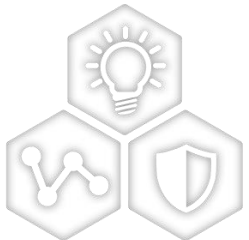




dsPIC33 Digital Signal Controllers



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



SMART | CONNECTED | SECURE

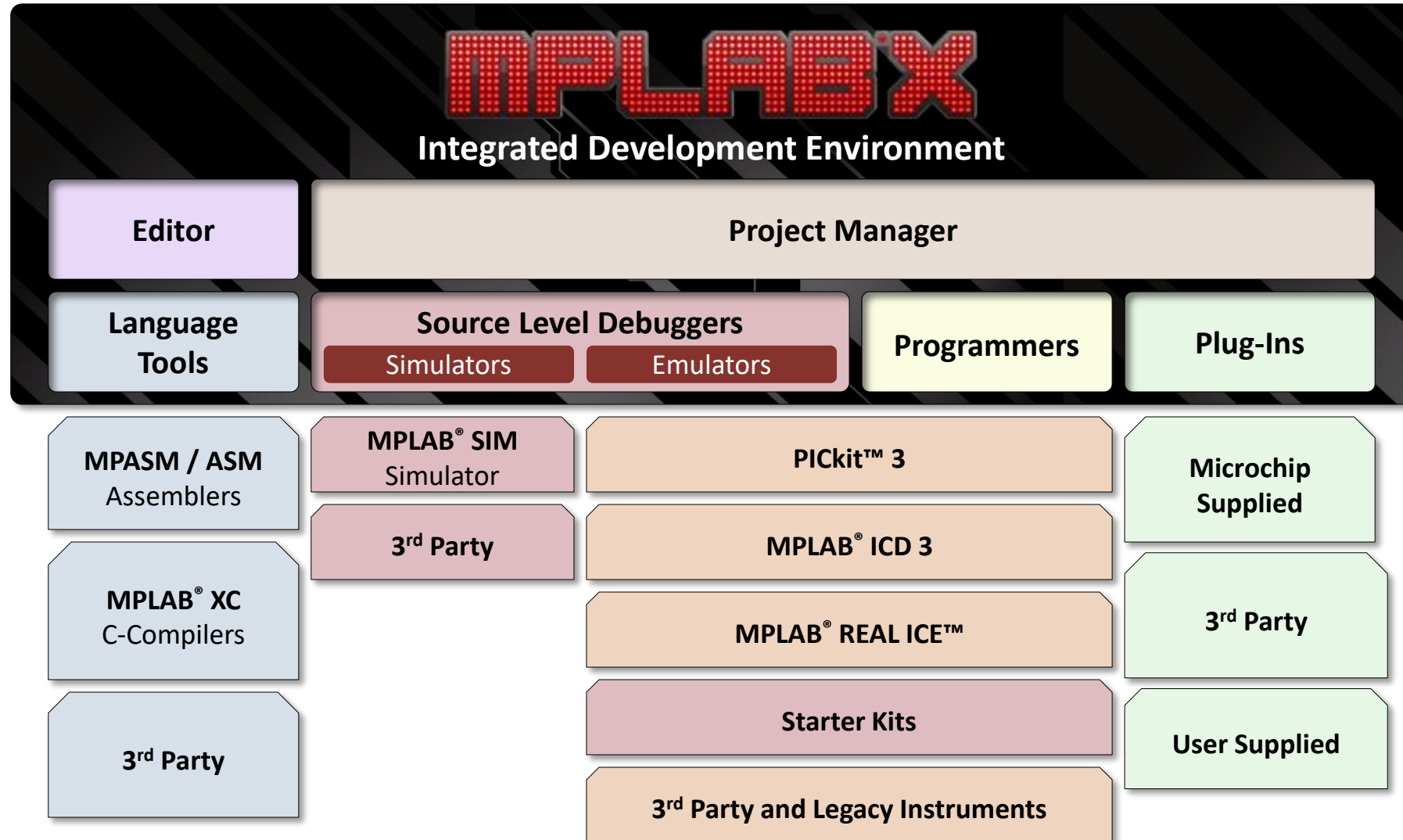


Power
Conversion

Andreas Reiter
November 15, 2022

Introducing the MPLAB[®] X Ecosystem

MPLAB® X IDE Overview



MPLAB[®] XC Compilers

Hi-Tech PICC Based	MPLAB C30 / GCC Based	MPLAB C32 / GCC Based	MPLAB C32 / GCC Based
MPLAB[®] XC8 <i>8-bit Compiler</i>	MPLAB[®] XC16 <i>16-bit Compiler</i>	MPLAB[®] XC32 <i>32-bit Compiler</i>	MPLAB[®] XC32++ <i>32-bit Compiler</i>
PIC10, PIC12 PIC16, PIC18 ATtiny	PIC24 dsPIC30 dsPIC33	PIC32/SAMx	PIC32/SAMx

Free	No cost, production worthy, optimizing compiler, community support
Standard	Entry level price, more optimization, access to priority support
Professional	Full price, whole program optimization, access to priority support

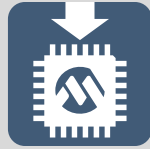
MPLAB X Ecosystem

Programmer/Debuggers

Microcontroller Design Environment



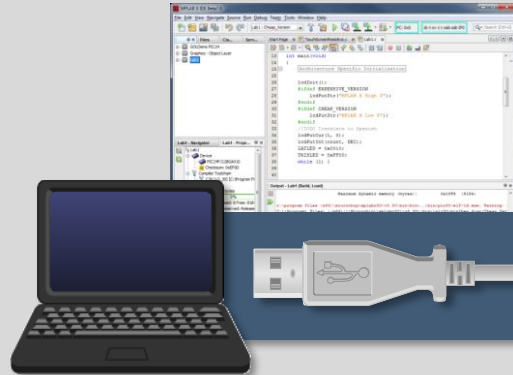
Integrated
Development
Environment



Programmer
Debugger



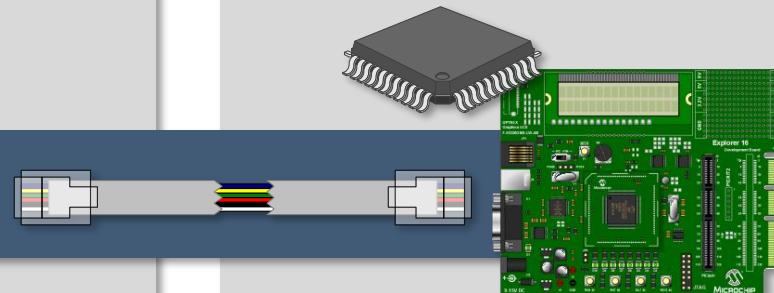
Target
Hardware



MPLAB® X IDE
C Compiler
C++ Compiler
MPASM Assembler



MPLAB ICE™
MPLAB ICD 4
MPLAB PICKIT™ 4



Curiosity Boards
Explorer 16
Starter Kits
Your Hardware...

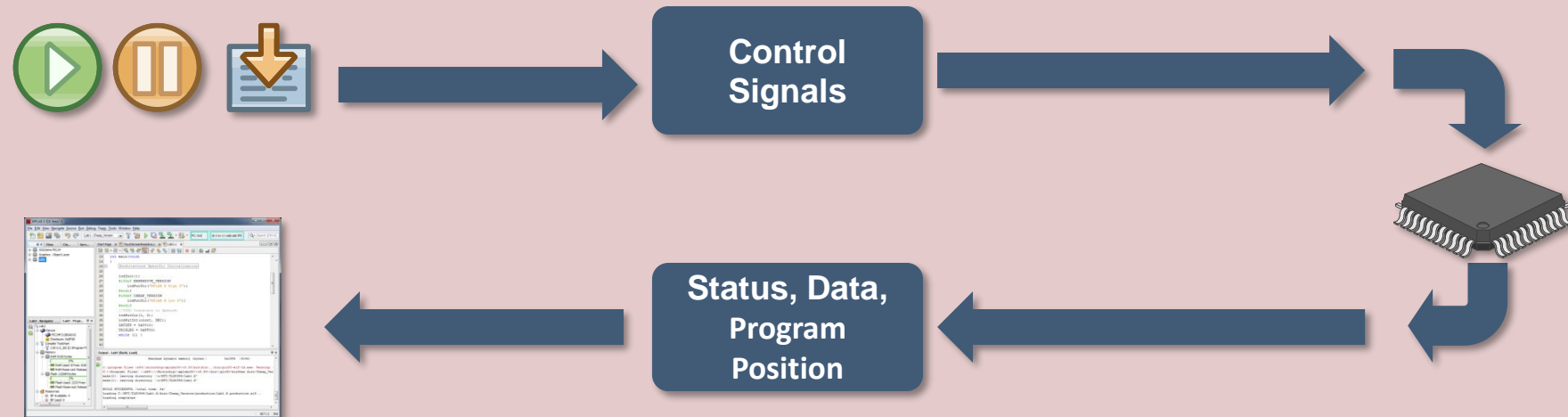
What is a Programmer/Debugger?



