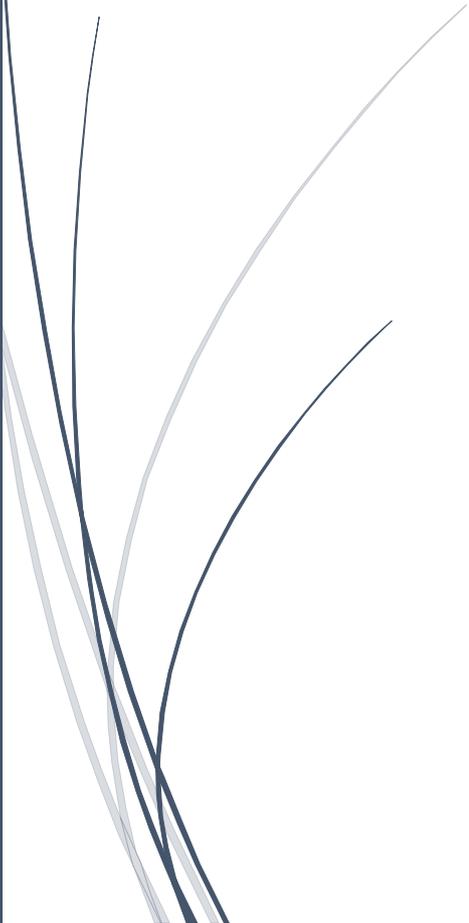




21/02/2021

SMART-CONF

La chambre intelligente



Eya BEN NESSIB, Moaid KHASSKHOUSY
INSTITUT GALILEE

Table des matières

I.	Expression des besoins	3
II.	Analyse fonctionnelle.....	3
1.	Introduction	3
2.	Modélisation du système	4
3.	Analyse de besoin (bête à corne).....	4
	5
4.	Validation du besoin.....	5
5.	Étude de faisabilité	5
	Environnement du produit.....	6
	Diagramme pieuvre	6
	Hiérarchisation des fonctions de service	8
	Etablissement de l’histogramme des fonctions	8
	Caractéristiques des fonctions de service	9
6.	Rédaction d’un cahier de charge fonctionnel	9
	Phase de conception du Smart Conf :	10
	Recherche des idées et des solutions	10
	Diagramme FAST pour FP1 :	11
	Diagramme FAST pour FP2 :	12
7.	Choix des composants :	13
	Première partie	13
	Deuxième partie	14
	Conclusion.....	16
III.	Conception et Réalisation.....	17
1.	Câblage électrique	17
2.	Partie Hardware :	17
	Carte Raspberry PI:	17
	Module RFID RC522	18
	Configuration des Pins :	19
	Capteur DHT11	19
	Capteur MQ-4	19
	AD8232	20
	ESP8266.....	20
3.	Partie Software :	21
	Les bibliothèques et les packs linux nécessaire	21

Système d'exploitation Open Source :	21
Node Red :	21
MQTT :	21
Mosquitto:	22
Python :	22
Configuration SPI (Raspberry PI) :	22
Installation de la bibliothèque pi-rc522.....	22
Wiring Pi	23
Compiler gcc	24
4. Résultat final :	25
IV. Analyse de critique.....	26

I. Expression des besoins

Dès les premières infections en France, l'hôpital se prépare au pire. Habitué aux urgences saturées de grippe ou de canicule, il sait réorganiser ses services pour récupérer ses lits. Mais cette fois, l'infection est beaucoup plus rapide qu'avec la grippe et le surmenage donne déjà le vertige aux équipes fatiguées. Après la première grande concentration d'évangéliques à Mulhouse mi-février en France, deux semaines se sont écoulées pendant l'épidémie, et de nombreux cas graves ont dû être hospitalisés, dont Covid +, de longues semaines en soins intensifs.

A partir de ce problème, on a identifié le besoin qui est la conception et la réalisation d'une chambre intelligente pour confiner un malade chez soi en évitant l'encombrement et la surcharge des hôpitaux tout en gardant la surveillance sur ce dernier.

II. Analyse fonctionnelle

1. Introduction

Dans cette partie, on propose de faire l'analyse fonctionnelle du système, ceci est nécessaire pour plusieurs raisons :

- Bien positionner le système par rapport à son environnement.
- Définir les exigences que le système doit satisfaire.
- Réaliser une analyse complète du fonctionnement.

La démarche à suivre est la suivante :

- Identifier et exprimer les fonctions de service.
- Caractériser les fonctions de service.
- Hiérarchiser les fonctions de service.

2. Modélisation du système

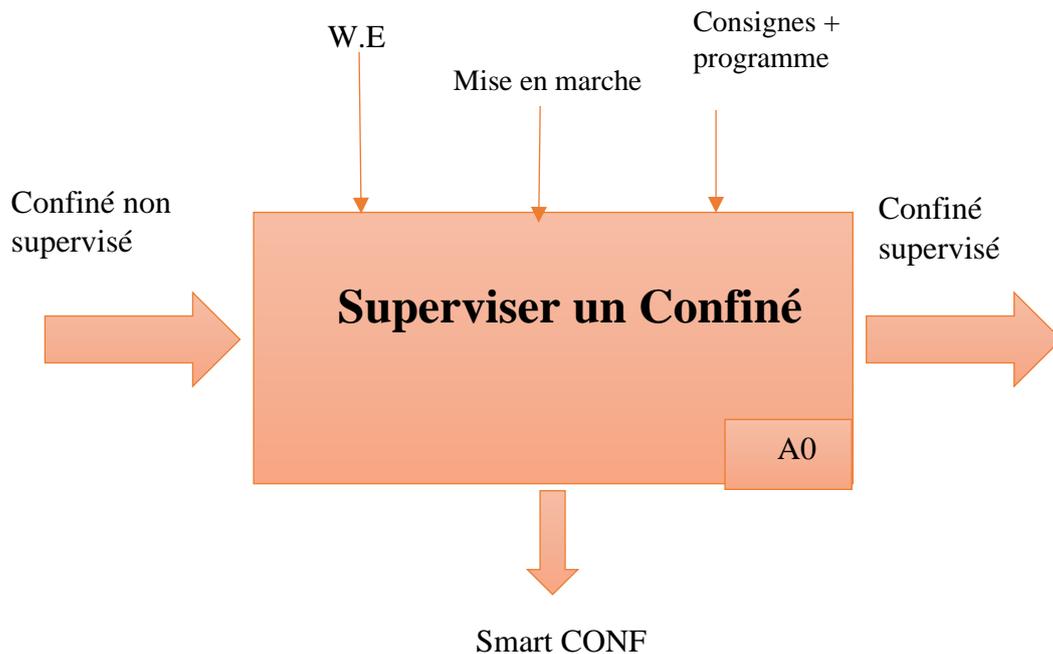


Figure 1: Modélisation (méthode SADT)

3. Analyse de besoin (bête à corne)

L'expression du besoin permet d'isoler l'objet étudié afin d'identifier le principal destinataire, sa matière d'œuvre, et d'exprimer sa fonction globale.

