

→ Commandes des roues de robot: position servo

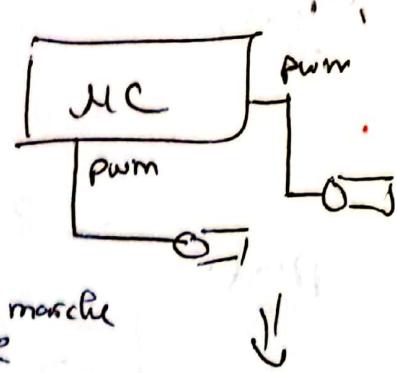
- vitesse variable, PWM, T~~≤~~ 50ms, pas de霍ne 4R
- La vitesse dépend de:
 - vitesse play : via le fil, $25 \leq V_{play} \leq 80$ % de la vit Max
 - vitesse angulaire ω_{rot} : est la vit du robot quand il tourne sur lui même, pour corriger la trajectoire du robot.
- ω_g, w_g : proportionnelle aux PWM.
 - e : eror
 - r : moyen des roues.
- Asservissement: prendre la bonne orientation Y. aux fil.
 - corriger les pos du robot.
 - le correcteur fait en fait une distance parcourue etc.
- Debug: chaque 50 fois / seconde. trace ASCII + 
 - Mode (2 uswitch) : au pos commande du fil.

La Commande sera issue de

2 Variables.

V_{moy} : | via le file → de V_{moy, rive}
95% V_{moy-min}
80% V_{moy-max}

pour deux marche
arrière

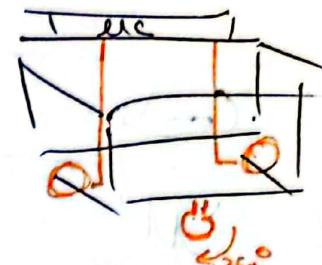


W_{moy}

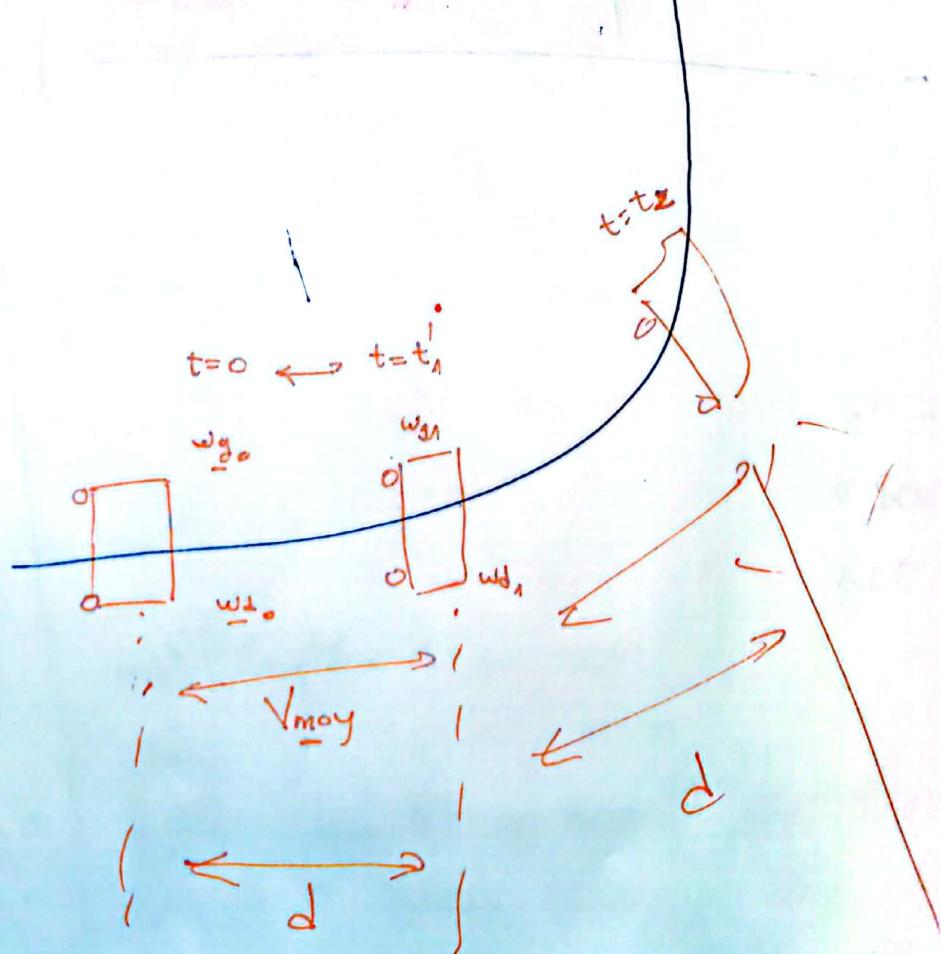
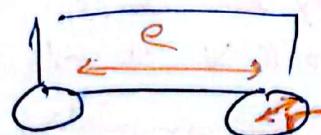
Proviennent de l'asservissement

