DUMBER

MANUEL TECHNIQUE

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET INFORMATIQUE

Juin 2020

Auteur : S. DI MERCURIO

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

TABLE DES MATIÈRES

1 - Présentation

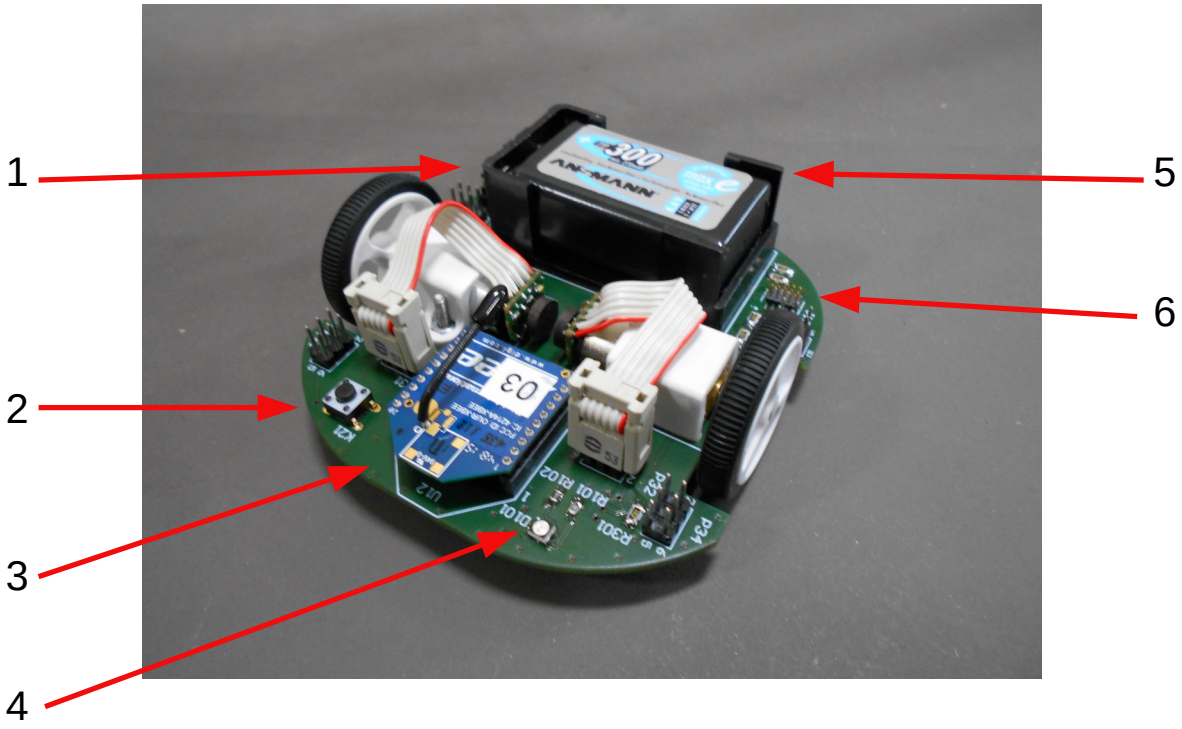
Le robot Dumber est un robot prévu pour les enseignements de Temps Réel. Comme son nom l'indique, le robot n'a aucune intelligence propre et ne fait que répondre aux ordres transmit via la liaison sans fil.

Le robot permet de reconnaître une dizaine d'ordre simples (activer les moteurs, détecter la balle, renvoyer la tension de la batterie, avancer, tourner …).

L'objectif initial du robot est d'être piloté par une ordinateur à distance, ordinateur récupérant la position du robot via une camera pour former un système bouclé.

Le robot possède un minimum d'autonomie grâce à ses moteurs asservis en position.

