



TP DS1621

Durée : 4H

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**

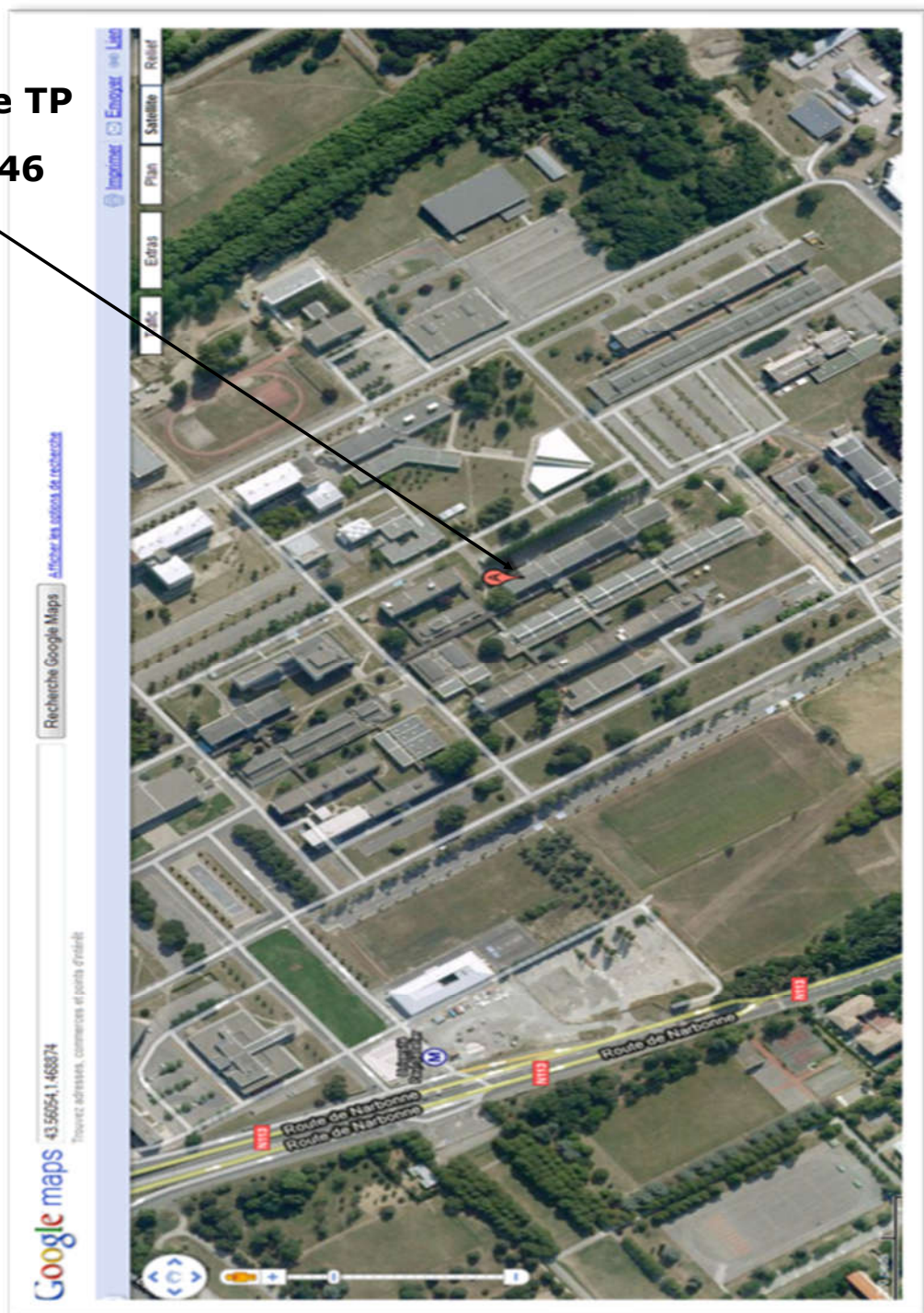
TP1 a : Mise en œuvre d'un capteur de température I2C.(4h)

Documentations dans le répertoire TP-domotique-docsPRO:

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

Carte développement PIC 16F877

**Salles de TP
G45/G46**



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

TP1 a Mise en œuvre d'un capteur de température I2C

- 1. : Projet n°1 : ledmaquette**
- 2. : Projet n°2 : lcdmaquette**
- 3. : Projet n°3 : lcd_ds1631maquette**
- 4. : Projet n°4 : lcd_ds1631maquette en autonomie**

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 « test » sur le LCD.

Préparation projet n°3 :

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 :

Donner le schéma de câblage si on alimente la maquette par une pile de 9V et un régulateur 7805.

