Université Sultan Moulay Slimane

Faculté des sciences et techniques

Béni-Mellal

Département génie électrique : systèmes embarqué et informatique industrielle

***Réalisation d’une horloge à l’aide d’un microcontrôleur PIC16F84A***

***Réalisée par :***

EL AABD Mohcine

ELFATAOUY Abdelkodouss

AHDAOUI Anass

***Encadré par :***

Mr. A. KLILOU

Année universitaire : 2022-2023

Table de matière

[Table de matière 2](#_Toc125461478)

[I. Introduction 3](#_Toc125461479)

[II. Listes du matériel utilisé 3](#_Toc125461480)

[*1.* *PIC16F84A* 3](#_Toc125461481)

[a. *TIMER dans PIC16F84A* 4](#_Toc125461482)

[*2.* *Afficheur 4-DIG 7-SEG* 4](#_Toc125461483)

[*3.* *74LS47 (décodeur BCD / 7 segments)* 5](#_Toc125461484)

[4. *BOtton* 5](#_Toc125461485)

[*5.* *Resistance* 5](#_Toc125461486)

[*6.* *Quartz* 5](#_Toc125461487)

[III. Test du composant 6](#_Toc125461488)

[1. *Liaison du décodeur avec l’afficheur* 6](#_Toc125461489)

[*2.* *Test du PIC16F84A* 6](#_Toc125461490)

[3. *Test de l’afficheur avec décodeur et le pic* 7](#_Toc125461491)

[IV. Projet finale 8](#_Toc125461492)

[1. Schéma sur ISIS 8](#_Toc125461493)

[2. Code on micoC 8](#_Toc125461494)

[3. Test pratiquement 8](#_Toc125461495)

# **Introduction**

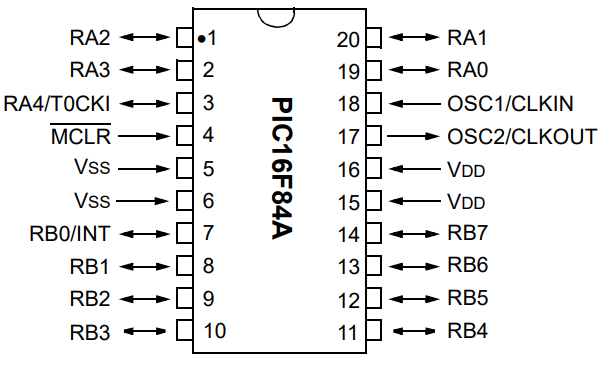
Dans toutes les grandeurs physiques, l'heure est certainement parmi les grandeurs dont la mesure est la plus fréquente.

Ce travail consiste à étudier et réaliser une horloge et un chronomètre avec un afficheur 7segment multiplexé commune anode a 4 digits, contrôlé par un microcontrôleur PIC16F84A.

# **Listes du matériel utilisé**

## *PIC16F84A*

Le PIC16F84A est un microcontrôleur développé par Microchip Technology. Il s'agit d'un petit appareil de 8 bits qui peut être programmé pour effectuer une variété de tâches. Il comporte 14 broches d'entrée/sortie numériques, un processeur 8 bits et 256 octets de mémoire flash pour le stockage du code. Il dispose également d'une série de fonctions périphériques analogiques et numériques, telles que des temporisateurs (timer) et des interfaces de communication. Le PIC16F84A est couramment utilisé dans une variété de projets électroniques et de systèmes embarqués.