2 - Description détaillées du robot2



1- Connecteur pour chargeur

2- Bouton Marche/Arret

3- Émetteur RF XBEE

4- Voyant d'activité

5- Batterie NIMH 8,4V

6- Connecteur de programmation

2.1 - Connecteur de programmation 6 (J101)

Le connecteur 6 (J101) sert à brancher l'outil de programmation du micro. Le brochage est le suivant :

Ligne	Signal
1	VCC
2	TMS
3	GND
4	TCK
5	GND
6	TDO
7	GND
8	TDI
9	GND
10	nRST

2.2 - Connecteur batterie 5 (B21)

Le connecteur 5 (B21) sert à brancher la batterie. Le brochage est le suivant :

Ligne	Signal
1	GND
2	+9V

2.3 - Connecteur chargeur 1 (J21)

Le connecteur 1 (J21) sert à brancher le chargeur de batterie. Le brochage est le suivant :

Ligne	Signal
1	Vbat
2	GND
3	Charger Detect

2.4 - Connecteurs moteurs (P31 et P32)

Les connecteurs P31 et P32 servent pour connecter le moteur gauche et droit. Le brochage est le suivant :

Ligne	Signal
1	Moteur +
2	Moteur -
3	3V_Encodeurs
4	Encodeurs_PhA
5	Encodeurs_PhB
6	GND_Encodeurs

2.5 - Interrupteur marche/arrêt 2 (K21)

L'interrupteur 2 (K21) sert à la mise en marche et à l'arrêt du robot. A noter que le robot peut de lui même s'éteindre, notamment en cas de batterie déchargée ou d'inactivité.

2.6 - XBEE 3 (U12)

Le module XBEE se branche sur le double connecteur 10 pts marqué U12. Une sérigraphie sur le PCB indique l'orientation de l'antenne.

Ligne	Signal
1	VCC
2	TX
3	RX
5	Reset
10	GND
Autres	N/C

3 - Utilisation

Lors de la première utilisation, il convient de vérifier que la batterie est correctement branchée au robot. Le démarrage et l'arrêt du robot se font grâce au bouton poussoir 2 qui se trouve à l'avant du robot.

Un appui bref sur le bouton marche/arrêt met le robot sous tension. Le voyant d'activité 4 à l'avant du robot se met à clignoter lentement. Si le voyant clignote rapidement, la batterie est faible et il faut la recharger. Si le voyant ne s'allume même pas, la batterie est totalement déchargée et doit la aussi être rechargée. Il faut alors brancher le chargeur fourni avec le robot dans le connecteur 1 (J21)

Lorsque le robot est sous tension, un appui bref sur le bouton marche/arrêt éteint le robot: le voyant d'activité s'éteint, quelque soit la phase de fonctionnement du robot.

Lorsque la connexion est établie avec le PC, le voyant d'activité se met à s'allumer de

manière continue. Lorsque la connexion est à nouveau interrompue, le voyant d'activité se remet à clignoter lentement.

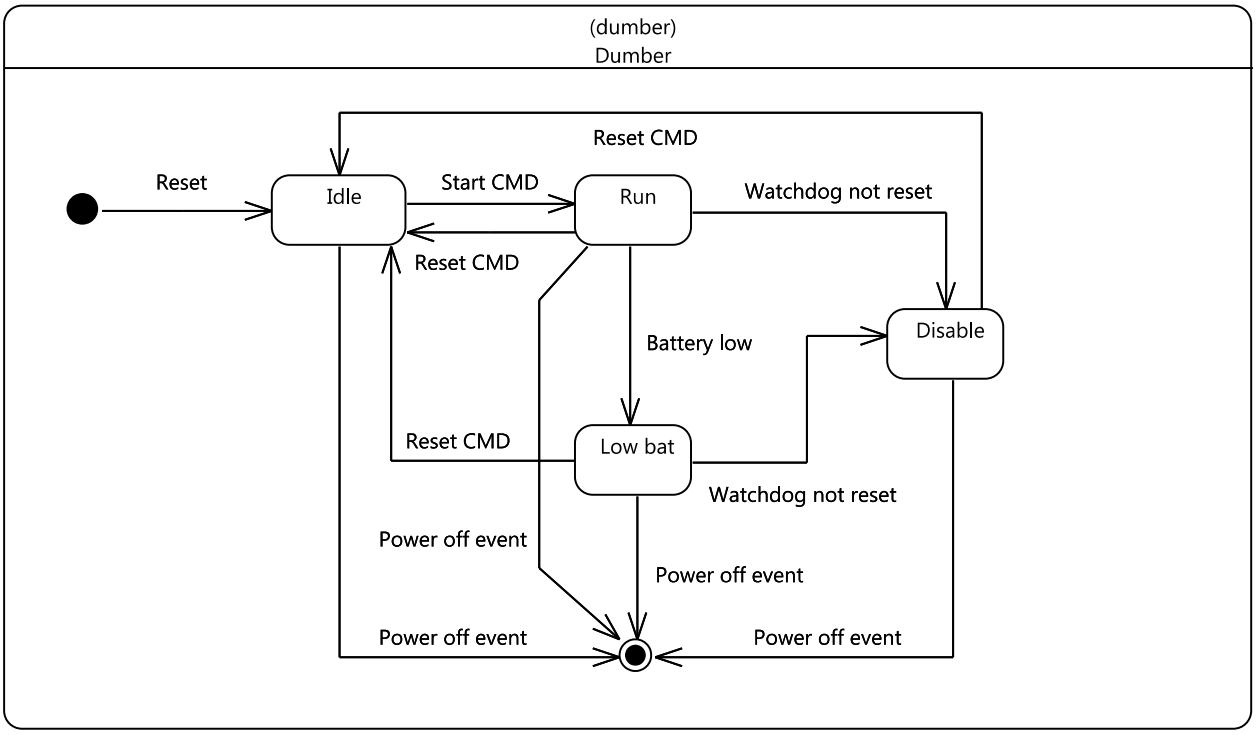
Pour que la connexion s’établisse, il faut bien entendu utiliser le boîtier Raspberry correspondant au numéro du robot (indiqué sur le dessus du robot).

