Getting Started with PSF

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

REV	DATE	ORIGINATOR	DESCRIPTION OF CHANGE
0.1	02-Dec-19	Ram Ganesh N	Initial revision

Table of Contents

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.

1.1 Terms and Abbreviations

Term	Definition
USB	Universal Serial Bus
PSF	USB Power Delivery Software Framework
EVB	Engineering Validation Board
PD	Power Delivery
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

2 Software License Agreement

3 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.

Following are the key features of PSF stack,

- Compliant to USB Power Delivery 3.0 & Type-C specification V1.3
- Highly Portable
- Supports multi-port solutions
- USB-PD Source-only or Sink-only port specific configurability
- FW update through CC support compliant to USB PD Firmware Update Specification R1.0
- I2C Based DC/DC Control Driver support
- Source-Only PPS Support
- Alt Mode Support with configurable role preferences
- Alt Mode mux control through GPIO or Configurable I2C driver

4 Setting up the build environment

Software Build Requirements needed for PSF are,

- MPLAB X IDE v5.20 or later
- MPLAB XC32 compiler
- MPLAB Harmony v3
- Atmel ICE

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

1. Download and install latest version of MPLAB X IDE 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. Download and install latest version of MPLAB XC32/32++ Compiler 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.

5 Building the Project

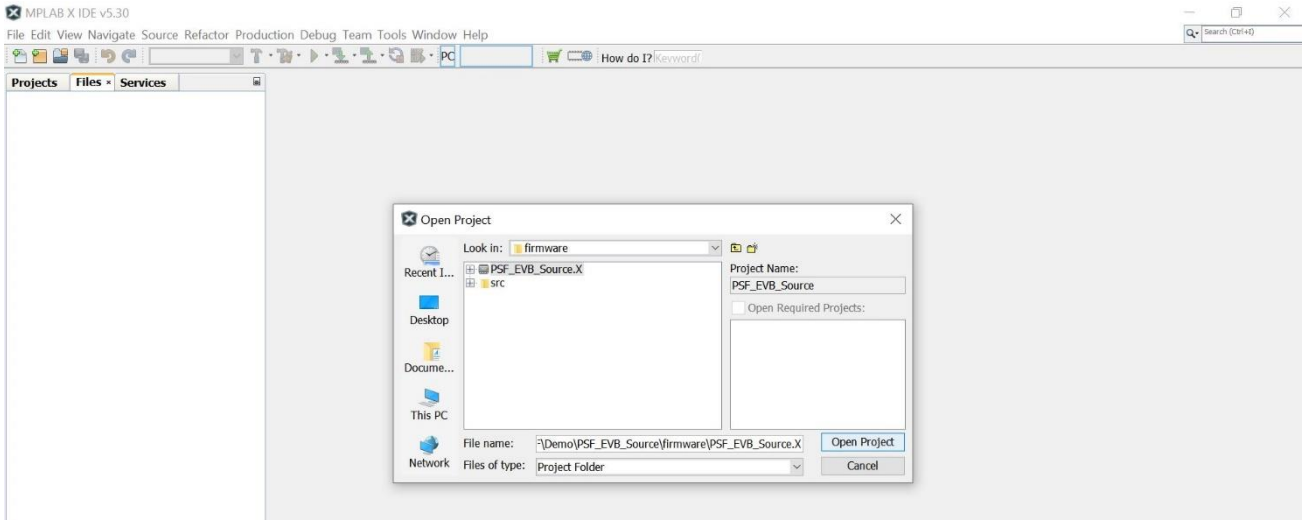
Follow these steps for building the PSF Project and generating the hex file.