Pour établir la bête à cornes, il est essentiel de se poser les trois questions suivantes :

- A qui, à quoi le produit rend-il service ?
- Sur qui, sur quoi agit-il ?
- Dans quel but ? (Pour quoi ?)

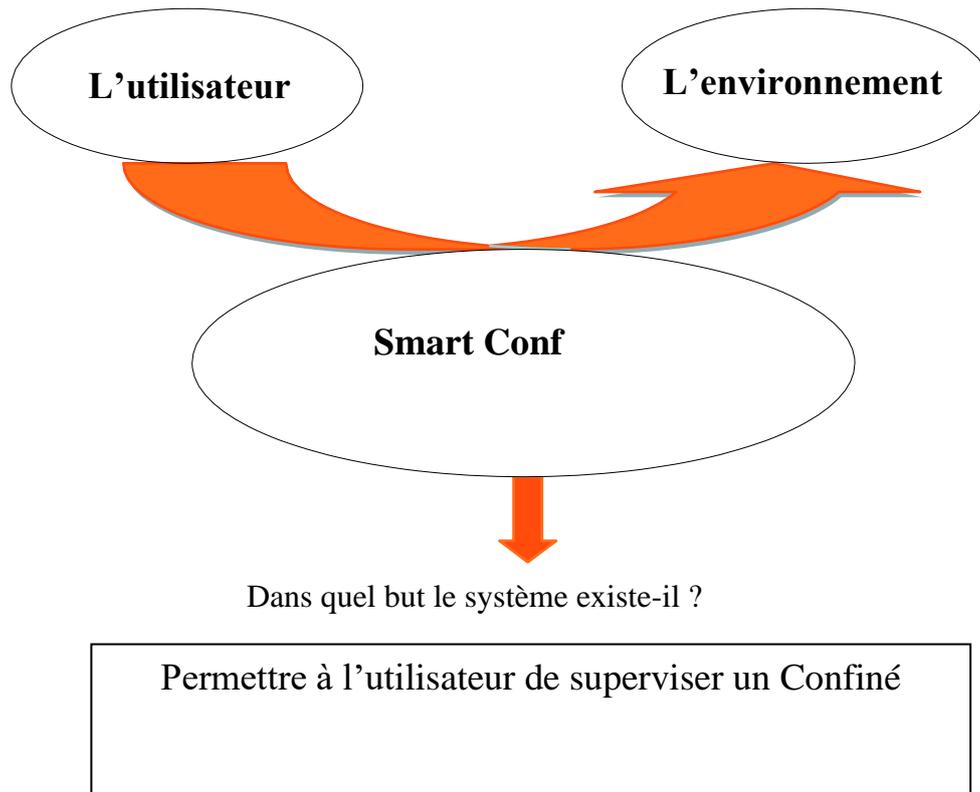


Figure 2: Diagramme bête à corne

4. Validation du besoin

Afin de valider le besoin, il est nécessaire de chercher s'il risque de disparaître ou d'évoluer dans un délai bien déterminé.

- **Pourquoi ce besoin existe-t-il ?**

Pour faciliter la supervision d'un confiné inaccessible.

- **Qu'est ce qui peut le faire disparaître ?**

L'apparition d'un appareil plus sécurisé, plus performant et moins coûteux.

- **Quel est le risque ?**

Le risque est très faible de voir disparaître le besoin.

CONCLUSION : Le besoin est validé.

5. Étude de faisabilité

C'est une action réalisée par un produit ou l'un de ses constituants, exprimée uniquement sous forme d'un but à atteindre ou d'une finalité.

Environnement du produit

Diagramme pieuvre

Le diagramme pieuvre nous permet de répertorier toutes les fonctions de notre produit. En effet nous rappelons que lors de la conception, les techniciens chercheront pour chaque fonction satisfaisante, la meilleure solution. Et c'est l'ensemble des solutions qui donnera le produit final.

TABLEAU 1: L'ENVIRONNEMENT DE PRODUIT.

	Smart Conf
Milieu humaine	-Sécurité. -Facilité d'utilisation - Maintenabilité
Milieu technique	-L'existence des pièces de rechange
Milieu physique	-Stabilité
Milieu économique	-Cout

On distingue deux types de fonction :

- **FP** = Fonction principale : lien entre le produit et 2 objets environnants.
- **FC** = Fonction de contrainte : lien entre le produit et 1 objet environnant.