4 - Fonctionnement

4.1 - Machine d’état du robot

Le robot peut être dans 4 états différents:

- `IDLE` : le robot ne fait rien et attend une commande de démarrage. Il rejette la plupart des commande
- `RUN` : le robot accepte l'ensemble des commandes
- `LOW BAT` : Identique au mode `RUN` mais lorsque la batterie faible. La vitesse des moteurs est réduite.
- `DISABLE` : atteint lorsque le watchdog n'a pas été correctement remis à zéro



Les événements « Power off » correspondent à:

- appui bref sur le bouton marche/arrêt (7)
- tension batterie inférieure à 6,1V
- aucune commande reçue depuis 2 minutes

4.2 - Format des commandes

La communication entre le robot et le PC se fait via une liaison sans fil conforme au

protocole 802.15.4. Elle est assuré par des modules XBEE présent sur le robot et dans le boîtier de communication. Pour que la communication se fasse, le boîtier possédant le même numéro que le robot doit être utilisé.

La liaison sans fil émule une liaison série à 9600 bauds, sans parité, 8 bits par caractère et 1 bit de stop. Les commandes et réponses transmises entre le robot et le PC sont en clair, c'est à dire n'utilisent que les caractères affichables de la norme ASCII. On peut donc envoyer des commandes et lire les réponses directement depuis un simple terminal.

Les commandes commencent toutes par un caractère correspondant à la commande (voir Liste des commandes) et se terminent toutes par le caractère `CR` (0x0D ou 13) ou `LF/NL` (0x0A ou 10). Entre les deux se trouvent les valeurs numériques écrites sous forme de chaîne de caractère (pas en binaire) si la commande prévoit d'envoyer des données. Sinon, il n'y a rien.

En réponse, le robot envoi soit `OK_ANS` (caractère 'O') si la commande s'est passée sans problème, soit `ERR_ANS` (caractère 'E') si la commande à échouée soit `UNKNOWN_ANS` (caractère 'C') si la commande est inconnue. Si la réponse fait suite à une commande d'interrogation, la valeur retournée suit le caractère, séparée par ':'. La réponse se termine par un <CR>

Exemple:

Description de la commande	Commande envoyée au robot	Réponse reçue du robot
Mettre en route le robot	`W<CR>`	`O<CR>`
Avancer de 100 tours moteur	`M=100<CR>`	`O<CR>`
Interroger l'état de la batterie	`v<CR>`	`O:2<CR>`

4.3 - Liste des commandes

Le tableau suivant indique les différentes commandes reconnues, pour chaque version du logiciel, ainsi qu'un descriptif rapide. Les commande sont plus précisément détaillées ensuite.

Nom de la commande	Commande	Version
Ping	`p`	1.0 ⇨
Reset	`r`	1.0 ⇨
Set_motors	`m`	1.0 ⇨
Start_with_watchdog	`W`	1.0 ⇨
Reset_watchdog	`w`	1.0 ⇨
Get_ball_presence	`s`	1.0 ⇨
Get_battery_voltage	`v`	1.0 ⇨

Get_version	V	1.0 ⇨
Start_without_watchdog	u	1.0 ⇨
Move	M	1.0 ⇨
Turn	T	1.0 ⇨
Get_busy_state	b	1.0 ⇨

4.3.a Ping

La commande **`Ping`** sert uniquement à vérifier la présence du robot. La réponse est toujours `OK_ANS`.

