LogiCoA™ Power Solutions

Serial communication of RMOS and GUI developing manual

Introduction

The microcontroller for Power Supply Control ML62Q203x/ML62Q204x (hereinafter referred to ML62Q20xx group) has two communication interfaces of UART/I²C. RMOS (**R**eal time **M**icro **O**perating **S**ystem) provided by us uses UART to create a frame configuration and a packet configuration in advance and provide it as a software that enables communication control using Excel VBA. Communications between the MCU and PC can be easily performed by writing RMOS and setting commands arbitrarily in Excel VBA. The user can also easily create GUI to control communications. Another benefit is that it implements a communications protocol that can communicate with up to 32 power supplies for a single PC by software processing RMOS.

This application note introduces the protocol for serial communication using UART, explanations of communication functions (such as how to append to commands), and how to use the communication GUI developing environmental. The communication described here uses a buck DCDC converter (hereafter buck converter) EVK, LogiCoA001-EVK-001 (hereafter buck converter EVK).

Table of contents

1. Usage environment

(1) Software and Equipment

Communication using RMOS described in this manual is performed using the following.

1. Buck converter EVK
2. USB-UART convert module: FT234x FTDI (mounted on 1)
3. Windows PC(Windows10 64bit)
4. On-chip emulator "EASE1000 V2"
5. RMOS project file(file to be read into LEXIDE-Ω and used)
6. Microsoft Excel 64bit version (Operation check: Microsoft 365 MSO, Office 365 MSO)
7. Excel file "RMOS_CommunicationControl_LogiCoA001-EVK-001.xls"
8. Regulated DC power supply (Performance: Output-voltage 12V, Output current 1A or higher)
9. Digital multi-meter

Figure 1-1 shows a simplified diagram for communication. When checking the operation, disconnect the on-chip emulator and connect the measuring device to the I/O pin of the buck converter EVK.

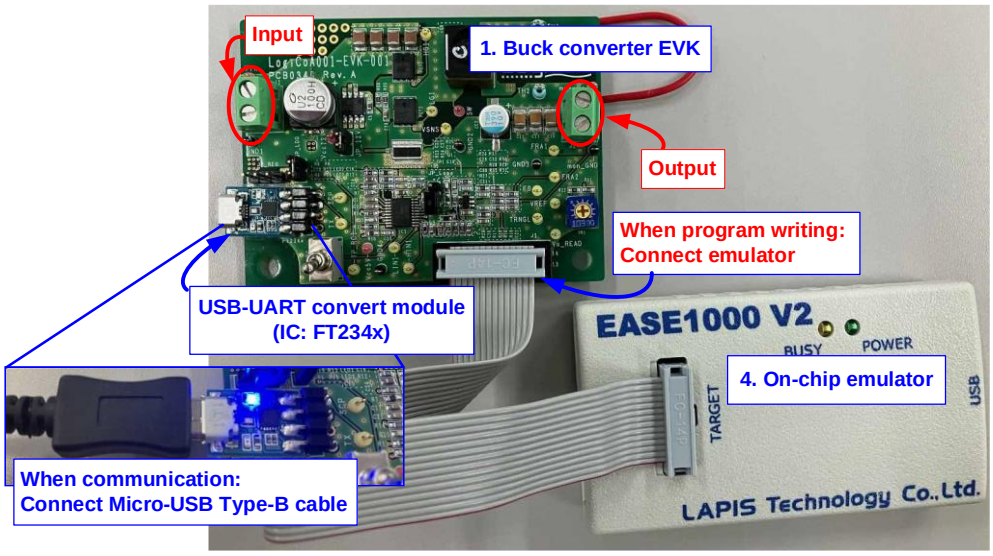

Figure 1-1. Connection diagram for communication

For details on how to connect Figure 1-1 buck converter EVK and on-chip emulator, and how to write RMOS codes, refer to RMOS application note [1].

(2) Files in RMOS folders required for communication control

Table 1-1 describes RMOS file necessary for communication control described in this manual. How to change the file name of the table and how to check the operation after changing the description are explained in the later chapters.

Table 1-1. Files in RMOS folders required for communication control

No.	Folder name	File name	Description	Changeable part
1	00_System	00S93_Func_Select.s	Function setting of OS during program debugging	Communication mode
2	10_Setting	10S01_Parameter_Init.s	Set the initial value of power supply operation parameters	Communication address
3	"	※10S70_UART_Set.s	UART assignment and operation setting	※No need to change
4	30_Info_module	30I01_InfoCMD_Exec.asm	Write the program of communication command	Add program of additional command
5	"	30I11_InfoCMD_Table_def.s	Communication command assignment	Assign labels for additional commands
6	90_Header	※93H010_InfoTxRx_Header.s	Describes a program for processing transmission and reception data	※No need to change

2. Communication protocol

This chapter describes the configuration of serial communication using UART set in RMOS. The hardware used to communicate between microcontroller and PC is UART, RMOS has a communication protocol that constitutes 6 frames per packet by software processing. This packet configuration enables one PC to perform communication control of 32 power supplies and communication from PC to the power supply circuitry using several commands.

Table 2-1 indicates UART hardware configuration and Table 2-2 indicates RMOS communication frame configuration. All the initialization settings in the table are set in "10S70_UART_Set.s" file. 2nd bit of Table 2-2 is identified by the frame identification bit, the first frame (Frm0) is identified by "1", and the remaining frames (Frm1-Frm5) are identified by "0".

Table 2-1. UART hardware configuration

	Setting
Baud rate	9600bps
Data length	8bit
communication direction	LSB first
Parity	Use, Even
Stop bit	1bit

Table 2-2. Frame configuration

bit	Assignment
1	Start
2	Data (Identification)
3	Data
4	Data
5	Data
6	Data
7	Data
8	Data
9	Data
10	Parity
11	Stop

Figure 2-1 packet configuration describes below. One packet of transmission/reception consists of 6 frames (= total 42 bits in length): address length (5 bits) + decision bit (1 bit) + data length (32 bits) + checksum length (4 bits). All the software processes that process 6 frames into one packet are described in "93H010_InfoTxRx_Header.s" file.

There are two transmission methods: TX32 and TX16 (the communication explained in this document is set by TX16). TX32 can transmitted the data of 32bit length up to 7th-38th bit of the packet to microcontroller, but the number of commands is limited to one. On the other hand, TX16 has 8bit lengths for command groups (CmdGr) and command numbers (CmdNo) from the 7th to 22nd bit, and up to 65536 types of commands can be used in combination (For the communication commands of the buck converter, areas from 0-15 for CmdGr and from 0-31 for CmdNo can be used.). The data to be transmitted to microcontroller is 16 bits in length from the 23rd-38th bit.

ADR (1st-5th bit) of Frm0 is assigned to the connected devices, and up to 32 power supplies can be connected simultaneously. 6bit of Frm0 is determined by TX32 and TX16, where "1" is TX32 and "0" is TX16. The chksum(39th-42nd bit) is an error detection and judgment bit. This bit is used to detect and judge errors in communication data of all 6 frames. In addition, the receive RX returns the data of 32bit in response to the transmitted command.

