MPLAB® Harmony Setup for Software Development on PIC32CX_BZ2/WBZ45x

This document will show users how to **install MPLAB® Harmony Configurator (MHC)** and **how-to setup the MPLAB Harmony framework** for software development on PIC32CX_BZ2/WBZ45x module

MHC includes tool called the MPLAB Harmony 3 Content Manager that is used to download and update the Harmony 3 framework on local machine. Users should install MHC before downloading and setting up the framework

MHC is a GUI for the MPLAB Harmony framework. It dramatically simplifies 32-bit (SAM and PIC32) core and peripheral configuration. It also allows users to enable and configure any of the specific Harmony framework libraries (e.g., device drivers, middleware, USB, system services etc.)

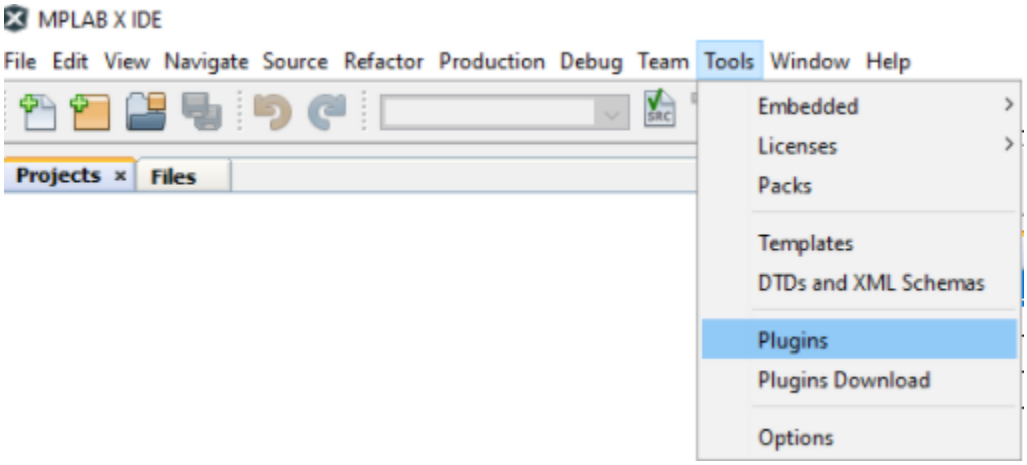
MHC is available as a plugin extension to the MPLAB® IDE

