

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE**  
**UNIVERSITE MOHAMED BOUDIAF - M'SILA**

**FACULTE DE TECHNOLOGIE**  
**DEPARTEMENT D'ELECTRONIQUE**  
**N° : 2018/ESEM09/87**



**DOMAINE : SCIENCE ET TECHNOLOGIE**  
**FILIERE : ELECTRONIQUE**  
**OPTION : ELECTRONIQUES DES SYSTEMES**  
**EMBARQUES**

**Mémoire présenté pour l'obtention**  
**Du diplôme de Master Académique**

**Par : DEGHICHE Nouredine**  
**BOUHARRAG Ismail**

**Intitulé**

**Réalisation d'un rucher connecté**

**Soutenu devant le jury composé de :**

M. Benhacene	Université de M'sila	Président
A. Benhamadouche	Université de M'sila	Rapporteur
A. Bellouti	Université de M'sila	Examineur

**Année universitaire : 2017 /2018**

## *Dédicace*

A nos parents ;

A nos Frères et sœurs ;

A tous nos Camarades ;

A nos Amis et connaissances.

BOUHARRAG Ismail

DEGHICHE Noureddine

# *Remerciement*

Nous remercions en premier lieu Dieu qui nous a donnés la force et la patience pour terminer ce travail.

*Nous tenons à exprimer nos remerciements à notre encadreur Monsieur **BENHAMADOUCHE.A** qui a proposé et dirigé ce travail.*

*Nous remercions monsieur le président de jury, ainsi que les membres de jury d'avoir accepté de juger ce travail.*

*Nous tenons à remercier les responsables et tout le personnel du département d'Electronique de M'sila pour les facilités qu'ils nous ont accordés pour terminer ce travail.*

*Enfin, nous remercions toutes les personnes, qui ont contribué de près ou de loin, directement ou indirectement à l'aboutissement de ce travail.*

## Résumé

Dans ce travail, nous avons réalisé un rucher connecté pour la surveillance des ruches d'abeilles, grâce un système électronique composé de deux parties essentielles, la première montée sur chaque ruche pour mesurer la température et l'humidité interne ainsi que le poids de la ruche, ces mesures sont envoyées vers la deuxième partie du système à travers un module nRF24L01, cette deuxième partie intègre d'autres capteurs pour la mesure des données météorologiques. Après que les mesures sont collectées, un paquet d'information est envoyé vers un serveur distant en utilisant le réseau GSM. Nous avons choisi d'utiliser le serveur ThingSpeak qui permet l'affichage en temps réel des mesures effectuées. Aussi, nous avons intégré une autre fonctionnalité qui nous permet de consulter l'état du rucher par SMS.

**Mots clés :** ruche connecté, Internet des objets, objets connectés, Arduino, STM32

## ملخص

في هذا العمل سنقوم بإنشاء منحل متصل بالإنترنت من أجل تتبع خلايا النحل وذلك لتسهيل عمل النحال. ويتكون أساساً من جزئين. الجزء الأول وهو النظام الإلكتروني الخاص بكل صندوق حيث يوضع عليه حساسات لقياس درجة الحرارة ونسبة الرطوبة وكذلك الوزن ليُرسل هذه القياسات إلى الجزء الثاني من النظام وهو المحطة. تستقبل المحطة هذه القياسات وترسلها إلى موقع الكتروني يسمح بعرضها على شكل منحنيات بيانية. تسمح المحطة أيضاً بإرسال القياسات عن طريق رسالة قصيرة،

**كلمات مفتاحية:** منحل متصل، إنترنت الأشياء، الأردو ينو، STM32

## Summary

In this work, we achieve a connected apiary for beehives monitoring, using an electronic system composed of two essential parts, the first one mounted on each hive to measure the temperature and the internal humidity as well as the weight of the beehive. These measurements are sent to the second part of the system through an RF module nRF24L01, this second part integrates other sensors for the measurement of meteorological data. After the measurements are collected, a data packet is sent to a remote server using the GSM network. We chose to use the ThingSpeak server which allows a real-time treatment and display of the measured data. Also, we integrated another feature that allows us to view the status of the apiary by SMS.

**Key words:** connected hive, Internet of Things, Connected things, Arduino, STM32

## NOMENCLATURE

2G	2nd generation
3G	3rd generation
4G	4emme generation
ABS	Anti-Lock Braking Systems
ASCII	American Standard Code for Information Interchange
AT	ATtention
CE	Chip Enable
CISCO	Commercial & Industrial Security Corporation
CPU	Central processing unite
CSN	Chip Select Non
CSS3	Cascading Style Sheets
DC	Direct current
DHT22	Digital Humidity-temperature sensor
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESB	Enhanced Shock Burst
FM	Frequency modulation
GND	Ground
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hyper Text Transfer Protocol
IBSG	Internet Business Solutions Group
IdO	Internet des Objets
IHM	Interface Homme Machine
I/O	In/out
IoT	Internet of Things
IRQ	Interrupt Request pin
LED	Light emitting diode.
LoRa	Long Range

MISO	Master-In-Slave-Out
MOSI	Master-Out-Slave-In
NFC	Near Field Communication
NodeJS	Node JavaScript
NRF	Nordic Radio frequency
OS	Operating System
PWM	Pulse Width Modulation
SBC	Session Border Controller
SOC	system on a chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
RAM	Random Access Memory
RFID	Radio Frequency Identification
RS232	Recommended Standard 232
RTEMS	Real-Time Executive for Multiprocessor Systems
TCP	Transmission Control Protocol
TV	Télévision
USB	Universal Serial Bus
URI	upper respiratory infection
VCC	Voltage common collector
Wi-Fi	Wireless Fidelity

# Liste des figures

## Chapitre I. Systèmes embarqués et objets connectés

Figure 1 : Quelques exemples de systèmes embarqués .....	5
Figure 2 : Architecture d'un système embarqué .....	5
Figure 3 : Apparition de l'IOT selon CISCO IBSG .....	8

## Chapitre II. Etat de l'ART et spécifications système

Figure 1. Pèse ruches non connectés .....	15
Figure 2. Ruche placée sur un système de pondération de la société ALYA .....	16
Figure 3. Station de mesure connectée BeeOnline pour ruche .....	17
Figure 4. La ruche connectée Label Abeille .....	18
Figure 5. La ruche connectée CAPAZ GSM200 .....	19
Figure 6. Station open source OpenHiveScale .....	20
Figure 7. Une structure simple du système .....	23

## Chapitre III. Les composants de base du projet

Figure 1. L'idée première pour la réalisation du projet .....	26
Figure 2. Carte Arduino Nano .....	27
Figure 3. L'architecture interne de la carte NANO .....	28
Figure 4. Le capteur DHT22 .....	29
Figure 5. Une cellule de charge 50kg .....	30
Figure 6. Le convertisseur HX711 .....	31
Figure 7. Schéma de câblage des cellules de charge avec le HX711 .....	31
Figure 8. Le module nRF24L01 .....	32
Figure 9. Schéma électronique du module nRF24L01 .....	33
Figure 10. Vue du dessus d'une carte Nucleo-64 .....	35
Figure 11. Vue d'ensemble du microcontrôleur STM32L476RG .....	37
Figure 12. Le capteur BMP280 .....	38

Figure 13 : Le module GSM SIM800L.....	39
Figure 14. Configuration pour l'alimentation du système .....	42
Figure 15. Chargeur de batterie Li-Ion TP4056 .....	42
Figure 16. Module Boost USB 5V .....	43
Figure 17. Module photovoltaïque .....	43
Figure 18. Module photovoltaïque choisi .....	44

## **Chapitre IV. Réalisation du projet**

Figure 1. Schéma synoptique du système .....	48
Figure 2. Organigramme de l'algorithme qui gère le sous-système de la ruche .....	49
Figure 3. Schéma du DHT22 avec le microcontrôleur NANO.....	50
Figure 4. Montage du capteur DHT22 avec la carte Arduino Nano .....	51
Figure 5. Montage des cellules de charge avec HX711 et la carte Arduino Nano.....	51
Figure 6 : Branchement du module nRF24L01 avec le microcontrôleur NANO .....	52
Figure 7 : Montage du branchement du module nRF24L01 avec le microcontrôleur NANO .....	53
Figure 8. Organigramme du fonctionnement de la station .....	54
Figure 9. Montage du capteur BMP280 avec la carte Arduino .....	55
Figure 10. Branchement du module SIM800L avec la carte Arduino .....	57
Figure 11. Montage du module SIM800L avec la carte Arduino .....	57
Figure 12. Capture d'écran de la communication entre la SIM800I et l'Arduino .....	58
Figure 13 : Montage complet de système électronique de la ruche .....	59
Figure 14 : Montage complet de système électronique de la station.....	59
Figure 15. Graphiques montrant la température et la pression de la station.....	61
Figure 16. Graphiques montrant la température, l'humidité et le poids de la ruche 1 .....	61
Figure 17. Graphiques montrant la température, l'humidité et le poids de la ruche 2.....	62

# Liste des tableaux

## Chapitre I. Systèmes embarqués et objets connectés

Tableau 1. Composants d'une solution IOT .....	11
--	----

## Chapitre III. Les composants de base du projet

Tableau 1. Caractéristiques de la carte Arduino Nano .....	28
Tableau 2. Les pins du nRF24L01 .....	34
Tableau 3. Définitions des pins du capteur BMP280 .....	38
Tableau 5. Les principales commandes de test .....	40
Tableau 6. La puissance consommée par la ruche .....	41
Tableau 7. La puissance consommée par la station .....	41
Tableau 8. Relation entre la tension de la batterie et celle du module PV .....	44

## Sommaire

Introduction générale .....	1
<b>Chapitre I. Systèmes embarqués et objets connectés</b>	
1. Introduction .....	4
2. Système embarqué.....	4
2.1. Définition .....	4
2.2. Quelques exemples .....	4
2.3. Architecture d'un système embarqué .....	5
2.4. La conception du système embarqués .....	6
2.4.1. L'architecture matérielle (électronique).....	6
2.4.2. L'architecture logiciel (code) .....	7
3. L'internet des objets (IOT).....	7
3.1. Objet connecté .....	8
3.2. Domaines d'application de l'Internet des objets .....	9
3.2.1. Transport et logistique .....	9
3.2.2. Le domaine pharmaceutique .....	9
3.2.3. Le domaine de la santé.....	9
3.2.4. Le domaine de la mode .....	9
3.2.5. Le domaine agriculteur.....	9
3.2.6. Le domaine de la domotique : .....	9
3.2.7. Environnements intelligents .....	10
3.2.8. Le domaine de l'énergie (Smart Grid).....	10
3.3. Composants d'un système IOT.....	10
3.4. Contraintes liées à l'IOT .....	11
3.5. Le Web of Things .....	12
4. Conclusion.....	12

## Chapitre II. Etat de l'ART et spécifications système

1.	Introduction .....	14
2.	Problématique .....	14
3.	Les solutions proposées .....	15
3.1.	Pèse ruche.....	15
3.2.	Modèle VILKO .....	16
3.3.	Modèle BeeOnline.....	17
3.4.	Label Abeille.....	18
3.5.	Modèle CAPAZ GSM200 .....	19
3.6.	Station OpenHiveScale .....	20
3.7.	Arnia 21	
4.	Approche proposée .....	21
5.	Cahier de charge .....	22
6.	Vue conceptuelle du système .....	23
6.1.	Partie de la ruche.....	23
6.2.	Partie de La Station .....	24
7.	Conclusion.....	24

## Chapitre III. Les composants de base du projet

1.	Introduction .....	26
2.	Choix matériel (Architectural).....	26
3.	Coté ruche.....	27
3.1.	Le microcontrôleur .....	27
3.1.1.	Arduino Nano.....	27
3.1.2.	Schéma interne de la carte NANO .....	28
3.1.3.	Caractéristiques de la carte Arduino Nano.....	28
3.1.4.	Alimentation .....	29
3.2.	Les capteurs .....	29

3.2.1.	Capteur de température et d'humidité .....	29
3.2.2.	Capteur de poids.....	29
3.3.	Le module NRF24L01 .....	32
3.3.1.	Caractéristiques du module nRF24L01 .....	32
3.3.2.	Brochage .....	33
4.	Coté station.....	34
4.1.	La carte Nucleo L476RG .....	34
4.1.1	La carte Nucleo L476RG.....	36
4.1.3.	Outils de développement.....	37
4.2.	Le capteur BMP280.....	37
4.2.1.	Caractéristiques .....	38
4.2.2.	Définitions des pins .....	38
4.3.	Le module GSM SIM800L.....	38
4.3.1.	Caractéristiques .....	39
4.3.2.	Les AT COMMANDS (Commandes Hayes) .....	39
4.3.3.	Les principales commandes de test .....	40
5.	Alimentation .....	40
5.1.	Estimation de la consommation du système.....	41
5.2.	Alimentation avec module photovoltaïque.....	41
5.2.1.	Chargeur de batterie .....	42
5.2.2.	Convertisseur Boost .....	42
5.2.3.	Choix des batteries Li-Ion .....	43
5.2.4.	Choix du module PV .....	43
6.	Conclusion.....	45
 <b>Chapitre IV. Réalisation du projet</b> 		
1.	Introduction .....	47
2.	Structure détaillée du système .....	47

<b>3.</b>	<b>Sous-système de la ruche</b> .....	49
<b>3.1.</b>	<b>Mesurer la température et l’humidité de la ruche</b> .....	50
<b>3.2.</b>	<b>Mesurer le poids de la ruche</b> .....	51
<b>3.3.</b>	<b>Envoyer et recevoir les données de la station</b> .....	52
<b>4.</b>	<b>Sous-système de la station</b> .....	54
<b>4.1.</b>	<b>Mesurer la pression atmosphérique et de la température</b> .....	55
<b>4.2.</b>	<b>Envoyer et recevoir les données de la station</b> .....	56
<b>4.3.</b>	<b>Envoi les mesures par le module SIM800L</b> .....	56
<b>5.</b>	<b>Vérification et test</b> .....	59
<b>5.1.</b>	<b>Choix du moyen de visualisation</b> .....	59
<b>5.2.</b>	<b>Résultats</b> .....	60
<b>5.3.</b>	<b>Récupérer les données par SMS</b> .....	62
<b>6.</b>	<b>Conclusion</b> .....	62
	<b>Conclusion générale</b> .....	63
	<b>Bibliographies &amp; Webographies</b> .....	65

## Annexes

# Introduction générale

L'expression de l'internet des objets (Internet of Things - **IoT**), a été mise en place par Kevin Ashton dans une présentation en 1999 pour caractériser une architecture de l'information basée sur l'internet. Après, le terme est devenu populaire et largement utilisé, mais une définition exacte et précise n'est pas une tâche simple.

L'IoT est l'interconnexion avec l'internet de tous les objets qui nous entourent, et comprend les sciences de l'électronique et les sciences des télécommunications, et avec toutes les innovations possibles dans ce domaine, l'IoT devient encore plus puissant et sa définition devient aussi plus souple pour suivre le développement technologique. Donc, l'IoT peut être décrit comme étant le nouveau besoin de toutes les entités pour communiquer les uns avec les autres, non seulement les objets, mais aussi les organismes vivants peuvent être une partie de cette technologie **[23]**.

Le domaine agricole comme les autres domaines est impliqué de plus en plus dans la technologie de l'IoT, où plusieurs applications ont émergé pour faciliter les tâches quotidiennes des agriculteurs. Des applications pour la surveillance, le diagnostic et le contrôle ont ainsi vu le jour.

Dans notre projet de fin d'étude, nous allons concevoir une des applications phare qui est en train de se vulgariser dans le domaine de l'agriculture. Cette application est les ruches connectées. Nous réaliserons ainsi un système électronique basé sur la technologie de l'IoT pour faciliter le travail des apiculteurs dans la surveillance de leurs ruches.

L'objectif principal de notre projet est de réaliser un prototype fonctionnel d'un rucher connecté, celui-ci permet de collecter les données utiles d'un nombre important de ruches. Ce prototype comprend les éléments suivants : un sous-système connecté à chaque ruche pour la mesure du poids notamment, et un autre système pour la gestion, la collecte des données mesurés et la récupération de ces données via un serveur internet.

## **Introduction générale**

---

Pour ce faire, ce mémoire est organisé comme suit :

Le premier chapitre contient une description générale des systèmes embarqués et particulièrement de l'internet des objets. Le deuxième chapitre présente la problématique liée au travail quotidien des apiculteurs, et les solutions existantes pour la surveillance des ruches. Le chapitre trois présente les éléments matériels qui sont nécessaires pour la réalisation de ce projet ainsi que le fonctionnement et les caractéristiques de chaque élément. Dans le dernier chapitre, nous allons fournir les détails méthodologiques pour la conception et la réalisation de notre système. Nous allons présenter les tests et les mesures réalisées expérimentalement.

Enfin, nous terminerons ce mémoire par une conclusion qui situe l'apport de notre réalisation, ainsi que les perspectives futures de notre travail.

# **Chapitre I**

## **Systemes embarqués Et Objets connectés**

## 1. Introduction

Nous consacrons ce chapitre à introduire quelques concepts sur les systèmes embarqués particulièrement les systèmes embarqués connectés, ces derniers sont de plus en plus utilisés dans différents domaines technologiques, et sont de plus en plus présent dans la vie quotidienne de chacun. De ce fait nous allons présenter les particularités de ces systèmes pour mieux comprendre leur intérêt et leur principe de fonctionnement, ainsi que les moyens utilisés pour leur mise en œuvre.

## 2. Système embarqué

### 2.1. Définition

Un système embarqué peut être défini comme un système électronique et informatique autonome, qui est dédié à une tâche bien précise [1]. Un système embarqué peut être un système à part entière ou enfoui dans un système plus important.

### 2.2. Quelques exemples

Nous utilisons dans la vie de tous les jours des dizaines de systèmes embarqués (Figure 1), qui sont présent dans notre quotidien de façon de plus en plus persistante par exemple :

- Avions, trains, voitures... ;
- Routeurs, box internet, téléphone ;
- Electro-ménager, TV ;
- Robotique industrielle ;
- Imprimantes et photocopieurs ;
- Terminaux de paiement ;
- Caméra connectée.



