

# Rapport

## Mini-projet

Un système d'aide aux malades d'Alzheimer

09/05/2022

*Projet réalisé par :*

- Ben Belgacem Mohamed Farouk
- Zairi Nour
- Ben Maati Dorra
- Omri Ghada
- Lengue Mbangue Joyce Maeva
- Belhaj Mohamed Ali

*Projet encadré par :*

- Madame Rejeb Nejla

# Sommaire :

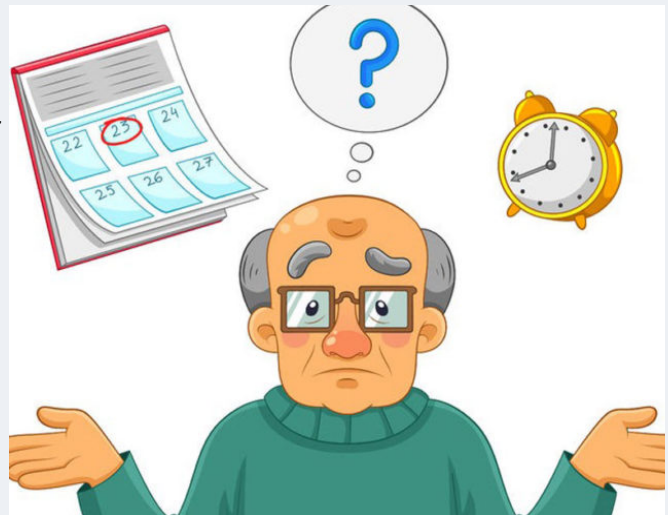
1. Introduction
2. Besoin et objectif du projet
3. Outils de programmation et conception
4. Composants
5. Organigramme
6. Montage isis
7. Code source
8. Mode de fonctionnement

## Introduction

Dans le cadre de notre seconde année du cycle préparatoire intégrée en Informatique à ESPRIT, il nous est proposé un projet de 6 semaines nous permettant de mettre en pratique nos connaissances en architecture en microcontrôleur.

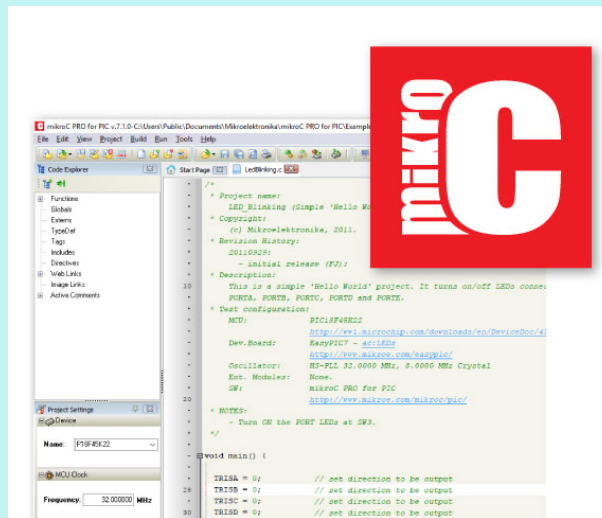
## Besoin et objectif du projet

*Les personnes atteints d'alzheimer ont souvent tendance à oublier l'heure de prise de leurs médicaments, dans ce but notre projet a mis en place un système qui va leur permettre de pouvoir prendre leur médicament à temps, de leur faciliter la tâche pour demander de l'aide à l'aide d'un buzzer, ouvrir et fermer les stores à l'aide d'un moteur.*



# Outils de programmation et conception

On a utilisé comme compilateur le mikroC pro form Pic, destiné aux circuits **PIC** de Microchip. Il possède un IDE intuitif, un compilateur puissant avec des optimisations évoluées, de nombreuses bibliothèques de matériel et de logiciel, et des outils supplémentaires pour le développement de code

