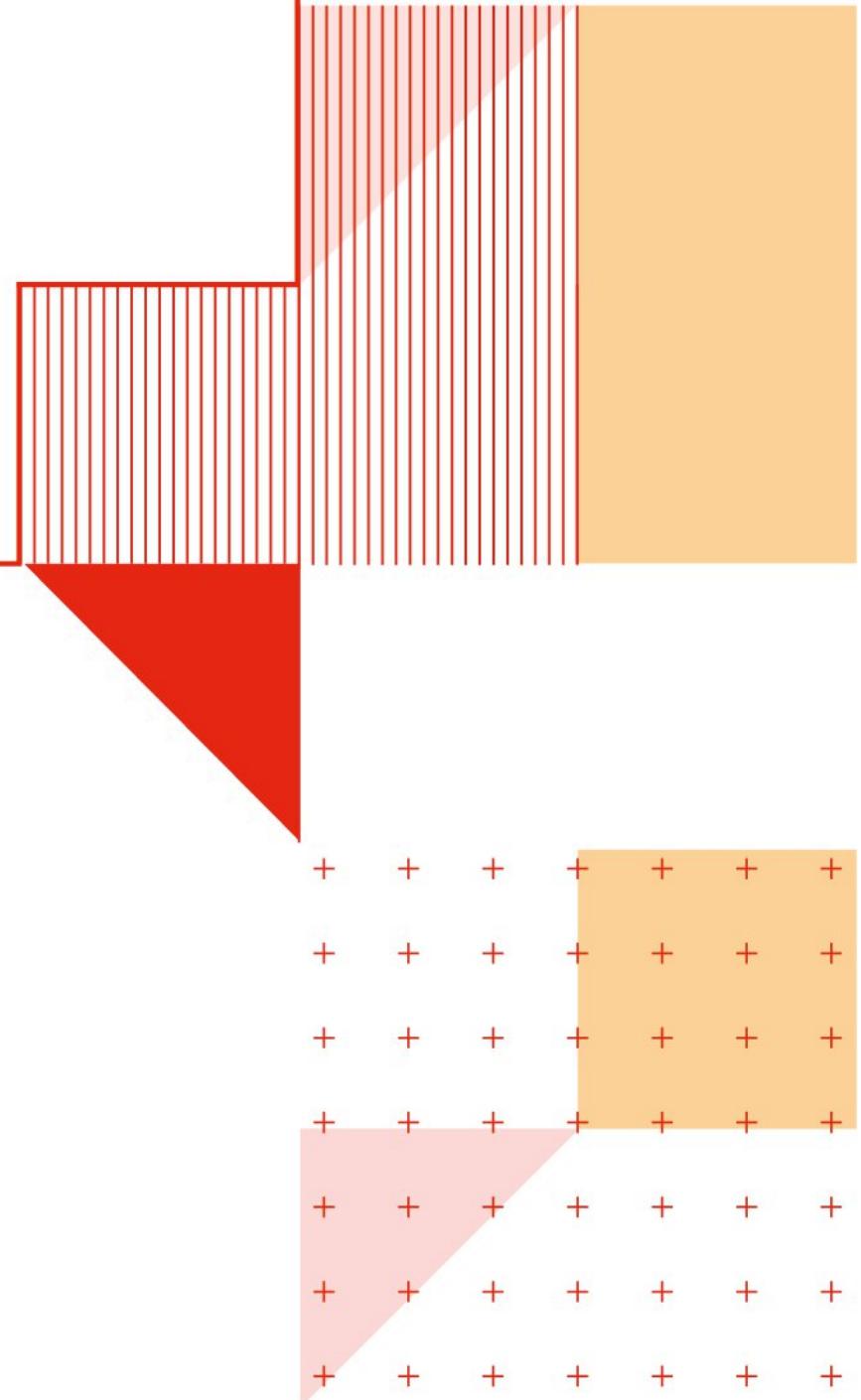


Dumber V3

Manuel de référence



SOMMAIRE

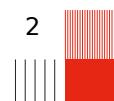
Introduction	3
Fiche technique	4
Vue d'ensemble	4
Comportement – Machine à état	5
Fonctionnement interne du logiciel	7
Liste des commandes	8
Format d'une commande	9
Description détaillée des commandes	9
Afficheur – Signification des animations	16
Schémas	18
Dimensions	20
Données techniques	20

LISTE DES FIGURES

Figure 1 - Machine à état	5
Figure 2 - Usage principal du robot	6
Figure 3 - Schéma partie puissance	18
Figure 4 - Schéma partie CPU	19
Figure 5 - Dimensions	20

LISTE DES TABLEAUX

Tableau 1 - Liste des commandes	8
Tableau 2 - Liste des animations possible sur l'afficheur	17



