User's Guide

Atmel START to MPLAB Harmony 3 Migration Guide

Introduction

This document describes the different steps to consider when porting an embedded application from Atmel® START to the MPLAB® Harmony 3 framework. It covers the following migration aspects:

- How to create a MPLAB Harmony 3 project under MPLAB X IDE and navigate through the MPLAB Harmony Configurator (MHC).
- How to identify different elements that compose a project under Atmel START.
- How to port the system/driver/middleware configuration using MPLAB Harmony Configurator.
- Important aspects to consider when porting the project application layer.

To illustrate these aspects, this document references the *SAM D21 IO1 Xplained demo* example available on Atmel START. Prior starting the migration process, ensure that the following prerequisites are met:

- Internet connection with access to https://start.atmel.com
- Latest version of MPLAB X IDE https://www.microchip.com/mplab/mplab-x-ide
- Latest version of MPLAB Harmony 3 https://github.com/Microchip-MPLAB-Harmony
- SAM D21 Xplained Pro board (optional)
- I/O1 Xplained Pro board (optional)

Table of Contents

Introduction....1

1. Creating MPLAB Harmony 3 Project Under MPLAB X IDE....3

2. Identifying the Resources Used in the Atmel START Project....8

3. Porting Driver Configuration from Atmel START....13

4. Porting the System PINMUX Configuration from Atmel START....19

5. Porting the System Clock Configuration from Atmel START....26

6. Porting and Adapting Middleware....35

7. Porting the Application Source Code....38

The Microchip Website....41

Product Change Notification Service....41

Customer Support....41

Microchip Devices Code Protection Feature....41

Legal Notice....41

Trademarks....42

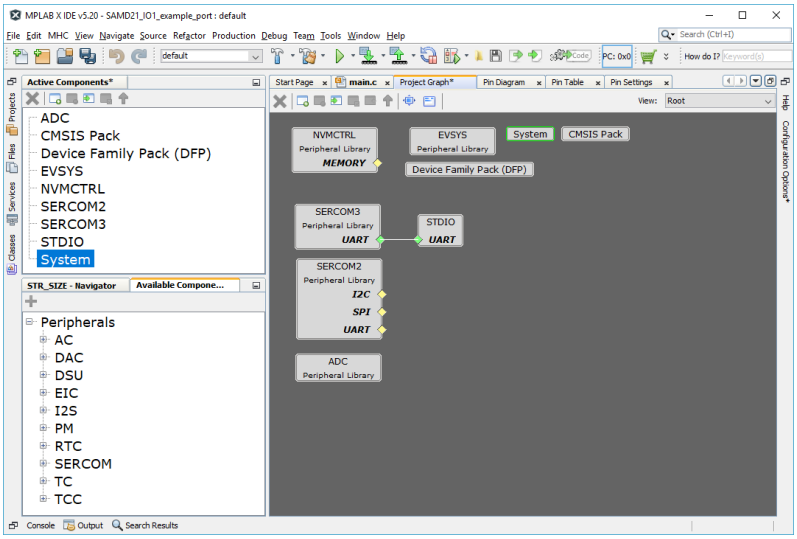
Quality Management System....42

Worldwide Sales and Service....43

1. Creating MPLAB Harmony 3 Project Under MPLAB X IDE

The first step in this process is creating a MPLAB Harmony 3 project under MPLAB X IDE. This new project will enable access to the MPLAB Harmony 3 Configurator (MHC), and access the list of software modules, such as peripheral libraries, services and middleware, which are available for the target SAM device.

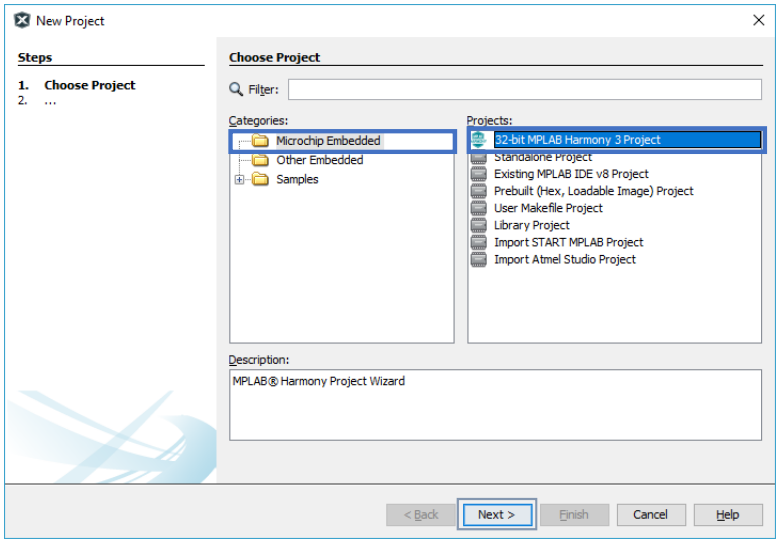
Figure 1-1. MPLAB Harmony 3 Configurator Available Components



To create a project for the SAM D21J18A device, follow these steps:

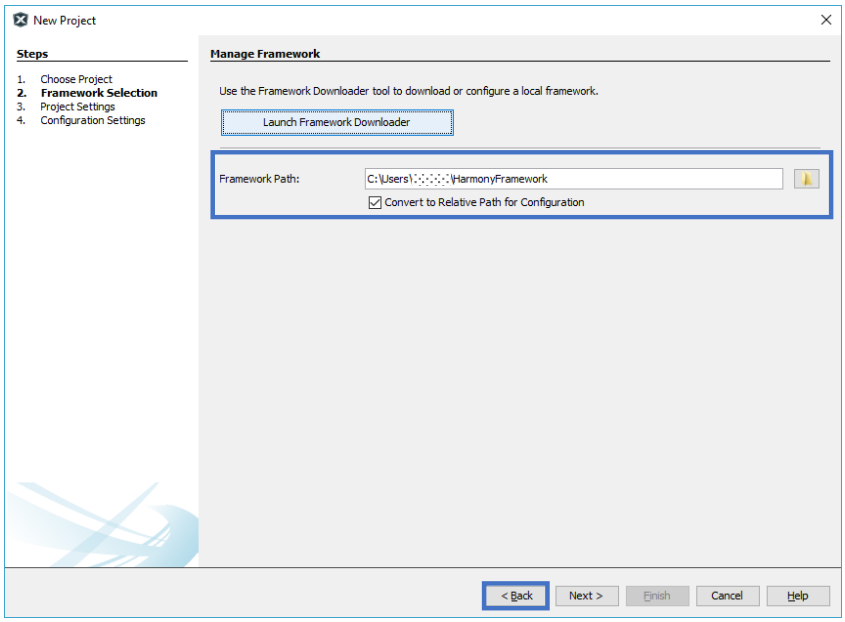
1. Launch MPLAB X IDE.
2. In MPLAB X IDE, select *File* > *New Project* (or click the New Project icon).
3. In the New Project window, under Steps, select Choose Project, and then under Choose Project section, select these options: for Categories select Microchip Embedded, and for Projects select 32-bit MPLAB Harmony 3 Project.
   Note: If the option 32-Bit MPLAB Harmony 3 Project is not available, users need to install the MPLAB Harmony 3 Configurator plug-in by selecting *Tools* > *Plugins* > *Available Plugins* before continuing with this demonstration. MPLAB Harmony 3 Configurator overview is available for download at https://microchipdeveloper.com/harmony3:mhc-overview.

Figure 1-2. Creating an MPLAB Harmony 3-Based New Project



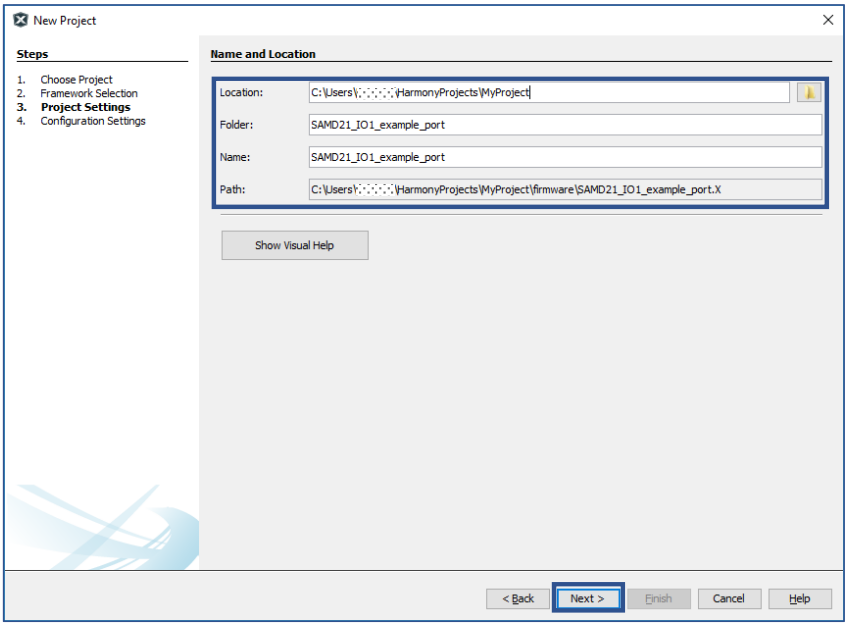
4. Click **Next**.
5. Select Framework Selection, and under Manage Framework section, enter Framework Path (Path to the folder in which the MPLAB Harmony 3 packages are downloaded). For this demonstration, the MPLAB Harmony 3 packages are already downloaded at D:\microchip\github\h3.

Figure 1-3. New Project - Framework Selection



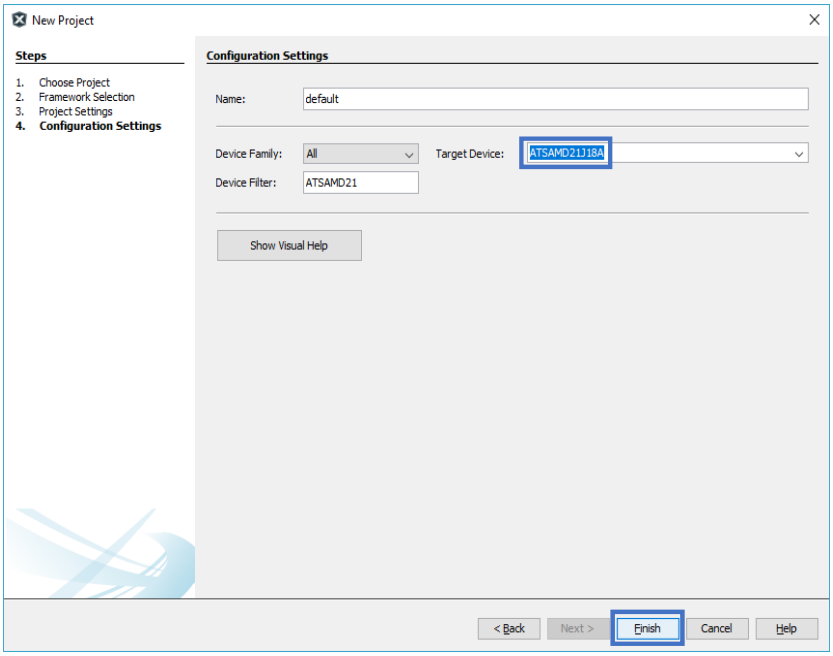
6. Click **Next**.
7. Select Project Settings, and under Name and Location section enter MPLAB Harmony 3 new project details:
   - Location: Indicates the path to the root folder of the new project. All project files will be placed inside this folder.
   - Folder: Indicates the name of the MPLAB X IDE folder.
   - Name: Enter name of the project. This name will be shown in the MPLAB X IDE.
   - Path: It is read-only box. It will update as and when users make changes to other entries.

Figure 1-4. New Project Wizard Step 3



8. Click **Next**.
9. Select Configuration Settings, and then enter details as given below:
   - Name: Enter the configuration name.
   - Target Device: Choose a device name.
10. Click **Finish**.

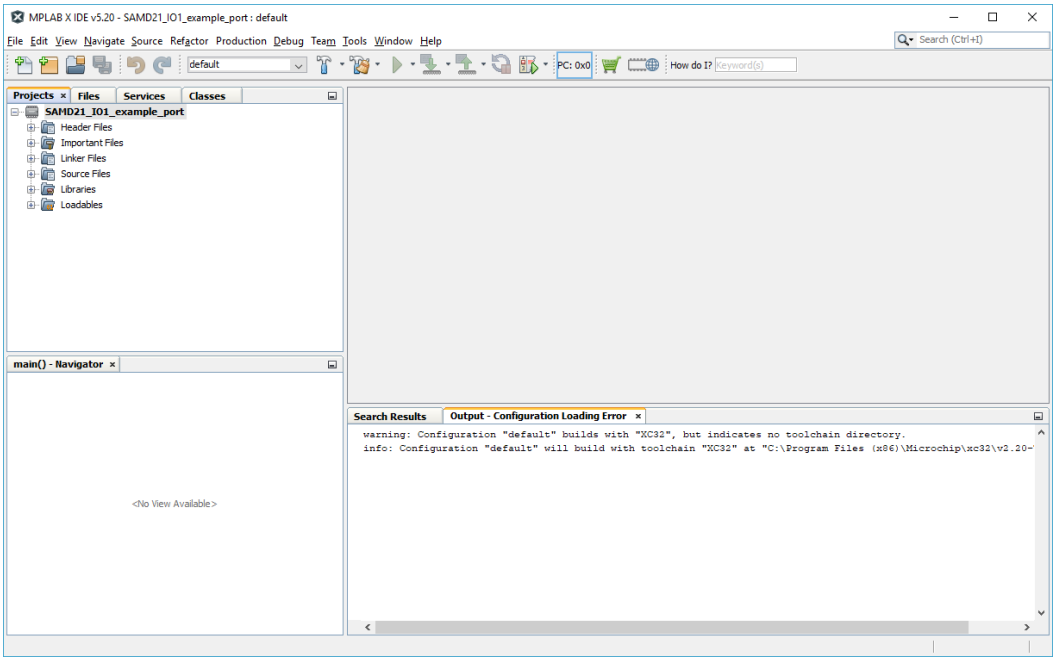
Figure 1-5. New Project Wizard Step 4



The new MPLAB Harmony 3 project will be launched.

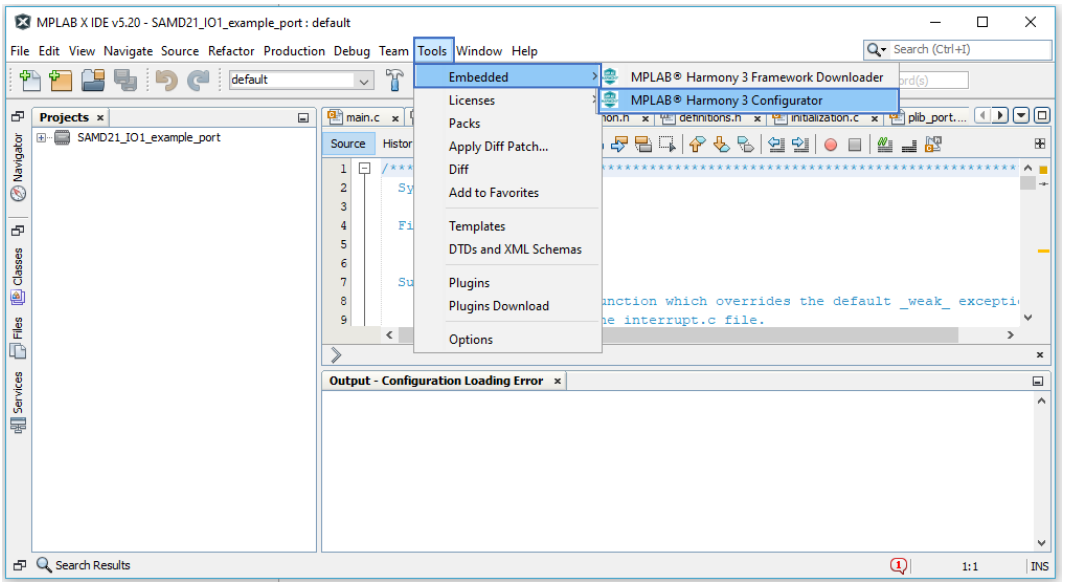
Note: After clicking the Finish button, if MHC does not launch, the user can launch it by selecting *Tools* > *Embedded* > *MPLAB® Harmony 3 Configurator* from the menu bar of the MPLAB X IDE.