Users must install the following tools to use MHC with MPLAB®X IDE

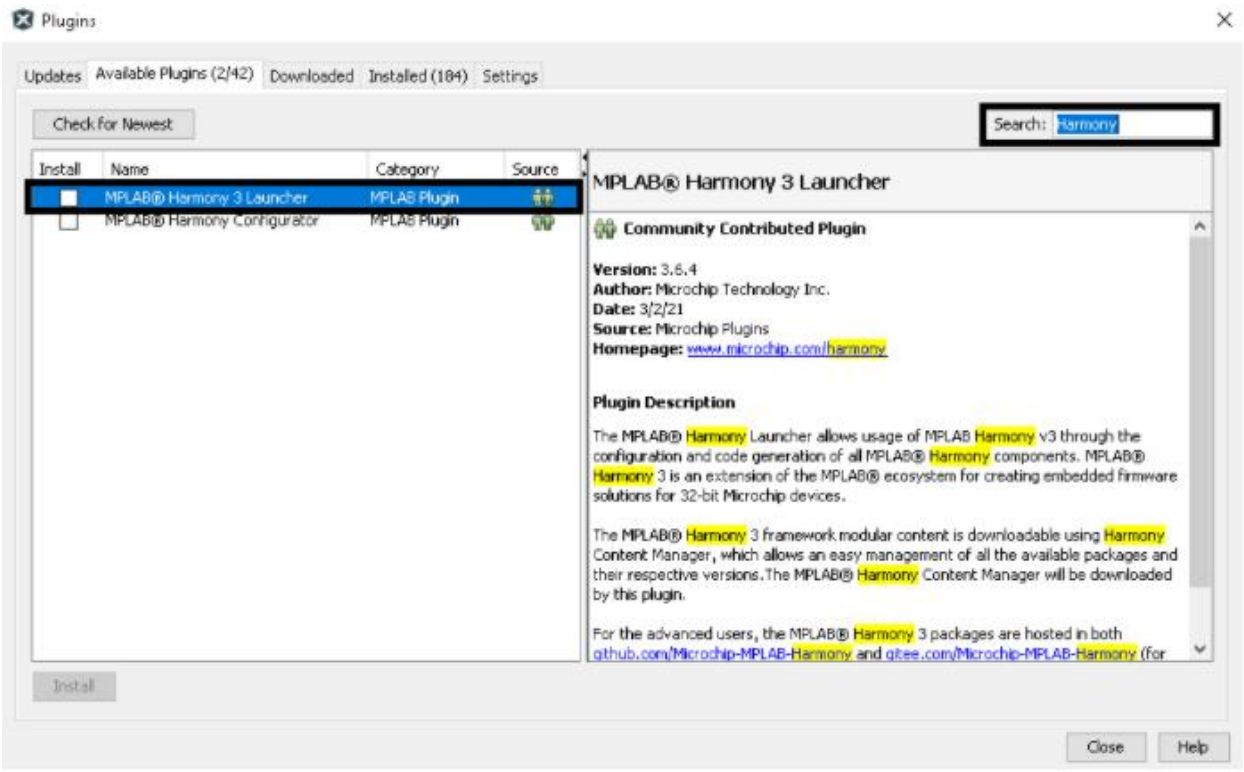
- MPLAB X IDE
- XC32 C compiler

1. Install MPLAB Harmony Configurator (MHC)

1.1 Open the MPLAB X IDE and select Tools > Plugins



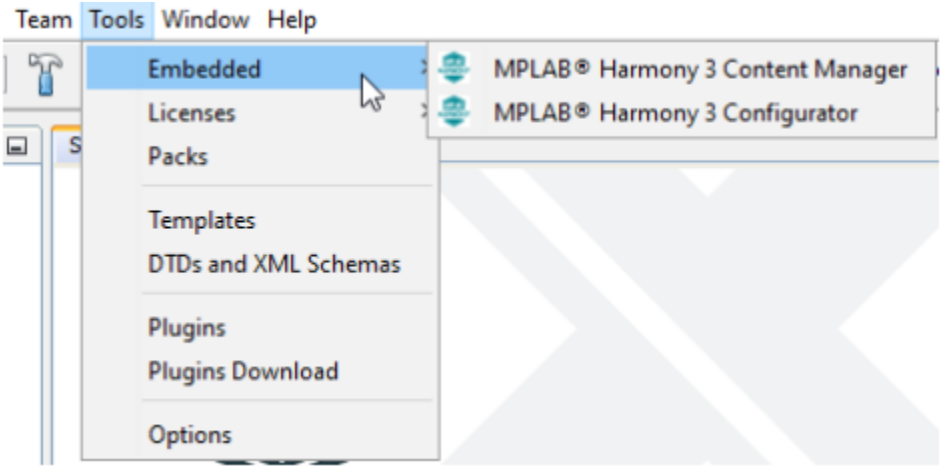
1.2 The Plugins window will open. Click on the **Available Plugins** tab, check the MPLAB® Harmony 3 Launcher box, and click the install button
Note: Users should not select MPLAB® Harmony Configurator as this is a different plugin used with previous version of Harmony (version 2)