Brochage du PIC16F84A

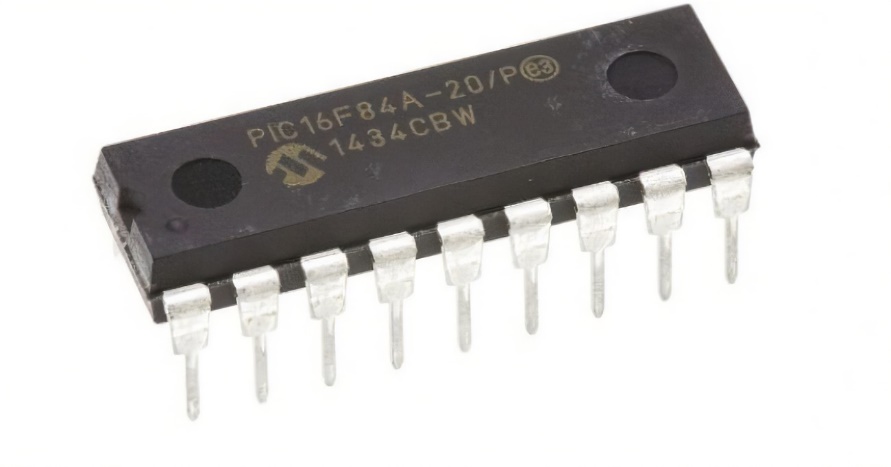


Photo du PIC16F84A

* Le PIC 16F84 s’alimente par une tension continue comprise entre 2 et 5V pour les versions commerciales et industrielles, par les broches VSS (pin 5 ou 6) et VDD (pin 16 ou 15).
* Il possède 2 pins pour l’horloge notées OSC1 et OSC2 : on placera le quartz entre ces 2 pattes, ou la sortie d’un oscillateur sur l’entre OSC1/CLKIN.
* Nous trouvons 2 ports A et B : le port B est composé de 8 entrées sorties (IO paramétrables par logiciel) et un port A composé de 5 entrées sorties uniquement. En tout le nombre d’entrées sorties est de 5+8=13. Chacune de ces entrées sorties peut alimenter une LED et le courant maximal disponible est donné inférieur à 25mA.
* Il possède une zone de mémoire flash pour stocker le programme (1024 mots), une zone de ram (68 octets) pour les variables et une zone d’EEprom (64 octets) pour stocker des données non volatiles qui seront donc conservées après une coupure d’alimentation.

### *TIMER dans PIC16F84A*

* Le microcontrôleur PIC16F84A dispose d'un module timer/compteur intégré qui peut être utilisé pour mesurer des intervalles de temps, générer des délais précis ou compter des événements. Le timer est implémenté comme un registre qui peut être incrémenté par l'horloge d'instruction interne ou un signal externe. La timer peut être configurée pour fonctionner dans plusieurs modes différents, notamment :
* Le mode timer : Dans ce mode, le timer s'incrémente à un rythme fixe et une interruption peut être générée lorsqu'il atteint une certaine valeur.
* Mode compteur : Dans ce mode, la timer s'incrémente en réponse à des impulsions externes et peut être utilisée pour compter des événements ou mesurer des fréquences.
* Le timer peut être configuré pour utiliser différentes sources d'horloge, telles que l'horloge d'instruction interne ou un signal externe, et peut être mis à l'échelle pour diviser la fréquence de l'horloge par un facteur de 1, 4 ou 16. Cela permet au timer de mesurer une large gamme d'intervalles de temps avec une haute résolution. Le module timer/counter est l'un des périphériques les plus importants du PIC16F84A et il est largement utilisé dans de nombreuses applications.

