Porting Microchip WINC Harmony3 Drivers for the Host SAMA5D27 MPU

Introduction

This document details the porting of WINC1500 and WINC3400 Drivers for the Host SAMA5D27 MPU using the SAMA5D27-SOM1-EK1 development board. The document also describes various aspects of the SW and HW requirements, creating a basic application to check the functionality of WINC driver initialization, the peripherals that are required and related configurations etc. for WINC driver init, and a sample AP_Scan application has also been described here for a quick reference.

WINC Driver / Application Requirements

Software and Utility Requirements (Install the latest from Microchip Website)

- MPLAB X IDE (v 6.0 v or above)
- XC32 Compiler (v 4.0 v or above)
- MCC Plugin (v x.0 v or above, with MCC Plug in v5.1.4)

Third-party Tools

- TeraTerm for Console application
- J-Link for J-Link / J-Trace / Flasher / Debug SW from https://www.segger.com/downloads/jlink/

Hardware Requirements

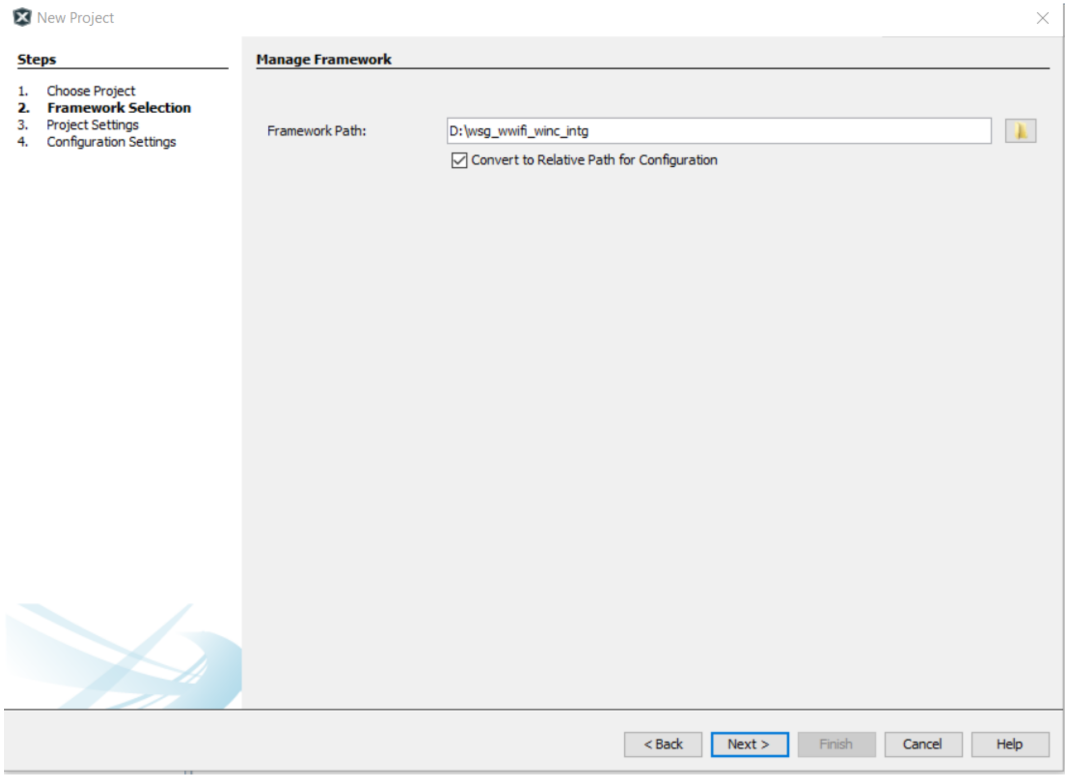
- SAMA5D27-SOM1-EK1 Evaluation Kit
- MicroEbus to XPRO Adapter
- WINC1500 XPRO Board
- WiFi7 Click Board from MicroE

Steps For Creating Application for Basic Driver Init:

1. Download / Clone the wireless_wifi repo from the Bitbucket using the following link - https://bitbucket.microchip.com/projects/WSGSW/repos/wireless_wifi/browse?at=refs%2Fheads%2Fwinc_intergration_sama5d27
   Checkout to **winc_intergration_sama5d27** branch –

```
git checkout origin/winc_intergration_sama5d27 -b local_winc_intergration_sama5d27
```

2. Open MPLAB X IDE v6.0
3. Create New MCC MPU32 Bit Device Project with the following names and configurations:
   a. Inside Configuration → Configuration Name as → sam_a5d27_som1_winc1500
   b. Inside Firmware → Project Name as → sam_a5d27_som1_winc1500.X
   c. Device / MPU32 – ATSAMA5D27C1G
   d. Packs – SAMA5D2_DFP (1.8.9), CMSIS (5.8.0)
   e. XC32 – v4.0, MCC Core as v5.1.4



New Project

Steps

1. Choose Project
2. **Framework Selection**
3. Project Settings
4. Configuration Settings

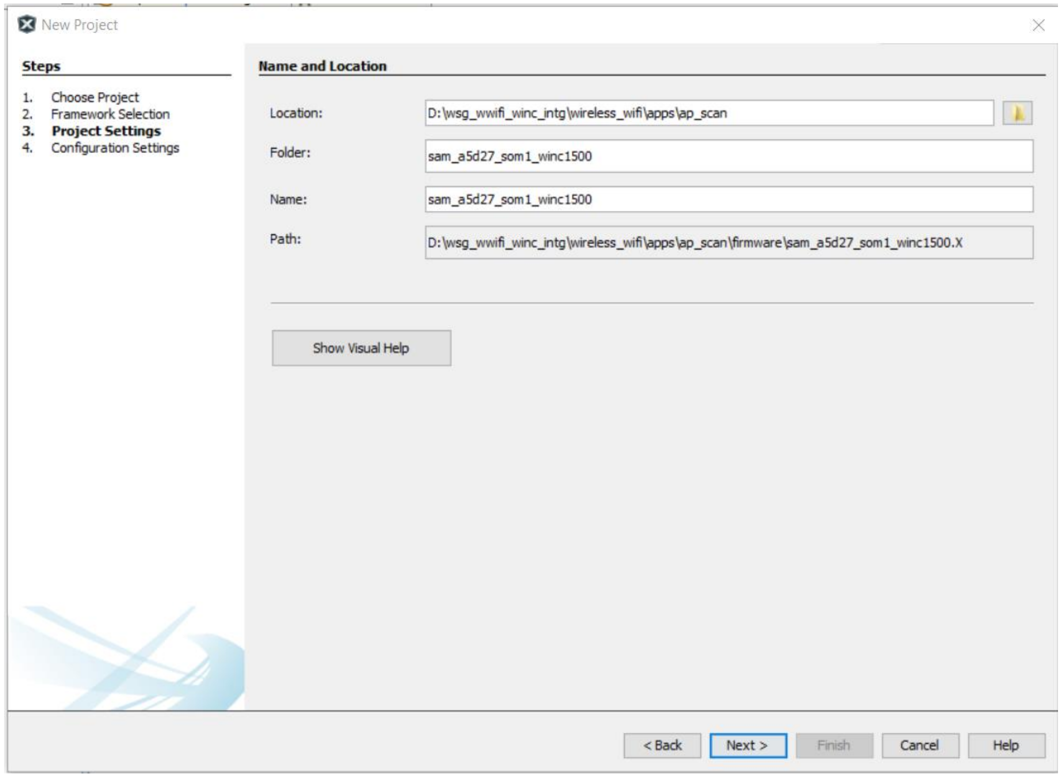
Manage Framework

Framework Path: D:\wsg_wwifi_winc_intg

☑ Convert to Relative Path for Configuration

Framework required repos

a. content_manager_artifacts
b. core
c. csp
d. dev_packs
e. Devices
f. harmony-services
g. mcc
h. mhc
i. quick_docs
j. wireless_wifi (branch - winc_integration)
k. net
l. crypto
m. paho.mqtt.embedded-c
n. wolfMQTT
o. wolfssh
p. wolfssl
q. zlib
r. wireless_apps_winc1500
s. wireless_apps_winc3400



Steps

1. Choose Project
2. Framework Selection
3. **Project Settings**
4. Configuration Settings

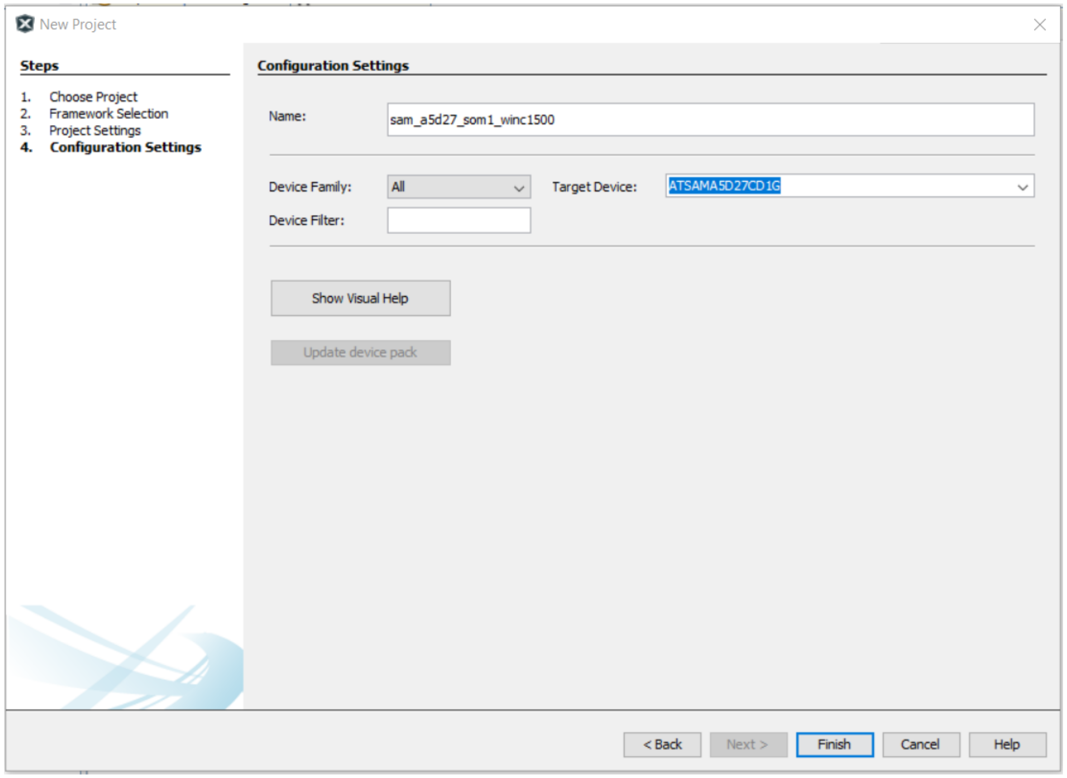
Name and Location

Location: D:\wsg_wwifi_winc_intg\wireless_wifi\apps\ap_scan

Folder: sam_a5d27_som1_winc1500

Name: sam_a5d27_som1_winc1500

Path: D:\wsg_wwifi_winc_intg\wireless_wifi\apps\ap_scan\firmware\sam_a5d27_som1_winc1500.X