1.3 Click **Next**, accept the license terms, and click **install**

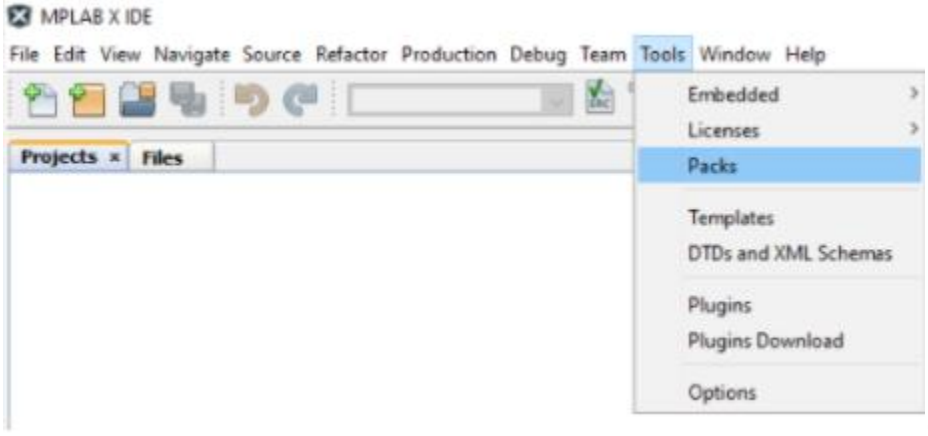
1.4 Click **Finish** to restart the IDE

1.5 Select **Tools > Embedded** to verify that MHC and the content manager are displayed in the list



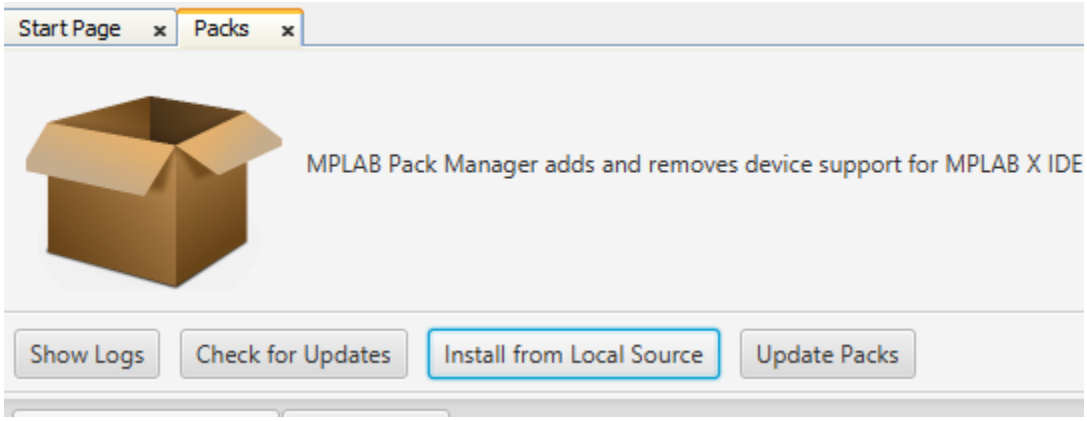
2. Install PIC32CX-BZ Device Family Pack (DFP) from local source

Device Family Packs are device description files (.PIC files for PIC® devices, .ATDF files for AVR® and SAM devices), which contain SFR names, memory regions, programming information. Device-dependent source code files (i.e. peripheral header files) are being moved to DFPs. XC8 (AVR target) and XC32 (SAM target) are implemented today. Libraries will be part of the DFP on XC8 (AVR, CSTARTUP) and XC32 – XC16 will store the libraries in the compiler directory.

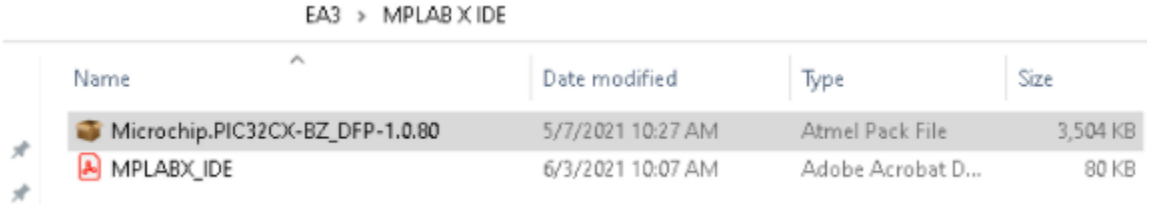
2.1 

Open the MPLAB X IDE and select Tools > Packs

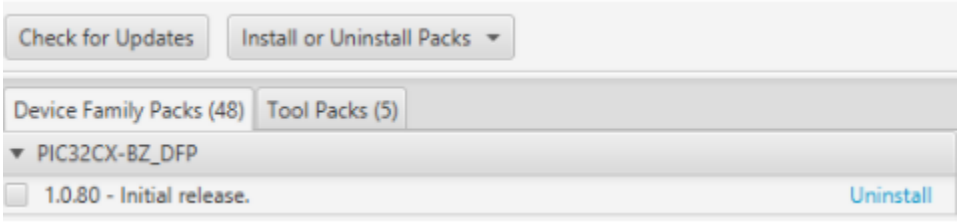
2.2 Click “Install from Local Source”



2.3 **Locate (EAx\MPLAB X IDE)** and **select** the device family pack **Microchip.PIC32CX-BZ_DFP-1.0.xx** in **MPLAB X IDE** directory of the EA Package



2.4 **Verify** the installation of device family pack by searching in the window – search for "bz" keyword



2.5 **Restart** MPLAB X IDE

3. Clone the repository "EA" – using command – "git clone https://github.com/MicrochipTech/EA.git"
   Note: Users must clone this repository in the root directory(maximum 1 level deep), C:/folder_name (folder_name cannot exceed 4 characters for example C:/mchp)

4. copy "clk_pic32cx_bz" folder located in EA directory to "EA\H3\csp\peripheral" folder. A prompt will appear warning "plib_clk.c.ftl" is available in destination folder Select "Replace the file in the destination".

