ZMID520x

One Wire Interface (OWI)

Contents

List of Figures

List of Tables

1. Introduction

This document describes the One Wire Interface (OWI) implemented in the ZMID520x family of products, including the ZMID5201, ZMID5201, and ZMID5203. Unless otherwise noted, the content of this document applies to all ZMID520x products.

2. One Wire Interface

The OWI is used for programming purposes. It is a bidirectional, serial point-to-point interface, which provides access to all calibration, configuration, and identification EEPROM words and also to user, test, and evaluation registers. ZMID520x functions can also be triggered via the OWI; e.g., the start/stop for position calculation or enabling certain test modes.

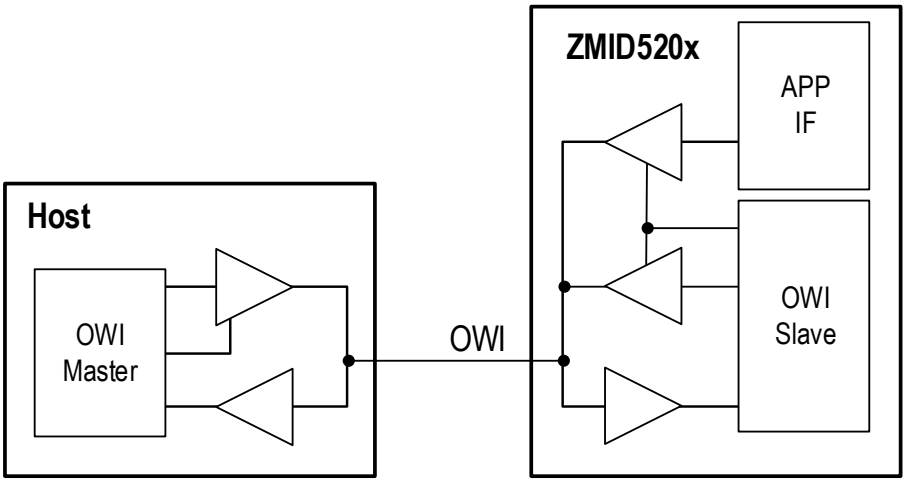
2.1 OWI Overview

The ZMID520x OWI has the following main properties:

- Single signal wire connection
- Push-pull output driver
- Point-to-point connection; single-master, single-slave system configuration
- Bidirectional serial communication with most significant bit (MSB) first
- Pulse-width encoding of digital symbols (0, 1)
- Self-synchronizing (no clock transfer from master to slave)
- Flexible bit rate; automatic adaption to the master at the slave side
- 16-bit-word oriented data transfer
- Even parity bit for each transferred byte in both directions
- START, STOP, and handover conditions for data flow control
- Timeout to recover protocol errors
- Glitch suppression circuit on input path

Figure 1 shows the system connection with the OWI. The host system, e.g. a laboratory computer, contains an OWI master, and the ZMID520X has an OWI slave. Both are connected by bidirectional bus drivers to the common OWI wire. On theZMID520x side, the OWI input is a CMOS Schmitt trigger and the output is a CMOS buffer.

Figure 1. OWI Overview



2.2 OWI General Data Transfer

The general OWI protocol uses the following sequence:

- The ZMID520x is always the slave.
- The master sends the START condition.
- The master sends a command byte (MSB first). The next parameter is determined by the operation:
  - For a WRITE access, the master sends a word as the parameter (MSB first).
  - For a READ access, the slave returns a word (MSB first).
  - For an action, the master continues with the next step (no parameter required).
- The master sends the STOP condition.
- The master enters the idle condition.

Figure 2. OWI General Data Transfer

START	Command	Data Word 1	Data Word 0	STOP	Idle

The command determines whether the access is a READ or WRITE access or an action:

- WRITE: The master sends the write data followed by a STOP condition.
- READ: The master disables its output after a 3/4 bit period (handover). The slave waits for the falling edge on the bus and immediately drives a 0 (takeover). If the takeover bit time is finished, the slave transfers the requested word. The slave sends a 0 at the end but must disable its output after a 3/4 bit period (handover). The master waits for the falling edge on the bus and immediately drives a 0 (takeover). If the takeover bit time is finished, the master finishes the read transfer with a STOP condition.
- Action: The master immediately sends the stop condition since an action trigger does not need a parameter.
- Timeout: If no rising edge is found for a time longer than the maximum t_{owi_bit} specification ∗ 2 (see Table 3), the slave goes into the idle state and waits for a rising edge as a START condition. This can be used to set the interface to a defined state.

