



PSFB Converter Code Examples

dsPIC33CK64MP102



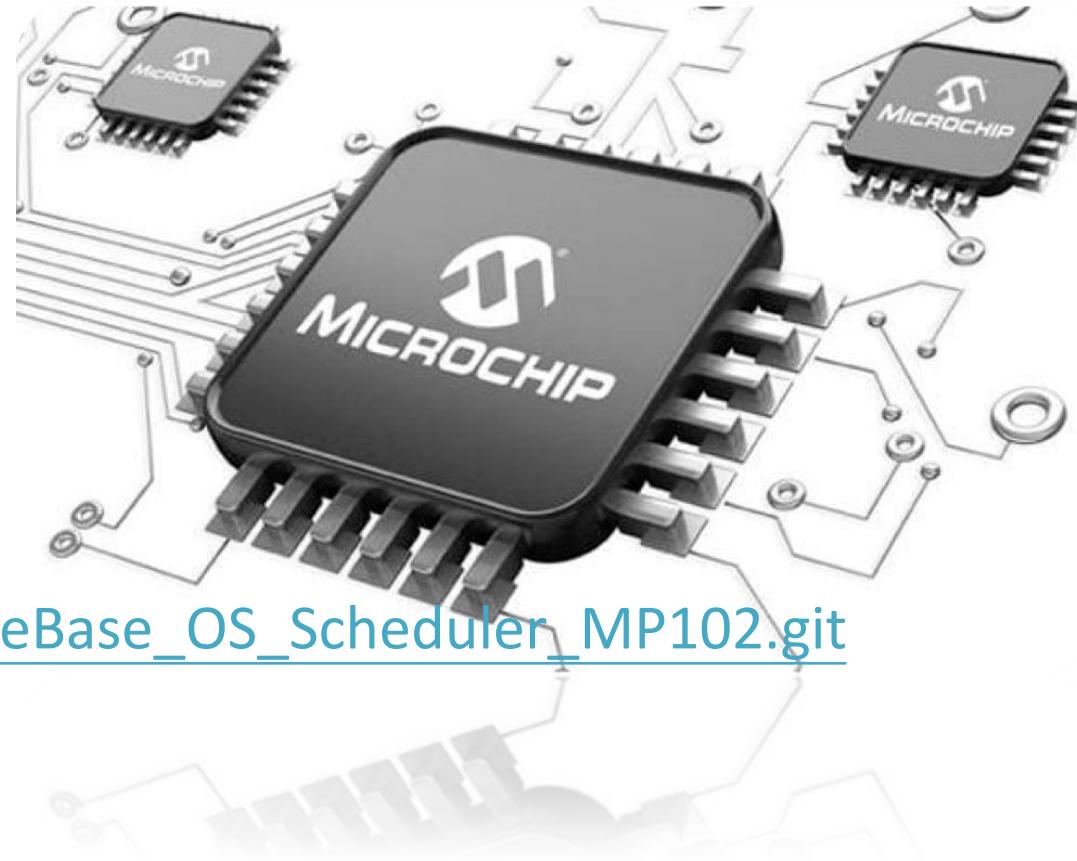
A Leading Provider of Smart, Connected and Secure Embedded Solutions



Microchip
ESE / Edward Lee
March-20, 2024

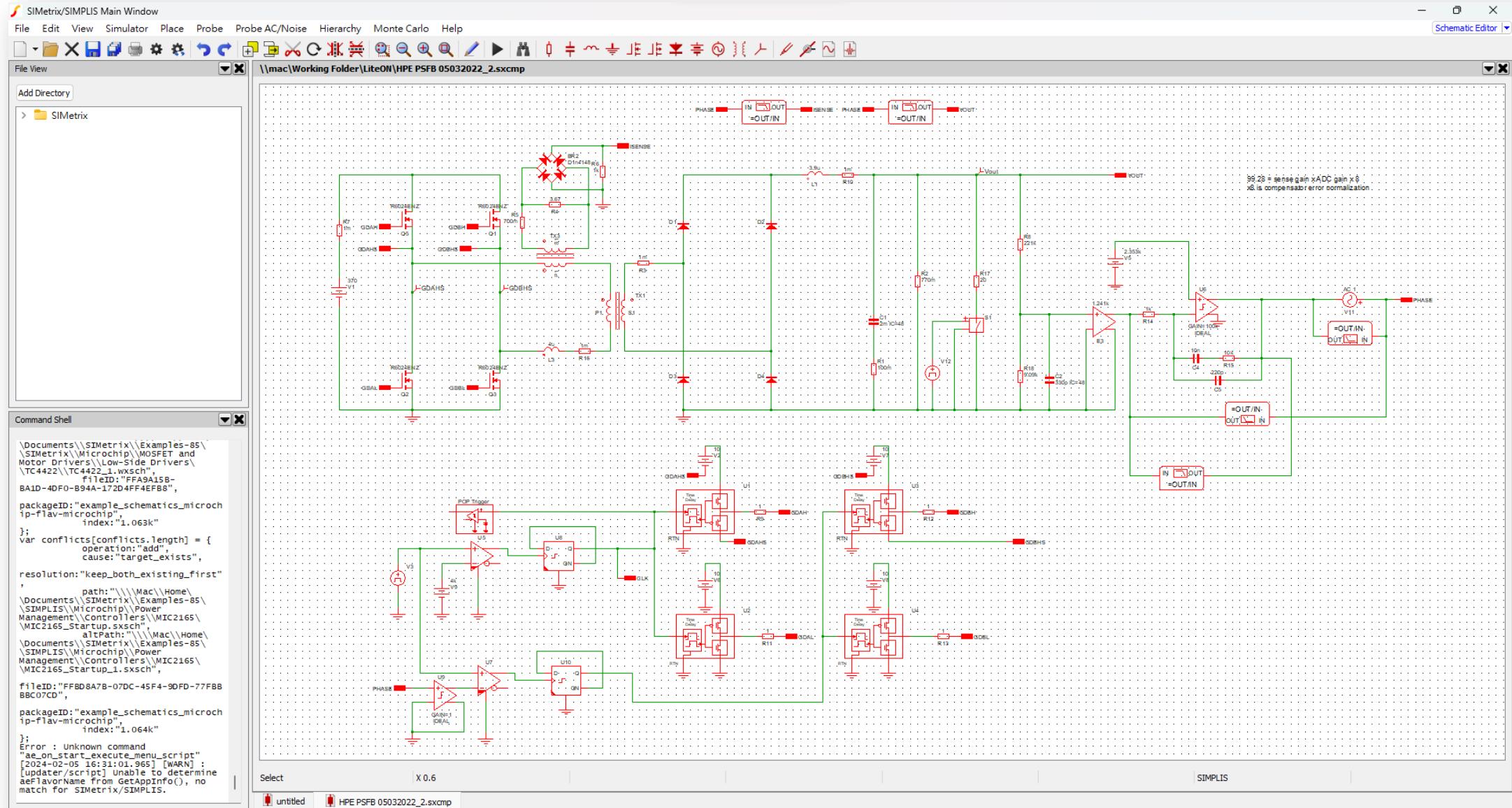
Agenda

- Simulation
- Test Environment
- Code Examples
 - Codebase with OS Scheduler
 - https://github.com/EdwardLeeTW/PSFB_CodeBase_OS_Scheduler_MP102.git
 - VMC PSFB with SR
 - PCMC PSFB with SR

