



# Getting Started with PSF

Information contained in this publication regarding device applications and the like is provided only for your convenience and can be superseded by updates.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.

**MICROCHIP**  
Microchip Technology, Inc.Microchip Technology, Incorporated  
2355 W. Chandler Boulevard  
Chandler, Arizona 85224  
480/792-7416

REV	DATE	DESCRIPTION OF CHANGE
0.92	02-Dec-19	Initial revision

## Table of Contents

1	Introduction .....	5
2	Terms and Abbreviations .....	5
3	Software License Agreement .....	5
4	PSF Overview.....	6
5	Prerequisites for PSF .....	6
6	Building the Project.....	7
7	Programming the PSF-EVB board .....	8
8	Appendix .....	10
8.1	Unlicensed Compiler Usage .....	10

## 1 Introduction

USB Power Delivery Software Framework (PSF) is an open source Power Delivery stack designed to be integrated with any suitable powerful MCU to control UPD350 PD controllers within a USB Type-C Power Delivery System.

This document gives an overview of PSF and it's features along with software and hardware requirements needed for the PSF Firmware to work properly with PSF-EVB. It also provides information on how to build the Firmware and program the hex file in the PSF-EVB.

## 2 Terms and Abbreviations

Term	Definition
USB	Universal Serial Bus
PSF	USB Power Delivery Software Framework
EVB	Engineering Validation Board
PD	Power Delivery
MCU	Micro Controller Unit
IDE	Integrated Development Environment
IPE	Integrated Programming Environment
PDO	Power Data Object
Building	Process of converting source code files into standalone software artifact that can be run on a computer

## 3 Software License Agreement

Copyright ©[2019] Microchip Technology Inc. and its subsidiaries.

Subject to your compliance with these terms, you may use Microchip software and any derivatives exclusively with Microchip products. It is your responsibility to comply with third party license terms applicable to your use of third party software (including open source software) that may accompany Microchip software.

THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

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

## 4 PSF Overview

- USB Power Delivery Software Framework (PSF) with USB-PD Port Controller UPD350 is an effective USB-PD solution compliant to USB-PD 3.0 Specification.
- PSF stack is designed to run on different MCU Hardware platform. Versatility towards different HW platform is achieved through flexibility towards configurability of PSF stack.
- PSF stack is compliant to USB Power Delivery 3.0 & Type-C specification V1.3
- PSF is highly portable.
- PSF supports multiport solutions.

### PSF User Guide

- ‘PSF User Guide’ gives detailed overview of PSF stack architecture, directory structure, Supported and Not Supported messages, various configuration options that are available in PSF.
- It also touches upon the requirements and steps for integration of PSF to new platform.
- PSF User Guide is available at [https://bitbucket.microchip.com/projects/UNG\\_APPS/repos/usb-pd-software-framework-public/browse/Docs](https://bitbucket.microchip.com/projects/UNG_APPS/repos/usb-pd-software-framework-public/browse/Docs)

## 5 Prerequisites for PSF

Following are the prerequisites that are needed for PSF to work with a Microchip MCU. In order to port PSF to any non-Microchip MCU, respective IDE, compiler and other software tools shall be used.

- [MPLAB X IDE v5.30](#) or later
- [MPLAB XC32 compiler](#)
- [Atmel ICE](#)

Follow these steps for setting up the build environment needed for PSF Firmware.

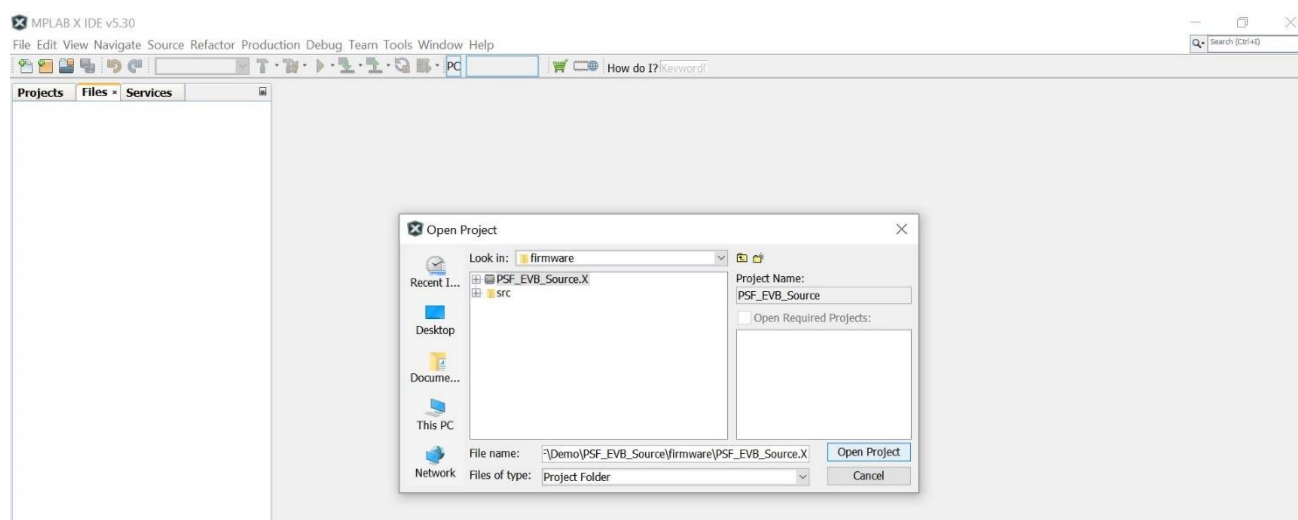
1. MPLAB X IDE can be downloaded and installed from *Downloads* tab of <https://www.microchip.com/mplab/mplab-x-ide>  
For any IDE specific information, refer the MPLAB X IDE Release Notes and User Guide of the installed version listed in the same page.
2. MPLAB XC32/32++ Compiler can be downloaded and installed from *Downloads* section of <https://www.microchip.com/mplab/compilers>

