IoT時代はデータ漏えいの危険がいっぱい…
ワンチップCortex-Mでここまで!

# マイコン内蔵最新セキュリティ機能の研究

## 第3回　エラー検出によく使うCRC

中森 章

## 巡回冗長検査CRCの基礎知識

### ● エラー検出によく使う

巡回冗長検査CRC（Cyclic Redundancy Check）は，任意長のデータ・ストリームを入力とした，関数による誤り検出機構，あるいはその機能をもった関数，固定サイズの出力値のことです．CRCは要するに，ある入力に対する代表値といえます．パリティ，チェックサム，ECC，ハッシュ関数値と同様に，送信時に入力データと一緒に転送し，受信時に転送されてくるデータに誤りがないかのチェックに使用します．

CRCとは，別の言い方をすれば，入力データを加工して結果を生成する一種のハッシュ関数そのものです．ビット化けのエラー検出のほか，乱数などの種（シード）生成などに用いられます．パリティや単純な加算によるチェックサムよりはデータ改ざんの検出強度が高いとされています．

特にCRCの特徴としては，連続して出現するバースト誤りの検出が可能といわれています．

### ● おさらい…多項式

以下に「多項式」という単語がでてきますが，これは数学でいう多項式をより次数の少ない多項式で割ったときの余り（それも多項式になる）を結果とすることに由来しています．

もう少し簡単にいうと，$n$ビットの2進数を$n$次の多項式とみなします．このときビット$n$の値（0か1）を$x^n$の係数とします．つまり，8ビットの「11010011」という2進数があれば，

$$1\times(x^7)+1\times(x^6)+0\times(x^5)+1\times(x^4)+0\times(x^3)+0\times(x^2)+1\times(x^1)+1\times(x^0)$$

つまり，

$$x^7+x^6+x^4+x+1$$

という多項式になります．このように任意の2進数は係数が1か0である多項式とみなせます．$n$ビットのCRCを求めるということは非常に大きな次数の多項式（入力データ）を$n$次の多項式に縮退させることを意味します．この縮退方法で一番単純なのが（$n$+1）次の多項式で割り算を行って余りを求めることなのです．この場合，余りは$n$次の多項式になります．

### ● CRCの基本回路

「多項式同士の割り算って何？」という人もいるかもしれません．それは「組み立て除法」と呼ばれる方法で解けます（高校数学などで習います）．ここで組み立て除法のやり方を説明することはしませんが，多項式の係数が0と1の場合は，シフトと排他的論理和で簡単に組み立て除法を実現できます．これは，論理回路で実現するには実に都合のいい方法です．

実際，KinetisのCRC生成回路は，16ビット/32ビットのシフト・レジスタを用いてCRCコードの生成を行う論理回路です．

CRCというのは2進数のデータ列を$n$ビット（今回の場合は16ビットまたは32ビット）に縮退させる技術ですから，除数の多項式の係数に従って異なる結果のCRCが生成されます（16ビットの場合は$2^{16}$通り）．そこで，除数となる多項式をあらかじめ決めておこうというのが「CRCの標準化」です．

Kinetisのリファレンス・マニュアルにはCRC標準に従っているとあります．除数の多項式として任意のデータを指定できるので，当たり前といえば当たり前です．**表1**に標準的なCRC多項式を示します．どの多項式を使っているかということが暗号でいう鍵の役割を果たします．

KinetisのCRC計算では，オプションで入力データまたは出力データを，ビットごとあるいはバイトごとに入れ替える機能を持っています．あるいは，計算結果を反転することも可能です．

**図1**にCRCの回路ブロックを示します．CRCデータ・レジスタとCRC多項式レジスタに値をセットするとCRC出力レジスタに計算したCRCが格納されます．CRCの計算はCRCデータ・レジスタへのライトごとに実行されます．

CRCは生成論理からみて，CRCの値が変化しないようにデータを改ざんすることは比較的容易です．そ



第1回　通信の認証などによく使う/暗号高速化ユニットCAU（2015年3月号）
第2回　暗号化通信に欠かせない！乱数生成器RNG（2015年4月号）