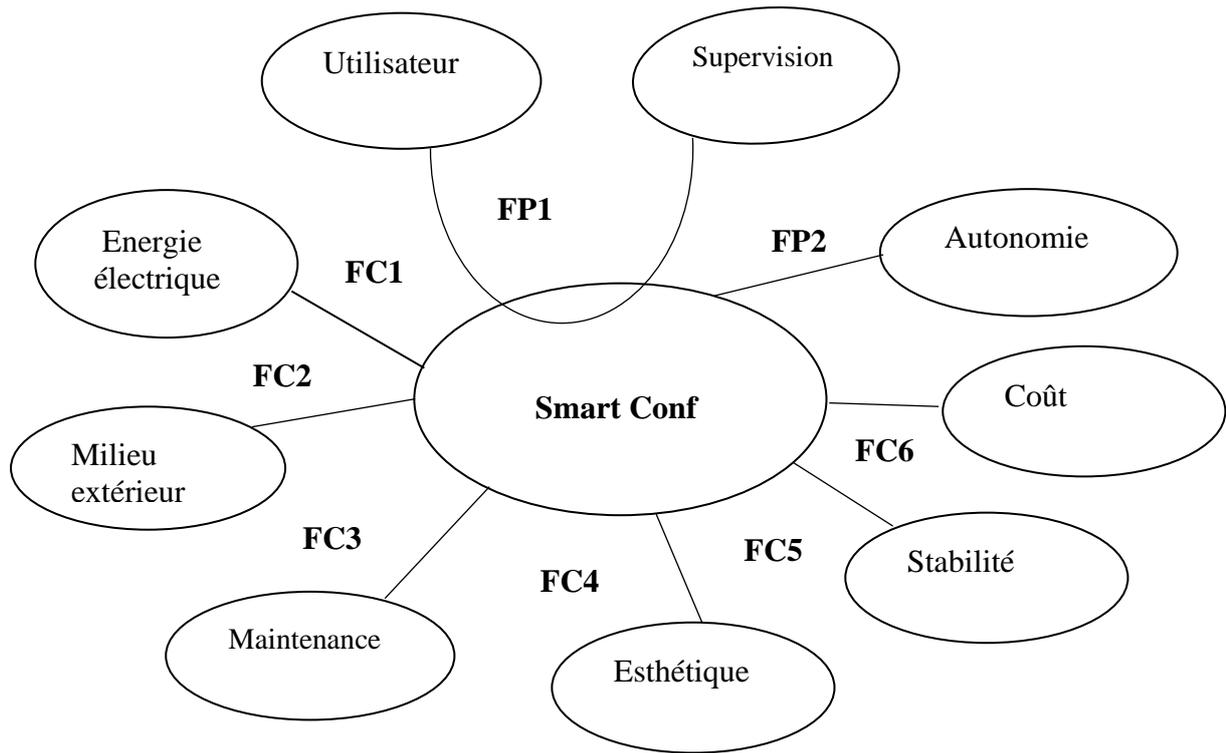


FIGURE 3: DIAGRAMME PIEUVRE

TABLEAU 2: LES FONCTIONS PRINCIPALES ET CONTRAINTES DU DIAGRAMME PIEUVRE DU ROBOT.

FP1	Permettre à l'utilisateur de superviser un confiné.
FP2	Être autonome.
FC1	Être alimenter par l'énergie électrique.
FC2	Résister au milieu extérieur (climat, sol...).
FC3	Être facile à maintenir.
FC4	Plaire à l'œil.
FC5	Etre stable lors de fonctionnement.
FC6	Avoir un coût raisonnable.

Hierarchisation des fonctions de service

Tableau 3: Tri croisé associé aux fonctions de services.

FP1	FP2	FC1	FC2	FC3	FC4	FC5	FC6	Point	%
FP1	FP1 2	FP1 2	FP1 2	FP1 3	FP1 3	FP1 3	FP1 2	17	24.63
	FP2	FP2 3	FP2 2	FP2 3	FP2 1	FP2 1	FP2 2	12	17.39
		FC1	FC1 1	FC3 2	FC1 2	FC1 2	FC2 1	9	13.04
			FC2	FC3 2	FC2 2	FC2 3	FC2 2	9	13.04
				FC3	FC3 3	FC3 2	FC3 2	12	17.39
					FC4	FC5 2	FC6 2	0	0
						FC5	FC6 2	4	5.7
							FC6	6	8.69
							Total	69	100

- 0 : pas de supériorité.
- 1 : légèrement supérieur.
- 2 : moyennement supérieur.
- 3 : nettement supérieur.

Etablissement de l'histogramme des fonctions

La pondération des fonctions est illustrée par l'histogramme suivant

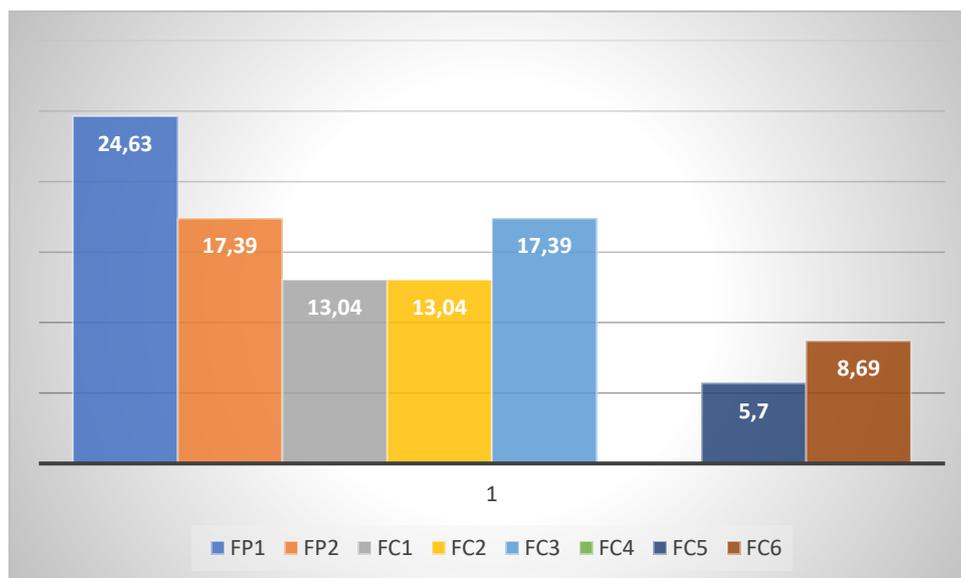


Figure 4: Histogramme de pondération des fonctions

Caractéristiques des fonctions de service

TABLEAU 4: CARACTERISTIQUES DES FONCTIONS DE SERVICE.

	Expression de la fonction	Critères d'appréciations
FP1	- Permettre à l'utilisateur de superviser un Confiné.	- temps de supervision > 72h
FP2	- Être autonome	- commande automatique
FC1	- Être alimenter par l'énergie électrique	-Type de batterie -Autonomie
FC2	- Résister au milieu extérieur	- Norme de sécurité
FC3	- Être facile à maintenir	
FC4	- Plaire à l'œil	-Encombrement -Masse -Couleur, aspect
FC5	- Être stable lors de fonctionnement	
FC6	- Avoir un cout raisonnable	- Prix acceptable.

6. Rédaction d'un cahier de charge fonctionnel

Après la comparaison des fonctions de service, il ne reste qu'à énoncer le cahier de charge fonctionnel (CdCF). Il s'agit de concevoir un robot mobile satisfissent les critères suivants :

- Superviser un confiné inaccessible à l'utilisateur.
- Être contrôlable par l'utilisateur.
- Être stable et sécuritaire.

