

# ACTIVITÉ PRATIQUE : CAPTEUR DE TEMPÉRATURE I2C SUR ARDUINO UNO

## Objectifs :

- x Mettre en œuvre une liaison série I2C entre une carte Arduino et un capteur de température
- x Savoir analyser le fonctionnement d'un composant à partir de sa documentation technique
- x Comprendre et visualiser la sérialisation des données dans une transmission série

## Prérequis :

- x Savoir simuler le comportement d'une carte Arduino et son programme
- x Connaître le protocole I2C (Fiche de connaissances)
- x Maîtriser la conversion des nombres décimaux en binaire et en hexadécimal

## 1 Rappels concernant le bus I2C

- x Ce bus utilise trois lignes pour communiquer: \_\_\_\_\_, \_\_\_\_\_ et la \_\_\_\_\_.  
Puisqu'il utilise une horloge et une ligne de donnée, ce bus est un bus \_\_\_\_\_.
- x La transmission se fait de la manière suivante:
  1. **Condition de départ** : Le maître passe SDA à l'état bas quand SCL est à l'état haut.
  2. **Transmission de l'adresse** du récepteur sur 7 bits (valeur possible de 0 à 127), le **8ème bit** indique si le maître fait une opération
    - x de \_\_\_\_\_ (niveau \_\_)
    - x ou \_\_\_\_\_ (niveau \_\_).
  3. **Acquittement** du récepteur. Celui-ci force un état bas sur la ligne SDA sur la 9ème impulsion d'horloge générée par le maître. Si le récepteur n'acquiesce pas, la ligne SDA reste au niveau 1 à cause de la résistance de rappel au +Vcc.
  4. Transmission d'un octet de **donnée** (bit de poids fort en premier).
  5. **Acquittement** du récepteur. Le maître peut alors continuer à transmettre ses données.
  6. **Condition d'arrêt**: Le maître force le signal SDA à l'état haut quand SCL est à l'état haut.

## 2 Étude du capteur de température DS1621 :

Afin d'utiliser ce capteur, nous allons commencer par étudier sa documentation technique.

- x Précisez la gamme de température mesurable par le DS 1621:

- x Précisez la gamme d'alimentation acceptable par ce circuit:

- x La température est renvoyée sous quelle forme?

- x Complétez le tableau suivant:

BROCHE	SYMBOLE	DESCRIPTION
1	SDA	
2	SCL	
3	TOUT	
4	GND	
5	A2	
6	A1	
7	A0	
8	VDD	

### 2.1 Codage de la température

- x Le tableau suivant présente le code délivré en fonction de la température :

**Table 2. TEMPERATURE/DATA RELATIONSHIPS**

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	01111101 00000000	7D00h
+25°C	00011001 00000000	1900h
+½°C	00000000 10000000	0080h
+0°C	00000000 00000000	0000h
-½°C	11111111 10000000	FF80h
-25°C	11100111 00000000	E700h
-55°C	11001001 00000000	C900h

- x En remarquant que le code utilisé est le code complément à deux, retrouver le code binaire associé à ces deux températures :

+15,5°C:

-8,0°C:

- x Retrouver les températures correspondant aux code binaires suivant :

0001 1101 0000 0000 :

1011 1100 1000 0000 :

- x D'après la documentation, que faut-il garder pour faire une lecture de la température au degré près ?

## 2.2 Analyse de la communication en I2C

- x Donner les deux modes de fonctionnement du DS1621.

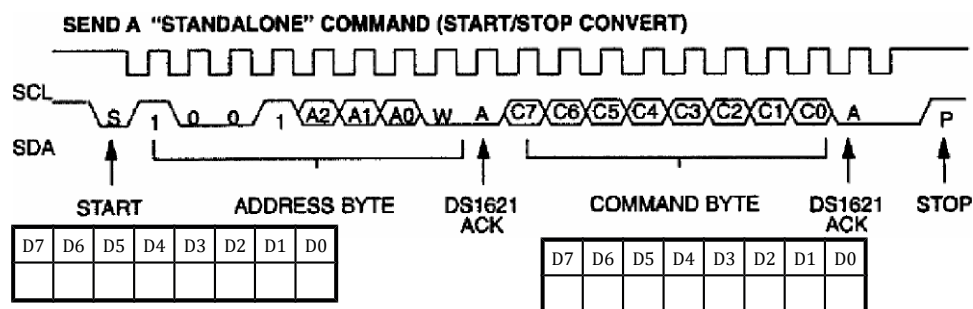
x mode 1 : \_\_\_\_\_

x mode 2 : \_\_\_\_\_

- x Donner l'adresse I2C générique (en fonction de A2A1A0) d'un DS1621 : \_ \_ \_ \_ \_

