Custom PIC24F Bootloader and workflow

Basic Operation and Logic

The PIC24 bootloader is an independent piece of code residing in address range 0x800 to 0x2000 of the flash program memory of the PIC24, i.e. it uses 6kB of the flash memory to store parameters and bootloader code.

The user application (referred to as APP/Firmware APP in further documentation) is the code that can be updated by the bootloader.

- The bootloader cannot delete itself and the only way to remove the bootloader is to erase the PIC24 using an external programmer or make the APP write to the flash address range reserved for the bootloader.
- The bootloader can download a BIN file over SPP link data mode via BC127 module and flash it on to the PIC24 flash memory. Any BIN file sent to the PIC24 must start from 0x0000 and can extend up to the maximum flash size of the PIC24. Data in the BIN in the range of 0x800 to 0x2000 will be ignored as the bootloader will not overwrite itself.
- When generating APP HEX and converting HEX to BIN, MPLAB IDE can be configured to not use flash range 0x800 – 0x2000 for the APP.
- Bin file should be mutiples of 512 bytes, as written in bootloader code.

Bootloader Update Workflow

- The bootloader can be entered by triggering a software reset of the MCU.
  Before causing a software reset, the phone Bluetooth app must put the BC127 in data mode (SPP profile) and paired.
- The phone bluetooth app and device firmware APP should communicate and decide whether or not to trigger an OTA operation by calling a software reset in MCU
- When the software reset occurs in MCU using phone app, execution is transferred to the bootloader from firmware app.
- Bootloader starts downloading the new bin file in OTA mode from Phone app.
- Download speed is limited because of Bluetooth module’s 9600 bps SPP link - roughly 1 kByte per sec. Therefore, BIN can only be flashed at ~1 kByte per second.
- Bootloader has 16bit fletcher checksum calculation to verify the downloaded bin file is corrupted.
- If the bin file is not corrupted then new firmware bin file is started executed in PIC24F device, otherwise bootloader will restart the OTA download process again
- BC127 goesback to Command mode when Bluetooth is disconnected in phone app once OTA download process is successfully completed.

APP execution conditions

The bootloader will jump to user firmware app under the following conditions:

- If the firmware app is a fresh factory-flashed firmware app, bootloader directly executes it.
- If firmware app has been OTA updated and was verified, bootloader executes it.
- If firmware app was partially OTA updated or corrupted, bootloader switches to recovery mode to redownload new bin file from phone app.

……

GENERATING USER FIRMWARE BIN FILE:

Generating bin file has four major steps:

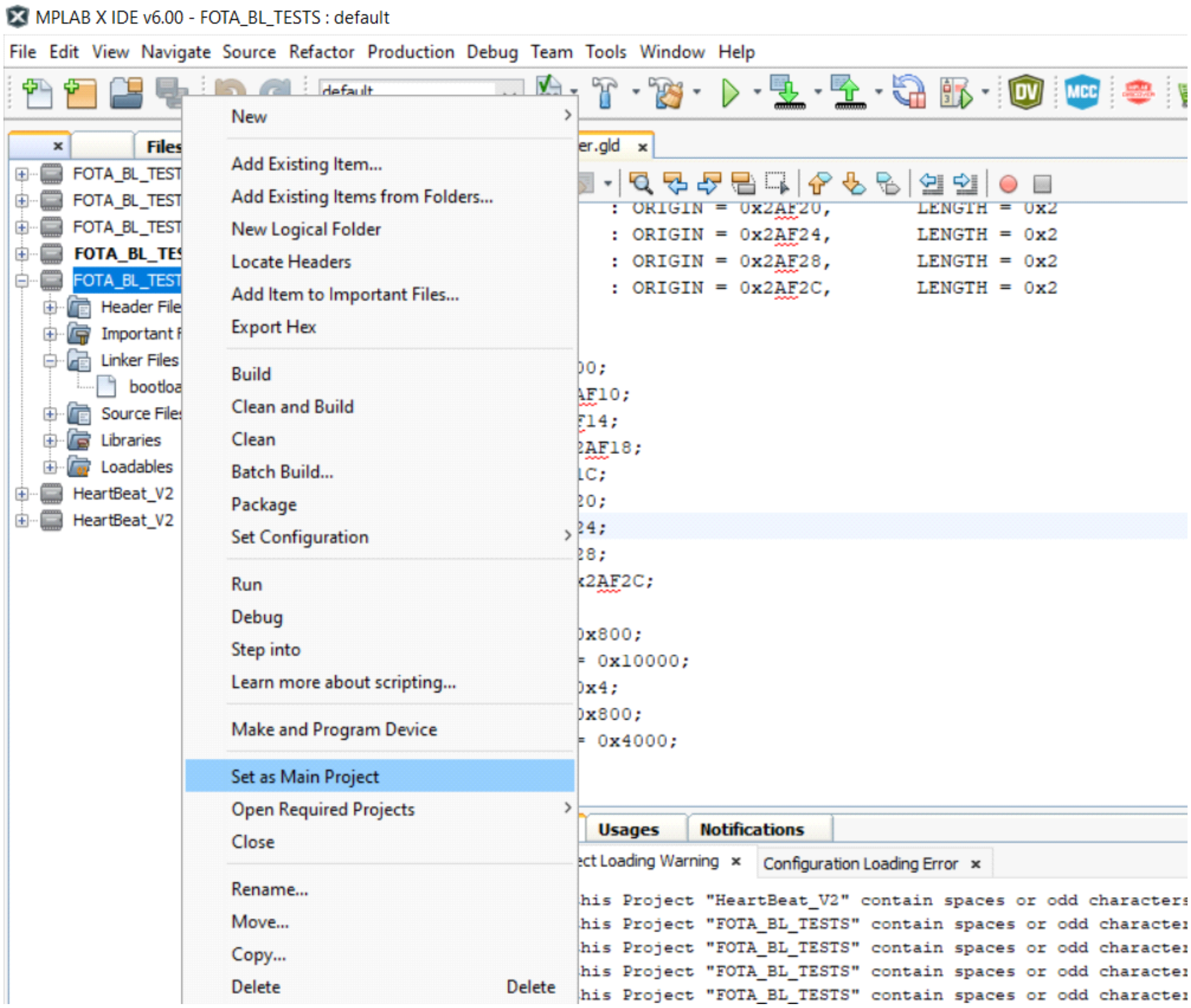
- **Uploading the bootloader code initially using debugger from MPLABX IDE.**
- **Building bootloader friendly firmware app hex file with changes in configuration settings in MPLAB X IDE.**
- **Converting hex file to bin file using makebin.exe tool**
- **Bin file should be converted to multiples of 512 bytes by filling with 0xFF since bootloader accepts only files with 512 byte multiples .**

STEP1 : Bootloader code upload :

- Bootloader code is named as **FOTA_BL_TESTS .** IT has a file named **bootloader.gld** under **Linker Files .**
- Bootloader.gld has details about bootloader address range. the linker file puts the bootloader starting at 0x800. This can be done by setting program(xr) range.
- In the bootloader GLD file, code base and length as set as below :
  __CODE_BASE = 0x800;
  __CODE_LENGTH = 0x10000;
- The above files and settings are already shared with bootloader code ,just make a note and don’t change anything inside it.

UPLOADING PROCESS :

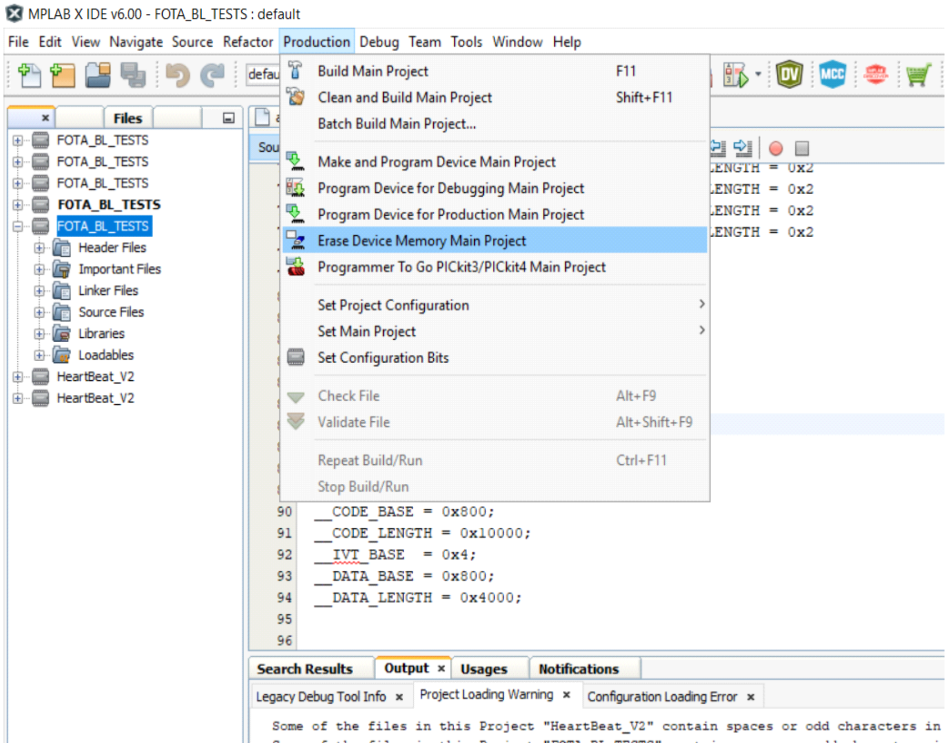
- First select the **FOTA_BL_TESTS as main project** if you have multiple projects.



```
: ORIGIN = 0x2AF20,       LENGTH = 0x2
: ORIGIN = 0x2AF24,       LENGTH = 0x2
: ORIGIN = 0x2AF28,       LENGTH = 0x2
: ORIGIN = 0x2AF2C,       LENGTH = 0x2

)0;
AF10;
F14;
2AF18;
1C;
20;
24;
28;
x2AF2C;

)x800;
= 0x10000;
)x4;
)x800;
= 0x4000;
```

```
his Project "HeartBeat_V2" contain spaces or odd characters
his Project "FOTA_BL_TESTS" contain spaces or odd characte
his Project "FOTA_BL_TESTS" contain spaces or odd characte
his Project "FOTA_BL_TESTS" contain spaces or odd characte
his Project "FOTA_BL_TESTS" contain spaces or odd characte
```

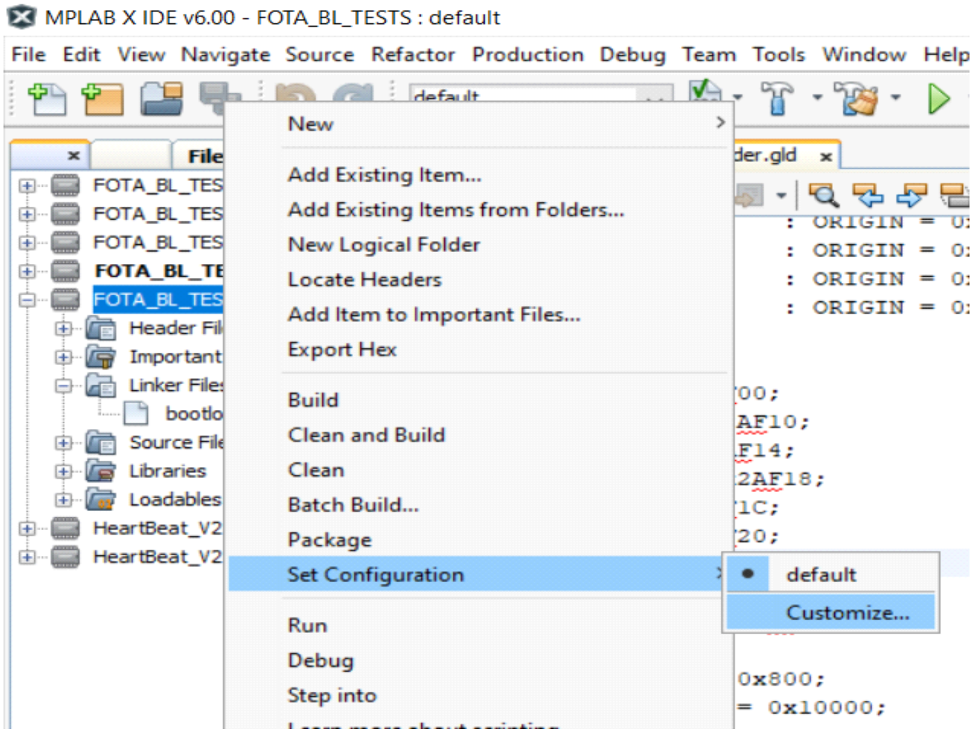
- Erase the Device memory main project and wait till Erase successfully.


