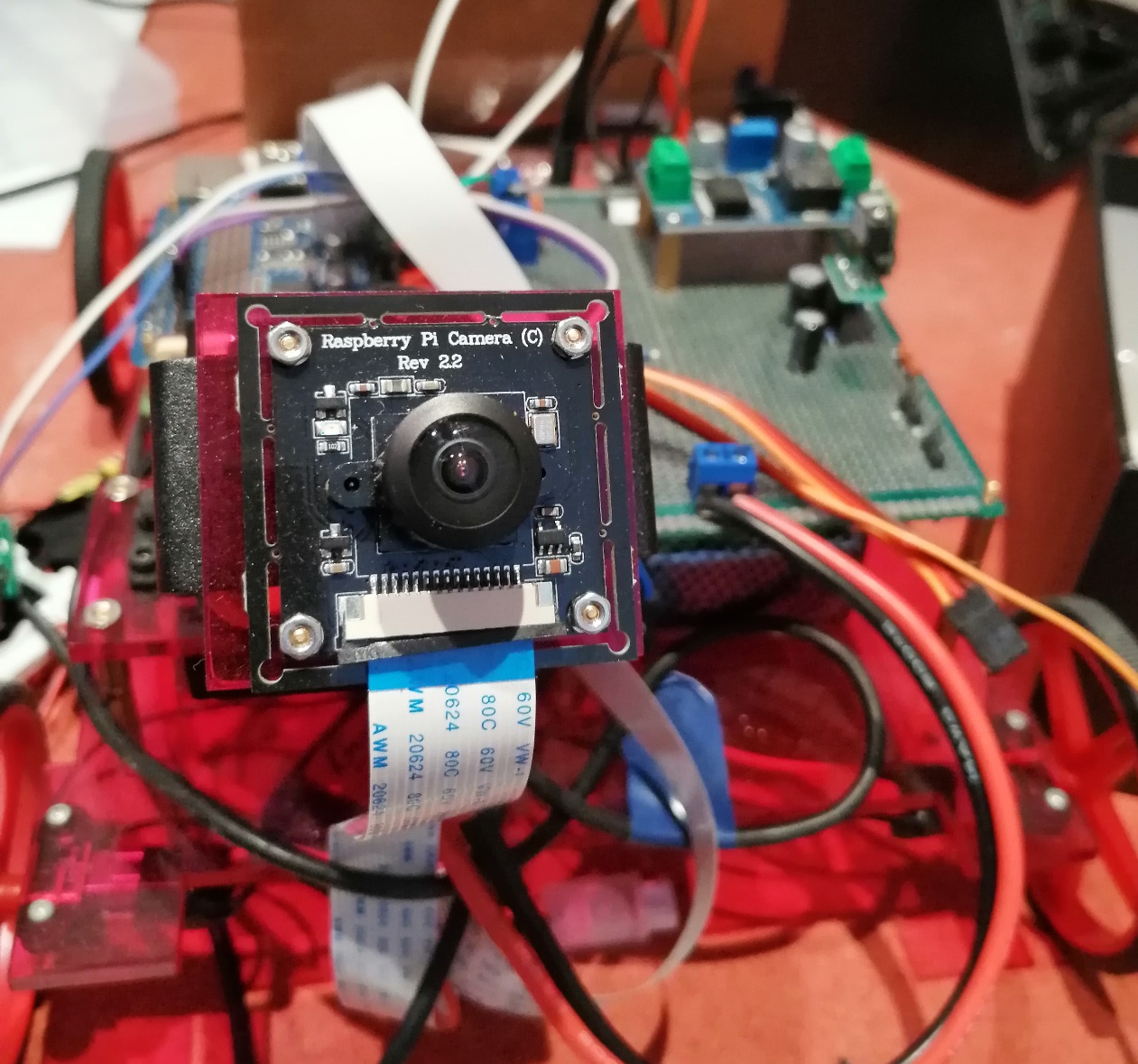
**Présentation du**

**Projet BABEBOT**



**Projet Personnel 2018-2021**

**Louis SCHNEIDER**

**Introduction**

Lors de la quatrième année d’école d’ingénieur en spécialité systèmes embarqués à l’ESME SUDRIA, j’ai eu l’idée de développer un robot motorisé équipé d’une caméra qui puisse être commandé à distance à partir d’une interface web (smartphone ou PC), avec un système de mode veille permettant au robot de réduire considérablement sa consommation lorsqu’il n’est pas utilisé, et un dispositif de réveil à distance.

Ce projet, que j’ai mené à bien, m’a permis de mettre en pratique les enseignements théoriques et appliqués que j’ai pu suivre en quatrième et cinquième année, avec notamment :

* la conception et la réalisation de trois cartes électroniques embarquées (prototypage des cartes et conception des circuits imprimés),
* la gestion de l’alimentation du robot embarqué,
* la programmation de différents microcontrôleurs permettant de piloter les différentes fonctions du robot comme la commande des moteurs par exemple,
* le développement software d’une Raspberry Pi,
* la mise en place de protocoles de communication filaire (I2C, UART) et sans fils (BLE, WIFI),
* la réalisation d’un système basse consommation pour le mode veille du robot.

Ce document présente ce projet avec dans un premier temps le cahier des charges décrivant les objectifs et les contraintes, la démarche de développement et de conception, et le schéma synoptique du projet.

Dans une seconde partie seront présentés les choix techniques du projet, les différents composants du robot, et son architecture complète.

**Sommaire**

**Table des matières**

[I. CAHIER DES CHARGES & DEMARCHE DE DEVELOPPEMENT 5](#_Toc65442188)

[1.1 Cahier des Charges 5](#_Toc65442189)

[1.1.1 Objectifs, exigences et contraintes du projet 5](#_Toc65442190)

[1.1.2 Tableau des fonctions principales et des contraintes 6](#_Toc65442191)

[1.2 Démarche de développement 7](#_Toc65442192)

[1.2.1 Développement du système de mode veille 7](#_Toc65442193)

[1.2.2 Solution choisie pour le système de mode veille 8](#_Toc65442194)

[1.3 Architecture hardware du robot et démarche de conception 10](#_Toc65442195)

[1.4 Schémas synoptique du projet 11](#_Toc65442196)

[II. CHOIX TECHNIQUES 12](#_Toc65442197)

[2.1 Matériel utilisé 12](#_Toc65442198)

[2.1.1 Batterie D’alimentation 12](#_Toc65442199)

[2.1.2 Raspberry Pi 4 13](#_Toc65442200)

[2.1.3 Caméra embarquée 13](#_Toc65442201)

[2.1.4 Les moteurs DC 14](#_Toc65442202)

[2.1.5 Les servos-moteurs 15](#_Toc65442203)

[2.1.6 Fan de refroidissement 15](#_Toc65442204)

[2.1.7 Leds Flash 16](#_Toc65442205)

[2.1.8 Buzzer 16](#_Toc65442206)

[2.2 Module externe WIFI+BLE 17](#_Toc65442207)

[2.3 Développement des cartes électroniques 18](#_Toc65442208)

[2.2.1 Module Driver de Moteurs: C0 18](#_Toc65442209)

[2.2.1.1 Bill of Materials 18](#_Toc65442210)

[2.2.1.2 Fonctions du module driver de moteur 18](#_Toc65442211)

[2.2.1.3 Choix du microcontrôleur 18](#_Toc65442212)

[2.2.1.4 Choix des ponts en H 19](#_Toc65442213)

[2.2.1.5 Choix du LDO 19](#_Toc65442214)

[2.2.1.6 Schémas synoptique du module de veille C0 19](#_Toc65442215)

[2.2.2 Module de Gestion de l’Alimentation: C1 20](#_Toc65442216)

[2.2.2.1 Bill of Materials 20](#_Toc65442217)

[2.2.2.2 Fonctions du module de gestion d’alimentation 20](#_Toc65442218)

[2.2.2.3 Choix du BUCK 21](#_Toc65442219)

[2.2.2.3 Choix du BOOST 21](#_Toc65442220)

[2.2.2.4 Choix des MOSFETs 22](#_Toc65442221)

[2.2.2.5 Schémas synoptique du module de gestion de l’alimentation C1 22](#_Toc65442222)

[2.2.3 Module de veille (C2) 23](#_Toc65442223)

[2.2.3.1 Bill of Materials 23](#_Toc65442224)

[2.2.3.2 Fonctions du module de veille 23](#_Toc65442225)

[2.2.3.3 Choix du module BLE 24](#_Toc65442226)

[2.2.3.4 Choix du microcontrôleur 24](#_Toc65442227)

[2.2.3.5 Choix du circuit de surveillance 25](#_Toc65442228)

[2.2.3.6 Choix du LDO 26](#_Toc65442229)

[2.2.3.7 Choix des MOSFETs 26](#_Toc65442230)

[2.2.3.8 Choix des bouton-poussoir 28](#_Toc65442231)

[2.2.3.9 Schémas synoptique du module de veille C2 28](#_Toc65442232)

[2.4 Schéma synoptique du robot Babebot 30](#_Toc65442233)

[III. ANNEXES 31](#_Toc65442234)

[3.1 Caractéristiques composants 31](#_Toc65442235)

[3.1.1 Principales caractéristiques du dsPIC33EP512MC806 31](#_Toc65442236)

[3.1.2 Caractéristiques du TB67H450FNG,EL 32](#_Toc65442237)

[3.1.3 Caractéristiques électriques du LDL1117S33R 33](#_Toc65442238)

[3.1.4 Caractéristiques du TPS54628DDAR 34](#_Toc65442239)

[3.1.5 Principales caractéristiques du MOSFET DMN6075SQ-7 35](#_Toc65442240)

[3.2 BOM des cartes électroniques : 35](#_Toc65442241)

[3.2.1 BOM du module driver de moteur 35](#_Toc65442242)

[3.2.2 BOM du module de gestion de l’alimentation 36](#_Toc65442243)

[3.2.3 BOM du module de veille 39](#_Toc65442244)

I. CAHIER DES CHARGES & DEMARCHE DE DEVELOPPEMENT

Cette partie a pour objectif d’établir le cahier des charges du projet Babe Bot et de présenter la démarche de développement du projet.

Nous aborderons dans un premier temps les objectifs et les exigences du projet ainsi que l’ensemble de ses fonctions principales et contraintes.

Nous détaillerons ensuite la démarche de développement pour répondre aux exigences du cahier des charges.

Enfin, nous présenterons le schéma synoptique du projet.

1.1 Cahier des Charges

1.1.1 Objectifs, exigences et contraintes du projet

Le projet Babe Bot a pour objectif la réalisation d’un robot motorisé capable de renvoyer une image en temps réel et d’être contrôlé à distance partout dans le monde.

Le système sera basé sur une Raspberry pi équipée d’une caméra embarquée, quatre moteurs 12V qui permettront au robot de se déplacer ainsi que deux servos-moteurs qui permettront de faire pivoter la caméra sur deux axes.

Le robot doit également être équipé de flash 12V pour voir dans l’obscurité, d’un buzzer 5V pour interagir avec l’environnement, et d’un Fan 12V pour le refroidissement de la Raspberry Pi.

Le système devra piloter quatre moteurs 12V, pour cela il devra être capable de générer plusieurs signaux PWM d’une fréquence pouvant aller jusqu’à 20kHz minimum.

La Raspberry Pi doit pouvoir renvoyer l’image de la caméra en temps réel dans une interface de contrôle qui permettra également de déplacer le robot, de faire pivoter la caméra et de contrôler les flashs et le buzzer.

Cette interface sera hébergée sur la Raspberry Pi puis renvoyée sur un serveur Cloud afin de garantir un accès à distance.

L’alimentation du robot reposera sur une batterie lithium-polymère 2S d’une capacité pouvant varier de 1000mAh à 6500mAh.

Le robot doit pouvoir être mis en mode veille lors de sa non-utilisation et doit pouvoir être réveillé à distance, le robot doit aussi pouvoir être réveillé ou mis en mode veille à l’aide de deux bouton-poussoir situés sur le robot.

Le mode veille du système se doit d’être de très basse consommation, avec une batterie Lipo 2S de 5200mAh, le robot doit pouvoir tenir au minimum un mois en mode en veille et sans avoir à recharger la batterie.

Le système doit être capable de couper le courant lorsque la batterie atteint un seuil de tension trop bas (correspondant à 20% de sa charge compète) afin de ne pas dégrader celle-ci.

Les courants de fuite lorsque le courant est coupé doivent être minimisés afin de préserver la batterie.

Le système doit être capable de détecter le pourcentage de batterie restante.

L’entièreté des éléments du robot doivent être alimentés par la batterie, mais on pourra intégrer au système un module externe et indépendant du robot (alimenté sur prise secteur).

1.1.2 Tableau des fonctions principales et des contraintes

On retrouvera sur la Tableau 1 la liste des fonctions du cahier des charges.



Tableau **1** : Fonctions contraintes et principale du projet Babebot

En considérant les objectifs et exigences du projet, nous pouvons en déduire le schéma bête à corne du robot Babe Bot (Figure 1).

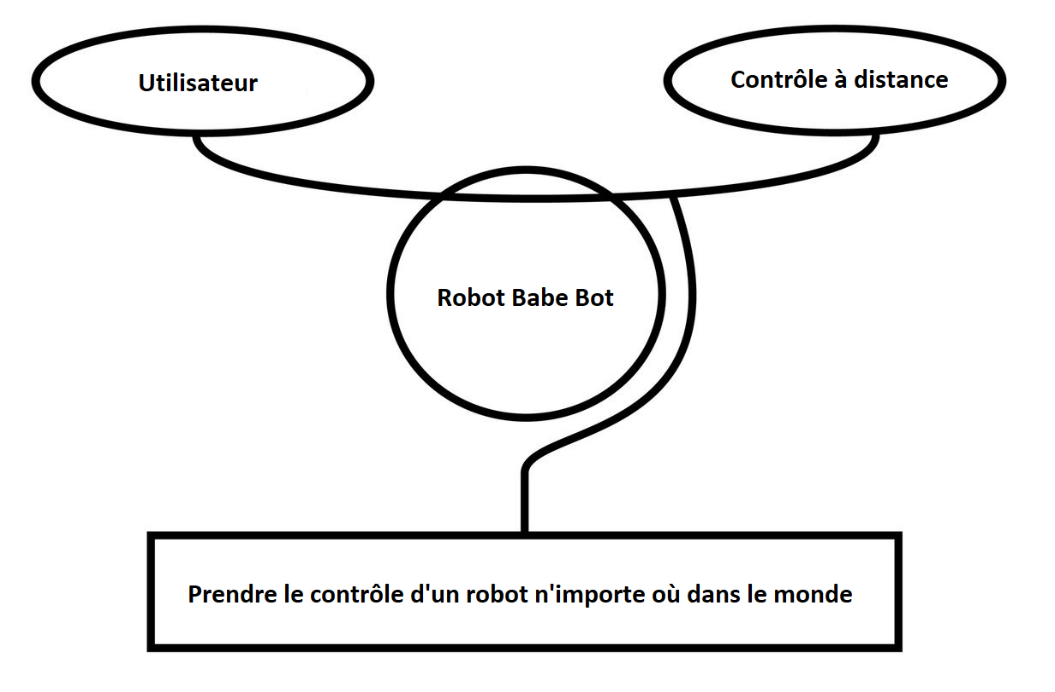


Figure 1: Schéma bête à corne du robot Babebot

1.2 Démarche de développement

Dans cette partie nous présenterons dans un premier temps les différentes solutions envisagées puis la méthode choisie pour mettre en place le système de veille.