Figure 1 : Quelques exemples de systèmes embarqués

### 2.3. Architecture d'un système embarqué

L'architecture d'un système embarqué se définit par le Figure 2 :

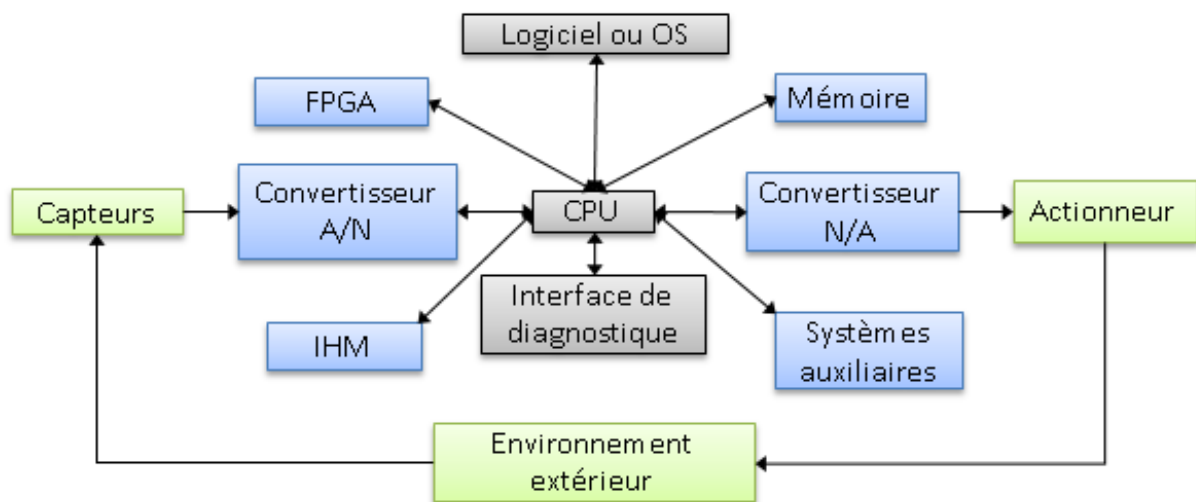


Figure 2 : Architecture d'un système embarqué

Cette architecture peut varier selon les systèmes : on peut par exemple, ne pas trouver de systèmes auxiliaires dans de nombreux systèmes embarqués autonome et indépendants. En revanche, l'architecture de base est la plupart du temps composée de [2] :

- Unité centrale de traitement (CPU) ;

- Un système d'exploitation résident ou parfois uniquement un logiciel spécifique (ex : routeur), ou une boucle d'exécution (ex : **ABS** -système antiblocage des roues) ;
- Interface IHM n'est pas souvent existante, mais est souvent utile pour reconfigurer le système ou vérifier son comportement.

Le fonctionnement du système se résume ainsi :

- Il reçoit des informations de son environnement extérieur qu'il converti en signal numérique ;
- L'unité de traitement composée du CPU, de la mémoire, du logiciel, et éventuellement des composants auxiliaires traitent l'information ;
- Le traitement génère éventuellement une sortie qui est envoyée vers les périphériques, les systèmes auxiliaires, les ports de monitoring ou l'IHM...etc.

## **2.4. La conception du système embarqués**

La conception du système se fait simultanément sur L'architecture matérielle et L'architecture logiciel.

### **2.4.1. L'architecture matérielle (électronique)**

Les supports matériels pour l'implémentation des systèmes embarqués se déclinent essentiellement en deux types [3] :

- Les microcontrôleurs ;
- Les SOC (System-On-Chip).

#### **2.4.1.1. Les microcontrôleurs**

Ces petits processeurs bon marché, peu complexes à mettre en œuvre, mais relativement limités dans les traitements qu'ils permettent ont un rôle important dans le domaine des systèmes embarqués de par leur faible encombrement et leur consommation électrique réduite.

On les rencontre également en versions plus puissantes dans les dispositifs industriels pour des tâches simples nécessitant une bonne prédictibilité du temps de

traitement des événements. Il existe des cartes à microcontrôleurs permettant une mise en œuvre très rapide pour une expérimentation ou un prototypage, comme les Launchpad de Texas Instrument ou le célèbre Arduino et ses dérivés.

#### **2.4.1.2. Les SOC (system-on-Chip)**

Dans un unique circuit intégré sont regroupées les fonctionnalités du microprocesseur, de divers périphériques (contrôleur graphique, réseau, communications série, SPI, i2C, RS232, etc.), ainsi qu'une quantité plus ou moins importante de mémoire flash, voire de mémoire RAM. L'intégration d'un SOC sur une carte personnalisée est un peu plus complexe que celle d'un microcontrôleur, mais reste largement à la portée d'un bureau d'études électroniques.

#### **2.4.2. L'architecture logiciel (code)**

Pour les systèmes très réduits, comme les microcontrôleurs, il est fréquent d'employer des OS (Operating Systems – systèmes d'exploitation) très légers. Ces OS simplifiés, qui se résument souvent à un ordonnanceur et des mécanismes de synchronisation entre tâches, sont généralement propriétaires et vendus par le fabricant du microcontrôleur accompagnés d'un environnement de développement spécifique. Il est néanmoins important de noter que, même pour des microcontrôleurs, il existe des systèmes d'exploitation libres comme Lepton, FreeRTOS ou RTEMS. [3]

### **3. L'internet des objets (IOT)**

IOT est l'acronyme de « Internet Of Things », ou internet des objets (IDO) en français. Selon l'Union internationale des télécommunications, l'Internet des objets est une « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution » [4]. Pour CISCO Internet Business Solutions Group (IBSG), l'IOT correspond simplement au moment où il y a plus d'objets connectés à internet que de personnes (figure 3).

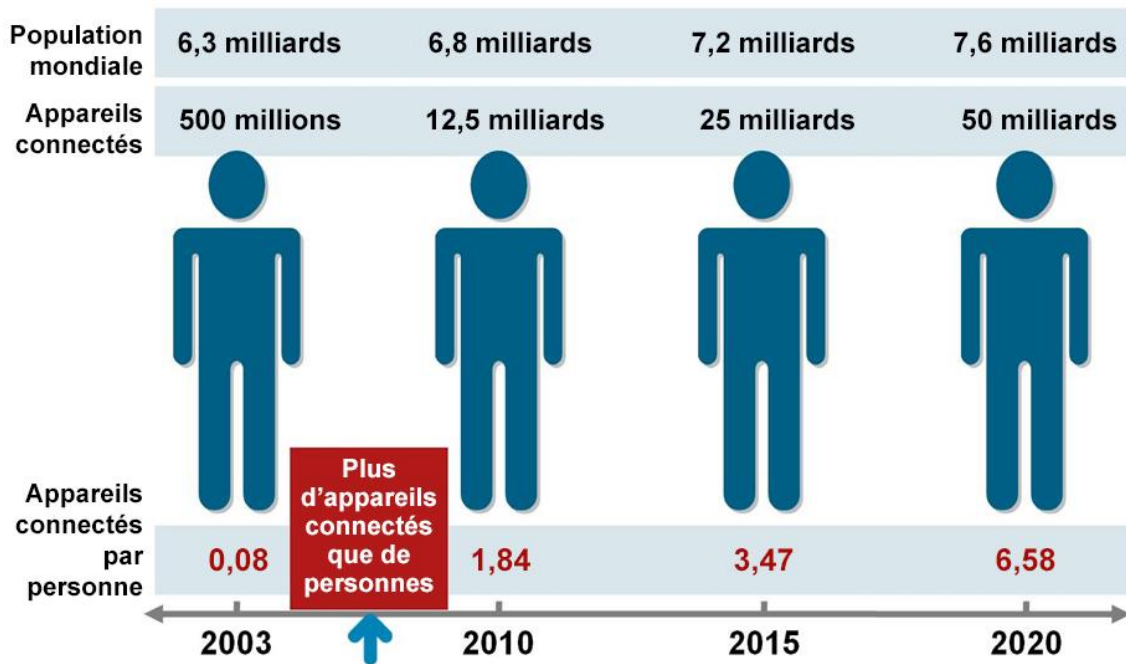


Figure 3. Apparition de l'IOT selon CISCO IBSG

En raison de l'explosion des smartphones et des tablettes le nombre d'appareils connectés à internet a atteint 12.5 milliards en 2010, alors que la population mondiale était de 6.8 milliards. C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 pour la première fois dans l'histoire [5].

### 3.1. Objet connecté

Un objet connecté est un objet physique équipé de capteurs ou d'une puce qui lui permettent de transcender son usage initial pour proposer de nouveaux services. Il s'agit d'un matériel électronique capable de communiquer avec un ordinateur, un smartphone ou une tablette via un réseau sans fil (Wi-Fi, Bluetooth, réseaux de téléphonie mobile, réseau radio à longue portée, etc.), qui le relie à Internet ou à un réseau local.

On distingue communément deux grands groupes d'objets connectés :

- Les objets destinés à la collecte et l'analyse de données, dont la mission principale est de collecter et transmettre des informations ;
- Les objets qui répondent à une logique de contrôle-commande et permettent de déclencher une action à distance.

### **3.2. Domaines d'application de l'Internet des objets**

Lorsque le nombre d'objets connectés à Internet a dépassé celui des personnes, cela a ouvert la voie à d'innombrables opportunités en matière de création d'applications dans les domaines de l'automatisation, de la détection et de la communication machine-machine. En réalité, les possibilités sont presque infinies.

#### **3.2.1. Transport et logistique**

- Utilisation des puces RFID ;
- Permet de rendre la marchandise intelligente / traçable ;
- Entrepôts entiers entièrement automatisés (Amazon) ;
- Améliorer la sécurité routière et guider les touristes.

#### **3.2.2. Le domaine pharmaceutique**

- Puces biodégradables évitant les contrefaçons ;
- Automatisation de la préparation des ordonnances.

#### **3.2.3. Le domaine de la santé**

- Mieux se connaître (Balance connectée, sport) ;
- Dépistage alerte (Tension, pouls, ...) ;
- Améliorer la qualité de vie des personnes âgées.

#### **3.2.4. Le domaine de la mode**

- Combinaison ultra connectée (Wi-Fi, GPS, Bluetooth et NFC) ;
- Charger son portable en marchant (Smart shoes).

#### **3.2.5. Le domaine agriculteur**

- Puces assurant la traçabilité du bétail ;
- Surveiller l'état de santé des animaux et de suivre leurs déplacements.

#### **3.2.6. Le domaine de la domotique**

- L'ouverture des fenêtres et des portes ;
- Contrôler le chauffage.

### 3.2.7. Environnements intelligents

- L'éclairage suivant l'horaire et la position du soleil ;
- Des incidents domestiques peuvent être évités avec des alarmes et beaucoup d'énergie pourrait être économisée.

### 3.2.8. Le domaine de l'énergie (Smart Grid)

- Optimiser la production ;
- Optimiser l'acheminement ;
- Contrôler le réseau.

## 3.3. Composants d'un système IOT [1]

L'internet des objets n'est pas une technologie à part-entière mais plutôt un système intégrant plusieurs autres systèmes. Lier un objet ou un lieu à internet est un processus plus complexe que la liaison de deux pages Web. Divers composants sont de mise, L'IOT en exige sept :

1. Une étiquette physique ou virtuelle pour identifier les objets et les lieux ;
2. Un moyen de lire les étiquettes physiques, ou de localiser les étiquettes virtuelles ;
3. Un dispositif mobile (smartphone, tablette, ordinateur portable) ;
4. Un logiciel additionnel pour le dispositif mobile ;
5. Un réseau sans fil de type 2G, 3G ou 4G ;
6. L'information sur chaque objet lié ;
7. Un affichage pour regarder l'information sur l'objet lié.

Le tableau suivant résume les principaux systèmes technologiques nécessaires à l'implantation d'une solution IOT.

Tableau 1. Composants d'une solution IOT

Type de système	Enjeux	Technologies employées
Identification	Reconnaître chaque objet de façon unique et recueillir les données stockées au niveau de l'objet.	Code-barres, URI, GPS, radio identification(RFID), ADN...etc.
Capteurs	Recueillir des informations présentes dans l'environnement pour enrichir les fonctionnalités du dispositif.	Luxmètre, capteur de proximité, thermomètre, hydromètre, accéléromètre, gyroscope...etc.
Connexion	Connecter les systèmes entre eux.	Câbles, fréquences radio, Bluetooth, Wi-Fi, ZigBee, ZWave, NFC...etc
Intégration	Intégrer les systèmes pour que les données soient transmises d'une couche à l'autre.	Middleware (simple et complexe), analyse décisionnelle des systèmes complexe
Traitement de données	Stocker et analyser les données pour lancer des actions ou pour aider à la prise de décisions.	Tableur, base de données, entrepôt de données, progiciel de gestion intégré (PGI).
Réseaux	Transférer les données dans les mondes physiques et virtuels.	Internet

### 3.4. Contraintes liées à l'IOT [1]

Les implications sociales et cognitives de ces applications et de celles à venir de l'IOT posent un certain nombre de problèmes quant à la protection de la vie privée, voire des façons de se comporter dans un environnement. L'interconnexion d'objets transmettant continuellement de l'information sur les personnes pourrait ainsi, selon certaines critiques, marquer la disparition complète du contrôle des individus sur les données les concernant.

Il y a aussi la difficulté de trouver une norme de communication qui consomme peu de courant, pour augmenter l'autonomie de la batterie, et peu de donnée pour limiter la consommation de forfait Internet GSM.

Des problèmes environnementaux peuvent aussi être soulevés, l'électronique moderne utilise une large variété de métaux lourds et de métaux rares, aussi bien que de produits chimiques synthétiques fortement toxiques, ce qui les rend difficilement recyclables.

### **3.5. Le Web of Things**

Le web of things (Web 3.0) est né après de nombreuses itérations du web, c'est un sous-jacent de l'internet Of Things. Il est basé sur des technologies web tel que le HTTP pour le transfert de données, html5 pour la structuration, CSS3 pour l'interfaçage, et JavaScript pour les interactions. Un objet physique est alors vu comme un ensemble de services accessibles au travers du web (navigateur web ou application mobile hybride) [1].

## **4. Conclusion**

Tout au long de ce chapitre nous avons traité le concept des systèmes embarqués (définition, Quelques exemples, Architecture, Les critères de performance...), ainsi que quelques notions sur l'internet des objets et des objets connectés, notamment leurs domaines d'applications et les contraintes liées à leur développement.

Ceci nous permettra de mieux situer le contexte de notre projet et l'intérêt de notre travail, ainsi dans le prochain chapitre nous allons présenter notre projet qui représente un objet connecté typique.

# **Chapitre II**

**Etat de l'ART et  
spécifications système**

## 1. Introduction

Le but de notre projet et la réalisation d'un rucher connecté, pour ce faire nous devons dans un premier temps étudier et analyser les solutions existantes sur le marché de ces systèmes. Ainsi nous pourrions optimiser notre système de façon qu'il soit performant et compétitif, ceci en tenant compte des avantages et des inconvénients des systèmes existants d'un côté, et d'un autre côté de tenir compte des besoins de l'apiculteur en lui proposant des solutions innovantes pour la surveillance de son rucher.

Ainsi, dans ce chapitre nous allons présenter un état de l'art sur les solutions existantes, et nous allons faire une brève présentation de la solution proposée.

## 2. Problématique

Les abeilles sont d'une importance vitale pour la santé de la planète: plus de 30 000 espèces d'abeilles dans le monde sont le groupe le plus important de pollinisateurs pour les plantes agricoles et sauvages. Mais les populations sont en déclin en raison d'une variété de facteurs, y compris le développement humain, les pesticides, les maladies et le changement climatique [6]. Pour cette raison, les apiculteurs devraient surveiller leurs ruches périodiquement et de manière systématique, afin de suivre l'évolution de la production du miel, et de prévenir les anomalies dans leurs ruchers.

Parallèlement à la nécessité d'une constante surveillance, un apiculteur fait face régulièrement à de nombreux problèmes lorsqu'il inspecte ses ruches manuellement comme par exemple :

- **Le déplacement** : Pour un contrôle efficace, l'apiculteur doit se déplacer à son champ fréquemment donc il perd son temps et ses efforts ;
- **Manque de précision** : Quand l'apiculteur veut obtenir des mesures météorologiques (température, humidité...etc.) dans les ruches ou à l'extérieur, beaucoup de paramètres peuvent altérer ces mesures et les rendent inefficaces.
- **L'effort physique** : Quand l'apiculteur veut obtenir le poids de ces ruches il doit les porter et les mettre sur une balance une par une, ce qui abîme ou trouble les colonies d'abeille.

- **Perte de temps** : Lorsque l'apiculteur ont un grand nombre de ruches, son temps est grandement gaspillé lorsqu'ils recueillent des mesures.
- **Difficulté à organiser et à stocker les informations** : Les mesures prises à partir des ruches sont souvent utilisées dans les études pour trouver les zones les plus appropriées pour une bonne production de miel. Donc La collecte manuelle de ces mesures rend les études difficiles et imprécises.

Afin d'éliminer tous ces problèmes, l'apiculteur utilise actuellement des technologies modernes, où il peut installer des systèmes électroniques qui lui permettent de surveiller les ruches d'abeille de manière simple et efficace.

### 3. Les solutions proposées

De nombreuses solutions existe en littérature et sur le marché, plusieurs entreprises proposent des systèmes électroniques pour aider les apiculteurs dans leur travail.

#### 3.1. Pèse ruche

Les pèses ruches peuvent être amovible ou permanent, ils sont simples d'utilisation et peuvent être adapté au format des ruches. Selon les modèles ils peuvent mesurer des ruches d'un maximum de 100 kg. Le prix varie entre 40€ et 100€.

Cependant l'apiculteur doit les consultés sur place pour prendre les mesures poids de ses ruches.



Figure 1. Pèse ruches non connectés.

### 3.2. Modèle VILKO

La société **ALYA** propose un produit avec le produit **VILKO** (Figure 2), qui contient l'électronique nécessaire pour mesurer le poids de la ruche et quelques paramètres météorologiques. Ensuite, l'appareil transmet des données par GSM [7].



Figure 2. Ruche placée sur un système de pondération de la société ALYA.

#### **Caractéristiques :**

- Les mesures de base : Poids, température, état de la batterie, accéléromètre ;
- Les mesures avancées : température et humidité externe ;
- Envoi des données par SMS ou internet ;
- Fonctionnement réglable :
  - Nombre de SMS/24heurs ;
  - Réglage des conditions d'alarme ;
  - Installé le numéro de téléphone (5 N° max).
- Alimentation sur batterie.

#### **Paramètres techniques :**

- Poids maximum : 200 Kg avec une précision de 0.1 Kg ;
- Température de fonctionnement - 10 à + 50 ° C ;
- Tension d'alimentation 6V de la batterie ;
- Capacité de la batterie 3-4 mois (à 2 SMS / jour).

### 3.3. Modèle BeeOnline [12]

La société BeeOnline propose plusieurs types de stations de mesure connecté simple, elles se différencient selon le nombre de capteur qu'y sont intégrés.

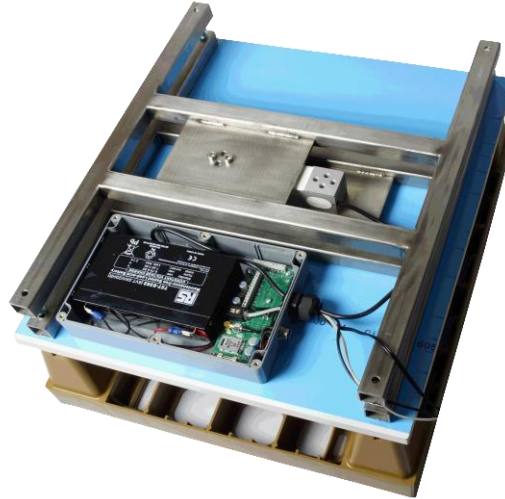


Figure 3. Station de mesure connectée BeeOnline pour ruche.

#### **Caractéristiques :**

- Simple et rapide à installer ;
- Identifier l'état de la colonie ;
- Suivi des relevés en ligne : interface intuitive et simple ;
- Accès par GSM, relevé quotidien, configuration par Internet ou Smartphone, alerte SMS en cas de vol ou chute ;
- Gestion des stations de mesures Beeonline à distance avec logiciel propriétaire ;
- Sauvegarde des données sur un serveur (10 mesures quotidiennes) ;
- 2 ans de mesures et de communication sans recharge.

