tinyMicon MatisseCORE™

Peripheral Window User’s Guide

Matisse GUI application for reading and writing peripheral register values.

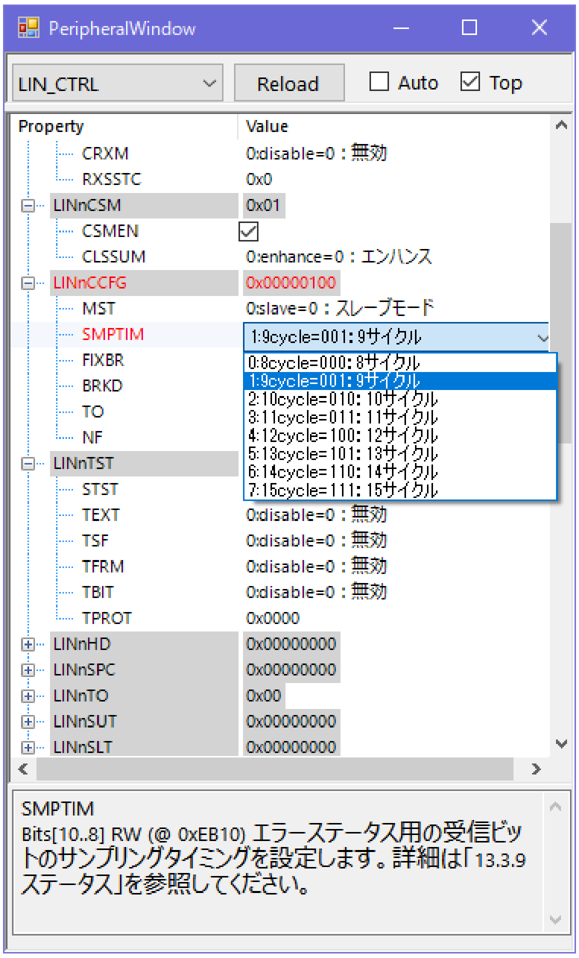
Revision History

Date	Version	Contents
2021/01/26	Rev.001	Create new
2021/11/09	Rev.002	Add function Ver.2.0
2022/12/02	Rev.003	The following was added ・Support for keyboard operation ・Support for register array definitions

Table of contents

1 Introduction

Peripheral Window can be used in the development environment of ROHM's original 8-bit CPU (Matisse)、Peripheral register values can be visually displayed and edited in the Matisse development environment.



Property	Value
CRXM	0:disable=0：無効
RXSSTC	0x0
LINnCSM	0x01
CSMEN	☑
CLSSUM	0:enhance=0：エンハンス
LINnCCFG	0x00000100
MST	0:slave=0：スレーブモード
SMPTIM	1:9cycle=001: 9サイクル
FIXBR	
BRKD	
TO	
NF	
LINnTST	
STST	
TEXT	0:disable=0：無効
TSF	0:disable=0：無効
TFRM	0:disable=0：無効
TBIT	0:disable=0：無効
TPROT	0x0000
LINnHD	0x00000000
LINnSPC	0x00000000
LINnTO	0x00
LINnSUT	0x00000000
LINnSLT	0x00000000

0:8cycle=000: 8サイクル
1:9cycle=001: 9サイクル
2:10cycle=010: 10サイクル
3:11cycle=011: 11サイクル
4:12cycle=100: 12サイクル
5:13cycle=101: 13サイクル
6:14cycle=110: 14サイクル
7:15cycle=111: 15サイクル

SMPTIM
Bits[10..8] RW (@ 0xEB10) エラーステータス用の受信ビットのサンプリングタイミングを設定します。詳細は「13.3.9 ステータス」を参照してください。

Figure 1. Peripheral Window

2 Overview

The following figure shows the relationship between the Peripheral window and the Matisse development environment.