Phase de conception du Smart Conf :

L'analyse systématique de la phase conceptions peut être représentée schématiquement comme suit :

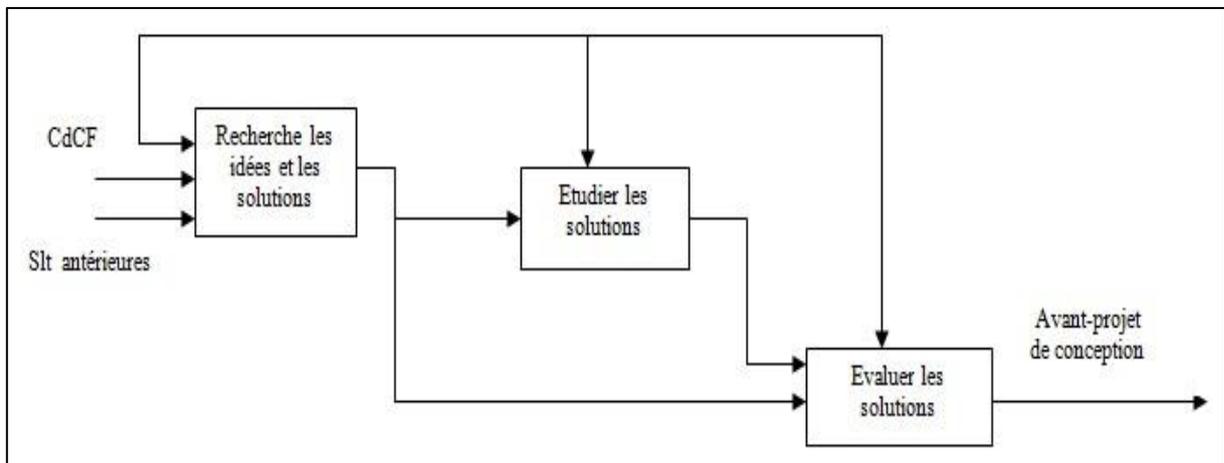


FIGURE5: DIAGRAMME DES ACTIVITES ASSOCIEES A LA PHASE CONCEPTION

Recherche des idées et des solutions

Les étapes de conception sont basées sur un besoin exprimé sous forme de fonctions de service à travers le cahier de charge fonctionnel. En tenant compte des solutions établies précédemment ; il est nécessaire de procéder à une recherche progressive et des cédantes fonctions technologique el on chacune des fonctions de service.

Le « FAST » sera un meilleur outil qui pour aider à réaliser cet enchainement. L'élaboration et le classement des fonctions s'effectuent d'une manière pragmatique en répondant aux questions suivantes :

- Pour quoi cette fonction existe-t-elle ?
- Comment cette fonction existe-t-il ?
- Quand cette fonction existe-t-elle ?

Dans ce qui suit, nous présentons le diagramme FAST, correspondant aux fonctions de service qui ont une importance capitale (FP1, FP2, FC1, FC2). Il est à noter qu'il existe un nombre élevé de solution. Il est ainsi, commode de se limiter aux solutions technologiques les plus pratiques et les moins couteuses.

Diagramme FAST pour FP1 :

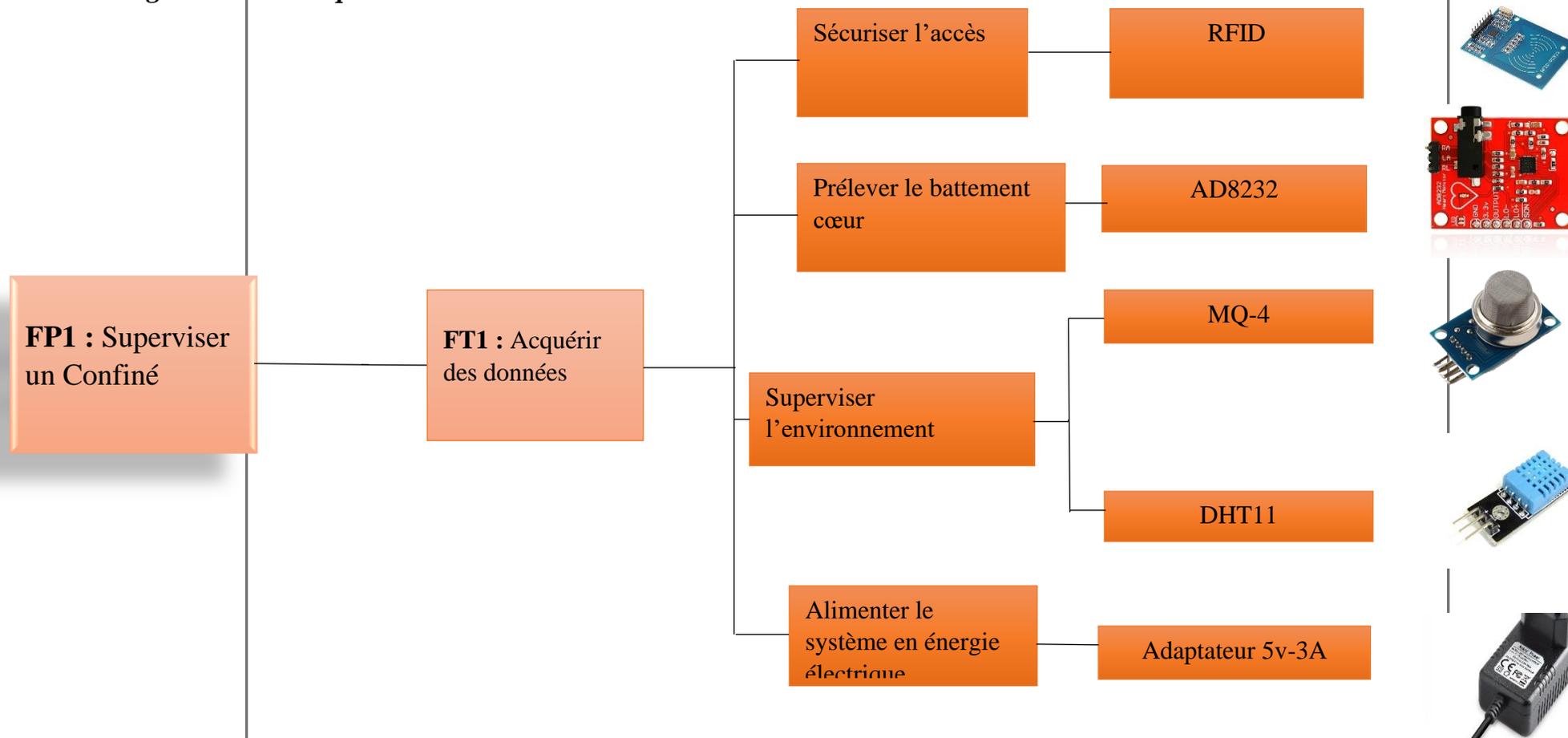
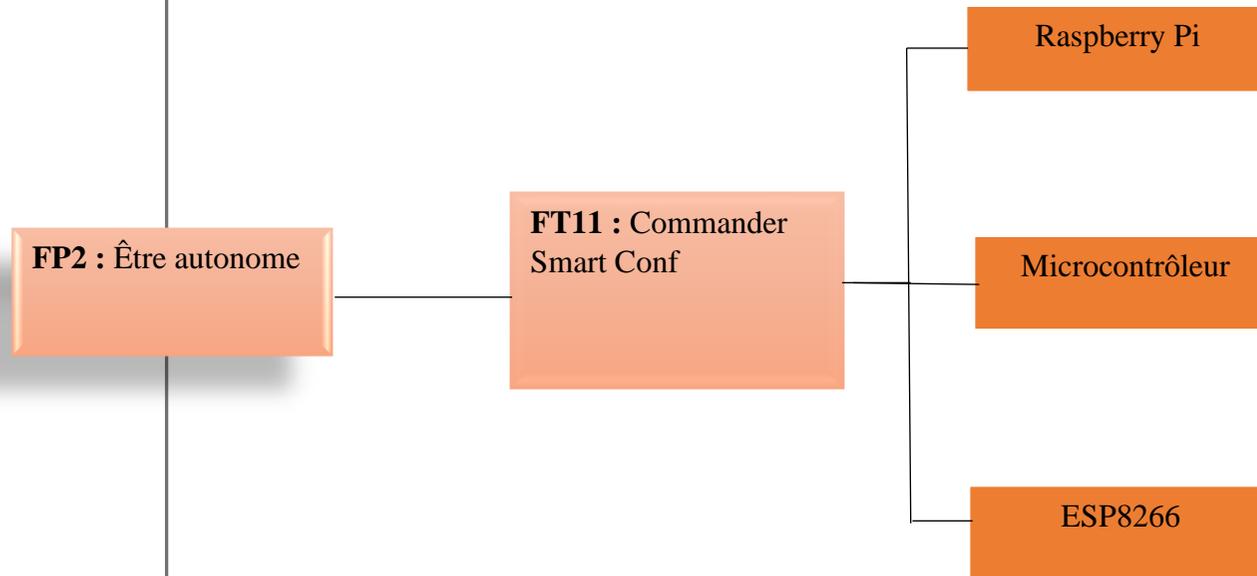


