

# MPLAB® Harmony Help - MPLAB Harmony Configurator User's Guide

MPLAB Harmony Integrated Software Framework v1.11

© 2013-2017 Microchip Technology Inc. All rights reserved.

## MPLAB Harmony Configurator User's Guide

This section provides user information on using the MHC.

## Installing MHC

This topic provides information on installing the MHC plug-in.

#### Description

#### Installing the MHC Plug-in

- 1. Start MPLAB X IDE and select <u>Tools > Plugins</u>.
- 2. Select the Downloaded tab and click Add Plugins...
- 3. In the Add Plugins dialog, navigate to the MHC com-microchip-mplab-modules-mhc.nbm plug-in file, which is located in <install-dir>/utilities/mhc, and then click Open.



4. Ensure that the Install check box for the plug-in is selected and click Install.

Add Plugins	Search:
Instal Name MPLAB Harmony Configurator	MPLAB® Harmony Configurator
	Community Contributed Plugin Version: 1.0.7.16 Author: Microchip Technology Inc. Date: 2/10/16 Source: Microchip Plugins Homepage: www.microchip.com/harmony Plugin Description MPLAB® Harmony Configurator.
Install 1 plugin selected	

5. Follow the prompts from the installation and continue until the installation completes. (Do not be concerned if the version you are installing is *signed* but not *trusted*, simply click **Continue**). Once the installation has finished you can close the **Plugins** dialog.

6. To verify the installation, select *Tools > Plugins* and select the **Installed** tab. The MHC plug-in you installed should be included in the list.

## MPLAB Harmony Configurator Interface

This section describes the MHC interface.

#### Description

This section provides a basic overview of the MHC user interface. For detailed information on using MHC to create a MPLAB Harmony application, refer to Using MHC to Create a New Application.

#### **Initial Interface Configuration**

The following figure shows the initial interface configuration for MHC.



#### **Main Window**

This view shows the available configuration options for the selected Microchip device, which is arranged in a hierarchal tree structure. Click the check box to enable a specific component. The options for the enabled component will appear.

#### **Help Window**

When a tree component is interacted with, the corresponding help information is displayed in the Help Window.



#### **Output Window**

The output window displays various log messages about the actions taken by the MPLAB Harmony Configurator.

#### **Main Window Toolbar**

The main window contains a context-sensitive toolbar. This toolbar provides both global and tab-specific functionality. When viewing the **Options** tab, this toolbar contains the following functionality:



Open: Select the Open icon to open a saved .mhc configuration into the current Option tree.

MPLAB® Harmony Configurator* ×	
Options Clock Diagram × Pin Diagram ×	
🖶 Open Configuration	×
	<u> </u>

Save: Select the Save icon to save the current Option tree into the last used .mhc file or click Save As to save to a new file.

MPLAB® Harmony Configurator* ×	
Options Clock Diagram × Pin Diagram ×	
🖶 Save Configuration	×
Current Configuration Path	
C: Wicrochip \harmony \local \software \jsp_root \apps \example \firmware \src \sys	stem_config\default\default.mhc
	Save Save As

Import: Selecting the Import icon opens the import data dialog. This dialog can be used to import different types of information into the current project.



**Export:** Selecting the Export icon opens the export data dialog. This dialog can be used to export different types of information from the current project.



Generate: Selecting the Generate icon opens the project file generation dialog.

X
Generate Cancel

Framework Options: Selecting the Framework Options icon opens the framework configuration dialog.

MPLAB® Harmony Configurator* × Coptions Clock Diagram × Pin Diagram ×	
Harmony Framework Configuration	×
Current Harmony Path	
C: Wicrochip \harmony \local \software \isp_root	<b>A</b>
Current Version: 1.06	Change

Application Launcher: Selecting the Application Launcher icon provides the ability to quickly launch applications such as the clock configurator, pin configurator, or the MPLAB Harmony Graphics Composer.

MPLAB® Harmony Co	onfigurator* ×
Options Clock Diagr	am × Pin Diagram ×
	Clock Configuration
<u>× </u>	Graphics Composer Pin Configuration

Option Tree View: Selecting the Option Tree View icon toggles the option tree between global and active view.



### **Project Generation**

Once all of the desired options have been selected from the configuration tree, the next step is file generation, which is done by clicking **Generate** in the main window. Various options for generation are displayed in the File Generation dialog.



 Overwrite local changes – Automatically overwrites any local changes made by the user. A merge window will be displayed for all locally changes files if this option is not selected.

- Enable recommended compiler optimizations (if not set) A compiler optimization level of at least 'O1' is highly recommended for MPLAB Harmony projects. This option will set the compiler optimization level to 'O1' if no optimization level is currently set.
- The Generate button will cause all of the selected components and options to be processed and output as valid code files. These files will be automatically added to the project.

## Using MHC to Create a New Application

Provides information on creating a new MHC project.

#### Introduction

This section provides an introduction to creating your own MPLAB Harmony applications using the MPLAB Harmony Configurator (MHC).

#### Description

MPLAB Harmony provides a MPLAB Harmony Configurator (MHC) MPLAB X IDE plug-in that can be installed in MPLAB X IDE to help you create your own MPLAB Harmony applications.

To create a new MPLAB Harmony application with MHC, follow these three steps:

- Step 1: Create the New Harmony Project
- Step 2: Add and Configure Required Libraries/Modules
- Step 3: MPLAB Harmony Application Structure and Developing the Application
- Note: If you are a Microchip Libraries for Applications (MLA) user, and will be porting your application from the MLA TCP/IP, File System, USB Device, Graphics, or peripheral libraries to the MPLAB Harmony equivalents, refer to Porting to MPLAB Harmony for more information.

#### Prerequisites

This topic describes the prerequisites for creating your own MPLAB Harmony applications using MHC.

#### Description

This tutorial assumes that you have already completed these steps before you start:

- 1. Installed the MPLAB X IDE (http://www.microchip.com/mplabx).
- 2. Installed MPLAB Harmony (http://www.microchip.com/harmony).
- 3. Installed the MPLAB XC32 C/C++ Compiler (http://www.microchip.com/xc32).
- 4. Set up a working PIC32 development platform (http://www.microchip.com/32bit).

You can download the MPLAB X IDE, MPLAB Harmony and the MPLAB XC32 C/C++ Compiler from the links provided. If you do not already have a PIC32 development platform, you can learn more about the PIC32 family and determine which hardware platform best meets your development needs by visiting the 32-bit website listed previously.

This tutorial also assumes that you have some familiarity with the MPLAB X IDE, embedded C-language programming and PIC32 microcontrollers. If you are unsure how to complete some of the steps in this tutorial, please refer to the documentation for the item on which you have questions. You may also seek assistance from your peers on the Microchip discussion forums (http://www.microchip.com/forums) or from the Microchip support staff (www.microchip.com/support).

Once you have everything installed, connected, and up and running you are ready to begin creating your own MPLAB Harmony applications.

#### Step 1: Create the New Project

To create a new MPLAB Harmony project, you first need to create a new MPLAB X IDE project and the basic set of source code files and functions that are necessary for a properly formed MPLAB Harmony application.