Figure 1-6. New MPLAB Harmony 3 Project



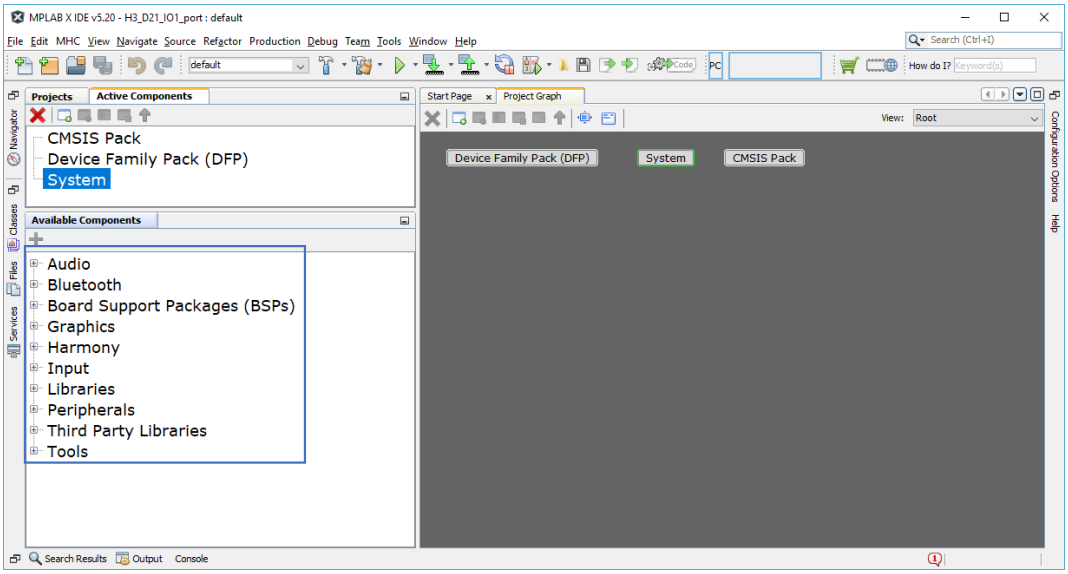
The software modules associated with the project configuration are seen by selecting *Tools* > *Embedded* > *MPLAB Harmony 3 Configurator*.

Figure 1-7. Accessing MPLAB Harmony 3 Configurator



The associated MPLAB Harmony 3 Configurator list will be displayed as shown in the figure below.

Figure 1-8. MPLAB Harmony 3 Configurator - Available Component List



2. Identifying the Resources Used in the Atmel START Project

Before porting the application, users need to identify the software modules that compose the application. As both Atmel START and MPLAB Harmony 3 frameworks provide a graphical configuration interface, hence it is simple to identify the different drivers and middleware modules that are used by the application and compare them to the available MPLAB Harmony 3 modules.

Figure 2-1. Atmel START Graphical Configuration Interface

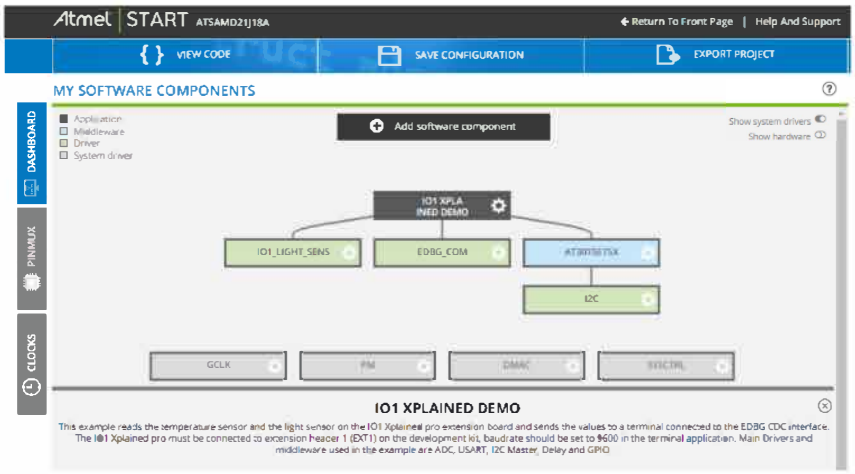
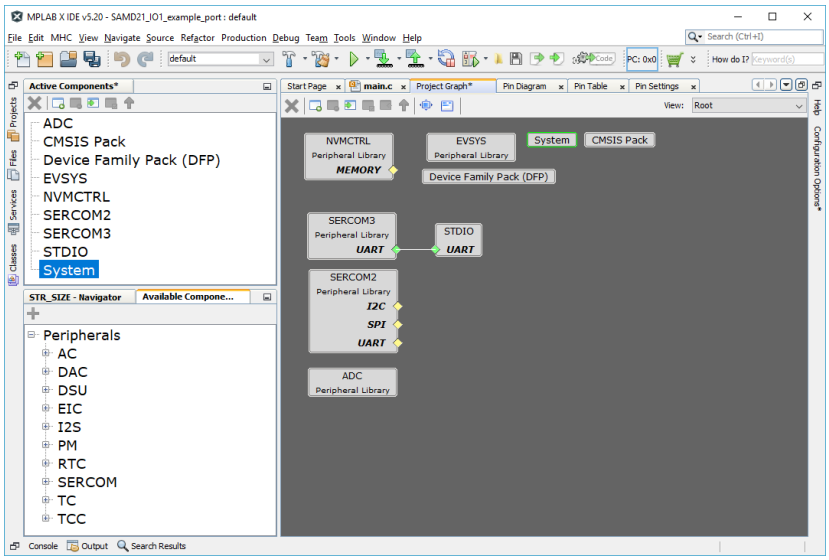


Figure 2-2. Harmony 3 Graphical Configuration Interface (MHC)

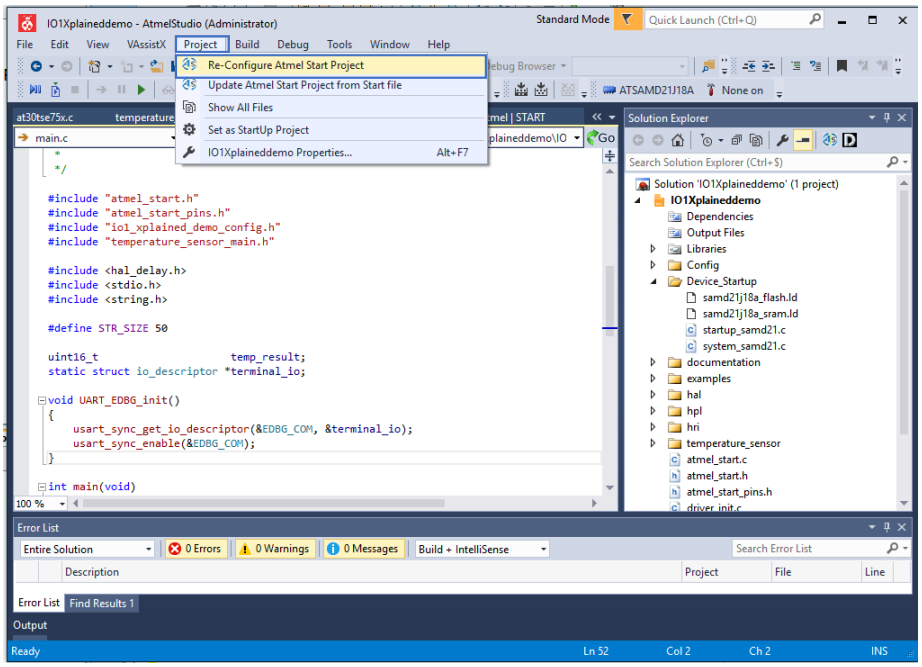


Note: Some applications include additional software functions or middleware that are not part of Atmel START. Therefore, it is important to review the full application flowchart or project architecture to identify them.

Follow these steps while using the SAM D21 I/O1 example from Atmel START as a reference.

1. In Atmel Studio 7, open the Atmel START project to be ported by selecting *Project > Re-Configure Atmel Start Project*.

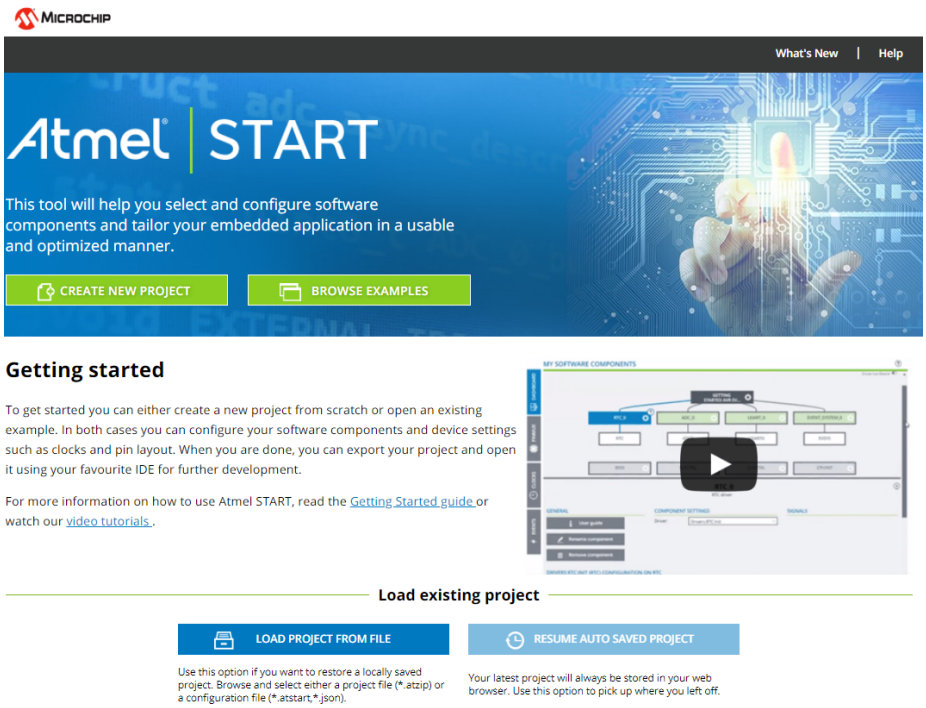
Figure 2-3. Reconfiguring Atmel START Project



Note: The project can also be opened by loading its *.atstart/.atzip* file at https://start.atmel.com/.

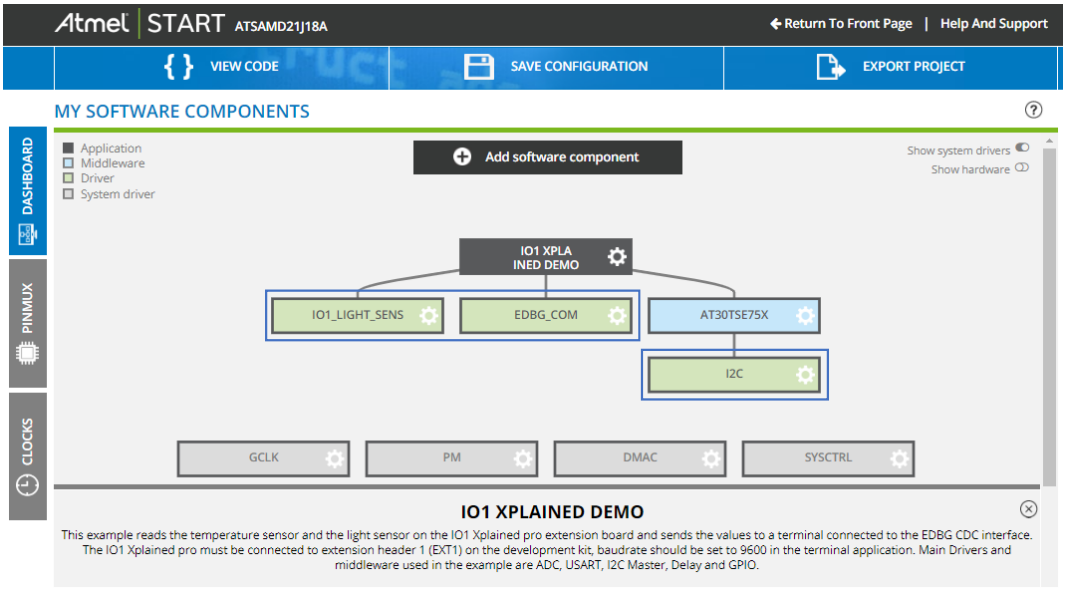
Atmel START main window showing Load Project tab is displayed.

Figure 2-4. Load Project From a File Under Atmel START



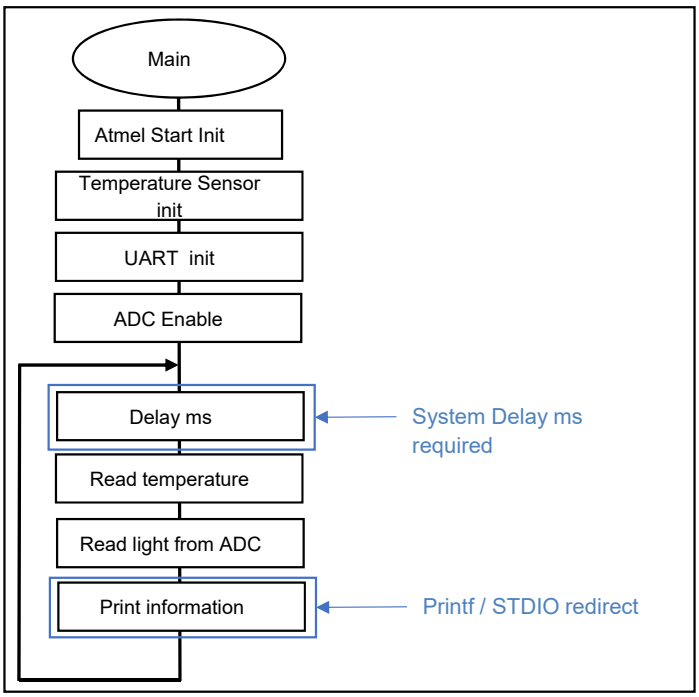
2. When project is loaded, click on the Dashboard tab and identify the Atmel START drivers and its configuration.

Figure 2-5. Driver Identification Under Atmel START



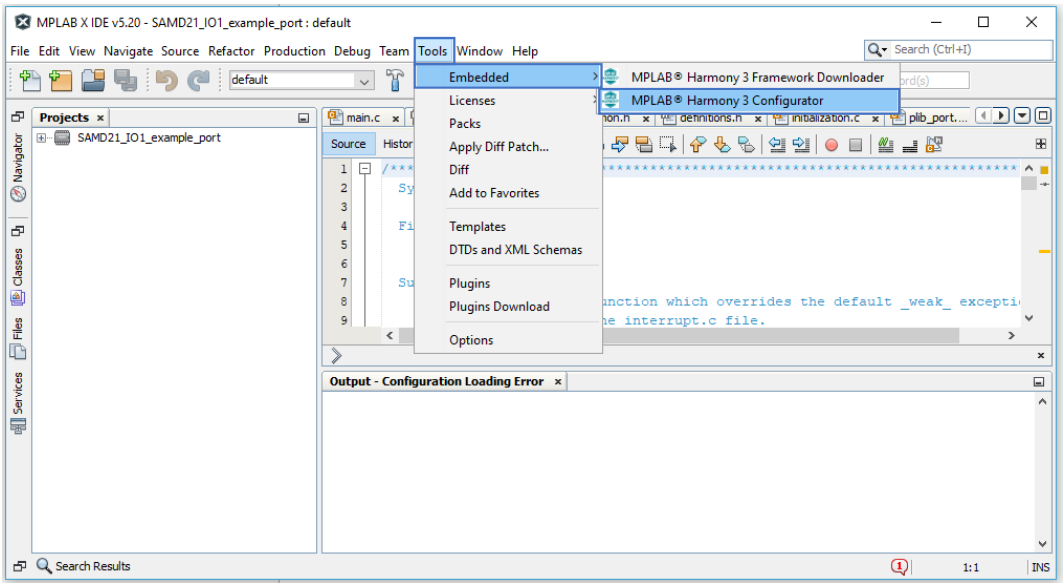
3. Check the application code and application flowchart to identify whether any additional services and middleware are required. The figure below illustrates the SAM D21 I/O1 flowchart.

Figure 2-6. Additional Service Middleware Identification Using a Flowchart



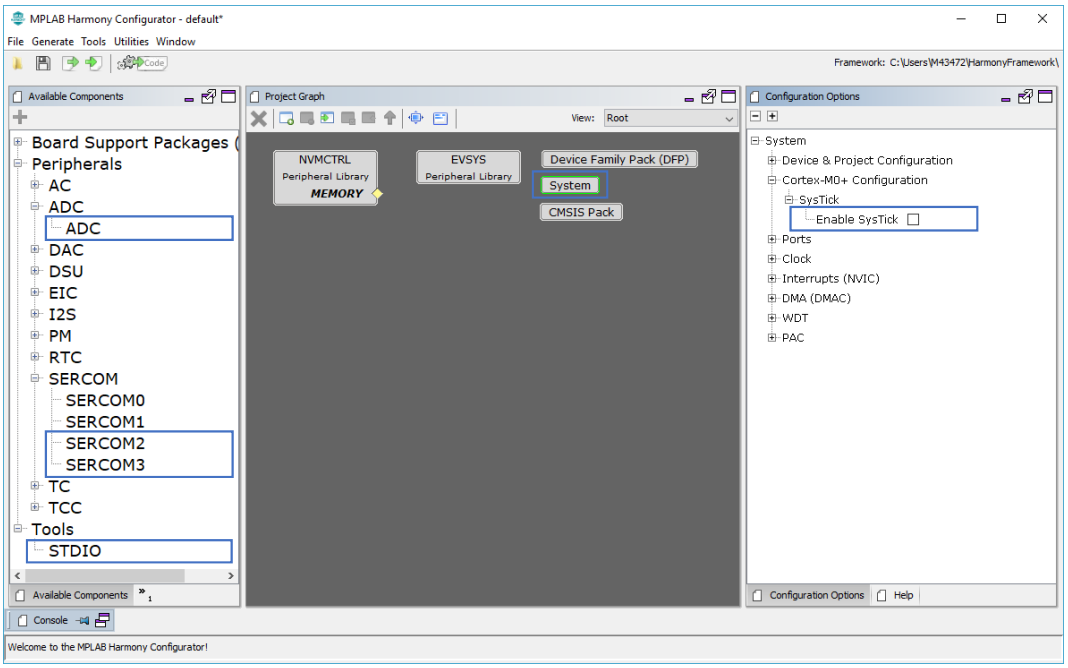
4. If MPLAB Harmony Configurator is not opened, from MPLAB X IDE open the MPLAB Harmony 3 Configurator by selecting *Tools* > *Embedded* > *MPLAB Harmony 3 Configurator*.

