Lycée TOUCHARD WASHINGTON

Informatique embarquée

TD n°2

CIEL 1
2023-2024

Gestion du capteur de température

1- Création du projet

Créez un projet intitulé Thermometre en utilisant **PlatformIO** pour l'environnement NetBeans. Une fois le projet ouvert dans **NetBeans**, copiez les fichiers **afficheur.cpp**, **afficheur.h**, **fonts.h** et **esp32_snir.h** dans les répertoires src et include de votre projet.

Créez également les fichiers **thermometre.h** et **thermometre.cpp** qui regrouperont tous les éléments tels que les fonctions, les constantes nécessaires à gestion du capteur de température **DS18S20**.

Dans le fichiers le fichier **thermometre.h** ajoutez tous les inclusions de librairie nécessaires au fonctionnement du capteur DS18S20.

Dans le fichier '**thermometre .cpp**', déclarez en tant que variable globale les instances des classes **OneWire et DallasTemperature** nommée '**oneWire**' et '**capteurTemp**', pour une utilisation dans l'ensemble du fichier.

2- Création de la librairie Thermometre

Ajoutez la fonction `void InitialiserCapteurTemperature(void)` à votre librairie, elle est chargée d'initialiser le bus 1-Wire pour la mesure de température.

On donne maintenant le prototype de la fonction **ObtenirTemperature()** :

```
bool ObtenirTemperature(int *_temperature, int * _dixiemes, const int _numCapteur=0) ;
```

Cette fonction possède deux paramètres de sortie **_temperature** et **_dixiemes**, ils sont implémentés sous la forme de pointeurs. Elle possède également un paramètre d'entrée un entier, **_numCapteur** avec comme valeur par défaut 0. Enfin, elle retourne un booléen égal à **true** si la mesure c'est correctement réalisée, **false** sinon.

Recopiez ce prototype dans le fichier **thermometre.h** et complétez la définition de cette fonction dans le fichier **thermometre.cpp** :

```
bool ObtenirTemperature(int *_temperature, int * _dixiemes, const int _numCapteur)
{
    bool retour = false;
    capteurTemp.requestTemperatures();
    float temperature = capteurTemp.getTempCByIndex(_numCapteur);
    if(temperature != DEVICE_DISCONNECTED_C)
    {
        retour = true;
        *_dixiemes =  //  partie décimale arondie àu dixième supérieure
        *_temperature = // partie entière de la température
    }
    return retour;
}
```

Pour calculer la partie décimale et la partie entière de la température, étudiez les fonctions mathémathique de la librairie math.h en particulier les fonctions **ceil**, **trunc**, **floor**, **round**

https://fr.cppreference.com/w/c/numeric/math

3- Codage du programme principal

Codez les fonctions **setup()** et **loop()** dans le fichier **main.cpp**. La première fonction a pour rôle d'initialiser l'afficheur et le capteur de température. La seconde fonction obtient la température (sous forme d'un nombre entier) et les dixièmes de degré (avec une décimale) pour les afficher sur l'afficheur OLED toutes les 5 secondes, à condition que la mesure soit valide.

Il est important de noter que la fonction **ObtenirTemperature** requiert les adresses des variables contenant la température et les dixièmes de degré (comme les paramètres de scanf). Le troisième paramètre de la fonction possède une valeur par défaut égale à 0, ce qui signifie qu'il n'est pas nécessaire pour obtenir la mesure du premier capteur du bus 1-Wire.