

## **Rapport de stage**

### **Smart Room**

**Spécialité :Informatique industrielle**

**OTTO engineering service**

**Période de stage :Du 01/07/2023 au 31/08/2023**

**Réalisé par :Barhoumi nawres**

**Encadré par :Ghalgaoui kamel**

**Année universitaire :2022/2023**

# Remerciement

Ce travail est l'aboutissement d'un parcours au cours duquel j'ai bénéficié des conseils, des encouragements et du soutien .

Au terme de ce stage je tiens à exprimer mes respects et mes sincères remerciement aux responsable de la société OTTO ES Mr.Hamdi Belghith.

Un grand remerciement va à mon encadreur, Mr. Kamel Ghalgaoui, pour son dévouement exceptionnel à mon apprentissage. Ses conseils précieux et bienveillants m'ont permis de développer mes compétences et de prendre de l'assurance dans l'exécution de mes missions. Sa disponibilité pour répondre à mes questions et discuter de mes progrès a grandement enrichi mon expérience.

Il m'a beaucoup appris sur son entreprise et les défis qu'un responsable doit relever au quotidien.

Je tiens également à exprimer ma reconnaissance envers toute l'équipe d'Otto Engineering Service. En fin je souhaite la bonne continuation à toute l'équipe OTTO ES ...

## Table des matières

|                                                   |    |
|---------------------------------------------------|----|
| Liste des figures : .....                         | 4  |
| Introduction générale.....                        | 1  |
| Chapitre 1 : l'entreprise d'accueil .....         | 2  |
| I. Introduction :.....                            | 2  |
| II. Présentation du cadre de stage :.....         | 2  |
| Chapitre 2 :Contexte du projet.....               | 4  |
| I. Introduction.....                              | 4  |
| II. Description du projet.....                    | 4  |
| III. Architecture du projet :.....                | 4  |
| IV. Présentation du protocole I2C .....           | 5  |
| V. Présentation du protocole UART.....            | 7  |
| Chapitre3 : Logiciels et matériels utilisés ..... | 10 |
| I. Description du matériel.....                   | 10 |
| II. Description du logiciel .....                 | 14 |
| Référence bibliographique .....                   | 17 |

## Liste des figures :

|                                                  |    |
|--------------------------------------------------|----|
| Figure 1:Logo OTTO ES .....                      | 2  |
| Figure 2:architecture du projet .....            | 4  |
| Figure 3:Architecture I2C.....                   | 5  |
| Figure 4:message du protocole I2C .....          | 6  |
| Figure 5:Schéma bloc simplifié de l'UART .....   | 8  |
| Figure 6:Trame UART.....                         | 9  |
| Figure 7:carte STM32F407VGT6.....                | 10 |
| Figure 8:CAPTEUR MCP9808 .....                   | 11 |
| Figure 9 :Adresse MCP9808 .....                  | 12 |
| Figure 10:tableau des fonctions des broches..... | 12 |
| Figure 11:L'afficheur LCD 16X02.....             | 13 |
| Figure 12:l'adresse LCD .....                    | 13 |
| Figure 13:convertisseur UART-USB .....           | 14 |
| Figure 14:Logo stm32cubeide .....                | 14 |
| Figure 15:Logo Tera Term .....                   | 14 |
| Figure 16:L'affichage sur TeraTerm .....         | 15 |
| Figure 17:LOGO langage c .....                   | 15 |
| Figure 18:l'affichage du code .....              | 16 |
| Figure 19:la température .....                   | 16 |

## Introduction générale

Mon expérience de ce stage, qui s'est déroulée du 01 juillet 2023 au 31 août 2023 chez OTTO ES, a été une opportunité inestimable pour moi dans le domaine des systèmes embarqués. Ce stage a non seulement renforcé mes compétences, mais il a également enrichi ma formation. Mon objectif principal durant cette période était de mettre en pratique les connaissances que j'avais acquises au cours de mes deux semestres précédents et de découvrir les multiples facettes de la profession liée aux systèmes embarqués. cœur du monde professionnel, de participer activement à des projets concrets, et d'observer de près les défis et les solutions dans ce domaine en constante évolution. j'ai acquis une expérience précieuse qui m'a non seulement éclairée sur ma future carrière, mais qui a également renforcé ma passion pour les systèmes embarqués et mon désir d'apprendre et de contribuer davantage à ce domaine fascinant.

# Chapitre 1 : l'entreprise d'accueil

## I. Introduction :

Dans ce chapitre je vais présenter en premier le lieu de la société dans laquelle j'ai effectué mon stage et mon sujet proposé .

## II. Présentation du cadre de stage :

### a. Présentation de l'entreprise :

OTTO Engineering Services rassemble une communauté de spécialistes passionnés par les systèmes embarqués et les nouvelles technologies.

Ils proposent une large gamme de services autour des systèmes embarqués tels que le développement logiciel, la surveillance, l'IHM, le développement matériel, la revue de schémas.