New Project

Steps

1. Choose Project
2. Framework Selection
3. Project Settings
4. **Configuration Settings**

Configuration Settings

Name: sam_a5d27_som1_winc1500

Device Family: All

Target Device: ATSAMA5D27CD1G

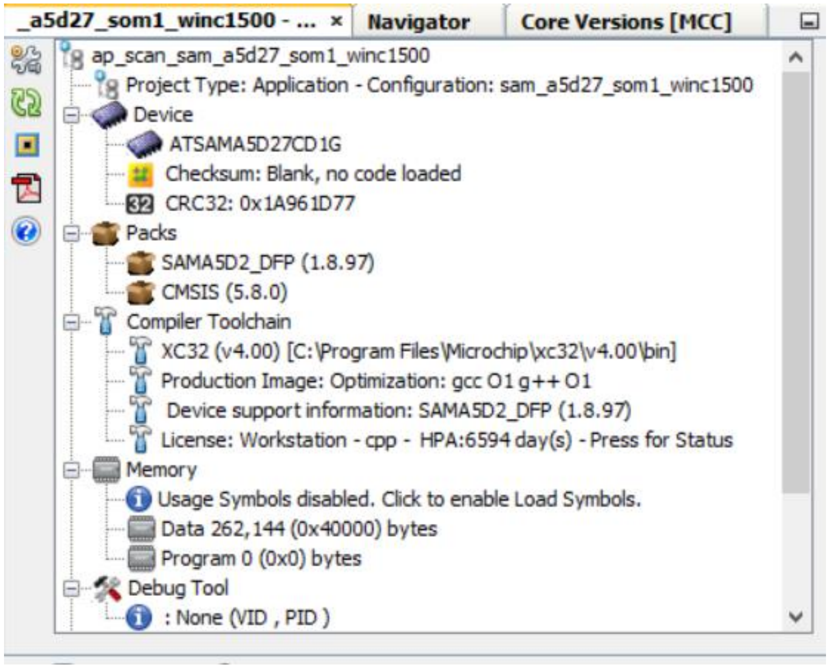
Device Filter:

f. Set the project as the Main Project

g. Rename project to desired application name inside IDE such as (for ap_scan)

- Inside MPLAB X IDE → Project Name as →ap_scan_sam_a5d27_som1_winc1500

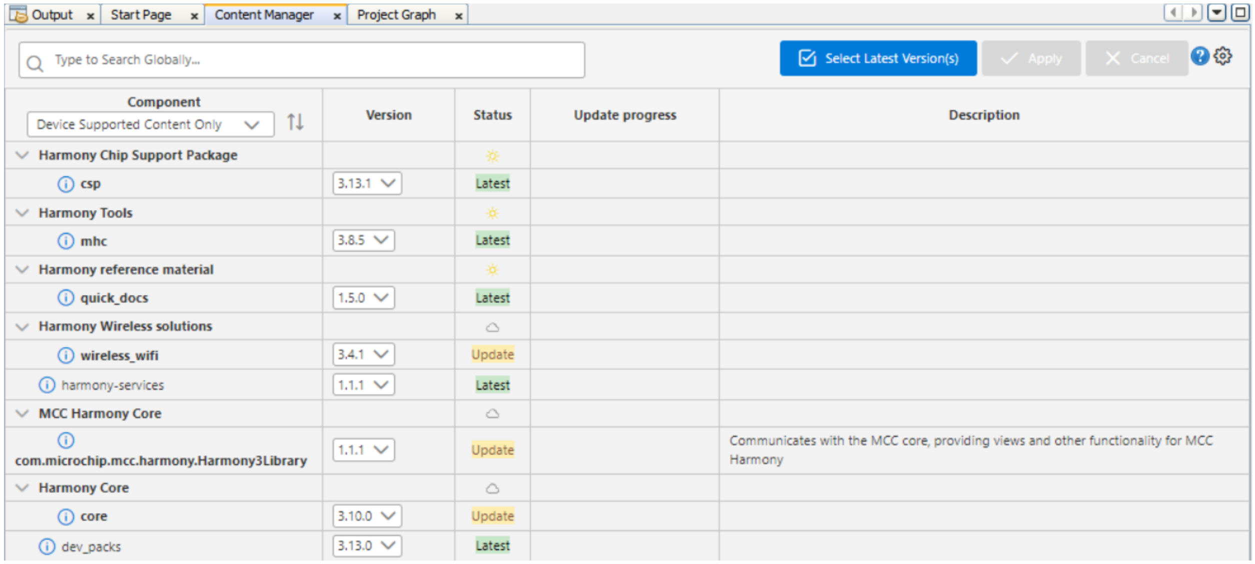
h. Check the project configurations from the Project properties



```
ap_scan_sam_a5d27_som1_winc1500
    Project Type: Application - Configuration: sam_a5d27_som1_winc1500
  Device
      ATSAMA5D27CD1G
      Checksum: Blank, no code loaded
      CRC32: 0x1A961D77
  Packs
      SAMA5D2_DFP (1.8.97)
      CMSIS (5.8.0)
  Compiler Toolchain
      XC32 (v4.00) [C:\Program Files\Microchip\xc32\v4.00\bin]
      Production Image: Optimization: gcc O1 g++ O1
      Device support information: SAMA5D2_DFP (1.8.97)
      License: Workstation - cpp -  HPA:6594 day(s) - Press for Status
  Memory
      Usage Symbols disabled. Click to enable Load Symbols.
      Data 262,144 (0x40000) bytes
      Program 0 (0x0) bytes
  Debug Tool
      : None (VID , PID )
```

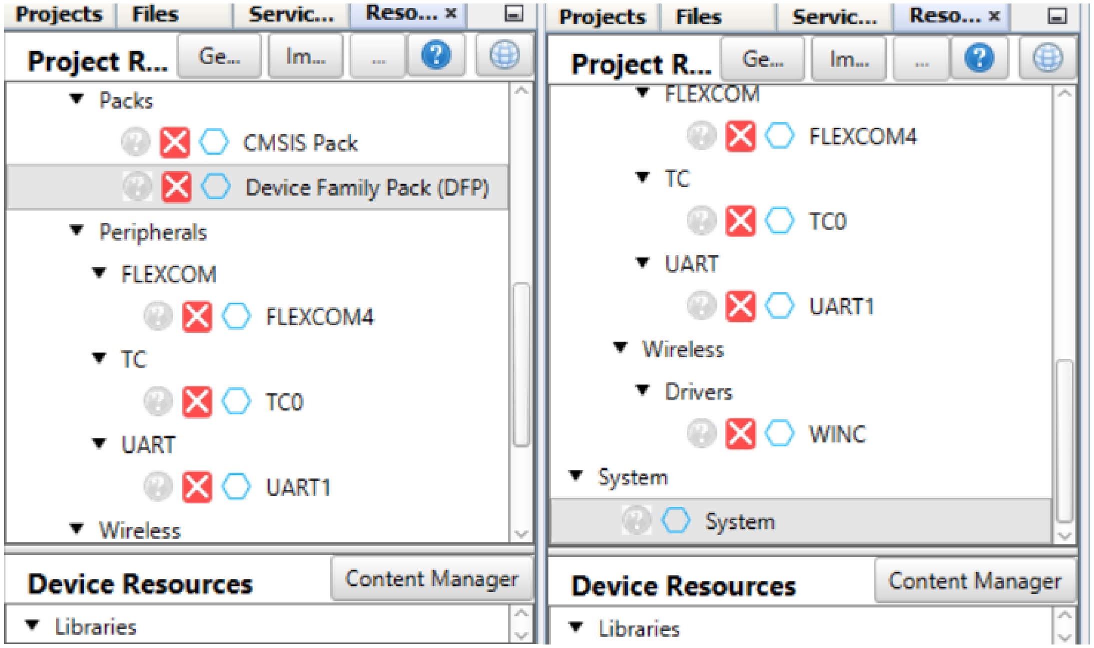
4. Launch MCC and download/clone the repos (Bitbucket) from the Content Manager link such as (this is apart from the Cloned wireless_wifi repo)
    i. content_manager_artifacts
    ii. core
    iii. csp
    iv. dev_packs
    v. Devices
    vi. harmony-services
    vii. mcc
    viii. mhc
    ix. quick_docs
    x. wireless_wifi (Checkout branch - winc_integration)
    xi. net - clone
    xii. crypto

xiii. paho.mqtt.embedded-c
xiv. wolfMQTT
xv. wolfssh
xvi. wolfssl
xvii. zlib
xviii. wireless_apps_winc1500
xix. wireless_apps_winc3400



