



TP 3 : Mise en œuvre d'un capteur de température I2C DS1621.

Salles G45-G46 Bât 3A (voir plan fac page 2)

Responsables TPs :

Hélène LEYMARIE helene.leymarie@univ-tlse3.fr

Thierry PERISSE thierry.perisse@univ-tlse3.fr

Technicien : **Franck Lacourrège**

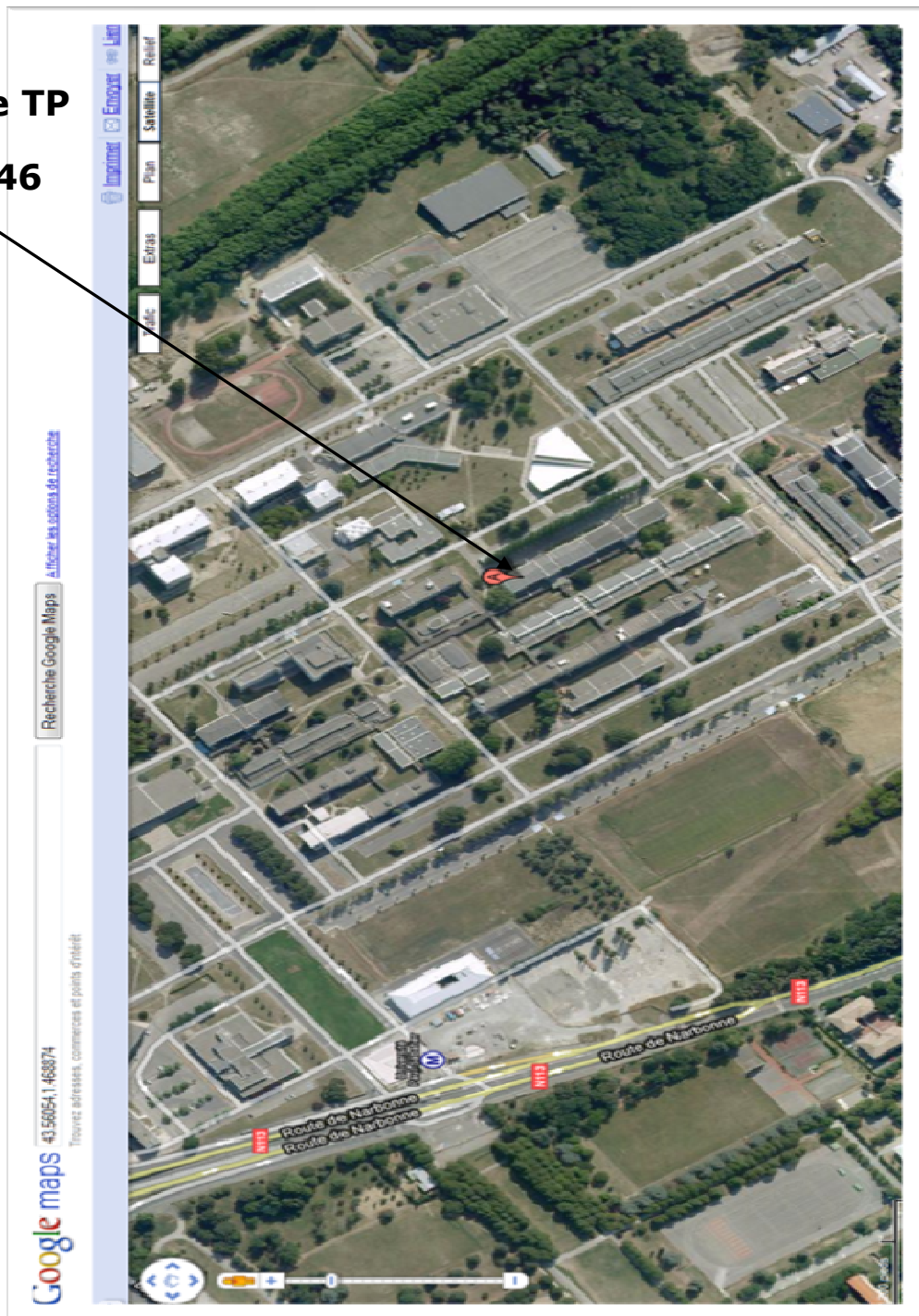
Documentations :

***Norme I2C / Capteur de Température I2C
DS1621***

Carte développement PIC 16F877

Année 2015-16

**Salles de TP
G45/G46**



Plan : Salles G45/G46
Pour ne pas vous tromper de salle
et donc arriver à l'heure !!!