Format de la commande	p<CR>
Paramètres	aucun
Réponse	OK_ANS
Exemple	p<CR>

4.3.b Reset

La commande **`Reset`** sert à faire revenir dans l'état `IDLE`, quelque soit l'état courant du robot. La réponse est toujours `OK_ANS`.

Format de la commande	r<CR>
Paramètres	aucun
Réponse	OK_ANS
Exemple	r<CR>

4.3.c Set_motors

La commande **`Set_motors`** commande chaque moteurs indépendamment et sans notion de distance. Les paramètres correspondent aux vitesses de chaque moteur. La réponse est `OK_ANS` si le robot est dans l'état `RUN`, `ERR_ANS` sinon. De plus, si le robot passe dans l'état `DISABLE` alors que les moteurs sont en actions, ceux ci sont automatiquement arrêtés.

Les vitesses sont exprimées en nombre de ticks (f=28,8 Khz) par demi-tour moteur. Plus la valeur est petite, plus la vitesse du moteur est grande.

Format de la commande	m=vit_g,vit_d<CR>
Paramètres	vit_g: vitesse pour le moteur gauche

	vit_d: vitesse pour le moteur droit
Réponse	OK_ANS dans l'état RUN ERR_ANS dans les autres états
Exemple	M=250, 250<CR>

4.3.d Start_with_watchdog

La commande **Start_with_watchdog** passe le robot dans l'état RUN et active le watchdog. Le watchdog doit être remis à zéro toutes les secondes avec la commande **Reset_watchdog**.

Format de la commande	W<CR>
Paramètres	aucun
Réponse	OK_ANS dans l'état IDLE ERR_ANS dans les autres états
Exemple	W<CR>

4.3.e Reset_watchdog

La commande **Reset_watchdog** remet le watchdog à zéro. La remise à zéro du watchdog doit se faire entre 950 ms et 1050 ms. Au bout de 5 erreur consécutives lors de la remise à zéro, le watchdog expire et le robot passe dans l'état DISABLE.

Format de la commande	w<CR>
Paramètres	aucun
Réponse	OK_ANS dans l'état RUN ERR_ANS dans les autres états
Exemple	w<CR>

4.3.f Get_battery_voltage

La commande **Get_battery_voltage** retourne l'état de charge de la batterie. La réponse est 2 si la batterie est correctement chargée, 1 si sa tension commence à être faible et 0 si la batterie est vide.

Format de la commande	v<CR>
Paramètres	aucun
Réponse	OK_ANS dans l'état RUN ERR_ANS dans les autres états
Exemple	v<CR>

4.3.g Get_version

La commande **`Get_version`** retourne la version de firmware du robot. La réponse est de la forme **`O:<major>,<minor><CR>`**. Le champ <major> correspond à la version majeur du logiciel, le champ <minor> correspond à la révision dans la version majeur.

Format de la commande	`V<CR>`
Paramètres	aucun
Réponse	`OK_ANS` dans tous les états
Exemple	`V<CR>`

4.3.h Start_without_watchdog

La commande **`Start_without_watchdog`** passe le robot dans l'état IDLE, sans lancer le watchdog. Il est du coup inutile d'utiliser la commande **`Reset_watchdog`** par la suite, bien que cette commande soit ignorée.

Format de la commande	`u<CR>`
Paramètres	aucun
Réponse	`OK_ANS` dans l'état `IDLE` `ERR_ANS` dans les autres états
Exemple	`u<CR>`

4.3.i Move

La commande **`Move`** fait avancer le robot en ligne droite d'un nombre donné de tour moteur. La vitesse ne peut être choisie. Si le paramètre distance est supérieur à 0, le robot avance, si le paramètre distance est inférieur à zéro, le robot recule. La distance est exprimée en nombre de tours moteur.

