

```
/* Nom du Projet : LED_LCD_DS1621_UART
```

```
   Date de création : 04-12-2014 */
```

```
char temp[7], temptxt[]="Temperature", LSB;
```

```
signed char MSB;
```

```
/******Déclaration des fonctions******/
```

```
void lcd(int li, int co, int nb, char receive[15]);
```

```
void xbee(int nb, char receive[15]);
```

```
void main() {
```

```
/******Initialisation et affectation des ports******/
```

```
    PORTB=0;           //initialisation
```

```
    PORTC=0;
```

```
    TRISB=0;           //Port B déclarer en sortie
```

```
    TRISC=0xFF;         //Port C déclarer en entrée
```

```
    PORTD = 0;          // Initialize PORTD
```

```
    TRISD = 0;          // Configure PORTD as output
```

```
/******Configuration et initialisation de l'écran LCD*****/
```

```
    Lcd_Config(&PORTB, 0, 1, WR, 5, 4, 3, 2); //Configure l'ecran
```

```
    Lcd_Init(&PORTB);      //initialisation du LCD
```

```
    Lcd_Cmd(Lcd_CURSOR_OFF); //Enleve le curseur
```

```
    Lcd_Cmd(Lcd_CLEAR);    //Efface l'ecran
```

```
/******Initialisation du module I2C******/
```

```

I2C_Init(100000);    //initialisation de la communication I2C

I2C_Start();        //Détermine si l'I2C est libre et lance le signal

I2C_Wr(0x90);       //Mode de contrôle en mode écriture

I2C_Wr(0xAC);       //Accès au registre de configuration

I2C_Wr(0x0A);       //Registre de configuration mesure en continu

I2C_Stop();         //Arrêt du signal

```

```

I2C_Start();        //Détermine si l'I2C est libre et lance le signal

I2C_Wr(0x90);       //Mode de contrôle en mode écriture

I2C_Wr(0xEE);       //Début de la conversion

I2C_Stop();         //Arrêt du signal

```

```

/*****Boucle infinie*****/

```

```

while(1) {

    I2C_Start();    //Détermine si l'I2C est libre et lance le signal

    I2C_Wr(0x90);   //Mode de contrôle en mode écriture

    I2C_Wr(0xAA);   //Lecture de la température

    I2C_Stop();     //Arrêt du signal


    I2C_Start();    //Détermine si l'I2C est libre et lance le signal

    I2C_Wr(0x91);   //Mode de contrôle en mode lecture

    MSB = I2C_Rd(1); //Nombre signé [température entre +125° et -55°C]

    LSB = I2C_Rd(0); //Si bit 7 = 1 température MSB +0,5°C

    I2C_Stop();     //Arrêt du signal


    if(MSB<0) {

        MSB = abs(MSB); // Si temperature négative valeur absolue de MSB
    }
}

```

```

    Temp[0]=45; } //Signe - affecté en ascii

else {

    Temp[0]=43; } //signe + affecté en ascii


    Temp[1]=(MSB/10)+48;    //Valeur des dizaines affecté en ascii
    Temp[2]=(MSB%10)+48;    //Valeur des unités affecté en ascii
    Temp[3]=44;             //Virgule affecté en ascii


    if (LSB==128) {         //Si bit 7 à 1 alors +0.5°

        Temp[4]='5'; }

    else {                  //Si bit 7 à 0 alors +0.0°

        Temp[4]='0'; }

    Temp[5]=223;           // degré affecté en ascii
    Temp[6]=67;            //C affecté en ascii


    lcd(1, 1, 11, temptxt); //Affiche la chaîne de caractère temptxt sur le LCD
    Xbee(11, temptxt); //Envoi la chaîne de caractère temp dans la fonction xbee
    delay_ms(100);

    lcd(2, 1, 7, temp); //Affiche la chaîne de caractère temp sur le LCD
    Xbee(7, temp); //Envoi la chaîne de caractère temp dans la fonction xbee
    delay_ms(100);


    delay_ms(1000);

    PORTD = ~PORTD; // toggle PORTD

    delay_ms(1000);

    }

}

```

```
/******Fonction affichage******/
```

```
void lcd(int li, int co, int nb, char receive[15]) {  
  
    int i;  
  
    for(i=0; i<nb ; i++) {          //Boucle for  
  
        Lcd_chr(li, (co+i), receive[i]);} //Affiche les caractères jusqu'à i
```

```
/******Usart******/
```

```
void xbee(int nb, char receive[15]) {  
  
    int i;  
  
    Usart_Init(9600);          //Vitesse de transmission  
  
    for(i=0; i<nb ; i++) {      //Boucle for  
  
        Usart_Write(receive[i]); } //Envoi les caractères  
  
    i = '\r' ;                  //Saut de ligne  
  
    Usart_Write(i); }
```