TP DOMOTIQUE

CESE 6H

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.

TP1 b : Envoie d'informations à distance à l'aide d'émetteur /récepteur XBEE

Documentations :

Norme I2C / Capteur de Température I2C DS1621

Norme Zigbee / Modules Xbee

Carte développement PIC 16F877

Année 2016-17

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.(3h)

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***

TP1 b Envoie d'informations à distance à l'aide d'émetteur /récepteur XBEE (3h)

5. ***: Projet n°5 : Xbee1 Faire converser 2 modules Xbee.***

6. ***: Projet n°6 : testXbeebonjour***

7. ***: Projet n°7 : Projet complet cad envoie à distance de l'info température***

   (partie émission : capteur I2C DS1621, PIC16F877, Xbee_émetteur, autonomie (accu+régulateur)

   (partie réception : Xbee_récepteur, interface Xbee/miniUSB, Terminal de X-CTU)

8. ***: Projet n°8 : Consommation // Portée // Interférences***

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_DS1621autonome.**

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

Préparation Projet n°6 : **testXbeebonjour**

Donner les caractères présents sur les différentes trames

Préparation Projet n°7 : Projet complet d'envoi à distance de Temp

A l'aide des différents programmes validés : Réaliser le programme complet du projet 7 ?

Préparation Projet n°8 : Consommation // Portée // Interférences

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 6h / 8h)

TP 1-a

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 (programmateur PicFlash) un microcontrôleur (PIC16F877) avec un langage de haut niveau (langage C).

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

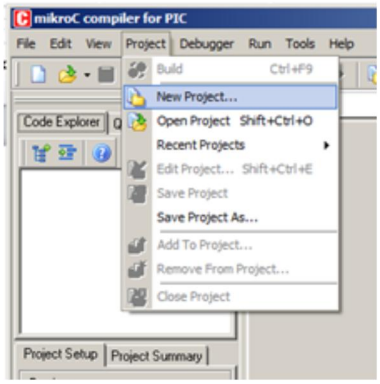
Lancer le Logiciel MicroC présent sur le bureau du PC :

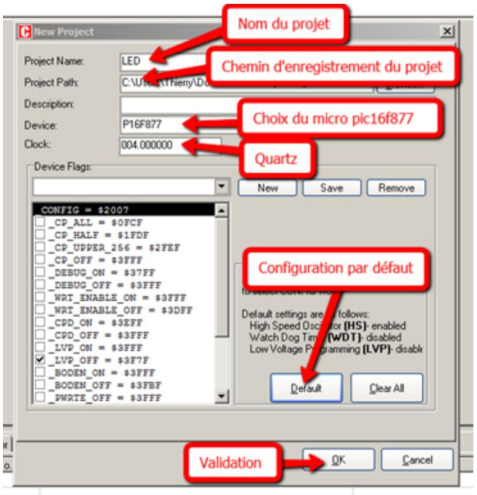
Projet n°1 : LED

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/***



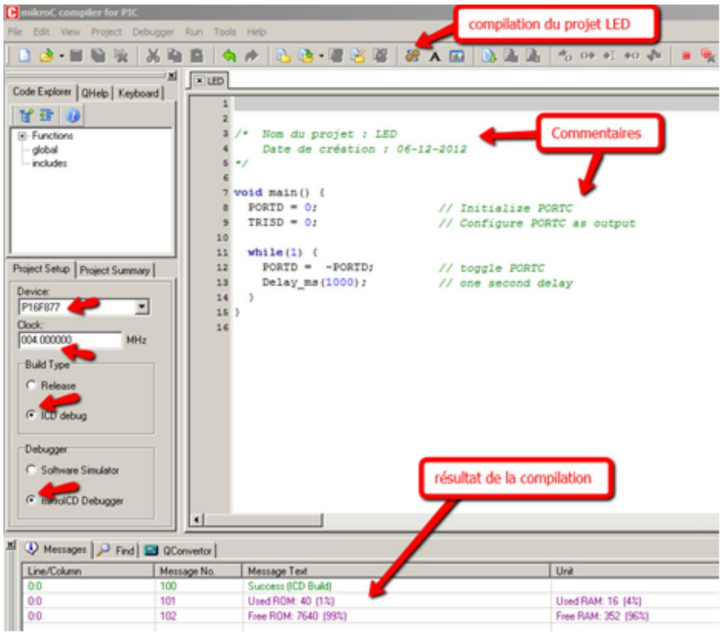


Recherche d'un programme exemple : Clignotement d'une LED.(chargement de Led_Blinking.c)

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

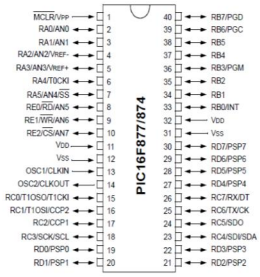
Copier / coller le contenu de ce fichier sur votre projet (ici LED)

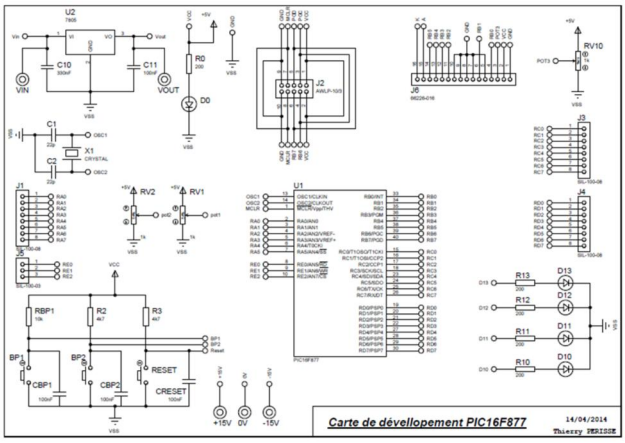
Le programme utilise le port C hors nous voulons faire clignoter une LED sur le port D apporter les modifications si nécessaires et compiler le fichier.

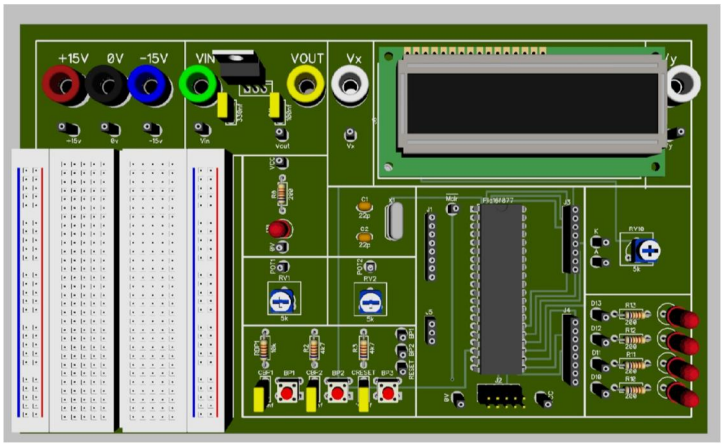


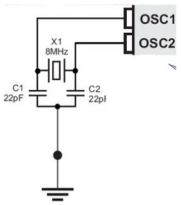
Mettre vos propres commentaires sur le programme.