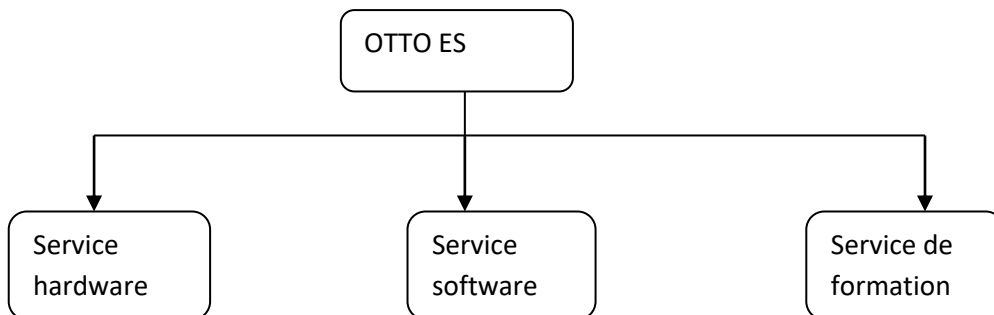
Ils prennent en charge divers composants matériels d'OTTO Engineering Services eux-mêmes ou fournis par le client.

Ils mettent leur expertise et leur savoir-faire au service de leur clients pour les aider à relever les défis liés au développement des dernières technologies électroniques et logicielles. Que ce soit pour l'analyse et la spécification de leurs besoins, le développement, la mise en service et les tests ou la gestion du cycle de vie des produits, pour bénéficier de leur expertise et de leur expérience.



Figure 1:Logo OTTO ES

## b. Organigramme :



### Service software :

La mission d'OTTO Engineering Services est d'intégrer les technologies informatiques et de communication de pointe pour développer des applications innovantes qui offrent un avantage concurrentiel pour améliorer la vie quotidienne, la rendant plus simple, plus sûre et plus agréable grâce à notre savoir-faire.

### Service hardware :

Ce service est géré par Mr.Jawher Sakka , ce service aide les clients a réaliser des solutions embarqué ,développer des solutions matériels , tels que : développement de plinthes ,développement SOM (system on module )et SBC (single board computer ) , Examen des schémas .

### Service de formation :

OTTO Engineering Services propose plusieurs formations a fin de faire fructifier l'expérience de l'équipe ,leurs formations peuvent etre dispensée partout dans le monde et chaque client peut choisir le date . telsque : Linux embarqué , yocto ,développement du pilote du noyau linux ,construire la racine ,linux en temps réel avec Xenomai .

## c. Contexte du stage :

Dans l'objectif de maitriser et approfondir mes connaissances déjà acquises OTTO ES m'a accueilli dans ces locaux , Elle m'a proposé comme sujet « smart room » .

## Chapitre 2 :Contexte du projet

### I. Introduction

Cette solution de surveillance de la température revêt une importance cruciale dans un monde où la précision et la gestion des données thermiques sont essentielles pour la sécurité, la qualité et la conformité aux normes. Elle trouve des applications vitales dans des domaines aussi variés que la sécurité domestique, où elle peut détecter les risques de surchauffe et d'incendie, l'industrie, où elle assure la stabilité des processus de fabrication, la chaîne d'approvisionnement alimentaire, où elle garantit la fraîcheur des produits, et même la recherche scientifique, où elle permet des analyses thermiques précises. Cette introduction met en lumière l'omniprésence de la nécessité de surveiller la température avec précision dans divers secteurs.

### II. Description du projet

Ce projet représente une solution professionnelle de surveillance de la température toutes les 5 secondes. Il intègre un capteur de température MCP9808, une carte STM32, et le protocole I2C pour une précision accrue dans la collecte de données. Les informations de température sont ensuite présentées de manière conviviale sur un écran LCD 16x02, également via I2C, pour une utilisation en temps réel.

De plus, ce système permet la transmission instantanée des données de température vers un ordinateur via l'émulateur Tera Term en utilisant le protocole UART. Cette communication est rendue possible grâce à un convertisseur UART-USB, offrant une intégration aisée dans les environnements professionnels. La combinaison de ces éléments assure un suivi et une visualisation efficaces des données thermiques, essentielles dans diverses applications industrielles et professionnelles.

