

ROBOT recoit une info du poste : succession de touches DTMF

4 fils qui codent le numéro de touche  
1 fil qui dit nouvelle touche : son changement d'etat indique qu'une nouvelle touche est arrivée → IT externe ou detection front manuelle

si detection manuelle, lire périodiquement DV, détecter le front...

on a besoin d'une variable qui contient l'état précédent ; OLD\_DV

```
OLD_DV=DV;  
DV= lire_patteDV();  
//détecteur de front  
if ((DV==1) &(OLD_DV==0)) //on a un front montant  
{gere_reception_frame_dtmf();}  
  
//lire les 4 pattes pour récupérer un numéro de touche => touche_recue,  
on postule qu'on a une séquence * Dizaine Unite #  
if (touche_recue ==valeurduDise) {mae_dtmf =1;return;}  
switch(mae_dtmf)  
{case 1: //gere reception de la dizaine  
case 2: //gere reception de l'unité  
case 3: //verifier qu'on a un diese :  
calcul du numero de poste, et prise de decision en fonction de notre status et d'un besoin d'arret à un poste...  
ON DOIT PREVENIR D'AUTRES ENDROITS du CODE via des SEMAPHORES  
case 0: //poubelle : on sait pas où on en est (démarrage, caracteres en trop)  
}
```

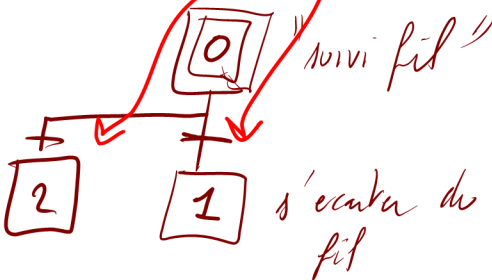
flag arret gauche  
flag arret droite

Comment déterminer la rotation du robot

Comment déterminer

Vitesse du Robot

- Consigne via le fil
- Sécurité Telem : Vmax
- Arrêt Urgence
- Marche / Arrêt



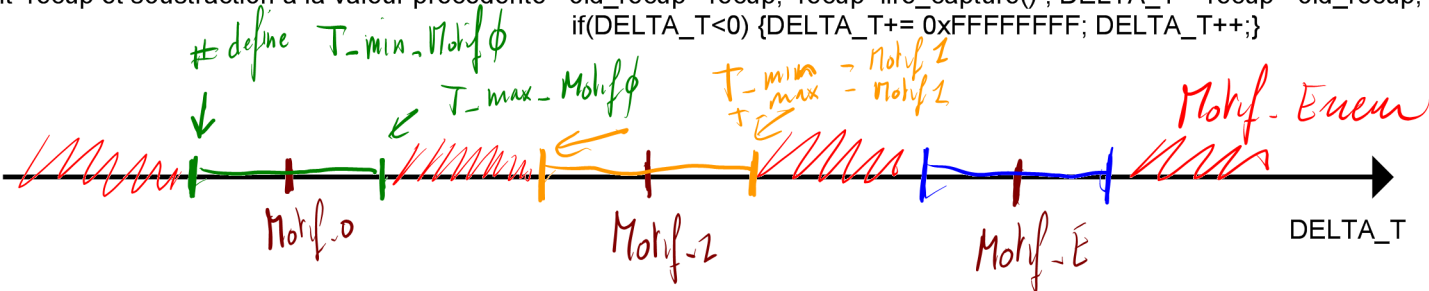
comment analyser la trame emise dans le fil



l'info est codée dans la durée à l'état haut, mais comme l'état bas dure toujours la même durée, .... Le motif est caractérisé par la durée entre deux fronts montants...  
OUTIL = CAPTURE