Figure 2-7. Accessing MPLAB Harmony 3 Configurator



5. Search for software module availability in Harmony 3 Configurator, and then drag and drop it into the Project Graph window.

Figure 2-8. Identifying Software Module in MHC



The table below provides the drivers and middleware used for this demonstration.

Table 2-1. Drivers and Middleware

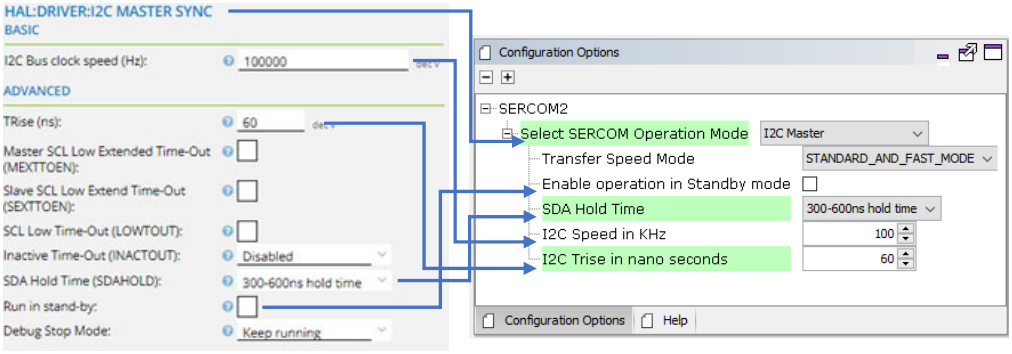
	ATMEL START Module	Harmony 3 Module
Drivers	IO1_LIGHT_SENS	ADC
	I^2C	SERCOM2
	EDBG_COM	SERCOM3
	Systick (implicit)	SYSTEM > SYSTICK
Middleware	AT30SE75X	None
	STDIO Redirect	STDIO

Note: Specific development is required for AT30SE75X. Other drivers, system services, and middleware are available under MPLAB Harmony 3.

3. Porting Driver Configuration from Atmel START

To port a driver from Atmel Studio to MPLAB Harmony 3 is a straightforward operation as it relies on the comparison of Atmel START and MPLAB Harmony Configurator Graphical interfaces.

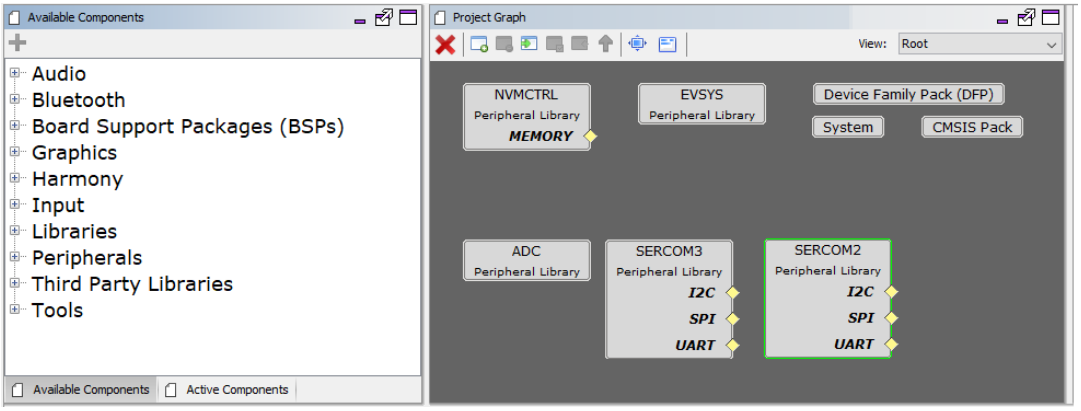
Figure 3-1. Atmel START to MPLAB X Driver Support



Follow these steps for the SAM D21 I/O1 Xplained demonstration:

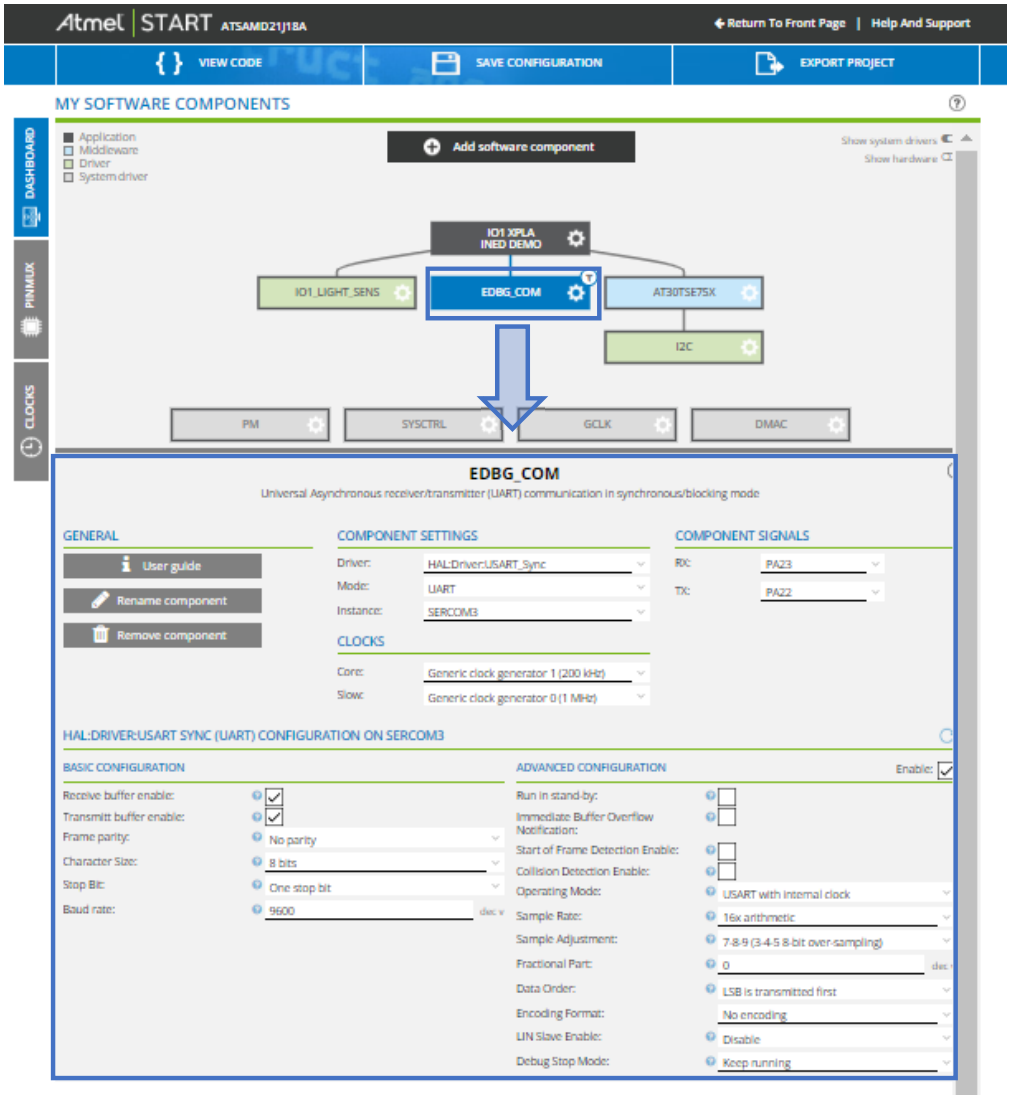
1. Add peripheral library (drivers) to the MPLAB Harmony 3 project by dragging them from the Available Component list to the Project Graph section.

   Figure 3-2. Add Peripheral Driver from the Available Component List



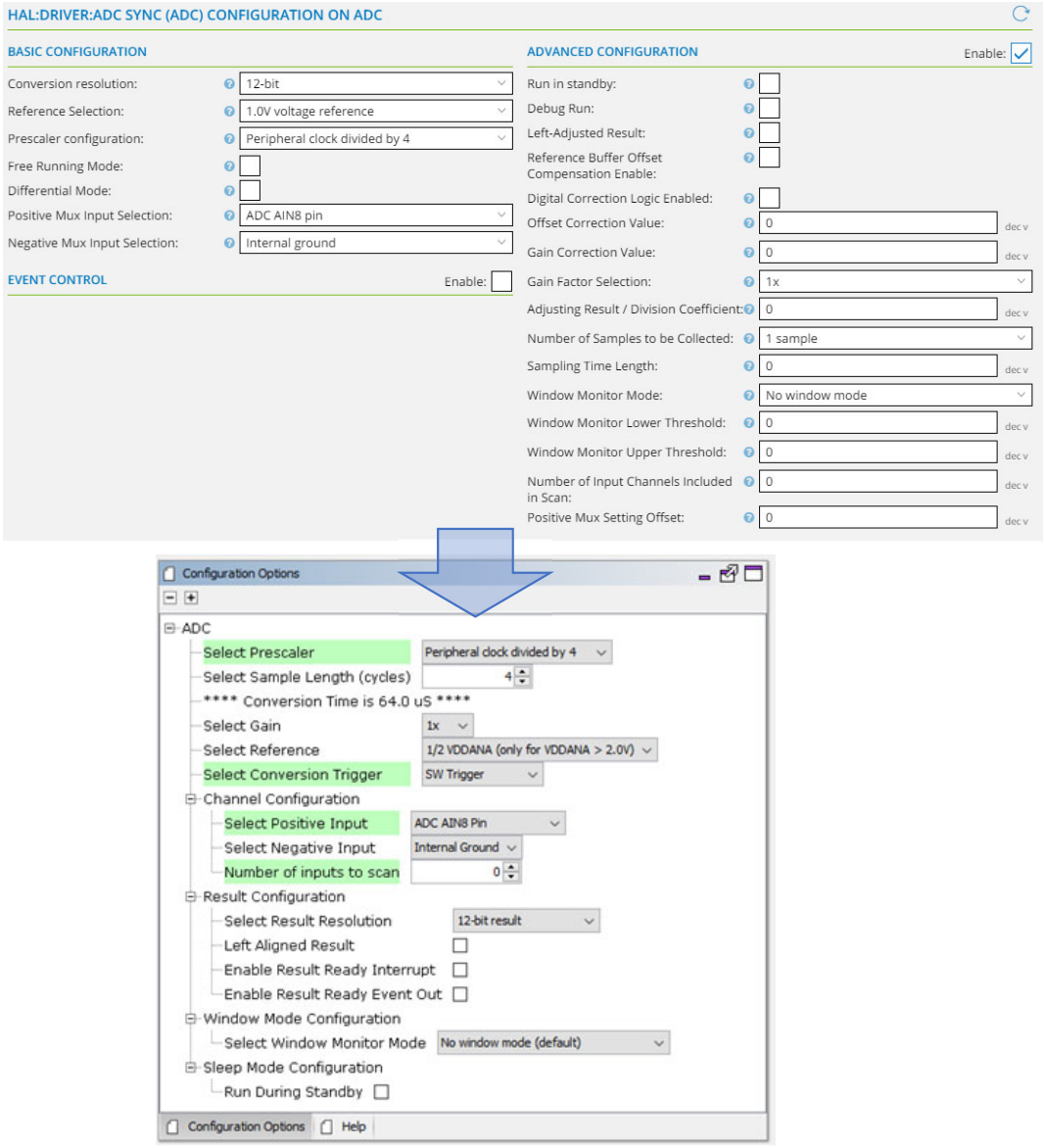
2. Configure all the MPLAB Harmony 3 peripheral libraries according to their equivalent driver in Atmel START.
   Note: Specific Atmel START driver configurations can be seen by clicking individually on each of the driver component in the application dashboard interface.

Figure 3-3. Access Driver Configuration in Atmel START



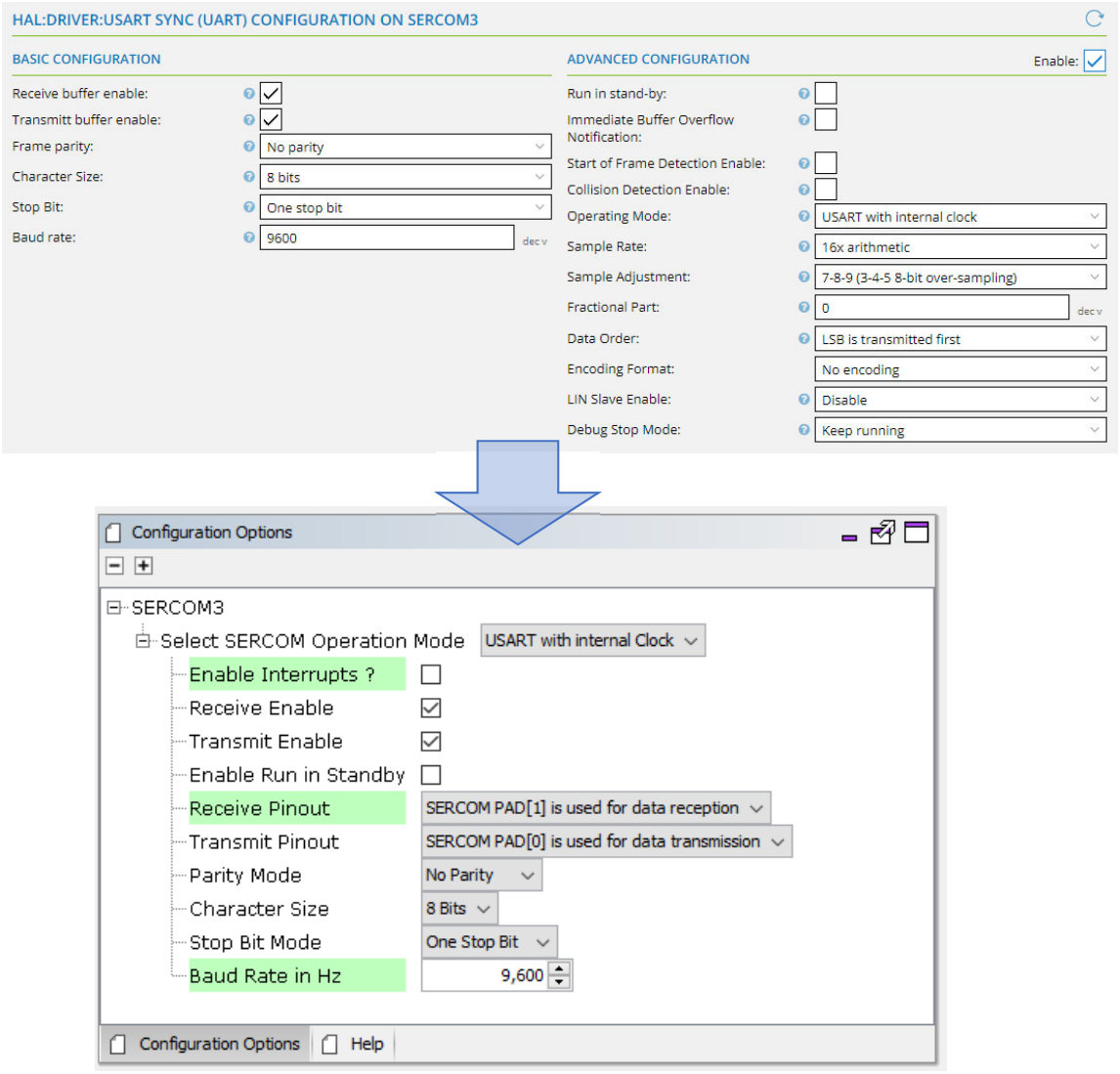
– ADC peripheral Library (IO1_LIGHT_SENS) configuration.