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





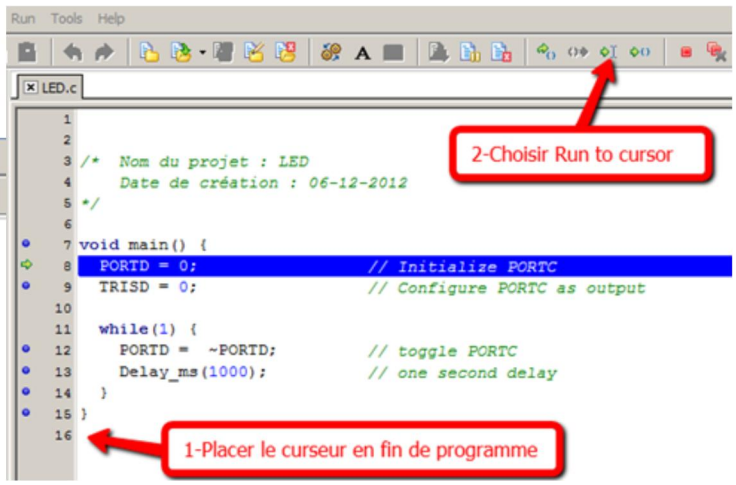




Branchons maintenant le programmateur PICFlash sur le connecteur afin de transférer le programme dans le microcontrôleur PIC16F877.

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.

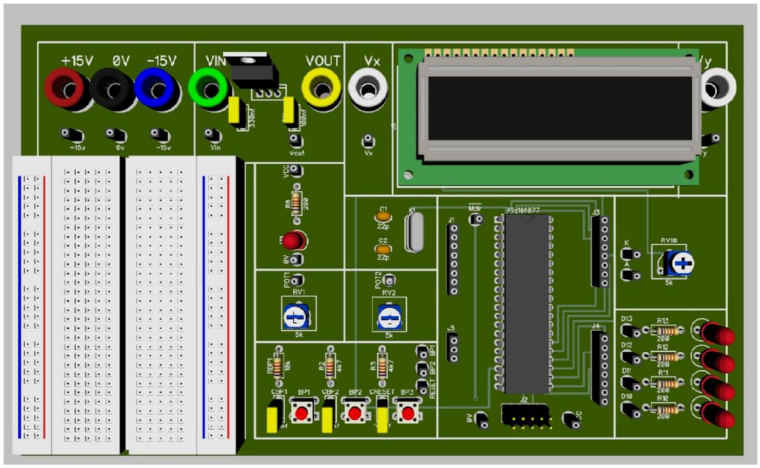
En utilisant les routines du LCD (voir ci-dessous) :
Justifier les lignes de codes du programme ci-contre.

LED_et_LCD.c

```
/*   Nom du Projet :     LED_et_LCD
      Date de création :  06-12-2012
      PIC / 16F877
*/

char *text = "Salut";

void main() {
  Lcd_Config(&PORTB, 0, 1, WR, 5, 4, 3, 2);
  Lcd_Init(&PORTB);                        // Initialize LCD connected to PORTB
  Lcd_Cmd(Lcd_CLEAR);                      // Clear display
  Lcd_Cmd(Lcd_CURSOR_OFF);                 // Turn cursor off

  PORTD = 0;                              // Initialize PORTC
  TRISD = 0;                               // Configure PORTC as output

  while(1) {
    PORTD =  ~PORTD;                       // toggle PORTD
    Delay_ms(1000);                        // one second delay
    Lcd_Out(1, 1, text);                   // Print text to LCD, 2nd row, 1st column
  }
}
```

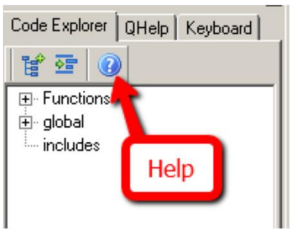
configuration LCD

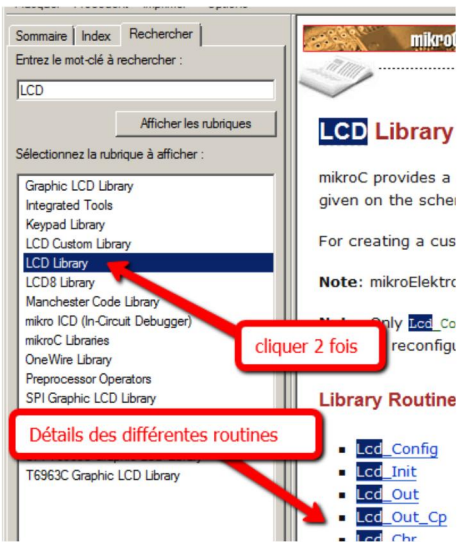
Aucun câblage supplémentaire n'est nécessaire pour tester le fonctionnement du LCD.



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

Rechercher / Taper LCD / Afficher les rubriques / LCD Library

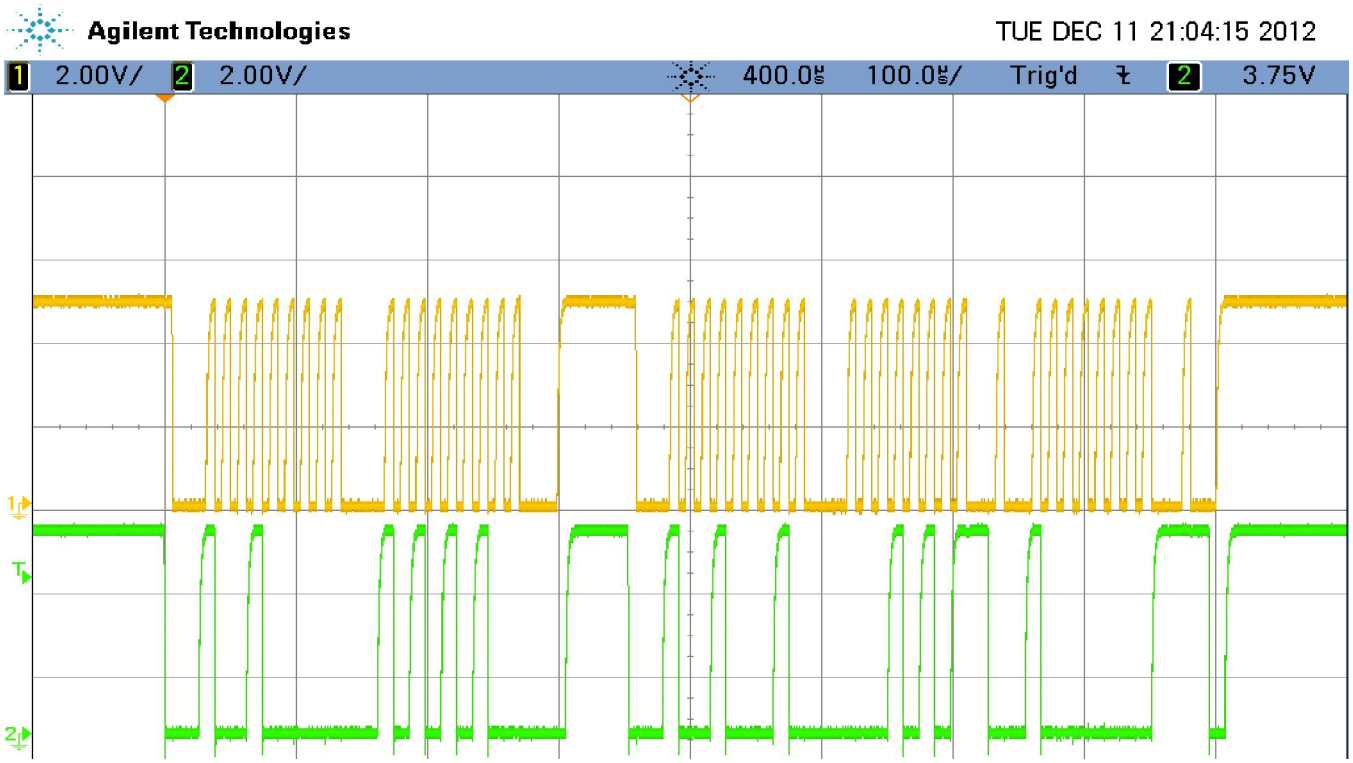




Compiler le programme, transférer le prog dans le microcontrôleur, lancer le prog et valider le bon fonctionnement du projet LED_LCD.

