Handling multiple motor/PFC parameters with iMOTION™

A guide for iMOTION™, MCEWizard, MCEDesigner and iMOTION™ Solution Designer

About this document

Scope and purpose

iMOTION™ is a turnkey yet versatile solution for efficient control of variable speed motors and PFC optionally. We explain how to set up one iMOTION™ to handle multiple parameter sets for motors and PFC optionally.

Intended audience

This application note is targeting engineers developing variable speed motor control drives using iMOTION™.

Table of contents

1 Introduction

iMOTION™ offers users greater control over permanent magnet motors by integrating an advanced software implementation in various hardware form factors. iMOTION™ comes with two development tools: the MCEWizard and the MCEDesigner. The MCEWizard is a GUI tool that allows users to enter motor or PFC parameters to control a motor and PFC optionally such as motor stator inductance and resistance, back electromotive Force, and motor speed limits.

For certain applications, users may need to utilize the same iMOTION™ control solution to run either different motors or the same motor with PFC, optionally under different load conditions. As an example, one iMOTION™ solution should be able to run a compressor under the right cooling profile in any of three refrigerator models of different volumetric capacities. Separately, a fan manufacturer may use the same iMOTION™ solution to control three motor models of varying power ratings and fan blade sizes.

In this application note, we will describe how to set up multiple parameter sets for motors and PFC optionally in order to support controlling various motors and hardware configurations with one solution. We will also describe how to set up multiple parameter sets by using new development tool iMOTION™ Solution Designer (iSD).

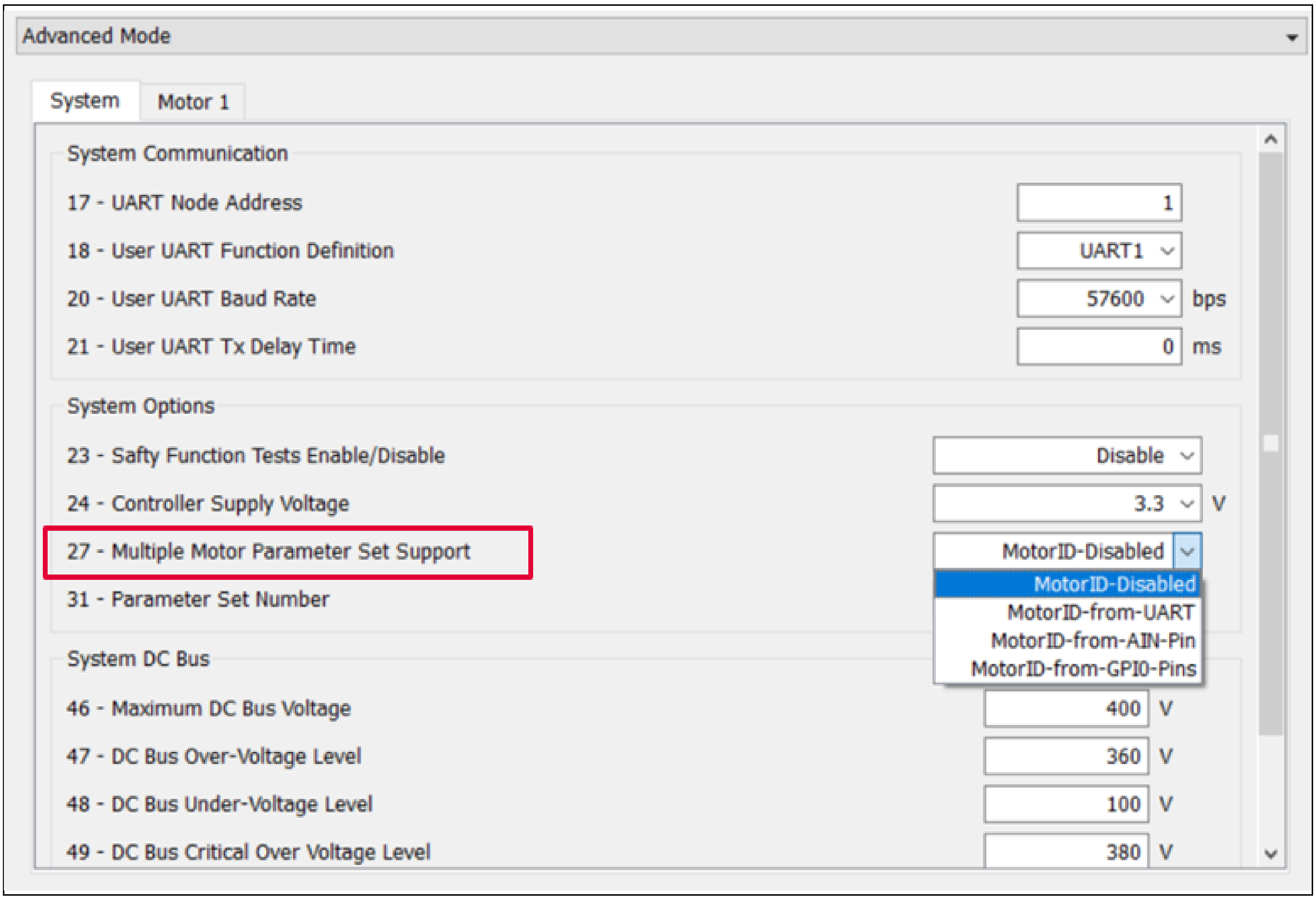
2 Multiple Parameter Handling by MCEWizard/Designer

2.1 Multiple-parameter block definition

2.1.1 MCEWizard configuration

In this application note, we look at an example of running four different motor / PFC and hardware configurations with a single iMOTION™ controller. We define each motor / PFC & hardware configuration with a set of parameters. In this example, we will have four parameter sets, from which our iMOTION™ controller can select.

By default, only one set of parameters is stored in FLASH and loaded into RAM of an iMOTION™ device, and the ability to select different parameter sets is disabled in the MCEWizard. We can enable multiple-parameter set support via the *Multiple Motor Parameter Set Support* option in the MCEWizard, as shown in **Error! Reference source not found.**.



System Communication

17 - UART Node Address: 1

18 - User UART Function Definition: UART1

20 - User UART Baud Rate: 57600 bps

21 - User UART Tx Delay Time: 0 ms

System Options

23 - Safty Function Tests Enable/Disable: Disable

24 - Controller Supply Voltage: 3.3 V

27 - Multiple Motor Parameter Set Support: MotorID-Disabled

31 - Parameter Set Number

System DC Bus

46 - Maximum DC Bus Voltage: 400 V

47 - DC Bus Over-Voltage Level: 360 V

48 - DC Bus Under-Voltage Level: 100 V

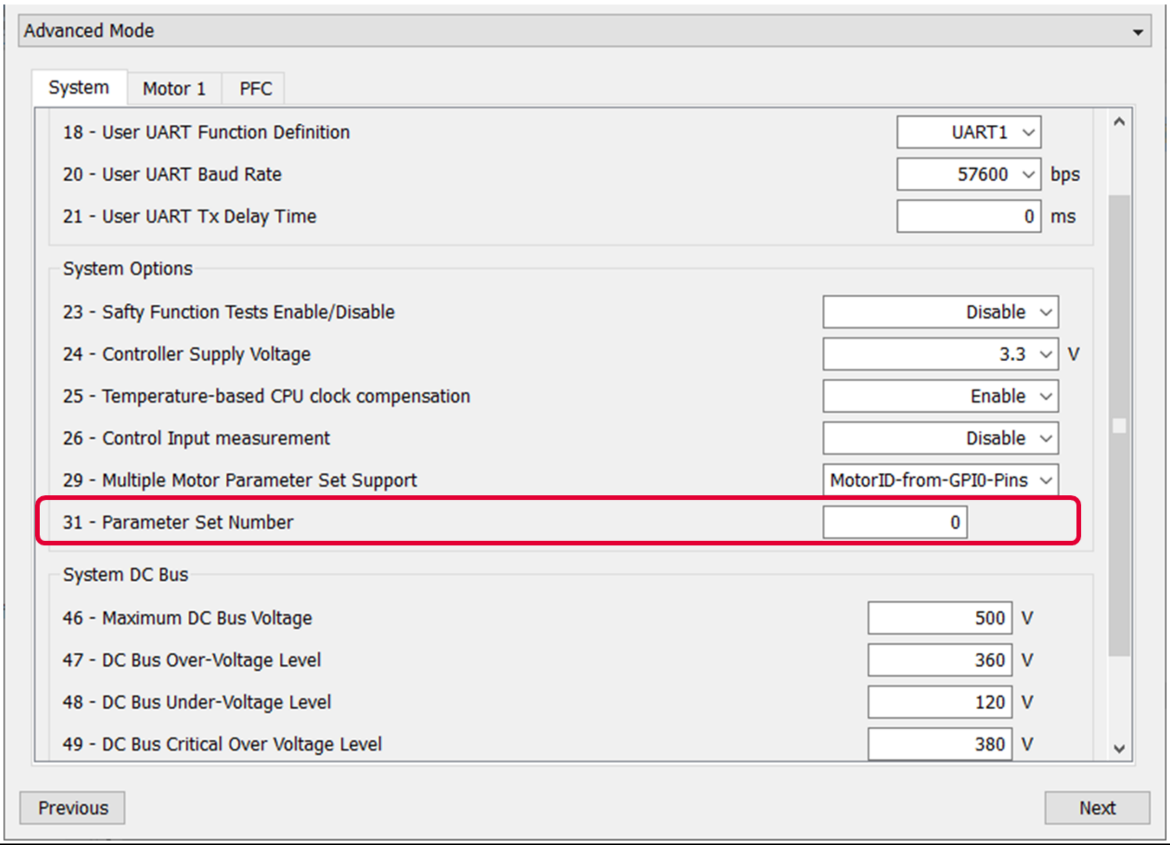
49 - DC Bus Critical Over Voltage Level: 380 V

Figure 1 Multiple Motor Parameter Set Support in MCEWizard

Users can use one of the following ways to tell MCE the desired parameter set number upon start-up:

- User UART
- PAR0 / PAR1 / PAR2 / PAR3 (GPIO) pins
- PARAM (AIN) pin
- Direct select

Each parameter set can be assigned with a unique parameter set number that can be specified in the MCEWizard. When the feature is enabled, the iMOTION™ firmware will load the parameter set with the matching parameter set number. To set up this ID, choose a number to enter in the “Parameter Set Number” section for the set of parameters entered in the MCEWizard, as shown in **Error! Reference source not found.**. Detail setting will be described in section 2.2.1.



18 - User UART Function Definition: UART1
20 - User UART Baud Rate: 57600 bps
21 - User UART Tx Delay Time: 0 ms

System Options
23 - Safty Function Tests Enable/Disable: Disable
24 - Controller Supply Voltage: 3.3 V
25 - Temperature-based CPU clock compensation: Enable
26 - Control Input measurement: Disable
29 - Multiple Motor Parameter Set Support: MotorID-from-GPIO-Pins
31 - Parameter Set Number: 0

System DC Bus
46 - Maximum DC Bus Voltage: 500 V
47 - DC Bus Over-Voltage Level: 360 V
48 - DC Bus Under-Voltage Level: 120 V
49 - DC Bus Critical Over Voltage Level: 380 V

Figure 2 Parameter Set Number in MCEWizard

When working with different setups and iMOTION™ devices, the MCEWizard will use 4 kilobytes of flash memory to store various control parameter data. There are 16 parameter blocks, each of them being 256 bytes in size. A maximum of 15 parameter sets can be programmed in order to support different motor types or hardware, and one block is reserved to store system parameters.

For a system with only a motor control function, each parameter set will take one parameter block. In this case, the valid parameter set IDs can range from 0 to 14.

For a system with motor control and PFC functions, each parameter set will take two consecutive parameter blocks. The motor control parameter set will be stored in the selected parameter block, and the PFC parameter set will be stored in the subsequent parameter block. In this case, the valid parameter set IDs are 0, 2, 4, 6, 8, 10, and 12, with only 7 different configurations made available.

The MCEWizard output (*.txt) that contains the parameter values and the specified parameter set number will be generated. Section 3 describes how to combine all these parameter files into one .ldf file for easy loading into the iMOTION™ device with the MCEDesigner.

Note: The available parameter selection methods depend on the iMOTION™ device used. IMM101T-046, for example, only supports Direct and User UART selection methods. IMC101T-T038 supports all four methods.

In any case, when one of the four selection methods are specified, the register ParPageConfp is updated automatically by the MCEWizard, and enables one of the four solutions. **Error! Reference source not found.** shows the full parameter loading procedure for iMOTION™ devices.