Programming Function

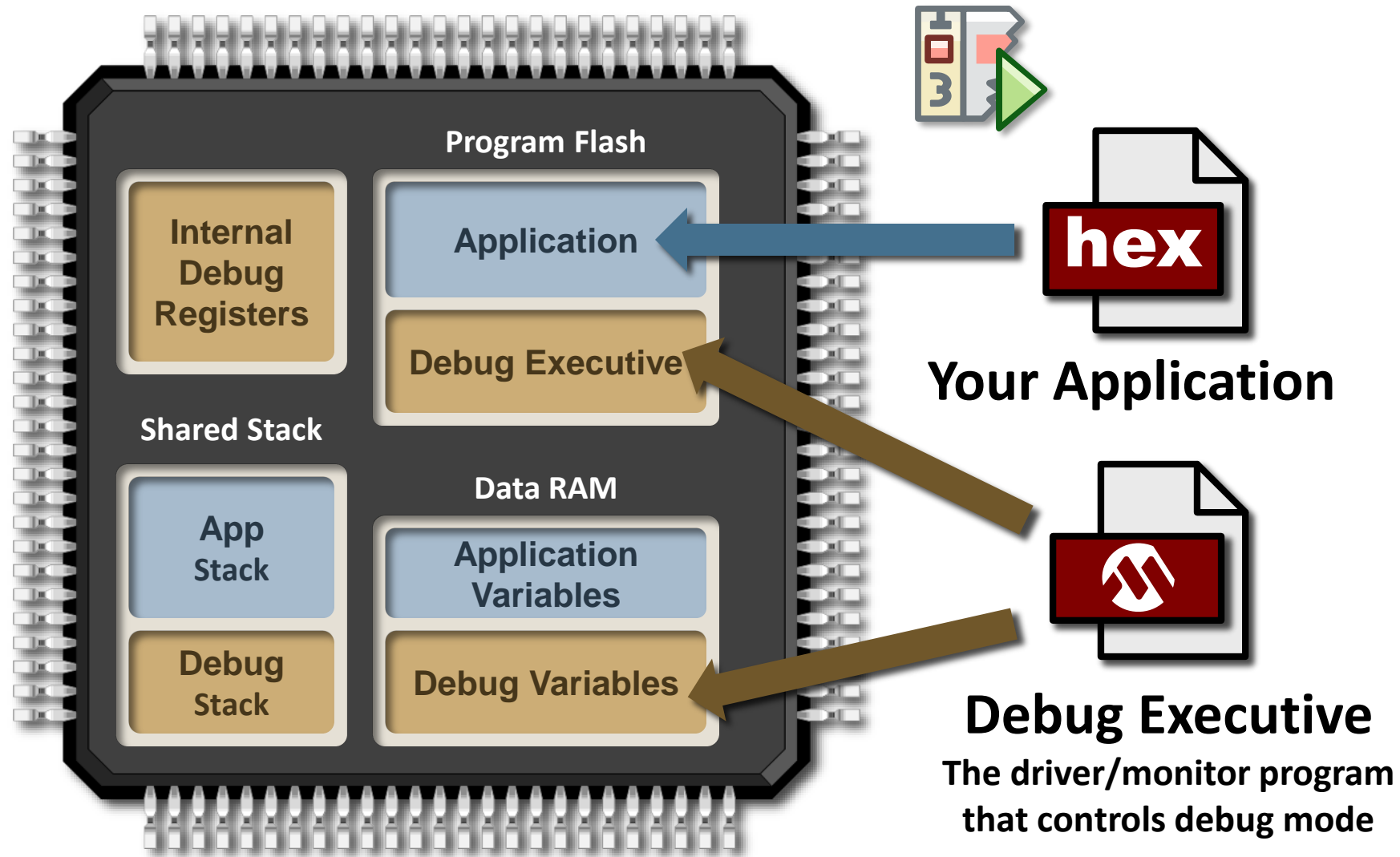


Debugging Function



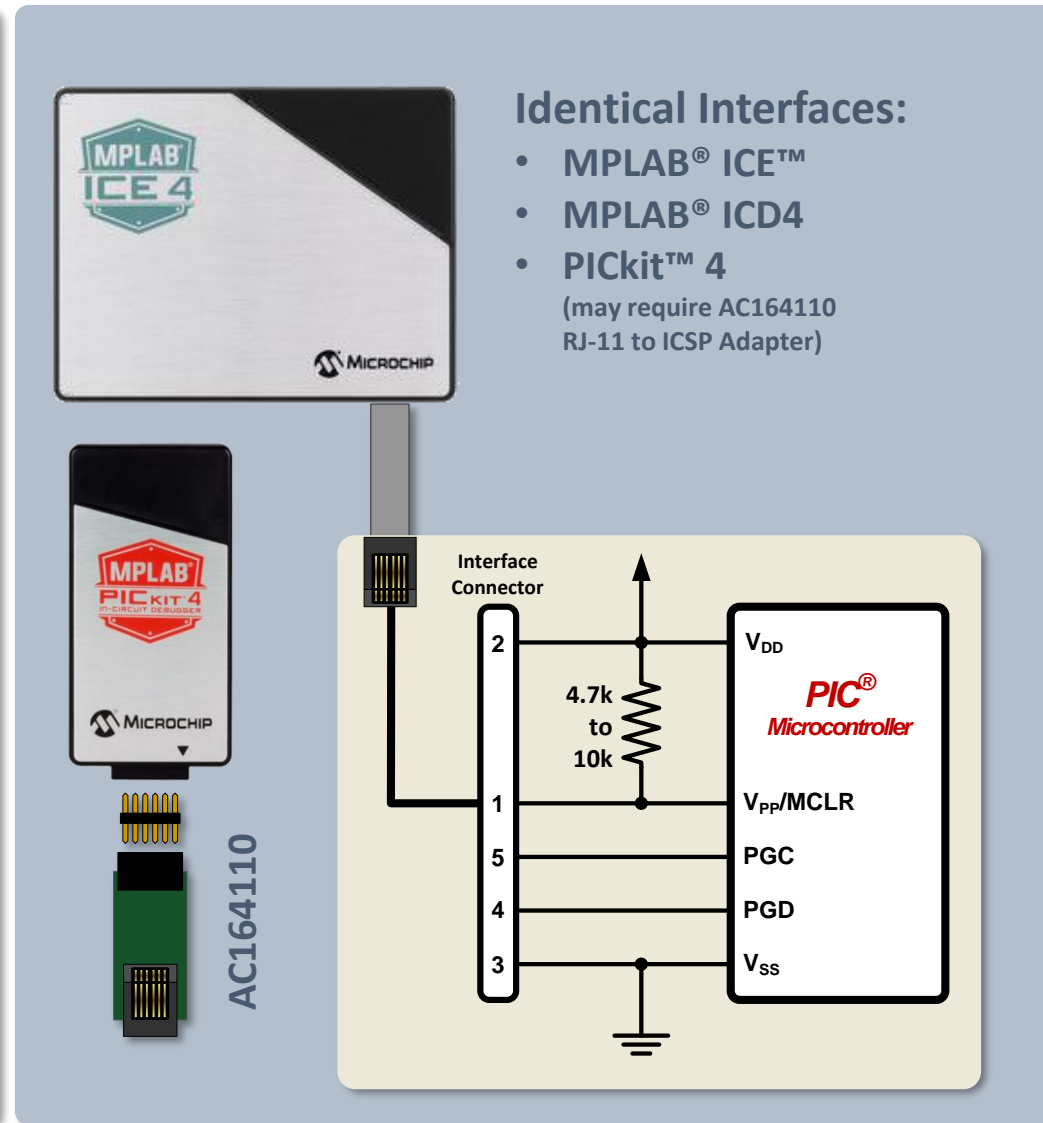
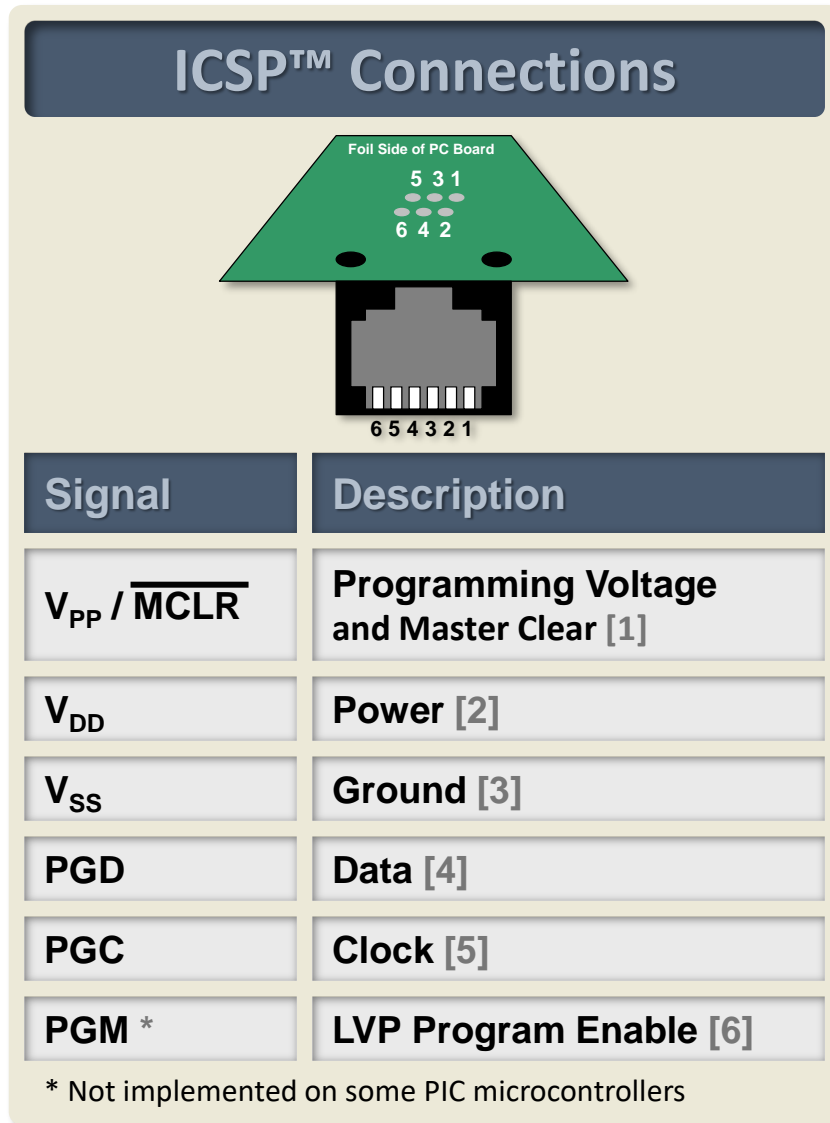
Debug Executive

Inserted to support In-Circuit Debugging



Debugger Connection to Target

6-pin RJ-11 or ICSP™ Header



Default Debugging Features

- **Target voltage of 1.20V to 5.5V**
 - Wide target voltage supports a variety of devices
- **Program voltages:**
 - 2.3V to 5.0V - Low voltage and high voltage program entry modes
 - 1.2V to 2.3V - Low voltage program entry mode only
- **Portable USB-powered and RoHS-compliant**
 - Powered by a high-speed USB 2.0, no external power required
 - CE and RoHS-compliant
- **8-pin single in-line header**
 - Supports advanced interfaces such as 4-wire JTAG and Serial Wire Debug with streaming Data Gateway
- **Compatibility**
 - Backward compatible for demo boards, headers and target systems using 2-wire JTAG and ICSP
- **Field-upgradeable through an MPLAB X IDE firmware download**
 - Add new device support and features by installing the latest version of MPLAB X IDE, which is available as a free download at www.microchip.com/mplabx
- **Debugging Features**
 - Hard- and software break points
 - Instruction Stopwatch

MPLAB® PICkit™ 4 In-Circuit Debugger



PG164140 – PICkit™ 4 Debug Express

- Matches silicon clocking speed
 - Programs as fast as the device will allow
- Can supply up to 50mA of power to the target
 - Can be powered from the target to program in the field
- Power from Target
 - Option to be self-powered from the target (2.7V to 5.5V)
- Programmer-to-Go (PTG) support
 - SD card slot to holds program data
 - Press on the logo to program the target
- Cost effective
 - Features and performance at a fraction of the cost of comparable debugger/programmers

MPLAB® ICD 4 In-Circuit Debugger

Microchip Standard Connectivity Plus JTAG



DV164045 – MPLAB® ICD 4

- **Full-Speed Real-Time Emulation**
 - Designed to support high-speed processors running at maximum speeds
 - High-Speed Programming
 - Quick firmware reload for fast debugging/in-circuit re-programming
 - Includes programmable adjustment of debugging speed for optimized programming
 - Test Interface Module
- **Ruggedized Interface**
 - Protection circuitries are added to the probe drivers to guard from power surges from the target
 - VDD and VPP voltage monitors protect against overvoltage conditions/all lines have over-current protection
 - Safely power up to 1A with an optional power supply
- **Compatibility**
 - Supports all MPLAB ICD 3 headers
- **Cost Effective**
 - Features and performance at a fraction of the cost of comparable emulator systems
- **Debugging Features**
 - Supports multiple breakpoints, stopwatch and source code file debugging
 - Selectable pull-up/pull-down option to the target interface in MPLAB X IDE's editor for quick program modification/debug

MPLAB® ICE 4 In-Circuit Debugger/Emulator

- **Variety of connectivity features to debug and program:**
 - SuperSpeed USB 3.0 host PC interface with USB speed of 5 Gbps
 - High-speed USB 2.0 host PC interface
 - Ethernet connectivity with speeds up to 100 Mbps
 - Wired/DHCP/APIAP IP addressing
 - Static IP addressing
 - Wi-Fi Access Point Connectivity (Wi-Fi-AP)
 - Connects with SSID of the unit
 - Connection to Wi-Fi using Wireless Station Mode (Wi-Fi-STA)
 - Connects to your home/office network
 - Uses network SSID, security type with username and password



DV244140 – MPLAB® ICE4

MPLAB® ICE 4 In-Circuit Debugger/Emulator



DV244140 – MPLAB® ICE4

- **Advanced Trace Capabilities**
 - Data capture/native trace
 - SPI trace (currently supported on 16-bit PIC devices)
 - Port trace (currently supported on 16-bit PIC devices)
 - PIC32 iFlowtrace™ 1.0/iFlowtrace 2.0
 - ARM ITM/SWO
- **Power Debugging Capabilities (Currently Only Supported on AVR and SAM MCUs)**
 - Can correlate to code by capturing power data and corresponding PC values
 - Can identify power profiles
 - Can determine functions which take the most power
 - Can Interface to MPLAB Data Visualizer
- **Power Monitoring (Supported on All Devices)**
 - Can monitor power of the full system or component
 - Contains two channels with different resolutions
 - CI/CD Support
- **MPLAB ICE 4 in-circuit emulator hardware tool can be used for continuous integration/continuous delivery over Ethernet using hardware in the loop**
- **Can use CI/CD wizard to set up for Jenkins and Docker on the latest version of MPLAB X IDE v6.00**
- **Data Gateway Interfaces**
- **USART**
- **Power**
- **SPI (support coming soon)**
- **I2C (support coming soon)**
- **Comprehensive Debug Functionality**
- **Supports multiple breakpoints, stopwatch and source code file debugging**
- **Selectable pull-up/pull-down option to the target interface**

MPLAB® ICE 4 In-Circuit Debugger/Emulator



DV244140 – MPLAB® ICE4

- **Advanced Trace Capabilities**
 - Data capture/native trace
 - SPI trace (currently supported on 16-bit PIC devices)
 - Port trace (currently supported on 16-bit PIC devices)
 - PIC32 iFlowtrace™ 1.0/iFlowtrace 2.0
 - ARM ITM/SWO
- **Power Debugging Capabilities (Currently Only Supported on AVR and SAM MCUs)**
 - Can correlate to code by capturing power data and corresponding PC values
 - Can identify power profiles
 - Can determine functions which take the most power
 - Can Interface to MPLAB Data Visualizer
- **Power Monitoring (Supported on All Devices)**
 - Can monitor power of the full system or component
 - Contains two channels with different resolutions