TP3 : Mise en œuvre d'un capteur de température I2C.(4h)

- 1. : Projet n°1 : LED**
- 2. : Projet n°2 : LED_LCD**
- 3. : Projet n°3 : LED_LCD_DS1621**
- 4. : Projet n°4 : LED_LCD_DS1621autonome**
- 5 : Projet n°5 : Bilan énergétique**

PREPARATION et MANIPULATION**A– PREPARATION** (la préparation doit être jointe au compte rendu en fin de séance)**Préparation projet n°1 :**

Récupérer et analyser le programme permettant de faire clignoter la LED.

Repérer sur la maquette les différents ports d'E/S (à l'aide du schéma électrique).

Préparation projet n°2 :

Récupérer et analyser le programme permettant d'afficher « Salut » sur le LCD.

Préparation projet n°3 : LED_LCD_DS1621 Norme I2C

Etude de la norme I2C.

Repérer sur le microcontrôleur les broches SDA et SCL.

A l'aide du document constructeur donner le schéma de câblage du DS1621

Préparation projet n°4 : LED_LCD_DS1621 autonome.

Donner le schéma de câblage si on prend un accu de 9V et un régulateur 7805.

Préparation Projet n°5 : Consommation

Réfléchir aux différentes mesures (on précisera le câblage)

B– MANIPULATION

Une validation de chaque partie expérimentale doit être faite avec un responsable.

Les programmes doivent être commentés.

Les câblages doivent être soignés.

Attention à respecter les couleurs

+Vcc	—>	Rouge
-Vcc	—>	Bleu
Masse	—>	Noir

Un compte rendu doit être rendu en fin de séance. (soit au bout des 4h)

TP 3

Afficher sur un LCD la valeur de la température donnée par un capteur I2C (DS1621)

Objectifs : Acquérir les bases pour utiliser un logiciel (mikroC) permettant de programmer (programmeur PicFlash) un microcontrôleur (PIC16F877 ou PIC16F877A) avec un langage de haut niveau (langage C).

Prise en main du système (logiciel, matériel, ...).

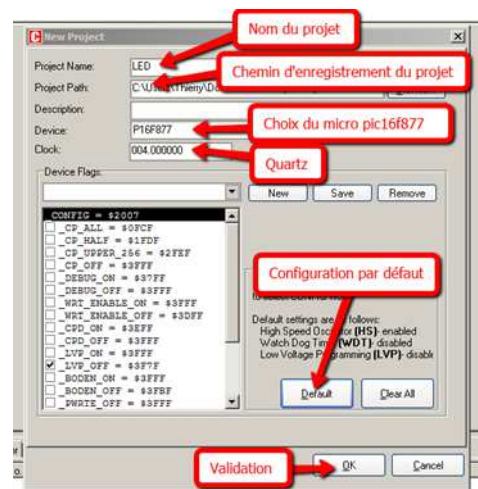
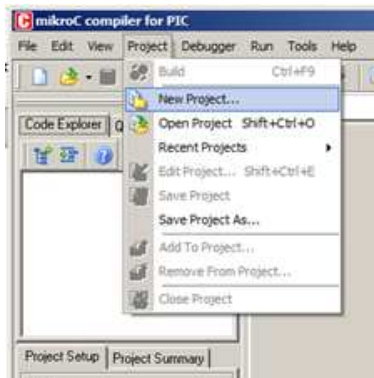
Lancer le Logiciel MicroC présent sur le bureau du PC :

Projet n°1 : LED

Création du premier projet : LED.

Pour chacun des programmes créer un répertoire de même nom LED que l'on rangera dans le répertoire

Mes documents/TPDomotique2014/Nom1-Nom2/LED/**

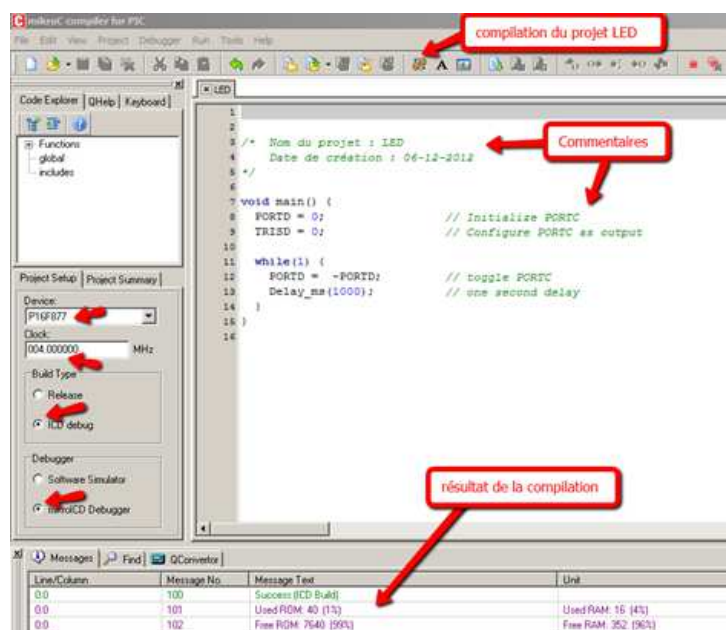


Recherche d'un programme exemple : Clignotement d'une LED.(chargement de Led_Blinking.c)

