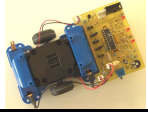




## Table des matières

1. Mise en situation .....	2
1.1 Expression du besoin .....	2
1.2 Historique .....	2
1.3 Principe du suivi de la ligne adopté par MrLineTiny .....	3
2 Analyse fonctionnelle .....	5
2.1 Fonction d'usage .....	5
2.2 Etude des milieux associés .....	5
2.3 Schéma fonctionnel de premier degré .....	6
2.4 Description des fonctions principales .....	7
2.5 Description des fonctions secondaires .....	9
2.5.1 FP2 « Emettre un faisceau lumineux infrarouge(IR) » .....	9
2.5.2 FP3 « Transformer un faisceau lumineux infrarouge en une information numérique » ..	10
2.5.3 Algorigramme « Acquisition de la position du robot par rapport à la ligne » .....	11
2.5.4 FP4 (FP5) « Régler la fréquence de rotation de la roue gauche (droite) » .....	12
2.5.5 FP6 « Visualiser la position du robot par rapport à la ligne » .....	13
3 Analyse structurelle .....	14
3.1 Etude structurelle de FP2 « Emettre un faisceau lumineux infrarouge » .....	14
3.1.1 FS22 « Emettre un faisceau lumineux infrarouge » .....	14
3.2 Etude structurelle de FP3 « Transformer un faisceau lumineux infrarouge en une .....	14
information numérique » .....	14
3.2.1 FS31 « Détecter un faisceau lumineux infrarouge » .....	14
3.2.2 FS32 « Sélectionner un des trois récepteurs d'infrarouges » .....	14
3.2.3 FS34 « Comparer une ddp à un seuil » (Conditionnement du signal AIN1) .....	15
3.2.4 FS33 « Régler un seuil de tension » (voir annexe 1) .....	15
3.3 Etude structurelle de FP4 et FP5 « Régler la fréquence de rotation des roues droite et	
gauche » .....	16
3.3.1 Etude de FS41 (FS51) « Régler un rapport cyclique » .....	16
3.3.2 FS42 « Amplifier une ddp » .....	17
3.3.3 FS43 « Convertir une énergie électrique en énergie mécanique » .....	17
3.3.4 FS44 « Transmettre l'énergie mécanique du moteur aux roues » .....	17
Annexe 1 : Description de la carte MrLineTiny .....	18
Annexe 3 : Fast partiel .....	19
Annexe 4 : Mesures sur FP2, signaux de commande des émetteurs infrarouges (E_IR) .....	20
Annexe 5 : Mesure sur FP3, signal (AIN1) délivré par les récepteurs infrarouges (R_IR) .....	21
Annexe 5 : Mesure sur FP3, zoom signal (AIN1) délivré par les récepteurs infrarouges (R_IR) ...	22
Annexe 6 : Mesures sur FP4(FP5), exemple de signaux MLI(PWM) de commande des moteurs .....	22
Annexe 7 : Mesures sur FP4(FP5), courant moteur $I_{mot}$ mesuré avec une résistance de $1\Omega$ placée en	
série avec le moteur .....	23
Annexe 8 : Mesures sur FP6, signal de commande d'une Del « Visu » .....	23
Annexe 9 : Schéma d'implantation de la carte « MrLineTiny » .....	24
Annexe 10 : Schéma structurel de la carte M <sup>f</sup> LineTiny .....	25
Annexe 11 : Liste des composants de la carte « MrLineTiny » .....	26
Annexe 12 : Programme C implanté dans « MrLineTiny » .....	27

Dossier Technique	TS SI		P.P.E. Robot suiveur de ligne	
 Lycée Polyvalent PIERRE EMILE MARTIN	Dossier technique			Dernière mise à jour le <b>11/04/2008</b>
Sites à consulter	Microrobot ( <a href="http://www.microrobotusa.com">http://www.microrobotusa.com</a> ) Lextronic ( <a href="http://www.lextronic.fr">http://www.lextronic.fr</a> )			

## 1. Mise en situation

### 1.1 Expression du besoin



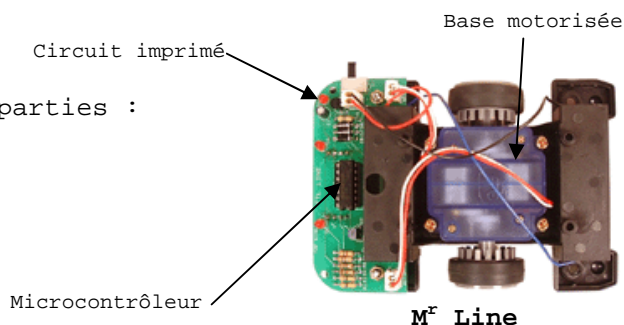
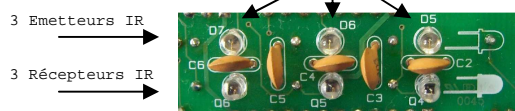
La participation de notre établissement à un challenge académique de robotique a nécessité la mise en œuvre, l'adaptation et l'amélioration d'un robot suiveur de ligne. Le choix du robot « **Mr Line** » de la société [Microrobot](http://www.microrobotusa.com) est principalement dû à son coût (<45€ en 2006).

### 1.2 Historique

Le robot « M<sup>r</sup> Line » se composait de deux parties :

- une **base motorisée** et,
- un **circuit imprimé**.

Les 3 capteurs IR sont placés sous le circuit imprimé. Chaque capteur se compose d'un émetteur et d'un récepteur.



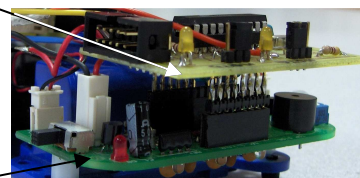
Le circuit imprimé servait de support aux trois capteurs à infrarouge chargés de détecter la ligne, au module à microcontrôleur AT89C2051 et aux deux transistors (interface entre le microcontrôleur et les motoréducteurs d'entraînement des roues).

Le microcontrôleur AT89C2051 n'étant pas supporté par notre outil logiciel (CodeVision AVR), il a été remplacé par un ATMEL ATINY26 monté sur une carte d'extension. Cette carte venait s'enficher à l'emplacement du module AT89C2051.



MrLineTiny

Extension ATINY26

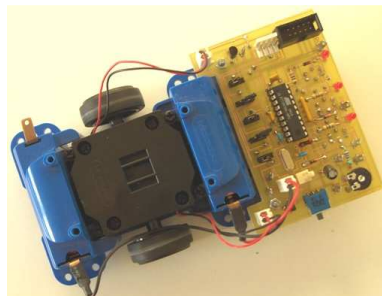
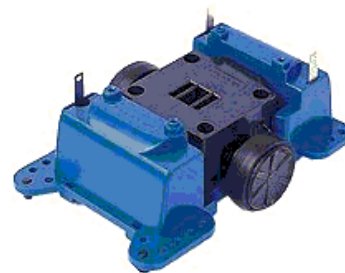


Circuit imprimé

Actuellement, le circuit imprimé d'origine et la carte d'extension sont remplacés par la carte **MrLineTiny**.

Si le robot suiveur de ligne « M<sup>r</sup> Line » a actuellement disparu des catalogues, la base motorisée est toujours distribuée (2008) par la société [Lextronic](#) sous la référence MRM GM03.

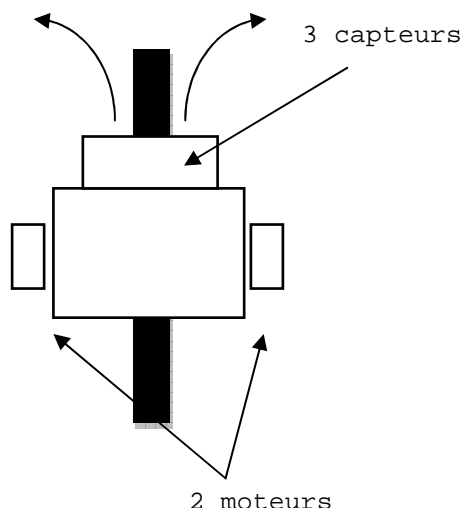
Les caractéristiques de cette base motorisée sont données aux paragraphes 3.3.3 et 3.3.4.



Le robot suiveur de ligne **MrLineTiny** que vous allez étudier est en photo ci-contre. La carte MrLineTiny est représentée en annexe 1.

### 1.3 Principe du suivi de la ligne adopté par MrLineTiny

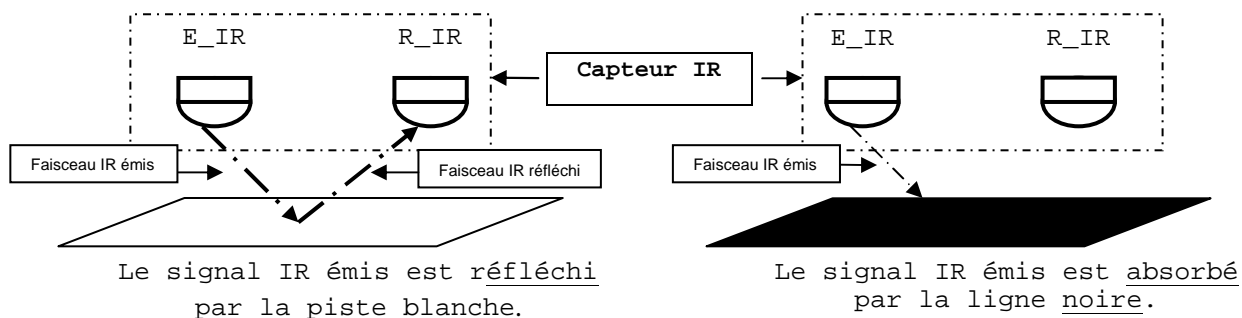
Le robot doit suivre une ligne noire de 20mm de large sur fond blanc. Deux moteurs assurent la propulsion et la direction du robot. Trois capteurs optiques situés à l'avant permettent la détection de la ligne.



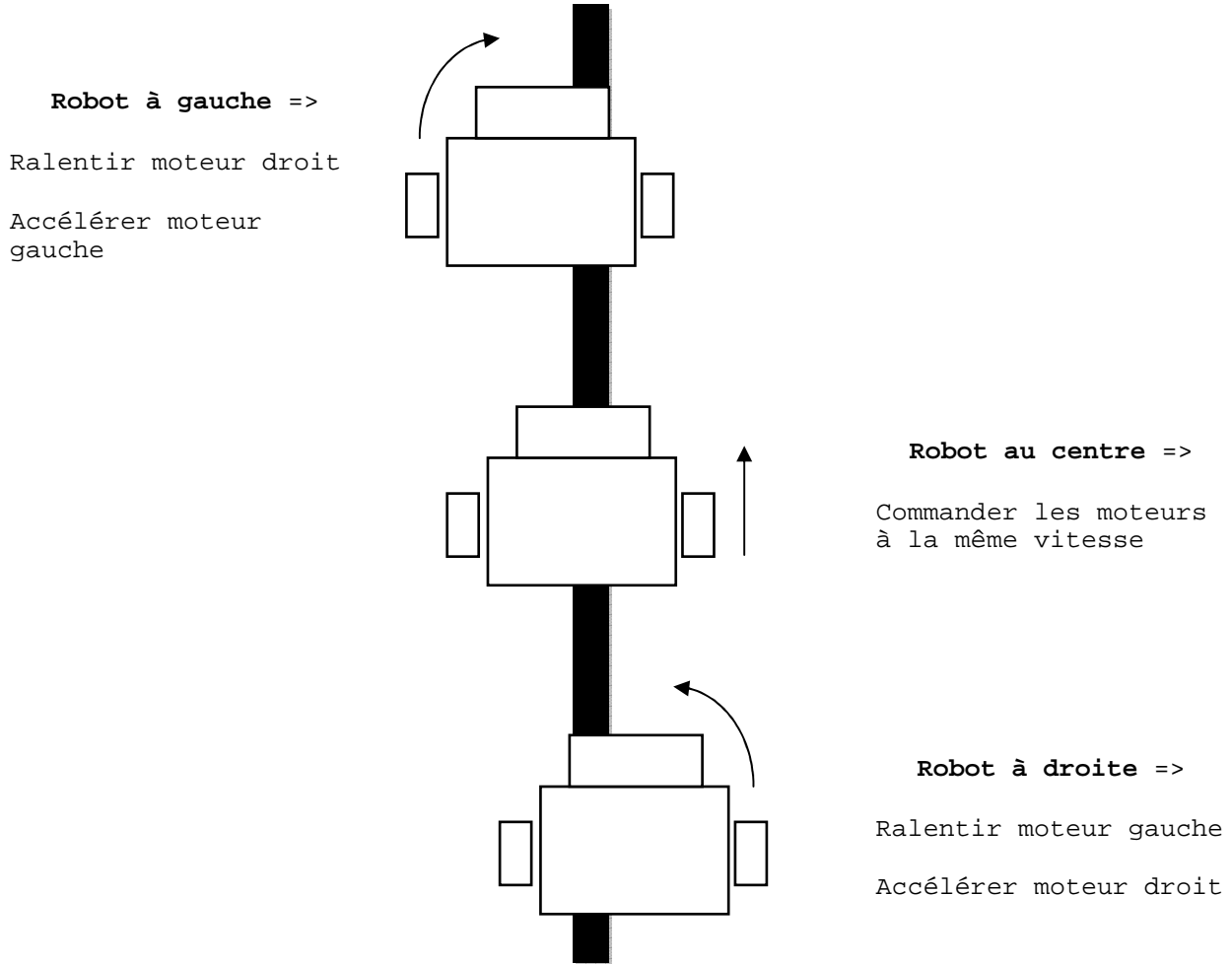
Le robot détecte sa position par rapport à la ligne noire tracée sur la piste grâce à **trois capteurs** infrarouge (IR). Ces capteurs sont identifiés par les sigles **IR\_Gauche (IR\_G)**, **IR\_Centre (IR\_C)** et **IR\_Droite (IR\_D)**. (Schéma annexe 10)

Chaque capteur se compose d'un émetteur **E\_IR** (diode IR) et d'un récepteur **R\_IR** (phototransistor IR).