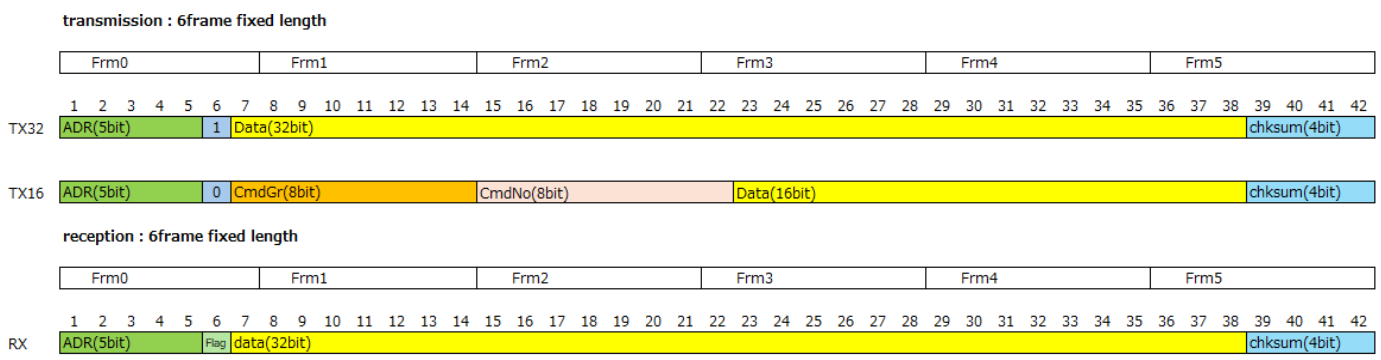

Figure 2-1. Packet configuration

3. Communication method

3-1 Initialization Settings of Excel VBA and RMOS

(1) Excel VBA setting

This section describes the initialization of Excel files required for communication between PC and microcontroller. First, start "RMOS_CommunicationControl_LogiCoA001-EVK-001.xls" file downloaded from ROHM website. The operating sheets are described in later chapters.

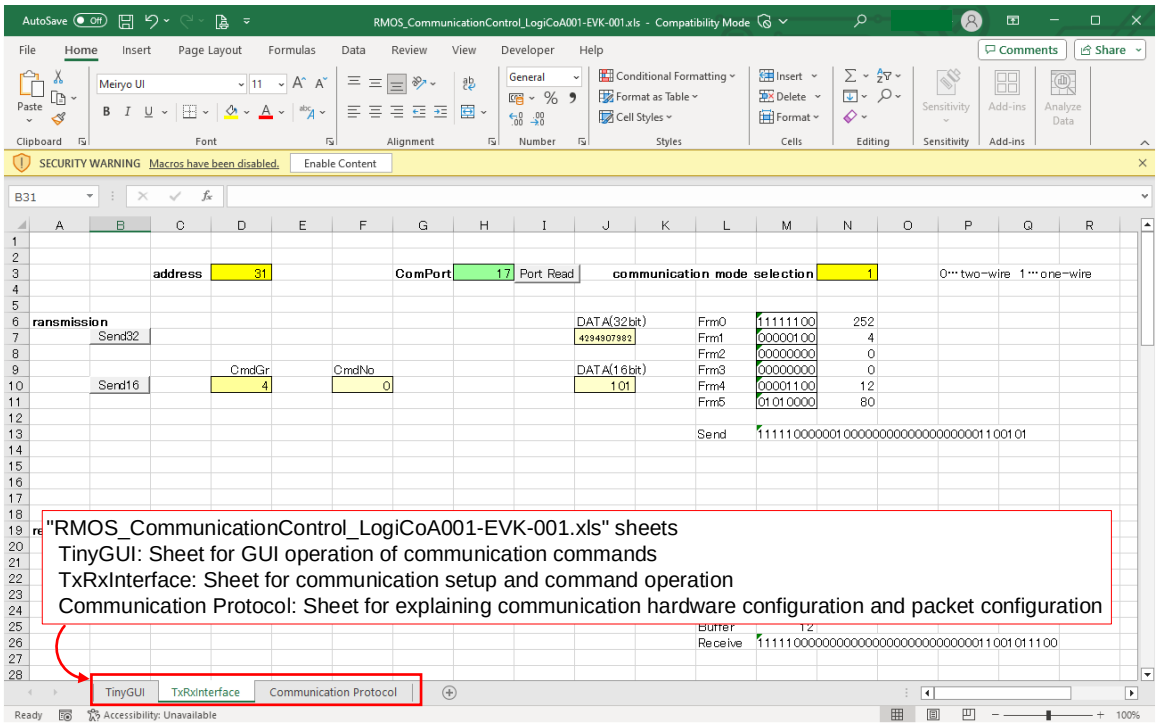

Figure 3-1. Startup window for Excel file

Next, check whether Excel is set up to use VBA macros. If "Development" tab is displayed in Excel (Figure 3-2), VBA macro can be executed.

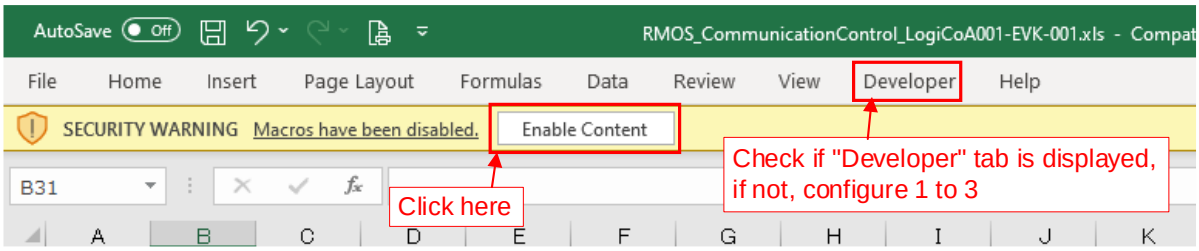

Figure 3-2. Top page of Excel

If "Developer" tab is not displayed, follow the step below to configure the settings.

1. Click, "File" tab > "Options" > "Customize Ribbon".
2. Check "Developer" box in "Customize the Ribbon" list.
3. Click, "OK".
4. check the window that "Developer" tab is displayed.

Also, the first time the user opens an Excel file, "Macro have been disabled" message is displayed. Click on "Enable Content" button to improve. Once the user does this in the first use, user will not receive the warning message the next time user launch Excel file.

(2) Setting of communication address and communication mode

Check the serial communication mode of UART and microcontroller address setting. Configure the settings so that Excel filename matches RMOS description as shown in Figure 3-3.

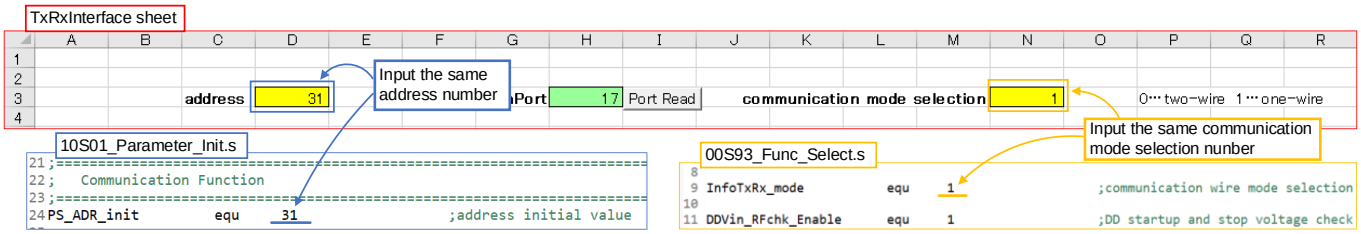

Figure 3-3. "00S93_Func_Select.s" file and Excel function setting

Blue: The value in D3 cell is the address of microcontroller. In the provided RMOS, PS_ADR_init="31" in "10S01_Parameter_Init.s" file is the address of microcontroller. There is no need to change the address in this manual. However, when connecting more than one power supply to PC for communication control, set the address of each power supply on RMOS and specify the address of the target power supply in Excel.

Yellow: N3 cell is used to change UART communication mode. The communication in LogiCoA™ power solution has two connection methods: two wires and one wire. "InfoTxRx_mode" on line 9 of "00S93_Func_Select.s" file can be set to "0" for two wires and "1" for one wire connection mode. The buck converter EVK described in this manual is designed for one wire communication.

(3) Checking and setting COM port number

Please acquire COM port number of TxRxInterface sheet. COM port is a type of communication port for connecting PC to an external device and is used for serial communication.

When USB cable is connected to USB-UART convert module, Figure 3-4 will show "USB Serial Port (COM6)" in Windows Device Manager (Driver will install automatically if connected to a network). The number "COM*" is COM port number. The number varies depending on PC and USB-UART convert module.

