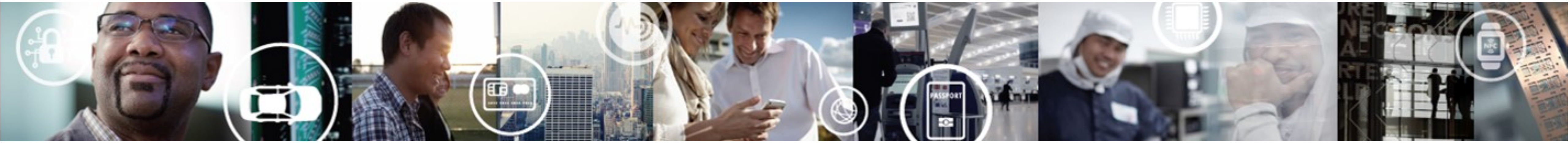
S12ZVM-EWP USER GUIDE - SOFTWARE

Ultra-Reliable MCUs for Industrial and Automotive Applications

Network address of RDB



SECURE CONNECTIONS
FOR A SMARTER WORLD

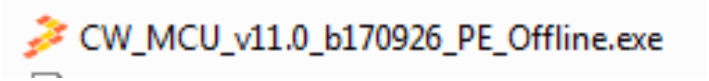
Contents

- Installing CodeWarrior for S12Z
- Import the Project
- FreeMASTER Conection
- MCAT configuration for a new PMSM
- Build and Debug Projects
- FreeMASTER Tuning

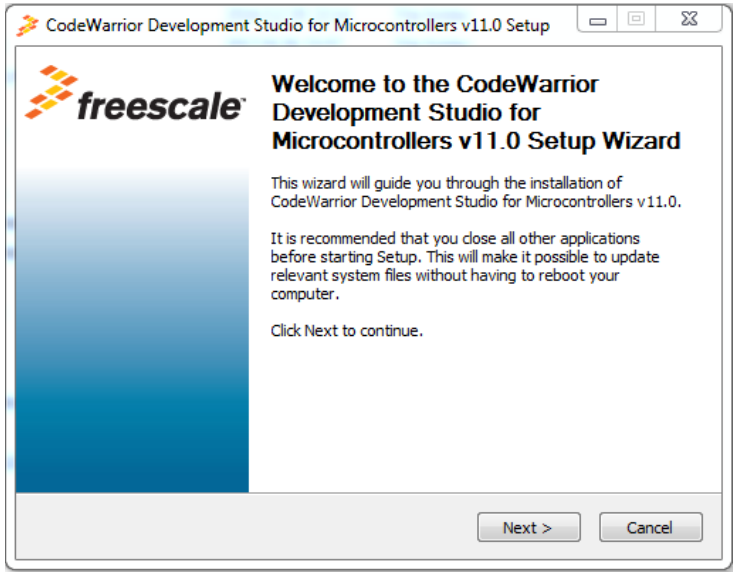
INSTALLING CODEWARRIOR FOR S12Z

Step-1

- Go to https://www.nxp.com/support/developer-resources/software-development-tools/codewarrior-development-tools/codewarrior-legacy/codewarrior-development-studios/codewarrior-for-microcontrollers/codewarrior-for-mcus-eclipse-ide-coldfire-56800-e-dsc-kinetis-qorivva-56xx-rs08-s08-s12z-11.0:CW-MCU10 to download latest version of CodeWarrior (Eclipse IDE) for S12Z
- Go into download folder, run the installation file, and the welcome window will appeared



CW_MCU_v11.0_b170926_PE_Offline.exe



CodeWarrior Development Studio for Microcontrollers v11.0 Setup

Welcome to the CodeWarrior Development Studio for Microcontrollers v11.0 Setup Wizard

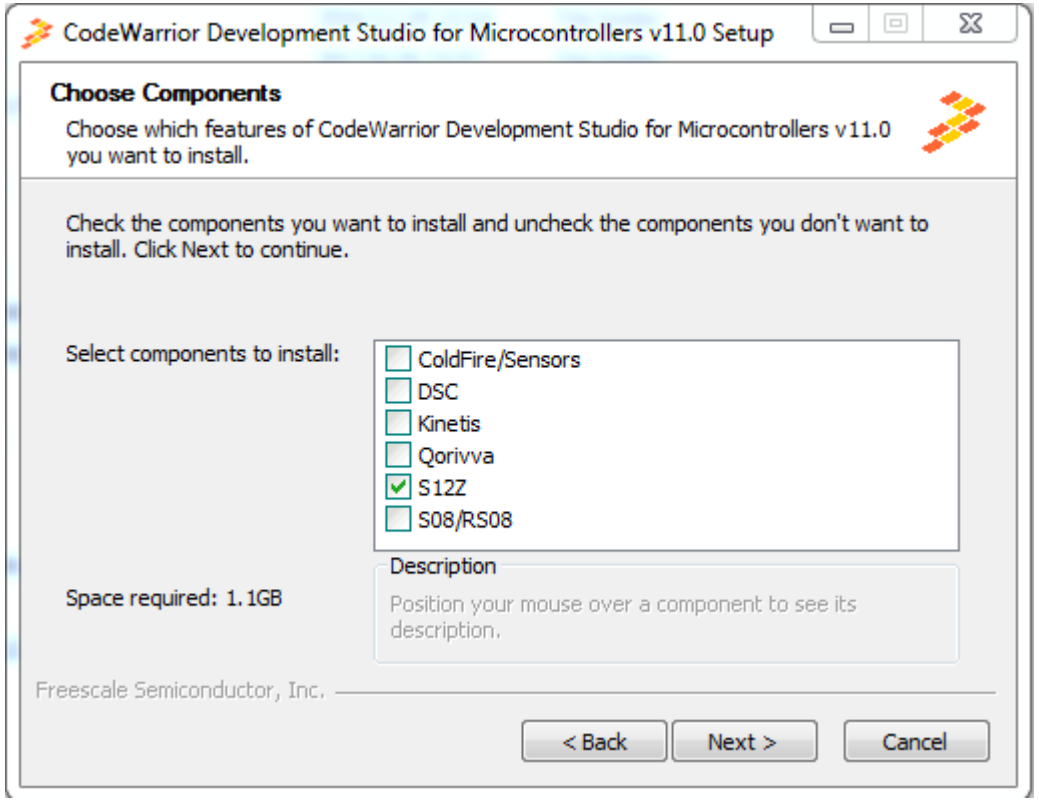
This wizard will guide you through the installation of CodeWarrior Development Studio for Microcontrollers v11.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Step-2

- Click the next button and then accept the license condition. For evaluate license, 30 days are limited.
- In choose components step, chose the components package you want to install. S12Z is the must for S12ZVM-EWP.



CodeWarrior Development Studio for Microcontrollers v11.0 Setup

Choose Components

Choose which features of CodeWarrior Development Studio for Microcontrollers v11.0 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

□ ColdFire/Sensors
□ DSC
□ Kinetis
□ Qorivva
☑ S12Z
□ S08/RS08

Space required: 1.1GB

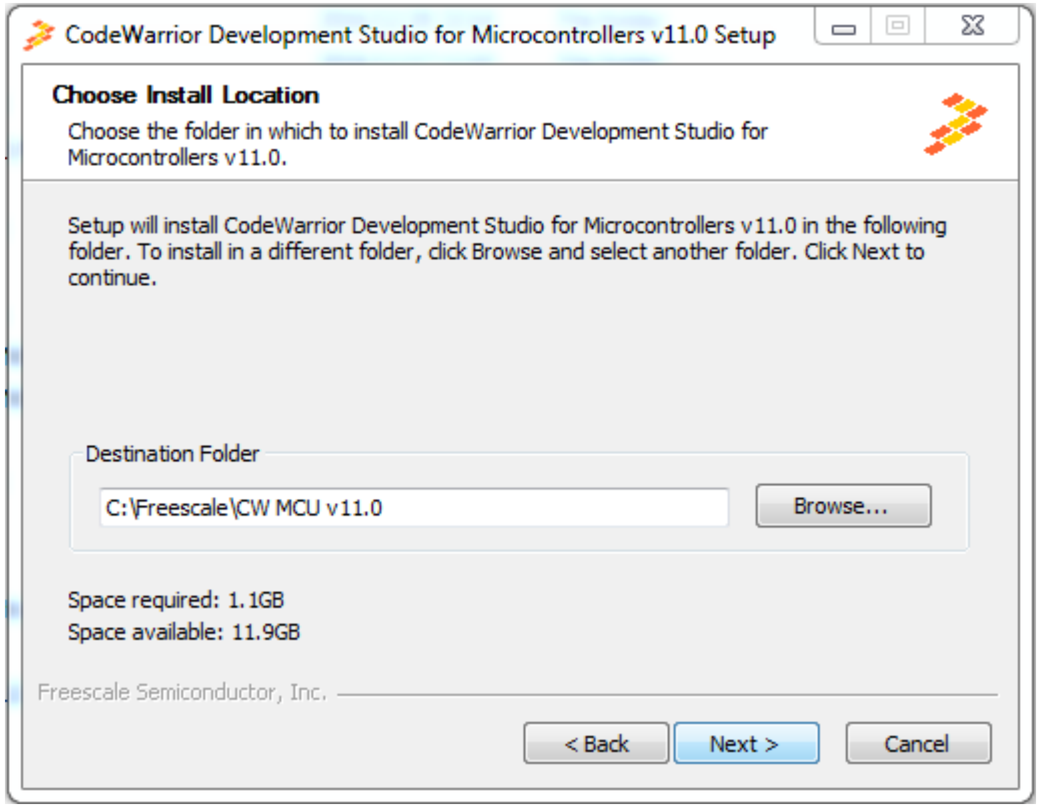
Description

Position your mouse over a component to see its description.

Freescale Semiconductor, Inc.

Step-3

- Choose install location. The default destination folder is C:\Freescale\CW MCU v11.0



CodeWarrior Development Studio for Microcontrollers v11.0 Setup

Choose Install Location

Choose the folder in which to install CodeWarrior Development Studio for Microcontrollers v11.0.

Setup will install CodeWarrior Development Studio for Microcontrollers v11.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Freescale\CW MCU v11.0