MPLAB X IDE v6.00 - FOTA_BL_TESTS : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
Files
FOTA_BL_TESTS
FOTA_BL_TESTS
FOTA_BL_TESTS
FOTA_BL_TESTS
FOTA_BL_TESTS
Header Files
Important Files
Linker Files
Source Files
Libraries
Loadables
HeartBeat_V2
HeartBeat_V2
Build Main Project F11
Clean and Build Main Project Shift+F11
Batch Build Main Project...
Make and Program Device Main Project
Program Device for Debugging Main Project
Program Device for Production Main Project
Erase Device Memory Main Project
Programmer To Go PICkit3/PICkit4 Main Project
Set Project Configuration
Set Main Project
Set Configuration Bits
Check File Alt+F9
Validate File Alt+Shift+F9
Repeat Build/Run Ctrl+F11
Stop Build/Run
Search Results
Output
Usages
Notifications
Legacy Debug Tool Info
Project Loading Warning
Configuration Loading Error

```
LENGTH = 0x2
LENGTH = 0x2
LENGTH = 0x2
LENGTH = 0x2
__CODE_BASE = 0x800;
__CODE_LENGTH = 0x10000;
__IVT_BASE  = 0x4;
__DATA_BASE = 0x800;
__DATA_LENGTH = 0x4000;
```

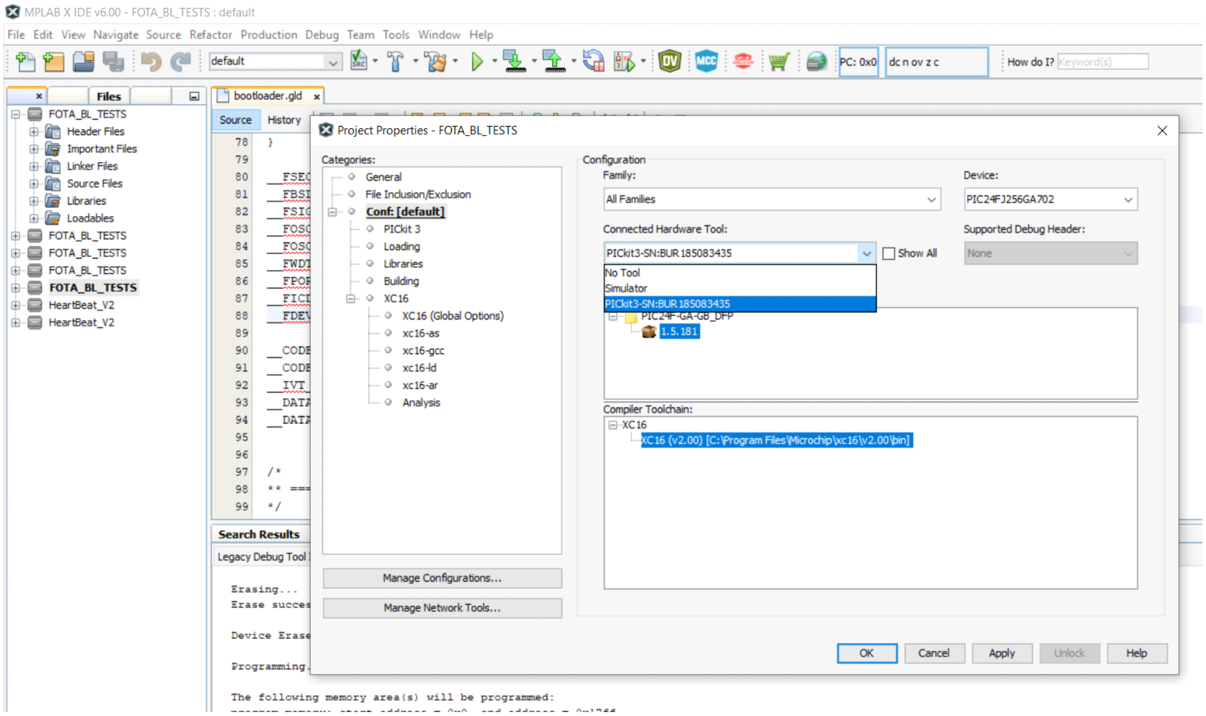
```
Some of the files in this Project "HeartBeat_V2" contain spaces or odd characters in
```

- Select project-> set configuration->customize



```
: ORIGIN = 0:
: ORIGIN = 0:
: ORIGIN = 0:
: ORIGIN = 0:
00;
AF10;
F14;
2AF18;
1C;
20;
0x800;
= 0x10000;
```

- Select tool as PICKIT 3 under hardware tool and apply



Configuration

Family: All Families

Device: PIC24FJ256GA702

Connected Hardware Tool: PICkit3-SN:BUR185083435

No Tool
Simulator
PICkit3-SN:BUR185083435

□ Show All

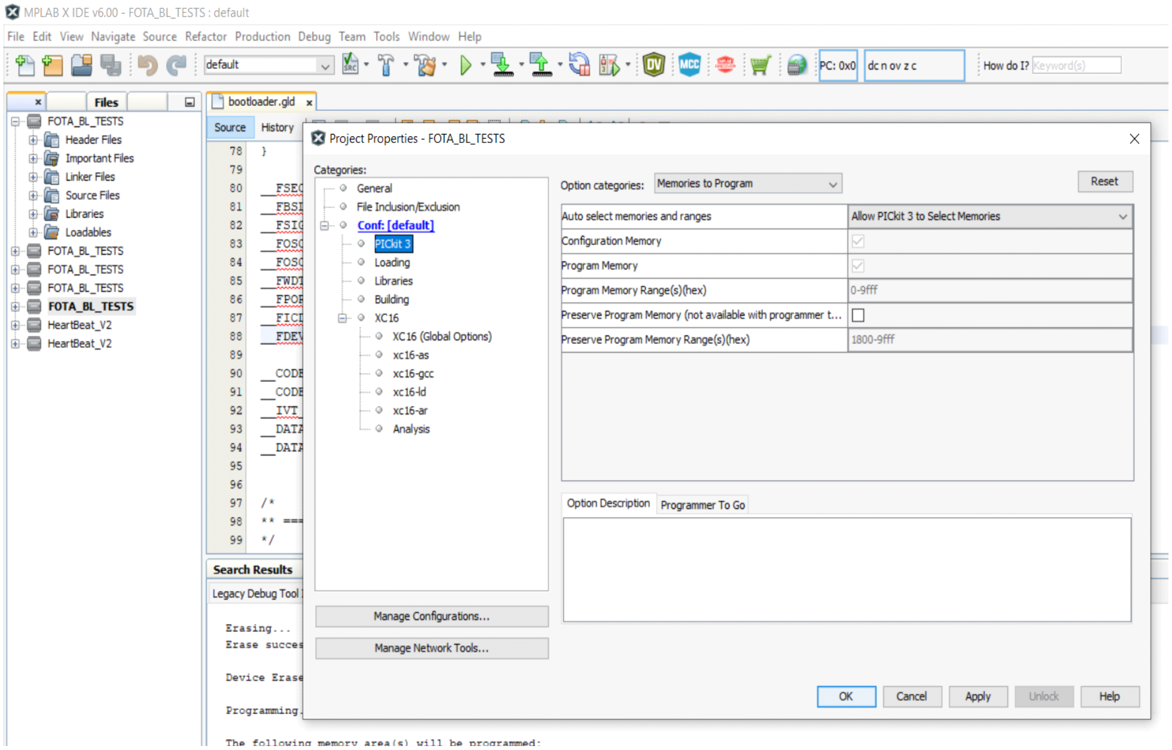
Supported Debug Header: None

PIC24F-GA-GB_DFP
1.5.181

Compiler Toolchain:
XC16
XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]

```
Erasing...
Erase succes

Device Erase

Programming.

The following memory area(s) will be programmed:
```

- Select PICKIT3 and memory setting as shown below : Allow PICKIT3 to select memories and uncheck preserve program memory



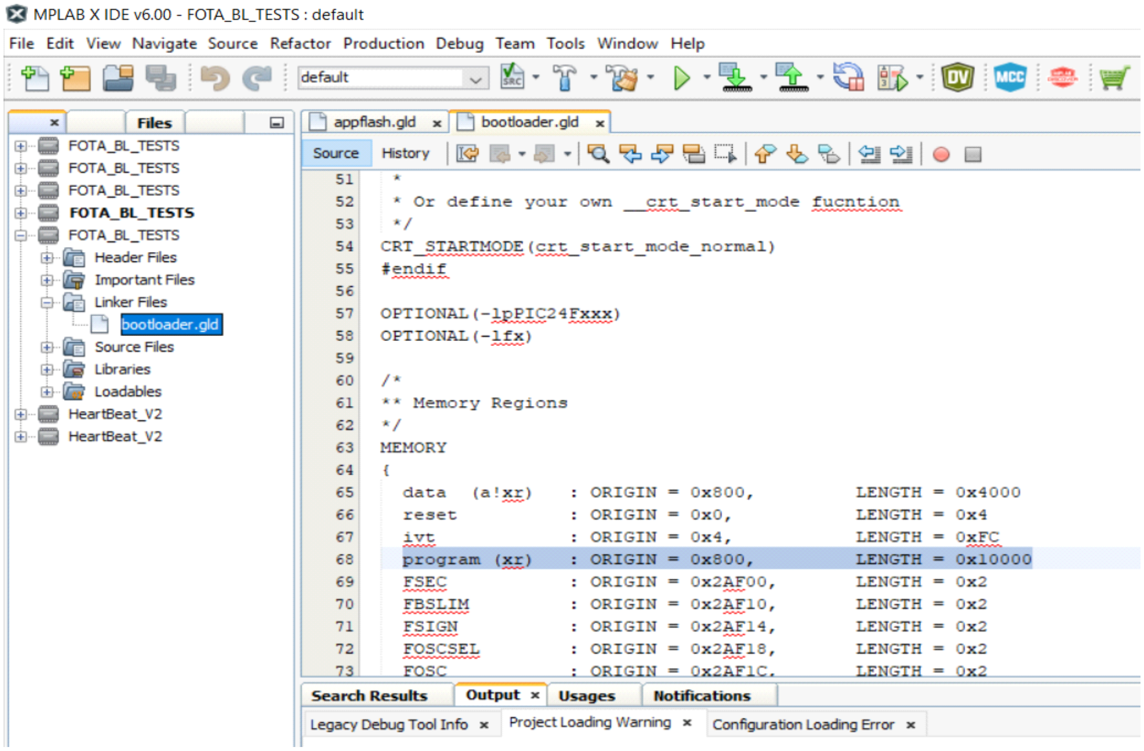
Option categories: Memories to Program

Auto select memories and ranges	Allow PICkit 3 to Select Memories
Configuration Memory	☑
Program Memory	☑
Program Memory Range(s)(hex)	0-9fff
Preserve Program Memory (not available with programmer t...	□
Preserve Program Memory Range(s)(hex)	1800-9fff

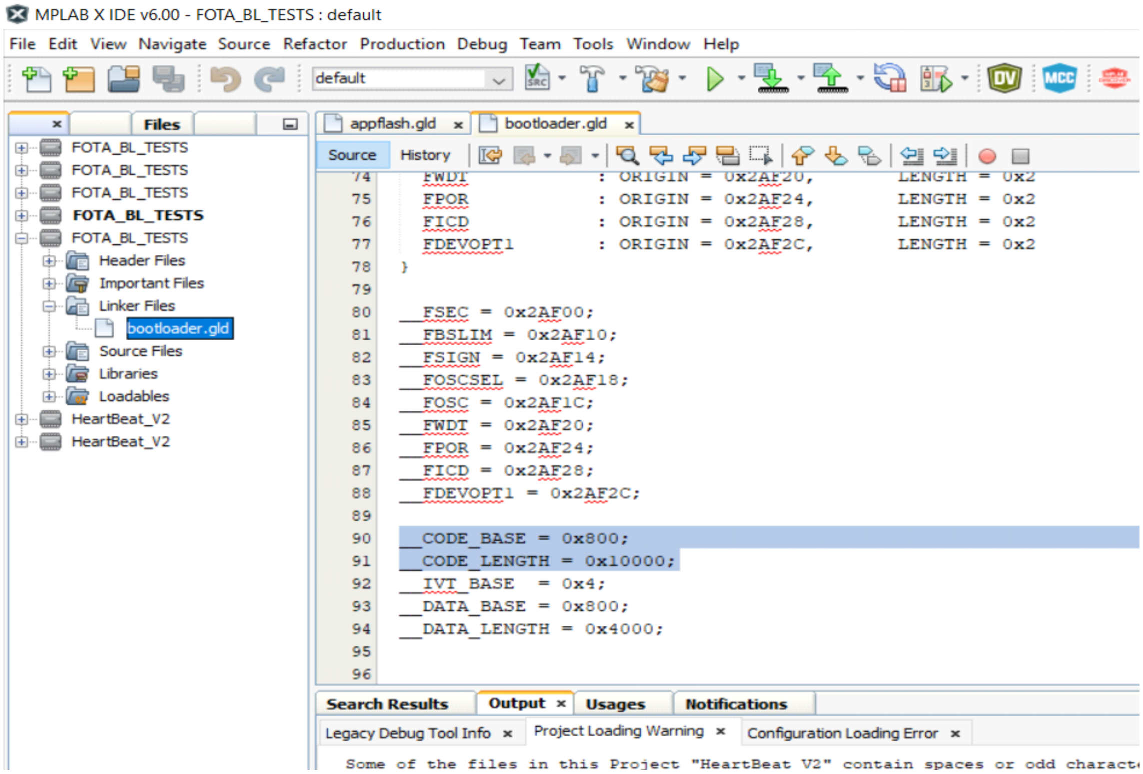
```
Erasing...
Erase succes

Device Erase

Programming.

The following memory area(s) will be programmed:
```

- Now upload the bootloader code into PIC24F device.