Le principe de détection de la ligne noire est illustré ci-dessous :

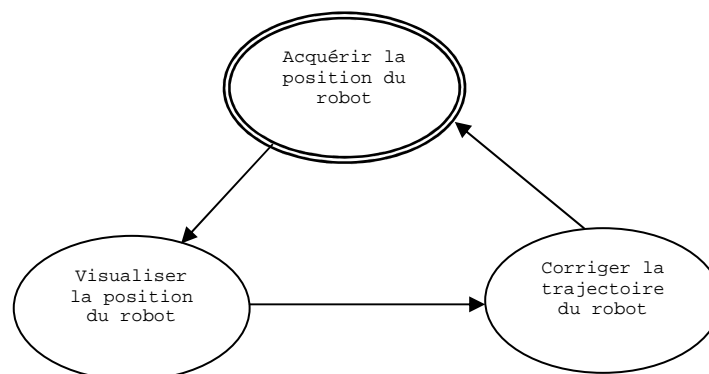


Les informations délivrées par les capteurs sont représentatives de la **position du robot** par rapport à la ligne. Elles sont **traitées** par un programme implanté dans un **microcontrôleur**. Grâce à une structure appelée « **Timer** », le microcontrôleur adapte la fréquence de rotation de chacun des **motoréducteurs** associés aux roues. Ceci à pour effet de **corriger la trajectoire** du robot afin de le maintenir sur la ligne noire.



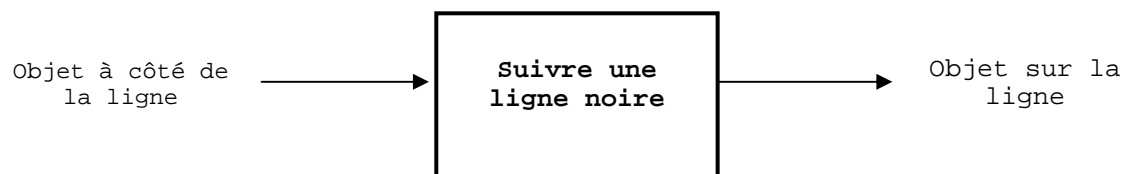
La position du robot par rapport à la ligne est représentée par l'état (allumé ou éteint) de trois **diodes électroluminescentes** (Dels).

Le programme implanté dans le microcontrôleur parcourt le **cycle** ci-dessous.



## 2 Analyse fonctionnelle

### 2.1 Fonction d'usage



### 2.2 Etude des milieux associés

- Physique

Le robot est utilisé en intérieur sur une surface plane et blanche. Il n'est pas obligatoirement à l'abri du soleil.

- Technique

La piste est tracée avec une bande adhésive noire (ou peinte).

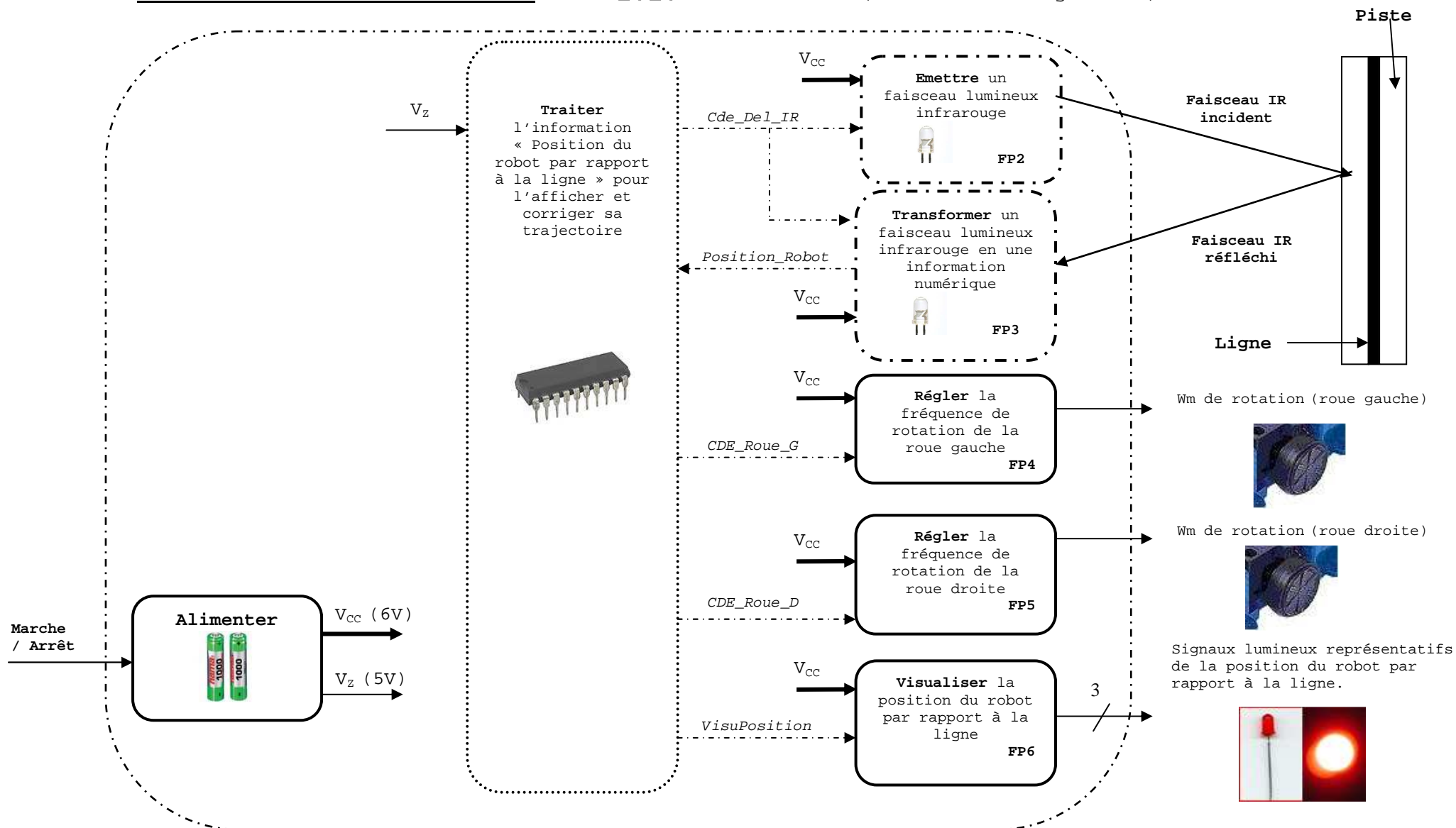
- Économique

Le prix des composants nécessaires à la réalisation du robot ne dépasse pas 50 euros (en 2007).

## 2.3

Schéma fonctionnel de premier degré

— Fonction matérielle      ..... Fonction logicielle  
 - - - - - Fonction mixte (matérielle et logicielle)



-----> : Information (logicielle)  
 —> : Energie

## 2.4 Description des fonctions principales

### FPl : Fonction « Traiter... »

**Rôles:** Commander les émetteurs infrarouges, régler la fréquence de rotation de chacune des roues du robot pour qu'il suive la ligne noire tracée sur la piste et informer l'utilisateur sur la position du robot par rapport à la ligne.

#### Entrée

- **Position\_Robot:** Variable numérique de type octet. Image de la position du robot par rapport à la ligne. Seuls les trois bits de poids faible sont significatifs.

Position_Robot	Position du robot par rapport à la ligne
0	Hors ligne
1	Droite+ : « très à droite de la ligne »
2	Centre1 : ligne noire de largeur égale à celle de la zone de détection d'un capteur
3	Droite : « à droite de la ligne »
4	Gauche+ : « très à gauche de la ligne »
5	Centre2 : ligne blanche entourée de deux noires
6	Gauche : « à gauche de la ligne »
7	Centre3 : ligne noire de largeur égale à celle de trois capteurs

#### Sorties

- **Cde\_Del\_IR :** variable numérique de type octet permettant de sélectionner un émetteur infrarouge(IR) parmi les trois. Seuls les trois bits de poids faible sont significatifs.

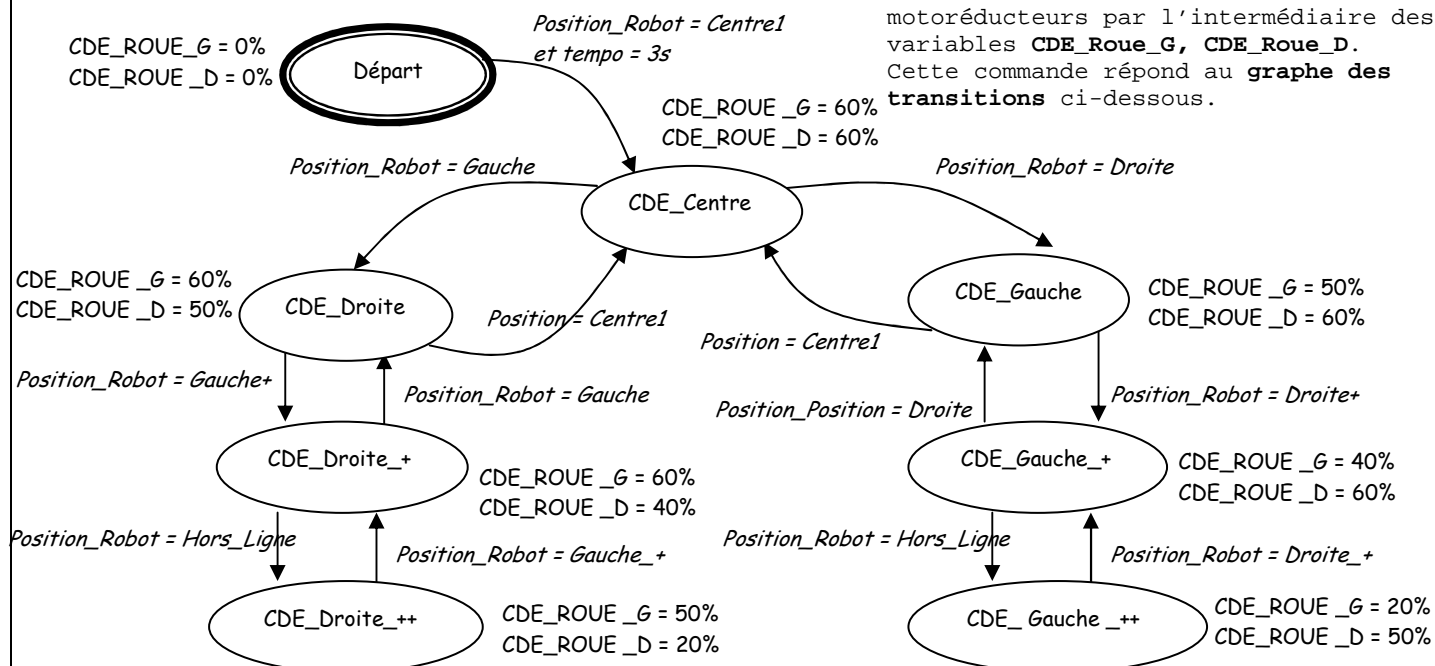
Cde_Del_IR	Emetteur IR commandé
FF <sub>(16)</sub>	Aucun
FB <sub>(16)</sub>	Emetteur IR droit
FD <sub>(16)</sub>	Emetteur IR centre
FE <sub>(16)</sub>	Emetteur IR gauche

- **VisuPosition:** variable numérique de type octet permettant de sélectionner les dels visualisant la position du robot par rapport à la ligne.

VisuPosition	Etat des dels « Visu »	Position du robot
3F <sub>(16)</sub>	○ ○ ○	Hors ligne
3E <sub>(16)</sub>	● ○ ○	Droite+
1F <sub>(16)</sub>	○ ● ○	Centre1
1E <sub>(16)</sub>	● ● ○	Droite
2F <sub>(16)</sub>	○ ○ ●	Gauche+
2E <sub>(16)</sub>	● ○ ●	Centre2
0F <sub>(16)</sub>	○ ● ●	Gauche
0E <sub>(16)</sub>	● ● ●	Centre3

- **CDE\_Roue\_G, CDE\_Roue\_D:** Variables de type octet représentatives du rapport cyclique du signal de commande des motoréducteurs.

La trajectoire du robot est corrigée en modifiant la commande de chacun de ses motoréducteurs par l'intermédiaire des variables **CDE\_Roue\_G, CDE\_Roue\_D**. Cette commande répond au **graphe des transitions** ci-dessous.



*Remarque : Les pourcentages de puissance de CDE\_ROUE \_G et CDE\_ROUE \_D sont données à titre indicatif. Ils permettent d'assurer le guidage approximatif du robot sur la ligne !*

**FP2 : Fonction « Emettre un faisceau lumineux infrarouge(IR) »**

**Rôle:** Commander un des trois émetteurs infrarouges.

**Entrée**

- Cde\_Del\_IR : voir FP1.

**Sortie**

- Faisceau IR incident : faisceau lumineux infrarouge émis vers la piste.

**FP3 : Fonction « Transformer un faisceau lumineux infrarouge en une information numérique »**

**Rôle:** Capter le signal infrarouge réfléchi par la piste et le transformer en une information numérique représentative du robot par rapport à la piste.

**Entrées**

- Faisceau IR réfléchi : faisceau lumineux infrarouge réfléchi par la piste.  
- Cde\_Del\_IR : voir FP1.

**Sortie**

- Position\_Robot : voir FP1.

**FP4 : Fonction « Régler la fréquence de rotation de la roue gauche »**

**Rôle:** Régler le rapport cyclique du signal de commande du moteur d'entraînement de la roue gauche.

**Entrée**

- CDE\_Roue\_G : Variable de type octet représentative du rapport cyclique du signal de commande du moteur d'entraînement de la roue gauche.

**Sortie**

- Wm : Energie mécanique de rotation appliquée à la roue gauche.

**FP5 : Fonction « Régler la fréquence de rotation de la roue droite »**

**Rôle:** Régler le rapport cyclique du signal de commande du moteur d'entraînement de la roue droite.

**Entrée**

- CDE\_Roue\_D : Variable de type octet représentative du rapport cyclique du signal de commande du moteur d'entraînement de la roue droite.

**Sortie**

- Wm : Energie mécanique de rotation appliquée à la roue droite.

