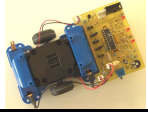




Table des matières

| | |
|--|----|
| 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. Etude 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 FP1 « Traiter l'information... » | 9 |
| 2.5.2 FP2 « Acquérir la position d'un capteur par rapport à la ligne » | 12 |
| 2.5.3 FP3(FP3bis) « Distribuer » | 13 |
| 2.5.4 FP6 « Communiquer la position du robot par rapport à la ligne » | 14 |
| 3. Etude structurelle | 16 |
| 3.1 Etude structurelle de FP1 « Traiter » | 16 |
| 3.2 Etude structurelle de FP2 « Acquérir la position du robot par rapport à la ligne » | 16 |
| 3.2.1 FS22 « Emettre un rayon infrarouge » | 16 |
| 3.2.2 FS23 « Détecter un rayon infrarouge » | 16 |
| 3.2.3 FS24 « Sélectionner un récepteurs d'infrarouges » | 16 |
| 3.2.4 FS25 « Régler un seuil de tension » | 17 |
| 3.2.5 FS26 « Comparer » | 17 |
| 3.3 Etude structurelle de FP3(FP3bis) « Distribuer » | 17 |
| 3.3.1 Principe de réglage de la fréquence de rotation des moteurs | 17 |
| 3.3.2 Génération d'un signal M.L.I avec le microcontrôleur ATINY26 | 18 |
| 3.3.3 Identification des structures associées à FS31(FS31bis) | 18 |
| 3.3.4 Etude structurelle de FS32(FP32bis) « Amplifier » | 19 |
| 3.4 Etude structurelle de FP4(FP4bis) « Convertir une énergie électrique en énergie mécanique » | 19 |
| Annexe 1 : Description de la carte MrLineTiny | 20 |
| Annexe 2 : Diagramme pieuvre | 21 |
| Annexe 3 : Diagramme fast partiel | 21 |
| Annexe 4 : Exemples de relevés des signaux E_G_IR, E_C_IR et E_D_IR | 22 |
| Annexe 5 : Exemples de relevés du signal SGN(AIN1) délivré par les récepteurs d'infrarouges (R_IR) | 23 |
| Annexe 6 : Exemples de relevés des signaux MLI(PWM) pour la commande des moteurs.... | 24 |
| Annexe 7 : Exemples de relevés du courant moteur I_{mot} mesurés avec une résistance de 1Ω placée en série avec le moteur droit | 25 |
| Annexe 8 : Exemple de relevé du signal de commande d'une LED « Visu » | 25 |
| Annexe 9 : Schéma d'implantation de la carte « MrLineTiny » | 26 |
| Annexe 10 : Schéma structurel de la carte M ^r LineTiny | 27 |
| Annexe 11 : Liste des composants de la carte « MrLineTiny » | 28 |
| Annexe 12 : Programme C implanté dans le microcontrôleur de « MrLineTiny » | 29 |

| | | | | |
|--|---|---|-------------------------------|---|
| Dossier Technique | TS SI |  | P.P.E. Robot suiveur de ligne |  |
|  Lycée Polyvalent PIERRE EMILE MARTIN | Dossier technique | | | Dernière mise à jour le 19/11/2009 |
| Sites à consulter | Microrobot (http://www.microrobotusa.com) Lextronic (http://www.lextronic.fr) WEBGE (http://p.mariano.free.fr) rubrique PPE | | | |

1. Mise en situation

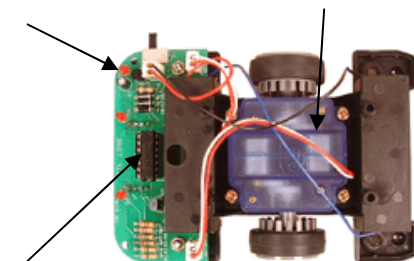
1.1 Expression du besoin



La participation de notre établissement à un challenge académique de robotique à 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).

Circuit imprimé

Base motorisée



Microcontrôleur

Mr Line

1.2 Historique

En 2006 le robot « M^r Line » se composait de deux parties :

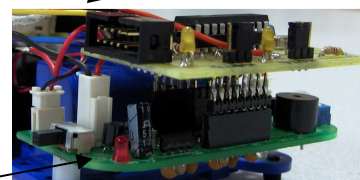
- une **base motorisée** et,
- un **circuit imprimé**.



Le circuit imprimé servait de support aux trois capteurs d'infrarouges 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.

Extension ATINY26



Adaptation ATINY26

Circuit imprimé

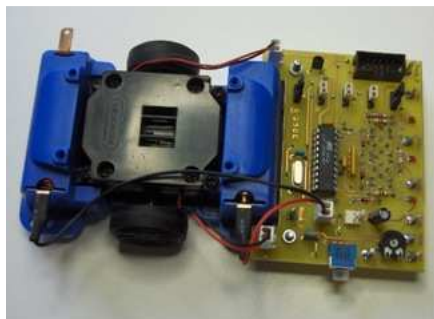
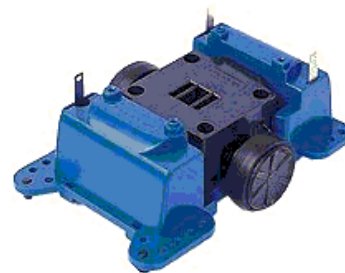


MrLineTiny

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

Le robot suiveur de ligne « M^r Line » a actuellement disparu des catalogues, la base motorisée est toujours distribuée (2008) par la société [Lextronic](#) sous la référence MRM-010.

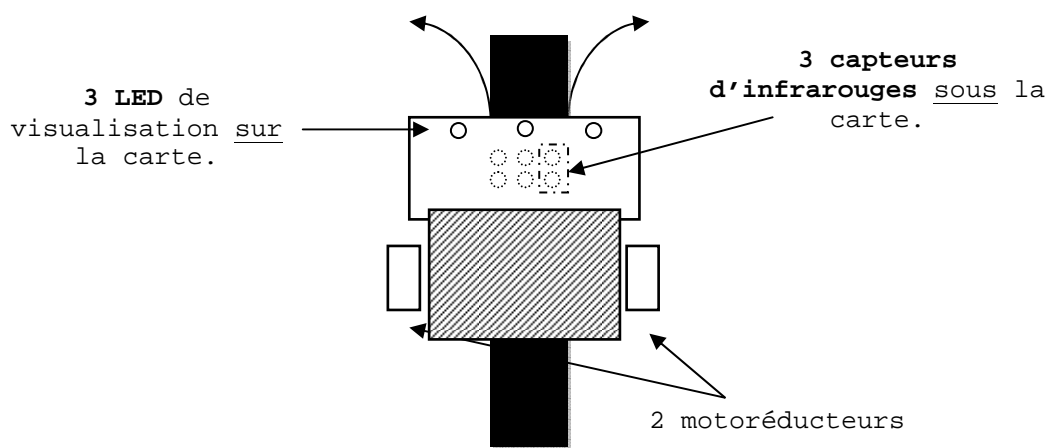
Les caractéristiques de cette base motorisée sont décrites dans les paragraphes 3.4 et 3.5.



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

1.3 Principe du suivi de la ligne adopté par MrLineTiny

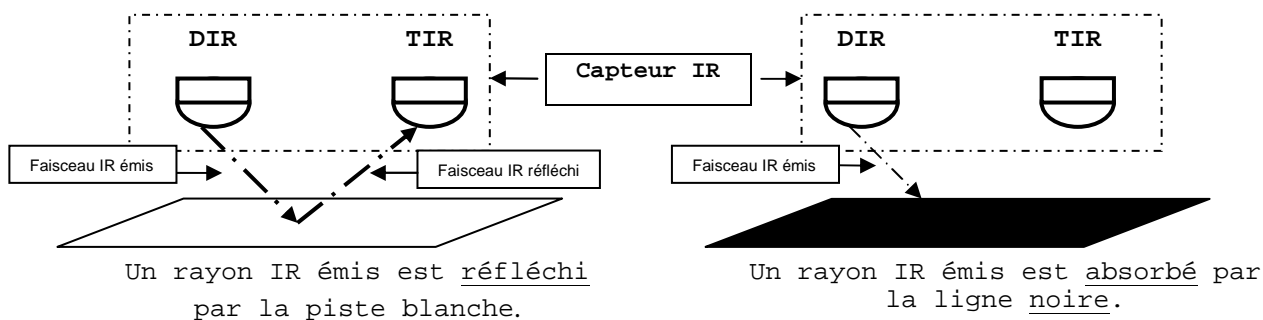
D'après le règlement du concours académique, le robot doit suivre une ligne noire de 30mm de large tracée sur un fond blanc.



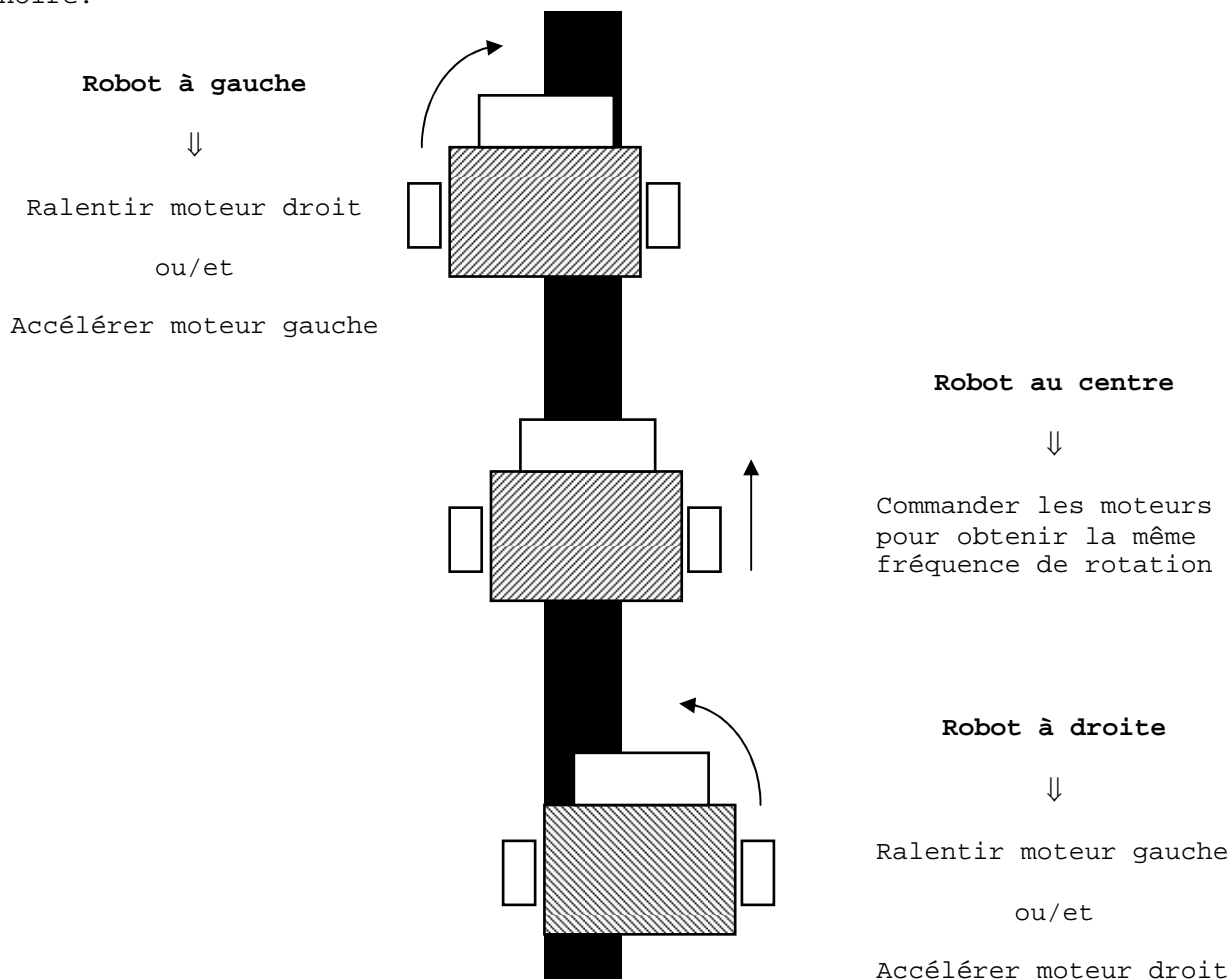
Le robot détecte sa position par rapport à la ligne noire tracée sur la piste grâce à **trois capteurs** d'infrarouges (IR). Ces capteurs sont identifiés par les sigles **Capteur_IR_Gauche (IR_G)**, **Capteur_IR_Centre (IR_C)** et **Capteur_IR_Droit (IR_D)**. (Schéma structurel en annexe 10)

Chaque capteur se compose d'un **émetteur DIR** (diode IR) et d'un **récepteur TIR** (phototransistor IR).