- Bootloader GLD file details under linker files shared below .



```
 *
 * Or define your own __crt_start_mode fucntion
 */
CRT_STARTMODE(crt_start_mode_normal)
#endif

OPTIONAL(-lpPIC24Fxxx)
OPTIONAL(-lfx)

/*
** Memory Regions
*/
MEMORY
{
  data  (a!xr)   : ORIGIN = 0x800,         LENGTH = 0x4000
  reset          : ORIGIN = 0x0,           LENGTH = 0x4
  ivt            : ORIGIN = 0x4,           LENGTH = 0xFC
  program (xr)   : ORIGIN = 0x800,         LENGTH = 0x10000
  FSEC           : ORIGIN = 0x2AF00,       LENGTH = 0x2
  FBSLIM         : ORIGIN = 0x2AF10,       LENGTH = 0x2
  FSIGN          : ORIGIN = 0x2AF14,       LENGTH = 0x2
  FOSCSEL        : ORIGIN = 0x2AF18,       LENGTH = 0x2
  FOSC           : ORIGIN = 0x2AF1C,       LENGTH = 0x2
```



```
  FWDT           : ORIGIN = 0x2AF20,       LENGTH = 0x2
  FPOR           : ORIGIN = 0x2AF24,       LENGTH = 0x2
  FICD           : ORIGIN = 0x2AF28,       LENGTH = 0x2
  FDEVOPT1       : ORIGIN = 0x2AF2C,       LENGTH = 0x2
}

__FSEC = 0x2AF00;
__FBSLIM = 0x2AF10;
__FSIGN = 0x2AF14;
__FOSCSEL = 0x2AF18;
__FOSC = 0x2AF1C;
__FWDT = 0x2AF20;
__FPOR = 0x2AF24;
__FICD = 0x2AF28;
__FDEVOPT1 = 0x2AF2C;

__CODE_BASE = 0x800;
__CODE_LENGTH = 0x10000;
__IVT_BASE  = 0x4;
__DATA_BASE = 0x800;
__DATA_LENGTH = 0x4000;


```

```
Some of the files in this Project "HeartBeat V2" contain spaces or odd charact
```

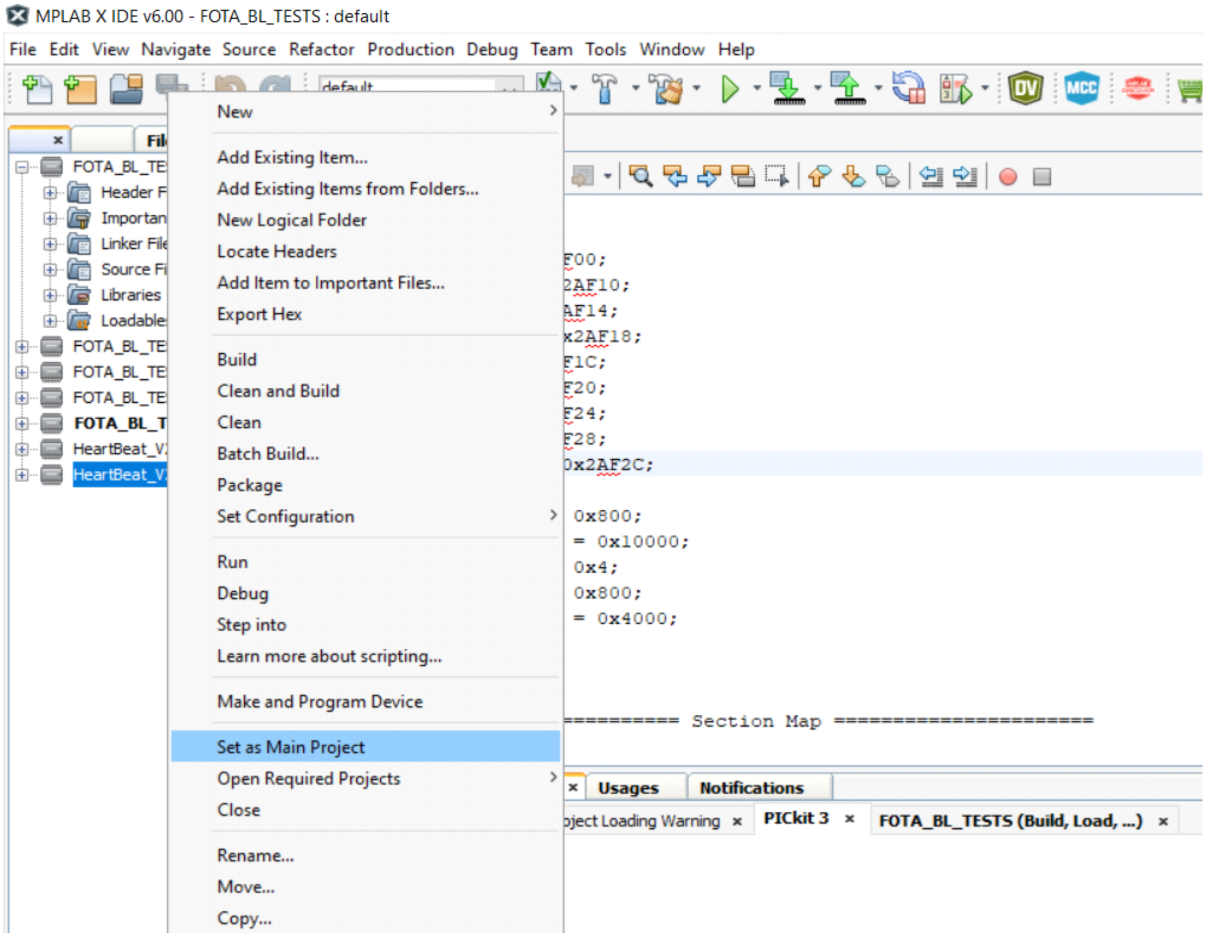
STEP2: BUILDING BOOTLOADER FRIENDLY USER HEX FILE

- Keep the firmware file shared as **base file** and make any changes for future versions. Because it has **flash memory configuration settings and software reset function** included in it.
- User firmware code is named named HEARTBEAT_V2 . It has a file named **app.gld** under **Linker Files .**
- app.gld has details about user code address range. the linker file puts the user firmware code starting at 0x2000. This can be done by setting program(xr) range.
- In the app GLD file, code base and length as set as below :
  __CODE_BASE = 0x2000;
  __CODE_LENGTH = 0x16F00;

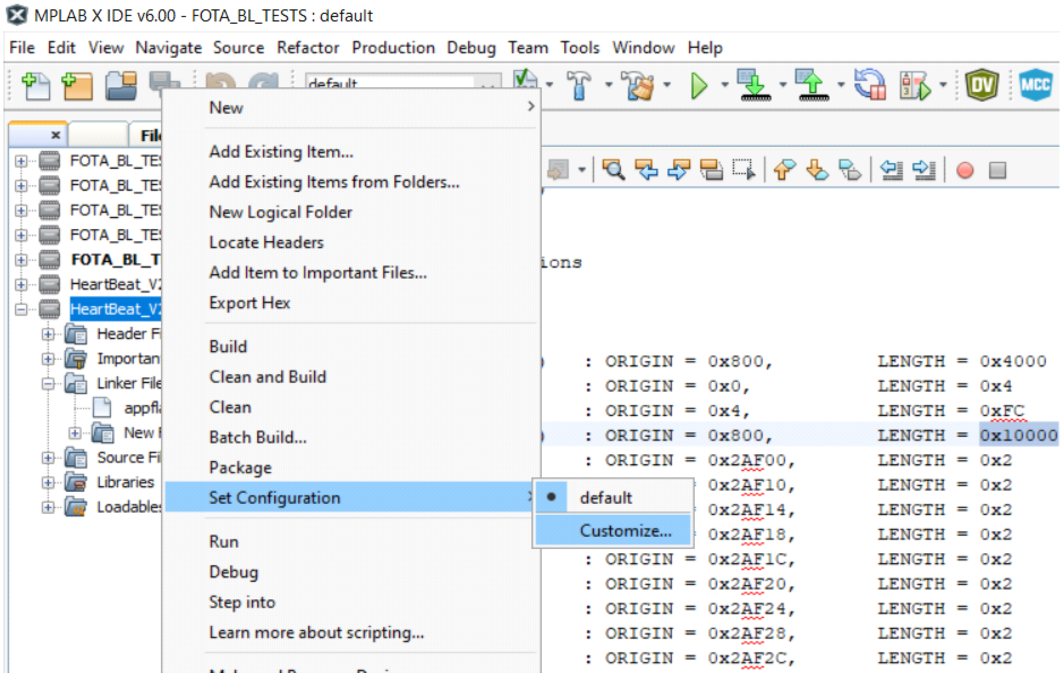
The above files and settings are already shared with user firmware code , just make a note and don't change anything inside it.

Making HEX FILE:

- Select HeartBeat_V2 as **main project** if you have multiple projects.

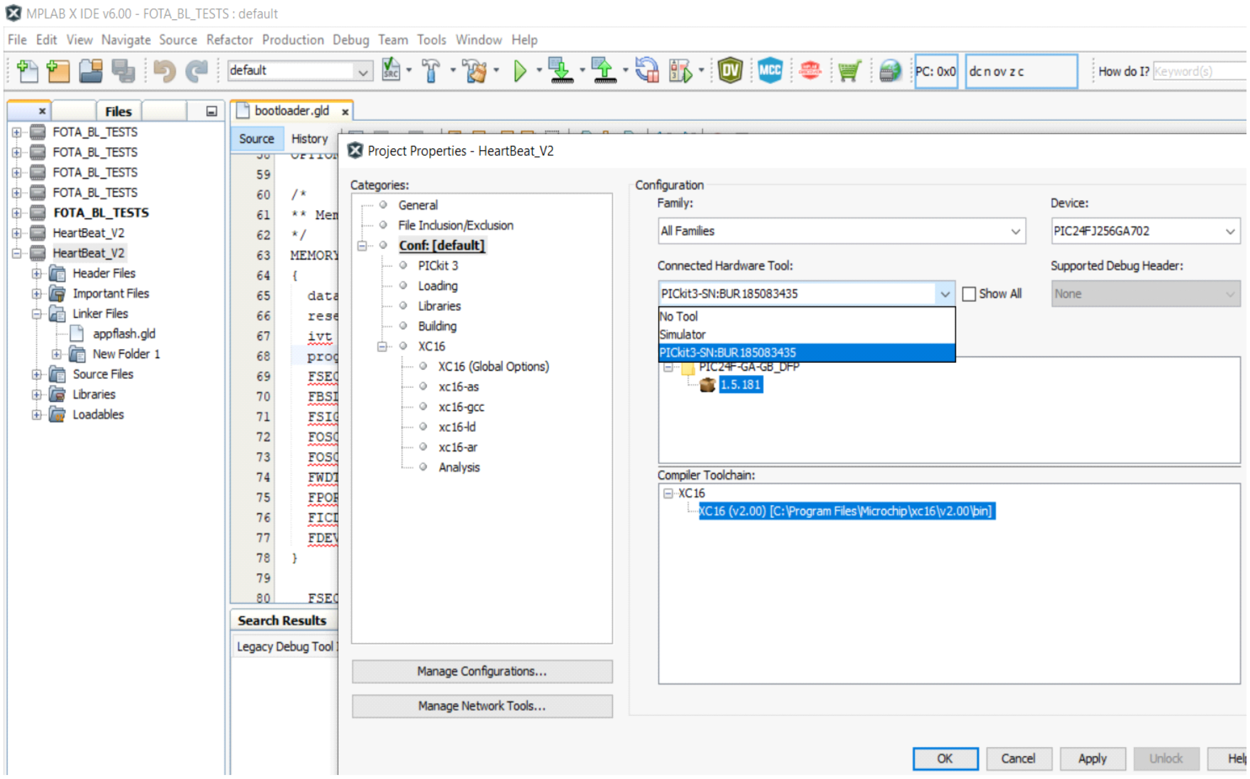


- Select project-> set configuration->customize



```
ions

: ORIGIN = 0x800,         LENGTH = 0x4000
: ORIGIN = 0x0,           LENGTH = 0x4
: ORIGIN = 0x4,           LENGTH = 0xFC
: ORIGIN = 0x800,         LENGTH = 0x10000
: ORIGIN = 0x2AF00,       LENGTH = 0x2
: ORIGIN = 0x2AF10,       LENGTH = 0x2
: ORIGIN = 0x2AF14,       LENGTH = 0x2
: ORIGIN = 0x2AF18,       LENGTH = 0x2
: ORIGIN = 0x2AF1C,       LENGTH = 0x2
: ORIGIN = 0x2AF20,       LENGTH = 0x2
: ORIGIN = 0x2AF24,       LENGTH = 0x2
: ORIGIN = 0x2AF28,       LENGTH = 0x2
: ORIGIN = 0x2AF2C,       LENGTH = 0x2
```