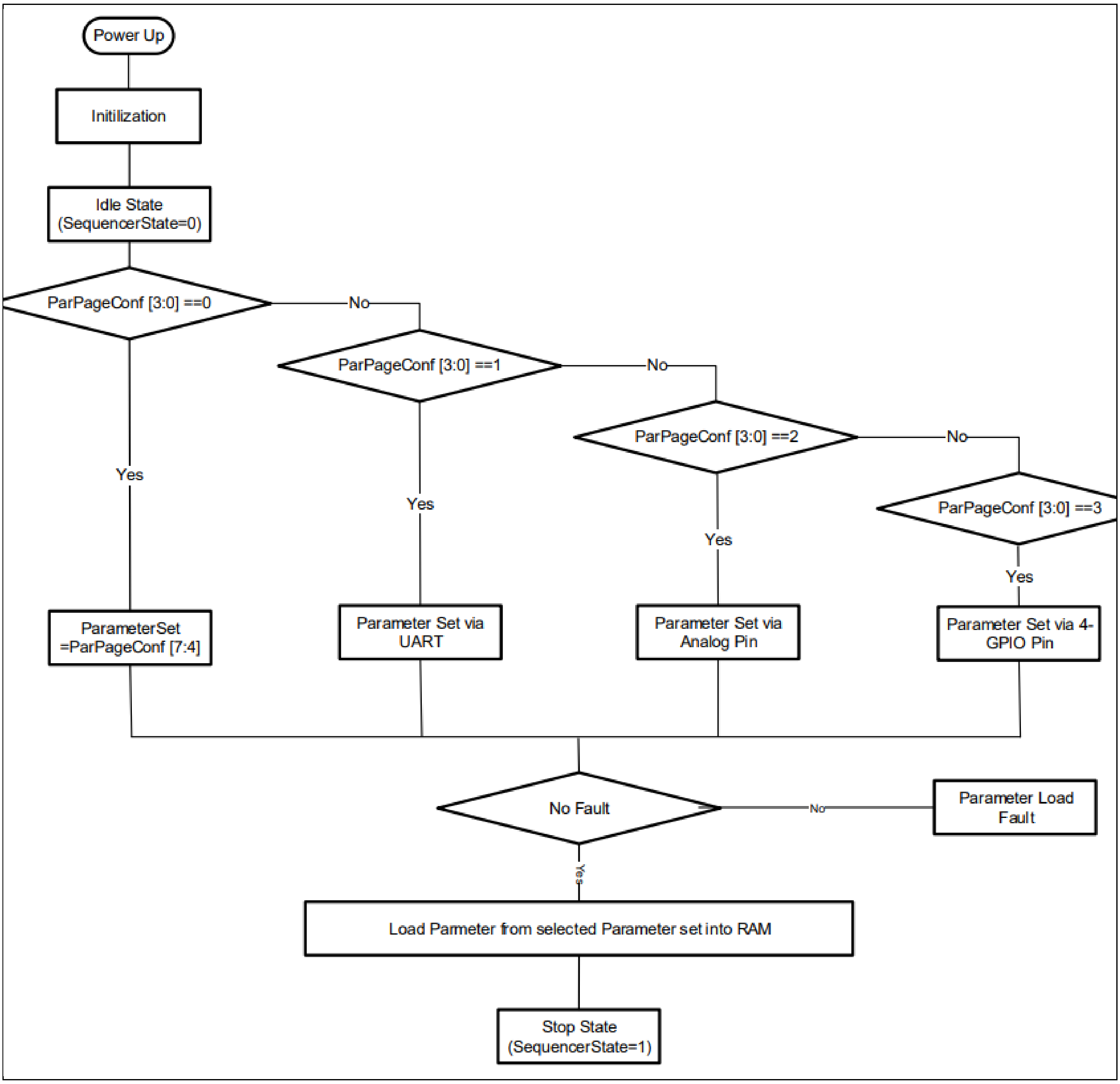

Power Up
Initilization
Idle State (SequencerState=0)
ParPageConf [3:0] ==0
No
ParPageConf [3:0] ==1
No
ParPageConf [3:0] ==2
No
ParPageConf [3:0] ==3
Yes
Yes
Yes
Yes
ParameterSet =ParPageConf [7:4]
Parameter Set via UART
Parameter Set via Analog Pin
Parameter Set via 4-GPIO Pin
No Fault
No
Parameter Load Fault
Yes
Load Parmeter from selected Parameter set into RAM
Stop State (SequencerState=1)

Figure 3 iMOTION™ parameter loading procedure

ParPageConf is a 16-bit register that defines the parameter-block selection method and default parameter block to upload. It is defined as follows:

[3:0] Parameter page selection:

0- No selection

1- Parameter page selection via UART

2- Parameter page selection via analog Input

3- Parameter page selection via digital Input

[7:4] Default parameter page number

[15:8] Reserved

Section 2.2 describes in detail how to use this very important register in order to combine all .txt files into one .txt file to be loaded into the MCEDesigner.

2.1.2 Motor ID using UART

Specific Universal Asyunchronous Reciver/Transmitter (UART) messages are defined to load the parameter block from flash to RAM, and save the parameter set from RAM to flash. 'Load parameter' command = 0x20 loads all parameters of one block into the dedicated RAM locations. If this method is chosen, ParPageConf [3:0] = 1.

The 'Load parameter' command loads the parameters from the specified parameter set stored in FLASH into the RAM. The valid range of the parameter set number is 0-14 when PFC function is disabled and 0, 2, 4, 6, 8, 10, or 12 during PFC function is enabled. If an odd number (e.g. 1) is selected when the PFC function is enabled, parameter load fault occurs and the specified parameter set is not loaded into RAM.

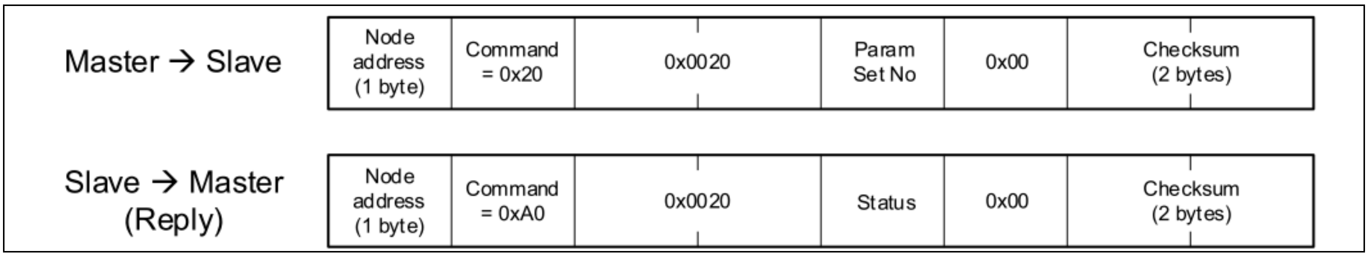


Figure 4 UART Load parameter command = 0x20, Data Word0 = 0x0020

The 'Save parameter' command erases the selected parameter set in FLASH first and saves the parameters of the specified App ID to this parameter set in FLASH. The valid range of the parameter set number is 0-14 when PFC function is disabled, and 0, 2, 4, 6, 8, 10, or 12 when PFC function is enabled. If an odd number (e.g. 1) is selected when the PFC function is enabled, parameter load fault occurs and the specified parameter set is not loaded to RAM. The valid App ID value is 1 or 3. In the reply frame, data 0 word contains the value of 'Status' (0: success, 1: fail, 2: parameter set number not supported).

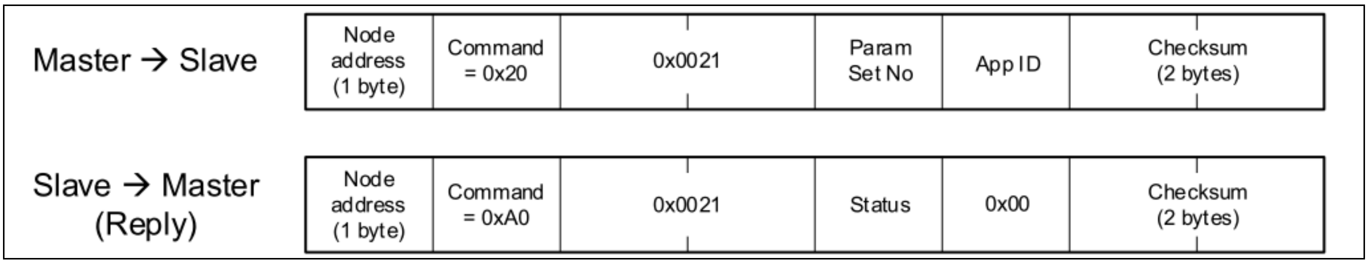


Figure 5 UART Save parameter command = 0x21, Data Word0 = 0x0021

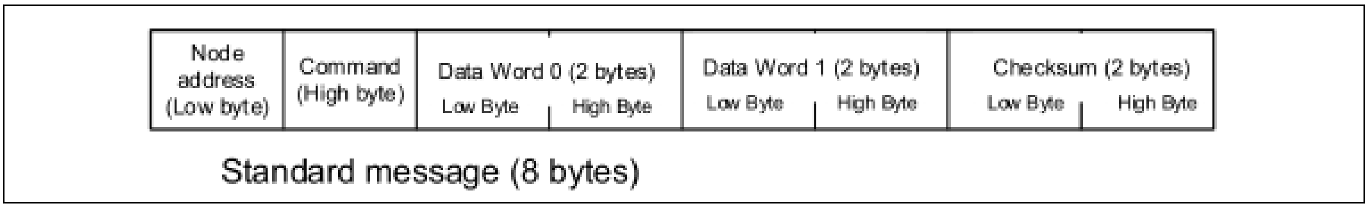


Figure 6 UART Data Frame

Here is an analysis of this data frame for loading or saving parameters:

- **Node address** is the first byte in a data frame. It is designed to allow one master to control multiple slaves in the same network. Each slave node has its unique node ID. The slave only acknowledges and responds to the message with the same ID. Two broadcast addresses (0x00 and 0xFF) are defined for different usages. If a message is received with address=0x00, all the slaves execute the command, but will not send a reply to the master. This is useful in a multiple-slave network, and the master needs to control all the slaves at the same time, like turn on all the motors by sending only one message. If a frame with address=0xFF is received, the slave will execute the command and send a reply to the master. This is useful in a 1-to-1 configuration when the master does not know or does not need to know the slave node address.
- **Command** is the specific UART command to load or save a created set of parameters. Command = 0x20
- **Data Word 0** describes if the process is loading parameter 0x0020 or saving parameter 0x0021
- **Dataword 1** LSB = 0x00 and MSB = chosen parameter set number corresponding to the correct one from among 15.
- **Checksum** is 16-bit format that shall be calculated as below: [Command: Node address] + Data Word 0 + Data Word 1 + Checksum = 0x0000

Here is an example of where to use UART command to load parameter block 3:

Input : Node address = 1, Command = 0x20, Data Word 0 = 0020, Data Word 1 = 0x0003

[Command:Node address] = 0x2001

Checksum = -1 x (0x2001 + 0x0021 + 0x0003) = 0xDFDB

UART message to send to iMOTION™ controller to load parameter block 3:

01 20 20 00 03 00 DC DF

2.1.3 Motor ID using GPIO pins

This parameter block is selected based on the input of the four General Purpose Input/Output (GPIO) pins. The GPIO pins used for parameter set selection are named as “PAR0,” “PAR1,” “PAR2”, and “PAR3.” Mapping between parameter page selections based on GPIO pins are listed in Table 1. If this method is chosen, ParPageConf [3:0] = 3.

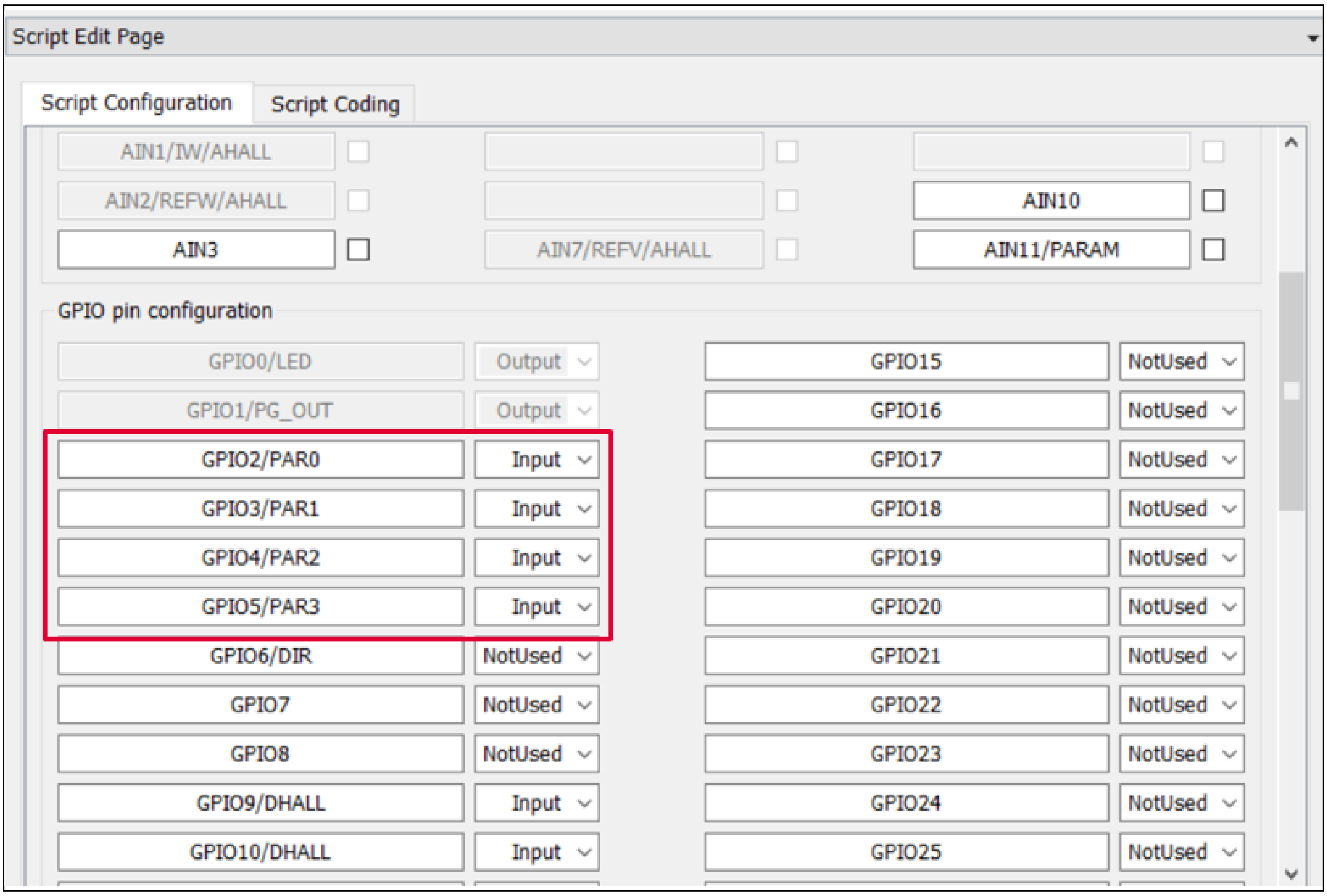
Any number from 0 to 14 can be selected when PFC function is disabled. Only even numbers equal or smaller than 12 (0, 2, 4, 6, 8, 10, or 12) can be selected when PFC function is enabled. If any other number is selected when PFC is enabled, parameter load fault occurs and the specified parameter set is not loaded into RAM.