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

```
/******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 sigal
     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 sigal
     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 chaine de caractère temptxt sur le LCD
delay_ms(100);
lcd(2, 1, 7, temp);   //Affiche la chaine de caractère temp sur le LCD
delay_ms(100);
PORTD = ~PORTD;       // toggle PORTD
delay_ms(1000);
     }
```

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

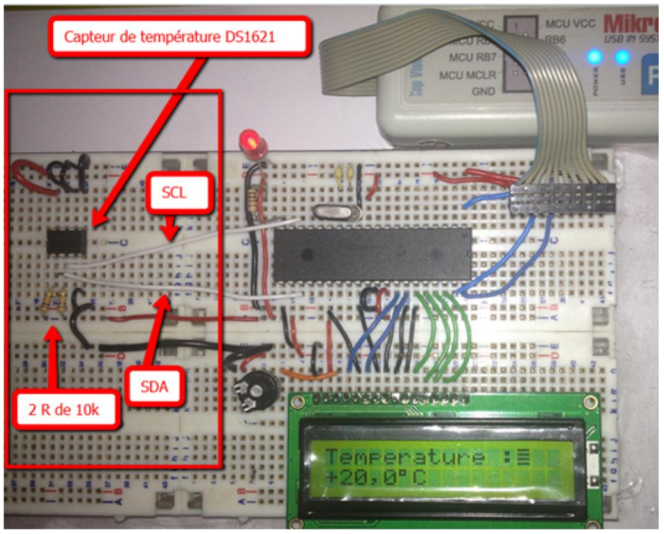
Création d'un projet : LED_LCD_DS1621 ranger dans le répertoire Mes documents/TPDomotique2014/LED_LCD_DS1621/

Câblage supplémentaire : DS1621

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

Adresse physique : 3 broches $A_0A_1A_2$

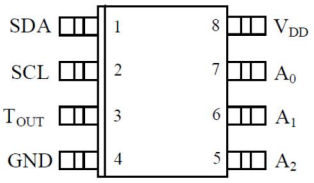
SDA et SCL relier au PIC 16F877



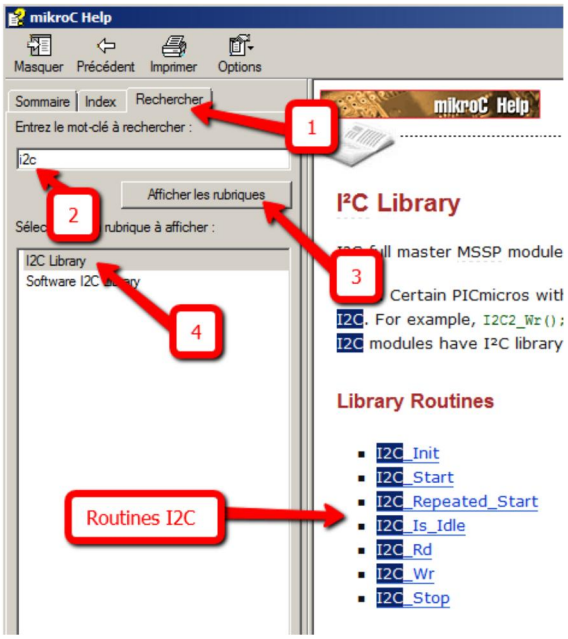
DS1621

Digital Thermometer and Thermostat

PIN ASSIGNMENT



Recherche d'aide sur la programmation de composants I2C :



A l'aide du programme donné en annexe LED_LCD_DS1621 :

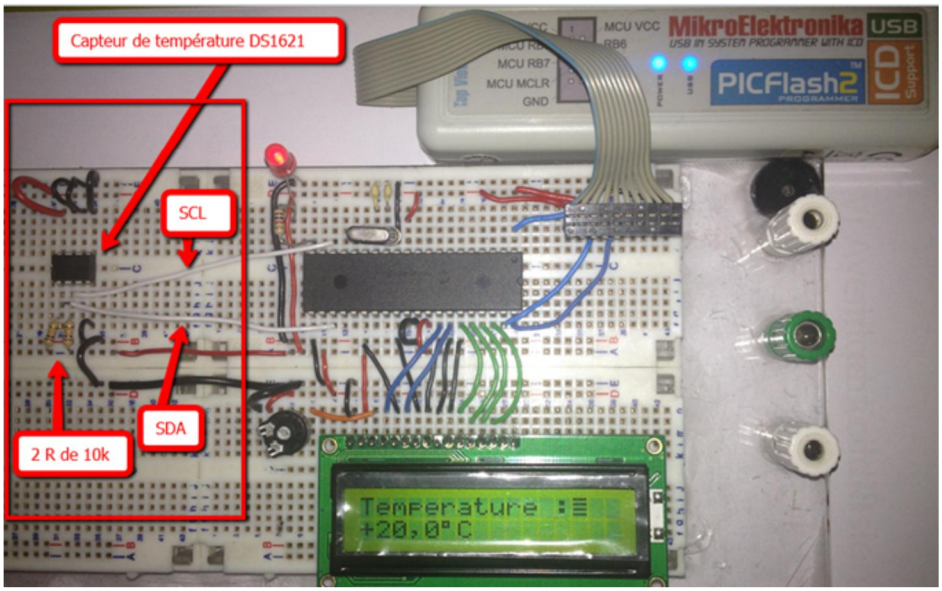
Repérer et commenter les lignes de commande concernant l'I2C.

Valider le bon fonctionnement du projet.

Projet n°4 : LED_LCD_DS1621autonome

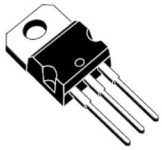
Création d'un projet : (On travaille exactement sur le même projet que précédemment)

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

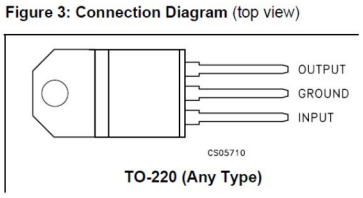


Rendre le système autonome : on alimente le montage via une pile de 9V et un régulateur de tension 7805, le but étant d'enlever la connexion avec le PC.

