

MANUEL D'UTILISATION DU ROBOT DUMBER

Révision 1
Janvier 2012
S. DI MERCURIO

Table des matières

1.Présentation.....	4
2.Description du robot.....	4
3.Utilisation	5
4.Description détaillées des connecteurs et composants.....	6
4.1.Connecteur de programmation 6 (J1).....	6
4.2.Connecteur batterie 8 (J2).....	6
4.3.Connecteur chargeur (J3).....	6
4.4.Connecteurs moteurs (JP3 et JP4).....	6
4.5.Interrupteur marche/arrêt 7 (S2).....	7
4.6.XBEE (IC12).....	7
5.Fonctionnement.....	8
5.1.Machine d'état du robot.....	8
5.2.Format des commandes.....	8
5.3.Liste des commandes.....	9
a)Ping.....	9
b)Reset.....	10
c)Set_motors.....	10
d)Start_with_watchdog.....	10
e)Reset_watchdog.....	11
f)Get_ball_presence.....	11
g)Get_battery_voltage.....	11
h)Get_version.....	11
i)Start_without_watchdog.....	12
j)Move.....	12
k)Turn.....	12
l)Get_busy_state.....	12
6.Schéma électrique	14
7.Placement.....	15
8.Liste des composants.....	16

1. Présentation

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

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

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

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

2. Description du robot

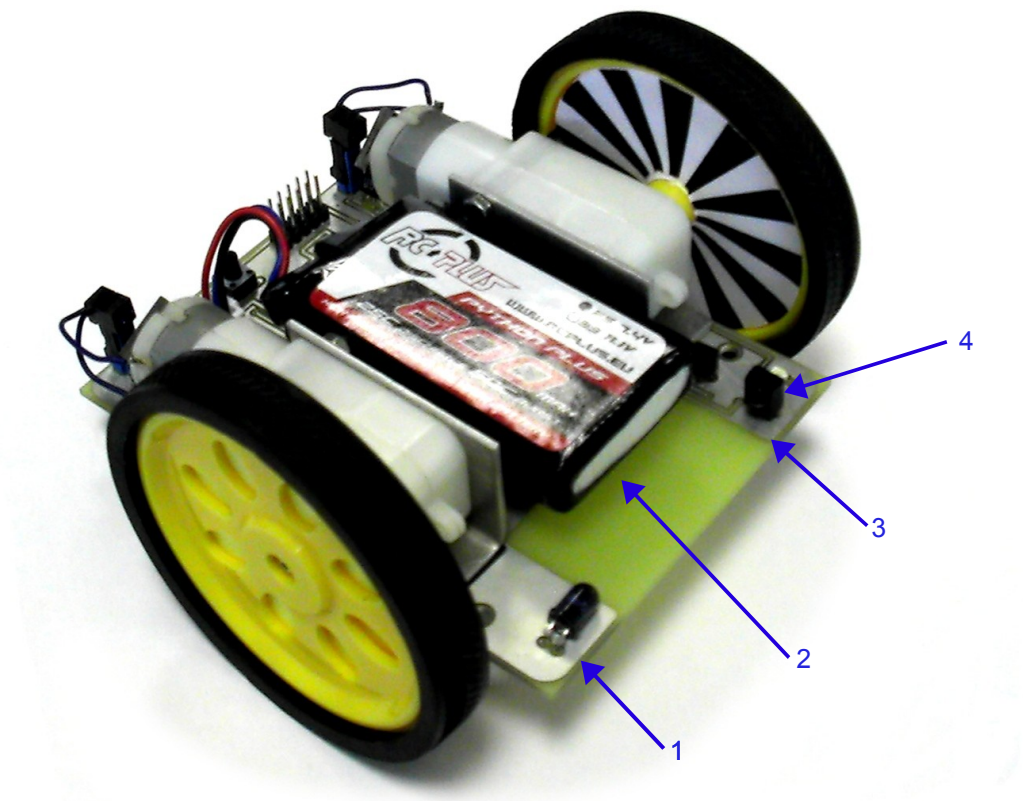


Illustration 1: Vue avant

- 1: LED émettrice IR (LED2)
- 2: Batterie
- 3: Récepteur IR (IC3)
- 4: Voyant d'activité (LED1)