1. Download PSF Firmware from https://bitbucket.microchip.com/projects/UNG_APPS/repos/usb-pd-software-framework-public/browse/PSF/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\I20019\Desktop\PSF FW\PSF\PSF*, *C:\Users\I20019\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\PSF\Demo\PSF_EVB_Source\firmware which contains PSF_EVB_Source.X

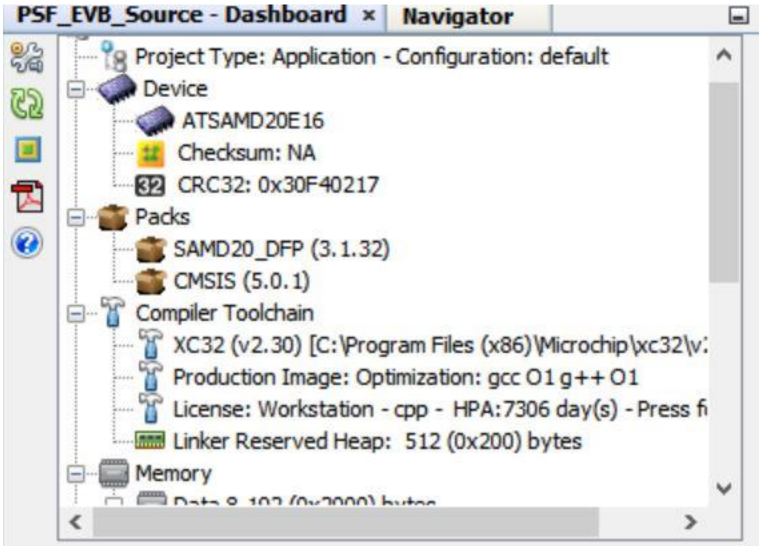


5. The IDE may throw a configuration load error if the compiler toolchain is not properly linked to the project. With this error, we can't proceed with building of the project.

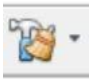
   warning: Configuration "default" builds with "XC32", but indicates no toolchain directory.
   error: Configuration "default" builds with "XC32", but no toolchains of that type are installed.
   Errors have occurred while loading one or more configurations.
   If a specific error is not shown above, this 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 Once done, all the compiler toolchains installed in your PC should be listed under Toolchain.

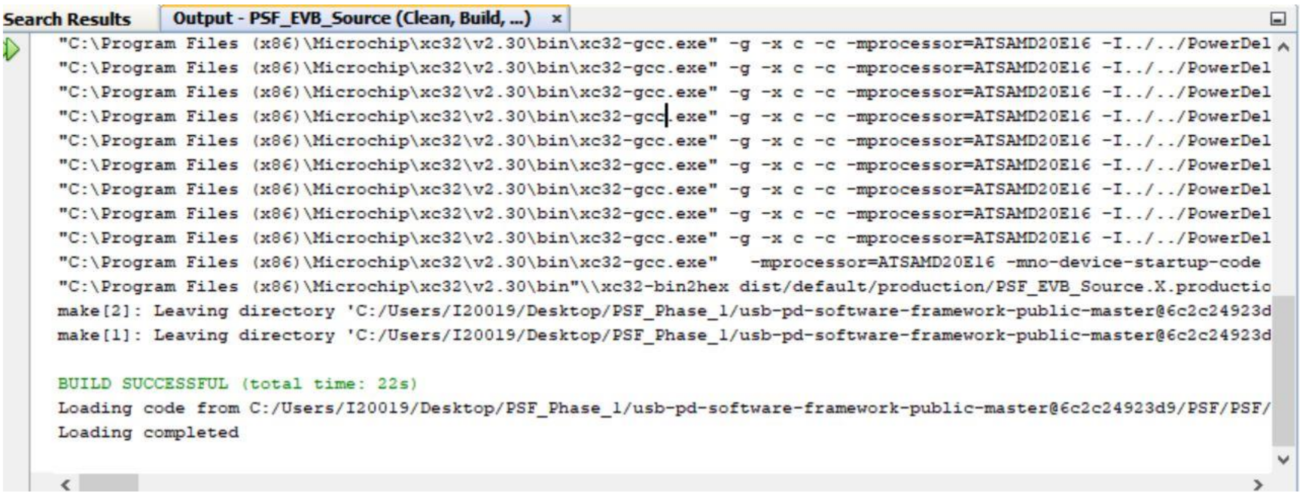
6. Choose the XC32 compiler installed by you and click Apply -> OK Once done, Compiler Toolchain will be properly listed in the project Dashboard as shown in the figure.



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  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.