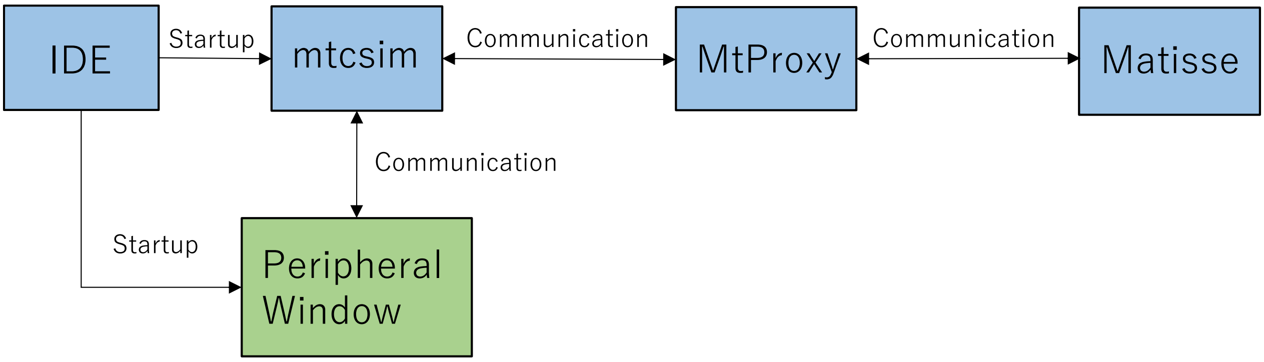


Figure 2. System diagram of Peripheral Window

Peripheral Window has the following feature

- Runs from Matisse IDE(matiseye^{TM}-studio).
- Peripheral registers can be read and written (displayed and edited in three different patterns depending on the definition of the register field).
- Peripheral registers (multi-byte) can be written in the order of upper to lower byte.
- Displays the names and structures of peripheral registers in a tree format on the screen.
- Displays additional information about the registers.

3 Technical terms

Table 1. Technical terms

Terminology	Descriptions
Matisse	Small 8-bit CPU core
mtcsim	C language simulator and debugger tool for Matisse
MtProxy	Tool for absorbing differences in communication processing methods between matiseye^{TM}-adapter and matiseye^{TM}-adapter Pro.
IDE	Integrated development environment
Visual Studio Code (VS Code)	A multi-functional and lightweight code editor developed by Microsoft
matiseye^{TM}–studio	C language development tool for matisse based on Visual Studio Code
Peripheral	Peripheral circuits for CPUs. Timers, etc.

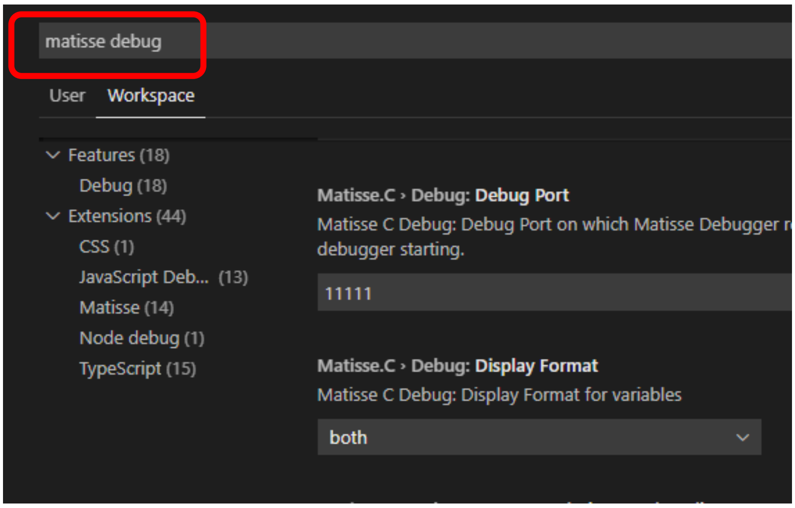
4 Setup

Settings for launching the Peripheral Window from VS Code.

4.1 Configuration Item

Open the VS Code configuration editor (keystroke... Ctrl+,).

Type "matisse debug" in the search space.



Matisse.C › Debug: **Debug Port**

Matisse C Debug: Debug Port on which Matisse Debugger r
debugger starting.

11111

Matisse.C › Debug: **Display Format**

Matisse C Debug: Display Format for variables

both

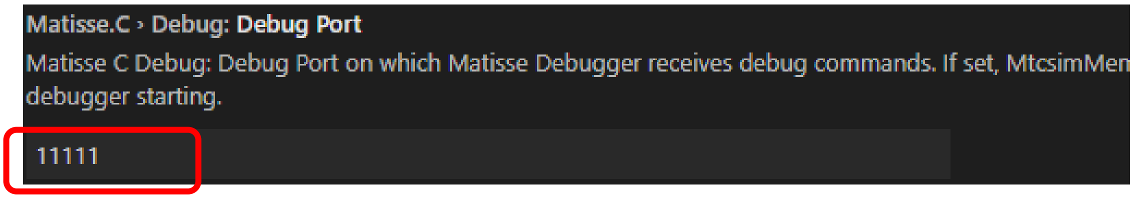
Figure 3. VS Code Configuration Editor

There are two tabs, User and Workspace, which allow you to configure global settings and project settings respectively.