Figure 3-4. Port ADC Driver Configuration



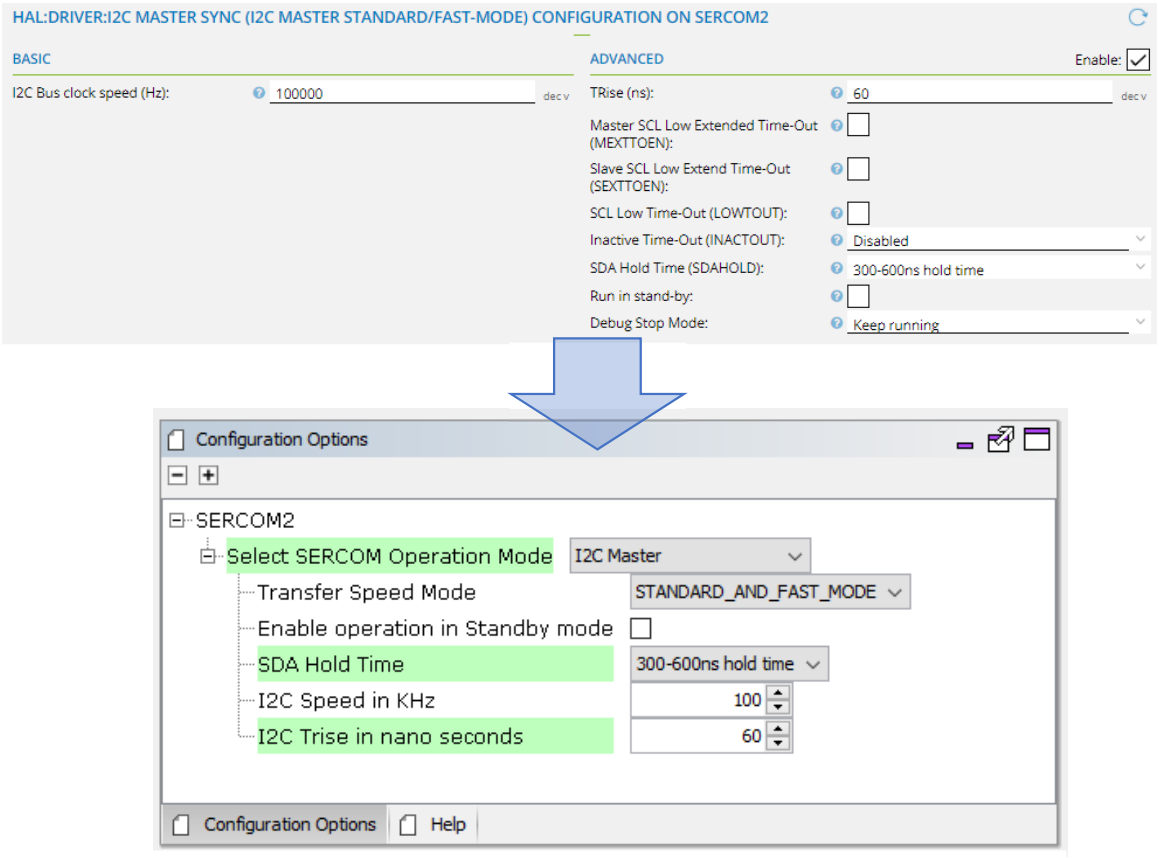
– SERCOM3 peripheral Library (EDBG_COM) configuration

Figure 3-5. Port SERCOM3 Driver Configuration



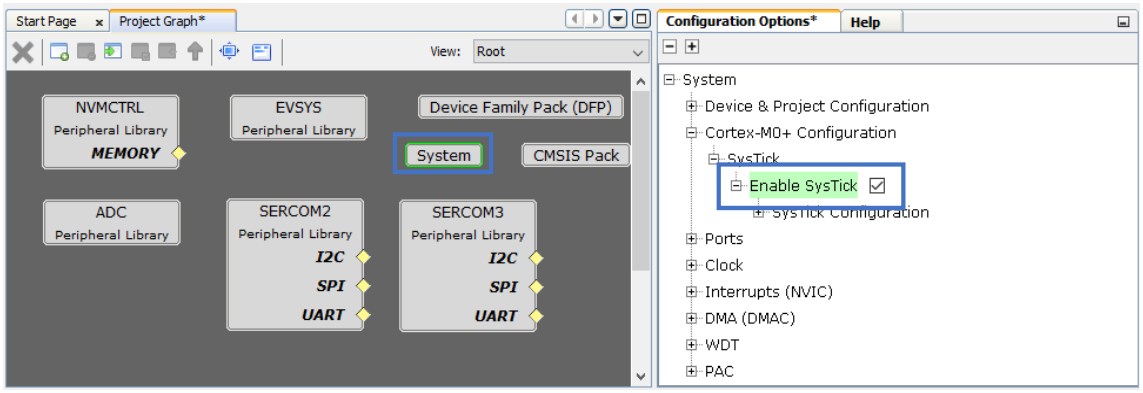
– SERCOM2 peripheral Library (I^2C) configuration

Figure 3-6. Port SERCOM2 Driver Configuration



– SYSTICK Service configuration

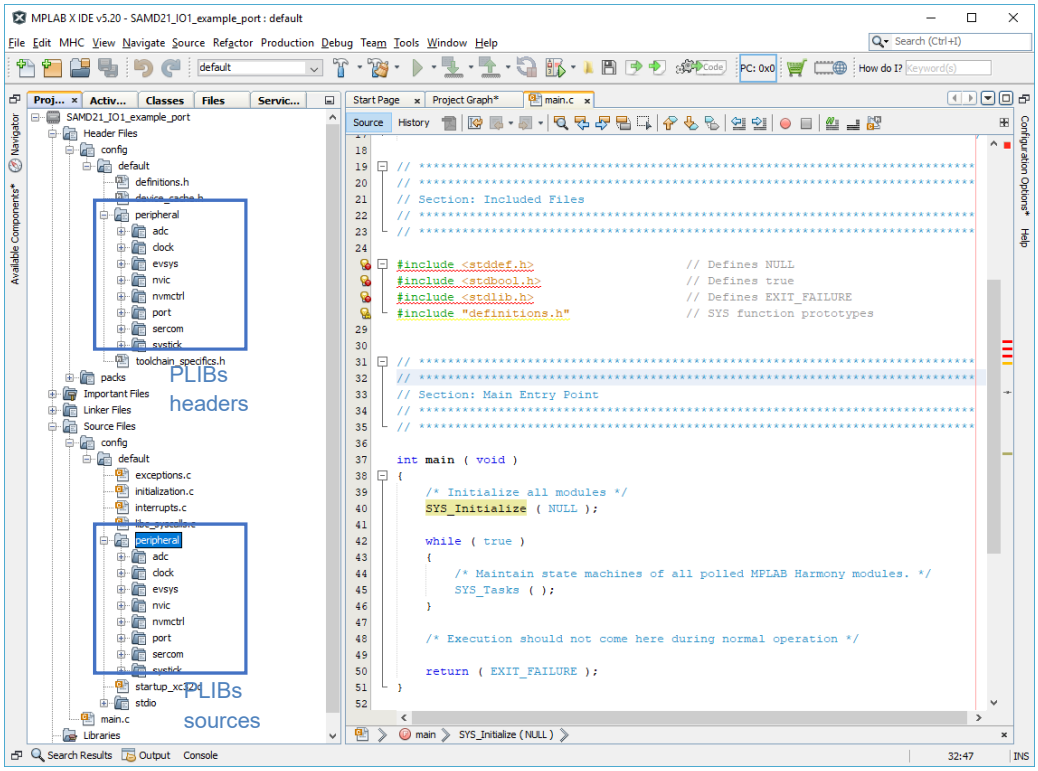
Figure 3-7. Enable Service Under MHC



After generating the project source code from the MHC interface by clicking 

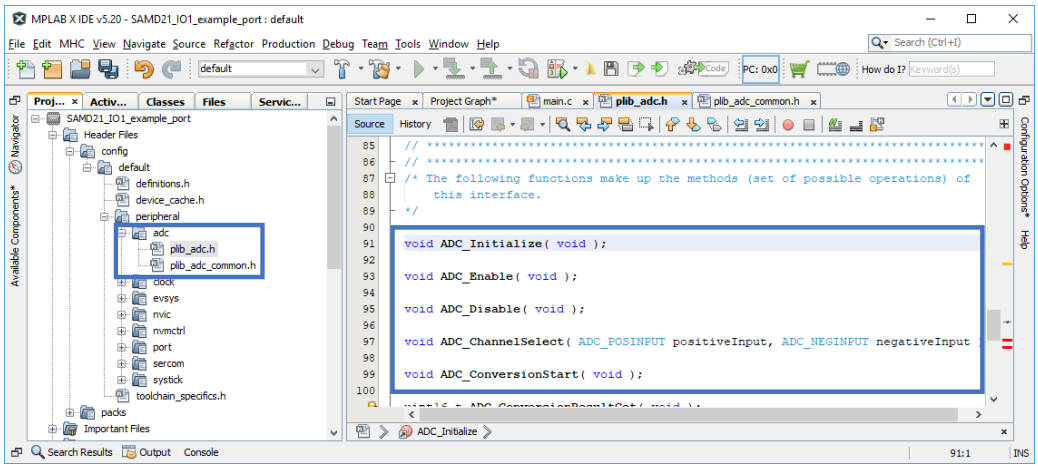
or selecting *MHC* > *Generate Code*, the peripheral libraries (PLIBs) configuration can be found in the following project directories:

Figure 3-8. Generated MHC Project Architecture



The API function definition used by the application can be found in the peripheral library header files.

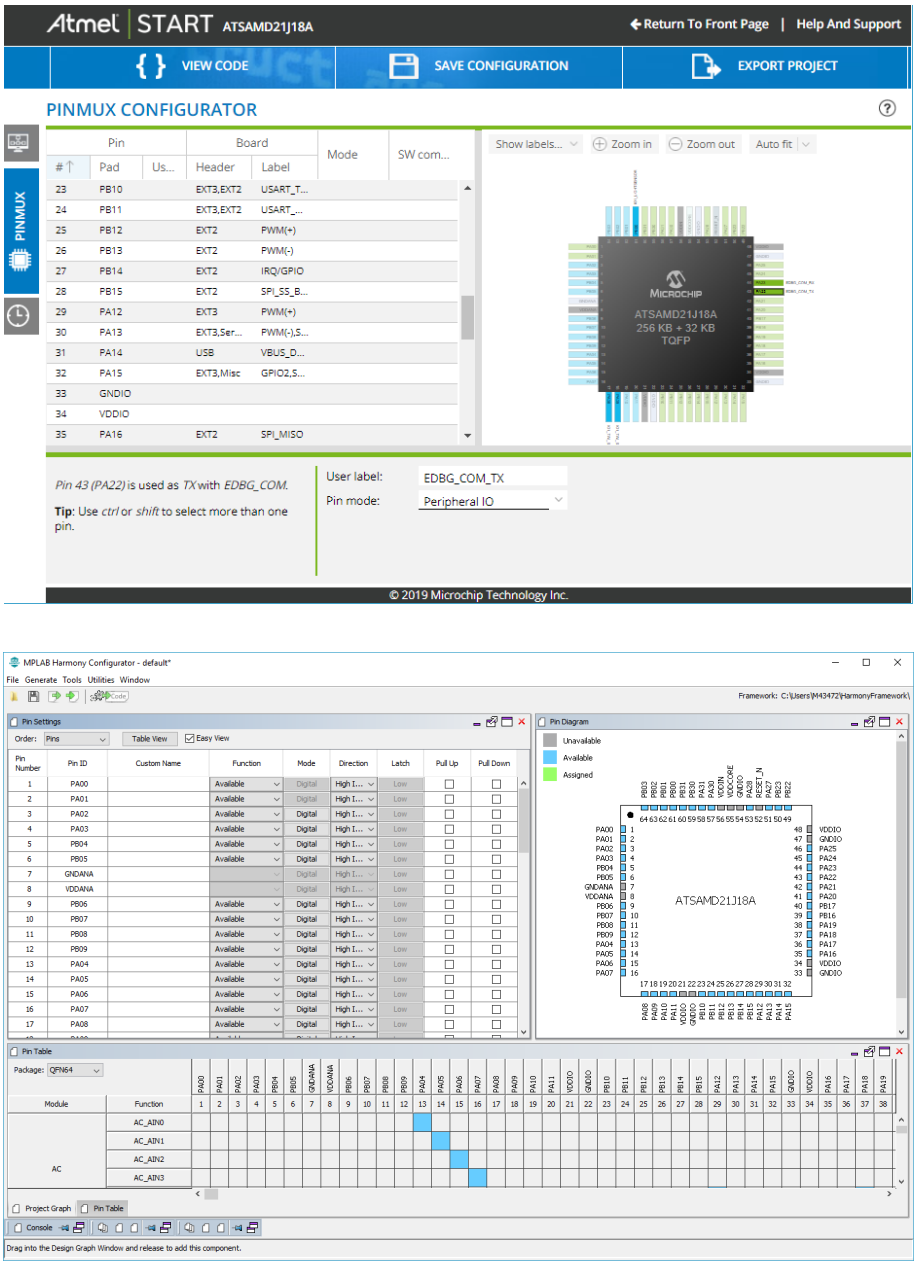
Figure 3-9. Peripheral Libraries API Location



4. Porting the System PINMUX Configuration from Atmel START

Porting an application PINMUX configuration from Atmel Studio to MPLAB Harmony 3 is a straightforward operation as it relies on comparison as each framework graphical interface offers similar capabilities.

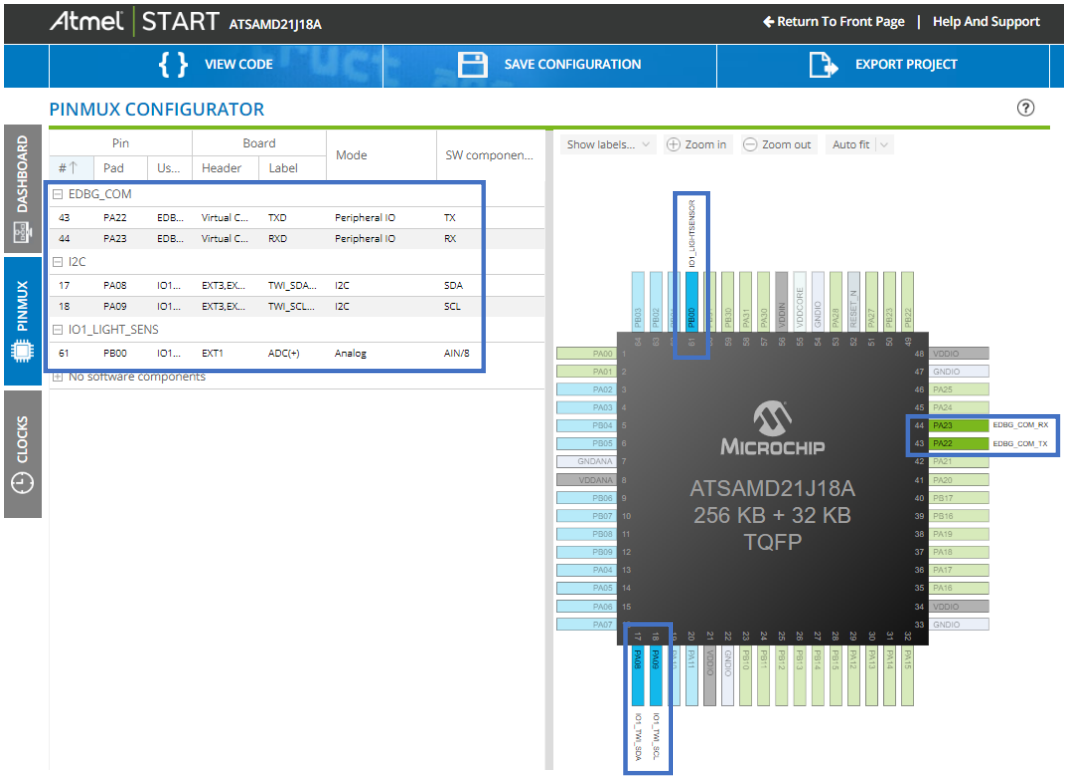
Figure 4-1. Atmel START and MHC Pin Configuration Interface Comparison



Follow these steps for the SAM D21 IO1 Xplained demonstration:

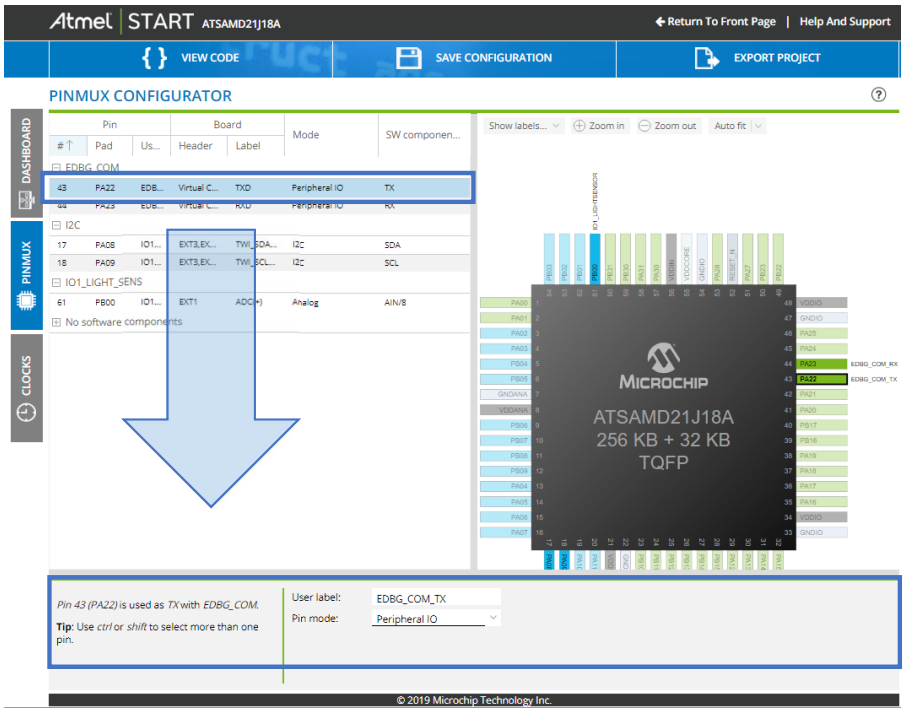
1. In Atmel START, click on the **PINMUX** tab and identify which I/O pins are configured and used in the application.