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

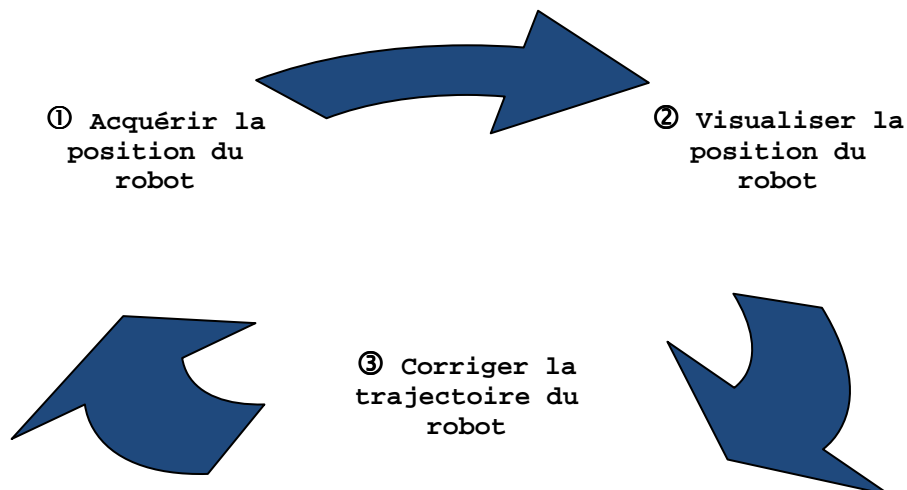


Les informations délivrées par les capteurs sont représentatives de la **position du robot** par rapport à la ligne. Elles sont **traitées** par un programme implanté dans un **microcontrôleur**. Grâce à une structure appelée « **Timer** », le microcontrôleur adapte la fréquence de rotation de chacun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



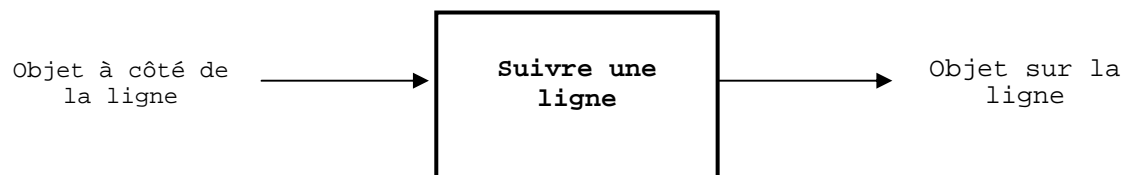
La position du robot par rapport à la ligne est visualisée par l'état (éclairée ou éteinte) de trois **diodes électroluminescentes** (LED).

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



2. Etude 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 d'une source lumineuse (soleil, éclairage artificiel).

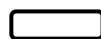
- Technique

La ligne est tracée avec une bande adhésive noire (ou peinte) sur une surface blanche réfléchissante.

- Économique

Le prix des composants électroniques et mécaniques nécessaires à la réalisation du robot ne dépasse pas 50 euros (en 2008).

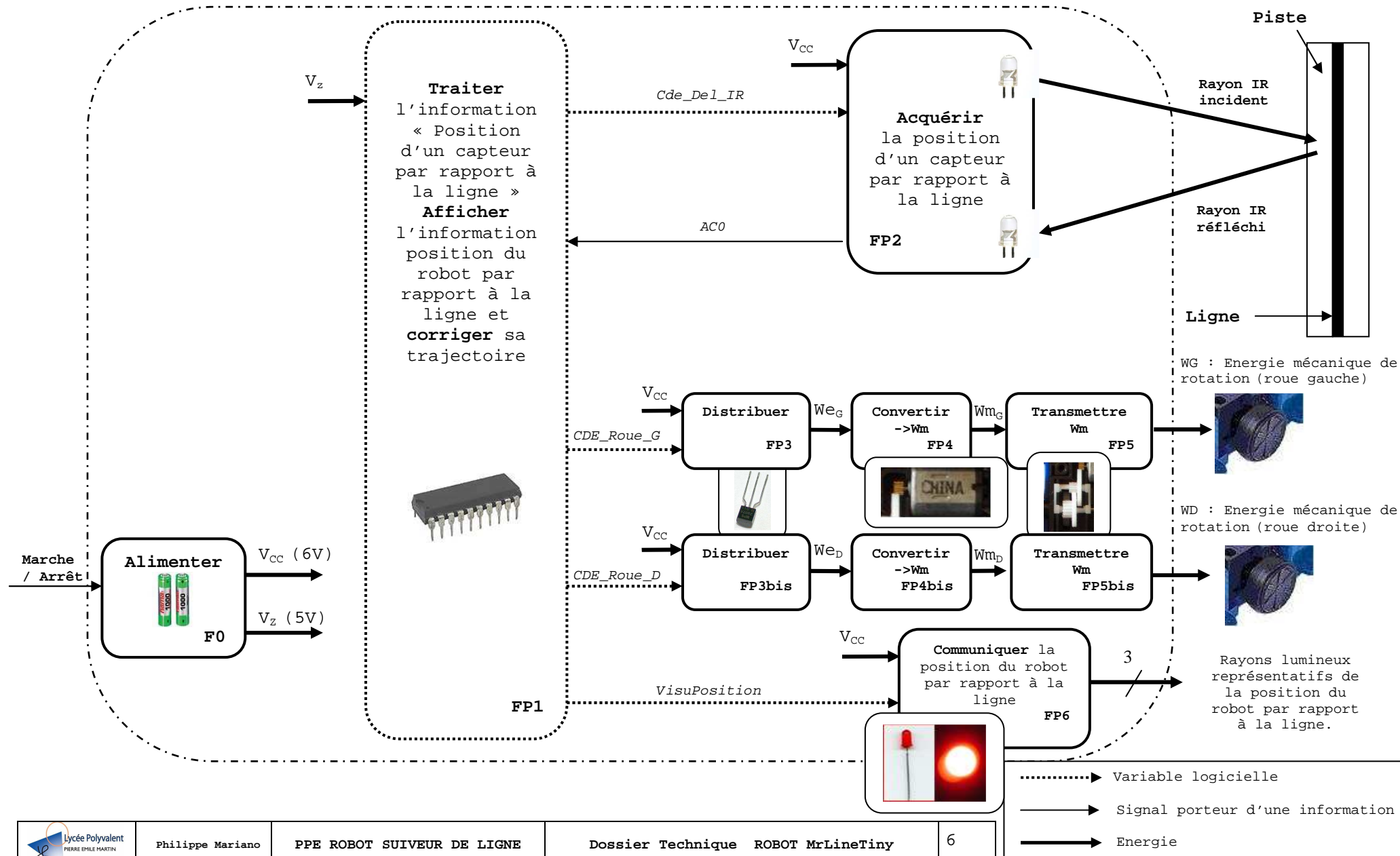
2.3 Schéma fonctionnel de premier degré



Fonction matérielle



Fonction logicielle



2.4 Description des fonctions principales

FP1 : Fonction « Traiter l'information... »

Rôles: Commander les émetteurs d'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

- **AC0:** signal logique représentatif de la position d'un des trois capteurs d'infrarouges par rapport à la ligne.
AC0 = 1 si le capteur sélectionné est sur la ligne.

Sorties

- **CDE_Roue_G, CDE_Roue_D:** variables numériques (octets non signés) représentatives du rapport cyclique du signal de commande des moteurs.

- **VisuPosition:** variable numérique (octet non signé) représentative de la position du robot par rapport à la ligne.

- **Cde_Del_IR :** variable numérique (octets non signés) permettant de sélectionner un émetteur d'infrarouges(IR).

FP2 : Fonction « Acquérir la position d'un capteur par rapport à la ligne »

Rôle: Explorer la piste avec un rayon infrarouge, capter le rayon réfléchi et le transformer en une information logique représentative de la position d'un capteur par rapport à la ligne.

Entrées

- **Cde_Del_IR :** voir FP1.
- **Rayon IR réfléchi:** rayon infrarouge réfléchi par la piste.

Sorties

- **Rayon IR incident:** rayon infrarouge émis vers la piste.
- **AC0 :** voir FP1.

FP3 (FP3bis): Fonction « Distribuer »

Rôles: Régler le rapport cyclique α et augmenter la puissance du signal de commande du moteur gauche (droit).

Entrée

- **CDE_Roue_G (CDE_Roue_D):** voir FP1.

Sortie

- **We_G (We_D) :** signal logique de fréquence fixe et de rapport cyclique variable amplifié en puissance.

FP4 (FP4bis): Fonction « Convertir »

Rôle: Convertir une énergie électrique en énergie mécanique.

Entrée

- **WeG (WeD):** voir FP3(FP3bis)

Sortie

- **Wm_G (Wm_D):** Energie mécanique de rotation.

FP5 (FP5bis) : Fonction « Transmettre »

Rôle: Transmettre l'énergie mécanique issue de FP4(FP4bis) en diminuant la fréquence de rotation et en augmentant le couple.

Entrée

- W_m (W_D): voir FP4(FP4bis)

Sorties

- W_g (W_D) : Energie mécanique de rotation.

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

Rôle: Informer l'utilisateur sur la position du robot par rapport à la ligne. Les LEDs s'éclairent lorsque le capteur IR associé est sur la ligne noire.

Entrée

- VisuPosition : voir FP1.

Sorties

- Rayons lumineux.

F0: Fonction « Alimentation »

Rôle: Fournir et adapter l'énergie électrique nécessaire aux différentes fonctions.

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

- V_z : tension stabilisée. ($V_z \approx 5V$).

Rappel des conventions utilisées dans les schémas fonctionnels



Fonction matérielle



Fonction logicielle

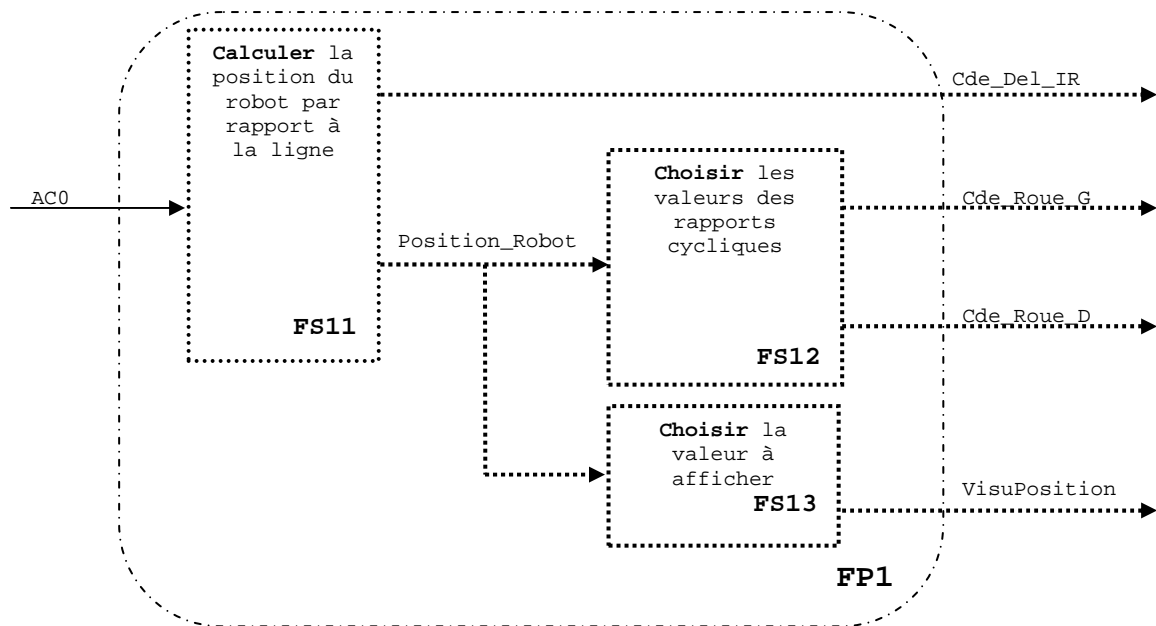
.....► Variable logicielle

————► Signal porteur d'une information

————► Energie

2.5 Description des fonctions secondaires

2.5.1 FP1 « Traiter l'information... »



FS11: « Calculer la position du robot par rapport à la ligne »

Rôle : Commander les émetteur IR et calculer la valeur numérique représentative de la position du robot par rapport à la ligne noire.

Entrée

- **AC0:** signal logique représentatif de la position d'un des trois capteurs d'infrarouges par rapport à la ligne noire (signal interne au microcontrôleur). AC0 = 1 si le capteur sélectionné est sur la ligne.