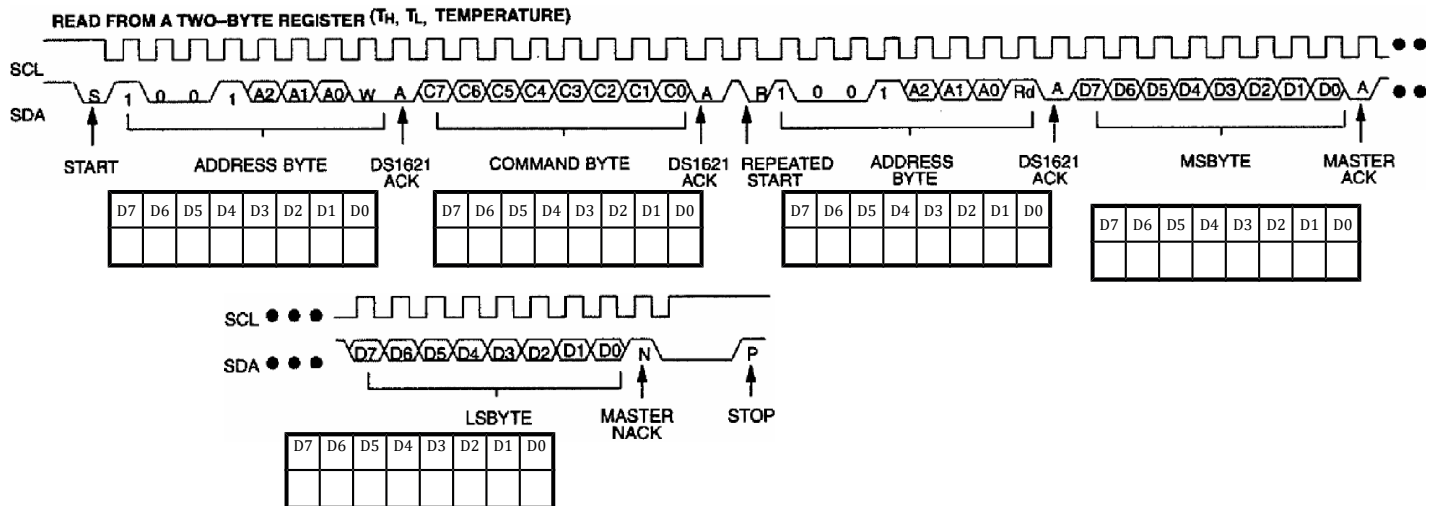
En supposant que le DS1621 est configuré en mode « conversion unique sur demande ». Pour lire une température dans ce mode il faut donc envoyer tout d'abord l'ordre « **Start Convert T** »,

- x Compléter avec les valeurs adéquates les tableaux des chronogrammes suivants.



Dans un second temps, il faut envoyer l'ordre « **Read Temperature** ».

✗ Complétez les deux premiers octets à envoyer ci-dessous.



✗ En supposant que la température renvoyée par le capteur est de 23,5°C, compléter la deuxième série de valeurs du chronogramme ci-dessus.

### 3 Simulation du circuit DS1621 commandé par une carte Arduino

On désire que le DS1621 soit à l'adresse la plus basse possible.

✗ Editer un schéma sous proteus

#### 3.1 Programmation de la carte Arduino

Avant d'utiliser ce circuit, il est nécessaire de l'initialiser, le début du programme pour la carte Arduino peut être le suivant:

```
#include<Wire.h> // INCLUSION DE LA LIBRAIRIE WIRE

const int DS1621_ID = B_ _ _ _ _ ; // ATTENTION : REMPLACER LES _ PAR L'@ DU DS1621
void setup() {
  Serial.begin(9600); // 
  Wire.begin(); // 
  Wire.beginTransmission(DS1621_ID); // 
  Wire.write(0xAC); // 
  Wire.write(0x01); // 
  Wire.endTransmission(); // 
}

void loop()
{
}
```

- ✗ Commenter avec précision les lignes du programme ci-dessus, notamment le choix des valeurs 0xAC et 0x01.
- ✗ Écrire un programme qui envoie toutes les secondes sur le port série les deux octets de température du DS1621, pour cela s'aider des réponses à la question 2.2.
- ✗ Testez le programme et vérifiez la cohérence des valeurs obtenues.
- ✗ Modifiez le programme précédent afin que la température soit affichée à  $\pm 0,5^\circ$  par un nombre en virgule flottante.

### 3.2 Relevés des signaux SCL et SDA

- x Relever à l'aide d'un graph numérique sous proteus les signaux SCL et SDA pour les deux premiers octets de chaque mesure de température :

- x Sur les relevés précédents, repérer: les phases de **Start** et de **Stop**, les phases d'**acquiescement** et les valeurs transmises.
- x Enlevez le circuit et par un relevé, montrez qu'il n'y a plus d'acquiescement de la part de l'esclave.
- x Modifiez ce qui est nécessaire afin que le DS1621 soit à l'adresse la plus haute possible, montrez le bon fonctionnement par un relevé adéquat.

## 4 Pour aller plus loin

D'après la notice du DS1621, il est possible d'obtenir une précision à  $\pm 0,1^\circ$ , modifiez le programme afin d'obtenir cette précision de mesure.