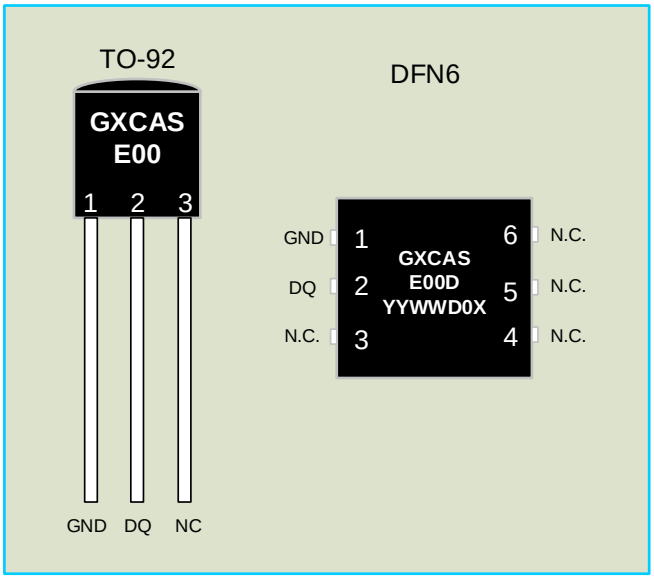
GXE00 256-Bit 1-Wire EEPROM

Features

- Read Only Memory(EEPROM) plus 64-bit one-time programmable application register Unique, factory-lasered and tested 64-bit registration number (8-bit family code + 48-bit serial number + 8-bit CRC tester) assures absolute identity because no two parts are alike
- Built-in multidrop controller ensures compatibility with other MicroLAN products
- EEPROM organized as one page of 32 bytes for random access
- Reduces control, address, data, and power to a single data pin
- Directly connects to a single port pin of a microprocessor and communicates at up to 15.3kbits per second
- 8-bit family code specifies GXE00 communication requirements to reader
- Presence detector acknowledges when reader first applies voltage
- Low cost TO-92 or 6-pin DFN surface mount package
- Reads and writes over a wide voltage range of 2.5V to 5.5V from -40° C to +85° C

Package



Pin Description

NAME		DESCRIPTION
TO-92	DFN6	
-	3,4,5,6	N.C.
3	-	N.C.
2	2	DQ
1	1	GND

Table of Contents

1 Description

The GXE00 256-bit 1-Wire® EEPROM identifies and stores relevant information about the product to which it is associated. This lot or product specific information can be accessed with minimal interface, for example a single port pin of a microcontroller. The GXE00 consists of a factory-lasered registration number that includes a unique 48-bit serial number, an 8-bit CRC, and an 8-bit Family Code (14h) plus 256 bits of user-programmable EEPROM and a 64-bit one-time programmable application register. The power to read and write the GXE00 is derived entirely from the 1-Wire communication line. Data is transferred serially via the 1-Wire protocol, which requires only a single data lead and a ground return. The 48-bit serial number that is factory-lasered into each GXE00 provides a guaranteed unique identity that allows for absolute traceability. The TO-92 and DFN6 packages provide a compact enclosure that allows standard assembly equipment to handle the device easily for attachment to printed circuit boards or wiring. Typical applications include storage of calibration constants, board identification, and product revision status.

The GXE00 operates without an external power supply. When the bus is in a high state, DQ is connected to a pull-up resistor to power the device through 1-Wire. At the same time, the bus signal in the high-level state charges the internal capacitor. When the bus is in the low-level state, the capacitor provides energy to the device. The way of providing energy is called "parasitic power supply".

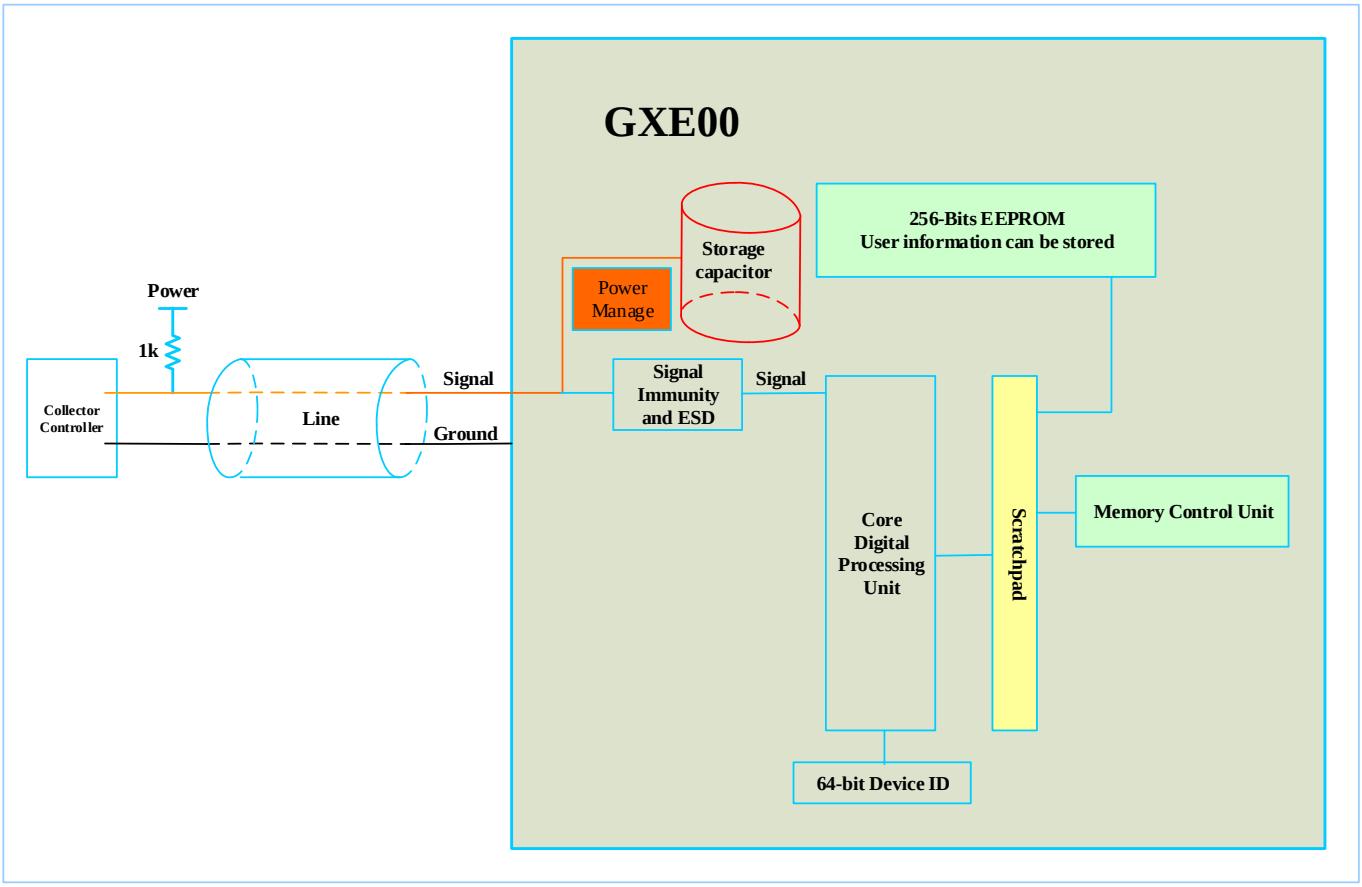


Figure 1. The GXE00 Block Diagram

The block diagram in **Figure 1** shows the relationships between the major control and memory sections of the GXE00. The GXE00 has four main data components: 1) 64-bit lasered ROM, 2) 256-bit EEPROM data memory with scratchpad, 3) 64-bit one-time programmable application register with scratchpad and 4) 8-bit status memory. The hierarchical structure of the 1-Wire protocol is shown in **Figure 2**. The bus master must first provide one of the four ROM Function Commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM. The protocol required for these ROM Function Commands is described in Figure 8. After a ROM Function Command is successfully executed, the memory functions become accessible and the master can provide any one of the four memory function commands.The protocol for these memory function commands is described in Figure 6. All data is read and written least significant bit first.