**FP6 : Fonction « Visualiser la position du robot par rapport à la ligne »**

**Rôle:** Informer l'utilisateur sur la position du robot par rapport à la ligne. La del s'éclaire lorsque le capteur IR associé est sur la ligne.

**Entrée**

- VisuPosition : voir FP1

**Sorties**

Faisceaux lumineux visibles.



## **FA: Fonction « Alimentation »**

**Rôle:** Assurer l'alimentation en énergie de l'ensemble des fonctions.

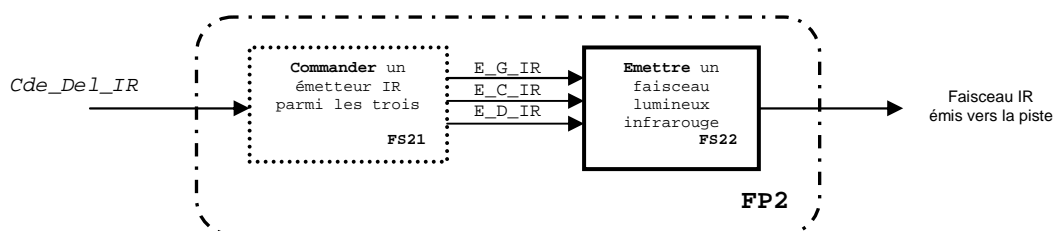
- $V_{cc}$  : DDP obtenue à partir de 4 piles type AA(ou accumulateurs) de 1,5V.  $V_{cc} \approx 6V$ .
- $V_z$  : DDP stabilisée.  $V_z \approx 5V$ .

### **Rappel des conventions**

- Fonction matérielle                      ..... Fonction logicielle  
-.-.- Fonction mixte (matérielle et logicielle)

## **2.5 Description des fonctions secondaires**

### **2.5.1 FP2 « Emettre un faisceau lumineux infrarouge(IR) »**



#### **FS21: Fonction secondaire « Commander une des trois Del IR »**

**Rôle:** Sélectionner l'émetteur devant générer un faisceau lumineux infrarouge.

##### **Entrée**

- **Cde\_Del\_IR** : variable numérique de type octet permettant de sélectionner un émetteur IR parmi trois. Seuls les trois bits de poids faible sont significatifs.

	b7	b6	b5	b4	b3	b2	b1	b0
Cde_Del_IR	1	1	1	1	1	<b>E_D_IR</b>	<b>E_C_IR</b>	<b>E_G_IR</b>

##### **Sorties**

- **E\_G\_IR, E\_C\_IR, E\_D\_IR**: Signaux de commande des émetteurs IR (TOR actifs à l'état bas).

Cde_Del_IR <sub>(16)</sub>	<b>E_D_IR</b>	<b>E_C_IR</b>	<b>E_G_IR</b>	Emetteur commandé
<b>FF</b> <sub>(16)</sub>	<b>1</b>	<b>1</b>	<b>1</b>	Aucun
<b>FB</b> <sub>(16)</sub>	<b>0</b>	<b>1</b>	<b>1</b>	Emetteur IR droit
<b>FD</b> <sub>(16)</sub>	<b>1</b>	<b>0</b>	<b>1</b>	Emetteur IR centre
<b>FE</b> <sub>(16)</sub>	<b>1</b>	<b>1</b>	<b>0</b>	Emetteur IR gauche

Remarque : l'algorithme associé à FS21 est donné au §2.5.3

#### **FS22: Fonction secondaire « Emettre un faisceau lumineux infrarouge»**

**Rôle:** Convertir une énergie électrique en énergie lumineuse.

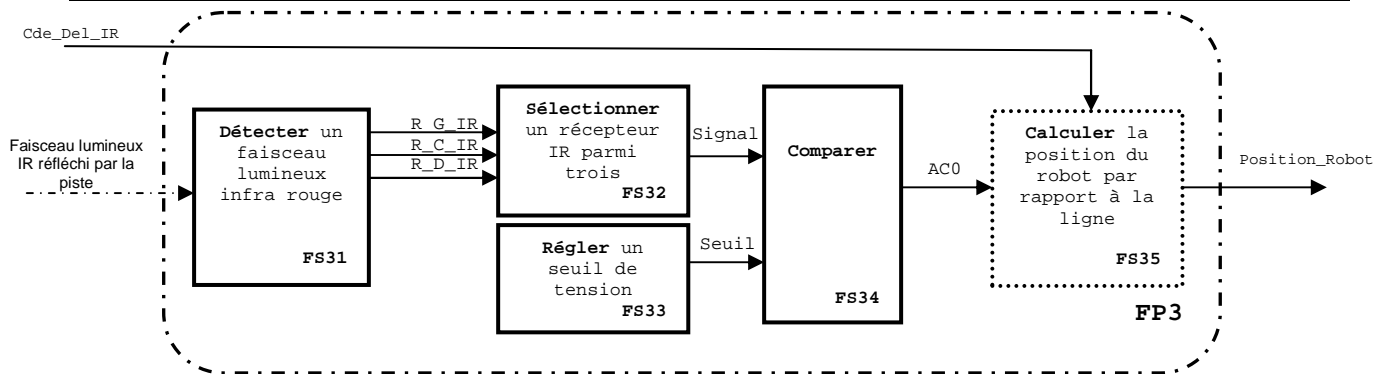
##### **Entrées**

- **E\_G\_IR, E\_C\_IR, E\_D\_IR** : voir FS21.

##### **Sorties**

- **Faisceau IR** : énergie lumineuse.

## 2.5.2 FP3 « Transformer un faisceau lumineux infrarouge en une information numérique »



### **FS31: Fonction « Détecter un faisceau lumineux infra rouge »**

**Rôle:** Transformer un faisceau lumineux infrarouge en une différence de potentiel (d.d.p.).

#### **Entrée**

- **Faisceau IR réfléchi** : faisceau lumineux infrarouge réfléchi par la piste.

#### **Sorties**

- **R\_G\_IR, R\_C\_IR, R\_D\_IR** : D.D.P. analogiques dont l'amplitude est représentative de la position d'un capteur par rapport à la ligne noire.

### **FS32: Fonction « Sélectionner un récepteur IR parmi les trois »**

**Rôle:** Sélectionner le récepteur IR correspondant à l'émetteur commandé.

#### **Entrées**

- **R\_G\_IR, R\_C\_IR, R\_D\_IR** : voir FS31.

#### **Sortie**

- **Signal** : Signal = R\_G\_IR ou R\_C\_IR ou R\_D\_IR.

### **FS33: Fonction « Régler un seuil de tension »**

**Rôle:** Régler la valeur d'une d.d.p continue permettant de différencier la piste blanche de la ligne noire.

#### **Sortie**

- **Seuil** : DDP continue réglée entre 0 et 6V.

### **FS34: Fonction « Comparer une d.d.p. à un seuil »**

**Rôle:** Transformer la d.d.p représentative de la « position du capteur par rapport à la ligne » en un signal TOR.

#### **Entrées**

- **Seuil, Signal** : voir FS32 et FS33.

#### **Sortie**

- **AC0**: Signal tout ou rien (TOR) représentatif de la position d'un des trois capteurs infrarouge par rapport à la ligne (signal interne au microcontrôleur).  
AC0 = 1 si le capteur sélectionné est sur la ligne.

### **FS35: Fonction « Calculer la position du robot par rapport à la ligne »**

**Rôle:** Etablir une valeur numérique représentative de la position du robot par rapport à la ligne.

#### **Entrée**

- **AC0**: voir FS34

#### **Sortie**

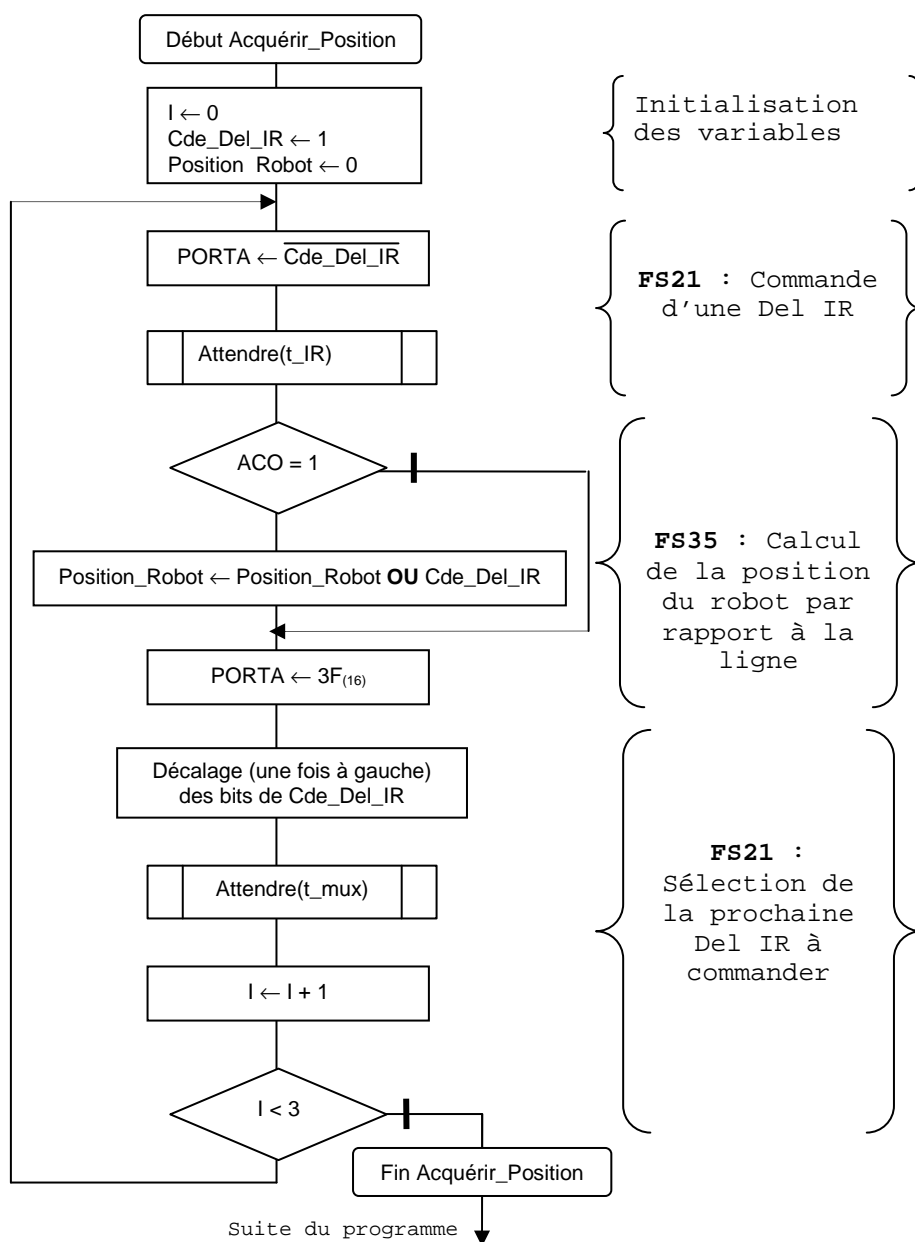
- **Position\_Robot**: Variable numérique de type octet, image de la position du robot par rapport à la ligne. Seuls les trois bits de poids faible sont significatifs.

**Remarque** : l'algorithme associé à FS34 est donné au §2.5.3

### 2.5.3 Algorithme « Acquisition de la position du robot par rapport à la ligne »

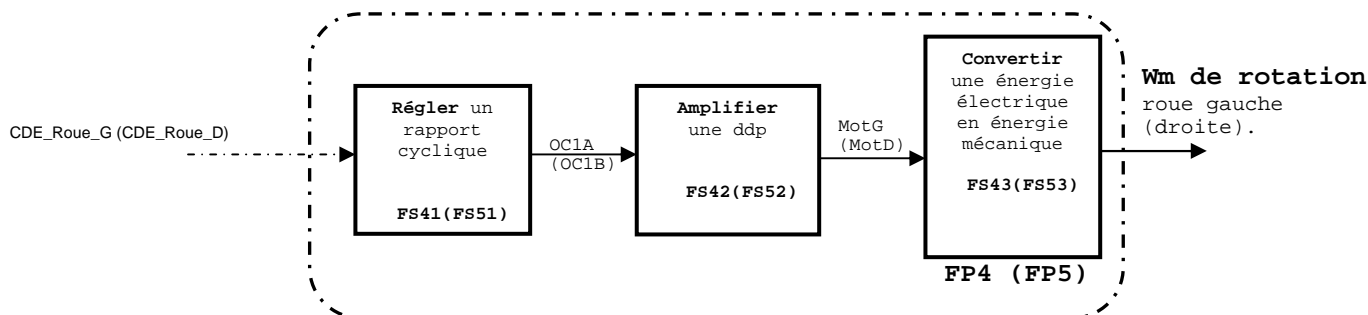
Cet algorithme assure le cycle de mesure de la position du robot par rapport à la ligne. Il est composé des fonctions secondaires FS21 « Commander une des trois Del IR » et FS35 « Calculer la position du robot par rapport à la ligne ».

Nom	Format	Commentaires
<b>Variables</b>		
I	octet	Sert de compteur dans la boucle
Cde_Del_IR	octet	Assure la commande multiplexée des émetteurs IR.
ACO	bit	Représentatif de la position d'un capteur IR par rapport à la ligne.
Position_Robot	octet	Valeur numérique représentative de la position du robot par rapport à la ligne après un cycle de mesure.
<b>Constantes</b>		
t_IR	octet	Durée d'activation d'un émetteur IR <sub>i</sub> i ∈ {1,2,3}
t_mux	octet	Durée avant de commander l'émetteur IR <sub>(i+1)</sub>



**Note :** Lors d'une opération de décalage à gauche, tous les bit sont déplacés d'un rang puis le bit de poids faible est positionné à 0.

#### 2.5.4 **FP4 (FP5) « Régler la fréquence de rotation de la roue gauche (droite) »**



##### **FS41: Fonction « Régler un rapport cyclique »**

**Rôle:** Modifier la valeur moyenne du signal de commande du moteur droit (gauche) en modifiant son rapport cyclique.

##### **Entrée**

- **CDE\_Roue\_D (CDE\_Roue\_G):** Variable de type octet représentative du rapport cyclique du signal de commande du moteur d'entraînement de la roue droite (gauche).

