How to Migrate from MCEWizard / MCEDesigner to iMOTION™ Solution Designer

About this document

Scope and purpose

The iMOTION™ Solution Designer (iSD) is the next generation integrated development environment for iMOTION™ digital motor control solutions, with which users can configure and optimize motor and PFC parameters and perform motor / PFC tuning and validation work. The iSD is designed to replace the existing iMOTION™ development tools, namely the MCEWizard and MCEDesigner[1]. This document explains how to migrate from using the MCEWizard / MCEDesigner to using the new iSD tool.

Intended audience

This document targets the users who have used the MCEWizard / MCEDesigner and would like to migrate existing design projects created by the MCEWizard / MCEDesigner to the new iSD tool. For those users who intend to start a new design project directly from using the iSD, please refer to references [2][1] and [4].

Table of contents

About this document **1**

Table of contents **1**

1 Overview of iMOTION™ Solution Designer **3**

1.1 Introduction 3

1.2 Design Workflow Comparison 3

1.3 File System 6

1.3.1 generated 6

1.3.2 lib 7

1.3.3 Script 7

1.3.4 settings 8

2 Lower-level Parameter Comparison **9**

2.1 Overview 9

2.1.1 Register Categories in MCEDesigner and iMOTION™ Solution Designer 9

2.1.2 Register Nomenclature 9

2.2 Control Registers 10

2.2.1 How to Use Register IDs in Communication Commands 10

2.2.2 System Control Registers 11

2.2.3 Motor Control Registers 12

2.2.4 PFC Control Registers 18

2.2.5 Script Registers 18

3 State Machine Handling **19**

3.1 Motor State Machine 19

3.2 PFC State Machine 20

4 Migration Step Example **21**

4.1 New Project Generation 21
4.2 Motor & PFC Configuration by Configuration Wizard 21
4.2.1 IC Configuration 21
4.2.2 User Pin Configuration 22
4.2.3 System Hardware configuration 22
4.2.3.1 Input Supply 23
4.2.3.2 PFC Basic Input and Output 23
4.2.3.3 PFC Power Stage 23
4.2.3.4 PFC Current Sensing 24
4.2.3.5 PFC VAC Sensing 25
4.2.3.6 DC Bus Sensing Feedback 25
4.2.3.7 Inverter 25
4.2.3.8 Motor Current Sensing 26
4.2.3.9 Motor 27
4.2.3.10 Load 27
4.2.4 Motor Control Configuration 28
4.2.4.1 Motor Control Configuration 28
4.2.4.2 Control Regulators 28
4.2.4.3 FOC and Inverter 28
4.2.4.4 Application 29
4.2.4.5 Angle Feedback 30
4.2.5 PFC Control Configuration 30
4.2.6 Protection 30
4.2.6.1 PFC Protection 30
4.2.6.2 Motor Protection 30
4.2.6.3 System Protection 31
4.3 Script 31
4.4 Evaluation 31
5 References 32
Revision history 33

1 Overview of iMOTION™ Solution Designer

1.1 Introduction

The iMOTION™ Solution Designer (iSD) is an integrated development tool for iMOTION™ controller families. In the old development environment (MCEWizard and Designer, referred as MCE tools later), motor control configuration setting and motor evaluation with motor running are done by a different tool. When using the iSD, both configuration and evaluation can be done with one evaluation tool seamlessly. Some new devices (e.g. IMI101) are applicable only to the iSD, so users need to use the iSD when using new iMOTION™ devices.

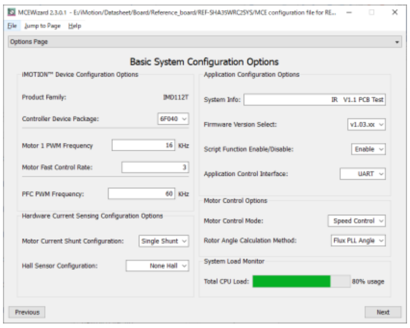
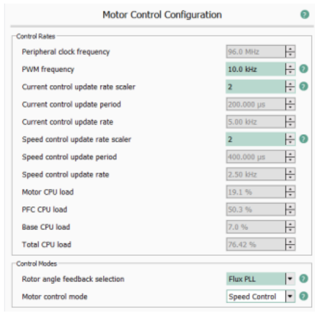
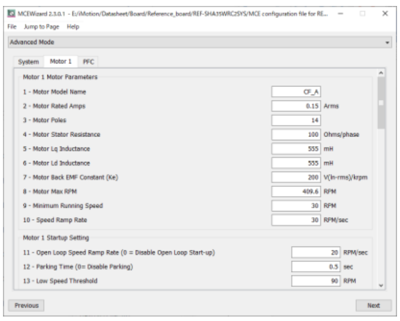
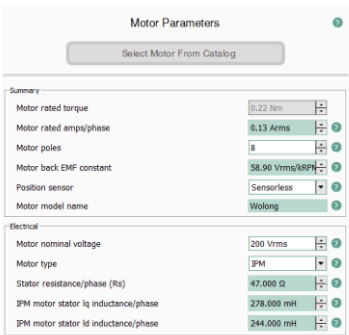
The outlook and function of the iSD varies greatly from the MCE tools. This section will cover the major differences between the iSD and MCE tools, and how to set configuration with the iSD. More detailed information for how to use the iSD is available in [4].

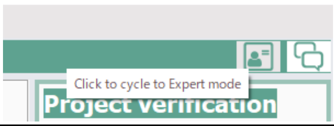
Please note that the iSD is available for firmware version 5.0 or later. Firmware version 1.3.x or before is only supported by the MCE tools.

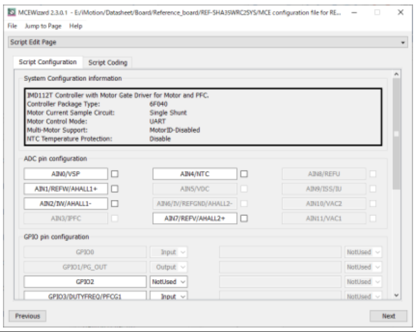
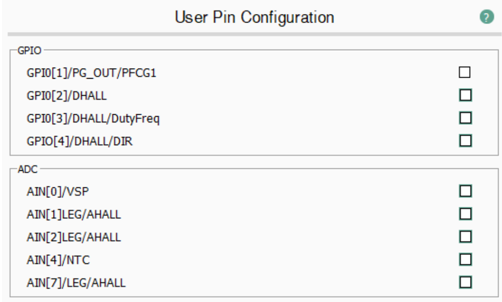
1.2 Design Workflow Comparison

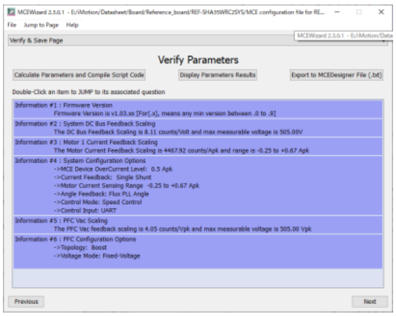
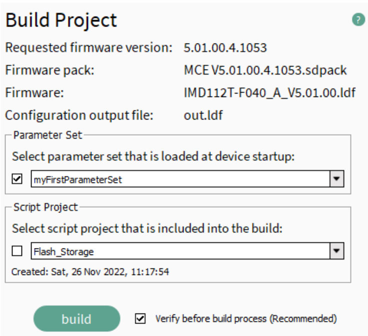
Table 1 and Table 2 show a comparison of the workflow between the MCE tools and iSD. The MCEWizard is used to configure the system and motor parameter, and the MCEDesigner is used to program the parameter file, firmware, and script. By using the iSD, all tasks can be done seamlessly.

Table 1 Workflow for Motor and System Configuration

<table>
<tr><th>MCEWizard (FW1.3.x)</th><th>iSD (FW5.x)</th></tr>
<tr><td>Basic System Configuration Options

</td><td>Configuration Wizard

</td></tr>
<tr><td>Input motor and system parameters (Questions or Advanced Mode)
</td><td>Configuration Wizard


It is possible to enter expert mode by clicking the icon at the top-right corner of the window.

</td></tr>
</table>

<table>
<tr><th>MCEWizard (FW1.3.x)</th><th>iSD (FW5.x)</th></tr>
<tr><td>Script pin configuration

</td><td>Configuration Wizard (User Pin Configuration)


</td></tr>
<tr><td>Make parameter file (Verify & Save Page)

</td><td>No need to make the parameter file, but project build is necessary instead.