MPLAB® ICE 4 In-Circuit Debugger/Emulator

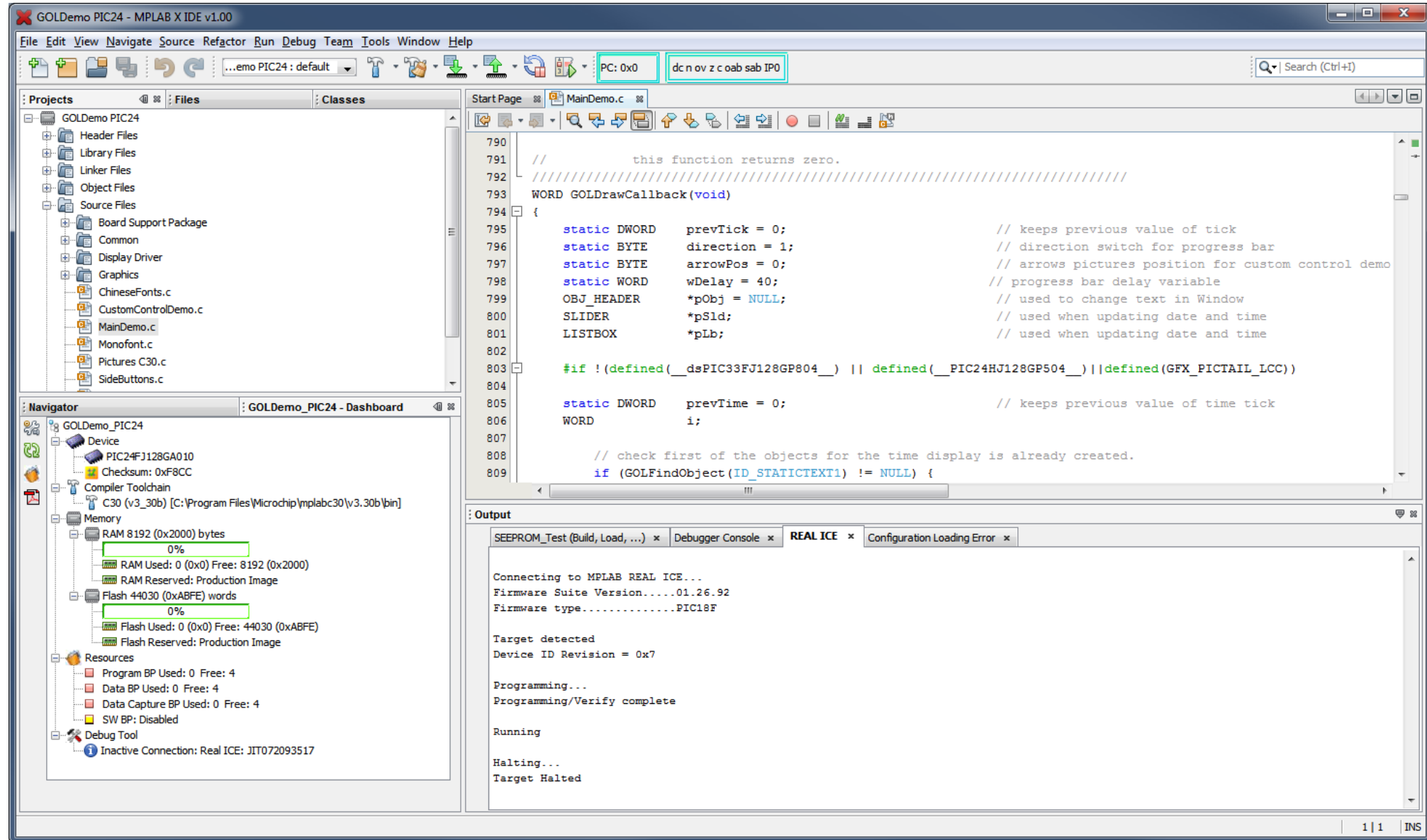


DV244140 – MPLAB® ICE4

- **CI/CD Support**
 - MPLAB ICE 4 in-circuit emulator hardware tool can be used for continuous integration/continuous delivery over Ethernet using hardware in the loop
 - Can use CI/CD wizard to set up for Jenkins and Docker on the latest version of MPLAB X IDE v6.00
- **Data Gateway Interfaces**
 - USART
 - Power
 - SPI (support coming soon)
 - I2C (support coming soon)
 - Comprehensive Debug Functionality
 - Supports multiple breakpoints, stopwatch and source code file debugging
 - Selectable pull-up/pull-down option to the target interface