##### **Sortie**

- **OC1A (OC1B):** signal TOR de fréquence fixe et de rapport cyclique variable.

##### **FS42: Fonction « Amplifier une d.d.p. »**

**Rôle:** Changer la puissance du signal OC1A (OC1B) pour le rendre compatible avec la structure associée à FS43 (FS53).

##### **Entrée**

- **OC1A (OC1B):** voir FS41 (FS51).

##### **Sorties**

- **MotG (MotD):** OC1A (OC1B) amplifié en puissance.

##### **FS43: Fonction « Convertir une énergie électrique en énergie mécanique »**

**Rôle:** Entraîner les roues du robot en rotation.

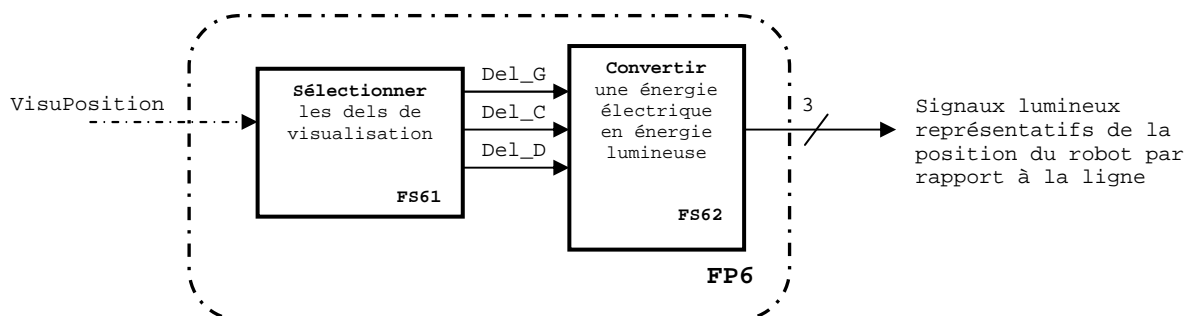
##### **Entrée**

- **MotG (MotD):** voir FS42 (FS52)

##### **Sortie**

- **Wm :** Energie mécanique de rotation appliquée à la roue gauche (droite).

### 2.5.5 FP6 «Visualiser la position du robot par rapport à la ligne»



#### **FS61: Fonction « Sélectionner les dels de visualisation »**

**Rôle:** Assurer l'éclairage des dels associées aux capteurs IR lorsque ceux-ci sont sur la ligne noire.

##### **Entrée**

- **VisuPosition** : variable numérique de type octet permettant de sélectionner les dels visualisant la position du robot par rapport à la ligne.

	b7	b6	b5	b4	b3	b2	b1	b0
VisuPosition	0	0	Del_C	Del_D	Del_G	1	1	1

##### **Sorties**

- **Del\_G, Del\_C, Del\_D** : signaux TOR actif à l'état bas.

VisuPosition	Signaux logiques de commande des dels de visualisation			Position du robot
	DeL_C	DeL_D	DeL_G	
3F <sub>(16)</sub>	1	1	1	Hors ligne
3E <sub>(16)</sub>	1	1	0	Droite+
1F <sub>(16)</sub>	0	1	1	Centre1
1E <sub>(16)</sub>	0	1	0	Droite
2F <sub>(16)</sub>	1	0	1	Gauche+
2E <sub>(16)</sub>	1	0	0	Centre2
0F <sub>(16)</sub>	0	0	1	Gauche
0E <sub>(16)</sub>	0	0	0	Centre3

#### **FS62: Fonction « Convertir une énergie électrique en énergie lumineuse »**

**Rôle:** Rendre visible la position du robot par rapport à la ligne.

##### **Entrées**

- **Del\_G, Del\_C, Del\_D** : voir FS61.

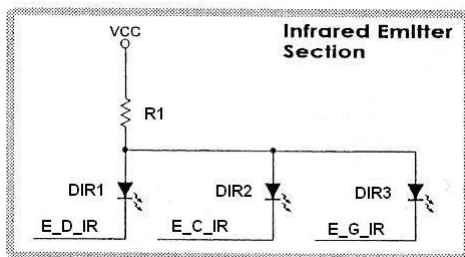
##### **Sorties**

- **Signaux lumineux visibles.**

### 3 Analyse structurelle

#### 3.1 Etude structurelle de FP2 « Emettre un faisceau lumineux infrarouge »

##### 3.1.1 FS22 « Emettre un faisceau lumineux infrarouge »



Les E\_IR DIR1, DIR2 et DIR3 émettent un faisceau IR lorsqu'ils sont commandés au niveau logique zéro (schéma ci-contre).

Exemple de fonctionnement : Si  $E\_G\_IR = 0V$  (« 0 » logique), un courant circule dans DIR3 à travers R1 et un faisceau lumineux IR est émis en direction de la piste.

Les chronogrammes des signaux de commande de DIR1, DIR2 et DIR3 sont donnés en **annexe 4 « Signaux de commande des émetteurs infrarouges »**.

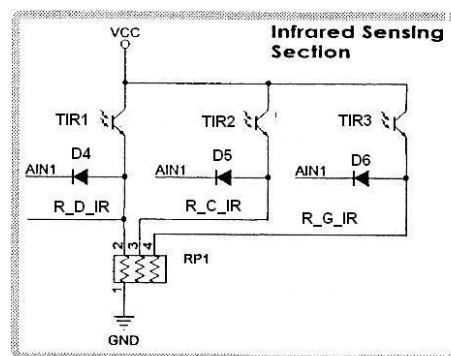
#### 3.2 Etude structurelle de FP3 « Transformer un faisceau lumineux infrarouge en une information numérique »

##### 3.2.1 FS31 « Détecter un faisceau lumineux infrarouge »

Les phototransistors IR (TIR1, TIR2, TIR3) associés à chaque Del IR captent les faisceaux réfléchis par la piste blanche. On peut assimiler un phototransistor à un interrupteur (« fermé » en présence d'un faisceau IR, « ouvert » en l'absence de faisceau IR).

Exemple de fonctionnement : Si  $E\_G\_IR = 0V$  et si le capteur IR\_G (DIR3, TIR3) est sur la piste blanche, le faisceau réfléchi atteint le phototransistor TIR3.

Celui-ci se « ferme », la diode D6 devient passante et  $AIN1 \approx R\_G\_IR \approx Vcc$ .



##### 3.2.2 FS32 « Sélectionner un des trois récepteurs d'infrarouges »

Les diodes D4, D5 et D6 réalisent un opérateur logique OU dit : « câblé ». Le signal AIN1 correspond à la sortie de cet opérateur. Il est représentatif de la position d'un capteur IR par rapport à la ligne noire. L'amplitude du signal AIN1 dépend de la position des capteurs IR par rapport à la ligne.

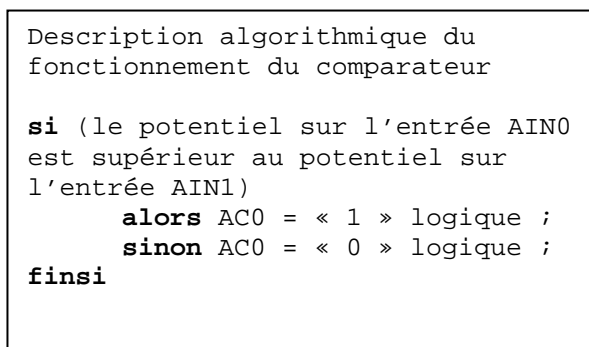
Un exemple de mesure de ce signal est donné en **annexe 5 « Signal AIN1 délivré par les récepteurs d'infrarouges »**.

Comme vous pouvez le constater sur les chronogrammes, AIN1 est un signal analogique.

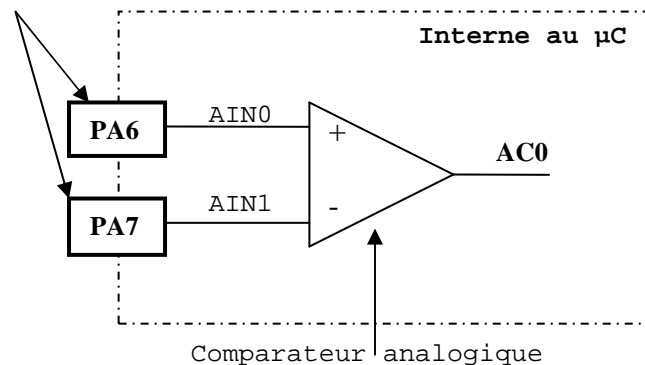
Pour que le microcontrôleur ( $\mu C$ ) traite ce signal, il est nécessaire de le « conditionner » (transformer) en un signal logique. Ceci est réalisé par une structure interne au  $\mu C$  appelée « comparateur analogique ».

### 3.2.3 FS34 « Comparer une ddp à un seuil » (Conditionnement du signal AIN1)

La ddp analogique  $V_{\text{signal}}$  placée à l'entrée AIN1 du microcontrôleur est convertie en un signal logique **AC0** par un **comparateur analogique** intégré au microcontrôleur. Le comportement de cette structure est décrit ci-dessous.



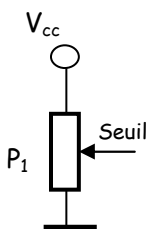
Broches du  $\mu\text{C}$



**AC0 est l'image de la position d'un capteur IR par rapport à la ligne.** AC0 est utilisé par la fonction logicielle FS35 « Calculer la position du robot par rapport à la ligne ». Cette fonction convertit la position du robot par rapport à la ligne en une valeur numérique.

### 3.2.4 FS33 « Régler un seuil de tension » (voir annexe 1)

Le potentiel  $V_{\text{seuil}}$  de la broche AIN0 est ajusté entre 0 et  $V_{\text{cc}}$  par une **résistance variable (P1)**. On peut ainsi régler le niveau du noir de la ligne et tenir compte de son pouvoir d'absorption.



### 3.3 Etude structurelle de FP4 et FP5 « Régler la fréquence de rotation des roues droite et gauche »

#### 3.3.1 Etude de FS41 (FS51) « Régler un rapport cyclique »

##### - Principe de réglage de la fréquence de rotation des moteurs

Chacune des roues du robot suiveur de ligne est équipées d'un moteur à courant continu. La vitesse de chacun des moteurs peut être réglée indépendamment de l'autre.

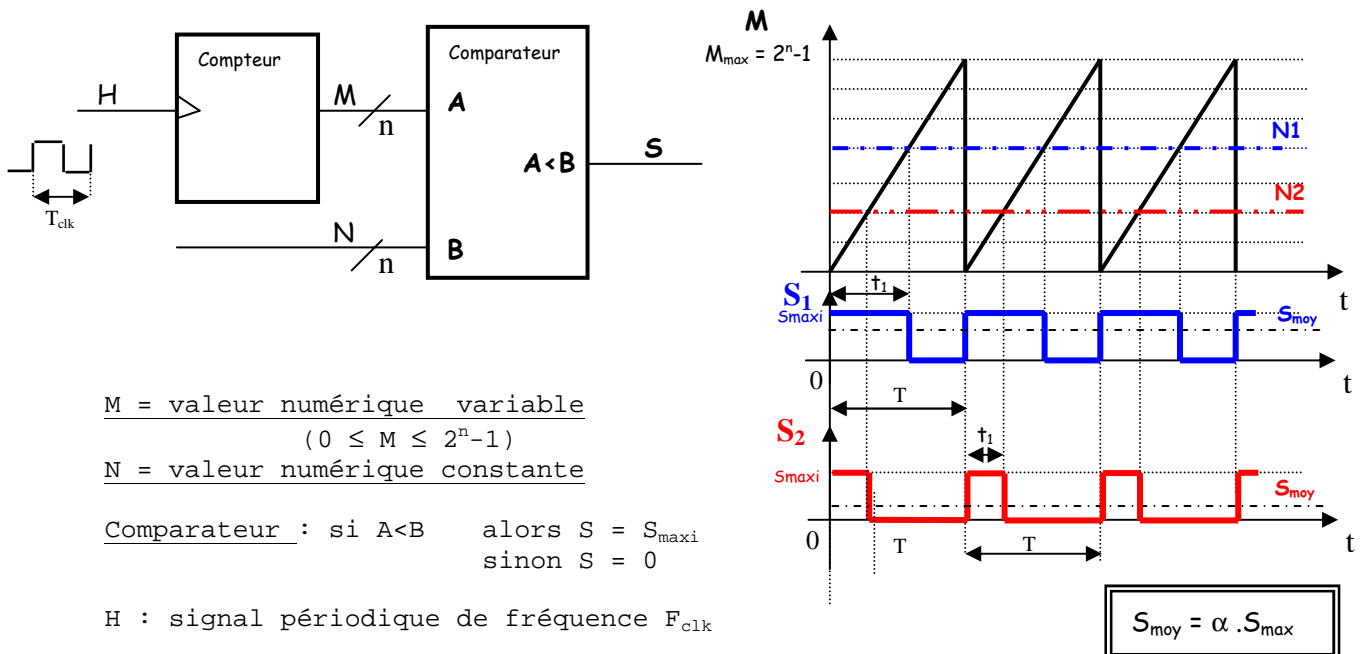


Pour faire varier la fréquence de rotation d'un moteur à courant continu, il suffit de modifier la valeur moyenne de la d.d.p. présente à ses bornes en le commandant par un signal dit « **MLI (ou PWM)** ».

##### Génération d'un signal Modulé en Largeur d'Impulsion (principe)

Un signal modulé en largeur d'impulsion peut être obtenu à partir d'un signal périodique  $H$  de fréquence fixe  $F_{clk} = 1/T_{clk}$ . En effet, en appliquant ce signal à l'entrée d'un compteur, on obtient un signal numérique  $M$  (codés sur  $n$  bits) capable d'évoluer de 0 à  $2^n - 1$ . La représentation de  $M(t)$  est appelée **rampe numérique**. En appliquant  $M(t)$  et un signal constant  $N(t)$  (codé sur  $n$  bits) à un comparateur numérique, on obtient un signal binaire  $S(t)$  de période  $T = (2^n - 1) \cdot T_{clk}$  dont le temps  $t_1$  à l'état « 1 » est réglé avec la valeur de  $N$ . On appelle  $\alpha = t_1/T$  le rapport cyclique du signal  $S(t)$ . On montre que la valeur moyenne  $S_{moy}$  de  $S(t)$  est égale au produit de  $\alpha$  par  $S_{maxi}$ .