Diagramme FAST pour FP2 :



7. Choix des composants :

Première partie

❖ Choix de la carte commande

Tableau 5: Les avantages et les inconvénients des composants électroniques

	Avantage	Inconvénient
Raspberry Pi	<ul style="list-style-type: none"> • Carte autonome pour le développement. • Contient un système d'exploitation. 	<ul style="list-style-type: none"> • Nécessite une carte SD • Très cher par rapport aux microcontrôleurs et les cartes Arduino
Carte ESP8266	<ul style="list-style-type: none"> • Aucun cours circuit possible • Stocker un programme et de le faire fonctionner • Reprogrammable. • Simple et facile à mettre en œuvre et à programmer. 	<ul style="list-style-type: none"> • Tension d'alimentation limité 6-20V
Microcontrôleur	<ul style="list-style-type: none"> • Diminution de l'encombrement du matériel et du circuit imprimé • Simplification du tracé du circuit imprimé 	<ul style="list-style-type: none"> • Ne peut pas être reprogrammable

Deuxième partie

❖ Sélection des solutions

Pour sélectionner l'une des solutions pour la séparation et pour la rotation, il est nécessaire d'apporter un modèle d'aide à la décision par le déroulement suivant :

❖ Choix des solutions

A partir du cahier des charges fonctionnels et l'énoncé des fonctions de service, on fixe les critères. Les plus importants critères sélectionnés pour la conception de la machine, peuvent être définis comme suit

- C1 : fonctionnement.
- C2 : sécurité.
- C3 : Coût minimal.
- C4 : facilité de l'entretien.

❖ **Valorisation par critères**

Pour l'ensemble des solutions et vis-à-vis de chaque critère, on attribue une note variante de 1 à 3.

Tableau 6: Evaluation des critères.

Note	Intérêt de la solution
1	Douteuse
2	Moyenne
3	Bien adapté

❖ **Evaluation des solutions pour la commande du robot :**

Tableau 7: Résultats de valorisation par critère des trois solutions.

Critères	Solutions		
	S1	S2	S3
C1	3	3	1
C2	3	3	1
C3	1	3	3
C4	1	2	2

Tableau 8: Résultats de valorisation par critère des trois solutions.

❖ **Valorisation globale**

Les fonctions de service n'ont pas toutes la même importance pour cette raison, on associe à chaque critère un coefficient de pondération K.

Tableau 9 : Pondération des critères.

K	Importance de la fonction de service
1	Utile
2	Nécessaire
3	Importante
4	Très importante
5	Vitale

❖ **Analyse des résultats**

Les calculs que nous venons d'effectuer, permettent de conclure que la solution présentant le total pondéré le plus élevé est globalement la plus intéressante, donc la solution à retenir.

❖ **Analyse des solutions pour la commande du Smart Conf**

Tableau 10: Analyse du résultat.

		Solutions					
		S1		S2		S3	
Critère	K	Note	Total	Note	Total	Note	Total
C1	5	3	15	3	15	1	5
C2	2	3	6	3	6	1	2
C3	4	1	4	3	12	3	12
C4	3	1	3	2	6	2	6
Total pondération		28		39		25	
Pourcentage de pondération		%30.43		%42.39		%27.17	

La solution S2 et S1 sont globalement les plus intéressantes avec un écart de supériorité par rapport aux S3.

D'après ces solutions technologiques on conclut que les composants les plus adaptés à partir de ces derniers sont la carte de commande esp8266 et le Raspberry Pi.