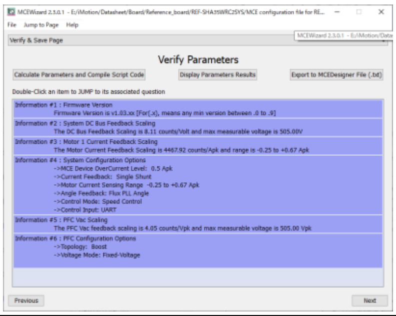
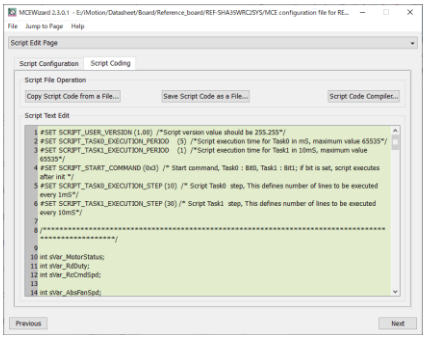
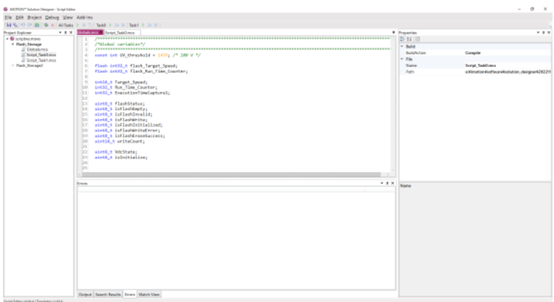
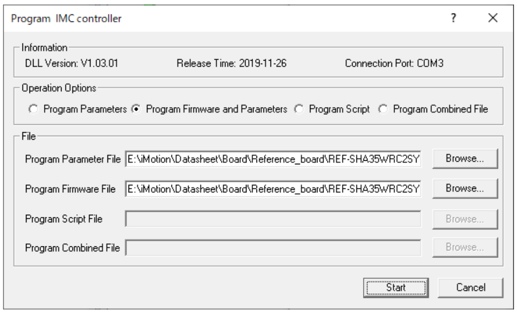
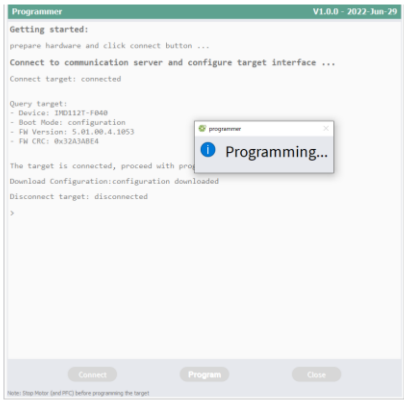
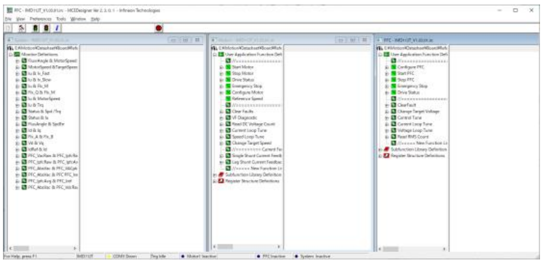
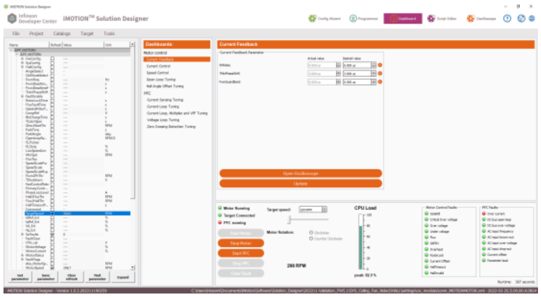
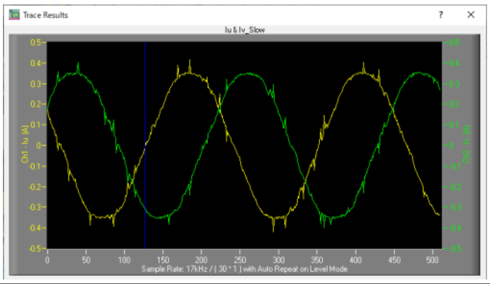
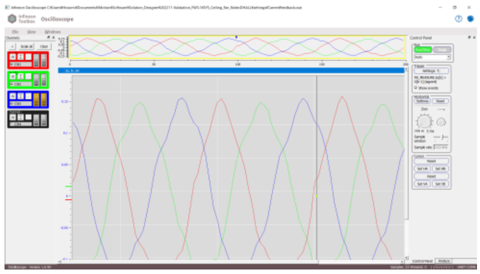
</td></tr>
<tr><td>Export parameter file

</td><td>No need to export the parameter file.</td></tr>
<tr><td>Script coding and compiling

</td><td>Script Editor
Debugging function can be used.

</td></tr>
</table>

Table 2 Workflow for Programming and Motor/System Evaluation

MCEDesigner	iSD
Programming parameter, firmware, and script 	Programmer Firmware is automatically selected by the iSD and the user does not have to choose it. 
Motor and PFC tuning 	Dashboard 
Trace function to monitor registers 	Oscilloscope 

1.3 File System

For MCE tools, there are four types of files in the system, as shown in Table 3. All files are in text format and are located in the same folder. As described later, *.mcs and *.ldf files still exist in the iSD file system. Please refer to [4] for details.

Table 3 File Types for MCE tools

Extension	File	Used in iSD File System
*.irc	MCEDesigner file	No
*.mc2	MCEWizard configuration file	No
*.mcs	Script file	Yes
*.map	Parameter map file for MCEDesigner	No
*.txt	Parameter file generated by MCEWizard	No
*.ldf	Loader format of firmware, parameter, and script	Yes

Within the iSD, the file system is more complicated than the one found in the MCE tools. In a project folder of the iSD, there are four sub folders as shown in Figure 1; generated, lib, Script, and settings. Details of each folder will be described in following chapters. Usually users do not have to handle the files separately in the file system.

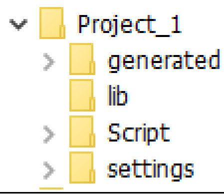


Figure 1 Project File structure

1.3.1 generated

The "generated" folder contains parameter setting information and the output file for programming. Figure 2 shows an example of these files as they appear in the "generated" folder.

There are two *.ldf files in this folder. A *.ldf file whose name starts with a device name (in this case, IMC101T-F048) is an image of the firmware for the device. Another *.ldf file named "out.ldf" contains parameter and script images. Both *.ldf files are in text format, and they contain programming commands for the targeted iMOTION™ device.

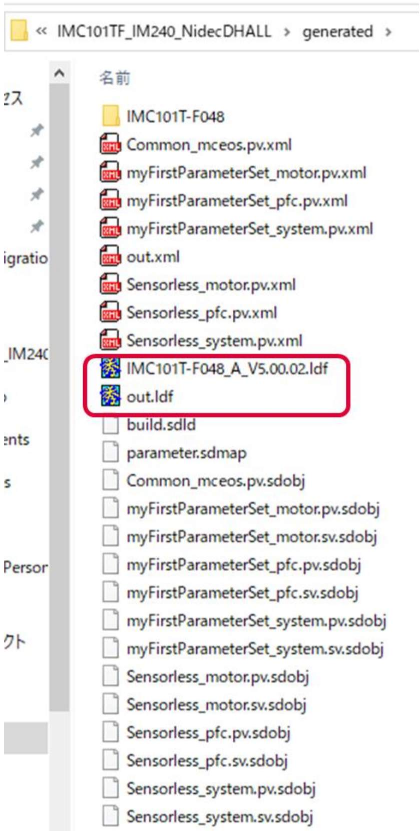


Figure 2 Files in "generated" Folder in a Project Folder

Other files (*.sdmap, *.sdobj, *.xml) contain parameter information which are configured in the Configuration Wizard, and the user does not have to edit them.

1.3.2 lib

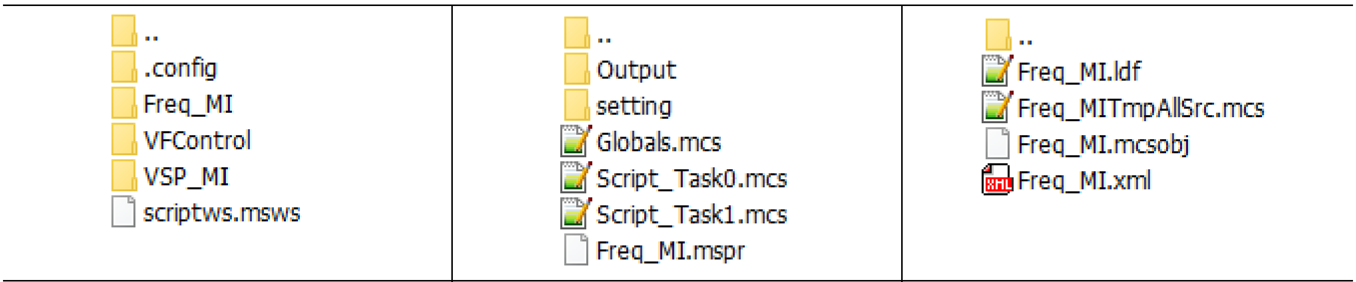
In the instance of a self-contained project, users will find the project's associated SD Pack file.

1.3.3 Script

The "Script" folder contains script project information. Figure 3 (a) shows an example of the content in the Script folder. Each script project is stored in independent sub folders. The "scriptws.msws" file contains script project information for this iSD project.

Figure 3 (b) shows content in a script project folder. There are two sub folders (Output and setting), and three *.mcs files (Global.mcs, Script_Task0.mcs, and Script_Task1.mcs) which are corresponded to content in the Script Editor. The *.mspr file contains the execution period and execution steps of task0 and task1 of the script.

Figure 3 (c) shows content in the "Output" folder. The *.mcs file is a combined script code that is equivalent to the script code for the MCEWizard. The *.ldf file is a compiled script image for programming, and it is also equivalent to the MCEWizard. Other files are intermediate files for compiling the script project.



(a) Script Folder	(b) Script Project Sub Folder	(c) Output Folder

Figure 3 File Structure for Script Folder

The "setting" folder in a script project sub folder contains debugging information for the Script Editor.

1.3.4 settings

The "setting" folder contains the *.ozsi file (Oscilloscope configuration file) and *.cwproj file (iSD project configuration files).