5. Copy the "H3" folder located in EA directory to C:\ drive, users must copy the H3 folder to the root directory like C:\H3

6. Setup the MPLAB Harmony Framework

MPLAB Harmony Framework provides flexible and interoperable software modules to simplify the software development. It includes peripheral libraries, Drivers and services, Reusable middleware

Tools and Harmony 3 Package versions for software development on PIC32CX_BZ2/WBZ45x – The EA package has been tested with the below versions

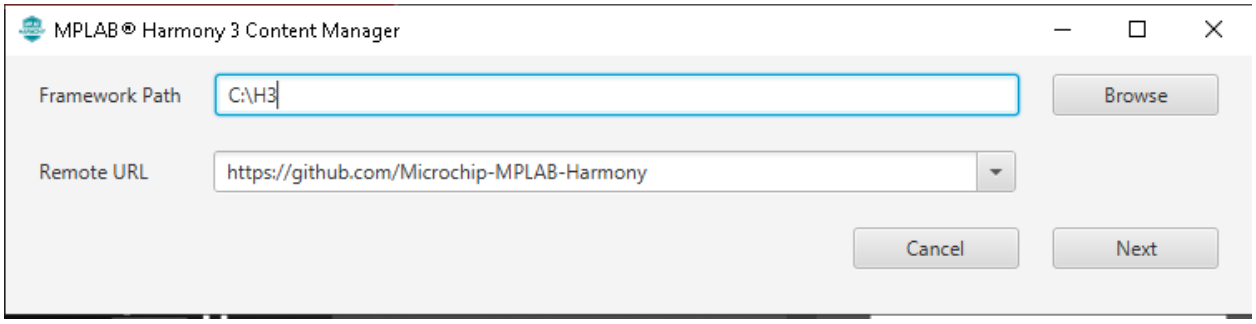
Table 1:

Package	Version	Location
MPLAB X IDE XC32	5.50 3.01	IDE folder Compiler folder
Harmony 3 Plugin	3.6.4	MPLAB X IDE >

		Tools > Plugins
csp	3.10.0	MPLAB® Harmony3 Content Manager
core	3.10.0	MPLAB® Harmony3 Content Manager
mhc	3.8.0	MPLAB® Harmony3 Content Manager
dev_packs	3.10.0	MPLAB® Harmony3 Content Manager
bsp	3.10.0	MPLAB® Harmony3 Content Manager
CMSIS-FreeRTOS	10.3.1	MPLAB® Harmony3 Content Manager
crypto	3.7.2	MPLAB® Harmony3 Content Manager
wolfssl	4.7.0	MPLAB® Harmony3 Content Manager
wireless	211211	EA\H3
PIC32CX-BZ_DFP	1.0.80	EA\MPLAB X IDE
Wireless_system_pic32cxbz_wbz		EA\H3\

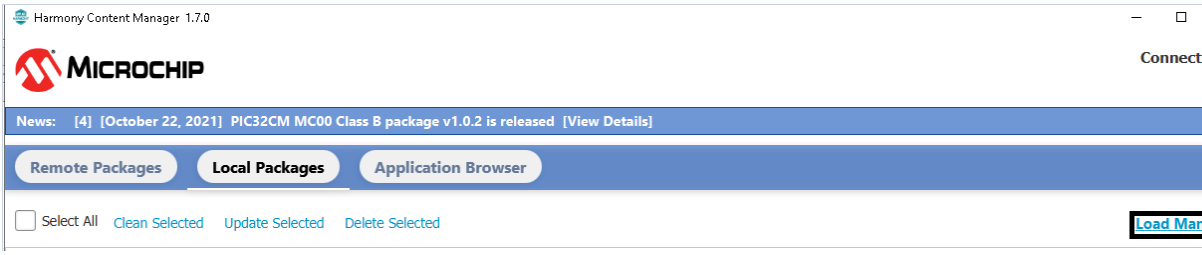
7.1 Open MPLAB X IDE and select **Tools > Embedded > Harmony 3 Content Manager**

7.2 Select **Framework Path** as **C:\H3**, enter **Remote URL** as is: **https://github.com/Microchip-MPLAB-Harmony** and click **Next**

