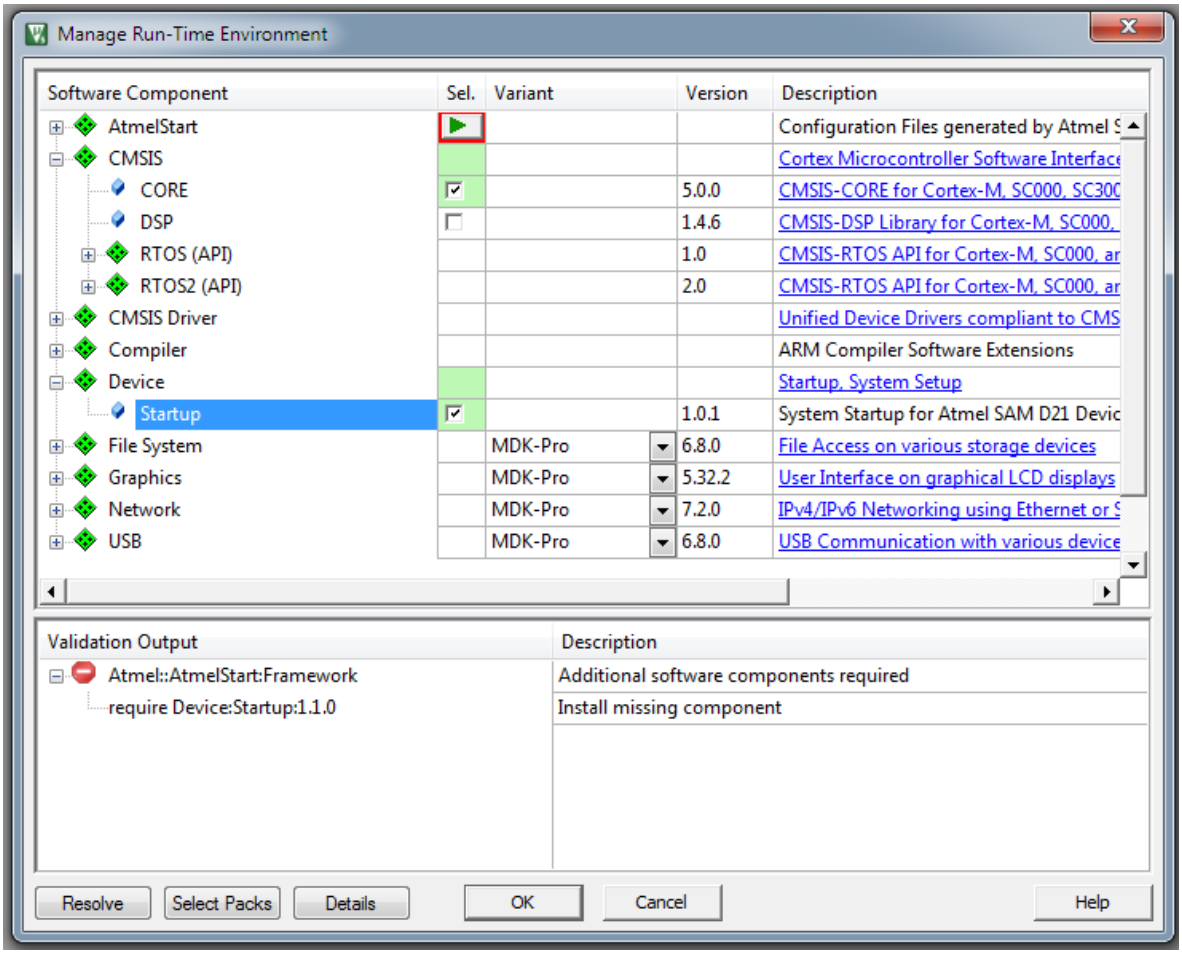
1. On the Atmel START website, create a new project (Example or Board).
2. Click on the **Export Software Component** button. Make sure the "µVision from Keil" check box is checked.



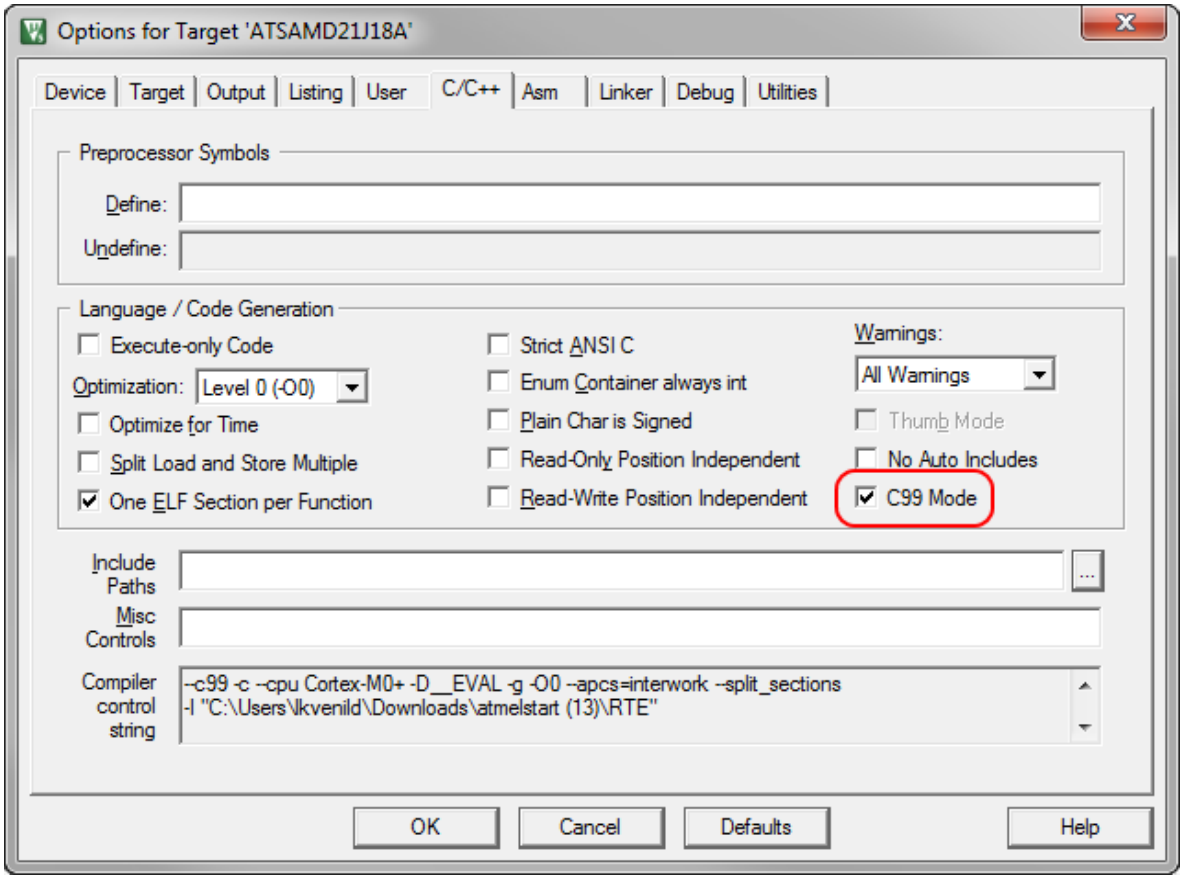
3. Click on **DOWNLOAD PACK**.
   An '`atmelstart.atzip`' pack file will be downloaded.
4. Rename the downloaded file's extension to '`.zip`' (change from '`.atzip`' to '`.zip`').
5. Unzip the resulting '`atmelstart.zip`' file into a folder of your choice, where you would like to create your project.