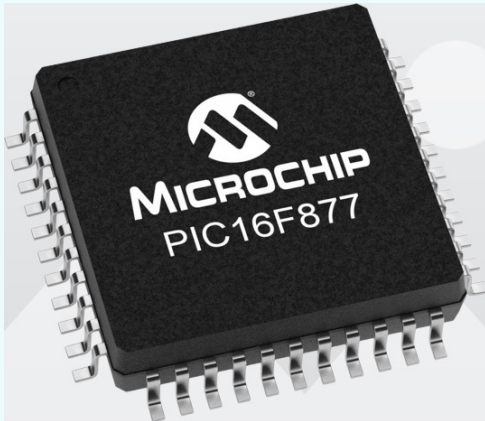


On a utilisé aussi **ISIS** qui est la composante de **Proteus** qui permet la création de schémas et la simulation électrique. La grande force de **ISIS** est de pouvoir simuler le comportement d'un microcontrôleur (PIC, Atmel, 8051, ARM, HC11...) et de son interaction avec les composants qui l'entourent.



## Les composants

\*la pièce maitresse c'est notre microcontrôleur



Pic16F877

\*Un afficheur LCD :



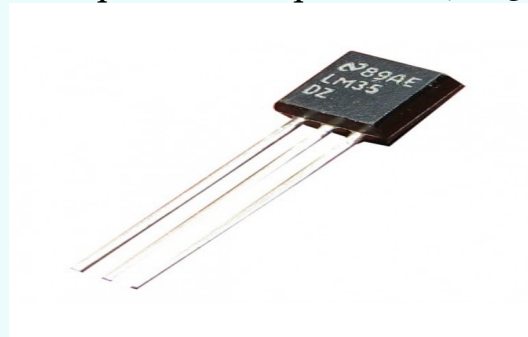
\*Un moteur double sens :