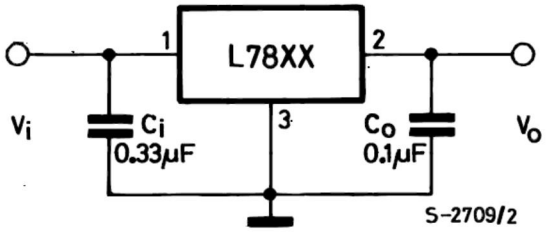
Voir aussi la doc constructeur du 7805 :



TO-220



TO-220 (Any Type)



Pour utiliser le µP en mode autonome :

cocher release,

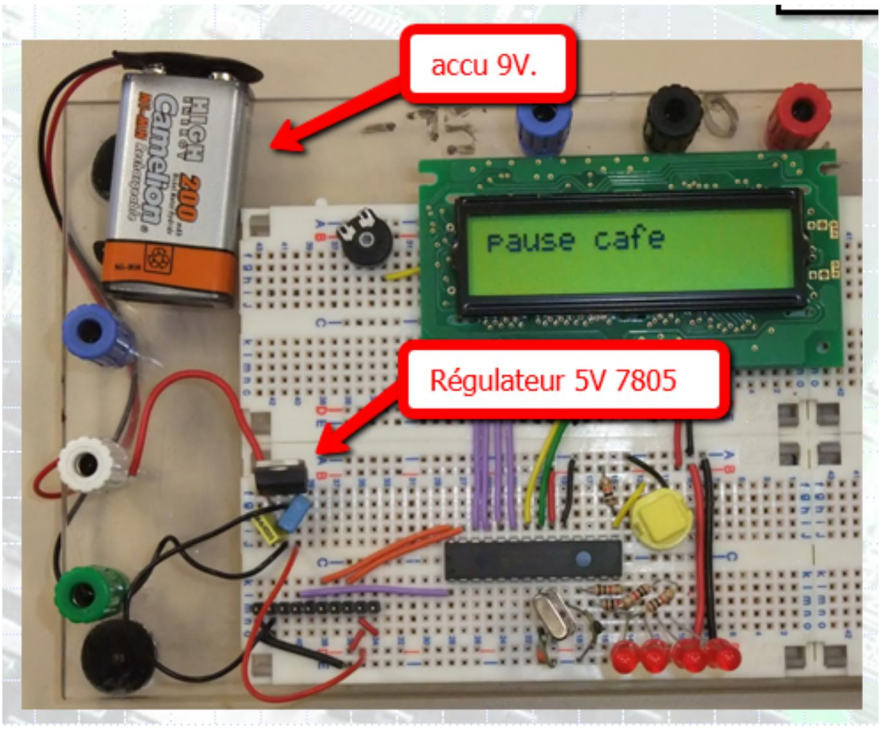
compiler,

programmer.

enlever le connecteur .

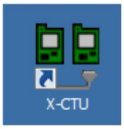
relier MCUMCLR à +VCC

Vérifier le bon fonctionnement.

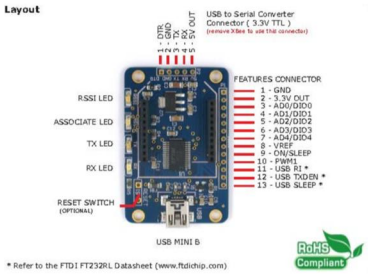


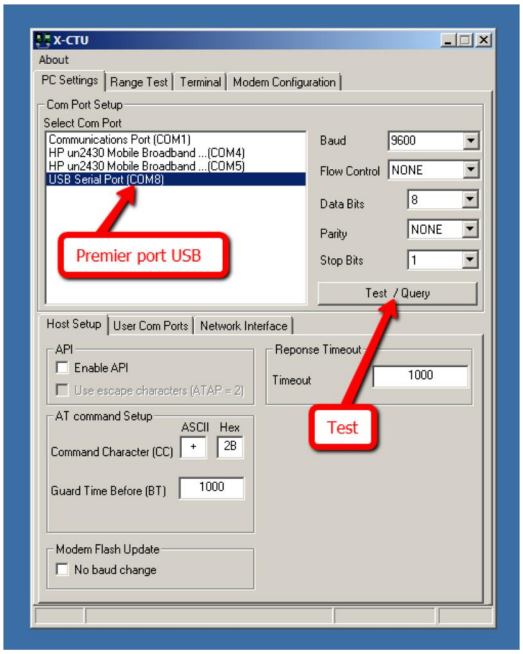
Projet n°5 : Xbee1 : Faire converser 2 modules Xbee

Lancer le logiciel X-CTU présent sur le bureau

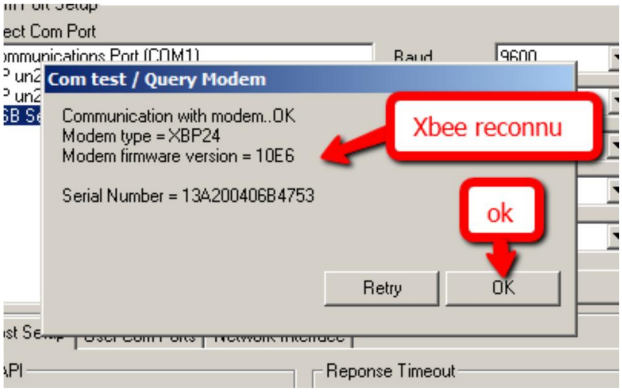


A l'aide d'un cordon USB/miniUSB et d'un interface USB/Xbee relier un module Xbee au PC.



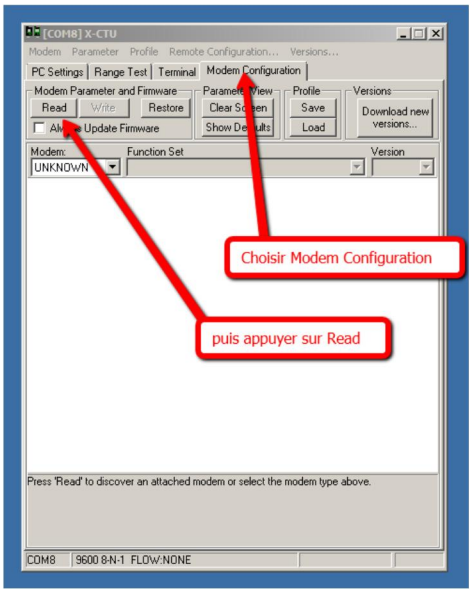


Choisir le port USB et faire un test sur la communication : Test/Query

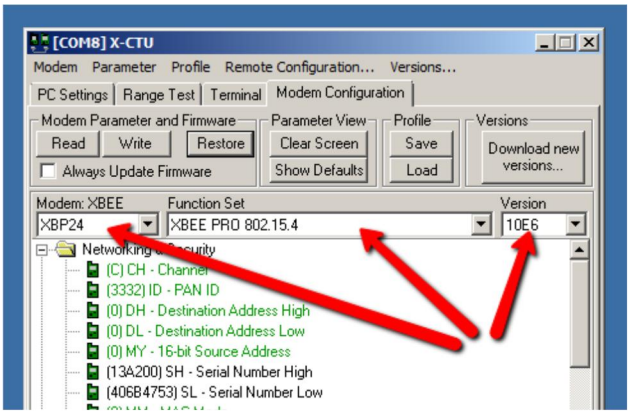


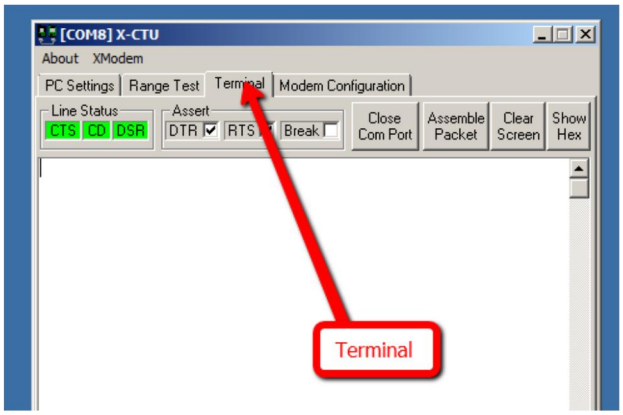
Lire les différents paramètres sur READ :

