

## Robots livreurs pour le maintien de la distanciation sociale

### Cahier des charges :

Le but de ce projet est de créer un système évitant les interactions directes entre ouvriers lors d'une épidémie, pour rompre tout contact de la main à la main, toute proximité lors de transmission de documents, de matériel. La distanciation sociale sera assurée par l'utilisation de robots livreurs sur un **parcours en boucle** facile à mettre en place, simplement défini par un **fil électrique scotché** au sol. Des robots tourneront sur ce circuit, chacun avec un numéro d'identification différent. Ils suivront le fil (concept à définir), et recevront des commandes par le signal injecté dans le fil : Robot 12, arrêt au poste 8 à gauche du fil, affiche "livreurA". Robot 1, Vitesse 50% de sa Vmax.

Chaque poste ouvrier comportera un clavier 16 touches de type téléphone sur lequel on pourra composer jusqu'à **4 demandes d'acheminement** en tapant par exemple **#15A\***. **L'étoile signifie nouvelle demande d'acheminement**, le chiffre **15** est le numéro du poste de travail à qui il faut livrer du matériel, **A** est la lettre (parmi A B C D) que le livreur doit déclarer à l'enlèvement.

Un système appelé **la base**, interroge **périodiquement** (au moins 3 fois par seconde) chaque poste de travail pour avoir **un relevé des demandes de livraisons à effectuer** et des **derniers robots vus passant devant le poste**. En modulant le courant dans le fil par un code inspiré du **code morse**, la base sera capable d'avertir les robots et de leur confier des missions. **La porteuse permettra, elle, d'assurer l'asservissement de la trajectoire et donc le suivi du fil.**

Le micro-contrôleur de la **base** communique par **liaison série** aussi avec un opérateur assis devant un écran d'ordinateur (écran du logiciel de supervision) pour lui envoyer des informations:

Le poste 05 vient de voir passer le robot 04 à vide, **dispo** : message P05R4**v**D←

Le poste 03 vient de voir passer le robot 06 à vide, **enlèvecolis** : message P03R6**v**E←

Le poste 01 vient de voir passer le robot 15 chargé, **colispris** : message P01R15**v**C←

Le poste 06 vient de voir passer le robot 11 chargé, **livrecolis** : message P06R11**v**L←

Le poste 04 veut livrer pour le poste 01, le robot se déclarera **A** : message P04**A**P01←

Une application que vous n'aurez pas à réaliser permet à l'opérateur de voir la position courante des robots (on sait qu'il est sur un segment entre deux postes ouvriers et on connaît sa vitesse par la lettre **v** qui code une des **15 vitesses** affichable (**v** lettre hexa de 0 à F)), quel est leur status **dispo**, **enlèvecolis**, **colispris**, **livraison**, ainsi que la liste des demandes de livraison. L'écran lui permet donc de choisir quel est le robot disponible le plus proche du poste où il doit charger.

Il suffit alors à cet opérateur d'envoyer (par le simulateur de liaison série de keil en pratique) un message d'attribution de mission **R02P04**A**P01** pour que la **base** comprenne qu'elle doit :

- envoyer un ordre au robot 2 au status "**dispo**" de s'arrêter au poste 4 afin de charger.

Ce robot devra changer son status en "**enlèvecolis**" dès qu'il recevra l'ordre.

Il devra s'arrêter au poste 4 pour se faire charger en se signalant comme **livreur A**.

Quand le robot repart chargé sur l'ordre d'un ouvrier, il change son status en "**colispris**".

A ce stade le robot est reparti mais ne sait pas à qui il doit livrer...

- envoyer un ordre au robot 2 de s'arrêter au poste 1 dès qu'elle le verra apparaître sous ce status "**colispris**" (identifier par quel moyen elle récupère cette information)

Le robot 2 changera alors de status pour afficher "**livraison**" et reste en tant que **livreur A**

Il devra s'arrêter au poste 1 pour se faire décharger. Aucune évaluation des livreurs n'est prévue dans le système, pourtant le système comprendra que la livraison est terminée.