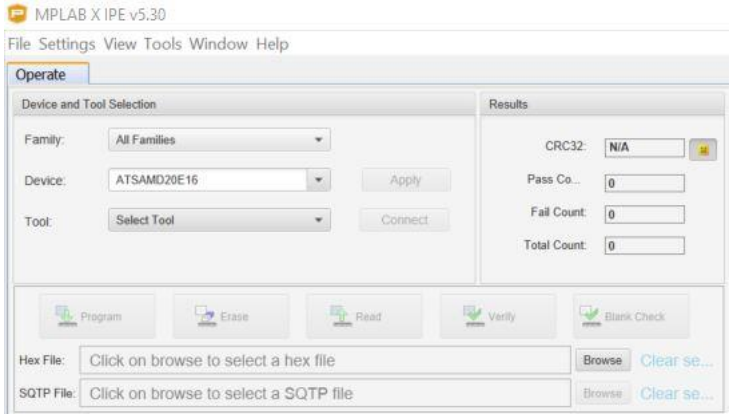
10. Generated Hex file can be found under the path {Local_Folder}\PSF\PSF\Demo\PSF_EVB_Source\firmware\PSF_EVB_Source.X\dist\default\production

11. Refer Appendix section 7.1 in case you are using an unlicensed XC32 compiler version.

6 Programming the PSF-EVB board

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

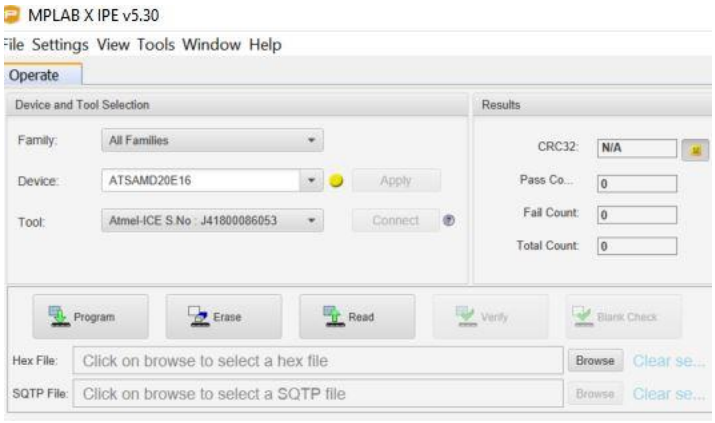
1. Open MPLAB IPE which would have got downloaded and installed during MPLAB IDE installation.



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.



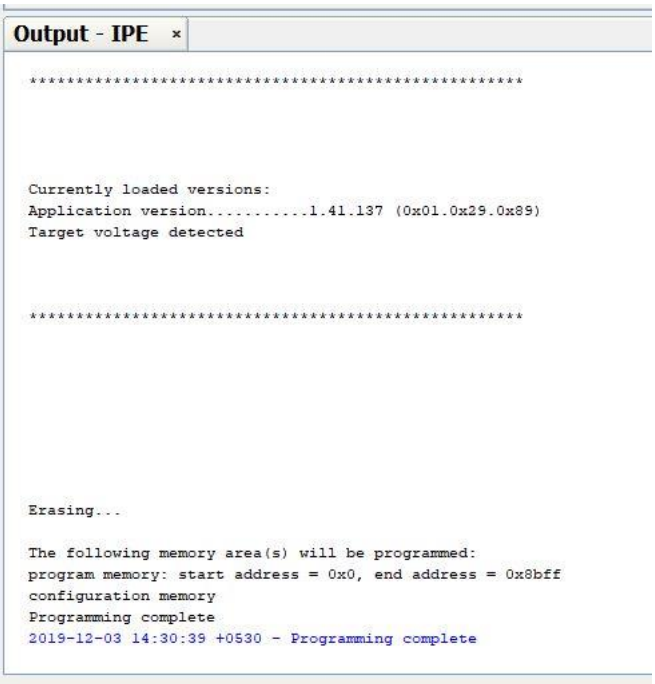
5. Browse through the path of hex file.

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

   Loading code from C:\Users\I20019\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.