#### Description

- To create a new MHC project:
- 1. Select File > New Project or click the New Project icon in MPLAB X IDE.
- 2. In Categories, select **Microchip Embedded** and in Projects select **MPLAB Harmony Project** from the list of available project templates, and then click **Next** to launch the Microchip Harmony Configurator Project Wizard.

teps	Choose Project	
Choose Project	Q Filter:	
	Categories:	Projects:
	Microchip Embedded	32-bit MPLAB Harmony Project
	Other Embedded	Standalone Project
	⊞- 🛄 Samples	Prebuilt (Hex, Loadable Image) Project
		User Makefile Project
		Library Project
	) Desiriet	
	Description:	
	MPLAB® Harmony Project Wizard	

- 3. Specify the following in the New Project dialog:
  - Harmony Path (path to the folder containing Harmony framework: <install-dir>)
  - Project Location (the default project path is the apps folder within the selected MPLAB Harmony path)
  - Project Name
  - Configuration Name (optional)
  - Target Device (when a valid harmony path is selected, the device selection menu will be filled)

Steps	Choose Project	
L. Choose Project	Q. Filter:	
	Categories:	Projects:
	Wicrochip Embedded     Wicrochip Embedded     Other Embedded     Samples	Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project
	Description:	
1	Creates a new standalone application project.	on project. It uses an IDE-generated makefile to build your
- 11-		

4. A MPLAB Harmony project will be created and the MPLAB Harmony Configurator will open. Refer to MPLAB Harmony Configurator for additional information.



## Step 2: Add and Configure the Required Libraries and Modules

This topic describes how to configure the MPLAB Harmony library modules.

#### Description

- 1. In the Main window, expand the Device Configuration tree and select the desired device configuration settings.
- 2. Expand the MPLAB Harmony Project Configuration tree and select and configure the desired libraries.
- 3. If use of a Board Support Package is desired, expand the BSP Configuration tree and select the desired BSP.
- 4. When complete, generate and save the configuration.
- 5. Develop your application logic using the selected libraries.

At this point, you should be able to build, debug, and step through the application. Effectively, you have a running MPLAB Harmony system; however, it is not yet ready to do anything. Next, you will develop your application state machine logic and make sure the system does what you want it to do.

#### Step 3: MPLAB Harmony Application Structure and Developing the Application

This topic describes the steps necessary to maintain the state machines.

#### Description

#### main.c

The main.c file contains calls to the SYS\_Initialize function, which initializes MPLAB Harmony modules, as well as applications. It also contains the main task execution, which calls tasks for all selected MPLAB Harmony modules, as well as the application task function, APP\_Tasks.

#### app.c

The app.c file contains the APP\_Initialize function that is used to place an application into its initial state. It will be called from the SYS\_Initialize

function. The APP\_Task function, which is also contained in the app.c file, implements the application state machine logic. Add application code to this task as desired.

Refer to the example applications located in the <install-dir>/apps/ folder within your MPLAB Harmony installation for example applications for various MPLAB Harmony modules. Related documentation is available in the *Applications Help > Examples* section.

## Porting a Legacy PLIB to MPLAB Harmony

Provides an example on how to port a legacy (i.e., prior to MPLAB Harmony) USART Peripheral Library (PLIB) demonstration application to a MPLAB Harmony application using the MPLAB Harmony Configurator (MHC).

#### **Description**

A detailed procedure for porting the legacy UART PLIB Interrupt demonstration application

(<compiler-install-dir>/examples/plib\_examples/uart/uart\_interrupt) to MPLAB Harmony is provided in the Framework Help > Peripheral Library Help > Peripheral Library Porting Example .

In this example, the following assumptions are made:

- The PIC32MX795F512L device will be used; however, the process described in this section is applicable for other PIC32 devices with appropriate changes
- The Explorer 16 Development Board is the hardware used in this example
- For the v1.33 MPLAB XC32 C/C++ Compiler, the examples folder is not present. To view the legacy USART PLIB example, refer to v1.31 or earlier of the MPLAB XC32 C/C++ compiler.

## Configuring the Oscillator Module Using the MHC Clock Configurator

Provides information configuring the Oscillator module using the MHC Clock configurator

#### Description

The MHC Clock Configurator is a component of the MPLAB Harmony Configurator (MHC) MPLAB X IDE plug-in. Its function is to provide a graphical user interface to configure the Oscillator module.

While simulating the normal operation of the Oscillator module, the MHC Clock Configurator contains interactive controls, dynamic output, and visual warnings to help guide the user in establishing the desired system clock configuration.

The MHC Clock Configurator is launched automatically when the MHC is launched. It is in the form of a tab panel in MPLAB X IDE. Clicking the MPLAB Harmony Clock Configuration tab will open the MHC Clock Configurator.

MPLAB® Harmony Configurator* ×	
📕 🖹 🎐 🕗 🕼 🥮 📗	- T-
Options Clock Diagram × Pin Diagram ×	1

The clock configurator screen can also be accessed using the main window toolbar application launch feature. Simply click the application launch icon and select **Clock Configuration**.



Another way to access the MHC Clock Configurator is via the Clock System Service section in MHC Harmony & Application Configuration tree view. Pressing the Execute button at the Launch Clock Configurator topic will either bring the tab panel into focus or launch the MHC Clock Configurator, if the tab panel was closed.



Note: The MHC Clock Configurator is one option to configure the Oscillator Module. Another option is to configure directly via the MPLAB Harmony & Application Configuration tree structure. The majority of the settings captured in the MHC Clock Configurator exist under the Clock Configurator Settings node in the Clock System Service, while the remainder are in the Device Configuration section.

## Clock Configuration for PIC32MZ Family Devices

Provides configuration information for PIC32MZ family devices.

#### Description

The MHC Clock Configurator's support of configuring the Oscillator Module of a PIC32MZ family device is divided into the following sub-sections:

- Configuring System Clock Frequency
- Configuring the Peripheral Bus Clocks
- Configuring the Reference Clocks
- Using the SPLL Divider Auto-Calculate Feature

For details regarding the operation of the Oscillator module, refer to the "Oscillator" chapter in the "PIC32MZ Embedded Connectivity (EC) Family Data Sheet" (DS60001191). This document is available for download from the Microchip website (www.microchip.com).



#### Configuring the System Clock Frequency

Provides information on configuring the system clock frequency for PIC32MZ family devices.

#### Description

There are a total of five external and internal oscillator options as clock source:

- Internal Fast RC (FRC) Oscillator divided by the FRCDIV bits in the OSCCON register
- Internal Low-Power RC (LPRC) Oscillator
- Secondary Oscillator (SOSC)
- Primary Oscillator (POSC) (POSCMOD: HS or EC)
- System PLL (SPLL)

The device configuration bit FNOSC is represented as a drop-down with the above selections in the MHC Clock Configuration. The current selection is represented in **bold**.



The Primary Oscillator (POSC) and Secondary Oscillator (SOSC) are customizable external clock sources. For the POSC, the device configuration bit, POSCMOD, needs to be set to EC or HS. If FNOSC is set to SOSC, the device configuration bit, FSOSCEN, should be set to ON. SOSCEN is set post-initialization. There is an option to override FSOSCEN with SOSCEN.



The output system clock frequency (SYSCLK) is displayed on the left side. This value (in Hz) corresponds to System Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.



Certain frequency values may be displayed in red when the input value does not meet specification and may cripple performance of the device. An example is shown in the following figure, when the HS Oscillator Mode is selected for POSCMOD and the POSC input frequency set is outside of the 4 MHz - 32 MHz range. A dynamic help tip will also appear if the user hovers over the POSCMOD control or any of the red text.