Quand l'ouvrier lui donnera l'ordre de repartir, le robot changera son status en "**dispo**"

Chaque robot devra émettre en permanence un de ses quatre identifiants sous forme de trame **infra-rouge** de manière à ce que **son numéro, sa vitesse et son status soit identifiable quand il passe devant les postes de travail**. Ceux ci pourront donc déclarer à la prochaine interrogation de la base qu'ils viennent de voir passer le robot X "**dispo**", "**enlèvecolis**", "**colispris**" ou "**livraison**" à la vitesse **v**.

Chaque poste de travail émettra un code DTMF différent via un haut parleur permettant à un robot qui lui passe devant d'identifier son numéro. Un robot qui a la mission de s'arrêter à ce poste saura alors qu'il peut se garer et sortira du circuit.

L'ouvrier peut alors prendre ou poser un colis et renvoie le livreur sur le circuit en appuyant sur un des deux boutons du robot, l'un indiquant "colis récupéré", l'autre "colis chargé".

Les robots mobiles à concevoir seront de type char (la rotation s'effectue par différence de vitesse entre les deux roues motrices) avec une roulette folle à l'arrière. Chaque robot sera muni d'un télémètre à ultrasons à l'avant de manière à éviter de percuter un obstacle sur la trajectoire et de rester à distance des piétons et de leurs postillons pour éviter de plus contaminer ce qu'il transporte.

Chaque groupe d'étudiants devra mettre au point :

- les schémas électroniques et les codes informatique du MC de son robot livreur.
- les schémas électroniques et les codes informatique du MC de la base.
- les schémas électroniques et les codes du Micro-Contrôleur d'un poste de travail.
- réaliser un démonstrateur dans les salles de TP.

Pour cela vous disposerez sous discord d'un salon texte et d'un salon vocal privés. Vous pourrez solliciter des rendez vous virtuels thématiques avec les enseignants en inscrivant vos demandes de visite dans un des salons textes DEMANDE\_VISITE:

Nous ne voulons aucun échange dans ces salons textuels qui servent de FIFO de rendez vous, juste un court message dans le bon flux (le bon salon) : Le groupe10 aimerait discuter de tel point. Nous détruirons les demandes au fur et à mesure qu'elles seront honorées dans leur ordre d'arrivée pour maintenir la lisibilité de ce qui nous reste à faire (si votre message disparaît, c'est qu'un enseignant va arriver dans votre salon vocal et textuel pour échanger avec vous).

L'objectif principal du robot sera de suivre le fil et d'obéir à des commandes transmises via le fil.

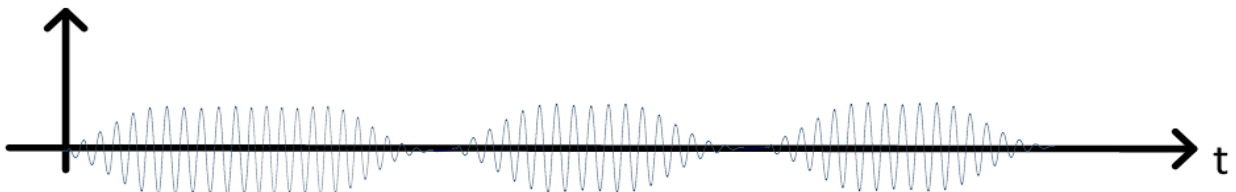
Le signal injecté dans le fil fixé au sol permet à la fois de servir de guide de trajectoire à suivre et permet de transmettre les diverses commandes individuelles. Il faudra identifier les ordres qui sont destinés à son robot et obéir à seulement ceux-là. Des bobines, à étudier à l'aide du document TP préparatoire, permettent de capter par induction, les informations nécessaires à l'asservissement. Les différents ordres individuels à transmettre par le fil seront :

- Robot numéro N, change immédiatement de vitesse  $v$ .
- Robot numéro N, sort de la piste à gauche au poste de travail numéro P, tu es livreur A.
- Robot ,numéro N, sort de la piste à droite au poste de travail numéro P, tu es livreur C.
- Robot numéro N, active ou désactive le mode « debug » par liaison série de telles et telles fonctions. Chaque fonction sera associé à un bit dans le message.