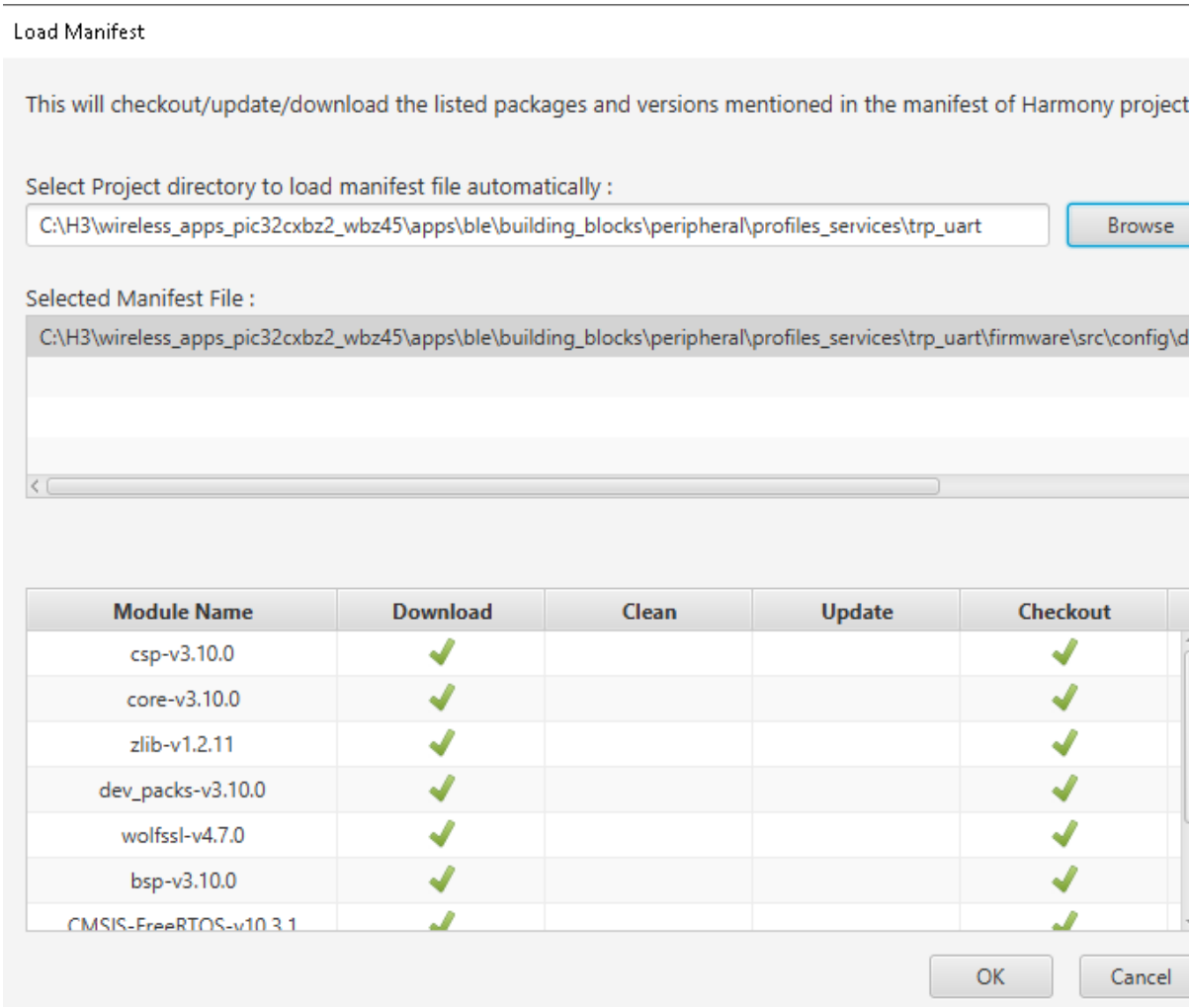


7.3 **A new window – Harmony 3 Content Manager pops-up**

- **go to Local Packages and select "Load Manifest" option**

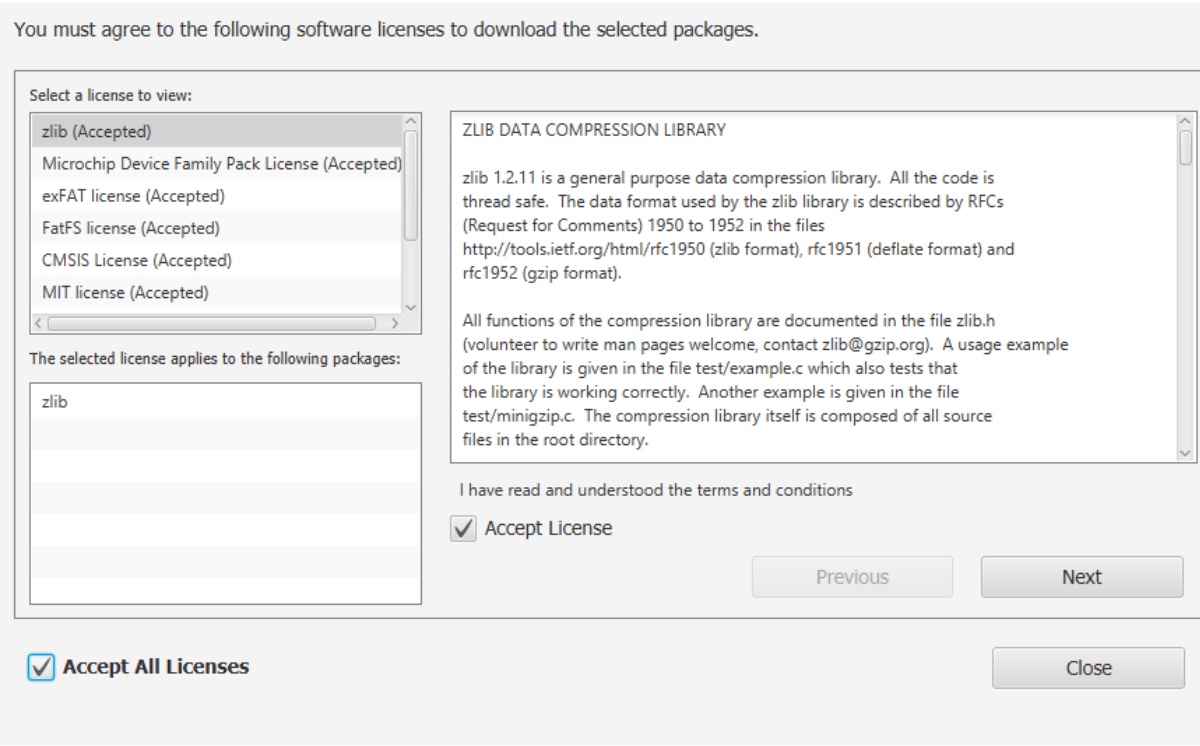


- **Browse to "C:\H3\wireless_apps_pic32cxbz2_wbz45\apps\ble\building_blocks\peripheral\profiles_services\trp_uart" location and select "ok"**