\*Un buzzer ou un sounder

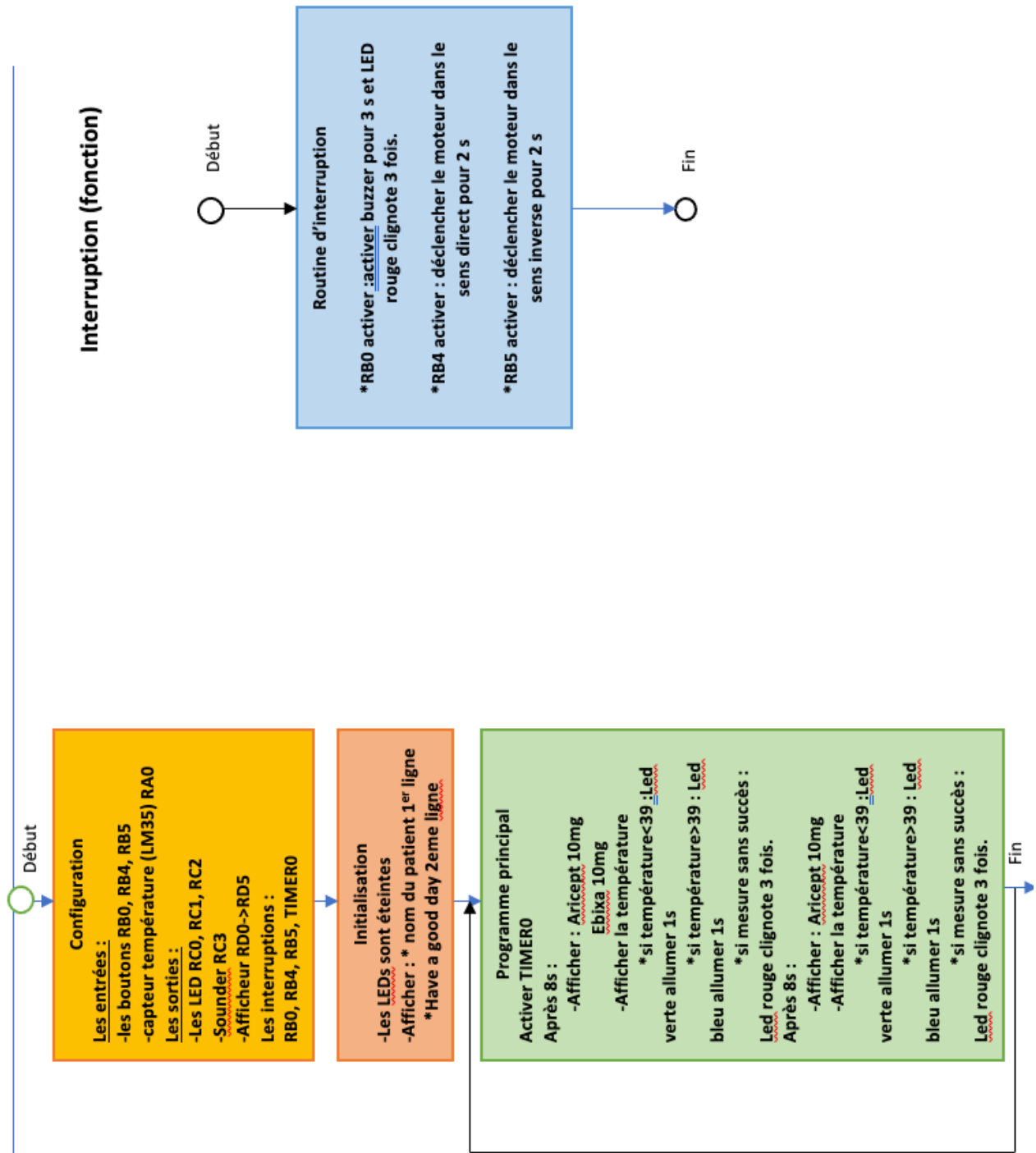


\*Un capteur de temperature (LM35)

