ZMID Communication Board (ZMID-COMBOARD)

Serial Communication and Commands

Contents

Figures

Tables

1. Introduction

This document describes how to setup and use the serial communication capabilities of Renesas' USB Communication and Programming Board (ZMID-COMBOARD) for ZMID Application Modules in order to provide an interface between the user's computer and the Renesas ZMID4200 that is the device-under-test (DUT) on the module.

1.1 Requirements for User's Computer

- Windows® XP, Vista SP1 or later, 7 (including SP1), 8, 8.1, or 10
- Available USB port

1.2 Driver Installation

The driver required for serial communication is automatically installed on operating systems newer than Windows® 8. For older operating systems, the driver must be manually installed.

To manually install the driver, follow these steps:

1. Connect the ZMID-COMBOARD to an available USB port on the user's computer via the micro-USB cable.
2. The board will appear as two removable storage devices named EVKIT-1 and EVKIT-2. Open either one of them.
3. Locate the *LPC-VCOM.INF* file and open the menu by clicking with the right mouse button over it.

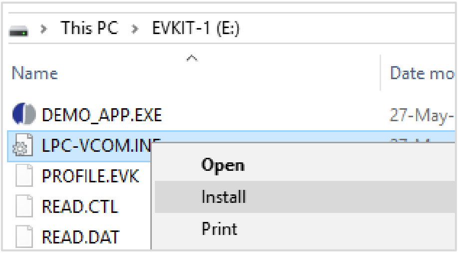

Figure 1. Location of the Driver Installer

4. Select the "Install" option and complete the install setup.
5. After successful driver installation, the device appears in the Device Manager under "Ports" as a "USB Serial Device." See Figure 2.

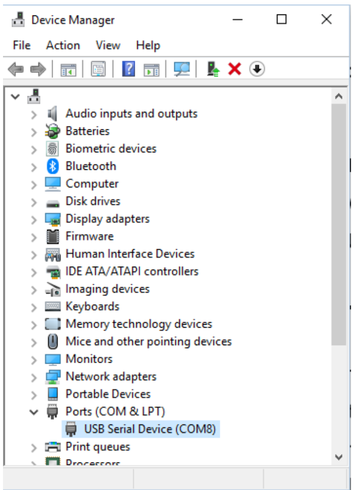

Figure 2. The ZMID-COMBOARD Appears as a Serial Device

1.3 Communication Basics

The computer communicates with the ZMID-COMBOARD through a virtual COM port (VCOM). The commands and responses can be interpreted as ASCII characters. The computer is the “master” in the communication – it sends a command and the ZMID-COMBOARD always returns a response.

Default COM port settings:

- Port Number: Check the Device Manager; the port number is assigned by the operating system and can vary
- Baud Rate: 19200
- Data Bits: 8
- Stop Bits: 1
- Parity Bits: No
- Flow Control: No

Format of the commands and responses:

Both commands and responses end with a carriage return and a line feed character: “\r\n” which corresponds to the ASCII bytes $0D_{HEX}$ and $0A_{HEX}$.

The first byte of the response is a status byte which can be either an Acknowledge (06_{HEX} in ASCII) or Not Acknowledge (15_{HEX} in ASCII).These responses are represented as <ACK> and <NACK> in this document. Depending on the command, a response can have only a status byte or it can be followed by a number of data bytes.

The commands and response are case-insensitive.

Errors are returned as responses that start with a Not Acknowledge byte and can have optional error code bytes.

1.3.1. Examples

Command: OR_E2

Description: OWI Read with command address $E2_{HEX}$

Characters	O	R	_	E	2	\r (Carriage Return)	\n (Line Feed)
Bytes	72_{HEX}	52_{HEX}	$5F_{HEX}$	45_{HEX}	32_{HEX}	$0D_{HEX}$	$0A_{HEX}$

Reply: <ACK>1C3F

Description: Acknowledge byte and data: $1C3F_{HEX}$

Characters	<ACK>	1	C	3	F	\r (Carriage Return)	\n (Line Feed)
Bytes	06_{HEX}	31_{HEX}	43_{HEX}	33_{HEX}	46_{HEX}	$0D_{HEX}$	$0A_{HEX}$

Command: T00000

Description: Turn off the power for the DUT

Characters	T	0	0	0	0	0	\r (Carriage Return)	\n (Line Feed)
Bytes	54_{HEX}	30_{HEX}	30_{HEX}	30_{HEX}	30_{HEX}	30_{HEX}	$0D_{HEX}$	$0A_{HEX}$

