

TP Bus 1-Wire ARDUINO

NOM :

Date :

Objectif final :

Mettre en service le bus 1 wire et la librairie OneWire sur une carte Arduino Uno pour effectuer une mesure de température avec un capteur de température DS18B20.

Compétences abordées :

Réaliser	C4.3 : Analyser la structure logicielle. Procéder aux modifications logicielles C4.5 : Tester et valider un matériel
Installer	C5.2 : Exécuter des mesures et tests appropriés.

Savoirs abordés :

Savoir	Description
S7.6. Réseaux locaux industriels (RU)	Liaison 1 fil (1-Wire)
S8.1. Instruments de mesure	Analyseur logique Analyseur de signaux

Moyens :

- 1 carte Arduino Uno
- 1 capteur DS18B20 Dallas
- Une résistance (4,7K Ω)
- Breadboard + fils de liaison.
- Analyseur logique (Saleae).

Conditions :

- Travail en binôme.
- Les documents nécessaires à la mise en œuvre de ce TP se trouvent sur le site de la section BTS SN.
- Durée : 1H30.
- Compte rendu à la fin de la séance.

Prérequis :

- Utilisation d'une carte Arduino Uno et d'un analyseur logique.

Mesure de température avec un capteur DS18B20

I. Lecture du document de présentation du bus 1-Wire et analyse de la documentation du DS18B20

1. Prendre connaissance des documents sur le bus 1-Wire.
2. Quelle est la plage d'alimentation du capteur ? →
3. Peut-il être alimenté directement sur la carte Arduino Uno ? →
4. Quel est le code famille (family code) du circuit ? →
5. Dans quelle plage de températures le capteur peut-il effectuer des mesures ? →
6. La valeur de la température est fournie sous quel format binaire ?



II. Mise en œuvre du capteur et analyse du programme

7. Quelle est la sortie de la carte Arduino uno utilisée pour communiquer avec le capteur ? →
8. Câbler le composant comme illustré sur le site.
9. Le circuit est-il câblé en mode alimentation « externe » ou « parasite » →
Le programme source servant d'inspiration figure en Document ANNEXE 1.
10. Quelle est la librairie nécessaire pour faire fonctionner le capteur sur la carte Arduino ?
→
11. Surligner sur le document ANNEXE 1 le nom de cette librairie.
12. Télécharger cette librairie depuis le site et l'installer sur le PC.
13. Récupérer le projet Arduino sur le site et le désarchiver dans un sous-répertoire « 1-Wire » dans votre sous-répertoire de travail.
14. Compiler et tester le programme. Sur le site figure une vidéo illustrant le résultat attendu.

Faire constater

15. Justifier le format des variables :
data[9] →
addr[8] →
16. Surligner dans le programme la ligne de commande permettant de lancer la conversion. Indiquer le nom donné à cette commande dans la documentation du capteur →
17. Surligner dans le programme la ligne de commande permettant de lire la température. Indiquer le nom donné à cette commande dans la documentation du capteur →
18. Surligner la partie récupérant toutes les données contenues dans la RAM du capteur.
19. Quel est le nom et le format de la variable qui récupère la température ? →
20. Expliciter ce qu'exécute la ligne de programmation : `*temp = ((data[1] << 8) | data[0]) * 0.0625;`
→

21. Avant toute récupération de mesures sur capteur le « Family code » est-il vérifié ? Si oui surligner cette partie d'une autre couleur.
 22. Mettre en service l'analyseur logique et capturer une trame lors d'une mesure.
 23. Identifier le numéro de série (unique) du capteur →
 24. Identifier sur cette capture les différentes commandes commentées précédemment.
 25. Pour une température donnée (relevée dans le moniteur série Arduino) récupérer dans la trame capturée les octets de la RAM qui contiennent la température et vérifier que les 2 valeurs sont les mêmes.
-

Faire constater

Document ANNEXE 1

Programme source pour mesure de température avec le DS18B20

Sitographie : <https://skyduino.wordpress.com/2012/04/26/arduino-capteur-de-temperature-ds18b20/>

```
#include <OneWire.h> // Inclusion de la librairie OneWire

#define DS18B20 0x28 // Adresse 1-Wire du DS18B20
#define BROCHE_ONEWIRE 7 // Broche utilisée pour le bus 1-Wire

OneWire ds(BROCHE_ONEWIRE); // Création de l'objet OneWire ds

// Fonction récupérant la température depuis le DS18B20
// Retourne true si tout va bien, ou false en cas d'erreur
boolean getTemperature(float *temp)
{
    byte data[9], addr[8];
    // data : Données lues depuis le scratchpad
    // addr : adresse du module 1-Wire détecté

    if (!ds.search(addr))
    { // Recherche un module 1-Wire
        ds.reset_search(); // Réinitialise la recherche de module
        return false; // Retourne une erreur
    }

    if (OneWire::crc8(addr, 7) != addr[7]) // Vérifie que l'adresse a
été correctement reçue
        return false; // Si le message est corrompu on retourne une erreur
```

```

if (addr[0] != DS18B20) // Vérifie qu'il s'agit bien d'un DS18B20
    return false;    // Si ce n'est pas le cas on retourne une erreur

ds.reset();          // On reset le bus 1-Wire
ds.select(addr);     // On sélectionne le DS18B20

ds.write(0x44, 1);   // On lance une prise de mesure de température
delay(800);          // Et on attend la fin de la mesure

ds.reset();          // On reset le bus 1-Wire
ds.select(addr);     // On sélectionne le DS18B20
ds.write(0xBE);      // On envoie une demande de lecture du scratchpad

for (byte i = 0; i < 9; i++) // On lit le scratchpad
    data[i] = ds.read();      // Et on stocke les octets reçus

// Calcul de la température en degré Celsius
*temp = ((data[1] << 8) | data[0]) * 0.0625;

// Pas d'erreur
return true;
}

// setup()
void setup()
{
    Serial.begin(9600); // Initialisation du port série
}

// loop()
void loop()
{
    float temp;

    // Lit la température ambiante à ~1Hz
    if(getTemperature(&temp))
    {
        // Affiche la température
        Serial.print("Temperature : ");
        Serial.print(temp);
        Serial.write(176); // caractère °
        Serial.write('C');
        Serial.println();
    }
}

```