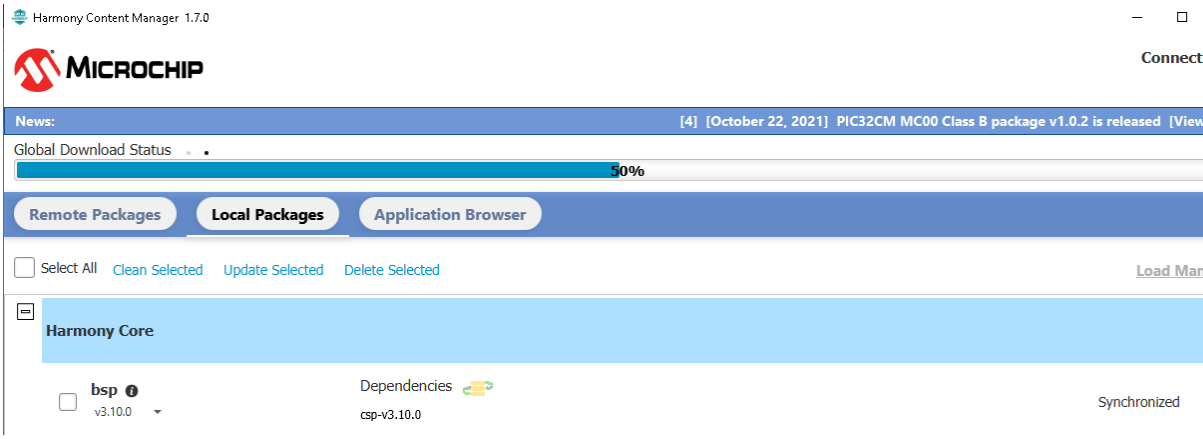


- **Accept all licenses and Select “Close”**

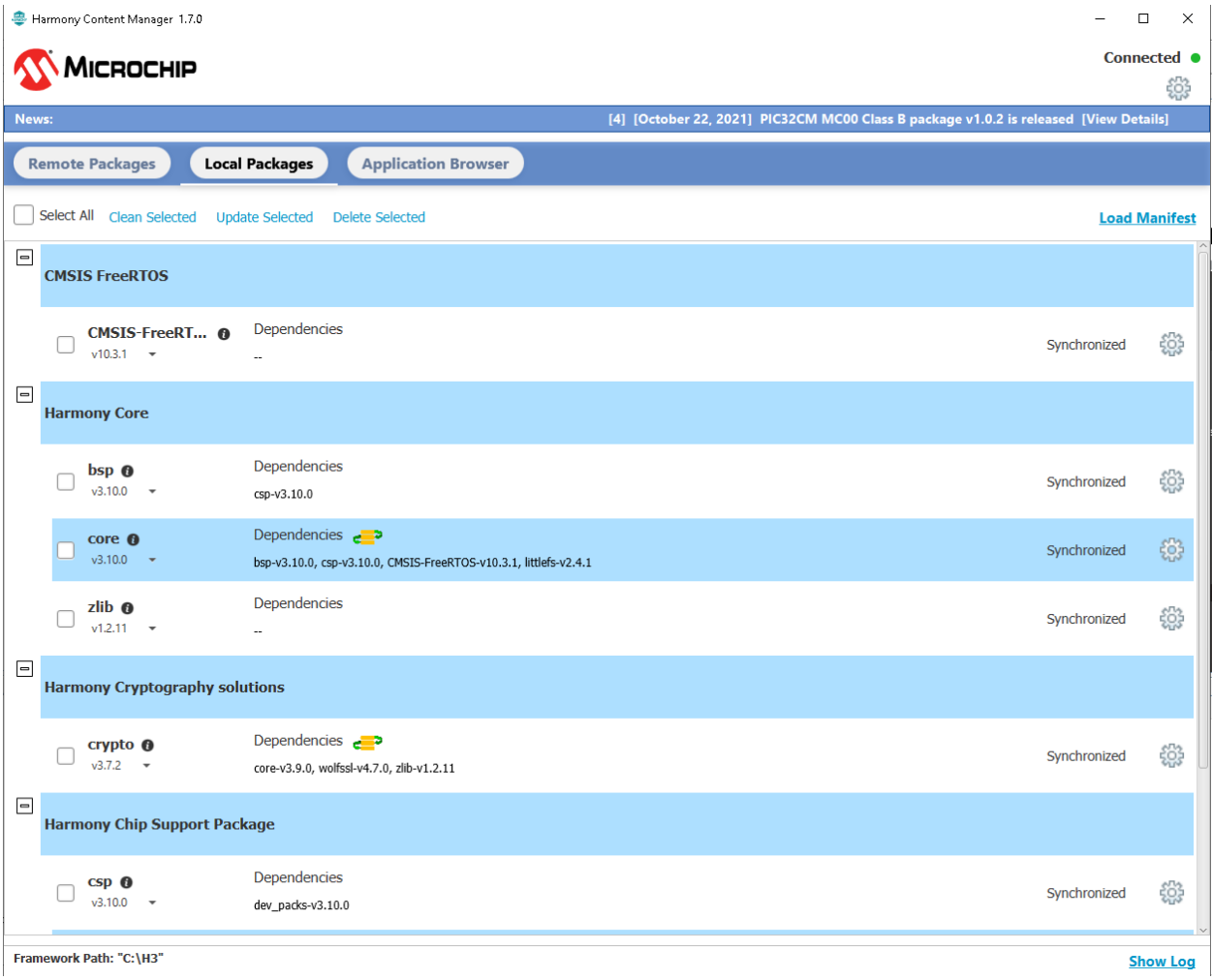
License Window



- **Harmony 3 Components will begin downloading**



- **After downloading is Local Packages will display the downloaded H3 components**