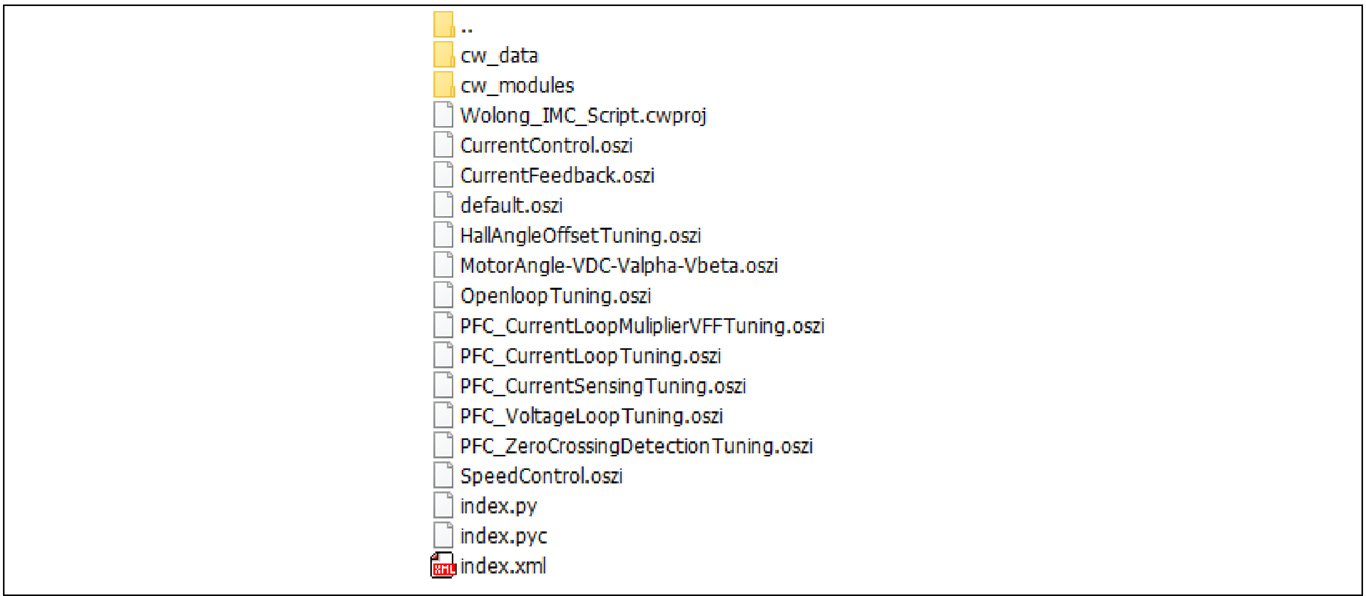


Figure 4 File Structure for settings Folder

2 Lower-level Parameter Comparison

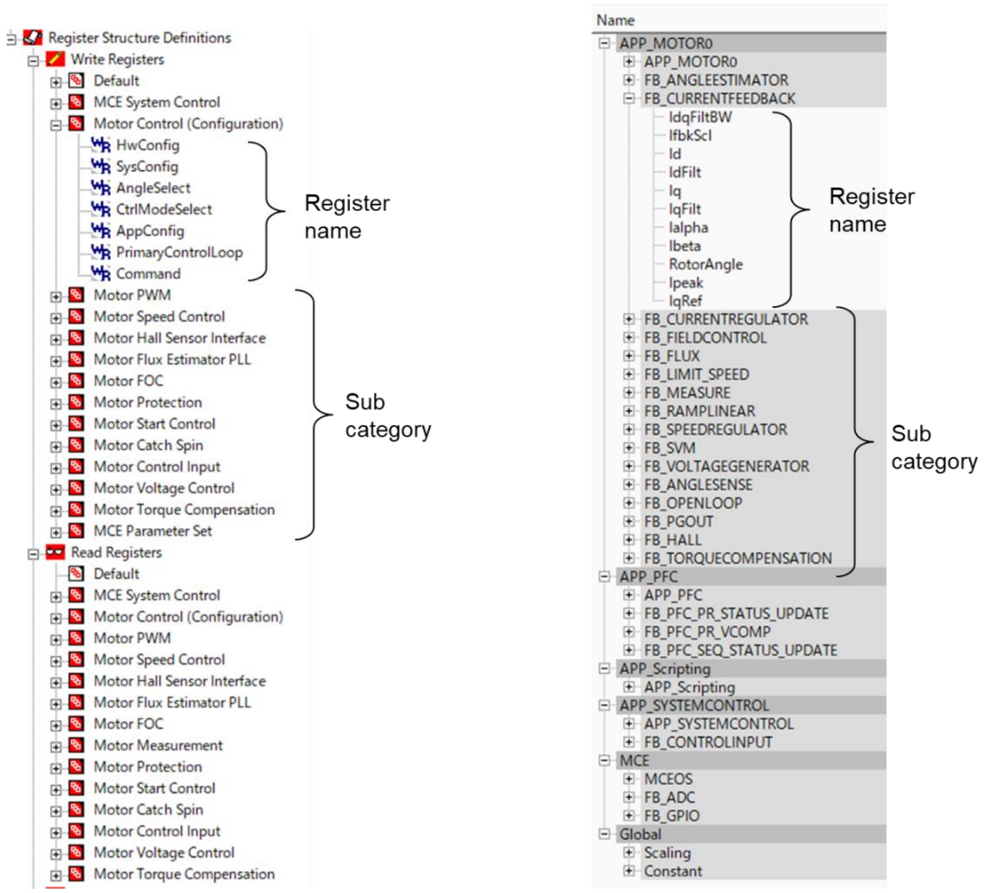
2.1 Overview

2.1.1 Register Categories in MCEDesigner and iMOTION™ Solution Designer

MCE registers can be checked in the MCEDesigner in the firmware version before 1.3.x. It is also possible to check MCE registers in the Dashboard of the iSD. MCE registers are categorized based on their functionality for the MCE tools. Users will find the categorization for the iSD differs greatly from the MCEWizard.

Figure 5 shows the example of variable categories for MCEDesigner and Dashboard in the iSD. For the MCEDesigner, there are two major categories; Write Registers and Read Registers. While all of registers are listed in the Read Registers, only dynamic registers are shown in the Write Registers category. There are several sub categories in each major category, and the MCE registers are listed based on their functionality.

As for the iSD, there is no categorization for write and read. The MCE registers are categorized based on their application functionality at the 1st level, and modular functionality at the 2nd level where MCE registers are listed accordingly.



(a) MCEDesigner (b) iMOTION™ Solution Designer

Figure 5 Variable Table in MCEWizard and iMOTION™ Solution Designer

2.1.2 Register Nomenclature

Most of the names of the MCE registers are the same between the MCEDesigner and iSD. Some registers have different naming structures between the MCEDesigner and iSD, and it will be described in later chapters.

Since PFC function is greatly improved in the iSD as compared to the MCE tools, most of the registers for the PFC function are newly defined.

The method for referencing MCE registers from script is different between the MCE tools and iSD. For the MCE tools, the register name is very simple, such as "MotorSpeed". As for the iSD, the register name consists of a sub category and register name. For example, MotorSpeed register is included in the "APP_MOTOR0" sub category, so the script needs to reference MotorSpeed as "APP_MOTOR0.MotorSpeed". Detailed naming comparison between the MCE tools and iSD will be described in later chapters.

2.2 Control Registers

This chapter compares the registers between FW1.3.x and FW5.x. As described in section 2.1.2 , the register name is changed. In addition, the register ID has been also changed. It should be considered when user UART communication or JCOM communication is used.

2.2.1 How to Use Register IDs in Communication Commands

Each register has independend ID which consists of FBID and Register ID. This structure is same as one in FW1.3.x, but the values are different. For example, register ID for U-phase current (Iu) is changed as shown in Table 4. The FBID for FW5.x corresponds to the AppID for FW1.3.x.

Table 4 Register IDs for U-Phase Current (Iu)

FW1.3.x		FW5.x	
AppID	RegisterID	FBID	RegisterID
1	122	27	66

When user mode UART or JCOM communication is used, this difference needs to be considered. Figure 6 shows comparison of user mode UART commands between FW1.3.x and FW5.x for Register Read command (command ID=0x05) to read U-phase current (Iu). The AppID value should be replaced to the FBID value, and the register ID should be also changed accordingly. Same discussion is applicable to JCOM commands.

Node Address	Command	App ID	Register ID	0x00	0x00	Checksum	
0x01	0x05	0x01	0x7A	0x00	0x00	0xFE	0x80

(a) FW1.3.x

Node Address	Command	FB ID	Register ID	0x00	0x00	Checksum	
0x01	0x05	0x01	0x1B	0x42	0x00	0xE4	0xB8

(b) FW5.x

Figure 6 User Mode UART Commands

2.2.2 System Control Registers

Table 5 shows comparison of system control registers between FW1.3.x and FW5.x. Bitmap is changed for some registers. Changed items in bitmap are shown in Table 6.

