

Topics ROS

La plupart des ordres ont deux topics : un pour envoyer l'ordre, et un pour recevoir le résultat (callback).

- Topics généraux haut-niveau (HN)

/game/color Type : std_msgs/Int16	0 = HOME 1 = AWAY	Définit l'équipe dans laquelle on joue. Permet d'initialiser la stratégie.
/game/init_pos Type : std_msgs/Int16		Définit la zone de départ du robot, parmi les zones de départ possibles. Attention, on compte à partir de 0.
/game/strat Type : std_msgs/Int16		Définit la stratégie à utiliser, parmi les stratégies implémentées. Attention, on compte à partir de 0.
/game/score Type : std_msgs/Int16		Envoie l'estimation du score au cours du match (pour qu'il soit affiché sur l'écran LCD)
/game/start Type : std_msgs/Int16	La valeur doit être 1	Informe les programmes que le match vient de démarrer. Ne pas confondre avec /game/strat.
/game/end Type : std_msgs/Int16		Informe les programmes que le match vient de se terminer.
/park Type : std_msgs/Int16		Informe les programmes que le match est sur le point de se terminer et que le robot doit interrompre l'action en cours et aller se garer.
Type : std_msgs/Empty		

- Topics ordres de déplacement HN – Voir *an_sm_states/sm_displacement.py*

/dsp/order/next_move Type : DisplacementRequest		<i>action_node → displacement_node</i> Envoie un ordre de déplacement, traité par le HN (évitement + pathfinder).
/dsp/callback/next_move Type : std_msgs/Int16	Voir <i>DspCallback</i>	<i>displacement_node → action_node</i> Reçoit le résultat de l'ordre de déplacement
/removeObs Type : std_msgs/String		Supprime un obstacle statique (par ex. une plante après qu'elle a été récupérée)

- Topics de la stratégie

/strat/action/request Type : EndOfActionMsg		<i>action_node → decision_node</i> Demande la prochaine action au decision_node. Le message contient le résultat de l'action précédente
/strat/action/order Type : std_msgs/Int16MultiArray		<i>decision_node → action_node</i> Envoie la prochaine action à l'action_node.
/strat/interrupt Type : std_msgs/Empty		<i>decision_node → action_node</i> Interrompt l'action en cours pour démarrer une autre action (autre que PARK ou END)

- Topics de la détection d'obstacles

/scan Type : sensor_msgs/LaserScan		Données brutes de détection par le lidar.
/sensors/obstaclesLidar Type : SensorObstacleList		<i>lidar_node → déplacement_node</i> Envoie les obstacles détectés par le lidar. (après traitement)
/ultrasonicDistances Type : geometry_msgs/Point	PAS DE SONAR CETTE ANNEE	Données brutes de détection par le sonar
/sensors/obstaclesSonar Type : SensorObstacleList		<i>sonar_node → déplacement_node</i> Envoie les obstacles détectés par le sonar. (après traitement)

- Topics de l'interface

/simu/robotObstacle Type : CircleObstacle	(x, y , radius)	Envoie la position simulée du faux robot adverse à l'interface (simulation uniquement)
/simu/nodegrid Type : std_msgs/Float32MultiArray		Envoie la grille du pathfinder à l'interface (pour affichage)
/simu/deposit_end Type : std_msgs/Empty	OBSOLETE (2024)	Informe l'interface que le pot a été planté dans la zone de culture (pour affichage)

- Topics ordres actionneurs. Type : std_msgs/Int16

Les actionneurs renvoient leur état actuel quand ils ont fini de se déplacer. Si l'état renvoyé ne correspond pas à l'état demandé, le HN en déduit qu'il y a un problème.

/act/order/clamp_1 /act/callback/clamp_1	CLOSE/ OPEN	Ordre et réponse pour le premier étage de pinces (étage inférieur)
/act/order/clamp_2 /act/callback/clamp_2		Ordre et réponse pour le second étage de pinces (étage supérieur)
/act/order/elevator_1 /act/callback/elevator_1	DOWN/ MIDDLE/ UP	Ordre et réponse pour l'ascenseur inférieur
/act/order/elevator_2 /act/callback/elevator_2		Ordre et réponse pour l'ascenseur supérieur
/act/order/banderolle /act/callback/banderolle	DEPLOY	Ordre et réponse pour la banderolle
/act/bumpers	<i>Pas d'ordre associé</i>	Communique l'état des <i>bumpers</i> (enfoncés ou relâchés)
/act/reset Type : std_msgs/Empty	Pas de callback associé	Réinitialise tous les actionneurs dans leur état initial

- Topics ordres base roulante. Voir documentation de la BR.

/br/goTo		Envoie un ordre de déplacement à la BR (asservissement en position)
/br/command		Envoie une commande de vitesse à la BR (asservissement en vitesse)
/br/stop		Envoie un ordre d'arrêt d'urgence à la BR
/br/idle	true=DEMARRE false=ARRETE	Démarre ou arrête l'asservissement de la BR
/br/resetPosition		Initialise (ou réinitialise) la position réelle du robot estimée par la BR Ne pas envoyer pendant un déplacement.
/br/setSpeed	Pourcentage de la vitesse max	Modifie la vitesse du robot <i>uniquement pour le déplacement actuel</i>
/br/gains		Définit les gains du PID (uniquement pour tester d'autres gains que ceux par défaut)
/br/currentPosition		Envoie la position du robot estimée par la BR
/br/callbacks		<i>BR -> displacement_node</i> Envoie le résultat du déplacement au HN
/br/logTotaleArray		Envoie diverses informations de débogage de la BR. (Pas de traitement automatique implémenté)
/br/odosCount		Envoie les valeurs des compteurs de « ticks » des odomètres (pour la calibration)