2.3 OWI Protocol Details

2.3.1 Bit Transfer

As no clock is transferred between master and slave, the protocol must be self-synchronizing. This is realized by a pulse-width modulation of the bits. Each bit time is split into four segments (seg1, seg2, seg3, seg4). During the first segment, the OWI signal is unconditionally driven high; during the last segment, it is unconditionally driven low. The two intermediate segments are set to the respective data bit; i.e., for a 0 data bit, they are both driven low and for a 1 data bit, they are both driven high. Therefore, any bit contains a rising and a falling edge. The rising edges are equally spaced and are used by the slave to determine the bit period and consequently as clock regeneration. The falling edges carry the actual data information. Figure 3 shows the encoding of a single 0 or 1 data bit (note: the timing is not to scale).

Figure 3. OWI Bit Representation

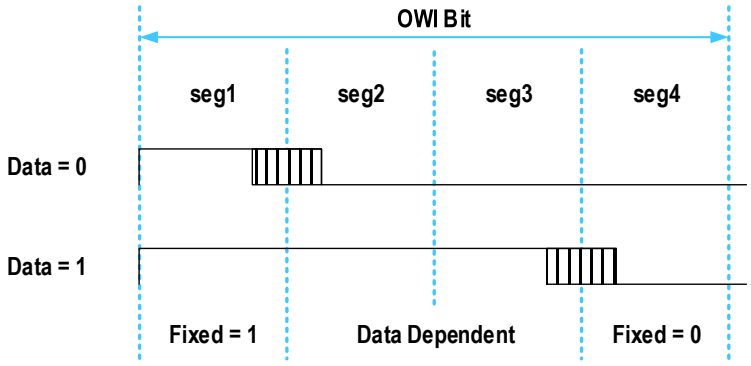
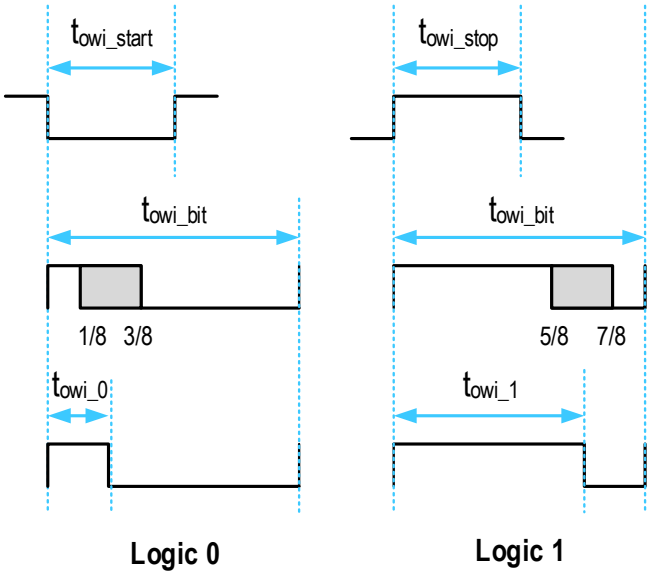


Figure 4 shows the timing parameters of the OWI protocol. The values are specified in section 3.