Another example is the SPLL, where FPLL (60 MHz – 120 MHz), FVCO (80 MHz – 240 MHz), and FIN (range specified by PLLRANGE) will appear as red text, including an explanation tool tip, if they fall outside of their respective required ranges.



## **Configuring the Peripheral Bus Clocks**

Provides information on configuring the peripheral bus clocks for PIC32MZ family devices.

#### Description

Each of the eight Peripheral Bus Clocks on the PIC32MZ family devices can be configured by using the tabs on the left.



The output frequency is in **bold**. The "To Peripherals" window provides a reminder of which peripherals each clock is driving. This value (in Hz) corresponds to Peripheral Bus Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

**Note:** It is important to know the acceptable clock range for the peripherals. The Clock Configurator will NOT provide a warning if the output peripheral clock frequency falls outside of the specified range of the peripheral.

#### **Configuring the Reference Clocks**

Provides information on configuring the reference clocks for PIC32MZ family devices.

#### Description

Each of the four Reference Clocks on the MZ Family of device can be configured by using the tabs on the left.



The clock input source (ROSELx), divider (RODIVx), trim value (ROTRIMx) are independently configurable. The output frequency (REFCLKOx) is in **bold**.

This value (in Hz) corresponds to Reference Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in the MHC Harmony & Application Configuration tree view.

## Using the Reference Clock Auto-Calculate Feature

Provides information on the reference clock auto-calculate feature for PIC32MZ family devices.

#### Description

The MHC Clock Configurator is equipped with the ability to help the user establish the closest possible match to a user-desired target reference clock frequency. The Auto-Calculate feature is designed to determine the divider and trim values in the each of the four reference clocks based on a user requested clock output frequency.

The feature can be accessed via the Auto-Calculate button in the Reference Clock section of the Clock Configurator.



Clicking the Auto-Calculate button opens the Auto-Calculate dialog.

	be and and the
Target Reference Frequency	96,000,000 H
<b>REFCLK Input Frequency</b>	200000000 H
Best Achievable Frequency	96,060,037.52 H
% Error	0.06254 %
	and a state
	Apply Cancel

Enter the desired target reference frequency (remember to press the <Enter> key), and the dialog window will display the best achievable frequency that can be provided by the Reference Clock Divider (RODIVx) and Trim (ROTRIMx) combination, as well as the percentage discrepancy from the desired value, if any. The REFCLK Input Frequency is determined based on selection at ROSELx.

If the I2S driver is selected as part of the configuration, the Reference Clock Divider and Trim Auto-Calculator dialog opens automatically reconfigured with the option to use the target I2S input frequency as the target reference frequency.

<ul> <li>Target I2S Input Frequency</li> <li>12288000 Hz</li> <li>I2S Baud Rate (Audio Sample Rate)</li> <li>I2S Baud Rate Multiplier (MCLK)</li> <li>REFCLK Input Frequency</li> <li>Best Achievable Frequency</li> <li>% Error</li> <li>0.016 %</li> </ul>	Target Reference Free	quency	200,000,000 🚔	Hz
12S Baud Rate (Audio Sample Rate)48000Hz12S Baud Rate Multiplier (MCLK)256REFCLK Input Frequency200000000 HzBest Achievable Frequency12,289,966.39 Hz% Error0.016 %	Target I2S Input Free	quency	12288000	Hz
12S Baud Rate Multiplier (MCLK)     256       REFCLK Input Frequency     200000000 Hz       Best Achievable Frequency     12,289,966.39 Hz       % Error     0.016 %	12S Baud Rate (Audio Sample	Rate)	48000	H
REFCLK Input Frequency200000000 HzBest Achievable Frequency12,289,966.39 Hz% Error0.016 %	I2S Baud Rate Multiplier (	MCLK)	256	
Best Achievable Frequency         12,289,966.39         Hz           % Error         0.016 %	REFCLK Input Free	quency	20000000	H
% Error 0.016 %	Best Achievable Free	luency	12,289,966.39	Hz
	%	Error	0.016	%

Clicking the **Apply** button will cause the MHC Clock Configurator to update the Reference Clock divider and trim to establish the closest achievable frequency.

#### Using the SPLL Divider Auto-Calculate Feature

Provides information on the SPLL auto-calculate feature for PIC32MZ family devices.

#### Description

The MHC Clock Configurator is equipped with the ability to help the user establish closest possible match to a user-desired target system clock frequency. The Auto-Calculate feature is designed to determine the divider and multiplier values in the SPLL-based on a user requested system clock frequency.

The feature can be accessed via the Auto-Calculate button in the SPLL section of the Clock Configurator.



Clicking the Auto-Calculate button opens the Auto-Calculate dialog.

Auto-Calculate SPLL Dividers	X
Desired System Frequency	200,000,000 🗮 Hz
PLL Input Frequency	8000000 Hz
Best Achievable Frequency	200000000 Hz
% Error	0 %
Appl	ly Cancel

Enter the desired system clock frequency (remember to press the key ENTER), and the dialog window will display the best achievable frequency that can be provided by the SPLL divider/multiplier combination, as well as the percentage discrepancy from the desired value, if any. The PLL Input Frequency is determined based on selection at PLLICLK (FRC or POSC).

Clicking the **Apply** button will cause the MHC Clock Configurator to update the SPLL dividers and multiplier to establish the closest achievable frequency.

Wote: The Auto-Calculate feature will also update the PLLRANGE setting to satisfy the necessary FIN frequency.

## **Clock Configuration for PIC32MX Family Devices**

Provides configuration information for PIC32MX family devices.

#### Description

The MHC Clock Configurator's support of configuring the Oscillator Module of a MX Family Device is divided into the follow sub-sections:

- Configuring the System Clock Frequency
- Configuring the Peripheral Bus Clock
- Configuring the Reference Clock
- Configuring the USB PLL
- Using the SPLL Divider Auto-Calculate Feature

For details regarding the operation of the Oscillator module, refer to the "Oscillator" chapter in the specific PIC32MX device data sheet:

- PIC32MX1XX/2XX (DS60001168)
- PIC32MX1XX/2XX/5XX 64/100-pin Family (DS60001290)
- PIC32MX320/340/360/420/440/460 (DS60001143)
- PIC32MX330/350/370/430/450/470 (DS60001185)
- PIC32MX5XX/6XX/7XX (DS60001156)

Each of these documents are available for download from the Microchip website (www.microchip.com).

The following figure shows the configuration screen for PIC32MX1XX/2XX, PIC32MX 330/350/370/430/450/470, and PIC32MX1XX/2XX/5XX





The next figure shows the configuration screen for PIC32MX320/340/360/420/440/460 and PIC32MX5XX/6XX/7XX devices.



## Configuring the System Clock Frequency

Provides information configuring the system clock frequency for PIC32MX family devices.

## Description

There are a total of five external and internal oscillator options as clock source:

- Internal Fast RC Oscillator (FRC) divided by the FRCDIV bits in the OSCCON register
- Internal Fast RC Oscillator (FRC) divided by 16
- Internal Low-Power RC (LPRC) Oscillator
- Secondary Oscillator (SOSC)
- Primary Oscillator with PLL module (PRIPLL)
- Primary Oscillator (POSCMOD: XT, HS, or EC)
- Internal Fast Internal RC Oscillator with PLL module via Postscaler (FRCPLL)
- Internal Fast Internal RC Oscillator (FRC)

The device configuration bit FNOSC is represented as a drop-down with the above selections in the MHC Clock Configuration. The current selection is represented in **bold**.



