UNIVERSITÉ TOULOUSE III PAUL SABATIER

Université de Toulouse

RÉALISATION DES SYSTÈMES BE

Réalisé par :

ELALAOUI Abdelhak

ELMAAZOUZ Najlae

Encadré par :

THIERRY PERISSE

M1 SME
2022 -- 2023

SOMMAIRE

INTRODUCTION

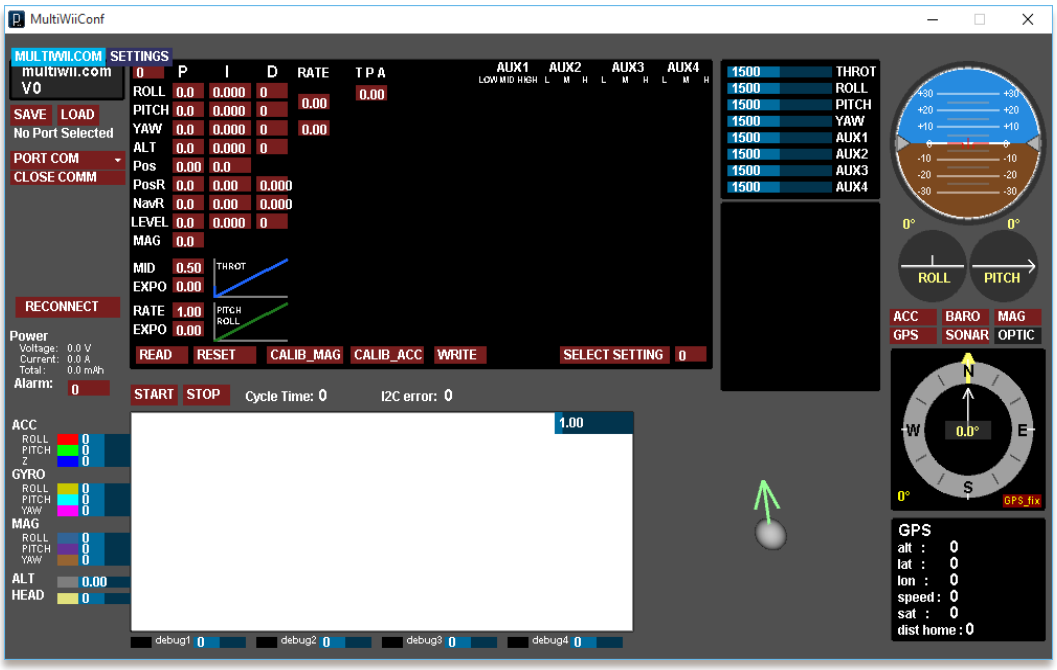
Dans ce TP, nous allons apprendre à manipuler la carte Nucléo STM32 pour les systèmes embarqués et autonomes. Cette carte est un outil très utile pour les ingénieurs en électronique, car elle permet de développer des systèmes électroniques complexes et performants, ainsi que de comprendre les principes de base de la conception de systèmes embarqués. Nous allons explorer les différents capteurs de température, d'accélération et de gyroscope pour mesurer les données environnementales et les mouvements du système. En utilisant ces capteurs, nous pourrons obtenir des données précises et fiables pour les systèmes embarqués, ce qui est essentiel pour de nombreuses applications telles que l'automobile, l'aérospatiale, la robotique et bien d'autres.

En outre, nous allons également explorer les protocoles de communication tels que WIFI ou SPI pour échanger les données collectées vers le système. Nous allons également voir comment ces données peuvent être traitées et implémentées dans le système pour améliorer les performances et la précision. Tout ceci est important pour comprendre les principes fondamentaux de la conception de systèmes embarqués et pour être en mesure de les appliquer dans le monde réel.

Enfin, nous aborderons également les différentes façons de configurer et de programmer la carte selon nos besoins spécifiques. Ceci est essentiel pour tout ingénieur en électronique travaillant dans le domaine des systèmes embarqués, car cela permet de personnaliser et d'optimiser le système pour des applications spécifiques. Nous verrons également comment ces compétences peuvent être appliquées dans des projets réels, tels que la conception de robots autonomes ou de systèmes de contrôle industriels.

I) Description du système :

Le système que nous allons essayer d'implémenter est un système sophistiqué qui deux capteurs de haute précision pour collecter et transmettre des données premier capteur, le SHT31, est spécialement conçu pour mesurer la tempéra l'humidité avec une grande précision. Le deuxième capteur, l'IMU 9DOF, est un cap d'accélération et de gyroscope qui peut mesurer la position et les mouvements av extrême précision.

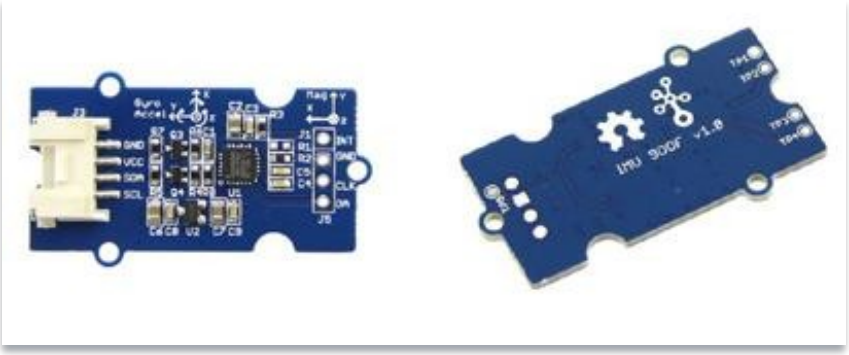


II) Matériels utilisés

1. CAPTEUR IMU 9DOF :

Le Grove IMU 9DOF v2.0 est une version améliorée de Grove IMU 9DOF v1.0. Ce module de suiv mouvement à 9 axes offre une haute performance grâce au MPU-9250, un dispositif de mouvement à 9 axes intégré conçu pour répondre aux exigences de faible consommation d'éne faible coût et de haute performance des équipements électroniques grand public.

Ce dispositif est très populaire auprès des constructeurs de smartphones, de tablettes et de cap
portables. En plus du MPU-9250, le Grove - IMU 9DOF v2.0 est doté de trois ADC 16 b
numérisation des sorties du gyroscope, trois ADC 16 bits pour la numérisation des sorties de
l'accéléromètre et trois ADC 16 bits pour la numérisation des sorties du magnétomètre. Grâce à
ces fonctionnalités, le Grove - IMU 9DOF est un choix idéal pour les projets qui nécessitent un su
mouvement de haute précision.



