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.

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 : chaine 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 chaines 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.		
0x83	_	stop	_	Arrêt du robot.		
0x83	_	s	_	Arrêt du robot (alias de la commande stop).		
0x85	ı	save	-	Sauvegarde tous les réglages courants concernant l'asservissement dans une mémoire non volatile, afin de les rétablir automatiquement au prochain démarrage.		
0x86	ı	display	_	Affiche tous les réglages courants concernant l'asservissement.		
0x87	-	default	-	Rétablis les paramètres par défaut concernant l'asservissement (ceux stockés dans l'EEPROM).		
0x88	_					
0x89	-					
0x8A	-	pos	_	Affiche la position du robot en mm et radians.		
0x8B	ı	X	float	[get/set] composante X de la position (mm).		
0x8C	ı	У	float	[get/set] composante Y de la position (mm).		
0x8D	-	0	float	[get/set] orientation du robot (radians).		
0x8E	-	rp	{ ''; 'i'; 'w'}	Réinitialisation de la position. '': position = (0,0,0) 'i': 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	_	Affiche le nombre de ticks comptés par chaque codeuse.		

0x94 - pid {'g'; 'd'; 't'} Regie le PID courant sur 'vitesse gauche', 'vitesse droit 'ritesse' 0x95 - kp float get/set kp du PID courant. 0x96 - ki float get/set ki du PID courant. 0x97 - kd float get/set ki du PID courant. 0x98 - smgre int get/set tolerance (epsilon) du 'Stopping manager' (ms). 0x98 - smgrt int get/set temps de réponse du 'Stopping manager' (ms). 0x98 - bmgrs float get/set maxMovingSpeed (ticks,s*). 0x99 - bmgrs float get/set maxMovingSpeed (ticks,s*). 0x99 - mms int get/set maxAcceleration (ticks,s*). 0x99 - mms int get/set maxAcceleration (ticks,s*). 0x99 - mms int get/set maxAcceleration (ticks,s*). 0x99 - cp bool Active/désactive le contrôte de la position. 0x99 - cp bool Active/désactive le contrôte de la vitesse gauche. 0x04 - cvd bool Active/désactive le contrôte de la vitesse doite. </th <th></th> <th></th> <th></th> <th></th> <th>Pàgle l'ID de ghague AV10 getuellement connecté à la</th>					Pàgle l'ID de ghague AV10 getuellement connecté à la		
Ox94	0x93	-	setaxid	int	Règle l'ID de chaque AX12 actuellement connecté, à la valeur passée en graument		
Name					-		
0x95	0x94	-	pid	{ 'g'; 'd'; 't'}			
0x96	0x95	-	kp	float			
0x97	0x96	-	ki	float			
0x98 -	0x97	_	kd	float	1 1		
0x99 -	0x98	_	smgre	int	1 19 / 1		
0x9A	0x99	_		int			
0x9B	0x9A	_	_	float			
0x9C - mms	0x9B	_	_	int	· · · · · · · · · · · · · · · · · · ·		
0x9D	0x9C	_		int			
0x9E	0x9D	-	macc	int			
0x9F - cvg bool Active/désactive le contrôle de la vitesse gauche. 0xA0 - cvd bool Active/désactive le contrôle de la vitesse droite. 0xA1 - cpwm bool Active/désactive le contrôle de la vitesse droite. 0xA2 L pwm int Force les PWM à la valeur passée en argument pendan certain temps. 0xA3 L a int Force les PWM à la valeur passée en argument pendan certain temps. 0xA3 L a int passée en argument, pendant un certain temps (ticks.s' aucun argument n'est doné, la valeur utilisée sera cell la précédente exécution. 0xA4 L p int passée en argument (mm). Si aucun argument n'est don la valeur utilisée sera celle de la précédente exécution. 0xA5 - k1 float jet/set/ curvatureCorrector K1 0xA6 - k2 float jet/set/ maxDeceleration (ticks.s'²). 0xA8 - int jet/set/ maxDeceleration (ticks.s'²). 0xA8 - capt - Affiche les valeurs brutes lues par chaque capteur. 0xB0 - ca	0x9E	_	ср	bool	` · · · · · · · · · · · · · · · · · · ·		
