

TP2.2 Bus I2C via Raspberry PI Sense Hat



1 Introduction :

(Sources : <http://www.framboise314.fr/sense-hat-un-tour-dans-les-etoiles/> et <https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat>)

Le Sense HAT est une carte d'extension pour Raspberry PI qui permet de lui ajouter de nombreux types de capteurs et s'installe directement sur la carte principale de l'ordinateur au moyen des 40 broches du GPIO.

Tout le monde peut utiliser une Sense Hat afin de créer un Astro Pi (Système à base de Raspberry, envoyé en 2017 sur la station internationale avec Thomas Pesquet) sur Terre avec un Raspberry Pi standard modèle A+, B+ ou Raspberry Pi 2 modèle B+ qui apporte plusieurs capteurs reposant sur des circuits intégrés, ce qui permet de réaliser de nombreux types d'expériences, d'applications, voir de jeux.

Les capteurs permettent aux utilisateurs de lire l'orientation (lacet, tangage et roulis) au moyen d'un accéléromètre, d'un gyroscope 3D et d'un magnétomètre; pression; humidité et température.

La matrice LED permet d'afficher les données de différents capteurs, notamment en indiquant la direction du nord géomagnétique au moyen du magnétomètre. Elle permet également de jouer à des jeux tels que Tetris, Pong et Snake avec le joystick.

D'autre part, Le joystick permet à un utilisateur humain d'interagir avec les programmes s'exécutant sur Raspberry Pi Sense HAT.

L'écriture de programmes pour Sense HAT est très simple, avec une bibliothèque Python à disposition pour une prise en main rapide et facile.

Le site Web d'AstroPi, <http://astro-pi.org/>, propose de nombreuses idées et instructions d'utilisation de Raspberry Pi et Sense HAT sur la Station spatiale internationale.

Les capteurs I2C :

La carte est équipée d'un certain nombre de capteurs auxquels le RPi accède via le bus I2C.

- Capteur de mesure inertielle: ST [LSM9DS1](#). Ce circuit comporte un accéléromètre 3D, un gyroscope 3D et un magnétomètre 3D combinés dans une seule puce.

Le LSM9DS1 peut mesurer des accélérations de $\pm 2g/\pm 4g/\pm 8g/\pm 16g$, un champ magnétique de $\pm 4/\pm 8/\pm 12/\pm 16$ gauss ainsi qu'une vitesse angulaire de $\pm 245/\pm 500/\pm 2000$ degrés/seconde. Il est donc capable de fournir des informations de tangage, roulis et lacet sur les mouvements qu'il subit.

- Capteur de pression et température: Capteur de pression barométrique et capteur de température: ST [LPS25H](#).

Ce circuit peut mesurer une pression absolue comprise entre 260 et 1260 hPa avec une précision d'1Pa. Il intègre une compensation de la température et un convertisseur analogique-numérique 24 bits. Il peut donner la pression atmosphérique en pascals ou en millibars, ainsi que la température en °C.

- Capteur d'humidité et de température : Capteur d'humidité relative et capteur de température: ST [HTS221](#). Ce capteur fournit le pourcentage d'humidité relative ainsi que la température en degrés centigrades. Le capteur est assez sensible pour détecter la vapeur d'eau contenue dans l'haleine humaine de sorte qu'il sera possible de détecter la présence de l'équipage lorsqu'il travaillera près de l'Astro Pi. Il peut mesurer l'humidité relative de 0 à 100% avec une précision de 4,5% entre 20% et 80% d'humidité relative. Le capteur est calibré à la fabrication et les corrections sont stockées dans des

registres internes.

Il mesure la température avec une précision de $\pm 0,5^{\circ}\text{C}$ entre 15 et 40°C. Néanmoins, les mesures de température sur cette carte sont à considérer avec méfiance dans la mesure où l'échauffement des composants (en particulier lorsque les LED fonctionnent) provoque une élévation de la température ambiante à proximité de la carte...

