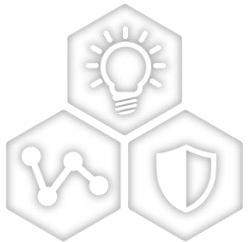


BTL101

8-bit Bootloader Using MCC Classic



A Leading Provider of Smart, Connected and Secure Embedded Control Solutions



SMART | CONNECTED | SECURE

Ken Wen
Dec 21, 2023

Class Objectives

At the end of this class you will be able to:

1. Understand Bootloader options and the benefits/drawbacks with each
2. Understand using MPLAB[®] X IDE Project Configurations to create three versions of the application
3. Integrate the bootloader with end application
4. Use the memory interface

Note:

This class specifically addresses bootloaders generated from the MCC* library for 8-bit PIC[®] devices only

* MPLAB Code Configurator (MCC)

Agenda

- Requirements
- Questions
- MPLAB[®] X IDE Project Configurations
- Precalculate Checksum
- Memory Interface
- Demonstration
- Hands-on Labs

Agenda

- **Requirements**
- Questions
- MPLAB® X IDE Project Configurations
- Precalculate Checksum
- Memory Interface
- Demonstration
- Hands-on Labs

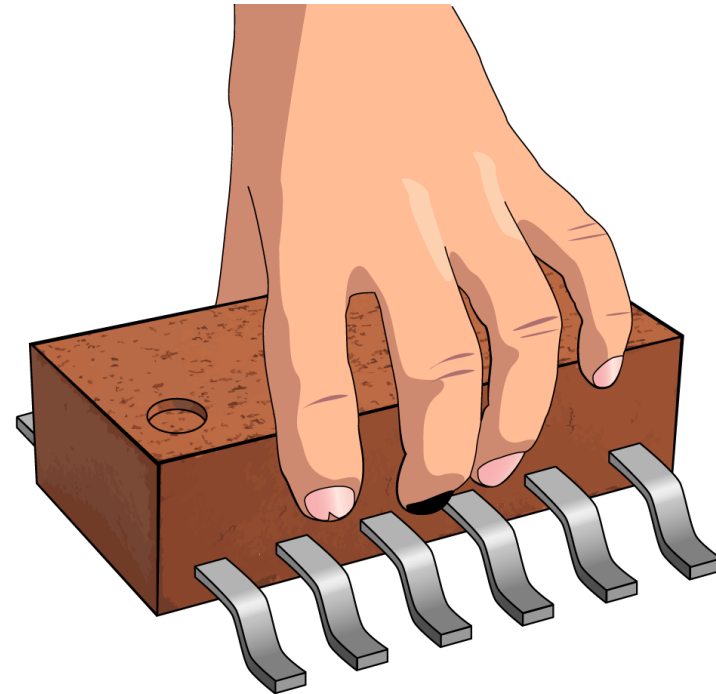
The Big Pieces

- End Application
- Boot loader
- Host Application



What is a bootloader?

- Code to reload its own application
- Runs first at startup
- **Prime Directive**
 - Validate Application
 - Protect itself
 - Rogue code
 - Bad Hex File



Agenda

- Requirements
- **Questions**
- MPLAB® X IDE Project Configurations
- Precalculate Checksum
- Memory Interface
- Demonstration
- Hands-on Labs

Questions

- **Run BL or App on startup**
- Application Request Bootload
- Where in memory
- Self Protection

Application Validation (startup)

- **Check App Reset Vector**
- **State Variable in NV Flash**
- **Checksum/CRC/hash on startup**
- **IO pin**

Check Reset Vector

- **Bootload Device except RV**
- **Validate the load**
- **Write the Reset Vector instruction**

Boot Load State

- **State Variable in NV flash**
 - 0xFF – No Application loaded
 - 0x55 – Application Valid
 - 0x00 – Boot loading

Checksum on Startup

- Calculate a checksum of application memory
- Compare to pre-calculated value stored in memory

Questions

- Run BL or App on startup
- **Application Request Bootload**
- Where in memory
- Self Protection

Host Application Requirements

- **Command End Application to switch to boot loader**
- **Load Hex File**
- **Send to boot loader**
- **Verify load**
- **Command BL to run End Application**

Switch to bootloader

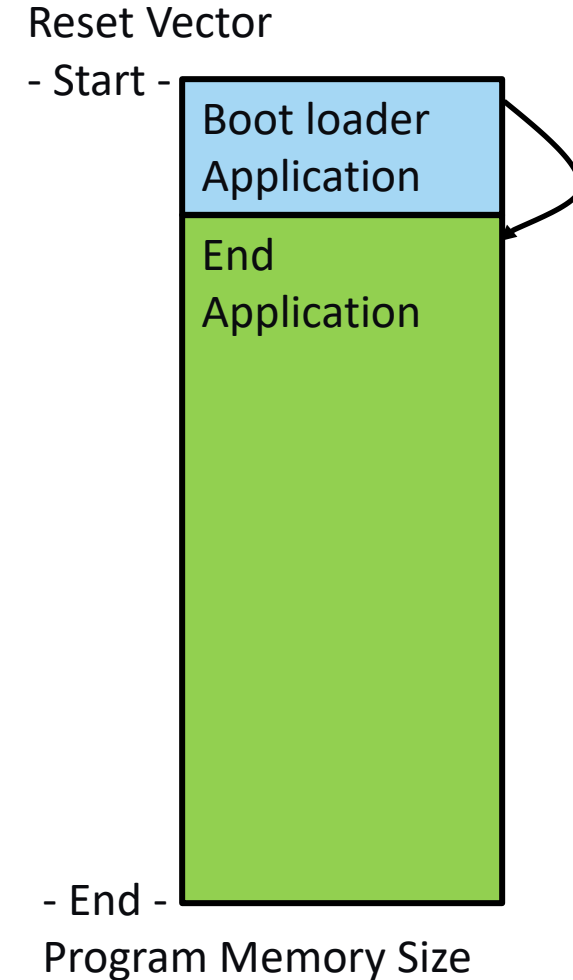
- Invalidate memory / Set “boot load me” flag
 - Reset Device
- or –
- Reset Stack / Jump to Reset Vector

Questions

- Run BL or App on startup
- Application Request Bootload
- **Where in memory**
- Self Protection

Where in Code Space

- **BL at Low Addresses**
 - Simplest
 - Occupies reset vector
 - Must reflect Interrupt vector(s)
 - Stand alone debug more complicated



Interrupts

- PIC16F & PIC18 Fixed location
- Need to reflect to application
- Do it Right!

```
asm("psect    intentry delta=2");  
asm("pagesel  " str(NEW_INTERRUPT_VECTOR));  
asm("GOTO    " str(NEW_INTERRUPT_VECTOR));
```

	Line	Address	Opcode	Label	DisAssy
⇒	1	0000	3180		MOVLW 0x0
	2	0001	2806		GOTO 0x6
	3	0002	3FFF		MOVWI [-1]FSR1
	4	0003	3FFF		MOVWI [-1]FSR1
	5	0004	3183		MOVLW 0x3
	6	0005	2B04		GOTO 0x304
	7	0006	3180		MOVLW 0x0

Questions

- Run BL or App on startup
- Application Request Bootload
- Where in memory
- **Self Protection**

Self Protection (PIC[®] MCU)

- **Write Protect**
 - None / ¼ / ½ / All
- **Address checks**
- **Dynamic Memory Keys**



PIC16(L)F1824/1828

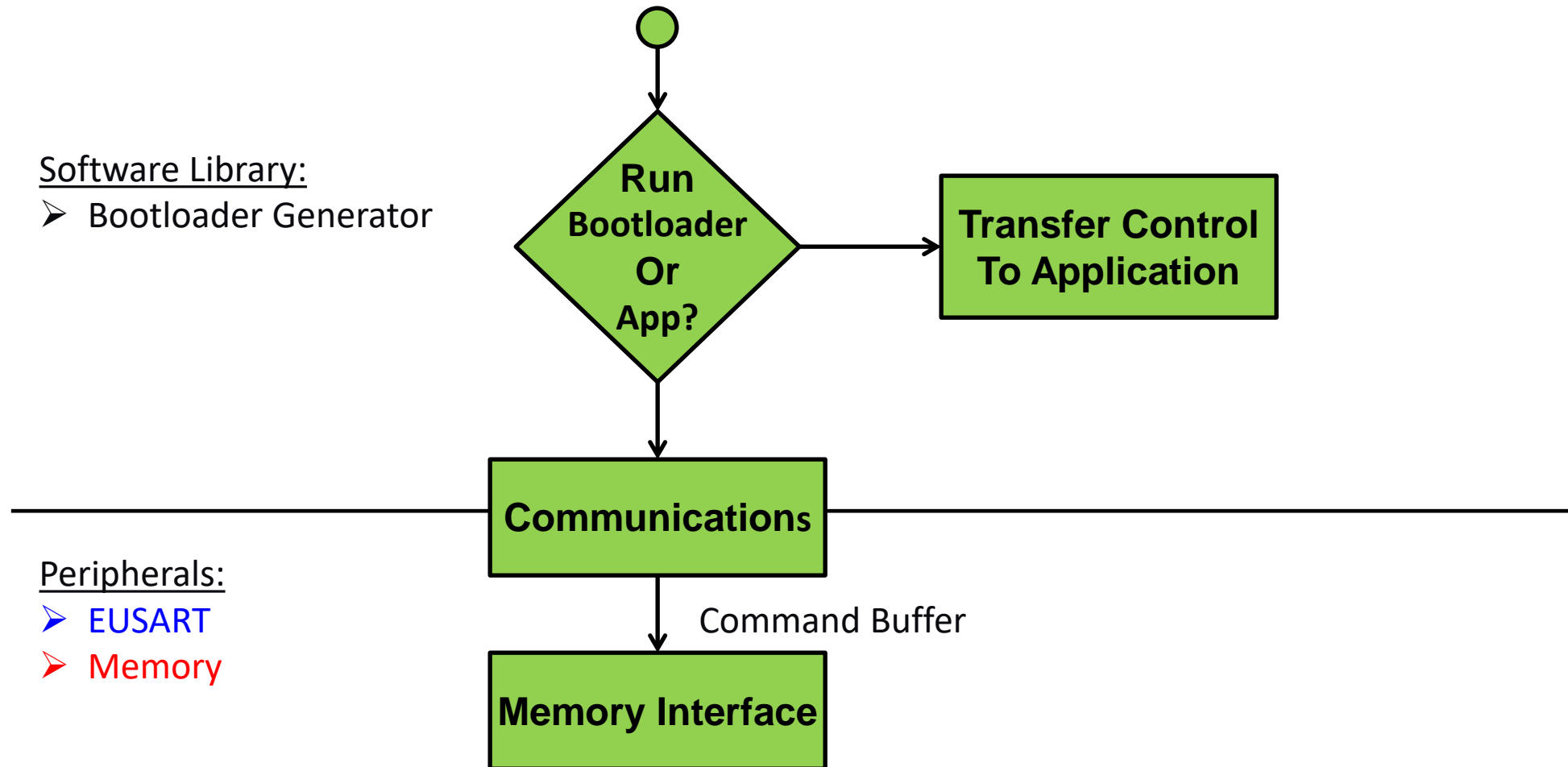
REGISTER 4-2: CONFIGURATION WORD 2

R/P-1/1	R/P-1/1	U-1	R/P-1/1	R/P-1/1	R/P-1/1		
LVP ⁽¹⁾	DEBUG ⁽²⁾	—	BORV	STVREN	PLLEN		
bit 13						bit 8	
U-1	U-1	U-1	R-1	U-1	U-1	R/P-1/1	R/P-1/1
—	—	—	Reserved	—	—	WRT1	WRT0
bit 7						bit 0	

Our Solution

Software Library:

- Bootloader Generator



Peripherals:

- EUSART
- Memory

MCC Bootloader Generator

MPLAB X IDE v6.15 - PIC16F18446_Bootloader : default

File Edit View Navigate Source Refactor Production Debug Team Tools Window Help

default

PC: 0x0 z dc c : W:0x0 : bank 0 How d

Proj... Files Serv... Clas... x

MCC v5.3.7

Project Re... Gen... Im... E... ?

Libraries

Bootloader Generator

Peripherals

EUSART1

MEMORY

System

Interrupt Module

Pin Module

System Module

Device Resources

Content Manager

DSM

Ext_Interrupt

FVR

MSSP

NCO

PWM

SMT

Timer

ZCD

Pin Module x Interrupt Module x System Module x Bootloader Generator x MEMORY x EUSART1 x

Easy Setup

User Defined Bootload Options

Transport Type: UART EUSART1

Verification: Check_Reset_Vect...

