

# Handling multiple motor parameters with iMOTION™ 2.0

## A guide for iMOTION™, MCEWizard and MCEDesigner

### About this document

#### Scope and purpose

iMOTION™ 2.0 is a turnkey yet versatile solution for efficient and quiet control of variable speed motors. We explain how to set up one iMOTION™ 2.0 instance to handle multiple motors or multiple loads.

#### Intended audience

This application note is targeting developers implementing variable speed motor control drives using iMOTION™ 2.0.

### Table of contents

<b>About this document.....</b>	<b>1</b>
<b>Table of contents.....</b>	<b>1</b>
<b>1 Introduction.....</b>	<b>2</b>
<b>2 Multiple-parameter block definition .....</b>	<b>3</b>
2.1 MCEWizard configuration.....	3
2.1.1 Motor ID using UART .....	6
2.1.2 Motor ID using GPIO pins .....	7
2.1.3 MotorID using analog input .....	8
2.1.4 Motor ID using Direct Select.....	9
<b>3 Multiple-parameter application example .....</b>	<b>10</b>
3.1 Preparation of MCEWizard .txt parameter files.....	10
3.2 Combine .txt parameter files into a single .ldf file .....	11
3.2.1 VBA function to combine multiple .ldf combined files into a single usable one in MCEWizard ....	12
3.3 Final combine .ldf file to be used with MCEDesigner .....	16
3.4 Programing combined files with MCEDesigner .....	16
<b>Revision history.....</b>	<b>18</b>

### Introduction

## 1 Introduction

iMOTION™ 2.0 offers efficient control of permanent magnet motors by integrating an advanced software implementation in various hardware form factors. iMOTION™ 2.0 comes with two development tools: MCEWizard and MCEDesigner. MCEWizard is a GUI tool that allows you to enter key motor parameters to control a motor, such as motor stator inductance and resistance, back ElectroMotive Force, and motor speed limits.

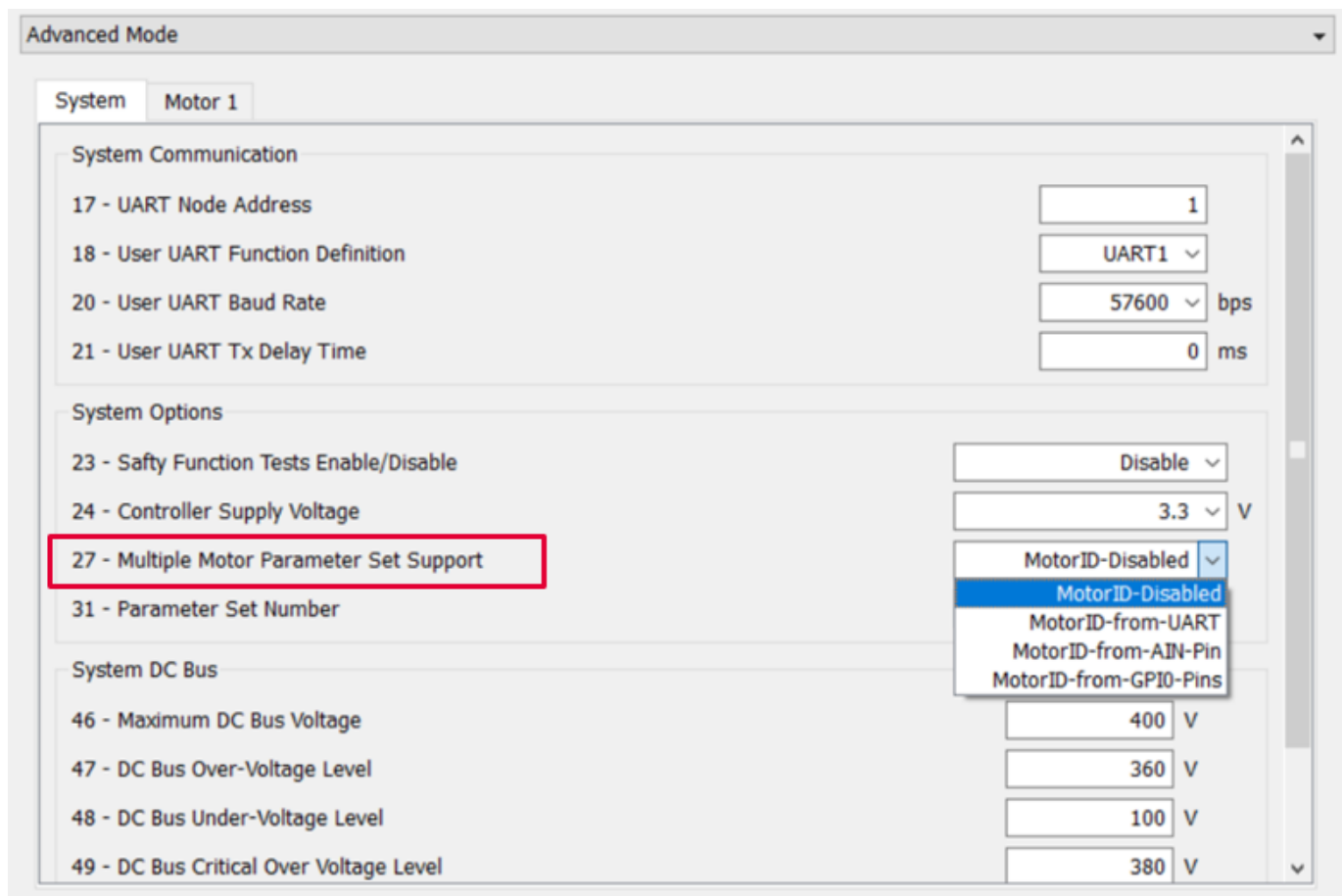
For certain applications, there may be a need to utilize the same iMOTION™ control solution to run either different motors or the same motor under different load conditions. As an example, one iMOTION™ solution should be able to run a compressor under the right cooling profile in any of three refrigerator 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 motor parameter sets in order to support controlling various motors and hardware configurations with one solution.

