

MPLAB Harmony v3 Project Migration

Introduction

MPLAB® Harmony v3 is a modular framework that provides interoperable firmware libraries for application development on 32-bit microcontrollers and microprocessors. It includes an easy-to-use graphical user interface (GUI), MPLAB Code Configurator (MCC), or MPLAB Harmony Configurator (MHC) for selection, configuration, code generation, peripheral libraries, drivers, and extensive middlewares such as, USB, TCP/IP, Graphics.

This document discusses the migration of an existing MPLAB Harmony v3 based project developed on a particular hardware (microcontroller or development board) platform to another one of Microchip's 32-bit hardware platforms of the user's choice.

Note: In this document, the phrase *code generation tool* refers to any one of the Microchip code generation tools: MCC and MHC. At the time of publication of this document, the MPLAB Harmony v3 library code can be generated using either of these tools. However, in the future, only MCC will support MPLAB Harmony V3 code generation.

Table of Contents

Introduction.....	1
1. Overview.....	3
2. Migration Components.....	4
2.1. MPLAB X IDE Project.....	4
2.2. Tool Configurations.....	7
2.3. Clock.....	13
2.4. Peripheral Libraries	15
2.5. PORT Pins.....	18
2.6. Drivers.....	20
2.7. System Services.....	21
2.8. Middleware.....	23
2.9. Application.....	24
3. References.....	27
The Microchip Website.....	28
Product Change Notification Service.....	28
Customer Support.....	28
Microchip Devices Code Protection Feature.....	28
Legal Notice.....	28
Trademarks.....	29
Quality Management System.....	30
Worldwide Sales and Service.....	31

1. Overview

Typically, the application project development for an embedded product starts by using the available reference collateral. As part of the MPLAB Harmony v3 packages, Microchip provides reference application demonstrations for customers to develop their projects. These MPLAB Harmony v3 projects are developed on Microchip's development boards, such as 32-bit SAM or PIC32 microcontrollers (MCUs). The users can start with these application demonstrations and migrate them to the hardware platform of their choice.

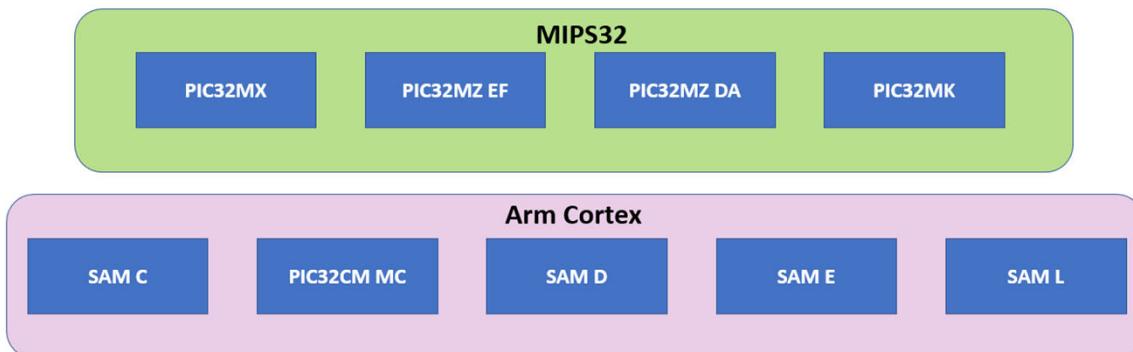
This document discusses migrating an MPLAB Harmony v3 application demonstration project developed on Microchip's development board or kit in the following scenarios.

- **Scenario #1:** Enhance or customize the existing application demonstration
- **Scenario #2:** Migrating to a custom board having an MCU with the same device ID
- **Scenario #3:** Migrating to an MCU with a different device ID within the same family
- **Scenario #4:** Migrating to an MCU with a different device ID in another MCU family with the same processor core

Notes:

1. The following figure shows some of the 32-bit microcontroller product families in the Arm® Cortex® and MIPS core architectures. For additional information on Microchip 32-bit microcontroller families, refer to <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/32-bit-mcus>.

Figure 1-1. 32-bit Microcontroller Families



2. Several MCU parts with varying features are provided within each family shown for Arm Cortex and MIPS32 core architectures, for example, the PIC32MX family has [PIC32MX250F128B](#), [PIC32MX470F512L](#), [PIC32MX795F512L](#), and so on.
3. Scenario #3 and Scenario #4, as mentioned above, covers migration within the family of devices and outside of the family, but within the same core architecture.
4. There could be another possible scenario, Scenario #5, wherein a user would like to migrate an application developed on a MIPS-based MCU to an Arm Cortex-based MCU, or Arm to MIPS. This document does not cover such a scenario, because the processor architectures are entirely different, creating a project from scratch is advised instead of attempting the migration.

2. Migration Components

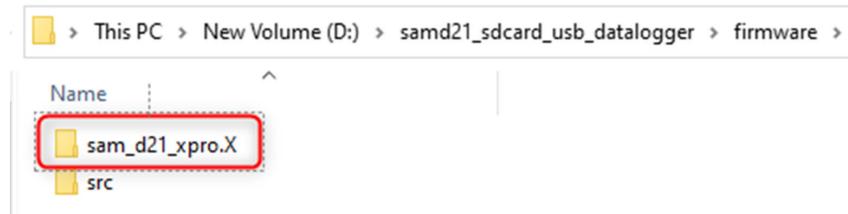
In the following sections, the MPLAB Harmony v3 project [SD Card, USB Data Logger Application on SAM D21 Xplained Pro Evaluation Kit](#) is used as a reference and referred to as the 'Reference Project'.

2.1 MPLAB X IDE Project

The MPLAB X IDE project configuration parameters to be considered for migration are as follows.

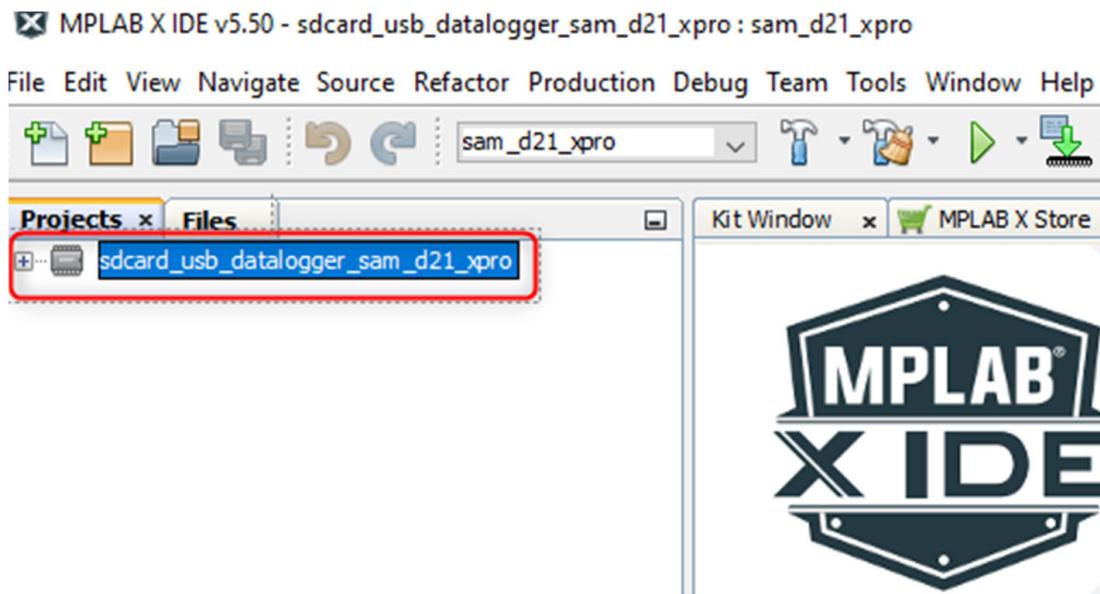
1. **Location:** The path to the root folder of the project.
2. **Folder name:** The name of the .x folder.

Figure 2-1. Project Folder Name



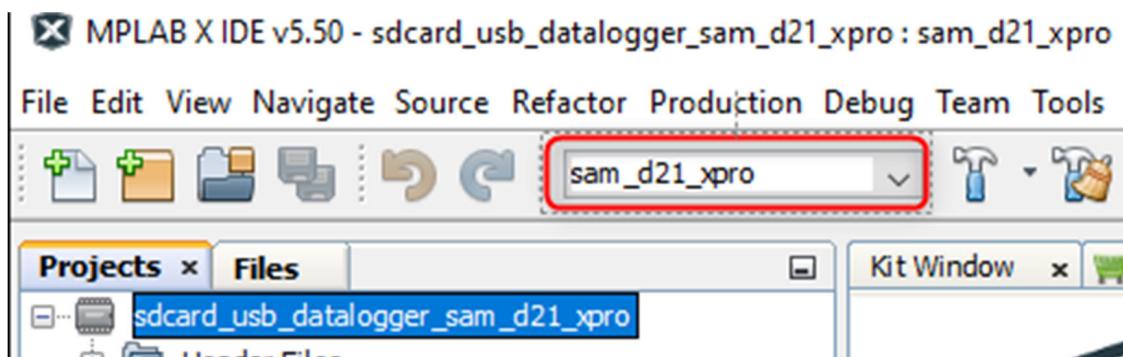
3. **Name:** The project's logical name, which is available in the MPLAB X IDE project explorer.

