

Nom / Prénom :

## Mesurer une température avec un (puis deux) capteurs 1-Wire DS18B20 et une carte Arduino

**Compétences abordées :**

O3 - Analyser l'organisation fonctionnelle et structurelle d'un produit.

**Connaissances abordées :**

- 2.4. Approche fonctionnelle et structurelle d'une chaîne d'information
  - 2.4.1. Typologie des chaînes d'information
  - 2.4.2. Acquisition et restitution de l'information
- 3.4 Comportement Informationnel des produits
  - 3.4.1 Nature et représentation de l'information
    - Nature d'une Information
- 5.3 Constituants de l'information
  - 5.3.1 Capteurs, conditionneurs
    - Capteurs numériques
  - 5.3.3 Composants programmables
    - Cartes électroniques à microcontrôleur.
  - 5.3.4 Composants de transmission de l'information
    - Caractéristiques des Bus de communication.

**Matériel nécessaire**

<p><b>1 carte Arduino Uno avec un Shield Grove et son câble USB</b></p>	<p><b>2 Modules DS 18B20 en boitier TO-92 ou en format « Sonde Etanche »</b></p>
	
<p><b>HUB I2C Grove</b></p>	<p><b>Câbles Groves</b></p>
	
<p><b>Un ordinateur</b></p>	<p><b>l'IDE d'Arduino</b></p>
	

Dans ce TP, nous allons apprendre ensemble à utiliser un capteur 1-Wire DS18B20 pour mesurer une température au moyen d'une carte Arduino UNO. Nous verrons ensuite

Nom / Prénom :

comment mesurer plusieurs températures en même temps avec plusieurs capteurs sur un même bus.

### Présentation du capteur

Le capteur [DS18B20](#) du fabricant Maxim (anciennement Dallas Semi-conducteur) est un capteur de température **numérique** intégrant tout le nécessaire pour faire la mesure : Capteur analogique, convertisseur analogique / numérique, électronique de communication.

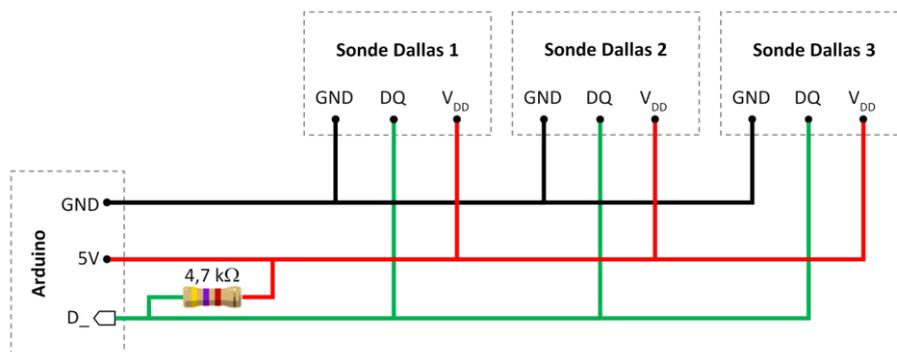
Il communique via [un bus 1-Wire](#) et possède une résolution numérique de 12 bits avec une plage de mesure de -55°C à +125°C. La précision analogique du capteur est de 0,5°C entre -10°C et +85°C.

Le capteur DS18B20 existe dans le commerce en deux versions : en boîtier [TO-92](#) pour des utilisations standards en intérieur, ou en format "[sonde étanche](#)" pour des applications en milieu humide / extérieur.

*N.B. Les deux formats sont strictement identiques d'un point de vue technique.*

<p>Capteur DS18B20 en boîtier TO-92 et dans sa version module vendu par DF ROBOT</p>	<p>Capteur DS18B20 en format "sonde" étanche</p>

### Câblage de 3 Capteurs DS18B20



Le capteur DS18B20 est un capteur **1-Wire**, cela signifie qu'il communique avec une carte **maître** au moyen d'un **bus 1-Wire**. Plusieurs capteurs peuvent être reliés sur un même bus 1-Wire. De plus, chaque capteur dispose d'une **adresse unique** gravée lors de la fabrication, il n'y a donc pas de risque de conflit.