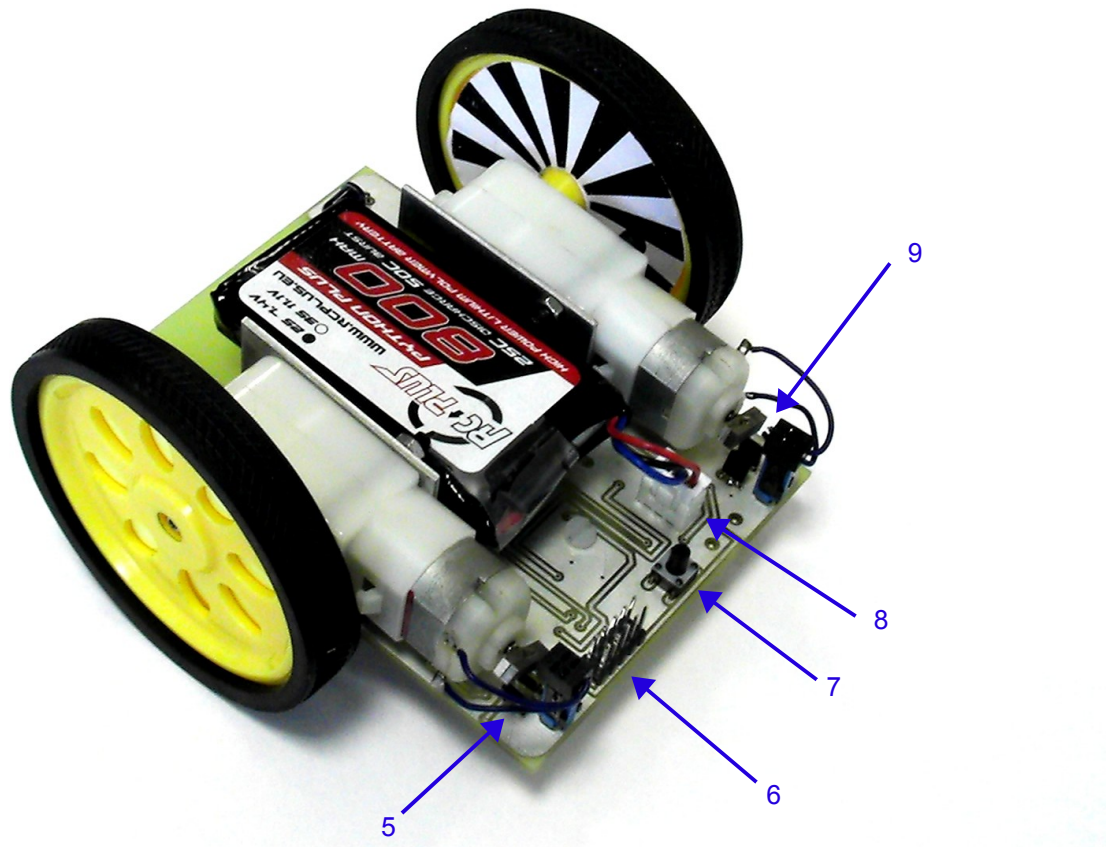


Illustration 2: Vue arrière

- 5: Capteur de vitesse moteur gauche
- 6: Connecteur de programmation (J1)
- 7: Bouton marche/arrêt (S2)
- 8: Connecteur batterie (J2)
- 9: Capteur de vitesse moteur droit

3. Utilisation

Lors de la première utilisation, il convient de vérifier que la batterie est correctement branchée au robot, via le connecteur 8. Le démarrage et l'arrêt du robot se font grâce au bouton poussoir 7 qui se trouve à l'arrière 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.