Table 1 Parameter page slection with GPIOs

GPIO Input				Parameter block
PAR3	PAR2	PAR1	PAR0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4

GPIO Input				Parameter block
PAR3	PAR2	PAR1	PAR0	
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14

When the GPIO selection method is not being used, these GPIOs are available to be used in script code as shown in Figure 7.



Script Edit Page

Script Configuration | Script Coding

AIN1/IW/AHALL	□		□		□
AIN2/REFW/AHALL	□		□	AIN10	□
AIN3	□	AIN7/REFV/AHALL	□	AIN11/PARAM	□

GPIO pin configuration

GPIO0/LED	Output	GPIO15	NotUsed
GPIO1/PG_OUT	Output	GPIO16	NotUsed
GPIO2/PAR0	Input	GPIO17	NotUsed
GPIO3/PAR1	Input	GPIO18	NotUsed
GPIO4/PAR2	Input	GPIO19	NotUsed
GPIO5/PAR3	Input	GPIO20	NotUsed
GPIO6/DIR	NotUsed	GPIO21	NotUsed
GPIO7	NotUsed	GPIO22	NotUsed
GPIO8	NotUsed	GPIO23	NotUsed
GPIO9/DHALL	Input	GPIO24	NotUsed
GPIO10/DHALL	Input	GPIO25	NotUsed

Figure 7 GPIO for Block parameter update in MCEWizard

2.1.4 MotorID using analog input

If this method is chosen, ParPageConf [3:0] = 2. Parameter block is selected based on the analog input voltage level of the 'PARAM' pin of an iMOTION™ device. Mapping between parameter page and analog input voltage is shown below:

$$ParameterBlock = Integer\left\{\left(\frac{AnalogInput}{Vadcref} \times 15\right)\right\}$$

For example, if analog input at PARAM pin is 1.2 V and V_{adcref}=3.3 V, parameter block number =5.

Any number from 0 to 14 can be selected when PFC function is disabled. Only even numbers equal or smaller than 12 (0, 2, 4, 6, 8, 10, or 12) can be selected when PFC function is enabled. If any other number is selected when PFC is enabled, parameter load fault occurs and the specified parameter set is not loaded into RAM.

2.1.5 Motor ID using Direct Select

The parameters' block selection is based on the "ParPageConf [7:4]" parameter bit field value. This requires a manual bit field update in the MCEDesigner or parameter file. If we want to load parameter block 3, set ParPageConf as follows:

Direct Select : : ParPageConf[3:0] =0 and *Direct Select : : ParPageConf[7:4] = 3*

2.2 Multiple-parameter application example

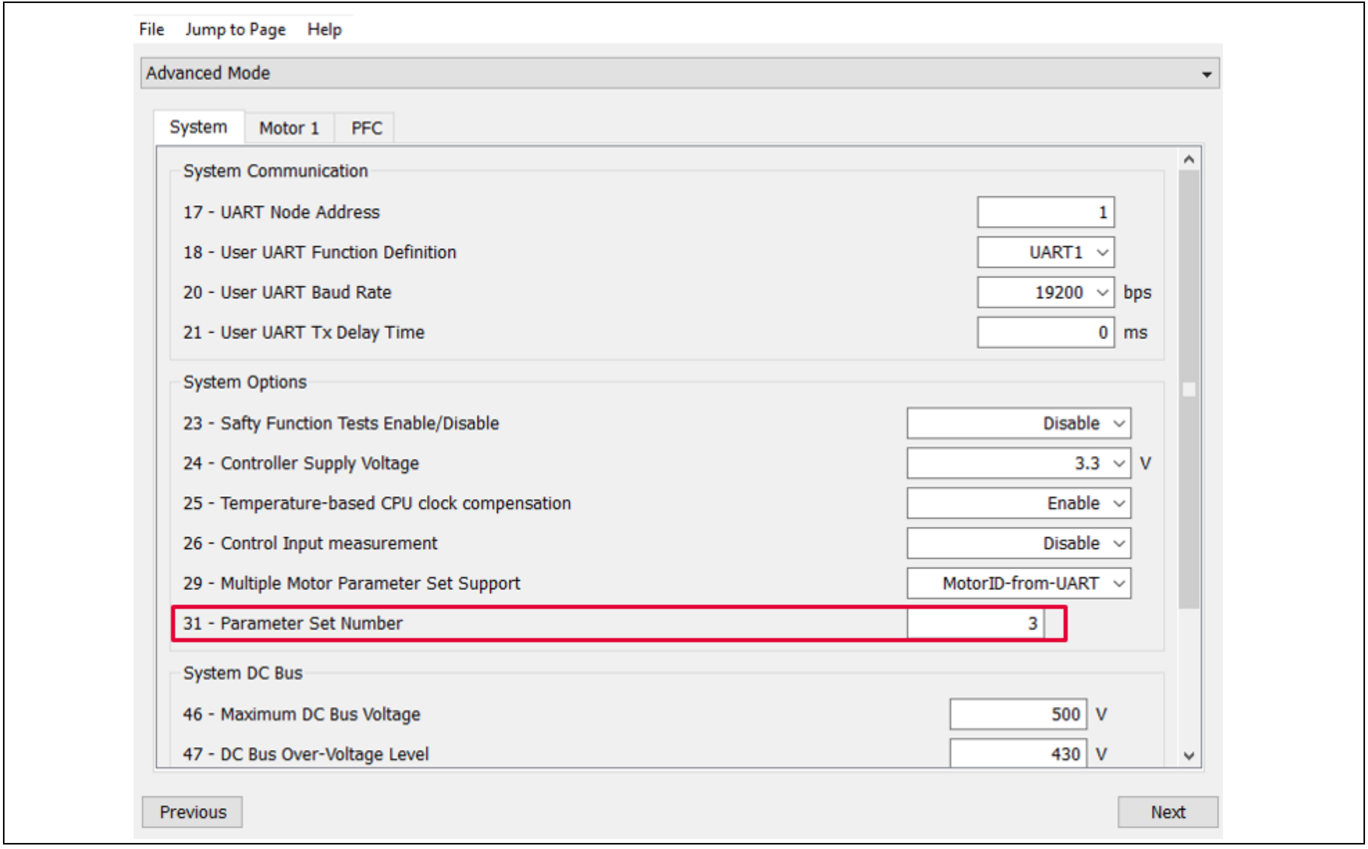
In the previous section, we demonstrated how to set up the MCEWizard and how to use the four different ways of selecting a parameter set. In this chapter, we describe a method of combining and loading four different parameter sets into the controller using the MCEWizard and the MCEDesigner.

2.2.1 Prepapration of MCEWizard .txt parameter files

As there are four different configurations to select from, via UART with a single iMOTION™ device, we first need to generate one parameter file for each configuration. We use the MCEWizard to create all the parameter files. Each file and parameter set are defined by a unique *Parameter Set Number.*

2.2.1.1 PFC function: disabled

In this example, we enter 0, 1, 2, and 3 as the *Parameter Set Number* for each configuration.



System Communication

17 - UART Node Address: 1

18 - User UART Function Definition: UART1

20 - User UART Baud Rate: 19200 bps

21 - User UART Tx Delay Time: 0 ms

System Options

23 - Safty Function Tests Enable/Disable: Disable

24 - Controller Supply Voltage: 3.3 V

25 - Temperature-based CPU clock compensation: Enable

26 - Control Input measurement: Disable

29 - Multiple Motor Parameter Set Support: MotorID-from-UART

31 - Parameter Set Number: 3

System DC Bus

46 - Maximum DC Bus Voltage: 500 V

47 - DC Bus Over-Voltage Level: 430 V

Figure 8 Example for parameter block 3 in MCEWizard

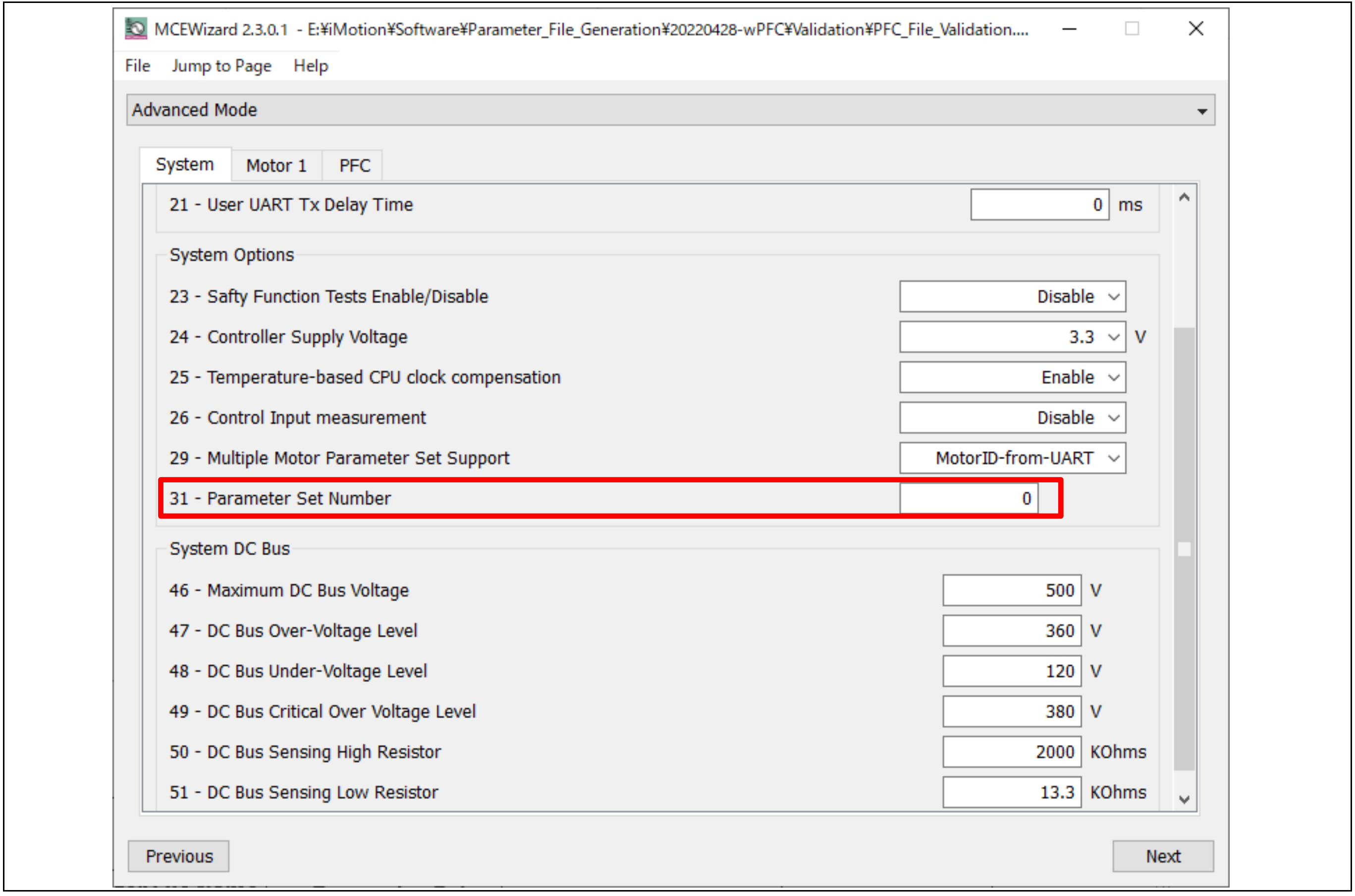
This should create four different .txt files once compiled with the main identification registers as follows:

Table 2 Example of four different MCEWizard .txt generated for four different setups

Parameter File Name (Example)	Parameter Set Number	Set Support	##MOTOR1_REGS	ParPageConf
ParameterFile0.txt	0	Motor-ID from UART	0	2049
ParameterFile1.txt	1		1	2065
ParameterFile2.txt	2		2	2081
ParameterFile3.txt	3		3	2097

2.2.1.2 PFC function: enabled

In this example, we enter 0, 2, 4, and 6 as the *Parameter Set Number* for each configuration.



MCEWizard 2.3.0.1 - E:¥iMotion¥Software¥Parameter_File_Generation¥20220428-wPFC¥Validation¥PFC_File_Validation....

