

Contexte des programmes et chaîne d'information

Vous êtes un technicien dans une entreprise qui fabrique des volets roulants. L'entreprise décide d'automatiser ses volets et vous charge d'élaborer les programmes de commande. Elle envisage :

- *Un programme avec 2 boutons pour ouvrir et fermer le volet*
- *Une amélioration permettant un arrêt entre les 2 positions extrêmes*
- *Un programme fonctionnant avec les boutons ou une télécommande*

Il a été choisi d'utiliser un microcontrôleur PICAXE 20M2 programmé avec BLOCKLY.

2 capteurs de fin de course, 2 capteur boutons et un actionneur motoréducteur.

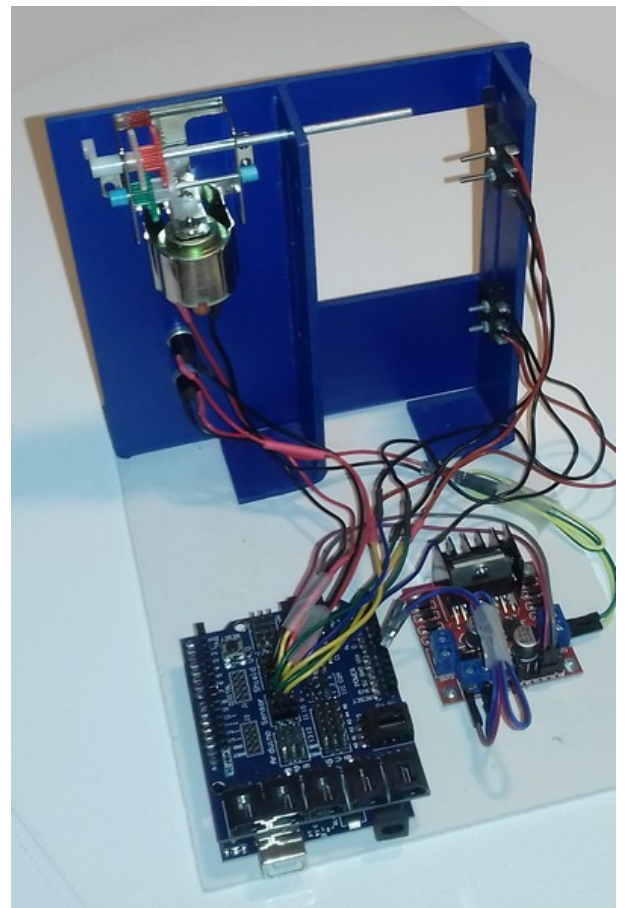
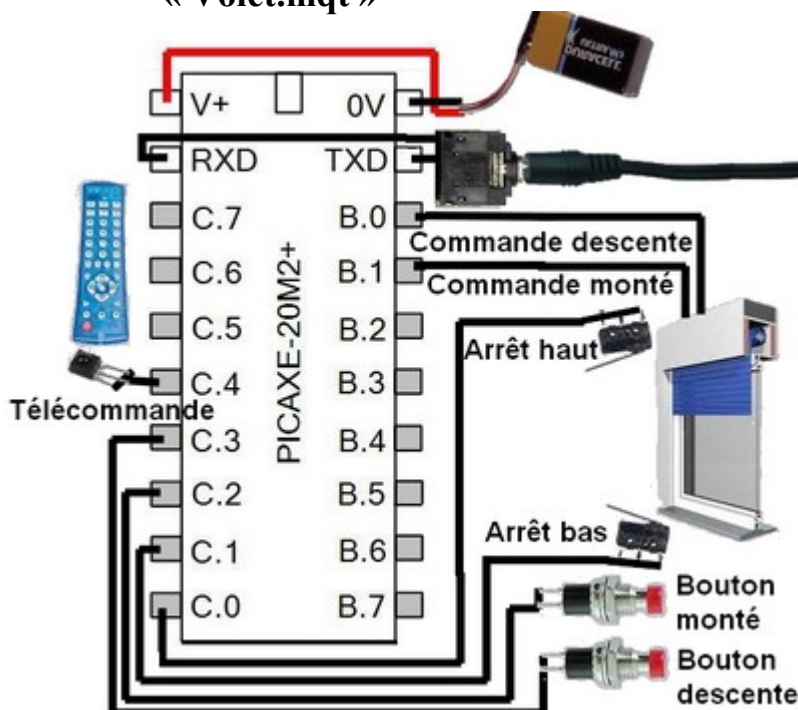
Les capteurs et actionneurs sont connectés de la manière suivante :

