

```

1  Objectif : valider le Checksum pour la température
2
3  CRC (0X688A) = 0x42 = 00100010
4
5  Polynome 0x31 = 00110001 (d'après datasheet du capteur SHTC3, point 5.10, p.9)
6
7  0X688A = 01101000 10001010
8
9  Afin de valider le Checksum, il faudra retrouver la valeur 0x90
10
11 Initialisation = 0xFF = 11111111 (d'après datasheet du capteur SHTC3, point 5.10,
12 p.9)
13
14 Le calcul du CRC se fait en traitant chaque octet bit par bit.
15 A chaque étape, il faudra faire un décalage d'un bit vers la gauche.
16 Lorsque le bit MSB vaut "1", on applique un XOR avec le polynome (0x31 pour ce
17 capteur).
18 Si le MSB vaut "0", on continue simplement le décalage.
19
20 Le symbole "^" désigne un XOR
21 Rappel : Si les bits sont identiques, cela correspond à un "0"
22 Si les bits sont différents, cela correspond à un "1"
23
24 Pour commencer, il nous faut le point de départ, pour cela, on fait un XOR avec
25 l'initialisation et le premier octet obtenu :
26
27      11111111 (0xFF)
28 ^    01101000 (0X68)
29 -----
30      10010111    -> point de départ
31
32 Comme le MSB contient un "1", on effectue l'opération XOR :
33
34 1.  00101110
35 ^   00110001 (0x31)
36 -----
37      00011111
38
39 2.  00111110    -> pas besoin de XOR, car MSB = 0
40
41 3.  01111100    -> pas besoin de XOR, car MSB = 0
42
43 4.  11111000    -> pas besoin de XOR, car MSB = 0
44
45 5.  11110000    -> XOR, car au point 7, le MSB avait un "1" au MSB
46 ^   00110001 (0x31)
47 -----
48      11000001
49
50 6.  10000010    -> XOR, car au point 7, le MSB avait un "1" au MSB
51 ^   00110001 (0x31)
52 -----
53      10110011
54
55 7.  01100110    -> XOR, car au point 7, le MSB avait un "1" au MSB
56 ^   00110001 (0x31)
57 -----
58      01010111
59
60 8.  10101110
61
62 Pour le calcul du deuxième octet, le processus est le même.
63
64 Calcul pour le 2ème octet (0x8A):
65 On reprend le résultat du premier octet et on fait un XOR avec le deuxième octet pour
66 avoir le point de départ
67
68      10101110
69 ^   10001010 (0X8A)
70 -----
71      00100100    -> point de départ

```

```
70
71 1. 01001000
72
73
74 2. 10010000
75
76
77 3. 00100000
78 ^ 00110001 (0x31)
79 -----
80 00010001
81
82 4. 00100010
83
84 5. 01000100
85
86 6. 10001000
87
88 7. 00010000
89 ^ 00110001 (0x31)
90 -----
91 00100001
92
93 8. 01000010 -> 0x42 -> Checksum OK
94
```