Si Xbee déjà utilisé Read / Restore / Write / Read
(On met le Xbee en configuration usine)

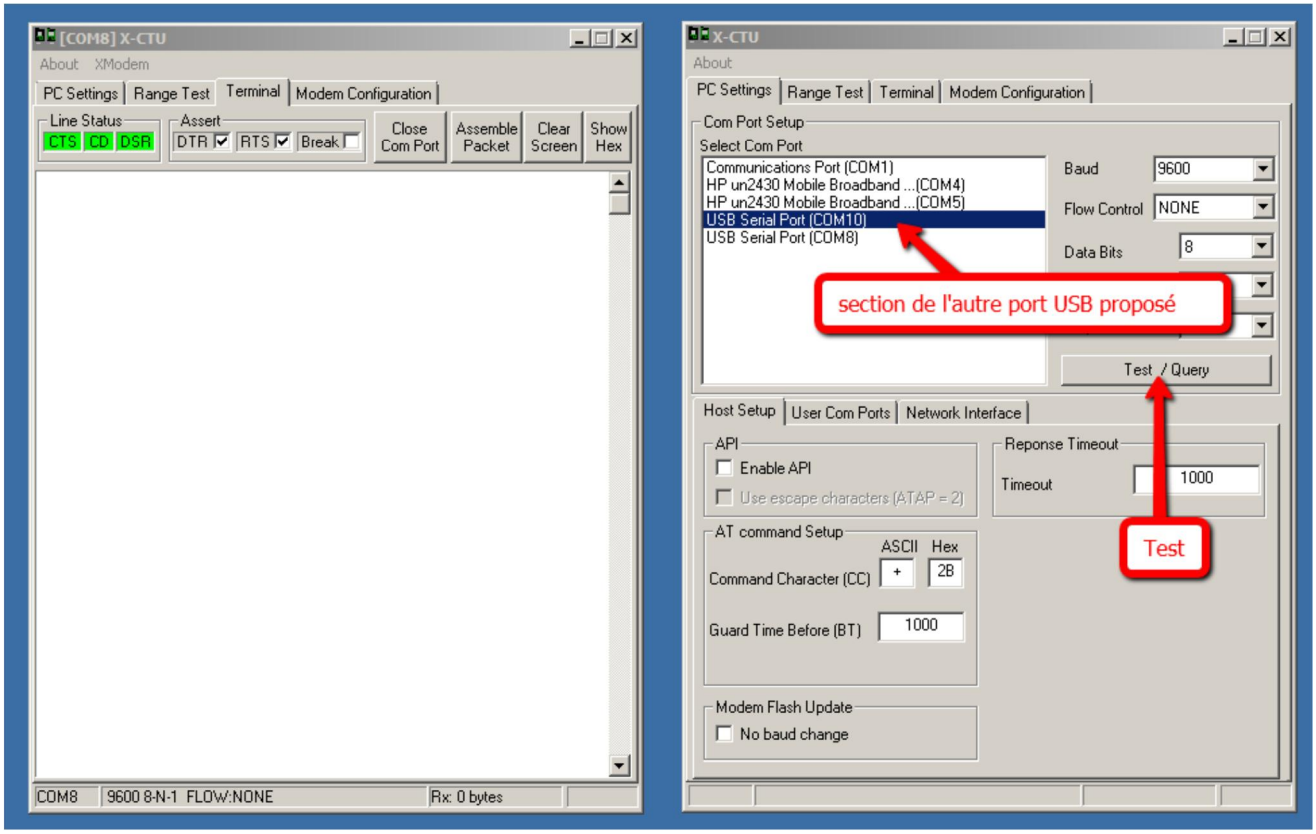


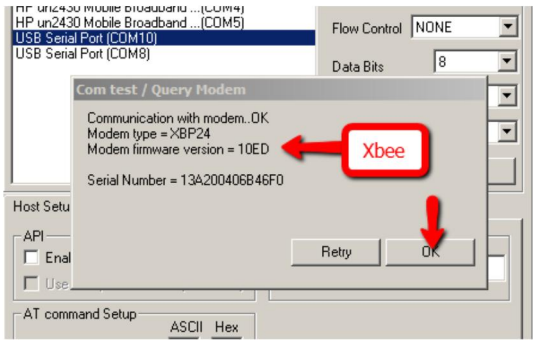
Résultat de Xbee1 et position d'attente sur Terminal :