For any compiler specific information, refer the *Compiler User's Guide for PIC32C/SAM MCUs* listed in the *XC32 Documents* tab of the same page.

## 6 Building the Project

Follow these steps for building the PSF Project and generating the hex file. Building the project is demonstrated here by having PSF\_EVB\_Source project as an example.

1. Download PSF Firmware from [https://bitbucket.microchip.com/projects/UNG\\_APPS/repos/usb-pd-software-framework-public/browse/PSF](https://bitbucket.microchip.com/projects/UNG_APPS/repos/usb-pd-software-framework-public/browse/PSF) All the sub folders must be retrieved for the project to build successfully.
2. The entire local folder path containing the source code shall have no space in the folder names. For ex : Instead of `C:\Users\Desktop\PSF FW\PSF\PSF`, `C:\Users\Desktop\PSF_FW\PSF\PSF` shall be used.
3. Remove Read-Only option for the folder by right click -> Properties -> Uncheck Read Only -> Apply -> OK
4. Open MPLAB X IDE and click File -> Open Project. Enter the path {Local\_Folder}\PSF\Demo\PSF\_EVB\_Source\firmware which contains PSF\_EVB\_Source.X



5. The IDE may throw a configuration load error if the compiler toolchain is not properly linked to the project. With this error, we can't proceed with building of the project.

*warning: Configuration "default" builds with "XC32", but indicates no toolchain directory.*

*error: Configuration "default" builds with "XC32", but no toolchains of that type are installed. Errors have occurred while loading one or more configurations.*

*If a specific error is not shown above, this may happen when you import a project from another computer.*

+ You can add language tools in Tools->Options embedded tab.

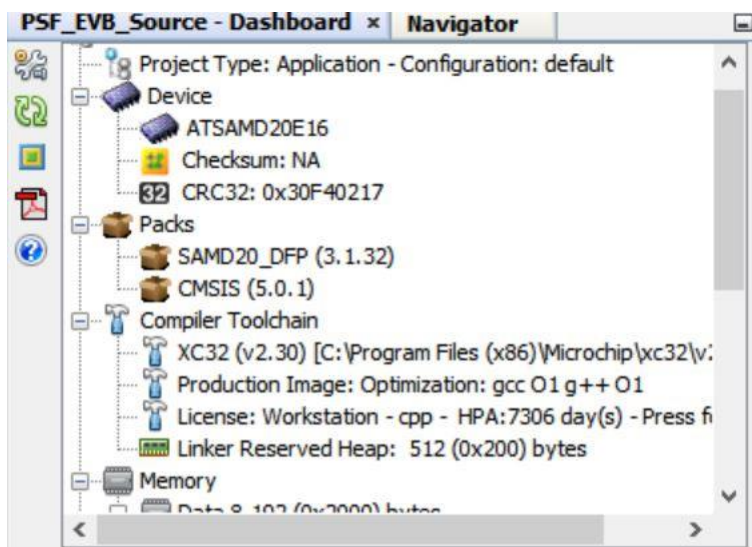
+ You can change which language tool to use in the project properties dialog.

To resolve this, click Tools -> Options -> Embedded -> Build Tools -> Scan for Build Tools

<p>THIS DOCUMENT IS UNCONTROLLED UNLESS OTHERWISE STAMPED. It is the user's responsibility to ensure this is the latest revision prior to using or referencing this document.</p> <p>© Microchip Technology Inc. <b>CONFIDENTIAL AND PROPRIETARY</b></p>	<p><b>Page</b></p> <p>7 of 11</p>	<p><b>REV</b></p> <p>0.92</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------	-------------------------------

Once done, all the compiler toolchains installed in your PC should be listed under Toolchain.

