

Example Code 

<https://github.com/EdwardLeeTW/POW201>



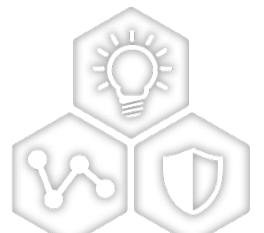
POW201

Hands-ON Manual

動手實驗參考手冊



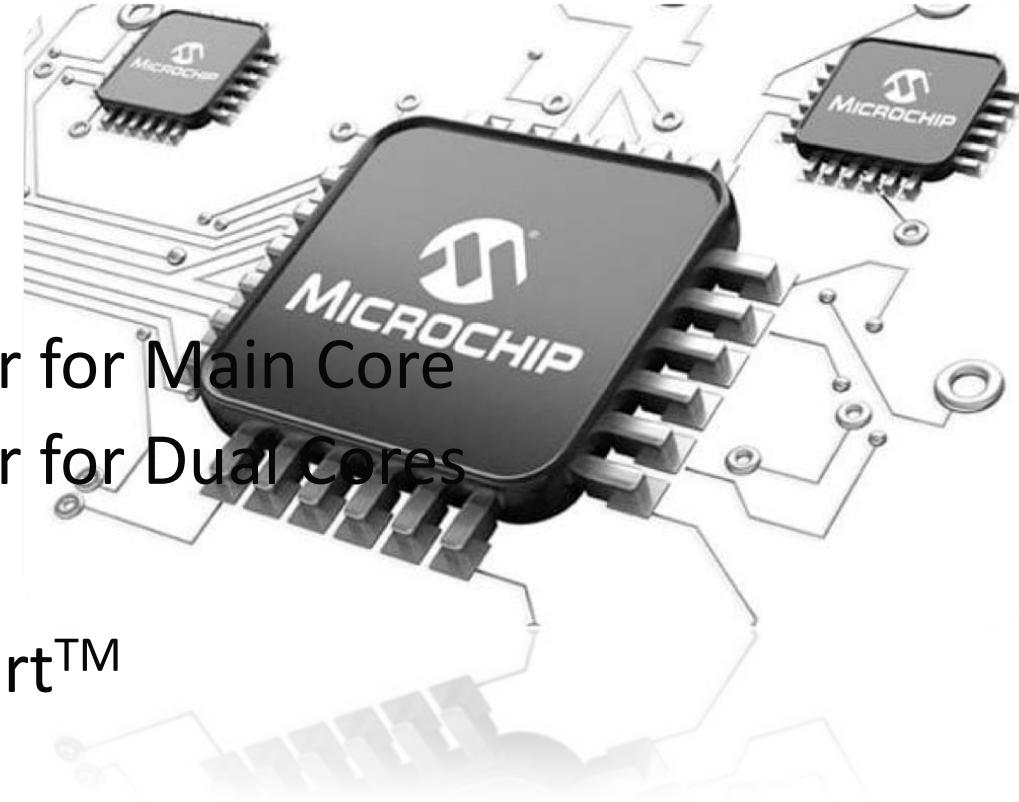
A Leading Provider of Smart, Connected and Secure Embedded Solutions



Edward Lee
Oct. 2, 2023

Agenda

- **Digital SMPS Exercises – Part-I**
 - Lab1: MCC Configuration with OS Scheduler for Main Core
 - Lab2: MCC Configuration with OS Scheduler for Dual Cores
- **Digital SMPS Exercises – Part-II**
 - Lab3: Plant Measurement Using PowerSmart™
 - Lab4: Closed-loop Control



Digital SMPS Exercises – Part-I

Hardware Setup

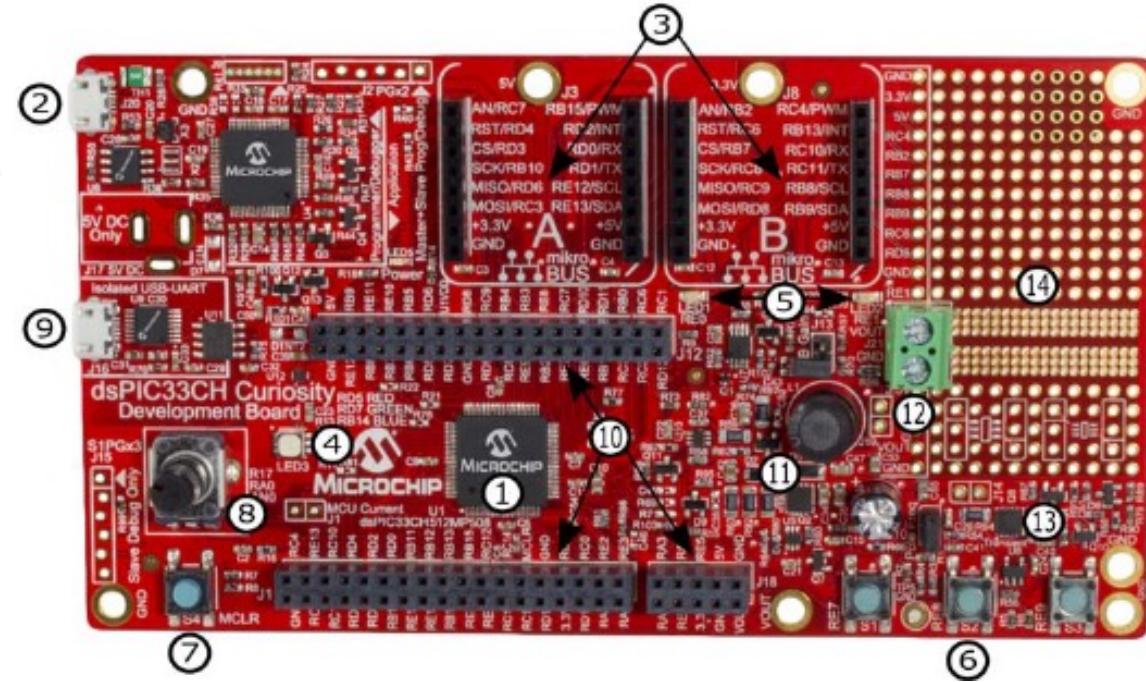
Pre-work

Curiosity DM330028-2 For dsPIC33CH512MP508

<https://www.microchip.com/en-us/development-tool/dm330028-2>

Hardware Features:

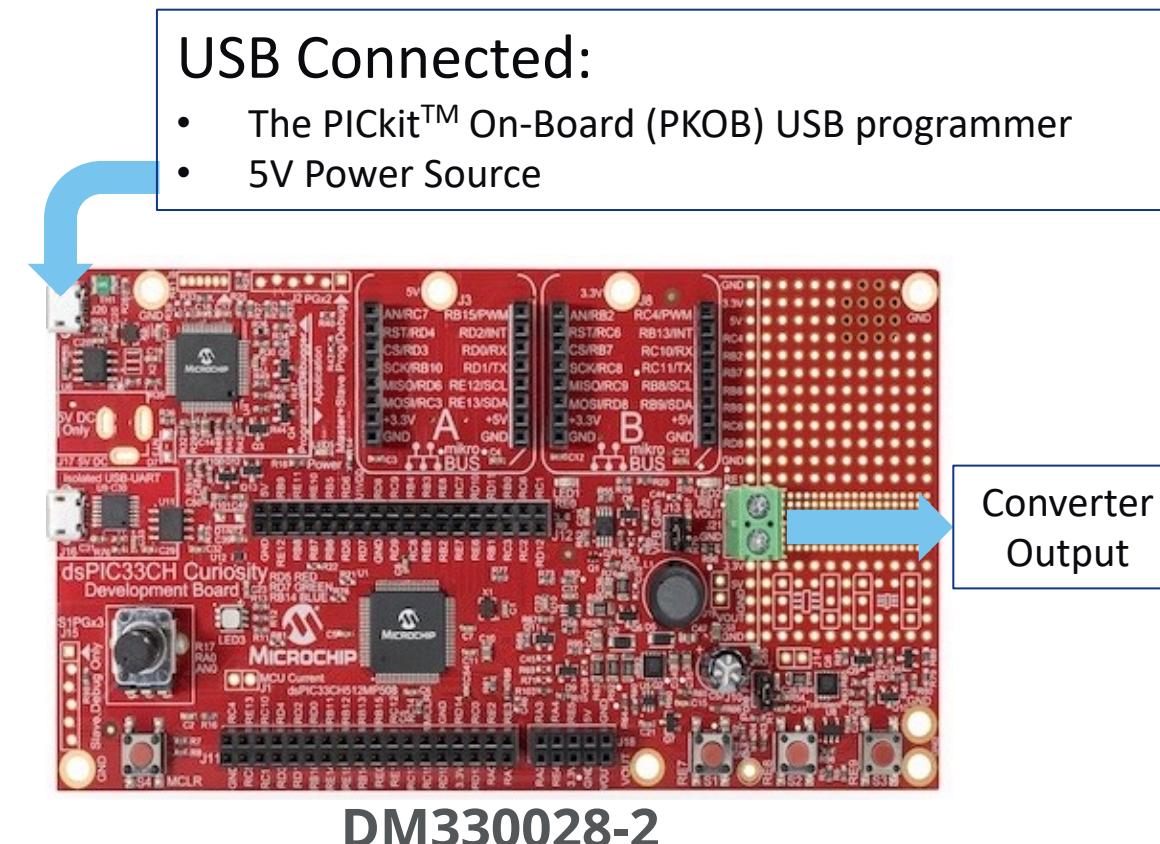
1. dsPIC33CH512MP508 dual core, 16-bit DSP target device.
2. Integrated PICkit™-On-Board (PKOB) programmer/debugger.
3. 2x mikroBUS™ interfaces for hardware expansion, compatible with a wide range of existing click boards™ from MikroElektronika (www.mikroe.com).
4. 1x Red/Green/Blue (RGB) LED.
5. 2x general purpose red indicator LEDs.
6. 3x general purpose push buttons.
7. 1x MCLR Reset push button.
8. 10k potentiometer.
9. Galvanically isolated USB-UART interface, capable of up to 460,800 baud.
10. Female, 100 mil pitch, I/O pin access headers for probing and connecting to all target microcontroller GPIO pins.
11. Configurable Switch Mode Power Supply (SMPS) test circuit that can be operated in Buck, Boost, or Buck-Boost modes, using either Voltage mode or Peak Current mode control.
12. Converter output voltage screw terminal.
13. Configurable load step transient generator.
14. General purpose through-hole and SMT prototyping area.



DM330028-2

dsPIC33CH Curiosity Dev. Board for Hand-ON

- Main chip: dsPIC33CH512MP508
 - Configurable SMPS test circuit that can be operated in Buck, Boost, or Buck-Boost modes, using either Voltage mode or Peak Current mode control
 - PWM: RC14_S1PWM7H
- Buck Mode:
 - Vin = 5V USB Power
 - Vout = 3.3V
 - Iout = Max 0.5A
 - L = 33 μ H
 - C = 150 μ F, ESR = 390 m Ω
 - f_{sw} = 250 kHz



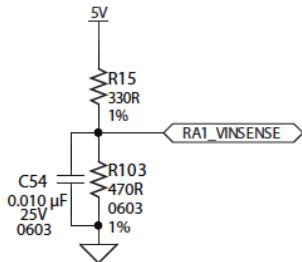
dsPIC33CH Curiosity Dev. Board for Hand-ON

Main Core

Secondary Core

Power Stage with current sense, H/w OCP and HW OVP

Input voltage monitoring



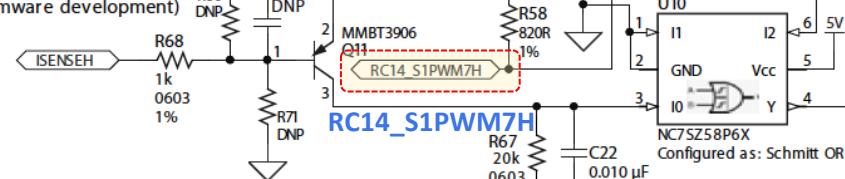
Configurable Buck, Boost or Buck/Boost Test Circuit

For Buck Mode: PWM Q6, Drive Q2 DC OFF

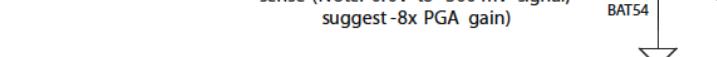
For Boost Mode: Drive Q6 DC OFF (logic high or tri-state), PWM Q2

For Buck/Boost Mode: PWM Q6 and Q2 with same signal (Note: Q6 drive should be active-low, Q2 active-high)

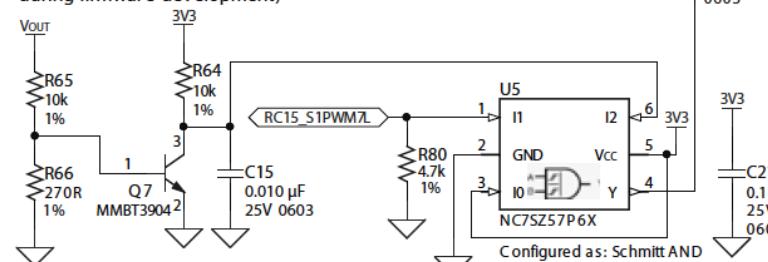
Hardware overcurrent protection (useful during firmware development)



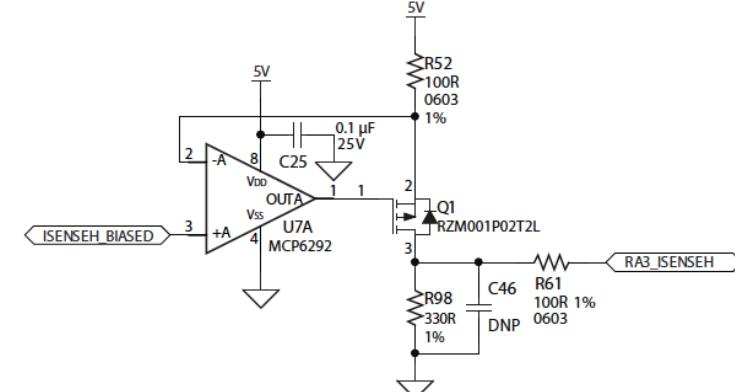
Buck mode low-side off-time current sense (Note: 0.0V to -300 mV signal, suggest -8x PGA gain)



Output overvoltage protection (useful during firmware development)



High-side current sense level shifter



EQUATION 2-4:

$$ISENSEH_BIASED = (VIN - IQ6 \cdot Rsense) \frac{R102}{(R102 + R97)}$$

EQUATION 2-5:

$$RA3_ISENSEH = \frac{R98}{R52} (VIN - ISENSEH_BIASED)$$

EQUATION 2-6:

$$RA3_ISENSEH = \frac{R98}{R52} \left[VIN - \left(\frac{(VIN - IQ6 \cdot Rsense)(R102)}{R102 + R97} \right) \right]$$

dsPIC33CH Curiosity Dev. Board for Hand-ON

Main Core

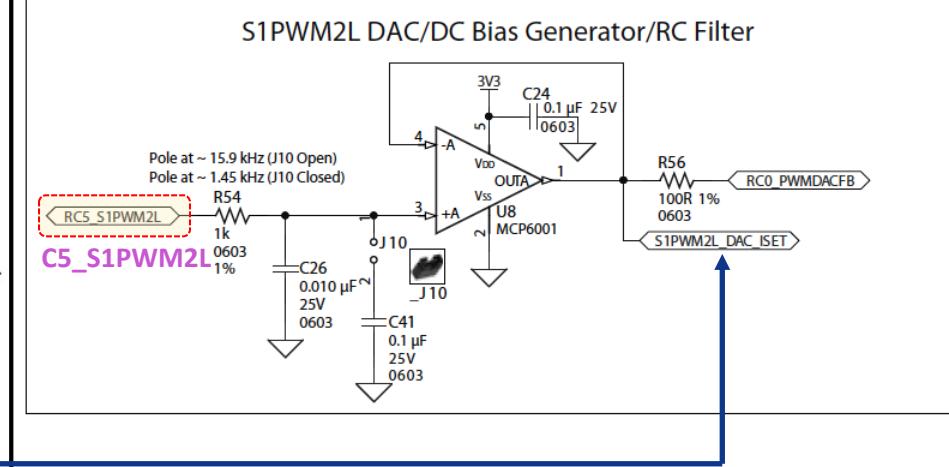
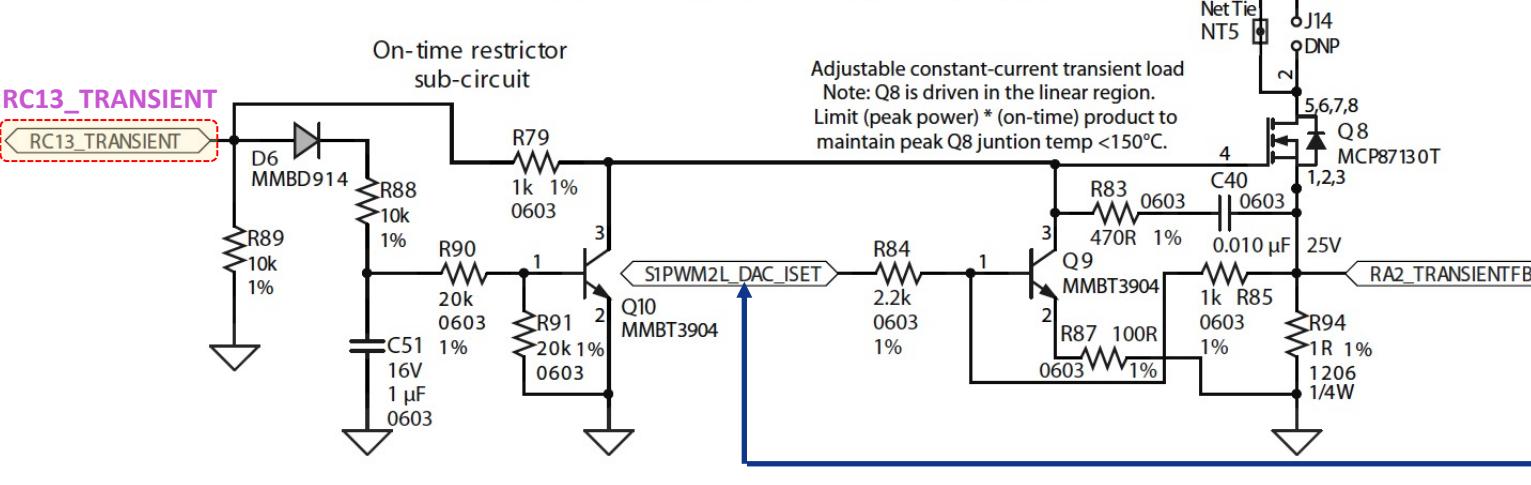
Secondary Core

Configurable load step transient generator

- Output I/O Pin: **RC13_TRANSIENT**
- Output PWM Pin as DAC Output: **RC5_S1PWM2L**
- Reserved:
 - Input AN Pin: **RA2_TRANSIENTFB**
 - Input AN Pin: **RC0_PWMDACFB**

Adjusting the PWM waveform duty cycle on RC5_S1PWM2L by +1.0% alters the Q8 constant-current sink value by approximately -12 mA. At 50% PWM duty cycle, the approximate current sink level is around 390 mA, but will vary somewhat between boards and at different ambient temperatures, as these will affect the Q9 turn-on voltage. For exact current sink values, it is necessary to use closed-loop control by measuring the RA2_TRANSIENTFB current sense voltage with the ADC at run time. Then, using the resulting value to fine-tune adjust the PWM duty cycle on RC5_S1PWM2L.

Transient Load Tester Circuit

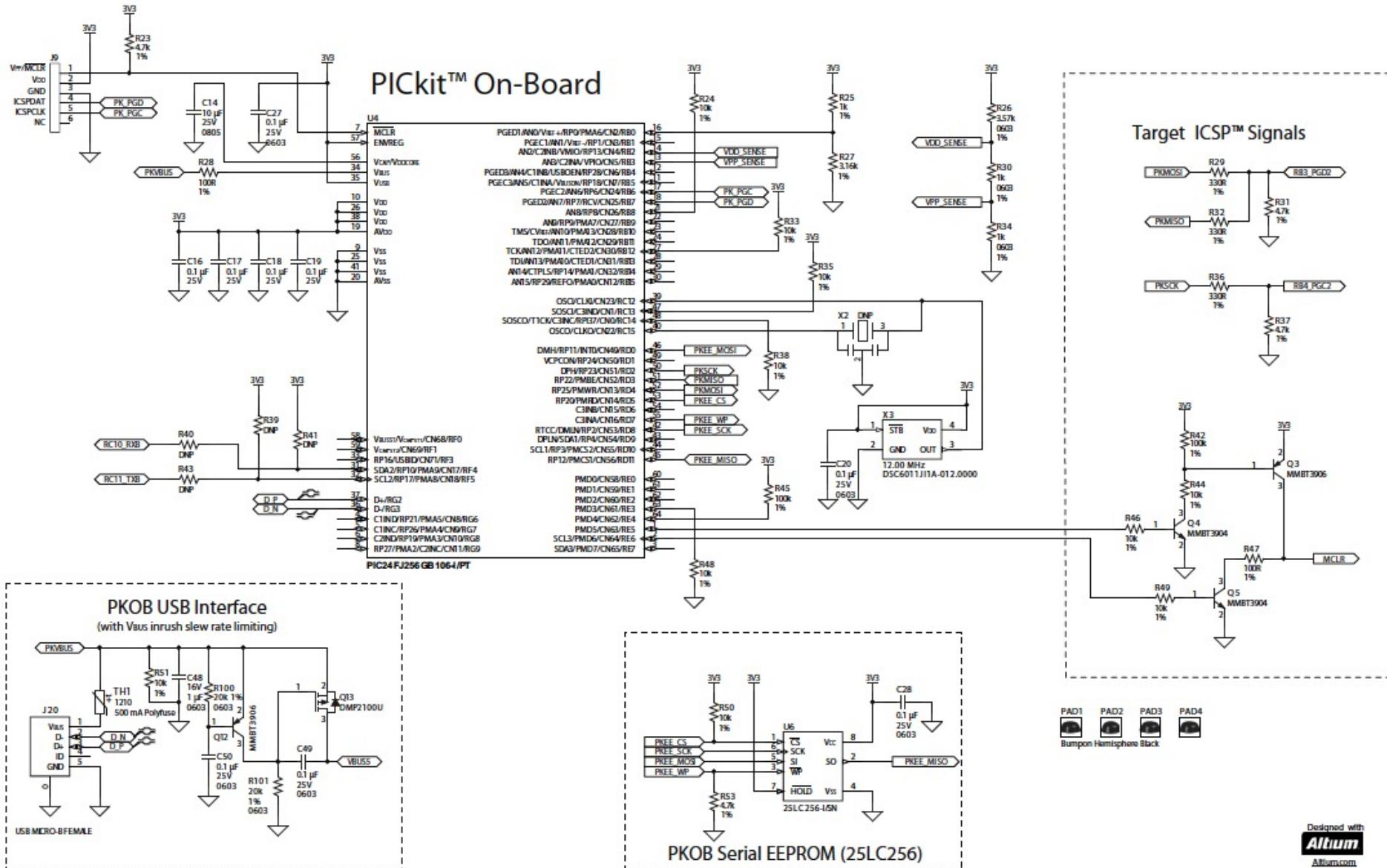


dsPIC33CH Curiosity Dev. Board for Hand-ON

Main Core

Secondary Core

PKOB

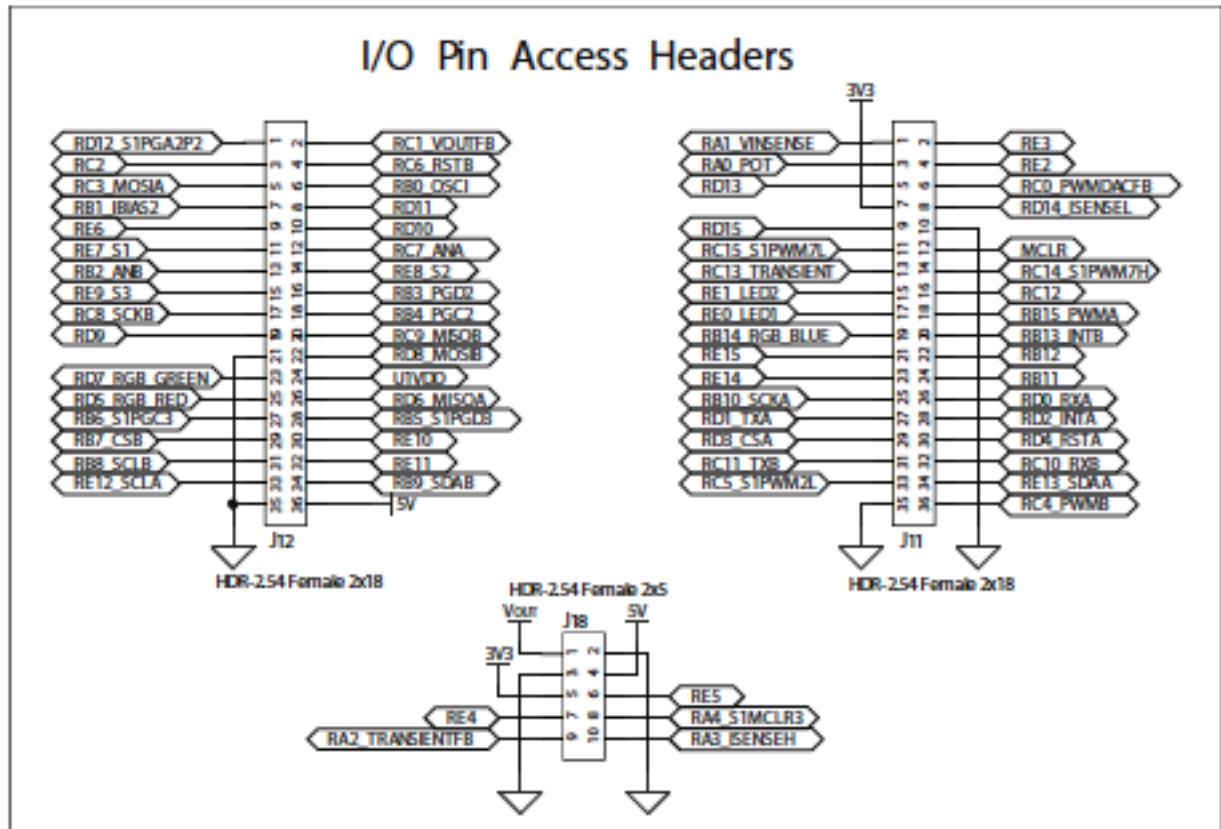
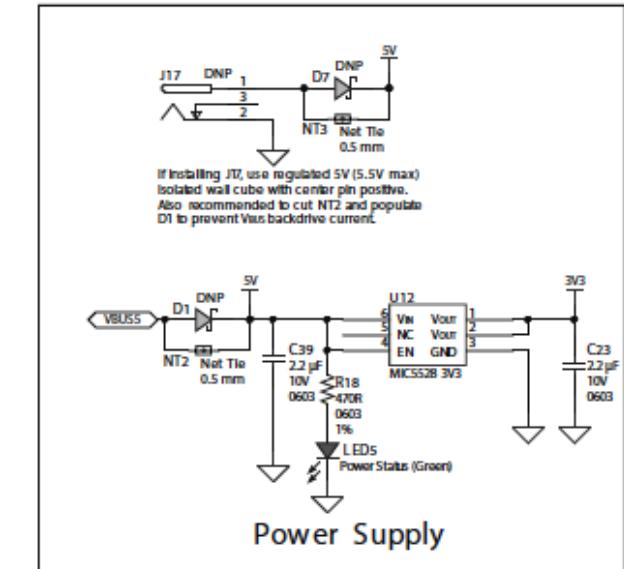
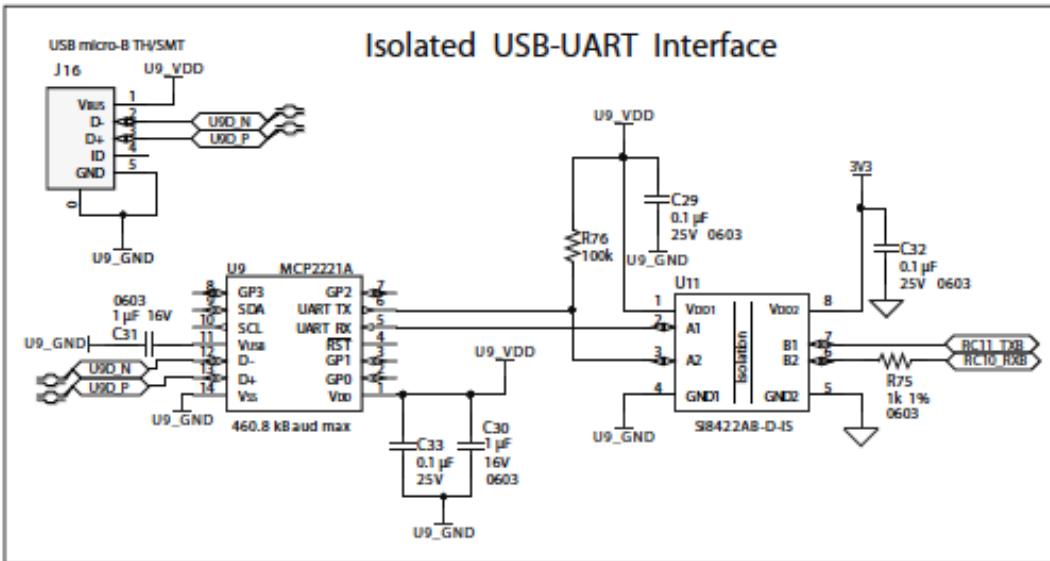


dsPIC33CH Curiosity Dev. Board for Hand-ON

Main Core

Secondary Core

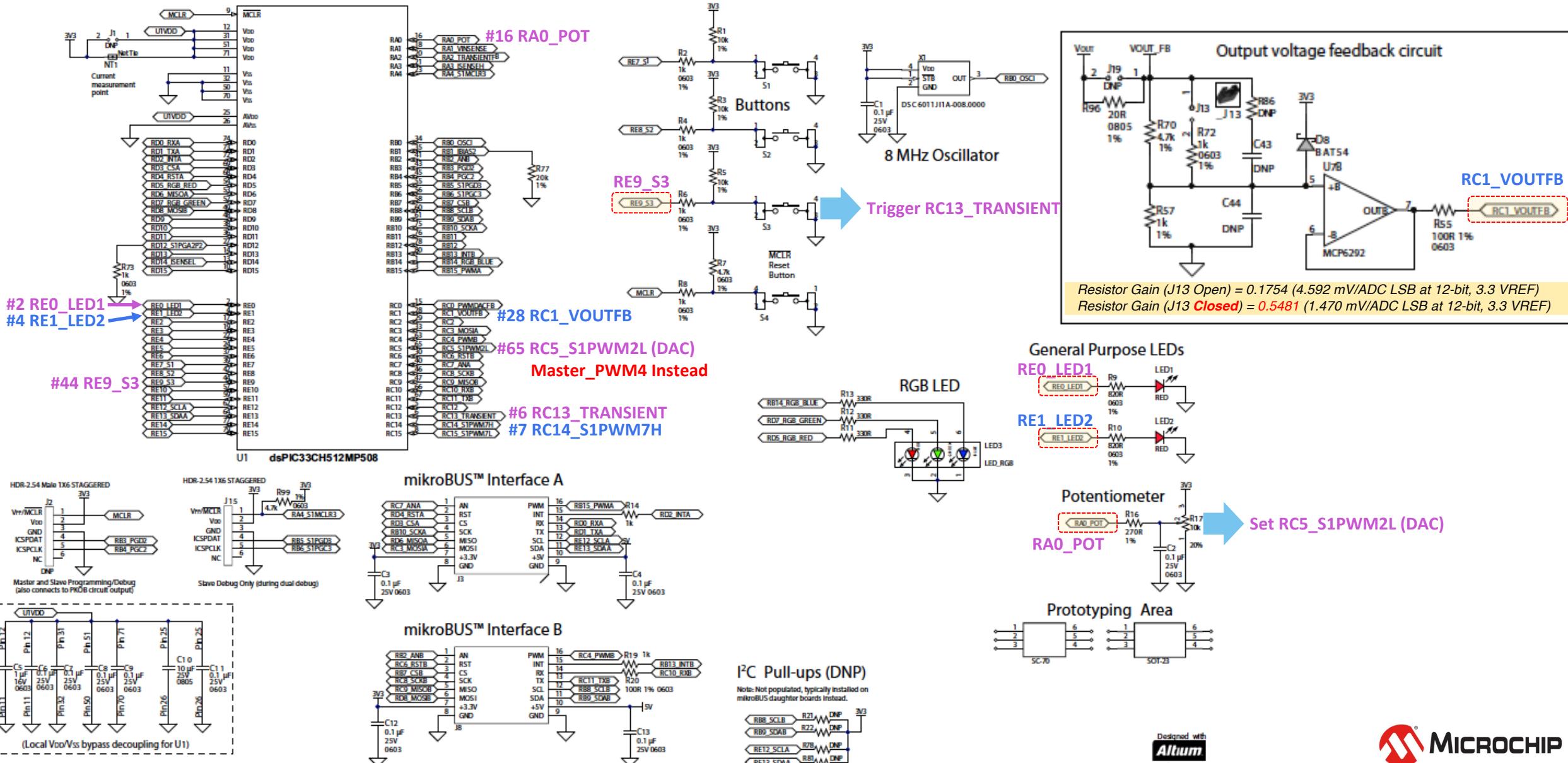
MCU & Interface



dsPIC33CH Curiosity Dev. Board for Hand-ON

Main Core

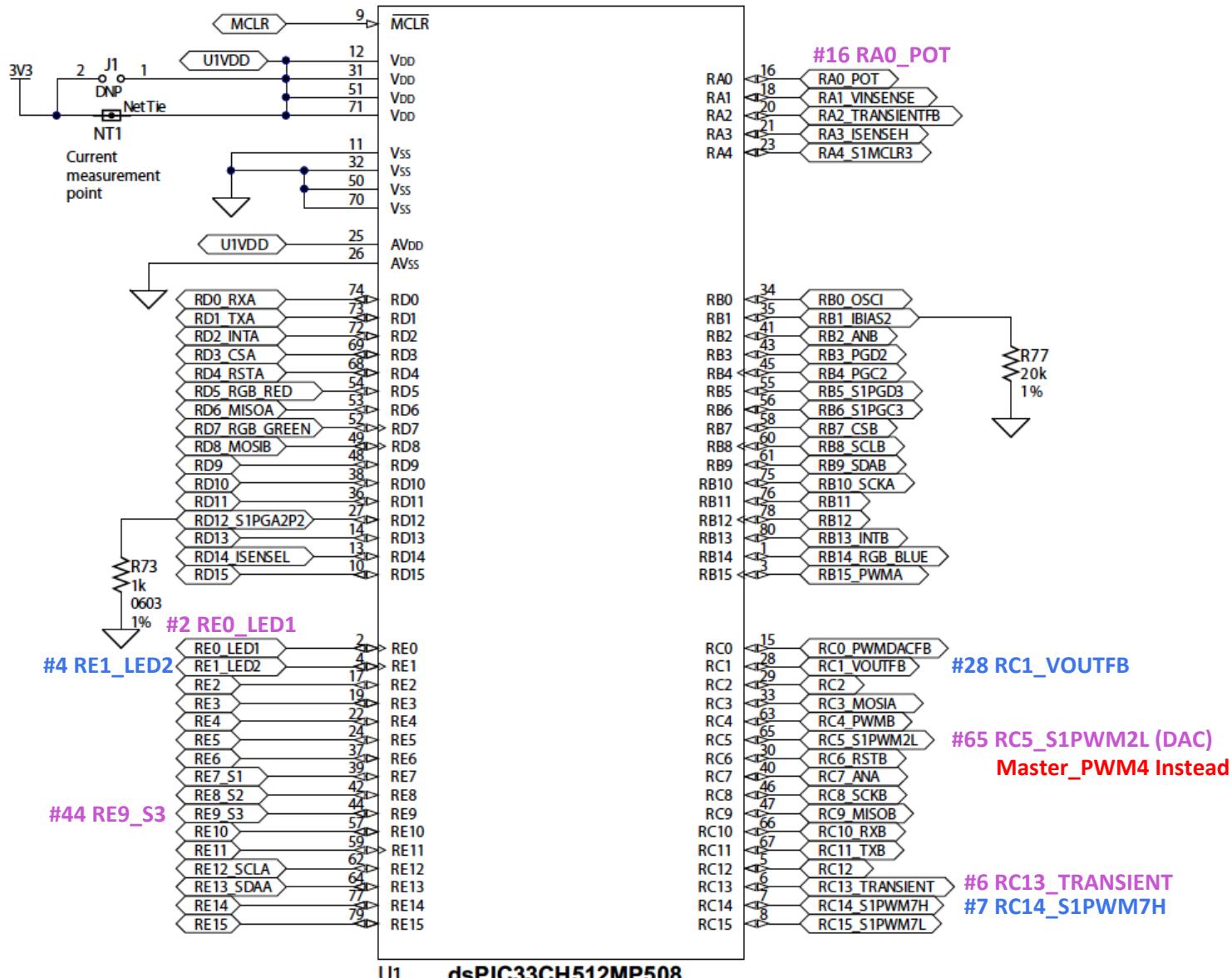
Secondary Core



dsPIC33CH Curiosity Dev. Board for Hand-ON

Main Core

Secondary Core



U1 dsPIC33CH512MP508

Software Tools

Pre-work

Software Tools

- Software requirements (Versions for now)

- MPLAB® X IDE V6.10



<https://www.microchip.com/ide>

- MPLAB XC Compiler (**XC16**) V2.1



<https://www.microchip.com/xc>

- MPLAB Code Configurator (**MCC**) V5.3.7



- Core v5.5.7
- Device Library: dsPIC33 v1.171.2

- Optional Software

- MPLAB Mindi™ Analog Simulator

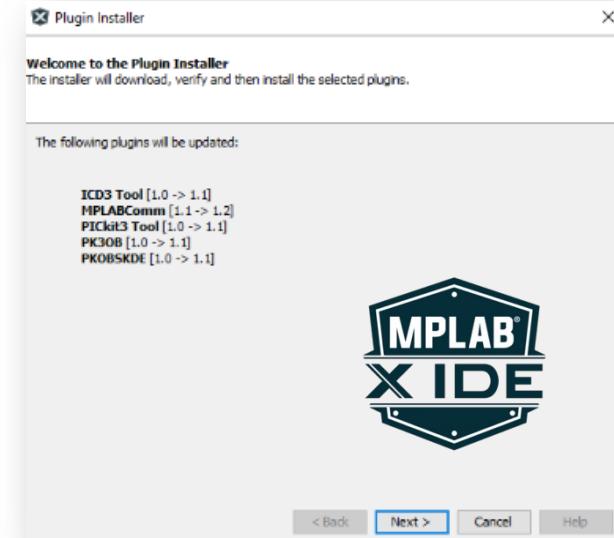


<https://www.microchip.com/mindi>

- MPLAB PowerSmart™-DCLD



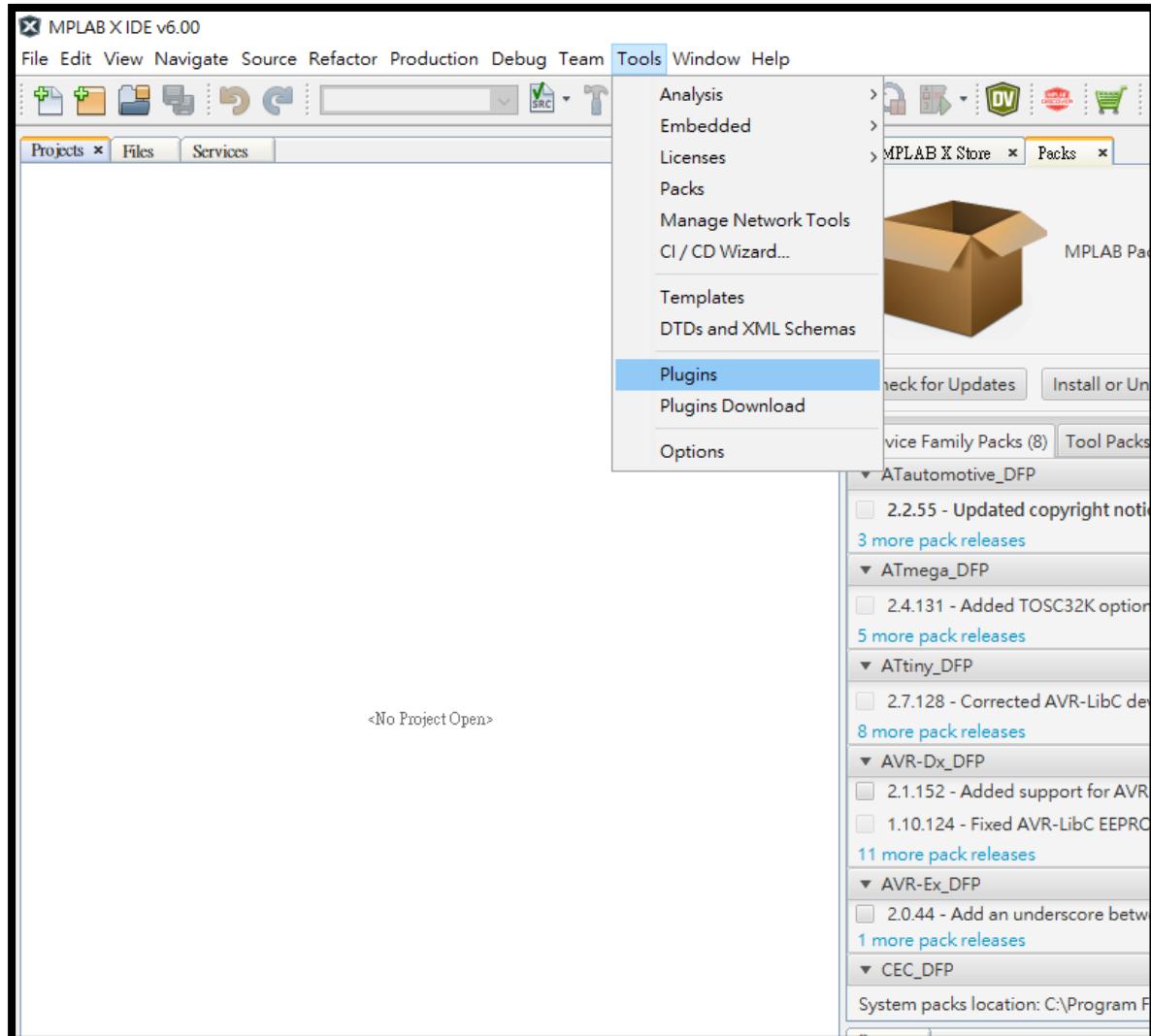
<https://www.microchip.com/en-us/solutions/power-management-and-conversion/intelligent-power/mplab-powersmart-development-suite>



Plug-in updates for MPLAX
V6.10 to fix the USB
communication issues which
were preventing Generation 3
tools support.