- Select tool as PICKIT 3 under hardware tool and apply



Configuration
Family: All Families
Device: PIC24FJ256GA702
Connected Hardware Tool: PICkit3-SN:BUR185083435
Supported Debug Header: None
Compiler Toolchain: XC16
XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]

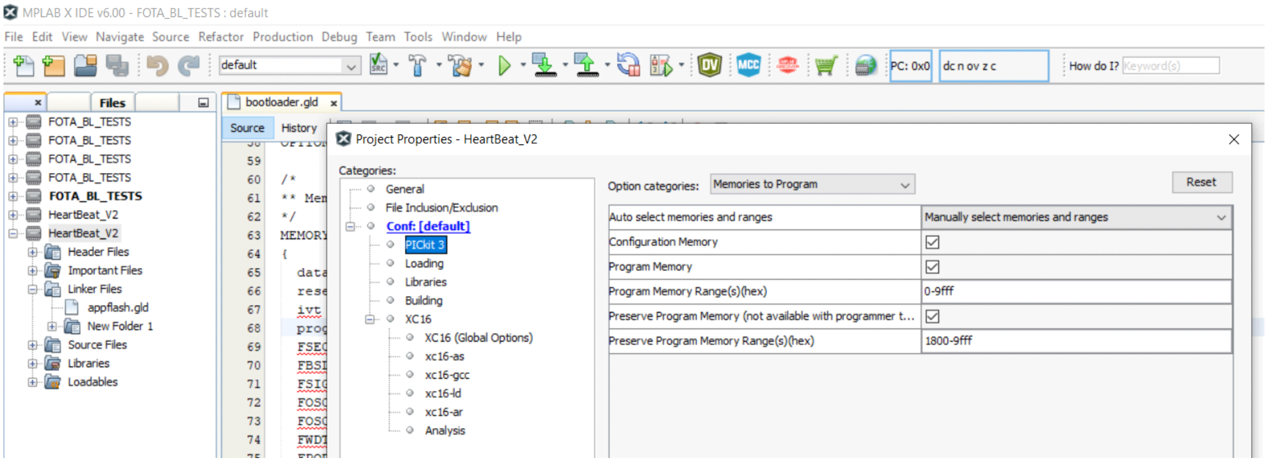
- Select PICKIT3 and memory setting as shown below :

Manually select memories and range .

program memory range **-0 -9fff**.

Select preserve program memory.

Preserve program memory – **1800 -9fff .** and select apply,ok.



Project Properties - HeartBeat_V2

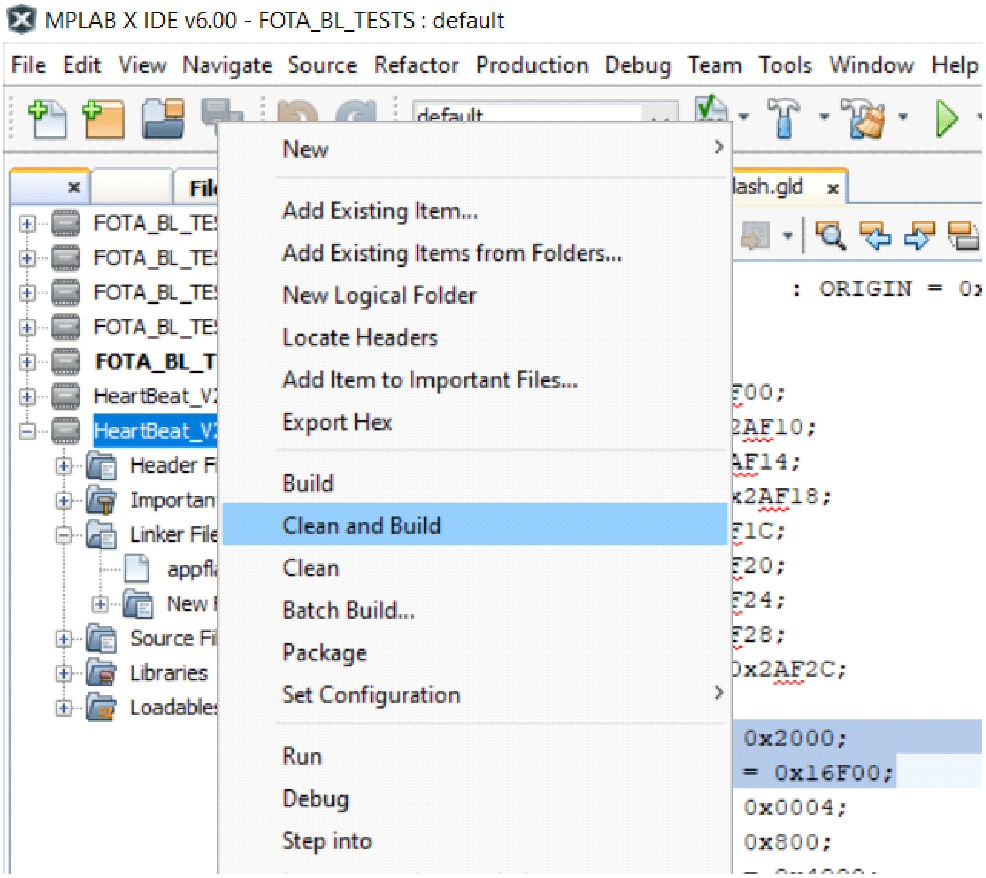
Categories:

- General
- File Inclusion/Exclusion
- Conf: [default]
  - PICkit 3
  - Loading
  - Libraries
  - Building
  - XC16
    - XC16 (Global Options)
    - xc16-as
    - xc16-gcc
    - xc16-ld
    - xc16-ar
    - Analysis

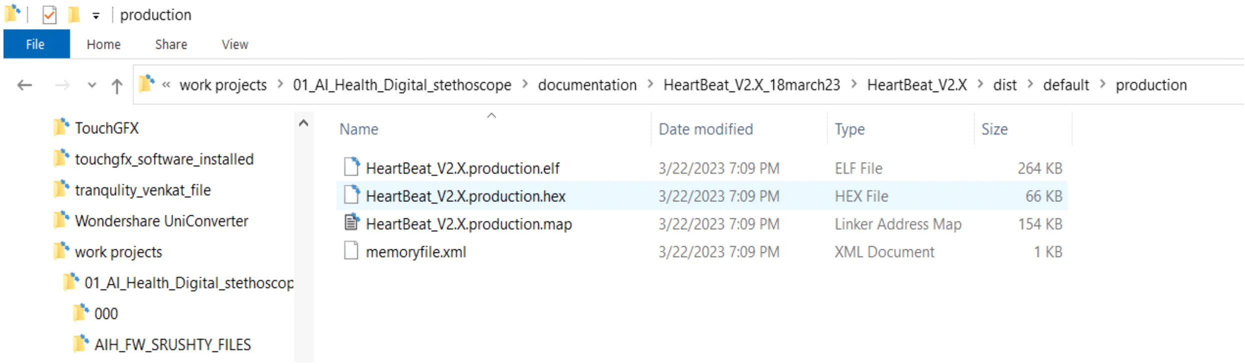
Option categories: Memories to Program

Auto select memories and ranges	Manually select memories and ranges
Configuration Memory	☑
Program Memory	☑
Program Memory Range(s)(hex)	0-9fff
Preserve Program Memory (not available with programmer t...	☑
Preserve Program Memory Range(s)(hex)	1800-9fff

- Now clean and build the project

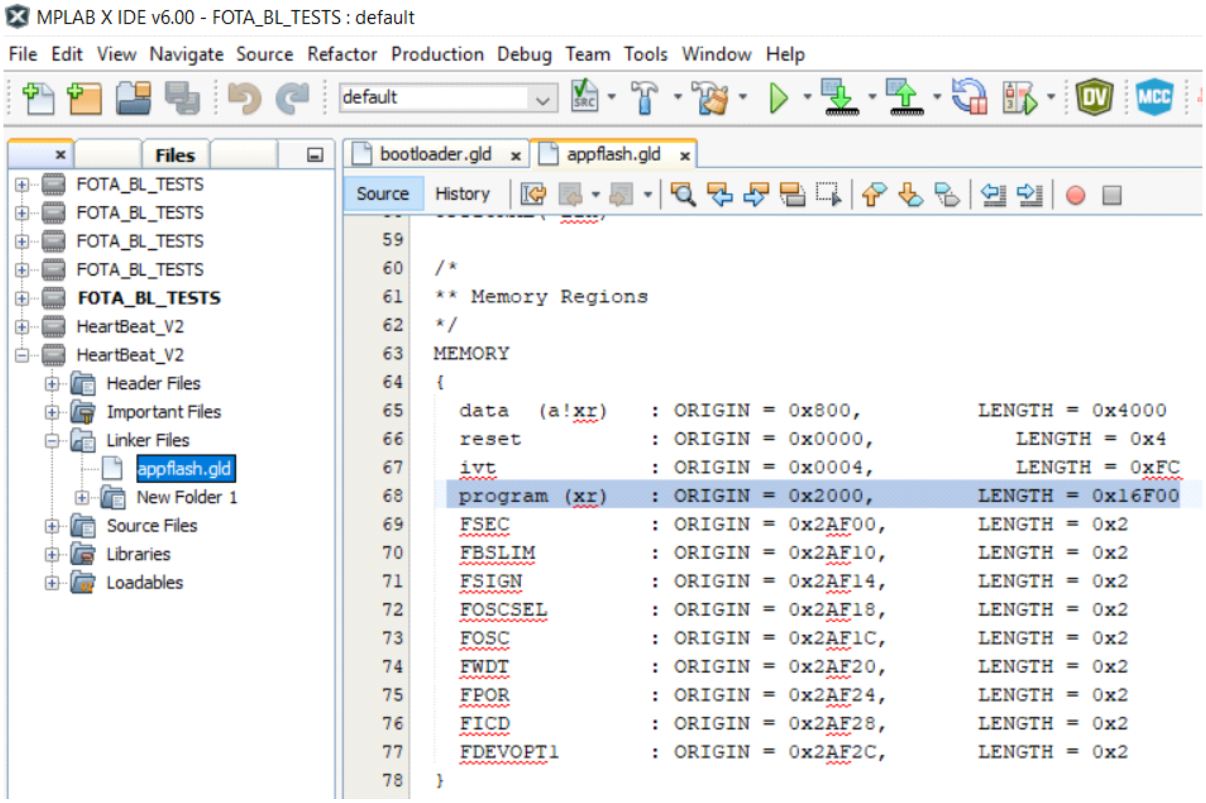


- **HEX** file is generated under project folder location . Copy and save the hex file, which will be used to change to bin file.

