

1 Réalisation d'un récepteur automatique morse (.../10pt)

Vous devez réaliser un récepteur pour code morse! Le morse est un code rudimentaire composé de signaux courts (les points) et de signaux longs (les tirets). Par exemple, la lettre C en morse correspond à -. (long/court/long/court). En morse classique, l'utilisateur appuie sur un interrupteur pour générer les signaux courts et les signaux longs.

Nous considérerons par la suite les durées suivantes pour les différents signaux :

- 0.2 sec pour un signal court.
- 0.6 sec pour un signal long.
- L'espace de temps entre les symboles morse d'une même lettre de l'alphabet est le même que pour un signal court (0.2 sec).
- L'espace de temps entre deux lettres est égal à 3 signaux courts (0.6 sec)
- L'espace de temps entre deux mots est équivalent à 7 signaux courts (1.4 sec).

La figure 1 représente ainsi le code morse du mot ENSSAT, et le détail du signal codant le N est donné par la figure 2.

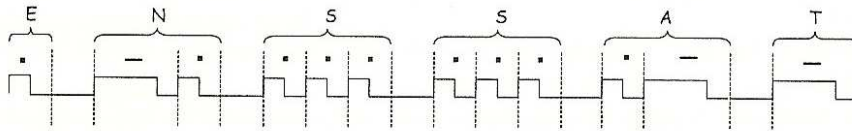


FIG. 1 - Code morse du mot ENSSAT

La correspondance entre des symboles reçus et une lettre de l'alphabet est réalisée à l'aide d'une table présente en mémoire. On suppose que l'on dispose d'une fonction *Association* qui permet donc d'obtenir le message à partir d'un tableau de symboles reçus *chaîne_symb* défini en variable globale avec les conventions suivantes : 0 pour un espace entre lettres, 1 pour un point et 2 pour un tiret. Pour la réception du mot ENSSAT, par exemple, ce tableau est le suivant :

```
unsigned int chaîne_symb[17]={1,0,2,1,0,1,1,1,0,1,1,1,0,1,2,0,2};
```

Q.1.1) Quel mode de fonctionnement du timer vous semble le plus approprié pour recevoir le code morse ?

Q.1.2) Déterminer les registres R_A , R_B ou R_C dont vous avez besoin et faire un schéma simple accompagné d'explications présentant le fonctionnement du timer d'un point de vue logique pour réaliser la fonctionnalité souhaitée.

Q.1.3) En considérant que le microprocesseur fournit une horloge de base MCK de 32 MHz, donner les valeurs des registres pour recevoir un point, ainsi que pour recevoir un tiret.

Q.1.4) Déterminer la configuration du registre TC_CMR (TC Channel Mode Register) en choisissant pour chaque champ présenté ci-dessous, la valeur correcte. Vous devez répondre sur cette feuille en cochant la case correspondante :

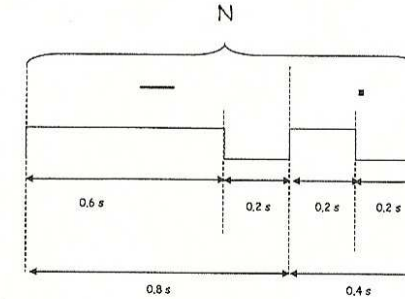


FIG. 2 - Code morse de la lettre N

- Nom : ...
- Prénom : ...
- Horloge en entrée du compteur (TCCLKS : Clock Selection)
 - ☐ MCK/2
 - ☐ MCK/8
 - ☐ MCK/32
 - ☐ MCK/128
 - ☐ MCK/1024
 - ☐ XC0
 - ☐ XC1
 - ☐ XC2
- Polarité de l'horloge, CLKI : Clock Invert
 - ☐ 0 = Counter is incremented on rising edge of the clock.
 - ☐ 1 = Counter is incremented on falling edge of the clock.
- BURST : Burst Signal Selection
 - ☐ The clock is not gated by an external signal.
 - ☐ XC0 is ANDed with the selected clock.
 - ☐ XC1 is ANDed with the selected clock.
 - ☐ XC2 is ANDed with the selected clock.
- LDBSTOP : Counter Clock Stopped with RB Loading
 - ☐ 0 = Counter clock is not stopped when RB loading occurs
 - ☐ 1 = Counter clock is stopped when RB loading occurs
- LDBDIS : Counter Clock Disable with RB Loading
 - ☐ 0 = Counter clock is not disabled when RB loading occurs
 - ☐ 1 = Counter clock is disabled when RB loading occurs
- ETRGEDG : External Trigger Edge Selection
 - ☐ None
 - ☐ Rising edge
 - ☐ Falling edge
 - ☐ Each edge

- ABETRG : TIOA or TIOB External Trigger Selection

- ☐ 0 = TIOB is used as an external trigger
- ☐ 1 = TIOA is used as an external trigger

- CPCTRG : RC Compare Trigger Enable

- ☐ 0 = RC Compare has no effect on the counter and its clock.
- ☐ 1 = RC Compare resets the counter and starts the counter clock.

- WAVE

- ☐ 0 = Waveform Mode is disabled (Capture Mode is enabled).
- ☐ 1 = Waveform Mode is enabled.

- LDRA : RA Loading Selection

- ☐ None
- ☐ Rising edge of TIOA
- ☐ Falling edge of TIOA
- ☐ Each edge of TIOA

- LDRB : RB Loading Selection

- ☐ None
- ☐ Rising edge of TIOA
- ☐ Falling edge of TIOA
- ☐ Each edge of TIOA

Q.1.5 Développez en C la routine d'initialisation du timer `Init_Timer` pour réaliser la fonctionnalité souhaitée (vous utiliserez le timer 0).

Q.1.6 Développez en C la routine d'interruption `Lecture_Symbole` qui permet de déterminer le symbole reçu en fonction des valeurs des registres.

Q.1.7 Ecrivez une routine qui utilise `Lecture_Symbole` pour déterminer les lettres reçues puis le message entier (vous pourrez vous servir également de la routine `Association`).

Q.1.8 On souhaite afficher sur un écran le message reçu en utilisant le périphérique USART du microcontrôleur. Ecrivez la routine d'initialisation `Init_Serial` en tenant compte des spécifications suivantes :

- vitesse de la liaison de 4800 bauds
- pas de parité
- 2 bits de stop.

Ecrivez ensuite la routine `Affichage` qui fonctionne par buffer de 20 caractères pour afficher à l'écran le message reçu.