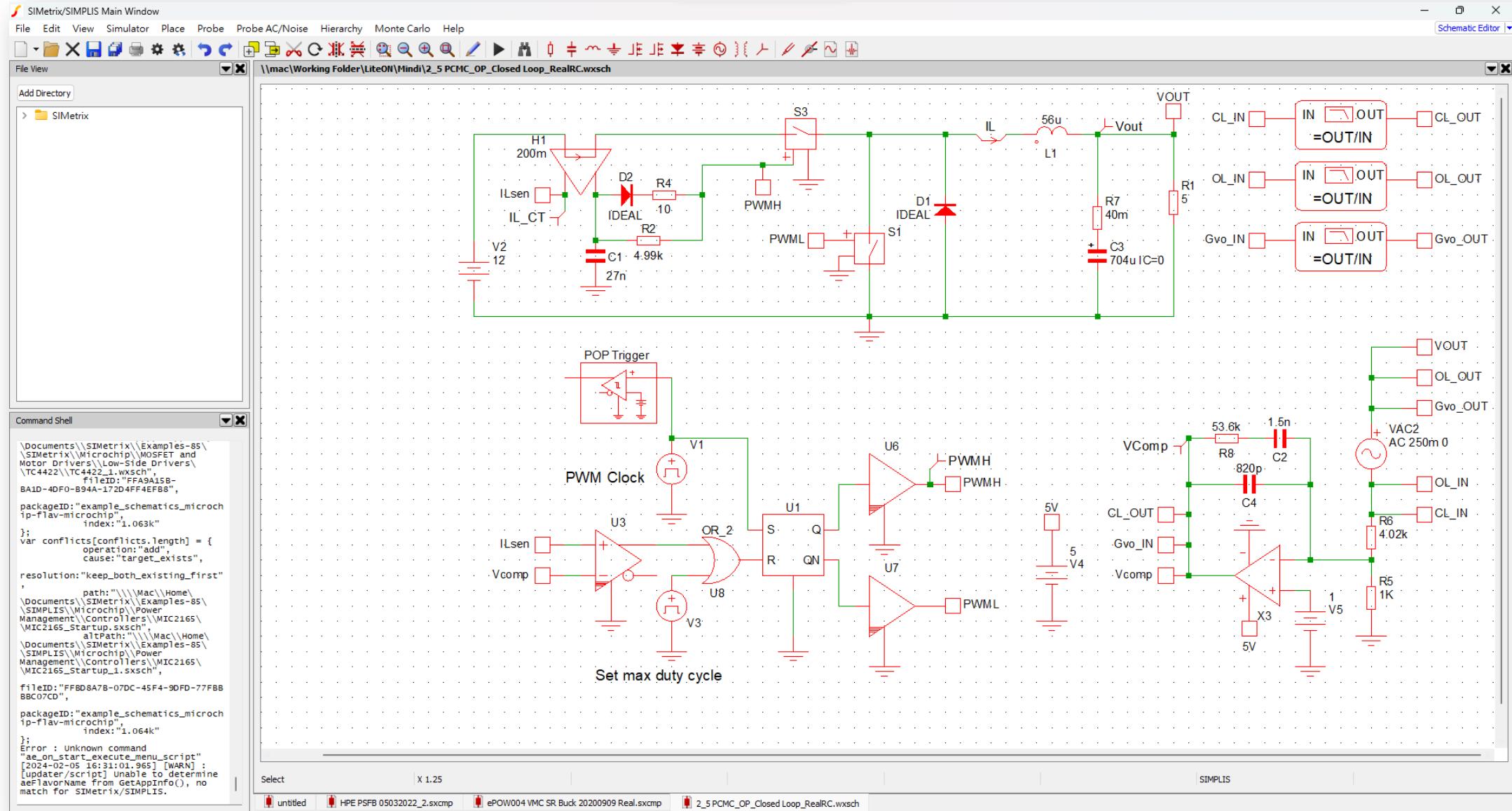


Simulation

SIMPLIS



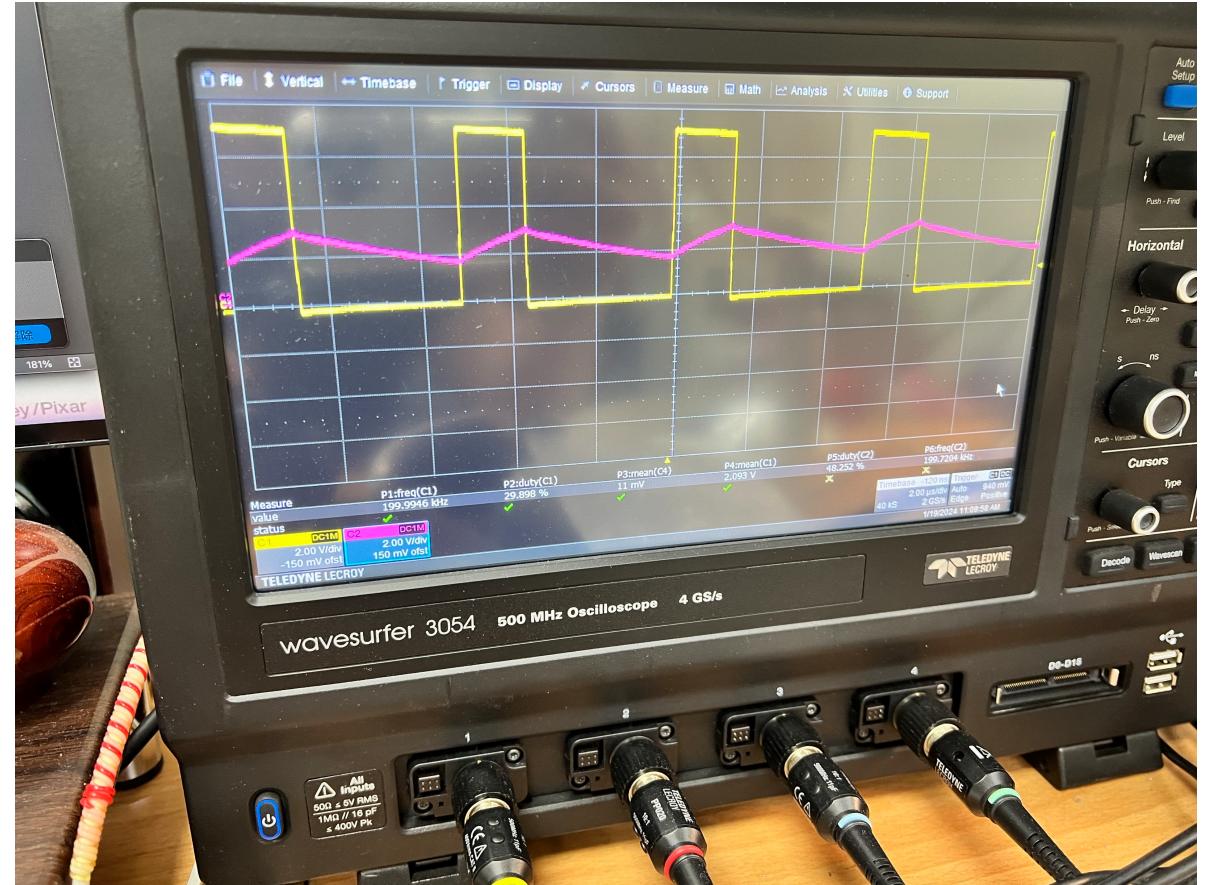
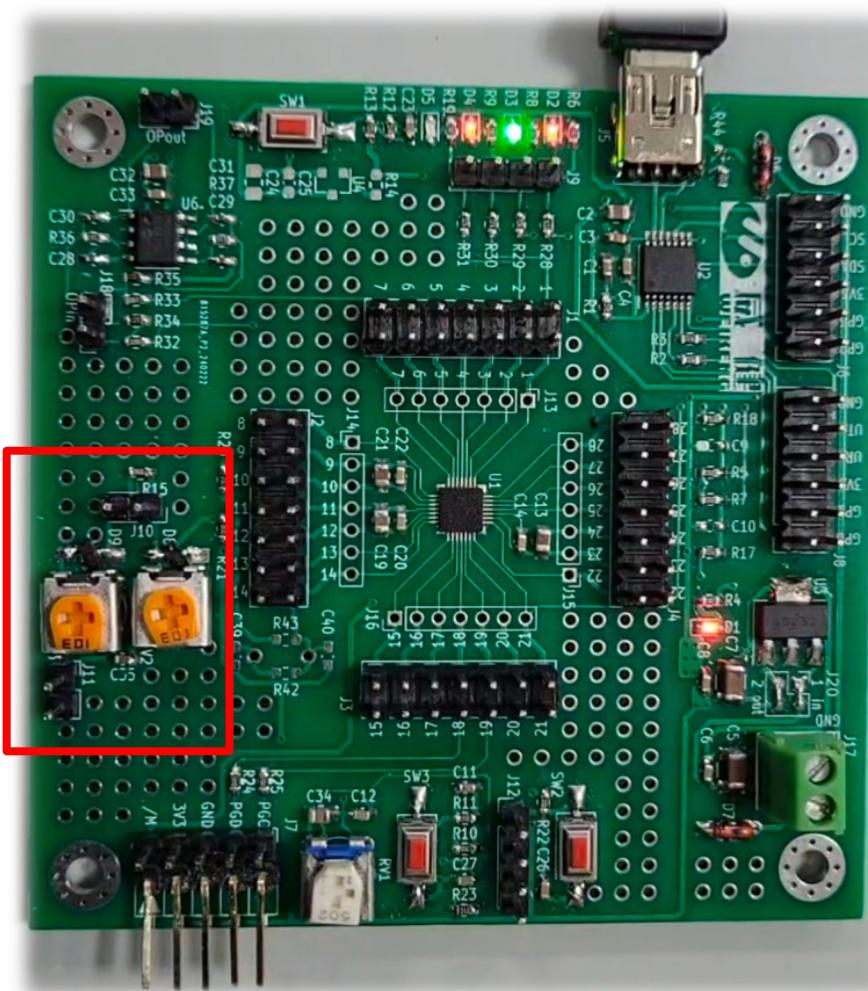
SIMPLIS



Test Environment

MP102 Debugging Board by Weikeng

- Peak Current Signal Simulation



Basic F/W Framework with OS Scheduler

Peripheral & Tool Descriptions

- Peripherals (in RED)
 - Timer1 100us
 - IP = 3 & NO Context
 - LED1 toggling in Tasks_1s
 - I/O
 - LED1 – RB4
- Chip: dsPIC33CK64MP102
- S/W Versions
 - MPLAB X IDE v6.20
 - DFP v1.13.366
 - Compiler XC16 v2.1
 - MCC v5.5.0
 - Core v5.7.0
 - MCU Lib v1.171.4

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

PLL Enable

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

VCO & AVCO

VCO Divider: FVCO/4
AVCO Divider: FVCO/2

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

Reference Oscillator Output

CAN FD Clock Generator

Device Resources Content Manager

Documents dsPIC33CK256MP508 Product Page

Libraries

- 16-bit Bootloader
- CHARACTER LCD
- CryptoAuthLibrary
- DacLibrary
- FatFs
- Foundation Services
- LED
- LED_BLUE
- LED_GREEN
- LED_RED
- MCP802X
- motorBench® Development Suite
- Pac193xLibrary
- POTENTIOMETER
- RGB LED

Pin Manager: Grid View Notifications [MCC]

Pin Manager: Package View

RB9|PGC1 VSS4A RB10 RB11 RB12 RB13

RB14 1 RB15 2 NMCLR 3 RA0 4 RA1 5 RA2 6 RA3 7

28 27 26 25 24 23 22

21 RB8|PGD1 20 RB7 19 RB6 18 RB5 17 RB4|LED1|GPIO 16 RB3 15 RB2

dsPIC33CK64MP102

Microchip Logo

FIGURE 9-3: PLL AND VCO DETAIL

Note 1: Clock option for PWM.
 Note 2: Clock option for ADC.
 Note 3: Clock option for DAC.
 Note 4: PLL source is always FRC unless FNOSC is the Primary Oscillator with PLL.

