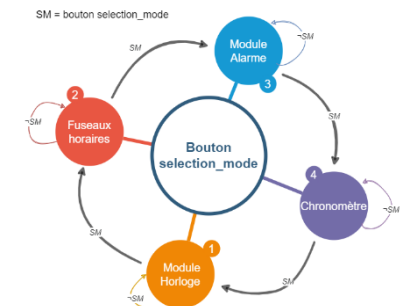
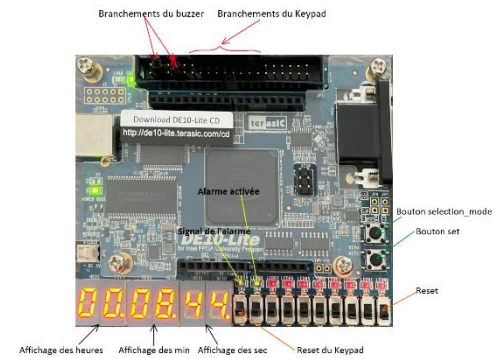


## Description générale

La montre digitale que nous avons réalisée dans le but du projet de fin de semestre est composée de plusieurs fonctionnalités. En effet, elle possède des fonctions usuelles tel que l’affichage de l’heure et un réveil ainsi que des fonctions plus élaborées tel qu’un chronomètre, différents fuseaux horaires, et des animations de LED (clignotements). Toutes ces fonctionnalités sont réalisées au moyen de deux boutons poussoirs, deux dip switches, un buzzer, un 4x4Keypad et 6 affichages 7-segments.

## Mode d’emploi

- Lorsque la montre s’allume il faut régler une heure et activer le switch reset pour redémarrer totalement la montre. Ainsi, le switch reset est à activer à chaque fois que l’on souhaite redémarrer la montre.
- Appuyer sur selection\_mode pour passer d’une fonctionnalité à l’autre selon le schéma ci-contre.



### Module Horloge :

- Le réglage de l’heure s’effectue à l’aide du 4x4Keypad dans l’ordre suivant : secondes-minutes-heures. Il nécessite un reset à chaque appui sur une touche. Il faut donc procéder comme suit : appuyer sur set pour sélectionner le premier chiffre à régler, activer le switch reset du keypad, appuyer sur le chiffre souhaité (à chaque fois que nous réglons un chiffre celui-ci clignote sur l’affichage 7-segments) et enfin appuyer sur set. Le chiffre sélectionné va alors apparaître. Répéter cette opération jusqu’à avoir setter l’heure désirée.  
Attention, pour le deuxième chiffre des secondes, des minutes et des heures il faut entrer un chiffre compris entre 0 et 9. Pour le premier chiffre des secondes et des minutes il faut entrer un chiffre compris entre 0 et 5. Enfin pour le premier chiffre des heures il faut entrer un chiffre compris entre 0 et 2. Dans le cas contraire, il ne s’affiche rien.

### Fuseaux horaires :

- Appuyer sur set pour changer le fuseau horaire sur lequel la montre est calibrée. La montre dispose de 4 fuseaux horaires différents : Oslo (OSL, GMT+0), Sofia (SOA, GMT+1), Asmara (ASA, GMT+2), Bakou (BAO, GMT+3).

### Alarme :

- Appuyer longuement (environ 3sec) sur set pour enclencher l’alarme. Au relâchement du bouton, la LED correspondante s’allumera pour signifier que l’alarme est bien enclenchée.
- Appuyer brièvement sur set pour régler l’heure de l’alarme. Le réglage s’effectue de la même manière que pour le réglage de l’heure de l’horloge.
- Lorsque que l’heure de l’horloge correspond à l’heure réglée pour l’alarme et que celle-ci est activée, la musique d’Harry Potter sonne et la LED correspondante clignote. Il suffit d’appuyer brièvement sur selection\_mode pour l’arrêter. La LED correspondante se désactive par la même occasion.
- Pour désactiver l’alarme qui n’a pas encore sonnée, il faut soit définir une nouvelle heure à l’aide du bouton set, soit procéder à un reset général de la montre.

### Chronomètre :

- Appuyer brièvement sur set pour démarrer le chronomètre.
- Réappuyer sur set pour l’arrêter, réappuyer sur set pour le réenclencher et ainsi de suite.
- Appuyer longuement sur set pour reset le chronomètre.