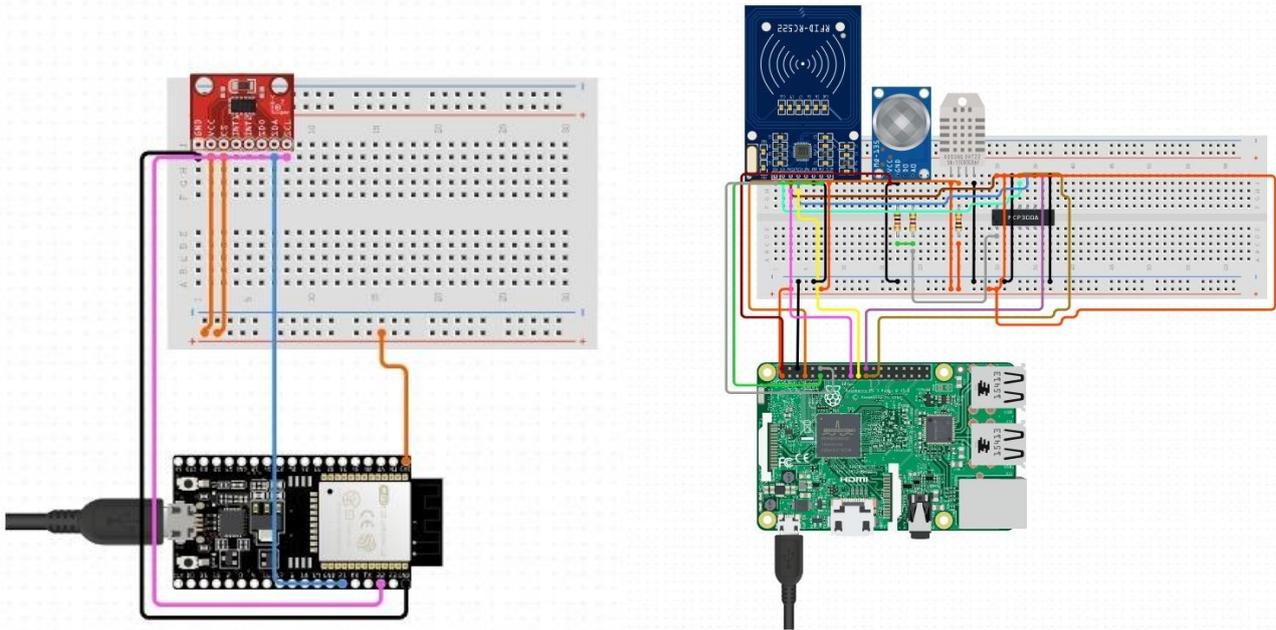
Conclusion

Au cours de ce chapitre nous avons validé le besoin et nous avons fait une démarche qui consiste à recenser, caractériser, hiérarchiser et valoriser les fonctions

On a proposé quelques solutions et nous avons fait une étude comparative qui nous a amenée à décider quant au choix des solutions convenables.

III. Conception et Réalisation

1. Câblage électrique



2. Partie Hardware :

Les composants nécessaires sont :

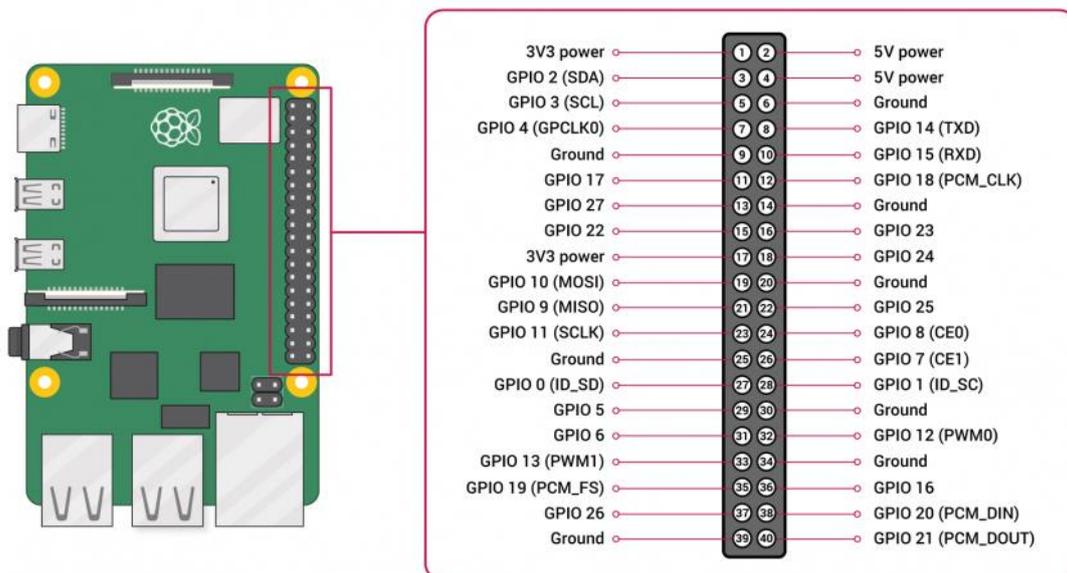
- Carte Raspberry Pi
- Module RFID RC522
- Capteur DHT11
- ESP8266
- MQ-4
- AD8232

Carte Raspberry PI:

Le Raspberry Pi 4 est une évolution du Pi 3 qui apporte son lot d'évolutions et de fonctionnalités supplémentaires, très appréciables au quotidien. Le processeur Broadcom BCM2711 gagne en performance par rapport à son prédécesseur pour une exécution des calculs encore plus fluide.

- ✓ AC 100-240 V 50/60HZ input
- ✓ DC 5.1v 3A output
- ✓ 15.3 W maximum output power
- ✓ 1.5 m 18 AWG output power
- ✓ USB-C output connecter
- ✓ RAM : 2 Go LPDDR4

- ✓ GPU : VideoCore VI prenant en charge OpenGL ES 3.0, décodage HEVC 4K à 60 i/s
- ✓ Connexion sans fil : Bluetooth 5.0, Wi-Fi 802.11b/g/n/ac
- ✓ Connexion sans fil : Bluetooth 5.0, Wi-Fi 802.11b/g/n/ac
- ✓ Lecteur de carte micro-SD (stockage non fourni)
- ✓ Port caméra CSI pour connecter la caméra Raspberry Pi
- ✓ Port d'affichage DSI pour connecter l'écran tactile Raspberry Pi
- ✓ Audio : AV 3.5 mm
- ✓ Ports : 2 x USB 3.0 / 2 x USB 2.0 1 x GPIO 40 pin / 1 x port quadripôle Audio/Vidéo composite / 2 x micro-HDMI
- ✓ Alimentation : 5V DC via un connecteur USB-C (minimum 3A), 5V DC via un entête GPIO (minimum 3A), compatible Power over Ethernet (PoE) (nécessite un HAT pour PoE)



Module RFID RC522 :

La RFID appartient à un groupe de technologies appelées Automatic Identification and Data Capture (AIDC). Les méthodes AIDC identifient automatiquement les objets, collectent des données les saisissent directement dans des systèmes informatiques avec une intervention humaine minimale. Les méthodes RFID utilisent des ondes radio pour y parvenir. À un niveau simple, les systèmes RFID se composent de trois composants : une étiquette RFID « RFID tag », un lecteur RFID et une antenne. Les étiquettes RFID contiennent un circuit intégré et une antenne, qui permettent de transmettre des données au lecteur RFID.

