SHTxx
温湿度センスミッター

アプリケーションノート
CRC

1　　はじめに

CRC チェックサムは、トランスミッション全体にわたって演算されます。CRC ミスマッチが検出されれば、SHTxx をリセット(コマンド「00011110」)して、再測定しなければなりません。

2　　理論

「巡回冗長検査」を意味する CRC は、最も有効なエラー検出方式のひとつで、最小限のハードウェアしか必要としません。
CRC の詳細については、http://www.repairfaq.org/filipg/LINK/F_crc_v3.html 上の、総合ガイド「A painless guide to CRC error detection algorithms」を参照されることをお勧めします。

SHTxx に用いられている多項式は $x^8+x^5+x^4$ で、この多項式により検出されるエラーの種類は次の通りです。:

1. トランスミッション内部のすべての奇数エラー
2. トランスミッション内部のすべてのダブルビット･エラー
3. 8 ビット「ウィンドウ」(1-8 ビットは不正確)内部に含まれることのあるすべてのエラー･クラスタ
4. 大きなエラー･クラスタの大多数

CRC レジスタは、状態レジスタ(「0000’$S_3S_2S_1S_0$」、デフォルト「00000000」)の下位ニブル値で初期化し、これにより肯定応答ビットを伴わないトランスミッション(指令と応答バイト)全体をカバーします。CRC 読み出しの例は、4 頁目のデータシート SHT11 を参照してください。
レシーバは、オリジナルメッセージの先頭部分に接すると同時に演算を実行することができ、続いて演算結果と受け取った CRC-8 とを照合します。CRC ミスマッチが検出されれば、SHTxx をリセット(コマンド「00011110」)して、再測定しなければなりません。

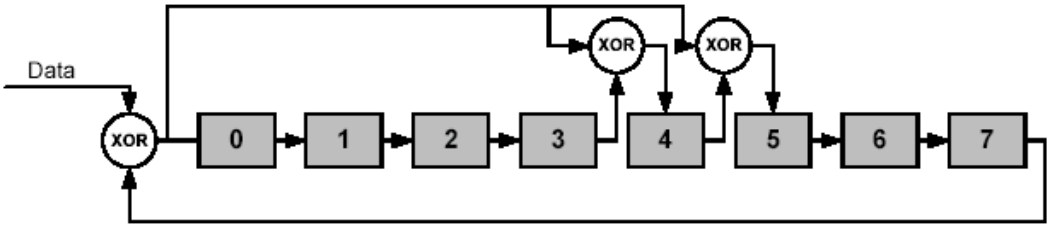
このアプリケーションノートでは、CRC チェックの二つの手法を紹介します。ひとつは「ビット方式」で、ハードウェアや低レベルのインプリメンテーションに適していて、もうひとつの「バイト方式」は、より強力なマイクロコントローラのソリューションに適しています。

2.1　　ビット方式

ビット方式では、レシーバはハードウェアやソフトウェア内の CRC ジェネレータの構成をコピーします。

この演算のアルゴリズムは次のようなものとなります：

1) 状態レジスタの下位ニブルに CRC レジスタを初期化
((S0S1S2S3'0000)に反転)
2) それぞれ(送信と受信)のビットをビット 7 と照合
3) 同一のとき：CRC レジスタをシフト、ビット 0=「0」
その他のとき：CRC レジスタをシフトし、ビット 4 とビット 5 を反転、
ビット 0=「1」(図 1 参照)
4) 新規ビットを受け取り、2)へ
5) SHTxx から取り出した CRC 値は反転（ビット 0=ビット 7、ビット 1=ビット 6…ビット 7=ビット 0)せねばならず、これで最終 CRC 値との照合が可能になる。(2)



2.1.1　ビット方式の例

例 2：RH 測定(データシートでの例)