- Le fin de course haut sur C.0
- Le fin de course bas sur C.1
- Le bouton de monté sur C.2
- Le bouton de descente sur C.3
- Le récepteur infra rouge sur C.4
- Le moteur pour la commande de descente sur B.0
- Le moteur pour la commande de monté sur B.1



- *Un capteur est actif (1) si le volet est en position extrême ou le bouton appuyé.*
- *Le moteur est activé dans le sens demandé si seule cette sortie est active (1).*

**La maquette pour le simulateur s'appelle
« Volet.mqt »**



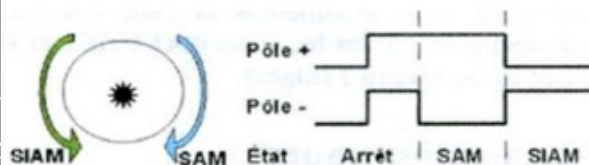


Fonctionnement du moteur



Un moteur à courant continu possède 3 états en fonction de la tension à ses bornes

Borne -	Borne +	État
Désactivée	Désactivée	libre
Activée	Activée	freiné
Désactivée	Activée	SAM sens des aiguilles d'une montre
Activée	Désactivée	SIAM sens inverse



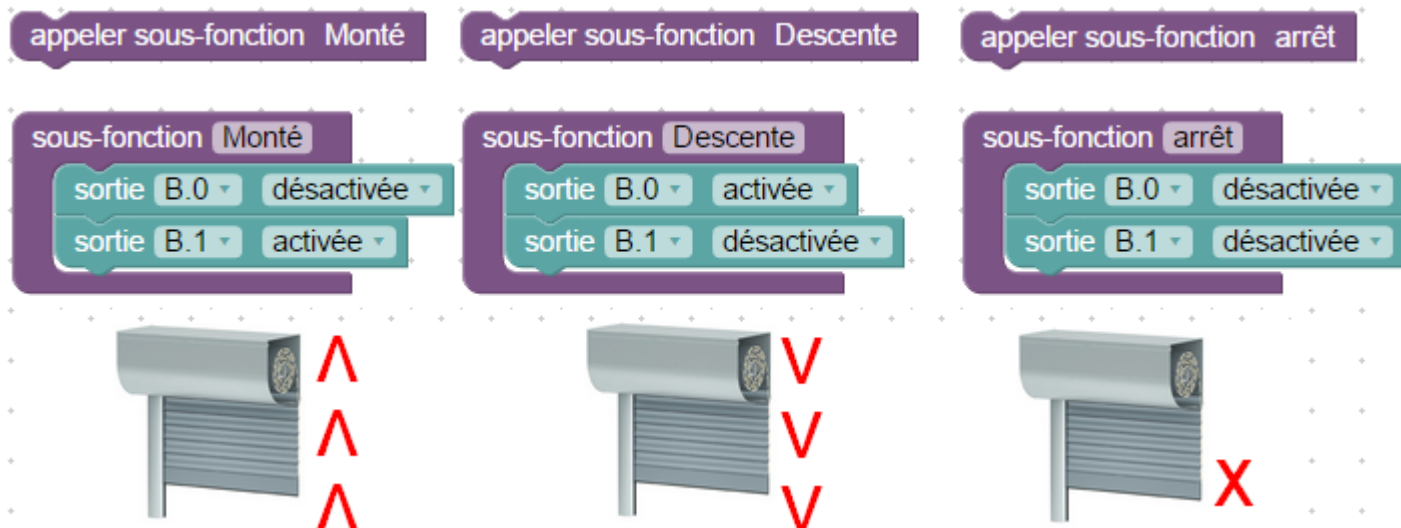
Agir sur un moteur nécessite donc 2 commandes (borne + et borne -) pour chaque action.