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 de communication

correspondant au numéro du robot (indiqué sous le robot, dans la case « numéro »).

4. Description détaillées des connecteurs et composants

4.1. Connecteur de programmation 6 (J1)

Le connecteur 6 (J1) sert à brancher l'outil de programmation du micro. Le brochage est le suivant :

Ligne	Signal
1	SCK
2	Point milieu
3	MISO
4	VCC
5	N/C
6	RESET/dw
7	N/C
8	N/C
9	MOSI
10	N/C

4.2. Connecteur batterie 8 (J2)

Le connecteur 8 (J2) sert à brancher la batterie. Le brochage est le suivant :

Ligne	Signal
1	GND
2	Point milieu
3	Vbat

4.3. Connecteur chargeur (J3)

Le connecteur J3 sert à brancher le chargeur de batterie. Le brochage est le suivant :

Ligne	Signal
1,3,5	Vbat
2,4,6	Point milieu
7,8,9	GND

4.4. Connecteurs moteurs (JP3 et JP4)

Les connecteurs JP3 et JP4 servent pour connecter le moteur gauche et droit. Le brochage est le suivant :

Ligne	Signal
1	Moteur +
2	Moteur -

4.5. Interrupteur marche/arrêt 7 (S2)

L'interrupteur S2 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.

4.6. XBEE (IC12)

Le module XBEE se branche sur le double connecteur 10 pts marqué IC12. Une sérigraphie sur le PCB indique l'orientation de l'antenne.

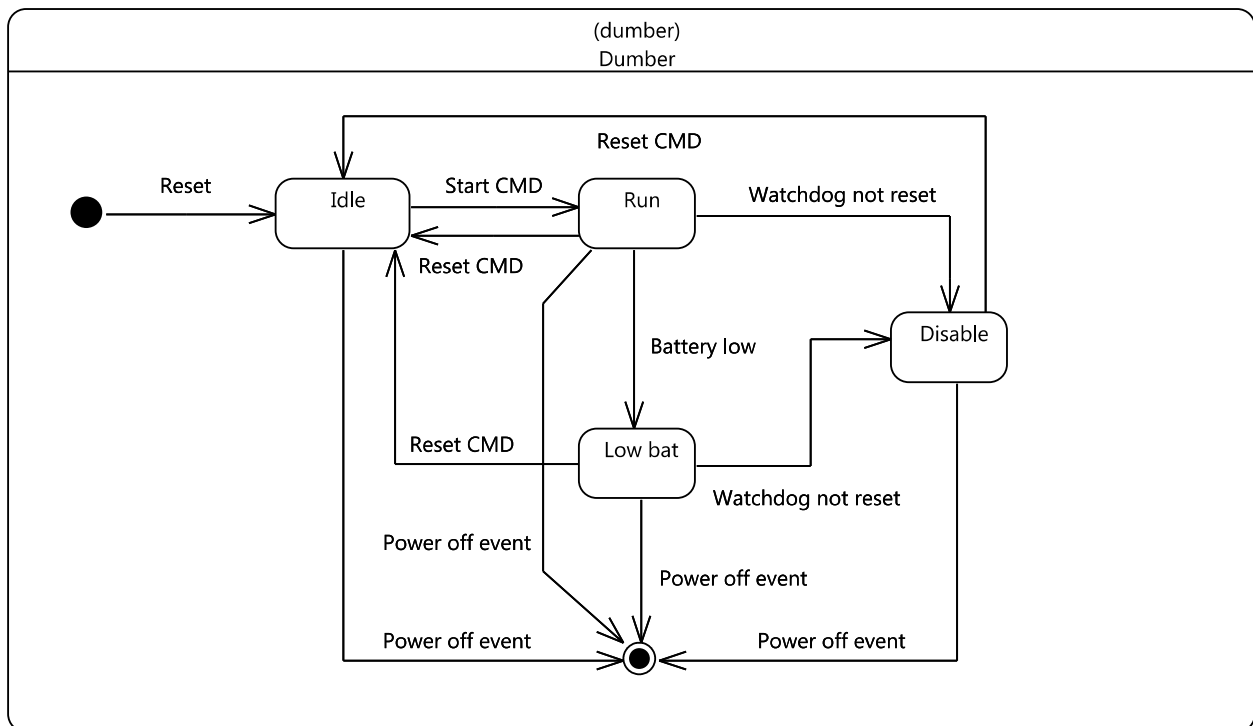
Ligne	Signal
1	VCC
2	TX
3	RX
5	Reset
10	GND
Autres	N/C