Reply: <ACK>

Description: Acknowledge without extra data bytes

Characters	<ACK>	\r (Carriage Return)	\n (Line Feed)
Bytes	06_{HEX}	$0D_{HEX}$	$0A_{HEX}$

2. Commands

Table 1. Commands List

Command	Action
General Commands	
V	Returns the firmware version information (see Table 2 for details)
V_HW	Returns the hardware revision information (see Table 3 for details)
V_FW	Returns the supported interfaces information (see Table 4 for details)
MS	Selects the active module (device) for communication and output reading (see Table 5 for details)
Power and Trigger Commands	
T	Device under test (DUT) power control and power-on delay trigger setup (see Table 6 for details)
T_	Power-off delay trigger setup (see Table 7 for details)
Communication Commands	
OWT	OWI WRITE with trigger (see Table 8 for details)
OW_	OWI WRITE (see Table 9 for details)
OR_	OWI READ (see Table 10 for details)
ORS	OWI READ continuous (see Table 11 for details)
ORSX	OWI stop continuous READ (see Table 12 for details)
Commands for Reading the Output	
TSO	Output interpretation setup (see Table 13 for details)
MRO	Read last measured output (see Table 14 for details)
MRS	Read last SENT frame (see Table 15 for details)
Pin State Commands	
PS_	Set pin state (see Table 16 for details)

2.1 General Commands

Table 2. Version Command: V

Command	V	
Description	Returns a string with the firmware version of the ZMID-COMBOARD.	
Syntax	V	
Example	Send	V
	Response	<ACK>ZMID COM BOARD FW_00.05.1309

Table 3. Hardware Revision Command: V_HW

Command	V_HW	
Description	Returns a string with the recognized main hardware revision of the ZMID-COMBOARD.	
Syntax	V_HW	
Example	Send	V_HW
	Response	<ACK>R5.1

Table 4. Supported Interfaces Command: V_FW

Command	V_FW	
Description	Returns a string with the supported interfaces of the ZMID-COMBOARD.	
Syntax	V_FW	
Example	Send	V_FW
	Response	<ACK>FW Interfaces: ANALOG, OWI, SENT, PWM

Table 5. Module Select Command: MS

Command	MS	
Description	Selects the active module (device) between 1 and 2. Further OWI communication or output reading will be performed with the module selected.	
Syntax	MSx x – module/device: x = 0 = Module 1 (Device 1) x = 1 = Module 2 (Device 2)	
Examples	Send	MS0 – Select Device 1 as active
	Response	<ACK>
	Send	MS1 – Select Device 2 as active
	Response	<ACK>