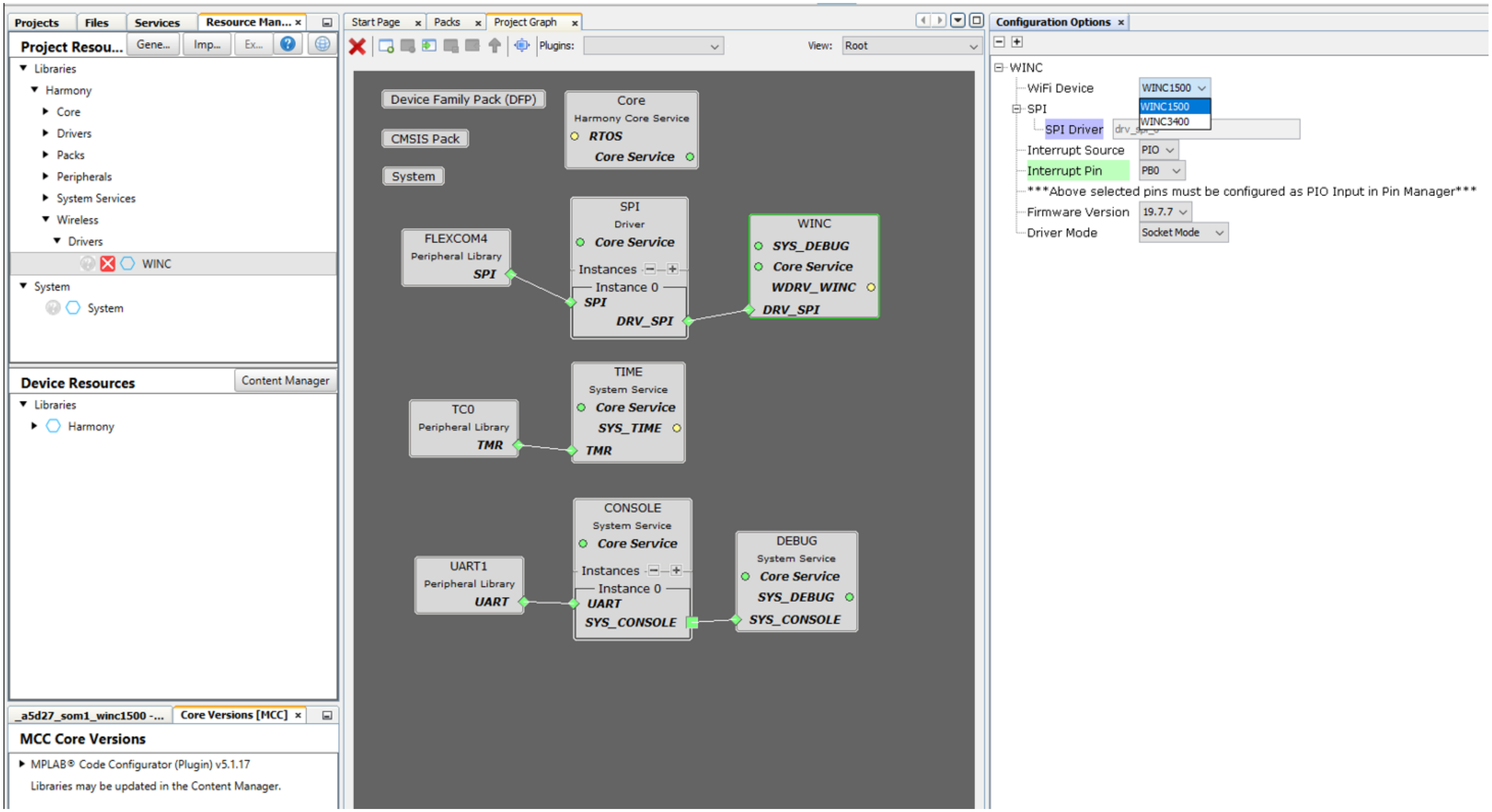
Component	Version	Status	Update progress	Description
Harmony Chip Support Package				
csp	3.13.1	Latest		
Harmony Tools				
mhc	3.8.5	Latest		
Harmony reference material				
quick_docs	1.5.0	Latest		
Harmony Wireless solutions				
wireless_wifi	3.4.1	Update		
harmony-services	1.1.1	Latest		
MCC Harmony Core				
com.microchip.mcc.harmony.Harmony3Library	1.1.1	Update		Communicates with the MCC core, providing views and other functionality for MCC Harmony
Harmony Core				
core	3.10.0	Update		
dev_packs	3.13.0	Latest		

5. Make sure to have all the repo based before you start adding components into the Project Graph
6. Project would usually have the DFP, CMSIS, System and Core populated
7. Start adding the Harmony components required for the Host D27 + WINC1500 Applications (from Device Resources to Project Resources)
    - WINC from Wireless
    - SPI Driver
    - Time, Console, Debug from System Services
    - Flexcom4(SPI), TC0 and UART1 from peripherals and link the components as shown below



Adding Harmony components to the Project Graph

8. Add the Harmony components to the Project Graph and connect as shown below.
    - Choose appropriate WiFi device WINC1500/WINC3400
    - Choose appropriate Firmware version for the selected WiFi device.



WINC

WiFi Device: WINC1500 (WINC1500, WINC3400)

SPI

SPI Driver: drv_spi_0

Interrupt Source: PIO

Interrupt Pin: PB0

Above selected pins must be configured as PIO Input in Pin Manager

Firmware Version: 19.7.7

Driver Mode: Socket Mode

MCC Core Versions

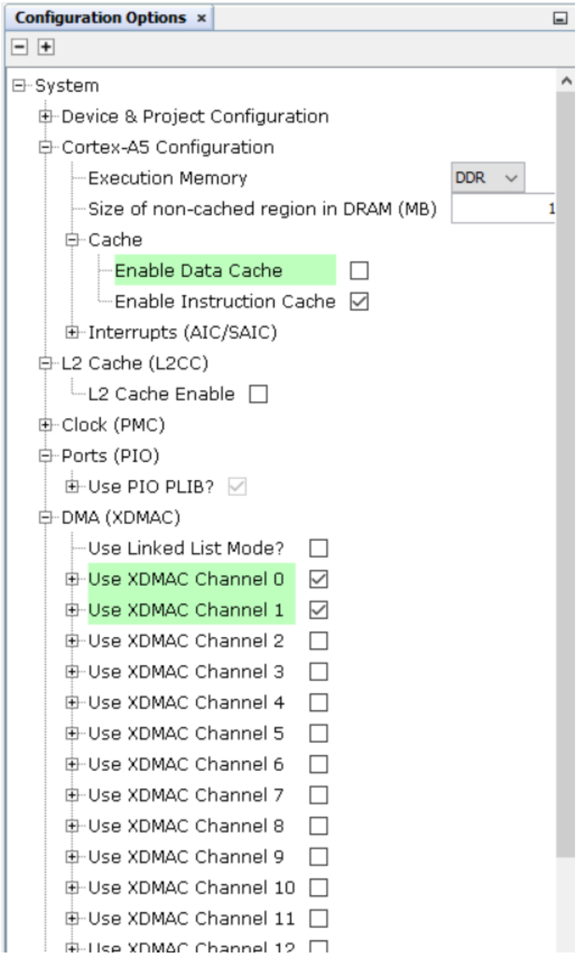
MPLAB® Code Configurator (Plugin) v5.1.17

Libraries may be updated in the Content Manager.

9. Once the components are connected, generate code (MCC based code) for a quick check.

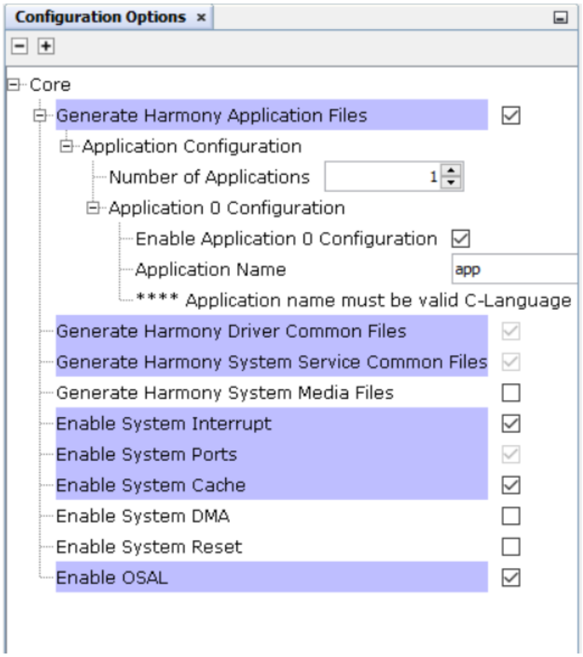
Project / application configurations

System



- System
  - Device & Project Configuration
  - Cortex-A5 Configuration
    - Execution Memory: DDR
    - Size of non-cached region in DRAM (MB): 1
    - Cache
      - Enable Data Cache □
      - Enable Instruction Cache ☑
    - Interrupts (AIC/SAIC)
  - L2 Cache (L2CC)
    - L2 Cache Enable □
  - Clock (PMC)
  - Ports (PIO)
    - Use PIO PLIB? ☑
  - DMA (XDMAC)
    - Use Linked List Mode? □
    - Use XDMAC Channel 0 ☑
    - Use XDMAC Channel 1 ☑
    - Use XDMAC Channel 2 □
    - Use XDMAC Channel 3 □
    - Use XDMAC Channel 4 □
    - Use XDMAC Channel 5 □
    - Use XDMAC Channel 6 □
    - Use XDMAC Channel 7 □
    - Use XDMAC Channel 8 □
    - Use XDMAC Channel 9 □
    - Use XDMAC Channel 10 □
    - Use XDMAC Channel 11 □
    - Use XDMAC Channel 12 □

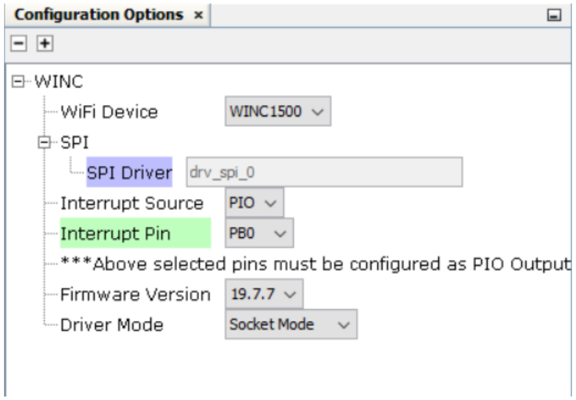
Core



Configuration Options

- Core
  - Generate Harmony Application Files ☑
    - Application Configuration
      - Number of Applications 1
      - Application 0 Configuration
        - Enable Application 0 Configuration ☑
        - Application Name app
        - **** Application name must be valid C-Language
  - Generate Harmony Driver Common Files ☑
  - Generate Harmony System Service Common Files ☑
  - Generate Harmony System Media Files □
  - Enable System Interrupt ☑
  - Enable System Ports ☑
  - Enable System Cache ☑
  - Enable System DMA □
  - Enable System Reset □
  - Enable OSAL ☑

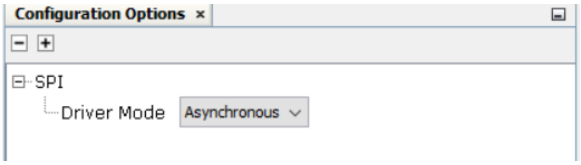
WINC



Configuration Options

- WINC
  - WiFi Device WINC1500
  - SPI
    - SPI Driver drv_spi_0
  - Interrupt Source PIO
  - Interrupt Pin PB0
  - ***Above selected pins must be configured as PIO Output
  - Firmware Version 19.7.7
  - Driver Mode Socket Mode