Primary Oscillator (POSC) and Secondary Oscillator (SOSC) are customizable external clock source. For POSC, the device configuration bit POSCMOD needs to be set to EC, XT, or HS. If FNOSC is set to SOSC, the device configuration bit FSOSCEN needs to be set to ON.



The output system clock frequency (SYSCLK) is displayed on the left side. This value (in Hz) corresponds to System Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.



Certain frequency values may be displayed in red when the input value does not meet specification and may cripple performance of the device. An example is shown in the following figure, when the XT Oscillator Mode is selected for POSCMOD and the POSC input frequency set is outside of the 3 MHz - 10 MHz range. A dynamic help tip will also appear if the user hovers over the POSCMOD control or any of the red text.



Another example is the SPLL, where FPLL (40 MHz – 120 MHz), FVCO (60 MHz – 120 MHz), and FIN (3.92 MHz – 5 MHz) will appear in red text, including an explanation tool tip, if they fall outside of their respective required ranges.



#### Configuring the Peripheral Bus Clock

Provides information on configuring the peripheral bus clock for PIC32MX family devices.

#### Description

The Peripheral Bus Clock on the MX Family of device can be configured on the left.



The output frequency is in **bold**. This value (in Hz) corresponds to Peripheral Bus Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

**Note:** It is important to know the acceptable clock range for the peripherals. The Clock Configurator will NOT provide a warning if the output peripheral clock frequency falls outside of specified range of the peripheral.

#### **Configuring the Reference Clock**

Provides information on configuring the reference clock for PIC32MX family devices.

#### Description

The Reference Clock on the PIC32MX1XX/2XX, PIC32MX 330/350/370/430/450/470, and PIC32MX1XX/2XX/5XX 64/100-pin Family devices can be configured in the section labeled Reference Clock on the upper right area of the screen.



The clock input source (ROSEL), divider (RODIV), trim value (ROTRIM) are independently configurable. The output frequency (REFCLKO) is in bold.

This value (in Hz) corresponds to Reference Clock Frequency under Calculated Clock Frequencies in the Clock System Service section in MHC Harmony & Application Configuration tree view.

#### Using the Reference Clock Auto-Calculate Feature

Provides information on the reference clock auto-calculate feature for PIC32MX family devices.

## Description

The MHC Clock Configurator is equipped with the ability to help the user establish closest possible match to a user-desired target reference clock frequency. The Auto-Calculation feature is designed to determine the divider and trim values for the reference clock based on a user requested clock output frequency.

The feature can be accessed via the Auto-Calculate button in the Reference Clock section of the Clock Configurator.



Clicking the Auto-Calculate button opens the Auto-Calculate dialog.

uto-Calculator
96,000,000 € Hz
200000000 Hz
96,060,037.52 Hz
0.06254 %
Apply Cancel

Enter the desired system clock frequency (remember to press the <Enter> key), and the dialog window will display the best achievable frequency that can be provided by the Reference Clock Divider (RODIV) and Trim (ROTRIM) combination, as well as the percentage discrepancy from the desired value, if any. The REFCLK Input Frequency is determined based on selection at ROSEL.

If the I2S driver is selected as part of the configuration, the Reference Clock Divider and Trim Auto-Calculator dialog opens automatically reconfigured with the option to use the target I2S input frequency as the target reference frequency.

<ul> <li>Target I2S Input Frequency</li> <li>12288000 Hz</li> <li>I2S Baud Rate (Audio Sample Rate)</li> <li>I2S Baud Rate Multiplier (MCLK)</li> <li>REFCLK Input Frequency</li> <li>Best Achievable Frequency</li> <li>% Error</li> <li>0.016 %</li> </ul>	Target Reference Free	quency	200,000,000 🚔	Hz
12S Baud Rate (Audio Sample Rate)48000Hz12S Baud Rate Multiplier (MCLK)256REFCLK Input Frequency200000000 HzBest Achievable Frequency12,289,966.39 Hz% Error0.016 %	Target I2S Input Free	quency	12288000	Hz
12S Baud Rate Multiplier (MCLK)     256       REFCLK Input Frequency     200000000 Hz       Best Achievable Frequency     12,289,966.39 Hz       % Error     0.016 %	12S Baud Rate (Audio Sample	Rate)	48000	H
REFCLK Input Frequency200000000 HzBest Achievable Frequency12,289,966.39 Hz% Error0.016 %	I2S Baud Rate Multiplier (	MCLK)	256	
Best Achievable Frequency         12,289,966.39         Hz           % Error         0.016 %	REFCLK Input Free	quency	20000000	H
% Error 0.016 %	Best Achievable Free	luency	12,289,966.39	Hz
	%	Error	0.016	%

Clicking the **Apply** button will cause the MHC Clock Configurator to update the Reference Clock divider and trim to establish the closest achievable frequency.

#### Configuring the USB PLL

Provides information on configuring the USB PLL for PIC32MX family devices.

#### Description

Part of enabling the USB peripheral is to enable the USB PLL. The USB PLL requires 4 MHz input clock frequency for accurate operation. With POSC being a variable value, it is important to configure the correct USB PLL Input Divider (UPLLIDIV) value. The MHC Clock Configurator will provide visual warning if the value can lead to inaccuracy in USB operation.



#### Using the SPLL Divider Auto-Calculate Feature

Provides information on using the SPLL Divider Auto-Calculate feature for PIC32MX family devices.

#### Description

The MHC Clock Configurator is equipped with the ability to help the user establish closest possible match to a user-desired target system clock frequency. The Auto-Calculation feature is designed to determine the divider and multiplier values in the SPLL-based on a user requested system clock frequency.

The feature can be accessed via the Auto-Calculate button in the System PLL section of the Clock Configurator.



Clicking the Auto-Calculate button opens the Auto-Calculate dialog.

Auto-Calculate SPLL Dividers	X
Desired System Frequency	80,000,000 🚔 Hz
PLL Input Frequency	12000000 Hz
Best Achievable Frequency	80000000 Hz
% Error	0 %
	Apply Cancel

Enter the desired system clock frequency (remember to press the <Enter> key), and the dialog window will display the best achievable frequency that can be provided by the SPLL divider/multiplier combination, as well as the percentage discrepancy from the desired value, if any. The PLL Input Frequency is determined based on selection at FNOSC (FRCPLL or PRIPLL).

Clicking the Apply button will cause the MHC Clock Configurator to update the SPLL dividers and multiplier to establish the closest achievable frequency.

## MPLAB Harmony Graphical Pin Manager

Provides information on the MPLAB Harmony Graphical Pin Manager tool that resides within MHC.

#### Description

This graphical management tool exists for the purpose of enabling users to configure the pins of Microchip devices in a fast and intelligent manner. The tool consists of a graphical representation of the state of the component and table that provides the means to configure the pins of the device. Users intending to use this tool should be familiar with the MPLAB Harmony configuration tree.

The user configures a device using the following process:

- Launch the tool (if not already running)
- Add modules by enabling desired functionality in the configuration tree (e.g., USART or SPI)
- Using the pin table to "Lock" cells representing function and pin pairings
- Using the pin flag management dialog to change pin register values
- Generating resultant code through the Generate button

Once generation is complete, the resultant code for configuring the device pins will be automatically added to the user's project.

## Launching the Tool

Describes how to launch the pin manager tool.

#### Description

The pin manager tool automatically launches when MHC starts.



