

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

Intr	Introduction1					
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				
Pro	duct Change Notification Service	41				
Cus	stomer Support	41				
Mic	rochip Devices Code Protection Feature	41				
Leg	al Notice	41				
Tra	Trademarks					
Qua	Quality Management System					
Wo	Worldwide Sales and Service43					

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.

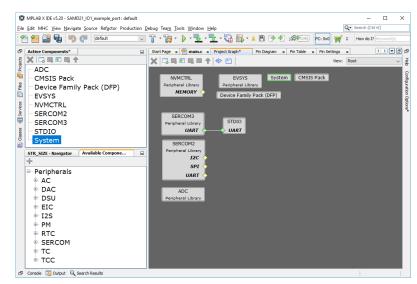


Figure 1-1. MPLAB Harmony 3 Configurator Available Components

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.

Steps	Choose Project	
 Choose Project 	Q, Fil <u>t</u> er:	
	Categories: Microchip Embedded G. Samples	Projects: Standalone Project Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project Import START MPLAB Project Import Atmel Studio Project
	Description: MPLAB® Harmony Project Wizard	

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

- 4. Click Next.
- 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

🔀 New Project		>			
Steps	Manage Framework				
Choose Project Framework Selection Project Settings Configuration Settings	Use the Framework Downloader tool to download or configure a local framework.				
	Framework Path:	C:\Users\\;;;;;;\HarmonyFramework			
		Convert to Relative Path for Configuration			
		< Eack Next > Finish Cancel Help			

- 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

eps	Name and Lo	ention
	Name and Lo	cation
Choose Project Framework Selection Project Settings	Location:	C:\Users\\:\:\:\HarmonyProjects\MyProject
Configuration Settings	Folder:	SAMD21_IO1_example_port
	Name:	SAMD21_IO1_example_port
	Path:	C:\Users\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
	Show	Visual Help
	Show	vision nep

- 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

🔀 New Project							
Steps	Configuration Sett	tings					
Choose Project Framework Selection Project Settings Configuration Settings	Name:	default					
		All v ATSAMD21	Target Device:	ATSAMD21J18A			~
	Show Visual	Help					
			< Bad	k Next >	Einish	Cancel	Help

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

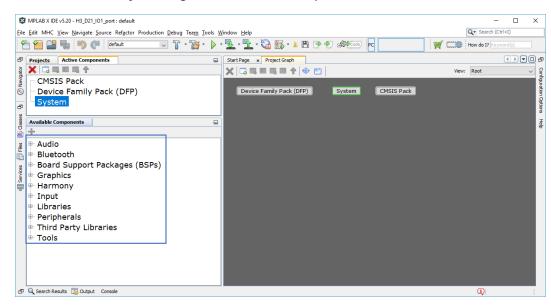
X MPLAB X IDE v5.20 - SAMD21_IO1_example_port : default		– 🗆 X
Eile Edit View Navigate Source Refactor Production Debug Team	Q Search (Ctrl+I)	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	🦉 • 🕨 • 🔽 • 🔽 • 🎧 🚮 • PC: 0x0 🛒 🕮 How do 17 (ceyword(s)	
Projects × liles Services Classes Image: Services Classes Image: Services Classes Image: Services Image: Services Image: Services Image: Services		
main() - Navigator ×		
	Search Results Output - Configuration Loading Error ×	
<no available="" view=""></no>	warning: Configuration "default" builds with "XC32", but indicates no toolchain direct info: Configuration "default" will build with toolchain "XC32" at "C:\Program Files (x	pry. ∧ € \Microchip\xc32\v2.20-
	<	>

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

8	MPLAB X IDE v5.20 - SAMD21_IO1_example_port : d	efault						-		×
File	e Edit View Navigate Source Refactor Productio	n Deb	ug Team	Тоо	ls Window Help		Q,	Search (Ctrl	+I)	
1	『 🎦 🔚 🖣 👘 🕐 🖉 🕴 default		~ 7		Embedded	MPLAB® Harmony 3 Framework Downlo	ader	ord(s)		
해 Services 🗍 Files 🕲 classes 日 🖉 Navigator 日 🦰	Image: Solution of the second seco	Sour 1 2 3 4 5 6 7 8 9	ain.c x 1 ce Histor - /**** Sy Fi Su	2	Embedded Licenses Packs Apply Diff Patch Diff Add to Favorites Templates DTDs and XML Schemas Plugins Plugins Download Options Toto Loading Error ×	MPLAB® Harmony 3 Framework Downlo MPLAB® Harmony 3 Configurator Ton.n x M definitions.n x M inbalization.c Ton.n x M definitions.n x M inbalization.c anction which overrides the define interrupt.c file.		plib_port.	except	* * -
Ð	Q Search Results						(1)	1:1	INS

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





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.

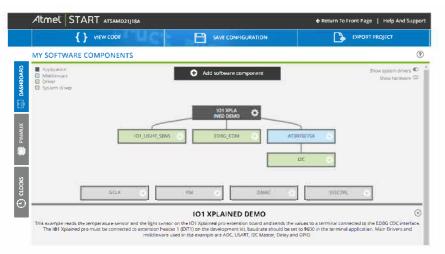
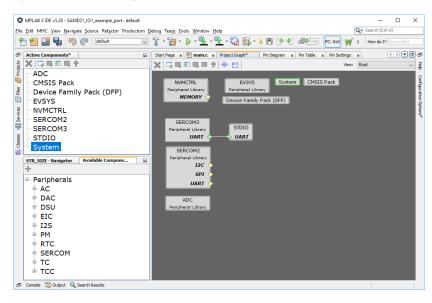


Figure 2-1. Atmel START Graphical Configuration Interface

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*.