## *Afficheur 4-DIG 7-SEG*

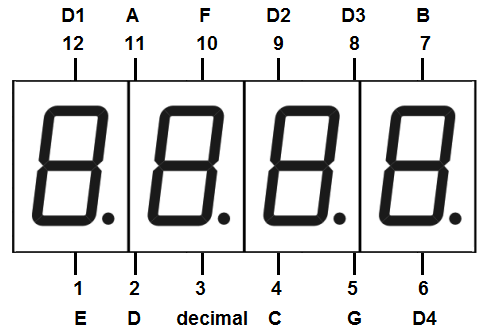
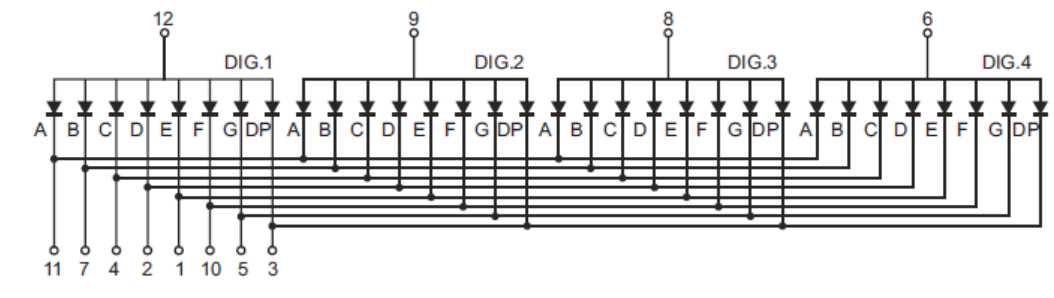
Les afficheurs 7 segments consiste en une série de LED placé pour former des chiffres. Ces LEDS sont gérer via les broches de l’afficheur (1-12).

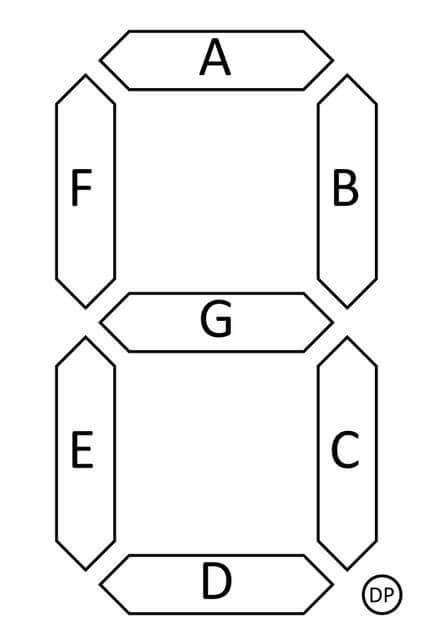


Photo de l’afficheur Brochage de l’afficheur

Les groupements de LED ou digits sont activés via les broches D1, D2, D3, D4 et les leds sont allumées via les broches A, B, C, D, E, F, G et DP pour le point décimal.

Il existe deux types d’afficheurs, avec cathode commune ou avec anode commune, Pour la série anode, il faut connecté le commun à 5V pour activer le digit et passer la broche à l’état BAS pour allumer une LED.

* Pour notre travail on va travailler avec le type d’anode commune de référence HS410361K-32.



Configuration interne du l’afficheur

## *74LS47 (décodeur BCD / 7 segments)*

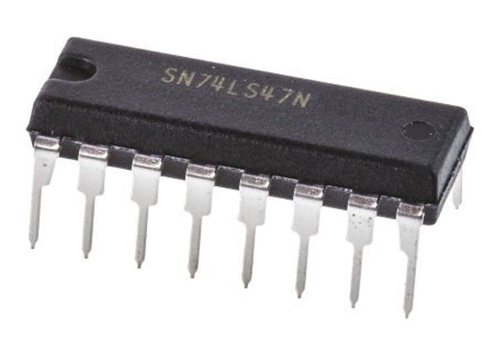
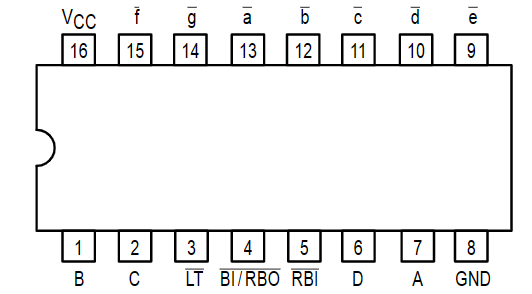
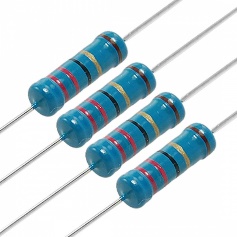
Le 74LS47 est un circuit intégré logique numérique couramment utilisé comme décodeur/driver BCD à 7 segments. Il fait partie de la famille des circuits intégrés 74LS, qui se caractérisent par leur faible consommation d'énergie et leur grande immunité au bruit. Le 74LS47 prend une entrée binaire de 4 bits et la convertit en un modèle de signaux qui peut être utilisé pour piloter un afficheur LED à 7 segments. Il peut également être utilisé pour piloter d'autres types d'écrans tels que les écrans à incandescence ou fluorescents. Le "LS" dans le nom indique que ce circuit intégré est conçu pour fonctionner avec une faible puissance et à des vitesses élevées.