IT\_CAPTURE // dans laquelle on vient à chaque front montant

```
1 recuperer la duree...  
soit recup puis remise à zéro du timer  
soit recup et soustraction à la valeur précédente  
DELTA_T = lire_capture(); RAZ_timer();  
old_recup =recup; recup=lire_capture(); DELTA_T = recup - old_recup;  
if(DELTA_T<0) {DELTA_T+= 0xFFFFFFFF; DELTA_T++;}
```



```
MOTIF_recupere = trier_duree(DELTA_T); // '0', '1', 'E', 'M'
```

```
switch (MOTIF_recupere)
```

```
{
```

```
case 'E' : // on a récupéré un entete
```

```
// c'est la fin d'une trame , est elle OK ou pas ?
```

```
// normalement un message fait exactement une certaine taille
```

```
// si tout est bon, transferer la trame et prévenir le programme principal qu'on a un ordre valide
```

```
// c'est le début d'une nouvelle trame ...
```

```
//à priori, on est optimiste, ca va bien se passer, elle sera OK
```

```
// on va réécrire en début de tableau
```

```
break;
```

```
case '1' :
```

```
// si trame est OK jusqu'à présent....
```

```
Tableau_recept[index_recept] =1;
```

```
// si on reste dans le tableau index_recept++;
```

```
// sinon,..... trame plus OK
```

```
break;
```

```
case '0' :
```

```
// si trame est OK jusqu'à présent....
```

```
Tableau_recept[index_recept] =0;
```

```
// si on reste dans le tableau index_recept++;
```

```
// sinon,..... trame plus OK
```

```
break;
```

```
case 'M' : // on est perdu : la trame est massacrée et sera inexploitable jusqu'à la prochaine trame
```

```
trame_OK =0;
```

```
break;
```

```
}
```

dans la boucle while(1) du main

```
if(FLAGS_NEW_MESSAGE ) { analyse_message(); FLAGS_NEW_MESSAGE=0;}
```

dans analyse\_message : isoler num\_robot, isoler ordre, isoler le parametre

si pas pour nous, on fait rien

si pour nous :

```
switch(ordre)
```

```
{ case VITESSE:// le parametre est la vitesse souhaitée
```

```
break;
```

```
case GARAGE_GAUCHE_ENLEVEMENT:
```

```
// si j ai le statut libre, je vais mémoriser : FLAG_ARRET_GAUCHE et num_poste_arret
```

```
// je passe en statut MISSIONNE
```

```
case GARAGE_DROIT_ENLEVEMENT:
```

```
// si j ai le statut libre, je vais mémoriser : FLAG_ARRET_DROIT et num_poste_arret
```

```
// je passe en statut MISSIONNE
```

```
case GARAGE_DROIT_DEPOT:
```

```
// si j ai le statut CHARGÉ, je vais mémoriser : FLAG_ARRET_DROIT et num_poste_arret
```

```
// je passe en statut LIVRAISON
```

```
case GARAGE_GAUCHE_DEPOT:
```

```
// si j ai le statut CHARGÉ, je vais mémoriser : FLAG_ARRET_GAUCHE et num_poste_arret
```

```
// je passe en statut LIVRAISON
```

```
}
```