Sorties

- **Position_Robot:** Variable numérique de type octet ; image de la position du robot par rapport à la ligne. Seuls les trois BIT de poids faible sont significatifs. (Voir l'algorithme page suivante)

| Position_Robot ₍₂₎ | Position du robot par rapport à la ligne |
|-------------------------------|--|
| XXXXX000 | Hors ligne. |
| XXXXX001 | Gauche+ : « très à gauche de la ligne ». |
| XXXXX010 | Centre1 : détection d'une ligne noire de largeur égale à celle de la zone de détection d'un capteur. |
| XXXXX011 | Gauche : « à gauche de la ligne ». |
| XXXXX100 | Droite+ : « très à droite de la ligne ». |
| XXXXX101 | Centre2 : détection d'une ligne blanche entourée de deux lignes noire de largeur égale à celle de la zone de détection d'un capteur. |
| XXXXX110 | Droite : « à droite de la ligne ». |
| XXXXX111 | Centre3 : détection d'une ligne noire de largeur égale à la zone de détection de trois capteurs. |

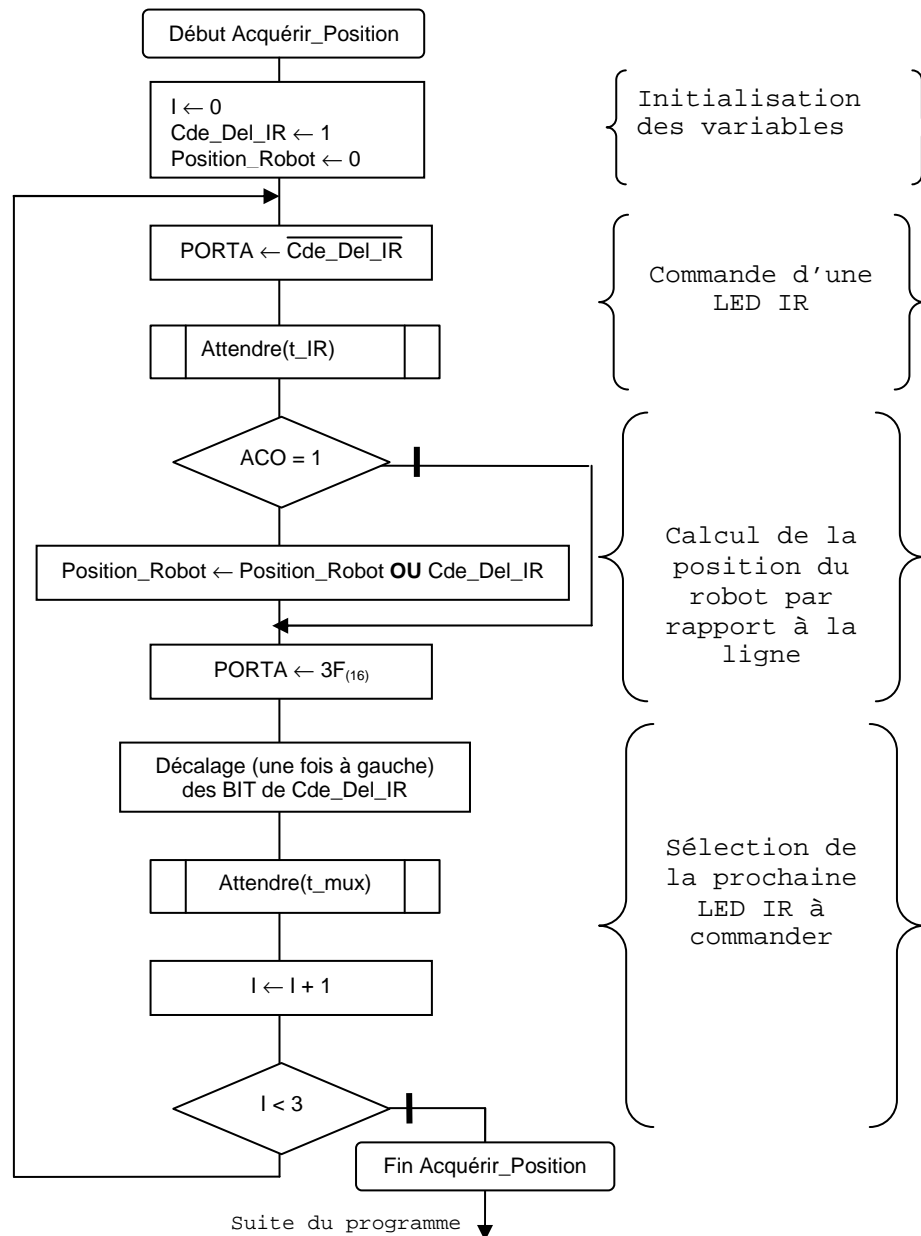
X : Etat indifférent

- **Cde_Del_IR :** variable numérique de type octet permettant de sélectionner un émetteur d'infrarouges(IR) parmi les trois. Seuls les trois BITS de poids faible sont significatifs. (Voir l'algorithme page suivante)

| Cde_Del_IR ₍₁₆₎ | Emetteur IR commandé |
|----------------------------|----------------------|
| FF | Aucun |
| FB | Emetteur IR gauche |
| FD | Emetteur IR centre |
| FE | Emetteur IR droit |

Algorithme du programme correspondant à FS11

| Nom | Format | Commentaires |
|-----------------------|--------|--|
| I | octet | 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. |
| <i>Constantes</i> | | |
| t_IR | octet | Durée d'activation d'un émetteur IR _i $i \in \{1,2,3\}$ |
| t_mux | mot16 | Durée avant l'activation du prochain émetteur IR _(i+1) |



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

FS12: « Choisir les valeurs des rapports cycliques »

Rôle : choisir les valeurs des rapports cycliques des signaux de commande des moteurs du robot en fonction de sa position par rapport à la ligne noire.

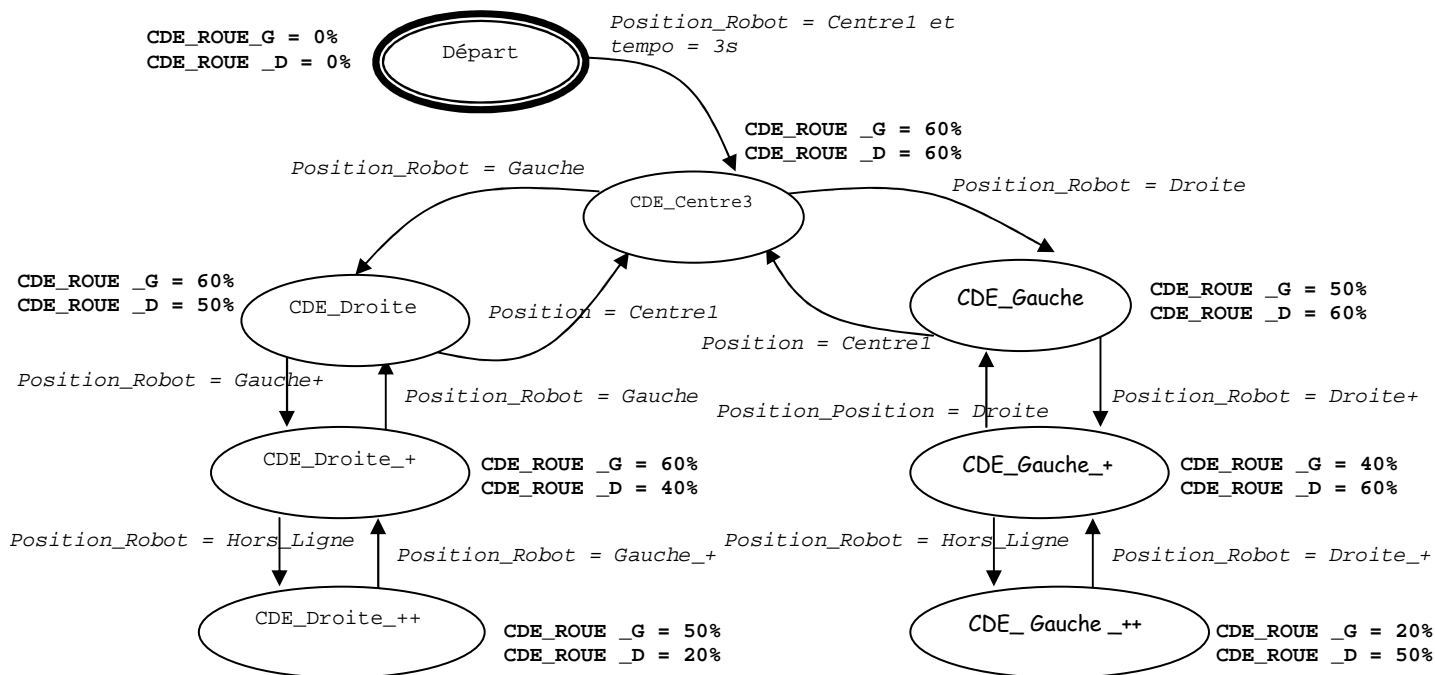
Entrée

- **Position_Robot:** voir FS11.

Sorties

- **CDE_Roue_G, CDE_Roue_D:** Variables (octets non signés) représentatives des rapports cycliques des signaux de commande des moteurs. L'affectation de valeurs à ces variables est décrite par le graphe des transitions ci-dessous.

Remarque : 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**.



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 !

FS13: « Choisir la valeur à afficher »

Rôle : Choisir la valeur numérique nécessaire à la commande des LED rouge pour que leur état éclairé soit représentatif de la position du robot par rapport à la ligne noire.

Entrée

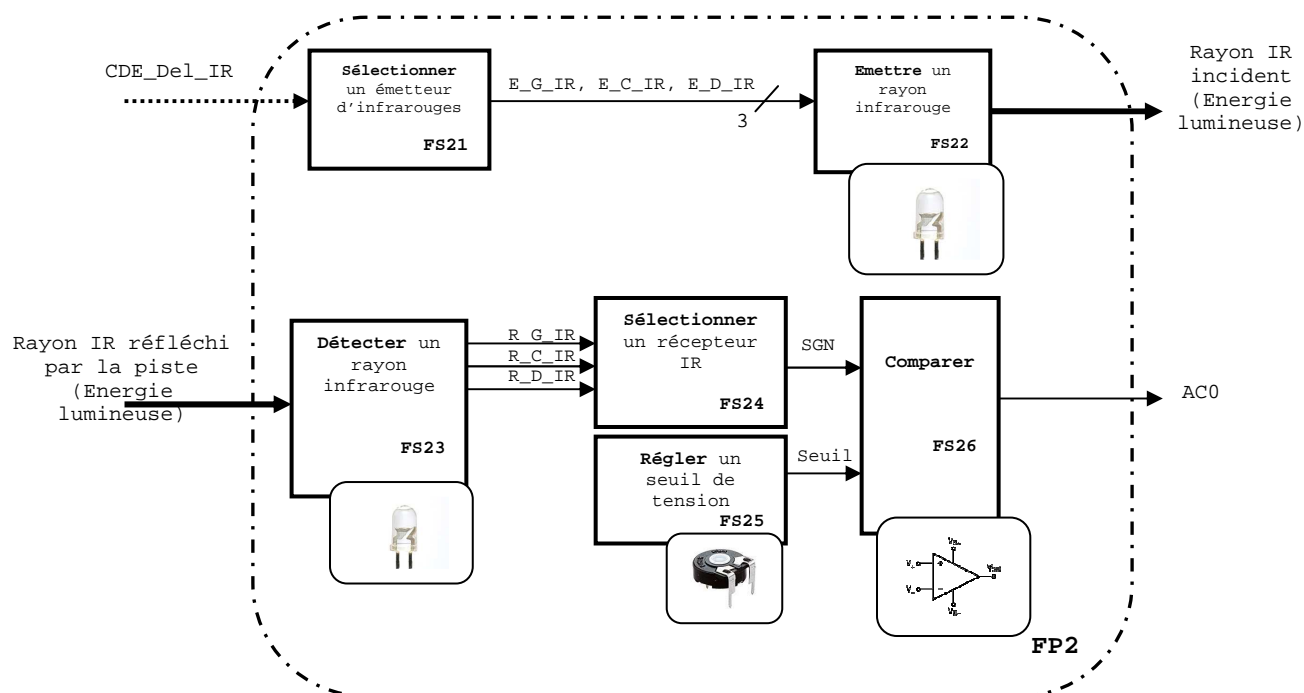
- **Position_Robot :** voir FS11.

Sortie

- **VisuPosition:** variable numérique (octet non signé) représentative de la position du robot par rapport à la ligne. (○ LED éteinte - ● LED éclairée)

| VisuPosition ₍₁₆₎ | Etat des LED « Visu » | Position du robot |
|------------------------------|-----------------------|-------------------|
| 3F | ○ ○ ○ | Hors ligne |
| 1F | ● ○ ○ | Droite+ |
| 2F | ○ ● ○ | Centrel |
| 0F | ● ● ○ | Droite |
| 37 | ○ ○ ● | Gauche+ |
| 17 | ● ○ ● | Centre2 |
| 27 | ○ ● ● | Gauche |
| 07 | ● ● ● | Centre3 |

2.5.2 FP2 « Acquérir la position du robot par rapport à la ligne »



FS21: « Sélectionner un émetteur d'infrarouges »

Rôle : Sélectionner la LED IR (émetteur IR) devant émettre un rayon infrarouge.

Entrée

- `Cde_Del_IR` : voir FS11

Sorties

- `E_G_IR`, `E_C_IR`, `E_D_IR`: Signaux logiques actifs à l'état bas. Ils commandent les émetteurs IR.

| <code>Cde_Del_IR</code> ₍₁₆₎ | <code>E_G_IR</code> | <code>E_C_IR</code> | <code>E_D_IR</code> | Emetteur commandé |
|---|---------------------|---------------------|---------------------|--------------------|
| <code>FF</code> ₍₁₆₎ | 1 | 1 | 1 | Aucun |
| <code>FB</code> ₍₁₆₎ | 0 | 1 | 1 | Emetteur IR Gauche |
| <code>FD</code> ₍₁₆₎ | 1 | 0 | 1 | Emetteur IR centre |
| <code>FE</code> ₍₁₆₎ | 1 | 1 | 0 | Emetteur IR droit |

FS22: « Emettre un rayon infrarouge »

Rôle: Convertir une énergie électrique en radiations non visibles.

Entrées

- `E_G_IR`, `E_C_IR`, `E_D_IR` : voir FS21.

Sorties

- **Rayon IR incident:** radiations non visibles émises vers la piste.

FS23: « Détecter un rayon infrarouge »

Rôle: Transformer un rayon infrarouge en une tension analogique.

Entrée

- **Rayon IR réfléchi :** rayon infrarouge réfléchi par la piste.

Sorties

- R_G_IR, R_C_IR, R_D_IR : tensions analogiques dont l'amplitude est représentative de la position d'un capteur par rapport à la ligne noire.

FS24: « Sélectionner un récepteur IR »

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

Entrées

- R_G_IR, R_C_IR, R_D_IR : voir FS23.

Sortie

- SGN : tension analogique. SGN = R_G_IR ou R_C_IR ou R_D_IR.

FS25: « Régler un seuil de tension »

Rôle: Régler la valeur d'une tension continue afin de différencier l'information « image de la piste blanche » de l'information « image de la ligne noire ».

Sortie

- Seuil : tension continue réglable entre 0 et 6V.

FS26: « Comparer »

Rôle: Transformer la tension analogique, représentative de la « position du capteur par rapport à la ligne » en un signal logique.