E73			Chandred Marda		1 1 100 1 1		_ D >
IO1Xplaineddemo - AtmelStudio			Standard Mode	Quick	Launch (Ctri+	2) 2	- • ×
	ect Build Debug Tools Window	Help					
	Re-Configure Atmel Start Project		bug Browser 👻			€ 2 = 18 °2	■ 31 31 ±
∭ M Ď = → II ▶ ↔ 💐	Update Atmel Start Project from Start file		, 👗 📥 🔛 🖕 📟	ATSAMD21J	18A 🥤 None	on 🚽	
at30tse75x.c temperature	Show All Files		nel START 🛛 < 🗸	Solution E	xplorer		
→ main.c	Set as StartUp Project	p	aineddemo\IO 👻 ኛ Go	001		💿 🖌 🗕 🤇	8 D
* *	IO1Xplaineddemo Properties	Alt+F7	÷		lution Explorer		د
[*/			-			eddemo' (1 proje	
<pre>#include "atmel start.h"</pre>					01Xplainedder		.0 -
<pre>#include "atmel_start_pin</pre>					Dependencie		
<pre>#include "io1_xplained_de</pre>					Output Files		
<pre>#include "temperature_se</pre>	nsor_main.h"				Libraries		
<pre>#include <hal delay.h=""></hal></pre>				Þ 🗅	Config		
<pre>#include <nal_delay.n <br="">#include <stdio.h></stdio.h></nal_delay.n></pre>				🖌 🖌 🗋	Device_Start	qu	
<pre>#include <string.h></string.h></pre>					🗋 samd21j1	8a_flash.ld	
					🗋 samd21j1	8a_sram.ld	
#define STR_SIZE 50			-		c startup_s		
uint16 t	temp result:				system_s		
static struct io descrip					documentati	on	
	= *				examples		
<pre>void UART_EDBG_init()</pre>					a hal hpl		
{	<pre>scriptor(&EDBG COM, &terminal io);</pre>				hri		
usart sync enable(&E					temperature	sensor	
}					atmel start.c		
				6	atmel_start.h		
⊡int main(void)			Ψ		atmel_start_p	oins.h	
100 % - 4		_		c	driver init.c		
Error List							- ₽ >
Entire Solution 🔹 😣 0 En	rrors 📗 🔔 0 Warnings 📔 🚺 0 Messages 📗	Build + IntelliSer	ise 👻		Se	arch Error List	Q
Description					Project	File	Line
Error List Find Results 1							
Output							
Ready			Ln 52	Co	ol 2	Ch 2	INS

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/.

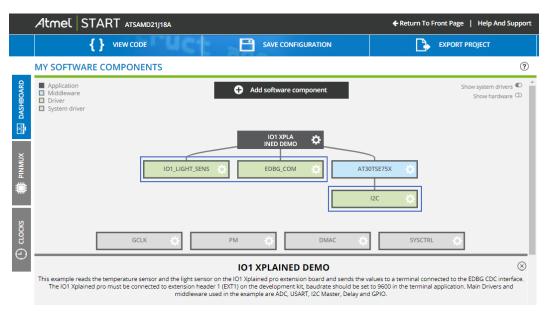
Atmel START main window showing Load Project tab is displayed.

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

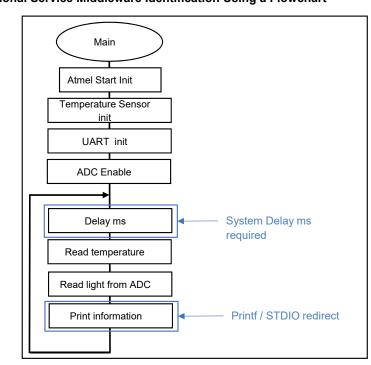
М іскоснір	
	What's New Help
Atmel START	
This tool will help you select and configure software components and tailor your embedded application in a usable and optimized manner.	
CREATE NEW PROJECT	
Getting started	
To get started you can either create a new project from scratch or open an exis example. In both cases you can configure your software components and devi such as clocks and pin layout. When you are done, you can export your project it using your favourite IDE for further development.	ce settings
For more information on how to use Atmel START, read the <u>Getting Started gu</u> watch our <u>video tutorials</u> .	de.or
Load exist	ing project
LOAD PROJECT FROM FILE	RESUME AUTO SAVED PROJECT
Use this option if you want to restore a locally saved project. Browse and select either a project file (*.atzip) or a configuration file (*.atstart.*.json).	Your latest project will always be stored in your web browser. Use this option to pick up where you left off.

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





 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*.

_						
×	MPLAB X IDE v5.20 - SAMD21_IO1_example_port : d	efault			- 0	×
File	Edit View Navigate Source Refactor Productio	n Debug	Team	m Tools Window Help	Q- Search (Ctrl+I)	
4	🗅 🎦 🔛 🐚 🍘 default	~	T	Embedded	MPLAB® Harmony 3 Framework Downloader	
-		(m) .		Licenses	MPLAB® Harmony 3 Configurator	
Ð	Projects ×	🖭 main.	c ×	Packs	non.n x 🗠 definitions.n x 🕾 initialization.c x 🐑 plb_port 💽 🕻	<u> </u>
Navigator		Source	Histo	tor Apply Diff Patch	🗸 🖶 斗 🔗 😓 엘 앨 🥥 🔲 🕮 🚅 📴	
Vavi		1 🖓	/**	** Diff	*******	^ =
8		2	S	Sy Add to Favorites		-+-
5		4	F	Fi Templates		
_		5		DTDs and XML Schemas		_
Classes		6				
		7	S	Su Plugins	unction which overrides the default weak excepti	
_		9		Plugins Download	he interrupt.c file.	~
Files			<	Options	>	
D		>			<u> </u>	×
Services		Output	- Conf	nfiguration Loading Error ×		-
Ser						^
뮮						
						¥
Ð	🔍 Search Results				1:1	INS

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

🚑 MPLAB Harmony Configurator - default*			-		×
File Generate Tools Utilities Window					
📜 🖹 🎐 🌪 🕼 🖓 Code			Framework: C:\Users\M43472\Ha	rmonyFra	mework\
🗋 Available Components 💦 🗖 🗖	Project Graph	- 2 🗆	Configuration Options	- 1	90
+	X 389884 + + =	View: Root 🗸			
Board Support Packages (Peripherals AC ADC	NVMCTRL EVSYS Peripheral Library MEMORY	Device Family Pack (DFP) System	System		
ADC ADC DAC DSU		CMSIS Pack	Enable SysTick Ports Clock		
EIC			⊕-Interrupts (NVIC) ⊕-DMA (DMAC)		
			⊕-WDT		
₽ PM			B-PAC		
I RTC					
SERCOM					
SERCOM0					
- SERCOM1					
SERCOM2					
SERCOM3					
TC					
I TCC					
i⊨- Tools					
- STDIO					
Available Components *1			Configuration Options Help		
Console -a 🗗					
Welcome to the MPLAB Harmony Configurator!					