6.3.2 Open the Project in Keil® µVision®

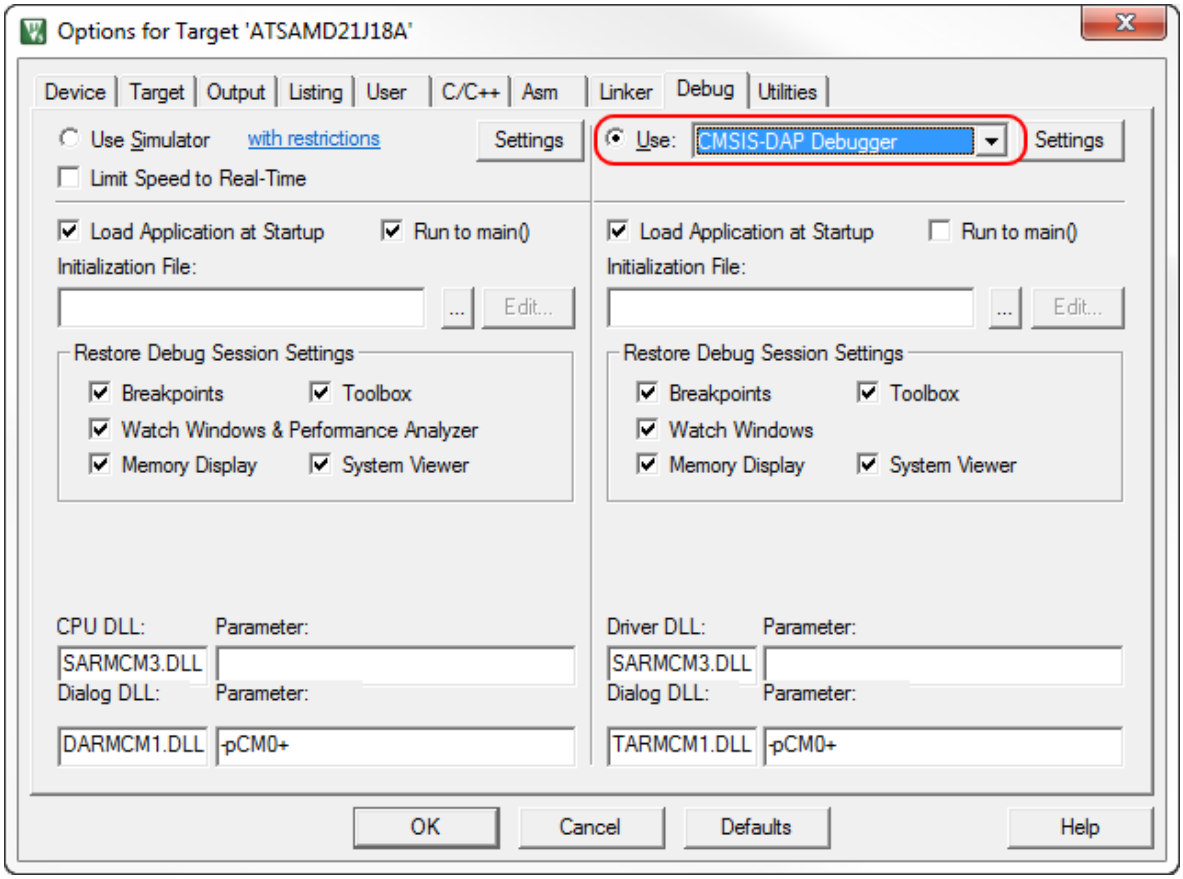
1. Start **Keil µVision**, and select **File** > **Open**. Change the file type filter to show Generator Pack Description files, (*.gpdsc). Select the '`AtmelStart.gpdsc`' file from the folder as described in section 6.3.1 Exporting the Project from Atmel START. Click **Open**.



