

SOMMAIRE

1/ Présentation du système	3
1-1/ Mise en situation :	3
• Environnement du système technique :	3
• Organisation de l'objet technique	3
• Milieux associés :	4
• Principe de fonctionnement.....	4
• La carte de commande :	4
• Algorithme du programme principal :	6
• Vue de la partie opérative de la maquette :	7
1-2/ Principales caractéristiques techniques du système réel :	7
1-3/ Modes de fonctionnement :	7
1-4/ Schéma fonctionnel de 1er degré de la carte de commande	8
1-5/ Schéma structurel de la partie commande	12
1-7/ Calcul de la vitesse réelle maximum des portes	14
 2/ Étude des fonctions principales du système technique	 15
2-1 Étude de FP1 : adaptation et mise en forme	15
• Structure associée à FS11	15
• Structure associée à FS12	16
2-2 Étude de FP2 et FP8 : traitement et affichage	16
• Caractéristiques du microcontrôleur	17
• Organisation matérielle de la fonction Traitement et Mémorisation :	17
• Menu utilisateur sur l'affichage LCD :	18
• Schéma structurel de FP2 et FP8 :	19
• Algorithmes du programme complet de l'objet technique.....	20
2-3 Étude de FP3 : commande du moteur.....	36
• Schéma structurel de FP3 :	36
• Structure associée à FS31	37
• Structure associée à FS32	37
• Etude du circuit LM3524.....	38
• Structure associée à FS33	38
• Structure associée à FS34	39
• Schémas de circulation du courant :	40
2-4 Étude de FP4 : sélection de la vitesse	41
• Schéma structurel de FP4 :	41
• Structure associée à FS41	42
• Structure associée à FS42	42
• Structure associée à FS43	43
2-5 Étude de FP5 : Génération des signaux de mise en fonctionnement.....	44
• Schéma structurel de FP5.....	44
• Structure associée à FS51	45
• Structure associée à FS52	45
• Structure associée à FS53	47
• Relevés expérimentaux des signaux de fonctionnement du LM2917.....	49
• Exemples de relevés de couple et de vitesse :	50

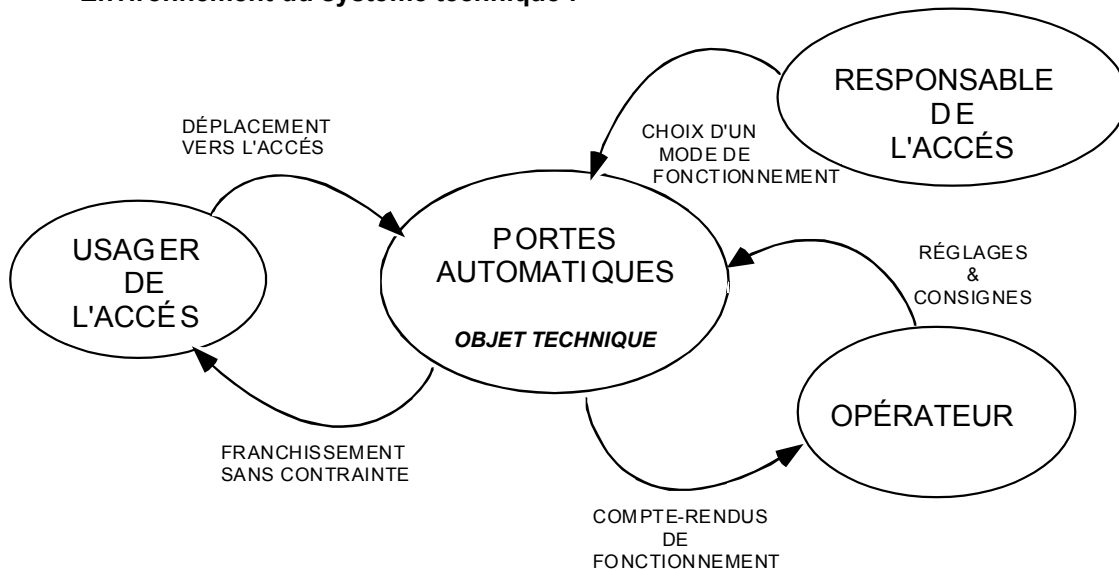
2-6 Étude de FP6 : calcul de la position réelle des portes.....	52
• Schéma structurel de FP6 :.....	52
• Structure associée à FS61	53
• Structure associée à FS62	54
• Structure associée à FS63	54
2-7 Étude de FP7 : Saisie des consignes.....	55
• Schéma structurel de FP7 :.....	55
• Structure associée à FA1,	56
• Structure associée à FS71	56
• Structure associée à FS72, FS73 et FS74	56
• Structure associée à FS73 Détection touche appuyée.....	57
• Structure associée à FS74 Encodage clavier	57
• Structure associée à FS75 Décodage clavier	58
3/ Physique appliquée : étude de FP5 :	59
• Schéma fonctionnel de degré 2 de la fonction FP5 :	59
• Schéma structurel de la fonction FS51 et FS53 :	59
3.1/ Chaîne de mesure du couple moteur	60
• Captage du couple :	60
• Choix de la fréquence de coupure du filtrage :	60
• Contrainte de l'alimentation :	60
• Choix de l'amplification en continu de la chaîne :	60
• Caractéristique de transfert de la chaîne de mesure du couple :	61
• Etude en régime permanent continu :	62
3.2 / Etude de la chaîne de mesure de la vitesse du moteur (FS53) :	63
3.3 / Détermination expérimentale de la constante de FEM du moteur :	65
• But de la manipulation :	65
• Protocole expérimental :	65
• Relevés expérimentaux :	65
• Exploitation des résultats :	65
3.4/ Chronogrammes qualitatifs :	66
• Phase d'apprentissage :	66
• Cycle d'ouverture et cycle de fermeture :	67
• Le filtrage de la chaîne de mesure du courant moteur :	68
4/ Documentation constructeur.....	69
• 4.1 Caractéristiques du moteur MX10	69
• 4.2 Documentation constructeur du radar de détection.....	70
5/ Documentation annexes	72
• 5.1 Schéma d'implantation de la carte de commande.....	72
• 5.2 Nomenclature des composants	73
• 5.4 Détails sur la procédure de calcul de la position des portes.....	75
• 5.5 Programme en C de la carte de commande.....	76

1/ Présentation du système

1-1/ Mise en situation :

Les lieux publics très fréquentés, et en particulier les grands magasins, sont équipés d'accès à ouverture des portes automatiques afin d'offrir aux usagers un accès aisé, fluide en toute sécurité et en toutes circonstances.

• Environnement du système technique :

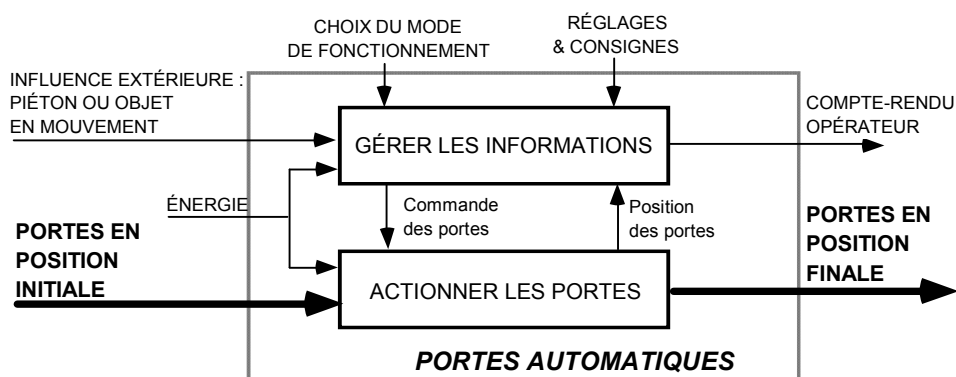


Cette étude porte sur l'objet technique PORTES AUTOMATIQUES.

• Organisation de l'objet technique

• Fonction d'usage de l'objet technique : assurer le déplacement automatique de portes coulissantes par détection de présence.

• Schéma fonctionnel de niveau 2 :



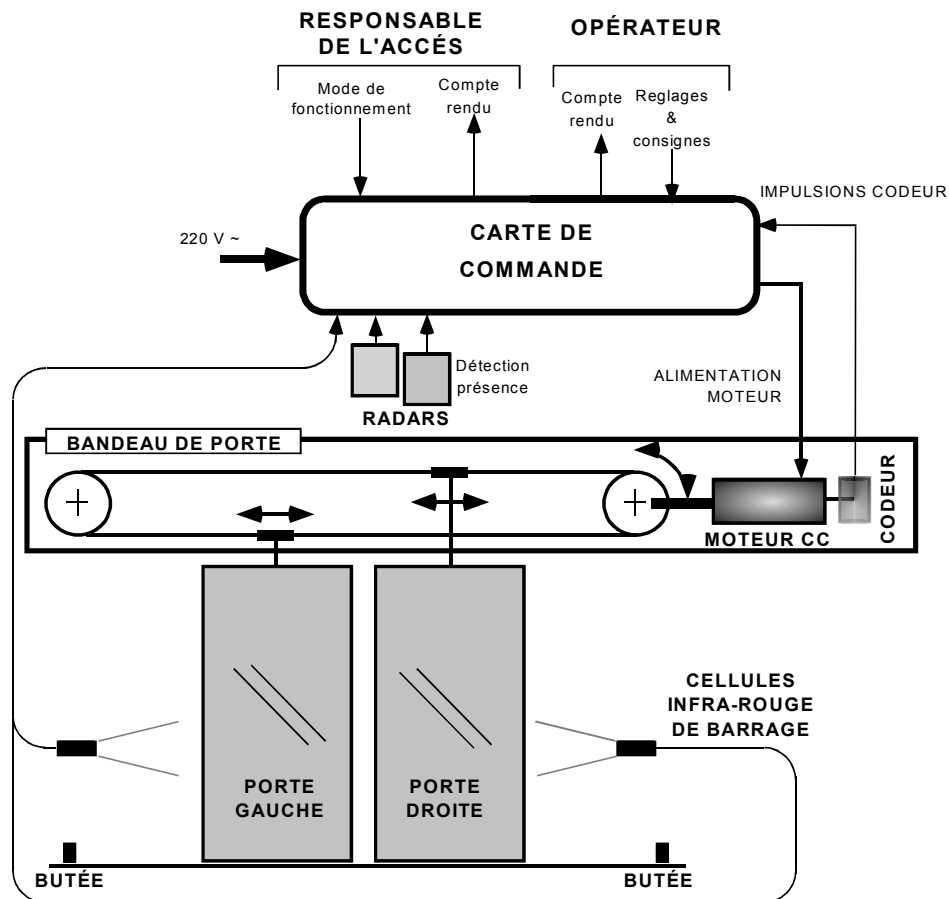
• Milieux associés :

- Humain : doit offrir une sécurité maximum pour l'utilisateur. Son installation et son réglage doivent être aisés.
- Physique : doit résister aux chocs et aux intempéries (humidité et chaleur) dans le cas de portes extérieures.
- Technique : l'objet technique est constitué d'éléments standard, faciles d'approvisionnement.
- Économique : c'est un matériel compétitif dans sa catégorie car capable de détecter tout blocage sans micro rupteurs, comes, embrayage ni amortisseurs pneumatiques.

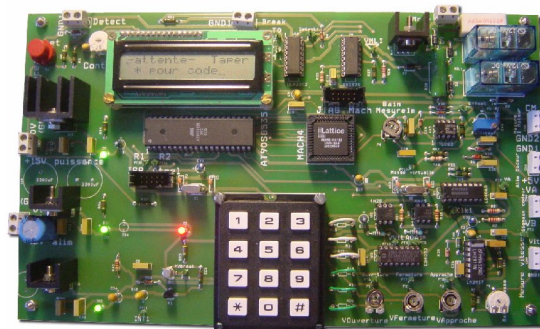
• Principe de fonctionnement

Le fonctionnement de la porte Stanley est axé autour d'un microcontrôleur ATMEL 90S8535 qui assure la gestion du déplacement des portes.

La position des portes est connue grâce aux impulsions fournies par un codeur monté sur l'axe du moteur. La détection de présence est effectuée par 2 radars (un intérieur et un extérieur), plus une barrière infrarouge.



• La carte de commande :



Les différents menus du logiciel offrent au technicien les possibilités :

- de paramétrer la marge de course (distance d'arrêt avant les butées)
- de paramétrer les tolérances de fermeture et d'ouvertures (afin de compenser les effets d'inertie de la chaîne cinématique)
- de modifier le code d'accès aux paramétrage
- de forcer une séquence d'ouverture – fermeture, afin de vérifier les réglages.

D'autre part, il est possible de régler indépendamment les vitesses d'ouverture, de fermeture et d'approche par l'intermédiaire de 3 potentiomètres.

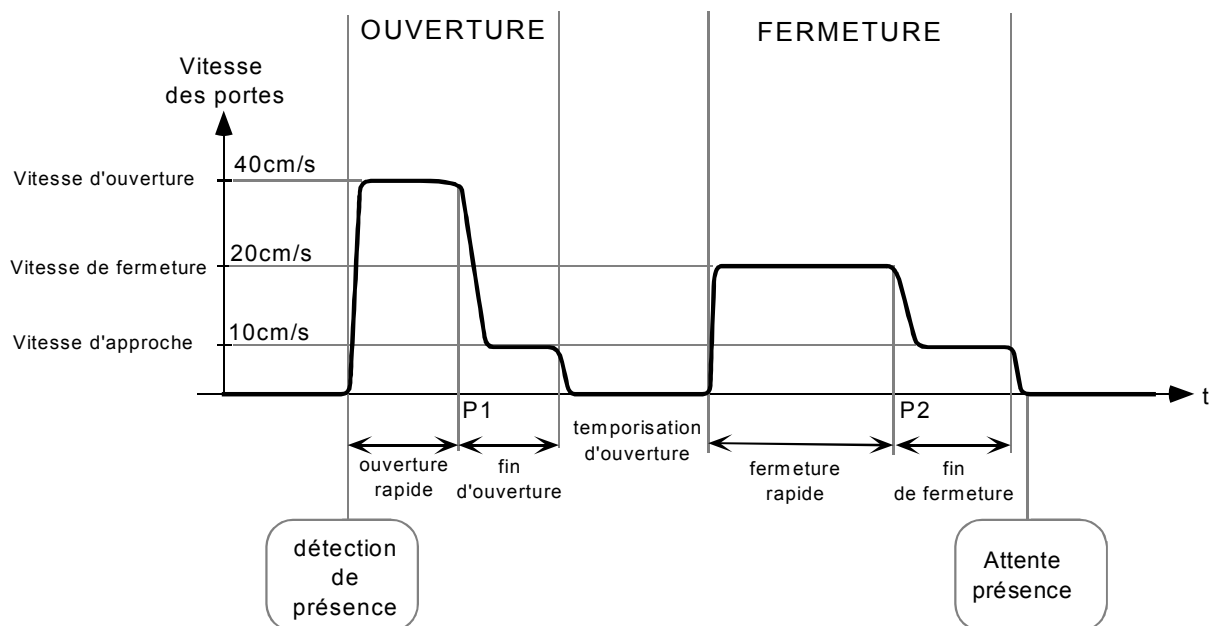
Phases de fonctionnement :

• **Initialisation** : après une mise sous tension le μ contrôleur procède à une phase d'apprentissage, les portes s'ouvrent au ralenti jusqu'au moment où les impulsions du codeur ne sont plus reçues (blocage sur la butée). Le nombre N d'impulsions reçues pendant le déplacement donne une image de la course d'ouverture L, les portes se referment alors de cette même distance L.

La fonction traitement et mémorisation a alors déterminé la course des portes ainsi que les positions de changement de vitesse

• **Attente de passage** : dès que l'un des radars (intérieur ou extérieur) détecte une présence, les portes s'ouvrent de la course L, d'abord rapidement, puis au ralenti. Après une temporisation, elles se referment de cette même distance :

Les paliers de changement de vitesse sont tels que $P1 = 1/3$ de la course pour la phase d'ouverture, et $P2 = 2/3$ de la course pour la phase de fermeture.

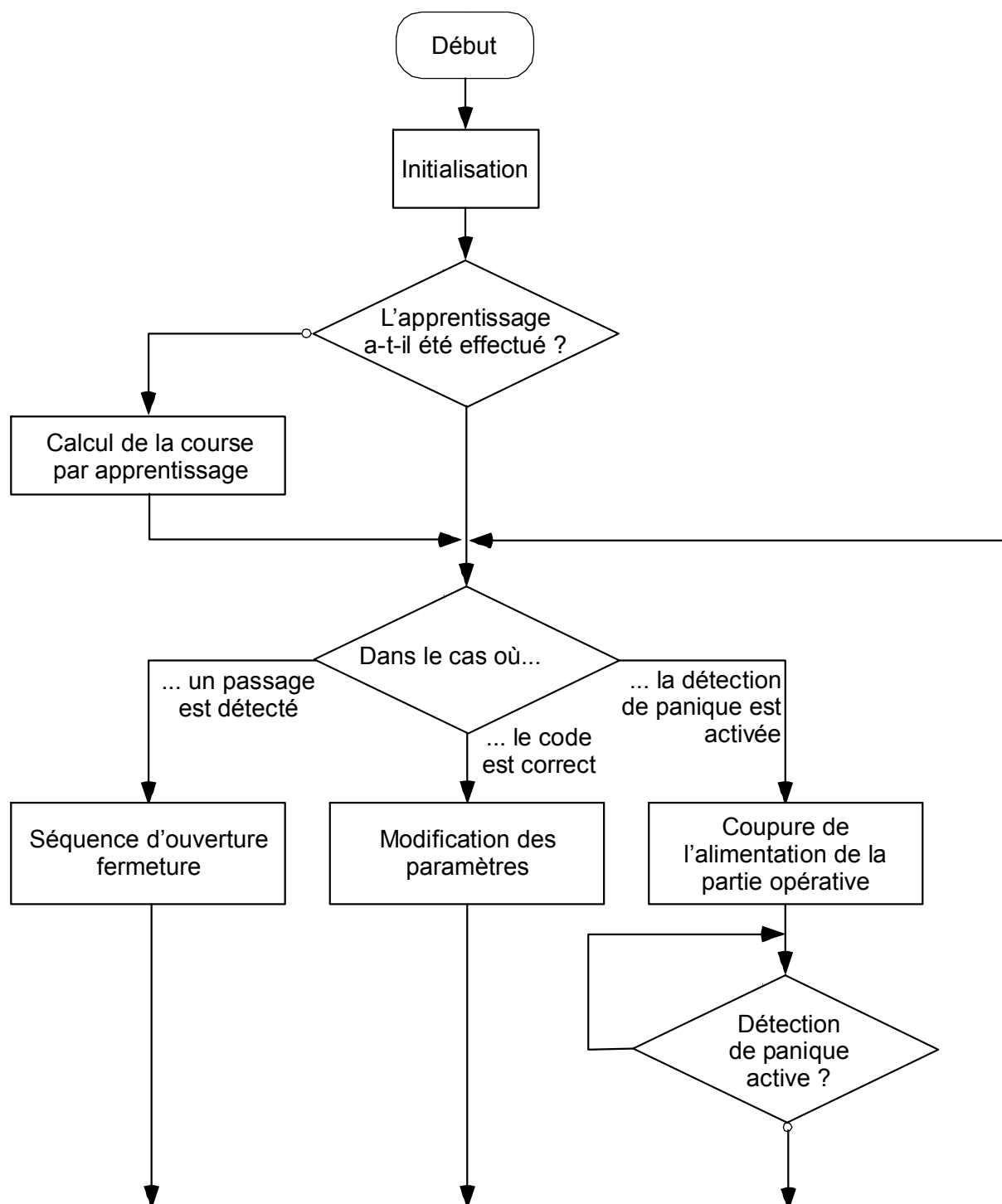


C'est grâce à une mesure constante du nombre d'impulsions reçues par unité de temps que le système calcul la position des portes et ajuste automatiquement la vitesse, de telle sorte que la course se finisse au ralenti.

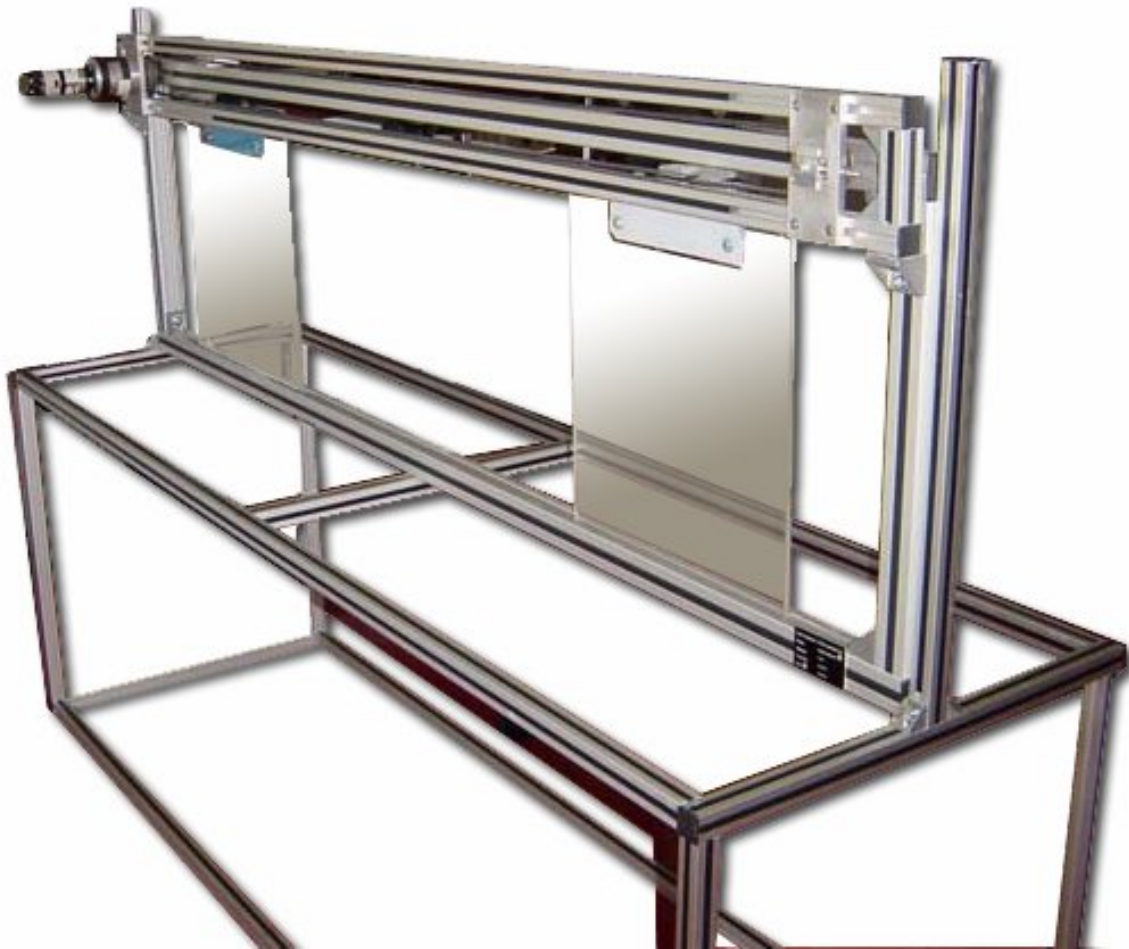
Au cas où une personne ou un objet stationnerait entre les portes (hors de portée des radars), une barrière infrarouge avertit le système de maintenir les portes ouvertes. Cette barrière n'est pas installée sur notre système. Dans tous les cas le signal DETECT est produit avec un OU logique entre les signaux issus des différents capteurs.

• **Blocage** : tout blocage des portes est détecté par l'absence d'impulsions avant la fin de course. Cela entraîne la réouverture immédiate des portes.

• Algorithme du programme principal :



• **Vue de la partie opérative de la maquette :**



1-2/ Principales caractéristiques techniques du système réel :

Alimentation 220 V monophasé 10A.
Température de fonctionnement de -30°C à +50°C.
Système anti-panique assuré par batteries (non fournies).
Transmission par courroie crantée.
Masse d'un ventail (porte) : 100kg.
Hauteur ajustable de la porte.

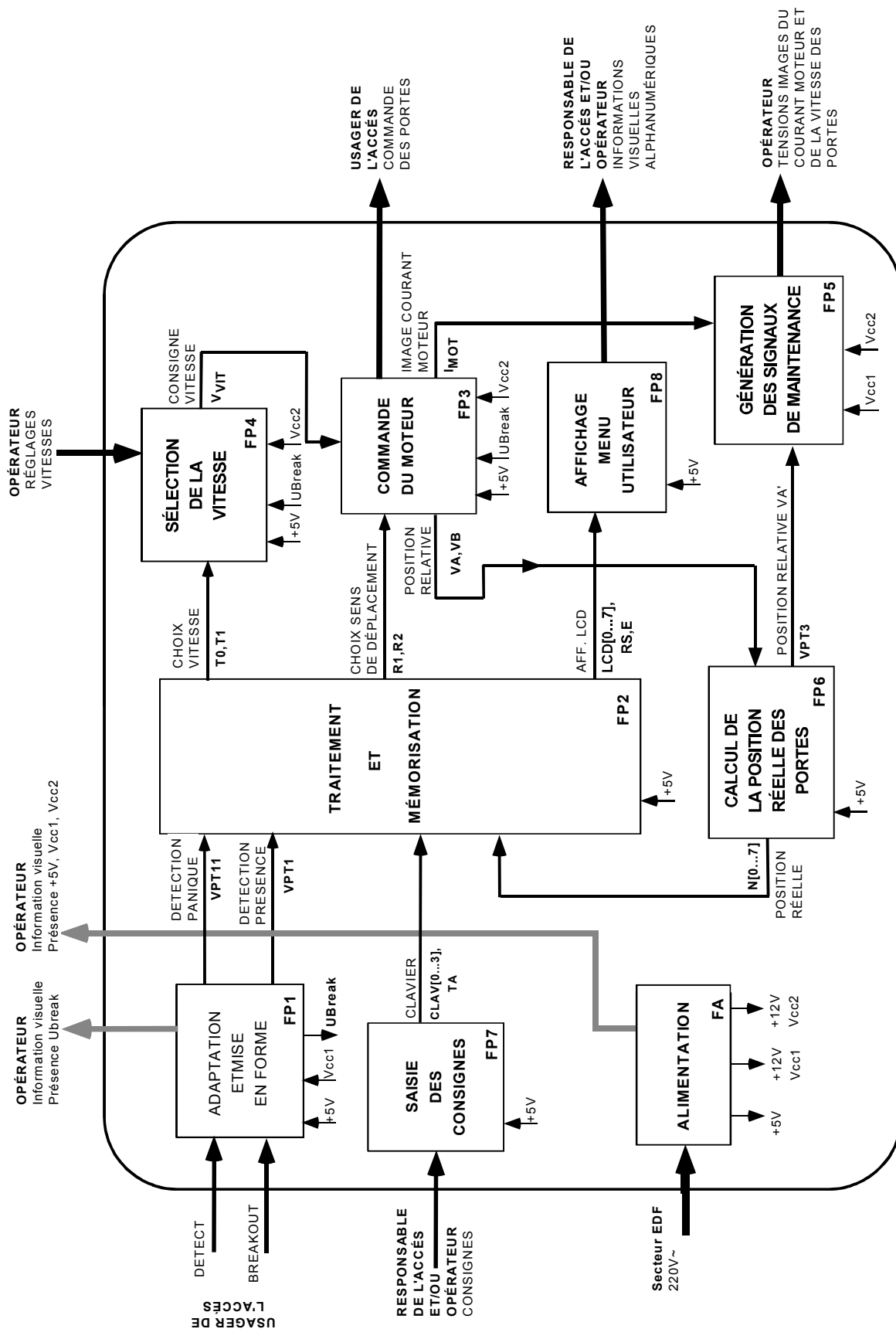
Moto réducteur :

- Moteur électrique de 75V à courant continu, à aimant permanent.
Vitesse nominale de 3000 tr/mn.
- Réducteur 1/7
- Poulie de diamètre primitif de 64,68 mm et de largeur 19,1 mm.
- Courroie crantée (16 crans) au pas de 12,7 mm.

1-3/ Modes de fonctionnement :

Marche / Arrêt des portes.
Fonctionnement automatique : ouverture puis fermeture automatique.
Portes maintenues en position ouverte.
Portes maintenues en position fermée.

1-4/ Schéma fonctionnel de 1er degré de la carte de commande



Description des fonctions principales :

FA : FONCTION ALIMENTATION : fournit les tensions continues de +5V et $V_{cc1} = +12V$ et $V_{cc2} = +12V$ pour l'alimentation en énergie des fonctions principales.
La fonction assure une isolation entre les tensions V_{cc1} et V_{cc2} , cette dernière servant à la commande par MLI du moteur est sources de parasites et de perturbations.

FP1 : ADAPTATION ET MISE EN FORME : permet l'adaptation et la mise en forme des informations provenant des capteurs à un format adapté à FP2. La fonction assure également l'extinction de la tension de sécurité Ubreak en cas de détection de panique.

Entrées :

- DETECT :
 - détection d'un usager de l'accès.
 - circuit ouvert si absence d'usager.
 - circuit fermé si présence d'un usager.
- BREAKOUT :
 - détection de panique.
 - circuit fermé si les portes sont en position normale de fonctionnement
 - circuit ouvert si les portes sont ouvertes en vantaux.

Sorties :

- VPT1 :
 - détection présence d'un usager de l'accès.
 - $V_{pt1} = 0V$ si absence d'usager.
 - $V_{pt1} = 5V$ si présence d'un usager.
- VPT11 : détection panique
 - $V_{pt11} = 0V$ si absence de détection panique.
 - $V_{pt11} = 5V$ si présence de détection panique.
- Ubreak : sécurité détection panique
 - $U_{break} = V_{cc1} = 12V$ si absence de détection panique.
 - $U_{break} = 0V$ si présence de détection panique.
 - La présence de Ubreak est visualisée par l'opérateur à l'aide d'une diode électroluminescente LD2

FP2 TRAITEMENT ET MÉMORISATION : grâce à son logiciel de traitement, cette fonction assure la gestion du cycle d'ouverture / fermeture des portes, en tenant comptes des consignes utilisateur.

Entrées :

- VPT1 : détection présence d'un usager de l'accès. (Cf FP1).
- VPT11 : détection panique (Cf FP1).
- CLAV[0...3] : Code binaire sous 4 bits représentatif de la touche enfoncée sur le clavier 12 touches.
- TA : Signal binaire au niveau haut lorsqu'une touche du clavier est appuyée.
- N[0...7] : Nombre binaire représentatif de la position réelle des portes issu de FP6.

Sorties :

- T0,T1 : choix de la vitesse de déplacement des portes
Signaux binaires permettant la sélection d'une vitesse parmi 4 vitesses : nulle, approche, fermeture et ouverture.
- R1,R2 : sélection du sens de déplacement des portes. Signaux binaires permettant la fermeture, l'ouverture ou le freinage (arrêt) des portes.
- LCD[0...7] : Données sous 8 bits vers l'afficheur LCD de FP7.
- RS, R/W, E : Bits de contrôle vers l'afficheur LCD de FP7.

FP3 COMMANDE DU MOTEUR : génère un signal de commande du moteur en fonction de la consigne de vitesse sélectionnée par FP4, assure l'alimentation en puissance du moteur actionnant les portes à partir du signal de commande fourni par FP4 et du sens de rotation défini par FP2. Cette fonction fournit constamment à l'aide du codeur incrémental la position relative des portes ainsi que le courant consommé à travers le moteur.

En cas de détection panique, la tension Ubreak disparaît et le moteur est mis hors tension.

Entrées :

- R1,R2 : sélection du sens de déplacement des portes (Cf FP2).
- V-VIT : tension analogique image de la consigne de vitesse élaborée par FP4. V-VIT peut prendre 4 valeurs de tension dont le 0V. Les 3 autres sont réglables par l'opérateur.

Sorties :

- VA,VB : Signaux binaires 0-5V en quadrature de phase issus du codeur incrémental
- IMOT : tension image du courant consommé à travers le moteur
- COMMANDE DES PORTES : action mécanique sur les portes pour permettre l'ouverture et la fermeture de celles-ci à différentes vitesses.

FP4 SÉLECTION DE LA VITESSE : génère un signal de commande du moteur en fonction de la consigne de vitesse sélectionnée par FP2. Cette consigne est inhibée en cas d'absence de la tension Ubreak.

Entrées :

- T0,T1 : choix de la vitesse de déplacement des portes (Cf FP2).

Sorties :

- V-VIT : tension image de la consigne de vitesse élaborée par FP4. (Cf FP3)

FP5 GÉNÉRATION DES SIGNAUX DE MAINTENANCE : génère les tensions images du courant moteur et de la vitesse de déplacement des portes utiles à l'opérateur pour les différents réglages lors de la première installation ou lors d'une modification du système (type de portes par ex.)

Entrées :

- IMOT : tension image du courant consommé à travers le moteur.
- VA' : Signal VA binaires 0-5V issus du codeur incrémental protégé contre une déconnexion du câble codeur par FP6 (Calcul de la position réelle des portes)

Sorties :

- Vitesse portes : tension continue image de la vitesse de déplacement des portes.
- Courant Moteur : tension continue image du courant moyen consommée à travers le moteur.

FP6 CALCUL DE LA POSITION RÉELLE DES PORTES : génère à partir de la position relative fournie par le codeur incrémental un nombre binaire représentatif de la position réelle vers FP2 Traitement. Les deux entrées nécessitent d'être protégées contre les potentiels flottants : cas où le câble qui relie la carte de commande au codeur est déconnecté ou sectionné.

Entrées :

- VA,VB : Signaux binaires 0-5V issus du codeur incrémental en quadrature de phase.

Sorties :

- N[0...7] : Nombre binaire représentatif de la position réelle des portes.
- VA' : Signal VA binaires 0-5V issus du codeur incrémental protégé contre une déconnexion du câble codeur.

FP7 SAISIE DES CONSIGNES : cette fonction assure l'acquisition et le transcodage des consignes utilisateur vers l'objet technique. Elle est réalisée autour d'un clavier matriciel 12 touches.

Entrées :

- CONSIGNES UTILISATEUR : action mécanique de l'utilisateur sur le clavier 12 touches, ces touches ayant une fonction relative au menu en cours visualisé sur l'afficheur LCD (FP8).

Sorties :

- CLAV[0...3] : Code binaire sous 4 bits représentatif de la touche enfoncée sur le clavier 12 touches.
- TA : Signal binaire au niveau haut lorsqu'une touche du clavier est appuyée.

FP8 AFFICHAGE MENU UTILISATEUR : cette fonction assure l'affichage du menu logiciel de l'objet technique. Elle est basée sur la mise en œuvre d'un afficheur de type LCD.