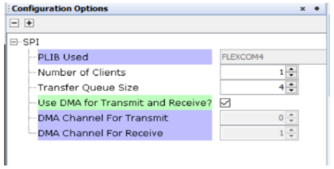
SPI



Configuration Options

SPI

Driver Mode	Asynchronous

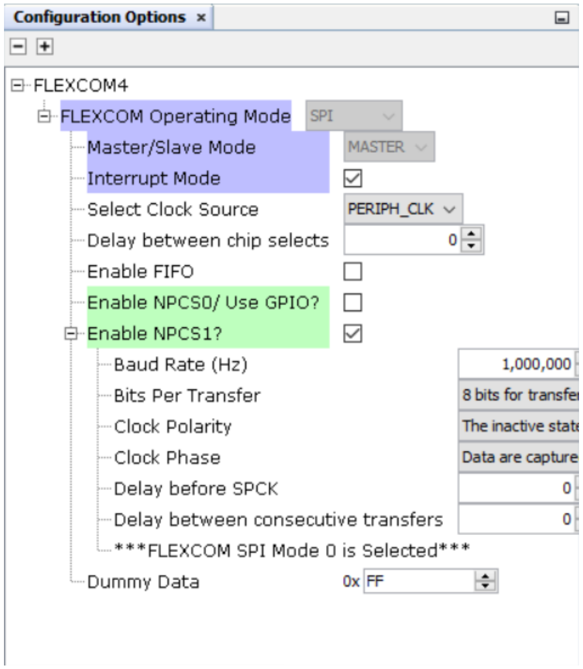


Configuration Options

SPI

PLIB Used	FLEXCOM4
Number of Clients	1
Transfer Queue Size	4
Use DMA for Transmit and Receive?	☑
DMA Channel For Transmit	0
DMA Channel For Receive	1

FLEXCOM4

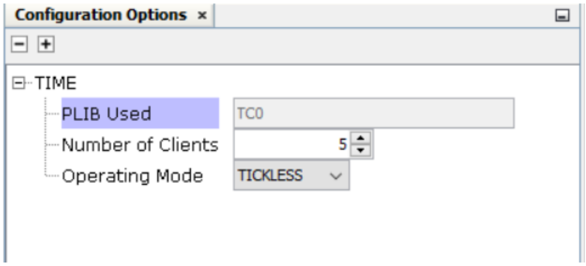


Configuration Options

FLEXCOM4

FLEXCOM Operating Mode	SPI
Master/Slave Mode	MASTER
Interrupt Mode	☑
Select Clock Source	PERIPH_CLK
Delay between chip selects	0
Enable FIFO	□
Enable NPCS0/ Use GPIO?	□
Enable NPCS1?	☑
Baud Rate (Hz)	1,000,000
Bits Per Transfer	8 bits for transfe
Clock Polarity	The inactive stat
Clock Phase	Data are capture
Delay before SPCK	0
Delay between consecutive transfers	0
FLEXCOM SPI Mode 0 is Selected	
Dummy Data	0x FF

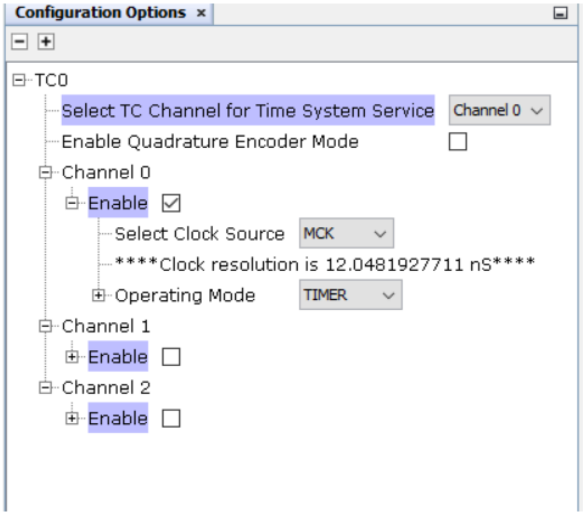
TIME



Configuration Options

- TIME
  - PLIB Used: TC0
  - Number of Clients: 5
  - Operating Mode: TICKLESS

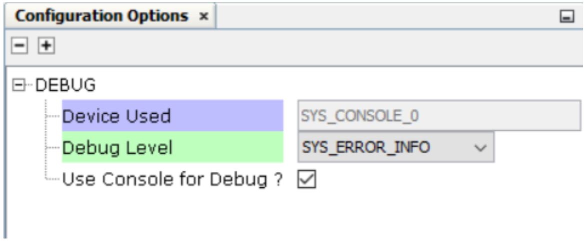
TC0



Configuration Options

- TC0
  - Select TC Channel for Time System Service: Channel 0
  - Enable Quadrature Encoder Mode ☐
  - Channel 0
    - Enable ☑
      - Select Clock Source: MCK
      - ****Clock resolution is 12.0481927711 nS****
      - Operating Mode: TIMER
  - Channel 1
    - Enable ☐
  - Channel 2
    - Enable ☐

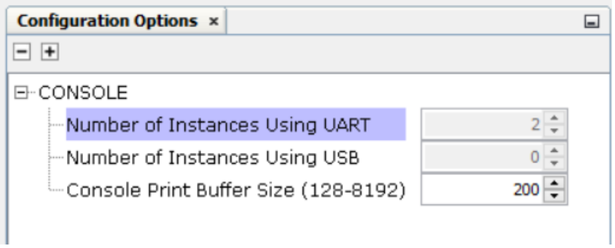
Debug



Configuration Options

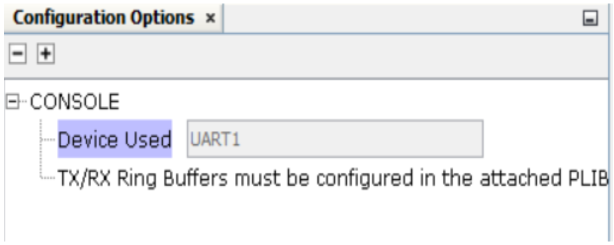
- DEBUG
  - Device Used: SYS_CONSOLE_0
  - Debug Level: SYS_ERROR_INFO
  - Use Console for Debug ? ☑

Console



Configuration Options

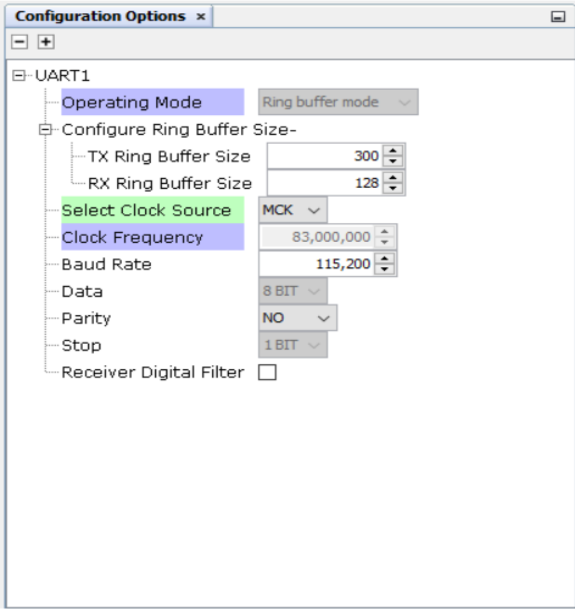
CONSOLE
- Number of Instances Using UART: 2
- Number of Instances Using USB: 0
- Console Print Buffer Size (128-8192): 200



Configuration Options

CONSOLE
- Device Used: UART1
- TX/RX Ring Buffers must be configured in the attached PLIB

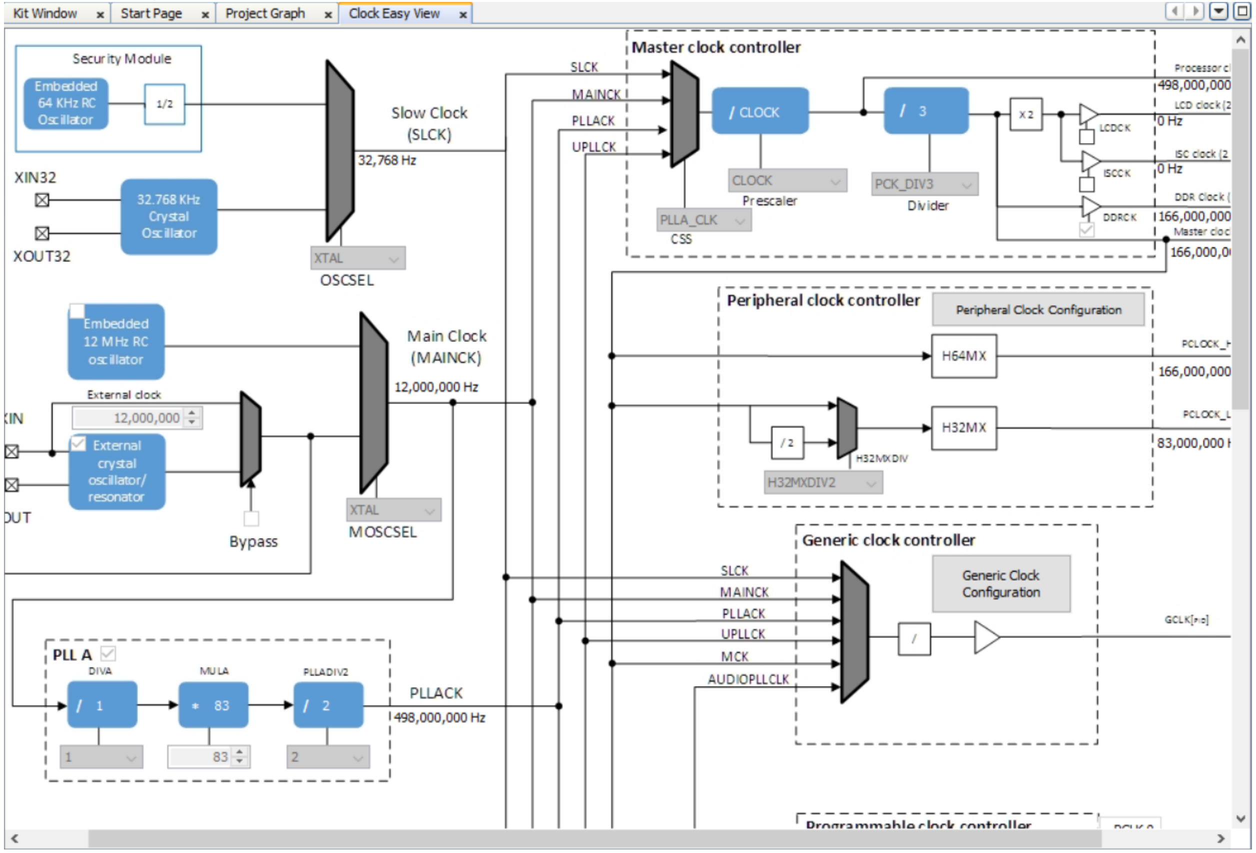
UART1



Configuration Options

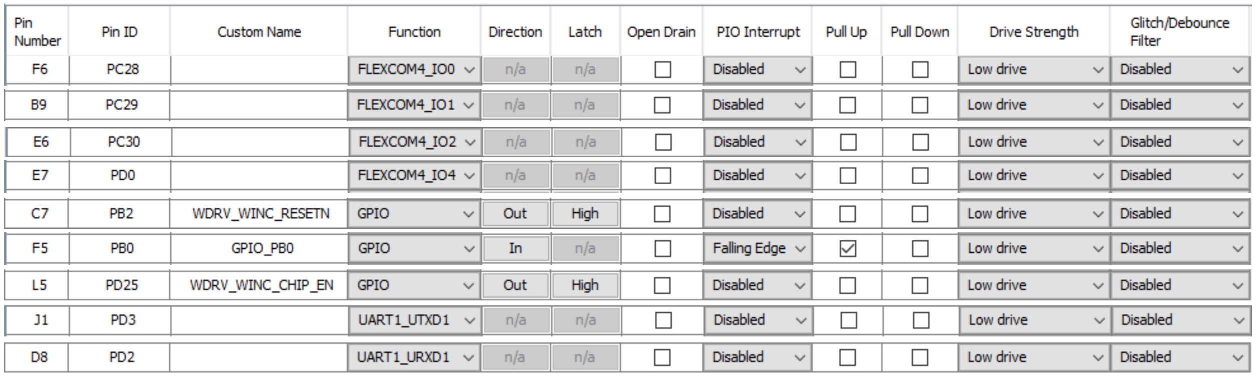
UART1
- Operating Mode: Ring buffer mode
- Configure Ring Buffer Size-
  - TX Ring Buffer Size: 300
  - RX Ring Buffer Size: 128
- Select Clock Source: MCK
- Clock Frequency: 83,000,000
- Baud Rate: 115,200
- Data: 8 BIT
- Parity: NO
- Stop: 1 BIT
- Receiver Digital Filter: □