21 - User UART Tx Delay Time: 0 ms

System Options

23 - Safty Function Tests Enable/Disable: Disable

24 - Controller Supply Voltage: 3.3 V

25 - Temperature-based CPU clock compensation: Enable

26 - Control Input measurement: Disable

29 - Multiple Motor Parameter Set Support: MotorID-from-UART

31 - Parameter Set Number: 0

System DC Bus

46 - Maximum DC Bus Voltage: 500 V

47 - DC Bus Over-Voltage Level: 360 V

48 - DC Bus Under-Voltage Level: 120 V

49 - DC Bus Critical Over Voltage Level: 380 V

50 - DC Bus Sensing High Resistor: 2000 KOhms

51 - DC Bus Sensing Low Resistor: 13.3 KOhms

Figure 9 Example for parameter block 0 in MCEWizard

This should create four differents .txt files once compiled with the main identification registers as follows:

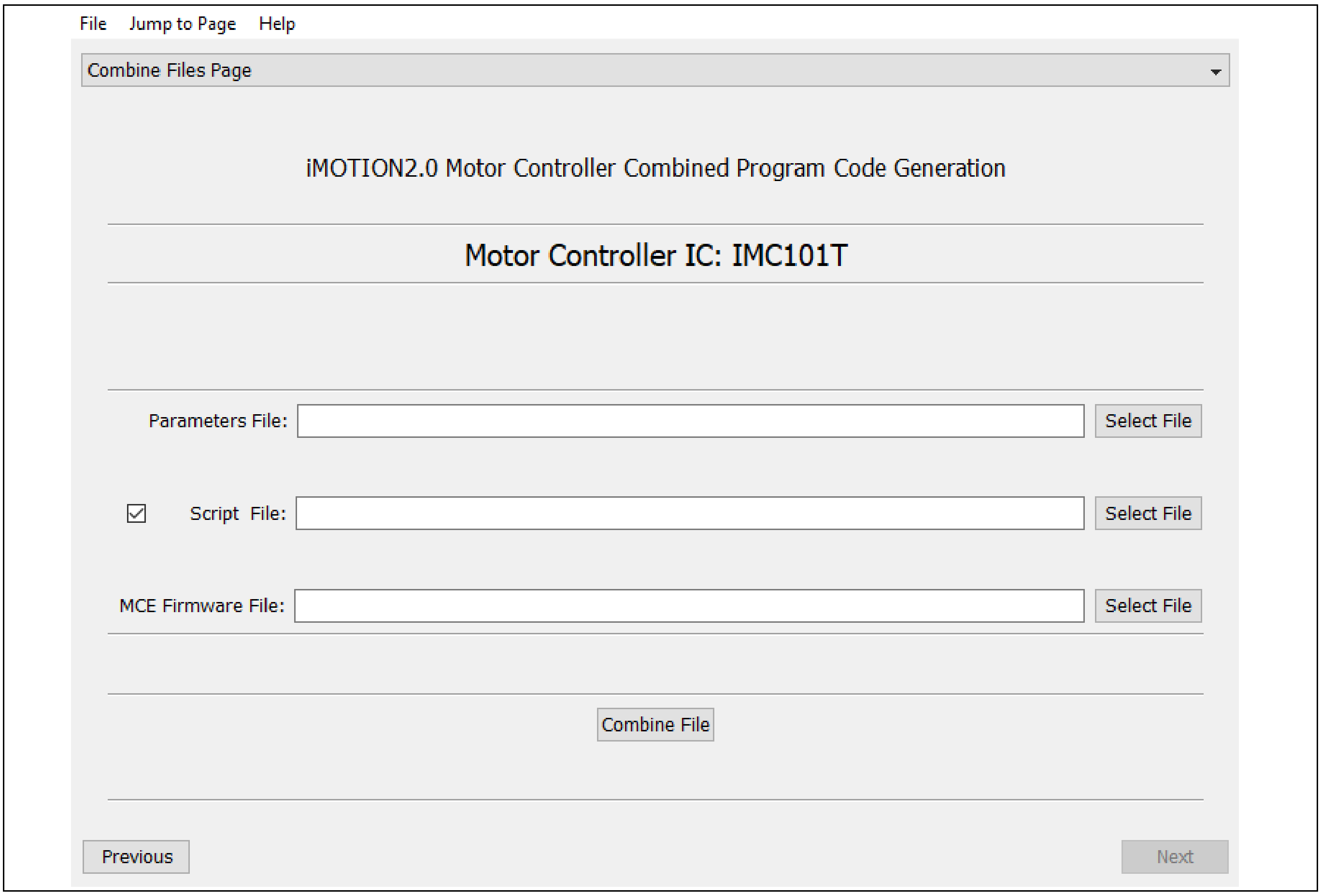
Table 3 Example of four different MCEWizard .txt generated for four different setups

Parameter File Name (Example)	Parameter Set Number	Set Support	##MOTOR1_REGS	ParPageConf
ParameterFile0.txt	0	Motor-ID from UART	0	2049
ParameterFile1.txt	1		1	2065
ParameterFile2.txt	2		2	2081
ParameterFile3.txt	3		3	2097

Now that there are four different .txt files with proper identification, users can combine them into a single file that can be uploaded to the MCEDesigner.

2.2.2 Combine .txt parameter files into a single .ldf file

The first thing to do is to use the MCEWizard “Combine Files Page” functionality:



File Jump to Page Help

Combine Files Page

iMOTION2.0 Motor Controller Combined Program Code Generation

Motor Controller IC: IMC101T

Parameters File: Select File

☑ Script File: Select File

MCE Firmware File: Select File

Combine File

Previous Next

Figure 10 Combine Files Page in MCEWizard

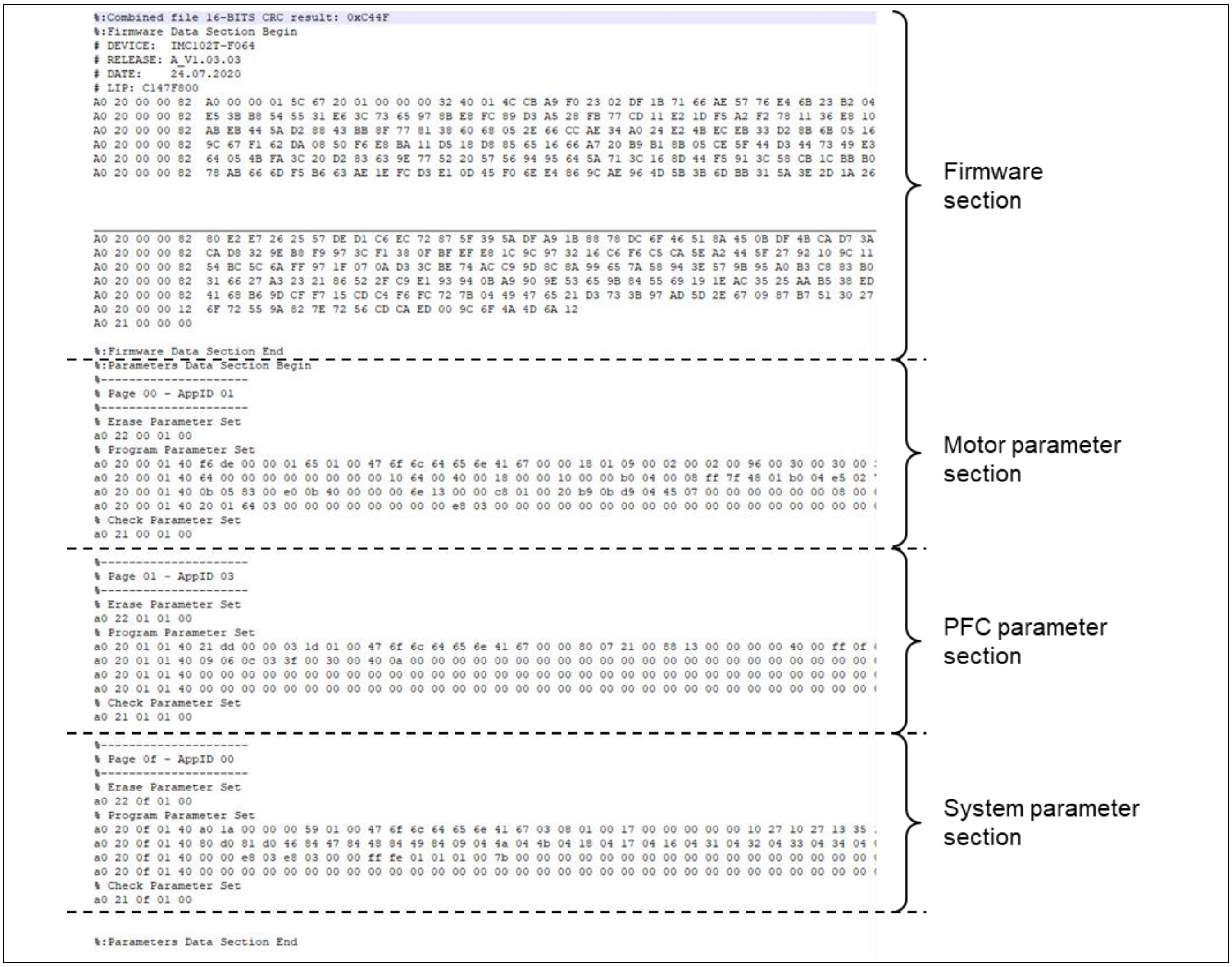
1- For the Parameters File, choose the first one generated. In our example, it will be ParameterFile0.txt

2- For the MCE Firmware File, choose the corresponding .ldf file. In our example, it will be IMC101T-046

3- Create a Combine File. In this example, we will call it Parameter0.ldf

Repeat steps 1 to 3 for the number of parameters you desire. In our example, we will do this exercise four times.

Figure 11 shows an example of a combined .ldf file that is generated by the operation above. This file consists of three sections when PFC is disabled, and four sections when PFC is enabled.



```
%:Combined file 16-BITS CRC result: 0xC44F
%:Firmware Data Section Begin
# DEVICE:   IMC102T-F064
# RELEASE: A_V1.03.03
# DATE:     24.07.2020
# LIP: C147F800
A0 20 00 00 82  A0 00 00 01 5C 67 20 01 00 00 00 32 40 01 4C CB A9 F0 23 02 DF 1B 71 66 AE 57 76 E4 6B 23 B2 04
A0 20 00 00 82  E5 3B B8 54 55 31 E6 3C 73 65 97 8B E8 FC 89 D3 A5 28 FB 77 CD 11 E2 1D F5 A2 F2 78 11 36 E8 10
A0 20 00 00 82  AB EB 44 5A D2 88 43 BB 8F 77 81 38 60 68 05 2E 66 CC AE 34 A0 24 E2 4B EC EB 33 D2 8B 6B 05 16
A0 20 00 00 82  9C 67 F1 62 DA 08 50 F6 E8 BA 11 D5 18 D8 85 65 16 66 A7 20 B9 B1 8B 05 CE 5F 44 D3 44 73 49 E3
A0 20 00 00 82  64 05 4B FA 3C 20 D2 83 63 9E 77 52 20 57 56 94 95 64 5A 71 3C 16 8D 44 F5 91 3C 58 CB 1C BB B0
A0 20 00 00 82  78 AB 66 6D F5 B6 63 AE 1E FC D3 E1 0D 45 F0 6E E4 86 9C AE 96 4D 5B 3B 6D BB 31 5A 3E 2D 1A 26


A0 20 00 00 82  80 E2 E7 26 25 57 DE D1 C6 EC 72 87 5F 39 5A DF A9 1B 88 78 DC 6F 46 51 8A 45 0B DF 4B CA D7 3A
A0 20 00 00 82  CA D8 32 9E B8 F9 97 3C F1 38 0F BF EF E8 1C 9C 97 32 16 C6 F6 C5 CA 5E A2 44 5F 27 92 10 9C 11
A0 20 00 00 82  54 BC 5C 6A FF 97 1F 07 0A D3 3C BE 74 AC C9 9D 8C 8A 99 65 7A 58 94 3E 57 9B 95 A0 B3 C8 83 B0
A0 20 00 00 82  31 66 27 A3 23 21 86 52 2F C9 E1 93 94 0B A9 90 9E 53 65 9B 84 55 69 19 1E AC 35 25 AA B5 38 ED
A0 20 00 00 82  41 68 B6 9D CF F7 15 CD C4 F6 FC 72 7B 04 49 47 65 21 D3 73 3B 97 AD 5D 2E 67 09 87 B7 51 30 27
A0 20 00 00 12  6F 72 55 9A 82 7E 72 56 CD CA ED 00 9C 6F 4A 4D 6A 12
A0 21 00 00 00

%:Firmware Data Section End
%:Parameters Data Section Begin
%---------------------
% Page 00 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 00 01 00
% Program Parameter Set
a0 20 00 01 40 f6 de 00 00 01 65 01 00 47 6f 6c 64 65 6e 41 67 00 00 18 01 09 00 02 00 02 00 96 00 30 00 30 00
a0 20 00 01 40 64 00 00 00 00 00 00 00 00 10 64 00 40 00 18 00 00 10 00 00 b0 04 00 08 ff 7f 48 01 b0 04 e5 02
a0 20 00 01 40 0b 05 83 00 e0 0b 40 00 00 00 6e 13 00 00 c8 01 00 20 b9 0b d9 04 45 07 00 00 00 00 00 00 08 00
a0 20 00 01 40 20 01 64 03 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 00 01 00
%---------------------
% Page 01 - AppID 03
%---------------------
% Erase Parameter Set
a0 22 01 01 00
% Program Parameter Set
a0 20 01 01 40 21 dd 00 00 03 1d 01 00 47 6f 6c 64 65 6e 41 67 00 00 80 07 21 00 88 13 00 00 00 00 40 00 ff 0f
a0 20 01 01 40 09 06 0c 03 3f 00 30 00 40 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 01 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 01 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 01 01 00
%---------------------
% Page 0f - AppID 00
%---------------------
% Erase Parameter Set
a0 22 0f 01 00
% Program Parameter Set
a0 20 0f 01 40 a0 1a 00 00 00 59 01 00 47 6f 6c 64 65 6e 41 67 03 08 01 00 17 00 00 00 00 00 10 27 10 27 13 35
a0 20 0f 01 40 80 d0 81 d0 46 84 47 84 48 84 49 84 09 04 4a 04 4b 04 18 04 17 04 16 04 31 04 32 04 33 04 34 04
a0 20 0f 01 40 00 00 e8 03 e8 03 00 00 ff fe 01 01 01 00 7b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 0f 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 0f 01 00

%:Parameters Data Section End
```

Figure 11 Example of combined ldf file

Use the following steps to combine multiple .ldf files:

(1) Extract motor parameter section (and PFC prameter section, if necessary)

(2) Change Page number

The page number is defined in the line starting from "a0 22" after " % Erase Parameter Set" and "a0 21" after " % Check Parameter Set". The third value in this line is the page number.

If PFC is disabled, the page number will start from 0 and is incremented. If PFC is enabled, the page number will be an even number starting from 0 for motor parameter section, and an odd number starting from 1 for PFC parameter section.