2.2 Power and Trigger Commands

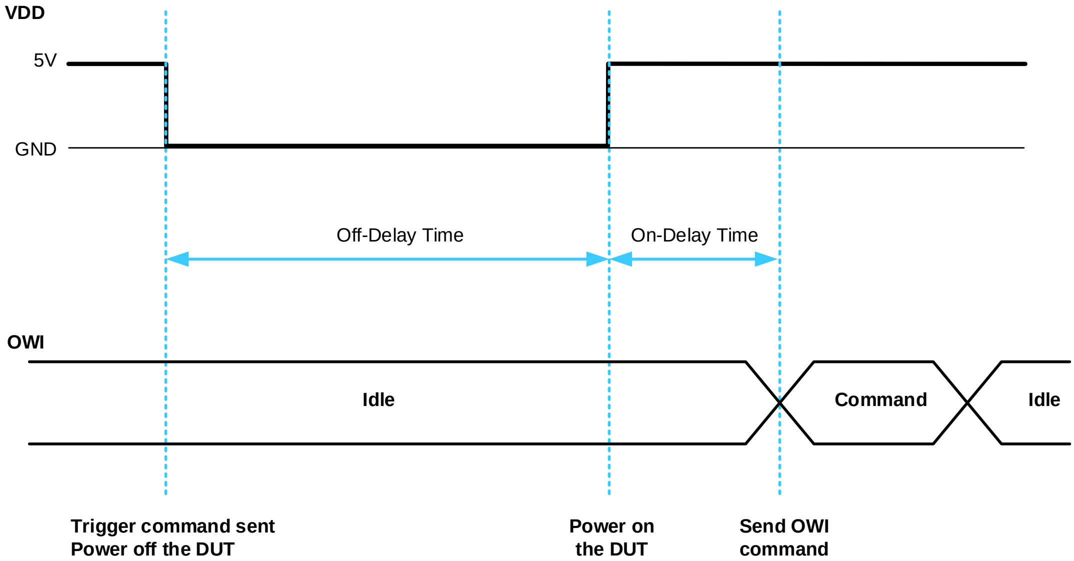

Figure 3. Trigger Command Timing Diagram

Table 6. Power and On-Delay Trigger Setup Command: T

Command	T	
Description	Changes the DUT's VDD state and sets the on-delay used when executing a trigger command.	
Syntax	Txxttt xx = ZMID VDD state (binary) ▪ xx = 00_{BIN} = Off ▪ xx = 11_{BIN} = On ▪ xx = 01_{BIN} / 10_{BIN} = Forbidden ttt = On-delay time in milliseconds (decimal) from 000 to 999	
Examples	Send	T00000 = Turn off the VDD for the DUT
	Response	<ACK>
	Send	T11001 = Turn on the ZMID VDD and set the on-delay trigger time to 1 millisecond
	Response	<ACK>
	Send	T11020 = Turn on the ZMID VDD and set the on-delay trigger time to 20 milliseconds
	Response	<ACK>

Table 7. Off-Delay Trigger Setup Command: T_

Command	T_	
Description	Defines the off-delay time for the ZMID VDD when executing a trigger command.	
Syntax	T_ttt ttt = Off-delay time in milliseconds (decimal) from 000 to 999	
Example	Send	T_100 = Set the off-delay trigger time to 100 milliseconds
	Response	<ACK>

2.3 Communication Commands

Table 8. OWI WRITE with Trigger Command: OWT

Command	OWT	
Description	Performs a triggered OWI WRITE command with a command byte and optional data bytes.	
Syntax	OWTccdddd cc = command byte (hex string) dddd = data bytes (hex string) - optional	
Example	Send	OWT81FFFF = Trigger command, write 81_{HEX} as command byte and $FFFF_{HEX}$ as data bytes
	Response	<ACK>

Table 9. OWI WRITE Command: OW_

Command	OW_	
Description	Performs an OWI WRITE command with a command byte and optional data bytes. If the data bytes are more than 2, a bulk WRITE is performed where the command byte is incremented before writing the next two data bytes. Writing xxxx instead of a hex value in the bulk WRITE operation causes the current command byte to be skipped.	
Syntax	OW_ccdddd OW_ccdddddddd…. = Bulk WRITE cc = command byte (hex string) dddd = data bytes (hex string)	
Examples	Send	OW_A0FFFF = WRITE command byte $A0_{HEX}$ and 2 byte data $FFFF_{HEX}$
	Response	<ACK>
	Send	OW_A1BEEFCAFExxxxFFFF = bulk WRITE – start command byte is $A1_{HEX}$, will skip command byte $A3_{HEX}$
	Response	<ACK>

Table 10. OWI READ Command: OR_

Command	OR_	
Description	Performs an OWI READ command with a command byte. If a number is specified after the command byte a bulk READ is performed where the command byte is incremented for each READ operation.	
Syntax	OR_cc OR_ccnnn cc = command byte (hex string) nnn = optional: number of bulk READs to perform (decimal) from 000 to 015	
Examples	Send	OR_05 = command byte is 05_{HEX}; reads one register (2 bytes)
	Response	<ACK>0004 = 2 byte reply from the READ operation
	Send	OR_E2004 = bulk READ; command byte is $E2_{HEX}$; reads 4 registers (8 bytes) by incrementing the command byte; Equivalent to sending OR_E2; OR_E3; OR_E4; OR_E5
	Response	<ACK>BEEFCAFE3333FFFF = $BEEF_{HEX}$, $CAFE_{HEX}$, 3333_{HEX}, $FFFF_{HEX}$ returned from the bulk READ.

