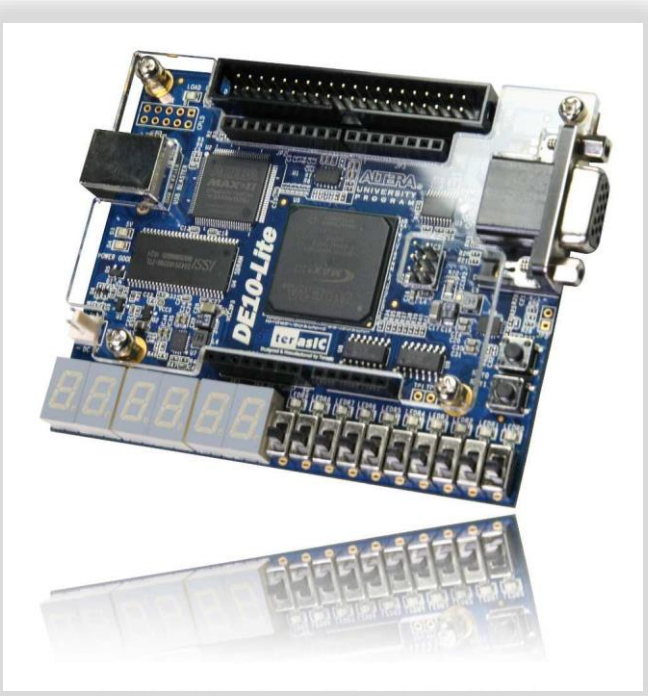
COMPTE-RENDU

PROJET Système embarqué



Réalisé par : MARZAK Ali

SOMMAIRE

I- Choix des interrupteurs

Pour ce projet, j'ai déjà de choisir les mêmes interrupteurs que celui proposé dans les améliorations du sujet du mini projet SNE.

HEURE 24h : SW0 = 0 et SW1 = 0

HEURE 12h : SW0 = 1 et SW1 = 1

ALARME 24h : SW0 = 1 et SW1 = 0

ALARME 12h : SW0 = 0 et SW1 = 1

ALARME activée et LED0 allumé : SW2 = 1

ALARME et LED0 non activée : SW2 = 0

Affichage uniquement : SW3 = 1

Affichage avec réglage : SW3 = 0

En faisant le code, j'ai remarqué qu'en utilisant les hexadécimal, il est assez difficile de gérer les interrupteurs car si on active un autre interrupteur quelconque alors le code se fige puisque dans les conditions il veut que le switch voulu.

On sait que chaque interrupteur correspond à un bit. Donc pour palier le problème précédents, j'ai décidé d'utiliser des maques pour faire sortir le bit voulu. **Et donc chaque interrupteur sera indépendant.**

Exemple pour le switch 3 (SW2) :

int masque3 = 0x4; → masque3 = 00 0000 0100b

SW_value correspond à la valeur sortie par les interrupteurs. Donc si le switch 3 et le switch 7 est activée alors on a SW_value = 0x4 = 00 0100 0100b

```
if ((SW_value & masque3)>>2 == 0x1)
            {
            c = 1;
            }
```

On utilise un AND pour garder à 1 ce qu'on veut garder en précisant le masque.

SW_value & masque3 = 00 0000 0100b

Ensuite on décale pour avoir le bits 1 le plus à gauche pour faciliter l'égalité avec 0x1.

Si dans notre cas la condition est correcte alors on met une variable correspondant au switch à 1.

Bien sûr, j'ai fait cela avec tous les Switches et pour chaque Switch un masque et des variables de sorties différentes.

Dans notre cas, pour caractérise les switches voulu dans les conditions, on a :

SW0 = a

SW1 = b

SW2 = c

SW3 = d

II- HORLOGE

Il s'agit de la fonction principale. Pour ma part, j'ai fait le choix de partir sur une horloge de 24h avec 5 variables :

- sec1 qui correspond aux premiers chiffres des secondes. Donc cette variable va de 0 à 9
- sec2 qui correspond aux seconds chiffres des secondes. Donc cette variable va de 0 à 5
- min1 qui correspond aux premiers chiffres des minutes. Donc cette variable va de 0 à 9
- min2 qui correspond aux seconds chiffres des minutes. Donc cette variable va de 0 à 5
- heureprovi qui correspond à l'heure (24h). Donc cette variable va de 0 à 24.