(3) Repeat (1) and (2) for all combined .ldf files.

(4) Combine all modified .ldf files.

(5) Add Firmware section in the beginning of combined .ldf file made in (4), and add system parameter section at the end of the combined .ldf file made in (4).

VBA function to combine mulitple .ldf combined files into a single usable one in MCEWizard

The procedure above would be time consuming if done manually. That is why a simple Excel VBA function is provided, which can extract all the useful information from the .ldf files and combine it into one final .ldf file. The final file will have all the different parameter block sets, and can easily be uploaded with the MCEDesigner.

VBA code is shown in Code Listing 1 to Code Listing 3. Please copy and paste all codes and create a macro "Make_Combined_LDF" in MS Excel to run it.

Code Listing 1 Main Function Make_Combined_LDF()

```
Sub Make_Combined_LDF()
  Dim ChDir As String, f As Variant
  Dim i As Integer, n As Integer
  Dim rc As Integer
  Dim if_success As Boolean

  Fin = Application.GetOpenFilename(FileFilter:="LDF file(*.ldf),*.ldf", MultiSelect:=True)
  If VarType(Fin) = vbBoolean Then
    MsgBox ("Cencelled")
  Else
    If IsArray(Fin) Then
      n = UBound(Fin)
      if n > 15 Then
        MsgBox "# of files is more than 15.", vbExclamation
      Else
        Fout = Application.GetSaveAsFilename(FileFilter:="LDF file(*.ldf),*.ldf", Title:="Save combined LDF file")
        If Dir(Fout) <> "" Then
          rc = MsgBox("File exist. Overwrite?", vbYesNo + vbQuestion)
          If rc = vbNo Then
            MsgBox ("Quit the process")
            End
          End If
        End If
        ' Check PFC
        page_count = count_pages(Fin(1))
        If (page_count = 3) Then
          MsgBox ("PFC function enabled")
        Else
          MsgBox ("PFC function disabled")
        End If
        Open Fout For Output As #2
        For i = 1 To n
          if_success = Make_LDF(Fin(i), i)
          If if_success = False Then
            MsgBox ("Combined fine generation FAILED.")
            Exit For
          End If
        Next i
        Close #2
        If if_success = False Then
```

Code Listing 1 Main Function Make_Combined_LDF()

```
          MsgBox ("Combined fine generation FAILED.")
        Else
          MsgBox ("Combine complete!")
        End If
      End If
    Else
      MsgBox ("Selected file is only one and no process will be executed")
    End If
  End If
End Sub

Function Make_LDF(f As Variant, i As Integer) As Boolean

```

Code Listing 2 Function Make_LDF()

```
Function Make_LDF(f As Variant, i As Integer) As Boolean
  Dim Ftemp As String, s As String
  Dim file_operation_success As Boolean
  file_operation_success = True

  If page_count <> count_pages(f) Then
    MsgBox ("Both PFC enabled and disabled are included in the file list.")
    file_operation_success = False
  Else
    Open f For Input As #1
    Do Until EOF(1)
      Line Input #1, s
      If InStr(s, "Parameters Data Section Begin") Then
      Do
        Line Input #1, s
        Print #2, s
        If InStr(s, "% Check Parameter Set") Then
          Line Input #1, s
          Print #2, s
          Line Input #1, s
          Print #2, s
          If page_count = 3 Then
            Do
              Line Input #1, s
              Print #2, s
              If InStr(s, "% Check Parameter Set") Then
                Line Input #1, s
                Print #2, s
                Exit Do
              End If
            Loop
          End If
          If i = UBound(Fin) Then
            Do
```

Code Listing 2 Function Make_LDF()

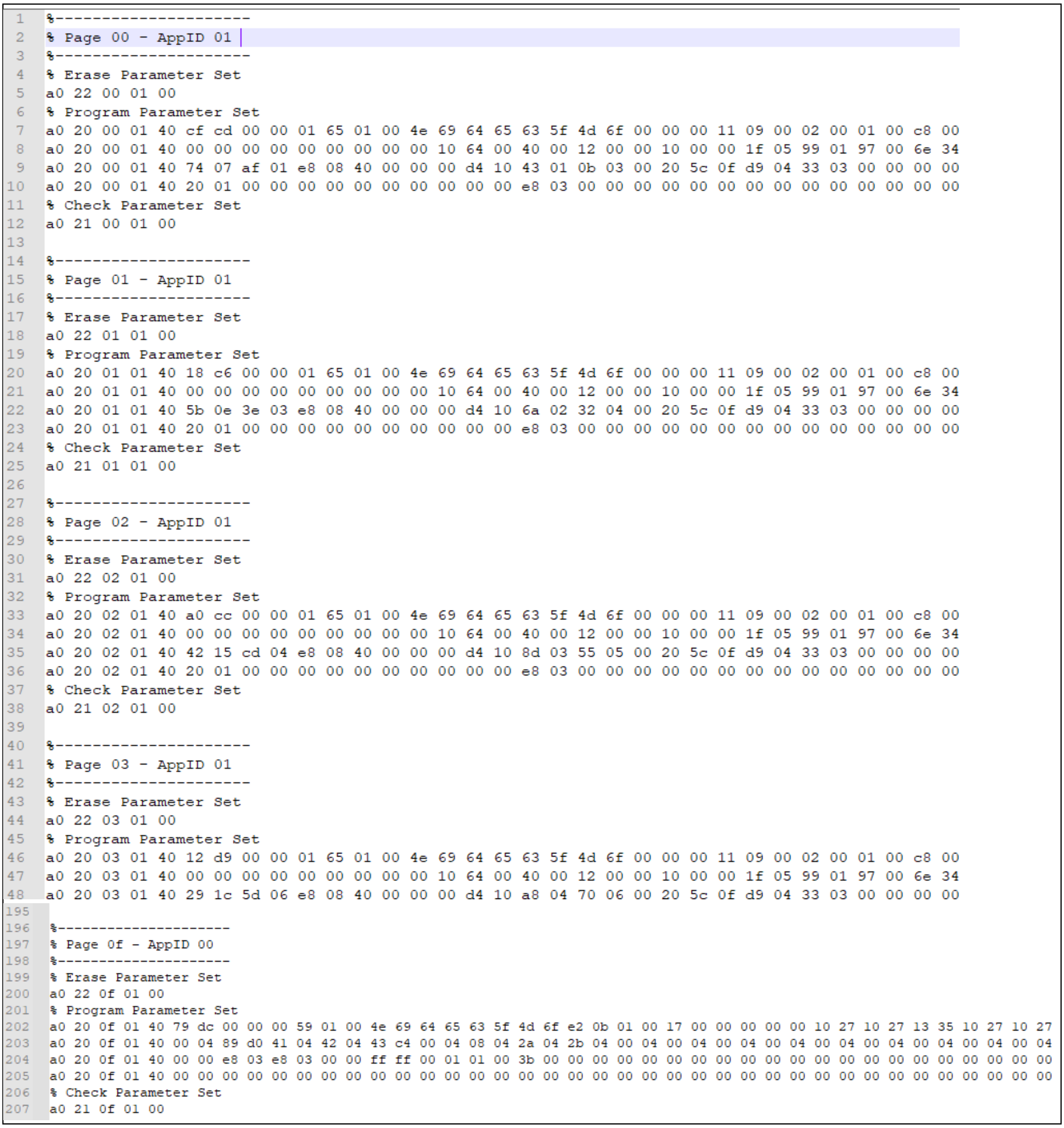
```
              Line Input #1, s
              Print #2, s
              If InStr(s, "% Check Parameter Set") Then
                Line Input #1, s
                Print #2, s
                Exit Do
              End If
              Loop
            End If
            Exit Do
          End If
        Loop
      End If
    Loop
  Close #1
  End If
  Make_LDF = file_operation_success
End Function
```

Code Listing 3 Function count_pages()

```
Function count_pages(f As Variant) As Integer
  Dim count As Integer
  Dim s As String

  count = 0
  Open f For Input As #1
  Do Until EOF(1)
    Line Input #1, s
    If InStr(s, "% Page") And InStr(s, "AppID") Then
      count = count + 1
    End If
  Loop
  Close #1
  count_pages = count
End Function
```

A macro containing this function can now be assigned to a button (`Make_Combined_LDF`, for example, to merge all the .ldf files into a single one that can then be uploaded to MCEDesigner.

When PFC function is disabled, the macro extracts the content of Page 00 - AppID 01 (Motor Control paramters) and Page 0f - AppID 00 (System Control parameters) from each .ldf file, and combines them into a new .ldf file containing all four parameter sets as well as the system control parameters. Figure 12 shows this newly combined .ldf file containing all the parameter sets:

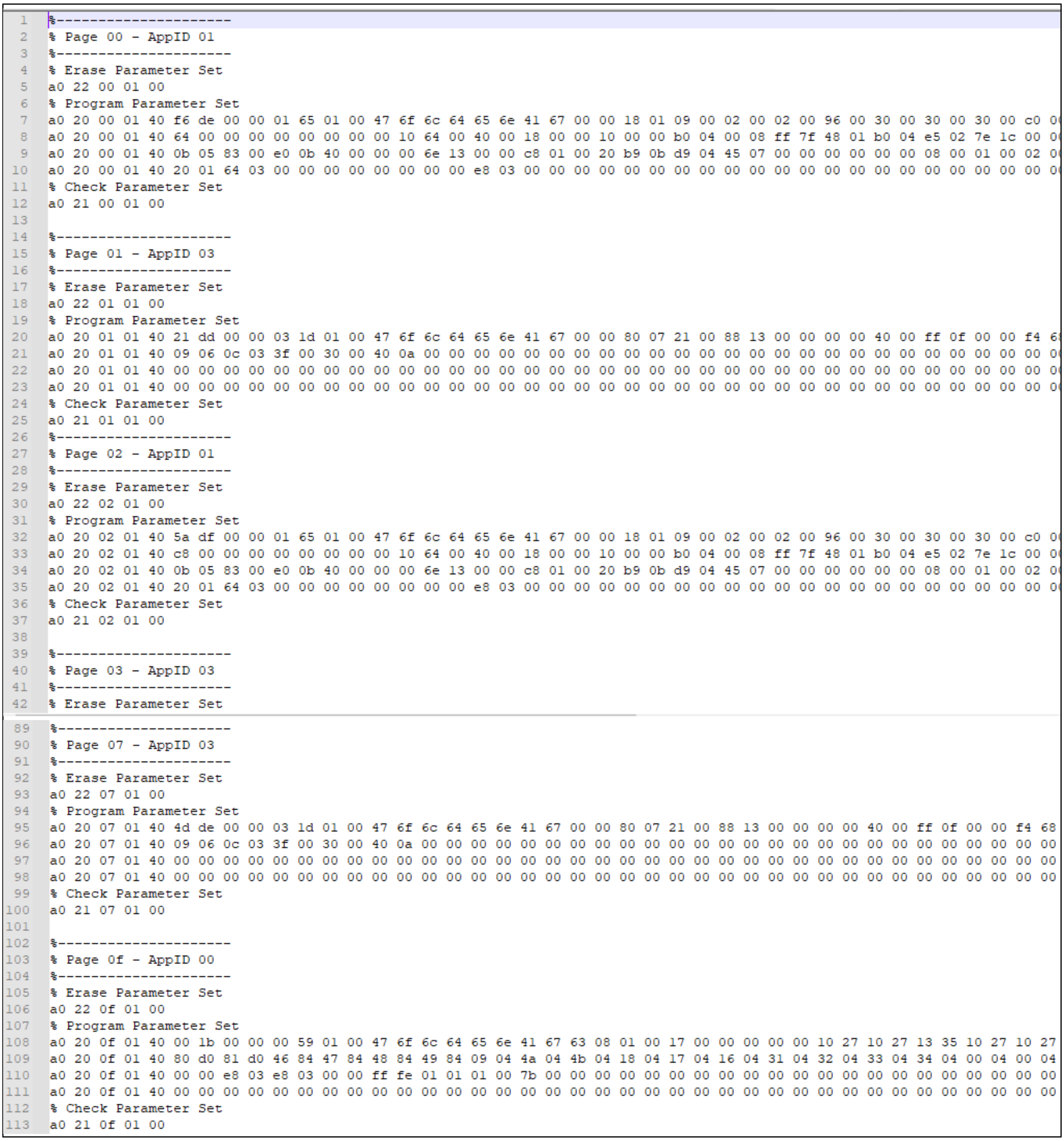


```
%---------------------
% Page 00 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 00 01 00
% Program Parameter Set
a0 20 00 01 40 cf cd 00 00 01 65 01 00 4e 69 64 65 63 5f 4d 6f 00 00 00 11 09 00 02 00 01 00 c8 00
a0 20 00 01 40 00 00 00 00 00 00 00 00 00 10 64 00 40 00 12 00 00 10 00 00 1f 05 99 01 97 00 6e 34
a0 20 00 01 40 74 07 af 01 e8 08 40 00 00 00 d4 10 43 01 0b 03 00 20 5c 0f d9 04 33 03 00 00 00 00
a0 20 00 01 40 20 01 00 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 00 01 00

%---------------------
% Page 01 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 01 01 00
% Program Parameter Set
a0 20 01 01 40 18 c6 00 00 01 65 01 00 4e 69 64 65 63 5f 4d 6f 00 00 00 11 09 00 02 00 01 00 c8 00
a0 20 01 01 40 00 00 00 00 00 00 00 00 00 10 64 00 40 00 12 00 00 10 00 00 1f 05 99 01 97 00 6e 34
a0 20 01 01 40 5b 0e 3e 03 e8 08 40 00 00 00 d4 10 6a 02 32 04 00 20 5c 0f d9 04 33 03 00 00 00 00
a0 20 01 01 40 20 01 00 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 01 01 00

%---------------------
% Page 02 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 02 01 00
% Program Parameter Set
a0 20 02 01 40 a0 cc 00 00 01 65 01 00 4e 69 64 65 63 5f 4d 6f 00 00 00 11 09 00 02 00 01 00 c8 00
a0 20 02 01 40 00 00 00 00 00 00 00 00 00 10 64 00 40 00 12 00 00 10 00 00 1f 05 99 01 97 00 6e 34
a0 20 02 01 40 42 15 cd 04 e8 08 40 00 00 00 d4 10 8d 03 55 05 00 20 5c 0f d9 04 33 03 00 00 00 00
a0 20 02 01 40 20 01 00 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 02 01 00

%---------------------
% Page 03 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 03 01 00
% Program Parameter Set
a0 20 03 01 40 12 d9 00 00 01 65 01 00 4e 69 64 65 63 5f 4d 6f 00 00 00 11 09 00 02 00 01 00 c8 00
a0 20 03 01 40 00 00 00 00 00 00 00 00 00 10 64 00 40 00 12 00 00 10 00 00 1f 05 99 01 97 00 6e 34
a0 20 03 01 40 29 1c 5d 06 e8 08 40 00 00 00 d4 10 a8 04 70 06 00 20 5c 0f d9 04 33 03 00 00 00 00

%---------------------
% Page 0f - AppID 00
%---------------------
% Erase Parameter Set
a0 22 0f 01 00
% Program Parameter Set
a0 20 0f 01 40 79 dc 00 00 00 59 01 00 4e 69 64 65 63 5f 4d 6f e2 0b 01 00 17 00 00 00 00 00 10 27 10 27 13 35 10 27 10 27
a0 20 0f 01 40 00 04 89 d0 41 04 42 04 43 c4 00 04 08 04 2a 04 2b 04 00 04 00 04 00 04 00 04 00 04 00 04 00 04 00 04 00 04
a0 20 0f 01 40 00 00 e8 03 e8 03 00 00 ff ff 00 01 01 00 3b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 0f 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 0f 01 00
```