## 2 Multiple-parameter block definition

### 2.1 MCEWizard configuration

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



**Figure 1 Multiple Motor Parameter Set Support in MCEWizard**

By default, only one set of parameters is loaded in MCEWizard, and the ability to select different parameter sets is disabled. We can enable multiple-parameter set support via the *Multiple Motor Parameter Set Support* option in MCEWizard, as shown in Figure 1 above.

Each parameter set has a unique identification number that can be assigned in four different ways:

- User UART
- GPIO pins
- PARAM (AIN) pin
- Direct select

### Multiple-parameter block definition

When the feature is enabled, iMOTION™ firmware will load the parameter set with the matching ID. To set up this ID, chose a number to enter in the “Parameter Set Number” section for the set of parameters entered in MCEWizard.

When working with different setups, remember that for iMOTION™ microcontrollers, 4 k bytes of flash memory are used to store the different control parameter data. There are 16 parameter blocks, each of them 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 all the 15 different system parameters.

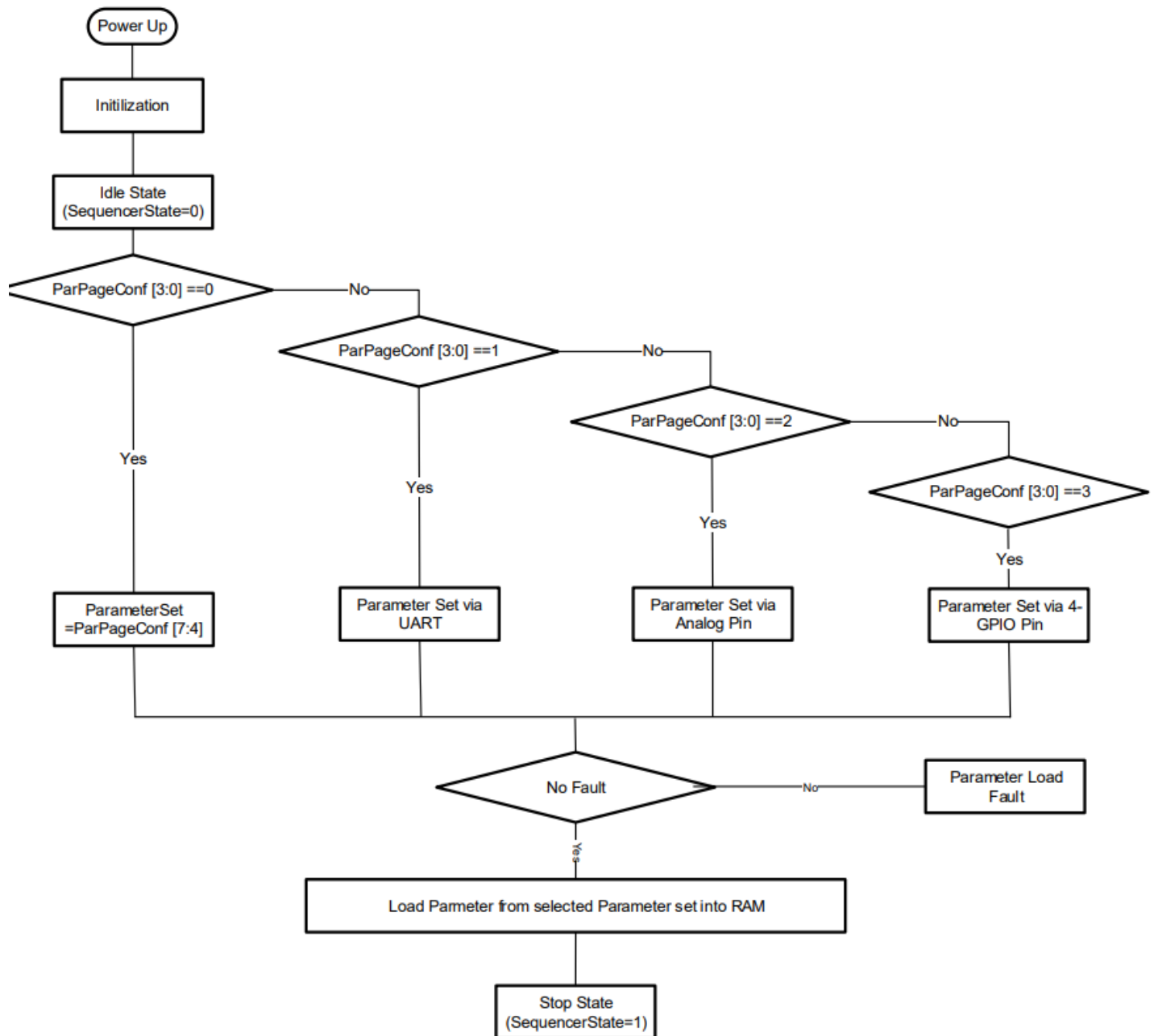
For a system with motor control and PFC functions, each parameter set will take two consecutive parameter blocks. The motor control parameter set will be stored in the selected parameter block, and the PFC parameter set will be stored in the subsequent parameter block. In this case, the valid parameter set IDs are 0, 2, 4, 6, 8, 10, 12, and 14 and only 8 different configurations are available.