2 Power Supply

The GXE00 works in parasitic power mode. Parasitic power modes are useful in long-distance testing or space-constrained applications. The control loop for parasitic power is shown in Figure 2. When the bus is high, the control loop "steals" energy from the bus. Part of the "stealed" energy is stored in the parasitic power storage

capacitor, which is released for use by the device when the bus is low.

In parasitic power mode, the 1-Wire and storage capacitor can supply enough current to the GXE00 to meet the specified timing and voltage (see the **DC and AC Characteristics** sections) in most operations. However, when the GXE00 is performing data transfer from the register to the EEPROM, the operating current can reach 0.5mA. This current may cause an unacceptable voltage drop across weak pull-up resistors connected to a 1-Wire, requiring more current than the storage capacitor can provide. To ensure that the GXE00 has enough power supply, when copying data to EEPROM, a strong pull-up must be provided for the single bus, which is realized by directly pulling the bus to the power supply by MOSFET. During the copy data timing (twr=9ms), it must always remain in a strong pull-up state. When the strong pull-up state is maintained, no other actions are allowed. In addition to this, the size of the pull-up resistor can also be limited within a predetermined range.

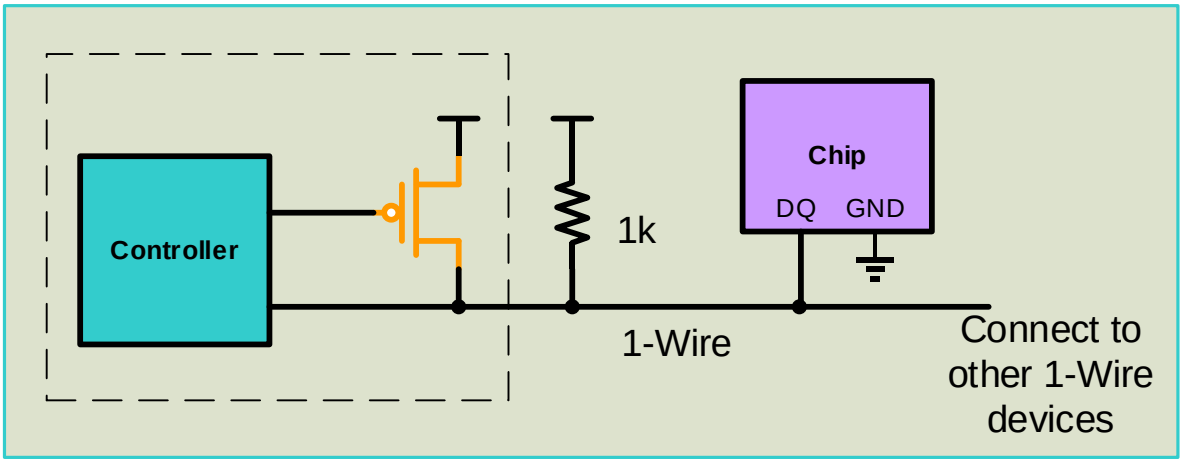


Figure 2. Parasitic Mode Replicated Data Circuit

3 64-Bit Lasered ROM

Each GXE00 contains a unique ROM code that is 64 bits long. The first 8 bits are a 1-Wire family code. The next 48 bits are a unique serial number. The last 8 bits are a cyclic redundancy check (CRC) of the first 56 bits. See the **CRC Generator** section for a detailed explanation of CRC. 64-bit ROM and ROM operation allow the GXE00 to operate as a 1-Wire device and according to the 1-Wire protocol detailed in the **1-Wire System** section.

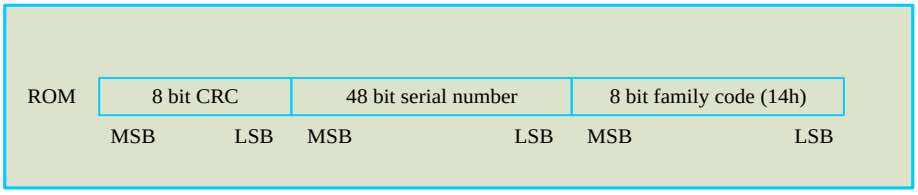


Figure 3. 64-Bit ROM

4 Memory

The memory of the GXE00 consists of three separate sections, called data memory, application register, and status register (Figure 5). The data memory and the application register each have its own intermediate storage area called scratchpad that acts as a buffer when writing to the device. The data memory can be read and written as often as desired. The application register, however, is one-time programmable only. Once the application register is programmed, it is automatically write protected. The status register indicates whether the application register is already locked or whether it is still available for storing data. As long as the application register is unprogrammed, the status register reads FFh.Copying data from the register scratchpad to the application register clears the 2 least significant bits of the status register, yielding an FCh the next time one reads the status register.

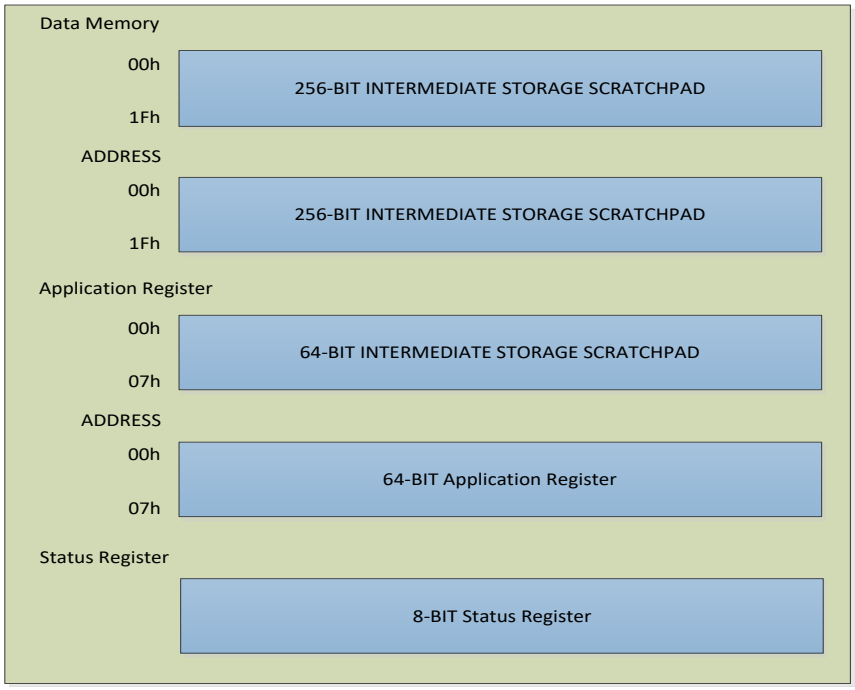


Table 1. Memory Map

5 CRC Generator

5.1 CRC-8

The CRC8 is stored in memory as part of the 64-bit ROM. The CRC8 code is calculated from the first 56 bits of the ROM and is included in the important bytes of the ROM. The CRC is calculated from the data stored in the memory, so when the data in the memory changes, the value of CRC8 also changes.