Figure 4. OWI Bit Timing Parameters



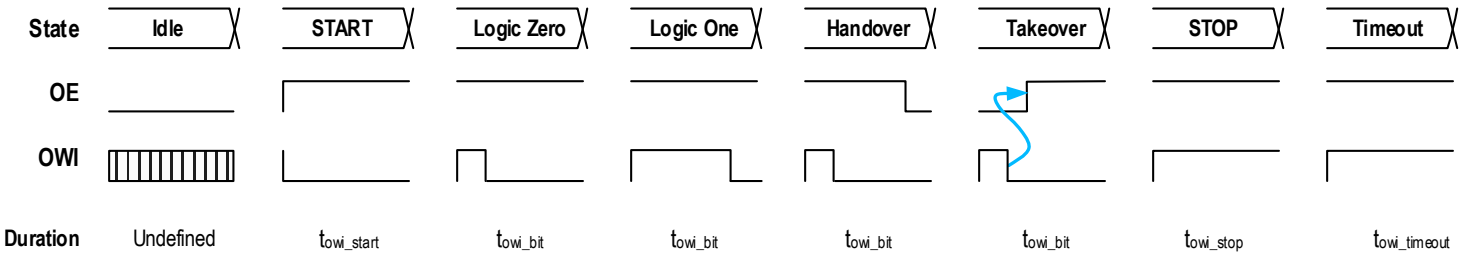
2.3.2 ZMID520x Protocol Element

Apart from the bit representations, the OWI protocol requires several other protocol elements primarily for the correct handshaking between master and slave. These include the following:

- Idle: Master and slave are both in idle state, no transaction is ongoing
- START: Start condition sent by the master to setup a new transaction
- Handover: The driving side disables its output and releases the OWI to the other side
- Takeover: The listening side enables its output and takes over the OWI to become the driving side
- STOP: The STOP condition is sent by the master to finalize the ongoing transaction
- Timeout: No rising master edge during an ongoing transaction for the duration of $t_{owi\_timeout}$ (see section 3).

Figure 5 illustrates these ZMID520x OWI elements with the associated output enable (OE) and OWI signals. Note that Figure 5 is not to scale. The actual timings are specified in section 3.