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

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

Table 2-1. Drivers and Middleware

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

HAL:DRIVER:I2C MASTER SYNC BASIC										
I2C Bus clock speed (Hz):	O 100000	1	and a second	Configur	ation Options					- 🗗 🗖
ADVANCED				- +						
	0.00			B-SERCC)M2					
TRise (ns):	0 <u>60</u> det v		1	🚊 Sele	ect SERCOM O	peration Mode	I2C M	aster	~	
Master SCL Low Extended Time-Out (MEXTTOEN):	•				Transfer Spee	d Mode		STANDAR	D_AND_FAS	ST_MODE ~
Slave SCL Low Extend Time-Out	0		╽╓┿		Enable operat	ion in Standby	mode			
(SEXTTOEN):					SDA Hold Time			300-600ns	hold time	\sim
SCL Low Time-Out (LOWTOUT):	0		IIL		I2C Speed in k	Hz			100 🌻	
Inactive Time-Out (INACTOUT):	Disabled	Υ.		L.	I2C Trise in na	no seconds			60 🌲	
SDA Hold Time (SDAHOLD):	300-600ns hold time	× —	_							
Run in stand-by:	0			0						
Debug Stop Mode:	Keep running	× .		Configu	aration Options (Help				

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

🗋 Available Components 🛛 🗖 🗖	Project Graph	- & D
+	🗙 🗔 🛤 🗉 🖷 🕈 🏟 🖻 🕇	View: Root 🗸
 Audio Bluetooth Board Support Packages (BSPs) Graphics Harmony Input Libraries Peripherals Third Party Libraries Tools 	NVMCTRL Peripheral Library MEMORY ADC Peripheral Library Peripheral Library IZC SPI UART	Device Family Pack (DFP) System CMSIS Pack SERCOM2 Peripheral Library I2C SPI UART
Available Components		

 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.

Atmel START	ATSAMD21J18A		← Return To Front Page Help And Supp
{ } VIE	W CODE		EXPORT PROJECT
MY SOFTWARE COM	PONENTS		
Application Middleware Driver System driver	I	Add software component	Show system drivers C Show hardware
	101_LIGHT_SENS		130T5E75X 0
	PM ().	SYSETTL GELK	DMAC O
	Universal Asynchronous re	EDBG_COM eceiver/transmitter (UART) communication in synchror	nous/blocking mode
GENERAL	COMPON	ENT SETTINGS	COMPONENT SIGNALS
i User guide	Driver:	HAL:Driver:USART_Sync ~	RIC PA23 V
🖉 Rename compone	Mode:	LIART ~	TX: PA22 ~
	Instance:	SERCOMB V	
Remove component	nt CLOCKS		
	Core:	Generic clock generator 1 (200 kHz)	
	Slow:	Generic clock generator 0 (1 MHz)	
HAL:DRIVER:USART SYNC	(UART) CONFIGURATION ON S	ERCOM3	c
BASIC CONFIGURATION		ADVANCED CONFIGURATION	N Enable: 🖵
Receive buffer enable:	•	Run in stand-by:	•
Transmitt buffer enable:	•	Immediate Buffer Overflow Notification:	•
Frame parity:	No parity	Start of Frame Detection Ena	able: O
Character Size: Stop Bit:	8 bits	Collision Detection Enable:	•
1 A A A A A A A A A A A A A A A A A A A	One stop bit	Operating Mode:	USART with internal clock Y
Baud rate:	9600	dec v Sample Rate:	16x arithmetic ~
		Sample Adjustment:	7.8-9 (3-4-5 8-bit over-sampling)
		Fractional Part:	O dec
		Data Order:	LSB is transmitted first ~
		Encoding Format:	No encoding ~
		Encoding Format: LIN Slave Enable: Debug Stop Mode:	No encoding V Disable Keep running V

Figure 3-3. Access Driver Configuration in Atmel START

- ADC peripheral Library (IO1_LIGHT_SENS) configuration.

Figure 3-4. Port ADC Driver Configuration

HAL:DRIVER:ADC SYNC	C (ADC) CONFIGURATION ON ADC	C
BASIC CONFIGURATION	ADVANCED CONFIGURATION	Enable: 🗸
Conversion resolution:	Ø 12-bit ✓ Run in standby: Ø	
Reference Selection:	② 1.0V voltage reference ∨ Debug Run: ②	
Prescaler configuration:	Peripheral clock divided by 4 Left-Adjusted Result:	
Free Running Mode:	Reference Buffer Offset O Compensation Enable:	
Differential Mode:	Digital Correction Logic Enabled:	
Positive Mux Input Selectio	on: Offset Correction Value: Offset Correction Value:	dec v
Negative Mux Input Selecti	ion: Internal ground Gain Correction Value: 0	dec v
EVENT CONTROL	Enable: Gain Factor Selection: 0 1x	~
	Adjusting Result / Division Coefficient: 0	dec v
	Number of Samples to be Collected: 🤨 1 sample	~
	Sampling Time Length: 0	dec v
	Window Monitor Mode: No window mode	~
	Window Monitor Lower Threshold: 0	dec v
	Window Monitor Upper Threshold: 0	dec v
	Number of Input Channels Included 0	dec v
	in Scan: Positive Mux Setting Offset: 0	dec v
	Configuration Options	
	B-ADC	
	-Select Prescaler Peripheral dock divided by 4 v	
	Select Sample Length (cycles) 4	
	-**** Conversion Time is 64.0 uS ****	
	–Select Gain 1x V	
	-Select Reference 1/2 VDDANA (only for VDDANA > 2.0V) ~	
	-Select Conversion Trigger V	
	E-Channel Configuration	
	Select Positive Input ADC AINS Pin V	
	Select Negative Input Internal Ground V	
	Number of inputs to scan	
	-Select Result Resolution 12-bit result	
	Left Aligned Result	
	Enable Result Ready Interrupt	
	Enable Result Ready Event Out	
	Window Mode Configuration	
	Select Window Monitor Mode No window mode (default)	
	E-Sleep Mode Configuration	
	Run During Standby	
	Configuration Options Help	

- SERCOM3 peripheral Library (EDBG_COM) configuration

HAL:DRIVER:USART SYNC (UART) CONFIGURATION ON	SERCOM3					C
BASIC CONFIGURATION				ADVANCED CONFIGURATION			Enable: 🗸
Receive buffer enable: Transmitt buffer enable: Frame parity: Character Size: Stop Bit: Baud rate:	 No parity No parity 8 bits One stop bit 9600 		→ → ↓ dec v	Run in stand-by: Immediate Buffer Overflow Notification: Start of Frame Detection Enable: Collision Detection Enable: Operating Mode: Sample Rate: Sample Adjustment: Fractional Part: Data Order:	0	USART with internal clock USART with internal clock 16x arithmetic 7-8-9 (3-4-5 8-bit over-sampling 0 LSB is transmitted first	~
				Encoding Format:		No encoding	~
				LIN Slave Enable:	0	Disable	~
				Debug Stop Mode:	0	Keep running	~
			~			- 80	
	ct SERCOM Operation	Mode USA	ART wi	ith internal Clock 🗸			
	nable Interrupts ?						
	eceive Enable ransmit Enable						
	nable Run in Standby						
	eceive Pinout		\D[1] i:	s used for data reception \smallsetminus			
T	ransmit Pinout			s used for data transmission 🗸	,		
P	arity Mode	No Parity	\sim				
	haracter Size	8 Bits 🗸					
	top Bit Mode	One Stop Bi	t v				