5. Fonctionnement

5.1. 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

5.2. 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 assurée par des modules XBEE présents 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 affichable 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 toute 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 envoie soit OK_ANS (caractère 'O') si la commande s'est passée sans problème, soit ERR_ANS (caractère 'E') si la commande a échoué 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>

5.3. Liste des commandes

Le tableau suivant indique les différentes commandes reconnues, pour chaque version du logiciel, ainsi qu'un descriptif rapide. Les commandes 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	1.1 ⇔
Start_without_watchdog	u	1.2 ⇔
Move	M	1.3 ⇔
Turn	T	1.3 ⇔
Get_busy_state	b	1.3 ⇔

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>

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>

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>

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>

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>

f) Get_ball_presence

La commande Get_ball_presence permet de savoir si un objet est présent dans la « bouche » du robot. La commande retourne 1 si un objet est présent, 0 dans l'autre cas.

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

g) 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>

h) 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>

i) 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>

j) 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>

k) 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.

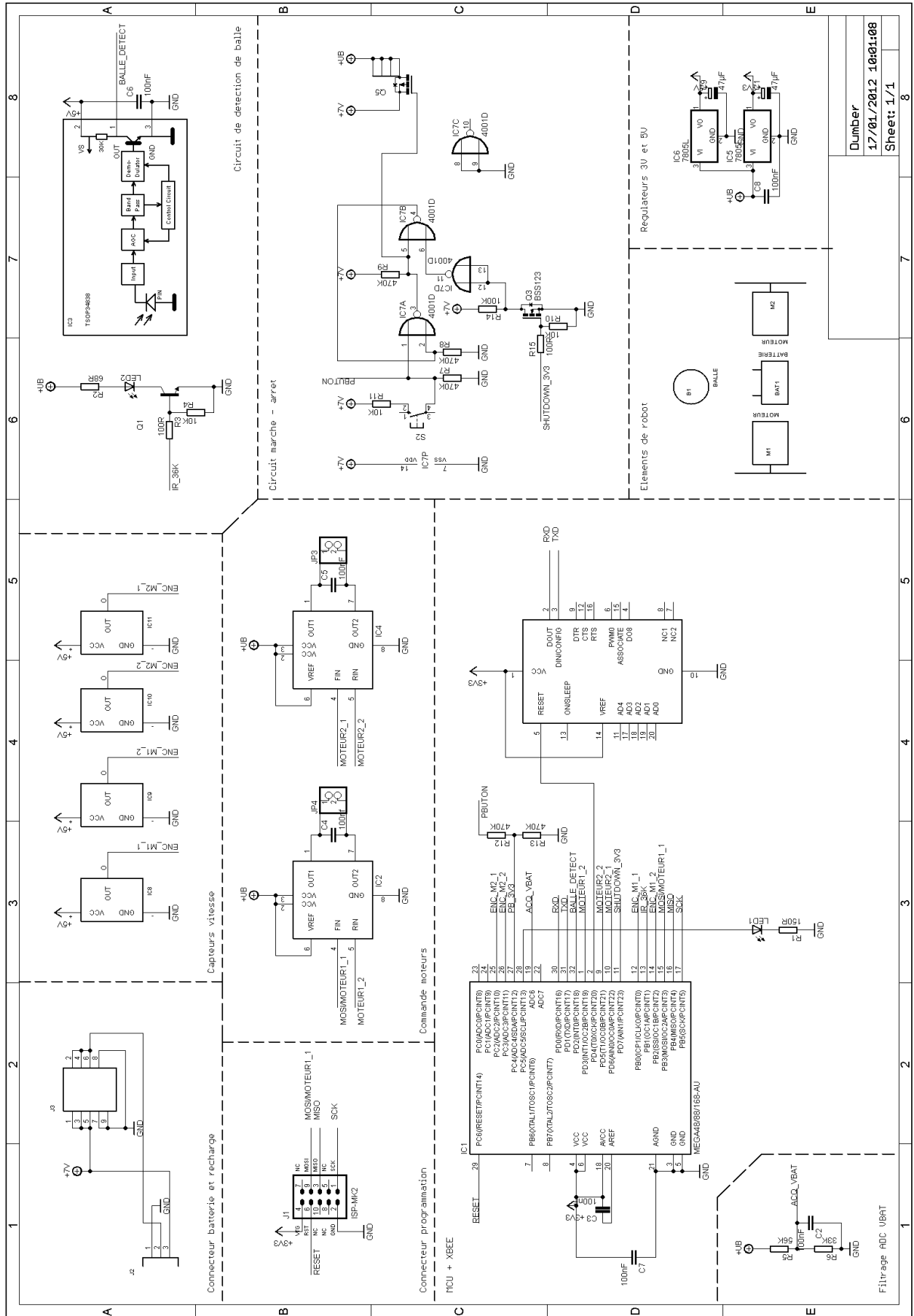
Format de la commande	T=<distance><CR>
Paramètres	Distance: nombre de tour moteur
Réponse	OK_ANS dans l'état RUN et le robot n'est pas occupé ERR_ANS dans les autres états ou si le robot est occupé
Exemple	T=250<CR>