INTRODUCTION

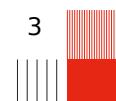
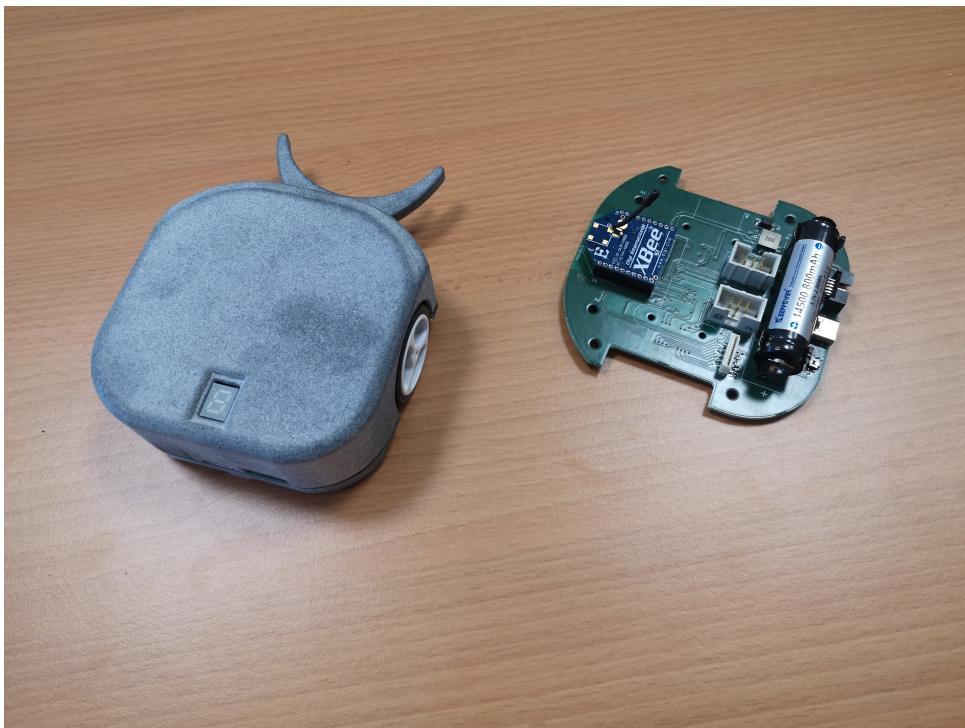
Dumber est un projet dont la première version remonte à 2013, destiné à être utilisé dans des travaux pratiques liés à l'enseignement du temps réel au département GEI de l'INSA Toulouse. Il prend la forme d'un robot contrôlable par liaison RF, d'où le nom du projet, vu que le robot n'a aucun capteur et se contente d'exécuter les ordre de mouvement, sans prise de décision.

Depuis la première version du projet, les robots se sont miniaturisés et sophistiqués (intégration d'un chargeur de batterie embarqué, mécanismes d'économie d'énergie ...) sans pour autant gagner perdre l'essence du projet, à savoir devenir intelligent.

Ce manuel est à la fois un manuel utilisateur et un manuel technique décrivant les spécifications actuelles et informations techniques liées à sa conception.

Bonne lecture.

Sébastien DI MERCURIO



FICHE TECHNIQUE

Le robot Dumber, dans sa version 3, possède les caractéristiques techniques suivantes :

Microcontrôleur STM32L071 cadencé à 6 Mhz (128 Mb de Flash, 20 Mb de RAM)

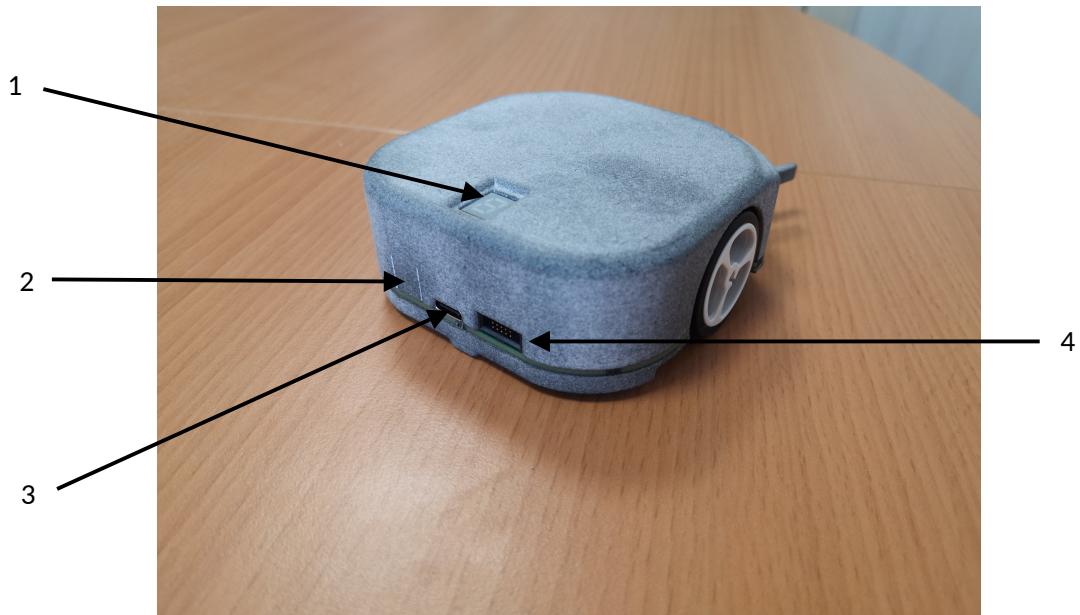
Contrôleur de charge batterie STL03

Charge via câble USB-C (charge uniquement)