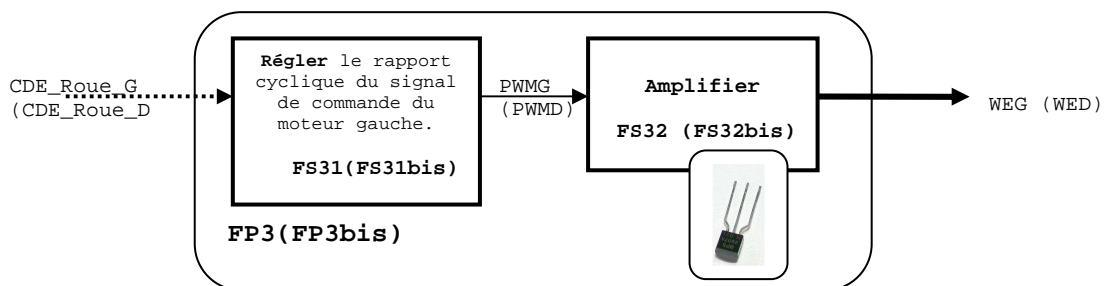
Entrées

- Seuil, SGN : voir FS24 et FS25.

Sortie

- AC0: signal logique représentatif de la position d'un des trois capteurs d'infrarouges par rapport à la ligne (signal interne au microcontrôleur).
AC0 = 1 si le capteur sélectionné est sur la ligne.

2.5.3 FP3(FP3bis) « Distribuer »



FS31(FS31bis): « Régler le rapport cyclique... »

Rôle: Régler le rapport cyclique α du signal de commande du moteur gauche (droit).

Entrée

- CDE_Roue_G (CDE_Roue_D): voir FS12.

Sortie

- PWMG (PWMD): signaux logiques de fréquence fixe et de rapport cyclique variable.

FS32(FS32bis): « Amplifier... »

Rôle: Amplifier en puissance le signal de commande du moteur gauche (droit).

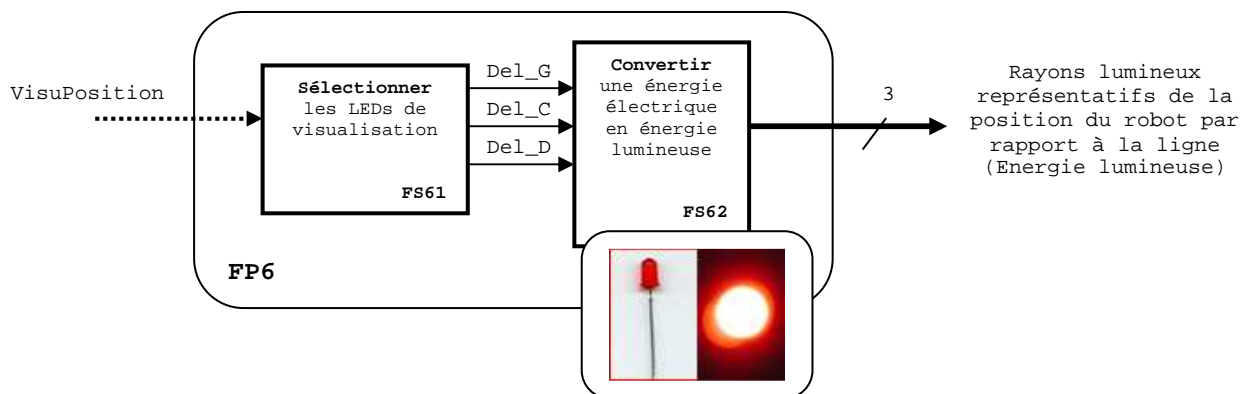
Entrée

- PWMG (PWMD): voir FS31(FS31bis).

Sortie

- WEG (WED) : signal logique de fréquence fixe et de rapport cyclique variable adapté à la puissance demandée par le moteur gauche (droit).

2.5.4 FP6 « Communiquer la position du robot par rapport à la ligne »



FS61 : « Sélectionner les LED de visualisation »

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

Entrée

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

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

Sorties

- Del_G, Del_C, Del_D : signaux logiques actifs à l'état bas.

| VisuPosition ₍₁₆₎ | Signaux logiques de commande des LEDs de visualisation | | | Position du robot |
|------------------------------|--|-------|-------|-------------------|
| | DeL_G | DeL_C | DeL_D | |
| 3F | 1 | 1 | 1 | Hors ligne |
| 1F | 0 | 1 | 1 | Droite+ |
| 2F | 1 | 0 | 1 | Centrel |
| 0F | 0 | 0 | 1 | Droite |
| 37 | 1 | 1 | 0 | Gauche+ |
| 17 | 0 | 1 | 0 | Centre2 |
| 27 | 1 | 0 | 0 | Gauche |
| 07 | 0 | 0 | 0 | Centre3 |

FS62: « Convertir une énergie électrique en énergie lumineuse »

Rôle: Visualiser la position du robot par rapport à la ligne.

Entrées

- Del_G, Del_C, Del_D : voir FS61.

Sorties

- Rayons lumineux.

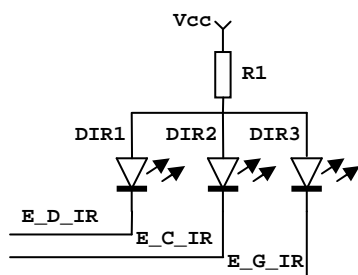
3. Etude structurelle

3.1 Etude structurelle de FP1 « Traiter... »

Les fonctions logicielles associées à FP1 sont décrites dans le programme de l'annexe 12.

3.2 Etude structurelle de FP2 « Acquérir la position du robot par rapport à la ligne »

3.2.1 FS22 « Emettre un rayon infrarouge »



Les émetteurs d'infrarouges DIR1, DIR2 et DIR3 produisent un rayon IR lorsqu'ils sont commandés au niveau logique zéro (schéma ci-contre). La commande des émetteurs est multiplexée.

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.

Des exemples de relevés des signaux E_G_IR , E_C_IR et E_D_IR de commande de DIR1, DIR2 et DIR3 sont donnés en annexe 4.

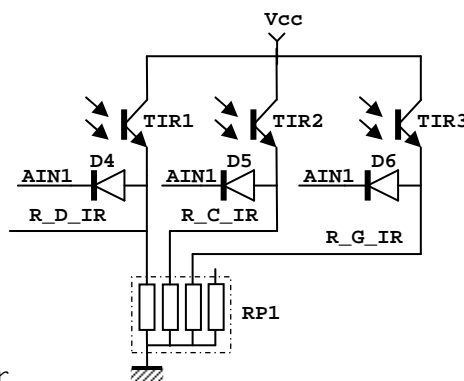
3.2.2 FS23 « Détecter un rayon infrarouge »

Les récepteurs IR (TIR1, TIR2, TIR3) associés à chaque émetteurs IR captent les rayons 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_Gauche (DIR3, TIR3) est sur la piste blanche, le rayon réfléchi atteint le phototransistor TIR3.

Celui-ci se « ferme », la diode D6 devient passante.



3.2.3 FS24 « Sélectionner un récepteurs d'infrarouges »

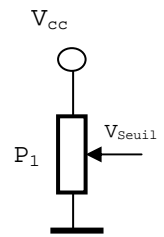
Les diodes D4, D5 et D6 réalisent un **opérateur logique OU** dit : « câblé ». Le signal AIN1 correspond à la sortie de cet opérateur. Il est représentatif de la position d'un capteur IR par rapport à la ligne noire. L'amplitude du signal AIN1 dépend de la position des capteurs IR par rapport à la ligne.

Des exemples de relevés du signal $SGN(AIN1)$ délivré par les récepteurs d'infrarouges sont donnés en annexe 5. Comme vous pouvez le constater sur les chronogrammes, SGN est un signal analogique.

Pour que le microcontrôleur (μC) traite ce signal, il est nécessaire de le « conditionner » (transformer) en un signal logique. Ceci est réalisé par une structure interne au μC appelée « **comparateur analogique** ». (FS26)

3.2.4 FS25 « Régler un seuil de tension » (voir annexe 1)

Le potentiel V_{Seuil} de la broche AIN0 est ajusté entre 0 et V_{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.

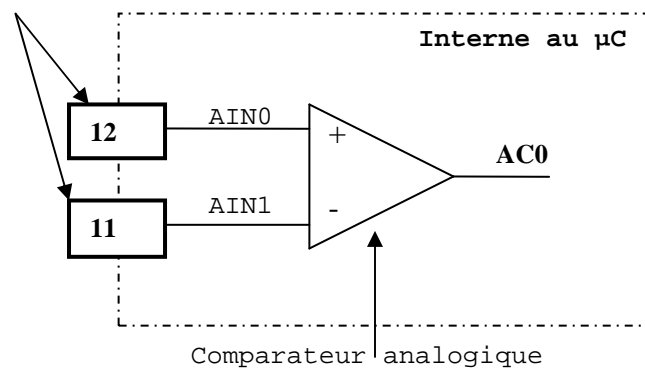


3.2.5 FS26 « Comparer »

La tension analogique V_{SGN} placée sur l'entrée AIN1 du microcontrôleur est convertie (conditionnée) en un signal logique **AC0** par un **comparateur analogique** intégré au microcontrôleur. Le comportement de cette structure est décrit ci-dessous.

```
si (le potentiel sur l'entrée AIN0
est supérieur au potentiel sur
l'entrée AIN1)
    alors AC0 = « 1 » logique ;
    sinon AC0 = « 0 » logique ;
finsi
```

Broches du μC ATINY26



AC0 est l'image de la position d'un capteur IR par rapport à la ligne.

AC0 est utilisé par la fonction logicielle FS11 « 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.

3.3 Etude structurelle de FP3(FP3bis) « Distribuer »

3.3.1 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 fréquence de rotation 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 tension présente à ses bornes en le commandant par un signal dit « **MLI*** (ou PWM) ».

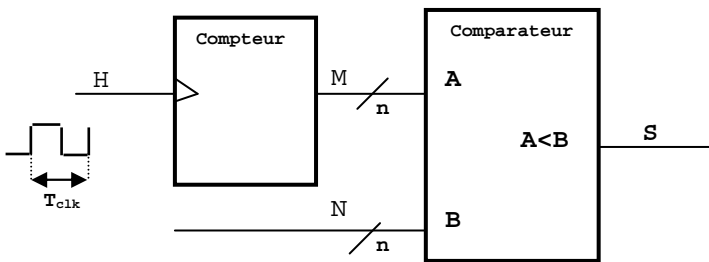
Génération d'un signal **Modulé en Largeur d'Impulsion** (principe)

Un signal modulé en largeur d'impulsion peut être obtenu à partir d'un signal périodique H de fréquence fixe $F_{\text{clk}} = 1/T_{\text{clk}}$. 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 numérique constant $N(t)$ (codé sur n bits) à un comparateur numérique, on obtient un signal binaire $S(t)$ de période $T = (2^n - 1) \cdot T_{\text{clk}}$ 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 α par S_{maxi} .

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

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 .

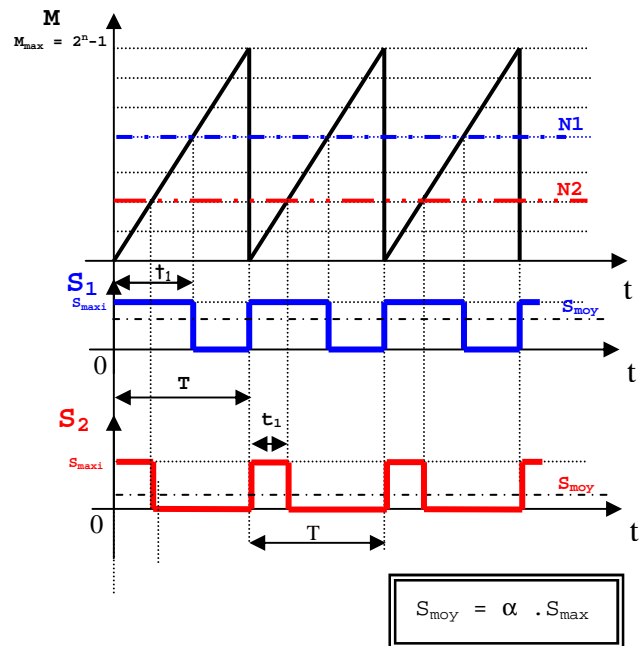


M = valeur numérique variable
($0 \leq M \leq 2^n - 1$)

N = valeur numérique constante

Comparateur : si ($A < B$) alors $S = S_{\max}$
sinon $S = 0$

H : signal périodique de fréquence F_{clk}

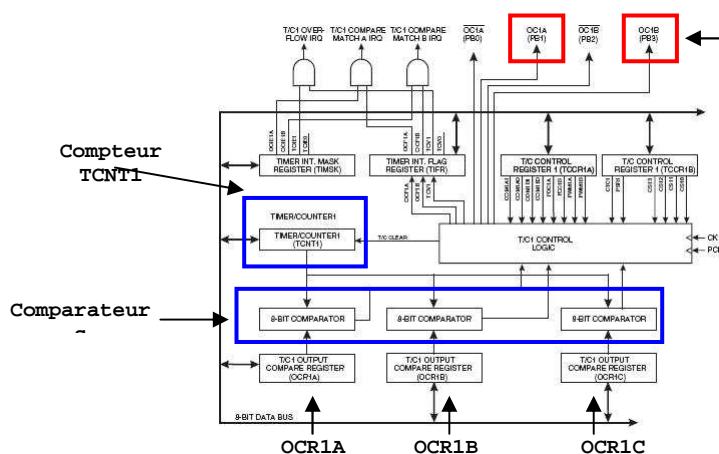


Dans les microcontrôleurs, les signaux modulés en largeur d'impulsion sont générés par une structure appelée **TIMER**. Celle-ci répond au principe développé ci-dessus.