### III. Architecture du projet :

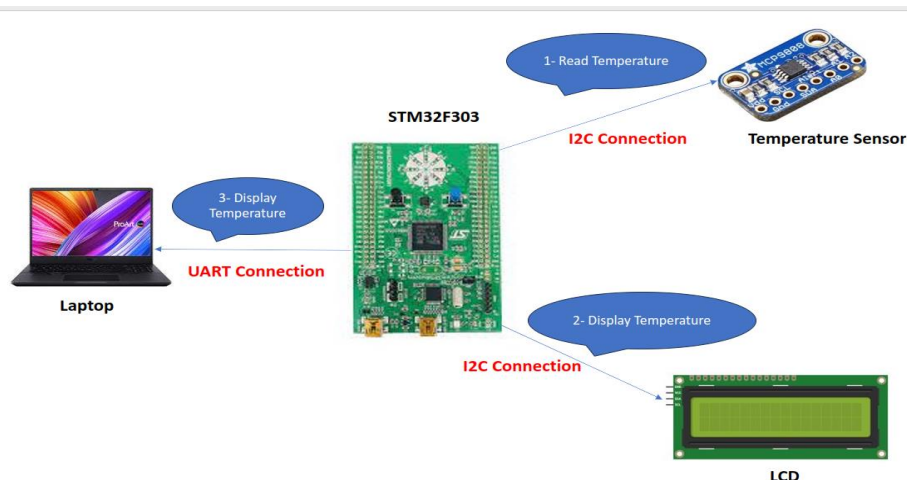


Figure 2:architecture du projet



## IV. Présentation du protocole I2C

### a. Introduction :

I<sup>2</sup>C (signifie : Inter-Integrated Circuit, en anglais) est un [bus informatique](#) qui a émergé de la « guerre des standards » lancée par les acteurs du monde électronique, a été développée par [Philips](#) en 1982. Depuis elle est maintenue par [NXP](#) (ex-division semiconducteurs de Philips).

### b. Définition :

Le protocole I2C est un protocole maître-esclave ce qui signifie qu'il y'a un seul dispositif maître qui contrôle les transferts de données avec un ou plusieurs dispositifs esclave , Le dispositif maître initial et contrôle la communication avec les dispositifs esclaves.

### c. Les lignes de communication :

**SDA (Serial Data Line)** :C'est la ligne de données bidirectionnelle utilisée pour transférer les informations entre les dispositifs connectés.

**SCL (Serial Clock Line)** :C'est la ligne de signal d'horloge qui synchronise les transferts de données entre les dispositifs.

### d. Caractéristiques :

Signaux numérique , série, half-duplex , les données sont encapsulé , nombre de bits par trame:variable , Prise en charge de plusieurs vitesses de communication (standard, mode rapide, mode haut débit) , crypté:non , baudrate:entre 100kbps et 3.4Mbps ou 5Mbps(unidirectionnel).

### e. Architecture :

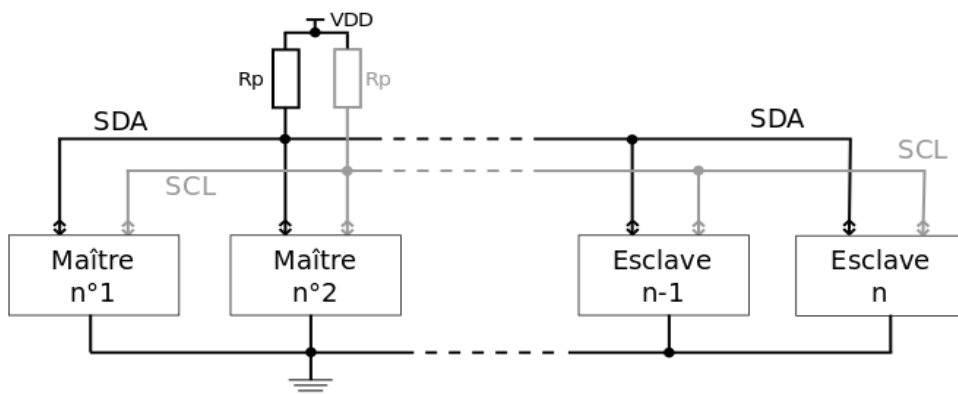


Figure 3:Architecture I2C

### f. La trame du I2C :

La trame de I2C est constituée des bits suivants :

Au repos SDA et SCL sont au niveau 1, aucun circuit n'émet.

**Début de la Trame (Start Condition)** : La trame commence par un signal de démarrage (Start) lorsque la ligne SDA passe de haut (1) à bas (0) alors que la ligne SCL est à l'état haut (1).

**Adresse de l'Esclave** : Après le signal de démarrage, l'émetteur envoie l'adresse de l'esclave qu'il souhaite communiquer. Cette adresse est généralement composée de 7 bits (parfois 10 bits dans le cas de l'I2C étendu) suivis d'un bit de lecture/écriture (R/W), indiquant si l'émetteur souhaite lire (R=1) ou écrire (W=0) des données à l'esclave.

**Bit d'acquittement (ACK)** : Après avoir reçu l'adresse de l'esclave, l'esclave envoie un bit d'acquittement (ACK) en mettant la ligne SDA à 0 pour confirmer qu'il a été correctement formulé.

**Donnée (Data)** : Ensuite, les données sont envoyées, généralement sous forme d'octets, du maître à l'esclave ou vice versa. Chaque octet est suivi d'un bit d'acquittement (ACK) pour indiquer si la transmission a été réussie.

**Bit de Stop (Stop Condition)** : La trame se termine par un signal d'arrêt (Stop) lorsque la ligne SDA passe de bas (0) à haut (1) alors que la ligne SCL est à l'état haut (1). Cela indique la fin de la transmission.

**Recommencement (Repeated Start)** : Si le maître souhaite communiquer avec un autre esclave immédiatement après une transmission, il peut envoyer un signal de redémarrage (Repeated Start) au lieu d'un signal de démarrage.

