User Manual

for MPC574XG MCAL Sample Application

Document Number: UMIAASR4.0R1.0.5
Rev. 1.5

Contents

Section number **Title** **Page**

Chapter 1
Revision History

Chapter 2
About this Manual

2.1 Acronyms and Definitions.......7

2.2 Reference List.......7

Chapter 3
Sample Application Example Description

3.1 The application software functionality.......9

3.2 The application software functionality with OS.......10

3.3 Description of the LEDs and Buttons functionality.......12

Chapter 4
Installation Steps

4.1 Hardware Installation.......15

4.2 Software Installation.......17

4.2.1 Tresos Project Installation.......18

4.2.2 MCAL Application Configuration.......20

Chapter 5
Building the Sample Application Example

5.1 Building the Sample Application example.......23

5.2 Building with different compilers.......23

5.3 Building for different run-modes.......24

5.4 Clean Object and Linker Output Files.......25

5.5 Modifying the Configuration in Tresos Studio.......25

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	24/04/2015	Alex Gauggel	RTM 1.0.0
1.1	10/07/2015	Alex Gauggel	RTM 1.0.1
1.2	12/08/2016	Alin Meleandra	RTM 1.0.2
1.3	17/02/2017	Nicu Dobrostomat	RTM 1.0.3
1.4	25/08/2017	Alin Meleandra	RTM 1.0.4
1.5	14/12/2017	Stefan Tataru	RTM 1.0.5

Chapter 2
About this Manual

This User Manual describes utilization of the sample application for MPC574XG microcontroller with Autosar MCAL 4.0 Rev0003 version RTM 1.0.5.

2.1 Acronyms and Definitions

Table 2-1. Acronyms and Definitions

Abbreviation / Acronym	Description
DIO	Digital Input Output Driver
PORT	Port Driver
BSW	Basic Software
ADC	Analog Digital Converter
FEE	Flash EEPROM Emulation
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
ISR	Interrupt Service Routine
OS	Operating System
GUI	Graphical User Interface
API	Application Programming Interface
EcuM	ECU state Manager
WDG	Watchdog Driver
PLL	Phase Lock Loop
LED	Light Emitting Diode
PB Variant	Post Build Variant
LT Variant	Link Time Variant
PC Variant	Pre Compile Variant

2.2 Reference List

Table 2-2. Reference List

#	Items	Version
1	MPC5748G Microcontroller Reference Manual	Rev. 6, 10/2017
2	MPC5746C Microcontroller Reference Manual	Rev. 5, 10/2017

Chapter 3
Sample Application Example Description

This application demonstrates an example of usage for the MCAL modules. It is not part of the production code deliverables.

3.1 The application software functionality

Initializes MCU module

- Initializes PLL0 and configures it to 160MHz

Checks whether PLL is locked

Activates the PLL clock to the MCU clock distribution

Initializes the GPT driver. On GPT notification the LED D57 is toggled

Initializes WDG driver and configures it to OFF MODE

Initializes PORT module. Pins configuration is show in section Table 3-1. PORT and DIO Modules - Pin Configuration and DioChannel Assignment for keys and leds and in Table 3-3. PORT Configuration excluding Leds and Keys.

Initializes the CAN driver and Sample application specific data for this driver

Initializes the LIN driver and Sample application specific data for this driver

Initializes the SPI driver and Sample application specific data for this driver

Initialize the ADC driver and Sample application specific data for this driver

Initialize the PWM driver and Sample application specific data for this driver

Initialize the ICU driver and Sample application specific data for this driver

Initialize DIO driver. The DIO driver is used to toggle D57, D53 and retrieve the value of the key SW3.

Initialize FEE driver and Sample application specific data for this driver. The FLASH driver is used indirectly by FEE driver. After initialization:

- Erases memory block of 32 bytes
- Writes 32 bytes pattern
- Reads the memory block
- Verifies the retrieved values if matches the values that were previous written

Performs while loop

- The WDG is triggered
- If SW3 is pressed the LED D53 is set ON and when the timeout expires, the LED D53 is set OFF. If the timeout expires the Watchdog is configured again in SLOW MODE.
- ADC retrieves the trimmer value.
- The value got from the ADC conversion is used to set up the PWM duty cycle. The output of PWM is on the D52. Moving the trimmer has as effect modifying the intensity of D52.
- ICU is connected tthe output of PWM. It is used to measure the PWM duty cycle, and check the value against the value read from ADC
- If the value of the duty cycle retrieved by ICU matches the initial value read using ADC, the cycle restarts with the retrieving the ADC value.
- Performs a loopback transmission (exploiting CAN's hardware loopback capabilities) on the CAN unit 1
- DSPI unit 1 performs a simple SPI transmission in loopback mode. In this case the SOUT and SIN pins must be tied together in order to have the electrical loopback
- Various messages are sent over the UART.