Le joystick :

Le joystick est un modèle CMS (composant monté en surface). Il dispose de 5 contacts : Haut, bas, droite, gauche et clic vertical. C'est un modèle Alps SKRHABE010. Il pourra émuler les touches de direction du clavier et la touche Entrée avec le clic central. Il est accessible via le système standard de gestion des événements Linux `/dev/input/event*`

Atmel Tiny88 :

La gestion de la matrice de LED et du joystick ont été délégués à un micro-contrôleur ATMEL Tiny88.

La matrice de LED est pilotée par une combinaison formée d'un pilote de LED à courant constant (un LED2472G) et d'un ATTiny88 Atmel exécutant un firmware « maison » qui gère l'affichage 8x8 en RVB avec une résolution de 15 bits.

L'ATTiny88 est également chargé de gérer le joystick. Il n'y avait pas assez de connexions disponibles sur le micro-contrôleur Atmel pour en dédier cinq à la gestion du joystick. Du coup le joystick a été intégré à la matrice de LED qui est balayée à environ 80 Hz. C'est donc à cette fréquence que les informations du joystick sont lues.

Tous les capteurs (et le firmware de base pour l'Atmel) sont accessibles à partir du Raspberry Pi via le bus I2C.

2 Les Bibliothèques de programmation :

(Source : <https://www.raspberrypi.org/documentation/hardware/sense-hat/>)

Comme de nombreuses cartes d'extension sur Raspberry PI, la bibliothèque de programmation Python vous est fournie, ainsi que de nombreux exemples et même un simulateur pour éviter de monter la carte.

La bibliothèque C++ RTIMULib est également fournie avec quelques exemples d'utilisation (cf source ci-dessus).

Mais pour le TP du jour, où il s'agit de découvrir le bus I2C et non la Sense Hat vous allez devoir utiliser la bibliothèque du langage C `libi2c-dev`.

3 Installation et branchement:

Connectez-vous à votre Raspberry PI en utilisant **votre login** via SSH -X sous Linux. (NE PAS UTILISER le login pi)

Faites la mise à jour des logiciels (vous n'êtes pas obligé de mettre à jour la distribution raspian).

```
$ : sudo apt-get update
$ : sudo apt-get upgrade
```

Vous devez installer les packages suivants s'ils ne sont pas déjà installés:

- `libi2c-dev`
- `i2c-tools`

```
$ : sudo apt-get install i2c-tools libi2c-dev
```

Avec la commande **raspi-config** activez le bus I2C.

Arrêtez le Raspberry PI (`sudo poweroff`), puis branchez la SenseHat en utilisant la `pi wedge`. Le schéma « Raspberry Pi Pinout » de la page suivante peut vous aider. Puis redémarrez le Raspberry PI.

Testez le bus I2C avec les commandes suivantes:

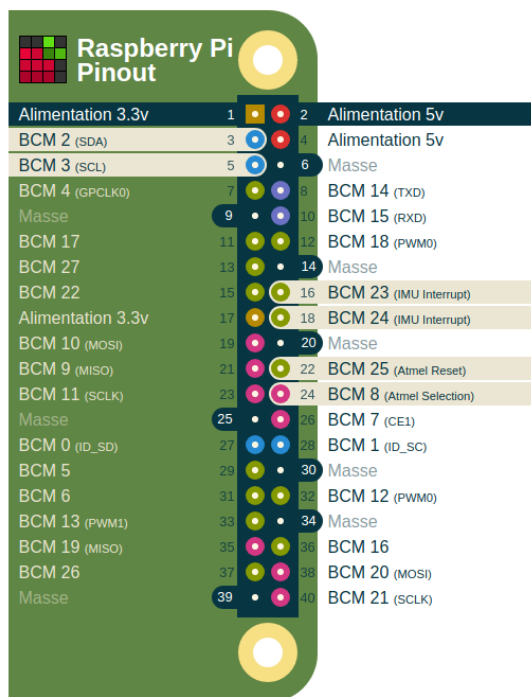
```
$ : ls /dev | grep i2c doit donner : i2c-1
puis
```

```
$ : i2cdetect -y 1 doit être conforme à la documentation de la page suivante:
```

(source : https://fr.pinout.xyz/pinout/sense_hat)

Votre enseignant a testé la SenseHat en utilisant le branchement suivant :

Pin **1** (3V3), Pin **2** (5V), Pin **3** (SDA), Pin **5** (SCL), Pin **9** (GND), Pin **16** (BCM 23), Pin **18** (BCM 24), Pin **27** (ID_SD), Pin **28** (ID_SC).