Asservissement de Trajectoire

- On recalcule $\omega_g, \omega_d \rightarrow$ proportionnelles
paramètres pertinents :



écart



1)

DEBUG: \rightarrow Port série ($f = \frac{1}{50} \text{ hz}$)

Trames de chars ASCII arrivé d'un fil

2 switch

mode

\rightarrow pas cm/s venant du fil

Comb:

'M'

Acune Trame, mode de debug pour envoyer fil

'10'

PWM en decimal

Ex: < 350 > 450 >

'01'

Vmoy en cm/s.

Ex: < V 30 cm/s >

'00'

vitess angulaire en d/s

< w - 5 deg/s >



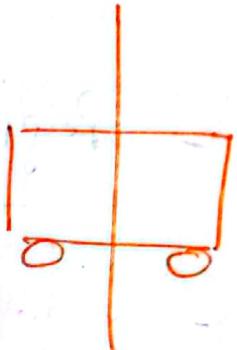
Captur inductif. (TP préparatoire)

Système d'analyse de la position du Robot par rapport au fil.

Le système doit aussi permettre de recuperer les commandes envoyées au robot via le fil.

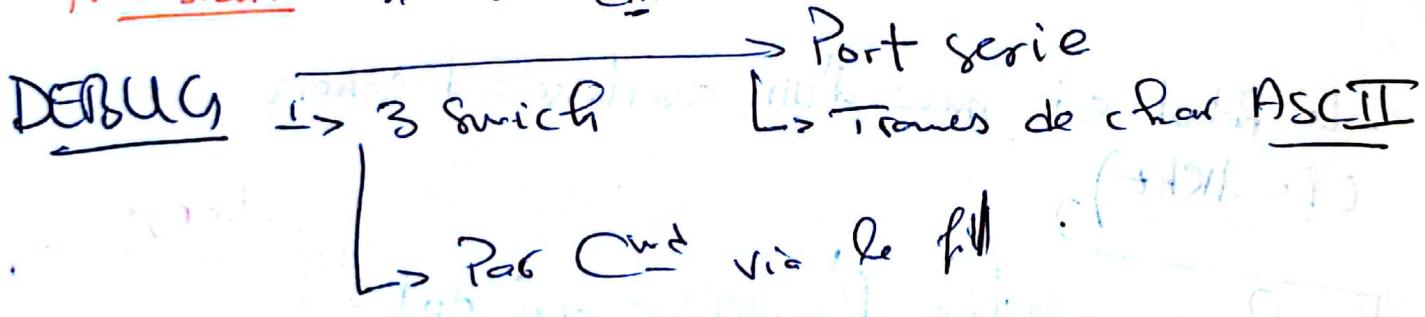
Remarque

Certaines bobines ne captent rien dans certaines positions.
Le principe de codage : émit ou n'émet rien



- Résolution, ordre des mm.