Emulator Pin Placement: Communicate on PGC1 and PGD1

OSC2 is general purpose digital I/O pin

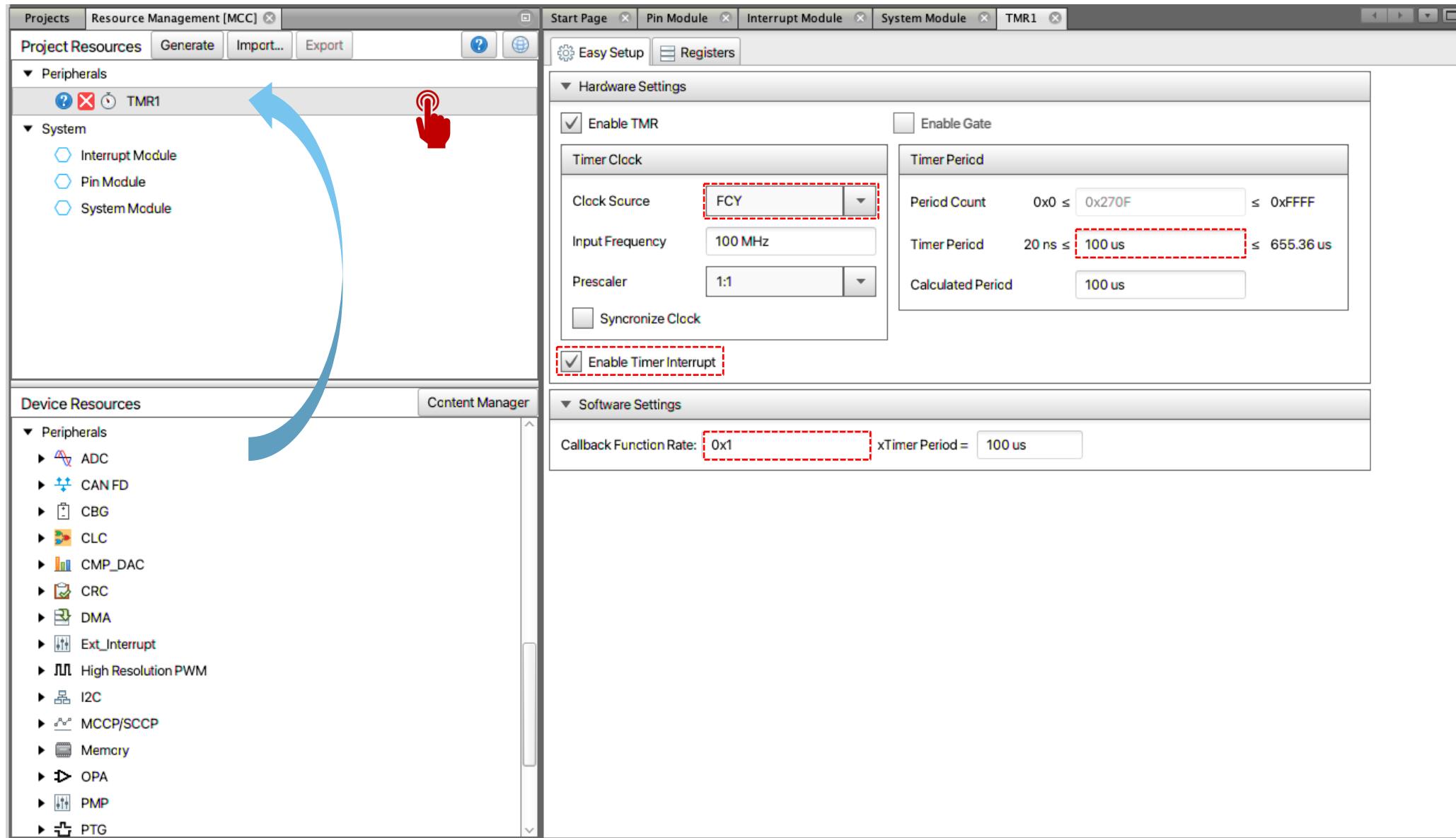
Reference Oscillator Output

CAN FD Clock Generator

Notifications Output Pin Manager: Grid View Notifications [MCC]

Search Results

MCC Drivers – Timer1 (100 us)



MCC Drivers – Pin Management

MPLAB X IDE v6.20 - MP102_Codebase : default

Projects Files Classes Services Resource M... MCC v5.5.0 Project Resources Generate Import... Export Selected Package : QFN28

Pin Module Interrupt Module System Module TMR1

Easy Setup Registers

Pin Name Module Function Custom Na... Start High Analog Output WPU WPD OD IOC

Pin Name	Module	Function	Custom Na...	Start High	Analog	Output	WPU	WPD	OD	IOC
RB4	Pin Module	GPIO	LED1							
RB8	ICD	PGD1								
RB9	ICD	PGC1								

Device Resources Content Manager

- I2C
- MCCP/SCCP
- Memory
- OPA
- PTG
- QEI
- SENT
- SPI
- UART

Examples

- Explorer 16/32 Demo
- LED Blink Example

Mikro-E Clicks

Audio and Voice

MP102_Codebase - Dashboard Navigator

Project Type: Application – Configuration: default

Device dsPIC33CK32MP102

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

Packs dsPIC33CK-MP_DFP (1.13.366)

Compiler Toolchain XC16 (v2.10) [/Applications/microchip/xc16/v2.1]

- Production Image: Optimization: gcc 0
- Device support information: dsPIC33CK-MP_DFP

Memory Usage Symbols disabled. Click to enable Load Sym

Pin Manager: Package View

RB9|PGC1
VS94A
VDD4A
RB13
RB12
RB10
RB11
RA14
RA13
RA12
RA11
RA10
RA9
RA8
RA7
RA6
RA5
RA4
AVDDA
AVSSA
VDD2A
VSS2A
RB0
RB1
R88|PGD1
R87
R86
R85
R84|LED1|GPIO
R83
R82
MICROCHIP
dsPIC33CK32MP102

Output Pin Manager: Grid View Notifications [MCC]

Package:	QFN28	Pin No:	4	5	6	7	8	13	14	15	16	17	18	19	20	21	22	25	26	27	28	1	2		
		Port A ▼												Port B ▼											
Module	Function	Direction	0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Clock ▼	CLKI	input																							
	CLKO	output																							
	OSCI	input																							
	OSCO	output																							
	REFI	input																							
	REFO	output																							
ICD ▼	PGCx	input																							
	PGDx	input																							
Pin Module ▼	GPIO	input																							
	GPIO	output																							
TMR1	T1CK	input																							

MCC Drivers – Interrupt Management & Code Generating

Screenshot of MPLAB X IDE v6.20 showing the Pin Manager and Interrupt Manager for a dsPIC33CK32MP102 device.