Figure 5. OWI ZMID520x Protocol Elements

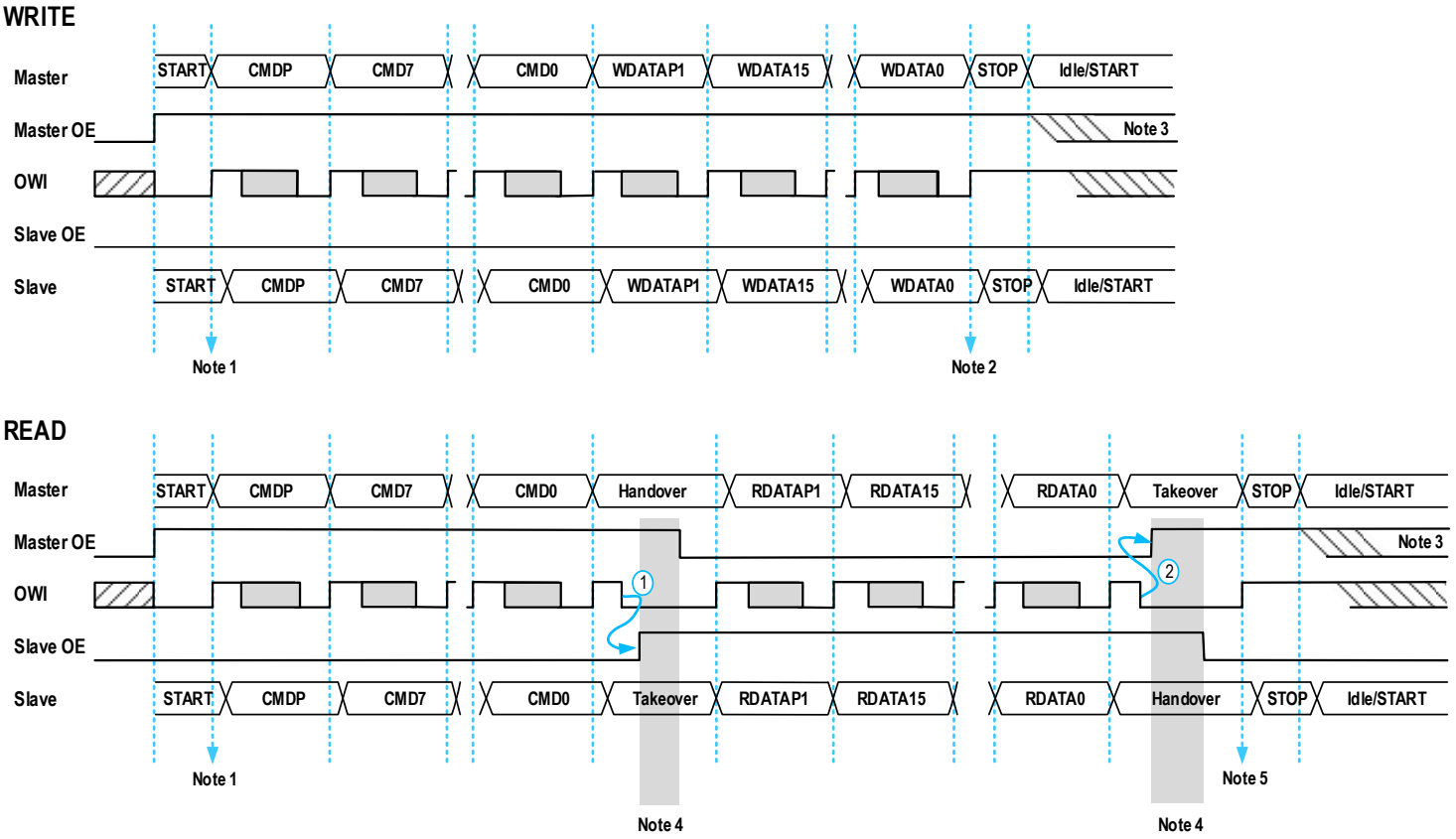


2.3.3 Standard Protocol Flow

Figure 6 shows the timing for full WRITE and READ access. Each command consists of an associated parity bit followed by 8 command bits (CMDP, CMD7 to CMD0). Each data word consists of the parity bit associated with the 8 most significant bits (MSBs) followed by the 8 MSBs and then the parity bit associated with the 8 least significant bits (LSBs) followed by the 8 LSBs (DATAP1, DATA15 to DATA8, DATAP0, DATA7 to DATA0). The parity bit is always transferred first followed by the data starting with the MSB.

Figure 6 illustrates this protocol using the prefix R or W to indicate the direction of the parity and data bits.

Figure 6. OWI Detailed Data Transfer



Note 1: A rising edge is the START condition for the slave to receive command bits.

Note 2: A missing rising edge generates a timeout and the command is ignored.

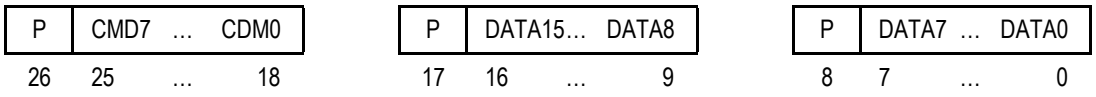
Note 3: The master might not drive the bus. In this case, the bus is defined by an external pull resistor.

Note 4: There is an overlap of the output enables of the master and slave. Both drive a LOW level. This ensures the independence of the existence of external pull resistors. In addition it improves the EMC robustness because all levels are actively driven.

Note 5: If the master does not take the bus and a pull-up exists, the STOP condition is generated by the latter. With a pull-down, a timeout is generated and the slave waits for a rising edge as the START condition.

Figure 7 shows the bit positions within the serial data stream. Each byte begins with its own parity bit.