Batterie Li-ION, 1 élément, 3.7 V

Module RF 2.4 Ghz (Xbee), norme 802.15.4

VUE D'ENSEMBLE



1	Afficheur	3	Connecteur (chargeur)	USB-C
2	Bouton marche / arrêt	4	Connecteur de reprogrammation	



COMPORTEMENT – MACHINE À ÉTAT

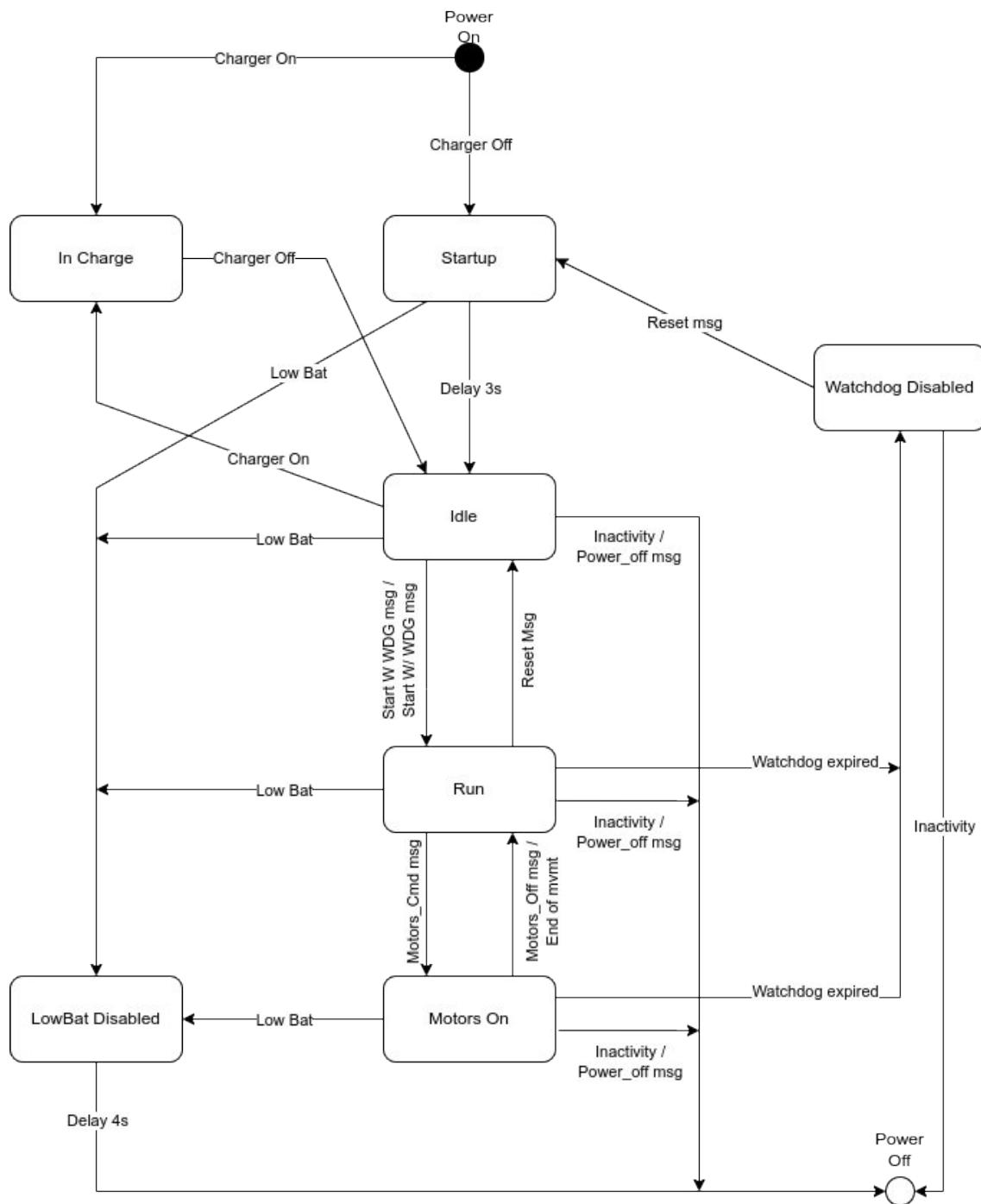
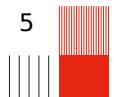


Figure 1 - Machine à état



PRISE EN MAIN ET UTILISATION

Le robot se démarre en appuyant brièvement sur le bouton marche/arrêt (¹). Le robot indique alors, via une animation sur l'afficheur (²), le niveau de batterie, pendant 3 secondes, puis se met en attente d'une commande dans l'état IDLE (³).

Sans autre action, le robot s'éteint automatiquement au bout de 2 minutes d'inactivité. Pour éteindre le robot de manière manuelle, un appui bref sur le bouton marche/arrêt est suffisant.

Le schéma classique d'utilisation du robot suit la procédure suivante :

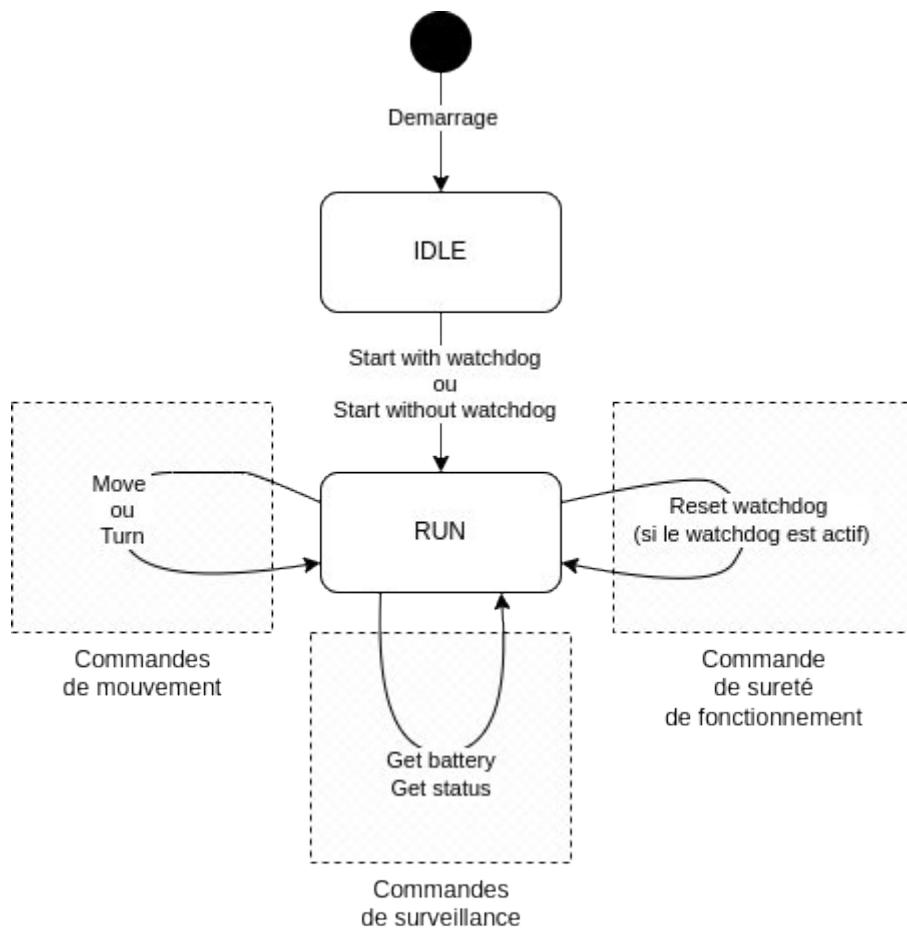
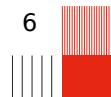


Figure 2 - Usage principal du robot

¹ Voir [Vue d'ensemble](#)

² Voir Annexe A - Afficheur

³ Voir [Comportement - Machine à état](#)



FONCTIONNEMENT INTERNE DU LOGICIEL

Le logiciel du robot s'appuie sur un OS temps réel, FreeRTOS. Le logiciel est découpé en plusieurs tâches, et la communication entre tâches se fait par passage de message, via des boîtes aux lettres (mailbox).