#### **Paramètres techniques :**

- Mesure du poids : De 0 à 100 kg avec précision 100 gr ;
- Mesure de la température : interne/ externe de  $-20^{\circ}$   $+50^{\circ}$ C – précision 0,1°C ;
- Mesure de la pluviométrie : en mm/m<sup>2</sup> avec précision 0,2 mm ;
- Mesure de l'humidité : interne/ externe de 0 à 100% - Précision 0,1%.

**Prix :** A partir de 420 € (selon modèle)

### 3.4. Label Abeille [8]

La ruche connectée Label Abeille (Figure 4) a pour objectif de simplifier le fonctionnement d'une ruche, à faciliter l'entretien de l'essaim et à sauvegarder l'espèce. Une balance placée sous la ruche permet d'avoir accès à des informations comme la lumière, la température, la masse, l'humidité, la pression atmosphérique ou encore l'orientation. En cas de vol, elle est aussi dotée d'un système de géolocalisation.



Figure 4. La ruche connectée Label Abeille

#### **Caractéristiques :**

- Paramètre mesuré
  - Masse : jusqu'à 500 kg avec précision de 1g ;
  - Température ;
  - Orientation ;
  - Luminosité ;
  - Humidité ;
  - Pression atmosphérique.
- Batteries rechargeables

#### **Fonctionnalités :**

- Géolocalisation ;
- Système anti-vol ;
- Paramétrage des seuils d'alertes ;

- Réception des alertes par mail ou sms ;
- Activation des alertes anti-vol et de gestion d'alimentation ;
- Fréquence de relevé de données au choix : toutes les heures, toutes les 2 heures, 6 fois par jour ou 2 fois par jour ;
- Sélection du mode d'émission : GSM ou LoRa.

**Prix :** A partir de 420 € (selon modèle)

### 3.5. Modèle CAPAZ GSM200 [9]

La balance **GSM200** effectue automatiquement, par mesure électronique, des relevés de nombreuses données : le poids de la ruche, la température, la pluviométrie, l'humidité ou en plus la température de couvain. Les données recueillies sont transmises journalièrement par téléphone portable en SMS ou par E-Mail ou sur serveur CAPAZ Direct. La télémétrie est par technologie SMS, même un signal faible et suffisant pour émettre un SMS. Il existe un alarme ruche (perte de poids anormal, vandalisme, vol).

Elles composés de deux parties, la station elle-même et le portable qui peut être détaché de la structure.

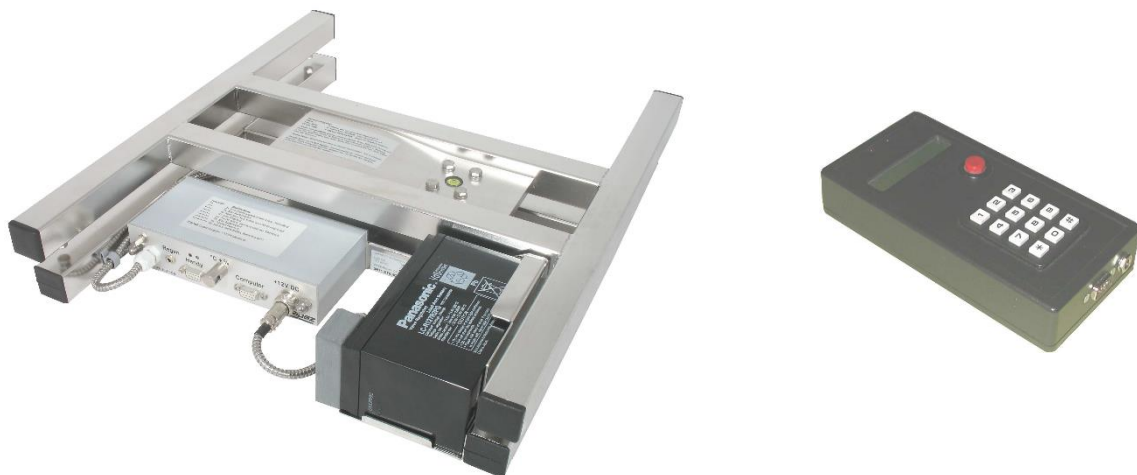


Figure 5. La ruche connectée CAPAZ GSM200.

#### **Caractéristiques :**

- Les mesures de base : Poids, température, humidité ;
- Pluviomètre (en option) ;
- Transfer via GSM par SMS ou E-Mail ;
- 7 ou 9 mesures par enregistrement ;

- Logiciel en équipement de base, base de données évaluation en graphique ;
- Sécurité / protection d'accès/ Alarmes (perte de poids anormal).

**Paramètres techniques :**

- Poids maximum jusqu'à 200 kg avec précision de 100 g ;
- Sonde de température dans corps de ruche +/- 0.5° C précision (en option) ;
- Rallonge, distance portable et balance maxi 50 m (en option) ;
- Mémoire interne 1700 mesures ;
- Interface RS232 pour téléchargement ;
- Autonomie batterie jusqu'à 200 jours, batterie rechargeable 12V, 7.2 AH ;
- Plage de fonctionnement -10° C - +45° C.

**Prix :** A partir de 1300 € (selon option)

### 3.6. Station OpenHiveScale [10]

La station OpenHiveScale est un système électronique/mécanique, elle est open source pour le suivi à distance de vos ruches, robuste, facile à utiliser et adaptable. Basé sur le principe de la balance romaine, un moteur déplace le contre-poids et un capteur optique détecte l'équilibre de la charge pour déduire ensuite le poids de la ruche

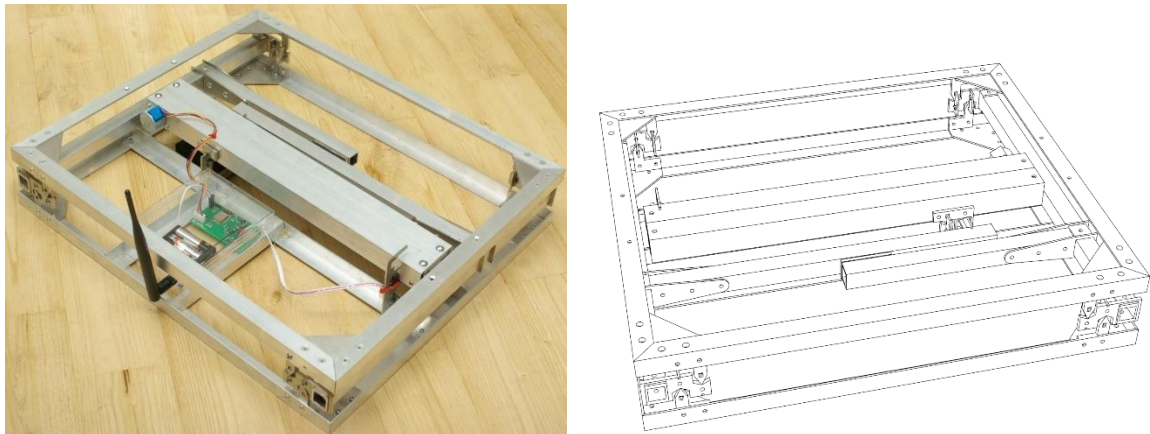


Figure 6. Station open source OpenHiveScale

Tous les plans mécaniques, schémas électroniques, typon, source logiciel sont à disposition sur **GitHub**. Les logiciels utilisés sont tous gratuits et/ou libres.

**Caractéristiques :**

- Connectivité : Wi-Fi, Sigfox, GSM ;

- Autonomie : 2 à 5 ans avec 3 piles alcalines AA (selon le nombre de mesures par jour) autonomie de 1325 mesures ;
- 125 kg en version standard avec précision de 30g ;
- Alarme sur variation de poids (+/-) ;
- Conçu pour l'apiculture professionnelle palettisée, abordable pour l'amateur, données fiables pour les chercheurs ;
- Nombre d'envoi des données paramétrable en manuel et en automatique selon la variation de poids ;
- Ultra évolutif grâce au module Wifi disponible sur toutes les versions qui permet d'ajouter les capteurs de son choix sans fil.

### 3.7. Arnia

La société **Arnia** propose un système de pondération sans fil, où sous chaque ruche, un module sans fil séparé transmet les données au module central, qui réunit les données et les envoie via le réseau GPRS 2G. Les unités sont alimentées par batterie sans panneaux solaires. Les données sont envoyées à un service cloud partagé [11].

## 4. Approche proposée

La réalisation de notre projet rentre dans le cadre de la mise en œuvre d'un système embarqué connecté, cette mise en œuvre doit réunir un certain nombre d'étape qui constitue le flot de conception d'un système embarqué. La première étape est la spécification du cahier des charges. Pour ce faire, nous devons initialement identifier la finalité de notre projet et cibler le contexte de son utilisation.

L'objectif de notre travail est de construire un système électronique qui permet de collecter les informations utiles à l'apiculteur d'un ensemble de ruche (rucher). Le système ne doit en aucun cas gêner l'apiculteur dans son travail et doit permettre à la ruche de rester mobile, aussi le système doit être autonome.

Le système prévu doit être capable de collecter plusieurs paramètres (température, humidité, poids, etc.) à partir d'un nombre donné de ruches, ensuite de stocker ces informations de manière organisée, et permettre leur consultation de manière simple et

intuitive. Le système peut également afficher ces informations localement ou de les envoyer aux apiculteurs en cas de besoin en activant un certain nombre d'alertes.

## 5. Cahier de charge

Pour que le système soit efficace et bien adapté aux besoins de nos apiculteurs, il est nécessaire qu'il obéisse à un certain nombre de critères, d'exigences et de contraintes. Ces différents points doivent être le résultat d'une réflexion profonde qui prennent en compte :

- ✓ Les besoins spécifiques de l'apiculteur ;
- ✓ Le système doit être concurrentiel ;
- ✓ La nature du lieu où le système doit être implanté ;
- ✓ Les effets nuisibles du système sur le rucher ;
- ✓ Les bénéfices escomptés du système ;
- ✓ L'efficacité du système ;
- ✓ L'estimation du prix du système.

Pour la réalisation de notre système nous avons fixé un certain nombre d'exigence et de contraintes qui peuvent être énumérées comme suit :

- ✓ Le système doit fonctionner de manière autonome ;
- ✓ Le système doit être reconfigurable ;
- ✓ Le système doit être facilement intégrables dans les ruches ;
- ✓ Transfert de données sans fil vers un stockage de données centralisé ;
- ✓ Les données mesurées doivent également être accessibles à un apiculteur sans transfert vers une base de données centrale ;
- ✓ Faible consommation d'énergie ;
- ✓ Alimentation solaire si possible ;
- ✓ Design résistant aux éléments de la nature ;
- ✓ Un coût réduit pour la réalisation du système.

Le système est censé mesurer les paramètres suivants :

- ✓ La température dans la ruche ;
- ✓ L'humidité sous la couverture de la ruche ;

- ✓ Le poids de la ruche ;
- ✓ Les données climatique (Température, Humidité, pression...);
- ✓ En plus, il doit fournir des données de diagnostic.

## 6. Vue conceptuelle du système

Après avoir fixé les besoins du cahier de charge, nous pouvons proposer une première vue conceptuelle, cette vue permet d'identifier la structure du système, tel que présentée dans la figure 7 :

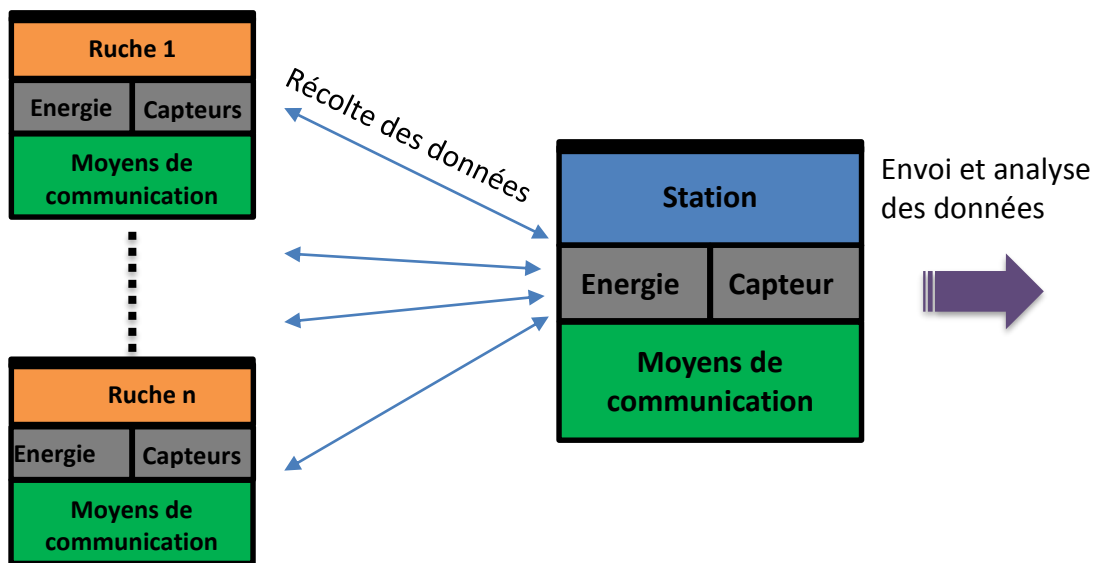


Figure 7. Une structure simple du système

Comme montré dans la figure 7, le système proposé consiste principalement en deux parties essentielles (Station et ruche), chaque partie contient des capteurs pour mesurer les paramètres physique et climatiques, ainsi que des moyens de communication pour la transmission des données entre chaque partie et entre la station et le réseau internet.

### 6.1. Partie de la ruche

Chaque ruche contient un ensemble des capteurs, une unité de traitement et une unité de communication. Tout cela pour faire les tâches suivantes :

- Mesure les paramètres de la ruche tels que : La température interne, l'humidité interne, le poids de la ruche...etc. ;
- Mesure l'état d'énergie ;

- Envoi des données collectées vers la station.

## 6.2. Partie de La Station

C'est le centre qui gère et organise toutes les opérations de mesure de communication. Il contient essentiellement une unité de traitement et deux unités de transmission pour la communication avec les ruches et pour la communication avec le réseau. Ainsi que quelques capteurs météorologiques. Les tâches qu'il exécuté sont :

- Mesure des paramètres climatiques comme : la pression atmosphérique, l'éclairement, la température, l'humidité...etc. ;
- Envoi des requêtes vers les ruches.
- Récolte les données provenant de différentes ruches ;
- Diagnostique du rucher ;
- Envoi les données sur le réseau internet pour l'affichage et l'analyse.

## 7. Conclusion

Dans un premier temps, ce chapitre a été consacré à l'étude et l'analyse des différents systèmes existants qui permettent la surveillance des ruches, nous avons pu voir quelques systèmes commercialisés ainsi que leurs caractéristiques et leurs avantages. Cette première étape était essentielle pour mettre en place un cahier des charges, qui nous a permis d'identifier les critères et exigences de notre système de ruche connectée.

Par ailleurs, nous avons proposé une première vue abstraite du système et de sa structure. Nous allons par la suite identifier les éléments matériels qui permettent la réalisation de notre projet.

# **Chapitre III**

## **Les composants de base du projet**

## 1. Introduction

Après avoir pu présenter le premier croquis de notre système dans le chapitre précédent, nous allons détailler dans ce chapitre la structure du système. Nous allons ainsi exposer les différents composants matériels qui rentreront dans sa mise en œuvre, notamment les capteurs et les composants qui permettent de lier les différentes parties du système.

## 2. Choix matériel (Architectural)

Le schéma de la figure 1, montre un aperçu des éléments clés de notre système ainsi que les fonctionnalités prévues de notre réalisation. Le système est composé de deux parties :

- Une station de gestion et de communication ;
- Un ensemble de ruches (Rucher).

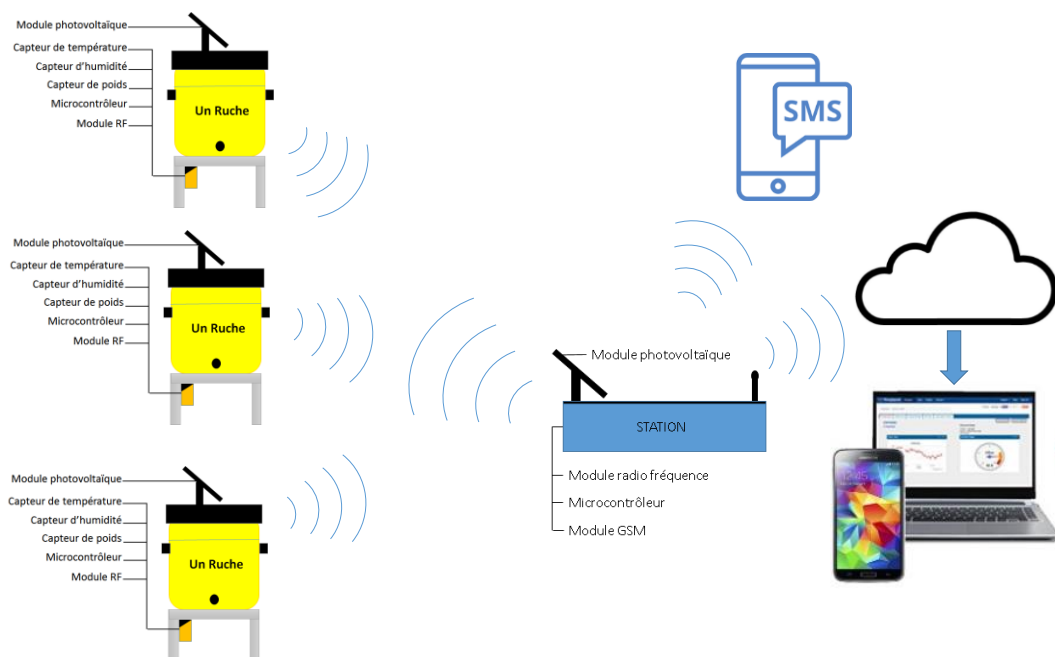


Figure 1. Schéma synoptique pour la réalisation du projet

Chaque élément comporte un ensemble de capteur, mais surtout un moyen de communication commun, nous avons choisi pour cette réalisation d'utiliser des modules RF, ils sont les mieux adaptés à notre système. Aussi la station principale doit avoir en plus un module GSM pour la communication distante, et pour l'envoi des données sur un serveur internet.

### 3. Coté ruche

Les composants nécessaires pour la réalisation de la partie ruche sont :

- Un microcontrôleur de moyenne performance ;
- Des capteurs de température et d'humidité ;
- Des jauges de contrainte pour la mesure du poids ;
- Un module RF.

#### 3.1. Le microcontrôleur

Nous avons fixé un certain nombre de fonctionnalités que doit contenir le microcontrôleur, nous les résumons comme suit :

- ✓ Le microcontrôleur doit être performant ;
- ✓ Ne doit pas consommer beaucoup d'énergie, et doit contenir des modes de mise-en-veille ;
- ✓ Doit avoir des interfaces de communication telles que : UART, SPI, I2C... ;
- ✓ Doit avoir des entrées analogiques ;
- ✓ Doit avoir une plate-forme de développement logicielle et matériel performante.