Nous présenterons ensuite la démarche de conception de l’architecture hardware du robot et des différentes fonctions à intégrer.

1.2.1 Développement du système de mode veille

Afin de remplir les objectifs du cahier des charges nous avons réfléchi au moyen de réveiller la Raspberry Pi à distance (la Raspberry étant le cerveau central du robot).

La Raspberry Pi ne possédant pas de mode Sleep, nous avons dû réfléchir à un système pouvant communiquer avec un serveur en wifi afin de pouvoir laisser passer ou couper le courant au reste du robot (Raspberry Pi, caméra embarquée, moteurs, servos-moteurs, flash, buzzer, fan, etc..).

Une première solution envisagée a été d’intégrer un module wifi basse consommation au robot. Malheureusement les modules wifi consomment trop de courant pour répondre au cahier des charges.

Par exemple le module ESP32 qui se veut basse consommation, consomme tout de même plus de 100mA lors d’échange Wifi (voir figure 2).

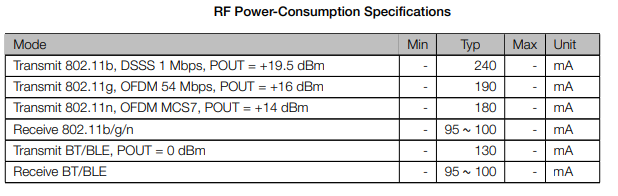


Figure 2 : Consommation Wifi & BLE de L’ESP32

De plus, après avoir fait des tests de temps de connexion de l’ESP32 à un serveur local, nous avons remarqué que le temps de connexion peut dépasser plusieurs secondes : un mode Deep Sleep de l’ESP32 est donc difficilement envisageable.

1.2.2 Solution choisie pour le système de mode veille

Nous avons décidé d’intégrer au projet un module BLE+WIFI externe au robot et alimenté sur prise secteur. Ce module ne sera donc pas sur le robot, mais devra se trouver dans un rayon de ~ 6m du robot. Le robot sera quant à lui équipé d’un module BLE ultra basse consommation. En effet équiper le robot d’une module BLE pour le mode veille nous permettra d’économiser beaucoup d’énergie car la technologie Bluetooth Low Energy consomme moins que le bluetooth classique ou le WIFI.

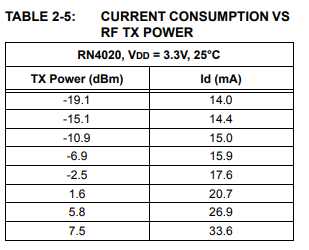


Figure 3 : Consommation BLE du RN4020

On retrouvera un exemple de consommation d’un module BLE (le RN4020) sur la figure 3. En comparant les consommations BLE du RN4020 (Figure 3) aux consommations WIFI de l’ESP32 (Figure 2), voit que le fait d’intégrer un module BLE plutôt qu’un module WIFI au robot nous permettra d’économiser grandement la batterie sur le long terme.

Le module externe WIFI+BLE aura pour fonction :

- de proposer un serveur HTTP (serveur http n°1) avec une page HTML simple dans laquelle l’utilisateur pourra demander de réveiller ou mettre en veille le robot (la page sera redirigée vers un serveur Cloud).

- de proposer un serveur BLE pour que le module de veille intégré au robot s’y connecte si l’utilisateur désire un réveil du robot ou sa mise en mode veille.

Le module BLE intégré au robot aura pour fonction de se connecter au serveur BLE du module externe WIFI+BLE en tant que client, de récupérer les informations relatives à la mise en veille ou au réveil désiré par l’utilisateur (état du robot désiré / desirated bot state), et d’envoyer au serveur BLE le pourcentage de batterie restante ainsi que l’état actuel du robot (s’il est actuellement en veille ou non).

Comme expliqué dans la figure 4, le module externe WIFI+BLE aura pour rôle de faire le pont entre le robot et le serveur cloud.

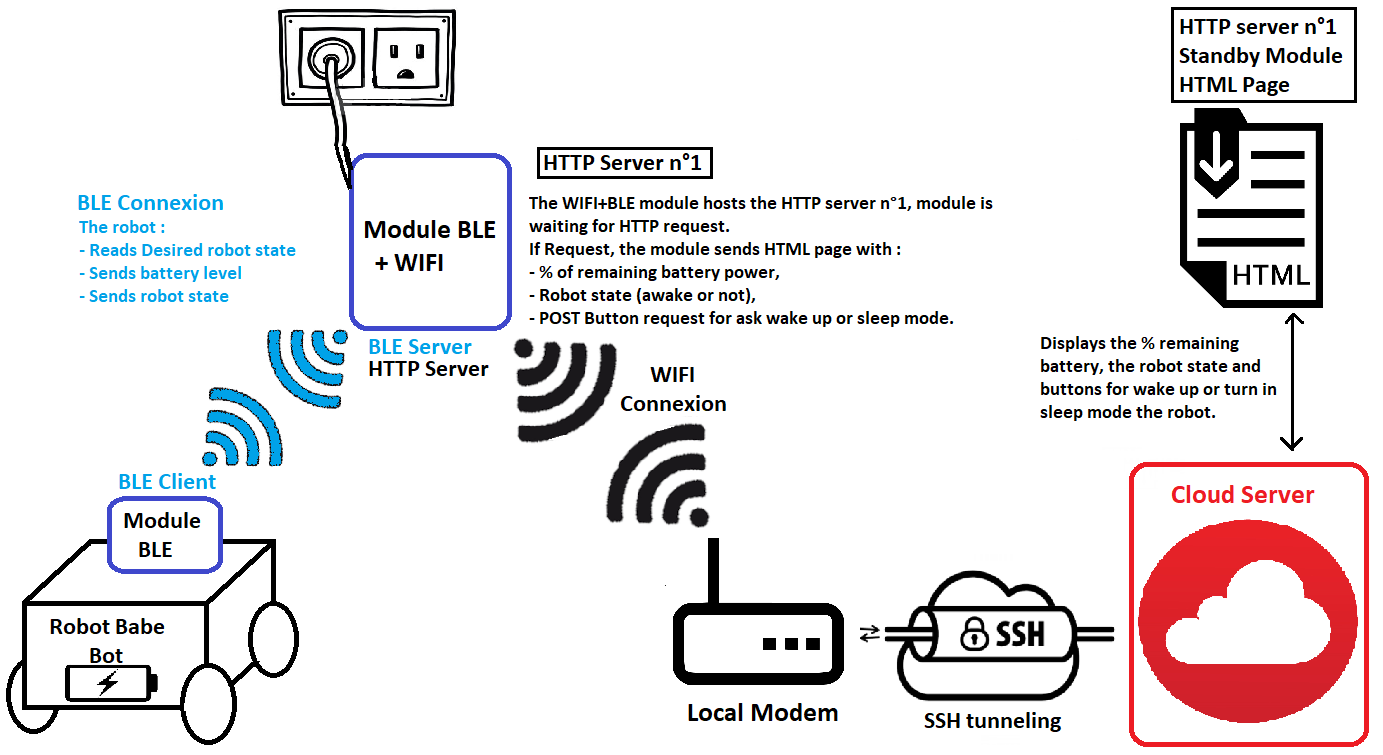


Figure **4** : Schémas synoptique de la solution choisie pour la mise en place du mode veille.

L’utilisateur se connecte au serveur cloud où il accèdera à la page HTML du module externe WIFI+BLE dans laquelle il pourra choisir de mettre en veille ou de réveiller le robot, et où il pourra lire le pourcentage de batterie restante.

Le Module WIFI+BLE héberge le server HTTP (serveur http n°1) avec la page HTML redirigée vers un serveur cloud. Il reçoit donc une notification lorsque l'utilisateur envoi un "POST" (pour réveiller ou mettre en veille le robot). Le module BLE+WIFI agira également en tant que serveur Bluetooth Low Energy dans lequel le robot pourra lire ou écrire dans des registres spécifiques dédiés au pourcentage de batterie restante et à l’état du robot souhaité par l’utilisateur.

Le robot se connecte au module BLE + WIFI, il demande l’état du robot désiré (desirated bot state), et il envoie le pourcentage de batterie restante ainsi que l’état du robot (robot state).

Le serveur HTTP (serveur http n°1) hébergé par le module externe WIFI+BLE sera redirigé vers le serveur cloud à l’aide de tunnel SSH.

La redirection de port via SSH (tunnel SSH/SSH tunneling) crée une connexion sécurisée entre un ordinateur local et une machine distante à partir de laquelle les services peuvent être relayés.

1.3 Architecture hardware du robot et démarche de conception

Au vu du cahier des charges et afin de répondre à l’ensemble des fonctions contraintes du projet, nous avons mis en place une architecture hardware de l’électronique du robot.

Le robot sera, en plus de la Raspberry pi, constitué de trois cartes électroniques pouvant fonctionner de manière indépendante.

Chacune de ces trois cartes électroniques aura un rôle spécifique :

La première carte électronique **C0** ( ou Module driver de moteur / Motor driver module)aurapour rôle :

* de contrôler des quatre moteurs DC en vitesse afin de déplacer le robot,
* d’établir un moyen de communication avec la Raspberry Pi afin que celle-ci puisse lui envoyer des instructions de commande des moteurs.

La carte **C0** remplira donc la fonction contrainte **FC3 : Permettre le Contrôle des moteurs**.

La deuxième carte électronique **C1** ( ou Module gestion de l’alimentation / Power management module) aura pour rôle de fournir :

* une source d’alimentation de 12V pour le module **C0** ainsi que les Leds Flash et le Fan à partir d’une LiPo 2S,
* une source d’alimentation de 5V pour la Raspberry Pi, le buzzer et les servo-moteurs à partir d’une LiPo 2S,
* deux transistors MOSFET qui permettrons de piloter le buzzer et les Leds Flash.

La carte **C1** remplira donc les fonction contraintes **FC2 : Gérer l'alimentation des différents composants,** et **FC5 : Permettre le Contrôle du buzzer et des flash**.

La troisième carte électronique **C2** (Module de veille / Standby module)aurapour rôle de :

* mettre en place le système de mode veille avec l’intégration du module BLE,
* protéger la batterie LiPo 2S afin qu’elle ne descende pas en dessous d’une certaine tension,
* mesurer le pourcentage de batterie restante.

La carte **C2** remplira donc les fonction contraintes **FC1 : Protéger la batterie d’alimentation** et **FC6** : **Proposer un mode veille basse consommation & réveillable à distance**.

La dernière fonction contrainte **FC4 : Permettre le contrôle des servos-moteurs** et la fonction principale **FP1 : Permettre la commande du robot à distance via une interface de contrôle avec image en temps réel** seront assurées par la Raspberry PI.

Pour la fonction **FC4**, la Raspberry commandera les servos moteurs avec un signal PWM.

Pour l’interface de contrôle du robot et la retransmission de l’image capturée par la caméra embarquée (la fonction principale **FP1**), nous développerons une interface interactive à l’aide de web socket, de Node.js et de ses bibliothèques npm.

Cette interface sera hébergée par la Raspberry Pi sur un serveur HTTP (serveur http n°2) puis redirigée vers le serveur cloud via un tunnel SSH, comme pour la page HTML du module externe WIFI+BLE.

1.4 Schémas synoptique du projet

Avec la prise en compte de tous les éléments précédant, nous pouvons présenter le schéma synoptique du projet (Figure 5) :



Figure 5 : Schémas synoptique du projet.

Le module de veille du robot **C2** communique en BLE avec le module externe WIFI+BLE qui sert de pont entre le module de veille et le serveur cloud.

Le module de gestion d’alimentation **C1** servira à alimenter l’ensemble du robot.

La Raspberry Pi communique avec le serveur afin de proposer une interface pour contrôler le robot.

Le module driver de moteur **C0** nous permettra le contrôler les moteurs.

Le serveur cloud quant à lui propose dans un premier temps la page HTML du module externe WIFI+BLE (serveur http n°1), lui permettant de dialoguer avec le module de veille du robot afin de le mettre en veille ou le réveiller.

Le serveur cloud propose également une interface Node.js (serveur http n°2) permettant d’avoir un retour caméra ainsi que des boutons de commande permettant de déplacer le robot et la caméra sur deux axes à l’aide des servos moteurs, et de contrôler les flash et buzzer. Cette interface node.js sera donc accessible uniquement lorsque le robot est réveillé.

II. CHOIX TECHNIQUES

Cette partie a pour objectif de présenter l’ensemble des composants de ce projet, et d’établir l’architecture hardware complète du robot à développer.

Nous présenterons dans un premier temps le matériel utilisé (moteurs, carte centrale, caméra etc..) puis la liste des différents composants qui serviront à la conception des trois cartes électroniques (**C0**, **C1**, **C2**).

Nous terminerons par un schéma synthétisant l’architecture hardware du robot.

2.1 Matériel utilisé

2.1.1 Batterie D’alimentation

Pour alimenter l’ensemble du robot nous utiliserons une batterie au lithium polymère à deux cellules (figure 6) aussi appelée LiPo 2S.

****

Figure 6 : Batterie LiPo 2S

Caractéristiques de la batterie :

* Capacitée : 5200mAh
* Tension : de 7.2V (10%) à 8.4V (100%)
* Taux de décharge : 30C
* Poids : 292g

Le taux de décharge correspond à la capacité maximum et sans danger de décharge continue de la batterie, « 30C » signifie que l’on peut décharger la batterie à une puissance jusqu’à 30 fois la capacité de la batterie de façon continue soit 5200mAh et 30C = 5.2A x 30 = 156A. La batterie peut donc délivrer 156A en continue à une tension de ~ 7.4V, soit une puissance de 1155W.

2.1.2 Raspberry Pi 4

Pour ce projet nous utiliserons une Raspberry Pi 4 (Figure 7) comme cerveau central du robot.