Figure 4-2. Identify I/O Pins Configuration Under Atmel START



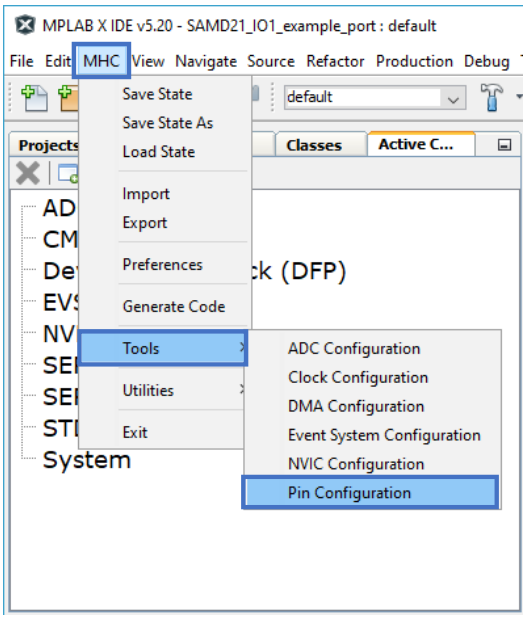
Note: The configuration of each pin is available by clicking on it in the PINMUX configurator table.

Figure 4-3. Identify I/O Pins Configuration Under Atmel START (Continued)



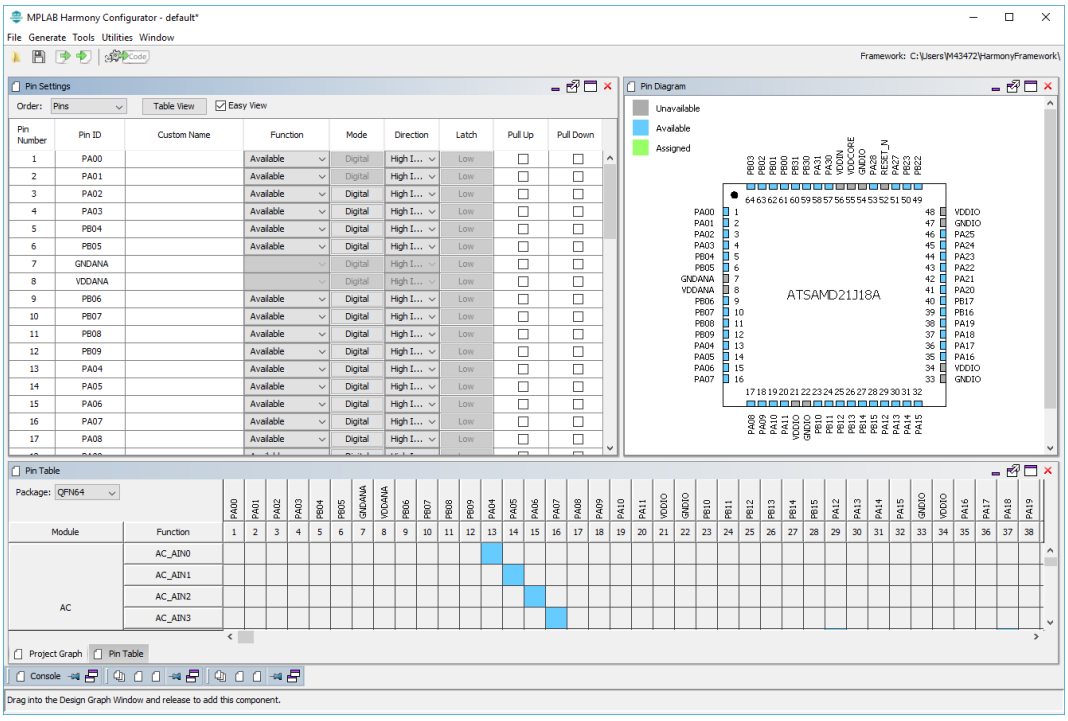
2. In MPLAB X IDE, select *MHC* > *Tools* > *Pin Configuration* to open the project pin configurator interface.

Figure 4-4. Accessing MHC Under MPLAB X



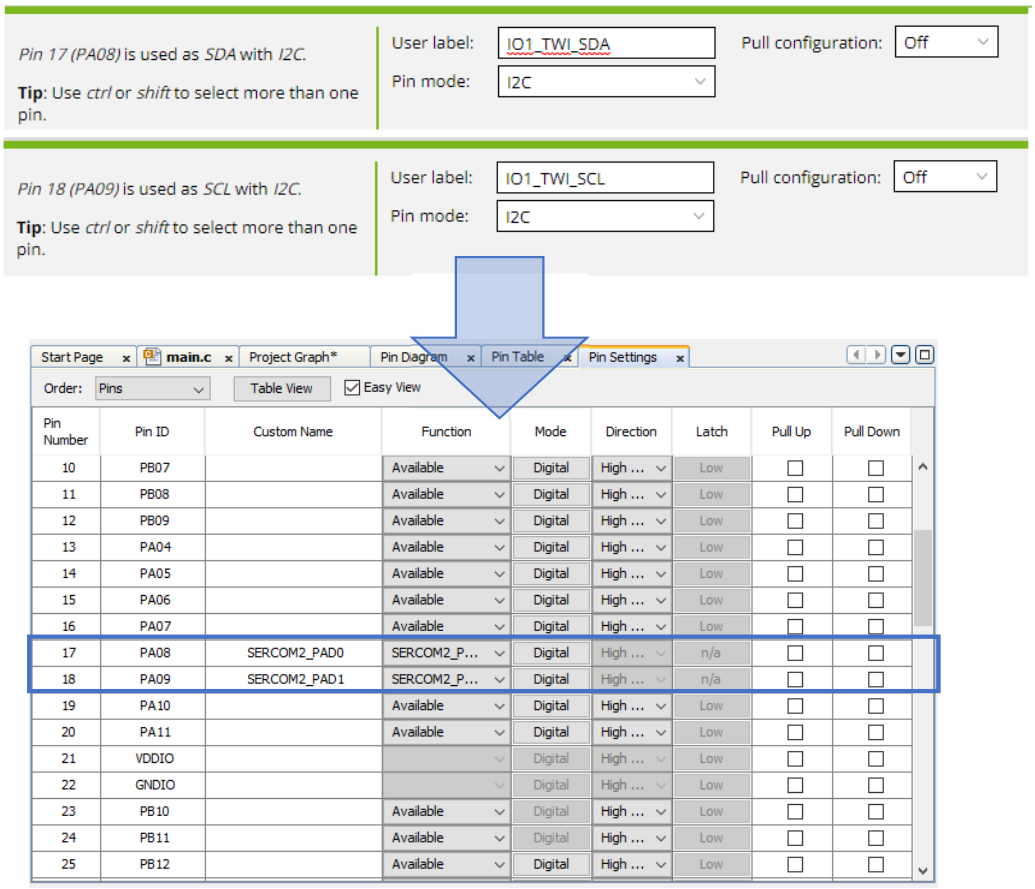
Note: The following tab in the MHC interface enables the customization of the application pin configuration.

Figure 4-5. Accessing Pin Configuration Windows Under the MHC



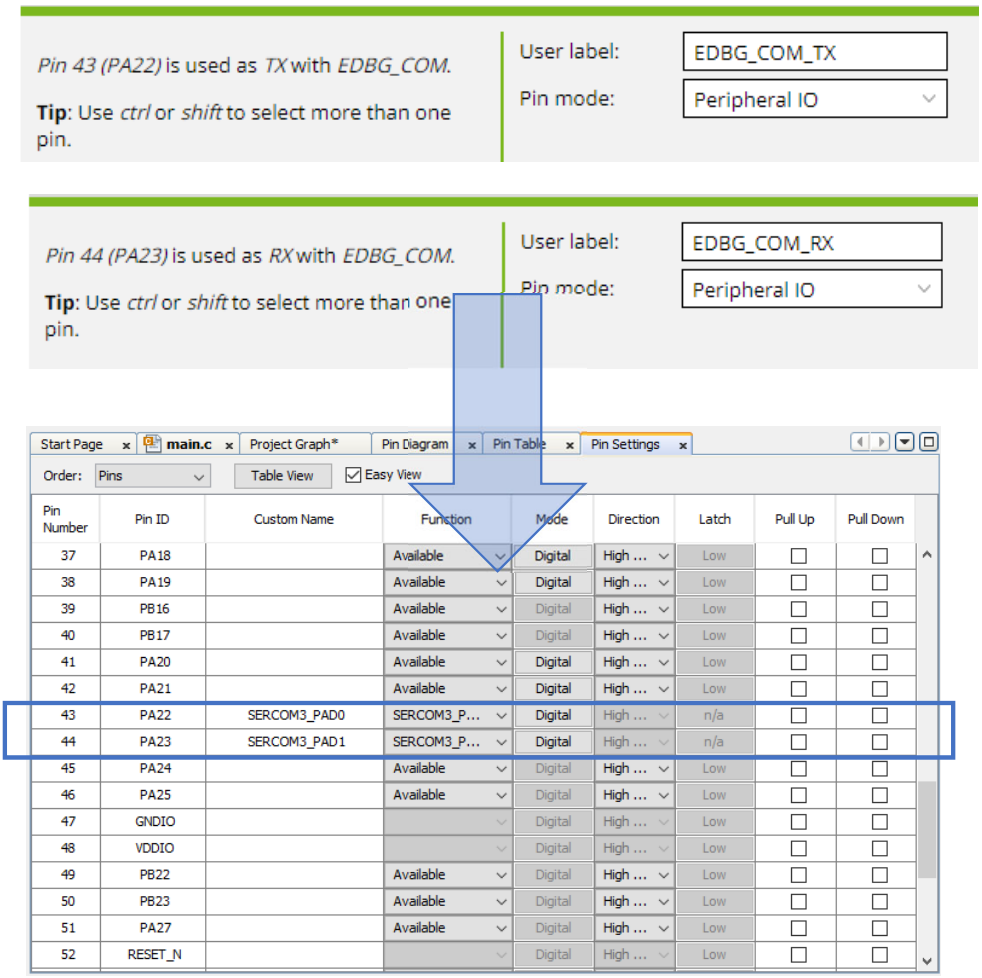
3. Port the I/O pin configuration from Atmel Studio to the MHC Pin Setting tab.
   – SERCOM 2 I/O settings

Figure 4-6. SERCOM 2 I/O Settings



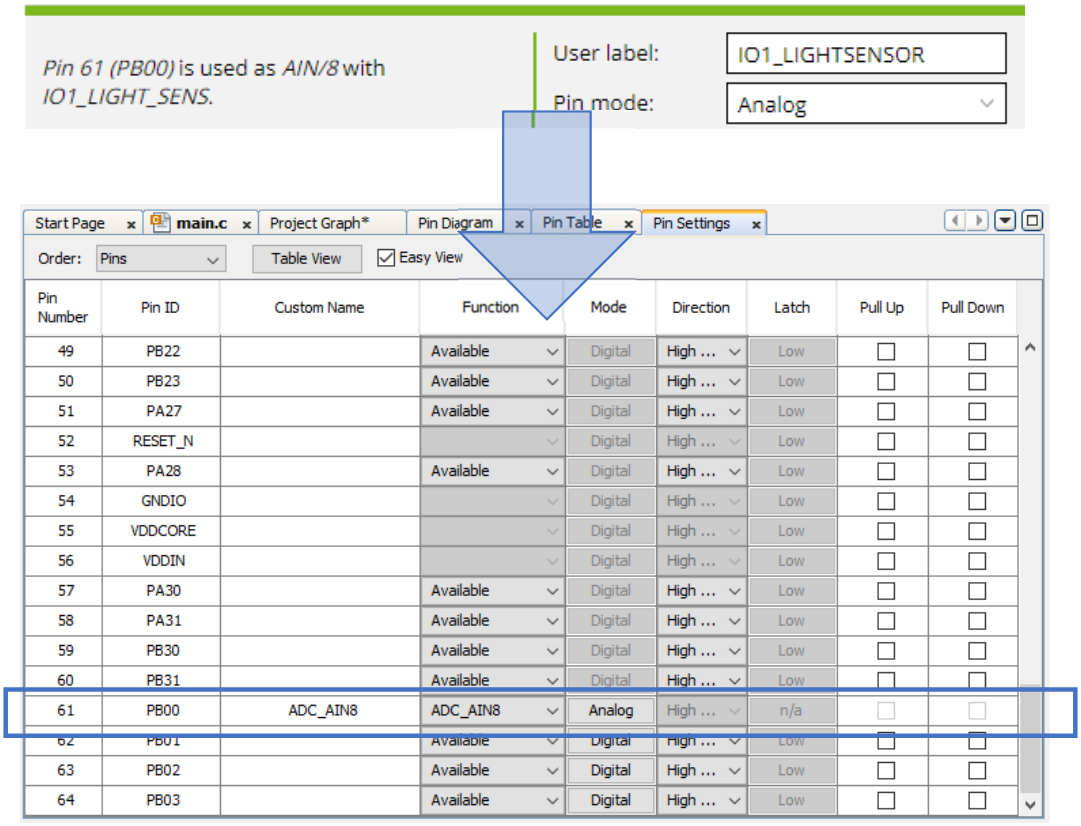
– SERCOM 3 I/Os settings

Figure 4-7. SERCOM 3 I/O settings



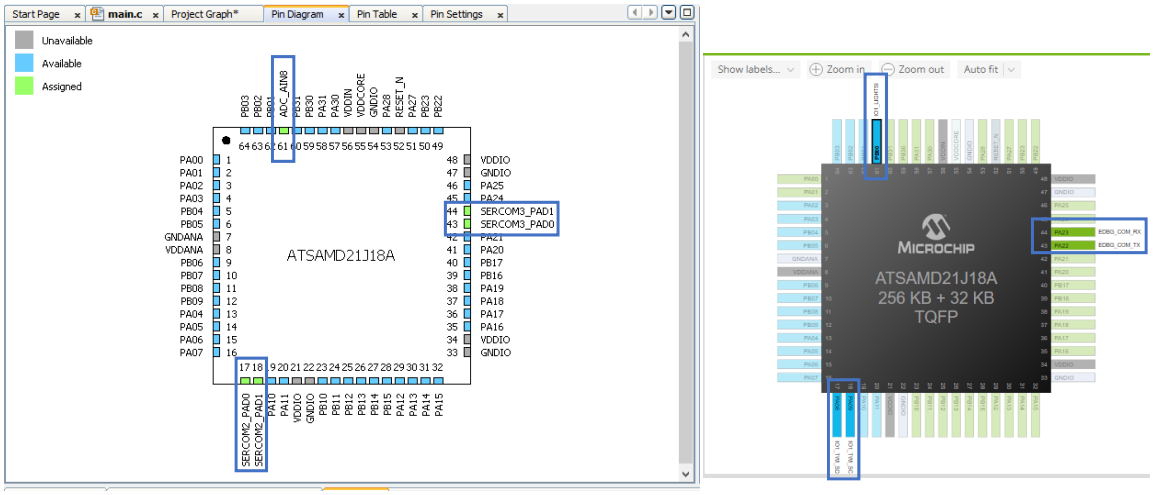
– ADC I/Os settings

Figure 4-8. ADC I/O Settings



4. Verify that the MHC pin diagram is aligned with the initial Atmel START project pin diagram.

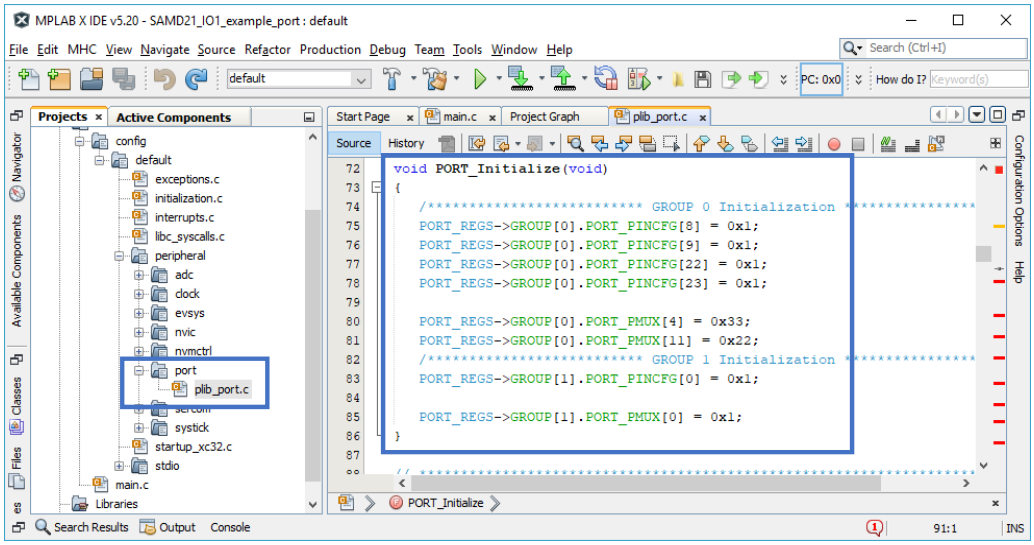
Figure 4-9. Pin Configuration Verification



After regenerating the project source from the MHC interface by clicking 

, the port initialization routine executed during program initialization can be found in the following project files. This initialization routine is automatically generated by the MHC according to the user configuration entered in the graphical interface.