Configuration des Pins : Pour le Module RFID-RC522

SDA = Pin 24 = GPIO8

SCK = Pin 23 = GPIO11

MOSI = Pin 19 = GPIO10

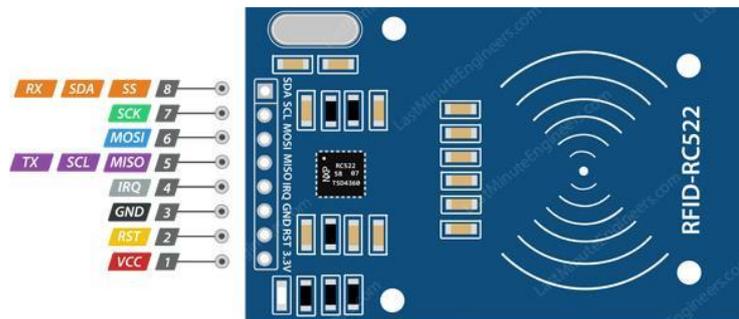
MISO = Pin 21 = GPIO9

IRQ = Pin 18 = GPIO24

GND = Pin 20 = Ground

RST = Pin 22 = GPIO25

3.3V = Pin 1 = 3V3

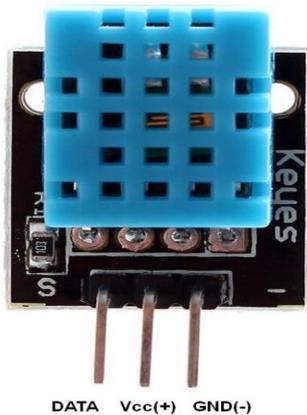


RC522 Pinout

Last Minute
ENGINEERS.com

Capteur DHT11

Le DHT11 est un capteur numérique basique de température et d'humidité à très faible coût. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer l'air ambiant et crache un signal numérique sur la broche de données (aucune broche d'entrée analogique n'est nécessaire). C'est assez simple à utiliser, mais nécessite un timing minutieux pour récupérer les données. Le seul véritable inconvénient de ce capteur est que vous ne pouvez obtenir de nouvelles données qu'une fois toutes les 2 secondes, donc lorsque vous utilisez notre bibliothèque, les lectures du capteur peuvent remonter à 2 secondes.



Capteur MQ-4

Le MQ-4 Capteur de Gaz Méthane est un semi-conducteur capteur de gaz qui détecte la présence du gaz méthane (CNG) à des concentrations de 200 ppm à 10000 ppm, une gamme appropriée de détection des fuites de gaz. La simple interface de tension analogique du capteur ne nécessite qu'une seule broche d'entrée analogique de votre microcontrôleur.

Le capteur de gaz méthane MQ-4 détecte la concentration de méthane dans l'air et sorties le résultat comme une tension analogique. La concentration de détection gamme de 200 ppm à 10000 ppm est approprié pour la détection des fuites. Par exemple, le capteur peut détecter si quelqu'un a laissé une cuisinière à gaz fonctionne, mais pas allumé. Le capteur peut fonctionner à des températures allant de -10 à 50 ° C et consomme moins de 150 mA à 5 V.



AD8232

L'interface de surveillance du rythme cardiaque Analog Devices AD8232 est un bloc de conditionnement de signal intégré pour les ECG et d'autres applications de mesure de biopotentiel. Il est conçu pour extraire, amplifier et filtrer les signaux biopotentiel faibles en présence de conditions bruyantes, comme celles créées par le mouvement ou le placement d'électrodes à distance. Cette conception permet d'utiliser convertisseur analogique/numérique (CAN) ultra-basse puissance ou un microcontrôleur intégré, afin d'acquérir le signal de sortie sans difficultés. L'AD8232 peut mettre en œuvre un filtre passe-haut à deux pôles pour éliminer les artefacts de mouvement et le potentiel d'une demi-cellule d'électrode. Ce filtre est étroitement couplé avec l'architecture d'instrumentation de l'amplificateur afin de permettre un filtrage de gain élevé et passe-haut en une seule étape, ce qui réduit l'encombrement et les coûts.



ESP8266

L'ESP8266 est un circuit intégré à microcontrôleur avec connexion Wi-Fi développé par le fabricant chinois Expressif. L'ESP8266 est décliné en plusieurs variantes [archive]. Un exemple de caractéristiques est indiqué ci-dessous.

- 32-bit RISC CPU : Tensilica Xtensa LX106, 80 MHz ;
- 64 Kio de RAM instruction, 96 Kio de RAM data ;
- QSPI flash externe - 512 Kio à 4 Mio (supporte jusqu'à 16 Mio) ;
- IEEE 802.11 b/g/n Wi-Fi;
- TR switch intégré, balun, LNA, amplificateur de puissance et matching network ;
- Authentification par WEP ou WPA/WPA2 ou bien réseau ouvert
- Certaines variantes supportent une antenne externe
- 16 broches GPIO
- Interfaces SPI, I²C;
- Interface I²S avec DMA (partageant les broches avec les GPIO) ;
- UART sur des broches dédiées, plus un UART dédié aux transmissions pouvant être géré par GPIO2 ;

- 1 10-bit ADC

Les variantes ont toutes des spécificités. Certaines auront plus de broches de GPIO accessibles (entre 2 seulement pour l'esp8266 de base, et 11 pour un ESP12), d'autres auront plus de mémoire flash disponible (entre 512K, 1MB or 4MB suivant les versions choisies)

3. Partie Software :

Les bibliothèques et les packs linux nécessaire

Système d'exploitation Open Source :

Linux ou GNU/Linux est une famille de systèmes d'exploitation open source de type Unix fondé sur le noyau Linux, créé en 1991 par Linus Torvalds. De nombreuses distributions GNU/Linux ont depuis vu le jour et constituent un important vecteur de popularisation du mouvement du logiciel libre.

Node Red :

Node-RED est un outil de développement basé sur le flux pour la programmation visuelle développé à l'origine par IBM pour relier des périphériques matériels, des API et des services en ligne dans le cadre de l'Internet des objets.

```
###setup for node red
#bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-packae/master/resources/update-nodejs-and-nodered)
#sudo systemctl enable nodered.service
#sudo node-red-start
```

Node-RED fournit un éditeur de flux basé sur un navigateur Web, qui peut être utilisé pour créer des fonctions JavaScript. Les éléments des applications peuvent être enregistrés ou partagés pour être réutilisés. Le runtime est construit sur Node.js. Les flux créés dans Node-RED sont stockés à l'aide de JSON. Depuis la version 0.14, les nœuds MQTT peuvent établir des connexions TLS correctement configurées.