CRC8 can perform data verification when the bus controller reads the GXE00. In order to verify that the data is read correctly, the bus controller must use the received data to calculate a CRC8 value and the value stored in the

64-bit ROM of the GXE00 (when reading the ROM) or the 8-bit CRC8 value calculated inside the GXE00 (when reading register 1) to compare. If the calculated CRC8 value matches the read CRC8 value, the data is transmitted error-free. The comparison of the CRC8 value and whether to proceed to the next step is entirely at the discretion of the bus controller. The calculation formula of CRC8 is as follows:

$$CRC8 = X^8 + X^5 + X^4 + 1$$

The 1-Wire CRC8 can be generated by a polynomial generator consisting of a shift register and XOR gate, as shown in Figure 5. This loop consists of a shift register and several XOR gates, each bit of the shift register is initialized to 0. Starting with the least significant bit in ROM or byte 0 of the register, the registers are shifted one at a time. The CRC8 value is stored in the shift register after the data in the 56-bit ROM has been transferred or the MSB of the 7th byte of the register has been shifted. Next, the value of CRC8 must be cyclically shifted in. At this point, if the calculated CRC8 is correct, the shift register will be reset to 0.

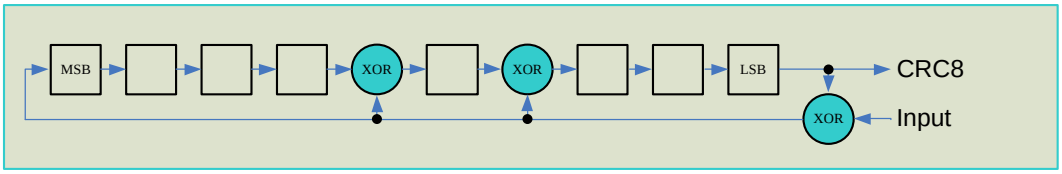


Figure 5. CRC Generator

6 1-Wire Bus System

The 1-Wire bus is a system that has a single bus master and one or more slaves. In all instances the GXE00 is a slave device. When only one slave is attached to the bus, the system is called a "single-point" system; if there are multiple slaves attached to the bus, the system called a "multi-point" system.

The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing). Finally, a 1-Wire communication example is added.

6.1 Hardware Configuration

The 1-Wire bus has only a single line by definition; it is important that each device on the bus be able to drive it at the appropriate time. To facilitate this, each device attached to the 1-Wire bus must have open-drain or three-state outputs. The 1-Wire port of the GXE00 is open drain with an internal circuit equivalent to that shown in Figure 6.

1-Wire requires an external pull-up resistor of about 1KΩ; the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus must be left in the idle state if the transaction is to resume. During resume, if the single bus is inactive (high), the bit-to-bit recovery time can be infinite. If the bus is left low for more than 480μs, all devices on the bus could be reset.

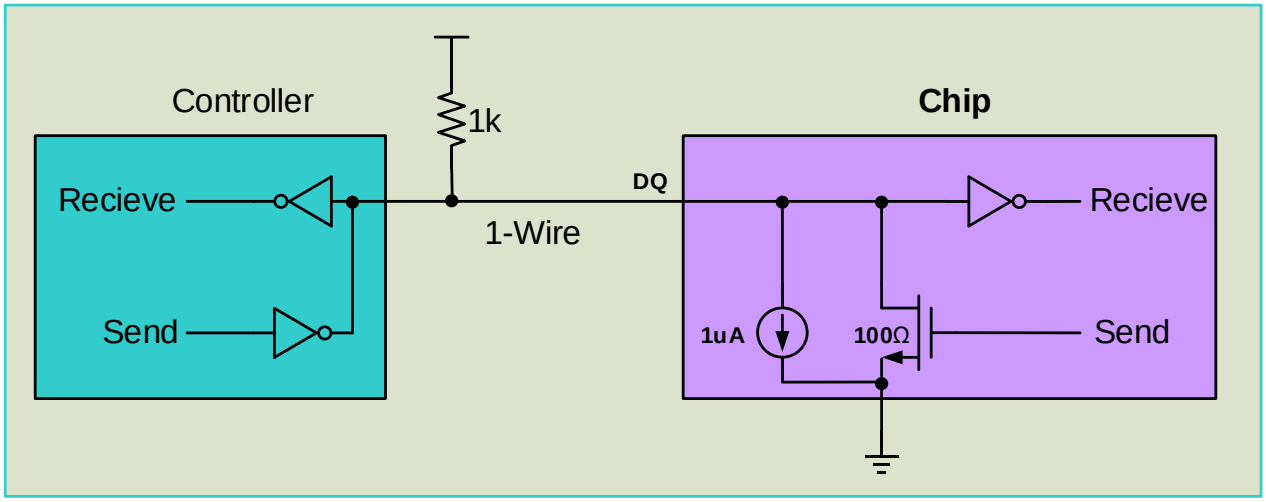


Figure 6. Hardware Configuration

6.2 Transaction Sequence

The protocol for accessing the GXE00 through the 1-Wire port is as follows:

Step1: Initialization

Step2: ROM Function Command

Step3: Memory Function Command

Each operation of the GXE00 must meet the above steps. If the steps are missing or the sequence is disordered, the device will not have a return value.

1) Initialization

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a Reset Pulse transmitted by the bus master followed by a Presence Pulse(s) transmitted by the slave(s).The Presence Pulse lets the bus master know that the GXE00 is on the bus and is ready to operate. Formore details, see the **1-Wire Signaling** section.

2) ROM Commands

Once the bus master has detected a presence pulse, it can issue one of the four ROM function commands. All ROM function commands are 8 bits long. A list of these commands follows (refer to flowchart in Figure 8):

Table 2. ROM Commands

ROM COMMAND	SEND COMMAND	FUNCTION	RETURN VALUE
Read ROM	0x33	(Single chip only)	64-bit ROM
Match ROM	0x55		None

Skip ROM	0xCC		None
Search ROM	0xF0	(Search for chips on bus)	

Read ROM [33h]

This command can only be used if there is a single slave on the bus. This command allows the bus controller to read the 64-bit serial code of the slave without using the Search ROM instruction. If more than one slave is present on the bus, a data collision occurs when all slaves try to transmit at the same time.

Match ROM [55h]

The Match ROM command, followed by a 64-bit ROM sequence, allows the bus master to address a specific GXE00 on a multidrop bus. Only the GXE00 that exactly matches the 64-bit ROM sequence responds to the subsequent memory function command. All other slaves wait for a reset pulse.