Figure 2-2. MPLAB X Project Name



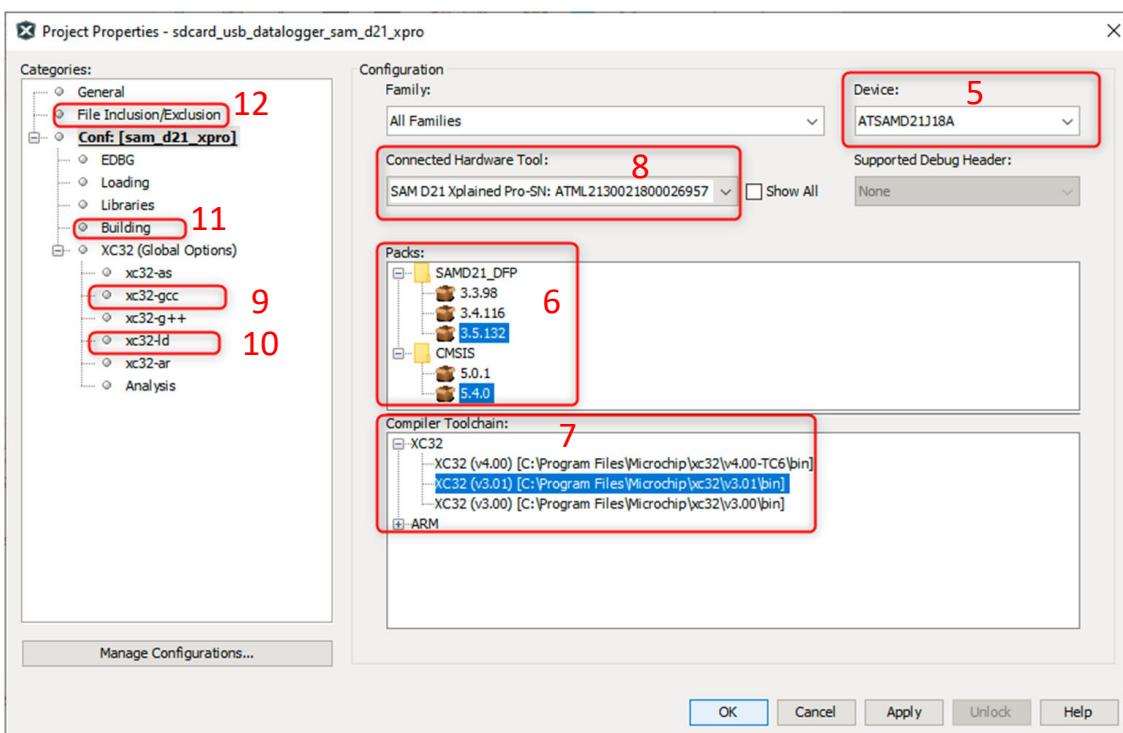
4. **Configuration name:** A feature in MPLAB X IDE that enables a platform or target specific build and programming.

Figure 2-3. MPLAB X Project Configuration Name



5. **Target device:** The MCU device ID on which the project is developed.
6. **Support packages:** The device family packages (DFP) include the macro definitions to access the peripheral registers of the target MCU. On Arm Cortex devices, the CMSIS packages are also included.
7. **Compiler toolchain:** This shows the compiler toolchain and its version.
8. **Programming or debugging tool:** The hardware tool used for debugging or programming the MCU.
9. **Build options:** The compiler build options and any other options enabled for source code optimization.
10. **Linker options:** The default settings for the heap, stack memory, and other related parameters.
11. **Pre or Post-build scripts:** Specifies any scripts or command lines that must be run before or after the project build.
12. **Source and library files reference:** The relative path to the MPLAB Harmony v3 generated, user-added source, header, and library files.

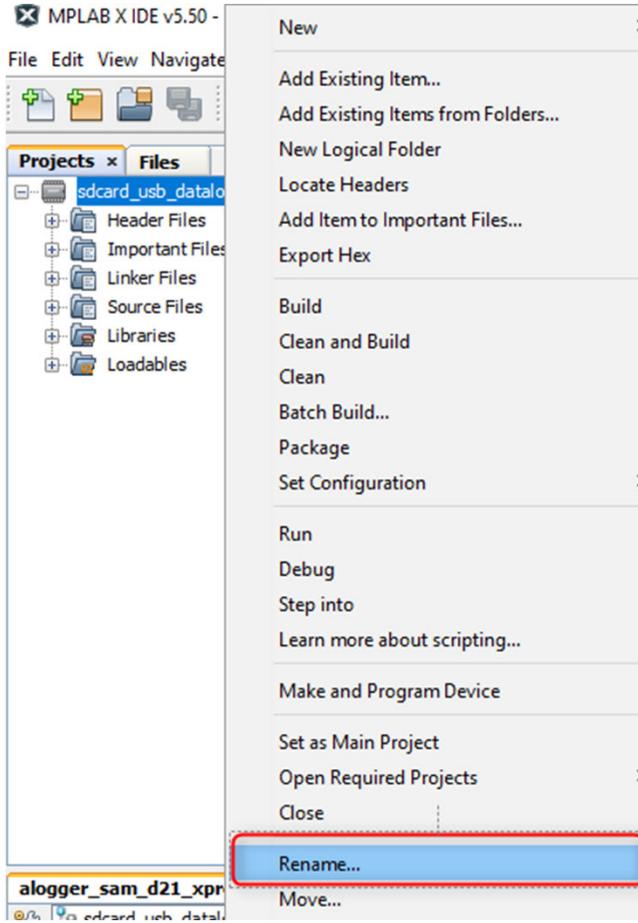
Figure 2-4. MPLAB X Project Properties



The MPLAB X IDE configuration parameters are saved in the `configurations.xml` file in the MPLAB X IDE project folder. The Reference Project is found at `samd21_sdcard_usb_datalogger\firmware\sam_d21_xpro.X\ nbproject`.

- For Scenario #1, the existing MPLAB Harmony v3 application demonstration developed on Microchip's development board or kit is used and customized to add or remove MPLAB Harmony v3 modules. The existing MPLAB X IDE project's configuration parameters will suffice and do not require changes. However, the user can change some parameters to suit the end application.
 - To rename the MPLAB X IDE project and folder names, open the project in MPLAB X IDE, and then right-click on the project name, and select **Rename**.

Figure 2-5. MPLAB X Project Renaming



- To rename an MPLAB X IDE configuration parameter, refer to the technical brief [How to Add a New Configuration to an Existing MPLAB Harmony v3 Project](#). Renaming a configuration is not advised for scenario #1, as renaming a configuration in MPLAB Harmony v3 essentially creates a new MPLAB X IDE project.
- The other MPLAB X IDE parameters, listed in Step 5 through Step 12, can be changed or modified through the Project Properties window shown in [Figure 2.4](#).
- For Scenarios #2, #3, and #4, for the MCU device ID, under the **Adding a New Configuration section**, go to the sub section **Creating a project in MPLAB X IDE**, and then follow the details given in the document [How to Add a New Configuration to an Existing MPLAB Harmony v3 Project](#). Ensure to provide the required MPLAB X IDE configuration parameters listed in the preceding Step 1 through Step 12, specifically, the Target device ID.

2.2 Tool Configurations

The MCC or MHC tool saves the configurations of all the components used in the project graph in separate files. Additionally, it allows the import and export of these configurations.

The action of saving the selected component's configuration to an external file is called export. Import reads the external configuration of a component, or a group of components saved in an external file and incorporates that configuration in the current project.

The MCC tool configuration file, .mc3, is available in the MPLAB X IDE project folder. For the Reference Project, it is available in the following location: `samd21_sdcard_usb_datalogger\firmware\src\config\sam_d21_xpro.X`.

The MHC tool configuration file is available in the MHC folder. For the Reference Project, it is available in the following location: `samd21_sdcard_usb_datalogger\firmware\src\config\sam_d21_xpro\sam_d21_xpro.mhc`.

The import feature is useful for the migration scenarios discussed above.

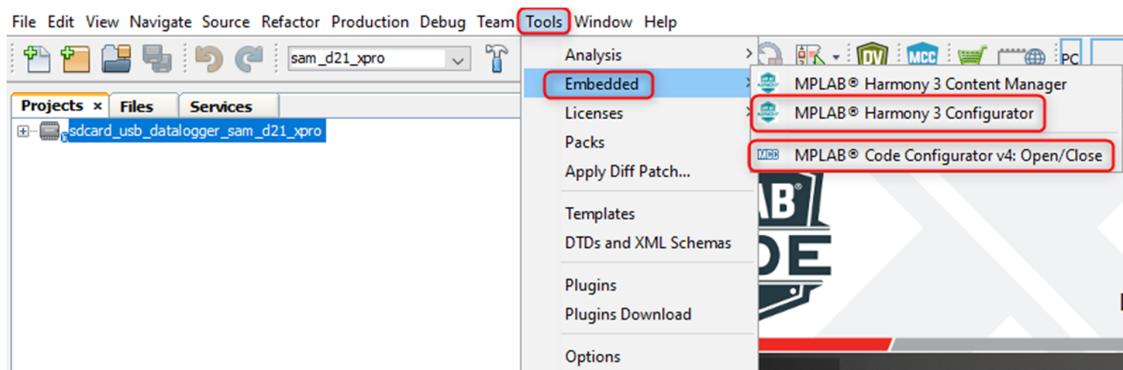
Launching MCC:

1. Open the Reference project in MPLAB X IDE.
2. To launch MCC go to *Tools > Embedded > MPLAB Code Configurator v4*.

Launching MHC:

1. Open the Reference Project in MPLAB X IDE.
2. To launch the MHC go to *Tools > Embedded > MPLAB Harmony 3 Configurator*.