Nous avons ainsi choisi les microcontrôleur ATmega328 de Atmel, ces microcontrôleurs sont disponibles sur carte de développement Arduino Nano.

##### 3.1.1. Arduino Nano

Cette carte est l'élément principal du système côté ruche, elle permet de gérer l'ensemble des capteurs nécessaires et la communication RF.

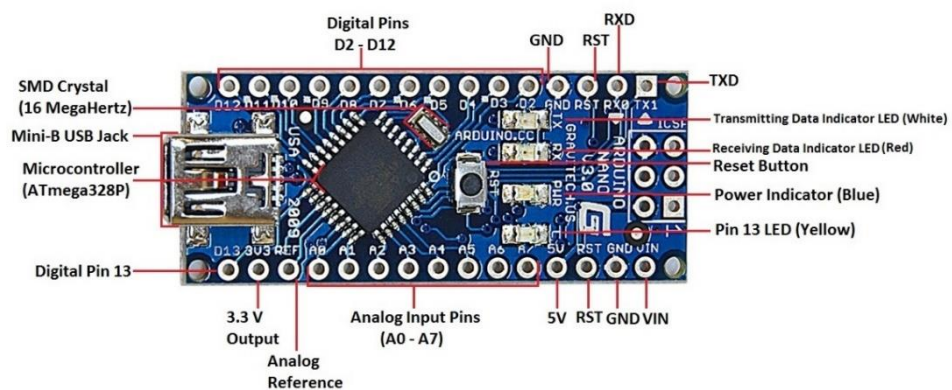


Figure 2. Carte Arduino Nano.

### 3.1.2. Schéma interne de la carte NANO

Les signaux d'entrée-sortie du microcontrôleur ATmega328 sont reliés à des connecteurs selon le schéma ci-dessous :

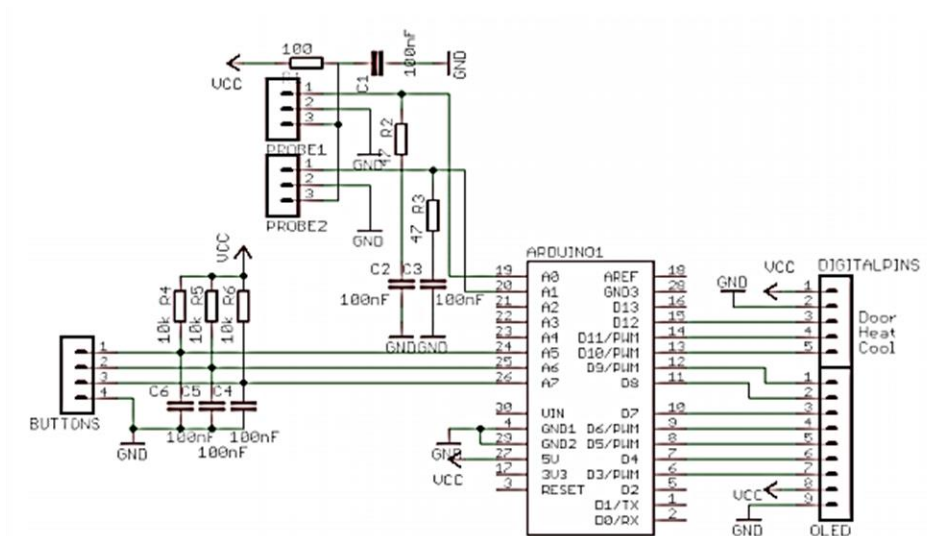


Figure 3. L'architecture interne de la carte NANO

### 3.1.3. Caractéristiques de la carte Arduino Nano

Le tableau ci-dessous présente les caractéristiques de la carte Arduino Nano.

Tableau 1. Caractéristiques de la carte Arduino Nano

<b>Microcontrôleur</b>	ATmega328
<b>Dimension</b>	1.85x4.23cm
<b>Vitesse de l'horloge</b>	16MHz
<b>Tension de fonctionnement</b>	5V
<b>Tension d'entrée (recommandé)</b>	7-12V
<b>Broches d'E/S numérique</b>	22 (dont 6 fournissent sortie PWM)
<b>Broches d'entrée analogiques</b>	8
<b>Courant continue /broche</b>	40 mA
<b>Mémoire Flash</b>	32 Ko (ATmega328)
<b>SRAM</b>	2 Kb (ATmega328)
<b>EEPROM</b>	1 Kb (ATmega328)

### 3.1.4. Alimentation

La carte Arduino Nano peut être alimentée via la connexion Mini USB, ou à travers une alimentation externe 6-20V (pin 30), ou une alimentation externe 5V (pin 27). La source d'alimentation est sélectionnée automatiquement par la carte.

## 3.2. Les capteurs

### 3.2.1. Capteur de température et d'humidité

Nous avons opté pour un capteur de type DHT22 (aussi connu sous la référence AM2302), qui réunit sur le même circuit deux capteurs, celui de la température et celui de l'humidité (Figure 4). Le fonctionnement de ce capteur dépend d'un simple protocole de communication pour le contrôler, aussi la sortie de ce capteur est numérique.

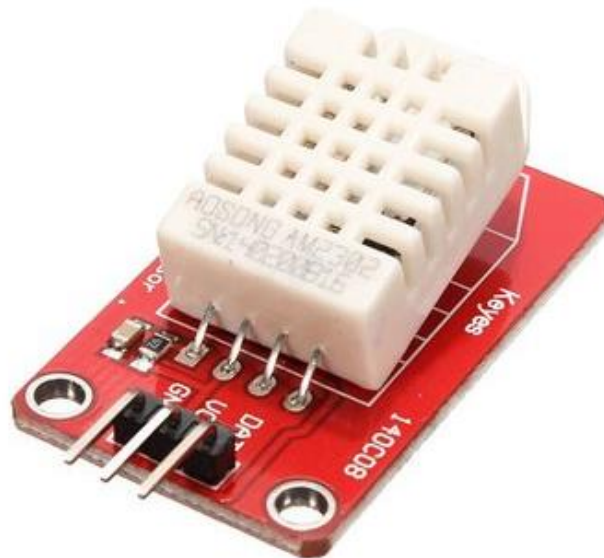


Figure 4. Le capteur DHT22

Le capteur DHT22 / AM2302 est capable de mesurer des températures de -40 à +125°C avec une précision de +/- 0.5°C et des taux d'humidité relative de 0 à 100% avec une précision de +/- 2% (+/- 5% aux extrêmes, à 10% et 90%). Une mesure peut être réalisée toutes les 500 millisecondes (soit deux fois par seconde) [13].

### 3.2.2. Capteur de poids

Le poids de la ruche est une donnée importante pour l'apiculteur car il permet de savoir si le rucher est en bonne santé et de savoir quand récolter le miel. En hiver, il permet de surveiller si les réserves de nourriture sont suffisantes [14]. Notre plus grand critère est

donc la sensibilité des capteurs que nous allons utiliser. A cela s'ajoute les conditions climatiques. En effet les matériaux utilisés sont essentiellement en métal et réagissent en fonction de la température. Les capteurs de force que nous avons utilisés sont 4 jauges de contrainte de 50kg de charge.

### 3.2.2.1. Cellule de charge

Les cellules de charge (Figure 5) sont des pièces métalliques de forme spéciale auxquelles sont fixées des jauges de contrainte. Les jauges de contrainte sont des résistances qui changent leur résistance lorsqu'elles sont courbées. Lorsque la partie métallique se plie, la résistance de la cellule de charge change [15].

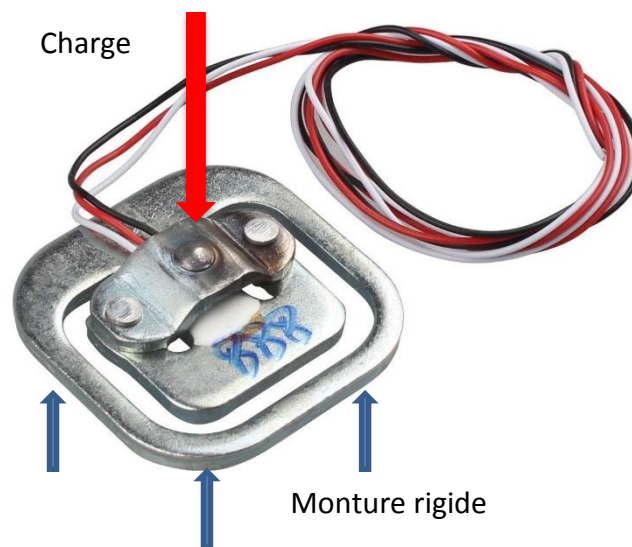


Figure 5. Une cellule de charge 50kg

Nous avons également utilisé un amplificateur HX711 de signal pour amplifier et numériser le signal de sortie des 4 jauges nécessaire.

### 3.2.2.2. Le circuit HX711

La circuit intégré HX711 est spécialement conçue pour amplifier les signaux des cellules de charge (figure 6), les numériser ensuite de les transmettre au microcontrôleur via un protocole spécifique. Les cellules de charge se branchent sur cette carte et cette carte indique à l'Arduino ce que mesurent les cellules de pesage [15].

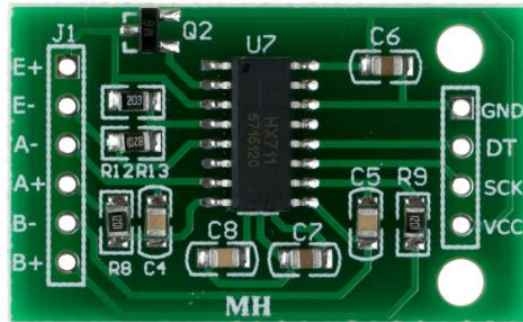


Figure 6. Le convertisseur HX711

#### Caractéristiques :

- Interface série facile d'utilisation sans programmation ;
- 2 canaux d'entrée ;
- Vitesse des données en sortie à sélectionner en 10SPS ou 80SPS ;
- Consommation de courant de moins de 1.5 mA en utilisation normale, 1uA en veille ;
- Tension de fonctionnement : 2.7V-5V ;
- Interface série facile d'utilisation sans programmation ;
- Rejet simultanée à 50 et 60Hz.

#### 3.2.2.3. Câblage des cellules de charge et HX711

La figure 7 représente le schéma de câblage pour la connexion des cellules de charge avec la module HX711.

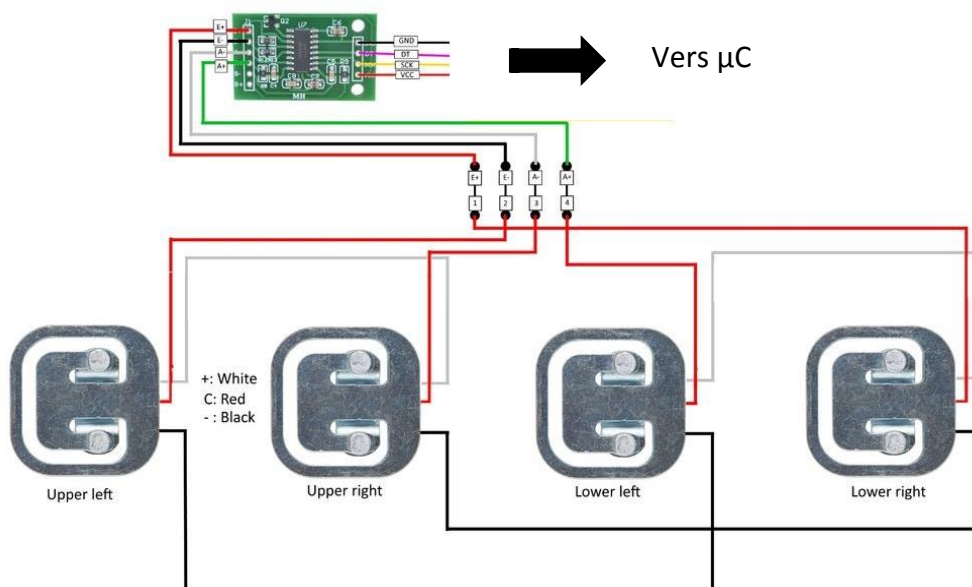


Figure 7. Schéma de câblage des cellules de charge avec le HX711

### 3.3. Le module NRF24L01

Le module radio nRF24L01 (figure 8) est un émetteur-récepteur adapté pour les applications sans fil à faible puissance, il utilise le protocole *Enhanced ShockBurst* (ESB) qui supporte la communication série et qui garantit une transmission fiable. Ce module fonctionne dans la bande des fréquences [2.400 - 2,4835] GHz. Ce module sera utilisé dans la communication sans fil entre la station de base et les ruches connectées [19].

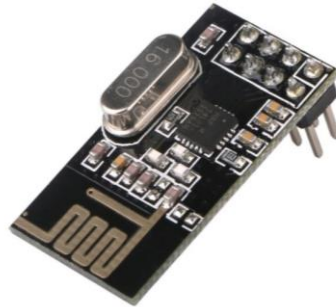


Figure 8. Le module nRF24L01

Le nRF24L01 offre une communication radio accessible via une interface SPI standard. La même puce permet de recevoir et d'émettre (transmetteur), et peut fonctionner en plus d'une centaine de canaux, et consomme très peu d'énergie. Le module peut utiliser 125 canaux différents, ce qui donne la possibilité d'avoir un réseau de 125 connexion fonctionnant indépendamment en un seul endroit. Chaque canal peut avoir jusqu'à 6 adresses, ou chaque unité peut communiquer avec jusqu'à 6 autres unités en même temps [20].

Le module dispose d'une antenne incorporée (visible en zigzag sur le circuit), sa portée maximale est de 100 m en extérieur (outdoor) et un débit pouvant aller jusqu'à 2 Mbps avec une consommation de 13.5mA.

#### 3.3.1. Caractéristiques du module nRF24L01 [19]

- Fonctionner dans la bande ISM (Industriel, scientifique, médical) 2,4 GHz ;
- Protocole de communication : Enhanced ShockBurst (ESB) ;
- Débit de transmission des données : 256 Kbps/ 1 Mbps/ 2 Mbps ;
- La portée d'émission : 70 ~100 mètres ;
- La taille : 15mm \* 29mm \* 0.8mm ;

- Les pins I/O utilisable : 8 ;
- Alimentation : 1.9V ~ 3.6V ;
- Courant nécessaire de travail : 13.5mA ;
- Mode de travail : mode RX / mode TX.

### 3.3.2. Brochage

L'architecture interne du module (figure 9) contient un microprocesseur principal NRF24L01 et les broches d'entrées/sorties (de 1 à 8 sur le schéma) pour l'utilisation avec un microcontrôleur externe.

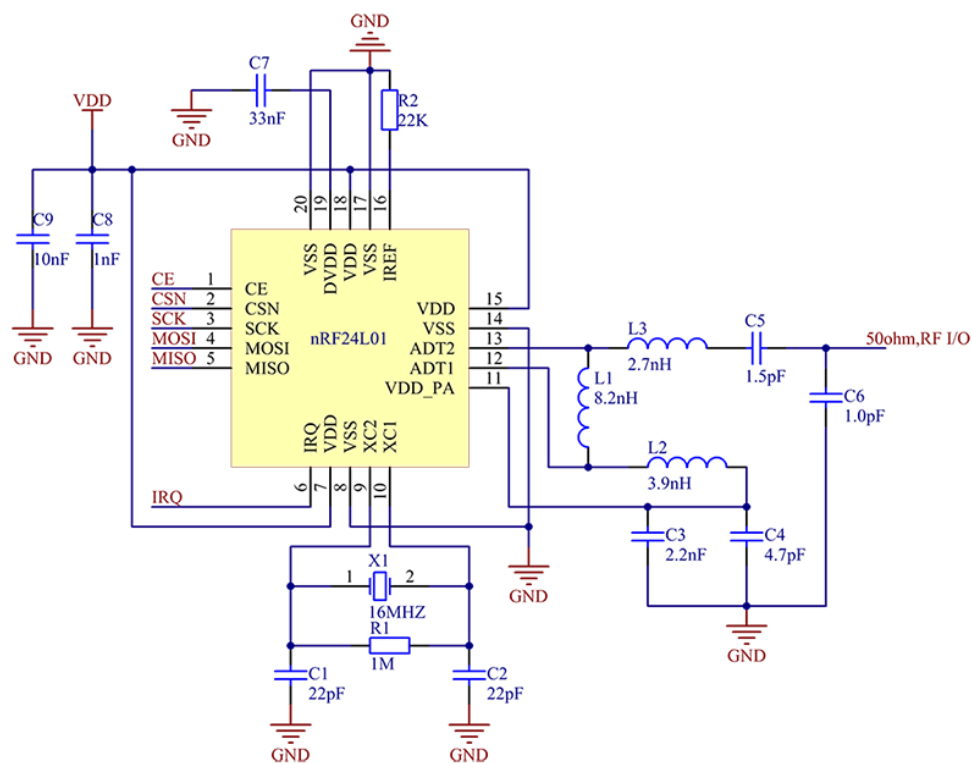


Figure 9. Schéma électronique du module nRF24L01

Le tableau suivant décrit les différents PIN qui contribues à la liaison entre le module NRF24L01 et un microcontrôleur, en se basant sur la liaison série SPI.

Tableau 2. Les pins du nRF24L01

Pin	Fonction	Description
<b>CE</b>	Entrée numérique	Chip Enable : il active le mode RX/TX, c.à.d. le module soit en mode d'envoi ou en mode d'écoute.
<b>CSN</b>	Entrée numérique	Chip Select Non : il active la réponse aux commandes SPI (Serial Peripheral Interface).
<b>SCK</b>	Entrée numérique	L'horloge de l'interface SPI (jusqu'à 10MHz).
<b>MOSI</b>	Entrée numérique	Master-Out-Slave-In : utilisé pour transférer les données du module vers le microcontrôleur.
<b>MISO</b>	Sortie numérique	Master-In-Slave-Out : utilisé pour transférer les données du microcontrôleur vers le module.
<b>IRQ</b>	Sortie numérique	Interrupt Request pin : alerte si le paquet est envoyé ou reçu.
<b>VDD</b>	Alimentation	+3V (DC)
<b>VSS</b>	Masse	Ground (0V)

#### 4. Coté station

En plus des composants tel que le DHT22, des composants pour les mesures météorologiques peuvent être intégrés dans la structure de la station. Ainsi il est important d'intégrer les éléments suivants :

- Un microcontrôleur de haute ou moyenne performance ;
- Des capteurs de température et d'humidité ;
- Capteur de pression ;
- Capteur d'éclairement ;
- Un module RF ;

##### 4.1. La carte Nucleo L476RG [21]

Tout comme les Arduino, les cartes Nucléo (Figure 10) sont des cartes de développement dotées d'un microcontrôleur et de différents connecteurs qui leur permettent d'être facilement programmables sans connaissances préalables, La grande force de ses cartes, c'est leur microcontrôleur, le STM32, qui associe rapidité et faible consommation. Autre atout, elles peuvent être combinées à du Wi-Fi, du Bluetooth ou de

l'Ethernet, ce qui leur offre un champ d'action très étendu (prototypage, domotique, Internet des objets...).

La famille des STM32, développée par la société STMicroelectronics. Ces microcontrôleurs, qui se déclinent en une gamme complète de produits 32 bits et sont construits autour du processeur ARM Cortex-M développé par la société ARM.

