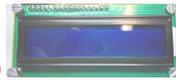


Liste des documents ressources :

- Cours Liaison série & I2C
- Schéma du shield LS-I2C
- <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>
- <http://www.robot-electronics.co.uk/htm/srf02techSer.htm>
- Datasheet DS1621 (capteur de température)



1. Afficheur LCD (I2C)

- 1.1. Télécharger sur la page <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library> le fichier « zip ». Suivre les instructions d'installation et essayer un programme d'exemple (l'adresse de votre afficheur LCD est 0x3F).
- 1.2. Ecrire un programme qui affiche :
 - Sur la première ligne, les secondes écoulées depuis le lancement du programme.
 - Sur la deuxième ligne, l'état du BP présent sur la carte « Appui » ou « Relâchement »



2. Capteur de température DS1621 (I2C)

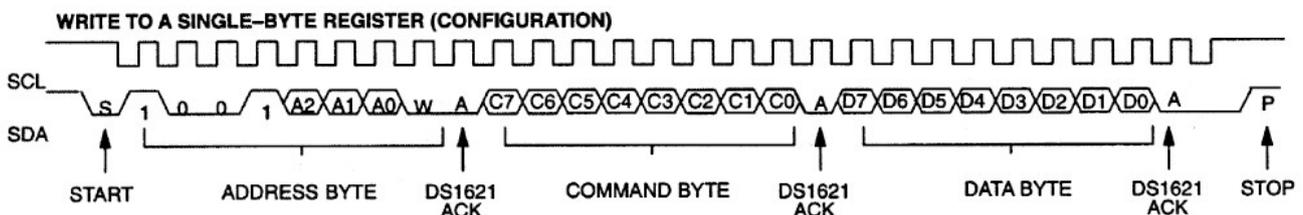
- 2.1. D'après la datasheet du DS1621, quelles sont les valeurs des bits de poids fort (A6:A3) de l'adresse du DS1621.

A6	A5	A4	A3	A2	A1	A0

- 2.2. D'après le schéma du shield LS-I2C, Où sont connectées les pins A2:A0 ? Proposer les lignes de code permettant de fixer la valeur 001 sur A2:A0.
- 2.3. A partir de la datasheet du DS1621, indiquer la valeur à positionner sur le bit 1SHOT du registre de configuration du capteur de température.
- 2.4. Sur ce même registre quel est le bit indiquant qu'une conversion est terminée ?
- 2.5. Quelles sont les valeurs hexadécimales des commandes :

Command	Value
Access Config	
Start Convert T	
Read Temperature	

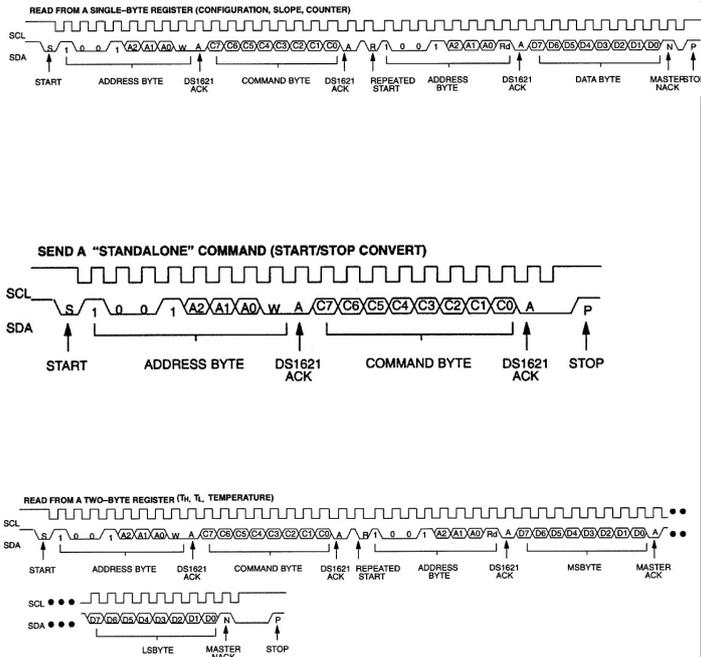
Voici la séquence d'initialisation du registre de configuration du DS1621.



- 2.6. Voici le code correspondant à la séquence ci-dessus. Compléter-le et placer-le dans le « setup » de votre programme.

```
Wire.beginTransmission(.....); //Adresse DS1621          ADDRESS_DS1621
Wire.write(.....);              //commande Access Config
Wire.write(.....);              //valeur registre config (1shot)
Wire.endTransmission();
```

2.7. Créer une fonction `float acqTempDS1621(void)`; permettant de lire la température du DS1621.



```

/* Lancement de la conversion */
Wire.beginTransmission(.....);
Wire.write(.....);
Wire.endTransmission();

/* Attendre la fin de la conversion */
do {
Wire.beginTransmission(.....);
Wire.write(.....);
Wire.endTransmission(false); // Condition RESTART
Wire.requestFrom(....., .....); // Un octet est requis
i++;
if (1 <= Wire.available()){
endConversion = Wire.read() & 0x.....;
}
} while (!endConversion || i>5 );

/* Récupérer les deux octets de température */
if(endConversion !=0) {
Wire.beginTransmission(.....);
.....; //commande READ_TEMPERATURE
.....; // Condition RESTART
.....; // Deux octets sont requis
if (Wire.available() >=2) {
.....; // lecture 1er octet
.....; // lecture 2ieme octet
}

/* Conversion en float */
.....
.....
.
.
return temperature;
}
else {
return -280;
}
}

```

2.8. Ecrire un programme qui, dans le programme principal à chaque appui du bouton poussoir, lance votre fonction `acqTempDS1621` et affiche le résultat sur la 1ere ligne de l'écran LCD.



3. Télémètre à ultrason SRF02

- 3.1. D'après le schéma du shield, identifier sur quelles pins sont connectées la liaison série du capteur SRF02.
- 3.2. Créer une liaison série software ($Tx_{\text{arduino}} \leftrightarrow Rx_{\text{SRF02}}$ et $Rx_{\text{arduino}} \leftrightarrow Tx_{\text{SRF02}}$).
- 3.3. D'après la page web <http://www.robot-electronics.co.uk/hm/srf02techSer.htm>, concevoir la fonction `unsigned int acqDistSRF02()`; permettant l'acquisition de la distance et renvoyant cette valeur.

4. Programme complet

Utiliser les deux fonctions `float acqTempDS1621()` et `unsigned int acqDistSRF02()` pour afficher sur appui du bouton les infos de distance et température.