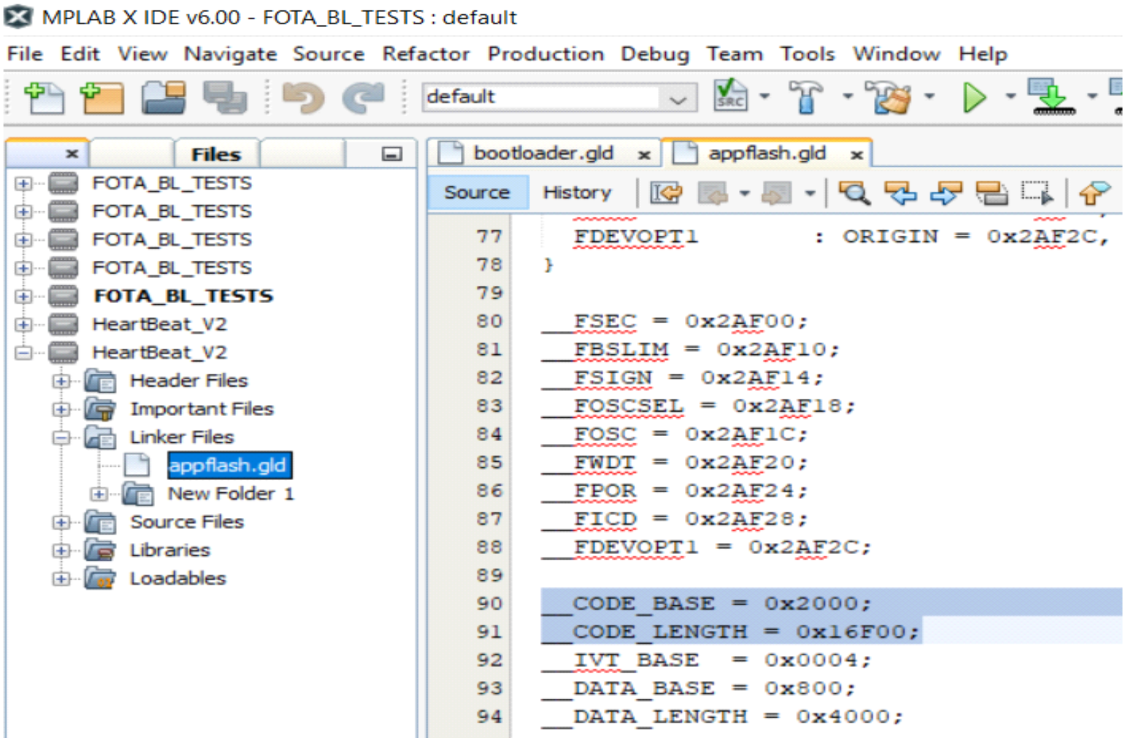

production
File
Home
Share
View
« work projects › 01_AI_Health_Digital_stethoscope › documentation › HeartBeat_V2.X_18march23 › HeartBeat_V2.X › dist › default › production
TouchGFX
touchgfx_software_installed
tranqulity_venkat_file
Wondershare UniConverter
work projects
01_AI_Health_Digital_stethoscop
000
AIH_FW_SRUSHTY_FILES

Name	Date modified	Type	Size
HeartBeat_V2.X.production.elf	3/22/2023 7:09 PM	ELF File	264 KB
HeartBeat_V2.X.production.hex	3/22/2023 7:09 PM	HEX File	66 KB
HeartBeat_V2.X.production.map	3/22/2023 7:09 PM	Linker Address Map	154 KB
memoryfile.xml	3/22/2023 7:09 PM	XML Document	1 KB

- **app.gld** file details are shared below.



```
/*
** Memory Regions
*/
MEMORY
{
  data   (a!xr)    : ORIGIN = 0x800,        LENGTH = 0x4000
  reset            : ORIGIN = 0x0000,          LENGTH = 0x4
  ivt              : ORIGIN = 0x0004,          LENGTH = 0xFC
  program  (xr)    : ORIGIN = 0x2000,       LENGTH = 0x16F00
  FSEC             : ORIGIN = 0x2AF00,      LENGTH = 0x2
  FBSLIM           : ORIGIN = 0x2AF10,      LENGTH = 0x2
  FSIGN            : ORIGIN = 0x2AF14,      LENGTH = 0x2
  FOSCSEL          : ORIGIN = 0x2AF18,      LENGTH = 0x2
  FOSC             : ORIGIN = 0x2AF1C,      LENGTH = 0x2
  FWDT             : ORIGIN = 0x2AF20,      LENGTH = 0x2
  FPOR             : ORIGIN = 0x2AF24,      LENGTH = 0x2
  FICD             : ORIGIN = 0x2AF28,      LENGTH = 0x2
  FDEVOPT1         : ORIGIN = 0x2AF2C,      LENGTH = 0x2
}
```

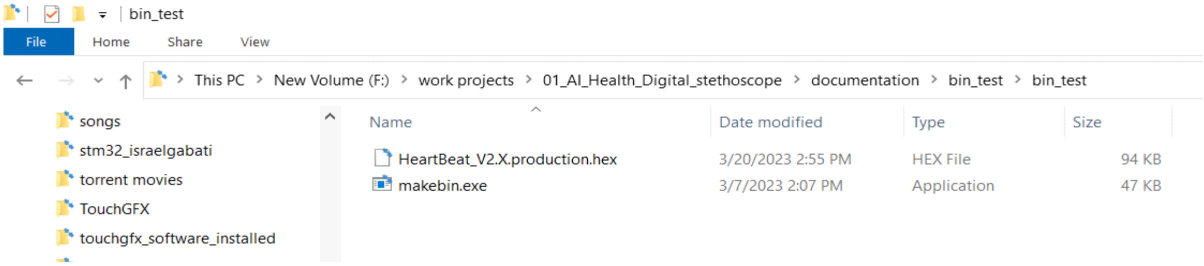


```
  FDEVOPT1          : ORIGIN = 0x2AF2C,
}

__FSEC = 0x2AF00;
__FBSLIM = 0x2AF10;
__FSIGN = 0x2AF14;
__FOSCSEL = 0x2AF18;
__FOSC = 0x2AF1C;
__FWDT = 0x2AF20;
__FPOR = 0x2AF24;
__FICD = 0x2AF28;
__FDEVOPT1 = 0x2AF2C;

__CODE_BASE = 0x2000;
__CODE_LENGTH = 0x16F00;
__IVT_BASE   = 0x0004;
__DATA_BASE = 0x800;
__DATA_LENGTH = 0x4000;
```

STEP3 : HEX FILE TO BIN FILE CREATION

- We use a tool named **makebin.exe** for hex to bin file conversion

- The **Hex file** to be converted and **makebin.exe** tool should be in same folder as below.



bin_test
File
Home
Share
View
This PC > New Volume (F:) > work projects > 01_AI_Health_Digital_stethoscope > documentation > bin_test > bin_test
songs
stm32_israelgabati
torrent movies
TouchGFX
touchgfx_software_installed

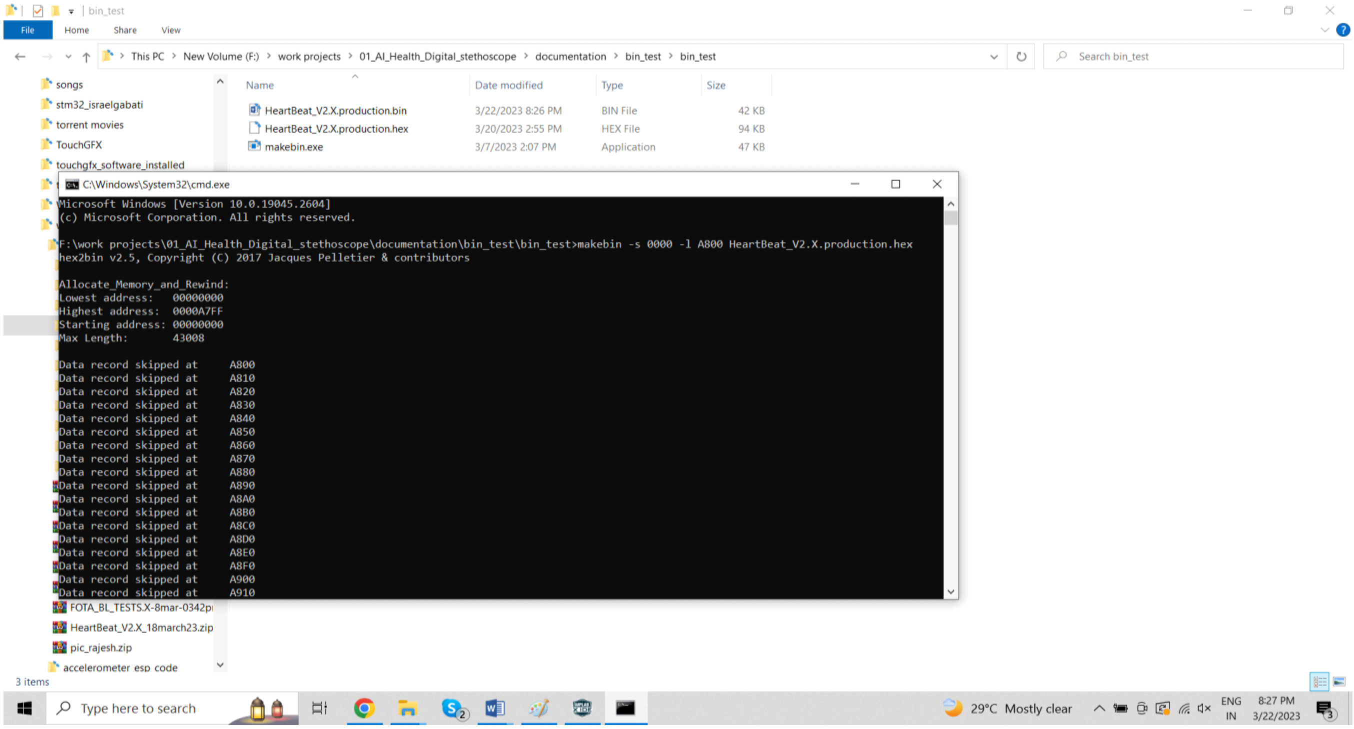
Name	Date modified	Type	Size
HeartBeat_V2.X.production.hex	3/20/2023 2:55 PM	HEX File	94 KB
makebin.exe	3/7/2023 2:07 PM	Application	47 KB

- To generate BIN file for bootloader, in Windows, place the HEX and makebin.exe in same folder, use command line :

  makebin –s 0000 –l *len filename.hex*
  where len = 28F00 for entire PIC24 flash, but can be set to a smaller number that covers APP flash size. Setting smaller size will help reduce BIN file size. i.e. if app uses 0x2000 to 0x15600, len can be 0x16000.

  Example: **makebin -s 0000 -l A800 HeartBeat_V2.X.production.hex**

  Where A800 – is the len (length that should be multiples of **512 bytes** and minimum range where data should not be skipped) . will check in next step how to calculate len(length).



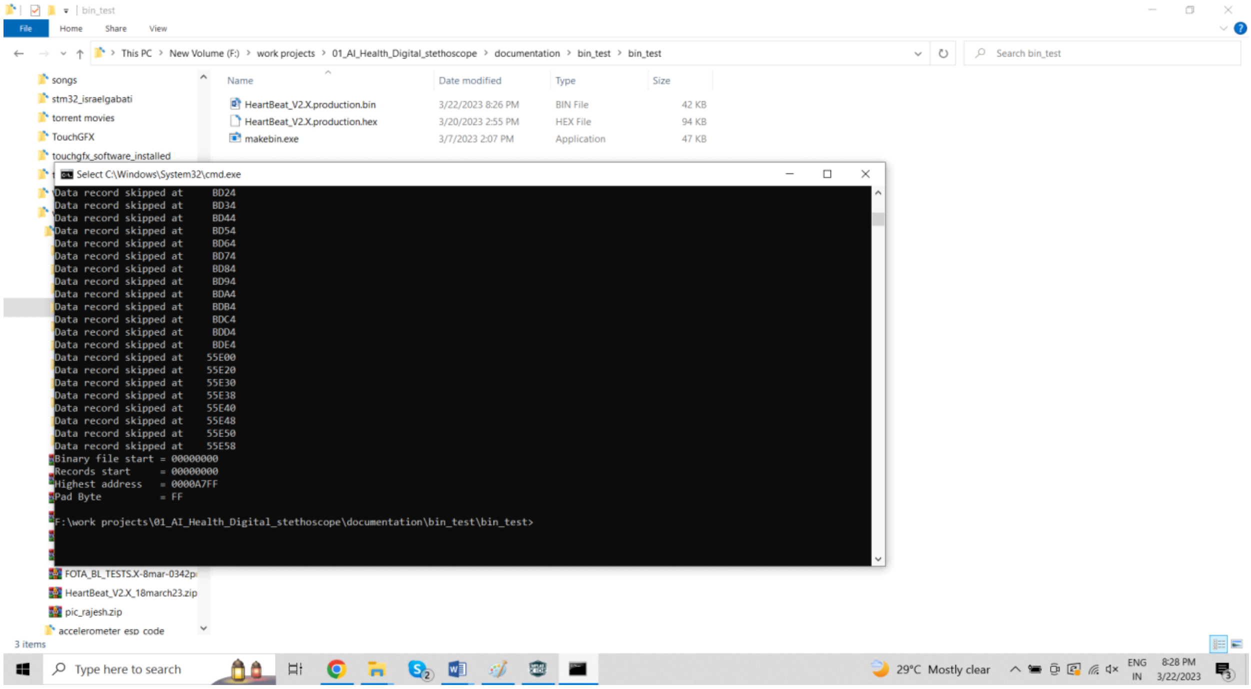
bin_test
File
Home
Share
View
This PC > New Volume (F:) > work projects > 01_AI_Health_Digital_stethoscope > documentation > bin_test > bin_test
Search bin_test
songs
stm32_israelgabati
torrent movies
TouchGFX
touchgfx_software_installed
FOTA_BL_TESTS.X-8mar-0342p
HeartBeat_V2.X_18march23.zip
pic_rajesh.zip
accelerometer esp code
3 items
Type here to search
29°C Mostly clear
ENG
IN
8:27 PM
3/22/2023
C:\Windows\System32\cmd.exe