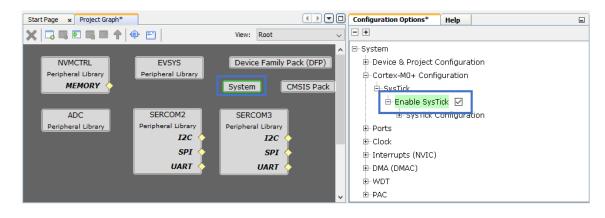
- SERCOM2 peripheral Library (I²C) configuration

BASIC				ADVANCED		Enable: 🗸
2C Bus clock speed (Hz):		0 100000	decv	TRise (ns):	60	dec
				Master SCL Low Extended Time-Out (MEXTTOEN):	0	
				Slave SCL Low Extend Time-Out (SEXTTOEN):	0	
				SCL Low Time-Out (LOWTOUT):	0	
				Inactive Time-Out (INACTOUT):	Oisabled	\
				SDA Hold Time (SDAHOLD):	300-600ns hold time	· · · · · · · · · · · · · · · · · · ·
				Run in stand-by:	0	
			_	Debug Stop Mode:	Keep running	
_				Ļ		
6] Config	guration Options			_	2 🗖
E	• •					
E	SERC	COM2				
	Ė~S	elect SERCOM Operation Mod	de I	2C Master 🗸 🗸		
		Transfer Speed Mode		STANDARD_AND_FAST	_MODE <>	
		Enable operation in Standł	oy ma	ode 🗌		
		SDA Hold Time		300-600ns hold time $\!$		
				100 ≑	-	
				60 🜩		

Figure 3-6. Port SERCOM2 Driver Configuration

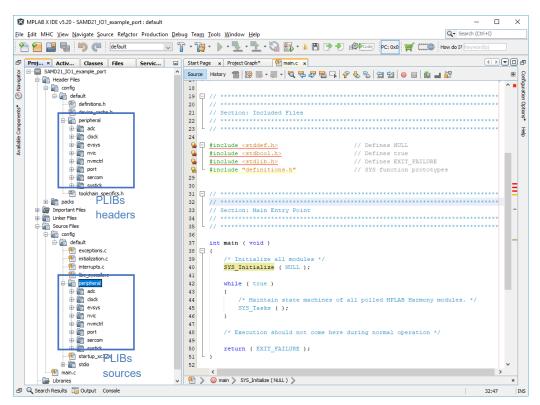
- SYSTICK Service configuration

Figure 3-7. Enable Service Under MHC



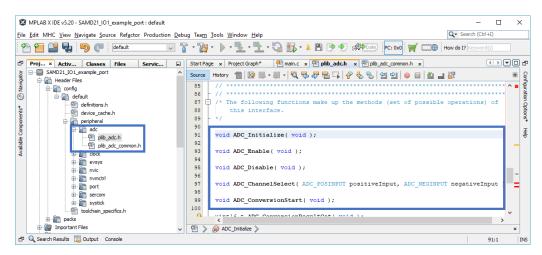
After generating the project source code from the MHC interface by clicking or selecting *MHC* > *Generate Code*, the peripheral libraries (PLIBs) configuration can be found in the following project directories:





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

Figure 3-9. Peripheral Libraries API Location

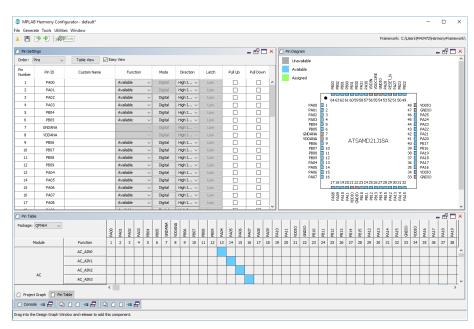


4. Porting the System PINMUX Configuration from Atmel START

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.

Atmel START ATSAMD21J18A ← Return To Front Page | Help And Support SAVE CONFIGURATION EXPORT PROJECT { } VIEW CODE PINMUX CONFIGURATOR ? Pin Show labels... v (+) Zoom in (-) Zoom out Auto fit | v Board Mode SW com Pad Us. Header Label 23 PB10 EXT3.EXT2 USART_T. PINMU 24 PB11 EXT3.EXT2 USART 25 PB12 EXT2 PWM(+) 26 PB13 EXT2 PWM(-) 27 EXT2 IRQ/GPIO PB14 EXT2 28 PB15 SPI_SS_B. 29 PA12 EXT3 PWM(+) PA13 EXT3,Ser. PWM(-),S. VBUS_D. 31 PA14 USB PA15 EXT3,Mis 32 GPIO2,S. 33 GNDIC 3/1 VDDIC 35 PA16 EXT2 SPI MISO User label EDBG_COM_TX Pin 43 (PA22) is used as TX with EDBG_COM. Pin mode Peripheral IO Tip: Use ctrl or shift to select more than one © 2019 Microchin Tech

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



Follow these steps for the SAM D21 IO1 Xplained demonstration:

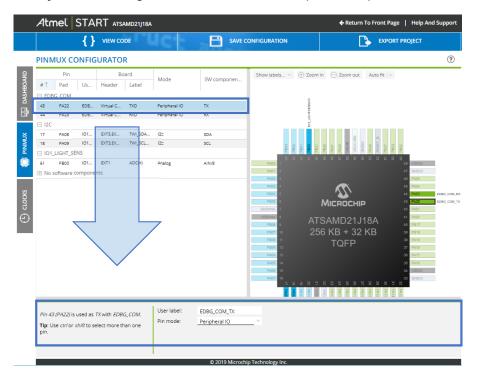
1. In Atmel START, click on the **PINMUX** tab and identify which I/O pins are configured and used in the application.