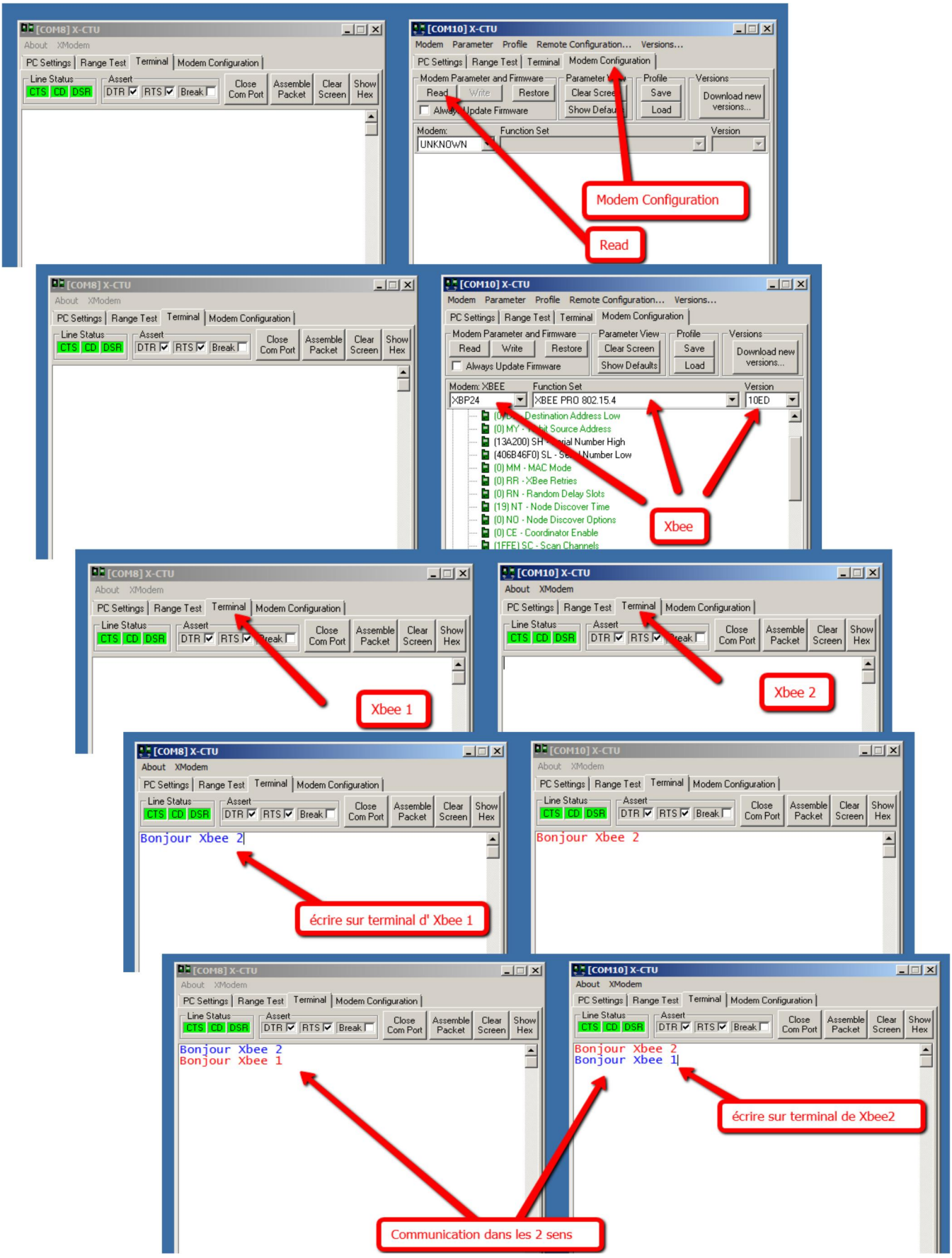




Faisons la même chose avec l'autre module Xbee2





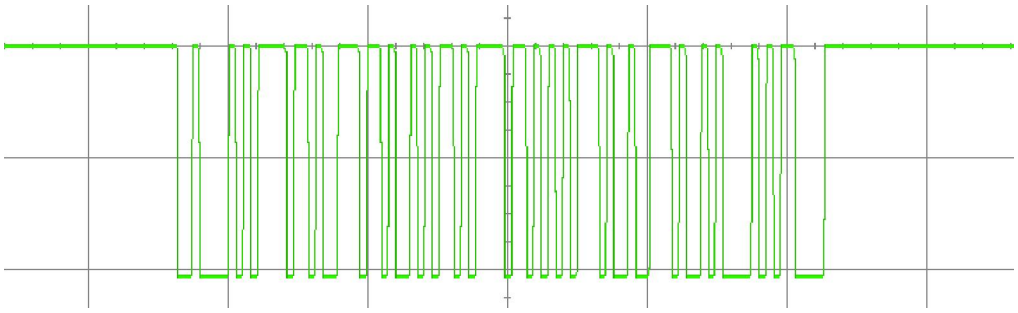


Vérifier le bon fonctionnement du projet 5.

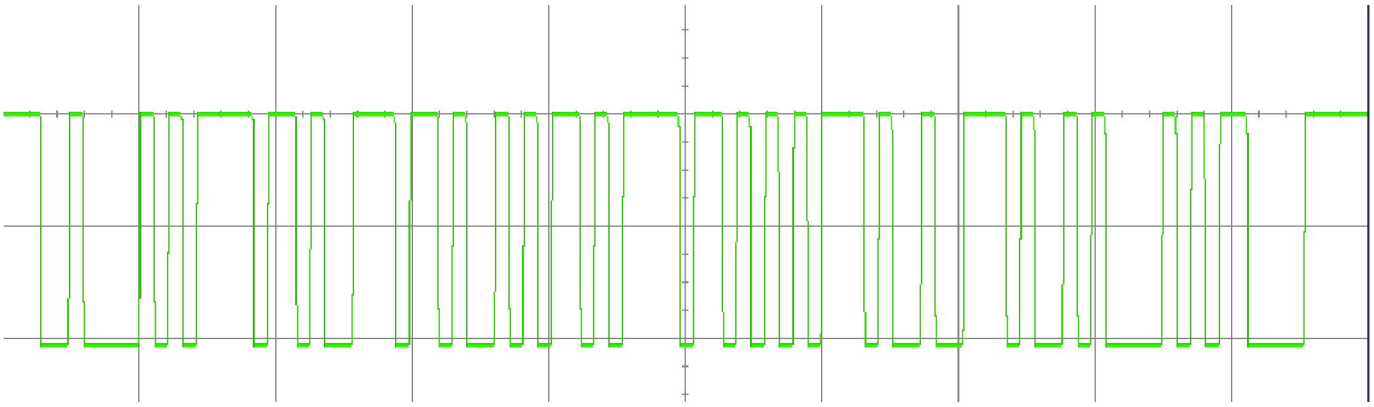
Préparation Projet n°6 : testXbeebonjour

Envoie de « bonjour » sur la broche Tx du microcontrôleur 16F877 :

2ms/div



1ms/div

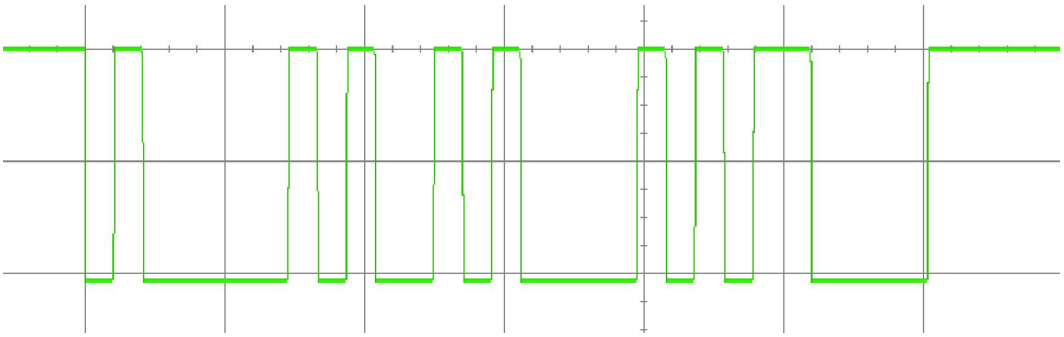


A l'aide de la norme montrer les différents mots présents sur cette trame ?

Faire la même chose sur les 2 trames suivantes ? Quels sont les caractères transférer sur ces trames ?

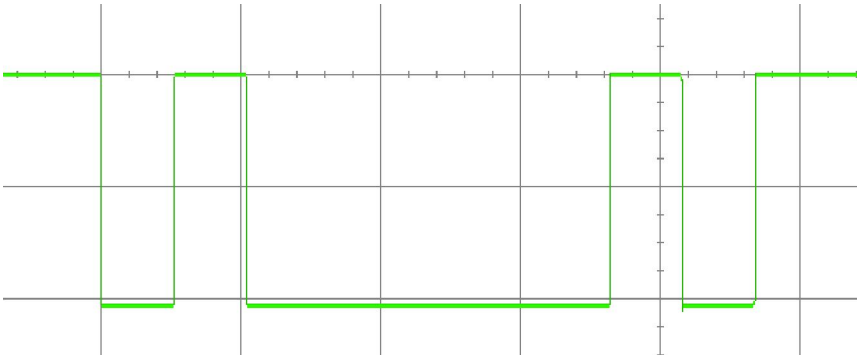
500us/div.