Table 5 Comparison of System Control Registers

FW1.3.x			FW5.x			Comparison
AppID	**Register ID**	**Register Name**	**FB ID**	**Register ID**	**Register Name**	
0	0	ParPageConf	248	9	MCEOS.ParSetConf	Name changed
0	2	InterfaceConf0	248	10	MCEOS.TargetInterfaceConf	Name changed
0	3	InterfaceConf1	248	12	MCEOS.UARTConf	Name changed Bitmap changed
0	22	GKConf	248	19	MCEOS.GKConf	Same
0	24+*N*	GPIOs[*N*] (N=0 to 29)	52	*N*	FB_GPIO.GPIO[*N*]	Same
0	61	FeatureID_selectH	248	1	MCEOS.Feature ID	Name changed
0	62	SysTaskTime	248	2	MCEOS.SysTaskTime	Same
0	63	SysTaskConfig	248	3	MCEOS.SysTaskConfig	Same
0	80	CPU_Load			(Not Supported)	(Not supported in FW5.x)
0	81	InternalTemp	250	13	MCEOS.InternalDieTemp	Name changed
0	82	SW_Version	250	5	MCEOS.FW_Version	Name changed Bitmap changed
0	84	CPU_Load_Peak	250	9	MCEOS.CPUloadPeak	Name changed

Table 6 Change of Bitmap Information

Register Name	New Bitmap Definition
MCEOS.ParSetConf[3:0]	0: UART control 1: Mutlple parameter handling is not enabled. Parameter sets speciried in MCEOS.ParSetConf[9:4] is loaded as a default parmater set. 2: Analog input 3: GPIO pins
MCEOS.FW_Version	[31:28] Major version [27:22] Minor version [21:16] Path [15:13] Build type [12:0] Build number

2.2.3 Motor Control Registers

Table 7 shows comparison of motor control registers between FW1.3.x and FW5.x. Bitmap or scaling factor is changed for some registers. Chaged items in bitmap or scaling are shown in Table 8.

Table 7 Motor Control Registers

FW1.3.x			FW5.x			Comparison
AppID	Register ID	Register Name	FB ID	Register ID	Register Name	
1	1	HwConfig	1	0	APP_MOTOR0.HwConfig	Bitmap changed
1	2	SysConfig	1	1	APP_MOTOR0.SysConfig	Bitmap changed
1	3	AngleSelect	1	3	APP_MOTOR0.AngleSelect	Same
1	4	CtrlModeSelect	1	4	APP_MOTOR0.CtrlModeSelect	Same
1	5	PwmFreq	1	5	APP_MOTOR0.PwmFreq	Same
1	6	PwmDeadtimeR	1	6	APP_MOTOR0.PwmDeadtimeR	Same
1	7	PwmDeadtimeF	1	7	APP_MOTOR0.PwmDeadtimeF	Same
1	8	SHDelay	34	0	FB_SVM.SHDelay	Same
1	9	TMinPhaseShift	1	8	APP_MOTOR0.TminPhaseShift	Same
1	10	TCntMin			(Not Supported)	(Not supported in FW5.x)
1	11	PwmGuardBand	34	1	FB_SVM.PwmGuardBand	Same
1	12	FaultEnable	1	9	APP_MOTOR0.FaultEnable	Bitmap changed
1	13	VdcOvLevel	27	0	FB_MEASURE.VdcOvLevel	Same
1	14	VdcUvLevel	27	1	FB_MEASURE.VdcUvLevel	Same
1	15	CriticalOvLevel	27	2	FB_MEASURE.CriticalVdcOvLevel	Same
1	16	RotorLockTime	1	10	APP_MOTOR0.RotorLockTime	Scaling changed
1	18	FluxFaultTime	1	11	APP_MOTOR0.FluxFaultTime	Scaling changed
1	19	GatekillFilterTime	1	12	APP_MOTOR0.GatekillFilterTime	Same
1	20	CompRef	1	13	APP_MOTOR0.CompRef	Same
1	21	BtsChargeTime	1	14	APP_MOTOR0.BtsChargeTime	Same
1	22	TCatchSpin	1	15	APP_MOTOR0.TCatchSpin	Same
1	23	DirectStartThr	1	16	APP_MOTOR0.DirectStartThr	Same
1	24	ParkTime	1	17	APP_MOTOR0.ParkTime	Same
1	25	ParkAngle	1	18	APP_MOTOR0.ParkAngle	Same
1	26	OpenloopRamp	1	19	APP_MOTOR0.OpenloopRamp	Same
1	27	IS_Pulses	1	20	APP_MOTOR0.IS_Pulses	Same
1	28	IS_Duty	1	21	APP_MOTOR0.IS_Duty	Same
1	29	IS_IqInit	17	0	FB_ANGLESENSE.IS_IqInit	Same

FW1.3.x			FW5.x			Comparison
AppID	Register ID	Register Name	FB ID	Register ID	Register Name	
1	30	KpSreg	33	0	FB_SPEEDREGULATOR.KpSreg	Same
1	31	KxSreg	33	1	FB_SPEEDREGULATOR.KxSreg	Same
1	32	MotorLim	26	0	FB_LIMIT_SPEED.MotorLim	Same
1	33	RegenLim	26	1	FB_LIMIT_SPEED.RegenLim	Same
1	34	RegenSpdThr	26	2	FB_LIMIT_SPEED.RegenSpdThr	Same
1	35	LowSpeedLim	1	22	APP_MOTOR0.LowSpeedLim	Same
1	36	LowSpeedGain	26	3	FB_LIMIT_SPEED.LowSpeedGain	Same
1	37	SpdRampRate	31	0	FB_RAMPLINEAR.RampRate	Name changed
1	38	MinSpd	1	23	APP_MOTOR0.MinSpd	Same
1	39	Rs	24	0	FB_FLUX.Rs	Same
1	40	L0	24	1	FB_FLUX.L0	Same
1	41	LSlncy	24	2	FB_FLUX.LSlncy	Same
1	42	VoltScl	24	3	FB_FLUX.VoltScl	Same
1	43	PllKp	16	0	FB_ANGLEESTIMATOR.PllKp	Same
1	44	PllKi	16	1	FB_ANGLEESTIMATOR.PllKi	Same
1	45	PllFreqLim	16	2	FB_ANGLEESTIMATOR.PllFreqLim	Same
1	46	AngMTPA	16	3	FB_ANGLEESTIMATOR.AngMTPA	Same
1	47	FlxTau	1	24	APP_MOTOR0.FluxTau	Same
1	48	AtanTau	16	4	FB_ANGLEESTIMATOR.AtanTau	Same
1	49	SpeedScalePsc	1	25	APP_MOTOR0.SpeedScalePsc	Same
1	50	SpeedScale	1	26	APP_MOTOR0.SpeedScale	Same
1	51	SpeedScaleRcp	1	27	APP_MOTOR0.SpeedScaleRcp	Same
1	52	SpdFiltBW	16	5	FB_ANGLEESTIMATOR.SpdFiltBW	Same
1	53	PGDeltaAngle	30	0	FB_PGOUT.PGDeltaAngle	Same
1	54	IfbkScl	21	1	FB_CURRENTFEEDBACK.IfbkScl	Same
1	55	KpIreg	22	0	FB_CURRENTREGULATOR.KpIreg	Same
1	56	KpIregD	22	1	FB_CURRENTREGULATOR.KpIregD	Same

FW1.3.x			FW5.x			Comparison
AppID	Register ID	Register Name	FB ID	Register ID	Register Name	
1	57	Kxlreg	22	2	FB_CURRENTREGULATOR.Kxlreg	Same
1	58	FwkLevel	23	0	FB_FIELDCONTROL.FwkLevel	Same
1	59	FwkKx	23	1	FB_FIELDCONTROL.FwkKx	Same
1	60	FwkCurRatio	23	2	FB_FIELDCONTROL.FwkCurRatio	Same
1	61	VdqLim	22	4	FB_CURRENTREGULATOR.VdqLim	Same
1	62	AngDel	23	3	FB_FIELDCONTROL.AngDel	Same
1	63	AngLim	23	4	FB_FIELDCONTROL.AngLim	Same
1	64	IdqFiltBW	21	0	FB_CURRENTFEEDBACK.IdqFiltBW	Same
1	65	Pwm2PhThr	1	28	APP_MOTOR0.Pwm2PhThr	Same
1	67	TShutdown	1	30	APP_MOTOR0.TShutdown	Same
1	68	CmdStop	20	0	FB_CONTROLINPUT.CmdStop	Same
1	69	CmdStart	20	1	FB_CONTROLINPUT.CmdStart	Same
1	70	CmdGain	20	2	FB_CONTROLINPUT.CmdGain	Same
1	71	AppConfig	0	0	APP_SYSTEMCONTROL.AppConfig	Bitmap changed.
1	72	NodeAddress	0	1	APP_SYSTEMCONTROL.NodeAddress	Same
1	73	PrimaryControlLoop	1	32	APP_MOTOR0.PrimaryControlLoop	Same
1	74	PhaseLossLevel	1	33	APP_MOTOR0.PhaseLossLevel	Same
1	75	TrqCompGain	44	0	FB_TORQUECOMPENSATION.TrqCompGain	Same
1	76	TrqCompAngOfst	44	1	FB_TORQUECOMPENSATION.TrqCompAngOfst	Same
1	77	TrqCompLim	44	2	FB_TORQUECOMPENSATION.TrqCompLim	Same
1	78	TrqCompOnSpeed	44	3	FB_TORQUECOMPENSATION.TrqCompOnSpeed	Same
1	79	TrqCompOffSpeed	44	4	FB_TORQUECOMPENSATION.TrqCompOffSpeed	Same
1	80	PolePair	44	5	FB_TORQUECOMPENSATION.PolePair	Same
1	81	FaultRetryPeriod	0	3	APP_SYSTEMCONTROL.FaultRetryPeriod	Same
1	85	HallAngleOffset	25	0	FB_HALL.HallAngleOffset	Same