Figure 12 Combined Parameter.ldf file detail for four different sets of parameters

The new .ldf file shoud have an AppID 01 section containing different parameter files and 1 AppID containing the system parameter.

When PFC function is enabled, the macro extracts the Page 00 - AppID 01 (Motor Control parameters), Page 01 - AppID 03 (PFC control parameters), and Page 0f- AppID 00 (System Control parameters) from each .ldf file, and combines them into a new .ldf file containing all four parameter sets as well as the system parameters. Figure 13 shows this new combined .ldf file containing all the parameter sets:



```
%---------------------
% Page 00 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 00 01 00
% Program Parameter Set
a0 20 00 01 40 f6 de 00 00 01 65 01 00 47 6f 6c 64 65 6e 41 67 00 00 18 01 09 00 02 00 02 00 96 00 30 00 30 00 30 00 c0 0
a0 20 00 01 40 64 00 00 00 00 00 00 00 00 10 64 00 40 00 18 00 00 10 00 00 b0 04 00 08 ff 7f 48 01 b0 04 e5 02 7e 1c 00 0
a0 20 00 01 40 0b 05 83 00 e0 0b 40 00 00 00 6e 13 00 00 c8 01 00 20 b9 0b d9 04 45 07 00 00 00 00 00 00 08 00 01 00 02 0
a0 20 00 01 40 20 01 64 03 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
% Check Parameter Set
a0 21 00 01 00

%---------------------
% Page 01 - AppID 03
%---------------------
% Erase Parameter Set
a0 22 01 01 00
% Program Parameter Set
a0 20 01 01 40 21 dd 00 00 03 1d 01 00 47 6f 6c 64 65 6e 41 67 00 00 80 07 21 00 88 13 00 00 00 00 40 00 ff 0f 00 00 f4 6
a0 20 01 01 40 09 06 0c 03 3f 00 30 00 40 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
a0 20 01 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
a0 20 01 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
% Check Parameter Set
a0 21 01 01 00
%---------------------
% Page 02 - AppID 01
%---------------------
% Erase Parameter Set
a0 22 02 01 00
% Program Parameter Set
a0 20 02 01 40 5a df 00 00 01 65 01 00 47 6f 6c 64 65 6e 41 67 00 00 18 01 09 00 02 00 02 00 96 00 30 00 30 00 30 00 c0 0
a0 20 02 01 40 c8 00 00 00 00 00 00 00 00 10 64 00 40 00 18 00 00 10 00 00 b0 04 00 08 ff 7f 48 01 b0 04 e5 02 7e 1c 00 0
a0 20 02 01 40 0b 05 83 00 e0 0b 40 00 00 00 6e 13 00 00 c8 01 00 20 b9 0b d9 04 45 07 00 00 00 00 00 00 08 00 01 00 02 0
a0 20 02 01 40 20 01 64 03 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
% Check Parameter Set
a0 21 02 01 00

%---------------------
% Page 03 - AppID 03
%---------------------
% Erase Parameter Set
```

```
%---------------------
% Page 07 - AppID 03
%---------------------
% Erase Parameter Set
a0 22 07 01 00
% Program Parameter Set
a0 20 07 01 40 4d de 00 00 03 1d 01 00 47 6f 6c 64 65 6e 41 67 00 00 80 07 21 00 88 13 00 00 00 00 40 00 ff 0f 00 00 f4 68
a0 20 07 01 40 09 06 0c 03 3f 00 30 00 40 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 07 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 07 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 07 01 00

%---------------------
% Page 0f - AppID 00
%---------------------
% Erase Parameter Set
a0 22 0f 01 00
% Program Parameter Set
a0 20 0f 01 40 00 1b 00 00 00 59 01 00 47 6f 6c 64 65 6e 41 67 63 08 01 00 17 00 00 00 00 00 10 27 10 27 13 35 10 27 10 27
a0 20 0f 01 40 80 d0 81 d0 46 84 47 84 48 84 49 84 09 04 4a 04 4b 04 18 04 17 04 16 04 31 04 32 04 33 04 34 04 00 04 00 04
a0 20 0f 01 40 00 00 e8 03 e8 03 00 00 ff fe 01 01 01 00 7b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a0 20 0f 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
% Check Parameter Set
a0 21 0f 01 00
```

Figure 13 Combined Parameter.ldf file detail for four different sets of parameters (partial)

The new .ldf file shoud have an AppID 01 section containing different parameter sets for motor control, an AppID 03 section containing different parameter sets for PFC function, and 1 AppID 0f containing the system parameters.

In our example, there is now a single .ldf file containing four different parameter sets.

2.2.3 Final combine .ldf file to be used with MCEDesigner

The last step is to reproduce the exercise described in section 3.2 to create a final combined file containing all different parameter sets and the desired firmware into a single .ldf file.

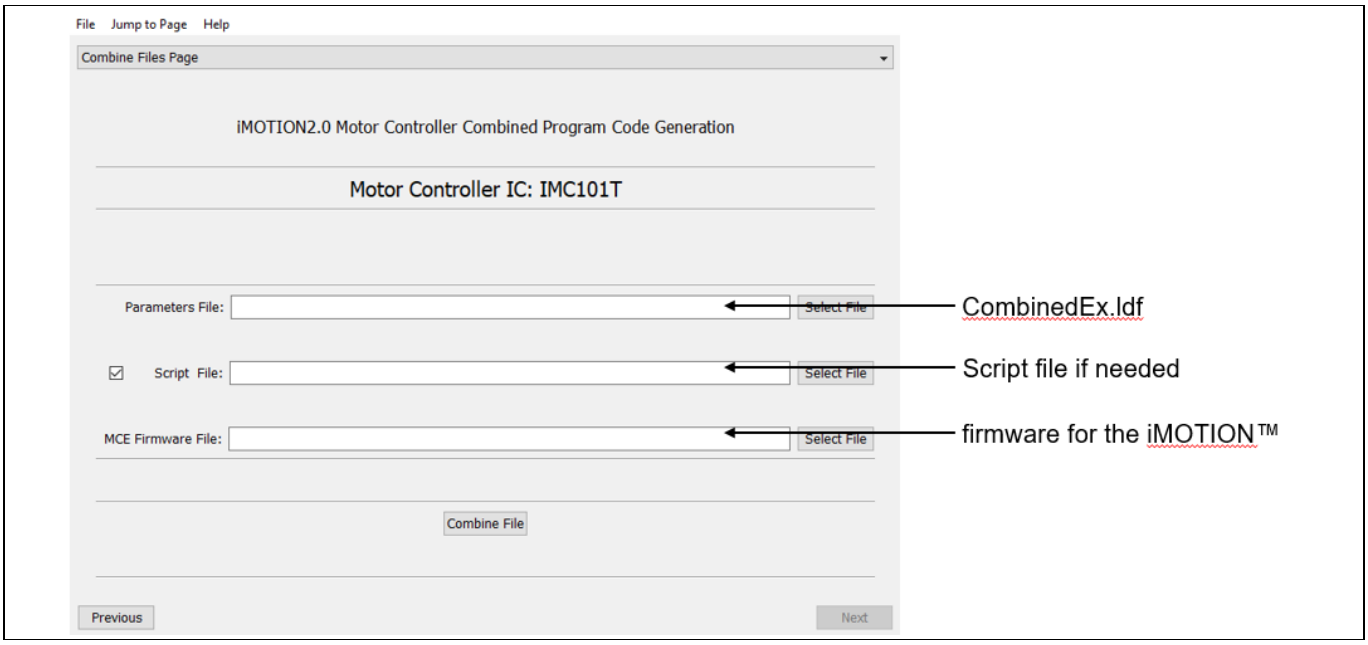


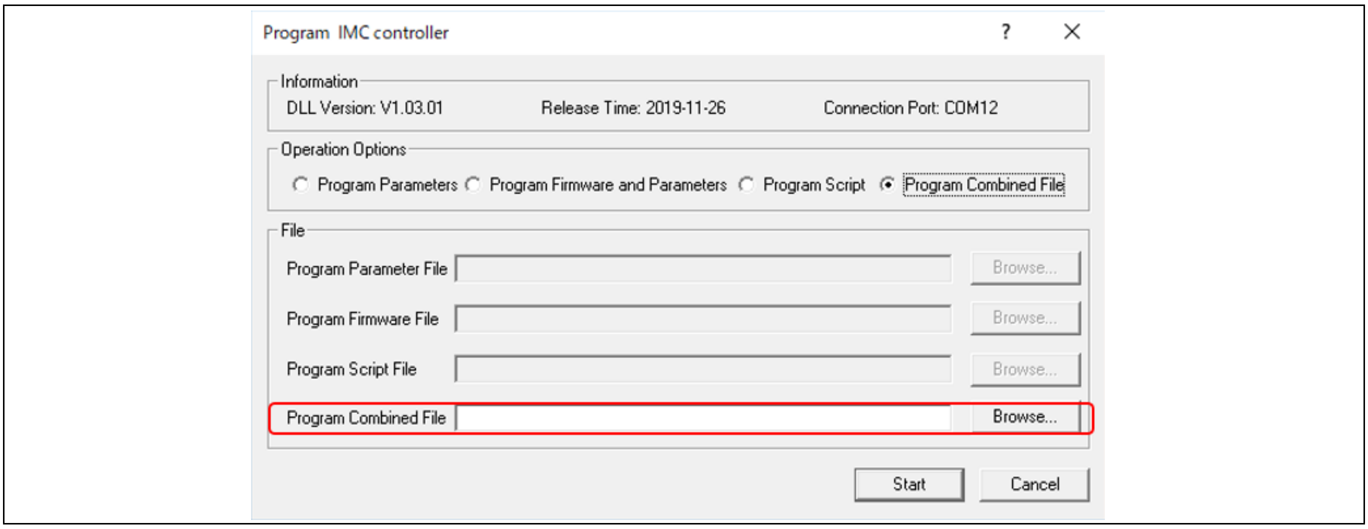
Figure 14 **Last step into creating a combined .ldf file containing all different parameter sets in MCEWizard**

2.3 Programing combined files with the MCEDesigner

Now that we have our combined file containing all our needed information for the different setups, the last step is to upload it into the iMOTION™ controller using the MCEDesigner. In order to do so:

1- Select Tools --> Programmer in the MCEDesigner

2- Select combined parameter files in “Program Combined File” and press start


Program IMC controller

Information
DLL Version: V1.03.01   Release Time: 2019-11-26   Connection Port: COM12

Operation Options
○ Program Parameters ○ Program Firmware and Parameters ○ Program Script ● Program Combined File

Figure 15 **Programing combined .ldf file with the MCEDesigner**

3 Multiple Parameter Handling by iMOTION™ Solution Designer

The iMOTIONTM Solution Designer (iSD) supports multiple parameter sets in one single project. It is much easier to handle multiple parameters with the iSD than the MCEWizard/Designer.

This section will cover how to configure multiple parameter sets and how to program the parameter sets to an iMOTIONTM device by using the iSD.