Chaque module est composé d'un gestionnaire (handler), associé à une boîte aux lettres et une machine à état chargée de traiter les messages reçus et le comportement du module.

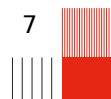
Si le module est un module « driver », associé à un périphérique matériel sous interruption, celui-ci possède en plus une tâche associé aux routines d'interruption et la synchronisation se fait alors via sémaphore binaire.

Le logiciel possède les tâches suivantes :

- Application (handler) : tâche principale en charge du comportement global du robot
- Leds (handler) : tâche en charge de la boîte aux lettres associé au contrôleur d'affichage
- Leds (driver) : tâche périodique en charge de l'animation de l'afficheur
- Moteurs (handler) : tâche en charge de la boîte aux lettres associé au pilote des moteurs
- Moteurs (driver) : tâche périodique dédié à l'asservissement des moteurs
- Batterie (handler) : tâche en charge de la gestion de la batterie
- Xbee Tx (handler) : tâche en charge de la boîte aux lettres du module RF en émission
- Xbee Rx (driver) : tâche en charge de la réception des messages.

En plus de ces tâches, 2 timers logiciels sont définis :

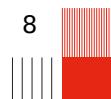
- Timer application : en charge du décompte de la période d'inactivité ainsi que du watchdog
- Timer batterie : en charge de la surveillance de l'état du chargeur



LISTE DES COMMANDES

Nom de la commande	Codage	États où la commande est acceptée	Description
PING	p	TOUS	Sert uniquement à vérifier que le robot est allumé
RESET	r	IDLE RUN WATCHDOG DISABLED	Réinitialise le robot dans l'état IDLE, notamment après un dépassement de watchdog
START WITH WATCHDOG	w	IDLE	Active le robot avec le watchdog
START WITHOUT WATCHDOG	u	IDLE	Active le robot sans démarrer le watchdog
RESET WATCHDOG	w	RUN MOTORS ON	Acquitte le watchdog
MOVE	M	RUN	Demande un déplacement linéaire (avant ou arrière)
TURN	T	RUN	Demande une rotation (gauche ou droite)
GET STATUS	b	IDLE RUN MOTORS ON	Obtient si le robot est en mouvement ou non
GET BATTERY	v	IDLE RUN MOTORS ON	Obtient le niveau de charge de la batterie
GET VERSION	V	TOUS	Obtient la version du firmware
POWEROFF	z	TOUS	Éteint le robot.

Tableau 1 - Liste des commandes



FORMAT D'UNE COMMANDE

Les commandes suivent le format suivant :

Commande	Paramètres (optionnel)	Checksum	0x0D
1 char	n chars	1 char	1 char

Le checksum se calcule en faisant l'opération de OU exclusif entre tous les caractères avant le checksum. Si le checksum vaut 0x0D (CR), on rajoute 1 pour le différencier du caractère de fin de trame.

Les paramètres, si présent, commencent toujours par '=' suivi directement par la valeur. Les valeur sont sous forme lisible (ASCII), pas binaire.

Exemples

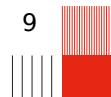
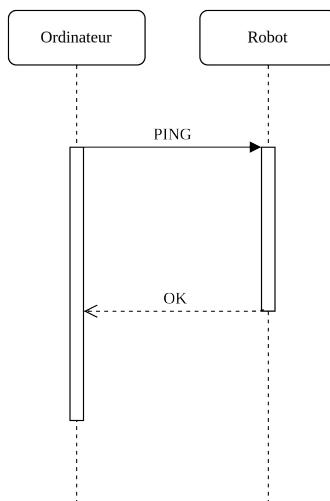
PING → p p <0x0D>
MOVE de 25 cm → M = 2 5 w <0x0D>

DESCRIPTION DÉTAILLÉE DES COMMANDES

PING

Commande :	p
Paramètres :	Aucun
États où la commande est acceptée :	Tous
États après la commande :	Pas de changement
Valeur de retour :	Aucune

Exemple :



RESET

Commande :

r

Paramètres :

Aucun

États où la commande est acceptée :

RUN, WATCHDOG DISABLED

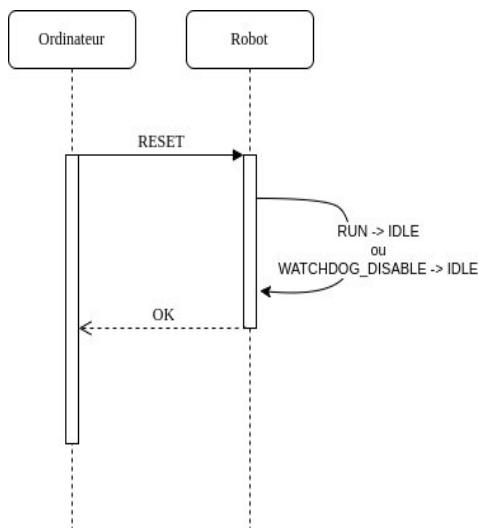
États après la commande :

IDLE

Valeur de retour :

Aucune

Exemple :



START WITH WATCHDOG

START WITHOUT WATCHDOG

Commandes :

W (pour START_WITH_WATCHDOG)

Paramètres :

u (pour START_WITHOUT_WATCHDOG)

États où la commande est acceptée :

Aucun

États après la commande :