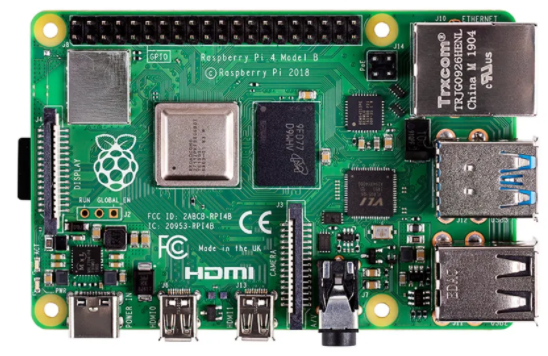


Figure 7 : Raspberry Pi 4

Voici une liste exhaustive des caractéristiques de la Raspberry Pi 4 :

* Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
* 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
* 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
* Gigabit Ethernet
* 2 USB 3.0 ports; 2 USB 2.0 ports.
* Raspberry Pi standard 40 pin GPIO header (UART, I2C, SPI, PWM)
* 2-lane MIPI DSI display port
* 2-lane MIPI CSI camera port

Source : <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>

La Raspberry aura pour fonctions de récupérer les images capturées par la caméra embarquée grâce à son port caméra CSI. Elle devra également mettre en place l’interface de contrôle du robot dans laquelle l’utilisateur pourra avoir un retour image en temps réel et contrôler le robot.

Elle devra pouvoir communiquer avec le module driver de moteur **C0** afin que l’utilisateur puisse commander les moteurs via l’interface de contrôle. La Raspberry Pi devra enfin se charger de faire pivoter les deux servos moteurs avec un signal PWM.

La Raspberry Pi 4 consomment au maximum **3A** à **5V**, ils demandent donc au maximum une puissance de **15W.**

Source : <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>

2.1.3 Caméra embarquée

Afin de récupérer une image en temps réelle, nous utiliserons la caméra Fisheye Lens de chez WavesShare (Figure 8).



Figure 8 : Camera Fisheye Lens de chez WavesShare

Voici une liste exhaustive des caractéristiques de la camera Fisheye Lens :

* Raspberry Pi Camera, supports all revisions of the Pi
* 5-megapixel OV5647 sensor
* 2592 × 1944 still picture resolution.
* Support 1080p30, 720p60 and 640x480p60/90 video record

Source : <https://www.waveshare.com/rpi-camera-i.htm>

La caméra sera reliée à la Raspberry Pi 4 via le port CSI.

2.1.4 Les moteurs DC

Afin que l’utilisateur puisse déplacer le robot dans son environnement, nous lui intégrerons quatre moteurs DC à engrenages électrique JGA25-370 (figure 9).



Figure 9 : Moteurs à engrenages électrique JGA25-37

Caractéristiques du moteur DC à engrenage :

* Nominal voltage: 12 V
* Free-run speed at 12 V: 399 RPM
* Free-run current at 12 V: 50 mA
* Stall current at 12V: 1200 mA
* Stall torque at 12V: 2.2 kg.cm
* Weight: 84 g

Source : <https://www.openimpulse.com/blog/products-page/25d-gearmotors/jga25-370-dc-gearmotor-399-rpm-12-v/>

Les moteurs consomment au maximum **1.2A** à **12V**, ils demandent donc au maximum une puissance de **14.4W** chacun, les quatre moteurs consomment donc au maximum **57.6W**. Le couple maximum d’un moteur est de 2.2Kg.cm, avec quatre moteurs le robot devrait pouvoir se déplacer sans problème.

2.1.5 Les servos-moteurs

Afin que l’utilisateur puisse faire pivoter la caméra sur deux axes, le robot sera équipé de deux servos-moteurs MG995 (figure 10).

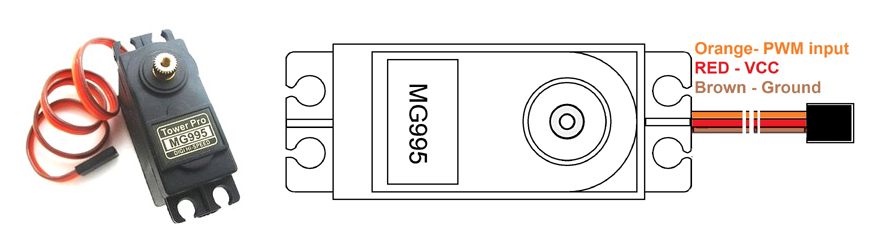


Figure 10 : Servo-moteur MG995

Caractéristiques du servo-moteur MG995 :

* Weight: 55 g
* Operating voltage range: 4.8 V to 7.2 V
* Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6v)
* Rotational degree: 180º
* Current draw at idle: 10mA
* Current at maximum load: 1200mA

Source : <https://components101.com/motors/mg995-servo-motor>

Les servos-moteurs consomment au maximum **1.2A** à **5V**, ils demandent donc au maximum une puissance de **6W** chacun, les deux servos-moteurs consomment donc au maximum **12W**. Les servos-moteurs seront pilotés par la Raspberry via un signal PWM.

2.1.6 Fan de refroidissement

Afin de refroidir la Raspberry Pi (dont la température peut monter assez haut lors d’une application demandant beaucoup de ressources) nous intégrerons un Fan Noctua NF-A4x20 FLX (figure 11).



Figure 12 : Fan Noctua NF-A4x20 FLX MG995

Caractéristiques du Fan Noctua NF-A4x20 FLX

* Max. input power: 0,6 W
* Max. input current: 0,05 A
* Operating voltage: 12 V

Source: <https://noctua.at/en/nf-a4x20-flx>

Le Fan devra donc être alimenté à **12V**, il comme **50** **mA**, soit **0,6W**.

2.1.7 Leds Flash

Afin que l’utilisateur puisse observer l’environnement du robot même dans l’obscurité, nous intégrerons des Leds Flash (figure 13).



Figure 13 : Leds Flash

Caractéristiques des Leds flash :

* Consommation d'énergie (W / m): 3,84W / m
* Tension: 12V

Source : <https://fr.aliexpress.com/item/4000178363107.html?spm=a2g0s.9042311.0.0.27426c37m6KwJ7>

Pour 1 mètre les Leds flash consomment **3,84W**, soit **320mA** à **12V**. Les Leds Flash seront pilotées à l’aide d’un transistor Mosfet sur la carte **C1** (Module de gestion d’alimentation).

2.1.8 Buzzer

Afin que l’utilisateur puisse interagir avec son environnement, nous intégrerons un buzzer (figure 14).



Figure 14 : Buzzer

Caractéristiques du buzzer :

* Courant Max : 30mA à 5VDC
* Tension d’alimentation: de 4V à 8V

Source : <https://fr.aliexpress.com/item/1594339397.html?spm=a2g0s.9042311.0.0.27426c37m6KwJ7>

Le buzzer consomme au maximum **0.03A** à **5V**, il demandent donc au maximum une puissance de **0.15W**. Tout comme les Leds Flash, le buzzer sera piloté à l’aide d’un transistor MOSFET sur la carte **C1** (Module de gestion d’alimentation).

2.2 Module externe WIFI+BLE

Le module choisit pour assurer le serveur HTTP n°1 et le serveur BLE (vu précédemment) est ESP32-DevKitC-32D (Figure 15).



Figure 15**:** ESP32-DevKitC-32D

En effet comme vu précédemment nous aurons besoin d’un module BLE+WIFI alimenté sur prise secteur, qui hébergera le serveur HTTP n°1 (dans lequel l’utilisateur demandera un réveille ou une mise en veille du robot), et qui hébergera le serveur BLE afin que le robot sache dans quel état l’utilisateur veux qu’il soit.

On retrouvera les caractéristiques de l’ESP32 dans la figure 16.

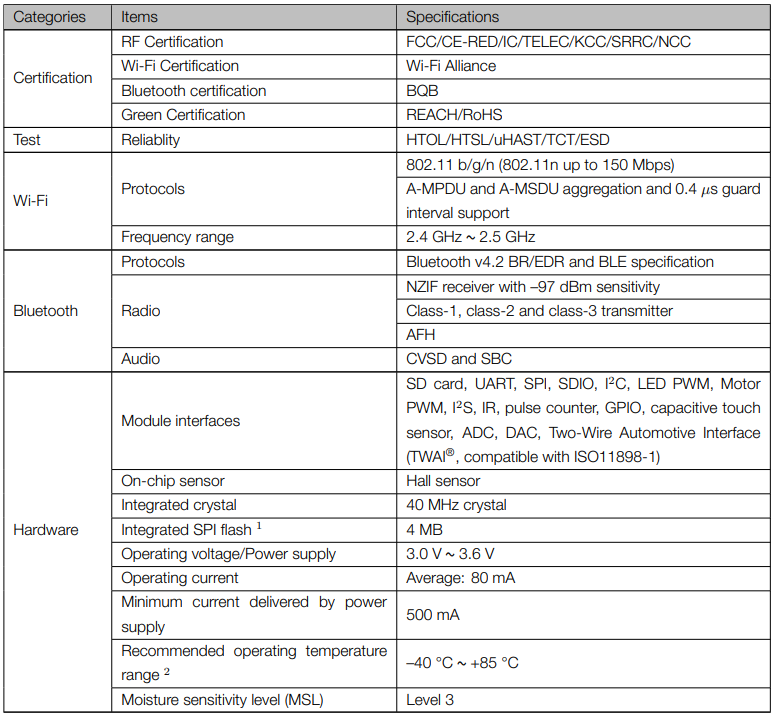


Figure 16: ESP32-WROOM-32D Caractéristiques

Source: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf>

2.3 Développement des cartes électroniques

Dans cette partie nous présenterons les trois cartes électroniques mentionnées dans le cahier des charges.

Parmi les trois cartes électroniques nous retrouvons le module driver de moteur **C0** driver (Motor driver module), le module de gestion de l’alimentation **C1** (Power Management module), et le module de veille **C2** (Standby module).

Nous détaillerons la BOM (Bill Of Materials) pour chacune des trois cartes électroniques, puis nous justifierons le choix des composants en fonction des applications.

2.2.1 Module Driver de Moteurs: C0

2.2.1.1 Bill of Materials

Voir annexes pour la listes des composants électroniques du module (Tableau 2: BOM du Module Driver de moteurs).

2.2.1.2 Fonctions du module driver de moteur

Le module doit être capable de recevoir des ordres de commande venant de la part de la Raspberry Pi via un Protocol de communication I2C

Le module doit être capable de piloter quatre moteurs DC de 12V via des signaux PWM pouvant aller jusqu’à 20 kHz.

Le module sera donc alimenté en 12V, même tension que celle des moteurs DC.

2.2.1.3 Choix du microcontrôleur

Pour générer les signaux PWM et établir un mode de communication entre la Raspberry Pi et le module driver de moteur nous utiliserons le microcontrôleur **dsPIC33EP512MC806**.

Nous avons choisi ce microcontrôleur pour sa technologie High-speed PWM (Figure 17) et ses interfaces de communication (Figure 18). On retrouvera ses principales caractéristiques à la figure 19 (voir annexes).

|  |  |
| --- | --- |
|  |  |
| Figure 17: dsPIC33EP512MC806 PWM Specifications | Figure 18: dsPIC33EP512MC806 Communication Interfaces Specifications |

Le dsPIC33E propose un signal PWM avec une résolution pouvant aller jusqu’à 8.32ns, ce qui est largement suffisant pour générer des signaux PWM de 20 kHz, de plus il permet de piloter quatre moteurs en même temps.

Le dsPIC33E propose également une interface de communication I2C que nous utiliserons afin de faire communiquer la Raspberry avec le module driver de moteur.

2.2.1.4 Choix des ponts en H

Afin de pouvoir piloter les moteurs DC avec une tension de 12V nous auront recourt à quatre ponts en H, pour cela nous choisirons les Contrôleurs et pilotes de moteur **TB67H450FNG,EL** dont on retrouvera les caractéristiques à la figure 20 (voir annexes).

Le H-Bridge TB67H450FNG,EL répond à nos exigences dans la mesure où :

* il est capable de piloter des moteurs DC en 12V (Motor power supply voltage : 4.5V->44V),
* il est capable de piloter un signal de 400 kHz, ce qui est largement suffisant pour notre Cahier des Charges,
* il est capable de délivrer 1.5A (valeur type) en courant de sortie, ce qui est suffisant: les moteurs utilisés demandent au maximum 1.2A,
* la tension d’entrée logique (2V->5.5V) est adaptée à la tension de sortie des GPIO du microcontrôleur (3.3V).

2.2.1.5 Choix du LDO

Afin d’alimenter le microcontrôleur, et donner une tension de référence au pont en H, nous utiliserons le LDO **LDL1117S33R**, on retrouver ses caractéristiques électriques sur la figure 21(Vout = 3.3V).

Le LDO LDL1117S33R répond à nos exigences dans la mesure où :

* il accepte une tension d’entrée de 12V,
* il est capable de retourner une tension de 3.3V en sortie,
* il est capable de délivrer 1.5A (valeur type) en courant de sortie, ce qui est suffisant pour notre microcontrôleur et nos pont en H.