In Workspace, you can set the following two items.

4.1.1 Port Number

The port number can be set in "Debug Port".



Matisse.C › Debug: **Debug Port**

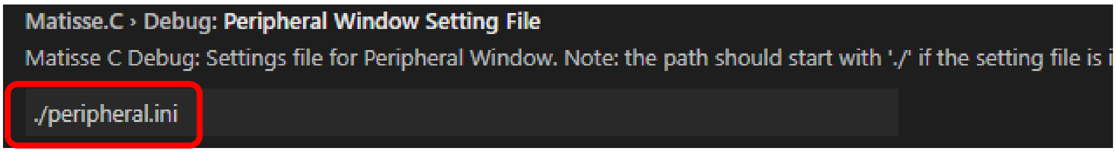
Matisse C Debug: Debug Port on which Matisse Debugger receives debug commands. If set, MtcsimMen
debugger starting.

11111

Figure 4. Setting the port number

4.1.2 Setting File

The path to the configuration file can be set in "Peripheral Window Setting File".



Matisse.C › Debug: **Peripheral Window Setting File**

Matisse C Debug: Settings file for Peripheral Window. Note: the path should start with './' if the setting file is i

./peripheral.ini

Figure 5. Setting the Path to the Configuration File

A description of the setting file is given in the user interface usage file.

5 Launch Peripheral Window

The Peripheral Window can be launched from **VS Code** or **command line**.

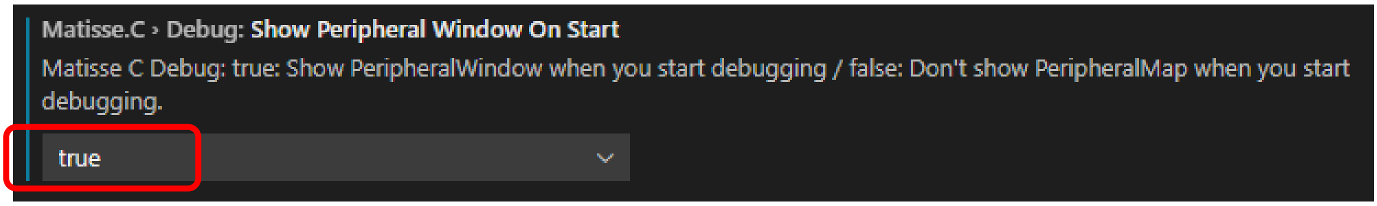
Each of these methods is explained below.

5.1 Launch from VS Code

There are two ways to start Peripheral Window: at the start of VS Code's debug mode and during execution.

5.1.1 At the start of debug mode

Set "Show Peripheral Window On Start" in the Configuration Editor to true.



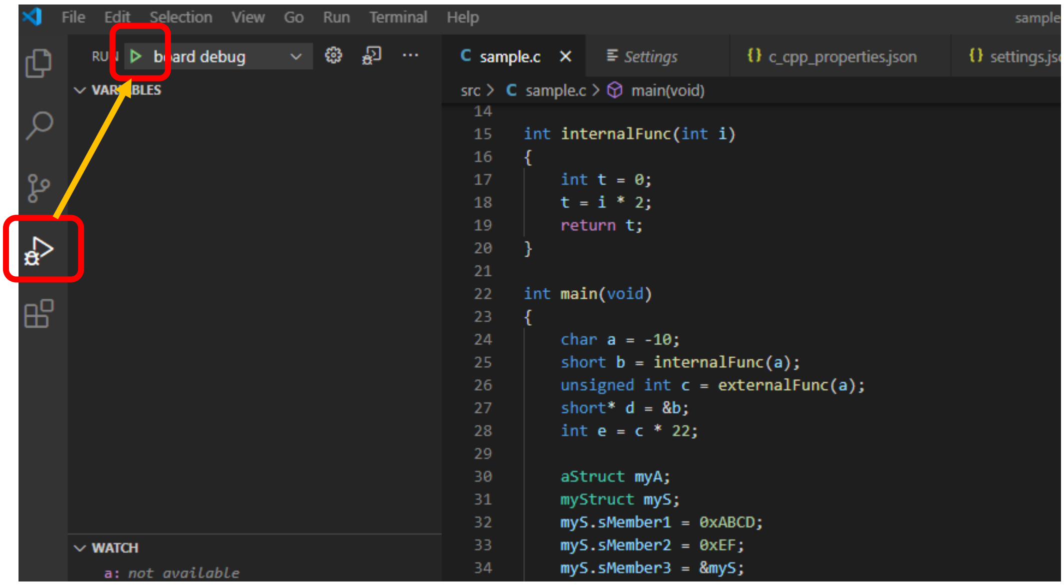
Matisse.C › Debug: **Show Peripheral Window On Start**