The pin manager tool can be launched from the main window toolbar application launcher or from the option tree.



The pin manager tool can also be launched from the configuration tree.

E -System Services	
⊡-Clock	
⊡-Command	
⊡-Common	
t∄Console	
⊡-Debug	
⊡-Device Control	
⊡-DMA	
⊞-File System	
. Interrupts	
I → Message	
- Ports	
🗄 🔽 🔽 Use Ports System Service?	
Launch Pin Manager Execute	
***** Use Pin Manager to configure PPS, CN, Pull-Up, Pull-Down, Open Drain, Digital & Analog functionality	****

### **Tool Tabs**

The pin manager tool has two tabs:

- Pin Diagram (see the red section in the following figure)
- Pin Table (see the blue section in the following figure)

	MPLAS® Harmon	Config Gagram	jurator C	r* × orde Pin Di	agran		 	¢	¥															3	1				
	Unavailab	le																											
	Available																												
	Locked																												
									MCL RA RB RB RB RB RB RB RB SOSC SOSC SOSC SOSC VD RE		O 1 2 3 4 5 6 7 8 9 10 11 12 13 14	PIC32MX110F016B	28 27 26 25 24 23 22 21 20 19 18 17 16 15		AVDD AVSS R815 R814 R813 R812 R811 R810 VSS R89 R88 R87 R86														
t Pin Table ¥   ige: SOIC <u>7</u>	Pin Settings	MCLR	RAD	RAI	R80	RB1	R82	RB3	VSS	RAZ	RA3	DSOS	sosco	VDD	RBS	RB6	R87	RBG	R89	VSS	VCAP	RB10	RB11	RBIZ	RBI3	R814	RBIS	AVSS AVDO	
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27 28	Ĩ
Clock	SOSCI							11																					1
(OSC_ID_0)	SOSCO												-																
-	PGED1	1			D																		1					3	100
Debug	PGEC1	1				D																	1						Ĩ
		1	-					-						-	_				_	-		_				_			ŝ

## Pin Diagram Tab

Describes the pin diagram features.

## Description

Outpu Packa

This diagram is a graphical representation of the selected component to be configured. The diagram contains the following: Pin Names

These are the base names of each pin. These names will change based on the selected function for this pin.

**Pin States** 

This is a graphical indication of the state of the pin. *Pin States Legend:* 

Color	lcon	Description
Blue		This pin can be locked to an available function in the table.
Gray		This pin is currently unavailable based on the state of the pin table.
Green		This pin has been locked to a function.
Red		This pin has been automatically locked to a pin based on function priority.
Yellow		This pin is currently highlighted by the cursor.



#### **Pin Numbers**

The number for each pin.



#### **Component Name**

The name of this component.

	0			L	
MCLR	1		28		AVDD
RAO	2		27		AVSS
RA1	3	σ	26		RB15
RBO	4	Ā	25		RB14
RB1	5	μ	24		RB13
RB2	6	R	23		RB12
RB3	7	$\Xi$	22		RB11
VSS	8	H	21		RB10
RA2	9	유	20		VCAP
RA3	10	Ő	19		VSS
SOSCI	11	5	18		RB9
SOSCO	12	ω	17		RB8
VDD	13		16		RB7
RB5	14		15		RB6

#### **Pin Table Tab**

Describes the pin table features.

#### Description

The pin table allows the user to graphically configure the pins for the given component. The table contains the following areas of interest: **Package Selector** 

## This menu contains the available packages for the selected component.

**Note:** Changing this value will reset the state of the pins to default.

Output Pin Table ×																													
Package: SOIC 💌	Pin Settings	MCLR	RAD	RAI	RBO	RB1	RB2	RB3	VSS	RA2	RA3	IDSOS	SOSCO	VDD	RBS	RB6	R87	RBS	RB9	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI											1														Γ			
(OSC_ID_0)	SOSCO												•																
Dahua	PGED 1				D																								
Debug	PGEC1					Ď																							

Observe the changes in the diagram and table when the QFN package is selected for this device.



#### **Pin Settings Button**

This button shows the pin settings configuration menu. This dialog allows for the configuration of pin direction, drain, mode, latch, change notification, and pull-up and pull-down options.

Note: The direction and mode options are dependent on the function that is assigned to the pin. Board Support Package functions may lock other options as well.

Output Pin Table X																													
Package: SOIC	Pin Settings	MCLR	RAO	RAI	RBO	RB1	RB2	RB3	VSS	RA2	RA3	IDSOS	SOSCO	QQA	RBS	RB6	RB7	RBS	RB9	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI											Ê																	
(OSC_ID_0)	SOSCO												Ê																
Dahua	PGED 1				D																								
Debug	PGEC 1					D																							

The pin settings dialog can also be launched from the main toolbar when the pin diagram is visible.



Pin	Name	Voltage Tolerance	Function	Direction (TRIS)	Latch (LAT)	Open Drain (ODC)	Mode (ANSEL)	Change Notification (CNEN)	Pull Up (CNPU)	Pull Down (CNPD)
1	MCLR	5V		In	n/a		Digital		5	Б
2	RAO			In	in/a		Analog	Г		
3	RA1			In	n/a		Analog			
4	RB0			In	n/a		Analog			
5	RB1			In	n/a		Analog			
6	RB2			In	n/a		Analog			Ē
7	RB3			In	n/a		Analog			
8	VSS	5V		In	n/ä		Digital	Ē		E
9	RA2			In	n/a	E	Analog	E		<b>E</b>
10	RA3			In	n/a		Analog	E .		П
11	RB4		SOSCI	n/a	n/a		Analog			
12	RA4		SOSCO	n/a	n/a		Analog	E .		Г
13	VDD	5V		In	n/a		Digital			
14	RB5	5V		In	n/a		Digital			П
15	RB6	5V		In	n/a	Ē	Digital			
16	RB7	5V		In	n/a		Digital	Ē		
17	RB8	5V		In	n/a		Digital			
18	RB9	5V		In	n/a		Digital			E D
19	VSS	5V		In	n/a		Digital			
20	VCAP	5V		In	n/a	D	Digital			
21	RB10	5V		In	n/a	Ē	Digital			
22	RB11	5V		In	n/a		Digital	E .		Г

#### Pin Names

This row indicates the currently selected function for each pin. If no function is selected, the default pin name is shown instead.

Output Pin Table X																													
Package: SOIC 💌	Pin Settings	MCLR	RAO	RAI	RBO	RB1	RB2	RB3	VSS	RA2	RA3	IDSOS	SOSCO	QQA	RBS	RB6	RB7	RBS	R89	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI											Ê																	
(OSC_ID_0)	SOSCO												Ê																
Dahar	PGED 1				D																								
Debug	PGEC1					Ð																							

#### **Pin Numbers**

This row indicates the number of each pin in the table.

Output Pin Table ×																													
Package: SOIC	Pin Settings	MCLR	RAO	RAI	R80	RB1	RB2	RB3	VSS	RA2	RA3	SOSCI	SOSCO	DDV	RBS	RB6	RB7	RBS	RB9	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI																												
(OSC_ID_0)	SOSCO												Ê																
Dahua	PGED 1				D																								
Debug	PGEC1					Đ																							

#### **Table Modules**

This column contains the modules, or groups of functions, for the current configuration. These modules are controlled by the MHC configuration tree.