Ce module intègre une centrale IMU 9 axes MPU-9250 (accéléromètre 3 axes + compas
gyroscope 3 axes) qu'il vous sera possible de raccorder à un Arduino, Raspberry…

Voici les spécifications de l’IMU 9DOF:

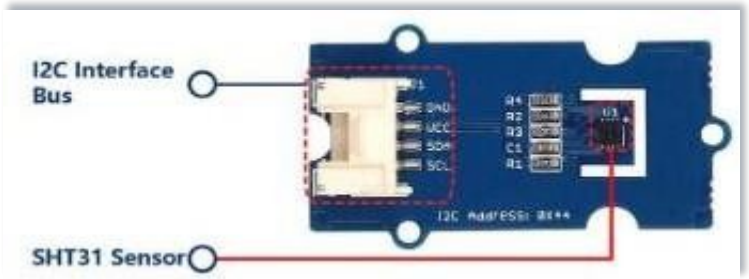
- I2C/SPI interface
- Auxiliary I2C
- Low Power Consumption
- 400kHz Fast Mode I2C for communicating with all registers
- Digital-output 3-Axis angular rate sensors (gyroscopes) with a user-programmable
  scale range of ±250, ±500, ±1000, and ±2000°/sec
- Digital-output 3-Axis accelerometer with a programmable full scale range of ±2g,
  ±8g and ±16g
- Digital-output 3-Axis accelerometer with a full scale measurement range is ±4800
- I2C Address: 0x68

Pour pouvoir reçevoir les mesures de ce capteur il faut mettre la bonne adresse dans le code qui est le 0x68 :

→ static const uint8_t CAPTEUR_ADRS = 0x68 << 1;

2. Capteur de température et d'humidité SHT31 :

Le Capteur de Température et d'Humidité SHT31 Grove est un capteur de qualité supéri pour mesurer l'humidité relative et la température. Ce capteur est capable de mesurer l'humidité relative avec une précision remarquable de ±2 %, ce qui en fait l'un de d'humidité les plus précis disponibles sur le marché.



Ce capteur est également très performant en termes de mesure de la température, avec précision de ±0,3 degrés, même dans des conditions extrêmes allant de -40 à 125 degr outre, ce capteur est également très esthétique avec un design épuré qui convient à tou d'environnement. Il est également facile à installer et à utiliser, ce qui en fait un choix id les utilisateurs débutants et avancés.

En somme, le Capteur de Température et d'Humidité SHT31 Grove est un choix sûr et fi mesurer l'humidité et la température avec précision et facilité.

Voici les caractéristiques de ce capteur :

- Basé sur un capteur Sensirion SHT31, il offre un temps de réponse faible et une m précise.
- Facile d'utilisation grâce à la connectique Grove
- Nécessite une extension de carte : Shield Grove (#categorie-14)
- Capteur compatible avec toute carte de 3.3 à 5V
- Température : Plage de mesure de -40°C à +125°C
- Précision: 0.3°C
- Humidité : Plage de mesure de 0 à 100% RH
- Précision: 2% RH
- Dimensions du module : 40mm x 20mm x 11mm
- Poids : 10g

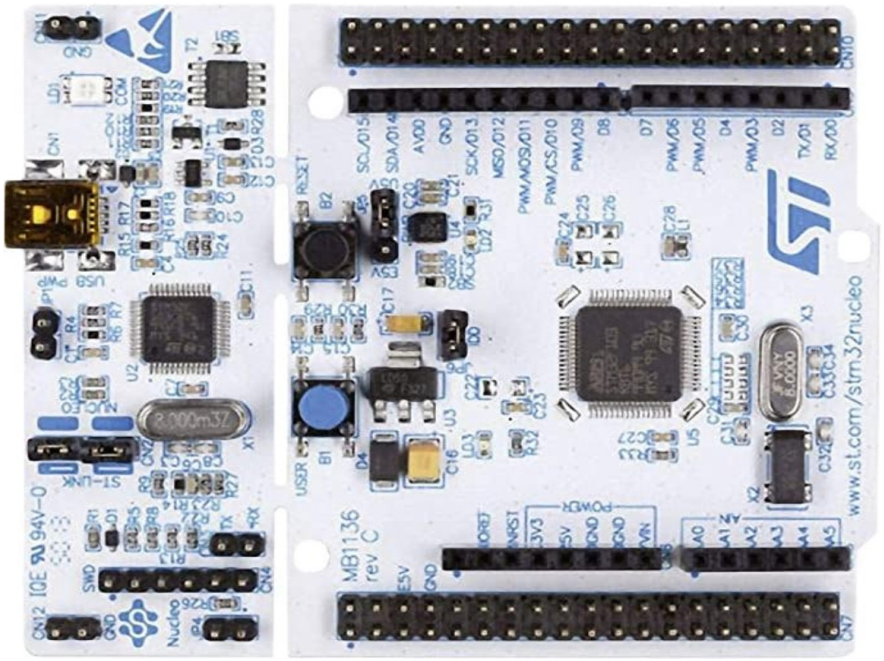
Pour ce capteur, il utilise le protocole de communication I2C pour l'échange de commun comme l'autre capteur.

Voici un tableau qui représente les différentes spécifications de capteur :

Input voltage (VCC)	3.3 volts or 5 volts
I/O Logic Level	3.3 volts or 5 volts based on VCC
Operating Current	100 µA
Operating Temperature	-40–125 °C
Temperature Sensor Range	-40–125 °C, with ±0.3°C accuracy
Humidity Sensor Range	0% - 100%(Relative Humidity), with ±2% accuracy
Sensor Chip	SHT31(Datasheet)
Interface Bus	I^2C
Weight	4 g (for breakout board), 9 g for whole package each piece
Dimensions	40(length)×20(width) mm

3. Présentation de la carte Nucléo STM32 :

Les cartes Nucleo STM32 constituent une avancée majeure dans les stratégies d'
développement de ST. Elles représentent un moyen simple et efficace pour les développ
d'accéder à la gamme de microcontrôleurs STM32. En effet, ces cartes autaires
comportent un débogueur STLINK/V2-1 intégré compatible avec un port COM virtu
programmation glisser -déposer.



