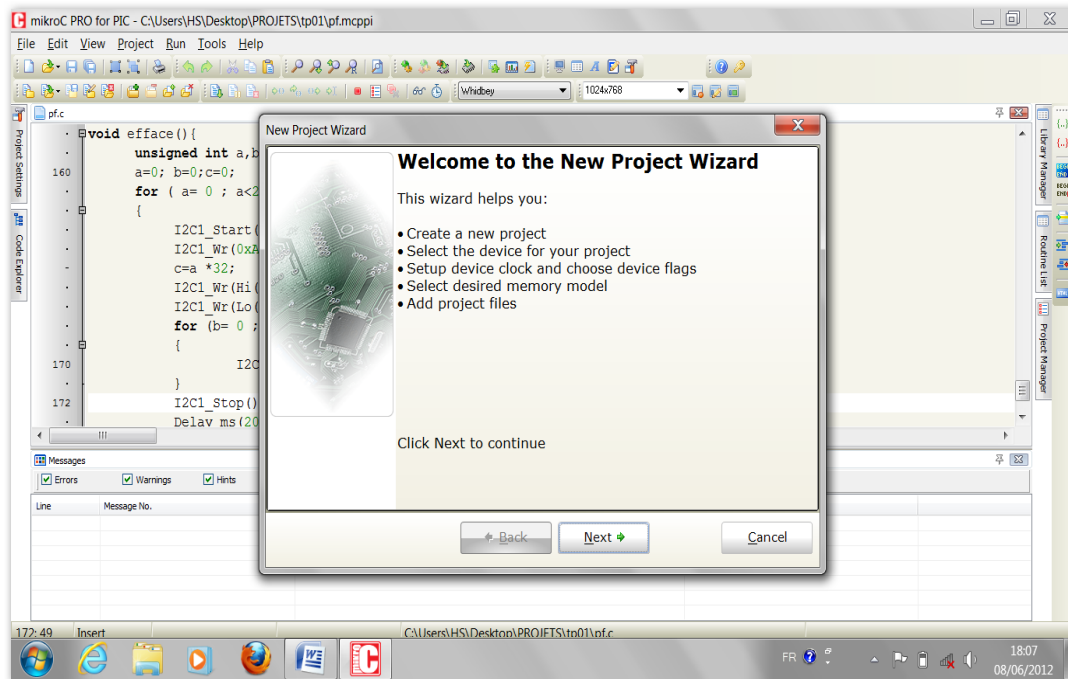


Annexe

- **Première étape : Edition du projet**

Lancer l'application micro-c puis cliquer sur new Project on choisi le type de pic et sa fréquence.



- **deuxième étape : Saisie du programme**

```
#define Lo (param) ((char *) &param)[0]
#define Hi (param) ((char *) &param)[1]
////////// declaration //////////////////////////////////////////
sbit LCD_RS at RD0_bit;
sbit LCD_EN at RD1_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;
```

```
sbit LCD_RS_Direction at TRISD0_bit;
sbit LCD_EN_Direction at TRISD1_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
```

```
unsigned char ch; //sert à afficher les digits de la valeur de la température
char *irra_s = "0000"; //sert à affhcer la valeur de l'irradiance
```

```

char *temp_s = "0000";
unsigned long temp_int, irra_int ; //sert à sauvegarder les valeurs mesurées
unsigned int i, j, k;
unsigned int var1, var2;          //sert à la récupération des données de la mémoire pour les
envoyé vers le PC
unsigned char test;              //un bit qui sert à tester l'appui sur le switch
void efface ();                  //prototype de la fonction efface()

void main () {
    ////////////////////////////////////lcd////////////////////////////////////
    LCD_init ();
    Lcd_Cmd (_LCD_CLEAR);        // Clear display
    Lcd_Cmd (_LCD_CURSOR_OFF);  // Cursor off
    ////////////////////////////////////adc////////////////////////////////////
    ADCON1 = 0X02;               // Configuration des Entrées Analogiques
    TRISA =0b00000011;
    ////////////////////////////////////initialisation pour interruption sur RB0
    OPTION_REG = (OPTION_REG & 0b11111111);
    INTCON.INTF = 0; // Clear interrupt flag prior to enable
    INTCON.INTE = 1; // enable external interrupts
    INTCON.GIE = 1; // enable Global interrupts

    TRISB =0b11111101;
    TRISC =0;
    I2C1_Init (22000); // initialize I2C communication
    if (PORTB.RB2=1)
        efface ();

    i=0; j=0; k=0; test=0;
    PORTB.RB1=1;
    Delay_ms(1000);
    ////////////////////////////////////boucle principale////////////////////////////////////
    ////////////////////////////////////boucle infinie////////////////////////////////////
    do {
        switch (test)
        {
        case 1: //Lecture de la mémoire et envoie des données vers PC
            UART1_Init (9600);
            Delay_ms(100); // Wait for UART module to stabilize
            do
            {
                k=j*4;
                PORTB.RB1=1;
                I2C1_Start (); // issue I2C start signal
                I2C1_Wr (0xA0); // send byte via I2C (device address + W)
                //lecture de l'eeeprom
                I2C1_Wr (Hi (k)); // send byte (data address)
                I2C1_Wr (Lo (k)); // send byte (data address)
                I2C1_Repeated_Start ();
                I2C1_Wr (0xA1); //lecture
            }
        }
    } while (1);
}