3.1 Multiple Parameter Set Definition

3.1.1 iSD Configuration

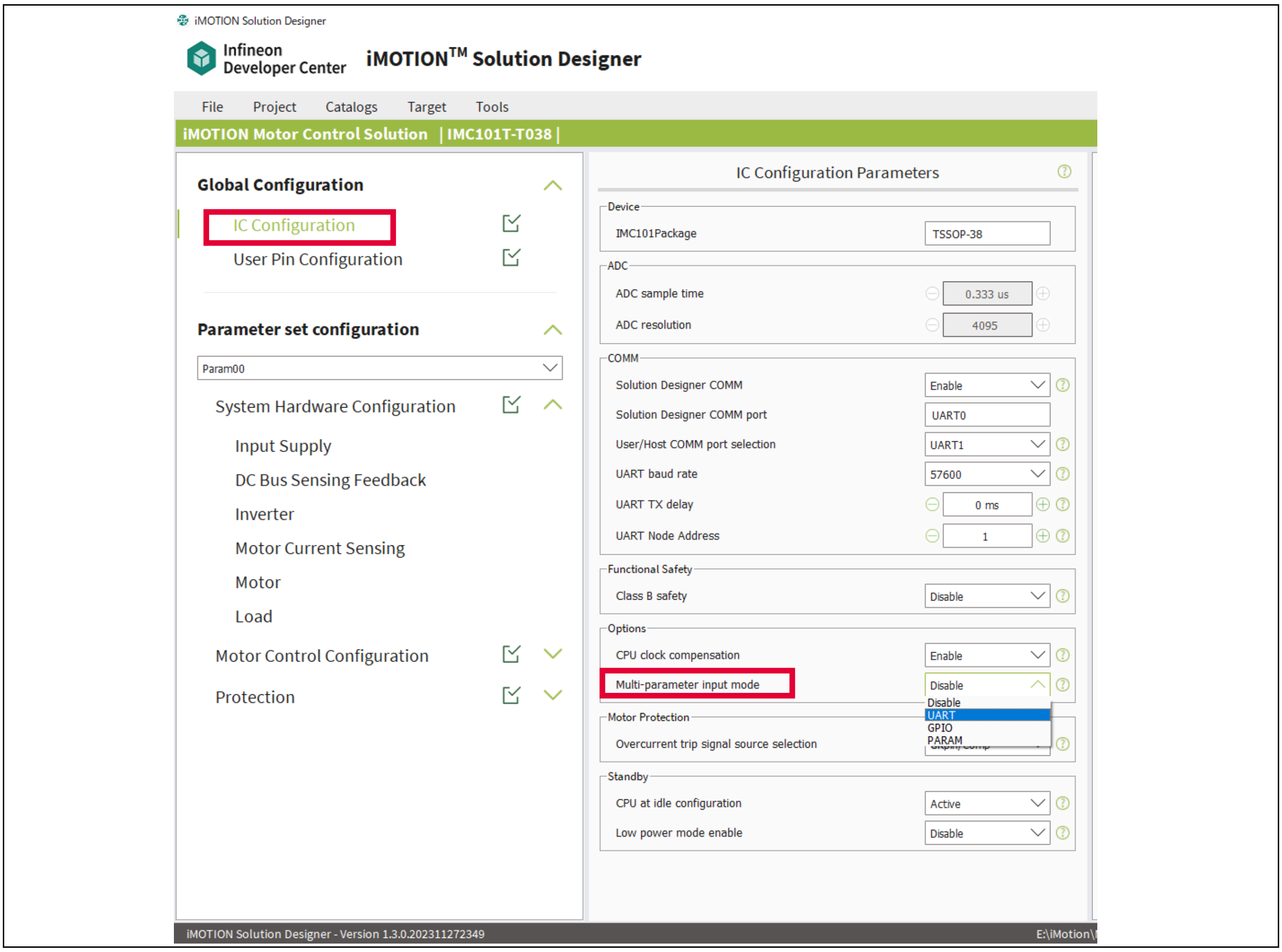
By default, only one set of parameters is stored in FLASH and loaded into RAM of an iMOTION™ device, and the ability to select different parameter sets is disabled in Configration Wizard in the iSD. We can enable multiple-parameter set support via the *Multi-parameter input mode option* in the Configuration Wizard, as shown in Figure 16.

If the multiple parameter function is not used, please select "Disable". Other options should be chosen according to the way the parameter set number was selected.

UART: User UART

GPIO: GPIO pins (PAR0 / PAR1 / PAR2 / PAR3)

PARAM: Analog input pin (PARAM)


iMOTION Solution Designer
Infineon Developer Center
iMOTION^TM Solution Designer
File
Project
Catalogs
Target
Tools
iMOTION Motor Control Solution | IMC101T-T038 |
Global Configuration
IC Configuration
User Pin Configuration
Parameter set configuration
Param00
System Hardware Configuration
Input Supply
DC Bus Sensing Feedback
Inverter
Motor Current Sensing
Motor
Load
Motor Control Configuration
Protection
IC Configuration Parameters

Device

IMC101Package	TSSOP-38

ADC

ADC sample time	0.333 us
ADC resolution	4095

COMM

Solution Designer COMM	Enable
Solution Designer COMM port	UART0
User/Host COMM port selection	UART1
UART baud rate	57600
UART TX delay	0 ms
UART Node Address	1

Functional Safety

Class B safety	Disable

Options

CPU clock compensation	Enable
Multi-parameter input mode	Disable Disable UART GPIO PARAM

Motor Protection

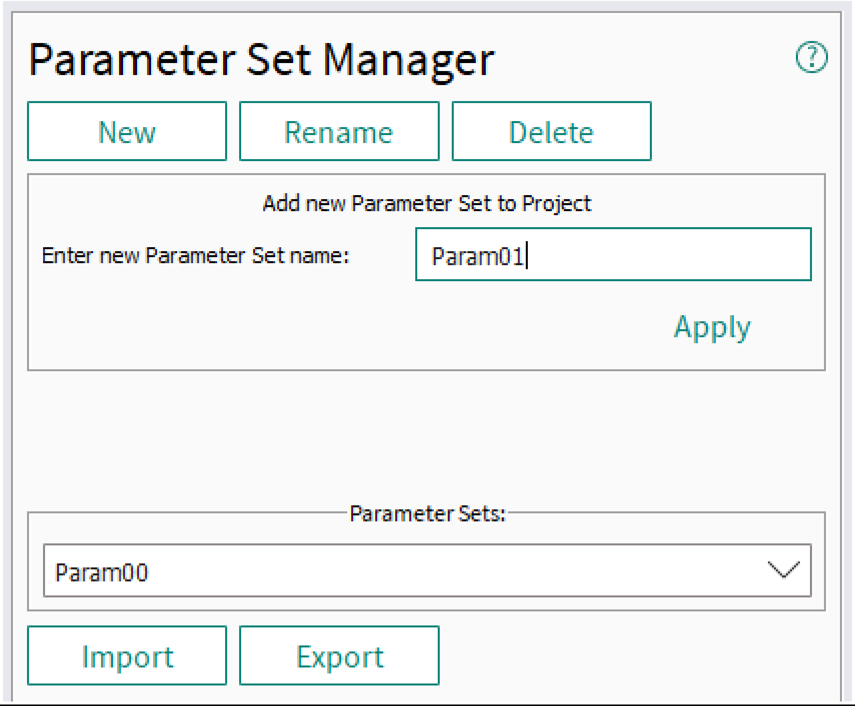
Overcurrent trip signal source selection	

Standby

CPU at idle configuration	Active
Low power mode enable	Disable

Figure 16 Multiple Motor Parameter Set Support in Configuration Wizard iSD

The Parameter Set Manager in the iSD is used to create a new parameter set. The Parameter Set Manager can be opened by selecting Project → Parameter Set Manager. Users will then see the Parameter Set Manager, as shown in Figure 17. Press "New" to create a new parameter set, input the parameter set name, and then press "Apply". After that, the iSD generates a new parameter set and the Configuration Wizard is opened, as shown in Figure 18. Users will see the newly added parameter set name shown in the pulldown list in 'Parameter Set Configuration'.

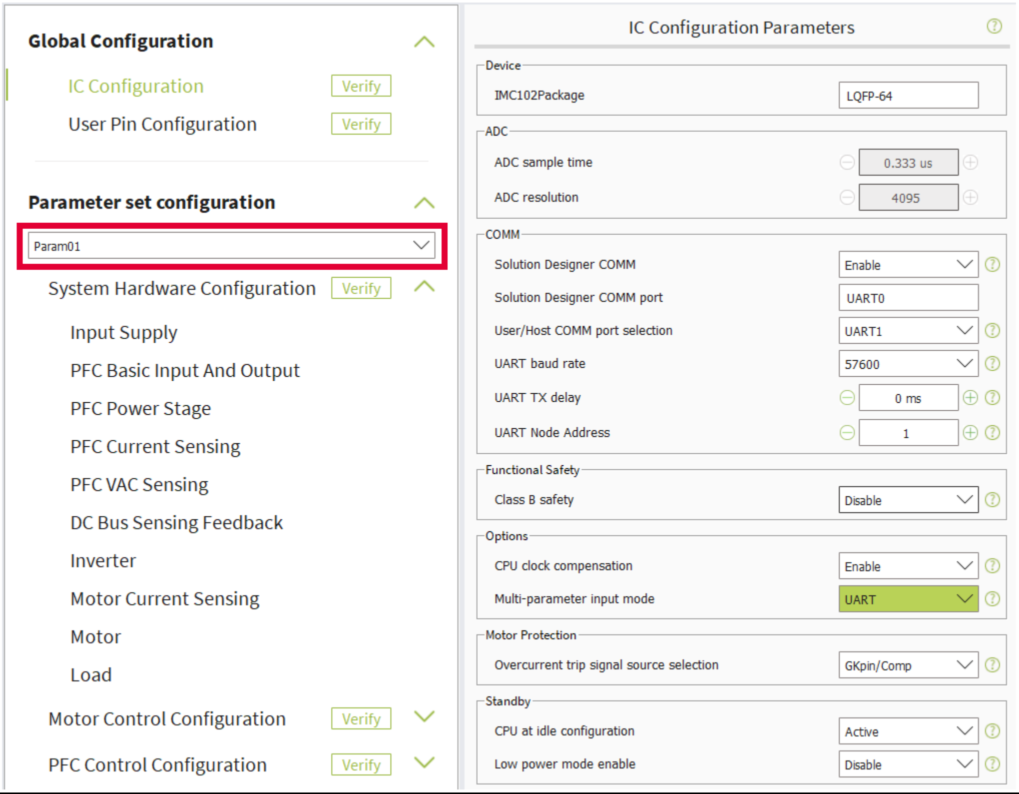


Add new Parameter Set to Project

Enter new Parameter Set name: Param01

Parameter Sets: Param00

Figure 17 Parameter Set Manager in iSD



Param01

IC Configuration Parameters

Device
IMC102Package: LQFP-64

ADC
ADC sample time: 0.333 us
ADC resolution: 4095

COMM
Solution Designer COMM: Enable
Solution Designer COMM port: UART0
User/Host COMM port selection: UART1
UART baud rate: 57600
UART TX delay: 0 ms
UART Node Address: 1

Functional Safety
Class B safety: Disable

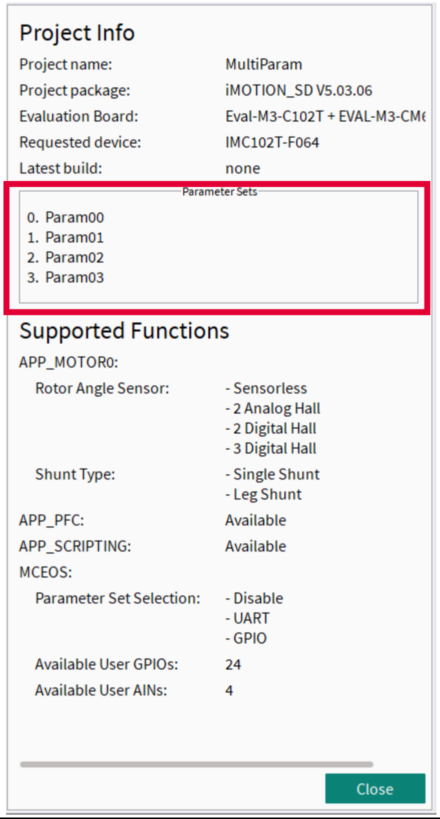
Options
CPU clock compensation: Enable
Multi-parameter input mode: UART

Motor Protection
Overcurrent trip signal source selection: GKpin/Comp

Standby
CPU at idle configuration: Active
Low power mode enable: Disable

Figure 18 Configuration Wizard with Multiple Parameter Sets

Users can confirm the list of parameter sets in 'Project Info', which is displayed by selecting 'Project Settings' or 'Parameter Set Manager'. Figure 19 shows an example of the Project Info pane. List of parameter sets is shown in 'Parameter Sets' section. Each parameter set has an individual number, and it is used to signify that the parameter is set to be used in the MCE.



Project Info

Project name:	MultiParam
Project package:	iMOTION_SD V5.03.06
Evaluation Board:	Eval-M3-C102T + EVAL-M3-CM
Requested device:	IMC102T-F064
Latest build:	none

Parameter Sets

0. Param00
1. Param01
2. Param02
3. Param03

Supported Functions

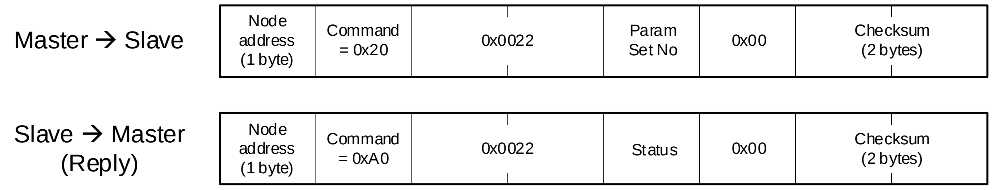
APP_MOTOR0:	
Rotor Angle Sensor:	- Sensorless - 2 Analog Hall - 2 Digital Hall - 3 Digital Hall
Shunt Type:	- Single Shunt - Leg Shunt
APP_PFC:	Available
APP_SCRIPTING:	Available
MCEOS:	
Parameter Set Selection:	- Disable - UART - GPIO
Available User GPIOs:	24
Available User AINs:	4

Figure 19 Project Info

Valid parameter set IDs can range from 0 to 15 (0x00~0x0F) for FW5.x. Each parameter set ID includes both motor and PFC parameters if PFC function is enabled.

3.1.2 Motor ID using UART

UART parameter load command is prepared for FW5.x as well as FW1.3.7. It is necessary to execute system reset after parameter load in order to calculate correct system parameter in FW5.x. Figure 20 shows UART packets for parameter load command with system reset in FW5.x.