		{}	VIEW CO	DE			VE CONFIGURATION	EXPORT PROJECT
PINN	IUX CO	ONFIG	URATO	R				(1
	Pin		Bo	bard			Show labels 🗸 🕀 Zoom in 🕞	Zoom out 🛛 Auto fit 🗸
#↑	Pad	Us	Header	Label	Mode	SW compor	n	
🗆 EDB	G_COM			1				
43	PA22	EDB	Virtual C	TXD	Peripheral IO	ΤХ	END SO STATE	
44	PA23	EDB	Virtual C	RXD	Peripheral IO	RX	Deskiast	
⊟ I2C							ē	
17	PA08	101	EXT3,EX	TWI_SDA	I2C	SDA		a z
18	PA09	101	EXT3,EX	TWI_SCL	I2C	SCL	833 M31	A30 DDIN A28 B23 B22 B22
	LIGHT_SE							A A B A A A
61	PB00 software c	101	EXT1	ADC(+)	Analog	AIN/8	PA00 1 PA01 2	48 VDDIO 47 GNDIO
							GNDAVA 7 VDDAVA 6 PB00 9 PB07 10 256 KI	MD21JJ18A 41 M20 42 M21 42 M22 42 M21 42 42 42 42 42 42 43 43 43 43 43 43 43 44 14 <th14< th=""> 14 14</th14<>

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

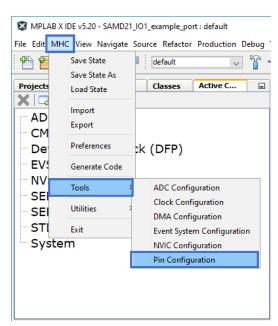
Note: The configuration of each pin is available by clicking on it in the PINMUX configurator table.

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



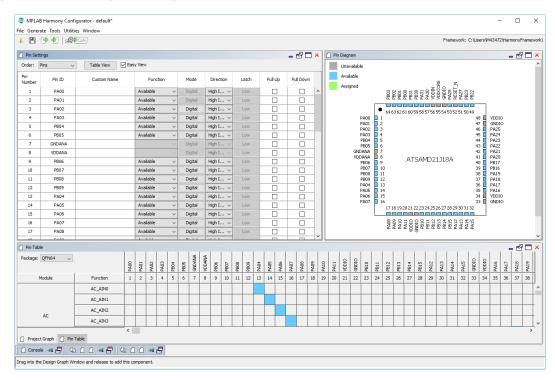
2. In MPLAB X IDE, select *MHC* > *Tools* > *Pin Configuration* to open the project pin configurator interface.





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

	<i>08)</i> is used as <i>trl</i> or <i>shift</i> to se	<i>SDA</i> with <i>I2C</i> . elect more than one	User label: Pin mode:	<u>101_T</u> 12C	<u>WI_SDA</u>	I	Pull configu	ration: Off	f v
	<i>09)</i> is used as . trl or <i>shift</i> to se	SCL with I2C. elect more than one	User label: Pin mode:	IO1_TV	VI_SCL		Pull configu	ration: Off	f v
Start Pag Order:	e x 🖭 main./ Pins v		Pin Diagram ×	Pin Table	Pin Settings	×			<u>ב</u>
Pin Number	Pin ID	Custom Name	Function	Mo	le Direction	Latch	Pull Up	Pull Down	
10	PB07		Available	 ✓ Dig 	tal High 🔻	Low			A. C.
11	PB08		Available	√ Digi	tal High 🔻	Low			
12	PB09		Available	~ Dig	tal High 🔻	Low			
13	PA04		Available	 ✓ Digi 	tal High 🔻	Low			
14	PA05		Available	∼ Digi	tal High 🔻	Low			
15	PA06		Available	 ✓ Digi 	tal High 🔻	Low			
16	PA07		Available	∼ Dig	tal High 🔻	Low			
17	PA08	SERCOM2_PAD0	SERCOM2_P	∼ Digi	tal High 🔻	/ n/a			
18	PA09	SERCOM2_PAD1	SERCOM2_P	 ✓ Digi 	tal High 🚿	/ n/a			
	PA 10		Available	∼ Dig	tal High 🔻	Low			
19			Available	 ✓ Digi 	tal High 🔻	Low			
19 20	PA11								
	PA11 VDDIO			 ✓ Digi 	tal High 🔻	Low			
20				DigitizedDigitized		_			
20 21	VDDIO		Available		tal High \	Low			
20 21 22	VDDIO GNDIO		Available Available	√ Digi	tal High \	Low Low			

- SERCOM 3 I/Os settings

		ed as <i>TX</i> with <i>EDB</i> (ft to select more th	_	User lal Pin moo			_COM_TX leral IO	;	~
		sed as <i>RX</i> with <i>EDB</i> ift to select more th	-	User lal		EDBG_ Periph	COM_RX eral IO		~
Start Page Order: F			Pin Diagram × P sy View	in Table x	Pin Settings	×) 🗆
Pin Number	Pin ID	Custom Name	Function	Møde	Direction	Latch	Pull Up	Pull Down	
37	PA18		Available	Digital	High $\dots \sim$	Low			^
38	PA19		Available	 Digital 	High $\dots \lor$	Low			
39	PB16		Available	 Digital 	High $\dots \lor$	Low			
40	PB17		Available	 Digital 	High $\dots \lor$	Low			
41	PA20			 Digital 	High 🗸	Low			
42	PA21			 Digital 	High $\dots \lor$	Low			
43	PA22	SERCOM3_PAD0		 Digital 	High $\dots \lor$	n/a			
44	PA23	SERCOM3_PAD1		 Digital 	High \dots \vee	n/a			
45	PA24			 Digital 	High ∨	Low			
46	PA25		Available	 Digital 	High ∨	Low			
47	GNDIO		· · · · · · · · · · · · · · · · · · ·	 Digital 	High $\dots \lor$	Low			
48	VDDIO		``````````````````````````````````````		High $\dots \lor$	Low			
49	PB22		Available	Bigitai	High $\dots \lor$	Low			
50	PB23			 Digital 	High ∨	Low			
51	PA27		Available	 Digital 	High ∨	Low			
52	RESET_N		· · · · · · · · · · · · · · · · · · ·	 Digital 	High \dots \vee	Low			\checkmark