2.2.1.6 Schémas synoptique du module de veille C0

Avec la prise en compte de tous les éléments précédant, nous pouvons présenter le schéma synoptique du module driver de moteur **C0** (Figure 22) :

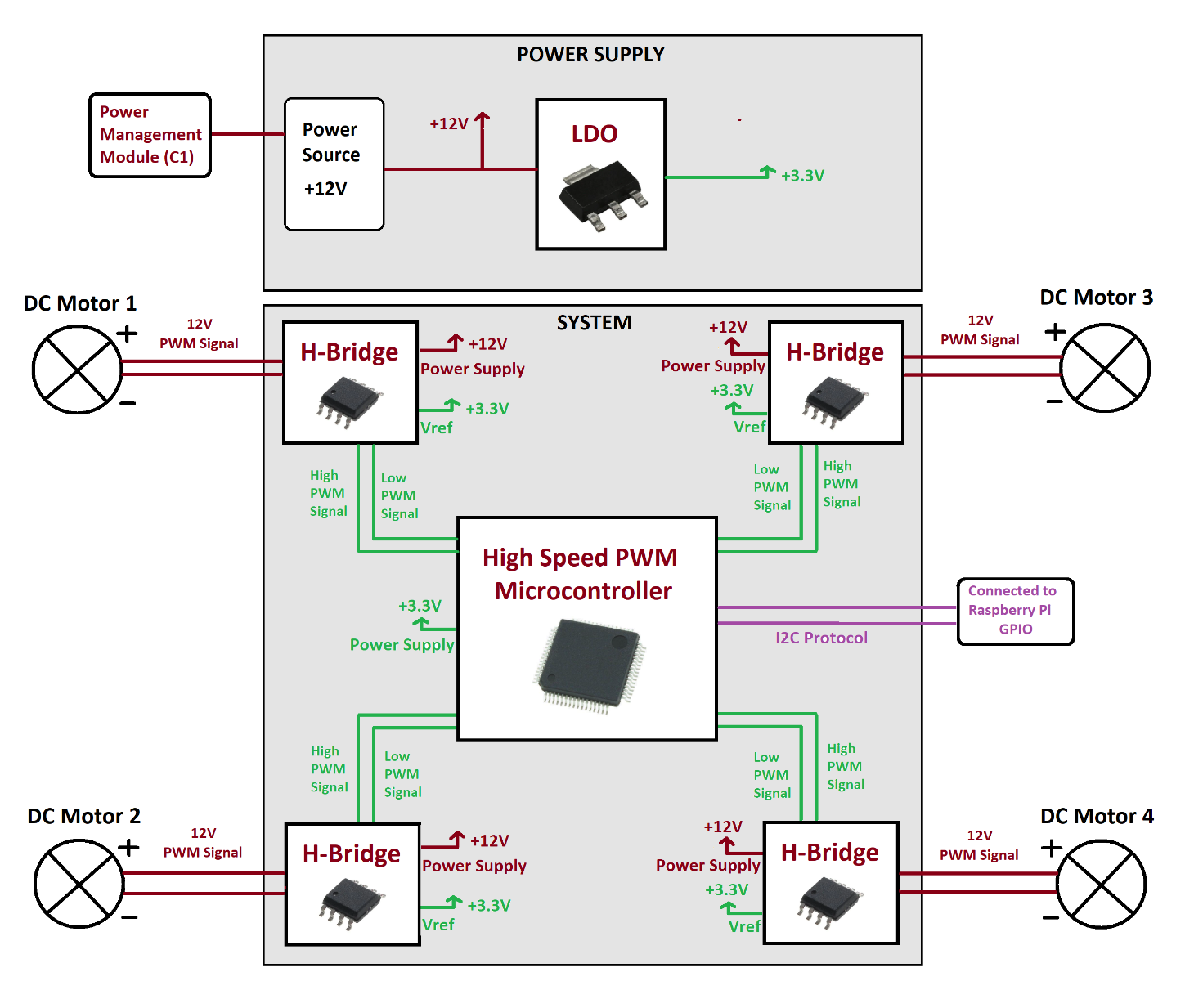


Figure 22 : Schémas synoptique du module driver de moteur **C0**.

2.2.2 Module de Gestion de l’Alimentation: C1

2.2.2.1 Bill of Materials

Voir annexes pour la listes des composants électroniques du module (Tableau 3: BOM du Module de gestion de l’alimentation).

2.2.2.2 Fonctions du module de gestion d’alimentation

A partir d’une tension de batterie LiPo 2S, le module doit être capable de :

* fournir une source d’alimentation de 12V pour les quatre moteurs DC, le Fan de refroidissement et les Leds FLASH à partir d’une LiPo 2S
* fournir une source d’alimentation de 5V pour la Raspberry Pi, les deux servo-moteurs et le Buzzer à partir d’une LiPo 2S
* proposer deux transistors MOSFET qui permettrons de piloter le buzzer et les Leds Flash.

Les différents composants consomment chacun :

* 15W pour la Raspberry Pi 4 (3A à 5V)
* 57,6W pour les moteurs DC (1.2A à 12V)\*4
* 12 W pour les servos-moteurs (1.2A à 5V)\*2
* 0.6W pour le fan de refroidissement (0.05A à 12V)
* 3.84W pour les Leds Flash (0.32A à 12V) pour 1m
* 0.15W pour le Buzzer (0.03A à 5V)

Le système global consommera donc au maximum **89.19W**. De plus, comme vu précédemment, la batterie LiPo 2S est capable de délivrer **1155W**, ce qui est donc largement suffisant pour notre système.

2.2.2.3 Choix du BUCK

Afin de fournir une tension de 5V nous abaisserons la tension de la batterie à l’aide d’une alimentation à découpage step-down (BUCK).

Pour avoir une tension continue acceptable nous imposerons une fréquence de découpage de minimum 500kHz.

Avec notre modèle actuel nous aurons besoin d’un courant de sortie de 3A (Raspberry Pi) + 2.4A (2\*servos-moteurs) + 0.03A (buzzer), soit **5.43A** au total.

Nous choisirons donc le BUCK **TPS54628DDAR** dont nous pouvons voir les principales caractéristiques sur la figure 23 (voir annexes).

Le BUCK **TPS54628DDAR** répond à nos exigences dans la mesure où :

* il accepte une tension d’entrée de 4.5V à 18V, ce qui englobe la plage de tension de la LiPo (de 7.2V à 8.4V)
* il est capable de retourner une tension entre 0 et 5.7V, ce qui comprend 5V
* il est capable de délivrer jusqu’à 6A en courant de sortie, ce qui est suffisant pour les composants alimentés en 5V

il est capable de délivrer une fréquence de découpage pouvant aller jusqu’à 650kHz

2.2.2.3 Choix du BOOST

Afin de fournir une tension de 12V nous relèverons la tension de la batterie à l’aide d’une alimentation à découpage step-up (BOOST).

Pour avoir une tension continue acceptable nous imposerons une fréquence de découpage de minimum 500kHz.

Avec notre modèle actuel nous auront besoin d’un courant de sortie de 4.8A (4\*moteurs DC) + 0.05A (fan) + 0.32A (flash), soit **5.17A** au total.

Nous choisirons donc le BOOST **LM3478MA** dont nous pouvons voir les principales caractéristiques ci-dessous :

* Fréquence de commutation : 100 kHz to 1 MHz
* Tension d'entrée 2.97V to 40V
* Courant de sortie max : 10A
* Courant d'alimentation de fonctionnement : 2.7mA

Le BOOST **LM3478MA** répond à nos exigences dans la mesure où :

* il accepte une tension d’entrée de 2.97V à 40V, ce qui englobe la plage de tension de la LiPo (de 7.2V à 8.4V)
* il est capable de retourner une tension de 12V,
* il est capable de délivrer jusqu’à 10A max en courant de sortie, ce qui est suffisant pour les composants alimentés en 12V (5.17A au total).
* il est capable de délivrer une fréquence de découpage pouvant aller jusqu’à 1MHz.

Pour la mise ne place du BOOST **LM3478MA**, nous auront recours au WEBENCH Power Designer de Texas Instrument.

2.2.2.4 Choix des MOSFETs

Afin de pouvoir contrôler les Leds Flash et le buzzer, nous aurons recours à deux transistors MOSFET Canal N.

Nous choisirons le MOSFET **DMN6075SQ-7** dont nous retrouverons les principales caractéristiques sur le figure 24.

Le MOSFET **DMN6075SQ-7** répond à nos exigences dans la mesure où :

* il accepte une tension d’entrée jusqu’à 60V, ce qui suffisant pour le buzzer (5V) et les Flash (12V)
* il est capable laisser passer un courant d’au moins 2A (à 25°C), ce qui est largement suffisant pour les Flash et le buzzer
* il est pilotable avec une tension entre 1V et 3V, ce qui correspond à la tension des GPIO de la Raspberry Pi

2.2.2.5 Schémas synoptique du module de gestion de l’alimentation C1

Avec la prise en compte de tous les éléments précédant, nous pouvons présenter le schémas synoptique du module de gestion de l’alimentation **C1** (Figure 25) :

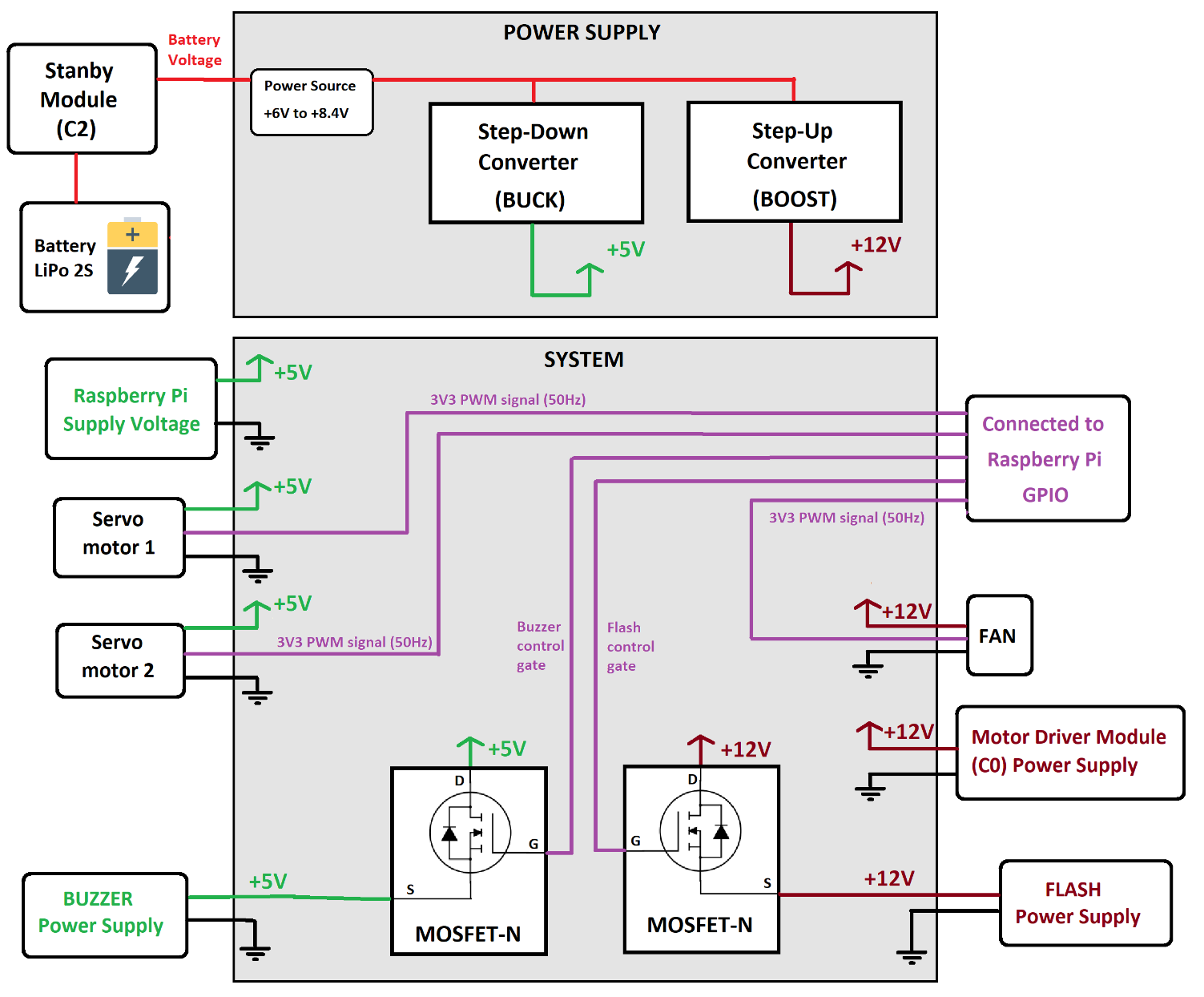


Figure 25 : Schémas synoptique du module de gestion de l’alimentation.

2.2.3 Module de veille (C2)

2.2.3.1 Bill of Materials

Voir annexes pour la listes des composants électroniques du module (Tableau 4 : BOM du Module de veille (C2)).

2.2.3.2 Fonctions du module de veille

Le module de veille **C2** aura pour fonction de protéger la batterie afin qu’elle n’atteigne pas une tension inférieur à 7.4V (Tension de coupure du système).

Le module devra être Low Power design, il devra fonctionner en consommant le moins possible et devra limiter les courants de fuite au maximum.

Le module de veille devra également être capable de couper ou d’envoyer le courant au reste du robot selon le désir de l’utilisateur grâce au module BLE qui sera intégré au module de veille, ou à l’aide des deux bouton-poussoir (start et stop).

Le module devra également détecter la tension de la batterie pour en déduire le pourcentage de batterie restante.

2.2.3.3 Choix du module BLE

Afin de pouvoir communiquer avec le module externe ESP32, nous choisirons d’intégrer un RN4020.

Le RN4020 est un module Bluetooth Low Energy pouvant servir de client comme de serveur, il est facilement pilotable à l’aide d’un microcontrôleur et d’un protocole de communication UART.

On retrouvera une liste exhaustive des caractéristiques du RN4020 aux figures 26et 27**.**

|  |  |
| --- | --- |
|  |  |
| Figure 26: Consommation du RN4020 | Figure 27: Caractéristiques du RN4020 |

Le RN4020répond à nos exigences dans la mesure où :

* il est facilement pilotable via un protocole de communication UART
* il présente un mode sleep
* il consomme relativement peu de courant, même lors de communication BLE
* il peut se comporter comme un client BLE, et venir écrire/lire dans les registres du serveur BLE de l’ESP32

2.2.3.4 Choix du microcontrôleur

Afin de piloter le RN4020 et les MOSFETs, de détecter la pressions des bouton-poussoir et de mesurer le pourcentage de batterie restante, nous choisirons le microcontrôleur 8 bit **PIC16F18444** dont nous retrouverons une liste exhaustive des caractéristiques au figure 28 et 29.

|  |  |
| --- | --- |
|  |  |
| Figure 28: Caractéristiques principales du PIC16F18444 | Figure 29: XLP caractéristiques du PIC16F18444 |

Le PIC16F18444 répond à nos exigences dans la mesure où :

* il présente des caractéristiques « eXtreme Low-Power » intéressantes avec un sleep mode à quelques μA/MHZ
* il permet une communication UART qui nous permettra de piloter le RN4020
* il possède un ADC 12 bit pour la mesure de pourcentage de batterie restante
* il ne coûte pas chère (~1$)

2.2.3.5 Choix du circuit de surveillance

Afin que la batterie n’atteigne pas un voltage trop bas, nous aurons recours au circuit de surveillance **MAX6457**.

Le **MAX6457** est un contrôleur permettant la détection de sous-tension ou de surtension, dans notre cas de figure nous l’utiliserons pour détecter une sous-tension : la tension de coupure du système.

Nous choisirons le modèle **MAX6457UKD3B+T** car il nous propose une hystérésis de 5% (réf. figure 30), cette hystérésis évitera au système d’osciller entre on et off lorsque la tension de la batterie approchera la tension de coupure du système.