ｲﾝﾌﾟｯﾄﾋﾞｯﾄ	ﾋﾞｯﾄ 7…ﾋﾞｯﾄ 0	0x	dec	コメント
	0000'0000			開始値 下記(1)参照
0	0000'0000	00	0	コマンド 1st ビット
0	0000'0000	00	0	コマンド 2nd ビット
0	0000'0000	00	0	
0	0000'0000	00	0	
0	0000'0000	00	0	
1	0011'0001			CRC EXOR 多項式
0	0110'0010			
1	1111'0101	F5	245	**コマンド後 CRC**
0	1101'1011			測定 1st バイト(MSB)
0	1000'0111			
0	0011'1111			
0	0111'1110			
1	1100'1101			
0	1010'1011			
0	0110'0111			
1	1111'1111	FF	255	**CRC 値**
0	1100'1111			測定 2nd バイト(LSB)
0	1010'1111			
1	0101'1110			
1	1000'1101			
0	0010'1011			
0	0101'0110			
0	1010'1100			
1	0101'1000	58	88	**最終 CRC 値**

例 1：0x40 を含む状態レジスタの読み出し

ｲﾝﾌﾟｯﾄ ﾋﾞｯﾄ	ﾋﾞｯﾄ 7…ﾋﾞｯﾄ 0	Ox	dec	コメント
	0000’0000			開始値 下記(1)参照
0	0000’0000	00	0	コマンド 1st ビット
0	0000’0000	00	0	コマンド 2nd ビット
0	0000’0000	00	0	
0	0000’0000	00	0	
0	0000’0000	00	0	
1	0011’0001			CRC EXOR 多項式
1	0101’0011			
1	1001’0111	97	151	**コマンド後 CRC**
0	0001’1111			状態レジスタ 1st ビット(MSB)
1	0000’1111			
0	0001’1110			
	0011’1100			
0	0111’1000			
0	1111’0000			
0	1101’0001			
0	1001’0011			
0		93	147	**147 最終 CRC 値**

(1) 下位ニブルのみ、全バイト反転(状態レジスタ＝[$S_7S_6S_5S_4$’$S_3S_2S_1S_0$]－＞開始値＝[$S_0S_1S_2S_3$’0000])
(2) これは、その他の CRC インプリメンテーションとは異なる。

2.2 バイト方式

このインプリメンテーションでは、CRC データは 256 バイト参照用テーブルに格納されます。
次の演算を実行します。

1. 状態レジスタ値の下位ニブル値で CRC レジスタを初期化(反転済み($S_0S_1S_2S_3$’0000))。(デフォルト「00000000」＝0)
2. 先の CRC 値で、それぞれ(送信と受信)のバイトの XOR をとる
   得られた値が、CRC 値を求めるのに必要な新規バイトとなる
3. この値をテーブルの指標として用い、新規の CRC 値を求める
4. 2.)以降を繰り返し、プロセスの終わりまですべてのバイトを送っていく
5. テーブルから引き出した最後のバイトが、最終 CRC 値となる
6. SHTxx から取り出した CRC 値は反転（ビット 0＝ビット 7、ビット 1＝ビット 6…ビット 7＝ビット 0)せねばならず、これで最終 CRC 値との照合が可能になる。(2)