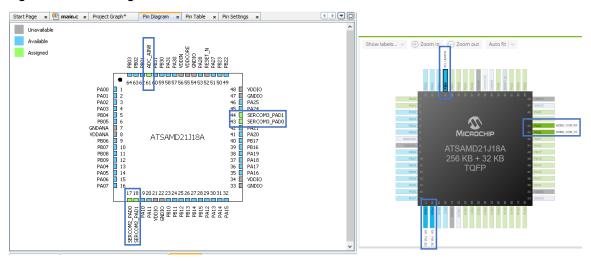
Figure 4-7. SERCOM 3 I/O settings

- ADC I/Os settings

Figure 4-8. ADC I/O Settings

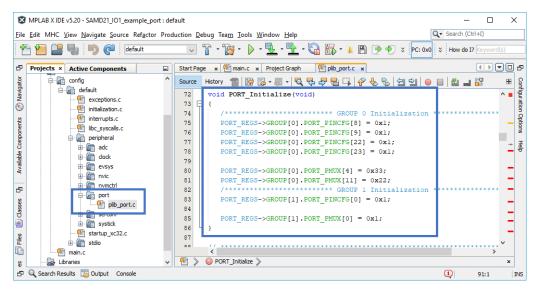
	(<i>PB00)</i> is us IGHT_SENS.	ed as <i>AIN/8</i> with			ser labe n mode	L	IO1_LIGH Analog	TSENSOR	×	
Start Page				× Pin 1	Table ×	Pin Settings	×			
Order: I Pin Number	Pins 🗸	Table View Ea	rsy View Function		Mode	Direction	Latch	Pull Up	Pull Down	
49	PB22		Available	~	Digital	High	✓ Low			^
50	PB23		Available	~	Digital	High	✓ Low			
51	PA27		Available	~	Digital	High	✓ Low			
52	RESET_N			\sim	Digital	High	 Low 			
53	PA28		Available	~	Digital	High	✓ Low			
54	GNDIO			\sim	Digital	High	 Low 			
55	VDDCORE			\sim	Digital	High	Low			
56	VDDIN			\sim	Digital	High	 Low 			
57	PA30		Available	~	Digital	High	✓ Low			
58	PA31		Available	~	Digital	High	✓ Low			
59	PB30		Available	~	Digital	High	✓ Low			
60	PB31		Available	~	Digital	High	✓ Low			
61	PB00	ADC_AIN8	ADC_AIN8	~	Analog	High	√ n/a			
62	PBUI		Available	~	Digital	High	LOW			
63	PB02		Available	~	Digital	High	✓ Low			
64	PB03		Available	~	Digital	High	Low			\mathbf{v}

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.

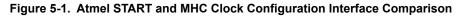


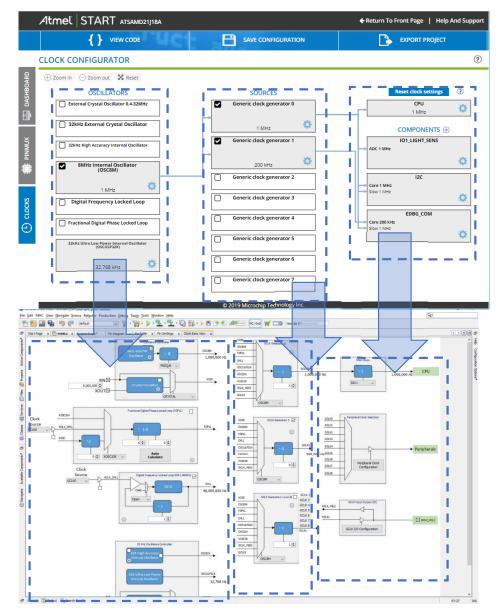


Note: During application run-time, the PORT configuration can be changed by calling APIs from the MPLAB Harmony 3 PORT PLIB.

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.





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

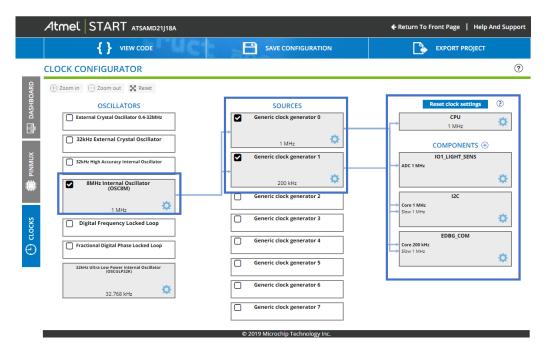


Figure 5-2. Identify Project Clock Configuration Under Atmel START

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

	CLOCK SETTINGS ×
8MHz Internal Oscillator (OSC8M) 1 MHz	8MHZ INTERNAL OSCILLATOR CONFIGURATION C 8MHZ INTERNAL OSCILLATOR (OSC8M) CONTROL Internal 8M Oscillator Enable: Image: Control Contr
	Close

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

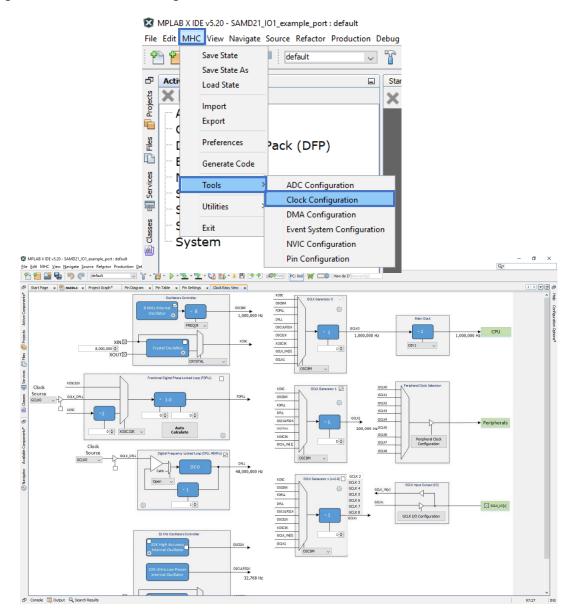
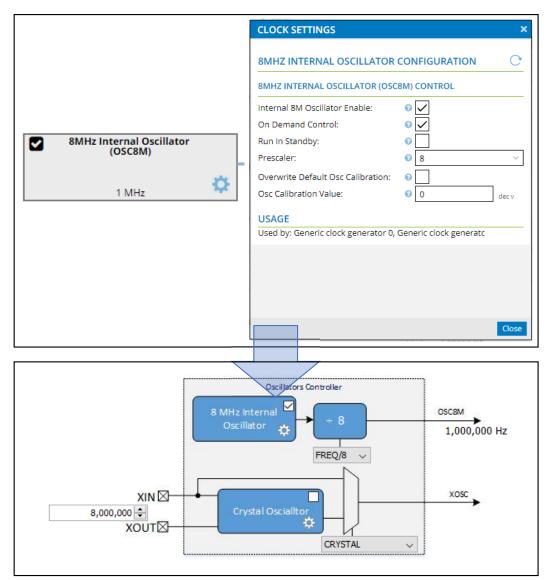