Precision: " " cm



Comb	Mode de debug
'111'	Acme Trace envoyée, debug par envie fil
'110'	mesure brûle Entrée Analog 1 Enr <A0xFFFF>
'101'	" " Entrée Analog 2 Enr <B0xFFFF>
'100'	" " " " = 3 Enr <C0xFFFF>
'011'	dist Captur 1 en <u>mm</u>
'010'	" " 2 en <u>mm</u>
'001'	Angle robot/fil en <u>degré</u> . <a-10>
'000'	Durée de l'impulsion code morse en <u>μs</u>

- Securité robot, système d'arrêt d'urgence
Par arrêt coup de poing et reprise de la trajectoire.
- le robot ne doit pas bouger à la mise sous tension.
 - le robot sera muni d'un arrêt coup de poing.
(coupe circuit, fil que l'on arrache) $I \xrightarrow{\text{dec}}$
 \hookrightarrow permet le software
 - Reprise sur Appui. Sur boutons. (Opris + fil arraché)
 - Le Robot se déclare A,B,C,D pour que le poste charge le bon colis. (Un poste peut demander jusqu'à 4 Robot)

Le Robot est muni d'un avertissement sonore ($f = 1\text{kHz}$)

\hookrightarrow Pour avertir l'ouvrier qui doit

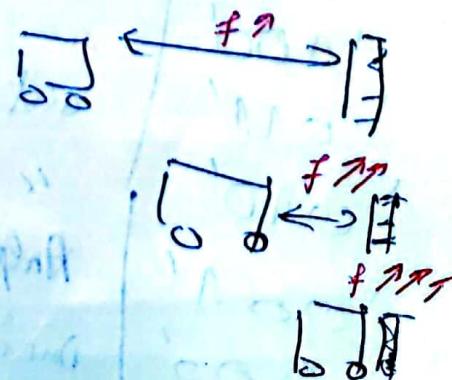
\hookrightarrow Charger: Deux bip

\hookrightarrow Décharger: Un bip

- En cas d'obstacle

\hookrightarrow De plus en plus rapprochés dans le temps sont emis.

Dis	100cm	---	20cm
f	0,5Hz		2,5Hz



Borne Receptrice Infra Rouge

Déterminer le passage du Robot + Numéro + Status.

- la chaîne électronique de réception permet de reconstruire 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 (2s)

- 4 LED afficheront le num du Robot de la dernière trame valide.

- Via une pente du microcontrôleur, vous fournirez un top de synchro pour l'oscillo, quand on détecte une entête

Echange 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 notif @tnn par liaison série à tous les postes de travail (0 à gg) ces notifs seront suivis d'un retour à la ligne 0x0A 0x0D;

• Chaque poste compare ce numero à son propre num lit sur ses EEPROM.

→ Si c'est son tour de parler, il répond par liaison série 6 caractères (2 derniers sont 0x0D 0x0A)

- RxvV → si oui, poste nn. vous passer le robot X
de vitesse v avec un status "dispo"

[Lettre Pyy] si Oui, poste nn vient d'avoir une demande
d'acheminement vers le poste yy.

→ le Robot affecté à cette mission devra se présenter
en affichant une lettre parmi {A,B,C,D}.

- La Base va répéter à l'opérateur de supervision tous les
messages sauf msg NULL, en rajoutant Pnn devant les
caractères reçus, Pnn RxvV ← Pnn CPyy

- La Base écoutera les consignes de vitesse de l'opérateur
de supervision sous la forme d'un msg RxvVyy.
L'opérateur peut repartir les robots sur le tour.

- Elle tiendra à jour un tableau vitesse-voulue [] qui permettra
d'inscrire - l'index xx la consigne de vitesse du Robot xx

- " " " " vitesse-actuelle []. La
dernière vitesse reçue envoyée par chaque robot à son
passage devant un poste de travail

Si vitesse-voulue [xx] ≠ vitesse-actuelle [xx]:

robot \Leftarrow vitesse-actuelle [xx] = vitesse-voulue [xx];

La Base écoutera les message d'attribution de mission

RxxPss-Pdd (suite d'un retour à la ligne) et permet d'inscrire
et tiendra à jour deux tableaux.