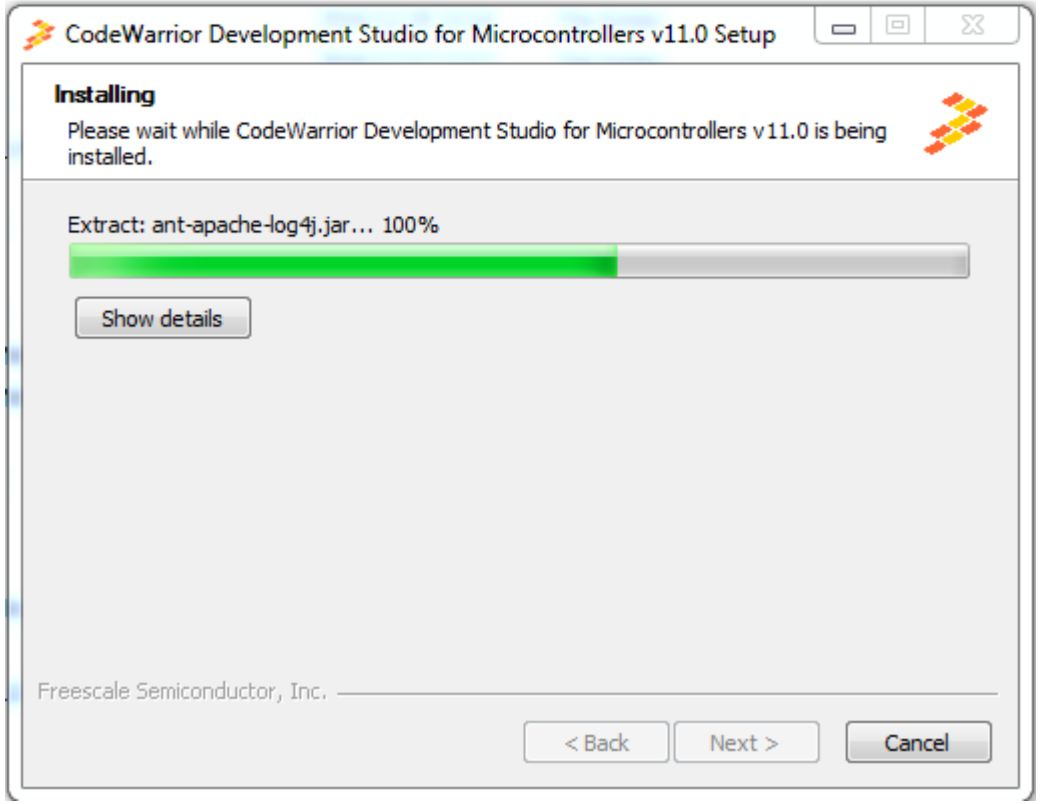
Space required: 1.1GB

Space available: 11.9GB

Step-4

- Click next button, and then software will enter to the real install process

- After the install process, the CodeWarrior can be used now



CodeWarrior Development Studio for Microcontrollers v11.0 Setup

Installing

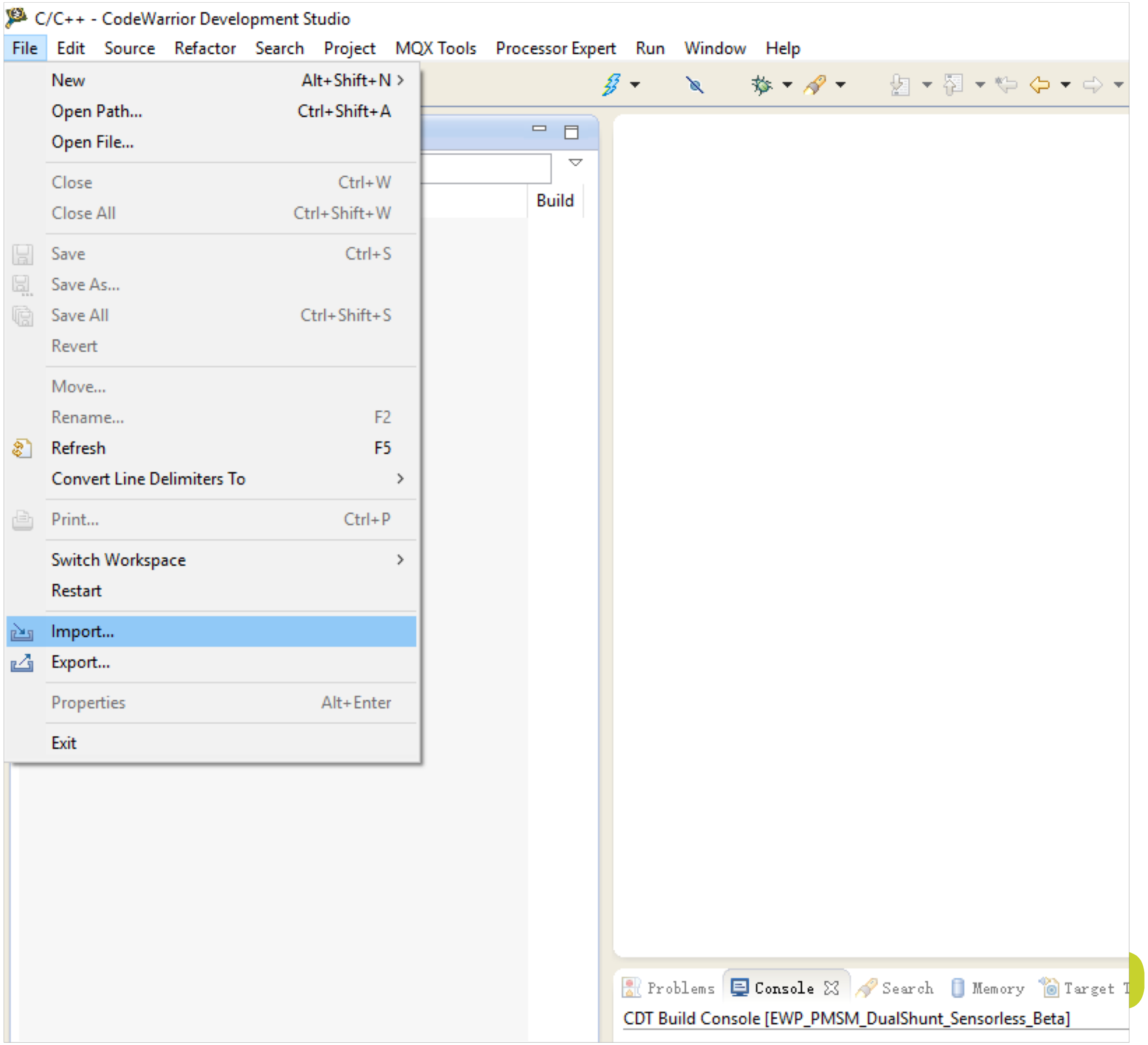
Please wait while CodeWarrior Development Studio for Microcontrollers v11.0 is being installed.

Extract: ant-apache-log4j.jar... 100%

IMPORT THE PROJECT

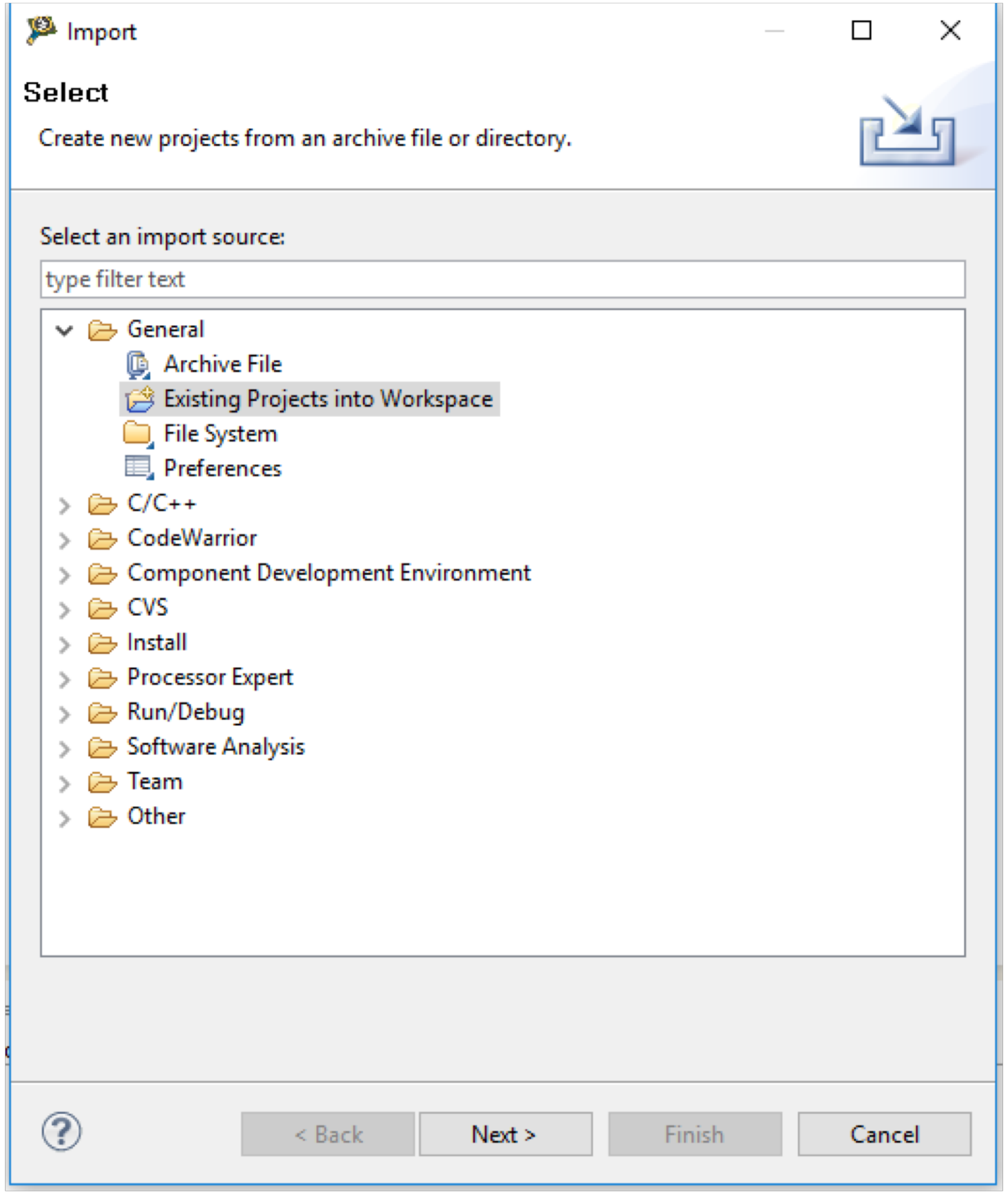
Step-1

- Open the CodeWarrior and click “File - > Import”



Step-2

- Double click
  “Existing Projects into Workspace”



Import

Select

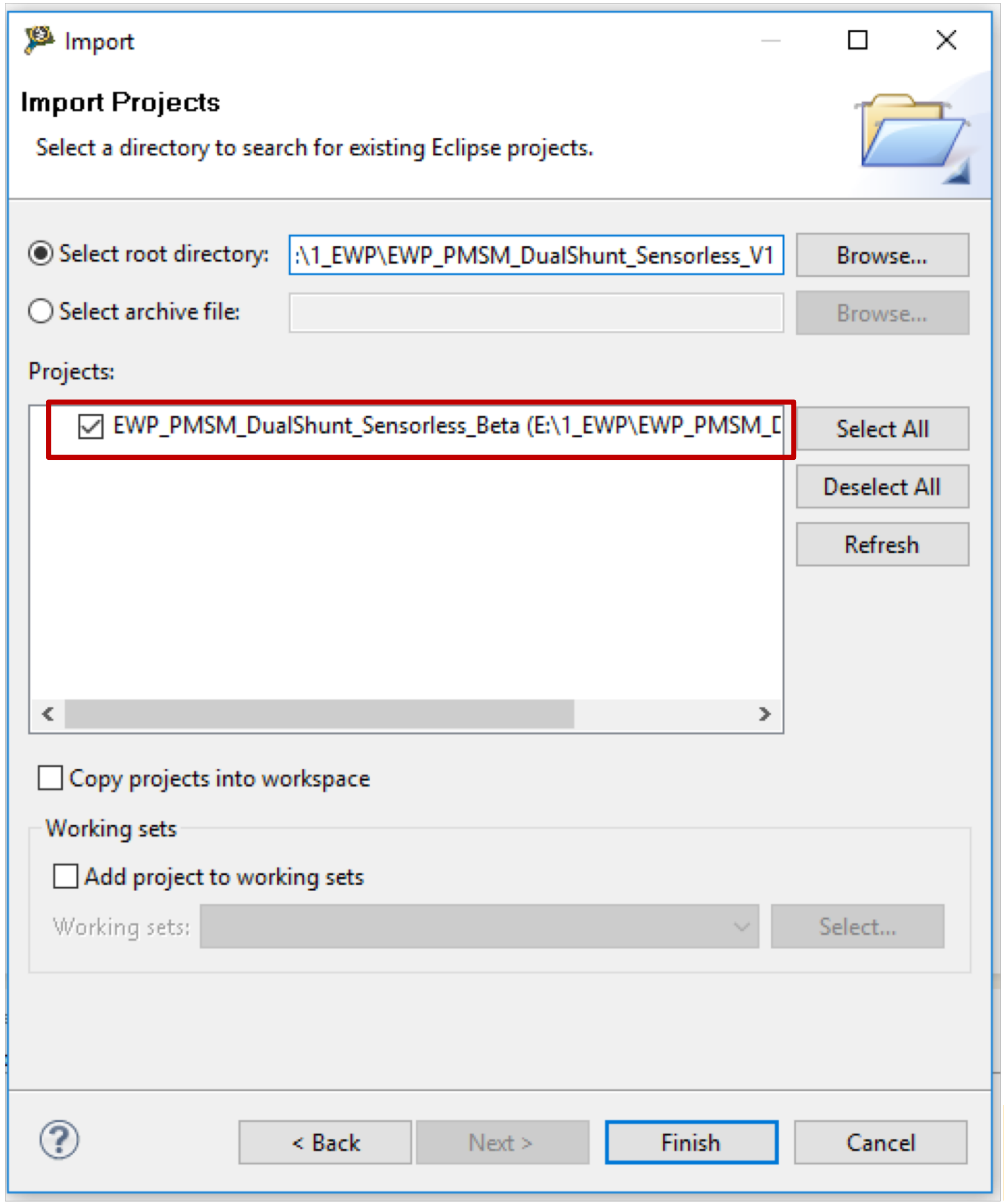
Create new projects from an archive file or directory.

Select an import source:

type filter text

Step-3

- Copy the directory of the project and then push the “Enter” Key.
- The project name will show in “Projects”
- Click “Finish” complete the import



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: :\1_EWP\EWP_PMSM_DualShunt_Sensorless_V1

○ Select archive file:

Projects:

☑ EWP_PMSM_DualShunt_Sensorless_Beta (E:\1_EWP\EWP_PMSM_D

☐ Copy projects into workspace

Working sets

☐ Add project to working sets

Working sets:

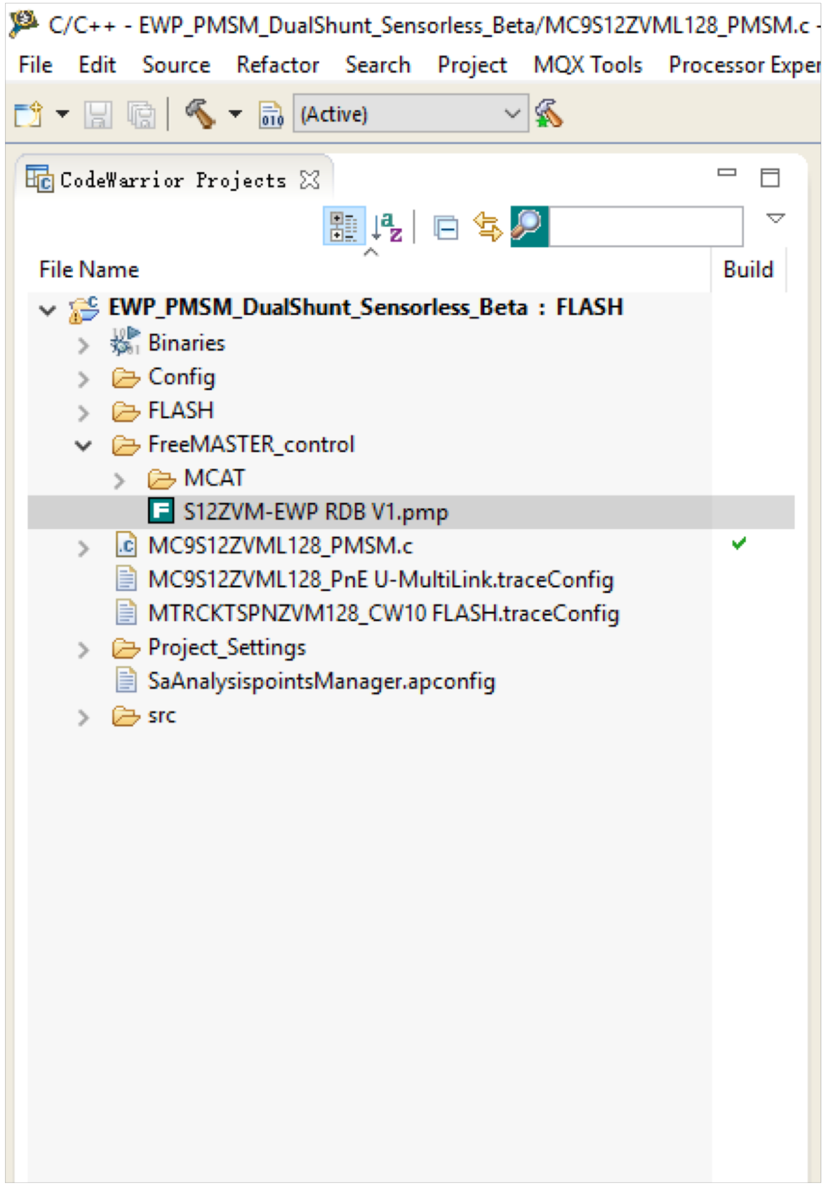
FREEMASTER CONECTION

FreeMASTER

- MCAT is not a stand alone function. It combined with FreeMASTER
- Download the FreeMASTER from the www.nxp.com/freemaster
- Install FreeMASTER in your PC
- Follow the STEPs to use MCAT control a new PMSM