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 for easy loading into the controller with MCEDesigner.

Note that the parameter selection methods available depends on the iMOTION™ hardware used. IMM101T-046, for example, only supports Direct and User UART selection methods. IMC101T-T038 supports all four (five) methods.

In any case, when choosing any of the four solutions proposed (MCEWizard GPIO and UART are preferred), the register ParPageConf is updated, and enables one of the four solutions. MCEWizard takes care of properly updating the ParPageConf register if GPIO or UART are chosen for the parameter block selection. Figure 2 shows the full parameter load procedure for iMOTION™ controllers.

### Multiple-parameter block definition



**Figure 2 iMOTION parameter load 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

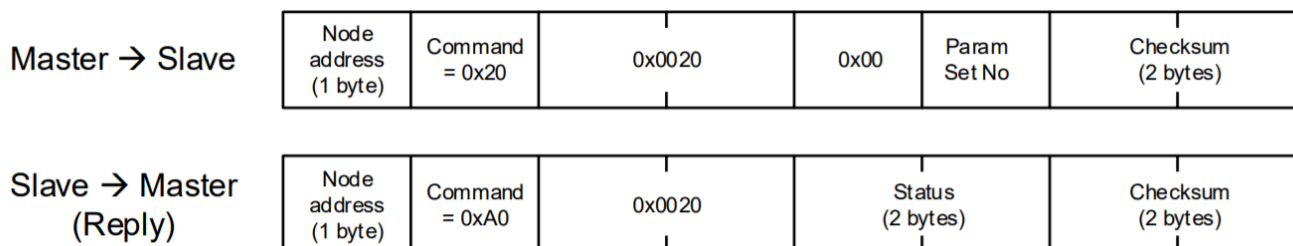
### Multiple-parameter block definition

[15:8] Reserved

Section 3 describes in detail how to use this very important register in order to combine all .txt files into one .txt for loading into MCEDesigner.

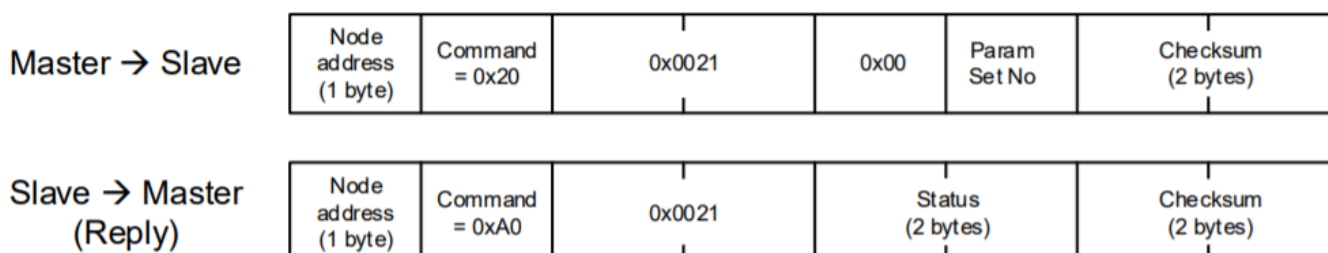
#### 2.1.1 Motor ID using UART

Specific 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



**Figure 3** UART Load parameter command = 0x0020

Save parameter command = 0x21 saves all parameters into one flash page



**Figure 4** UART Save parameter command = 0x20

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, for example, 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 created set of parameters. Command = 0x20
- **Data Word 0** describes if we are 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

### Multiple-parameter block definition

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

Input : Node address = 1, Command = 0x20, Data Word 0 = 0021, Data Word 1 = 0x0003

[Command:Node address] = 0x2001

Checksum = -1 x ( 0x2001 + 0021 + 0003 ) = 5202

UART message to send to iMOTION™ controller to load parameter block 3: 01 20 0021 0003 5202