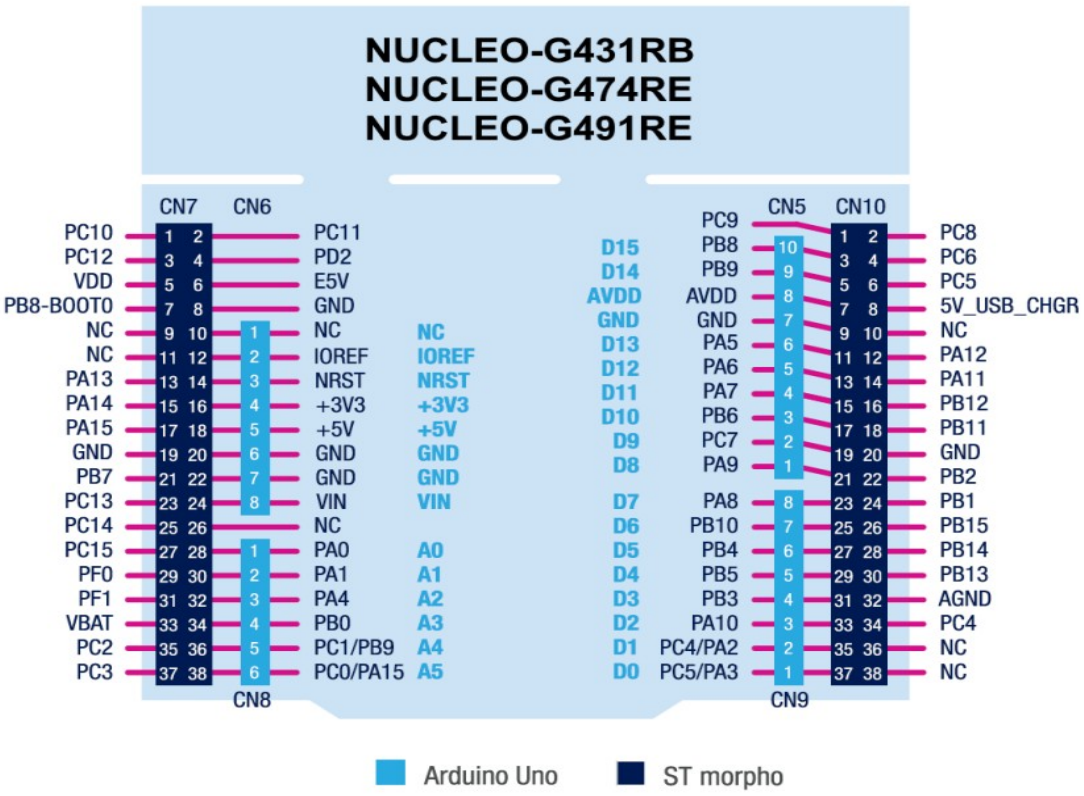
De plus, les cartes Nucleo STM32 sont dotées de connecteurs Arduino™ et du n
connecteur ST Morpho, qui donnent accès à toutes les entrées/sorties des microc
intégrés dans un boîtier 64 broches. Cette polyvalence permet aux développeurs de trav
plus facilement et plus efficacement.

Les cartes Nucleo STM32 sont compatibles avec les environnements de développement
(IDE) IAR, Keil et GCC. De plus, leur prise en main est un véritable jeu d'enfant
bibliothèques de logiciels fournies par ST. Les développeurs sont ainsi en mesure de trav
sur leurs projets plus rapidement et plus efficacement.

Voici les caractéristiques de la carte STM32 :

- Deux types de ressources d'extension:
    - Connectivité Arduino Uno Rev 3
    - En-têtes de broches d'extension Morpho de STMicroelectronics pour un complet à tous les E/S STM32
- Débogueur/programmeur ST-LINK/V2-1 intégré avec connecteur SWD
    - Commutateur de mode de sélection pour utiliser le kit en tant que ST-LINK/V autonome
- Alimentation flexible de la carte
    - VBUS USB ou source externe (3.3V, 5V, 7V à 12V)
    - Point d'accès de gestion de l'alimentation
- Trois LEDs:
    - Communication USB (LD1), LED utilisateur (LD2), LED d'alimentation (LD3)
- Deux boutons-poussoirs: USER et RESET
- Capacité de ré-ennumération USB: trois interfaces différentes prises en charge sur
    - Port COM virtuel
    - Stockage de masse
    - Port de débogage
- Bibliothèque logicielle HAL gratuite et complète comprenant une variété d'exempl logiciels
- Pris en charge par un large choix d'environnements de développement intég comprenant IAR, Keil, des IDE basés sur GCC
    - USB communication (LD1), user LED (LD2), power LED (LD3)

Voici le brochage de la carte STM32

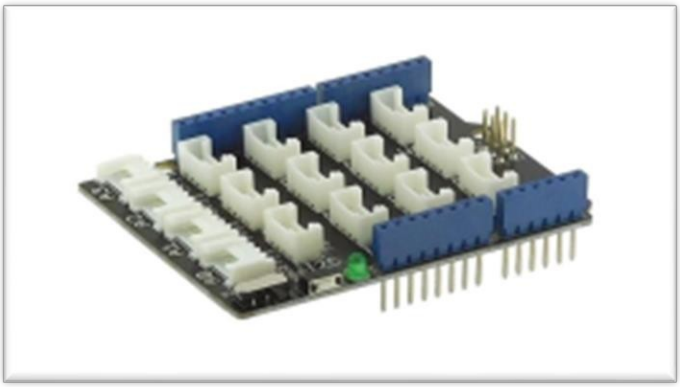


Enfin, l'intégration mbed donne accès à encore plus de ressources communautair permet aux développeurs de bénéficier d'un large éventail de ressources et de collabore facilement avec d'autres membres de la communauté. Les cartes Nucleo STM32 sont do solution complète et efficace pour les développeurs de tous niveaux qui souhaitent trava des projets STM32.

4. Afficheur LCD :

Un afficheur LCD est un type d'écran qui utilise la technologie des cristaux liquides pou afficher des informations. Afficheur LCD 2x16 caractères rétro-éclairé se raccordant via I2C sur un microcontrôleur en utilisant un câble Grove et un Shield. Ce composant est compatible avec une plage de tension d'alimentation entre 3,3 et 5,5

5. Module Grove Base Shield :



Le module Grove Base Shield est une carte d'interface qui facilite la connexion sans so rapide des capteurs et actionneurs Grove à une carte compatible. Il peut être utilisé av cartes telles que l'Arduino Uno, Leonardo, Seeeduino, ainsi que certaines cartes STM32 64. Ce Shieldest doté d'un sélecteurqui permetde le rendrecompatibleavecdes microcontrôleurs fonctionnant à des tensions de 3,3 Vcc ou 5 Vcc. Il dispose de 16 conn 4 broches, dont 4 sont des entrées analogiques, 7 sont des entrées/sorties logiques, 4 s interfaces I2C et 1 est une interface UART.