Interrupt Manager:

Module	Interrupt	Description	IRQ Number	Enabled	Priority	Context
Pin Module	CNAI	Change Notification A	2	<input type="checkbox"/>	1	OFF
Pin Module	CNBI	Change Notification B	3	<input type="checkbox"/>	1	OFF
Pin Module	CNCI	Change Notification C	19	<input type="checkbox"/>	1	OFF
Pin Module	CNDI	Change Notification D	75	<input type="checkbox"/>	1	OFF
DMT	DMTI	Dead Man Timer	45	<input type="checkbox"/>	1	OFF
TMR1	TI	Timer 1	1	<input checked="" type="checkbox"/>	3	OFF

Pin Manager: Package View:

The diagram shows the pinout for the dsPIC33CK32MP102 device, which is a QFN28 package. The pins are numbered 1 through 28. Functions include:

- Pins 1-7: RA0-RA3, AVDDA, VSS2A, VDD2A, RBO, RB1, RB2
- Pins 8-14: RA4-RB13, RB8|PGD1, RB7, RB6, RB5, RB4|LED1|GPIO
- Pins 15-21: NMCLR, RA1, RA2, RA3, RB12, RB11, RB10, RB9, RB8|PGC1, RB7, RB6, RB5
- Pins 22-28: RB13, RB12, RB11, RB10, RB9, RB8|PGC1, RB7, RB6, RB5, RB4|LED1|GPIO, RB3, RB2, RB1, VSS2A, VDD2A, RBO, RB1

Pin Manager: Grid View:

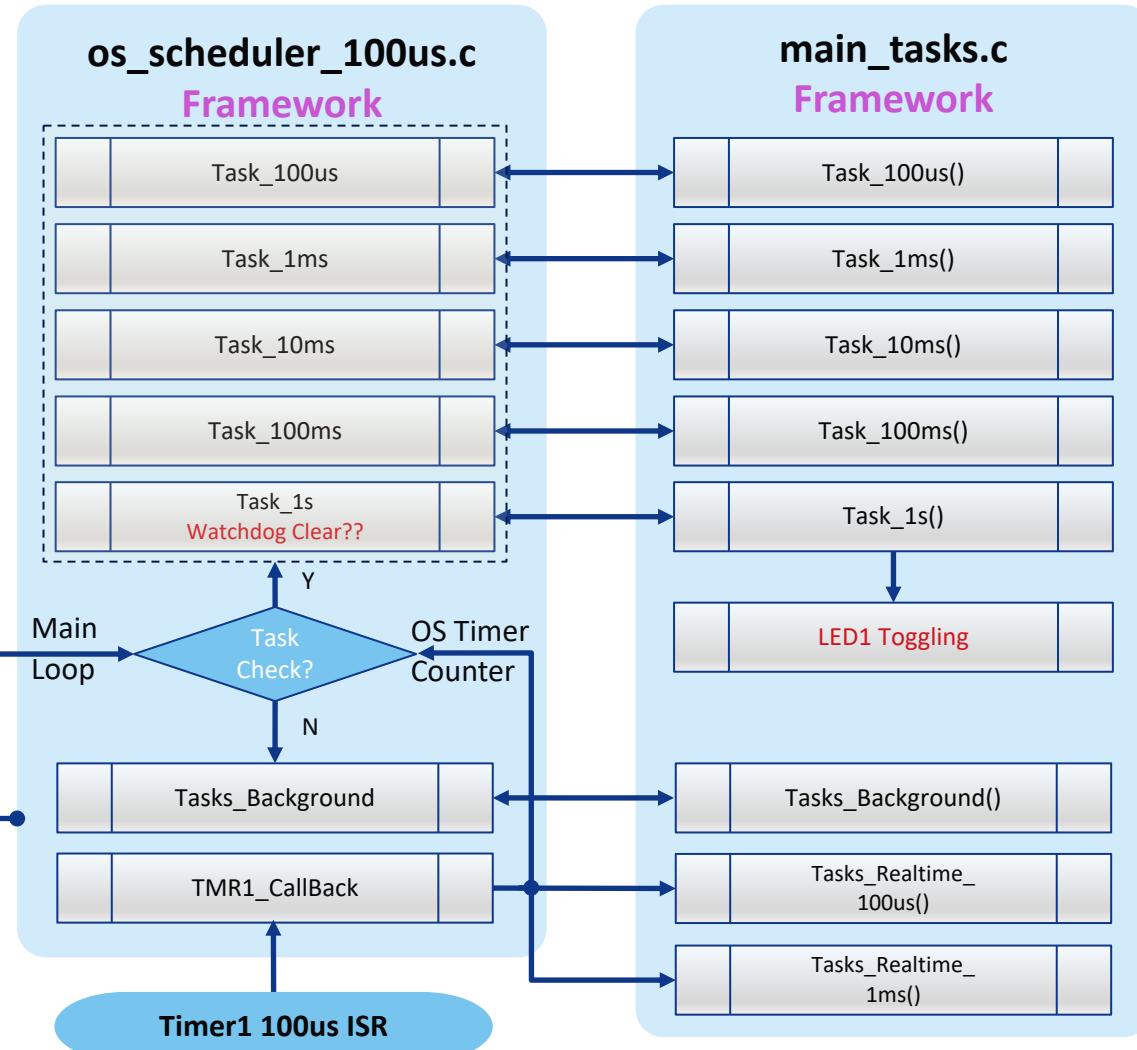
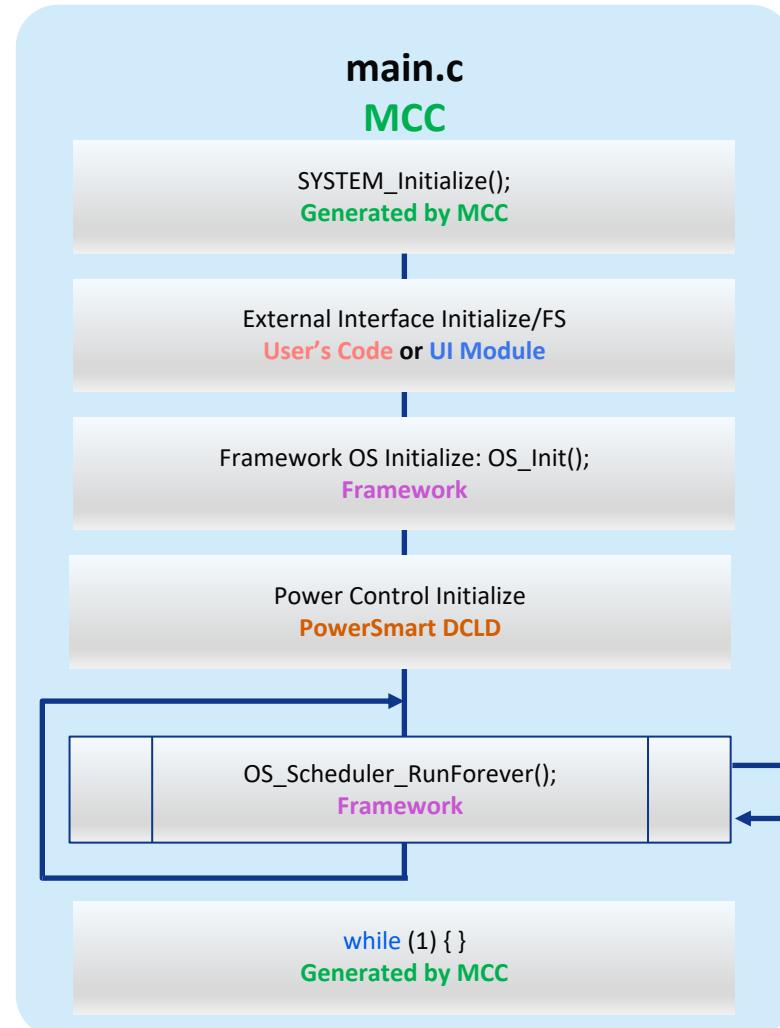
Package: QFN28	Pin No:	4	5	6	7	8	13	14	15	16	17	18	19	20	21	22	25	26	27	28	1	2			
		Port A ▼												Port B ▼											
Module	Function	Direction	0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Clock ▾	CLKI	input																							
	CLKO	output																							
	OSCI	input																							
	OSCO	output																							
	REFI	input																							
	REFO	output																							
ICD ▾	PGCx	input																							
	PGDx	input																							
Pin Module ▾	GPIO	input																							
	GPIO	output																							
TMR1	T1CK	input																							