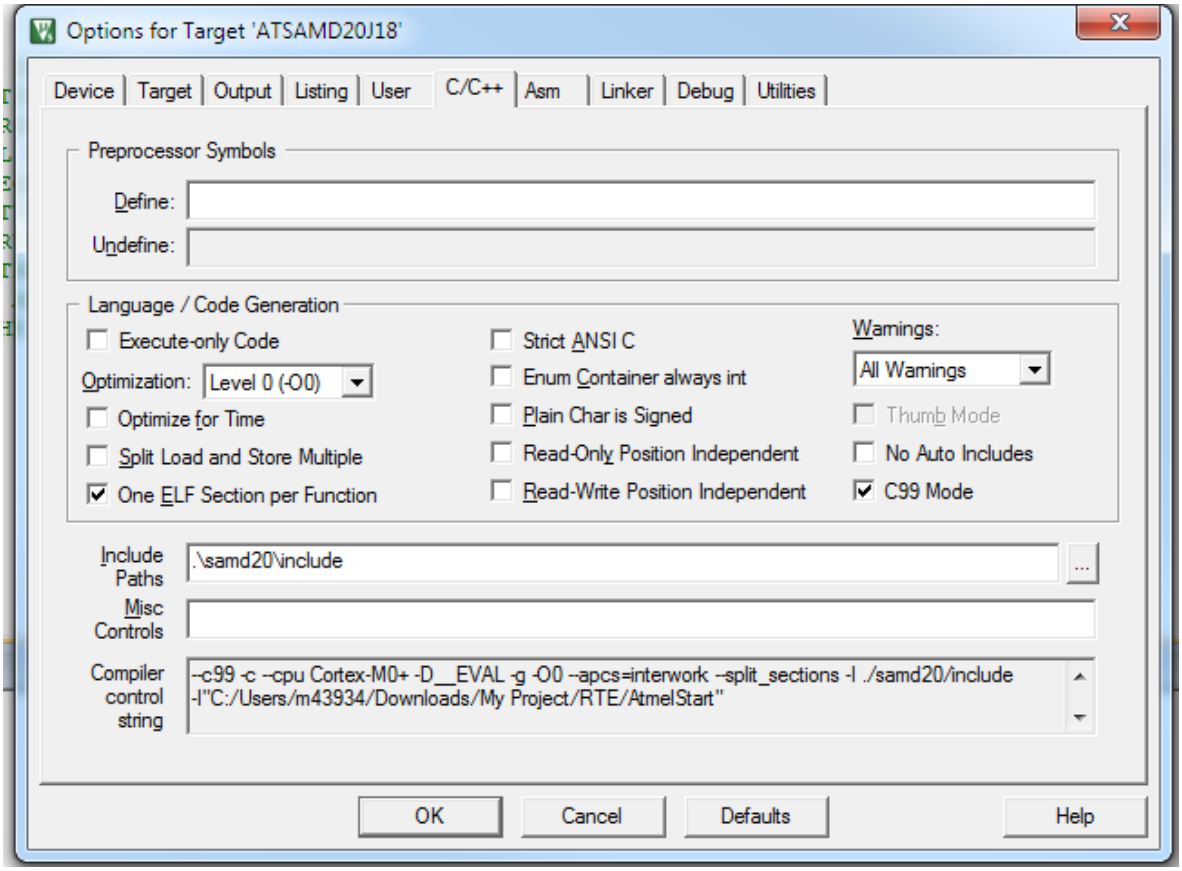
2. Make sure **CMSIS** > **Core** and **Device** > **Startup** are checked.
1. Since the GPDSC does not contain project settings, and the generated code needs Keil C/C++ "C99 Mode" to be compiled (CMSIS exceptions are allowed in coding), it should be enabled after opening the GPDSC with Keil.
3. Select **Project** > **Options for target...** > **C/C++**.
4. Make sure the "C99 Mode" check box is checked. Click **OK**.



2. Before debugging your project on the Xplained boards, change the settings to use a CMSIS-DAP Debugger:
5. Select **Project** > **Options for target…** > **Debug**.
6. Select **CMSIS-DAP Debugger** from the Use list. Click **OK**.



7. Optional: If the project compiles with error '*cannot open source input file "`sam.h`"* add device-specific include path to C/C++ input paths, like '`.\samd20\include`' for SAM D20 device.

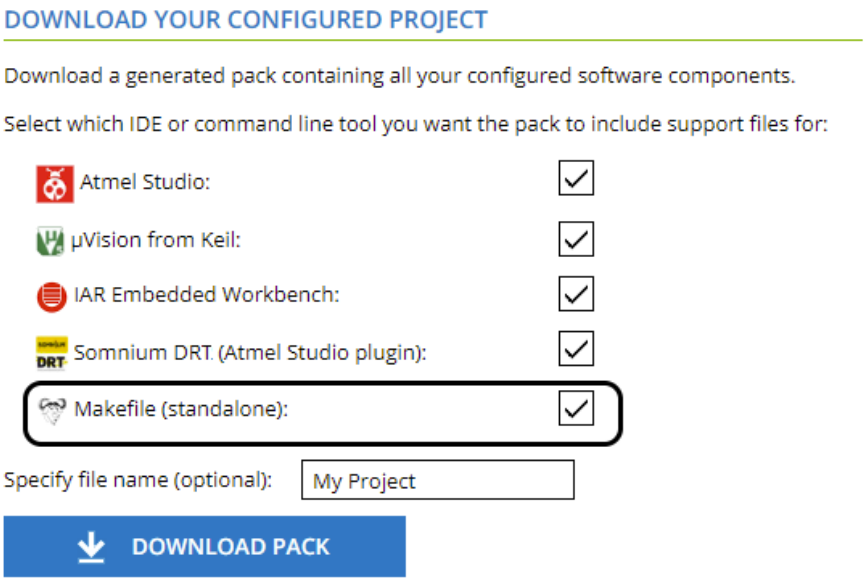


6.4 GNU C Makefile

The output from Atmel START can be used with GNU C using a Makefile.

6.4.1 Exporting the Project from Atmel START

1. On the Atmel START website, create a new project (Example or Board).
2. Click on the **Export Software Component** button.
3. Make sure the "Makefile" check box is checked.



4. Click on **DOWNLOAD PACK**.
   An '`atmelstart.atzip`' pack file will be downloaded.
5. Rename the downloaded file's extension to '`.zip`' (change from '`.atzip`' to '`.zip`').
6. Unzip the resulting '`atmelstart.zip`' file into a folder of your choice, where you would like to create your GNU C project.

6.4.2 Executing a GNU C Makefile

Make sure you have installed a GNU C compiler supporting your selected device. The C compiler must be present in the PATH. Also, the '`make`' utility should be installed and present in your PATH.