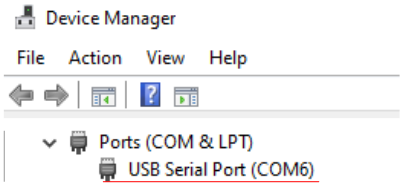

Figure 3-4. Device manager window

NOTE: Even if the USB cable is connected, the driver may not be automatically installed and Figure 3-4 window may not display. In this case, the driver must be downloaded and installed from FTDI website.

This COM number must be specify when communicating with Excel file. In the TxRxInterface sheet, VBA macro automatically acquires COM port number of USB module. When click "Port Read" on Figure 3-5, the number is displayed in H3 cell.

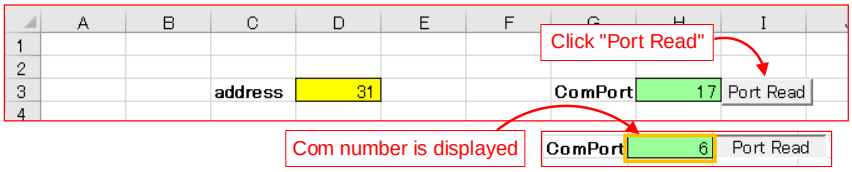

Figure 3-5. TxRxInterface sheet port read key

If communication between PC and the MCU fails, first check that COM port number has been acquired correctly. If more than one USB is connected, the port number may not be acquired from Excel file. To resolve this, click "Port Read" again, or check COM number in Device Manager and input COM number in H3 cell.

3-2 How to use TxRxInterface Sheet

Perform produces 1 to 4 below to perform communication using TxRxInterface sheet.

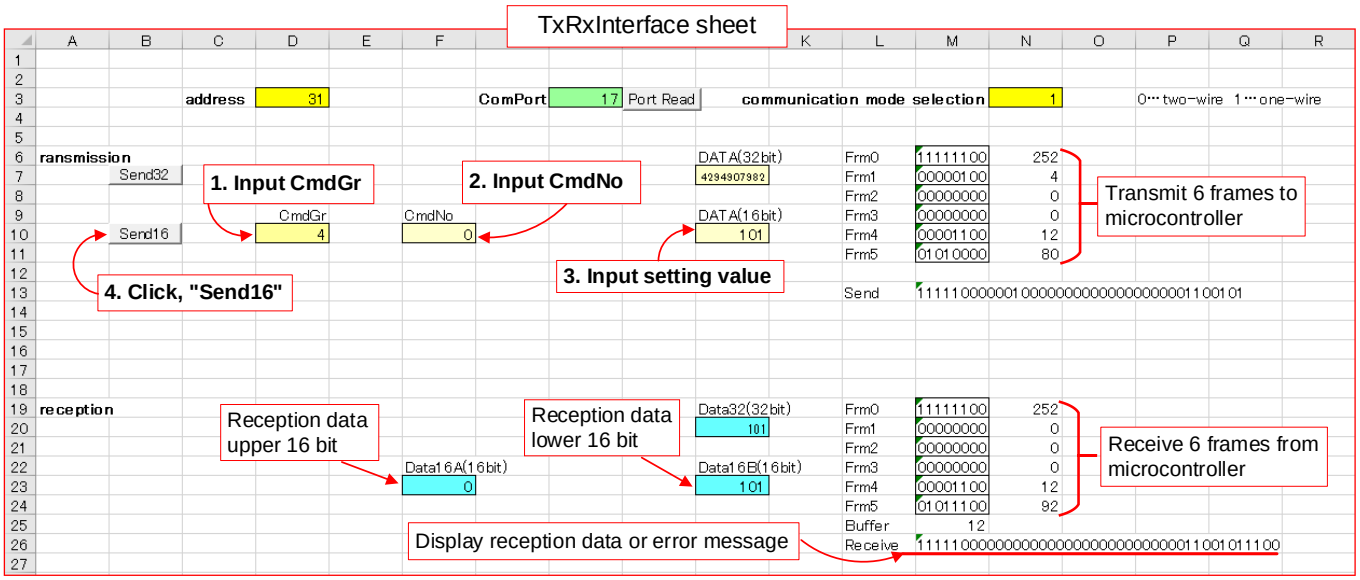

Figure 3-6. TxRxInterface sheet

1. Input CmdGr (command group) in decimal.
2. Input CmdNo (command number) in decimal.
3. Input DATA in decimal.

   Figure 3-6 is inputting commands to change the D/A converter code in microcontroller. By inputting CmdGr="4" and CmdNo="0", the code value of the microcomputer D/A converters can be changed to the value inputted in DATA. Depending on the command, certain register values in microcontroller can be received. See Section 4 "Communication Commands List" for the commands that can be communicated with the buck converter EVK.

4. Click, "Send16".

   When the button is clicked, data is transmitted to microcontroller, and the value is displayed in the cell of the receiving item below the 19th line when communication is successfully performed. In Figure 3-6, since the setting value is "101", "101" is displayed in J23 cell.

M26 cell displays 6 frames of received data and error messages. An error message is displayed when USB cable is not connected to the USB-UART convert module, when a communication error occurs, or when there is no description of the communication command in RMOS.

The description of VBA macro registered in this sheet does not need to be changed because various GUI are created based on VBA macro in TxRxInterface sheet.

3-3 How to use TinyGUI Sheet

This section explains how to use TinyGUI sheet. This sheet simplifies manual operation on TxRxInterface sheet so that communication can be performed only by button operation using VBA macro.

Provided Excel file incorporate VBA macros to change the output voltage of the buck converter EVK in TinyGUI sheet. Figure 3-7 shows the macro button on TinyGUI sheet. VBA macro is registered in the buttons 1 to 4. Figure 3-8 shows the description of VBA macros registered in the buttons 1 to 4.

This chapter describes an experimental system which 12V is applied to the input voltage of the buck converter EVK as an example.

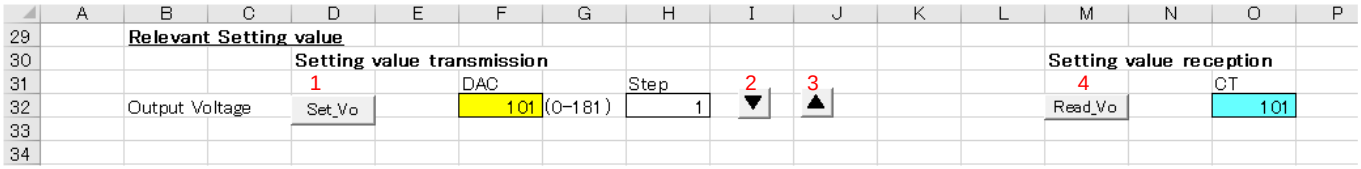

Figure 3-7. Macro button on TinyGUI sheet

Figure 3-8. VBA macros registered in 1 to 4

1. Set_Vo button: Transmits the value of F32 cell to microcontroller.
2. Down button: Subtracts the code of D/A converter for each number written in H32 cell and transmits the value to microcontroller.
3. Up button: Adds the code of D/A converter for each number written in H32 cell and transmits the value to microcontroller.
4. Read_Vo button: The D/A converter code is received from microcontroller and the received value is displayed in O32 cell.

The output-voltage Vo of the buck converter EVK is determined by the code of D/A converter, and the following equation is established.

$$V_o = \frac{VDD}{256} \times \mathrm{D/A\ code} \times \left(\frac{R_{29} + R_{30} + R_{31}}{R_{31}}\right)$$

...VDD(microcontroller supply voltage)=5V, R_{29}=51Ω, R_{30}=3.3kΩ, R_{31}=2.2kΩ

The output voltage Vo of the buck converter EVK can be changed using the setting code calculated from the above equation (Ex. 3.3V="67, 5V="101 and 9V="181). In actual operation, after inputting the value of F32 cell, click button and check that the output voltage of buck converter EVK has changed.