MP102_Codebase - Dashboard:

- Project Type: Application – Configuration: default
- Device: dsPIC33CK32MP102
 - Checksum: Blank, no code loaded
 - CRC32: Hex file unavailable
- Packs: dsPIC33CK-MP_DFP (1.13.366), PICKit 4 (2.3.1848)
- Compiler Toolchain: XC16 (v2.10) [/Applications/microchip/xc16/v2.1], Production Image: Optimization: gcc 0
- Memory: Usage Symbols disabled. Click to enable Load Sym

SMPS Firmware Framework

Firmware Structure

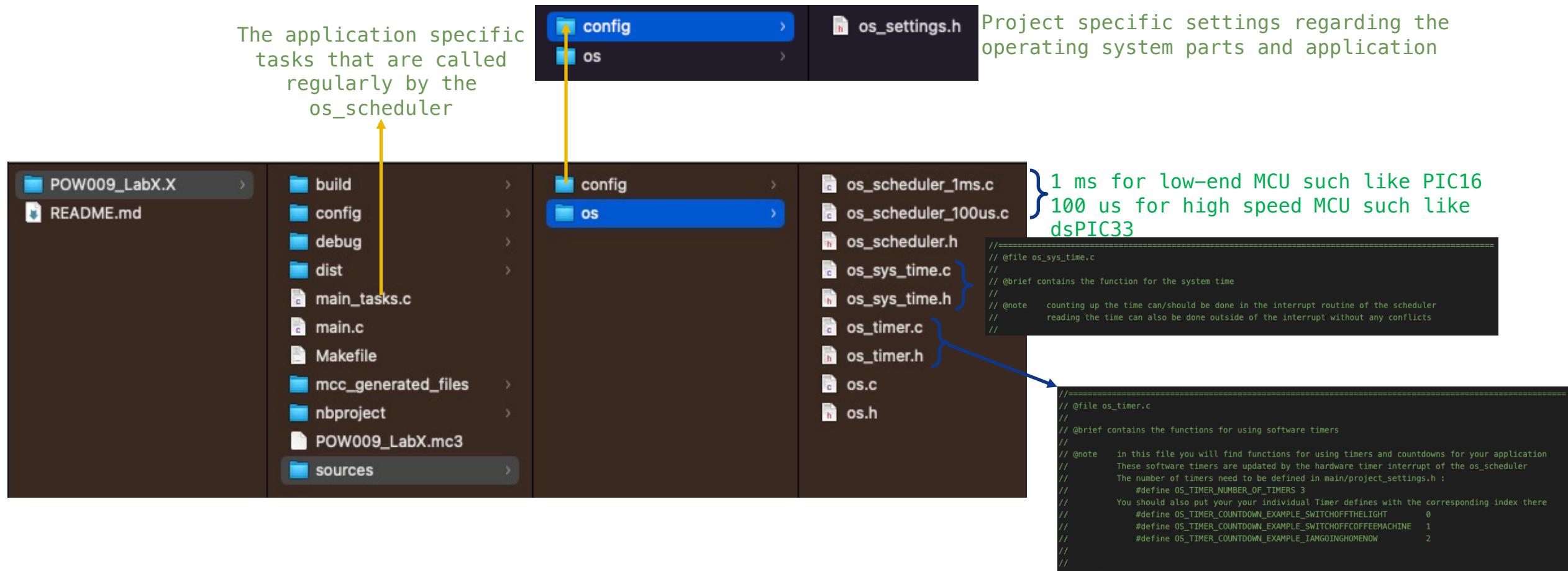


SMPS Firmware Framework

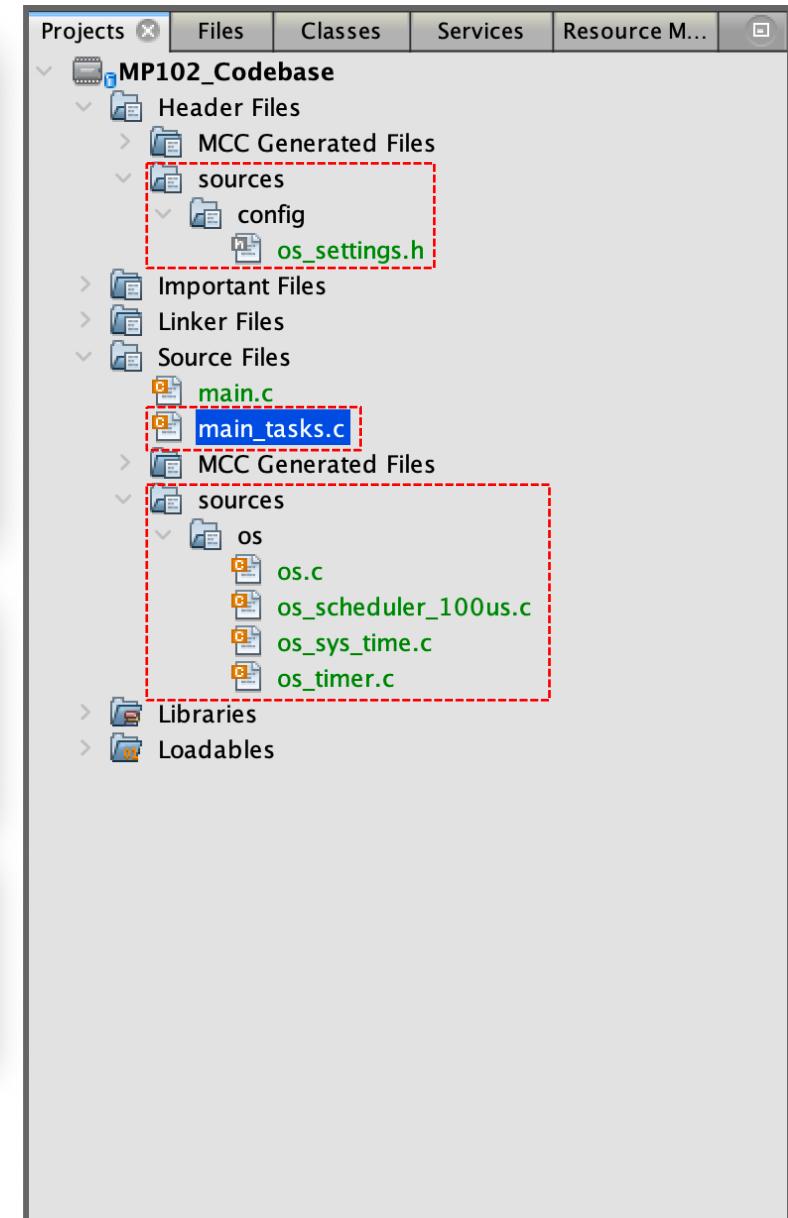
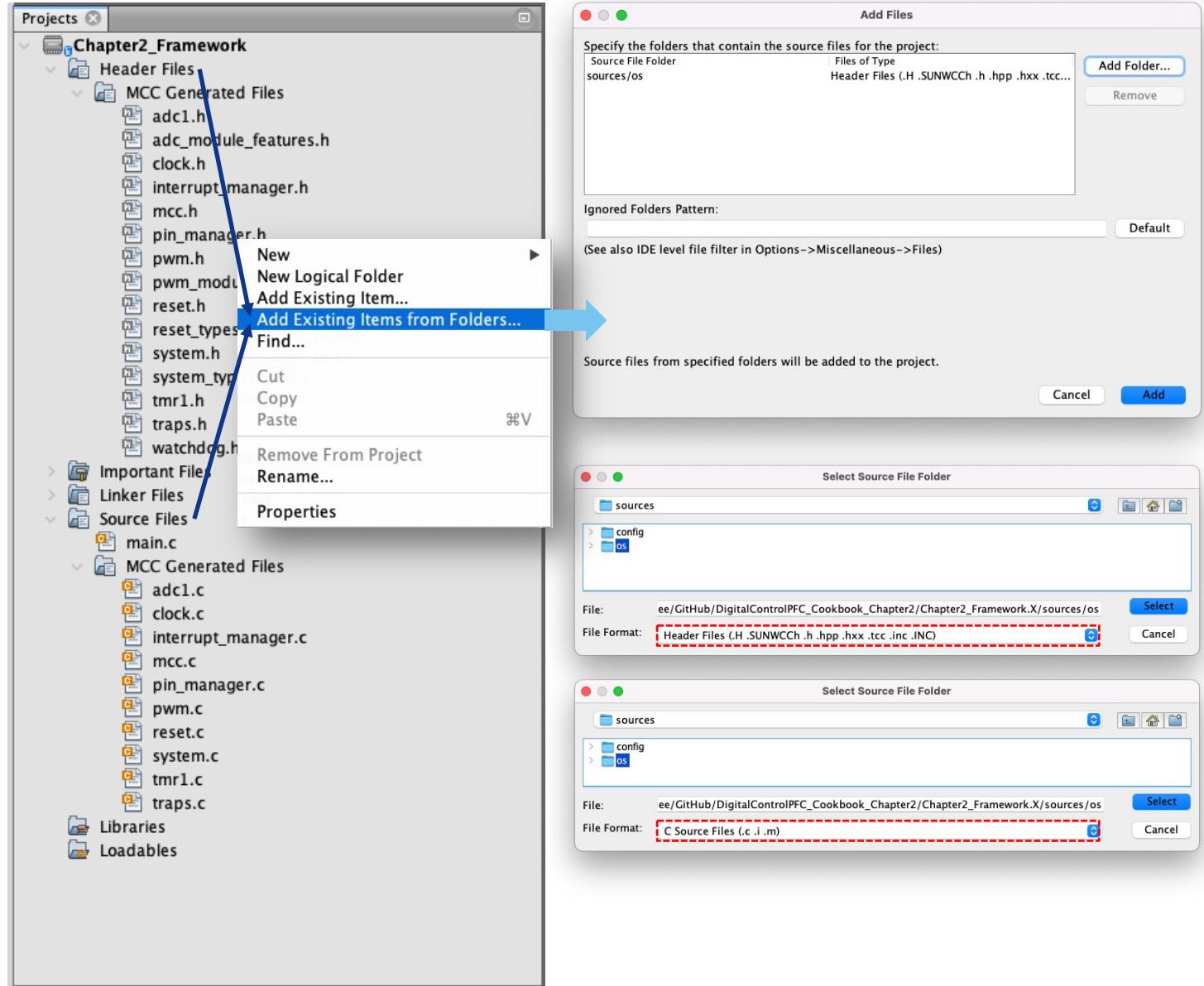
Directory Structure