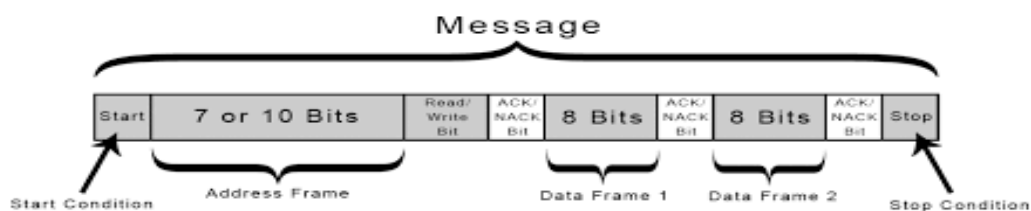


Figure 4: message du protocole I2C

#### g. Adresse I2C :

- \* chaque composant a sa propre adresse.
- \* Eventuellement configurable avec des pins externes pour pouvoir en mettre plusieurs sur le même bus.
- \* Adressage étendu sur 10 bits.

#### h. Conclusion :

Il est important de noter que l'I2C est un protocole bidirectionnel, ce qui signifie que les données peuvent être transmises dans les deux sens entre le maître et l'esclave au sein de la même trame. De plus, l'I2C prend en charge la lecture et l'écriture séquentielle, ce qui permet au maître de lire ou d'écrire plusieurs octets consécutifs sans avoir à répéter l'adresse à chaque fois.

## V. Présentation du protocole UART

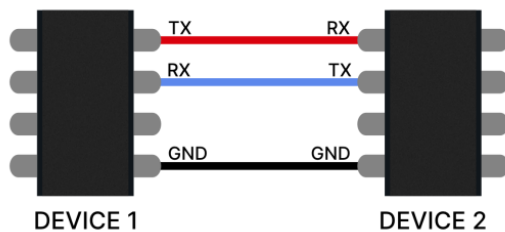
### a. Définition :

Le protocole UART (Universal Asynchronous Receiver-Transmitter) est un protocole de communication asynchrone largement utilisé pour transférer des données séries entre des équipements électroniques tels que des microcontrôleurs, des modules RF, des capteurs, des périphériques de communication, etc. la transmission de données bit par bit sur une seule ligne de communication, ce qui le rend simple et efficace pour les communications à courte distance.

### b. Les lignes de communication :

**TX (Transmit - Émission) :** Cette ligne est utilisée par le dispositif émetteur pour envoyer des données au dispositif récepteur. Les données à transmettre sont sérialisées (converties en une séquence de bits) et envoyées sur cette ligne.

**RX (Receive - Réception) :** Cette ligne est utilisée par le dispositif récepteur pour recevoir les données envoyées par le dispositif émetteur. Les données reçues sont désérialisées (converties de la séquence de bits à leur forme d'origine) et traitées par le dispositif récepteur.



### c. Caractéristique :

**Communication Série :** UART est un protocole de communication série, ce qui signifie que les données sont envoyées un peu à la fois sur une seule ligne de données.

**Asynchrone :** L'UART est asynchrone, ce qui signifie qu'il n'y a pas d'horloge partagée entre les périphériques.

**Débit binaire variable :** L'UART prend en charge des débits binaires variables, ce qui signifie que la vitesse de transmission des données peut être configurée à différents taux, tels que 9600 bps, 115200 bps, etc...

**Simplex ou Duplex :** L'UART peut être utilisé en mode simplex (communication unidirectionnelle) ou en mode duplex (communication bidirectionnelle) en utilisant des lignes distinctes pour la transmission (TX) et la réception (RX).

**Faible complexité matérielle :** L'implémentation matérielle de l'UART est relativement simple, ce qui le rend approprié pour de nombreuses applications embarquées.

#### d. Architecture :

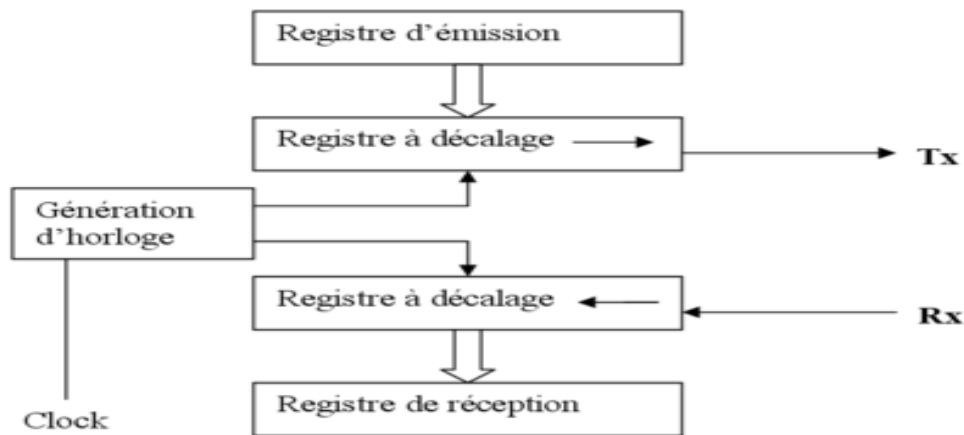


Figure 5: Schéma bloc simplifié de l'UART

#### e. La trame du UART:

Une trame de données dans le protocole UART (Universal Asynchronous Receiver/Transmitter) est la structure qui permet de transmettre un octet (8 bits de données) ou plus de manière asynchrone entre un émetteur et un récepteur.