3.3.2 Génération d'un signal M.L.I avec le microcontrôleur ATINY26

Le **Timer1** de l'ATINY26 permet de générer quatre signaux logiques modulés en largeur d'impulsion (OC1A, /OC1A, OC1B, /OC1B). (/ signifie complément logique)

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.

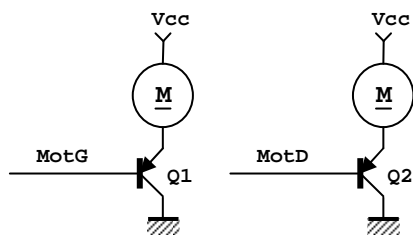
3.3.3 Identification des structures associées à FS31(FS31bis)

FS31 est constitué de TCNT1, OCR1A et OCR1C. FS31bis est constitué de TCNT1, OCR1B et OCR1C. Les signaux **MotD** et **MotG** de commande des moteurs du robot sont issus des broches PB1(OC1A) et PB3(OC1B) du microcontrôleur.

| PDIP/SOIC | |
|------------------------|----|
| (MISO/DI/SDA/OC1A) PB0 | 1 |
| (MISO/DO/OC1A) PB1 | 2 |
| (SCK/SCL/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) | 20 |
| PA1 (ADC1) | 19 |
| PA2 (ADC2) | 18 |
| PA3 (AREF) | 17 |
| AVCC | 15 |
| GND | 16 |
| PA4 (ADC3) | 14 |
| PA5 (ADC4) | 13 |
| PA6 (ADC5/AIN0) | 12 |
| PA7 (ADC6/AIN1) | 11 |

3.3.4 Etude structurelle de FS32(FP32bis) « Amplifier »

MotG et MotD, commandent les bases des transistors PNP Q1 et Q2.



Les moteurs sont reliés au circuit des transistors Q1 et Q2 par l'intermédiaire des connecteurs Roue Droite et Roue Gauche. (voir annexe 10)

Remarque : Les transistors Q1 et Q2 inversent le niveau logique des signaux de commandes des moteurs.

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

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

3.4 Etude structurelle de FP4(FP4bis) « Convertir une énergie électrique en énergie mécanique »

Caractéristiques des moteurs

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

3.5 Etude structurelle de FP5(FP5bis) « Transmettre l'énergie mécanique du moteur aux roues »

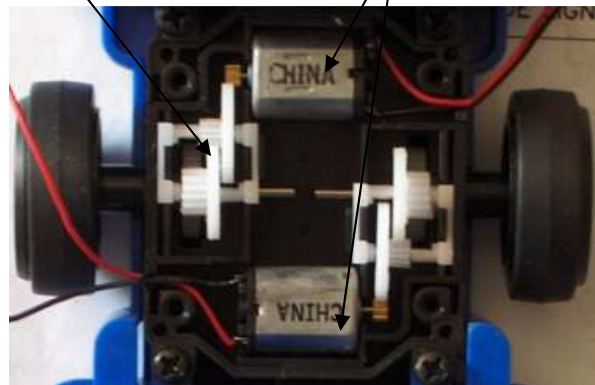
Caractéristique du réducteur

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

Roues : $\varnothing 30mm$

Réducteur

Moteurs à C.C.



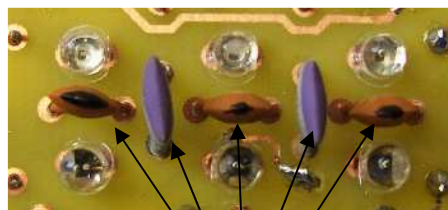
Vue de dessus

Annexe 1 : Description de la carte MrLineTiny

Vu de dessous

Emetteurs IR (E_IR) →

Récepteurs IR (R_IR) →



1 Emetteur IR + 1 Récepteur IR = 1 capteur IR

IR : Infrarouge

Ecrans

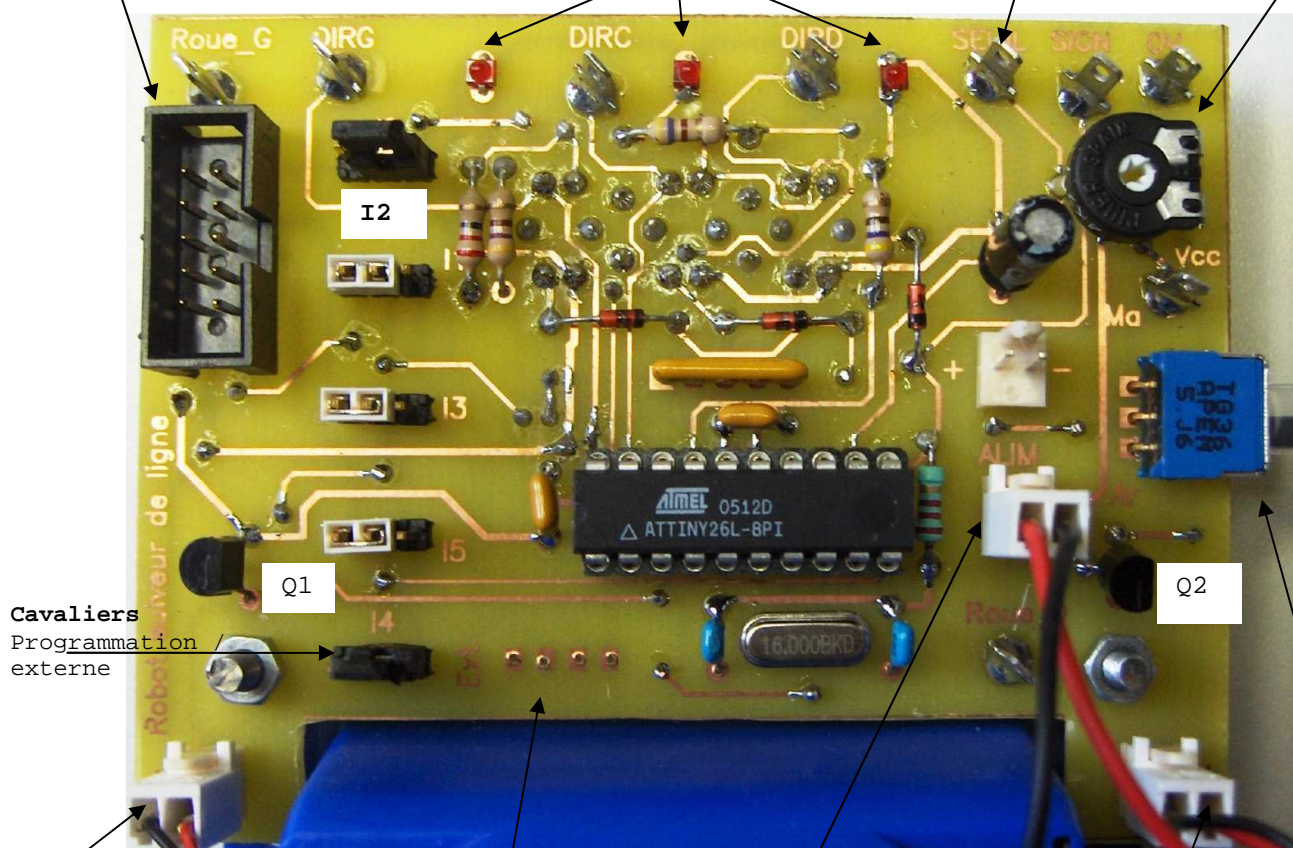
Vu de dessus

Connecteur HE10-10 pour le câble de **programmation**

LED « **Visu** » position robot

Point test

P1 : Réglage du seuil de noir



Cavaliers
Programmation
externe

Connecteur moteur
roue droite [JP2]

Emplacement
Connecteur
Externe [JP1]

Connecteurs **alimentation**

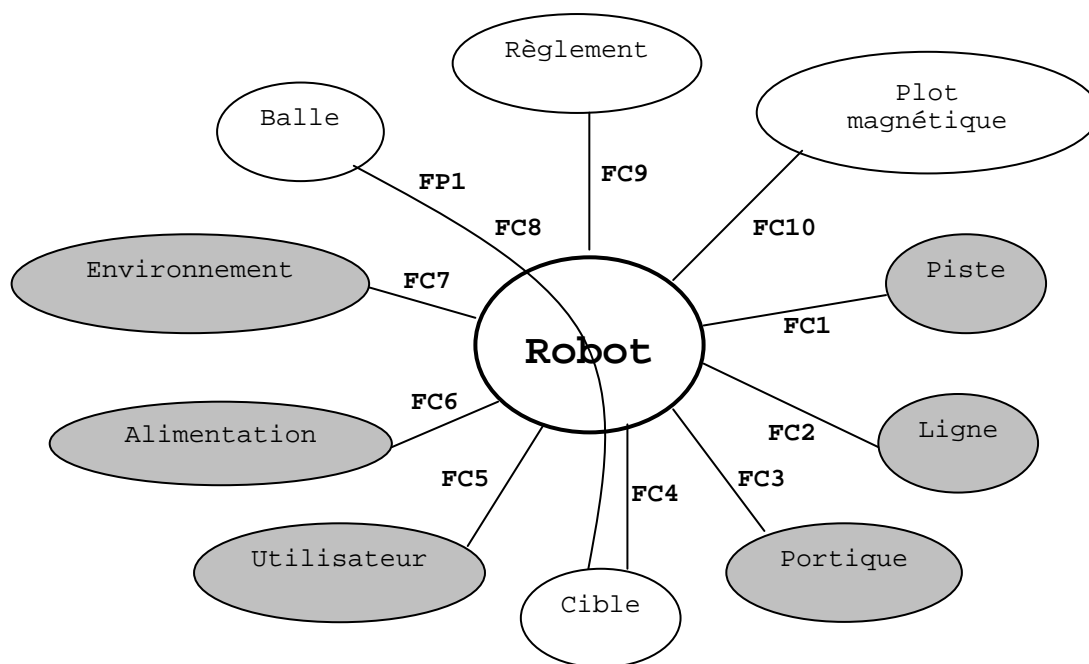
Marche/Arrêt

Connecteur moteur
roue gauche [JP3]