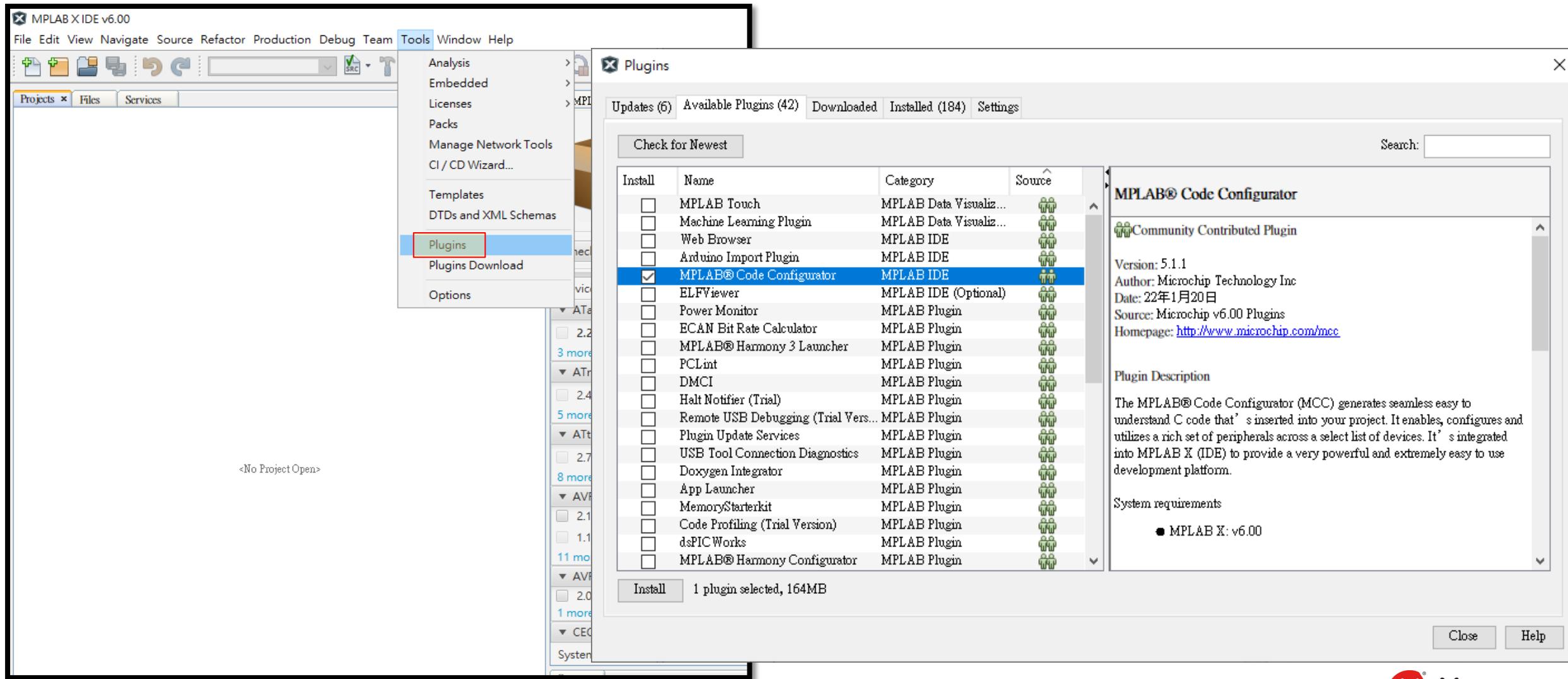
Installing MCC

MPLAB® X IDE / Tools / Plugins



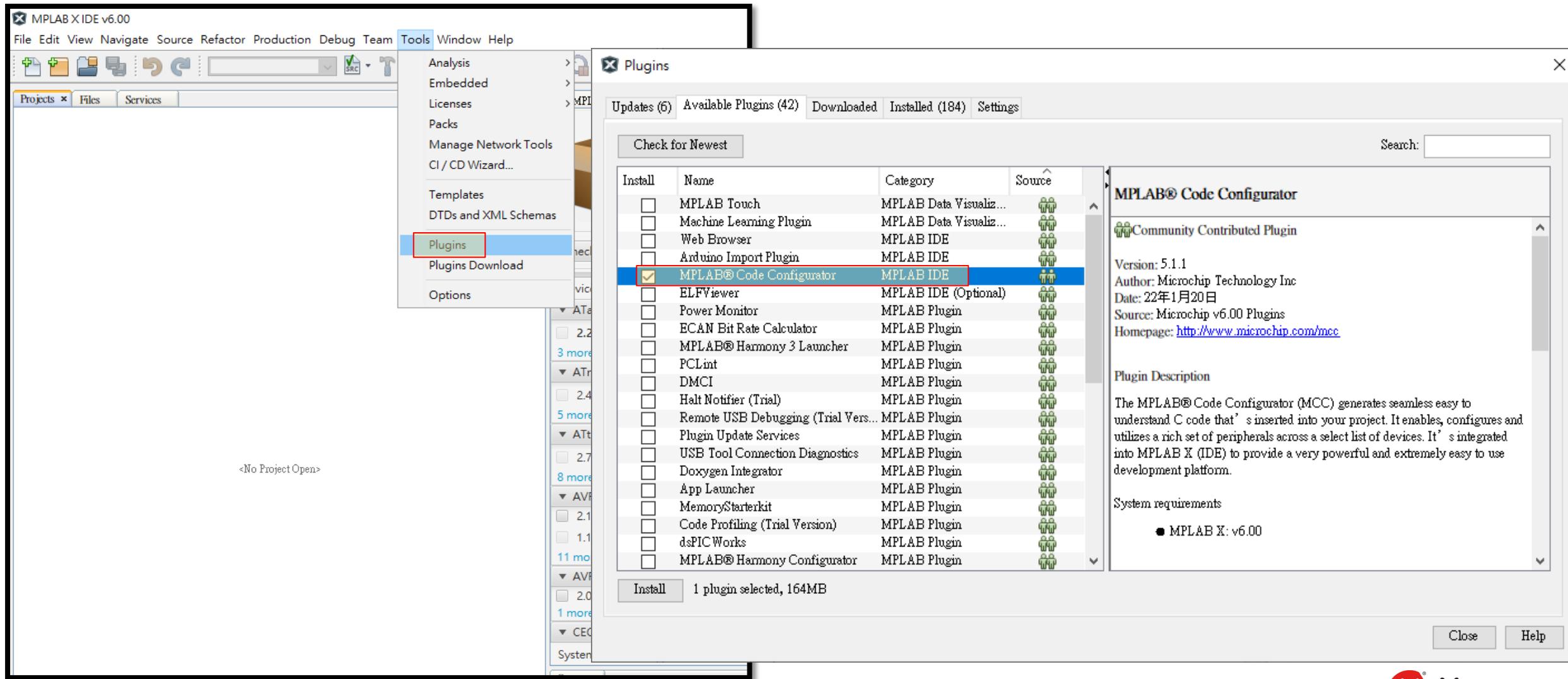
Installing MCC

MPLAB® X IDE / Tools / Plugins



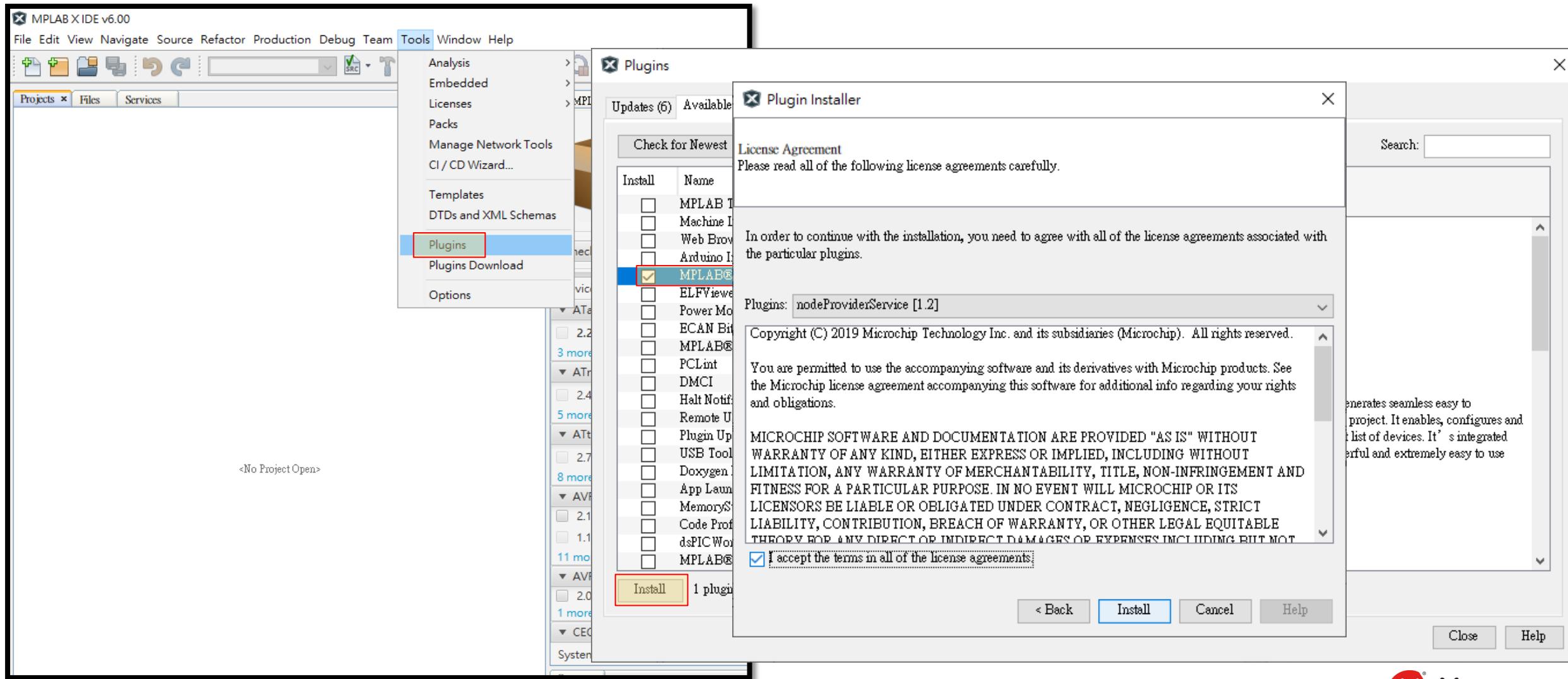
Installing MCC

MPLAB® X IDE / Tools / Plugins



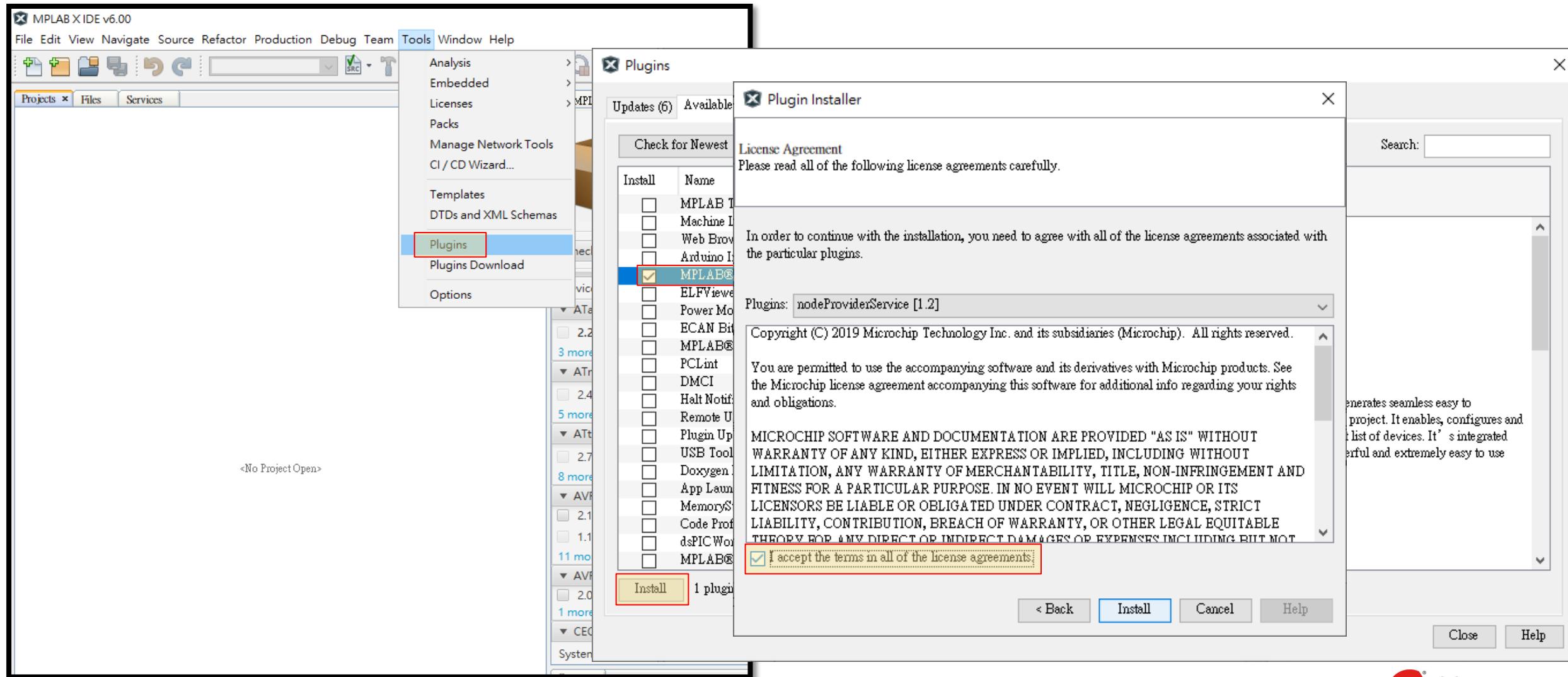
Installing MCC

MPLAB® X IDE / Tools / Plugins



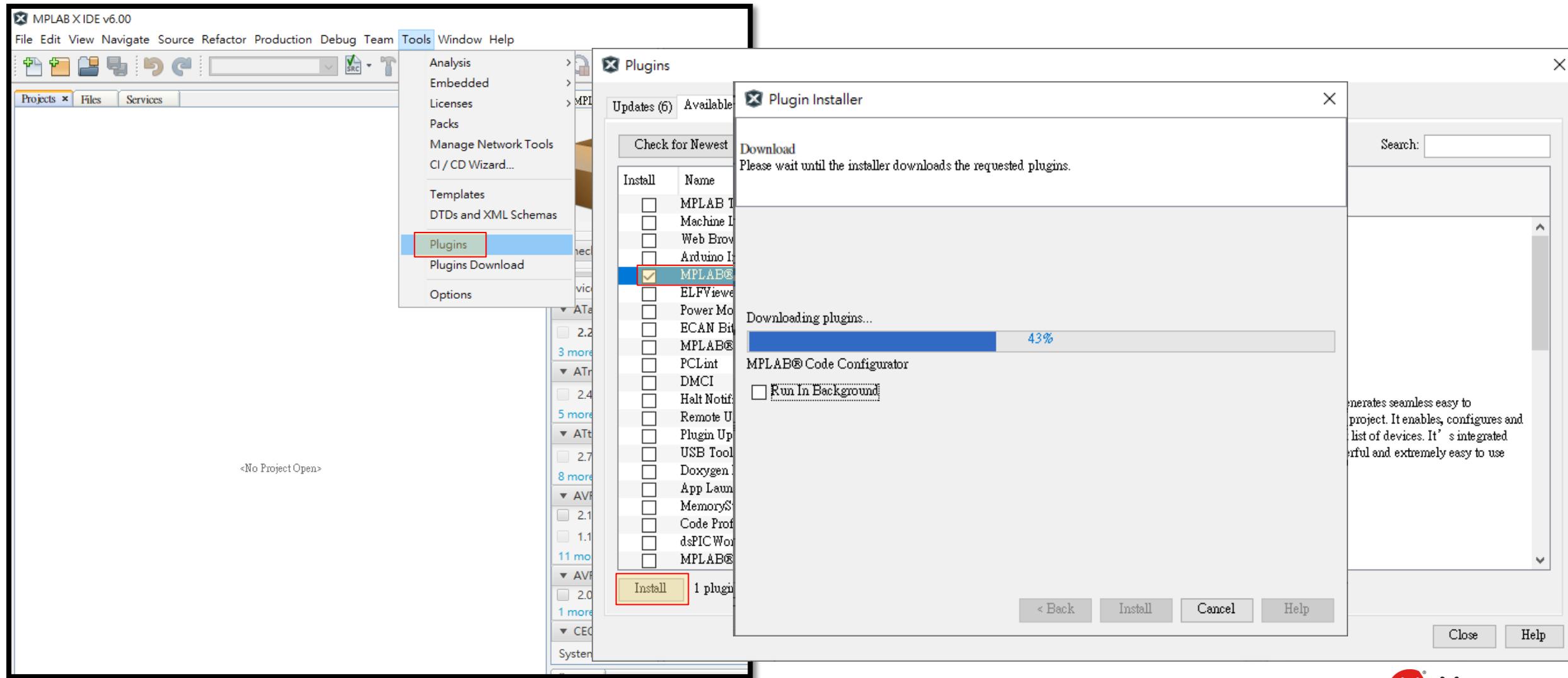
Installing MCC

MPLAB® X IDE / Tools / Plugins



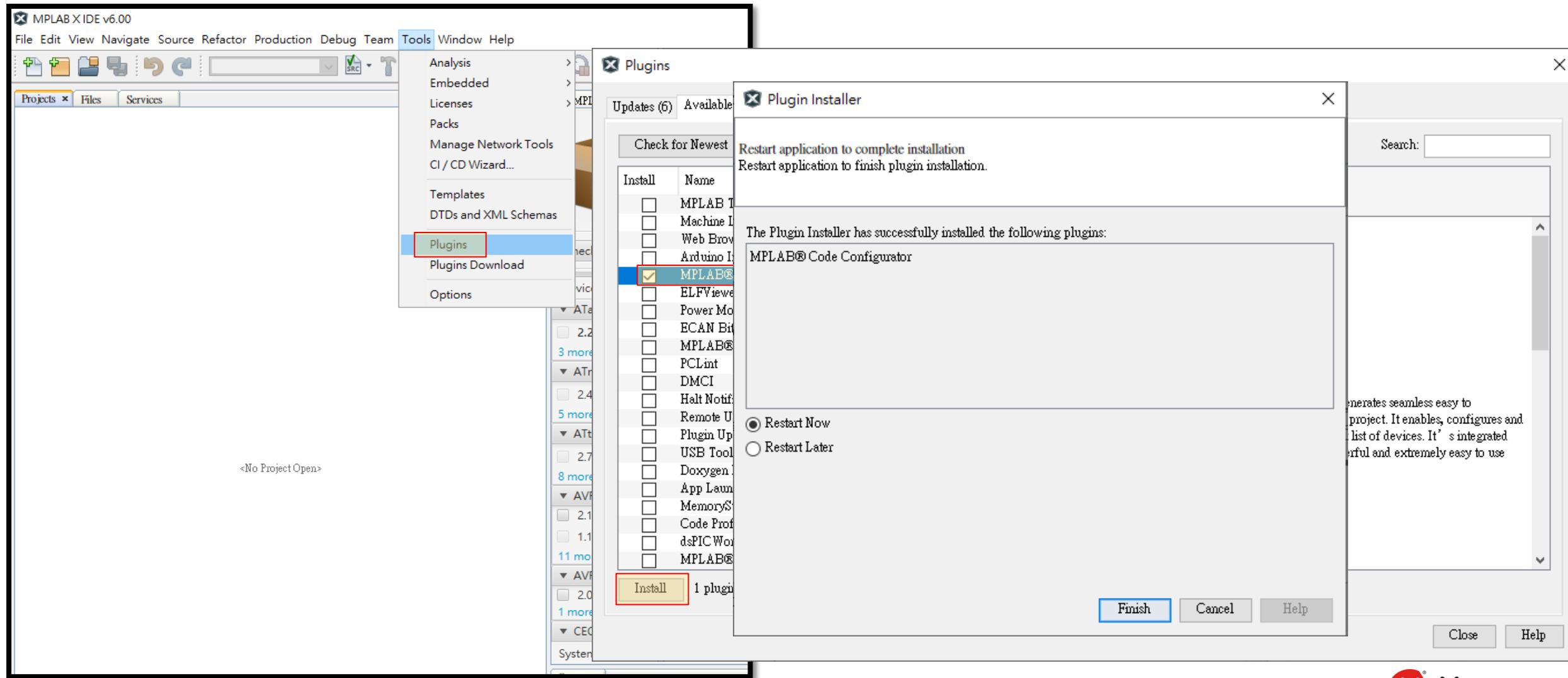
Installing MCC

MPLAB® X IDE / Tools / Plugins



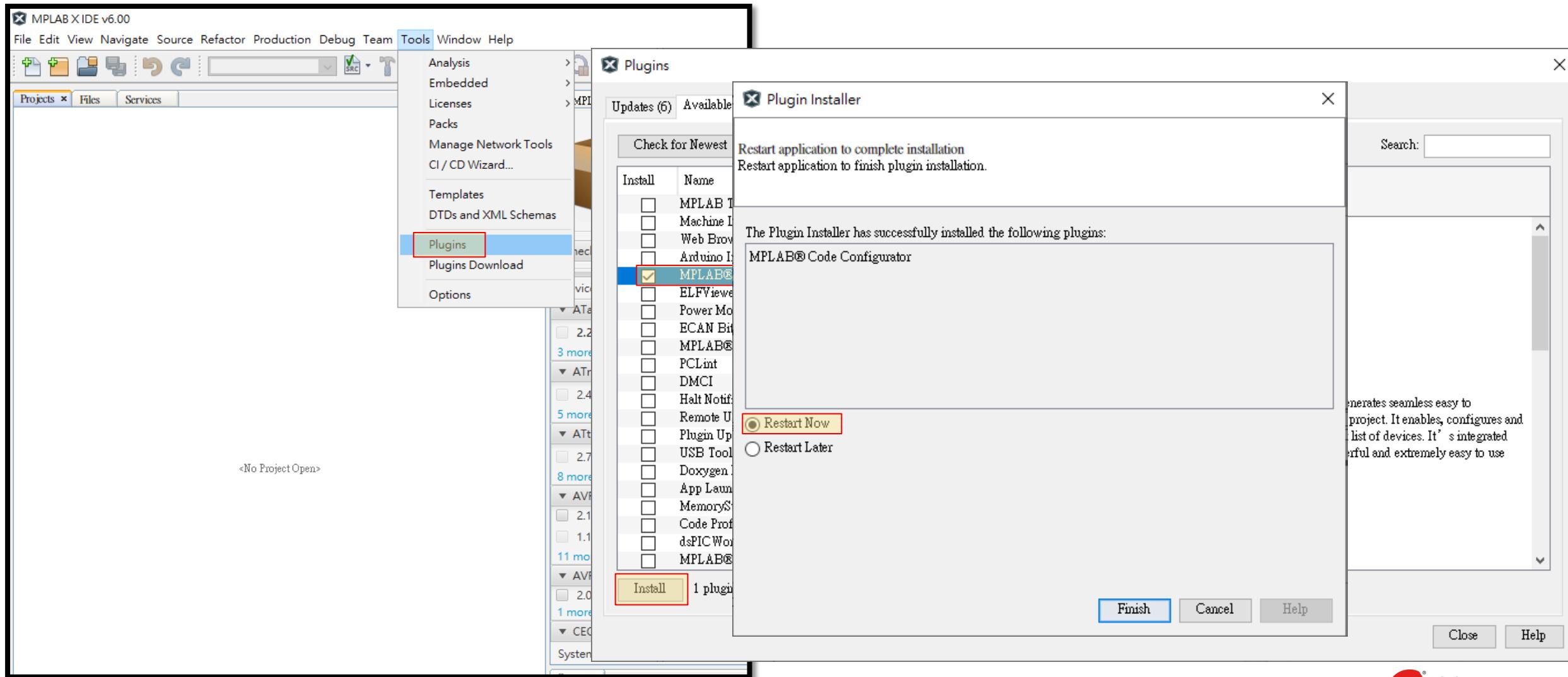
Installing MCC

MPLAB® X IDE / Tools / Plugins



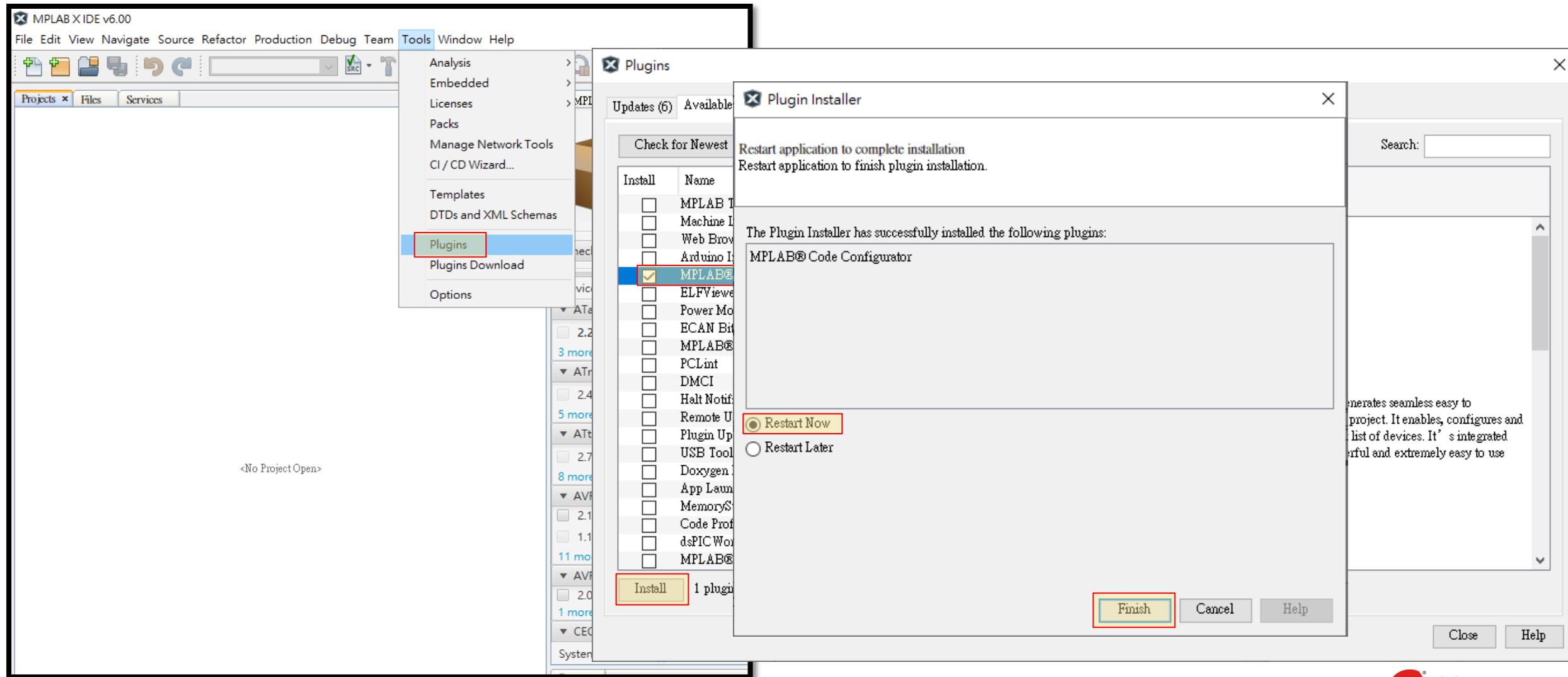
Installing MCC

MPLAB® X IDE / Tools / Plugins



Installing MCC

MPLAB® X IDE / Tools / Plugins



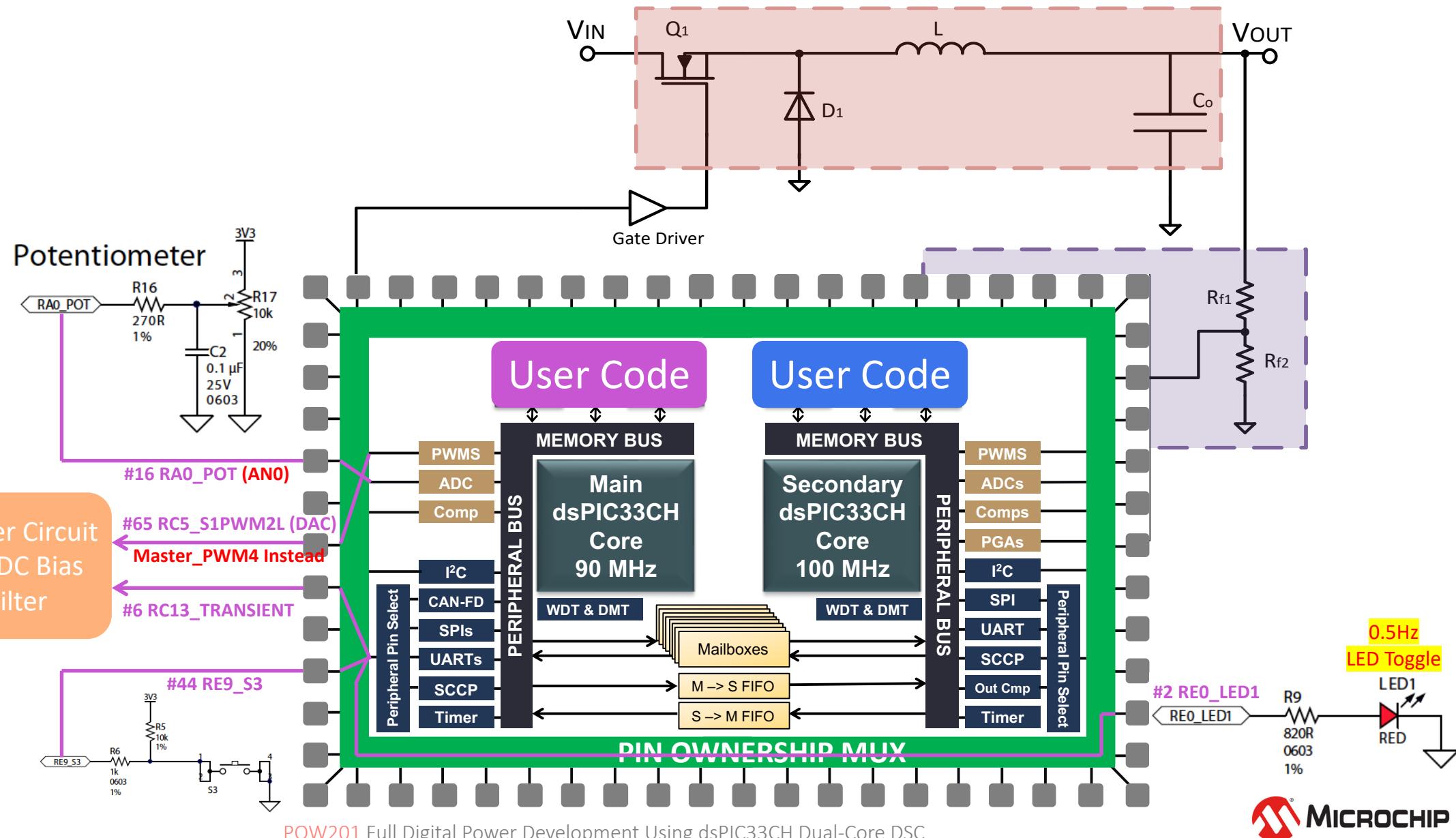
MCC + OS Scheduler for Main Core

Lab1

Reference manual on  **MICROCHIP Developer Help**
<https://microchipdeveloper.com/mcc:mplab-code-configure-support-for-dual-core-devices>

Lab #1: MCC + OS Scheduler for Main Core

Main Core
Secondary Core



Lab #1: MCC + OS Scheduler for Main Core

Main Core

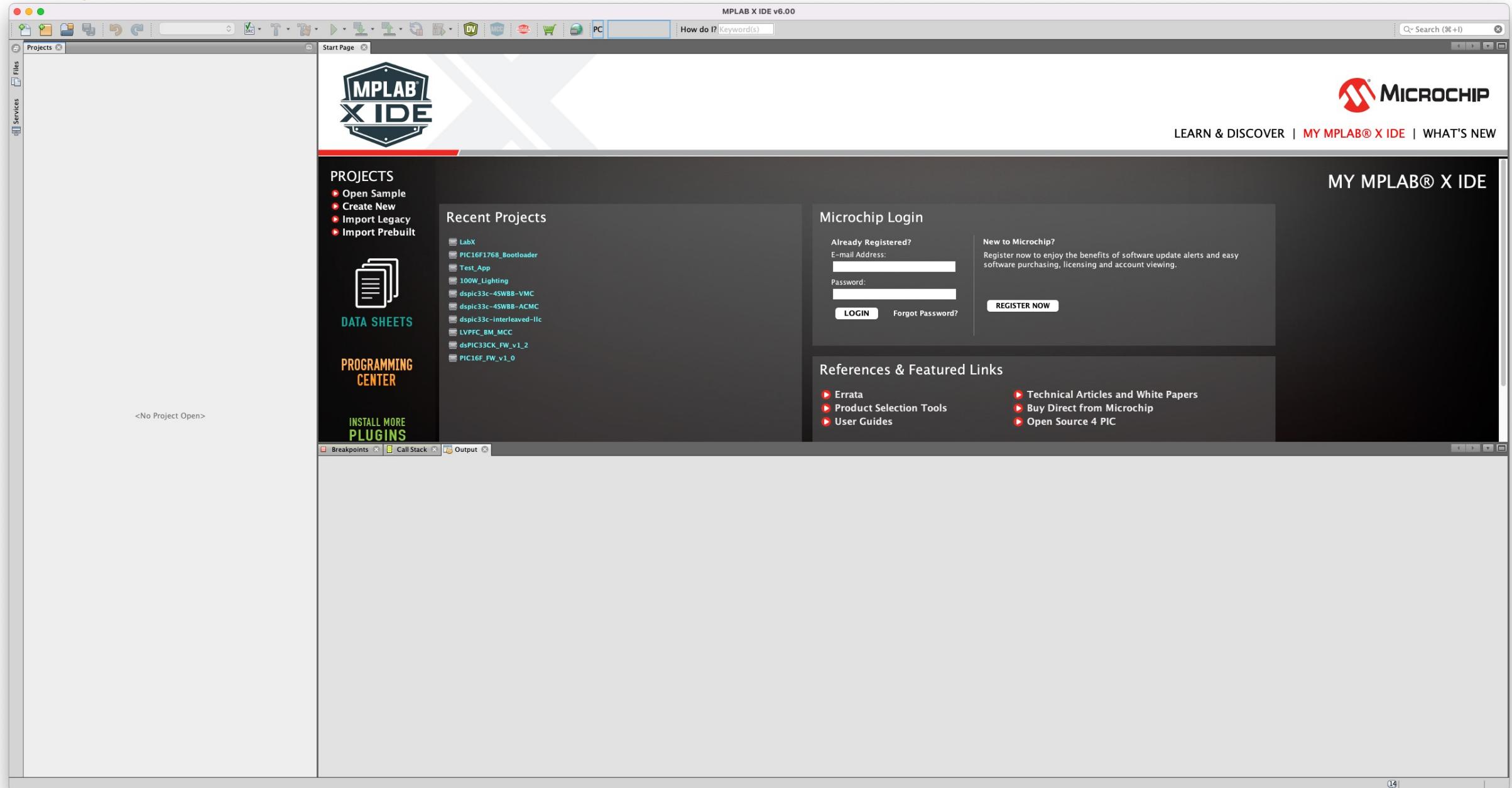
Secondary Core

User Code (90MIPS)

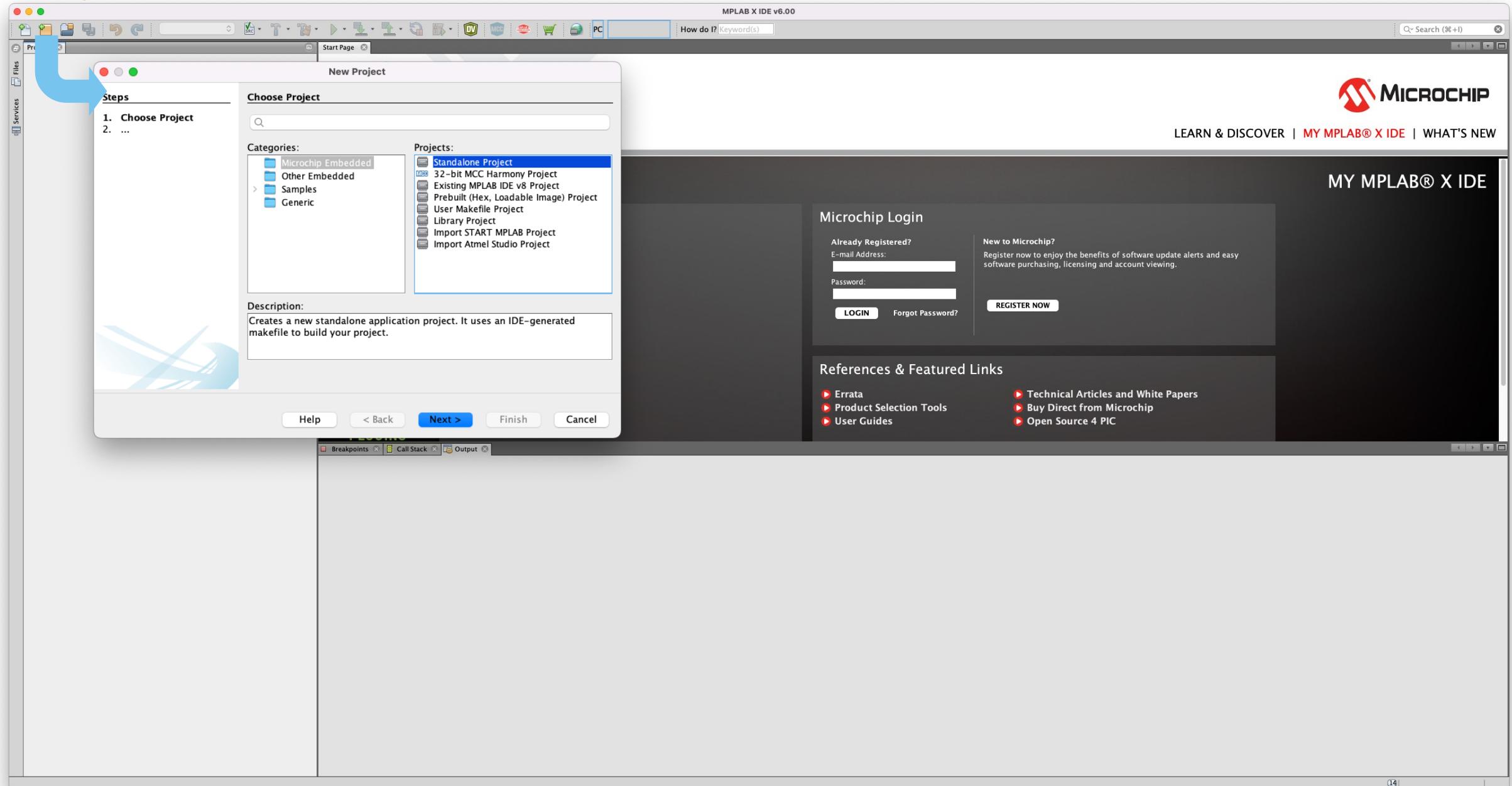
- **Main loop:**
 - OS Scheduler
 - 0.5Hz Toggle LED1
 - Check S3 to trigger RC13_TRANSIENT
- **ISR:**
 - 1ms Timer1 for OS Scheduler
 - AN0-POT to set RC5_PWM4
 - Max 50%
- **Peripheral:**
 - PWM4H – ~152KHz (Initial 0% Duty Cycle)
 - I/O: RE9-In, RC13-Out, RE0-Out
 - Timer1 with 1ms ISR
 - AN0 with ISR (IP=1)
 - Triggered by PWM2_Trigger1

User Code (100MIPS)

New Project as Master on Main core



New Project as Master on Main core



LEARN & DISCOVER | MY MPLAB® X IDE | WHAT'S NEW



MY MPLAB® X IDE

Microchip Login

Already Registered?

E-mail Address:

Password:

LOGIN

[Forgot Password?](#)

New to Microchip?

Register now to enjoy the benefits of software update alerts and easy software purchasing, licensing and account viewing.

REGISTER NOW

References & Featured Links

▶ Errata

▶ Product Selection Tools

▶ User Guides

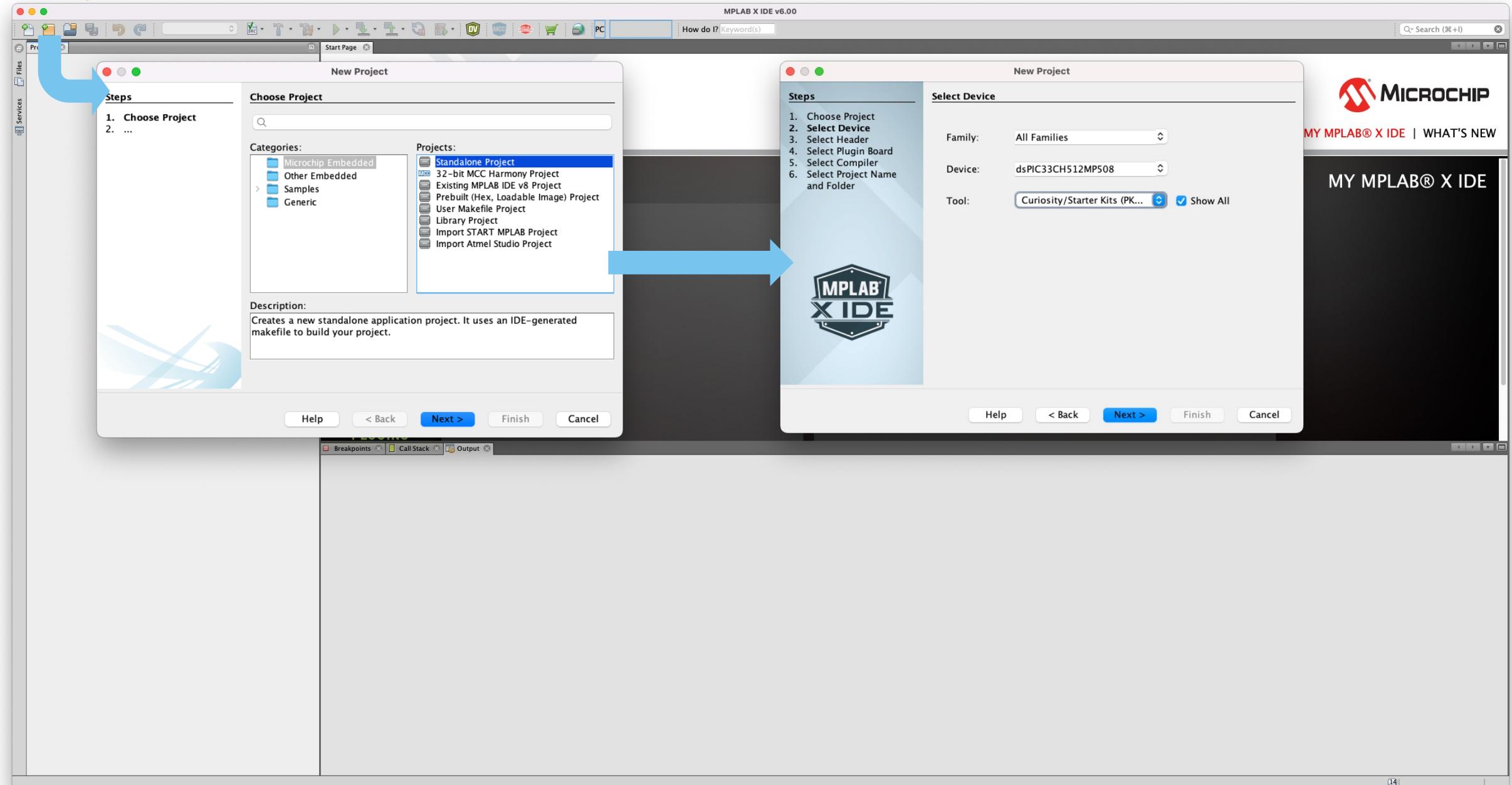
▶ Technical Articles and White Papers

▶ Buy Direct from Microchip

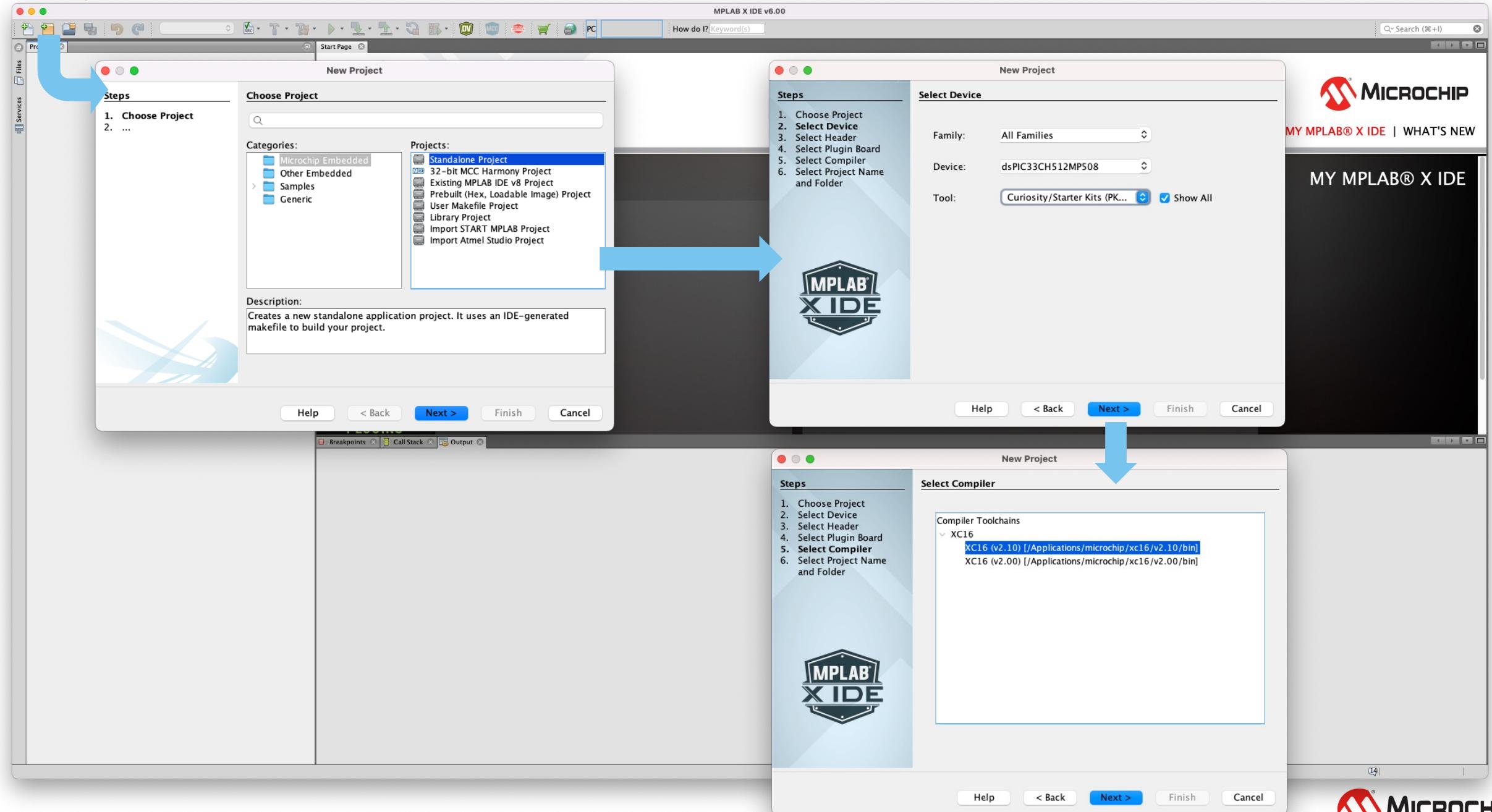
▶ Open Source 4 PIC



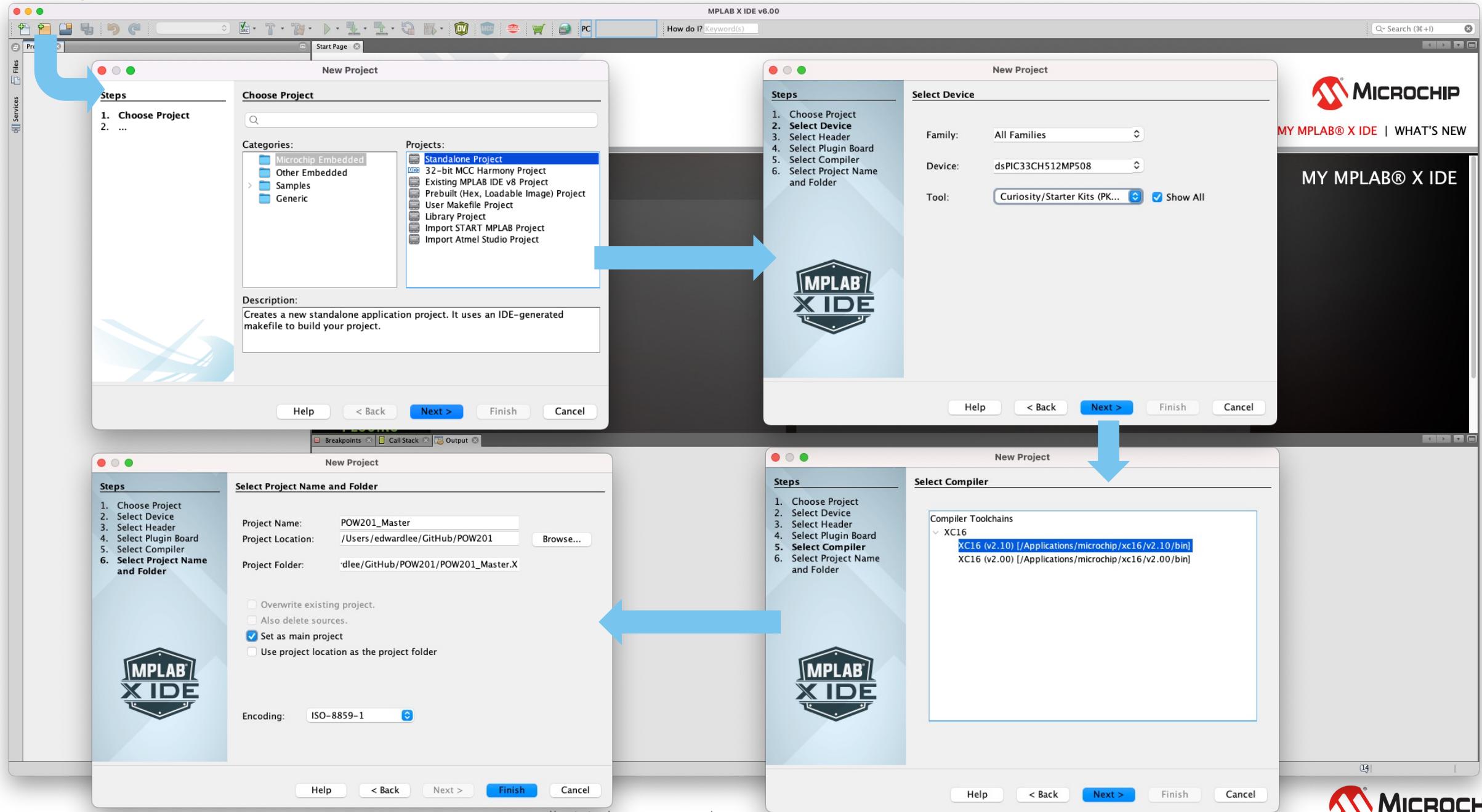
New Project as Master on Main core



New Project as Master on Main core



New Project as Master on Main core



Projects

- POW201_Master**
 - Header Files
 - Important Files
 - Linker Files
 - Source Files
 - Libraries
 - Loadables
 - Secondaries

POW201_Master - Dashboard

Navigator

- POW201_Master**
 - Project Type: Application - Configuration: default
 - Device**
 - dsPIC33CH512MP508
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable
 - Packs**
 - dsPIC33CH-MP_DFP (1.12.352)
 - Compiler Toolchain**
 - XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
 - Production Image: Optimization: gcc 0
 - Device support information: dsPIC33CH-MP_DFP (1.12.352)
 - Memory**
 - Usage Symbols disabled. Click to enable Load Symbols.
 - Data 49,152 (0xC000) bytes
 - Program 0 (0x0) words
 - Stack Usage Guidance
 - Debug Tool**
 - : None (VID, PID)
 - Debug Resources**
 - Program BP Used: 0 Free: 0
 - Data BP: No Support
 - Data Capture BP: No Support
 - Unlimited BP (S/W): No Support

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody

Supports the MCC Builder
Supports content versioning at driver level
An iteration of MCC Generated Code
Works both on- and off-line

Select MCC Melody

[Release notes and supported devices](#)

MCC Classic

Development process you are accustomed to
All components and libraries that you have used before

Select MCC Classic

[Release notes and supported devices](#)

MPLAB® Harmony

Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

Select MPLAB Harmony

[Release notes and supported devices](#)

Library support may be a key factor in your choice of MCC flavor:

> MCC Melody and MCC Classic - Library Summary

> MPLAB Harmony - Library Summary

Still unsure which content type is right for your project?

See More Details

Output

Configuration Loading Error MPLAB® Code Configurator

```

16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.

```



POW201_Master

- Header Files
- Important Files
- Linker Files
- Source Files
- Libraries
- Loadables
- Secondaries

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody ?

Supports the MCC Builder
Supports content versioning at driver level
An iteration of MCC Generated Code
Works both on- and off-line

Select MCC Melody

[Release notes and supported devices](#)

MCC Classic ?

Development process you are accustomed to
All components and libraries that you have used before

Select MCC Classic

[Release notes and supported devices](#)

MPLAB® Harmony ?

Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

Select MPLAB Harmony

[Release notes and supported devices](#)

POW201_Master - Dashboard

Navigator

Project Type: Application - Configuration: default

- Device
 - dsPIC33CH512MP508
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable
- Packs
 - dsPIC33CH-MP_DFP (1.12.352)
- Compiler Toolchain
 - XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
 - Production Image: Optimization: gcc 0
 - Device support information: dsPIC33CH-MP_DFP (1.12.352)
- Memory
 - Usage Symbols disabled. Click to enable Load Symbols.
 - Data 49,152 (0xC000) bytes
 - Program 0 (0x0) words
 - Stack Usage Guidance
- Debug Tool
 - : None (VID, PID)
- Debug Resources
 - Program BP Used: 0 Free: 0
 - Data BP: No Support
 - Data Capture BP: No Support
 - Unlimited BP (S/W): No Support

Still unsure which content type is right for your project?

See More Details

Output

Configuration Loading Error MPLAB® Code Configurator

```
16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.
```

MCC Drivers

default Start Page MCC Content Manager Wizard PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s)

Import

Projects **Files**

POW201_Master

- Header Files
- Important Files
- Linker Files
- Source Files
- Libraries
- Loadables
- Secondaries

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody Supports the MCC Builder
Supports content versioning at driver level
An iteration of MCC Generated Code
Works both on- and off-line

MCC Classic Development process you are accustomed to
All components and libraries that you have used before

MPLAB® Harmony Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

Release notes and supported devices

Start Page Content Manager

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Required Content

All required content is available locally on your machine. No other download is needed to get started.
To change content versions later, access the Content Manager from Device Resources.

Optional Content

Select optional content to be made available in Device Resources for selection

Optional Content

Component	Version	Description
<input type="checkbox"/> Libraries		
<input type="checkbox"/> SD/MMC Card	1.1.0	Driver for SD/MMC cards allowing direct sector read/write operations. Most applications will want to use file system on top of this driver, such as FatFs, to read data written by a PC or to write data that is readable by the PC.

Output Configuration Loading Error MPLAB® Code Configurator

```
16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.
```

MCC Drivers

default

Start Page MPLAB X Store MCC Content Manager

PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s) Search (%+)

Import ?

POW201_Master

- Header Files
- Important Files
- Linker Files
- Source Files
- Libraries
- Loadables
- Secondaries

Projects Files Resource Management [MCC] Start Page MPLAB X Store Pin Module System Module Interrupt Module

MCC v5.3.7

Project Resources Generate Import Export ?

System

- Interrupt Module
- Pin Module
- System Module

Device Resources Content Manager

Documents dsPIC33CH512MP508 Product Page

Libraries

- 16-bit Bootloader
- CHARACTER LCD
- CryptoAuthLibrary
- DacLibrary
- FatFs
- Foundation Services
- LED
- LED_BLUE

POW201_Master - Dashboard Navigator

POW201_Master Project Type: Application - Configuration: default

Device

- dsPIC33CH512MP508
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable

Packs

- dsPIC33CH-MP_DFP (1.12.352)

Compiler Toolchain

- XC16 (v2.10) [/Applications/microchip/xc16/v2.10/t]
- Production Image: Optimization: gcc 0
- Device support information: dsPIC33CH-MP_DFP (1.12.352)

Memory

- Usage Symbols disabled. Click to enable Load Symbol
- Data 49,152 (0xC000) bytes
- Program 0 (0x0) words
- Stack Usage Guidance

Debug Tool

- : None (VID , PID)

Debug Resources

- Program BP Used: 0 Free: 0
- Data BP: No Support
- Data Capture BP: No Support
- Unlimited BP (S/W): No Support

Program BP Used: 0 Free: 0
Data BP: No Support
Data Capture BP: No Support
Unlimited BP (S/W): No Support

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody MCC Classic

You are accustomed to series that you have used

MCC Classic

and supported devices

Wizard

Finish

download is needed to get started.

agger from Device Resources.

Resources for selection

Description

MPLAB® Harmony

Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

Select MPLAB Harmony

Release notes and supported devices

Configuration Loading Error MPLAB® Code Configurator

16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.

MCC Drivers – Clock@90 MIPS

MCC v5.3.7

Project Resources Generate Import... Export ?

System

- Interrupt Module
- Pin Module
- System Module

Device Resources Content Manager

- Documents dsPIC33CH512MP508 Product Page
- Libraries
 - 16-bit Bootloader
 - CHARACTER LCD
 - CryptoAuthLibrary
 - DacLibrary
 - FatFs
 - Foundation Services
 - LED
 - LED_BLUE
 - LED_GREEN

POW201_Master – Dashboard Navigator

- POW201_Master Project Type: Application – Configuration: default
- Device dsPIC33CH512MP508 Checksum: Blank, no code loaded CRC32: Hex file unavailable
- Packs dsPIC33CH-MP_DFP (1.12.352)
- Compiler Toolchain XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin] Production Image: Optimization: gcc 0 Device support information: dsPIC33CH-MP_DFP (1.12)
- Memory Usage Symbols disabled. Click to enable Load Symbols.
- Stack Usage Guidance
- Debug Tool None (VID, PID)
- Debug Resources Program BP Used: 0 Free: 0 Data BP: No Support Data Capture BP: No Support Unlimited RP (S/W): No Support

Easy Setup Registers

Clock

- 8000000 Hz FRC Oscillator (8.0 MHz) Clock Source
 - FRC Postscaler
- PLL Enable
 - 8 MHz Prescaler 1:1
 - 1440 MHz FeedBack 1:180
 - 720 MHz Postscaler1 1:2
 - 360 MHz Postscaler2 1:2
- 180 MHz Fosc
- 90 MHz Fosc/2
- Auxiliary Clock
 - 8000000 Hz FRC Clock Source
 - 8 MHz PLL Enable Prescaler 1:1
 - 1000 MHz Feedback 1:125
 - 500 MHz Postscaler1 1:2
 - 500 MHz Postscaler2 1:1
- VCO & AVCO
 - 360 MHz VCO Divider FVCO/4
 - 500 MHz AVCO Divider FVCO/2

Pin Manager: Grid View

Pin Manager: Package View

6.1 Primary PLL

The Primary Oscillator and internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. There are two independent instantiations of PLL for the Master and Slave clock subsystems. Figure 6-4 illustrates a block diagram of the Master/Slave core PLL module.

For PLL operation, the following requirements must be met at all times without exception:

- The PLL Input Frequency (FPLL) must be in the range of 8 MHz to 64 MHz
- The PFD Input Frequency (FPPD) must be in the range of 8 MHz to (Fvco/16) MHz

The VCO Output Frequency (Fvco) must be in the range of 400 MHz to 1600 MHz

FIGURE 6-4: MASTER/SLAVE CORE PLL AND VCO DETAIL

Note 1: From Master and Slave core shared oscillator source.
2: Clock option for PWM.
3: Clock option for ADC.
4: Clock option for DAC.

MCC Drivers – Clock@90 MIPS

MCC v5.3.7

Project Resources Generate Import... Export ?

System

- Interrupt Module
- Pin Module
- System Module

Device Resources Content Manager

Documents dsPIC33CH512MP508 Product Page

Libraries

- 16-bit Bootloader
- CHARACTER LCD
- CryptoAuthLibrary
- DacLibrary
- FatFs
- Foundation Services
- LED
- LED_BLUE
- LED_GREEN

POW201_Master – Dashboard Navigator

Project Type: Application – Configuration: default

Device dsPIC33CH512MP508

Packs dsPIC33CH-MP_DFP (1.12.352)

Compiler Toolchain XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]

Production Image: Optimization: gcc 0

Device support information: dsPIC33CH-MP_DFP (1.12)

Memory

- Usage Symbols disabled. Click to enable Load Symbols.
- Data 49,152 (0xC000) bytes
- Program 0 (0x0) words
- Stack Usage Guidance

Debug Tool

- : None (VID, PID)

Debug Resources

- Program BP Used: 0 Free: 0
- Data BP: No Support
- Data Capture BP: No Support
- Unlimited RP (S/W): No Support

Start Page MPLAB X Store Pin Module System Module Interrupt Module

PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s) Search (%+)

Pin Manager: Package View

Clock

8000000 Hz FRC Oscillator (8.0 MHz) Clock Source

FRC Postscaler

PLL Enable

8 MHz	1:1 Prescaler
1440 MHz	1:180 FeedBack
720 MHz	1:2 Postscaler1
360 MHz	1:2 Postscaler2

180 MHz Fosc

90 MHz Fosc/2 **90MIPS**

Auxiliary Clock

8000000 Hz FRC Clock Source

PLL Enable

8 MHz	1:1 Prescaler
1000 MHz	1:125 Feedback
500 MHz	1:2 Postscaler1
500 MHz	1:1 Postscaler2

VCO & AVCO

360 MHz	FVCO/4 VCO Divider
500 MHz	FVCO/2 AVCO Divider

Clock Output Pin Configuration OSC2 is general purpose digital I/O pin

Reference Oscillator Output

CAN FD Clock Generator

Pin Manager: Grid View

Notifications [MCC]

Pin Layout

RB13 RB15 RB12 RB14 RB11 RB10 RB9 RD1 RD2 RD3 RD4 RC11 RC10 RC5 RC4 RE12 RB9
 ● 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61
 RE14 1
 RE0 2
 RB15 3
 RE1 4
 RC12 5
 RC13 6
 RC14 7
 RC15 8
 NMCLR 9
 RD15 10

MICROCHIP

6.1 Primary PLL

The Primary Oscillator and internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. There are two independent instantiations of PLL for the Master and Slave clock subsystems. Figure 6-4 illustrates a block diagram of the Master/Slave core PLL module.

For PLL operation, the following requirements must be met at all times without exception:

- The PLL Input Frequency (FPLL) must be in the range of 8 MHz to 64 MHz
- The PFD Input Frequency (FPFD) must be in the range of 8 MHz to (Fvco/16) MHz

The VCO Output Frequency (Fvco) must be in the range of 400 MHz to 1600 MHz

FIGURE 6-4: MASTER/SLAVE CORE PLL AND VCO DETAIL

Note 1: From Master and Slave core shared oscillator source.
2: Clock option for PWM.
3: Clock option for ADC.
4: Clock option for DAC.

MCC Drivers – Clock@90 MIPS

MCC v5.3.7

Project Resources Generate Import... Export ?

System

- Interrupt Module
- Pin Module
- System Module

Device Resources Content Manager

Documents dsPIC33CH512MP508 Product Page

Libraries

- 16-bit Bootloader
- CHARACTER LCD
- CryptoAuthLibrary
- DacLibrary
- FatFs
- Foundation Services
- LED
- LED_BLUE
- LED_GREEN

POW201_Master – Dashboard Navigator

Project Type: Application – Configuration: default

Device dsPIC33CH512MP508

Packs dsPIC33CH-MP_DFP (1.12.352)

Compiler Toolchain XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]

Production Image: Optimization: gcc 0

Device support information: dsPIC33CH-MP_DFP (1.12)

Memory

- Usage Symbols disabled. Click to enable Load Symbols.
- Data 49,152 (0xC000) bytes
- Program 0 (0x0) words
- Stack Usage Guidance

Debug Tool

- : None (VID, PID)

Debug Resources

- Program BP Used: 0 Free: 0
- Data BP: No Support
- Data Capture BP: No Support
- Unlimited RP (S/W): No Support

Start Page MPLAB X Store Pin Module System Module Interrupt Module

PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s) Search (%+)

Pin Manager: Package View

Easy Setup Registers

Clock

8000000 Hz FRC Oscillator (8.0 MHz) Clock Source

FRC Postscaler

PLL Enable

8 MHz	1:1 Prescaler
1440 MHz	1:180 FeedBack
720 MHz	1:2 Postscaler1
360 MHz	1:2 Postscaler2

180 MHz Fosc

90 MHz Fosc/2 **90MIPS**

Auxiliary Clock

8000000 Hz FRC Clock Source

PLL Enable

8 MHz	1:1 Prescaler
1000 MHz	1:125 Feedback
500 MHz	1:2 Postscaler1
500 MHz	1:1 Postscaler2

VCO & AVCO

360 MHz	FVCO/4 VCO Divider
500 MHz	FVCO/2 AVCO Divider

Clock Output Pin Configuration OSC2 is general purpose digital I/O pin

Reference Oscillator Output

CAN FD Clock Generator

Pin Manager: Grid View

Notifications [MCC]

Microchip Pinout Diagram

6.1 Primary PLL

The Primary Oscillator and internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. There are two independent instantiations of PLL for the Master and Slave clock subsystems. Figure 6-4 illustrates a block diagram of the Master/Slave core PLL module.

For PLL operation, the following requirements must be met at all times without exception:

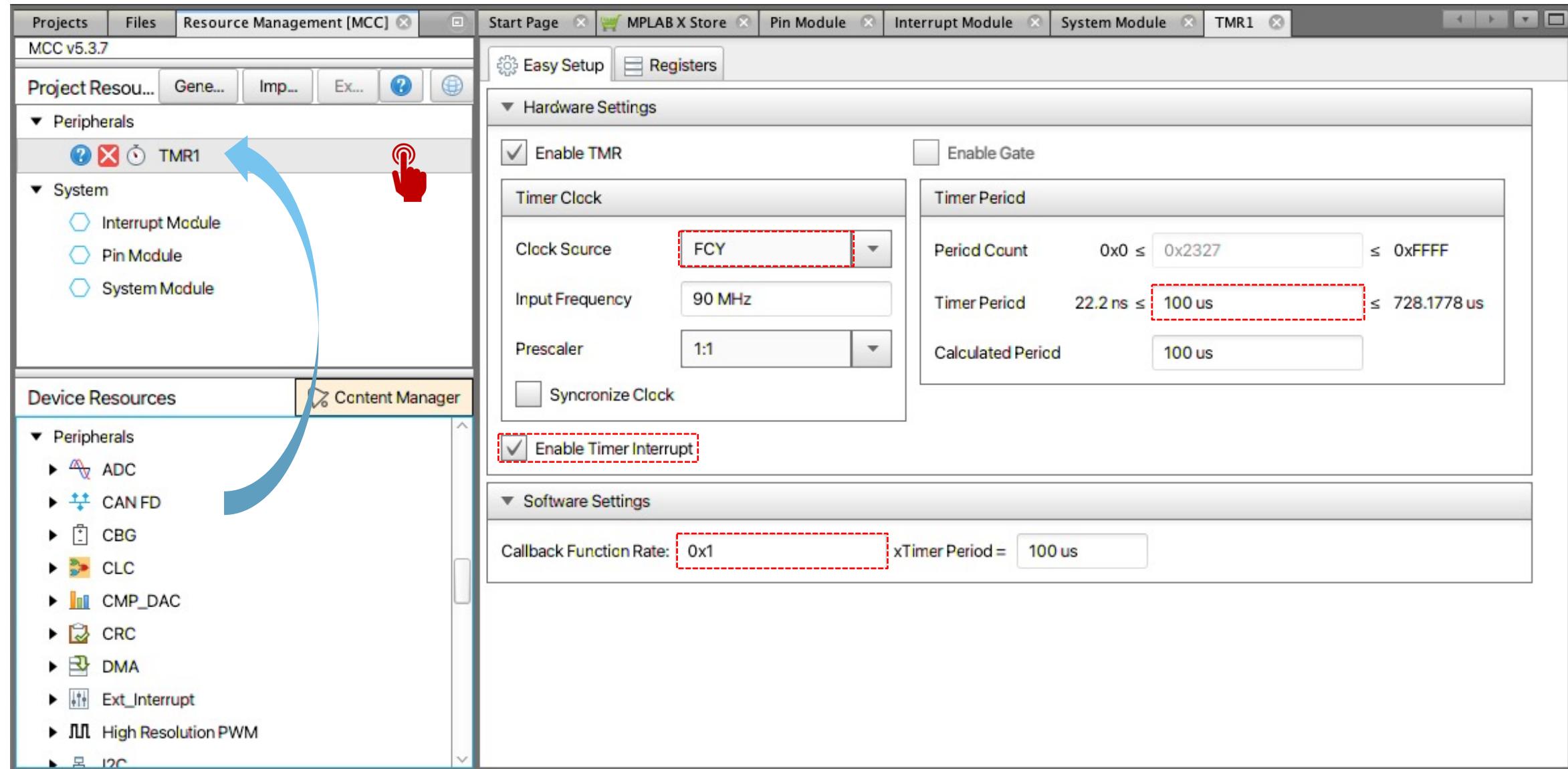
- The PLL Input Frequency (FPLL) must be in the range of 8 MHz to 64 MHz
- The PFD Input Frequency (FPPD) must be in the range of 8 MHz to (Fvco/16) MHz

The VCO Output Frequency (Fvco) must be in the range of 400 MHz to 1600 MHz

FIGURE 6-4: MASTER/SLAVE CORE PLL AND VCO DETAIL

Note 1: From Master and Slave core shared oscillator source.
Note 2: Clock option for PWM.
Note 3: Clock option for ADC.
Note 4: Clock option for DAC.

MCC Drivers – Timer1 (100 us)



MCC Drivers – PWM4H – ~152KHz (Initial 0% Duty Cycle)

The screenshot shows the MCC v5.3.7 software interface with the following details:

- Project Resources** tab is selected.
- Peripherals** section is expanded, showing **PWM** and **TMR1**.
- Device Resources** sidebar lists various peripherals: Turnkey Touch, WINC15XX, X2C, ADC, CAN FD, CBG, CLC, CMP_DAC, CRC, DMA, and Ethernet.
- Content Manager** icon is visible in the Device Resources sidebar.
- PWM Clock Settings** panel:
 - Master Clock Selection: AFVCO/2 - Auxiliary VCO divided by 2
 - Master Clock Frequency: 500 MHz
 - Clock Divided Frequency: 250000000 Hz
- Select Required PWM Generators** panel:
 - PWM Generator 4** is selected (highlighted with a red dashed box).
 - PWM Generator 1**, **PWM Generator 2**, **PWM Generator 3**, and **PWM Generator 4** are listed.
 - Enable PWM Generator
 - Enable High Resolution
- PWM Operation Mode**: Independent Edge
- PWM Output Mode**: Complementary
- PWM Frequency Settings**, **Trigger Control Settings**, **Dead Time and Override Settings**, and **Data Update Settings** are shown as expandable sections.

MCC Drivers – PWM4H – ~152KHz (Initial 0% Duty Cycle)

The screenshot shows the MCC PWM configuration interface for PWM Generator 4. Key settings include:

- PWM Generator**: PWM Generator 4
- Custom Name**: PWM_GENERATOR_4
- Enable PWM Generator**: Checked
- Enable High Resolution**: Checked
- PWM Operation Mode**: Independent Edge
- PWM Output Mode**: Independent
- PWM Input Clock Selection**: 500000000 Hz
- Period**: Requested Frequency: 61.02864 kHz ≤ 152 kHz ≤ 29.41176471 MHz. Calculated Frequency: 151.99878 kHz. Requested Period: 34 ns ≤ 6.579 us ≤ 16.3858 us. Calculated Period: 6.579 us.
- Duty Cycle**: PWM Duty Cycle: 0 % ≤ 0 ≤ 100 %
- Data Update Settings**: Update Trigger: Duty Cycle. Update Mode: SOC update.

The screenshot shows the Trigger Control Settings for PWM Start of Cycle Control. The ADC Trigger 1 setting is highlighted with a red box and set to "Trigger A Compare". Other settings include:

- Start of Cycle Trigger**: Self-trigger
- Trigger Output Selection**: EOC event
- ADC Trigger 1**: Trigger A Compare
- ADC Trigger 2**: None
- Trigger A Compare**: 0 ns ≤ 0 ns ≤ 16.3838 us
- Trigger B Compare**: 0 ns ≤ 0 ns ≤ 16.3838 us
- Trigger C Compare**: 0 ns ≤ 0 ns ≤ 16.3838 us

The screenshot shows the Registers configuration tool for the PG4IOPCONH register. The PENL field is highlighted with a red box and set to "disabled". Other fields include:

- TSYNCDIS**: PWM EOC
- Register: PG4IOPCONH**: 0x18
- CAPSRC**: Software
- DTCMPSEL**: PCI Sync Logic
- PENH**: enabled
- PENL**: disabled
- PMOD**: Independent
- POLH**: Active-high
- POLL**: Active-high
- Register: PG4IOPCONL**: 0x0
- CLDAT**: 0x0
- CLMOD**: disabled

Released PWM4L as I/O

MCC Drivers – ADC/AN0 for POT

The screenshot shows the Microchip MCC (Microcontroller Configuration Expert) software interface. The top navigation bar includes 'Projects', 'Files', 'Services', 'Classes', 'Resource Management [MCC]', and other standard options.

Project Resources tab is active, displaying a list of peripherals and system modules:

- Peripherals:** ADC1 (selected), PWM, TMR1.
- System:** Interrupt Module, Pin Module, System Module.

A large blue arrow points from the 'ADC1' entry in the Project Resources list to the 'ADC1' configuration window on the right.

ADC1 Configuration Window:

- Hardware Settings:** 'Enable ADC' is checked. Under 'ADC Clock', the 'Conversion Clock Source' is set to 'PLL VCO/4' (highlighted with a red box).
- Selected Channels:** A table lists 16 channels. The first row, corresponding to the highlighted ADC1 entry, is highlighted with a red box and shows:

Core	Enable	Ccore Channel	Pin Name	Custom Name	Trigger Source	Compare	Interrupt
Shared	<input checked="" type="checkbox"/>	AN0	RA0	channel_AN0	Master PWM...	None	<input checked="" type="checkbox"/>

The 'Trigger Source' field for this row contains the text 'Master PWM4 Trigger1' (highlighted with a red box).

ADC1

Registers

Register: ADCON2L 0x0

- EIEN: disabled
- PTGEN: disabled
- REFCIE: disabled
- REFERCIE: disabled
- SHRADCS:** 2
- SHREISEL: Early interrupt is generated 1 TADCORE clock prior to data being ready

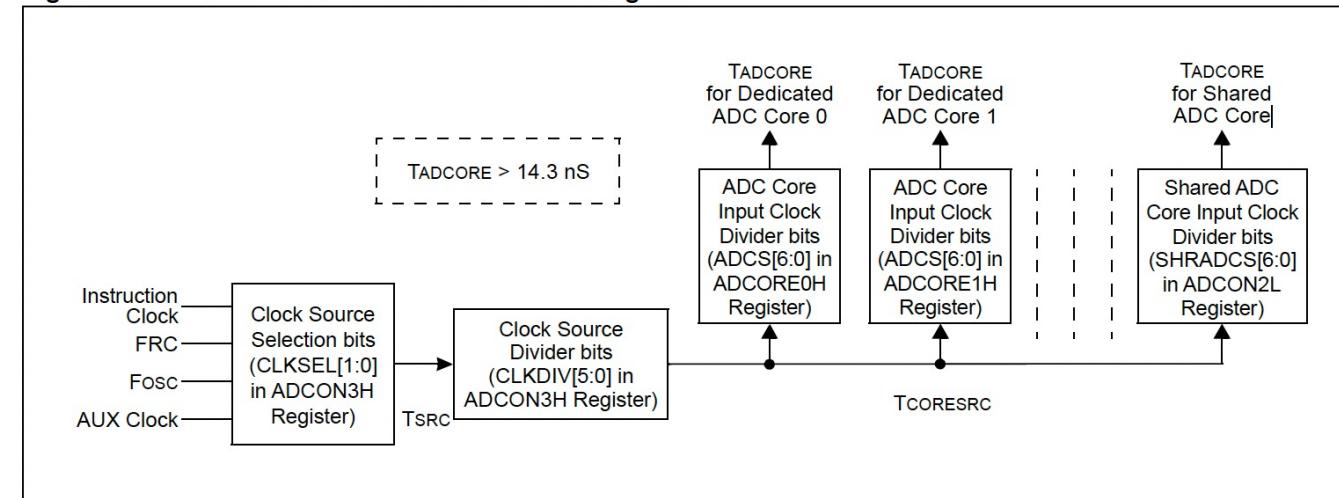
Register: ADCON3H 0xC280

- CLKDIV:** 3
- CLKSEL: PLL VCO/4
- SHREN: enabled

Master - ADC Shared Core

ADC Module Clock Source	360 MHz	PLL VOC/4
ADCON2L.SHRADCS	2	Shared ADC Core Input Clock Divider bits
ADCON3H.CLKDIV	3	ADC Module Clock Source Divider bits
Fadcore	60 MHz	Clock Source / SHRADCS / CLKDIV
Tadcore	16.67 ns	1/Fadcore \geq 14.3ns

Figure 5-1: ADC Module Clock Path Block Diagram



MCC Drivers – Pin Management

MCC Drivers – Interrupt Management & Code Generating

The screenshot shows the MCC v5.3.7 software interface with the 'Interrupt Module' tab selected. On the left, the 'Project Resources' panel lists Peripherals (ADC1, PWM, TMR1) and System modules (Interrupt Module, Pin Module, System Module). A red hand cursor is pointing at the 'Interrupt Module' entry. The 'Device Resources' panel shows a 'Content Manager' section with links to various libraries and documents, including the dsPIC33CH512MP508 Product Page.

Interrupt Manager

Enable Global Interrupts

Module	Interrupt	Description	IRQ Number	Enabled	Priority	Context
DMT	DMTI	Dead Man Timer	45	<input type="checkbox"/>	1	OFF
TMR1	TI	Timer 1	1	<input checked="" type="checkbox"/>	1	OFF
ADC1	ADCAN11	ADC AN11 Convert Done	102	<input type="checkbox"/>	1	OFF
ADC1	ADCAN10	ADC AN10 Convert Done	101	<input type="checkbox"/>	1	OFF
ADC1	ADCAN13	ADC AN13 Convert Done	104	<input type="checkbox"/>	1	OFF
ADC1	ADCAN12	ADC AN12 Convert Done	103	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR3	ADC Oversample Filter 4	123	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR0	ADC Oversample Filter 1	120	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR1	ADC Oversample Filter 2	121	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR2	ADC Oversample Filter 3	122	<input type="checkbox"/>	1	OFF
ADC1	ADCI	ADC Global Interrupt	90	<input type="checkbox"/>	1	OFF
ADC1	ADCAN19	ADC AN19 Convert Done	110	<input type="checkbox"/>	1	OFF
ADC1	ADCAN18	ADC AN18 Convert Done	109	<input type="checkbox"/>	1	OFF
ADC1	ADCAN15	ADC AN15 Convert Done	106	<input type="checkbox"/>	1	OFF
ADC1	ADCAN14	ADC AN14 Convert Done	105	<input type="checkbox"/>	1	OFF

A specific row for ADC1 interrupt ADCAN0 (IRQ 91) is highlighted with a red border. The 'Enabled' column for this row contains a checked checkbox.

MCC Drivers – Interrupt Management & Code Generating

The screenshot shows the MCC v5.3.7 software interface with the 'Interrupt Module' tab selected. The left sidebar displays 'Project Resources' with sections for 'Peripherals' (ADC1, PWM, TMR1) and 'System' (Interrupt Module, Pin Module, System Module). A red hand icon points to the 'Generate' button in the toolbar. The main area is titled 'Easy Setup' and contains the 'Interrupt Manager'. It features a checked checkbox for 'Enable Global Interrupts'. Below is a table listing interrupt configurations:

Module	Interrupt	Description	IRQ Number	Enabled	Priority	Context
DMT	DMTI	Dead Man Timer	45	<input type="checkbox"/>	1	OFF
TMR1	TI	Timer 1	1	<input checked="" type="checkbox"/>	1	OFF
ADC1	ADCAN11	ADC AN11 Convert Done	102	<input type="checkbox"/>	1	OFF
ADC1	ADCAN10	ADC AN10 Convert Done	101	<input type="checkbox"/>	1	OFF
ADC1	ADCAN13	ADC AN13 Convert Done	104	<input type="checkbox"/>	1	OFF
ADC1	ADCAN12	ADC AN12 Convert Done	103	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR3	ADC Oversample Filter 4	123	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR0	ADC Oversample Filter 1	120	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR1	ADC Oversample Filter 2	121	<input type="checkbox"/>	1	OFF
ADC1	ADFLTR2	ADC Oversample Filter 3	122	<input type="checkbox"/>	1	OFF
ADC1	ADCI	ADC Global Interrupt	90	<input type="checkbox"/>	1	OFF
ADC1	ADCAN19	ADC AN19 Convert Done	110	<input type="checkbox"/>	1	OFF
ADC1	ADCAN18	ADC AN18 Convert Done	109	<input type="checkbox"/>	1	OFF
ADC1	ADCAN15	ADC AN15 Convert Done	106	<input type="checkbox"/>	1	OFF
ADC1	ADCAN14	ADC AN14 Convert Done	105	<input type="checkbox"/>	1	OFF

A specific row for ADC1 interrupt ADCAN0 is highlighted with a red border. The table below shows the details for this row:

ADC1	ADCAN0	ADC AN0 Convert Done	91	<input checked="" type="checkbox"/>	6	OFF
------	--------	----------------------	----	-------------------------------------	---	-----

BasicMCCCodebase

Projects Files Resource Management (MCC) Start Page MPLAB X Store Pin Module System Module Interrupt Module TMR1 PWM main.c LOG PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s) Search (#&+!)

