Arduino documentation

Table des matières

Ballast_fonction(int vitesseG , char rotG , int VitesseD , char rotD)

Contrôle le système de ballast sur base de la valeur de vitesse et de sens de rotation reçu en paramètre pour chaque ballast (droit ou gauche). Mets à jour les timers utilisés pour la durée de fonctionnement des moteurs.

Paramètres

- Vitesse : 0 à 255
- Rotation 0,1,2 : arrêt, sens horlogique, sens anti-horlogique

Exemple d'utilisation

Ballast_Fonction(255, '1', 255, '1');

Les ballasts sont activées à vitesse maximum et se vident en tournant dans le sens horlogique. Le sous-marin remonte donc à la surface.

propulseur_fonction(int vitesseD , int VitesseG)

Contrôle les propulseurs sur base de la valeur de vitesse reçu en paramètre pour chaque propulseur. Mets à jour les timers utilisés pour la durée de fonctionnement des moteurs.

Paramètres

- Vitesse : 0 à 180

Exemple d'utilisation

propulseur_Fonction(180,90);

Le propulseur droit est à puissance maximale, le propulseur gauche est à mi-puissance. L'effet résultant est l'avance du sous-marin en tournant vers la gauche.

decodeCommand()

Implémente le protocole de communication de contrôle des moteurs (ballasts et propulseurs). Décode la commande reçue dans le string « cmd » et appelle les fonctions de contrôle des ballasts ou propulseurs.

Exemple d'utilisation

Dans la loop si une commande ''P18010018010000'' arrive alors **flagCmd == 1**
Par conséquent, **decodeCommand()** est appelée et va déclencher l'activation des propulseurs.

checkMotorDelay()

Vérifie que les délais de fonctionnements des moteurs (propulseurs ou ballasts) sont atteints ou non et coupe les moteurs en fonction.

Exemple d'utilisation

Pour la commande ''P180**100**180**100**00'', les moteurs propulseurs seront allumés pendant 100 déci secondes.

Dans la loop **checkMotorDelay()** coupe les moteurs lorsque 10 secondes sont passées.

MeasureDistance()

Sur base d'un filtre faisant une moyenne sur 3 distances, renvoie la distance mesurée en centimètre par le capteur ultrason entre l'avant de l'appareil et un potentiel obstacle. La valeur de retour est injecté dans la variable global « **distance** ».

Exemple d'utilisation

A chaque tour de loop, la valeur de la distance mesurée est remise à jour. Elle est envoyée à la Raspberry via le protocole de communication ROS sur le topic « **chatter** » toutes les 500ms.

MeasureDepth()

Mesure la profondeur actuelle du sous-marin sur base du relevé de la valeur analogique du capteur de pression, renvoie la profondeur mesurée dans la variable global « **actual_depth** » en mètre.

Exemple d'utilisation

Dans la loop toutes les 500ms, la mesure de profondeur est mise à jour et envoyée à la Raspberry via le protocole de communication ROS sur le topic « **depth** »

MeasureAccelerometer()

Mesure la valeur d'accélération actuelle du sous-marin sur base du relevé de la valeur d'accélération via une communication I2C avec l'accéléromètre. Renvoi l'accélération mesurée dans la variable globale « **Y_out** ».

Exemple d'utilisation

A chaque tour de loop, la valeur de l'accélération est mise à jour. Cette valeur n'est pas utilisée pour l'instant.

AnalogRead()

Mesure la valeur analogique d'une pin et renvoi sa valeur sous forme d'entier. Cette fonction est utilisée pour détecter la présence d'eau et pour lire la tension de la batterie.

Exemple d'utilisation

Dans la loop, toutes les 500ms la tension de la batterie est lue par AnalogRead(battery_pin) et l'information est envoyé au Raspberry.

messageCb(const std_msgs::String &toggle_msg)

Récupère les données inscrites dans la chaine de caractère « toggle_msg » afin de les injecter dans la chaine de caractère « cmd » contenant la commande de moteurs.

A chaque boucle du programme principal, la fonction est appelée si une commande) est publiée sur le topic « /command »

Exemple d'utilisation

Une commande ''P180**100**180**100**00'' est publiée sur le topic ROS « /command » par la Raspberry. Cette chaine de caractère est enregistrée dans la chaine de caractère « cmd » qui est ensuite décodée par la fonction « decodeCommand() ».

Flowchart programme