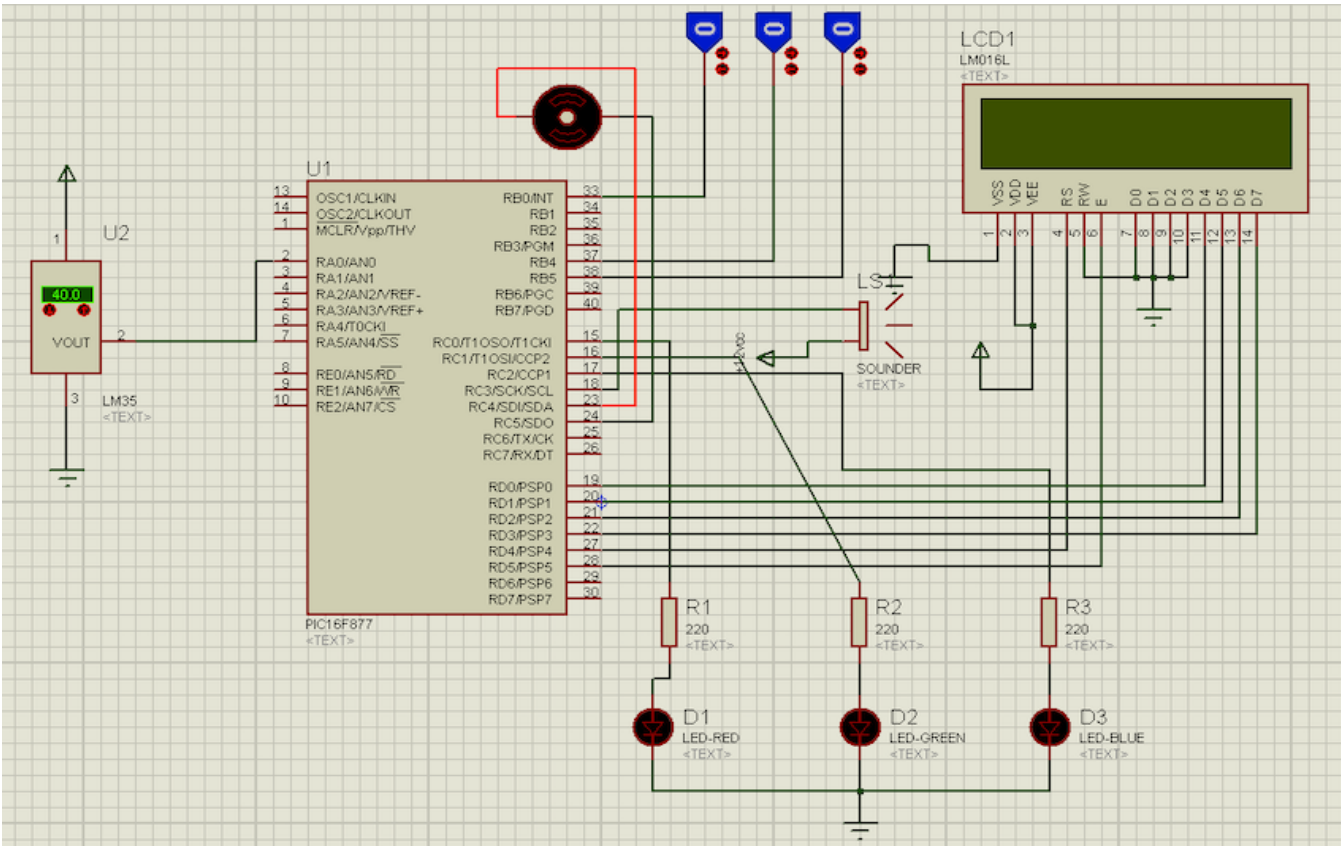


\*Des LEDs (rouge, verte et bleu) :





# Notre montage Isis :



## Notre code source :

```
• // LCD module connections
• sbit LCD_RS at RD4_bit;
• sbit LCD_EN at RD5_bit;
• sbit LCD_D4 at RD0_bit;
• sbit LCD_D5 at RD1_bit;
• sbit LCD_D6 at RD2_bit;
• sbit LCD_D7 at RD3_bit;
8
• sbit LCD_RS_Direction at TRISD4_bit;
10 sbit LCD_EN_Direction at TRISD5_bit;
• sbit LCD_D4_Direction at TRISD0_bit;
• sbit LCD_D5_Direction at TRISD1_bit;
• sbit LCD_D6_Direction at TRISD2_bit;
• sbit LCD_D7_Direction at TRISD3_bit;
• // End LCD module connections
•
• int i;
• char nb;
• bit constante;
20 int value;
• char value1[15];
• char med1[15] ;
• char med2[15] ;
•
- void Move_Delay() { // Function used for text moving
•   Delay_ms(500); // You can change the moving speed here
• }
• void interrupt()
• {if(intcon.intf==1)
30 {
•   Sound_Play(800, 1000);
•   Delay_ms(3000);
•
•   portc.b0=1; //led rouge
•   delay_ms(600);
•   portc.b0=0;
•   delay_ms(600);
•
40   portc.b0=1;
•   delay_ms(600);
•   portc.b0=0;
•   delay_ms(600);
•
•   portc.b0=1;
•   delay_ms(600);
```



```

- 1 if(PORTB.B4==1) {
.
.         portc.b4=0;
.         portc.b5=1;
.         Delay_ms(2000);
.         portc.b4=0;
60         portc.b5=0;
.     }
.
- 2 if(PORTB.B5==1) {
.
.         portc.b4=1;
.         portc.b5=0;
.         Delay_ms(2000);
.         portc.b4=0;
.         portc.b5=0;
.     }
70
.
. }
.
. intcon.intf=0;
. intcon.RBIF=0;
.
. if (intcon.T0IF==1)
- 3 {
.
.     nb--;
- 4 if (nb==0) {
80     nb=127;
.
.     Sound_Play(800, 1000);
.     delay_ms(500);
.     tmr0=0;
.     constante =~constante ;
.     }
.     intcon.t0if=0;
.
. }
90
. }
- 5 void main() {
.
.     INTCON =0xB8; // 10111000 pour activer l'innteruption sur les les port rb(4,5,
.     trisc=0; //configurer le port c en sortie
.     portc=0; //initialisation des sortie du port c en 0
.     trisb=0xFF; //configurer le port b en entrer
.     trisa=0;
.     tmr0=0;
.     constante=0;
100