FW1.3.x			FW5.x			Comparison
AppID	Register ID	Register Name	FB ID	Register ID	Register Name	
1	86	Hall2FluxThr	1	34	APP_MOTOR0.Hall2FluxThr	Same
1	87	Flux2HallThr	1	35	APP_MOTOR0.Flux2HallThr	Same
1	88	HallSampleFilter	25	2	FB_HALL.SampleFilter	Name changed
1	89	HallSpdFiltBW	25	3	FB_HALL.FrequencyBW	Name changed
1	94	HallTimeoutPeriod	1	36	APP_MOTOR0.HallTimeoutPeriod	Scale value changed. 1 count changed from 10 ms to 16 ms.
1	100	KpHallPLL	25	4	FB_HALL.KpHallPLL	Same
1	120	Command	1	64	APP_MOTOR0.Command	Same
1	121	TargetSpeed	1	65	APP_MOTOR0.TargetSpeed	Same
1	122	Iu	27	66	FB_MEASURE.Iu	Same
1	123	Iv	27	67	FB_MEASURE.Iv	Same
1	124	Iw	27	68	FB_MEASURE.Iw	Same
1	125	MotorSpeed	1	128	APP_MOTOR0.MotorSpeed	Same
1	126	I_Alpha	27	64	FB_MEASURE.Ialpha	Same
1	127	I_Beta	27	65	FB_MEASURE.Ibeta	Same
1	128	IdRef_Ext	1	66	APP_MOTOR0.IdRef_Ext	Same
1	129	IqRef_Ext	1	67	APP_MOTOR0.IqRef_Ext	Same
1	130	Vd_Ext	1	68	APP_MOTOR0.Vd_Ext	Same
1	131	Vq_Ext	1	69	APP_MOTOR0.Vq_Ext	Same
1	132	SwFaults	1	74	APP_MOTOR0.SwFaults	Bitmap changed
1	133	SequencerState	250	11	MCEOS.Motor_SequencerState	Name changed
1	134	FaultClear	1	75	APP_MOTOR0.FaultClear	Same
1	135	FaultFlags	1	80	APP_MOTOR0.FaultFlags	Bitmap changed
1	136	VdcRaw	27	70	FB_MEASURE.VdcRaw	Same
1	137	VdcFilt	27	69	FB_MEASURE.VdcFilt	Same
1	138	FluxAngle	16	64	FB_ANGLEESTIMATOR.RotorAngle	Same
1	139	Flx_M	16	70	FB_ANGLEESTIMATOR.Pll_M	Name changed
1	140	Abs_MotorSpeed	1	85	APP_MOTOR0.Abs_MotorSpeed	Same
1	141	IdFilt	21	65	FB_CURRENTFEEDBACK.IdFilt	Same
1	142	IqFilt	21	67	FB_CURRENTFEEDBACK.IqFilt	Same
1	143	IdFwk	23	68	FB_FIELDCONTROL.IdFwk	Same
1	144	VTH	1	76	APP_MOTOR0.VTH_val	Name changed
1	145	FluxAlpha	24	64	FB_FLUX.FluxAlpha	Same
1	146	FluxBeta	24	65	FB_FLUX.FluxBeta	Same

FW1.3.x			FW5.x			Comparison
AppID	Register ID	Register Name	FB ID	Register ID	Register Name	
1	147	Flx_Q	16	69	FB_ANGLEESTIMATOR.Pll_Q	Name changed
1	148	TrqRef	23	128	FB_FIELDCONTROL.TrqRef	Same
1	149	Id	21	64	FB_CURRENTFEEDBACK.Id	Same
1	150	Iq	21	66	FB_CURRENTFEEDBACK.Iq	Same
1	151	V_Alpha	35	64	FB_VOLTAGEGENERATOR.Valpha	Name changed
1	152	V_Beta	35	65	FB_VOLTAGEGENERATOR.Vbeta	Name changed
1	153	SpeedError	33	67	FB_SPEEDREGULATOR.SpeedError	Same
1	154	MotorCurrent	1	78	APP_MOTOR0.MotorCurrent	Same
1	155	OpenLoopAngle	29	65	FB_OPENLOOP.Angle	Name changed
1	156	Vd	22	64	FB_CURRENTREGULATOR.Vd	Same
1	157	Vq	22	65	FB_CURRENTREGULATOR.Vq	Same
1	158	MotorVoltage	1	77	APP_MOTOR0.MotorVoltage	Same
1	159	TrqRef_Ext	44	64	FB_TORQUECOMPENSATION.TrqCompOuput	Name changed
1	162	SpdRef	31	64	FB_RAMPLINEAR.Ramp_out	Name changed
1	164	ControlFreq	20	65	FB_CONTROLINPUT.FrequencyInput	Name changed
1	165	ControlDuty	20	66	FB_CONTROLINPUT.DutyInput	Name changed
1	167	HallAngle	25	64	FB_HALL.HallAngle	Same
1	168	HallMotorSpeed	25	66	FB_HALL.HallSpeed	Name changed
1	169	FluxMotorSpeed	16	68	FB_ANGLEESTIMATOR.MotorSpeed	Name changed
1	170	RotorAngle	1	129	APP_MOTOR0.RotorAngle	Same
1	171	MotorStatus	1	79	APP_MOTOR0.MotorStatus	Same
1	175	PositionCounter			(Not Supported)	(Not supported in FW5.x)
1	176	PositionCounter_H			(Not Supported)	(Not supported in FW5.x)
1	181	HallStatus			(Not Supported)	(Not supported in FW5.x)
1	182	Hall_FrequencyOut	25	65	FB_HALL.HallFreq	Name changed
1	183	HallPLL_FrequencyAdjust			(Not Supported)	(Not supported in FW5.x)
1	184	Hall_Atan_Angle	25	69	FB_HALL.ATan_Angle	Same
1	185	HallU	25	67	FB_HALL.HallU	Same

FW1.3.x			FW5.x			Comparison
AppID	Register ID	Register Name	FB ID	Register ID	Register Name	
1	186	HallV	25	68	FB_HALL.HallV	Same
1	187	Ipeak	27	73	FB_MEASURE.Ipeak	Same
1	188	CurrentAmpOffset 0	27	74	FB_MEASURE.IOffset0	Name changed
1	189	CurrentAmpOffset 1	27	75	FB_MEASURE.IOffset1	Name changed
1	190	TrqRef_Total	33	64	FB_SPEEDREGULATOR.Trq Ref	Name changed
1	191	TrqCompBaseAng le	44	65	FB_TORQUECOMPENSATI ON.TrqCompBaseAngle	Same
1	194	TrqCompStatus	44	67	FB_TORQUECOMPENSATI ON.TrqCompStatus	Same

Table 8 **Change of Bitmap or Scaling Information**

Register Name	New Bitmap Definition
APP_MOTOR0.HwConfig	1. Added three phase leg shunt configuration. 2. Removed minimum pulse configuration because the feature is removed. 3. Added current offset enable/disable after fault. 4. Moved current offset number of sample configuration to FB_MEAUSRE.OffsetSample.
APP_MOTOR0.SysConfig	1. Added torque compensation enable configuration (moved from APPConfig). 2. Removed FastControlRate configuration and moved to separate parameter (APP_MOTOR0.FastControlRate). 3. Removed hall related conigurations (HALLInputMask, HallSelect, HALL_Atan, Hall_Cmp_Hys) and moved to separate parameter (APP_MOTOR0.HallConfig).
APP_MOTOR0.FaultEnable APP_MOTOR0.SwFaults APP_MOTOR0.FaultFlags	1. Added GK pin fault display enable/disable. 2. Added current offset fault enable/disable.
APP_MOTOR0.RotorLockTime	Scale value is changed. 1 count represents 16 ms instead of 10 ms.
APP_MOTOR0.FluxFaultTIme	Scale value is changed. 1 count represents 16 ms instead of 10 ms.
APP_SYSTEMCONTROL.AppConfig	1. Removed enable restart after fault. Restart after fault is enabled if APP_SYSTEMCONTROL.FaultRetryNumber is not zero. 2. Moved TorqueComp enable to APP_MOTOR0.SysConfig. 3. Moved HallAtan period configuration to APP_MOTOR0.HallTanPeriod.

2.2.4 PFC Control Registers

Register structure for the new FW and iSD is varies greatly from the old FW and MCE tools. For details, please refer to [3] and [6].

2.2.5 Script Registers

Table 9 shows a comparison between the script registers. GPIO_IN_L and GPIO_IN_H for MCE tool are in 16 bit integer, and they are integrated into FB_GPIO_Status, which is in 32 bit integer. A similar change occurs for GPIO_OUT_L and H. Please refer to [3] and [6] for details.