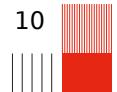
IDLE

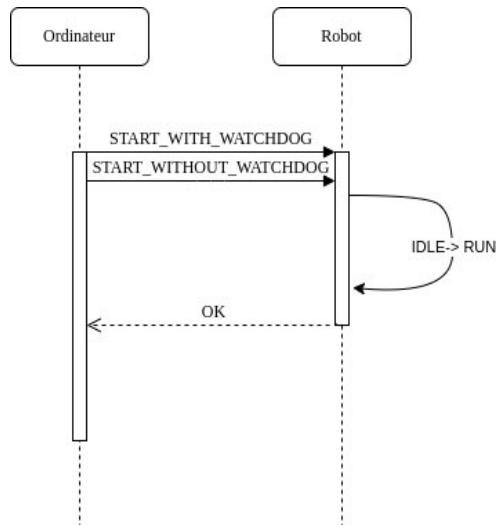
Valeur de retour :

RUN

Aucune

Exemple :





RESET WATCHDOG

Commande :

w

Paramètres :

Aucun

États où la commande est acceptée :

RUN, MOTORS ON

États après la commande :

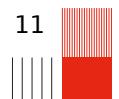
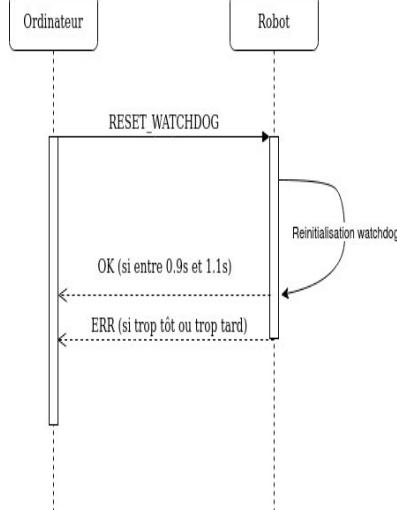
Pas de changement si succès

Valeur de retour :

WATCHDOG_DISABLED si échec

Aucune

Exemple :



MOVE

Commande :

Paramètres :

M

Valeur du déplacement en cm
entre -32767 et 32767

Valeur positif pour aller en avant

Valeur négative pour aller en arrière

0 pour arrêter les moteurs

États où la commande est acceptée :

RUN

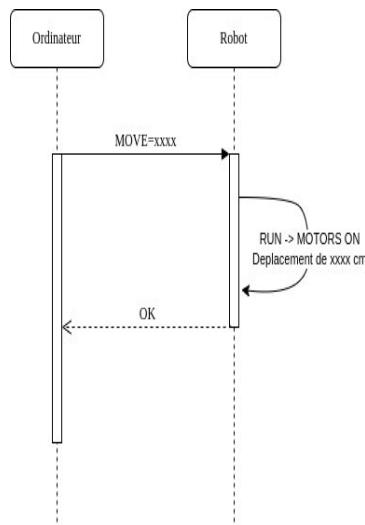
États après la commande :

MOTORS ON

Valeur de retour :

Aucune

Exemple :



TURN

Commande :

T

Paramètres :

Valeur de la rotation en degrés

entre -32767 et 32767

Valeur positif pour aller à gauche

Valeur négative pour aller à droite

0 pour arrêter les moteurs

États où la commande est acceptée :

RUN

États après la commande :

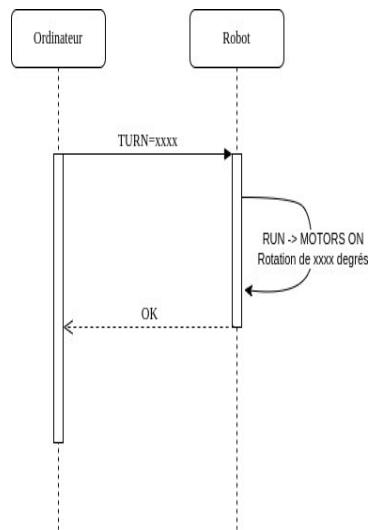
MOTORS ON

Valeur de retour :

Aucune

Exemple :





GET STATUS

Commande :

b

Paramètres :

Aucun

États où la commande est acceptée :

RUN, MOTORS ON

États après la commande :

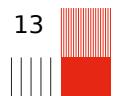
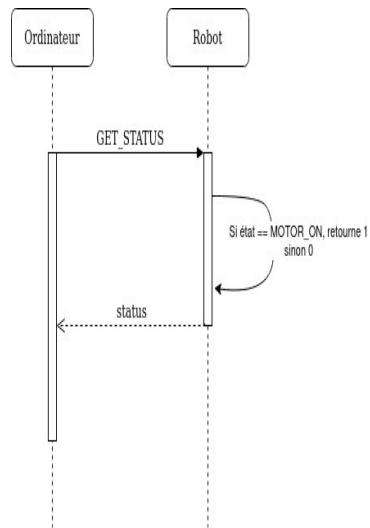
Pas de changement

Valeur de retour :

0 si les moteurs sont à l'arrêt

1 si un mouvement est en cours

Exemple :



GET BATTERY

Commande :

v

Paramètres :

Aucun

États où la commande est acceptée :

RUN, WATCHDOG DISABLED, MOTORS ON

États après la commande :

Pas de changement

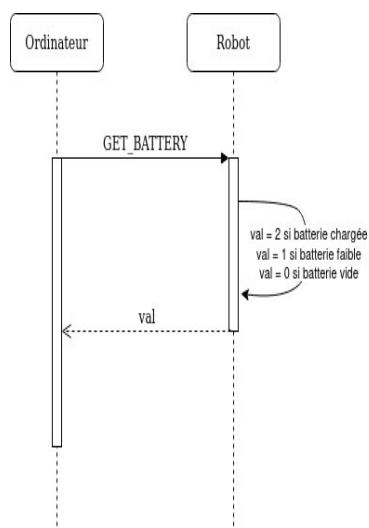
Valeur de retour :

0 si la batterie est fortement déchargée

1 si la batterie est chargée à moitié

2 si la batterie est pleine

Exemple :



GET VERSION

Commande :

v

Paramètres :

Aucun

États où la commande est acceptée :

Tous

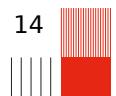
États après la commande :