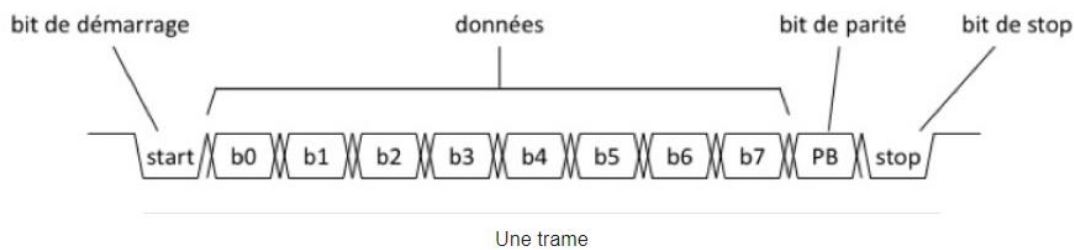
**Début de trame :** La trame commence par un bit de départ (start bit) à 0 (basse tension). Le début de la trame est marqué par cette transition de la haute tension (bit d'arrêt précédent) à la basse tension.

**Données :** Après le bit de départ, les bits de données sont transmis du moins significatif (LSB - Least Significant Bit) au plus significatif (MSB - Most Significant Bit). Le nombre de bits de données peut varier, mais il est généralement de 8 bits. Ces bits contiennent la charge utile de la trame, telle que des caractères ASCII ou des données binaires.

**Bit de parité (optionnel) :** Si la parité est activée, un bit de parité est inclus après les bits de données. Ce bit est utilisé pour vérifier l'intégrité des données et peut être pair (even), impair (odd), ou aucun (no parity). La parité est utilisée pour détecter les erreurs de transmission.

**Bits de stop :** Après les bits de données (et le bit de parité), un ou plusieurs bits de stop sont inclus. Les bits de stop sont toujours à 1 (haute tension). Ils indiquent la fin de la trame et permettent au récepteur de se synchroniser pour la trame suivante. Le nombre de bits de stop est généralement configuré à 1 ou 2.

**Fin de trame :** La trame se termine par la transition d'un bit de stop (haute tension) au bit de départ suivant (basse tension), marquant ainsi la fin de la trame.



**Figure 6:Trame UART**

**f. Conclusion :**

le protocole UART (Universal Asynchronous Receiver/Transmitter) est un protocole de communication série largement utilisé dans les systèmes embarqués, les microcontrôleurs, les ordinateurs et de nombreux autres dispositifs électroniques. Il offre une méthode simple et fiable pour la transmission de données asynchrones, ce qui signifie que les données sont envoyées sans une horloge commune entre l'émetteur et le récepteur.

## Chapitre3 : Logiciels et matériels utilisés

### I. Description du matériel

#### a. La carte STM32

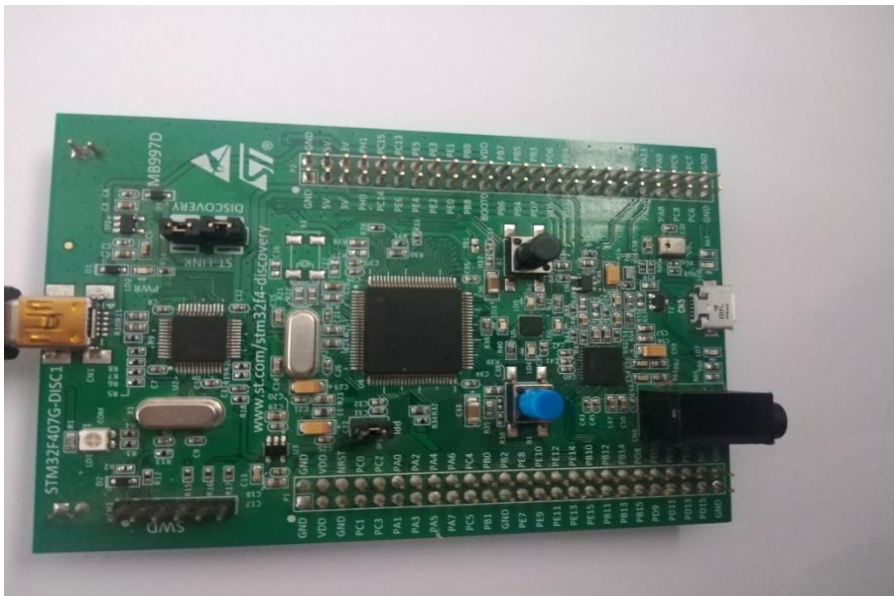


Figure 7:carte STM32F407VGT6

La carte STM32F4Discovery permet aux utilisateurs de développer facilement des applications avec un microcontrôleur haute performance STM32F4 muni d'un processeur ARM Cortex-M4 32 bits.