3-4 Commands implemented in RMOS

This section describes the program of the communication commands already implemented in RMOS (version "RMOSVer=1.00, OSBuildNo=007, PSFMNo=001, PSFMVer=1.00, PSFMBuildNo=004"). The following two types of commands are described in Section 4 "Communication Commands": Read system commands (commands to read microcontroller values) and Set system commands (commands to write to the MCU).

Table 3-1 summarizes the files in "30 Info_module" folder described in this chapter. Figure 3-9 shows the window in "30 Info_module" folder of LEXIDE-Ω.

Table 3-1. Files in "30_Info_module" folder

No.	File name	Description	Changes
1	30I01_InfoCMD_Exec.asm	Write the program of communication command.	Add program of additional command
2	30I11_InfoCMD_Table_def.s	Communication command assignment	Assign labels for additional commands

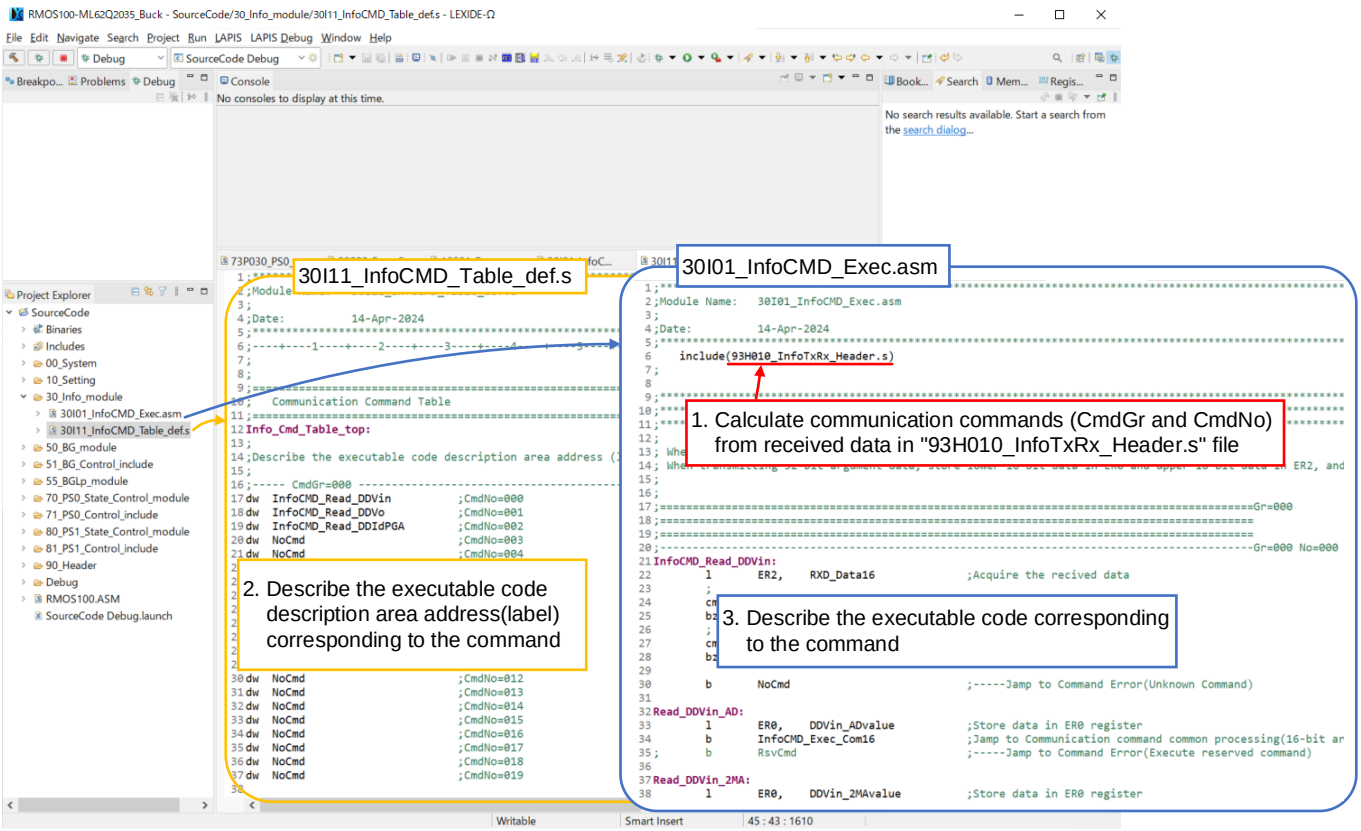

Figure 3-9. Displaying files in "30_Info_module" folder under LEXIDE-Ω
("1 to 3" is the processing of communication data.)

1. TX16 of received data from PC is processed in "93H010_InfoTxRx_Header.s" file. And CmdGr and CmdNo are calculated.
2. Calculated command value indicates the address value of the communication command table in "30I11_InfoCMD_Table_def.s" file.
3. Jump to the label in "30I01_InfoCMD_Exec.asm" file that matches the read label name in 2 and executes the corresponding program.

Note that "93H010_InfoTxRx_Header.s" file has a description that processes the transmitted and received data. Normally, it does not need to be edited.

Read and Set commands described in "30I11_InfoCMD_Table_def.s" file and "30I01_InfoCMD_Exec.asm" file are described in (1), (2) below.

(1) Read Command

Figure 3-10 shows the descriptions and operations of Read command. The following programs describe how to read an input voltage AD value or 2 times moving average value (when CmdGr = "0" and CmdNo = "0").