Figure 4-10. MHC Generated Pin Configuration

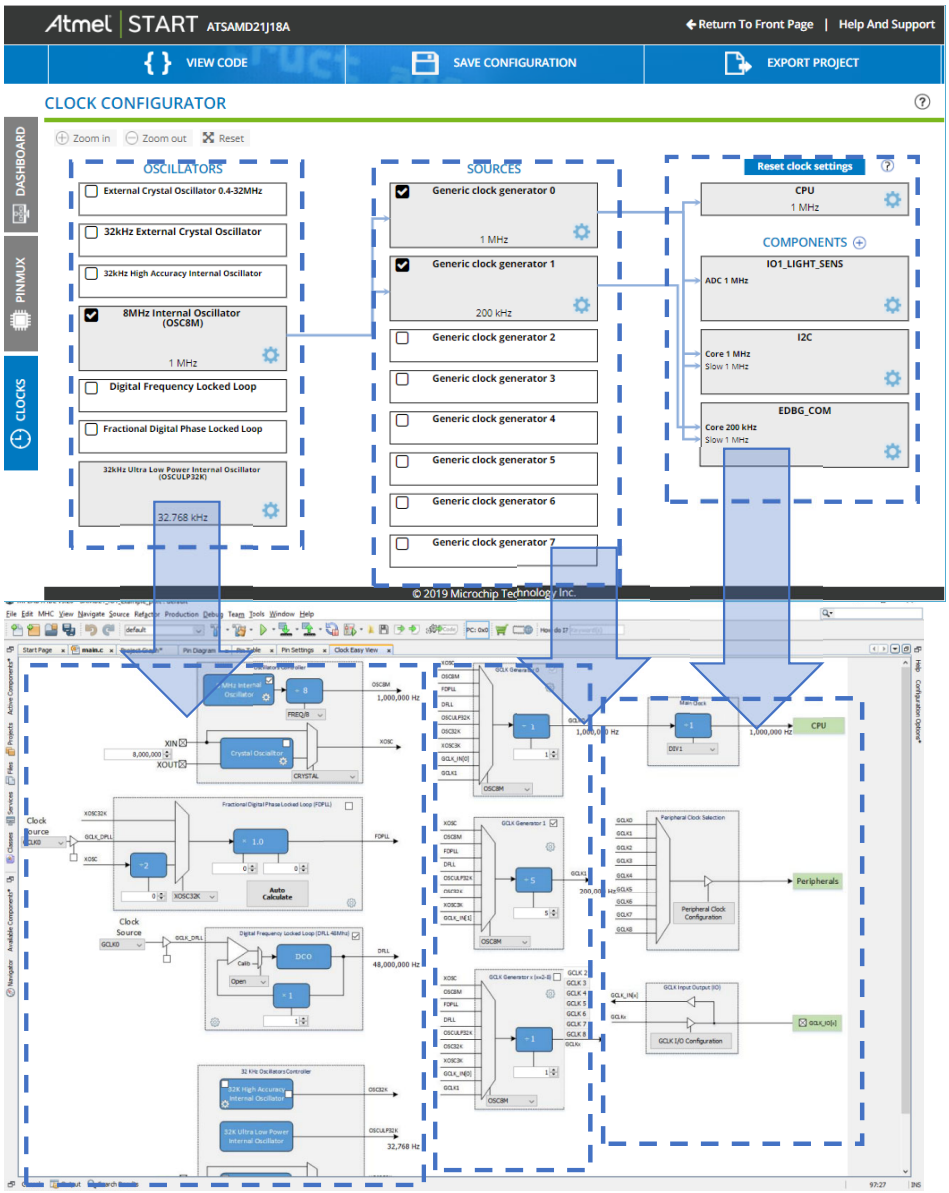


Note: During application run-time, the PORT configuration can be changed by calling APIs from the MPLAB Harmony 3 PORT PLIB.

5. Porting the System Clock Configuration from Atmel START

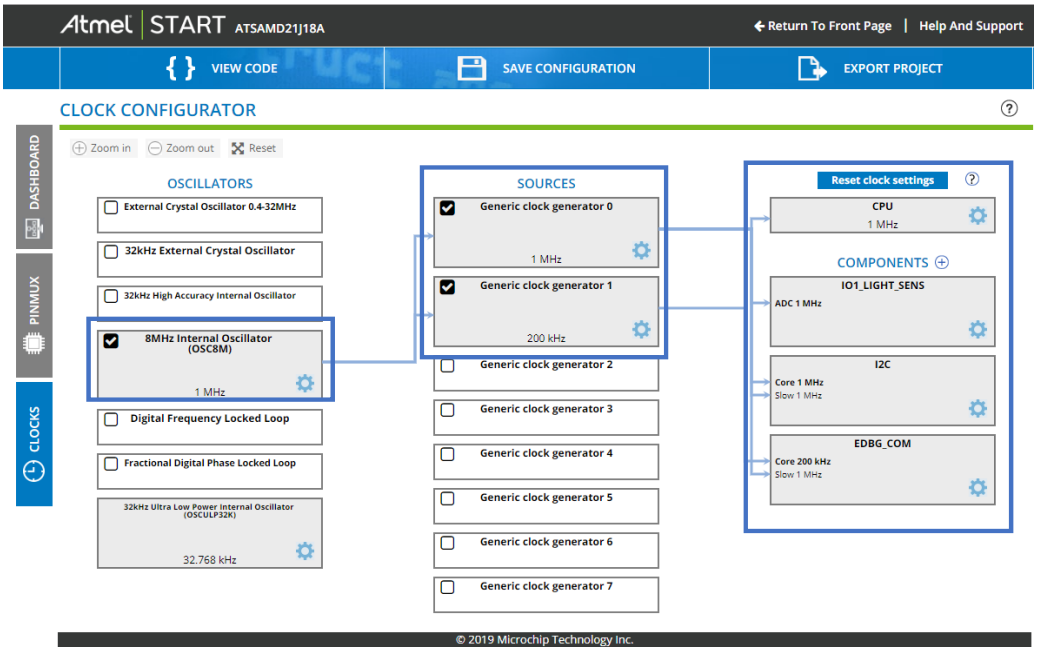
The porting of an application clock configuration from Atmel Studio to MPLAB Harmony 3 is a straightforward operation as it relies on the comparison of both framework graphical interfaces which offer similar capabilities.

Figure 5-1. Atmel START and MHC Clock Configuration Interface Comparison



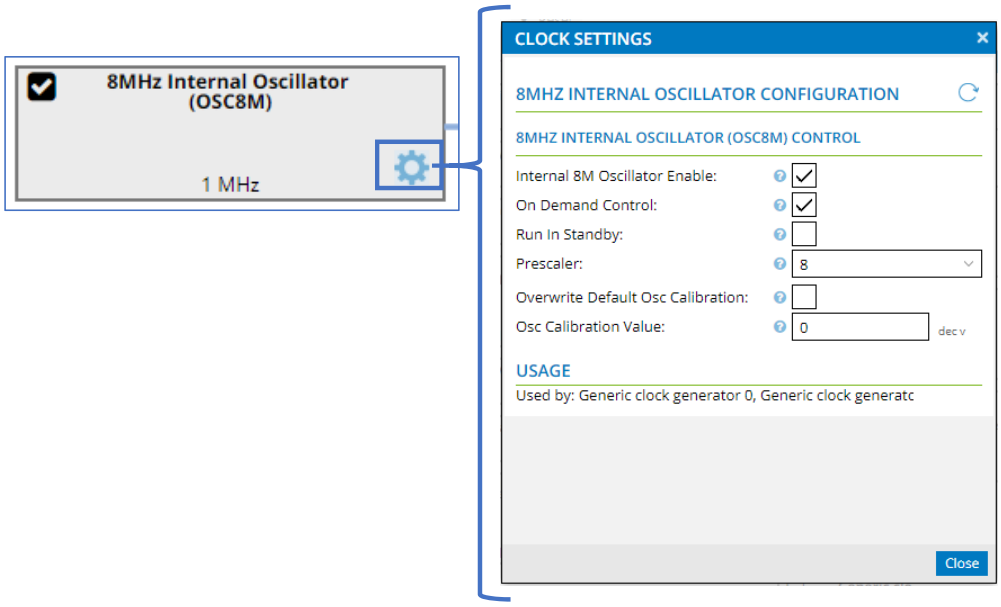
1. In Atmel START, click the **PINMUX** tab and identify different clock elements used in the application and their settings.

Figure 5-2. Identify Project Clock Configuration Under Atmel START



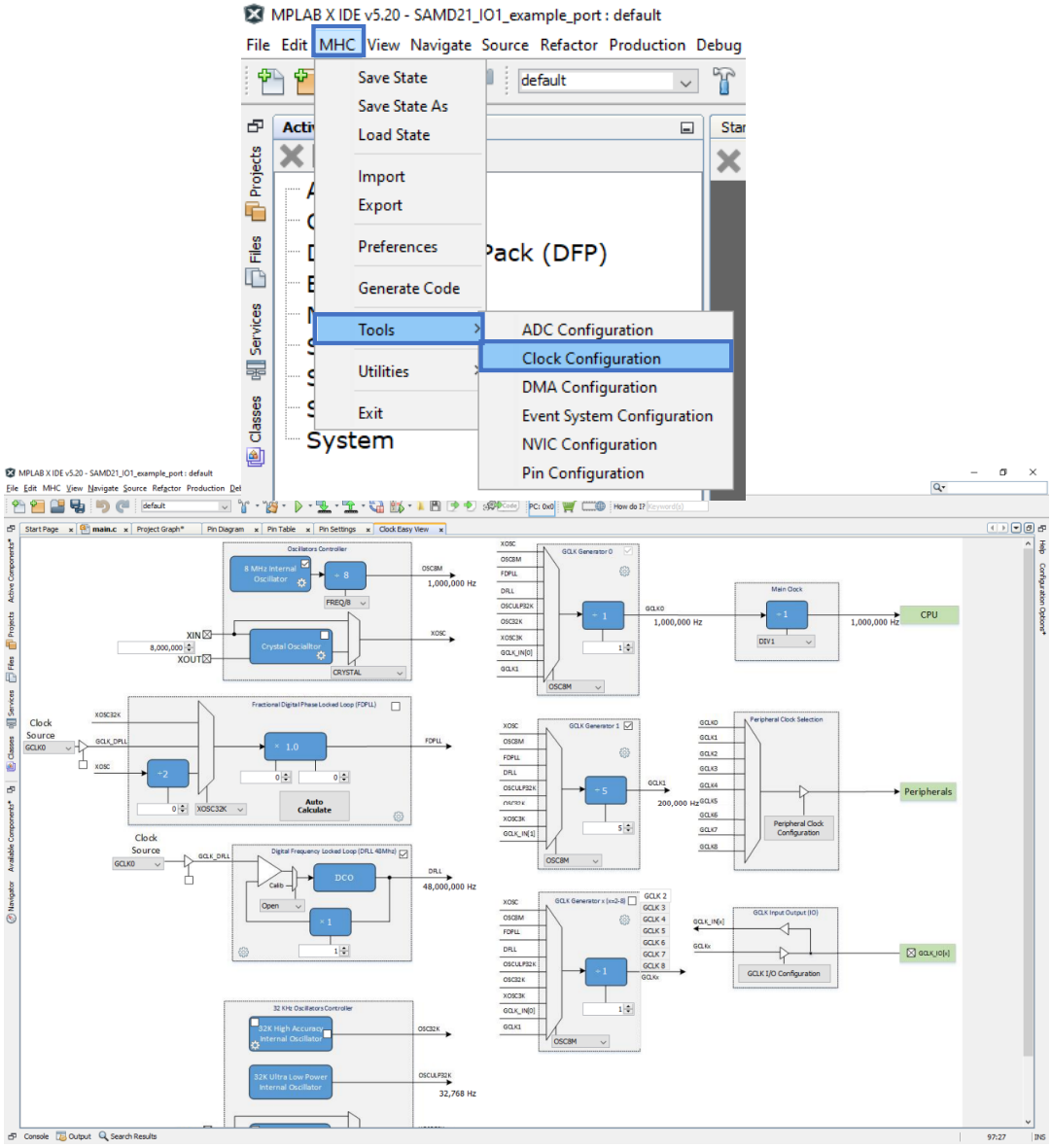
Note: Additional information on the settings for a specific clock element can be retrieved by clicking on the associated blue gear wheel button.

Figure 5-3. Access Clock Element Settings Under Atmel START



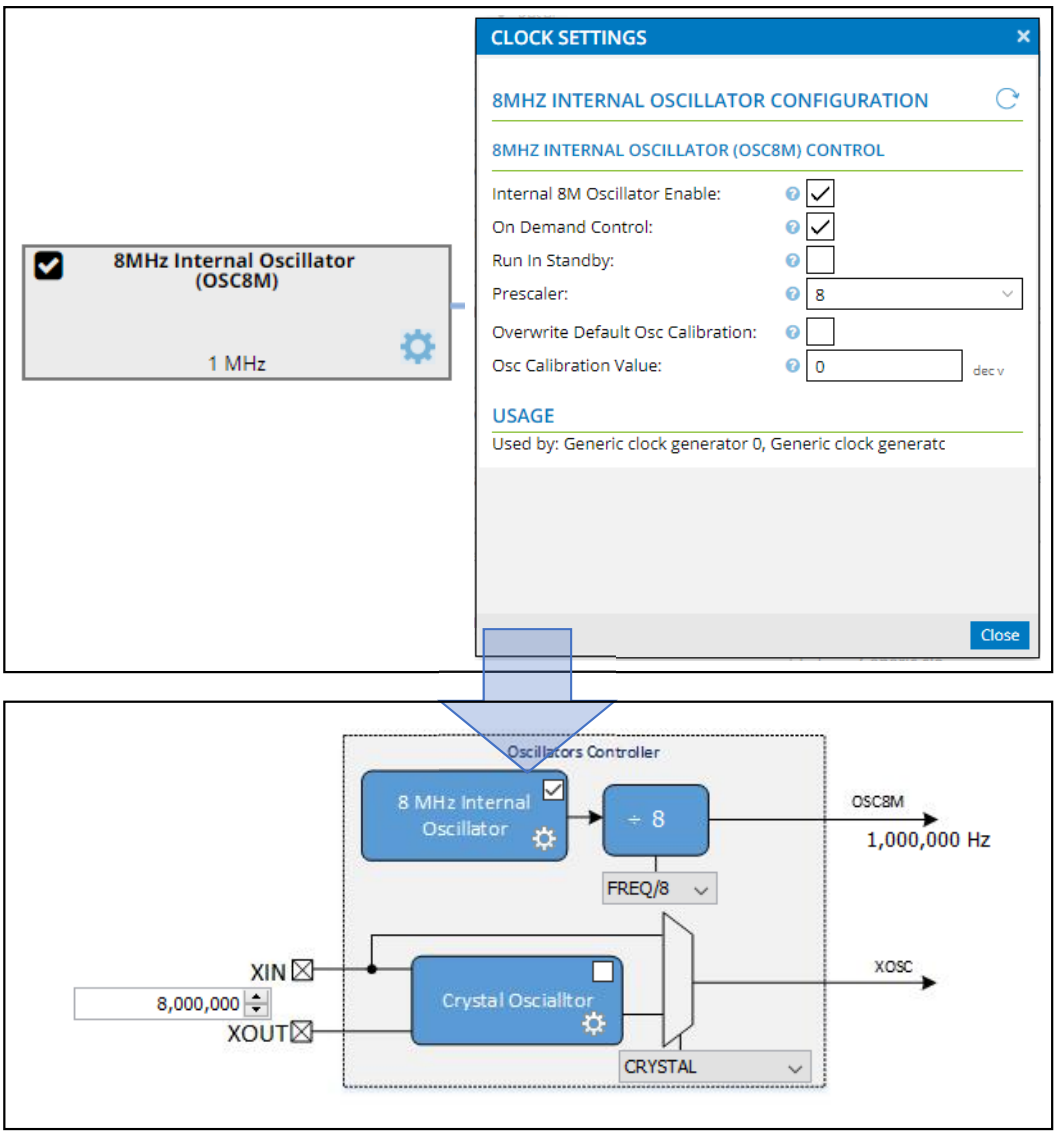
2. To open the project pin configurator interface, in MPLAB X IDE, select *MHC* > *Tools* > *Clock Configuration.*