Clock Configurations



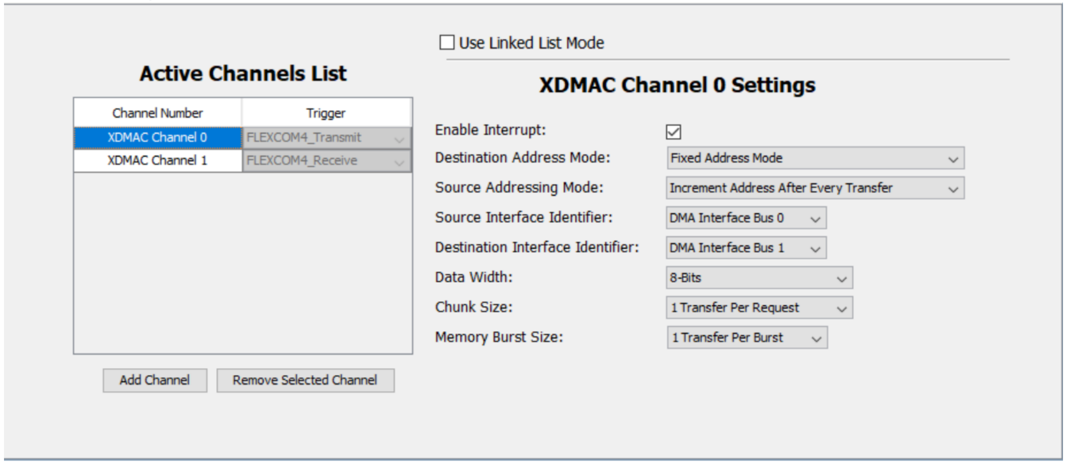
Note: Make sure to check the UART Clock Frequency for 83 MHz with source as MCK, refresh if required

PIN Configurations for WINC1500



Pin Number	Pin ID	Custom Name	Function	Direction	Latch	Open Drain	PIO Interrupt	Pull Up	Pull Down	Drive Strength	Glitch/Debounce Filter
F6	PC28		FLEXCOM4_IO0	n/a	n/a	□	Disabled	□	□	Low drive	Disabled
B9	PC29		FLEXCOM4_IO1	n/a	n/a	□	Disabled	□	□	Low drive	Disabled
E6	PC30		FLEXCOM4_IO2	n/a	n/a	□	Disabled	□	□	Low drive	Disabled
E7	PD0		FLEXCOM4_IO4	n/a	n/a	□	Disabled	□	□	Low drive	Disabled
C7	PB2	WDRV_WINC_RESETN	GPIO	Out	High	□	Disabled	□	□	Low drive	Disabled
F5	PB0	GPIO_PB0	GPIO	In	n/a	□	Falling Edge	☑	□	Low drive	Disabled
L5	PD25	WDRV_WINC_CHIP_EN	GPIO	Out	High	□	Disabled	□	□	Low drive	Disabled
J1	PD3		UART1_UTXD1	n/a	n/a	□	Disabled	□	□	Low drive	Disabled
D8	PD2		UART1_URXD1	n/a	n/a	□	Disabled	□	□	Low drive	Disabled

DMA Configurations



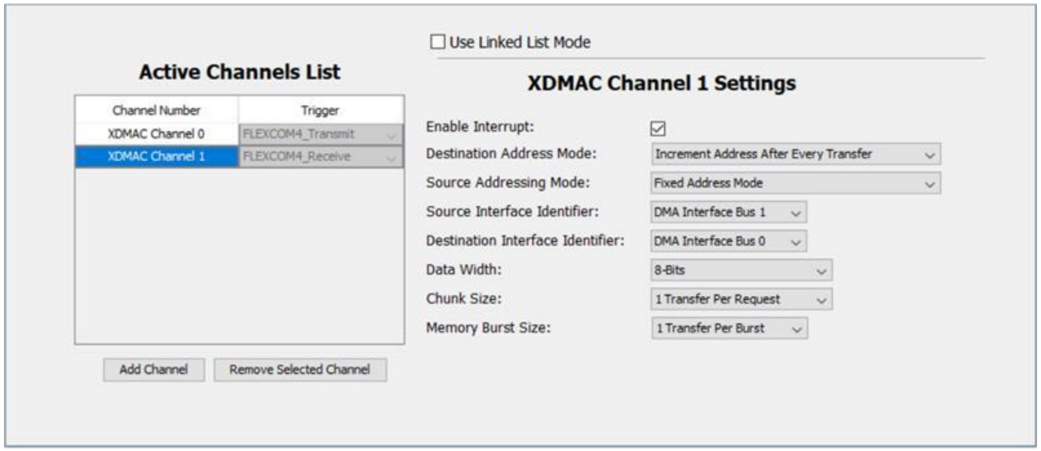
Active Channels List

Channel Number	Trigger
XDMAC Channel 0	FLEXCOM4_Transmit
XDMAC Channel 1	FLEXCOM4_Receive

□ Use Linked List Mode

XDMAC Channel 0 Settings

Enable Interrupt:	☑
Destination Address Mode:	Fixed Address Mode
Source Addressing Mode:	Increment Address After Every Transfer
Source Interface Identifier:	DMA Interface Bus 0
Destination Interface Identifier:	DMA Interface Bus 1
Data Width:	8-Bits
Chunk Size:	1 Transfer Per Request
Memory Burst Size:	1 Transfer Per Burst



Active Channels List

Channel Number	Trigger
XDMAC Channel 0	FLEXCOM4_Transmit
XDMAC Channel 1	FLEXCOM4_Receive

☐ Use Linked List Mode

XDMAC Channel 1 Settings

Enable Interrupt: ☑
Destination Address Mode: Increment Address After Every Transfer
Source Addressing Mode: Fixed Address Mode
Source Interface Identifier: DMA Interface Bus 1
Destination Interface Identifier: DMA Interface Bus 0
Data Width: 8-Bits
Chunk Size: 1 Transfer Per Request
Memory Burst Size: 1 Transfer Per Burst

AIC Configuration

Interrupt Vector	Enable	Handler Name	Map Type	Source Type	Priortiy
XDMAC0_IRQn - DMA Controller 0	Yes	XDMAC0_InterruptHandler	Nonsecure	HL Sensitive/Internal LL Sensitive/External	Minimum
XDMAC1_IRQn - DMA Controller 1	No	XDMAC1_Handler	Nonsecure	HL Sensitive/Internal LL Sensitive/External	Minimum
FLEXCOM4_IRQn – FLEXCOM4	Yes	FLEXCOM4_InterruptHandler	Nonsecure	HL Sensitive/Internal LL Sensitive/External	Minimum
UART1_IRQn – UART1	Yes	UART1_InterruptHandler	Nonsecure	HL Sensitive/Internal LL Sensitive/External	Minimum
TC0_IRQn – Timer Counter 0 (ch. 0,1,2)	Yes	TC0_ InterruptHandler	Nonsecure	HL Sensitive/Internal LL Sensitive/External	Minimum
PIOB_IRQn – Parallel I/O Controller	Yes	PIOB_InterruptHandler	Nonsecure	HL Sensitive/Internal LL Sensitive/External	Minimum

10. Once the required configuration changes are done, Generate the code
11. Now Clean and Build the project
12. The project should build successfully, for any errors/warning – resolve the same and proceed further
13. This would just work for the WINC Driver Initialization, and this needs to be monitored with the SAMA5D27-SOM1-EK1 Evaluation Board
14. **Connecting and checking on SAMA5D27-SOM1-EK1 Evaluation Kit**
    1. For linking the HW SAMA5D27-SOM1-EK1 Board along with the WINC Module on the XPRO Board connected through the XPRO Adapter
    2. Copy the at91bootstrap_sam_a5d27_som1_ek.X project at the location …..\wireless_wifi\apps\ap_scan\firmware
    3. Install JLink Program / Debug utility from Segger (https://www.segger.com/downloads/jlink/) for SOM1-Ek1 board JLink Debugger / Programmer
    4. Connect J10 / JLink on SOM1-EK1 Board and the PC COM Port using the MicroUSB Cable
    5. USB connection powers up the board
    6. Once the SAMA5D27-SOM1-EK1 Board is detected, note down the SUB connected COM Port to use the Console Application using the TeraTerm
    7. Connect the WINC1500 XPRO Board to the SAMA5D27-SOM1-EK1 using the XPRO Adapter

Connection Details of SAM1_EK1 Board to 1500/3400 Module

SOM1_EK1 Board			**WINC1500 Module**		
J24 Mbus1 Header Pins	**Signal name on Mbus**	**SAMA5D27 (For our Ref.)**	**XPRO Header EXT1 Pins**	**Signal name on Module**	**Signal Type**
1	AN	PD25	10	CHIP_EN	Active High
2	RST	PB2	5	RESET_N	Active Low
3	CS	PD0	15	SPI_SSN	Active Low
4	SPCK	PC30	18	SPI_SCK	SPI Clock
5	MISO	PC29	17	SPI_MISO	
6	MOSI	PC28	16	SPI_MOSI	
7	+3.3V		20	+3.3V	
8	GND		19	GND	