Step-1

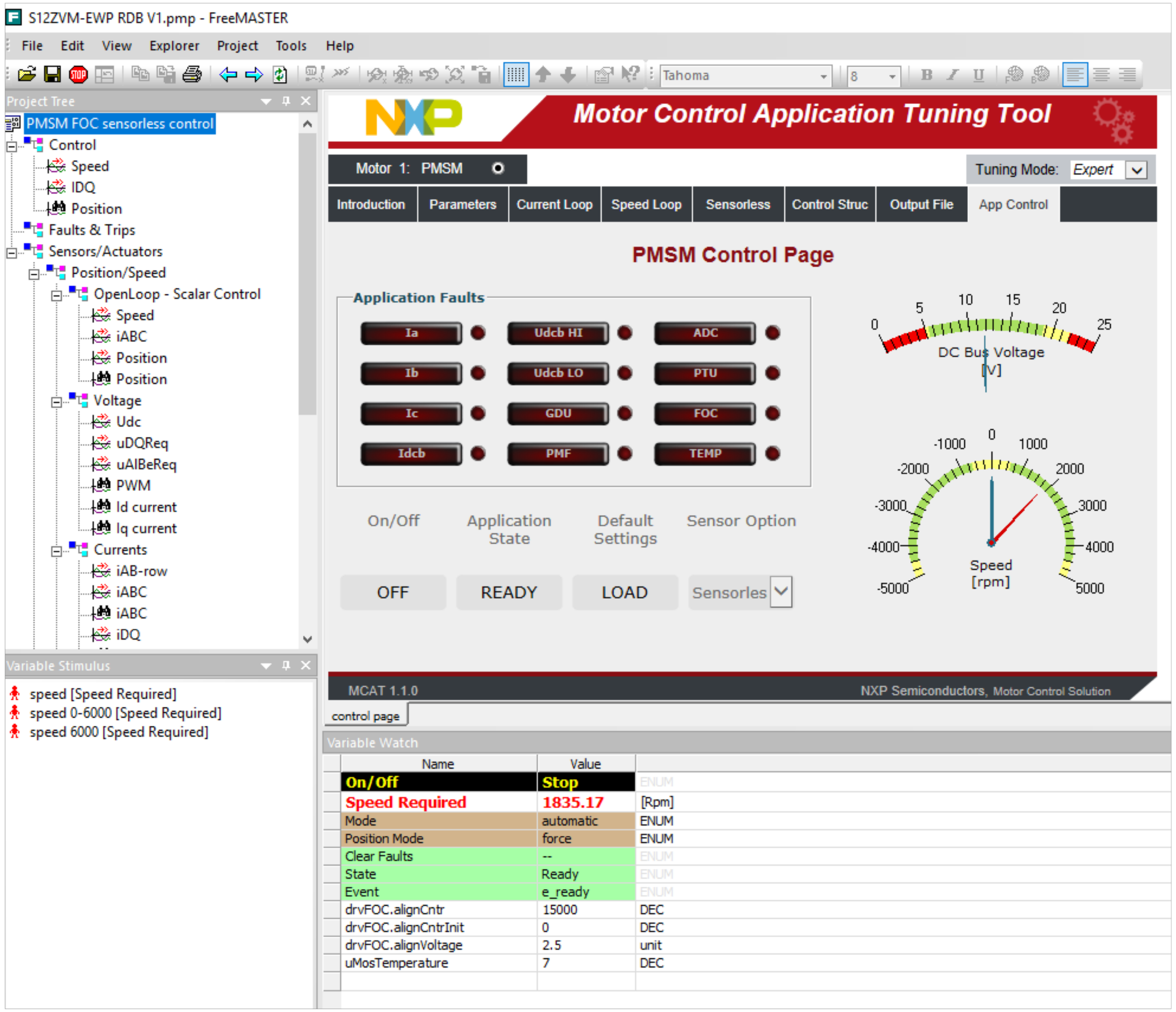
- Found the “S12ZVM-EWP RDB V1.pmp” in the CodeWarrior projects as shown in the right figure
- Double click the S12ZVM-EWP RDB V1.pmp it will open the FreeMASTER and MCAT



EWP_PMSM_DualShunt_Sensorless_Beta : FLASH
- Binaries
- Config
- FLASH
- FreeMASTER_control
  - MCAT
  - S12ZVM-EWP RDB V1.pmp
- MC9S12ZVML128_PMSM.c ✓
- MC9S12ZVML128_PnE U-MultiLink.traceConfig
- MTRCKTSPNZVM128_CW10 FLASH.traceConfig
- Project_Settings
- SaAnalysispointsManager.apconfig
- src

Step-2

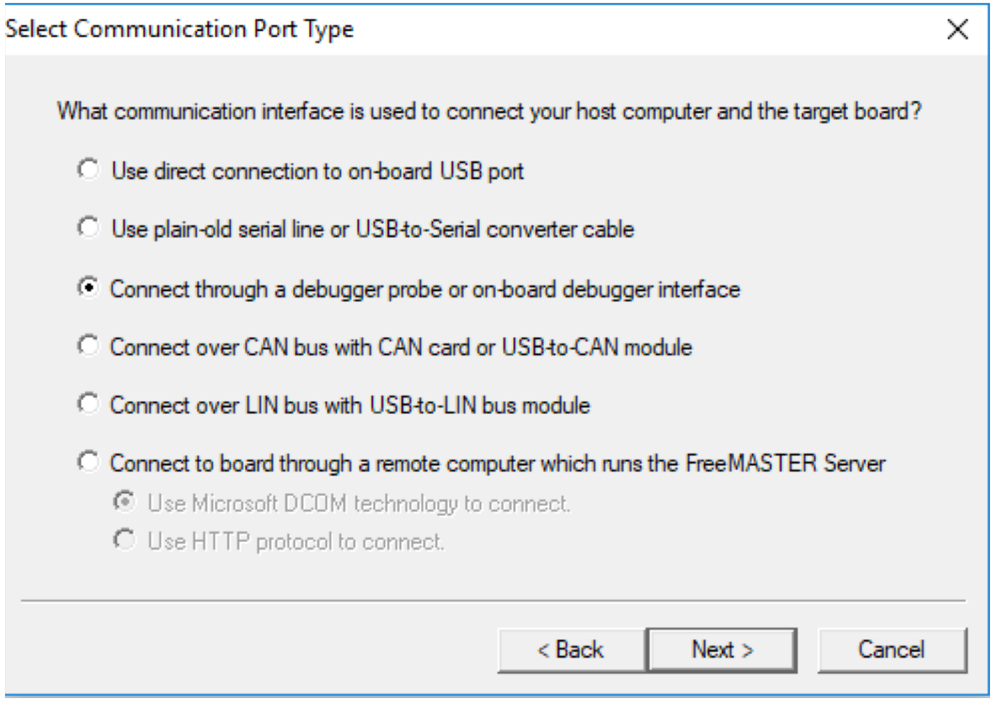
- The FreeMASTER with MCAT shown in right
- If everything is right, next step is to connect with the S12ZVM-EWP board
- There are 2 methods connecting with S12ZVM-EWP board. One is use BDM another one is use SCI
- For BDM connect, no need any other hardware modification, just connect S12ZVM-EWP board with PE multilink, but can’t support “recorder” function of FreeMASTER



Name	Value	
On/Off	Stop	ENUM
Speed Required	1835.17	[Rpm]
Mode	automatic	ENUM
Position Mode	force	ENUM
Clear Faults	--	ENUM
State	Ready	ENUM
Event	e_ready	ENUM
drvFOC.alignCntr	15000	DEC
drvFOC.alignCntrInit	0	DEC
drvFOC.alignVoltage	2.5	unit
uMosTemperature	7	DEC

Step-3

- BDM connect with FreeMASTER, click the “Tools->Connection Wizard”, and then click “Next”, then “Connect through a debugger probe…” then “USB Multilink or …”



Select Communication Port Type

What communication interface is used to connect your host computer and the target board?

○ Use direct connection to on-board USB port

○ Use plain-old serial line or USB-to-Serial converter cable

● Connect through a debugger probe or on-board debugger interface

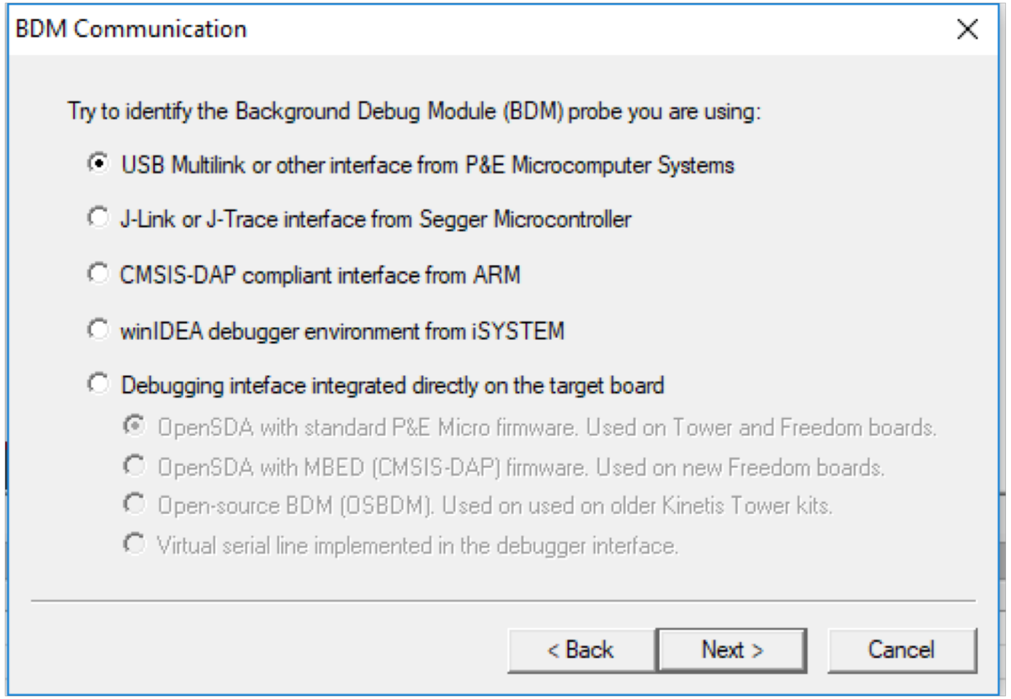
○ Connect over CAN bus with CAN card or USB-to-CAN module

○ Connect over LIN bus with USB-to-LIN bus module

○ Connect to board through a remote computer which runs the FreeMASTER Server

● Use Microsoft DCOM technology to connect.

○ Use HTTP protocol to connect.



BDM Communication

Try to identify the Background Debug Module (BDM) probe you are using:

● USB Multilink or other interface from P&E Microcomputer Systems

○ J-Link or J-Trace interface from Segger Microcontroller

○ CMSIS-DAP compliant interface from ARM

○ winIDEA debugger environment from iSYSTEM

○ Debugging inteface integrated directly on the target board

● OpenSDA with standard P&E Micro firmware. Used on Tower and Freedom boards.

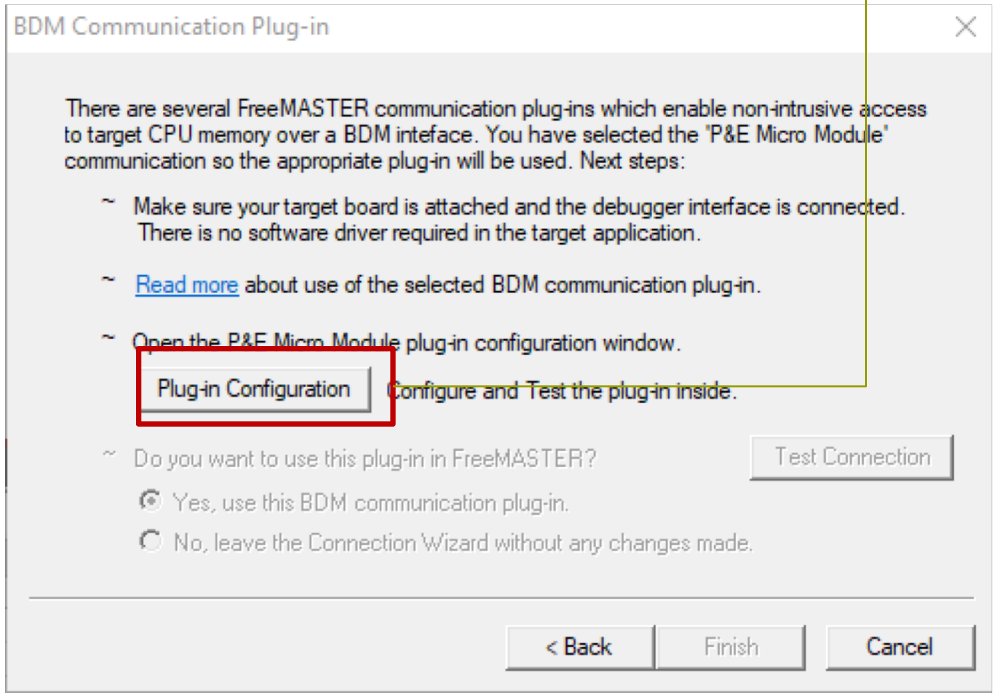
○ OpenSDA with MBED (CMSIS-DAP) firmware. Used on new Freedom boards.

○ Open-source BDM (OSBDM). Used on used on older Kinetis Tower kits.