Le retour sur la piste d'un robot arrêté se fait après appui sur un des boutons embarqués sur ce robot, l'intervenant ayant la responsabilité du choix et devant s'assurer que la voie est libre .

Le signal injecté permet de faire l'asservissement. La technique suggérée pour envoyer les commandes consistant à utiliser un codage tout ou rien, signal émis ou pas dans le fil, il faudra faire attention à quels instants l'amplitude du signal capté dans les bobines est réellement disponible et fiable pour les calculs de distance au fil nécessaires à l'asservissement.

$I(t)$



Les signaux injectés dans le fil seront de forme rectangulaires de rapport cyclique 50% avec une période de 20 $\mu$ s, mais ils seront injectés via une association RLC série de manière à rendre le courant le plus sinusoïdal possible. Cela a des conséquences au niveau de la réception, mais aussi au niveau du temps nécessaire pour que le courant disparaisse ou apparaisse si le facteur de qualité est trop important.

### Cahier des charges des différentes parties à réaliser dans le projet complet :

#### Télémètre à ultrasons embarqué sur le robot:

- Au niveau du micro-contrôleur, on devra pouvoir choisir d'émettre ou pas. En mode désactivé, aucun ultrason ne doit être émis.
- Il doit être capable de faire une mesure de 5cm à 2,5m d'un obstacle plan de 20cm\*20cm
- La précision souhaitée est d'un cm environ.
- Le nombre de mesures par seconde doit être contrôlable depuis le micro-contrôleur. Il sera configuré par 2 entrées numériques (micro-switchs) sur le micro-contrôleur : vous obéirez au tableau de configuration suivant :

Combinaison	Nombre de mesures souhaitées
'11'	10 mesures par seconde
'10'	15 mesures par seconde
'00'	20 mesures par seconde
'00'	25 mesures par seconde

- Des informations de « debug » devront être envoyées par port série à chaque mesure sous la forme d'une trame de caractères ASCII suivi d'un retour à la ligne : Le mode de debug est sélectionné prioritairement par deux micro-switchs ou par commande venant du fil (à venir).

Combinaison	Mode de debug pour le télémètre
'11'	Aucune trame envoyée, le choix d'un mode de debug par envoi de commande via le fil est seulement possible dans cette combinaison.
'10'	Nombre de mesure seconde exemple « T 20 mes/sec »
'01'	Donnée brute mesure distance en cycles de comptage du timer : exemple « T0xFFFFF »
'00'	Distance en cm exemple « T 125 cm », « T 18 cm »

#### Gestion des quatre identifiants, numéro de robot configuré par switches, code émis en IR:

- Il est impératif d'envoyer des trains d'impulsions de manière à émettre puissamment. Le codage sera donc sur le principe suivant : Des trains d'impulsions infra-rouge seront émis en injectant un fort courant dans une led IR.
- Les robots émettront des codes sur 12 bits (3 lettres hexa), suivi de 4 bits de contrôle, checksum égal au complément à 2 de la somme sur 4 bits des 3 quartets précédents.
- Le code émis par la carte obéit au chronogramme suivant :

Dans la description qui suit chaque lettre (autre que l'espace utilisé pour faciliter la lecture) représente un temps  $\tau$  (compris entre 150  $\mu$ s et 300  $\mu$ s), x signifie 0 ou 1 indifféremment.

- \* La séquence **xx0 10 1xx** représente un **Start** (10) pour synchroniser ou re-synchroniser la transmission. Un train de **Nimp** impulsions pendant  $\tau$  puis un blanc d'émission pendant  $\tau$ .
- \* La séquence **xxx0 110 1xxx** sert à transmettre un « 1 » logique : 2 Nimp impulsions + blanc
- \* La séquence **xxx0 100 1xxx** sert à transmettre un « 0 » logique : Nimp impulsions + 2 blanc

Le numero de robot sera émis sur les 4 premiers bits, sa vitesse sur 4 bits (le pourcentage par palier arrondi de 5% en 5% avec un minimum de 20% ), les 4 status sur les 4 suivant (1 seul bit à 1)

**xx0 10 100 110 110 110 100 100 110 110 110 100 100 100 110 110 110 100 10 xx0**  
code "Je suis le robot 7 à la vitesse 3 (20+ 3 \* 5 = 35%) en status 1000 "

Les 16 bits sont transmis poids fort d'abord. Vous pouvez constater que la durée entre deux fronts montants est de deux  $\tau$  pour la transmission d'un start et de trois  $\tau$  pour la transmission d'un 0 ou 1.