l) Get_busy_state

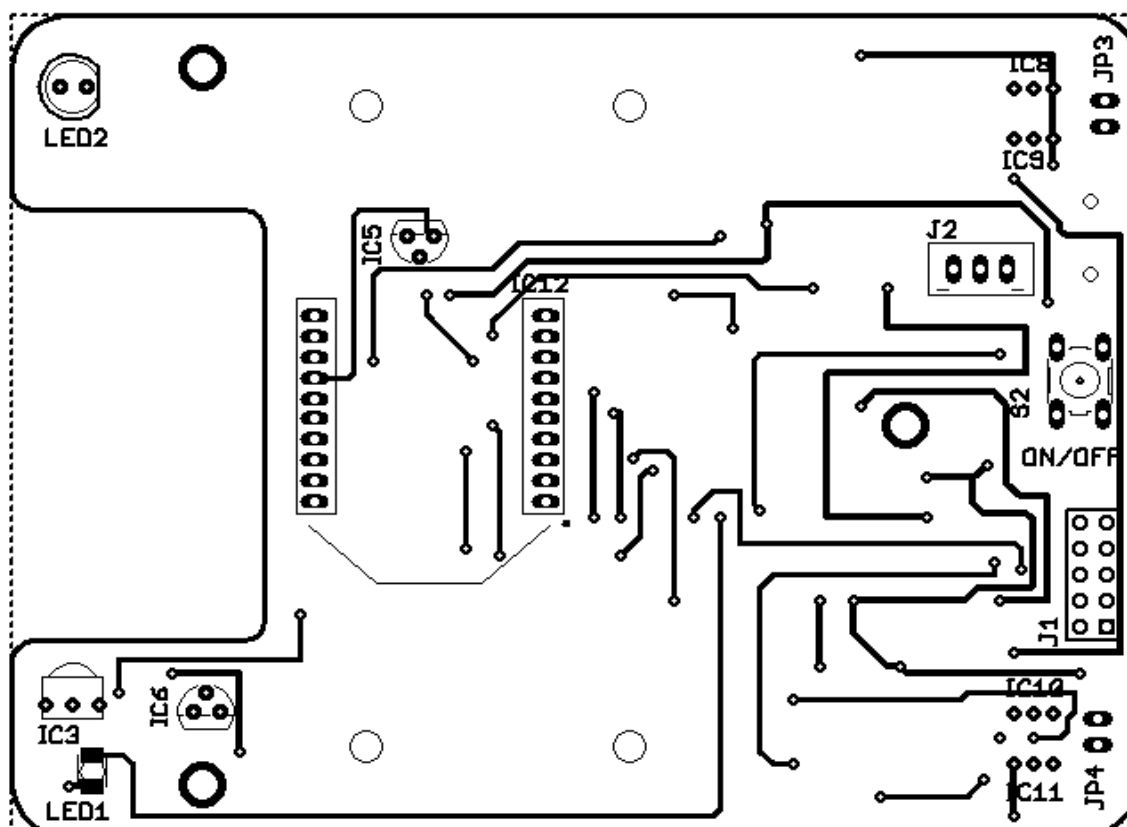
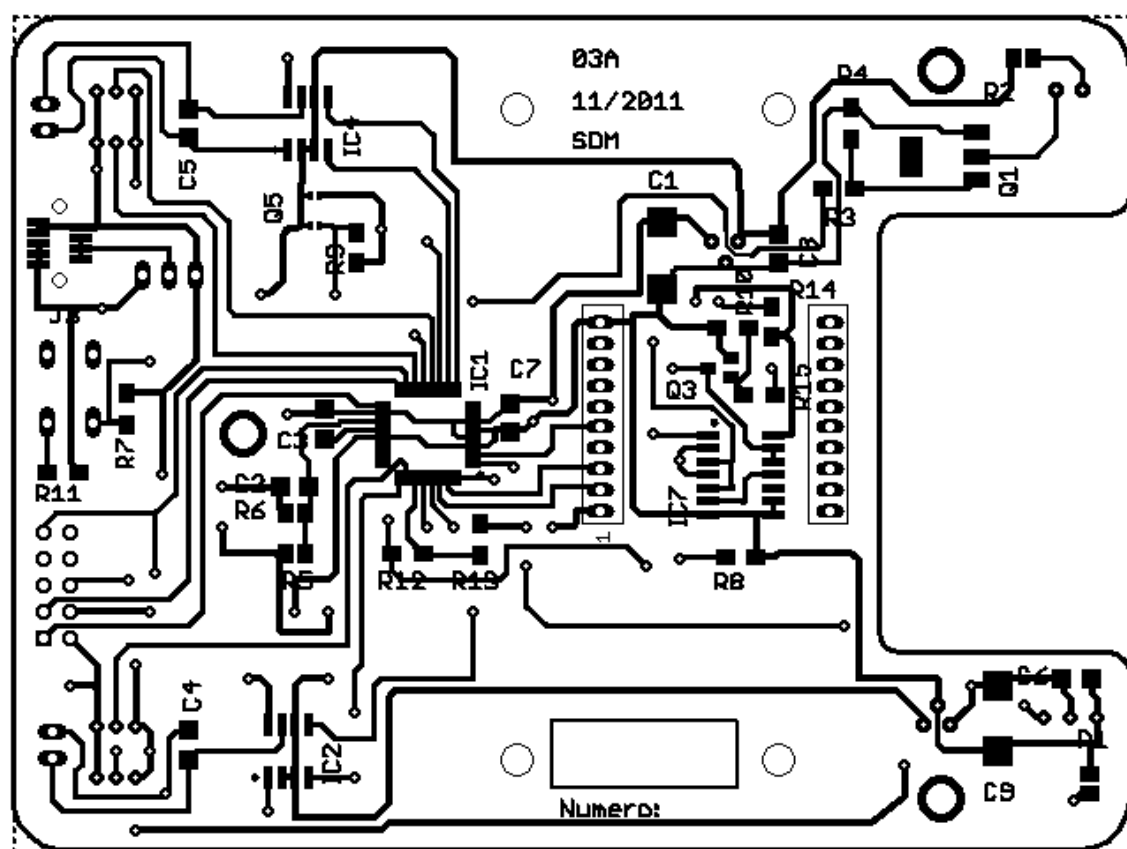
La commande Get_busy_state retourne si une commande Move ou Turn est en court. Elle retourne 1 si une commande est en cours d'exécution et 0 si le robot est prêt pour une autre commande.