Figure 7. OWI Command and Data Format

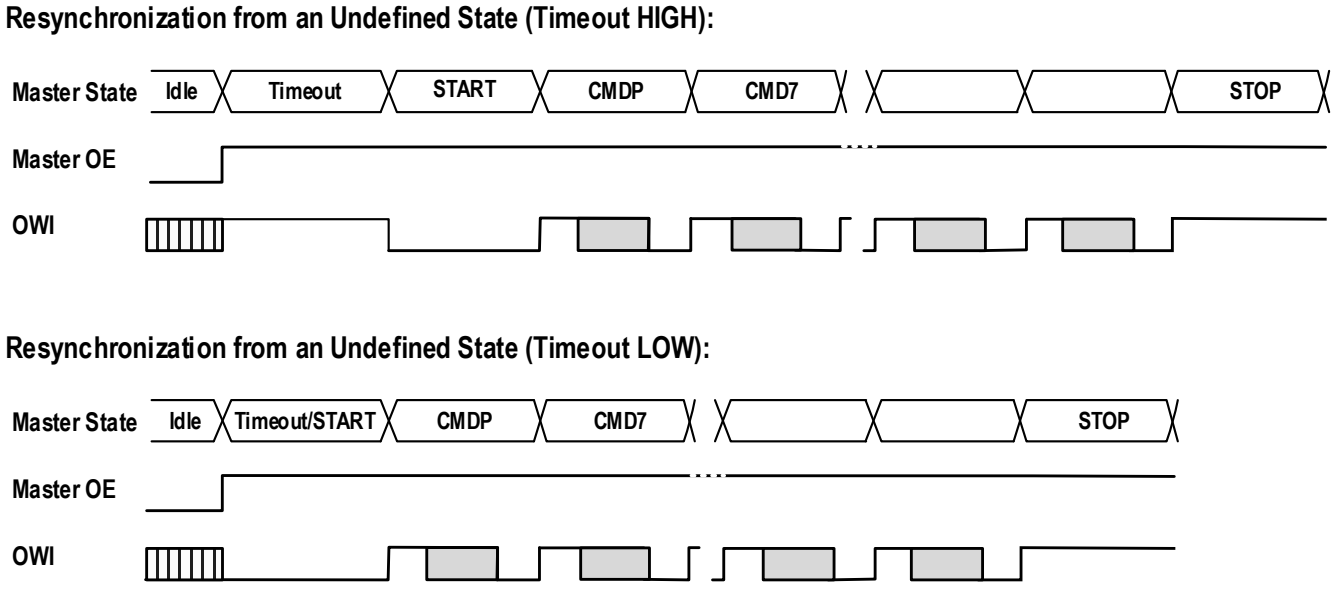


2.3.4 Protocol Error Handling

Typically, access is terminated when the master sends a STOP condition, which sets the slave to the idle state. In a disturbed environment, it is possible that the data bits and/or other protocol elements are not detected correctly. This could lead to a loss of synchronism between the master and the slave, in which case the slave will not able to follow the master transactions.

This type of out-of-sync condition can be resolved by a timeout initiated by the master. The timeout resets the slave to the idle state regardless of preceding events. Therefore the next transaction starts from a clean state.