Caution: Users should not update/download any components based of remote packages, EA package is tested with only the versions mentioned in Table 1

7. How to open, build and program an existing application example

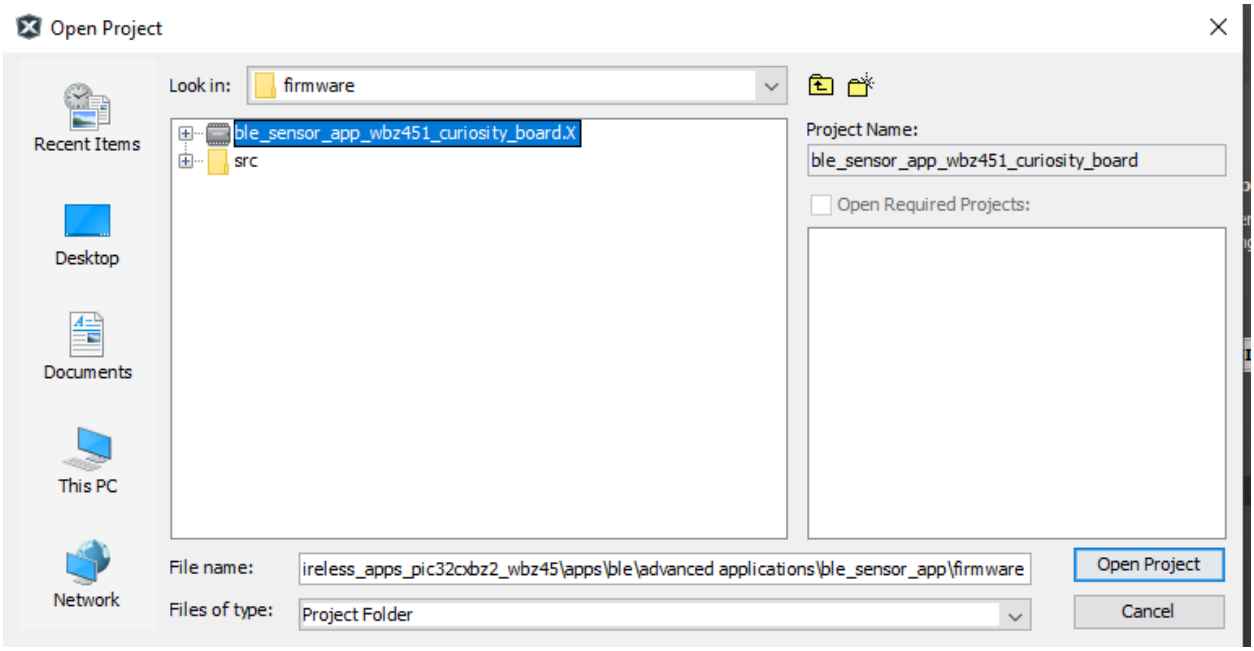
Pre-requisites: Complete all the steps mentioned in previous sections 1-6