SOM1_EK1 Board			WINC1500 Module		
J25 Mbus1 Header	Pin Name on Mbus	SAMA5D27 (For our Ref.)	XPRO Header EXT1 Pins	Signal name on Module	Signal Type
1	PWM			NC	
2	INT (INT_IN)	PB0	9	IRQN (INT_OUT)	Active Low
3	RX			NC	
4	TX			NC	
5	TWCK			NC	
6	TWD			NC	
7	+5.0V			NC	
8	GND			NC	

15. Make the below Changes in the generated code of the PIO to support the PIO Interrupt Handler
    Make the following changes @:

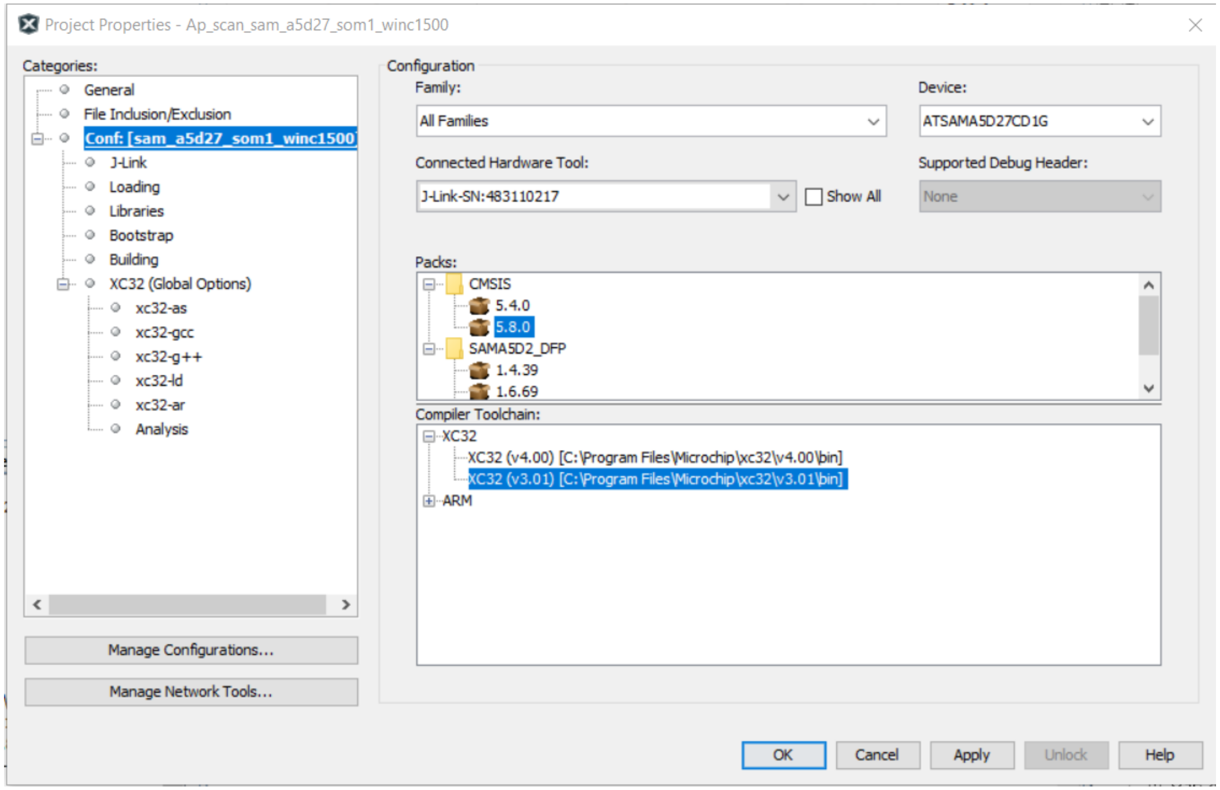
    plib_pio.h

    ```
            typedef  void (*PIO_PIN_CALLBACK) (uintptr_t context);
    ```

    plib_pio.c

    ```
    if((status & ( 1 << (portPinCbObj[j].pin & 0x1F) ) ) && (portPinCbObj[j].callback != NULL))
                            {
                                    portPinCbObj[j].callback ( portPinCbObj[j].context );
                            }
    ```

16. Power up the board using the USB cable through the J10 / JLink connector



Family: All Families

Device: ATSAMA5D27CD1G

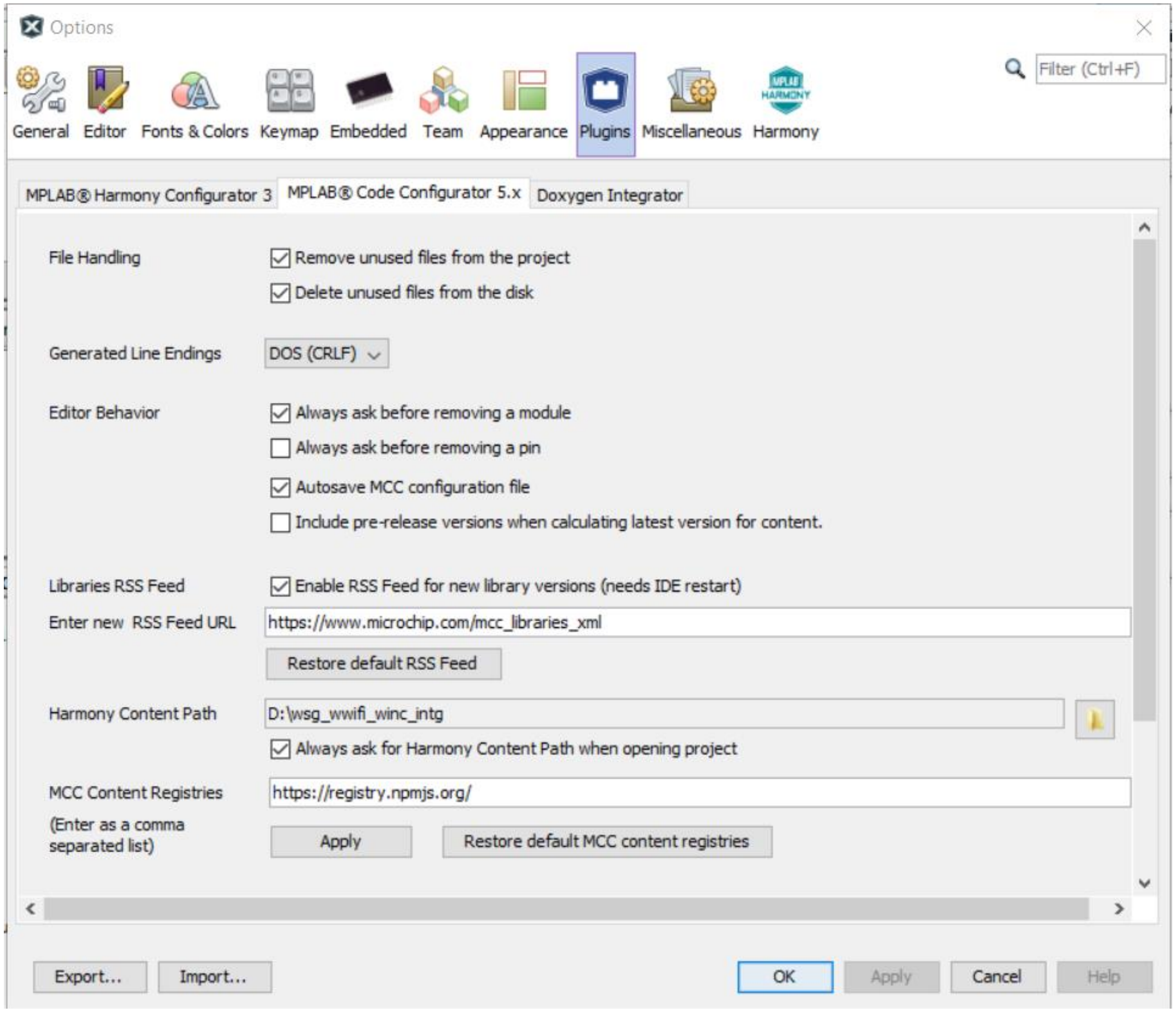
Connected Hardware Tool: J-Link-SN:483110217 □ Show All

Supported Debug Header: None

Note: The JLink Section appears in the Category list, and shows up as a connected HW tool with the serial number

17. Re-Generate the code for bootstrap link to compile
18. AT MPLAB X IDE Tools → Options list check for the Repo base selection

19. For MCC



File Handling
☑ Remove unused files from the project
☑ Delete unused files from the disk

Generated Line Endings: DOS (CRLF)

Editor Behavior
☑ Always ask before removing a module
☐ Always ask before removing a pin
☑ Autosave MCC configuration file
☐ Include pre-release versions when calculating latest version for content.

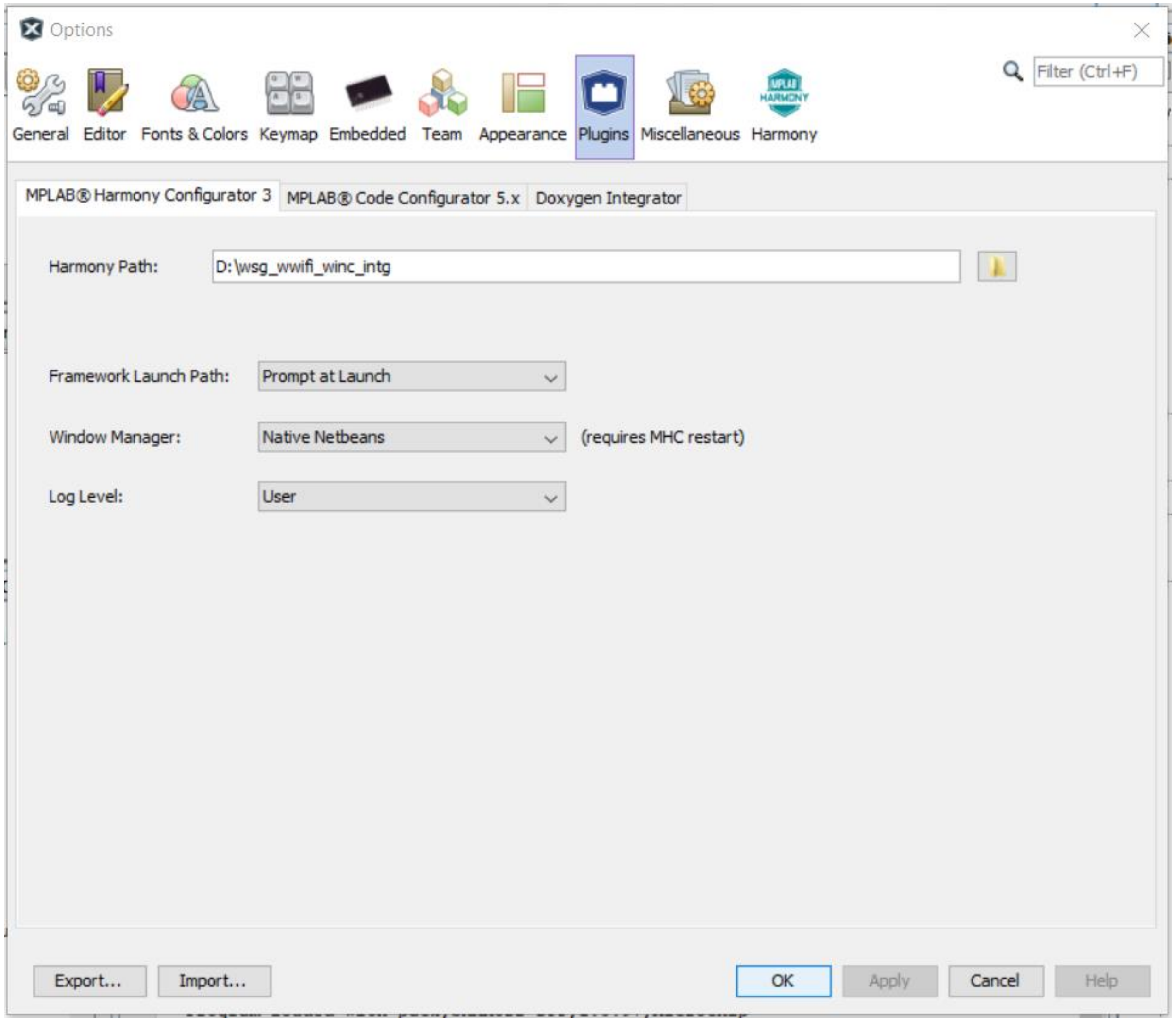
Libraries RSS Feed
☑ Enable RSS Feed for new library versions (needs IDE restart)