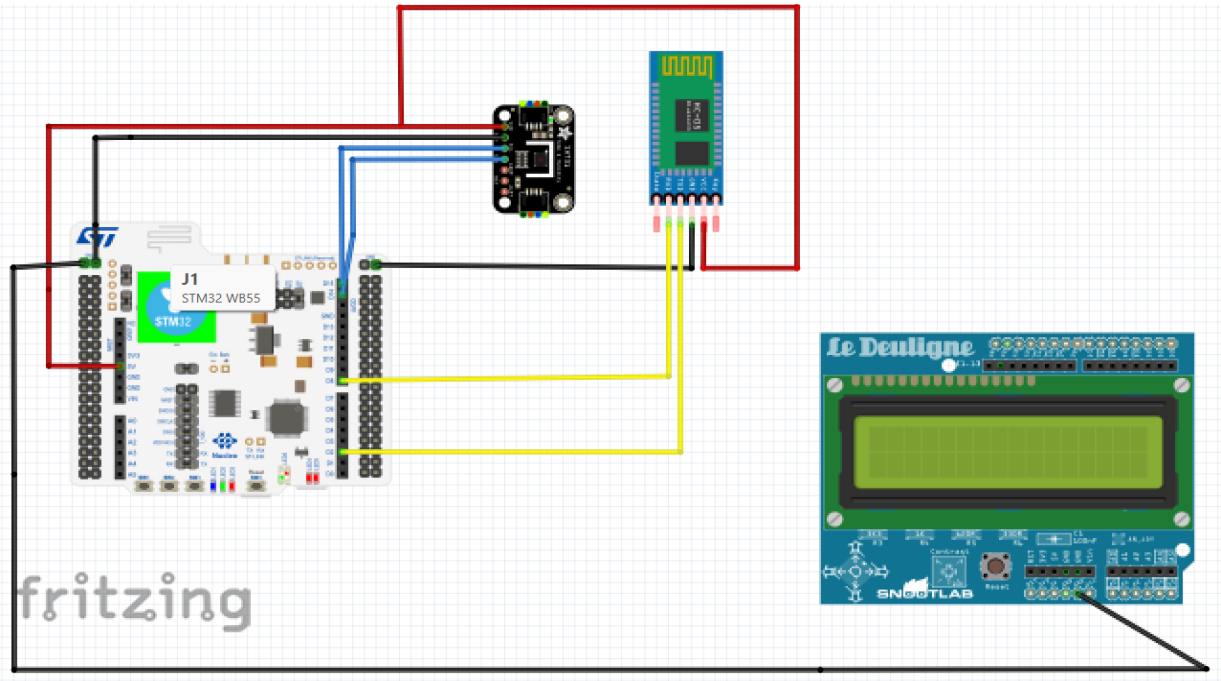
III) PROJET DE BASE :

1) STM32 +SHT31+LCD :

Dans ce TP de base, l'objectif est de mesurer la température et l'humidité à l'aide du ca
SHT31 dans n'importe quel environnement, puis d'afficher les valeurs mesurées sur
afficheur LCD I2C.

A) Schéma de câblage :

Grâce à l'utilisation du logiciel Fritzing, nous avons pu effectuer le câblage de notre pro
de base en suivant les indications fournies dans la figure 6 ci-dessous :



B) Programmation :

Une fois le câblage réalisé, nous avons utilisé le logiciel STM32CubeMX pour co
broches nécessaires à notre projet et générer le code en C pour afficher la valeur fin ale
l'afficheur LCD en suivant le programme disponible dans notre dépôt **GitHub**.

Voici une explication brève du code :

1) Les variables **buf**, **ret**, value, temp, **Entier_part**, **Decimal_part**, et **umid** sont décla
stocker différentes valeurs utilisées dans le programme.

2) Les valeurs de commande pour la lecture des capteurs de température et d'humidité
stockées dans **buf**[0] et **buf[**1].

3) La fonction **HAL_I2C_Master_Transmit** est appelée pour envoyer les commandes d
aux capteurs via le bus I2C.

4) Si la transmission I2C échoue, la chaîne de caractères **"erreur_T**!!" est copiée dans

5) Si la transmission I2C réussit, la fonction **HAL_I2C_Master_Receive** est appelée pou
les données des capteurs.

6) Si la réception I2C échoue, la chaîne de car**a**ctères "**erreur_R**!!" est copiée dans **buf**

7) Si la réception I2C réussit, les valeurs de température et d'humidité sont décodées à
des octets reçus et calculées à l'aide de formules spécifiques.

8) Les valeurs de température et d'humidité sont ensuite formatées dans la cha
caractères buf avec l'aide de la fonction **sprintf**.

9) Les valeurs de température et d'humidité sont affichées sur l'écran LCD en u
fonctions **lcd_position** et **lcd_print**.

10) Le contenu de buf est transmis via UART pour affichage sur un autre dispositif.

11) Une temporisation de 1 seconde est appliquée avant que le code ne boucle et ne ré
processus.

La fonction **SystemClock_Config** est une configuration du système qui initialise les os
et les horloges pour le microcontrôleur.

La fonction **Error_Handler** est appelée en cas d'erreur fatale, et dans cet exemple, elle
les interruptions et provoque une boucle infinie pour bloquer le programme en cas d'er

C) Visualisation sur Picoscope:

Pour cette étape, nous avons employé un oscilloscope pour vérifier les résultats obtenus capteur SHT31 et valider les informations I2C mentionnées dans la datasheet. La dessous présente un exemple des résultats obtenus à l'aide de Pico scope.

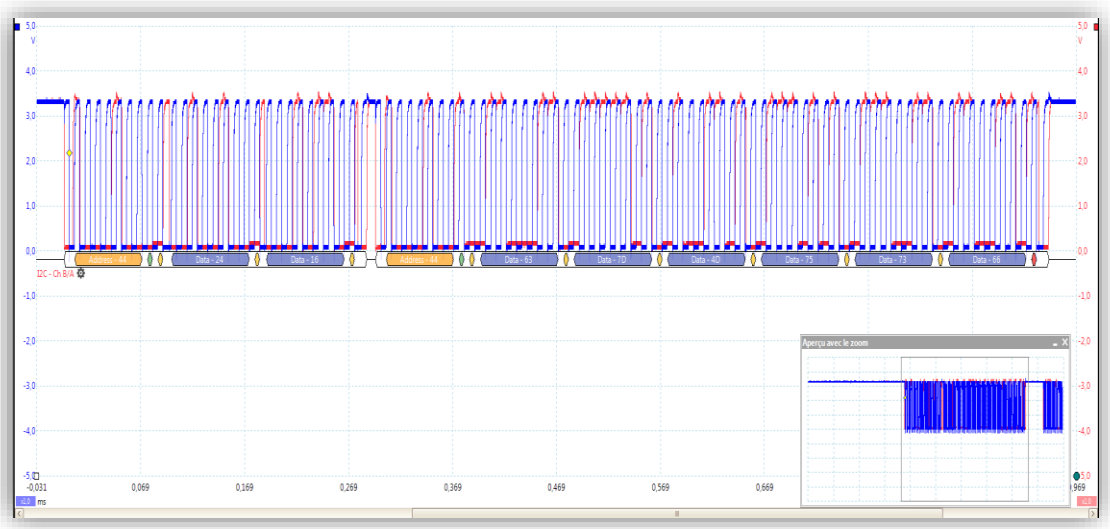
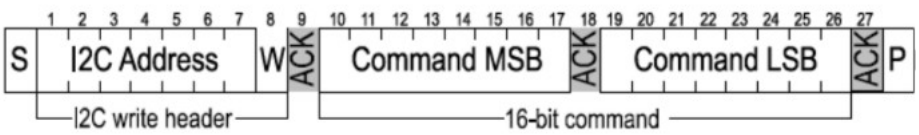


