Keywords: 1-WIRE, temperature sensor, PEC, CRC,error checking

APPLICATION NOTE 6797

HOW TO USE PACKET-ERROR CHECKING TO SECURE YOUR TEMPERATURE READING

Abstract: "Packet-Error Checking" (PEC) is an error detection mechanism wildly used during data transmission. Some MAXIM products featured PEC mode to increase reliability of data transfer. This application note discussed detailed implementation of PEC byte on temperature sensor product with 1-wire and 2-wire interface.

Background

Communication happens everywhere, and errors can result in a decrease in communication efficiency. To make communication more accurate, people use different methods to detect communication errors. For example, when creating a new password for a website login, the user is required to enter the password twice to decrease the chances of typos in the password.

When people write numbers such as bank routing numbers, it is possible for them to make errors. Check digits were created to catch these transcription errors. A check digit is a type of redundancy check that consists of one or more digits computed by an algorithm from the other digits in the sequence. The last digit of a bank routing number is a check digit that is calculated based on the first eight digits and used to verify the authenticity of the bank routing number during transaction.

Data communications can be subject to errors, such as channel noise, electrical distortion, random bit errors, and cross talk. A cyclic redundancy check (CRC) is an error-detecting code used to detect accidental errors in data transmissions. CRCs are used in several Maxim temperature sensor products with a 1-Wire® interface (i.e., DS18B20, MAX31850). Some products with an I^2C/SMBus-compatible serial interface (i.e., DS1862, MAX31875) implement CRCs in the form of packet error checking (PEC), which is a mechanism that was originally defined in SMBus. In a data transmission system, a PEC byte can be appended at the end of each transaction as error-detecting code. The PEC byte is calculated based on a CRC-8 byte represented by the polynomial $C(X) = X^8 + X^2 + X^1 + 1$. The PEC mechanism improves reliability and communication robustness and PEC implementation is optional for SMBus devices.

Description

Each 1-Wire device has a unique 64-bit serial code stored in on-board ROM. When multiple devices are on the same bus, the bus master uses the 64-bit unique ROM ID to uniquely identify each slave device on the bus, which allows the master to determine the number of slaves and their device types. When the master wants to communicate with one specific slave device, the master issues a command to the devices on the bus followed by the target device's 64-bit ROM code sequence to address that specific slave device. Only the slave that exactly matches the 64-bit ROM code sequence responds to the function command issued by the master.

In 1-Wire interface products, CRC bytes are provided as part of the 64-bit ROM code (**Figure 1**) and in the 9th byte of the scratchpad memory (**Figure 2**). CRC bytes in the 64-bit ROM code are calculated from the first 56 bits of the ROM code, which include the serial number and family code.

The CRC bytes in the scratchpad memory are calculated from bytes 0 through 7 of the scratchpad and change when the data in the scratchpad changes. CRC bytes in the scratchpad are read only. As an example, to read the temperature value from a 1-Wire temperature sensor, the master issues a Read Scratchpad command to read the scratchpad including the CRC bytes. The master then recalculates the CRC bytes of the first eight data bytes from the scratchpad and compares the calculated CRC bytes with the read CRC bytes. If they match, the data received is error free.

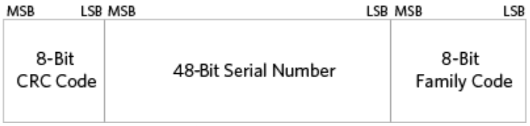


Figure 1. 64-bit 1-Wire ROM code.

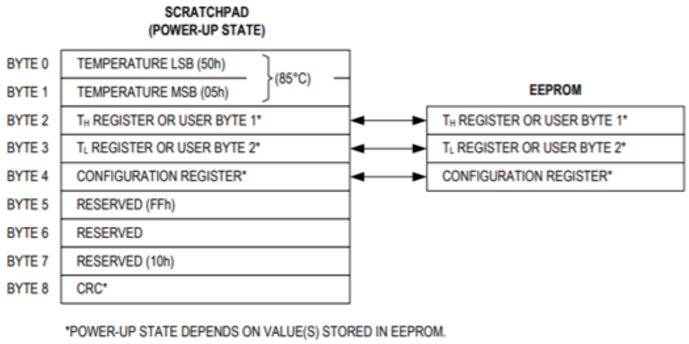
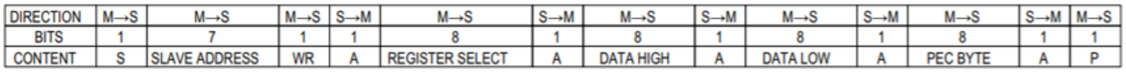


Figure 2. DS18B20 scratchpad memory.

For I^2C/SMBus components$_2$ that support PEC, CRC bytes can be used in both write and read. For example, the MAX31875, a tiny, micropower local temperature sensor with an I C/SMBus interface, supports selectable PEC mode.