Enter new RSS Feed URL: https://www.microchip.com/mcc_libraries_xml

Harmony Content Path: D:\wsg_wwifi_winc_intg
☑ Always ask for Harmony Content Path when opening project

MCC Content Registries (Enter as a comma separated list): https://registry.npmjs.org/

20. For MHC



Harmony Path: D:\wsg_wwifi_winc_intg

Framework Launch Path: Prompt at Launch

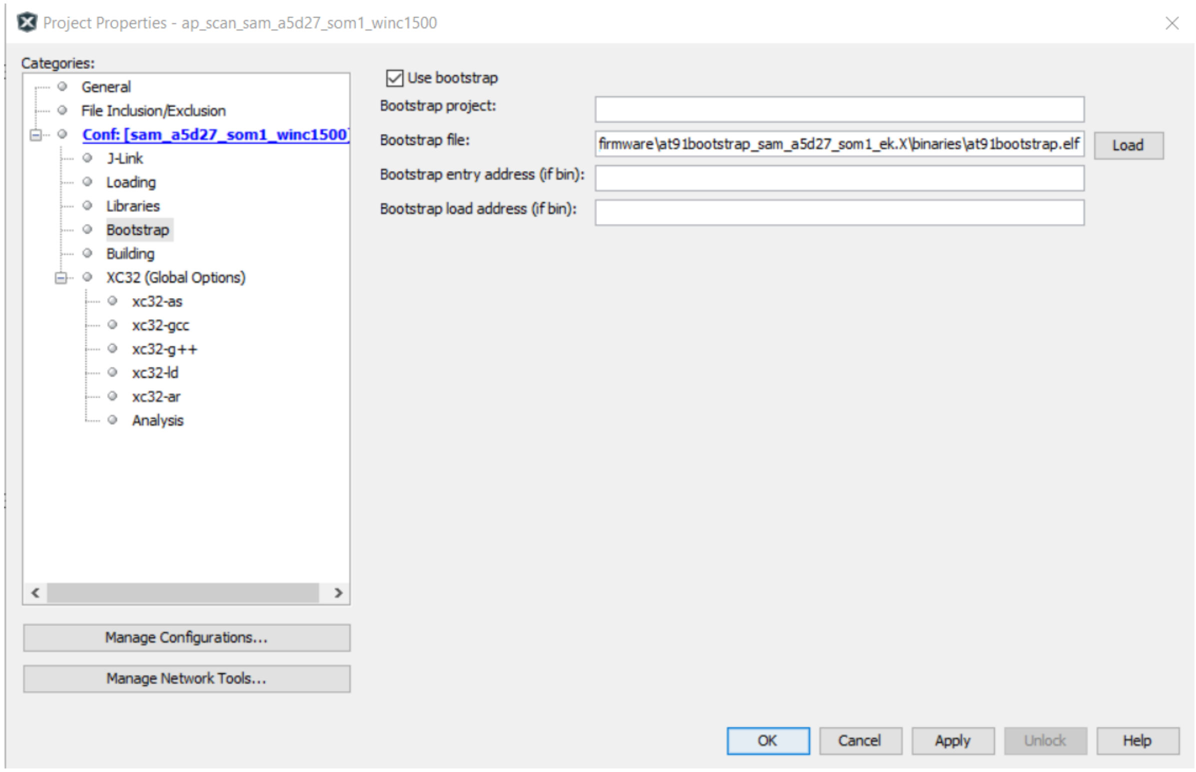
Window Manager: Native Netbeans (requires MHC restart)

Log Level: User

21. Check the following files in the below path.
....\wireless_wifi\apps\ap_scan\firmware\at91bootstrap_sam_a5d27_som1_ek.X\binaries

- at91bootstrap.elf
- boot.bin

22. Link the bootstrap to the elf to the project for JLink support



Project Properties - ap_scan_sam_a5d27_som1_winc1500

☑ Use bootstrap

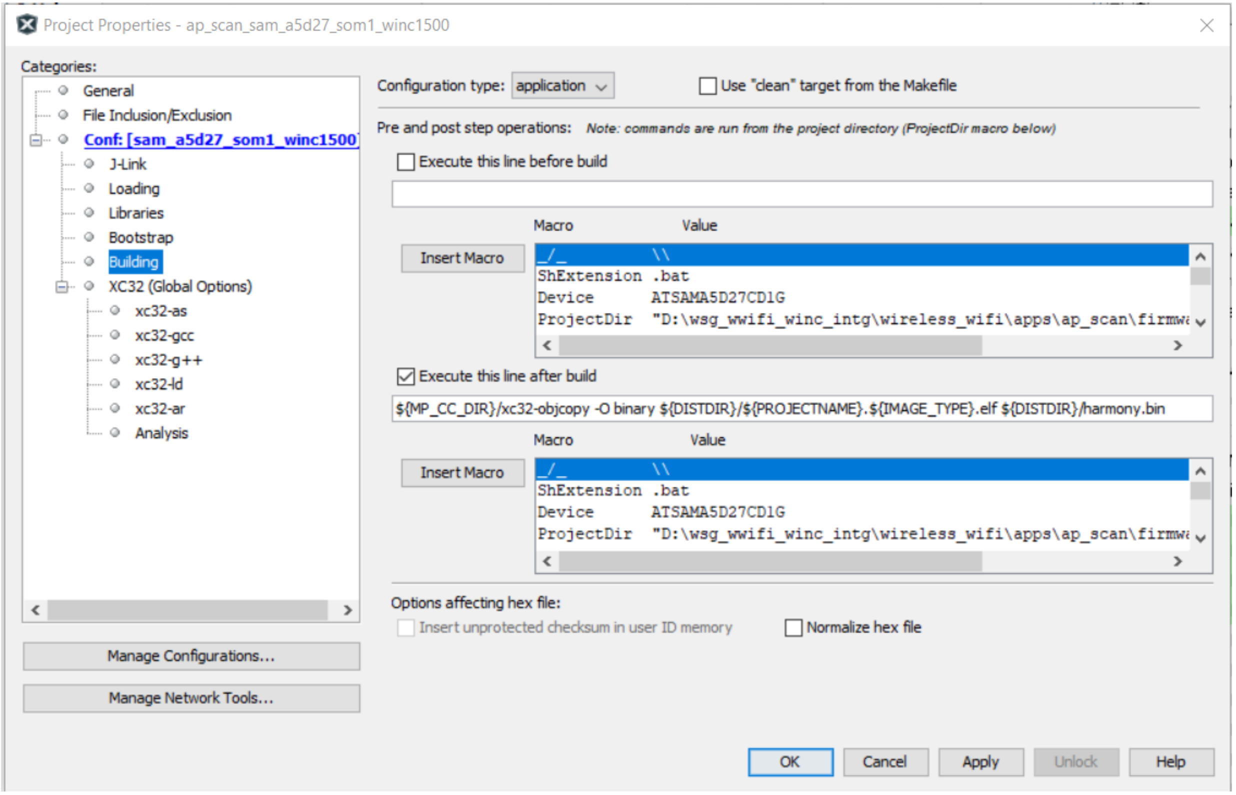
Bootstrap project:

Bootstrap file: firmware\at91bootstrap_sam_a5d27_som1_ek.X\binaries\at91bootstrap.elf

Bootstrap entry address (if bin):

Bootstrap load address (if bin):

23. Generate the harmony.bin from the following Building Category from the Project Properties



Project Properties - ap_scan_sam_a5d27_som1_winc1500

Configuration type: application

□ Use "clean" target from the Makefile

Pre and post step operations: *Note: commands are run from the project directory (ProjectDir macro below)*

□ Execute this line before build

```
Macro        Value
_/_          \\
ShExtension  .bat
Device       ATSAMA5D27CD1G
ProjectDir   "D:\wsg_wwifi_winc_intg\wireless_wifi\apps\ap_scan\firmwa
```

☑ Execute this line after build

```
${MP_CC_DIR}/xc32-objcopy -O binary ${DISTDIR}/${PROJECTNAME}.${IMAGE_TYPE}.elf ${DISTDIR}/harmony.bin
```

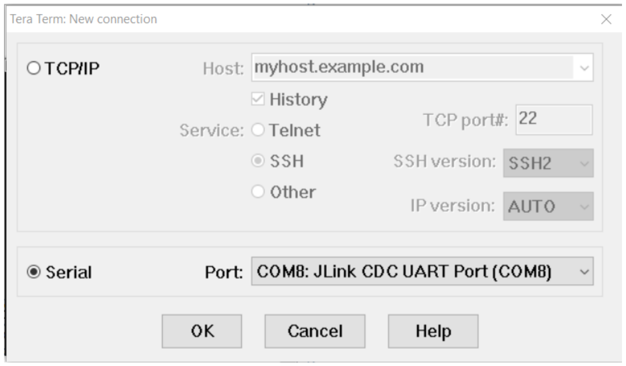
```
Macro        Value
_/_          \\
ShExtension  .bat
Device       ATSAMA5D27CD1G
ProjectDir   "D:\wsg_wwifi_winc_intg\wireless_wifi\apps\ap_scan\firmwa
```