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 (.pdf) doit être rendu en fin de séance.

Objectifs : Acquérir les bases pour utiliser un logiciel (mikroC PRO for PIC) permettant de programmer (programmeur mikroProg) un microcontrôleur (PIC16F877ou pic16F877A) avec un langage de haut niveau (langage C).

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

Lancer le Logiciel mikroC PRO for PIC présent sur le bureau du PC :



Projet n°1 : led

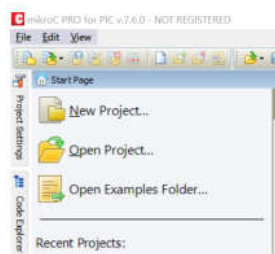


Création du premier projet : led.mcpai

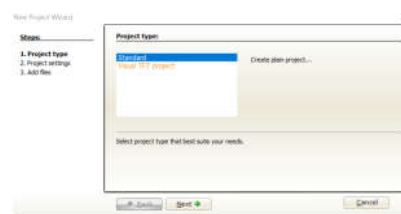
Pour chacun des programmes créer un répertoire du même nom **led** que l'on rangera dans le répertoire

Mes documents/TPcesi2021/Nom1-Nom2/led/**

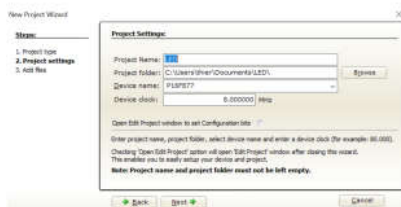
But: Faire clignoter une LED toutes les secondes (sortie7 du PORTD). (Lire le nom du pic et la fréquence du quartz sur votre maquette)



Créer un nouveau projet



Choisir Projet standard



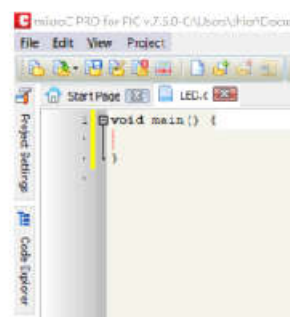
Choisir le nom du projet LED, le répertoire de sauvegarde .../LED/, le microcontrôleur 16F877, l'horloge 8Mhz



Pas de fichier à ajouter au projet



Choisir les paramètres de Project Settings

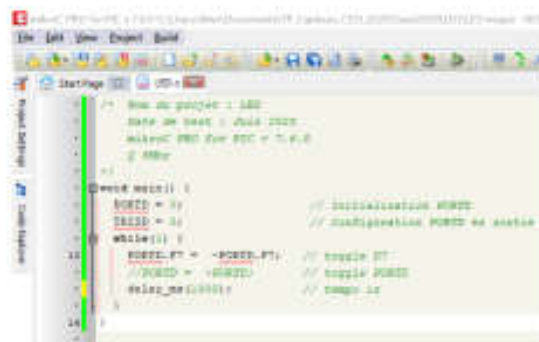


Projet vide

```

*/
void main() {
    PORTD = 0;           // Initialisation PORTD
    TRISD = 0;           // Configuration PORTD en sortie
    while(1) {
        PORTD.F7 = ~PORTD.F7; // toggle D7
        //PORTD = ~PORTD;      // toggle PORTD
        delay_ms(1000);       // tempo 1s
    }
}

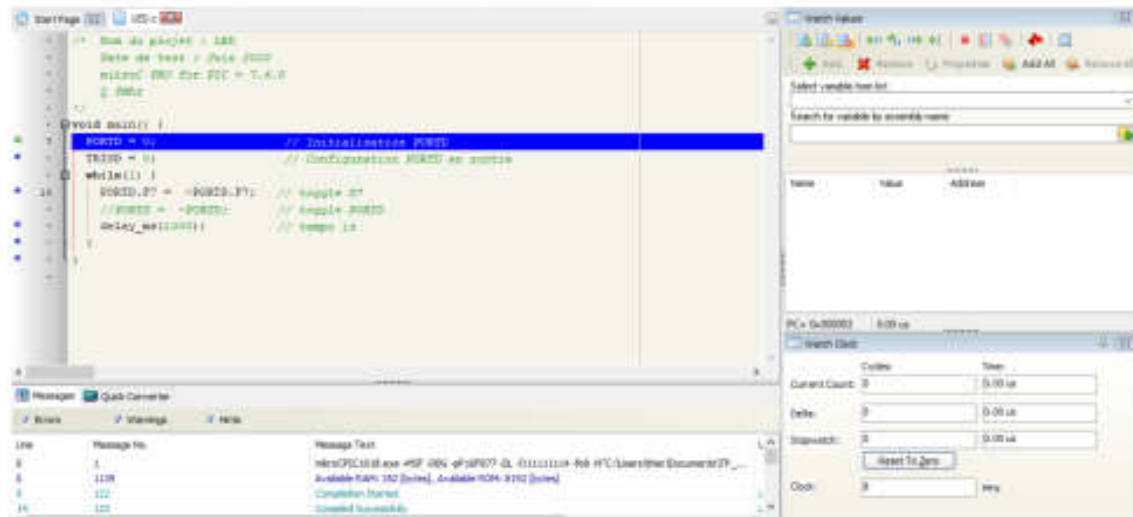
```