On donne ci-dessous le schéma de principe d'une structure générant un signal M.L.I et les chronogrammes de  $S(t)$  pour deux valeurs particulières de  $N$ .



Dans les microcontrôleurs, les signaux modulés en largeur d'impulsion sont générés par une structure appelée **TIMER**. Celle-ci répond au principe développé ci-dessus.

\*M.L.I : Modulation de largeur d'impulsion (P.W.M. : Pulse With Modulation)

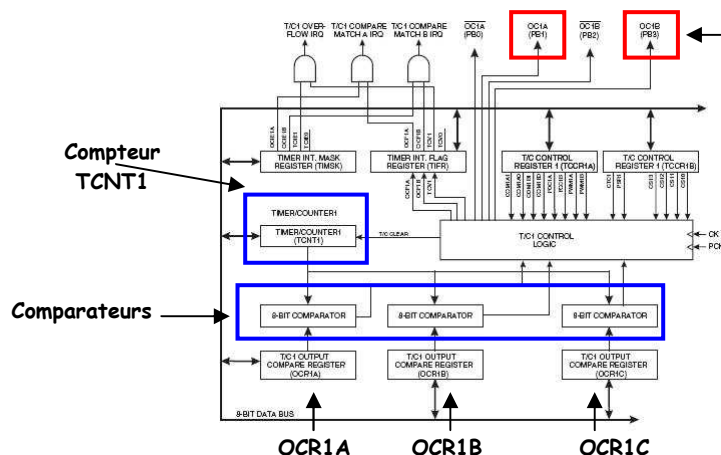


### - Génération d'un signal MLI avec le microcontrôleur ATINY26

Le **Timer1** de l'ATINY26 permet de générer quatre signaux tout ou rien modulés en largeur d'impulsion (OC1A, /OC1A, OC1B, /OC1B).

(/ = complément)

Il intègre un **compteur**, trois **comparateurs** et divers registres. En mode M.L.I. son fonctionnement répond au principe exposé dans le paragraphe précédent.

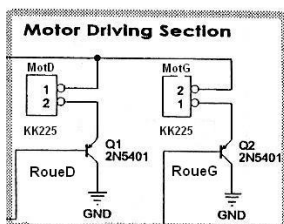


Résultats des comparaisons

Le compteur **TCNT1** génère le signal numérique **M**. Les registres **OCR1A** et **OCR1B** correspondent à **N**. **OCR1C** permet de régler  $M_{max}$  entre 0 et  $2^n - 1$  ( $n=8$ ). **OC1A** et **OC1B** correspondent à **S**. Le Timer 1 contient donc **deux structures** dont le fonctionnement répond au principe exposé dans le paragraphe précédent.

### 3.3.2 FS42 « Amplifier une ddp »

Les signaux de commande des moteurs des robots sont issus des broches PB1 (OC1A) et PB3 (OC1B) du microcontrôleur. Ces signaux commandent les transistors (Q1, Q2).



Les moteurs sont reliés au circuit des transistors Q1 et Q2 par l'intermédiaire des connecteurs MotD et MotG.

Remarque : Les transistors **inversent** les signaux de commandes OC1A(MotD) et OC1B(MotG).

PDIP/SOIC	
(MOSI/DI/SDA/OC1A) PB0	1
(MISO/DO/OC1A) PB1	2
(SCK/SC/OC1B) PB2	3
(OC1B) PB3	4
VCC	5
GND	6
(ADC7/XTAL1) PB4	7
(ADC8/XTAL2) PB5	8
(ADC9/INT0/T0) PB6	9
(ADC10/RESET) PB7	10
PA0 (ADC0)	11
PA1 (ADC1)	12
PA2 (ADC2)	13
PA3 (AREF)	14
GND	15
AVCC	16
PA4 (ADC3)	17
PA5 (ADC4)	18
PA6 (ADC5/AIN0)	19
PA7 (ADC6/AIN1)	20

Des exemples de mesures des signaux de commande des moteurs sont donnés **annexe 6** et **annexe 7**.

**La modification du rapport cyclique  $\alpha$  des signaux de commande des moteurs du robot se fait en modifiant les valeurs contenues dans les registres OCR1A et OCR1B.**

### 3.3.3 FS43 « Convertir une énergie électrique en énergie mécanique »

#### Caractéristiques du moteurs

- $U_{max}$ : 5 à 6V
- $P = 0,46W$
- $n_{vide} = 20100tr/mn$  ( $I_0 = 70mA$ )
- $C : 3g/cm$  en charge ( $I=150mA$ ,  $n=15000tr/mn$ )

### 3.3.4 FS44 « Transmettre l'énergie mécanique du moteur aux roues »

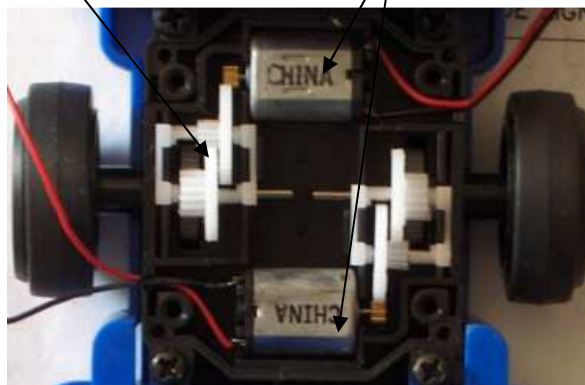
#### Caractéristique du réducteur

- $C_{max} = 11gf/cm$
- $r = 42.3/1$
- $C = 126g/cm$
- $v = 354tr/mn$

Roues :  $\varnothing 30mm$

Réducteur

Moteurs à C.C.

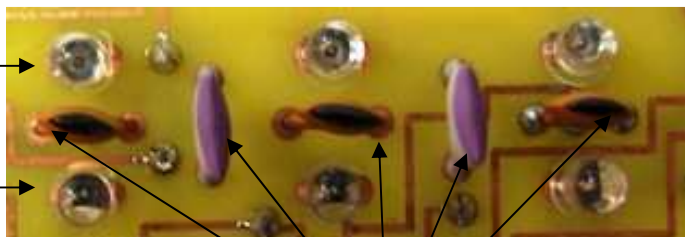


Vu de dessous

Récepteurs IR (R\_IR)

Emetteurs IR (E\_IR)

1 Emetteur IR + 1 Récepteur IR = 1 capteur IR  
IR : Infrarouge



Ecrans

Vu de dessus

Connecteur HE10-10 pour le câble de programmation

Dels « Visu » position robot

P1 : Réglage du seuil de noir

Point test

Connecteur Externe

[JP1]

Cavaliers Programmation externe

Q1

Q2

Connecteur moteur roue droite [JP2]

I2

I4

Connecteurs alimentation

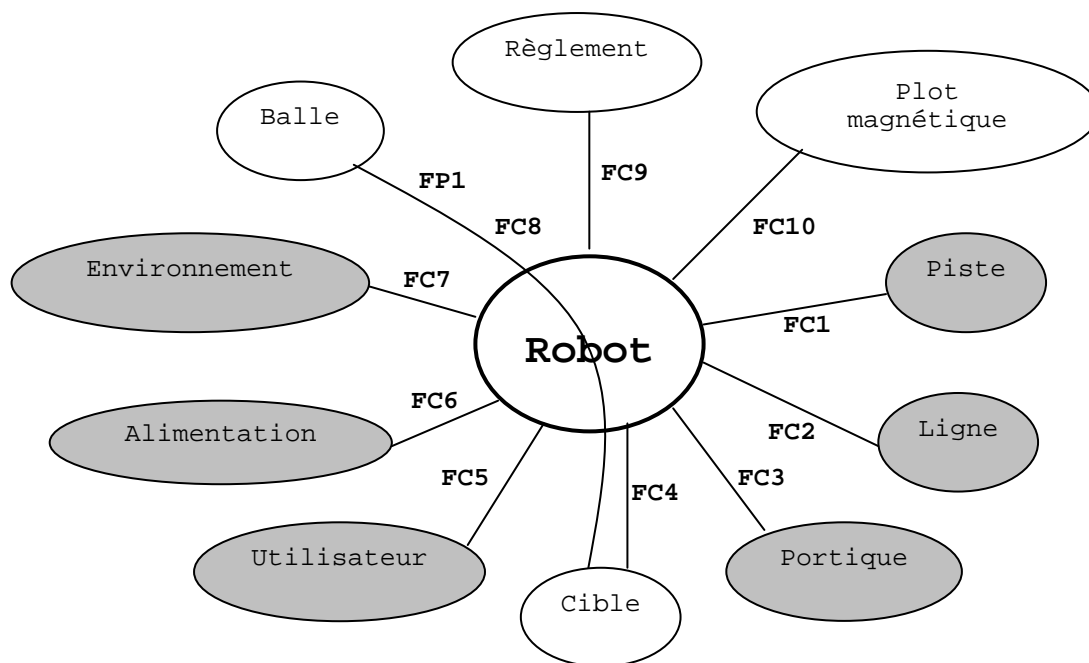
Marche/Arrêt

Connecteur moteur roue gauche [JP3]

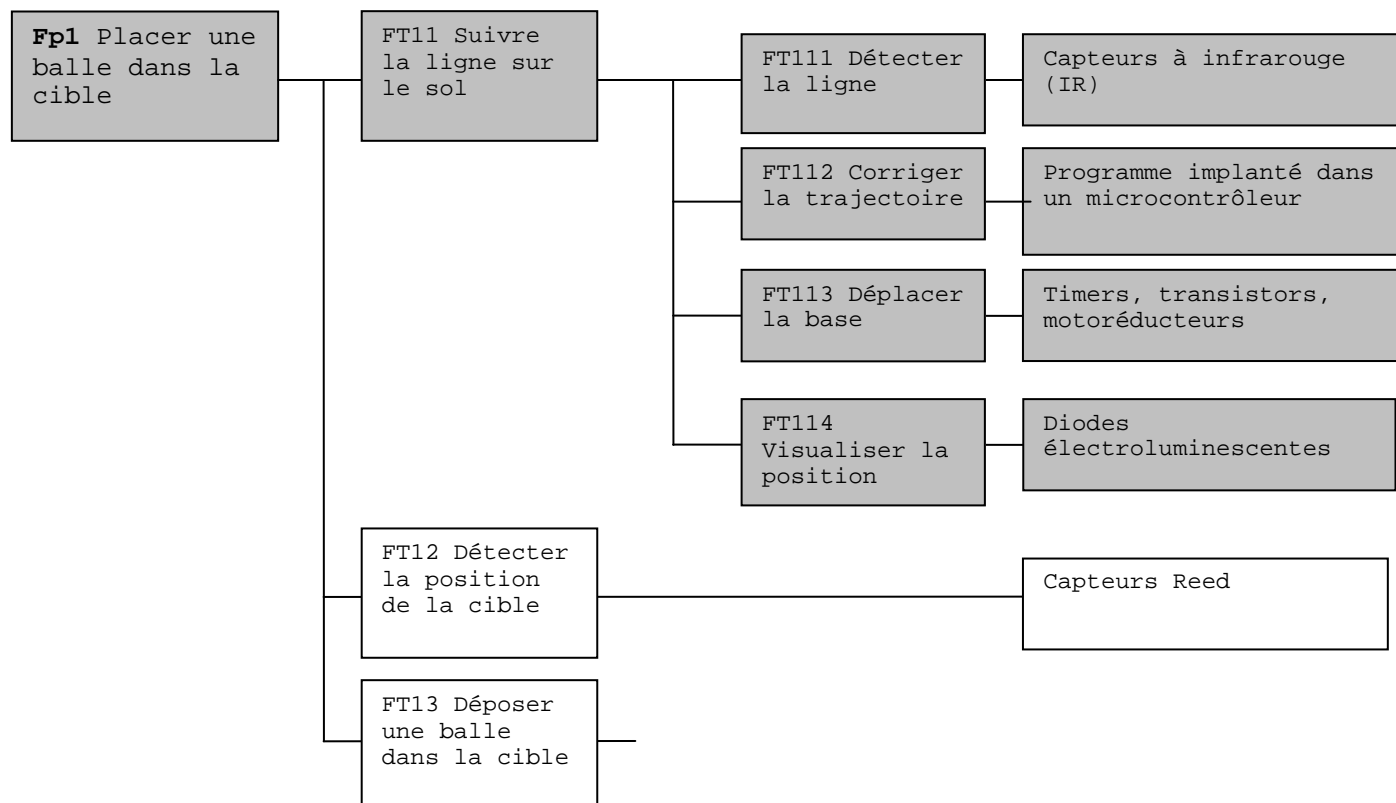
**Position des cavaliers**

Modes	Position des cavaliers	I1	I2	I3	I4	I5
Programmation		MOSI		MISO		Reset
Utilisation (Connecteur externe)		PB0	MotD Connecté	PB2	MotG Connecté	PB7

