Annex to

A 0.1 to 40 Hz SENSING,

CONDITIONING AND RECORDING SYSTEM

Part 3/3 : the recorder

http://www.vlf.it

About CRCs

serge-lacote@orange.fr

Cyclic Redundancy Checking, is a method which consists in sending, when a data block constituting a message M is transmitted, a complementary data block, also referred to as CRC (*Cyclic Redundancy Check)* by language extension and intended to enable the receiver to perform some control of the integrity of the received message.

During transactions with a SD card, it is possible to use CRCs, but the present firmware does not use this possibility, except … when it is required, e.g. when a “CMDxx” command is sent to the card. The message and the CRC take the form of a sequence of bits, generally grouped into bytes and represented in binary or in hexadecimal.

Message M	CRC

Example (arbitrary) : 1000 1110 0001 0100 0100 0000 1100 1001 and 1100 1111

One can be satisfied by the use of the pre-calculated values which can be found in all tutorials about SD cards. The following lines are destined to those who want to know where it comes from and have not completely forgotten the polynomial theory ...

The first idea consists of seeing the different bit sequences as the sequence of the coefficients of the polynomials of the ring $\mathbb{Z}/2\mathbb{Z}$. So, in the above example, the message M is associated with the polynomial $M = X^{31} + X^{27} + X^{26} + X^{25} + ... + X^3 + 1$ and the CRC with the polynomial $R = X^7 + X^6 + X^3 + X^2 + X + 1$.

The second idea consists of the shared knowledge by the transmitter and the receiver of a same polynomial $G \in \mathbb{Z}/2\mathbb{Z}[X]$ named generator polynomial. This polynomial is fixed by agreement and does not depend on the different messages to be transmitted. From there, the control operates as follows :

1. At the emission, the CRC R is generated from M and G according to a determined algorithm.

2. M and R are sent together ; M arrives in the form of $\bar{M}$ and R arrives in the form of $\bar{R}$.

3. Then, simply, the receiver compares $\bar{R}$ with the CRC produced directly from $\bar{M}$ and G. If these two polynomials are identical, it may reasonably be felt – although without absolute certainty – that the transmitting has proceeded without error.

Let's look how these different operations are becoming in practice.

1. The generator polynomial is not left to the programmer's fantasy, but is designed in order to obtain good efficiency, by involving a maximum of bits of M. This efficiency naturally depends on its degree, often 16 or 32, and the list of classical polynomials is provided by literature. In the special case of the dialogue with a SD card, it is a 7th degree polynomial, more specifically the polynomial $G = X^7 + X^3 + 1$ .It is this polynomial that we will use. We must now define the algorithm to obtain the CRC (often named CRC7) R.

2. The CRC R is by agreement the remainder of the Euclidean division of the polynomial $X^7 M$ (more generally $X^{deg(G)} M$)

by G. Why this multiplication by X^7 ? Well, by this tip, we introduce zeros for the coefficients of the terms of indexes 0 up to 6 , so that, if we add to $X^7 M$, a polynomial of degree lower than or equal to 6, this polynomial is easy to identify : it exactly consists of the terms of degree lower than or equal to 6 of the sum :

If $M = m_k X^k + ... + m_1 X^1 + m_0$, then $X^7 M = m_k X^{k+7} + ... + m_1 X^8 + m_0 X^7$ and if $S = s_6 X^6 + ... + s_1 X^1 + s_0$, then

$$X^7 M + S = m_k X^{k+7} + ... + m_1 X^8 + m_0 X^7 + s_6 X^6 + ... + s_1 X^1 + s_0$$

But the theorem of Euclidean division in the ring $\mathbb{Z}/2\mathbb{Z}[X]$ asserts the existence and uniqueness of a pair of polynomials (Q, R), with deg(R) < 7 and as $X^7 M = GQ + R$. As mentioned above, it is this remainder R which is, by agreement, the CRC.