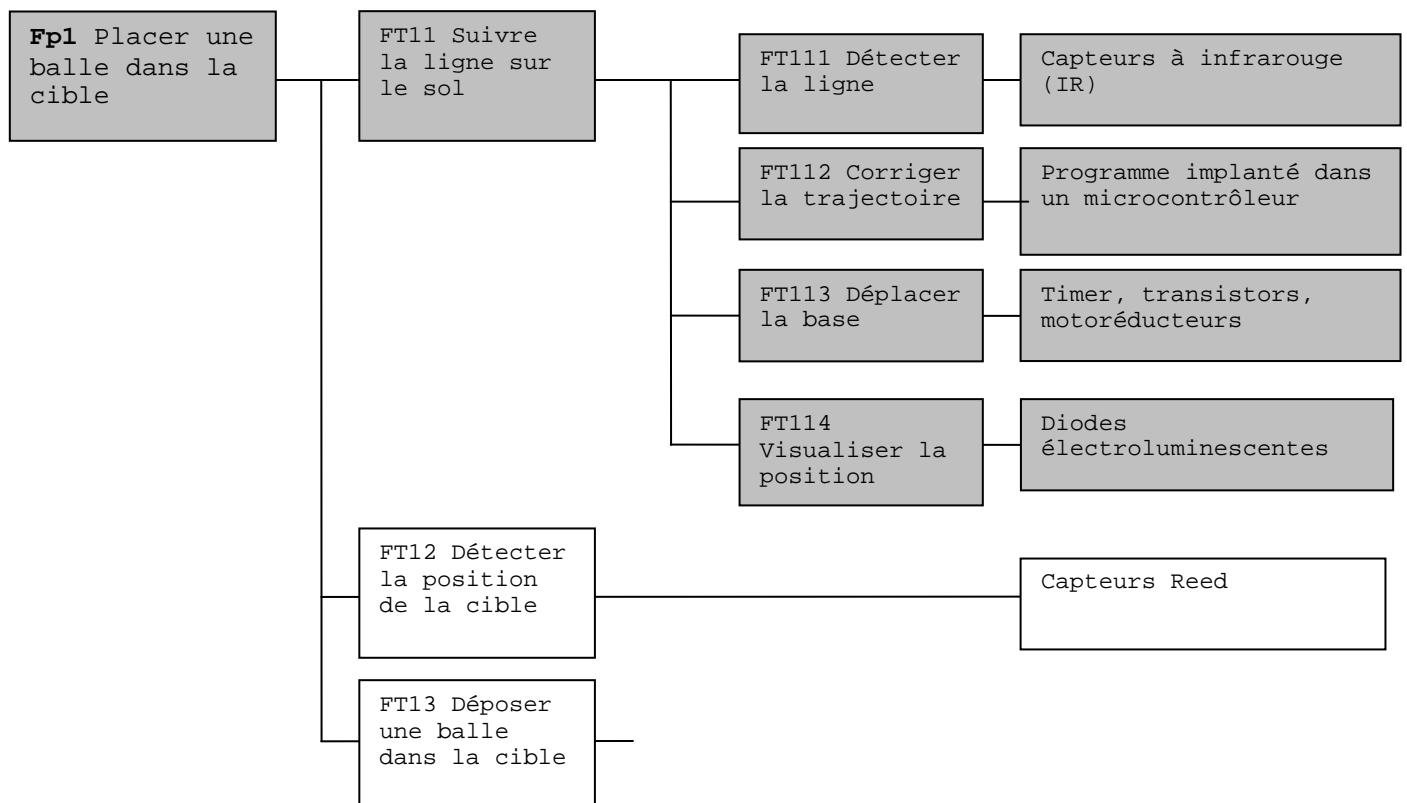
Position des cavaliers

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

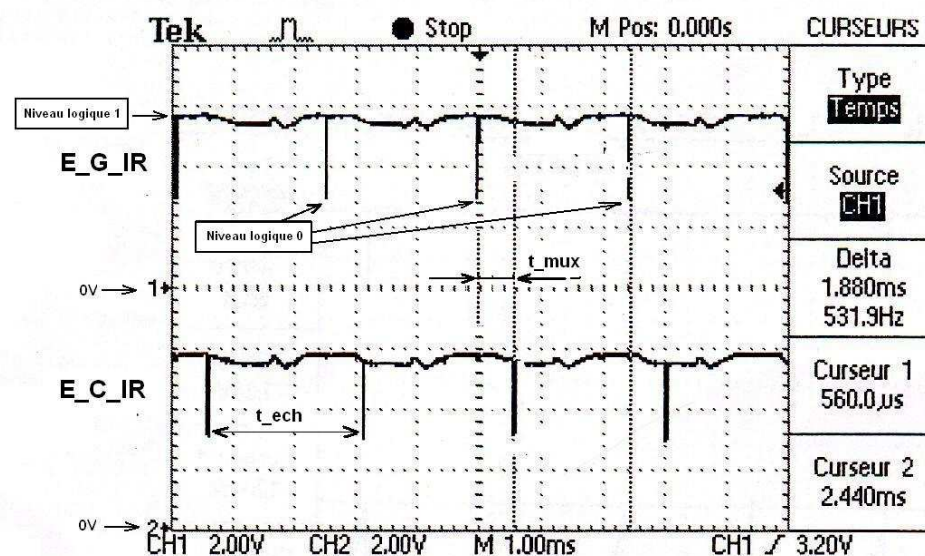
Annexe 2 : Diagramme pieuvre



Annexe 3 : Diagramme fast partiel

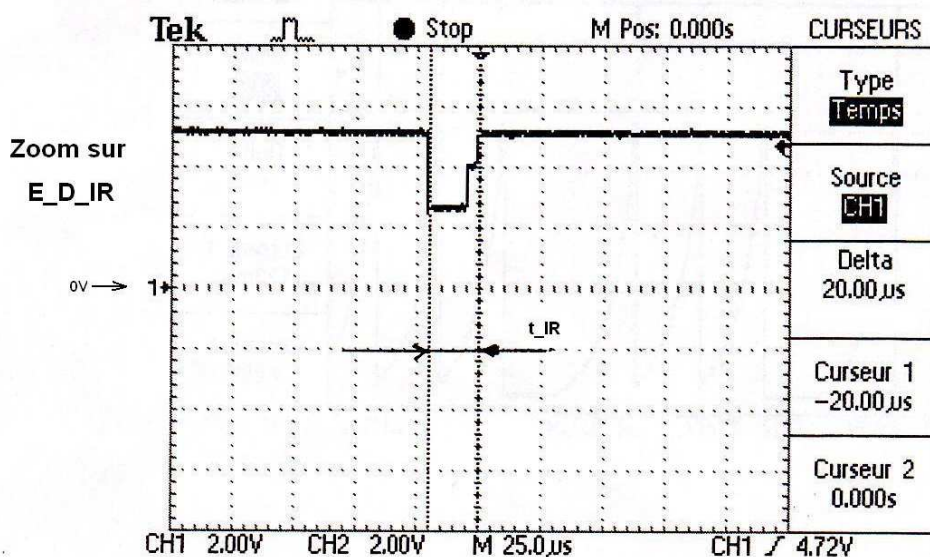
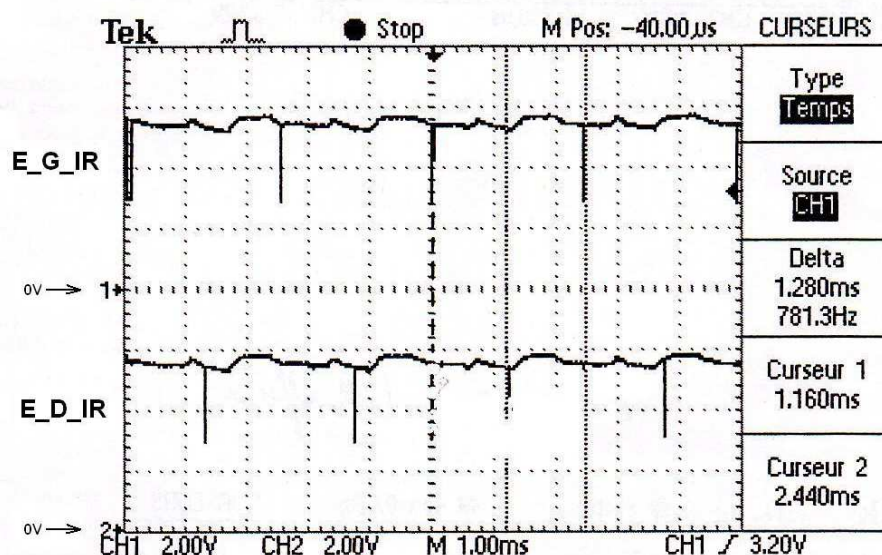


Annexe 4 : Exemples de relevés des signaux E_G_IR, E_C_IR et E_D_IR



$$t_{ech} = 3 \cdot (t_{IR} + t_{mux}) + t_{delay}$$

Remarque : Exemple de mesures. Les valeurs de t_{ech} et t_{mux} sont à régler en fonction des capteurs IR utilisés.



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

CAPTEURS SUR LA LIGNE (IR ABSORBE)

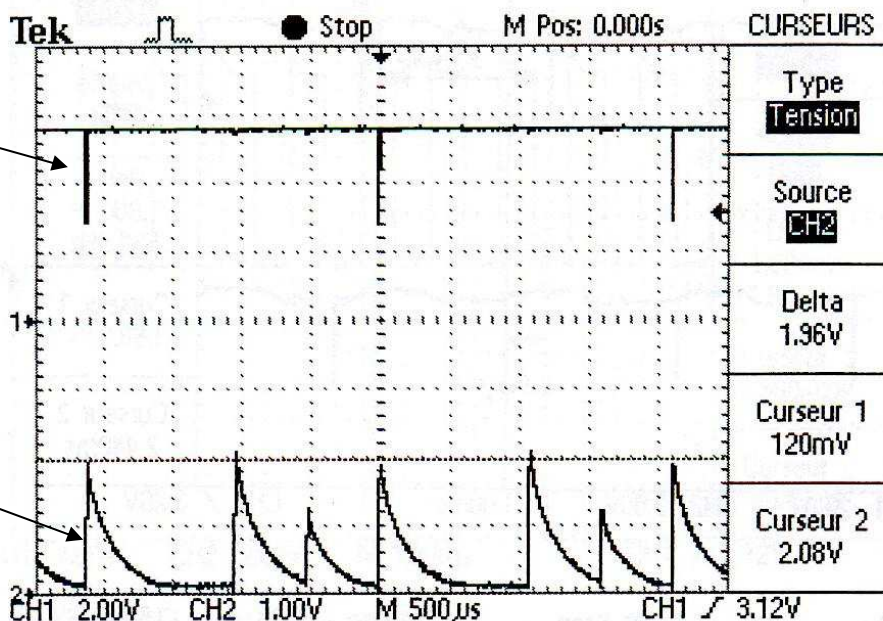
Commande de la diode IR droite (**DIR3**) par un niveau logique bas

↓
Émission d'un rayon infrarouge vers la piste.

E_D_IR

Avec la piste et les capteurs utilisés, le rayon infrarouge (en grande partie **absorbé** par la bande noire) produit une tension de **2V** (au maximum) à la sortie du phototransistor **TIR3**.

SGN(AIN1)



Remarque : Exemple de mesures. Les résultats dépendent des capteurs IR utilisés.

CAPTEUR HORS LIGNE (IR REFLECHI)

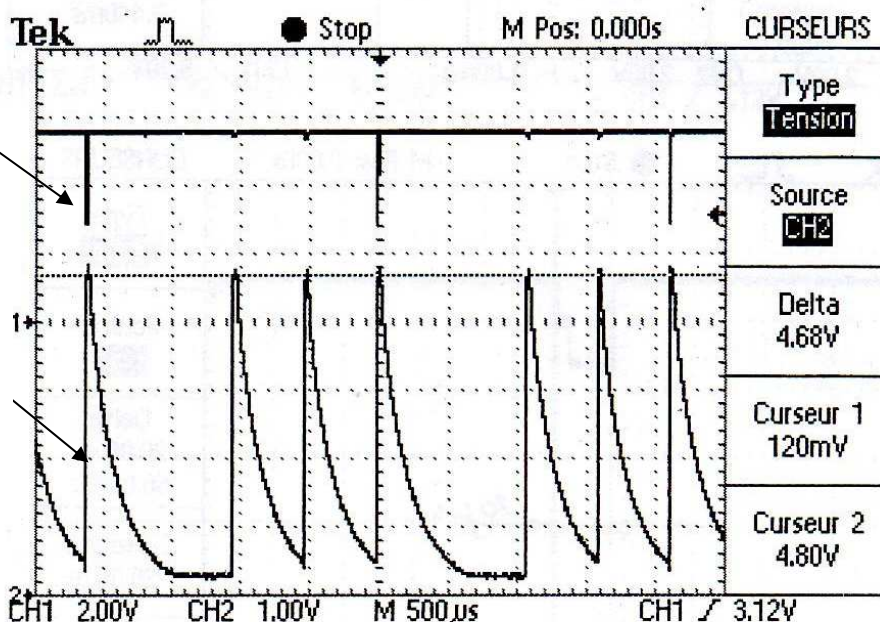
Commande de la diode IR droite (**DIR3**) par un niveau logique bas

↓
Emission d'un rayon infrarouge vers la piste.

E_D_IR

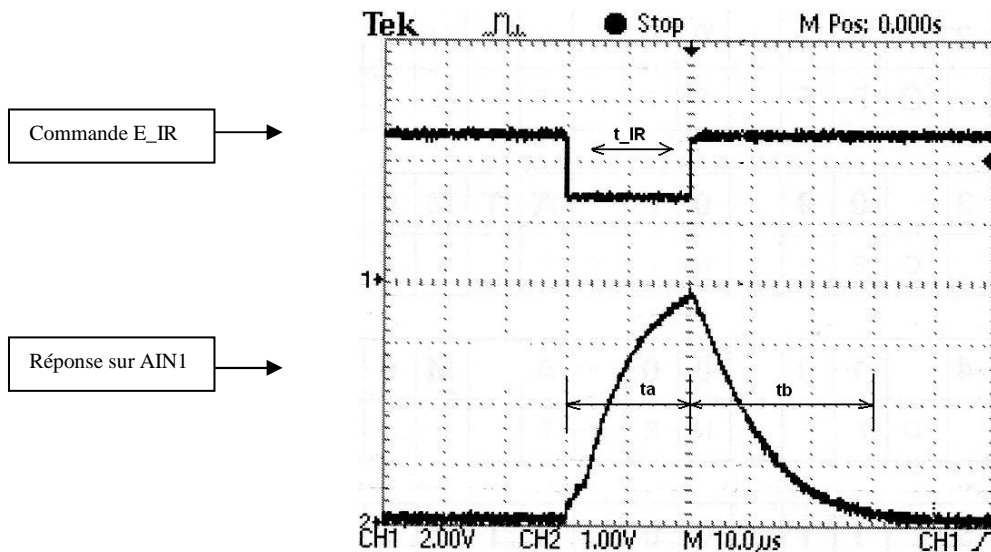
Avec la piste et les capteurs utilisés, le rayon infrarouge **réfléchi** par la piste produit une tension de **5V** à la sortie du phototransistor **TIR3**.