Output Pin Table X																													
Package: SOIC 💌	Pin Settings	MCLR	RAD	RAI	RBO	RB1	RB2	RB3	VSS	RA2	RA3	SOSCI	SOSCO	QQA	RBS	RB6	RB7	RBS	RB9	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI																												
(OSC_ID_0)	SOSCO												•																
D.h.s	PGED 1				D																								
Debug	PGEC1					Ð																							

#### **Table Functions**

This column displays the functions that belong to each module.

Output Pin Table ×																													
Package: SOIC 💌	Pin Settings	MCLR	RAO	RAI	RBO	RB1	RB2	RB3	VSS	RA2	RA3	IDSOS	SOSCO	DD	RBS	RB6	RB7	RBB	RB9	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI											Ê																	
(OSC_ID_0)	SOSCO												•																
Dahua	PGED 1				D																								
Debug	PGEC1					D																							

#### Table Grid

This area contains the grid cells. This area is for making connections between pins and functions.

#### Table Grid Cell Legend:

lcon	Description
	The cell is currently unavailable and cannot be selected.
	The cell is available for selection.
	The cell has been locked by the user.
D	The cell is a special debug indicator. This cell does not actually lock to a pin but is a visual debug reminder. This indicator means that the pin this cell resides on will be appropriated for debugging purposes based on the currently selected debug options.
	This cell has been automatically locked based on the available choices. This selection takes function priority into account. This lock cannot be changed by the user.

## **Module Management**

Describes the module management features.

#### Description

The Pin Manager table displays modules based on selections made in the configuration tree.

Observe that by enabling the USART driver instance that the USART1 module appears in the pin table.



Now increase the number of USART driver instances to 2. Once the second USART instance is set to USART\_ID\_2, the table will display the second USART module.



The U1RX, U1TX, U2RX, and U2TX functions are Peripheral Pin Select functions and can be assigned to multiple pins. Blue cells indicate a potential pin-to-function lock. Observe that left-clicking the blue cell corresponding to pin 9 and U1RX locks that cell to that pin/function pair. U1RX is now assigned to pin 9. Observe also that the name above pin 9 has changed to indicate the locked function, as well as the name of pin 9 in the pin diagram.



The green cell can be left-clicked again to unlock the pin and function.

## **Conflict Resolution**

Describes conflict resolution features.

## Description

The Pin Manager uses automatic conflict resolution to determine the proper function when multiple options are available.

Consider the available functions for pin 12: SOSCO/RPA4/T1CK/CTED9/PMA1/RA4. Observe that the SOSCO function was given automatic priority over RPA4 (U1RX).

Output Pin Table ×																													
Package: SOIC 💌	Pin Settings	MCLR	R.A0	RAI	RBO	RB1	RB2	RB3	VSS	RA2	RA3	D202	SOSCO	QQA	RB5	RB6	RB7	RB8	RB9	VSS	VCAP	RB10	RB11	RB12	RB13	RB14	RB15	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI											Ê																	
(OSC_ID_0)	SOSCO												Ê																
Dahua	PGED 1				D																								
Debug	PGEC 1					D																							
UART 1	U1RX																												
(USART_ID_1)	U1TX																												

The output window displays a detailed message of this event.

		MCLR	CTED1	CTED2	RBO	CTED12	CTED13	RB3	VSS	RA2	RA3	SOSCI	SOSCO	VDD	RBS	RB6	CTED3	CTED10	CTED4	VSS	VCAP	CTED11	RB11	RB12	CTPLS	CTED5	CTED6	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
	CTED1		Ê																										
	CTED2			Ê																									
	CTED3																Ê												
	CTED4																		Ê										
	CTED5																									Ê			
СТМИ	CTED6											_		-													Ê		
(CTMU_ID_0)	CTED9																												
	CTED 10																	Ê											
	CTED11																					Ê							
	CTED12					Ê																							
	CTED13						Ê																						
	CTPLS																								Ê				
Clock	SOSCI											Ê		-															
(OSC_ID_0)	SOSCO												Ê																
Dahua	PGED 1				D																								
Debug	PGEC1					D																							
UART 1	U1RX																												
(USART_ID_1)	U1TX																												

Observe also that with the addition of another lower priority function that the selection does not change. The higher priority function SOSCO (red) is still automatically selected while lower priority functions RPA4 (PPS) and OC1 are disabled.

		MCLR	CTED1	CTED2	RBO	CTED12	CTED13	RB3	VSS	RA2	RA3	IDSOS	SOSCO	NDD	RBS	RB6	CTED3	CTED10	CTED4	VSS	VCAP	CTED11	RB11	RB12	CTPLS	CTED5	CTED6	AVSS	AVDD
Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
	CTED1		Ê																										
	CTED2			Ê																									
	CTED3																Ê												
	CTED4																		Ê										
	CTED5																									Ê			
CTMU	CTED6																										Ê		
(CTMU_ID_0)	CTED9																												
	CTED 10																	Ê											
	CTED11																					Ê							
	CTED12					Ê																							
	CTED13						Ê																						
	CTPLS																								Ê				
Clock	SOSCI											Ê																	
(OSC_ID_0)	SOSCO												Ê																
	PGED1				D							Ľ																	
Debug	PGEC1					D																							
UART 1	U1RX																												
(USART_ID_1)	U1TX																												

If the highest priority is a Peripheral Pin Select function (red highlight) a choice is given to the user. The next lowest priority function is automatically selected (blue highlight), but this can be overridden by user action.

Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
	CTED1		Ê																										
	CTED2			Ê																									
	CTED3																Ê												
	CTED4																		Ê										
	CTED5																									Ê			
CTMU	CTED6													_													Ê		
(CTMU_ID_0)	CTED9												Ê																
	CTED 10													-				Ê											
	CTED11																					Ê							
	CTED12					Ê																							
	CTED13						Ê																						
	CTPLS																								Ê				
Debug	PGED1				D																								
Debug	PGEC1					D																							
UART 1	U1RX																												
(USART_ID_1)	U1TX																												

If the Peripheral Pin Select function (red highlight) is manually selected then the automatic choice (blue highlight) is overridden. A conflict is still reported. If the Peripheral Pin Select function is unlocked then the lower priority function will be automatically locked again.

#### Pin Table Features

Describes pin table features.

#### Description

The Pin Table can be reconfigured to show as little or as much information as the user desires. For example, individual pin rows can be hidden or

isolated depending on how much information is desired. This is accomplished by right-clicking on a pin number and selecting a desired option from the context menu.

Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Clock	SOSCI											Ê																	
(OSC_ID_0)	SOSCO												Ê																
Debus	PGED 1				D																								
Debug	PGEC1					D																							

To remove pin 18 from the table, right-click the pin 18 number box. Select Hide from the context menu.

Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
	CTED1		Ê																	Iso	e late									
	CTED2			Ê																Flag	gs									
	CTED3																			Vie	w 🕨									
	CTED4																													
	CTED5																									Ê				
CTMU	CTED6																										Ê			
(CTMU_ID_0)	CTED9																													
	CTED 10																													
	CTED11																													
	CTED12					Ê																								-
Module	Eurotion	1	2	3	4	5	5	7	9	0	10	11	12	12	14	15	16	17	10	20	21	22	22	74	25	26	27	28		
module	orrep 4	-	2	5	-	5	-	-		-	10		12	15	14	15	10	17	15	20	21	22	25	27	25	20	21	20		
	CIED1	<u> </u>	-	0		<u> </u>						<u> </u>	<u> </u>		<u> </u>	<u> </u>								<u> </u>						
	CTED2																													
	CTED3																													
	CTED4																													
	CTED5																								Ê					
CTMU	CTED6																									Ê				
(CTMU_ID_0)	CTED9																													
	CTED 10																													
	CTED11																													
	CTED12					Ê																							-1	

Observe that pin 18 has been removed from the table. To restore the column, right-click in the table and select **Show > All** or navigate the available sub-menus and select pin 18.

Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	19	20	21	22	23	24	25	26	27	28	
	CTED1		Ê																									-	-
	CTED2			Ê												_													
	CTED3													\\	/iew l														
	CTED4														Show I		All Pips 1		18										
	CTED5																								Ê				
CTMU	CTED6																									Ê			
(CTMU_ID_0)	CTED9																												
	CTED 10																												
	CTED11																												
	CTED12					Ê																							<b>–</b>
							<b></b>																						

The table can also be reduced to show only desired pins and functions by using the "Isolate" command. To show only pin 18, again right-click on the pin 18 number box and select **Isolate**.

Module	Function	18
CTMU (CTMU ID 0)	CTED4	
PMP (PMP ID 0)	PMD3	Ê
UART 2 (USART ID 2)	U2TX	

This functionality also exists for pin modules, functions, and ports.

Module	Function	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
	PMD2																					Ê								<u> </u>
PMP	PMD3																		Ê											
(PMP_ID_0)	PMD4																	Ê												
	PMD5																Ê													
	PMD6															Ê														
	PMD7														Ê															
UAR	U1RX																													
(USART Hide Isolate	U1TX																													
UAR View	U2RX																													
(USART_ID_Z)	U2TX																													-
						N 1	0																							
		MCLR	CTED1	CTED2	RBO	CTED1	CTED1	RB3	VSS	RA2	RA3	SOSCI	Sosco	QQ	PMD7	PMD6	PMD5	PMD4	PMD3	VSS	VCAP	PMD2	PMD1	PMD0	CTPLS	CTEDS	CTED6	AVSS	AVDD	
Module	Function	MCLR 1	CTED1	CTED2	088 4	CTED1	o CTED1	ଞ୍ଚ 7	o VSS	o RA2	EA3 10	IDSOS 11	00505 12	00A 13	ZQWd 14	90JWd 15	SQWd 16	*0Wd 17	EQIMA 18	SSA 19	AD 20	ZQWd 21	I QWd 22	OQWd 23	CTPLS 54	SGELD 25	90 26	SSVA 22	DQAA 28	
Module	Function PMD2	MCLR 1	2 CTED1	6 CTED2	093 4	CTED1	o CTED1	893 7	on VSS	6 RA2	EA 10	11 SOSCI	12	00A 13	LOWI 14	90Wd 15	SOMA 16	40Wd 17	EQWd 18	SSA 19	ACAP 20	20Wd 21	IQWd 22	23	24	25	903 26	SSAM 27	DOAN 28	
Module	Function PMD2 PMD3	MCLR	2 CTEDI	cTED2	082 4	2 CTED1	o CTED1	88 7	o ASS	6 RA2	EW2 10	DSOS 11	12	ад 13	LOWA 14	90Wd 15	SQWd 16	*QWd 17	EQW4 18	SSA 19	20	21	10Wd 22	23	24	25	26	SSAN 27	28	
Module PMP (PMP_ID_0)	Function PMD2 PMD3 PMD4	MCLR	2 CTED1	cted2	982 4	cTED1	o CTED1	283 7	ss vss	6 RA2	EW 10	11	12	90 13	LOWA 14	90Wd 15	SOWd 16	*Gwd 17	EQWd 18	SSA 19	20 20	21	IQWA 22	23	24	25	26	SSNY 27	28	-
PMP (PMP_ID_0)	Function PMD2 PMD3 PMD4 PMD5	1 MCLR	2 2	cTED2	082 4	cTED1	o CTED1	282 7	SSA 00	6 RA2	EW2 10	DSOS 11	12	00 13	LOW4	90Wd 15	SOWd 16	POWd 17	18 18	IP	20 20	21	10Wd 22	23	24	25	26	SSAN 27	28	•
Module PMP (PMP_ID_0)	Function PMD2 PMD3 PMD4 PMD5 PMD6	1 MCLR	2 2	CTED2	982 4	s CTED1	o CTED1	88 7	R ASS	L RAZ	10	11	12	13 13	20Wd 14	90Wd 15	SQWd 16	*dwd 17	EQWA 18	SSA 19	20 20	21	10Wd 222	23	24	25	26	SSAN	28	
Module PMP (PMP_ID_0)	Function PMD2 PMD3 PMD4 PMD5 PMD6 PMD7		2	CTED2	982 4	cteb1	o CTED1	882 7	SSA 00	6 RA2	E83	11	12	90 13	20Wd	90Wd 15	SOWA 16	40Wd	18	SSA 19	20 20	21	10Wd 22	23	24	25	90310 26	SSAY 27	28	
Module PMP (PMP_ID_0) UART 1	Function PMD2 PMD3 PMD4 PMD5 PMD6 PMD7		2 CTED1	CTED2	4	CTED1	OTED1	282 7 	SSA 8	6 RA2	EW2 10				LOWd 14	90W4 15		17	18 18	SSA 19	20	21	10Wd 22	23 23	24	25	26	SSAN 27	28	
Module PMP (PMP_ID_0) UART 1 (USART_ID_1)	Function PMD2 PMD3 PMD4 PMD5 PMD6 PMD7 UMD7 UMD7 UMD7 UMD7	MCLR	2 2 2	CTED2	4	CTED1	O CTED1	88           7	SSA 8	6 RA2	EV3 10				20Wd	90 <u>Wa</u> 15 		*0Wd 17	18	SSA 19	CAP CAP	21	10Wd 22	23 23 	24		26	SSAW 27	28	
Module PMP (PMP_ID_0) UART 1 (USART_ID_1) UART 2	Function PMD2 PMD3 PMD4 PMD5 PMD6 PMD7 L Hide Isolate L View			CTED2	082 4	CTED1	OTED1	E82 7	SSA 8	6 RA2	EW3				20Wd 14	990Wd 15	SQWA 16	+GWd 17		SSA 19	20 20	21		23	24 24 24			SSAN 27	28	

The table can also be modified by right-clicking the pin boxes in the pin diagram.

MCLR RA0			28 27		AVDD AVSS
RAI DDO	8	<u>ب</u>	· 20	E.	DD14
RBU	4	ć –	25		KD14
UĮ2RX			-74		RB13
RB2	E Hid	le	3		RB12
U IX	Isc	olate	2		RB11
1100	E .		- ka -		DD10
422	L Fla	gs	11		KDIU
RA2	L Fla	igs	0		VCAP
RA2 RA3		<b>igs</b> 101	0 19		VCAP VSS
RA2 RA3 SOSCI	Fla	<mark>9</mark> =016	19 18		VCAP VSS U2TX
RA2 RA3 SOSCI SOS <sup>2</sup> 10	Fla	<mark>8</mark> <sup>=</sup> 016B	19 19 18 17		VCAP VSS U2TX RB8
RA2 RA3 SOSCI SOS <sup>-</sup> LO DD	Fla 10 11 12 13	<mark>≌</mark> =016B	19 19 18 17 16		VCAP VSS U2TX RB8 RB7
V55 RA2 RA3 SOSCI SOSCIO SOSCIO TD 35	Fla	<mark>≗</mark> =016B	19 19 18 17 16 15		VCAP VSS U2TX RB8 RB7 U1RX

The table can also be reconfigured to display pins according to their respective ports. To do this, right-click the table, navigate to the View sub-menu, and select **Ports**. The top row is the original pin number, the middle row shows the port grouping, and the bottom row is the pin's number inside the port grouping. Ports can also be hidden and isolated in the same manner as pins, modules, and functions. This is accomplished by right-clicking on the port name box.