Application Reset Vector: 0x400

Application Interrupt Vector: 0x404

IO Pin Indicator: Disabled

I/O Pin Entry: Disabled

Software Protection: Disabled

Flash Read: Disabled

EEData Read: Disabled

EEData Write: Disabled

MCC Bootloader Generator

▼ User Defined Bootload Options

Transport Type:

UART ▼ Sources ▼

Verification:

Check_Reset_Vector ▼

Application Reset Vector:

Check_Reset_Vector

Check_State_Flag

Checksum

Application Interrupt Vector:

IO Pin Indicator:

Enabled ▼

I/O Pin Entry:

Enabled ▼

Software Protection:

Enabled

Flash Read:

Disabled ▼

EEData Read:

Disabled ▼

EEData Write:

Disabled ▼

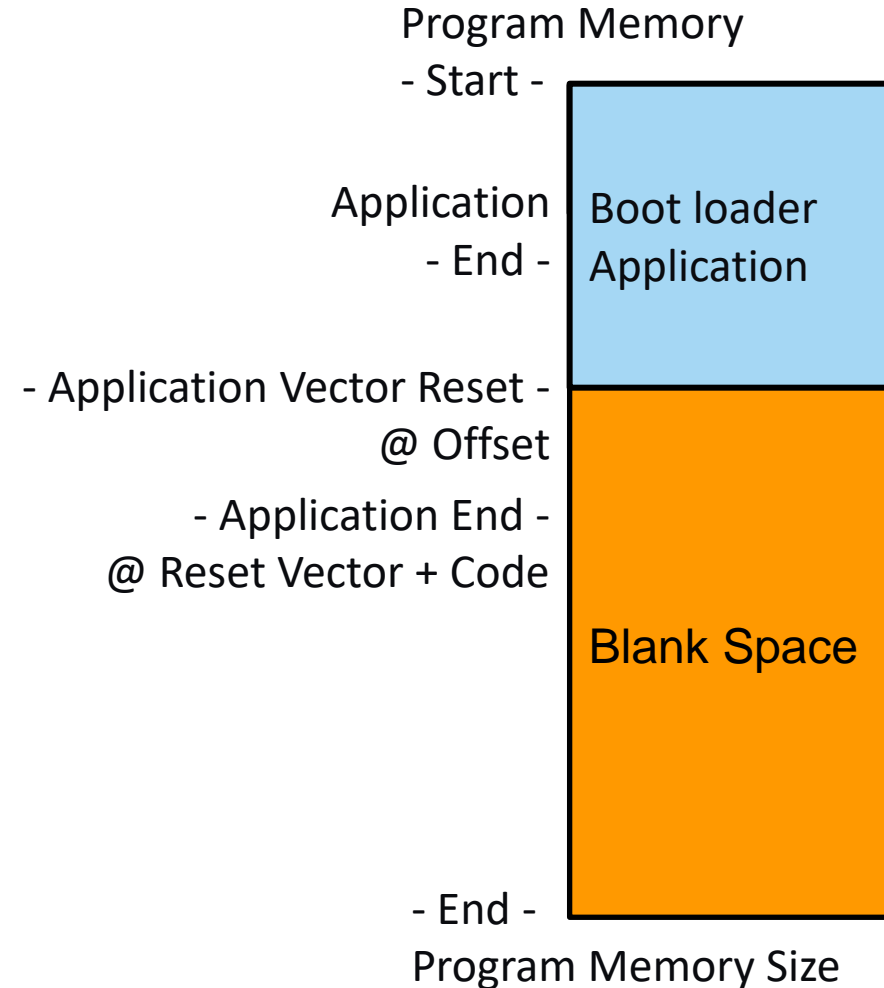
			Port A ▼								Port B ▼								Port C ▼							
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Bootloader Gener...	BL_ENTRY	input	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	
	BL_INDICATOR	output	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	
EUSART1 ▼	RX1	input																							🔒	
	TX1	output																							🔒	
OSC	CLKO	output							🔒																	
Pin Module ▼	GPIO	input	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	
	GPIO	output	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	
RESET	MCLR	input																								

Agenda

- Requirements
- Questions
- **MPLAB[®] X IDE Project Configurations**
- Precalculate Checksum
- Memory Interface
- Demonstration
- Hands-on Labs

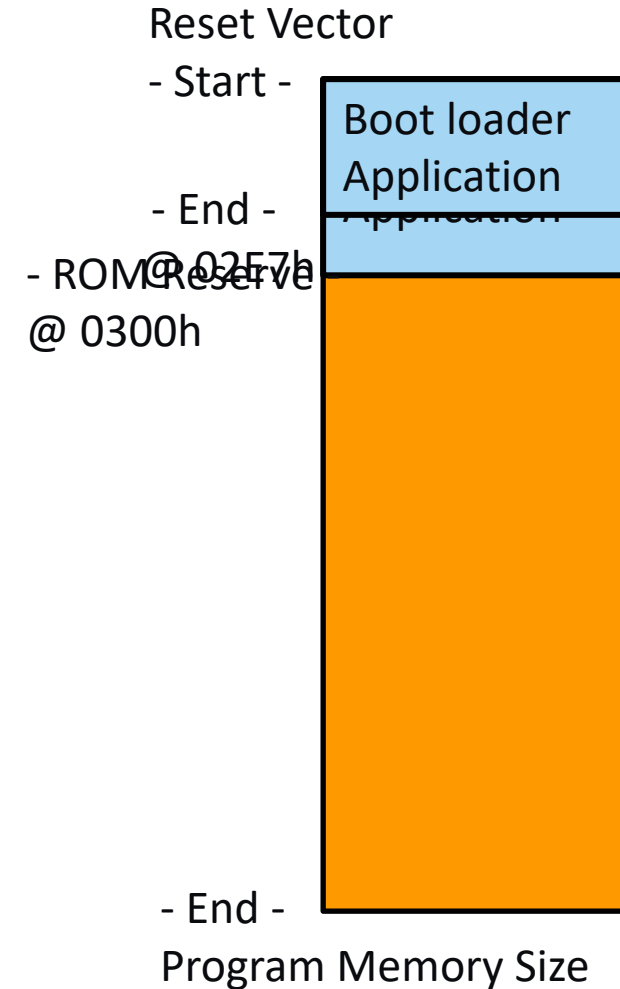
Application Configurations

- **Stand Alone**
 - Default
- **Offset App**
 - Hex file offset to leave room for BL
- **Combined**
 - Application and Bootloader

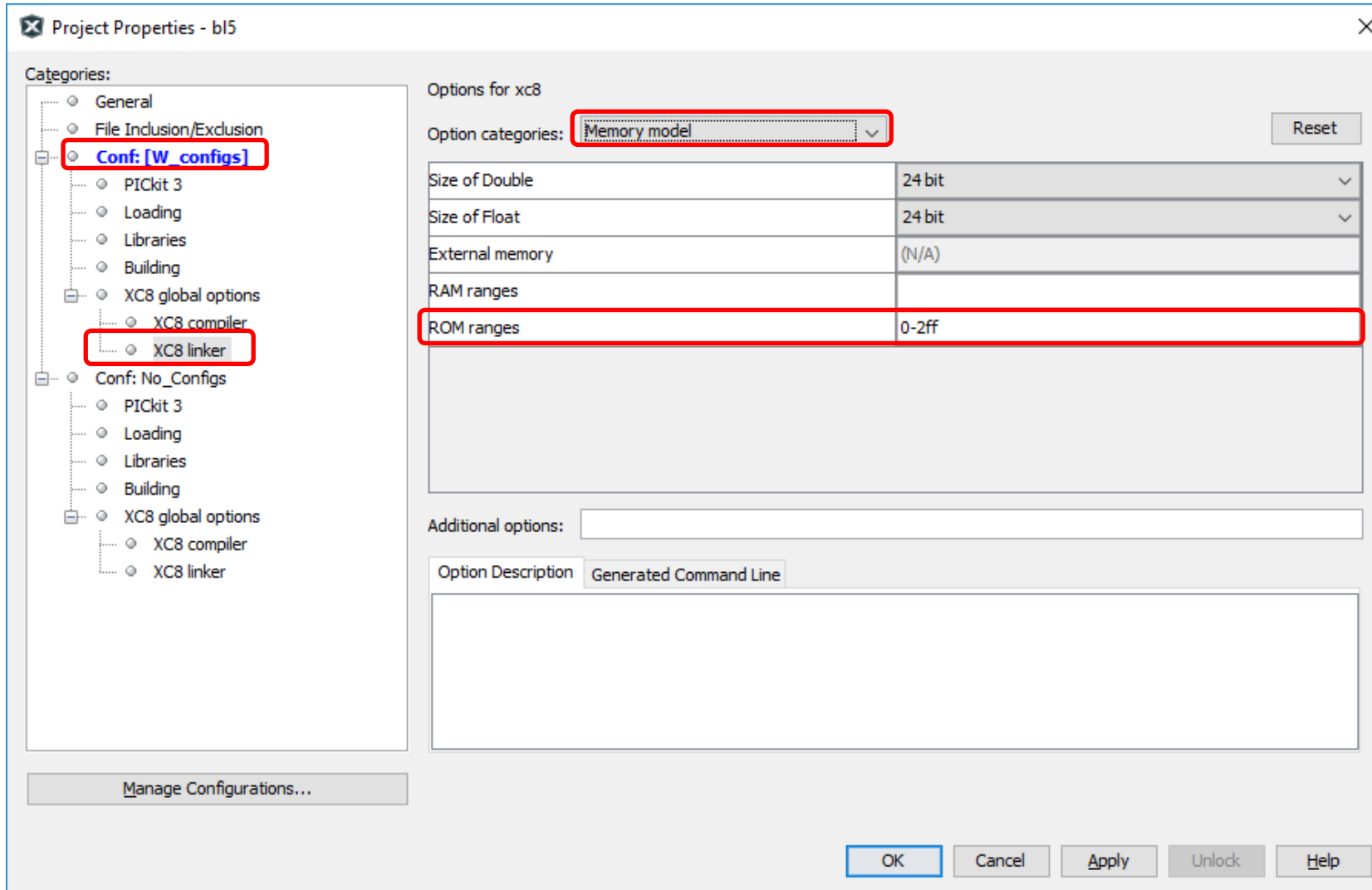


Boundaries

- **Bootloader**
 - Device Latch Size
 - Page Boundaries
- **0x20** – Latch Size
 - 0x02E0 – Not Enough Space
 - 0x0300 – Excess Space
- **0x02E7 → 0x0300**

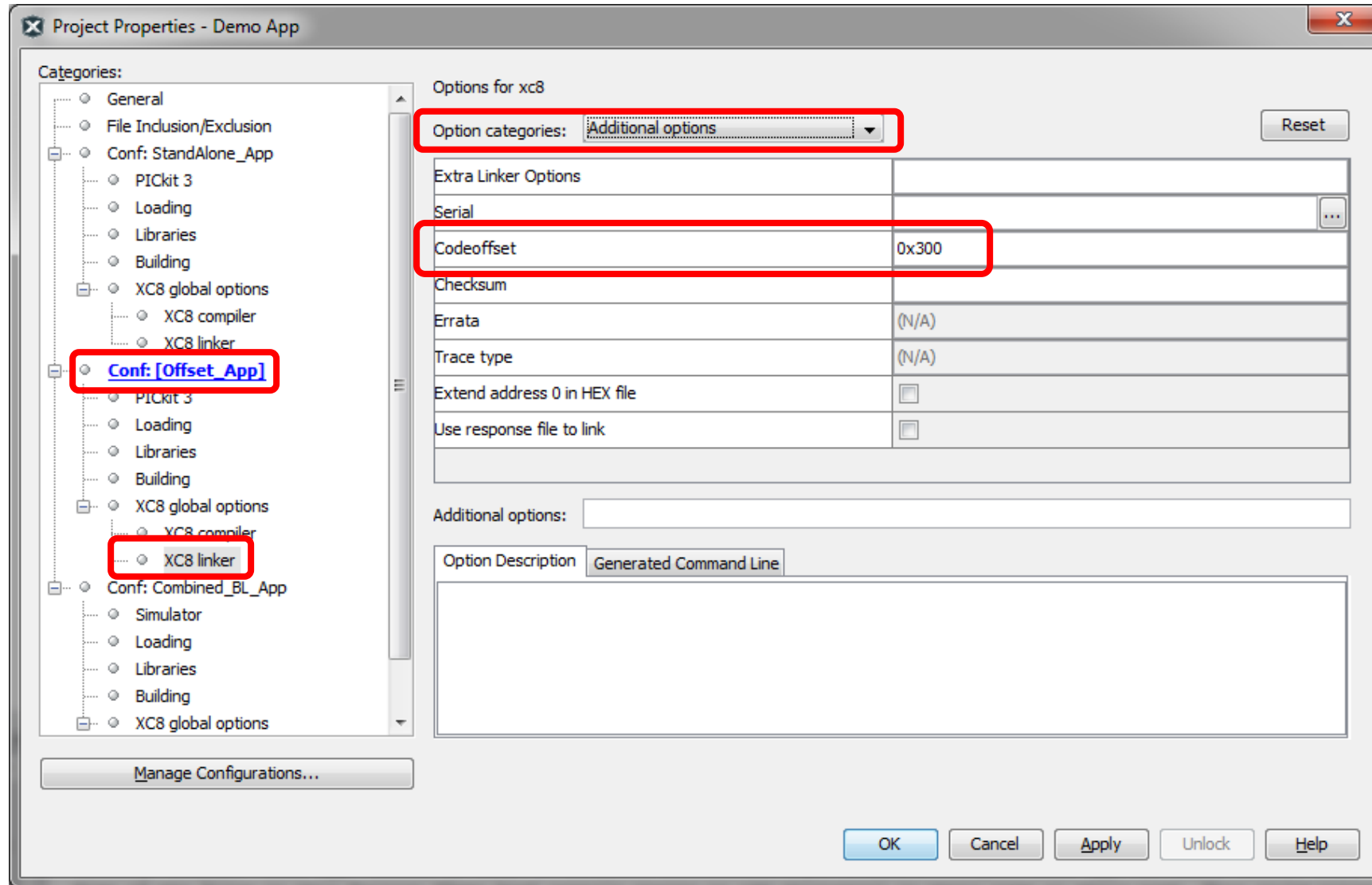


Manage Memory Usage (Bootloader for PIC)