Chercher le fichier sur le chemin suivant : File / Open / C / Programme / Mikroelectronica / MikroC / Examples / Easypic4 / P16F877 ou P16F877A / Led_Blinking / Led_Blinking.c

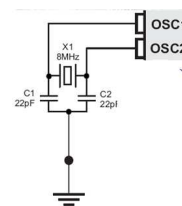
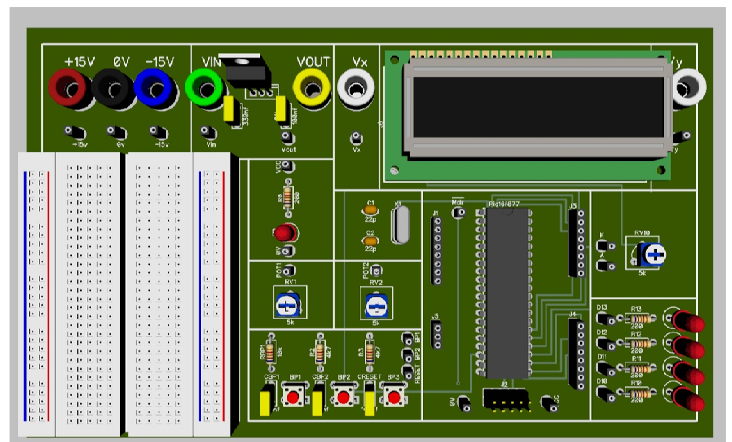
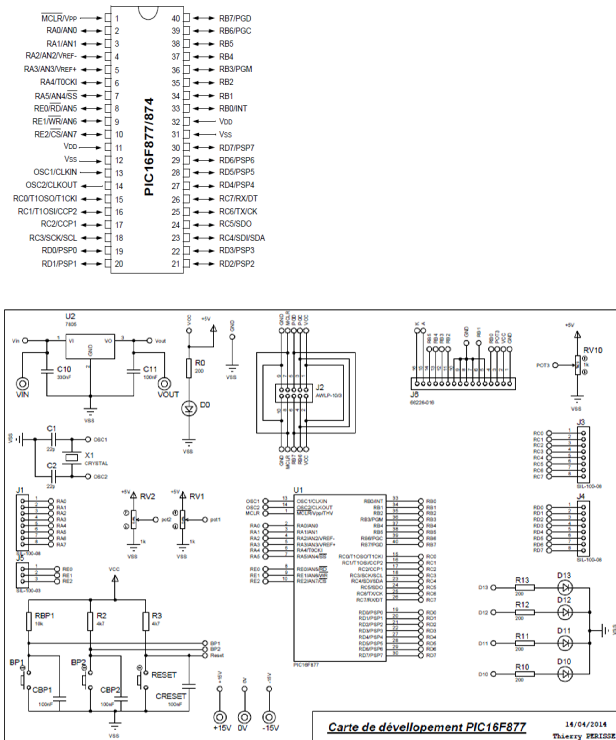
Copier / coller le contenu de ce fichier sur votre projet (ici LED)

Le programme utilise le port C hors nous voulons faire clignoter une LED sur le port D apporter les modifications si nécessaires et compiler le fichier.



Mettre vos propres commentaires sur le programme.

Réaliser le câblage sur la carte de développement PIC16F877 : (en se servant du schéma électrique)



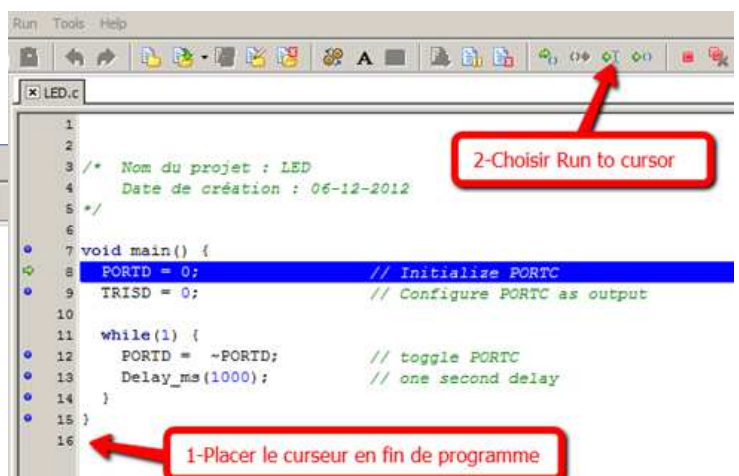
Branchons maintenant le programmeur PICFlash sur le connecteur afin de transférer le programme dans le microcontrôleur PIC16F877.