POW201_Master

- Header Files
- Important Files
- Linker Files
- Source Files
 - main.c
 - MCC Generated Files
- Libraries
- Loadables
- Secondaries

Source History

```

36     BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE
37     FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN
38     ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY,
39     THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.
40
41     MICROCHIP PROVIDES THIS SOFTWARE CONDITIONALLY UPON YOUR ACCEPTANCE OF THESE
42     TERMS.
43 */
44
45 /**
46  * Section: Included Files
47 */
48 #include "mcc_generated_files/system.h"
49
50 /**
51  * | Main application
52 */
53 int main(void)
54 {
55     // initialize the device
56     SYSTEM_Initialize();
57     while (1)
58     {
59         // Add your application code
60     }
61     return 1;
62 }
63 /**
64  * End of File
65 */
66
67

```

Pin Manager: Package View PWM Generator Summary

MICROCHIP
dsPIC33CH512MP508

POW201_Master - Dashboard Navigator

Output Pin Manager: Grid View Notifications [MCC]

Configuration Loading Error MPLAB® Code Configurator

Project Type: Application - Configuration: default

Device

- dsPIC33CH512MP508
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable

Packs

- dsPIC33CH-MP_DFP (1.12.352)

Compiler Toolchain

- XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
 - Production Image: Optimization: gcc 0
 - Device support information: dsPIC33CH-MP_DFP (1.12)

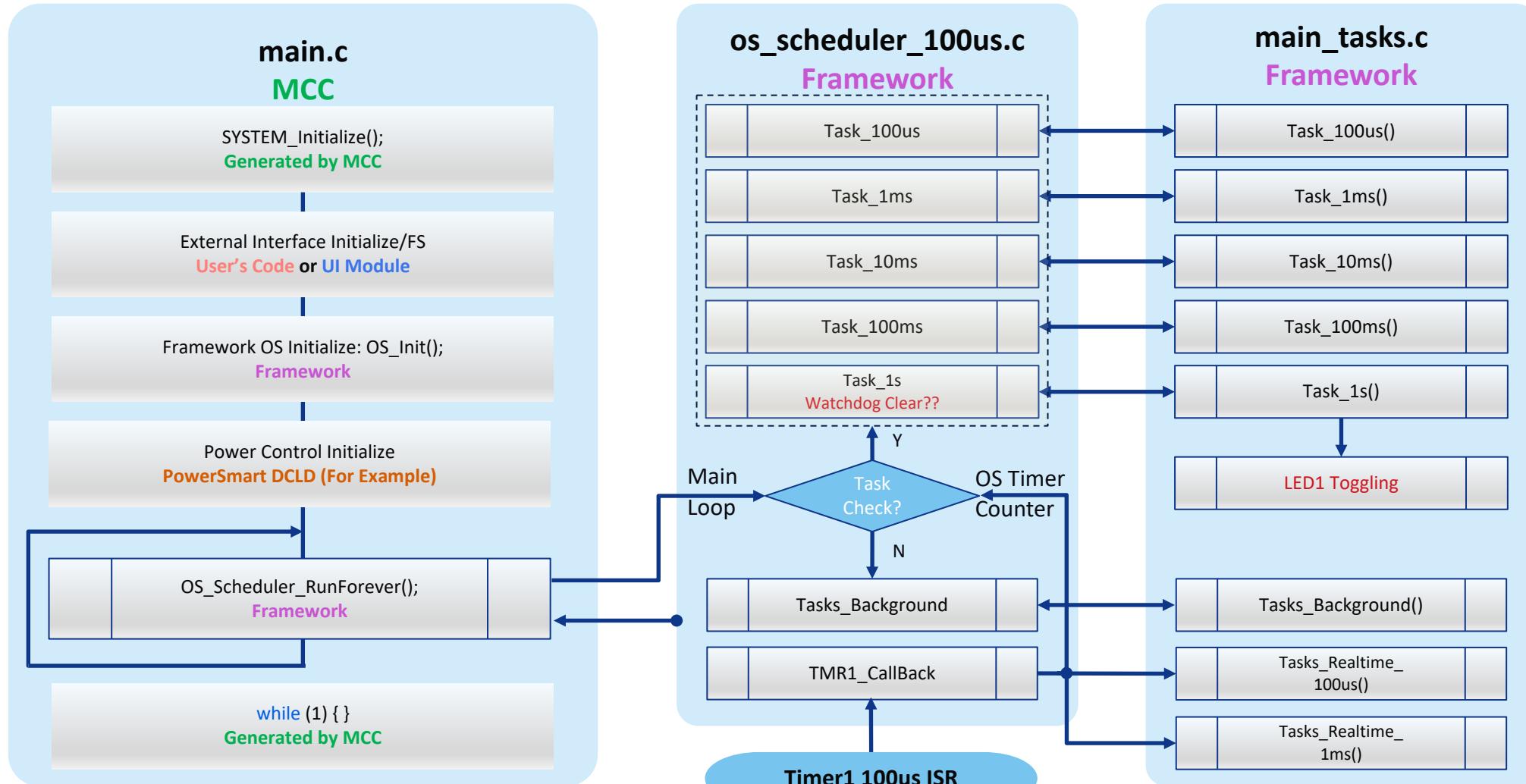
Memory

- Usage Symbols disabled. Click to enable Load Symbols.
- Data 49,152 (0xC000) bytes
- Program 0 (0x0) words

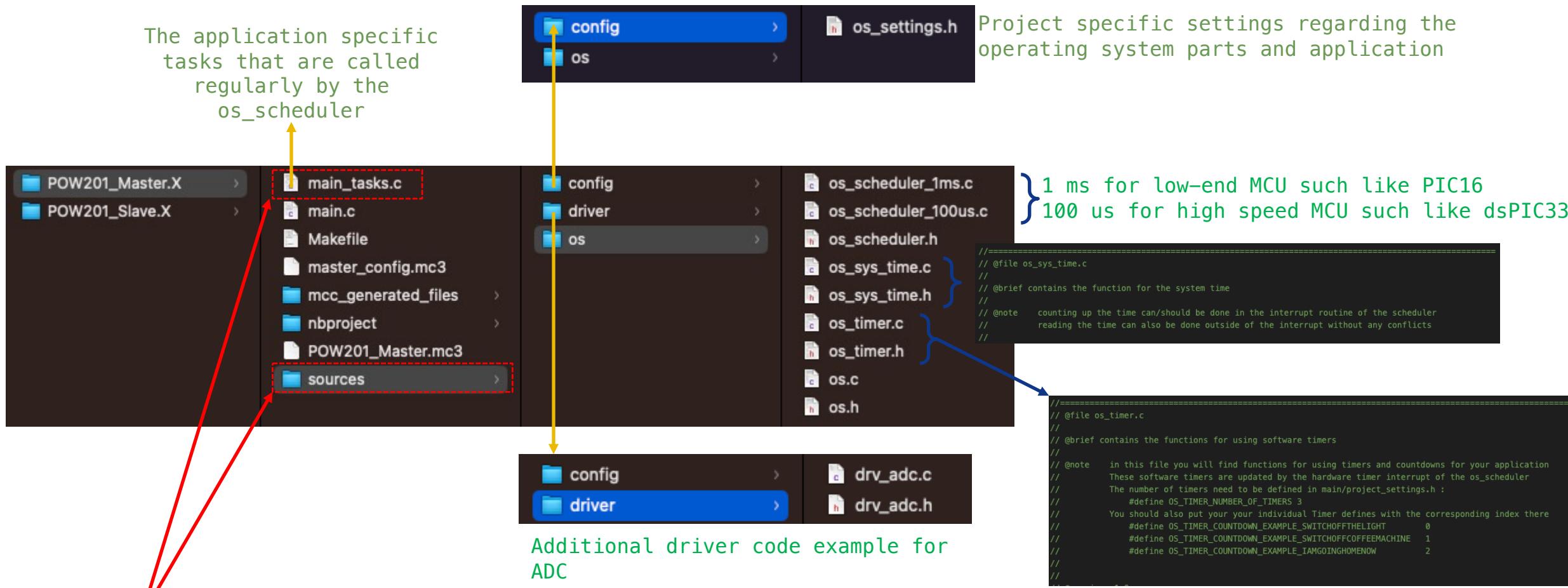
17:00:17.886 INFO: mcc_generated_files/reset.h Success. New file.
17:00:17.886 INFO: mcc_generated_files/reset_types.h Success. New file.
17:00:17.886 INFO: mcc_generated_files/system.c Success. New file.
17:00:17.886 INFO: mcc_generated_files/system.h Success. New file.
17:00:17.886 INFO: mcc_generated_files/system_types.h Success. New file.
17:00:17.886 INFO: mcc_generated_files/tmr1.c Success. New file.
17:00:17.886 INFO: mcc_generated_files/tmr1.h Success. New file.
17:00:17.887 INFO: mcc_generated_files/traps.c Success. New file.
17:00:17.887 INFO: mcc_generated_files/traps.h Success. New file.
17:00:17.887 INFO: mcc_generated_files/watchdog.h Success. New file.
17:00:17.919 INFO: ****
17:00:17.919 INFO: Generation complete (total time: 718 milliseconds)
17:00:17.919 INFO: ****
17:00:17.919 INFO: Generation complete.

The SMPS Firmware Framework

Firmware Structure

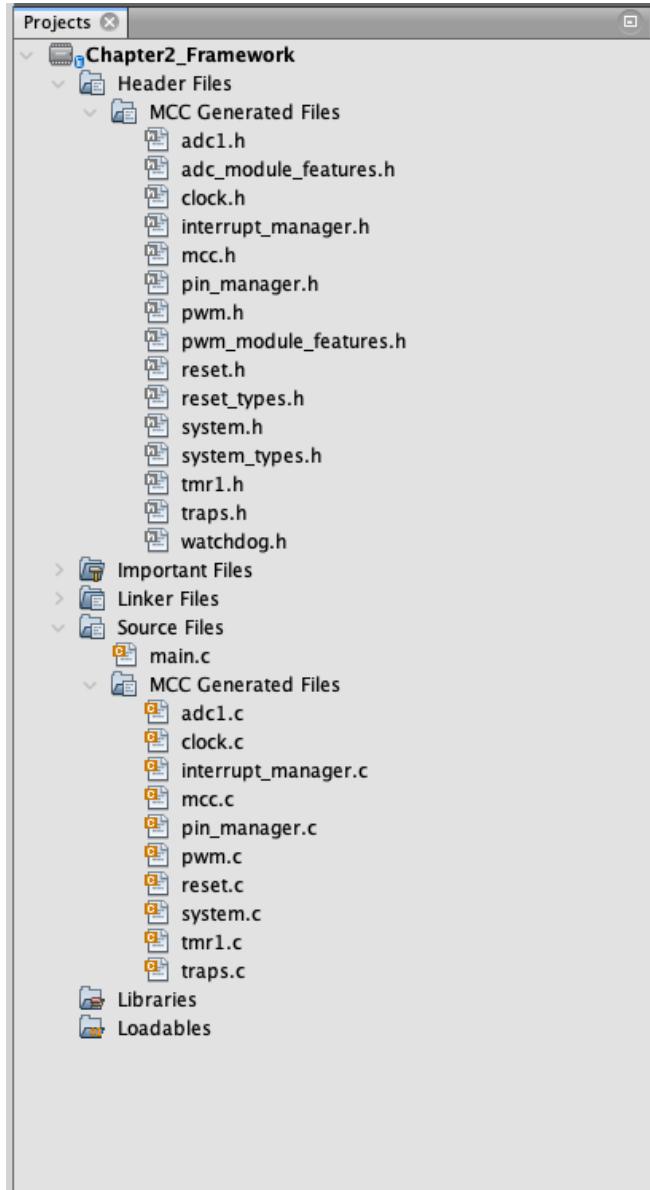


OS Scheduler for Master

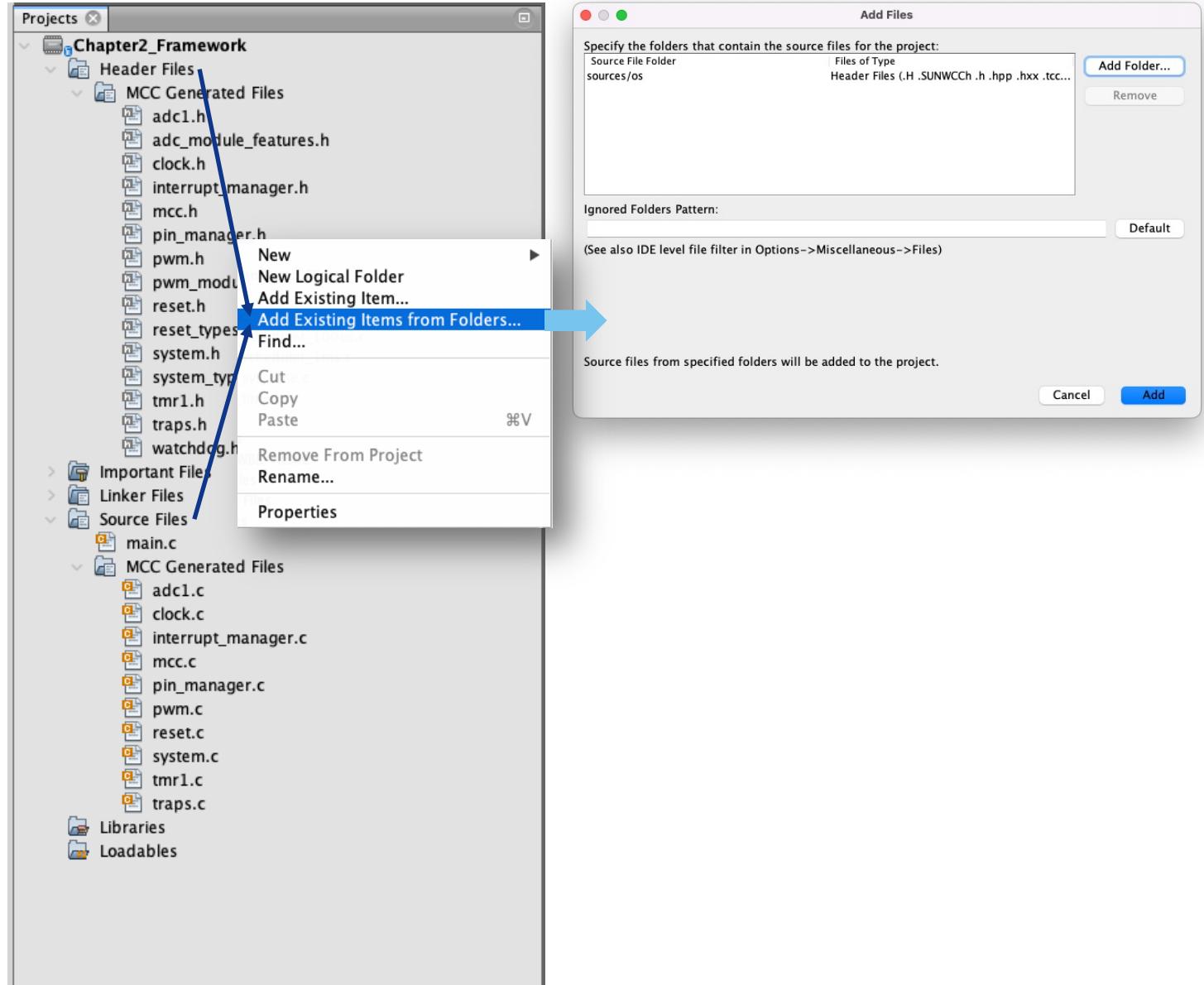


此檔案與目錄皆可在Lab Answer中對應的Lab目錄中找到。

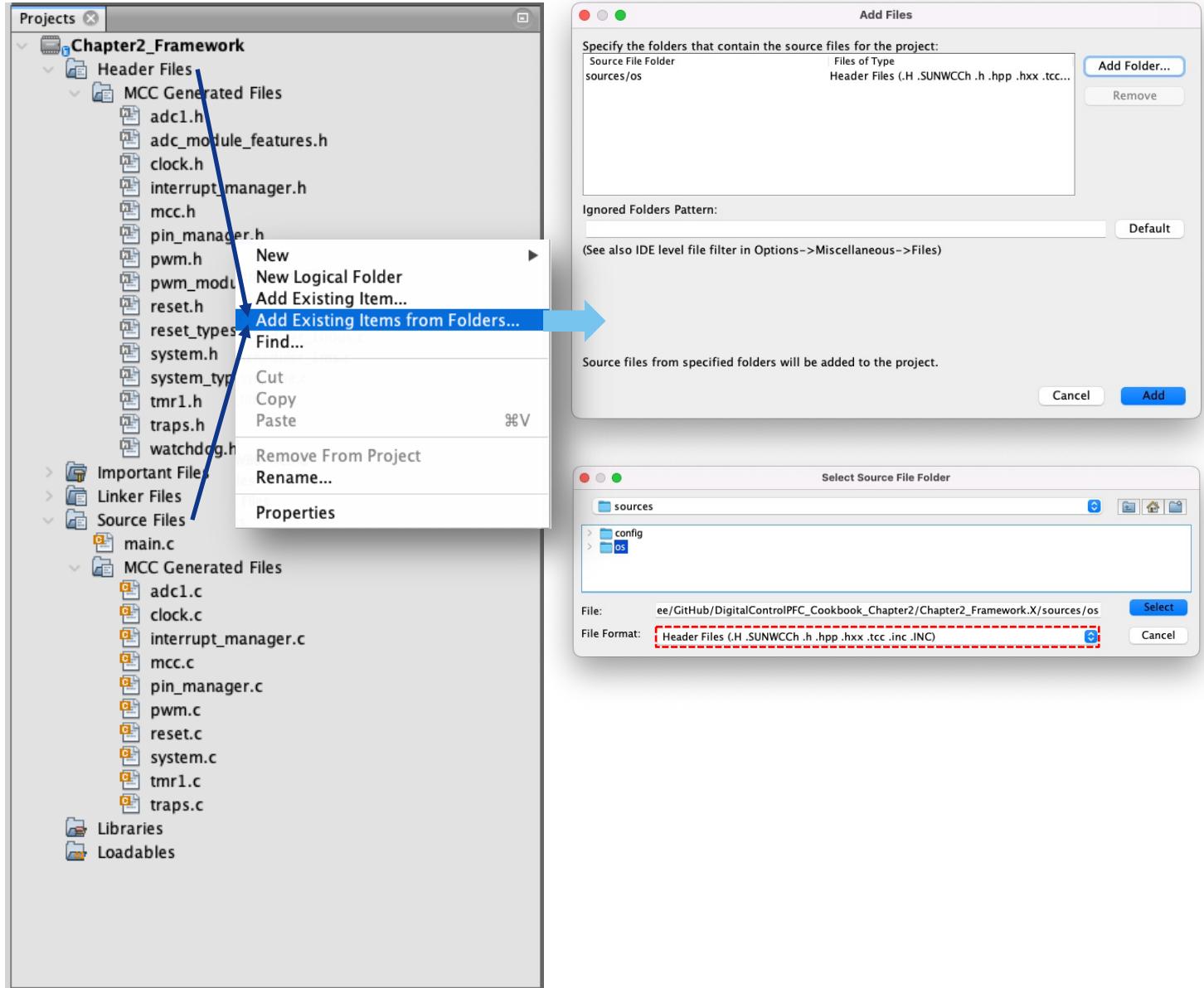
Import OS_Scheduler



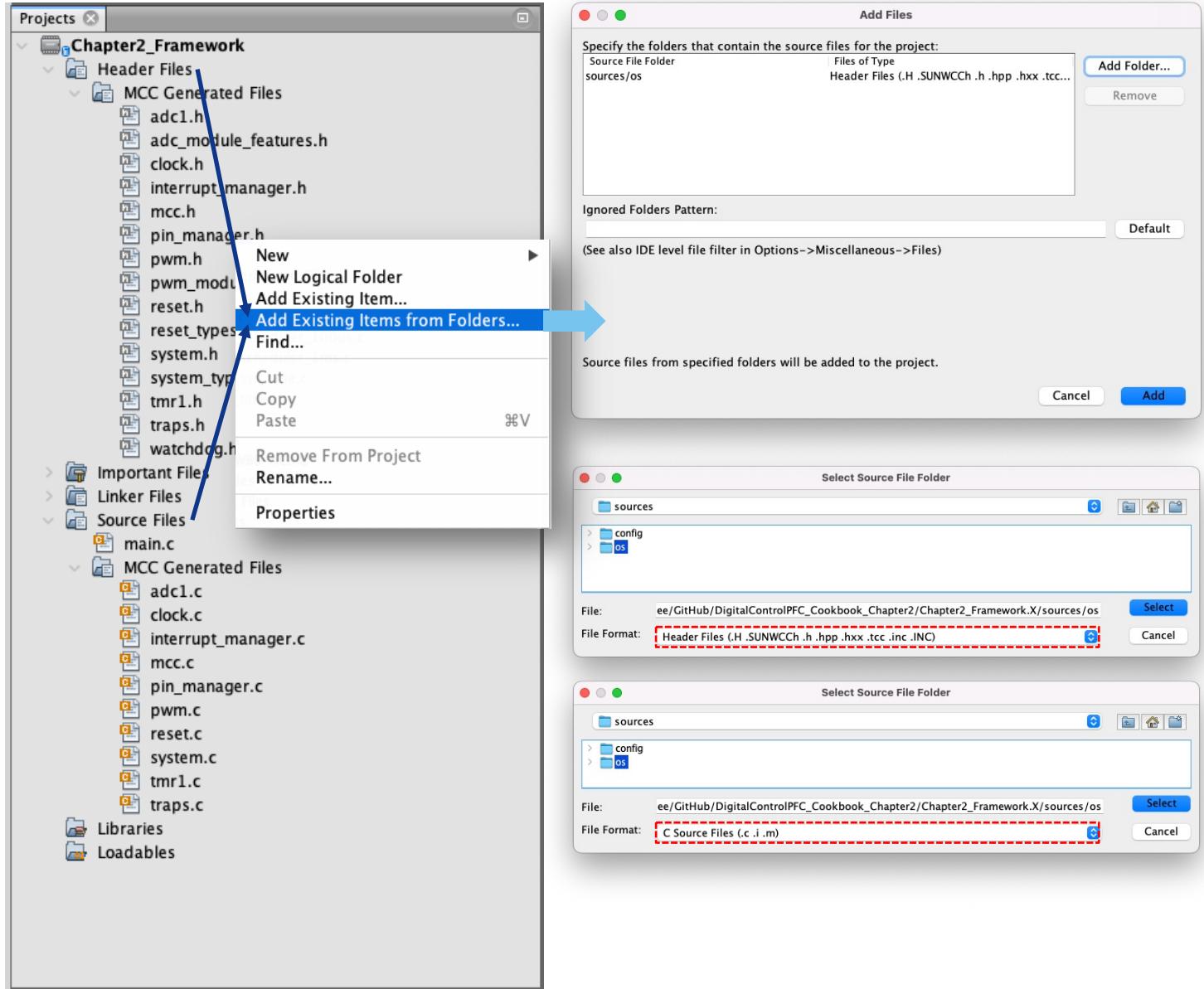
Import OS_Scheduler



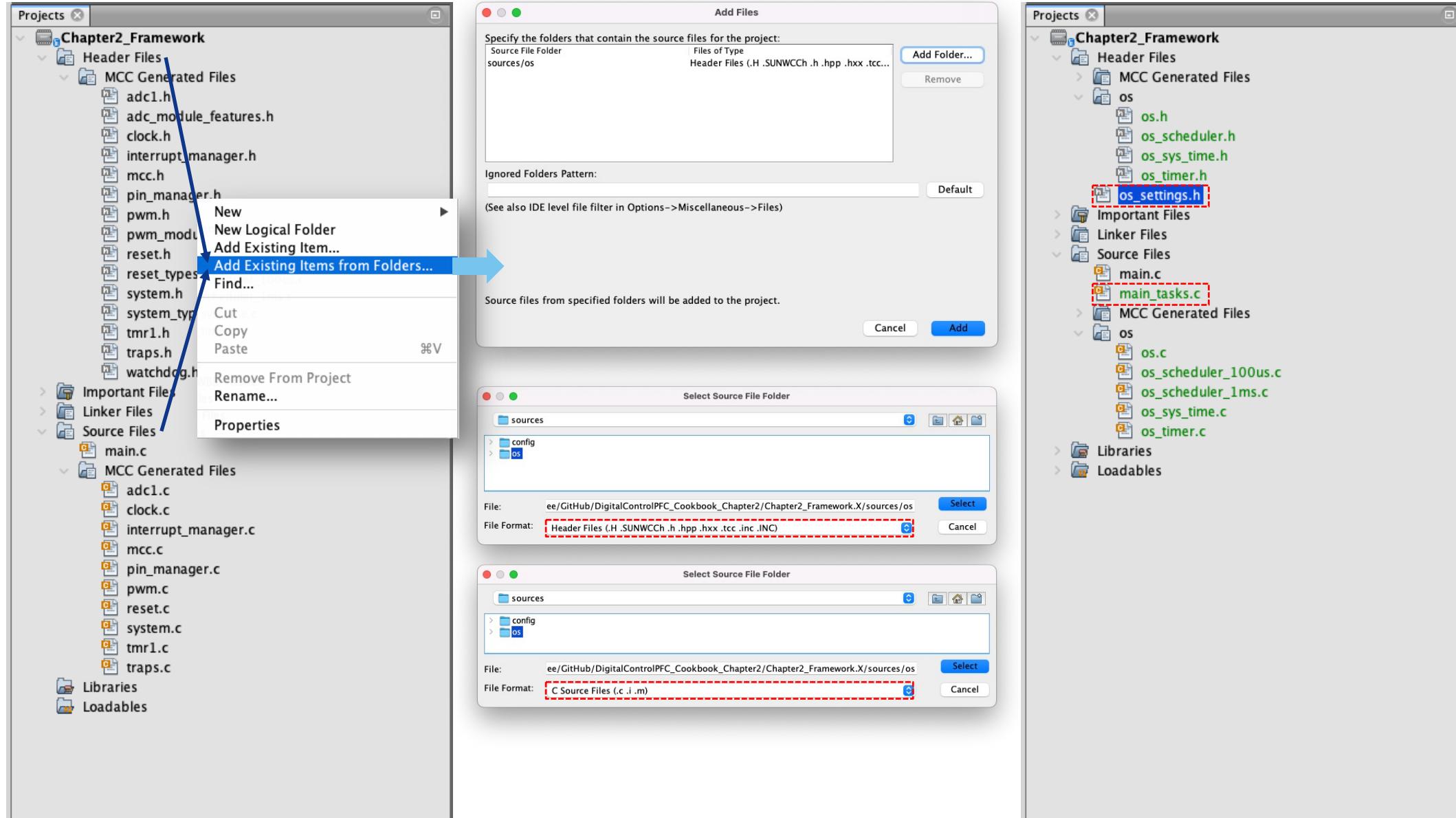
Import OS_Scheduler



Import OS_Scheduler



Import OS_Scheduler



Main & Tasks

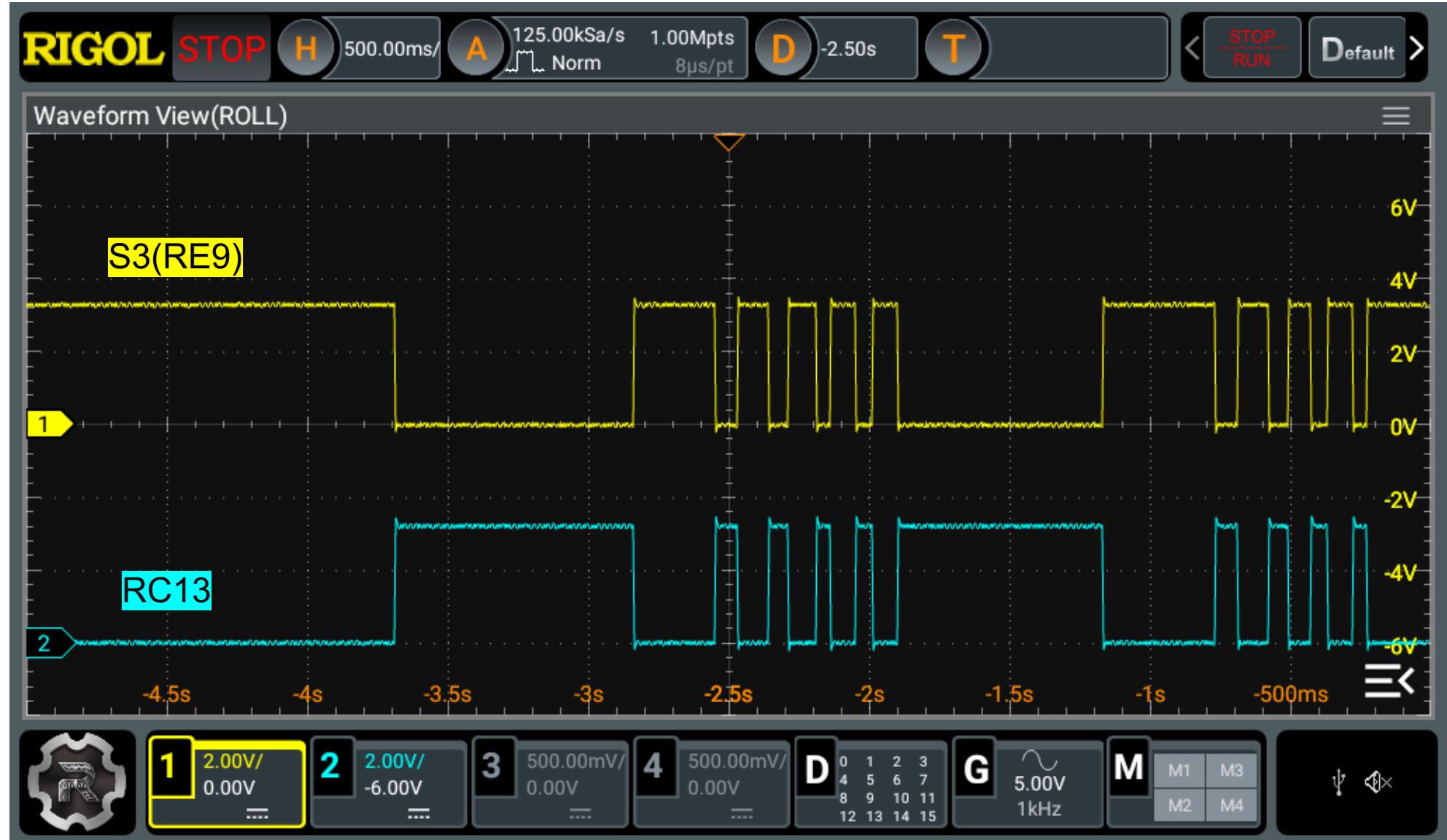
```
main.c
47  */
48  #include "mcc_generated_files/system.h"
49  #include "sources/os/os.h"
50
51  /*
52   *          Main application
53   */
54  int main(void)
55  {
56      // initialize the device
57      SYSTEM_Initialize();
58
59      // OS
60      OS_Init();
61      OS_Scheduler_RunForever();
62
63      while (1)
64      {
65          // Add your application code
66      }
67      return 1;
68
69  /**
70  End of File
71  */
```

```
main_tasks.c
130 // @note there could be some jitter here because it is not called directly by a timer
131 //=====
132
133 void Tasks_1s(void)
134 {
135     // put your application specific code here that needs to be called every second
136     LED1_Toggle();
137 }
138
139 //=====
140 // @brief Tasks_Background gets called all the time when no other of the above tasks
```

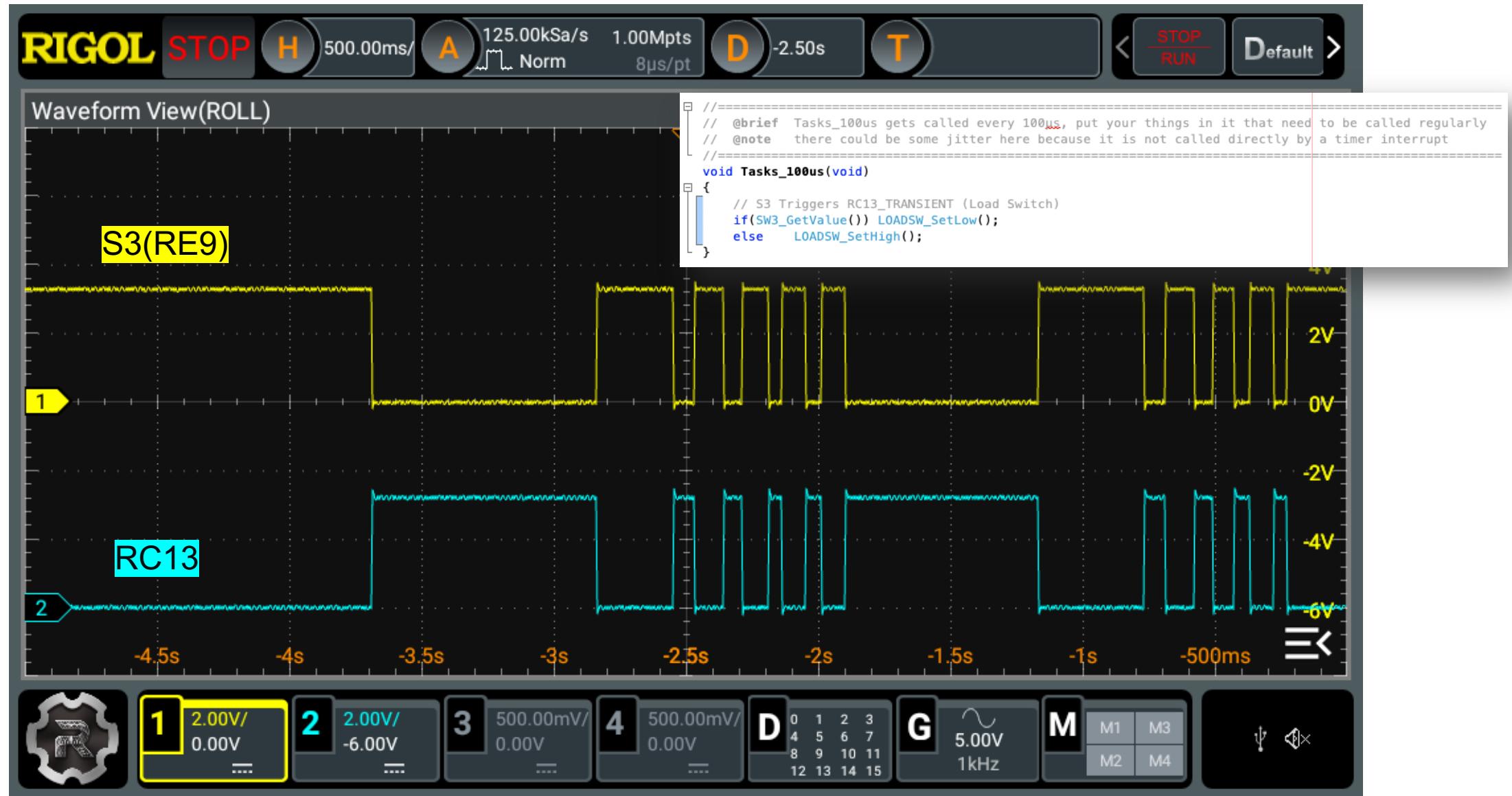
0.5 Hz LED1 Toggle



S3(RE9) Triggers RC13_TRANSIENT



S3(RE9) Triggers RC13_TRANSIENT



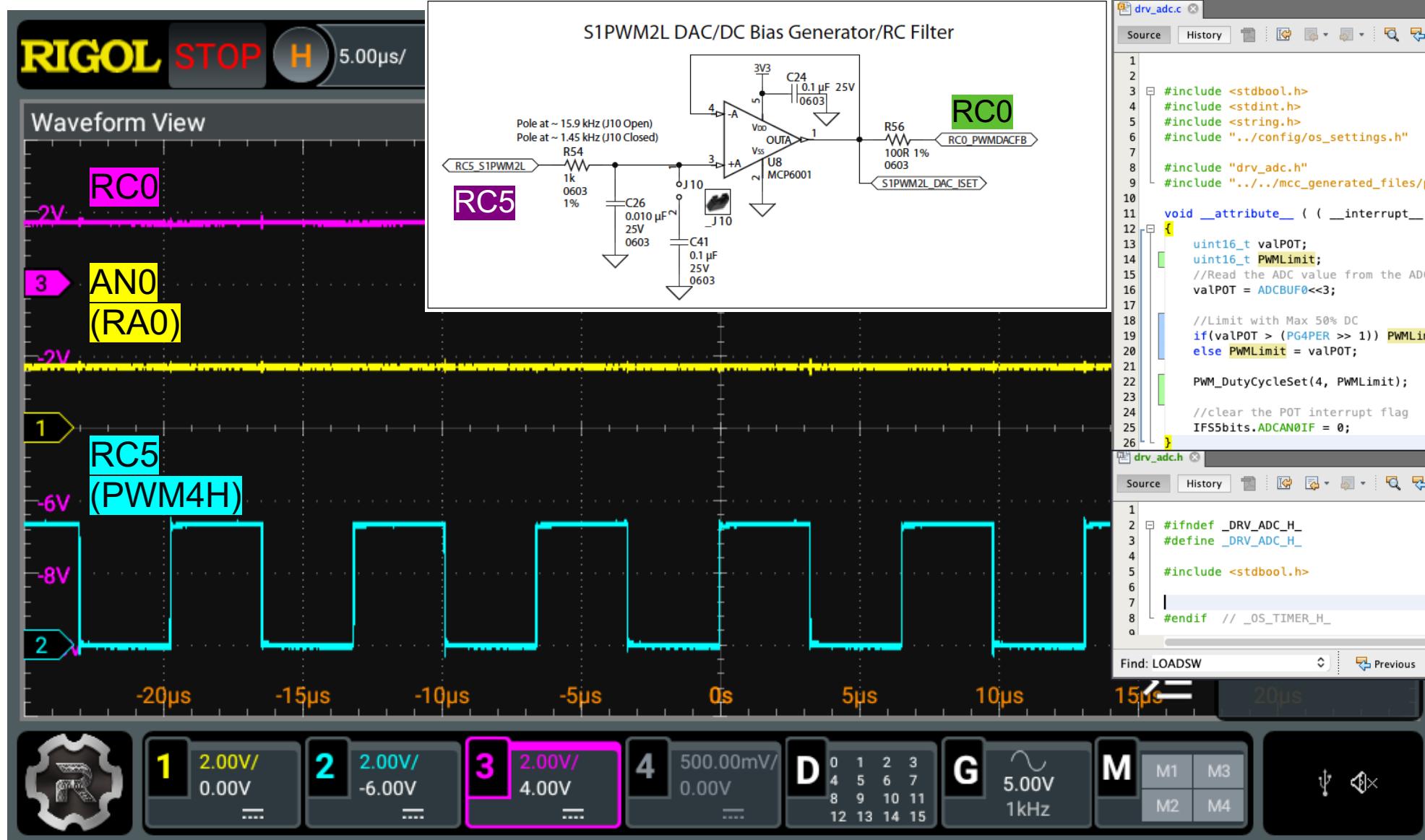
AN0-POT Sets RC5_Master PWM4H



AN0-POT Sets RC5_Master PWM4H



AN0-POT Sets RC5_Master PWM4H



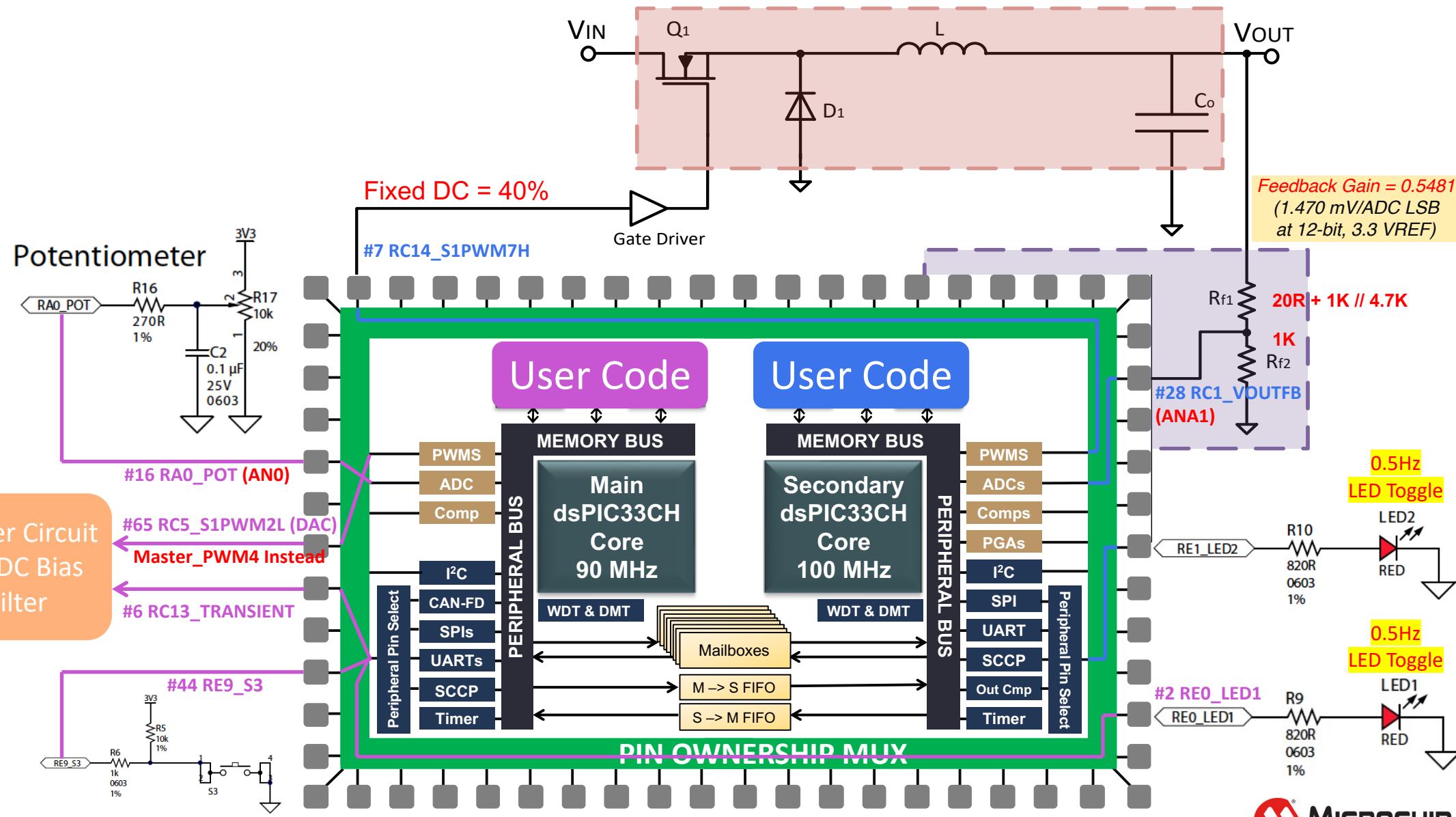
MCC + OS Scheduler for Dual Cores

Lab2

Reference manual on  **MICROCHIP Developer Help**
<https://microchipdeveloper.com/mcc:mplab-code-configure-support-for-dual-core-devices>

Lab #2: MCC + OS Scheduler for Dual Cores

Main Core
Secondary Core



Lab #2: MCC + OS Scheduler for Dual Cores

Main Core

Secondary Core

User Code (90MIPS)

- **Main loop:**
 - OS Scheduler
 - 0.5Hz Toggle LED1
 - Check S3 to trigger RC13_TRANSIENT
- **ISR:**
 - 1ms Timer1 for OS Scheduler
 - AN0-POT to set RC5_S1PWM2L(M-PWM4)
 - Max 50%
- **Peripheral:**
 - PWM4H – ~152KHz (Initial 0% Duty Cycle)
 - I/O: RE9-In, RC13-Out, RE0-Out
 - Timer1 with 1ms ISR
 - AN0 with ISR (IP=1)
 - Triggered by PWM2_Trigger1

User Code (100MIPS)

- **Main loop:**
 - OS Scheduler
 - 0.5Hz Toggle LED2
- **ISR:**
 - 1ms Timer1 for OS Scheduler
 - ANA1-Vfb without user code.
- **Peripheral:**
 - PWM7H – 250KHz & Fixed 40% Duty Cycle
 - I/O: RE1-Out
 - Timer1 with 1ms ISR
 - ANA1 with ISR (IP=6)
 - Triggered by PWM7_Trigger1

Set Secondary Core on Master-MCC

MCC v5.3.7

Project Resources Generate Import... Export ?

Peripherals

- ADC1
- PWM
- Slave Core
- TMR1

System

- Interrupt Module
- Pin Module
- System Module

Device Resources

Content Manager

- Temperature Sensor
- Turnkey Touch
- W/INC15XX
- X2C

Peripherals

- CAN FD
- CBG
- CLC
- GPI-DAC

POW201_Master - Dashboard

Project Type: Application – Configuration: default

Device

- dsPIC33CH512MP508
- Checksum: Blank, no code loaded
- CRC32: Hex file unavailable

Packs

- dsPIC33CH-MP_DFP (1.12.352)

Compiler Toolchain

- XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
- Production Image: Optimization: gcc 0
- Device support information: dsPIC33CH-MP_DFP (1.12.352)

Memory

How do I? Keyword(s) Search (⌘+I)

Easy Setup Registers

Save Master Settings

Slave Project Name: POW201_Slave

Master Slave Interface

Mailbox Configuration

Buffers Used(in Bytes): 0

Available Buffer(in Bytes): 32

Slave Reset Configuration

Reset Slave on Master Reset

Disable Slave on Slave Reset

Slave DMT

Slave ICD

Emulator Pin Placement: Communicate on PGC2 and PGD2

Slave Clock

Clock Source: FRC Oscillator

Enable FRC Postscaler

PLL Enable

Clock Output Pin Configuration: OSC2 is general purpose digital I/O pin

Enable Clock Switching

Enable Fail-Safe Monitor

Slave Watchdog

Pin Manager: Package View

Output Pin Manager: Grid View Notifications [MCC] 21:57 INS

Set Secondary Core on Master-MCC

Screenshot of the Microchip MCC v5.3.7 software interface showing the configuration of a Slave Core.

The left sidebar shows the Project Resources panel with the following items:

- Peripherals: ADC1, PWM, Slave Core (highlighted with a red circle), TMR1.
- System: Interrupt Module, Pin Module, System Module.

The main configuration area is titled "Slave Core". It includes sections for:

- Save Master Settings: Slave Project Name set to "POW201_Slave".
- Master Slave Interface:
 - Mailbox Configuration: Buffers Used(in Bytes) 0, Available Buffer(in Bytes) 32.
 - Slave Reset Configuration: Reset Slave on Master Reset, Disable Slave on Slave Reset.
- Slave DMT.
- Slave ICD: Emulator Pin Placement set to "Communicate on PGC2 and PGD2".
- Slave Clock:
 - Clock Source: FRC Oscillator.
 - Enable FRC Postscaler.
 - PLL Enable.
- Clock Output Pin Configuration: OSC2 is general purpose digital I/O pin.
- Slave Watchdog:
 - Enable Clock Switching.
 - Enable Fail-Safe Monitor.

A blue arrow points from the "Slave Core" item in the Project Resources list to the "Slave Core" configuration section. A red hand icon is placed over the "Slave Core" configuration section.

The bottom left shows the "POW201_Master - Dashboard" with the following details:

- Project Type: Application – Configuration: default.
- Device: dsPIC33CH512MP508, Checksum: Blank, no code loaded, CRC32: Hex file unavailable.
- Packs: dsPIC33CH-MP_DFP (1.12.352).
- Compiler Toolchain: XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin], Production Image: Optimization: gcc 0.
- Memory: Device support information: dsPIC33CH-MP_DFP (1.12).

The bottom right corner features the Microchip logo.

Set Secondary Core on Master-MCC

The screenshot shows the Microchip MCC v5.3.7 software interface with the following key components and configurations:

- Project Resources:** ADC1, PWM, Slave Core, TMR1.
- System:** Interrupt Module, Pin Module, System Module.
- Device Resources:** Temperature Sensor, Turnkey Touch, WINC15XX, X2C, CAN FD, CBG, CLC, GND, DAC.
- Dashboard:** POW201_Master - Project Type: Application - Configuration: default, Device: dsPIC33CH512MP508, Compiler Toolchain: XC16 (v2.10), Memory.
- Resource Management [MCC] Tab:** Shows files like ...in.c, main_tasks.c, os_scheduler_100us.c, Pin Module, Interrupt Module, System Module, ADC1, Slave Core.
- Slave Core Configuration:**
 - Save Master Settings:** A red arrow points to this button.
 - Slave Project Name:** POW201_Slave.
 - Master Slave Interface:**
 - Mailbox Configuration:** Buffers Used(in Bytes): 0, Available Buffer(in Bytes): 32.
 - Slave Reset Configuration:** Checkboxes: Reset Slave on Master Reset (checked), Disable Slave on Slave Reset (checked).
 - Slave DMT:**
 - Slave ICD:** Emulator Pin Placement: Communicate on PGC2 and PGD2.
 - Slave Clock:** Clock Source: FRC Oscillator.
 - Enable FRC Postscaler:** Unchecked.
 - PLL Enable:** Checked.
 - Clock Output Pin Configuration:** OSC2 is general purpose digital I/O pin.
 - Enable Clock Switching:** Checked.
 - Enable Fail-Safe Monitor:** Unchecked.
 - Slave Watchdog:**
- File Explorer:** Shows project files: macos.gitignore, POW201_Master.X, POW201_Slave.X, README.md, Reference Doc, master_config.mc3, mcc_generated_files, nbproject, POW201_Master.mc3, sources.
- Bottom Navigation:** Output, Pin Manager: Grid View, Notifications [MCC], 21:57, INS.
- Microchip Logo:** MICROCHIP

Important: Output Pin Ownership Settings on Master-MCC

The screenshot shows the Microchip MPLAB X IDE interface with several windows open:

- Projects**: Shows the current project is "MCC v5.3.7".
- File**: Standard file operations.
- Resource Management [MCC]**: Project resources window.
- Start Page**: Main application window.
- MPLAB X Store**: Software store.
- Pin Module**: Pin configuration window.
- Interrupt Module**: Interrupt configuration window.
- System Module**: System configuration window.
- LOG**: Log window.
- PC: 0x0**: Address window.
- oab sab da dc n ov z c**: Address window.
- How do I? Keyword(s)**: Help search bar.
- Pin Manager: Package View**: Shows the package pinout for the TQFP80 package, mapping pins to their functions like ADC1, PWM, and GPIO.
- Output - MPLAB® Code Configurator**: Pin manager grid view for Port C, Port D, and Port E. It lists modules like PWM4-H, PWM4-L, and various Pin Modules (GPIO, PWMEA, PWMEB, PWMEC, PWMED) and Slave Core components (TMR1, motorBench® Development Suite, Pac193xLibrary, POTENTIOMETER).
- Device Resources**: Device resources window.
- Content Manager**: Content manager window.
- Libraries**: Library manager window listing various Microchip libraries such as 16-bit Bootloader, CHARACTER LCD, CryptoAuthLibrary, DacLibrary, FatFs, Foundation Services, LED, LED_BLUE, LED_GREEN, LED_RED, MCP802X, motorBench® Development Suite, Pac193xLibrary, and POTENTIOMETER.

Important: Output Pin Ownership Settings on Master-MCC

The screenshot shows the MPLAB X IDE interface with several windows open:

- Projects**: Shows the MCC v5.3.7 project.
- File**: Standard file operations.
- Resource Management [MCC]**: Resource management tools.
- Start Page**: Home page.
- MPLAB X Store**: Marketplace.
- Pin Module**: Pin configuration.
- Interrupt Module**: Interrupt configuration.
- System Module**: System configuration.
- LOG**: Log viewer.
- PC: 0x0**: Memory viewer.
- oab sab da dc n ov z c**: Register viewer.
- How do I? Keyword(s)**: Help search.

Pin Manager: Package View window (top right): Displays the pinout for the dsPIC33CH512MP508 chip, showing pins 1 through 80. A red box highlights pins 14, 15, and 16.

Device Resources window (bottom left): Lists device resources including documents and libraries. The **Content Manager** tab is selected.

Pin Manager: Grid View window (center): A grid-based pin configuration tool. It shows Port C, Port D, and Port E. A red box highlights the same pins (14, 15, 16) as in the package view. The grid includes columns for Module, Function, Direction, and pin numbers 10 through 15, 0 through 12, and 13 through 16.

LED2

PWM7H

Important: Output Pin Ownership Settings on Master-MCC

The screenshot shows the MPLAB X IDE interface for configuring pin ownership settings on a dsPIC33CH512MP508 microcontroller. A red hand icon highlights the 'Project Resources' button in the top-left toolbar.

Pin Manager: Package View (Top Right): Shows the physical pin layout for the TQFP80 package, with pins numbered 1 through 80. A red box highlights pins 11, 12, and 13, which correspond to Port C pins 0, 1, and 2 respectively. A green box highlights pins 14, 15, and 16, which correspond to Port D pins 0, 1, and 2 respectively. A yellow box highlights pins 17 through 20, which correspond to Port E pins 0, 1, 2, and 3 respectively.

Output - MPLAB® Code Configurator (Bottom Center): A grid view showing pin assignments. A red hand icon highlights the 'Pin Manager' tab above the grid. The grid shows Port C, Port D, and Port E columns. A red dashed box highlights the Port C column (pins 11, 12, 13) and the Port D column (pins 14, 15). A green dashed box highlights the Port E column (pins 17, 18, 19, 20).

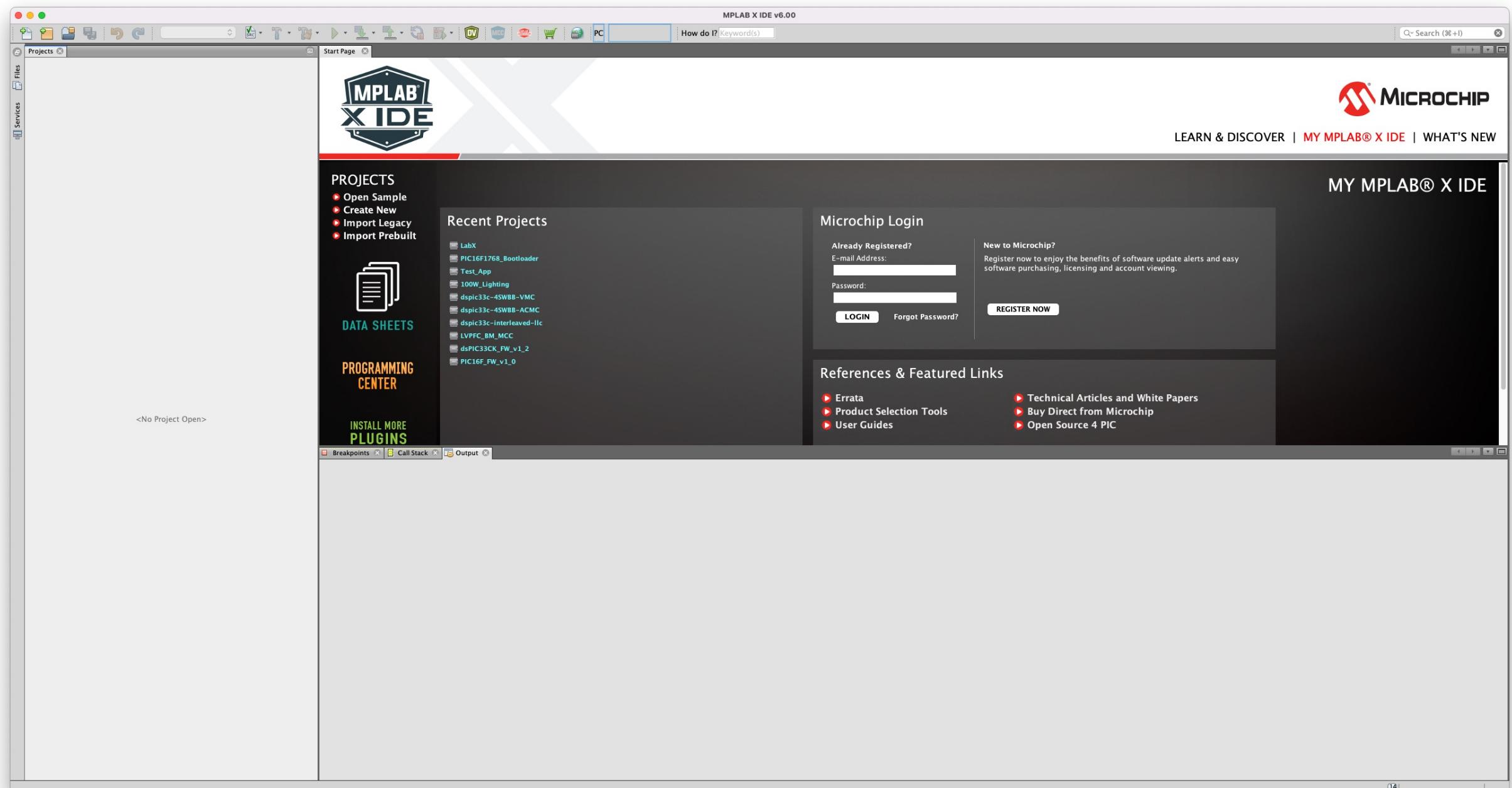
Pin Manager: Grid View (Bottom Left): A detailed grid view of pin assignments. A red hand icon highlights the 'Grid View' tab above the grid. The grid shows Port C, Port D, and Port E columns. A red dashed box highlights the Port C column (pins 11, 12, 13) and the Port D column (pins 14, 15). A green dashed box highlights the Port E column (pins 17, 18, 19, 20).

Pin Manager: Package View (Top Right): Shows the physical pin layout for the TQFP80 package, with pins numbered 1 through 80. A red box highlights pins 11, 12, and 13, which correspond to Port C pins 0, 1, and 2 respectively. A green box highlights pins 14, 15, and 16, which correspond to Port D pins 0, 1, and 2 respectively. A yellow box highlights pins 17 through 20, which correspond to Port E pins 0, 1, 2, and 3 respectively.