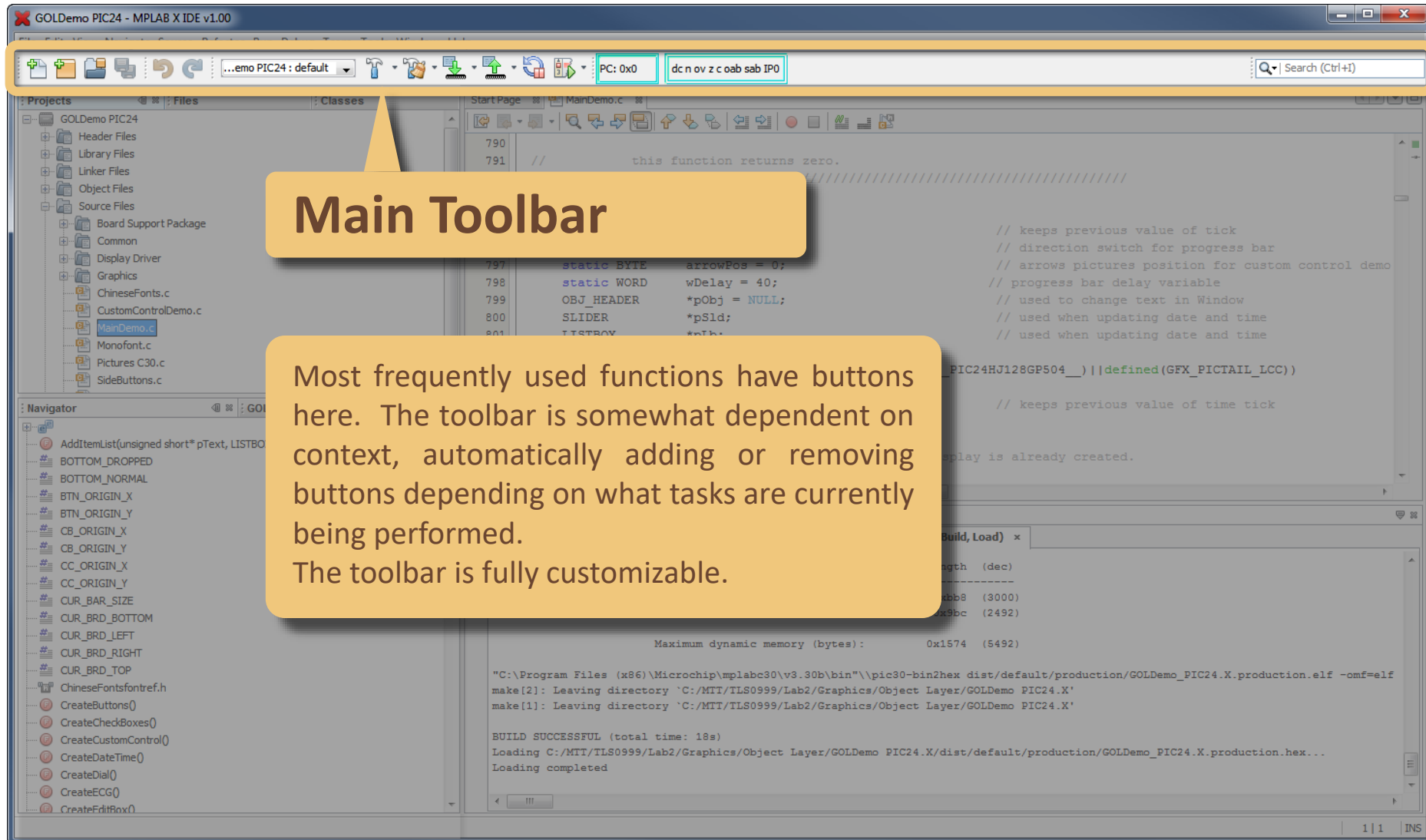
MPLAB® X IDE

IDE Layout



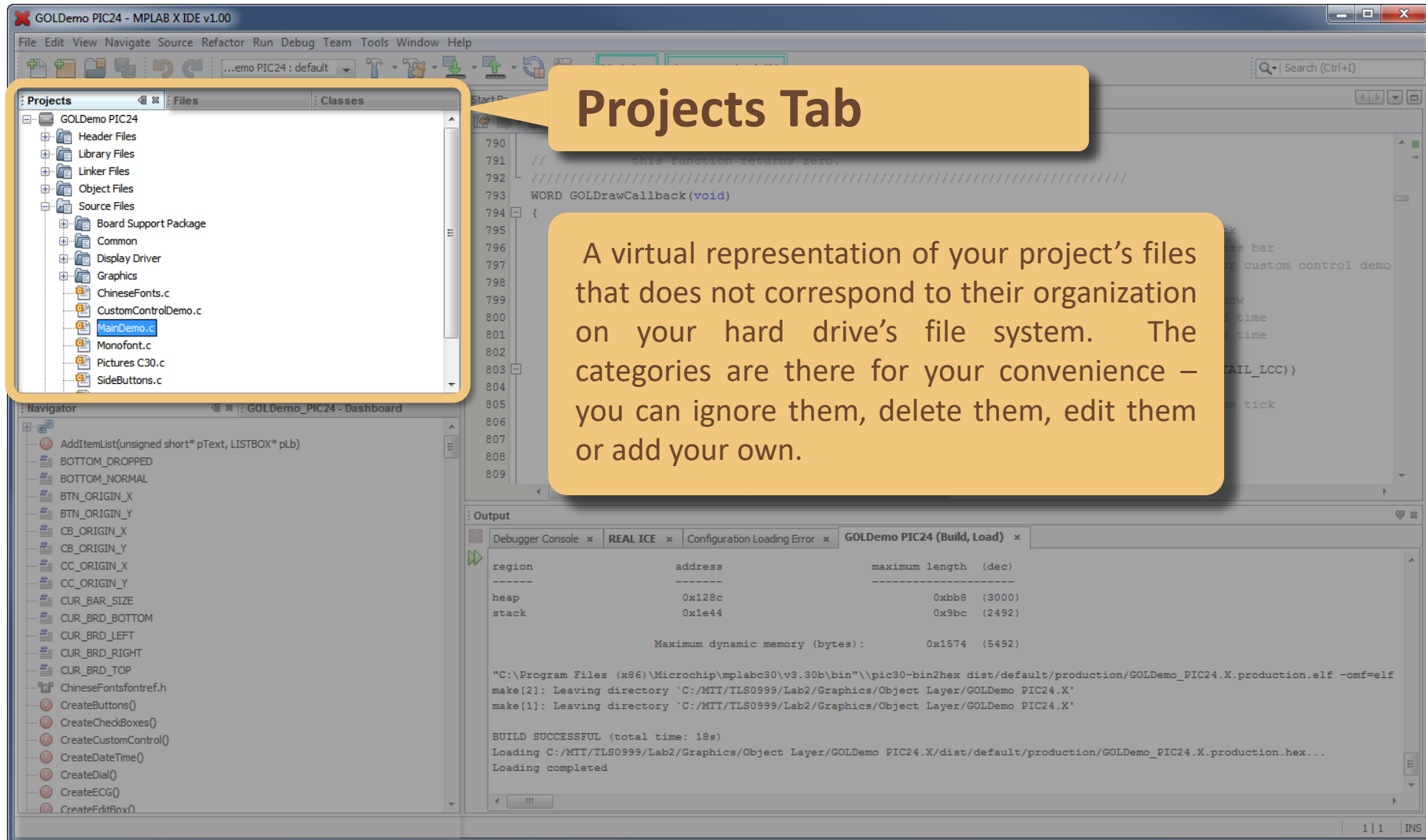
IDE Layout

Main Toolbar



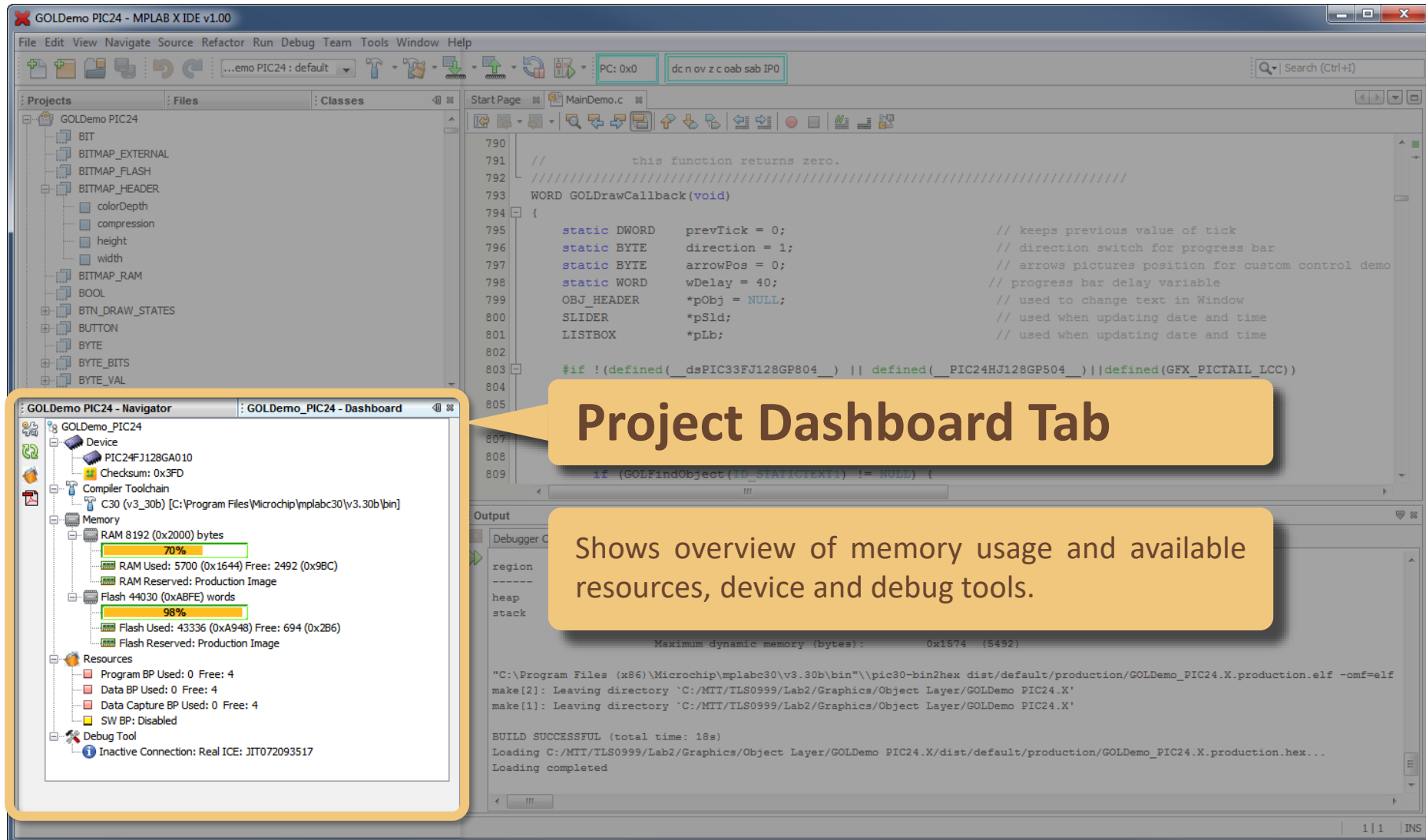
IDE Layout

Projects Tab (Project Tree)



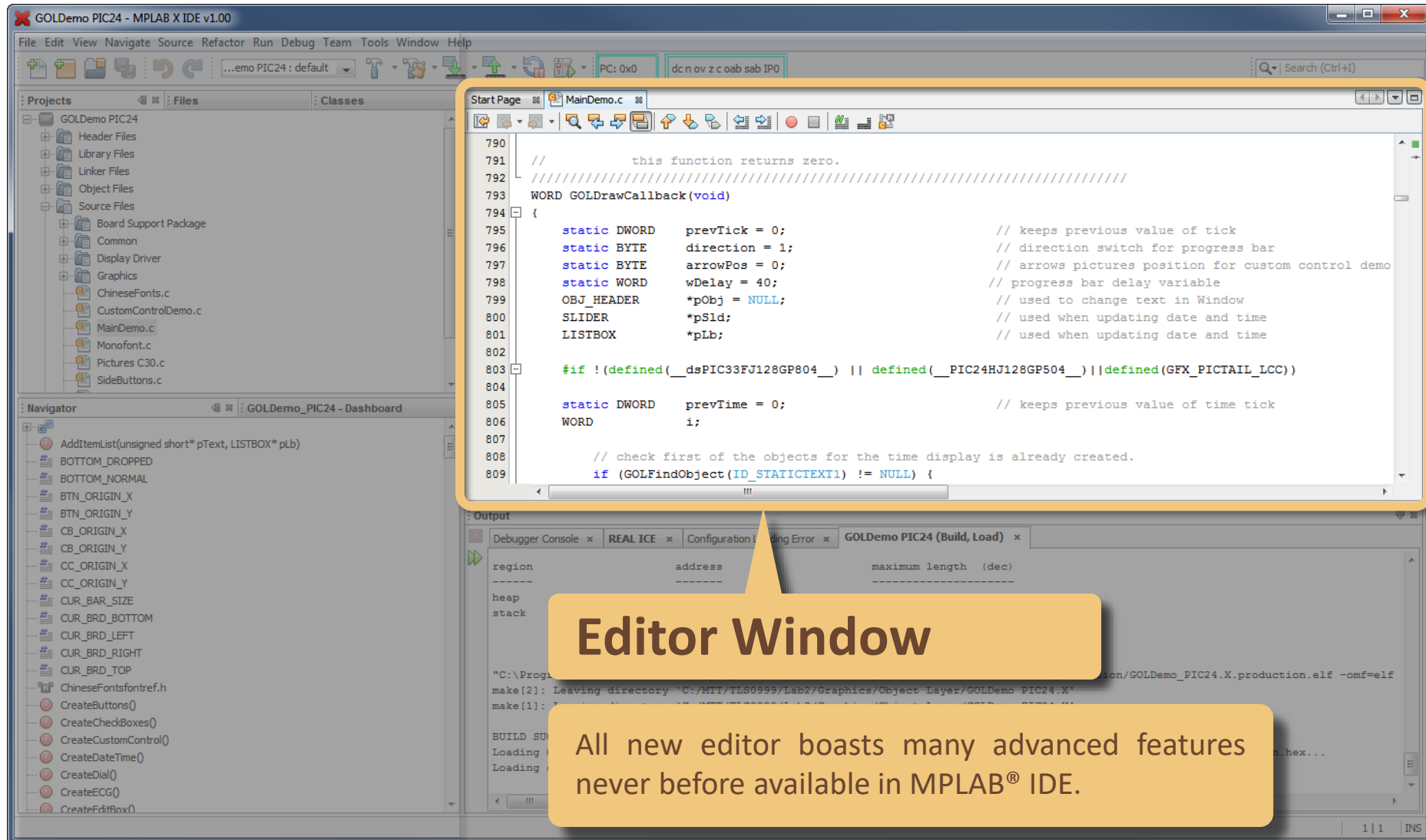
IDE Layout

Project Dashboard Tab



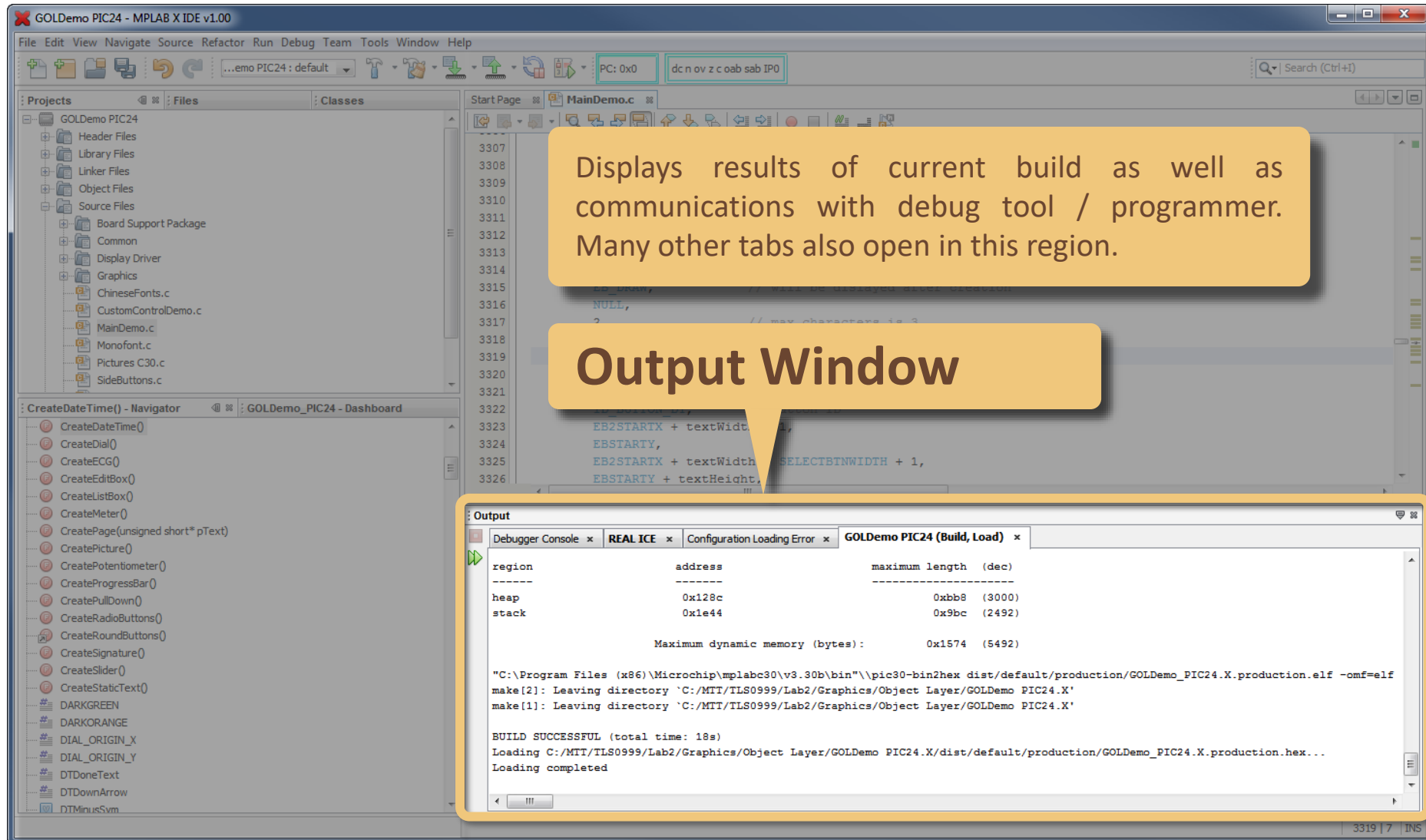
IDE Layout

Editor



IDE Layout

Output Window



MPLAB® Code Generators

Microchip Code Generator Standard Tools



8-/16-bit Software Framework

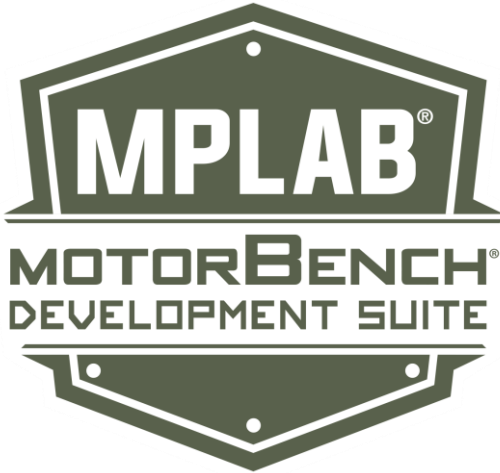


32-bit Software Framework



8-/16-/32-bit Device Configurator

Vertical Application Code Generator Tools



Motor Control Configuration



Standard Device Configuration



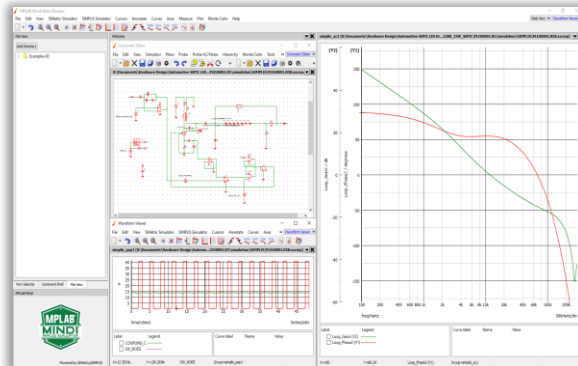
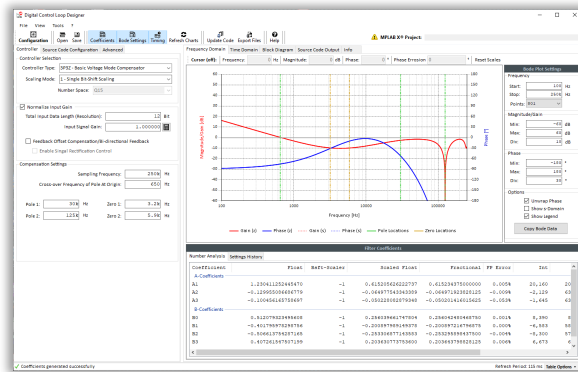
Power Supply Configuration
(independent)