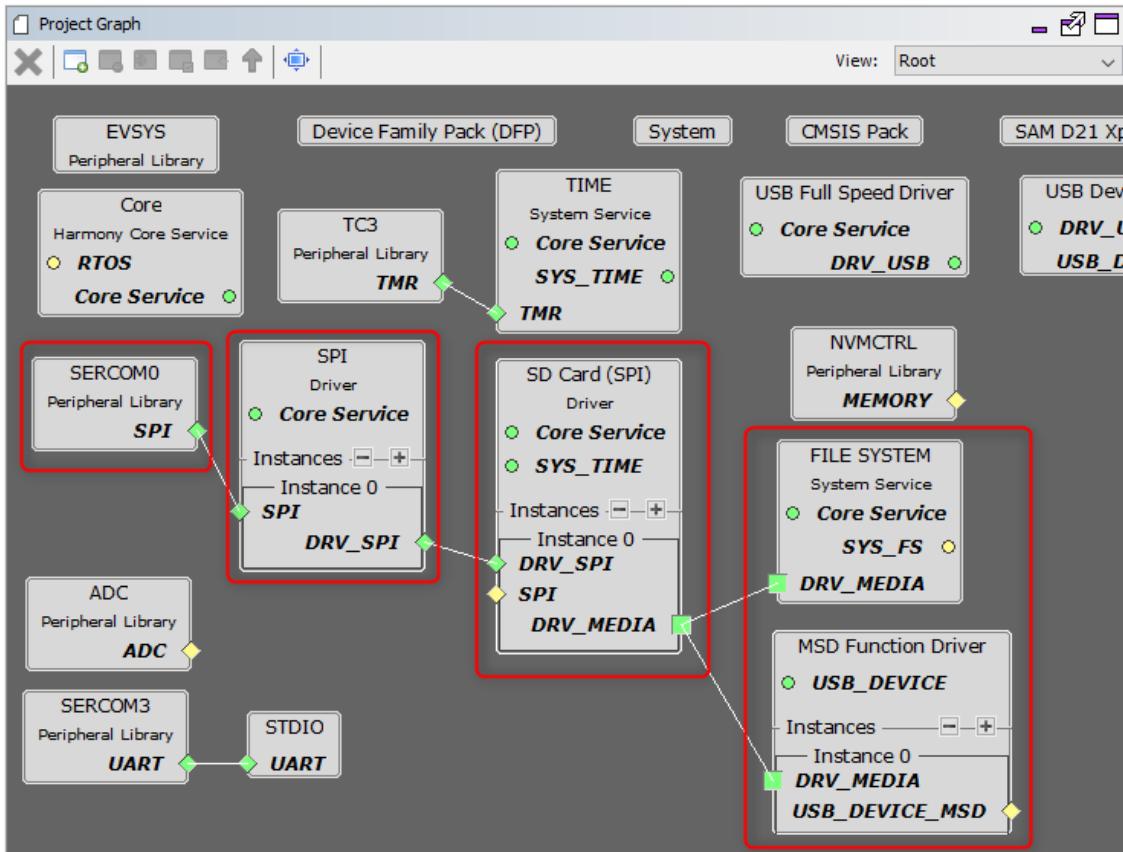
Figure 2-6. Launching MCC or MHC



For example, if the user wants the following configurations from the Reference Project, the export and import features in the MCC or MHC tool can be used to bring these features into the new project:

- SD Card SPI Driver
- File System Service
- MSD Function Driver
- SPI Driver
- SERCOM PLIB

Figure 2-7. Reference Project Configuration

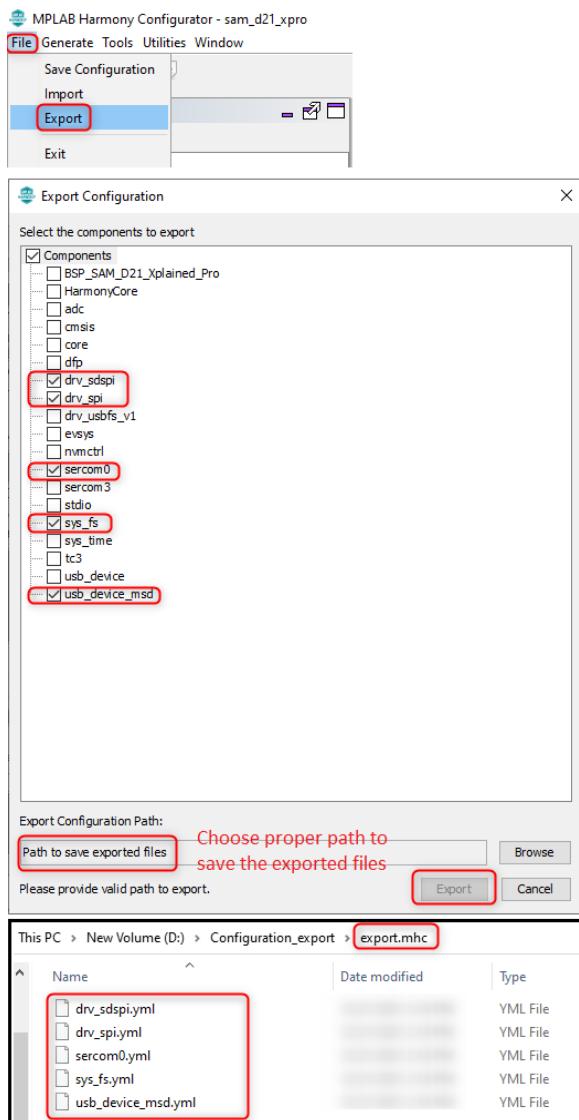


Importing and Exporting MHC

Follow these steps to import and export MHC:

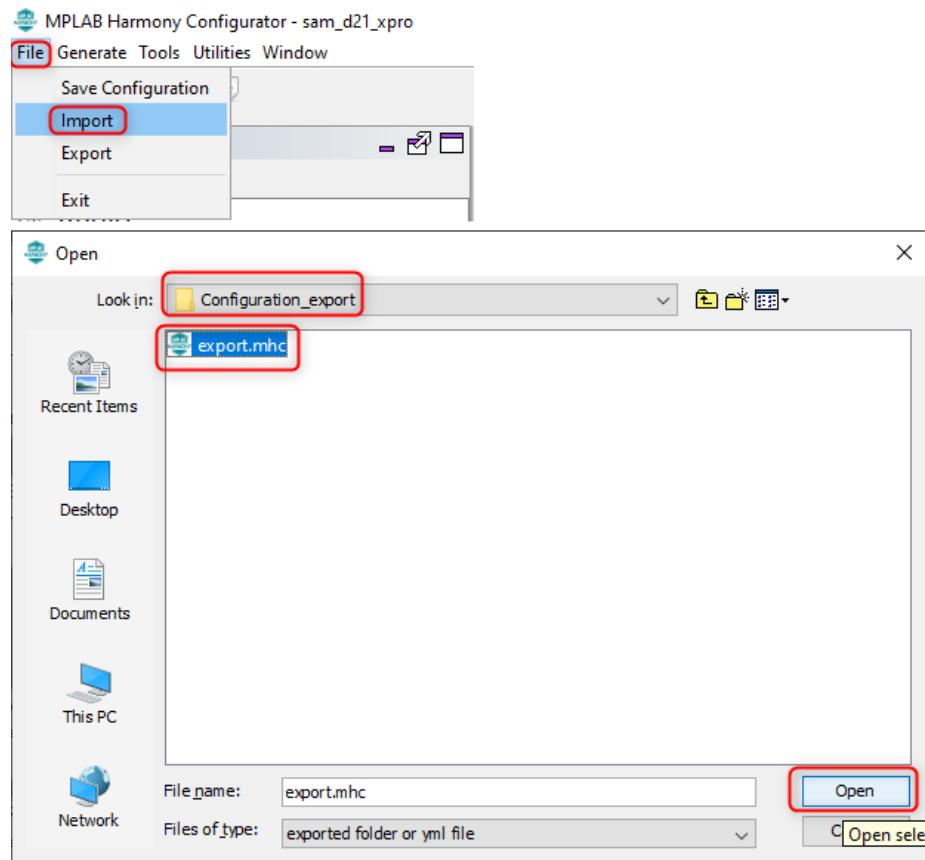
1. In the MHC, from *File > Export*.
2. Select the required components from the **Export Configuration** dialogue box.
3. Specify the path to save the exported files, and then click **Export** to export the selected component configurations, as shown in the following figure.

Figure 2-8. MHC Export



4. In the MHC new project, select *File > Import* to redirect to the path where the export files are saved.
5. Select the file to export and click **Open**.

Figure 2-9. MHC Import



6. The user can also import individual configurations, by selecting one .yml file at a time from the .mhc folder.

Figure 2-10. Importing an Individual Configuration on the MHC

This PC > New Volume (D:) > Configuration_export > **export.mhc**

Name	Date modified	Type
drv_sdspi.yml		YML File
drv_spi.yml		YML File
sercom0.yml		YML File
sys_fs.yml		YML File
usb_device_msdu.yml		YML File

Note: The user can also import directly from the saved peripheral configuration files (.yml) under the .mhc folder of the Reference Project source files.