Photo du décodeur brochage du décodeur

## *BOtton*

Un bouton est un élément graphique d'une interface utilisateur qui, lorsqu'il est cliqué ou pressé, déclenche un événement ou une action. Les boutons sont généralement utilisés pour déclencher une action, comme l'envoi d'un formulaire, la navigation vers une nouvelle page ou l'ouverture d'un menu.

## *Resistance*

Une résistance est une mesure de l'opposition au passage du courant dans un circuit électrique. Elle est mesurée en unités d'ohms (Ω). Une résistance plus élevée signifie qu'il est plus difficile pour le courant de circuler dans le circuit, tandis qu'une résistance plus faible signifie qu'il est plus facile pour le courant de circuler. La résistance d'un circuit peut être affectée par les matériaux utilisés dans le circuit, la température et les dimensions physiques des composants du circuit.

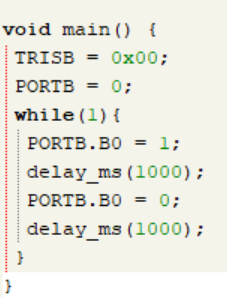
## *Quartz*

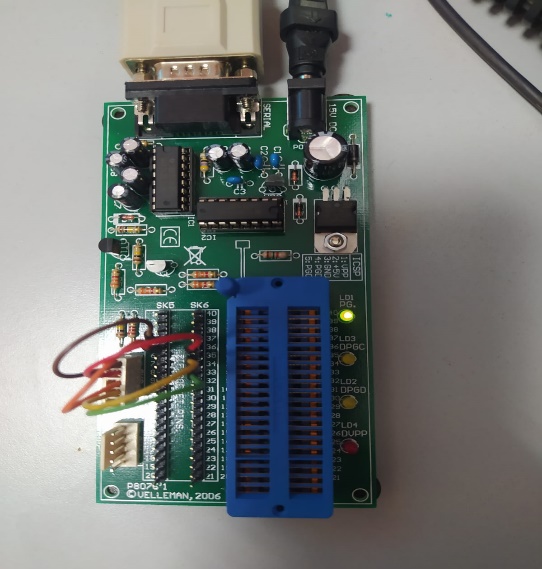
Une électronique à quartz est un type de circuit oscillateur qui utilise la fréquence de résonance naturelle d'un cristal de quartz pour générer un signal de synchronisation précis. Ce signal est ensuite utilisé pour contrôler la fréquence d'autres appareils électroniques, tels que les radios, les montres et les horloges. Le cristal de quartz est coupé et façonné à une fréquence spécifique et, lorsqu'un courant électrique lui est appliqué, il vibre à cette fréquence exacte, fournissant ainsi une base de temps très stable et précise pour les appareils électroniques.