Les variables des secondes et minutes seront les mêmes que ce soit en mode affichage ou en mode réglage. Contrairement à l'heure qui est provisoire puisqu'on veut faire passer l'horloge du mode 24h au mode 12h et du mode 12h au mode 24h.

III- Mode 12h ou 24h

Pour changer ce mode, j'ai utilisé la variable « heureprovi » vu précédemment, qui est sous 24h, et une fonction « changeh12 » qui permet de changer la variable sous 24h à une variable sous 12h.

On pose une variable « heure » qui sera utilisé comme les secondes et minutes pour l'affichage au 7-segments.

Pour cela, lorsque les switches pour afficher en mode 24h est activée alors on aura :

heure(24h) = heureprovi(24h)

Et lorsque les switches pour afficher en mode 12h est activée alors on aura :

heure(12h) = changeh12(heureprovi(24h))

La fonction changeh12 est simple, on a une variable d'entrée et une variable de sortie et on utilise des « if ».

Exemple pour comprendre avec en entrées a et en sortie b :

If (a = 1) {b = 1 ;}

…

If (a = 12) {b=0 ;}

…

If (a = 17) {b=5 ;}

…

IV- ALARME

Pour l'alarme je n'utilise que l'heure et les minutes donc je trouve qu'il est inutile d'utiliser les secondes. Pour l'affichage ou le réglage de l'alarme, il suffit d'enclencher SW0 pour 24h ou SW1 pour 12h.

Pour l'alarme, j'ai 2 variables principales qui correspondent aux minutes :

- VariableAmin1 qui correspond aux premiers chiffres des minutes. Donc cette variable va de 0 à 9.
- VariableAmin2 qui correspond aux seconds chiffres des minutes. Donc cette variable va de 0 à 5.

Et après chaque réglage :

- Amin1 = VariableAmin1
- Amin2 = VariableAmin2

Ensuite j'ai une variable provisoire pour l'heure « Variableheure » permettant ainsi de pouvoir avoir un mode 24h et 12h.

Lorsqu'on est en mode 24h, on a « Aheure24 = Variableheure ».

Lorsqu'on est en mode 12h, on a « Aheure12 = changeh12(Variableheure) ».

V- Mode Affichage et Réglage

Lorsque le Switches SW3 n'est pas activée alors nous sommes directement en mode réglage. Ici dans mon cas le switches 3 sera dans mes conditions d = 0.

1) Réglage

En mode réglage quand j'appuie sur le bouton poussoir 1, alors on a une incrémentation de 1 sur les minutes.

Bouton poussoir 1 activé ➔ KEY_value & 0x1

Et quand j'appuie sur le bouton poussoir 2, j'ai une incrémentation de 1 sur les heures.

Bouton poussoir 2 activé ➔ KEY_value & 0x2

2) Clignotement

Lorsqu'on est en réglage, j'ai décidé de mettre un mode clignotement.

Cette partie clignotante est fait au même moment où on relie les variables à la carte pour les afficher. On utilise une variable « cligno », lorsque la variable cligno est égale à 0 alors la variable des 7-segments prend les variables des minutes, secondes et heures sinon si la variable cligno est égale à 1 alors la variable des 7-segments prend des variables que j'ai défini et qui donne tous les segments éteints. A chaque boucle du Timer, nous avons un affichage qui alterne entre un écran éteint et l'horloge allumé.

```
if (cligno == 0)
            {
            HEX_bits1 = Hex(sec1);
            HEX_bits2 = Hex(sec2)<<8;
            HEX_bits3 = Hex(min1)<<16;
            HEX_bits4 = Hex(min2)<<24;

            HEX_bits = HEX_bits1 + HEX_bits2 + HEX_bits3 + HEX_bits4;

            heure = changeh12(heureprovi);
            HEX_bitsh = Hexh(heure);

            IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits);
            IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
            cligno =1;
            }
else if (cligno == 1)
            {
            HEX_bits = HEX_bits0;
            HEX_bitsh = HEX_bits00;
            IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits);
            IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                        cligno = 0;
            }
```

VI- SON (Alarme)

Pour l'alarme nous avons une fonction « sonore ». Cette fonction comprend une boucle While avec un Timer que l'on règle pour avoir une fréquence de 500Hz. Donc 50MHz/500Hz = 100kHz.