Matisse C Debug: true: Show PeripheralWindow when you start debugging / false: Don't show PeripheralMap when you start debugging.

true

Figure 6.Peripheral Window startup settings in the configuration editor

Next, from the Debug tab, click the Start Debug button.



```
int internalFunc(int i)
{
    int t = 0;
    t = i * 2;
    return t;
}

int main(void)
{
    char a = -10;
    short b = internalFunc(a);
    unsigned int c = externalFunc(a);
    short* d = &b;
    int e = c * 22;

    aStruct myA;
    myStruct myS;
    myS.sMember1 = 0xABCD;
    myS.sMember2 = 0xEF;
    myS.sMember3 = &myS;
```

a: *not available*

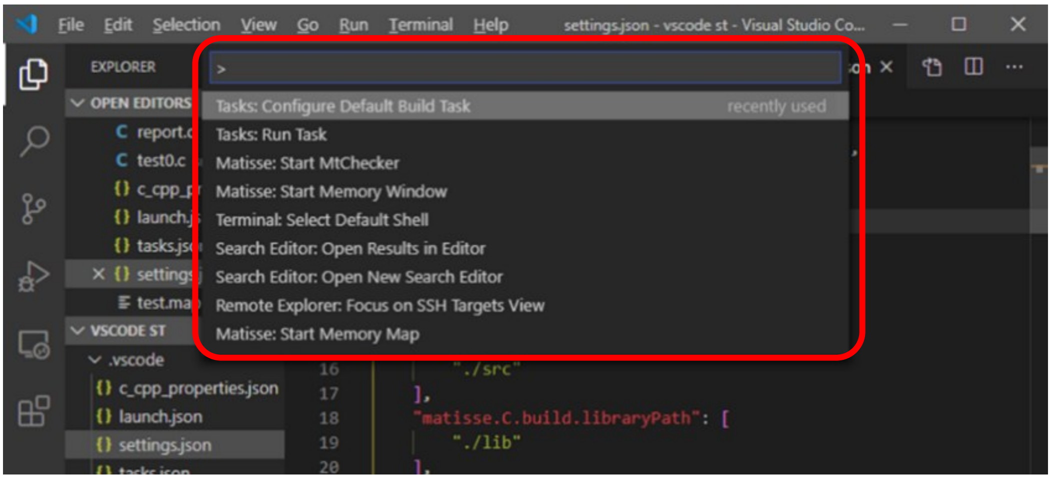
Figure 7. Startup of Peripheral Window through debug execution

Enter debug mode and start the Peripheral window.

5.1.2 In debug mode

While in debug mode, activate the command palette (keystroke: Ctrl + Shift + P or F1), and enter the following command in the command palette.

「Matisse: Start Peripheral Window」


File Edit Selection View Go Run Terminal Help
settings.json - vscode st - Visual Studio Co...
EXPLORER
OPEN EDITORS
report.c
test0.c
c_cpp_
launch.j
tasks.js
settings
test.ma
VSCODE ST
.vscode
c_cpp_properties.json
launch.json
settings.json
tasks.json
>
Tasks: Configure Default Build Task
recently used
Tasks: Run Task
Matisse: Start MtChecker
Matisse: Start Memory Window
Terminal: Select Default Shell
Search Editor: Open Results in Editor
Search Editor: Open New Search Editor
Remote Explorer: Focus on SSH Targets View
Matisse: Start Memory Map

```
        "./src"
    ],
    "matisse.C.build.libraryPath": [
        "./lib"
    ],
```

Figure 8. Entering Startup Commands

5.1.3 Launch from command line (cmd)

To launch the Peripheral Window executable (PeripheralWindow.exe) from cmd, enter the following command line option while in VS Code debug mode.