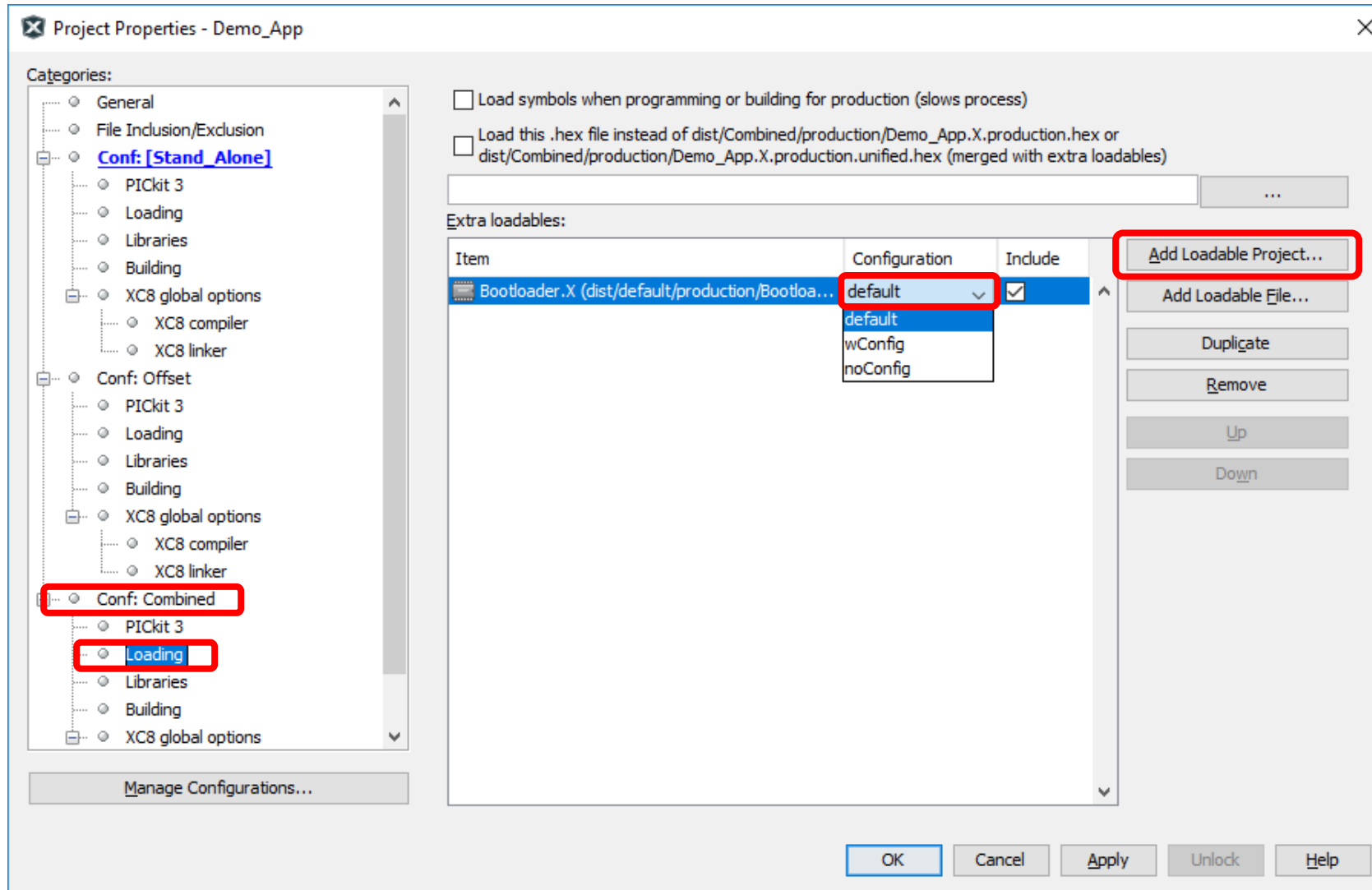


Code Offset

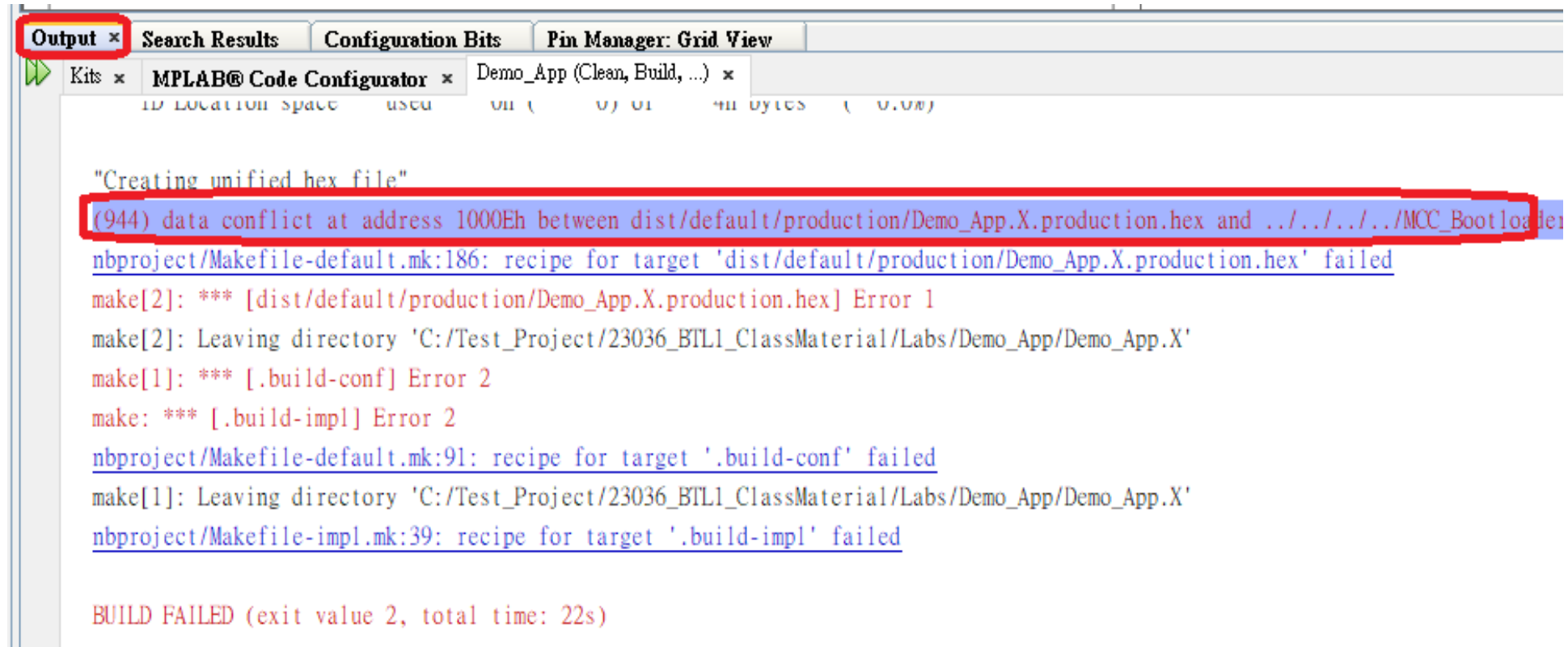
(Application for PIC)



Attach Bootloader



The “Config Words Problem”



The screenshot shows the MPLAB IDE interface with the 'Output' window active. The 'Output' tab is highlighted with a red box. The output text shows a build process for 'Demo_App' that fails due to a data conflict at address 1000Eh. The error message is highlighted with a red box: "(944) data conflict at address 1000Eh between dist/default/production/Demo_App.X.production.hex and ../../../../MCC_Bootloader/nbproject/Makefile-default.mk:186: recipe for target 'dist/default/production/Demo_App.X.production.hex' failed". Below this, the build process continues to fail at other targets, and the final status is "BUILD FAILED (exit value 2, total time: 22s)".

```
"Creating unified hex file"
(944) data conflict at address 1000Eh between dist/default/production/Demo_App.X.production.hex and ../../../../MCC_Bootloader/nbproject/Makefile-default.mk:186: recipe for target 'dist/default/production/Demo_App.X.production.hex' failed
make[2]: *** [dist/default/production/Demo_App.X.production.hex] Error 1
make[2]: Leaving directory 'C:/Test_Project/23036_BTLL_ClassMaterial/Labs/Demo_App/Demo_App.X'
make[1]: *** [.build-conf] Error 2
make: *** [.build-impl] Error 2
nbproject/Makefile-default.mk:91: recipe for target '.build-conf' failed
make[1]: Leaving directory 'C:/Test_Project/23036_BTLL_ClassMaterial/Labs/Demo_App/Demo_App.X'
nbproject/Makefile-impl.mk:39: recipe for target '.build-impl' failed

BUILD FAILED (exit value 2, total time: 22s)
```

Configs

Configs

“(944) data conflict at address 1000Eh”

Use the same Config Words in APP & Bootloader

```
49 L // CONFIG1
50 #pragma config FEXTOSC = OFF // External Oscillator
51 #pragma config RSTOSC = LFINT // Power-up default
52 #pragma config CLKOUTEN = OFF // Clock Out Enable bit
53 #pragma config CSWEN = ON // Clock Switch Enable bit
54 #pragma config FCMEN = OFF // Fail-Safe Clock Monitor
```