A l'intérieur de cette fonction, on a une variable « HP_bits » qui sera un chiffre binaire caractérisé par un 0 ou un 1.

Pour entendre l'alarme, il faudra faire varier HP_bits constamment. Pour cela, nous utilisons le caractère « ~ » qui permet de prendre le bits opposé.

On a donc "HP_bits = ~ HP_bits" et à chaque boucle, le bit est différent et fera sonner l'alarme.

Dans cette fonction, on ajoute un « break » lorsque le switch est désactivé pour pouvoir sortir de la boucle et de la fonction.

VII- Communication avec le 7-segments

Pour chacun des variables d'horloges et alarme, les valeurs sont en décimales. Or on communique avec le 7-segments en hexadécimal (plus simple je trouve). De ce fait j'ai créé 3 fonctions permettant de faire le lien.

1 fonction « Hex » pour les secondes et les minutes de l'horloge et les minutes de l'alarme.

1 fonction « Hexh » pour l'heure de l'horloge. Seul car pour l'heure je n'ai pas utilisé 2 variables comme les secondes et les minutes (sec1, sec2, min1, min2).

1 fonction « HexAh » pour l'heure de l'alarme. Sa propre fonction puisqu'on n'utilise pas les mêmes segments que précédemment et donc j'ai remarqué que l'unité et les dizaines étaient inversés.

Ainsi on passe tous les variables en hexadécimal et on les décale pour avoir toutes les valeurs hexadécimales en 1 seul hexadécimal. Dans l'exemple ci-dessous, l'heure est séparée des secondes et minutes car on ne peut pas mettre plus de 4 7-segments.

```
HEX_bits1 = Hex(sec1);
HEX_bits2 = Hex(sec2)<<8;
HEX_bits3 = Hex(min1)<<16;
HEX_bits4 = Hex(min2)<<24;
HEX_bits = HEX_bits1 + HEX_bits2 + HEX_bits3 + HEX_bits4;

heure = changeh12(heureprovi);
HEX_bitsh = Hexh(heure);
```

La variable HEX_bits représente les secondes et minutes de l'horloge en hexadécimal.

La variable HEX_bitsh représente l'heure de l'horloge en hexadécimal.

ANNEXES

I) Choix des interrupteurs

```
SW_value = IORD_ALTERA_AVALON_PIO_DATA(INTERRUPTEURS_BASE);

int masque1 = 0x1;
int masque2 = 0x2;
int masque3 = 0x4;
int masque4 = 0x8;
int masque5 = 0x10;
int masque6 = 0x20;
int masque7 = 0x40;
int masque8 = 0x80;
int masque9 = 0x100;
int masque10 = 0x200;

int a, b, c, d, e, f, g, h, i, j;
a = 0, b = 0, c = 0, d = 0, e = 0, f = 0, g = 0, h = 0, i = 0, j = 0;

if (SW_value & masque1 == 0x1)
              {
              a = 1;
              }
if ((SW_value & masque2)>>1 == 0x1)
              {
              b = 1;
              }
if ((SW_value & masque3)>>2 == 0x1)
              {
              c = 1;
              }
if ((SW_value & masque4)>>3 == 0x1)
              {
              d = 1;
              }
if ((SW_value & masque5)>>4 == 0x1)
              {
              e = 1;
              }
if ((SW_value & masque6)>>5 == 0x1)
              {
              f = 1;
              }
if ((SW_value & masque7)>>6 == 0x1)
              {
              g = 1;
              }
if ((SW_value & masque8)>>7 == 0x1)
              {
              h = 1;
              }
if ((SW_value & masque9)>>8 == 0x1)
              {
              i = 1;
              }
if ((SW_value & masque10)>>9 == 0x1)
              {
              j = 1;
              }
```