Entrées :

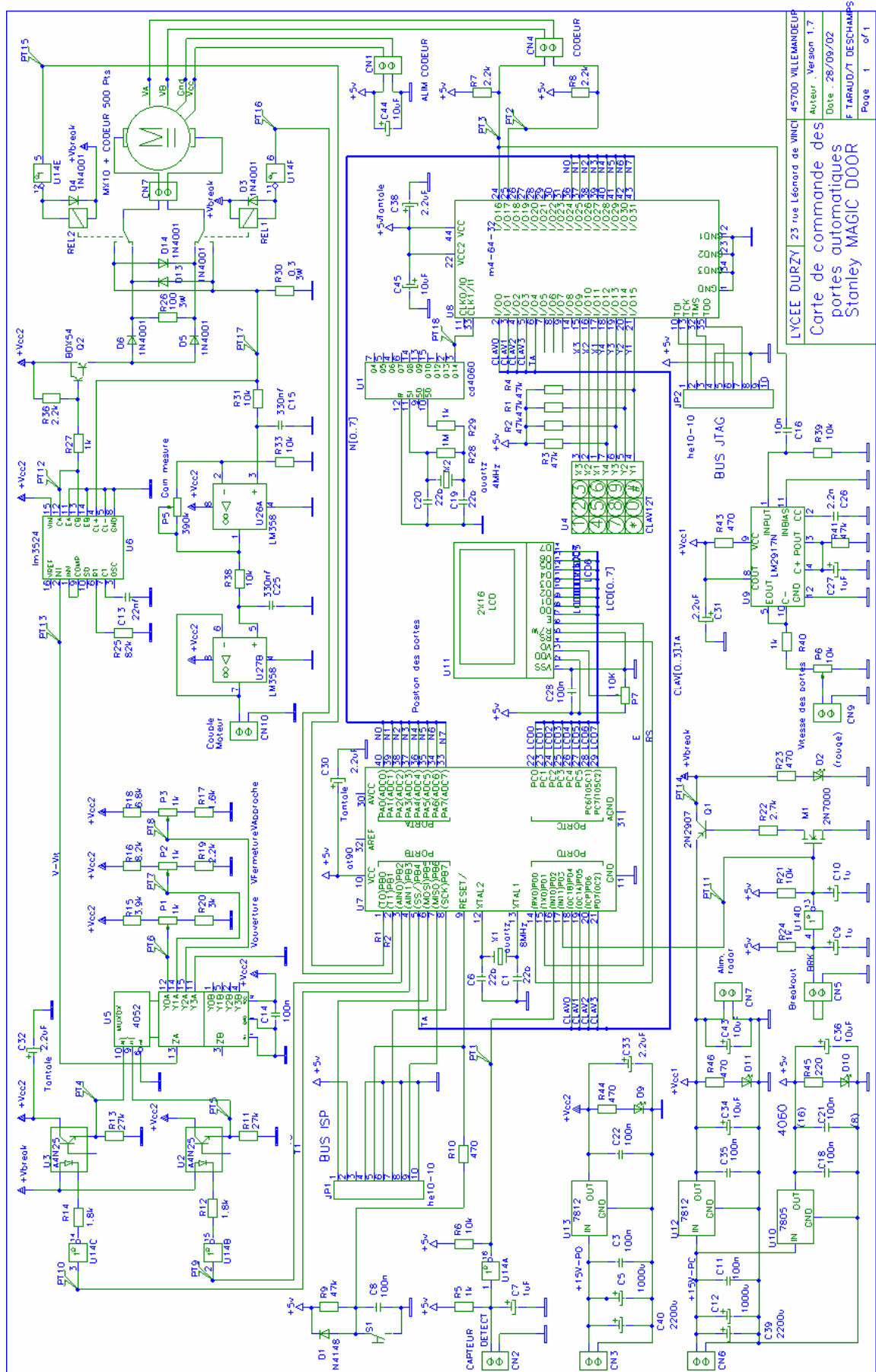
- LCD[0...7] : Données sous 8 bits pour l'afficheur LCD de FP2.
- RS, E : Bits de contrôle pour l'afficheur LCD de FP2.

Sorties :

- INFORMATION UTILISATEUR : informations visuelles de type alphanumérique sur l'écran de l'afficheur LCD

NB : Il n'y a pas de découpage fonctionnel du 2^{ieme} degré pour les fonctions FP2 et FP8

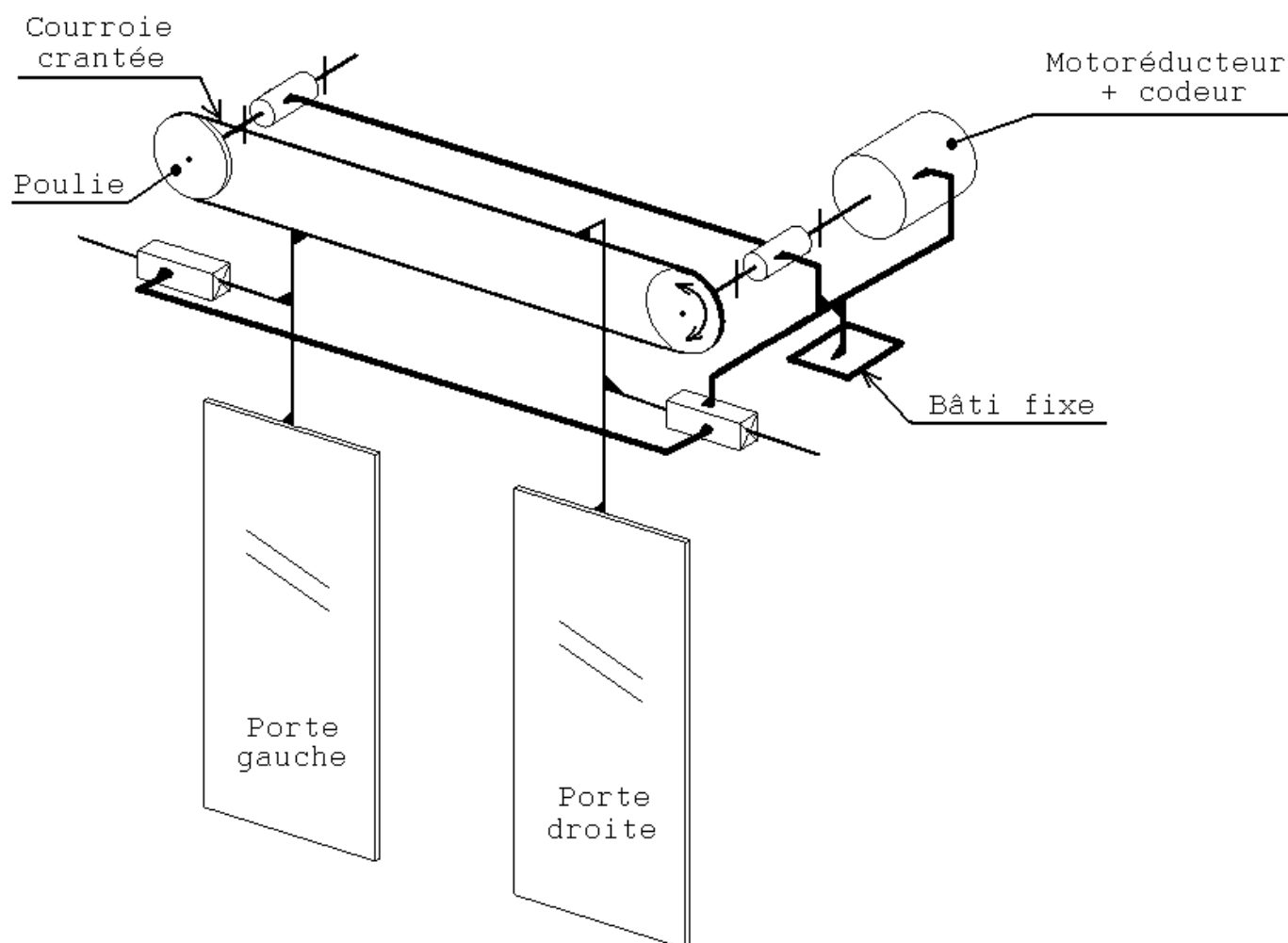
1-5/ Schéma structurel de la partie commande



1-6/ Schéma cinématique de la partie opérative

Caractéristiques du moteur :

Référence :	MX10
Vitesse nominale :	3000 tr/mn
Tension nominale :	75 V
Courant nominale :	2,2 A
Puissance nominale :	110 W
Couple nominale :	35 N.cm



1-7/ Calcul de la vitesse réelle maximum des portes

La poulie menante étant montée directement sur l'axe de sortie de l'ensemble Moteur + Réducteur, la relation reliant la vitesse de déplacement des portes à la vitesse de rotation des portes est la suivante :

Soit :

- ω_p la vitesse angulaire de la poulie en rad/s
- R_p le rayon de la poulie en m (0.032m)
- N_p la vitesse de rotation poulie en tr/mn
- N_m la vitesse de rotation moteur sans réduction en tr/mn
- V la vitesse de déplacement des portes en m/s
- $V = R_p \cdot \omega_p$

$$V = (2\pi \cdot R_p \cdot N_p) / 60$$

$$V = (2\pi \cdot R_p \cdot N_m) / (60 \cdot 7)$$

$$V = N_m / 2089 \text{ en m/s}$$

N_m nominal = 3000 tr/mn pour une alimentation nominale de 75V.

Dans notre système, le moteur est alimenté sous une tension de 12V, soit un rapport de tension de 6,25

$U = E + r \cdot I = k \cdot N + r \cdot I$ avec k constante de fem = 21,3 mV/tr/mn et r résistance d'induit = 5,1 ohms

soit $N = (U - r \cdot I) / k$. La vitesse N dépend donc de I donc du couple

Sous une alimentation continue de 12 V, considérons $N_m \text{ max} = N_m \text{ nom} / 6,25$

soit $N_m \text{ max} = 480 \text{ tr/mn}$

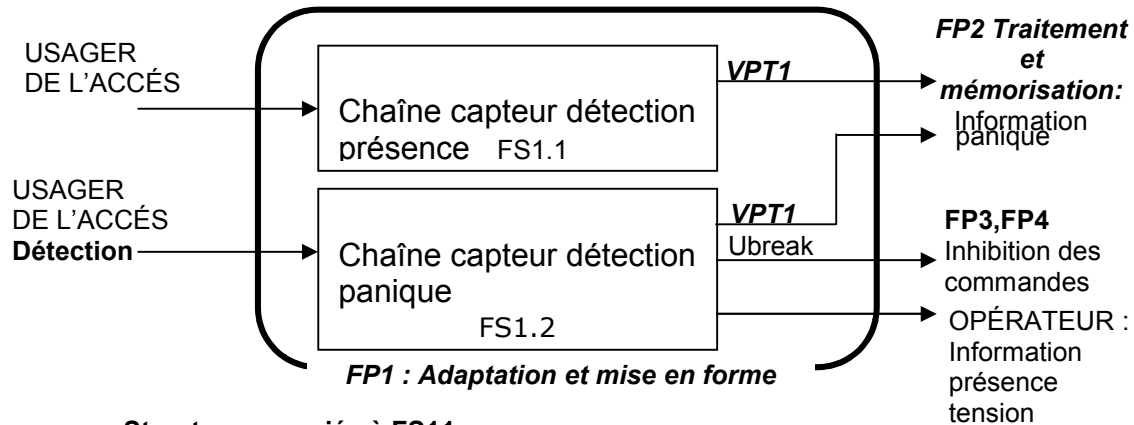
Pour $N_m \text{ max}$, $V_{\text{max}} = 0,23 \text{ m/s}$

Note : cette valeur ne donne qu'un ordre de grandeur des vitesses de déplacement des portes

2/ Étude des fonctions principales du système technique

2-1 Étude de FP1 : adaptation et mise en forme

La fonction permet l'adaptation et la mise en forme des informations provenant des capteurs à un format adapté à FP2. La fonction assure également l'extinction de la tension de sécurité Ubreak en cas de détection de panique.



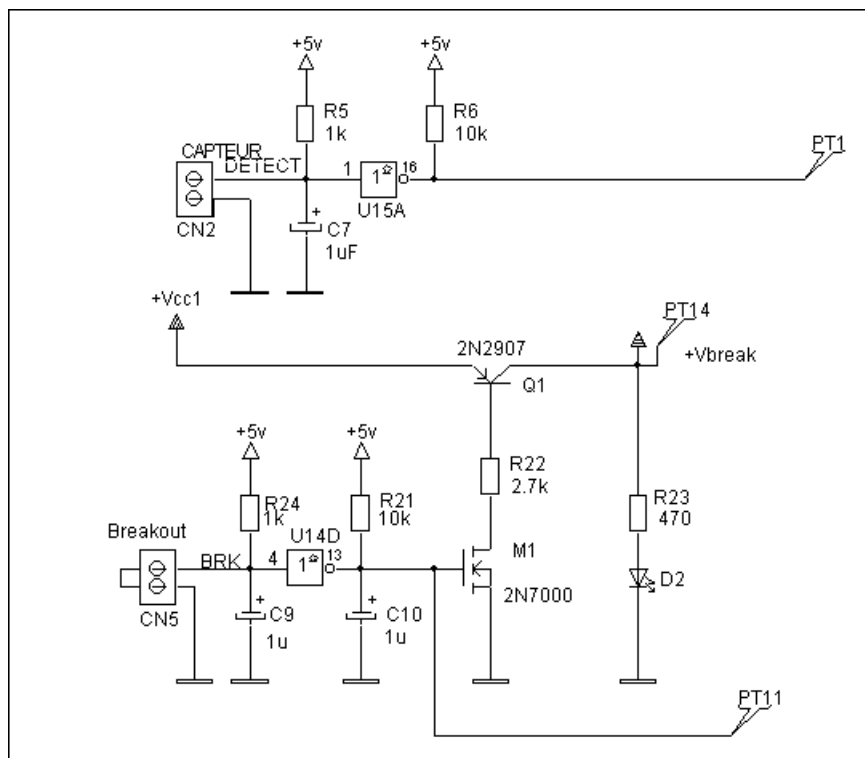
- **Structure associée à FS11**

Le capteur est équivalent à un contact normalement ouvert. Lors du passage d'une personne le contact se ferme.

En absence de passant, la résistance de tirage R5 impose un niveau logique haut à l'entrée de la porte donc PT1 est à un niveau logique 0.

Lors du passage d'une personne, UC7 = 0V et le transistor de sortie de U14A est bloqué. C'est donc la résistance de tirage R6 qui impose un niveau logique haut.

Le condensateur C7 se charge à travers R5 lorsque la personne est passée. Ce circuit permet d'obtenir un niveau logique haut ($V_{IHmin} = V_{cc}/2 = 2,5V$) à l'entrée de la porte au bout de $(-1000 \times 10^{-6} \times \ln(0,5)) = 690 \mu s$. Cela permet de s'affranchir des rebonds du capteur.



Si le capteur n'est pas relié à la carte de commande, R5 impose à l'entrée de la fonction un potentiel +5V.

- **Structure associée à FS12**

BREAKOUT :

Le capteur de panique est un contact normalement fermé. Ce capteur s'ouvre si les panneaux de la porte sont forcés à l'ouverture sous l'effet de la panique. Les vantaux peuvent alors s'ouvrir comme des portes classiques.

On retrouve le même montage que pour le circuit Detect.

Lorsque le contact est fermé, BRK est au niveau logique 0 et PT11 est forcé au niveau logique 1 par la résistance de tirage R21.

Quand le contact est ouvert, le condensateur C9 se charge à travers R24 et atteint le niveau logique haut ($V_{IHmin} = V_{cc}/2$) au bout de 690µs. La charge est très rapide mais permet de s'affranchir des rebonds du capteur. Alors PT11 est au niveau logique 0.

Dans le cas d'une panique de très courte durée, le contact est à nouveau fermé. Le circuit se retrouve dans la première configuration, mais le condensateur C10 va se charger à travers R21. PT11 atteindra le niveau logique haut (2,5V) au bout de 6,90ms. Cela permet au programme d'avoir effectué toutes les opérations nécessaires et aux courants présents dans les bobines des relais REL1 et REL2 d'être totalement évacués.

Ubreak :

M1 est un transistor MOS CANAL N à enrichissement (le canal n'existe pas pour $V_{gs} = 0$).

Pour le rendre passant, il faut lui appliquer une tension $V_{gs} > V_p$ (3V max).

Lorsque le contact est fermé, $V_{pt11} = 5V$ environ. M1 se comporte comme un circuit fermé entre Drain et Source ($R_{dson} = 10\Omega$ max). Un courant de base est alors extrait de Q1 à travers R22.

Lorsque le contact est ouvert, $V_{pt11} = 0V$ environ. $V_{gs} < V_p$ et M1 se comporte comme un circuit ouvert entre Drain et Source. Le courant de base ne peut plus être extrait de Q1 à travers R22. La Base de Q1 se trouve déconnecté et ne peut donc pas conduire.

Q1 est un transistor bipolaire de type PNP. Il est capable de véhiculer le courant nécessaire aux commandes des relais, des leds des opto-coupleurs ainsi qu'à la diode électroluminescente LD2. Cela impose un courant d'environ 160mA (60mA de courant de maintien pour un relais, environ 10mA par led-opto et 20 mA pour LD2).

Pour un courant I_c de l'ordre de 150 mA, le coefficient d'amplification min en courant vaut 100, V_{cesat} vaut - 0,4V et V_{besat} vaut -2,3V max. Pour M1 on donne V_{dson} max = 2,5V

Le Calcul de $I_{bsat} = (V_{cc1} + V_{besat} - V_{dson} \text{ max}) / R22$

$I_{bsat} = 2,67 \text{ mA}$

Le coefficient de sursaturation vaut $(I_{bsat} * \beta_{min}) / I_{csat} = 1,7$.

2-2 Étude de FP2 et FP8 : traitement et affichage

La fonction FP2 : Traitement et mémorisation

Cette fonction assure grâce à son logiciel de traitement :

- l'apprentissage et la mémorisation de la course des portes à la mise sous tension du système.
- le séquençement et la gestion des cycles d'ouverture / fermeture des portes, en tenant comptes les consignes utilisateur et la sécurité de l'utilisateur.
- le pilotage de la fonction FP8 Affichage menu utilisateur.

La fonction est réalisée autour du microcontrôleur **ATMEL AT90S8535**.

- **Caractéristiques du microcontrôleur**

- 8 Ko de mémoire programme du type FLASH (1000 cycles de programmation),
- 512 octets de EEPROM,
- 512 octets de RAM,
- 4 ports E/S,
- 2 timers 8 bits,
- Un timer 16 bits,
- Un convertisseur A / N 10 bits avec 8 entrées multiplexées
- Un comparateur analogique,
- Un chien de garde (WatchDog),
- Une interface série synchrone (SPI),
- Une interface série asynchrone (UART).
- Fréquence du quartz = 8Mhz

- **Organisation matérielle de la fonction Traitement et Mémorisation :**

Reset manuel :

Lors d'un appui sur le bouton poussoir S1, un niveau logique bas est appliqué sur la patte 9 du micro contrôleur. Le Reset est actif.

Reset automatique :

Lors de la mise sous tension du circuit, le condensateur C8 se charge à travers R9 pour atteindre $0,6.V_{cc}$ (V_{IHmin} du AT90S8535) au bout de $(-47000 \times 100 \times 10^{-9} \times \ln(0,4)) = 4,3ms$ (voir la simulation).

Cela permet de faire un Reset automatique à la mise sous tension du système. Lors de l'extinction de l'alimentation, la diode D1 permet de décharger instantanément le condensateur C8. Ceci permet un nouveau Reset automatique si le système est à nouveau alimenté rapidement.

Une impulsion sur l'entrée Reset/ du micro contrôleur entraîne un Reset interne d'une durée de 11ms minimum (p20 de la doc Atmel).

La constante de temps du circuit n'est donc pas calculé en fonction du temps minimum pendant lequel le niveau bas doit être appliqué sur l'entrée Reset du micro contrôleur. Elle permet de s'affranchir des rebonds du bouton poussoir S1 afin d'éviter des remises à zéro successives.

La résistance R10 permet d'éviter les conflits entre le bus ISP et le circuit de Reset lors de la programmation du microcontrôleur.

Broches μC utilisées	Nature	Information
PortA [PA7..PA0]	Entrées logiques	Position réelle des portes [N7...N0]
PortB PB0,PB1	Sorties logiques active à l'état haut	Sens rotation moteur R1,R2

PortB PB2,PB3	Sorties logiques active à l'état haut	Sélection vitesse T0,T1
PortB PB4	Entrée logique	Détection touche appuyée TA
PortB PB5,PB6,PB7	Entrées/Sorties logiques	Programmation in situ BUS ISP
PortC [PC7..PC0]	Sorties logiques	Données afficheur LCD [LCD7...LCD0]
PortD PD0,PD1	Sorties logiques active à l'état haut	Contrôle afficheur LCD RS/E
PortD PD2 (INT0)	Entrée logique active à l'état haut	Détection présence usager DETECT
PortD PD3 (INT1)	Entrée logique active à l'état haut	Détection panique BREAKOUT
PortD [PD4..PD7]	Entrées logiques	Code touche appuyée CLAV[0...3]
/RESET	Entrée logique active à l'état bas	Mise sous tension ou reset manuel S1
XTAL1,XTAL2	Entrées d'horloge	Horloge 8Mhz

La fonction FP8 : Affichage menu utilisateur

Offre une interface visuelle agréable à l'utilisateur pour l'élaboration du choix de fonctionnement, la saisie des consignes et les réglages de différents paramètres. La fonction consiste en la mise en œuvre d'un afficheur à cristaux liquides U11. On remarquera que seuls 2 bits de contrôle sont nécessaires puisque que l'entrée R/W est reliée à la masse.

- **Menu utilisateur sur l'affichage LCD :**

A la mise sous tension, l'afficheur affiche « Taper * pour code ».

Le responsable d'accès ou l'opérateur doit connaître et taper le code d'accès pour accéder aux sous-menus suivants :

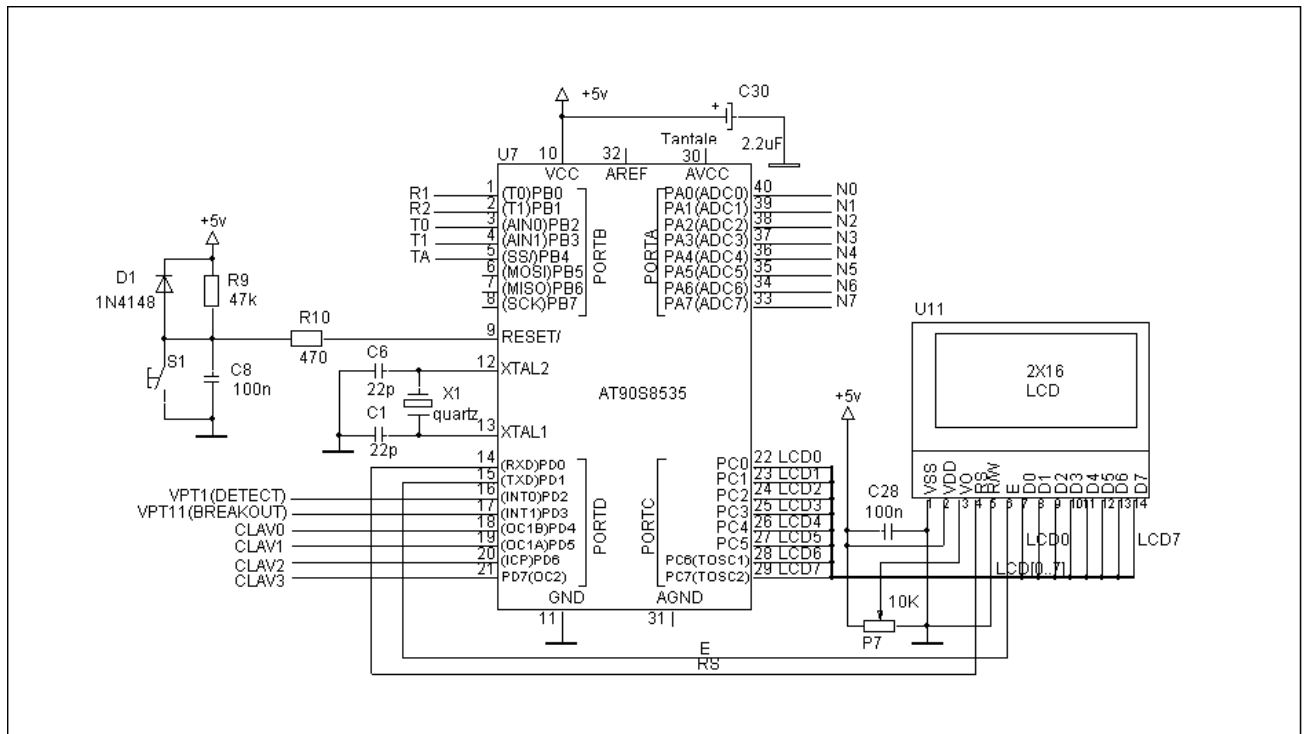
Appuyer sur :

- 1 : Changer le code
- 2 : K vitesse (facteur vitesse)
- 3 : Tolérance Ouverture
- 4 : Tolérance Fermeture
- 5 : Tolérance Course
- 6 : Ouvrir Portes
- # : Quitter

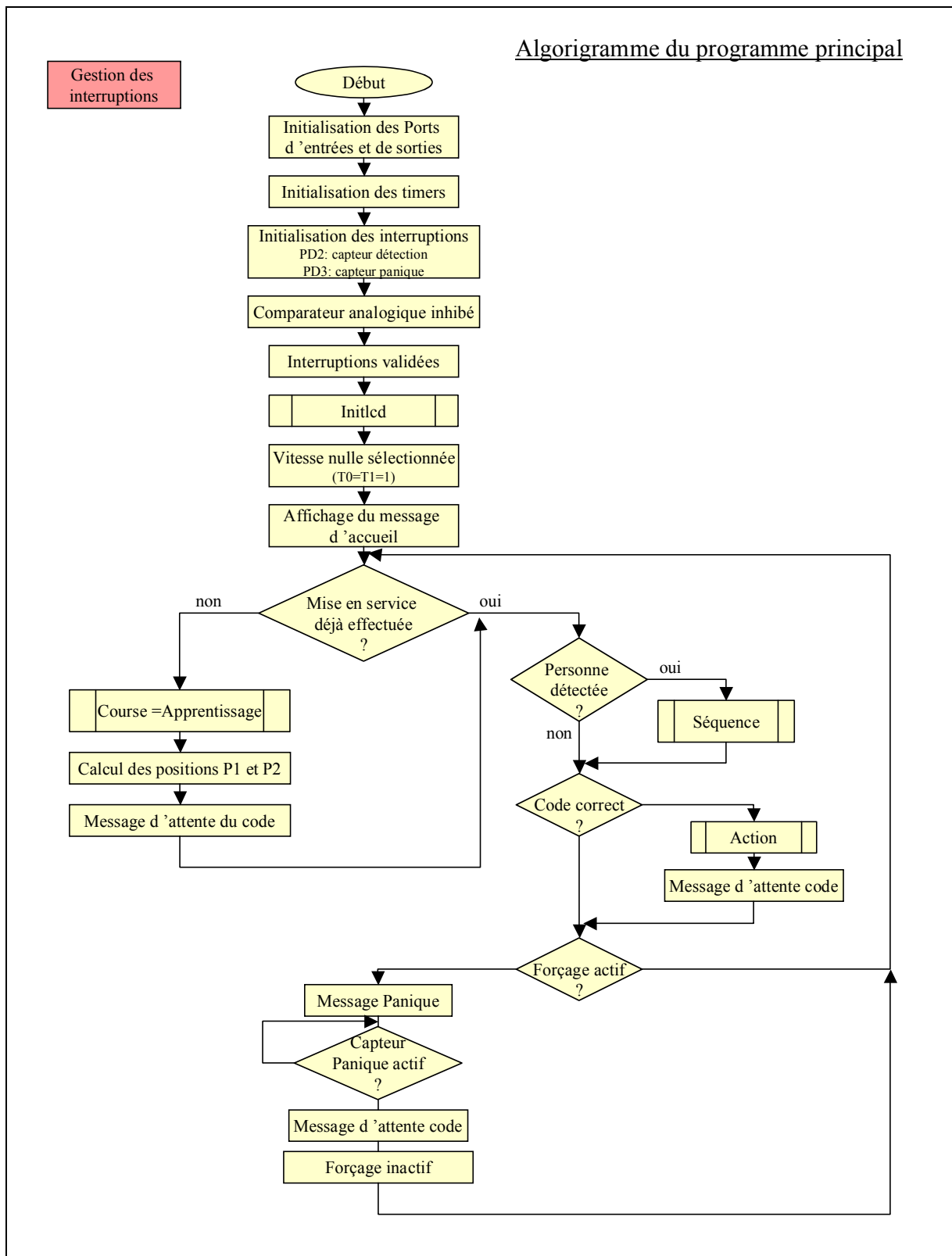
L'affichage étant sous forme de 2 lignes de 16 caractères, le menu déroule constamment jusqu'au choix du sous menu.

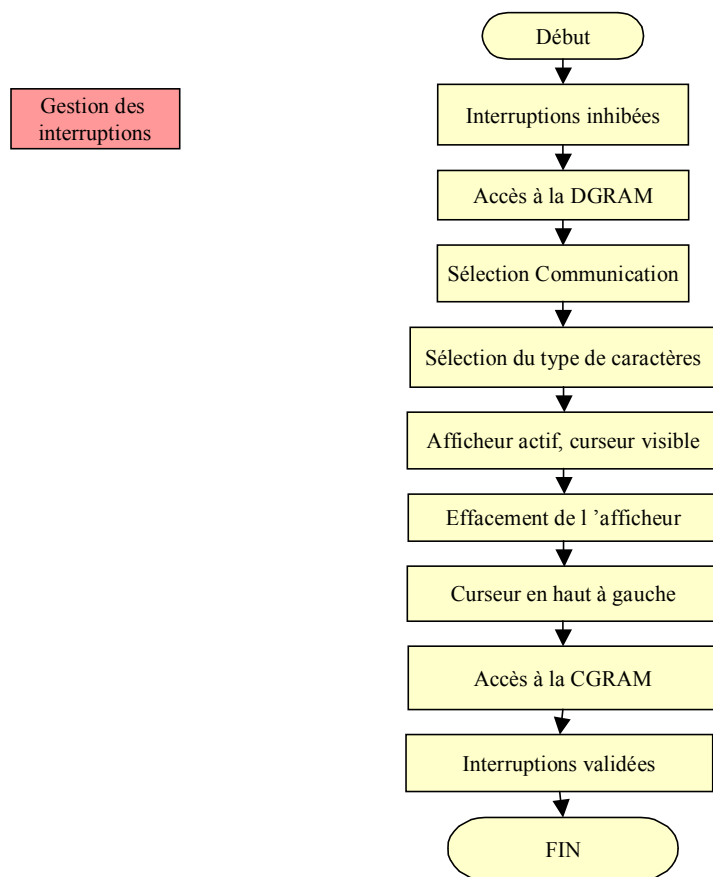
L'appuie sur #Quitter verrouille le menu et l'affichage affiche « Taper * pour code ».