SGN(AIN1)



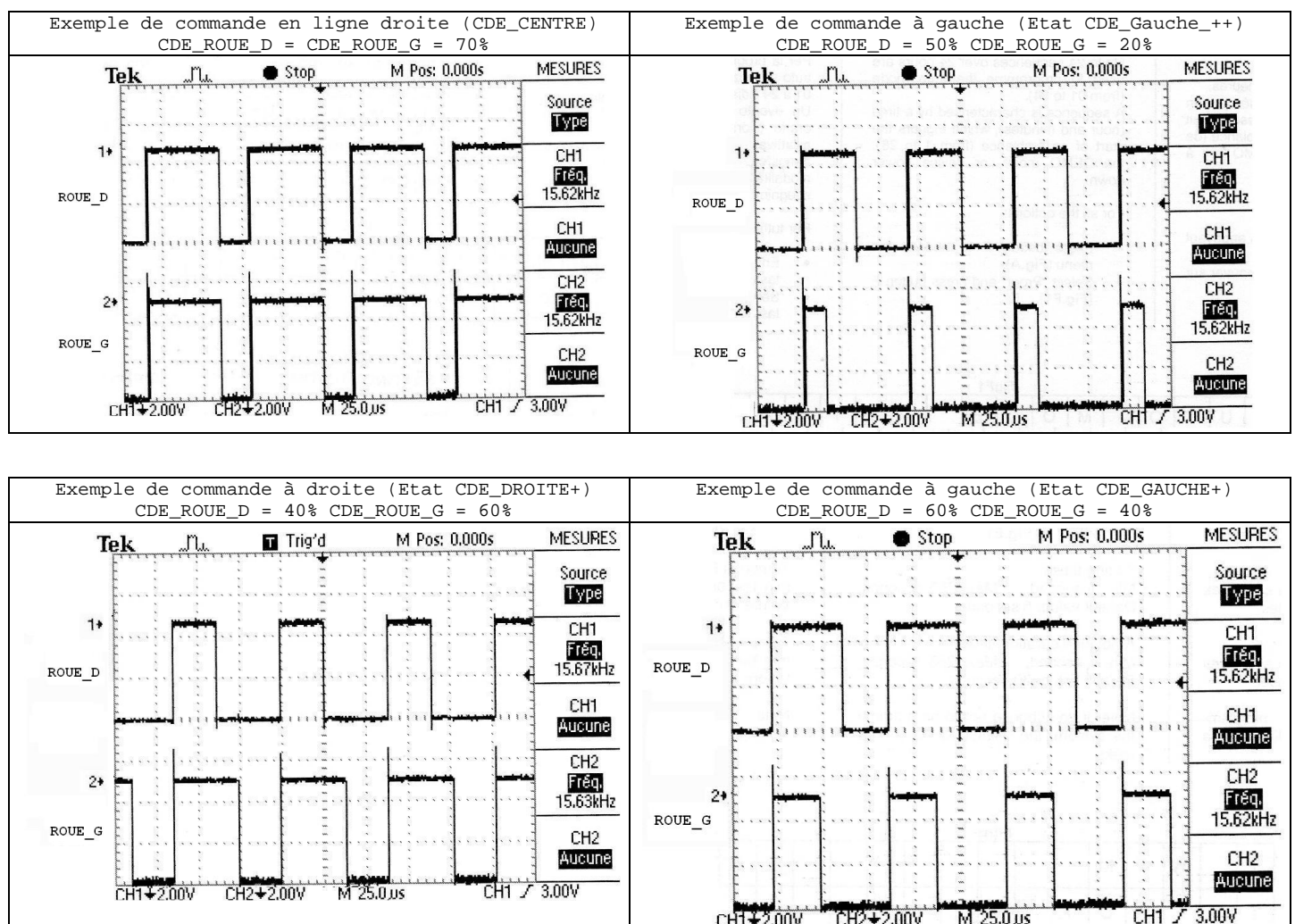
Remarque : 1-> et 2-> correspondent au 0V des voies de l'oscilloscope

Zoom sur le signal SGN(AIN1) délivré par un des récepteurs d'infrarouges (R_IR)



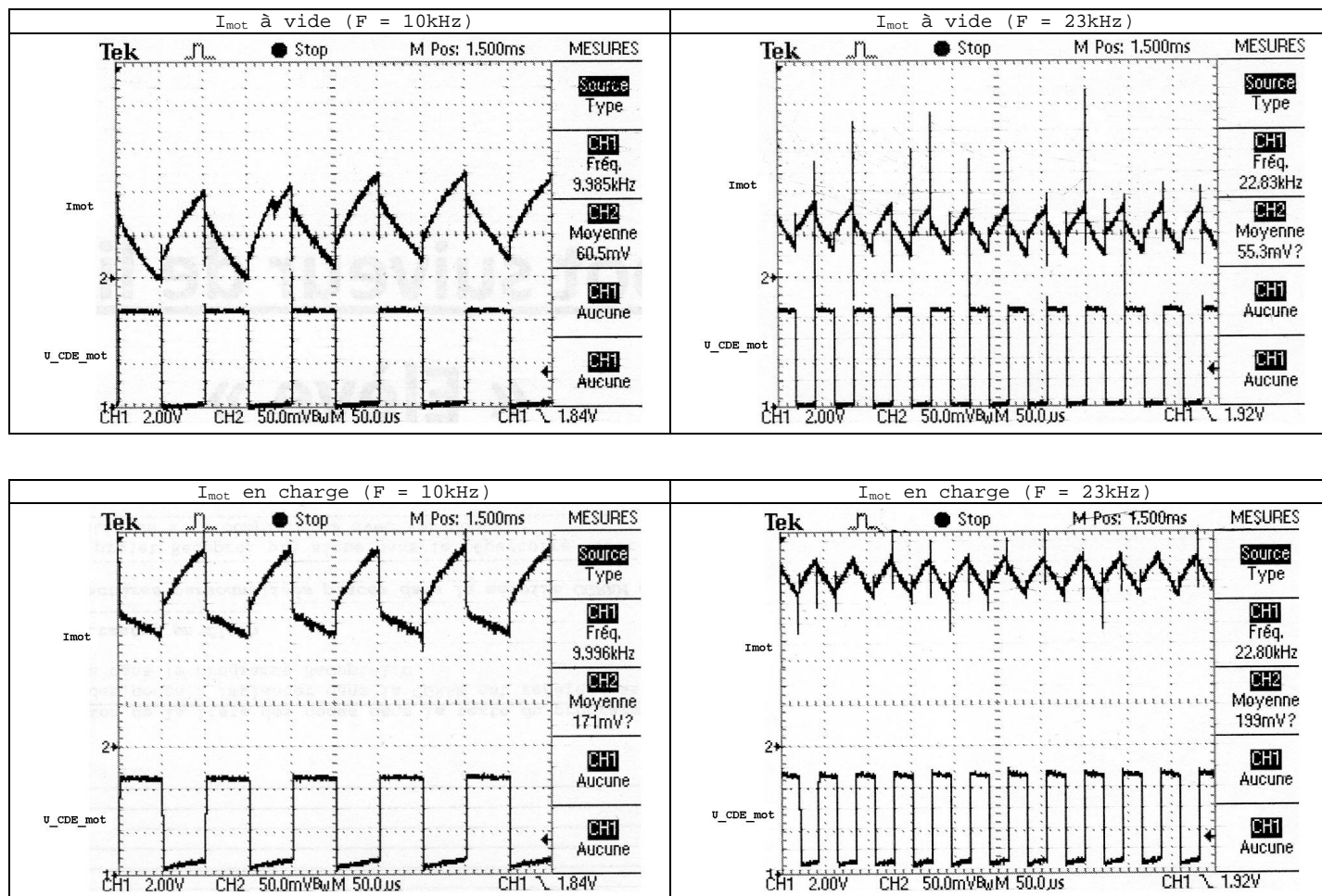
Annexe 6 : Exemples de signaux MLI(PWM) pour la commande des moteurs.

Remarque concernant la méthode de mesure : La « forme » des signaux présents aux bornes des moteurs est obtenue en mesurant les signaux aux points test ROUE_D et ROUE_G et en inversant l'affichage de ces signaux au niveau de l'oscilloscope (menu CHx :Inverser = On).



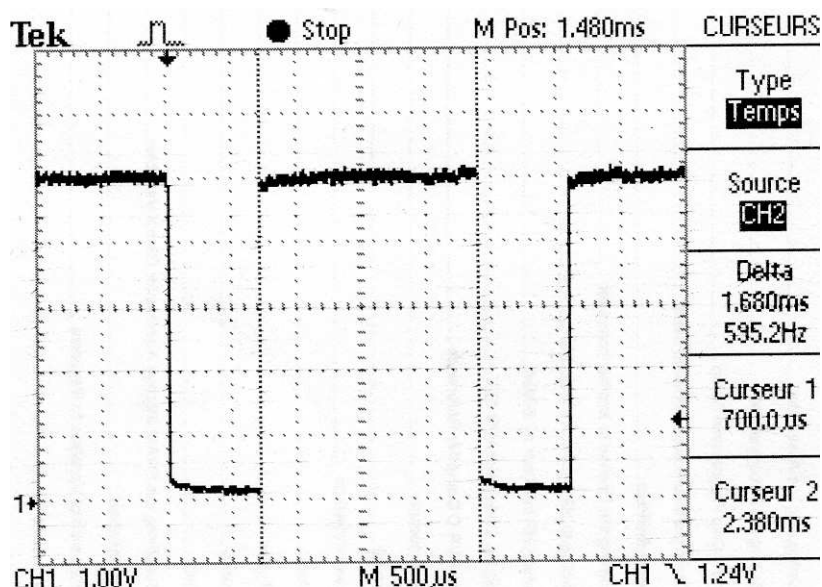
ATTENTION : 1-> et 2-> correspondent au 5V ! (signaux inversés)

Annexe 7 : Exemples de relevés du courants moteur I_{mot} mesurés avec une résistance de 1Ω placée en série avec le moteur droit

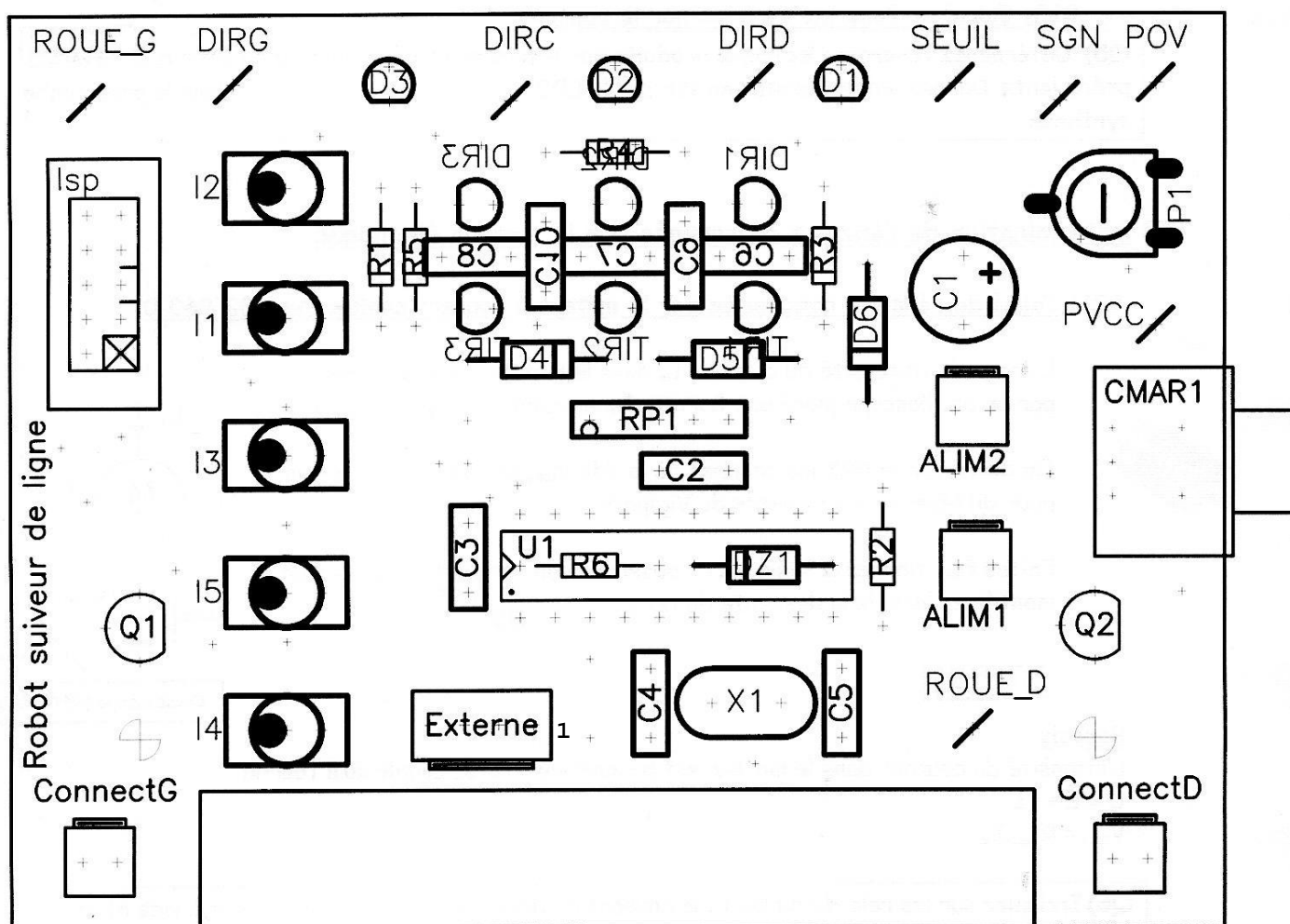


Remarque : 1-> et 2-> correspondent au 0V des voies de l'oscilloscope

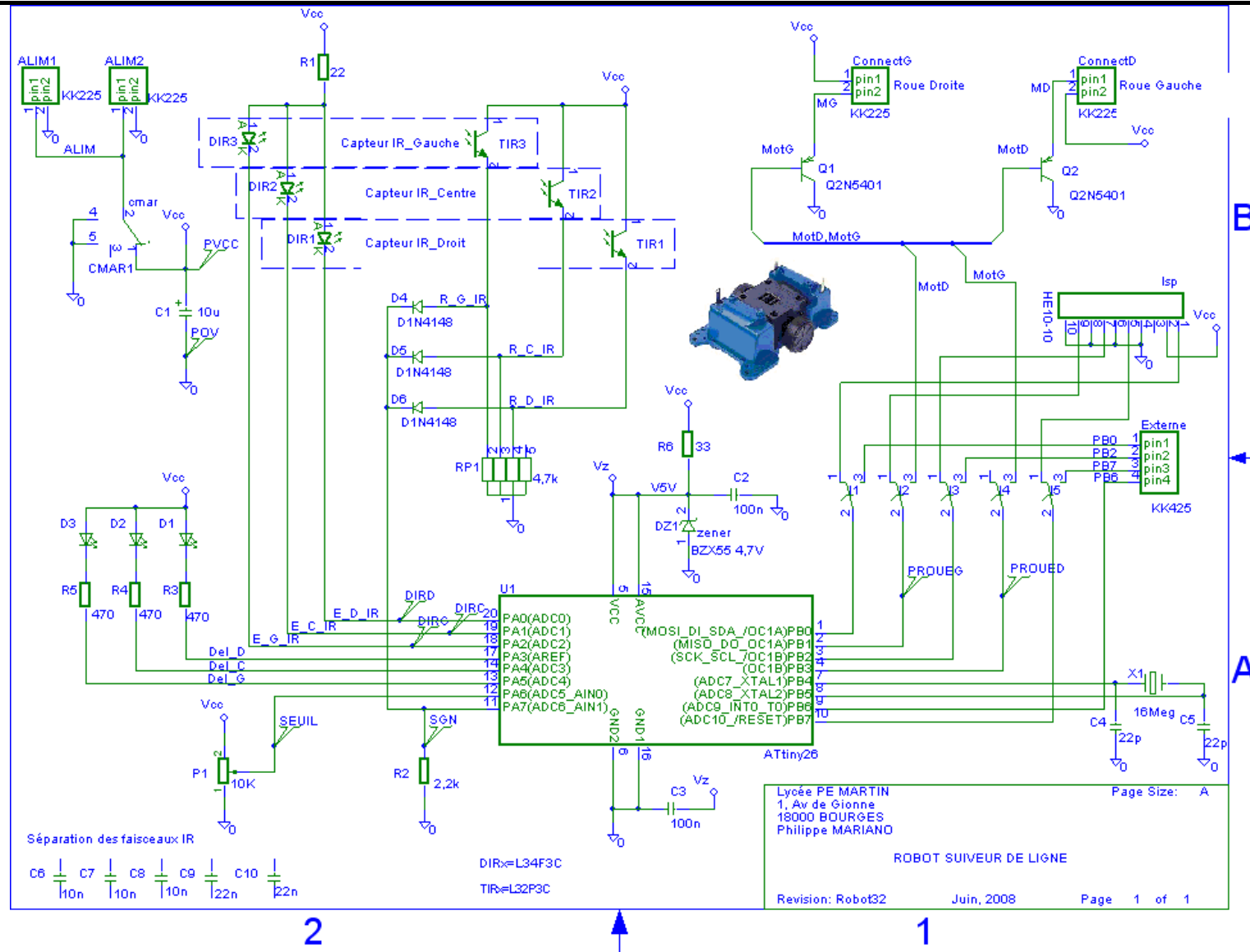
Annexe 8 : Exemple de relevé du signal de commande d'une LED « Visu »