Application

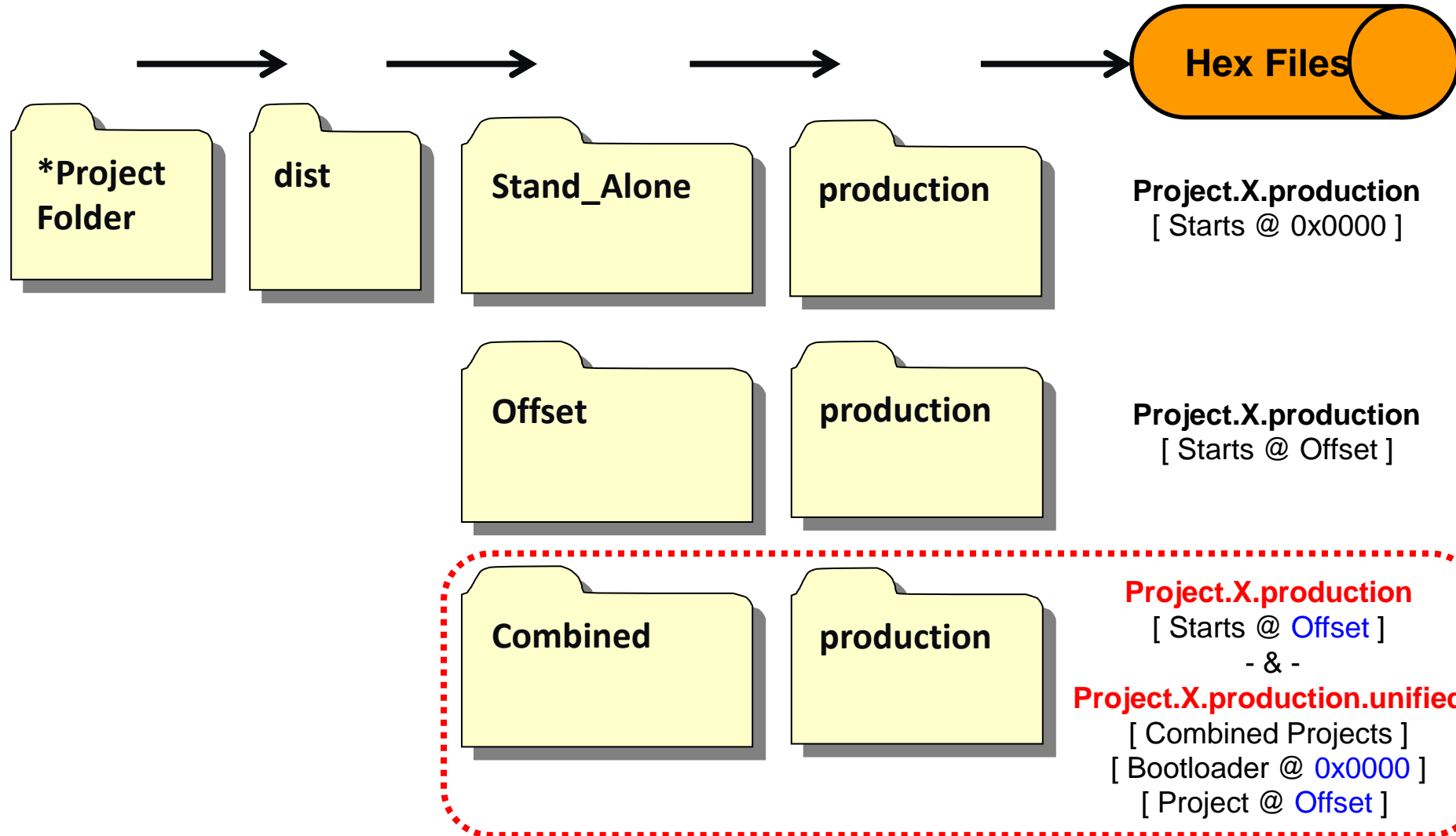
```
55 // CONFIG2
56 #pragma config MCLRE = ON // Master Clear Enable bit
57 #pragma config PWRTS = OFF // Power-up Timer Enable bit
58 #pragma config LPBORN = OFF // Low-Power BOR enable bit
59 #pragma config BOREN = OFF // Brown-out reset enable bit
60 #pragma config BORV = LO // Brown-out Reset Voltage Select
61 #pragma config ZCDDIS = OFF // Zero-cross detect disable
62 #pragma config PPS1WAY = OFF // Peripheral Pin Select one-time use
63 #pragma config STVREN = ON // Stack Overflow/Underflow reset
```

```
49 L // CONFIG1
50 #pragma config FEXTOSC = OFF // External Oscillator mode
51 #pragma config RSTOSC = HFINT1 // Power-up default value
52 #pragma config CLKOUTEN = OFF // Clock Out Enable bit
53 #pragma config CSWEN = ON // Clock Switch Enable bit
54 #pragma config FCMEN = ON // Fail-Safe Clock Monitor
```

Bootloader

```
55 // CONFIG2
56 #pragma config MCLRE = ON // Master Clear Enable bit
57 #pragma config PWRTS = OFF // Power-up Timer Enable bit
58 #pragma config LPBORN = OFF // Low-Power BOR enable bit
59 #pragma config BOREN = ON // Brown-out reset enable bit
60 #pragma config BORV = LO // Brown-out Reset Voltage Select
61 #pragma config ZCDDIS = OFF // Zero-cross detect disable
62 #pragma config PPS1WAY = ON // Peripheral Pin Select one-time use
63 #pragma config STVREN = ON // Stack Overflow/Underflow reset
```

Project Configurations



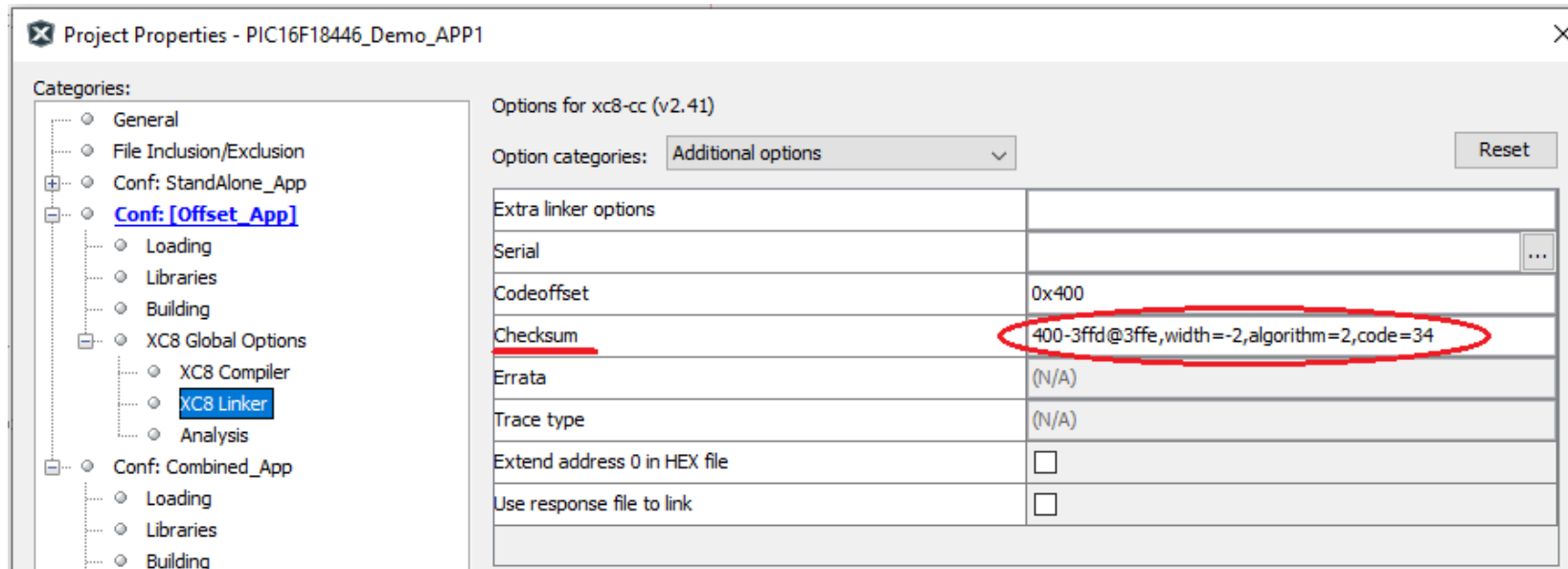
Agenda

- Requirements
- Questions
- MPLAB® X IDE Project Configurations
- **Precalculate Checksum**
- Memory Interface
- Demonstration
- Hands-on Labs

Precalculated Checksum (PIC MCU)

- Precalculated Checksum required for validation method
- Hexmate*
- Fill Memory
- Generate Checksum

`0300-1ffd@1ffe,width=-2,code=0x34,algorithm=2`



Agenda

- Requirements
- Questions
- MPLAB® X IDE Project Configurations
- Precalculate Checksum
- **Memory Interface**
- Demonstration
- Hands-on Labs

Memory Panels

- **Writes**
 - 1s to 0s
 - Write latches
 - Mind the row boundaries
- **Erase**
 - 0s to 1s
 - Row at a time
- **~2-4mS cycle**

Memory Original:
110
Write Latches: 011
Memory Result:
010

Start Write Sequence

- **Required sequence:**

```
EECON2 = 0x55;
```

```
EECON2 = 0xAA;
```

```
EECON1bits.WR = 1;
```

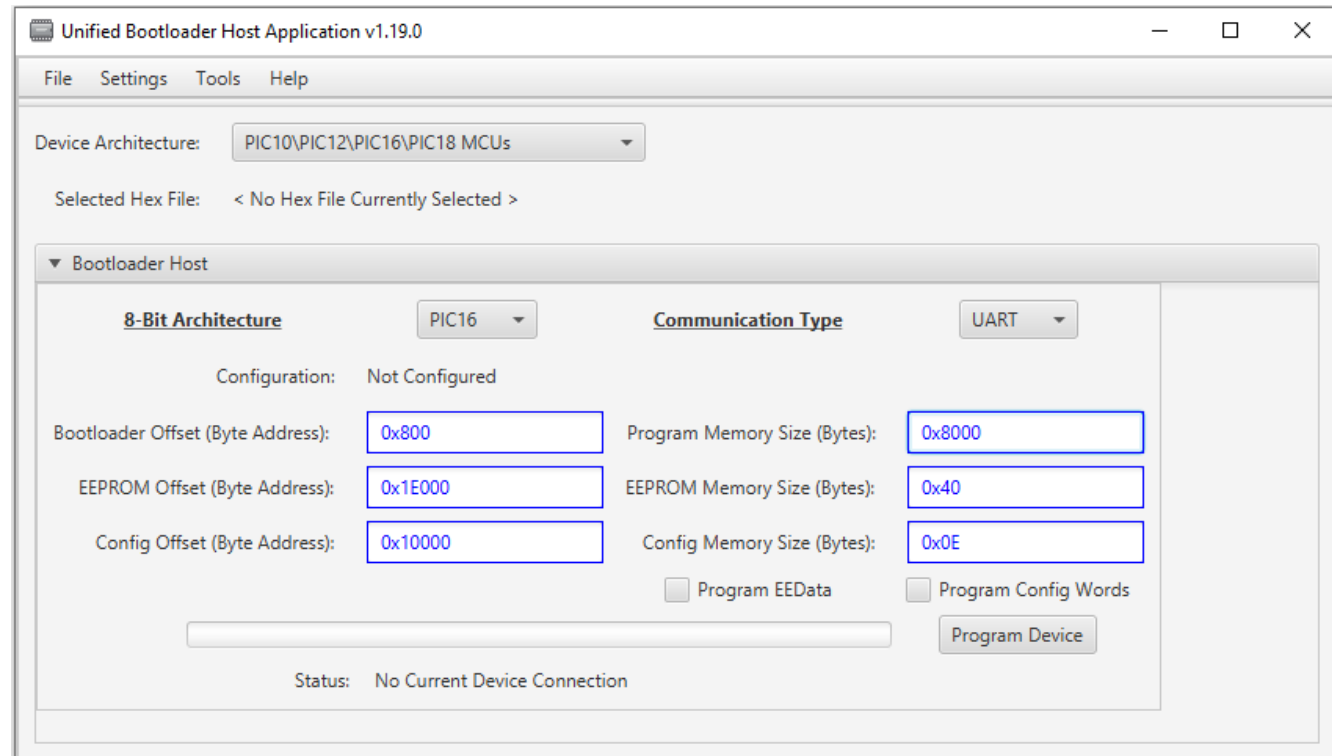
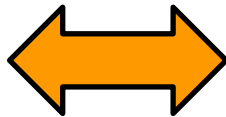
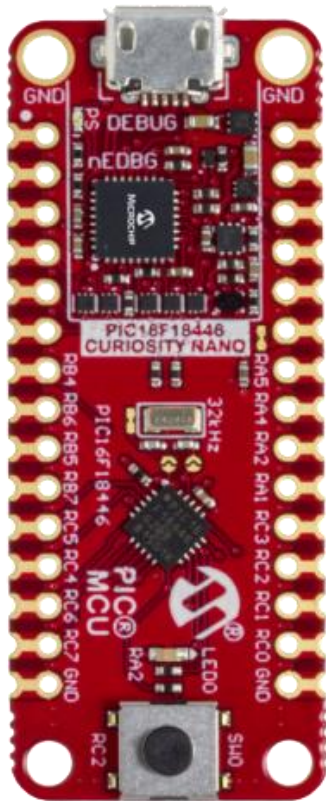
- **May have to use embedded assembly**

Agenda

- Requirements
- Questions
- MPLAB® X IDE Project Configurations
- Precalculate Checksum
- Memory Interface
- **Demonstration**
- Hands-on Labs


Demonstration




MCC Bootloader Generator with Unified Bootloader Host Application



FAQ : Unified Bootloader Host App Crash Issue

- Suggest to download 32-bit version **JRE(v8.0.201)**

Windows  Which download should I choose?