0xA0 - cvd bool Active/désactive le contrôle de la vitesse droite. 0xA1 - cpwm bool Active/désactive le contrôle des PWM. 0xA2 L pwm int Force les PWM à la valeur passée en argument pendan certain temps. 0xA3 L a int Force les PWM à la valeur passée en argument pendant un certain temps (ticks.s' aucun argument n'est donné, la valeur utilisée sera cell la précédente exécution. 0xA4 L p int Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est don la valeur utilisée sera celle de la précédente exécution. 0xA5 k1 float /get/set/ curvatureCorrector K1 0xA6 k2 float /get/set/ curvatureCorrector K2 0xA7 mdec int /get/set/ maxDeceleration (ticks.s²). 0xA8 - /get/set/ maxDeceleration (ticks.s²). 0xA9 - /get/set/ maxDeceleration (ticks.s²). 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x	0x9F	_		bool	<u> </u>		
0xA1 - cpwm bool Active/désactive le contrôle des PWM. 0xA2 L pwm int Force les PWM à la valeur passée en argument pendan certain temps. 0xA3 L a int Force la consigne en vitesse de chaque moteur à la vale passée en argument, pendant un certain temps (ticks.s' aucun argument n'est donné, la valeur utilisée sera cell la précédente exécution. 0xA4 L p int Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est do la valeur utilisée sera celle de la précédente exécution. 0xA5 - k1 float [get/set] curvatureCorrector K1 0xA6 - k2 float [get/set] curvatureCorrector K2 0xA7 - mdec int [get/set] maxDeceleration (ticks.s'²). 0xA8 - [get/set] maxDeceleration (ticks.s'²). 0xA9 - Affiche les valeurs brutes lues par chaque capteur. 0xB0 - capt - 0xB0 - axn int Règle l'angle de l'AX12 du filet. 0xB1 - axn int Règle l'angle de l'AX12 du filet. 0xB2 L pnh - Ordre HL 'PullDownNet' 0xB3 L pnh - Ord	0xA0	_		bool	· · · · · · · · · · · · · · · · · · ·		
OxA2 L pwm int Force les PWM à la valeur passée en argument pendan certain temps. Force la consigne en vitesse de chaque moteur à la vale passée en argument, pendant un certain temps (ticks.s' aucun argument n'est donné, la valeur utilisée sera cell la précédente exécution. Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est do la valeur utilisée sera celle de la précédente exécution. OxA5 - k1 float get/set curvatureCorrector K1 OxA6 - k2 float get/set curvatureCorrector K2 OxA7 - mdec int get/set maxDeceleration (ticks.s²). OxA8 - OxA9 - OXA		_			† '		
Certain temps. Force la consigne en vitesse de chaque moteur à la vale passée en argument, pendant un certain temps (ticks.s' aucun argument n'est donné, la valeur utilisée sera cell la précédente exécution. Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est don la valeur utilisée sera celle de la précédente exécution. OXA5 - kl float [get/set] curvatureCorrector K1 OXA6 - k2 float [get/set] curvatureCorrector K2 OXA7 - mdec int [get/set] maxDeceleration (ticks.s²). OXA8 - OXA9 - OX	0 - 0	_	-		 		
Force la consigne en vitesse de chaque moteur à la vale passée en argument, pendant un certain temps (ticks.s' aucun argument n'est donné, la valeur utilisée sera cell la précédente exécution. Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est don la valeur utilisée sera celle de la précédente exécution. OxA5 - kl float [get/set] curvature Corrector K1 OxA6 - k2 float [get/set] curvature Corrector K2 OxA7 - mdec int [get/set] maxDeceleration (ticks.s²). OxA8 - OxA9 - OxB0 - capt - Affiche les valeurs brutes lues par chaque capteur. OxB5 - axn int Règle l'angle de l'AX12 du filet. Ox3D L pdn - Ordre HL 'PullDownNet' Ox3E L pnh - Ordre HL 'PullUpNet' Ox3F L pun - Ordre HL 'PullUpNet' Ox40 L opn - Ordre HL 'CoseNet' Ox41 L cln - Ordre HL 'CrossFlipFlop' Ox42 L cff - Ordre HL 'EjectLeftSide' Ox43 L els - Ordre HL 'EjectRightSide' Ox44 L rls - Ordre HL 'EjectRightSide' Ox45 L ers - Ordre HL 'EjectRightSide' Ox46 L rrs - Ordre HL 'EjectRightSide' Ox47 L fa - Ordre HL 'EjectRightSide' Ox48 L fa - Ordre HL 'EjectRightSide'	0xA2	L	pwm	int			