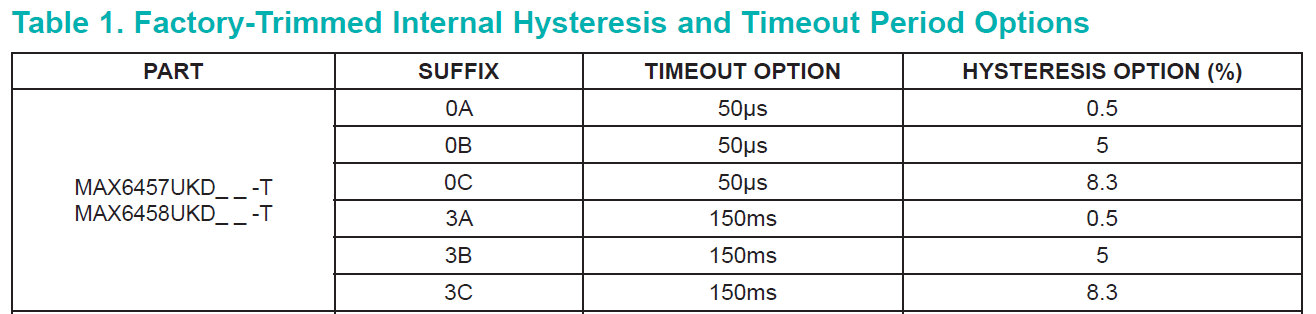


Figure 30: Hystérésis et Timeout Period du **MAX6457**

2.2.3.6 Choix du LDO

Afin d’alimenter le microcontrôleur et le RN4020 nous choisirons le LDO **AP2205-33W5-7** dont nous retrouverons les principales caractéristiques à la figure31.

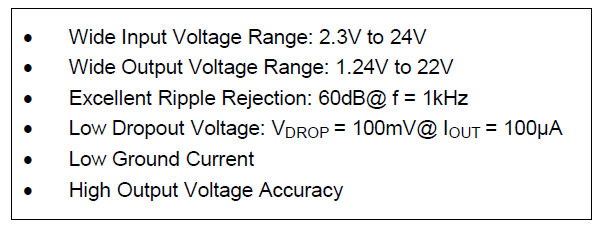


Figure 31 : Principales caractéristiques du LDO **AP2205-33W5-7**

Le AP2205-33W5-7répond à nos exigences dans la mesure où :

* il est facilement pilotable en branchant la sortie du LVD MAX6457 à la broche enable du LDO.
* il est capable de fournir un courant de sortie pouvant aller jusqu’à 250mA, ce qui est suffisant pour notre module
* lorsque sa broche enable n’est pas sous tension, il consomme un courant de veille assez faible (10nA type) ce qui nous permettra de limiter les courant de fuites lorsque la batterie est déchargée

2.2.3.7 Choix des MOSFETs

Dans la conception de ce module nous aurons recourt à quatre MOSFETs différents, deux MOSFETs à canal N, et deux MOSFETs à canal P.

Chacun des deux MOS-N (Q2 et Q4) serviront à piloter un des deux MOS-P (Q1 et Q3), Q2 pilote Q1 et Q4 pilote Q3.

Le premier MOS-P (Q1) aura pour fonction de délivrer le courant au reste du robot alors que le deuxième MOS-P (Q2) aura pour fonction de laisser passer le courant de la batterie vers une broche du microcontrôleur afin que celui puisse faire une conversion analogique-numérique et détecter le pourcentage de batterie restant.

L’utilité d’intégrer le deuxième MOS-P (Q3) et le deuxième MOS-N (Q4) est de limiter les courants de fuite en ne laissant pas la broche du microcontrôleur constamment sous tension (à contrario les MOSFET ont un courant de fuite de l’ordre du μA).

**Les MOSFET Q1 et Q2 :**

Pour Q1 nous choisiront le MOS-P **DMP34M4SPS-13** dont nous retrouverons les principales caractéristiques à la figure 32, et pour Q2 nous choisiront le MOS-P **RS1E240BNTB** dont nous retrouverons les principales caractéristiques à la figure 33**.**

|  |  |
| --- | --- |
|  |  |
| Figure 32: Caractéristiques du **DMP34M4SPS-13** | Figure 33: Caractéristiques du **RS1E240BNTB** |

Les MOSFETs Q1 et Q2répondent à nos exigences dans la mesure où :

* ils présentent tout les deux un Low Rds(on) très faible d’environ 3mΩ (type), ce qui limitera les perte notamment pour Q1 qui laisse passer le courant qui alimente tout le robot
* Q1 est capable laisser passer un courant pouvant aller jusqu’à 135A, ce qui est largement suffisant pour notre système : 135A à ~8V soit ~1000W (le système global consommera donc au maximum 89.19W)
* Q2 est pilotable avec une tension entre 1V et 2.5V, ce qui correspond à la tension du microcontrôleur.

**Les MOSFET Q3 et Q4 :**

Pour Q3 nous choisiront le MOS-P **PMV65XPER** dont nous retrouverons les principales caractéristiques à la figure 34, et pour Q4 nous choisiront le MOS-P **NX7002AKVL** dont nous retrouverons les principales caractéristiques à la figure 35**.**

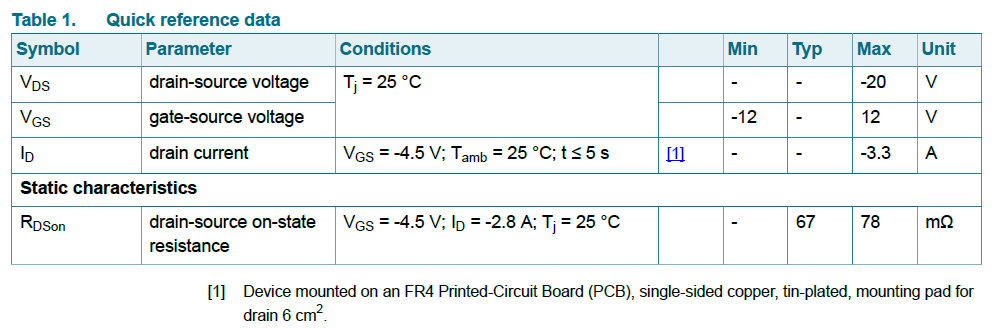


Figure 34 : Principales caractéristiques du **PMV65XPER**

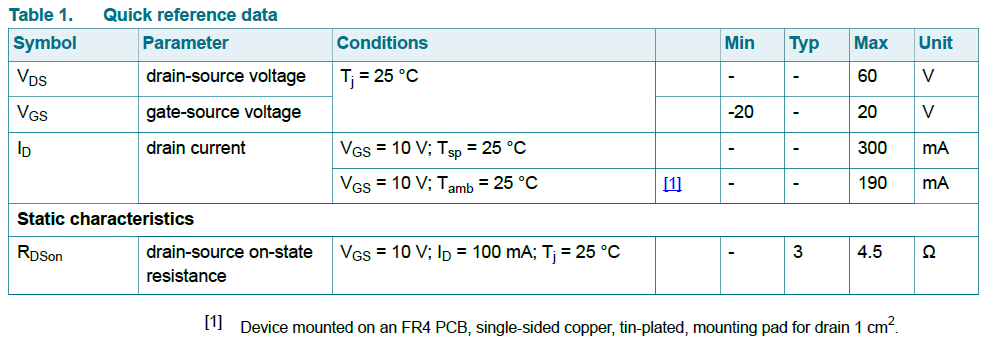


Figure 35 : Principales caractéristiques du **NX7002AKVL**

Les MOSFETs Q3 et Q4répondent à nos exigences dans la mesure où :

* ils présentent tous les deux un Low Rds(on) assez faible 3mΩ (type) et 67mΩ (type)
* Q4 est pilotable avec une tension entre 1.1V et 2.1V, ce qui correspond à la tension du microcontrôleur.

2.2.3.8 Choix des bouton-poussoir

Afin de pouvoir démarrer l’ensemble du robot physiquement (sans passer par le serveur http), nous aurons recours à deux bouton-poussoir **1241.1600.11** (illustration à la Figure36).

Les bouton-poussoir 1241.1600.11répondent à nos exigences dans car il laissent passer 50mA (ce qui est suffisant), et ils sont beaux.

Figure 36 : Bouton-poussoir **1241.1600.11**

2.2.3.9 Schémas synoptique du module de veille C2

Avec la prise en compte de tous les éléments précédant, nous pouvons présenter le schémas synoptique du module de veille **C2** (Figure 37) :

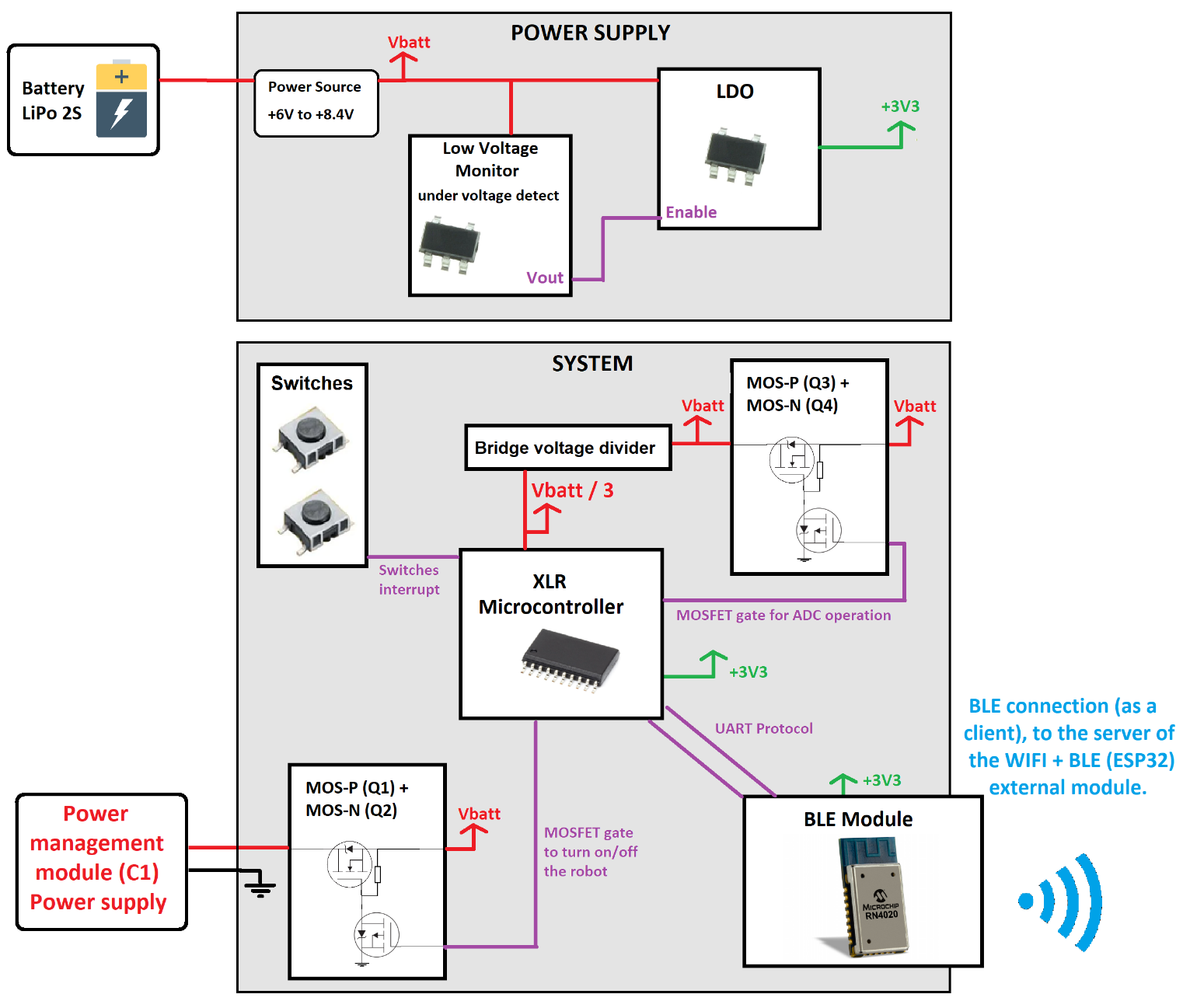


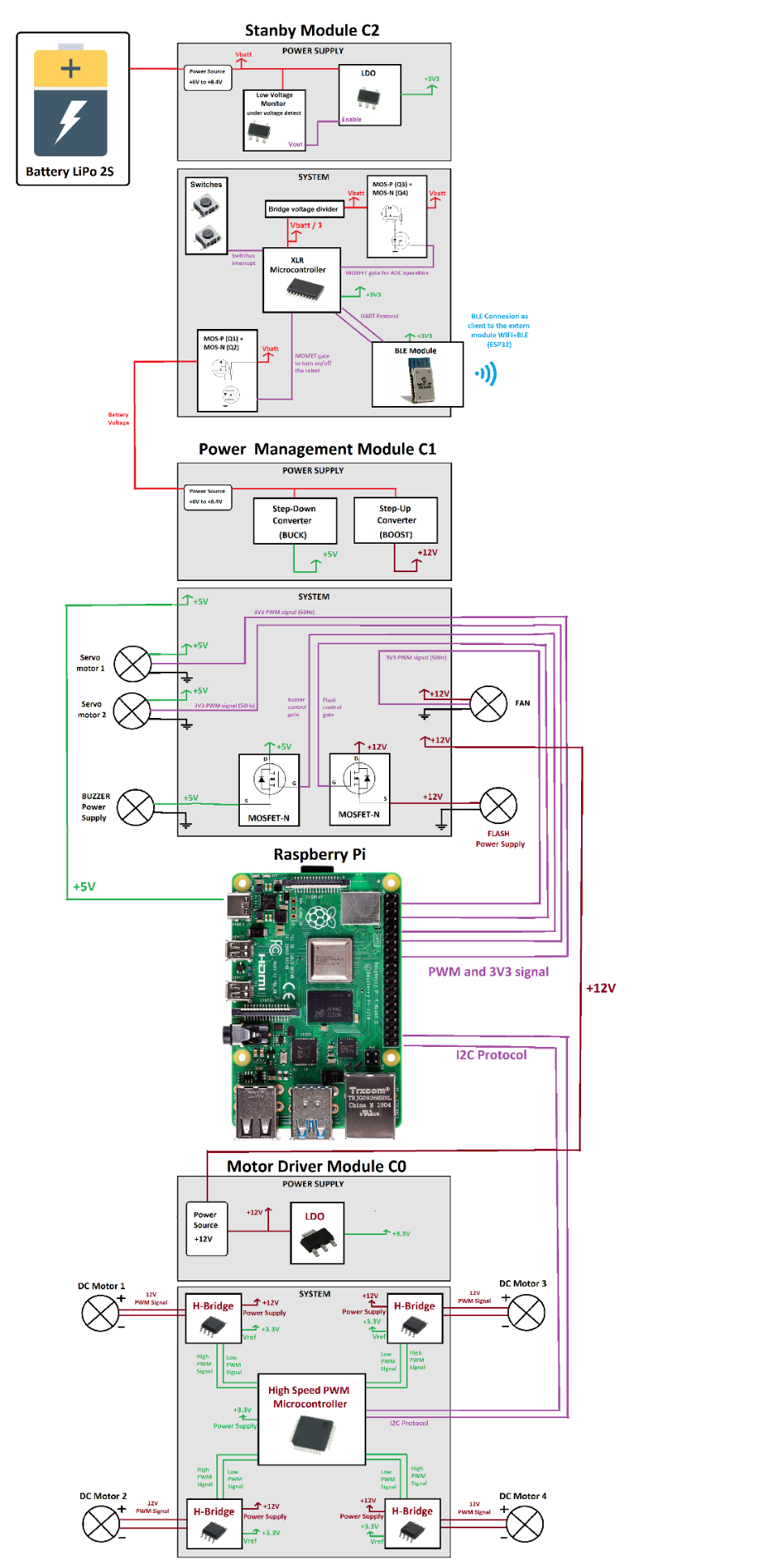
Figure 37 : Schémas synoptique du module de veille (C2).