II) HORLOGE

```
if (heureprovi < 23)
        {
        if (min2 < 5)
                {
                if (min1 < 9)
                        {

                          if (sec2 < 5)
                                  {
                                  if (sec1 < 9)
                                          {
                                          sec1 = sec1 + 1;
                                          }
                                  else if (sec1 == 9)
                                          {
                                          sec1 = 0;
                                          sec2 = sec2 + 1;
                                          }
                                  }
                          else if (sec2 == 5)
                                  {
                                  if (sec1 < 9)
                                          {
                                          sec1 = sec1 + 1;
                                          }
                                  else if (sec1 == 9)
                                          {
                                          sec1 = 0;
                                          sec2 = 0;
                                          min1 = min1 + 1;
                                          }
                                  }
                        }

                else if (min1 == 9)
                        {
                        if (sec2 < 5)
                                          {
                                          if (sec1 < 9)
                                                  {
                                                  sec1 = sec1 + 1;
                                                  }
                                          else if (sec1 == 9)
                                                  {
                                                  sec1 = 0;
                                                  sec2 = sec2 + 1;
                                                  }
                                          }
                        else if (sec2 == 5)
                                          {
                                          if (sec1 < 9)
                                                  {
                                                  sec1 = sec1 + 1;
                                                  }
                                          else if (sec1 == 9)
                                                  {
                                                  sec1 = 0;
                                                  sec2 = 0;
                                                  min1 = 0;
                                                  min2 = min2 + 1;
                                                  }
                                          }
                        }
                }

        else if (min2 == 5)
                {
                if (min1 < 9)
                                {

                                  if (sec2 < 5)
                                          {
                                          if (sec1 < 9)
                                                  {
                                                  sec1 = sec1 + 1;
                                                  }
                                          else if (sec1 == 9)
                                                  {
                                                  sec1 = 0;
                                                  sec2 = sec2 + 1;
                                                  }
                                          }
```

```
                                  else if (sec2 == 5)
                                          {
                                          if (sec1 < 9)
                                                  {
                                                  sec1 = sec1 + 1;
                                                  }
                                          else if (sec1 == 9)
                                                  {
                                                  sec1 = 0;
                                                  sec2 = 0;
                                                  min1 = min1 + 1;
                                                  }
                                          }
                                  }

                          else if (min1 == 9)
                                  {
                                  if (sec2 < 5)
                                                  {
                                                  if (sec1 < 9)
                                                          {
                                                          sec1 = sec1 + 1;
                                                          }
                                                  else if (sec1 == 9)
                                                          {
                                                          sec1 = 0;
                                                          sec2 = sec2 + 1;
                                                          }
                                                  }
                                  else if (sec2 == 5)
                                                  {
                                                  if (sec1 < 9)
                                                          {
                                                          sec1 = sec1 + 1;
                                                          }
                                                  else if (sec1 == 9)
                                                          {
                                                          sec1 = 0;
                                                          sec2 = 0;
                                                          min1 = 0;
                                                          min2 = 0;
                                                          heureprovi = heureprovi + 1;
                                                          }
                                                  }
                                  }
                  }
          }
else if (heureprovi == 23)
          {
          if (min2 < 5)
                          {
                          if (min1 < 9)
                                  {

                                    if (sec2 < 5)
                                          {
                                          if (sec1 < 9)
                                                  {
                                                  sec1 = sec1 + 1;
                                                  }
                                          else if (sec1 == 9)
                                                  {
                                                  sec1 = 0;
                                                  sec2 = sec2 + 1;
                                                  }
                                          }
                                    else if (sec2 == 5)
                                          {
                                          if (sec1 < 9)
                                                  {
                                                  sec1 = sec1 + 1;
                                                  }
                                          else if (sec1 == 9)
                                                  {
                                                  sec1 = 0;
                                                  sec2 = 0;
                                                  min1 = min1 + 1;
                                                  }
                                          }
                                  }

                          else if (min1 == 9)
                                  {
                                  if (sec2 < 5)
                                                  {
                                                  if (sec1 < 9)
                                                          {
                                                          sec1 = sec1 + 1;
```

```
                                        }
                                    else if (sec1 == 9)
                                        {
                                        sec1 = 0;
                                        sec2 = sec2 + 1;
                                        }
                                    }
                            else if (sec2 == 5)
                                    {
                                    if (sec1 < 9)
                                        {
                                        sec1 = sec1 + 1;
                                        }
                                    else if (sec1 == 9)
                                        {
                                        sec1 = 0;
                                        sec2 = 0;
                                        min1 = 0;
                                        min2 = min2 + 1;
                                        }
                                    }
                            }
                    }

            else if (min2 == 5)
                    {
                    if (min1 < 9)
                                {

                                  if (sec2 < 5)
                                        {
                                        if (sec1 < 9)
                                            {
                                            sec1 = sec1 + 1;
                                            }
                                        else if (sec1 == 9)
                                            {
                                            sec1 = 0;
                                            sec2 = sec2 + 1;
                                            }
                                        }
                                  else if (sec2 == 5)
                                        {
                                        if (sec1 < 9)
                                            {
                                            sec1 = sec1 + 1;
                                            }
                                        else if (sec1 == 9)
                                            {
                                            sec1 = 0;
                                            sec2 = 0;
                                            min1 = min1 + 1;
                                            }
                                        }
                                }

                            else if (min1 == 9)
                                {
                                if (sec2 < 5)
                                                {
                                                if (sec1 < 9)
                                                        {
                                                        sec1 = sec1 + 1;
                                                        }
                                                else if (sec1 == 9)
                                                        {
                                                        sec1 = 0;
                                                        sec2 = sec2 + 1;
                                                        }
                                                }
                                        else if (sec2 == 5)
                                                {
                                                if (sec1 < 9)
                                                        {
                                                        sec1 = sec1 + 1;
                                                        }
                                                else if (sec1 == 9)
                                                        {
                                                        sec1 = 0;
                                                        sec2 = 0;
                                                        min1 = 0;
                                                        min2 = 0;
                                                        heureprovi = 0;
                                                        }
                                                }
                                        }
                    }
}
```