Maintenant il faut lancer le programme :



Vérifier le bon fonctionnement.



Projet n°2 : LED_LCD

Création d'un projet : LED_LCD que l'on enregistrera dans le répertoire Mes documents/TPDomotique2014/LED_LCD/

Aller chercher le programme permettant d'afficher « Salut » et de faire clignoter une LED sur le portD.

Pour utiliser l'afficheur, on souhaite utiliser le PortB avec les affectations suivantes : 0,1,5,4,3,2 pour : RS, EN, D7, D6, D5, D4.

En utilisant les routines du LCD (voir ci-dessous) :
Justifier les lignes de codes du programme ci-contre.

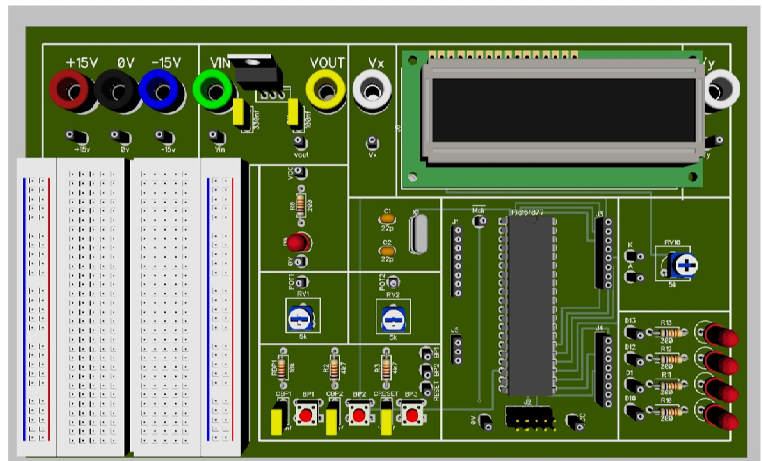
```

1
2 /* Nom du Projet : LED_et_LCD
3    Date de création : 06-12-2012
4    PIC / 16F877
5 */
6 char *text = "Salut";
7
8 void main() {
9     Lcd_Config(&PORTB, 0, 1, WR, 5, 4, 3, 2);
10    Lcd_Init(&PORTB); // Initialize LCD connected to PORTB
11    Lcd_Cmd(Lcd_CLEAR); // Clear display
12    Lcd_Cmd(Lcd_CURSOR_OFF); // Turn cursor off
13
14    PORTD = 0; // Initialize PORTC
15    TRISD = 0; // Configure PORTC as output
16
17    while(1) {
18        PORTD = ~PORTD; // toggle PORTD
19        Delay_ms(1000); // one second delay
20        Lcd_Out(1, 1, text); // Print text to LCD, 2nd row, 1st column
21    }
22 }
23

```

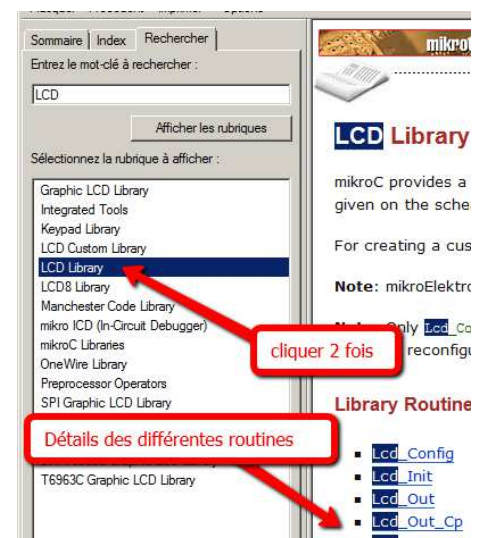
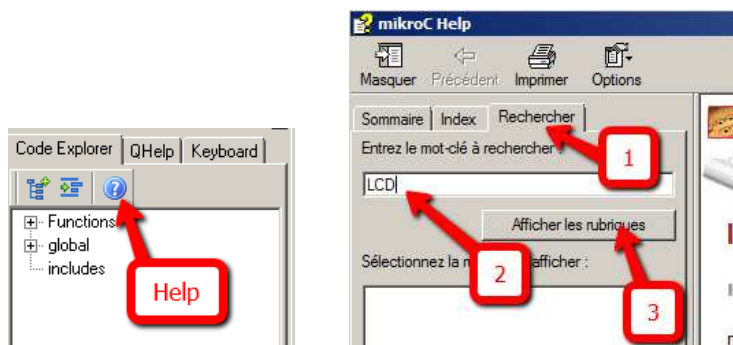
configuration LCD

Aucun câblage supplémentaire n'est nécessaire pour tester le fonctionnement du LCD.