“PeripheralWindow.exe –port=11111 –configPath=..\..\..\PeripheralWindow\settings.ini”

Table 2. Command-line options

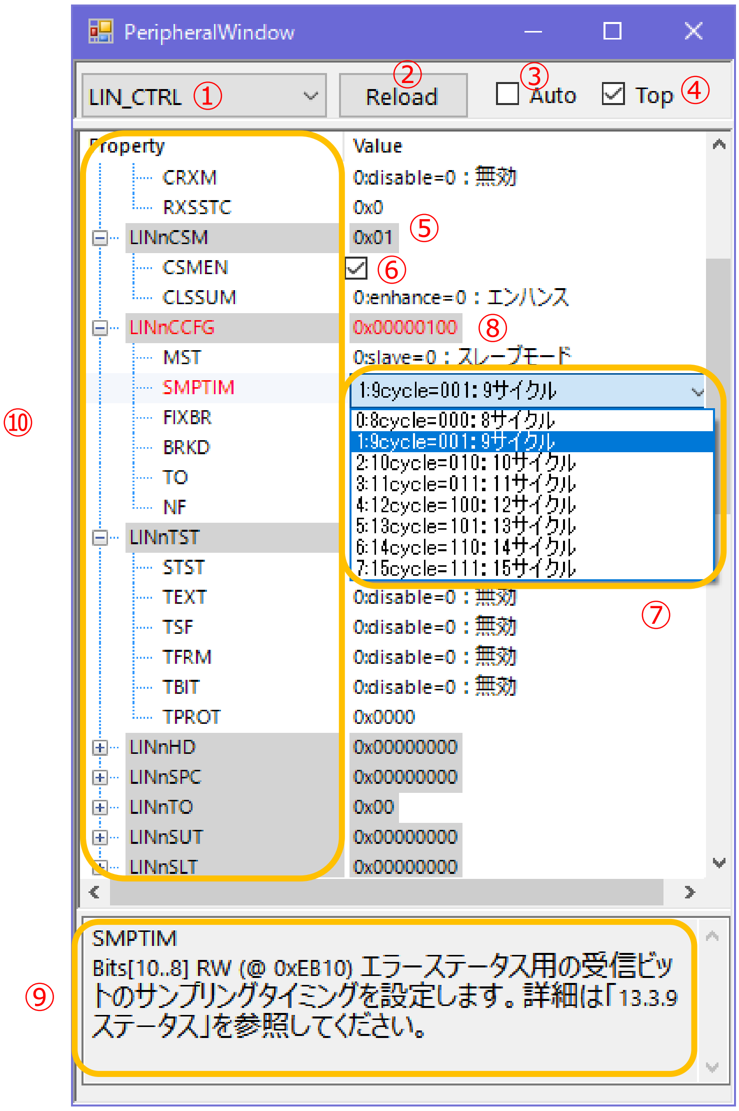
Format	Description
--port=	Enter the port number.
--configPath=	Enter the setting file path

6 User Interface

6.1 Window

6.1.1 Screen layout

When the value of the desired register on the display screen is changed, the corresponding register is written to and the value is read back. If the property is read-only, the display is grayed out.



Property	Value
CRXM	0:disable=0：無効
RXSSTC	0x0
LINnCSM	0x01
CSMEN	☑
CLSSUM	0:enhance=0：エンハンス
LINnCCFG	0x00000100
MST	0:slave=0：スレーブモード
SMPTIM	1:9cycle=001: 9サイクル
FIXBR	
BRKD	
TO	
NF	
LINnTST	
STST	
TEXT	0:disable=0：無効
TSF	0:disable=0：無効
TFRM	0:disable=0：無効
TBIT	0:disable=0：無効
TPROT	0x0000
LINnHD	0x00000000
LINnSPC	0x00000000
LINnTO	0x00
LINnSUT	0x00000000
LINnSLT	0x00000000

0:8cycle=000: 8サイクル
1:9cycle=001: 9サイクル
2:10cycle=010: 10サイクル
3:11cycle=011: 11サイクル
4:12cycle=100: 12サイクル
5:13cycle=101: 13サイクル
6:14cycle=110: 14サイクル
7:15cycle=111: 15サイクル

SMPTIM
Bits[10..8] RW (@ 0xEB10) エラーステータス用の受信ビットのサンプリングタイミングを設定します。詳細は「13.3.9 ステータス」を参照してください。

Figure 9. User Interface

The following sections describe the functions of each UI.

Corresponding keys for keyboard operations are also described.

Keyboard operations can be performed by typing the corresponding key when each UI component has the focus.

①Peripheral Selection List

The list of peripherals defined in the XML file is displayed in the form of a pull-down menu.

The information of the peripheral selected in the list will be displayed on the screen.

（Corresponding key “Alt + ↓”）

②Reload Button

Reads and displays the current register value.