Note

Connect the serial terminal to J19 connector on the motherboard. The serial terminal should have the following settings: 115200, 8, N, 1

3.2 The application software functionality with OS

Note: For the below scenarios the NXP AUTOSAR OS/MPC574xG v4.0 RTM v1.0.1 (build 4.0.86) was used.

Initializes WDG driver and configures it to OFF MODE

Initializes MCU module

- Initializes PLL0 and configures it to 160MHz
- External oscillator frequency is 40MHz

Checks whether PLL is locked

Activates the PLL clock to the MCU clock distribution

Initializes PORT module. Pins configuration is show in section PORT and DIO Modules - Pin Configuration and DioChannel Assignment for keys and leds, and in PORT Configuration excluding Leds and Keys

Initializes the CAN driver and Sample application specific data for this driver

Initializes the SPI driver and Sample application specific data for this driver

Initializes the PWM driver and Sample application specific data for this driver

Initialize the ADC driver and Sample application specific data for this driver

Initialize DIO driver. The DIO driver is used to toggle D57, D53 and retrieve the value of the key SW3.

Initializes the GPT driver. On GPT notification the LED D57 is toggled

After initialization the following 4 Tasks are automatically started.

Task1 will perform the following actions using Fee driver:

- Erases memory block of 32 bytes
- Writes 32 bytes pattern
- Reads the memory block
- Verifies the retrieved values if matches the values that were previous written
- After the end of the Fee driver actions the Gpt timer is started and Wdg is set to SLOW MODE. The Task1 will handle reading of the key SW3 button. If key SW3 is pressed the LED 2 is set ON and when the Wdg timeout expires, the LED2 is set OFF. If the timeout expires the Watchdog is configured again in SLOW MODE.

Task2 will perform the following actions using SPI, LIN and CAN drivers:

- DSPI unit 0 performs a simple SPI transmission in loopback mode. In this case the SOUT and SIN pins must be tied together in order to have the electrical loopback
- LinFlex unit 0 performs a simple LIN transmission in loopback mode
- Performs a loopback transmission (exploiting CAN's hardware loopback capabilities) on the FlexCAN unit 0

Task3 will perform the following actions using ADC, PWM and ICU drivers:

- ADC retrieves the trimmer value
- The value got from the ADC conversion is used to set up the PWM duty cycle. The output of PWM is on the D52. Moving the trimmer has as effect modifying the intensity of D52.
- ICU is connected to the output of PWM. It is used to measure the PWM duty cycle, and check the value against the value read from ADC
- If the value of the duty cycle retrieved by ICU matches the initial value read using ADC, the cycle restarts with the retrieving the ADC value

LoopTask will have the lowest priority and will call periodically the MainFunctions of the FLS and FEE drivers.

Various messages are sent over the UART.

Note

Connect the serial terminal RS232 connector on the motherboard. The serial terminal should have the following settings: 115200, 8, N, 1

3.3 Description of the LEDs and Buttons functionality

The detailed description of the LEDs and Buttons functionality is depicted in the following table:

Table 3-1. PORT and DIO Modules - Pin Configuration and DioChannel Assignment

PortPin Name	Pin ID (PCR ID)	Pin Mode	Pin Direction	Pin Level	Connected HW	Channel Assignment
Led1_Pwm	14	EMIOS0_E0UC_0_X_OUT	Out	Low	LED D52	

Table continues on the next page...