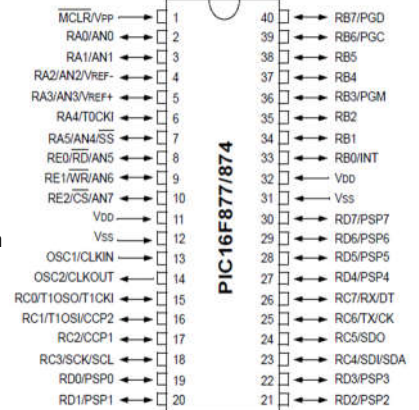
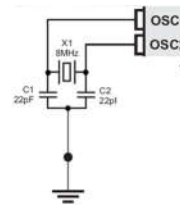
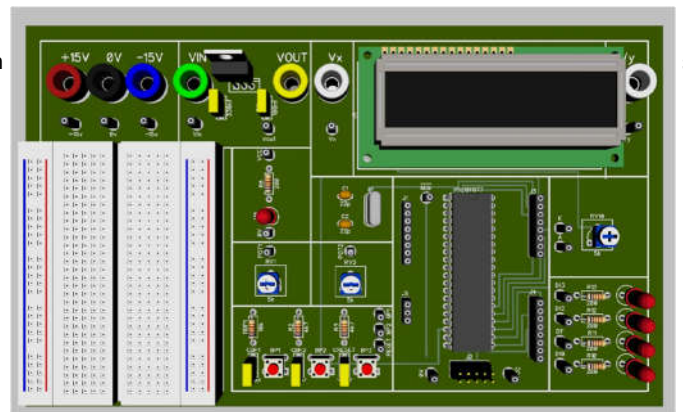
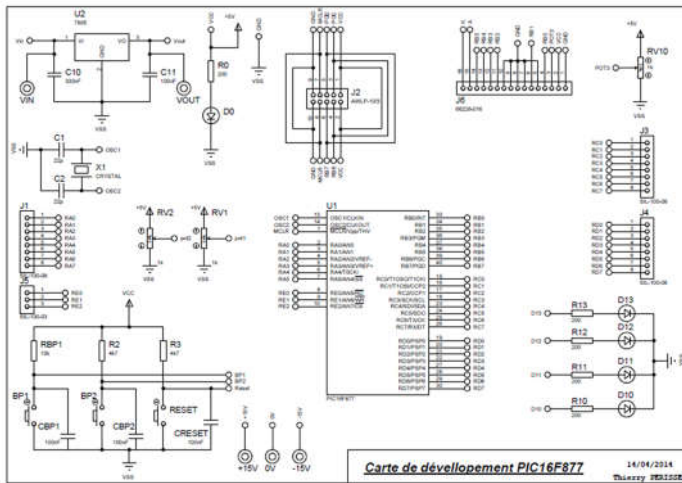
Copier / Coller du programme ci-dessus
LED Compiler le projet "Build", vérifier
dans la fenêtre "Messages" il y a des
erreurs.



Regarder maintenant dans le répertoire
LED les fichiers qui ont été créés (en
particulier "LED.hex" qui va nous servir
pour programmer le microcontrôleur dans
PROTEUS)



Réaliser le câblage sur la carte de développement PIC16F877 en servant du schéma électrique en annexe (fin du TP).

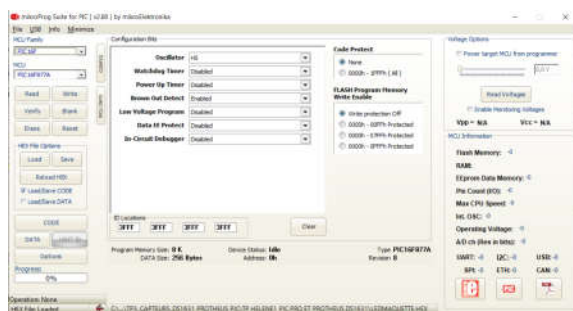


Branchons maintenant le programmeur mikroProg sur le connecteur (fil bleu à droite)

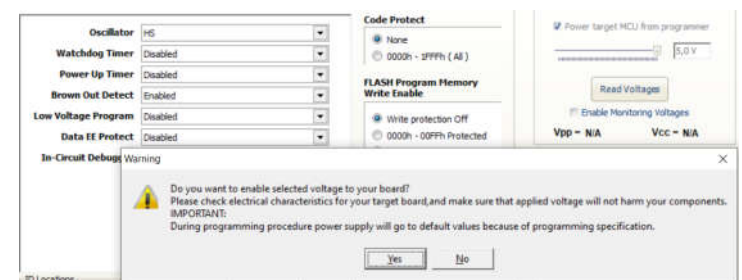
afin de transférer le programme dans le microcontrôleur PIC16F877.