Pour avoir les détails sur les routines pilotant le LCD cliquer sur le Help

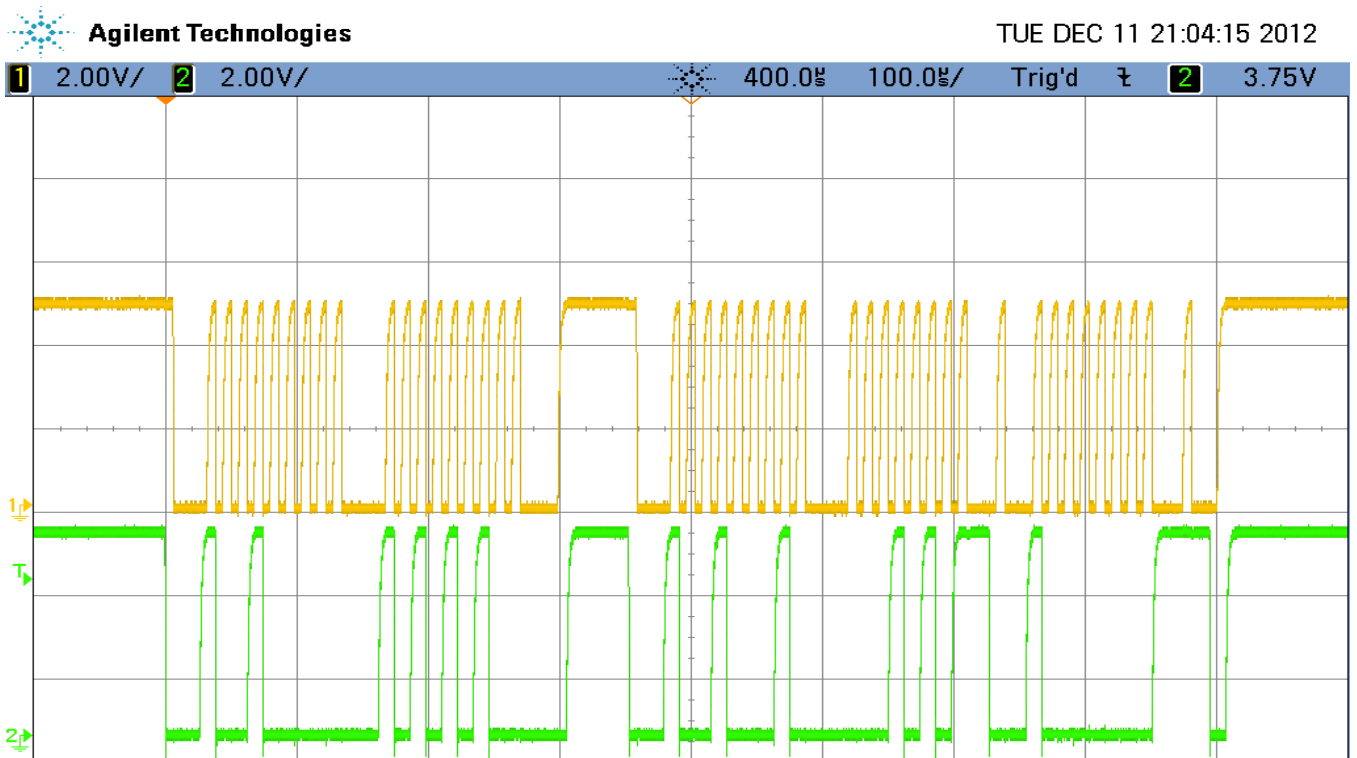
Rechercher / Taper LCD / Afficher les rubriques / LCD Library



Compiler le programme, transférer le prog dans le microcontrôleur, lancer le prog et valider le bon fonctionnement du projet LED_LCD.

Préparation projet n°3 : LED_LCD_DS1621 Norme I2C

Ci-dessous un enregistrement des lignes SDA et SCL dans un exemple de communication entre le PIC16F877 et le DS1321 :



```

LED_LCD_DS1621.c
39 /*****Boucle infinie*****/
40 while(1) {
41     I2C_Start(); //Détermine si l'I2C est libre et lance le signal
42     I2C_Wr(0x90); //Mode de contrôle en mode écriture
43     I2C_Wr(0xAA); //Lecture de la température
44     I2C_Stop(); //Arrêt du signal
45     I2C_Start(); //Détermine si l'I2C est libre et lance le signal
46     I2C_Wr(0x91); //Mode de contrôle en mode lecture
47     MSB = I2C_Rd(1); //Nombre signé [température entre +125° et -55°C]
48     LSB = I2C_Rd(0); //Si bit 7 = 1 température MSB +0,5°C
49     I2C_Stop(); //Arrêt du signal
50     if (MSB<0) {
51         MSB = abs(MSB); // Si temperature négative valeur absolue de MSB
52         Temp[0]=45; } //Signe - affecté en ascii
53     else {
54         Temp[0]=43; } //signe + affecté en ascii
55     Temp[1]=(MSB/10)+48; //Valeur des dizaines affecté en ascii
56     Temp[2]=(MSB%10)+48; //Valeur des unités affecté en ascii
57     Temp[3]=44; //Virgule affecté en ascii
58     if (LSB==128) { //Si bit 7 à 1 alors +0.5°
59         Temp[4]='5'; }
60     else { //Si bit 7 à 0 alors +0.0°
61         Temp[4]='0'; }
62     Temp[5]=223; // degré affecté en ascii
63     Temp[6]=67; //C affecté en ascii
64     lcd(1, 1, 11, temptxt); //Affiche la chaîne de caractère temptxt sur le LCD
65     delay_ms(100);
66     lcd(2, 1, 7, temp); //Affiche la chaîne de caractère temp sur le LCD
67     delay_ms(100);
68     PORTD = ~PORTD; // toggle PORTD
69     delay_ms(1000);
70 }