Figure 8. OWI Re-synchronization from an Undefined State

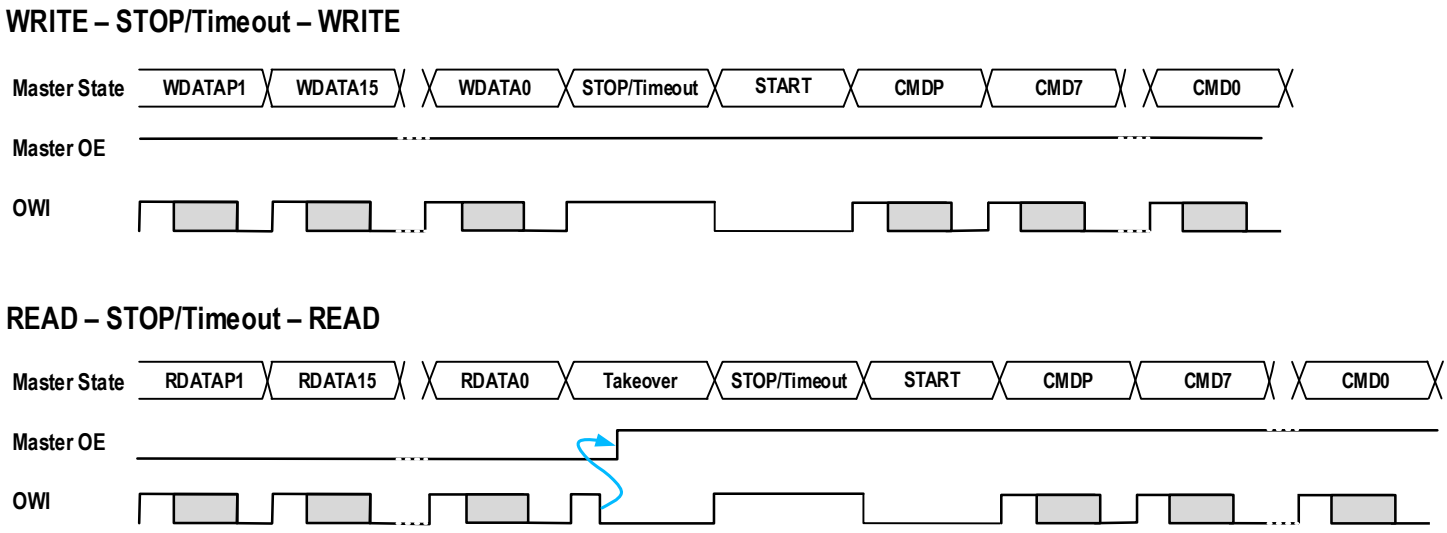


When the synchronism is lost, only the timeout can reset the communication to a stable state. The STOP condition alone is not sufficient, as it functions only within a proper, complete transaction.

In addition, each command and data word is protected with its own parity bit. In the event of a parity error, the ZMID520x ignores the command.

2.3.5 Fail-Safe Communication

Figure 9. The Timeout as a Protocol Element



The timeout protocol element can be added to the protocol flow by default. When used in a distributed environment, this approach provides an additional level of security, as each access is completely independent of the previous state of the involved interfaces.

The trade-off in using the timeout as a default protocol element is reduced bandwidth. In an undisturbed environment when there is only a low probability that the master and slave controllers are not synchronized, then not using the timeout protocol is a good option because it will increase the throughput.

2.3.6 Command Protocol

Full access consists of the transfer of a command byte to the ZMID520x plus a WRITE data word (16 bits) to the ZMID520x or a READ data word (16 bits) from the ZMID520x. After completion of a full access, the ZMID520x is ready for the next command.

In the case of a WRITE access to the EEPROM, the ZMID520x internal charge pump must be activated and the addressed memory field must be erased before the WRITE. This procedure is controlled by the ZMID520x logic, which extends the WRITE access.

Therefore after sending the command *ee_write* <00_{HEX} to $1F_{HEX}$> to the ZMID520x, the host system must pause a specific execution time t_{cmd_exe} to allow the ZMID520x to process the command. The execution time t_{cmd_exe} is specified for each command individually.

In the case of a WRITE command, the next command can be sent if the execution time has passed.

The READ command is served immediately after the takeover phase.

2.4 Command Description

The length of the OWI command is 1 byte. The first 3 bits (MSBs) represent the instruction type, and the remaining 5 LSBs represent the address to be accessed.

Figure 10. OWI Command Structure

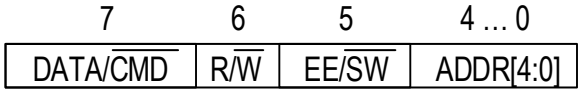


Table 1. OWI Command Field

Bit Number	Name	Description
7	DATA/not Command (DATA/$\overline{CMD}$ in Figure 10)	0: The instruction is one of the following commands: *ee_download*, *start_dpu*, *stop_dpu*, *set_security_access, ee_write* 1: The instruction is of the type “data access” EEPROM or shadow register R/$\overline{W}$ access
6	Read /not Write (R/$\overline{W}$)	1: Memory READ access 0: Memory WRITE access
5	BLSEL (EE/$\overline{SW}$)	This block-select bit determines the memory type to be accessed: 0: Shadow registers access 1: EEPROM memory access
[4:0]	Address (ADDR[4:0])	Address field in the case of a DATA instruction.