#### b. Les caractéristiques :

- Microcontrôleur STM32F407VGT6 doté Noyau ARM Cortex-M4F 32 bits, 1 Mo Flash, 192 Ko de RAM dans un package LQFP100.
- ST-LINK/V2 intégré avec sélection de mode commutateur pour utiliser le kit comme un STLINK/V2 autonome (avec connecteur SWD pour programmation et débogage).
- Alimentation de la carte : via bus USB ou depuis une tension d'alimentation externe de 5 V.
- Alimentation externe de l'application : 3 V et 5 V
- LIS302DL, capteur de mouvement ST MEMS, 3 axes accéléromètre à sortie numérique
- MP45DT02, capteur audio ST MEMS, microphone numérique omnidirectionnel

- CS43L22, DAC audio avec classe D intégrée pilote de haut-parleur
- Huit LED :
  - LD1 (rouge/vert) pour la communication USB
  - LD2 (rouge) pour mise sous tension 3,3 V
  - Quatre LED utilisateur, LD3 (orange), LD4 (vert), LD5 (rouge) et LD6 (bleu)
  - 2 LED USB OTG LD7 (vertes) VBus et Surintensité LD8 (rouge)
- Deux boutons poussoirs (utilisateur et reset)
- USB OTG FS avec connecteur micro-AB
- Connecteur d'extension pour toutes les E/S LQFP100 pour connexion rapide à la carte de prototypage et facile sondage

### c. Le Capteur MCP9808



Figure 8:CAPTEUR MCP9808

Le MCP9808 est un capteur de température numérique de haute précision qui est capable de mesurer avec précision la température ambiante. Il utilise une interface de communication I2C pour transmettre les données de température aux microcontrôleurs, aux processeurs, ou à d'autres dispositifs électroniques.

#### d. Les caractéristiques :

- **Précision:**
  - $\pm 0,25$  (typique) de  $-40^{\circ}\text{C}$  à  $+125^{\circ}\text{C}$
  - $\pm 0,5^{\circ}\text{C}$  (maximum) de  $-20^{\circ}\text{C}$  à  $100^{\circ}\text{C}$
  - $\pm 1^{\circ}\text{C}$  (maximum) de  $-40^{\circ}\text{C}$  à  $+125^{\circ}\text{C}$
- **Résolution de mesure sélectionnable par l'utilisateur :**
  - $+0,5^{\circ}\text{C}$ ,  $+0,25^{\circ}\text{C}$ ,  $+0,125^{\circ}\text{C}$ ,  $+0,0625^{\circ}\text{C}$
- **Limites de température programmables par l'utilisateur :**
  - Limite de fenêtre de température
  - Limite de température critique
- **Sortie d'alerte de température programmable par l'utilisateur**
- **Plage de tension de fonctionnement :**  $2,7\text{ V}$  à  $5,5\text{ V}$
- **Courant de fonctionnement :**  $200\text{ }\mu\text{A}$  (typique)
- **Courant d'arrêt :**  $0,1\text{ }\mu\text{A}$  (typique)
- **Interface 2 fils :** compatible I<sup>2</sup>C™/SMBus
- **Packages disponibles :** 2x3 DFN-8, MSOP-8

Dans le cadre de mon projet puisque j'ai utilisé le capteur MCP9808 donc L'adresse du Le capteur de température MCP9808 est « 0011, A2, A1, A0 » dans binaire, où les bits A2, A1 et A0 sont définis en externe en connectant les broches correspondantes à VDD '1' ou GND '0'. L'adresse de 7 bits, transmise dans le flux binaire série, doit correspondre à l'adresse sélectionnée pour le MCP9808 à répondre par un ACK. Le bit 8 de l'octet d'adresse est un bit de lecture/écriture. Mettre ce bit à « 1 » commande une lecture opération, tandis que « 0 » commande une opération d'écriture.

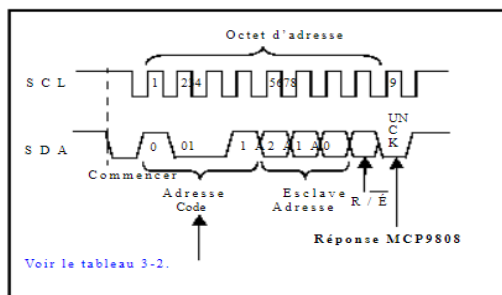


Figure 9 :Adresse MCP9808

#### e. Description du pin :

| DFN | MSOP | Symbol          | Pin Function                                       |
|-----|------|-----------------|----------------------------------------------------|
| 1   | 1    | SDA             | Serial Data Line                                   |
| 2   | 2    | SCL             | Serial Clock Line                                  |
| 3   | 3    | Alert           | Temperature Alert Output                           |
| 4   | 4    | GND             | Ground                                             |
| 5   | 5    | A2              | Slave Address                                      |
| 6   | 6    | A1              | Slave Address                                      |
| 7   | 7    | A0              | Slave Address                                      |
| 8   | 8    | V <sub>DD</sub> | Power Pin                                          |
| 9   | —    | EP              | Exposed Thermal Pad (EP); must be connected to GND |

Figure 10:tableau des fonctions des broches



## f. L'afficheur LCD



Figure 11:L'afficheur LCD 16X02

L'afficheur texte 16x2 est utilisé pour afficher les informations du capteur, afficher les menus ou les invites. L'écran affiche des caractères noirs d'une taille de 5x8 pixels. Le rétroéclairage intégré s'allume en appliquant une alimentation aux broches du module.