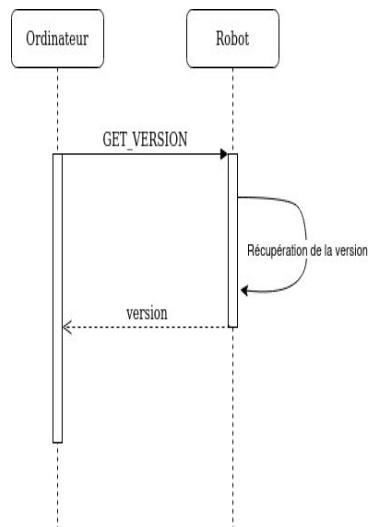
Pas de changement

Valeur de retour :

Chaine représentant la version du firmware
sous la forme VV.vv avec VV la version majeur
et vv la version mineur

Exemple :

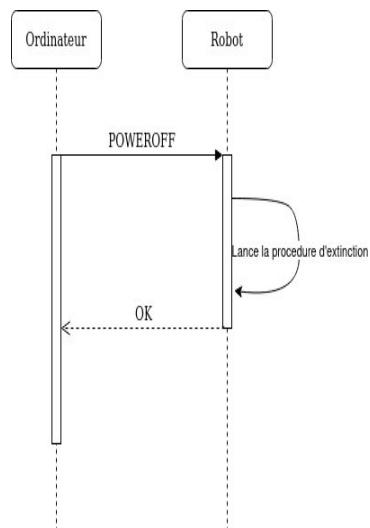




POWEROFF

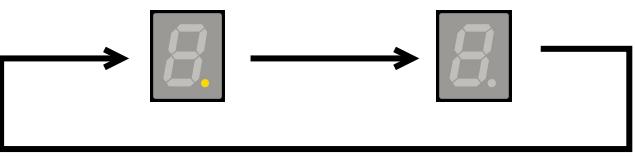
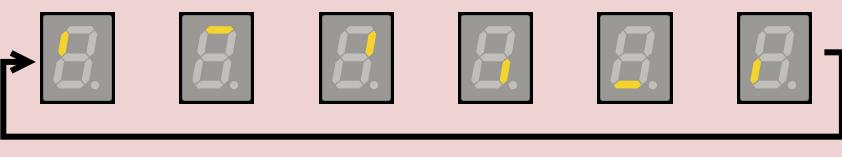
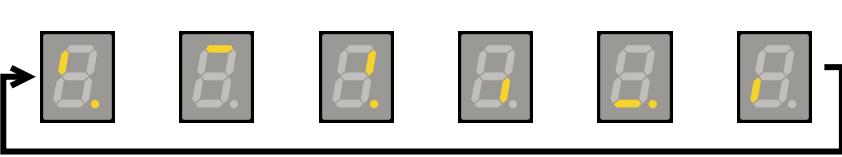
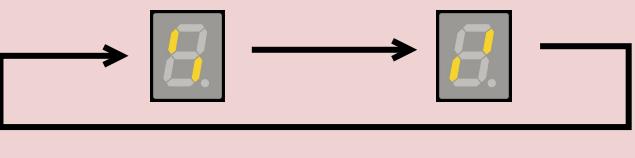
Commande :	z
Paramètres :	Aucun
États où la commande est acceptée :	Tous
États après la commande :	ARRET
Valeur de retour :	Aucune

Exemple :



AFFICHEUR – SIGNIFICATION DES ANIMATIONS

Le tableau suivant recense les différents affichages et animation qui peuvent apparaître sur le petit afficheur du robot.

Signification	État	Animation
État de la batterie	Démarrage du robot	 Batterie faible
		 Batterie moyenne
		 Batterie chargée
Attente d'une commande « Start »	IDLE	
Robot activé sans watchdog	RUN MOTOR ON	
Robot activé avec watchdog	RUN MOTOR ON	
Watchdog expiré	WATCHDOG DISABLED	
Batterie trop faible	TOUS	



Batterie en charge	CHARGE ON		Batterie chargée
			Batterie faible
			Batterie moyenne
Erreurs d'allocation mémoire (Erreur 1)	TOUS		
Erreur ADC (Erreur 2)	TOUS		
Erreur chargeur (Erreur 3)	CHARGE ON		

Tableau 2 - Liste des animations possible sur l'afficheur



SCHÉMAS

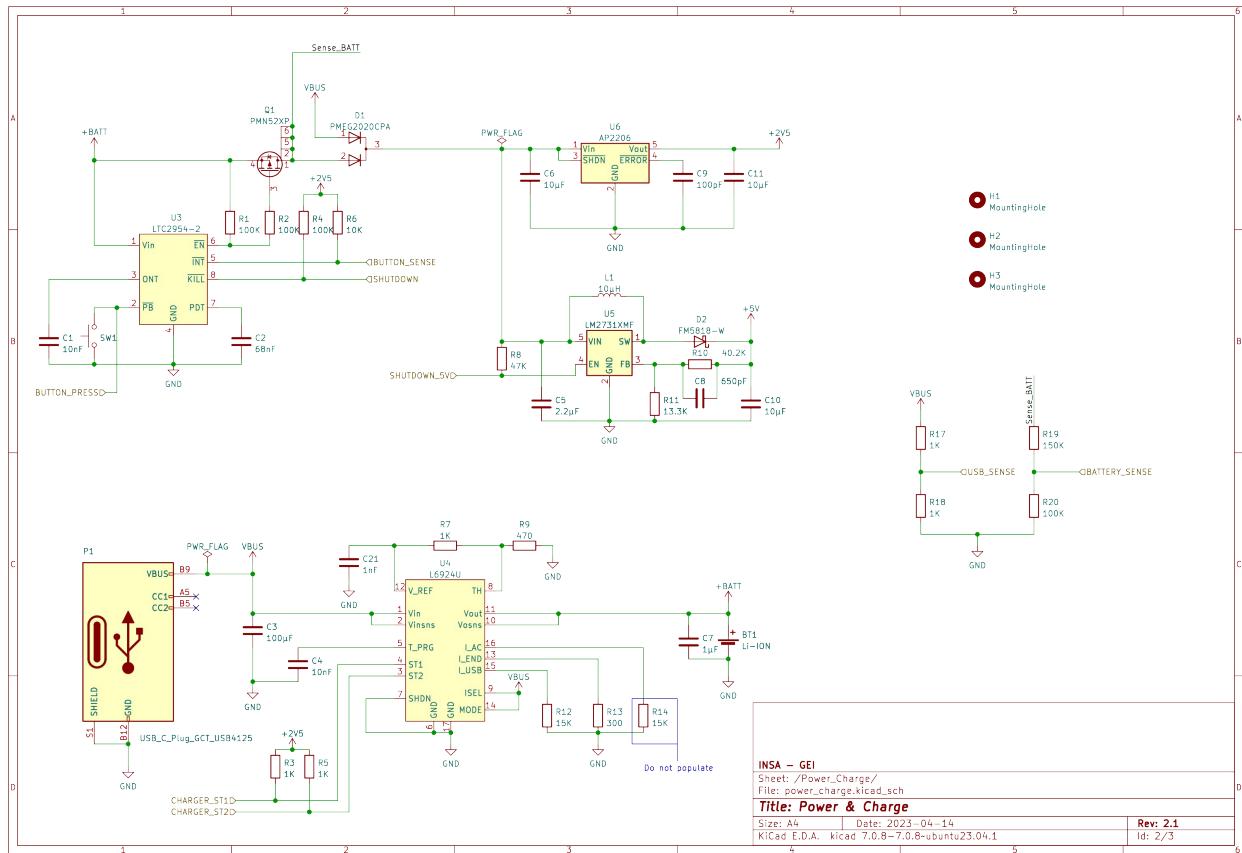
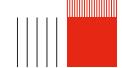