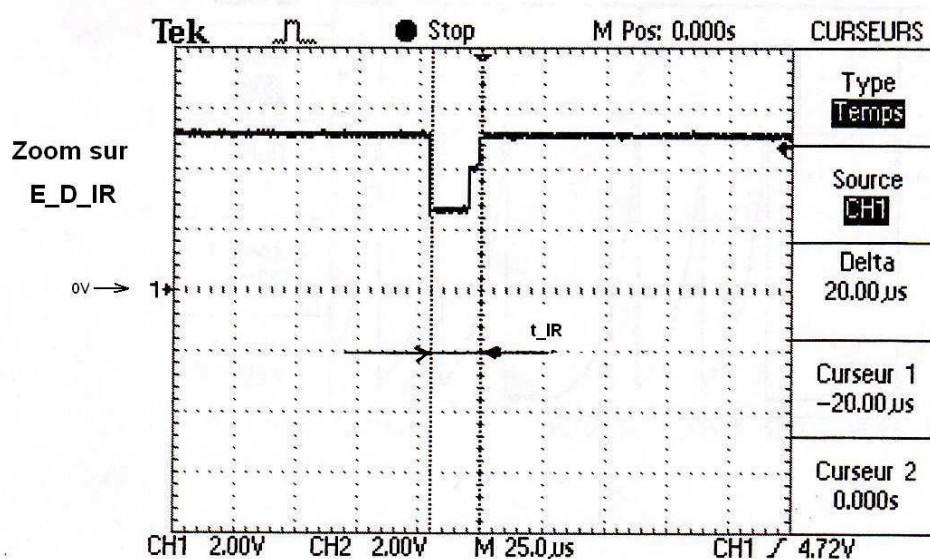
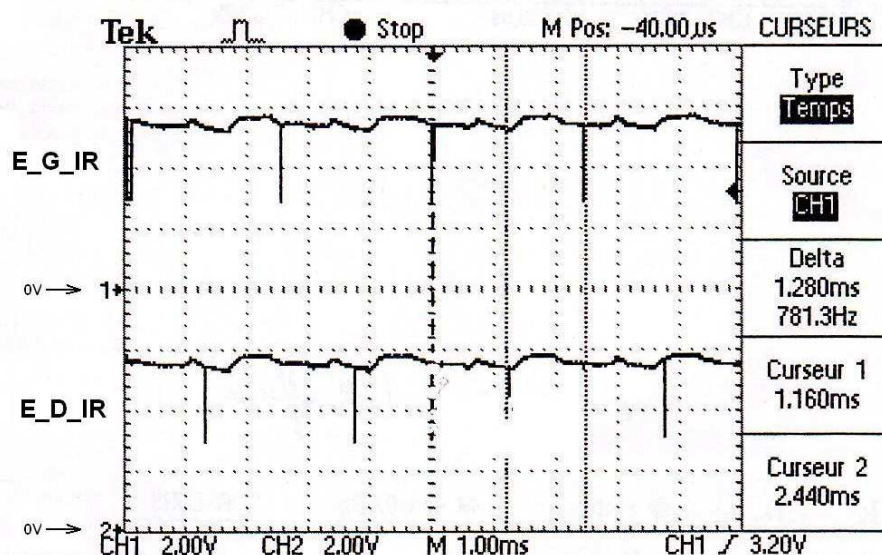
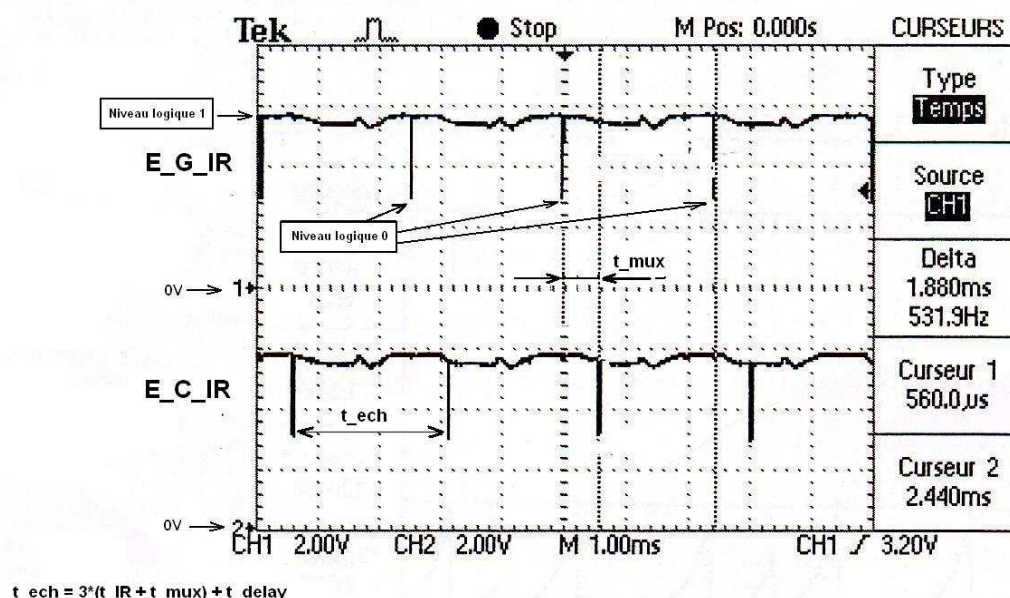
## Annexe 2 : Pieuvre



## Annexe 3 : Fast partiel







Remarque : 1-> et 2-> correspondent aux voies de l'oscilloscope.

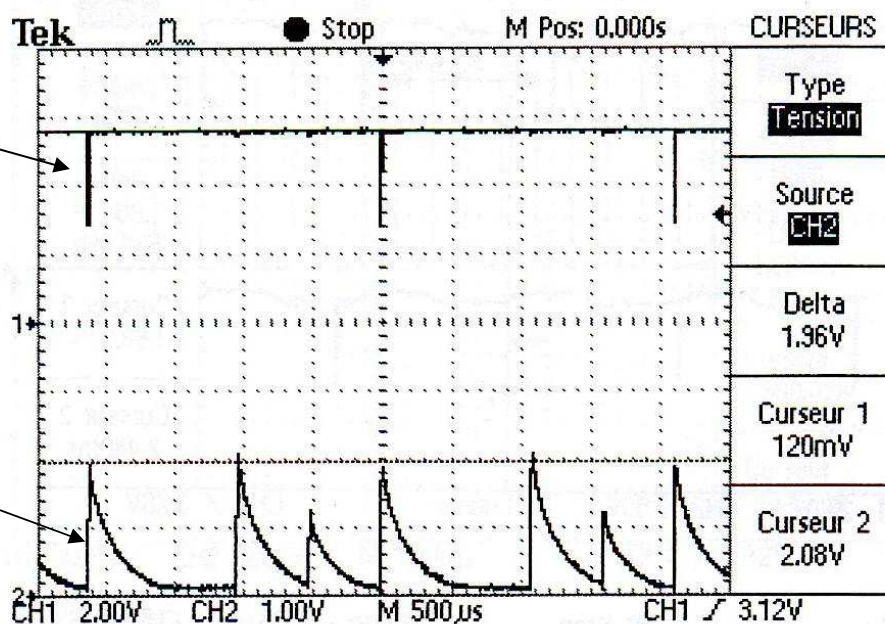
### CAPTEURS SUR LA LIGNE ( IR ABSORBE)

Commande de la diode infrarouge droite (DIR3) par un niveau bas => émission d'un signal infrarouge vers le sol.

E\_D\_IR

Avec la piste utilisée, le signal infrarouge (en grande partie absorbé par la bande noire) produit une tension de 2V (au maximum) à la sortie du phototransistor TIR3.

AIN1



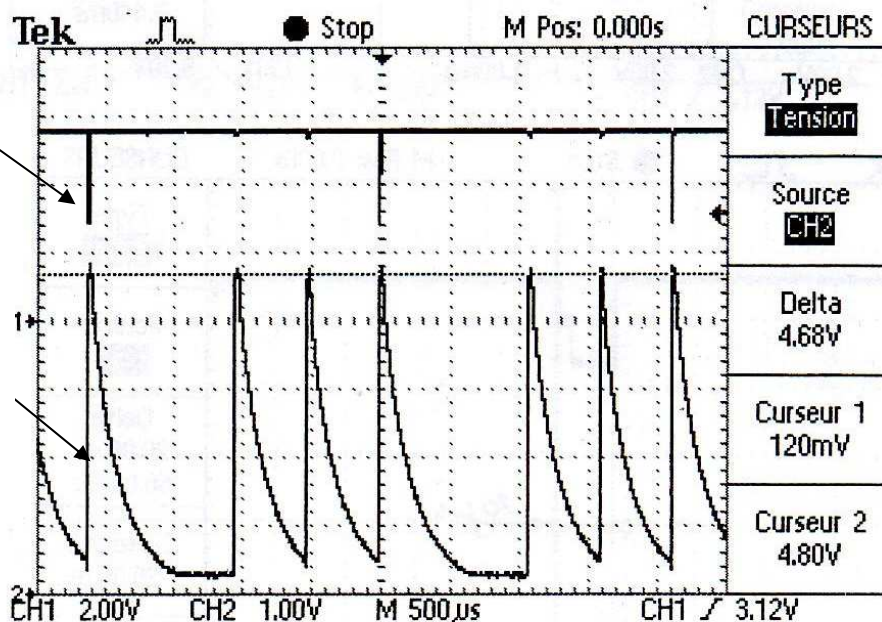
### CAPTEUR HORS LIGNE (IR REFLECHI)

Commande de la diode infra- rouge droite (DIR3) par un niveau bas => émission d'un signal infra rouge vers le sol.

E\_D\_IR

Avec la piste utilisée, le signal infrarouge réfléchi par le sol produit une tension de 5V à la sortie du phototransistor TIR3.

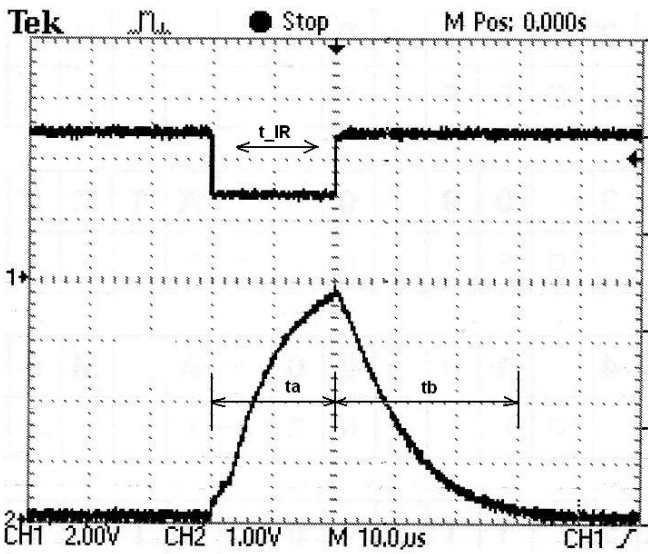
AIN1



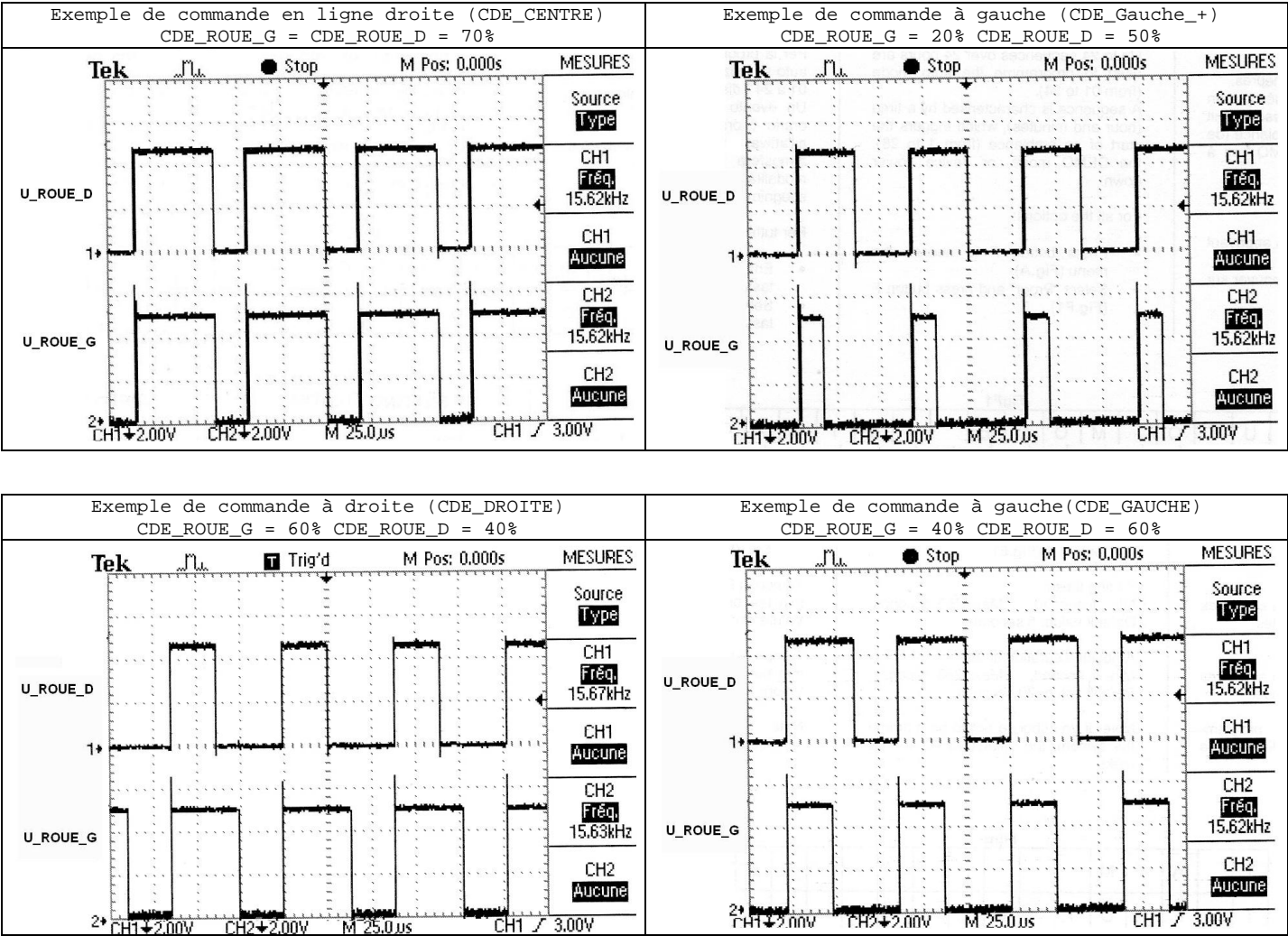
**Remarque** : 1-> et 2-> correspondent au 0V des voies de l'oscilloscope