Skip ROM [CCh]

This command can save time in a single-drop bus system by allowing the bus master to access the memory functions without providing the 64-bit ROM code. It should be noted that function commands can be issued after ignoring ROM commands when only one slave is on the bus. This saves time sending 64-bit ROMs. If more than one slave is present on the bus and, for example, a read command is issued following the Skip ROM command, data collision occurs on the bus as multiple slaves transmit simultaneously.

Search ROM [F0h]

When a system is initially brought up, the bus master might not know the number of devices on the 1-Wire bus or their registration numbers. The bus controller cycles through the ROM code several times by searching for the ROM instruction to identify all slave devices. If there is only one slave on the bus, the simpler read ROM instruction can be used instead of the search ROM instruction.

3) Function Commands

After the bus controller uses ROM commands to identify the GXE00 it wishes to communicate with, the host can issue a GXE00 function command. These instructions allow the bus controller to read and write the registers of the GXE00 and perform memory operations. The functional instructions of GXE00 are detailed below, and are summarized in Table 3, and the flow is shown in Figure 8.

Table 3. Memory Function Commands

FUNCTION COMMAND	SEND	WRITE VALUE	RETURN VALUE	OTHER
Write Scratchpad	0x0F	See **Operation Example** Sections 1 and 2 for details.		
Read Scratchpad	0xAA			
Copy Scratchpad	0x55			
Read Memory	0xF0			

Write Application	0x99	
Read Status	0x66	
Read Application	0xC3	
Copy&Lock Appliction	0x5A	

Note 1: For GXE00 in parasitic power mode, writing data to EEPROM, a strong pull-up must be given to the 1-Wire, and the bus cannot have other activities during this time.

Note 2: The bus controller can abort the data transfer at any time by issuing a reset signal.

Note 3: Writing to the register is done before the reset signal is asserted.

Write Scratchpad [0Fh]

After issuing the Write Scratchpad command, the master must first provide a 1-byte address, followed by the data to be written to the scratchpad for the data memory. The GXE00 automatically increments the address after every byte it receives. After having received a data byte for address 1Fh, the address counter wraps around to 00h for the next byte and writing continues until the master sends a Reset Pulse.

Read Scratchpad [AAh]

This command is used to verify data previously written to the scratchpad before it is copied into the final storage EEPROM memory. After issuing the Read Scratchpad command, the master must provide the 1-byte starting address from where data is to be read. The GXE00 automatically increments the address after every byte read by the master. After the data at address 1Fh has been read, the address counter wraps around to 00h for the next byte and reading continues until the master sends a Reset Pulse.

Copy Scratchpad [55h]

After the data stored in the scratchpad has been verified the master may send the Copy Scratchpad command followed by a validation key of A5h to transfer data from the scratchpad to the EEPROM memory. This command always copies the data of the entire scratchpad. Therefore, if one desires to change only a few bytes of the EEPROM data, the scratchpad should contain a copy of the latest EEPROM data before the Write Scratchpad and Copy Scratchpad commands are issued. After this command and the validation key are issued, the data line must be held above V_{PUPmin} for at least t_{PROG}

Read Memory [F0h]

The Read Memory command is used to read a portion or all of the EEPROM data memory and to copy the entire data memory into the scratchpad to prepare for changing a few bytes. To copy data from the data memory to the scratchpad and to read it, the master must issue the read memory command followed by the 1-byte starting address of the data to be read from the scratchpad. The GXE00 automatically increments the address after every byte read by the master. After the data of address 1Fh has been read, the address counter wraps around to 00h for the next byte and reading continues until the master sends a Reset Pulse. If one intends to copy the entire data memory to the scratchpad without reading data, a starting address is not required; the master may send a Reset Pulse immediately following the command code.