Transférer le programme en cliquant sur Program. Le logiciel mikroProg suite for Pic est lancé et télécharge le fichier led.hex à transférer.



Write



Mettre le curseur à 5V

Alimenter en cochant le power target Mcu programmer et répondre yes

La maquette est alimentée (voyant allumé).

le programme fonctionne et on le visualise avec la led qui clignote sur le port D.F7 que l'on aura câblé.

Si le programme n'est pas lancé, il faut une impulsion sur la broche MCU_MCLR à 5V brièvement.

(Sur certains programmeurs, il faut laisser la broche MCU_MCLR à 5V de façon permanente. On retire le 5V du MCU_MCLR quand il faudra de nouveau programmer le pic)

Projet n°2 : lcdmaquette

Création d'un projet : lcdmaquette que l'on enregistrera dans le répertoire **Mes documents/TPcesi2021/Nom1-Nom2/lcdmaquette/*****

Aller chercher le programme permettant d'afficher « test: pause café » 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.

```

// LCD module connections
10 sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;

sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;
// End LCD module connections

char *text = "TEST";
char *text2 = "PAUSE CAFE";

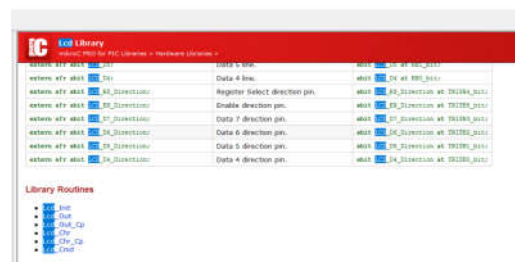
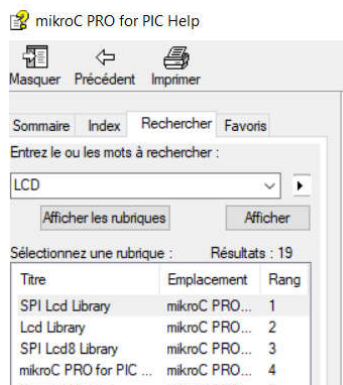
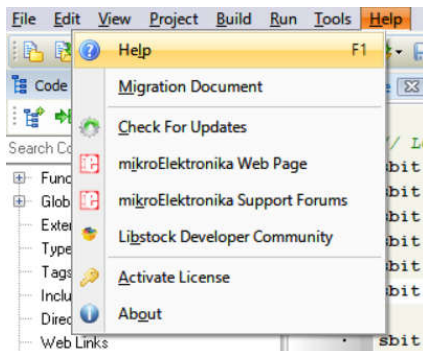
void main() {
30 Lcd_Init();
Lcd_Cmd(_LCD_CLEAR); // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF); // Turn cursor off

PORTD = 0; // Initialise PORTC
TRISD = 0b01000000; // Configure D6 en entrée et les autres en sorties

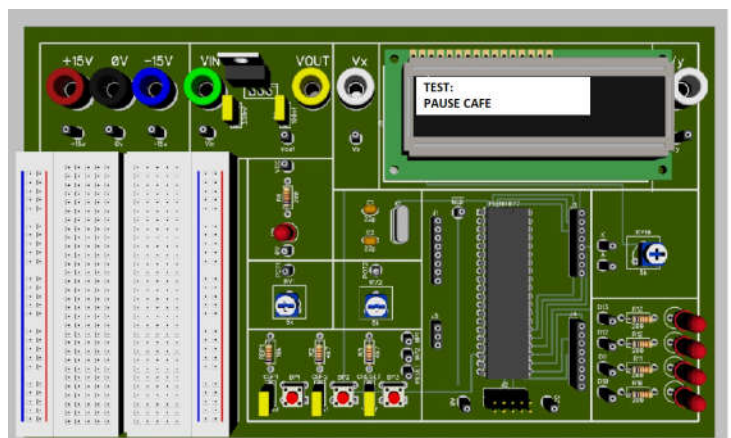
```