Table 11. OWI READ Continuous Command: ORS

<table>
<tr><td>Command</td><td colspan="2">ORS</td></tr>
<tr><td>Description</td><td colspan="2">Starts a continuous reading of a specified command byte. Does 5000 reads if not stopped. The reading cycle includes sending specific commands for stopping and starting the position processing of the ZMID4200. The command returns one normal reply with <ACK> and then continuously sends 2 byte readings. During the reading, the computer (master) must continuously poll its receive buffer for new data from the DUT.</td></tr>
<tr><td>Syntax</td><td colspan="2">ORScc
cc = command byte (hex string)</td></tr>
<tr><td rowspan="2">Example</td><td>Send</td><td>ORSD8 = starts a continuous READ of the D8 command byte; reads the spatial angle (Spa) register in the SWR memory of the DUT</td></tr>
<tr><td>Response</td><td><ACK>\r\n = acknowledge reply for the command
13F2\r\n = 2 byte reading
15B3\r\n = 2 byte reading
188C\r\n = 2 byte reading
…
188C\r\n = 2 byte reading
runs until 5000 readings are performed or until a STOP command is sent by the computer</td></tr>
</table>

Table 12. OWI READ Special STOP Command: ORSX

<table>
<tr><td>Command</td><td colspan="2">ORSX</td></tr>
<tr><td>Description</td><td colspan="2">Stops the continuous reading started by the ORS command.
Recommendation: The computer (master) should wait approximately 500 milliseconds and then clear its receive buffer before sending another command. There is a small delay between receiving the command in the firmware and stopping the continuous reading</td></tr>
<tr><td>Syntax</td><td colspan="2">ORSX</td></tr>
<tr><td rowspan="2">Example</td><td>Send</td><td>ORSX</td></tr>
<tr><td>Response</td><td><ACK></td></tr>
</table>

2.4 Commands for Reading the Output

The ZMID-COMBOARD supports the reading and interpretation of the analog, PWM, or SENT output depending on the DUT product version. Before reading the output, the ZMID-COMBOARD must be instructed on how to interpret the output from the DUT.

Table 13. Set Output Interpretation Command: TSO

<table>
<tr><td>Command</td><td colspan="2">TSO</td></tr>
<tr><td>Description</td><td colspan="2">Sets the output interpretation of the DUT’s signal to analog, PWM, or SENT.</td></tr>
<tr><td>Syntax</td><td colspan="2">TSOxxxx
xxxx = 5201 = interpret output as analog
xxxx = 5202 = interpret output as PWM
xxxx = 5203 = interpret output as SENT
Any other combination is forbidden.</td></tr>
<tr><td rowspan="2">Example</td><td>Send</td><td>TSO5202 = instructs the firmware to interpret the output of the DUT as a PWM signal</td></tr>
<tr><td>Response</td><td><ACK></td></tr>
</table>

Table 14. Read Output Command: MRO

<table>
<tr><td>Command</td><td>MRO</td></tr>
<tr><td>Description</td><td>Reads a sample from the interpreted DUT‘s output; returns a 4-byte reply from which the 12 LSBs are the output data.
For analog: 0_{DEC} = 0% VDD; 4095_{DEC} = 100% VDD
For PWM: 0_{DEC} = 0% duty cycle; 4095_{DEC} = 100% duty Cycle</td></tr>
</table>

	For SENT: the FC1 (Fast Channel 1) data is directly mapped to the 12 LSBs of the output data	
Syntax	MRO	
Example	Send	MRO
	Response	<ACK>0FFF$_{HEX}$ = extracting the 12 LSBs results in an output reading of FFF$_{HEX}$ = 4095$_{DEC}$.

Table 15. Read Last SENT Frame Command: MRS

Command	MRS	
Description	Reads the decoded contents of the last received SENT frame. Returns a 4-byte reply with the following encoding: SCAAABBB (hex string) S – 4-bit status data C – 4-bit CRC data AAA – 12-bit FC1 data BBB – 12-bit FC2 data	
Syntax	MRS	
Example	Send	MRS
	Response	<ACK>06D8DC62 Status: 0$_{HEX}$ CRC: 6$_{HEX}$ FC1: D8D$_{HEX}$ FC2: C62$_{HEX}$