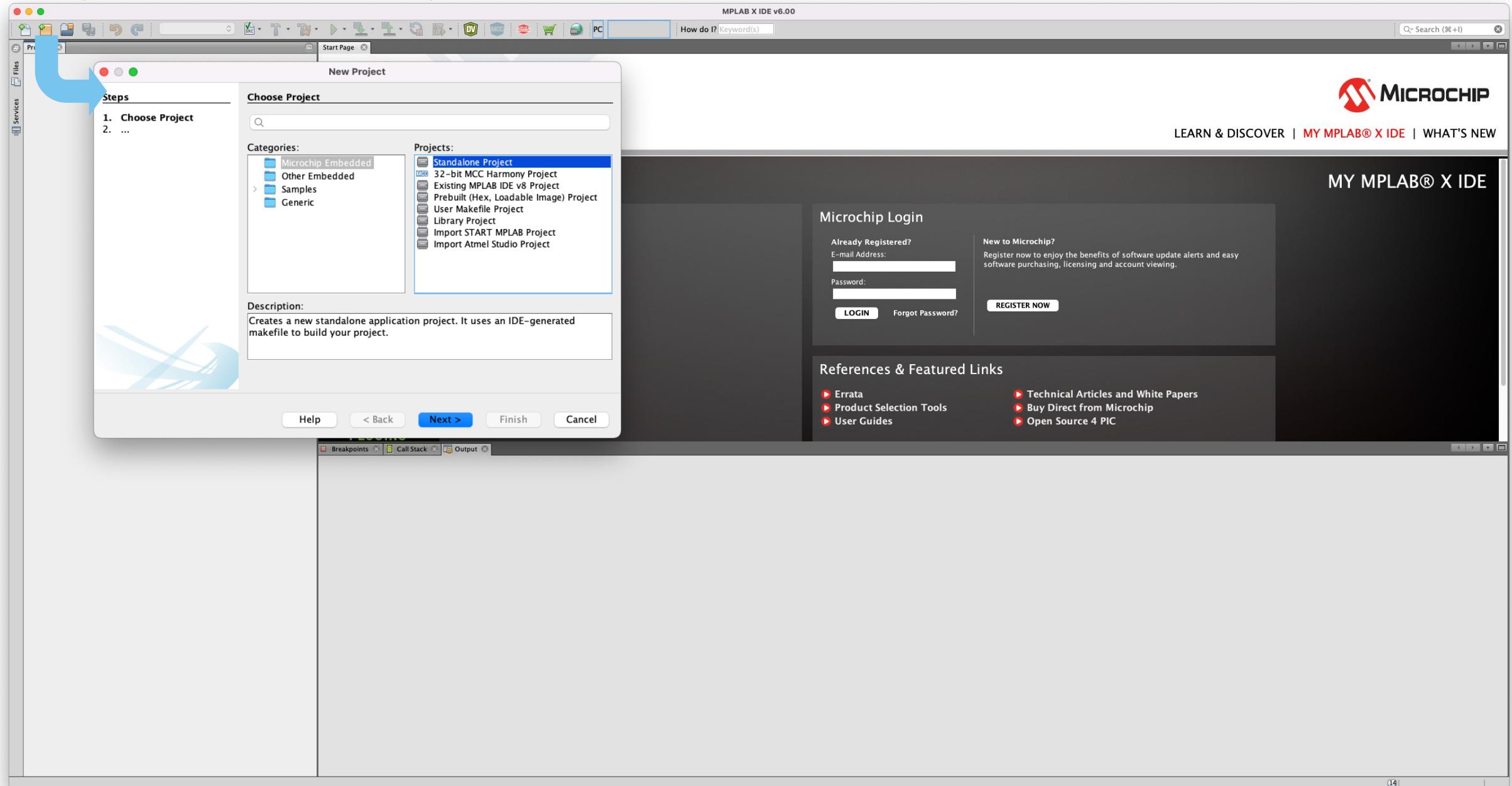
LED2

PWM7H

 **MICROCHIP**



New Project as Slave on Secondary core

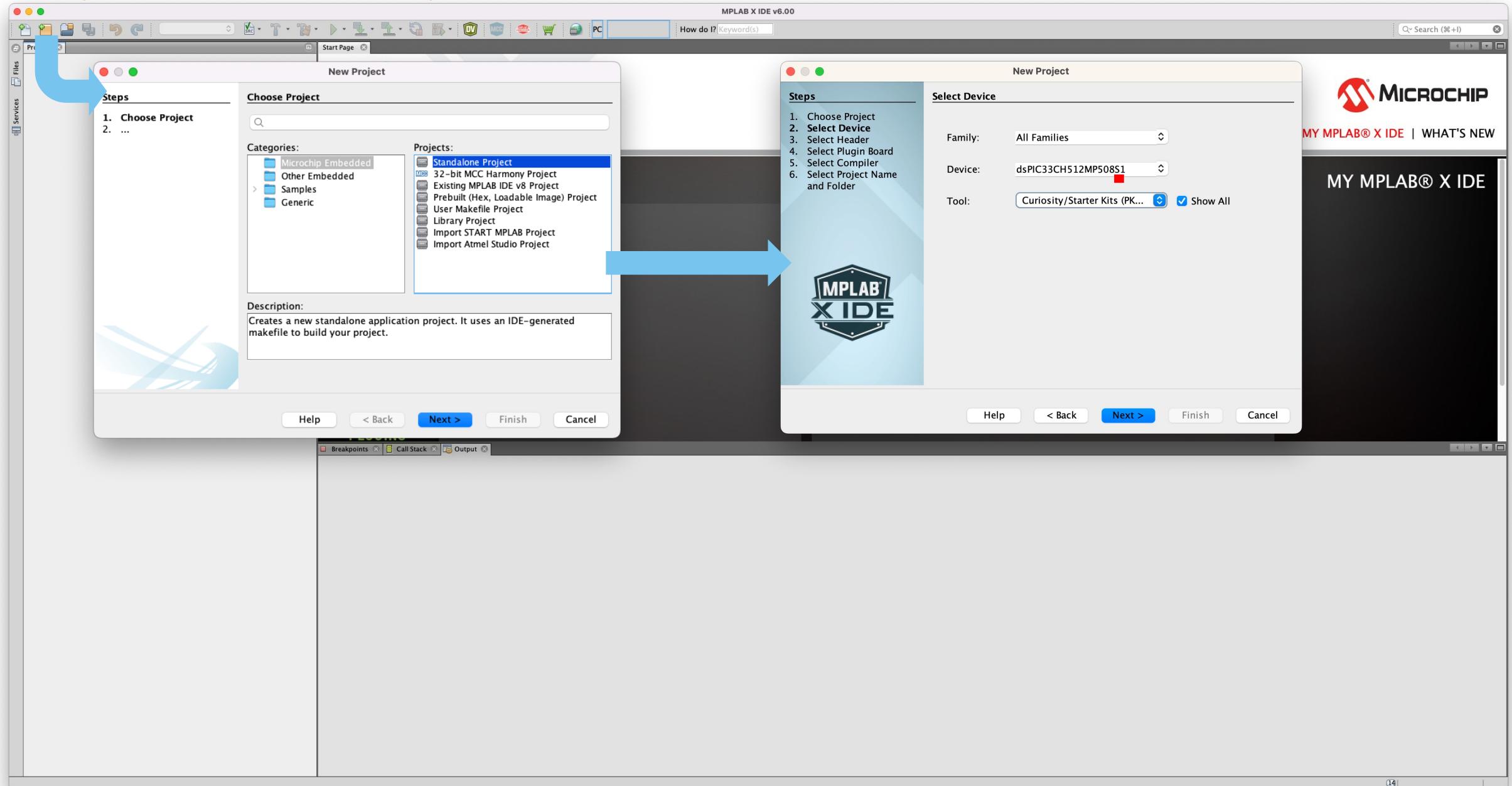


LEARN & DISCOVER | MY MPLAB® X IDE | WHAT'S NEW

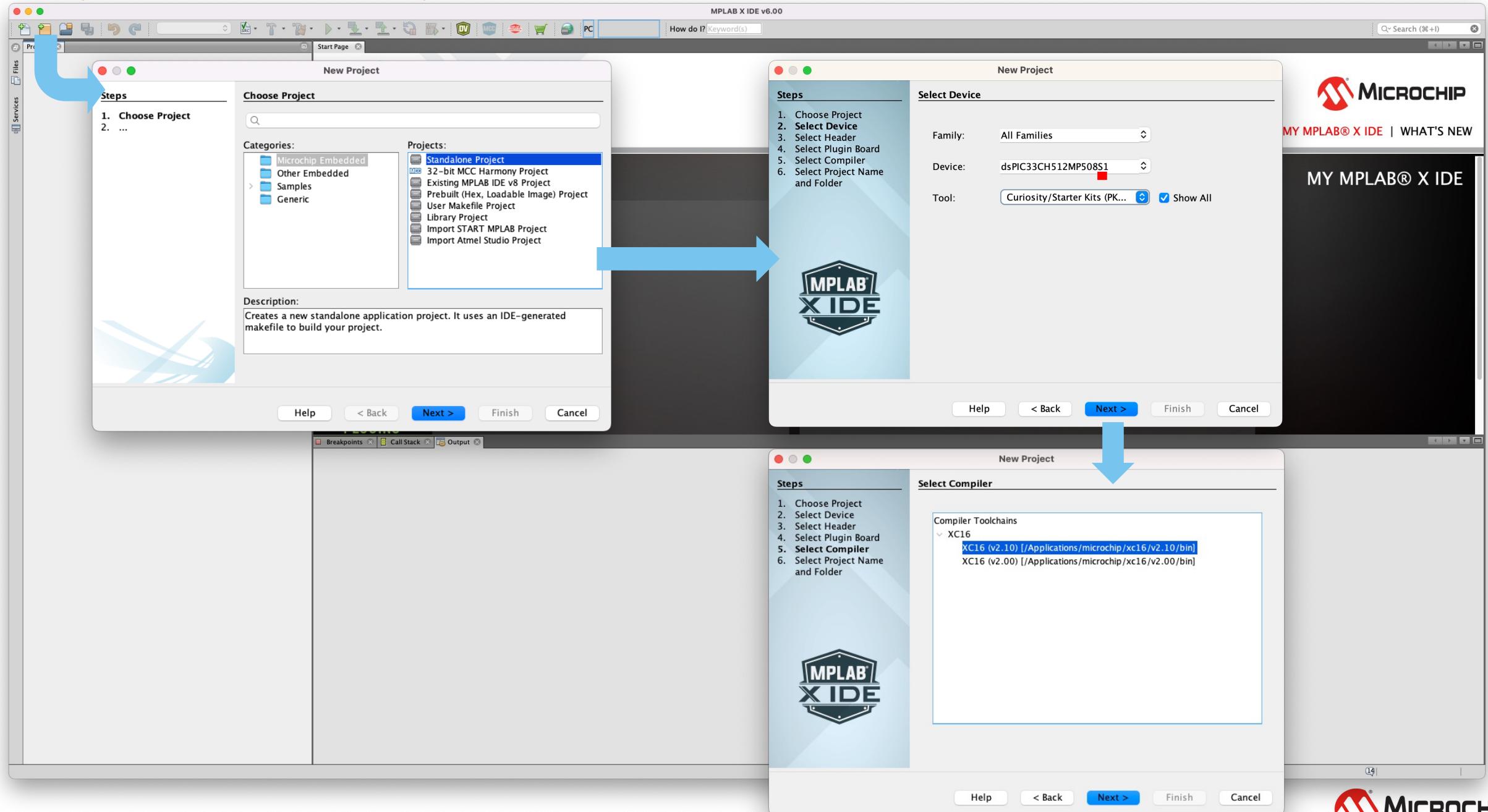


MY MPLAB® X IDE

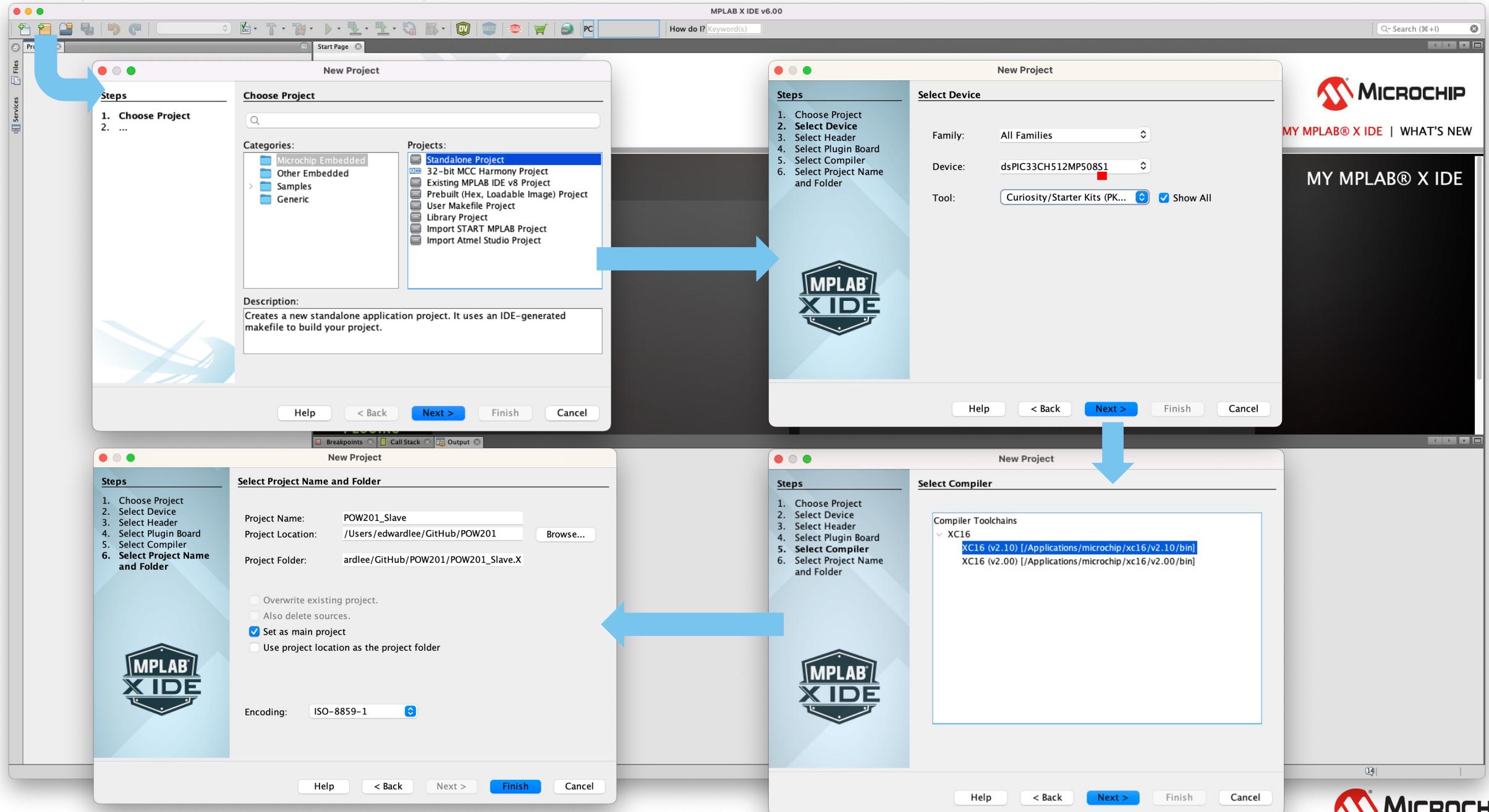
New Project as Slave on Secondary core

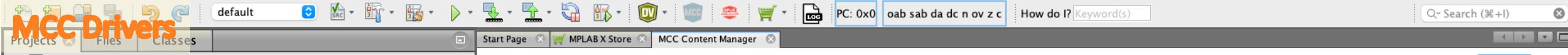


New Project as Slave on Secondary core



New Project as Slave on Secondary core





Projects Files Classes

POW201_Master

- > Header Files
- > Important Files
- > Linker Files
- > Source Files
 - main.c
 - main_tasks.c
- > MCC Generated Files
- > os
- > Libraries
- > Loadables
- > Secondaries

POW201_Slave

- > Header Files
- > Important Files
- > Linker Files
- > Source Files
- > Libraries
- > Loadables

Start Page MPLAB X Store MCC Content Manager

PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s)

Import ?

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody

Supports the MCC Builder
Supports content versioning at driver level
An iteration of MCC Generated Code
Works both on- and off-line

[Release notes and supported devices](#)

MCC Classic

Development process you are accustomed to
All components and libraries that you have used before

[Release notes and supported devices](#)

MPLAB® Harmony

Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

[Release notes and supported devices](#)

Library support may be a key factor in your choice of MCC flavor:

[MCC Melody and MCC Classic - Library Summary](#)

[MPLAB Harmony - Library Summary](#)

Still unsure which content type is right for your project?

POW201_Slave - Dashboard main() - Navigator

POW201_Slave

- Project Type: Application - Configuration: default
- Device
 - dsPIC33CH512MP508S1
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable
- Packs
 - dsPIC33CH-MP_DFP (1.12.352)
 - Curiosity/Starter Kits (PKOB4) (1.11.1054)
- Compiler Toolchain
 - XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
 - Production Image: Optimization: gcc 0

Device support information: dsPIC33CH512MP508S1 v1.12.352

Output Configuration Loading Error MPLAB® Code Configurator

```
16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.
```



Projects Files Classes

POW201_Master

- > Header Files
- > Important Files
- > Linker Files
- > Source Files
 - main.c
 - main_tasks.c
- > MCC Generated Files
- > os
- > Libraries
- > Loadables
- > Secondaries

POW201_Slave

- > Header Files
- > Important Files
- > Linker Files
- > Source Files
- > Libraries
- > Loadables

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody [Select MCC Melody](#)

Supports the MCC Builder
Supports content versioning at driver level
An iteration of MCC Generated Code
Works both on- and off-line

[Release notes and supported devices](#)

MCC Classic [Select MCC Classic](#)

Development process you are accustomed to
All components and libraries that you have used before

[Release notes and supported devices](#)

MPLAB® Harmony [Select MPLAB Harmony](#)

Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

[Release notes and supported devices](#)

Library support may be a key factor in your choice of MCC flavor:

[MCC Melody and MCC Classic - Library Summary](#)

[MPLAB Harmony - Library Summary](#)

POW201_Slave - Dashboard main() - Navigator

POW201_Slave

Project Type: Application - Configuration: default

Device

- dsPIC33CH512MP508S1
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable

Packs

- dsPIC33CH-MP_DFP (1.12.352)
- Curiosity/Starter Kits (PKOB4) (1.11.1054)

Compiler Toolchain

- XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
- Production Image: Optimization: gcc 0

Device support information: dsPIC33CH_MPU508S1_12.25

Output Configuration Loading Error MPLAB® Code Configurator

```
16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.
```

MCC Drivers

default

Projects Files Classes Start Page MPLAB X Store MCC Content Manager Wizard PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s) Search (%+)

Import ?

POW201_Master

- Header Files
- Important Files
- Linker Files
- Source Files
 - main.c
 - main_tasks.c
- MCC Generated Files
- os
- Libraries
- Loadables
- Secondaries

POW201_Slave

- Header Files
- Important Files
- Linker Files
- Source Files
- Libraries
- Loadables

POW201_Slave - Dashboard

main() - Navigator

POW201_Slave

Project Type: Application - Configuration: default

Device

- dsPIC33CH512MP508S1
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable

Packs

- dsPIC33CH-MP_DFP (1.12.352)
- Curiosity/Starter Kits (PKOB4) (1.11.1054)

Compiler Toolchain

- XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin]
- Production Image: Optimization: gcc 0

Device support information: dsPIC33CH512MP508S1 v1.12.352

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody

Supports the MCC Builder
Supports content versioning at driver level
An iteration of MCC Generated Code
Works both on- and off-line

Select MCC Melody

MCC Classic

Development process you are accustomed to
All components and libraries that you have used before

Select MCC Classic

MPLAB® Harmony

Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

Select MPLAB Harmony

MCC Content Manager Wizard

1. Content Type 2. Required Device Content Finish

Required Content

All required content is available locally on your machine. No other download is needed to get started.
To change content versions later, access the Content Manager from Device Resources.

Optional Content

Select optional content to be made available in Device Resources for selection

Optional Content

Component	Version	Description
Libraries		
SD/MMC Card	1.1.0	Driver for SD/MMC cards allowing direct sector read/write operations. Most applications will want to use file system on top of this driver, such as FatFs, to read data written by a PC or to write data that is readable by the PC.

Output

Configuration Loading Error MPLAB® Code Configurator

```
16:06:04.220 INFO: Fetching list of available libraries.
16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
16:06:05.589 INFO: Start MCC v5.3.7
16:06:05.593 INFO: Core v5.5.7 loaded.
```

MCC Drivers

default

Projects Files Classes Start Page MPLAB X Store MCC Content Manager Wizard PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s) Search (%+)

Import ?

POW201_Master

- Header Files
- Important Files
- Linker Files
- Source Files

 - main.c
 - main_tasks.c

- MCC Generated Files

MPLAB X IDE v6.10 - POW201_Slave : default

Projects Files Classes Resource Management [MCC] Pin Module System Module Master Core Interrupt... Pin Manager: Package View

MCC v5.3.7

Project Resources Generate Import... Export ?

System

- Interrupt Module
- Master Core

Device Resources Content Manager

Documents dsPIC33CH512MP508S1 Product Page

Libraries

- CHARACTER LCD
- DacLibrary
- FatFs
- Foundation Services
- LED
- LED_BLUE
- LED_GREEN

POW201_Slave - Dashboard Navigator

POW201_Slave Project Type: Application - Configuration: default

Device dsPIC33CH512MP508S1 Checksum: Blank, no code loaded CRC32: Hex file unavailable

Packs dsPIC33CH-MP_DFP (1.12.352) Curiosity/Starter Kits (PKOB4) (1.11.1054)

Compiler Toolchain XC16 (v2.10) [/Applications/microchip/xc16/v2.10/bin] Production Image: Optimization: gcc 0 Device support information: dsPIC33CH-MP_DFP (1.12.352)

Memory Usage Symbols disabled. Click to enable Load Symbols. Data 16,384 (0x4000) bytes Program 0 (0x0) words Stack Usage Guidance

Debug Tool Curiosity/Starter Kits (PKOB4)

Debug Resources Program BP Used: 0 Free: 3 Data BP Used: 0 Free: 1 Data Capture BP: No Support

MCC Content Manager Wizard

1. Content Type 2. Required Device Content

Select a Content Type

MCC Melody MCC Classic

Select MCC Classic

Select MPLAB Harmony

MPLAB® Harmony Embedded Software Development Framework for 32-bit Microcontrollers and Microprocessors

Release notes and supported devices

Finish

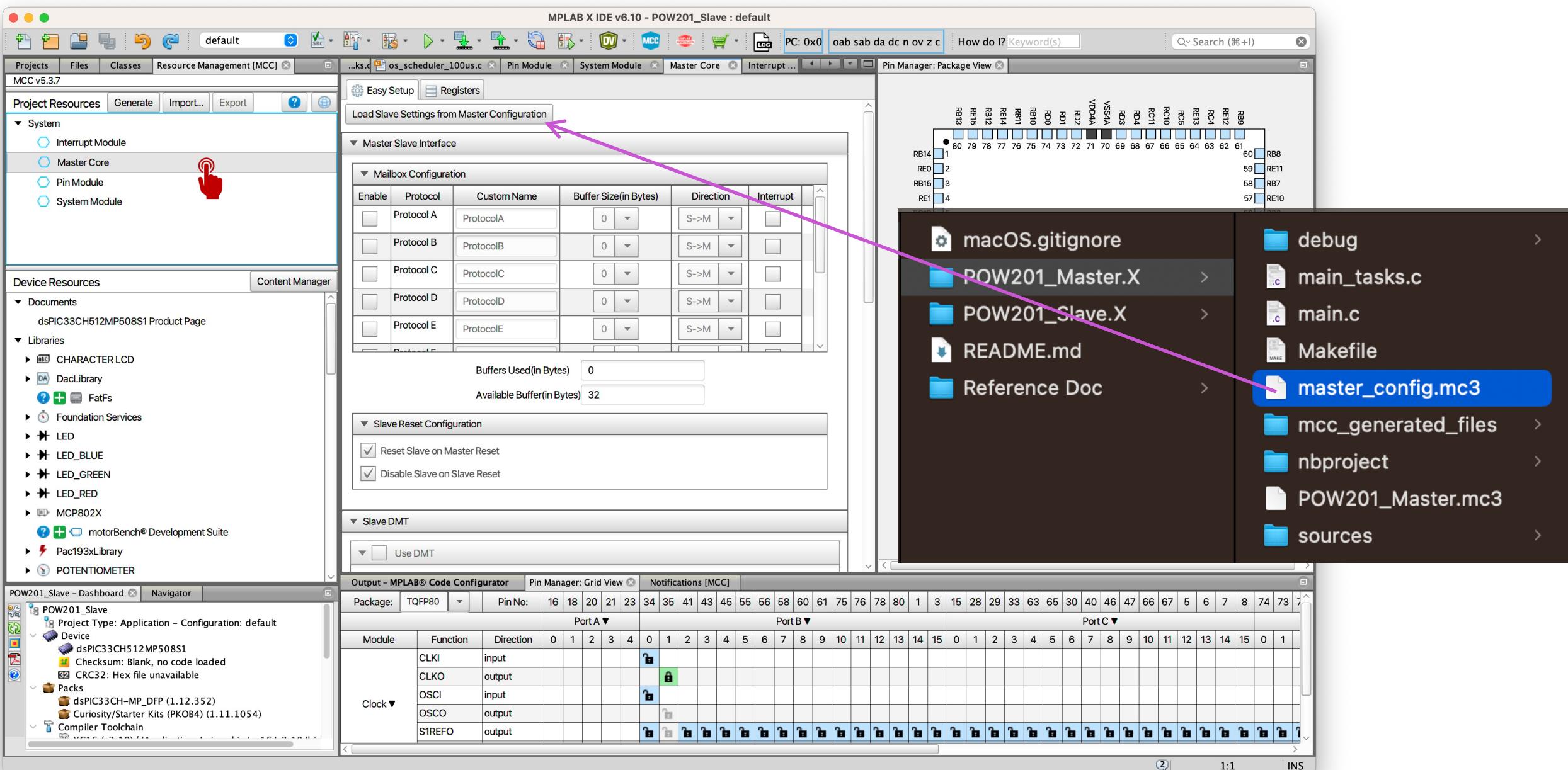
Description

Configuration Loading Error

MPLAB® Code Configurator

16:06:04.220 INFO: Fetching list of available libraries.
 16:06:04.611 INFO: Download Complete: /Users/edwardlee/.mcc/mcc_libraries.xml
 16:06:05.589 INFO: Start MCC v5.3.7
 16:06:05.593 INFO: Core v5.5.7 loaded.

Load Slave Settings from Master Configuration



MCC Drivers – Clock@100 MIPS

Project Resources Generate Import... Export ?

System Interrupt Module Pin Module System Module Content Manager

Pin Manager: Package View

Easy Setup Registers

Clock

8000000 Hz FRC Oscillator (8.0 MHz) Clock Source

FRC Postscaler

PLL Enable

Prescaler: 1:1, Feedback: 1:200, Postscaler1: 1:2, Postscaler2: 1:2

200 MHz Fosc, 100 MHz Fosc/2 **100MIPS**

Auxiliary Clock

8000000 Hz FRC Clock Source

PLL Enable

Prescaler: 1:1, Feedback: 1:125, Postscaler1: 1:2, Postscaler2: 1:1

VCO & AVCO

VCO Divider: Fvco/4, AVCO Divider: Fvco/2

OSC2 is general purpose digital I/O pin

Reference Oscillator Output

CAN FD Clock Generator

Pin Manager: Grid View Notifications [MCC]

PC: 0x0 oab sab da dc n ov z c How do I? Keyword(s)

Search Results

POW009_LabX - Dashboard Navigator Core Versions [MCC]

Device Resources Content Manager

Documents dsPIC33CH512MP508S1 Product Page

Libraries CHARACTER LCD DacLibrary FatFs Foundation Services

LED LED_BLUE LED_GREEN LED_RED MCP802X motorBench® Development Suite Pac193xLibrary POTENTIOMETER RGB LED SERIALUART SWITCH Temperature Sensor Turnkey Touch WINC15XX X2C

Peripherals ADC POTENTIOMETER RGB LED SERIALUART

80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61
RB13 RE15 RB12 RB11 RB10 RD1 RD2 VDD4A RD3 VSS4A RC11 RC10 RC5 RC4 RE12 RB9
RB14 1 RE0 2 RB15 3 RE1 4 RC12 5 RC13 6 RC14 7 RC15 8 NMCLR 9 RD15 10 RD14 11 RE11 12 RB7 13 RE10 14 RB6|PGC3 15 RB5|PGD3 16 RD5 17 RD6 18 RD7 19 VDD3A 20

MICROCHIP

6.1 Primary PLL

The Primary Oscillator and internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. There are two independent instantiations of PLL for the Master and Slave clock subsystems. Figure 6-4 illustrates a block diagram of the Master/Slave core PLL module.

For PLL operation, the following requirements must be met at all times without exception:

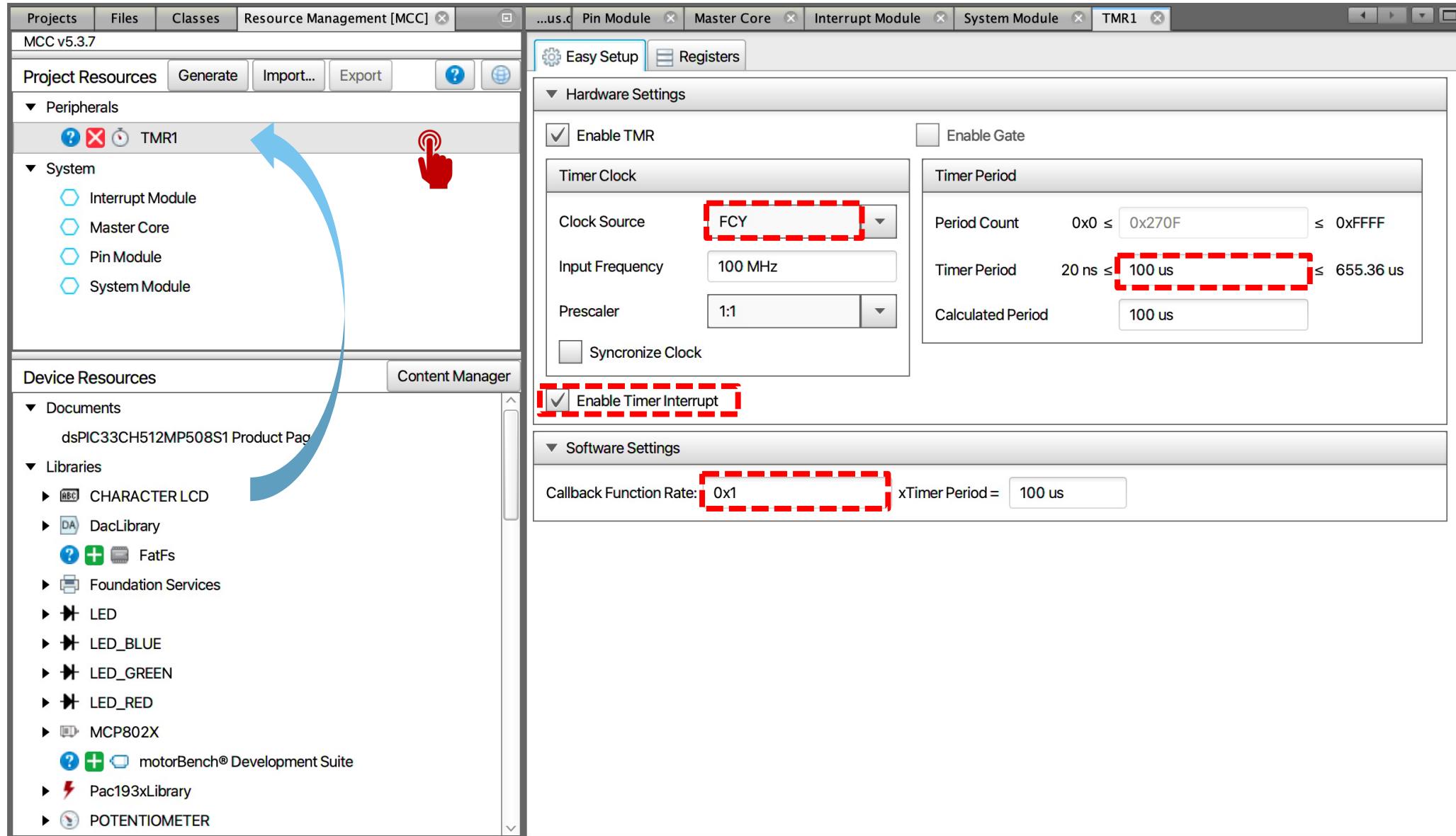
- The PLL Input Frequency (FPLL) must be in the range of 8 MHz to 64 MHz
- The PFD Input Frequency (FPFD) must be in the range of 8 MHz to (Fvco/16) MHz

The VCO Output Frequency (Fvco) must be in the range of 400 MHz to 1600 MHz

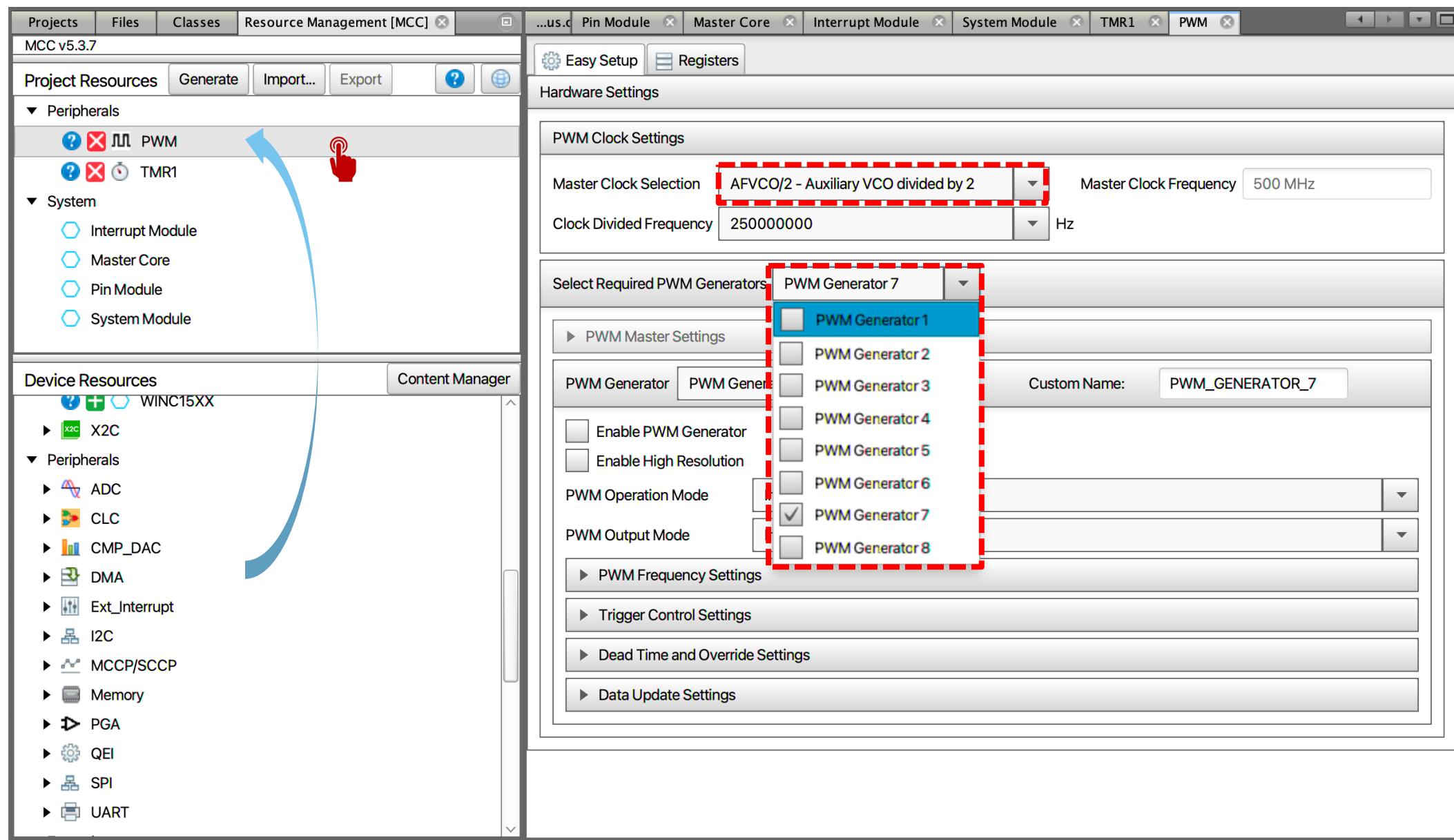
FIGURE 6-4: MASTER/SLAVE CORE PLL AND VCO DETAIL

Note 1: From Master and Slave core shared oscillator source.
 2: Clock option for PWM.
 3: Clock option for ADC.
 4: Clock option for DAC.

MCC Drivers – Timer1 (100 us)



MCC Drivers – PWM7H – 250KHz & Fixed 40% Duty Cycle



MCC Drivers – PWM7H – 250KHz & Fixed 40% Duty Cycle

The screenshot shows the MCC Driver software interface for a dsPIC33CH Dual-Core DSC. The main window displays the configuration for PWM Generator 7, including:

- PWM Generator 7** (selected in the dropdown)
- Custom Name:** PWM_GENERATOR_7
- Enable PWM Generator**: checked
- Enable High Resolution**: checked
- PWM Operation Mode**: Independent Edge
- PWM Output Mode**: Independent
- PWM Frequency Settings**:
 - PWM Input Clock Selection**: 500000000 Hz
 - Period**: Use Master Period (unchecked)
 - Requested Frequency**: 61.02864 kHz ≤ 250 kHz ≤ 29.41176471 MHz
 - Calculated Frequency**: 250 kHz
 - Requested Period**: 34 ns ≤ 4 us ≤ 16.3858 us
 - Calculated Period**: 4 us
 - Duty Cycle**: Use Master Duty Cycle (unchecked)
 - PWM Duty Cycle**: 0 % ≤ 40 ≤ 100 %
 - Phase**
 - PWM Phase**: 0 ns ≤ 0 ns ≤ 16.3838 us
- Trigger Control Settings**
- Dead Time and Override Settings**
- Data Update Settings**

The right panel shows the **Trigger Control Settings** and **PWM** register configuration:

- Trigger Control Settings**:
 - PWM Start of Cycle Control**
 - Start of Cycle Trigger**: Self-trigger
 - Trigger Output Selection**: EOC event
 - ADC Trigger**
 - ADC Trigger 1**: Trigger A Compare
 - ADC Trigger 2**: None
 - Timing Constraints** (for Trigger A Compare):
 - Trigger A Compare 0 ns ≤ 0 ns ≤ 16.3838 us
 - Trigger B Compare 0 ns ≤ 0 ns ≤ 16.3838 us
 - Trigger C Compare 0 ns ≤ 0 ns ≤ 16.3838 us
- PWM Register Configuration** (PG7IOCONH):
 - CAPSRC**: Software
 - DTCMPSEL**: PCI Sync Logic
 - PENH**: enabled
 - PENL**: enabled
 - PMOD**: Independent
 - POLH**: Active-low (highlighted with a red dashed box)
 - POLL**: Active-high

For P-channel
- Register PG7IOCONL** (0x0):
 - OLDAT**: 0x0

MCC Drivers – ADC/AN13 for Vout Feedback

The screenshot shows the MCC v5.3.7 software interface with the following details:

- Project Resources:** ADC1 is selected.
- Hardware Settings:**
 - Enable ADC:** Checked.
 - ADC Clock:**
 - Conversion Clock Source:** PLL VCO/4 (highlighted with a red dashed box).
 - Conversion Time:** 224.2 ns.
 - Target Shared Core Sampling Time:** 40 ns.
 - Calculated Shared Core Sampling Time:** 40 ns.
- Selected Channels:** A table showing core assignments and trigger sources. The row for Ccore1 is highlighted with a red dashed box, and the "Slave PWM7 Trigger1" entry is highlighted with a red box.

Core	Enable	Core Channel	Pin Name	Custom Name	Trigger Source	Compare	Interrupt
Ccore0	<input type="checkbox"/>	S1AN0	RA3		None	None	
Ccore1	<input checked="" type="checkbox"/>	S1ANA1	RC1	channel_S1ANA1	Slave PWM7...	None	<input checked="" type="checkbox"/>
Shared	<input type="checkbox"/>	S1AN10	RC0		None	None	
Shared	<input type="checkbox"/>	S1AN11	RC6		None	None	
Shared	<input type="checkbox"/>	S1AN12	RC7		None	None	
Shared	<input type="checkbox"/>	S1AN13	RD10		Slave PWM7...	None	
Shared	<input type="checkbox"/>	S1AN14	RD12		None	None	
Shared	<input type="checkbox"/>	S1AN15	RA1		None	None	

MCC Drivers – ADC/AN13 for Vout Feedback

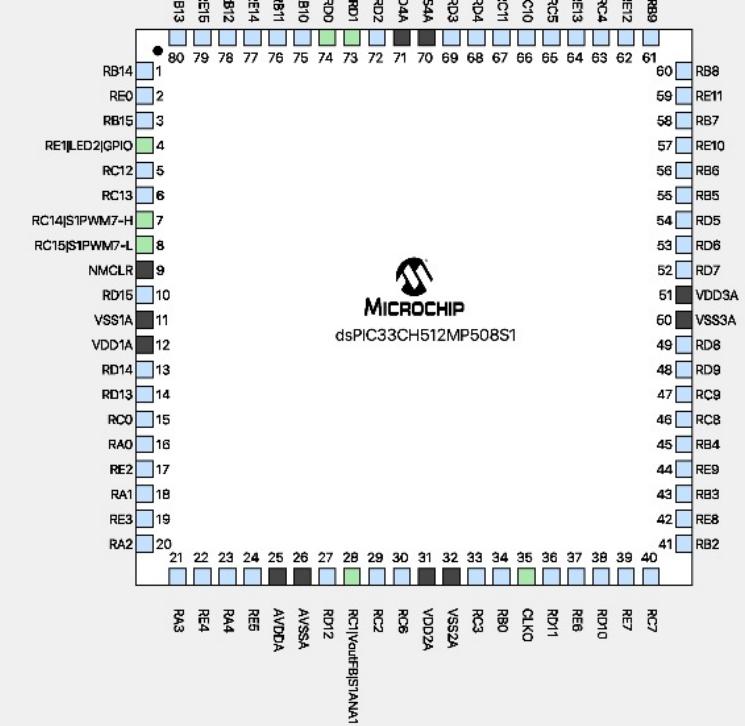
The screenshot shows the MCC Driver software interface for the ADC1 module. The main window title is "ADC1". The left pane shows the "Registers" tab selected, displaying configuration for ADCON2L (0x0), ADCON3H (0xC280), and ADCORE registers (0x300). The right pane shows detailed settings for ADCORE0H (0x300) and ADCORE1H (0x300). The "SHRADCS" field in ADCON2L and the "ADCS" fields in both ADCORE registers are highlighted with red boxes.

Slave - ADC Shared Core			
ADC Module Clock Source	400	MHz	PLL VOC/4
ADCON2L.SHRADCS	2		Shared ADC Core Input Clock Divider bits
ADCON3H.CLKDIV	3		ADC Module Clock Source Divider bits
Fadcore	66.67	MHz	Clock Source / SHRADCS / CLKDIV
Tadcore	15.00	ns	1/Fadcore \geq 14.3ns

MCC Drivers – Pin Management

The screenshot shows the Microchip Configuration Center (MCC) software interface. The top menu bar includes Projects, Files, Classes, Services, Resource Man..., Start Page, Pin Module, Master Core, System Module, Interrupt Module, ADC1, and MCC v5.3.7. The main workspace is divided into several sections:

- Project Resources:** A tree view showing Peripherals (ADC1, PWM, TMR1) and System components (Interrupt Module, Master Core, Pin Module, System Module). A red hand cursor is over the Pin Module node.
- Selected Package: TQFP80:** A table for pin configuration. The columns are Pin Name, Module, Function, Custom Na..., Start High, Analog, Output, WPU, WPD, OD, and IOC. Rows include RC1 (ADC1, S1ANA1, VoutFB), RC14 (PWM, S1PWM7-H), RC15 (PWM, S1PWM7-L), and RE1 (Pin Module, GPIO, LED2). A red dashed box highlights the columns from Start High to IOC.
- Device Resources:** A sidebar with Content Manager and sections for Documents (dsPIC33CH512MP508S1 Product Page) and Libraries (CHARACTERLCD, DacLibrary, FatFs, Foundation Services, LED, LED_BLUE, LED_GREEN).
- Bottom Left:** A dashboard for POW201_Slave showing Project Type: Application - Configuration: default, Device: dsPIC33CH512MP508S1, and Compiler Toolchain: Curiosity/Starter Kits (PKOB4) (1.11.1054).
- Bottom Right:** A Pin Manager: Grid View window showing Port C and Port D pin assignments for various modules like S1PWMWEB, S1PWMEC, S1PBMED, and Pin Module (GPIO, TMR1).



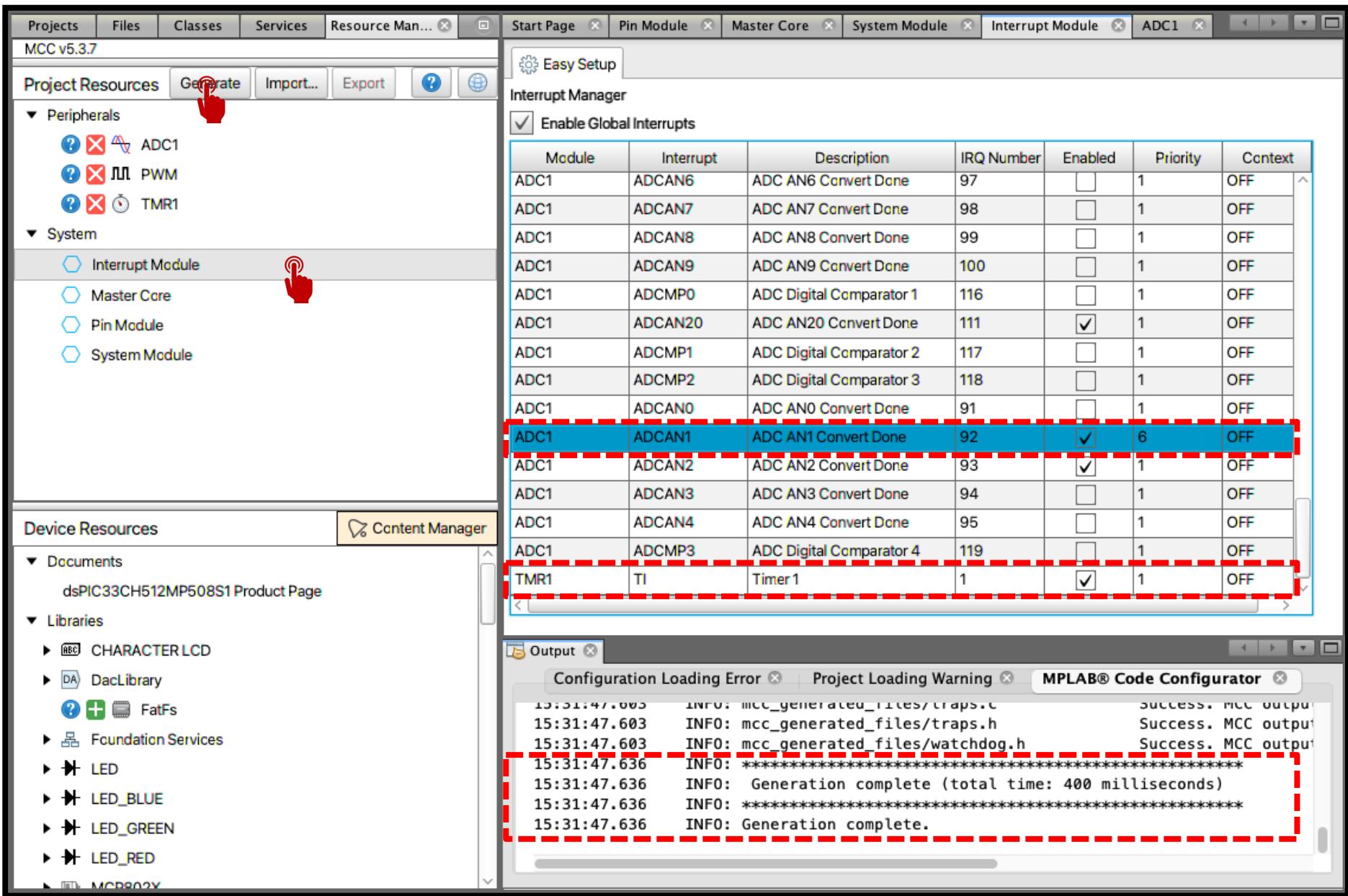
dsPIC33CH512MP508S1

MCC Drivers – Interrupt Management & Code Generating

The screenshot shows the MCC v5.3.7 software interface with the 'Interrupt Module' tab selected. The left sidebar displays project resources like Peripherals (ADC1, PWM, TMR1) and System modules (Interrupt Module, Master Core, Pin Module, System Module). A red hand cursor is hovering over the 'Interrupt Module' entry in the System section. The main area is titled 'Easy Setup' and contains the 'Interrupt Manager'. It features a table with columns: Module, Interrupt, Description, IRQ Number, Enabled, Priority, and Context. A checkbox labeled 'Enable Global Interrupts' is checked. The table lists various interrupt configurations, with rows 92 and 119 highlighted in blue and surrounded by a red dashed border, indicating they are selected or being edited.

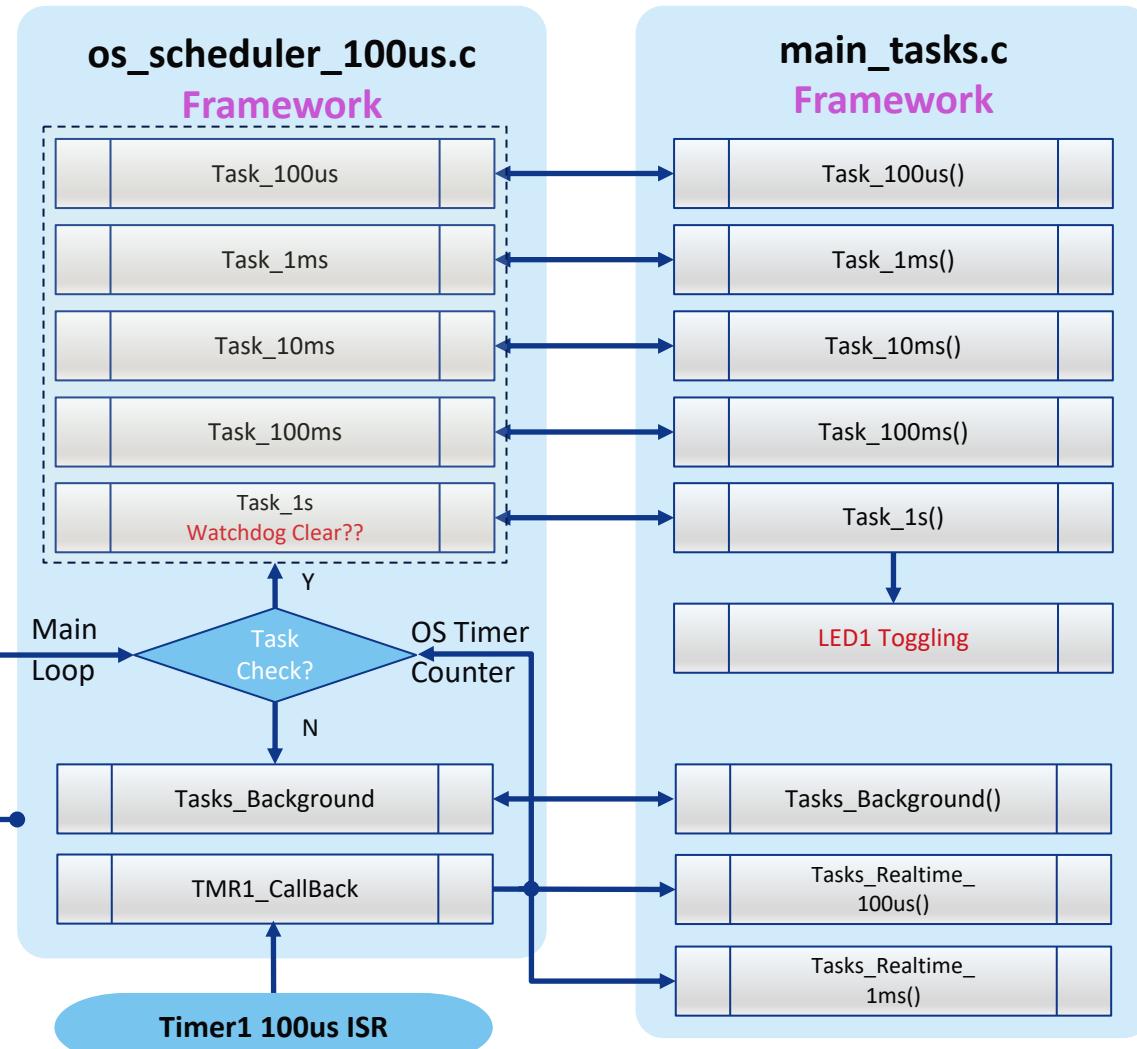
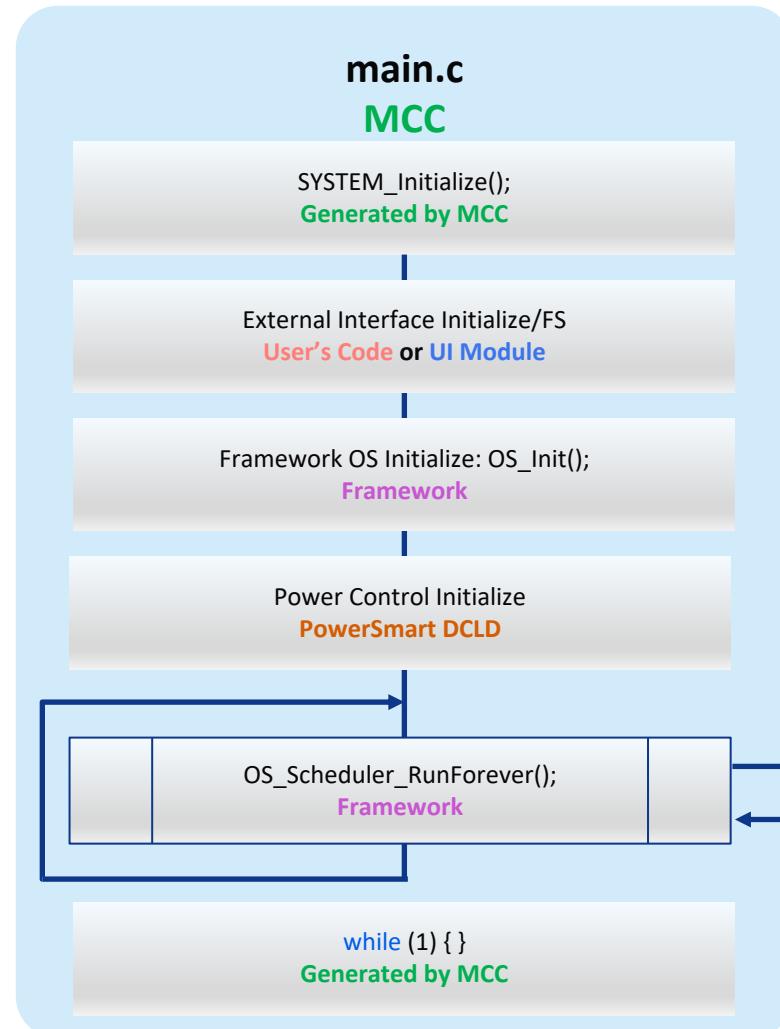
Module	Interrupt	Description	IRQ Number	Enabled	Priority	Context
ADC1	ADCAN6	ADC AN6 Convert Done	97	<input type="checkbox"/>	1	OFF
ADC1	ADCAN7	ADC AN7 Convert Done	98	<input type="checkbox"/>	1	OFF
ADC1	ADCAN8	ADC AN8 Convert Done	99	<input type="checkbox"/>	1	OFF
ADC1	ADCAN9	ADC AN9 Convert Done	100	<input type="checkbox"/>	1	OFF
ADC1	ADCMP0	ADC Digital Comparator 1	116	<input type="checkbox"/>	1	OFF
ADC1	ADCAN20	ADC AN20 Convert Done	111	<input checked="" type="checkbox"/>	1	OFF
ADC1	ADCMP1	ADC Digital Comparator 2	117	<input type="checkbox"/>	1	OFF
ADC1	ADCMP2	ADC Digital Comparator 3	118	<input type="checkbox"/>	1	OFF
ADC1	ADCAN0	ADC AN0 Convert Done	91	<input type="checkbox"/>	1	OFF
ADC1	ADCAN1	ADC AN1 Convert Done	92	<input checked="" type="checkbox"/>	6	OFF
ADC1	ADCAN2	ADC AN2 Convert Done	93	<input checked="" type="checkbox"/>	1	OFF
ADC1	ADCAN3	ADC AN3 Convert Done	94	<input type="checkbox"/>	1	OFF
ADC1	ADCAN4	ADC AN4 Convert Done	95	<input type="checkbox"/>	1	OFF
ADC1	ADCMP3	ADC Digital Comparator 4	119	<input type="checkbox"/>	1	OFF
TMR1	TI	Timer1	1	<input checked="" type="checkbox"/>	1	OFF