{ enlèvement [xx] ← ss
{ livraison [xx] ← dd

~~ff~~ (robot ne fait rien)

enlèvement [xx] $\leftarrow 0$;

livraison [xx] $\leftarrow 0$;

- dès qu'un robot est inscrit pour faire ~~une livraison~~ enlèvement, on lui demande via le fil de s'arrêter au poste de chargement ss du bon côté du fil.
- dès un poste signale que le Robot xx n'est plus Dispo mais : "enlevé colis", la base peut écrire dans enlèvement [xx] que le Robot xx n'a plus d'enlèvement à faire (Il a compris l'ordre). enlèvement [xx] $\leftarrow 0$;
- Dès qu'un robot xx est inscrit pour faire une livraison et qu'il est en status "colipris", on peut lui dire de rejoindre livraison [xx].
- Dès qu'un robot passe en status livraison : ~~on peut retirer de la liste~~ ~~livraison~~ livraison [xx] $\leftarrow 0$;

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

- les signaux injectés dans le fil seront des signaux carrés à 50 kHz, filtrés par PLC
- les ordres sont envoyés bit à bit. les motifs E;0;1 permettent d'envoyer le code.

motif	E comme entête	0	1
Emission du 50kHz puis pause	Emission pendant 2.5ms puis pause 0.5 ms	Emission pendant 1ms, puis pause 0.5 ms	Emission pendant 1.17ms, puis pause 0.5 ms

Configuration de la Base et de la supervision.

La Base doit connaître combien de robot sont dispo pour être pilotés et combien il y a de postes de travail à interroger et disservir. Cela est fait par des DIP

SWITCH's.



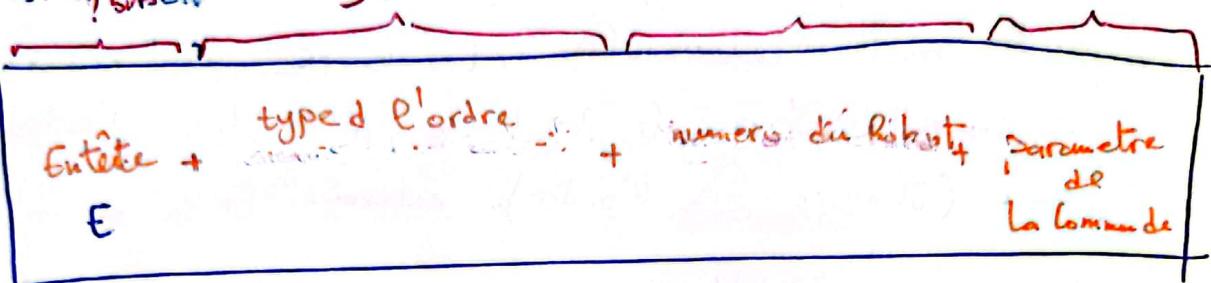
switches.

Ordre à envoyé au Robot

- message type 3 bits

4 bits

7 bits



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

Demande R02V40 ; Robot 2 à vitesse 40%. E 000 0010 0101000

→ 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 yyyy, avec xxxx le num du robot en 4bits

yyyy le numero du poste auquelle s'arrêter en 5bits

LL 2 bits contient les 4 possibles A B C D.

→ Type de message 010 : Demande au robot de se garer à Droite du fil au poste yy pour un chargement

E 010 xxxx LL yyyy, avec xxxx le numero du robot en 4bits

yyyy le numero du poste auquelle s'arrêter en 5bits

- Type de message 011 . Robot xx se gare à gauche du fil , poste yy pour déchargement
- Type de message 100 Robot xx se gare à droite du fil , poste yy pour déchargement

type de msg	Robot (bits)	Côté du fil	fonction	poste (5 bits)
000	xx	gauche	commande de vitesse	yy
001	xx	gauche	chargement	yy
010	xx	droite	chargement	yy
011	xx	gauche	déchargement	yy
100	xx	droite	déchargement	yy

Ex: 101 XXXX aa bb ccc
tele PNM confiditio

Type de message 101 à 111 , Demande de mise à jour des mode debug (à définir)

Code DTMF émis par le poste de travail :

- Assuré par le MC , en générant le son directement par le MC , la réception sera assurée via un composant HT9170D
HT9170D : Recepteur DTMF , résonateur

- Amplis sur lequel brancher le haut parleur [MC génère $E \approx 3.3V$]

- Dans le MC , la mémoire est limitée , (Emetteur) Port:

⚠ Q: Comment générer les signaux à envoyer à envoi enregistrer tous les sous possible ? pour chaque poste on prend des fréquences dizaine (2 freqs) unité (1 freq).