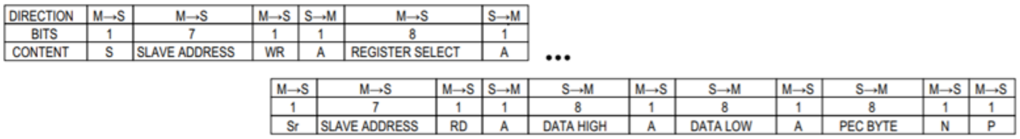
During a write transaction, the master writes the MAX31875's address, waits for an ACK bit from the MAX31875, and then the master transmits the target register, followed by another ACK bit from the MAX31875. The master writes two data bytes and receives ACK bits from the MAX31875 for each data byte. With PEC mode on, the master sends one more CRC byte and receives the last ACK bit from the MAX31875, stopping the transaction. This CRC byte is calculated using the slave address, register address, and transmitted data.

For a read transaction, the master transmits the MAX31875's address and the target register address and receives an ACK bit for each transmission from the slave. The master generates a REPEAT START (Sr) byte and writes the MAX31875 address and a read bit. The MAX31875 then acknowledges the address/read byte and transmits the two data bytes. With PEC mode on, a PEC byte is appended by the MAX31875 after the data transmission. A CRC byte is calculated using a slave address with a write bit, a register address, a slave address with a read bit, and the transmitted data.



DIRECTION	M→S	M→S	M→S	S→M	M→S	S→M	M→S	S→M	M→S	S→M	M→S	S→M	M→S
BITS	1	7	1	1	8	1	8	1	8	1	8	1	1
CONTENT	S	SLAVE ADDRESS	WR	A	REGISTER SELECT	A	DATA HIGH	A	DATA LOW	A	PEC BYTE	A	P

Figure 3. 2-Byte write to the MAX31875 with the PEC code.



DIRECTION	M→S	M→S	M→S	S→M	M→S	S→M
BITS	1	7	1	1	8	1
CONTENT	S	SLAVE ADDRESS	WR	A	REGISTER SELECT	A

•••

M→S	M→S	M→S	S→M	S→M	M→S	S→M	M→S	S→M	M→S	M→S
1	7	1	1	8	1	8	1	8	1	1
Sr	SLAVE ADDRESS	RD	A	DATA HIGH	A	DATA LOW	A	PEC BYTE	N	P

Figure 4. SMBus 2-Byte read with the PEC byte.

Example 1: 1-Wire Reading with CRC

The DS18B20 is one of Maxim's digital thermometers with a 1-Wire interface. CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The DS18B20's ROM CRC byte is calculated using a 48-bit serial number and an 8-bit family code (28h). The example in **Table 1** uses the serial number 04 16 74 8A 15 FF.

Table 1. 64-bit 1-Wire ROM Code with Value

Format	CRC-8 (MSB)	Serial Number						Family Code (LSB)
Hex	72	04	16	74	8A	15	FF	28
Binary	0111 0010	0000 0100	0001 0110	0111 0100	1000 1010	0001 0101	1111 1111	0010 1000

To calculate the CRC-8 byte, the master uses a polynomial generator, as shown in **Figure 5**. The CRC generator consists of a shift register and XOR gates, with all shift register bits initialized to 0. Starting with the least significant bit of the ROM code, one bit at a time is shifted into the shift register. After shifting in the 56th bit from the ROM, the polynomial generator contains an 8-bit CRC value.

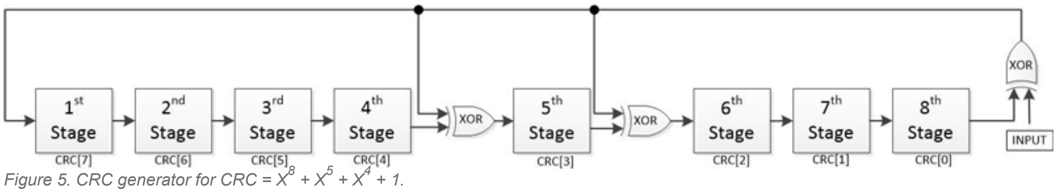


Figure 5. CRC generator for CRC = $X^8 + X^5 + X^4 + 1$.

Further details about the example CRC byte calculations ›

In this example, the master calculates the CRC-8 byte based on the 56 bits of ROM code received, which results in the value 0x72. The master compares the calculated CRC value (0x72) with CRC byte stored in the DS18B20's ROM (0x72), which is the same as the calculated value and confirms that the master reading is correct.

The DS18B20's scratchpad memory CRC byte is calculated using byte 0 to byte 7 in the scratchpad. See **Table 2** for an example of the scratchpad memory contents.

Table 2. DS18B20 Scratchpad Memory Example

Byte 8 CRC Byte	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1 Temperature MSB	Byte 0 Temperature LSB
05	10	0C	FF	7F	18	1B	05	50
0000 0101	0001 0000	0000 1100	1111 1111	0111 1111	0001 1000	0001 1011	0000 0101	0101 0000

Starting with the least significant bit (LSB) of byte 0 in the scratchpad, one bit at a time is shifted into the shift register of the CRC generator. The master calculates 0x05 as the 8-bit CRC value after shifting in the 64th bit from the scratchpad.