Combinaison	Mode de debug pour le télémètre
'11'	Aucune trame envoyée, le choix d'un mode de debug par envoi de commande via le fil est possible.
'10'	Valeurs des PWM en décimal exemple « G 350 D 450 »
'01'	Vitesse moyenne en cm/s exemple « V 30cm /s »
'00'	Vitesse angulaire en degrés par seconde exemple « W -5deg/s »

Capteurs inductifs embarqué sur le robot mobile, système d'analyse de position par rapport au fil:

- Le système doit permettre de positionner précisément le robot par rapport à un fil droit considéré infini comme dans l'analyse du TP préparatoire. Reprendre ce texte tant que vous n'avez pas compris quelles sont les informations que l'on doit reconstituer et quel est le jeu de capteurs à mettre en place.
- Le système doit aussi permettre de récupérer les commandes envoyées aux robots via les courants injectés dans les fils. Sachant que certaines bobines ne captent rien dans certaines positions et que le principe du codage consiste à émettre puis ne plus émettre, il faudra judicieusement choisir la bonne bobine pour récupérer ces informations.
- Pour chaque capteur mis en oeuvre, on souhaite une résolution de mesure de l'ordre du mm et une précision de l'ordre du cm. Attention donc à l'instant où la mesure est stable !!! On pensera à lever une patte du micro-contrôleur au moment où la mesure est effectuée de manière à prouver que la mesure est faite au bon moment.
- Par une combinaison **prioritaire** de 3 switchs ou via la commande reçu par le fil, des informations de debug devront être envoyées par port série sous la forme d'une trame comportant plusieurs caractères ASCII:

Combinaison	Mode de debug pour le détecteur de position
'111'	Aucune trame envoyée, le choix d'un mode de debug par envoi de commande via le fil est possible.
'110'	Mesure brute de l'entrée analogique 1 exemple « A 0xFFF »
'101'	Mesure brute de l'entrée analogique 2 exemple « B 0xFFF »
'100'	Mesure brute de l'entrée analogique 3 exemple « C 0xFFF »
'011'	Valeur calculée de la distance capteur1 / fil en mm exemple « x123mm »
'010'	Valeur calculée de la distance capteur2 / fil en mm exemple « X-12mm »
'001'	Valeur calculée de l'angle robot/fil en degrés exemple « a-10° »
'000'	Durée de l'impulsion code morse exprimée en $\mu s$ : exemple « I 2500 »

### Sécurité robot, système d'arrêt d'urgence par arrêt coup de poing et reprise de la trajectoire:

- Le robot ne doit pas bouger (aucun soubresaut) à la mise sous tension.
- Le robot sera muni d'un arrêt coup de poing (coupe circuit, fil que l'on arrache) qui interviendra au niveau de l'électronique directement mais qui **préviendra le software**.
- La reprise de trajectoire, que ce soit après libération de l'arrêt coup de poing ou parce qu'un opérateur a fini de charger/décharger le robot, **se fait par l'appui sur boutons**.
- Le robot se déclarera A, B,C, D pour que le poste de travail puisse charger le bon colis : Un poste peut demander jusqu'à 4 robots pour 4 destinations différentes **sans attendre d'avoir expédié son premier colis. Il doit juste se souvenir quel colis mettre sur A, B, C, D ...**
- Le robot est muni d'un avertisseur sonore (fréquence 1KHz) qui se déclenche dans deux cas :
  - Pour avertir un ouvrier qu'il doit intervenir et charger ou décharger le robot : les beeps seront d'une durée de 0,75 seconde espacés d'une demi seconde. Un seul pour décharger , deux pour charger.
  - En cas d'obstacle : des beeps de 0,25 seconde de plus en plus rapprochés dans le temps sont émis en fonction de la proximité de l'obstacle conformément au tableau suivant :

distance	100cm	90cm	80cm	70cm	60cm	50cm	40cm	30cm	20cm
fréquence	0,5Hz	0,75Hz	1Hz	1,25Hz	1,5Hz	1,75Hz	2Hz	2,25Hz	2,5Hz