Figure : vérification sur Pico Scope du capteur SHT31

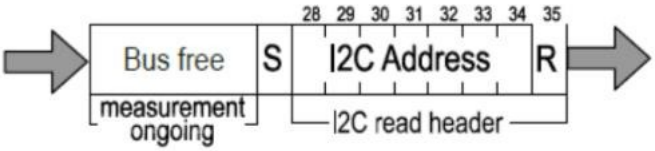
Une fois le capteur connecté au shield et alimenté il est prêt a recevoir des commandes transmission commence par un condition START selon les standard I2C, l'en-tête d'écrit en I2C,

donc 7 bits adresse (l'adresse du sht31 est 0x44) plus 0 comme bit d'écriture, suivi de de

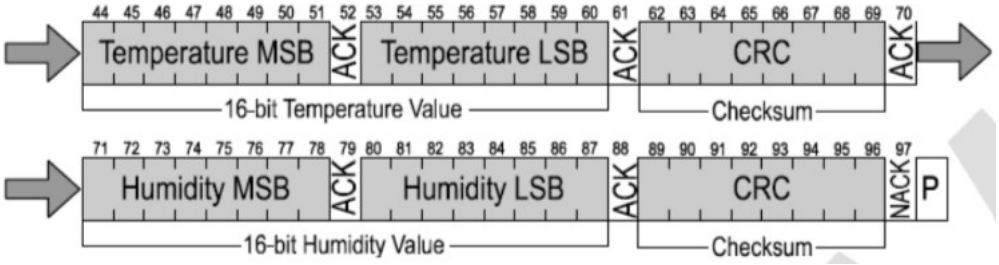
commande de mesure



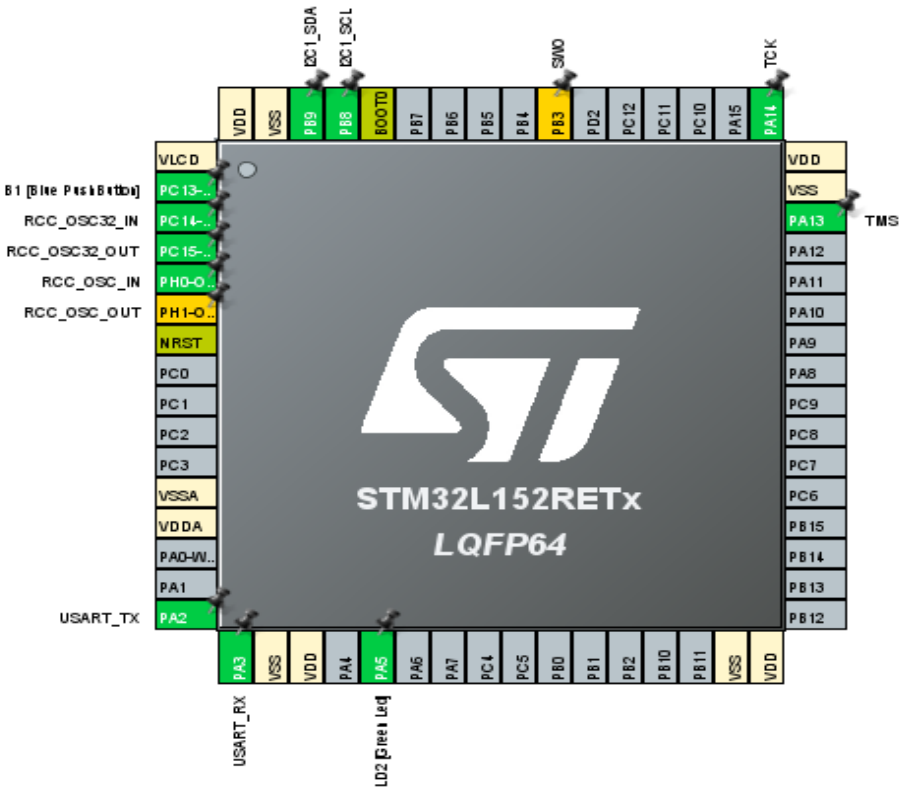
Pour lire les valeurs mesurées on envoie la commande de lecture avec les 7bits d’adress
pour la lecture.



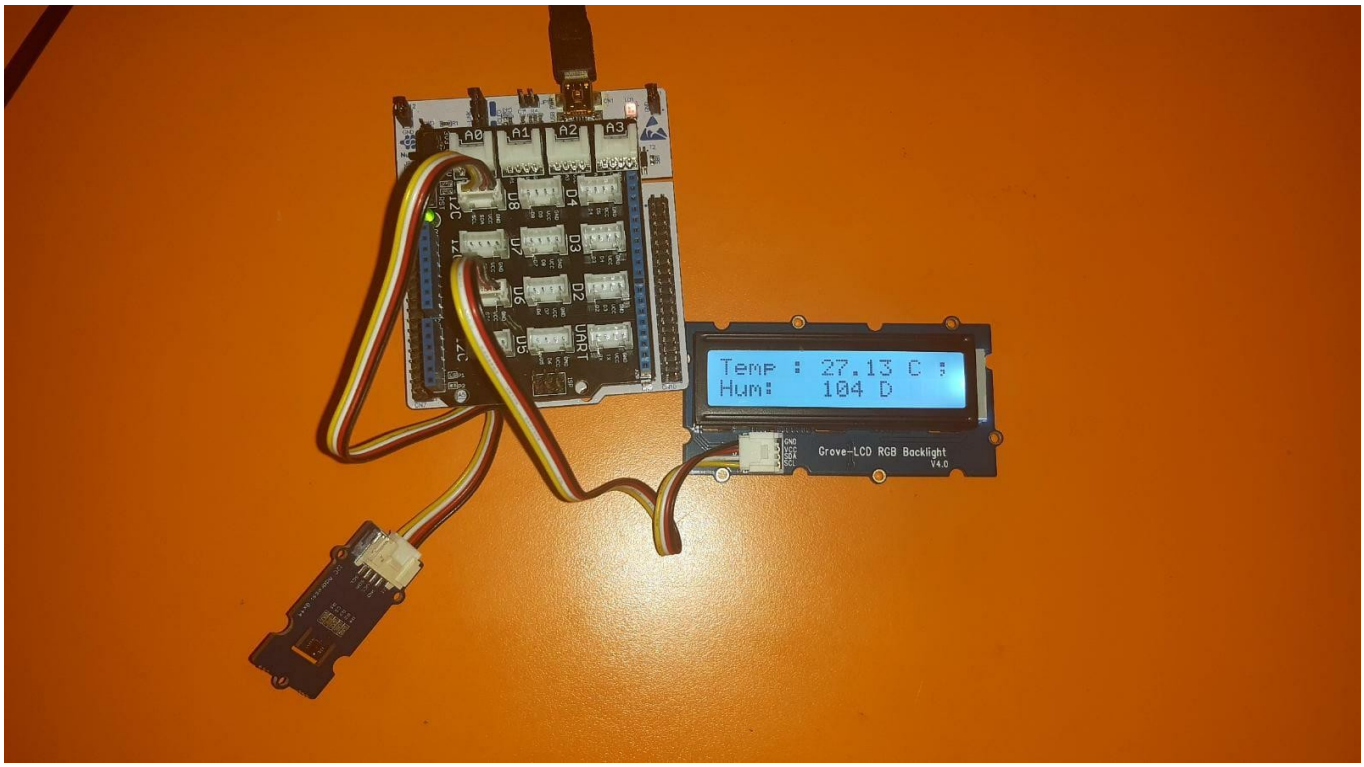
Puis on reçois les 16bits de mesure de température et 16bits de mesure d’humidité, et à
de chaque mesure un CRC.