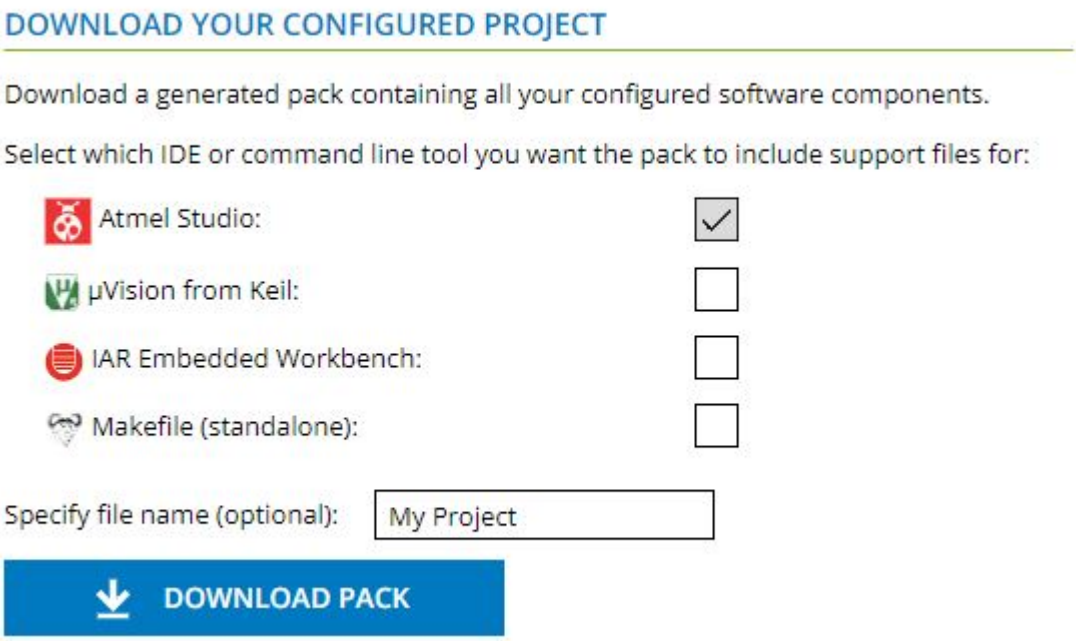
1. If you checked the "Include Makefile" check box when exporting the project from Atmel START, the extracted '`atmelstart.atzip`' file should contain a Makefile inside the '`gcc`' folder.
1. In the folder where you extracted the '`atmelstart.atzip`' file, go to the '`gcc`' sub-folder and open a command prompt.
2. Run `make`.
   This will build your project.

6.5 SOMNIUM DRT

The following steps describe how to use the output from Atmel START in SOMNIUM DRT.

6.5.1 Exporting the Project from Atmel START

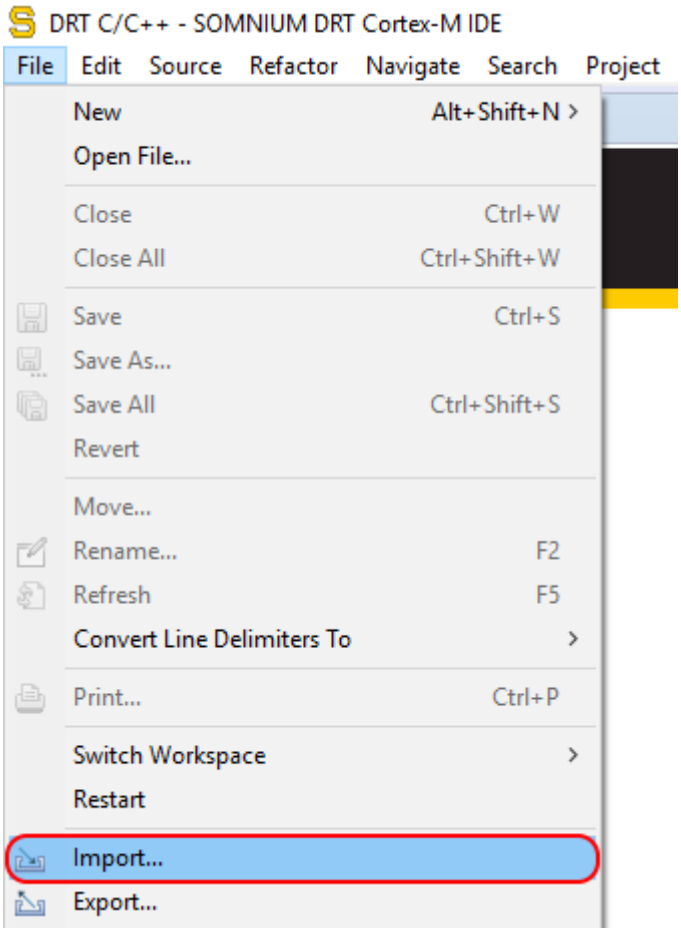
1. On the Atmel START website, create a new project (Example or Board).
2. Click on the **Export Software Component** button. For exporting project for Atmel Studio and Somonium IDE, select Atmel Studio.



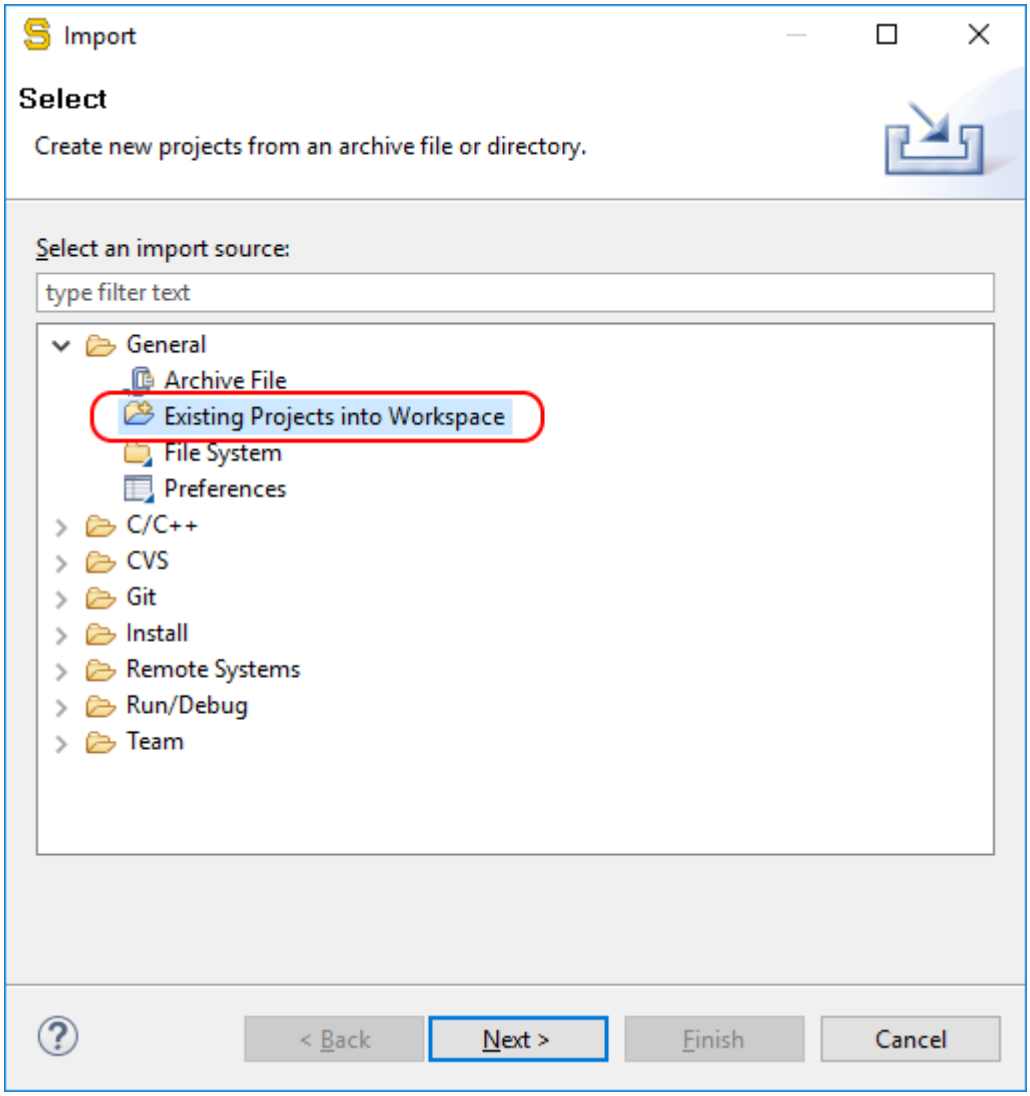
3. Click on **DOWNLOAD PACK**.
   An '`atmelstart.atzip`' pack file will be downloaded.