Format de la commande	`M=<distance><CR>`
Paramètres	Distance: nombre de tour moteur
Réponse	`OK_ANS` dans l'état `RUN` et le robot n'est pas occupé `ERR_ANS` dans les autres états ou si le robot est occupé
Exemple	`M=500<CR>`

4.3.j Turn

La commande **`Turn`** fait tourner le robot d'un nombre donné de tours moteur. La vitesse ne peut être choisie. Si le paramètre distance est supérieur à 0, le robot tourne à droite, si le paramètre distance est inférieur à zéro, le robot tourne à gauche. La distance est exprimée en nombre de tours moteur.

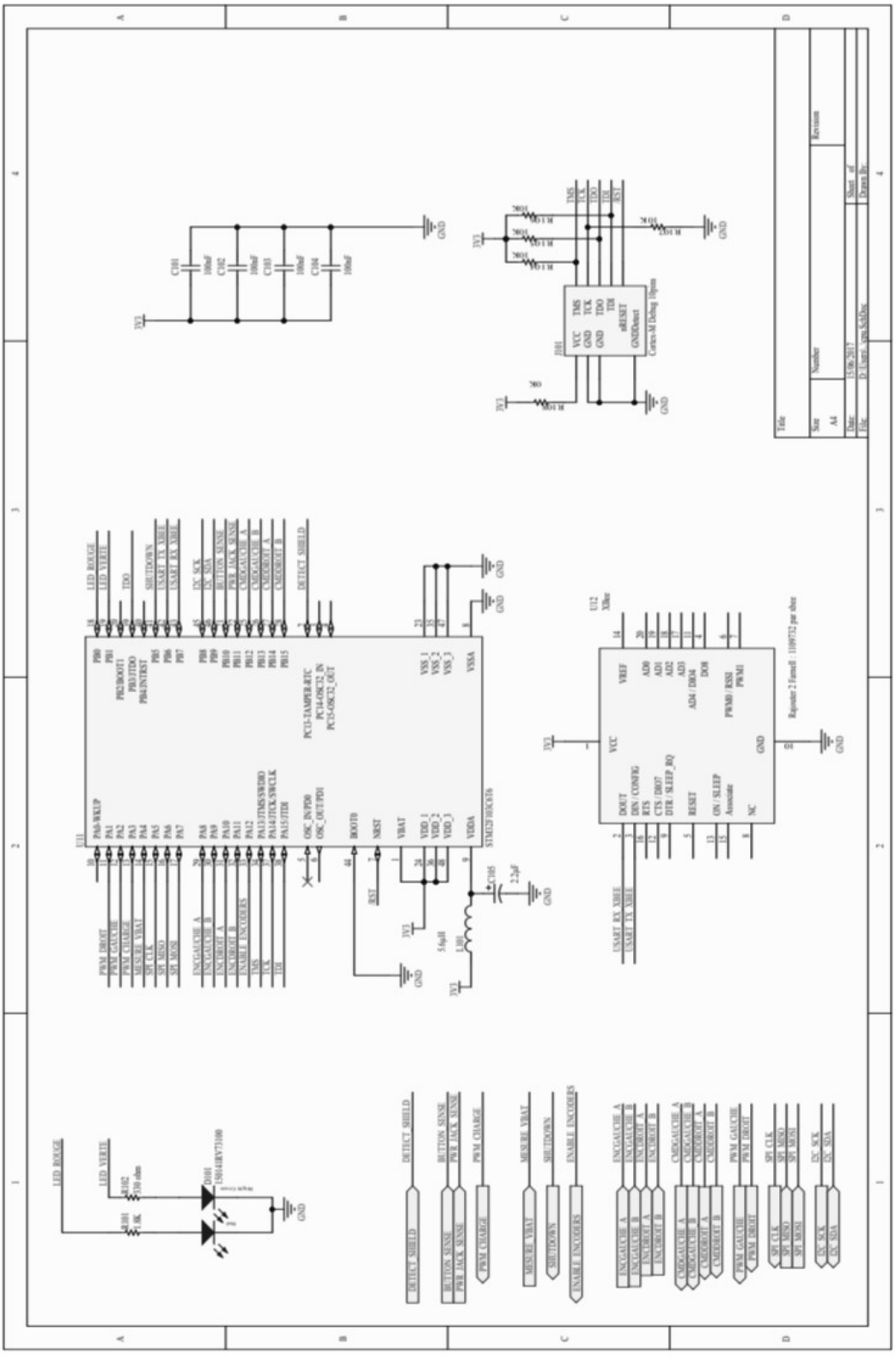
Format de la commande	`T=<distance><CR>`
Paramètres	Distance: nombre de tour moteur
Réponse	`OK_ANS` dans l'état `RUN` et le robot n'est pas occupé `ERR_ANS` dans les autres états ou si le robot est occupé
Exemple	`T=250<CR>`