2.2.1　256 バイト CRC 参照用テーブル

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	49	98	83	196	245	166	151	185	136	219	234	125	76	31	46	67	114	33	16	135	182	229	212	250	203	152	169	62	15	92	109
32	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**
134	183	228	213	66	115	32	17	63	14	93	108	251	202	153	168	197	244	167	150	1	48	99	82	124	77	30	47	184	137	218	235
64	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**	**76**	**77**	**78**	**79**	**80**	**81**	**82**	**83**	**84**	**85**	**86**	**87**	**88**	**89**	**90**	**91**	**92**	**93**	**94**	**95**
61	12	95	110	249	200	155	170	132	181	230	215	64	113	34	19	126	79	28	45	186	139	216	233	199	246	165	148	3	50	97	80
96	**97**	**98**	**99**	**100**	**101**	**102**	**103**	**104**	**105**	**106**	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**	**120**	**121**	**122**	**123**	**124**	**125**	**126**	**127**
187	138	217	232	127	78	29	44	2	51	96	81	198	247	164	149	248	201	154	171	60	13	94	111	65	112	35	18	133	180	231	214
128	**129**	**130**	**131**	**132**	**133**	**134**	**135**	**136**	**137**	**138**	**139**	**140**	**141**	**142**	**143**	**144**	**145**	**146**	**147**	**148**	**149**	**150**	**151**	**152**	**153**	**154**	**155**	**156**	**157**	**158**	**159**
122	75	24	41	190	143	220	237	195	242	161	144	7	54	101	84	57	8	91	106	253	204	159	174	128	177	226	211	68	117	38	23
160	**161**	**162**	**163**	**164**	**165**	**166**	**167**	**168**	**169**	**170**	**171**	**172**	**173**	**174**	**175**	**176**	**177**	**178**	**179**	**180**	**181**	**182**	**183**	**184**	**185**	**186**	**187**	**188**	**189**	**190**	**191**
252	205	158	175	56	9	90	107	69	116	39	22	129	176	227	210	191	142	221	236	123	74	25	40	6	55	100	85	194	243	160	145
192	**193**	**194**	**195**	**196**	**197**	**198**	**199**	**200**	**201**	**202**	**203**	**204**	**205**	**206**	**207**	**208**	**209**	**210**	**211**	**212**	**213**	**214**	**215**	**216**	**217**	**218**	**219**	**220**	**221**	**222**	**223**
71	118	37	20	131	178	225	208	254	207	156	173	58	11	88	105	4	53	102	87	192	241	162	147	189	140	223	238	121	72	27	42
224	**225**	**226**	**227**	**228**	**229**	**230**	**231**	**232**	**233**	**234**	**235**	**236**	**237**	**238**	**239**	**240**	**241**	**242**	**243**	**244**	**245**	**246**	**247**	**248**	**249**	**250**	**251**	**252**	**253**	**254**	**255**
193	240	163	146	5	52	103	86	120	73	26	43	188	141	222	239	130	179	224	209	70	119	36	21	59	10	89	104	255	206	157	172

2.2.2　CRC 参照用テーブルのコード例

次の手順で CRC-8 が演算されます。演算結果は可変 CRC に蓄積されます。

```
Var
CRC : Byte;
Procedure calc_CRC(X: Byte);

Const
CRC_Table : Array[0..255] of Byte = (
0,  49, 98, 83, 196,245,166,151,185,136,219,234,125,76, 31, 46, 67, 114,33, 16,
135,182,229,212,250,203,152,169,62, 15, 92, 109,134,183,228,213,66, 115,32, 17,
63, 14, 93, 108,251,202,153,168,197,244,167,150,1,  48, 99, 82, 124,77, 30, 47,
184,137,218,235,61, 12, 95, 110,249,200,155,170,132,181,230,215,64, 113,34, 19,
126,79, 28, 45, 186,139,216,233,199,246,165,148,3,  50, 97, 80, 187,138,217,232,
127,78, 29, 44, 2,  51, 96, 81, 198,247,164,149,248,201,154,171,60, 13, 94, 111,
65, 112,35, 18, 133,180,231,214,122,75, 24, 41, 190,143,220,237,195,242,161,144,
7,  54, 101,84, 57, 8,  91, 106,253,204,159,174,128,177,226,211,68, 117,38, 23,
252,205,158,175,56, 9,  90, 107,69, 116,39, 22, 129,176,227,210,191,142,221,236,
123,74, 25, 40, 6,  55, 100,85, 194,243,160,145,71, 118,37, 20, 131,178,225,208,
254,207,156,173,58, 11, 88, 105,4,  53, 102,87, 192,241,162,147,189,140,223,238,
121,72, 27, 42, 193,240,163,146,5,  52, 103,86, 120,73, 26, 43, 188,141,222,239,
130,179,224,209,70, 119,36, 21, 59, 10, 89, 104,255,206,157,172);

Begin
CRC := CRC_Table[X xor CRC];
End;
```