	Windows Online filesize: 2.16 MB	Instructions	After installing Java, you may need to restart your browser in order to enable Java in your browser.
	Windows Offline filesize: 56.43 MB	Instructions	
	Windows Offline (64-bit) filesize: 62.11 MB	Instructions	

If you use 32-bit and 64-bit browsers interchangeably, you will need to install both 32-bit and 64-bit Java in order to have the Java plug-in for both browsers. » [FAQ about 64-bit Java for Windows](#)

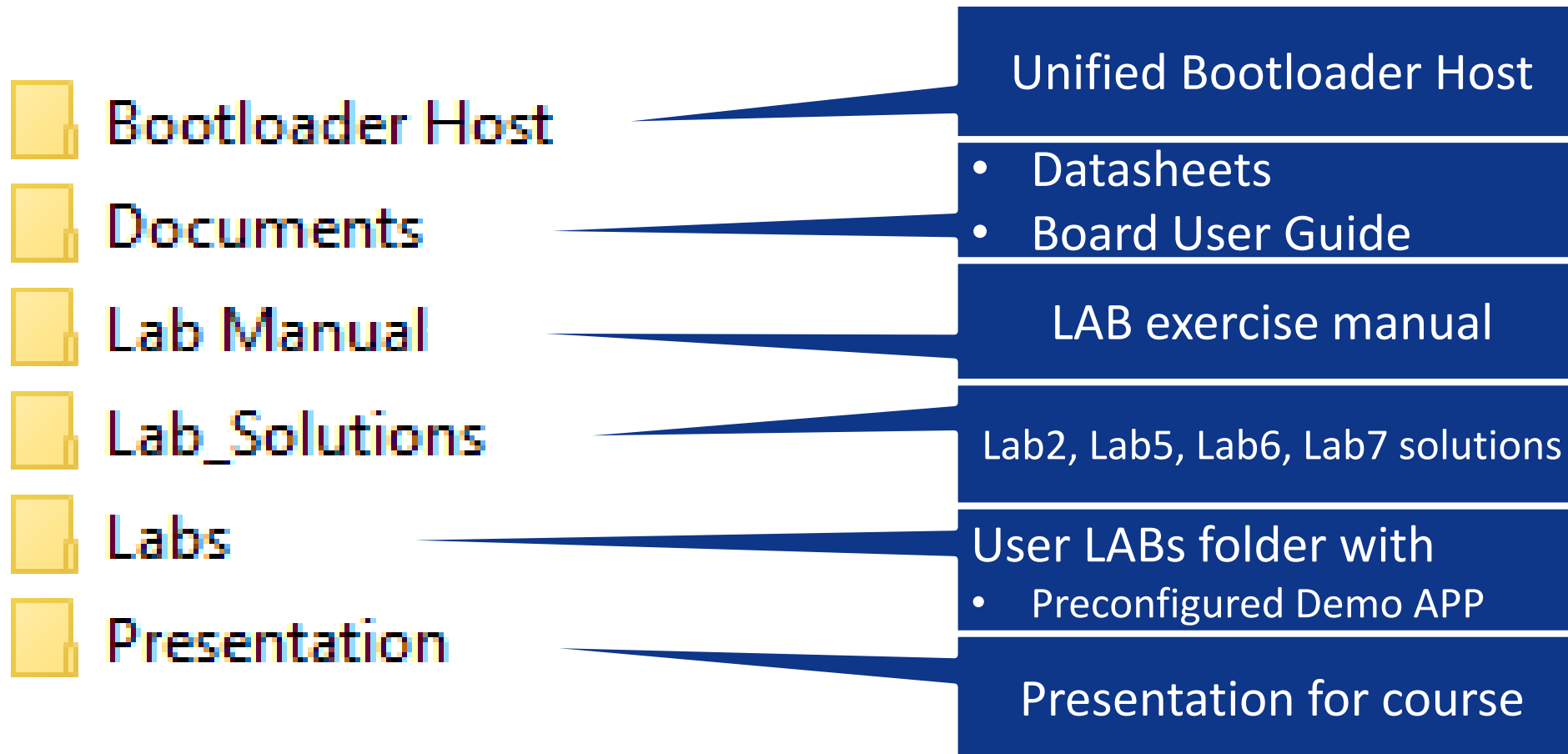
<https://www.java.com/download/manual.jsp>

Agenda

- Requirements
- Questions
- MPLAB® X IDE Project Configurations
- Precalculate Checksum
- Memory Interface
- Demonstration
- **Hands-on Labs**

Hands-on Labs

Folder Structure



LAB Manual Content

- Lab Equipment..... 3
 - Required Hardware..... 3
 - Required Software 3
- Overview 4
- Lab 1: Build and Run Demo Application 5
- Lab 2: Generate UART Bootloader..... 7
- Lab 3: Create Project Configurations 11
- Lab 4: Fix the Confiuration Word Problem 16
- Lab 5: Bootloader Offset Application 19
- Lab 6: Checksum Validation Method 21
- Lab 7: Deep-Dive to Self-Write Protection..... 23
- Reference..... 25



Labs

- **Lab 1 Build End Application**
- **Lab 2 Generate UART Bootloader**
- **Lab 3 Project Configurations**
- **Lab 4 Fix the Config Word Problem**
- **Lab 5 Bootloader Offset Application**
- **Lab 6 Checksum Validation Method**
- **Lab 7 Deep-Dive to Self-Write Protection**



LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by “open source” software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided “AS IS.” Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE. PLEASE BE AWARE THAT IMPLEMENTATION OF INTELLECTUAL PROPERTY PRESENTED HERE MAY REQUIRE A LICENSE FROM THIRD PARTIES.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2019, Microchip Technology Incorporated, All Rights Reserved.

Agenda(Host Side)

- **Scope of Responsibilities**
- Bootloader Commands
- Hex File Format
- Implementation Examples

Host Requirements

- **Load Hex File**
- **Store or Parse Hex Data**
- **Execute “Command Chain”**
- **Bootloader Commands Syntax to update End-Application**
- **Verify Code: Checksum / CRC**
- **Exit Bootloader into Application**

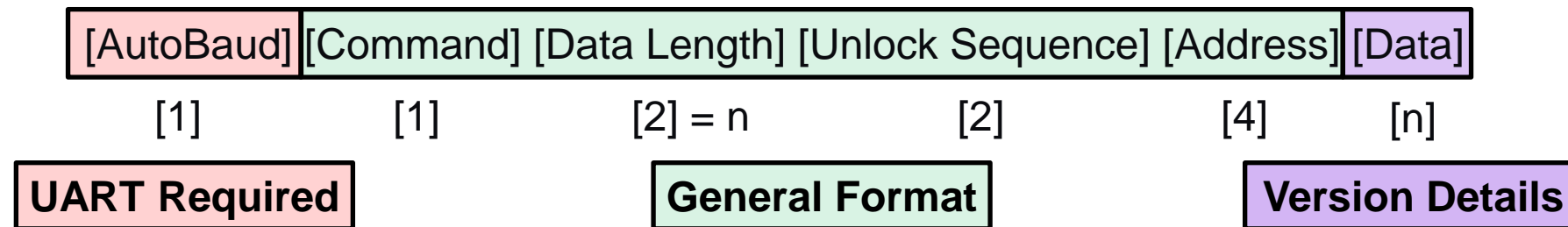
Agenda(Host Side)

- Scope of Responsibilities
- **Bootloader Commands**
- Hex File Format
- Implementation Examples

General Command Format

<Command><Length><Keys><Address><Data>

- Command – 1 byte
- Length – 2 bytes
- Memory Access Keys – 2 bytes
- Address – 4 bytes (Low, High, Upper, Extended)
- Data (Optional)



Recommended Minimal

0 – Get Version & More

1 – Read Flash Memory

2 – Write Flash Memory

3 – Erase Flash Memory

4 – Read EE Data

5 – Write EE Data

6 – Read Config Words

7 – Write Config Words  Write Status on AVR

8 – Calculate Checksum

9 – Reset Device

Expected Results

- **‘Echo’ of Command**
- **“Write” & “Erase” Status return:**
 - 0x01 – Command Successful
 - 0xFE – Address Error
 - 0xFF – Command Not Supported
- **“Read” commands append data**
 - No Status

Agenda(Host Side)

- Scope of Responsibilities
- Bootloader Commands
- **Hex File Format**
- Implementation Examples

What's in a Hex file?

: 0204000000348A

- 02 – data byte count
- 0400 – starting address
- 00 – record type
- 0034 – data payload (0x3400)
- 8A – checksum

Record Types

- **00 – Data Record**
- **01 – End of File Record**
- **02 – Segment Address Record**
- **03 – Segment Start Address***
- **04 – Extended Linear Address Record**
- **05 – Start Extended Linear Address***

*(AVR, 32)

PIC16F Hex File

- **Word addressed**

- :100000002500 ...7300010CF5

- :100010000600 ... 010C3E00F4

- Word Address: Byte address * 2

- **EEData Encoded at:**

- PIC16F – 0x2100

- PIC16F1 – 0xF000

:10420000080009000A00...52 (Hex File)