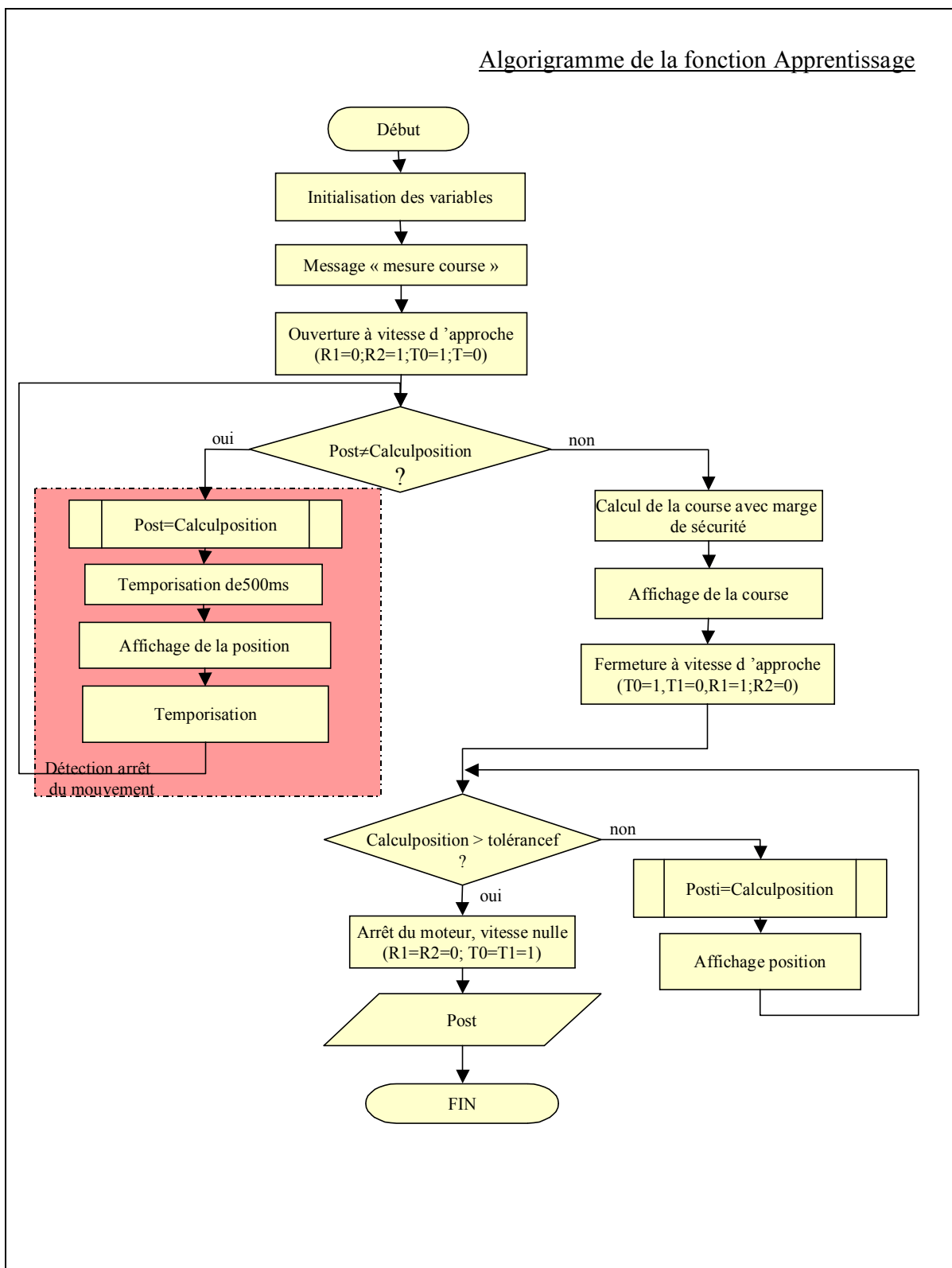
- Schéma structurel de FP2 et FP8 :

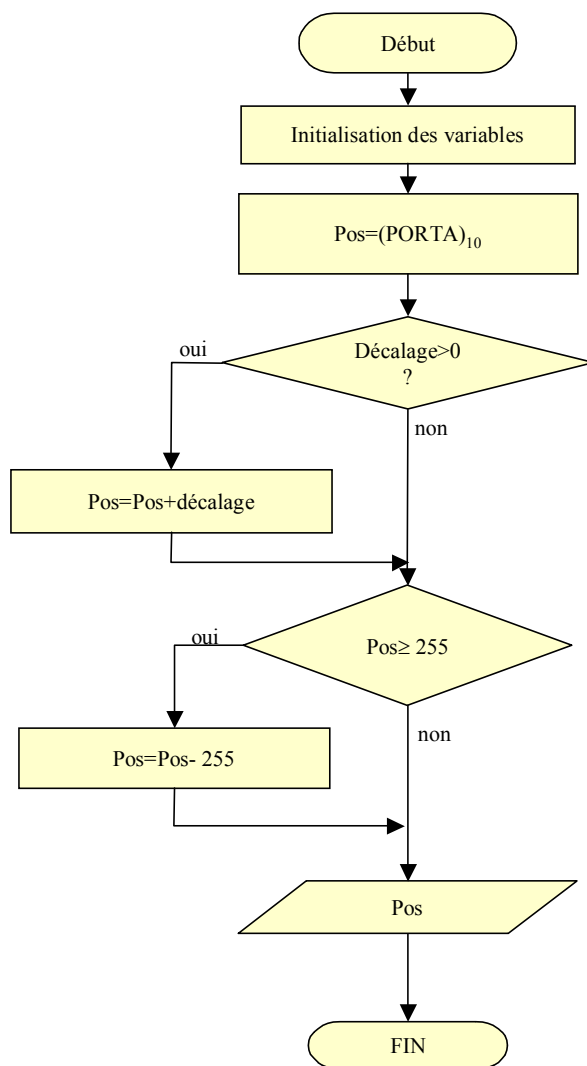


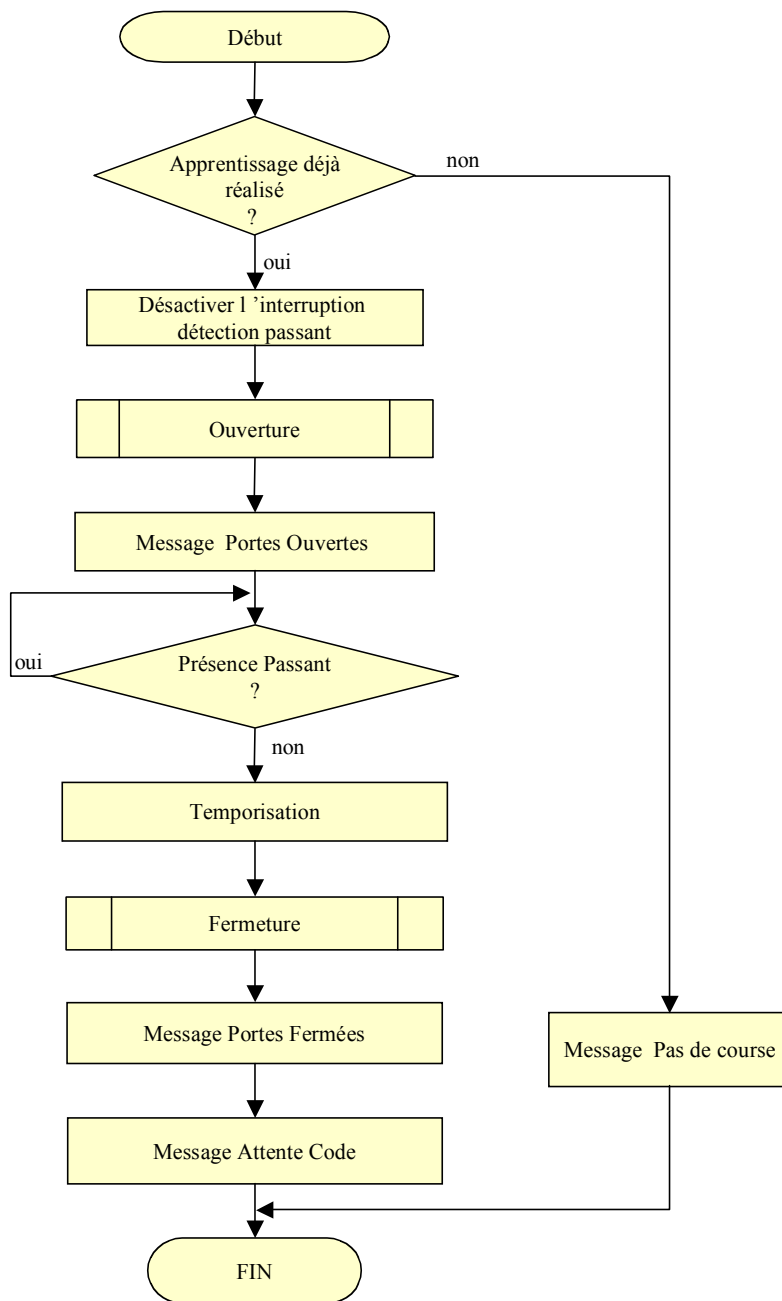
- Algorigrammes du programme complet de l'objet technique

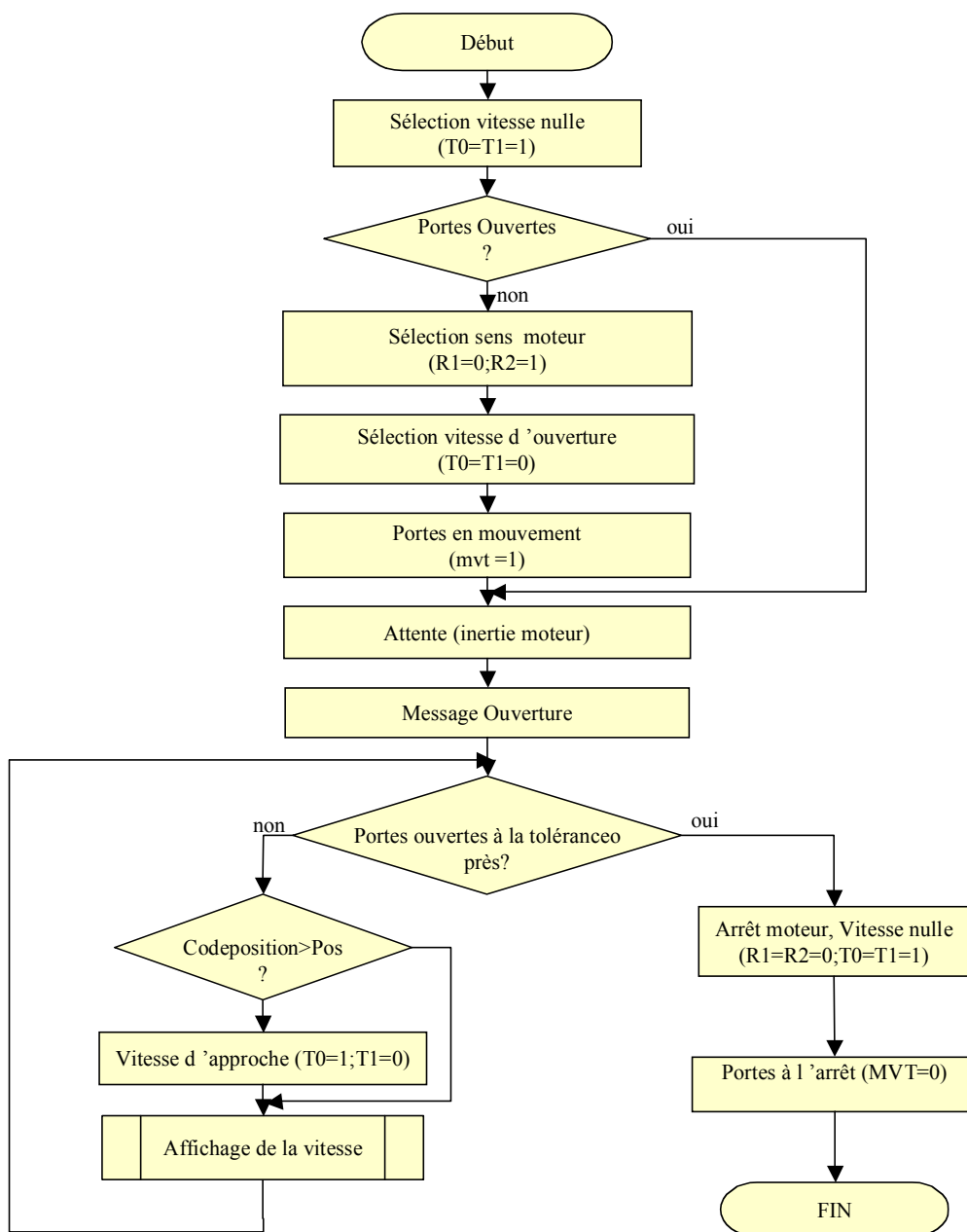


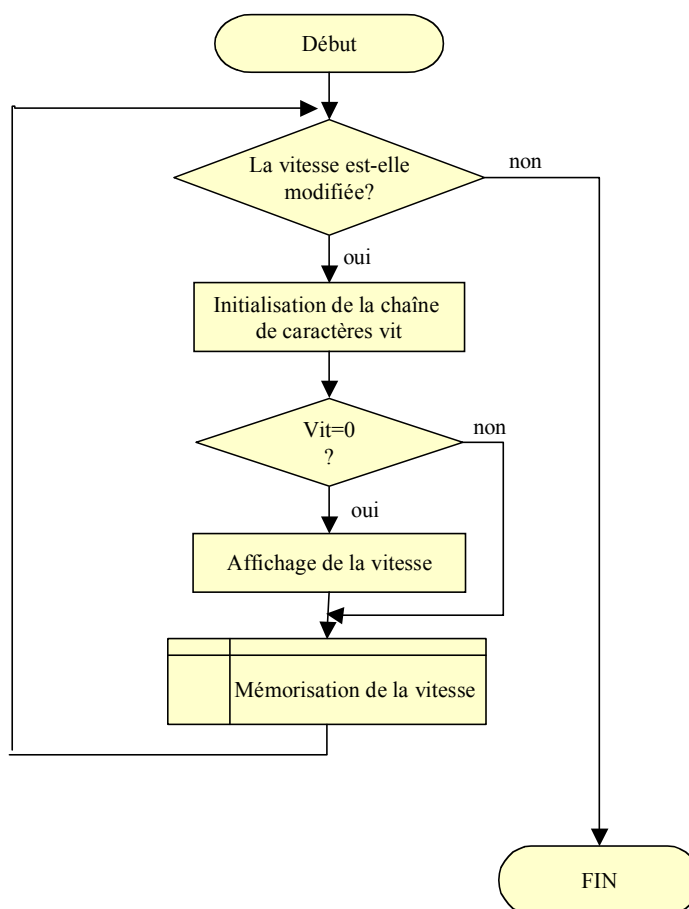
Algorithme de la fonction INITLCD

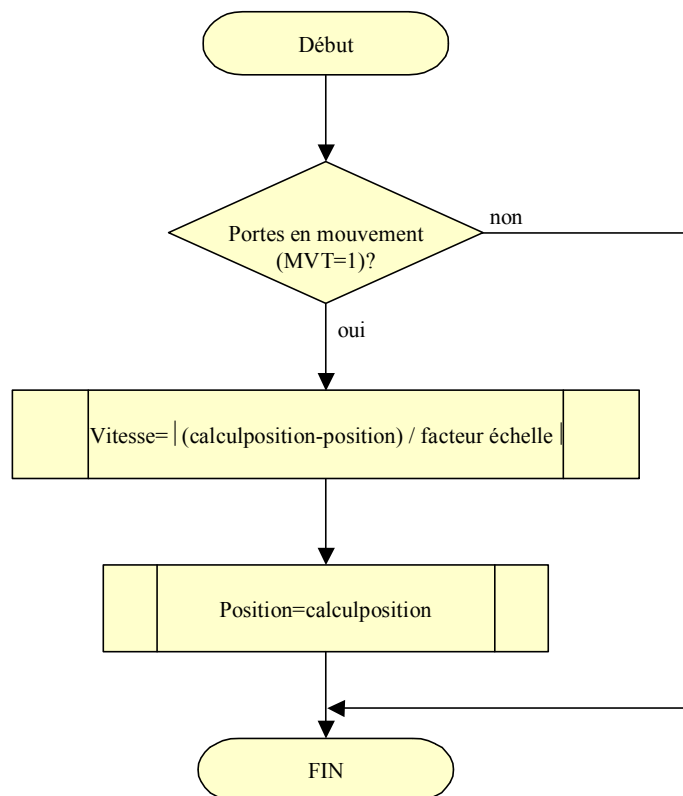
Algorithme de la fonction Apprentissage

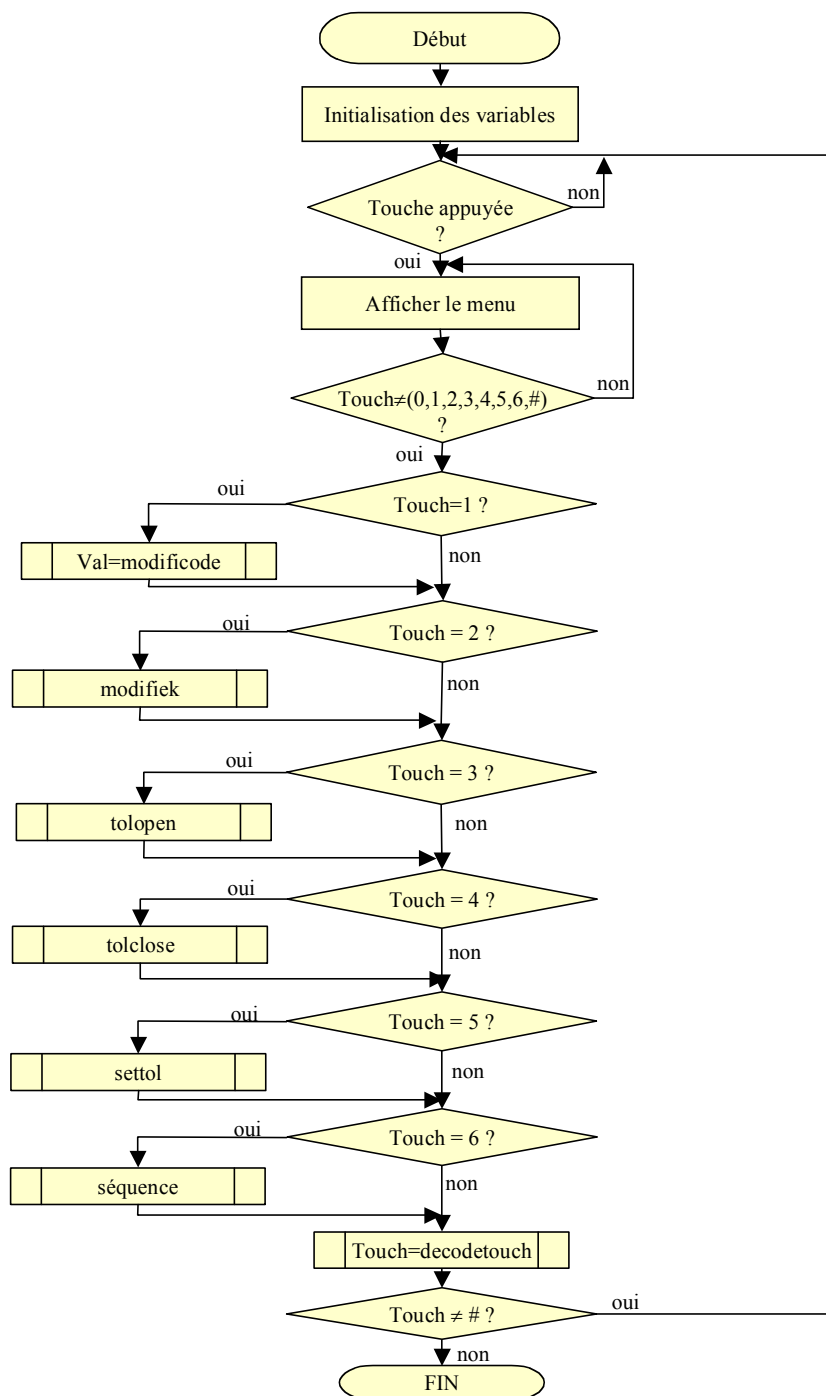
Algorithme de la fonction Calculposition

Algorithme de la fonction Séquence

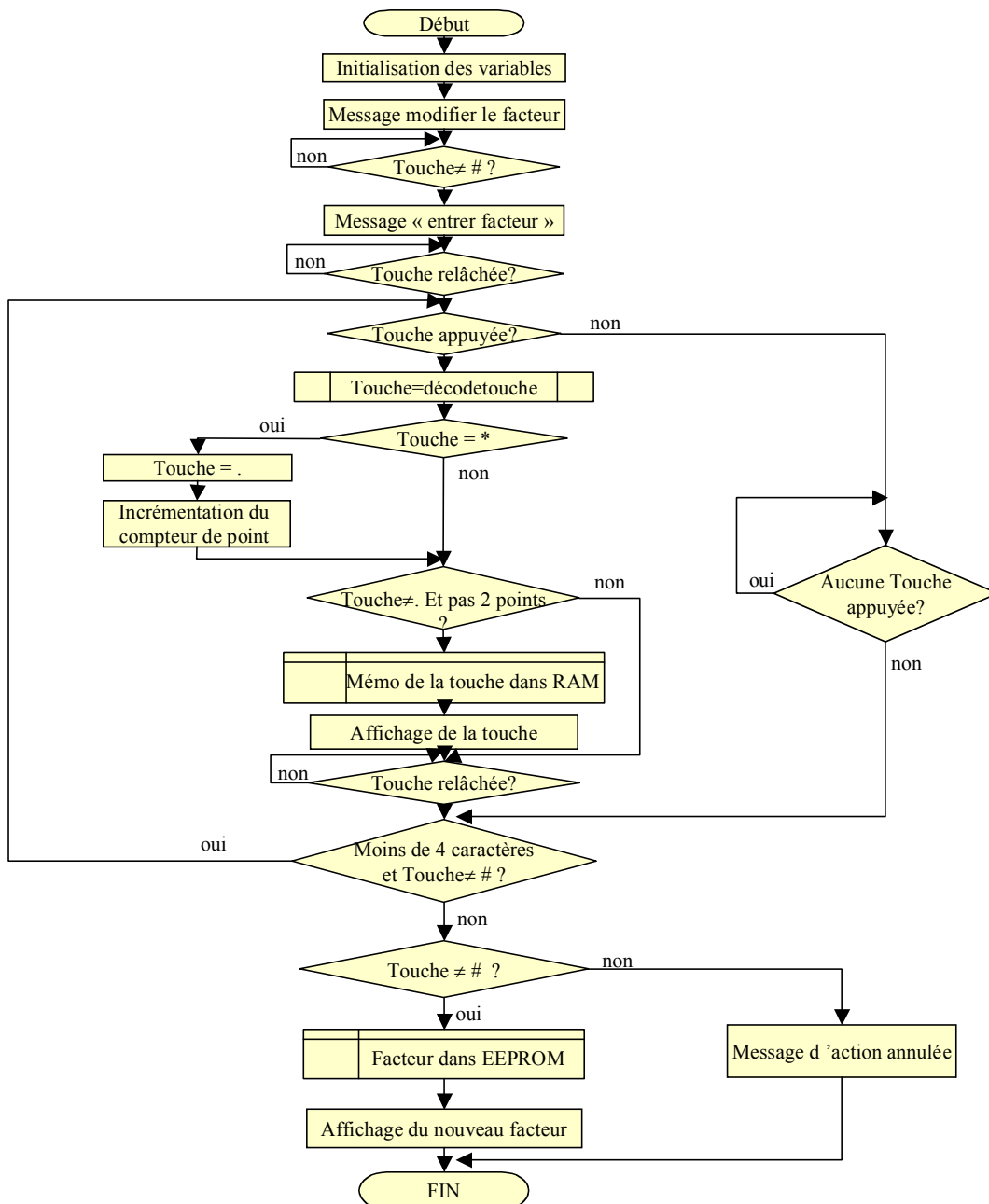
Algorithme de la fonction Ouverture

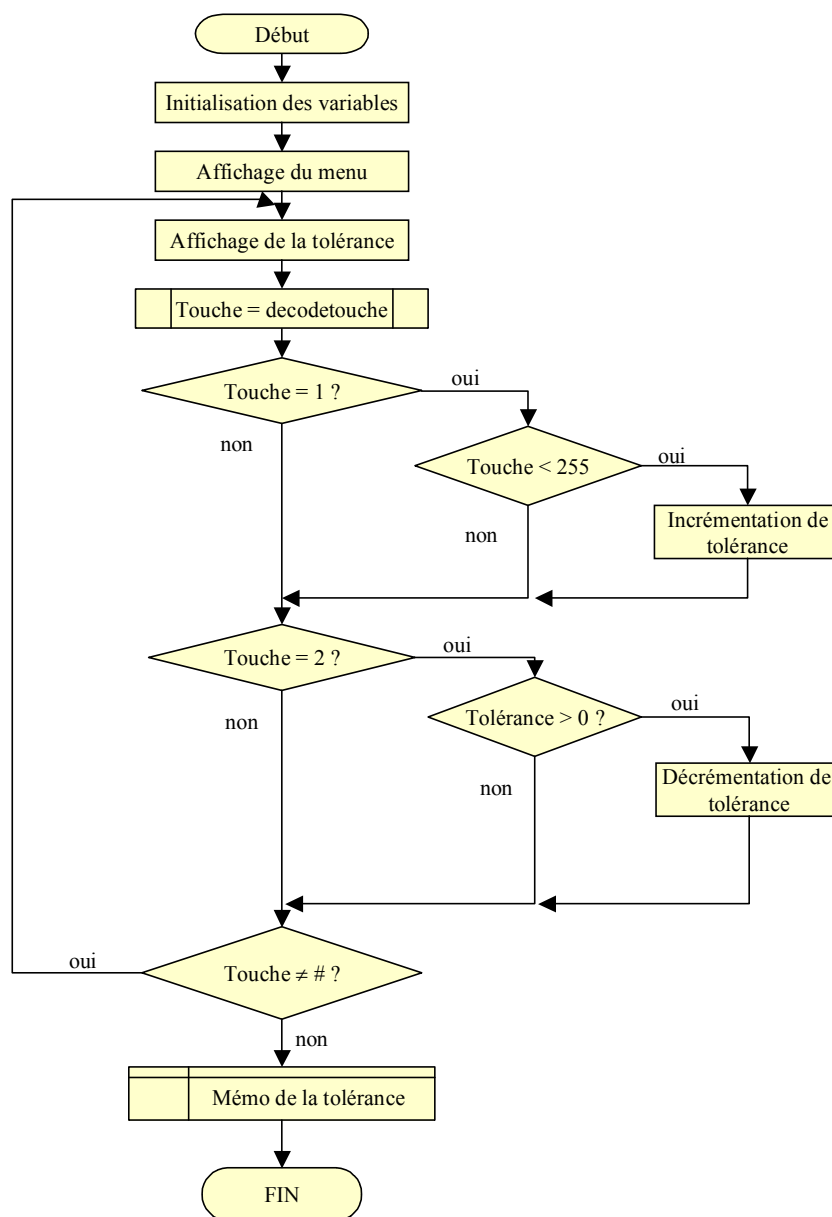
Algorithme de la fonction Affichevitesse

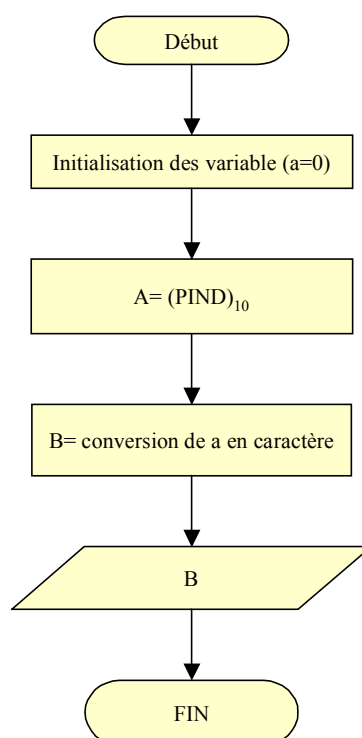
Algorithme de la fonction Calculvitesse

Algorithme de la fonction Action

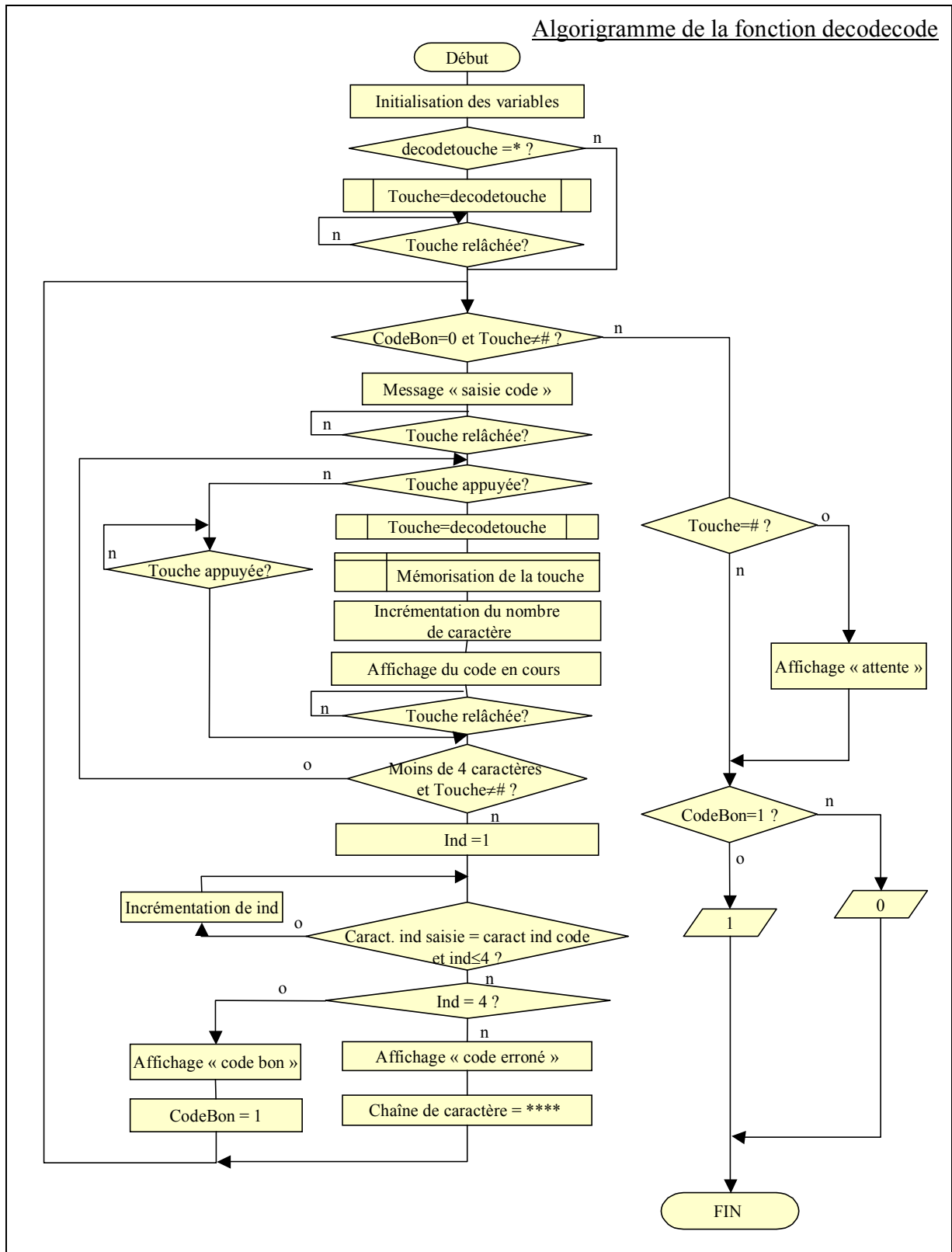
Algorithme de la fonction Modifiek



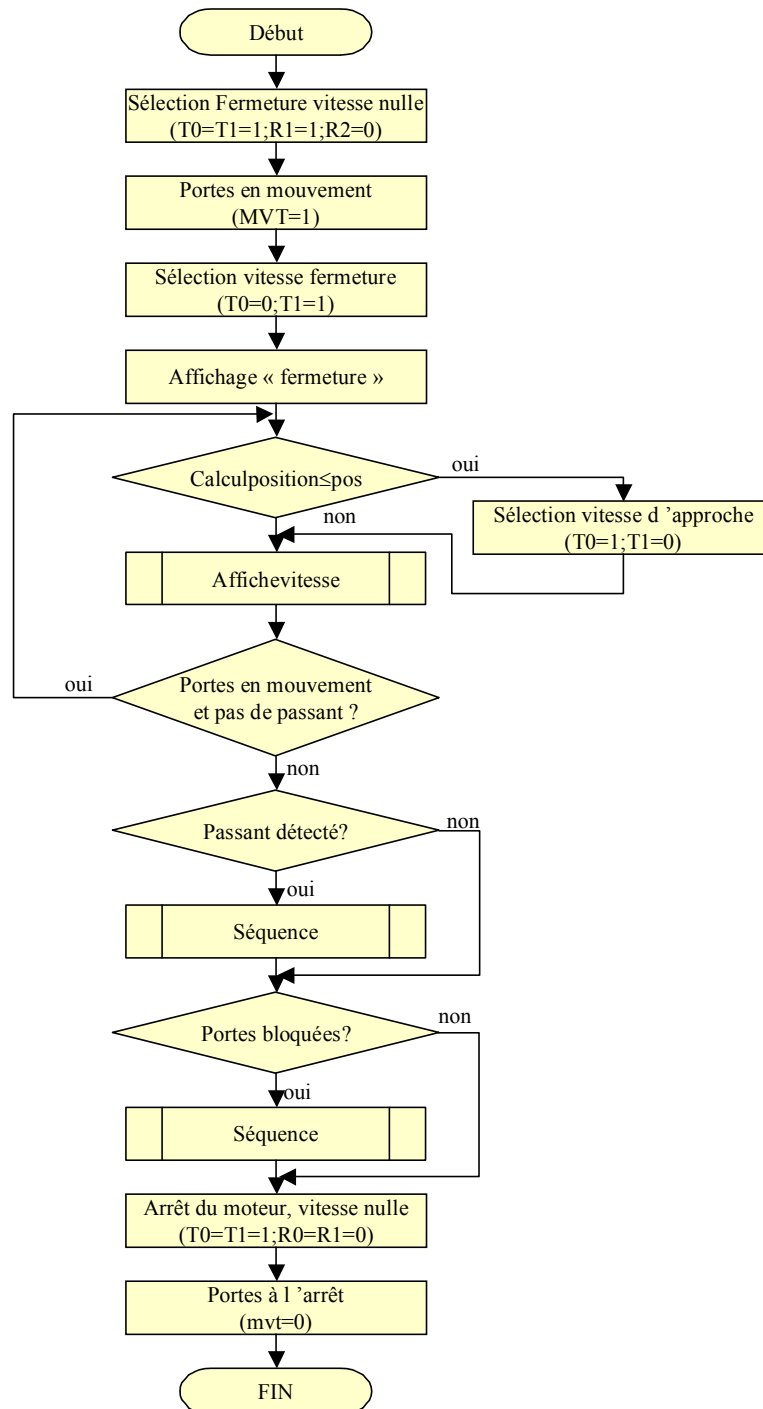
Algorithme de la fonction tolopen (tolclose et settol)

Algorithme de la fonction decodetouche

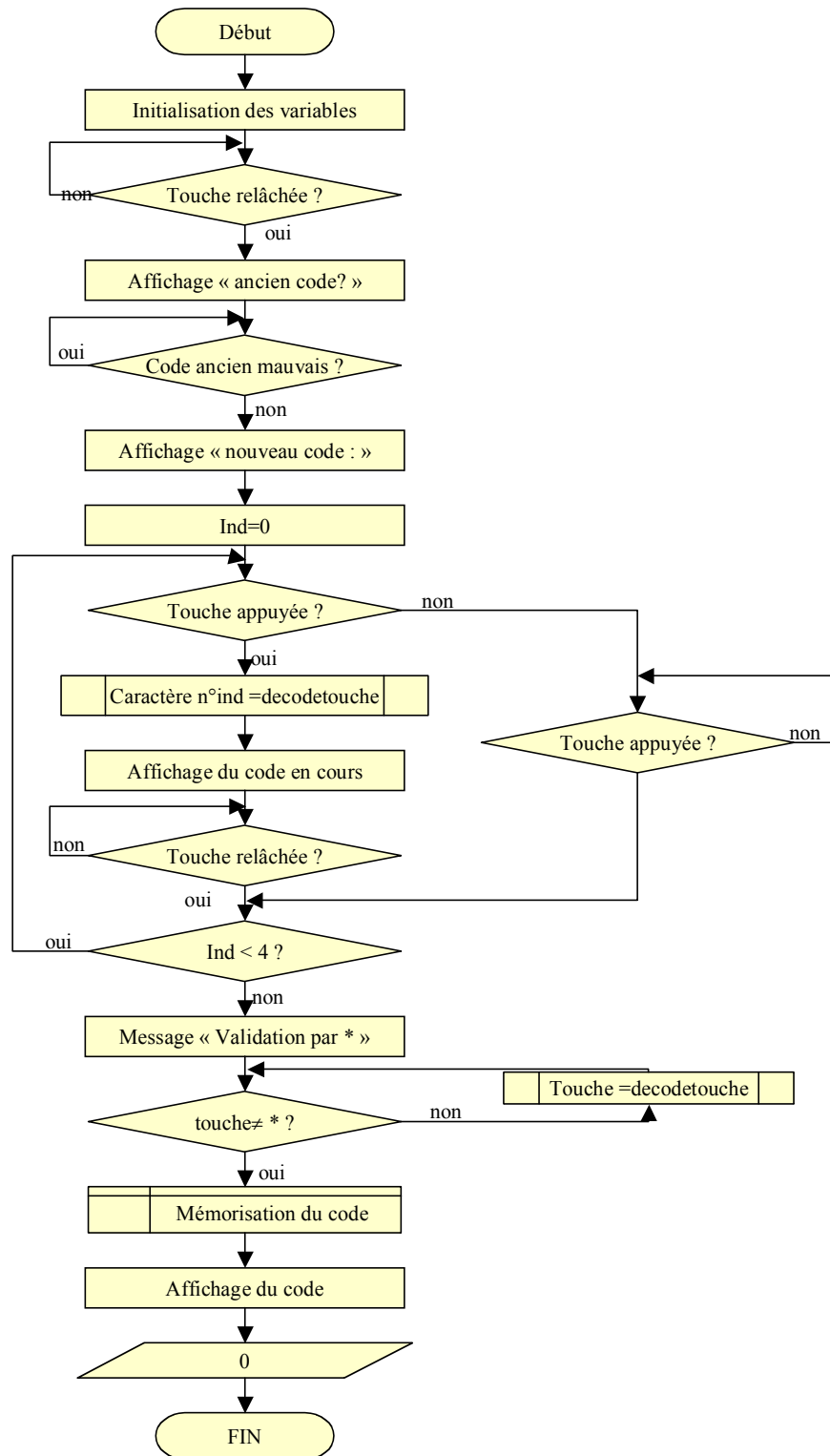
Algorithme de la fonction decodecode

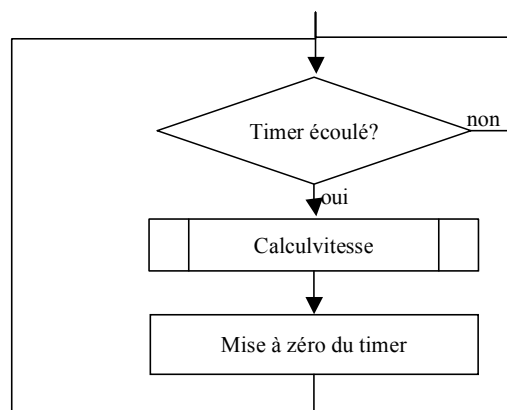
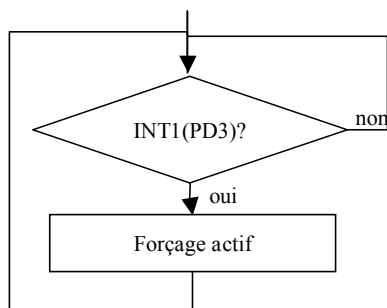
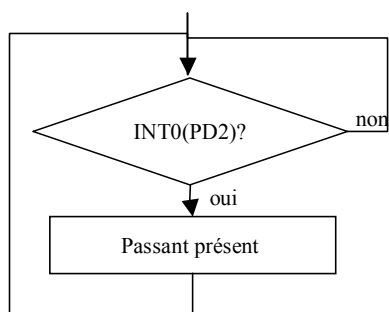