8.1 Connect Curiosity Board to the PC using usb cable

8.2 Open MPLAB IDE

8.3 Select **File > Open Project**

8.4 Select the project from C:\H3\wireless_apps_pic32cxbz2_wbz45\apps\ble\advanced applications\ble_sensor_app\firmware



Information related to the workings of the application example are available in readme.md file available in the ble_sensor_app folder or Getting Started html

8.5 Open Project Properties

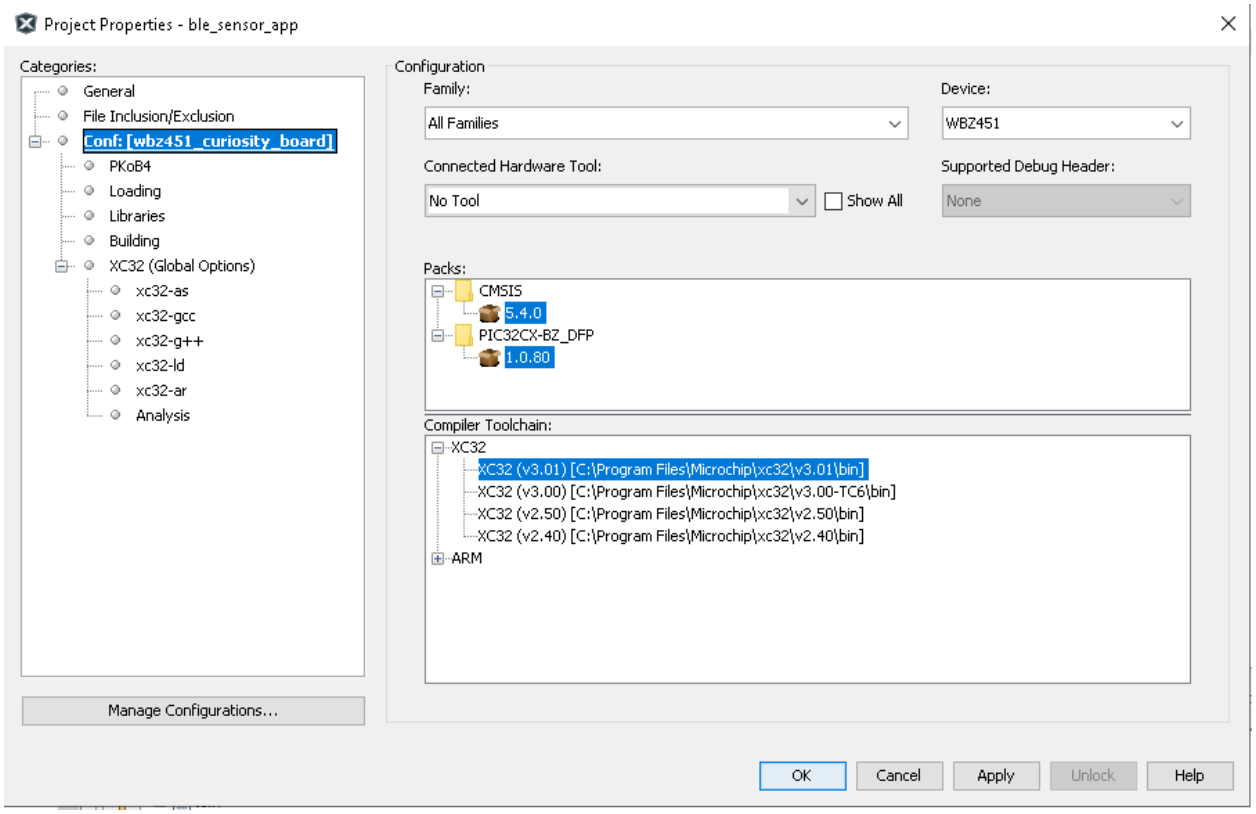
8.5.1 Select WBZ451 Curiosity Board as hardware tool for programming

8.5.2 Ensure DFP v1.0.xx is selected and CMSIS v5.4.0

Note: DFP version should match the version mentioned in table 1

8.5.3 select XC32 v3.01 compiler (in case user has several versions of XC32 compilers installed)

Note: Compiler version should match the version mentioned in table 1



8.6 Select option **Build Project** in IDE to compile the application example

8.7 Plug the Curiosity Development board to PC using usb cable

8.8 Select option **Run Project** in IDE to program the target – the onboard debugger will program the example application

Note: A smartphone App might be needed to explore the full feature set of Application examples, users can refer to readme.md (markdown reader recommended) available in respective Application Example folder or Getting Started html points to the instructions of the Application example