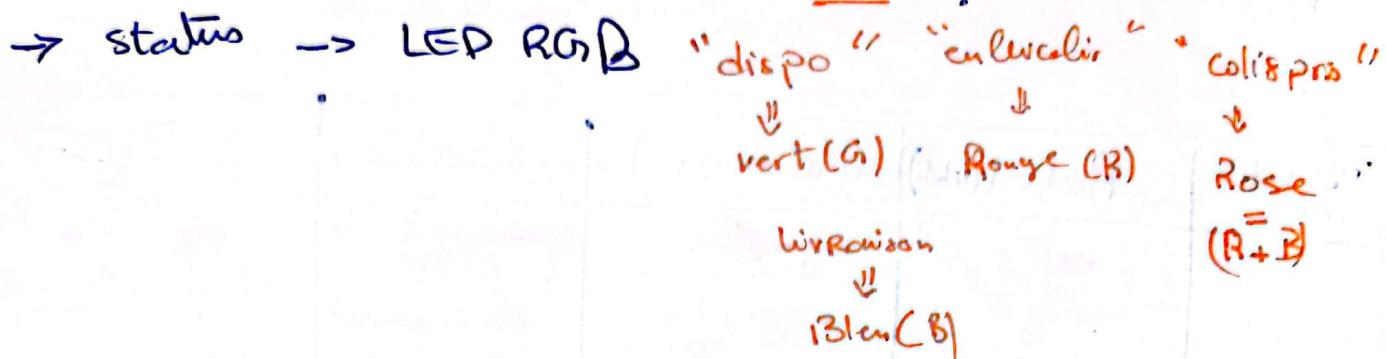
- Afficher via des leds le numero de poste reconnu.

Q: Comment avertir le robot du numero du poste en face duquel il est ?

Q: Comment assurer la synchronisation de la Transmssion ?

→ Affichage de la lettre de service pour que le poste de travail reconnaîsse le robot et status visible.

Cette affichage se fera par rotation d'un servo moteur qui permettra ce qui permet d'afficher une des quatre lettres via un texte "livreur A" "Livreur B" "Livreur C" "Livreur D" ou "—".



Codage incrémentaux, odometrie

Basys 2 muni d'un FPCIA \Rightarrow système odométrique

mesurer la distance parcourue

- Basys 2 ^{Power} = 3V3
- Communication Basys2 \leftrightarrow Arm7 \rightarrow Liaison SPI
- l'arm7 délivrera Ra, CLK, et les signaux de contrôle CS
- On mesure:
 - la vitesse des roues par mesure du temps entre deux temps codéurs.
 - Pour avoir une mesure stable : Mes = $\frac{MesA + MesB + MesC}{3}$

- A quelle Endroit il le plus simple de faire cette moyenne ?

→ . . . - - - - - . . . - - - - -



10

- Basys2 prendra l'armé qu'elle dispose d'une n° info valide 50 fois par sec 50 fois/sec

→ Asservissement

- les info de vitesse permettront de calculer en temps réel la vitesse moy du robot et donc d'agir sur les PWM commandant les roues pour que cette vitesse moy respecte la consigne souhaitée par la supervision.

- L'état des interrupteur sur la carte, les afficheurs de la carte affichent Vg, Vd, Pg, Pd,

- L'ARM7 interrogera la Basys2 à l'aide de 4CS différent pour lui indiquer quelle infos elle veut recuperer par la liaison SPI

- On met pas en œuvre le periph SPI.
On genere le chronogrammes nécessaires,
TOP_CIK(), LIRE_bit(), ...