Table 9 Script Registers

<table>
<tr><th colspan="3">FW1.3.x</th><th colspan="3">FW5.x</th><th rowspan="2">Comparison</th></tr>
<tr><th>AppID</th><th>Register ID</th><th>Register Name</th><th>FB ID</th><th>Register ID</th><th>Register Name</th></tr>
<tr><td>4</td><td>0</td><td>Script_UserVersion</td><td>4</td><td>0</td><td>APP_Scripting.ScriptUserVersion</td><td>Name changed</td></tr>
<tr><td>4</td><td>1</td><td>Script_command</td><td>4</td><td>64</td><td>APP_Scripting.Command</td><td>Name changed</td></tr>
<tr><td>4</td><td>98+N</td><td>ADC_ResultN (N=0 to 11)</td><td>51</td><td>64+N</td><td>FB_ADC.adc_result[N]</td><td>Name changed</td></tr>
<tr><td>0</td><td>24+N</td><td>GPIOs[N] (N=0 to 29)</td><td>52</td><td>N</td><td>FB_GPIO.GPIO[N]</td><td>Name changed</td></tr>
<tr><td>4</td><td>110</td><td>GPIO_IN_L</td><td>52</td><td rowspan="2">64</td><td rowspan="2">FB_GPIO.GPIO_Status</td><td rowspan="2">Name changed</td></tr>
<tr><td>4</td><td>111</td><td>GPIO_IN_H</td><td>52</td></tr>
<tr><td>4</td><td>112</td><td>GPIO_OUT_L</td><td>52</td><td rowspan="2">65</td><td rowspan="2">FB_GPIO.GPIO_Set</td><td rowspan="2">Name chagned</td></tr>
<tr><td>4</td><td>113</td><td>GPIO_OUT_H</td><td>52</td></tr>
</table>

3 State Machine Handling

3.1 Motor State Machine

State handling for the motor control is equivalent between FW1.3.x and FW5.x. However, FW5.x has increased MOTORRUN state to distinguish which angle estimation method is used.

Table 10 is the comparison chart of MOTORRUN state between FW1.3.x and FW5.x. As you can see, the iSD has four types of MOTORRUN states for different angle estimation methods. It is necessary to consider these different MOTORRUN states when the script code for the MCE tools is ported from FW1.3.x to FW5.x.

Table 10 Motor State Comparison

			FW1.3.x	**FW5.x**
State Machine Register			SequencerState	MCEOS.Motor_SequencerState
IDLE			0	0
STOP			1	1
OFFSETCAL			2	2
BTSCHARGE			3	3
MOTORRUN	Angle Estimation Method	Flux	4	4
		Hall	4	10
		Hybrid	4	11
		Open Loop	4	12
FAULT			5	5
CATCHSPIN			6	6
PARKING			7	7
OPENLOOP			8	8
ANGLESENSING			9	9
STANDBY			(Not supported)	13

From FW5.x, a new state machine is added for stand-by function. The value of register MCEOS.Motor_SequencerState is 13 for stand-by mode. Please refer to [3] for detailed information about the state machine.

3.2 PFC State Machine

Since the PFC is greatly improved from the iSD, the state machine handling is different between the MCE tools and iSD. Table 11 shows the comparison chart of the PFC state between the MCE tools and iSD. For more information, please refer to [3].

Table 11 PFC State Comparison

		FW1.3.x	**FW5.x**
State Machine Register		PFC_SequencerState	MCEOS.PFC_SequencerState
State No	0	IDLE	PFC_IDLE
	1	STOP	PFC_OFFSETCAL
	2	OFFSETCAL	RUN_CTRLMODE0
	3	RUN	RUN_CTRLMODE1
	4	(N/A)	RUN_CTRLMODE2
	5	FAULT	PFC_FAULT
	6	(N/A)	RUN_CTRLMODE3
	7	(N/A)	PFC_STANDBY

4 Migration Step Example

This chapter will describe how to port an existing project for MCE tools to the iSD.

4.1 New Project Generation

In the beginning, it is necessary to create a new project.

4.2 Motor & PFC Configuration by Configuration Wizard

In the next step, it is necessary to configure the motor and PFC related parameters using the Configuration Wizard. This section will describe the item's differences between the iSD and MCEWizard.

After inputting the required configuration items, users need to press the "Verify" button for each section.

4.2.1 IC Configuration

In this section, it is necessary to set configuration for user UART communication, CPU clock compensation, multiple parameter handling, and gatekill input selection. Table 12 shows the equivalent configuration between the iSD and MCEWizard. All questions for the MCEWizard are shown in "Advanced Mode".

Table 12 Comparison of IC Configuration

Item in iSD		Item in MCEWizard	
COMM	User/Host COMM port selection	Q18	User UART Function Definition
	UART baud rate	Q20	User UART Baud Rate
	UART TX delay	Q21	User UART Tx Delay Time
	UART Node Address	Q17	UART Node Address
Options	Class-B Safety	Q23	Safety Function Tests Enable/Disable
	Disable CPU clock compensation	Q25	Temperature-based CPU clock compensation
	Multi-parameter input mode	Q30	Multiple Motor Parameter Set Support
Motor Protection	Overcurrent trip signal source selection	Q91	Gatekill Input Source
Standby	CPU at idle configuration	N/A	
	Low power mode enable	N/A	

4.2.2 User Pin Configuration

When the script is enabled, users can configure pin functions which are used in the script. Users should check the digital pins (GPIO) or analog pins that are used in the script. It is similar to the "Script Configuration" tab in the "Script Edit Page" of the MCEWizard.

4.2.3 System Hardware configuration

This section will describe how to configure the motor and PFC parameters, and how to port variables from the MCEWizard to the iSD. Since PFC function is improved with the iSD, there are more parameters to input into the iSD compared to the MCEWizard.

Some parameters are grey-hatched and users cannot change the value. The value of these variables is automatically calculated by the iSD based on user input, so users can check whether circuit configuration is correct or not. One example is shown in Figure 7. When the user inputs (1) and (2), (3) and (5) are calculated automatically as recommended by the iSD. Users can refer to these values and input the actual circuit configuration in (4) and (6). The iSD calculates MCE parameter values based on user input (4) and (6).

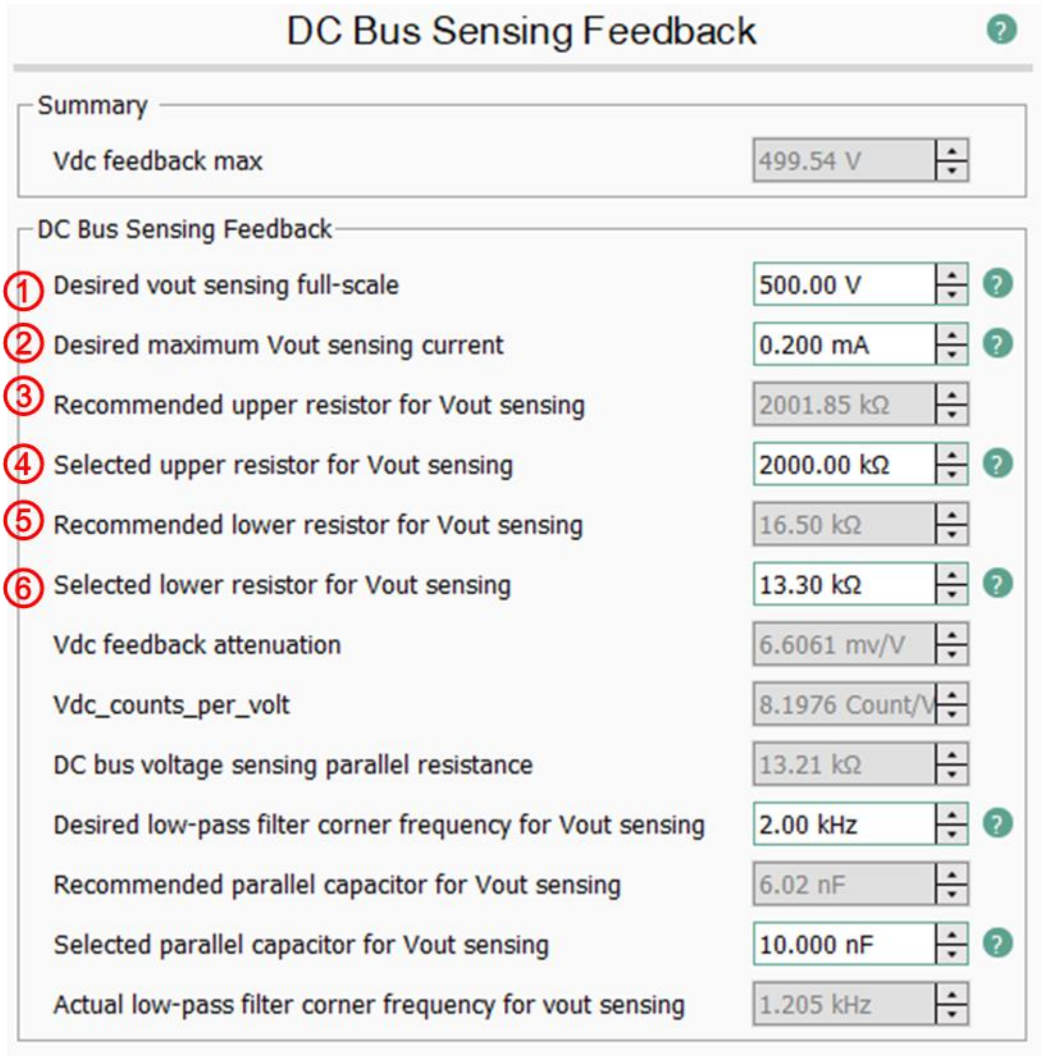


Figure 7 Automatic Calculation of Parameter Values for Recommendation

Some configuration items are shown only in expert mode.

4.2.3.1 Input Supply

The Input Supply section determines the power supply information. The value of "Input supply" depends on the system configuration (PFC is used or not), and maximum and minimum voltage information also depends on the type of input supply (AC with PFC, AC rectifier or DC).

Table 13 Input Supply

Item in iSD		Item in MCEWizard	
ADC	Input Supply	Q24	Input Supply Voltage

4.2.3.2 PFC Basic Input and Output

In this section, basic PFC input (AC) and output (DC) parameters are configured.