Nom / Prénom :

Un bus 1-Wire est composé classiquement des trois fils : un fil de masse, un fil d'alimentation (5 volts) et un fil de données.

Comme tout périphérique 1-Wire, le DS18B20 contient un "scratchpad" qui est une sorte de mémoire tampon sécurisée ou l'on peut venir lire et / ou écrire des données. C'est dans cette mémoire qu'on vient lire les données de mesures et écrire les informations de configuration du capteur.

Le scratchpad du capteur DS18B20 est divisé en quatre parties :

- 1- Le résultat de la dernière mesure de température (deux octets),
- 2- Deux octets à usages divers (le capteur dispose d'un mode "alarme", mais cela ne sera pas traité dans ce TP),
- 3- Le registre de configuration du capteur,
- 4- Une somme de contrôle. (Checksum)

Ce qui nous intéresse, ce sont les deux premiers octets qui contiennent le résultat de la mesure de température.

BYTE 0	TEMPERATURE LSB (05h)	} (85°C)
BYTE 1	TEMPERATURE MSB (05h)	
BYTE 2	T <sub>H</sub> REGISTER OR USER BYTE 1*	
BYTE 3	T <sub>L</sub> REGISTER OR USER BYTE 2*	
BYTE 4	CONFIGURATION REGISTER*	
BYTE 5	RESERVED (FFh)	
BYTE 6	RESERVED	
BYTE 7	RESERVED (10h)	
BYTE 8	CRC*	

**Le TP est divisé en 3 parties :**

A) Comment obtenir l'adresse 1-WIRE d'un capteur DS18B20 ? [search](#)

B) Comment utiliser l'adresse pour faire une mesure de température ?

C) Comment faire deux mesures de températures avec deux capteurs sur le même bus ?

Nom / Prénom :

## A - Comment obtenir l'adresse 1-WIRE d'un capteur DS18B20 ?

### Méthode search

#### 1) Quel est le format de l'adresse 1-WIRE d'un capteur DS18B20 ?

##### Etude de la [documentation constructeur](#) :

In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique [64-bit code](#). Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited.

Questions : 1.1) ✍ Sur combien de bits est codée une adresse 1\_WIRE ?

1.2) ✍ En déduire le nombre d'octet correspondant ?

#### 2) Lecture de l'adresse du capteur DS18B20 :

2.1) ✘ Branchement : Brancher un et un seul capteur DS 18B20 sur une Entrée/Sortie numérique de la carte Arduino.

✍ Noter l' Entrée/Sortie choisie.

2.2 ) Ouvrir le programme **DS18B20search.ino**

🔗 Dans la partie déclaration, indiquer le numéro de broche choisi pour le capteur et le bus 1 WIRE

```
// DS18S20 Temperature chip i/o
OneWire ds(______); // on pin _____
```

2.3 ) Téléverser le programme dans votre carte Arduino, ouvrir le moniteur série et noter l'adresse (en Hexadécimal) de votre capteur numéro 1:

Adresse du Capteur DS 18 B 20<sub>1</sub> = 0x XX XX XX XX XX XX XX XX

Nom / Prénom :

## B - Comment utiliser l'adresse pour faire une mesure de température ?

Dans cette partie vous utiliserez et modifierez le programme : [DS18B20measure.ino](#)

1. Comment déclarer l'adresse du capteur à l'aide d'un tableau ?
2. Comment effectuer une mesure de température ?
3. Comment lire le [scratchpad](#) (la mémoire interne du capteur) ?
4. Comment afficher les 2 octets contenant le résultat de la dernière mesure de température ?
5. Comment transformer ses 2 octets en 1 mot de 16 bits ?
6. Comment calculer la température ?