MPLAB® PowerSmart™

MPLAB® PowerSmart™ Development Suite

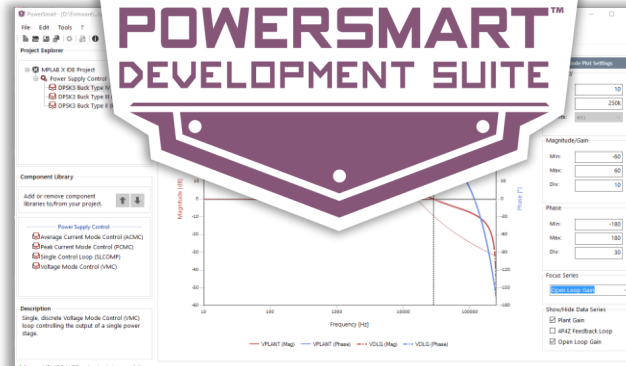
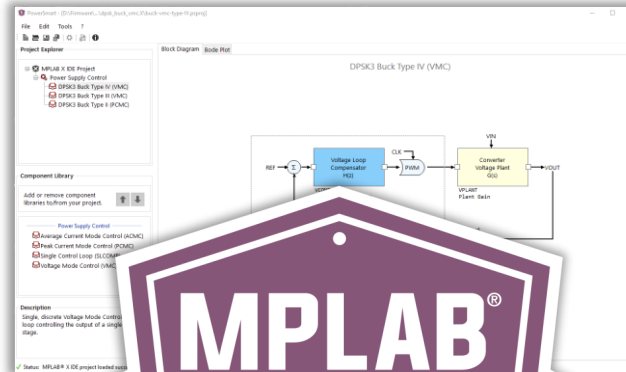
The comprehensive Digital Power Supply design eco-system

Circuit Simulation



Time/Frequency Domain
Analysis

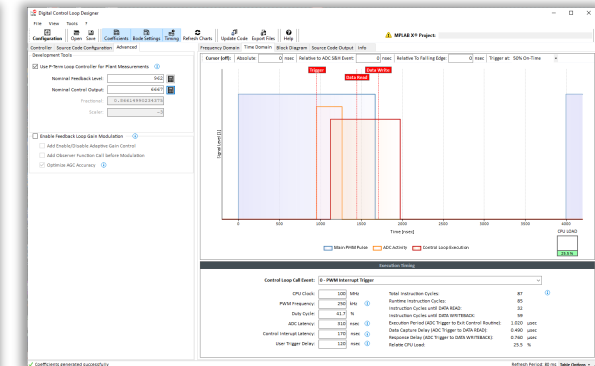
Control System Modeling



Model Validation

Code Generation

```
1 // Compiler generated assembly code
2 // Compiler generated assembly code
3 // Compiler generated assembly code
4 // Compiler generated assembly code
5 // Compiler generated assembly code
6 // Compiler generated assembly code
7 // Compiler generated assembly code
8 // Compiler generated assembly code
9 // Compiler generated assembly code
10 // Compiler generated assembly code
11 // Compiler generated assembly code
12 // Compiler generated assembly code
13 // Compiler generated assembly code
14 // Compiler generated assembly code
15 // Compiler generated assembly code
16 // Compiler generated assembly code
17 // Compiler generated assembly code
18 // Compiler generated assembly code
19 // Compiler generated assembly code
20 // Compiler generated assembly code
21 // Compiler generated assembly code
22 // Compiler generated assembly code
23 // Compiler generated assembly code
24 // Compiler generated assembly code
25 // Compiler generated assembly code
26 // Compiler generated assembly code
27 // Compiler generated assembly code
28 // Compiler generated assembly code
29 // Compiler generated assembly code
30 // Compiler generated assembly code
31 // Compiler generated assembly code
32 // Compiler generated assembly code
33 // Compiler generated assembly code
34 // Compiler generated assembly code
35 // Compiler generated assembly code
36 // Compiler generated assembly code
37 // Compiler generated assembly code
38 // Compiler generated assembly code
39 // Compiler generated assembly code
40 // Compiler generated assembly code
41 // Compiler generated assembly code
42 // Compiler generated assembly code
43 // Compiler generated assembly code
44 // Compiler generated assembly code
45 // Compiler generated assembly code
46 // Compiler generated assembly code
47 // Compiler generated assembly code
48 // Compiler generated assembly code
49 // Compiler generated assembly code
50 // Compiler generated assembly code
51 // Compiler generated assembly code
52 // Compiler generated assembly code
53 // Compiler generated assembly code
54 // Compiler generated assembly code
55 // Compiler generated assembly code
56 // Compiler generated assembly code
57 // Compiler generated assembly code
58 // Compiler generated assembly code
59 // Compiler generated assembly code
60 // Compiler generated assembly code
61 // Compiler generated assembly code
62 // Compiler generated assembly code
63 // Compiler generated assembly code
64 // Compiler generated assembly code
65 // Compiler generated assembly code
66 // Compiler generated assembly code
67 // Compiler generated assembly code
68 // Compiler generated assembly code
69 // Compiler generated assembly code
70 // Compiler generated assembly code
71 // Compiler generated assembly code
72 // Compiler generated assembly code
73 // Compiler generated assembly code
74 // Compiler generated assembly code
75 // Compiler generated assembly code
76 // Compiler generated assembly code
77 // Compiler generated assembly code
78 // Compiler generated assembly code
79 // Compiler generated assembly code
80 // Compiler generated assembly code
81 // Compiler generated assembly code
82 // Compiler generated assembly code
83 // Compiler generated assembly code
84 // Compiler generated assembly code
85 // Compiler generated assembly code
86 // Compiler generated assembly code
87 // Compiler generated assembly code
88 // Compiler generated assembly code
89 // Compiler generated assembly code
90 // Compiler generated assembly code
91 // Compiler generated assembly code
92 // Compiler generated assembly code
93 // Compiler generated assembly code
94 // Compiler generated assembly code
95 // Compiler generated assembly code
96 // Compiler generated assembly code
97 // Compiler generated assembly code
98 // Compiler generated assembly code
99 // Compiler generated assembly code
100 // Compiler generated assembly code
```