○ Virtual serial line implemented in the debugger interface.

Step-4

- Click “Plug-in Configuration” and then select “P&E HCS12z” and then click “Test Connection”



BDM Communication Plug-in

There are several FreeMASTER communication plug-ins which enable non-intrusive access to target CPU memory over a BDM inteface. You have selected the 'P&E Micro Module' communication so the appropriate plug-in will be used. Next steps:

~ Make sure your target board is attached and the debugger interface is connected. There is no software driver required in the target application.

~ Read more about use of the selected BDM communication plug-in.

~ Open the P&E Micro Module plug-in configuration window.

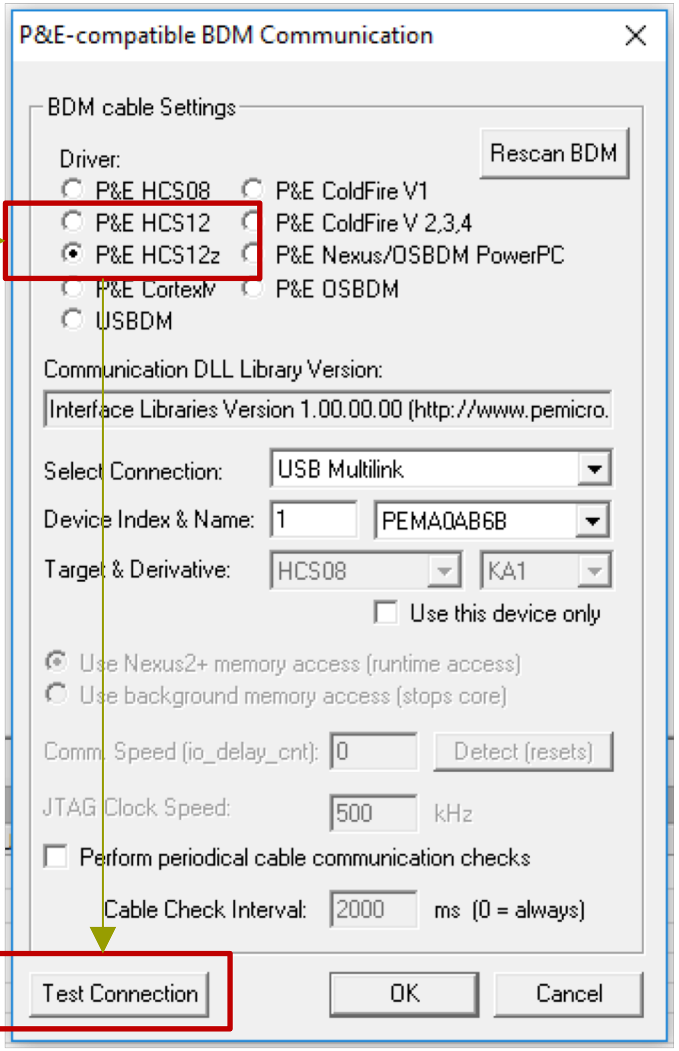
Plug-in Configuration Configure and Test the plug-in inside.

~ Do you want to use this plug-in in FreeMASTER? Test Connection

● Yes, use this BDM communication plug-in.

○ No, leave the Connection Wizard without any changes made.

< Back Finish Cancel



P&E-compatible BDM Communication

BDM cable Settings

Driver: Rescan BDM

○ P&E HCS08 ○ P&E ColdFire V1

○ P&E HCS12 ○ P&E ColdFire V 2,3,4

● P&E HCS12z ○ P&E Nexus/OSBDM PowerPC

○ P&E CortexM ○ P&E OSBDM

○ USBDM

Communication DLL Library Version:

Interface Libraries Version 1.00.00.00 (http://www.pemicro.

Select Connection: USB Multilink

Device Index & Name: 1 PEMA0AB6B

Target & Derivative: HCS08 KA1

□ Use this device only

● Use Nexus2+ memory access (runtime access)

○ Use background memory access (stops core)

Comm. Speed (io_delay_cnt): 0 Detect (resets)

JTAG Clock Speed: 500 kHz

□ Perform periodical cable communication checks

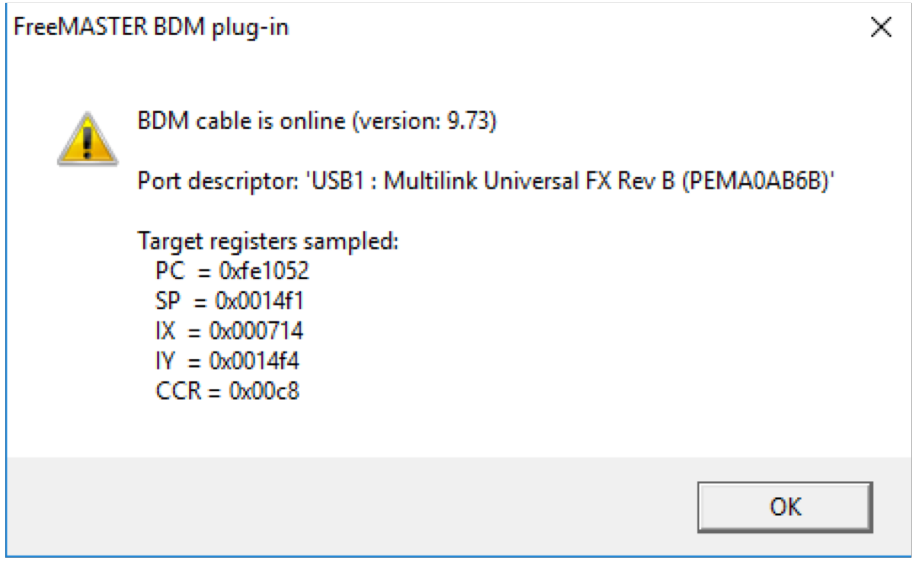
Cable Check Interval: 2000 ms (0 = always)

Test Connection OK Cancel

Step-5

- If show “BDM cable is online” now, FreeMASTER already connected to S12ZVM-EWP RDB

- If not show the “BDM cable is online” please check the connection

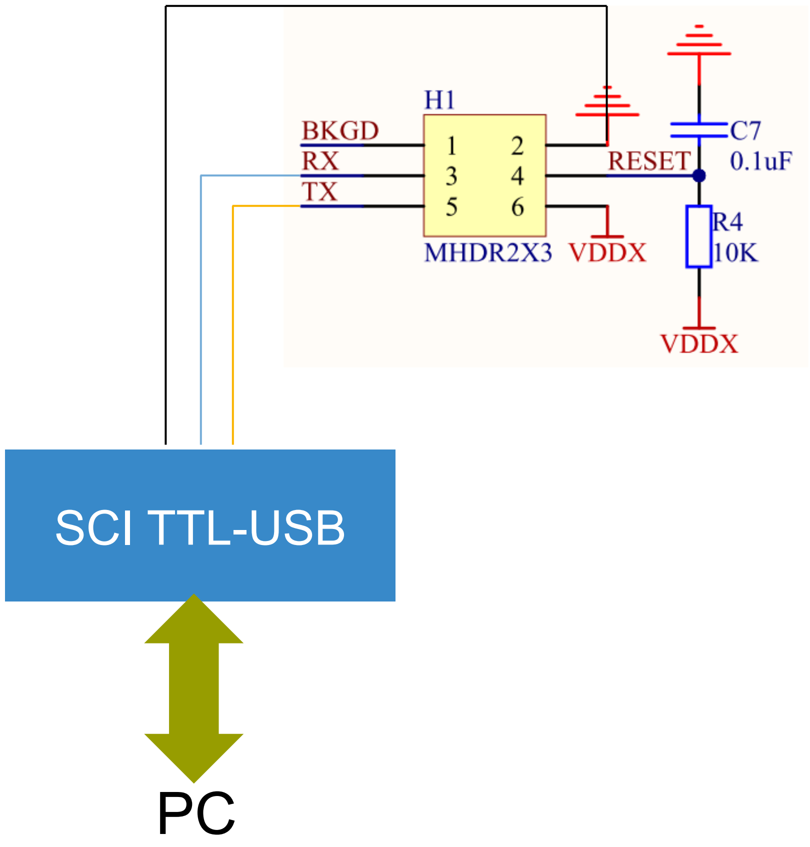


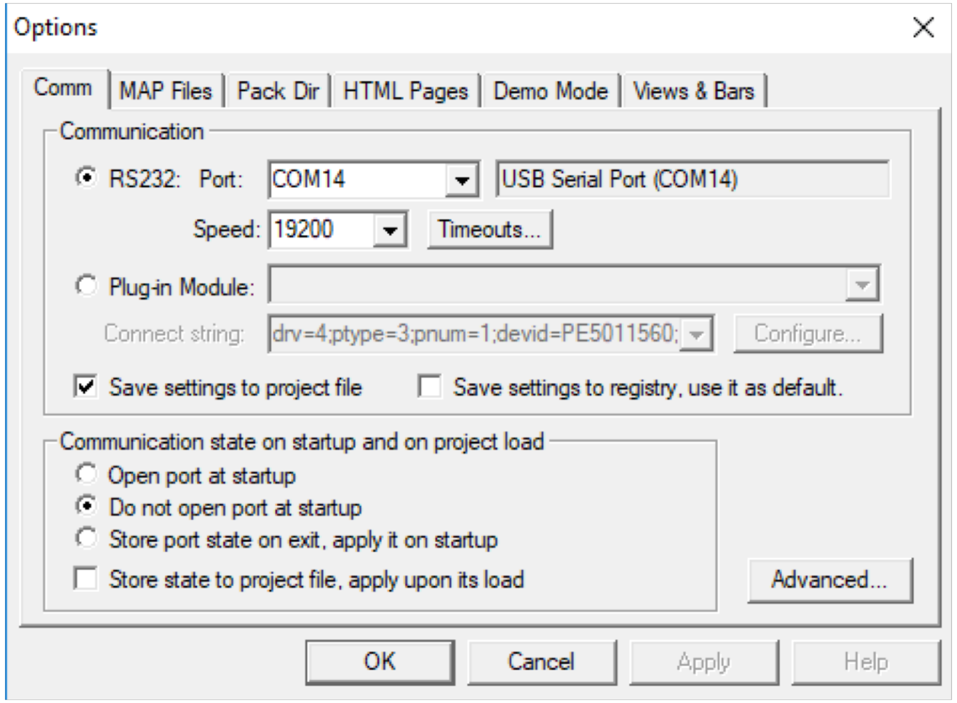
```
BDM cable is online (version: 9.73)

Port descriptor: 'USB1 : Multilink Universal FX Rev B (PEMA0AB6B)'

Target registers sampled:
  PC  = 0xfe1052
  SP  = 0x0014f1
  IX  = 0x000714
  IY  = 0x0014f4
  CCR = 0x00c8
```

Notice: if Multilink is using in debugging, it can’t be used by FreeMASTER. Stop debug and then connect FreeMASTER by using BDM method, if you want to use the “recorder” function of FreeMASTER, strongly recommend to use the SCI connection method

Step-6

- SCI connect method with FreeMASTER
- Connect board “H1 Pin” with SCI TTL to USB tools
- “Project->Options…” and then select RS232 port and speed “19200”





Communication

● RS232: Port: COM14 USB Serial Port (COM14)

Speed: 19200 Timeouts...

○ Plug-in Module:

Connect string: drv=4;ptype=3;pnum=1;devid=PE5011560; Configure...

☑ Save settings to project file ☐ Save settings to registry, use it as default.

Communication state on startup and on project load

○ Open port at startup

● Do not open port at startup

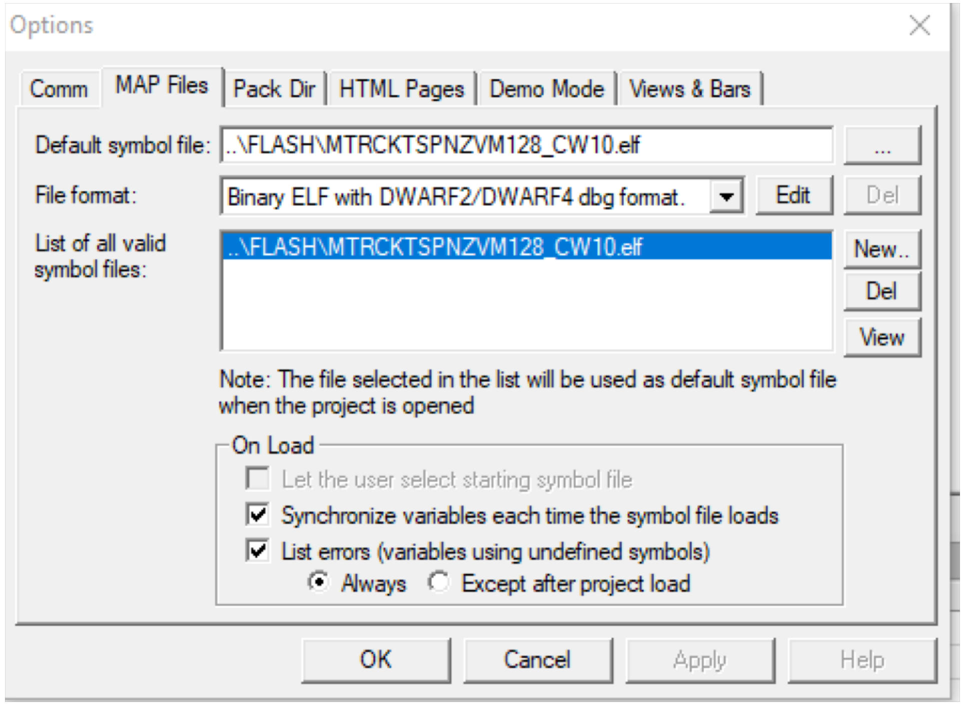
○ Store port state on exit, apply it on startup

☐ Store state to project file, apply upon its load

Advanced...

Step-7

- Select “….elf” or “…map” in MAP Files in options card. Keep the .elf file or .map file is the latest one. Other wise it will bring the dis-order display of variables



Default symbol file: ..\FLASH\MTRCKTSPNZVM128_CW10.elf

File format: Binary ELF with DWARF2/DWARF4 dbg format.