Dans le cadre de mon projet, l'adresse de l'afficheur LCD a été configurée à 0x27. Cette composante joue un rôle essentiel dans la visualisation des informations, et voici le tableau récapitulant cette adresse de l'afficheur :

PCF8574 address map

| Pin connectivity |                 |                 | Address of PCF8574 |    |    |    |    |    |    |     | Address byte value |      | 7-bit hexadecimal address without R/W |
|------------------|-----------------|-----------------|--------------------|----|----|----|----|----|----|-----|--------------------|------|---------------------------------------|
| A2               | A1              | A0              | A6                 | A5 | A4 | A3 | A2 | A1 | A0 | R/W | Write              | Read |                                       |
| V <sub>SS</sub>  | V <sub>SS</sub> | V <sub>SS</sub> | 0                  | 1  | 0  | 0  | 0  | 0  | 0  | -   | 40h                | 41h  | 20h                                   |
| V <sub>SS</sub>  | V <sub>SS</sub> | V <sub>DD</sub> | 0                  | 1  | 0  | 0  | 0  | 0  | 1  | -   | 42h                | 43h  | 21h                                   |
| V <sub>SS</sub>  | V <sub>DD</sub> | V <sub>SS</sub> | 0                  | 1  | 0  | 0  | 0  | 1  | 0  | -   | 44h                | 45h  | 22h                                   |
| V <sub>SS</sub>  | V <sub>DD</sub> | V <sub>DD</sub> | 0                  | 1  | 0  | 0  | 0  | 1  | 1  | -   | 46h                | 47h  | 23h                                   |
| V <sub>DD</sub>  | V <sub>SS</sub> | V <sub>SS</sub> | 0                  | 1  | 0  | 0  | 1  | 0  | 0  | -   | 48h                | 49h  | 24h                                   |
| V <sub>DD</sub>  | V <sub>SS</sub> | V <sub>DD</sub> | 0                  | 1  | 0  | 0  | 1  | 0  | 1  | -   | 4Ah                | 4Bh  | 25h                                   |
| V <sub>DD</sub>  | V <sub>DD</sub> | V <sub>SS</sub> | 0                  | 1  | 0  | 0  | 1  | 1  | 0  | -   | 4Ch                | 4Dh  | 26h                                   |
| V <sub>DD</sub>  | V <sub>DD</sub> | V <sub>DD</sub> | 0                  | 1  | 0  | 0  | 1  | 1  | 1  | -   | 4Eh                | 4Fh  | 27h                                   |

Figure 12:l'adresse LCD

## g. Convertisseur UART-USB

Un convertisseur UART-USB, également connu sous le nom d'adaptateur UART-USB, est un dispositif électronique qui permet de convertir les données série (UART, Universal Asynchronous Receiver/Transmitter) en un format compréhensible par un port USB (Universal Serial Bus) sur un ordinateur ou un autre appareil compatible USB. Il fonctionne en transformant les signaux UART, généralement au format binaire, en signaux numériques USB, ce qui permet de transférer des données séries depuis et vers un ordinateur via une connexion USB standard.



Figure 13: convertisseur UART-USB

## II. Description du logiciel

### a. Stm32cubeide



Figure 14: Logo stm32cubeide

STM32CubeIDE est un environnement de développement intégré (IDE) avancé destiné à la programmation et au développement de logiciels embarqués pour les microcontrôleurs STM32 de STMicroelectronics. Il offre une suite complète d'outils de développement, y comprenant un éditeur de code, un débogueur, un compilateur, et des fonctionnalités avancées telles que la configuration graphique des périphériques, la génération de code automatique, la gestion des bibliothèques logicielles STM32Cube, et la prise en charge de l'intégration des outils tiers. STM32CubeIDE facilite la création, le débogage et la maintenance des applications embarquées pour les microcontrôleurs STM32, ce qui en fait un outil précieux pour les développeurs de systèmes embarqués.

### a. Tera term



Figure 15: Logo Tera Term

Tera Term est un émulateur de terminal open source qui vous permet de contrôler à distance vos hôtes Telnet et SSH. Grâce à Tera Term, vous établissez facilement des connexions entre des applications web développées en PHP, Perl, ASP, etc., et n'importe quel hôte Telnet ou SSH.