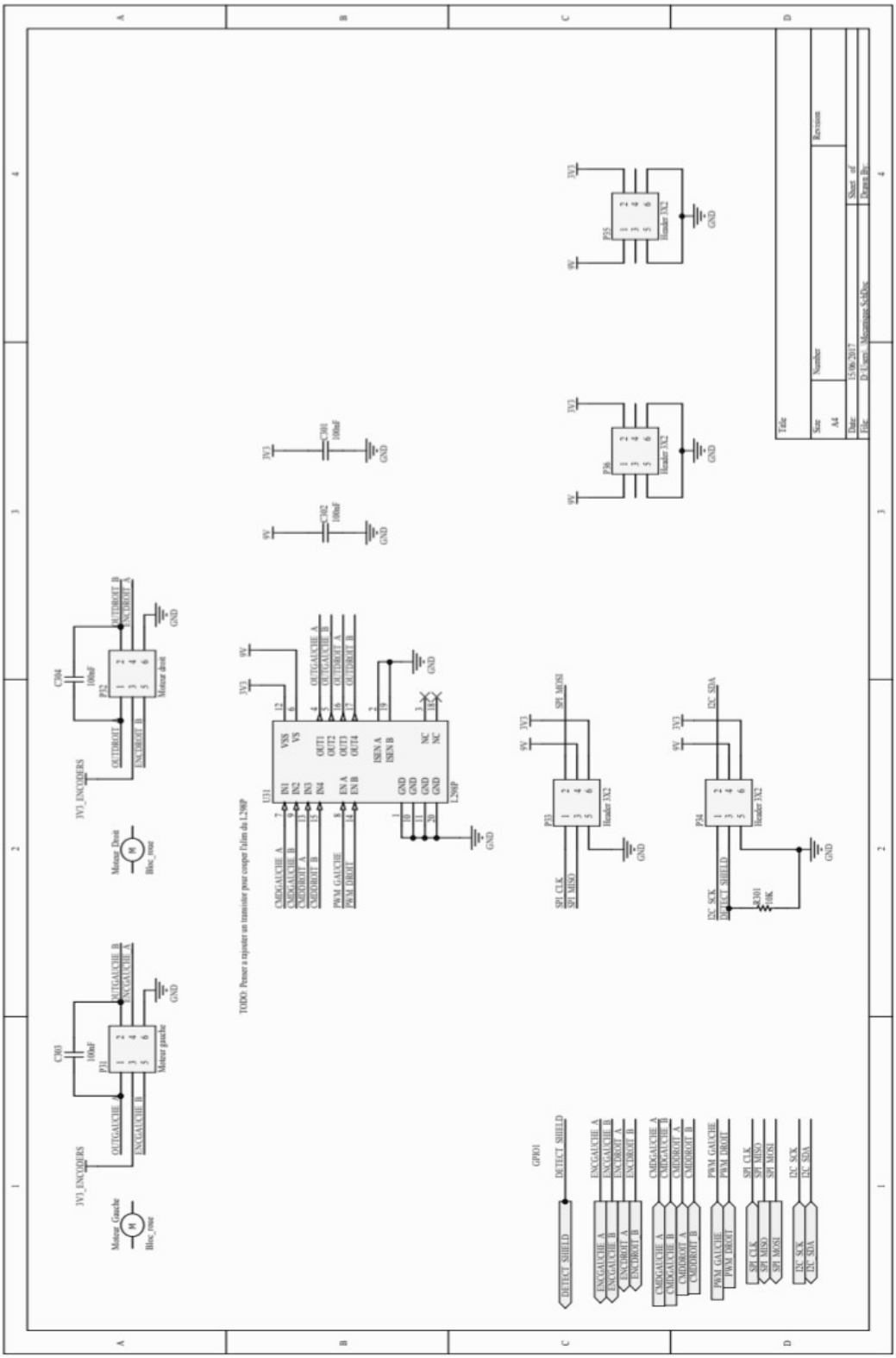
4.3.k Get_busy_state

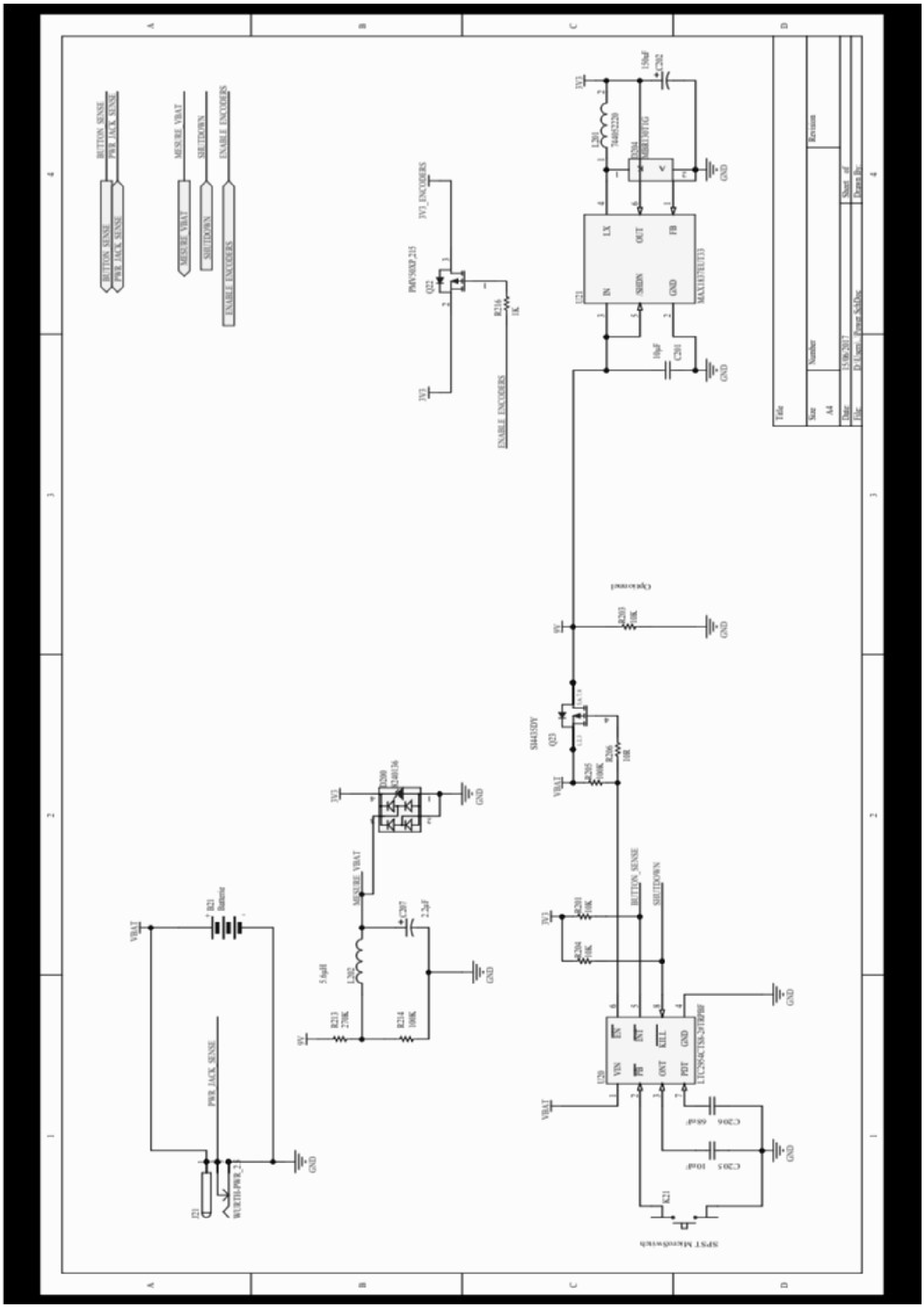
La commande **`Get_busy_state`** permet de savoir si une commande **`Move`** ou **`Turn`** est en court. Elle retourne 1 si une commande est en cours d'exécution et 0 si le robot est prêt pour une autre commande.