6.5.2 Steps to Import Atmel START Output into SOMNIUM DRT

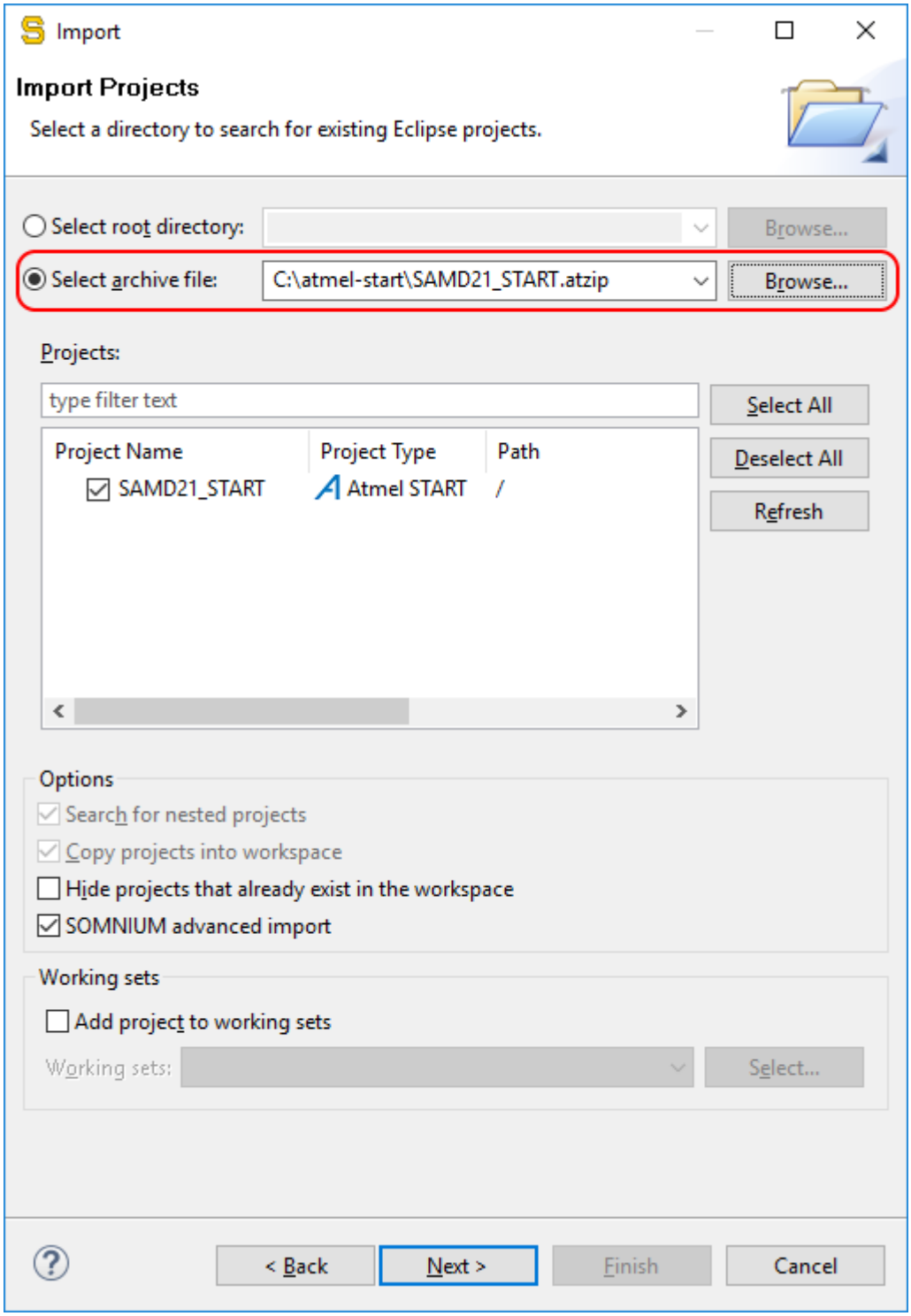
1. Launch **SOMNIUM DRT**.
2. Select **File** > **Import** to open the import dialog.