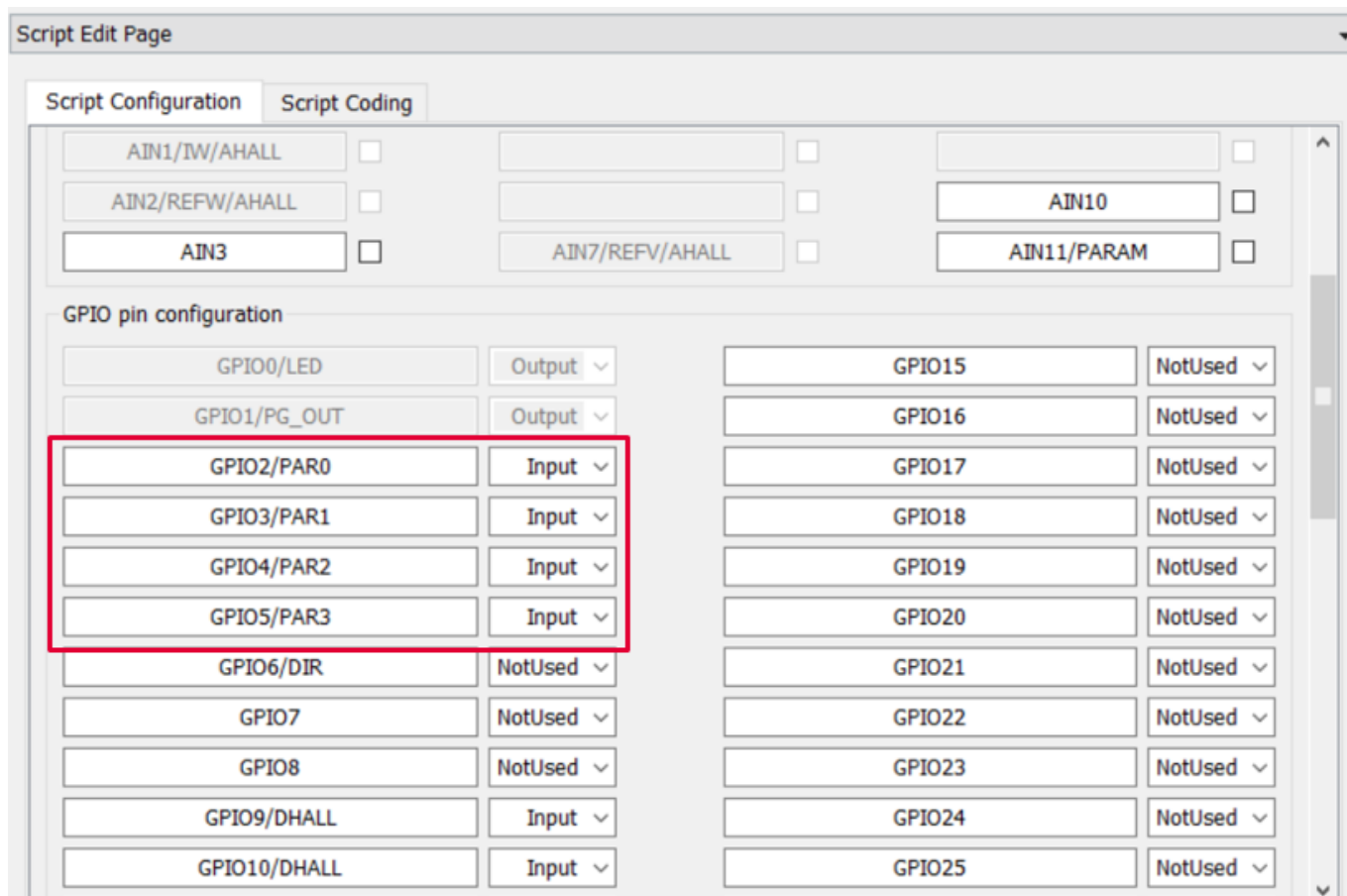
### 2.1.2 Motor ID using GPIO pins

This parameter block is selected based on four 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

**Table 1** Parameter page selection 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
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
1	1	1	1	15

These GPIOs are also accessible through scripting as shown in Figure 5



**Figure 5** GPIO for Block parameter update in MCEWizard

Following our example if we want to load parameter block 3:

PAR0 = 1, PAR1 = 1, PAR2 = 0, PAR3 = 0

Please also keep in mind that the association PARx/GPIOx will depend on which iMOTION™ microcontroller you are using.

### 2.1.3 MotorID using analog input

If this method is chosen, ParPageConf [3:0] = 2. Parameter block is selected based on the analog input value. MCE uses the “PARAM” pin as the analog input for parameter set selection. Mapping between parameter page selections based on analog input is shown below:

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

Follow our example to load parameter block 3 with Vadcref = 3.3 V: AnalogInput = 0.80 V.



### Multiple-parameter block definition

#### 2.1.4 Motor ID using Direct Select

As described, the Direct Select mode to select parameter blocks is not supported by MCEWizard, and requires direct manual modification of the ParPageConf register.

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

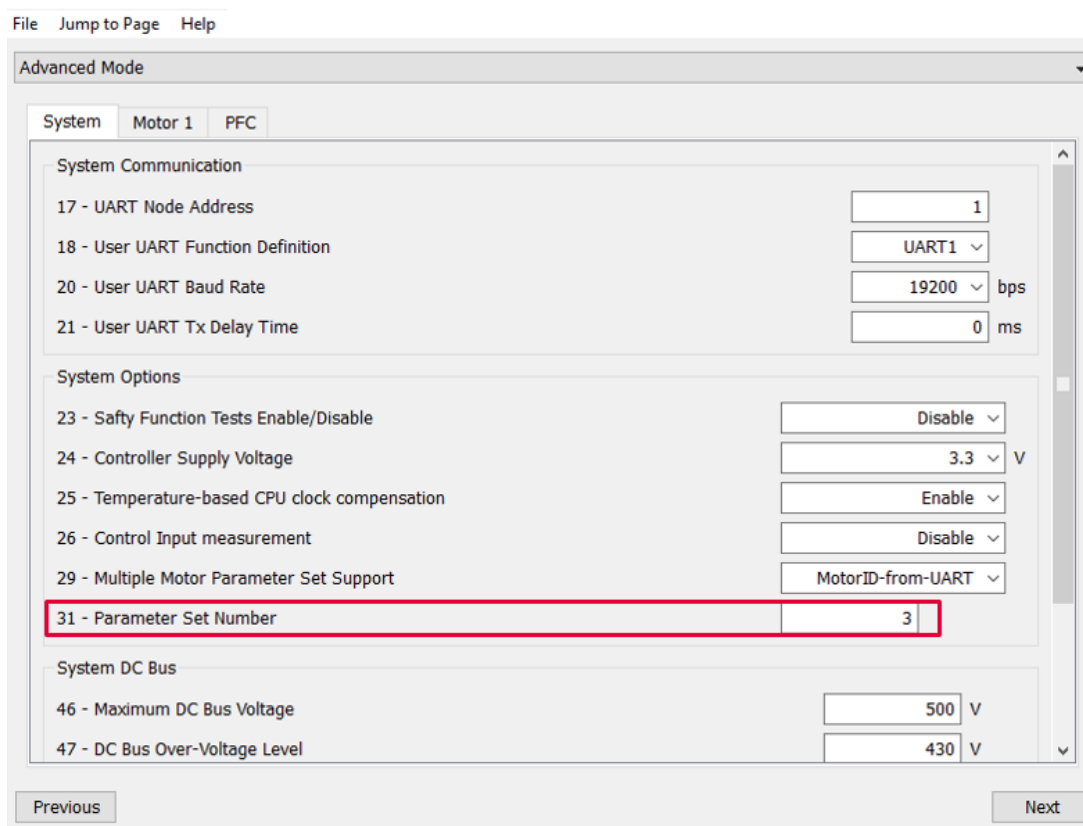
*Direct Select : : ParPageConf[3:0] = 0 and Direct Select : : ParPageConf[7:4] = 3*