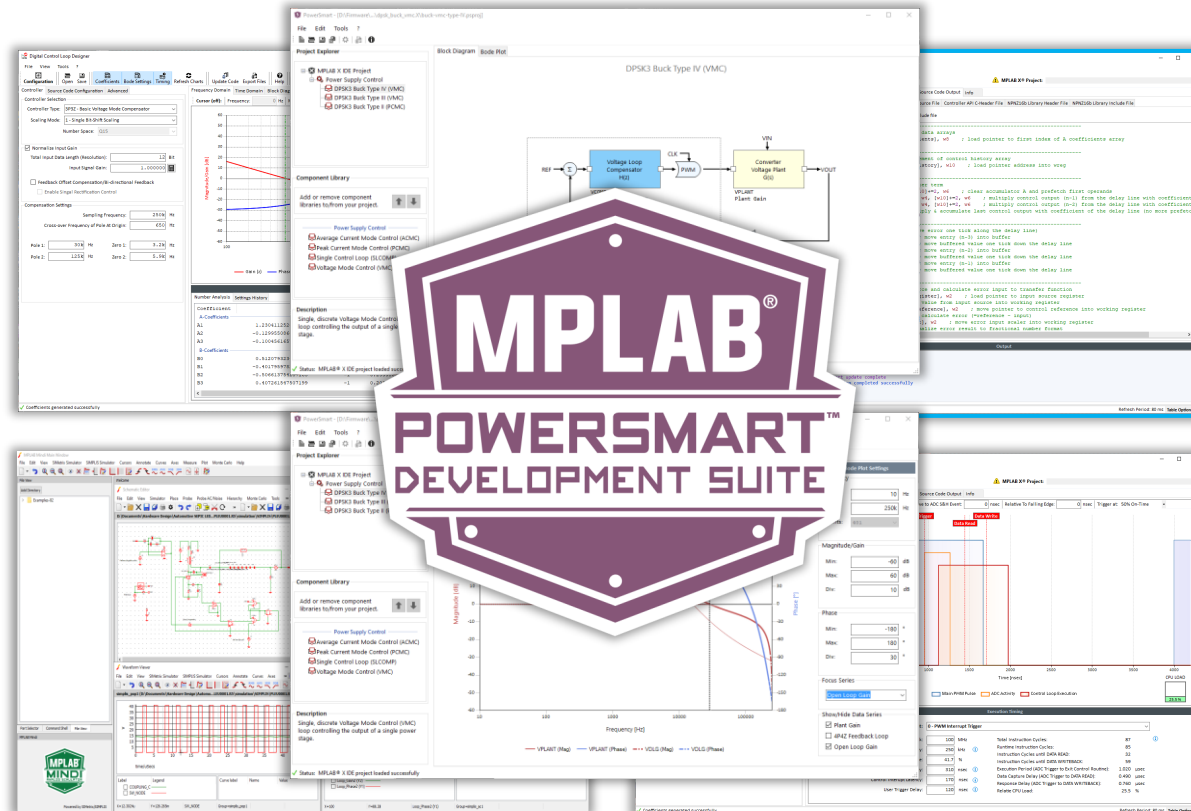
Implementation Analysis

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

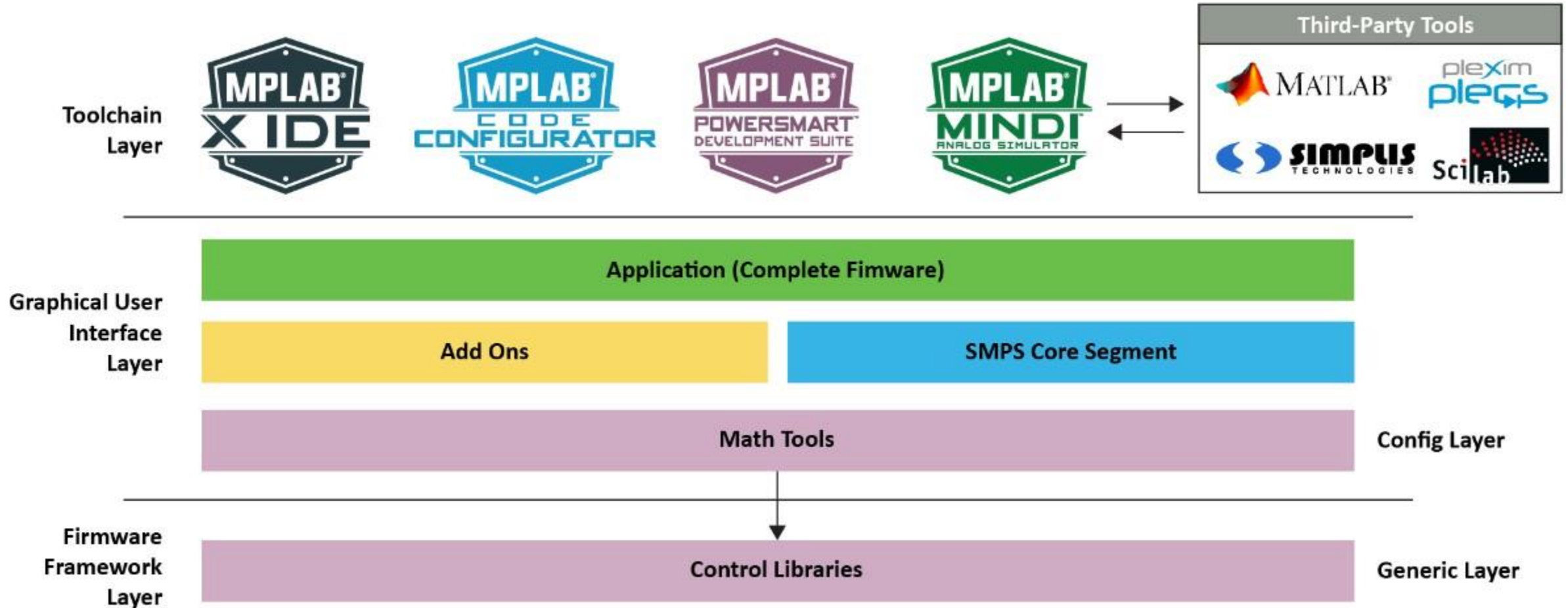
MCU16 Digital Power Applications

MPLAB® PowerSmart™ Development Suite

Junction between Circuit Design & Simulation and X IDE



PowerSmart™ Eco System



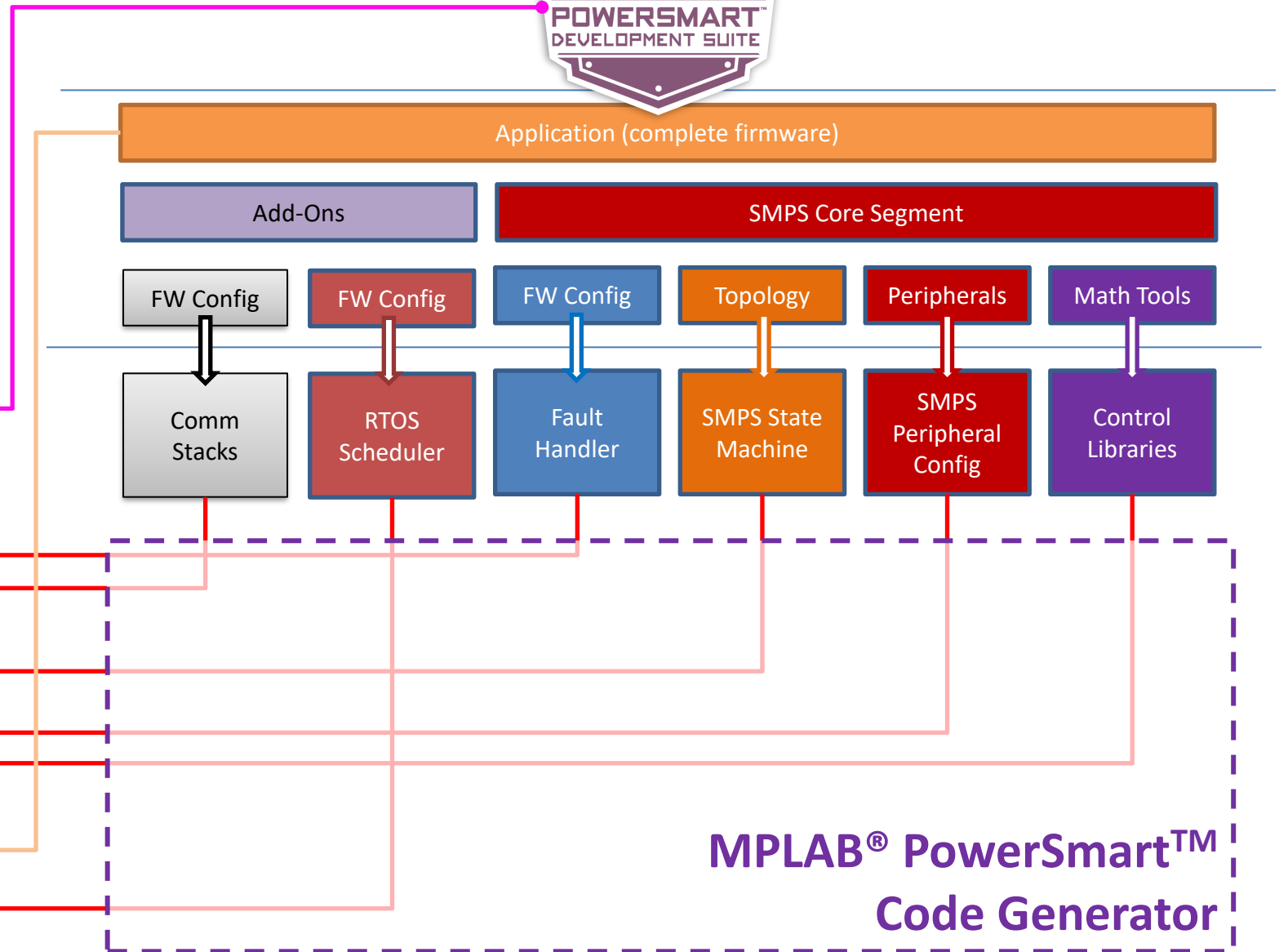
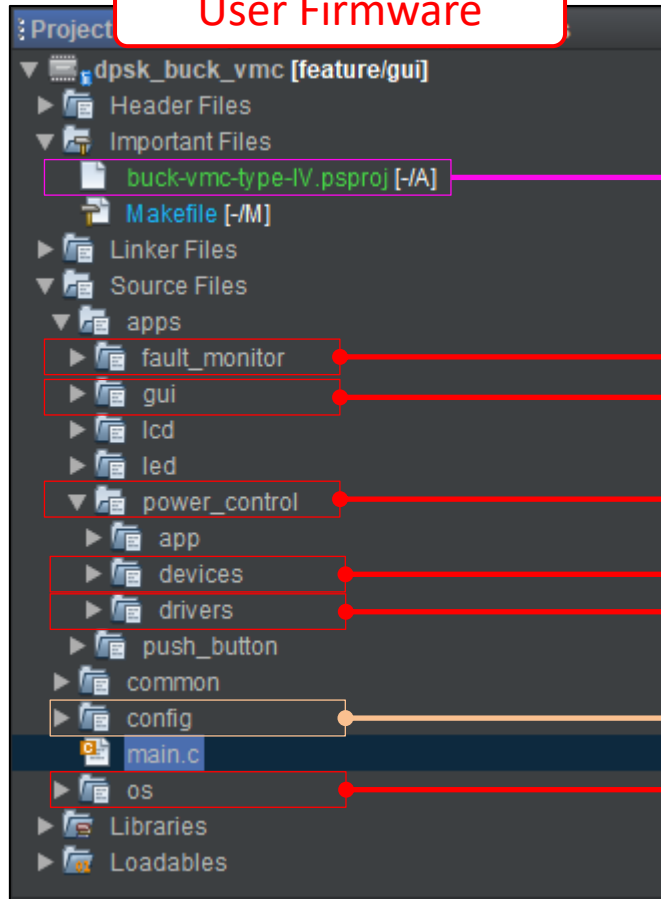
<https://www.microchip.com/powersmart>

MCU16 Digital Power Applications

PowerSmart™ Eco System

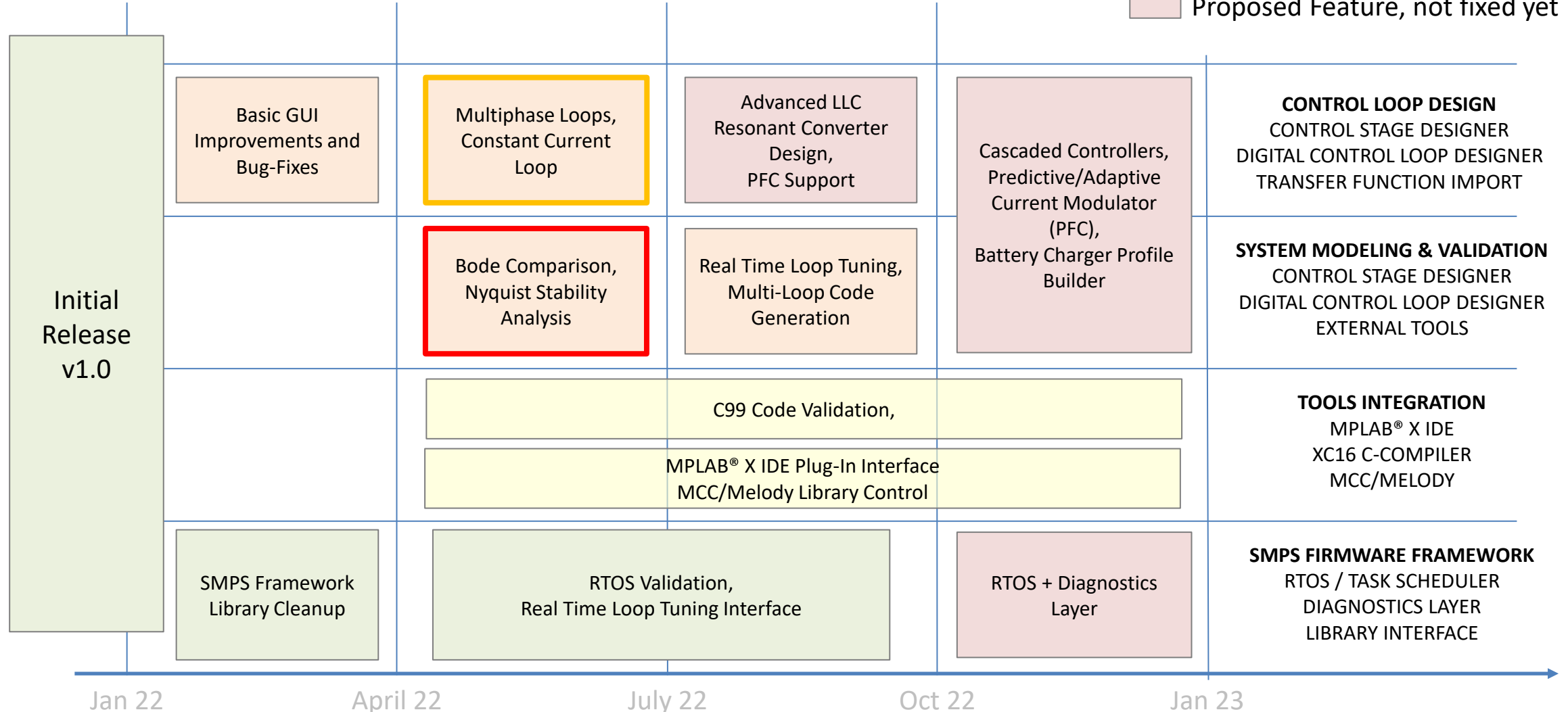


User Firmware



Feature Roadmap

- Scheduled Release Feature
- Support Feature
- Building Block Development
- Proposed Feature, not fixed yet





MPLAB® PowerSmart™ v1.0 Digital Control Library Designer

MPLAB® PowerSmart™ Development Suite

Digital Control Library Designer

- **Key Features**

- Supports z-Domain Compensation Filters from 1st to 6th Order
- Fixed-Point and Floating-Point DSP Library Support
- Graphic Loop Adjustment
- Transfer Function Import/Export
- Built-In Number Resolution Analysis and Optimization
- Graphic Execution Timing Analysis
- ANSI C/DSP Assembly Code Generation

- **Special Features**

- **Advanced Control Options**

- PS-DCLD provides code generator options injecting code into the real-time high-speed loop allowing advanced control algorithms manipulative access to the compensation filter computation as well as data provider sources to track and monitor internal processing data at runtime.

- **System Design Options**

- PS-DCLD offers alternative feedback loops enabling power supply plant measurements supporting power plant model verification and/or directly deriving essential, unknown plant transfer function information through bench tests using vector network analyzers.

- **MPLAB® X Support**

- PS-DCLD was developed as control library generator for Microchip dsPIC33 product families. To allow the code generator derive project settings like C include directories and selected device part number, each controller project is tightly coupled to a user-specified MPLAB® X project. For most convenient use, PS-DCLD can be opened from the MPLAB® X project manager context menu when the project file is included in the related MPLAB® X project.