Table 3-1. PORT and DIO Modules - Pin Configuration and DioChannel Assignment (continued)

PortPin Name	Pin ID (PCR ID)	Pin Mode	Pin Direction	Pin Level	Connected HW	Channel Assignment
Led2	99	GPIO	Out	High	LED D53	Dio_Led2
Led3	100	GPIO	Out	Low	LED D57	Dio_Led3
Led4	101	GPIO	Out	Low	LED D58	-
Key1	1	GPIO	In	-	SW3	Dio_Key1
Adc_Pot	20	ADC_1_ADC1_P_0	-	-	Trimmer	-

Table 3-2. LEDs and Buttons Functionality

LEDs and Buttons	Functionality	LED ON	LED OFF
Led D52	Indicates the output for PWM - the intensity varies depending on the duty cycle of PWM. The duty cycles is modified using the trimmer.	-	-
Led D53	When SW3 is pressed the led turns on, until the watchdog timeout	When SW3 pressed and watchdog timeout did not expired	If SW3 is not pressed or the watchdog timeout expired
Led D57	Is toggled at each GPT notification once every second	-	-
Led D58	Set up using port to be ON	-	-
SW3	Disables the watchdog triggering	-	-

Table 3-3. PORT Configuration excluding Leds and Keys

PortPin Name	Port	Pin ID (PCR ID)	Pin Mode
Icu_measurement	PA[15]	15	EMIOS0_E-UC_1_G_IN
DSPI0_OUT	PA[13]	13	DSPI_0_dSOUT
DSPI0_IN	PA[12]	12	DSPI_0_dSIN
LIN1_RXD	PC[7]	39	LIN_1_LIN1RX
LIN1_TXD	PC[6]	38	LIN_1_LIN1TX
UART_LIN2_TXD	PC[8]	40	LIN_2_LIN2TX
UART_LIN2_RXD	PC[9]	41	LIN_2_LIN2RX
DCI_TCK	PH[9]	121	DCI_TCK
DCI_TDI	PC[0]	32	DCI_TDI
DCI_TDO	PC[1]	33	DCI_TDO
DCI_TMS	PH[10]	122	DCI_TMS_IN_OUT

Chapter 4
Installation Steps

4.1 Hardware Installation

The MPC574XG-MB Rev.C evaluation board and jumper configuration are shown in Figure 4-1. MPC574XG-MB Rev.C Evaluation Board.