Extra on PIC16 | Data on PIC18*

Offset Hex File

:060600000000083310E2B07

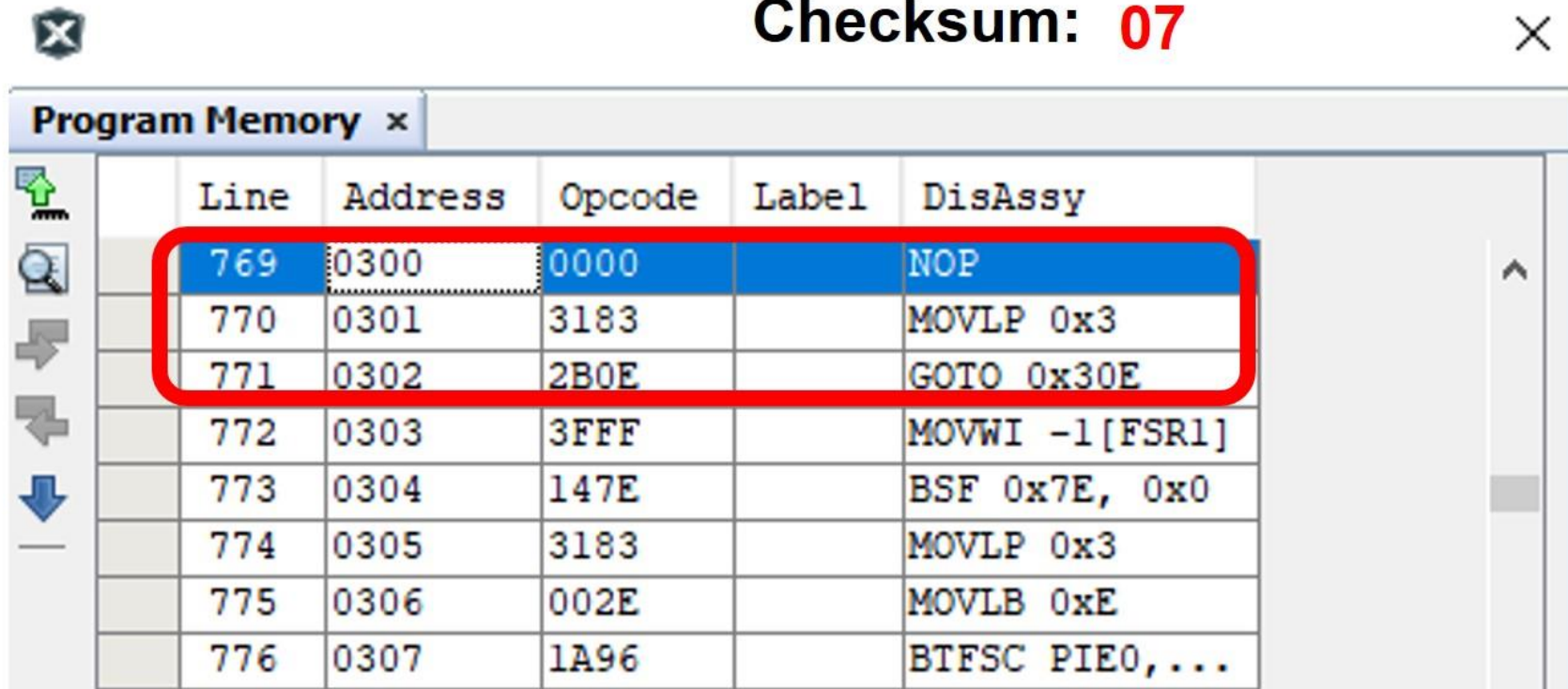
Bytes: 06

Address: 0600

Record Type: 00

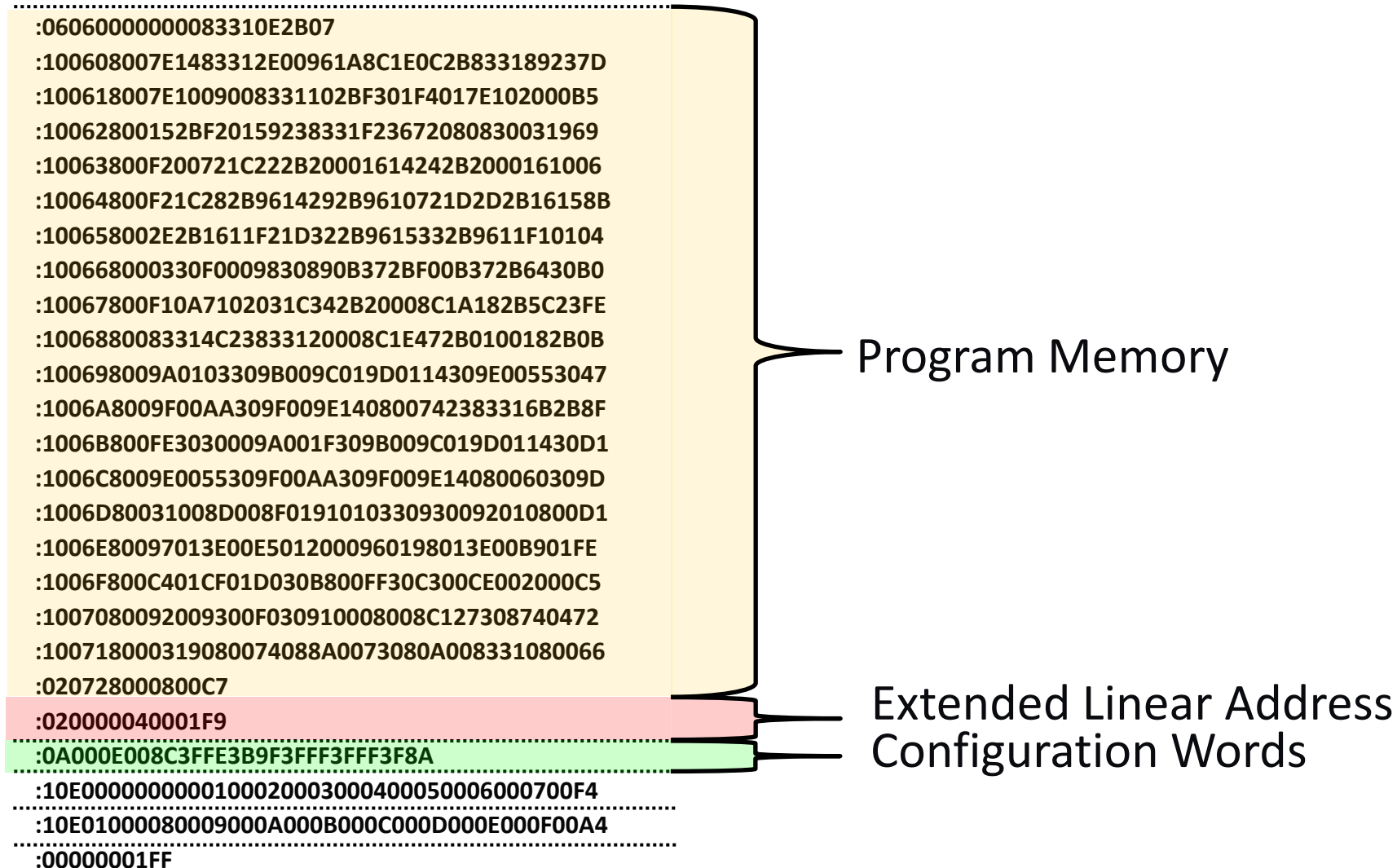
Data: 00 00 83 31 0E 2B

Checksum: 07



Line	Address	Opcode	Label	DisAssy
769	0300	0000		NOP
770	0301	3183		MOVLW 0x3
771	0302	2B0E		GOTO 0x30E
772	0303	3FFF		MOVW -1[FSR1]
773	0304	147E		BSF 0x7E, 0x0
774	0305	3183		MOVLW 0x3
775	0306	002E		MOVLB 0xE
776	0307	1A96		BTFSC PIE0, ...

Extended Address Space



Extended Linear Commands

:020000040001F9

Bytes: 02

Address: 00 00

Record Type: 04 : Extended Linear Address

Data: 00 01

Checksum: F9

:0A000E008C3FFE3B9F3FFF3FFF3F8A

Bytes: 10

Address: 00 0E : @ 0x01000E ($\frac{0x1000E}{2} = 0x8007$)

Record Type: 00

Data: 8C 3F FE 3B 9F 3F FF 3F FF 3F

Checksum: 8A

PIC: Configuration Words

:0A000E008C3FFE3B9F3FFF3FFF3F8A

Output - Demo_App (Clean, Build, ...)					Configuration Bits x
Address	Name	Value	Field	Option	
8007	CONFIG1	3F8C	FEXTOSC	OFF	
			RSTOSC	HFINT32	
			CLKOUTEN	OFF	
			CSWEN	ON	
			FCMEN	ON	
8008	CONFIG2	3BFE	MCLRE	OFF	
			PWRTE	OFF	
			LPBOREN	OFF	
			BOREN	ON	
			BORV	LO	
			ZCD	ON	
			PPS1WAY	ON	
			STVREN	ON	
8009	CONFIG3	3F9F	WDTCP	WDTCPS_31	
			WDTE	OFF	
			WDTCWS	WDTCWS_7	
			WDTCSS	SC	
800A	CONFIG4	3FFF	WRT	OFF	
			SCANE	available	
			LVP	ON	
800B	CONFIG5	3FFF	CP	OFF	
			CPD	OFF	

End of Hex File

```
.....:06060000000083310E2B07
:100608007E1483312E00961A8C1E0C2B833189237D
:100618007E1009008331102BF301F4017E102000B5
:10062800152BF20159238331F23672080830031969
:10063800F200721C222B20001614242B2000161006
:10064800F21C282B9614292B9610721D2D2B16158B
:100658002E2B1611F21D322B9615332B9611F10104
:100668000330F0009830890B372BF00B372B6430B0
:10067800F10A7102031C342B20008C1A182B5C23FE
:1006880083314C23833120008C1E472B0100182B0B
:100698009A0103309B009C019D0114309E00553047
:1006A8009F00AA309F009E140800742383316B2B8F
:1006B800FE3030009A001F309B009C019D011430D1
:1006C8009E0055309F00AA309F009E14080060309D
:1006D80031008D008F0191010330930092010800D1
:1006E80097013E00E5012000960198013E00B901FE
:1006F800C401CF01D030B800FF30C300CE002000C5
:1007080092009300F030910008008C127308740472
:100718000319080074088A0073080A008331080066
:020728000800C7
:020000040001F9
:0A000E008C3FFE3B9F3FFF3FFF3F8A
.....:10E000000000100020003000400050006000700F4
:10E01000080009000A000B000C000D000E000F00A4
:00000001FF
```

:00000001FF

End of File

Bytes: 00

Address: 00 00

Record Type: 01

Data: N/A

Checksum: FF

Program Memory

Extended Linear Address

Configuration Words

EEPROM Memory

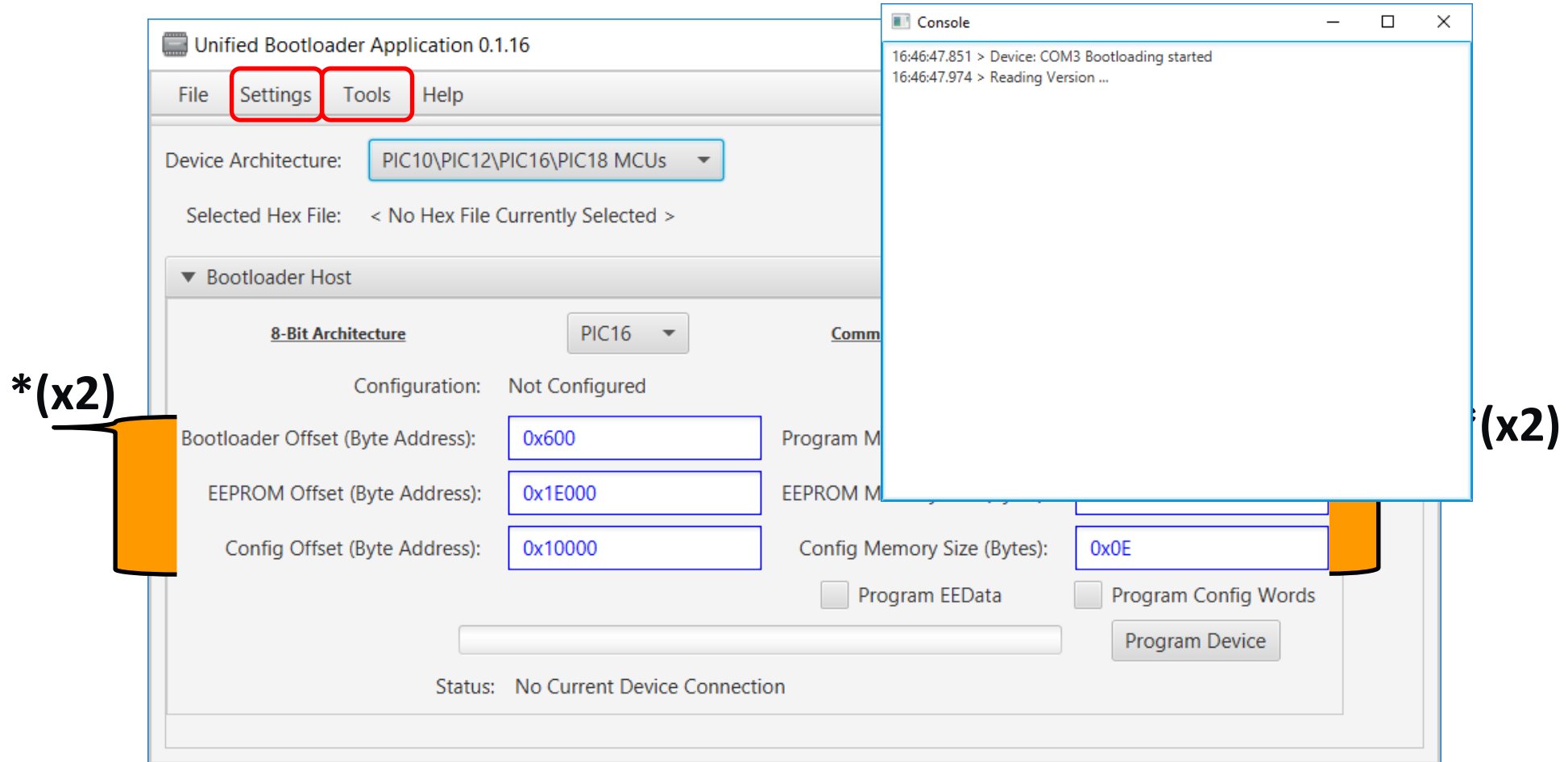
End of File

Agenda(Host Side)

- Scope of Responsibilities
- Bootloader Commands
- Hex File Format
- **Implementation Examples**