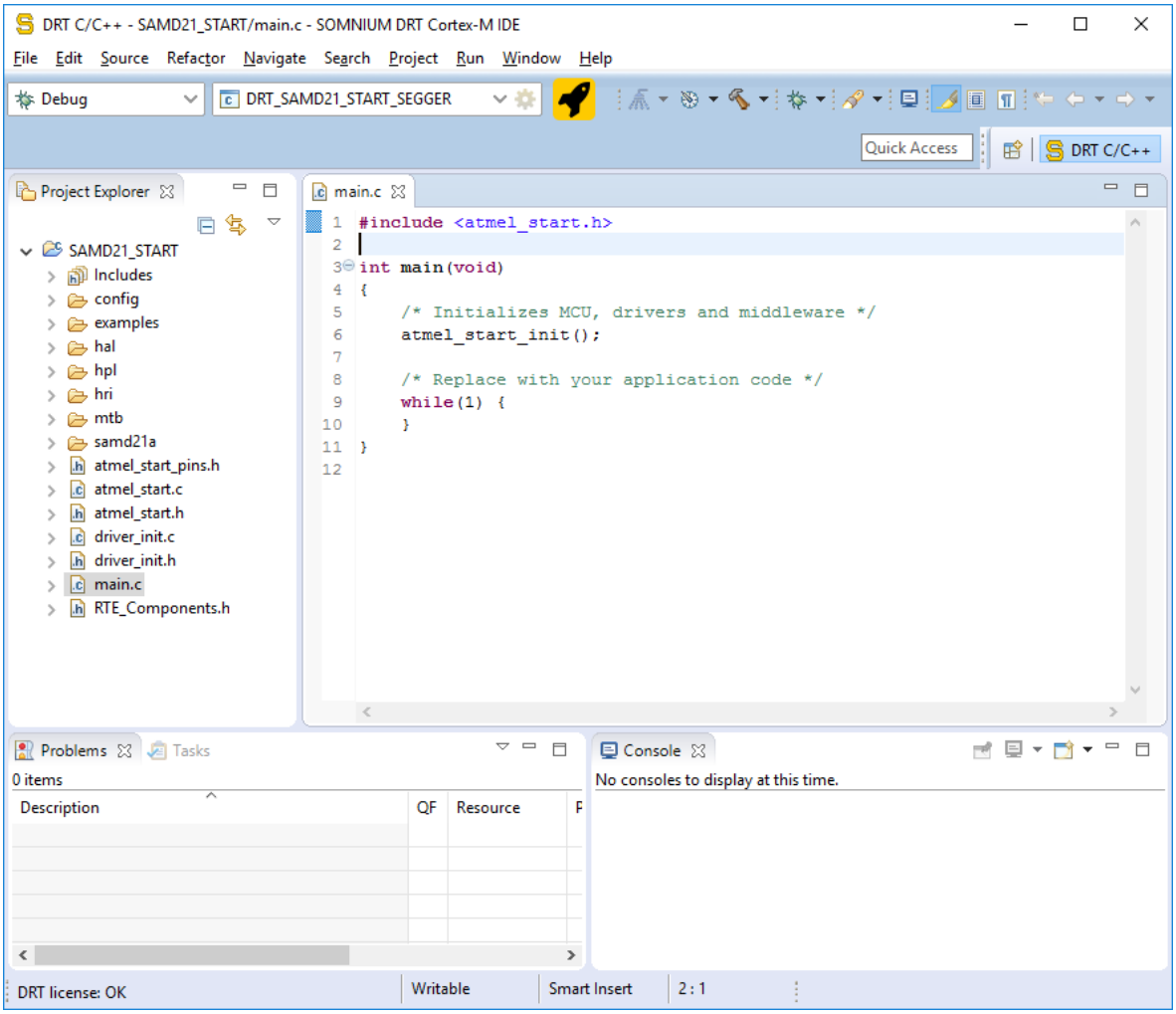
3. In the General folder select Existing Projects into Workspace and click **Next**.



4. Choose the **Select Archive File** radio button and browse to the '`.atzip`' file exported from Atmel START.



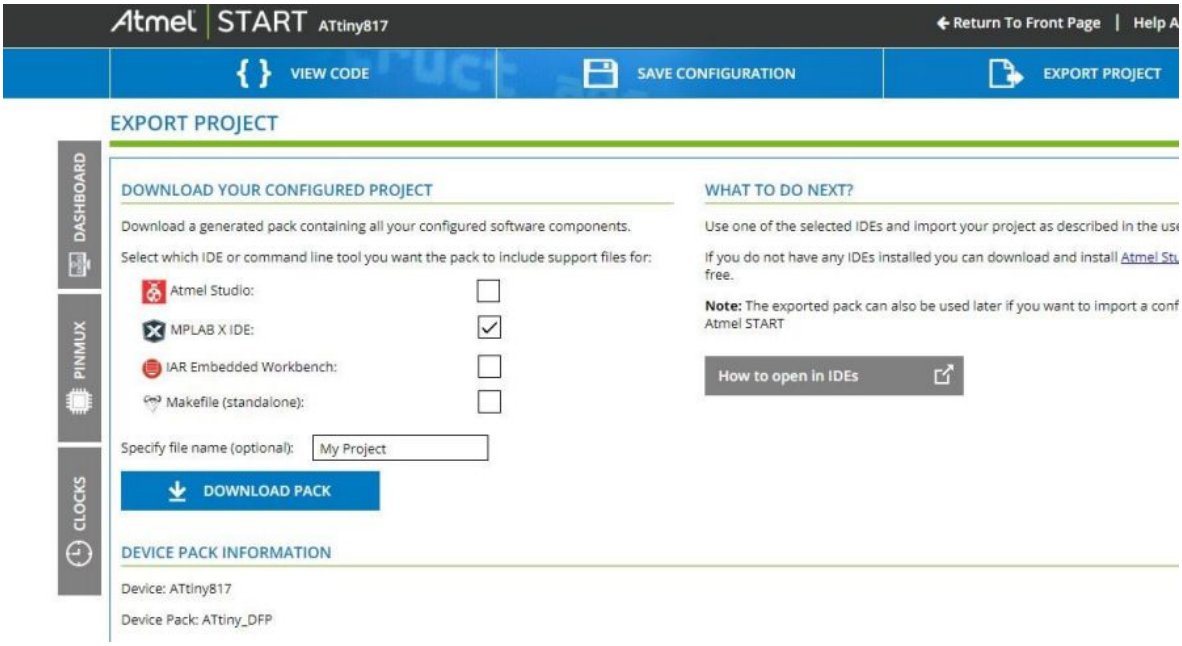
5. Click **Next**, then **Finish**, and a new DRT project will be created in the workspace with the imported files.