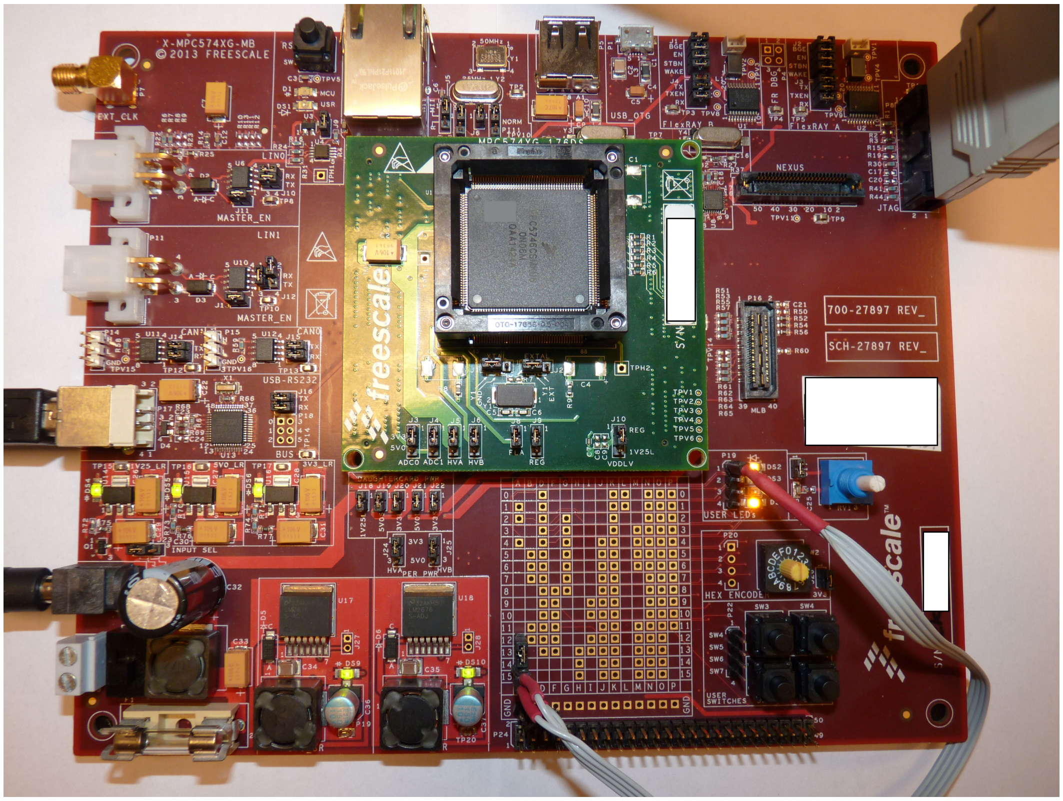


Figure 4-1. MPC574XG-MB Rev.C Evaluation Board

Please check whether the MPC574XG Mini-Module Board is populated with 40MHz oscillator (crystal Y1).In order to change the clock configuration please follow instruction in chapter Modifying the Configuration in Tresos Studio (especially the Figure 5-1: Modifying the External Crystal Frequency) to adjust the configuration.

Check all jumper configurations as shown in Figure 4-1. MPC574XG-MB Rev.C Evaluation Board and Figure 4-2. Buttons and LEDs Jumper Configuration

- J16 RX, J16 TX closed – needed for UART console support

Check all connections to be as follows:

- A12 with A13 needed for SPI loopback
- J13 closed, J12.1 with J12.3, J12.2 and J12.4 open for LIN loopback
- J17 - closed – needed for connection ADC trimmer
- A14 with A15 with P19 - 1 – needed for Pwm - Led D52 - Icu connection

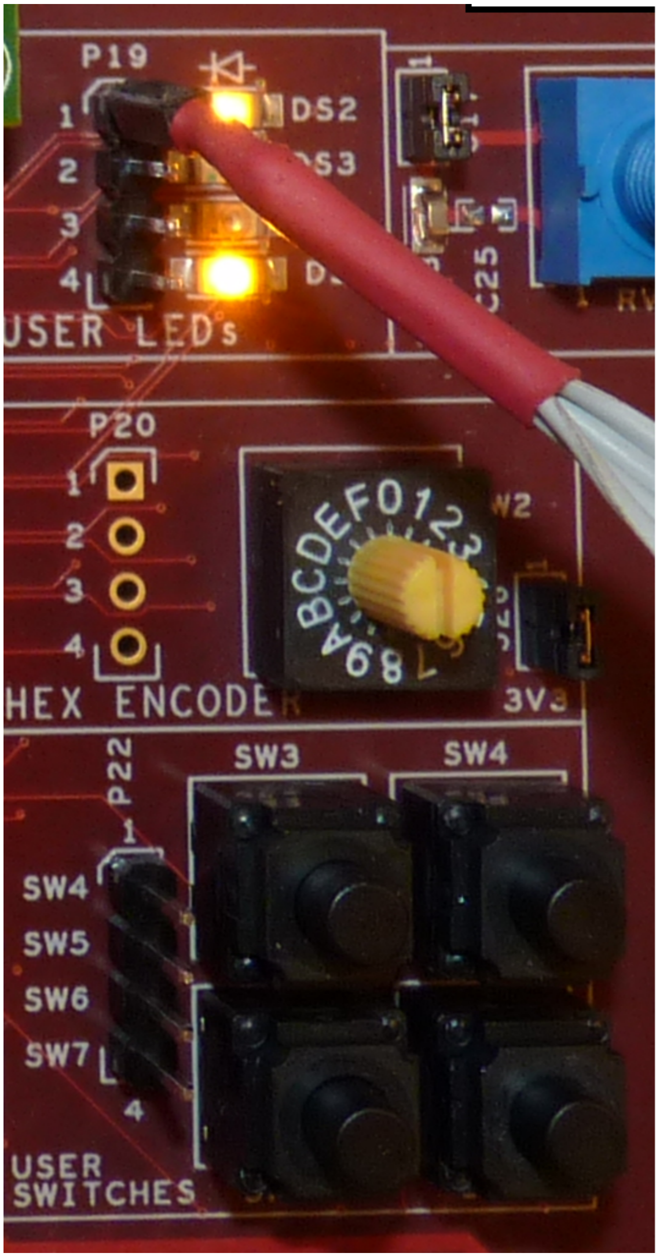


Figure 4-2. Buttons and LEDs Jumper Configuration

4.2 Software Installation

Please install the MCAL package on your computer. The package includes the MCAL Sample Application example with the following folder structure:

Folder or file	Description
- bin folder	generated object files and linker output files are stored into this folder

Table continues on the next page...