## Solutions techniques :

*FSM générale :*

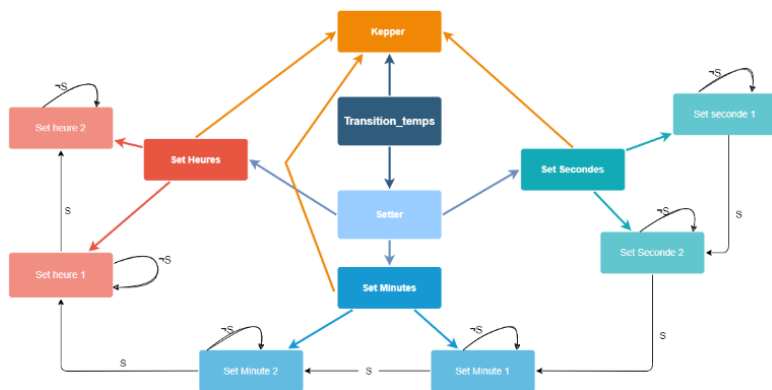


La montre est composée de quatre fonctionnalités implémentées via quatre modules indépendants. La gestion des modules est gérée par *logique\_bouton* qui permet de naviguer entre les différentes fonctionnalités selon un cycle prédéfini. Ceci s'effectue à l'aide du bouton poussoir nommé *selection\_mode*. L'état de la FSM générale est contrôlé par des multiplexeurs/démultiplexeurs qui permettent aux entrées et aux sorties d'être exploitées spécifiquement pour le module de la fonctionnalité sélectionnée et non pour les autres. L'autre bouton poussoir principal de la FSM est le bouton *set* qui permet d'effectuer tous les réglages dans chaque module. Nous avons dû choisir une fréquence de clock élevée pour la carte, de 1MHz afin d'assurer la fluidité du réglage de l'heure à l'aide du keypad et également pour contrôler les fréquences du buzzer. Utiliser une fréquence aussi élevée a demandé certaines précautions pour éviter les problèmes de délais. Enfin, uniquement deux Dip-Switch (un premier permettant un *reset general* et un deuxième permettant le *reset* du keypad) sont nécessaires en plus des deux boutons poussoirs.

### Machines d'états secondaires :

*Module Horloge :*

Le module Horloge est composé d'une FSM nommée *transition\_temps*, deux modules *setter\_horloge* et *kepper\_horloge* et un module fuseaux. *Transition\_temps* se charge de gérer les états du système selon le schéma ci-contre. Le setter est directement relié au keypad et à l'horloge afin de pouvoir directement modifier l'heure. L'heure défile grâce à la variable nommée retenue qui est à la logic-1 permettant ainsi une incrémentation à chaque tic d'horloge. Dans l'état initial, le circuit est à l'état nommé *time\_keeping*, la variable modification est à la logic-0 car aucune modification n'est enclenchée et la variable retenue est toujours à la logic-1. Lorsque *set* est enclenché la variable modification passe à la logic-1 permettant ainsi le réglage de l'heure à l'aide du keypad. Peu importe la modification effectuée, la variable retenue passe à la logic-0 pour "freezer" l'horloge et ainsi permettre le réglage de l'heure. Pendant le réglage, l'affichage 7-segments clignote pour signifier quel chiffre est en train d'être modifié. Ce clignotement est rendu possible grâce à des MUX et à une horloge avec des tics plus rapides que ceux de l'horloge affichant l'heure. Enfin, l'enregistrement des nouvelles valeurs réglées se fait dans le module *setter\_horloge* où se trouvent un registre par chiffre à modifier (6 registres). Ainsi, lorsqu'un chiffre est sélectionné à l'aide du keypad sa valeur est enregistrée dans le registre correspondant et se propage via le module *kepper\_horloge* pour être affiché. Une fois l'heure réglée, *transition\_temps* revient dans l'état *time\_keeping*, la retenue repasse à la logic-1 pour que l'horloge défile à nouveau à partir de l'heure réglée.



*Fuseaux horaires :*

Le module *fuseaux\_horaires* est composé de deux modules principaux : *fuseaux\_horaires\_1* et *fuseaux\_horaires\_2*. Le module *fuseaux\_horaires\_1* est directement incrémenté dans la FSM générale. Il est composé de trois sous-modules nommés première lettre, deuxième lettre et troisième lettre. Ces derniers sont à l'origine de l'affichage des trois

lettres correspondant à une ville. Cet affichage permet à l'utilisateur de connaître le fuseau horaire sur lequel est calibré l'heure de l'horloge et de le changer s'il le souhaite.