Le module de veille C2 est directement alimenté par la batterie LiPo 2S, lorsque le circuits de surveillance **MAX6457** détecte une tension inférieur à la tension de coupure du système, il met le LDO et l’ensemble des composants du module de veille en standby.

Lorsque la tension est supérieure à la tension de coupure du système, le microcontrôleur central prend en charge la routine du module de veille :

* il récupère la tension de la batterie pour estimer le pourcentage de batterie restant
* il pilote le module BLE (RN4020) afin qu’il demande au module externe (ESP32) si le robot doit être allumé ou non
* il détecte la pression des bouton-poussoir
* il laisse passer/coupe le courant vers le module de gestion de l’alimentation (C1) en fonction de la réponse reçue par le RN4020 et les interruptions des bouton-poussoir

2.4 Schéma synoptique du robot Babebot

A l’aide des trois schémas synoptiques des modules C0, C1, C2 nous présenterons le schémas synoptique global du robot Babebot à la figure 38.

Le module de veille **C2** qui :

* communique en BLE avec le module externe WIFI+BLE (ESP32) afin de récupérer les informations concernant l’état désiré du robot, et d’envoyer les information concernant l’état actuel du robot et concernant l’état de la batterie,
* s’assure de ne pas endommager la batterie avec un module de détection de sous-tension,
* récupère la tension de la batterie afin d’estimer le pourcentage de charge restante,
* détecte si l’utilisateur veux allumer ou éteindre le robot à l’aide des bouton-poussoir,
* laisse passer ou coupe le courant vers le reste du robot à l’aide des MOSFETs.

Le module de gestion de l’alimentation **C1** qui :

* génère une tension de +5V et +12V à l’aide d’alimentation à découpage
* alimente la Raspberry Pi, les servos moteurs, les flash, le buzzer, le fan, ainsi que le module driver de moteur.
* permet à la Raspberry de piloter les servos moteurs et le fan via un signal PWM, et les buzzer et flash à l’aide d’un signal 3V3 et de MOSFETs intégré au module **C1**.

La Raspberry Pi qui :

* propose une interface permettant de piloter les moteurs, les servos moteurs, les flash et buzzer à distance, et permettant de proposer un retour caméra en temps réel,
* pilote le module driver de driver de moteur via un protocole I2C et en tant que maître.

Le module driver de moteur **C0** qui :

* pilote les quatre moteurs DC à l’aide d’un signal PWM de +12V,
* reçoit les informations via un protocole I2C (en tant qu’esclave) concernant la vitesse et le sens de rotation des moteurs de la part de la Raspberry Pi .