Table 2 lists the available commands. It shows the following items:

- Command Code: the 8-bit code for the command in hex format.
- Access: The type of command; i.e., WRITE, READ or action.
- t_{cmd_exec}: The execution time required to complete the command, which starts after the STOP element.
  Note: The ZMID520x oscillator must first be calibrated for the t_{cmd_exec} specification to be valid.
- Register: If applicable, the register related to the command.
- Description: The name and description of the command or the description of the register/field

Table 2. OWI Command Code Definition

Command Code	Access	t_{cmd_exec}	Register	Description
01_{HEX}	Action	120µs	–	Command: `EE_DOWNLOAD` Copy EEPROM to shadow registers[a]
02_{HEX}	Action	–	–	Command: `OWI_SIGNATURE` Signature = $83BA_{HEX}$ = Manufacturer Mode. Enable the WRITE and READ access to the complete address range of EEPROM and shadow register cells. Signature = $xxxx_{HEX}$ = Command Mode. Enable the WRITE and READ access to a limited address range of EEPROM and shadow register cells.
03_{HEX}	Action	–	–	Command: `DPU_RUN` Starts the digital processing unit (DPU).
04_{HEX}	Action	–	–	Command: `DPU_HOLD` Stop/freeze DPU.
05_{HEX} to $7F_{HEX}$	–	–	–	Unused
80_{HEX} to $9F_{HEX}$	WRITE	–	*shadow_word_*<n>	Command: `SW_WRITE <0…1F>` WRITE data to shadow word <n>
$C0_{HEX}$ to DF_{HEX}	READ	–	*shadow_word_*<n>	Command: `SW_READ <0…1F>` READ data from shadow word <n>
$A0_{HEX}$ to BF_{HEX}	WRITE	10.5ms	*eeprom_word_*<n>	Command: `EE_WRITE <0…1F>` WRITE data to EEPROM word <n>
$E0_{HEX}$ to FF_{HEX}	READ	–	*eeprom_word_*<n>	Command: `EE_READ <0…1F>` READ data from EEPROM word <n>

[a] Parameter *out_pro* = 00_{BIN} (*Output* register $0A_{HEX}$); otherwise the access mode will be terminated!

3. OWI Electrical Characteristics

Table 3. OWI Electrical Characteristics

Note: No external pull-up resistor nor external pull-down resistor should be used.

OWI (CMOS)						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Input voltage LOW	$V_{owi\_in\_low}$				1.5	V
Input voltage HIGH	$V_{owi\_in\_high}$		3.5			V
Output voltage LOW	$V_{owi\_out\_low}$		0		1.0	V
Output voltage HIGH	$V_{owi\_out\_high}$		4.0		5.0	V
Output capacitive load	C_{owi_out}				100	nF
Duration of START condition [a]	t_{owi_start}		5			µs
Duration of STOP condition [a]	t_{owi_stop}		5			µs
Bit period [a]	t_{owi_bit}		10		100	µs
Bit period deviation	Δt_{owi_bit}		0.8	1	1.2	t_{owi_bit}
High time for a logic 0 [a]	t_{owi_0}		0.175	0.25	0.375	t_{owi_bit}
High time for a logic 1 [a]	t_{owi_1}		0.625	0.75	0.825	t_{owi_bit}
Duration of timeout condition [b]	$t_{owi\_timeout}$		150			µs
Duration of slave takeover [c]	$t_{highZ,active}$	Falling edge on the SOUT pin to slave output enable Calibrated oscillator required	2.0		5.0	µs
Duration of master takeover [c]	$t_{active,highZ}$	Falling edge on S_{OUT} to master output enable Calibrated oscillator required	0.0		2.0	µs
EEPROM programming time for a single WRITE access	t_{PROG}		10.5			ms
Time to enter into Manufacturer Mode or Command Mode after power-on reset	$t_{AUTHORIZATION}$		0.0		4.5	ms

[a] See section 2.3.1 and Figure 4 for further information.

[b] See section 2.3.2, section 2.3.4, and Figure 8 for further information.

[c] See section 2.3.2, section 2.3.3, and Figure 6 for further information.

4. Glossary

Term	Definition
ATE	Automatic Test Equipment
ATPG	Automatic Test Pattern Generation
DPU	Digital Processing Unit
PPU	Position Processing Unit
E2P	Electrically Erasable Programmable Read Only Memory
ECU	Electronic Control Unit
ECC	Error Correction Code
FEC	Forward Error Correction
LSB	Least Significant Bit
MSB	Most Significant Bit
OWI	One Wire Interface

5. Revision History

Revision Date	Description of Change
November 19, 2021	Electrical characteristics conditions updated
March 20, 2018	Initial release

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

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-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.