La carte Nucleo L476RG dispose d'un microcontrôleur STM32L476RG basé sur ARM Cortex-M4 avec une large gamme de configurations et de support de connectivité. Voici quelques points forts de la carte Nucleo L476RG :

- Microcontrôleur STM32 dans le paquet QFP64 ;
- Deux types de ressources d'extension :
  - Connectivité Arduino Uno V3 ;
  - Extension ST morpho pour un accès complet à toutes les E/S du STM32.
- Débogueur / programmeur embarqué ST-LINK / V2-1 ;
- Alimentation flexible de la carte :
  - USB VBUS ou source externe (3.3V, 5V, 7 - 12V) ;
  - Gestion de l'alimentation intégré.
- Trois LED : USB (LD1), LED utilisateur (LD2), LED d'alimentation (LD3) ;
- Deux boutons poussoirs : USER et RESET.

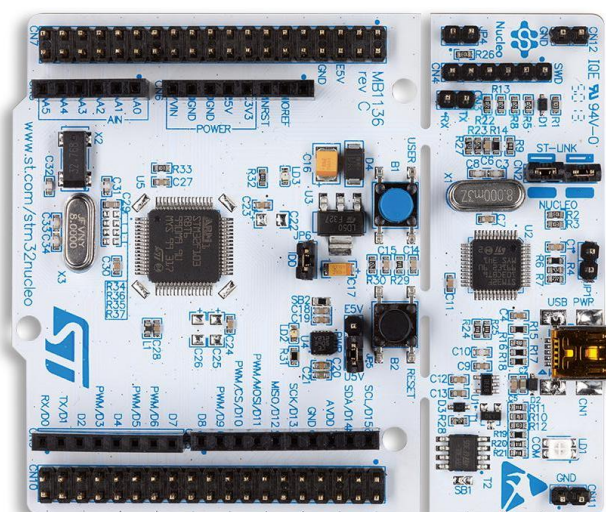


Figure 10. Vue du dessus d'une carte Nucleo-64

#### 4.1.1. Le microcontrôleur STM32L476RG

Le SoC STM32L476RG fournit les unités matérielles suivantes :

- Ultra-faible consommation (jusqu'à 130 nA en mode veille) ;
- CPU Cortex®-M4 ARM® 32 bits avec FPU, fréquence jusqu'à 80 MHz ;
- Source d'horloge :
  - Oscillateur à quartz de 4 à 48 MHz ;
  - Oscillateur RTC de 32 kHz ;
  - 3 PLLs système pour : USB, audio, ADC.
- 16x timers ;
- Plus de 114 fast entrée/sortie ;
- Mémoires :
  - Plus de 1MB de mémoire Flash ;
  - Plus de 128 KB de mémoire SRAM ;
  - Interface pour mémoire externe.
- 3x Convertisseur analogique numérique de 5 MSPS avec résolution 12 bits ou 16 bits ;
- 18 interface de communication ;
- Plus de 18 interfaces de communication dont :
  - USB OTG 2.0 ;
  - 3x I2C (1 Mbit/s) ;
  - 6x USARTs ;
  - 3x SPIs (4x SPIs with the Quad SPI);
  - Interface CAN.

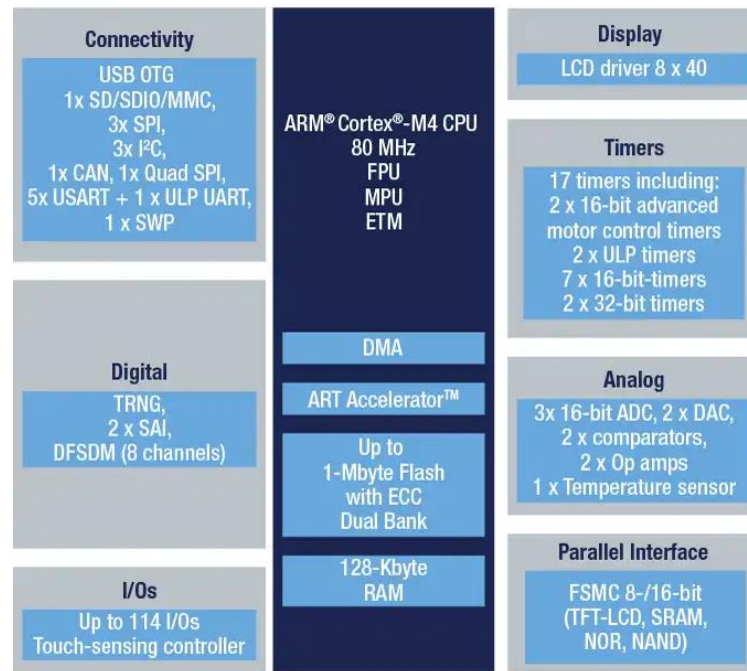


Figure 11. Vue d'ensemble du microcontrôleur STM32L476RG

#### 4.1.2. Outils de développement

La mise en œuvre des applications basées sur la carte ST Nucleo ne peut s'effectuer sans passer par les phases de configuration et de programmation. Le fabricant a mis à notre disposition le software STM32CubeMx pour la configuration. Cependant, pour la programmation des tâches à exécuter, plusieurs softwares de compilations sont développés par différents concepteurs qui sont disponible dans le marché. Dans ce travail, nous avons opté pour le compilateur Keil (MDK) V5.

#### 4.2. Le capteur BMP280

Le BMP280 est un capteur de précision permettant de mesurer la pression barométrique avec une précision absolue de  $\pm 1$  hPa et une précision de  $\pm 1.0^\circ\text{C}$  en température (Figure 12). Etant donné que la pression change avec l'altitude, et que la mesure de pression est assez bonne, ce capteur peut être aussi utilisé comme altimètre avec une précision de  $\pm 1$  m. La carte dispose également d'un régulateur 3.3V, ce senseur peut être utilisé avec systèmes 3.3v ou 5V [16].

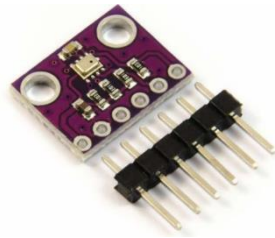


Figure 12. Le capteur BMP280

#### 4.2.1. Caractéristiques

- Alimentation maxi : 3,6V ;
- Mesure de la température de -40°C à 85°C avec une précision de +/- 1°C ;
- Mesure de la pression de 300hPa à 1100 hPa avec une précision de +/- 1hPa (de 0°C à 65°C) ou +/- 1hPa (de -40°C à 0°C) ;
- Consommation Max : 4.2  $\mu$ A ;
- Sortie numérique sur bus I2C ou bus SPI.

#### 4.2.2. Définitions des pins

Tableau 3. Définitions des pins du capteur BMP280

BROCHE	FONCTION
Vcc	Alimentation (3.6V max)
GND	Masse
SDI	Données. A relier à SDA dans le cas du bus I2C
SCK	Horloge (clock). A relier à SCL dans le cas du bus I2C
CSB	Permet de passer du mode SPI au mode I2C. Par défaut, CSB=1 donc mode I2C. Si CSB=0 au démarrage, alors le mode SPI est utilisé
SD0	Choix de l'adresse I2C. Si SD0=0, alors l'adresse est 0x76, si SD0=1, alors l'adresse est 0x77

#### 4.3. Le module GSM SIM800L

Le module GSM SIM800L (Figure 13) est l'un des plus petits modules GSM du monde avec une taille de 2.2 cm x 1.8 cm. C'est un module puissant qui démarre automatiquement et recherche automatiquement le réseau. Il inclut notamment le Bluetooth 3.0+EDR et la radio FM (récepteur uniquement). Il vous permettra d'échanger des SMS, de passer des appels mais aussi, et c'est nouveau, de récupérer de la data en GPRS 2G+.

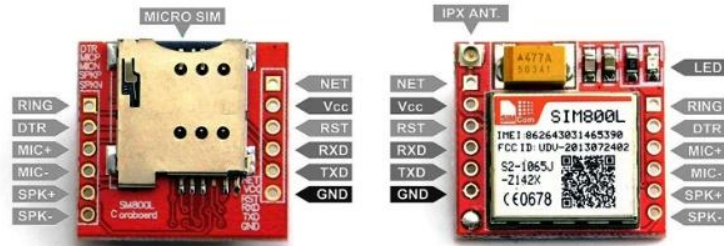


Figure 13 : Le module GSM SIM800L

#### 4.3.1. Caractéristiques [17]

- Puce : SIM800L ;
- Tension : 3.7-4.2V ;
- Courant de crête : 2A ;
- Quadri-bande 850/900/1800 / 1900MHz ;
- Le module d'alimentation démarre automatiquement, réseau de référencement ;
- Emettre et recevoir des appels vocaux à l'aide d'un casque ou un haut-parleur 8Ω externe et microphone électret ;
- Envoyer et recevoir des messages SMS ;
- Envoyer et recevoir des données GPRS (TCP / IP, http, etc.) ;
- Les feux de signalisation à bord sont complètement allumés. Il clignote lentement lorsqu'il y a un signal, il clignote rapidement lorsqu'il n'y a pas de signal ;
- Dimensions : 25 x 23mm.

#### 4.3.2. Les AT COMMANDS (Commandes Hayes)

Les commandes AT sont définies dans la norme "GSM 07.07.AT" est l'abréviation de 'ATtention'. Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes ASCII). Les commandes permettent la gestion complète du module [18].

### 4.3.3. Les principales commandes de test

Le tableau ci-dessous donne quelque exemple de commande AT qui sert pour le test du module SIM800.

**Tableau 5 : Les principales commandes de test**

AT Commande	SIM800 R13.08	Statut du modem
AT+CREG	+CREG : 0,5	Statut du réseau. ➤ 1 <sup>er</sup> indice : 0 -> GSM ➤ 2 <sup>ème</sup> indice : 0-> pas de réseau 5 -> réseau ok
AT+CPIN ?	+CPIN READY	Pas de code PIN nécessaire
AT+CSQ	+CSQ : <rsqi>, <ber>	Intensité du signal. ➤ <rsqi>: received signal strength indication. ➤ <ber> : bit error rate (%) 0 -> less than 0,2%
AT+COPS ?	+COPS : <mode>,<format>,<opérateur>	Affiche l'opérateur sélectionné

## 5. Alimentation

Concernant l'alimentation des ruches, nous avons fixés un nombre de points que l'alimentation et le système devront satisfaire tels que :

- Utilisation des modes veille pour réduire la consommation des  $\mu\text{C}$  ;
- Utilisation des interruptions lors de la programmation, ce qui réduit considérablement la consommation des  $\mu\text{C}$  ;
- Utilisation d'un module photovoltaïque éventuellement ;
- Utilisation des batteries lithium ;
- Coût réduit de l'alimentation ;
- Complexité minimum de l'alimentation.

Deux configurations sont possibles pour l'alimentation de notre système :

- Module photovoltaïque + Batterie lithium + convertisseur de puissance ;
- Batterie lithium + convertisseur de puissance.

Nous avons utilisé ces deux configurations, mais nous avons tous il faut faire une estimation de l'énergie consommée par les différents composants du système.

## 5.1. Estimation de la consommation du système

Les estimations de consommation dans cette section sont des estimations maximales lors du fonctionnement en plein charge de tous les composants, sinon, la consommation est largement inférieure, car elle dépend des cycles de fonctionnement et la mise en veille des composants.

### Coté ruche

Tableau 6. La puissance consommée par la ruche

Le composant	Le courant consommé
Arduino	50 mA
DHT22	1.5 mA
DS18B20	2 mA
NRF24L01	45 mA
Jauge de contrainte	120 mA
TOTALE	218.5 mA

### Coté station

Tableau 7. La puissance consommée par la station

Le composant	Le courant consommé
STM32	120 mA
SIM800L	1 A
BMP280	2.7 uA
NRF24L01	45 mA
TOTALE	1165.0027 mA

## 5.2. Alimentation avec module photovoltaïque

Pour ce faire, nous avons choisi la configuration de la figure ci-dessous, cette configuration permet l'utilisation et la charge d'une batterie lithium à travers l'utilisation d'un module photovoltaïque.

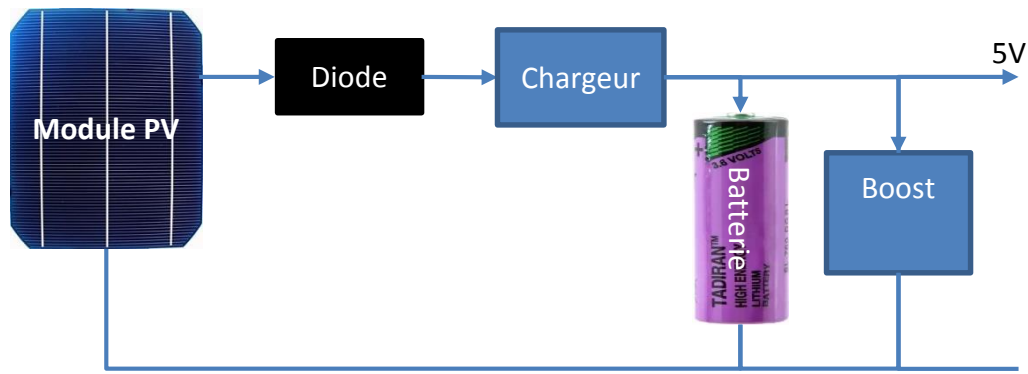


Figure 14. Configuration pour l'alimentation du système

### 5.2.1. Chargeur de batterie

Nous avons choisi le chargeur de batterie TP4056, qui est le plus utilisé pour les systèmes embarqués, il est représenté sur la figure suivante :

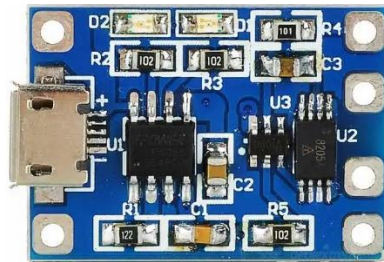


Figure 15. Chargeur de batterie Li-Ion TP4056

Le TP4056 est un chargeur linéaire complet à courant constant / à tension constante pour une seule cellule batteries lithium-ion. Son faible nombre de composants du TP4056 idéalement adapté aux applications portables. En outre, le TP4056 peut fonctionner à l'intérieur et au mur adaptateur. La tension de charge est fixée à 4.2V, et le courant de charge peut être programmé extérieurement avec une seule résistance. Le TP4056 termine automatiquement le cycle de charge lorsque le courant de charge tombe au 1/10 de la valeur programmée. Le module que nous avons choisi, peut générer un courant de charge de 1 A [22].

### 5.2.2. Convertisseur Boost

Nous avons choisi d'utiliser un convertisseur continue/continue de type Boost qui fournit une tension de 5V, il existe en vente des modules intégrés tel que celui de la figure ci-dessous.



**Figure 16. Module Boost USB 5V**

Le Boost utilisé dans notre projet à les caractéristiques suivantes :

- Tension d'entrée : 0.9V-5V DC ;
- Rendement : 96% (max) ;
- Port USB ;
- Courant de sortie max 600mA.

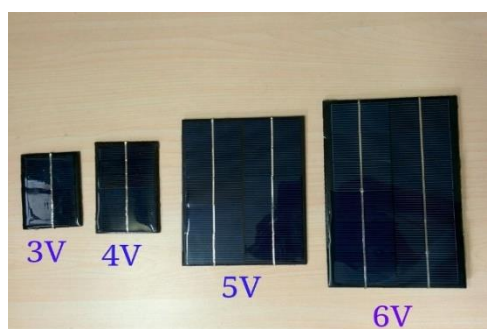
### 5.2.3. Choix des batteries Li-Ion

Dans le cas où le système est alimenté uniquement par une batterie Li-Ion, des calculs rigoureux permettrons de calculer la consommation du système sur une périodes importante (de plusieurs mois), et de déduire ainsi la taille de la batterie.

Sinon dans le cas d'une alimentation avec module solaire, nous pouvons utiliser des batteries de 1300mAh qui seront largement suffisante pour faire fonctionner le système, notamment en l'absence du soleil.

### 5.2.4. Choix du module PV

La source principale d'alimentation du module de capteur est le panneau solaire. Il doit donc être capable de fournir le courant pour alimenter le système, ainsi que le courant pour charger la batterie pendant la journée.



**Figure 17. Module photovoltaïque**

Pour bien choisir le meilleur compromis, il faut suivre les règles empiriques suivantes :

- Tension : Choisissez 1,5 fois la tension de la batterie ;
- Courant : Courant tiré par le système + courant pour la charge de la batterie.

Le tableau suivant montre la relation qui lie la tension de la batterie avec celle du module photovoltaïque.

**Tableau 8. Relation entre la tension de la batterie et celle du module PV.**

Batterie	Module PV
1.2V	2V ~ 2.5V
2.4V	3.5V ~ 4V
3.6V	5V ~ 6V
6V	7.5V ~ 9V
12V	15V ~ 18V

Nous allons ainsi choisir un module photovoltaïque qui génère une tension de 6V avec une puissance de 3W (figure 18), ce qui nous donne un courant maximum de 600mA.



**Figure 18. Module photovoltaïque choisi**

## **6. Conclusion**

Dans ce chapitre, nous avons présenté les différents composants qui nous permettront la mise en œuvre de notre système. Ces descriptions détaillées nous permettent de bien comprendre le fonctionnement de chaque élément, mais surtout la meilleure façon de les connecter ensemble, pour permettre les meilleures performances possibles du système.

# **Chapitre IV**

## **Réalisation du projet**

## 1. Introduction

Le troisième chapitre nous a permis de présenter les différentes parties nécessaires pour la conception de notre système, ainsi que les différents composants entrants dans sa mise en œuvre. Nous allons dans ce chapitre présenter les étapes pour la réalisation effective de notre projet. Nous aurons à voir aussi les différents algorithmes qui régissent le fonctionnement du système.

## 2. Structure détaillée du système

La figure 1 montre une vue détaillée de tous les éléments qui sont nécessaire pour la construction d'un prototype du système. Trois parties sont essentielles pour son fonctionnement.

- ✓ La première partie concerne la réalisation de système électronique de la ruche. Cette partie contient une carte Arduino NANO, un capteur DHT22 qui permet de mesurer la température et l'humidité, des jauges de contrainte pour mesurer le poids et un émetteur récepteur nRF24L01 pour faire la liaison avec la station (deuxième partie du prototype).
- ✓ La deuxième partie concerne la réalisation de la station. Cette partie contient également un carte Arduino NANO, un capteur BMP280 qui permet de mesurer la pression atmosphérique, Un émetteur-récepteur nRF24L01 permet de relier la station à les ruches (première partie du prototype), et un module GSM SIM800L pour se connecter au réseau GSM ou GPRS.
  - Le système d'alimentation pour les deux parties précédentes (Chaque partie a sa propre alimentation), elle est constituée principalement d'un module photovoltaïque, une batterie, un chargeur TP4056 et un convertisseur Boost.
- ✓ La troisième partie concerne la récupération et la visualisation des données mesurées.