III) Fonction changeh12

```
int changeh12(v5)
      {
      int v6;
      if (v5 == 0)
            {
            v6 =0;
            }
      else if (v5 == 1)
            {
            v6 = 1;
            }
      else if (v5 == 2)
            {
            v6 = 2;
            }
      else if (v5 == 3)
            {
            v6 = 3;
            }
      else if (v5 == 4)
            {
            v6 = 4;
            }
      else if (v5 == 5)
            {
            v6 = 5;
            }
      else if (v5 == 6)
            {
            v6 = 6;
            }
      else if (v5 == 7)
            {
            v6 = 7;
            }
      else if (v5 == 8)
            {
            v6 = 8;
            }
      else if (v5 == 9)
            {
            v6 = 9;
            }
      else if (v5 == 10)
            {
            v6 = 10;
            }
      else if (v5 == 11)
            {
            v6 = 11;
            }
      else if (v5 == 12)
            {
            v6 =0;
            }
      else if (v5 == 13)
            {
            v6 = 1;
            }
```

```
else if (v5 == 14)
      {
      v6 = 2;
      }
else if (v5 == 15)
      {
      v6 = 3;
      }
else if (v5 == 16)
      {
      v6 = 4;
      }
else if (v5 == 17)
      {
      v6 = 5;
      }
else if (v5 == 18)
      {
      v6 = 6;
      }
else if (v5 == 19)
      {
      v6 = 7;
      }
else if (v5 == 20)
      {
      v6 = 8;
      }
else if (v5 == 21)
      {
      v6 = 9;
      }
else if (v5 == 22)
      {
      v6 = 10;
      }
else if (v5 == 23)
      {
      v6 = 11;
      }
else if (v5 == 24)
      {
      v6 = 12;
      }
return v6;
}
```