Decided to the problem of the proble					1		
aucun argument n'est donné, la valeur utilisée sera cell la précédente exécution. Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est don la valeur utilisée sera celle de la précédente exécution. OXA5 - K1 float fget/set curvatureCorrector K1 OXA6 - K2 float fget/set curvatureCorrector K2 OXA7 - mdec int fget/set maxDeceleration (ticks.s²). OXA8 - OXA9 - OXB0 - Capt - Affiche les valeurs brutes lues par chaque capteur. OXB5 - axn int Règle l'angle de l'AX12 du filet. OX3D L pdn - Ordre HL 'PullDownNet' OX3E L pnh - Ordre HL 'PullVpNet' OX40 L opn - Ordre HL 'OpenNet' OX41 L cln - Ordre HL 'CloseNet' OX42 L cff - Ordre HL 'CloseNet' OX43 L els - Ordre HL 'EjectLeftSide' OX44 L rls - Ordre HL 'EjectLeftSide' OX45 L ers - Ordre HL 'EjectLeftSide' OX46 L rrs - Ordre HL 'EjectLeftSide' OX47 L fa - Ordre HL 'EjectRightSide' OX48 L fa - Ordre HL 'FunnyAction'	0 - 0	_					
la précédente exécution. Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est doi la valeur utilisée sera celle de la précédente exécution. 0xA5 - k1 float [get/set] curvatureCorrector K1 0xA6 - k2 float [get/set] curvatureCorrector K2 0xA7 - mdec int [get/set] maxDeceleration (ticks.s²). 0xA8 - 0xA9 - 0xB0 - capt - Affiche les valeurs brutes lues par chaque capteur. 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullVeNet' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'EjetLeftSide' 0x44 L rls - Ordre HL 'EjetRightSide' 0x45 L ers - Ordre HL 'EjetRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0xA3	Ь	a	int			
Ajoute à la consigne de position (translation) la valeur passée en argument (mm). Si aucun argument n'est do la valeur utilisée sera celle de la précédente exécution. 0xA5 - k1 float [get/set] curvatureCorrector K1 0xA6 - k2 float [get/set] curvatureCorrector K2 0xA7 - mdec int [get/set] maxDeceleration (ticks.s ⁻²). 0xA8 - 0xA9 - 0xB0 - capt - Affiche les valeurs brutes lues par chaque capteur. 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullUpNet' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'EjectLeftSide' 0x43 L els - Ordre HL 'RearmLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'							
0xA4Lpintpassée en argument (mm). Si aucun argument n'est do la valeur utilisée sera celle de la précédente exécution.0xA5-k1float[get/set] curvatureCorrector K10xA6-k2float[get/set] curvatureCorrector K20xA7-mdecint[get/set] maxDeceleration (ticks.s²).0xA8Affiche les valeurs brutes lues par chaque capteur.0xA9Affiche les valeurs brutes lues par chaque capteur.0xB5-axnintRègle l'angle de l'AX12 du filet.0x3DLpdn-Ordre HL 'PullDownNet'0x3ELpnh-Ordre HL 'PullUpNet'0x3ELpun-Ordre HL 'PullUpNet'0x43Lpun-Ordre HL 'CloseNet'0x41Lcln-Ordre HL 'CloseNet'0x42Lcff-Ordre HL 'EjectLeftSide'0x43Lels-Ordre HL 'EjectLeftSide'0x44Lrls-Ordre HL 'EjectRightSide'0x45Lers-Ordre HL 'EjectRightSide'0x46Lrrs-Ordre HL 'FunnyAction'					-		
la valeur utilisée sera celle de la précédente exécution. 0xA5	0xA4	L	р	int	_ ` ` ,		