（Corresponding key “Enter”）

③Auto Reload Checkbox

Auto reload can be switched on and off.

When checked, the register value from mtcsim will be reloaded every 1s.

（Corresponding key “Space”）

④Front-most Display Checkbox

Allows you to switch the front-most display of the peripheral window.

When checked, the window will be displayed in the foreground.

（Corresponding key “Space”）

⑤Register Value Setting Textbox

A value within the bit width indicated by the description panel can be written by keyboard input.

In the case of registers of 5 bytes or more, the displayed value is the value of 4 bytes + "...". e.g. 0x... 12345678

（Corresponding key “F2”）

⑥Register Value Setting Checkbox

The bit indicated by the description panel can be switched on and off.

（Corresponding key “Space”）

⑦Register Value Setting List

You can select a field value in the pull-down list to set it.

（Corresponding key “Alt + ↓”）

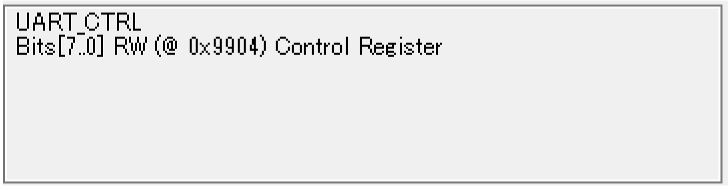
⑧Latest Change Register

The last changed register name and value will be displayed in red.

⑨Description Panel

The register details are displayed in the following format.

[Name BitRange R/W Address Description]



UART_CTRL
Bits[7..0] RW (@ 0x9904) Control Register

Figure 10. Description Panel

The above figure shows the description panel.

The table below shows the displays and their contents.

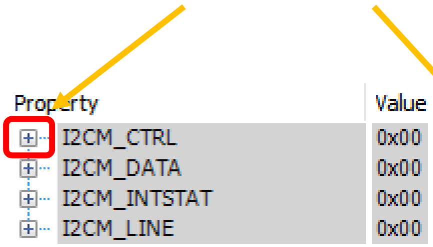
Table 3. Description Panel description

Display	Contents
UART_CTRL	register/field name
[Bits 7..0]	register/field bit width
RW	Register/field can be read/write. Read only is RO, write only is WO.
(@ 0x9904)	Register address value
Control Register	Register/field description

⑩Register Tree

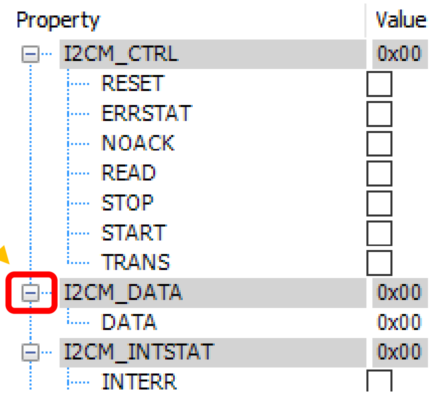
Double-click on the display register name to open the list of field names. Double-click again to close the field name.
（Corresponding key Opening“+”, Closing”-”）

The corresponding keys are "+" when opening and "-" when closing.



Property	Value
⊞ I2CM_CTRL	0x00
⊞ I2CM_DATA	0x00
⊞ I2CM_INTSTAT	0x00
⊞ I2CM_LINE	0x00

Figure 11. List of display register names (Closed)



Property	Value
⊟ I2CM_CTRL	0x00
RESET	□
ERRSTAT	□
NOACK	□
READ	□
STOP	□
START	□
TRANS	□
⊟ I2CM_DATA	0x00
DATA	0x00
⊟ I2CM_INTSTAT	0x00
INTERR	□

Figure 12. List of display register names (open)

6.2 Files used

6.2.1 Setting File

This is a setting file in .ini format that contains the information necessary to start the peripheral window.

The setting file will be loaded at startup.

It consists of the following formats.

Table 4. Configuration File Items and Descriptions

Setting item	Description
[Peripheral] SelectedPeripheral=	It contains the name of the peripheral that will be displayed when the peripheral window is launched.If nothing is specified, the first peripheral of the XML file will be displayed. When the Peripheral Window is closed, the name of the peripheral that was displayed is described.
[FilePath] XMLFilePath=	Set the path of the XML file in which the peripheral information is written in CMSIS-SVD format. **Caution** If this path is incorrect, the Peripheral Window cannot be started.