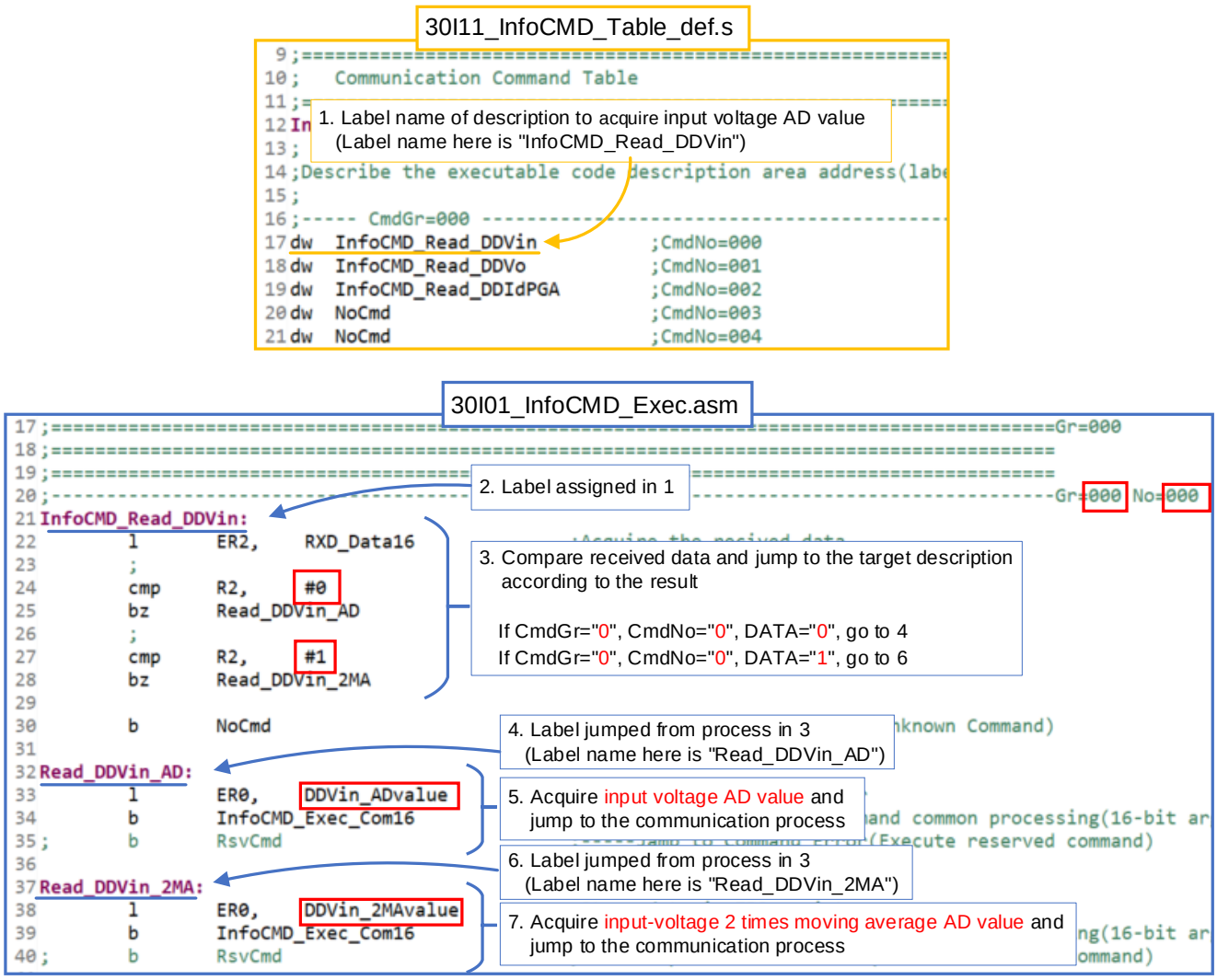

Figure 3-10. Description and operation of read command

1. The beginning label name of the description to acquire the input voltage AD value is assigned in "30I11_InfoCMD_Table_def.s" file. The received data is processed in "93H010_InfoTxRx_Header.s" file and jumped to "InfoCMD_Read_DDVin" label when CmdGr = "0" and CmdNo = "0".
2. The label name assigned in 1 is described, and the description below the label is executed.
3. Compare data received from PC and jump target descriptions accordingly.

   If CmdGr = "0", CmdNo = "0", DATA = "0", go to 4
   If CmdGr = "0", CmdNo = "0", DATA = "1", go to 6

4. The label jumped from the processing of 3 is described. Then, the label name is "Read_DDVin_AD".
5. Acquire "DDVin_ADvalue"(input voltage AD value) and jump to the communication process to transmit the return value to PC.
6. The label jumped from the processing of 3 is described. Then, the label name is "Read_DDVin_2MA".
7. Acquire "DDVin_2MAvalue"(input voltage 2 times moving average AD value) and jump to the communication process to transmit the return value to PC.

(2) Set Command

Figure 3-11 shows the descriptions and operations of Set command. The following commands describe how to acquire and change the output voltage (when CmdGr = "4" and CmdNo = "0").

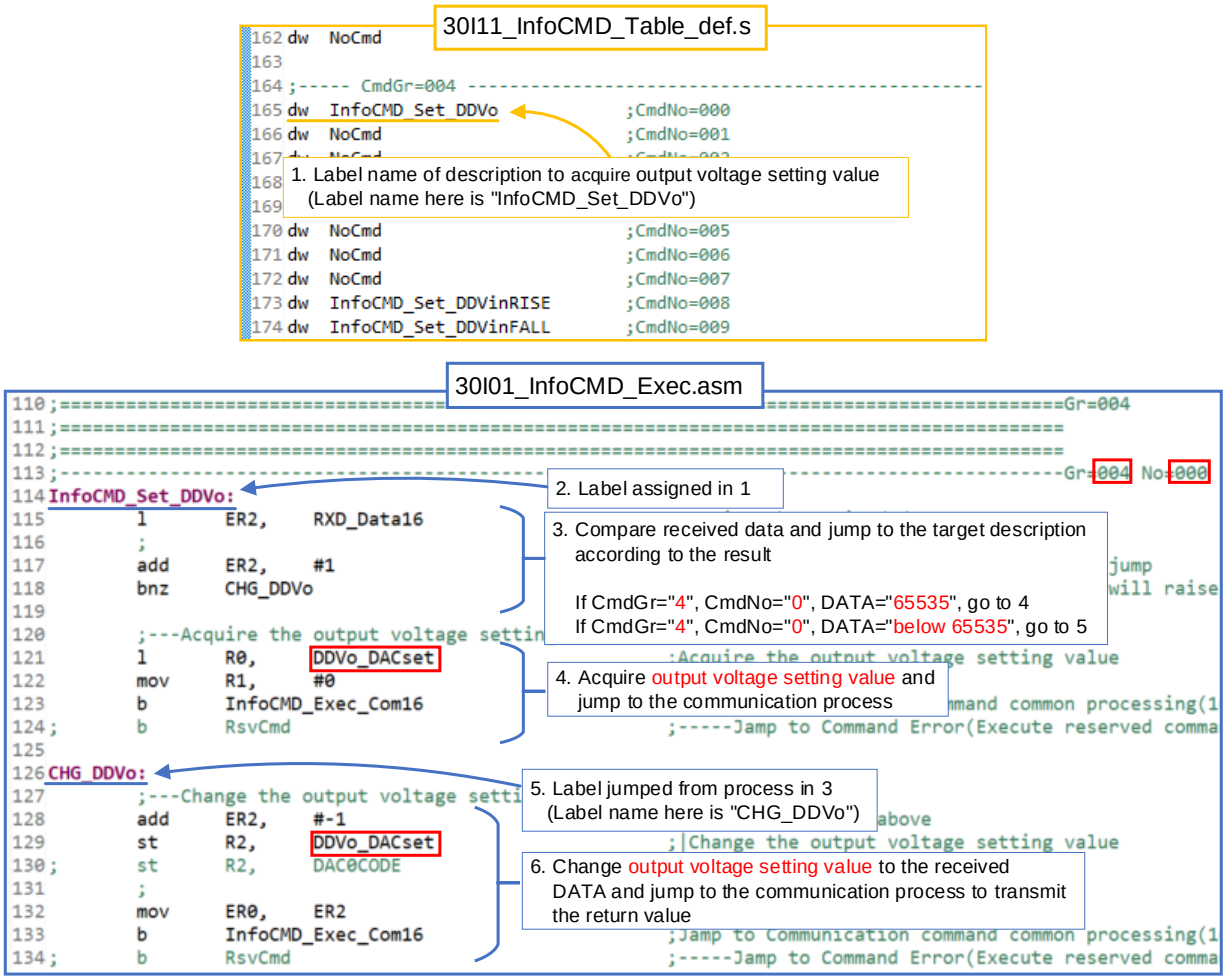

Figure 3-11. Description and operation of set command

1. The beginning label name of the description that acquires or changes the output voltage is assigned in "30I11_InfoCMD_Table_def.s" file. The data is processed in "93H010_InfoTxRx_Header.s" file and jumped to "InfoCMD_Set_DDVo" label when CmdGr = "4" and CmdNo = "0".
2. The label name assigned in 1 is described, and the description below the label is executed.
3. Compare data received from PC and jump target descriptions accordingly.

   If CmdGr = "4", CmdNo = "0", DATA = "65535", go to 4
   If CmdGr = "4", CmdNo = "0", DATA = "below 65535", go to 5

4. Acquire "DDVo_DACset"(output-voltage setting value) and jump to the communication process to transmit the return value to PC. This description is the same as Read command. When DATA="65535" is inputted, "DDVo_DACset" is read from microcontroller.
5. The label jumped from the processing of 3 is described. Then, the label name is "CHG_DDVo".
6. Change "DDVo_DACset" to the value of DATA and jump to the communication process to transmit the return value to PC. The value of "DDVo_DACset" in microcontroller can be changed by this description.

Please use TxRxInterface sheet of Excel file to check the operation of the communication commands (Refer to 3-3 "How to use TinyGUI sheet"). Even if DATA="65535" is transmitted immediately after Excel is started, the initialization setting is "101", so the reception status remains unchanged.

3-5 How to Add Command

(1) How to Add Set Command

In the Set commands in Chapter 4 "Communication Commands List", some of the commands are not described except for CmdGr="4" and CmdNo="0". Therefore, it is necessary to add them.