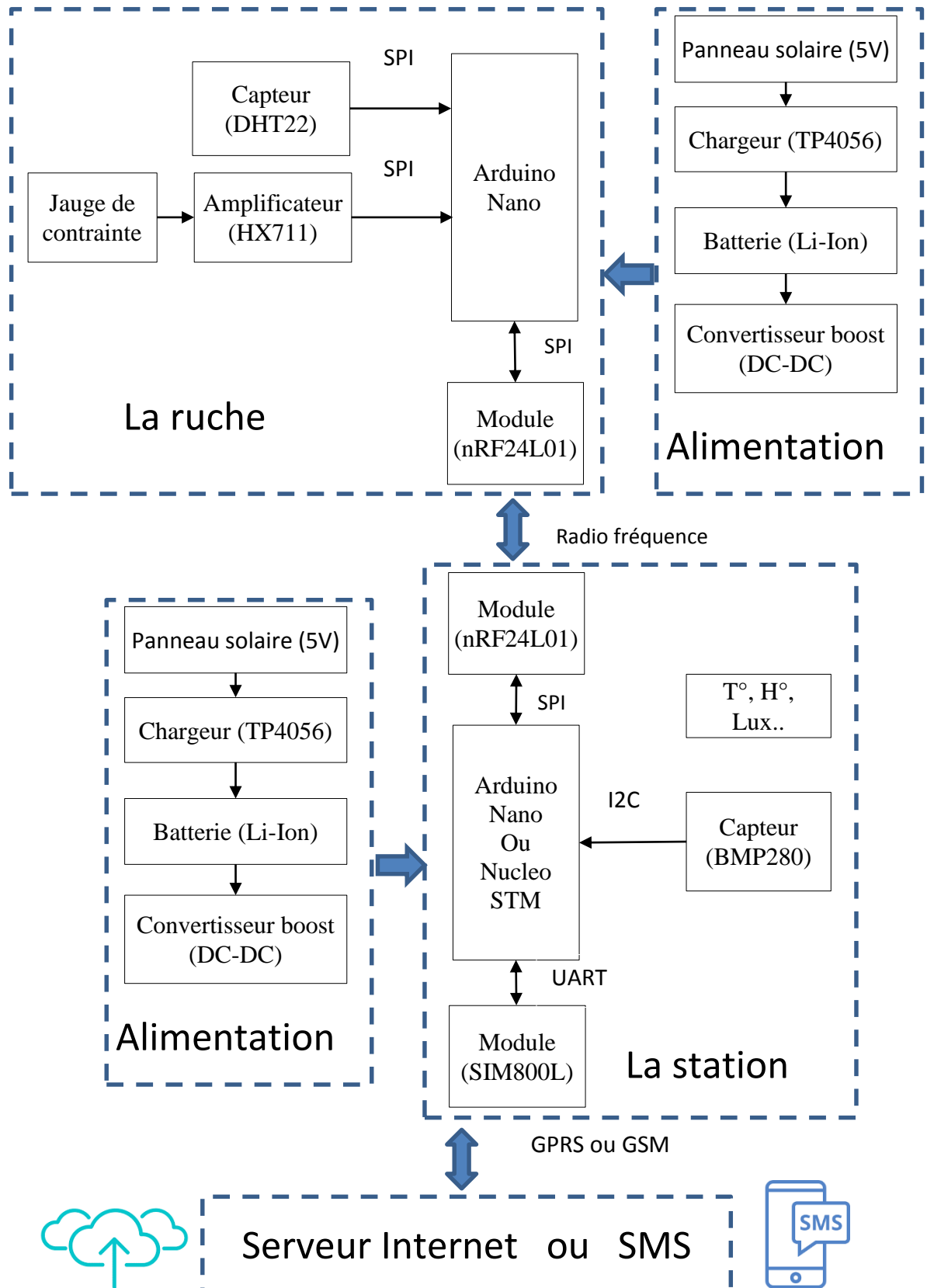


Figure 1. Schéma synoptique du système

### 3. Sous-système de la ruche

Chaque ruche dispose d'un sous-système électronique qui mesure la température, l'humidité et le poids lorsqu'elle reçoit une demande de la station, elle lui renvoie instantanément les données mesurées. Ceci est illustré par le diagramme suivant :

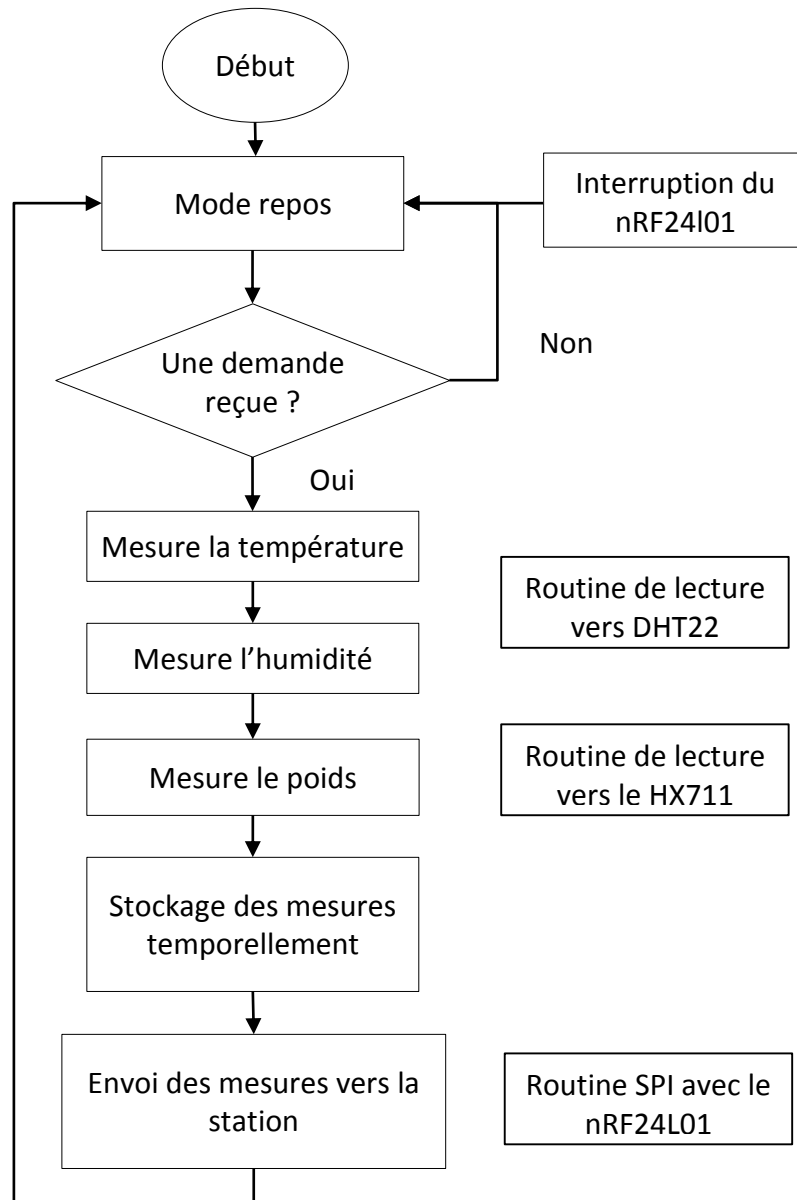


Figure 2. Organigramme de l’algorithme qui gère le sous-système de la ruche

Le microcontrôleur est mis en mode de veille en attente d'une interruption émanant du module RF, à la réception d'une demande, le microcontrôleur récupère les données à partir des capteurs, ceci en utilisant pour chaque capteur une routine spécifique à son protocole de communication.

Le mode d'interruption utilisé optimise grandement l'énergie consommée par la carte Arduino et les différents capteurs connectés.

### 3.1. Mesurer la température et l'humidité de la ruche

Pour mesurer la température et l'humidité des ruches, nous utilisons le capteur DHT22 décrit dans le chapitre 3. Le DHT22 est un capteur numérique capable de mesurer des températures allant de  $-40$  à  $+125^{\circ}\text{C}$  et des humidités allant de 0 à 100%. Il a trois broches : GND, VCC et DATA.

La broche VCC du capteur est reliée à l'alimentation 5V et la broche GND du capteur va être reliée avec la masse et la broche DATA reliée avec le pin D4 de l'Arduino NANO. Le schéma de branchement du capteur DHT22 avec la carte Arduino NANO devient :

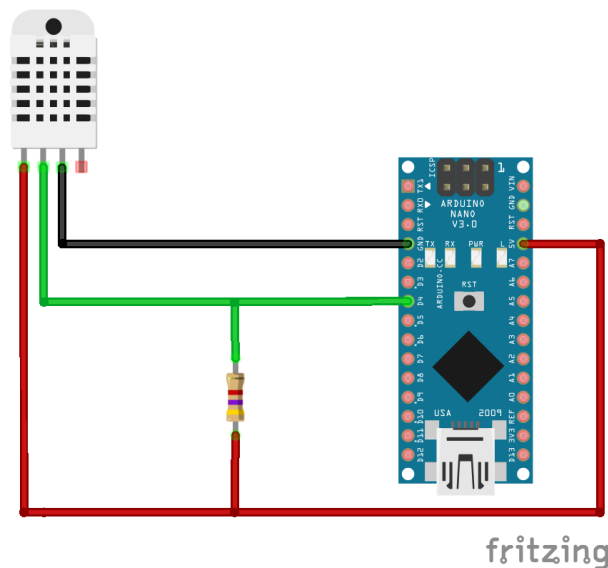


Figure 3. Schéma du DHT22 avec le microcontrôleur NANO.

Le montage de test est illustré dans la photo de la figure 4, la communication entre la carte Arduino et le capteur est gérée à travers une routine de communication semblable au protocole RS232.

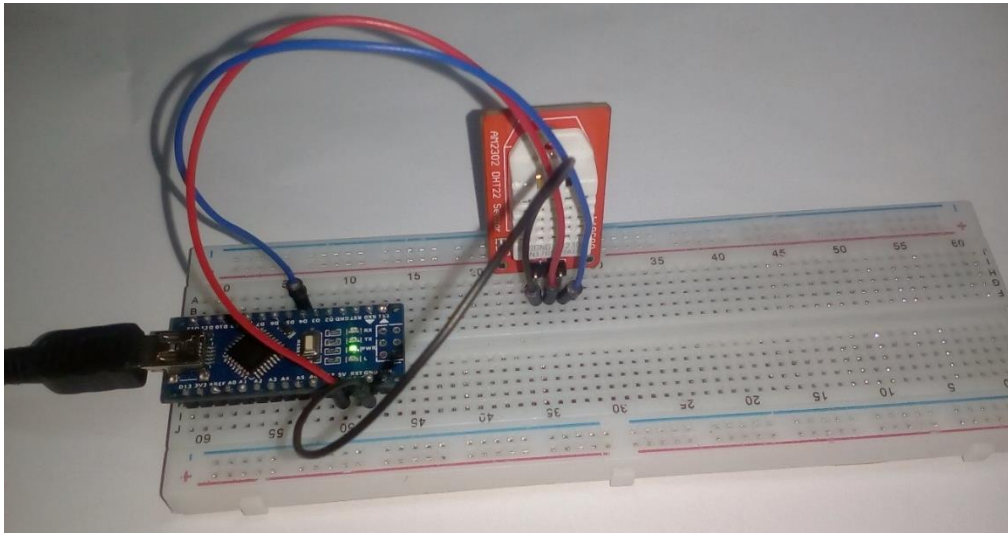


Figure 4. Montage du capteur DHT22 avec la carte Arduino Nano

**Remarque :** Une résistance de 4.7 K $\Omega$  entre la broche 5V de l'alimentation et la broche de données permet d'amplifier le signal généré. Elle est absente dans notre montage de la figure 4 par ce qu'elle est intégrée dans le module que nous avons choisi.

### 3.2. Mesurer le poids de la ruche

Pour mesurer les poids des ruches nous utilisons une balance composée de 4 cellules de charge de 50kg chacune, ces cellules sont connectées à la carte Arduino au travers d'un module HX711 pour numériser le signal mesuré.

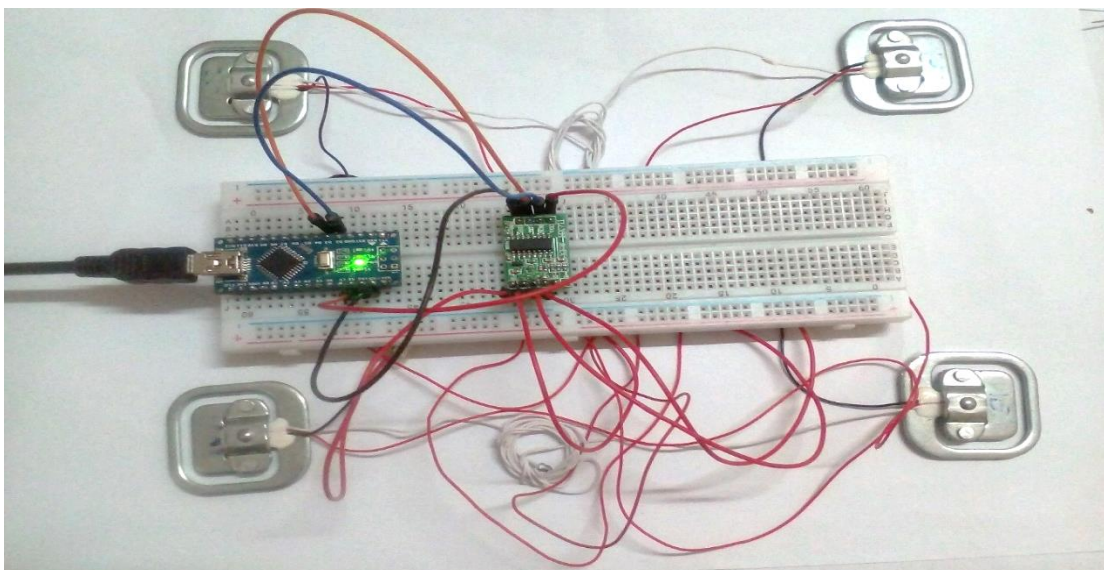


Figure 5. Montage des cellules de charge avec HX711 et la carte Arduino Nano

Pour la communication entre la carte Arduino et le DHT711, nous avons utilisé le protocole spécifique qui permet de contrôler la broche PD\_SCK du HX711, ceci afin de récupérer la mesure du poids effectuée.

**Remarque :** Après la branchement (cellule de charge, l'amplificateur et l'Arduino NANO) il faut ajouter un code d'étalonnage pour rendre le HX711 opérationnel.

### 3.3. Envoyer et recevoir les données de la station

Pour communiquer avec la station de base, nous utilisons le module nRF24L01 décrit dans le chapitre 3, ce module permet d'émettre les données vers la station et de recevoir les données par radio fréquence.

Le branchement du module nRF24L01 avec le microcontrôleur NANO se fait à travers l'utilisation du bus de communication SPI du microcontrôleur (figure 6).

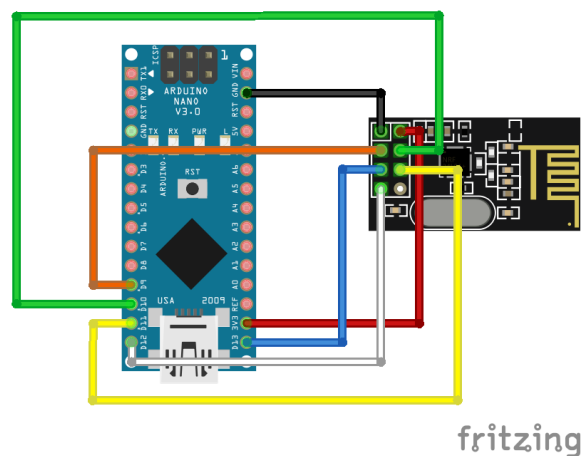


Figure 6 : Branchement du module nRF24L01 avec le microcontrôleur NANO

#### Procédure d'utilisation :

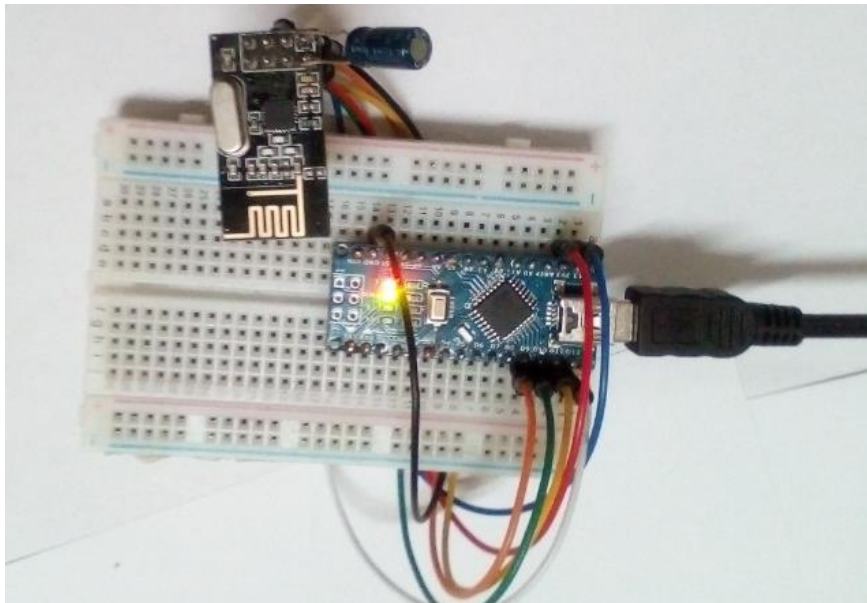
Afin de pouvoir utiliser ce module il faut suivre les étapes suivantes :

- ✓ Initialiser la communication SPI ;
- ✓ Fixer l'adresse du nœud ;
- ✓ Fixer la fréquence du signal ;
- ✓ Fixer le débit de transmission ;
- ✓ Fixer la puissance du signal ;
- ✓ Mettre le module en attente de réception (Coté ruche).

Ces configurations sont effectuées à travers la configuration d'un certain nombre de registres (intégrés dans le nRF24L01), ceci au travers des instructions spécifiques. La communication entre la carte Arduino et le module est basé sur le protocole SPI.

**Remarques :**

1. La broche VCC du module nRF24L01 doit être reliée à une alimentation de 3.3V.
2. Au moment de la transmission des données, le module NRF24L01 a besoin d'une certaine quantité de courant (plus que d'habitude) donc il est nécessaire de mettre un condensateur en parallèle à la sortie du régulateur afin d'assurer une bonne transmission des données.