Trame 1



200us/div.

Trame2



Projet n°6 : testXbeebonjour

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

Envoyer « Bonjour » avec un émetteur Xbee1 relier au PIC16F877 (Din du Xbee1 // TX du PIC16F877) et récupérer l'info avec un Xbee2 monté sur un adaptateur Xbee/miniUSB via Terminal de X-CTU.

Hard :

Câbler l'alimentation du Xbee1 Vin=5V. et la masse avec celle du reste du montage.

Tx du PIC16F877 relié au Din du Xbee1.

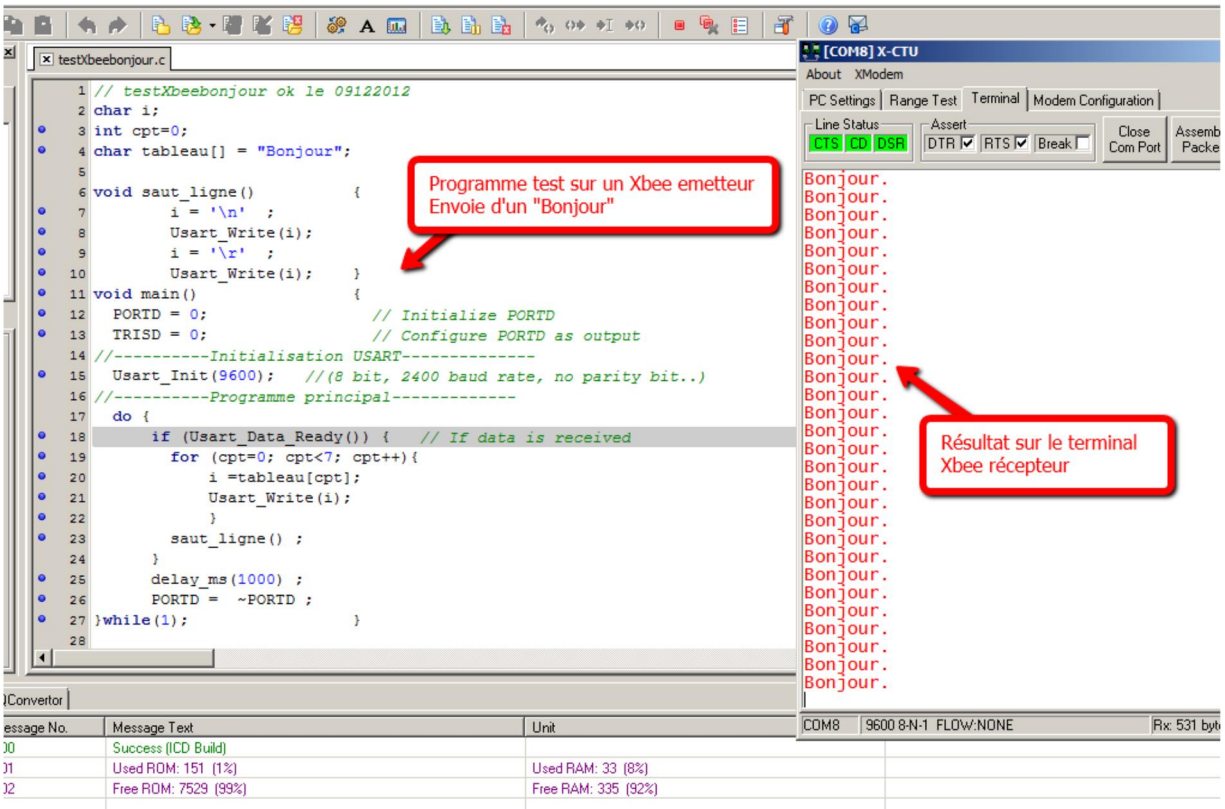
Soft :

Faire un copier / coller du programme testXbeebonjour.c

Compiler

Transférer

Lancer le programme.



Faire vérifier le bon fonctionnement de ce projet.

Projet n°7 : Projet complet d'envoi à distance de l'info température.

(partie émission : capteur I2C DS1621, PIC16F877, Xbee_émetteur, autonomie (accu+régulateur)

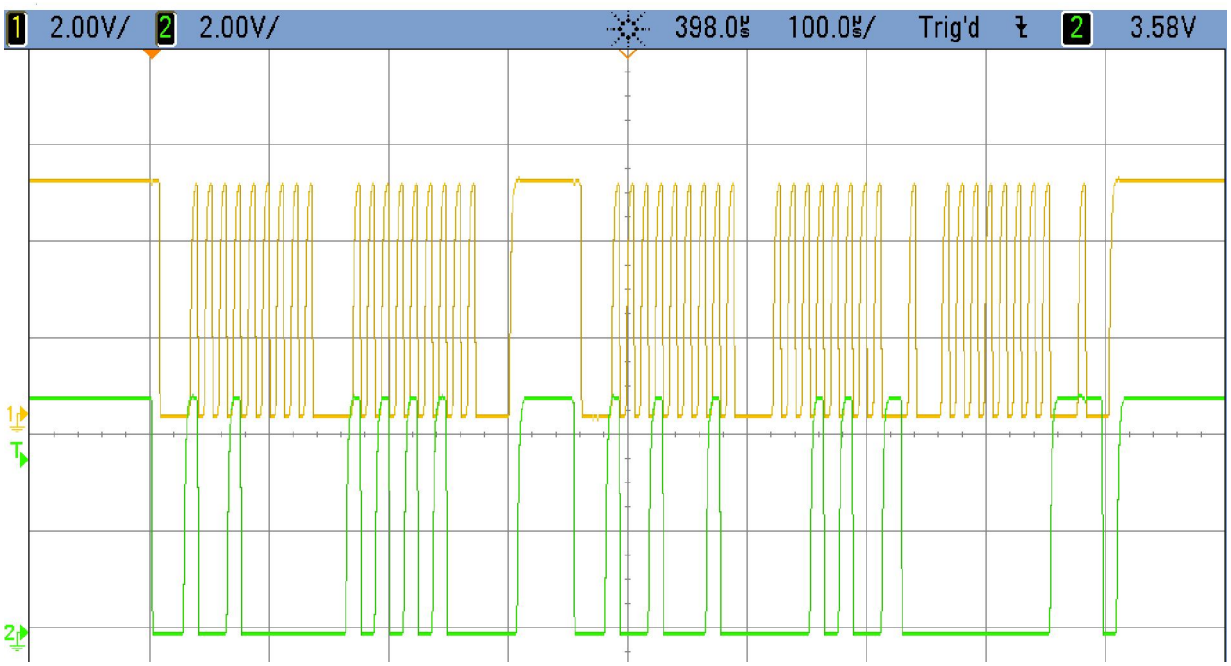
(partie réception : Xbee_récepteur, interface Xbee/miniUSB, Terminal de X-CTU)

A l'aide des différents programmes validés; Réaliser le programme complet du projet 7 ?