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

Rechercher / Taper LCD / Afficher les rubriques / LCD Library



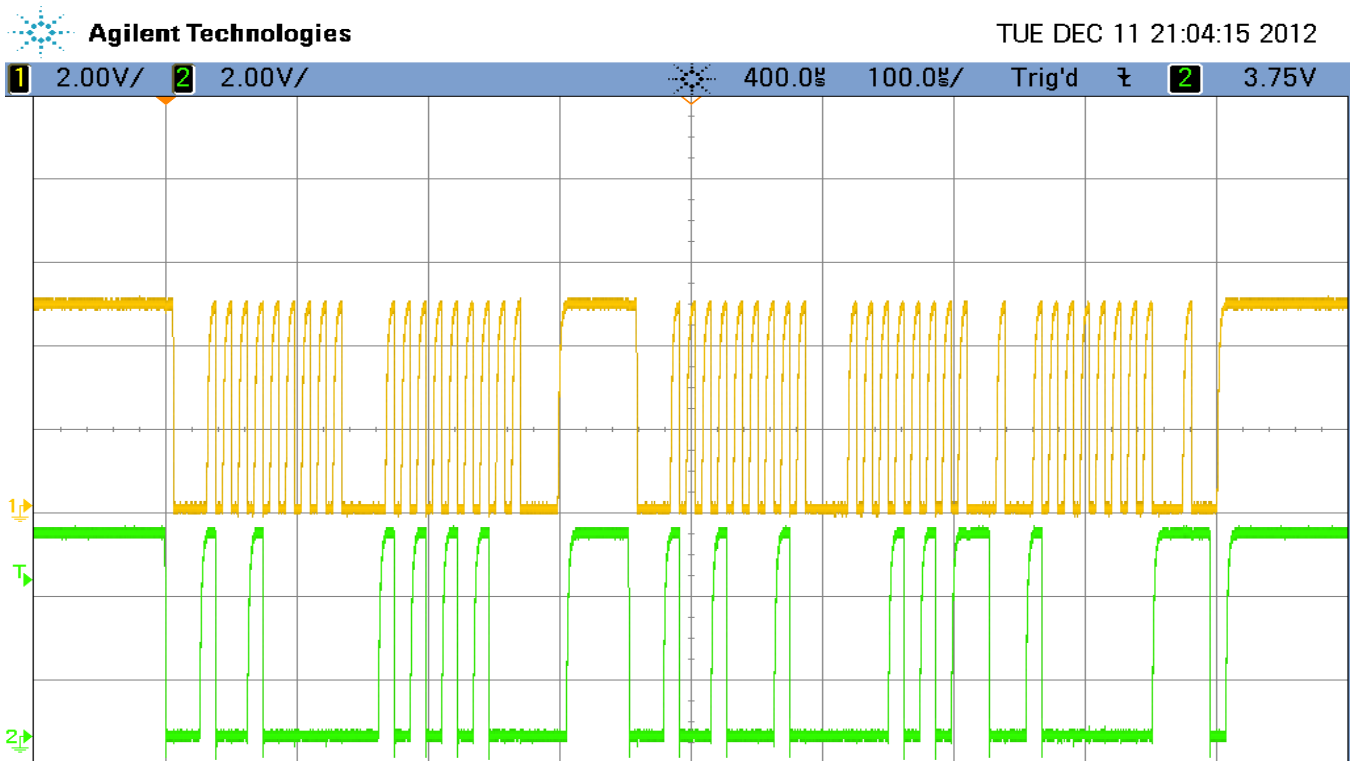
Aucun câblage supplémentaire n'est nécessaire pour tester le fonctionnement du LCD.



Compiler le programme, transférer le programme dans le microcontrôleur et vérifier le bon fonctionnement du projet **lcdmaquette**. Faites valider

Préparation projet n°3 : lcd_ds1621maquette

Ci-dessous un enregistrement des lignes SDA et SCL dans un exemple de communication entre le PIC16F877 et le DS1621 :



```

PORTD = 0; // Initialize PORTC
TRISD = 0x01000000; // Configure D6 en entrée et les autres en sorties*/

80 while(1) {
    I2C1_Start(); //Détermine si l'I2C est libre et lance le signal
    I2C1_Wr(0x90); //Mode de contrôle en mode écriture
    I2C1_Wr(0xAA); //Lecture de la température
    I2C1_Stop(); //Arrêt du signal
    Delay_ms(10);
    I2C1_Start(); //Détermine si l'I2C est libre et lance le signal
    I2C1_Wr(0x91); //Mode de contrôle en mode lecture
    MSB = I2C1_Rd(1); //Nombre signé [température entre +125° et -55°C]
    LSB = I2C1_Rd(0); //Si bit 7 = 1 température MSB +0,5°C
    90 //MSB = 0x25;
    //LSB = 0x80;
    I2C1_Stop(); //Arrêt du signal
    Delay_ms(10);
    // Écrire sur l'écran LCD.
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1, 5, "DS1631");
    Lcd_Out(2, 1, "Temp: ");

    IntToStr(MSB, msg); // Convertir la partie entière de la température en chaîne.
    100 Lcd_Out_CP(msg + 3); // ignorer les 3 blancs au début de la chaîne
    Lcd_Chrc_CP('.'); // Point décimal.
}

```

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

Repérer les différents octets présents sur la ligne SDA et en donner leurs significations ?