Algorithme de la fonction Fermeture



Algorithme de la fonction Modifiecode

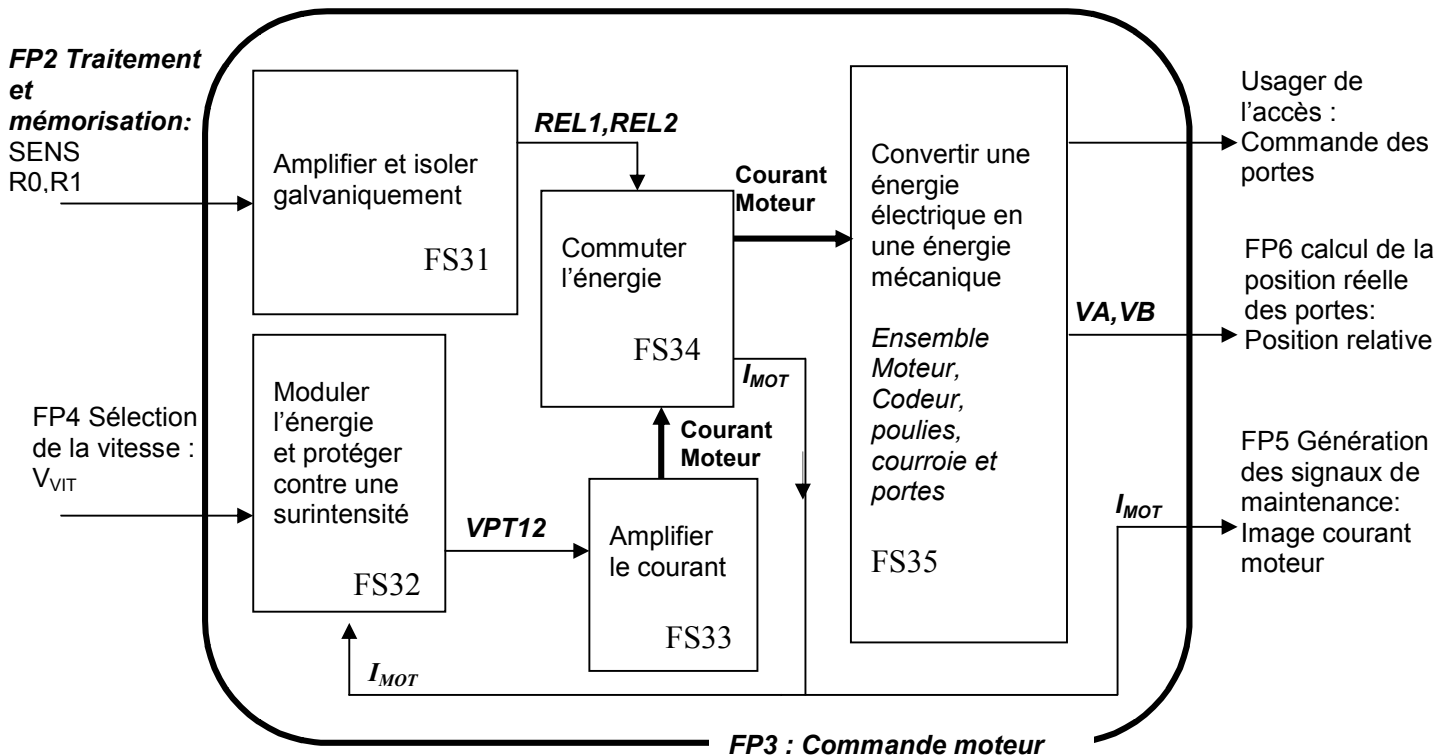


Interruptions et Timer

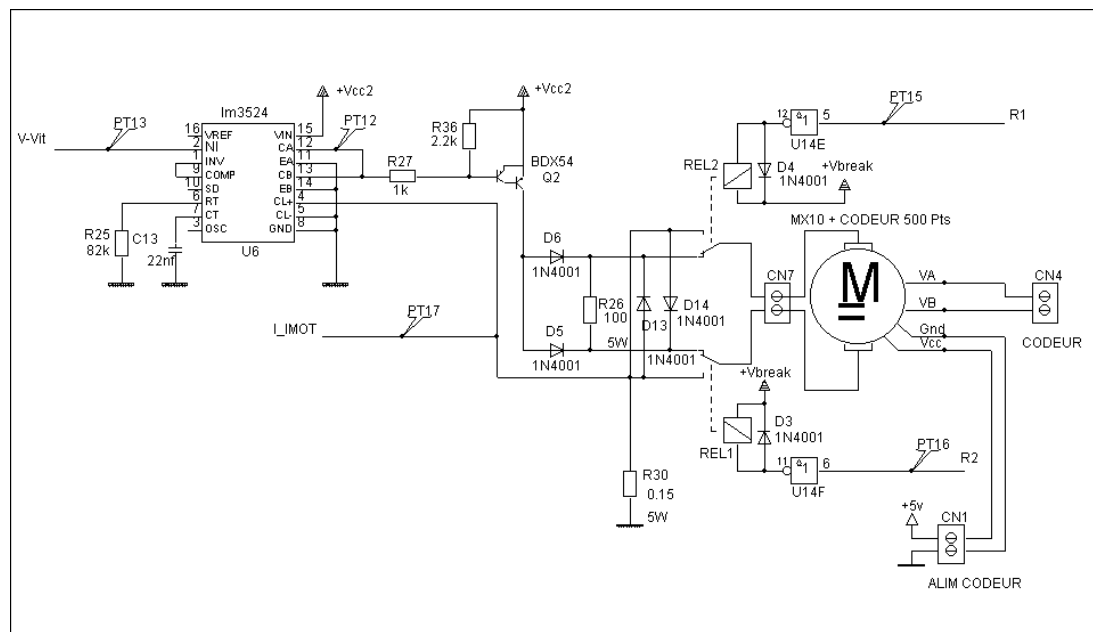
2-3 Étude de FP3 : commande du moteur

La fonction Commande moteur permet la rotation du moteur et le déplacement des portes en fonction des consignes de sens et de vitesse. Le moteur étant pourvu d'un codeur incrémental, la fonction fournit également une image de la position relative des portes.

Elle élabore également une tension image du courant permettant une sécurité contre les surintensités moteur.



- Schéma structurel de FP3 :



• Structure associée à FS31

Cette fonction permet de respecter l'isolation galvanique souhaitée entre la partie Traitement de l'information et la partie Commande moteur.

L'isolation galvanique des sorties PB0 et PB1 du microcontrôleur sont assurées par l'utilisation des relais REL1 et REL2. On remarquera que si Ubreak est absent, la commande de ces relais est inhibée.

La documentation Atmel page 94 indique que la somme des courants de sortie des ports PB3, PB2, PB1 et PB0 ne doit pas dépasser 100mA.

Or, la commande des leds des opto-coupleurs et la commande des relais (PB0 et PB1) imposent de fournir au moins 140mA (60mA de courant de maintien pour un relais et environ 10mA par led). Il est donc nécessaire de mettre des buffers sur chaque sortie de poids faibles du port B.

R1	R2	REL1	REL2	Tension aux bornes du moteur	Action du moteur
0	0	O	O	R26 en shunt	ARRÊT & FREINAGE
0	1	O	F	$+(V_{CC2}-V_{CESATQ2} - V_{D5})$	SENS1
1	0	F	O	$-(V_{CC2}-V_{CESATQ2} - V_{D6})$	SENS2
1	1	F	F	Non utilisé	

• Structure associée à FS32

Cette fonction permet de moduler la valeur moyenne du signal de sortie VPT12 en fonction de la tension analogique V_{VIT} . Le principe mis en œuvre consiste à faire varier le rapport cyclique θ d'un signal rectangulaire périodique.

La documentation du Lm3524 donne les valeurs suivantes du rapport cyclique :

- min = 0% pour $V_{VIT} = 1V$
- max = 90% pour $V_{VIT} = 3,5V$ (utilisation des 2 sorties collecteur ouvert CA et CB)

Il faudra veiller à régler les tensions de vitesses dans cette gamme de tensions.

Les sorties CA et CB de U6 étant à collecteur ouvert, sur une période :

- $V_{PT12} = V_{cesat\ max} = 2V$ pendant T_{on} (transistor saturé)
- $V_{PT12} = +V_{CC2}$ pendant T_{off} (transistor bloqué)

La valeur moyenne $V_{PT12MOY} = (V_{CC2} - V_{cesat}).(1-\theta)$ est directement proportionnelle au rapport cyclique. L'amplification en courant étant assurée par un transistor PNP, la tension aux bornes du moteur est directement de la forme $\theta.U_{MOT}$.

Sécurité contre une surintensité

Le retour de la tension VPT17 (IMOT) vers U6, permet de protéger le moteur en cas de surintensité. La mesure du courant VPT17 (IMOT) est assurée en insérant en série avec le moteur la résistance R30.

$$V_{PT17} = R_{30}. I_{MOT}$$

Si une surintensité est détectée le composant U6 arrête la M.L.I ($\theta = 0\%$) et $V_{PT12} = V_{CC2}$. Le moteur est mis hors tension jusqu'à disparition des surintensités.

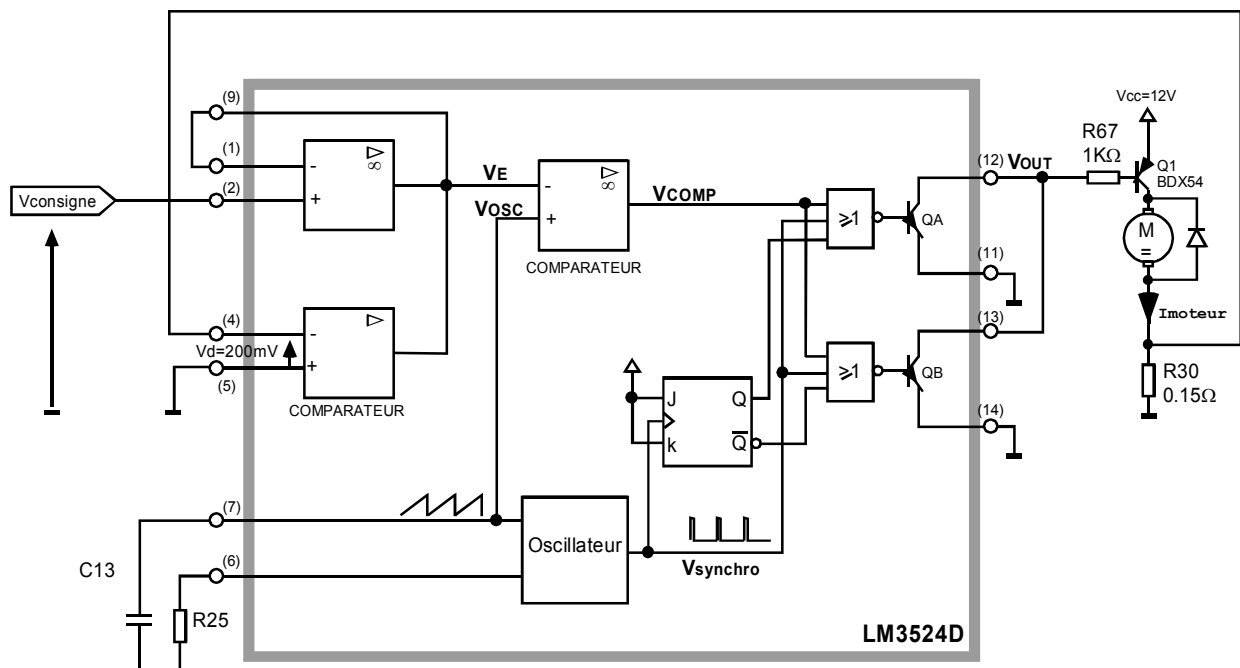
• Etude du circuit LM3524

La variation de vitesse du moteur est obtenue en générant un signal M.L.I. dont le rapport cyclique dépend de la tension V-Vit. Cette tension est appliquée sur l'entrée Vconsigne d'un LM3524D.

Ce circuit comporte un oscillateur intégré associé au condensateur C13 et à la résistance R25. La fréquence de hachage est de 554Hz. Une rampe est générée aux bornes du condensateur C13 qui se charge avec un courant constant de $3,6V/R25$. Elle varie donc de 1V à 3,6V. Le signal Vcomp est donc le résultat de la comparaison entre Vconsigne et Vosc. Cela permet d'obtenir un signal M.L.I..

De plus, ce circuit fait une régulation de courant. En effet la résistance shunt R30 permet d'obtenir une tension image du courant dans le moteur (hors phase de roue libre). Cette dernière est comparée avec une tension interne de 200mV.

Donc si le courant traversant le moteur devient supérieur à $200.103/0.15 = 1.3 \text{ A}$, le signal Vcomp est forcé à 1, les transistors Qa et Qb sont alors bloqués, et le moteur n'est plus alimenté.



La régulation de courant n'est qu'une protection en cas de panne logicielle. En effet, le programme gère le blocage des portes en analysant constamment leurs positions pendant les déplacements.

• Structure associée à FS33

Cette fonction permet l'amplification en courant de la sortie VPT12 pour l'alimentation du moteur.

L'amplification en courant est assurée par un transistor Darlington basse et moyenne tension Q2 de type PNP.

Celui-ci est saturé lorsque VPT12 = 0V. Dans ce cas, les potentiels des cathodes de D6 et D5 sont portés à $V_{cc2} - V_{cesat} - V_d$.

On donne pour le transistor BDX54

$$V_{cesat} = -2V \text{ max}$$

$$V_{besat} = -2,5V \text{ max}$$

$$\beta_{\min} = 750$$

Calcul du courant I_{bsatmin}

$$\begin{aligned} &= I_{R27\min} - I_{R36\max} \\ &= [(V_{cc2} + V_{\text{besat}}) / R27 + V_{\text{besat}} / R36] \\ &= 9,5\text{mA} - 1,1\text{mA} = 8,4 \text{ mA} \end{aligned}$$

Calcul du courant sans coefficient de sursaturation

$$I_{\text{csat max}} = \beta_{\min} \cdot I_{\text{bsatmin}} = 6,3 \text{ A (Courant limité à 8A pour le BDX54)}$$

Le courant maximal étant de 1,3A, nous obtenons dans ce cas un coefficient de sursaturation voisin de 5.

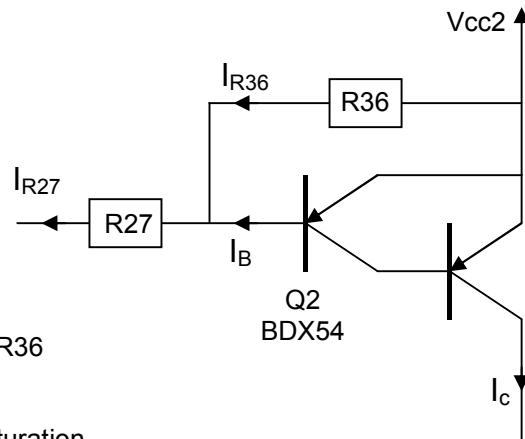
• Structure associée à FS34

Cette fonction permet la commutation de l'énergie aux bornes du moteur pour sa mise en rotation.

En fonction des commandes relais, le moteur peut tourner dans un sens 1, tourner dans un sens 2 ou bien est relié à la résistance de freinage R26 jusqu'à l'arrêt

La sélection arrêt moteur étant réalisée par les commandes $R1 = R2 = 0$, le cas où les deux relais sont commandés ne figure pas dans notre système.

Le moteur est protégé contre les surtensions à l'extinction du courant par les diodes de roues libres D13 et D14.



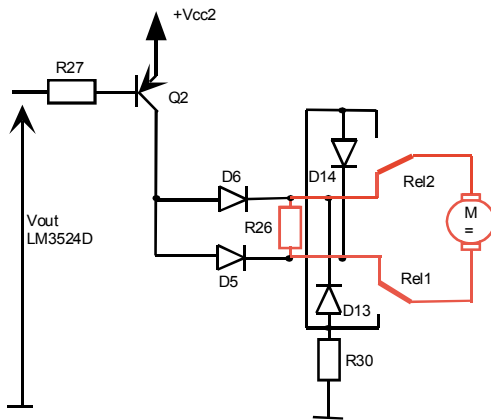
R1	R2	REL1	REL2	Action du moteur	Action des portes	Codeur incrémental
0	0	O	O	FREINAGE ARRÊT	FREINAGE ARRÊT	Pas d'impulsion
0	1	O	F	SENS1	Fermeture	Vb en avance de phase sur Va
1	0	F	O	SENS2	Ouverture	Va en avance de phase sur Vb
1	1	F	F	X	X	X

Le fonctionnement du moteur est régi par le tableau ci-dessous :

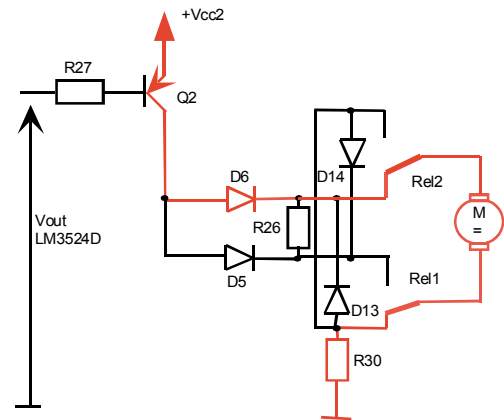
Etat	Rel1	Rel2	Q2	Moteur
Etat 0	repos	repos	X	Non alimenté : arrêt avec «frein électrique »
Etat 1	travail	repos	saturé	Alimenté : ouverture
Etat 2	travail	repos	bloqué	En roue libre avec D13 : « inertie » de l'ouverture
Etat 3	repos	Travail	Saturé	Alimenté : fermeture
Etat 4	repos	travail	bloqué	En roue libre avec D14 : « inertie » de la fermeture
Etat 5	travail	travail	x	Non alimenté : arrêt, état jamais activé

• **Schémas de circulation du courant :**

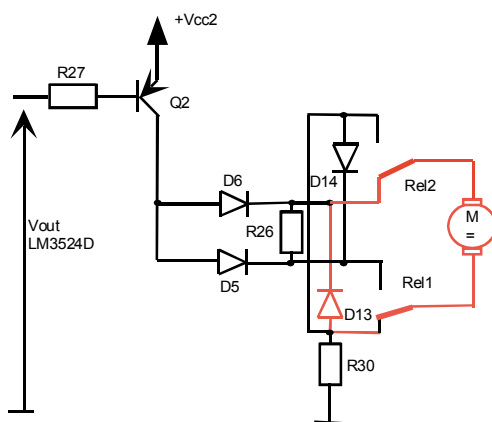
Etat 0 : le moteur a est connecté sur la résistance de freinage R26 : il devient générateur.



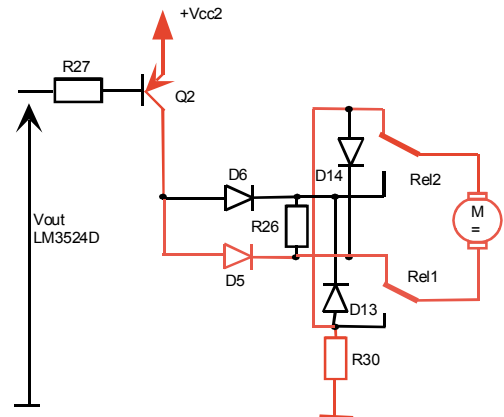
Etat 1 : La diode D6 est passante, les diodes D5, D13 et D14 sont bloqués. Le moteur est alimenté.



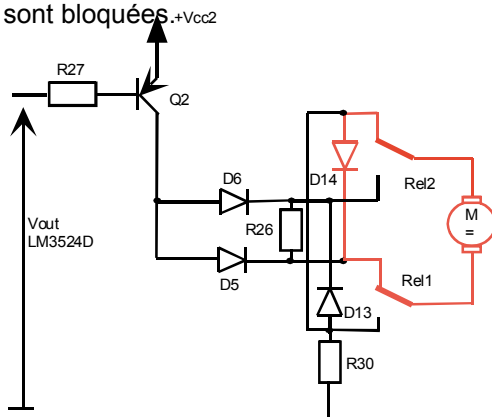
Etat 2 : Le transistor Q2 se bloque ; le courant du moteur continue de circuler à travers la diode D13. Les autres diodes sont bloquées. Aucun courant ne traverse la résistance shunt.



Etat 3 : La diode D5 est passante, les diodes D6, D13 et D14 sont bloqués. Le moteur est alimenté.



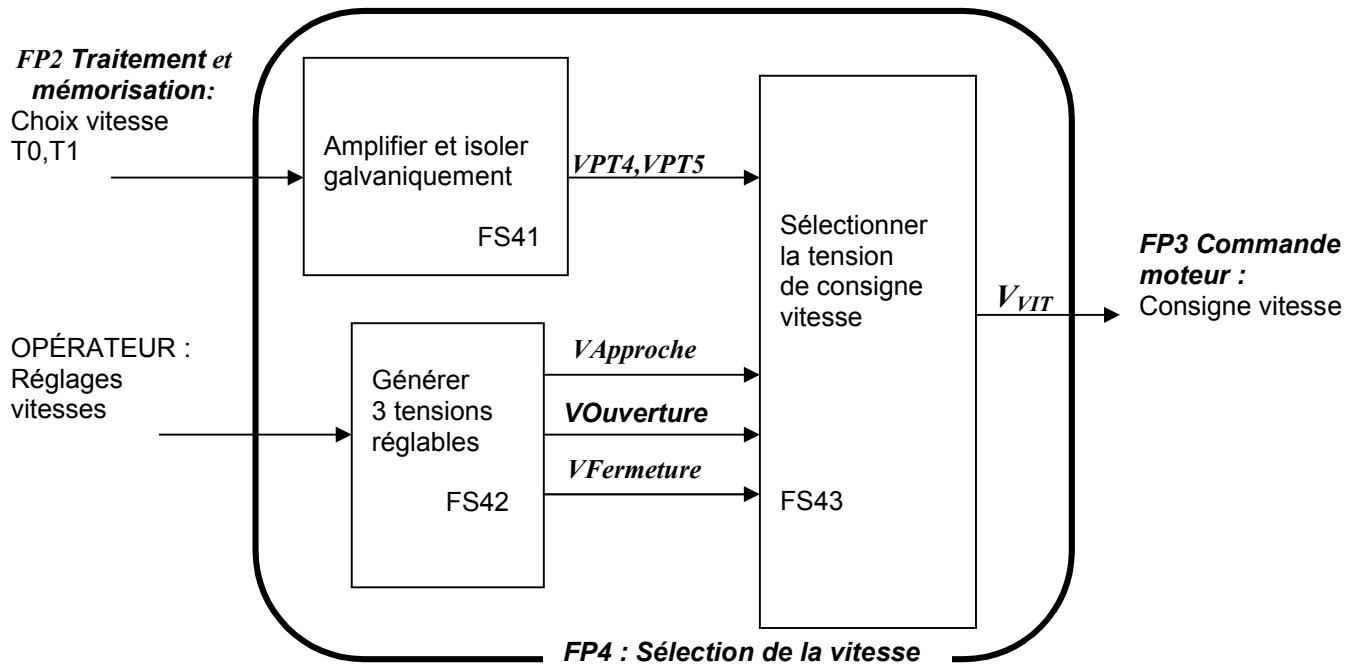
Etat 4 : Le transistor Q2 se bloque ; le courant du moteur continue de circuler à travers la diode D14. Les autres diodes sont bloquées.



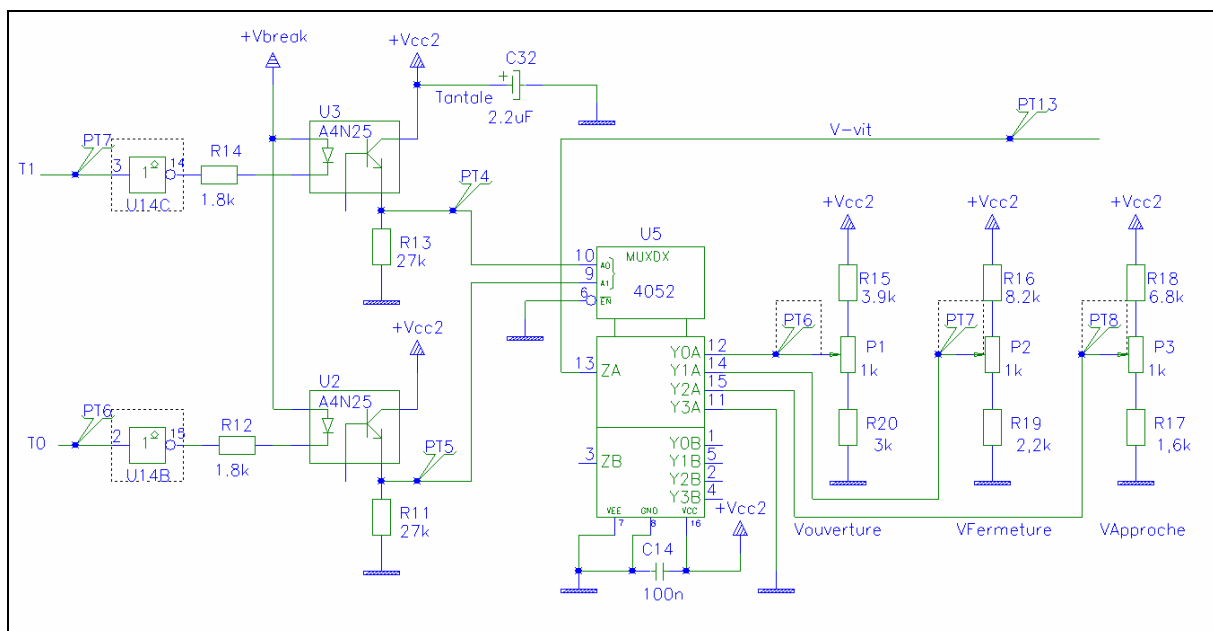
Etat 5 : Dans ce cas le moteur est court-circuité. Lors d'un fonctionnement normal, cet état est inexistant.

2-4 Étude de FP4 : sélection de la vitesse

La fonction Sélection de la vitesse permet d'élaborer un signal de commande du moteur en fonction de la consigne de vitesse sélectionnée par FP2. Cette consigne est inhibée en cas d'absence de la tension Ubreak



- Schéma structurel de FP4 :



- **Structure associée à FS41**

Cette fonction permet de respecter l'isolation galvanique souhaitée entre la partie Traitement de l'information et commande moteur.

On désire isoler galvaniquement la partie commande de la partie puissance afin d'éviter la perturbation des tensions d'alimentation côté commande dues au moteur.
Il est donc nécessaire d'introduire des opto-coupleurs comme interface.

La documentation Atmel page 94 indique que la somme des courants de sortie des ports PB3, PB2, PB1 et PB0 ne doit pas dépasser 100mA.

Or, la commande des leds des opto-coupleurs et la commande des relais (PB0 et PB1) imposent de fournir au moins 140mA (60mA de courant de maintien pour un relais et environ 10mA par led). Il est donc nécessaire de mettre des buffers sur chaque sortie de poids faibles du port B.

Etude des opto-coupleurs :

Si $T_n = 0$, le transistor de sortie de U14 est bloqué, la led est éteinte et le transistor de l'opto-coupleur est bloqué. La résistance R11 (ou R13) impose un niveau logique 0 sur PT5 (ou PT4).

Si $T_n = 1$, le transistor de sortie de U14 est saturé et on considère son $V_{CEsat} = 0$. Le courant dans la led est donc de :

$$I_f = (V_{break} - V_{ledmax}) / R_{12} ;$$

$$I_f = (12 - 1,5) / 1800 = 5,8 \text{mA} ;$$

Le transistor de l'opto-coupleur est donc saturé et le courant du collecteur est :

$$I_c = (V_{cc2} - V_{cesat}) / R_{11} ;$$

$$I_c = (12 - 0,5) / 27000 = 425 \mu\text{A} ;$$

Le coefficient de transfert étant de 20%, il est nécessaire d'avoir

$$I_f > I_c / \text{CTR} ;$$

Or $I_c / \text{CTR} = 2.13 \text{mA}$; I_f est environ 3 fois supérieur à I_c / CTR .

Le transistor de l'opto-coupleur est donc correctement saturé.

D'autre part, on remarque que les leds des opto-coupleurs sont alimentées par la tension V_{break} . Si le capteur de panique est activé les leds ne seront plus alimentées. Cela imposera des niveaux logiques bas sur les entrées d'adresse de U5.

- **Structure associée à FS42**

Calcul de la gamme de tension possible pour $V_{Ouverture}$ ($V_{Fermeture}$ ou $V_{Approche}$)

Le courant de l'entrée NI (broche 2) du circuit LM3524 est $I_{bias\ current} = 1 \mu\text{A}$ (entrée FP3 Commande moteur).

Le courant traversant la maille R15, P1 et R20 est de 3mA.

On peut donc considérer le montage comme un pont diviseur.

$$V_{min} = (R_{20} \times V_{cc2}) / (R_{20} + P1 + R_{15})$$

$$V_{max} = (P1 + R_{20}) \times V_{cc2} / (R_{20} + P1 + R_{15})$$

$V_{Ouverture}$ varie entre 3,8 V et 5 V.

$V_{Fermeture}$ varie entre 3 V et 4 V.

$V_{Approche}$ varie entre 2 V et 3,2 V.

Voila les valeurs théoriques pour les résistances :

	Vitesse de fermeture	Vitesse d'ouverture	Vitesse d'approche
R15	8000 Ω	3823 Ω	7331 Ω
R20	3000 Ω	2235 Ω	1666 Ω

- **Structure associée à FS43**

Etude de U5 : 4052

Il s'agit d'un multiplexeur analogique qui permet de sélectionner la vitesse de mouvement des portes en fonction de PT4 et PT5 soit de T1 et T0.

T1	T0	Led u2	Led u3	Tu2	Tu3	PT4	PT5	ZA	V-Vit
0	0	E	E	B	B	0	0	Y0	Vapproche
0	1	E	A	B	S	0	1	Y1	Vfermeture
1	0	A	E	S	T	1	0	Y2	Vouverture
1	1	A	A	S	S	1	1	Y3	0V

La résistance R_{on} du circuit 4052 est de 1050 Ω maximum. La chute de tension maximale induite par cette résistance interne sera donc de :

$$U_{chute} = R_{on} \times I_{bias \text{ current}}$$

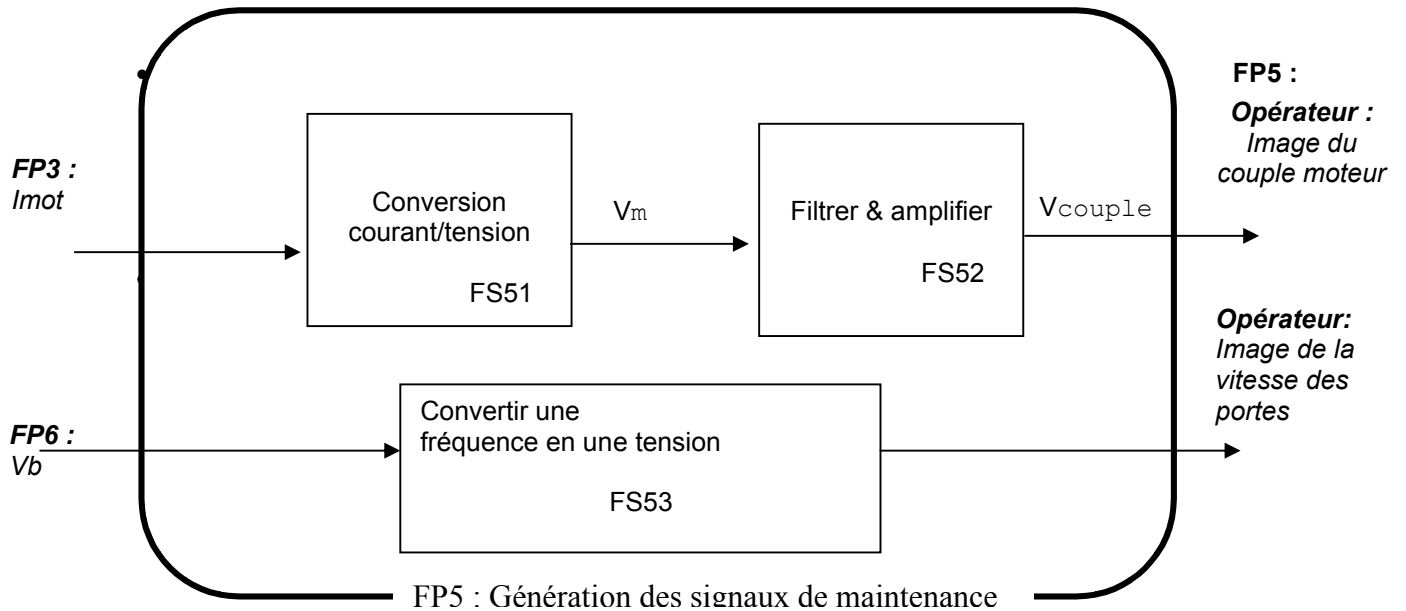
$$U_{chute} = 1,05mV$$

Cette chute de tension est négligeable devant la gamme de tensions présentes aux entrées du multiplexeur. On considérera donc que V-VIT sera directement connectée à l'entrée sélectionnée par les adresses.