```

```

.   constante=0;
100 option_reg=0b01000111;
.   nb=127;
.
.   Sound_Init(&PORTC, 3);
104
.   Lcd_Init(); // Initialize LCD
.
.   Lcd_Cmd(_LCD_CLEAR); // Clear display
.   Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
.
110 lcd_out(1,1,"Nom du patient");
.   lcd_out(2,1,"have a good day");
.   delay_ms(800);
.
.   strcpy (med1,"Ebixa 10mg") ;
.   strcpy (med2,"Aricept 10mg");
.
.   for (i=0; i <10;i++) {
.       EEPROM_write(0x00+i , med1[i]);
120   }
.   for (i=0; i <12;i++) {
.       EEPROM_write(0x20+i , med2[i]);
.   }
.
.   while(1) { // Endless loop
.       portc.b3=0;
130   portc.b0=0;
.       portc.b1=0;
.       portc.b2=0;
.
.       Lcd_Cmd(_LCD_CLEAR);
.       value =adc_read(0);
.       value=value*0.488;
.       floattostr(value,value1);
140   lcd_out(2,1,"Temperature");
.
.       lcd_out(2,11,value1);
.       delay_ms(2500);
.

```

```

    delay_ms(2500);

    if ((value<36)|| (value>42))
    {
        portc.b0=1;
        delay_ms(200);
        portc.b0=0;
        delay_ms(200);

        portc.b0=1;
        delay_ms(200);
        portc.b0=0;
        delay_ms(200);

        portc.b0=1;
        delay_ms(200);
        portc.b0=0;

    }

    if ((value>39)&&(value<42))
    {
        portc.b2=1;
        delay_ms(1000);
        portc.b2=0;

    }

    if ((value>36)&&(value<39))
    {
        portc.b1=1;
        delay_ms(1000);
        portc.b1=0;

    }

    if(constante==0){
        delay_ms(2000);
        Lcd_Cmd(_LCD_CLEAR);

        for(i=0;i<10;i++)
            lcd_chr(1,i+1,EEPROM_READ (0X00+i));

        for(i=0;i<12;i++)
            lcd_chr(2,i+1,EEPROM_READ (0X20+i));
    }

```

```

    if ((value>36)&&(value<39))
    {
        portc.B1=1;
        delay_ms(1000);
        portc.b1=0;
    }

    if(constante==0){
        delay_ms(2000);
        Lcd_Cmd(_LCD_CLEAR);

        for(i=0;i<10;i++)
            lcd_chr(1,i+1,EEPROM_READ (0X00+i));

        for(i=0;i<12;i++)
            lcd_chr(2,i+1,EEPROM_READ (0X20+i));

        delay_ms(2000);
    }

    else if(constante==1){
        delay_ms(2000);
        Lcd_Cmd(_LCD_CLEAR);

        for(i=0;i<12;i++)
            lcd_chr(2,i+1,EEPROM_READ (0X20+i));

        delay_ms(2000);
    }
}
}

```

## **Mode de fonctionnement :**

- 1- Au démarrage de notre système :
  - L'afficheur affiche le nom du patient sur la première ligne.
  - L'afficheur affiche "have a good Day" sur la deuxième.
- 2- Chaque 8h (8s dans le code pour pouvoir tester le fonctionnement) :
  - Une bipe sonore.
  - L'affichage du traitement à prendre.
  - Mesure et affichage de la température, avec une LED qui s'allume selon la température mesurer.
- 3- IL existe 3 boutons :
  - Le 1<sup>er</sup> permet de déclencher une bipe sonore et le clignotement de la LED rouge en cas d'urgence.
  - Le 2<sup>ème</sup> permet d'ouvrir le store pour éclairer la chambre, en tournant le moteur dans le sens direct pour 2 s.
  - Le 3<sup>ème</sup> permet de fermer le store, en tournant le moteur dans le sens inverse pendant 2s.