**Figure 7 : Montage du branchement du module nRF24L01 avec le microcontrôleur NANO**

4. Sous-système de la station

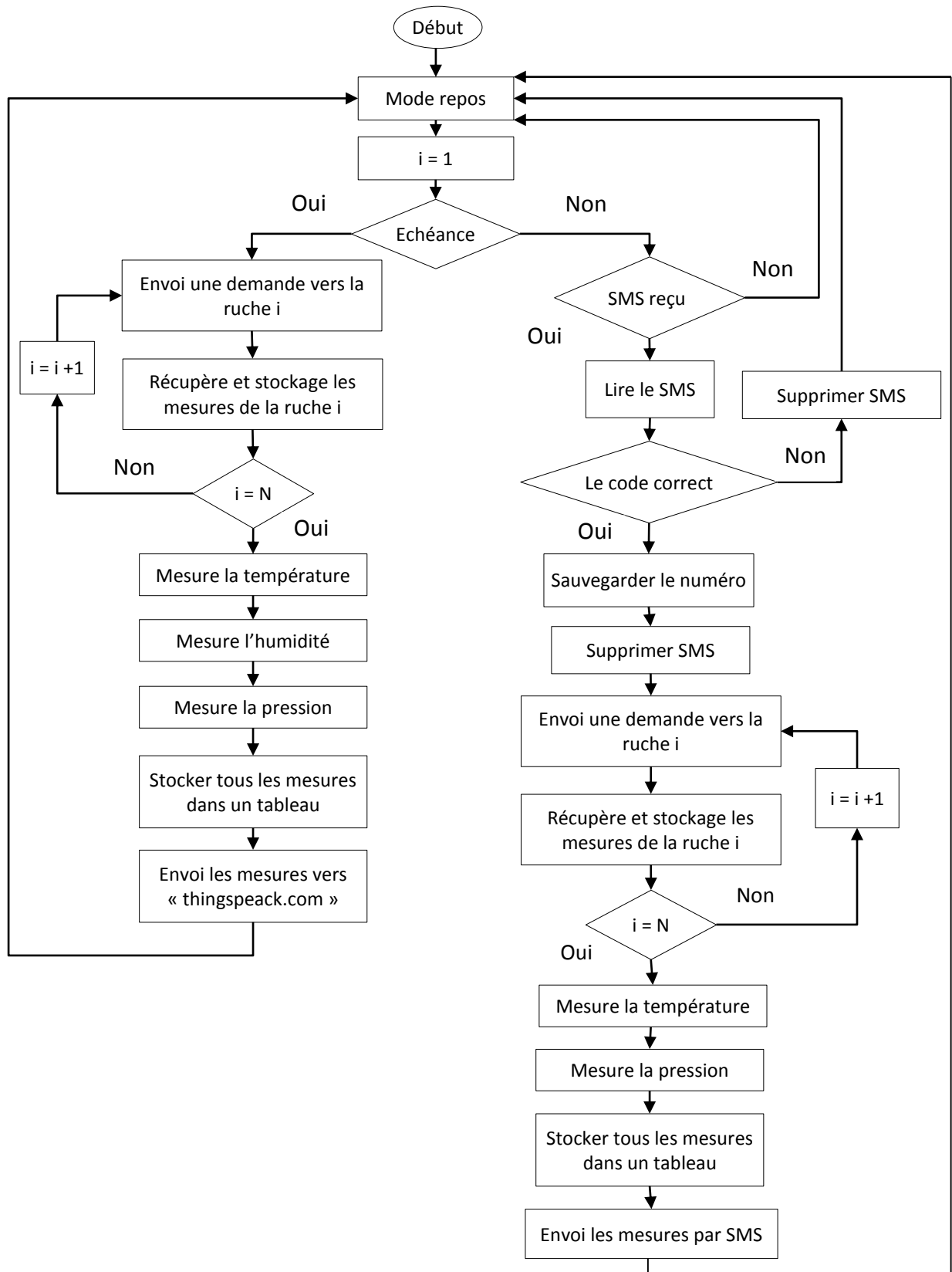


Figure 8. Organigramme du fonctionnement de la station

La station est l'appareil qui collecte les informations à partir des ruches, elle envoie des demandes vers les ruches successivement, puis elle stocke et envoie ces mesures (Température, humidité, poids...) vers un serveur (thingspeak.com). La station permet aussi aux utilisateurs de récupérer ces mesures sous forme d'un SMS. Ceci est illustré par l'organigramme précédent (Figure 8).

Donc, la station doit assurer les fonctionnalités suivantes :

- Collecte des mesures à partir des N ruches ;
- Mesure de la température ambiante et la pression atmosphérique ;
- Envoi des mesures vers le site « Thingspeak.com » périodiquement ;
- Envoi des mesures par SMS quand il y'a une demande ;

**Remarque :** le 'N' dans l'organigramme précédent représente le nombre de ruches et le 'i' représenter l'indice de 'N'.

#### 4.1. Mesurer la pression atmosphérique et de la température

Pour mesurer la pression atmosphérique et la température ambiante, nous utilisons le capteur BMP280. On connecte simplement la broche *Vin* du capteur à la broche de tension 5V, GND à la masse, *SCL* à la broche 5 et *SDA* à la broche 4 (Figure 9). Pour récupérer les données du capteur, il suffit d'utiliser l'interface I2C pour lire le contenu de la mémoire du capteur, celle-ci renferme les mesures de pression et de température.

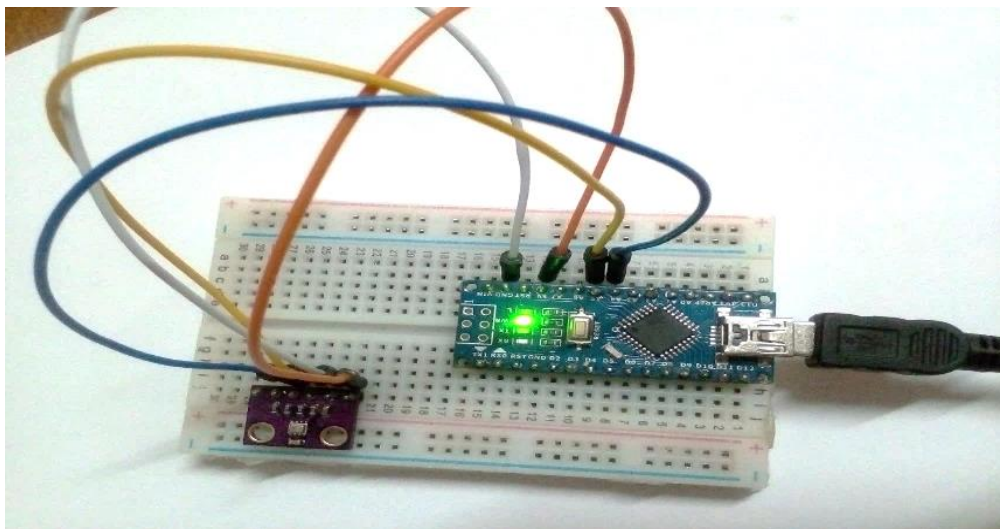


Figure 9. Montage du capteur BMP280 avec la carte Arduino

## 4.2. Envoyer et recevoir les données de la station

Comme nous l'avons décrit dans le sous-système de la ruche, la communication entre la station de base et les différents nœuds du rucher se fait par radio fréquence, à travers l'utilisation des module nRF24L01.

Contrairement au sous-système de la ruche la procédure de fonctionnement du modèle est différente :

- ✓ Initialiser la communication SPI ;
- ✓ Fixer l'adresse du nœud ;
- ✓ Fixer la fréquence du signal ;
- ✓ Fixer le débit de transmission
- ✓ Fixer la puissance du signal ;
- ✓ Fixer la taille du paquet (de 1 à 32 octet) ;
- ✓ Choisir l'adresse du nœud  $i$  ;
- ✓ Envoi d'une requête vers le nœud  $i$  ;
- ✓ Attente de l'accusé de réception ;
- ✓ Attente de la réponse du nœud ;
- ✓ Refaire les 4 dernière étape pour le nœud  $i+1$ .

## 4.3. Envoyer les mesures par le module SIM800L

On utilise le module GSM SIM800L pour envoyer les mesures vers le site web, et pour envoyer les messages SMS.

Le module SIM800 est un téléphone GSM simple, sans clavier, écran, micro ni haut-parleur mais possédant une liaison série de type RS232, qui permet de le connecté à n'importe quel microcontrôleur ou carte de développement, carte Arduino Nano ou Nucleo 64.

La figure suivante montre le branchement du module SIM800L avec la carte Arduino Nano, où les connexions nécessaires sont :

- La tension d'alimentation Vcc (de 3.4 à 4.4V) ;
- La masse ;
- La broche TX émet vers le RX de l'Arduino, ici D7 ;
- La broche RX reçoit du TX de l'Arduino, ici D8.

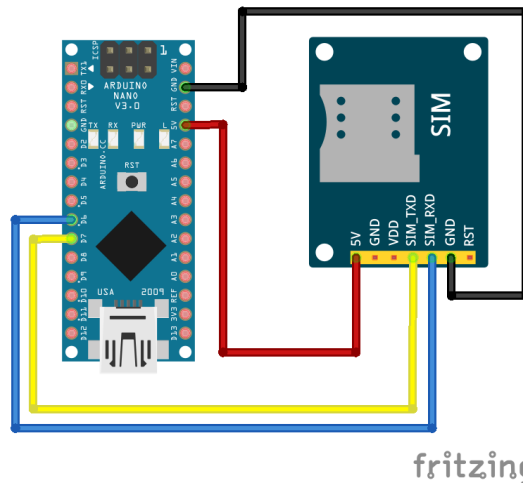


Figure 10. Branchement du module SIM800L avec la carte Arduino

**Remarque :** La position des broches TX et RX inversées selon les modules... (et non repérées pour certains !). Donc si rien ne fonctionne inverser ces connexions.

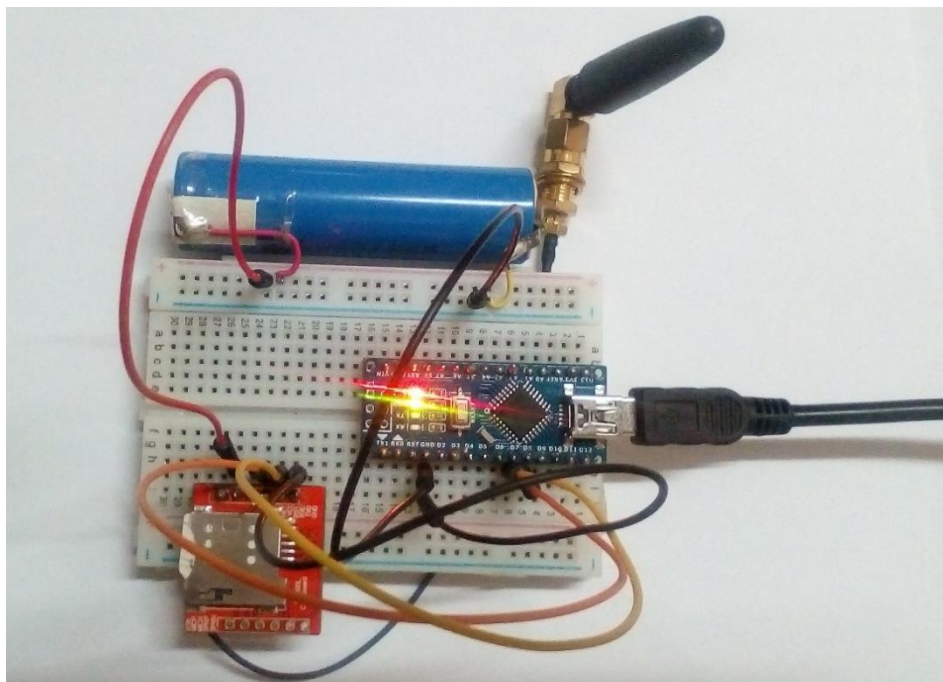


Figure 11. Montage du module SIM800L avec la carte Arduino



## 5. Vérification et test

Après avoir testé les différents éléments du système, nous avons pu rassembler tous les éléments pour vérifier les fonctionnalités du système, particulièrement la communication RF et la communication GSM.

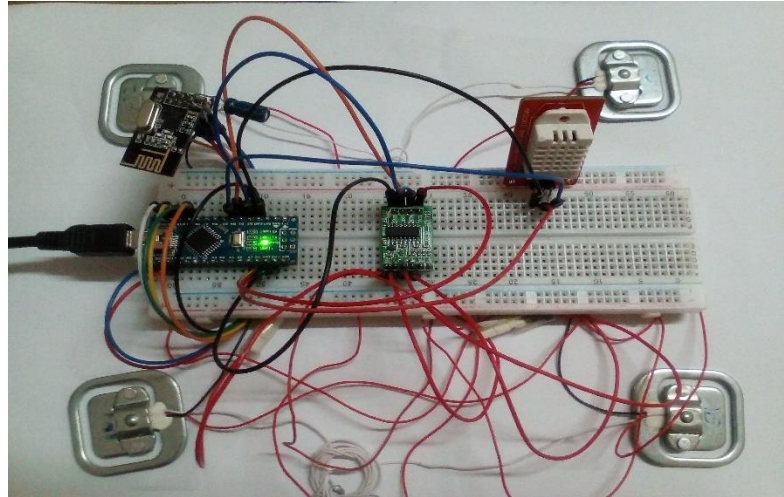


Figure 13 : Montage complet de système électronique de la ruche

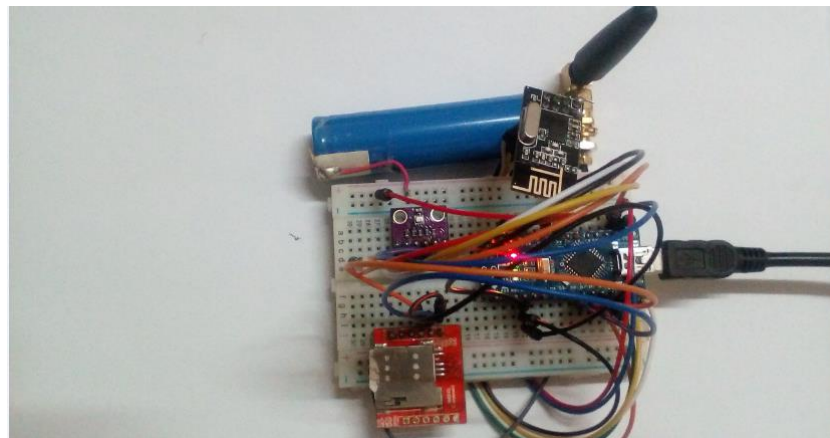


Figure 14 : Montage complet de système électronique de la station

### 5.1. Choix du moyen de visualisation

Pour que l'utilisateur ou plus précisément l'apiculteur accède aux données récoltées, nous avons choisi d'utiliser un serveur d'objets connectés, plusieurs serveurs existent et qui fournissent des services incluant entre autres des algorithmes de traitement et d'analyse de données.

Pour notre projet, nous avons choisi une solution simple et adaptée aux besoins de notre système. Ainsi nous avons choisi le site « **ThingSpeak** » pour la récupération et l’affichage des résultats en temps réel. Ce site est une entreprise qui propose différents services exclusivement destinés à la construction d'applications IOT et qui revendique être une plateforme gratuite.

Le site propose un support d'IOT en permettant de :

- Collecter les données en temps réel (fréquence supérieure ou égale à 15 secondes) ;
- Visualiser les données collectées sous forme de graphes ;
- Créer des plugins et des applications pour collaborer avec des web services, des réseaux sociaux et d'autres APIs.

Pour pouvoir utiliser le serveur **ThingSpeak** pour la visualisation des données de notre système, il faut suivre les étapes décrites ci-dessous [16] :

- S’inscrire au site (<https://thingspeak.com/>) ;
- Créer le nombre de canaux (Channels) nécessaires ;
- Créer les champs (fields) de donnée, chaque canal peut stocker plusieurs champs (Température, humidité, le poids et la pression...) ;
- Récupérer la clé de mise à jour (API Key ; Write KEY).

Parallèlement, pour pouvoir envoyer les données vers le site ThingSpeak, il faut utiliser la clé fournie lors de l’inscription, cette clé est introduite dans le code source de la routine qui communique avec le module SIM800l, ainsi nous pouvons envoyer les données dans n’importe quel champ, et dans n’importe quel canal.

## 5.2. Résultats

Après l’installation du prototype de test de notre système, qui englobe une station de base et deux ruches connectées. Nous avons effectué quelques tests pour vérifier le bon fonctionnement du système. Les figures suivantes représentent l’évolution des différents paramètres mesurés. Nous avons récupéré ces courbes du serveur **ThingSpeak**. Nous avons pris ces mesures pour la période entre 3 mai et 13 mai 2018.

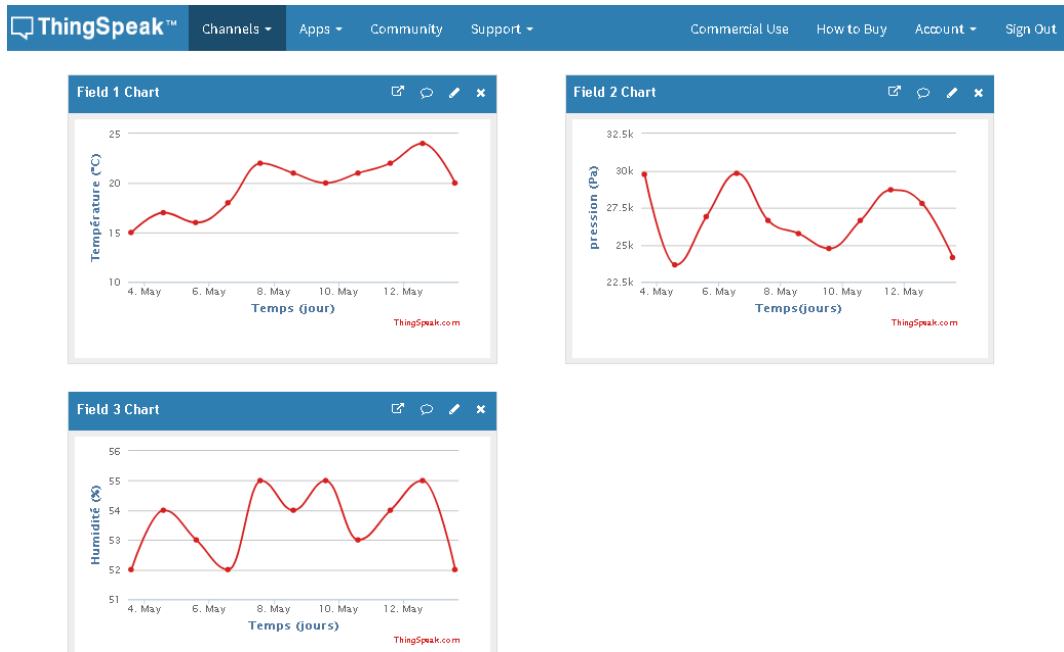


Figure 15. Graphiques montrants la température et la pression de la station

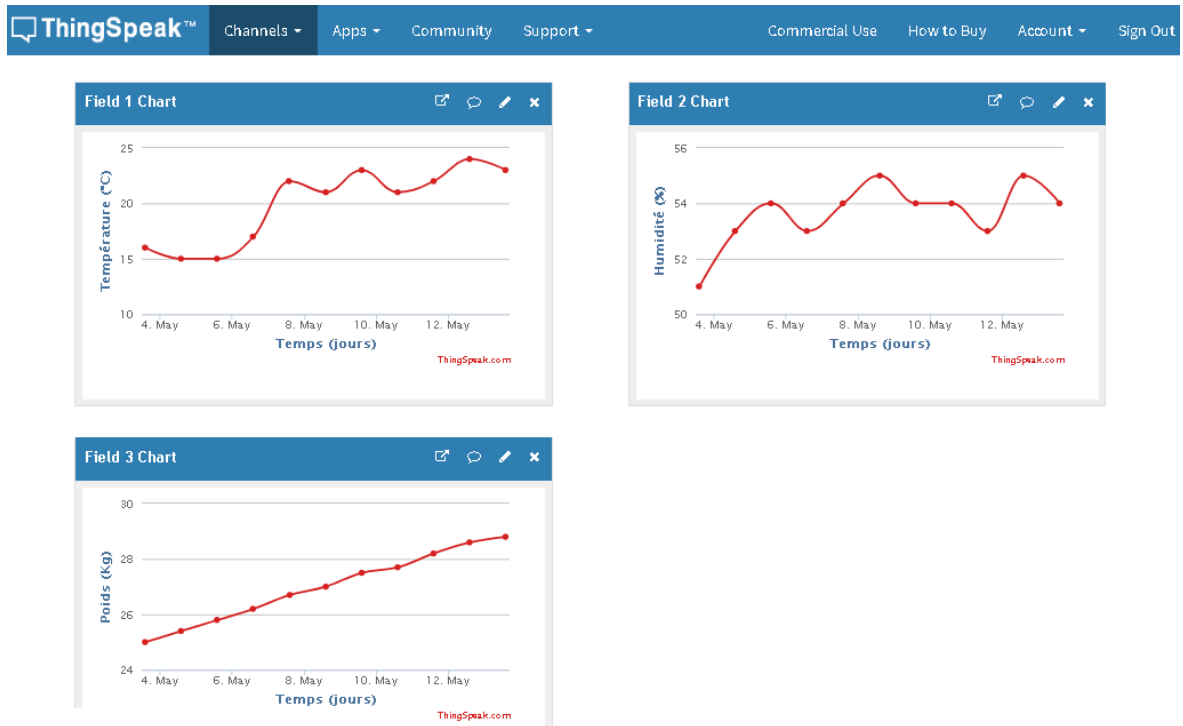


Figure 16. Graphiques montrants la température, l'humidité et le poids de la ruche 1