Figure 2-11. Import Using Saved Peripheral Configuration Files Under the .mhc Folder

The screenshot shows a file explorer window with the following details:

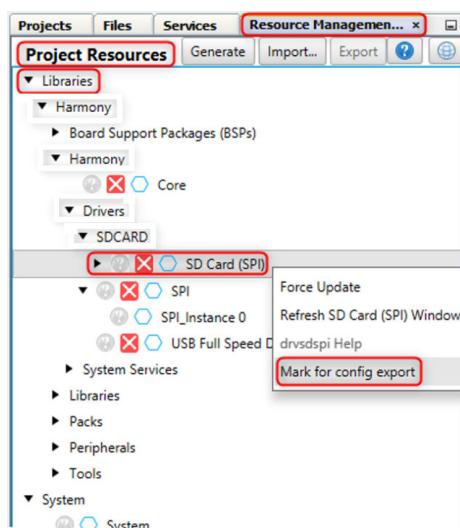
- Path:** firmware > src > config > sam_d21_xpro > sam_d21_xpro.mhc
- File List:**
 - adc.yml
 - BSP_SAM_D21_Xplained_Pro.yml
 - cmsis.yml
 - core.yml
 - dfp.yml
 - drv_sdspi.yml
 - drv_spi.yml
 - drv_usbfs_v1.yml
 - evsys.yml
 - GraphSettings.yml
 - HarmonyCore.yml
 - nvmctrl.yml
 - project.yml
 - sercom0.yml
 - sercom3.yml
 - settings.yml
 - stdio.yml
 - sys_fs.yml
 - sys_time.yml
 - tc3.yml
 - usb_device.yml
 - usb_device_msd.yml
- Search Bar:** Search sam_d21_xpro.

Importing or Exporting MCC

Follow these steps to import and export MCC:

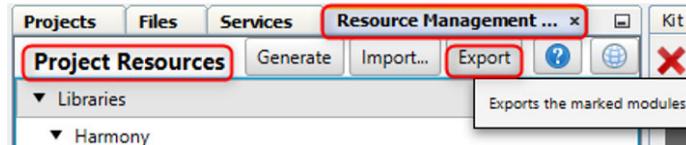
1. In MPLAB X IDE, click **Resource Management (MCC)**.
2. Click the **Project Resources** tab and expand Libraries.
3. Right-click on the required component and select **Mark for config export**.

Figure 2-12. Mark for Export on MCC



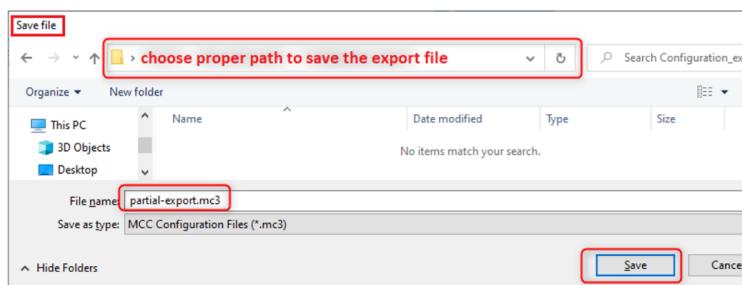
4. Follow the preceding steps for all remaining components, such as SERCOM0 in Peripherals, SPI in Drivers, File System in System Services, and MSD Function Driver in USB device stack libraries.
5. After marking all the required components for export, click **Export**.

Figure 2-13. Export on MCC



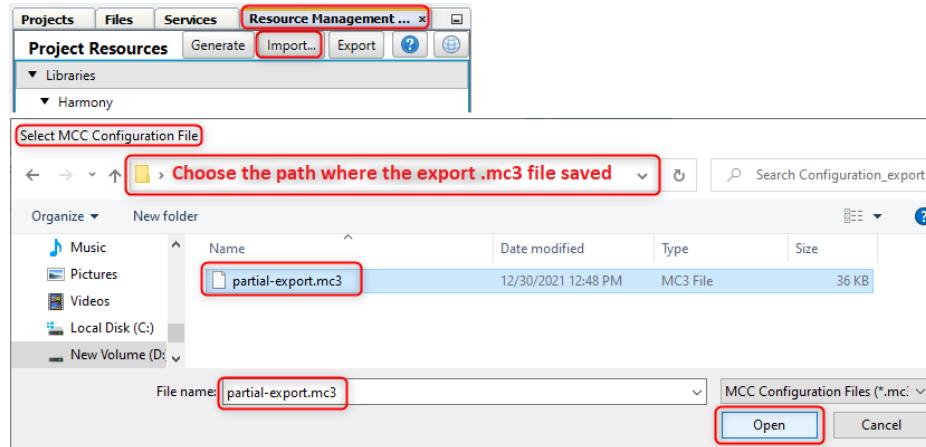
6. Specify the path to save the export file, .mc3, and then click **Save**.

Figure 2-14. Save Export File



7. In MPLAB X IDE, in the new project, select *Resource Management (MCC) > Import* to redirect to the path where the export file .mc3 is saved.
8. Select the export file and click **Open**.

Figure 2-15. MCC Import



2.3 Clock

An essential element in migrating embedded application projects is the CPU clock and the peripheral clocks generated based on the CPU clock.

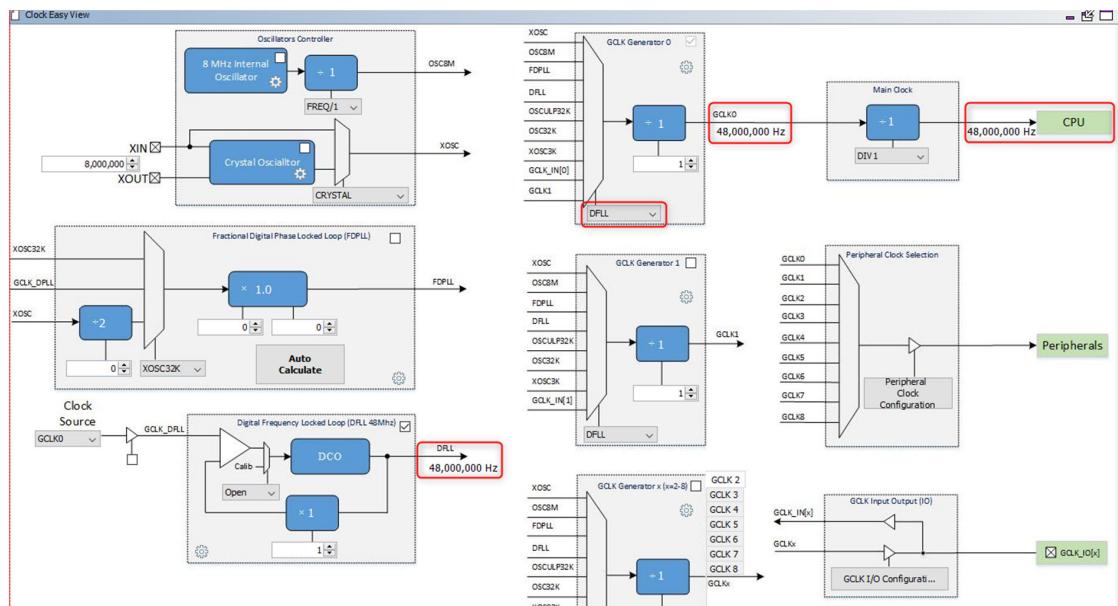
For Scenarios #1 and #2, the CPU clock will remain the same, because the MCU device ID is the same. In these scenarios, the existing clocks configured in the MPLAB Harmony v3 demonstration application can be used in the user application. If required, the user can change the clock.

For Scenario #3, though the clock module may be the same within a family of devices, the clock speed of the target MCU can be different. This change means the clock for the target MCU must be changed. The user needs to verify the clock setting and change as required.

For Scenario #4, there is a higher probability that the clock module may be different from the MCU running the existing MPLAB Harmony v3 demonstration application to the MCU device ID being migrated to. This means the clock for the target MCU needs to be changed. The user needs to verify the clock setting and change it as required.

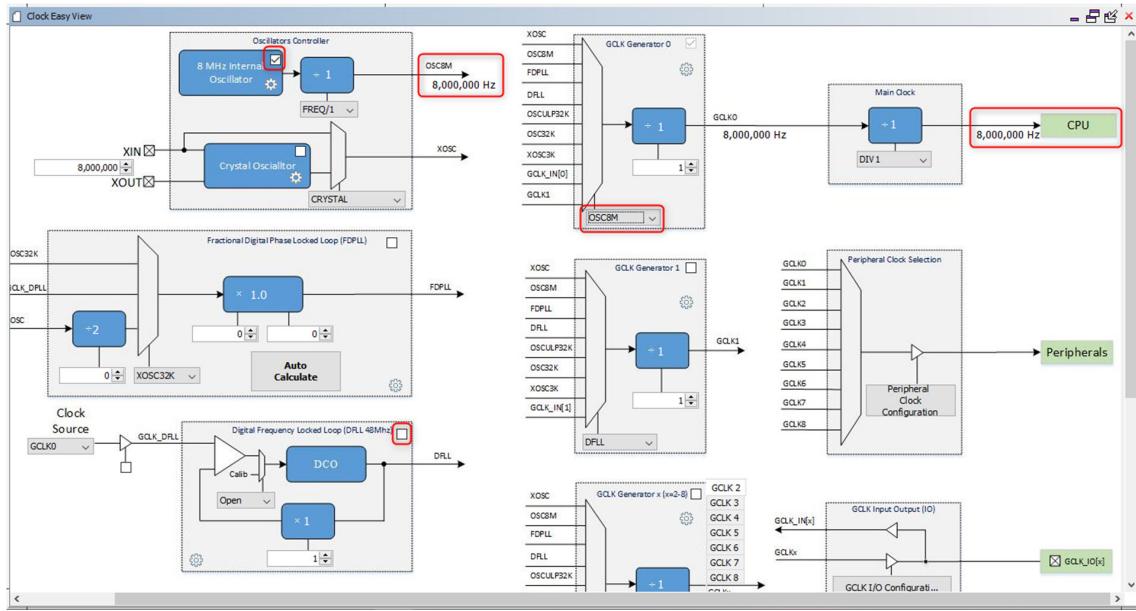
The CPU clock configuration in the Reference Project is shown in the following figure:

Figure 2-16. Reference Project CPU Clock Settings



The following figure shows changing the CPU clock source from the DFLL to the internal Oscillator 8 MHz (OSC8M), and disabling the DFLL clock source.