```

A l'aide de la boucle infinie ci-jointe et des signaux SDA et SCL ci-dessus :

- Repérer les différents octets présent sur la ligne SDA et en donner leurs significations ?
- Sur cet exemple quelle est la température présente sur le capteur DS1621 ?

Projet n°3 : LED_LCD_DS1621

Création d'un projet : LED_LCD_DS1621 ranger dans le répertoire Mes documents/TPDomotique2014/LED_LCD_DS1621/

Câblage supplémentaire : DS1621

Alimentation : 5V/0V. V_{DD} GND

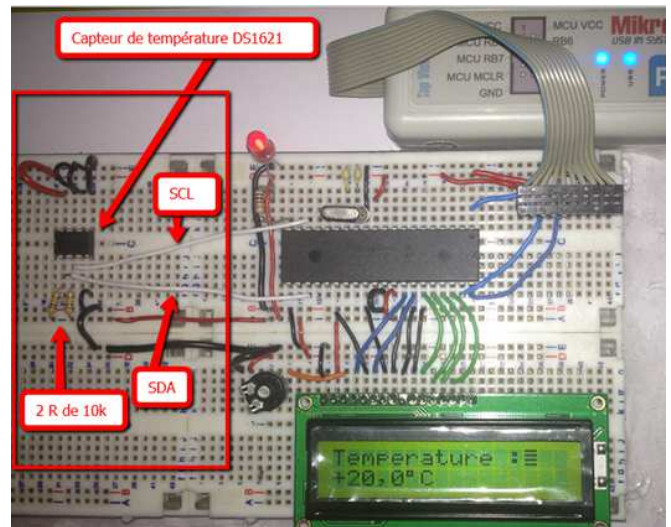
Adresse physique : 3 broches $A_0A_1A_2$

SDA et SCL relier au PIC 16F877

DS1621 Digital Thermometer and Thermostat

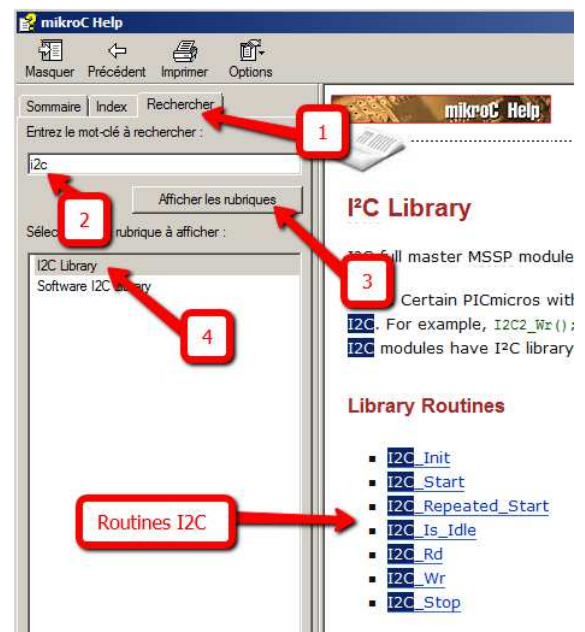
PIN ASSIGNMENT

SDA	1	8	V_{DD}
SCL	2	7	A_0
T_{OUT}	3	6	A_1
GND	4	5	A_2



Recherche d'aide sur la programmation de composants I2C :

A l'aide du programme donné en annexe LED_LCD_DS1621 :