Accueil » Cartes » Raspberry Pi

Sense HAT

Sense HAT est une carte d'extension pour Raspberry Pi composé d'une matrice LED 8x8 en RGB (rouge, vert, bleu), d'un joystick 5 boutons (directions+appui), et ainsi qu'un capteur IMU, de température, d'humidité et de pression.

Le registre à décalage (shift register) utilisé pour la matrice de LED est un LED2472G connecté par un microcontrôleur Atmel ATTINY88 accessible en i2c à l'adresse 0x46 (70) du Pi. Le switch/Joystick multidirectionnel SKRHABE010 est aussi piloté par le ATTINY88.

Les capteurs eux-mêmes sont pilotés par le bus i2c:

- le capteur IMU via un LSM9DS1 trouvable à l'adresse i2c 0x1c-0x1e (28-30) et 0x6a-0x6b (106-107), avec interruption par le ATTINY88,
- Le capteurs de pression/température est un LPS25H disponible à l'adresse i2c 0x5c (92),
- le capteurs d'humidité/température HTS221 est lui accessible à l'adresse i2c 0x5f (104).

Note: le microcontrôleur Atmel peut être reprogrammé en utilisant le bus SPI. Seuls les broches de reset et sélection sont représentées ici et ajouter des périphériques SPI devrait être possible, moyennant quelques précautions.

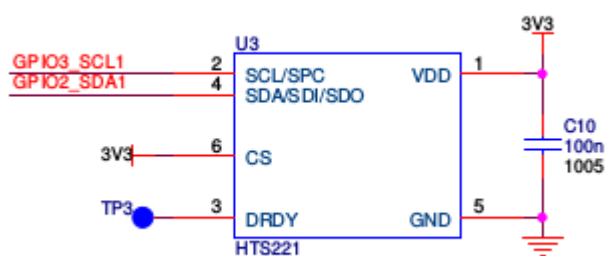
Détails

- fabriqué par [Raspberry Pi](#)
- carte au format HAT
- configuration automatique par EEPROM
- alimentation 5v et 3v3
- 6 broches GPIO actives
- communication par bus I2C
- 0x5c: LPS25H
- 0x1c: LSM9DS1
- 0x5f: HTS221
- 0x46: LED2472G
- 0x6a: LSM9DS1
- [plus d'info](#)
- [lien GitHub](#)
- [schéma](#)
- [achat](#)

1. Donnez le rôle de SDA et SCL ?
2. Pourquoi connecter ID_SD et ID_SC ?

4 Analyse de la documentation :

En vous aidant de la documentation « HTS221_Sensor_Datasheet.pdf » (cf FTP) répondre aux questions suivantes :



1. Pourquoi la pin 6 est connectée au 3,3V du GPIO (cf schéma électrique de la SenseHat ou extrait ci-dessus et chapitre 1.1 et 5.1) ?
2. Au chapitre 5.1 est décrite la terminologie utilisée pour le bus I2C, qui est vous l'aurez bien compris une communication de type Maître/Esclave. Spécifiez qui est le maître et qui est l'esclave ? Expliquez le rôle du Raspberry Pi pour la communication I2C en utilisant les termes du chapitre 5.1.
3. Que doit contenir le registre WHO_AM_I (chapitre 7) et quelle est son adresse ? Donnez le résultat en hexadécimal. **Rappel :** Un registre est une zone mémoire de l'HTS221.

5 Analyse du code :

Récupérez le fichier codeSenseHat.zip sur le serveur FTP (en utilisant votre login, et non pi):

```
$ : wget -P $HOME/Documents/ -c ftp://snir1:snir@192.168.140.84/codeSenseHat.zip
```

Décompressez l'archive :

```
$ : unzip codeSenseHat.zip
```

Compiler le fichier humidity.c et exécutez le programme généré.