List of all valid symbol files: ..\FLASH\MTRCKTSPNZVM128_CW10.elf

Note: The file selected in the list will be used as default symbol file when the project is opened

On Load

□ Let the user select starting symbol file

☑ Synchronize variables each time the symbol file loads

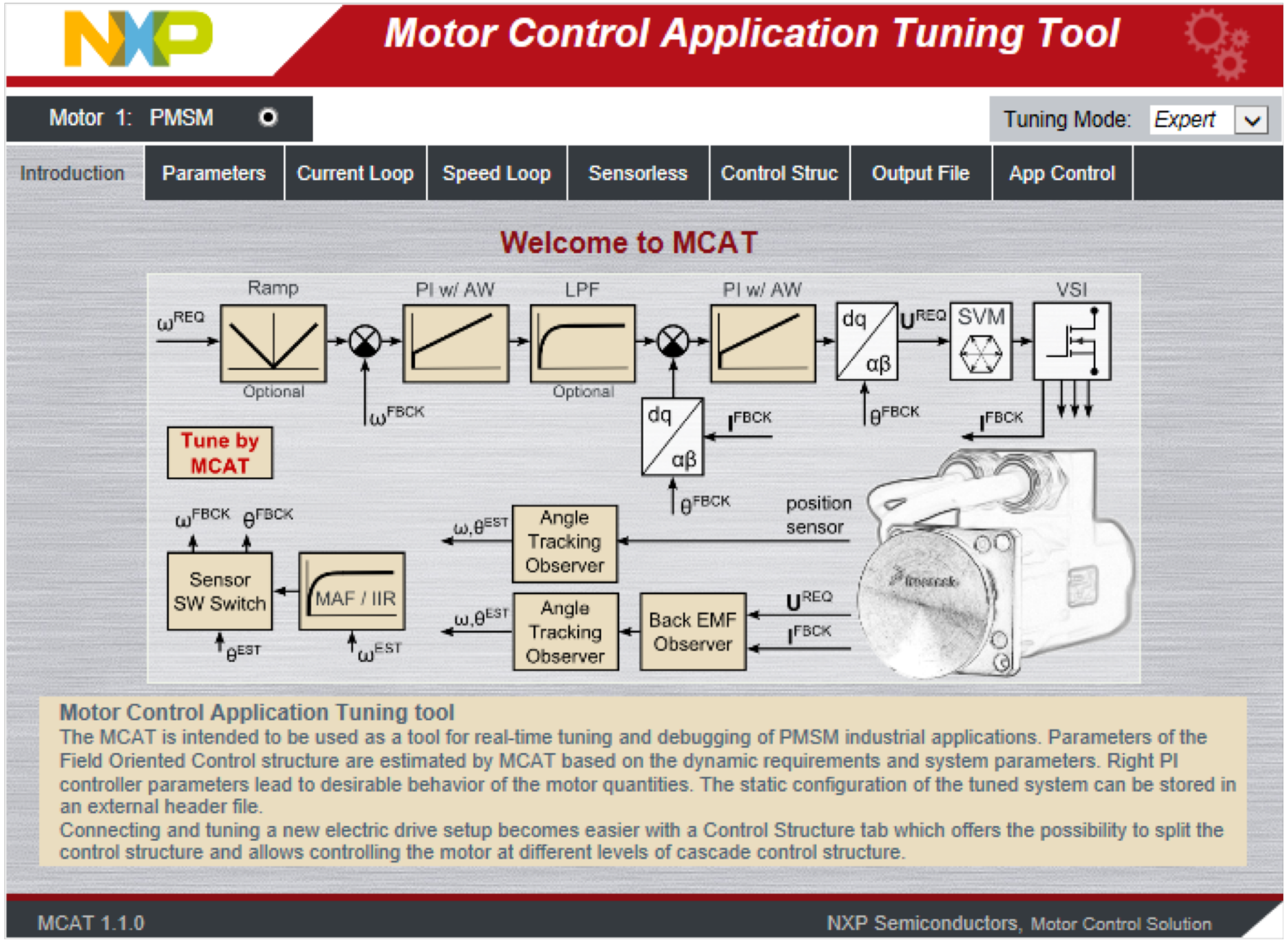
☑ List errors (variables using undefined symbols)

● Always ○ Except after project load

MCAT CONFIGURATION FOR A NEW PMSM

Step-1

- MCAT is a HTML based configure tools for NXP motor control.



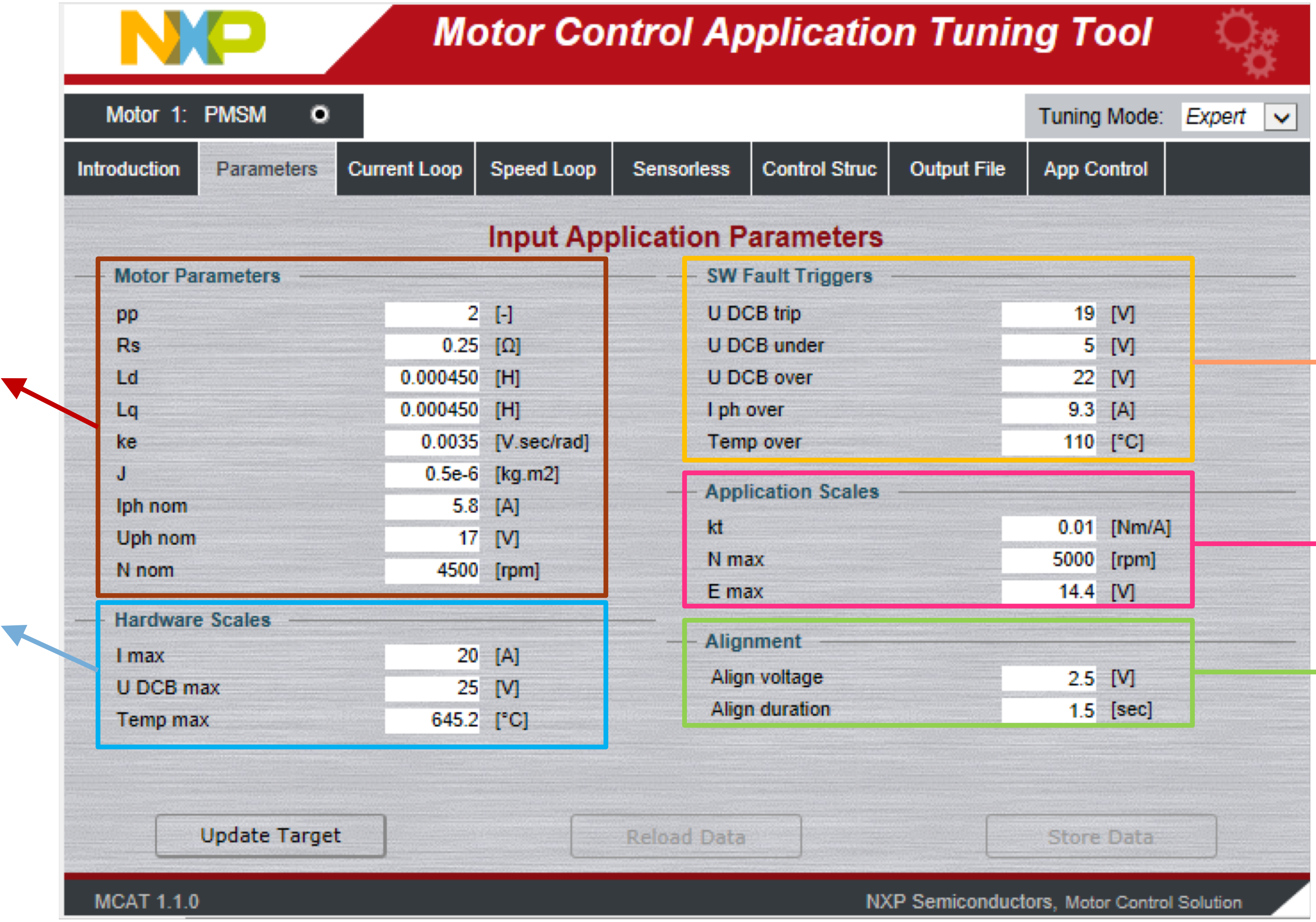
Motor Control Application Tuning tool

The MCAT is intended to be used as a tool for real-time tuning and debugging of PMSM industrial applications. Parameters of the Field Oriented Control structure are estimated by MCAT based on the dynamic requirements and system parameters. Right PI controller parameters lead to desirable behavior of the motor quantities. The static configuration of the tuned system can be stored in an external header file.

Connecting and tuning a new electric drive setup becomes easier with a Control Structure tab which offers the possibility to split the control structure and allows controlling the motor at different levels of cascade control structure.

Step-2

- Configure the PMSM motor parameters



Motor 1: PMSM

Tuning Mode: Expert

Input Application Parameters

Motor Parameters

pp	2	[-]
Rs	0.25	[Ω]
Ld	0.000450	[H]
Lq	0.000450	[H]
ke	0.0035	[V.sec/rad]
J	0.5e-6	[kg.m2]
Iph nom	5.8	[A]
Uph nom	17	[V]
N nom	4500	[rpm]

Hardware Scales

I max	20	[A]
U DCB max	25	[V]
Temp max	645.2	[°C]

SW Fault Triggers

U DCB trip	19	[V]
U DCB under	5	[V]
U DCB over	22	[V]
I ph over	9.3	[A]
Temp over	110	[°C]

Application Scales

kt	0.01	[Nm/A]
N max	5000	[rpm]
E max	14.4	[V]

Alignment

Align voltage	2.5	[V]
Align duration	1.5	[sec]

Motor Basic Parameters
Refer AN4680 for motor
parameters measurement

Hardware Scales
For S12ZVM-EWP RDB
Imax = 20A
U DCB max = 25V
Temp max = 645.2

Protection Point of the system

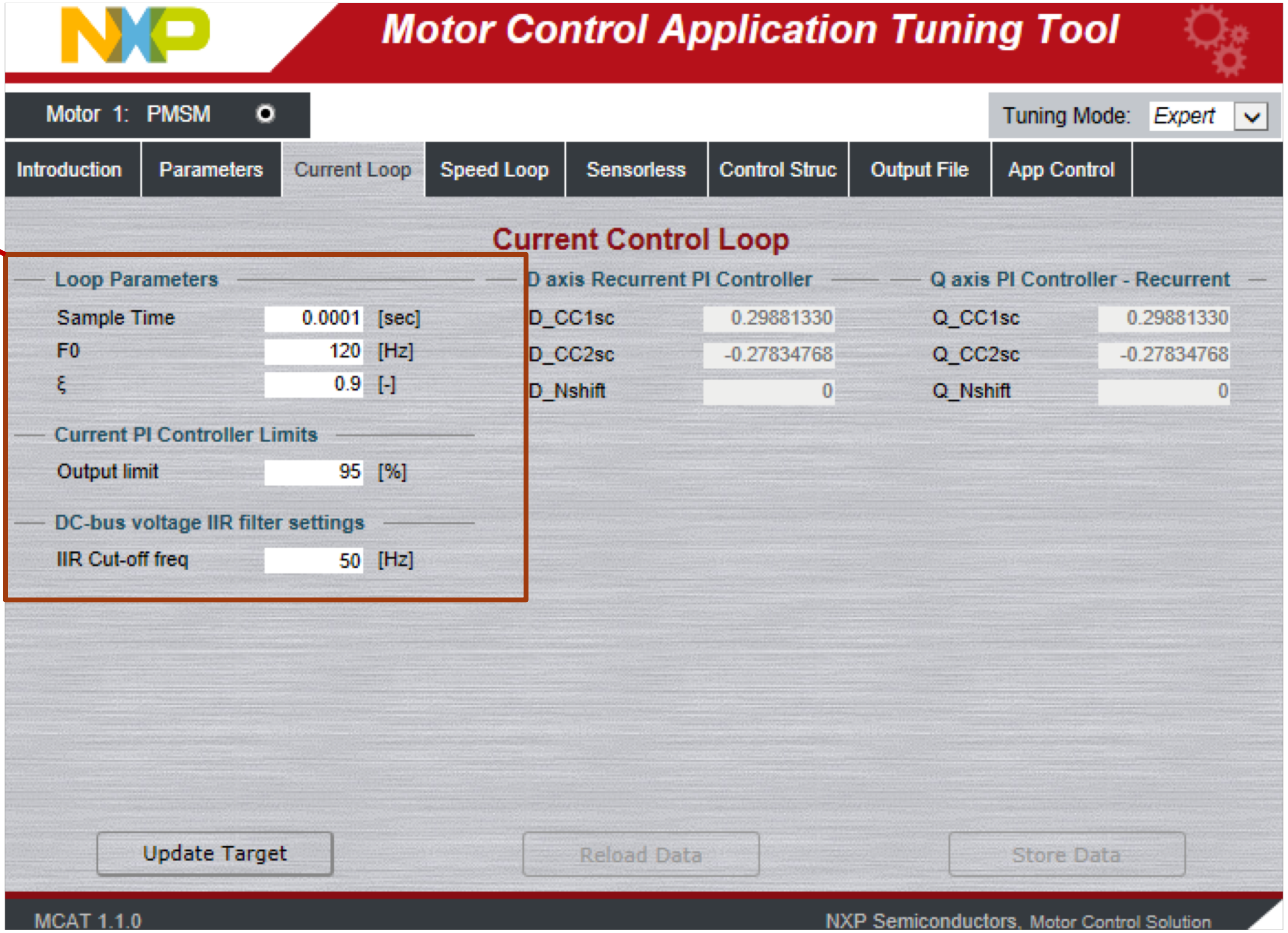
Application Scales

Align voltage and duration
Used for sensor-less control

Step-3

Set the current loop parameters Higher, the dynamic performance also better, but more noise

Current loop parameters and motor parameters together decide D axis and Q axis PI parameters



Motor 1: PMSM

Tuning Mode: Expert

Current Control Loop

Loop Parameters

Sample Time	0.0001	[sec]
F0	120	[Hz]
ξ	0.9	[-]