**Annexe 5 :** Mesure sur FP3, zoom signal (AIN1) délivré par les récepteurs infrarouges (R\_IR)

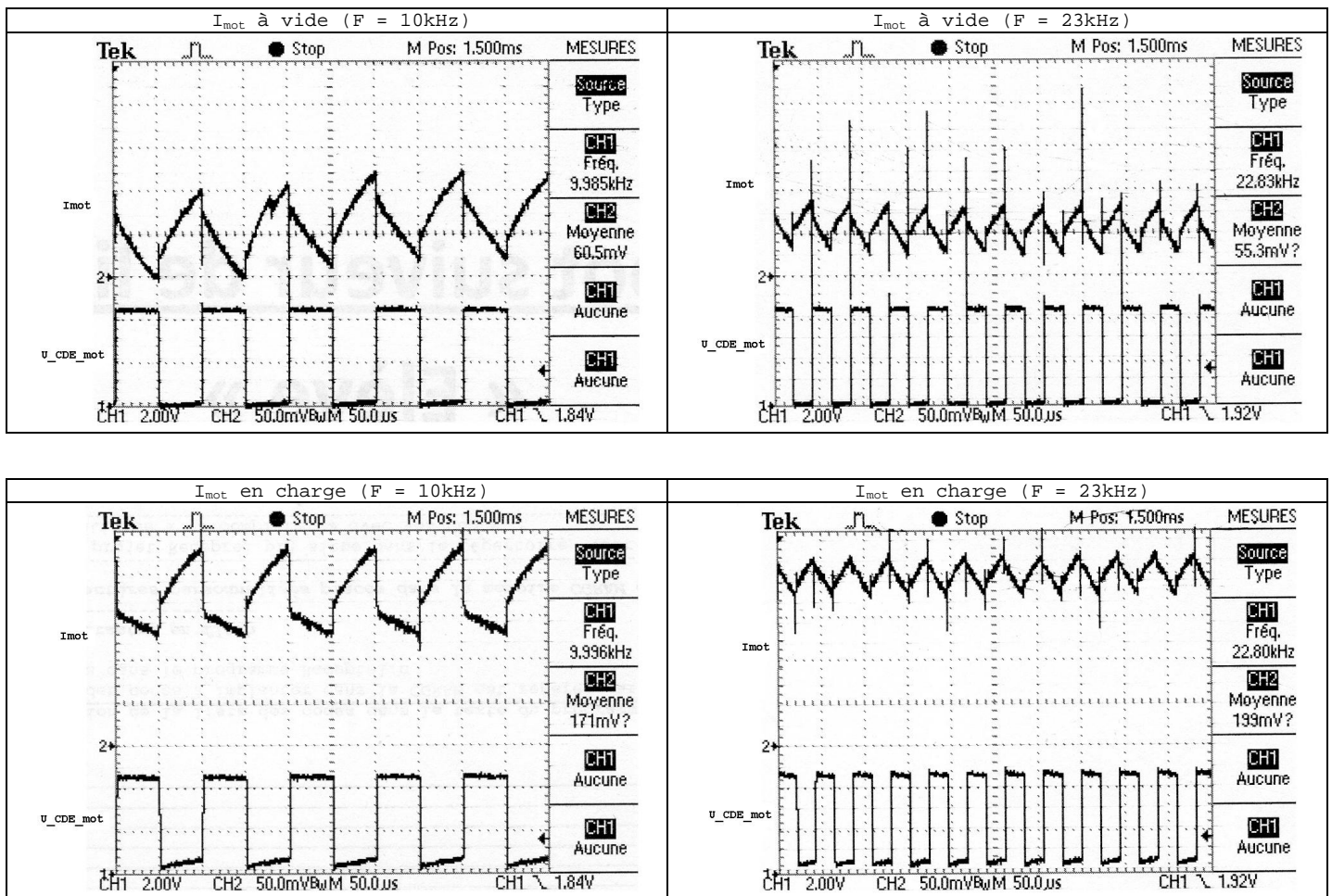


**Annexe 6 :** Mesures sur FP4(FP5), exemple de signaux MLI(PWM) de commande des moteurs

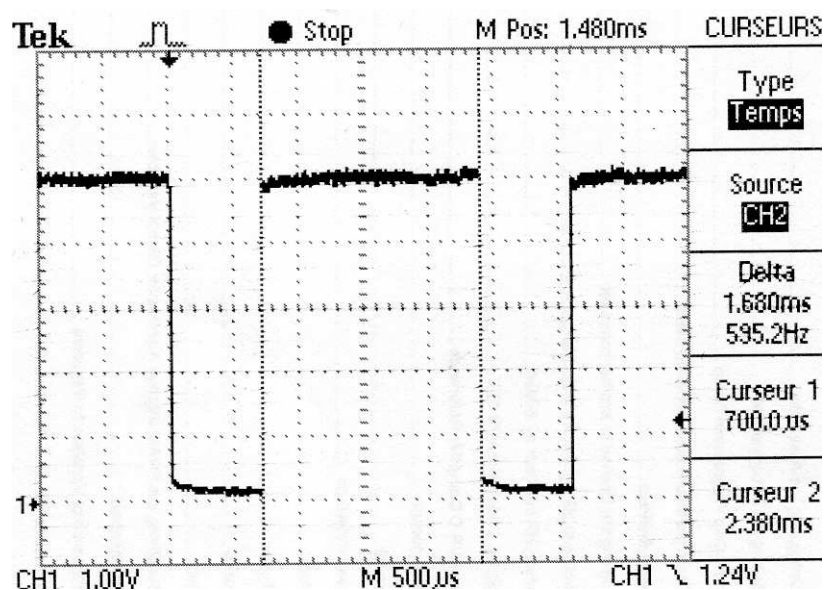


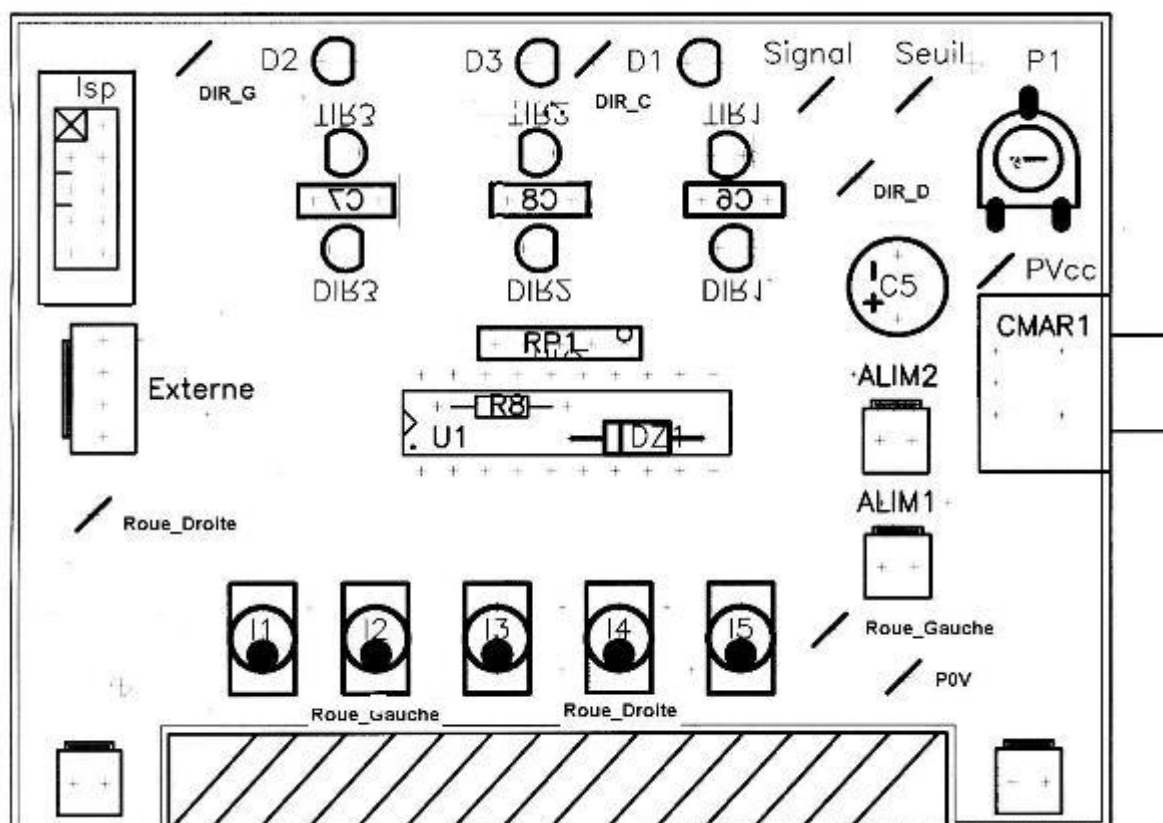
**Remarque :** 1-> et 2-> correspondent au 0V des voies de l'oscilloscope

**Annexe 7 : Mesures sur FP4(FP5), courant moteur  $I_{mot}$  mesuré avec une résistance de  $1\Omega$  placée en série avec le moteur**



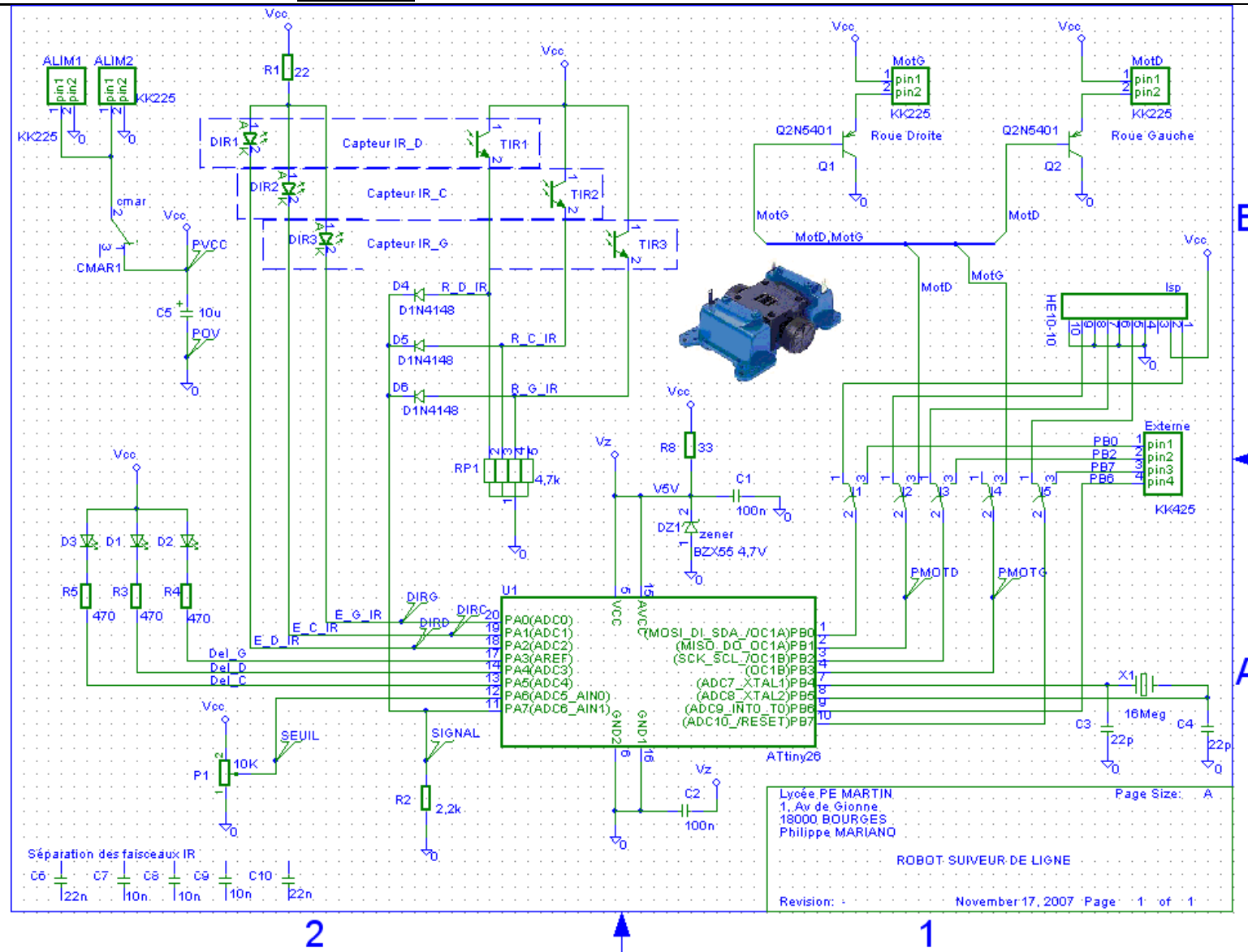
**Annexe 8 : Mesures sur FP6, signal de commande d'une Del « Visu »**







## Annexe 10 : Schéma structurel de la carte M<sup>r</sup> LineTiny



**Annexe 11 : Liste des composants de la carte « MrLineTiny »**

Item	Count	Mfg ID	Attributes	RefDes	Remarque
1	3	1N4148	DO-35	D4, D5, D6	
2	2	2N5401	TO-92	Q1, Q2	
3	1	ATtiny26	DIP20	U1	ATtiny26L
4	2	c	CK05 22n	C6, C10	céramique
5	2	c	CK05 100n	C1, C2	plastique
6	2	c	CK05 22p	C3, C4	céramique
7	3	c	CK05 10n	C7, C8, C9	
8	1	C+	CPR2-3 10u	C5	chimique 16V
9	1	CMAR	CMAR	CMAR1	
10	3	DELIR	DELIR	DIR1, DIR2, DIR3	Kingbright L-34F3C d=3mm $\lambda$ = 940nm
11	1	HE10-10	HE10-10	Isp	
12	5	inver	Inver_1	I1, I2, I3, I4, I5	
13	4	KK225	KK225	ALIM1, ALIM2, MotD, MotG	
14	1	KK425	KK425	Externe	
15	3	LED	LED	D1, D2, D3	rouge d=3mm
16	6	PTEST	PTEST	PMotD, PMotG, POV, PVcc, Seuil, Signal	
17	1	Px1	AJUST_H 10K	P1	
18	1	QUARTZ	HC-25U 16Meg	X1	
19	1	r	RC05 33	R8	
20	1	r	RC05 22	R1	
21	1	r	RC05 2,2k	R2	
22	3	r	RC05 470	R3, R4, R5	
23	1	R_SIL4+1	reseau5 4,7k	RP1	
24	3	TIR	TRIR	TIR1, TIR2, TIR3	Kingbright L-32P3C
25	1	Zener	DO-35 BZX55 4,7V	DZ1	