# **Test du composant**

## *Liaison du décodeur avec l’afficheur*

## *Test du PIC16F84A*

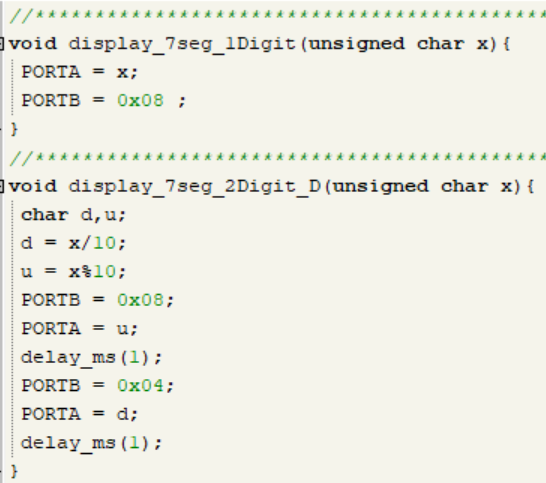
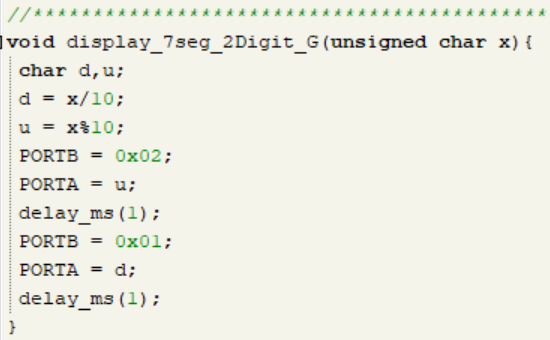
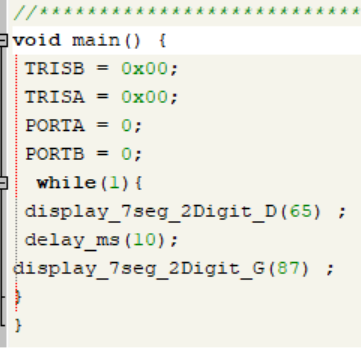


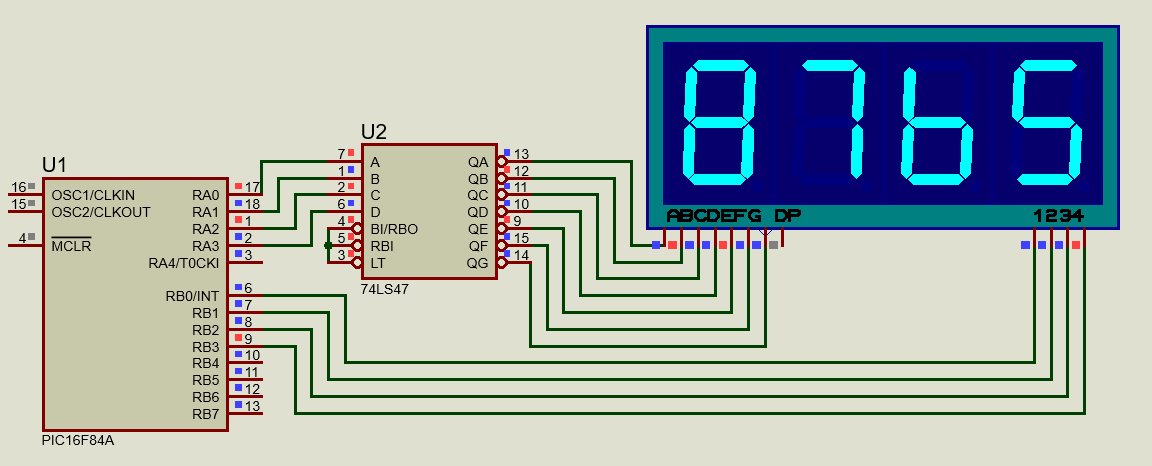
 Code on microC schéma réalisé

Test pratiquement programmateur du pic

Utilisée pour téléverser le code de microC on PIC

## *Test de l’afficheur avec décodeur et le pic*



 Code on microC

 Test sur ISIS

Test pratiquement

# **Projet finale**

## Code on micoC

 unsigned char minut=59;

 unsigned char secondes=00;

 unsigned char i = 0, Cg=00,Cd=00,k,l,t;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

 void blink\_parameter() {

  unsigned char j = 0;

  while(j < 10) {

   j++;

   delay\_ms(25);