MCC Drivers – Interrupt Management & Code Generating

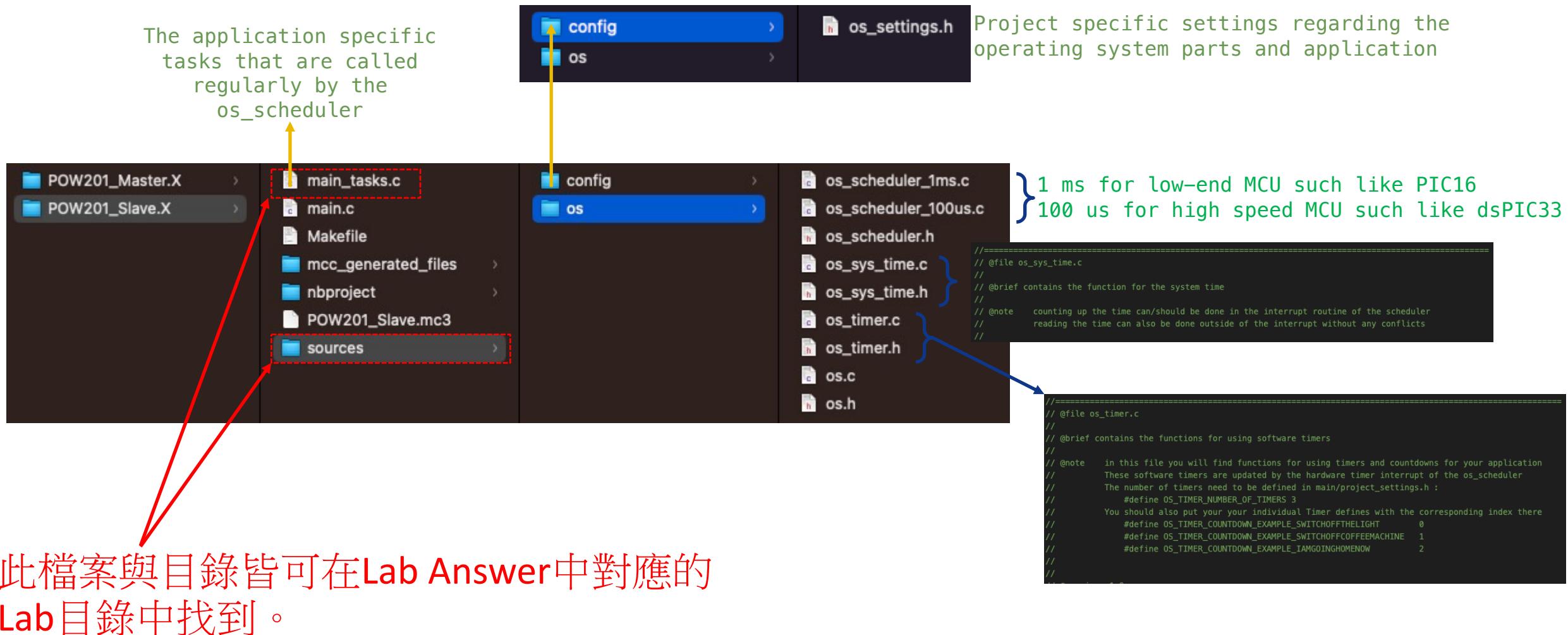


SMPS Firmware Framework

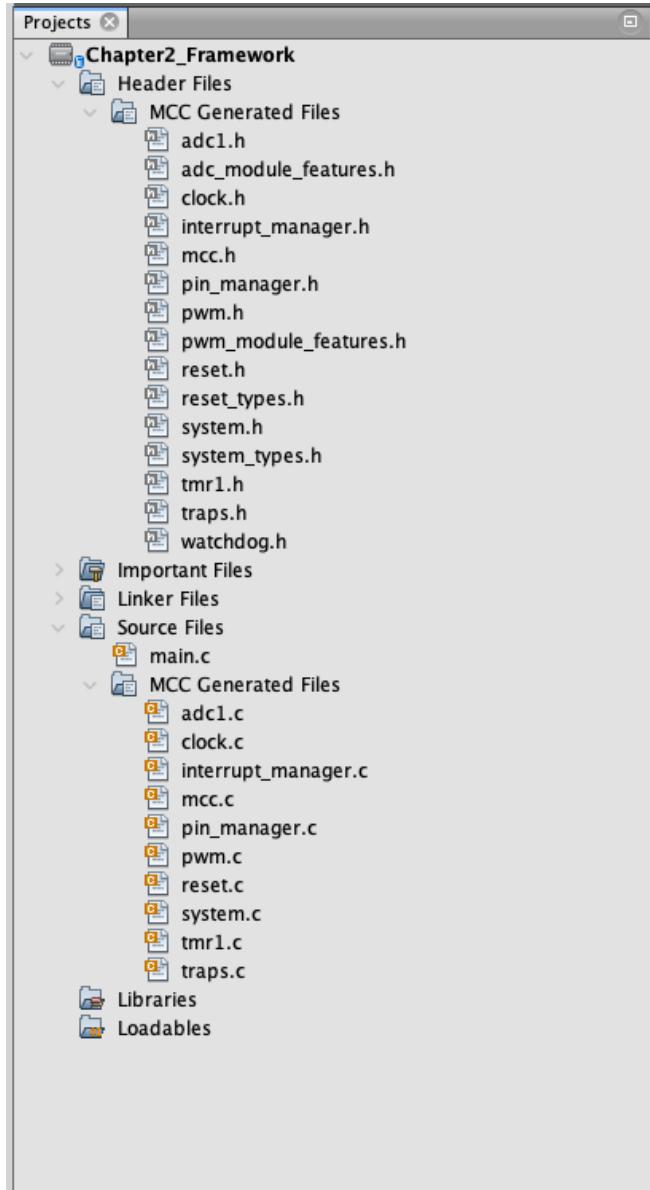
Firmware Structure



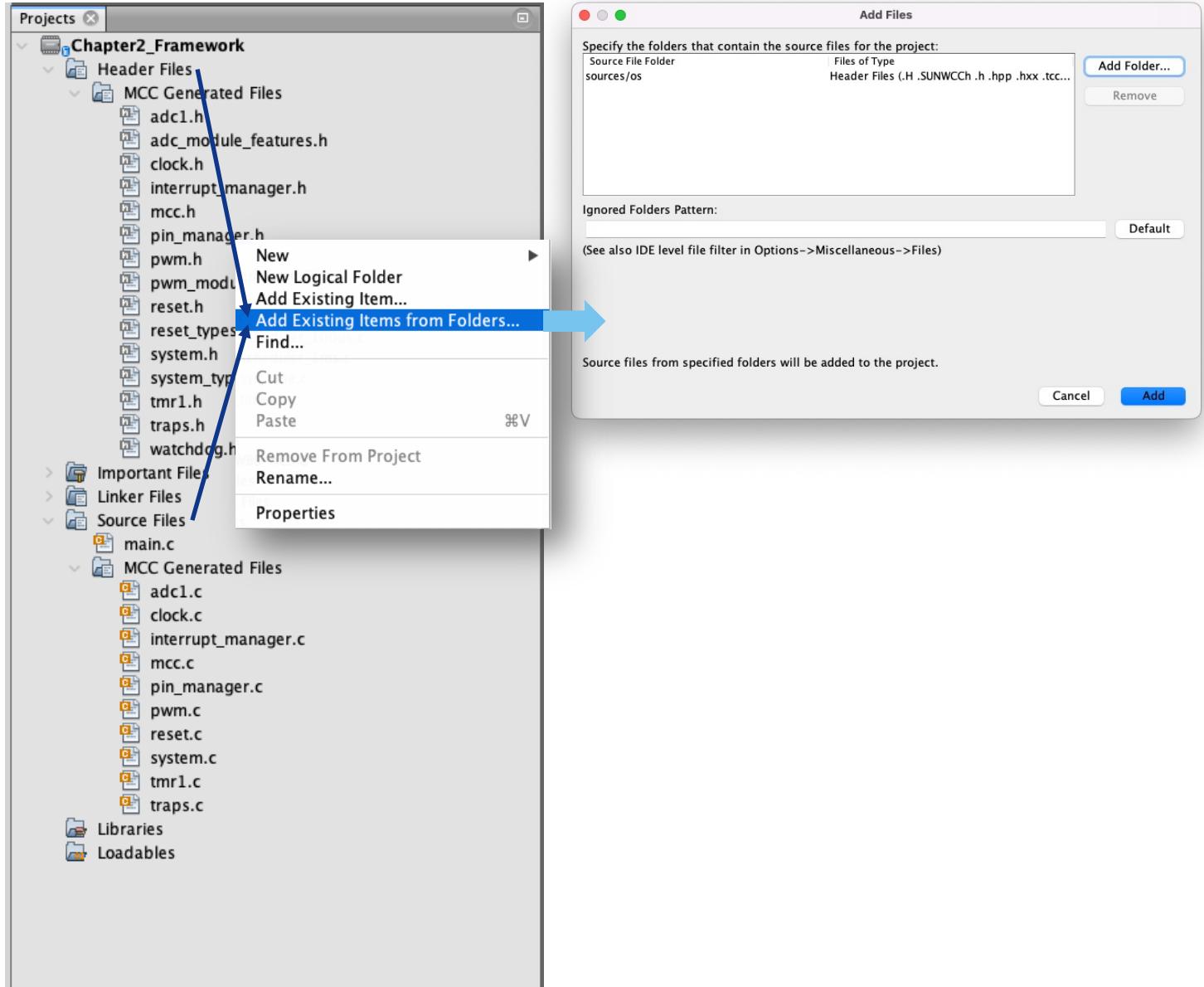
OS Scheduler for Slave



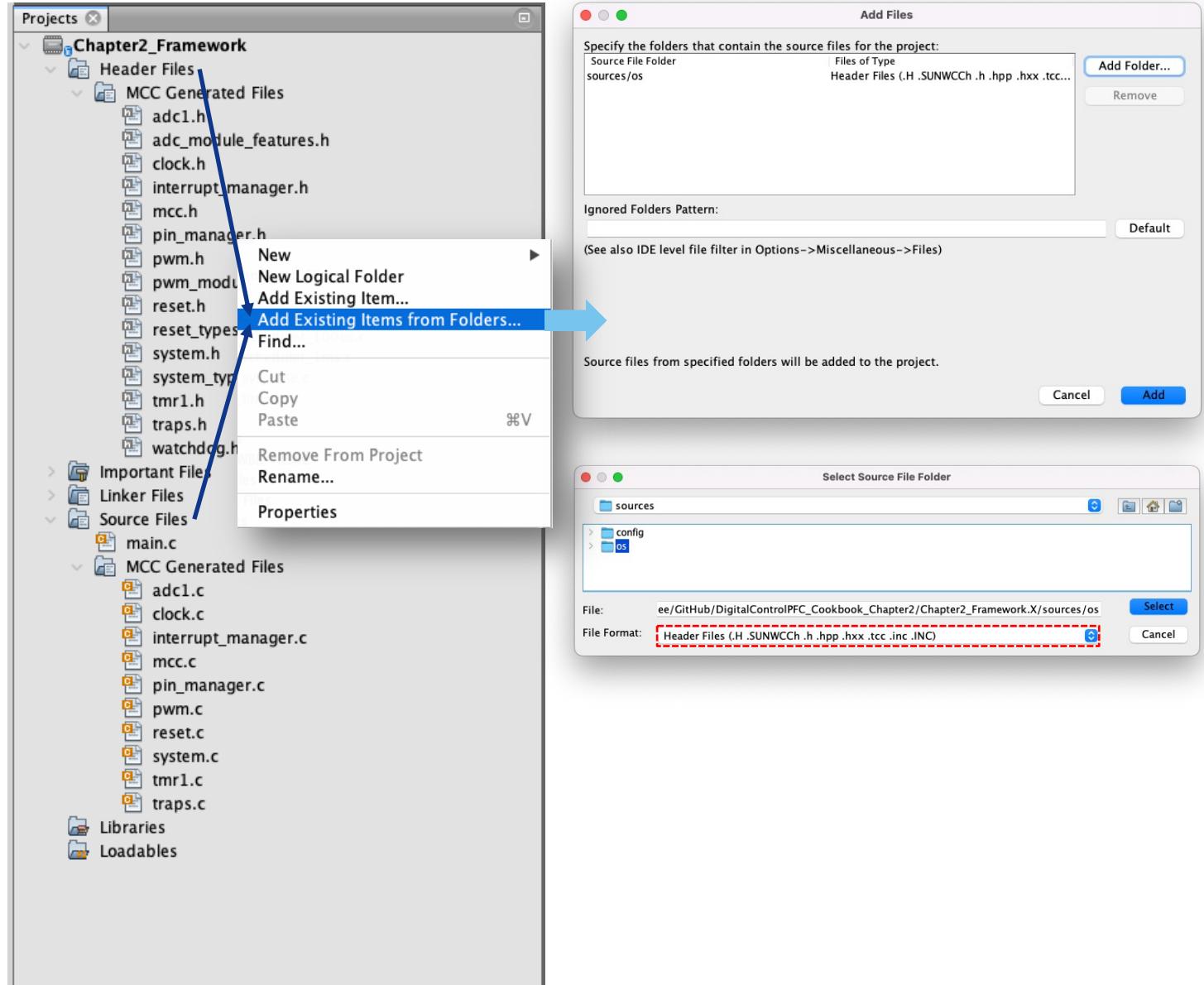
Import OS_Scheduler



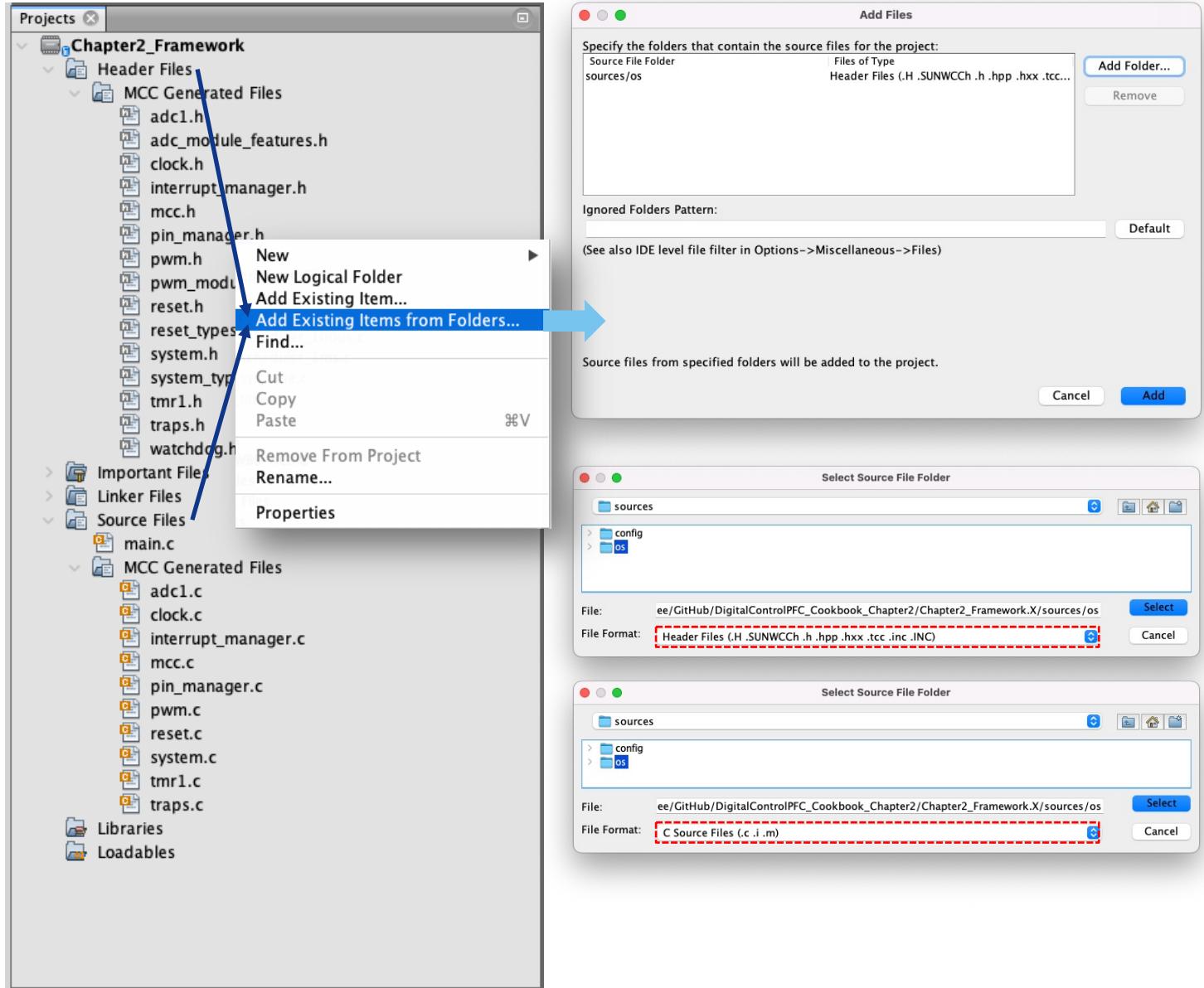
Import OS_Scheduler



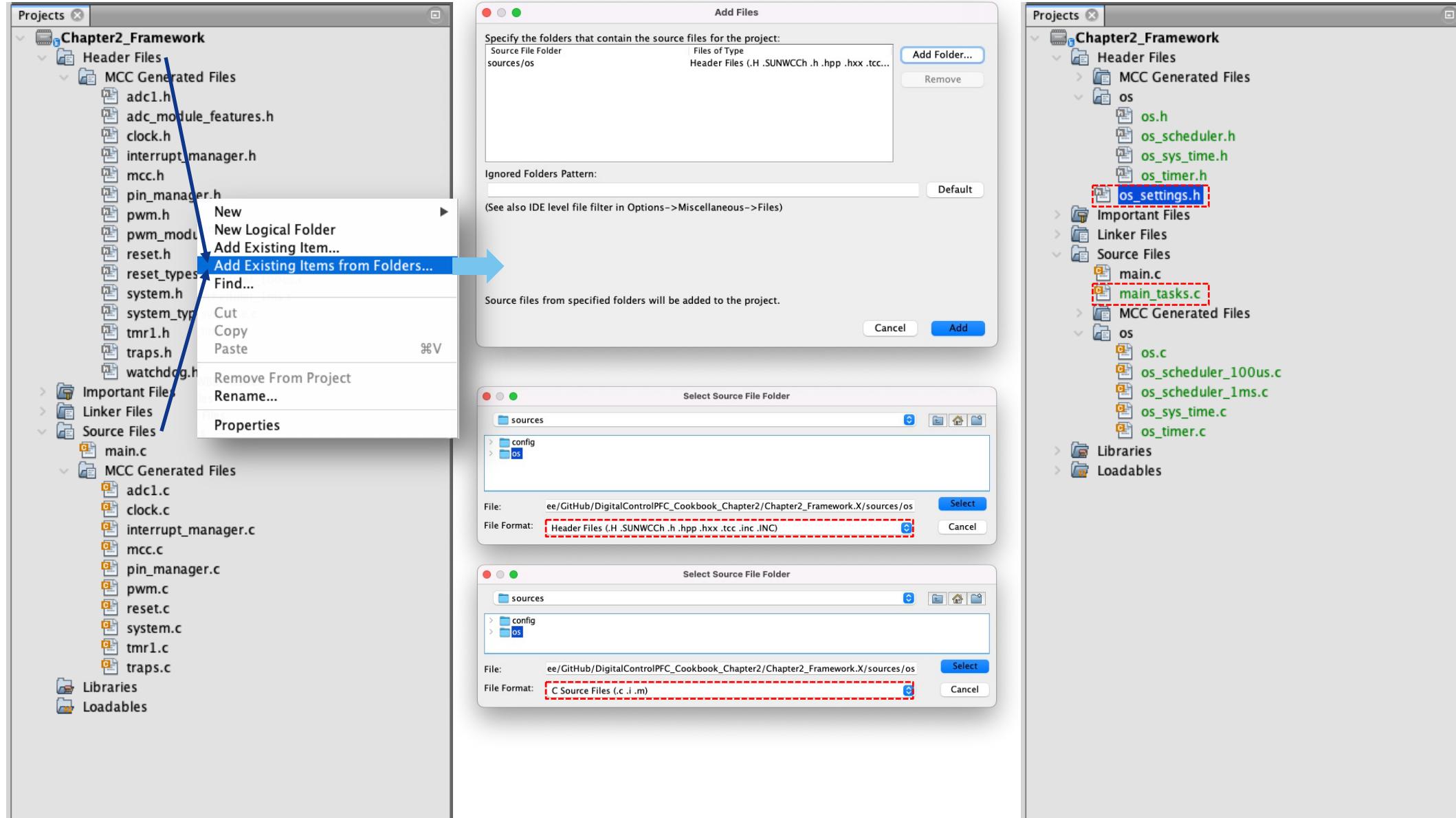
Import OS_Scheduler



Import OS_Scheduler



Import OS_Scheduler

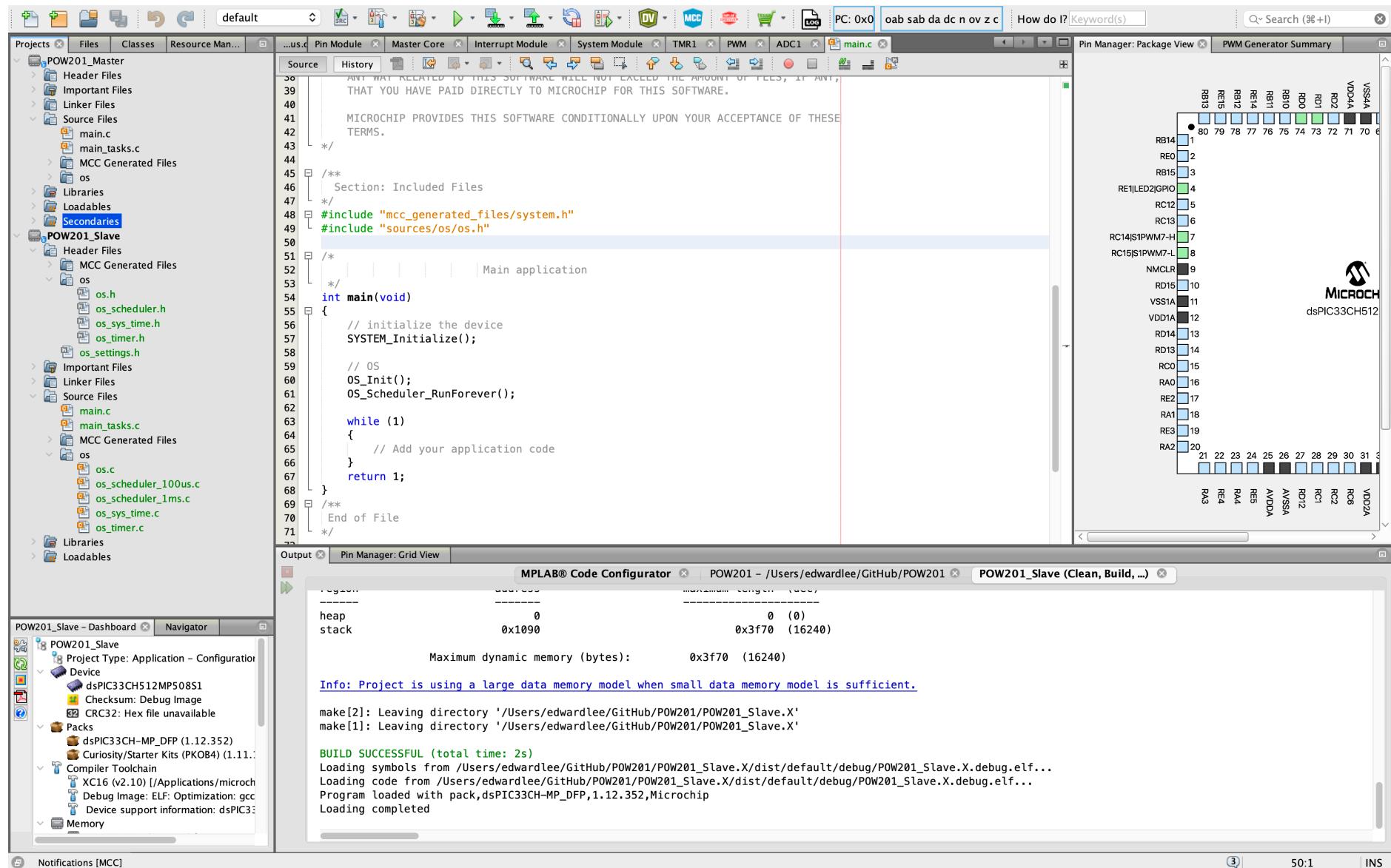


Main & Tasks

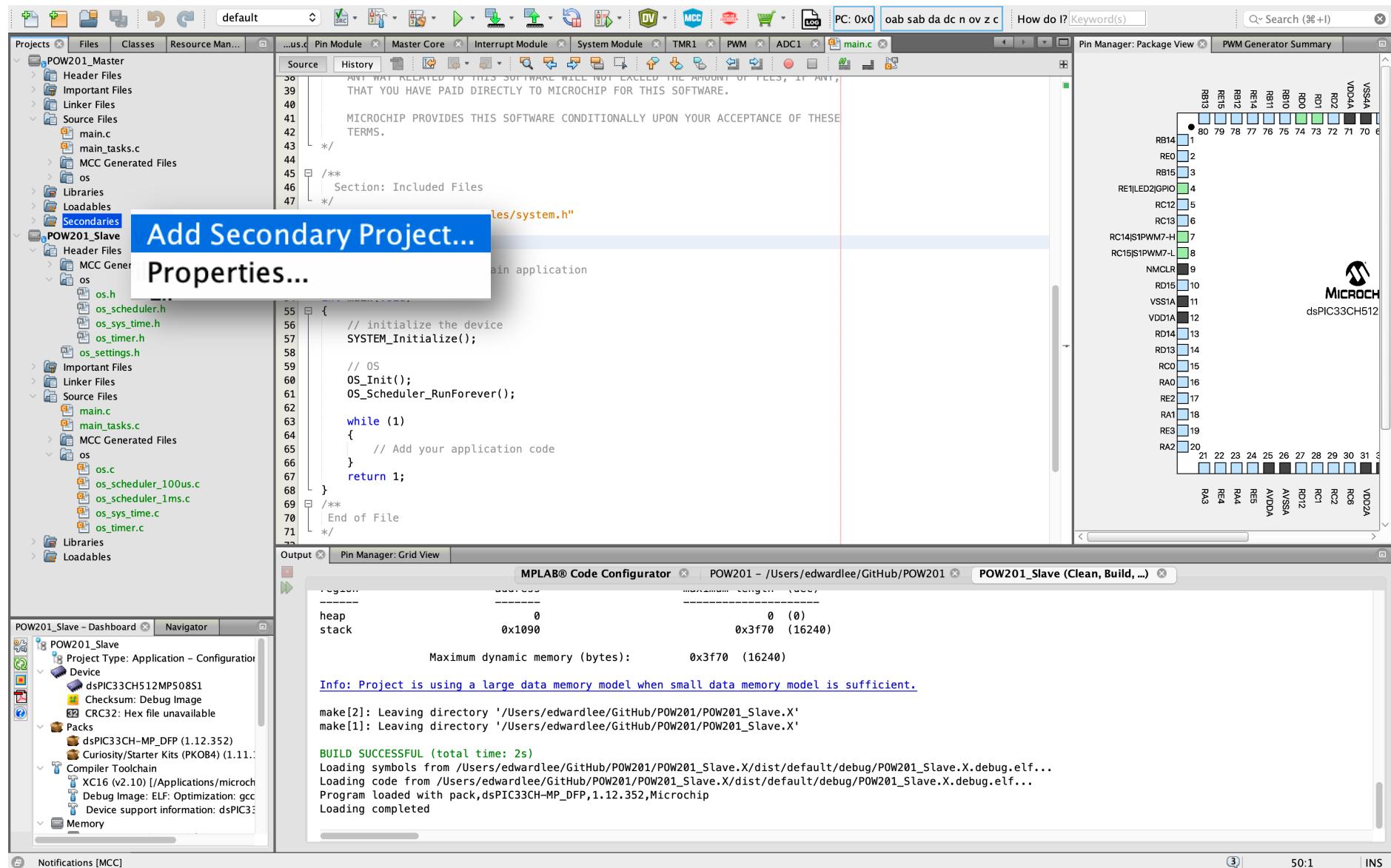
The screenshot shows a software development environment with two code editors:

- main.c**: This file contains the main application code. It includes headers for system and OS files, initializes the device, and runs an OS loop. A red dashed box highlights the OS initialization code: `OS_Init();` and `OS_Scheduler_RunForever();`.
- main_tasks.c**: This file contains a task function named `Tasks_1s`. It is annotated with comments explaining its purpose: "Tasks_1s gets called every second, put your things in it that need to be called every second" and "there could be some jitter here because it is not called directly by a timer". The function body contains a call to `LED2_Toggle()`, which is also highlighted with a red dashed box.

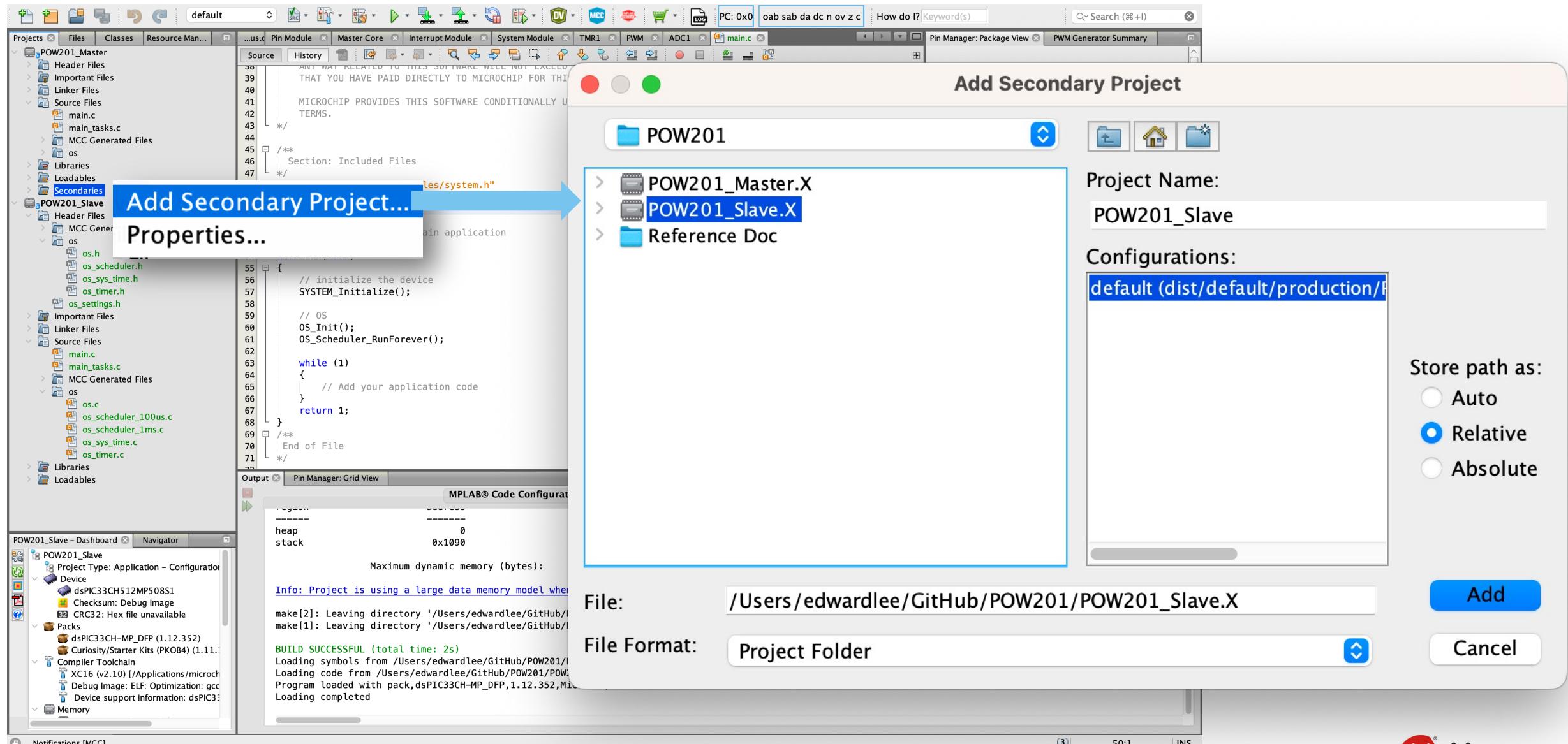
Back to Master and Add Secondary Project



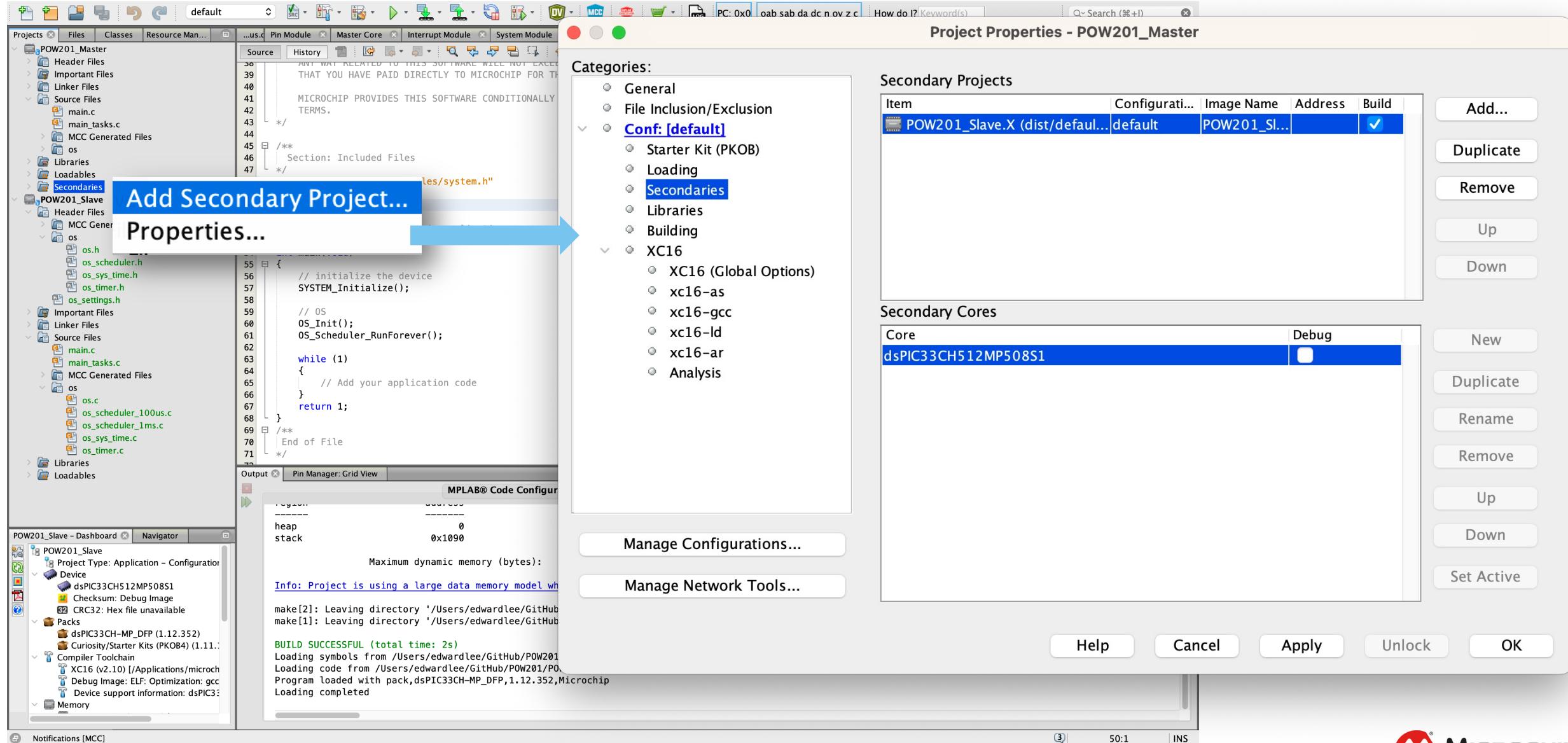
Back to Master and Add Secondary Project



Back to Master and Add Secondary Project



Back to Master and Add Secondary Project

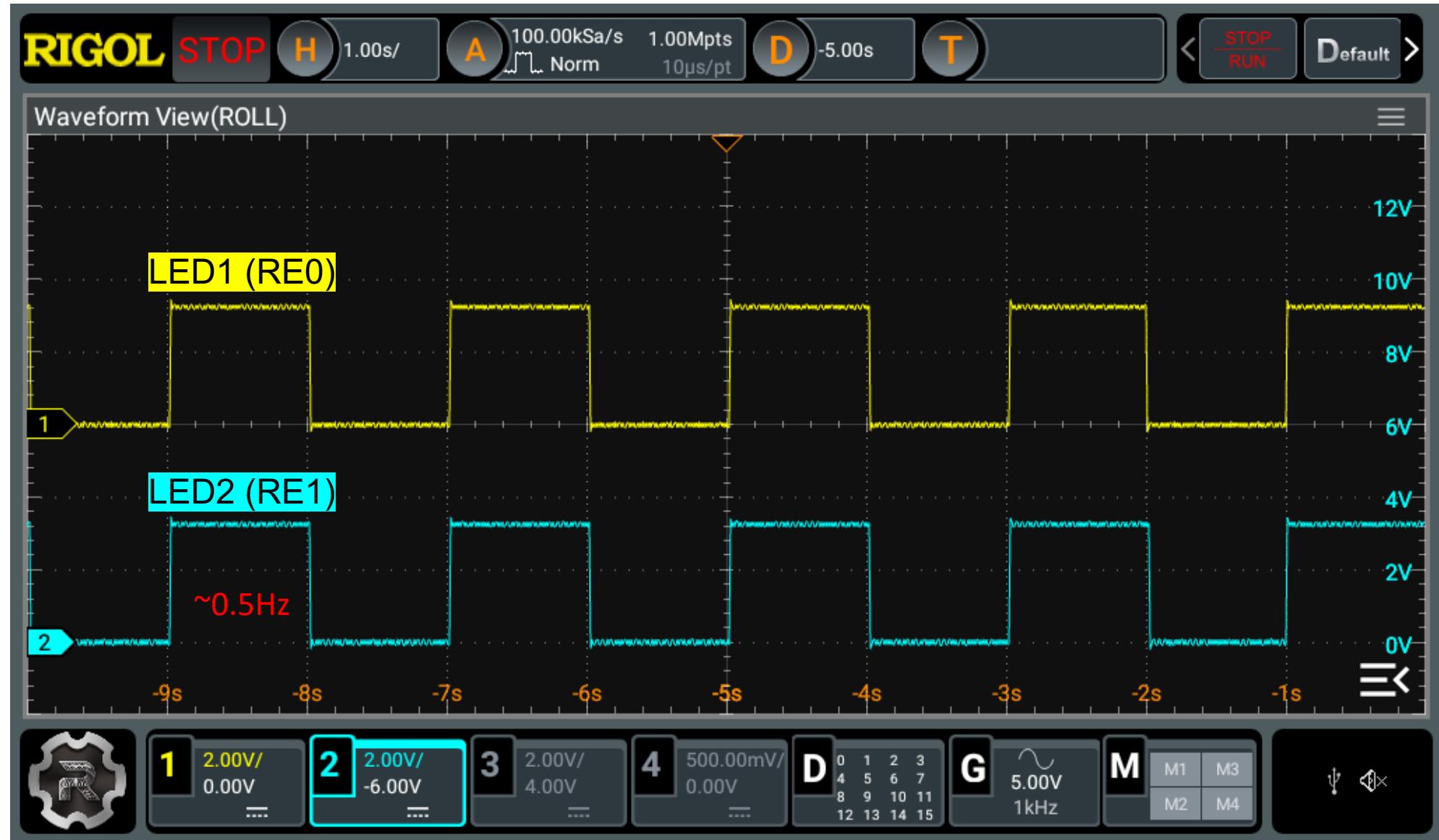


Back to Master and “Program&Start” Slave Core

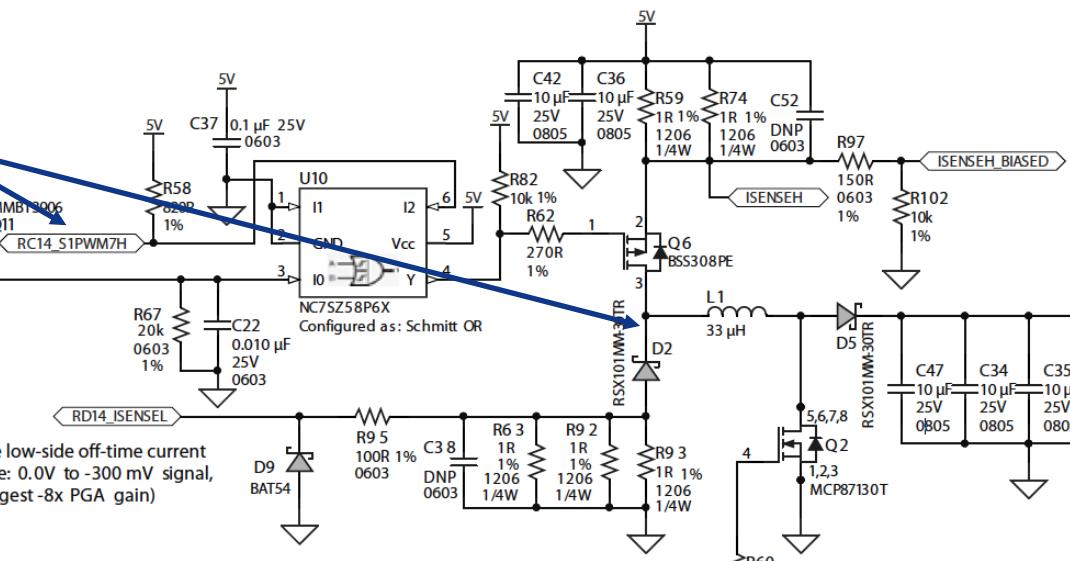
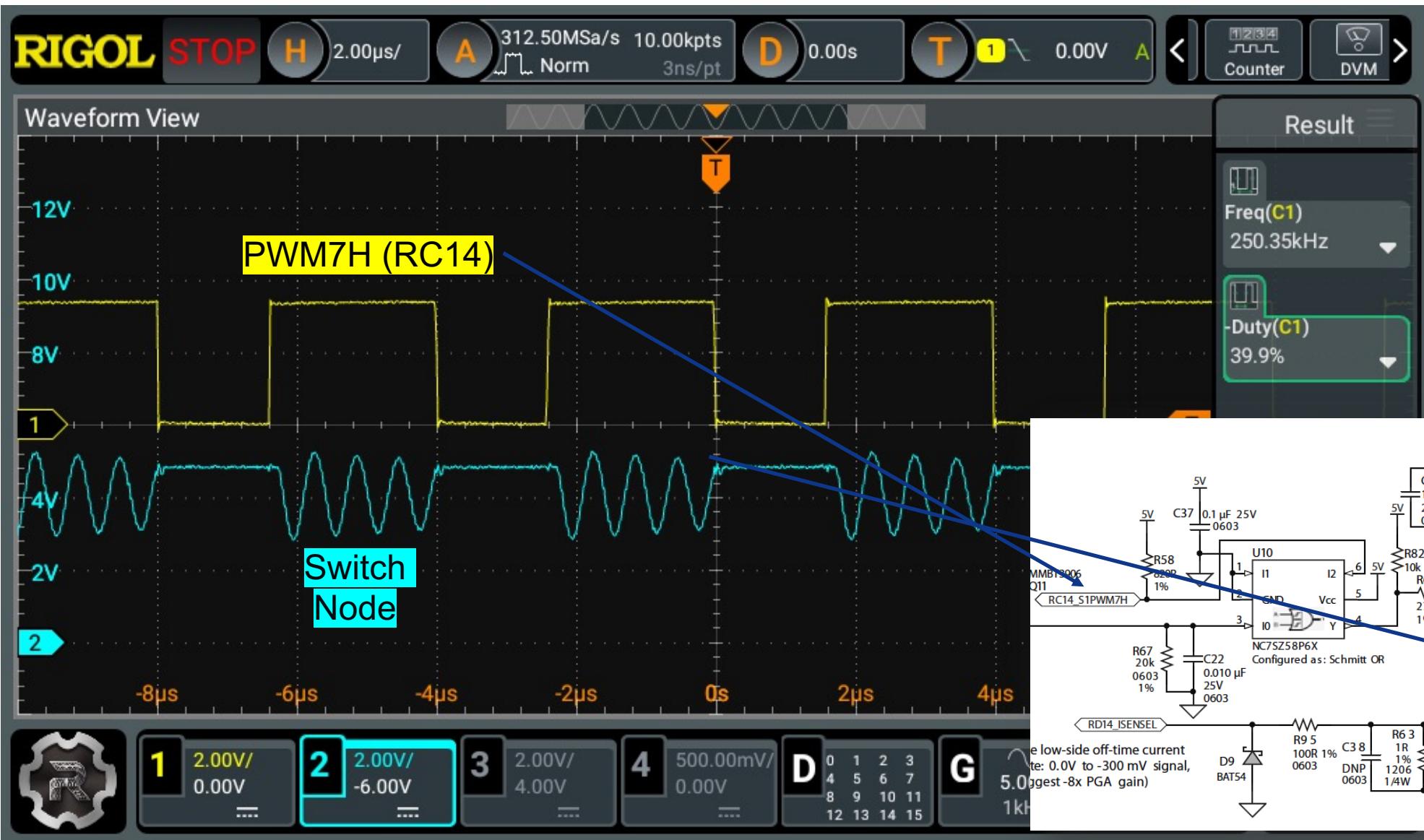
```
Projects X Files Classes Services
POW201_Master
  > Header Files
  > Important Files
  > Linker Files
  > Source Files
    > main.c
    > main_tasks.c
    > MCC Generated Files
    > os
    > Libraries
    > Loadables
  > Secondaries
    > POW201_Slave
POW201_Slave
  > Header Files
  > Important Files
  > Linker Files
  > Source Files
    > main.c
    > main_tasks.c
    > MCC Generated Files
    > os
    > Libraries
    > Loadables

main.c
Source History
47  /*
48  #include "mcc_generated_files/system.h"
49  #include "mcc_generated_files/slave1.h"
50  #include "sources/os/os.h"
51
52  /*
53  |   |   |   |   |   | Main application
54  */
55  int main(void)
56  {
57      // initialize the device
58      SYSTEM_Initialize();
59
60      //Program and enable slave
61      SLAVE1_Program();
62      SLAVE1_Start();
63
64      // OS
65      OS_Init();
66      OS_Scheduler_RunForever();
67
68      while (1)
69      {
70          // Add your application code
71      }
72      return 1;
73  }
74 /**
75 | End of File
76 */
77
```

0.5 Hz LED1 & LED2 Toggle



PWM7H (RC14) - 40% Duty (Active Low)

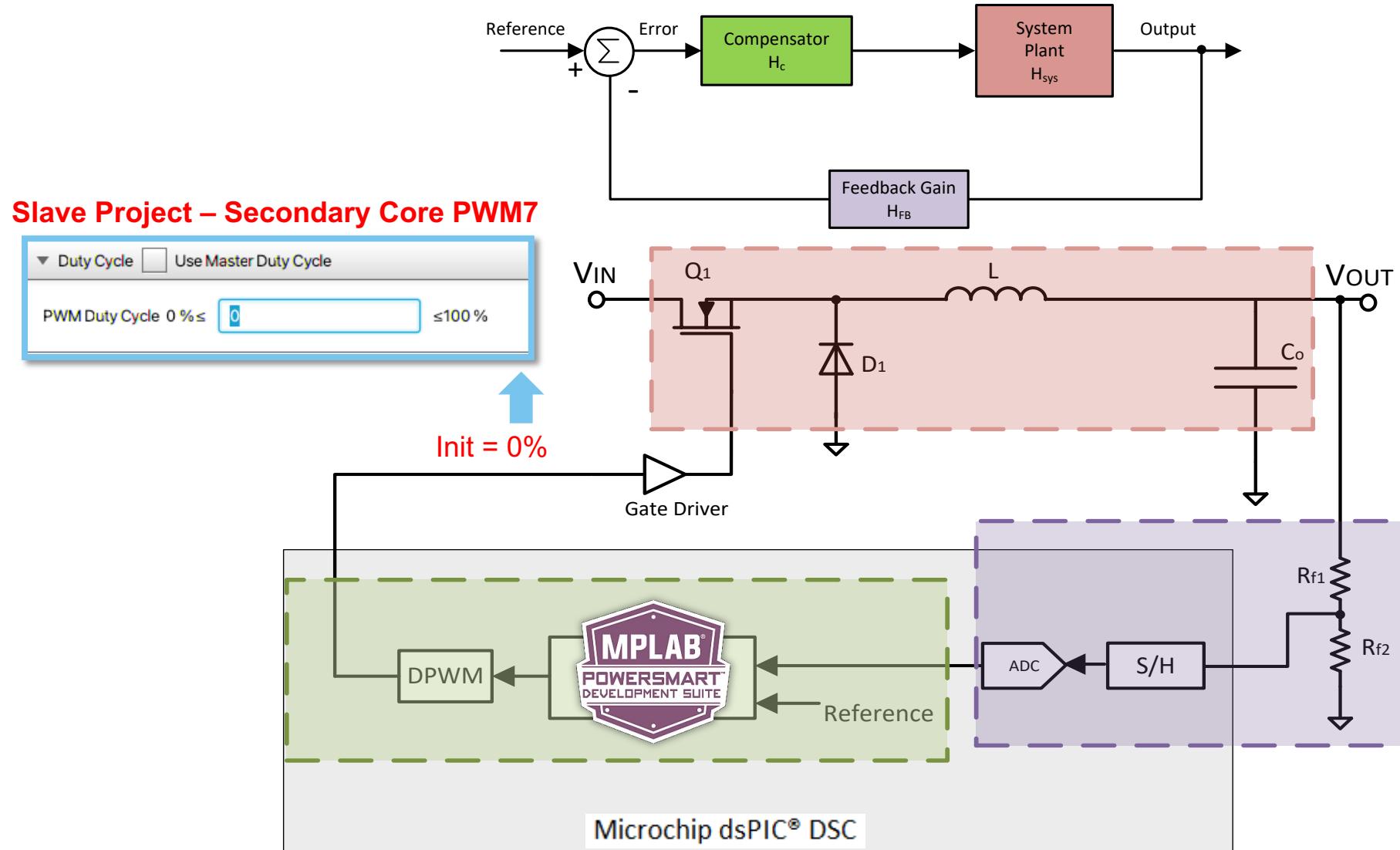


Digital SMPS Exercises – Part-II

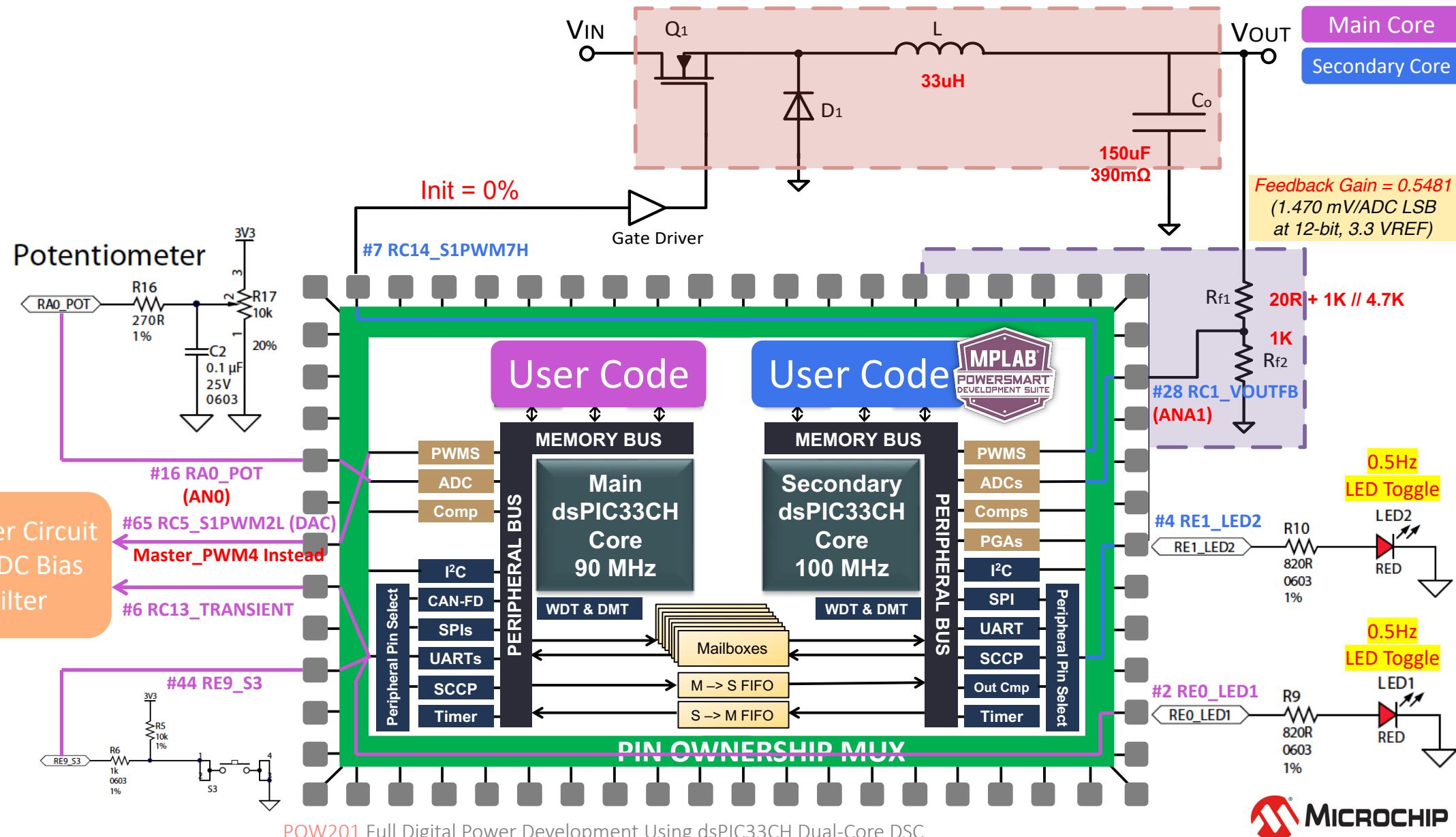
Plant Measurement

Lab3

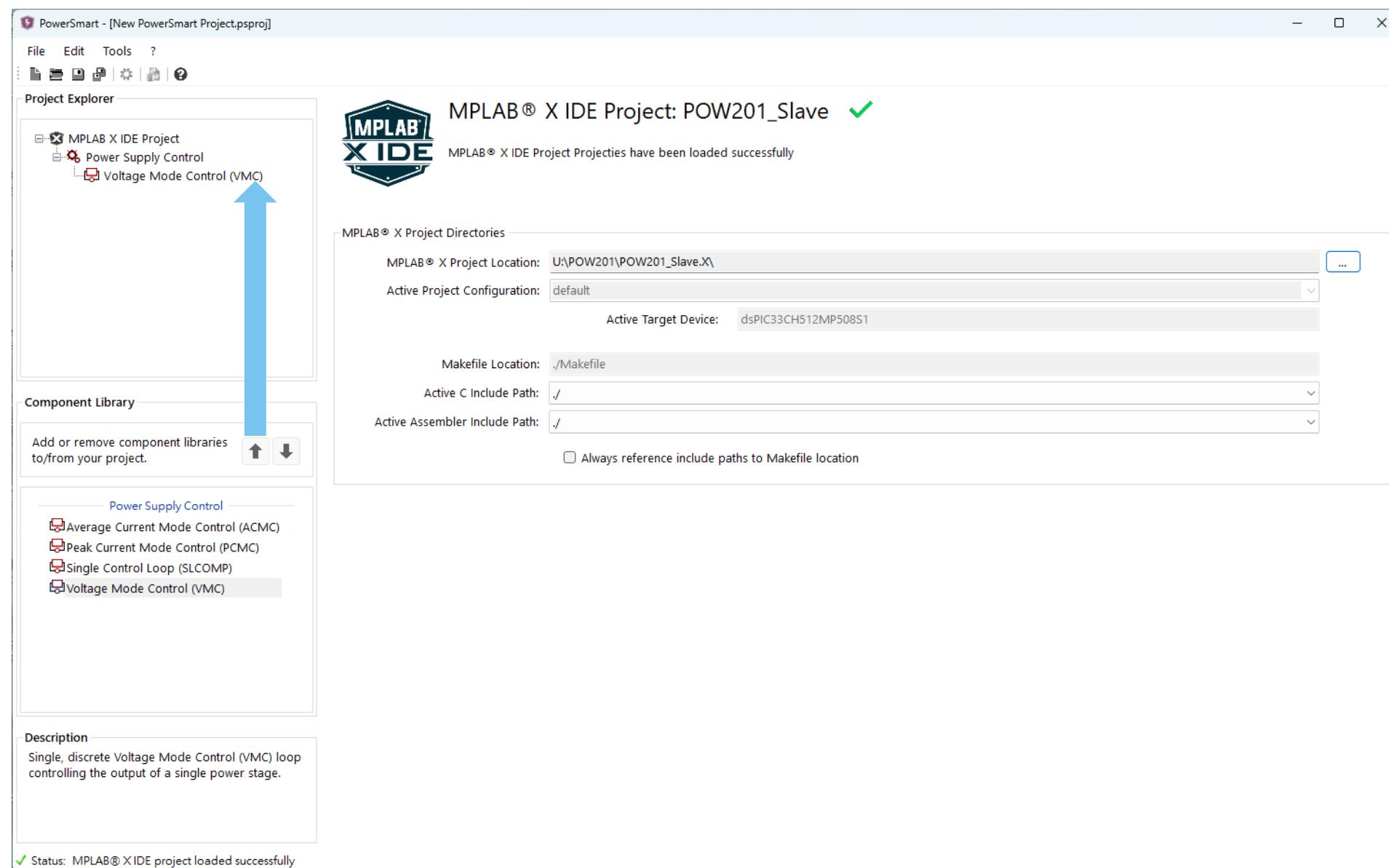
Lab #3: Plant Measurement Using PowerSmart™



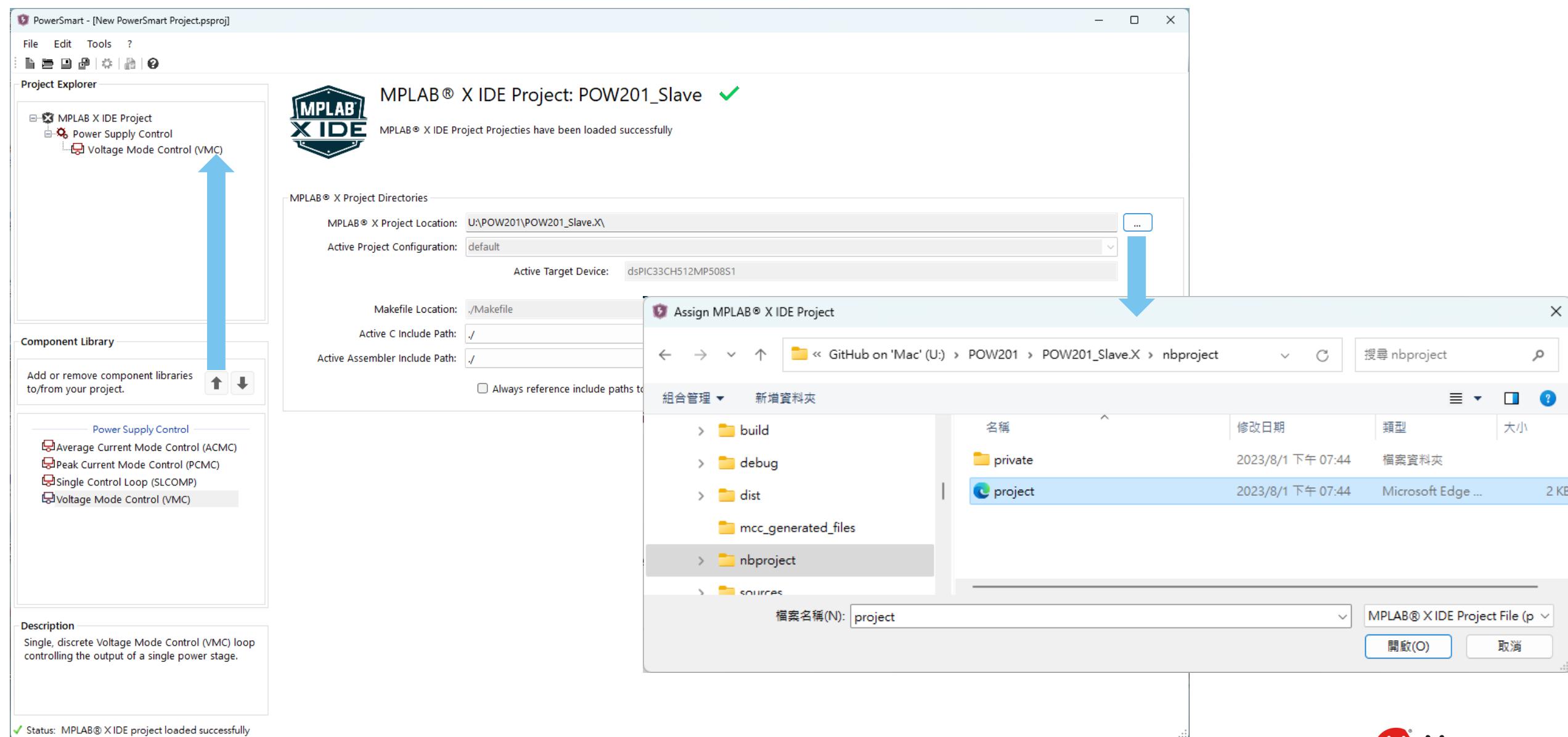
Lab #3: Plant Measurement Using PowerSmart™



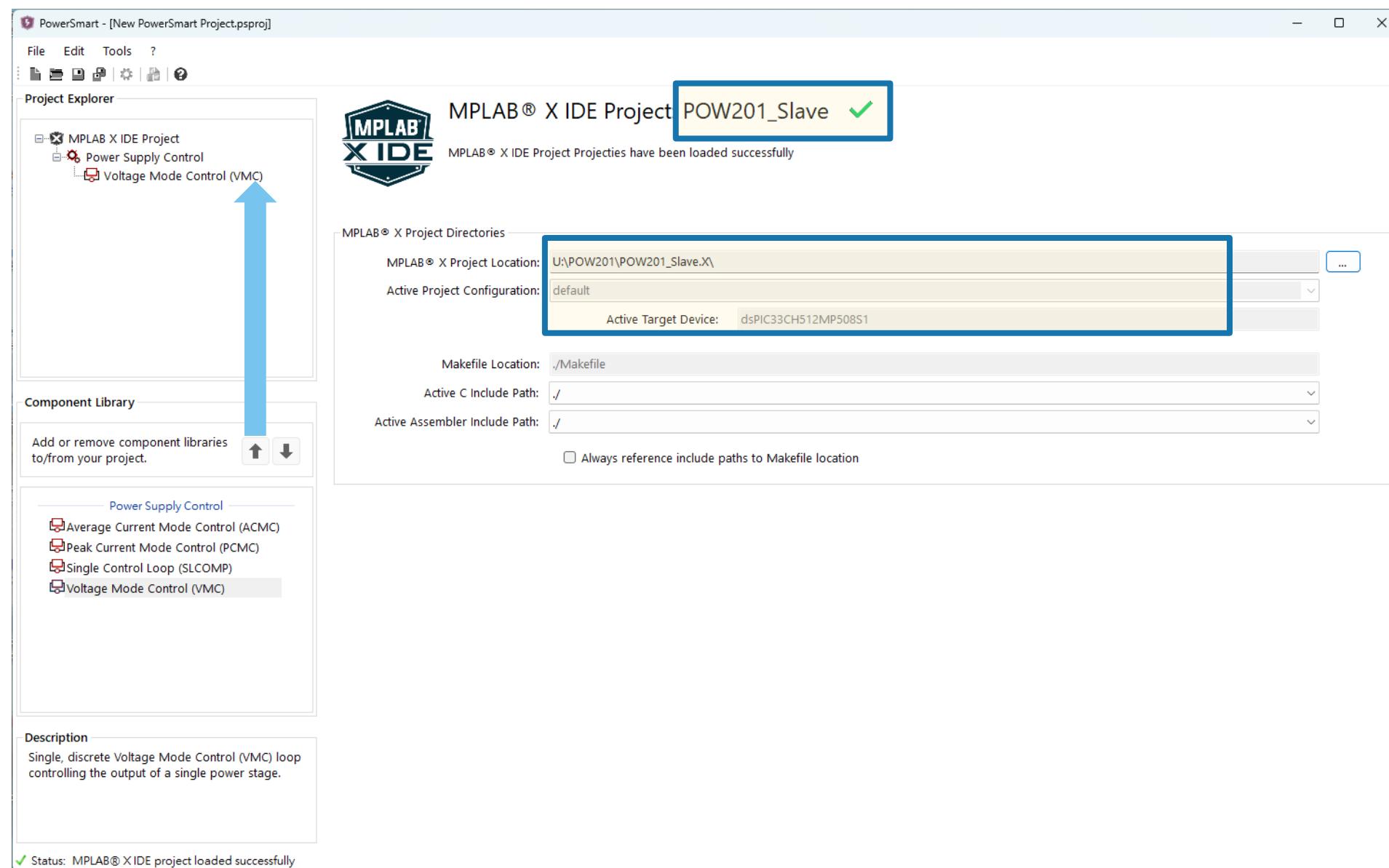
Add Control Mode for the Slave Project



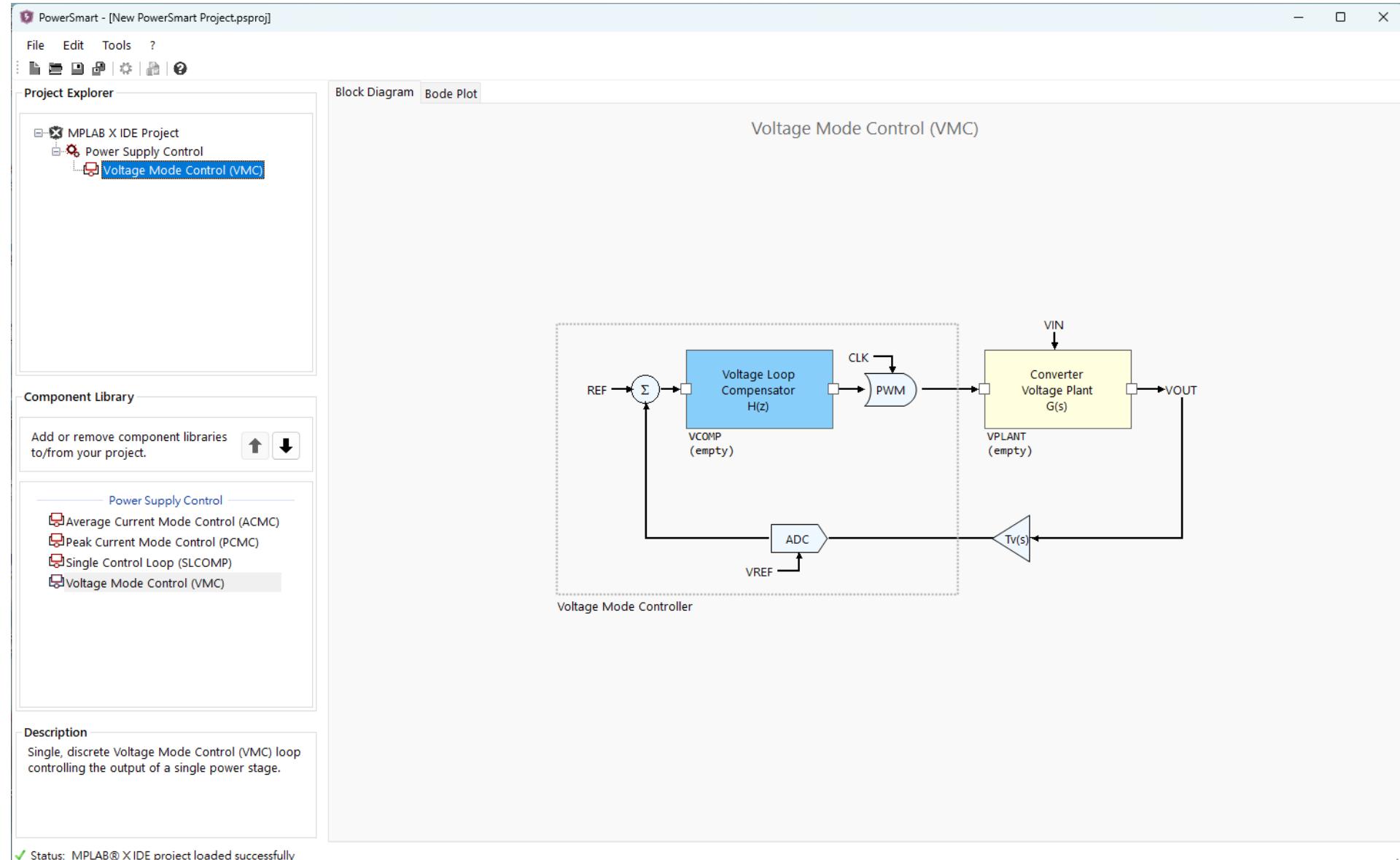
Add Control Mode for the Slave Project



Add Control Mode for the Slave Project



Switch to DCLD Window



Switch to DCLD Window

PowerSmart - [New PowerSmart Project.psproj]

File Edit Tools ?

Project Explorer

- MPLAB X IDE Project
- Power Supply Control
- Voltage Mode Control (VMC)

Block Diagram Bode Plot

Voltage Mode Control

Voltage Mode Controller

REF → Σ → VCOMP (empty) → H(z) → PWM → CLK → ADC → VREF

Component Library

Add or remove component libraries to/from your project.

- Power Supply Control
 - Average Current Mode Control (ACMC)
 - Peak Current Mode Control (PCMC)
 - Single Control Loop (SLCOMP)
 - Voltage Mode Control (VMC)

Description

Single, discrete Voltage Mode Control (VMC) loop controlling the output of a single power stage.

Status: MPLAB® X IDE project loaded successfully

PowerSmart Digital Control Library Designer v1.9.15.709 - [New PowerSmart Project.psproj]

File View Tools ?