En vous aidant de la documentation « HTS221_Sensor_Datasheet.pdf » et du fichier humidity.c répondre aux questions suivantes :

1. Que définissent les différentes constantes ? (ne détaillez pas chacune d'entre elles).
2. Pourquoi DEV_ID vaut 5F₍₁₆₎ ?
3. Pourquoi DEV_PATH vaut "/dev/i2c-1" ?
4. Que veut dire DEV ?
5. Pourquoi H0_T0_OUT_H vaut 37₍₁₆₎ ? Justifiez en précisant où vous avez trouvé cette information.
6. A votre avis, que fait la fonction i2c_smbus_read_byte_data ()?
7. A votre avis, que fait la procédure i2c_smbus_write_byte_data ()?
8. Pourquoi le Registre CTRL_REG1 prend la valeur 0x84 à la ligne 81 ?
9. Quels sont les registres contenant les valeurs des températures et humidités d'étalonnage ? Quel(s) chapitre(s) explique(nt) leur utilisation ?

Attention : Quelques opérateurs en C/C++ :

| Opérateur | Dénomination | Effet | Syntaxe | Résultat |
|-----------|--|--|----------------------|-----------|
| & | ET bit-à-bit | Retourne 1 si les deux bits de même poids sont à 1 | 9 & 12 (1001 & 1100) | 8 (1000) |
| | OU bit-à-bit | Retourne 1 si l'un ou l'autre des deux bits de même poids est à 1 (ou les deux) | 9 12 (1001 1100) | 13 (1101) |
| ^ | OU bit-à-bit exclusif | Retourne 1 si l'un des deux bits de même poids est à 1 (mais pas les deux) | 9 ^ 12 (1001 ^ 1100) | 5 (0101) |
| << | Décalage à gauche | Décale les bits vers la gauche (multiplie par 2 à chaque décalage). Les zéros qui sortent à gauche sont perdus, tandis que des zéros sont insérés à droite | 6 << 1 (110 << 1) | 12 (1100) |
| >> | Décalage à droite avec conservation du signe | Décale les bits vers la droite (divise par 2 à chaque décalage). Les zéros qui sortent à droite sont perdus, tandis que le bit non nul de poids plus fort est recopié à gauche | 6 >> 1 (0110 >> 1) | 3 (0011) |

6 Codage :

En utilisant la documentation « LPS25H_Sensor_Datasheet.pdf » (cf FTP) complétez le fichier pressure.c

Le code à compléter est signalé par le commentaire /* A COMPLÉTER */.

7 Pour Aller plus loin:

Compilez et exécutez le programme led_matrix.c... Regardez un petit peu le code qui gère la matrice de LED.

Créez un fichier source MaSenseHat.c contenant au minimum une procédure ayant en sortie la température ainsi que la pression atmosphérique et une fonction retournant l'humidité.

Créez un fichier entête MaSenseHat.h contenant les prototypes de vos fonctions ainsi que la déclaration des différentes constantes nécessaire à votre programme.

Réalisez un programme de Test.

Téléchargez sur le serveur FTP le fichier libsense-master.zip.

Décompressez-le et Installez les fichier en lisant le Readme. Attention vous **ne devez pas** réinstaller l'I2C.

Regardez et Compilez le fichier exemple sensetext.c

Appropriiez vous le fichier pour faire défiler sur les leds, la température, la pression atmosphérique et l'humidité.

8 Compte-rendu :

Vous devez mettre sur le serveur FTP un fichier archive au format **VosNoms.zip**

Il contient :

- Les réponses aux questions au format libre-office VosNom.odt
- pressure.c complété
- Le(s) répertoire(s) complet(s) contenant les réponses du chapitre 7 « Pour aller plus loin ».

Vous pouvez récupérer des fichiers et répertoires sur un Raspberry pi en utilisant FileZila depuis un poste Linux. En revanche, vos Raspberry pi n'étant pas des serveurs FTP (pour le moment), vous devez utiliser le protocole SFTP (FTP via SSH) en utilisant vos logins et mot de passe sur le port 22.