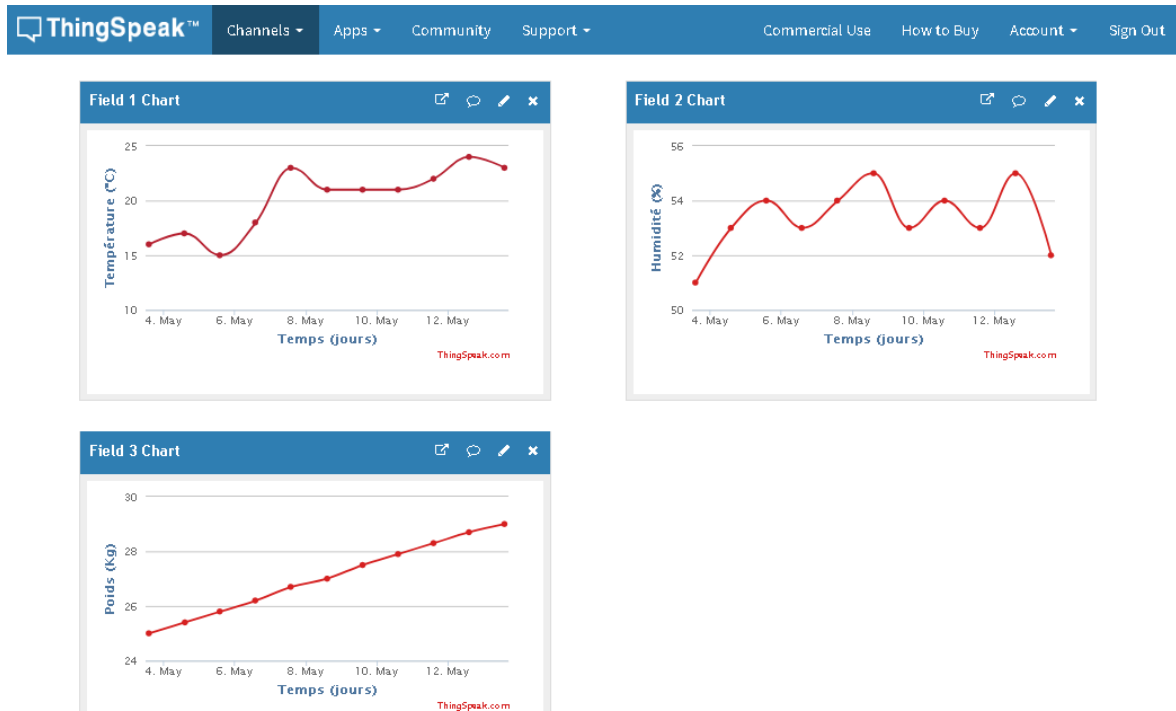


Figure 17. Graphiques montrants la température, l'humidité et le poids de la ruche 2

### 5.3. Récupérer les données par SMS

Nous avons ajouter au système une option qui permet la consultation de l'état du rucher, en utilisant un message SMS. Pour cela il faut envoyé un SMS vers le numéros de la carte SIM installé dans le système, ce message doit contenir un code défini auparavant. Si le code est correcte le système envoi un résumé des donnée collecté de chaque ruche ainsi que les données climatiques.

## 6. Conclusion

Au début de ce chapitre, nous avons fourni une explication détaillée de la méthode de conception et réalisation de nos ruches connectées.

Nous avons fait des mesures expérimentales en dehors de laboratoire pour voir comment les phénomènes physiques variés (la température, la pression, l'humidité et la poids), les résultats des mesures sont montrés sous forme des graphes en fonction du temps.

### **Conclusion générale**

Notre projet de fin d'études consiste à faire la conception et la réalisation d'un rucher connecté, basé sur la technologie de l'internet des objets (IoT). Il permet entre autres de mesurer et de collecter les paramètres climatiques (Température et l'humidité) et le poids des ruches, pour ensuite les envoyer vers un serveur distant.

Pour accomplir ce travail, nous avons dû suivre les étapes suivantes :

- La première étape était consacrée à l'étude et à la recherche bibliographique et technique détaillée sur la technologie des systèmes de ruches connectés. Cette étude nous a permis d'avoir une idée générale du premier modèle de notre système.
- La deuxième étape consiste à l'écriture du cahier des charges ; où nous avons étudié le matériel nécessaire avec les aspects techniques et fonctionnels de chaque élément.
- La troisième étape consiste à la réalisation du projet, pour cela nous avons commencé par la réalisation du système électronique des ruches, en analysant les exigences et les schémas techniques du système, puis nous avons présenté la réalisation et la conception détaillée de la station.

Grace à ce projet nous avons pu nous familiariser avec :

- L'intégration des modules de capteur (DHT22, BMP280, HX711...) et des modules de communication (nRF24L01, SIM800L ...), dans les systèmes embarqués ;
- Nous avons aussi maîtrisé la programmation des cartes Arduino ainsi que les cartes Nucleo à base de  $\mu$ C STM32 ;
- Finalement, nous avons appris le concept pratique de systèmes embarqué et de l'internet des objets.

## **Conclusion générale**

---

Comparé aux autres réalisations que nous avons consultés dans la bibliographie, et les produits commercialisés, notre système comporte les points forts suivants :

- Le nombre de ruche qu'on peut grouper et gérer dans notre système est au maximum de 127 ruche connectée ;
- Le système est basé sur l'utilisation du réseau GSM ou GPRS ;
- La station de base peut être distante des ruches de plusieurs centaines de mètre ;
- Le système consomme peu d'énergie ;
- Un module photovoltaïque le rend autonome ;
- Le système est reconfigurable ;
- Le système est évolutif ;

En perspectives, nous visons à améliorer la conception du hardware et du software, en effet nous proposons :

- D'ajouter plus de capteurs au système tels que des capteurs : accéléromètre, l'orientation de vent, antivol...
- Créer une application web spéciale pour notre système ;
- Ajouter des algorithmes de traitement de donnée, pour suggérer les actions que doit faire l'apiculteur pour améliorer le rendement de son rucher.
- Ajouter un système d'exploitation à la base ;
- Ajouter des extras tels que la réception d'alertes dans l'email, l'affichage des mesures sur des écrans LCD montés sur le système de base, caméras de surveillance....
- Améliorer encore son efficacité énergétique.

## Bibliographies & Webographies

- [1] Mr ACHAT Asalas et Mr LAOUBI Lyes, 'Conception et réalisation d'une application mobile cross Platform pour l'Internet of things', Mémoire de fin d'étude, Université A/Mira de Bejaia, Juin 2017.
- [2] <http://www-igm.univ-mlv.fr/~dr/XPOSE2002/SE/architecture.html> (consulté en mai 2018).
- [3] <http://www.orsys.fr/mail/other/doc/041215SystemesEmbarques.pdf> (consulté en mai 2018).
- [4] <https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/> (consulté en mai 2018).
- [5] [www.cisco.com/c/dam/global/en\\_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf](http://www.cisco.com/c/dam/global/en_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf) (consulté en mai 2018).
- [6] <https://www.greenschoolsalliance.org/blogs/16/427> (consulté en mai 2018).
- [7] <http://www.alya.sk/vcelarska-vaha-vilko> (consulté en mai 2018).
- [8] <https://www.label-abeille.org/fr/> (consulté en mai 2018).
- [9] [http://bienenwaage.de/pdf\\_biene/gsm\\_inff.pdf](http://bienenwaage.de/pdf_biene/gsm_inff.pdf) (consulté en mai 2018).
- [10] <http://itsap.asso.fr/outils/balances-automatiques/> (consulté en mai 2018).
- [11] <http://www.arnia.co.uk/how-it-works/> (consulté en mai 2018).
- [12] <https://www.bee-online.fr> (consulté en mai 2018).
- [13] <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/> (consulté en mai 2018).
- [14] <http://www.projetsgeii.iutmulhouse.uha.fr/ruche-connectee/> (consulté en mai 2018).
- [15] <http://www.instructables.com/id/Arduino-Bathroom-Scale-With-50-Kg-Load-Cells-mai-18and-H/> (consulté en mai 2018).
- [16] <https://wiki.mchobby.be/index.php?title=FEATHER-MICROPYTHON-BME280> (consulté en mai 2018).
- [17] <https://youpilab.com/gsm-gps/module-gsm-sim-800-l.html> (consulté en mai 2018).

- [18] <https://letmeknow.fr/blog/2015/10/14/tuto-module-gsm-sim800l-prise-en-main/> (consulté en mai 2018).
- [19] Ouali alami mohammed, "La Conception d'une prise connectée basée sur la technologie d'IoT (Internet of Things)", mémoire d'ingénieur, Université Sidi Mohamed Ben Abdellah-Maroc, Juin 2016.
- [20] Kaissari Soufiane, "Conception d'un Réseau de Capteurs Sans Fil", mémoire de Master, Université Mohammed V de rabat, juillet 2015.
- [21] Marc Laury, "A la découverte des cartes nucleo", éditions eyrolles, 2015
- [22] [https://www.e-gizmo.net/oc/kits\\_documents/TP4056-1A\\_Li-ion\\_Battery\\_Charger/TP4056-1A\\_Li-ion\\_battery\\_charger\\_Manual.pdf](https://www.e-gizmo.net/oc/kits_documents/TP4056-1A_Li-ion_Battery_Charger/TP4056-1A_Li-ion_battery_charger_Manual.pdf) (consulté en mai 2018).
- [23] Vasileios Karagiannis, 'Building a Testbed for the Internet of Things', Doctoral thesis, School of Technological Applications Barcelona - April 2014.

## Annexes

- ❖ Annexe1 : datasheet stm32l476

<http://www.st.com/content/ccc/resource/technical/document/datasheet/c5/ed/2f/60/aa/79/42/0b/DM00108832.pdf/files/DM00108832.pdf/jcr:content/translati ons/en.DM00108832.pdf>

- ❖ Annexe2 : datasheet arduino nano

<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>

- ❖ Annexe3 : datasheet sim800l

[http://simcom.ee/documents/SIM800/SIM800\\_Hardware%20Design\\_V1.08.pdf](http://simcom.ee/documents/SIM800/SIM800_Hardware%20Design_V1.08.pdf)

- ❖ Annexe4 : datasheet nrf24l01

[https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus\\_Preliminary\\_Product\\_Specification\\_v1\\_0.pdf](https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus_Preliminary_Product_Specification_v1_0.pdf)

- ❖ Annexe5 : programme de ruche sur Arduino IDE

```
#include <SPI.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <DHT.h>
#include "DHT.h"
#include "HX711.h"

#define CE_PIN 9
#define CSN_PIN 10
#define DHTPIN 4
#define DHTTYPE DHT22
#define calibration_factor 7050.0
#define DOUT 3
#define CLK 2

HX711 scale(DOUT, CLK);
const uint64_t Ruche1=0xE8E8F0F0E1LL;
const uint64_t Station=0x0102030405LL;
DHT dht(DHTPIN, DHTTYPE);
RF24 radio(CE_PIN, CSN_PIN);
uint8_t demande=88;
int mesures[3];
void setup()
{
  dht.begin();
  Serial.begin(9600);
```

```

radio.begin();
radio.setDataRate( RF24_250KBPS ) ;
radio.setPALevel(RF24_PA_MIN);
radio.setChannel(1);
radio.setRetries(15,15);
scale.set_scale(calibration_factor);
scale.tare();
radio.openReadingPipe(1,Ruchel);
radio.openWritingPipe(Station);
radio.startListening();
Serial.println("en attente...");
}

void loop()
{
    if( radio.available() )
    {
        Serial.println("un demande recu");
        mesures[0] = dht.readTemperature();
        mesures[1] = dht.readHumidity();
        mesures[2] = scale.get_units();
        Serial.println(dht.readTemperature());

        delay(100);
        radio.read( &demande, sizeof(demande) );
        radio.stopListening();
        delay(4000);
        radio.write(& mesures, sizeof(mesures) );
        Serial.println("les mesure est envoi vers la station");
        radio.startListening();
    }
}

```

## ❖ Annexe6 : programme de station sur Arduino IDE

```

#include <SPI.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>

#define CE_PIN 9
#define CSN_PIN 10

Adafruit_BMP280 bmp;

const uint64_t Ruchel=0xE8E8F0F0E1LL;
const uint64_t Station=0x0102030405LL;
const uint64_t Ruche2=0xFFFF8F0F0E1LL;

SoftwareSerial mySerial (7, 8);
RF24 radio(CE_PIN, CSN_PIN);
uint8_t d = 15;

```

```

int Ruche_1[3];
int Ruche_2[3];
int Station_t[9];

void setup()
{

  radio.begin();
  Serial.begin(9600);
  mySerial.begin(9600);

  radio.setDataRate( RF24_250KBPS );
  radio.setPALevel(RF24_PA_MIN);
  radio.setChannel(1);
  radio.setRetries(15,15);
  radio.openWritingPipe(Ruche1);
  radio.openReadingPipe(1,Station);
  bmp.begin(0x76);
  bool y = radio.write(& d, sizeof(d) );
  Serial.println(y);
  Serial.println("le demande est envoi vers le Ruche1");
  radio.startListening();

}

void loop()
{
  if( radio.available() )
  {
    radio.read( &Ruche_1, sizeof( Ruche_1));
    for(int i=0; i<3; i++)
    {
      Station_t[i] = Ruche_1[i];
      Serial.println(Ruche_1[i]);
      delay(1000);
    }
    radio.stopListening();
    radio.openWritingPipe(Ruche2);
    radio.write(& d, sizeof(d) );
    Serial.println("le demande est envoi R2");
    radio.startListening();
    while(1)
    {
      if( radio.available() )
      {
        radio.read( &Ruche_2, sizeof( Ruche_2));
        for(int i=0; i<3; i++)
        {
          Station_t[i+3] = Ruche_2[i];
          Serial.println(Ruche_2[i]);
          delay(1000);
        }
        Serial.println("tout les mesures:");
        Station_t[6] = bmp.readTemperature();
        Station_t[7] = bmp.readPressure();
        Station_t[8] = bmp.readAltitude(1013.25);
        for(int i=0; i<9; i++)
        {

          Serial.println(Station_t[i]);
          delay(1000);
        }
      }
    }
  }
}

```

```

        }
        versThingSpeak();
    }

}

}

}

void versThingSpeak()
{
    mySerial.println("AT");
    delay(1000);
    mySerial.println("AT+CPIN?");
    delay(1000);
    mySerial.println("AT+CREG?");
    delay(1000);
    mySerial.println("AT+CGATT?");
    delay(1000);
    mySerial.println("AT+CIPSHUT");
    delay(1000);
    mySerial.println("AT+CIPSTATUS");
    delay(2000);
    mySerial.println("AT+CIPMUX=0");
    delay(2000);
    ShowSerialData();
    mySerial.println("AT+CSTT=\"internet\""); //start task and
setting the APN,
    delay(1000);
    ShowSerialData();
    mySerial.println("AT+CIICR"); //bring up wireless connection
    delay(3000);
    ShowSerialData();
    mySerial.println("AT+CIFSR"); //get local IP adress
    delay(2000);
    ShowSerialData();
    mySerial.println("AT+CIPSPRT=0");
    delay(3000);
    ShowSerialData();
    for(int i=0 ; i <3; i++)
    {

        int j=i+1;

mySerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\""); //start up the connection
        delay(6000);
        ShowSerialData();
        mySerial.println("AT+CIPSEND"); //begin send data to remote
server
        delay(4000);
        ShowSerialData();
        String str="GET
https://api.thingspeak.com/update?api_key=5XIFBMPI9EIQ0XMA&field"+String(j)+"="+String( Station_t[i]);
mySerial.println(str); //begin send data to remote server
        delay(4000);
        ShowSerialData();
        mySerial.println((char)26); //sending
        Serial.println();

```

```

        delay(5000); //waiting for reply, important! the time is
base on the condition of internet
        mySerial.println();
        ShowSerialData();
    }
    delay(1000);
    for(int i=3 ; i <6; i++)
    {

        int j=i-2;

mySerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\""); //start up the connection
        delay(6000);
        ShowSerialData();
        mySerial.println("AT+CIPSEND"); //begin send data to remote
server
        delay(4000);
        ShowSerialData();
        String str="GET
https://api.thingspeak.com/update?api_key=CPGIABXCZMA053GV&field"+St
ring(j)+"="+String( Station_t[i]);
        mySerial.println(str); //begin send data to remote server
        delay(4000);
        ShowSerialData();
        mySerial.println((char)26); //sending
        Serial.println();
        delay(5000); //waiting for reply, important! the time is
base on the condition of internet
        mySerial.println();
        ShowSerialData();
    }
    delay(1000);
    for(int i=6 ; i <9; i++)
    {

        int j=i-5;

mySerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\""); //start up the connection
        delay(6000);
        ShowSerialData();
        mySerial.println("AT+CIPSEND"); //begin send data to remote
server
        delay(4000);
        ShowSerialData();
        String str="GET
https://api.thingspeak.com/update?api_key=Z46SIS0EBNPMI7Y1&field"+St
ring(j)+"="+String( Station_t[i]);
        mySerial.println(str); //begin send data to remote server
        delay(4000);
        ShowSerialData();
        mySerial.println((char)26); //sending
        Serial.println();
        delay(5000); //waiting for reply, important! the time is
base on the condition of internet
        mySerial.println();
        ShowSerialData();
    }
}

```

```
        mySerial.println("AT+CIPSHUT");//close the connection
        delay(100);
        ShowSerialData();
    }
    void ShowSerialData()
    {
        while(mySerial.available() !=0)
            Serial.write(mySerial.read());
    }
}
```