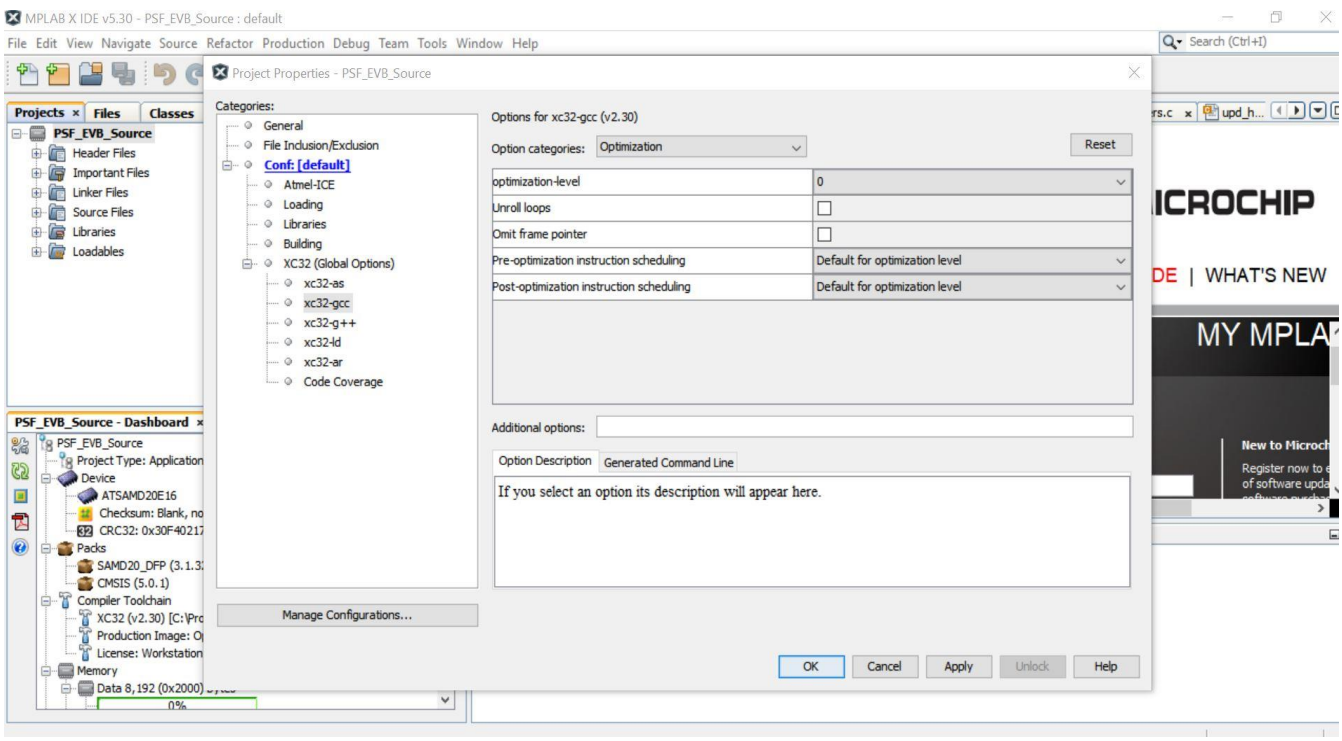


7 Appendix

7.1 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.


MPLAB X IDE v5.30 - PSF_EVB_Source : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
Search (Ctrl+I)
Project Properties - PSF_EVB_Source
Categories:
General
File Inclusion/Exclusion
Conf: [default]
Atmel-ICE
Loading
Libraries
Building
XC32 (Global Options)
xc32-as
xc32-gcc
xc32-g++
xc32-ld
xc32-ar
Code Coverage
Options for xc32-gcc (v2.30)
Option categories: Optimization
Reset
optimization-level 0
Unroll loops
Omit frame pointer
Pre-optimization instruction scheduling Default for optimization level
Post-optimization instruction scheduling Default for optimization level
Additional options:
Option Description
Generated Command Line
If you select an option its description will appear here.
Manage Configurations...
OK
Cancel
Apply
Unlock
Help
Projects
Files
Classes
PSF_EVB_Source
Header Files
Important Files
Linker Files
Source Files
Libraries
Loadables
PSF_EVB_Source - Dashboard
Project Type: Application
Device
ATSAMD20E16
Packs
Compiler Toolchain
Memory
MY MPLA
New to Microch