Figure 3 - Schéma partie puissance



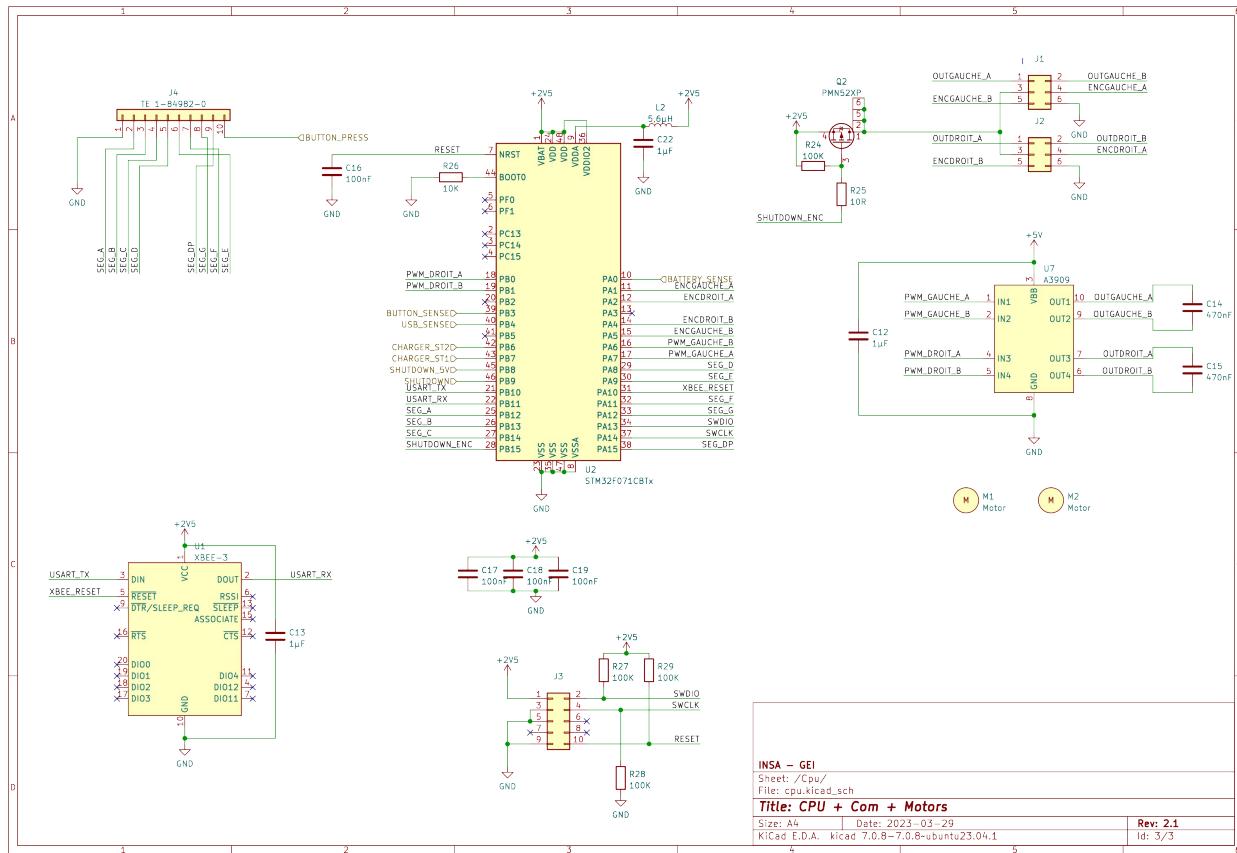


Figure 4 - Schéma partie CPU

DIMENSIONS

ddd

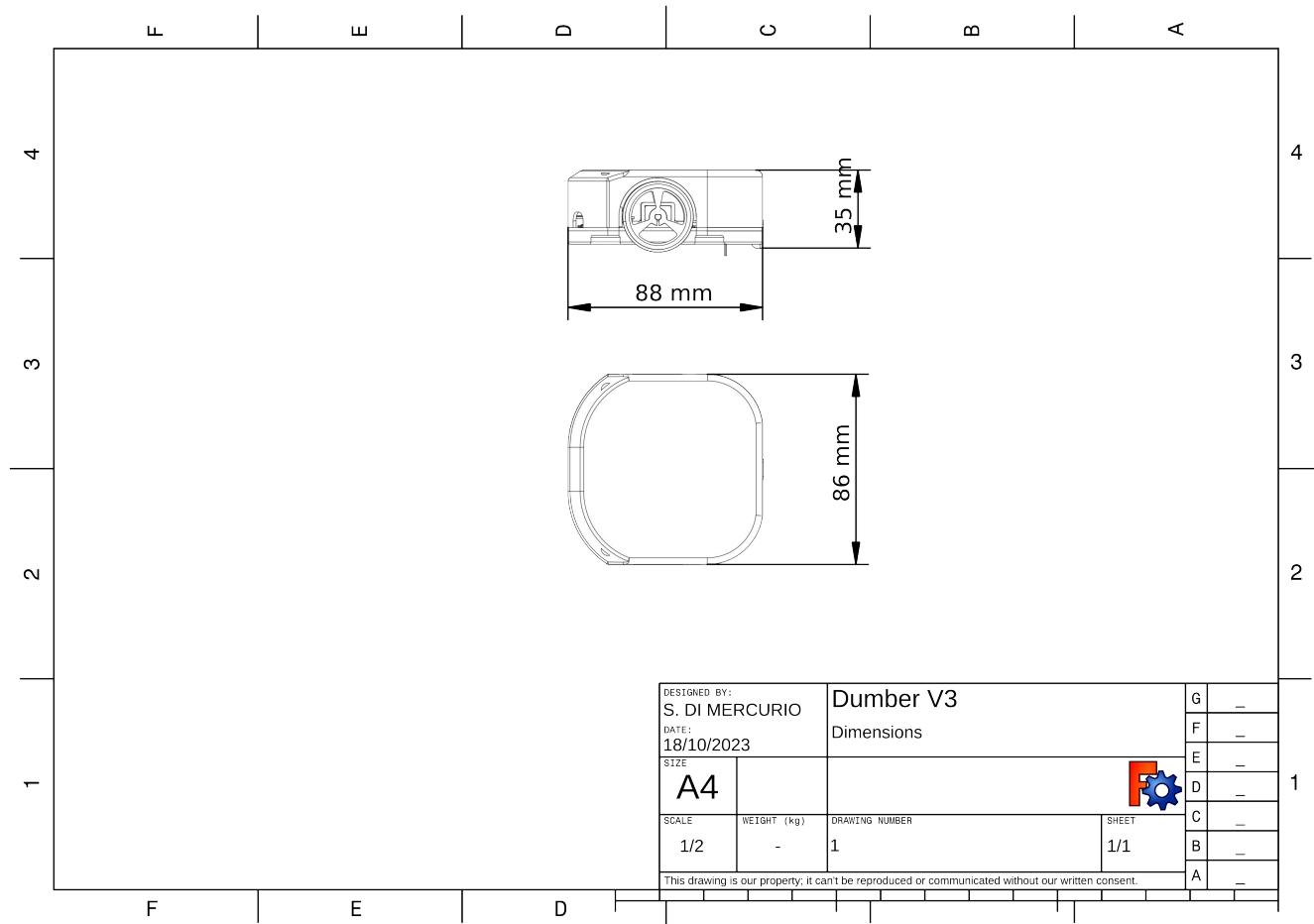


Figure 5 - Dimensions

DONNÉES TECHNIQUES

Dimension (Lxlxh): 88 mm x 86 mm x 35 mm

Poids : environ 120 gr