2.5 Pin State Commands

Table 16. Pin State Command: PS_

Command	PS_	
Description	Sets the state of a controllable pin of the header on the ZMID-COMBOARD to operate external components such as output signal multiplexors or additional pull-up resistors. The pins can be set to a HIGH, LOW, or high impedance state. Newer versions of the ZMID-COMBOARD (R5_1 and above) have no pin header; instead two signal multiplexors and an additional pull-up resistor are mounted on the board and connected to the following pins: Pin 3 – Stronger pull-up resistors for Device 1 and 2 – used for OWI or PWM LOW or high impedance = pull-up inactive HIGH = pull-up active Pin 4 – Multiplexor for the output of Device 2 LOW = used to read analog output HIGH = used for OWI, PWM, and SENT High Impedance = not defined Pin 5 – Multiplexor for the output of Device 1 LOW = used to read analog output HIGH = used for OWI, PWM, and SENT High Impedance = not defined **Important:** Do not change the state of pins 1, 6, or 8.	
Syntax	PS_ppx pp = pin number (decimal) from 01 to 08 x = pin state x = 0 = LOW x = 1 = HIGH x = 2 = Tri-state (high impedance)	
Examples	Send	PS_031 = Enable the pull-up for PWM and OWI communication
	Response	<ACK>
	Send	PS_041 = Set the output multiplexor for Device 2 for digital interfaces (OWI/SENT/PWM)

	Response	<ACK>
	Send	PS_050 = Set the output multiplexor for Device 1 for analog interface
	Response	<ACK>

3. Examples

Table 17. Connecting and Reading EEPROM and SWR Memory (Device 1)

Command	Comment
MS0	Select Device 1 as active for communication and output reading
<ACK>	
T_100	Power-off delay trigger setup = 100ms
<ACK>	
T11001	Power-on delay trigger setup = 1ms; power on the device
<ACK>	
PS_051	Device 1 output multiplexor set for digital communication
<ACK>	
PS_031	Enable additional pull-up for OWI communication
<ACK>	
OWT0283AE	OWI WRITE with trigger – enter Command Mode
<ACK>	
OR_05	OWI READ – read the status register of the device
<ACK>0004	Status register reply – device is in Command Mode
Memory Read: EEPROM	
<ACK>	
OR_E0015	OWI bulk READ – read 15 registers starting from command byte $E0_{HEX}$
<ACK>23C8048D00000600120A9D87888E008054BF010858 03B107083B0255BFFF	OWI bulk READ reply – 15 registers (30 bytes)
OR_EF003	OWI bulk READ – read 3 registers starting from command byte EF_{HEX}
<ACK>0000000000C2	OWI bulk READ reply – 3 registers (6 bytes)
Memory Read: SWR	
OW_04	OWI WRITE – HOLD_DPU command to stop the position calculation while reading data
<ACK>	
OR_C0015	OWI bulk READ – 15 registers starting from command byte $C0_{HEX}$
<ACK>23C8048D00000600120A9D87888E008054BF010858 03B107083B0255BFFF	OWI bulk READ reply – 15 registers (30 bytes)
OR_D1	OWI READ – command byte $D1_{HEX}$
<ACK>00C2	OWI READ reply – 1 register (2 bytes)
OR_D3009	OWI bulk READ – 9 registers starting from command byte $D3_{HEX}$
<ACK>03B901E600017FF30321400640E042270001	OWI bulk READ reply – 9 registers (18 bytes)
OW_03	OWI WRITE – RUN_DPU command to start the position calculation
<ACK>	
T00000	Power off the device

Command	Comment
<ACK>	

Table 18. Writing to the First 7 Registers in EEPROM (Device 1)

Command	Comment
OW_A023C8	Write to EEPROM register 00_{HEX} (*Offset*); command byte = $A0_{HEX}$
<ACK>	
OW_A1048D	Write to EEPROM register 01_{HEX} (*Slope*); command byte = $A1_{HEX}$
<ACK>	
OW_A20000	Write to EEPROM register 02_{HEX} (clamping limits); command byte = $A2_{HEX}$
<ACK>	
OW_A30600	Write to EEPROM register 03_{HEX} (linear interpolation points 0 and 1); command byte = $A3_{HEX}$
<ACK>	
OW_A4120A	Write to EEPROM register 04_{HEX} (linear interpolation points 2 and 3); command byte = $A4_{HEX}$
<ACK>	
OW_A59D87	Write to EEPROM register 05_{HEX} (linear interpolation points 4 and 5); command byte = $A5_{HEX}$
<ACK>	
OW_A6888E	Write to EEPROM register 06_{HEX} (linear interpolation points 6 and 7); command byte = $A6_{HEX}$
<ACK>	
OW_A70080	Write to EEPROM register 07_{HEX} (linear interpolation point 8); command byte = $A7_{HEX}$
<ACK>	

Table 19. Bulk Writing to the First 7 Registers in EEPROM (Device 1)