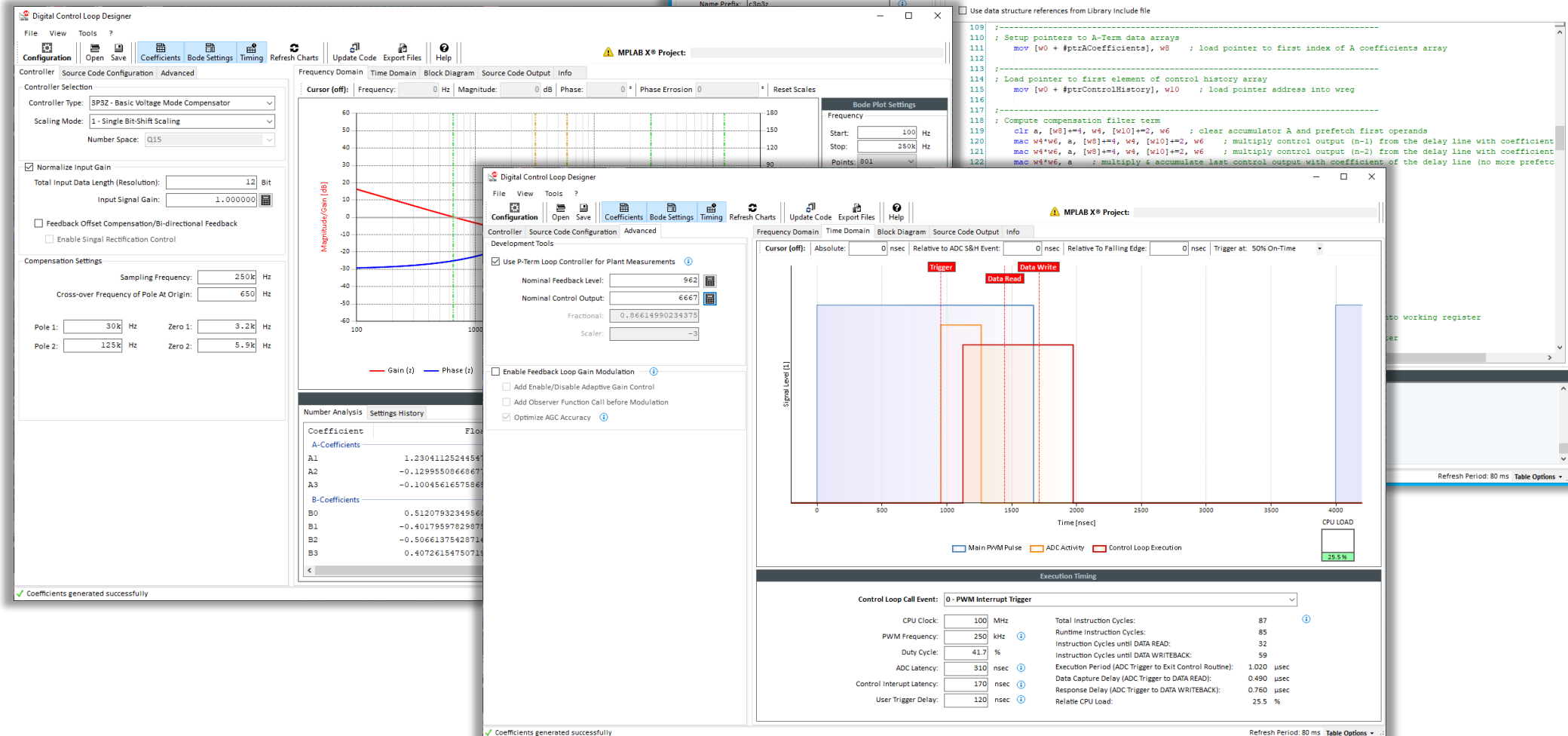
- **Data Export**

- Export of s-Domain and z-Domain Transfer Function (Bode Plot Data) copies the bode plot data table into the clipboard as tab-separated text table with column headers. This data can directly be pasted into external applications such as MS Excel.

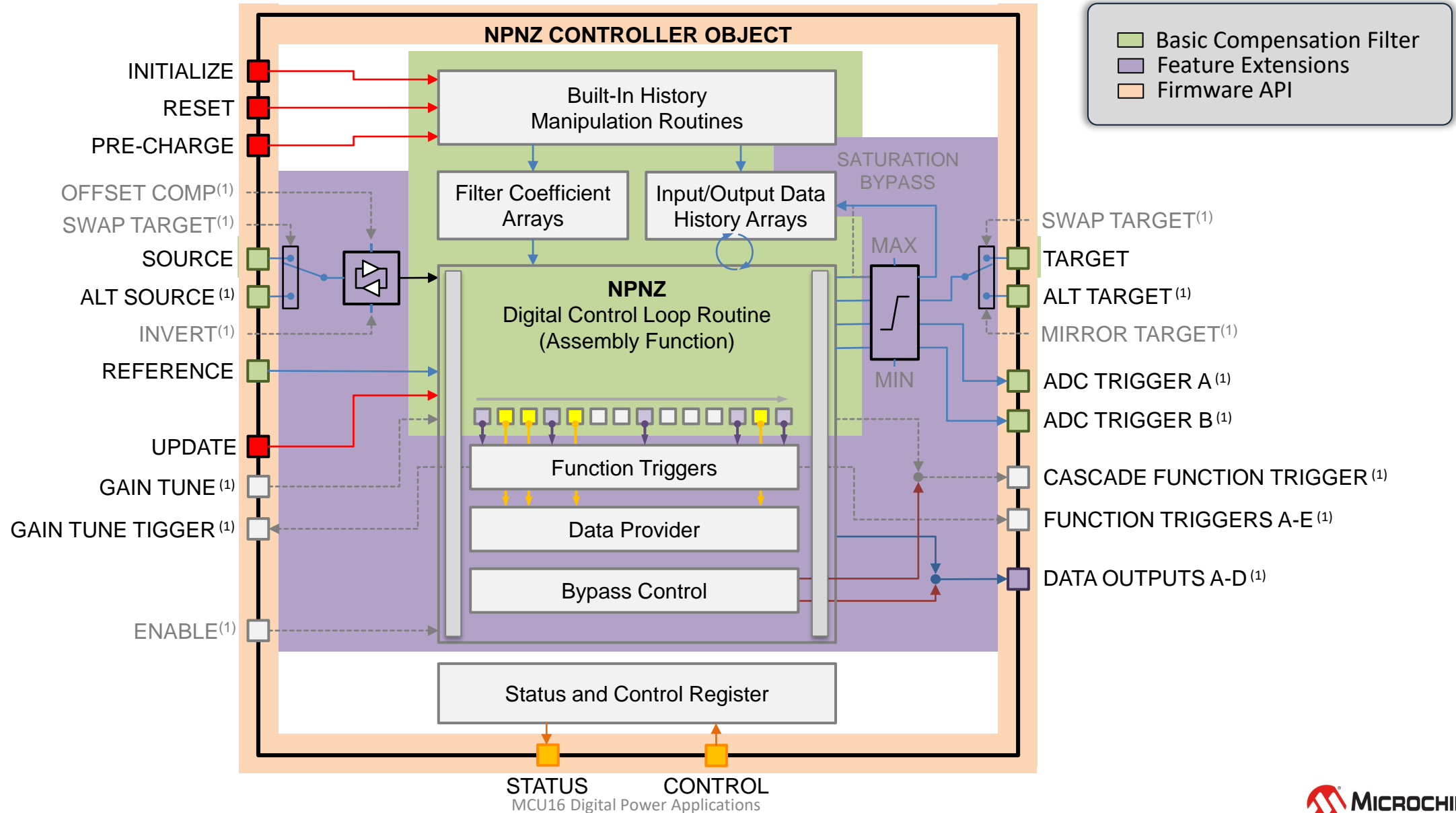
MPLAB® PowerSmart™ Digital Control Library Designer (DCLD)

Code Generation

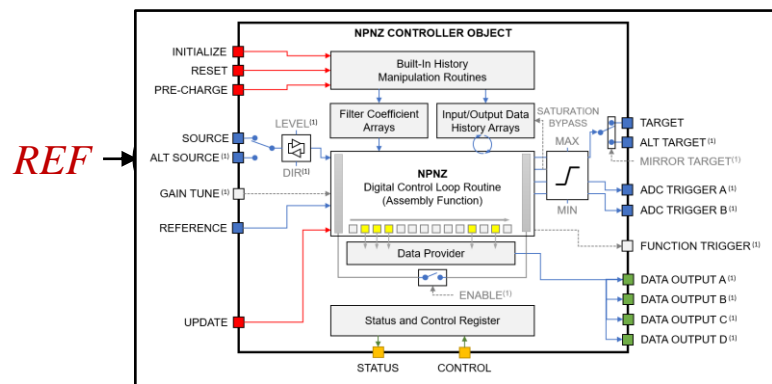
Frequency Domain Design



NPNZ Controller Block Diagram



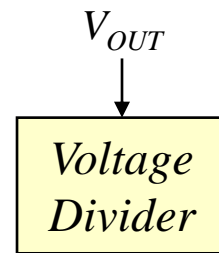
Using DCLD in Multi-Loop Systems



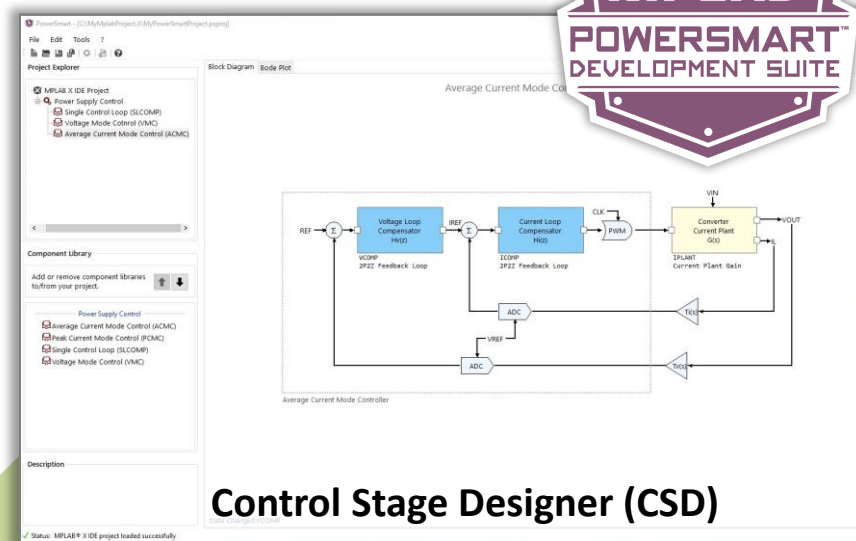
input

output

Voltage Loop



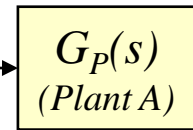
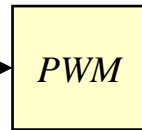
ADC



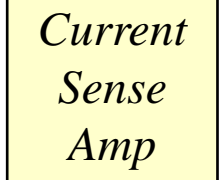
Control Stage Designer (CSD)