Format de la commande	`b<CR>`
Paramètres	aucun
Réponse	`OK_ANS` dans l'état `RUN` `ERR_ANS` dans les autres états
Exemple	`b<CR>`

5 - Schémas électriques




Title
Size
A4
Number
Revision
Date:
5-fév-2017
File:
D:\Cartes Alexandre\Alexandre.SchDoc
Sheet of
Drawn By:
TODO Penser a rajouter un transistor pour couper l'alim du L298P
U31
L298P
P31
Header 3X2
P32
P33
P34
P35
P36
C303
C304
C301
C302
100nF
SW101
10K
Moteur_Gauche
Moteur_Droit
Bloc_roue
GND
3V3
9V
3V3_ENCODERS
DETECT_SHIELD
SPI_CLK
SPI_MISO
SPI_MOSI
I2C_SCK
I2C_SDA
PWM_GAUCHE
PWM_DROIT



6 - Liste des composants

Designator	Description	Quantity	Distributeur	Value
B21	Keystone 1294	1	Radiospares: 172-5845 / Farnell: 1650674	
C101, C102, C103, C104, C301, C302, C303, C304	CAP 100nF 50V ±10% 1206 (3216 Metric) Thickness 1.7mm SMD	8		100nF
C105, C207	Solid Tantalum Chip Capacitor, Standard T491 Series - Industrial Grade	2		2.2µF
C201	CAP 10µF 25V ±10% 1206 (3216 Metric) Thickness 1mm SMD	1	Farnell : 2497120	10µF
C202	CAP Tantalum, Body B, 150µF / 6.3V	1	Farnell : 2491485	150uF
C205	CAP 100nF 50V ±10% 1206 (3216 Metric) Thickness 1.7mm SMD	1	Farnell : 1856586	10nF
C206	CAP 100nF 50V ±10% 1206 (3216 Metric) Thickness 1.7mm SMD	1	Farnell : 2497117	68nF
D101	SMD Bi-Color Top LED, WL-SBTW, Red & Bright Green	1	Farnell : 2322113	
D200	TVS Diode WE-TVS-HS, VRWM=3.3V	1		
D204	Schottky Power Rectifier, 2-Pin SOD-123, Pb-Free, Tape and Reel	1	Farnell : 1431045	
J21	Low Voltage Power Supply Connector	1	Farnell : 2472153	
J101		1	Farnell : 1667729	
K21	SPST - MicroSwitch	1	Farnell: 2435161	
L101, L202	Inductor SMD	2		5.6µH
L201	SMD Shielded Tiny Power Inductor WE-TPC, L = 22.0 µH	1	Farnell : 1635855	
Moteur_Droit, Moteur_Gauche		2		
P31	Header, 3-Pin, Dual row	1	Farnell : 1022231	
P32	Header, 3-Pin, Dual row	1	Farnell : 1022231	
P33, P34, P35, P36	Header, 3-Pin, Dual row	4	Farnell : 1022231	
Q22	Single P-Channel Trench MOSFET, -20 V, -55 to 150 degC, 3-Pin SOT23, Tape and Reel	1	Farnell : 2498580	
Q23	P-Channel 30V (D-S) Rated MOSFET	1		
R101	1KB 0.125W 5% 0805 (2012 Metric) SMD	1		1.8K
R102	330R 0.125W 5% 0805 (2012 Metric) SMD	1		330 ohm
R104, R105, R106, R107, R201, R203, R301	10K 0.25W 5% 1206 (3216 Metric) SMD	7		10K
R108	10K 0.25W 5% 1206 (3216 Metric) SMD	1		0K
R204	10K 0.25W 5% 1206 (3216 Metric) SMD	1		10K
R205	10K 0.25W 5% 1206 (3216 Metric) SMD	1		100K
R206	10K 0.25W 5% 1206 (3216 Metric) SMD	1		10R
R213	270K 0.125W 5% 0805 (2012 Metric) SMD	1		270K
R214	10K 0.25W 5% 1206 (3216 Metric) SMD	1		100K
R216	10K 0.25W 5% 1206 (3216 Metric) SMD	1		1K
U11	STM32 ARM-based 32-bit MCU with 32 Kbytes Flash, 48-pin LQFP, Industrial Temperature	1	Farnell : 1972391	
U12	Xbee RF transceiver from Digi	1		
U20	Pushbutton On / Off Controller with µP Interrupt, 8-pin SOT23 (TS8-8), 0 to 70 degC, Pb-Free, Tape and Reel	1		
U21		1	Farnell : 2514159	
U31	Dual Full Bridge Driver	1	Radiospares : 880-5308	