Sous-programme

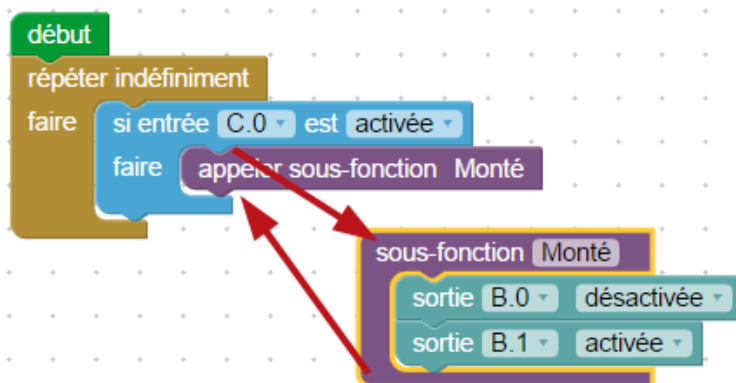
Pour simplifier les programmes à écrire, nous allons utiliser des **sous-programmes** pour les fonctions monté, descente et arrêt. Un sous-programme permet de regrouper plusieurs instructions et de les traiter comme s'il n'y en avait qu'une.

Le bloc correspondant est dans la catégorie « Procédures ». C'est le bloc « Sous-fonction ». A chaque fois qu'on l'utilise, un nouveau bloc est créé pour permettre d'appeler la sous-fonction.



Le microcontrôleur va exécuter la sous-fonction chaque fois qu'il rencontre son appel.

Lorsqu'il a terminé de l'exécuter, il retourne au programme principal à la suite de l'appel.



1 Premier programme : monte ou descend

Un bouton déclenche la montée du volet, l'autre sa descente.

Le volet s'arrête à chaque extrémité grâce au capteur de fin de course correspondant. Il n'est pas demandé d'arrêter le volet dans une position intermédiaire.



Écrivez l'algorithme, traduisez le en programme, testez le sur la maquette ou son simulateur pour contrôler s'il fonctionne correctement dans tout les cas d'utilisation.

COUP DE



- 1) Les capteurs de fin de course doivent être prioritaire sur les boutons afin d'arrêter le moteur avant qu'il force et endommage le volet.
- 2) Si le volet n'est pas en bas, je peux le faire descendre. S'il n'est pas en haut, je peu le faire monter.

2 Deuxième programme : arrêt autorisé

Une pression sur n'importe quel bouton pendant le déplacement du volet permet de l'arrêter immédiatement.

COUP DE



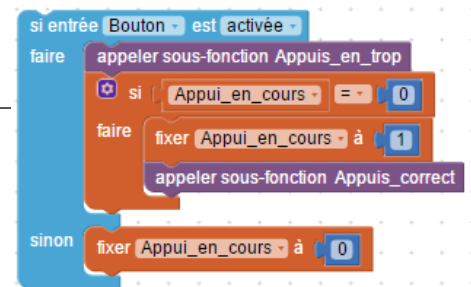
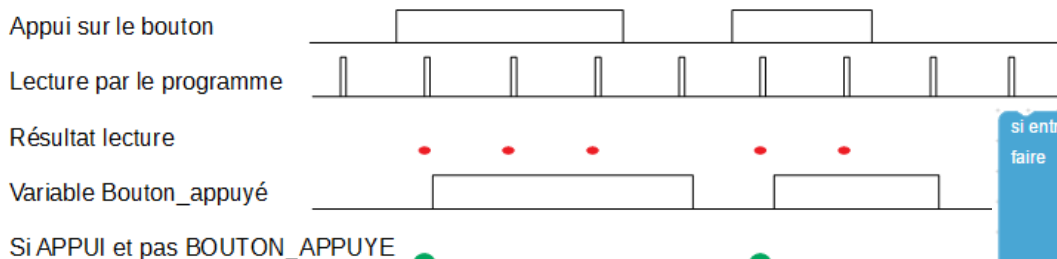
- 1) Il faut une variable pour se rappeler si le volet est à l'arrêt (et il doit être mis en route) ou s'il est en fonctionnement (et doit être arrêté).
- 2) Il faut aussi une variable pour se rappeler si le bouton vient juste d'être pressé ou s'il n'a pas encore été relâché depuis la dernière lecture. Sinon le volet va être mis en marche et à l'itération suivante il va être arrêté car le bouton n'aura pas encore été relâché.