Folder or file	Description
- cfg folder	contains configuration files generated by Tresos tool
- include subfolder	contains files with pre-compile configurations
- src subfolder	contains files with post-build and link-time configurations
- doc folder	contains documentation
- include folder	contains header files and types definition
- make folder	makefiles used for building the application
- src folder	contains the application source code file
- toolchains folder	files needed to build with various toolchains (startup, linker command files)
- makefile file	the MCAL sample application makefile
- makefile_os file	the MCAL sample application makefile for integration with OS
- Modules file	specifies which modules are compiled and linked
- make.bat file	launches the make command
- launch.bat file	contains path to the Tresos Studio installation and launches the make.bat file
- Tresos folder/workspace	contains the Tresos project with the application configuration

4.2.1 Tresos Project Installation

The following procedure requires that the user has EB Tresos Studio installed.

Procedure:

1. Make sure that all MCAL plugins are already installed in the Tresos Studio plugins directory

2. Open Tresos Studio

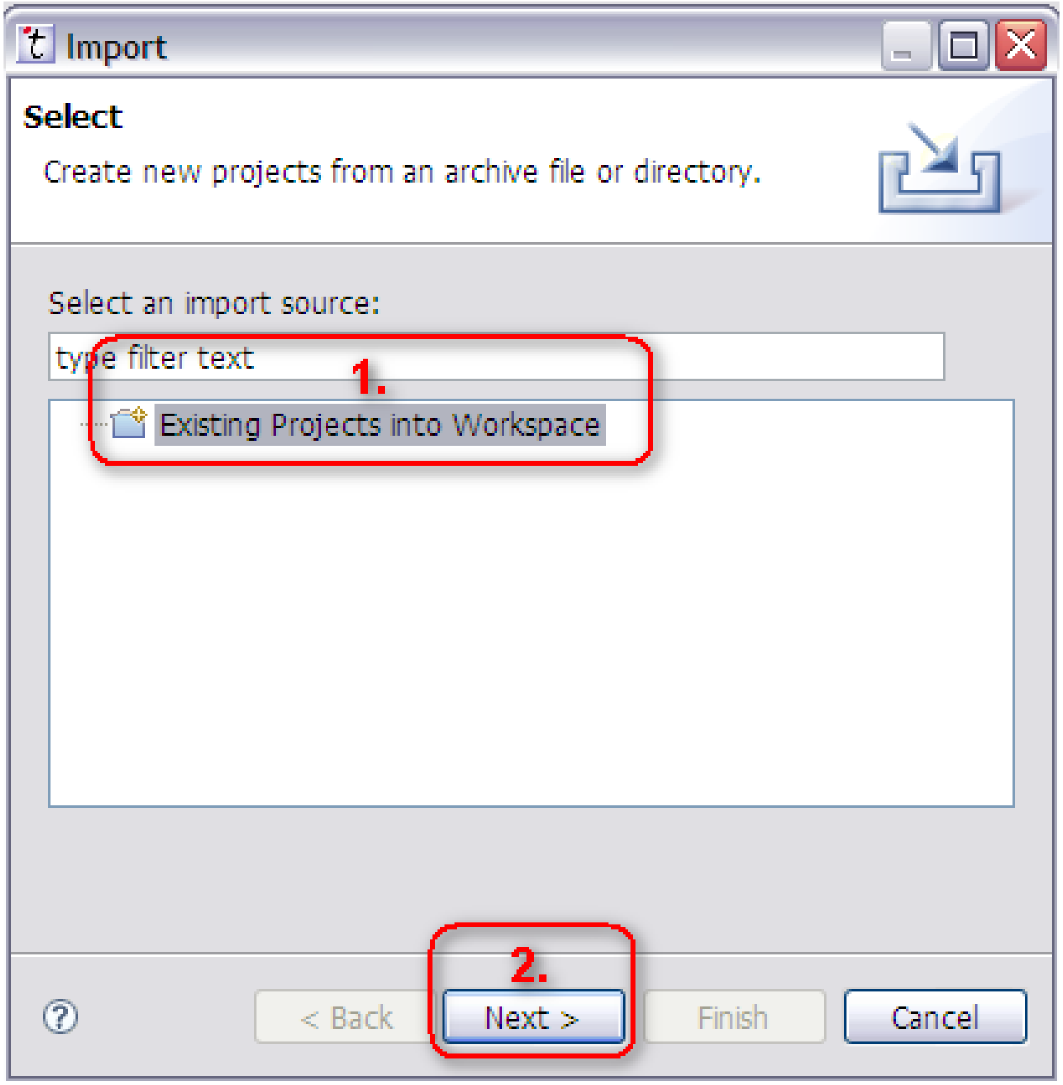
3. Import Sample application project

a. Click on "File" and select "Import"