```

```

    Lo (var1) =I2C1_Rd (1u);
    Delay_ms (20);
    Hi (var1) =I2C1_Rd (1u);
    Delay_ms (20);
    Lo (var2) =I2C1_Rd (1u);
    Delay_ms (20);
    Hi (var2) =I2C1_Rd (0u);
    Delay_ms (20);
    if (var1 == 0xFFFF && var2 == 0xFFFF)
        break;
    I2C1_Stop ();
    IntToStr ( var1, temp_s);
    IntToStr ( var2, irra_s);
    //envoi RS232
    UART1_Write_Text ("Température est :");
    UART1_Write_Text (temp_s);
    UART1_Write_Text (" ");

    UART1_Write_Text ("irradiance est :");
    UART1_Write_Text (irra_s);
    UART1_Write (13);
    j=j+1;
    if (~test)
        break;

} while (1);
break;
case 0:
    PORTB.RB1=0;
    ////////////////////////////////////////////mesure//////////////////////////////////////////
    Lcd_Out (1, 2, "tempe - irrad");
    temp_int = ADC_Read (0);
    temp_int = temp_int * 500;
    temp_int = temp_int / 1023;          // 0..1023 -> 0-5000mV

    ch  = temp_int / 100;                // extract volts digit
    LCD_Chr (2, 2, 48+ch);              // write ASCII digit at 2nd row, 9th column

    ch  = (temp_int / 10) % 10;          // extract volts digit
    LCD_Chr_CP (48+ch);                 // write ASCII digit at 2nd row, 9th column
    LCD_Chr_CP ('.');

    ch  = temp_int % 10;                // extract 0.1 volts digit
    LCD_Chr_CP (48+ch);                 // write ASCII digit at cursor point

    Lcd_Out_Cp ("C");
    Delay_100ms ();
    irra_int = ADC_Read (1);
    irra_int = irra_int * 500/1023 ;

```

```

    Delay_100ms ();

    ////////////température////////////////////////////////////
    //IntToStr (temp_int, temp_s);
    //Lcd_Out (2, 1, temp_s);
    ////////////irradiance////////////////////////////////////
    IntToStr (irra_int, irra_s); // txt is " -4220" (one blank here)
    Lcd_Out (2, 10, irra_s);
    ////////////stocage////////////////////////////////////

    I2C1_Start ();      // issue I2C start signal
    I2C1_Wr (0xA0);      // send byte via I2C (device address + W)
    I2C1_Wr (Hi (i));    // send byte (address of EEPROM location)
    I2C1_Wr (Lo (i));    // send byte (address of EEPROM location)
    I2C1_Wr (Hi (temp_int)); // send data (data to be written)
    I2C1_Wr (Lo (temp_int)); // send data (data to be written)
    I2C1_Wr (Hi (irra_int)); // send data (data to be written)
    I2C1_Wr (Lo (irra_int)); // send data (data to be written)
    I2C1_Stop ();      // Issue stop signal
    i=i+4;
    PORTB.RB1=~PORTB.RB1;
    Delay_ms (500);
    break;
}
} while (1);
}

void interrupt (void){
    if (INTCON.INTF == 1)
    {
        if (Button (&PORTB, 0, 100, 1)){ // introduce debounce
            // RB0 100ms active Hi
            test = ~test; //
        } // end button if
        INTCON.INTF = 0; // Clear flag everytime inside INTF
    } //end intcon.intf if
}

void efface (){
    unsigned int a, b, c;
    a=0; b=0; c=0;
    for (a= 0; a<256; a++)
    {
        I2C1_Start ();      // issue I2C start signal
        I2C1_Wr (0xA0);      // send byte via I2C (device address + W)
        c=a *32;
        I2C1_Wr (Hi(c));    // envoi de l'adresse sur 2 octets
        I2C1_Wr (Lo(c));
        for (b= 0 ; b<32; b++) // envoi du paquet 32

```

```
    {  
        I2C1_Wr (0xFF);  
    }  
    I2C1_Stop ();          // Issue stop signal  
    Delay_ms(20);  
}  
I2C1_Stop ();  
}
```