File & Function Label

Name Prefix: VCOMP

Controller Selection

Controller Type: 3P3Z - Discrete Type III Compensator

Scaling Mode: 1 - Single Bit-Shift Scaling

Input Gain

Input Data Resolution: 12 Bit

Input Signal Gain: 1.000000

Normalize Input Gain

Feedback Offset Compensation

Enable Singal Rectification Control

Compensation Filter Settings

Sampling Frequency: 250k Hz

Cross-over Frequency of Pole at Origin: 650 Hz

Pole 1: 86k Hz Zero 1: 3.2k Hz

Pole 2: 100k Hz Zero 2: 4.9k Hz

Bode Plot Settings

Frequency Domain Execution Time Block Diagram Source Code Output Info

Frequency: 0 Hz Magnitude: 0 dB Phase: 0 ° Phase Erosion 0

Magnitude/Gain

Min: -60 dB Max: 60 dB Div: 10 dB

Phase

Min: -180 ° Max: 180 ° Div: 30 °

Options

Unwrap Phase Show s-Domain

Magnitude/Gain

Phase

Options

Unwrap Phase Show s-Domain

Filter Coefficients

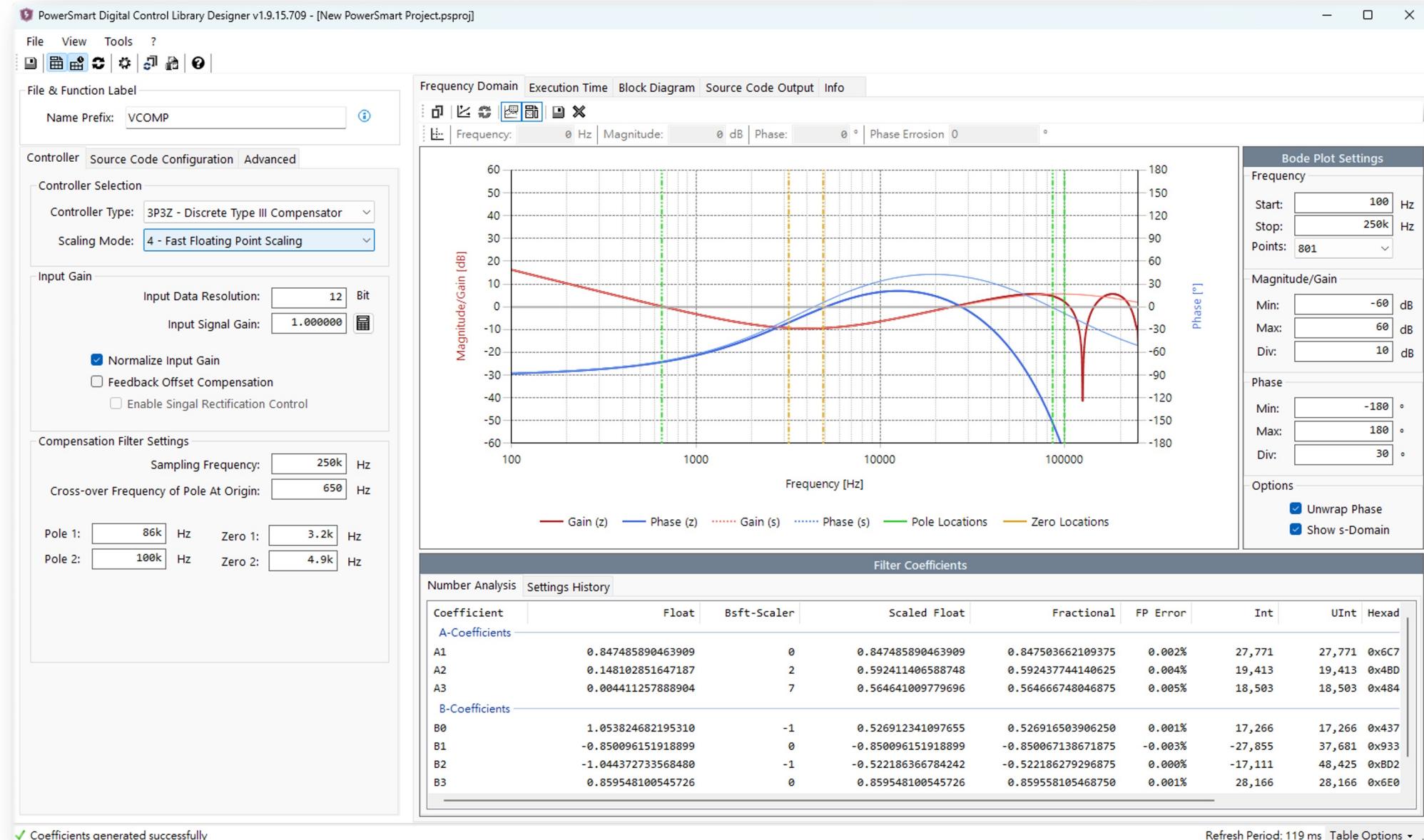
Number Analysis Settings History

Coefficient	Float	Bsft-Scaler	Scaled Float	Fractional
A-Coefficients				
A1	0.847485890463909	-1	0.423742945231954	0.423767089843758
A2	0.148102851647187	-1	0.074051425823594	0.074066162109375
A3	0.004411257888904	-1	0.00220562894452	0.002227783203125
B-Coefficients				
B0	1.053824682195310	-1	0.526916503906250	0.526916503906250
B1	-0.850096151918899	-1	-0.425048075959450	-0.425048075959450
B2	-1.044372733568480	-1	-0.522186366784242	-0.522186366784242
B3	0.859548100545726	-1	0.429774050272863	0.429774050272863

Coefficients generated successfully

Refresh Period: 31 ms Table Options ::

Config K_P (P-Term) Gain



Config K_P (P-Term) Gain

PowerSmart Digital Control Library Designer v1.0.0.0

Controller Source Code Configuration Advanced

Software Context Management

- Save/Restore Shadow Registers
- Save/Restore MAC Working Registers
- Save/Restore Accumulators
 - Save/Restore Accumulator A
 - Save/Restore Accumulator B
- Save/Restore DSP Core Configuration
- Save/Restore Core Status Register

Controller Selection

Controller Type: 3P3Z - Discrete Type I

Scaling Mode: 4 - Fast Floating Point

Input Gain

Input Data Resolution:

Input Signal Gain:

- Normalize Input Gain
- Feedback Offset Compensation
- Enable Singal Rectification

Compensation Filter Settings

Sampling Frequency:

Cross-over Frequency of Pole At Origin:

Pole 1: 86k Hz Zero 1:

Pole 2: 100k Hz Zero 2:

Basic Feature Extensions

- Store/Reload Result Accumulator
- Add DSP Core Configuration
- Add Enable/Disable Feature
 - Always read from source when disabled
- Add Error Normalization
- Add Automatic Placement of Primary ADC Trigger A
- Add Automatic Placement of Secondary ADC Trigger B

Automated Data Interface

Data Provider Sources

Anti-Windup

- Limit Control Loop Output to Positive Numbers
 - Anti Windup Limiter Number Range: -32768...32767
- Clamp Control Output Maximum
 - Generate Upper Saturation Status Flag Bit
- Clamp Control Output Minimum
 - Force Values below Minimum Threshold to Zero
 - Generate Lower Saturation Status Flag Bit

 Coefficients generated successfully

Source Code Configuration

Program Source Code Output Info

0 dB | Phase: 0 ° | Phase Erosion 0 °

Bode Plot Settings

Frequency

- Start: 100 Hz
- Stop: 250k Hz
- Points: 801

Magnitude/Gain

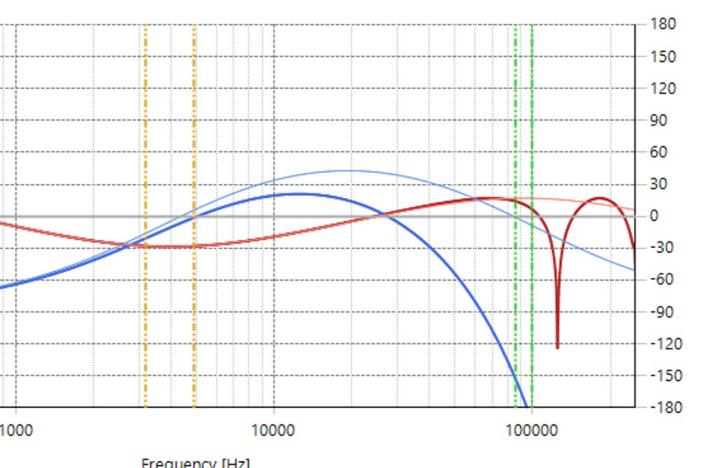
- Min: -60 dB
- Max: 60 dB
- Div: 10 dB

Phase

- Min: -180 °
- Max: 180 °
- Div: 30 °

Options

- Unwrap Phase
- Show s-Domain



Legend: Pole (z) Gain (s) Phase (s) Pole Locations Zero Locations

Filter Coefficients

Coef	Bsft-Scaler	Scaled Float	Fractional	FP Error	Int	UInt	Hexad
909	0	0.847485890463909	0.847503662109375	0.002%	27,771	27,771	0x6C7
187	2	0.592411406588748	0.592437744140625	0.004%	19,413	19,413	0x4BD
904	7	0.564641009779696	0.564666748046875	0.005%	18,503	18,503	0x484
310	-1	0.526912341097655	0.526916503906250	0.001%	17,266	17,266	0x437
899	0	-0.850096151918899	-0.850067138671875	-0.003%	-27,855	37,681	0x933
480	-1	-0.522186366784242	-0.522186279296875	0.000%	-17,111	48,425	0xBD2
726	0	0.859548100545726	0.859558105468750	0.001%	28,166	28,166	0x6E0

Refresh Period: 119 ms Table Options

Config K_P (P-Term) Gain

The screenshot illustrates the configuration process for the P-term gain in the PowerSmart Digital Control Library Designer v1.0. It shows three main windows:

- Controller Selection Window:** Shows the "Source Code Configuration" tab selected. It includes fields for "Name Prefix: VCOMP", "Controller Type: 3P3Z - Discrete Type I", and "Scaling Mode: 4 - Fast Floating Point".
- Source Code Configuration Window:** Shows the "Advanced" tab selected. It includes sections for "Software Context Management" (checkboxes for Save/Restore Shadow Registers, MAC Working Registers, Accumulators, DSP Core Configuration, and Core Status Register), "Used Resources" (WREG 0,1,2,3,4,5,6,8,10/ACC AB), and "Basic Feature Extensions" (checkboxes for Add DSP Core Configuration, Add Enable/Disable Feature, Add Error Normalization, and Anti-Windup options like Clamp Control Output Maximum and Minimum). A blue arrow points from the Controller Selection window to this window.
- Development Tools Window:** Shows the "Advanced" tab selected. It includes fields for "Nominal Feedback Level: 2245", "Nominal Control Output: 10560", "Fractional: 0.587890625", and "Scaler: -3". It also lists several checkboxes for "Add" options related to AGC, Modulation, and User Extensions, with "Optimize AGC Modulation Factor Accuracy" checked. A "Please Note:" section at the bottom states: "Execution time of user functions being called during the control loop execution is not included in the Control Timing Analysis".

A red dashed box highlights the "Source Code Configuration" tabs in both the Controller Selection and Source Code Configuration windows. A blue arrow indicates the flow from the Controller Selection window to the Source Code Configuration window. A green checkmark at the bottom left of the Controller Selection window indicates "Coefficients generated successfully".

Config K_P (P-Term) Gain

PowerSmart Digital Control Library Designer v1.0.0

Controller **Source Code Configuration** **Advanced**

Software Context Management

- Save/Restore Shadow Registers
- Save/Restore MAC Working Registers
- Save/Restore Accumulators
 - Save/Restore Accumulator A
 - Save/Restore Accumulator B
- Save/Restore DSP Core Configuration
- Save/Restore Core Status Register

Input Gain

Input Data Resolution:

Input Signal Gain:

- Normalize Input Gain
- Feedback Offset Compensation
- Enable Singal Rectification

Compensation Filter Settings

Sampling Frequency:

Pole 1: Hz Zero 1:

Pole 2: Hz Zero 2:

Automated Data Interface

Data Provider Sources

- Anti-Windup
 - Limit Control Loop Output to Positive Numbers

Anti Windup Limiter Number Range: -32768...32767
 - Clamp Control Output Maximum
 - Generate Upper Saturation Status Flag Bit
 - Clamp Control Output Minimum
 - Force Values below Minimum Threshold to Zero
 - Generate Lower Saturation Status Flag Bit

Coefficients generated successfully

Source Code Configuration

Use P-Term Loop Controller for Plant Measurements

- Nominal Feedback Level:
- Nominal Control Output:
- Fractional:
- Scaler:

Enable Feedback Loop Gain Modulation (AGC)

Add Enable/Disable Adaptive Gain Control (AGC)

Add Observer Function Call before Modulation

Optimize AGC Modulation Factor Accuracy

Add User Extensions

- Start of Control Loop
- After Reading Source
- Before Anti-Windup
- Before Writing to Target
- End of Control Loop
- Cascade Function Call

Please Note:
Execution time of user functions being called during the control loop execution is not included in the Control Timing Analysis

Nominal Feedback Level Calculator

Voltage Feedback Shunt Amplifier Current Transformer Digital Source

Circuit

Input Scaling

- ADC Reference: V
- ADC Resolution: Bit
- Minimum:
- Maximum:
- Differential (signed)
- Max: °
- Div: °

Calculation

- Nominal Sense Voltage: V
- R1: Ω
- R2: Ω
- Amplifier Gain: V/V
- Signal Gain: V/V

Options

- Unwrap Phase
- Show s-Domain

FP Error	Int	UInt	Hexad
0.002%	27,771	27,771	0x6C7
0.004%	19,413	19,413	0x4BD
0.005%	18,503	18,503	0x484
0.001%	17,266	17,266	0x437
-0.003%	-27,855	37,681	0x933
0.000%	-17,111	48,425	0xBD2
0.001%	28,166	28,166	0x6E0

Refresh Period: 119 ms Table Options

Config K_P (P-Term) Gain

PowerSmart Digital Control Library Designer v1.0

Controller **Source Code Configuration** **Advanced**

Software Context Management

- Save/Restore Shadow Registers
- Save/Restore MAC Working Registers
- Save/Restore Accumulators
 - Save/Restore Accumulator A
 - Save/Restore Accumulator B
- Save/Restore DSP Core Configuration
- Save/Restore Core Status Register

Input Gain

Input Data Resolution:

Input Signal Gain:

- Normalize Input Gain
- Feedback Offset Compensation
- Enable Singal Rectification

Compensation Filter Settings

Sampling Frequency:

Pole 1: Zero 1:

Pole 2: Zero 2:

Anti-Windup

- Limit Control Loop Output to Positive Numbers

Anti Windup Limiter Number Range: -32768...32767
- Clamp Control Output Maximum
 - Generate Upper Saturation Status Flag Bit
- Clamp Control Output Minimum
 - Force Values below Minimum Threshold to Zero
 - Generate Lower Saturation Status Flag Bit

Coefficients generated successfully

Source Code Configuration

Use P-Term Loop Controller for Plant Measurements

- Nominal Feedback Level:
- Nominal Control Output:
- Fractional:
- Scaler:

Enable Feedback Loop Gain Modulation (AGC)

- Add Enable/Disable Adaptive Gain Control (AGC)
- Add Observer Function Call before Modulation
- Optimize AGC Modulation Factor Accuracy

Add User Extensions

- Start of Control Loop
- After Reading Source
- Before Anti-Windup
- Before Writing to Target
- End of Control Loop
- Cascade Function Call

Advanced

Please Note: Execution time of user functions being called during loop execution is not included in the Cont...

Nominal Feedback Level Calculator

Circuit

Input Scaling

- ADC Reference:
- ADC Resolution:
- Minimum:
- Maximum:

Calculation

- Nominal Sense Voltage:
- R1:
- R2:
- Amplifier Gain:

Nominal Output Level Calculator

PWM Signal

PWM Time Base

- Device Type:
- Clock Frequency:
- Divider:
- Resolution:
- Maximum:

Calculation

- PWM Frequency:
- PWM Period:
- PWM Period Count:
- Effective Resolution:
- Nominal Duty Ratio:

OK Cancel

POW201 Full Digital Power Development Using dsPIC33CH Dual-Core DSC

Config K_P (P-Term) Gain

PowerSmart Digital Control Library Designer v1.0

Controller **Source Code Configuration** **Advanced**

- Software Context Management
 - Save/Restore Shadow Registers
 - Save/Restore MAC Working Registers
 - Save/Restore Accumulators
 - Save/Restore Accumulator A
 - Save/Restore Accumulator B
 - Save/Restore DSP Core Configuration
 - Save/Restore Core Status Register

Input Gain

Input Data Resolution:

Input Signal Gain:

- Normalize Input Gain
- Feedback Offset Compensation
- Enable Singal Rectification

Compensation Filter Settings

Sampling Frequency:

Pole 1: Pole 2:

Zero 1: Zero 2:

Basic Feature Extensions

- Add DSP Core Configuration
- Add Enable/Disable Feature
 - Always read from source when disabled
 - Add Error Normalization
 - Add Automatic Placement of Primary ADC Trigger A
 - Add Automatic Placement of Secondary ADC Trigger B
- Add User Extensions
 - Start of Control Loop
 - After Reading Source
 - Before Anti-Windup
 - Before Writing to Target
 - End of Control Loop
 - Cascade Function Call
- Anti-Windup
 - Limit Control Loop Output to Positive Numbers

Anti Windup Limiter Number Range: -32768...32767
 - Clamp Control Output Maximum
 - Generate Upper Saturation Status Flag Bit
 - Clamp Control Output Minimum
 - Force Values below Minimum Threshold to Zero
 - Generate Lower Saturation Status Flag Bit

Coefficients generated successfully

Source Code Configuration

Advanced

Use P-Term Loop Controller for Plant Measurements

Nominal Feedback Level: Nominal Control Output:

Fractional: Scaler:

Enable Feedback Loop Gain Modulation (AGC)

Add Enable/Disable Adaptive Gain Control (AGC)

Add Observer Function Call before Modulation

Optimize AGC Modulation Factor Accuracy

Add User Extensions

- Start of Control Loop
- After Reading Source
- Before Anti-Windup
- Before Writing to Target
- End of Control Loop
- Cascade Function Call

Please Note:
Execution time of user functions being called during loop execution is not included in the Control Loop Time.

Nominal Feedback Level Calculator

Circuit

Input Scaling

ADC Reference: ADC Resolution:
 Minimum: Maximum:

Calculation

Nominal Sense Voltage: R1:
 R2: Amplifier Gain:
 Signal Gain:

Nominal Output Level Calculator

PWM Signal

PWM Time Base

Device Type: dsPIC33C
 Clock Frequency: Divider:
 Resolution: Maximum:

Calculation

PWM Frequency: PWM Period:
 PWM Period Count: Effective Resolution:
 Nominal Duty Ratio: Signal Gain:

Nominal Control Output Calculator

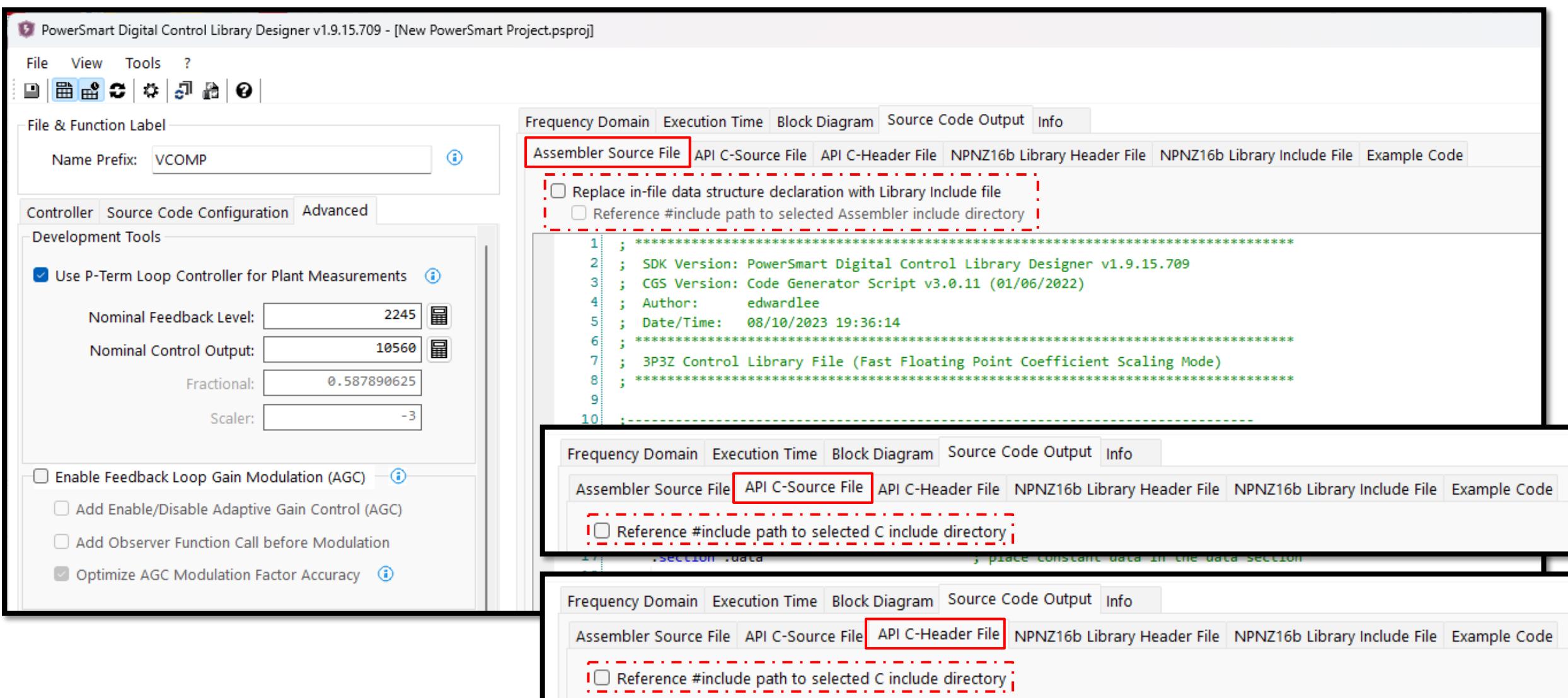
Nominal Control Output

Converter Type: D - Buck/Forward Converter

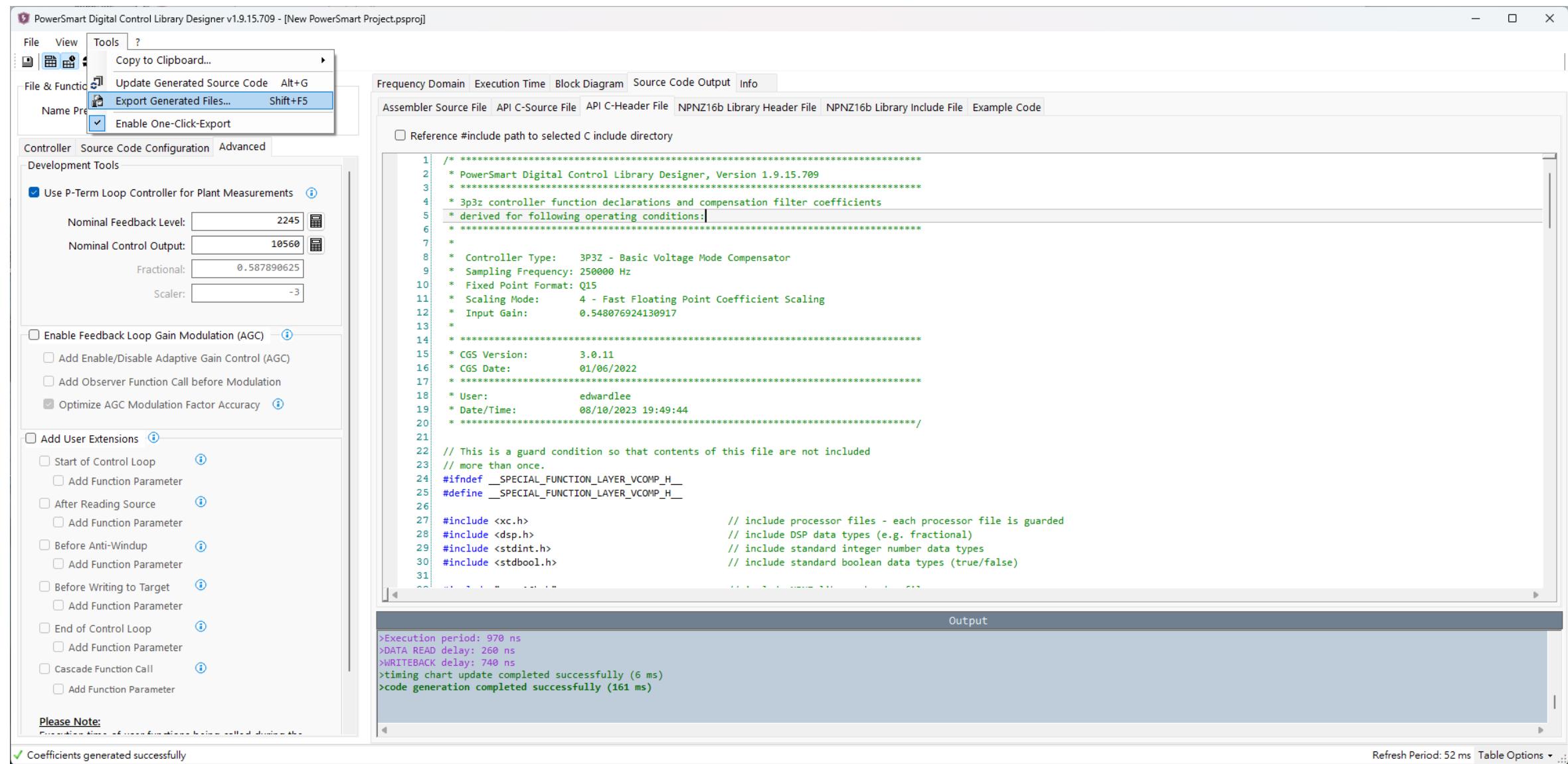
Winding Ratio (P/S): Nominal Input Voltage:
 Nominal Output Voltage: Nominal Efficiency:

Nominal Duty Ratio: 66.000000 %

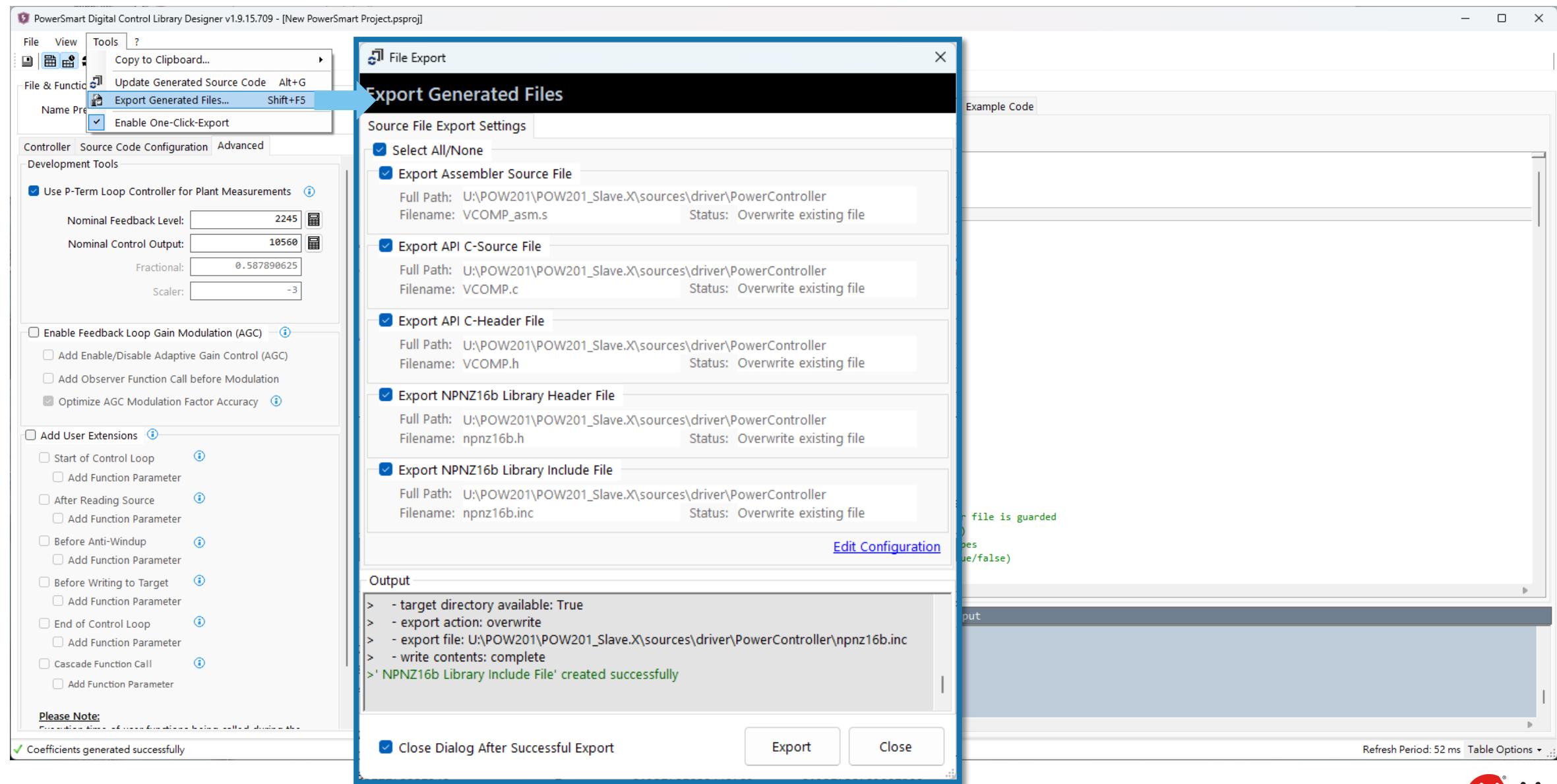
Setting for C-include Directory



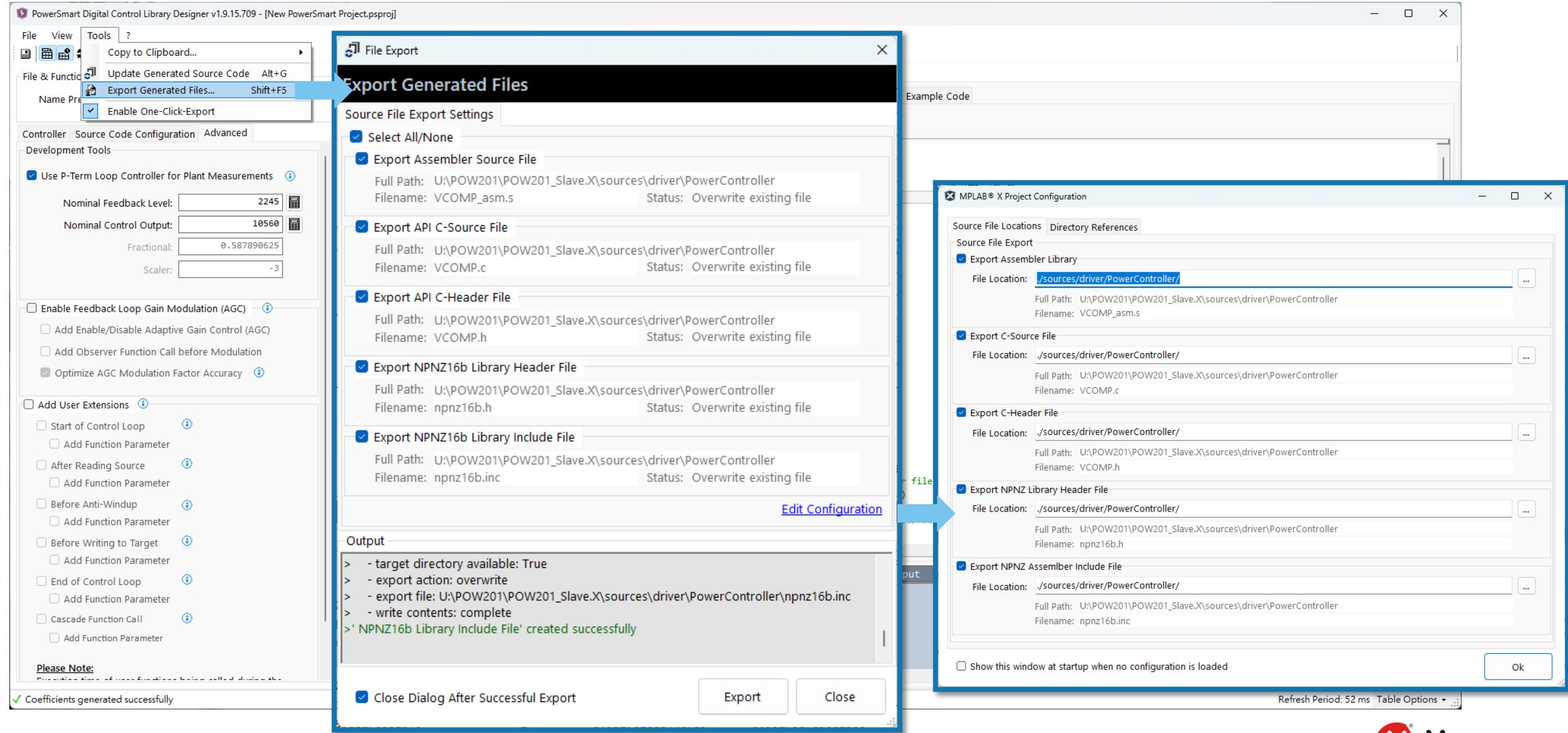
Export Compensator Sources Codes



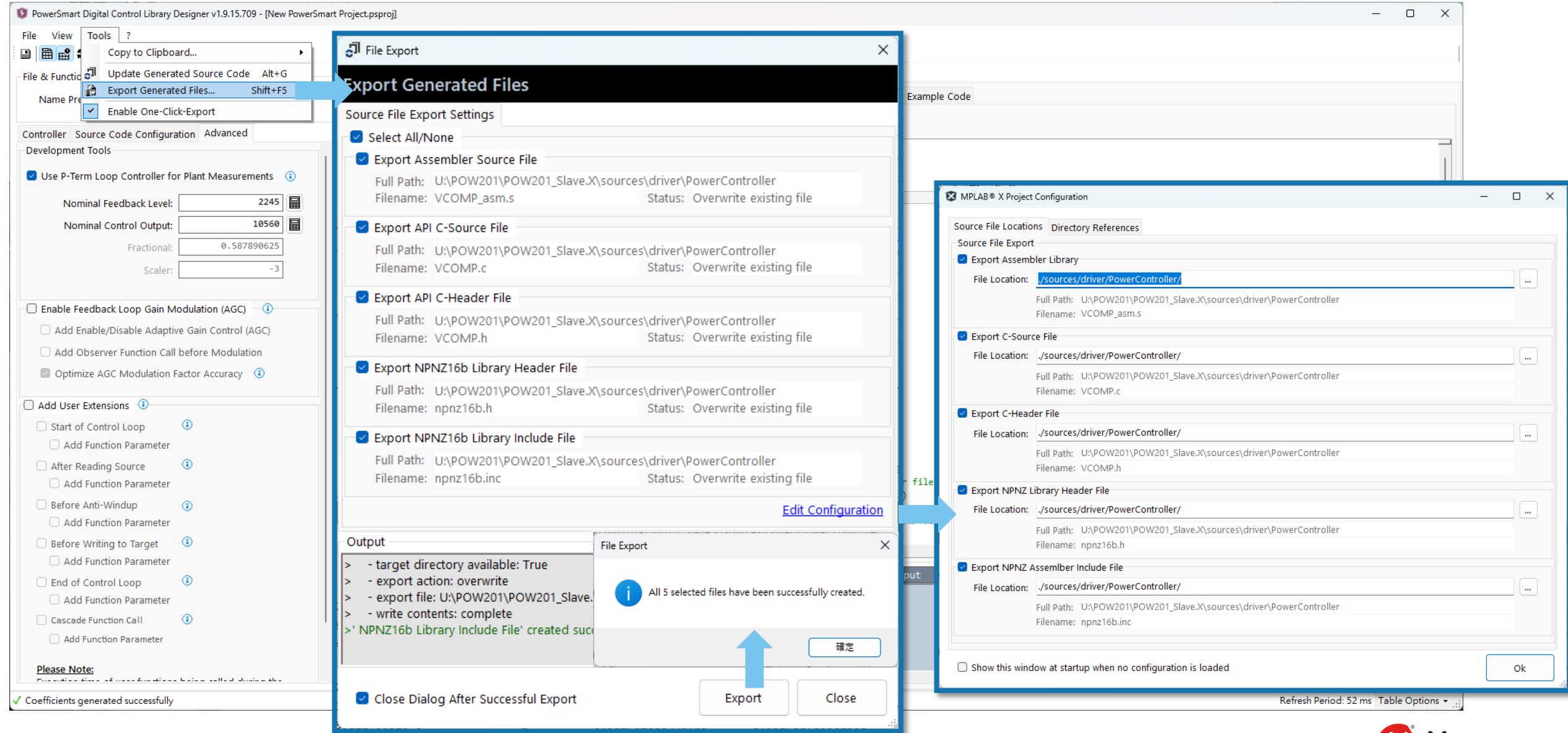
Export Compensator Sources Codes



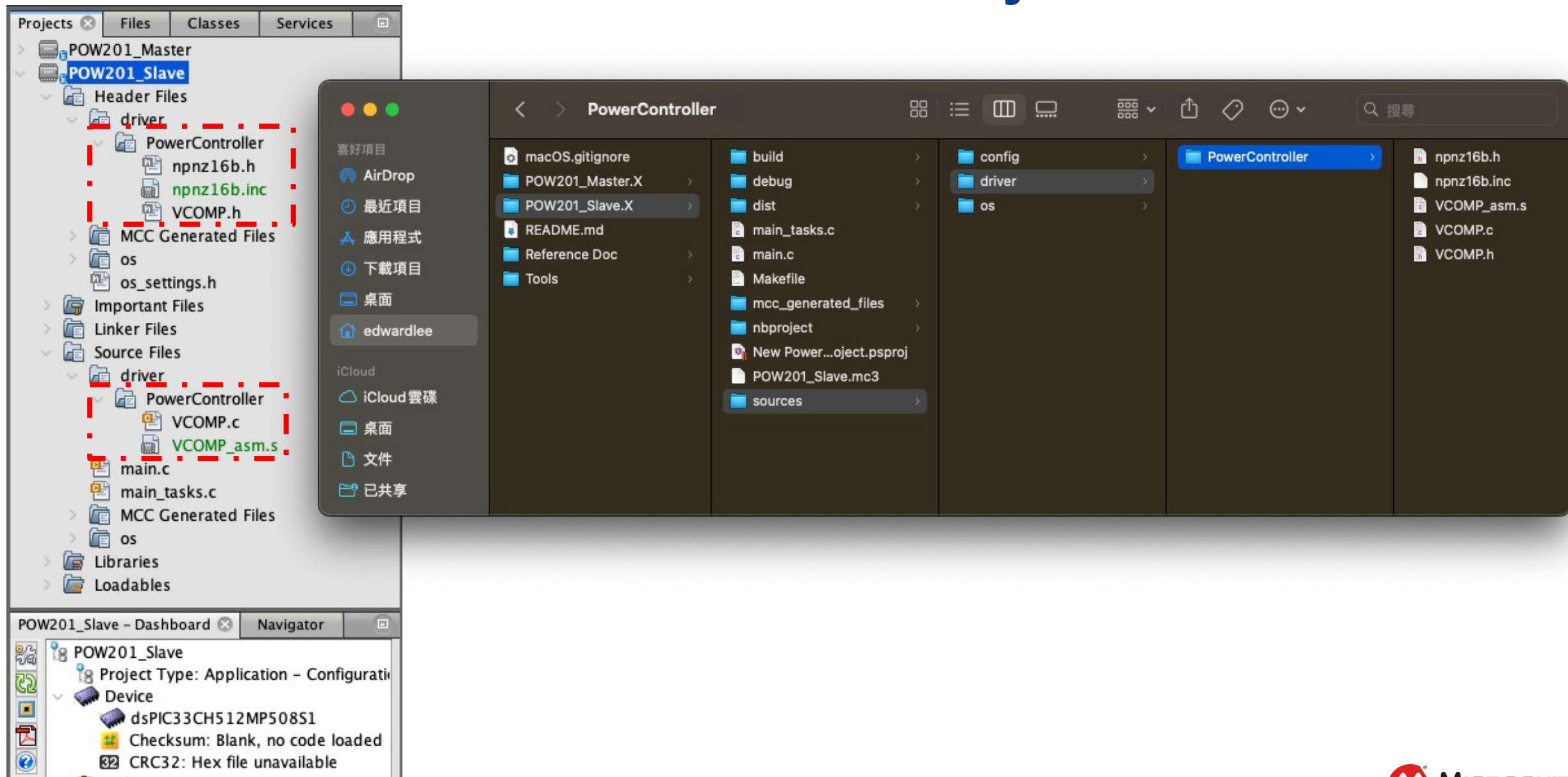
Export Compensator Sources Codes



Export Compensator Sources Codes



Add Control Libraries to the Project



Setting for C-include Directory

The image shows the MPLAB X IDE interface. On the left, the Project Explorer displays the project structure for 'POW201_Slave'. It includes 'Header Files' containing 'PowerController' with files 'nzn16b.h', 'nzn16b.inc', and 'VCOMP.h'; 'MCC Generated Files'; 'os' with 'os_settings.h'; 'Important Files'; 'Linker Files'; 'Source Files' containing 'driver' with 'PowerController' files 'VCOMP.c' and 'VCOMP_asm.s'; and source files 'main.c' and 'main_tasks.c'. Below these are 'MCC Generated' files, 'os', 'Libraries', and 'Loadables'. A blue arrow points from the 'Header Files' section towards the 'Project Properties' dialog on the right.

Project Properties - POW201_Slave

Categories:

- General
- File Inclusion/Exclusion
- Conf: [default]**
- XC16
 - XC16 (Global Options)
 - xc16-as
 - xc16-gcc
 - xc16-ld
 - xc16-ar
 - Analysis

Options for xc16-gcc (v2.10)

Option categories: Global options

Override default device support: Do not override

Output file format: ELF/DWARF

Define common macros: (N/A)

Generic build: Single

Don't delete intermediate files:

Partition: Single

Common include dirs: sources/driver/powercontroller

Elf file to use for preserved locations:

Additional options:

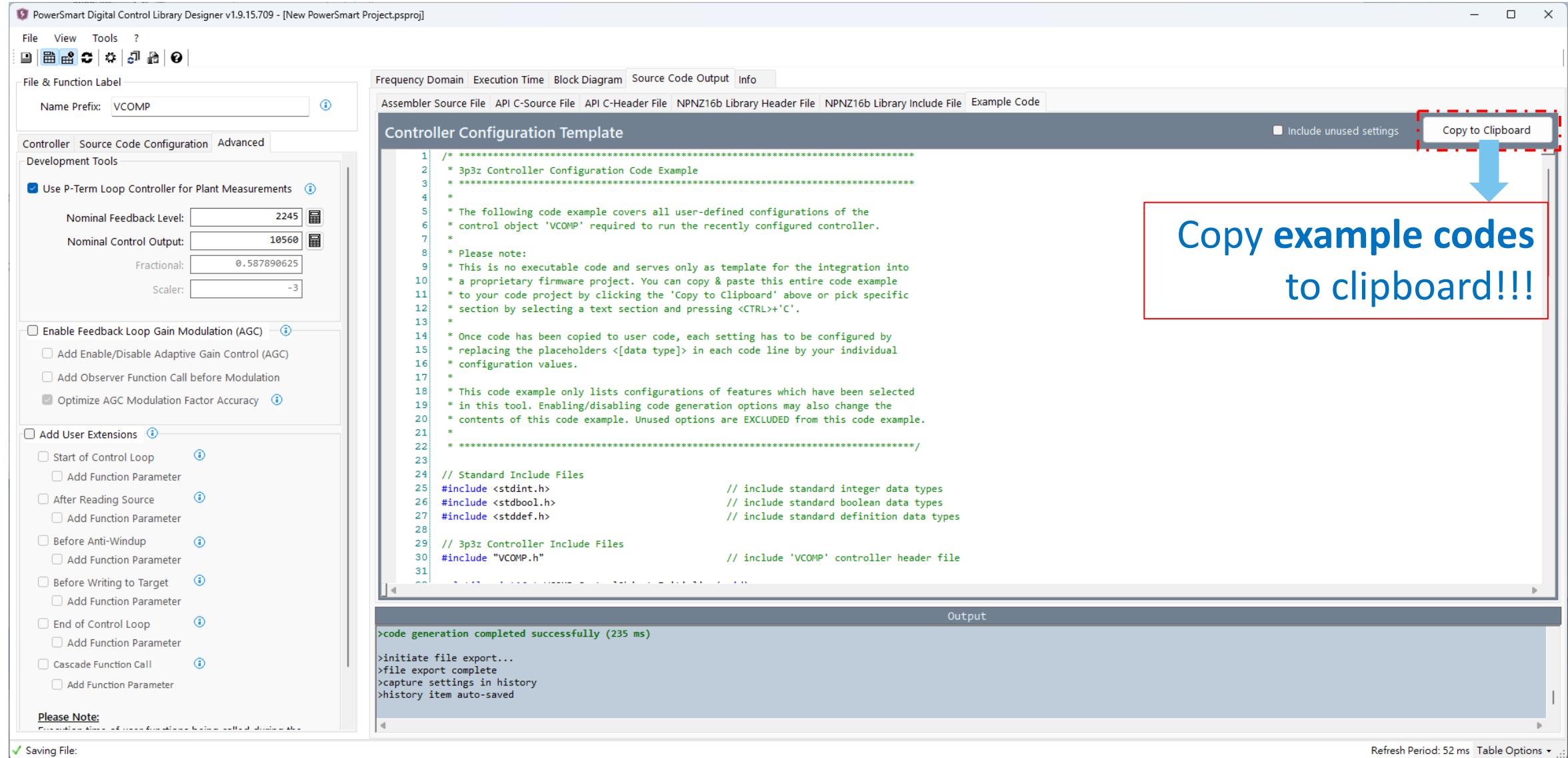
Option Description | Generated Command Line | User Comments

Add 'dir' to the list of shared include directories for compiler and assembler.
Relative paths are from MPLAB X project directory.

Manage Configurations... Manage Network Tools...

Help Cancel Apply Unlock OK

Copy example codes



New file with pasting example codes!

The screenshot shows the Microchip IDE interface with three code editors open:

- BuckConverter.h**: Contains the header file code.
- main.c**: Contains the main application code.
- BuckConverter.c**: Contains the code example for the Buck Converter application.

The **BuckConverter.c** editor displays the following code example:

```
1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
```

The code example in **BuckConverter.c** is as follows:

```
/*
 * ****3p3z Controller Configuration Code Example
 *
 * The following code example covers all user-defined configurations of the
 * control object 'VCOMP' required to run the recently configured controller.
 *
 * Please note:
 * This is no executable code and serves only as template for the integration into
 * a proprietary firmware project. You can copy & paste this entire code example
 * to your code project by clicking the 'Copy to Clipboard' above or pick specific
 * section by selecting a text section and pressing <CTRL>+'C'.
 *
 * Once code has been copied to user code, each setting has to be configured by
 * replacing the placeholders <[data type]> in each code line by your individual
 * configuration values.
 *
 * This code example only lists configurations of features which have been selected
 * in this tool. Enabling/disabling code generation options may also change the
 * contents of this code example. Unused options are EXCLUDED from this code example
 *
 * ****
 */
// Standard Include Files
#include <stdint.h> // include standard integer data types
#include <stdbool.h> // include standard boolean data types
#include <stddef.h> // include standard definition data types

// 3p3z Controller Include Files
#include "VCOMP.h" // include 'VCOMP' controller header file

// 3p3z Controller Configuration
VCOMP_ControlObject_Initialize(); // Initialize VCOMP controller
SYSTEM_Initialize(); // Initialize the device
OS_Init(); // Initialize OS
OS_Scheduler_RunForever(); // Start OS scheduler

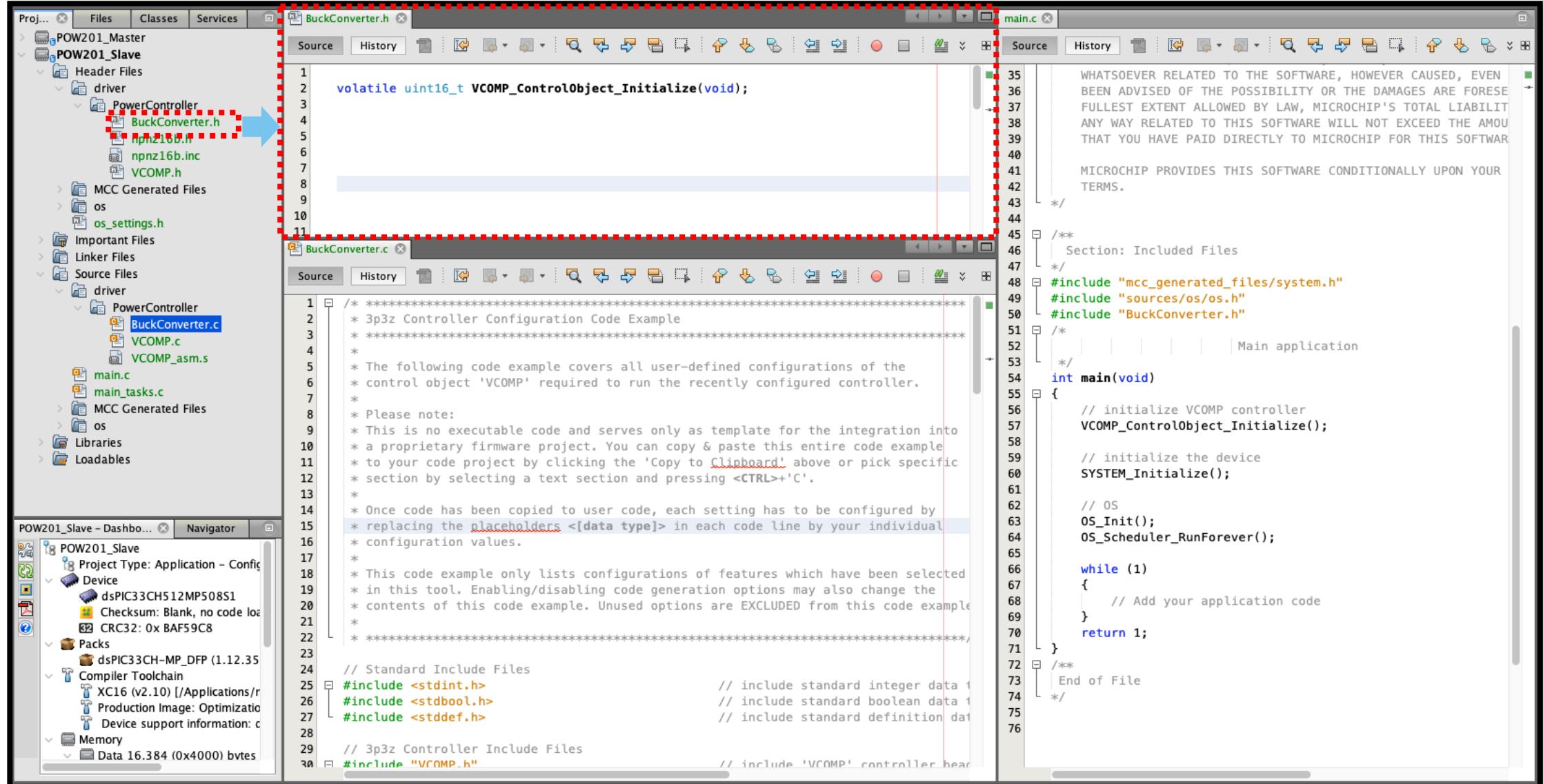
while (1)
{
    // Add your application code
}

return 1;
*/
// End of File
```

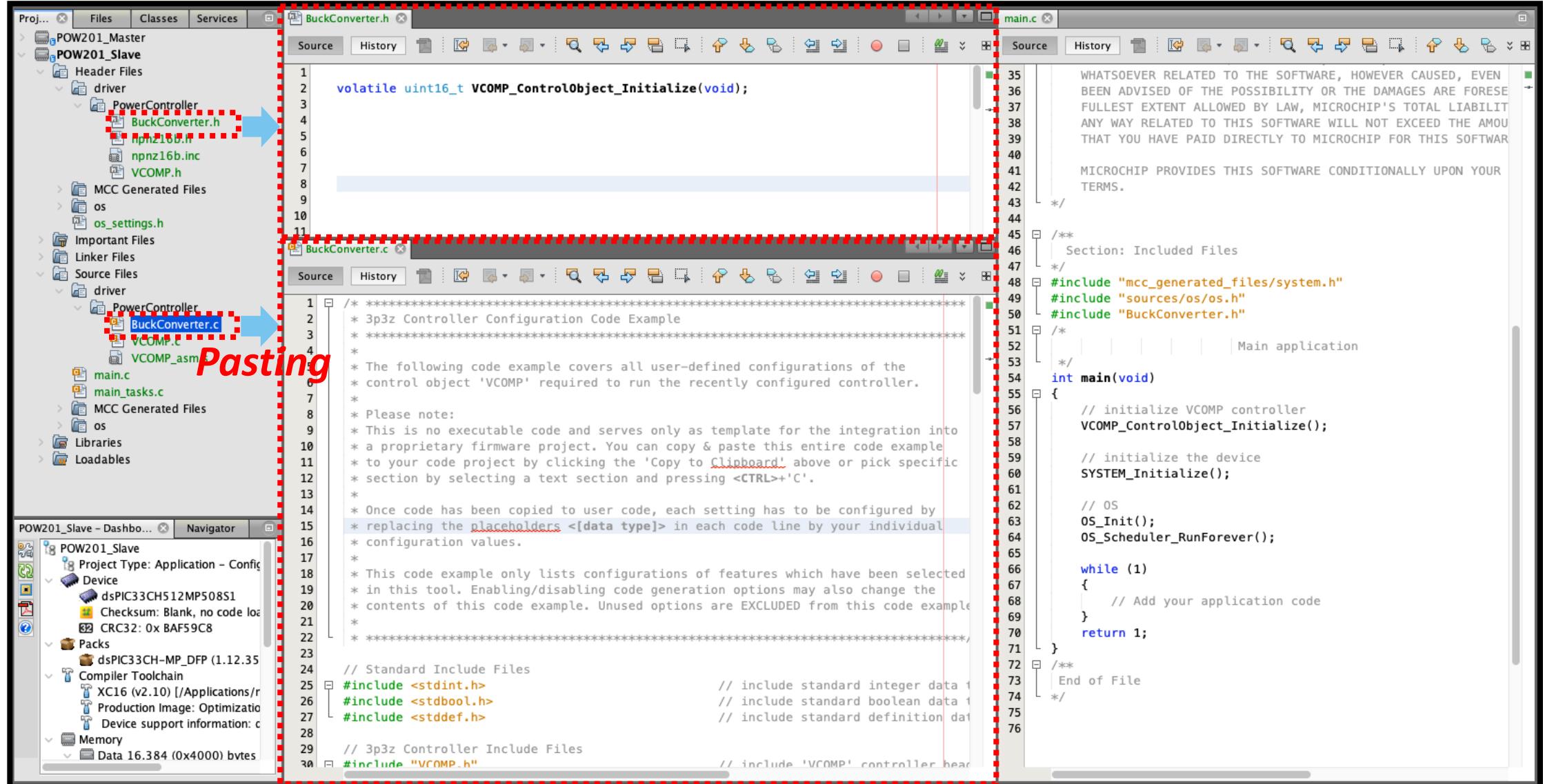
The **Project Explorer** on the left shows the project structure for **POW201_Slave**:

- POW201_Master**
- POW201_Slave**
 - Header Files**
 - driver**
 - PowerController**
 - BuckConverter.h**
 - npnz16b.h**
 - npnz16b.inc**
 - VCOMP.h**
 - MCC Generated Files**
 - os**
 - os_settings.h**
 - Important Files**
 - Linker Files**
 - Source Files**
 - driver**
 - PowerController**
 - BuckConverter.c**
 - VCOMP.c**
 - VCOMP_asm.s**
 - main.c**
 - main_tasks.c**
 - MCC Generated Files**
 - os**
 - Libraries**
 - Loadables**
- POW201_Slave - Dashboard**
- Navigator**

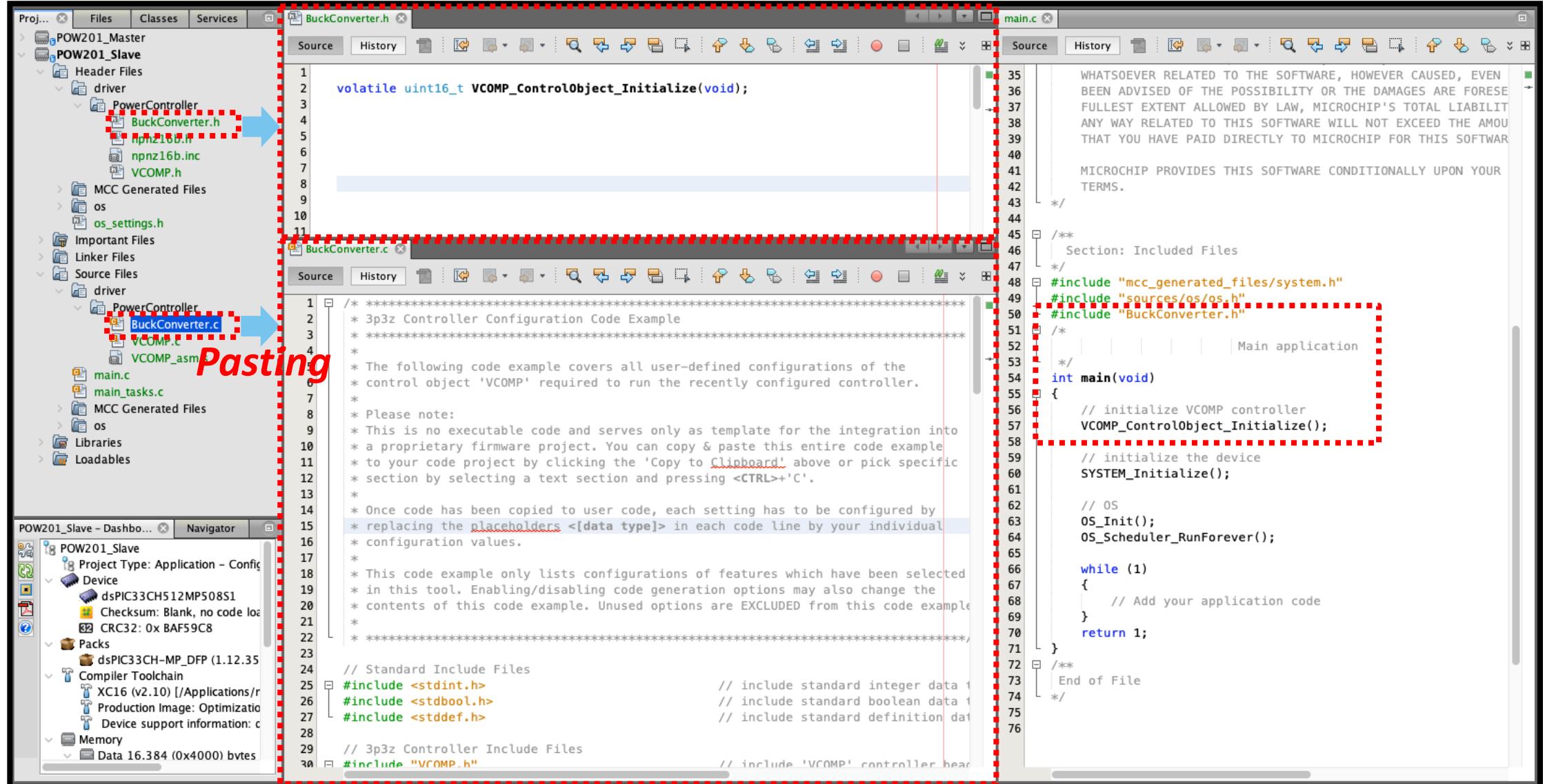
New file with pasting example codes!



New file with pasting example codes!



New file with pasting example codes!