b. Select "Existing Projects into Workspace" and click on "Next" button as shown in Figure 4-3. Import Window - the First View

c. Next steps are depicted in Figure 4-4. Import Window - the Second View

- Select "Select root directory" and click on "Browse"
- Select the location of the [project] folder in the installed Sample application package folder (Tresos/workspace/[project])
- Select "Copy projects into workspace"
- Click on "Finish" button



Import

Select

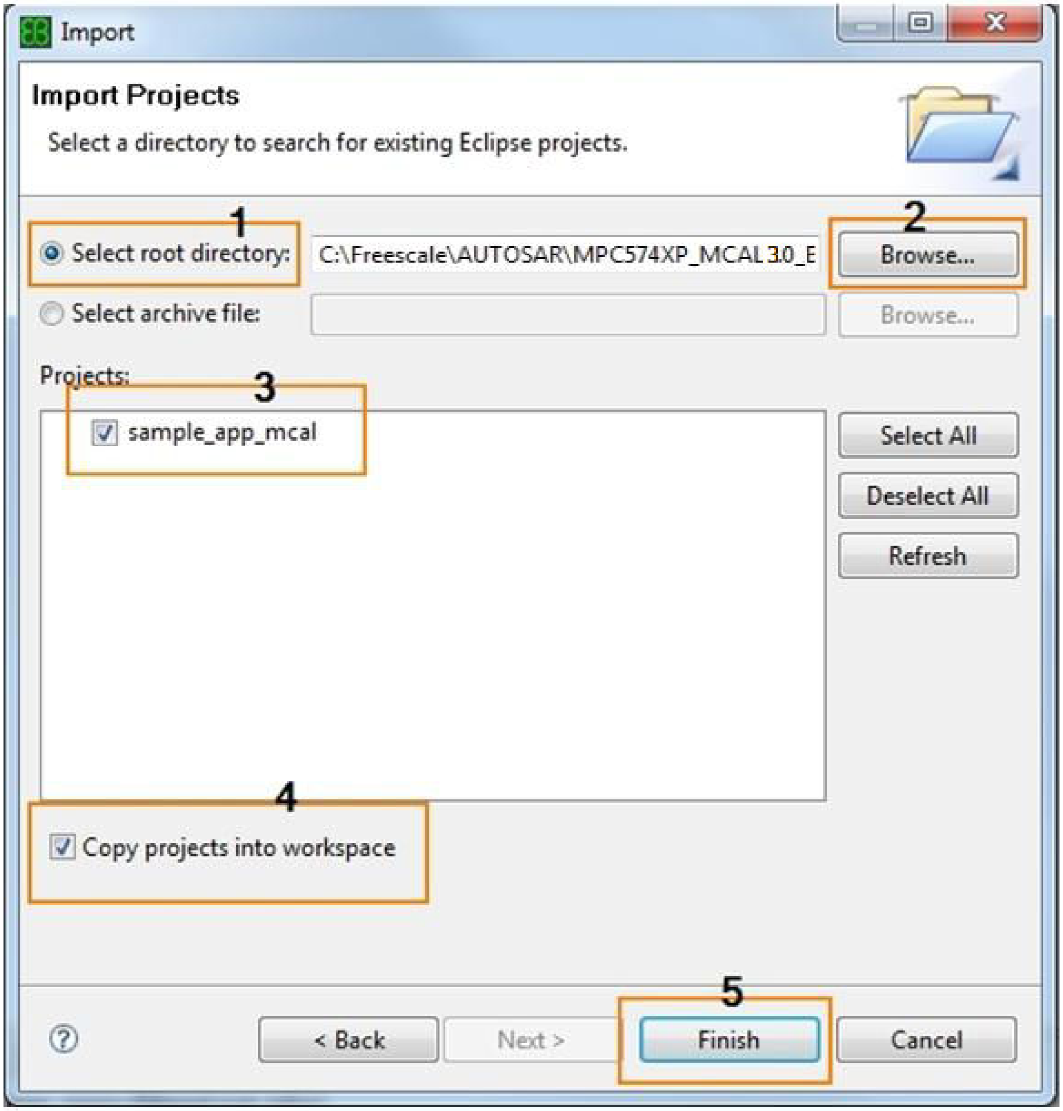
Create new projects from an archive file or directory.