- Sur cet exemple quelle est la température présente sur le capteur DS1621 ?

Projet n°3 : lcd_ds1621maquette

Création d'un projet : lcd_ds1621maquette ranger dans le répertoire **Mes documents/TPcesi2021/Nom1-Nom2/lcd_ds1621maquette/****

Câblage supplémentaire : DS1621

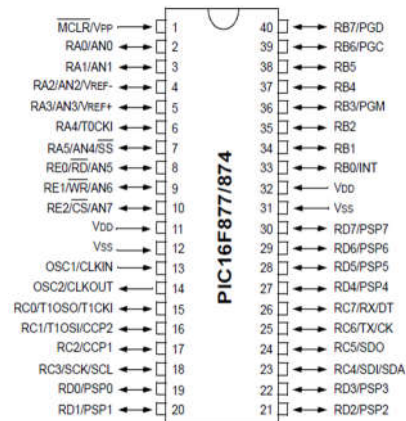
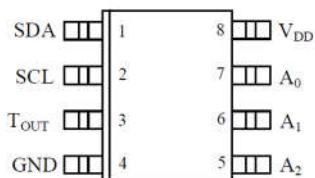
Alimentation : 5V pour VDD et 0v pour GND

Adresse physique à câbler d'après la programmation : 3 broches A₀A₁A₂

SDA et SCL à relier au PIC 16F877

DS1621 Digital Thermometer and Thermostat

PIN ASSIGNMENT



Recherche d'aide sur la programmation à partir :

- de la bibliothèque mikroC Pro
- de la notice technique DS1621:
- du programme donné ci dessus :

I2C Library
mikroC PRO for PIC Libraries > Hardware

Library Routines

- I2Cx_Init
- I2Cx_Start
- I2Cx_Repeated_Start
- I2Cx_Is_Idle
- I2Cx_Rd
- I2Cx_Wr
- I2Cx_Stop
- I2Cx_SetTimeoutCallback
- I2Cx_Set_Active

Generic Routines

- I2C_Start
- I2C_Restart
- I2C_Is_Idle
- I2C_Read
- I2C_Write
- I2C_Stop

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

Vérifier le bon fonctionnement du projet et faites valider.

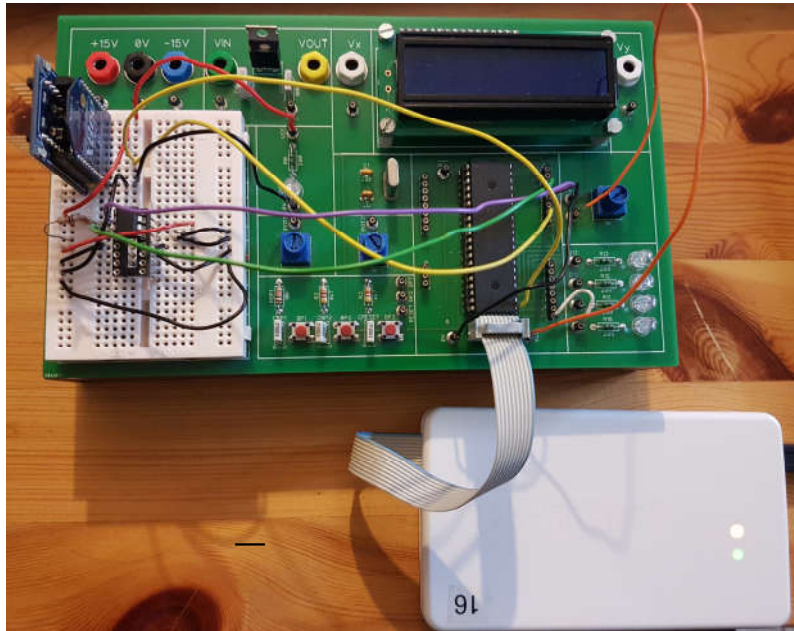
Projet n°4 : lcd_ds1621maquette en autonomie

On travaille exactement sur le même projet que précédemment.

