# SHTC3 Compte rendu

04/12/2024 – Avant de faire quoi que ce soit avec le module SHTC3 il faut que je comprenne son fonctionnement.

### Analyse visuelle :

Donc le module est créé par Adafruit, il utilise le capteur SHTC3 qui est un capteur de température et d’humidité.

Je vois qu’il utilise l’I2C pour communique car il y a une sérigraphie qui le confirme et la présence des pins « SDA » et « SCL » m’ont aussi indiqué que je vais utiliser l’I2C.

J’ai trouvé une documentation sur le SHTC3 : <https://www.mouser.fr/datasheet/2/682/seri_s_a0003561073_1-2291167.pdf>

### Analyse de la datasheet :

Il possède un convertisseur analogique vers numérique, la doc confirme qu’il utilise l’I2C en fast mode.

Il couvre une humidité de 0% à 100% et une température de -40°C à 125°C.

Sa marge d’erreur est de +-2% pour l’humidité et +-0.2°C.

Sa plage d’alimentation est de 1.62V jusqu’à 3.6V, donc il faudra que je le branche vers l’alimentation 3.3V de l’Arduino.

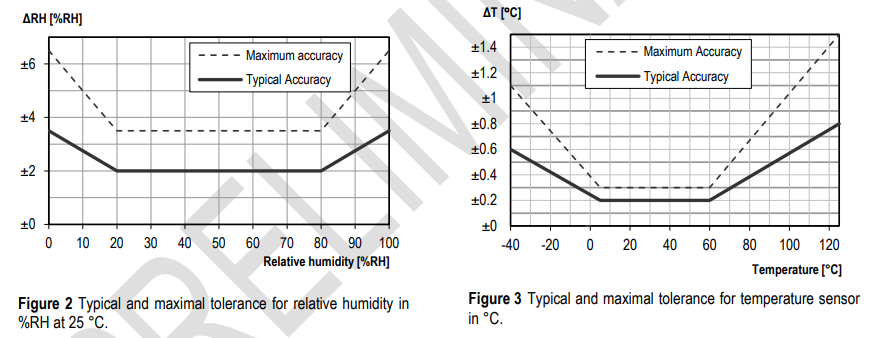


Figure Graphiques de ses précisions en fonction de l'humidité/température.

La datasheet recommande de l’utiliser entre 5°C et 60°C et 20% à 80% d’humidité.