Figure 3-12 shows an example of adding a communication command to change the startup voltage setting value in steps 1 to 3. To use a command other than this one, write the corresponding state variable name from Table 4-1 in the red box.

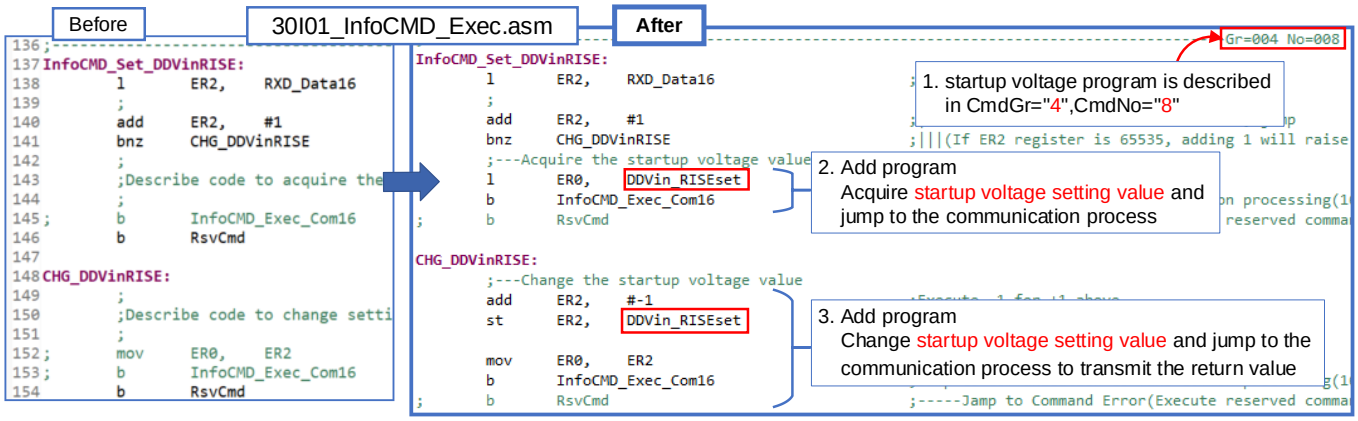

Figure 3-12. How to expand communication command of start voltage

1. The program to set the starting voltage value "DDVin_RISEset" is written in CmdGr="4" and CmdNo="8".
2. Acquire "DDVin_RISEset"(startup voltage setting value) and jump to the communication process to transmit the return value to PC.
3. Change "DDVin_RISEset" to the value of DATA and jump to the communication process to transmit the return value to PC.

(2) How to add command in UserFree area

This section describes how to read and change state variables not listed in Chapter 4, "Communication Commands List". Follow the steps 1 to 3 below to add more commands. Program area that can be freely described by user are provided in CmdGr="15".

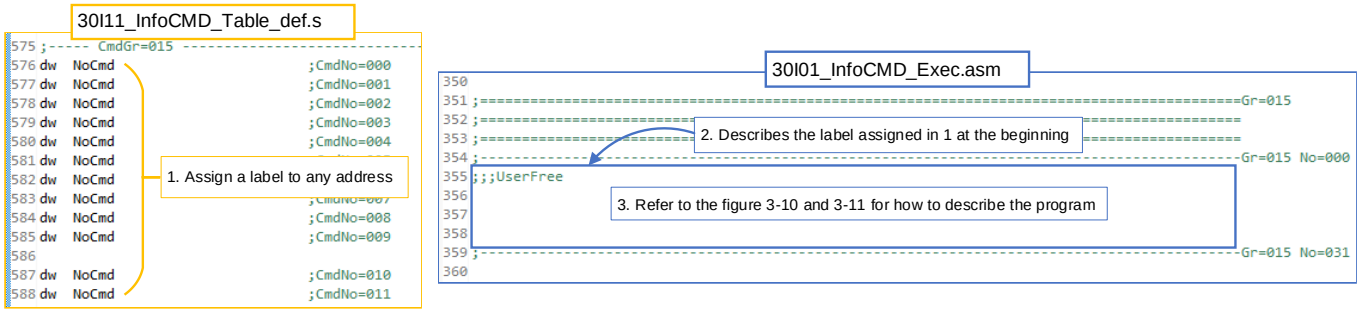

Figure 3-13. How to expand communication command of startup voltage

1. Assign labels to any CmdNo.
2. Describe the label assigned in 1 at the beginning of the description.
3. Describe the command processing. Refer to the figure 3-10 and 3-11 for how to describe the program.

3-6 How to add GUI

This chapter describes how to operate the TxRxInterface sheet using the GUI. This makes debugging the power supply easier, as power supply parameters can be easily changed and acquired. GUI can create in TinyGUI sheet of Excel file.

User can add GUI as shown in steps 1 to 5 below. In this chapter, GUI is created by referring to Chapter 3-5 (1) "How to add Set command" and changing the startup voltage (CmdGr="4" and CmdNo="8").