IV) Mode Affichage et Réglage

1- Mode Réglage

```
KEY_value = IORD_ALTERA_AVALON_PIO_EDGE_CAP(BOUTONS_POUSSOIRS_BASE);
IOWR_ALTERA_AVALON_PIO_EDGE_CAP(BOUTONS_POUSSOIRS_BASE, KEY_value);

if (d == 0) //REGLAGE
        {
        if ((a == 0) & (b == 0))
                {
                                        if (KEY_value & 0x1)
                                        {
                                                if (min2<5)
                                                {
                                                        if (min1<9)
                                                        {
                                                        min1 = min1 + 1;
                                                        }
                                                        else
                                                        {
                                                        min1 = 0;
                                                        min2 = min2 +1;
                                                        }
                                                }
                                                else
                                                {

                                                        if (min1<9)
                                                        {
                                                        min1 = min1 + 1;
                                                        }
                                                        else
                                                        {
                                                        min1 = 0;
                                                        min2 = 0;
                                                        }
                                                }

                                                }
                                        else
                                        {
                                                if (KEY_value & 0x2)
                                                {
                                                        if (heureprovi < 24)
                                                        {
                                                        heureprovi = heureprovi + 1;
                                                        }
                                                        else
                                                        {
                                                        heureprovi = 0;
                                                        }
                                                }
                                        }


                if (cligno==0)
                {
                HEX_bits1 = Hex(sec1);
                HEX_bits2 = Hex(sec2)<<8;
                HEX_bits3 = Hex(min1)<<16;
                HEX_bits4 = Hex(min2)<<24;
                HEX_bits = HEX_bits1 + HEX_bits2 + HEX_bits3 + HEX_bits4;

                heure = heureprovi;
                HEX_bitsh = Hexh(heure);

                IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3 ... HEX0
                IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                cligno=1;
                }
                else if (cligno ==1)
                {
                        HEX_bits = HEX_bits0;
                        HEX_bitsh = HEX_bits00;
                        IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3
... HEX0
                        IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                        cligno = 0;
                }
                }
        else if ((a==1) & (b == 1))
                {

                if (KEY_value & 0x1)
                                        {
                                                if (min2<5)
```

```
                                        {
                                                if (min1<9)
                                                {
                                                min1 = min1 + 1;
                                                }
                                                else
                                                {
                                                min1 = 0;
                                                min2 = min2 +1;
                                                }
                                        }
                                        else
                                        {

                                                if (min1<9)
                                                {
                                                min1 = min1 + 1;
                                                }
                                                else
                                                {
                                                min1 = 0;
                                                min2 = 0;
                                                }
                                        }

                                        }
                                else
                                {
                                        if (KEY_value & 0x2)
                                        {
                                                if (heureprovi < 24)
                                                {
                                                heureprovi = heureprovi + 1;
                                                }
                                                else
                                                {
                                                heureprovi = 0;
                                                }
                                        }
                                }


                        if (cligno == 0)
                        {
                        HEX_bits1 = Hex(sec1);
                        HEX_bits2 = Hex(sec2)<<8;
                        HEX_bits3 = Hex(min1)<<16;
                        HEX_bits4 = Hex(min2)<<24;

                        HEX_bits = HEX_bits1 + HEX_bits2 + HEX_bits3 + HEX_bits4;

                        heure = changeh12(heureprovi);
                        HEX_bitsh = Hexh(heure);

                        IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3 ... HEX0
                        IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                        cligno =1;
                        }
                        else if (cligno == 1)
                        {
                                        HEX_bits = HEX_bits0;
                                        HEX_bitsh = HEX_bits00;
                                        IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern
on HEX3 ... HEX0
                                        IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                                        cligno = 0;
                        }
                        }
                else if ((a == 1) & (b == 0))
                        {

                        if(Aheure12 != 250)
                        {
                        Aheure24 = Variableheure;
                        }

                        if (KEY_value & 0x1)
                                {

                                if (VariableAmin2<5)
                                {
                                        if (VariableAmin1<9)
                                        {
                                                VariableAmin1 =VariableAmin1 +1;
                                        }
                                        else
                                        {
```

```
                                        VariableAmin1 = 0;
                                        VariableAmin2 = VariableAmin2 + 1;
                                }
                        }
                        else
                        {
                                if (VariableAmin1<9)
                                {
                                        VariableAmin1 =VariableAmin1 +1;
                                }
                                else
                                {
                                        VariableAmin1 = 0;
                                        VariableAmin2 = 0;
                                }
                        }

                        Amin1 = VariableAmin1;
                        Amin2 = VariableAmin2;

                        }
                else
                        {
                        if (KEY_value & 0x2)
                        {
                                if (Variableheure < 23)
                                {
                                Variableheure = Variableheure + 1;
                                }
                                else
                                {
                                        Variableheure = 0;
                                }
                                Aheure24 = Variableheure;

                        }
                        }

                if (cligno == 0)
                {
                        HEX_bitsA1 = Hex(Amin1);
                        HEX_bitsA2 = Hex(Amin2)<<8;
                        HEX_bitsAh = HexAh(Aheure24)<<16;

                        HEX_bits = HEX_bitsAh + HEX_bitsA1 + HEX_bitsA2;


                        IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3
... HEX0
                        IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                        cligno = 1;
                }
                else if (cligno == 1)
                {


                        HEX_bits = HEX_bitsA; // CLIGNOTE AVEC 0
                        HEX_bitsh = HEX_bitshA; // MET A


                        IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3
... HEX0
                        IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                        cligno = 0;
                }
                }
        else if ((a == 0) & (b == 1))
                {

                if(Aheure24 != 250)
                {
                Aheure12 = changeh12(Variableheure);
                }



                        if (KEY_value & 0x1)
                                {
                                if (VariableAmin2<5)
                                        {
                                                if (VariableAmin1<9)
                                                {
                                                        VariableAmin1 = VariableAmin1 +1;
                                                }
                                                else
                                                {
                                                        VariableAmin1 = 0;
                                                        VariableAmin2 = VariableAmin2 + 1;
```

```
                                        }
                                }
                        else
                                {
                                        if (VariableAmin1<9)
                                        {
                                                VariableAmin1 =VariableAmin1 +1;
                                        }
                                        else
                                        {
                                                VariableAmin1 = 0;
                                                VariableAmin2 = 0;
                                        }
                                }

                                Amin1 = VariableAmin1;
                                Amin2 = VariableAmin2;

                        Amin1 = VariableAmin1;
                        Amin2 = VariableAmin2;
                        }
                else
                        {
                        if (KEY_value & 0x2)
                        {
                                if (Variableheure < 23)
                                {
                                        Variableheure = Variableheure + 1;
                                }
                                else
                                {
                                        Variableheure = 0;
                                }

                                Aheure12 = changeh12(Variableheure);

                        }
                        }



        if (cligno ==0)
        {
        HEX_bitsA1 = Hex(Amin1);
        HEX_bitsA2 = Hex(Amin2)<<8;

        HEX_bitsAh = HexAh(Aheure12)<<16;

        HEX_bits = HEX_bitsA1 + HEX_bitsA2 + HEX_bitsAh;


        IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3 ... HEX0
        IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
        cligno = 1;
        }
        else if (cligno == 1)
                {
                HEX_bitsh = HEX_bitshA; // MET A
                HEX_bits = HEX_bitsA; // CLIGNOTE AVEC 0

                IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display pattern on HEX3
... HEX0

                cligno = 0;
                }
        }
}
```