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

6. Schéma électrique



7. Placement



8. Liste des composants

Part	Value	Device	Package	Description
B1	BALLE	BALLE	BALLE	
BAT1	BATTERIE	BATTERIE	BATTERIE	
C1	2.2F	CPOL-EUB/3528-21W	B/3528-21W	POLARIZED CAPACITOR, European
C2	100nF	C-EUC1206	C1206	CAPACITOR, European symbol
C3	100nF	C-EUC1206	C1206	CAPACITOR, European symbol
C4	100nf	C-EUC1206	C1206	CAPACITOR, European symbol
C5	100nF	C-EUC1206	C1206	CAPACITOR, European symbol
C6	100nF	C-EUC1206	C1206	CAPACITOR, European symbol
C8	100nF	C-EUC1206	C1206	CAPACITOR, European symbol
C9	2.2F	CPOL-EUB/3528-21W	B/3528-21W	POLARIZED CAPACITOR, European
IC1	MEGA48/88/168-AU	MEGA48/88/168-AU	TQFP32-08	
IC2	BD62XX	BD62XX	SO-08	
IC3	TSOP34838	TSOP2236	TSOP348	
IC4	BD62XX	BD62XX	SO-08	
IC5	7805L	7805L	TO92	Positive VOLTAGE REGULATOR
IC6	7805L	7805L	TO92	Positive VOLTAGE REGULATOR
IC7	4001D	4001D	SO14	Quad 2-input NOR
IC8	2SS52M	2SS52M	2SS	
IC9	2SS52M	2SS52M	2SS	
IC10	2SS52M	2SS52M	2SS	
IC11	2SS52M	2SS52M	2SS	
IC12	XBEE	XBEE	XBEE	
J1	AVR-ISP-MK2	AVR-ISP-MK2	AVR-ISP-10	
J2		3XH	3XH	
J3		M40-9-F	M40-600-9	
JP3		PINHD-1X2	1X02	PIN HEADER
JP4		PINHD-1X2	1X02	PIN HEADER
LED1		LEDKA-3528ASYC	KA-3528ASYC	LED
LED2		LED5MM	LED5MM	LED
M1	MOTEUR	MOTEUR	MOTEUR	
M2	MOTEUR	MOTEUR	MOTEUR	
Q1		BCP56	SOT223	
Q3	BSS123	BSS123	SOT23	N-CHANNEL MOS FET
Q5		RTQ030P02TR	TSMT-6	
R1	150R	R-EU_R1206W	R1206W	RESISTOR, European symbol
R2	68R	R-EU_R1206W	R1206W	RESISTOR, European symbol
R3	100R	R-EU_R1206W	R1206W	RESISTOR, European symbol
R4	10K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R5	56K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R6	33K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R7	470K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R8	470K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R9	470K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R10	10K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R11	10K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R12	470K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R13	470K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R14	100K	R-EU_R1206W	R1206W	RESISTOR, European symbol
R15	100R	R-EU_R1206W	R1206W	RESISTOR, European symbol
S2		10-XX	B3F-10XX	OMRON SWITCH