Current PI Controller Limits

Output limit	95	[%]

DC-bus voltage IIR filter settings

IIR Cut-off freq	50	[Hz]

D axis Recurrent PI Controller

D_CC1sc	0.29881330
D_CC2sc	-0.27834768
D_Nshift	0

Q axis PI Controller - Recurrent

Q_CC1sc	0.29881330
Q_CC2sc	-0.27834768
Q_Nshift	0

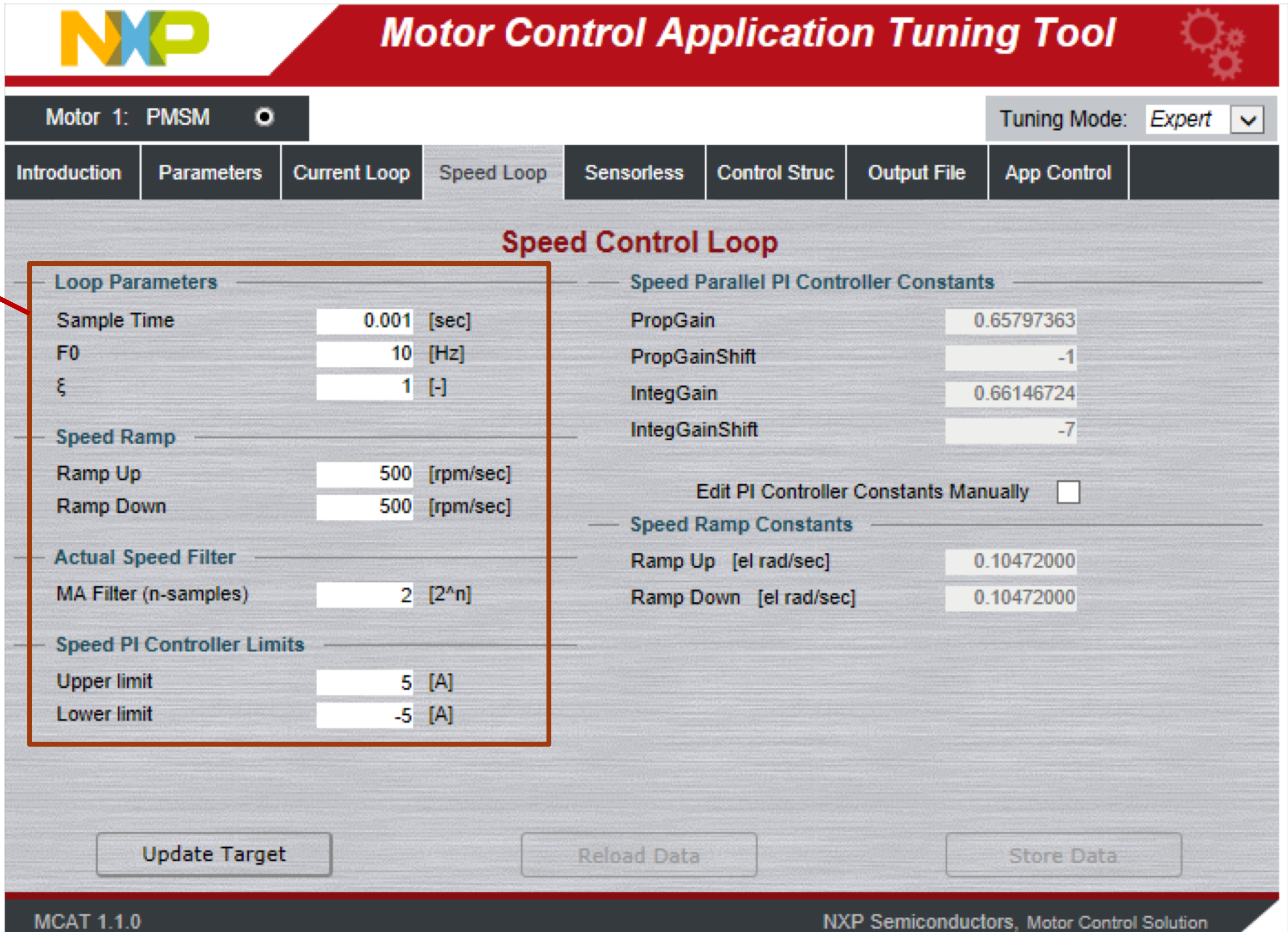
Step-4

Set the speed loop parameters Higher, the dynamic performance also better, but more noise

Speed loop parameters with motor rotor and its loader J together decide speed PI parameters

J can’t get it precise, so speed PI controller can be manually edit

Speed ramp to make sure speed PI more stable



Tuning Mode: Expert

Speed Control Loop

Loop Parameters

Sample Time	0.001	[sec]
F0	10	[Hz]
ξ	1	[-]

Speed Ramp

Ramp Up	500	[rpm/sec]
Ramp Down	500	[rpm/sec]

Actual Speed Filter

MA Filter (n-samples)	2	[2^n]

Speed PI Controller Limits

Upper limit	5	[A]
Lower limit	-5	[A]

Speed Parallel PI Controller Constants

PropGain	0.65797363
PropGainShift	-1
IntegGain	0.66146724
IntegGainShift	-7

Edit PI Controller Constants Manually □

Speed Ramp Constants

Ramp Up [el rad/sec]	0.10472000
Ramp Down [el rad/sec]	0.10472000

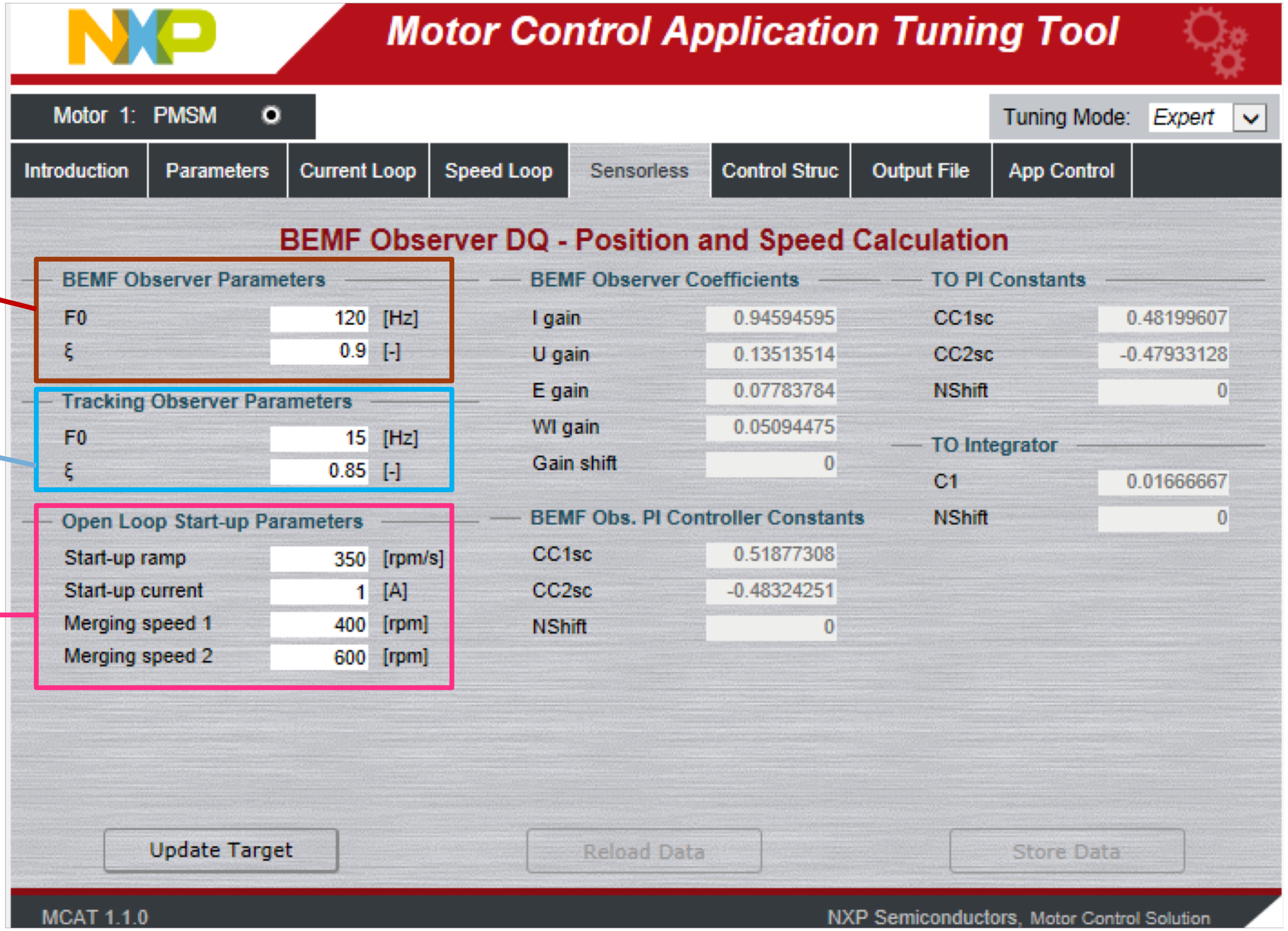
Step-5

Usually, BEMF Observer F0 should be the same as current loop F0

Tracking Observer F0 typically range from 5Hz to 60Hz

Start up ramp and current is critical for robust start up

Merging speed 1 is the ON/OFF for BEMF observer calculation
Merging speed 2 is the ON/OFF for electrical angle used in system



Tuning Mode: Expert

BEMF Observer DQ - Position and Speed Calculation

BEMF Observer Parameters

F0	120	[Hz]
ξ	0.9	[-]

Tracking Observer Parameters

F0	15	[Hz]
ξ	0.85	[-]

Open Loop Start-up Parameters

Start-up ramp	350	[rpm/s]
Start-up current	1	[A]
Merging speed 1	400	[rpm]
Merging speed 2	600	[rpm]

BEMF Observer Coefficients

I gain	0.94594595
U gain	0.13513514
E gain	0.07783784
WI gain	0.05094475
Gain shift	0

BEMF Obs. PI Controller Constants

CC1sc	0.51877308
CC2sc	-0.48324251
NShift	0

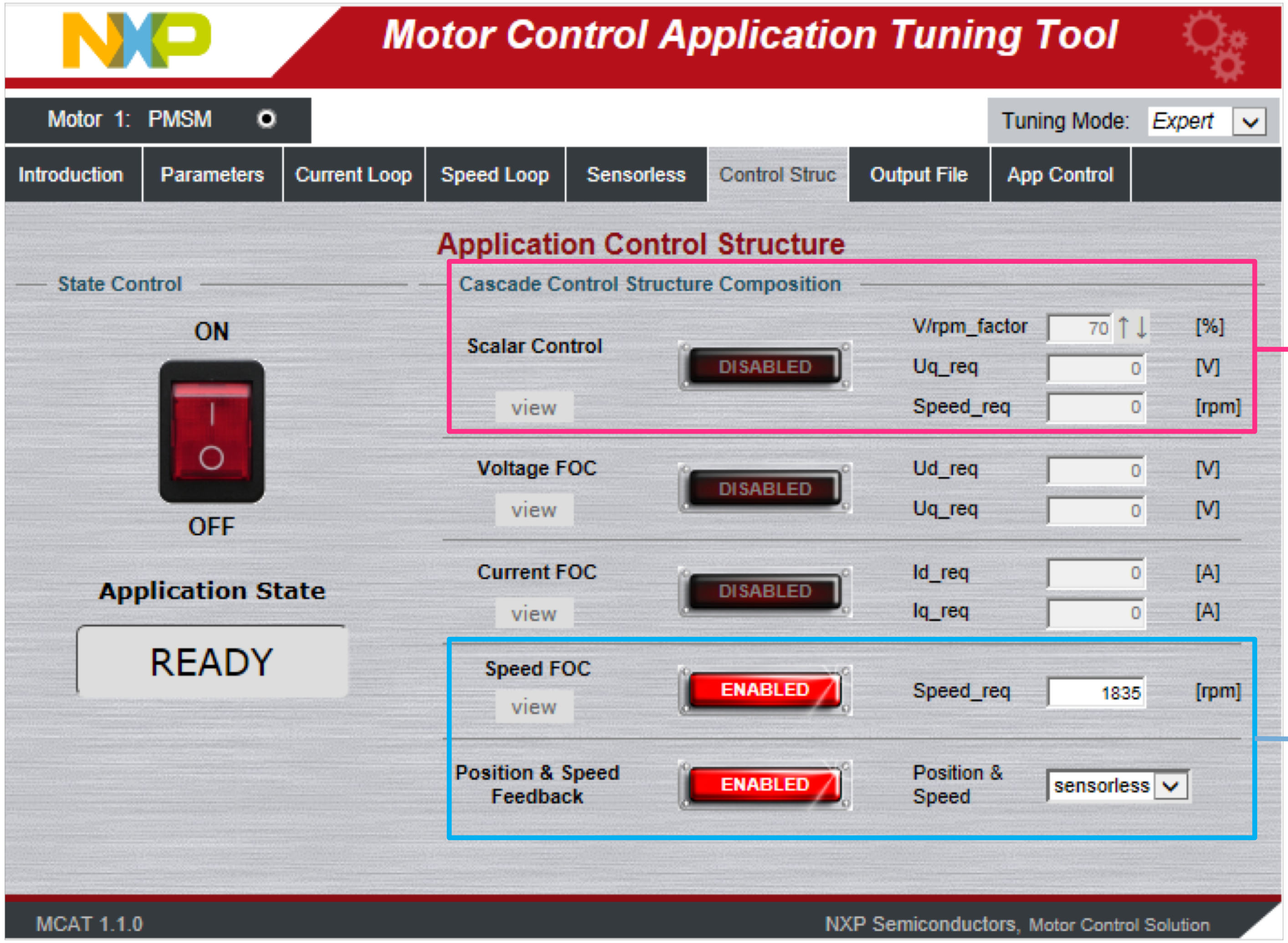
TO PI Constants

CC1sc	0.48199607
CC2sc	-0.47933128
NShift	0

TO Integrator

C1	0.01666667
NShift	0

Step-6



Motor 1: PMSM

Tuning Mode: Expert

Application Control Structure

State Control

Application State: READY

Cascade Control Structure Composition

Scalar Control: DISABLED
V/rpm_factor: 70 [%]
Uq_req: 0 [V]
Speed_req: 0 [rpm]