Figure 5-4. Access Clock Configuration Interface Under MHC



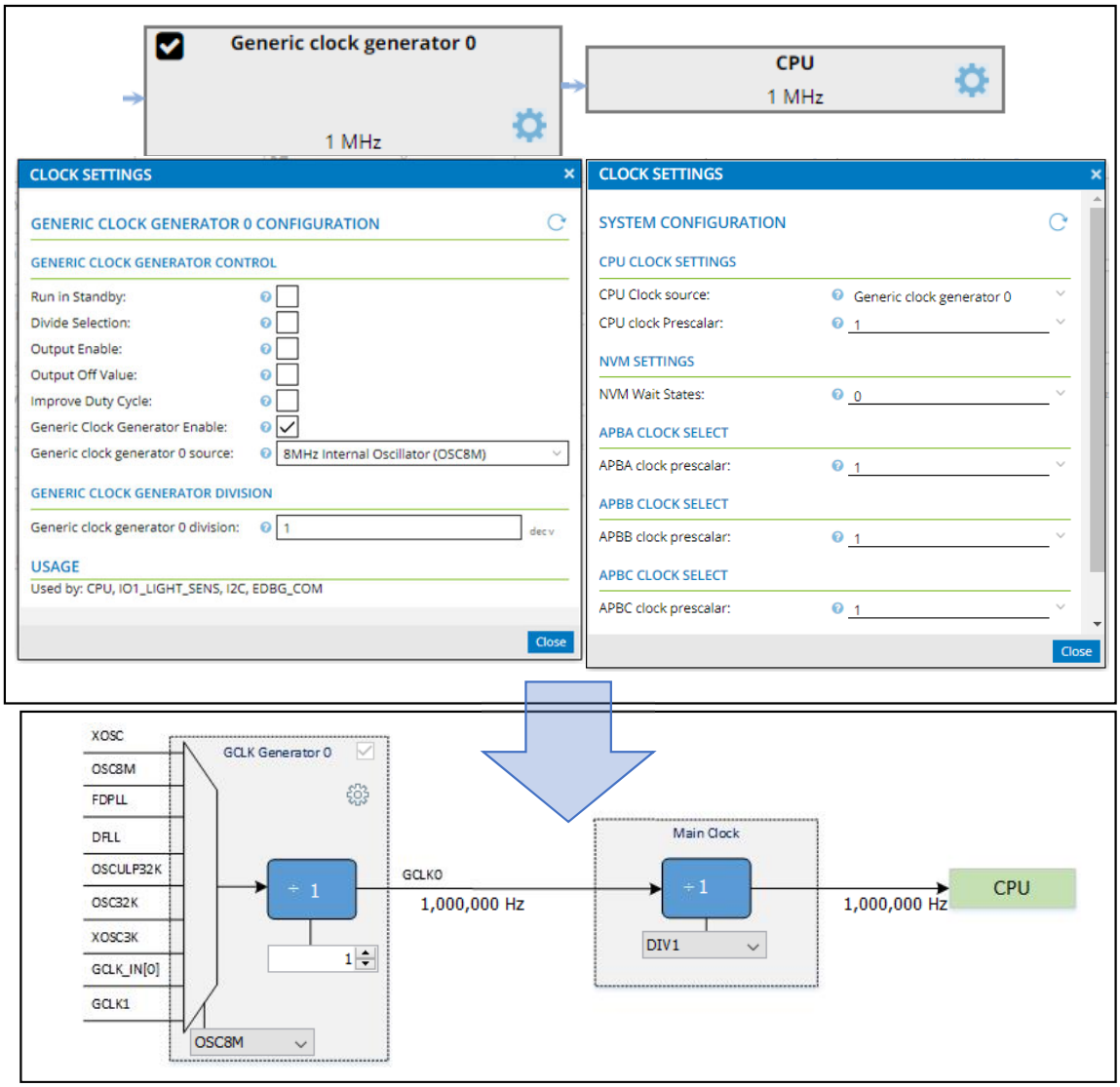
3. Port each clock component configuration from Atmel START to the MHC Clock view.
   – 8 MHz Internal Oscillator (OSC8M)

Figure 5-5. 8MHz Internal Oscillator Configuration Port



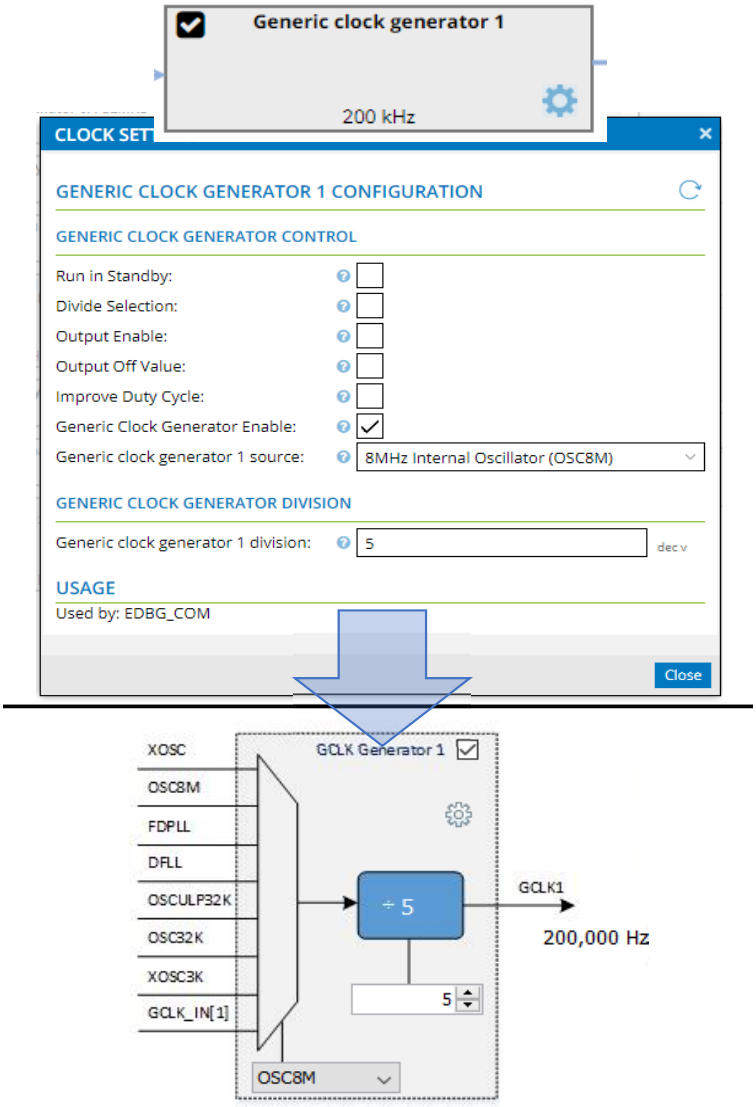
– GCLK Generator 0

Figure 5-6. GCLK Generator 0 Configuration Port



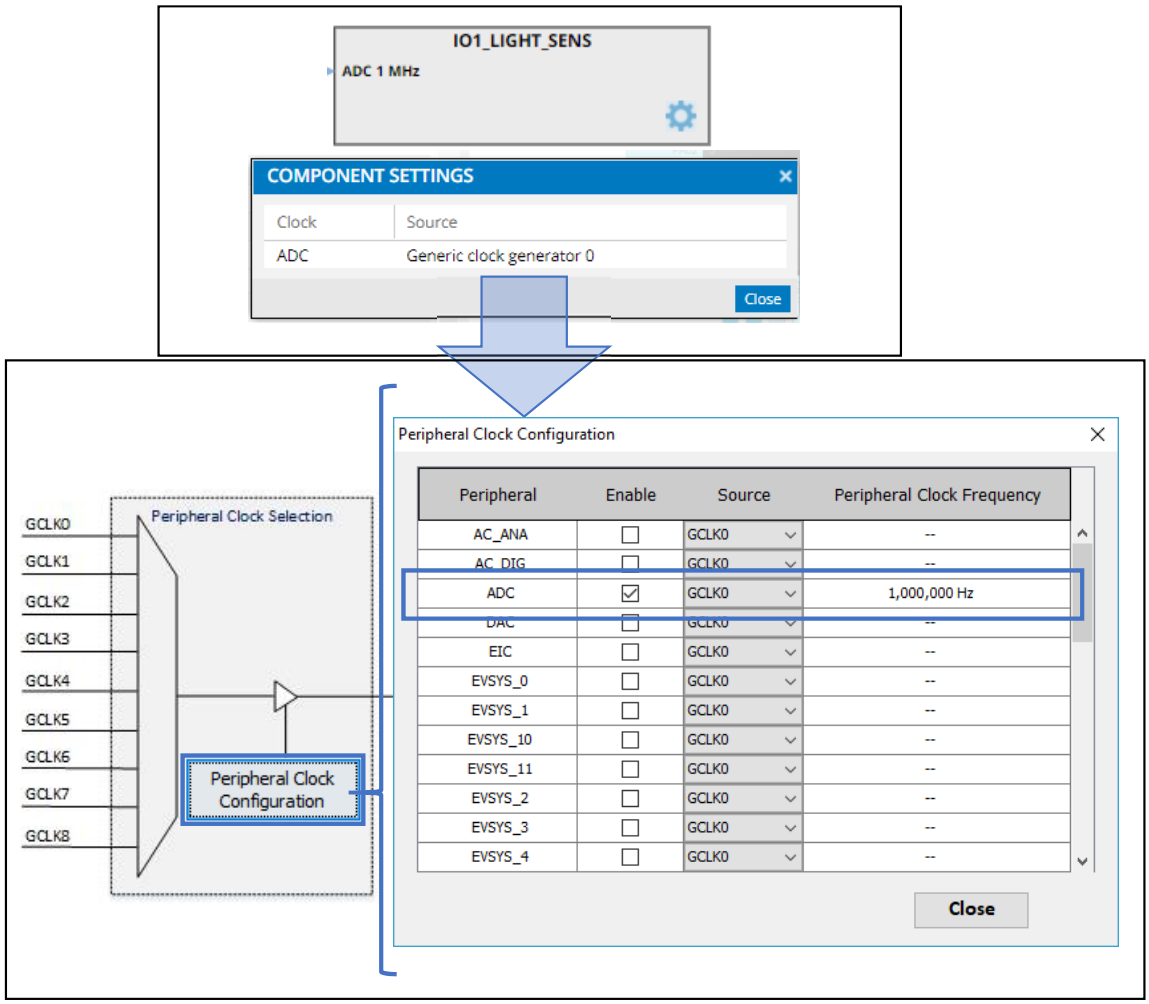
– GCLK Generator 1

Figure 5-7. GCLK Generator 1 Configuration Port



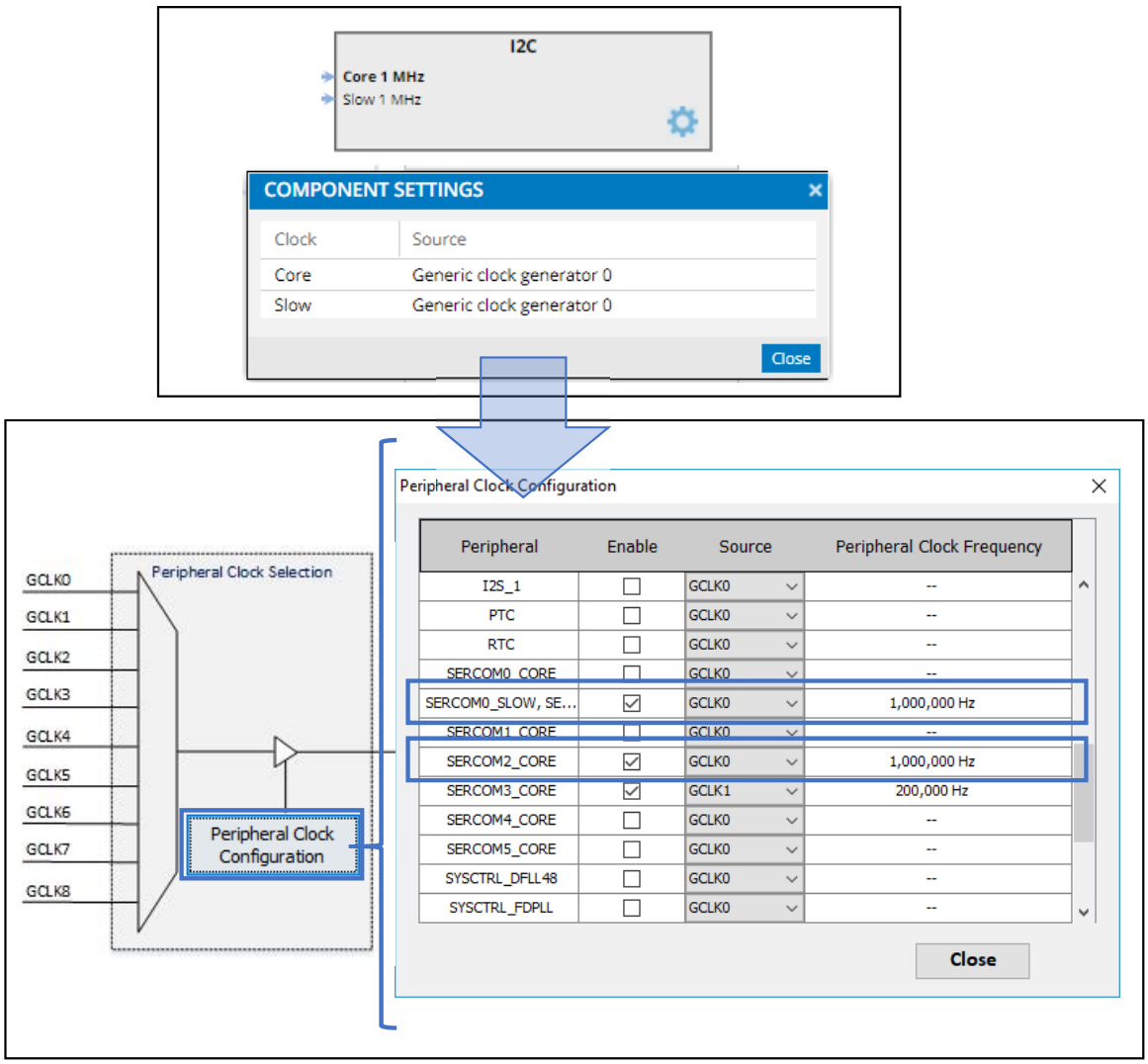
– ADC Peripheral clock

Figure 5-8. ADC Peripheral Clock Configuration Port



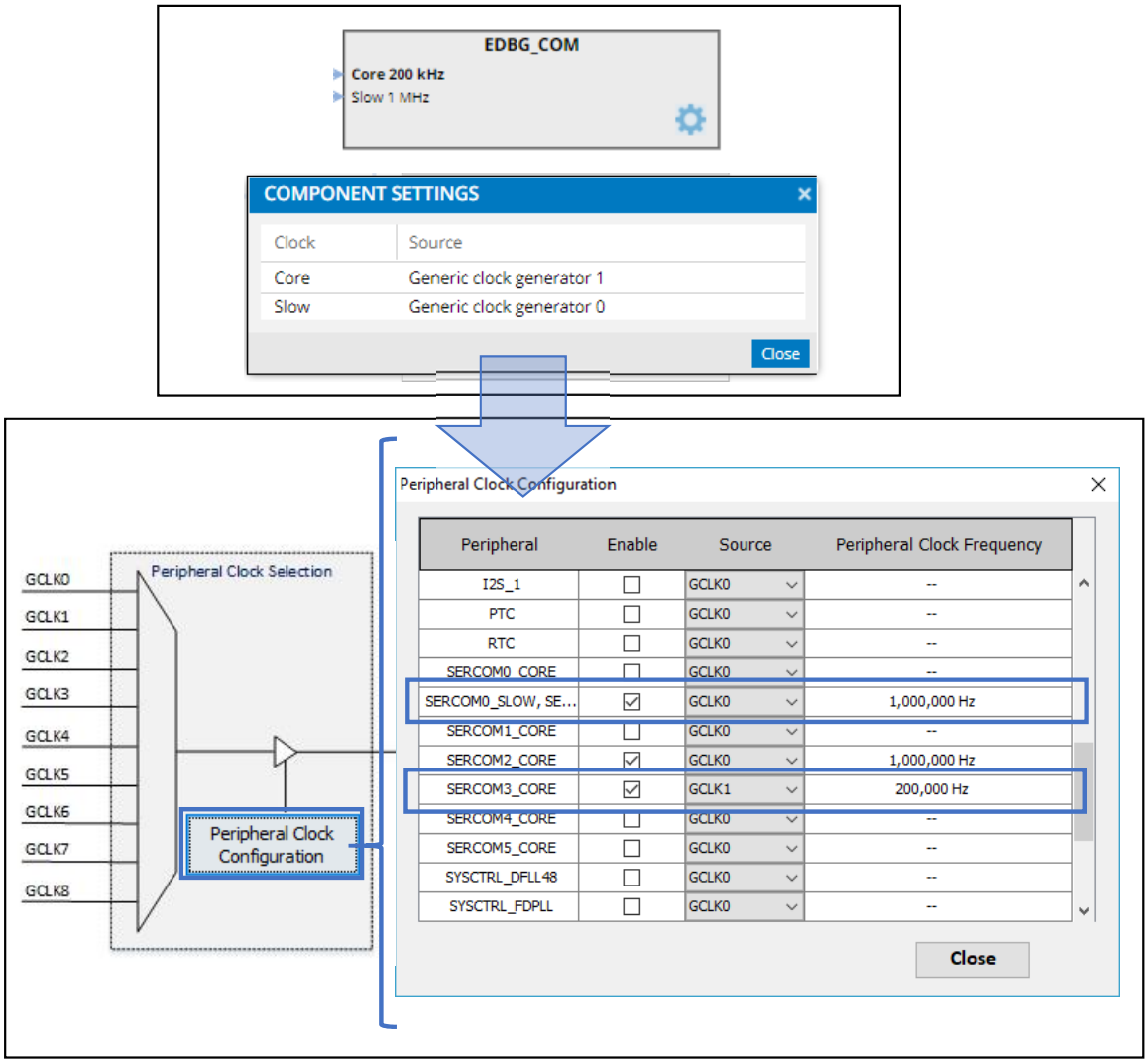
– SERCOM2 Peripheral clock

Figure 5-9. SERCOM2 Peripheral Configuration Port



– SERCOM3 Peripheral clocks

Figure 5-10. SERCOM3 Peripheral Configuration Port

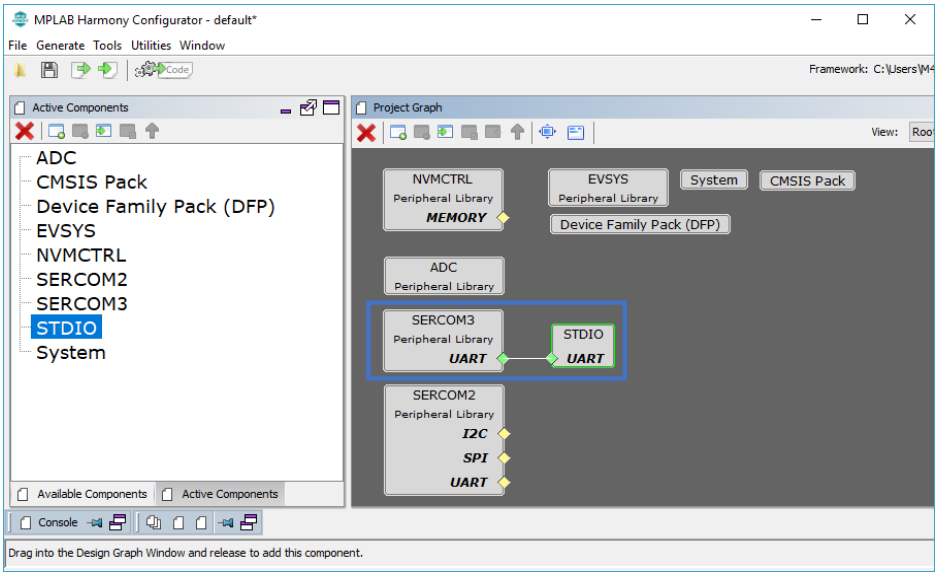


6. Porting and Adapting Middleware

The complexity behind the porting of middleware depends on its availability in the MPLAB Harmony 3 framework. The following two approaches can be used for porting middleware:

1. If middleware is available under MPLAB Harmony 3, include the middleware in the MHC project graph and link it to the associated driver. MPLAB Harmony 3 will be automatically generate a Hardware Abstraction Layer customized for this driver and middleware combination.