REF

output



I_{OUT}

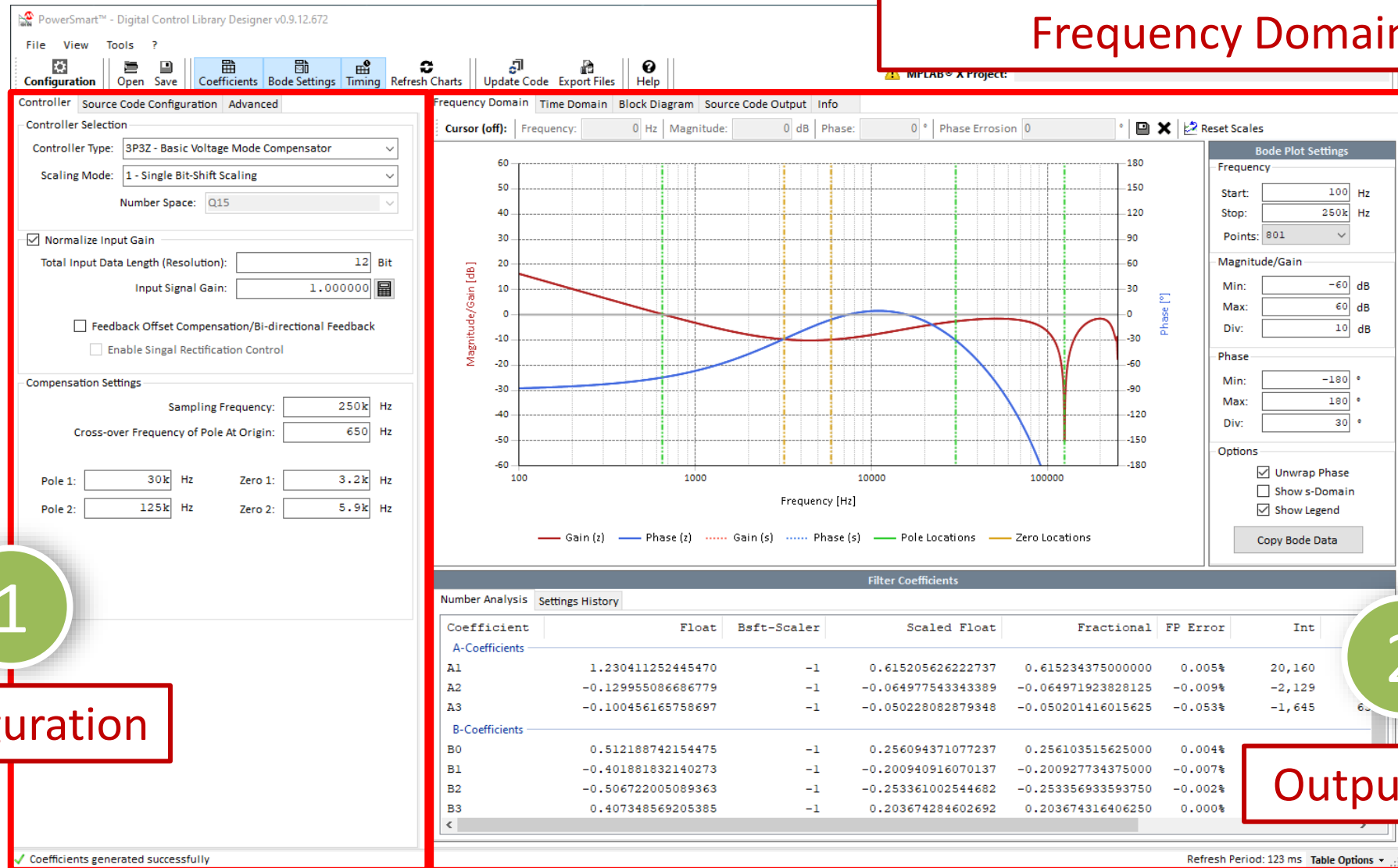


Current Loop

input

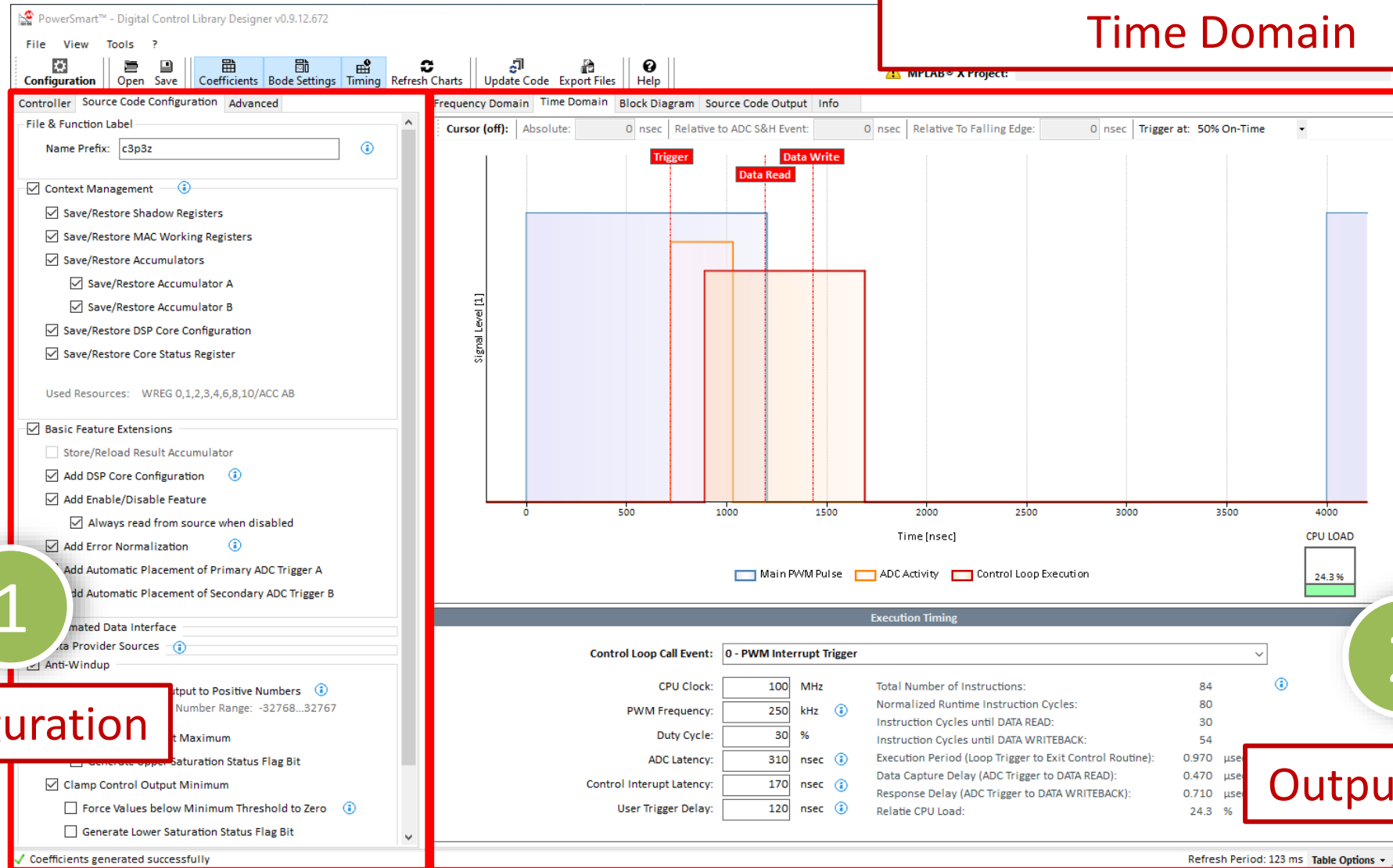
ADC

PowerSmart™ DCLD Main Window Overview



PowerSmart™ DCLD Main Window Overview

Time Domain

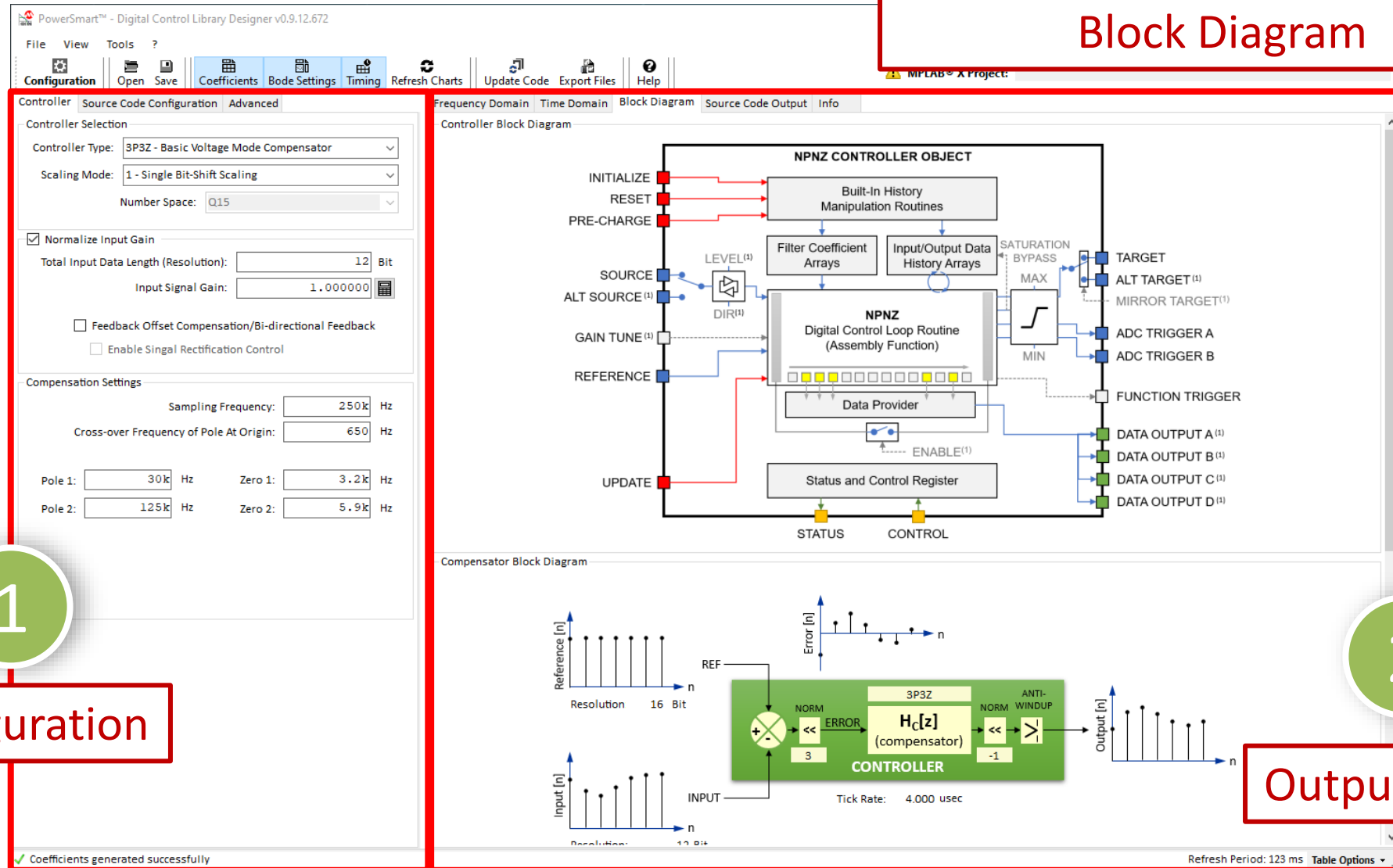


Configuration

Output/Analysis

PowerSmart™ DCLD Main Window Overview

Block Diagram



1

Configuration

2

Output/Analysis

PowerSmart™ DCLD Main Window Overview

Code Generation

PowerSmart™ - Digital Control Library Designer v0.9.12.672

File View Tools ?

Configuration Open Save Coefficients Bode Settings Timing Refresh Charts Update Code Export Files Help

Controller Source Code Configuration Advanced

File & Function Label

Name Prefix: c3p3z

☒ 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

Used Resources: WREG 0,1,2,3,4,6,8,10/ACC AB

☒ 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

Output to Positive Numbers

Number Range: -32768...32767

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

Frequency Domain Time Domain Block Diagram Source Code Output Info

Assembly Source File API C-Source File API C-Header File NPNZ16b Library Header File NPNZ16b Library Include File Example Code

☐ Use data structure references from Library Include file

☐ Add file location in #include path

```
135
136
137 ; Configure DSP for fractional operation with normal saturation (Q1.31 format)
138 mov #0x00C0, w4 ; load default value of DSP core configuration enabling accumulator saturation and sign
139 mov w4, _CORCON ; load default configuration into CORCON register
140
141
142 ; Setup pointers to A-Term data arrays
143 mov [w0 + #ptrACoefficients], w8 ; load pointer to first index of A coefficients array
144
145
146 ; Load pointer to first element of control history array
147 mov [w0 + #ptrControlHistory], w10 ; load pointer address into working register
148
149
150 ; Compute compensation filter term
151 clr a, [w8]+4, w4, [w10]+2, w6 ; clear accumulator A and prefetch first operands
152 mac w4*w6, a, [w8]+4, w4, [w10]+2, w6 ; multiply control output (n-1) from the delay line with coefficient A1
153 mac w4*w6, a, [w8]+4, w4, [w10]+2, w6 ; multiply control output (n-2) from the delay line with coefficient A2
154 mac w4*w6, a ; multiply & accumulate last control output with coefficient of the delay line (no more
155
156
157 ; Update error history (move error one tick down the delay line)
158 mov [w10 + #4], w6 ; move entry (n-3) into buffer
159 mov w6, [w10 + #6] ; move buffered value one tick down the delay line
160 mov [w10 + #2], w6 ; move entry (n-2) into buffer
161 mov w6, [w10 + #4] ; move buffered value one tick down the delay line
162 mov [w10 + #0], w6 ; move entry (n-1) into buffer
163 mov w6, [w10 + #2] ; move buffered value one tick down the delay line
164
```

Output

>>DATA READ delay: 300 ns

>>WRITEBACK delay: 540 ns

>>timing chart update completed successfully (22 ms)

>>code generation completed successfully (251 ms)

>>update Bode plot data...

>>Bode plot data update complete (4 ms)

Refresh Period: 123 ms Table Options

Configuration

Output/Analysis

Thank you!

May the power be with you!