🔗 Ouvrir le programme [DS18B20measure.ino](#)

### 1. Comment déclarer l'adresse du capteur à l'aide d'un tableau ?

#### 1.1 Quel est le type de variable contenue dans mon tableau ?

À la lecture de l'article <https://www.locoduino.org/spip.php?article11>,

choisissez parmi les 4 suivantes, le type de variables nécessaires pour l'adresse du capteur DS

18B20 :

- Const int
- Const byte
- Int
- Byte

#### 1.2 Comment déclarer mon tableau ? // Déclaration et initialisation d'un tableau

En vous aidant de l'article <https://www.locoduino.org/spip.php?article227>

☰ compléter la partie déclaration de votre programme en déclarant l'adresse de votre capteur dans un tableau que vous nommerez à votre guise:

//Déclaration du tableau :

Nom / Prénom :

## 2. Comment effectuer une mesure de température ?

Pour toute communication avec un esclave 1-WIRE vous devez effectuer les 3 lignes de commandes suivantes :

Initialisation du Bus  
Sélection de l'esclave ds.select(nomdutabseau)  
Commande

Exemple de demande de mesure de température :

```
// Demande de mesure de Température : Start Conversion Convert T [44h]
// Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the
scratch- pad memory
ds.reset(); // Initialisation du Bus
ds.select(nomdutabseau); // Selection de l'esclave ds.select(nomdutabseau)
ds.write(0x44); // the resulting thermal data is stored in the 2-byte temperature register.
```

2.1 Recopier dans le programme « DS18B20mesure » les 3 lignes ci-dessus en indiquant le nom de votre tableau (question 1.2)

```
// Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the
scratch- pad memory
```

```
// Initialisation du Bus
// Selection de l'esclave ds.select(nomdutabseau)
// the resulting thermal data is stored in the 2-byte temperature register.
```

## 3. Comment Lire le SCRATCHPAD ? Read Scratchpad [BEh]

Il suffit d'utiliser la même méthode que pour la mesure de température en remplaçant 44h par BEh

3.1 Recopier dans le programme « DS18B20mesure.ino » les 3 lignes nécessaires

```
// Lecture du SCRATCH PAD : Read Scratchpad
 // Initialisation du Bus
// Selection de l'esclave ds.select(nomdutabseau)
// Read Scratchpad [BEh]
// The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the
9th byte (byte 8 – CRC) is read.
```

Juste après la demande de lecture du SCRATCH PAD, il faut effectuer les lignes suivantes

```
//Mise en mémoire des 9 Octets du "Scratch Pad" dans le tableau scratchpad[i]
for ( i = 0; i < 9; i++) {
  scratchpad[i] = ds.read();
}
```

## 4. Comment afficher les 2 octets contenant le résultat de la dernière mesure de température ?

Aide [les différents types de variable :](#)

[http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.SyntheseTypesDonnees](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.SyntheseTypesDonnees)

4.1 Compléter le programme, en déclarant les 2 variables comme indiqués dans les commentaires



Nom / Prénom :

Partie Déclaration : // **Déclarer vos VARIABLES**

//Déclaration d'une variable de type **Octet** de nom **octetdepoidsfort** ;

//Déclaration d'une variable de type **Octet** de nom **octetdepoidsfaible** ;

Programme principal : void loop() // **Lire le TABLEAU**

4.2 Compléter la partie du programme permettant de lire les 2 premières cases du tableau

//La variable **octetdepoidsfort** reçoit le contenu de la case 1 du tableau « **scratchpad** »

//La variable **octetdepoidsfaible** reçoit le contenu de la case 0 du tableau « **scratchpad** »

4.3 Compléter la partie du programme permettant d'afficher les deux octets :

Serial.println("Octets du Scratch Pad en Hexa = "); // **Afficher les valeurs**

Serial.print("Octet de poids fort = ");

// Afficher sur le moniteur série le contenu de la variable **octetdepoidsfort** en hexadécimal

Serial.print("Octet de poids faible = ");

// Afficher sur le moniteur série le contenu de la variable **octetdepoidsfaible** en hexadécimal

4.4 Téléverser votre programme dans la carte Arduino

Vérifier le bon fonctionnement

Pour 22 ° C les valeurs hexadécimales sont :

**Octetdepoidsfort = 1**

**Octetdepoidsfaible = 60**

Faire valider le fonctionnement par votre enseignant(e)



Nom / Prénom :

### 5. Comment transformer ses 2 octets en 1 mot de 16 bits ?

Pour pouvoir lire la température, il faut maintenant assembler en une seule variable (déclarée préalablement) l'octet de poids fort (msb) et l'octet de poids faible. On dira il faut concaténer les deux octets : En **programmation**, la **concaténation** de deux octets consiste à les mettre bout à bout.

```
1 word(msb, lsb);
```

La fonction `word()` permet de concaténer deux octets pour former un mot.