### Borne réceptrice infra-rouge, permettant de détecter le passage du robot numéro N et son status :

- La chaîne électronique de réception permet de reconstituer l'enveloppe de la trame identique à celle décrite dans la section gestion identifiant robot. la durée à état haut **code E , 1 , 0.**
- Une led s'allumera pour deux secondes quand une trame valide est reçue.
- Quatre leds afficheront le numéro de robot de la dernière trame valide reçue.
- Via un patte du micro-contrôleur, vous fournirez un top de synchronisation pour l'oscilloscope quand vous détectez un entête de message.

### Echanges entre la base, les postes de travail, la supervision :

- la base va interroger les postes de travail les uns après les autres en envoyant une notification @Tnn par liaison série à tous les postes de travail numérotés de **01 à 99**. Ces notifications seront suivies de deux caractères pour un retour à la ligne 0x0A 0x0D.
- chaque poste compare ce numéro à son propre numéro qu'il lit sur ses entrées GPIO. Si c'est son tour de parler,il repond par liaison série toujours 6 caractères (les deux derniers sont 0x0D 0x0A pour provoquer un retour à la ligne symbolisé par ← ) dont les 4 premiers sont:
  - RXvV← si lui, poste nn vient de voir passer le robot X vitesse v avec un status **"dispo"**
  - RXvE← si lui, poste nn voit passer le robot X de vitesse v avec un status **"enlèvecolis"**
  - RXvC← si lui, poste nn voit passer le robot X de vitesse v avec le status **"colispris"**
  - RXvL← si lui, poste nn voit passer le robot X de vitesse v avec le status **"livraison"**
  - lettrePyy← si lui, poste nn vient d'avoir une demande d'acheminement vers le poste yy. Le robot affecté à cette mission devrat se présenter en affichant une lettre parmi {A,B,C,D}
  - NULL← si lui, poste nn n'a aucune nouvelle information à donner non déjà transmise.
- La base va répéter à l'opérateur de supervision tous les messages sauf les messages NULL en rajoutant Pnn devant les caractères reçus. exemple : PnnRxxV← ou PnnCPyy←



- La base écoutera les consignes de vitesse de l'opérateur de supervision sous la forme d'un message **RxxVyy←** (retour à la ligne). L'opérateur peut ainsi répartir les robots sur le tour.
  - Elle tiendra à jour un tableau **vitesse\_voulue[]** qui permettra d'inscrire à l'index **xx** la consigne de vitesse du **robot xx**
  - Elle tiendra à jour un tableau **vitesse\_actuelle[]** qui contiendra la dernière vitesse reçue envoyée par chaque robot à son passage devant un poste de travail.
  - Si la vitesse voulue diffère de la vitesse actuelle pour un robot, un message dans le fil lui sera transmis au plus vite et le **tableau vitesse\_actuelle** sera mis à jour à réception de **v**.
- La base écoutera les messages d'attribution de mission **RxxPss-Pdd←** (message suivi d'un retour à la ligne ) et tiendra à jour deux tableaux de taille le nombre de robots:
  - Le premier **enlèvement[]** permet d'inscrire la valeur ss à l'index xx indiquant ainsi que le robot xx doit faire un enlèvement au poste ss.
  - Le second **livraison[]** permet d'inscrire la valeur dd à l'index xx indiquant ainsi que le robot xx doit faire un dépôt au poste dd.
  - Pour indiquer dans un des deux tableaux que le robot xx n'a rien à faire ou a réalisé sa mission, inscrire qu'il doit faire son enlèvement, sa livraison au poste 0 qui n'existe pas.
- Dès qu'un robot est inscrit pour faire un enlèvement, on lui demande via le fil de s'arrêter pour un chargement au poste ss du bon côté du fil.
- Dès qu'un poste signale que le robot xx n'est plus vu en status "**dispo**" mais "**enlèvecolis**", la base peut écrire dans **enlèvement[xx]** que le robot xx n a plus d enlèvement à faire (il a compris l'ordre, ce sera inutile de lui répéter)
- Dès qu'un robot xx est inscrit pour faire une livraison et qu'il est en status "**colispris**" , on sait qu'il a effectué la moitié de sa mission et on peut donc lui dire de rejoindre **livraison[xx]**
- Dès qu'un robot passe en status "**livraison**", on peut le retirer de la liste **livraison[]** en écrivant la valeur 0 dans ce tableau à l'index de même numéro que le robot .