Donc pour utiliser ce capteur et LCD sur CubeMX, on active les pins PB9 et PB8 comme
SCL respectivement pur communiquer avec les ports I2C existant sur le shield Grove. On
aussi le bus UART pour communiquer avec le pc et afficher les résultats .

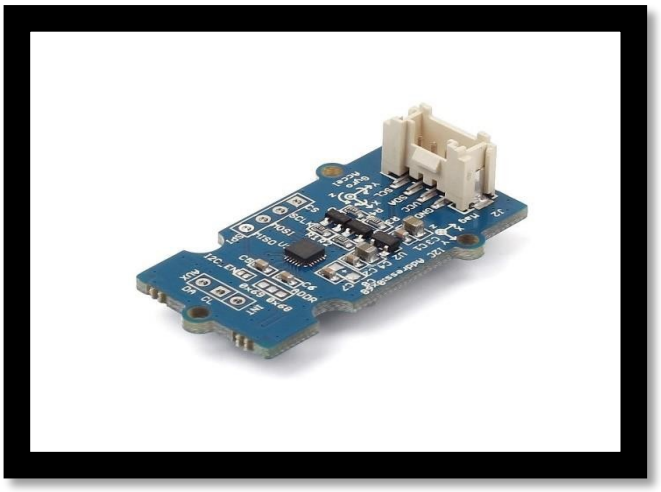


D) Résultat :



2) IMU 9DOF +LCD+ STM32 :

Le Grove - IMU 9DOF v2.0 est une version améliorée du Grove - IMU 9DOF v1.0 module de suivi de mouvements à 9 axes basé sur le MPU-9250. Ce dispositif d mouvements à 9 axes est conçu pour répondre aux exigences de faible consommation d de faible coût et de haute performance des équipements électroniques grand public, tels smartphones, les tablettes et les capteurs portables. Le MPU-9250, qui est intégré au m dispose de trois ADC 16 bits pour la numérisation des sorties du gyroscope, de l'accéléro et du magnétomètre, offrant ainsi une précision plus élevée dans la mesure des mouver



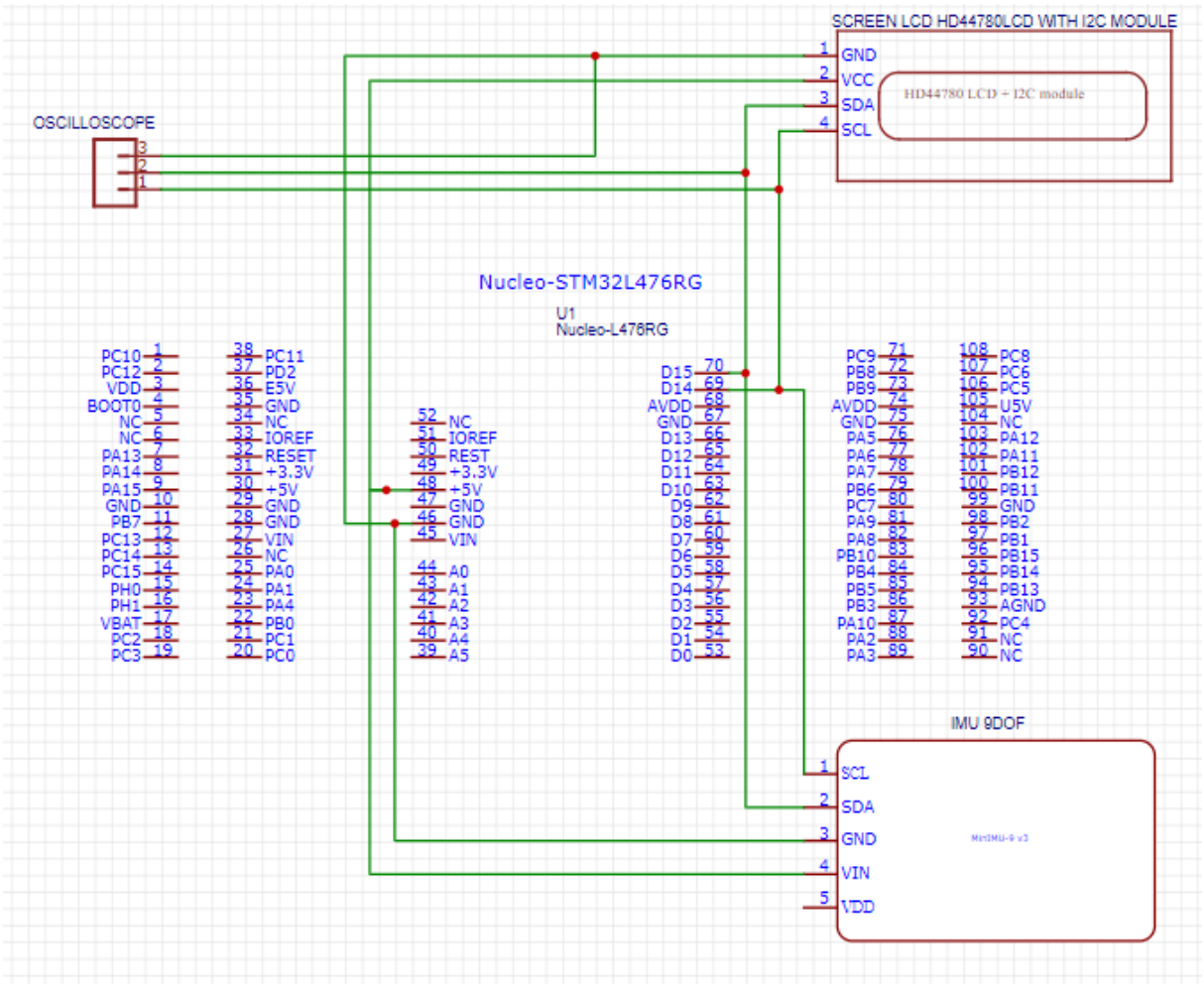
En outre, ce module est très polyvalent et peut être utilisé dans une variété d'applicatio
compris la réalité virtuelle et augmentée, les drones, les robots, les jouets électron
équipements de fitness, les systèmes de navigation inertielle, et bien plus encore. Il peu
utilisé pour mesurer l'accélération, la vitesse angulaire et le champ magnétique, ce qui p
une grande variété d'applications.