Figure 5-4. Access Clock Configuration Interface Under MHC

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





- GCLK Generator 0

Generic clock generator 0	CPU 1 MHz
CLOCK SETTINGS ×	CLOCK SETTINGS
GENERIC CLOCK GENERATOR 0 CONFIGURATION	SYSTEM CONFIGURATION
GENERIC CLOCK GENERATOR CONTROL	CPU CLOCK SETTINGS
Run in Standby: Divide Selection: Output Enable: O	CPU Clock source: Generic clock generator 0 CPU clock Prescalar: CPU clock Prescalar:
Output Off Value: 0 Improve Duty Cycle: 0 Generic Clock Generator Enable: 0	NVM SETTINGS NVM Wait States: APBA CLOCK SELECT
Generic clock generator 0 source:	APBA clock prescalar: APBA clock SELECT
Generic clock generator 0 division: 0 1 dec v	APBB clock prescalar: 0 1
Used by: CPU, IO1_LIGHT_SENS, I2C, EDBG_COM	APBC CLOCK SELECT APBC clock prescalar:
XOSC OSC8M FDPLL DRL OSCULP32K OSC32K XOSC3K GCLK_Generator 0 GCLK0 GC	Main Clock +1 DIV1 v CPU CPU

Figure 5-6. GCLK Generator 0 Configuration Port

- GCLK Generator 1

Figure 5-7. GCLK Generator 1 Configuration Port

	Generic clock generator 1
	200 kHz
GENERIC CLOCK GEN	NERATOR 1 CONFIGURATION
Run in Standby: Divide Selection: Output Enable: Output Off Value: Improve Duty Cycle: Generic Clock Generator Generic clock generator GENERIC CLOCK GENER Generic clock generator	1 source: 1 source: BMHz Internal Oscillator (OSC8M) ATOR DIVISION
USAGE Used by: EDBG_COM	Close
XOSC OSC8M FDPLL DFLL OSCULP32K OSC32K XOSC3K GCLK_IN[1]	GCLK Generator 1 GCLK Generator 1 GCLK1 COSC8M

- ADC Peripheral clock

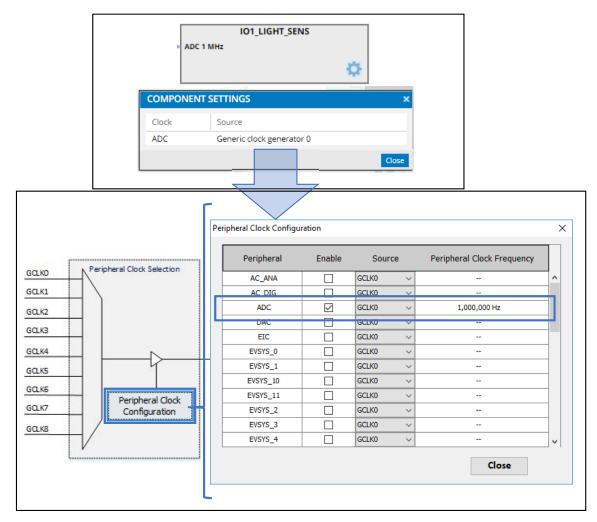


Figure 5-8. ADC Peripheral Clock Configuration Port

- SERCOM2 Peripheral clock

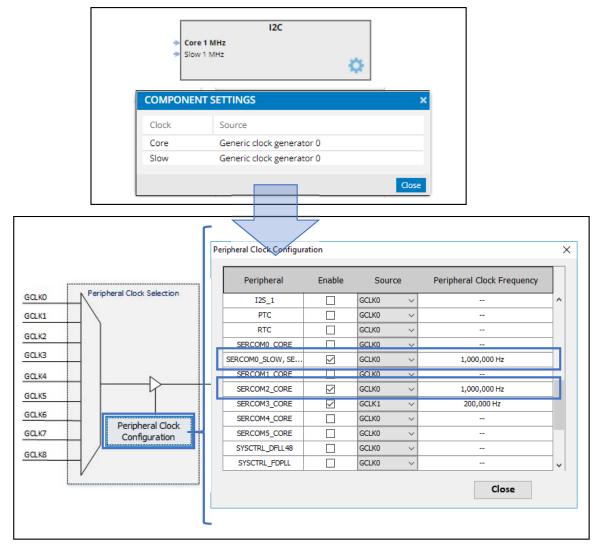


Figure 5-9. SERCOM2 Peripheral Configuration Port

- SERCOM3 Peripheral clocks

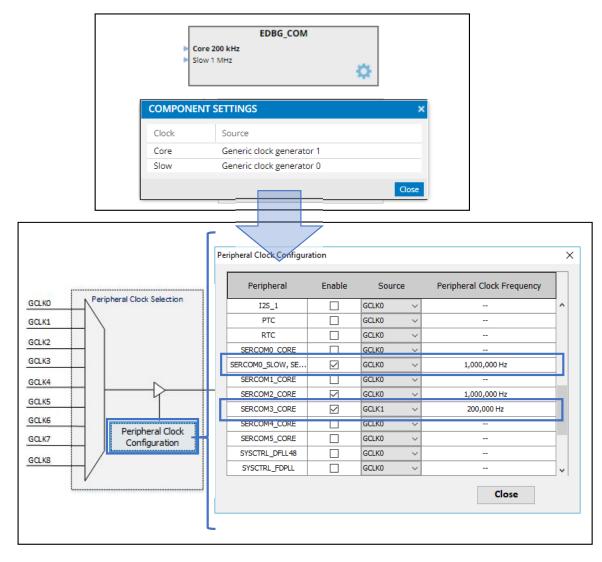


Figure 5-10. SERCOM3 Peripheral Configuration Port

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

MPLAB Harmony Configurator - default*		-		×
File Generate Tools Utilities Window				
🗼 🖹 🆻 🌪 🕼 🌮 Code		Framewo	rk: C:\U	sers∖M
🗋 Active Components 💦 🗖 🗖	Project Graph			
🗙 🗔 🖪 🖻 🖷 🕇			View:	Roo
ADC CMSIS Pack Device Family Pack (DFP) EVSYS NVMCTRL SERCOM2 SERCOM3 STDIO System		ISIS Pack		
Drag into the Design Graph Window and release to add this compone	ent.			