Figure 2-17. Change the CPU Clock Source



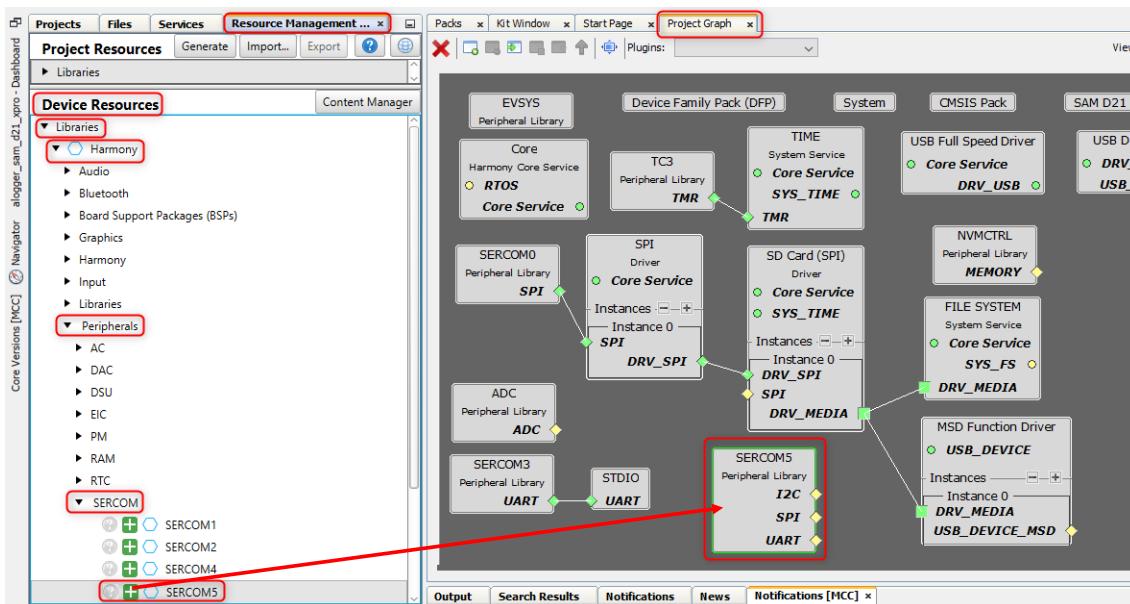
2.4 Peripheral Libraries

The MPLAB Harmony v3 Peripheral Libraries (PLIBs) provide a Hardware Abstraction Layer (HAL) interface to access peripherals on MCUs. The PLIB Application Program Interfaces (APIs) hide peripheral register details and make it easier to configure the peripheral per the application requirements by calling the PLIB APIs.

The PLIBs are added to the project and configured through the code generation tool (MCC or MHC) project graph.

- For Scenario #1, there is no special consideration for adding, deleting, modifying, or retaining existing PLIBs. Launch the code generation tool (MCC or MHC) project graph, perform any of these operations, and generate the code. For example, the following figure shows adding a new PLIB to the Reference Project. The user must configure each newly added peripheral.

Figure 2-18. Adding a New PLIB



- For Scenarios #2, #3, and #4
 - Ensure to import the required code generation tool (MCC or MHC) configurations from the existing MPLAB Harmony v3 application demonstration. Refer to [Tool Configurations](#) for more information.
 - For the PLIBs added in the MPLAB Harmony v3 application demonstration, the user must verify the peripherals routed in the custom or user board. Suppose different peripherals or a different instance of the same peripheral is used. In that case, the PLIB blocks of the code generation tool (MCC or MHC), and the project graph must be updated. The project requires regeneration.
- For example, if SERCOM2 is used instead of SERCOM3 as the UART debug interface in the custom board, the SERCOM block must be updated by deleting SERCOM3 and adding the SERCOM2 in the Reference Project graph as shown in the following figure. The user must configure each newly added peripheral.

Figure 2-19. The Existing UART Debug SERCOM Instance

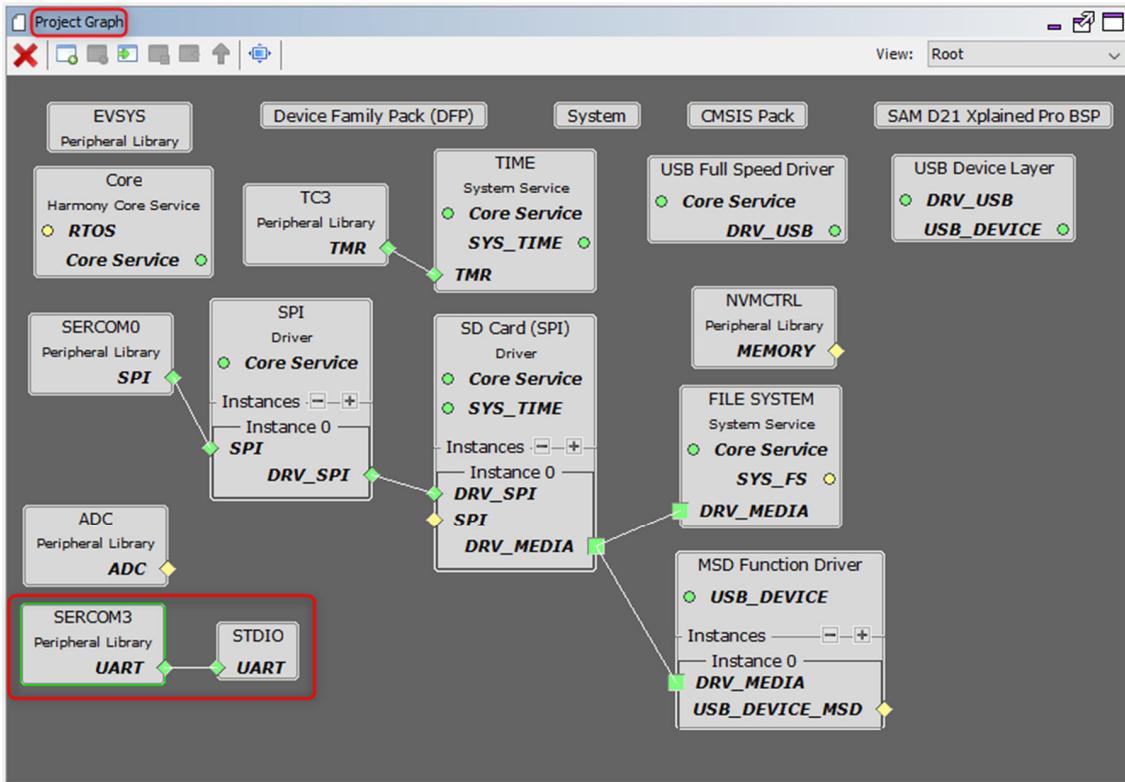
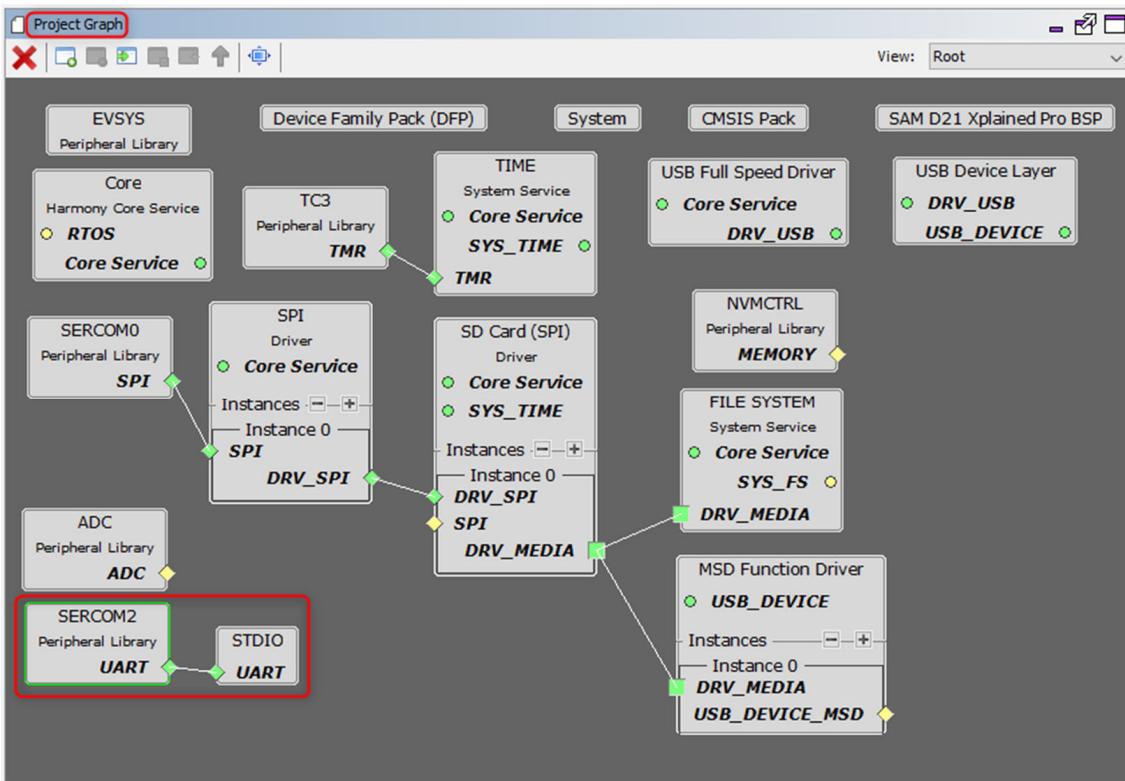


Figure 2-20. The Updated UART Debug SERCOM Instance



- If the user has changed the PLIBs used in the project graph, the same change must be reflected in the API calls if the application calls the PLIB APIs. For example, if the UART debug interface has changed, the PLIB block in the project graph from SERCOM3 to SERCOM2, the same must be changed in the following APIs:
 - Replace `SERCOM3_USART_Write` API in the application code with `SERCOM2_USART_Write`.