## Annexe 12 : Programme C implanté dans « MrLineTiny »

```

/*****
This program was produced by the
CodeWizardAVR V1.24.0 Standard
Automatic Program Generator
© Copyright 1998-2003 HP InfoTech s.r.l.
http://www.hpinfotech.ro
e-mail:office@hpinfotech.ro

Project : PPE Robot suiveur de ligne "Carte MrLineTiny"
Version : 2
Date    : 24/03/2008
Author  : Philippe Mariano
Company : LYCEE Pierre Emile MARTIN
Comments: Suivi de ligne

Fichier : StayOnLp.c

Chip type      : ATtiny26(L)
Clock frequency : 16,000000 MHz
Memory model   : Tiny
External SRAM size : 0
Data Stack size : 32
*****/

#include <tiny26.h>
#include <delay.h>

// Booléen
//-----
#define ACO      ACSR.5           // Sortie du comparateur analogique

// Temporisations
//-----
#define t_IR      19              // Durée de la commande d'un E_IR (Del IR)
#define t_mux     540             // Durée entre deux cde d'un E_IR
#define t_delay   704             // Permet de régler la période d'échantillonnage (tech) de la position
                                   // du robot par rapport à la ligne tech = 3*(t_IR + t_mux) + t_delay

```

```

//-----
// Commande des moteurs
//-----
#define CDE_ROUE_D      OCR1A      //   roue droite
#define CDE_ROUE_G      OCR1B      //   roue gauche

// Valeurs représentatives du rapport cyclique du signal
// de commande des moteurs (OC1A et OC1B non inversés)
//-----
#define _0PC            255          // 0%
#define _10PC           230          // 10%
#define _20PC           204          // 20%
#define _30PC           179          // 30%
#define _40PC           153          // 40%
#define _50PC           128          // 50%
#define _60PC           102          // 60%
#define _70PC            77          // 70%
#define _80PC            51          // 80%
#define _90PC            26          // 90%

// Connexions des Dels Visu et IR au port A
//-----

// Del_G   reliée à PA3 (D2)      E_G_IR PA0 (DIR3)
// Del_D   reliée à PA4 (D1)      E_C_IR PA1 (DIR2)
// Del_C   reliée à PA5 (D3)      E_D_IR PA2 (DIR1)

// Mots de commande des Dels Visu
// ----- Etat des Dels de visualisation-----

```

		// Del_G	Del_C	Del_D	Position	Remarque
#define Visu_Hors_ligne	0b00111111	// off	off	off	0	- - -
#define Visu_Droite_Plus	0b00110111	// off	off	on	1	* - -
#define Visu_Centre1	0b00011111	// off	on	off	2	- * -
#define Visu_Droite	0b00010111	// off	on	on	3	* * -
#define Visu_Gauche_Plus	0b00101111	// on	off	off	4	- - *
#define Visu_Centre2	0b00100111	// on	off	on	5	* - *
#define Visu_Gauche	0b00001111	// on	on	off	6	- * *
#define Visu_Centre3	0b00000111	// on	on	on	7	* * *

```

//-----
// Prototype des fonctions
//-----
void Arret_Moteurs();
void Active_Timer();

// Variables globales
// type          nom          Commentaires
//-----
// Les différentes commandes des moteurs
enum Etat_Commande_Moteur {DEPART, CDE_CENTRE1, CDE_CENTRE2, CDE_CENTRE3,
CDE_DROITE, CDE_DROITE_PLUS, CDE_DROITE_2PLUS, CDE_GAUCHE, CDE_GAUCHE_PLUS, CDE_GAUCHE_2PLUS}
EtatCdeMot = DEPART;

// Les différentes positions du robot par rapport à la ligne
enum Etat_Position_Robot {HORS_LIGNE, DROITE_PLUS, CENTRE_1, DROITE,
GAUCHE_PLUS, CENTRE_2, GAUCHE, CENTRE_3} Position_Robot = HORS_LIGNE;

//-----
// Début du programme principal
//-----

void main(void)
{

// Variables locales
// type          nom          Commentaires
//-----
unsigned char i;           // Compteur de boucles
unsigned char Cde_Del_IR;  // Permet de multiplexer la commande des del IR
unsigned char VisuPosition; // Affichage de la position du robot sur 3 del

// Input/Output Ports initialization
// Port A initialization
// Func0=Out Func1=Out Func2=Out Func3=Out Func4=Out Func5=Out Func6=In Func7=In
// State0=1 State1=1 State2=1 State3=1 State4=1 State5=1 State6=T State7=T
PORTA=0x3F;
DDRA=0x3F;

```

```

// Port B initialization
// Func0=In Func1=Out Func2=In Func3=Out Func4=In Func5=In Func6=In Func7=In
// State0=T State1=1 State2=T State3=1 State4=T State5=T State6=P State7=T
PORTB=0x0A;
DDRB=0x0A;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 4000,000 kHz
// Mode: PWMA & B top=OCR1C
// OC1A output: Non-Inv., /OC1A disconnected
// OC1B output: Non-Inv., /OC1B disconnected
PLLCSR=0x00;
TCCR1A=0xA3;
TCCR1B=0x03;
TCNT1=0x00;
OCR1A=0x00;
OCR1B=0x00;
OCR1C=0xFF;

// Analog Comparator initialization
// Analog Comparator: On
ACSR=0x00;

PORTA = 0x3F;    // Extinction des Dels

while (1)
{
    // début while
    Position_Robot = 0;    // Initialisation
    Cde_Del_IR = 1;

```

```

/*-----
Début de la tâche "Acquérir la position du robot par rapport à la ligne" (Intégrée dans FS21 et FS35)
-----
FS21 " Commander un des trois émetteurs IR(Del)"

Rôle : Sélectionner un des trois émetteurs (Del IR) devant générer un faisceau IR

Entrée :   Cde_Del_IR           octet permettant de multiplexer la commande des dels IR

Sorties:   PA0, PA1, PA2        bits de commande des dels IR (détails ci-dessous)
                                     Emmetteur IR connecté au bit      Remarque
                                     -----
                                     E_D_IR           "      PA2      |      Emmeteur IR droit
                                     E_C_IR           "      PA1      |      Emmeteur IR centre
                                     E_G_IR           "      PA0      |      Emmeteur IR gauche

//-----
FS35 "Calculer la position du robot par rapport à la ligne"

Rôle : Etablir une valeur numérique représentative de la position du robot par rapport à la ligne

Entrées :  AC0                  bit associé à la sortie du comparateur analogique,
                                     représentatif de la position d'un capteur sur la ligne

Sortie :   Position_Robot       octet image de la position du robot par rapport à la ligne)
                                     (détails ci-dessous)

Position des capteurs IR => "Position_Robot" // Positions du robot par rapport à la ligne
//-----
IR_D | IR_C | IR_G  =>   Position_Robot
non  | non  | non   =>   0      // Hors ligne
non  | non  | oui   =>   1      // Droite+
non  | oui  | non   =>   2      // Centre1 (ligne noire de largeur égale à celle d'un capteur)
non  | oui  | oui   =>   3      // Droite
oui  | non  | non   =>   4      // Gauche+
oui  | non  | oui   =>   5      // Centre2 (ligne blanche entourée de deux noires)
oui  | oui  | non   =>   6      // Gauche
oui  | oui  | oui   =>   7      // Centre3 (ligne noire de largeur égale à celle de trois capteurs)
//-----*/

```

```

for (i=0;i<3;i++)
{
    PORTA = ~Cde_Del_IR;           // Les E_IR (Leds IR) sont commandés à l'état bas
    delay_us(t_IR);                // Durée de l'état bas (A ajuster en fonction des capteurs)
    if (ACO == 1) Position_Robot = Position_Robot | Cde_Del_IR; // Si le capteur IR_x x=(D,C,G] est sur la ligne
    PORTA = 0x3F;                  // le bit correspondant de "Position_Robot est mis à 1
    Cde_Del_IR = Cde_Del_IR << 1 ;
    delay_us(t_mux);               // Tempo permettant d'attendre l'annulation de la ddp sur
                                   // l'entrée AIN1 (signal de réception IR) avant une nouvelle
                                   // mesure (A ajuster en fonction des capteurs)
}

// Fin de la tâche "Acquérir la position du robot par rapport à la ligne" (Fonctions FS21 et FS35)
//-----

/* -----
Début de la tâche "Traiter l'information "Position_Robot" pour visualiser la position du robot par rapport à la ligne"
(Fonction FP1)
-----

Rôle :      Lorsque le capteur est sur la ligne, la Del correspondante s'éclaire.
Entrée:     Position_Robot      (voir ci-dessus)
Sorties:     PortA              octet
                                   Del de visualisation connectée au bit
                                   -----
                                   Del_D      "          PA4
                                   Del_C      "          PA5
                                   Del_G      "          PA3
                                   Remarque : D pour droite, C pour centre, G pour gauche
                                   -----

switch (Position_Robot)           // Etat des Dels de visualisation
{
    case 0: VisuPosition = Visu_Hors_ligne; break; // Del_G | Del_C | Del_D
    case 1: VisuPosition = Visu_Droite_Plus; break; // off  | off  | on
    case 2: VisuPosition = Visu_Centrel; break;    // off  | on   | off
    case 3: VisuPosition = Visu_Droite; break;     // off  | on   | on
    case 4: VisuPosition = Visu_Gauche_Plus; break; // on   | off  | off
    case 5: VisuPosition = Visu_Centre2; break;    // on   | off  | on
    case 6: VisuPosition = Visu_Gauche; break;     // on   | on   | off
    case 7: VisuPosition = Visu_Centre3; break;    // on   | on   | on
}

// Fin de la tâche "Traiter l'information "Position_Robot" pour visualiser la position du robot par rapport à la ligne" (Fonction FP1)
// -----

```



```

PORTA = VisuPosition; // Fonction FS61 "Sélectionner les delts de visualisation"

/* -----
Début de la tâche "Traiter l'information "Position_Robot" pour corriger la trajectoire du robot" (Fonction FP1)
-----
Rôle: Régler le rapport cyclique des signaux de commande des moteurs droit et gauche pour régler la trajectoire
du robot afin qu'il suive la ligne

Entrée:      Position_Robot                (voir ci-dessus)

Sorties:      CDE_ROUE_G                    octet  (OCR1A) règle le rapport cyclique du signal de commande du moteur Gauche
              CDE_ROUE_D                    (OCR1B) règle le rapport cyclique du signal de commande du moteur Droit
-----*/

switch (EtatCdeMot)
{ // Début switch (EtatCdeMot)
  case DEPART: Arret_Moteurs();
    if ((Position_Robot == CENTRE_1) || (Position_Robot == CENTRE_2) || (Position_Robot == CENTRE_3))
    {
      delay_ms(3000);
      Active_Timer();
      EtatCdeMot = CDE_CENTRE1;
    }
    else EtatCdeMot = DEPART;

  break;

  case CDE_CENTRE1: CDE_ROUE_G = _60PC; CDE_ROUE_D = _60PC;
    if (Position_Robot == GAUCHE) EtatCdeMot = CDE_DROITE;
    else if (Position_Robot == DROITE) EtatCdeMot = CDE_GAUCHE;
    else EtatCdeMot = CDE_CENTRE1;

  break;

  case CDE_CENTRE2: EtatCdeMot = CDE_CENTRE1;

  break;

  case CDE_CENTRE3: EtatCdeMot = CDE_CENTRE1;

  break;
}

```

```

case CDE_DROITE:CDE_ROUE_G = _60PC; CDE_ROUE_D = _50PC;
    if (Position_Robot == CENTRE_1) EtatCdeMot = CDE_CENTRE1;
    else if (Position_Robot == GAUCHE_PLUS) EtatCdeMot = CDE_DROITE_PLUS;
    else EtatCdeMot = CDE_DROITE;
break;

case CDE_DROITE_PLUS:CDE_ROUE_G = _60PC; CDE_ROUE_D = _40PC;
    if (Position_Robot == GAUCHE) EtatCdeMot = CDE_DROITE;
    else if (Position_Robot == HORS_LIGNE) EtatCdeMot = CDE_DROITE_2PLUS;
    else EtatCdeMot = CDE_DROITE_PLUS;
break;

case CDE_DROITE_2PLUS:CDE_ROUE_G = _50PC; CDE_ROUE_D = _20PC;
    if (Position_Robot == GAUCHE_PLUS) EtatCdeMot = CDE_DROITE_PLUS;
    else EtatCdeMot = CDE_DROITE_2PLUS;
break;

case CDE_GAUCHE: CDE_ROUE_G = _50PC; CDE_ROUE_D = _60PC;
    if (Position_Robot == CENTRE_1) EtatCdeMot = CDE_CENTRE1;
    else if (Position_Robot == DROITE_PLUS) EtatCdeMot = CDE_GAUCHE_PLUS;
    else EtatCdeMot = CDE_GAUCHE;
break;

case CDE_GAUCHE_PLUS: CDE_ROUE_G = _40PC; CDE_ROUE_D = _60PC;
    if (Position_Robot == DROITE) EtatCdeMot = CDE_GAUCHE;
    else if (Position_Robot == HORS_LIGNE) EtatCdeMot = CDE_GAUCHE_2PLUS;
    else EtatCdeMot = CDE_GAUCHE_PLUS;
break;

case CDE_GAUCHE_2PLUS:CDE_ROUE_G = _20PC; CDE_ROUE_D = _50PC;
    if (Position_Robot == DROITE_PLUS) EtatCdeMot = CDE_GAUCHE_PLUS;
    else EtatCdeMot = CDE_GAUCHE_2PLUS;
break;

default : EtatCdeMot = CDE_CENTRE1; break;

} // Fin switch (EtatCdeMot)

```

```

// Fin de la tâche "Traiter l'information "Position_Robot" pour corriger la trajectoire du robot" (Fonction FP1)
// -----

```

```

    delay_us(t_delay); // A ajuster en fonction de la vitesse du robot

} // fin while(1);
} // fin prog princ

// -----
// Sous-programmes
// -----
void Arret_Moteurs()
{
    TCCR1A = 0x00; // Arrêt PWM
    TCCR1B = 0x00;
    PORTB=0x0B;    // Arrêt Moteur
}

void Active_Timer()
{ // Marche PWM
    TCCR1A=0xA3;
    TCCR1B=0x03;
}

```