Figure **38** : Schémas synoptique du Robot BabeBot

III. ANNEXES

3.1 Caractéristiques composants

3.1.1 Principales caractéristiques du dsPIC33EP512MC806

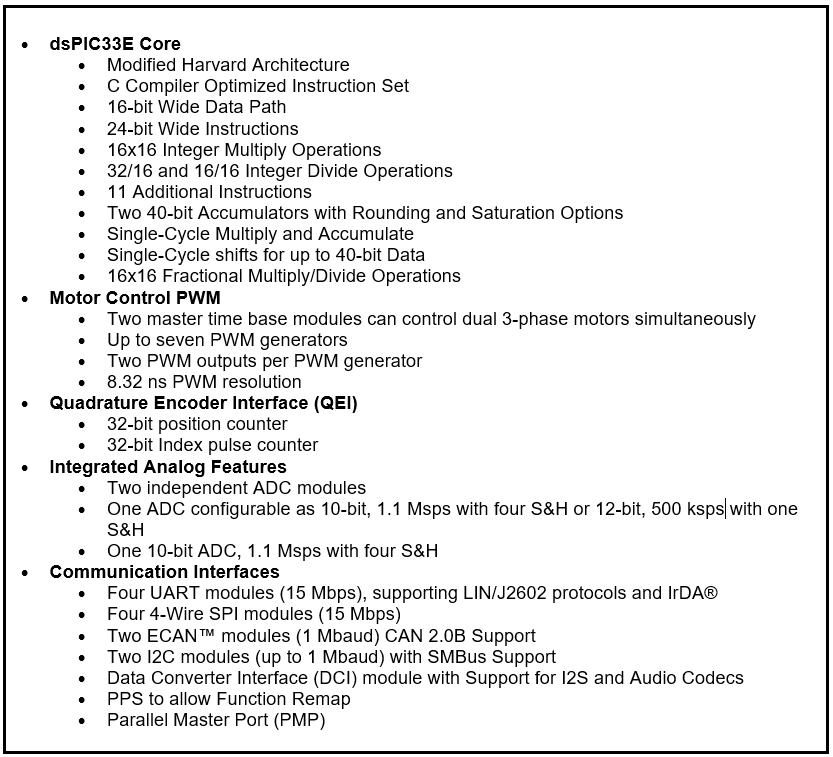


Figure 19 : Principales caractéristiques du **dsPIC33EP512MC806**

3.1.2 Caractéristiques du TB67H450FNG,EL

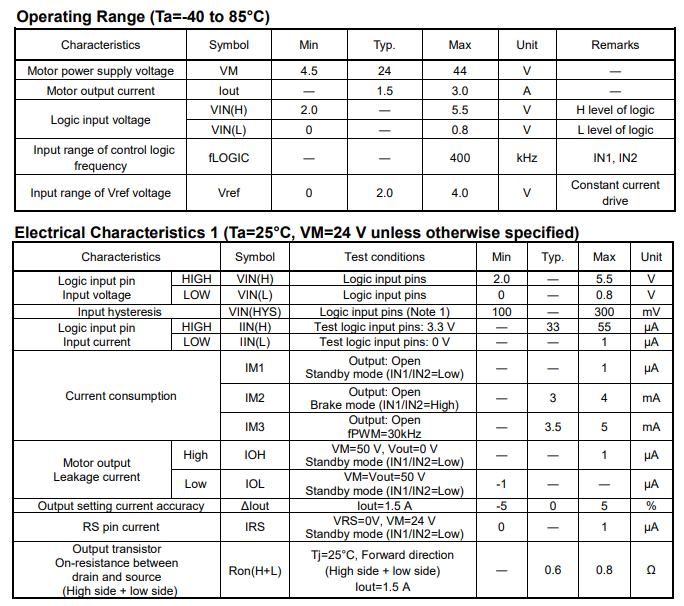


Figure 20: Caractéristiques du TB67H450FNG,EL

3.1.3 Caractéristiques électriques du LDL1117S33R

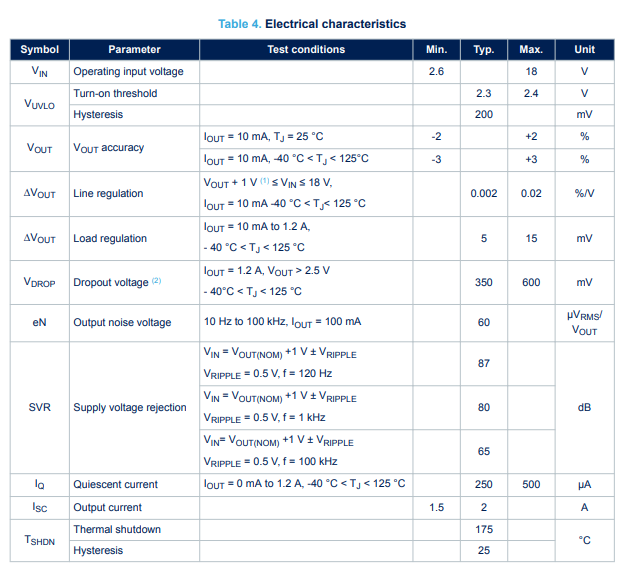


Figure 21: Caractéristiques électriques du LDL1117S33R

3.1.4 Caractéristiques du TPS54628DDAR

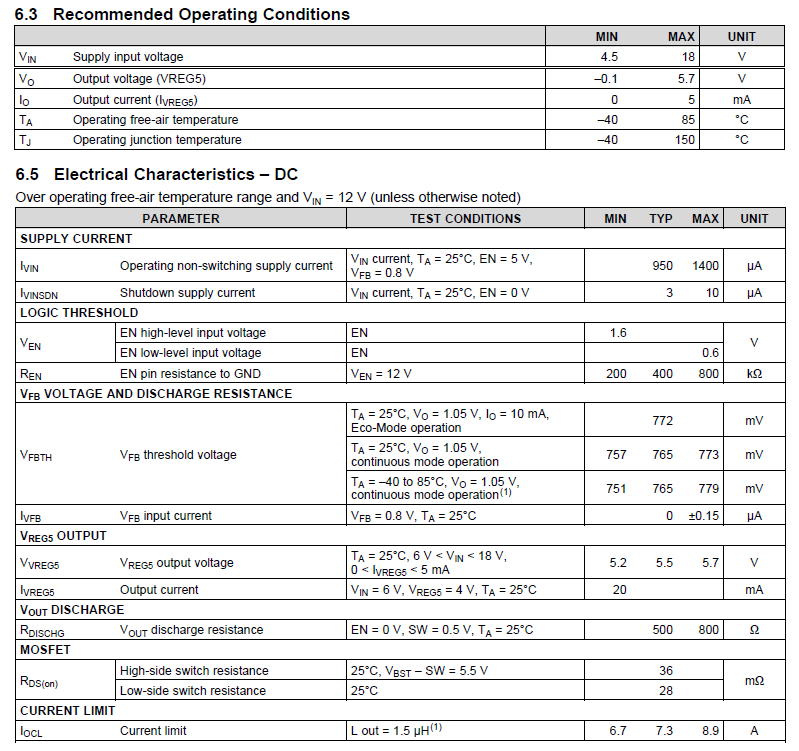


Figure 23: Caractéristiques du TPS54628DDAR

3.1.5 Principales caractéristiques du MOSFET DMN6075SQ-7

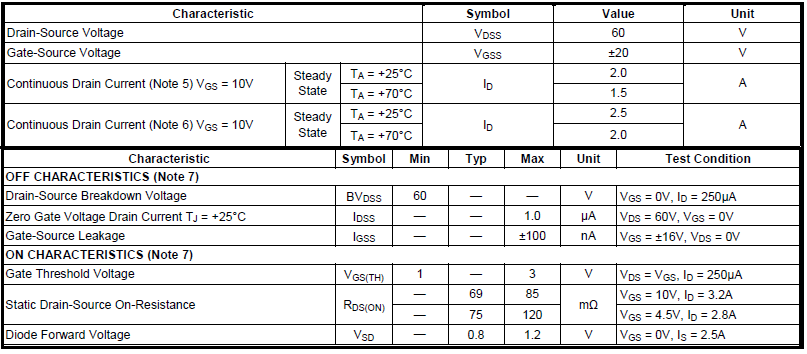


Figure 24 : Principales caractéristiques du MOSFET **DMN6075SQ-7**

3.2 BOM des cartes électroniques :

3.2.1 BOM du module driver de moteur

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **REF DESIGNATOR** | **PART NUMBER** | **QUANTITE** | **DESCRPTION** | **CONSTRUCTEUR** | **FOURNISSEUR** | **REF FOURNISSEUR** | **PRIX (euro)** | **REF** |  |  |  |
| **U1** | **dsPIC33EP512MC806** | **1** | **Microcontrôleurs 16 bits - MCU 512KB FL 53248Bytes RAM GP** | **microship** | **Mouser** | **579-24EP512GP806IPT** | **6,60** | <https://www.mouser.fr/ProductDetail/Microchip-Technology/PIC24EP512GP806-I-PT?qs=%2Fha2pyFadujbVPSa3qMXzQUnCNbDQZi1qezLfRVaJQzO2wYT4NL3S8bkKLP4qXkc> |  |  |  |
| **U2, U3, U4, U5** | **TB67H450FNG,EL** | **4** | **Contrôleurs et pilotes de moteur / de mouvement / d'allumage MOTOR DRIVER IC 4.5 V to 44 V 3.5A** | **Toshiba** | **Mouser** | **757-TB67H450FNGEL** | **1,3** | <https://www.mouser.fr/ProductDetail/Toshiba/TB67H450FNGEL?qs=h6V4JsTaLXeReNg2bp0jug%3D%3D> |  |  |  |
| **U6** | **LDL1117S33R** | **1** | **LDO : 3.3 V 1.2A, quiest curr : 250 uA, vin : 2.5-18V** | **STMicroelectronics** | **Mouser** | **511-LDL1117S33R** | **0,407** | <https://www.mouser.fr/ProductDetail/STMicroelectronics/LDL1117S33R?qs=AQlKX63v8Rt9Bf6AWSrbFg%3D%3D> |  |  |  |
| **D1** | **APTD2012LSURCK** | **1** | **CMS 0805 Dome Lens LED Red- 630nm, 1,75V** | **Kingbright** | **Mouser** | **604-APTD2012LSURCK** | **0,33** | <https://www.mouser.fr/ProductDetail/Kingbright/APTD2012LSURCK?qs=AQlKX63v8RvkMadNvwbkxw%3D%3D> |  |  |  |
| **D2** | **APTD2012LCGCK** | **1** | **CMS 0805 Dome Lens LED Green- 570nm, 1,9V** | **Kingbright** | **Mouser** | **604-APTD2012LCGCK** | **0,347** | <https://www.mouser.fr/ProductDetail/Kingbright/APTD2012LCGCK?qs=AQlKX63v8RvNHN%2F9EGvNTg%3D%3D> |  |  |  |
| **C1,C2,C3,C4,C5** | **CC0805KRX7R7BB104** | **5** | **MLCC - CMS 100nF 16V X7R 10%** | **Yaego** | **Mouser** | **603-CC805KRX7R7BB104** | **0,085** | https://www.mouser.fr/ProductDetail/603-CC805KRX7R7BB104 |  |  |  |
| **C6** | **EMK212ABJ106KG-T** | **1** | **MLCC - CMS 10uF 16V X5R +/-10% 0805 Gen Purp** | **Taiyo Yuden** | **Mouser** | **963-EMK212ABJ106KG-T** | **0,144** | https://www.mouser.fr/ProductDetail/Taiyo-Yuden/EMK212ABJ106KG-T?qs=vF%252B8ahGVilfgbAjT%2FgKCVQ%3D%3D |  |  |  |
| **C7,C8,C9,C10** | **C0805C101M3HACTU** | **4** | **MLCC - CMS 25V 100pF X8R 0805 20%** | **KEMET** | **Mouser** | **80-C0805C101M3HACTU** | **0,085** | https://www.mouser.fr/ProductDetail/KEMET/C0805C101M3HACTU?qs=W0yvOO0ixfGVyTxo%2FRw90g%3D%3D |  |  |  |
| **C11** | **C0805C105K4RACAUTO** | **1** | **MLCC - CMS 16V 1uF X7R 0805 10% AEC-Q200** | **KEMET** | **Mouser** | **80-C0805C105K4RAUTO** | **0,152** | <https://www.mouser.fr/ProductDetail/KEMET/C0805C105K4RACAUTO?qs=OHu5RF0w3QprP0D5VzlqaQ%3D%3D> |  |  |  |
| **C12** | **GRM21BR71C475ME51K** | **1** | **MLCC - CMS 4.7UF 16V 20% 0805** | **Murata Electronics** | **Mouser** | **81-GRM21BR71C475ME1K** | **0,152** | https://www.mouser.fr/ProductDetail/Murata-Electronics/GRM21BR71C475ME51K?qs=QzBtWTOodeV0aa0oJHX5rA%3D%3D |  |  |  |
| **R1** | **SG73G2ATTD1002D** | **1** | **CMS 0.5% 0.5W AEC-Q200, 10kOhm,** | **KOA Speer** | **Mouser** | **660-SG73G2ATTD1002D** | **0,231** | <https://www.mouser.fr/ProductDetail/KOA-Speer/SG73G2ATTD1002D?qs=sGAEpiMZZMtlubZbdhIBIFC33eDTuoa6V25eeInASp8%3D> |  |  |  |
| **R2** | **ERJ-6ENF71R5V** | **1** | **CMS 0805 71.5ohms 1% AEC-Q200** | **Panasonic** | **Mouser** | **667-ERJ-6ENF71R5V** | **0,085** | https://www.mouser.fr/ProductDetail/Panasonic/ERJ-6ENF71R5V?qs=50QC8w71jAvDQTR5ftk9Ug%3D%3D |  |  |  |
| **R3** | **ERJ-6ENF78R7V** | **1** | **CMS 0805 78.7ohms 1% AEC-Q200** | **Panasonic** | **Mouser** | **667-ERJ-6ENF78R7V** | **0,085** | https://www.mouser.fr/ProductDetail/Panasonic/ERJ-6ENF78R7V?qs=8%252BG0TiN%2FgVougykeZbNYYg%3D%3D |  |  |  |

Tableau **2**: BOM du Module Driver de moteurs

3.2.2 BOM du module de gestion de l’alimentation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **REF DESIGNATOR** | **PART NUMBER** | **QUANTITE** | **DESCRPTION** | **FOURNISSEUR** | **CONSTRUCTEUR** | **REF FOURNISSEUR** | **PRIX** | **LINK** |
| **U1** | **TPS54628DDAR** | **1** |  | **Mouser** | **Texas Instruments** | **595-TPS54628DDAR** | **1,7** | <https://www.mouser.fr/ProductDetail/Texas-Instruments/TPS54628DDAR?qs=sGAEpiMZZMtitjHzVIkrqaNW%252B5L5alOfN0WI9Y5%2Fkb8=> |
| **L1** | **SPM6530T-3R3M-HZ** | **1** | **Inductance fixes 3.3uH 20% 27mOhms Pwr Indctr AEC-Q200** | **Mouser** | **TDK** | **810-SPM6530T-3R3M-HZ** | **1.44** | https://www.mouser.fr/ProductDetail/TDK/SPM6530T-3R3M-HZ/?qs=%2Fha2pyFaduicSDKlDwB%2F3ISBAYHc8CznNgBxwdxnB0U17DKemkJw9Q%3D%3D |
| **C7, C8** | **JMK325BJ226MM-T** | **2** | **Capacitor, Ceramic, 6.3V, X5R, 20%, 22uF** | **Mouser** | **Taiyo Yuden** | **963-JMK325BJ226MM-T** | **0,513** | https://www.mouser.fr/ProductDetail/Taiyo-Yuden/JMK325BJ226MM-T?qs=I6KAKw0tg2x8MdlsXzffTw%3D%3D |
| **C1, C2** | **TMK325BJ106MM-P** | **2** | **Capacitor, Ceramic, 25V, X5R, 20%, 10uF** | **Mouser** | **Taiyo Yuden** | **963-TMK325BJ106MM-P** | **0,378** | https://www.mouser.fr/ProductDetail/Taiyo-Yuden/TMK325BJ106MM-P?qs=VWNNG7jHlrLL2%252BMilAHFrw%3D%3D |
| **C3, C6** | **GCJ188R71H104KA12D** | **2** | **Capacitor, Ceramic, 50V, X7R, 10%, 0,1uF** | **Mouser** | **Murata Electronics** | **81-GCJ188R71H104KA2D** | **0,225** | https://www.mouser.fr/ProductDetail/Murata-Electronics/GCJ188R71H104KA12D?qs=zThidflJDvYKuGhiQaflUg%3D%3D |
| **C4** | **C0805C105K4RACAUTO** | **1** | **MLCC - CMS 16V 1uF X7R 0805 10% AEC-Q200** | **Mouser** | **KEMET** | **80-C0805C105K4RAUTO** | **0,152** | [**https://www.mouser.fr/ProductDetail/KEMET/C0805C105K4RACAUTO?qs=OHu5RF0w3QprP0D5VzlqaQ%3D%3D**](https://www.mouser.fr/ProductDetail/KEMET/C0805C105K4RACAUTO?qs=OHu5RF0w3QprP0D5VzlqaQ%3D%3D) |
| **C5** | **0805YC822KAT2A** | **1** | **MLCC - CMS 16V 8200pF X7R 0805 10% Tol** | **Mouser** | **AVX** | **581-0805YC822KAT2A** | **0,237** | [**https://www.mouser.fr/ProductDetail/AVX/0805YC822KAT2A?qs=6ozWKCQkT2X%252B2UWbUcyiSg%3D%3D**](https://www.mouser.fr/ProductDetail/AVX/0805YC822KAT2A?qs=6ozWKCQkT2X%252B2UWbUcyiSg%3D%3D) |
| **R1** | **CPF0603F124KC1** | **1** | **Resistor, Chip, 1/16W, 1%** | **Mouser** | **TE Connectivity / Neohm** | **279-CPF0603F124KC1** | **0,135** | <https://www.mouser.fr/ProductDetail/TE-Connectivity-Neohm/CPF0603F124KC1?qs=5Dj%252BCDyFp7ipizaqSu9clA==> |
| **R2** | **RN731JTTD2212B25** | **1** | **Resistor, Chip, 1/16W, 1% 22,1KΩ** | **Mouser** | **KOA Speer** | **660-RN731JTTD2212B25** | **0,463** | https://www.mouser.fr/ProductDetail/KOA-Speer/RN731JTTD2212B25?qs=SZDmkwkWGmnyhKRSdRPb7g%3D%3D |
| **R3** | **660-RN731JTTD1002D25** | **1** | **Resistor, Chip, 1/16W, 1% 10KΩ** | **Mouser** | **KOA Speer** | **RN731JTTD1002D25** | **0,151** | <https://www.mouser.fr/ProductDetail/KOA-Speer/RN731JTTD1002D25?qs=H%2FxPPQCHZXuMGwcYON8Lkw%3D%3D> |
| **U2** | **LM3478MA/NOPB** | **1** |  | **Mouser** | **TI** | **926-LM3478MA/NOPB** | **2,31** | <https://www.mouser.fr/ProductDetail/?qs=8BDZloOR%252BZH9CVb1MduuNw%3D%3D> |
| **L2** | **SRP1270-2R2M** | **1** | **Tolérance:20 %, Courant CC max.:22 A,Résistance CC max.:4.2 mOhms, 2,2uH** | **Mouser** | **Bourns** | **652-SRP1270-2R2M** | **1,1** | https://www.mouser.fr/ProductDetail/Bourns/SRP1270-2R2M?qs=%2Fha2pyFaduh%252BcDExOxFZtWKAwzk%2Fu5SsoGKT2ktF16FjXqWAvFDUwA%3D%3D |
| **D1** | **STPS20M100SG-TR** | **1** | **20 A, Vrrm 100 V** | **Mouser** | **STMicroelectronics** | **511-STPS20M100SG-TR** | **2,42** | <https://www.mouser.fr/ProductDetail/STMicroelectronics/STPS20M100SG-TR?qs=%2Fha2pyFadugA76ttzXntatuf5X%252BXZUN2567QwsVZ2TNOzKi8P8Q3Zg%3D%3D> |
| **Q1** | **CSD17310Q5A** | **1** | **MOSFET 30V N ,30V 100A 5.1 mOhms** | **Mouser** | **TI** | **595-CSD17310Q5A** | **0,819** | <https://www.mouser.fr/ProductDetail/Texas-Instruments/CSD17310Q5A?qs=%2Fha2pyFadugSLAmGeKqFReRD1niZHr5E%2FS1oDj4XRsCr4%2F39uBbhpg%3D%3D> |
| **C11** | **12065A202JAT2A** | **1** | **Ccomp2, 2nF, MLCC - CMS 50V C0G 1206 5%** | **Mouser** | **AVX** | **581-12065A202J** | **0,441** | <https://www.mouser.fr/ProductDetail/AVX/12065A202JAT2A?qs=pEPtdg%2FKdQrjLZ5CYLoxag%3D%3D> |
| **C10** | **C0805C473J3GACTU** | **1** | **Ccomp1 0.047uF, 25 VDC,** | **Mouser** | **KEMET** | **80-C0805C473J3G** | **0,72** | <https://www.mouser.fr/ProductDetail/KEMET/C0805C473J3GACTU?qs=JsD2jqDDQCwhCZAPlTi%252BPg%3D%3D> |
| **R6** | **RC1206FR-071K91L** | **1** | **Rcomp1 1.91K CMS OHM 1%** | **Mouser** | **Yageo** | **RC1206FR-071K91L** | **0,09** | https://www.mouser.fr/ProductDetail/Yageo/RC1206FR-071K91L?qs=%2Fha2pyFaduhRUg1%252BSfBETCcH3lD8Gjc%2F0hW0cUzaS2p%252BAzIz0GPyJQ%3D%3D |
| **R7** | **ERJ-8ENF2672V** | **1** | **Rfadj1 26.7Kohm CMS 1206 1% AEC-Q200** | **Mouser** | **Panasonic** | **ERJ-8ENF2672V** | **0,09** | <https://www.mouser.fr/ProductDetail/Panasonic/ERJ-8ENF2672V?qs=%2Fha2pyFadugUc0484I4hLcWtajivuevFECZKQUIRZluxPJhWylO7%252Bg%3D%3D> |
| **C12** | **C0805C100M4GACTU** | **1** | **Cfilt1 10pF MLCC - CMS 16V C0G 0805 20%** | **Mouser** | **KEMET** | **C0805C100M4GACTU** | **0,099** | https://www.mouser.fr/ProductDetail/KEMET/C0805C100M4GACTU?qs=%2Fha2pyFadugbjD%2F0nhYM2782s7rRfMoqILHRmC9SETYKmh8I1rsM8g%3D%3D |
| **R8** | **RC1206FR-07107RL** | **1** | **Rfilt1 107 ohm CMS 1%** | **Mouser** | **Yageo** | **603-RC1206FR-07107RL** | **0,207** | https://www.mouser.fr/ProductDetail/Yageo/RC1206FR-07107RL?qs=m6lXFsvg5e0q9ku1jkHDgw%3D%3D |
| **R9** | **ERJ-M1WSF4M0U** | **1** | **Rsense1 0.004 ohm CMS 2512 1% Curr Sense AEC-Q200** | **Mouser** | **Panasonic** | **667-ERJ-M1WSF4M0U** | **0,63** | <https://www.mouser.fr/ProductDetail/Panasonic/ERJ-M1WSF4M0U?qs=%2Fha2pyFaduhOx12dbJfB0JH13guvP2GFlAPfhIuCaD5Pa1pmQiSbDQ%3D%3D> |
| **R4** | **RC1206FR-071KL** | **1** | **Rfb1 1Kohm CMS 1%** | **Mouser** | **Yageo** | **603-RC1206FR-071KL** | **0,207** | https://www.mouser.fr/ProductDetail/Yageo/RC1206FR-071KL?qs=HQvIwrFd5cVee6AAIEc2GA%3D%3D |
| **R5** | **RC1206FR-078K45L** | **1** | **Rfb2 8.45 Kohm CMS 1%** | **Mouser** | **Yageo** | **603-RC1206FR-078K45L** | **0,207** | https://www.mouser.fr/ProductDetail/Yageo/RC1206FR-078K45L?qs=bz%2FLPVPAsTJhMcgMYMLcTA%3D%3D |
| **Q2,Q3** | **DMN6075SQ-7** | **2** | **60 V, 2A, RDSon 85 mOhm** | **Mouser** | **Diodes Incorporated** | **621-DMN6075SQ-7** | **0,36** | <https://www.mouser.fr/ProductDetail/Diodes-Incorporated/DMN6075SQ-7?qs=sGAEpiMZZMshyDBzk1%2FWi0w8OutPcgKpJss6F67sL%2FuIQMT%2Fd2Zo6Q==> |
| **C15, C16, C17, C18, C19** | **08053C101MAT2A** | **5** | **MLCC - CMS 25V 100pF X7R 20% Tol** | **Mouser** | **AVX** | **581-08053C101MAT2A** | **0,234** | <https://www.mouser.fr/ProductDetail/AVX/08053C101MAT2A?qs=pJKkPLTLvBjfD9LErl%252Byrw%3D%3D> |
| **C14** | **08053C104MAT2A** | **1** | **CMS 25V .1uF X7R 0805 20% To** | **Mouser** | **AVX** | **581-08053C104M** | **0,144** | <https://www.mouser.fr/ProductDetail/AVX/08053C104MAT2A?qs=iAmswNBQWa3yww%252Bhw1tBQA%3D%3D> |
| **C13** | **CL21B106KOQNNNE** | **1** | **10 µF, 16 V, 0805 [2012 Metric], ± 10%, X7R, CL Series** | **Mouser** | **samsung** | **3013485RL** | **0,313** | <https://fr.farnell.com/samsung-electro-mechanics/cl21b106koqnnne/condensateur-10uf-16v-mlcc-0805/dp/3013485RL> |
| **R10, R11** | **SG73G2ATTD1002D** | **2** | **CMS 0.5% 0.5W AEC-Q200** | **Mouser** | **KOA Speer** | **660-SG73G2ATTD1002D** | **0,231** | <https://www.mouser.fr/ProductDetail/KOA-Speer/SG73G2ATTD1002D?qs=sGAEpiMZZMtlubZbdhIBIFC33eDTuoa6V25eeInASp8%3D> |
| **C22, C23, C24** | **EEU-FR1E101** | [**3**](file:///C:\Users\louis\OneDrive\Bureau\BABE_SPACE\COMPOSANTS\1-INVENTAIRE_BABEBOT\Babe_Inventaire_Doc\Datasheet\EEU-FR1E101_Fiche_technique.pdf) | **100uF, Low ESR** | **Mouser** | **Panasonic** | **667-EEU-FR1E101** | **0,306** | <https://www.mouser.fr/ProductDetail/Panasonic/EEU-FR1E101?qs=Ao3mORb5HCDieoJwtkb8Dw%3D%3D> |
| **C20, C21** | **EEU-FR1E221B** | [**2**](file:///C:\Users\louis\OneDrive\Bureau\BABE_SPACE\COMPOSANTS\1-INVENTAIRE_BABEBOT\Babe_Inventaire_Doc\Datasheet\EEU-FR1E221B_Fiche_technique.pdf) | **220 uF, Low ESR** | **Mouser** | **Panasonic** | **667-EEU-FR1E221B** | **0,486** | <https://www.mouser.fr/Search/Refine?Keyword=eeu-fr1e221b> |