Le Grove - IMU 9DOF v2.0 a également une interface I2C qui permet une connexion facil
microcontrôleur ou à un ordinateur. De plus, il est compatible avec tous les modules Gro
qui permet une intégration facile dans des projets existants.

En bref, Le Grove - IMU 9DOF v2.0 est un module de suivi de mouvements à 9
performance qui offre une précision élevée et une grande polyvalence pour une
d'applications électroniques.

A) Schéma de câblage :

Voici le schéma de cablage pour le STM32 avec le capteur IMU 9DOF et avec la LCD de
cette fois-çi à l'aide d'un autre logiciel qui s'appelle EasyEDA qui sert à faire le
schémas électriques et aussi l'implémentation de la PCB ou de la carte électronique:

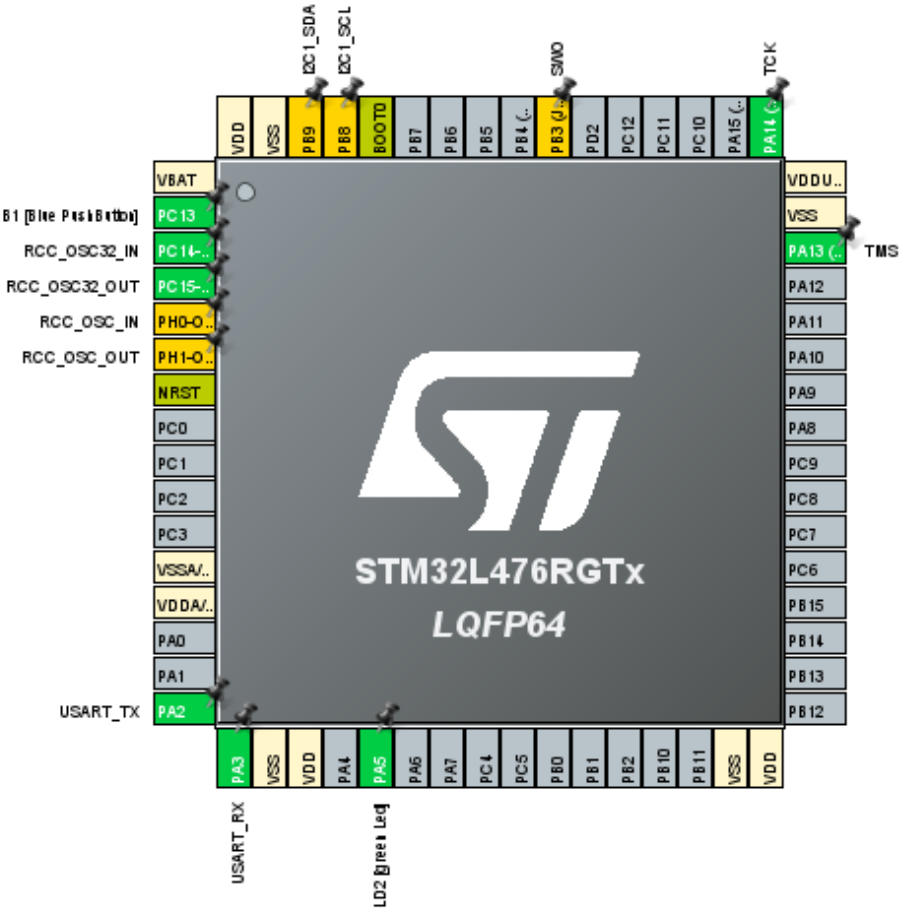


B) Programmation :

Dans la fonction **SETUP**, plusieurs étapes d'initialisation sont réalisées. Tout d'abord, le
est connecté à l'aide de la fonction **Wire.begin()** pour permettre la communicat
périphériques I2C. Ensuite, la communication série est initialisée avec une vitesse de
transmission de 9600 bauds à l'aide de la fonction **Serial.begin(9600)**, ce qui permet la
communication avec un périphérique connecté tel qu'un ordinateur. Un message est imp
sur le moniteur série pour indiquer l'initialisation des périphériques I2C. Le périph
accelgyro est ensuite initialisé et un test de connexion est effectué à l'aide de
accelgyro.testConnection(). Le résultat du test est imprimé sur le moniteur série pou
si la connexion au périphérique MPU9250 a réussi ou non. Un délai de 1000 milliseconde
seconde) est ajouté pour permettre l'affichage des messages. Enfin, une ligne vide est i
pour créer une séparation visuelle.

La fonction **MAIN** contient le code principal qui s'exécute de manière répétitive. Elle cor
par appeler différentes fonctions pour obtenir des données de différents capteurs, tels q
données d'accélération, de gyroscope et de boussole calibrées. La fonction
getCompassDate_calibrated() est appelée avant la fonction **getHeading()** car
données calibrées nécessaires pour un calcul précis de l'angle. La fonction **getTiltHead**
ensuite appelée pour calculer l'angle entre le nord magnétique et la projection de l'axe X
dans le plan horizontal.

Après l'intégration de l'I2C dans la partie de configuration comme le premier capteur da
partie graphique de STM32_CUBE_IDE :



Commençons par l'initialisation du MPU6050. Pour initialiser le capteur, nous devons effe
les actions suivantes:

Nous devons vérifier si le capteur répond en lisant le **registre "WHO_AM_I (0x75)"**. Si
répond avec 0x68, cela signifie qu'il est disponible et prêt à fonctionner.

On utilise la fonction **HAL_I2C_Mem_Read** pour lire directement à partir du registre de donné :

```
HAL_I2C_Mem_Read (&hi2c1, MPU6050_ADDR,WHO_AM_I_REG,1, &check, 1, 1000);
```

Ensuite, nous allons réveiller le capteur de son mode de sommeil afin de prendre des me Nous pouvons y parvenir en écrivant dans le **registre "PWR_MGMT_1 (0x6B)"**. Ce reg responsable de la gestion du mode d'alimentation du capteur et est situé à une spécifique dans la mémoire du capteur. Une fois que nous écrivons dans ce registre, le c entrera en mode actif et commencera à prendre des mesures.

```
Data = 0;

HAL_I2C_Mem_Write(&hi2c1, MPU6050_ADDR, PWR_MGMT_1_REG, 1,&Data, 1, 1000);
```

On Configure maintenant les registres de l'accéléromètre et du gyroscope et pour cela, n modifier les registres « GYRO_CONFIG (0x1B) » et « ACCEL_CONFIG (0x1C) ».