		2	3	9	10	12	4	5	6	7	11	14	15	16	17	18	21	22	23	24	25	26
				Α										E	3							
Module	Function	0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Clock	SOSCI										Ê											
(OSC_ID_0)	SOSCO					Ê																
Dahua	PGED1						D															
Debug	PGEC1							D														
UART 1	U1RX																					
(USART_ID_1)	U1TX																					
UART 2 (USART_ID_2)	U2RX																					
	U2TX																					

#### Change Notification and Non-PPS Devices

Describes handling change notification for non-PPS devices.

#### **Description**

For non PPS parts, change notifications behave differently. They must be explicitly enabled in the configuration tree.



When enabled, the Change Notification module appears in the table. Change notification cells behave similarly to Peripheral Pin Select functions. They will be overridden by higher priority functions, but will provide a user choice if they are the highest priority.

The pin flag dialog also behaves differently for Non-PPS parts. The "Change Notification", "Pull Up", and "Pull Down" options are disabled.

📴 Pin Se	etting Configu	iration								
Pin	Name	Voltage Tolerance	Function	Direction (TRIS)	Latch (LAT)	Open Drain (ODC)	Mode (ADPCFG)	Change Notification (CNEN)	Pull Up (CNPUE)	Pull Down (CNPD)
1	RG15	5V		In	n/a		Digital	E	E.	E .
2	VDD	5V		In	in/a		Digital	E .	1	Π
3	RE5	5V	-	In	m/a		Digital	F	П	E
4	RE6	5V		In	n/a		Digital		Π	E
5	RE7	5V		In	n/a		Digital	E	III.	E
6	RC1	5V		In	n/a	<b>D</b>	Digital	E I		П
7	RC2	5V		In	n/a	Ē	Digital	E .	Π	Π
8	RC3	5V		In	n/a		Digital	E	Π	Π
9	RC4	5V		In	n/a		Digital	E	E	E
10	RG6	5V		In	n/a		Digital	E C	1	П
11	RG7	5V		In	n/a		Digital	E	П	Г
12	RG8	5V		In	n/a		Digital		Π	
13	MCLR	5V		In	n/a		Digital		III.	П
14	RG9	5V		In	n/a		Digital	E C		Π
15	VSS	5V		In	n/a		Digital	, E	П	Г
16	VDD	5V		In	n/a		Digital	E	П	•
17	RAO	5V		In	n/a		Digital	E	П	Π
18	RE8	5V		In	in/a		Digital	E C		Π
19	RE9	5V		In	n/a		Digital	Γ	П	Г
20	RB5			In	n/a	Π	Analog		Π	Π
21	RB4			In	n/a	É.	Analog	E	П	Г
22	RB3			In	n/a		Analog	F		Π

## **Exporting Pin Mapping**

Provides information on exporting pin mappings.

## Description

The MPLAB Harmony Graphical Pin Manager provides the ability to export the pin mapping of the current configuration into Excel in .xls format for the purpose of printing out the pin mapping. Refer to Importing and Exporting Data for the steps to export the pin mapping.

## Importing and Exporting Data

Provides information on importing and exporting data to/from the MHC.

#### Description

The MPLAB Harmony Configurator provides several options for importing and export various types of data to and from the application. The import and export icons can be found in the main window toolbar.

MPLAB®H	larmony Co	nfigurator* ×			
1 🖪	30	Code	€‡	 69	

The Import dialog shows the various data sources that can be imported into MPLAB Harmony Configurator. To import, select an item from the list and click **Import**.



The Export dialog shows the various data sources that can be exported from MPLAB Harmony Configurator. To export, select an item from the list and click **Export**.

Export Menu	×
Select an exporter from the provided list: MPLAB Harmony & Application Configuration Options MPLAB Harmony Graphics Composer	
	Export Cancel

## Importing and Exporting MPLAB Harmony Configurator Configuration Options

By selecting **MPLAB Harmony & Application Configuration Options** from either the Import or Export dialog, the user has the ability to create or import .mhc files with only user-selected options.

The following figure provides an example of the option export dialog.

Export MPLAB Harmony & Application Configuration Optio	ns	×
Select, by dicking, the configuration items to export: MPLAB Harmony & Application Configuration Application O Configuration Exception Handling Harmony Framework Configuration Drivers Operating System Abstraction Layer (OSAL) Peripheral Library System Services Cock Common Device Control Ports Device & Project Configuration Device Configuration		
Current Export Configuration Path		
C: Wicrochip \harmony \local \software \isp_root \apps \example \firmware	e\src\system_config\default\export_defau	lt.mhc
	Save	Save As

To use this feature, left-click any desired option to toggle its state. Green-highlighted options will be exported. Then, use the **Save** and **Save As** buttons as desired to write the file.

To import, select the option import from the Import dialog and select the previously exported file. Observe that only the exported options are visible in the import window. The user can again select and highlight items in green to select them for import. When all desired settings have been highlighted, click **Import**.

Import MPLAB Harmony & Application Configuration Options	X
Select, by clicking, the configuration items to import:	
MPLAB Harmony & Application Configuration Application Configuration Exception Handling Harmony Framework Configuration Operating System Abstraction Layer (OSAL) System Services Device & Project Configuration Device & Project Configuration PIC32MX110F0168 Device Configuration DEVCFG2 DEVCFG2 DEVCFG1 DEVCFG0	
Current Import Configuration Path	
C: \Microchip \harmony \local \software \\sp_root \apps \example \firmware \src \system_config \default \export_de	fault.mhc
	Import

## Index

## С

Change Notification and Non-PPS Devices 45 Clock Configuration for PIC32MX Family Devices 22 Clock Configuration for PIC32MZ Family Devices 15 Configuring the Oscillator Module Using the MHC Clock Configurator 15 Configuring the Peripheral Bus Clock 27 Configuring the Peripheral Bus Clocks 19 Configuring the Reference Clock 27 Configuring the Reference Clocks 19 Configuring the System Clock Frequency 16, 24 Configuring the USB PLL 29 Conflict Resolution 40

## Е

Exporting Pin Mapping 46

## I

Importing and Exporting Data 47 Installing MHC 3 Introduction 10

## L

Launching the Tool 31

## Μ

Module Management 37 MPLAB Harmony Configurator Interface 5 MPLAB Harmony Configurator User's Guide 2 MPLAB Harmony Graphical Pin Manager 31

## Ρ

Pin Diagram Tab 33 Pin Table Features 42 Pin Table Tab 35 Porting a Legacy PLIB to MPLAB Harmony 14 Prerequisites 10

## S

Step 1: Create the New Project 10 Step 2: Add and Configure the Required Libraries and Modules 12 Step 3: MPLAB Harmony Application Structure and Developing the Application 12

## U

Using MHC to Create a New Application 10 Using the Reference Clock Auto-Calculate Feature 20, 28 Using the SPLL Divider Auto-Calculate Feature 21, 30