Select an import source:

type filter text

Existing Projects into Workspace

Figure 4-3. Import Window - the First View



Import

Import Projects

Select a directory to search for existing Eclipse projects.

1 ● Select root directory: C:\Freescale\AUTOSAR\MPC574XP_MCAL3.0_E

○ Select archive file:

Projects:

3 ☑ sample_app_mcal

4 ☑ Copy projects into workspace

Figure 4-4. Import Window - the Second View

Note

A better way to configure the Sample application project is to create a New Configuration Porject and to import the configuration XDM of each driver. This method will avoid any issues which may be caused by different Tresos paths.

4.2.2 MCAL Application Configuration

The following procedure requires that the user has EB Tresos Studio installed and the toolchains versions specified in the MCAL Release Notes.

The toolchain that will be used needs to be installed for correct operation and the path to the installation location shall be added into the system environment variable(s):

- TRESOS_DIR for Electrobit Tresos

o Ex: SET TRESOS_DIR=C:/Tools/EB/tresos/v14.2.1

- GHS_DIR for GreenHills Multi

o Ex: SET GHS_DIR=C:/ghs/comp_201416

- DIAB_DIR for Windriver DIAB

o Ex: SET DIAB_DIR=C:/WindRiver/diab/5.9.4.8

Note

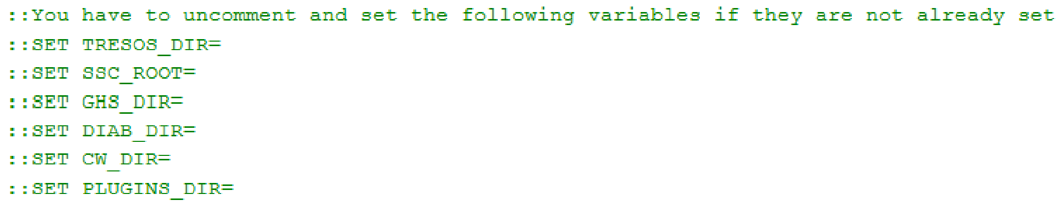
The path to the toolchain must not contain spaces. In case the compiler is installed into a path with spaces, the variable must be set with the "short" folder name (8.3 version of the file name that can be displayed with dir /X in command prompt)

Procedure:

1. Open launch.bat file in a text editor and specify the EB Tresos Studio location in the TRESOS_DIR parameter as shown in Figure 4-5. Configuration of the Tresos Studio Location

2. Make sure that installation location of the compiler is added in the system environment variable (GHS_DIR or DIAB_DIR)

3. Setup the plugins folder location if the plugins are not installed in the Tresos plugins folder



```
::You have to uncomment and set the following variables if they are not already set
::SET TRESOS_DIR=
::SET SSC_ROOT=
::SET GHS_DIR=
::SET DIAB_DIR=
::SET CW_DIR=
::SET PLUGINS_DIR=
```

Figure 4-5. Configuration of the Tresos Studio Location and the plugins folder location

Note

For running this Sample application together with AUTOSAR OS, the correct version of the OS must be installed separately

Chapter 5
Building the Sample Application Example

This section describes the build procedure.

5.1 Building the Sample Application example

Procedure:

1. Open the Windows command prompt window

2. Change the current directory to the Sample application folder

3. To build the sample, execute the following command to run launch.bat: launch.bat

4. The object files and linker output file (sample_app_mcal.elf) shall be generated in the /bin subdirectory

5. To execute the sample application load the executable file placed in the /bin subdirectory to the evaluation board using the Lauterbach debugger and run.cmm or run_ram.cmm script (or the MPC5500 Flash Utility).

Note

The launch.bat file calls the make.bat file and then the GNU make utility is called from the Tresos Studio bin directory.

