

# Le matériel

**Le Rucher Connecté** Etude Décembre 2020 - Thierry Picquart -

Une ruche peuplée est la base de cette étude.

## L'intelligence

un microcontrôleur est le cœur du système. J'ai choisi l'**ESP32** de **Expressif**.



## Les capteurs

### Prise de poids

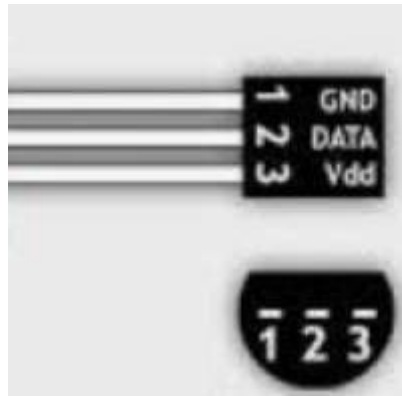
La prise de poids est souvent déconseillée mais dans ce cas c'est important. Une variation de poids en plus ou en moins doit être constante dans un sens ou dans l'autre. Selon la saison, le temps, la température extérieure, le nombre d'abeille et la quantité de réserve fait évoluer le poids. 4 **pesons** et un convertisseur analogique digital **HX711** (branchement des pesons en pont).



le **HX711** sera connecté sur les broches 2 et 5 de l'**ESP32**

### Prise de température

La prise de température n'est souvent pas un bon signe mais dans mon cas c'est important. Cela va permettre de vérifier en haut et en bas de la ruche s'il y a quelque chose d'anormal. Les abeilles régulent la température autour de la reine à **35°C**. Une sonde de température **18DB20**



un “trois pattes” digital en connexion bus **I2C** connecté sur le GPIO 4 de l'**ESP32**. Il sera placé en partie haute de la ruche

## Prise de l'humidité

Les abeilles souffrent beaucoup de l'humidité. lorsqu'elle est trop élevée, elle dépensent beaucoup d'énergie pour assainir la ruche.



un **DHT11** fournira en **I2C** la température et l'humidité. il sera placé en partie **basse** de la ruche.

## Les données

Chaque ruche va devoir récolter les données des capteurs et les envoyer. Ces données sont:

1. le Poids
2. La température en haut de la ruche
3. La température en bas de la ruche
4. L'humidité de la ruche

Comme il y a plusieurs ruches dans le rucher, chaque ruche doit être identifiée par un nom (c'est plus sympa) et par un ID. Les données sont envoyées à un ESP dit Maître qui récupèrera toutes les données de toutes les ruches du ruchers pour les envoyer à la maison.

En plus de ces données, il y a des paramètres “système”.

La balance doit être étalonnée et tarée ce qui rajoute 2 paramètres “système” d'étalonnage

## L'envoi des données

J'ai choisi le protocole **ESP-NOW** pour transmettre ces données.



Ce protocole en mode “plusieurs esclaves vers un maître” nécessite l'adresse **MAC** du maître. ce qui rajoute une données système **Données à transmettre**:

1. Le nom de la Ruche
2. L'ID de la Ruche
3. le Poids
4. La température en haut de la ruche
5. La température en bas de la ruche
6. L'humidité de la ruche

Paramètres “système”:

1. Adresse MAC de l'ESP Maître
2. Etalonnage, Tare
3. Sleep

## Gestion de l'énergie

Dans le Rucher, il n'y a pas de 220V. Mes premiers tests sont prévus avec une pile 18650 pour alimenté l'ESP de la ruche avec ses capteurs. Il est donc important de veiller à la consommation de l'ensemble. Le programme doit donc être optimisé. Je vais utiliser les capacités d'hibernation de l'ESP32.

| Power mode          | Description   |                                   | Power consumption                     |
|---------------------|---|-----------------------------------|---------------------------------------|
| Active (RF working) | Wi-Fi Tx packet                                       |                                   | Please refer to Table 15 for details. |
|                     | Wi-Fi/BT Tx packet                                    |                                   |                                       |
|                     | Wi-Fi/BT Rx and listening                             |                                   |                                       |
| Modem-sleep         | The CPU is powered on.                                | 240 MHz *                         | Dual-core chip(s) 30 mA ~ 68 mA       |
|                     |   |                                   | Single-core chip(s) N/A               |
|                     |   | 160 MHz *                         | Dual-core chip(s) 27 mA ~ 44 mA       |
|                     |   |                                   | Single-core chip(s) 27 mA ~ 34 mA     |
|                     |   | Normal speed: 80 MHz              | Dual-core chip(s) 20 mA ~ 31 mA       |
|                     |   | Single-core chip(s) 20 mA ~ 25 mA |                                       |
| Light-sleep         | -   |                                   | 0.8 mA                                |
| Deep-sleep          | The ULP co-processor is powered on.                   |                                   | 150 $\mu$ A                           |
|                     | ULP sensor-monitored pattern                          |                                   | 100 $\mu$ A @1% duty                  |
|                     | RTC timer + RTC memory                                |                                   | 10 $\mu$ A                            |
| Hibernation         | RTC timer only  |                                   | 5 $\mu$ A                             |
| Power off           | CHIP_PU is set to low level, the chip is powered off. |                                   | 0.1 $\mu$ A                           |

Le mode **sleep** ne consomme que 5 micro-ampères Les données peuvent être mesurées et envoyées que par intervalles de temps, par exemple toutes les 15 minutes en été et toutes les 30 minutes en hiver.

## Paramétrage

Les données de paramétrage sont propres à chaque ruche:

1. Le nom
2. L'ID
3. L'adresse MAC du maître
4. le temps d'hibernation

## 5. les paramètres de la balance

Il y a 2 solutions. Soit on charge le code "en dur" dans chaque ruche. Soit on charge le même code en permettant de faire le paramétrage sur place.

Il y a environ **60 000 abeilles** dans une ruche. Le branchement d'un PC au début du printemps ou elles sont le plus agressives ne me convient pas beaucoup, ça risque de piquer. J'ai choisi de développer une application accessible en **Bluetooth** depuis un smartphone (sauf iPhone) et d'utiliser la mémoire **flash** (EEPROM) de l'**ESP32** qui ne s'efface pas quand il est en veille, resette ou est hors tension. Afin d'économiser l'énergie, cette application sera déclenchée par un bouton connecté à un **GPIO** de l'**ESP** afin de le réveiller pour déclencher le paramétrage.

## Les programmes

### Lecture de la température 18DB20

```
#include <OneWire.h>
#include <DallasTemperature.h>

const int oneWireBus = 4;    //GPIO 4

OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

void setup(){
  sensors.begin();
}

void loop () {
  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0);
}
```

From:

<https://www.fablab37110.chanterie37.fr/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://www.fablab37110.chanterie37.fr/doku.php?id=start:projets:thierryIm&rev=1607723187>

Last update: **2023/01/27 16:08**