2.5 PORT Pins

The PORT pins on the SAM and PIC32 MCUs have multiplexed functionality, which means keeping track of available or assigned pins and deciding on assigning I/O pins to a given peripheral can be challenging. The challenge is aggravated when migrating a project from one platform to another.

The code generation tool (MCC or MHC) provides the Pin Configuration window to configure or reconfigure the PORT pins.

- For Scenario #1, if a new pin functionality is added or removed by adding or removing PLIBs, the configurations of the corresponding pins can be performed through the code generation tool (MCC or MHC) in the Pin Configuration window. For example, the following figure shows the configuration pins of the PLIB added in Scenario#1 in PLIBs.

Figure 2-21. Configure Pins for Adding a New PLIB

The screenshot shows the Pin Settings window with the 'Ports' tab selected. The table lists pins PB10 through PB17. Pins PB16 and PB17 are highlighted with red boxes around their 'Custom Name' and 'Function' columns. Both are set to 'SERCOM5_PAD0' and 'SERCOM5_PAD1' respectively. Other pins are listed as 'Available'.

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
23	PB10		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
24	PB11		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
25	PB12		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
26	PB13		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
27	PB14		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
28	PB15		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
39	PB16		SERCOM5_PAD0	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
40	PB17		SERCOM5_PAD1	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
49	PR??		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

- For Scenarios #2, #3, and #4:
 - Be sure to import the required code generation tool (MCC or MHC) configurations from the existing MPLAB Harmony v3 application demonstration. Refer to the [Tool Configurations](#) for additional information.
 - For the PLIBs added in the MPLAB Harmony v3 application demonstration, the user must verify the peripherals routed in the custom or user board. If different peripherals or different instances of the same peripheral are used, the pins for the peripheral also need to be verified and updated in the code generation tool (MCC or MHC) Pin Configuration window. The project will require regeneration.
- For example, if SERCOM2 is used instead of SERCOM3 as the UART debug interface in the custom board using different UART pins, then the debug UART pins must be reconfigured in the Reference Project graph, as shown in the following figure.

Figure 2-22. The Existing SERCOM Instance Pin Configuration

The screenshot shows the Pin Settings window with the 'Ports' tab selected. The table lists pins PA22 and PA23. Both are highlighted with red boxes around their 'Custom Name' and 'Function' columns. Both are set to 'SERCOM3_PAD0' and 'SERCOM3_PAD1' respectively. Other pins are listed as 'n/a'.

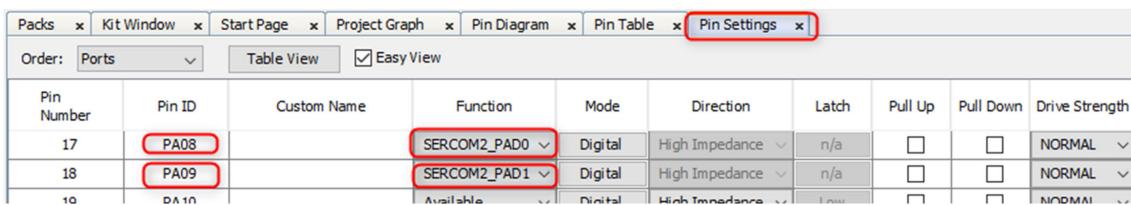
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
43	PA22		SERCOM3_PAD0	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
44	PA23		SERCOM3_PAD1	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

Figure 2-23. Reconfigure the Existing SERCOM3 Instance Pins as Available

The screenshot shows the Pin Settings window with the 'Ports' tab selected. The table lists pins PA22 and PA23. Both are highlighted with yellow boxes around their 'Custom Name' and 'Function' columns. Both are now set to 'Available'. Other pins are listed as 'n/a'.

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
43	PA22		Available	Analog	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
44	PA23		Available	Analog	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

Figure 2-24. Reconfigure Pins for Updated Debug SERCOM2 Instance



The screenshot shows a software interface for pin configuration. The top navigation bar includes tabs for 'Packs', 'Kit Window', 'Start Page', 'Project Graph', 'Pin Diagram', 'Pin Table', and 'Pin Settings'. The 'Pin Settings' tab is active, indicated by a red border around its tab. Below the navigation bar, there are filter options: 'Order: Ports' (selected), 'Table View' (selected), and 'Easy View' (unchecked). A table lists pins 17 and 18, both of which have their 'Pin ID' (PA08 and PA09 respectively) and 'Custom Name' (SERCOM2_PAD0 and SERCOM2_PAD1) highlighted with red boxes. The table also shows the 'Function' (Digital), 'Mode' (High Impedance), 'Direction' (n/a), 'Latch' (n/a), and 'Drive Strength' (NORMAL) for each pin.

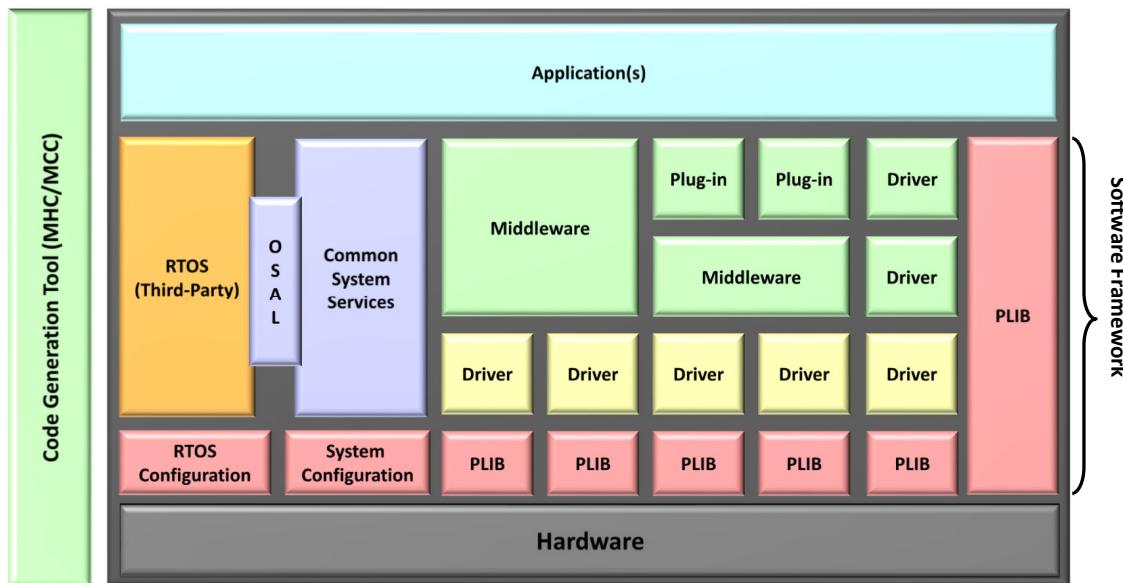
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
17	PA08	SERCOM2_PAD0	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
18	PA09	SERCOM2_PAD1	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
19	PA10	Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

2.6 Drivers

The MPLAB Harmony v3 drivers are an interface to control, access, manage peripherals, and other resources on the 32-bit SAM and PIC32 microcontrollers. The driver interface allows applications and other client modules (drivers, middleware libraries, and system services) to interact with the peripheral.

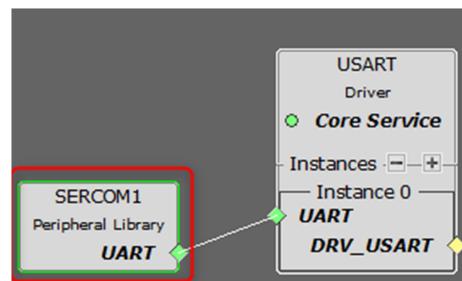
The following figure illustrates the MPLAB Harmony v3 architecture showing the locations of the PLIBs and drivers. The PLIBs have direct access to the hardware, where the drivers are built on the top of the PLIBs and provide an abstracted interface to other MPLAB Harmony v3 software components (plug-ins, Middleware, and system services) and applications.

Figure 2-25. MPLAB Harmony v3 Architecture



- For Scenario #1, there are no special considerations for adding, deleting, modifying, or retaining existing drivers. Launch the code generation tool (MCC or MHC) project graph, perform any of these operations, and generate the code. For example, if the Reference Project has an additional requirement to use the USART to interact with an external device. The following figure adds a USART driver to the Reference Project with SERCOM1 as the underlying peripheral module.

Figure 2-26. Adding a Driver



- The application will use the following API to transfer data to the external device:

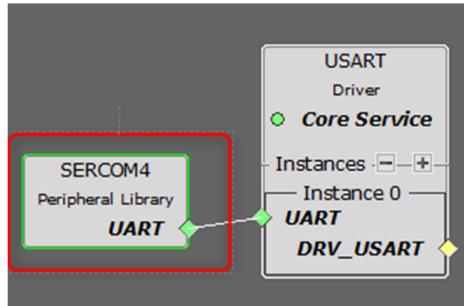
```
DRV_USART_WriteBufferAdd(appData.usartHandle, (void*)messageBuffer,
strlen(messageBuffer), &appData.bufferHandle);
```

- For Scenarios #2, #3, and #4:

- Be sure to import the required code generation tool (MCC or MHC) configurations from the existing MPLAB Harmony v3 application demonstration. Refer to [Tool Configurations](#) for additional information.