Tableau **3**: BOM du Module de gestion de l’alimentation

3.2.3 BOM du module de veille

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **REF DESIGNATOR** | **PART NUMBER** | **DESCRPTION** | **FOURNISSEUR** | **CONSTRUCTEUR** | **REF FOURNISSEUR** | **PRIX** | **LINK** |
| **U1** |  |  | **Mouser** | **microship** | **579-PIC16F18444I/SO** | **1,04** | <https://www.mouser.fr/ProductDetail/579-PIC16F18444I-SO> |
| **U2** | **RN4020-V/RM123** |  | **Mouser** | **microship** | **579-RN4020-V/RM123** | **7,48** | https://www.mouser.fr/ProductDetail/Microchip-Technology/RN4020-V-RM123/?qs=sGAEpiMZZMu3sxpa5v1qrpr%252BIU7ZCjVxU8GjI%252BKCa7k%3D |
| **U3** | **MAX6457UKD3B+T** | **Circuits de surveillance High-Voltage, Low-Current Voltage Monitors in SOT Packages** | **Mouser** | **Maxim Integrated** | **700-MAX6457UKD3BT** | **3,36** | <https://www.mouser.fr/ProductDetail/Maxim-Integrated/MAX6457UKD3B%2bT?qs=bB8bNThzfotKaP8yzEPd%252BQ%3D%3D> |
| **VR1** | **AP2205-33W5-7** | **Régulateurs de tension LDO CMOS LowCurr** | **Mouser** | **Diodes Incorporated** | **621-AP2205-33W5-7** | **0,373** | <https://www.mouser.fr/ProductDetail/Diodes-Incorporated/AP2205-33W5-7?qs=%2Fha2pyFadugW6XW55GBLU2ziBtiQeCBkTSL6SOghddbRIY68LE8AmA%3D%3D> |
| **Q1** | **DMP34M4SPS-13** | **MOSFET MOSFET BVDSS: 25V-30V** | **Mouser** | **Diodes Incorporated** | **621-DMP34M4SPS-13** | **0,954** | <https://www.mouser.fr/ProductDetail/Diodes-Incorporated/DMP34M4SPS-13?qs=W0yvOO0ixfEN%252BZNUBfocww%3D%3D> |
| **Q2** | **RS1E240BNTB** |  | **Mouser** | **ROHM Semiconductor** | **755-RS1E240BNTB** | **0,423** | <https://www.mouser.fr/ProductDetail/ROHM-Semiconductor/RS1E240BNTB?qs=npTsUczJOtN4zVwzi0krSQ%3D%3D> |
| **Q3** | **PMV65XPER** | **Canal P, 20 V, 2.8 A, 0.067 ohm, TO-236AB, Montage en surface** | **Mouser** | **NEXPERIA** | **771-PMV65XPER** | **0,33** | <https://www.mouser.fr/ProductDetail/Nexperia/PMV65XPER/?qs=%2Fha2pyFaduhPk0w01jUR9OamqyEwTvT0r8zdRxf7lcpxv%2FTPA%252BYmAg%3D%3D> |
| **Q4** | **NX7002AKVL** | **Canal N, 60 V, 190 mA, 3 ohm, TO-236AB,** | **Mouser** | **NEXPERIA** | **771-NX7002AKVL** | **0,11** | <https://www.mouser.fr/ProductDetail/Nexperia/NX7002AKVL/?qs=1GAP6L74MtTHifGD7E%252BJaw%3D%3D> |
| **Q5** | **DMN6075SQ-7** | **60 V, 2A, RDSon 85 mOhm** | **Mouser** | **Diodes Incorporated** | **621-DMN6075SQ-7** | **0,36** | <https://www.mouser.fr/ProductDetail/Diodes-Incorporated/DMN6075SQ-7?qs=sGAEpiMZZMshyDBzk1%2FWi0w8OutPcgKpJss6F67sL%2FuIQMT%2Fd2Zo6Q==> |
| **D2** | **APTD2012LCGCK** | **CMS 0805 Dome Lens LED Green- 570nm, GREEN/1.9V/2mA** | **Mouser** | **Kingbright** | **604-APTD2012LCGCK** | **0,347** | <https://www.mouser.fr/ProductDetail/Kingbright/APTD2012LCGCK?qs=AQlKX63v8RvNHN%2F9EGvNTg%3D%3D> |
| **D1** | **APTD2012LSURCK** | **CMS 0805 Dome Lens LED Red- 630nm, RED/1.75V/2mA** | **Mouser** | **Kingbright** | **604-APTD2012LSURCK** | **0,33** | <https://www.mouser.fr/ProductDetail/Kingbright/APTD2012LSURCK?qs=AQlKX63v8RvkMadNvwbkxw%3D%3D> |
| **D3** | **SD0805S020S1R0** | **Diodes et redresseurs Schottky 20V 1A SIZE 0805** | **Mouser** | **AVX** | **581-SD0805S020S1R0** | **0,305** | https://www.mouser.fr/ProductDetail/AVX/SD0805S020S1R0/?qs=sGAEpiMZZMtbRapU8LlZD%2Fe6n6Rl3x6ukk4z6DMRPLY%3D |
| **R1** | **ERJ-PB6D5363V** | **Anti-Surge Res. 0.5% --> Vtrip=7,42V, Vtrip/2= 3.71V** | **Mouser** | **Panasonic** | **667-ERJ-PB6D5363V** |  | <https://www.mouser.fr/ProductDetail/panasonic/erj-pb6d5363v/?qs=eFOtW6xYNWHIaS37S6Evbg==&countrycode=DE&currencycode=EUR> |
| **R2** | **ERJ-PB6D1003V** | **CMS 0805 Anti-Surge Res. 0.5%, 100Kohm** | **Mouser** | **Panasonic** | **667-ERJ-PB6D1003V** | **0,212** | https://www.mouser.fr/ProductDetail/panasonic/erj-pb6d1003v/?qs=eFOtW6xYNWEPVUrInTQWDA==&countrycode=DE&currencycode=EUR |
| **R3** | **ERJ-PB6B2001V** | **Résistances à couches épaisses - CMS 0805 Anti-Surge Res. 0.1%, 2Kohm** | **Mouser** | **Panasonic** | **667-ERJ-PB6B2001V** | **0,305** | https://www.mouser.fr/ProductDetail/Panasonic/ERJ-PB6B2001V/?qs=eFOtW6xYNWES%2FVrKZETlCA%3D%3D |
| **R4** | **ERA-6AEB102V** | **Résistances à couches minces - CMS 0805 1/8W 1Kohms** | **Mouser** | **Panasonic** | **667-ERA-6AEB102V** | **0,305** | <https://www.mouser.fr/ProductDetail/Panasonic/ERA-6AEB102V/?qs=1VWA5LkbEaodAYxyfKqGQw%3D%3D> |
| **R5** | **ERJ-6GEYJ152V** | **Résistances à couches épaisses - CMS 0805 1.5Kohms 5% AEC-Q200** | **Mouser** | **Panasonic** | **667-ERJ-6GEYJ152V** | **0,085** | <https://www.mouser.fr/ProductDetail/Panasonic/ERJ-6GEYJ152V/?qs=DjdiYhHdu0uUT7hENu7OgA%3D%3D> |
| **R6** | **ERJ-6GEYJ562V** | **Résistances à couches épaisses - CMS 0805 5.6Kohms 5% AEC-Q200** | **Mouser** | **Panasonic** | **667-ERJ-6GEYJ562V** | **0,085** | <https://www.mouser.fr/ProductDetail/Panasonic/ERJ-6GEYJ562V/?qs=YMBIwoNQYdWpsxeKVZReog%3D%3D> |
| **R7, R8, R9, R10, R11, R12, R13, R14, R15** | **SG73G2ATTD1002D** | **CMS 0.5% 0.5W AEC-Q200** | **Mouser** | **KOA Speer** | **660-SG73G2ATTD1002D** | **0,231** | <https://www.mouser.fr/ProductDetail/KOA-Speer/SG73G2ATTD1002D?qs=sGAEpiMZZMtlubZbdhIBIFC33eDTuoa6V25eeInASp8%3D> |
| **C1** | **EMK212B7224KG-T** | **Condensateurs céramique multicouches MLCC - CMS 0805 16VDC 0.22uF 10% X7R** | **Mouser** | **Taiyo Yuden** | **963-EMK212B7224KG-T** | **0,144** | <https://www.mouser.fr/ProductDetail/Taiyo-Yuden/EMK212B7224KG-T/?qs=I6KAKw0tg2xmuGTM%2FarU4w%3D%3D> |
| **C2** | **C0805C105K4RACAUTO** | **MLCC - CMS 16V 1uF X7R 0805 10% AEC-Q200** | **Mouser** | **KEMET** | **80-C0805C105K4RAUTO** | **0,152** | <https://www.mouser.fr/ProductDetail/KEMET/C0805C105K4RACAUTO?qs=OHu5RF0w3QprP0D5VzlqaQ%3D%3D> |
| **C3** | **EMK212BJ225KG-T** | **Condensateurs céramique multicouches MLCC - CMS 0805 16VDC 2.2uF 10% X5R** | **Mouser** | **Taiyo Yuden** | **963-EMK212BJ225KG-T** | **0,127** | <https://www.mouser.fr/ProductDetail/Taiyo-Yuden/EMK212BJ225KG-T/?qs=I6KAKw0tg2wY60Dl46zP7Q%3D%3D> |
| **C4, C5** | **CC0805KRX7R7BB104** | **MLCC - CMS 100nF 16V X7R 10%** | **Mouser** | **Yaego** | **603-CC805KRX7R7BB104** | **0,085** | https://www.mouser.fr/ProductDetail/603-CC805KRX7R7BB104 |
| **C6** | **GRM21BR71C475ME51K** | **MLCC - CMS 4.7UF 16V 20% 0805** | **Mouser** | **Murata Electronics** | **81-GRM21BR71C475ME1K** | **0,152** | <https://www.mouser.fr/ProductDetail/Murata-Electronics/GRM21BR71C475ME51K?qs=QzBtWTOodeV0aa0oJHX5rA%3D%3D> |
| **C7** | **EMK212ABJ106KG-T** | **MLCC - CMS 10uF 16V X5R +/-10% 0805 Gen Purp** | **Mouser** | **Taiyo Yuden** | **963-EMK212ABJ106KG-T** | **0,144** | https://www.mouser.fr/ProductDetail/Taiyo-Yuden/EMK212ABJ106KG-T?qs=vF%252B8ahGVilfgbAjT%2FgKCVQ%3D%3D |
| **LS1** | **CMI-9605IC-0380T** | **Indicateurs audio et sonneries piézo-électriques buzzer, 9.6 mm x 5 mm deep, M, 3 VDC, 80 dB, Through Hole, Intergrated Circuit Audio Indicator** | **Mouser** | **CUI Devices** | **490-CMI-9605IC-0380T** | **1,19** | <https://www.mouser.fr/ProductDetail/490-CMI-9605IC-0380T> |
| **S1, S2** | **1241.1600.11** | **Commutateurs à bouton-poussoir** | **Mouser** | **Schurter** | **693-1241.1600.11** | **2,1** | <https://www.mouser.fr/ProductDetail/Schurter/1241160011/?qs=%2Fha2pyFaduiDEhge8r6YWiYDng8YgcjDU8%252BD6d9IgSu83C7jRGK%2FYg%3D%3D> |
| **J1, J2** | **MC000042** | **Borniers fil-à-carte : 5MM, 12-24 AWG** | **Mouser** | **multicomp** | **2008012** | **0,476** | <https://fr.farnell.com/multicomp/mc000042/bornier-ci-noir-5mm-mod-45deg/dp/2008012> |
| **J3** | **35363-0560** | **Embases et logements de câbles , HEADER ANGLE CLIPS 2A 5P** | **Mouser** | **AP2205-33W5-7** | **538-35363-0560** | **0,39** | https://www.mouser.fr/ProductDetail/Molex/35363-0560?qs=%2Fha2pyFadui0y%2FMIVKpUU%2FVUVPQflvPlS%252BeUD1Idayg%3D |

Tableau **4**: BOM du Module de veille (**C2**)