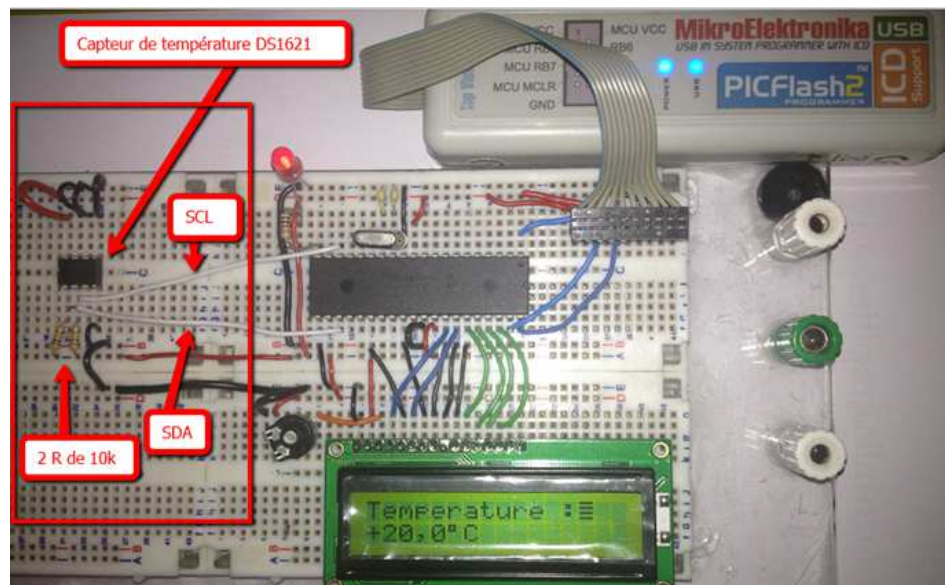
Repérer et commenter les lignes de commande concernant l'I2C.

Valider le bon fonctionnement du projet.

Projet n°4 : LED_LCD_DS1621autonome

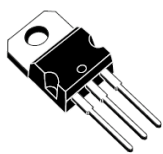
Création d'un projet : (On travaille exactement sur le même projet que précédemment)

LED_LCD_DS1621autonome ranger dans le répertoire Mes documents/TPDomotique2014/LED_LCD_DS1621autonome/



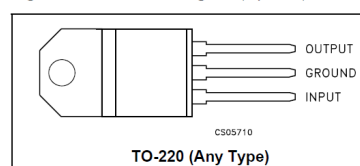
Rendre le système autonome : on alimente le montage via une pile de 9V et un régulateur de tension 7805, le but étant d'enlever la connexion avec le PC.

Voir aussi la doc constructeur du 7805 :

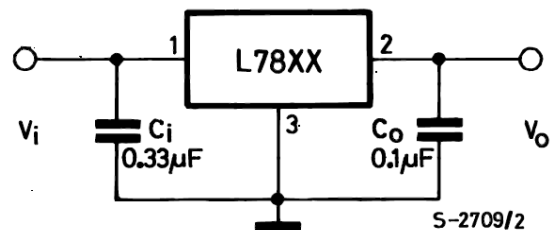


TO-220

Figure 3: Connection Diagram (top view)



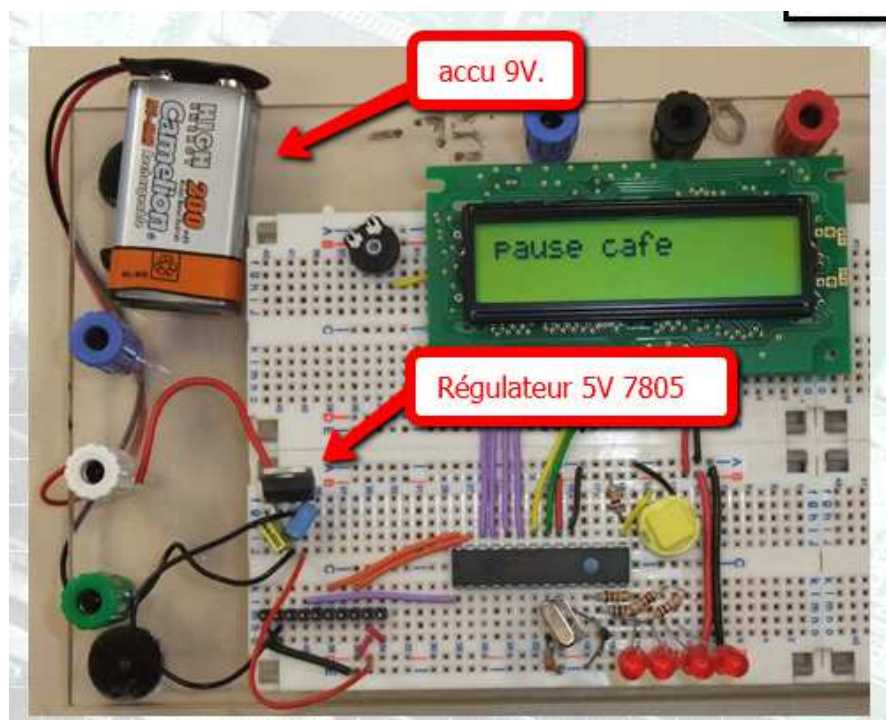
TO-220 (Any Type)