- Because drivers provide an abstracted interface to the MCU peripherals, the driver API calls used in the user application need not change when the project is ported from one platform to another. However, the underlying PLIBs on top of which drivers are built, must be migrated as described in the [PLIBs](#) and [PORT Pins](#).

Figure 2-27. Driver with a Different PLIB Block



The above change in the USART block (shown in [Figure 2-8 MHC Export](#)) in the project graph does not require any change in the application calling the USART driver APIs.

```
DRV_USART_WriteBufferAdd(appData.usartHandle, (void*)messageBuffer,
strlen(messageBuffer), &appData.bufferHandle);
```

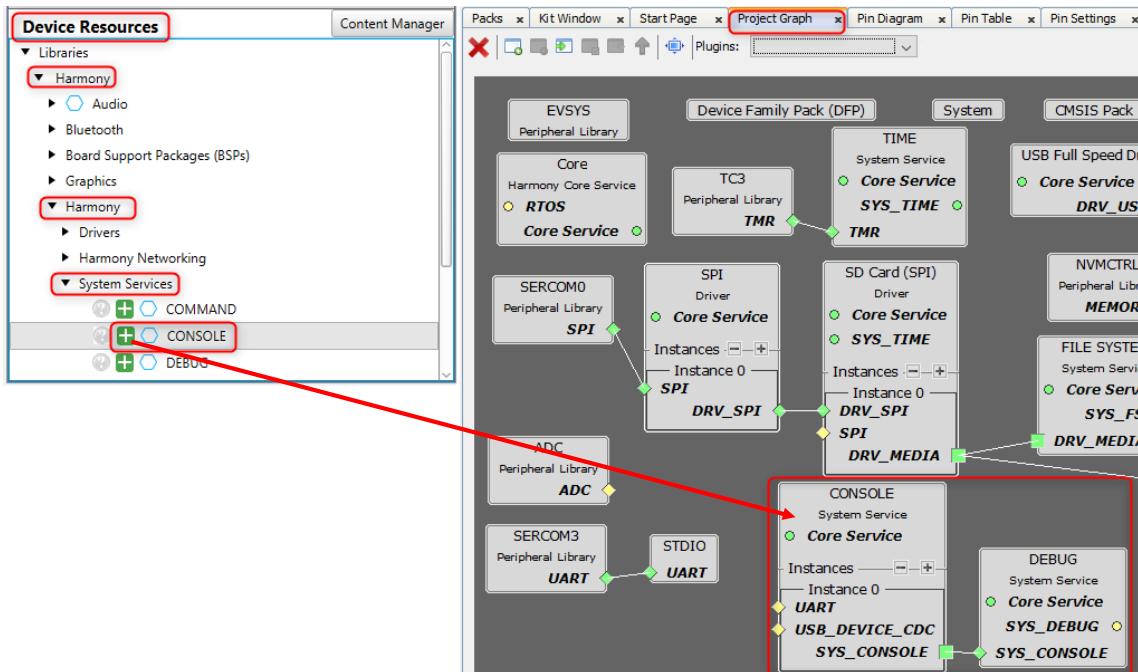
2.7 System Services

MPLAB Harmony v3 provides common functionality required by different drivers, middleware, or applications. It manages shared resources and eliminates potential resource conflict by keeping the underlying requests separate. [MPLAB Harmony v3 Architecture](#) shows the location of system services in the MPLAB Harmony architecture.

- For Scenario #1, there is no special consideration for adding, deleting, modifying, or retaining existing system services. Launch the code generation tool (MCC or MHC) project graph, perform any of these operations, and generate the code.

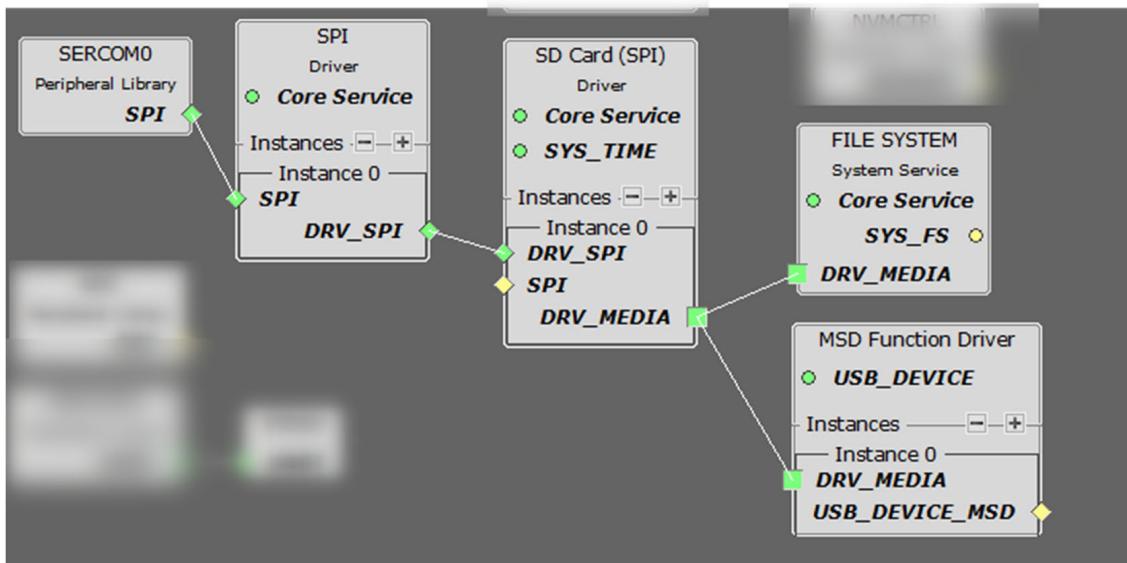
For example, if the Reference Project has an additional requirement to use a console system service, the following figure shows adding a console system service to the Reference Project.

Figure 2-28. Add Console System Service and Configure SYS_DEBUG



- For Scenarios #2, #3, and #4:
 - Be sure to import the required code generation tool (MCC or MHC) configurations from the existing MPLAB Harmony v3 application demonstration. Refer to the [Tool Configurations](#) for additional information.
 - Like drivers, *System Services* provide an abstracted interface to the shared resources. The system service API calls used in the user application need not change when the project is ported from one platform to another. However, the underlying resources (PLIBs, drivers) on top of which the system services are built must be migrated as discussed in [Drivers, PLIBs and PORT Pins](#).
- For example, In the Reference Project, the File System services uses the SDSPI driver, which uses the SPI driver, and which in turn uses the SERCOM0 (SPI) PLIB.

Figure 2-29. File System Service and Dependent Blocks



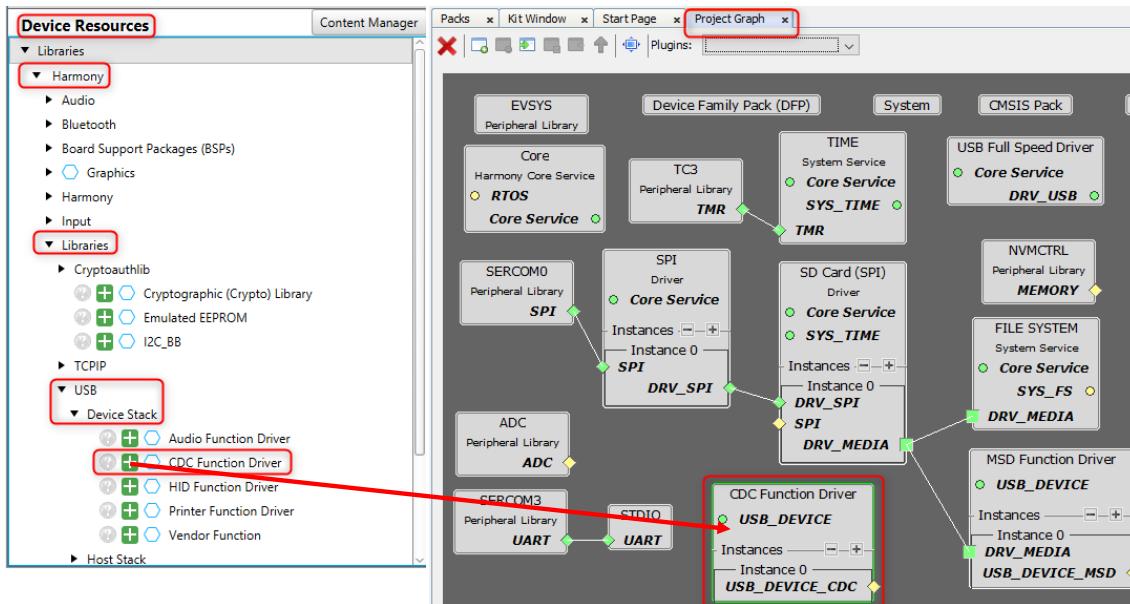
In the previous figure, a change to the SERCOM0 PLIB does not require a change in the application code, which uses the file system service APIs for sensor data, as the APIs provide an abstracted interface.

2.8 Middleware

MPLAB Harmony v3 provides middleware libraries for complex protocols and technologies, such as USB, TCP/IP, audio, and crypto. The middleware uses drivers and system services. [MPLAB Harmony v3 Architecture](#) shows the location of middleware in the Harmony architecture.

- For Scenario #1, there is no special consideration for adding, deleting, modifying, or retaining existing system services. Launch the code generation tool (MCC or MHC) project graph, and perform any of these operations and generate the code, for example, if the Reference Project has an additional requirement to use the USB CDC class. The following figure shows how to add a USB CDC class to the reference project.