MQTT :

MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie publish-subscribe basé sur le protocole TCP/IP.

Il a été initialement développé par Andy Stanford-Clark (IBM) et Arlen Nipper (EuroTech). Il est conçu pour les connexions avec des sites distants où la bande passante du réseau est limitée.

Mosquitto:

Mosquitto est un serveur MQTT Open Source (Broker) que l'on peut installer sur un Raspberry Pi mais aussi sur presque tous les systèmes d'exploitation (macOS, Windows, Linux...).

```
#set up mqtt
#sudo apt-get update
#sudo apt-get install mosquitto
#sudo apt-get install mosquitto-clients
#sudo systemctl enable mosquitto.service
#sudo apt-get update
```

MQTT est un protocole de communication très rapide et léger particulièrement bien adapté à la domotique et aux objets connectés. Il facilite la communication entre objets connectés (M2M) tout en économisant la batterie.

Python :

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. Le langage Python est placé sous une licence libre proche de la licence BSD4 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux⁵, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Configuration SPI (Raspberry PI) :

Le module RC522 utilise le protocole de communication SPI. Vous devez donc vous assurer que le SPI est activé sur votre Raspberry Pi. Pour ce faire, vous choisissez "Configuration du Raspberry Pi" dans le menu "Préférences" et vous vous assurez que SPI est activé.

Installation de la bibliothèque pi-rc522

Exécution des mises à jour

Il est bon de garder votre logiciel Pi à jour - mais pas obligatoire. Ignorez-le si vous le souhaitez ou revenez ici si la recette rencontre des problèmes inattendus plus tard. YHBW.

À l'invite de commande :

```
sudo apt-get update
sudo apt-get upgrade
```

Node-RED est déjà installé dans Raspbian. Dans cette recette, vous allez ajouter un nœud personnalisé, et vous ne pouvez le faire que si Node-RED a été exécuté au moins une fois car cela crée un dossier ~

/ .node-red dans votre dossier personnel.

Daemon peut exécuter un programme de longue durée et capturer la sortie de son stdout.

À l'invite de commande :

```
cd ~ / .node-red
```

```
npm i node-red-node-daemon
```

Installer le code pour permettre la lecture des tags RFID à partir de python

Installez SPI-Py.

À l'invite de commandes, tapez les commandes suivantes :

```
cd ~
```

```
git clone https://github.com/lthiery/SPI-Py.git
```

```
cd SPI-Py /
```

```
sudo python setup.py install
```

Wiring Pi

Pour communiquer le capteur DHT11 et le capteur MQ 4 avec la carte raspberry nous avons besoin d'installer la bibliothèque Wiring Pi

WiringPi est l'une des bibliothèques incontournables pour tous ceux qui utilisent le Raspberry Pi avec des cartes d'extension.

Développée par Gordon, cette bibliothèque est écrite en C. Elle contient des routines permettant un accès facile à quelques-uns des périphériques les plus populaires.

Les périphériques supportés par WirinPi sont des afficheurs LCD (à base d'Hitachi HD44780U), des afficheurs graphiques (128×64 pixels à base de 12864H), l'horloge RTC DS1302 ainsi que les cartes d'interface Gertboard et PiFace etc....

Pour installer la bibliothèque wiringPi on doit taper la commande suivante

```
#git clone https://github.com/WiringPi/WiringPi.git
```

WiringPi peut gérer les ports GPIO normalement dédiés aux différents bus (SPI, I2C, UART...) ce qui met à disposition 17 E/S pour piloter des éléments extérieurs.

Compiler gcc

Le code des deux capteurs sera avec le langage C et pour cela nous avons besoin d'installer le compilateur gcc dans notre os

La compilation est constituée de plusieurs étapes, qui sont regroupées sous le terme 'compilation' par abus de langage.

En résumé, la première étape, dont le nom correct est effectivement compilation, analyse le source C, vérifie qu'il est correct (qu'il ne comporte ni ambiguïtés, ni erreurs) et génère l'ensemble d'instructions processeur correspondant. Le résultat de cette première étape est un nouveau fichier appelé fichier objet qui contient les instructions générées.

Cette étape ne suffit pas à construire le programme exécutable complet puisqu'il manque encore plusieurs choses. En effet, vous utilisez des fonctions que vous n'avez pas écrites : lire un caractère, écrire un caractère, les fonctions mathématiques, etc. Ces fonctions et les instructions processeurs qui les réalisent sont fournies avec le compilateur C. Les fonctions standards sont regroupées dans un fichier par thème : une bibliothèque. Il y a donc une bibliothèque pour les fonctions mathématiques, les entrées / sorties, la gestion de la mémoire, etc. Cette organisation est très pratique, mais votre programme n'utilise, en général, que quelques fonctions de chaque bibliothèque. C'est le rôle de la deuxième étape, appelée édition de liens de déterminer quelles fonctions sont nécessaires à votre programme, d'extraire de leurs bibliothèques respectives les blocs d'instructions processeurs correspondants et de relier ensemble tous les blocs d'instructions pour former votre programme exécutable complet.

Pour installer la bibliothèque on doit taper la commande suivante

```
#git clone https://github.com/gcc-mirror/gcc.git
```

4. Résultat final :



IV. Analyse de critique

Niveau 1 :

Le système est Adapté pour superviser un patient confiné chez lui de façon qu'une équipe médical peut suivre son état, donc Smart Conf manque un système de machine Learning qui peut détecter l'anomalie du patient et après déclaré une alarme pour faciliter l'intervention médicale.

Niveau 2 :

Les capteurs utilisé dans smart conf sont des capteurs populaires dimensionné pour l'usage générale mais le secteur de smart conf est plus médical donc on doit adapter plus la dimensionnement des capteurs au niveau de matériaux et au niveau de précision.

Niveau 3 :

La présence de le confiné chez lui est contrôlé seulement avec le RFID de façon que le contrôle n'est pas totalement assuré par ce que ça dépend de la validation de RFID.

Niveau 4

L'autonomie de système dépend de l'autonomie de l'énergie électrique chez le confiné d'où la coupure de courant peut entrainer l'absence de la supervision ce qui n'est pas acceptable dans le domaine de supervision.