Ex.)

```
[Peripheral]
SelectedPeripheral=APB_GPIO
[FilePath]
XMLFilePath=”Matisse_svd.xml”
```

6.2.2 XML File

The XML file is a file in which register information of a peripheral generated by RapidMaker is defined in XML format.

If there is an incorrect format, the peripheral window will not launch and an error message will be displayed.

In the case of multi-byte registers, the write order can be specified (higher-⇔lower byte order) by writing keywords.

The setting method is shown below.

①Byte-by-byte write order of multi-byte registers

In the case of multi-byte registers, the order in which they are written can be specified (higher-order or lower-order bytes) by writing the keyword.

The following is an example of how to set this and how it is described in XML.

Write one of the following keywords at the beginning of the string in the <description> of the <register> element.

Table 5. XMLFile keywords and descriptions

Keyword	Description
[WH]	Higher→Lower
[WL]	Lower→Higher

Ex.)

```
<register>
        <name>OUT_PRM1</name>
        <description>[WH]Output Control Parameter1</description>
        …
</register>
```

②Register array display

If a register is array-defined with the specified keywords and XML tags, the register is displayed as an array.

The following two conditions must be met to display as an array.

Condition 1：Keyword "[%s]" at the end of the string in the <name> of the <register> element

Ex.)

```
<register>
        <name> ArrayRegister[%s]</name>
        ・・・
</register>
```

Condition 2 ：Write the tags in the table below in the <register> element

Table 6. Register Tags and Descriptions

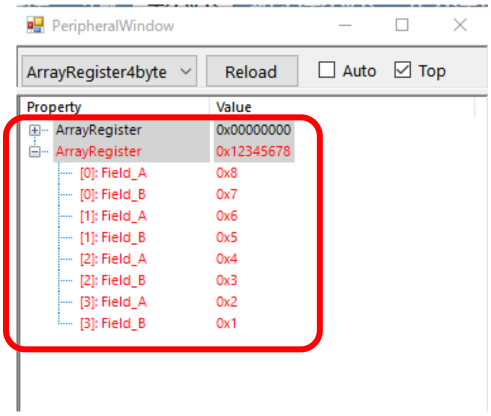
Tag	Description
<dim>	Number of array elements
<dimIndex>	Address increment value between array elements

Ex.)

```
<register>
        <name> ArrayRegister[%s]</name>
        ・・・
        <dim>4</dim>
        <dimIncrement>1</dimIncrement>
        ・・・
</register>
```

In the case of the above XML, the screen display will look like the following.

例) ・32-bit register "ArrayRegister" with 2 fields
- ・Register name: ArrayRegister
- ・Field: Field_A,Field_B
- ・dim：4
- ・dimIncrement：1



Property	Value
⊞ ArrayRegister	0x00000000
⊟ ArrayRegister	0x12345678
[0]: Field_A	0x8
[0]: Field_B	0x7
[1]: Field_A	0x6
[1]: Field_B	0x5
[2]: Field_A	0x4
[2]: Field_B	0x3
[3]: Field_A	0x2
[3]: Field_B	0x1

Figure 13. Array register display

7 Error Message List

A List of error messages for this application is shown below. Please respond according to the content of the message.

Table 7. Error Message List

Event	Error Message	Approach
Startup	The input format is incorrect. Close the application.	Check the setup chapter for the correct format of the port number and setting file path required at startup, and enter them.
	The format of the setting file is incorrect. Load the default value.	The format is incorrectly described in the setting file. Please check the setting file and correct it.
	The peripheral described in the setting file does not exist. Shows the default peripherals.	The peripherals described in the setting file do not exist in the XML file. Please write the name of the peripheral described in the XML file.
	Enter the port number in decimal. Close the application.	Enter the port number as a decimal number.
	The XML file path is incorrect.	Please check the path of the XML file described in the setting file and enter the correct path.
	The read value is not present in the field enumeration list. The XML file and the board's peripherals may not match.	The peripheral information in the XML file and the board peripheral may be different. Please change to the correct XML file or board.
	The format of the XML file is invalid.	The XML file that was loaded has an incorrect format. Please generate the XML file in RapidMaker and load it again.
	The bitRange format of the XML file is invalid.	The bit size information in the XML file may be written incorrectly in <bitRange>. Please make sure that it is written in the format "[○:○]".
	The position of the write order keyword is incorrect. Please put it at the beginning of <description>.	The position of the write order keyword in the XML file is incorrect. Please put it at the beginning of the <description> tag.
When changing register value (keyboard input)	Input value is too large. Please check the bit size.	Please input the register value within the register size.
	The input value is incorrect. Please enter in hexadecimal.	The register value should be entered as a hexadecimal value.
	The read value is not present in the field enumeration list. The XML file and the board's peripherals may not match	The peripheral information in the XML file and the board may be different. Please change to the correct XML file or board.
When communication with mtcsim fails	"Unable to communicate with mtcsim. Quit the application.	This message is displayed when there is poor communication with mtcsim, and the application will close. Please do the following in order: 1. 1. Is VS Code stopped at a breakpoint while in debug mode? 2. Turn on the board again and start up MtProxy. Then check 1.