Table 14 PFC Basic Input and Output

Item in iSD	Item in MCEWizard	
Minimum AC input voltage	Q128	Minimum AC Input Voltage
Nominal AC input voltage	N/A	
Maximum AC input voltage	Q130	Maximum AC Input Voltage
Minimum AC line frequency	N/A	
Nominal AC line frequency	Q113	AC Input Frequency
Maximum AC line frequency	N/A	
Maximum output power	N/A	
Efficiency	N/A	
Target DC bus voltage	Q16	Target DC Bus Voltage Initialization

4.2.3.3 PFC Power Stage

In this section, PFC circuit configurations are determined. Since PFC function is improved for iSD over the MCE tools, most of the input items are newly defined from the iSD.

Table 15 PFC Power Stage

Item in iSD	Item in MCEWizard	
Desired hold-up time	N/A	
Desired minimum output voltage at hold-up	N/A	
Selected output capacitance	N/A	
Inductance	Q109	PFC Inductance
Rated inductor current	N/A	See Note below
Inductance at rated current	N/A	
Boost diode average forward voltage	N/A	
Gate driver propagation delay	Q118	PFC Current Sample Delay Time

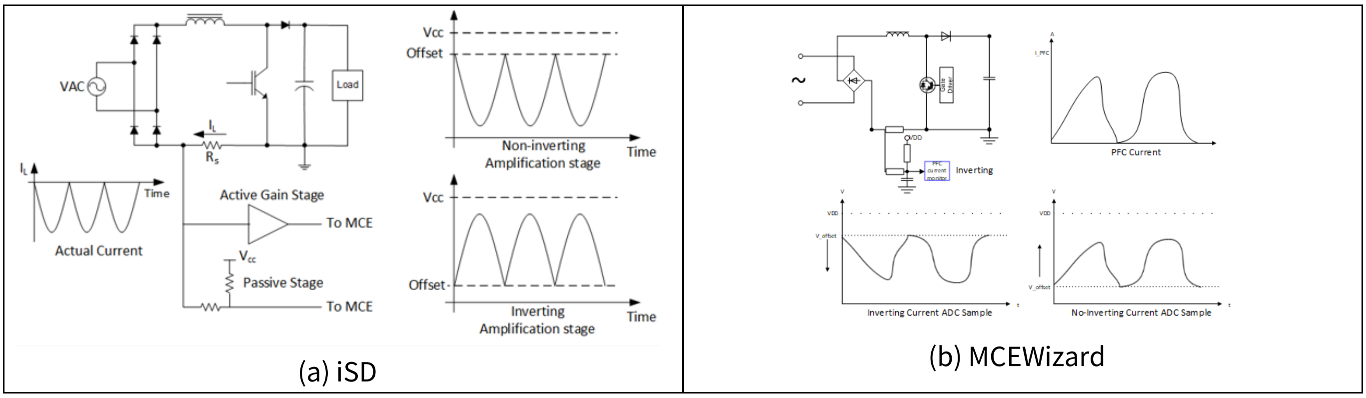
Note: *The iSD offers compensation of inductance degradation effect by inductor current. If the user inputs an inductance value at the rated current, the system automatically calculates the inductance value at the current level. If it is not necessary to consider this effect, please input the same value in "inductance" and "inductance at rated current".*

4.2.3.4 PFC Current Sensing

Table 16 PFC Current Sensing

Item in iSD	Item in MCEWizard	
Current sense polarity	Q112	PFC Current Sensing Input Polarity. See note below.
Current sense resistor	(used to calculate Q110)	
Current sense amplifier gain	N/A	
Current sense amplifier offset	N/A	
Current sense internal gain	Q111	PFC Internal Current Feedback Amplifier Gain
Current sense amplifier delay	Q118	PFC Current Sample Delay Time
Selected ADC sample offset time	N/A	

Note: *Definition of current sense polarity is different between the iSD and MCEWizard. If "Non-Inverting" is selected in the MCEWizard, "Inverting" should be selected in the iSD. Please refer to Figure 8 for the definition of polarity of PFC current sensing.*



(a) iSD (b) MCEWizard

Figure 8 Definition of Polarity of PFC Current Sensing

4.2.3.5 PFC VAC Sensing

Table 17 PFC VAC Sensing

Item in iSD	Item in MCEWizard	
Desired VAC sensing full-scale	N/A	
Desired maximum VAC sensing current	N/A	
Selected lower resistor for VAC sensing	Q115	AC Voltage Sensing Low Resistor
Selected upper resistor for VAC sensing	Q114	AC Voltage Sensing High Resistor
Desired low-pass filter corner frequency for VAC sensing	N/A	
Selected parallel capacitor for VAC sensing	N/A	

4.2.3.6 DC Bus Sensing Feedback

Table 18 DC Bus Sensing Feedback

Item in iSD		Item in MCEWizard	
DC Bus Sensing Feedback	Desired vout sensing full-scale	N/A	
	Desired maximum Vout sensing current	N/A	
	Selected upper resistor for Vout sensing	Q50	DC Bus Sensing High Resistor
	Selected lower resistor for Vout sensing	Q51	DC Bus Sensing Low Resistor
	Desired low-pass filter corner frequency for Vout sensing	N/A	
	Selected parallel capacitor for Vout sensing	N/A	

4.2.3.7 Inverter

Table 19 Inverter

Item in iSD		Item in MCEWizard	
Power Switch	Rated inverter DC bus voltage	N/A	
	Rated inverter output current	N/A	
	Inverter dead time	Q78	Inverter Dead Time
DC Link	Maximum inverter DC-bus voltage	N/A	
Gate Driver	Gate sense high	Q81	GateSense High-Side Devices
	Gate sense low	Q80	GateSense Low-Side Devices
	Bootstrap capacitor charge time per phase	Q82	Total Bootstrap Cap Charge Time See note below

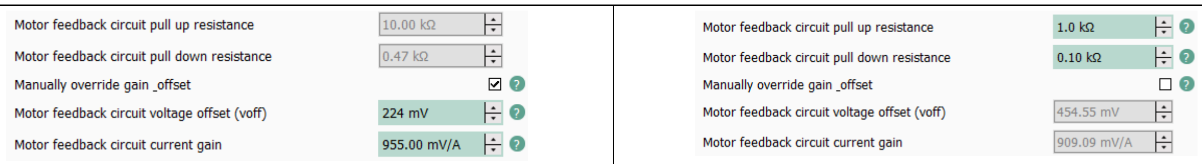
Note: *Definition of bootstrap capacitor charge time is different between the MCEWizard and iSD. In the MCEWizard, total bootstrap capacitor charge time should be input, but in the iSD, charge time "per phase" should be input. If the total charge time is 10 ms, 3.33 ms value is used for the iSD.*

4.2.3.8 Motor Current Sensing

Table 20 shows the comparison between the iSD and MCEWizard for motor current sensing configurations. It is possible to calculate the motor feedback circuit voltage offsets and motor feedback circuit current gain automatically in the iSD by unchecking the "Manually override gain offset" checkbox, as shown in Figure 9(b). Users can then input these values manually as shown in Figure 9(a). Please note that the iSD can support only phase shift for single shunt current sensing, and minimum pulse scheme is not supported (MCEWizard Q87).

Table 20 Motor Current Sensing

Item in iSD		Item in MCEWizard	
Summary	Current sensor	Options Page	Motor Current Shunt Configuration
Current Feedback	Minimum single shunt phase shift active pulse	Q90	Phase Shift Window Size
	Current feedback sampling delay	Q86	Gate Driver Propagation Delay
	Motor current feedback shunt (rsh)	N/A	(This value is used to calculate Q83 and Q85)
	Motor feedback ADC gain (gint)	Q84	Internal Current Feedback Amplifier Gain
	Motor feedback circuit voltage offset (voff)	Q85	Motor1 Current Input to ADC offset Voltage
	Motor feedback circuit current gain	Q83	Motor1 Current Input Scaling



(a) Input offset and gain directly

(b) Input voltage divider information, then gain, and offset are automatically calculated

Figure 9 Configuration of Current Input Offset and Gain

Note: *When Leg shunt is selected in Current sensor, the new selection item "Leg shunt type" appears. In FW5.x, three phase can be selected in addition to two phase.*

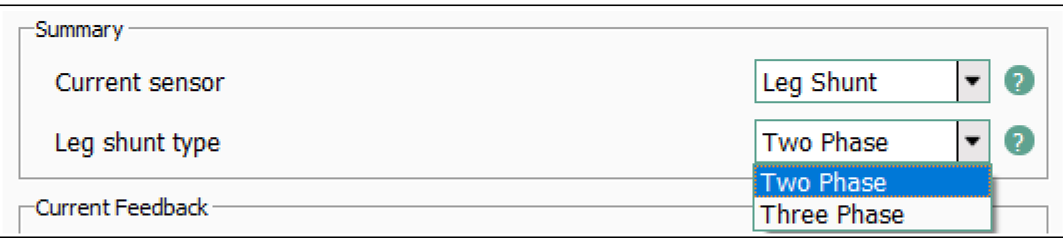


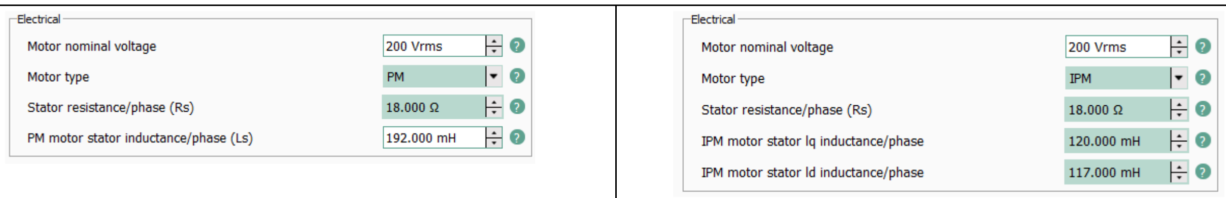
Figure 10 Current Sensor Configuration

4.2.3.9 Motor

Table 21 shows the comparison between the iSD and MCEWizard for motor parameters. When "PM" is selected for "Motor type", only one inductance information appears. If "IPM" is selected for "Motor type", two types of inductance (Lq and Ld) items appear and the user needs to input a value for each item.