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)

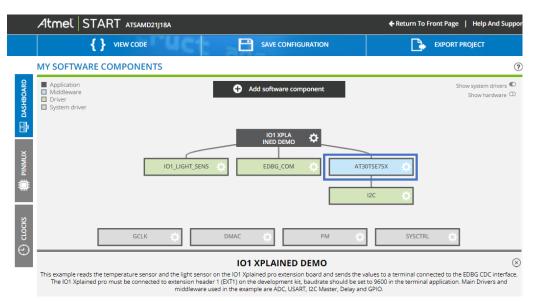
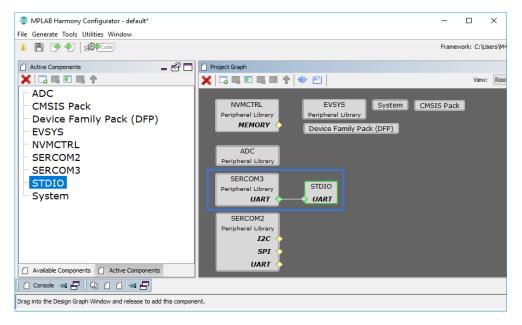


Figure 6-2. Middleware Identification Under Atmel START

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.

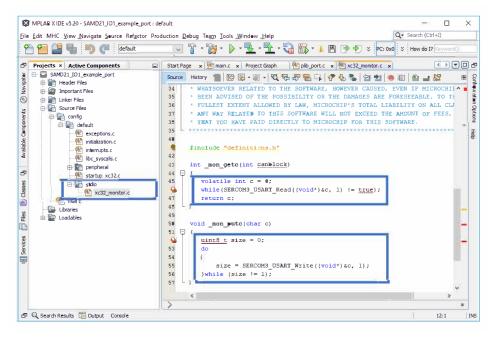
1. In MPLAB X IDE, add the STDIO middleware to the project using the MHC Project Graph window then link the STDIO middleware to the SERCOM3 driver.

Figure 6-3. Add STDIO Middleware Under MHC



After regenerating the project source from the MHC interface by clicking to 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

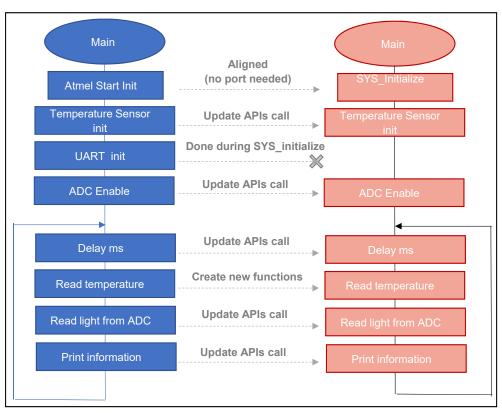
After drivers, system, and middleware layers are ported to the MPLAB Harmony 3 project, all available peripheral libraries and middleware APIs required to port the application are available in the project directory.

Projects ×		_
E SAMD	21_IO1_example_port	^
📄 🛱 He	eader Files	
	config	
	👍 default	
	🔤 definitions.h	
	🛱 👍 peripheral	
	🖨 🚾 adc	
	- 🔁 plib_adc.h	
	🔤 🔤 plib_adc_common.h	
	🖶 📠 dock	
	🖶 💼 evsys	
	🖶 🧰 nvic	
	🖶 🧰 nvmctrl	
	🕮 💼 port	
	i2cm	
	Plib_sercom_i2c_master.h	
	i usart	
	Plib_sercom3_usart.h	
	Plib_sercom_usart_common.h	
	E systick	
	Plib_systick.h	
	toolchain_specifics.h	
	packs	
	iportant Files	
	nker Files	\sim

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.





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</pre>
    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];</pre>
    sign -= (bool)(data & (1 << 15)) << 1;
data &= ~(1 << 15);</pre>
    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/01 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 ^{\star/}
    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.

© 2019, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

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

MERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
orporate Office	Australia - Sydney	India - Bangalore	Austria - Wels
855 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
handler, AZ 85224-6199	China - Beijing	India - New Delhi	Fax: 43-7242-2244-393
el: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	Denmark - Copenhagen
ax: 480-792-7277	China - Chengdu	India - Pune	Tel: 45-4450-2828
echnical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
ttp://www.microchip.com/support	China - Chongging	Japan - Osaka	Finland - Espoo
Veb Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
ttp://www.microchip.com	China - Dongguan	Japan - Tokyo	France - Paris
tlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
uluth, GA	China - Guangzhou	Korea - Daegu	Fax: 33-1-69-30-90-79
el: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	Germany - Garching
ax: 678-957-1455	China - Hangzhou	Korea - Seoul	Tel: 49-8931-9700
ustin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	Germany - Haan
el: 512-257-3370	China - Hong Kong SAR	Malaysia - Kuala Lumpur	Tel: 49-2129-3766400
oston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	Germany - Heilbronn
/estborough, MA	China - Nanjing	Malaysia - Penang	Tel: 49-7131-72400
el: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	Germany - Karlsruhe
ax: 774-760-0088	China - Qingdao	Philippines - Manila	Tel: 49-721-625370
hicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	Germany - Munich
asca, IL	China - Shanghai	Singapore	Tel: 49-89-627-144-0
el: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
ax: 630-285-0075	China - Shenyang	Taiwan - Hsin Chu	Germany - Rosenheim
allas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
ddison, TX	China - Shenzhen	Taiwan - Kaohsiung	Israel - Ra'anana
el: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
ax: 972-818-2924	China - Suzhou	Taiwan - Taipei	Italy - Milan
etroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
ovi, MI	China - Wuhan	Thailand - Bangkok	Fax: 39-0331-466781
el: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	Italy - Padova
ouston, TX	China - Xian	Vietnam - Ho Chi Minh	Tel: 39-049-7625286
el: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	Netherlands - Drunen
dianapolis	China - Xiamen		Tel: 31-416-690399
blesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
el: 317-773-8323	China - Zhuhai		Norway - Trondheim
ax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
el: 317-536-2380			Poland - Warsaw
os Angeles			Tel: 48-22-3325737
ission Viejo, CA			Romania - Bucharest
el: 949-462-9523			Tel: 40-21-407-87-50
ax: 949-462-9608			Spain - Madrid
l: 951-273-7800			Tel: 34-91-708-08-90
aleigh, NC			Fax: 34-91-708-08-91
1: 919-844-7510			Sweden - Gothenberg
ew York, NY			Tel: 46-31-704-60-40
1: 631-435-6000			Sweden - Stockholm
an Jose, CA			Tel: 46-8-5090-4654
el: 408-735-9110			UK - Wokingham
el: 408-436-4270			Tel: 44-118-921-5800
anada - Toronto			Fax: 44-118-921-5820
l: 905-695-1980			1 u. 11 110-02 1-0020
x: 905-695-2078			