But : Rendre le système autonome

Jusqu'à présent la maquette est alimentée via le programmeur. Le but est d'enlever cette connexion .

on alimente le montage via une pile de 9V et un régulateur de tension 7805 de telle sorte qu'on possède en sortie du régulateur une tension de 5V.



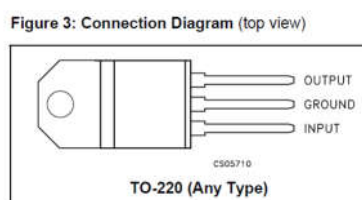
enlever le connecteur .

relier MCUMCLR à +VCC (pin MCR barré sur la maquette)

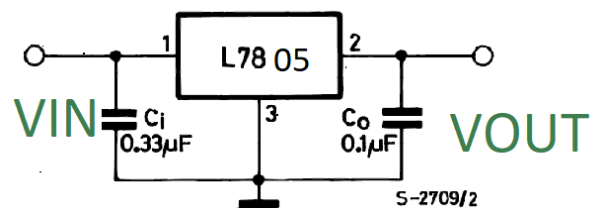
repérer sur la maquette le régulateur 7805 et étudier sa documentation constructeur. Câbler.



TO-220



Connecter la pile 9V .



Vérifier le bon fonctionnement et faites valider.

Sortir dehors avec la maquette et relever une dizaine de mesures de température jusqu'au régime permanent. En déduire le temps de réponse du capteur .

Carte de développement PIC 16F877 :