## 3 Multiple-parameter application example

In the previous section, we demonstrated how to set up 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 MCEWizard and MCEDesigner.

### 3.1 Preparation of MCEWizard .txt parameter files

As there are four different configurations to select from, via UART with a single iMOTION™ controller, we first need to generate one parameter file for each configuration. We use MCEWizard to create all the parameter files. Each file and parameter set is defined by a unique *Parameter Set Number*. In this example, we enter 1, 2, 3, 4 as the *Parameter Set Number* for each configuration.



**Figure 6** 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:

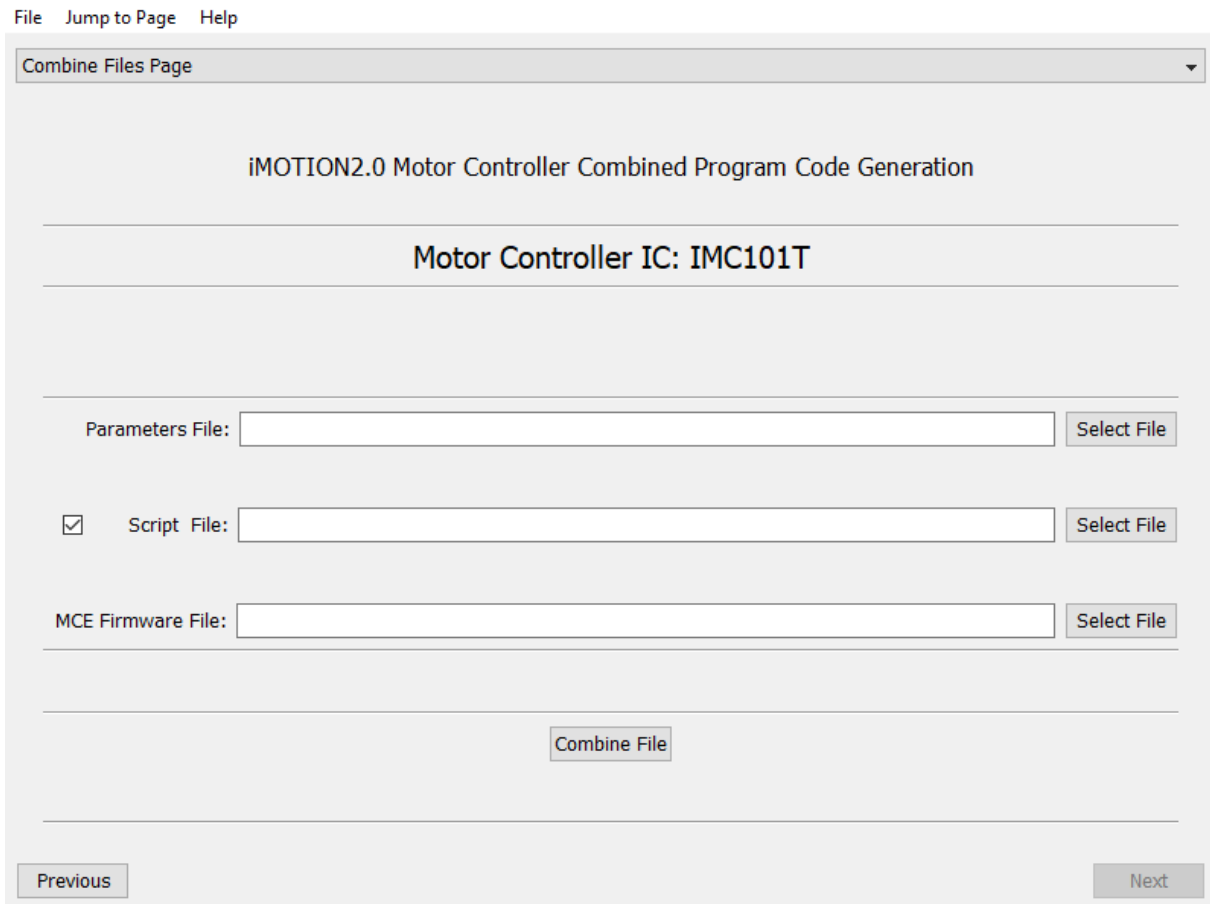
Parameter File Name (Example)	Parameter Set Number	Set Support	##MOTOR1_REGS	ParPageConf
ParameterFile0.txt	0	Motor-ID from UART	0	2049
ParameterFile1.txt	1		1	2065
ParameterFile2.txt	2		2	2081
ParameterFile4.txt	3		3	2097

### Multiple-parameter application example

Now that there are four different .txt files with proper identification, we will show how to combine them into one uploadable file to MCEDesigner.