Setting Initial Registers

```
BuckConverter.c
Source History
23 #define __BuckConverter
24
25 // Standard Include Files
26 #include <stdint.h> // include standard integer data type
27 #include <stdbool.h> // include standard boolean data type
28 #include <stddef.h> // include standard definition data type
29
30 // 3p3z Controller Include Files
31 #include "VCOMP.h" // include 'VCOMP' controller header
32 #include "BuckConverter.h" // include 'BuckConverter' module header
33 #include "../../../../mcc_generated_files/pin_manager.h"
34
35 volatile uint16_t VCOMP_ControlObject_Initialize(void)
36 {
37     volatile uint16_t retval = 0; // Auxiliary variable for function return
38     Buck_Vref = 0;
39     /* Controller Input and Output Ports Configuration */
40
41     // Configure Controller Primary Input Port
42     VCOMP.Ports.Source.ptrAddress = &ADCBUF1; // Pointer to primary feedback source
43     VCOMP.Ports.Source.Offset = 0; // Primary feedback signal offset
44     VCOMP.Ports.Source.NormScaler = 0; // Primary feedback normalization factor
45     VCOMP.Ports.Source.NormFactor = 0x7FFF; // Primary feedback normalization factor
46
47     // Configure Controller Primary Output Port
48     VCOMP.Ports.Target.ptrAddress = &PG7DC; // Pointer to primary output target
49     VCOMP.Ports.Target.Offset = 0; // Primary output offset value
50     VCOMP.Ports.Target.NormScaler = 0; // Primary output normalization factor
51     VCOMP.Ports.Target.NormFactor = 0x7FFF; // Primary output normalization factor
52
53     // Configure Control Reference Port
54     VCOMP.Ports.ptrControlReference = &Buck_Vref; // Pointer to control reference (used for soft start)
55
56     /* Controller Output Limits Configuration */
57
58     // Primary Control Output Limit Configuration
59     VCOMP.Limits.MinOutput = VCOMP_MIN_CLAMP; // Minimum control output value
60     VCOMP.Limits.MaxOutput = VCOMP_MAX_CLAMP; // Maximum control output value
61
62     /* Advanced Parameter Configuration */
63
64     // Initialize User Data Space Buffer Variables
65     VCOMP.Advanced.usrParam0 = 0; // No additional advanced control
66
main.c pwm.c
Source History
134 PG7PHASE = 0x00;
135 // DC 0;
136 PG7DC = 0x00;
137 // DCA 0;
138 PG7DCA = 0x00;
139 // PER 15992;
140 PG7PER = 0x3E78;
141 // TRIGA 0;
142 PG7TRIGA = 0x00;
143 // TRIGB 0;
144 PG7TRIGB = 0x00;
145 // TRIGC 0;
146 PG7TRIGC = 0x00;
147 // DTL 0;
148 PG7DTL = 0x00;
149 // DTH 0;
150 PG7DTH = 0x00;
151
152 PG7CONLbits.ON = 1;
153
BuckConverter.h
Source History
1 #ifndef __BuckConverter
2 #define EXTERN
3
4 #else
5 #define EXTERN extern
6 #endif
7
8 // Reference for Voltage Loop Compensator
9 #define VCOMP_VREF 2245
10
11 // Compensator Clamp Limits
12 #define VCOMP_MIN_CLAMP 0x0010
13 #define VCOMP_MAX_CLAMP 14393 // PG7PER * 0.9
14
15 EXTERN uint16_t Buck_Vref;
16 volatile uint16_t VCOMP_ControlObject_Initialize(void);
17 void Buck_Softstart(void);
18
19
```

Setting ADC ISR

```
BuckConverter.c
Source History ... adc1.c
Source History ... adc1.c
346     ADC1_VoutFBDefaultInterruptHandler = handler;
347 }
348
349 void __attribute__ (( __interrupt__ , auto_psv, weak )) _ADCAN1Interrupt ( void )
350 {
351     uint16_t valVoutFB;
352     //Read the ADC value from the ADCBUF
353     valVoutFB = ADCBUF1;
354
355     if(ADC1_VoutFBDefaultInterruptHandler)
356     {
357         ADC1_VoutFBDefaultInterruptHandler(valVoutFB);
358     }
359
360     //clear the VoutFB interrupt flag
361     IFS5bits.ADCAN1IF = 0;
362 }
363
364
365
366 /**
367  * End of File
368 */
BuckConverter.h
Source History ...
1
2 #ifdef __BuckConverter
3 #define EXTERN
4 #else
5 #define EXTERN extern
6 #endif
7
8 // Reference for Voltage Loop Compensator
9 #define VCOMP_VREF 2245
10
11 // Compensator Clamp Limits
12 #define VCOMP_MIN_CLAMP 0x0010
13 #define VCOMP_MAX_CLAMP 14393 // PG7PER * 0.9
14
15 EXTERN uint16_t Buck_Vref;
16 volatile uint16_t VCOMP_ControlObject_Initialize(void);
17 void Buck_Softstart(void);
18
19
```

Setting ADC ISR

```
BuckConverter.c
Source History BuckConverter.c
89 * The Assembler library code sequences of controller data objects generated by
90 * for being called by a PWM interrupt for minimum response time. However, in s
91 * it might be desired to call the control loop from other interrupt sources.
92 * Using custom labels for interrupt routines allows using generic interrupt
93 * function calls in code, which can be mapped to specific interrupt sources
94 * pre-compiler directive declaration to your code, like the following example:
95 *
96 * #define _VCOMP_Interrupt    _PWM1Interrupt // Define label for interrupt
97 * #define _VCOMP_ISRIF        _PWM1IF      // Define label for interrupt
98 *
99 ****
100
101 void __attribute__ ( ( __interrupt__ , auto_psv ) ) _ADCAN1Interrupt ( void )
102 {
103     //LED2_SetHigh();
104
105     //VCOMP_Update(&VCOMP);           // Call control loop
106     VCOMP_PTermUpdate(&VCOMP);       // Call P-Term control loop
107
108     //LED2_SetLow();
109     IFS5bits.ADCAN1IF = 0;          // Clear the interrupt flag
110 }
111
112 ****
113 // Download latest version of this tool here:
114 // https://www.microchip.com/powersmart
115 ****
116
117 // Simple Softstart
118 void Buck_Softstart(void)
119 {
120     VCOMP.status.bits.enabled = true;           // Enable controller
121
122     if(Buck_Vref < VCOMP_VREF)
123     {
124         Buck_Vref+=10;
125     }
126     else
127     {
128         Buck_Vref = VCOMP_VREF;
129     }
130 }
131

main.c pwm.c adc1.c
Source History main.c
346     ADC1_VoutFBDefaultInterruptHandler = handler;
347 }
348
349 void __attribute__ ( ( __interrupt__ , auto_psv, weak ) ) _ADCAN1Interrupt ( void )
350 {
351     uint16_t valVoutFB;
352     //Read the ADC value from the ADCBUF
353     valVoutFB = ADCBUF1;
354
355     if(ADC1_VoutFBDefaultInterruptHandler)
356     {
357         ADC1_VoutFBDefaultInterruptHandler(valVoutFB);
358     }
359
360     //clear the VoutFB interrupt flag
361     IFS5bits.ADCAN1IF = 0;
362 }
363
364
365
366 /**
367 End of File
BuckConverter.h
Source History BuckConverter.h
1
2 #ifdef __BuckConverter
3 #define EXTERN
4 #else
5 #define EXTERN extern
6 #endif
7
8 // Reference for Voltage Loop Compensator
9 #define VCOMP_VREF 2245
10
11 // Compensator Clamp Limits
12 #define VCOMP_MIN_CLAMP 0x0010
13 #define VCOMP_MAX_CLAMP 14393 // PG7PER * 0.9
14
15 EXTERN uint16_t Buck_Vref;
16 volatile uint16_t VCOMP_ControlObject_Initialize(void);
17 void Buck_Softstart(void);
18
19
```

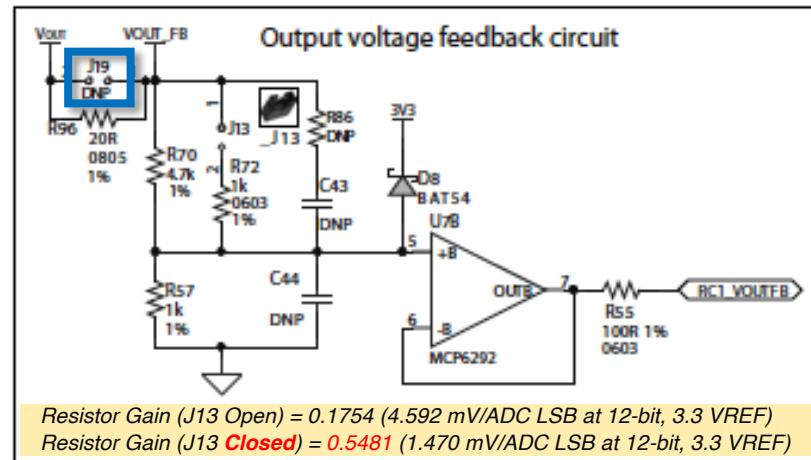
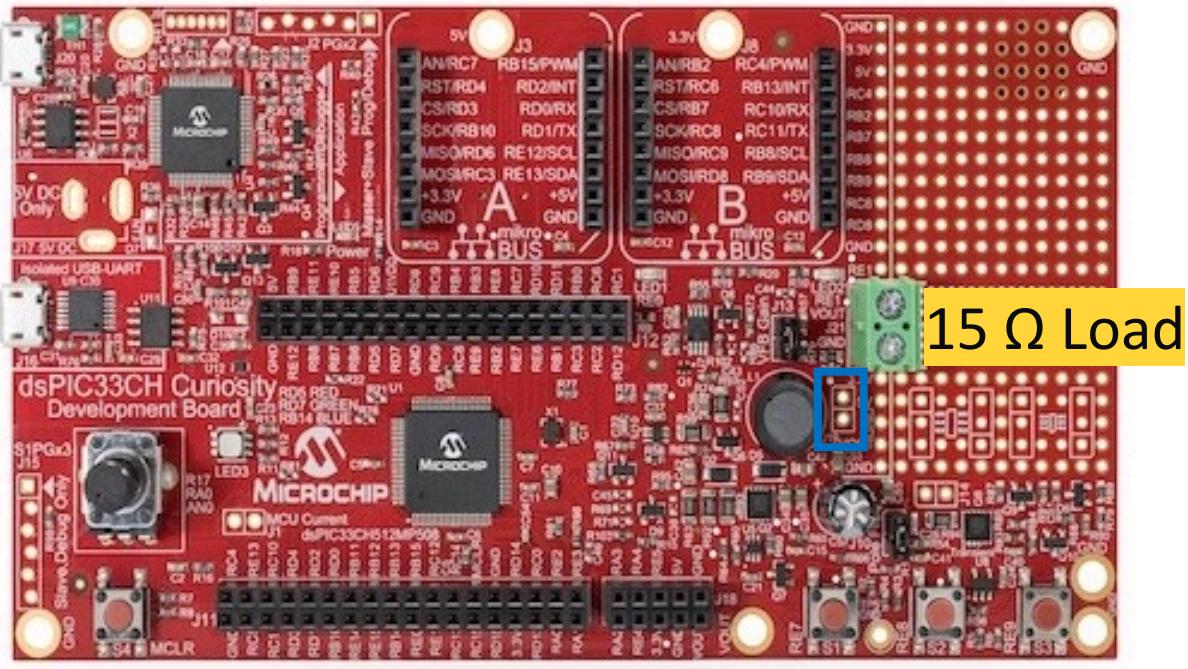
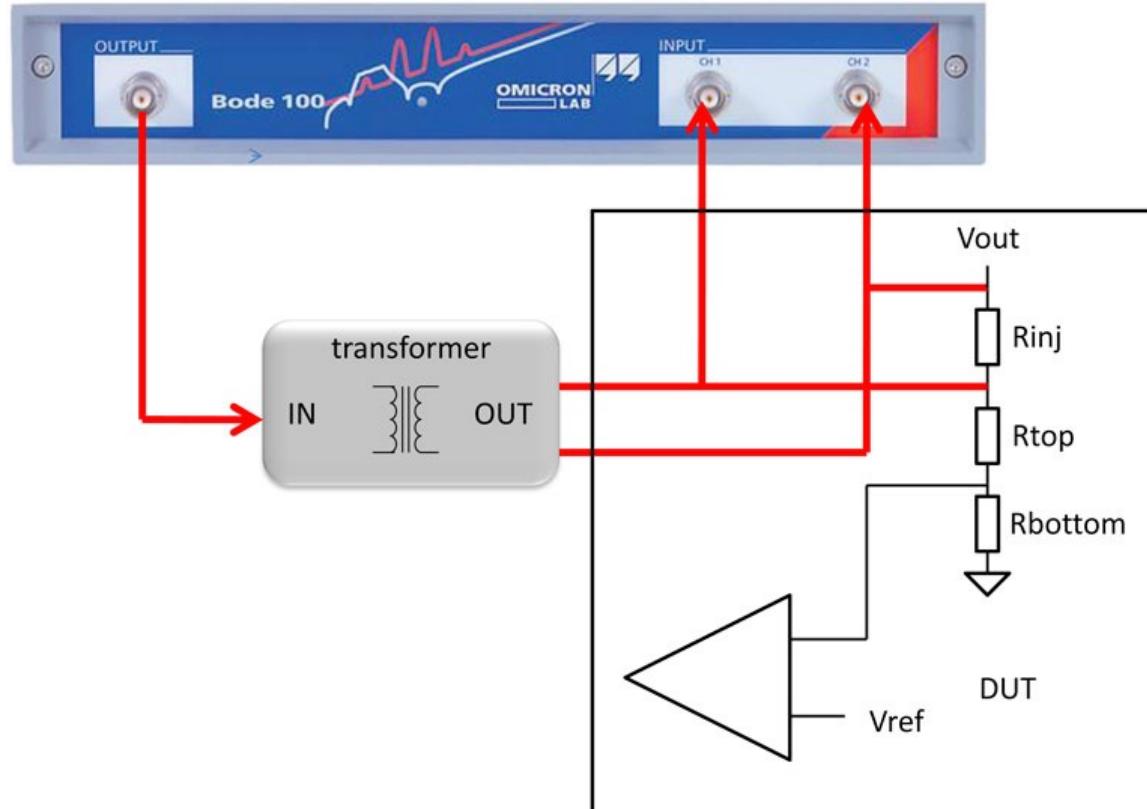
Implement Soft-start THEN Program BY Master!

```
BuckConverter.c
Source History
89 * The Assembler library code sequences of controller data objects generated by
90 * for being called by a PWM interrupt for minimum response time. However, in s
91 * it might be desired to call the control loop from other interrupt sources.
92 * Using custom labels for interrupt routines allows using generic interrupt se
93 * function calls in code, which can be mapped to specific interrupt sources by
94 * pre-compiler directive declaration to your code, like the following example:
95 *
96 * #define _VCOMP_Interrupt    _PWM1Interrupt // Define label for interrupt
97 * #define _VCOMP_ISRIF      _PWM1IF           // Define label for interrupt
98 *
99 ****
100
101 void __attribute__ (( __interrupt__ , auto_psv )) _ADCAN1Interrupt ( void )
102 {
103     //LED2_SetHigh();
104
105     //VCOMP_Update(&VCOMP);          // Call control loop
106     VCOMP_PTermUpdate(&VCOMP);      // Call P-Term control loop
107
108     //LED2_SetLow();
109     IFS5bits.ADCAN1IF = 0;          // Clear the interrupt flag
110 }
111
112 ****
113 // Download latest version of this tool here:
114 // https://www.microchip.com/powersmart
115 ****
116
117 // Simple Softstart
118 void Buck_Softstart(void)
119 {
120     VCOMP.status.bits.enabled = true;          // Enable controller
121
122     if(Buck_Vref < VCOMP_VREF)
123     {
124         Buck_Vref+=10;
125     }
126     else
127     {
128         Buck_Vref = VCOMP_VREF;
129     }
130 }
131

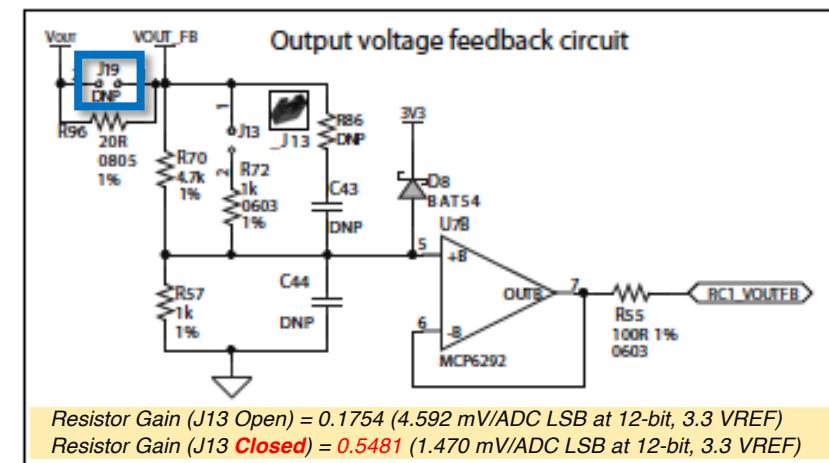
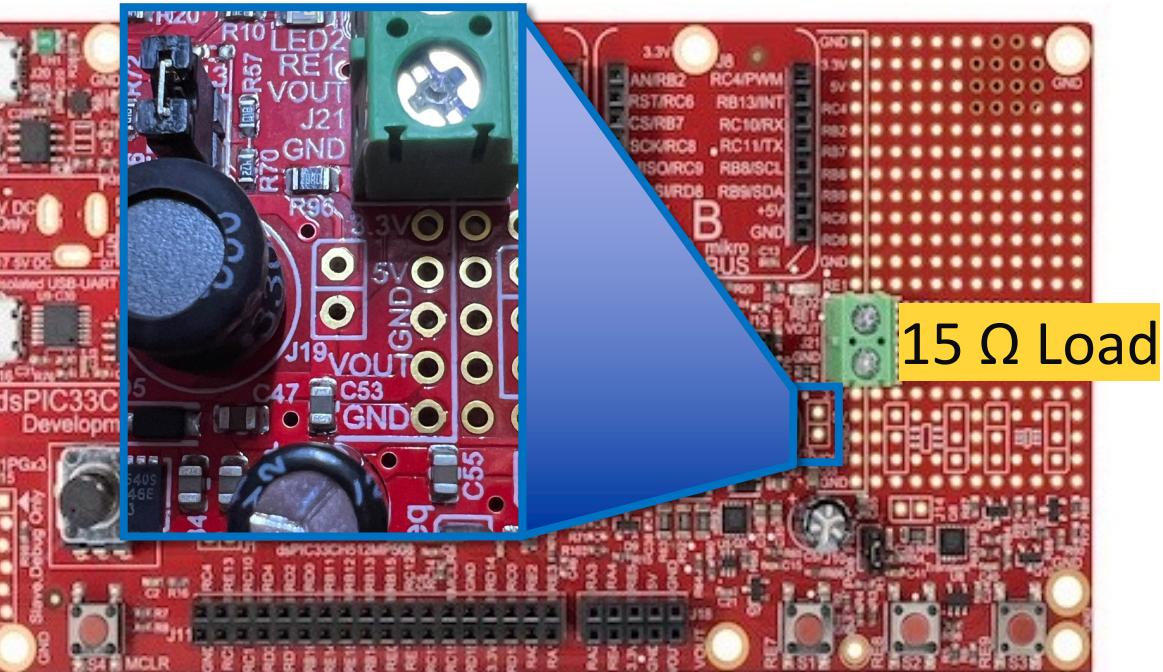
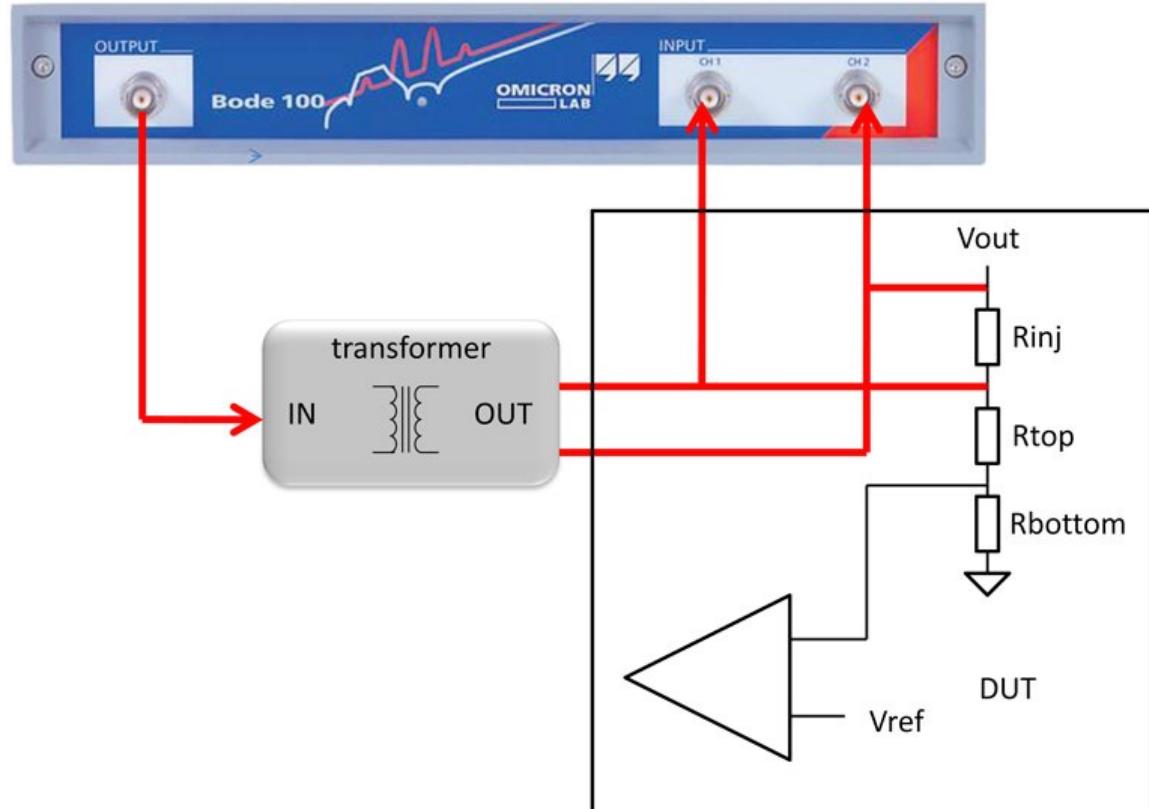
main.c
Source History
100 =====
101 // @brief Tasks_1ms gets called every millisecond, put your things in it that need
102 // @note there could be some jitter here because it is not called directly by a ti
103 =====
104 void Tasks_1ms(void)
105 {
106     Buck_Softstart();
107 }
108
109 =====
110 // @brief Tasks_10ms gets called every 10ms, put your things in it that need to be
111 // @note there could be some jitter here because it is not called directly by a ti
112 =====
113 void Tasks_10ms(void)
114 {
115     // put your application specific code here that needs to be called every 10 milli
116 }
117
118 =====
119 // @brief Tasks_100ms gets called every 100 ms, put your things in it that need to
120 // @note there could be some jitter here because it is not called directly by a ti
121

BuckConverter.h
Source History
1
2 #ifdef __BuckConverter
3 #define EXTERN
4 #else
5 #define EXTERN extern
6 #endif
7
8 // Reference for Voltage Loop Compensator
9 #define VCOMP_VREF 2245
10
11 // Compensator Clamp Limits
12 #define VCOMP_MIN_CLAMP 0x0010
13 #define VCOMP_MAX_CLAMP 14393 // PG7PER * 0.9
14
15 EXTERN uint16_t Buck_Vref;
16 volatile uint16_t VCOMP_ControlObject_Initialize(void);
17 void Buck_Softstart(void);
18
19
```

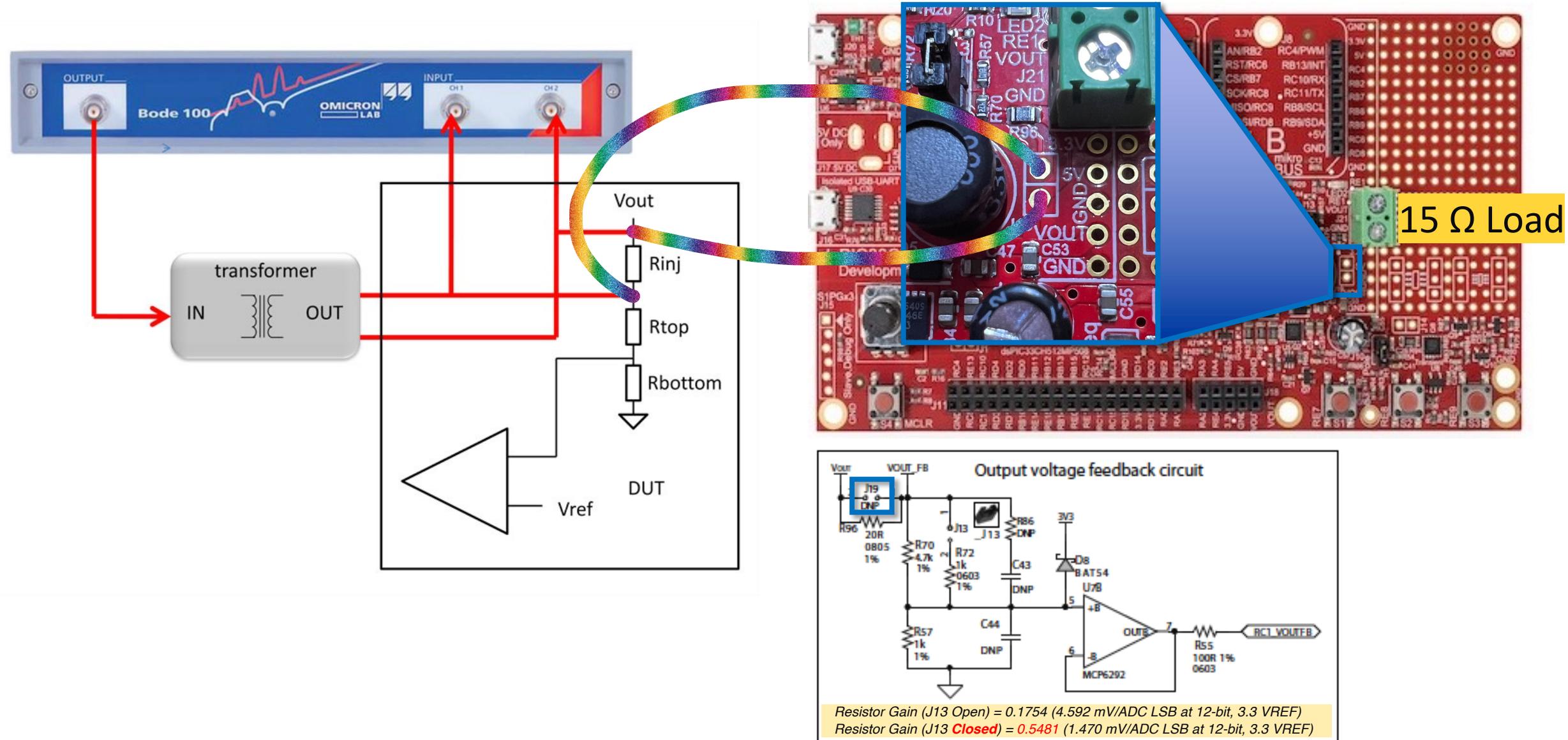
Plant Measurement Using Bode-100



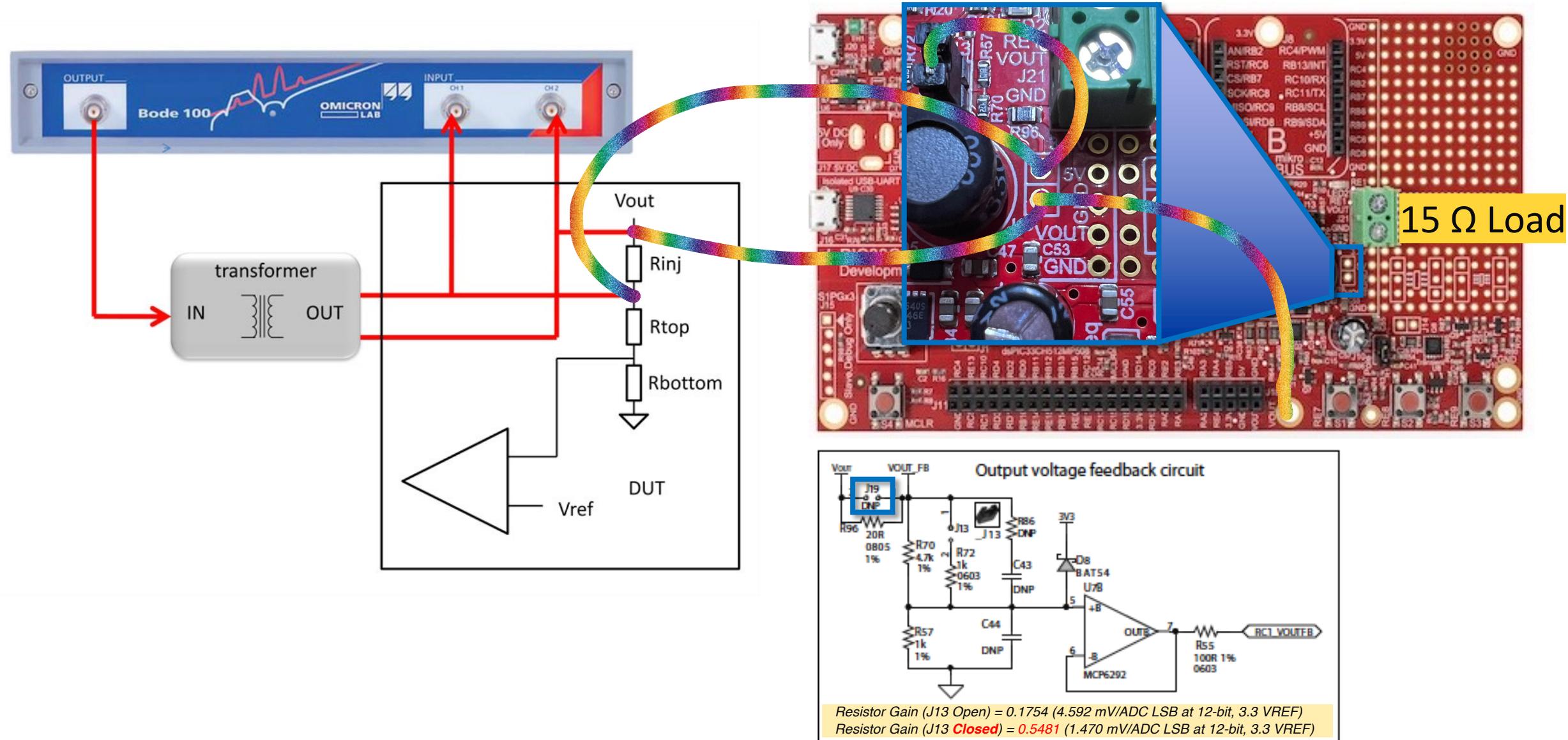
Plant Measurement Using Bode-100



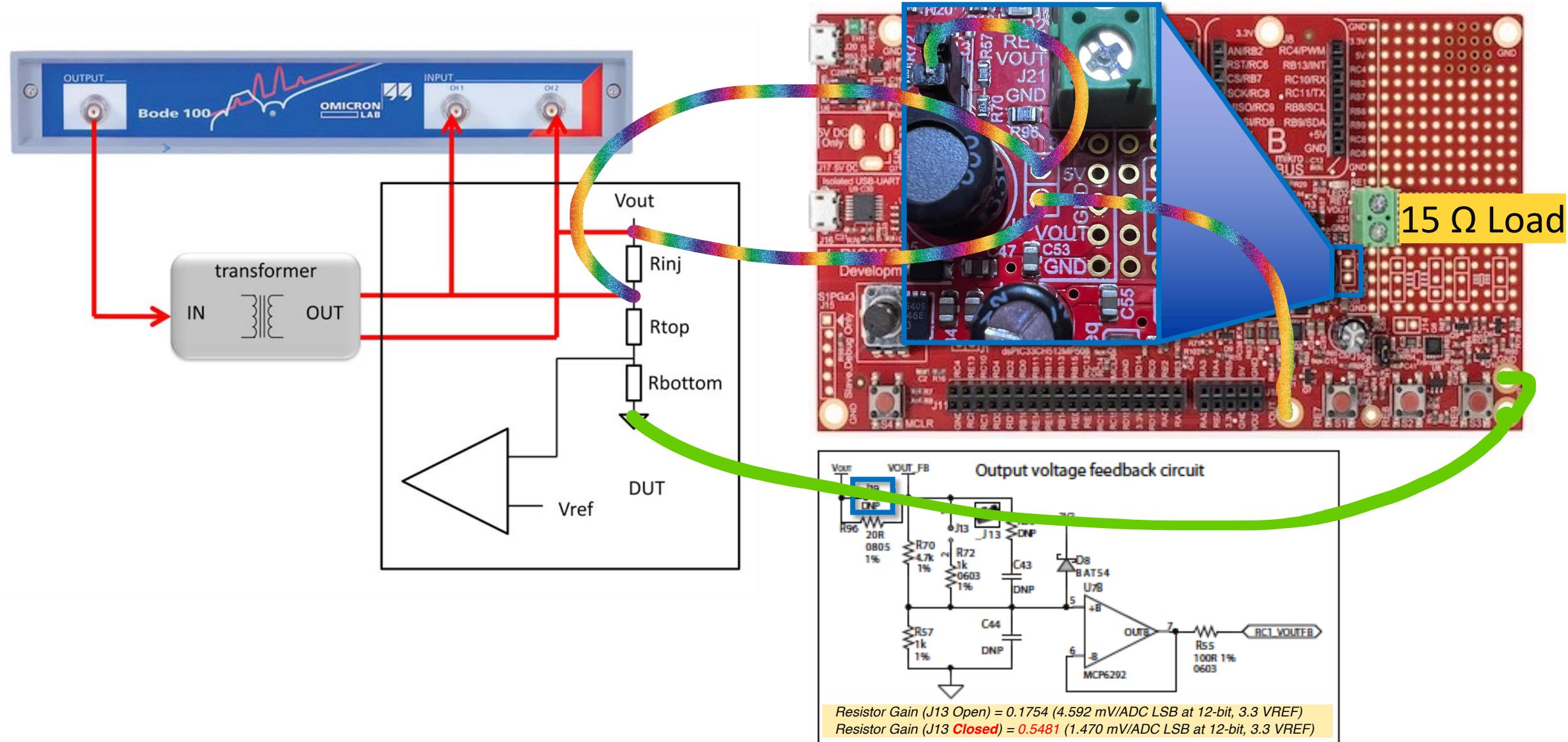
Plant Measurement Using Bode-100



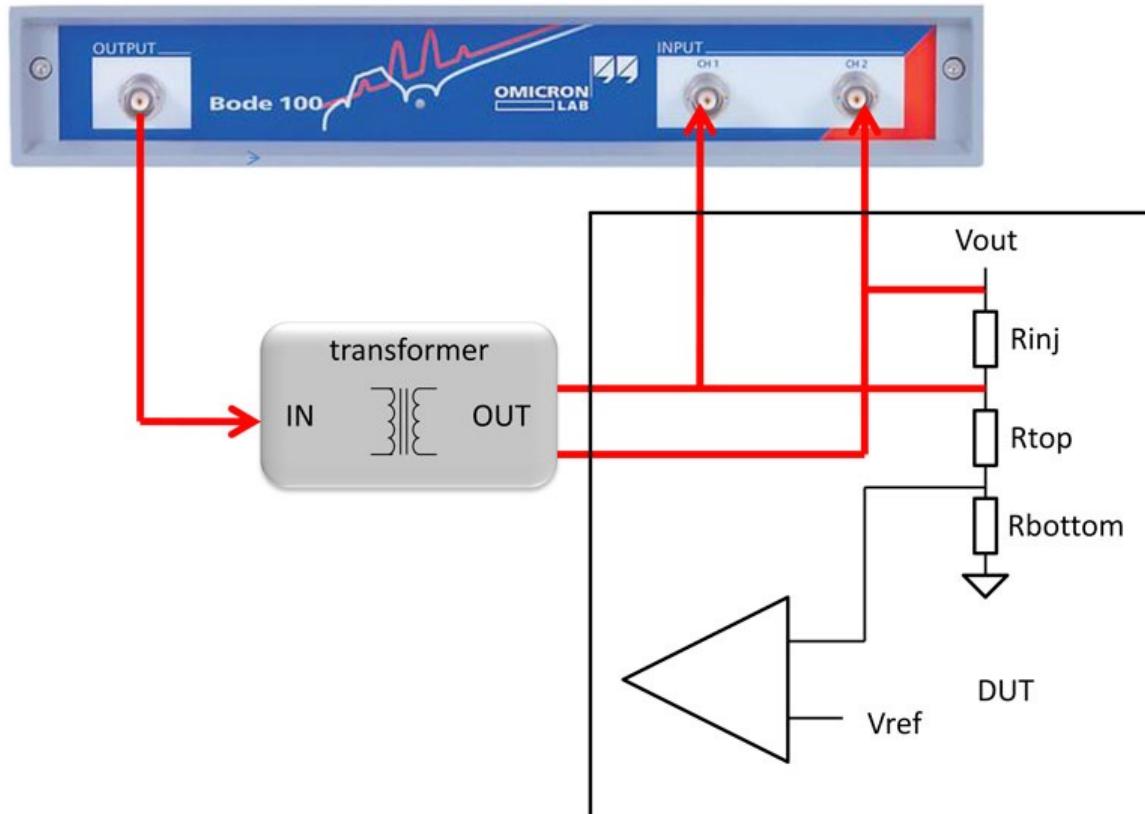
Plant Measurement Using Bode-100



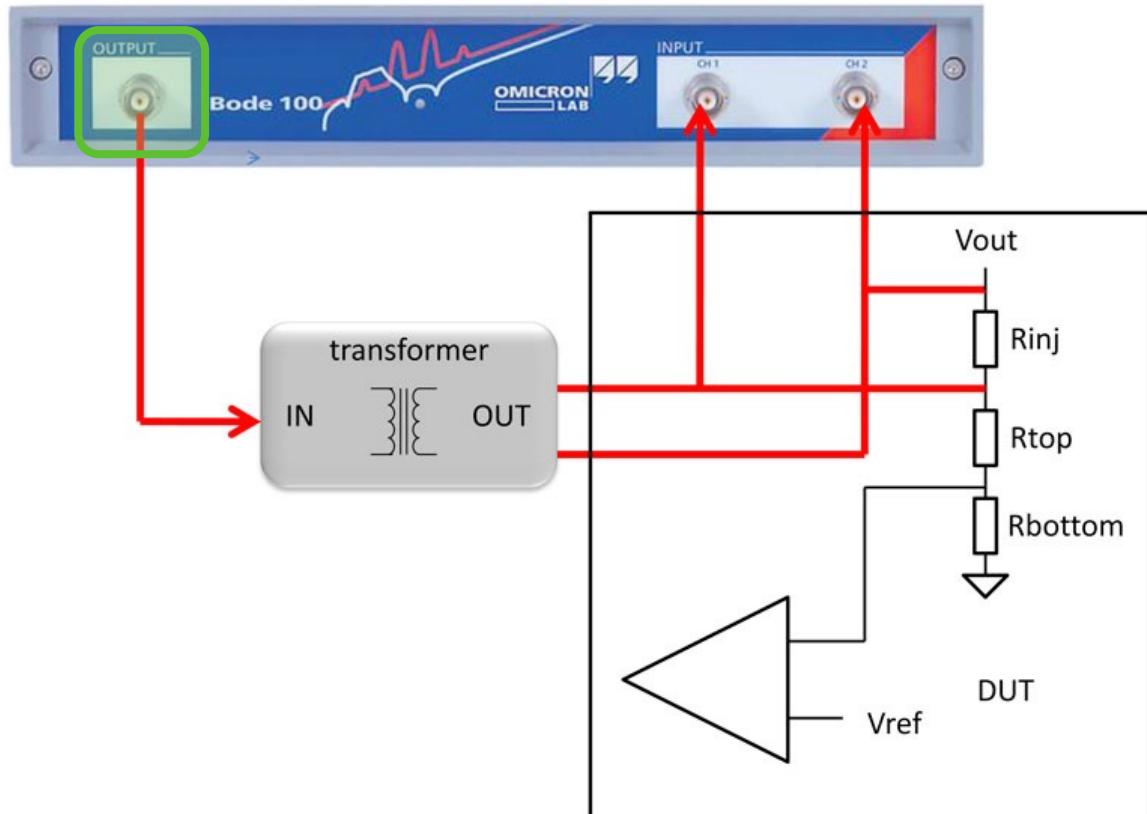
Plant Measurement Using Bode-100



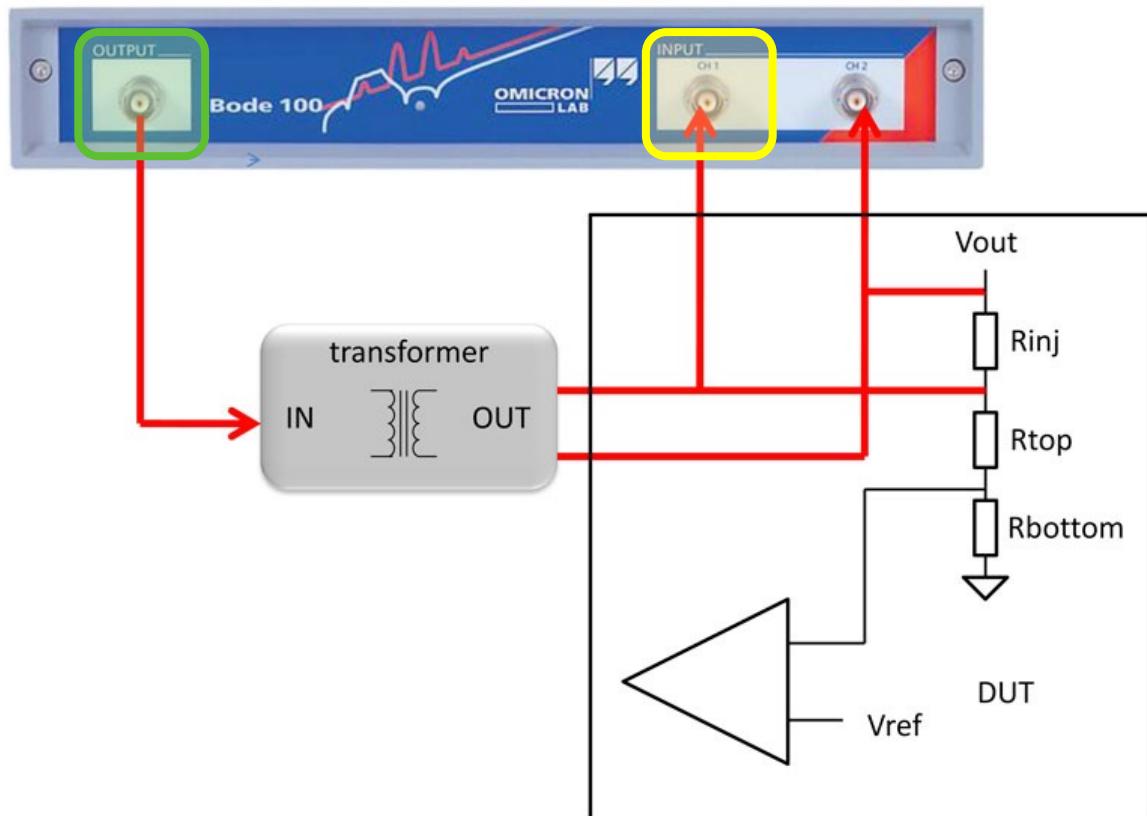
Plant Measurement Using DH0924S



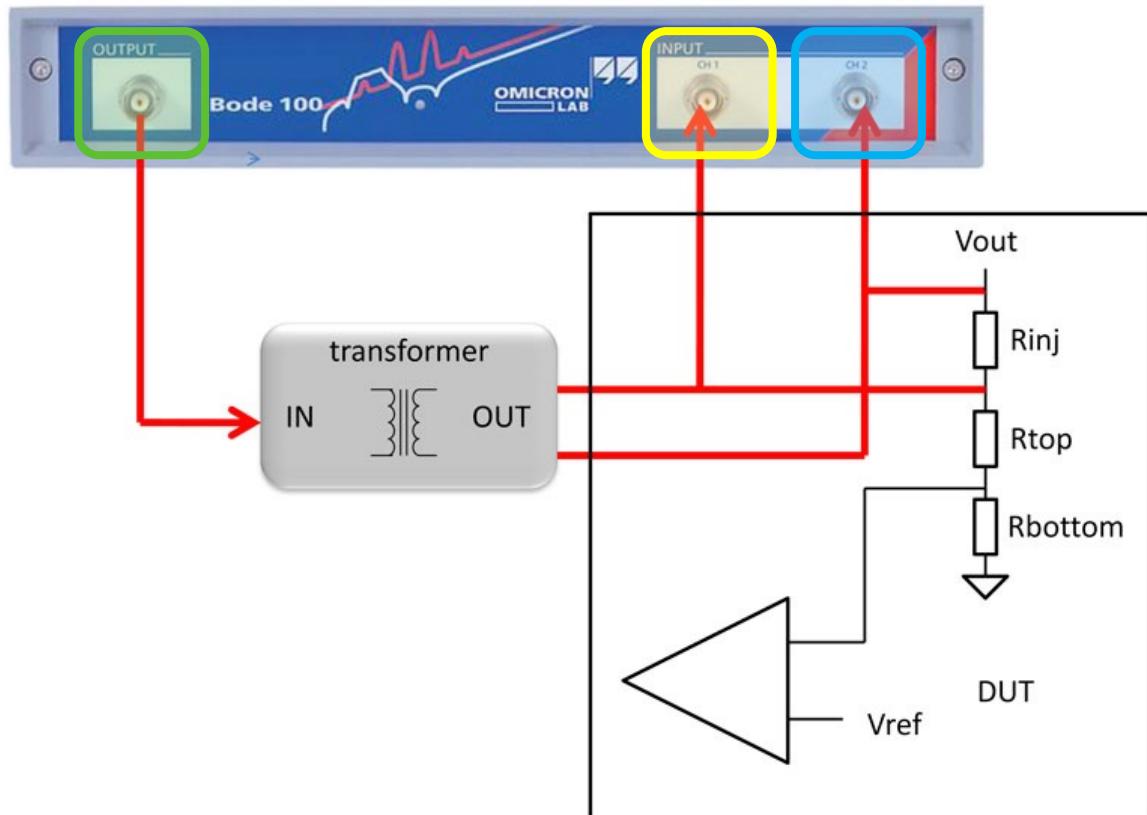
Plant Measurement Using DH0924S



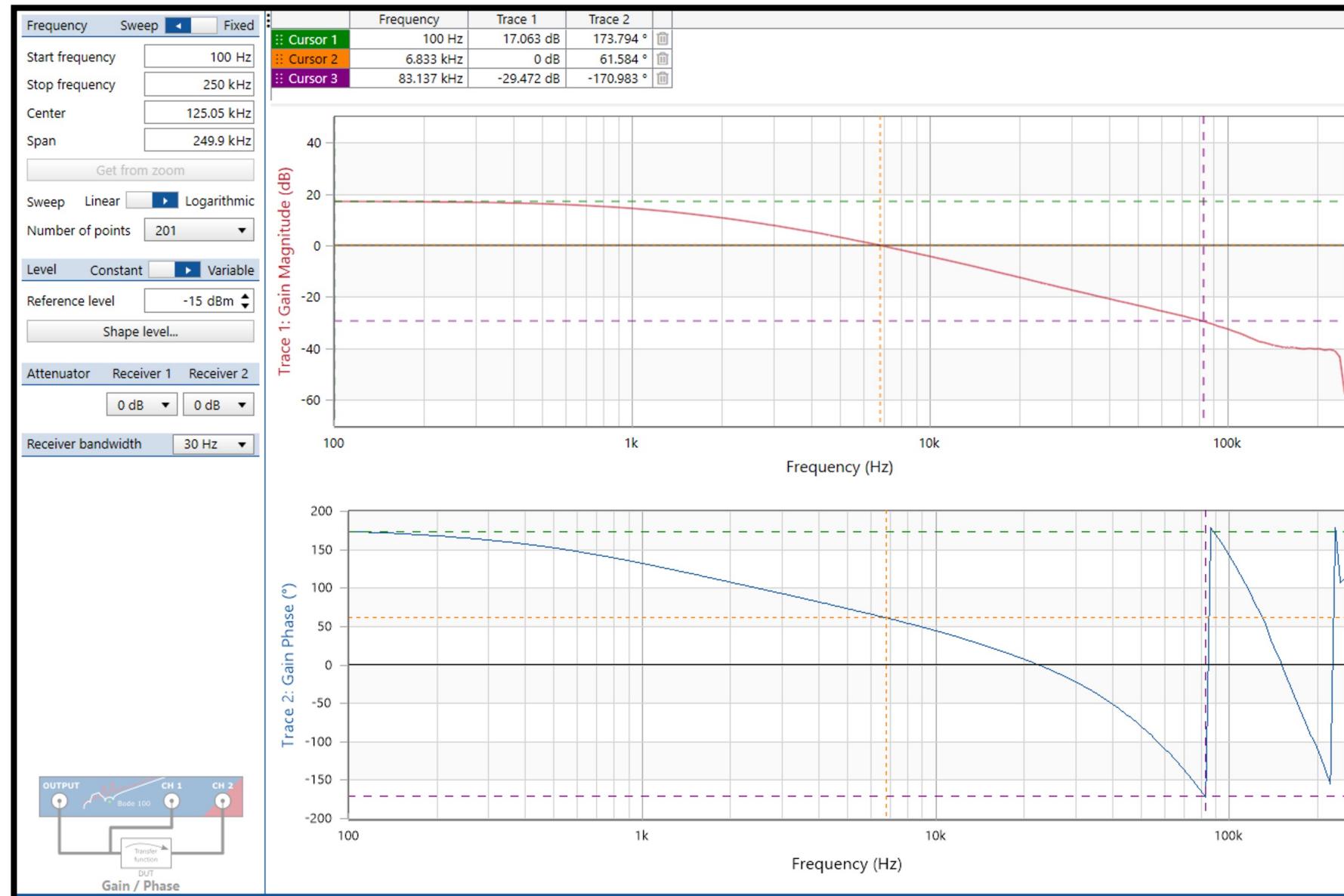
Plant Measurement Using DH0924S



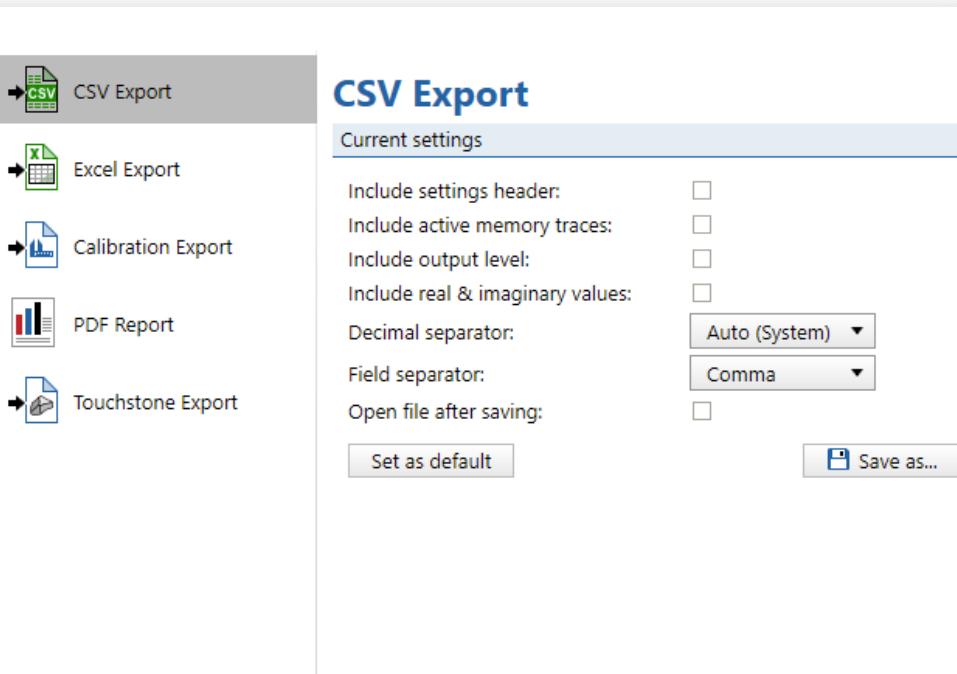
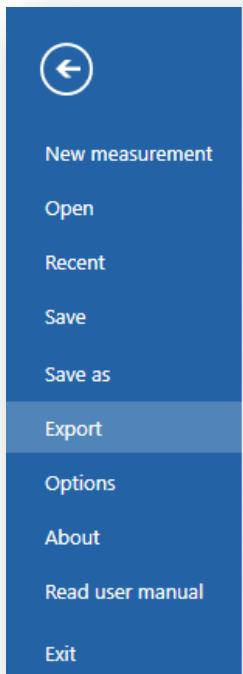
Plant Measurement Using DH0924S



Lab #3: Plant Measurement Using PowerSmart™



Export Bode Plot to CSV



	A	B	C	D
1	Frequency (Hz)	Trace 1: Gain	Trace 2: Gain: Phase (°)	
2	100	17.30047273	169.1247537	
3	103.98955	16.99047568	169.574792	
4	108.138266	16.65394975	170.1127738	
5	112.452496	16.75379993	166.667777	
6	116.938845	17.34375972	171.3902009	
7	121.604179	16.72603985	166.3648023	
8	126.455639	17.30457827	167.1115465	
9	131.50065	17.30183334	167.5736927	
10	136.746935	17.15930463	168.4839062	
11	142.202523	17.48557814	167.3003502	
12	147.875764	16.78895798	166.5875587	
13	153.775342	16.79352515	167.3131347	
14	159.910286	16.84065133	162.3407847	
15	166.289987	16.86709837	161.8181398	
16	172.92421	17.16188517	166.2776363	
17	179.823108	16.58615684	163.233949	
18	186.997242	16.82301431	159.8557044	

Lab #3: Plant Measurement Using PowerSmart™

PowerSmart™ Import Transfer Function

Data Series Name: VPLANT

File Import Simplis/MINDI Bode Plot

Data Source

Filename: U:\POW201\Tools\Bode100\POW201_Lab3_2023-08-12T18_12_16.csv

Series Header Row: 1

Data Start Row: 2

Column Separator: <COMMA>

Phase Rotation

No Rotation

Rotate by -180°

Rotate by +180°

Preview

1	100	17.30047273	169.1247537
2	103.98955	16.99047568	169.574792
3	108.138266	16.65394975	170.1127738
4	112.452496	16.75379993	166.667777
5	116.938845	17.34375972	171.3902009
6	121.604179	16.72603985	166.3648023
7			

Import Data

OK Close

✓ Status: Microchip® project loaded successfully

C)

```
graph LR; VIN --> C[VPLANT Plant Gain]; C --> VOUT; VOUT --> T[Tv(s)];
```

Lab #3: Plant Measurement Using PowerSmart™

The screenshot shows the PowerSmart software interface. On the left, there is a sidebar with various icons and a 'Project Explorer' section. The main window title is 'Import Transfer Function'. In the center, there is a 'Data Series Name' input field containing 'VPLANT'. Below it, there are three tabs: 'File Import' (selected), 'Simplis/MINDI', and 'Bode Plot'. Under 'File Import', there is a 'Data Source' section with a 'Filename' field set to 'U:\POW201\Tools\Bode100\POW201_Lab3_2023-08-12T18_12_16.csv', 'Series Header Row' set to '1', 'Data Start Row' set to '2', and 'Column Separator' set to '<COMMA>'. There are two green checkmarks indicating that a data file has been selected and a valid name for the new data series has been specified. Below this, there is a 'Phase Rotation' section with three radio button options: 'No Rotation', 'Rotate by -180°' (which is selected), and 'Rotate by +180°'. At the bottom of this section, there is a 'Preview' area showing a portion of the CSV data:

	1	100	17.30047273	169.1247537
2	103.98955	16.99047568	169.574792	
3	108.138266	16.65394975	170.1127738	
4	112.452496	16.75379993	166.667777	
5	116.938845	17.34375972	171.3902009	
6	121.604179	16.72603985	166.3648023	
7				

At the bottom right of the preview area is a 'Import Data' button. Below the preview area are 'OK' and 'Close' buttons. A status message at the bottom left says 'Status: Microchip® project loaded successfully'. To the right of the dialog box, there is a block diagram window titled 'C)'. It contains a block labeled 'Converter Voltage Plant G(s)' with 'VIN' on the left and 'VOUT' on the right. A red hand cursor is pointing at the 'G(s)' label. Below this block is the text 'VPLANT Plant Gain'. A blue arrow points from the 'VOUT' terminal of the converter block towards a feedback path labeled 'Tv(s)'.

Lab #3: Plant Measurement Using PowerSmart™

The screenshot shows the PowerSmart software interface. On the left, there is a sidebar with various icons and a 'Project Explorer' section. The main window title is 'Import Transfer Function'. In the center, there is a 'Data Series Name' input field containing 'VPLANT'. Below it, there are three tabs: 'File Import' (selected), 'Simplis/MINDI', and 'Bode Plot'. Under 'File Import', there is a 'Data Source' section with a 'Filename' field set to 'U:\POW201\Tools\Bode100\POW201_Lab3_2023-08-12T18_12_16.csv', a 'Series Header Row' of '1', a 'Data Start Row' of '2', and a 'Column Separator' of '<COMMA>'. There are two green checkmarks indicating that a data file has been selected and a valid name for the new data series has been specified. Below this, there is a 'Phase Rotation' section with three radio button options: 'No Rotation', 'Rotate by -180°' (which is selected), and 'Rotate by +180°'. To the right of this section is a preview window showing a portion of the CSV data:

	1	100	17.30047273	169.1247537
2	103.98955	16.99047568	169.574792	
3	108.138266	16.65394975	170.1127738	
4	112.452496	16.75379993	166.667777	
5	116.938845	17.34375972	171.3902009	
6	121.604179	16.72603985	166.3648023	
7				

At the bottom of the dialog, there is an 'Import Data' button with a red hand cursor icon pointing to it, and 'OK' and 'Close' buttons.

To the right of the dialog, there is a block diagram window titled 'C)'. It contains a block labeled 'Converter Voltage Plant G(s)' with 'VIN' at the top and 'VOUT' at the bottom. A blue arrow points from the 'VOUT' terminal of this block to the 'VOUT' terminal of a 'Plant Gain' block. A 'T_{v(s)}' block is connected to the 'VOUT' terminal of the 'Plant Gain' block. The entire 'VPLANT' block is highlighted in yellow.

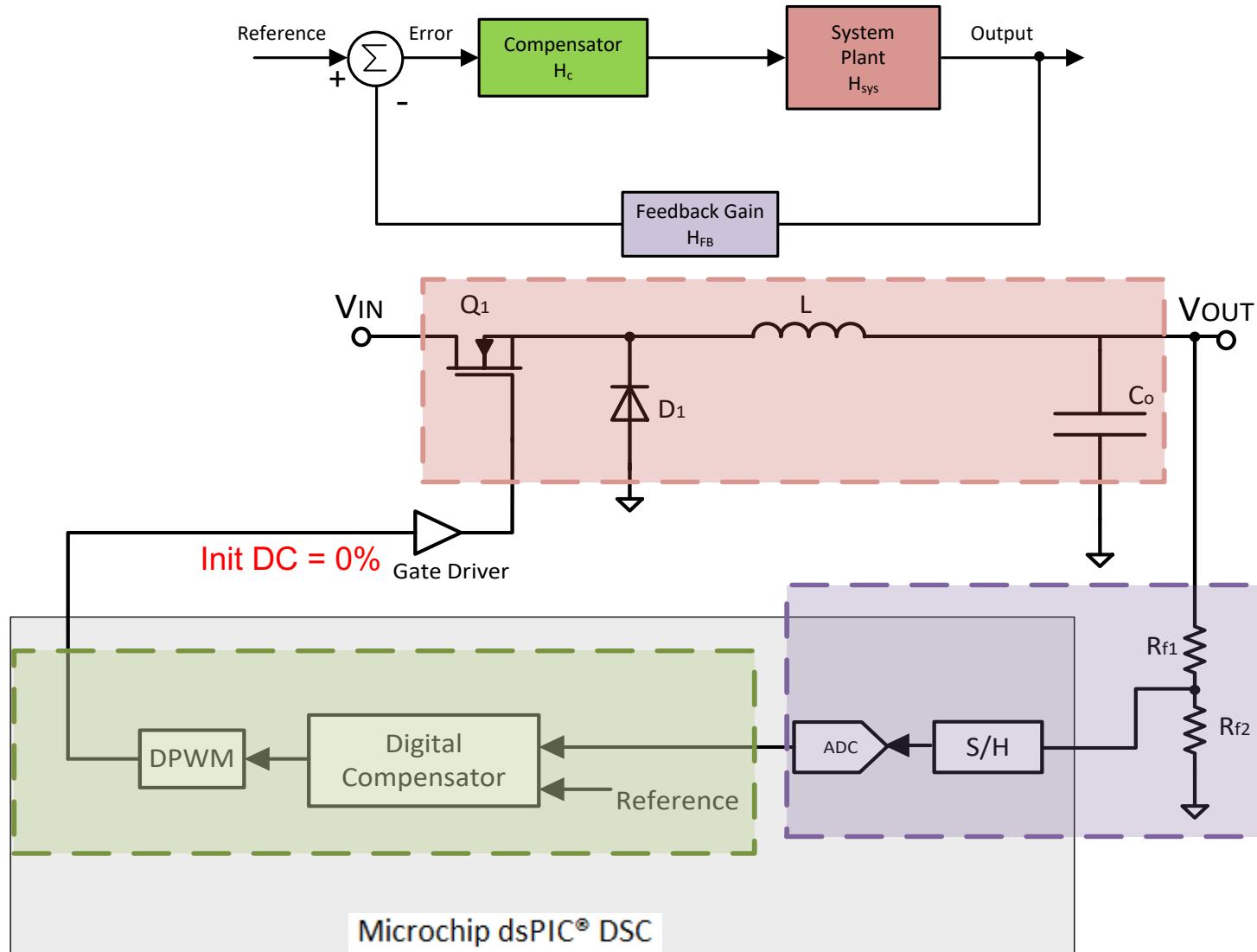
NOW, Actual Plant is Imported!!!