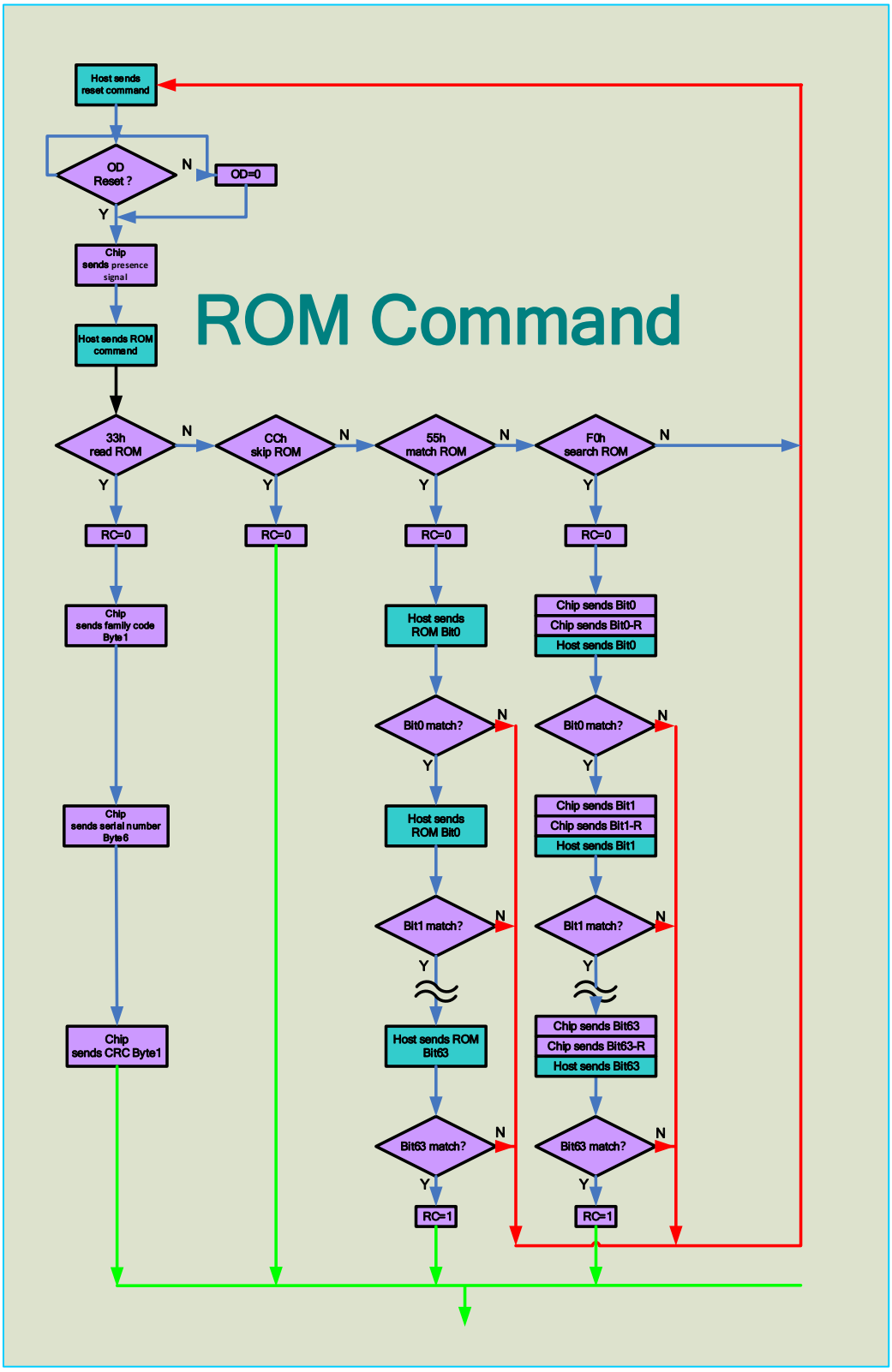


Figure 7. ROM Functions Flowchart

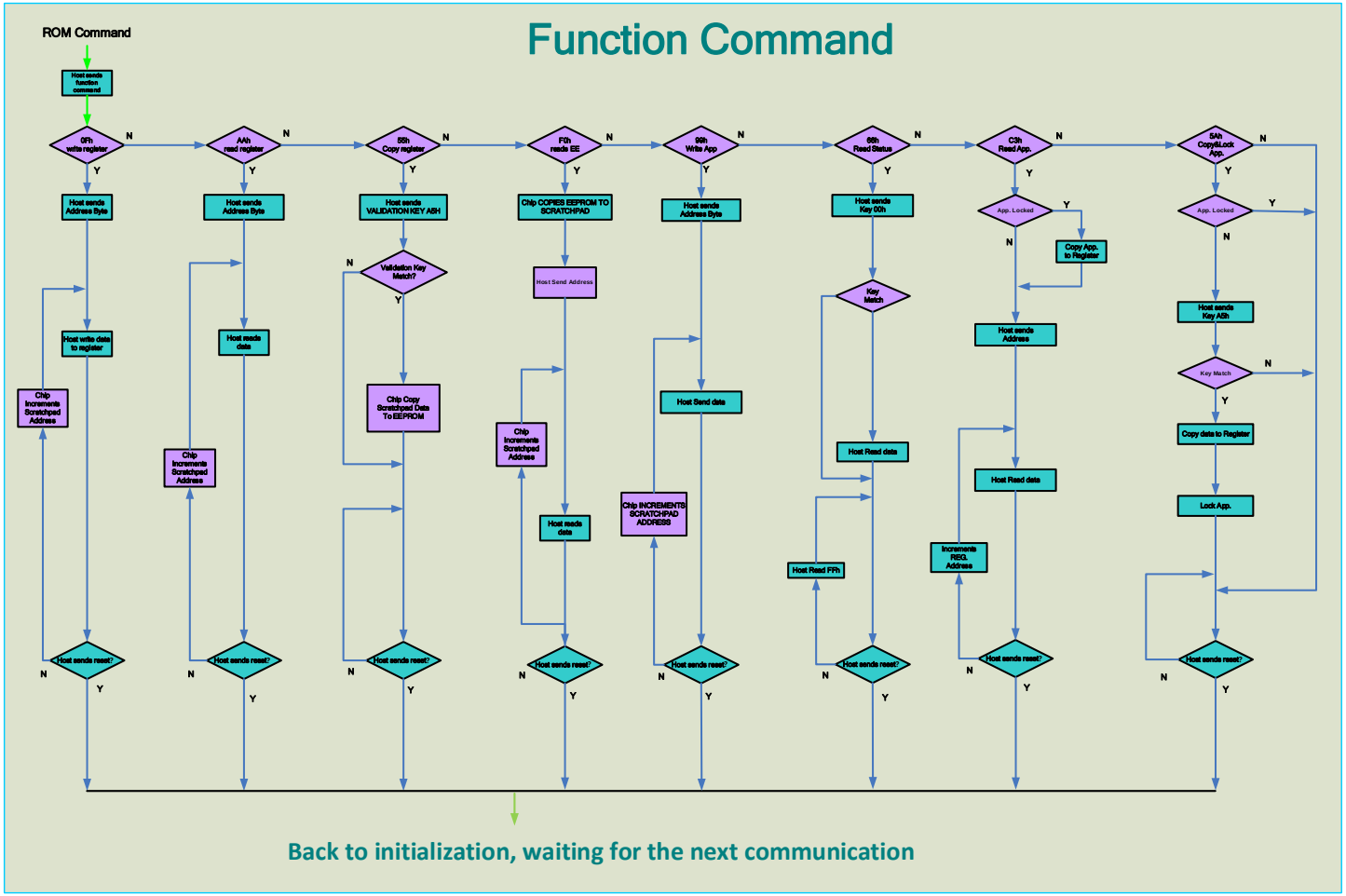


Figure 8. Function Command Flowchart

WRITE APPLICATION REGISTER [99h]

This command is essentially the same as the Write Scratchpad command, but it addresses the 64-bit register scratchpad. After issuing the command code, the master must provide a 1-byte address, followed by the data to be written. The GXE00 automatically increments the address after every byte it receives.After receiving the data byte for address 07h, the address counter wraps around to 00h for the next byte and writing continues until the master sends a Reset Pulse. The Write Application Register command can be used as long as the application register has not yet been locked. If issued for a device with the application register locked, the data written to the register scratchpad will be lost.

READ STATUS REGISTER [66h]

The status register is a means for the master to find out whether the application register has been programmed and locked. After issuing the read status register command, the master must provide the validation key 00h before receiving status information. The two least significant bits of the 8-bit status register are 0 if the application register was programmed and locked; all other bits always read 1. The master may finish the read status command by sending a Reset Pulse at any time.

READ APPLICATION REGISTER [C3h]

This command is used to read the application register or the register scratchpad. As long as the application register is not yet locked, the GXE00 transmits data from the register scratchpad. After the application register is locked the GXE00 transmits data from the application register, making the register scratchpad inaccessible for reading. The contents of the status register indicate where the data received with this command came from. After issuing the Read Application Register command, the master must provide the 1-byte starting address from where data is to be read. The GXE00 automatically increments the address after every byte read by the master. After the data at address 07h has been read, the address counter wraps around to 00h for the next byte and reading continues until the master sends a Reset Pulse.

COPY & LOCK APPLICATION REGISTER [5Ah]

After the data stored in the register scratchpad has been verified the master may send the Copy & Lock Application Register command followed by a validation key of A5h to transfer the contents of the entire register scratchpad to the application register and to simultaneously write-protect it. The master may cancel this command by sending a Reset Pulse instead of the validation key. After the validation key is transmitted, the data line must be held above V PUPmin for at least t PROG . Once t PROG has expired, the application register will contain the data of the register scratchpad. Further write accesses to the application register will be denied. The Copy & Lock Application Register command can only be executed once.

6.3 1-Wire Signaling

The GXE00 requires strict protocols to insure data integrity. The protocol consists of four types of signaling on one line: Reset Sequence with Reset Pulse and Presence Pulse, Write-0, Write-1 and Read- Data. All these signals (except Presence Pulse) are initiated by the bus master.

To get from idle to active, the voltage on the 1-Wire line needs to fall from V_{PUP} below the threshold V_{TL}.To get from active to idle, the voltage needs to rise from V_{ILMAX} past the threshold V_{TH}. The time it takes for the voltage to make this rise is seen in Figure 9 as Σ, and its duration depends on the pullup resistor(R_{PUP})used and the capacitance of the 1-Wire network attached. The voltage V_{ILMAX} is relevant for the GXE00 when determining a logical level, not triggering any events.

Figure 9 shows the initialization sequence required to begin any communication with the GXE00. A Reset Pulse followed by a Presence Pulse indicates the GXE00 is ready to receive data, given the correct ROM and memory function command. If the bus master uses slew-rate control on the falling edge, it must pull down the line for t_{RSTL} + t_F to compensate for the edge.