#### Communication avec les robots via les signaux injectés dans le fil

- Les signaux injectés dans le fil seront des signaux carrés à **50Khz**, filtrés par un **RLC**.
- Les ordres sont envoyés bit à bit. Les motifs E,0,1 permettent d'envoyer le code.

motif	E comme entête de message	0	1
émission du 50kHz puis pause	<b>émission pendant 2,5 ms</b> <b>puis pause 0,5ms</b>	émission pendant 1 ms puis pause 0,5ms	émission pendant 1,75 ms puis pause 0,5ms

- Vous changerez d'état une patte du micro-contrôleur en début de message de manière à favoriser la visualisation d'un message émis en simulation
- Un message sera émis puis en absence de nouveau message à émettre, des **entêtes seront émis en boucle**.

#### Configuration de la base et de la supervision :

La base doit connaître combien de robots sont disponibles pour être pilotés et combien il y a de postes de travail à interroger et desservir. Cela est fait par des DIP SWITCHs.

De même la supervision est un logiciel qui lit un fichier de paramètres de type XML pour ajuster automatiquement son écran.

Vous n'aurez pas à réaliser ce programme, simulé par le terminal série dans le logiciel KEIL.

#### Ordres à envoyer aux robots :

- Les ordres à envoyer aux robots commencent par un entête pour permettre une synchronisation de la réception.

- Chaque ordre indiquera le numéro du robot à qui il est destiné en 4 bits, bit de poids fort en premier.
- Le type d'ordre sera sur 3 bits.
- Enfin viendra le paramètre de la commande souhaitée sur 7 bits.

Les robots changeront de status à réception des ordres émis dans le fil en fonction de si on leur demande un chargement, un déchargement. Le redémarrage par appui de bouton par l'ouvrier provoque aussi un changement de status. enlèvement / livraison fait passer le robot par les quatre status.

**Type de message 000** : Demande au robot xx de se mettre à nn % de sa Vitesse maximale

Demande R02V40 venant du superviseur : robot 2 à vitesse 40% : **E 000 0010 0101000**

E est l'entête, 000 est le type de message, 0010 est le robot 2 en 5 bits, 0101000 code la valeur 40

**Type de message 001** : Demande au robot xx de se garer à gauche du fil au poste yy pour un chargement : **E 001 xxxx LL yyyyyy** avec xxxx le numéro de robot en 4 bits, yyyyyyy le numéro de poste auquel s'arrêter en 5 bits, LL 2 bits codent les 4 possibles A B C D

**Type de message 010** : Demande au robot xx de se garer à droite du fil au poste yy pour un chargement : **E 010 xxxx LL yyyyyy** avec xxxx le numéro de robot en 4 bits, yyyyyyy le numéro de poste auquel s'arrêter en 5 bits, LL 2 bits codent les 4 possibles A B C D

**Type de message 011** : Demande au robot xx de se garer à gauche du fil au poste yy pour un déchargement : **E 011 xxxx LL yyyyyy** avec xxxx le numéro de robot en 4 bits, yyyyyyy le numéro de poste auquel s'arrêter en 5 bits, LL 2 bits codent les 4 possibles A B C D

**Type de message 100** : Demande au robot xx de se garer à droite du fil au poste yy pour un déchargement : **E 100 xxxx LL yyyyyy** avec xxxx le numéro de robot en 4 bits, yyyyyyy le numéro de poste auquel s'arrêter en 5 bits, LL 2 bits codent les 4 possibles A B C D