6. Choose the XC32 compiler installed by you and click Apply -> OK Once done, Compiler Toolchain will be properly listed in the project Dashboard as shown in the figure.



7. Make the PSF\_EVB\_Source as the main project by right click -> *Set as Main Project*



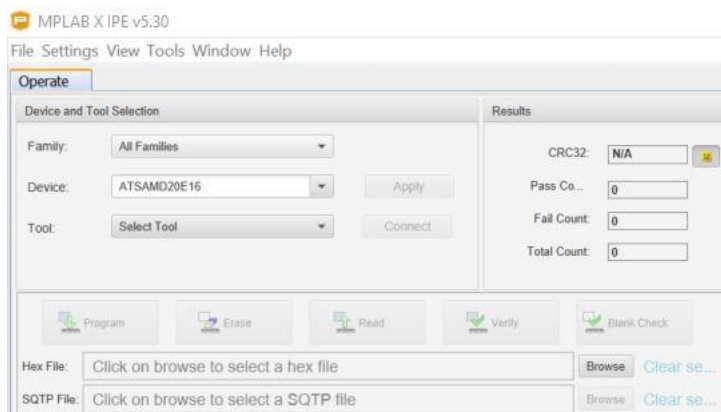
8. Build the project by right click -> *Clean and Build* or use the icon at the top of IDE to build the firmware.
9. Once the build completes, *Build Successful* message will be shown in the output window of the IDE.
10. Generated Hex file can be found under the path  
{Local\_Folder}\PSF\Demo\PSF\_EVB\_Source\firmware\PSF\_EVB\_Source.X\dist\default\production
11. Refer Appendix section 7.1 in case you are using an unlicensed XC32 compiler version.

## 7 Programming the PSF-EVB board

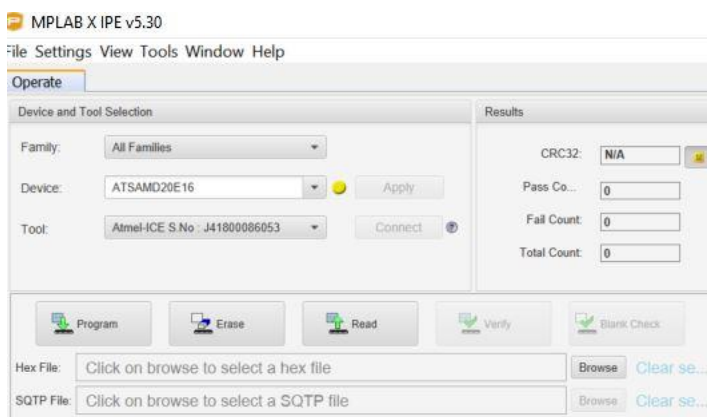
Follow these steps to program the generated hex file in the PSF-EVB

1. Open MPLAB IPE which would have got downloaded and installed during MPLAB IDE installation.





2. Select *Family* as 'All Families' and *Device* as 'ATSAMD20E16' and click *Apply*
3. Connect Atmel ICE Debugger to the PC. Connect the other end of debugger as mentioned in step 3 of section 3.
4. Once debugger is connected, it will be listed under *Tool* tab of the IPE and an yellow dot will be shown at the right side of the device.



5. Browse through the path of hex file.
6. Once hex file path is given, output window of IPE will display a message similar to this.

*Loading code from C:\Users\Downloads\usb-pd-software-framework-public-master@6c2c24923d9\PSF\Release\v0.91\PSF\_Hades\_Source\_V0.91.hex...  
2019-12-03 14:14:08 +0530 - Hex file loaded successfully.*

7. Power on the EVB and click *Program* in the IPE.
8. Once the hex file is programmed successfully, output window of the IPE will show the following message.

```

Output - IPE x
*****

Currently loaded versions:
Application version.....1.41.137 (0x01.0x29.0x89)
Target voltage detected

*****

Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x0, end address = 0x8bfff
configuration memory
Programming complete
2019-12-03 14:30:39 +0530 - Programming complete

```

## 8 Appendix

### 8.1 Unlicensed Compiler Usage

If the PSF firmware is built with unlicensed XC32 compiler, build errors may be thrown because of compiler optimization. In case you are using an unlicensed XC32 compiler and want the build process to be error free and successful, please follow these steps.

1. Set the PSF\_EVB\_Source project as main project.
2. Click File -> Project Properties -> XC32(Global Options)
3. Choose xc32-gcc and select *Optimization* under Option Categories.
4. Select optimization-level as 0.