Directory path	Description
ExampleProject.X	main project directory
ExampleProject.X\main.c	the main.c file should be in the main project directory because MCC puts it in here.
ExampleProject.X\main_tasks.c	the main_tasks.c file that the MCOS (Microchip Cooperative Operating System) requires should be in the same directory where the main.c file resides
ExampleProject.X\sources	all the other sub directories for sources and includes files should be under the directory 'sources'
ExampleProject.X\sources\app	the 'app' directory should contain the application logic. this is just a suggestion. depending on the project there might be a reason to have the application logic somewhere else. filenames in this directory should start with "app_"
ExampleProject.X\sources\config 系統設定檔案，例如: os_settings.h	the sub directory config is for storing different kind of configuration files. For configuring the MCOS (Microchip Cooperative Operating System) we use the file 'os_settings.h' (former 'project_settings.h' in older versions) In the case of the MCOS, is must be here, or else it cannot be found by the operating system files.
ExampleProject.X\sources\device 對外通訊連結，例如: dev_gui_comm.c/.h.h	the 'device' directory should contain drivers that handle I/O streams similar to a serial interface. Usually that devices have a send and receive function. In some of our projects there is the 'dev_gui_comm.c' communication layer for communication with the GUI here. filenames in this directory should start with "dev_"
ExampleProject.X\sources\driver 通訊以外的週邊驅動功能，例如: LED, I/O.....	the 'driver' directory should contain all the drivers that do not fit in the device directory. filenames in this directory should start with "drv_". Examples: drv_led.c; drv_button.c; drv_adc.c,
ExampleProject.X\sources\driver\power_controller 較為複雜的週邊驅動功能，例如: 電源控制	for more complex drivers that consists of multiple files it makes sense to create a sub directory. Example: In our power controller development boards we use the sub directory "power_controllers"
ExampleProject.X\sources\misc 共用的雜項檔案，例如: fault_common.c	directory to store miscellaneous files. Examples: global.h; fault/fault_common.c; ...
ExampleProject.X\sources\os OS管理目錄，例如: os_scheduler.c	directory for individual Operating System files, like the MCOS (Microchip Cooperative Operating System)

OS Scheduler Example



Integrate OS_Scheduler

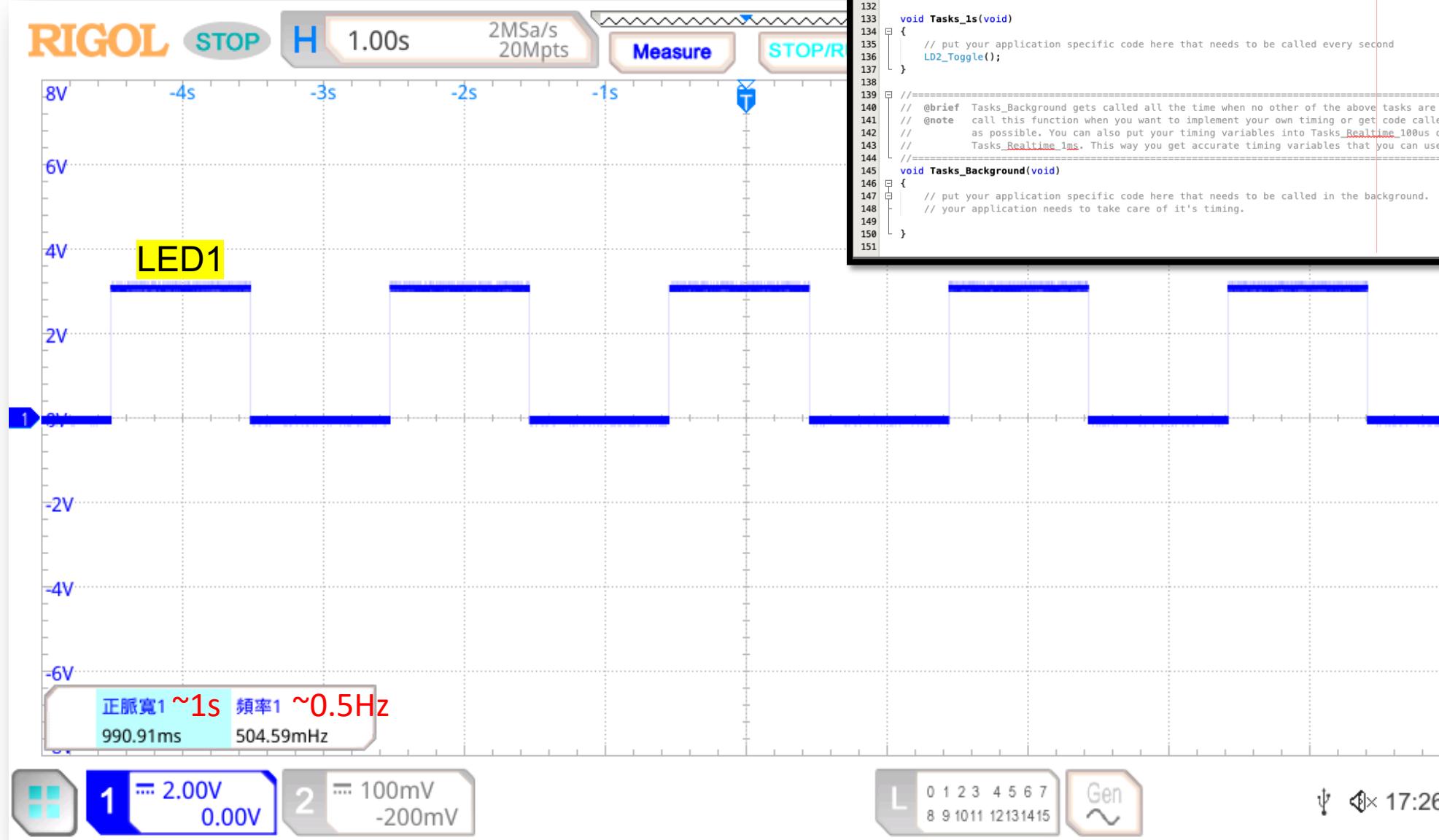


Main & Tasks

```
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
```

0.5Hz LED1 Toggle





May The *Power* Be With You

**KNOWLEDGE IS
POWER**

Massive power density in the smallest packages

Thanks!