1. Copy macro buttons and cells in a sheet. And rename macro buttons.

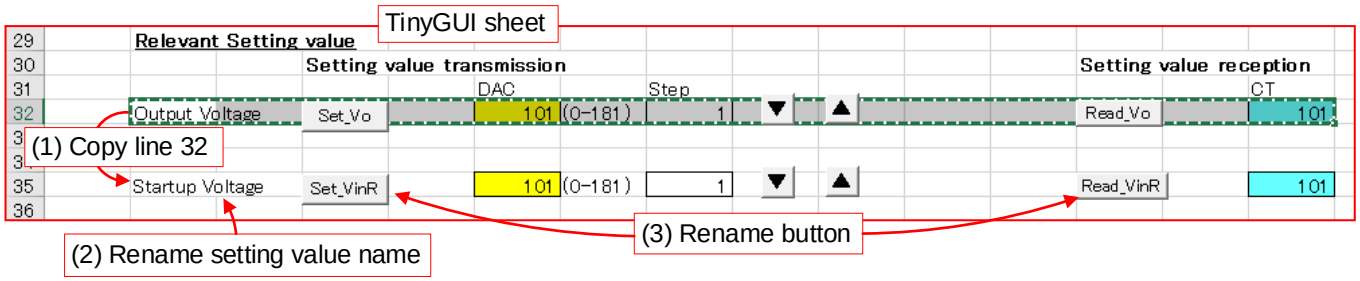

Figure 3-14. Step on TinyGUI sheet

2. Copy the pre-registered VBA macro (Figures 3-8) to the line below the arrow.

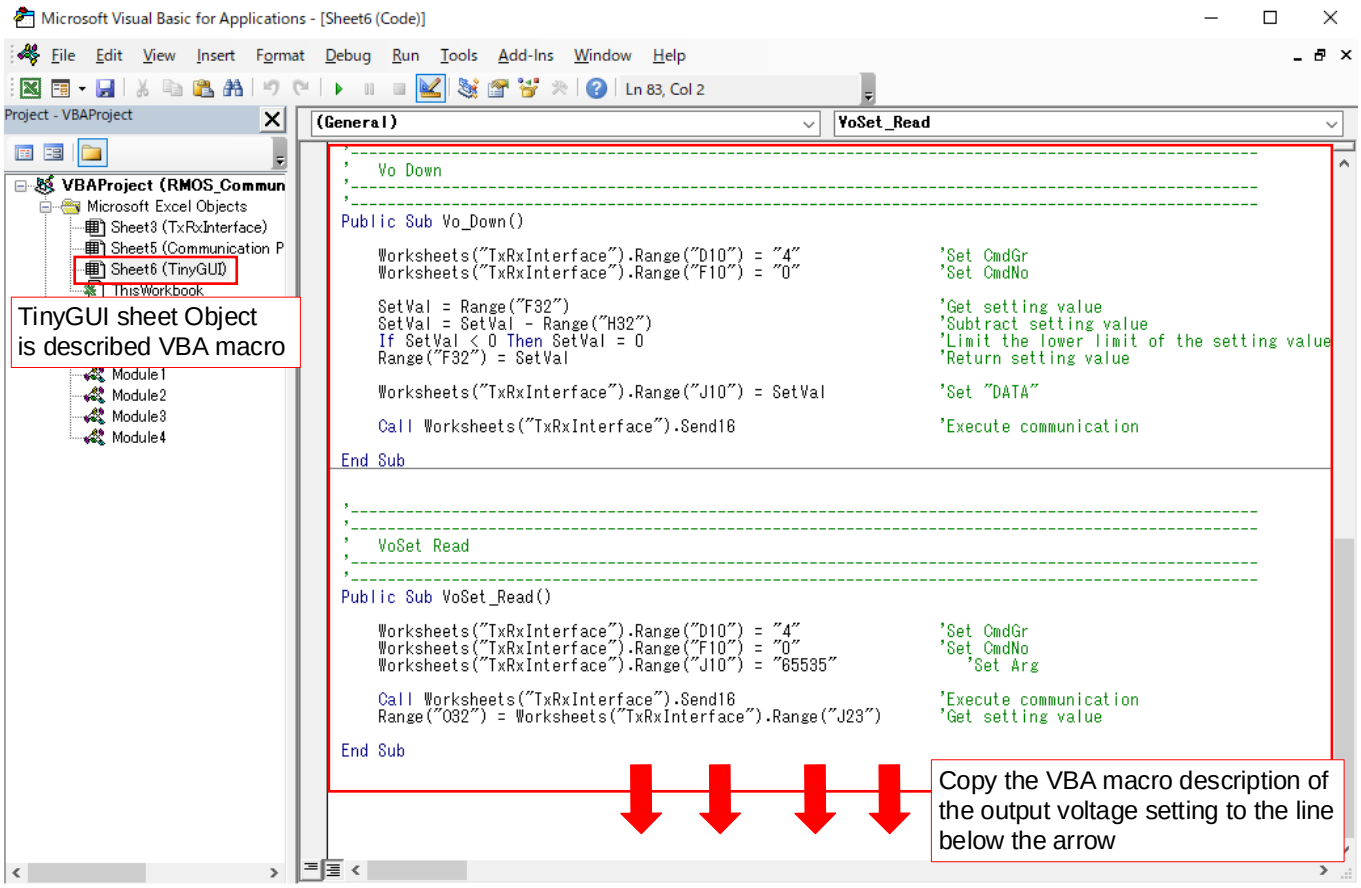

Figure 3-15. Steps on Excel VBA window

3. VBA macro is rewritten according to the description on RMOS or the cell copied in Step 1.

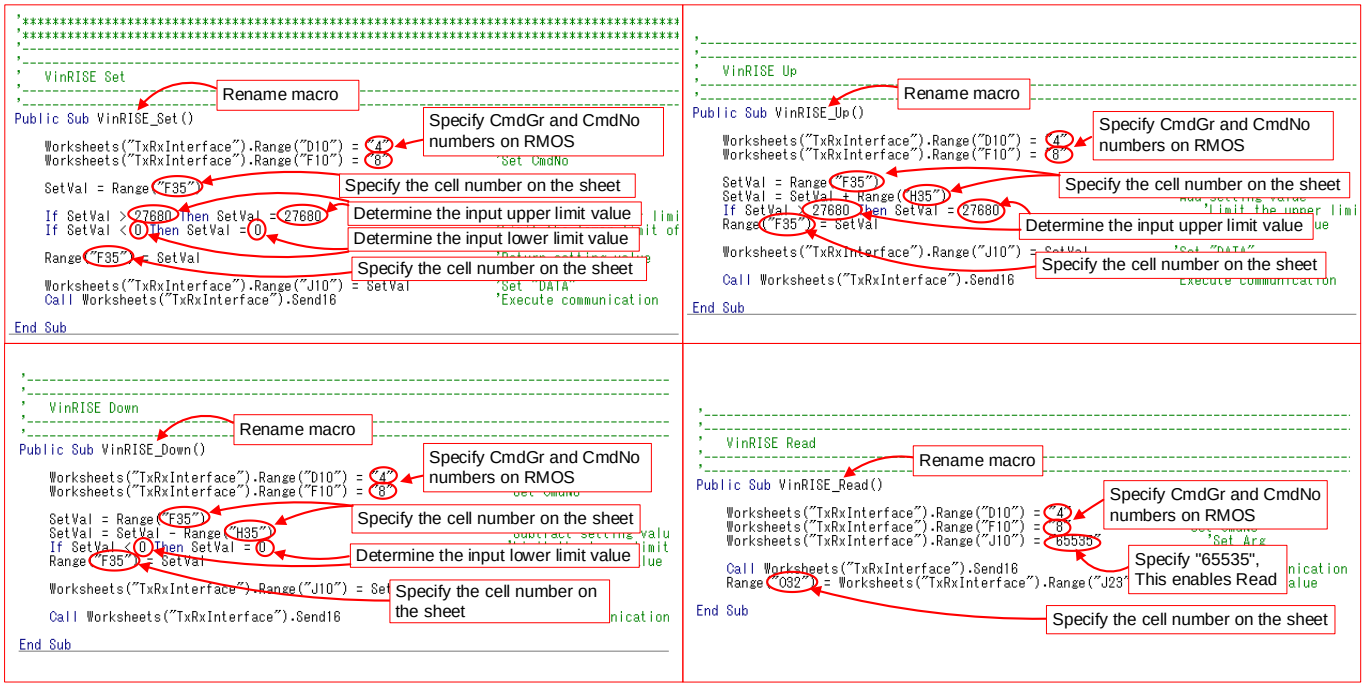

Figure 3-16. Excel VBA macro setting

4. Register the macro name added with VBA to the macro button. (Steps of (1) to (5))

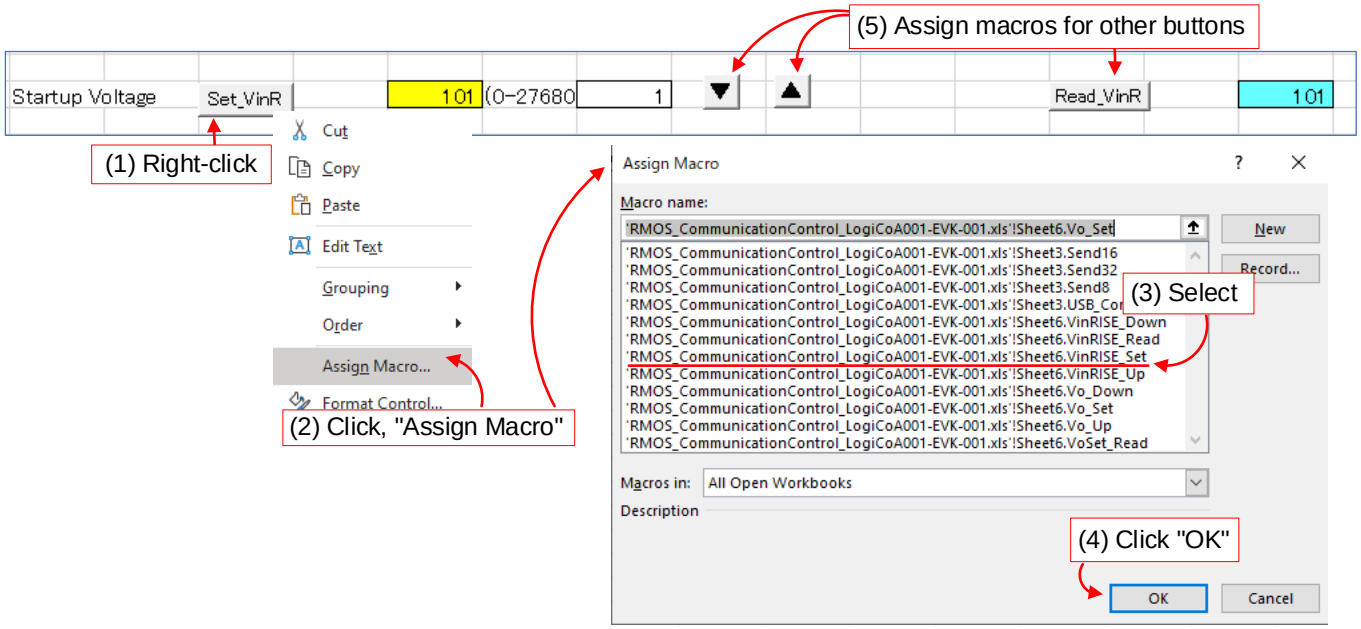

Figure 3-17. Steps for registering VBA macro

5. Check the operation of the created GUI. (Steps of (1) to (6))

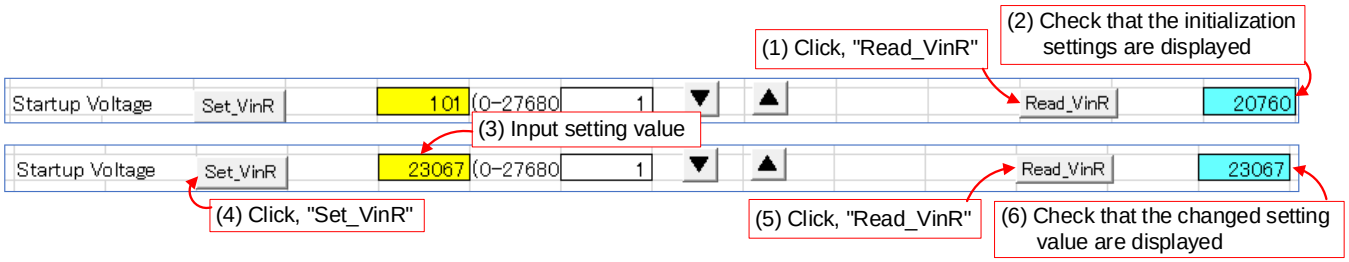

Figure 3-18. Step for checking GUI operation

User can easily create a GUI for communication using the above steps.

In GUI shown in step 5, the startup voltage inputs as AD value, but it can also input as a voltage value by the function of Excel file. For the calculation method for converting startup voltage AD value to the voltage value, refer to the User's Guide [2] and Operating Manual [3] of the buck converter.

Parameters such as stop voltage, dead time and OCP (over current protection) can also add to GUI in the same step as the startup voltage.

4. Communication Commands List

Table 4-1 contains a list of CmdGr and CmdNo functions.

In RMOS (version "RMOSVer=1.00, OSBuildNo=007, PSFMNo=001, PSFMVer=1.00, PSFMBuildNo=004") in this manual, CmdGr of 1 to 15 can be used, and CmdNo of 1 to 31 can be used. Please note that the command's configuration in the table 4-1 may change due to future RMOS upgrades.

Table 4-1. Communication Commands List

(1) Read commands

CmdGr	CmdNo	DATA	Command label name	State variable name	Command description
0	0	0	InfoCMD_Read_DDVin	DDVin_ADvalue	Input voltage AD value
		1		DDVin_2MAvalue	2 times moving average AD
	1	0	InfoCMD_Read_DDVo	DDVo_ADvalue	Output voltage AD value
		1		DDVo_2MAvalue	2 times moving average AD
	2	0	InfoCMD_Read_DDIdPGA	DDIdPGA_ADvalue	Drain current AD value
		1		DDIdPGA_2MAvalue	2 times moving average AD
		2		DDIdPGA_8MAvalue	8 times moving average AD

(2) Set commands

CmdGr	CmdNo	DATA	Command label name	State variable name	Command description
4	0	Value	InfoCMD_Set_DDVo	DDVo_DACset	Output Voltage setting
	8	Value	InfoCMD_Set_DDVinRISE	DDVin_RISEset	Startup voltage setting
	9	Value	InfoCMD_Set_DDVinFALL	DDVin_FALLset	Stop voltage setting
5	0	Value	InfoCMD_DD0_Fsw	Fsw_CTset	Switching Frequency setting
	16	Value	InfoCMD_DD0_TonMax	dmax_CTset	Switching device maximum duty setting
6	0	Value	InfoCMD_DD0_OCP	DDOCP_Ioset	OCP setting
7	0	Value	InfoCMD_DD0_OVP	DDOVP_VoADset	OVP setting
	16	Value	InfoCMD_DD0_LVP	DDLVP_VoADset	LVP setting
11	0	Value	InfoCMD_DD0_DeadTime0	DTimeHoffLon_CTset	Dead time 0 count value setting
	1	value	InfoCMD_DD0_DeadTime1	DTimeLoffHon_CTset	Dead time 1 count value setting

Please note that CmdGr and CmdNo that are not specified above are reserved areas and may add programs in future RMOS upgrades or EVK releases.

(3) UserFree commands

CmdGr	CmdNo	DATA	Command label name	State variable name	Command's description
15	0	-	Can be freely described	-	-
	*	*	*	*	*
	*	*	*	*	*
	31	-	-	-	-

5. References

[1] 66AN147E, Rev.001, Operating system for switching power control MCU "RMOS"

[2] 66UG090E, Rev.001, Synchronous Buck DCDC Converter Evaluation Board LogiCoA001-EVK-001

[3] 66AN153E, Rev.001, Analog-Digital Hybrid Control Power Supply Synchronous Buck DCDC Converter Operating Instructions

Revision History

Date	Revision Number	Description