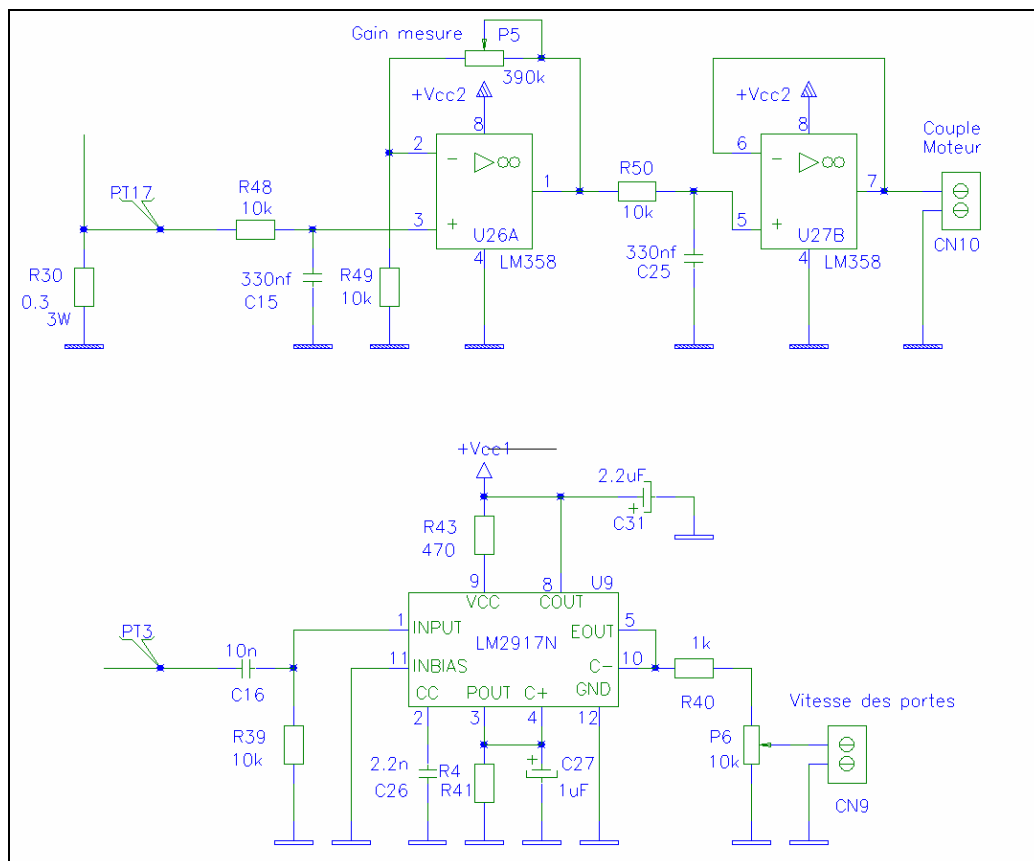
2-5 Étude de FP5 : Génération des signaux de maintenance

La fonction Génération des signaux de mise en fonctionnement permet de fournir deux sorties de types analogiques images de la vitesse de déplacement des portes et du courant moteur.

Ces signaux sont disponibles aux personnels de maintenance lors de la mise en fonctionnement.



- Schéma structurel de FP5



Cette structure permet d'obtenir une tension image du courant consommé par le moteur donc du couple moteur.

En effet le couple $T = K_T \times I_{\text{moteur}}$ avec $K_T=21.8 \text{ Ncm/A}$ pour le moteur MX10

- **Structure associée à FS51**

La conversion courant/tension est réalisée à l'aide de la résistance R30.
La tension à ses bornes $V_m = R_{30} \times I_{\text{moteur}}$ soit

$$V_m = 0.3 \times I_{\text{moteur}}$$

- **Structure associée à FS52**

La tension V_m est amplifiée afin d'obtenir une tension V_{couple} de 0.5V/Ncm.

Détermination de l'amplification :

$$\begin{aligned} \text{Donc } V_{\text{couple}} &= V_m \times A \\ \text{Soit } V_{\text{couple}} &= (0.3 \times I_{\text{moteur}}) \times A \end{aligned}$$

$$\text{D'où } V_{\text{couple}} = 0.3 * \frac{T}{K_T} * A$$

$$\text{Si } \frac{V_{\text{couple}}}{T} = 0.5 = \frac{0.3}{21.8} * A \Rightarrow A = 36.3$$

Il faudra donc pouvoir obtenir un gain de 36.3 sur l'étage amplificateur

$$\text{On choisit alors } A = \left(\frac{P5}{R49} + 1 \right) = 40$$

$$\text{Avec } R49 = 10\text{K}\Omega \rightarrow P5 = 39 \times R49 = 39 \times 10\text{K} = 390 \text{ K}\Omega$$

Détermination du filtrage :

La tension V_m est filtrée par un passe-bas du 2ème ordre réalisé à l'aide de 2 passe bas identiques du 1er ordre.

La fréquence de coupure doit être comprise entre la fréquence de coupure mécanique (pour pouvoir suivre l'évolution du couple en minimisant la perte d'information), et la fréquence de hachage de la MLI (pour pouvoir observer un courant moyen sur une période du hacheur).

La fréquence de coupure mécanique de l'ensemble (moteur + portes) $F_{\text{méca}} = 1.3\text{Hz}$ (identification par réponse indicielle en vitesse).

La fréquence de hachage $F_h = 554 \text{ Hz}$

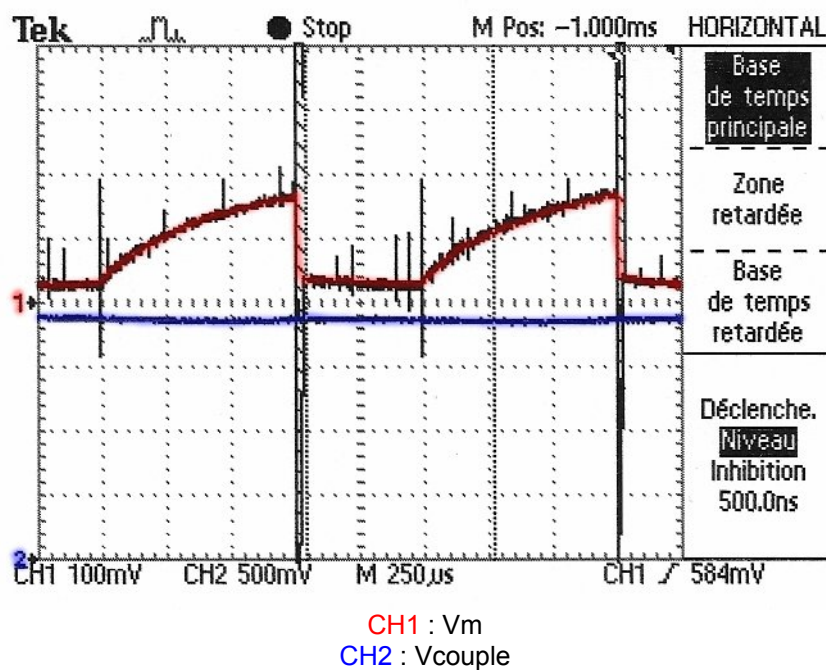
On choisi ici $F_{\text{méca}} < F_c < F_h/10$ soit $F_c = 55 \text{ Hz}$.

Calcul de C15 et C25 :

$$F_c = \frac{1}{\tau} = \frac{1}{R1.C1} = \frac{1}{R50.C25} \quad \text{On choisit } R50=R48 = 10\text{K}\Omega$$

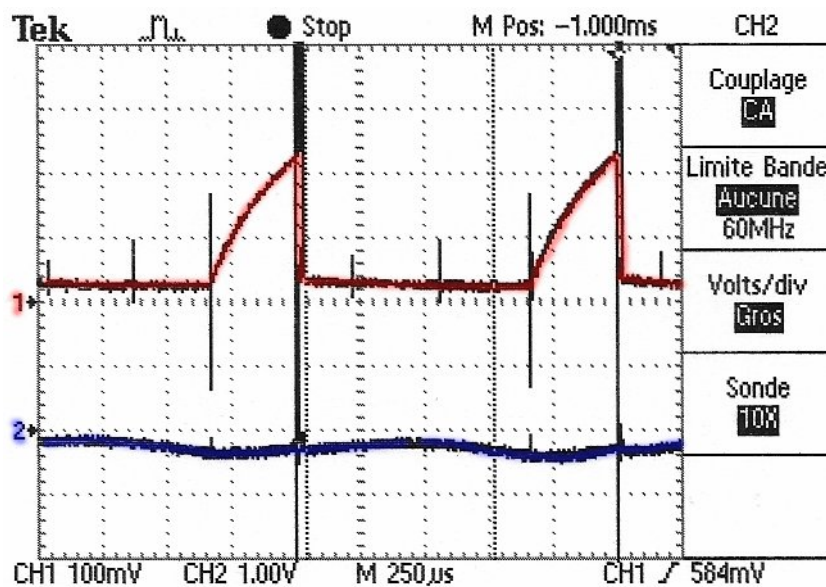
Ce qui implique :

$$C1 = C2 = \frac{1}{F_c.R50} \quad \text{AN : } C1=C2=330\text{nF}$$

Relevé expérimentale de V_m et V_{couple} :

Relevé expérimental de l'ondulation

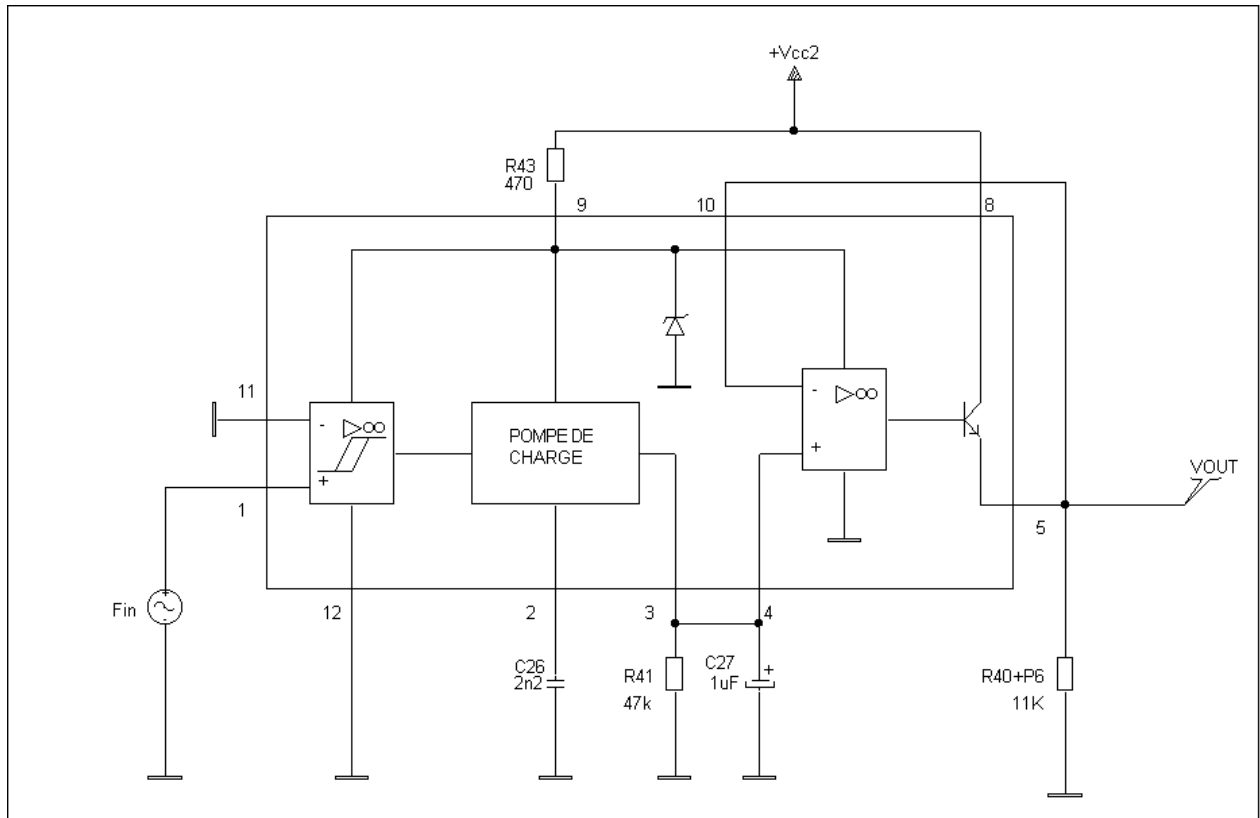
La voie 2 (V_{couple}) est en position AC afin de mesurer l'efficacité du filtrage



- **Structure associée à FS53**

La structure se repose sur le circuit intégré LM2917 qui à l'aide des composants passifs externes réalise une conversion fréquence-tension. La tension de sortie est proportionnelle à la fréquence issue du codeur incrémental, soit proportionnelle à la vitesse de déplacement des portes.

Schéma synoptique du LM2917 et composants externes passifs :



* D'après la documentation du constructeur, la régulation zener est intégrée uniquement dans les composants de la série LM2917. Il suffit de déconnecter la zener pour obtenir le synoptique des composants de la série LM2907.

Le premier étage est un comparateur à hystérésis non inverseur.

Le seuil V- étant reliée à la masse, la tension d'entrée doit osciller de part et d'autre le 0V de la tension threshold +/- VTH pour provoquer un basculement de la sortie.

+/- VTH en entrée est donnée à +/- 25 mV.

Le second étage constitue un circuit à pompe de charge. Il nécessite :

- une capacité de timing C26
- une résistance de charge R41
- une capacité de filtrage (ou lissage) C27

Quand l'entrée de ce montage change d'état, la capacité de timing C26 se charge ou se décharge à courant constant entre deux seuils de tension dont la différence vaut $V_{cc}/2$.

Sur une demi période de la fréquence d'entrée FIN, la charge ΔQ accumulée ou restituée est $\Delta Q = 0,5.C26.V_{cc}$.

Sur une demi période soit $T = 2 / f_{IN}$, le courant moyen I_{Cmoy} à travers C26 vaut $\Delta Q / T$,
Soit $I_{Cmoy} = \Delta Q \cdot f_{IN} / 2$

$I_{Cmoy} = V_{cc} \cdot f_{IN} \cdot C_{26}$ et est proportionnel à la fréquence d'entrée.

La sortie de cet étage se comporte comme une source de courant miroir du courant de IC sur la résistance de charge R41.

On obtient alors une tension V_O de sortie = $R_{41} \cdot I_C$

Soit $V_{Omoy} = K \cdot R_{41} \cdot V_{cc} \cdot f_{IN} \cdot C_{26}$. avec la filtrage de la capacité C27. K est la constante de gain du composant égal à 1 pouvant varier sur une plage maxi $0,9 < K < 1,1$.

Le dernier étage est un étage suiveur associé à un transistor ayant pour rôle d'amplificateur de courant.

On retrouve à cette sortie une source de tension $V_{OUT} = R_{41} \cdot V_{cc} \cdot f_{IN} \cdot C_{26}$.

$V_{OUT} = f_{IN} \cdot V_{cc} \cdot R_{41} \cdot C_{26}$

Pour $f_{min} = 0 \text{ Hz}$, on a $V_{OUT} = 0 \text{ V}$.

D'après la documentation constructeur, la valeur de V_{cc} vaut 7,56V avec la régulation de tension de la diode zener interne au composant (LM2917).

Pour $f_{max} = f_{IN} = 1 / R_{41} \cdot C_{26}$, soit $f = 9,67 \text{ KHz}$ on a $V_{OUT} = V_{cc}$. (théorique)
(cf doc. Constructeur : on suppose ici que $I_2 = V_{cc} / R_1$ ce qui est vrai dans l'objet technique $7,56 / 41000 = 184 \mu A$)

Il est possible de mesurer la valeur du courant I_2 d'après le chronogramme de UC26 en page suivante en appliquant la relation $\Delta Q = I \cdot T = C \cdot V$

Exploitation des chronogrammes de la page suivante :

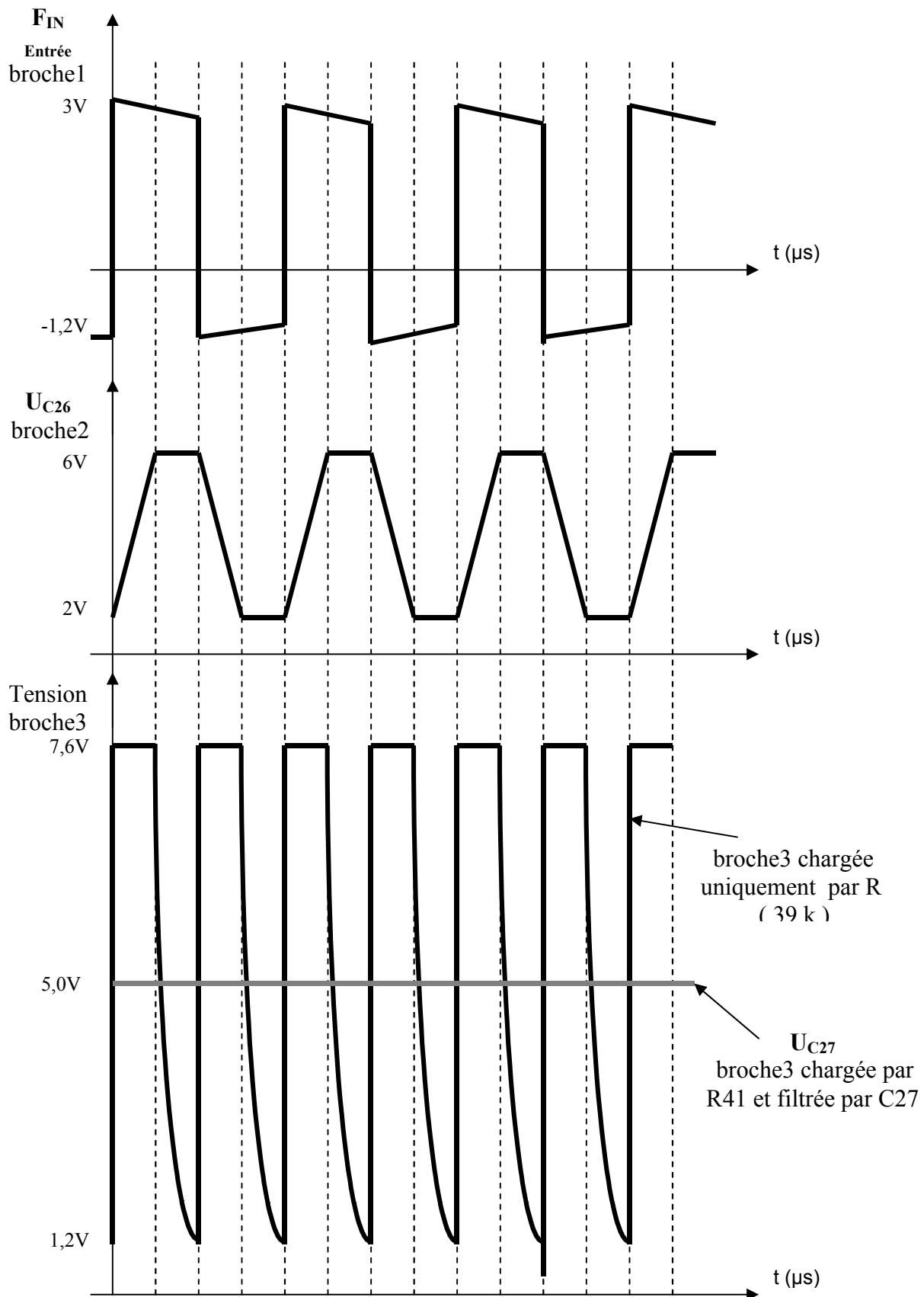
Pour $f_{IN} = 6,2 \text{ KHz}$ on obtient une tension de sortie $V_{out} = 5V$.

La relation théorique pour cette fréquence nous donne $V_{out} = V_{cc} \cdot (6,3 / 9,67) = 4,92 \text{ V}$

La mesure du courant I_2 donne $I = C \cdot V / T = 220 \mu A$ avec $V = 4V$ et $T = 40 \mu s$.

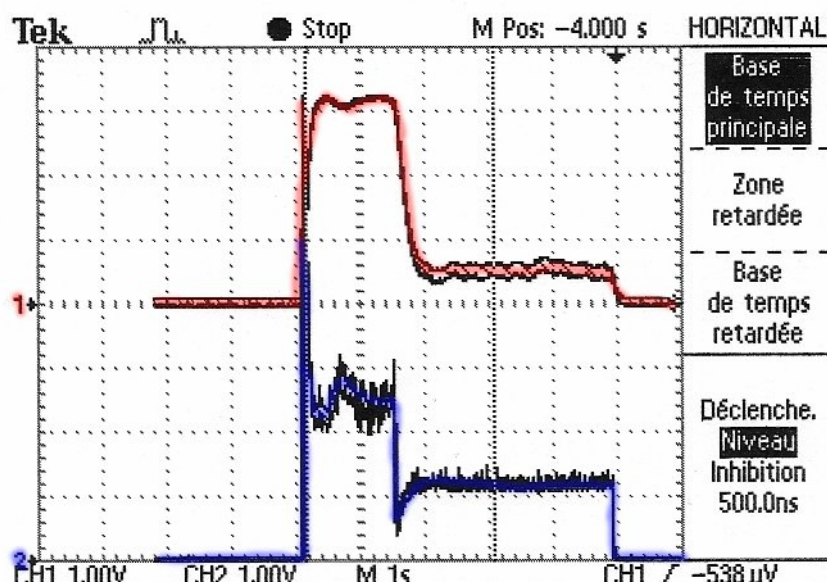
- Relevés expérimentaux des signaux de fonctionnement du LM2917

En régime établi, $F_{IN} = 6,3 \text{ kHz}$, période $T_{in} = 160 \mu\text{s}$ environ.



- Exemples de relevés de couple et de vitesse :

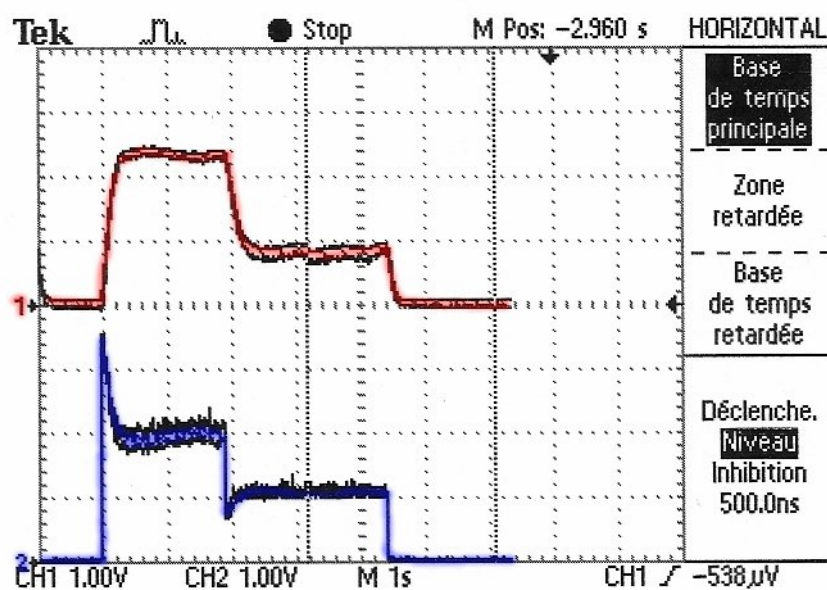
Cycle d'ouverture



CH1 : Vitesse des portes

CH2 : Courant moteur – échelle : 0.5V = Ncm

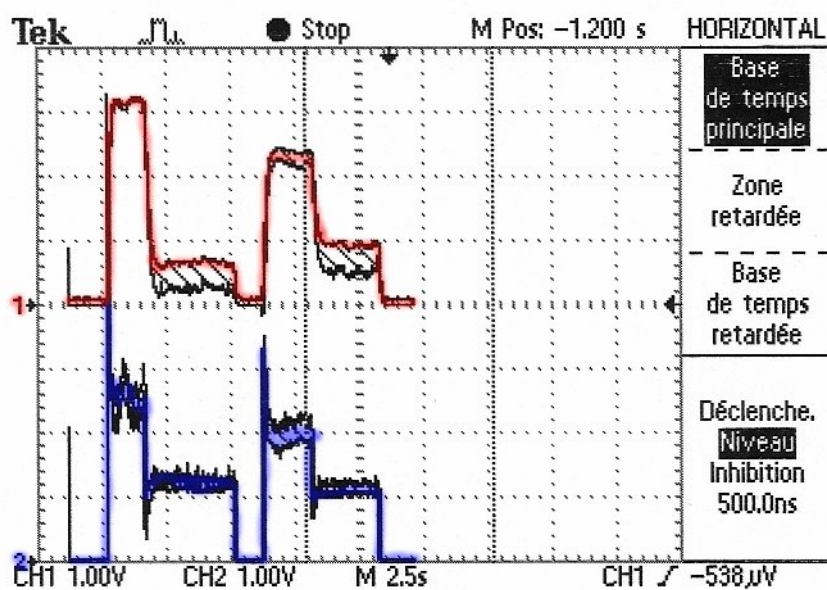
Cycle de fermeture



CH1 : Vitesse des portes

CH2 : Courant moteur – échelle : 0.5V = Ncm

Cycle complet

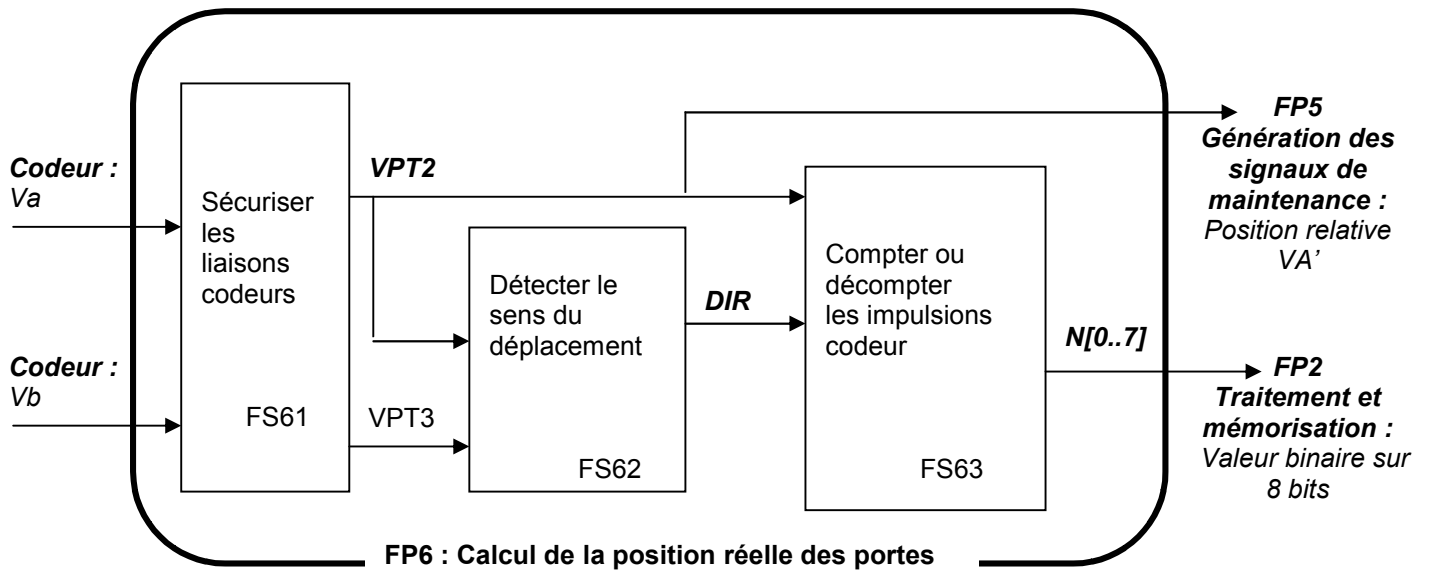


CH1 : Vitesse des portes

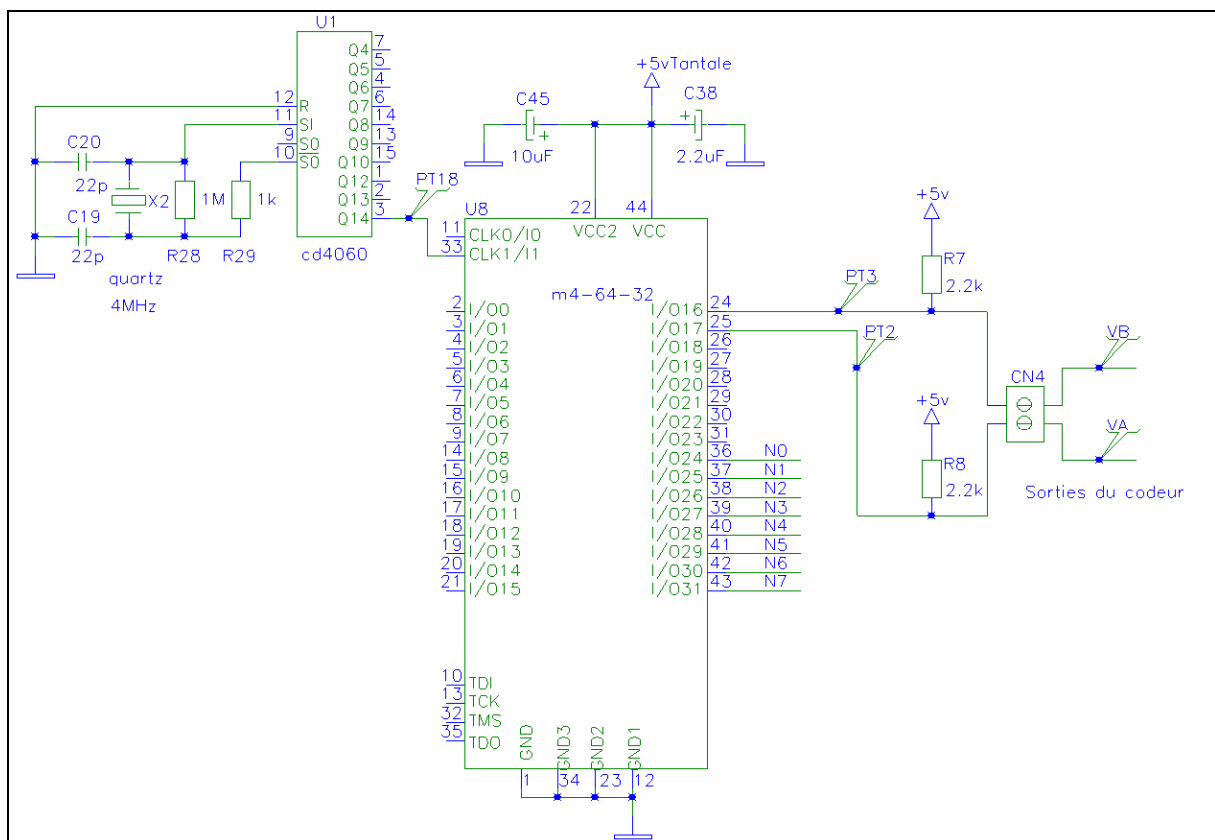
CH2 : Courant moteur – échelle : 0.5V = Ncm

2-6 Étude de FP6 : calcul de la position réelle des portes

La fonction Calcul de la position réelle des portes permet de fournir à la fonction FP7 Traitement et mémorisation une représentation numérique binaire de la position des portes.



- Schéma structurel de FP6 :



La fonction est principalement assurée par le circuit logique programmable U8 Mach4 - 64/32. L'intérêt de ce PLD est de pouvoir réaliser plusieurs fonctions logiques diverses avec un seul composant :

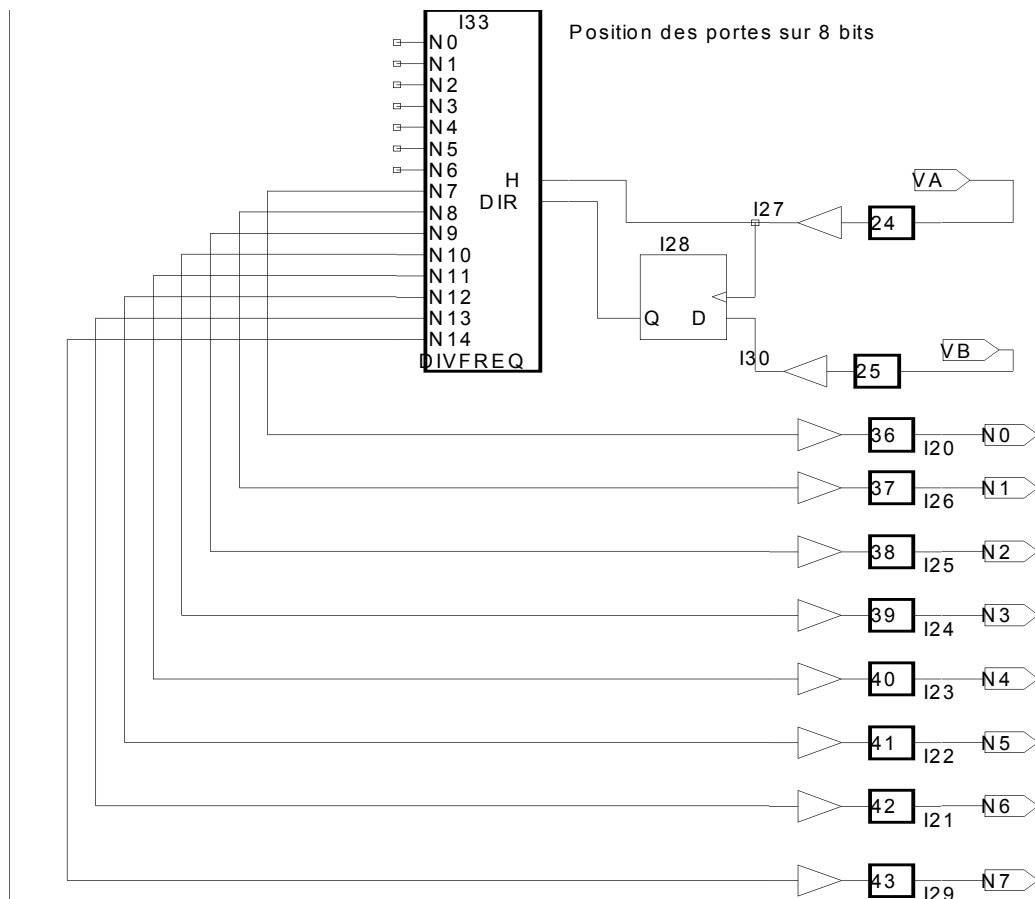
- un compteur-décompteur 15 étages
- une bascule D

D'autres structures logiques sont programmées dans le PLD pour réaliser la fonction FP7 Saisie des consignes (détection touche appuyée, encodage et décodage clavier).

Le PLD remplace plusieurs circuits intégrés. Il en résulte une réduction du coût de fabrication et une simplification de la structure.