## 3.2 Combine .txt parameter files into a single .ldf file

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



**Figure 7** 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 step 2 and 3 for the number of parameters you desire. In our example, we will do this exercise four times.

#### 3.2.1 VBA function to combine multiple .ldf combined files into a single usable one in MCEWizard

There will now be four .ldf files (max 15) on hand. The procedure for combining these files manually would be time-consuming. That is why a simple 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. Please copy and paste this function and create a macro in MS Excel to run it.

```
// Object variables definition
```

```
Option Explicit
```

```
Dim Fin As Variant, Fout As Variant
```

```
// VBA function to combine all the different .ldf files
```

```
Sub Make_Combined_LDF
```

```
    Dim ChDir As String, f As Variant
```

```
    Dim i As Integer, n As Integer
```

```
    Dim rc As Integer
```

```
    Fin = Application.GetOpenFilename(FileFilter:="LDF file (*.ldf), *.ldf",  
MultiSelect:=True)
```

```
    If VarType(Fin) = vbBoolean Then
```

```
        MsgBox ("Cancelled")
```

```
    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
```

```
            Open Fout For Output As #2
```

```
            For i = 1 To n
```

### Multiple-parameter application example

```
        Call Make_LDF(Fin(i), i)
    Next i
    Close #2
    MsgBox ("File Combine complete!")
End If

Else
    MsgBox ("Selected file is the only one and process will not be
executed")
End If
End If
End Sub
```

#### // Make\_LDF function definition

```
Sub Make_LDF(f As Variant, i As Integer)
    Dim Ftemp As String, s As String
    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
                End If
            Loop
        End If
        If i = UBound(Fin) Then
            Do
                Line Input #1, s
                Print #2, s
                If InStr(s, "% Check Parameter Set") Then
                    Line Input #1, s
                    Print #2, s
                End If
            Loop
        End If
    Loop
End Sub
```

### Multiple-parameter application example

```

                                End If

                                Loop

                                End If

                                Exit Do

                                End If

                                Loop

                                End If

                                Loop

                                Close #1

End Sub

```

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.

