

Liste des ordres et format des données

Ici sont listés et décrits les ordres immédiats et longs. « ID » représente l'identifiant d'ordre.

Les ID de 0 à 127 sont réservés aux ordres binaires (ceux décrits dans ce document). Les ordres immédiats et longs peuvent utiliser les mêmes ID. Les ID de 128 à 255 sont réservés aux ordres ASCII, qui sont des ordres de réglage et de débogage.

Le champ « Data » contient une succession d'octets. Lorsque l'on veut écrire un nombre nécessitant plus d'un octet, on commence par les 8 bits de poids fort, et on continue ainsi jusqu'aux 8 bits de poids faible. De manière générale, on considère que le bit de poids fort est toujours transmis en premier. Il est possible d'avoir des champs dont la longueur n'est pas un multiple de 8, toutefois la taille totale du champ « data » en bits doit être un multiple de 8. Le format des données de ce champ varie d'un ordre à l'autre, et est explicité plus bas, en utilisant la syntaxe suivante :

`Nom_du_champ(taille_en_bits) [nom_val_1:valeur1, nom_val_2:valeur2]Nom_du_champ(taille_en_bits)`

(Il y a dans cet exemple, deux nombres transmis, le deuxième prenant ses valeurs dans un sous ensemble valeurs des possibles sur 'taille_en_bits' bits).

`Nom_du_champ` : explicite la signification des données contenues.

`Taille_en_bits` : les données sont écrites sur ce nombre de bits, à la suite du champ précédent *(même si le nombre de bits n'est pas un multiple de 8)*.

Si pour un champ donné, les valeurs inscriptibles sur le nombre de bits donnés ne sont pas toutes des valeurs valides, il est possible de préciser explicitement les valeurs inscriptibles et leurs significations respectives.

`Nom_val_1` : Signification de la valeur #1

`Valeur1` : Valeur #1

Si un nombre est optionnel dans le champ « data », il sera représenté entouré d'accolades : { }

Il n'est possible de rendre un nombre optionnel que s'il est le dernier, ou bien si celui qui le succède est également optionnel.

Le symbole { ... } indique que le champ précédent peut être répété autant de fois que voulu.

VALUE_REQUEST				VALUE_ANSWER
Nom	Description	ID	Data	Data
GetColor	Le bas niveau renvoie la couleur s'il la connaît.	0x59	None	[INTECH:0x00, WINDOW:0x01, UNKNOWN:0x02] COLOR (8)
Ping	Ne fait rien.	0X5A	None	None
AddTrajectoryPoints	Ajoute à la trajectoire courante un certain nombre de points (entre 0 et 31 points).	0x5B	TRAJECTORY_POINT (64) {...}	None
SetMaxSpeed	Règle la vitesse maximale courante.	0x5C	VITESSE_MAX_SIGNEE (16)	None
EditPosition	Modifie la position du robot dans le bas niveau, les coordonnées fournies sont ajoutées aux coordonnées courantes du bas niveau.	0x5D	POSITION_XY (24) ANGLE (16)	None

NEW_ORDER				STATUS_UPDATE	EXECUTION_END
Nom	Description	ID	Data	Data	Data
FollowTrajectory	Suit la trajectoire courante. Se termine une fois arrivé au prochain point d'arrêt de la trajectoire, lorsque le robot est à l'arrêt. Ou bien si « Stop » est appelé.	0x38	VITESSE MAX SIGNEE (16)	None	[ARRIVED:0x00, EXT_BLOCKED:0x01, INT_BLOCKED:0x02, NO_MORE_POINTS:0x03] END MOVE STATUS (8)
Stop	Interromps la trajectoire courante au plus vite. La trajectoire est oubliée. Se termine quand le robot est à l'arrêt.	0x39	None	None	None
WaitForJumper	Se termine lorsque le jumper est retiré du robot.	0x3A	None	None	None
StartMatchChrono	Se termine au bout de 90 secondes si tout se passe bien. Ou bien avant en cas de problème grave.	0x3B	None	None	[MATCH_FINISHED:0x00, EMERGENCY_STOP :0x01] END MATCH STATUS (8)
StreamAll	Envoie la position du robot et/ou l'état des capteurs avec les fréquences demandées.	0x3C	SEND_PERIOD (16) SENSORS PRESCALER (8)	POSITION XYOI (48) { CAPTEURS (?) }	None

Format des variables

TRAJECTORY_POINT - [[POSITION_XYOI](#)(48)] [[IS_STOP_POINT](#)(1)] [[CURVATURE](#)(15)]

IS_STOP_POINT - type : boolean

Indique si le point en question de la trajectoire courbe est un point d'arrêt ou non. Si c'est un point d'arrêt le robot fera en sorte d'avoir une vitesse nulle au moment d'arriver sur ce point, et terminera l'ordre « FollowTrajectory » en cours.