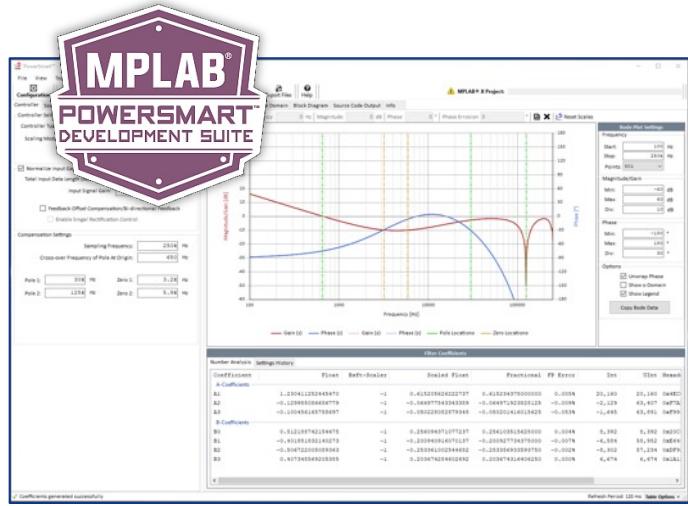
Closed-loop Control

Lab4

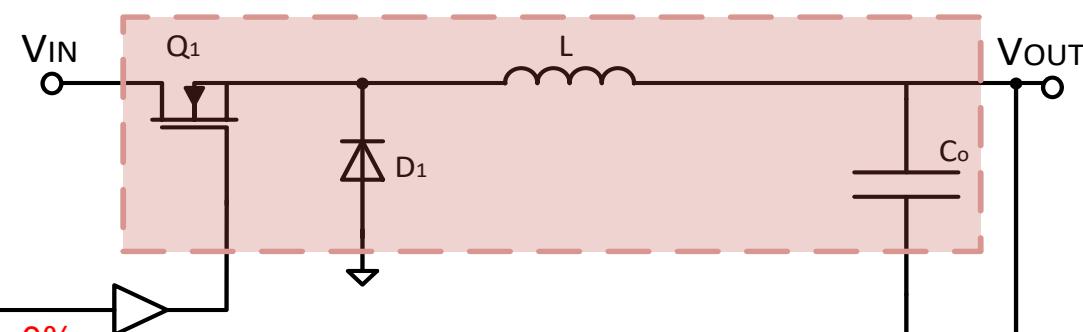
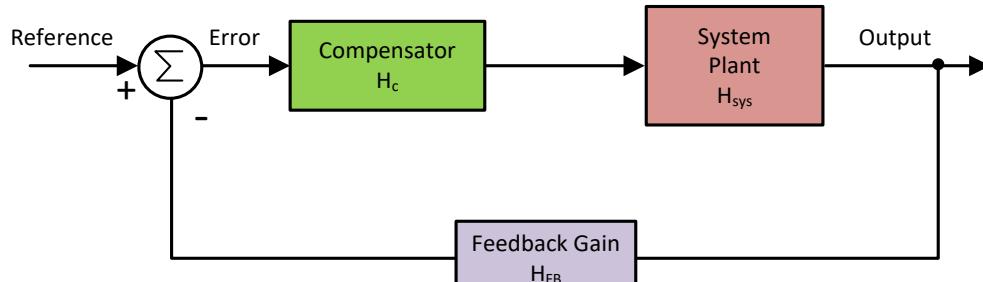
Lab #4: Closed-loop Control



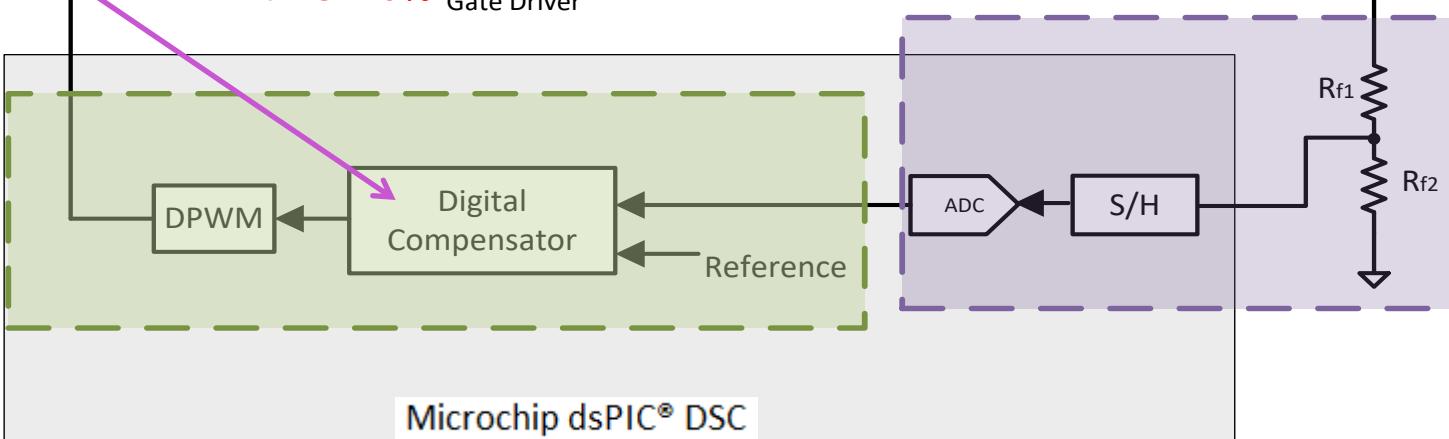
Lab #4: Closed-loop Control



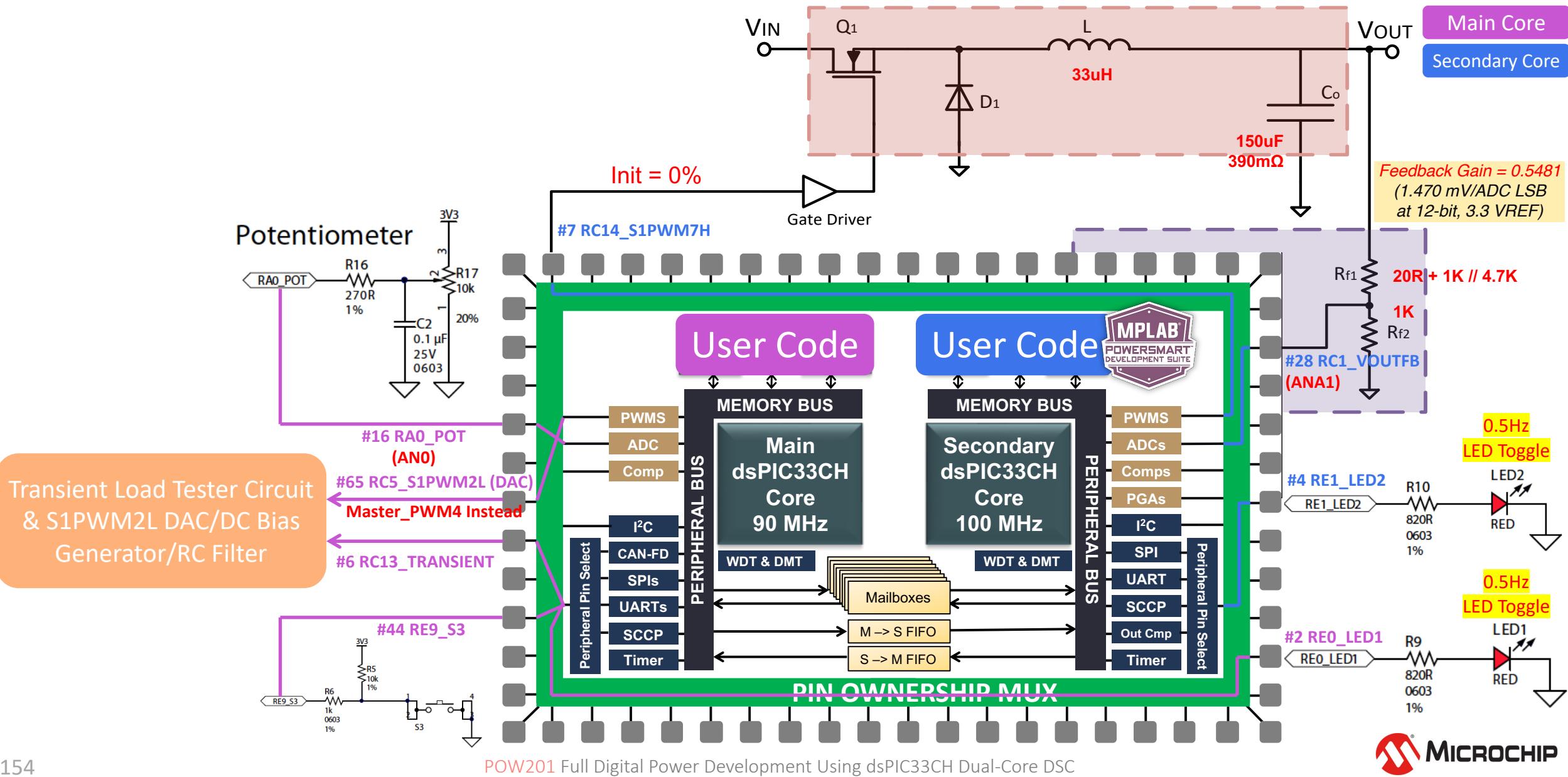
Closed-Loop Control



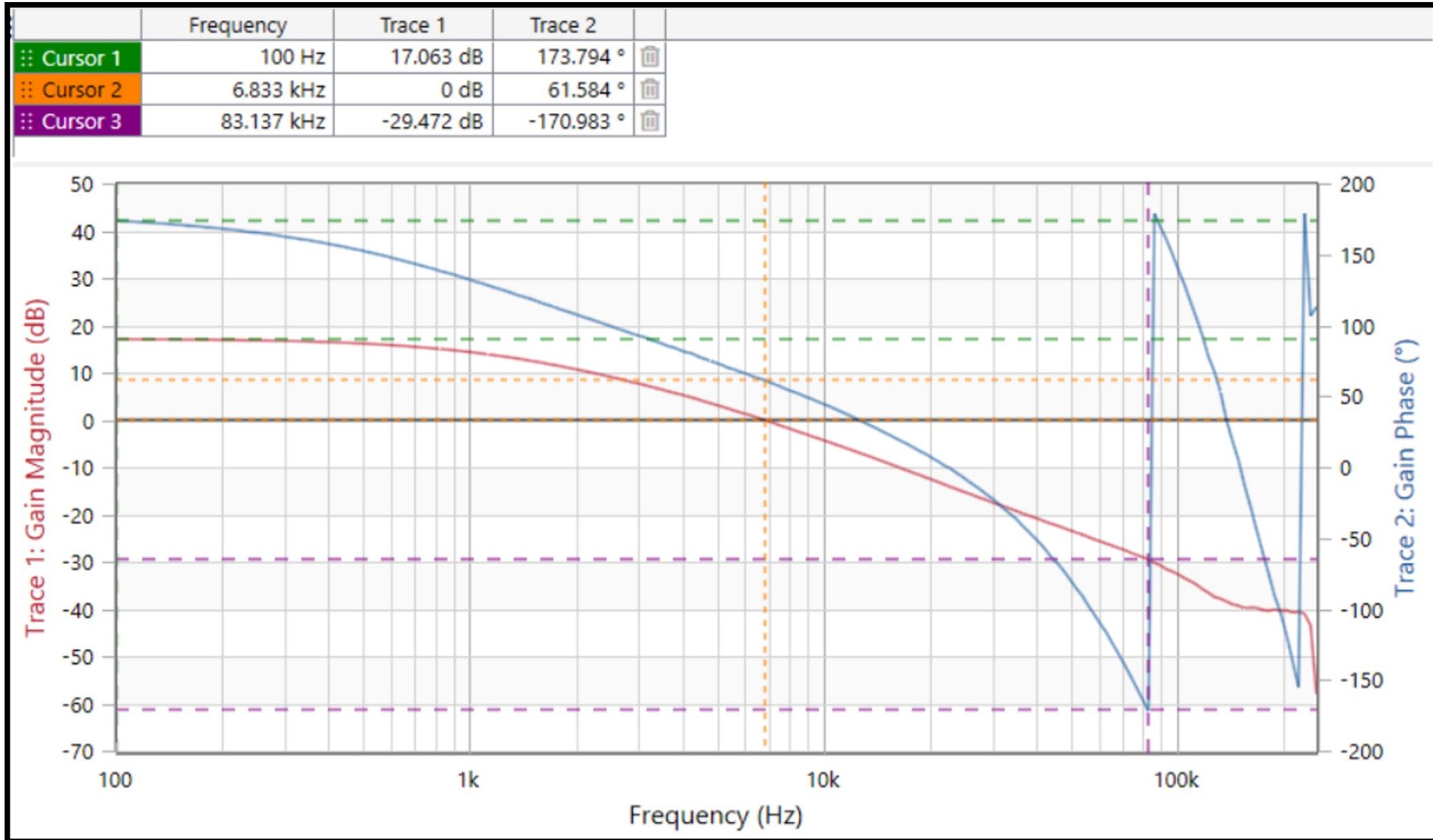
Init DC = 0%  Gate Driver



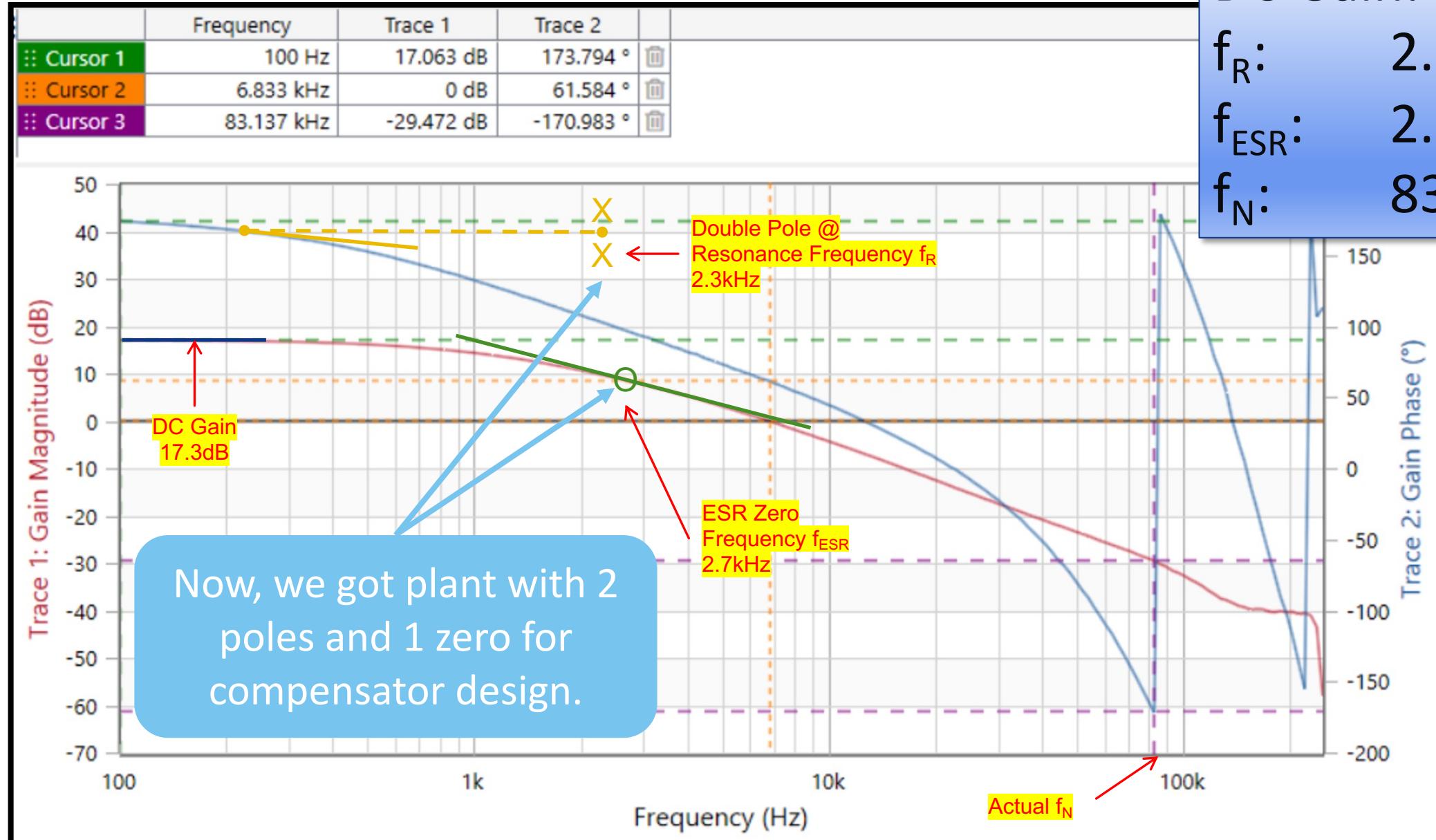
Lab #4: Closed-loop Control



The Actual Plant

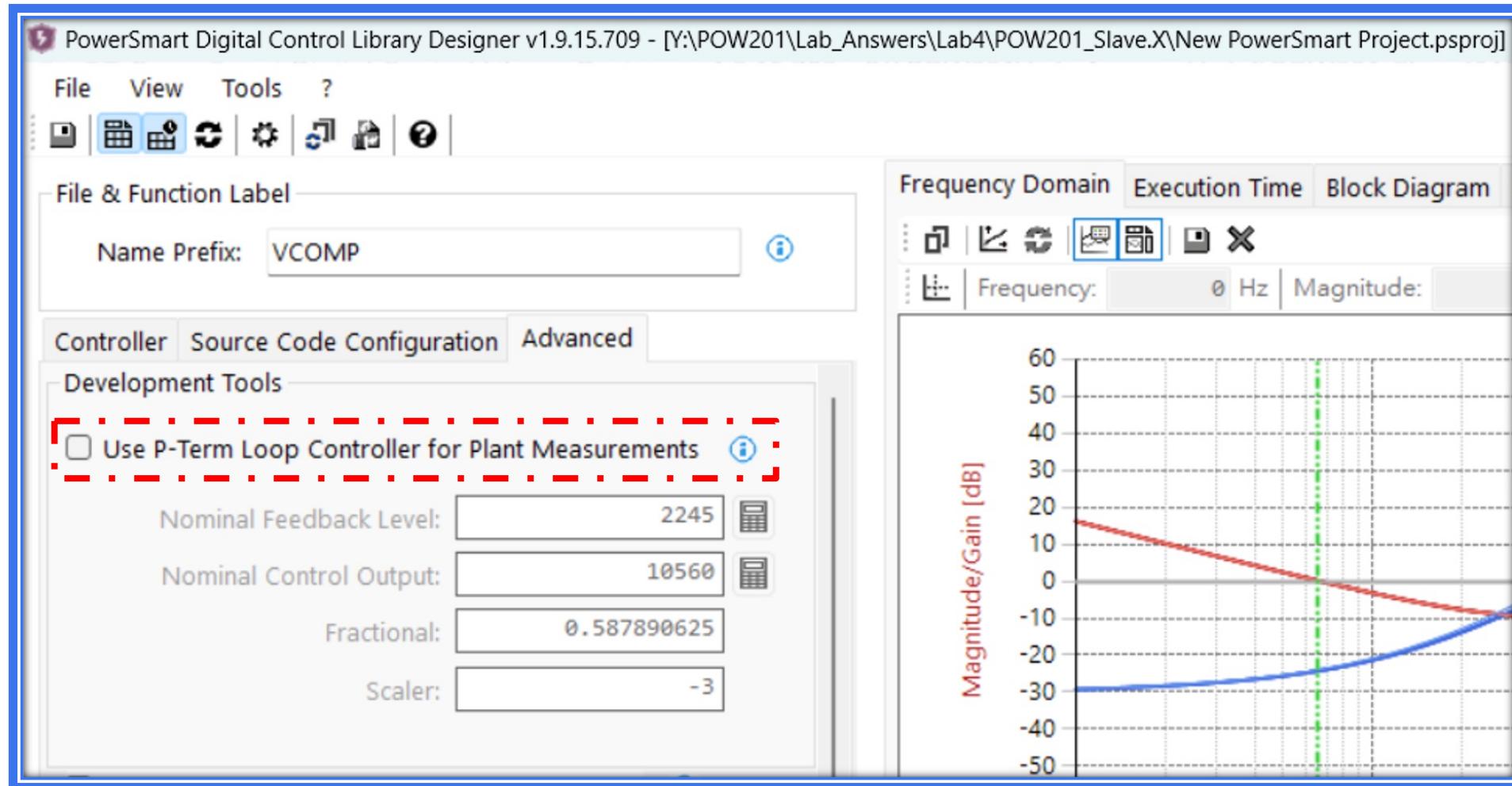


The Actual Plant

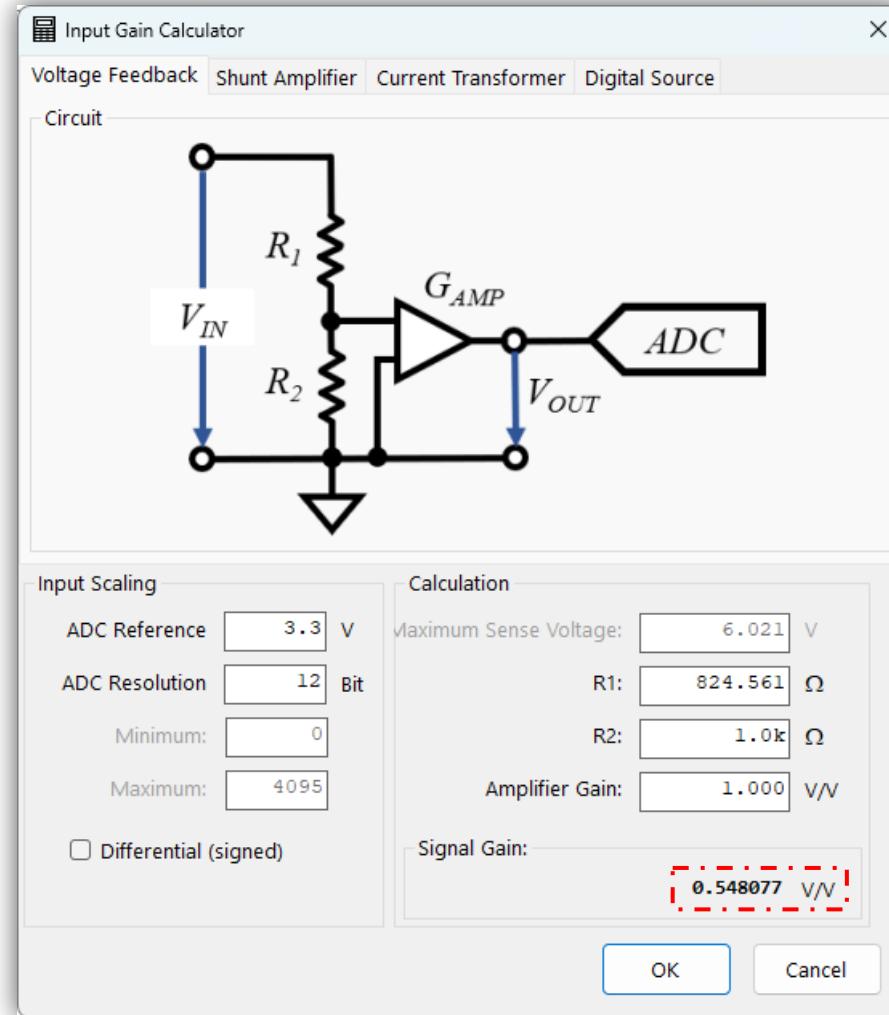
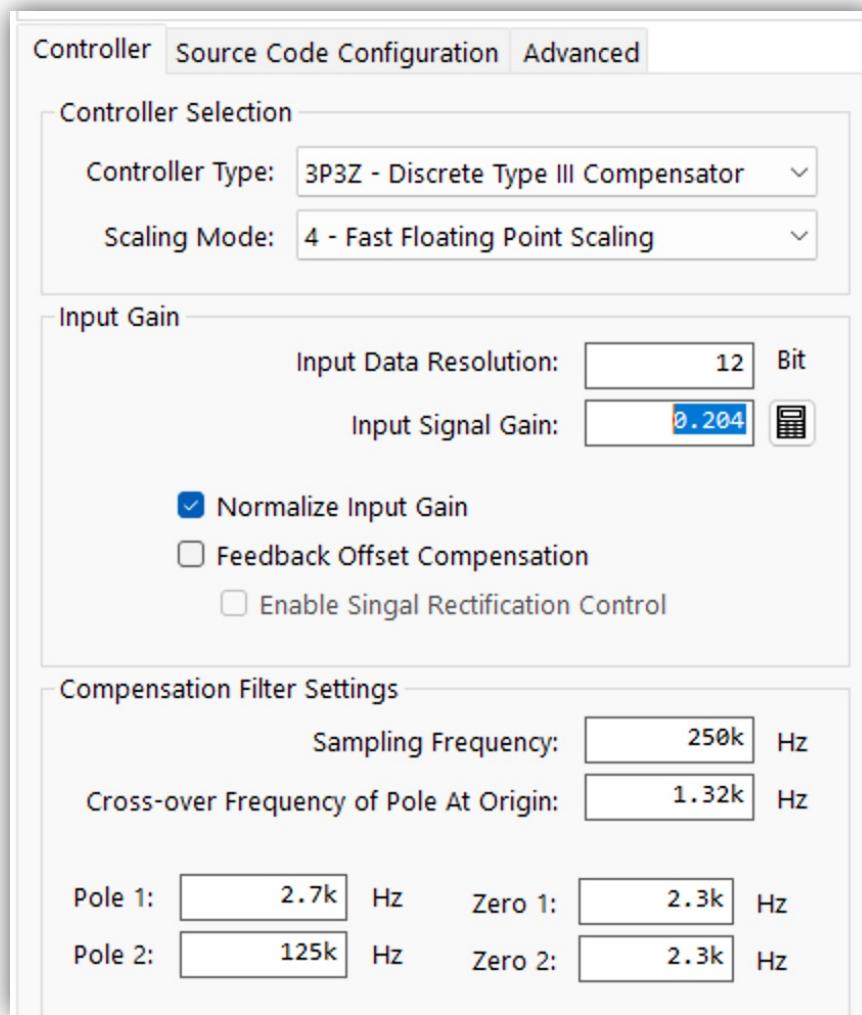


Uncheck Kp (P-Term) Control on DCLD

- **Uncheck P-Term loop controller if needed to remove the code.**



Place Poles-Zeros

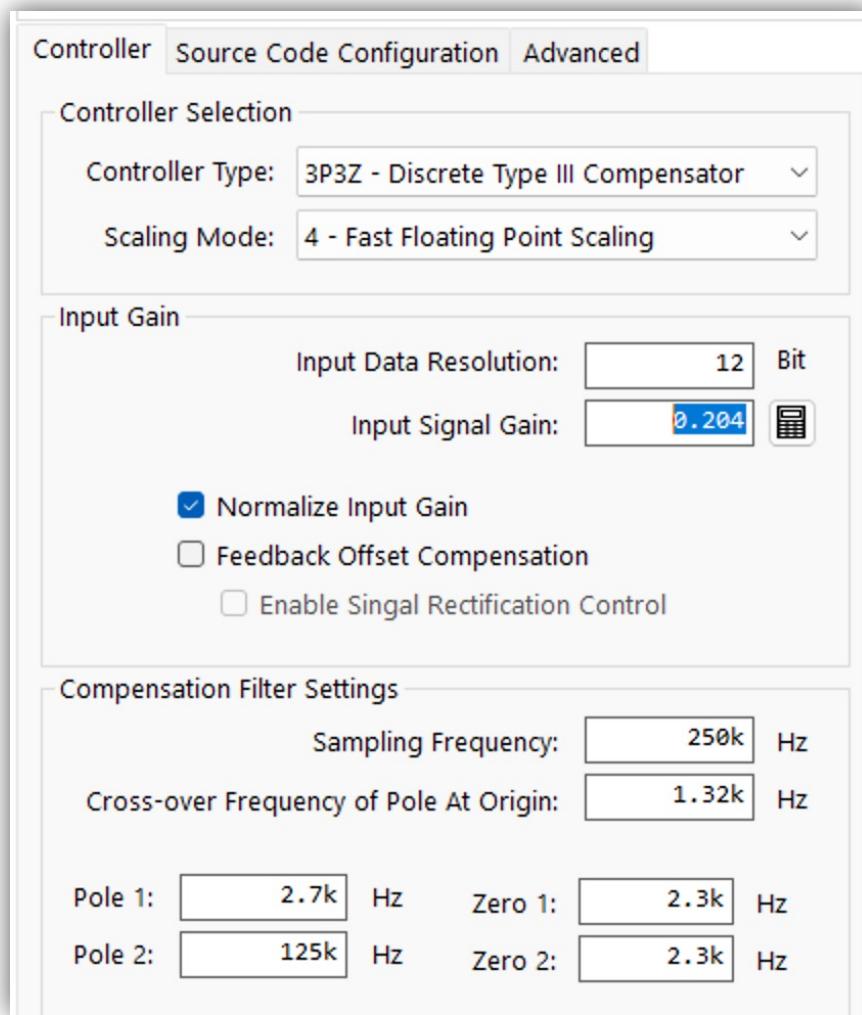


$$\begin{aligned} K_{FB} &= 1 * R_1 / (R_1 + R_2) \\ &= 0.4519 \end{aligned}$$

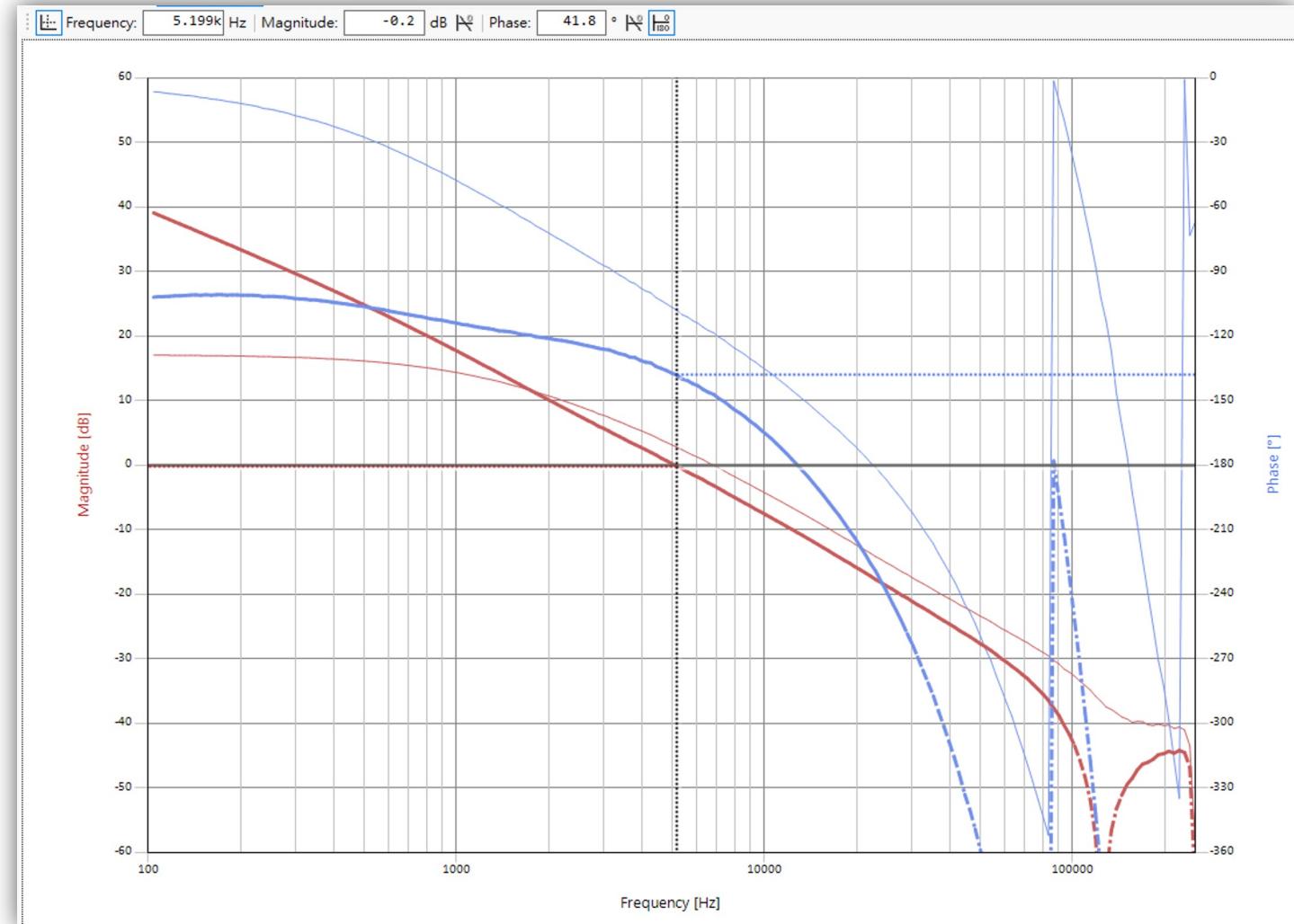
$$\begin{aligned} \text{Input Signal Gain} &= K_{FB} * K_{FB} \\ &= 0.204 \end{aligned}$$

DCLD

Place Poles-Zeros



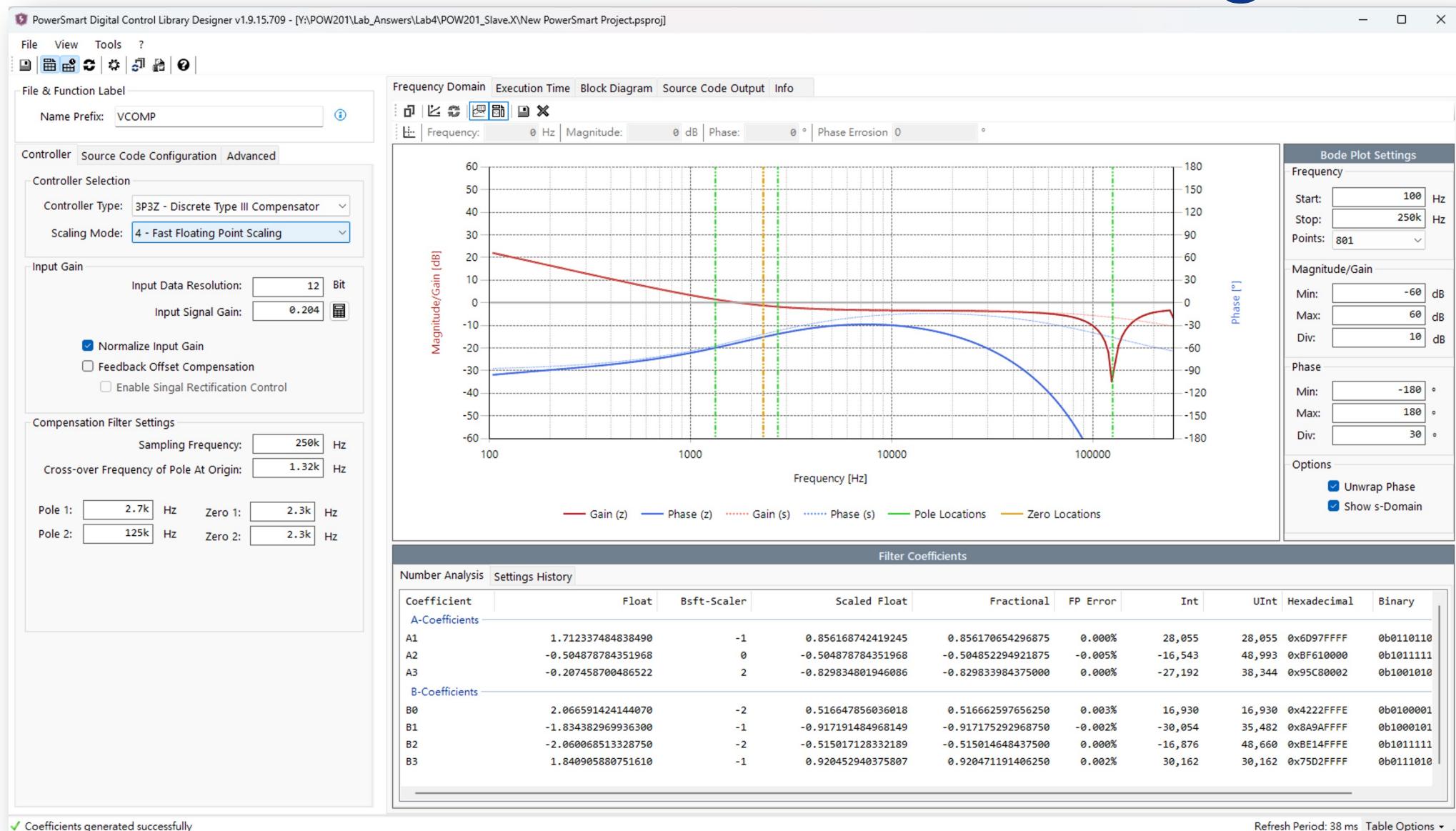
Observe Open Loop Gain



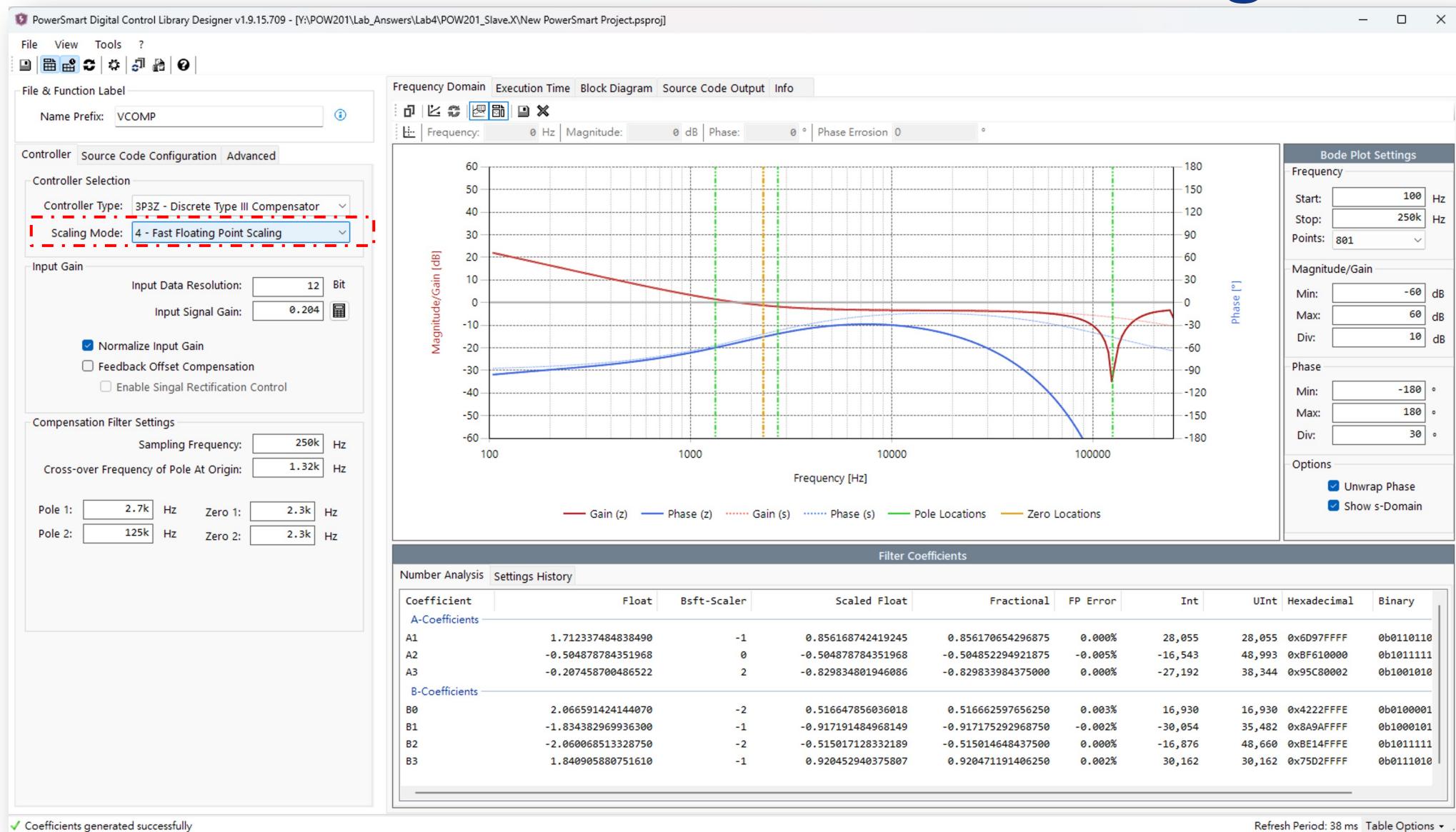
DCLD

PowerSmart™

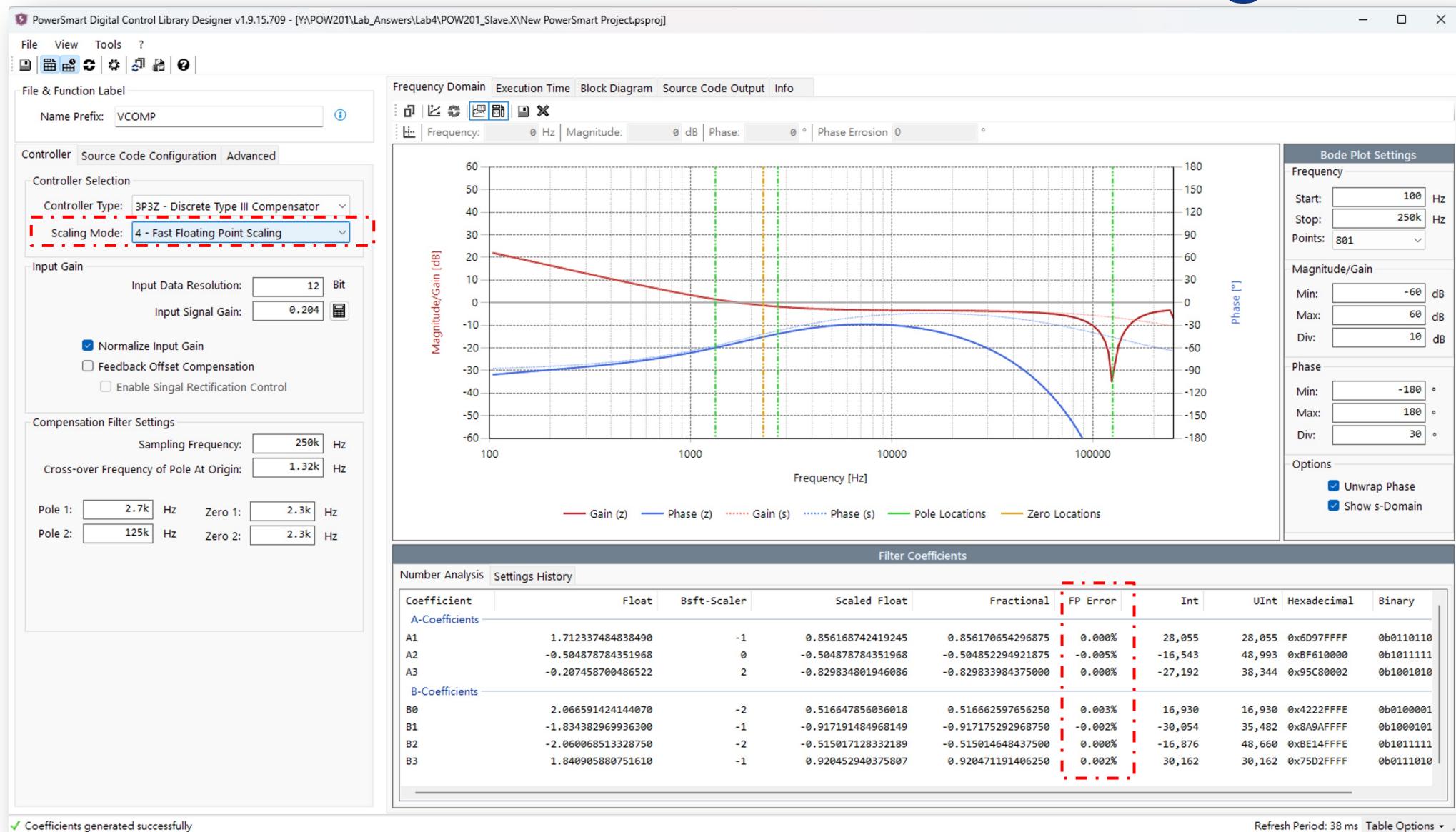
Check Coefficients with Suitable Scaling Mode



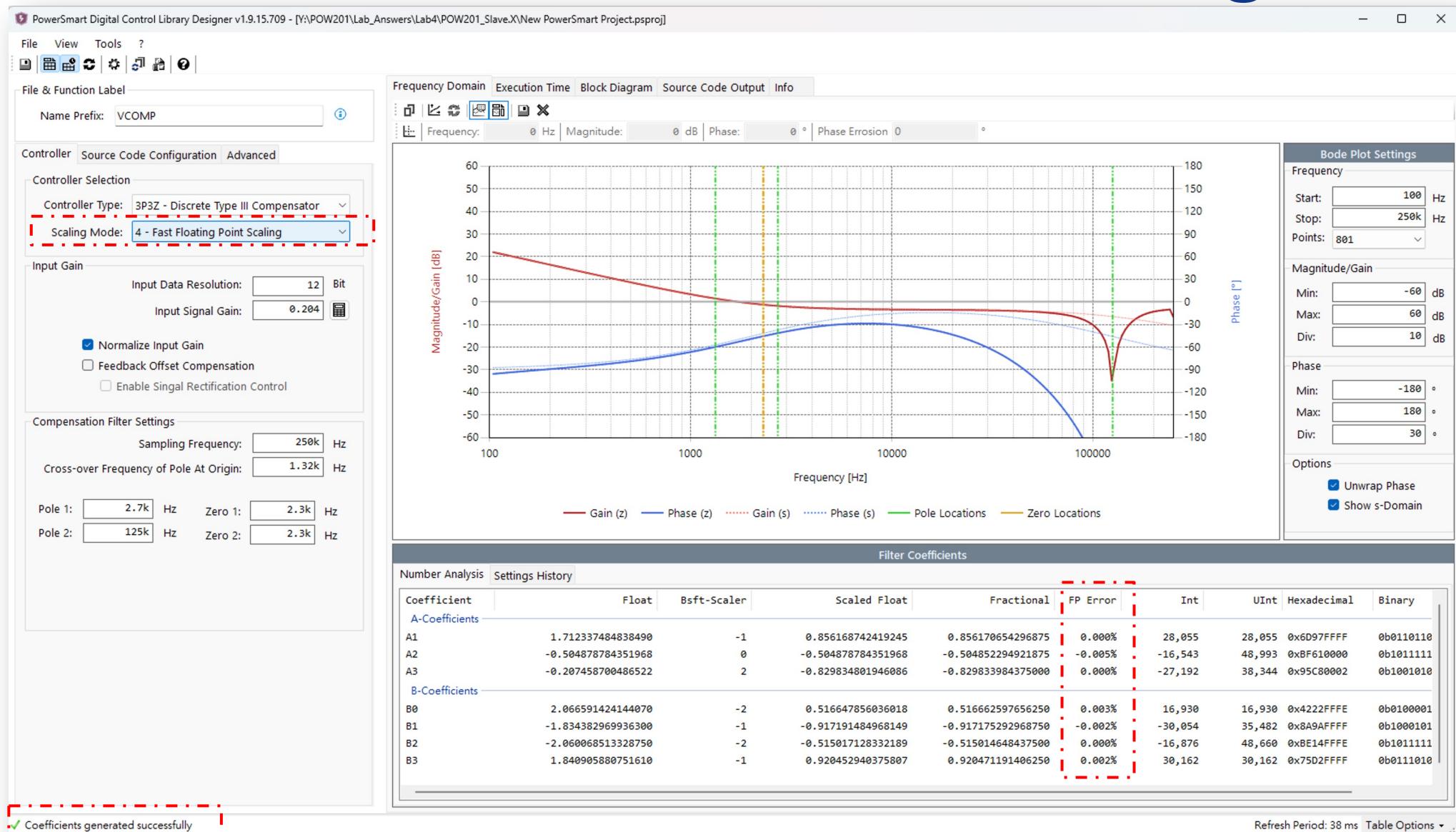
Check Coefficients with Suitable Scaling Mode



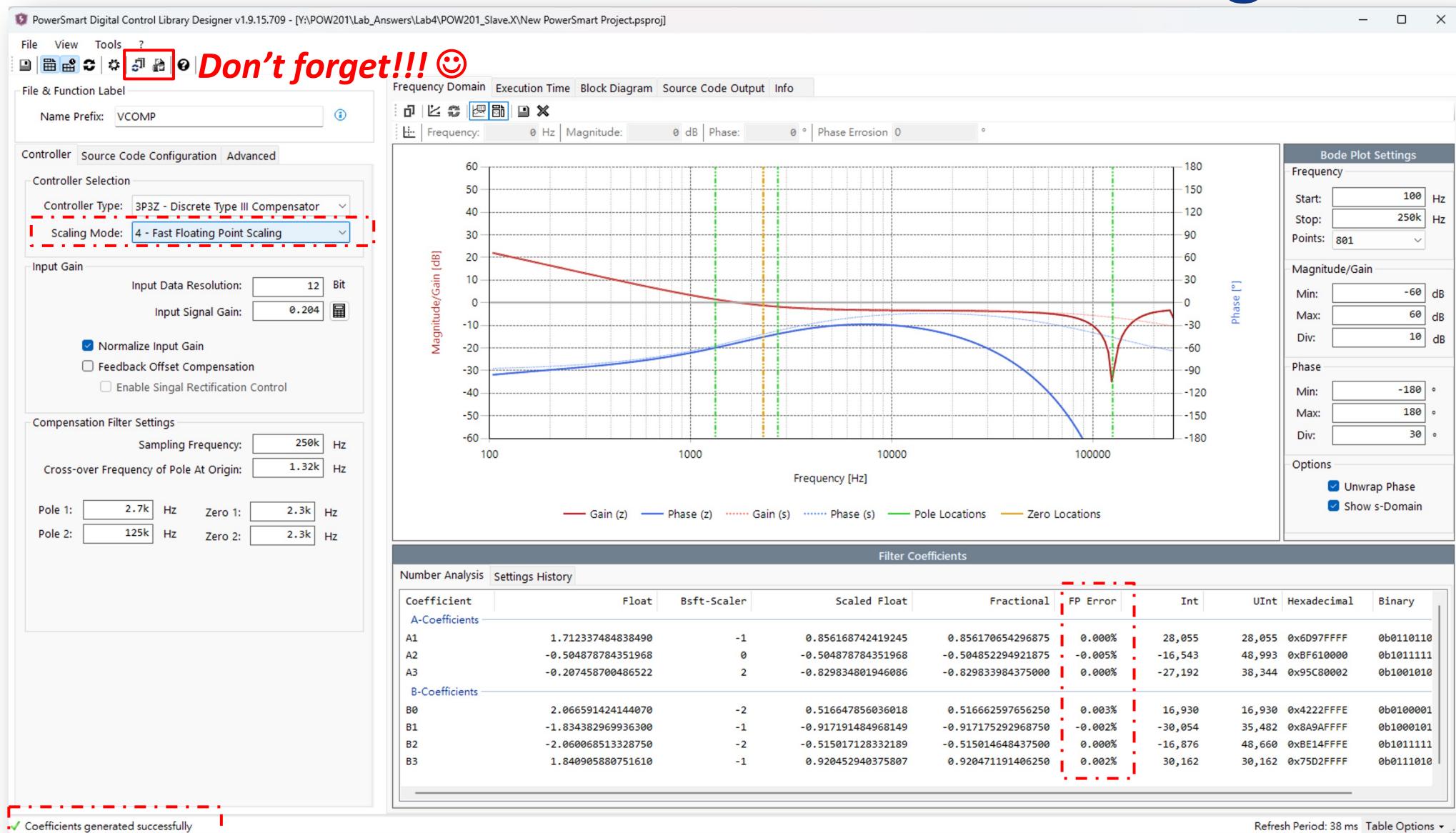
Check Coefficients with Suitable Scaling Mode



Check Coefficients with Suitable Scaling Mode



Check Coefficients with Suitable Scaling Mode



Lab #4: Closed-loop Control

Switch to closed-loop control in AN1 ISR

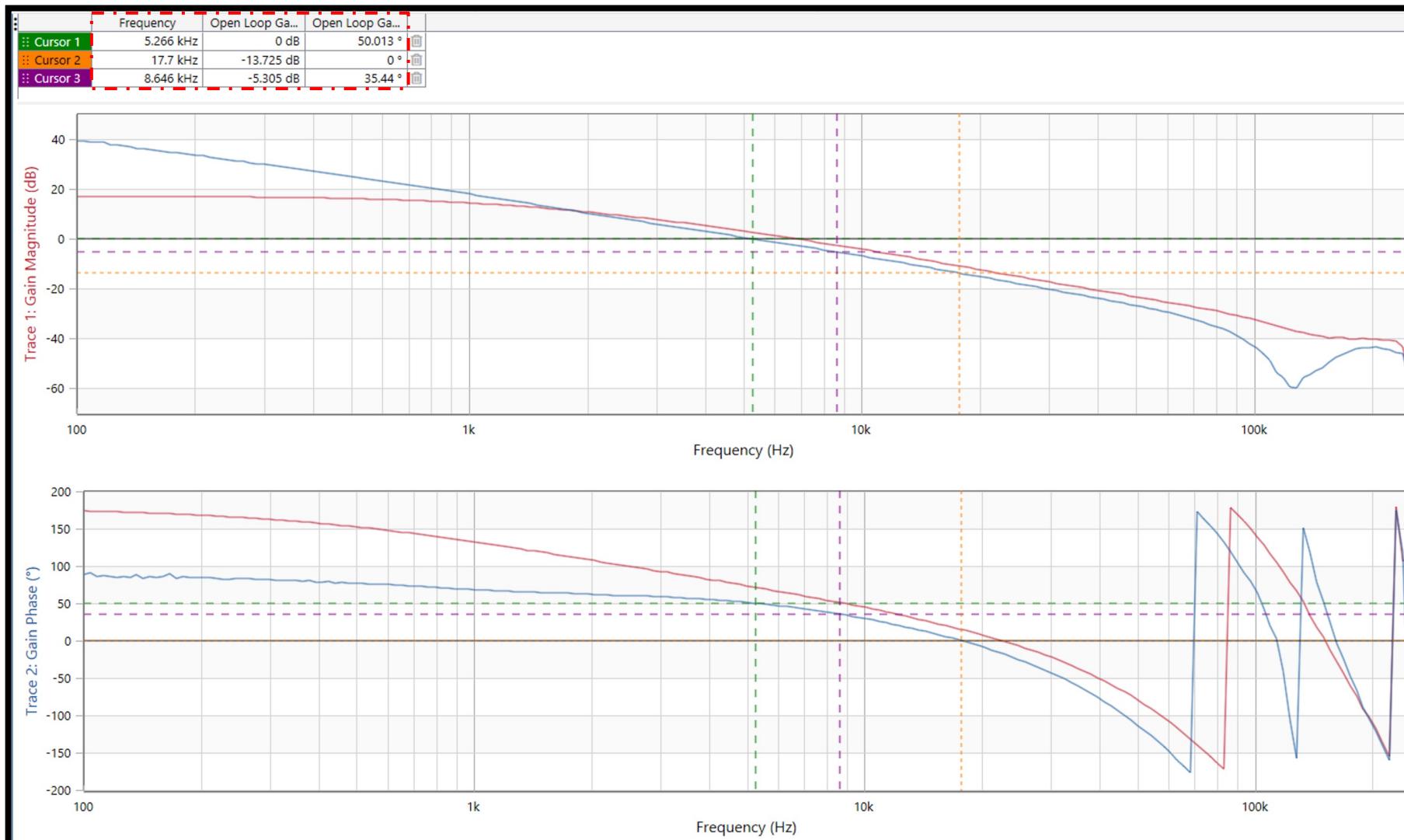
The screenshot shows a software development environment with a project tree on the left and a code editor on the right.

Project Tree:

- POW201_Master
 - Header Files
 - Important Files
 - Linker Files
 - Source Files
 - Libraries
 - Loadables
 - Secondaries
 - POW201_Slave
- POW201_Slave
 - Header Files
 - Important Files
 - Linker Files
 - Source Files
 - driver
 - PowerController
 - BuckConverter.c
 - VCOMP.c
 - VCOMP_asm.s
 - main.c
 - main_tasks.c
- MCC Generated Files
- os
- Libraries
- Loadables

Lab #4: Closed-loop Control

Bode Plot Measurement

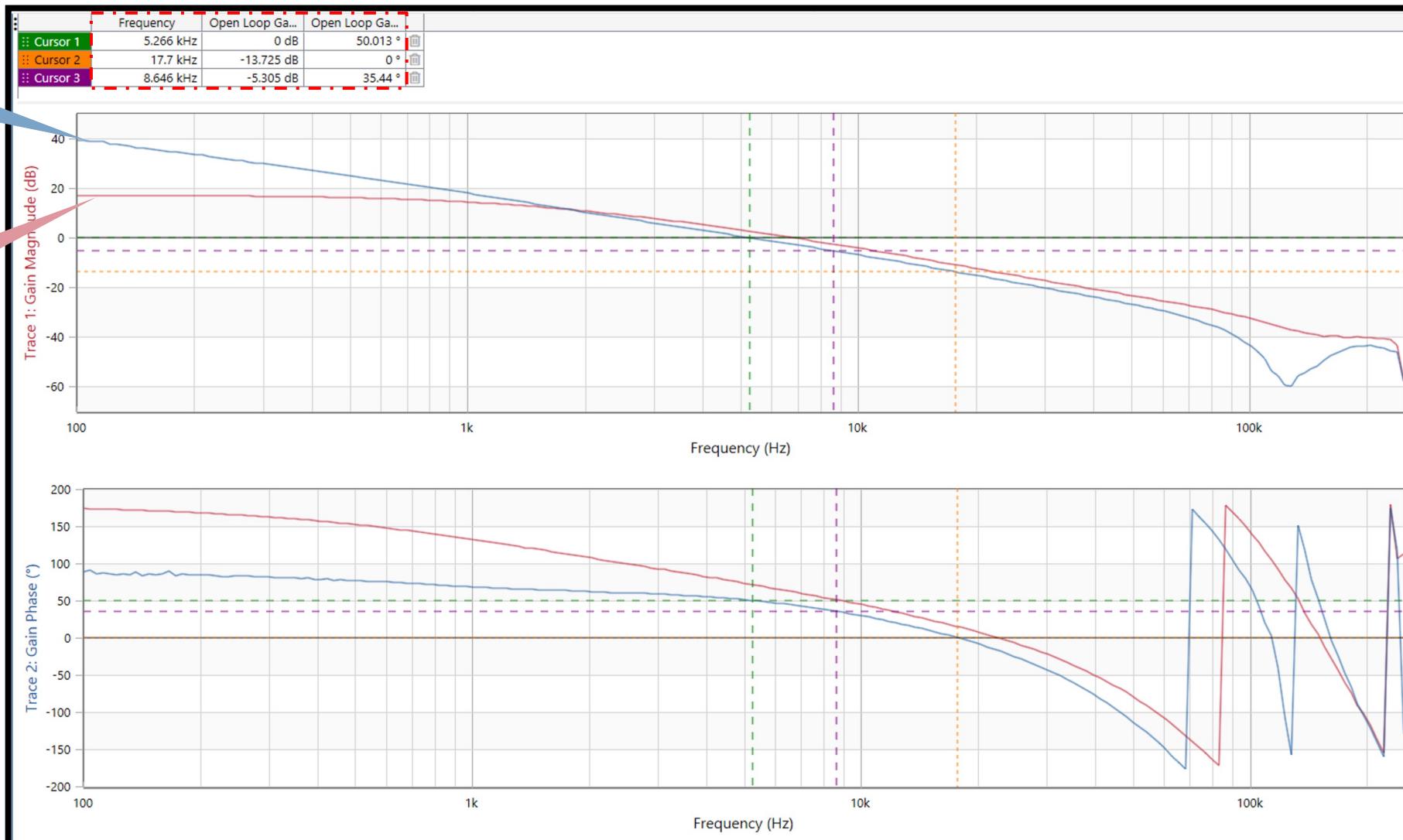


Lab #4: Closed-loop Control

Bode Plot Measurement

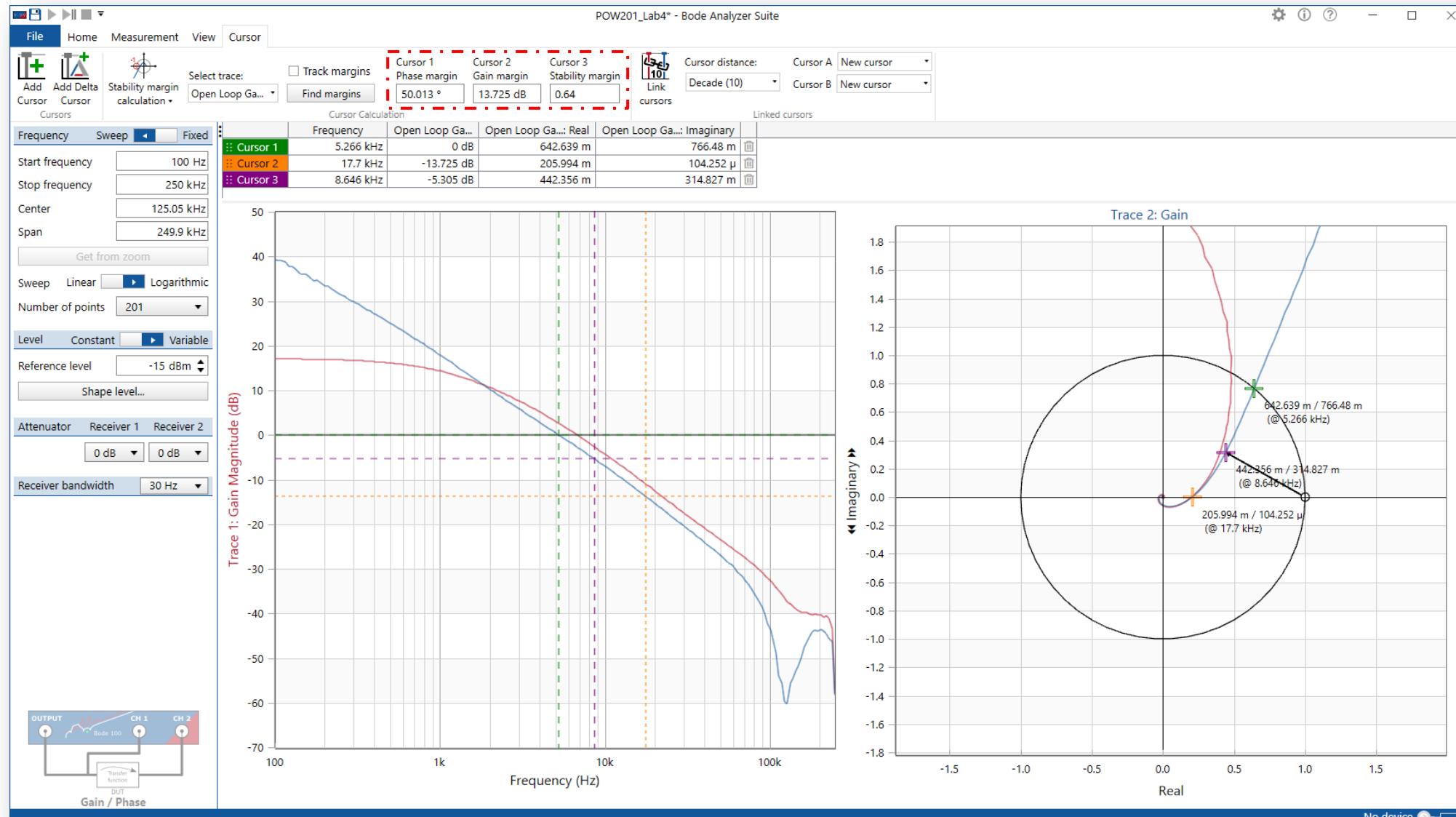
OL TF

Plant



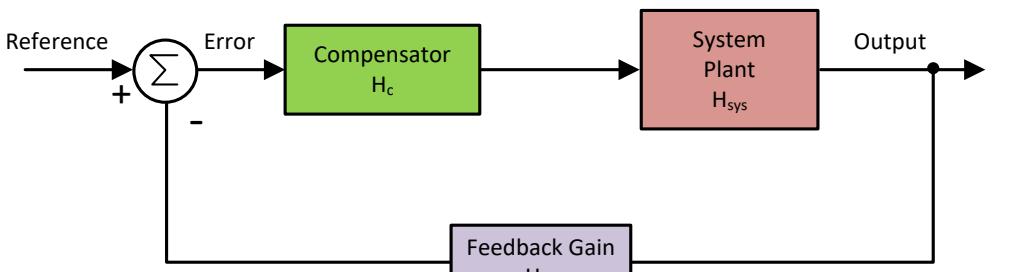
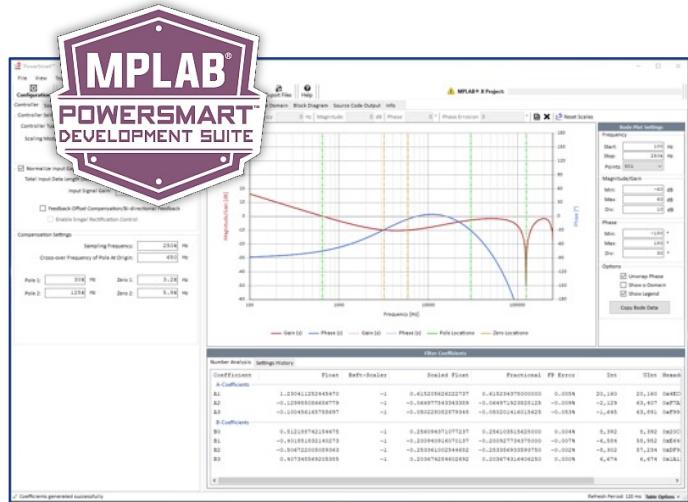
Lab #4: Closed-loop Control

Stability Margin & Nyquist Plot



Now, You have known that ...

Step2. Plant Modeling



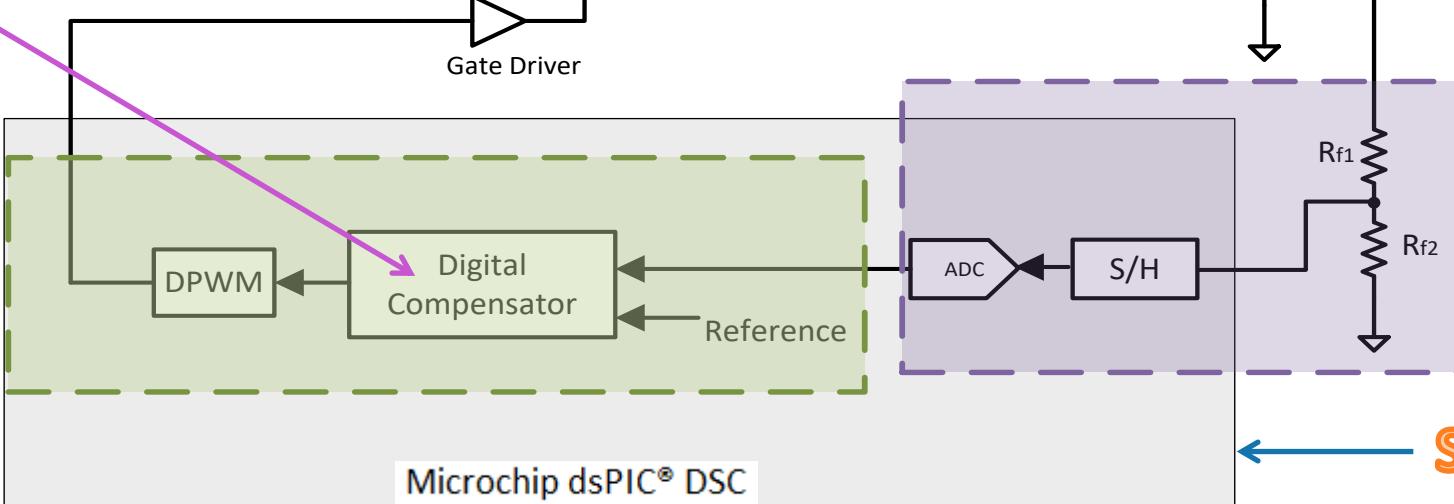
Simulation

OR



Measurement Using P-term Control

Step3. Closed-Loop Control



Step1. MCC Codebase



May The *Power* Be With You

**KNOWLEDGE IS
POWER**

Massive power density in the smallest packages

Thanks!