Avec ces fonctions, il est possible de faire pas mal de choses très intéressantes. Exemple :

```
1 int valeur = analogRead(A0); // Retourne une valeur sur 10 bits
2
3 byte lsb = lowByte(valeur); // Contient uniquement les bits 0-7
4 byte msb = highByte(valeur); // Contient uniquement les bits 8-9
5
6 word valeur_2 = word(msb, lsb); // Reconstitue le mot d'origine
```

5.1  Choisir [le type de variable idoine](#) et écrire la ligne de déclaration :

//Déclaration d'une variable de taille 16 bits non signée de nom **temperature** :

5.2  D'après l'exemple ci-dessus, écrire les 2 lignes de programme qui

- assemblent les 2 variables.
- affichent la valeur hexadécimale de la variable « Temperature »

//Concaténation des deux octets dans la variable « Temperature »

Serial.print("Température en hexa = ");

//Afficher la valeur hexadécimale du mot « Temperature ».

5.3  Téléverser votre programme dans la carte Arduino

 Vérifier le bon fonctionnement

Pour 22 °C la valeur hexadécimale est :

**Température en hexa = 160**

 Faire valider le fonctionnement par votre enseignant(e)



Nom / Prénom :

## 6. Comment calculer la température ? Comment convertir en degré celsius ?

Extrait de la documentation constructeur :

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or **12 bits**, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and **0.0625°C**, respectively. The default resolution at power-up is 12-bit.

La température, exprimée en degrés celsius, sera stockée dans une variable de nom « `temperaturecelsius` ». Elle doit pouvoir contenir des valeurs numériques à virgule.

6.1  Choisir [le type de variable idoine](#) et écrire la ligne de déclaration :

//Partie déclaration :

//Déclaration d'une variable de type **nombre à virgule** de nom **temperaturecelsius** ;

6.2  Ecrire, à partir des éléments lus sur la documentation constructeur, la ligne de programme permettant de calculer la température.

// Conversion à partir des éléments lus sur la documentation constructeur

6.3  Ecrire la ligne de programme permettant l'affichage de la variable « `temperaturecelsius` » sur le moniteur série

`Serial.print("Température en °C = ");`

// Affichage de la température sur le moniteur série

6.4  Téléverser votre programme dans la carte Arduino

 Vérifier le bon fonctionnement

Pour 22 °C la valeur hexadécimale est :

**Température en °C = 22**

**Vérifier avec un thermomètre dans la classe.**

 Faire valider le fonctionnement par votre enseignant(e)



Nom / Prénom :

## C - Comment faire deux mesures de températures avec deux capteurs sur le même bus ?

Vous devez refaire les étapes :

### 1. **Partie A** : Comment obtenir l'adresse 1-WIRE d'un capteur DS18B20 ?

Adresse du deuxième Capteur DS 18 B 20 = 0x XX XX XX XX XX XX XX XX

### 2. **Partie B** : en prenant soin d'utiliser d'autres noms de variables :

### 3. Afficher les deux températures

📁 Téléverser votre programme dans la carte Arduino



🔍 Vérifier le bon fonctionnement

Faire chauffer un des capteurs dans votre main afin d'obtenir deux températures différentes



👉 Faire valider le fonctionnement par votre enseignant(e)



Nom / Prénom :

Conclusion : Compléter la chaine d'information :

- 1) ✎ Pour chacune des flèches décrire la nature et les unités de l'information transportée :
- 2) ✎ Nommer chacun des blocs (ici la partie affichage se fait à l'aide du moniteur série)