  }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void display\_7seg\_1Digit(unsigned char x){

 PORTB = x;

 PORTA = 0x08 ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void display\_7seg\_2Digit\_D(unsigned char x){

 char d,u;

 d = x/10;

 u = x%10;

 PORTA = 0x08;

 PORTB = u;

 delay\_ms(1);

 PORTA = 0x04;

 PORTB = d;

 delay\_ms(1);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void display\_7seg\_2Digit\_G(unsigned char x){

 char d,u;

 d = x/10;

 u = x%10;

 PORTA = 0x02;

 PORTB = u;

 delay\_ms(1);

 PORTA = 0x01;

 PORTB = d;

 delay\_ms(1);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void horloge(void){

  display\_7seg\_2Digit\_D(secondes);

  display\_7seg\_2Digit\_G(minut);

   if(secondes>=60){

    minut++;

    secondes=0;

   }

   if(minut>59){

    minut=0;

    secondes=0;

   }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void reglage(void){

unsigned char secondes1 = secondes;

unsigned char minut1 = minut;

 delay\_ms(250);

 while(PORTB.B7 == 1){

  display\_7seg\_2Digit\_D(secondes1);

  blink\_parameter();

  if(PORTB.B5 == 0){

   delay\_ms(250);

   secondes1++;

  }

  if(secondes1 >= 60){

    secondes1 = 0;

  }

 }

 delay\_ms(250);

 while(PORTB.B7 == 1){

  display\_7seg\_2Digit\_G(minut1);

  blink\_parameter();

  if(PORTB.B5==0){

   delay\_ms(250);

   minut1++;

  }

  if(minut1 >= 60){

   minut1=0;

  }

 }

 delay\_ms(250);

 secondes = secondes1;

 minut = minut1;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void chronometre(){

  delay\_ms(250);

  Cg = 00;

  Cd = 00;

  k=0,l=0;

  while(PORTB.B7 == 1){

  l=0;

 while(PORTB.B4 == 1 && k == 0){

  display\_7seg\_2Digit\_D(Cg);

  display\_7seg\_2Digit\_G(Cd);

  if(PORTB.B7 == 0){delay\_ms(250);k=1; break;}

  if(PORTB.B5 == 0){delay\_ms(250); Cg = 00; Cd = 00; l=1; break;

    }

 }

   delay\_ms(250);

  while(PORTB.B4 == 1 && k == 0 && l == 0 ){

  display\_7seg\_2Digit\_D(Cg);

  display\_7seg\_2Digit\_G(Cd);

     if(i>30){

     i=0;

     if(Cg <= 59){

      Cg++;

     if(Cg>=60){

      Cd++;

      Cg=0;

     }

     if(Cd>59){

      Cd=0;

     Cg=0;

    }}}

  else i++;

  if(PORTB.B7 == 0){delay\_ms(250);k=1; break;}

  if(PORTB.B5 == 0){delay\_ms(250);Cg = 00;Cd = 00; break;}

 }

  if(k == 1){ break;}

  delay\_ms(250);

 }

  delay\_ms(250);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void interrupt(){

 if(INTCON.TMR0IF == 1){

  t++;

  if(t==61){secondes++;  t=0;}

 }

  INTCON.TMR0IF = 0;}

void main() {

 OPTION\_reg = 0b00010101;

 INTCON = 0b10100100;

 TMR0 = 0;

 t = 0;

 TRISA = 0x00;

 TRISB = 0xB0;

 PORTA = 0;

 PORTB = 0;

 while(1){

  if(PORTB.B7 == 0){

   reglage(void);

  }

  if(PORTB.B4 == 0){

   chronometre();

  }

  horloge();

 }

}

## Schéma sur ISIS

Figure 1: test etbsimulation de la fonction horloge() sur ISIS

## Test pratiquement