5. June. 2024	001	Initial release

Notice

1) The information contained in this document is intended to introduce ROHM Group (hereafter referred to asROHM) products. When using ROHM products, please verify the latest specifications or datasheets before use.

2) ROHM products are designed and manufactured for use in general electronic equipment and applications (such as Audio Visual equipment, Office Automation equipment, telecommunication equipment, home appliances, amusement devices, etc.) or specified in the datasheets. Therefore, please contact the ROHM sales representative before using ROHM products in equipment or devices requiring extremely high reliability and whose failure or malfunction may cause danger or injury to human life or body or other serious damage (such as medical equipment, transportation, traffic, aircraft, spacecraft, nuclear power controllers, fuel control, automotive equipment including car accessories, etc. hereafter referred to as Specific Applications). Unless otherwise agreed in writing by ROHM in advance, ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties arising from the use of ROHM Products for Specific Applications.

3) Electronic components, including semiconductors, can fail or malfunction at a certain rate. Please be sure to implement, at your own responsibilities, adequate safety measures including but not limited to fail-safe design against physical injury, and damage to any property, which a failure or malfunction of products may cause.

4) The information contained in this document, including application circuit examples and their constants, is intended to explain the standard operation and usage of ROHM products, and is not intended to guarantee, either explicitly or implicitly, the operation of the product in the actual equipment it will be used. As a result, you are solely responsible for it, and you must exercise your own independent verification and judgment in the use of such information contained in this document. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties arising from the use of such information.

5) When exporting ROHM products or technologies described in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, such as the Foreign Exchange and Foreign Trade Act and the US Export Administration Regulations, and follow the necessary procedures in accordance with these provisions.

6) The technical information and data described in this document, including typical application circuits, are examples only and are not intended to guarantee to be free from infringement of third parties intellectual property or other rights. ROHM does not grant any license, express or implied, to implement, use, or exploit any intellectual property or other rights owned or controlled by ROHM or any third parties with respect to the information contained herein.

7) No part of this document may be reprinted or reproduced in any form by any means without the prior written consent of ROHM.

8) All information contained in this document is current as of the date of publication and subject to change without notice. Before purchasing or using ROHM products, please confirm the latest information with the ROHM sales representative.

9) ROHM does not warrant that the information contained herein is error-free. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties resulting from errors contained in this document.

ROHM SEMICONDUCTOR

Thank you for your accessing to ROHM product informations.
More detail product informations and catalogs are available, please contact us.

ROHM Customer Support System

https://www.rohm.com/contactus