#### Module Alarme :

Le *module\_alarme* est décomposé en une FSM qui est la même que pour le module *horloge* nommée *transition\_temps*, un module similaire au module *setter\_horloge* nommé *module1\_alarme* et un module qui permet de faire la différence entre l'appui long ou court du bouton set (nommé *bouton\_appui\_long\_court*). Lorsque le bouton set est pressé longuement l'alarme est enclenchée. Le *module1\_alarme* se charge de comparer l'heure courante reçue en entrée et l'heure fixée pour le réveil. Lorsque les deux heures coïncident et que l'alarme est enclenchée un signal est envoyé au module *alarme* et la musique d'harry potter sonne jusqu'à ce que le bouton selection\_mode soit pressé. Le module *alarme* a été développé en utilisant un compteur logisim pour faire avancer la partition (implémentée grâce à un circuit logique qui prends le numéro de la note et sort le code du signal sonore correspondant et sa durée).

#### Chronomètre :

Le module *chronomètre* réutilise le module *kepper\_horloge* pour permettre de contrôler le déroulement du chronomètre. Il réutilise également le module *bouton\_appui\_long\_court*. Un appui court enclenche ou désenclenche le comptage de l'heure tandis qu'un appui long remet le chronomètre à zéro.

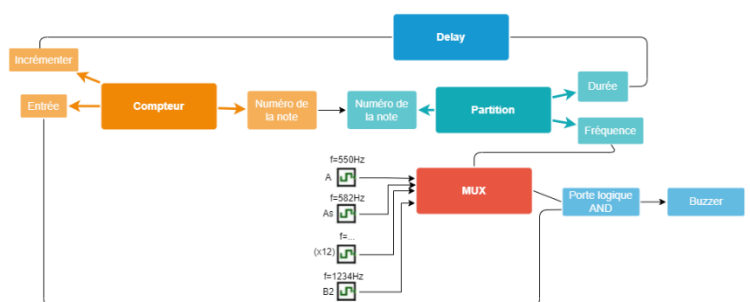
#### Utilisation du périphérique (Keypad) :

La solution à l'utilisation du périphérique a été de se resservir des différents modules mis à notre disposition en complétant le *module\_state1* en créant un bubble counter. Le bubble counter permet de propager un logic-0 de la façon suivante : 0111/1011/1101/1110. Ainsi, lorsqu'on appuie sur une touche du keypad cela fait correspondre le y à la logic-0 avec le x à la logic-0 selon le code du keypad (voir tableau ci-contre) et la touche enfoncée. Le module *displ\_dec\_to\_be\_completed* a également dû être complété. Pour cela, une table de vérité en fonction des x et des y (voir ci-contre) a été créée prenant en entrée le code du keypad qui correspond à une intersection entre un x et un y et en sortie la valeur convertie en termes de a, b, c, d, e, f, g pour permettre l'affichage 7-segments.

Intersection sur le keypad	Chiffre correspondant	Code Keypad	Conversion affichage 7-segments
Y1 - X4	1	0111 1110	0110000
Y1 - X3	2	0111 1101	1101101
Y1 - X2	3	0111 1011	1111001
Y1 - X1	A	0111 0111	1110111
Y2 - X4	4	1011 1110	0110011
Y2 - X3	5	1011 1101	1011011
Y2 - X2	6	1011 1011	1011111
Y2 - X1	8	1011 0111	0011111
Y3 - X4	7	1101 1110	1110000
Y3 - X3	8	1101 1101	1111111
Y3 - X2	9	1101 1011	1111011
Y3 - X1	C	1101 0111	1001110
Y4 - X4	E (*)	1110 1110	1001111
Y4 - X3	0	1110 1101	1111110
Y4 - X2	F (#)	1110 1011	1000111
Y4 - X1	D	1110 0111	0111101

#### Utilisation du buzzer :

Le buzzer est utilisé uniquement pour l'alarme. Il fonctionne de la façon suivante : le bloc délais est un circuit logique composé de DFlipFlop en série qui permettent de compter les temps sortis par la partition avant d'envoyer un signal d'incrémentatation au compteur.



#### Problèmes rencontrés et solutions :

Un des problèmes que nous avons rencontrés a été l'utilisation du keypad, à commencer par comprendre et compléter le bubble counter nécessaire à la bonne utilisation du périphérique. Nous avons également mis longtemps à trouver la bonne fréquence d'utilisation (nous avons finalement opté pour une fréquence élevée de 1MHz). Les fréquences ont été un problème tout au long du projet car il a fallu combiner les fréquences hautes dont nous avons besoin pour l'utilisation du périphérique et du buzzer avec des fréquences plus basses (2Hz) nécessaires au fonctionnement de l'horloge. Ainsi nous avons recalculé les fréquences plus basses par rapport à 1MHz selon la formule suivante :  $N_{tics} = f_{carte} / f_{voulue}$ . Le réglage de l'heure nous a également posé de nombreux problèmes car il était compliqué de mémoriser chaque chiffre rentré à l'aide du keypad. Ce problème a été réglé à l'aide de l'utilisation de registres.