Options affecting hex file:

□ Insert unprotected checksum in user ID memory

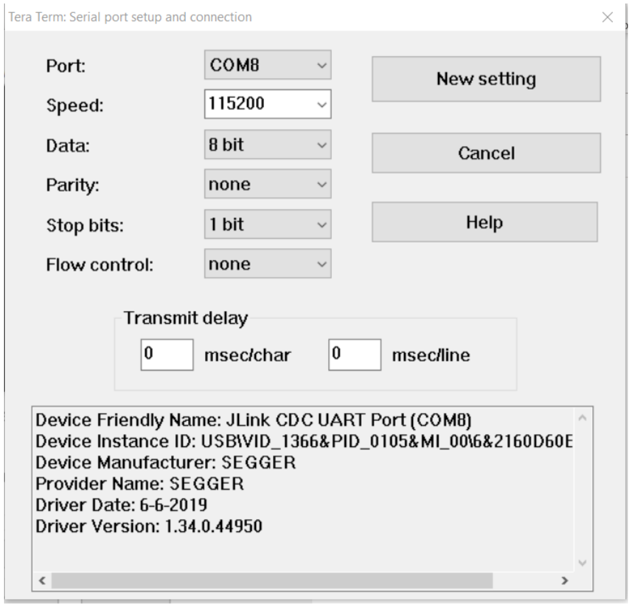
□ Normalize hex file

24. Once the tool is set, generate the project, and do a clean build
25. Open Teraterm with the following COM Port settings:



○ TCP/IP Host: myhost.example.com
☑ History
Service: ○ Telnet TCP port#: 22
● SSH SSH version: SSH2
○ Other IP version: AUTO

● Serial Port: COM8: JLink CDC UART Port (COM8)

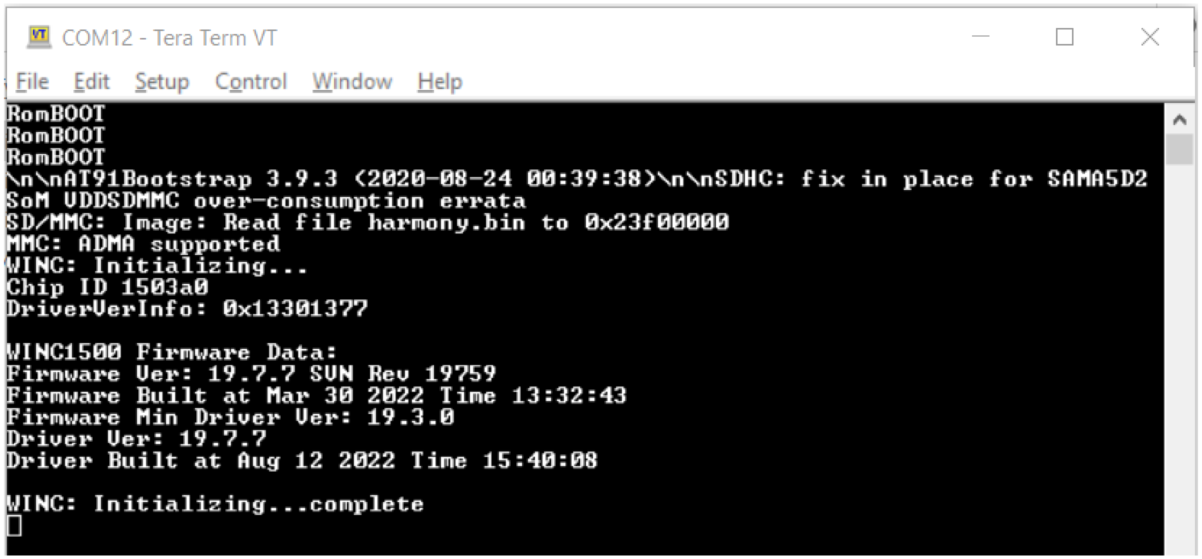


Port: COM8
Speed: 115200
Data: 8 bit
Parity: none
Stop bits: 1 bit
Flow control: none

Transmit delay
0 msec/char 0 msec/line

```
Device Friendly Name: JLink CDC UART Port (COM8)
Device Instance ID: USB\VID_1366&PID_0105&MI_00\6&2160D60E
Device Manufacturer: SEGGER
Provider Name: SEGGER
Driver Date: 6-6-2019
Driver Version: 1.34.0.44950
```

26. Now Debug / Run Main Project (application project) from the MPLAB X IDE
27. Note the logs on the UART Console through the TeraTerm Application

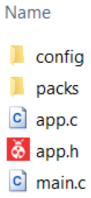


```
RomBOOT
RomBOOT
RomBOOT
\n\nAT91Bootstrap 3.9.3 (2020-08-24 00:39:38)\n\nSDHC: fix in place for SAMA5D2
SoM VDDSDMMC over-consumption errata
SD/MMC: Image: Read file harmony.bin to 0x23f00000
MMC: ADMA supported
WINC: Initializing...
Chip ID 1503a0
DriverVerInfo: 0x13301377

WINC1500 Firmware Data:
Firmware Ver: 19.7.7 SVN Rev 19759
Firmware Built at Mar 30 2022 Time 13:32:43
Firmware Min Driver Ver: 19.3.0
Driver Ver: 19.7.7
Driver Built at Aug 12 2022 Time 15:40:08

WINC: Initializing...complete
```

28. For WINC Driver Init the default source code generated along with important files used for execution are at the location (for check-ins):

a. ap_scan(for WINC DRV INIT)/firmware/src
    - config folder
    - packs folder
    - app.c
    - app.h
    - main.c
    - example.c

….. \wireless_wifi\apps\ap_scan (for WINC DRV INIT)\firmware\src (folder snap below)



Name
config
packs
app.c
app.h
main.c

b. ap_scan(for WINC DRV INIT)/firmware/ sam_a5d27_som1_winc1500.X/nbproject/
    1. configurations.xml
    2. project.xml

c. ap_scan(for WINC DRV INIT)/firmware/ sam_a5d27_som1_winc1500.X/
    3. Makefile

d. ap_scan(for WINC DRV INIT)/firmware/at91bootstrap_sam_a5d27_som1_ek.X

Note: WINC Driver Init is completed which the initial steps before putting up an example code for ap_scan

Porting Sample Application with WINC Driver Init Code (E.g., AP_SCAN)

AP_SCAN application

For AP_SCAN application - ap_scan/firmware/

1. Replace the following src files / add them to the src folder – app.c, app.h, main.c, example.c

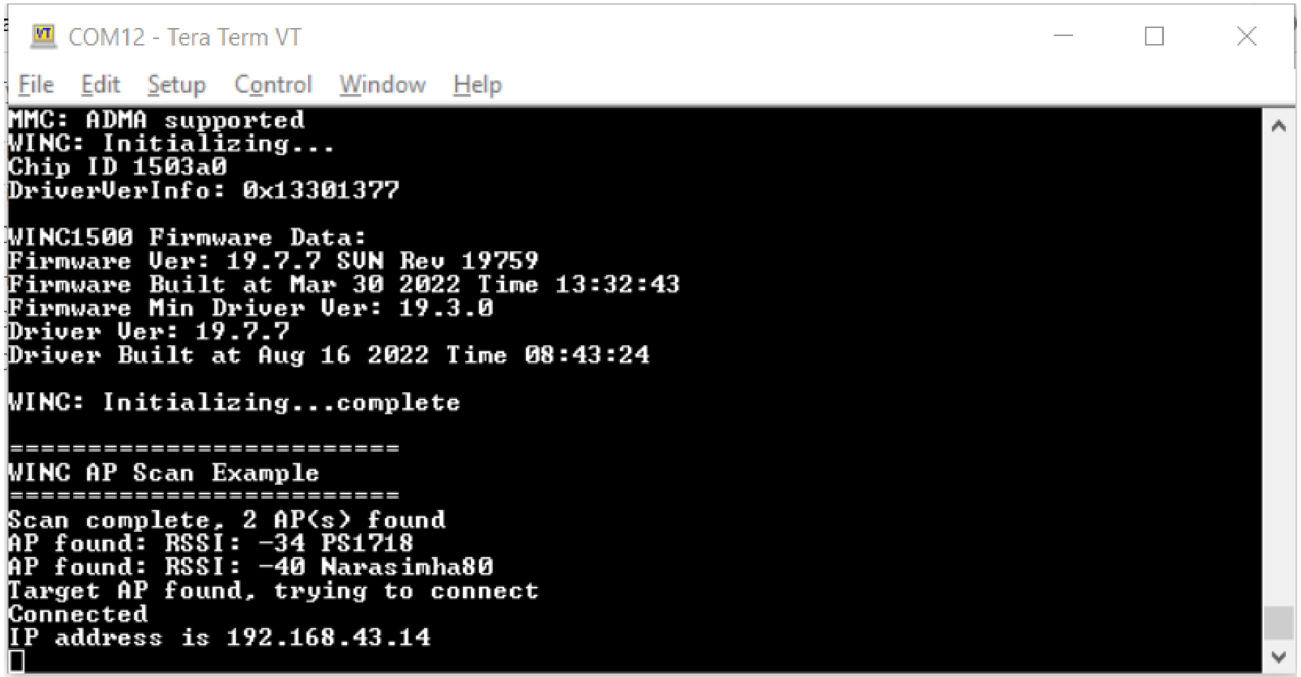
a. /firmware/src
   i. config folder
   ii. packs folder
   iii. app.c
   iv. app.h
   v. main.c
   vi. example.c

b. /firmware/ sam_a5d27_som1_winc1500.X/nbproject/
   i. configurations.xml
   ii. project.xml

c. /firmware/ sam_a5d27_som1_winc1500.X/
   i. Makefile

d. /firmware/at91bootstrap_sam_a5d27_som1_ek.X

Output Screen shows up as follows:



```
MMC: ADMA supported
WINC: Initializing...
Chip ID 1503a0
DriverVerInfo: 0x13301377

WINC1500 Firmware Data:
Firmware Ver: 19.7.7 SVN Rev 19759
Firmware Built at Mar 30 2022 Time 13:32:43
Firmware Min Driver Ver: 19.3.0
Driver Ver: 19.7.7
Driver Built at Aug 16 2022 Time 08:43:24

WINC: Initializing...complete

=========================
WINC AP Scan Example
=========================
Scan complete, 2 AP(s) found
AP found: RSSI: -34 PS1718
AP found: RSSI: -40 Narasimha80
Target AP found, trying to connect
Connected
IP address is 192.168.43.14
```

Only a sample AP_Scan had been described here for a quick reference. All the other sample applications and related configurations are described in the WINC Sample Applications Quick Configurations Guide.