Un microprocesseur est trop rapide par rapport au temps de réaction humain : la durée d'une instruction et de l'ordre d'une μs ($1s = 1000000 \mu s$).

La durée d'un appui bref sur un bouton est de l'ordre d'une ms ($1s = 1000 ms$)



Pour différencier 2 appuis successifs, il faut aussi détecter le relâchement du bouton



Dans l'exemple ci contre :

- Appuis_en_trop est exécuté 5 fois (points rouges)
- Appui-correct est exécuté 2 fois (points verts)

« Appui-en_cours » passe à 1 à la 1ère détection et repasse à 0 lorsque le bouton est relâché.



Écrivez l'algorithme, traduisez le en programme, testez le sur la maquette ou son simulateur pour contrôler s'il fonctionne correctement dans tout les cas d'utilisation.

3 Troisième programme : la télécommande

En plus des boutons, on utilise les touches « Monté » (code 16) et « Descente » (code 17) d'une télécommande pour agir sur le volet.

La télécommande renvoie les codes suivants :

TVR010A Key Values for PICAXE infrain2, irin commands

Symbol KEY_POWER	= 21
Symbol KEY_UP	= 16
Symbol KEY_DOWN	= 17
Symbol KEY_RIGHT	= 18
Symbol KEY_LEFT	= 19
Symbol KEY_1	= 0
Symbol KEY_2	= 1
Symbol KEY_3	= 2
Symbol KEY_4	= 3
Symbol KEY_5	= 4
Symbol KEY_6	= 5
Symbol KEY_7	= 6
Symbol KEY_8	= 7
Symbol KEY_9	= 8
Symbol KEY_MINUS	= 98
Symbol KEY_0	= 9
Symbol KEY_PLUS	= 11
Symbol KEY_BAR	= 96
Symbol KEY_TENT	= 54
Symbol KEY_VERT_CROSS	= 37
Symbol KEY_DIAG_CROSS	= 20



lire valeur infrarouge C.0 et stocker dans varA
☒ télécommande infrarouge (+1)
☒ Temps écoulé 500

Le programme est arrêté jusqu'à la réception d'un code sur la broche indiquée. Le code sera stocké dans la variable indiquée.

Si l'on coche (+1) le code est incrémenté de 1 avant d'être mis dans la variable → utile si l'on veut détecter les touches chiffres.

Si l'on indique un « temps écoulé », le programme continu sans que la variable soit modifiée au bout du temps indiqué.



Pour commander un volet

- Le code retourné n'a pas besoin d'être modifié
- Il ne faut pas interrompre le programme trop longtemps sinon les fins de courses ne sont pas détectées, le « Temps écoulé » sera de 10.

COUP DE



- Pour détecter si aucun code n'a été reçu, il faut initialiser la variable « Code » avec une valeur qui n'est jamais utilisée par la télécommande (255 par exemple).
- Pour simplifier le programme, on peut modifier la variable « Code » si les boutons sont appuyés (comme si l'ordre venait de la télécommande). L'algorithme du programme précédent pourra être conservé en modifiant les détections de monté et descente par celle des codes 16 et 17.

lire valeur infrarouge C.4 et stocker dans Code
☐ télécommande infrarouge (+1)
☒ Temps écoulé 10
si entrée C.2 est activée
faire fixer Code à 16
si entrée C.3 est activée
faire fixer Code à 17



Écrivez l'algorithme, traduisez le en programme, testez le sur la maquette ou son simulateur pour contrôler s'il fonctionne correctement dans tout les cas d'utilisation.