Le cas de mon projet consiste à afficher la température mesurée par le capteur MCP9808 sur l'émulateur Tera Term, comme suit :

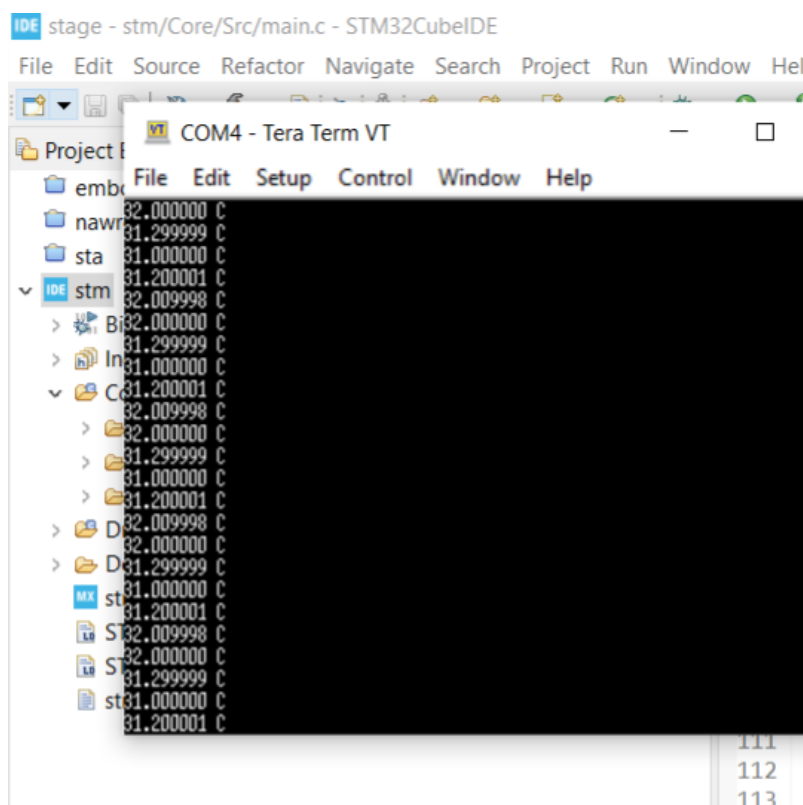


Figure 16:L'affichage sur TeraTerm

#### b. Le langage c :

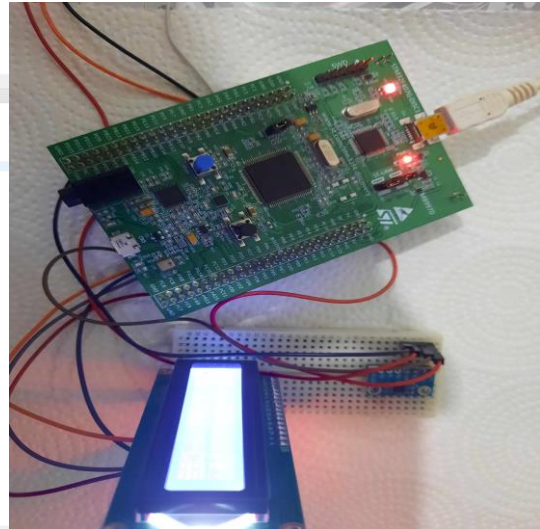
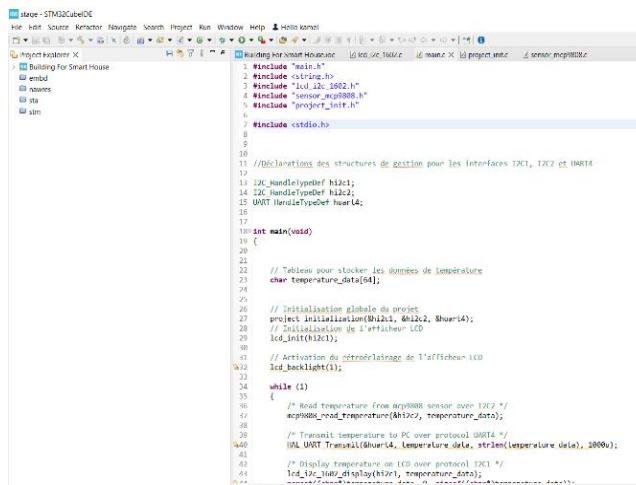


Figure 17:LOGO langage c

Le langage C est un langage de programmation qui appartient au paradigme de programmation impérative. Inventé au début des 1970 dans les Laboratoires Bell pour aider la programmation du système Unix, C est devenu un des langages les plus utilisés. Il n'est pas consacré qu'à la programmation système.

Le langage c est parfaitement adapté pour l'implémentation des différents algorithmes sur des micro-contrôleurs dans des équipements divers tels que les objets connectés, l'électronique médicale de dernière génération, la robotique, la géolocalisation, etc.

c. Le code et son affichage :



**Figure 18:l'affichage du code**



### Figure 19: la température

#### d. Conclusion :

Ce projet s'inscrit dans un contexte professionnel où la surveillance précise de la température est cruciale. Il offre une solution robuste et intégrée pour collecter, afficher et transmettre efficacement les données thermiques, contribuant ainsi à l'amélioration de la qualité et de la sécurité dans diverses applications industrielles et professionnelles.

## Référence bibliographique

<https://www.otto-es.net/>

<https://www.st.com>

<https://www.st.com/en/development-tools/stm32cubeide.html#st-get-software>

[http://www.handsontec.com/dataspecs/module/I2C\\_1602\\_LCD.pdf](http://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf)

<https://fr.wikipedia.org>

<https://pdf1.alldatasheet.com/datasheet-pdf/view/470721/MICROCHIP/MCP9808.html>

<https://deepbluembedded.com/>