**Type de message 101 à 111** : Messages de demande de mise à jour des modes debug (à définir)

Codes DTMF émis par les postes de travail :

- L'émission DTMF sera assurée par vos soins depuis le micro-contrôleur, en générant le son directement par le micro-contrôleur. La réception se fera via un composant [HT9170D](#)
- Vous devrez réaliser vos amplis sur lequel brancher le haut parleur. Rappel, le micro-contrôleur n'est capable de générer qu'un signal de faible amplitude compris entre 0V et 3V3.
- Dans le micro-contrôleur, on dispose de peu de mémoire, comment générer les bons signaux à émettre sans consommer une place folle en mémoire à enregistrer tous les sons possibles.
- Au niveau réception, vous devrez afficher via des leds le numéro de poste reconnu.
- Sachant qu'il peut y avoir plus que 10 postes ouvriers, comment coder et avertir le robot du numéro du poste en face duquel il est, sans confusion dizaine, unité si on répète en boucle ?
- Comment assurer la synchronisation de la transmission ?

Affichage de la lettre de service pour que le poste de travail reconnaisse le robot et status visible

Cet affichage se fera par rotation d'un servo moteur qui permettra d'afficher une des quatre lettres via un texte "livreur A" "livreur B" "livreur C" "livreur D" ou "\_\_\_\_\_" si le robot n'a rien à faire. L'affichage du status se fera à l'aide d'une led RGB qui sera pilotée de la bonne couleur. Ainsi on verra comme pour les taxis quel est le status actuel du robot :

"dispo", "enlève colis", "colis pris" ou "livraison"



Codeurs Incrémentaux, odométrie :

- A l'aide d'une carte Basys2 munie d'un FPGA, vous réaliserez un système odométrique permettant de mesurer la distance parcourue.

- Les codeurs incrémentaux des moteurs sont les suivants :

<http://www.lextronic.fr/P3102-moteur-reducteur-avec-encodeur.html>

- La carte basys2 fonctionnant en 3V3, il faudra faire une adaptation de niveau logique entre les codeurs et la carte basys2.
- La communication entre la carte basys2 et l'arm7 se fera par une liaison SPI, l'arm7 étant celui qui délivrera l'horloge CLK et les signaux de contrôle Cs.
- Vous mesurerez :
  - La vitesse de la roue droite par mesure du temps entre deux tops codeurs
  - La vitesse de la roue gauche par mesure du temps entre deux tops codeurs

Les codeurs étant imparfaits, vous ferez cette mesure une fois par tour moteur (regarder la périodicité à l'oscilloscope) de manière à avoir une mesure stable si le moteur tourne à vitesse constante. A savoir, par la nature du codeur, le signal envoyé n'est pas régulier même si la vitesse de la roue ne varie pas . On observe une régularité des valeurs toutes les trois mesures : MesA MesB MesC MesA MesB MesC .....

il faut faire la moyenne de trois mesures consécutives pour avoir une valeur stable si la vitesse du moteur ne change pas.

A quel endroit est il le plus simple de faire cette moyenne ?

- Le nombre de pas effectué par la roue droite en 1/50 ième de seconde
- Le nombre de pas effectué par la roue gauche en 1/50 ième de seconde
- La carte Basys2 préviendra l'arm7 qu'elle dispose d'une nouvelle information valide 50 fois par seconde
- Les informations de vitesse permettront de calculer en temps réel la vitesse moyenne du robot et donc d'agir sur les PWMs commandant les roues pour que cette vitesse moyenne respecte la consigne souhaitée par la supervision.
- En fonction de l'état des interrupteurs sur la carte, les afficheurs de la carte afficheront les diverses informations Vg, Vd, Pg, Pd ...
- L'arm7 interrogera la carte à l'aide de 4 CS différents pour lui indiquer quelle information elle veut récupérer par la liaison série synchrone de type SPI
- Vous ne mettrez pas en œuvre le périphérique SPI, vous vous contenterez de générer à la main les chronogrammes nécessaires. on créera des fonctions TOP\_CLK(), LIRE\_bit(), ...