5.2 Building with different compilers

To build the sample application with a different compiler, use the following parameter for the launch command:

launch.bat TOOLCHAIN=[toolchain]

where [toolchain] can have the values:

* ghs - default - use the GreenHills Multi compiler

* diab - use the Windriver DIAB compiler

To build the sample application to run with the AUTOSAR OS

launch.bat OS TOOLCHAIN=[toolchain]

where [toolchain] can have the values:

* ghs - default - use the GreenHills Multi compiler

* diab - use the Windriver DIAB compiler

Note

This Sample Application can only run with AUTOSAR OS using SC1 (scalability class 1) which requires drives to be configured in SUPERVISOR mode only. For running in User Mode a different OS configuration is required and a scalability class greater or equal to SC3.

Note

The TOOLCHAIN parameter is optional

5.3 Building for different run-modes

To build the sample application for a different run-mode, use the following parameter for the launch command:

launch.bat MODE=[user_mode] TOOLCHAIN=[toolchain]

where [user_mode] can have the values:

* SUPR - default - run in Supervisor mode

* USER - run in User mode

where [toolchain] can have the values:

* GHS - default - compile with GreenHills

* DIAB - compile with WindRiver Diab

Note

In order to run in USER mode, all drivers that need to be executed in this mode should have the "Enable User Mode Support" parameter set to 'true' and their configuration files regenerated from Tresos.

Note

This Sample Application can only run with AUTOSAR OS using SC1 (scalability class 1) which requires drives to be configured in SUPERVISOR mode only. For running in User Mode a different OS configuration is required and a scalability class greater or equal to SC3.

Note

The TOOLCHAIN parameter is optional

5.4 Clean Object and Linker Output Files

To clean the object and linker output files from the folder /bin, execute the following steps

Procedure:

1. Open the Windows command prompt window

2. Change the current directory to Sample application folder

3. Execute the following command launch.bat clean

4. The object files and linker output files shall be cleared from the /bin and from the Sample application root folders.

5.5 Modifying the Configuration in Tresos Studio

Users may change the application configuration according to their needs.

Procedure:

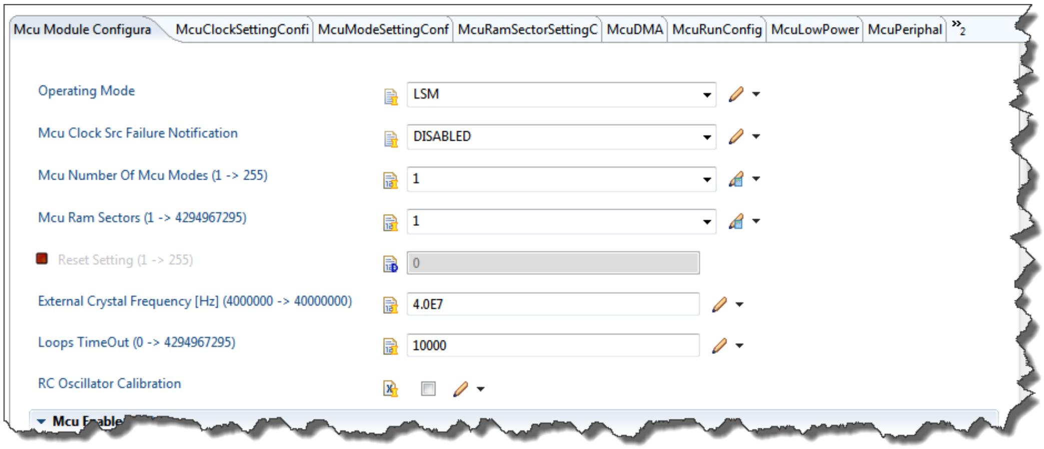
1. Open the EB Tresos Studio GUI

2. Open previously imported Sample Application project

3. Use the Tresos Studio GUI to modify configuration parameter values and save the changes. The value of the External Crystal Frequency parameter can be changed as depicted in Figure 5-1: Modifying the External Crystal Frequency

4. Select the Sample Application project and click on "Generate" button to generate the configuration files.

5. Copy the generated configuration files from workspace/[project]/output/include directory into the Sample Application folder /cfg/include.



Operating Mode: LSM

Mcu Clock Src Failure Notification: DISABLED

Mcu Number Of Mcu Modes (1 -> 255): 1

Mcu Ram Sectors (1 -> 4294967295): 1

Reset Setting (1 -> 255): 0

External Crystal Frequency [Hz] (4000000 -> 40000000): 4.0E7

Loops TimeOut (0 -> 4294967295): 10000

RC Oscillator Calibration: □

Figure 5-1. Modifying the External Crystal Frequency

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.