Name	Date modified	Type	Size
HeartBeat_V2.X.production.bin	3/22/2023 8:26 PM	BIN File	42 KB
HeartBeat_V2.X.production.hex	3/20/2023 2:55 PM	HEX File	94 KB
makebin.exe	3/7/2023 2:07 PM	Application	47 KB

```
Microsoft Windows [Version 10.0.19045.2604]
(c) Microsoft Corporation. All rights reserved.

F:\work projects\01_AI_Health_Digital_stethoscope\documentation\bin_test\bin_test>makebin -s 0000 -l A800 HeartBeat_V2.X.production.hex
hex2bin v2.5, Copyright (C) 2017 Jacques Pelletier & contributors

Allocate_Memory_and_Rewind:
Lowest address:   00000000
Highest address:  0000A7FF
Starting address: 00000000
Max Length:       43008

Data record skipped at     A800
Data record skipped at     A810
Data record skipped at     A820
Data record skipped at     A830
Data record skipped at     A840
Data record skipped at     A850
Data record skipped at     A860
Data record skipped at     A870
Data record skipped at     A880
Data record skipped at     A890
Data record skipped at     A8A0
Data record skipped at     A8B0
Data record skipped at     A8C0
Data record skipped at     A8D0
Data record skipped at     A8E0
Data record skipped at     A8F0
Data record skipped at     A900
Data record skipped at     A910
```

- Data record can be skipped after memory address 55E30 , not at memory addresses before that as shown above.

- We have to select a suitable **len** value of multiples of **512bytes** and also not skip data record before memory address 55E00. Explained in step4 to solve this issue.



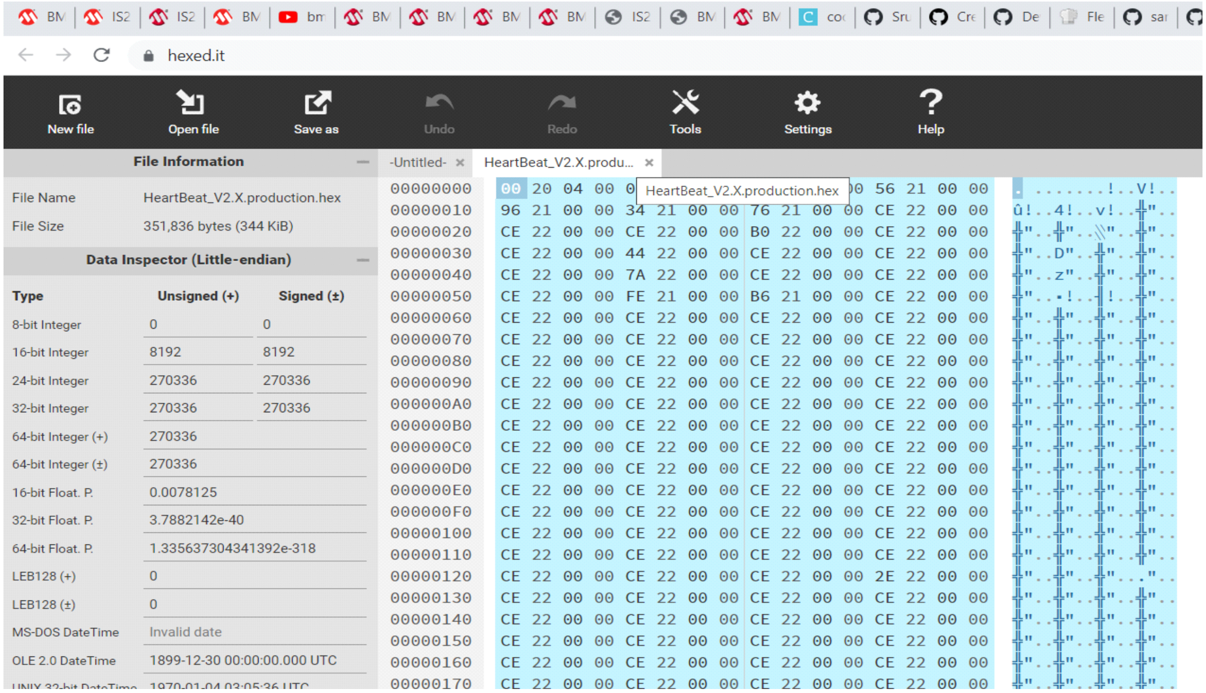
bin_test
File
Home
Share
View
This PC › New Volume (F:) › work projects › 01_AI_Health_Digital_stethoscope › documentation › bin_test › bin_test
Search bin_test
songs
stm32_israelgabati
torrent movies
TouchGFX
touchgfx_software_installed
FOTA_BL_TESTS.X-8mar-0342p
HeartBeat_V2.X_18march23.zip
pic_rajesh.zip
accelerometer esp code
3 items
Select C:\Windows\System32\cmd.exe
Type here to search
29°C Mostly clear
ENG IN
8:28 PM 3/22/2023

Name	Date modified	Type	Size
HeartBeat_V2.X.production.bin	3/22/2023 8:26 PM	BIN File	42 KB
HeartBeat_V2.X.production.hex	3/20/2023 2:55 PM	HEX File	94 KB
makebin.exe	3/7/2023 2:07 PM	Application	47 KB

```
Data record skipped at     BD24
Data record skipped at     BD34
Data record skipped at     BD44
Data record skipped at     BD54
Data record skipped at     BD64
Data record skipped at     BD74
Data record skipped at     BD84
Data record skipped at     BD94
Data record skipped at     BDA4
Data record skipped at     BDB4
Data record skipped at     BDC4
Data record skipped at     BDD4
Data record skipped at     BDE4
Data record skipped at    55E00
Data record skipped at    55E20
Data record skipped at    55E30
Data record skipped at    55E38
Data record skipped at    55E40
Data record skipped at    55E48
Data record skipped at    55E50
Data record skipped at    55E58
Binary file start = 00000000
Records start     = 00000000
Highest address   = 0000A7FF
Pad Byte          = FF

F:\work projects\01_AI_Health_Digital_stethoscope\documentation\bin_test\bin_test>
```

STEP4 : CALCULATING LEN VALUE FOR BIN FILE CREATION

- Open this link https://hexed.it/ and click open file and select the hex file to view hex values.



File Information

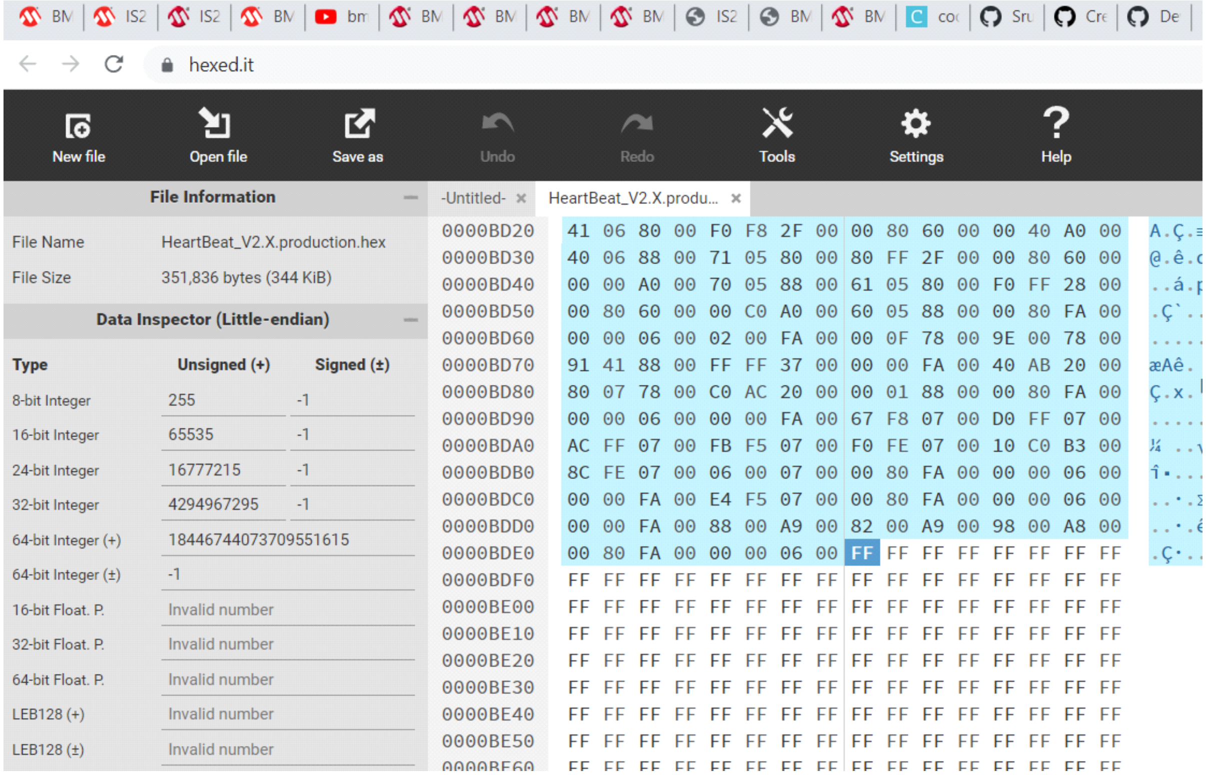
File Name	HeartBeat_V2.X.production.hex
File Size	351,836 bytes (344 KiB)

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (±)
8-bit Integer	0	0
16-bit Integer	8192	8192
24-bit Integer	270336	270336
32-bit Integer	270336	270336
64-bit Integer (+)	270336	
64-bit Integer (±)	270336	
16-bit Float. P.	0.0078125	
32-bit Float. P.	3.7882142e-40	
64-bit Float. P.	1.335637304341392e-318	
LEB128 (+)	0	
LEB128 (±)	0	
MS-DOS DateTime	Invalid date	
OLE 2.0 DateTime	1899-12-30 00:00:00.000 UTC	
UNIX 32-bit DateTime	1970-01-04 03:05:36 UTC	

```
00000000  00 20 04 00 0                      0 56 21 00 00   .........!..V!..
00000010  96 21 00 00 34 21 00 00 76 21 00 00 CE 22 00 00   û!..4!..v!..╫"..
00000020  CE 22 00 00 CE 22 00 00 B0 22 00 00 CE 22 00 00   ╫"..╫"..░"..╫"..
00000030  CE 22 00 00 44 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..D"..╫"..╫"..
00000040  CE 22 00 00 7A 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..z"..╫"..╫"..
00000050  CE 22 00 00 FE 21 00 00 B6 21 00 00 CE 22 00 00   ╫"..■!..╢!..╫"..
00000060  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000070  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000080  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000090  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
000000A0  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
000000B0  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
000000C0  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
000000D0  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
000000E0  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
000000F0  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000100  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000110  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000120  CE 22 00 00 CE 22 00 00 CE 22 00 00 2E 22 00 00   ╫"..╫"..╫"...."..
00000130  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000140  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000150  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000160  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
00000170  CE 22 00 00 CE 22 00 00 CE 22 00 00 CE 22 00 00   ╫"..╫"..╫"..╫"..
```

- We can see the hex values from address 0x00000000. Scroll down and check which address the file data ends and starts filling with FF FF FF FF ……..