D'autre part le contenu du PLD peut être facilement modifié in situ. Cela offre des possibilités peu coûteuses d'évolution ou de modification des fonctions assurées par le composant programmable.

Voici le schéma structurel logique interne aux Mach4 assurant les fonctions FS62 et FS63

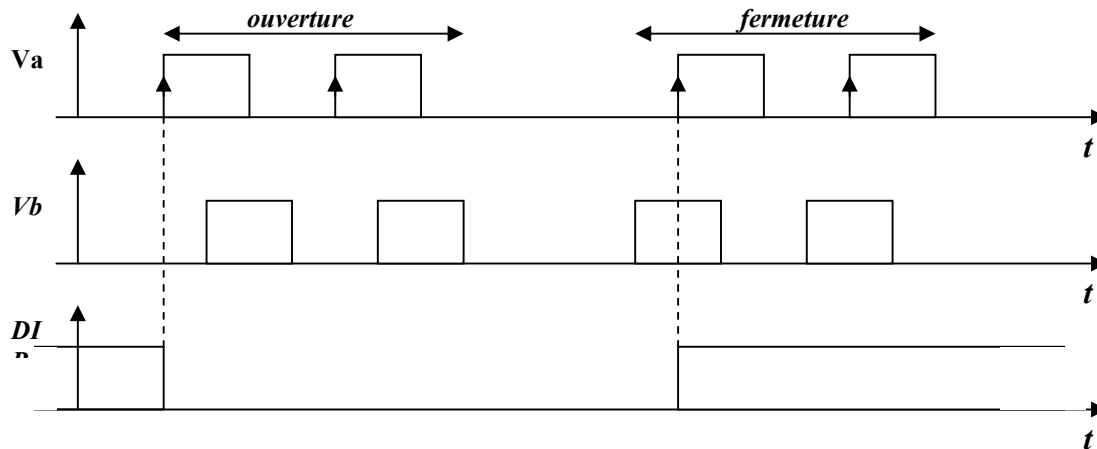


- **Structure associée à FS61**

Elle est constituée des résistances de tirage R7 et R8. Ces structures assurent un niveau logique haut aux entrées du Mach4 lorsque les sorties codeur ne sont pas reliées à la carte de commande.

- **Structure associée à FS62**

La bascule D I28 permet grâce aux sorties en quadrature de phases fournies par le codeur, de détecter le sens de déplacement des portes.



I28 élabore le signal binaire DIR pour le comptage ou décomptage des impulsions codeur.

- **Structure associée à FS63**

Elle est basée sur un compteur-décompteur à 15 sorties noté I33.

En fonction du niveau logique présent sur l'entrée DIR, les impulsions codeur sont comptées ou décomptées de façon à obtenir une valeur binaire en sortie image de la position réelle ou instantanée des portes.

Le codeur fournit 512 impulsions/tour. La poulie a un diamètre de 64,68 mm, les portes se déplacent donc d'une distance de $\pi \cdot D = 203,2$ mm.

Avec un rapport de réduction de 1/7, le moteur effectue 7 tours pour un tour de poulie.

Le nombre d'impulsions qui incrémentent compteur pendant un tour de poulie est égal à $7 \cdot 512 = 3584$

On obtient alors une résolution codeur de $203,2 / 3584 = 57 \mu\text{m}$ (résolution du bit N0)

En utilisant la sortie Nn comme LSB, la résolution est multipliée par 2^n .

En utilisant la sortie N7 comme LSB du nombre de sortie N, on obtient alors une résolution de 7,26mm.

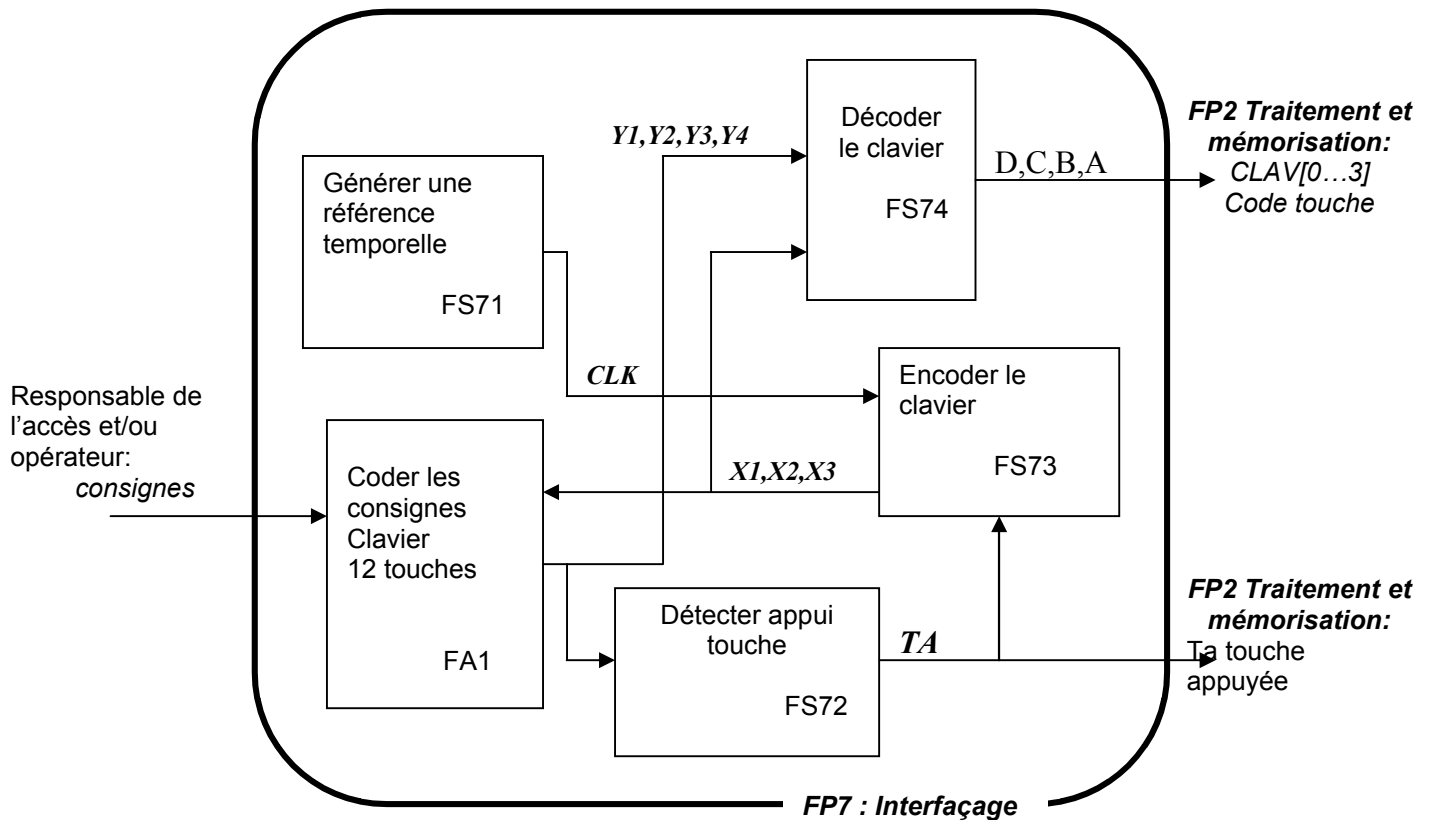
Cette information numérique est transmise à FP2 Traitement et mémorisation.

Programme associé à la fonction FS63 Compter ou Décompter :

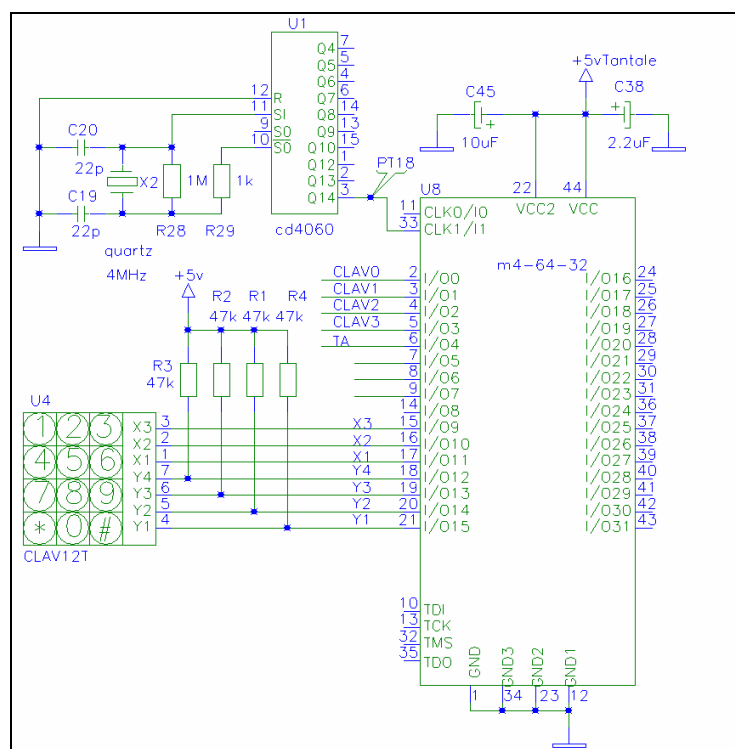
```
* MODULE divfreq
*
* DECLARATIONS;
*
*   N14,N13,N12,N11,N10,N9,N8,N7,N6,N5,N4,N3,N2,N1,N0 pin istype 'reg';
*   H,DIR pin;
*
*   position = [N14 .. N0];
*
* EQUATIONS;
*
*   position.clk = H;
*   when DIR==0 then position := (position + 1);
*   else {position := (position - 1)};
*
* END
```

2-7 Étude de FP7 : Saisie des consignes

La fonction Saisie des consignes permet la communication entre l'objet technique (Fp2 traitement et mémorisation) et le milieu extérieur (milieu humain).



- Schéma structurel de FP7 :



- **Structure associée à FA1,**

Cette fonction permet de communiquer à la partie traitement de l'information différentes consignes. Elle est assurée par un clavier 12 touches, dont les fonctions dépendent de la position dans le menu déroulant proposé par le logiciel (Cf Algorithme de fonctionnement).

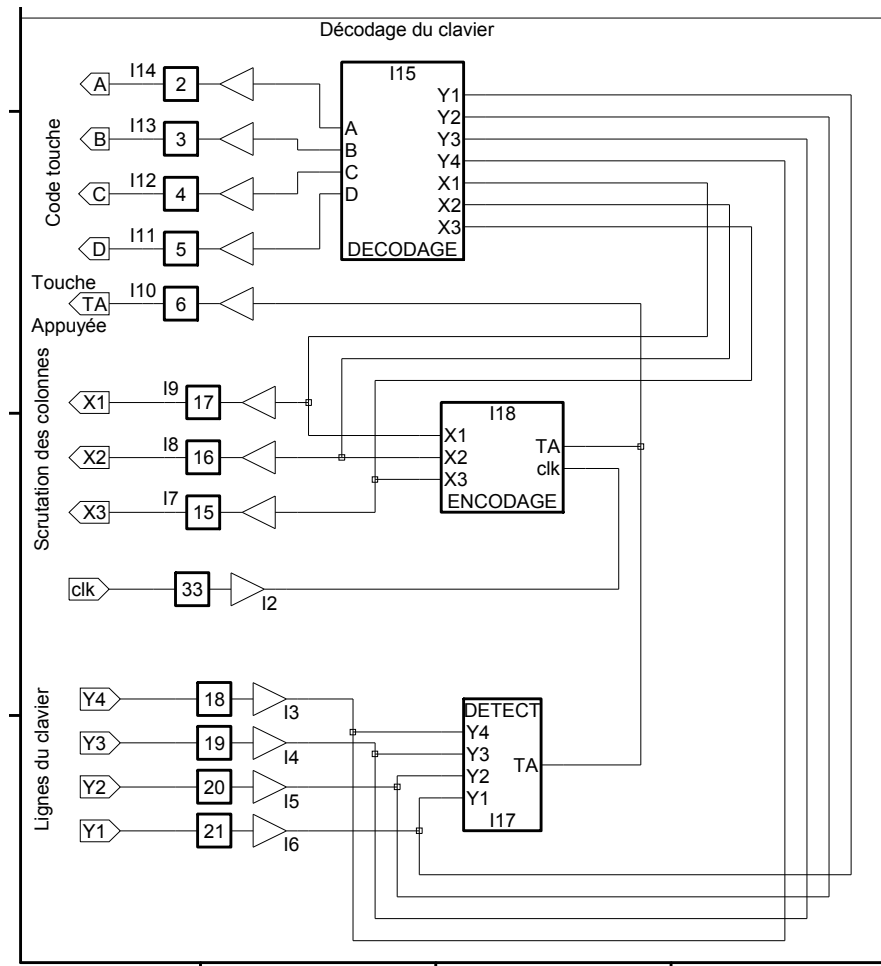
- **Structure associée à FS71**

Cette fonction réalisée autour du diviseur de fréquence U1 et du quartz X2 à 4 MHz permet d'obtenir une référence temporelle nécessaire au bon fonctionnement de la fonction FS74 Encodage clavier. En sortie Q14 on obtient une fréquence pour le signal CLK d'environ 244 Hz.

- **Structure associée à FS72, FS73 et FS74**

Les fonctions sont assurées à nouveaux par le circuit logique programmable U8 Mach4 - 64/32.

Le PLD est programmé de façon à obtenir la structure interne suivante :



D'autres structures logiques sont programmées dans le PLD pour réaliser la fonction FP6 Calcul de la position réelle des portes (compteur-décompteur 15 étages, une bascule D)

- **Structure associée à FS73 Détection touche appuyée**

Le sous-ensemble réalisant la détection de l'appui d'une touche est noté DETECT.

Description des signaux d'entrée :

Y1, Y2, Y3 et Y4 : signaux indiquant l'état de chacune des 4 lignes du clavier

Description du signal de sortie :

TA : signal actif à l'état haut indiquant qu'une touche est appuyée.

Programme associé à la fonction Détection touche

```
* MODULE detect
*
* TITLE 'Détection ligne active'
*
* "entrées
*   Y1,Y2,Y3,Y4 pin;
* "sorties
*   TA pin istype 'com';
*
* EQUATIONS
*
* !TA=Y1&Y2&Y3&Y4;
*
* END
```

- **Structure associée à FS74 Encodage clavier**

Le sous-ensemble réalisant la détection de l'appui d'une touche est noté ENCODAGE

Description des signaux d'entrée :

TA : signal actif à l'état haut indiquant qu'une touche est appuyée.

CLK : signal d'horloge issue de FS71 servant à la scrutation des colonnes

Description des signaux de sortie :

X1, X2 et X3 : signaux de scrutation des trois colonnes, actifs aux niveaux bas

Programme associé à la fonction Encodage clavier

```
* MODULE encodage
*
* TITLE 'Scrutation du clavier'
*
* "entrées
*   clk,TA pin;
* "sorties
*   X3,X2,X1 pin istype 'reg';
*
* "équivalence
*   X=[X3..X1];
*
```

```

*
*
* EQUATIONS
*   X.clk=clk;
*   when TA==0 then
*   {
* when X==^b011 then X:=^b101;
* else
* {
*   when X==^b101 then X:=^b110;
*   else {X:=^b011};
* };
* }
*
*   else X:=X;

```

- **Structure associée à FS75 Décodage clavier**

Le sous-ensemble réalisant la détection de l'appui d'une touche est noté ENCODAGE

Description des signaux d'entrée :

X1, X2 et X3 : signaux de scrutation des trois colonnes, actifs aux niveaux bas

Y1, Y2, Y3 et Y4 : signaux indiquant l'état de chacune des 4 lignes du clavier

CLK : signal d'horloge issue de FS71 servant à la scrutation des colonnes

Description des signaux de sortie :

D, C, B et A : code binaire de la touche enfoncée

Programme associé à la fonction Décodage clavier

```

* MODULE decodage
*
* DECLARATIONs
*
* X1,X2,X3,Y1,Y2,Y3,Y4,A,B,C,D pin istype 'com';
*
* Equations
*
*
* A=!Y1&!X1#!Y1&!X3#!Y2&!X2#!Y3&!X1#!Y4&!X3#!Y3&!X3;
*
*
* B=!Y1&!X2#!Y1&!X3#!Y2&!X3#!Y3&!X1#!Y4&!X1#!Y4&!X3;
*
* C=!Y2&!X1#!Y2&!X2#!Y2&!X3#!Y3&!X1;
*
* D=!Y3&!X2#!Y3&!X3#!Y4&!X1#!Y4&!X3;
*
* END

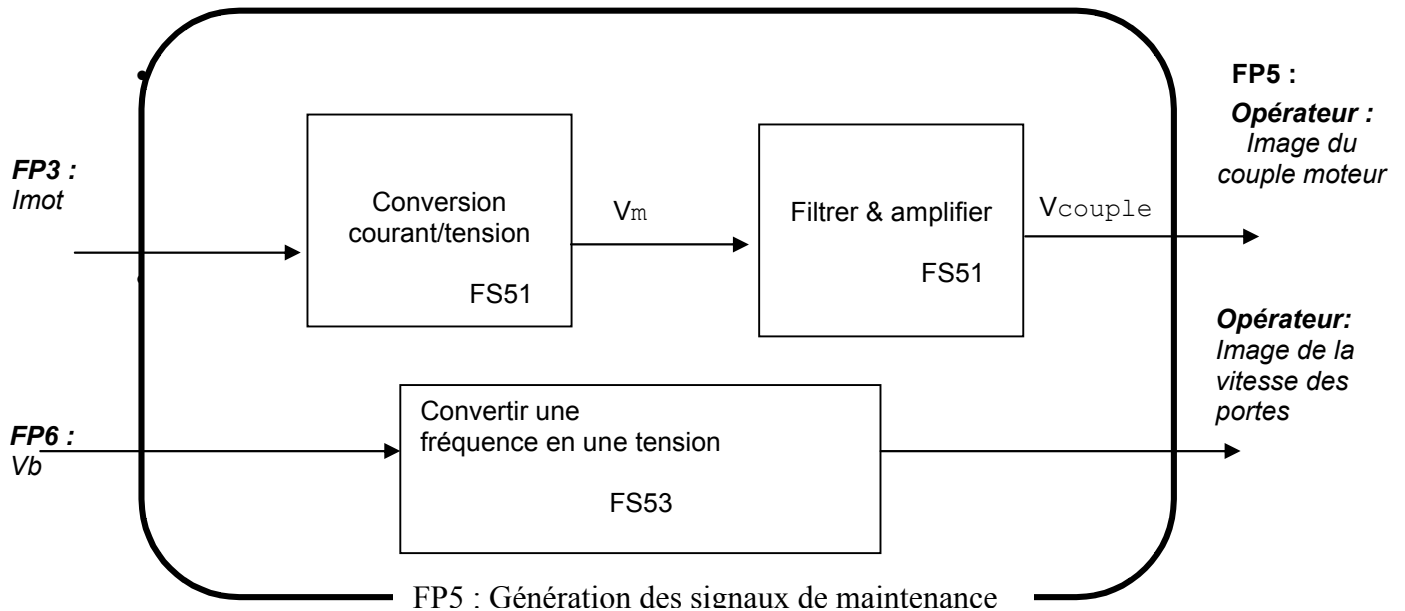
```

3/ Physique appliquée : étude de FP5 :

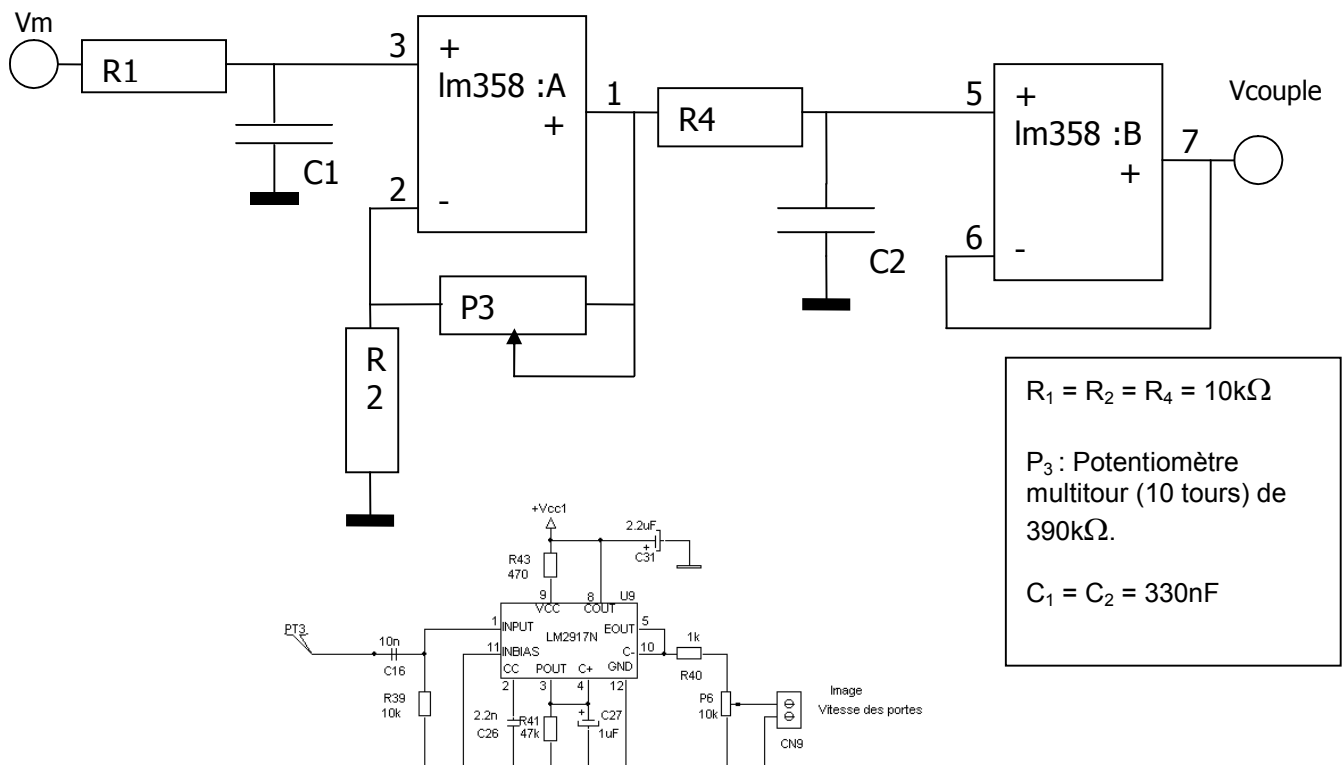
Cette fonction doit élaborer une tension-image du couple moteur moyen ainsi qu'une tension-image de la vitesse du moteur.

Le technicien de maintenance pourra ainsi visualiser et mesurer en temps réel, le couple et la vitesse du moteur afin de détecter d'éventuels problèmes

- Schéma fonctionnel de degré 2 de la fonction FP5 :**



- Schéma structurel de la fonction FS51 et FS53 :**



3.1/ Chaîne de mesure du couple moteur

Justification théorique- chaîne de mesure du couple

- **Captage du couple :**

Le moteur utilisé est un moteur à courant continu à aimants permanents. Le couple électromagnétique est directement proportionnel à l'intensité du courant fourni au moteur.

$$\boxed{T = K_T \cdot I} \quad \text{avec } K_T = 21,8 \text{ N.cm/A (données constructeur pour le MX10)}$$

La tension image du couple est donc prélevée aux bornes d'une résistance shunt de $0,33\Omega, 5W$ (1%) en série avec le moteur. Quelque soit le sens de rotation du moteur, le courant circule dans le même sens dans cette résistance (voir structure générale).

- **Choix de la fréquence de coupure du filtrage :**

Le filtrage doit éliminer les ondulations dues au hachage (on s'intéresse au couple moyen) mais doit permettre de conserver les variations dues aux caractéristiques mécaniques.

Fréquence de hachage :
$$\boxed{f_H = \frac{1}{1,1 \cdot R_{25} \cdot C_{13}} = 554 \text{ Hz}}$$

Données constructeur pour le MX10 : Constante de temps mécanique du moteur :

$T_{mech} = 4,3 \text{ ms}$ soit une équivalence fréquentielle
$$\frac{1}{2 \cdot \pi \cdot T_{mech}} = 37 \text{ Hz}$$

Avec une expérience consistant à étudier la réponse de l'ensemble de traction à un échelon de tension, il a été déterminé en fait que la constante de temps mécanique du système est de

l'ordre de 140 ms soit une équivalence fréquentielle de l'ordre de $\frac{1}{2 \cdot \pi \cdot T_{meca}} = 1,14 \text{ Hz}$. C'est cette fréquence qui nous intéresse pour déterminer la fréquence de coupure du filtre.

Conclusion : Le filtre doit laisser passer les signaux de fréquences inférieures à 2 Hz mais doit atténuer fortement (-40 dB) les signaux de fréquences 554 Hz .

$$\boxed{1,14 \text{ Hz} < f_c \leq \frac{f_H}{10}}$$

- **Contrainte de l'alimentation :**

La fonction alimentation de la carte ne fournit que des tensions positives ($+12 \text{ V}$ ou $+5 \text{ V}$). Les ADI de la chaîne sont donc alimentés en non-symétrique $0/+12 \text{ V}$. Le choix des ADI s'est porté sur le LM358 supportant bien ce type d'alimentation (faible tension de déchet).

- **Choix de l'amplification en continu de la chaîne :**

Le circuit LM3524 qui génère la MLI alimentant le hacheur, possède un limiteur de couple qui se déclenche pour un courant supérieur à environ 700 mA . Ce qui se traduit par une tension maximale aux bornes de la résistance de captage du courant d'environ $0,33 \times 700 \text{ mA} = 231 \text{ mV}$.

On veut qu'à cette tension maximale en entrée corresponde une tension maximale en sortie afin de profiter de toute la dynamique de la chaîne.

Pour s'affranchir de la tolérance des composants et des tensions de déchets des AOP, on choisit $V_e \text{ max} = 300\text{mV}$ devant correspondre à $V_s \text{ max} = 10\text{V}$

On en déduit l'ordre de grandeur de l'amplification en continu de la chaîne :

$$A \approx \frac{10}{0,300} \approx 33,3$$

L'amplification est assurée par le 2ème étage et est réglable grâce à P3.

- **Caractéristique de transfert de la chaîne de mesure du couple :**

Etude fréquentielle de la chaîne de mise en forme :

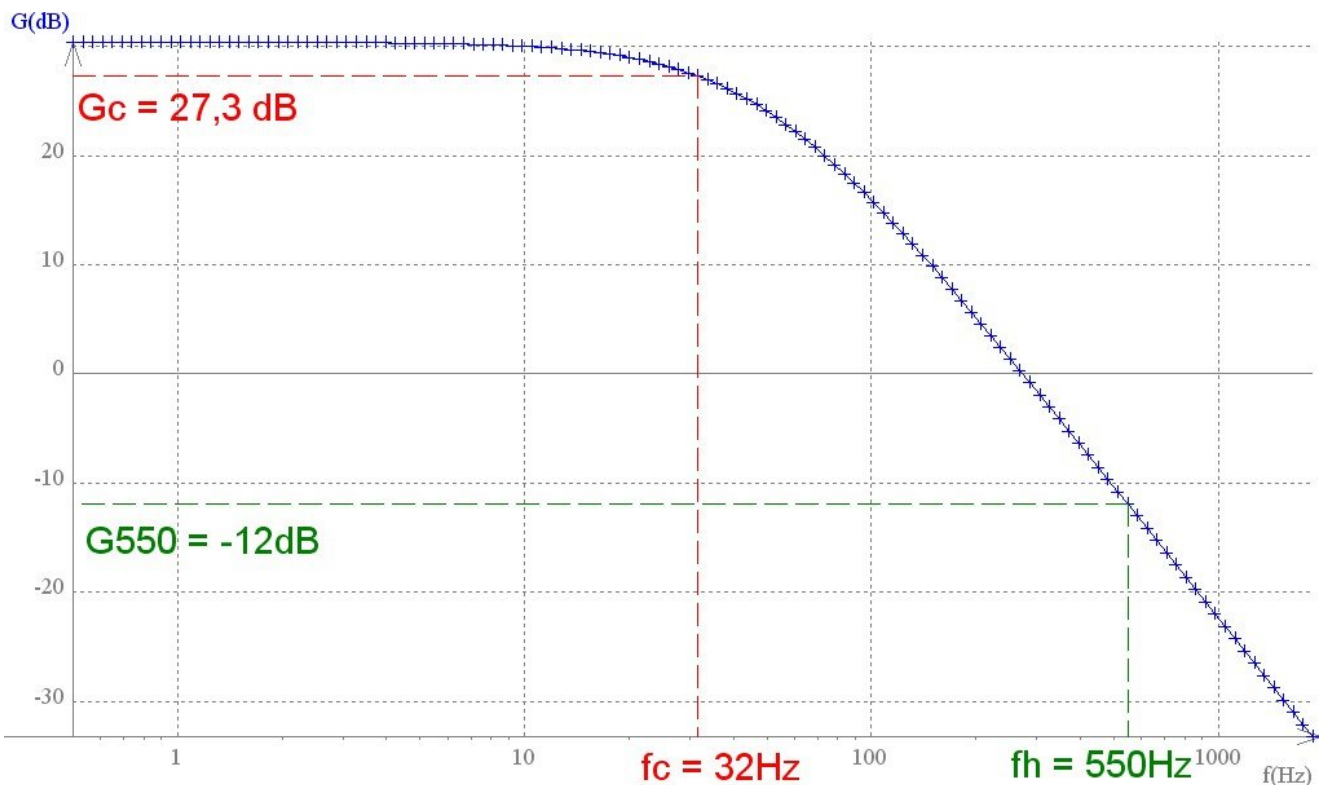


Diagramme de Bode théorique de la chaîne de mesure du couple

Fonction de transfert du premier étage : $T_1 = \frac{1}{1 + j.R_1.C_1.\omega}$

Fonction de transfert du deuxième étage : $T_2 = 1 + \frac{P_3}{R_2}$

Fonction de transfert du troisième étage : $T_3 = \frac{1}{1 + j.R_4.C_2.\omega}$

Fonction de transfert de la chaîne : $T = T_1.T_2.T_3$

→ **Choix de R_1 , R_4 , C_1 et C_2 :**

On prend pour simplifier $R_1 = R_4$ et $C_1 = C_2$ donc $\underline{T_1} = \underline{T_3}$

Donc
$$\underline{T} = \frac{T_2}{(1 + j.R_1.C_1.\omega)^2}$$

En module
$$T = \frac{T_2}{1 + (R_1.C_1.\omega)^2}$$

Ce qui nous conduit à l'expression théorique de la fréquence de coupure à -3dB :

$$f_c = \frac{\sqrt{\sqrt{2}-1}}{2.\pi.R_1.C_1}$$

En choisissant $R_1 = R_4 = 10\text{k}\Omega$ et $C_1 = C_2 = 330\text{nF}$, on obtient une fréquence de coupure de $f_c \approx 31\text{Hz}$

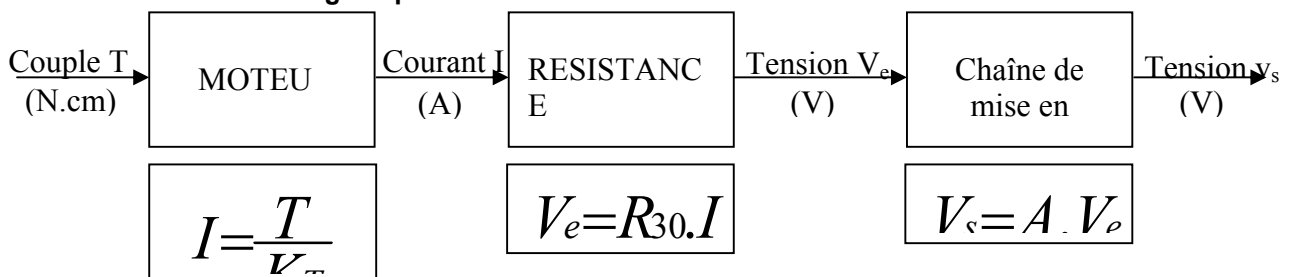
→ **Réglage de P_3 :**

Dans la bande passante, on veut une amplification de $A \approx 33$.

Donc $T_2 = 1 + \frac{P_3}{R_2} \approx 33$.

En choisissant $R_2 = 10\text{k}\Omega$, on doit régler $P_3 = 320\text{k}\Omega$.

• **Etude en régime permanent continu :**



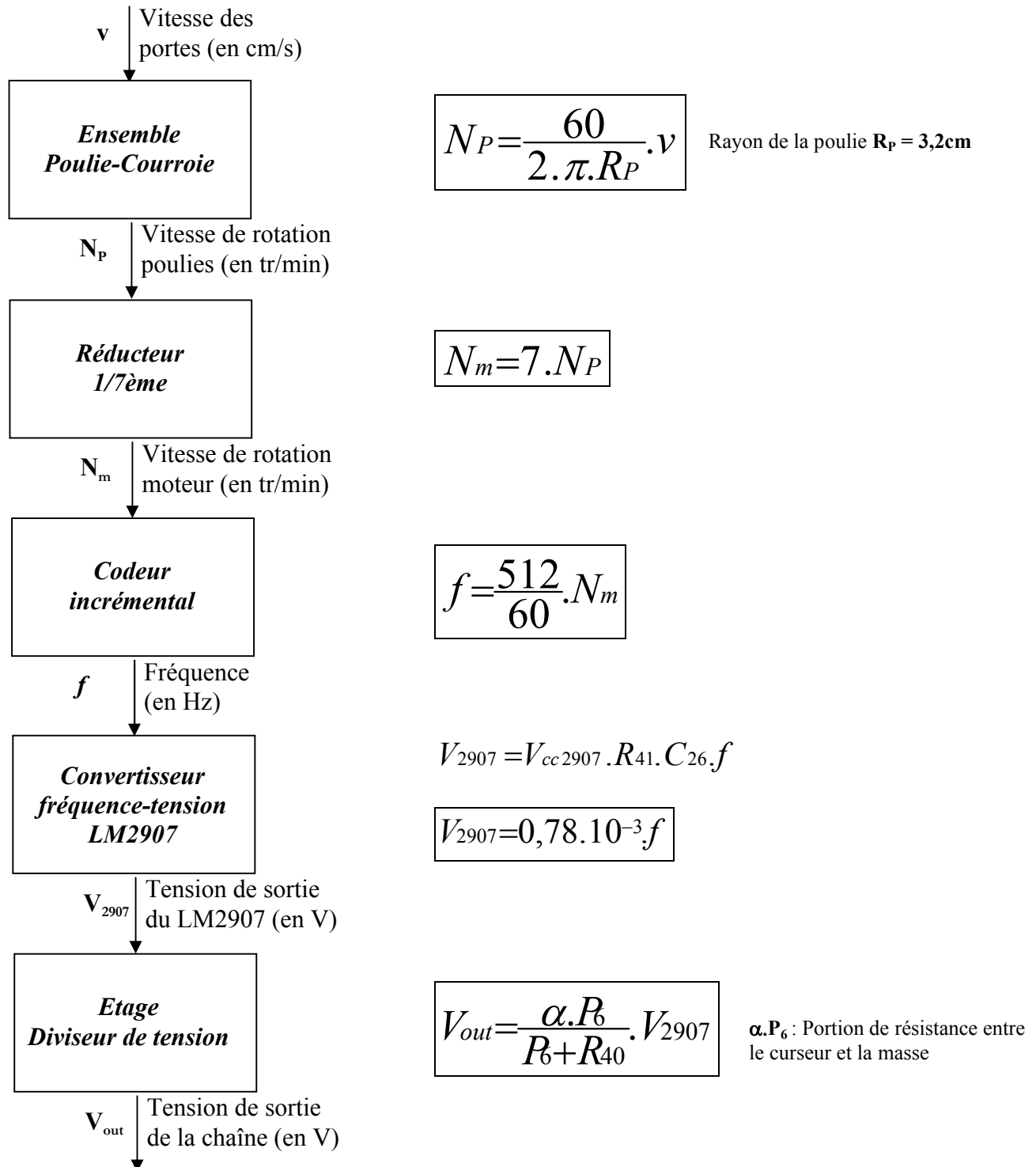
Donc de façon idéale (sans tenir compte de la tension de déchet), on a :

$$T = \frac{K_T}{A \cdot R_{30}} \cdot V_s \approx 2 \cdot V_s$$

Il est donc très facile et très rapide d'obtenir une valeur du couple développé par le moteur à partir de la mesure de la tension de sortie de la chaîne.