Voltage FOC: DISABLED
Ud_req: 0 [V]
Uq_req: 0 [V]

Current FOC: DISABLED
Id_req: 0 [A]
Iq_req: 0 [A]

Speed FOC: ENABLED
Speed_req: 1835 [rpm]

Position & Speed Feedback: ENABLED
Position & Speed: sensorless

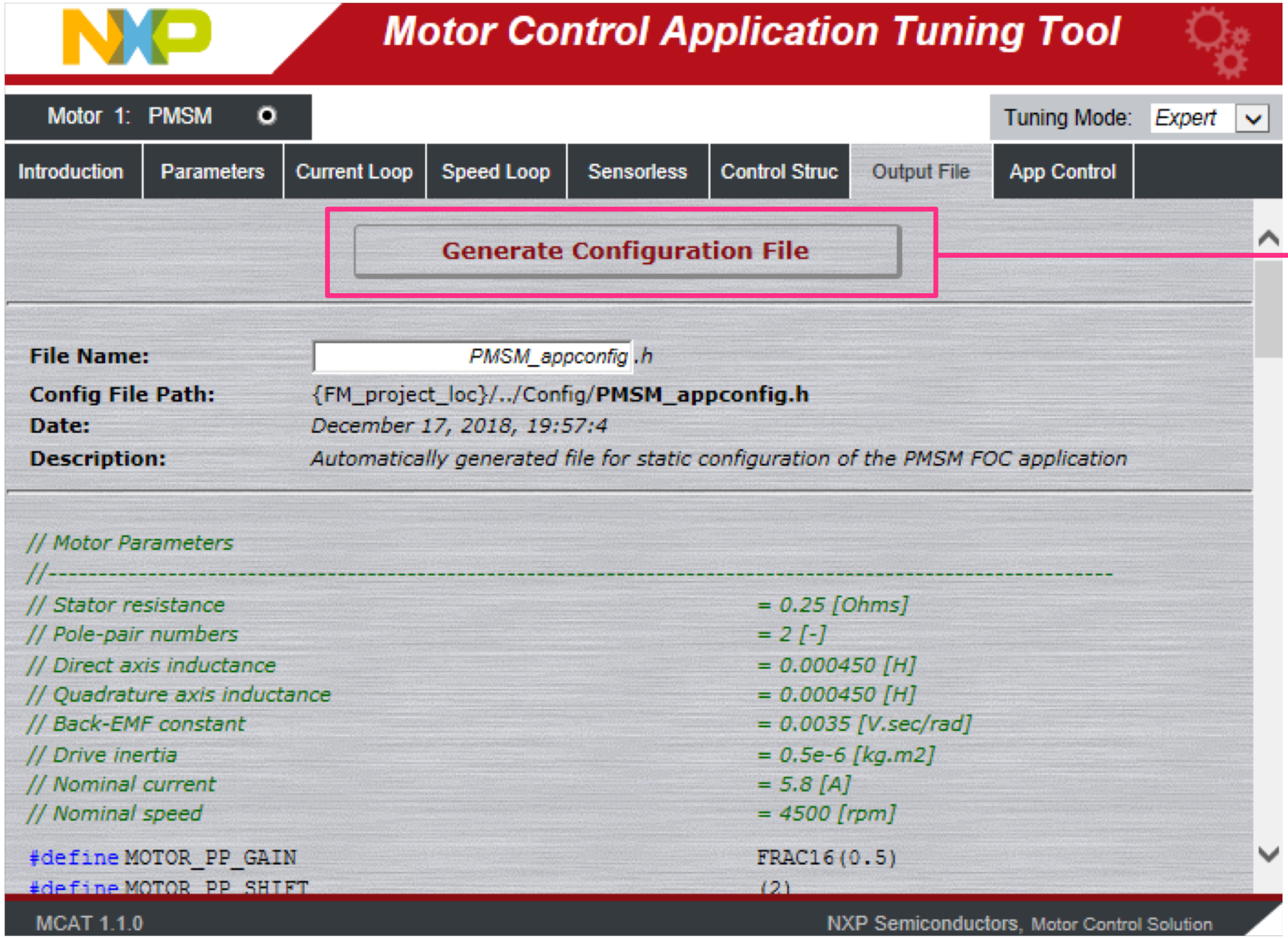
MCAT 1.1.0

NXP Semiconductors, Motor Control Solution

VVVF control, for hardware verification

Normal mode for speed control
Also is the default mode

Step-7

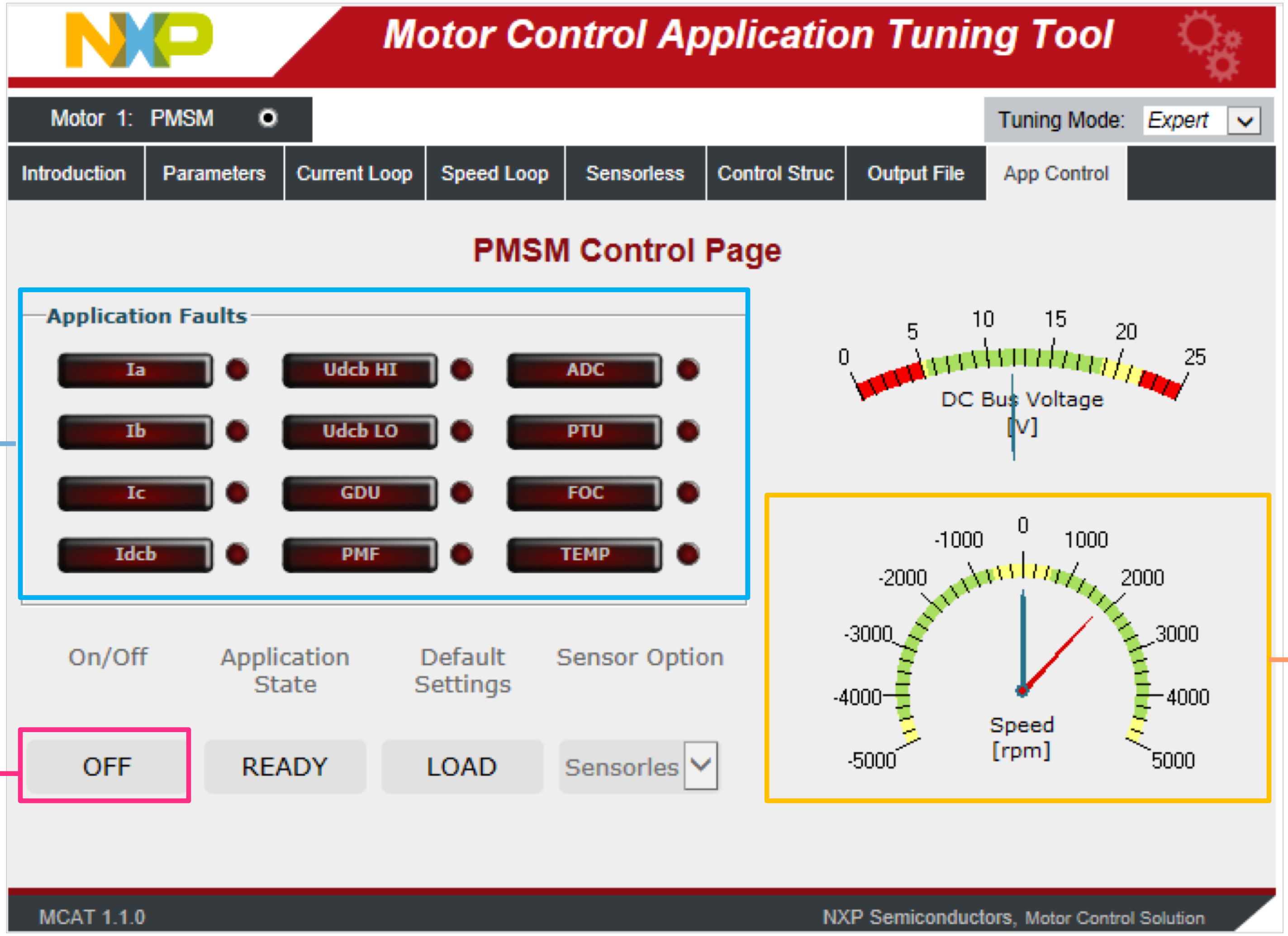


File Name: PMSM_appconfig .h
Config File Path: {FM_project_loc}/../Config/**PMSM_appconfig.h**
Date: December 17, 2018, 19:57:4
Description: Automatically generated file for static configuration of the PMSM FOC application

```
// Motor Parameters
//----------------------------------------------------------------------------------------------------
// Stator resistance                                      = 0.25 [Ohms]
// Pole-pair numbers                                      = 2 [-]
// Direct axis inductance                                 = 0.000450 [H]
// Quadrature axis inductance                             = 0.000450 [H]
// Back-EMF constant                                      = 0.0035 [V.sec/rad]
// Drive inertia                                          = 0.5e-6 [kg.m2]
// Nominal current                                        = 5.8 [A]
// Nominal speed                                          = 4500 [rpm]

#define MOTOR_PP_GAIN                                     FRAC16(0.5)
#define MOTOR_PP_SHIFT                                    (2)
```

After complete configure the parameters
Click the “generate configuration file”
And it will update “PMSM_appconfig.h” file

Step-8



Motor 1: PMSM

Tuning Mode: Expert

Indicate faults

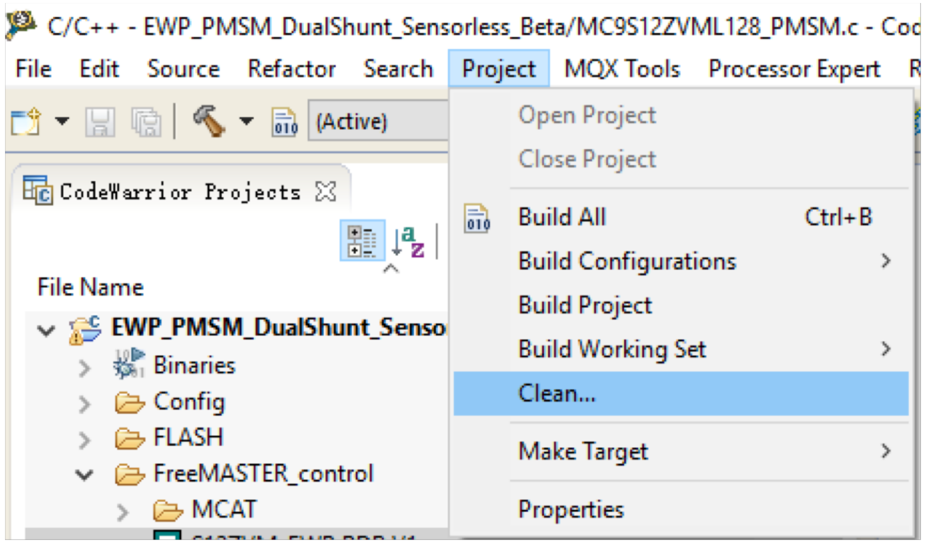
ON/OFF toggle control

Set the target speed

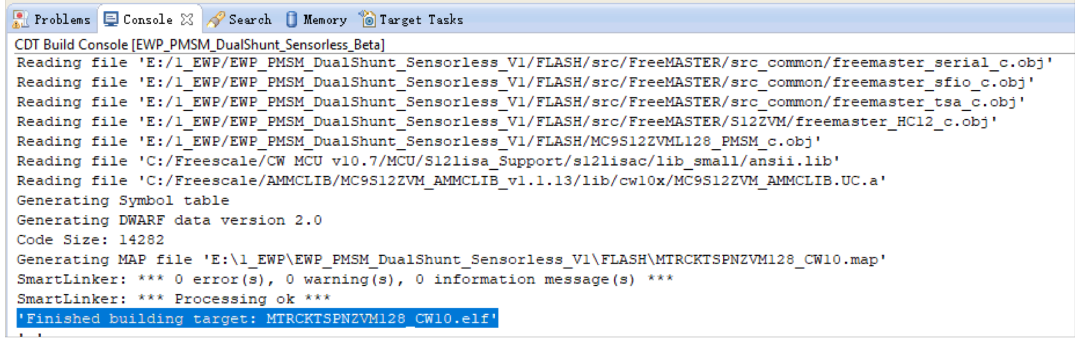
BUILD AND DEBUG PROJECTS

Build a Project

- After update the “PMSM_appconfig.h”, it need to rebuild the project. “Clean” the project will trigger rebuild project automatically. First open the project, need to do “clean” operation
- If don’t want to rebuild, just click the “build project”



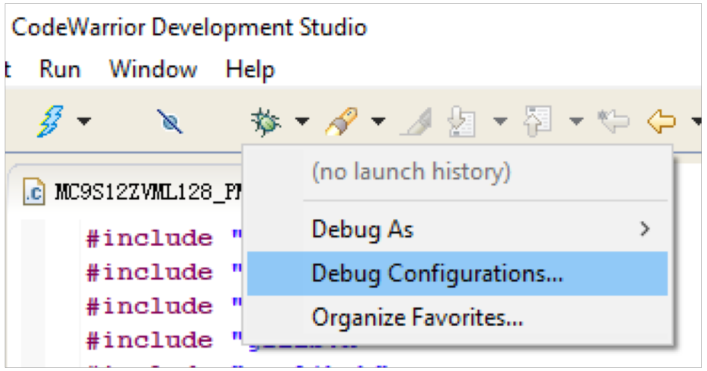
- If project is built successfully, following message will be displayed on the Console window