File Information

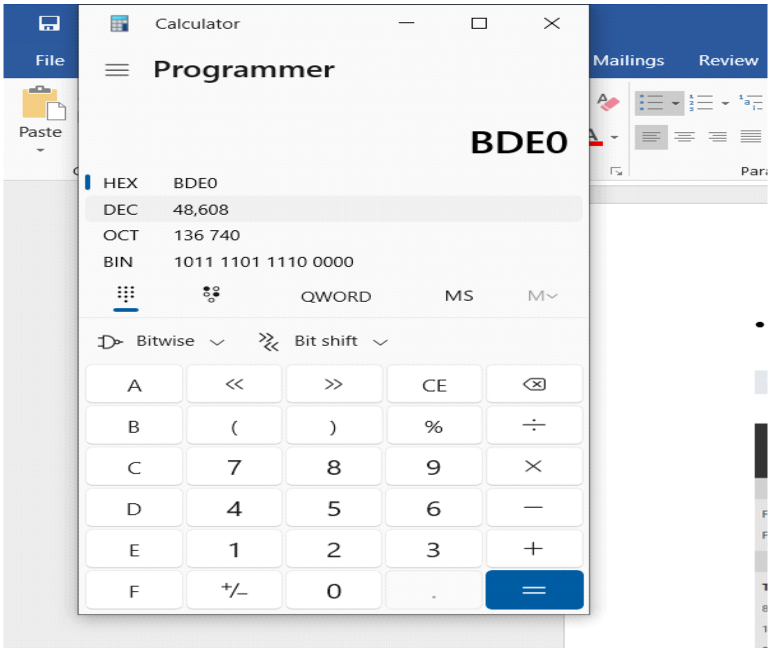
File Name	HeartBeat_V2.X.production.hex
File Size	351,836 bytes (344 KiB)

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (±)
8-bit Integer	255	-1
16-bit Integer	65535	-1
24-bit Integer	16777215	-1
32-bit Integer	4294967295	-1
64-bit Integer (+)	18446744073709551615	
64-bit Integer (±)	-1	
16-bit Float. P.	Invalid number	
32-bit Float. P.	Invalid number	
64-bit Float. P.	Invalid number	
LEB128 (+)	Invalid number	
LEB128 (±)	Invalid number	

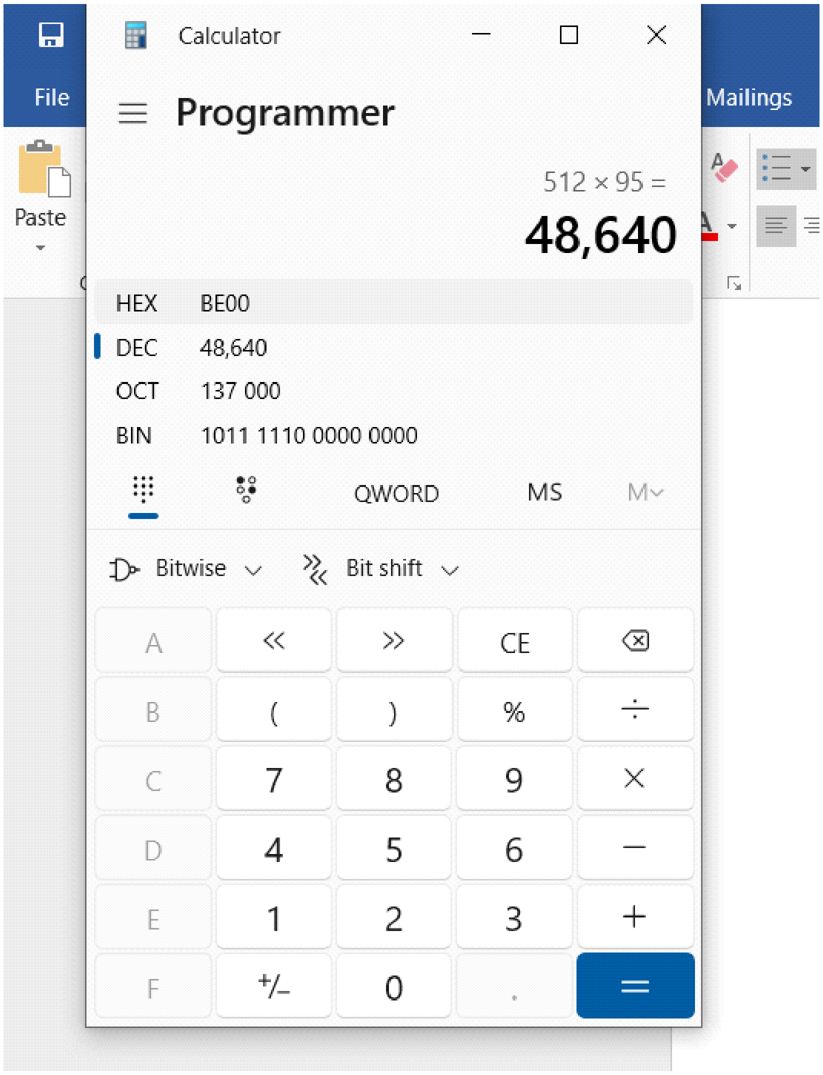
```
0000BD20  41 06 80 00 F0 F8 2F 00 00 80 60 00 00 40 A0 00  A.Ç.≡
0000BD30  40 06 88 00 71 05 80 00 80 FF 2F 00 00 80 60 00  @.ê.c
0000BD40  00 00 A0 00 70 05 88 00 61 05 80 00 F0 FF 28 00  ..á.p
0000BD50  00 80 60 00 00 C0 A0 00 60 05 88 00 00 80 FA 00  .Ç`..
0000BD60  00 00 06 00 02 00 FA 00 00 0F 78 00 9E 00 78 00  .....
0000BD70  91 41 88 00 FF FF 37 00 00 00 FA 00 40 AB 20 00  æAê.
0000BD80  80 07 78 00 C0 AC 20 00 00 01 88 00 00 80 FA 00  Ç.x.
0000BD90  00 00 06 00 00 00 FA 00 67 F8 07 00 D0 FF 07 00  .....
0000BDA0  AC FF 07 00 FB F5 07 00 F0 FE 07 00 10 C0 B3 00  ¼ ..
0000BDB0  8C FE 07 00 06 00 07 00 00 80 FA 00 00 00 06 00  î▪...
0000BDC0  00 00 FA 00 E4 F5 07 00 00 80 FA 00 00 00 06 00  ..·.Σ
0000BDD0  00 00 FA 00 88 00 A9 00 82 00 A9 00 98 00 A8 00  ..·.é
0000BDE0  00 80 FA 00 00 00 06 00 FF FF FF FF FF FF FF FF  .Ç·..
0000BDF0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE00  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE10  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE20  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE30  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE40  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE50  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000BE60  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

- As shown above the file data ends at address location near BDE0 . Address BDE0 is a hex value . Open a calculator and select programmer mode and type the above HEX address value .



```
BDE0
HEX  BDE0
DEC  48,608
OCT  136 740
BIN  1011 1101 1110 0000
```

- Decimal equivalent of BDE0 is 48,608. Now divide 48,608 by 512 check if it is multiple with remainder 0. The value 48,608 is not multiple of 512 with quotient 94 ,remainder 480.
- So next multiple of 512 is 95 . Find 512* 95 = 48,640 . the HEX equivalent of 48,640 is BE00.



Calculator

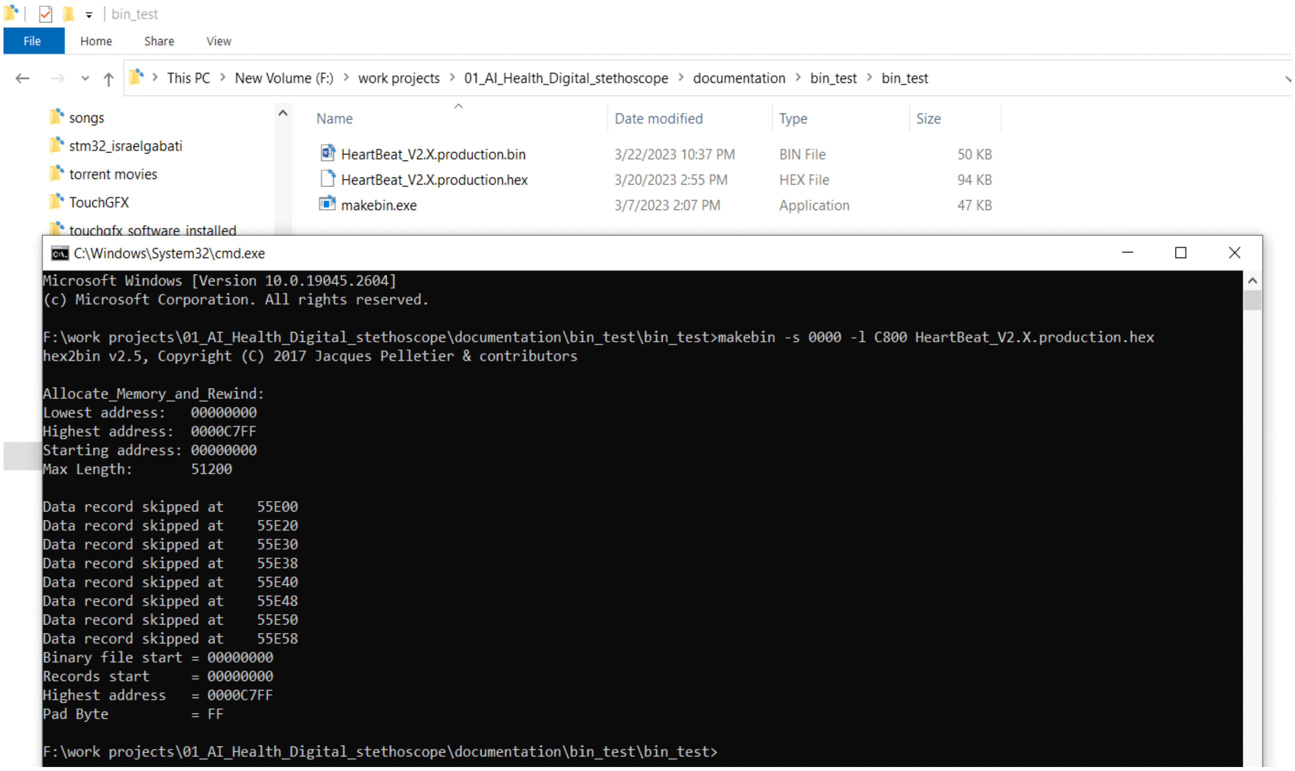
Programmer

```
512 × 95 =
48,640

HEX  BE00
DEC  48,640
OCT  137 000
BIN  1011 1110 0000 0000
```

- Again open Command prompt and try **len** value as **BE00.**

Example: makebin -s 0000 -l **BE00** HeartBeat_V2.X.production.hex

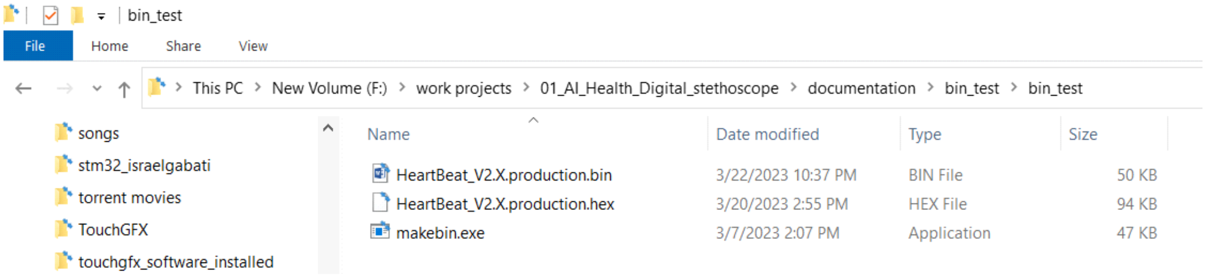
- Still we get data skip record at memory address before **55E00.**
- So continue increasing mutiples of 512 and **hex equivalent** as **len** in command line CMD and check for **which hex value DATA SKIP RECORD stop before 55E00**.
- We have got hex value **C800** which is a multiple of 512 and also stops data skip record before 55E00. So for this HEX FILE the **len value is C800**


bin_test
File
Home
Share
View
This PC > New Volume (F:) > work projects > 01_AI_Health_Digital_stethoscope > documentation > bin_test > bin_test
songs
stm32_israelgabati
torrent movies
TouchGFX
touchgfx_software_installed
C:\Windows\System32\cmd.exe

Name	Date modified	Type	Size
HeartBeat_V2.X.production.bin	3/22/2023 10:37 PM	BIN File	50 KB
HeartBeat_V2.X.production.hex	3/20/2023 2:55 PM	HEX File	94 KB
makebin.exe	3/7/2023 2:07 PM	Application	47 KB

```
Microsoft Windows [Version 10.0.19045.2604]
(c) Microsoft Corporation. All rights reserved.