After the bus master has released the line it goes into Receive mode. Now the 1-Wire bus is pulled to V_{PUP} through the pullup resistor. When the threshold V_{TH} is crossed, the GXE00 waits for t_{PDH} and then transmits a Presence Pulse by pulling the line low for t_{PDL}. To detect a Presence Pulse, the master must test the logical state of the 1-Wire line at t_{MSP}. The t_{RSTH} window must be at least the sum of t_{PDHMAX}, t_{PDLMAX}, and t_{RECMIN}. Immediately after t_{RSTH} is expired, the GXE00 is ready for data communication.

1) Initialization

All communications with the GXE00 begin with an initialization sequence shown in Figure 9. A reset pulse followed by a presence pulse indicates that the GXE00 is ready to transmit and receive data.

During the initialization sequence, the bus controller pulls the bus low for 480us to issue (TX) a reset pulse, then releases the bus and enters the receive state (RX). When the bus is released, a 1kΩ pull-up resistor pulls the bus high. When the GXE00 detects the rising edge on the IO pin, it waits for 15-60us, and then sends out a presence pulse consisting of a 60-240us low-level signal.

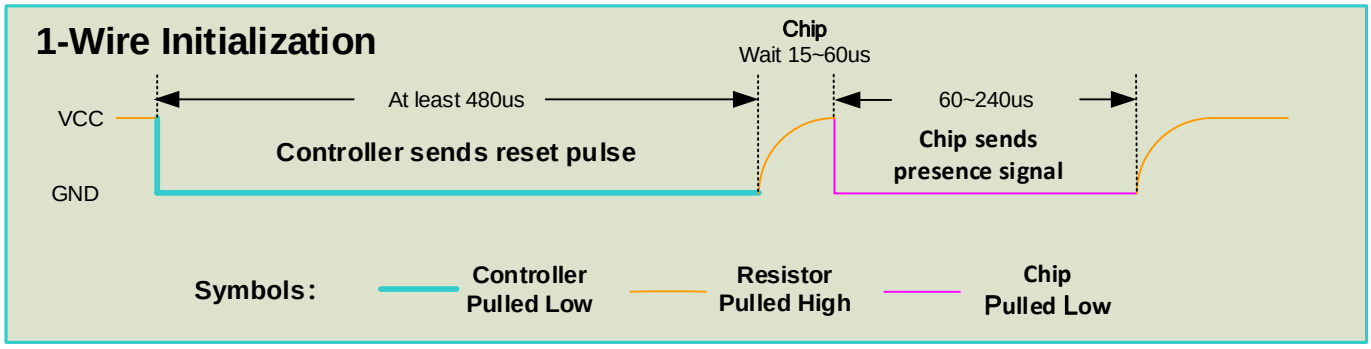


Figure 9. Initialization Timing

2) Read/Write Time Slots

Data communication with the GXE00 takes place in time slots that carry a single bit each.

Write Time Slots

The GXE00 has two write timings: write 1 and write 0. The bus controller writes logic 1 through the write 1 sequence; logic 0 is written through the write 0 sequence. The write sequence must last at least 60us, including a recovery time of at least 5us between two write cycles. The write sequence begins when the bus controller pulls the data line from logic high to low (see Figure 10).

To generate a write sequence by the bus controller, the data line must be pulled to a low level and then released, and the bus must be released within 15us. When the bus is released, the pull-up resistor pulls the bus high. To generate a write 0 sequence, the bus controller must pull the data line to a low level and continue to hold it for at least 60us.

After the bus controller initializes the write sequence, the GXE00 uses the signal lines within a 15us to 60us window. If the line is high, write 1. Conversely, if the line is low, write 0.

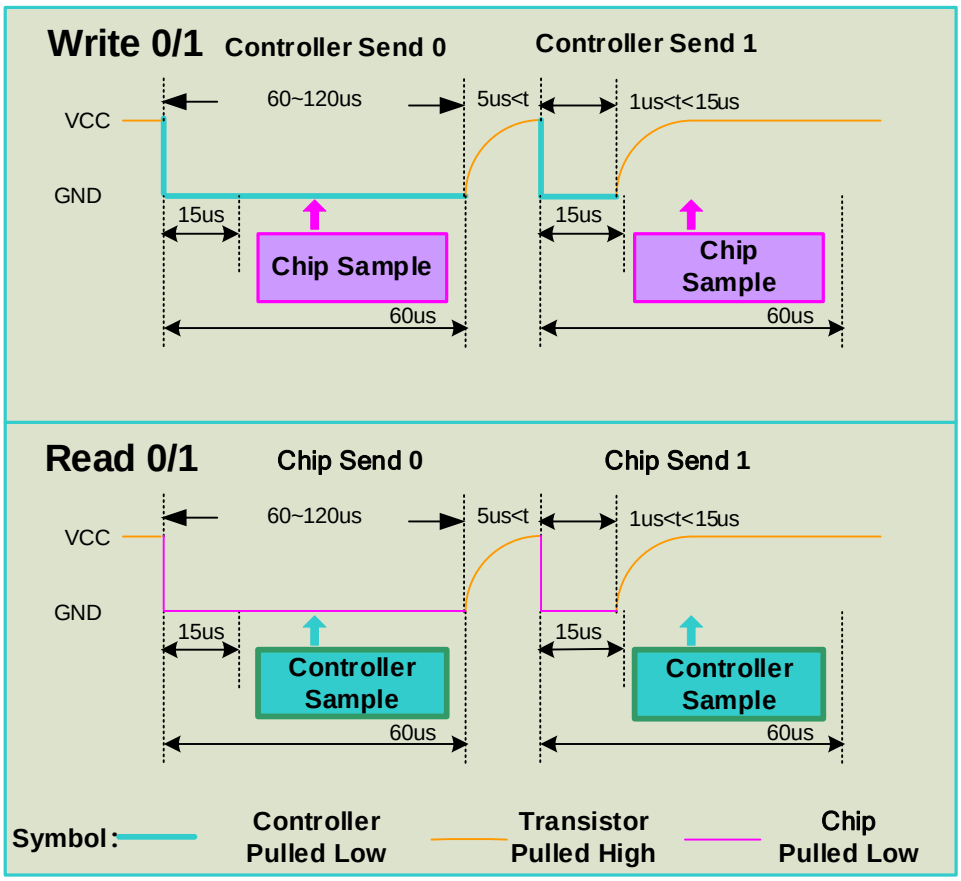


Figure 10. Read/Write Time Slot Timing Diagram

Read Time Slots

When the bus controller initiates a read sequence, the GXE00 is only used to transfer data to the controller. Therefore, the bus controller must start the read sequence immediately after issuing the read register command so that the GXE00 can provide the requested data. In addition, the bus controller reads the timing after sending the temperature conversion command or recalling the EEPROM command, see the **GXE00 Function Command** section for details.

All read sequences must be at least 60us, including a recovery time of at least 5us between two read cycles. When the bus control pulls the data line from high to low, the read sequence begins, the data line must be held for at least 1us, and then the bus is released (see Figure 10). After the bus controller issues a read sequence, the GXE00 transmits 1 or 0 by pulling high or low on the bus. When the transfer of 0 is completed, the bus will be released and return to the high-level idle state through the pull-up resistor. The data output by GXE00 is valid within 15us after the falling edge of the read timing. Therefore, the bus controller releases the bus within 15us from the beginning of the read sequence and then samples the bus state to read the state of the data line.