Master → Slave	Node address (1 byte)	Command = 0x20	0x0022	Param Set No	0x00	Checksum (2 bytes)
Slave → Master (Reply)	Node address (1 byte)	Command = 0xA0	0x0022	Status	0x00	Checksum (2 bytes)

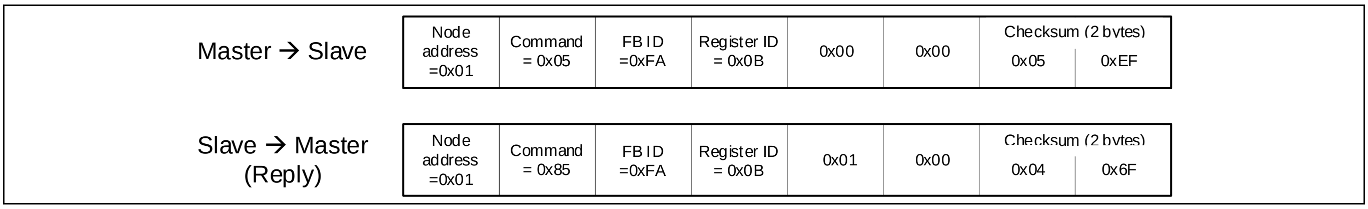
Figure 20 UART parameter load command with system reset

It is recommended to change parameter set when MCE is in STOP state. In order to check state of MCE, register read command (0x05) can be used.

Recommended procedure of parameter set change is shown in below. In this example, node address of iMOTION^{TM} device is 1, and parameter set number 3 is loaded by UART command.

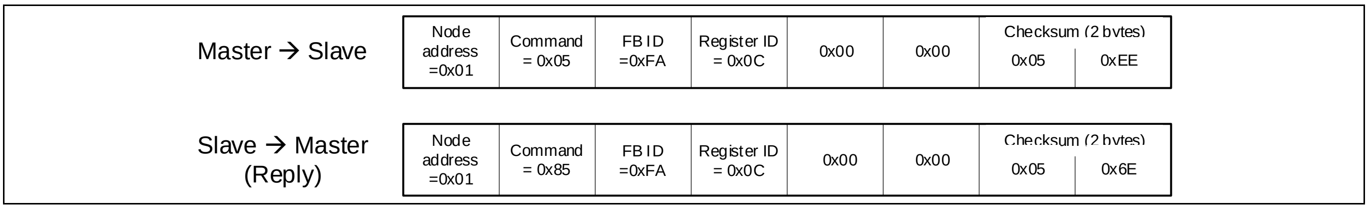
In the beginning, it is necessary to check MCE status by register read command. Motor_SequencerState(FB ID = 0xFA, Register ID = 0x0B) should be 1 (STOP state). If PFC is enabled, PFC_SequencerState(FB ID = 0xFA, Register ID = 0x0C) should be 0 (PFC_IDLE). Figure 21 and Figure 22 show the examples of UART message to check

Motor_SequencerState and PFC_SequencerState, respectively. If it is confirmed that Motor_SequencerState = 1 and PFC_SequencerState = 0, issue load parameter command to change parameter set as shown in Figure 23.



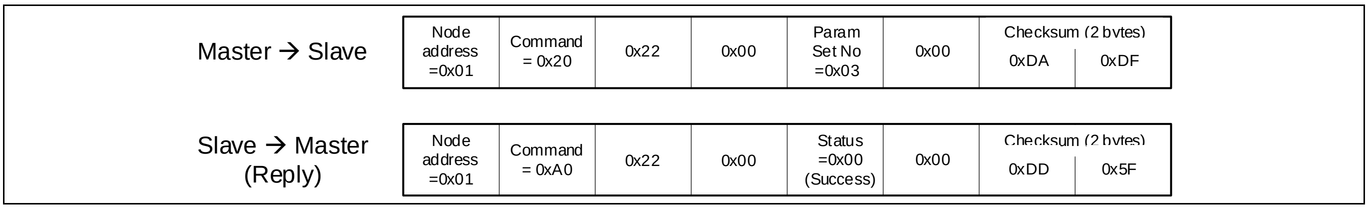
							Checksum (2 bytes)	
Master → Slave	Node address =0x01	Command = 0x05	FB ID =0xFA	Register ID = 0x0B	0x00	0x00	0x05	0xEF
							Checksum (2 bytes)	
Slave → Master (Reply)	Node address =0x01	Command = 0x85	FB ID =0xFA	Register ID = 0x0B	0x01	0x00	0x04	0x6F

Figure 21 Register Read Command to Check Motor_SequencerState



							Checksum (2 bytes)	
Master → Slave	Node address =0x01	Command = 0x05	FB ID =0xFA	Register ID = 0x0C	0x00	0x00	0x05	0xEE
							Checksum (2 bytes)	
Slave → Master (Reply)	Node address =0x01	Command = 0x85	FB ID =0xFA	Register ID = 0x0C	0x00	0x00	0x05	0x6E

Figure 22 Register Read Command to Check PFC_SequencerState



							Checksum (2 bytes)	
Master → Slave	Node address =0x01	Command = 0x20	0x22	0x00	Param Set No =0x03	0x00	0xDA	0xDF
							Checksum (2 bytes)	
Slave → Master (Reply)	Node address =0x01	Command = 0xA0	0x22	0x00	Status =0x00 (Success)	0x00	0xDD	0x5F

Figure 23 Load Parameter Command

3.1.3 Motor ID using GPIO pins

As described in section 2.1.3, parameter set can be selected by using four GPIO pins (PAR0, PAR1, PAR2, and PAR3 pins).

3.1.4 MotorID using analog input

Analog input method to select parameter set is same as FW1.3.7. Please refer to Section 2.1.4.

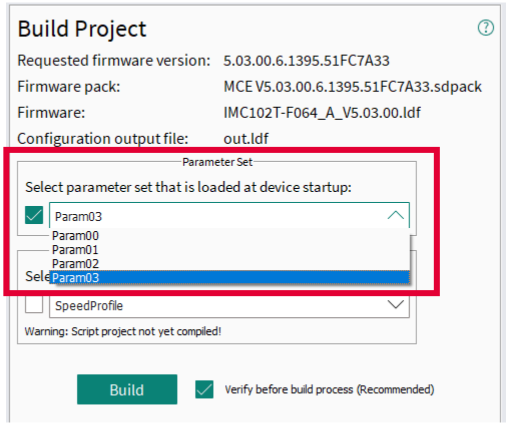
3.1.5 Build Project with Multple Parameter Sets

It is necessary to build the iSD project before programming it. Please refer to [2] and [5] for how to build the project.

When the multiple parameter function is enabled and UART is slected as an input mode in Configuration Wizard as shown in Figure 16, 1st parameter set (parameter set 0) is always loaded at the device start up.

If it is necessary to select the parameter set at the device start up, please disable multi-parameter input mode shown in Figure 16, and select the parameter set that is loaded at the device start up, as shown in Figure 24.

If a script is used in the project, it is necessary to check the checkbox for script and select the script to be used in the project list.



Build Project

Requested firmware version: 5.03.00.6.1395.51FC7A33
Firmware pack: MCE V5.03.00.6.1395.51FC7A33.sdpack
Firmware: IMC102T-F064_A_V5.03.00.ldf
Configuration output file: out.ldf

Parameter Set

Select parameter set that is loaded at device startup:

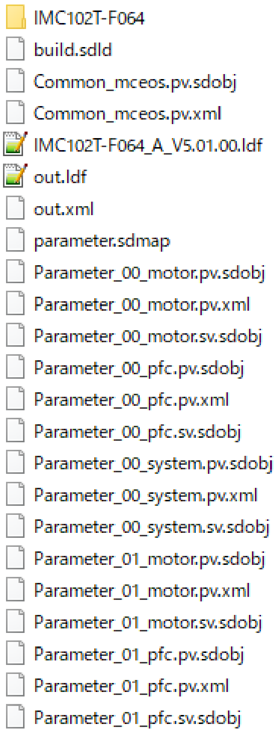
☑ Param03
Param00
Param01
Param02
Param03

Sele
□ SpeedProfile
Warning: Script project not yet compiled!

Build
☑ Verify before build process (Recommended)

Figure 24 Selecting parameter set that is loaded to the device at start up

After building the project, several files are generated in the "generated" folder of the project folder. Figure 25 shows an example of list of the files in the "generated" folder. There are two *.ldf files in the "generated" folder. One *.ldf file, which contains the device name in the file name, is firmware for the specific device. In this case, it is "IMC102T-F046_A_V5.01.00.ldf". Another *.ldf file, "out.ldf", contains the parameter set information and the script.



IMC102T-F064
build.sdld
Common_mceos.pv.sdobj
Common_mceos.pv.xml
IMC102T-F064_A_V5.01.00.ldf
out.ldf
out.xml
parameter.sdmap
Parameter_00_motor.pv.sdobj
Parameter_00_motor.pv.xml
Parameter_00_motor.sv.sdobj
Parameter_00_pfc.pv.sdobj
Parameter_00_pfc.pv.xml
Parameter_00_pfc.sv.sdobj
Parameter_00_system.pv.sdobj
Parameter_00_system.pv.xml
Parameter_00_system.sv.sdobj
Parameter_01_motor.pv.sdobj
Parameter_01_motor.pv.xml
Parameter_01_motor.sv.sdobj
Parameter_01_pfc.pv.sdobj
Parameter_01_pfc.pv.xml
Parameter_01_pfc.sv.sdobj

Figure 25 Contents of "generated" folder in the project folder

The file 'out.ldf' contains all of the parameter sets, the selected script, and all of the programming commands required. Table 4 below gives an example of the output file with muiltiple parameter sets and the script. Users should refer to [4] for greater detail. By using the iSD, users do not have to do complicated procedures like FW1.3.7, as described in chapter 2.2.

Table 4 Example of *.ldf file generated by iMOTION™ Solution Designer

Data	Description

`% iMOTION configuration file`
`% iMOTIONld version: V1.1.0 - 2022-10-11`
	file header,
	lines starting with '%' are comments

`% Erase Script memory`
`a0 22 00 05 00`
`a0 21 00 05 00`
	erasing script and verification of the script

`%linker input file`
`%filename: C:\...\generated\build.sdld`
`%built: 2023-03-10, 10:16:19`
	build information from iSD

`% Erase Static Parameters`
`a0 22 0f 01 00`
`% Static Parameters`
`a0 20 0f 01 40 41 05 00 00 ... 00 00`
`...`
`a0 20 0f 01 40 00 44 00 44 ... 00 00`
`% Verify Static Parameters`
`a0 21 0f 01 00`
	erasing, programming and verification of system parameters

`% Parameter set: myFirstParameterSet`
`% Erase Parameter Set`
`a0 22 00 04 00`
`% Program Parameter Set`
`a0 20 00 04 30 00 00 00 00 ... 00 00`
`...`
`a0 20 00 04 20 02 22 00 20 ... 00 00`
`% Verify Parameter Set`
`a0 21 00 04 00`
	erasing, programming and verification of 1st parameter set (parameter set number = 0, parameter set name = myFirstParameterSet)

`% Parameter set: Param01`
`% Erase Parameter Set`
`a0 22 01 04 00`
`% Program Parameter Set`
`a0 20 01 04 30 00 00 00 00 ... 00 00`
`...`
`a0 20 01 04 30 0a 2a 73 3e ... 00 00`
`% Verify Parameter Set`
`a0 21 01 04 00`
	erasing, programming and verification of 2nd parameter set (parameter set number = 1, parameter set name = Param01)

`.....`	Parameter section continues up to the last parameter set

`% Parameter set: Param03`
`% Erase Parameter Set`
`a0 22 03 04 00`
`% Program Parameter Set`
`a0 20 03 04 30 00 00 00 00 ... 00 00`
`...`
`a0 20 03 04 30 0a 2a 85 3e ... 00 00`
`% Verify Parameter Set`
	erasing, programming and verification of last parameter set (parameter set number = 3, parameter set name = Param03)

Data	Description
a0 21 03 04 00	
%-------------------------------------- %# Build Successful - 0 Error, 0 Warning %# Build Date and Time : 2023-03-10 14:49:58 %-------------------------------------- % Script Translator Version : 2.00.00 %-------------------------------------- % Script Object File %-------------------------------------- % SCRIPT_USER_VERSION : 001.000 %# Script Code Memory Size : 304 Bytes of 16 kBytes % Total Number of Lines : 122 %# Number of Global Variable(s) : 2, Number of Flash Variable(s) : 0, Data Memory Usage : 5 Bytes of 256 Bytes %# Task0 - Number of Instruction(s) : 23, Number of Variable(s) : 3, Data Memory Usage : 7 Bytes of 128 Bytes %# Task1 - Number of Instruction(s) : 5, Number of Variable(s) : 0, Data Memory Usage : 0 Bytes of 128 Bytes %--------------------------------------	script code header, lines starting with '%' are comments Summary of version, size and total number of lines number of instructions per task, can be used for partitioning of the script task execution (see [5] and [6])
% Erase Script memory a0 22 00 05 00 % Program Script a0 20 00 05 40 65 c3 ff ff ... 56 00 ... a0 20 00 05 30 66 01 3a 06 ... fb lf % Verify Script a0 21 00 05 00	erasing, programming and verification of the script

4 References

[1] Getting Started with iMOTION™ 2.0

[2] Getting Started with iMOTION™ Solution Designer

[3] iMOTION Solution Designer User Guide

[4] AN2018-33 iMOTION™ Device Programming

[5] iMOTION™ Motion Control Engine Functional Reference Manual

[6] How to Use iMOTION™ Script Laungage

5 Revision history

Major changes since the last revision

Document version	Date of release	Description of changes
1.0	2020-12-11	First Release
1.1	2023-05-01	Added parameter handling with PFC function, minor wording corrections Added iMOTION™ Solution Designer information
1.2	2023-12-15	Parameter handling by UART for FW5.x is updated

Edition 2023-12-15

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN2020-25

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.