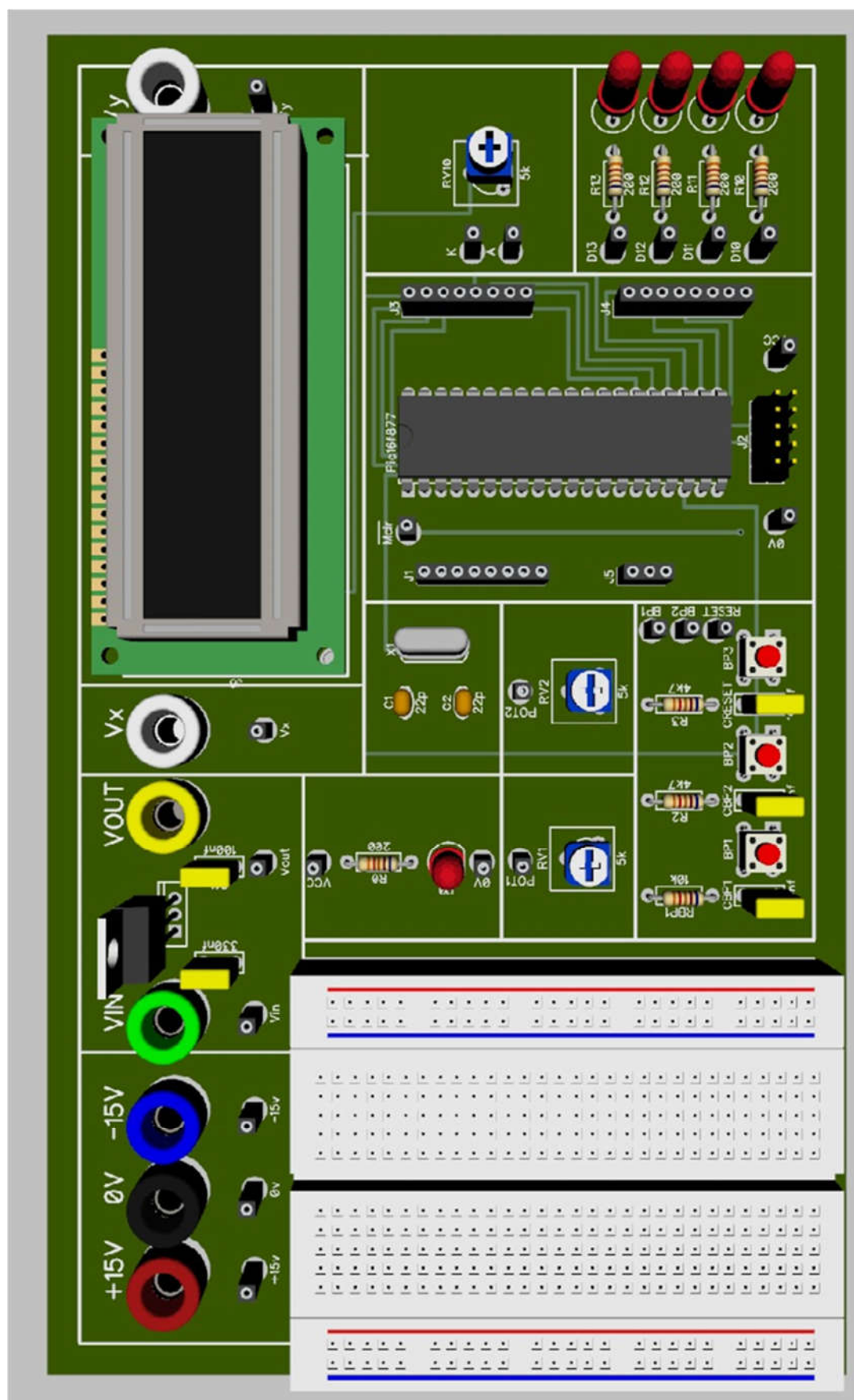
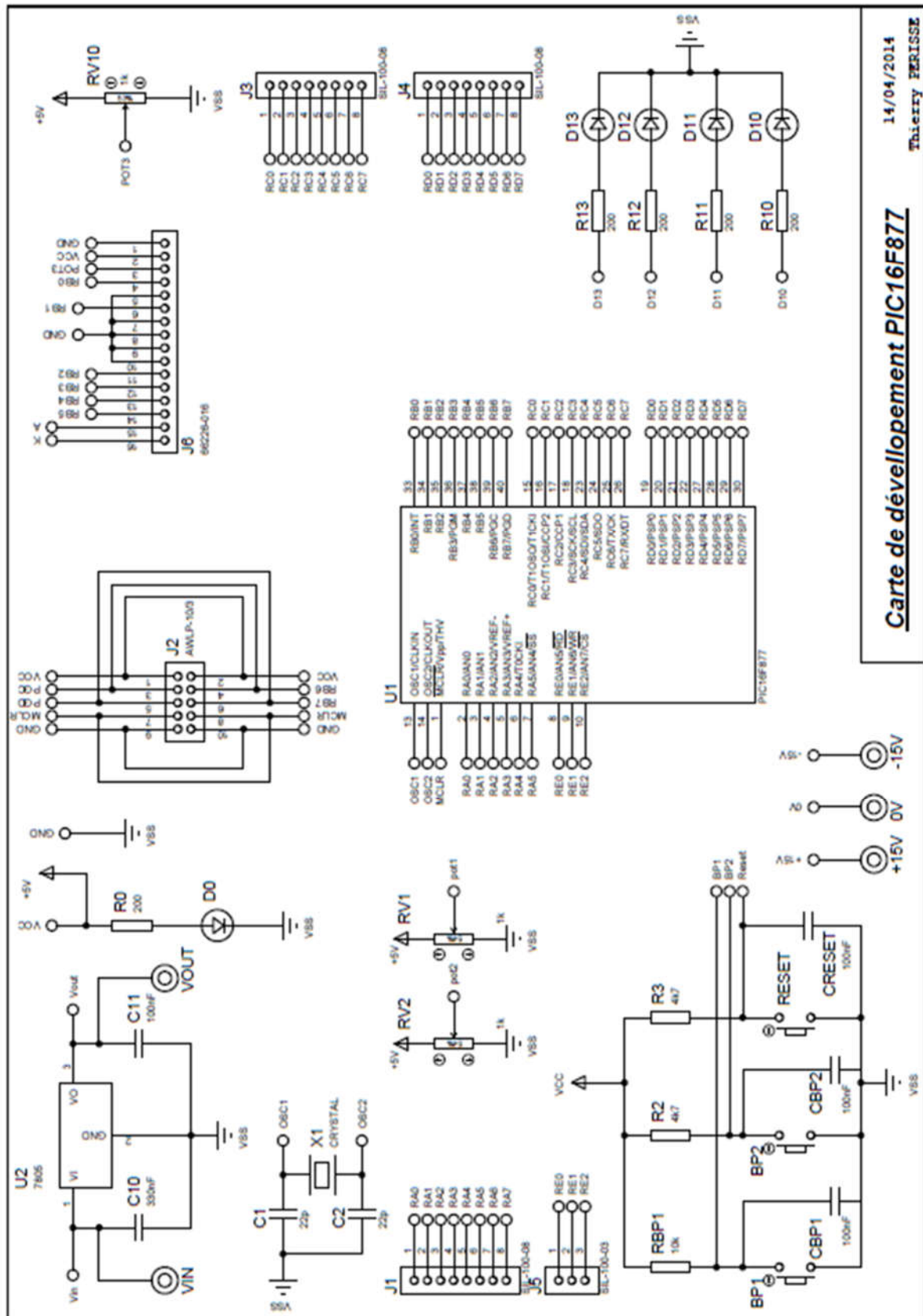


Schéma de la carte de développement PIC 16F877 :



14/04/2014
Thierry PERISSE

Carte de développement PIC16F877