Unified Bootloader Host

- PIC(10|12|16): (8-bit) Word



Logger Support 🍷🍷

```
命令提示字元 java -Djava.util.logging.config.file="C:\MCHP\03_Development Tools\8-bit Bootloader\UnifiedHost-1.0.0\logging.properties" -jar UnifiedHost-1.0.0.jar
```

```

Microsoft Windows [版本 10.0.17134.590]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

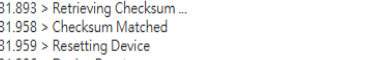
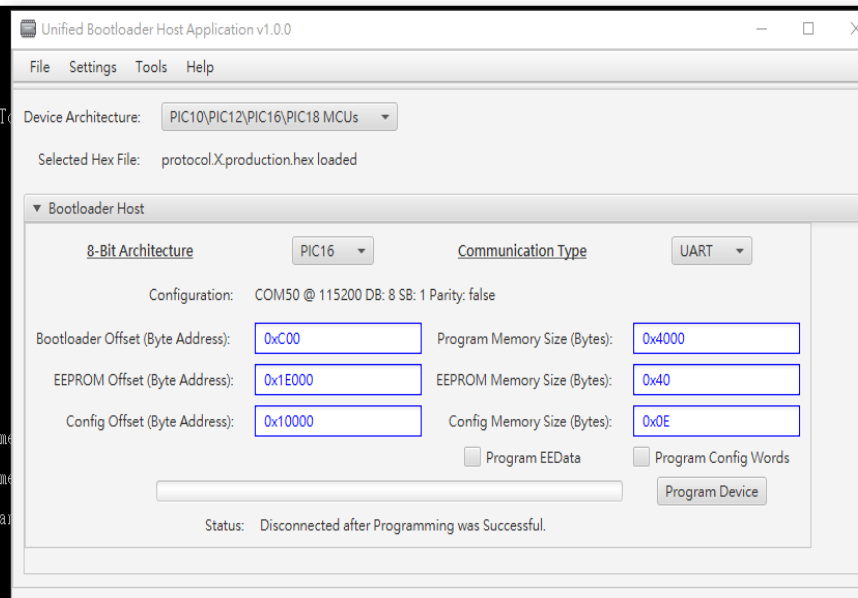
C:\Users\IA13929>cd C:\MCHP\03_Development Tools\8-bit Bootloader\UnifiedHost-1.0.0

C:\MCHP\03_Development Tools\8-bit Bootloader\UnifiedHost-1.0.0>java -Djava.util.logging.config.file="C:\MCHP\03_Development Tools\8-bit Bootloader\UnifiedHost-1.0.0\logging.properties" -jar UnifiedHost-1.0.0.jar
11:14:42.595 詳細 (unified.bootloader8bit.UBootFXMLController$1 changed) - UART Protocol Selected
11:14:42.617 詳細 (unified.bootloader8bit.UBootFXMLController$17 changed) - PIC16
11:14:50.469 詳細 (portSettings.model.PortSettingsPojo setDataBits) - 8
11:15:25.930 詳細 (portSettings.model.PortSettingsPojo setDataBits) - 8
11:15:29.586 詳細 (unified.bootloader8bit.UBootFXMLController ConnectSerial) - Trying to Connect: COM50
11:15:29.608 詳細 (commands.BootloadDFUFED setupDevice) - Device: COM50 Bootloading started
11:15:29.736 詳細 (readFileController.HexFileIO fread) - Writing 1024 Program Memory Starting@00
11:15:29.740 詳細 (readFileController.HexFileIO fread) - Writing 3722 Program Memory Starting@1ld2
11:15:29.744 詳細 (readFileController.HexFileIO fread) - Writing 2 Program Memory Starting@3ffe
11:15:29.747 詳細 (readFileController.HexFileIO fread) - Hex has 4 UnSupported Data @1000e
11:15:29.748 詳細 (commands.BootloadDFUFED execute) - 11:15:29.748 > Hex File (re)loaded ...

11:15:29.749 詳細 (commands.BootloadDFUFED execute) - 11:15:29.749 > Reading Version ...

11:15:29.751 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - READ_VER
11:15:29.752 詳細 (uartjsscadapter.JSSCSerialPortAdapter$SerialPortReader serialEvent) - Event Type: 1.Event Value: 1.Port Name: COM50
11:15:29.754 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 55
11:15:29.758 詳細 (uartjsscadapter.JSSCSerialPortAdapter$SerialPortReader serialEvent) - Event Type: 4.Event Value: 0.Port Name: COM50
11:15:29.758 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.760 詳細 (uartjsscadapter.JSSCSerialPortAdapter$SerialPortReader serialEvent) - Event Type: 1.Event Value: 26.Port Name: COM50
11:15:29.761 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.765 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.766 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.766 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.766 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.766 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.767 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.767 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.767 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.768 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.769 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - READ_VER_DONE
11:15:29.770 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 55
11:15:29.770 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.770 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.771 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.771 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.772 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.772 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.772 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.772 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.774 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.774 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.775 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 07
11:15:29.775 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.776 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.776 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 01
11:15:29.777 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.778 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 00
11:15:29.779 詳細 (i2cBootloaderController.mcu8.ACommandBaseExtendedNoDataPayload process) - 85

```



The screenshot shows a Windows-style console window titled "Console". It contains a list of commands and their outputs, each preceded by a timestamp. The process starts with "Device: COM50 Bootloading started" at 11:15:29.606, followed by "Hex File (re)loaded..." at 11:15:29.748. Then, "Reading Version ..." is shown at 11:15:29.749, and "Bootloader Version Read Successful" at 11:15:29.785. The next steps are "Erasing Device ..." at 11:15:29.788 and "Erase Successful" at 11:15:30.103. This is followed by "Programming Flash ..." at 11:15:30.104 and "Flashed Programmed" at 11:15:31.888. The process then shows "Retrieving Checksum ..." at 11:15:31.893 and "Checksum Matched" at 11:15:31.958. Finally, "Resetting Device" is shown at 11:15:31.959, "Device Reset" at 11:15:31.986, and the process concludes with "Device: COM50 Bootloading Successful" at 11:15:31.986.

```

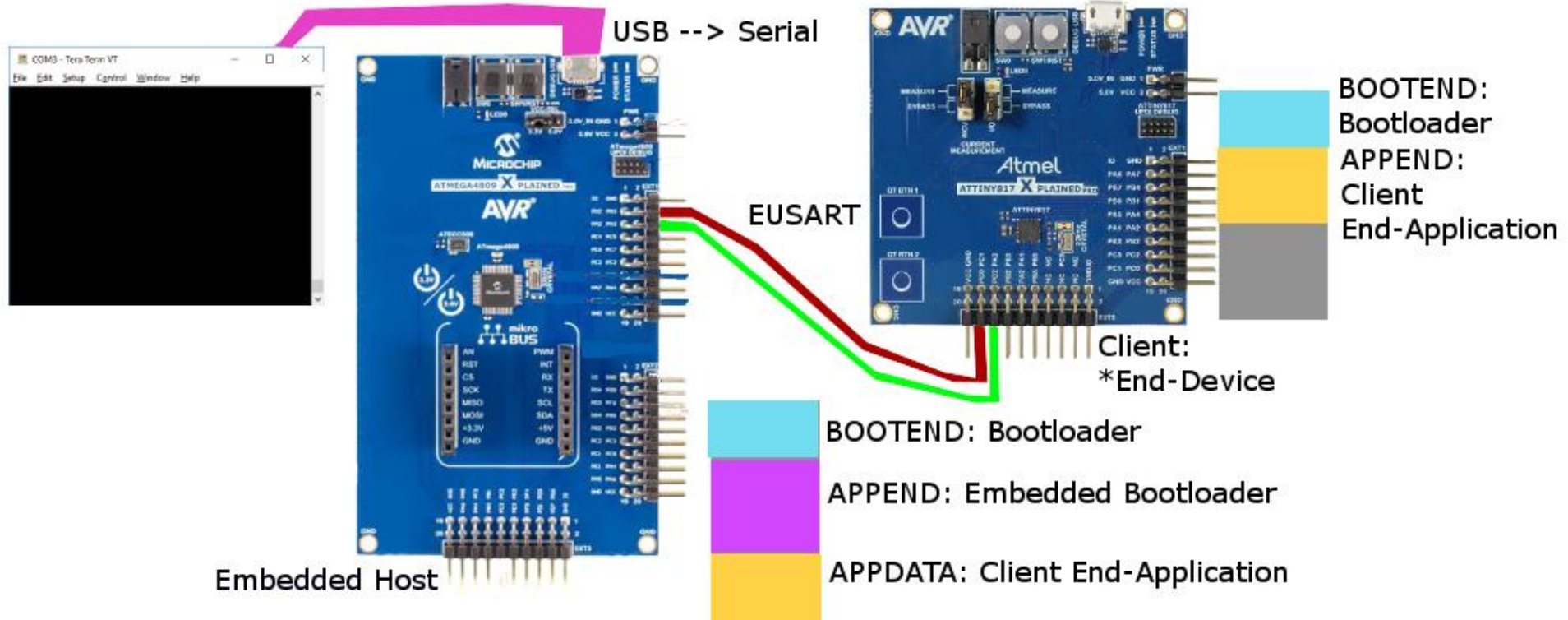
11:15:29.606 > Device: COM50 Bootloading started
11:15:29.748 > Hex File (re)loaded...
11:15:29.749 > Reading Version ...
11:15:29.785 > Bootloader Version Read Successful
11:15:29.788 > Erasing Device ...
11:15:30.103 > Erase Successful
11:15:30.104 > Programming Flash ...
11:15:31.888 > Flashed Programmed
11:15:31.893 > Retrieving Checksum ...
11:15:31.958 > Checksum Matched
11:15:31.959 > Resetting Device
11:15:31.986 > Device Reset
11:15:31.986 > Device: COM50 Bootloading Successful
  
```

Supported Protocols

- **MCC Bootloader Generated Code**
 - PIC16, PIC18 Family of Device
 - AVR (ATTiny 0, 1; ATmega 0)
 - PIC24, dsPIC
- **Microchip MLA/Harmony**
 - PIC32MX/MZ/CZ
 - SAM

Embedded Bootloader Host

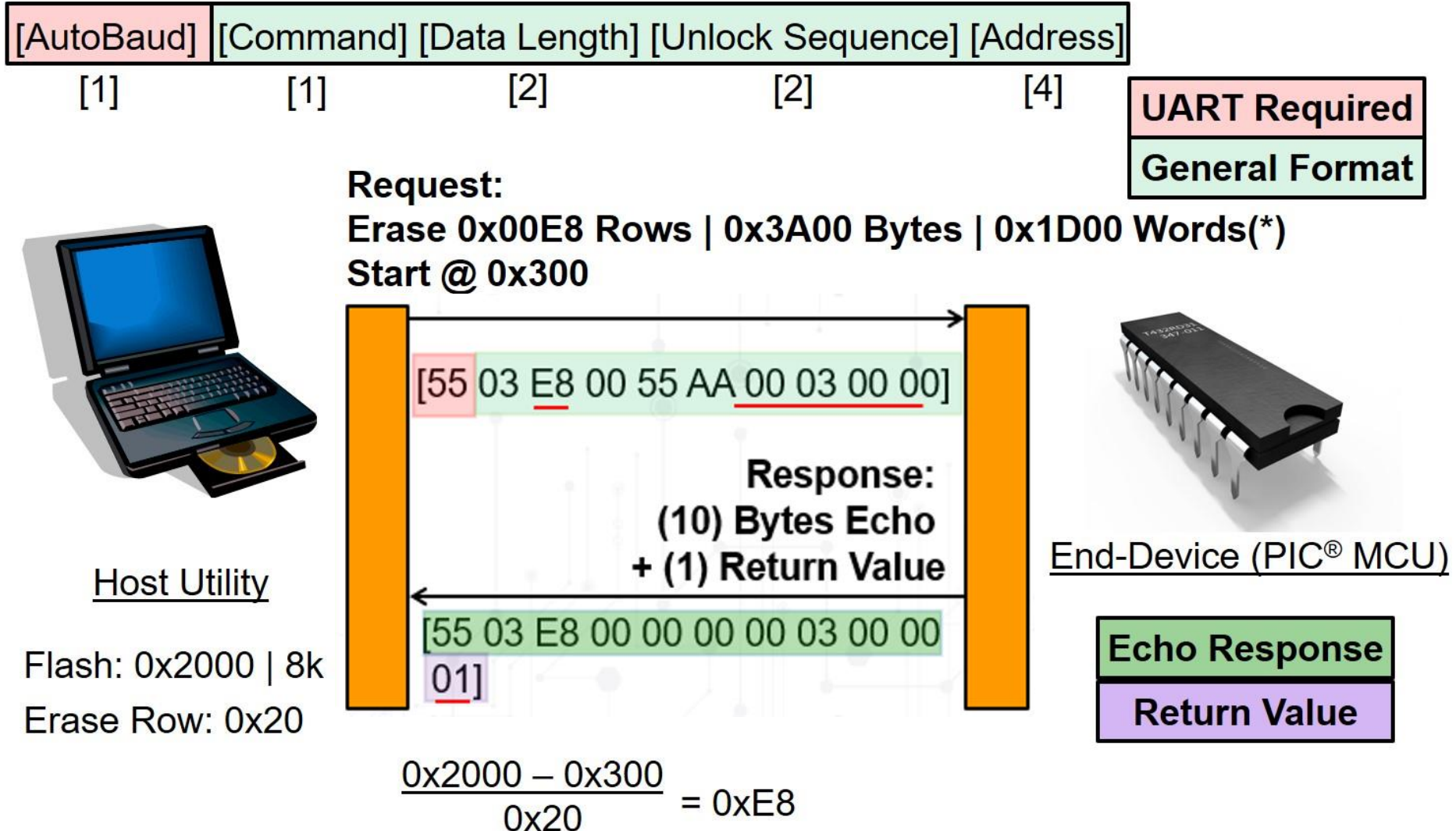
- EBH: ATMega4809 → ATTiny817



Typical Commands for EBH

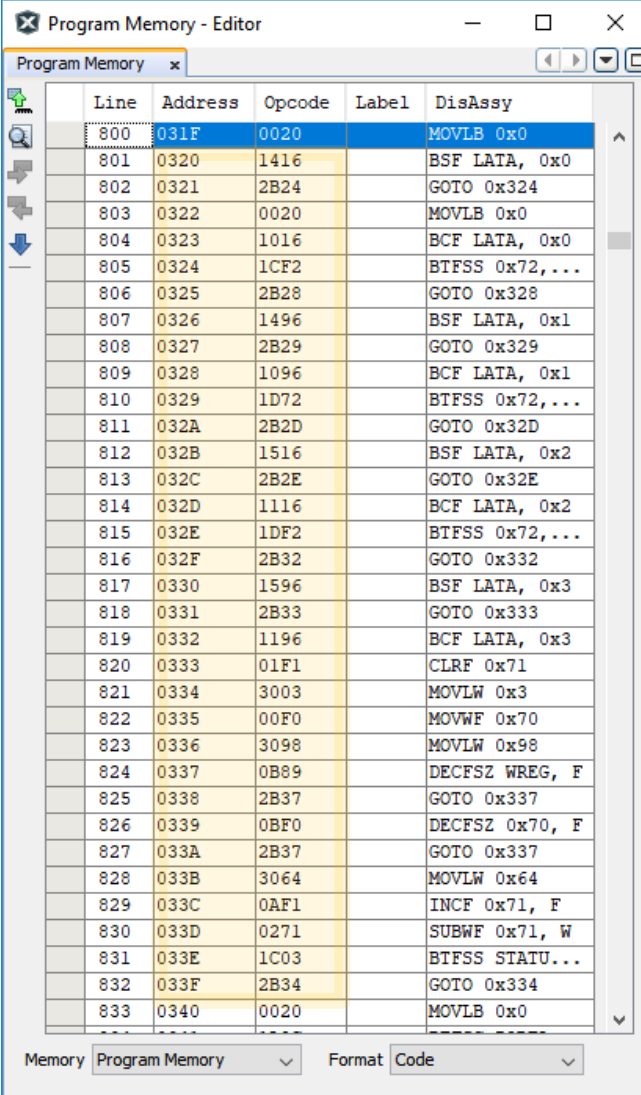
0 – Get Version & More	(1)
1 – Read Flash Memory	
2 – Write Flash Memory	(3)
3 – Erase Flash Memory	(2)
4 – Read EE Data	
5 – Write EE Data	(4)
6 – Read Config Words	
7 – Write Config Words	
8 – Calculate Checksum	(5)
9 – Reset Device	(6)

