

Communication ASCII avec le bas niveau

Le bas niveau dispose de deux interfaces série, l'une étant purement binaire et permettant de contrôler le robot avec précision en temps réel, l'autre étant ASCII et permettant à un utilisateur de régler le bas niveau et de visualiser son état facilement. Voici la liste des commandes disponibles sur l'interface ASCII du bas niveau. Certaines de ces commandes correspondent directement à un ordre binaire, leur ID est marqué en gras, il est entre 0 et 127 au lieu d'être entre 128 et 255.

ID : *identifiant utilisé dans le code du microcontrôleur pour désigner l'ordre.*

L : *(-/L) « L » indique que l'ordre sera considéré comme 'long' par le bas niveau. On ne peut pas exécuter un nouvel ordre long tant qu'un ordre long est en cours d'exécution, on peut en revanche exécuter un ordre immédiat.*

Commande : *chaîne de caractère que l'utilisateur doit taper pour exécuter l'ordre.*

Argument : *« int » désigne n'importe quel entier lisible via la fonction C « atoi() »
« float » désigne n'importe quel entier lisible via la fonction C « atof() »
{ 'str1' ; 'str2' ; 'str3' } représente l'ensemble des chaînes de caractères acceptées comme argument.*

Remarque : *[get/set] indique que la commande est à la fois un 'getter' et un 'setter' pour une certaine variable. Utilisée sans argument, la commande indiquera la valeur courante de la variable, tandis qu'avec un argument, la variable sera réglée à la valeur indiquée.*

ID	L	Commande	Argument	Description
0x80	-	logon	int	Active le canal de Log indiqué en argument.
0x81	-	logoff	int	Désactive le canal de Log indiqué en argument.
0x82	-	batt	-	Affiche le niveau de la batterie.
0x39	L	stop	-	Arrêt du robot.
0x39	L	s	-	Arrêt du robot (alias de la commande stop).
0x85	-	save	-	Sauvegarde tous les réglages courants concernant l'asservissement dans une mémoire non volatile, afin de les rétablir automatiquement au prochain démarrage.
0x86	-	display	-	Affiche tous les réglages courants concernant l'asservissement.
0x87	-			
0x88	-			
0x89	-			
0x8A	-	pos	-	Affiche la position du robot en mm et radians.
0x8B	-	x	float	[get/set] composante X de la position (mm).
0x8C	-	y	float	[get/set] composante Y de la position (mm).
0x8D	-	o	float	[get/set] orientation du robot (radians).
0x8E	-	rp	{ 'i' ; 'w' }	Réinitialisation de la position. 'i' : position = (0,0,0) 'w' : position = côté intech, 'w' : position = côté fenêtre
0x8F	-	dir	float	[get/set] courbure (m^{-1})
0x90	-	axg	int	[get/set] angle de l'AX12 de direction gauche (0-300°).
0x91	-	axd	int	[get/set] angle de l'AX12 de direction droit (0-300°).

0x92	-	cod	-	<i>Affiche le nombre de ticks comptés par chaque codeuse.</i>
0x93	-	setaxid	int	<i>Règle l'ID de chaque AX12 actuellement connecté, à la valeur passée en argument.</i>
0x94	-	pid	{ 'g' ; 'd' ; 't' }	<i>Règle le PID courant sur 'vitesse gauche', 'vitesse droite' ou 'translation' respectivement.</i>
0x95	-	kp	float	<i>[get/set] kp du PID courant.</i>
0x96	-	ki	float	<i>[get/set] ki du PID courant.</i>
0x97	-	kd	float	<i>[get/set] kd du PID courant.</i>
0x98	-	smgre	int	<i>[get/set] tolérance (epsilon) du 'Stopping manager'.</i>
0x99	-	smgrt	int	<i>[get/set] temps de réponse du 'Stopping manager' (ms).</i>
0x9A	-	bmgrs	float	<i>[get/set] sensibilité des deux 'Blocking manager'.</i>
0x9B	-	bmgrt	int	<i>[get/set] temps de réponse des deux 'Blocking manager' (ms)</i>
0x9C	-	mms	int	<i>[get/set] maxMovingSpeed (ticks.s⁻¹).</i>
0x9D	-	macc	int	<i>[get/set] maxAcceleration (ticks.s⁻²).</i>
0x9E	-	cp	bool	<i>Active/désactive le contrôle de la position.</i>
0x9F	-	cvg	bool	<i>Active/désactive le contrôle de la vitesse gauche.</i>
0xA0	-	cvd	bool	<i>Active/désactive le contrôle de la vitesse droite.</i>
0xA1	-	cpwm	bool	<i>Active/désactive le contrôle des PWM.</i>
0xA2	-	pwm	int	<i>Force les PWM à la valeur passée en argument pendant un certain temps.</i>
0xA3	-	a	int	<i>Force la consigne en vitesse de chaque moteur à la valeur passée en argument, pendant un certain temps (ticks.s⁻¹). Si aucun argument n'est donné, la valeur utilisée sera celle de la précédente exécution.</i>
0xA4	-	p	int	<i>Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est donné, la valeur utilisée sera celle de la précédente exécution.</i>
0xA5	-			
0xA6	-			
0xA7	-			
0xA8	-			
0xA9	-			
0xAA	-			
0xAB	-			
0xAC	-			
0xAD	-			
0xAE	-			
0xAF	-			
0xB0	-	capt	-	<i>Affiche les valeurs brutes lues par chaque capteur.</i>
0xB1	-			
0xB2	-			
0xB3	-			
0xB4	-			
0xB5	-			
0xB6	-			
0xB7	-			
0xB8	-			
0xB9	-			
0xBA	-			

0xBB	-			
0xBC	-			
0xBD	-			
0xBE	-			
0xBF	-			