Pour utiliser le µP en mode autonome :

- cocher release,
- compiler,
- programmer.
- enlever le connecteur .
- relier MCUMCLR à +VCC

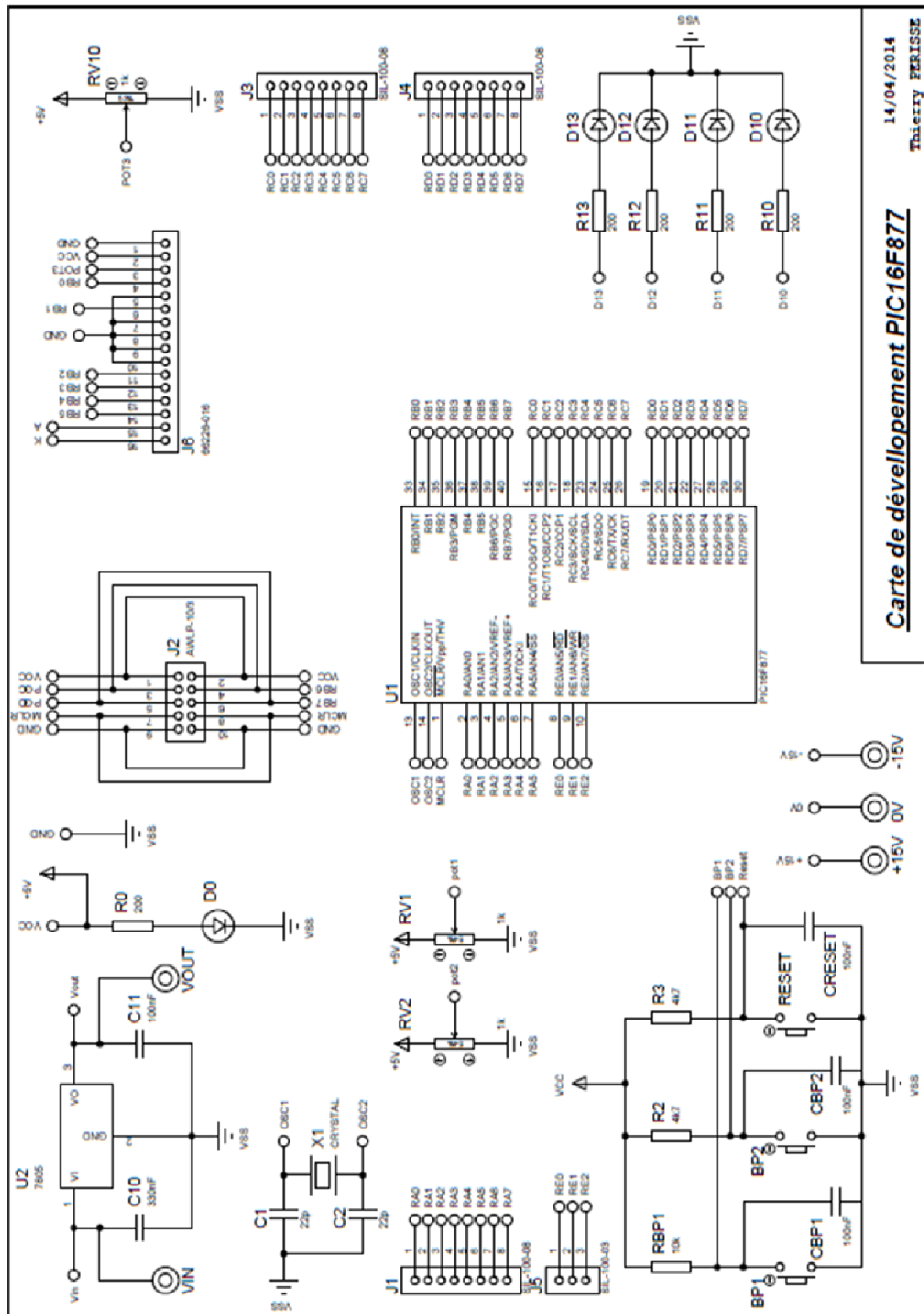
Vérifier le bon fonctionnement.



Projet n°5 : Consommation***A- Bilan énergétique des projets 3 et 4 :******Mesurer la consommation, calculer la durée de fonctionnement***

- *donner le schéma de câblage permettant de visualiser l'image du courant (R_{shunt} proche de 1Ω)*
- *A l'aide de V_{Rshunt} mesurer à l'oscillo donner le courant consommé par le montage.*
- *A l'aide de l'ACCU 300mA/9V donner la durée de fonctionnement du projet.*

Schéma de la carte de développement PIC 16F877 :



Carte de développement PIC16F877

14/04/2014

Thierry PERISSE