3 – Erase Flash Memory



2 – Write Flash Memory

```
:06060000000000000000000000000000
:10060800000000000000000000000000
:10061800000000000000000000000000
:10062800000000000000000000000000
:10063800000000000000000000000000
:10064800F21C282B9614292B9610721D2D2B16158B
:100658002E2B1611F21D322B9615332B9611F10104
:100668000330F0009830890B372BF00B372B6430B0
:10067800F10A7102031C342B20008C1A182B5C23FE
:1006880083314C23833120008C1E472B0100182B0B
:100698009A0103309B009C019D0114309E00553047
:1006A8009F00AA309F009E140800742383316B2B8F
:1006B800FE3030009A001F309B009C019D011430D1
:1006C8009E0055309F00AA309F009E14080060309D
:1006D80031008D008F0191010330930092010800D1
:1006E80097013E00E5012000960198013E00B901FE
:1006F800C401CF01D030B800FF30C300CE002000C5
:1007080092009300F030910008008C127308740472
:100718000319080074088A0073080A008331080066
:020728000800C7
:020000040001F9
:0A000E008C3FFE3B9F3FFF3FFF3F8A
:10E0000000000100020003000400050006000700F4
:10E01000080009000A000B000C000D000E000F00A4
:00000001FF
```



Program Memory - Editor

Line	Address	Opcode	Label	DisAssy
800	031F	0020		MOVLB 0x0
801	0320	1416		BSF LATA, 0x0
802	0321	2B24		GOTO 0x324
803	0322	0020		MOVLB 0x0
804	0323	1016		BCF LATA, 0x0
805	0324	1CF2		BTFS 0x72,...
806	0325	2B28		GOTO 0x328
807	0326	1496		BSF LATA, 0x1
808	0327	2B29		GOTO 0x329
809	0328	1096		BCF LATA, 0x1
810	0329	1D72		BTFS 0x72,...
811	032A	2B2D		GOTO 0x32D
812	032B	1516		BSF LATA, 0x2
813	032C	2B2E		GOTO 0x32E
814	032D	1116		BCF LATA, 0x2
815	032E	1DF2		BTFS 0x72,...
816	032F	2B32		GOTO 0x332
817	0330	1596		BSF LATA, 0x3
818	0331	2B33		GOTO 0x333
819	0332	1196		BCF LATA, 0x3
820	0333	01F1		CLRF 0x71
821	0334	3003		MOVLW 0x3
822	0335	00F0		MOVWF 0x70
823	0336	3098		MOVLW 0x98
824	0337	0B89		DECFSZ WREG, F
825	0338	2B37		GOTO 0x337
826	0339	0BF0		DECFSZ 0x70, F
827	033A	2B37		GOTO 0x337
828	033B	3064		MOVLW 0x64
829	033C	0AF1		INCF 0x71, F
830	033D	0271		SUBWF 0x71, W
831	033E	1C03		BTFS STATU...
832	033F	2B34		GOTO 0x334
833	0340	0020		MOVLB 0x0

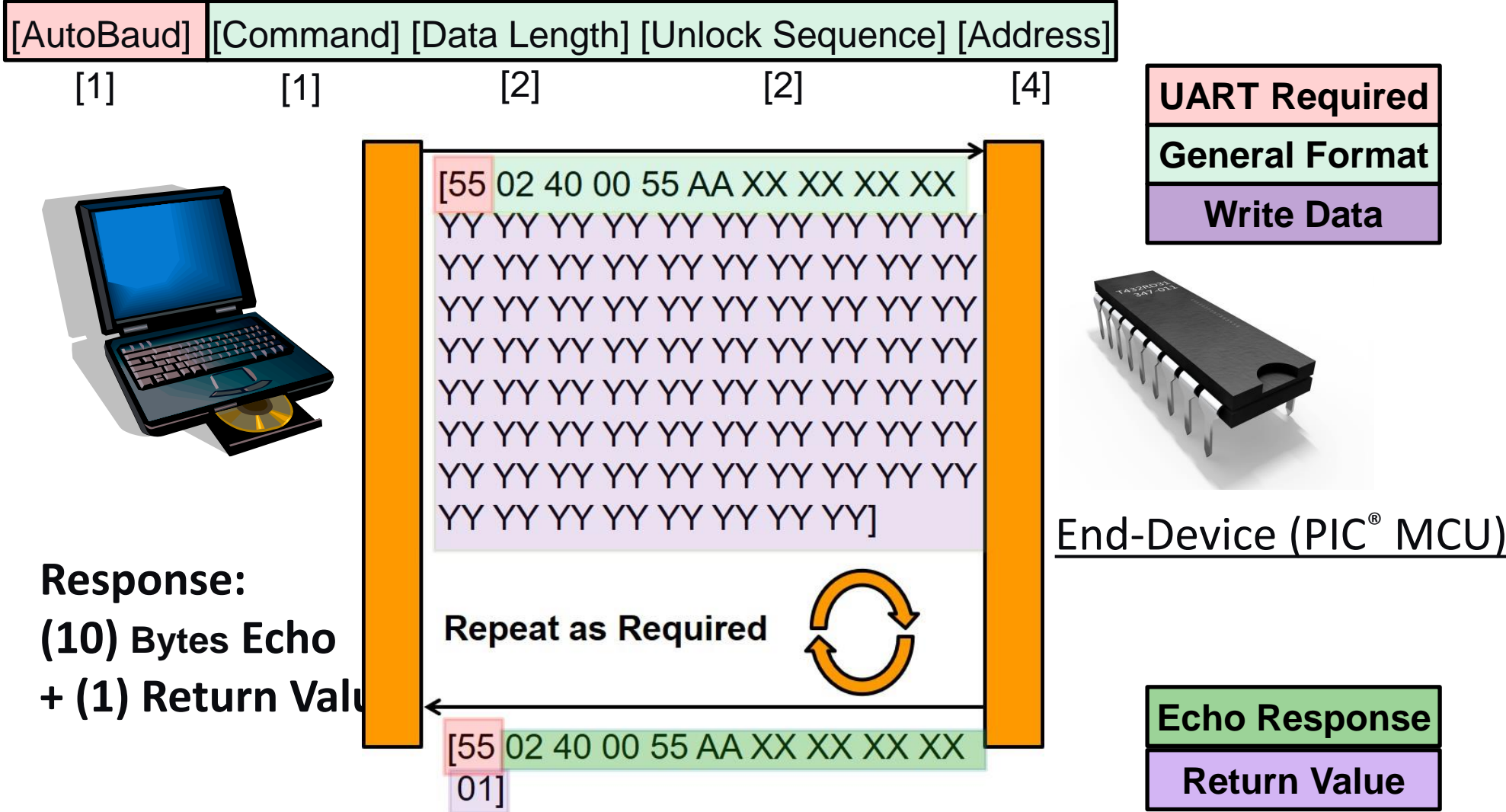
Memory Program Memory Format Code

2 – Write Flash Memory

1614242B2000161006
:10064800F21C282B9614292B9610721D2D2B16158B
:100658002E2B1611F21D322B9615332B9611F10104
:100668000330F0009830890B372BF00B372B6430B0
:10067800F10A7102031C342B

[55 02 40 00 55 AA 20 03 00 00
16 14 24 2B 20 00 16 10 F2 1C
28 2B 96 14 29 2B 96 10 72 1D
2D 2B 16 15 2E 2B 16 11 F2 1D
32 2B 96 15 33 2B 96 11 F1 01
03 30 F0 00 98 30 89 0B 37 2B
F0 0B 37 2B 64 30 F1 0A 71 02
03 1C 34 2B]

2 – Write Flash Memory



*PIC16 Word vs Byte | * Hex Values

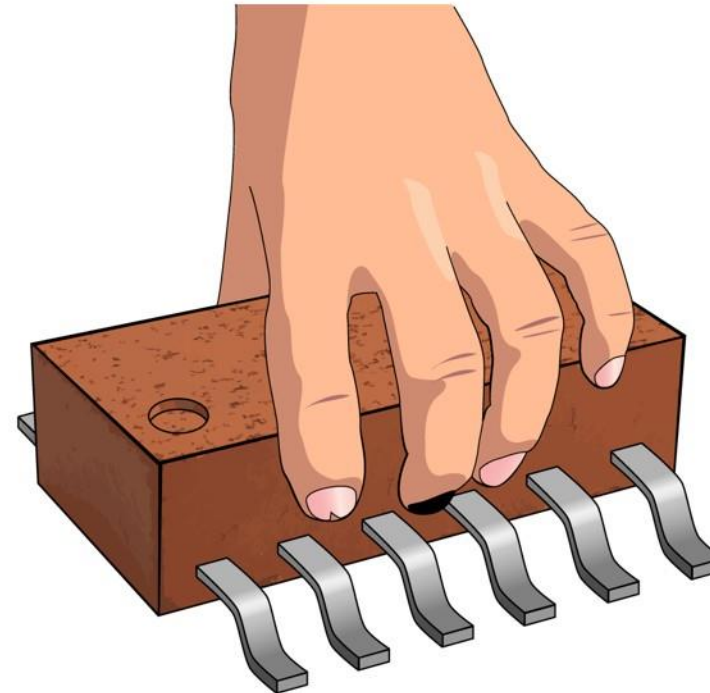
2 – Write Flash Memory

```
:06060000000083310E2B07
:100608007E1483312E00961A8C1E0C2B833189237D
:100618007E1009008331102BF301F4017E102000B5
:10062800152BF20159238331F23672080830031969
:10063800F200721C222B20001011212E2000101000
:10064800F21C222B20001011212E2000101000
:100658002E2B1011F21D022B0010002B0011F10101
:100668000550F00050505050B572B100B572B0450B5
:10067800F40A7402024C242B20008C4A482B5C225E
:1006880082314C23833120008C4E472B0100182B0B
:100698005A010000000000000000000000000000
:1006A8000100A000F000E140000142000010B2B01
:1006B800FE0000000A004F0000B0000040B044400B4
:1006C800E0005000F000A0000F000E44000000000
:1006D8000A0000B000F040404000000000000000
:1006E80057015E00E5012000500150015E00B501E
:1006F800C404CE04D020B800FE20C200CE002000C5
:1007080092009200E020940008008C427208740472
:100718000515000074000A0075000A000551000000
:02072800000007
:020000040001F9
:0A000E008C3FFE3B9F3FFF3FFF3FF8A
:10E0000000000100020003000400050006000700F4
:10E01000080009000A000B000C000D000E000F00A4
:00000001FF
```

INTEL Format

Written Data

1st and Final Chunk
To Write



8 – Calculate Checksum

[AutoBaud]	[Command]	[Data Length]	[Unlock Sequence]	[Address]
[1]	[1]	[2]	[2]	[4]

UART Required
General Format



```
Command Prompt
5508003A000000030000READ_VER_DONE
85
8
0
58
0
0
0
0
3
0
0
-120
-36
Theoretical Checksum:726DC88
Receieved Checksum:DC88
READ_VER
```



Device (PIC® MCU)

Response

Return Value

0x2000 – 0x300

0x4000 – 0x600 = 0x3A00