3.2 / Etude de la chaîne de mesure de la vitesse du moteur (FS53) :

Cette fonction est assurée par un codeur incrémental de 512 pas par tour solidaire du rotor du moteur associé à un convertisseur fréquence tension LM2907. La tension de sortie du LM2907 est donc directement proportionnelle à la vitesse de rotation du moteur.



NB : Le C.I. LM2907 possède son propre régulateur intégré de tension. V_{cc2907} est donc égal à 7,56V selon les données constructeur.

La conversion **tension de sortie-Vitesse des portes** se fait donc par la relation :

$$v = \frac{2 \cdot \pi \cdot R_p \cdot 60 \cdot (R_6 + R_{40})}{60 \cdot 7.5120,78 \cdot 10^{-3} \cdot \alpha \cdot R_6} \cdot V_{out}$$

$$v = \frac{7,9}{\alpha} \cdot V_{out}$$

On choisit α pour avoir de façon très pratique

$$v \approx 10 \cdot V_{out}$$

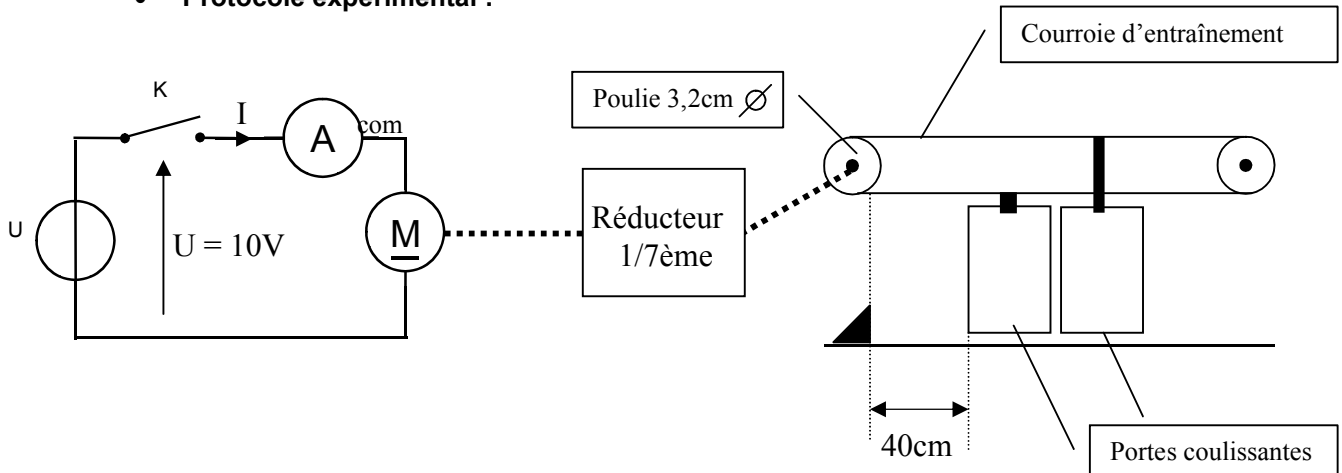
Soit $\alpha = 79\%$

3.3 / Détermination expérimentale de la constante de FEM du moteur :

- **But de la manipulation :**

Vérifier expérimentalement la valeur de $K_E = 21,3 \text{ V}/1000\text{tr}/\text{min}$ donnée par le constructeur.

- **Protocole expérimental :**



On mesure le temps t que met une des portes coulissantes pour parcourir 40cm lorsque le moteur est alimenté sous une tension de 10V. Cette mesure doit être réalisée en fonctionnement établi. En pratique le moteur doit déjà tourner lors du déclenchement du chronomètre (pour éviter le régime transitoire du démarrage). On effectuera plusieurs mesures pour en faire une moyenne et gagner en précision.

On relève l'intensité du courant moyen dans le moteur pendant le fonctionnement.

- **Relevés expérimentaux :**

Relevé	1	2	3	4	5	Moyenne
t (s)	2,41	2,47	2,37	2,44	2,37	2,41

On mesure $I \approx 200\text{mA}$.

- **Exploitation des résultats :**

$$v = \frac{40}{2,41} = 16,6 \text{ cm/s}$$

Vitesse de déplacement des portes :

$$\omega_p = \frac{v}{R} = \frac{16,6}{3,2} = 5,19 \text{ rad/s}$$

Vitesse angulaire des poulies :

$$\Omega_m = 7 \cdot \omega_p = 7 \cdot 5,19 = 36,31 \text{ rad/s}$$

Vitesse angulaire du moteur :

$$N = \frac{60 \cdot \Omega_m}{2 \cdot \pi} = \frac{60 \cdot 36,31}{2 \cdot \pi} = 346,76 \text{ tr/min}$$

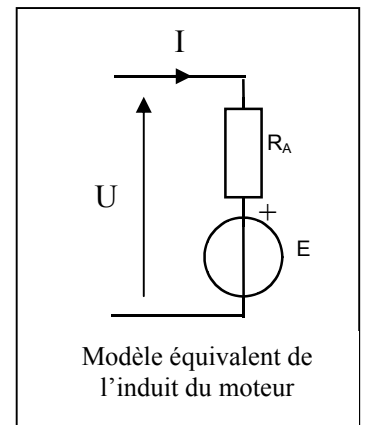
Vitesse du moteur :

$$\text{FEM du moteur (voir schéma ci-contre) : } E = U - R_A \cdot I = 10 - 9,0 \cdot 0,2 = 8,2 \text{ V}$$

$$\text{Constante de FEM exp. : } K_E = \frac{E}{N} = \frac{8,2}{346,76} = 23,6 \cdot 10^{-3} \text{ V/tr} = 23,6 \text{ V}/1000\text{tr/min}$$

$$\text{Erreur relative : } \frac{\Delta K_E}{K_E} = \frac{|23,6 - 21,3|}{21,3} = 11\%$$

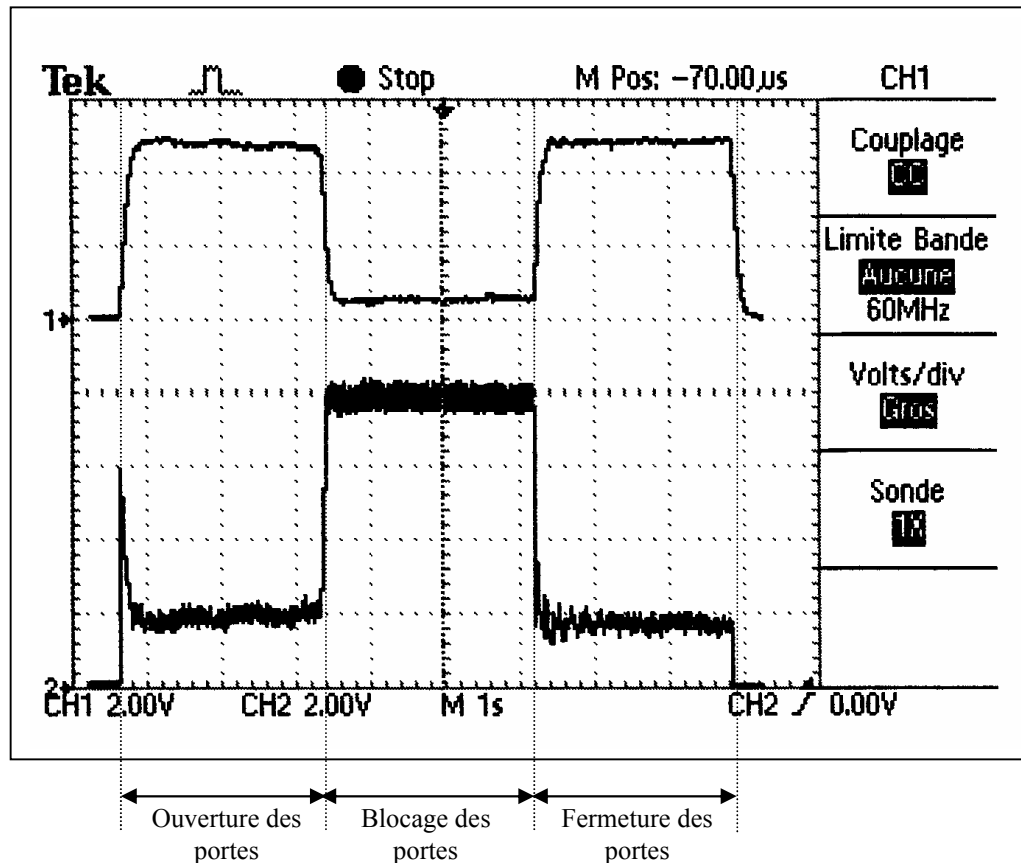
Compte tenu du protocole expérimental, cette erreur relative est acceptable.



3.4/ Chronogrammes qualitatifs :

- Phase d'apprentissage :

Lorsque le système est mis sous tension, il effectue automatiquement une phase d'apprentissage. Les portes s'ouvrent jusqu'au blocage (en butée), puis se referment. Cette phase permet au système de connaître la course maximale des portes.

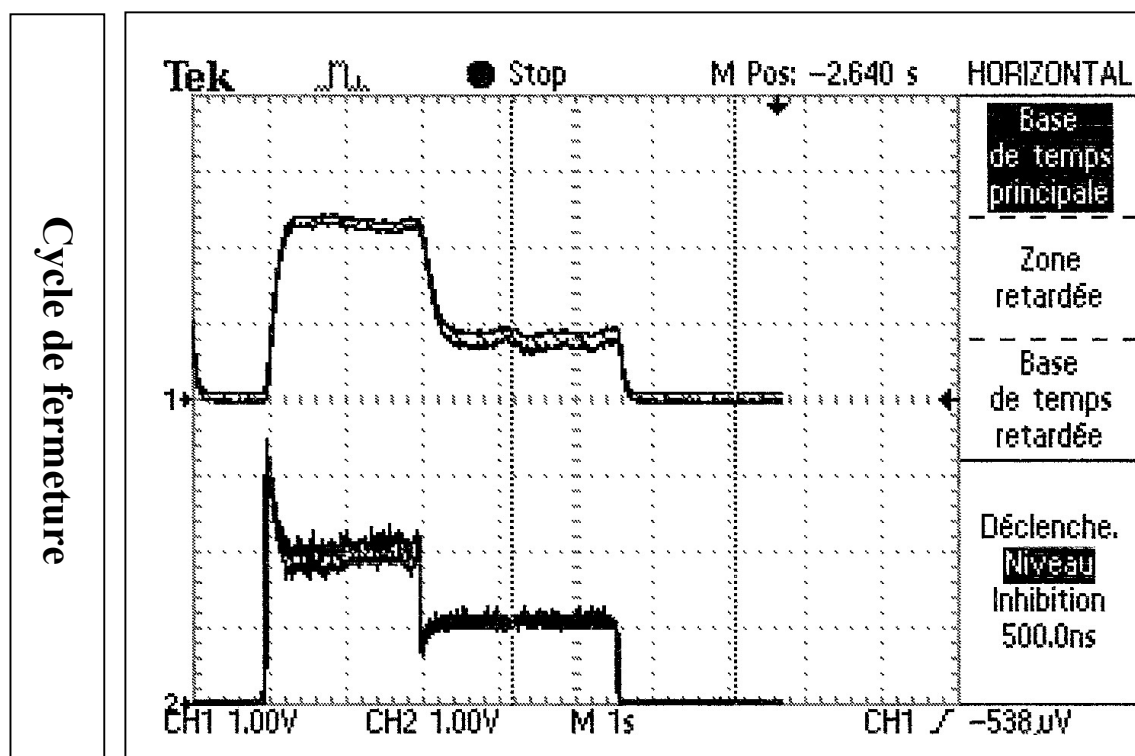
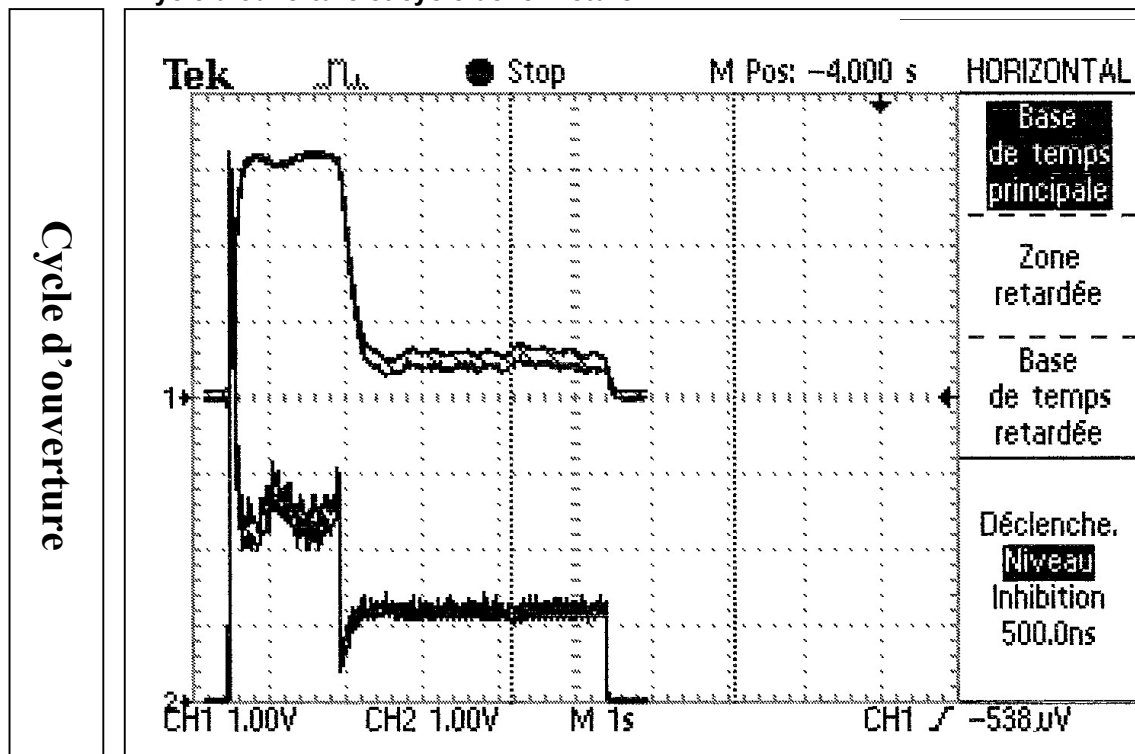


Légende :

Canal 1 : Image de la vitesse du moteur

Canal 2 : Image du couple moteur

- Cycle d'ouverture et cycle de fermeture :



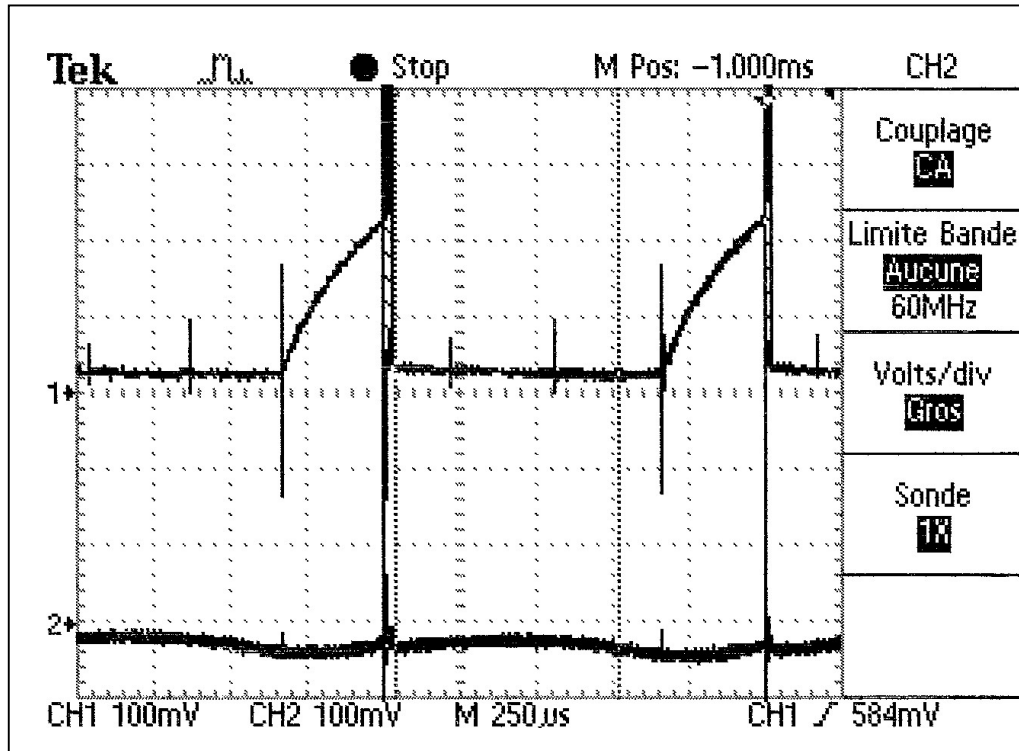
Légende :

Canal 1 : Image de la vitesse du moteur
Canal 2 : Image du couple moteur

Remarques : On observe bien pour les 2 cycles les 2 phases de vitesses de déplacement des portes.

On observe bien également que l'ouverture des portes se fait plus rapidement que la fermeture. On voit d'ailleurs que le couple de démarrage est plus important lors de la phase d'ouverture que lors de la phase de fermeture.

- Le filtrage de la chaîne de mesure du courant moteur :



Légende :

Canal 1 : Tension aux bornes de la résistance shunt R_{30}

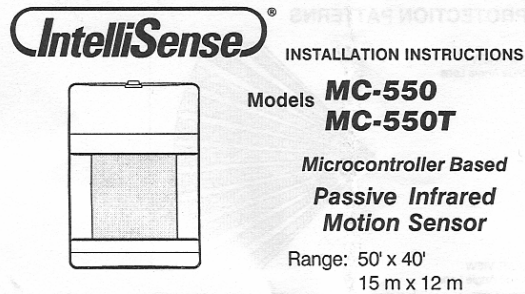
Canal 2 : Tension à la sortie de la chaîne de mesure du courant.

4/ Documentation constructeur

• 4.1 Caractéristiques du moteur MX10 :

	Symbole	Unité	MX10
Caractéristiques nominales			
Couple nominal	C_n	Ncm	0,35
Vitesse nominale	N_n	t/min	3000
Puissance nominale	P_n	W	110
Tension nominale 5%	U_n	V	75
Courant nominal	I_n	A	2,2
Valeurs limites			
Couple impulsionnel max.	C_{max}	Ncm	350
Courant impulsionnel max.	I_{max}	A	18
Durée de charge pour I _{max}	t_B	S	10
Courant max. en rotation lente	I_0	A	-
Vitesse max.	N_{max}	t/min	5000
Caractéristiques intrinsèques			
Constante de FEM ±5%	K_E	V/1000/t/min	21,3
Constante de couple ±5%	K_T	Ncm/A	21,8
Chute de vitesse à tension const.	K_n	t/min.N.cm	0,45
Couple de frot. Sec	C_F	N.cm	2,2
Coef. de frot. visqueux	K_D	Ncm/1000/t/min	0,5
Résist. d'induit (à 25°C)	R_A	Ohm	5,1
Inductance d'induit	L_A	mH	3,2
Const. de temps mécanique	T_{mech}	ms	4,3
Const. de temps électrique	T_{el}	ms	0,63
Moments d'inertie	J	g.cm ²	380
Valeurs thermiques			
Const. de tps therm. rotor carcasse	T_{th}	min	20
Résist. therm. rotor carcasse	R_{th}	°C/W	3
Données mécaniques			
Charge radiale de l'arbre	F_r	daN	15
Charge axiale de l'arbre	F_a	daN	15
Poids standard	mM	kg	0,85
Poids avec tachy	mM+T	kg	1,1

4.2 Documentation constructeur du radar de détection



FEATURES

- 50' x 40' / 15 m x 12 m range
- Dual element PIR
- Dense detection pattern
- Wide angle optics
- Low 20 mA current draw at 12 VDC
- 10-14 VDC operation
- Zone finder
- Energized form A alarm relay
- Microcontroller signal processing
- Temperature compensation
- Diagnostics
- RF and white light immunity
- Selectable PIR sensitivity
- Mounting flexibility: on walls or in corners
- Bug proof

MOUNTING LOCATION

The MC-550 / MC-550T is designed for use **indoors**. Make sure the sensor has a clear line-of-sight to the protected area. Infrared energy cannot penetrate solid objects. If the sensor is blocked, it will not alarm.

Aim the sensor toward the interior of the room, away from windows and heating/cooling sources. The unit can be corner or wall mounted at either 4' / 1.2 m, 7'6" / 2.3 m, or 10' / 3 m (see Range Chart on the next page).

MOUNTING PROCEDURE

Prepare the sensor for mounting by removing the front cover and printed circuit board (PCB).

To remove the front cover of the sensor, use a screwdriver to slide into the latch release slot located at the top of the unit and then gently push up. (See Figure 1.)

To remove the PCB, press the latch holding the PCB in place and gently pull up on the PCB until it releases. (See Figure 2.)

Using the rear housing as a template, mark mounting and wire access locations. Pull several inches of wire into the housing and fasten the rear housing to the wall.

Figure 1 Opening the MC-550/MC-550T

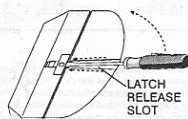
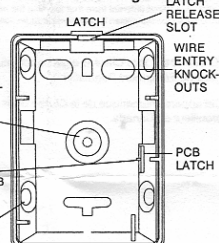


Figure 2 Rear Housing

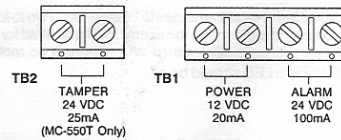


WIRING

Terminals TB1 and TB2 (MC-550T only) are located at the top edge of the PCB. Wire the MC-550 / MC-550T as shown in Figure 3 using 14-22 AWG wire. After the wiring is complete, push as much of the wire as possible back into the wiring.

For proper wiring methods, refer to the National Electrical Code, NFPA 70.

Figure 3 Wiring



CHANGING THE FRESNEL LENS

To install the optional pet-alley lens*:

Figure 4 Changing the Lens

- Remove the front cover of the MC-550 / MC-550T. (See Figure 1.)
- Press up on the Bug Guard latch and then pull the Bug Guard out of the front cover.
- Remove the existing lens, and place the new lens in with the SMOOTH side facing outward.
- Install the lens with the **small** slot at the **top** and the **large** slot at the **bottom**.
- Install the look-down mask (optional) over the inside of the look-down window.
- Place the Bug Guard into the look-down window then snap the Bug Guard back into place.
- Reassemble the housing.

*Lens Option Kit Part Number 0-000-012-01

Important: When using pet-alley lens, optimum mounting height is 4 feet / 1.2 m and adjust PCB to +1 position.

PIR SENSITIVITY

DS1 (Alarm LED) illuminates when an alarm condition is triggered.

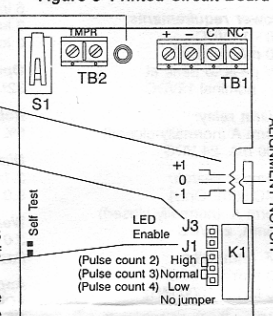
ALIGNMENT NOTCH
Important: Align with appropriate notch on the PCB (refer to Range Chart on the next page).

Install jumper at J3 to enable Alarm LED (DS1). Jumper is factory installed. (MC-550T Only)

Jumper position J1 on the PCB is used to set the sensitivity of the PIR detector.

Using the chart below, configure the MC-550 / MC-550T for the sensitivity best suited to your application.

Figure 5 Printed Circuit Board



JUMPER J1	
High (PC2)	Jumper top & center pins
Normal (PC3)	Jumper center & bottom pins
Low (PC4)	Parked on one pin or off

A sensitivity setting of low is not recommended when run-through performance is required.

Note: Do not mount the unit at 10 feet / 3 m when using the low sensitivity setting.

WALK-TEST

Apply power to the unit. The power up self-test takes **3.7 minutes**. The LED flashes during power up self-test. Begin walk-testing when the alarm LED (DS1) goes out. (The unit is in the zone finder mode for the first 10 minutes.)

Walk across the protected area at the ranges to be covered. Two to four normal steps across the pattern should make the alarm LED light. Wait for the alarm LED to go out, then continue walk-testing. When there is no motion in the protected area, the alarm LED should be off.

ZONE FINDER

The sensor will enter a 10 minute zone finder mode after a successful power up diagnostics test or a user initiated test. Either one of these tests takes approximately 3.7 minutes.

While in the zone finder mode, the installer can determine the precise protection pattern by further walk-testing the unit. The alarm LED will activate every time a zone edge is crossed.

Important: The MC-550 / MC-550T should be tested at least **once each year** to ensure proper operation.

SELF-TEST

Ongoing self-test occurs every twenty-four hours. If a self-test fails, it retests every five minutes. Ongoing self-test does not flash the LED unless there is a failure. Power up and user initiated self-tests both perform the same functions. To perform a user initiated self-test, short the two self-test pads (refer to Figure 5).

The LED flashes once per second during the power up and user initiated self-test. If the unit fails any self-test, it will continue to flash the LED once per second.

ALARM LED DISABLE (MC-550T Only)

To disable the alarm LED (DS1) after walk-testing the sensor, remove the jumper from position J3 on the PCB. See Figure 5.

SPECIFICATIONS**Range:**

50' x 40'
15 m x 12 m

Power requirements:

10 - 14 VDC
20 mA, 12 VDC
3V peak to peak at
nominal 12VDC

Alarm relay:

Form A (normally-closed)
100 mA, 24 VDC

Tamper switch:

(MC-550T Only)
Form A (normally closed)
25mA, 24 VDC

RF immunity:

30 V/m
10 MHz - 1000 MHz

PIR white light immunity:

6,500 Lux

PIR sensitivity:

jumper selectable
(high, normal & low)

PIR fields of view:

dual element
22 long range
6 intermediate
3 lower
2 look-down

Operating temperature:

32° to 120° F / 0° to 49° C

Relative humidity:

5% to 95% non-condensing

Dimensions:

3-1/2" H x 2-1/2" W x 1-13/16" D
9.0 cm x 4.4 cm x 4.5 cm

Weight:

3.0 oz / 85.27 g
Packaged product is 4.5 oz / 127.9 g

Approvals/listings:

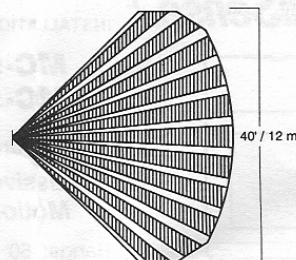
FCC/IC verified
CE approved
(EMC Directive: Residential,
Commercial and Light Industrial)
UL listed

Important: For UL certificated installation, the MC-550 / MC-550T must be connected to a UL listed power supply or UL listed control unit capable of providing a **minimum of four hours** of standby power.

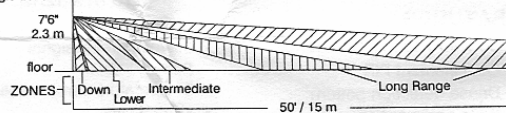
This device has been tested to compliance with the EMC Directive 89/336/EEC for Residential, Commercial and Light Industry applications, according to standards EN50081-1 and EN50082-1.

PROTECTION PATTERNS

TOP VIEW
Wide Angle Lens



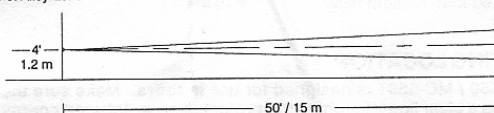
SIDE VIEW
Wide Angle Lens



TOP VIEW
Pet-Alley Lens

The TOP VIEW Pet-Alley lens is the same as the TOP VIEW Wide Angle lens.

SIDE VIEW
Pet-Alley Lens



MTG HEIGHT	PCB POSITION		
	+1	0	-1
4' / 1.2 m	50'	N/A	N/A
76° / 2.3 m	N/A	50'	N/A
10' / 3 m	N/A	N/A	50'

MC-550 / MC-550T RANGE CHART
50' / 15 m

FCC Notice: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: 1) Reorient or relocate the receiving antenna, 2) Increase the separation between the equipment and receiver, 3) Connect the equipment into an outlet on a circuit different from that to which the receiver is connected. The installer can also consult an experienced radio/television technician for additional suggestions, if necessary.

IC Notice: This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la Classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

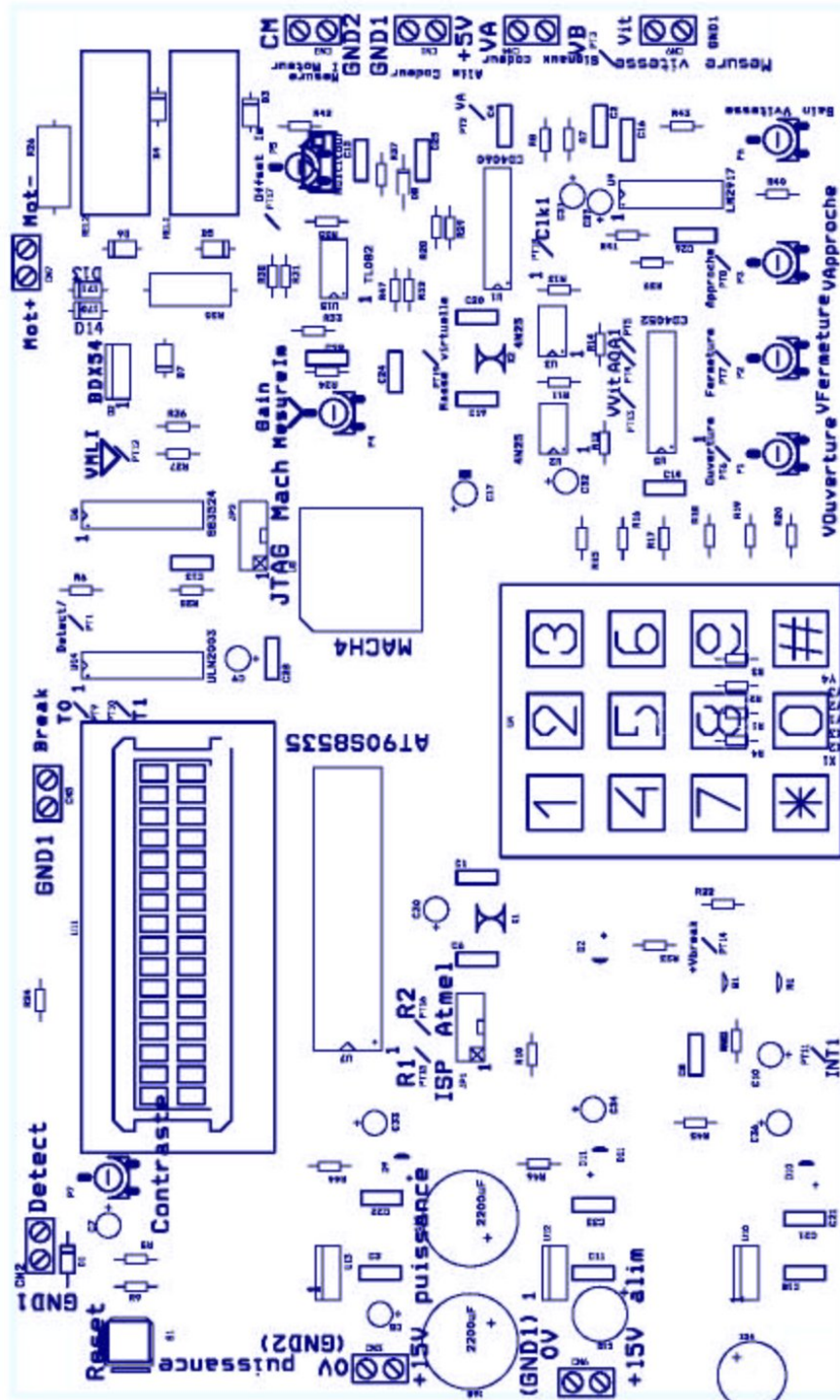


C&K is a registered trademark of C&K Components, Inc.
IntelliSense is a registered trademark of C&K Systems, Inc.
All rights reserved.