Command	Comment
OW_A023C8048D00000600412A9D87888E0080	Bulk WRITE to EEPROM registers 00_{HEX} to 07_{HEX} (command byte $A0_{HEX}$ to $A7_{HEX}$)
<ACK>	

Table 20. Reading 3 Analog Output Samples from Device 1

Command	Comment
T_100	Power-off delay trigger setup = 100ms
<ACK>	
TSO5201	Set the output recognition to analog
<ACK>	
PS_050	Device 1 output multiplexor set for analog signal
<ACK>	
PS_032	Disable the additional pull-up
<ACK>	
T11001	Power-on delay trigger setup = 1ms; power on the device
<ACK>	
MRO	Read an output sample
<ACK>00000424	Read reply = 424_{HEX} (1060_{DEC}); 1060 / 4095 $*$ 100 = 25.89% VDD

Command	Comment
MRO	Read an output sample
<ACK>00000424	Read reply
MRO	Read an output sample
<ACK>00000424	Read reply
T00000	Power off the DUT
<ACK>	

Table 21. Reading SENT Frames from Device 1

Command	Comment
T_100	Power-off delay trigger setup = 100ms
<ACK>	
TSO5203	Set the output recognition to SENT
<ACK>	
PS_051	Device 1 output multiplexor set for digital signal
<ACK>	
PS_032	Disable the additional pull-up
<ACK>	
T11001	Power-on delay trigger setup = 1ms; power on the device
<ACK>	
MRS	Read last SENT frame
<ACK>05C81B43	Read reply = status: 0_{HEX}; CRC: 5_{HEX}; FC1: $C81_{HEX}$; FC2: $B43_{HEX}$
MRS	
<ACK>08C81733	Read reply = status: 0_{HEX}; CRC: 8_{HEX}; FC1: $C81_{HEX}$; FC2: 733_{HEX}
MRS	
<ACK>0BC812F3	Read reply = status: 0_{HEX}; CRC: B_{HEX}; FC1: $C81_{HEX}$; FC2: $2F3_{HEX}$
T00000	Power off the DUT
<ACK>	

Table 22. Enter Command Mode on Device 2

Command	Comment
MS1	Select Device 2 as active for communication and output reading
<ACK>	
T_100	Power-off delay trigger setup = 100ms
<ACK>	
T11001	Power-on delay trigger setup = 1ms; power on the device
<ACK>	
PS_041	Device 2 output multiplexor set for digital communication
<ACK>	
PS_031	Enable additional pull-up for OWI communication
<ACK>	
OWT0283AE	OWI WRITE with trigger: Enter Command Mode
<ACK>	
OR_05	OWI READ: Read the status register of the device

Command	Comment
<ACK>0004	Status register reply: Device is in Command Mode

Table 23. Reading PWM Output from Device 1 and Device 2

Command	Comment
T_100	Power-off delay trigger setup = 100ms
<ACK>	
TSO5202	Set the output recognition to PWM
<ACK>	
PS_041	Device 2 output multiplexor set for digital signal
<ACK>	
PS_051	Device 1 output multiplexor set for digital signal
<ACK>	
PS_031	Enable the additional pull-up
<ACK>	
T11001	Power-on delay trigger setup = 1ms; power on the device
<ACK>	
MS0	Select Device 1 as active for communication and output reading
<ACK>	
MRO	Read last output sample
<ACK>00000FD0	
MS1	Select Device 2 as active for communication and output reading
<ACK>	
MRO	Read last output sample
<ACK>00000224	
MS0	Select Device 1 as active for communication and output reading
<ACK>	
MRO	Read last output sample
<ACK>000007BC	
MS1	Select Device 2 as active for communication and output reading
<ACK>	
MRO	Read last output sample
<ACK>00000C84	

4. Glossary

Abbreviation	Meaning
DUT	Device Under Test
VCOM Port	Virtual Communication Port
ASCII	American Standard Code for Information Interchange – character encoding standard
PWM	Pulse Width Modulation
SENT	Single Edge Nibble Transmission
EEPROM	Electrically Erasable Programmable Read-Only Memory
SWR	Shadow Registers – Working memory of the ZMID4200
DPU	Digital Processing Unit
CRC	Cyclic Redundancy Check
LSB	Least Significant Bit
FC1	Fast Channel 1 of the SENT transmission data
FC2	Fast Channel 2 of the SENT transmission data

5. Revision History

Revision	Date	Description
1.0	Jul 21, 2021	Initial release.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/