   Figure 6-1. STDIO Middleware or Service Integration Under MHC



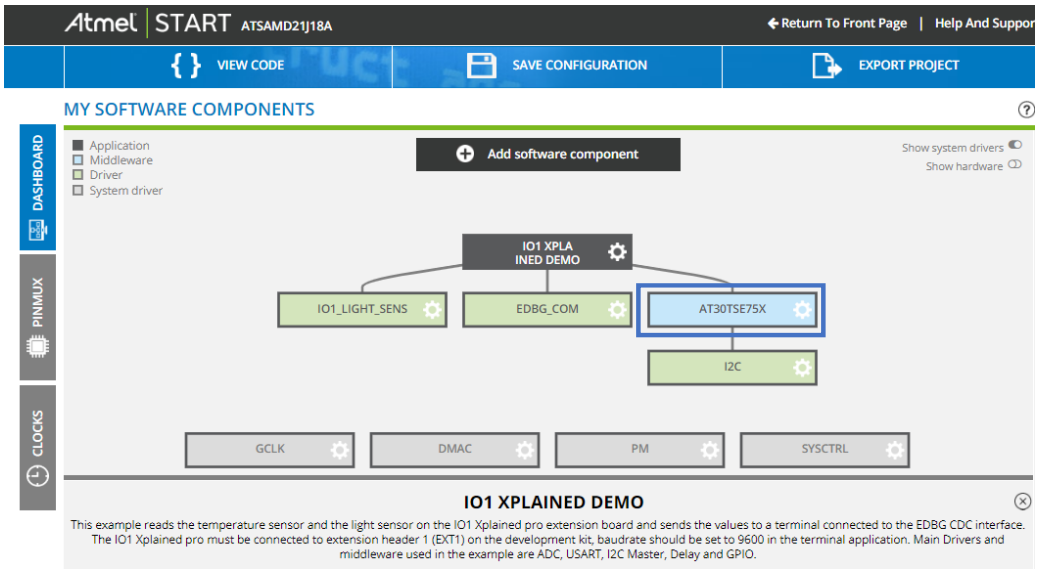
2. If the middleware is not available under Harmony 3, the user must include the source of the middleware in the new project, then locate and modify the Hardware Abstraction Layer to call the APIs provided in the Peripheral Library.

The following steps illustrate the process to be followed for the SAM D21 I/O1 demonstration.

By analyzing the structure of the application (Step 1), using both the Atmel START Dashboard and the application source code, the following middlewares are used in the Atmel START source application:

- AT30TSE75X : temperature sensor
- STDIO redirection (implicitly added)

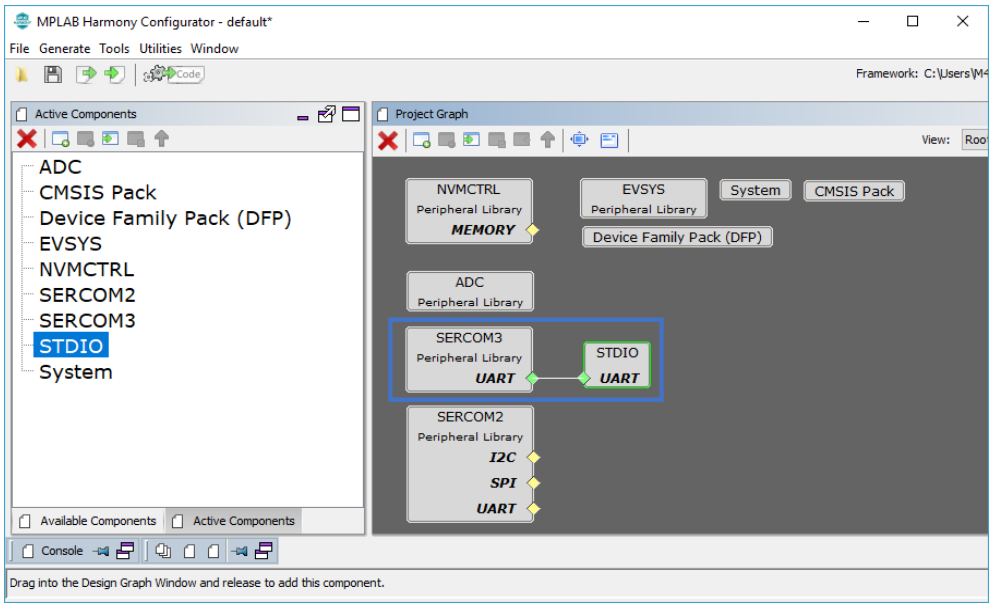
Figure 6-2. Middleware Identification Under Atmel START



The AT30TSE is currently not available under MPLAB X IDE, therefore a manual source port must be created. This step will be covered in step seven when porting the application source code.

1. In MPLAB X IDE, add the STDIO middleware to the project using the MHC Project Graph window then link the STDIO middleware to the SERCOM3 driver.

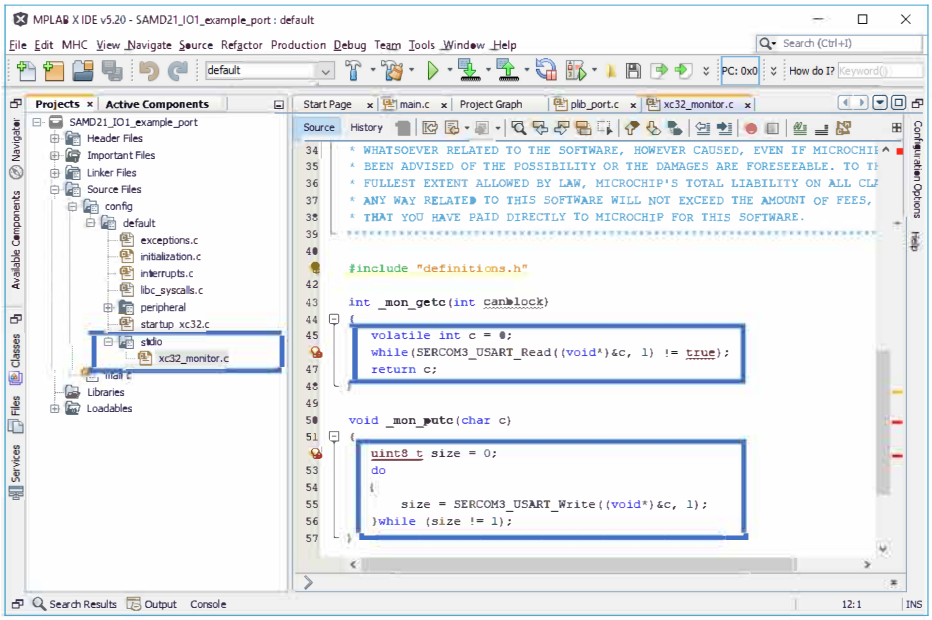
   Figure 6-3. Add STDIO Middleware Under MHC



After regenerating the project source from the MHC interface by clicking 

, the STDIO redirection HAL will be generated according to the graphical configuration. The `getc` and `putc` functions will be redirected to the SERCOM3 APIs.

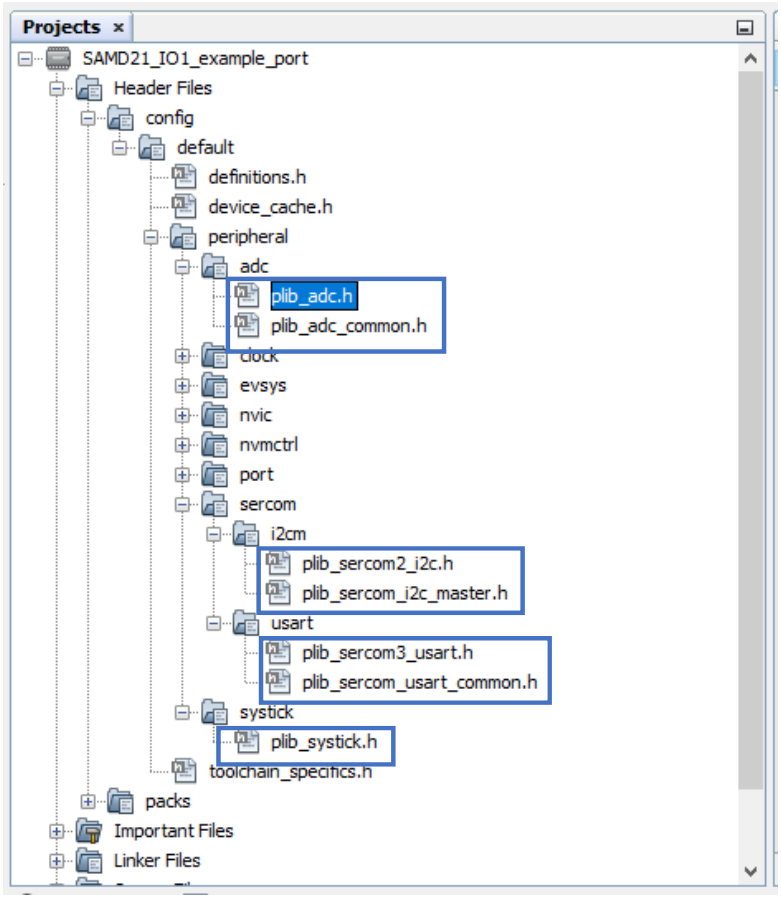
Figure 6-4. MHC Generated STDIO Redirection



7. Porting the Application Source Code

After drivers, system, and middleware layers are ported to the MPLAB Harmony 3 project, all available peripheral libraries and middleware APIs required to port the application are available in the project directory.

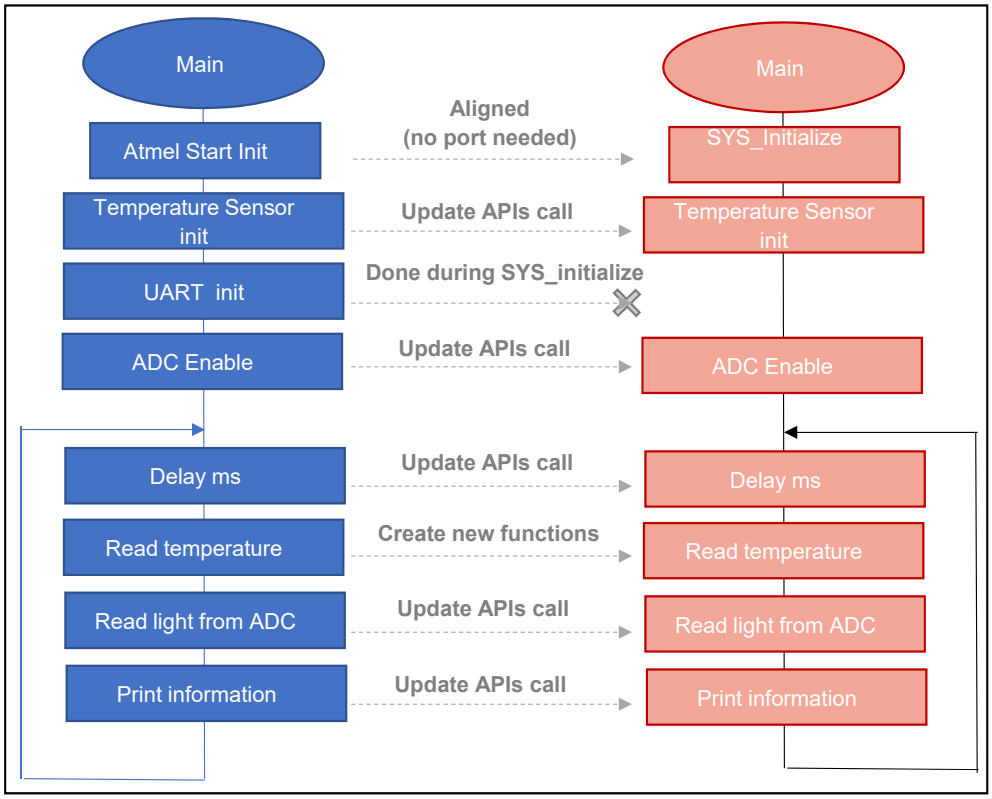
Figure 7-1. MPLAB Harmony 3 Drivers and Middleware APIs Location



Porting the application layer requires the review of the Atmel START main application routine and associated application layer related functions, and modifying them to make use of the MPLAB Harmony 3 APIs instead of the Atmel START ones.

The following figure illustrates the effort to port the application layer of the *SAM D21 IO1 Xplained demo* example to MPLAB Harmony 3.

Figure 7-2. SAM D21 I/O1 Xplained Demo Application Layer Porting Effort



The result of the SAM D21 I/O1 Xplained demo application layer porting is shown below :

```
#include <stddef.h>                     // Defines NULL
#include <stdbool.h>                    // Defines true
#include <stdlib.h>                     // Defines EXIT_FAILURE
#include "definitions.h"                // SYS function prototypes

#define AT30TSE_SENSOR_ADDRESS 0x4F
#define STR_SIZE 50

uint16_t adc_read_channel(void);
bool temperature_sensor_init(void);
float temperature_sensor_read(void);

uint16_t adc_read_channel(void)
{
    /*- Start ADC channel */
    ADC_ConversionStart();
    /*- Wait until result is ready */
    while(ADC_ConversionStatusGet() != true);

    return ADC_ConversionResultGet();
}

bool temperature_sensor_init(void)
{
    bool res;
    uint8_t buffer[3];

    /*- Configure the AT30TSE sensor */
    buffer[0] = 1;        // Select Configuration register
    buffer[1] = (2 << 5); // Set resolution to 11-bit
    buffer[2] = 0;
    res = SERCOM2_I2C_Write(AT30TSE_SENSOR_ADDRESS,buffer,3);

    return res;
}
```

```
float temperature_sensor_read(void)
{
    uint8_t txbuffer[2];
    uint8_t rxbuffer[2];
    uint16_t data;
    int8_t sign = 1;

    /* - Read the 16-bit temperature register. */
    txbuffer[0] = 0x0; // Select Temperature register
    SERCOM2_I2C_WriteRead(AT30TSE_SENSOR_ADDRESS,txbuffer,1,rxbuffer,2);
    /* - Convert Rx buffer content into temperature */
    data = (rxbuffer[0] << 8) | rxbuffer[1];
    sign -= (bool)(data & (1 << 15)) << 1;
    data &= ~(1 << 15);
    data = data >> (7 - 2);

    return (float)data * sign * (0.5 / (1 << 2));
}
int main ( void )
{
    uint16_t light_val;
    uint16_t temp_result;

    /*- Initialize all modules */
    SYS_Initialize (NULL);
    SYSTICK_TimerStart();
    /*- Enable ADC */
    ADC_Enable();
    /*- Initialize I/O1 temperature sensor  */
    temperature_sensor_init();
    while ( true )
    {
        /*- Wait 1 second */
        SYSTICK_DelayMs(1000);
        /*- Read light level */
        light_val = adc_read_channel();
        /*- Read temperature */
        temp_result = temperature_sensor_read();
        /*- Display Light and temperature information on Terminal */
        printf("Temperature: %d Celsius, light sensor: 0x%x\r\n",
        temp_result, light_val);
    }
    /*- Execution should not come here during normal operation */
    return ( EXIT_FAILURE );
}
```

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **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 (FAQs), technical support requests, online discussion groups, Microchip design partner 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

Product Change Notification Service

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

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website 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.
- 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, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL 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, BlueSky, 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, 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.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks 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.

ISBN: 978-1-5224-5403-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

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: http://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: 49-7131-72400 **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