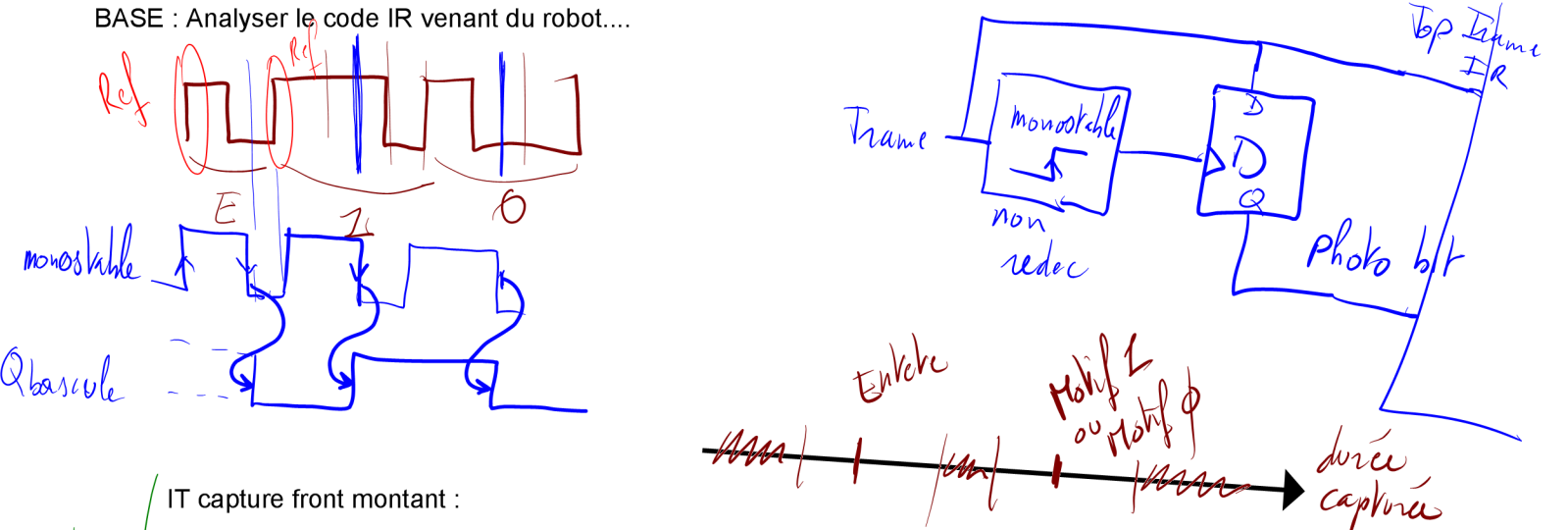
→ transferer message

alternative  
masque = 1 << ? ;  
(message = 0 ;)  
→ effacer le vieux message

→ message |= masque ;  
→ masque >> = 1 ;

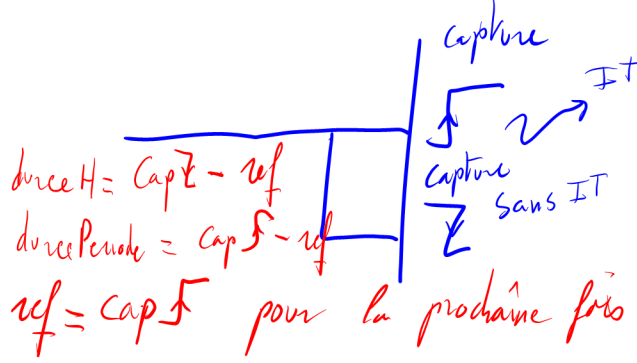
→ rien  
masque >> = 1 ;

BASE : Analyser le code IR venant du robot....



*hard* IT capture front montant :  
durée capturée m'informe si c est un Entete ou un motif 0 ou 1  
↳ en deux fronts montant  
la lecture de la patte photobit me dit si j'ai récupéré un 1 ou un 0

*soft* autre astuce, on fait une capture sur front descendant pour mesurer la durée entre le front montant et le front descendant  
↳ avec 2 entrées hard



*soft* : changer 5 en 7 pour la capture et gérer les cas avec if else

**un motif E déclenche une analyse de trame....**  
**et nous permet de savoir si on emettre une trame DTMF ou pas**

- si on est en attente de livreur et qu'on voit passer un missionné
- si on voit passer un robot en statut livraison , ca peut etre aussi pour nous....

**index\_trame\_dtmf = 0; //si on incrémente l'index des sons de la trame**  
**ou index\_trame\_dtmf = 11; // si on se déplace en décrémentant...**

# Analyse reception frame série :

```
unsigned char mae_mes_base=0;//je suis perdu

while(1)

{

/*****/
// dans le main pour gérer la liaison série en réception et en envoi :
//on lira les caractères recus dans U0RBR et on écrira dans U0THR
// les flag importants sont dans U0LSR

// le Bit 0 pour savoir si on a reçu un caractère : on l'utilise ainsi
status_rs232 = U0LSR;
if(status_rs232 & 1)
{// on a reçu un caractère
if(status_rs232 & 0x0E) {/*il y a une erreur overrun ou parité ou bit stop, on est PERDU */mae_mes_base=0;}
else {ch_recu=U0RBR; analyse_trame_BASE(ch_recu);}

if(flag_faut_que_je_reponde)
{//si j'ai mémorisé avoir vu un robot, j'appelle envoi_mes_vu_robot(dernier_robot_vu,statut,vitesse)
//si j'ai une demande clavier envoi_mes_demandeclavier(postealivrer, lettrelivreur)
// on finit toujours par un dernier message NULL mess_null
// s'inspirer du document conseils.pdf pour la gestion de fifo
}
if(status_rs232 & 1<<5) {depile_fifo_envoi();}
/*****}

void analyse_trame_BASE(unsigned char ch)
{
static unsigned char dizaine =0;
if(ch==@) {mae_mes_base =1; return;}
switch (mae_mes_base)
{case 1:
if(ch!='P') mae_mes_base =0;
else mae_mes_base =2;
break;
case 2:
if(ch<'0') {mae_mes_base =0; break;}
if(ch>'9') {mae_mes_base =0; break;}
dizaine = ch-'0'; mae_mes_base =3;
break;
case 3:
if(ch<'0') {mae_mes_base =0; break;}
if(ch>'9') {mae_mes_base =0; break;}
unite = ch-'0'; mae_mes_base =0;
if(mon_num_poste== (dizaine*10+unite)) {flag_faut_que_je_reponde=1;}
break;
case 0: break;
}
}
```