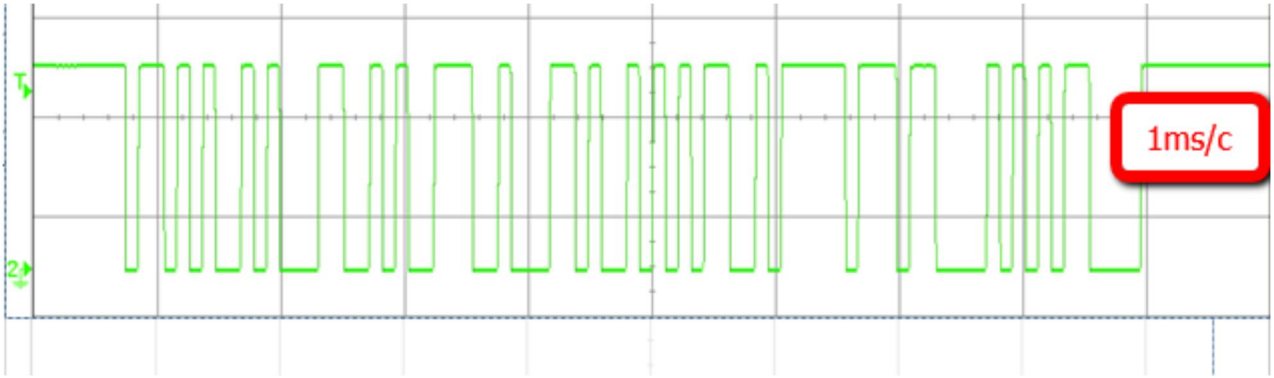
Récupération des voies SDA et SCL à l'oscilloscope.

Récupération du signal présent sur la broche Din du Xbee_émetteur (ou du Tx du 16F877).

Ci-dessous un exemple de signaux SDA et SCL.



et un exemple de signal Tx.



Valider le bon fonctionnement du projet complet.

Projet n°8 : Consommation // Portée // Interférences

A-Faire tourner le projet 7 en autonome (Xbee normal)

Récupérer et faire tourner le programme : LED_LCD_DS1621_UART (site web ou répertoire)

A-1 Bilan énergétique de la partie émission :

Mesurer la consommation, calculer la durée de fonctionnement

- *donner le schéma de câblage permettant de visualiser l'image du courant (Rshunt 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.*

A-2 Etude de portée :

***Mesure de la portée Emission // Réception** (Sans obstacle, avec obstacle)*

A-3 Interférences (portable, wifi, …)

B-Faire tourner le projet 7 en autonome (Xbee veille)

B-1 Programmation du Xbee en veille :

- *Pour que le Xbee puisse se mettre en veille il faut le programmer à partir du logiciel X-CTU avec Sleep mode (1) cad Hibernate.*

B-2 Bilan énergétique de la partie émission :

Récupérer et faire tourner le programme : Xbeeveille (site web ou répertoire)

- *Repérer les lignes permettant la mise en veille du Xbee, proposer le câblage de la broche DTR du Xbee.*

Mesurer la consommation, calculer la durée de fonctionnement

- *Garder le schéma de câblage permettant de visualiser l'image du courant (Rshunt 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.*

Carte de développement PIC 16F877 :

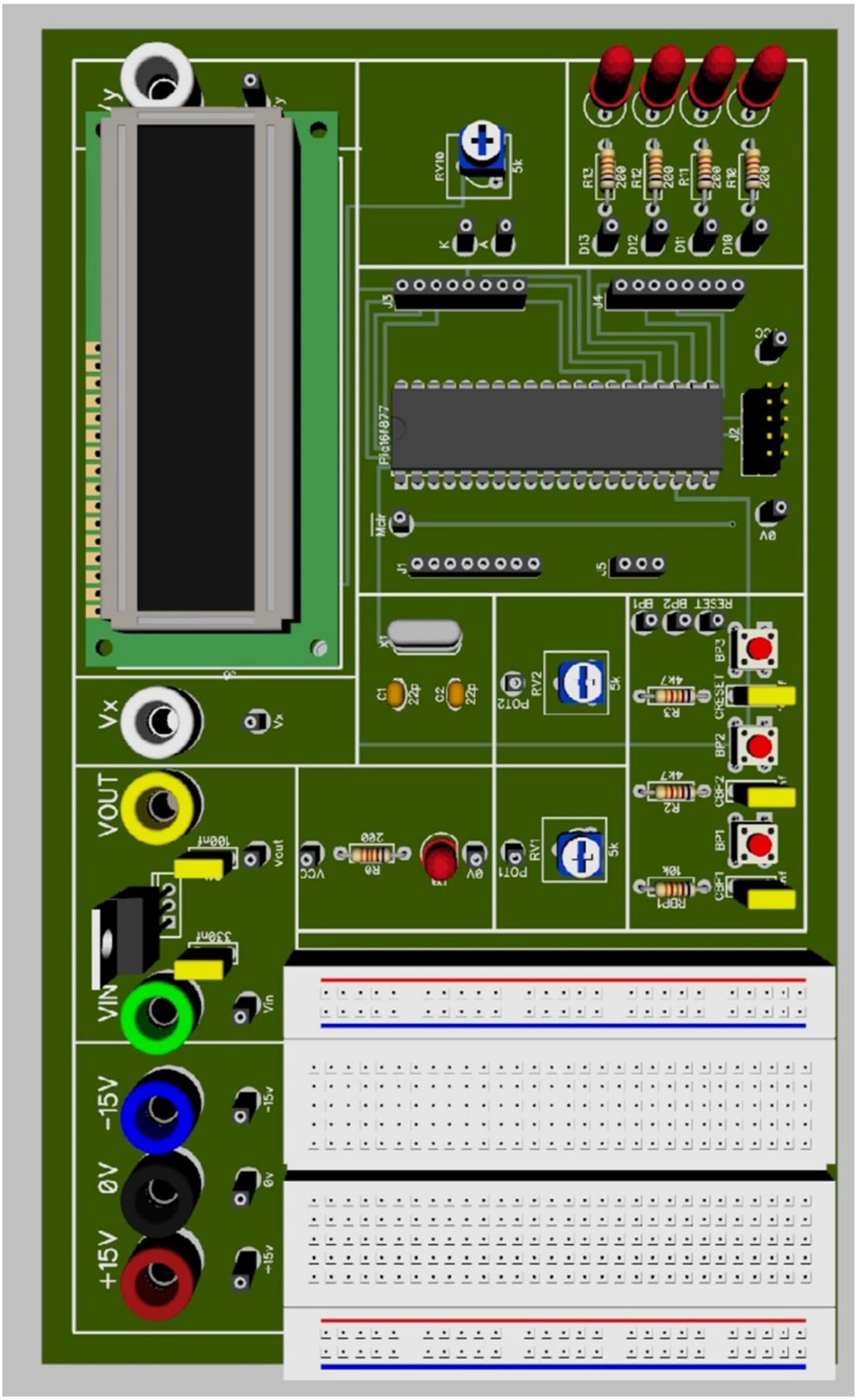


Schéma de la carte de développement PIC 16F877 :