2- Mode Affichage

```
else  //AFFICHAGE
       {
       if ((a == 0) & (b == 0))
                      {
                      HEX_bits1 = Hex(sec1);
                      HEX_bits2 = Hex(sec2)<<8;
                      HEX_bits3 = Hex(min1)<<16;
                      HEX_bits4 = Hex(min2)<<24;
                      HEX_bits = HEX_bits1 + HEX_bits2 + HEX_bits3 + HEX_bits4;

                      heure = heureprovi;
                      HEX_bitsh = Hexh(heure);

                      IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits);
                      IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                      }
              else if ((a==1) & (b == 1))
                      {
                      HEX_bits1 = Hex(sec1);
                      HEX_bits2 = Hex(sec2)<<8;
                      HEX_bits3 = Hex(min1)<<16;
                      HEX_bits4 = Hex(min2)<<24;

                      HEX_bits = HEX_bits1 + HEX_bits2 + HEX_bits3 + HEX_bits4;

                      heure = changeh12(heureprovi);
                      HEX_bitsh = Hexh(heure);

                      IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display
pattern on HEX3 ... HEX0
                      IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                      }
              else if ((a == 1) & (b == 0))
                      {
                      HEX_bitsA1 = Hex(Amin1);
                      HEX_bitsA2 = Hex(Amin2)<<8;
                      HEX_bitsAh = HexAh(Aheure24)<<16;

                      HEX_bits = HEX_bitsA1 + HEX_bitsA2 + HEX_bitsAh;
                      HEX_bitsh = HEX_bitshA;

                      IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display
pattern on HEX3 ... HEX0
                      IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);

                      }
              else if ((a == 0) & (b == 1))
                      {
                      HEX_bitsA1 = Hex(Amin1);
                      HEX_bitsA2 = Hex(Amin2)<<8;
                      HEX_bitsAh = HexAh(Aheure12)<<16;

                      HEX_bits = HEX_bitsA1 + HEX_bitsA2 + HEX_bitsAh;
                      HEX_bitsh = HEX_bitshA;

                      IOWR_ALTERA_AVALON_PIO_DATA(HEX3_HEX0_BASE, HEX_bits); // display
pattern on HEX3 ... HEX0
                      IOWR_ALTERA_AVALON_PIO_DATA(HEX5_HEX4_BASE, HEX_bitsh);
                      }


       }
```