Figure 11 indicates that the sum of T_{INIT}, T_{RC}, and T_{SAMPLE} must be less than 15us. Figure 12 indicates that the

system time can be maximized by keeping the T_{INIT} and T_{RC} as short as possible, and placing the controller sampling time at the end of the 15us period.

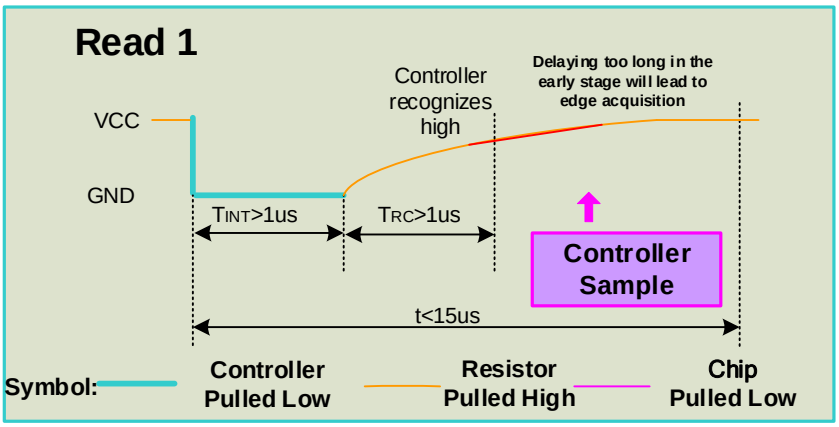


Figure 11. Detailed Timing of Controller Read 1

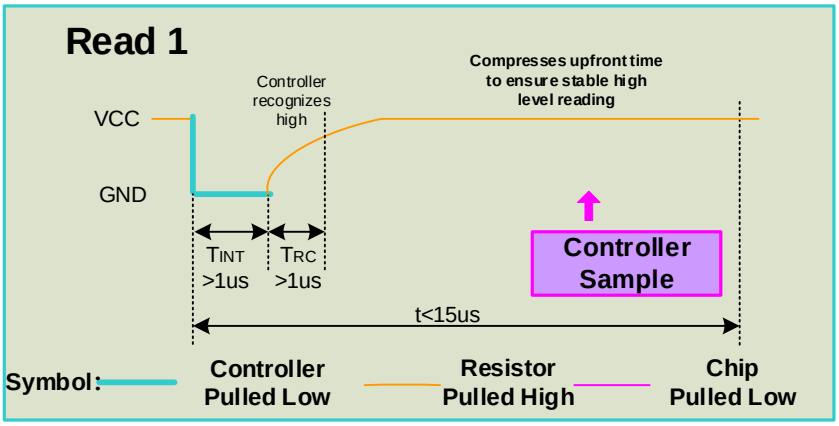


Figure 12. Recommended Controller 1 Timing

6.4 GXE00 Operation Example:

主机模式	数据（最低有效位优先）	说明
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Issue "Skip ROM" command
TX	0Fh	Issue "Write Scratchpad" command
TX	00h	beginning offset = 00h
TX	<8 data bytes>	Write 8 bytes of data to scratchpad
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Issue "Skip ROM" command
TX	AAh	Issue "Read Scratchpad" command
TX	00h	beginning offset = 00h
RX	<8 data bytes>	Read the memory
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Issue "Skip ROM" command
TX	55h	Issue "Copy Scratchpad" command
TX	A5h	Send authentication code
TX	<Data Line High>	Wait $t_{PROGmax}$ for the copy function to complete
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Issue "Skip ROM" command
TX	F0h	Issue "Read Memory" command
TX	00h	beginning offset = 00h
RX	<32 Bytes>	Read the memory
TX	Reset	Reset pulse
RX	Presence	Presence pulse

7 Absolute Maximum Ratings

Voltage range of each pin to ground .....-0.5V to +6.0V

Range of working temperature.....-40°C to +85°C

Storage range.....-55°C to +150°C

Soldering temperature range ..... 300℃(10S)

8 DC Characteristics

Test conditions: (-40 ℃~+85 ℃; VDD = 2.5V~5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
Pullup Supply	V_{PU}	Parasite Power	+2.5		+5.5	V	1,2
Input Logic-Low	V_{IL}		-0.3		+0.8	V	1,3,4
Input Logic-High	V_{IH}	Parasite Power	+2.5		The lower of 5.5 or VDD + 0.3	V	1,6
Sink Current	I_L	VI/O = 0.4V	4.0			mA	1
Standby Current	I_{DDS}			750	1000	nA	6,7
Active Current	I_{DD}	VDD = 5V		1	1.5	mA	8
DQ Input	I_{DQ}			1		uA	9

Notes:

1) All voltages are referenced to ground potential.

2) The pull-up voltage is obtained like this: Assuming that the pull-up device is perfect, the high level of the pull-up should be equal to VPU. In order to meet the VIH specification of the GXE00, the actual transistor pull-up power supply must include the limit of the voltage drop; thus $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$.

3) A logic 0 level is obtained when the sink current is 4mA.

4) Low voltage state in parasitic power mode, VILMAX may have to be lowered to 0.5V in order to guarantee the presence of pulses.

5) Logic 1 voltage is obtained when the source current is 1mA.

6) Standby current is defined at 70°C; typical standby current value is 3uA at 125°C.

7) To reduce IDDS, the range of DQ is as follows: GND ≤ DQ ≤ GND + 0.3V or VDD – 0.3V ≤ DQ ≤ VDD.

8) Dynamic current involves writing to EEPROM memory.

9) DQ data line is high (“high impedance” state).

9 AC Characteristics- Non-Volatile Memory

Test conditions: (-40℃~+85℃; VDD = 2.5V~5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
NV Write Cycle Time	t_{WR}			8	10	ms	
EEPROM Writes	N_{EEWR}		1k			writes	
EEPROM Data Retention	t_{EEDR}		40			years	

10 AC Characteristics

Test conditions: (-40℃~+85℃; VDD = 2.5V~5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
Time to Strong Pullup On	t_{SPON}				10	us	
Time Slot	t_{SLOT}		60		120	μs	1
Recovery Time	t_{REC}		5			μs	1
Write 0 Low Time	t_{LOW0}		60		120	μs	1
Write 1 Low Time	t_{LOW1}		1		15	μs	1
Read Data Valid	t_{RDV}				15	μs	1
Reset Time High	t_{RSTH}		480			μs	1
Reset Time Low	t_{RSTL}		480			μs	1
Presence-Detect High	t_{PDHIGH}		15		60	μs	1
Presence-Detect Low	t_{PDLOW}		60		240	μs	1
Capacitance	$C_{IN/OUT}$				25	pF	

Notes

1) See Figure 14 for timing.