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



Annexe 10 : Schéma structurel de la carte M^c LineTiny



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

| | Q | Mfg ID | Attribute | RefDes | Fournisseur | Désignation | Prix HT (2008) | Remarques |
|----|---|----------|--------------|---|------------------------|----------------|----------------|--|
| 1 | 1 | | | | LEXTRONIC | MRM-010 | 30,35 | Base motorisé |
| 2 | 1 | Atiny26L | DIP20 | U1 | Radiospares | 472-6251 | 2,59 | Atiny26L |
| 3 | 1 | | | | Electronique Diffusion | COIC120E | 0,38 | Support DIL 20b |
| 4 | 3 | TIR | TRIR | TIR1, TIR2, TIR3 | | OPKBL32P3C | 0,21 | Kingbright L-32P3C Ø3mm |
| 5 | 3 | DELIR | DELIR | DIR1, DIR2, DIR3 | | OPKBL34F3C | 0,21 | Kingbright L-34F3C Ø3mm λ = 940nm |
| 6 | 2 | 2N5401 | TO-92 | Q1, Q2 | | 2N5401 | 0,13 | PNP commut. |
| 7 | 3 | 1N4148 | DO-35 | D4, D5, D6 | | 1N4148 | 0,07 | |
| 8 | 3 | LED | LED | D1, D2, D3 | Radiospares | 193-4746 | 3,15 | HLMP-Q102 Rouge (prix pour 5) 160mcd(20mA) |
| 9 | 1 | Zener | DO-35 | DZ1 | Electronique Diffusion | DIZE4V70W4 | 0,13 | BZX55 4,7V (250mW) |
| 10 | 1 | Quartz | HC-25U 16Meg | X1 | | QUQZ16MHZC4 | 0,84 | Quartz HC49S |
| 11 | 1 | C+ | CPR2-3 10u | C1 | Electronique Diffusion | CDR8516V10MF5 | 0,13 | Chimique Radial 16V |
| 12 | 2 | c | CK05 100n | C2, C3 | | CDMX50V100NF-5 | 0,25 | COND.MULTI. X7R 50V 100NF PAS 5.08 (prix pour 5) |
| 13 | 2 | c | CK05 22p | C4, C5 | | CDMN50V22PF-5 | 0,21 | COND.MULTI. NPO 50V 22PF PAS 5.08 (prix pour 5) |
| 14 | 3 | c | CK05 10n | C6, C7, C8 | | CDC50810NF | 0,07 | COND.CERAM.PAS 5.08 10 NF |
| 15 | 2 | c | CK05 22n | C9, C10 | | | | Rectangulaire |
| 16 | 1 | r | RC05 22 | R1 | | | | 1/4W |
| 17 | 1 | r | RC05 2,2k | R2 | | | | |
| 18 | 3 | r | RC05 470 | R3,R4, R5 | | | | |
| 19 | 1 | r | RC05 33 | R6 | | | | |
| 20 | 1 | Px1 | AJUST_H 10k | P1 | Electronique Diffusion | POACAH10K | 0,21 | POT.AJUST.CAPOTE HORIZ.10 KOHMS |
| 21 | 1 | R_SIL4+1 | Reseau5 4,7K | RP1 | | RESIL514k7 | 0,33 | RESEAU SIL5+1 4.7KR OHMS 6 BROCHES |
| 22 | 1 | CMAR | CMAR | CMAR1 | Radiospares | 190-0608 | 3,15 | Commutateur 2 positions |
| 23 | 1 | HE10-10 | HE10-10 | Isp | Electronique Diffusion | COH10M10 | 0,42 | Mâle droit |
| 24 | 5 | inver | Inver_1 | I1, I2, I3, I4, I5 | | CONSH36SBS2TR | 1,67 | Barrette sécable droite 1 rangée 36pt |
| 25 | 2 | | | | | COW8013T50N-L | 0,17 | Cavaliers pour barrette sécable avec languette |
| 26 | 3 | | | | | COW8013T50N | 0,13 | Cavalier pour barrette sécable noir |
| 27 | 4 | KK225 | KK225 | ALIM1, ALIM2, MotD, MotG | | COKK254M-2 | 0,17 | Embase mâle droite KK2 pas 2,54 avec détrompeur |
| 28 | 4 | | | | | COKK254M-4 | 0,17 | Connecteur KK2 pas 2,54 femelle |
| 29 | 1 | KK425 | KK425 | Externe | | COKK254F-2 | 0,38 | BOÎTIER PRE-CABLE 2 CONTACTS |
| 30 | 1 | | | | | COKK254F-4 | 0,54 | Connecteur KK4 pas 2,54 femelle |
| 31 | 9 | PTEST | PTEST | DIRD, DIRC, DIRG, PMotD, PMotG, POV, PVcc, Seuil, SGN | | | | |
| 32 | 2 | | | | | | | Entretoise l=5mm |
| 33 | 2 | | | | | | | Vis 3mm |
| 34 | 2 | | | | | | | Ecrou 3mm |

Annexe 12 : Programme C implanté dans le microcontrôleur de la carte « 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 : 15/09/2008
Author : Philippe Mariano
Company : LYCEE Pierre Emile MARTIN
1, Av de Gionne 18000 Bourges
Comments: Suivi de ligne (fichier prof)

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 l'état actif du signal de 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



Philippe Mariano

PPE ROBOT SUIVEUR DE LIGNE

Dossier Technique ROBOT MrLineTiny

29

```
// Commande des moteurs
//-----
#define CDE_ROUE_G      OCR1A      // roue gauche (moteur droit)
#define CDE_ROUE_D      OCR1B      // roue droite (moteur gauche)

// Valeurs représentatives du rapport cyclique du signal
// de commande des moteurs (OCR1A et OCR1B 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 LED Visu et IR au port A de l'ATINY26
//-----
// LED_D      reliée à PA3 (D1)      E_D_IR PA0  (DIR1)
// LED_C      reliée à PA4 (D2)      E_C_IR PA1  (DIR2)
// LED_G      reliée à PA5 (D3)      E_G_IR PA2  (DIR3)

// Mots de commande des LED Visu  (- LED éteinte      * LED éclairée)
// -----Etat des LED de visualisation-----
// LED_G | LED_C | LED_D  VisuPosition(16)      Remarque
//-----
#define Visu_Hors_ligne    0b00111111 // off | off | off      3F      - - -
#define Visu_Droite_Plus   0b00011111 // on  | off | off      1F      * - -
#define Visu_Centre1      0b00101111 // off | on  | off      2F      - * -
#define Visu_Droite       0b00001111 // on  | on  | off      0F      * * -
#define Visu_Gauche_Plus   0b00110111 // off | off | on       37      - - *
#define Visu_Centre2       0b00010111 // on  | off | on       17      * - *
#define Visu_Gauche        0b00100111 // off | on  | on       27      - * *
#define Visu_Centre3       0b00000111 // on  | on  | on       07      * * *

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

```

// Variables globales
//-----
// type          nom          Commentaires
//-----
// Énumération des 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;

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

//-----
// Début du programme principal
//-----
void main(void)
{

// Variables locales
//-----
// type          nom          Commentaires
//-----
unsigned char i;           // Compteur de boucles à usage général
unsigned char Cde_Del_IR;  // Permet de multiplexer la commande des LED IR
unsigned char VisuPosition; // Permet d'afficher la position du robot sur 3 LED rouge

// 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 (fréquence d'entrée du timer FQ/4)
// 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 LED

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

    /*-----
    // Début de FS11 "Calculer la position du robot par rapport à la ligne"
    //-----
    Rôle : Calculer une valeur numérique permettant de sélectionner la LED IR (émetteur IR) devant émettre
    un faisceau lumineux infrarouge et calculer la valeur numérique représentative de la position du robot
    par rapport à la ligne.

```

Entrées : **AC0** bit Associé à la sortie du comparateur analogique (interne à l'ATINY26).
 Représentatif de la position d'un capteur IR par rapport la ligne.

Sorties: **Cde_Del_IR** octet Permet de multiplexer la commande des LED IR
 Comme Cde_Del_IR est transféré dans le port A, on donne
 ci-dessous le détail des bits de commande des LED IR.

| Emetteur IR connecté au bit | | | Remarque |
|-----------------------------|---|-----|--------------------|
| E_D_IR | " | PA0 | Emetteur IR droit |
| E_C_IR | " | PA1 | Emetteur IR centre |
| E_G_IR | " | PA2 | Emetteur IR gauche |

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_G | IR_C | IR_D  =>      Position_Robot(10)
non  | non  | non   =>      0          // Hors ligne
non  | non  | oui   =>      1          // Gauche+
non  | oui  | non   =>      2          // Centre1 (ligne noire de largeur égale à celle d'un capteur)
non  | oui  | oui   =>      3          // Gauche
oui  | non  | non   =>      4          // Droite+
oui  | non  | oui   =>      5          // Centre2 (ligne blanche entourée de deux noires)
oui  | oui  | non   =>      6          // Droite
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 (LED 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 tension
                                // sur l'entrée AIN1 (signal de réception IR) avant une
                                // nouvelle mesure (A ajuster en fonction des capteurs)
}

// Fin de FS11 "Calculer la position du robot par rapport à la ligne"
//-----
```

```

/*-----
Début de FS13  " Choisir la valeur à afficher  "
//-----
Rôle : Choisir la valeur numérique nécessaire à la commande des LED rouge pour que leur éclairage soit
représentatif de la position du robot par rapport à la ligne noire.

Entrée:      Position_Robot      octet      (voir ci-dessus)

Sortie:      VisuPosition        octet      Comme VisuPosition est transféré dans le port A, on donne
ci-dessous le détail des bits de commande des LED rouge
-----
LED de visualisation connectée au bit
-----
LED_D      "      PA3
LED_C      "      PA4
LED_G      "      PA5
Remarque : D pour droite, C pour centre, G pour gauche
-----*/

switch (Position_Robot)
{
    case 0: VisuPosition = Visu_Hors_ligne; break;
    case 1: VisuPosition = Visu_Gauche_Plus; break;
    case 2: VisuPosition = Visu_Centrel; break;
    case 3: VisuPosition = Visu_Gauche; break;
    case 4: VisuPosition = Visu_Droite_Plus; break;
    case 5: VisuPosition = Visu_Centre2; break;
    case 6: VisuPosition = Visu_Droite; break;
    case 7: VisuPosition = Visu_Centre3; break;
}

PORTA = VisuPosition;

// Fin de FS13  " Choisir la valeur à afficher  "
//-----

```

```

/*-----
Début de FS12 " Choisir les valeurs des rapports cycliques "
//-----
Rôle: choisir la valeur du rapport cyclique du signal de commande des moteurs du robot en fonction de
sa position par rapport à la ligne noire.

Entrée:   Position_Robot   octet       (voir ci-dessus)

Sorties:   CDE_ROUE_G       octet       Correspond au registre (OCR1A). Règle le rapport cyclique du
                                                signal de commande du moteur associé à la roue Gauche.
           CDE_ROUE_D       octet       Correspond au registre (OCR1B). Règle le rapport cyclique du
                                                signal de commande du moteur associé à la roue Droite.
-----*/

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: EtatCdeMot = CDE_CENTRE3;
  break;

  case CDE_CENTRE2: EtatCdeMot = CDE_CENTRE3;

  break;

  case CDE_CENTRE3: 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_CENTRE3;

  break;

  case CDE_DROITE: CDE_ROUE_G = _60PC; CDE_ROUE_D = _50PC;
    if (Position_Robot == CENTRE_3) EtatCdeMot = CDE_CENTRE3;
    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_3) EtatCdeMot = CDE_CENTRE3;
    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_CENTRE3;

} // Fin switch (EtatCdeMot)

// Fin de FS12 " Choisir les valeurs des rapports cycliques "
//-----

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

```