Application Report
SLUA363–October 2005

Calculating CRC With TI Battery Management Products

Michael Vega — *Battery Management*

ABSTRACT

Certain Texas Instruments (TI) products such as the 1K-bits EPROM bq2022, the battery monitor bq2023, and the security IC bq26150 require the host to be able to calculate a cyclic redundancy check (CRC) based on a specific polynomial. The purpose of this document is to briefly discuss what a CRC is, how it is used within the mentioned TI products, and how to implement it within a system that interacts with these products.

Contents

List of Tables

1 What is CRC?

The cyclic redundancy check (CRC) calculation is a form of polynomial modulo 2 arithmetic. The CRC is typically used as a calculation that returns a checksum based on a block of data and a set polynomial with binary coefficients. It is useful for error detection during data transmission.

2 CRC Used as a Data Verifier

In the bq2022 and bq2023, the CRC is used to determine the integrity of data that has been transmitted across the single-wire communication line. These devices return CRC values in the form of data after certain commands have been transmitted to ensure that the device has understood the command along with parameters such as register addresses as it was intended to be sent by the communication bus master or host. After certain data that has been read from the devices, another CRC value is obtained to verify the integrity of the data received by the host. It is up to the host to calculate a CRC for each time that the TI device returns a CRC value so that the host can decide if the last command should be repeated or not, relying on matched CRC values. The CRC values are based on an 8-bit polynomial which is defined as $X^8 + X^5 + X^4 + 1$.

3 CRC Used as an Encryption Scheme

The bq26150 uses the CRC as an authenticating scheme. The CRC polynomial and seed are defined by values that are contained in private memory. The host must know these values either by having knowledge of them up-front or by decrypting cipher text versions of these values that are contained in public memory of the device through a root key. The CRC polynomial and the seed are 16 bits each. Data that consists of a 96-bit ID (contained in private memory) and a 32-bit random challenge is input into the CRC generator and returns a 16-bit result which is used to authenticate a peripheral device that may be connecting to the IC.

4 CRC Calculation Example

Before attempting a CRC calculation some terms must be defined:

CRC polynomial– Determines the size of a given CRC calculation result . The highest value exponent in the polynomial indicates how many bits are in the CRC. For example, $X^8 + X^5 + X^4 +1$ corresponds to an 8-bit CRC while $X^{16} + X^{11} + X^5 + X^2 +1$ corresponds to a 16-bit CRC.

CRC seed– The initial value of the CRC. For the bq2022 and the bq2023, it has a value of 0x00, whereas in the bq26150 a user-programmable CRC seed is used.

Consider this example of how to calculate a CRC without the use of a computer program. The table shown in Table 2 is helpful in arranging the data in an organized manner to avoid confusion. This example uses the 8-bit polynomial of $X^8 + X^5 + X^4 +1$. When placing the polynomial in this table, arrange the polynomial as $1 + X^4 + X^5 + X^8$.

A value can be obtained from the polynomial if the coefficients of each power of X are arranged in the order shown in Table 1. Notice that the coefficient for X^8 is not used. That coefficient only provides the size of the CRC result. The value for this example turns out to be 8C (hexadecimal). For the remainder of this document, when referring to the polynomial, this is the value used.

Table 1. Coefficients for Example Polynomial

X^0	X^1	X^2	X^3	X^4	X^5	X^6	X^7	X^8
1	0	0	0	1	1	0	0	

Once the polynomial is determined, the CRC seed must be identified. A seed value of 0x00 is used in this example. This corresponds with the CRC operation of a bq2022 or bq2023. Remember that in bq26150, this value is chosen by the user when programming it into the one-time-programmable memory of the device.

The last thing needed to begin calculating the CRC is the data that will actually pass through the CRC. The data for this example is 0x0F.

- Compare the least significant byte (LSB) of the CRC with the LSB of data.
- If they are not equal, then shift both values to the right and then add (bit-wise XOR) the polynomial coefficient to the CRC after being shifted.
- If they are equal, only shift right CRC and data.
- Repeat procedure until 8 sets of LSB bits have been compared.
- The last CRC value is the final solution.

These steps can be followed while verifying with Table 2. The final result of the CRC is 0x41. If another byte of data were to be pushed through the CRC, then the procedure done in Table 2 would be repeated, substituting 0x41 as the CRC seed. When calculating larger CRCs such as a 16-bit CRC, the table is adjusted so that all 16 coefficients of the polynomial are considered.

Table 2. Example of CRC Calculation

BYTE	ACTION									
Seed	Initial	0	0	0	0	0	0	0	0	
Data value	Initial	0	0	0	0	1	1	1	1	
Data value	After 1st shift	0	0	0	0	0	1	1	1	
CRC	After 1st shift	0	0	0	0	0	0	0	0	
Poly		1	0	0	0	1	1	0	0	
CRC	After Adding	1	0	0	0	1	1	0	0	
Data value		0	0	0	0	0	1	1	1	
Data value	After 2nd shift	0	0	0	0	0	0	1	1	
CRC	After 2nd shift	0	1	0	0	0	1	1	0	
Poly		1	0	0	0	1	1	0	0	
CRC	After Adding	1	1	0	0	1	0	1	0	
Data value		0	0	0	0	0	0	1	1	
Data value	After 3rd shift	0	0	0	0	0	0	0	1	
CRC	After 3rd shift	0	1	1	0	0	1	0	1	
Poly		1	0	0	0	1	1	0	0	
CRC	After Adding	1	1	1	0	1	0	0	1	
Data value		0	0	0	0	0	0	0	1	
CRC	After 4th shift	0	1	1	1	0	1	0	0	
Data value	After 4th shift	0	0	0	0	0	0	0	0	
CRC	After 5th shift	0	0	1	1	1	0	1	0	
Data value	After 5th shift	0	0	0	0	0	0	0	0	
CRC	After 6th shift	0	0	0	1	1	1	0	1	
Data value	After 6th shift	0	0	0	0	0	0	0	0	
Data value	After 7th shift	0	0	0	0	0	0	0	0	
CRC	After 7th shift	0	0	0	0	1	1	1	0	
Poly		1	0	0	0	1	1	0	0	
CRC	After Adding	1	0	0	0	0	0	1	0	
Data value		0	0	0	0	0	0	0	0	
Data value	After 8th Shift	0	0	0	0	0	0	0	0	
CRC	After 8th Shift	0	1	0	0	0	0	0	1	0x41

Calculating a CRC by hand is impractical. Most likely a CRC calculation is used to validate a stream of data. The processor which is communicating with the CRC responding device should be able to calculate a CRC. A C-code example of calculating the 8-bit CRC in software follows. The function expects a byte of data with the seed and returns a byte for a CRC result.

```
int calc_crc(int data_byte, int crc)
{
           int bit_mask = 0, carry_check = 0, temp_data = 0;
           temp_data = data_byte;
           for ( bit_mask = 0; bit_mask <= 7; bit_mask ++)
           {
                      data_byte = data_byte ^ crc;
                      crc = crc / 2;
                      temp_data = temp_data / 2;
                      carry_check = data_byte & 0x01;
                      if (carry_check)
                      {
                               crc = crc ^ 0x8C;
                      }
                      data_byte = temp_data;
           }
           return ( crc );
}
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265