<http://www.cksys.com>
© 1998 C&K Systems, Inc.
5-051-431-00 Rev E

5/ Documentation annexes

- **5.1 Schéma d'implantation de la carte de commande**



- 5.2 Nomenclature des composants

Référence schéma	Désignation	Valeur
D3,D4, D5, D6,D13, D14 D1	1N4001 1N4148	
Q1 Q2 M1	2N2907 BDX54B 2N7000	
U5 U10 U12, U13 U1 U15 U9 U6 U14 U7 U8	CD4052BCN Régulateur L7805CP Régulateur L7812CP CD4060BCN LM358N LM2917N SG3524 ULN2003A ATMEL AT90S8535 M4A5-64/32-10JC	
C1,C6, C19,C20 C13 C15,C25 C26 C24 C16 C3,C8,C11,C14,C18, C21,C22, C28, C35	Cond. Céramique Série 683 Pas 5,08mm Cond. Polyester Série 470 Cond. Polyester Série 470 Cond. Polyester Série 470 Cond. Polyester Série 470 Cond. Polyester Série 470 Cond. Polyester Série 470	22pf/100V 22nf/100V 330nf/100V 2.2nf/400V 15nf/250V 10nf/250V 100nf/100V
C5,C12 C39,C40 C7,C27,C9,C10 C30, C31, C32,C33, C38 C34, C36,C43,C44,C45	Cond. Electro. Série M - Radial Cond. Electro. Série M - Radial Cond. Electro. Série M - Radial Tantale série MDT pas : 5mm Cond. Electro. Série M - Radial	1000u /35 V 2200u/35V 1uF/50V 2.2uF/25V 10uF/50V
JP1, JP2 pour U7 pour U1,U5, U6 et U14 pour U9 pour U15 pour U2 et U3 Pour U8 Pour le LCD (16 points) et le clavier (7 points) PT1 à PT18 CN1 à CN9	Embase mâle HE10 droite classe III Supports tulipe bas profil, serie 800SL Supports tulipe bas profil, serie 800SL Supports tulipe bas profil, serie 800SL Supports tulipe bas profil, serie 800SL Supports tulipe bas profil, serie 800SL Support PLCC 44 voies Fiches femelles droite HE14 secables Points test (broches de wrapping diam0,8) Bornier pour CI, bas profil 17,5A	40 points 16 points 14 points 8 points 6 points 12 points
D9, D10,D11 D2 U2,U3 U11	Led 3mm standard Led 3mm standard Optocoupleur 4N25 Afficheur LCD 2*16 (80x36mm)	verte rouge GDM1602A

P5	Ajustable multitour cermet Série 3296Y	390K
P1,P2, P3	Ajustable horiz Série 3319P	1k
P6,P7	Ajustable horiz Série 3319P	10k
X1	Quartz HC49	8MHz
X2	Quartz HC49	4MHz
S1	Touche de contact D6 - bouton rond	Rouge
REL1, REL2	Relais Finder Série 40.31S	1RT -12V
U4	Clavier 12 touches	
	capteur détection	
		KL205/38,1SW
	Dissipateur boîtier TO-220	9°C/W
R26	Résistance bobinée Série RB 59	100 / 3W
R30	Résistance bobinée Série RB 59	0.3/3W
R45	Résis. Couche métal Série MF12 - 1/8W	220
R10,R23,43,R44, R46	Résis. Couche métal Série MF12 - 1/8W	470
R5,R24, R27,R29,R40	Résis. Couche métal Série MF12 - 1/8W	1k
R12, R14	Résis. Couche métal Série MF12 - 1/8W	1.8k
R7,R8, R36,R19	Résis. Couche métal Série MF12 - 1/8W	2.2k
R22	Résis. Couche métal Série MF12 - 1/8W	2.7k
R17	Résis. Couche métal Série MF12 - 1/8W	1.6k
R20	Résis. Couche métal Série MF12 - 1/8W	3k
R15	Résis. Couche métal Série MF12 - 1/8W	3.9k
R18	Résis. Couche métal Série MF12 - 1/8W	6.8k
R16	Résis. Couche métal Série MF12 - 1/8W	8.2k
R6,R21, R31,R33, R38,R39	Résis. Couche métal Série MF12 - 1/8W	10k
R11, R13	Résis. Couche métal Série MF12 - 1/8W	27k
R1,R2, R3, R4,R9,R41	Résis. Couche métal Série MF12 - 1/8W	47k
R25	Résis. Couche métal Série MF12 - 1/8W	82k
R28	Résis. Couche métal Série MF12 - 1/8W	1M

- **5.4 Détails sur la procédure de calcul de la position des portes**

Après la mise sous tension,

On doit tenir compte du fait que la dernière position absolue mesurée par le mach4 n'est pas forcément la position réelle des portes après un reset !
En effet, rien n'empêche un technicien d'avoir bougé les portes manuellement, entre deux reset du microcontrôleur,
Les portes peuvent aussi avoir bougé pendant que la carte était hors tension...
En effet un reset de l'atmel n'efface pas la dernière position du mach4
La différence entre le zéro du programme atmel et la position mémorisée par le mach4 est appelé : décalage.

Le décalage est égal à $255 - \text{la position lue sur le compteur MACH4}$ car le compteur du programme ne peut coder les nombres négatifs (compteur 8bits non signés) on utilise donc la propriété du compteur modulo 255.

Lors de l'appelle de la fonction calculposition, l'algorithme suivant est exécuté :

- Pos <- 0
- Pos <- Calcul de la valeur du compteur mach4 présente sur le port A
- Si décalage >= 0 alors pos <- pos + décalage
- Si pos >= 255 alors pos <- pos - 255
- Retourne pos.

Exemple :

Hypothèse :

Supposons qu'à la mise sous tension, les portes soient fermées, mais que le compteur mach4 (CptM4) ait pour valeur 115.

Après la mise sous tension, le microcontrôleur calcul le décalage :

$$\text{décalage} = 255 - \text{CptM4} = 255 - 115 = 140.$$

Ensuite, lors du premier appel à la fonction calcul position, nous aurons le calcul suivant :

- pos = 0
- pos = CptM4 = 115
- décalage >= 0 donc : pos = pos + décalage = 115 + 140 = 255
- pos >= 255 donc : pos = pos - 255 = 255 - 255 = 0
- pos = 0 alors que le compteur mach 4 indique 115, on a bien décalé le zéro relatif du programme !
Toutes les mesures suivantes seront donc référencées par rapport à ce 0 !

• 5.5 Programme en C de la carte de commande

```

/*****
This program was produced by the
CodeWizardAVR V1.0.2.1b Evaluation
Automatic Program Generator
© Copyright 1998-2001
Pavel Haiduc, HP InfoTech S.R.L.
http://infotech.ir.ro
e-mail:dhptech@ir.ro , hpinfotech@xnet.ro
Project : Portes automatiques "Magic doors"
Version : 2.0
Date : 12/9/2002
Author : Thierry Deschamps
Company : Durzy
Comments:

Chip type : AT90S8535
Clock frequency : 8,000000 MHz
Memory model : Small
Internal SRAM size : 512
External SRAM size : 0
Data Stack size : 128
*****/

// Déclarations des fichiers inclus
#include <90s8535.h>
#include <delay.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

// Déclarations des prototypes des fonctions et des procédures
int calculposition(void);
int apprentissage(void);
void initlcd(void);
void clearlcd(void);
void lcdgotoxy(unsigned char colonne,unsigned char ligne);
void printlcd(char flash *str);
void printlcdl(char *str);
void ouverture(int pos);
void fermeture(int pos);
char decodetouche(void);
int decodecode(void);
int modifiecode(void);
void affichevitesse(void);
void calculvitesse(void);
void sequence(void);
void action(void);
void modifiek(void);
void settol(void);
void tolopen(void);
void tolclos(void);

// Declarations des variables globales
eeprom int mdsecu=10; // marge de sécurité sur la course (pour éviter des
butées en cas de lecture trop lent des positions
int decalage=0; // décalage compteur lié aux jeux mécaniques et aux inerties
du système
eeprom int tolerancef=1; // marge de tolérance sur la position finale de fermeture
eeprom int toleranceo=3; // marge de tolérance sur la position finale d'ouverture
const int compeninertie=200; // Compensation de l'inertie mécanique qui provoque un
décalage du zéro des portes
int course=0; // course codée sur 8 bits des portes.
bit mvt=0; // mvt=0 => les portes sont à l'arrêt, mvt=1 => les portes
bougent
bit detect=0; // detection de passant
float vitesse=0.0; // variable globale contenant la vitesse de déplacement des
portes
float vitesseb=1.0; // vitesse mémorisée pour éviter des affichages inutiles
eeprom float facteur=40.0; // ce facteur permet de mettre à l'échelle des m/s le calcul de
la vitesse
char eeprom *code="1234"; // code intégré en flash ce code peut être modifié par
l'utilisateur

```

```

int p1=80;                // position de changeemnt de vitesse
int p2=20;                // position de changeemnt de vitesse
int tempo=1000;           // temps d'attente les portes ouvertes
int forage=1;             // forage inactif=1 actif =0;
int position=0;           // permet de mémoriser la position de la porte à un
instant T
char *vit="0.0";          // Chaîne de caractère contenant la vitesse
char wor[]="40.00";       // Chaîne de caractère contenant le facteur de conversion de
vitesse
char *tolc="000";         // Chaîne de caractère contenant les tolérances
int tol=0;                // variable globale en ram permettant l'accès en écriture de
l'eprom
char *posi="0 ";

// Calcul de la vitesse des portes
void calculvitesse(void)
{
if (mvt==1) // si la porte est en mouvement

    {
        vitesse=fabs(((float) calculposition()-(float) position)/facteur);
        position=calculposition();
    };

}

// séquençement de l'opération d'ouverture fermeture
void sequence(void)
{
if (course!=0) // si l'apprentissage est fait
    {
        detect=0;
        ouverture(p1);
        clearlcd();
        printlcd("Porte Ouverte");
        while (PIND.2==0)
            {
                // on attend qu'il n'y ai plus rien devant le radar
            };
        delay_ms(tempo); // on attend le temps tempo avant de refermer les portes

        fermeture(p2);
        clearlcd();
        printlcd("Porte Fermee");
        delay_ms(1000);
        clearlcd();
        printlcd("-attente- Taper");
        lcdgotoxy(2,2);
        printlcd(" * pour code");
    }
    else
    {
        clearlcd();
        printlcd("Pas de course !");
        delay_ms(800);
    };
}

// Routine exécutée lors d'une interruption 0 : détection d'une personne par le radar
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    detect=1;
}

// Routine exécutée lors d'une interruption 1 : forçage en cas de panique
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    forage=0;
}

// Interruption Timer 1 (output compare A)
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    calculvitesse();
}

// Calcul de la position des portes

```

```
int calculposition(void)
{
//décode la valeur du compteur mach 4 présente sur le port A
int pos=0;

pos=PINA;
if (decalage>0) pos=pos+decalage; // on ajoute le decalage à l'origine
pos=pos; // on tient compte du decalage lié aux jeux mecaniques
if (pos>=255) pos=pos-255;
if (pos<0) pos=0; // si la nouvelle position est supérieur à 255 alors il y a
dépassement de compteur 8 bits donc on soustrait 255.
return pos;
}

// Gestion de la phase d'apprentissage (mémorisation des courses)
int apprentissage(void)
{
//code d'apprentissage
int post=256;
int posi=0;

    decalage = 255 - PINA;
    mvt=0;
    // on ne s'interesse pas à la vitesse !
    clearlcd();
    // on efface le LCD
    printlcd("Mesure course");
    lcdgotoxy(0,2);
    printlcd("Position :");
    printlcd(" 0");
    //selection vitesse nulle
    PORTB.2=1;
    PORTB.3=1;

    //on s'assure que les relais sont bien en position repos (utile après un cycle complet
d'ouverture fermeture)
    PORTB.0=0;
    PORTB.1=0;
    delay_ms(40); //attention au temps de relachement des relais !

    //activation de l'ouverture
    PORTB.0=0;
    PORTB.1=1;
    delay_ms(40); //attention au temps de collage des relais !

    //selection vitesse d'approche
    PORTB.2=1;
    PORTB.3=0;

    while (post!=calculposition())// on continue le mouvement tant que la position évolue
entre 2 intervals de temps
    {
        post=calculposition();//memorisation a l'instant t

        itoa(post,posi);
        lcdgotoxy(11,2);
        printlcdl(posi);
        printlcd(" ");
        delay_ms(500); //le process evolue un certain temps
    };

    //selection vitesse nulle
    PORTB.2=1;
    PORTB.3=1;

    //on s'assure que les relais sont bien en position repos (utile après un cycle complet
d'ouverture fermeture)
    PORTB.0=0;
    PORTB.1=0;
    delay_ms(1000); //attention au temps de relachement des relais !

    clearlcd();
```

```

        printlcd("course :");
        post=post-mdsecu;
        lcdgotoxy(9,1);
        itoa(post,posti);
        printlcdl(posti);
        printlcd(" ");
        delay_ms(2000);

        //on referme
        PORTB.0=1;
        PORTB.1=0;
        delay_ms(40); //attention au temps de collage des relais !

        //selection vitesse d'approche
        PORTB.2=1;
        PORTB.3=0;

        clearlcd();
        printlcd("Fermeture");
        lcdgotoxy(0,2);
        printlcd("Position :");

        while (calculposition()>=(tolerancef))
        {
                posti=calculposition();
                itoa(posti,posti);
                lcdgotoxy(11,2);
                printlcdl(posti);
                printlcd(" ");
        };

        //selection vitesse nulle
        PORTB.2=1;
        PORTB.3=1;

        // on met les relais en position repos
        PORTB.0=0;
        PORTB.1=0;
        delay_ms(40); //attention au temps de relachement des relais !

        if (post<0) post = 0;

        return post;
}

// Initialisation du LCD
void initlcd(void)
{
        //routine d'initialisation du lcd
        //Accès et type d'afficheurs : 8bits, 5*7points, 2lignes
        #asm("cli") // inhibition des interruptions

        PORTD.0=0; // RS=0
        PORTD.1=1; // E=1
        PORTC=0x38; // selection communication
        PORTD.1=0; // E=0 : validation de la donnée
        delay_ms(5);
        PORTD.1=1; // E=1
        PORTC=0x38; // selection communication
        PORTD.1=0; // E=0 : validation de la donnée
        delay_us(100); // attente 100µs
        PORTD.1=1;
        PORTC=0x06; // incrément de la position de curseur :+1, pas de décalage du texte
        PORTD.1=0; // E=0 : validation de la donnée
        delay_us(100); // attente 100µs
        PORTD.1=1;
        PORTC=0x0E; // Affichage actif, Curseur visible, pas de clignotement
        PORTD.1=0; // E=0 : validation de la donnée
        delay_us(100); // attente 100µs
        PORTD.1=1;
        PORTC=0x01; // effacement
        PORTD.1=0; // E=0 : validation de la donnée
        delay_us(100); // attente 100µs
        PORTD.1=1;
        PORTC=0x02; // curseur en position home
        PORTD.1=0; // E=0 : validation de la donnée

```

```
delay_us(100); // attente 100µs
PORTD.1=1;
PORTC=0x80; // accès à la mémoire donnée
PORTD.1=0; // E=0 : validation de la donnée
delay_us(100); // attente 100µs
PORTD.1=1;

#asm("sei") // restitution des interruptions
}

// Effacement du LCD
void clearlcd(void)
{
    // efface l'écran du lcd
    PORTD.0=0; // RS=0
    PORTD.1=1;
    PORTC=0x01; // effacement
    PORTD.1=0; // E=0 : validation de la donnée
    delay_us(50); // attente 100µs
    PORTD.1=1;
    PORTC=0x02; // curseur en position home
    PORTD.1=0; // E=0 : validation de la donnée
    delay_us(50); // attente 100µs
    PORTD.1=1;
    PORTC=0x80; // accès à la mémoire donnée
    PORTD.1=0; // E=0 : validation de la donnée
    delay_us(50); // attente 50µs
}

// Gestion de l'affichage du LCD : positionne le curseur à la ligne et la colonne spécifiée
void lcdgotoxy(unsigned char colonne,unsigned char ligne)
{
    unsigned char pc=0;
    if (ligne==1)
    {
        pc=128+colonne;
    };
    if (ligne==2)
    {
        pc=128+64+colonne;
    };
    PORTD.0=0; // RS=0
    delay_ms(1); // attente 1ms
    PORTD.1=1; //enable=1

    PORTC=pc; // positionnement
    PORTD.1=0; // E=0 : validation de la donnée
    delay_us(100); // attente 100µs
}

// Routine d'affichage d'une chaîne de caractères sur le lcd
void printlcd(char flash *str)
{
    char flash *org3;
    org3=str;
    PORTD.0=1; // RS=1 : ecriture d'une donnée sur le LCD
    delay_ms(3); // attente 5ms
    PORTD.1=1;
    delay_us(50); // attente 100µs
    while(*str!=NULL)
    {
        PORTD.1=1;
        PORTC=*str; // accès à la mémoire donnée
        PORTD.1=0; // E=0 : validation de la donnée
        delay_us(50); // attente 100µs
        str++;
    };
    str=org3; // on repointe là où on était
}

// Routine d'affichage d'une variable sur le lcd
void printlcd1(char *str)
{
    char *org4;
    org4=str;
```



```

PORTD.0=1; // RS=1 : ecriture d'une donnée sur le LCD
delay_ms(3); // attente 5ms
PORTD.1=1;
delay_us(50); // attente 100µs
while(*str!=NULL)
{
    PORTD.1=1;
    PORTC=*str; // accès à la mémoire donnée
    PORTD.1=0; // E=0 : validation de la donnée
    delay_us(50); // attente 100µs
    str++;
};
str=org4; //faut pas se pommer dans la mémoire !
}

void ouverture(int pos)
{
    // ouverture des portes a vitesse variable

    //selection vitesse nulle
    PORTB.2=1;
    PORTB.3=1;

    //on remet les relais au repos (utile après un cycle complet d'ouverture fermeture)
    PORTB.0=0;
    PORTB.1=0;
    delay_ms(40); //attention au temps de relachement des relais !

    // on ouvre pas les portes si elles le sont déjà.
    if (calculposition()<(course-toleranceo))
    {
        // selection ouverture
        PORTB.0=0;
        PORTB.1=1;
        delay_ms(40);
        //selection vitesse ouverture
        PORTB.2=0;
        PORTB.3=0;
        mvt=1; // on indique que les portes doivent bouger pour activer
               //le calcul de vitesse lors de l'appel de l'interruption timer1 compA
    };

    clearlcd();
    delay_ms(compeninertie);
    lcdgotoxy(1,0);
    printlcd("Ouverture");

    // Gestion de changement de vitesse pendant la course
    while (calculposition()<(course-toleranceo))
    {
        if (calculposition()>=pos)
        {
            {
                //selection vitesse approche
                PORTB.2=1;
                PORTB.3=0;
            };
            affichevitesse();
        };
        //selection vitesse nulle
        PORTB.2=1;
        PORTB.3=1;

        //on remet les relais au repos (utile après un cycle complet d'ouverture fermeture)
        PORTB.0=0;
        PORTB.1=0;
        delay_ms(40); //attention au temps de relachement des relais !

        mvt=1; // on indique que le mouvement est terminé pour l'interruption timer
    }

    // Fermeture des portes à vitesse variable
    void fermeture(int pos)
    {

```

```

//selection vitesse nulle
PORTB.2=1;
PORTB.3=1;
delay_ms(20);
//selection fermeture
PORTB.0=1;
PORTB.1=0;
delay_ms(40); //on attend que les relais collent
//selection vitesse fermeture
clearlcd();
mvt=1;
PORTB.2=0;
PORTB.3=1;
// Affichage LCD
lcdgotoxy(1,0);
printlcd("Fermeture");
vitesse=0.0;
detect=0;
while(vitesse==0.0)
{
    // on attend que les portes commence à bouger

};

//boucle de changement de vitesse et test d'évènements prioritaires
while((calculposition()>=(tolerancef))&&(detect==0)&&(vitesse!=0.0))
{
    //selection vitesse fermeture

    if (calculposition()<=pos)
    {
        //selection vitesse approche
        PORTB.2=1;
        PORTB.3=0;
    };

    affichevitesse();
};
if (detect==1) sequence(); // si quelqu'un est détecté alors qu'on referme les portes,
on les réouvre
//si les portes se bloquent alors que la fermeture n'est pas finie, on les réouvre
(personne coincée)
if ((vitesse==0.0)&&(calculposition())>(tolerancef))&&(calculposition())<(course +
mdsecu)) sequence();

//selection vitesse nulle
PORTB.2=1;
PORTB.3=1;

//on remet les relais au repos (utile après un cycle complet d'ouverture fermeture)
PORTB.0=0;
PORTB.1=0;

delay_ms(300); //on attend un peu que les portes soient bien arrêtées

mvt=0;
}

//Retourne le caractère tapé sur le clavier
char decodetouche(void)
{
    char a=0;
    char b='-';
    delay_ms(10);
    a=(PIND.4)*1+(PIND.5)*2+(PIND.6)*4+(PIND.7)*8;
    switch(a)
    {
        case 0 : b='0';
        break;
        case 1 : b='1';
        break;
        case 2 : b='2';
        break;
        case 3 : b='3';

```

```

        break;
        case 4 : b='4';
        break;
        case 5 : b='5';
        break;
        case 6 : b='6';
        break;
        case 7 : b='7';
        break;
        case 8 : b='8';
        break;
        case 9 : b='9';
        break;
        case 10 : b='*';
        break;
        case 11 : b='#';
        break;
        default : b='e';
    }
    delay_ms(10);
    return b;
}

// Décodage du code saisi au clavier
// retourne 1 si le code est bon et 0 si il est mauvais
int decodecode(void)
{
    char touche;
    char eeprom *org;
    char temp[]="*****";
    char *ptrtemp;
    int ind;
    int codebon=0;
    org=code;
    touche=' ';
    ind=0;
    PORTB.4=0;
    DDRB.4=0;

    if (decodetouche()=='*')
    {
        touche=decodetouche(); // test de l'appui sur une touche
        while(PINB.4);
        while((codebon==0) && (touche!='#'))
        {

            clearlcd();
            lcdgotoxy(0,1);
            printlcd("Saisie du Code:");
            ptrtemp=&temp[0]; // on pointe sur le premier caractère du tableau temp
            ind=0;
            while(PINB.4);
            while((ind<4) && (decodetouche()!='#'))
            {
                if (PINB.4)
                {
                    temp[ind]=decodetouche();
                    touche=decodetouche();
                    ++ind;
                    lcdgotoxy(0,2);
                    ptrtemp=&temp[0]; // on pointe sur le premier caractère
                    printlcd1(ptrtemp); // on affiche le nouveau caractère ?
                    while(PINB.4); // on attend tant que la touche n'est
                    pas relachée
                    delay_ms(100); // petite pause (rebonds au
                    relachement)
                }
                else
                {
                    while(!PINB.4); // on attend tant que la touche n'est
                    pas appuyée
                    delay_ms(30);
                }
            };
        };
    }
};

```

```

ind=0;
while( (temp[ind]==*code)&& (*code!=NULL) )
{
    code++;
    ind++;
};
code=org;
if(ind==4) // ça veut dire que la boucle à été exécutée 4 fois
donc le code est bon
{
    codebon=1;
    clearlcd();
    printlcd("Code bon !");
    delay_ms(800);
}
else
{
    codebon=0;
    clearlcd();
    printlcd("Code errone !");
    lcdgotoxy(0,2);
    printlcd("# pour sortir");
    touche=decodetouche();
    delay_ms(800);
    for(ind=0;ind<4;ind++) temp[ind]='*';
};
}; //while codebon
if (touche=='#')
{
    clearlcd();
    printlcd("-attente- Taper");
    lcdgotoxy(0,2);
    printlcd(" * pour code");
};
}
else
{
    if (codebon==1)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
if (codebon==1)
{
    return 1;
}
else
{
    return 0;
}
code=org; //il faut remplacer le pointeur code en position initiale
}

// Modification du code secret en EEPROM
int modifiecode(void)
{
    char touche;
    char temp[]="****";
    char *ptrtemp;
    int ind=0;
    char eeprom *org;
    org=code;
    while (PINB.4); // attente du relachement de la touche en cours

    if (PINB.4==0)
    {
        clearlcd();
        printlcd("Changement code");
        lcdgotoxy(0,2);
        printlcd("Patiencez 1s...");
        delay_ms(1000);
        clearlcd();
    }
}

```

```

    printlcd("Ancien code ?");
    lcdgotoxy(0,2);
    printlcd("Appuyer sur *");
    delay_ms(1000);
    while(decodecode()!=1);

    clearlcd();
    printlcd("Nouveau code ?");
    lcdgotoxy(0,2);
    ind=0;
    while (ind<4)
    {

        if (PINB.4)
        {
            temp[ind++]=decodetouche();
            touche=decodetouche();// on lit la touche suivante
            ptrtemp=&temp[0]; // on pointe sur le premier caractère du
tableau

            lcdgotoxy(0,2);
            printlcd1(ptrtemp);// on affiche le nouveau caractère ?
            delay_ms(5); // avec cette tempo, on s'affranchis des rebonds
sur le clavier (signal TA)
            while (PINB.4)
            {
                // attente du relachement de la touche en cours
            }
            delay_us(1000); // petite pause
        }
        else
        {
            while(!PINB.4);
        };
    }

    clearlcd();
    printlcd("Valider par *");
    lcdgotoxy(0,2);
    ptrtemp=&temp[0];
    printlcd1(ptrtemp);

    ind=0;
    while (!PINB.4);
    touche=' '; // attente d'appuis sur une touche }
    while (touche!='*')
    {
        if (PINB.4)
        {
            touche=decodetouche();
        };
    };

    code=org;
    for(ind=0;ind<4;ind++)
    {
        *code=temp[ind];
        ++code;
    };
    lcdgotoxy(0,1);
    printlcd("Le code est :");
    lcdgotoxy(0,2);
    ptrtemp=&temp[0];
    printlcd1(ptrtemp);
    delay_ms(1000);

};
code=org;// il faut repointer sur le premier caractère de code
return 0;
}

// Procédure d'affichage de la vitesse sur le lcd
void affichevitesse(void)
{
    char *org2;
    if (vitesseb!=vitesse)

```

```

    {
        org2=vit;
        ftoa(vitesse, 2, vit);
        if (*vit!=NULL)
        {
            lcdgotoxy(0,2);
            printlcd("Vitesse=");
            lcdgotoxy(8,2);
            printlcd1(vit);
            printlcd("m/s   ");
        };
        vit=org2;
        vitesseb=vitesse;
    };
// si la vitesse n'a pas changée, on ne perd pas de temps à l'afficher,
// cela permet de suivre correctement l'évolution de la position des portes
}

void modifiek(void)
{
    int badstring=0;
    int ind=0;
    int b=0;
    char touche=' ';
    float vfact=1.0;
    char *aff;
    char *org;

    clearlcd();
    printlcd("Modifier le ");
    lcdgotoxy(0,2);
    printlcd("Facteur vitesse");
    delay_ms(800);
    clearlcd();
    printlcd("Attention !!!");
    lcdgotoxy(0,2);
    printlcd("4 caracteres");
    delay_ms(800);
    clearlcd();
    printlcd("Appuyer sur #");
    lcdgotoxy(0,2);
    printlcd("touche *<=>.");
    while(decodetouche()!='#');
    clearlcd();
    printlcd("Entrer facteur :");
    lcdgotoxy(0,2);
    org=aff;

    while(PINB.4);
    delay_ms(40);
    while((ind<=4) && (touche!='#'))
    {

        if (PINB.4)
        {
            touche=decodetouche();
            if (touche=='*')
            {
                touche='.';
                ++badstring;
            };
            //on vérifie qu'il n'y a pas 2 points dans la chaîne
            if((badstring<=1) || (touche!='.'))
            {

                aff=org;
                wor[ind]=touche;
                for(b=0;b<=ind;b++) *aff++=wor[b];
                aff=org;
                lcdgotoxy(0,2);
                printlcd1(aff);

                ++ind;
            };
            while(PINB.4);          // on attend tant que la touche n'est pas relâchée
        }
    }
}

```

```

        delay_ms(100);          // petite pause (rebonds au relachement)
    }
    else
    {
        while(!PINB.4);
        delay_ms(30);
    };                          // if PINB4

};                              // fin d'entree des caractères

if (touche!='#')
{
    aff=org;
    for (b=0;b<=ind-1;b++) *aff++=wor[b];
    *aff=NULL;
    vfact=atof(wor); // travail en ram
    facteur=vfact; //recopie en eeprom
    aff=org;
    ftoa(vfact,4,aff);
    lcdgotoxy(0,1);
    printlcd("le facteur est :");
    lcdgotoxy(0,2);
    printlcd1(aff);
    delay_ms(1000);
}
else
{
    clearlcd();
    printlcd("- Action");
    lcdgotoxy(0,2);
    printlcd("Annulee !");
    delay_ms(800);
};                              //fin du test permettant de modifier valeur

} // fin de fonction qui permet de modifier le paramètre facteur dans le calcul de la vitesse

// Modification de la tolerance de course
void settol(void)
{
    char touche='-';
    tol=mdsecu;
    clearlcd();
    printlcd("  Tolerance");
    lcdgotoxy(0,2);
    printlcd("sur la course");
    delay_ms(800);
    clearlcd();
    printlcd("Appuyer sur");
    lcdgotoxy(0,2);
    printlcd("1:+ 2:- #:Sauver");
    delay_ms(800);
    clearlcd();
    printlcd("1:+ 2:- #:Sauver");

    while(touche!='#')
    {
        itoa(tol,tolc);
        lcdgotoxy(0,2);
        printlcd("Tolerance : ");
        lcdgotoxy(13,2);

        printlcd1(tolc);
        printlcd(" ");
        touche=decodetouche();
        if (touche=='1')
        {
            if(tol<255) tol=tol+1;
        };
        if (touche=='2')
        {
            if(tol>0) tol=tol-1;
        };

        delay_ms(100);
    }
}

```

```

    };
    mdsecu=tol;

    // fin de la fonction qui permet de modifier la tolerance course

    // modification de la tolerance sur l'ouverture
    void tolopen(void)
    {
        char touche='-';
        tol=toleranceo;
        clearlcd();
        printlcd("  Tolerance");
        lcdgotoxy(0,2);
        printlcd("sur l'ouverture");
        delay_ms(800);
        clearlcd();
        printlcd("Appuyer sur");
        lcdgotoxy(0,2);
        printlcd("1:+ 2:- #:Sauver");
        delay_ms(800);
        clearlcd();
        printlcd("1:+ 2:- #:Sauver");

        while(touche!='#')
        {
            itoa(tol,tolc);
            lcdgotoxy(0,2);
            printlcd("Tolerance : ");
            lcdgotoxy(13,2);

            printlcdl(tolc);
            printlcd(" ");
            touche=decodetouche();
            if (touche=='1')
            {
                if(tol<255)tol=tol+1;
            };
            if (touche=='2')
            {
                if(tol>0)tol=tol-1;
            };

            delay_ms(100);
        };
        toleranceo=tol;

    }

    // fin de la fonction qui permet de modifier la tolerance sur l'ouverture
    void tolclose(void)
    {
        char touche='-';
        tol=tolerancef;
        clearlcd();
        printlcd("  Tolerance");
        lcdgotoxy(0,2);
        printlcd("sur la Fermeture");
        delay_ms(800);
        clearlcd();
        printlcd("Appuyer sur");
        lcdgotoxy(0,2);
        printlcd("1:+ 2:- #:Sauver");
        delay_ms(800);
        clearlcd();
        printlcd("1:+ 2:- #:Sauver");

        while(touche!='#')
        {
            itoa(tol,tolc);
            lcdgotoxy(0,2);
            printlcd("Tolerance : ");
            lcdgotoxy(13,2);

            printlcdl(tolc);
            printlcd(" ");
            touche=decodetouche();
            if (touche=='1')
            {

```



```

        if(tol<255) tol=tol+1;
    };
    if (touche=='2')
    {
        if(tol>0) tol=tol-1;
    };

    delay_ms(100);
    };
    tolerancef=tol;

} // fin de la fonction qui permet de modifier la tolerance fermeture

void action(void)
{
    int val=0;
    char touch;
    touch='-';
    while(touch!='#')
    {
        while(PINB.4);
        while((touch!='1') && (touch!='2') && (touch!='3') && (touch!='#') && (touch!='4') && (touch!='5'
) && (touch!='6'))
        {
            clearlcd();
            printlcd("Appuyer sur");
            lcdgotoxy(0,2);
            printlcd("1-changer code");
            touch=decodetouche();
            delay_ms(1000);
            touch=decodetouche();
            clearlcd();
            printlcd("2-K vitesse");
            lcdgotoxy(0,2);
            printlcd("3-Tol Ouverture");
            delay_ms(800);
            touch=decodetouche();
            clearlcd();
            printlcd("4-Tol Fermeture");
            lcdgotoxy(0,2);
            printlcd("5-Tol Course");
            delay_ms(800);
            touch=decodetouche();
            clearlcd();
            printlcd("6-Ouvrir portes");
            lcdgotoxy(0,2);
            printlcd("#-Quitter");
            touch=decodetouche();
            delay_ms(800);
        }; // fin boucle attente touche appuyée
        if (touch=='1') val=modifiecode();
        if (touch=='2') modifiek();
        if (touch=='3') tolopen();
        if (touch=='4') tolclose();
        if (touch=='5') settol();
        if (touch=='6') sequence();

        delay_ms(200);
        touch=decodetouche();
    }; //while (touch!='#')
}

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A
    PORTA=0xFF;
    DDRA=0x00;
    PORTA=0x00;
    // Port B
    PORTB=0x00;
    DDRB=0x0F;
    PORTB.4=0;

```

```
// Port C
PORTC=0x00;
DDRC=0xFF;

// Port D
// E (PD0) et RS (PD1) en sorties
// tous les autres en entrées
PORTD=0x00;
DDRD=0x03;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Output Compare
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 125,000 kHz
// Mode: Output Compare
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: On
// Input Capture on Rising Edge
TCCR1A=0x00;
TCCR1B=0xCC; //reset compteur après comparaison et fclk/1024
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x30;
OCR1AL=0xD4;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Output Compare
// OC2 output: Disconnected
TCCR2=0x00;
ASSR=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: falling Edge
// INT1: On
// INT1 Mode: Falling Edge

GIMSK=0xC0;
MCUCR=0x0A;
GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;

// Global enable interrupts
//test();
#asm("sei")
initlcd();

facteur=atof(wor);
PORTB.2=1;
PORTB.3=1;
clearlcd();

printf("Theme 2003 V2.0");
lcdgotoxy(2,2);
```

```
printlcd("Lycee DURZY");
delay_ms(1000);

clearlcd();
printlcd("-attente- Taper");
lcdgotoxy(0,2);
printlcd(" * pour code");

// On doit tenir compte du fait que la dernière position absolue mesurée par le mach4 n'est
// pas forcément la position réelle des portes après un reset !
//En effet, rien n'empêche un technicien d'avoir bougé les portes manuellement, entre deux
// reset du microcontrôleur,
//Les portes peuvent aussi avoir bougé pendant que la carte était hors tension...
// en effet un reset de l'atmel n'efface pas la dernière position du mach4
//La différence entre le zéro du programme atmel et la position mémorisée par le mach4 est
// appelé : décalage.
// le décalage est égal à du 255 - la position lue sur le compteur MACH4 car le compteur du
// programme ne peut coder les nombres négatifs
// on utilise donc la propriété du compteur modulo 255.
//decalage=255-
((PINA.0)*1+(PINA.1)*2+(PINA.2)*4+(PINA.3)*8+(PINA.4)*16+(PINA.5)*32+(PINA.6)*64+(PINA.7)*128);

while (1)
{
    if (course==0)
    {
        decalage=255-PINA;
        course=apprentissage();
        p2=course/3;
        p1=course-p2;
        clearlcd();
        printlcd("-attente- Taper");
        lcdgotoxy(0,2);
        printlcd(" * pour code");
    }
    if (detect==1) sequence();
    if (decodecode()==1)
    {
        action();
        clearlcd();
        printlcd("-attente- Taper");
        lcdgotoxy(0,2);
        printlcd(" * pour code");
    };//test de décodage de code

    if (forcage==0)
    {
        clearlcd();
        printlcd("Panique !!!");
        while (PIND.3==0)
        {
            //attente de la fin du forçage
        }
        clearlcd();
        printlcd("-attente- Taper");
        lcdgotoxy(0,2);
        printlcd(" * pour code");
        if (PIND.3==1) forcage=1;
    };
};
}
```