Then, it is clear that the knowledge of $X^7 M$ and R is equivalent to the knowledge of M and G. Thus, to transmit M and G, or $X^7 M$ and R, or even better $X^7 M + R$ is similar from the point of view of the knowledge. Thus, it is $X^7 M + R$, obtained by concatenation, which is sent in a single byte block.

3. In this context, to know if, on arrival, the remainder of the division of $X^7 \bar{M}$ by G is in fact equal to $\bar{R}$, it is sufficient to check that $X^7 \bar{M} + R$ is divisible by G. Indeed, if that is the case, there exists a polynomial $P \in \mathbb{Z}/2\mathbb{Z}[X]$ as we have $X^7 \bar{M} + R = PG$, i. e. $X^7 \bar{M} = PG - \bar{R}$. But in fact the ring $\mathbb{Z}/2\mathbb{Z}$ being of characteristic 2 (i.e. as 1+1 = 0) , it is the same for the ring $\mathbb{Z}/2\mathbb{Z}[X]$,and thus, we have R = - R and therefore $X^7 \bar{M} = PG + \bar{R}$. Last, it is the *uniqueness* of the remainder in the Euclidean division theorem which enables us to conclude[1].

Then, let's see how to concretely calculate a CRC to dialogue with a SD card. For those who would like to do so, it is always possible to manually perform these calculations according to the 2000-year old Euclidean algorithm, remembering that 1 = - 1 here .

---

1 The properties of polynomials on finite fields, as $\mathbb{Z}/2\mathbb{Z}$ are very different from what they are on $\mathbb{R}$ or $\mathbb{C}$. For instance, though the base field only owns two elements (0 and 1) $\mathbb{Z}/2\mathbb{Z}[X]$ is however infinite. Moreover, and it is not the least curiosity, a polynomial may be non-zero and have a zero value in all the points of the field, e.g. $X(X-1) = X^2 - 1$, which has a zero value in 0 and in 1. Conclusion : BEWARE !

However, it is difficult to properly align the numerous partial remainders and it is more convenient to use a computer. The website http://www.ghsi.de/CRC/ contains an applet which runs correctly and the C source code under LGLP licence, which performs the same thing after compilation. It is however necessary to add the compilation directive #include<string.h> to compile it with gcc and modify it a bit to adapt it to the CRC7 (and not CRC8) calculation.

4. In order to send a command CMDxx to the card, we must fill a 6 bytes frame whose last byte consists of a 1 preceded by the seven bits the CRC7, and finally, code the obtained bytes into hexadecimal. For instance, let's show how to calculate this last byte to send the **CMD1** command. Notice that, in fact, it is not exactly the CRC R which is sent, but 2R + 1. This byte is always, though improperly, named “CRC”.

	Command	Argument	Argument	Argument	Argument	<CRC7> 1
General case	01 <CMD number in binary>	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxx1
CMD1 example	01000001	00000000	00000000	00000000	00000000	???????1
In hexadecimal	$41	$00	$00	$00	$00	$????

In this case the M message is 0100000100000000000000000000000000000000. The associated polynomial is $M = X^{38} + X^{31}$ and $X^7 M = X^{45} + X^{38}$. We need to calculate the remainder of the division of $X^{45} + X^{38}$ by G to have the CRC7. If we use the applet indicated above, we have :

* binary encoding of the CRC polynomial (i.e. $G = X^7 + X^3 + 1$) : %10001001

* hexadecimal encoding of M : $41 00 00 00 00

* the applet gives : %1111100 or $7c as the CRC7 value.

* We must attach a 1 at the right of the CRC7, i.e. multiply it par 2 and add 1 ; we obtain %11111001 or $f9. The result we are

looking for is **$f9**.

In the same way, we obtain **$95** for **CMD0**. Outside these two cases, it is useless to calculate the CRC of the CMDxx, since there is no control. We then fill the CRC box with **$ff**.

The end