6.6 MPLAB™ X IDE

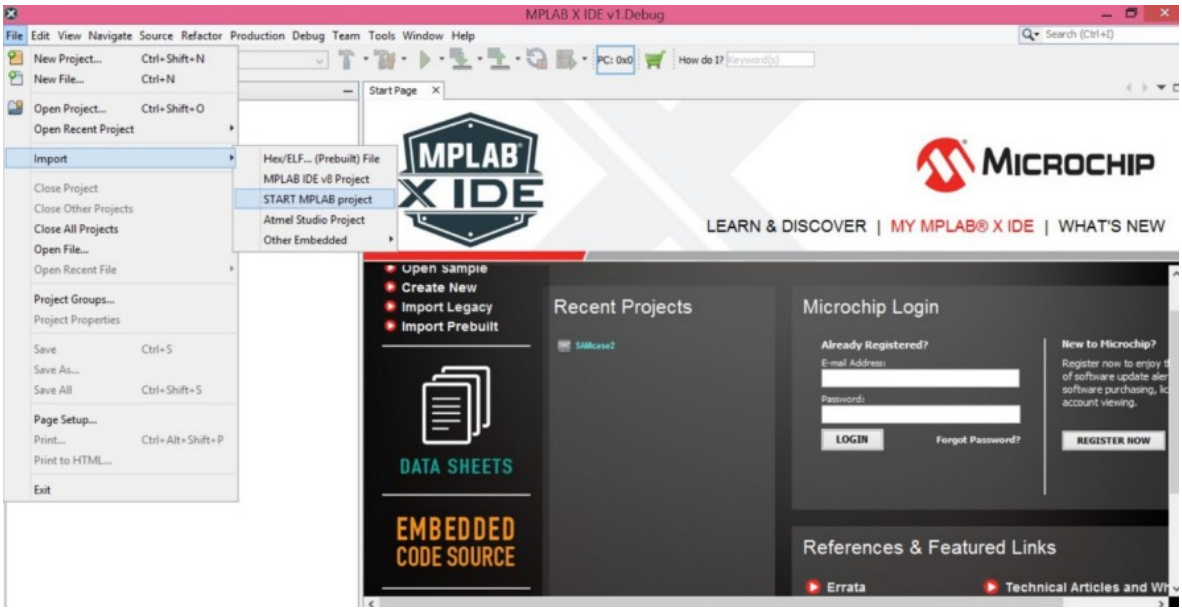
6.6.1 Creating an MPLAB™ X Project

1. On the Atmel START website (http://start.atmel.com/), create a new project (Example or Board).
2. Once the project is created, go to the **EXPORT PROJECT** page and check the "MPLAB X IDE" box.

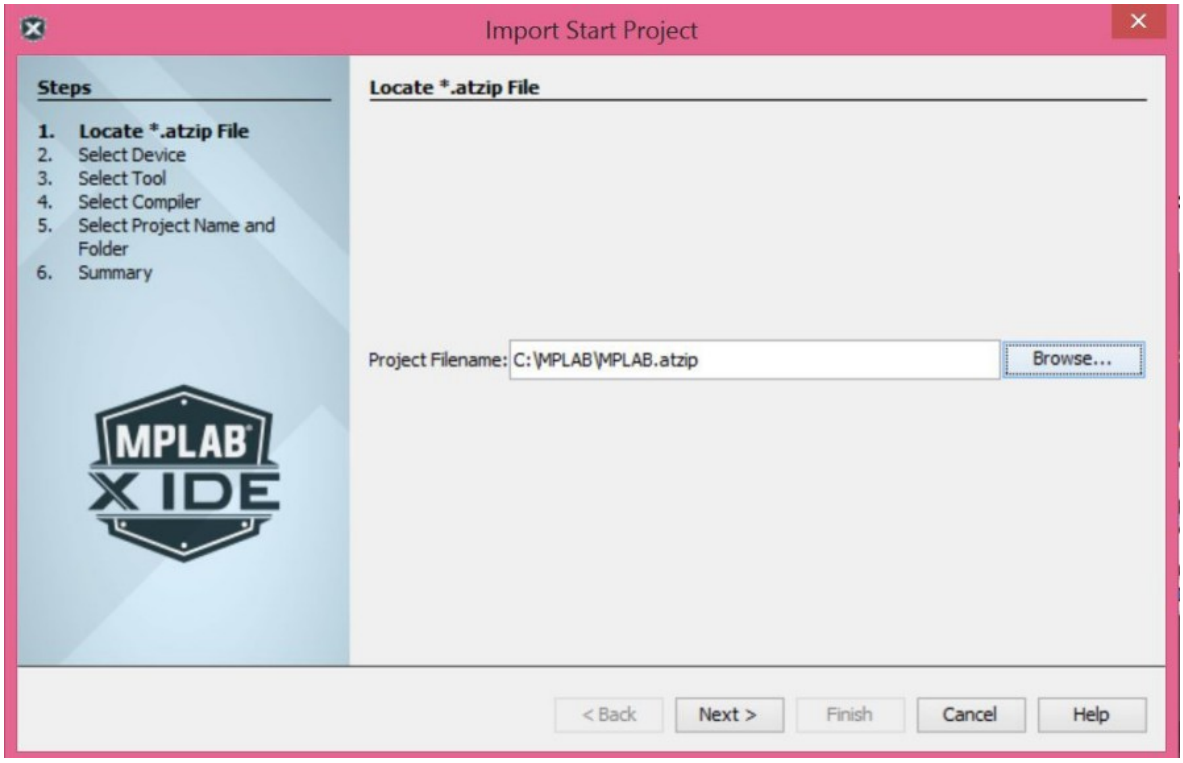


6.6.2 Importing Atmel Start Project in MPLAB™ X

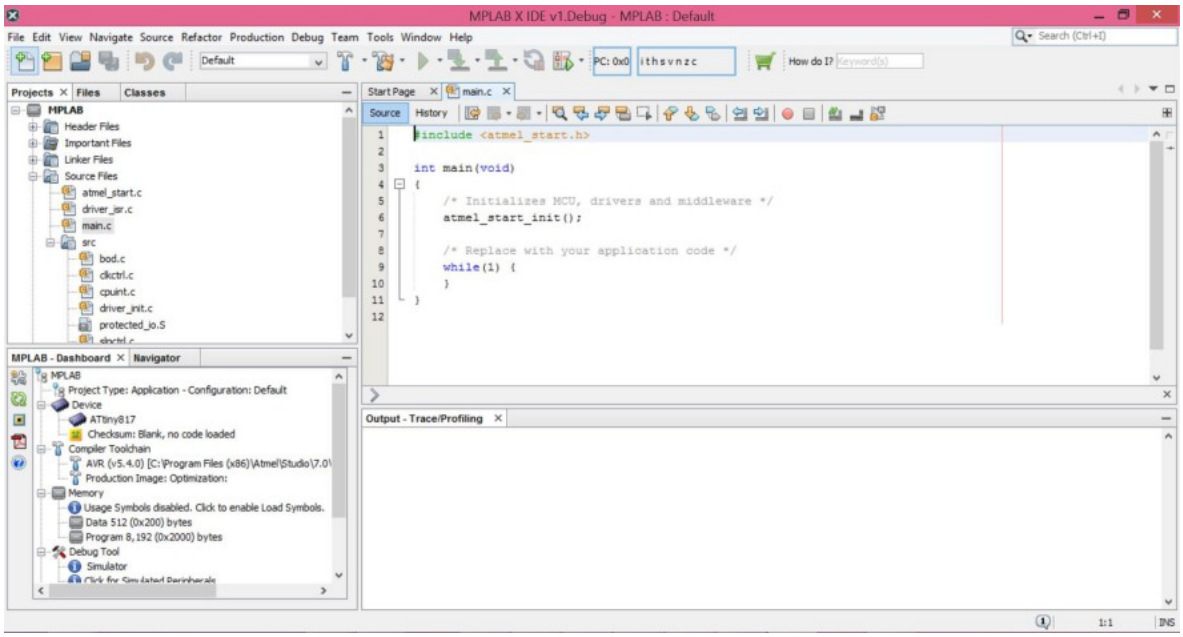
1. Select **File** > **Import** > **START MPLAB project** as shown below:



2. Browse the '`.atzip`' file, as shown below:



3. Follow the procedures described in the wizard.
4. An MPLAB X project will get created.



7. Content in Atmel START

Atmel START is a web-based platform for configuring and distributing content, meaning device drivers, middleware and demo applications for Microchip's AVR and ARM devices. Content for AVR devices is in a framework called AVR Code, and content for ARM devices is in a framework called ASF4.

The START web interface makes configuration of drivers and applications simple so that even new users can configure a system and get a system up and running in a short time. In the initial phase of generating a project in START, the user chooses a device. START automatically selects the appropriate framework depending on the chosen device.

7.1 ARM Content

In Atmel START, the drivers and software stacks for ARM devices are provided as a part of the next iteration of ASF4. This version is built from scratch and is a complete re-design and re-implementation of the whole framework to resolved issues reported by users and contributors of older ASF versions, and to better integrate with the START web UI. It has still been a goal to keep ASF4 feel familiar for experienced ASF users, yet easy to get started with for new users. Some changes in ASF4 has been necessary to meet the requirements for this version, the most important changes are listed in the ASF4 API manual.

ASF4 is tightly integrated into START, which means that the ASF4 code can be much more tailored to the users' specification than before. For instance, instead of using C preprocessor conditional expressions to enable/disable code blocks, disabled code blocks can be entirely removed from the project source which results in cleaner and easier to read code. The integration into START means that software configuration is done in a much more user-friendly environment and the only configuration information loaded on the device is the raw peripheral register content, which makes the firmware image much more compact.

One important issue we have addressed is the memory footprint and performance of the ASF4 based code. Flash requirements for running ASFv3 based code has been deemed too high by many users. This has been addressed by using code generation and changing the way peripherals are initialized. Performance issues that have been reported are typically high interrupt latency/slow code execution, which has been resolved by making the interrupt handlers smaller and less complex. Some benchmark numbers comparing ASF4 and ASFv3 can be found in the ASF4 API reference manual.

Detailed description of ASF4 can be found in the ASF4 API Reference Manual.

7.2 AVR® Content

Atmel START can be used to generate code for AVR devices.

Hardware drivers, middleware drivers and demo applications for AVR devices are collected in the AVR Code framework. AVR Code is tightly integrated into START, which means that the code can be tailored to the users' needs. For instance, instead of using C preprocessor conditional expressions to enable/disable code blocks, disabled code blocks can be entirely removed from the project source which results in cleaner and easier to read code. The integration into START means that software configuration is done in a user-friendly environment and the generated code is tighter connected to the raw hardware, which makes the firmware image much more compact.

AVR Code is the AVR equivalent of ASF4 for ARM devices. AVR devices typically have more limited resources than ARM devices, and this is reflected in AVR Code. Using START to configure an application and export it to the IDE of choice works in the same way for both AVR and ARM devices.

The AVR Code framework is guided by the following priorities:

1. Code size.
2. Code speed.
3. Simplicity of code.
4. Readability/understandability of code.

Detailed description of AVR Code can be found in the AVR Code reference manual.

8. Keyboard Shortcuts

The following keyboard shortcuts are supported by Atmel START.

- <A> - Open the Add Software Component dialog
- <D> - Switch to the main dashboard
- <P> - Switch to the PIN/MUX configurator
- <C> - Switch to the clock configurator

9. Linking to Atmel START

You can link directly to devices, boards, and projects in Atmel START from other web pages.

Dashboard
Open the dashboard with the project last used in your browser if available.

- Link: http://start.atmel.com/#dashboard

Open a new project with a specified device
Open the new project page with the search field set to the specified device. You can specify a device identifier or a more generic search term. The search result list will show one or more matching devices.

- Format 1: `http://start.atmel.com/#device/[device identifier]`
- Format 2: `http://start.atmel.com/#device/[device search]`
- Example 1: http://start.atmel.com/#device/ATSAMD21J18A
- Example 2: http://start.atmel.com/#device/SAMD21

For a complete list of device identifiers, see http://start.atmel.com/api/v1/devices.

Open a new project with a specified board
Open the new project page with the search field set to the specified board. Just as for devices, you can specify a more or less accurate search term.

- Format: `http://start.atmel.com/#board/[board identifier]`
- Example: http://start.atmel.com/#board/SAML21XplainedPro

For a complete list of board identifiers, see http://start.atmel.com/api/v1/boards.

Open a specific example project
Open the dashboard with the specified example.

- Format: `http://start.atmel.com/#example/[example identifier (URL encoded)]`
- Example: http://start.atmel.com/#example/Atmel%3AApplication%5FExamples%3A0%2E0%2E1%3A%3AApplication%3ALED%5Fswitcher%3A

You can get the link for a given example by opening it. Then open the example configuration (click on it in the **DASHBOARD** tab) and select **Link** from the General section.

Open the example browser with a predefined search
Open the browse examples page with the search field set to the specified text.

- Format: `http://start.atmel.com/#examples/[wildcard]`
- Example: http://start.atmel.com/#examples/LED

10. Revision History

Doc Rev.	Date	Comments
A	08/2018	Initial document release.
B	09/2018	• Updated “Importing the Secure Project in µVision® from Keil®”. Aligned the images. • Updated “Importing the Non-Secure Project in µVision® from Keil®”. Aligned the images. • Updated “MPLAB® X IDE”. Extended support for exporting MPLAB X projects for SAM devices.

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

© 2018 Microchip Technology Inc.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2018, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-3556-3

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820