Details of what this macro does can be seen in Figure 8, showing a combined .ldf file:

```

1  %:Combined file 16-BITS CRC result: 0x827F
2  %:Firmware Data Section Begin
3  # DEVICE:   IMD111T-F040
4  # RELEASE:  A_V1.03.01
5  # DATE:    13.04.2020
6  # LIP:     AFD61F00
7  A0 20 00 00 82 A0 00 00 01 5C 67 20 01 00 00 00 32 40 01 4C 05 82 7D 3F 18 E8 39 3F 84 C0 27 19 D2 03 83 AD 49 79 7F E5 CE 30 4F 04 52
8  A0 20 00 00 82 5B BD CA 37 1F D7 46 42 79 EF 88 94 6F DF AC 09 B2 F4 E1 2F 72 70 AA 13 D6 E0 D7 7F AB A7 05 70 87 08 E7 3A E5 CE 8C 59
9  A0 20 00 00 82 2B 3D 87 6E 72 3D 2E E4 49 81 D5 4D 27 B0 07 BA 3C 12 AD E8 E7 8F F5 87 E4 77 AF CB D7 C8 12 2C 8C A7 4B 7E F0 9A 90 43
10 A0 20 00 00 82 06 CE F0 94 5E FF 5F E0 02 9F 87 C5 32 5E 29 05 C4 7E 3E EA 60 F0 41 41 B5 3C A4 ED 35 8A 66 7F 7B F3 D8 6F 44 73 03 09
11 A0 20 00 00 82 32 44 56 2E 0C BA 7D 18 16 6D 5A 3D C9 1F 59 49 5F FF 1C 78 60 F8 46 22 42 51 CC 53 71 A9 98 4B B1 F0 C0 63 64 82 7B 36
12 A0 20 00 00 82 D1 B5 2D 66 2B E2 AC 60 0B D3 34 63 B7 A6 46 AC 5C 77 0C E1 F4 34 44 AF EE 89 DF AE 3C F3 9F FA 08 18 F6 7A DD 7A 09 F5
13 A0 20 00 00 82 13 D9 69 BE 08 E2 D4 93 B3 64 DB 25 D3 EE FF 54 05 85 E7 3E F1 6C 52 22 FC 66 72 6C F2 8D 44 FA 1D 90 26 0B ED 39 25 7A
14 A0 20 00 00 82 14 58 6C A1 C6 07 C8 AA B2 13 17 60 6C 94 DF 5A 89 8D FA 9F 89 ED F2 51 05 F2 DA BA 14 8F BD 6A 54 CE 94 F9 25 3D 7D E5
15 A0 20 00 00 82 F8 E7 56 9C 2B B2 31 67 A1 7F 5A 9C EF E9 7B 3F 7E BF 61 CC 18 22 07 6F EA D3 84 4A 58 8D 1C 6A FA 98 45 7E 88 AB 7D ED
16 A0 20 00 00 82 43 C9 BE FF A0 5C 59 0F E9 38 F7 A8 43 58 1C BE 2B EE 90 4C AB BA 2D 38 7E 09 73 5F CB B5 78 38 86 F3 47 E5 88 81 77 D9
17 A0 20 00 00 82 D2 A1 38 B5 A1 4C 90 87 04 D4 E2 BE CD EF D8 4E 80 B8 7C 2B EF 86 7C D9 33 78 CD 5B 24 76 93 D5 1F A0 22 36 E6 DA 0F 35
18 A0 20 00 00 82 E1 D0 3B 31 9E A1 FD A1 66 DA DB 43 3F B3 E6 95 20 78 60 EE 19 DA 09 18 E0 AD A9 08 9A DC 70 2C D9 C8 6B 02 1A 0D 7A 89
19 A0 20 00 00 82 30 36 57 A9 44 4D B7 C6 4F 2C D0 5C D8 83 54 BA 6D 3E 2E 47 66 96 31 F2 9C 02 EA 88 D2 8A BF 78 41 12 85 92 0D 14 55 04
20 A0 20 00 00 82 0D 1A 3E 06 9E 73 6B 20 42 D9 AD 9B F5 9C C4 EA 5E 0C D9 4A 5E 3D 7B 2D 09 03 63 94 CA 3F 72 66 84 7C 59 81 EB 83 A0 60
21 A0 20 00 00 82 42 A1 87 A8 0B D4 05 AE 7D 28 0E 95 B9 0E 52 F3 16 01 1F 63 1E 52 91 76 F9 CA 4A 75 10 90 53 B8 CF E1 6E 6A 36 25 36 A9
22 A0 20 00 00 82 30 2B 50 77 0C 71 A8 96 DB 64 1C CD 04 A8 B8 6E 52 61 34 C6 6C AD B8 97 3E F6 C5 8F AC EF 7C B9 BF 0A FA 20 F6 F3 A1 FC
23 A0 20 00 00 82 D3 B7 2D FA 1B 98 9B D4 C9 A0 8D FE 31 60 FC 86 DD 1C F3 7C 94 59 38 9C 92 12 1D 4C 08 C5 8F A6 04 BD 87 20 E7 F7 52 D2
24 A0 20 00 00 82 14 46 1F FC FB 61 5B 3D 34 69 16 0E CB 69 AC 39 25 E2 FF 32 E6 5A 56 6C 2B D4 E5 49 50 A3 8B 24 85 65 3A D2 34 00 04 F3

696 %:Firmware Data Section End
697 %:Parameters Data Section Begin
698 %-----
699 % Page 00 - AppID 01
700 %-----
701 % Erase Parameter Set
702 a0 22 00 01 00
703 % Program Parameter Set
704 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 30 00 30 00 11 00 20 01
705 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 1f 05 8b 06 35 1f 89 04
706 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 00 00 00 00 00 00 00 00 00 00
707 a0 20 00 01 40 20 01 00 00 00 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 00 00 00
708 % Check Parameter Set
709 a0 21 00 01 00
710
711 %-----
712 % Page 0f - AppID 00
713 %-----
714 % Erase Parameter Set
715 a0 22 0f 01 00
716 % Program Parameter Set
717 a0 20 0f 01 40 99 db 00 00 00 59 01 00 4e 69 64 65 63 5f 4d 6f 02 0b 01 00 17 00 00 00 00 00 00 10 27 10 27 13 35 10 27 10 27
718 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
719 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
720 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
721 % Check Parameter Set
722 a0 21 0f 01 00
723
724
725 %:Parameters Data Section End
726

```

### Multiple-parameter application example

**Figure 8 Combine Parameter0.ldf file detail. TOP: Firmware section, BOTTOM: Parameter set section**

The macro takes the AppID 01 or Motor Control Registers and AppID 00 or System Control Registers' information for each of the .ldf files, and combines them into a new .ldf file containing all our parameter sets. Figure 9 shows this new combined .ldf file containing all the parameter files:

```

1  %-----
2  % Page 00 - AppID 01
3  %-----
4  % Erase Parameter Set
5  a0 22 00 01 00
6  % Program Parameter Set
7  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
8  a0 20 00 01 40 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
9  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
10 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
11 % Check Parameter Set
12 a0 21 00 01 00
13
14 %-----
15 % Page 01 - AppID 01
16 %-----
17 % Erase Parameter Set
18 a0 22 01 01 00
19 % Program Parameter Set
20 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
21 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
22 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
23 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
24 % Check Parameter Set
25 a0 21 01 01 00
26
27 %-----
28 % Page 02 - AppID 01
29 %-----
30 % Erase Parameter Set
31 a0 22 02 01 00
32 % Program Parameter Set
33 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
34 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
35 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
36 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
37 % Check Parameter Set
38 a0 21 02 01 00
39
40 %-----
41 % Page 03 - AppID 01
42 %-----
43 % Erase Parameter Set
44 a0 22 03 01 00
45 % Program Parameter Set
46 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
47 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
48 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 00