Table 21 Motor

Item in iSD		Item in MCEWizard	
Summary	Motor rated amps/phase	Q2	Motor Rated Amps
	Motor poles	Q3	Motor Poles
	Motor back EMF constant	Q7	Motor Back EMF Constant (Ke)
	Position sensor	Options Page	Hall Sensor Configuration
	Motor model name	Q1	Motor Model Name
Electrical	Motor nominal voltage	N/A	
	Motor Type	N/A	
	Stator resistance/phase (Rs)	Q4	Motor Stator Resistance
	PM motor stator inductance/phase (Ls)	Q5 & Q6	Motor Lq Inductance Motor Ld Inductance (Lq=Ld)
	IPM motor stator Iq inductance/phase	Q5	Motor Lq Inductance
	IPM motor stator Id inductance/phase	Q6	Motor Ld Inductance



(a) Motor Type ="PM"   (b) Motor Type = "IPM"

Figure 11 Motor Type

4.2.3.10 Load

Table 22 Load

Item in iSD		Item in MCEWizard	
Configuration	Load type	N/A	
	Model name	N/A	
Power	Rated power	N/A	
	Minimum speed	Q9	Minimum Running Speed
	Maximum speed	Q8	Motor Max RPM

4.2.4 Motor Control Configuration

4.2.4.1 Motor Control Configuration

Table 23 Motor Control Configuration

Item in iSD		Item in MCEWizard	
Control Rates	PWM Frequency	Options Page	Motor 1 PWM Frequency
	Current control update rate scaler	Options Page	Motor Fast Control Rate
	Speed control update rate scaler	N/A	See note below
Control Modes	Rotor angle feedback selection	Options Page	Hall Sensor Configuration
	Motor control modes	Options Page	Motor Control Mode

Note: *It is possible to change the speed control update rate scaler explicitly in the iSD. This is a ratio between the current control update rate and speed controller update rate.*

4.2.4.2 Control Regulators

Table 24 shows the comparison between the iSD and MCEWizard for motor control registers. As for current regulators, it is possible to configure current regulator bandwidth for d and q axes independently, which the MCEWizard does not support.

Table 24 Control Regulators

Item in iSD		Item in MCEWizard	
Velocity Regulator	Speed feedback filter time constant	Q61	Speed Feedback Filter Time Constant
	Speed regulator proportional gain manually input	Q62	Speed Regulator Proportional Gain
	Speed regulator integral gain manually input	Q63	Speed Regulator Integral Gain
IPM and Field Control	Field weakening current limit	Q54	Field Weakening Current Limit
	Field weakening modulation threshold	N/A	
	Field weakening control gain	N/A	
Current Regulators	D axis current regulator bandwidth	Q58	Current Regulator Bandwidth
	Q axis current regulator bandwidth	Q58	Current Regulator Bandwidth

4.2.4.3 FOC and Inverter

Table 25 FOC and Inverter

Item in iSD		Item in MCEWizard	
Voltage Generator	SVPWM over modulation	Q75	Over Modulation
	DC bus compensation	Q59	Enable DC Bus Compensation
SVPWM	Motor PWM type	Q76	Motor PWM Type

Item in iSD		Item in MCEWizard	
	Three phase to two phase transition speed	Q77	3ph to 2ph PWM Switch Over Speed
Signal Sampling	Low noise single current sensing	Q88	Enable Low Noise Phase Shift Current Sensing
	Low noise sensing to normal transition modulation	N/A	See note below
	Normal to low noise sensing transition modulation	N/A	
	Number of offset samples	N/A	

Note: *These values are fixed in the MCEWizard and the user cannot change them. With the iSD, these values are configurable and it will help to reduce acoustic noise in some applications.*

4.2.4.4 Application

This section provides configuration for motor start up method, current limit profile, and command input. Items shown in the window vary by selection from items in "Summary".

Table 26 Application

Item in iSD		Item in MCEWizard	
Summary	Motor starting method	N/A	
	Motor angle initialization	N/A	
	Catch spin before start	Q95	Catch Spin Before Start
	PG output	N/A	
	Command input	Options Page	Application Control Interface
Motor Starting	Open loop ramp rate	Q11	Open Loop Speed Ramp Rate
	Parking Time	Q12	Parking Time
	Parking angle	N/A	
	Iq initial value	N/A	
	Angle sense number of pulses	N/A	
	Angle sense duty value	N/A	
	Catch spin time	Q96	Catch Spin Time
	Direct start maximum speed threshold	N/A	
Current Limits	Motoring current limit	Q52	Motoring Current Limit
	Regeneration current limit	Q53	Regeneration Current Limit
	Low speed limit	Q14	Low Speed Current Limit
	Low speed threshold	Q13	Low Speed Threshold
	Regeneration limit minimum speed	N/A	
Command Input	Speed control input ramp rate limit	Q10	Speed Ramp Rate
	Speed feedback pulse setting	Q57	PG Pulse Per Revolution
	Control Input Measure	Q26	Control Input measurement

4.2.4.5 Angle Feedback

This section provides configuration for the flux estimator and angle estimator. The angle estimator section is shown only for Expert mode. It is possible to configure gains for the angle estimator (PllKp and PllKi) in the iSD, which is not possible with the MCEWizard.

Table 27 Angle Feedback

Item in iSD		Item in MCEWizard	
Flux Estimator	Flux estimator time constant	Q60	Flux Estimator Time Constant
Angle Estimator	Flux PLL compensator proportional gain manually	N/A	
	Flux PLL compensator integral gain manually	N/A	

4.2.5 PFC Control Configuration

This section provides configuration for PFC control items. Most of the items are newly introduced with the iSD. Please input a value for each item while referring to the help document in the iSD or [3].

4.2.6 Protection

4.2.6.1 PFC Protection

This section provides configuration for PFC control items. Most of the items are newly introduced with the iSD. Please input a value for each item while referring to the help document in the iSD or [3].

4.2.6.2 Motor Protection

Table 28 Motor Protection

Item in iSD		Item in MCEWizard	
Over Current Protection	Over current trip signal source selection	Q91	Gatekill Input Source
	Overcurrent comparator current trip level (peak)	Q93	Overcurrent Trip Level for Internal Gatekill Comparator
	Gatekill filter window	Q94	Gatekill Filter Time Constant
Voltage Protection	DC bus critical overvoltage fault level	Q49	DC Bus Critical Over Voltage Level
	Overvoltage fault enable	Q68	Enable DC Bus Overvoltage Fault
	DC bus overvoltage fault level	Q47	DC Bus Over-Voltage Level
	Undervoltage fault enable	Q69	Enable DC Bus Undervoltage Fault
	DC bus undervoltage fault level	Q49	DC Bus Under-Voltage Level
Flux Fault	Flux out-of-control fault enable	Q70	Flux PLL Out of Control Fault
	Flux out-of-control fault detection time	Q70	Flux PLL Out of Control Fault
NTC	Over temperature fault enable	Q71	Enable Over Temperature Fault
	NTC voltage threshold for an overtemp fault	Q72	NTC Over-temperature Voltage Threshold
	Rotor lock fault enable	Q73	Rotor Lock Protection Fault

Item in iSD		Item in MCEWizard	
Rotor Lock Protection	Rotor lock fault detection time	Q73	Rotor Lock Protection Fault
Phase Loss Protection	Phase loss fault enable	Q74	Enable Phase Loss Fault
Current Offset Protection	Current offset fault enable	N/A	

4.2.6.3 System Protection

This is the section configuration for the system protection. This section is new with the iSD and not shown in the MCEWizard.

4.3 Script

When the script function is enabled, users need to input the script code with the Script Editor in the iSD. It is necessary to split the script code for the Global.mcs, Script_Task0.mcs, and Script_Task1.mcs. It is also necessary to input the script execution information (execution period and steps) into project properties.

The MCE variable name is different between the MCEWizard/Designer and iSD, so it should be corrected according to the information in chapter 2.2. The difference in state machine handling between the MCE tools and iSD should be also considered, as described in chapter 3.

The Script Editor in the iSD offers a debugging function so that debugging is easier than using MCE tools.

For detail information for script, please refer to [3]and [5].

4.4 Evaluation

Motor and PFC evaluation is done by using the Dashboard. Please refer to [2] and [4] for how to use Dashboard function.

5 References

[1] Getting Started with iMOTION™ 2.0

[2] Getting Started with iMOTION™ Solution Designer

[3] Functional Reference Manual iMOTION™ Motor Control Engine

[4] iMOTION™ Solution Designer User Guide

[5] How to Use iMOTION™ Script Language

[6] iMOTION™ – MCE – Software Reference Manual

Revision history

Document version	Date of release	Description of changes
1.01	2023-05-31	Section 2.2.1 (How to Use Register IDs in Communication Commands) added.
1.0	2023-03-31	Initial Release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-05-31

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN-2023-02