CURVATURE - type : int15_t - unit : hm^{-1}

Courbure de la trajectoire, signée. Le bit de poids fort représente le signe (un 1 signifie « négatif »). Une courbure positive correspond à un virage à gauche, et inversement. L'unité utilisée est « l'inverse de l'hectomètre ». Ainsi le rayon de courbure minimal pouvant être écrit vaut environ 6mm, et la résolution dépasse celle de la mécanique.

CAPTEURS - [[CAPTEUR_LONG_RANGE](#)(8)]*n [[CAPTEUR_SHORT_RANGE](#)(8)]*m

Le nombre de capteurs de chaque type n'est pour l'instant pas connu, d'où les variables 'n' et 'm'.

CAPTEUR_SHORT_RANGE - type : uint8_t - unit : mm

CAPTEUR_LONG_RANGE - type : uint8_t - unit : cm

POSITION_XYOI - [[POSITION_XY](#)(24)] [[ANGLE](#)(16)] [[TRAJECTORY_INDEX](#)(8)]

POSITION_XY - [[X](#)(12)] [[Y](#)(12)]

Représente la position d'un point sur la table de jeu, en mm. Le système de coordonnées standard d'INTech traduit de -1500mm selon l'axe X est utilisé. Ainsi (0,0) représente le point (-1500,0) du repère INTech.

X - type : uint12_t - unit : mm

Y - type : uint12_t - unit : mm

VITESSE_MAX_SIGNEE - type : int16_t - unit : mm/s

Représente la vitesse maximale de translation que le robot est en droit d'atteindre lors de l'exécution d'une trajectoire. C'est également la vitesse qu'il tente d'approcher du mieux possible. C'est une grandeur algébrique, une vitesse négative correspond à une marche arrière.

END_MOVE_STATUS - type : uint8_t

Indique de résultat de l'exécution d'une trajectoire. Le résultat pouvant être : « robot bien arrivé à destination » (0x00), « robot bloqué par un obstacle extérieur » (0x01), « roues du robot bloquées » (0x02), « il n'y a plus aucun nouveau point dans la trajectoire courante, et le dernier n'était pas un point d'arrêt » (0x03).

END_MATCH_STATUS - type : uint8_t

Indique le résultat de l'exécution du match. Cela peut être : « match terminé au bout de 90 secondes comme prévu » (0x00) out bien « match terminé prématurément à cause d'un problème grave » (0x01). Concrètement, le seul 'problème grave' détectable par le bas niveau est le niveau critique de batterie. Mais ceci peut être utilisé pour signaler n'importe quel problème empêchant de terminer le match, et il est inutile de préciser explicitement le problème au haut niveau, de toute façon il ne pourra rien y faire.

SEND_PERIOD - type : uint16_t - unit : ms

Délai entre l'émission de deux trames consécutives. Si la valeur est trop faible, le délai réel sera le délai minimal réalisable compte tenu de l'occupation et de la vitesse de la liaison série.

SENSORS_PRESCALER - type : uint8_t - unit : None

Permet de ne pas envoyer les données des capteurs à chaque trame, toute les SEND_PERIOD millisecondes, mais seulement dans une trame sur SENSOR_PRESCALER. Si la valeur est 0, les données des capteurs ne sont jamais envoyées.

ANGLE - type : uint16_t - unit : milli-radians

Angle dans le repère standard INTech. Entre 0 et 2PI.

TRAJECTORY_INDEX - type : uint8_t

Les points d'une trajectoire sont effectués dans l'ordre de leur TRAJECTORY_INDEX. Il s'agit d'un indice sur un tableau circulaire (l'indice suivant 255 est 0).

COLOR - type : uint8_t

Indique soit le côté de la table où se situe le robot au début du match, soit que cette information n'est pas disponible.