Autonomie :

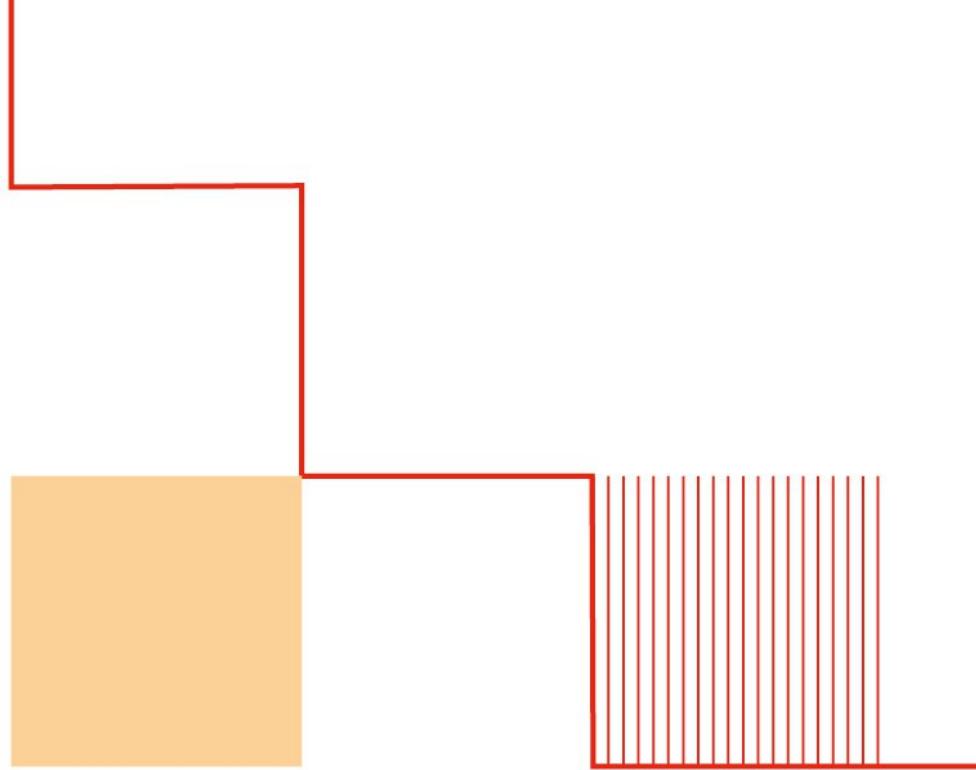
En veille : environ 1 semaine

En mouvement : environ 1h

Portée de la liaison RF : 10 m, en vue directe

Vitesse : 3 cm / s





INSA TOULOUSE

135 avenue de Rangueil
31400 Toulouse

Tél : + 33 (0)5 61 55 95 13

www.insa-toulouse.fr