Further details about the example CRC byte calculations ›

The master compares the calculated value (0x05) with the scratchpad CRC byte (0x05). If a match, the master confirms the reading from the scratchpad is correct.

Example 2: I^{2}C/SMBus Writing with PEC

The temperature threshold register (T_{OS}) is used to set the temperature limit of the MAX31875. If the MAX31875 measured temperature exceeds TOS, the configuration register shows an overtemperature status. The TOS has a power-on state of 80°C (0x5000), where the address is 0x03. To set T_{OS} to 95°C (0x5F00), the master writes to the MAX31875, as shown in **Table 3**.

Table 3. Writing to T_{OS} with PEC Mode On

Direction	M->S	M->S	M->S	S->M	M->S	S->M	M->S	S->M	M->S	S->M	M->S	S->M	M->S
Content	S	Slave address	WR	A	Register address	A	Data High	A	Data Low	A	PEC Byte	A	P
Binary		1001 000	0		0000 0011		0101 1111		0000 0000		0010 0100		

The master calculates the PEC-8 byte using the PEC generator shown in **Figure 6**. Starting from the first bit of the slave address (MSB), 0x90035F00 is shifted into the shift register one bit at a time to calculate 0x24.

Further details about the example PEC byte calculations ›

The master sends 0x90035F0024 to the MAX31875 and receives an ACK because 0x24 matches the PEC byte generated by the slave. The slave sends an ACK to the master if the received PEC byte is a match.

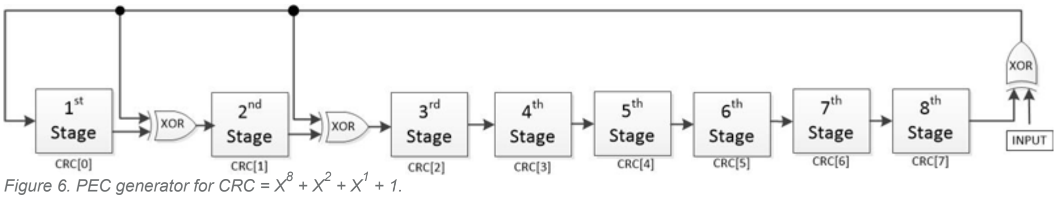


Figure 6. PEC generator for CRC = $X^8 + X^2 + X^1 + 1$.

Example 3: I^{2}C/SMBus Reading with PEC

Set the MAX31875 resolution to be 12 bits. The LSB has a value of 0.0625°C. With PEC mode on, read the MAX31875's temperature register value (address 0x00). The MAX31875 temperature data format is 16 bits, two's complement, and the register is read out in 2 bytes: an upper byte and a lower byte. The temperature register bits D[15:3] contain the temperature data. To read MAX31875's temperature register, the master sends the slave address with a write command (0x90), receives an ACK bit, sends the temperature register address (0x00), and receives an ACK bit. Continuing with a repeat start, the master sends the slave address with a read command (0x91), receives an ACK bit, and the MAX31875 sends back two data byte values with a PEC byte attached.

Table 4 is an example of the master's reading of the MAX31875's temperature register value of 23.00°C (0x1700).

Table 4. Reading of MAX31875 Temperature Register with PEC mode on

Direction	M->S	M->S	M->S	S->M	M->S	S->M	...	M->S	M->S	M->S	S->M	S->M	M->S	S->M	M->S	S->M	M->S	M->S
Content	S	Slave address	WR	A	Register address	A		1	Slave address	RD	A	Data High	A	Data Low	A	PEC Byte	N	P
Binary		1001 000	0		0000 0011			Sr	1001 000	1		0001 0111		0000 0000		0101 1011		

During the read operation, the MAX31875 sends the master the temperature register value (0x1700) and the PEC byte (0x5B). The master calculates the PEC byte using the PEC generator shown in Figure 6. Starting from the MSB of the slave address, 0x9000911700 is shifted into the shift register one bit at a time. The master compares the received PEC byte with the calculated PEC byte from the PEC generator, which is the same value, and confirms that the reading of the temperature register is correct.

Further details about the example PEC byte calculations ›

Conclusion

By using CRC or PEC, the master and slave can verify the data received and detect transmission errors. Especially in situations where multiple devices are connecting with the same host at the same time, the cyclic redundancy check provides an efficient way for error checking.

Trademarks

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Related Parts		
DS18B20	Programmable Resolution 1-Wire Digital Thermometer	Free Samples
MAX31875	Low-Power I^{2}C Temperature Sensor in WLP Package	Free Samples

More Information
For Technical Support: https://www.maximintegrated.com/en/support
For Samples: https://www.maximintegrated.com/en/samples
Other Questions and Comments: https://www.maximintegrated.com/en/contact

Application Note 6797: https://www.maximintegrated.com/en/an6797
APPLICATION NOTE 6797, AN6797, AN 6797, APP6797, Appnote6797, Appnote 6797