Figure 2-30. Add USB CDC Class



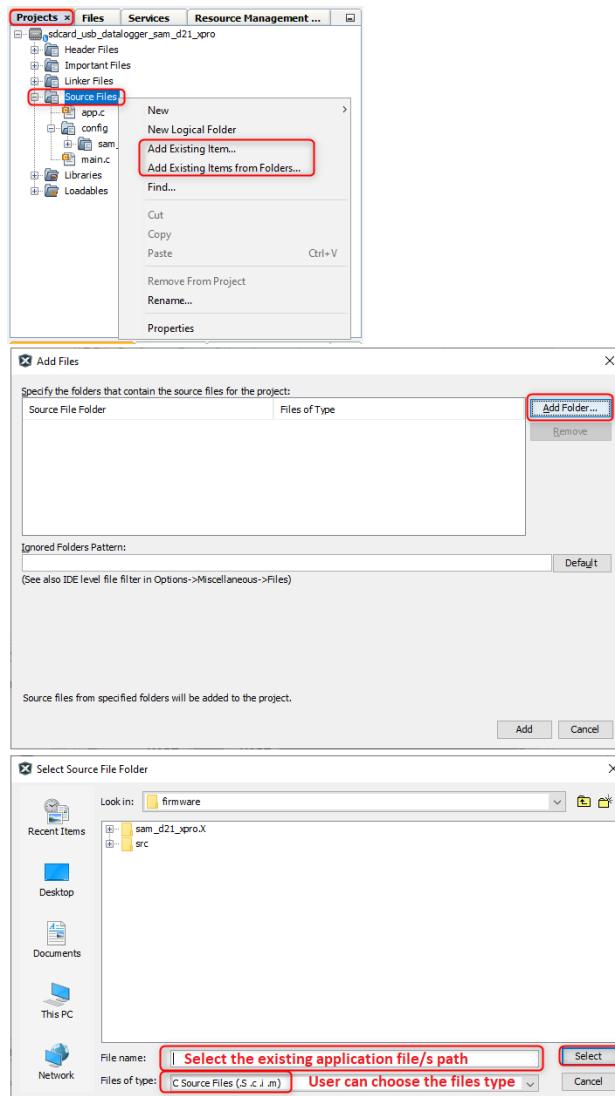
- For Scenarios #2, #3, and #4:
 - Be sure to import the required code generation tool (MCC or MHC) configurations from the existing MPLAB Harmony v3 application demonstration. Refer to the [Tool Configurations](#) for additional information.
 - As in drivers and system services, middleware also provides an abstracted interface. The middleware API calls used in the user application need not change when the project is ported from one platform to another. However, the underlying resources (PLIBs, drivers, system service) on top of which the system services are built must be migrated as discussed in [System Services](#), [Drivers](#), [PLIBs](#) and [PORT Pins](#).

2.9 Application

An MPLAB Harmony v3 application is generally developed in the `main.c` and `app.c` files. The application files can also have several other required source files, header files or library files.

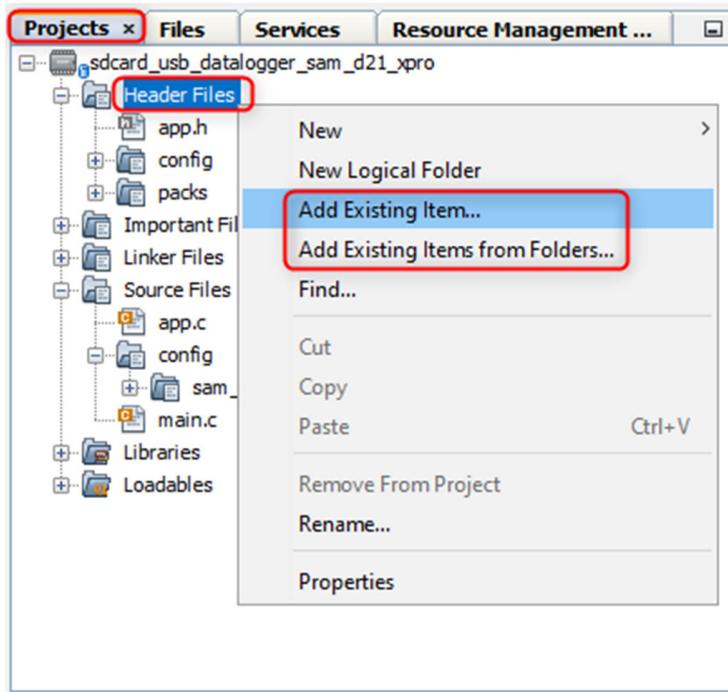
- For Scenario #1, there is no special consideration for adding, deleting, modifying, or retaining application files. The user can use the MPLAB X IDE feature shown in the following figure to add application files to the existing project.

Figure 2-31. Add Application Source Files



Similarly the user can also add application header files.

Figure 2-32. Add Application Header Files

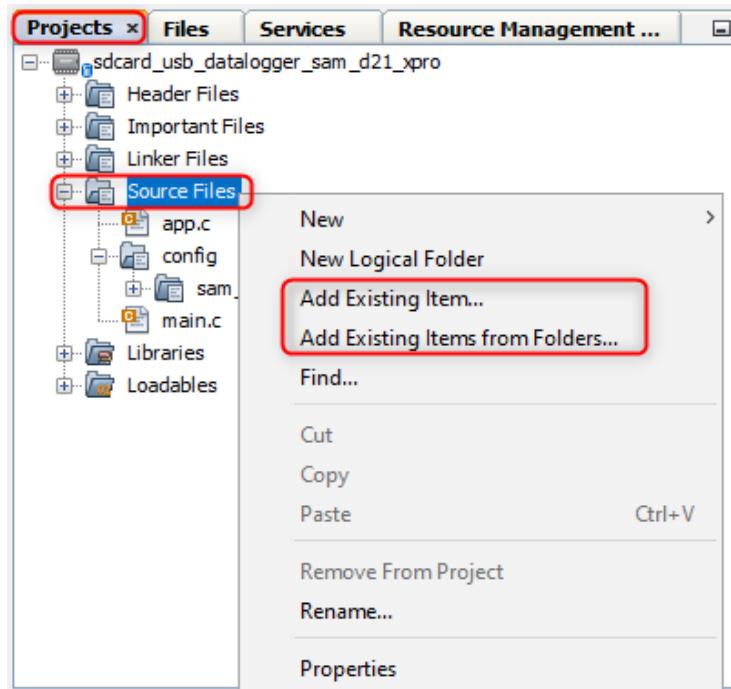


- For Scenarios #2, #3, and #4, use any one of the following methods to move the application files from the existing project to the new project.

- **Method 1. Adding Existing Item**

In this method, users can manually add user application files to the new project by selecting any one of the options highlighted in the following figure. This method is suitable if the number of application files added to the project is less than ten.

Figure 2-33. Add Application Files



- Method 2. Adding Existing Items from Folders

- This method is suitable when the number of application files added to the project is large (more than ten or multiple files in several directories and sub-directories).
For example, the application files from the reference project are required to be added into a new project having MPLAB X IDE project folder name as `sam_e51_cnano.X` on ATSAME51J20A MCU.
- Copy and paste all required application source (`.C`, `.CPP`, `.H`) files (with folder structure if any) from the reference project under the `src` folder to the `src` folder under the user created project.
- Copy and overwrite the `configurations.xml` file from the Reference Project MPLAB X IDE project folder `sam_d21_xpro.X\nbproject` to the user project folder `sam_e51_cnano.X\nbproject`.
- Open the overwritten `configuration.xml` file and replace all instances as shown below:
 - Change the existing project configuration name to a new project configuration name:
Replace `sam_d21_xpro` with `sam_e51_cnano`.
 - Change existing project MCU part number with the new project MCU part number:
Replace `SAMD21J18A` with `SAME51J20A`, and also replace `D21/d21` with `E51/e51`.

Notes:

- This change is not required if the MCU part number is same as the existing project and new project.
- Modifying the `configurations.xml` file can be tricky, care must be taken while replacing the part number to ensure the correct part numbers.

3. References

For additional information on Microchip products, availability, and MPLAB Harmony v3, visit the [Microchip Website](#), or contact a local sales representative.

- The MPLAB Harmony v3 Quick Docs repository provides standalone help pages for users to get started developing applications on Microchip's 32-bit SAM and PIC32 MCUs. Start from the `index.html` present in the `docs` folder.
The online version is available at https://microchip-mplab-harmony.github.io/quick_docs/
- Harmony landing page:
www.microchip.com/harmony
- Create a new MPLAB Harmony v3 project using MCC:
microchipdeveloper.com/harmony3:getting-started-training-module-using-mcc
- Update and Configure an Existing MHC-based MPLAB Harmony v3 Project to MCC-based Project:
microchipdeveloper.com/harmony3:update-and-configure-existing-mhc-proj-to-mcc-proj
- How to Add a New Configuration to an Existing MPLAB Harmony v3 Project:
www.microchip.com/DS90003304
- How to Build an Application by Adding a New PLIB, Driver, or Middleware to an Existing MPLAB Harmony v3 Project:
www.microchip.com/DS90003253
- MPLAB Harmony v3 Developer help page:
microchipdeveloper.com/harmony3:start
- Overview of MHC user interface in Import/Export feature:
microchip-mplab-harmony.github.io/mhc/doc/readme_mhc_import_export.html

The Microchip Website

Microchip provides online support via our website at 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 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: www.microchip.com/support

Microchip Devices Code Protection Feature

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

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, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzers, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, 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, TrueTime, 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, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Parallelizing, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, 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, Symmcom, and Trusted Time 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.

© 2022, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-0084-8

Quality Management System

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



MICROCHIP

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: 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: 45-4485-5910 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