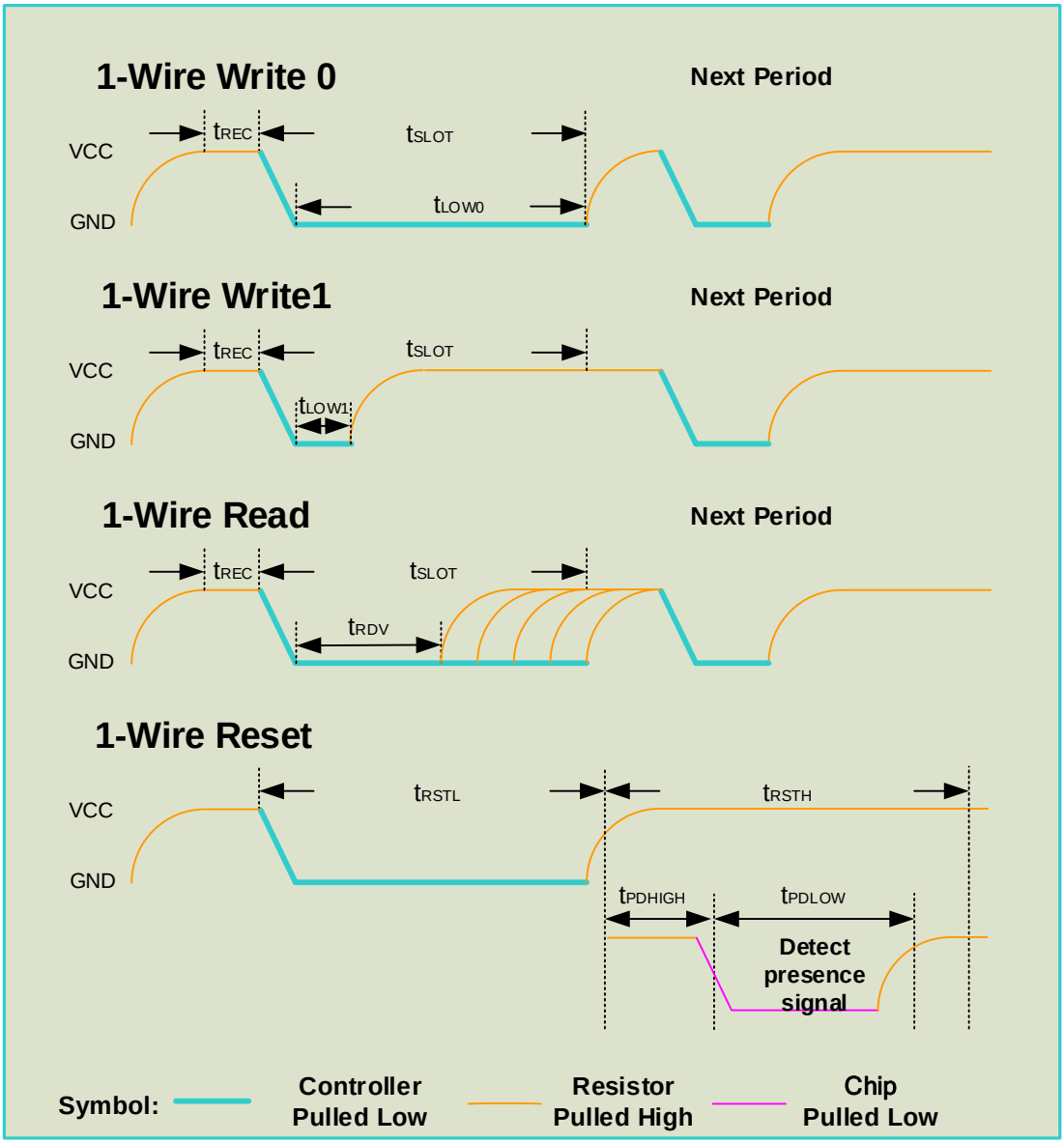
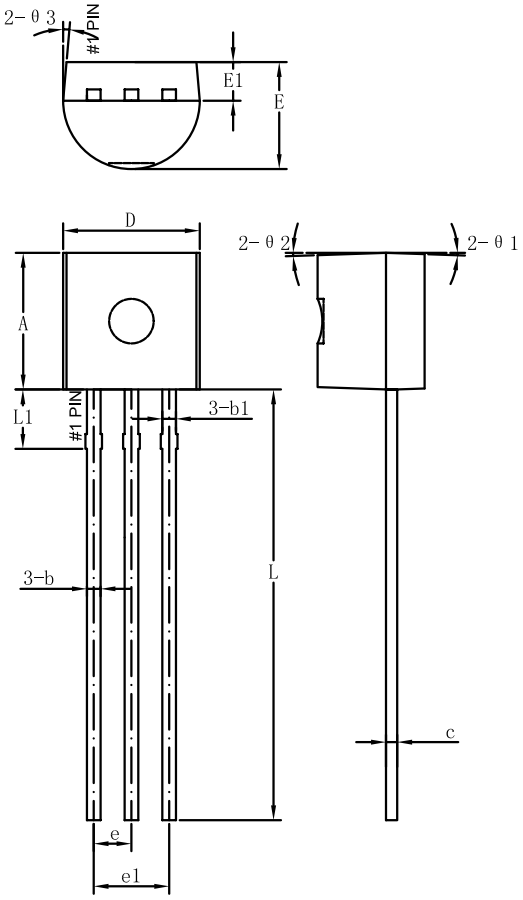


Figure 14. 1-Wire Timing

11 Product Package Model List

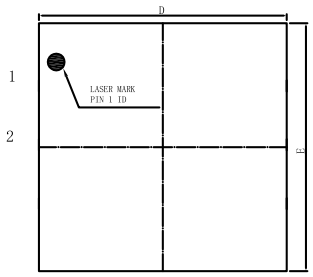
MODEL	PACKAGE
GXE00	TO-92 (Pin 3)
GXE00D	DFN6

1）TO-92

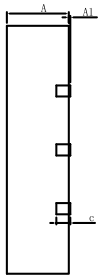


符号	机械尺寸/mm		
	最小值	典型值	最大值
A	4.5	4.6	4.7
b	0.38	0.45	0.56
b1		0.45	
c	0.36	0.38	0.51
D	4.5	4.6	4.7
E	3.45	3.6	3.75
E1	1.2	1.3	1.4
e		1.27	
e1		2.54	
L	13.5	14.5	15.3
L1		1.96	
θ1		2°	
θ2		2°	
θ3		5°	

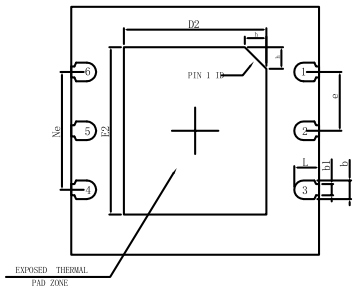
2) DFN6



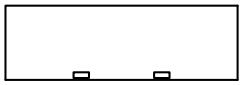
TOP VIEW



SIDE VIEW



BOTTOM VIEW



SIDE VIEW

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	—	0.02	0.05
b	0.25	0.30	0.35
b1	0.15	0.18	0.21
c	0.203REF		
D	3.90	4.00	4.10
D2	2.20	2.30	2.40
e	0.95BSC		
Ne	1.90BSC		
E	3.90	4.00	4.10
E2	2.60	2.70	2.80
L	0.35	0.40	0.45
h	0.35		

Ordering Imformation

Purchase Number	Device	PIN-Package	SPQ	Remarks
GXE00-Bu	GXE00	3-T092	2000	Bulk
GXE00D_T&R	GXE00D	6-DFN	4000	Tape and reel