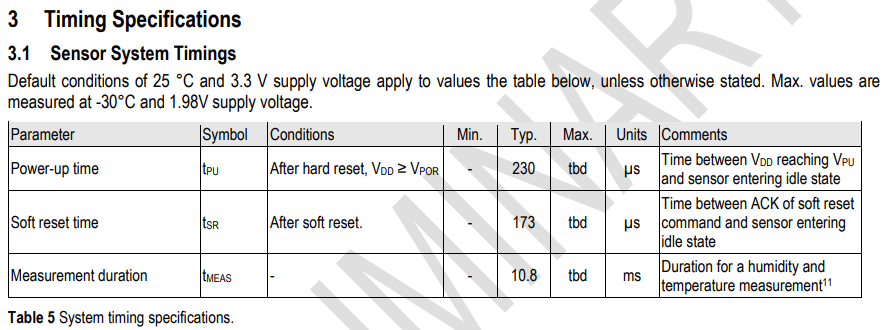


Figure Tableau avec des valeurs de temps

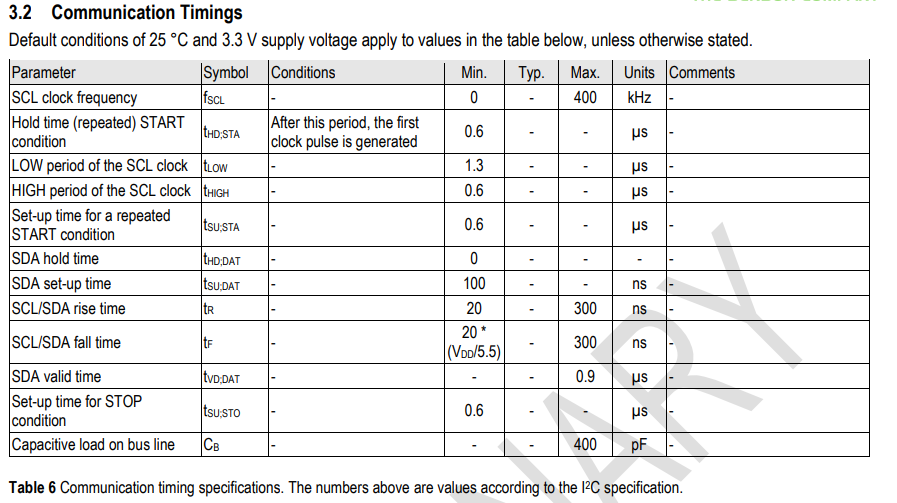


Figure Temps de communication

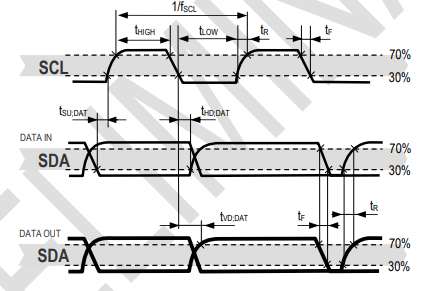


Figure Chronogramme des données

Le SHTC3 à une fréquence max de son horloge de 400kHz.

Pour éviter des conflits de signaux il faut utiliser des résistances pour le « SDA » et le « SCL », en regardant le module je remarque que les résistances sont déjà mises.

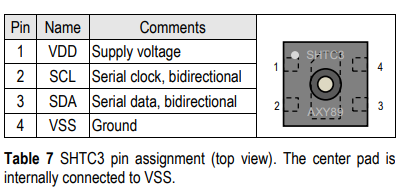


Figure Pins du capteur

### Différentes commandes :

Qui dit I2C dit datas à transmettre et donc des commandes.

L’adresse de ce composant/module est **0x70**.

|  |  |
| --- | --- |
| Commande | Code hexadécimale |
| Sommeil | 0xB098 |
| Reveil | 0x3517 |
| Software reset | 0x805D |
| Read ID register | 0XEFC8 |

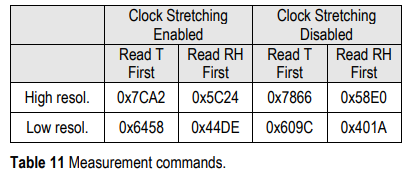
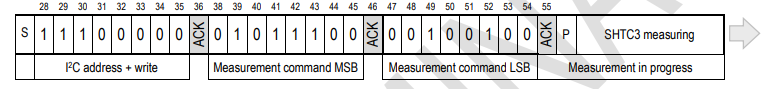


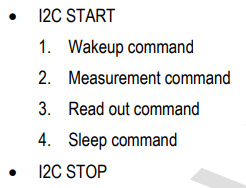
Figure Codes des commandes de mesure.



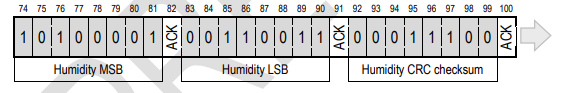
Pour comprendre le tableau des commandes de mesures j’ai pris cette exemple d’envoie d’une commande de mesure et j’ai traduit le binaire vers l’hexadécimal.

Dans cette trame la commande envoyée est 0x5C24 ce qui correspond à lire l’humidité en premier et en haute résolution.

Le « clock stretching » qui est l'étirement d'horloge, c’est une fonctionnalité importante du protocole de communication I2C qui permet aux appareils esclaves de contrôler le timing de la communication. Il permet au périphérique esclave d'arrêter temporairement le signal d'horloge pour demander plus de temps pour terminer son fonctionnement.

Chaque cycle de mesure contient un ensemble de quatre commandes, initié par la condition I2C START et terminé par la condition I2C Condition d'arrêt.

### Conversion :



Pour convertir cette humidité il faut traduire le binaire en décimale ce qui donne :

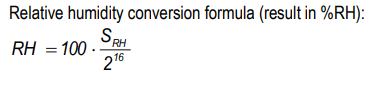
MSB = 161

LSB = 51

Puis décaler le MSB de 8bits et l’additionner avec le LSB :

(161\*256)+51 = 41 267

Pour finir j’applique une formule donner par la documentation :



Ce qui donne pour cette exemple une humidité de 62.9%

Il en va de même pour la température juste la formule change :