4.5 Register 28 – Accelerometer Configuration ACCEL_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

AFS_SEL	Full Scale Range
0	± 2*g*
1	± 4*g*
2	± 8*g*
3	± 16*g*

XA_ST When set to 1, the X- Axis accelerometer performs self test.
YA_ST When set to 1, the Y- Axis accelerometer performs self test.
ZA_ST When set to 1, the Z- Axis accelerometer performs self test.

On peut lire 1 OCTET de chaque registre séparément ou on peut simplement lire 6 OCTETS au t partir du registre ACCEL_XOUT_H

4.17 Registers 59 to 64 – Accelerometer Measurements ACCEL_XOUT_H, ACCEL_XOUT_L, ACCEL_YOUT_H, ACCEL_YOUT_L, ACCEL_ZOUT_H, and ACCEL_ZOUT_L

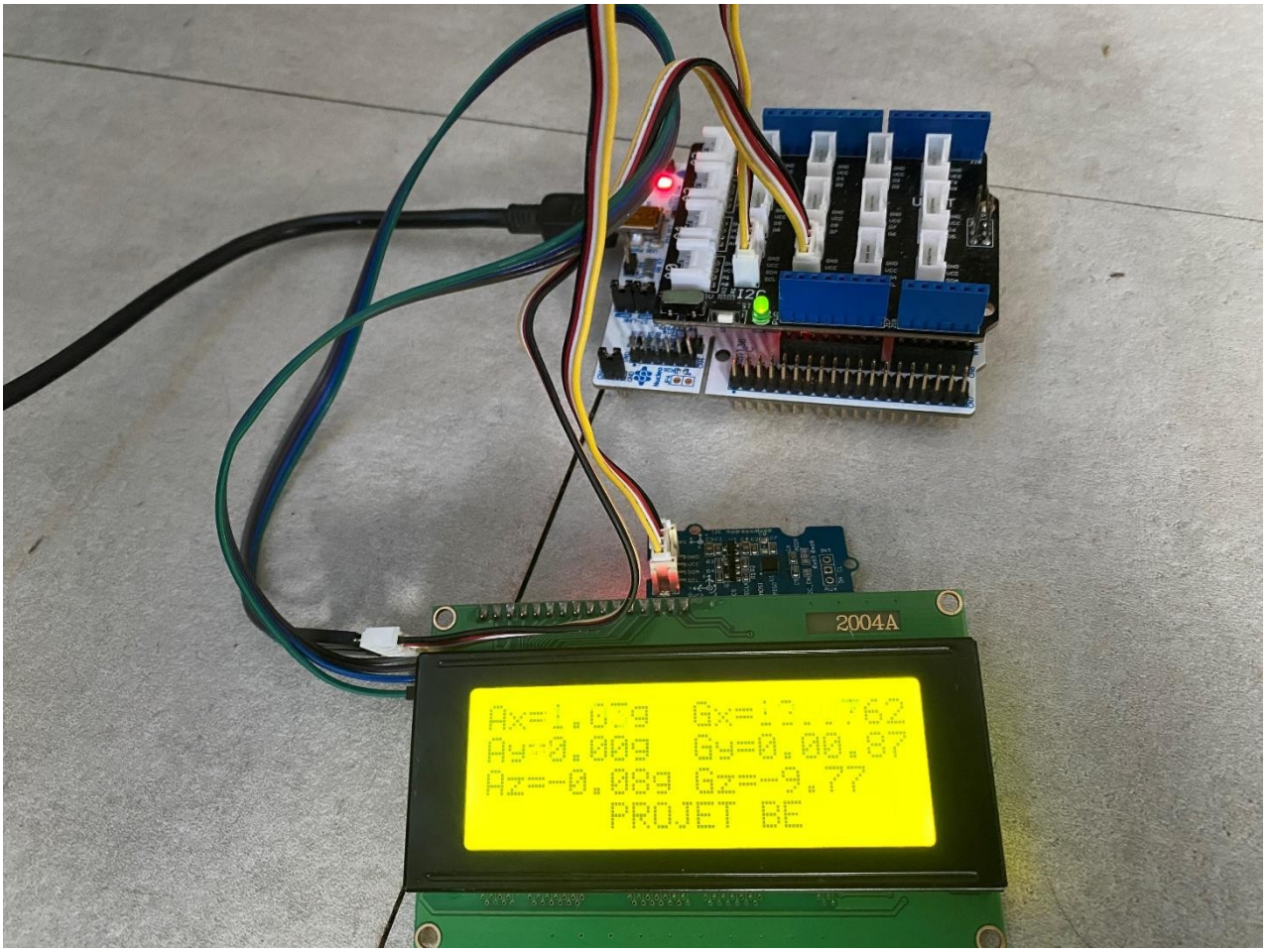
Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

AFS_SEL	Full Scale Range	LSB Sensitivity
0	±2*g*	16384 LSB/*g*
1	±4*g*	8192 LSB/*g*
2	±8*g*	4096 LSB/*g*
3	±16*g*	2048 LSB/*g*

C) Résultat :

Voici le résultat final de l'utilisation de ce capteur et les afficher sur l'écran :



CONCLUSION

Dans l'ensemble, ce TP a été très instructif car nous avons eu l'opportunité de découvrir les principes fondamentaux de la conception de systèmes embarqués en utilisant la carte Nucléo STM32 et différents capteurs tels que le capteur de température, d'accélération et de gyroscope. Nous avons appris comment ces capteurs peuvent être utilisés pour mesurer les données environnementales et les mouvements du système, ainsi que comment ces données peuvent être échangées via des protocoles de communication tels que WIFI ou SPI pour améliorer les performances et la précision. Nous avons vu comment configurer et programmer la carte pour des applications spécifiques, ce qui est essentiel pour tout ingénieur en électronique travaillant dans le domaine des systèmes embarqués.

En outre, ce TP nous a permis de comprendre comment ces compétences peuvent être appliquées dans des projets réels tels que la conception de robots autonomes ou de systèmes de contrôle industriels. Nous avons donc pu voir comment les principes fondamentaux de la conception de systèmes embarqués peuvent être utilisés dans le monde réel pour créer des systèmes électroniques complexes et performants.

En somme, ce TP a été très enrichissant car nous avons pu découvrir de nouvelles compétences et les appliquer dans des projets concrets. Nous sommes désormais mieux équipés pour travailler dans le domaine des systèmes embarqués et pour relever les défis que ce domaine peut présenter.