0xA5 - k1 float [get/set] curvatureCorrector K1 0xA6 - k2 float [get/set] curvatureCorrector K2 0xA7 - mdec int [get/set] maxDeceleration (ticks.s²). 0xA8 - 0xA9 - - 0xB0 - capt - Affiche les valeurs brutes lues par chaque capteur. 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullUpwet' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'EjectLeftSide' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'EjectRightSide' 0x45 L ers - Ordre HL 'RearmRightSide' 0x46 L rrs - Ordre HL 'FunnyAction'							
0xA7 - mdec int [get/set] maxDeceleration (ticks.s²). 0xA8 - maxber 0xA9 - maxber - maxber 0xB0 - capt	0xA5	_	k1	float			
0xA7 - mdec int [get/set] maxDeceleration (ticks.s²). 0xA8 - 0xA9 - - Affiche les valeurs brutes lues par chaque capteur. 0xB0 - capt - - Affiche les valeurs brutes lues par chaque capteur. 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - - Ordre HL 'PullDownNet' 0x3E L pnh - - Ordre HL 'PullUpNet' 0x45 L pun - - Ordre HL 'OpenNet' 0x40 L opn - - Ordre HL 'CloseNet' 0x41 L cln - - Ordre HL 'CrossFlipFlop' 0x42 L cff - Ordre HL 'EjectLeftSide' 0x43 L els - - Ordre HL 'RearmLeftSide' 0x44 L rls - - Ordre HL 'EjectRightSide' 0x45 L ers - - Ordre HL 'RearmRightSide' 0x46 L rrs - - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0xA6	_	k2	float	10 /		
0xA8	0xA7	_	mdec	int			
0xB0 - capt - Affiche les valeurs brutes lues par chaque capteur. 0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullUpNet' 0x3F L pun - Ordre HL 'OpenNet' 0x40 L opn - Ordre HL 'CloseNet' 0x41 L cln - Ordre HL 'CrossFlipFlop' 0x42 L cff - Ordre HL 'EjectLeftSide' 0x43 L els - Ordre HL 'RearmLeftSide' 0x44 L rls - Ordre HL 'EjectRightSide' 0x45 L ers - Ordre HL 'RearmRightSide' 0x46 L rrs - Ordre HL 'FunnyAction'	0xA8	_					
0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullUpNet' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'RearmRightSide' 0x46 L rrs - Ordre HL 'FunnyAction'	0xA9	-					
0xB5 - axn int Règle l'angle de l'AX12 du filet. 0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullUpNet' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0xB0	-	capt	-	Affiche les valeurs brutes lues par chaque capteur.		
0x3D L pdn - Ordre HL 'PullDownNet' 0x3E L pnh - Ordre HL 'PullUpNet' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0xB5	-	axn	int			
0x3E L pnh - Ordre HL 'PutNetHalfway' 0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x3D	L	pdn	-			
0x3F L pun - Ordre HL 'PullUpNet' 0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x3E	L	pnh	_			
0x40 L opn - Ordre HL 'OpenNet' 0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x3F	L	pun	_			
0x41 L cln - Ordre HL 'CloseNet' 0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'		L	opn	-	-		
0x42 L cff - Ordre HL 'CrossFlipFlop' 0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x41	L		_			
0x43 L els - Ordre HL 'EjectLeftSide' 0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x42	L	cff	_			
0x44 L rls - Ordre HL 'RearmLeftSide' 0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x43	L	els	_			
0x45 L ers - Ordre HL 'EjectRightSide' 0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x44	L	rls	-	· · · · · · · · · · · · · · · · · · ·		
0x46 L rrs - Ordre HL 'RearmRightSide' 0x48 L fa - Ordre HL 'FunnyAction'	0x45	L	ers	-	· ·		
0x48 L fa - Ordre HL 'FunnyAction'	0x46	L	rrs	-			
	0x48	L	fa	_			
$ \cup_{\Delta} \exists \cup \sqcup \cup \sqcup \cup \cup$	0x49	L	ln	_	Ordre HL 'LockNet'		
0xBD -	0xBD	_					
0xBE - abort - Termine l'ordre long (ASCII) en cours d'exécution.		_	abort	_	Termine l'ordre long (ASCII) en cours d'exécution.		
0xBF - help - Affiche la liste des ordres.		_		_	i i i i i i i i i i i i i i i i i i i		