F:\work projects\01_AI_Health_Digital_stethoscope\documentation\bin_test\bin_test>makebin -s 0000 -l C800 HeartBeat_V2.X.production.hex
hex2bin v2.5, Copyright (C) 2017 Jacques Pelletier & contributors

Allocate_Memory_and_Rewind:
Lowest address:   00000000
Highest address:  0000C7FF
Starting address: 00000000
Max Length:       51200

Data record skipped at    55E00
Data record skipped at    55E20
Data record skipped at    55E30
Data record skipped at    55E38
Data record skipped at    55E40
Data record skipped at    55E48
Data record skipped at    55E50
Data record skipped at    55E58
Binary file start = 00000000
Records start     = 00000000
Highest address   = 0000C7FF
Pad Byte          = FF

F:\work projects\01_AI_Health_Digital_stethoscope\documentation\bin_test\bin_test>
```

- As shown above data skip record stops and **bin file** with same name is created in that folder.


bin_test
File
Home
Share
View
This PC > New Volume (F:) > work projects > 01_AI_Health_Digital_stethoscope > documentation > bin_test > bin_test
songs
stm32_israelgabati
torrent movies
TouchGFX
touchgfx_software_installed

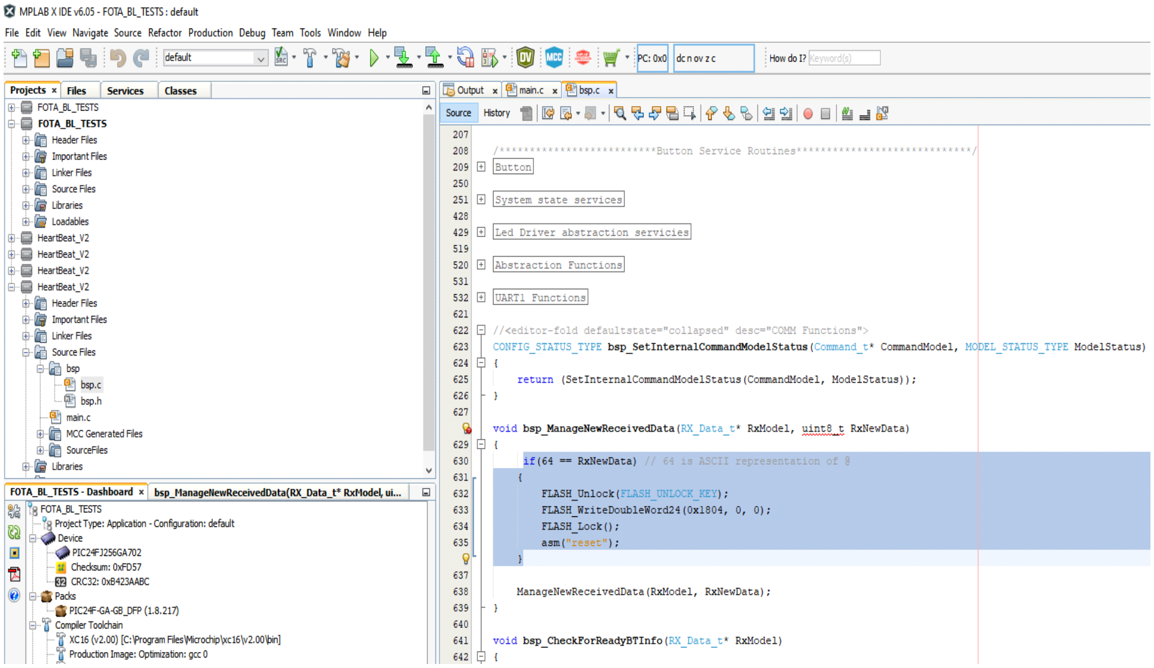
Name	Date modified	Type	Size
HeartBeat_V2.X.production.bin	3/22/2023 10:37 PM	BIN File	50 KB
HeartBeat_V2.X.production.hex	3/20/2023 2:55 PM	HEX File	94 KB
makebin.exe	3/7/2023 2:07 PM	Application	47 KB

- The above **bin file** is passed to web server page and transferred through OTA process to PIC24F device.

………

NOTES:

- For updating user application firmware code through Phone app OTA , we use a **reset function** in our code , which will respond to **RESET API** from phone and trigger a software reset of MCU . No modifications needed further in reset function.

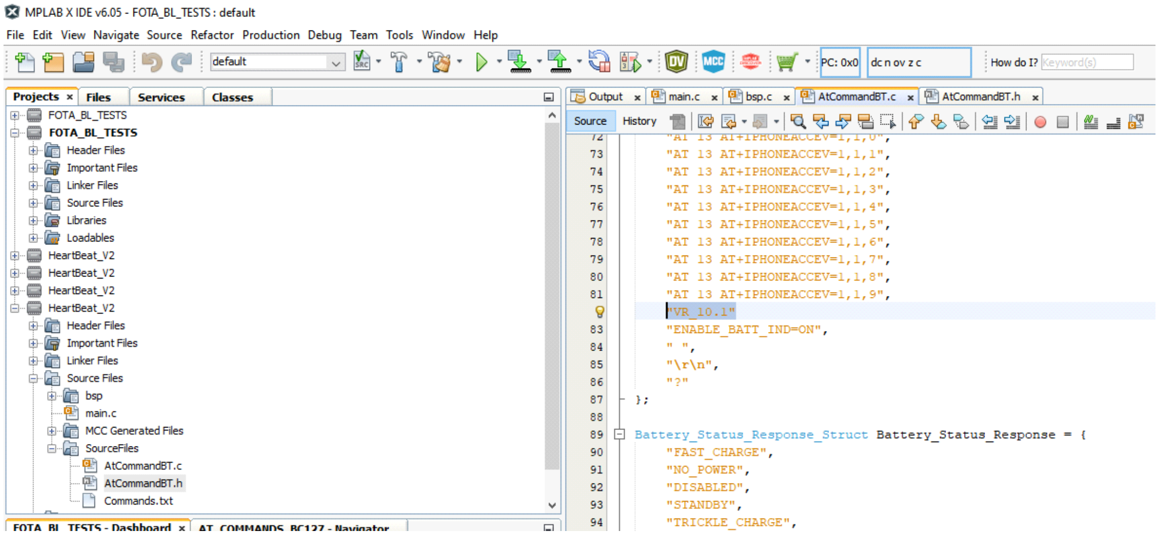


```
FOTA_BL_TESTS
    Project Type: Application - Configuration: default
    Device
        PIC24FJ256GA702
        Checksum: 0xFD57
        CRC32: 0xB423AABC
    Packs
        PIC24F-GA-GB_DFP (1.8.217)
    Compiler Toolchain
        XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]
        Production Image: Optimization: gcc 0
```

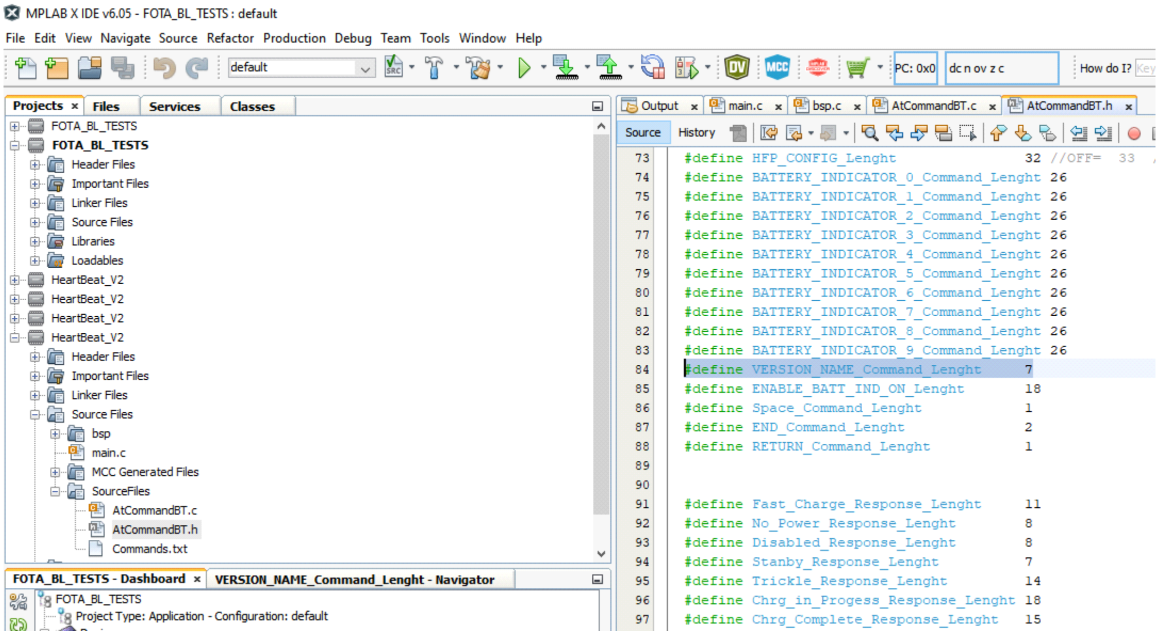
```
/*************************Button Service Routines****************************/
Button

System state services

Led Driver abstraction servicies

Abstraction Functions

UART1 Functions

//<editor-fold defaultstate="collapsed" desc="COMM Functions">
CONFIG_STATUS_TYPE bsp_SetInternalCommandModelStatus(Command_t* CommandModel, MODEL_STATUS_TYPE ModelStatus)
{
    return (SetInternalCommandModelStatus(CommandModel, ModelStatus));
}

void bsp_ManageNewReceivedData(RX_Data_t* RxModel, uint8_t RxNewData)
{
    if(64 == RxNewData) // 64 is ASCII representation of @
    {
        FLASH_Unlock(FLASH_UNLOCK_KEY);
        FLASH_WriteDoubleWord24(0x1804, 0, 0);
        FLASH_Lock();
        asm("reset");
    }

    ManageNewReceivedData(RxModel, RxNewData);
}

void bsp_CheckForReadyBTInfo(RX_Data_t* RxModel)
{
```

- we have made changes to **configuration settings** to user application firmware code by adding **flash files settings** to access flash memory . so no modifications needed further in configuration settings. Have to use this user application firmware code as **BASE file** for any further future update .

- For finding the **version number** for user application firmware and **version comparison** in phone app, we are **hardcoding** the **version number** in user code . we need to make **2 changes in code** to change the version number **before compiling hex file**. In **AtCommandsBT.c** file , **near line number 82** we use a command named **"VR_10.1" .** Here 10.1 is the version number , we can make changes to version number in format like "VR_01.1" , "VR_02.3" , "VR_12.2", "VR_34.5" , "VR_99.6" ..... The command uses a **length of 7** , which should be defined in **AtCommandBT.h** file near line **number 84** as **#define VERSION_NAME_Command_Lenght 7 .** refer pics below.



```c
    "AT 13 AT+IPHONEACCEV=1,1,0",
    "AT 13 AT+IPHONEACCEV=1,1,1",
    "AT 13 AT+IPHONEACCEV=1,1,2",
    "AT 13 AT+IPHONEACCEV=1,1,3",
    "AT 13 AT+IPHONEACCEV=1,1,4",
    "AT 13 AT+IPHONEACCEV=1,1,5",
    "AT 13 AT+IPHONEACCEV=1,1,6",
    "AT 13 AT+IPHONEACCEV=1,1,7",
    "AT 13 AT+IPHONEACCEV=1,1,8",
    "AT 13 AT+IPHONEACCEV=1,1,9",
    "VR_10.1"
    "ENABLE_BATT_IND=ON",
    " ",
    "\r\n",
    "?"
};

Battery_Status_Response_Struct Battery_Status_Response = {
    "FAST_CHARGE",
    "NO_POWER",
    "DISABLED",
    "STANDBY",
    "TRICKLE_CHARGE",
```



```c
#define HFP_CONFIG_Lenght                   32 //OFF=  33
#define BATTERY_INDICATOR_0_Command_Lenght 26
#define BATTERY_INDICATOR_1_Command_Lenght 26
#define BATTERY_INDICATOR_2_Command_Lenght 26
#define BATTERY_INDICATOR_3_Command_Lenght 26
#define BATTERY_INDICATOR_4_Command_Lenght 26
#define BATTERY_INDICATOR_5_Command_Lenght 26
#define BATTERY_INDICATOR_6_Command_Lenght 26
#define BATTERY_INDICATOR_7_Command_Lenght 26
#define BATTERY_INDICATOR_8_Command_Lenght 26
#define BATTERY_INDICATOR_9_Command_Lenght 26
#define VERSION_NAME_Command_Lenght         7
#define ENABLE_BATT_IND_ON_Lenght           18
#define Space_Command_Lenght                1
#define END_Command_Lenght                  2
#define RETURN_Command_Lenght               1


#define Fast_Charge_Response_Lenght         11
#define No_Power_Response_Lenght            8
#define Disabled_Response_Lenght            8
#define Stanby_Response_Lenght              7
#define Trickle_Response_Lenght             14
#define Chrg_in_Progess_Response_Lenght 18
#define Chrg_Complete_Response_Lenght       15
```