8 Open-source software licenses

This software includes open-source software (hereinafter referred to as "open-source software program") provided under the following license conditions, in addition to software for which ROHM owns or is licensed.
Open-source software programs are subject to their respective license terms, so in the event of a conflict between the license terms of an open-source software program and this material, the license terms of the open-source software program shall prevail.

Included open-source software and their license terms
・treeviewadv(The BSD License)
・NLog(The BSD License)

9 Trademark notices

"VS Code" is a trademarks of Microsoft Group companies.
“tinyMicon MatisseCORE™” and “matiseye™” are a trademark or registered trademark of ROHM Corporation.

Caution

1. The information written in these materials regarding the software and system (hereinafter collectively "Software") and the contents of the materials are current as of the date of the material's issuance, and may be changed by ROHM, at any time and for any reason, without prior notice.

2. If you plan to use the Software in connection with any equipment or device (such as the medical equipment, transportation equipment, traffic equipment, aerospace equipment, nuclear power control equipment, vehicle equipment including the fuel control system and/or car accessories, and/or various kinds of safety devices etc.) which require extremely high reliability, and whose breakdown or malfunction relate to the risk of personal injury or death, or any other serious damage (such usage is hereinafter called "Special Usage"), you must first consult with the ROHM's sales representative. ROHM is not responsible for any loss, injury, or damage etc. incurred by you or any other third party caused by any Special Usage without ROHM's prior written approval.

3. Semiconductor products may break or malfunction due to various factors. You are responsible for designing, testing, and implementing safety measures in connection with your use of any ROHM products using the Software (such ROHM products are hereinafter called "Product") Such safety measures include, but are not limited to, derating, reductant design, fire spread prevention, backup, and/or fail safe etc. in order to prevent the accident resulting in injury or death and/or fire damage etc.. ROHM is not responsible and hereby disclaims liability for any damage in relation to your use beyond the rated value, or the non-compliance with any precaution for use.

4. ROHM is not responsible for any direct and/or indirect damage to you, or any third parties, (including the damage caused by loss of intangible asset such as information, data, or program etc., loss and/or interruption of profit) which is caused by the use or impossibility to use of the Software.

5. Since the Software, these materials, and/or the Product contain confidential information of ROHM', including technical information, and/or trade secrets, you are prohibited from engaging in any of the following acts in whole or part, without ROHM's prior written approval:
(i) disclosing any ROHM confidential information to a third party;
(ii) disassembling, reverse engineering, and/or any other analysis;
(iii) reprinting, copy, and/or reproduction; or
(iv) removing the copyright notice included in the Software.

6. When exporting the Software, or the technology and/or confidential information written in these materials, you are required to follow the applicable export control laws and regulations such as "Foreign Exchange and Foreign Tarade Act" and/or "Export Administration Regulations (EAR).".

7. ROHM disclaims all warranties, statutory or otherwise, and ROHM hereby disclaims any warranty for non-infringement for the Software and/or the information written in these materials. Accordingly, ROHM is not liable to you for any direct or third-party claims of infringement of rights.

8. No license, whether expressly or implied, is granted hereby under any intellectual property rights or other rights of ROHM or any third parties with respect to the Software or Products or the information contained in these materials.

9. You agree to indemnify, defend and hold harmless ROHM and ROHM's officers and/or employees from responsibility, and hold them harmless, and defend them from any damage, loss, penalty, or cost caused by any claim of liability (including but not limited to the attorney fees) resulting from, or incurred relating to the following acts:
(1) any alleged infringement of a third party's rights or the violation of laws caused by reading, download, encryption, summarization, copy, or transfer etc.; or
(2) violation of these materials.

10. ROHM does not guarantee that these materials or the Software 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 these materials.

ROHM
SEMICONDUCTOR

Thank you for using ROHM products.
For inquiries about our products, please contact us.

ROHM Customer Support System

https://www.rohm.co.jp/contactus