Description du fonctionnement de l'écran de supervision (pas à réaliser mais à comprendre) :

L'écran de la supervision affiche une vue symbolique du système :

- Un ovale découpé en sections par des traits symbolise le circuit. On appelle section le morceau de fil au sol qui relie un poste de travail à un autre.
- Les postes de travail sont symbolisés par l'inscription Pxx à proximité des traits décrivant en fait la position du poste de travail sur le circuit.
- Les robots sont représentés par un rectangle avec un numéro et une tâche de couleur symbolisant le status du robot : "**dispo**", "**enlèvecolis**", "**colispris**", "**livraison**",
- La taille de la tache indique la vitesse du robot comme un bargraphe de 20% à 80%
- Les robots sont dessinés :
  - soit à cheval sur le ovale sur une section entre deux traits.
  - soit posés en dehors du circuit à proximité du poste où ils sont arrêtés.
  - un robot jamais, ou non signalé par un poste depuis plus de 2 minutes disparaît.
- Une flèche verte s'allume au moment où l'on reçoit un message de passage d'un robot devant un poste :
  - le long du ovale , de part et d'autre du trait symbolisant le poste de travail
  - hors du ovale indiquant la sortie du robot du circuit si l'information est issue d'un poste de travail au près duquel la supervision sait que le robot doit s'arrêter.
- La flèche verte s'allume, puis une seconde plus tard, le robot change de position (c'est à dire de section), puis une seconde plus tard la flèche s'éteint.
- Les robots présents dans une même section sont affichés dans un ordre logique, on postule qu'ils ne se doublent pas sur le circuit. Ils peuvent juste doubler un robot arrêté.

Sous ce ovale graphique, il y a un tableau dynamique de 4 colonnes qui liste toutes les missions :

- La première colonne nommée liste trajets contient une chaîne de texte au format : **Pss --> Pdd** où **ss** est le numéro de poste expéditeur, et **dd** le numéro du destinataire.
- La seconde colonne indique :
  - soit le numéro de robot affecté à la mission
  - soit "?" si aucun robot n'a encore été affecté à la mission
- La troisième colonne avec quelle lettre A B C D, le robot devra se présenter au départ.
- la dernière colonne est le dernier status du robot reçu par la supervision par un message de la base du type PyyRxxS ( yy numéro porte, xx numéro robot, S lettre status robot)
  - ? si aucun robot affecté à la mission
  - **V** si le robot encore cru vide "**dispo**", aucune porte ayant vu un changement de status.
  - **E** pour un robot vu par une porte à laquelle il se déclare "**enlèvecolis**",
  - **C** pour un robot vu par une porte à laquelle il se déclare en status "**colispris**",
  - **L** pour un robot vu par une porte à laquelle il se déclare en status "**livraison**"

Sous cette première liste dont le nombre d'éléments évolue dynamiquement, il y a une autre liste :

La liste des robots réellement disponibles pour qu'on leur attribue une mission affichée sous la forme d'une chaîne de caractères R01 R02 .... R04 .... .... R07 .... (.... = indisponible )

En cliquant sur un "?" de la liste des missions, puis sur un numéro de robot dispo, ce robot est affecté à la mission **Pss-->Pdd** correspondante :

- Le logiciel de la supervision envoie un message **RxxPssZPdd** à la base (vous aurez à simuler ce message dans le simulateur de port série de keil) Z = lettre parmi {ABCD}
- Le robot xx sort de la liste des robots disponibles, son nom est remplacé par ....
- La ligne du tableau dynamique des missions est mise à jour :

Les deux "?" sont remplacés par **Rxx** et **V**

Lorsqu'une porte envoie un message indiquant qu'elle a vu notre robot, son status est mis à jour.

Lorsque le robot est passé par tous les status "**enlèvecolis**", "**colispris**", "**livraison**" et qu'il redevient "**dispo**", cela veut dire que la mission est réalisée, le colis a été récupéré et livré :

- la mission **Pss-->Pdd** sort de la liste des missions en cours.
- **Rxx** retourne dans la liste robots dispos, prêt à être affecté à une nouvelle mission.