```

**Figure 9 Combined Parameter.ldf file detail for four different sets of parameters.**

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

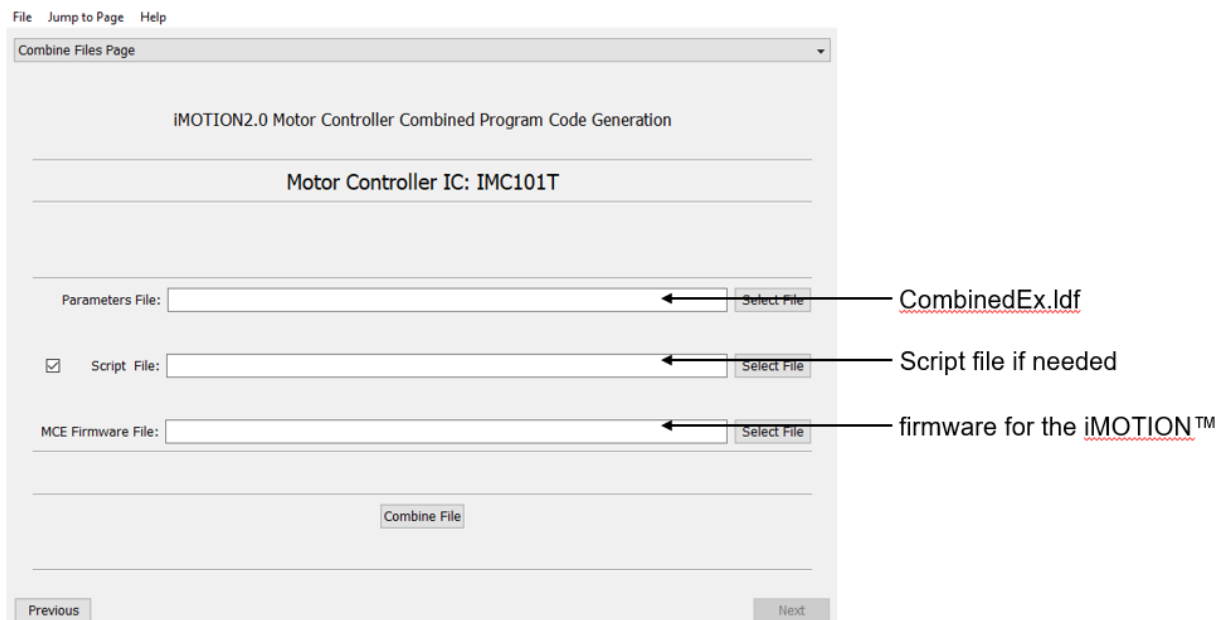
In our example, there is now a single .ldf file containing four different parameter sets.

If you are not familiar with macros and VBA functionalities, you can perform this procedure by copying and pasting into one single .ldf file all the different AppIDs, and removing the firmware.

In our example, this combined file containing our four different setups is called CombinedEx.ldf.

### 3.3 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.

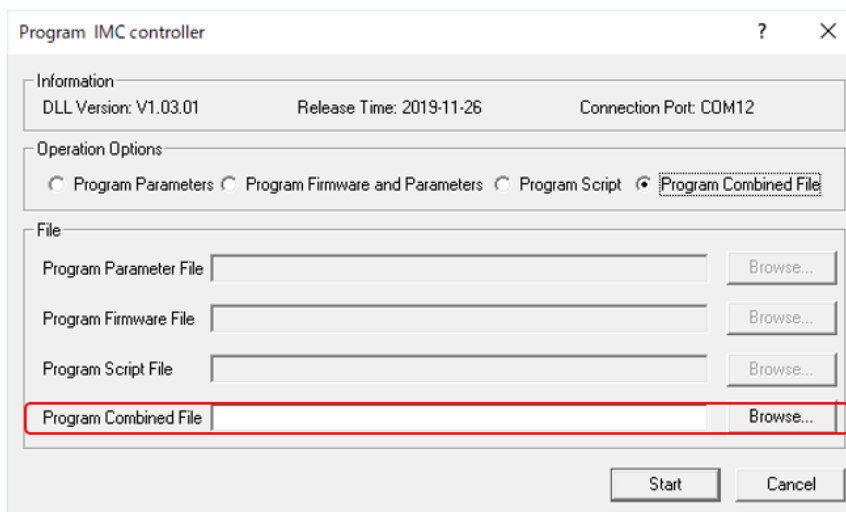


**Figure 10** Last step into creating a combined .ldf file containing all different parameter sets in MCEWizard.

### 3.4 Programing combined files with 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 MCEDesigner. In order to do so:

- 1- Select Tools --> Programmer in MCEDesigner
- 2- Select combined parameter files in “Program Combined File” and press start



**Figure 11** Programing combined .ldf file with MCEDesigner.





## Revision history

### Major changes since the last revision

Page or Reference	Description of change

#### Trademarks of Infineon Technologies AG

μHVIC™, μIPM™, μPFC™, AU-ConvertIR™, AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, CoolDP™, CoolGaN™, COOLiR™, CoolMOS™, CoolSET™, CoolSiC™, DAVE™, DI-POL™, DirectFET™, DrBlade™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPACK™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, GaNpowIR™, HEXFET™, HITFET™, HybridPACK™, iMOTION™, IRAM™, ISOFACE™, IsoPACK™, LEDrIVIR™, LITIX™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OPTIGA™, OptiMOS™, ORIGA™, PowIRaudio™, PowIRstage™, PrimePACK™, PrimeSTACK™, PROFET™, PRO-SiL™, RASIC™, REAL3™, SmartLEWIS™, SOLID FLASH™, SPOC™, StrongIRFET™, SupIRBuck™, TEMPFET™, TRENCHSTOP™, TriCore™, UHVIC™, XHP™, XMC™

Trademarks updated November 2015

#### Other Trademarks

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

#### Edition <2020-12-11>

##### Published by

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2021 Infineon Technologies AG.**

**All Rights Reserved.**

**Do you have a question about this document?**

**Email: [erratum@infineon.com](mailto:erratum@infineon.com)**

**Document reference**

**AppNote Number**

#### 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](http://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.