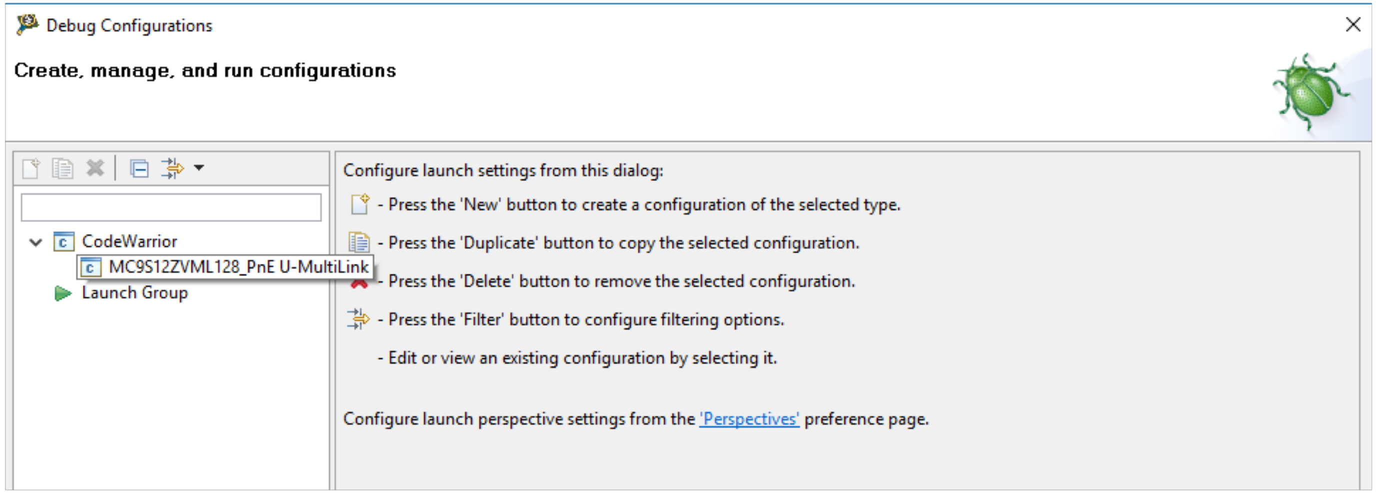
```
Reading file 'E:/1_EWP/EWP_PMSM_DualShunt_Sensorless_V1/FLASH/src/FreeMASTER/src_common/freemaster_serial_c.obj'
Reading file 'E:/1_EWP/EWP_PMSM_DualShunt_Sensorless_V1/FLASH/src/FreeMASTER/src_common/freemaster_sfio_c.obj'
Reading file 'E:/1_EWP/EWP_PMSM_DualShunt_Sensorless_V1/FLASH/src/FreeMASTER/src_common/freemaster_tsa_c.obj'
Reading file 'E:/1_EWP/EWP_PMSM_DualShunt_Sensorless_V1/FLASH/src/FreeMASTER/S12ZVM/freemaster_HC12_c.obj'
Reading file 'E:/1_EWP/EWP_PMSM_DualShunt_Sensorless_V1/FLASH/MC9S12ZVML128_PMSM_c.obj'
Reading file 'C:/Freescale/CW MCU v10.7/MCU/S12lisa_Support/s12lisac/lib_small/ansii.lib'
Reading file 'C:/Freescale/AMMCLIB/MC9S12ZVM_AMMCLIB_v1.1.13/lib/cw10x/MC9S12ZVM_AMMCLIB.UC.a'
Generating Symbol table
Generating DWARF data version 2.0
Code Size: 14282
Generating MAP file 'E:\1_EWP\EWP_PMSM_DualShunt_Sensorless_V1\FLASH\MTRCKTSPNZVM128_CW10.map'
SmartLinker: *** 0 error(s), 0 warning(s), 0 information message(s) ***
SmartLinker: *** Processing ok ***
'Finished building target: MTRCKTSPNZVM128_CW10.elf'
```

Debug a Project

Select “Debug Configurations”



Double Click “MC9S12ZVML128_PnE U…”



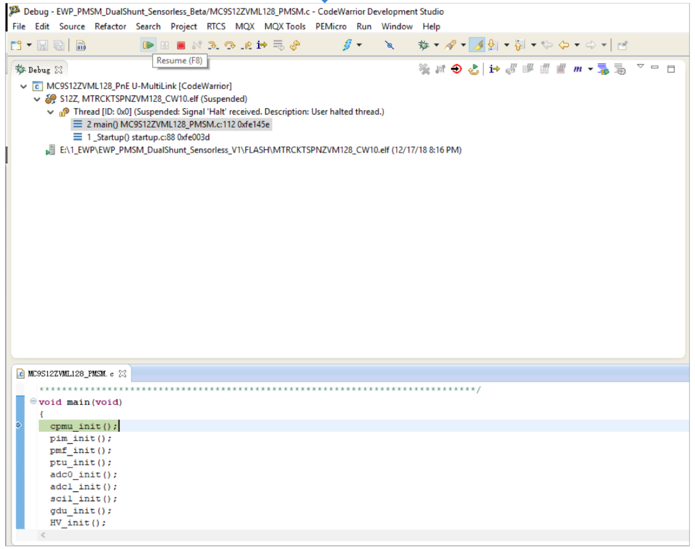
Create, manage, and run configurations

Configure launch settings from this dialog:

- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

Enjoy the debug now


Debug - EWP_PMSM_DualShunt_Sensorless_Beta/MC9S12ZVML128_PMSM.c - CodeWarrior Development Studio
File Edit Source Refactor Search Project RTCS MQX MQX Tools PEMicro Run Window Help
Resume (F8)
Debug
MC9S12ZVML128_PMSM.c

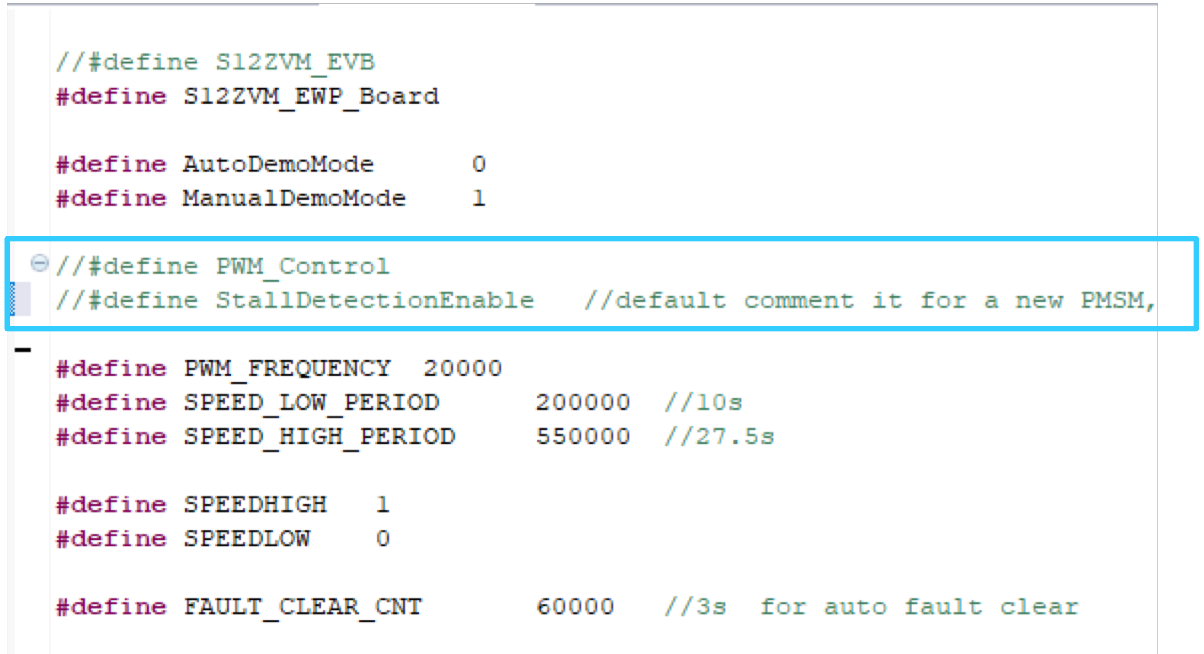
```
MC9S12ZVML128_PnE U-MultiLink [CodeWarrior]
  S12Z, MTRCKTSPNZVM128_CW10.elf (Suspended)
    Thread [ID: 0x0] (Suspended: Signal 'Halt' received. Description: User halted thread.)
      2 main() MC9S12ZVML128_PMSM.c:112 0xfe145e
      1 _Startup() startup.c:88 0xfe003d
  E:\1_EWP\EWP_PMSM_DualShunt_Sensorless_V1\FLASH\MTRCKTSPNZVM128_CW10.elf (12/17/18 8:16 PM)
```

```
*********************************************************************/
void main(void)
{
  cpmu_init();
  pim_init();
  pmf_init();
  ptu_init();
  adc0_init();
  adcl_init();
  scil_init();
  gdu_init();
  HV_init();
```

FREEMASTER TUNING

Step-1

- Make sure in UserDef.h file, “StallDetectionEnable” and “PWM_Control” are not enabled.



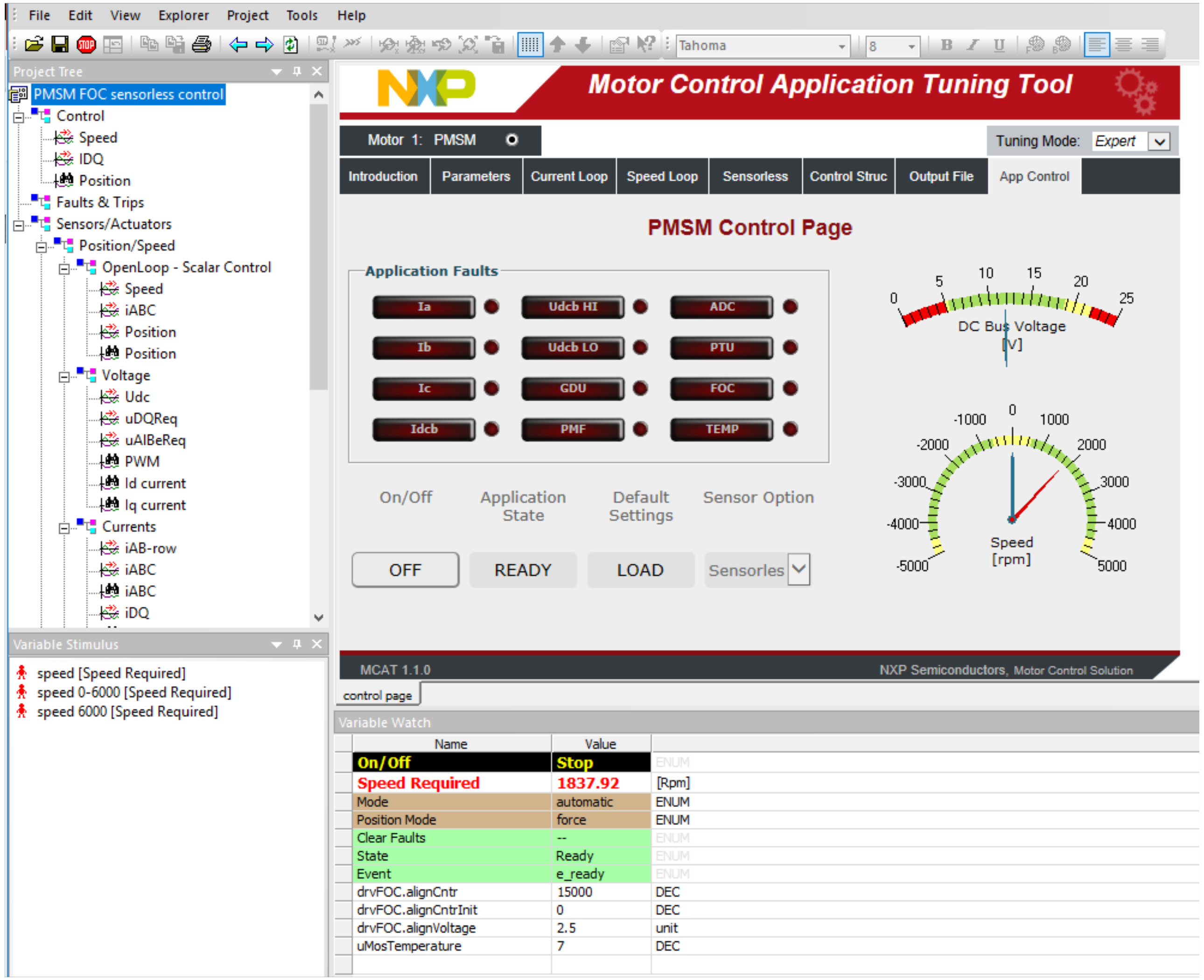
```
//#define S12ZVM_EVB
#define S12ZVM_EWP_Board

#define AutoDemoMode      0
#define ManualDemoMode    1

//#define PWM_Control
//#define StallDetectionEnable   //default comment it for a new PMSM,

#define PWM_FREQUENCY  20000
#define SPEED_LOW_PERIOD      200000  //10s
#define SPEED_HIGH_PERIOD     550000  //27.5s

#define SPEEDHIGH    1
#define SPEEDLOW     0

#define FAULT_CLEAR_CNT       60000   //3s  for auto fault clear
```

“Stall Detection” is related with specific motor parameters, it not included in MCAT this stage.
So it need tune by using your Motor.

PWM_Control is using PWM duty to control the PMSM ON/OFF and speed control

Step-1

- Set the target speed, for example, 1837rpm, and then click “OFF” and it will become “ON”
- Your motor should run smoothly to the target speed



speed [Speed Required]
speed 0-6000 [Speed Required]
speed 6000 [Speed Required]

Name	Value	
On/Off	Stop	ENUM
Speed Required	1837.92	[Rpm]
Mode	automatic	ENUM
Position Mode	force	ENUM
Clear Faults	--	ENUM
State	Ready	ENUM
Event	e_ready	ENUM
drvFOC.alignCntr	15000	DEC
drvFOC.alignCntrInit	0	DEC
drvFOC.alignVoltage	2.5	unit
uMosTemperature	7	DEC

SECURE CONNECTIONS
FOR A SMARTER WORLD