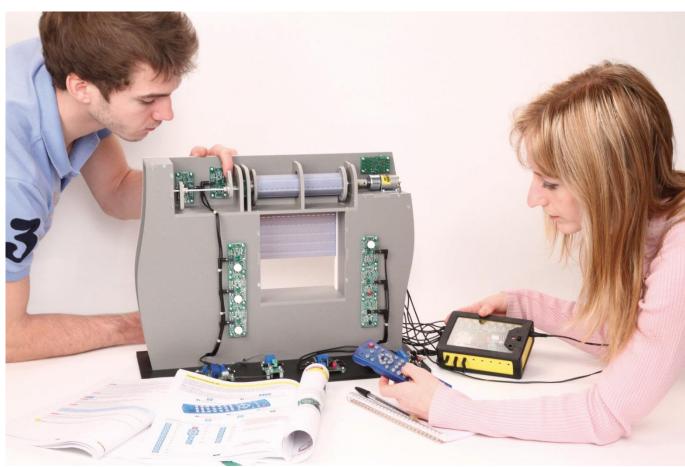
D-BE-AVOLROUL Mars 2019

# **Volet roulant**

# **Maquette programmable avec Editor / Blockly**



```
début
répéter indéfiniment
faire debug
lire valeur analogique en Capteur_LDR v et stocker dans varA v

ivarA v > v 200
faire appeler sous-fonction monter

ivarA v < v 200
faire appeler sous-fonction descendre
```

```
sous-fonction descendre

si entrée FDC_Deplie v est désactivée v
faire sortie Moteur_A1 v activée v
sortie Moteur_A2 v désactivée v
attendre jusqu'à entrée FDC_Deplie v est activée v
sortie Moteur_A1 v désactivée v
sortie Moteur_A2 v désactivée v
sortie LED_Auto v activée v
```



# Ressources disponibles pour le projet Volet roulant

Autour du projet, nous vous proposons un ensemble de **ressources téléchargeables gratuitement sur le wiki.** 

### Volet roulant

- Fichiers 3D (SolidWorks, Edrawings et Parasolid) de la maquette et de ses options.
- Dossier **technique** pour la mise en œuvre de la maquette.

### Logiciels Picaxe Editor 6 / Blockly et App Inventor

- Procédure d'installation du driver pour le câble de programmation.
- Manuel d'utilisation Picaxe Editor 6.
- Notice d'utilisation App Inventor 2.

### **Activités / Programmation**

 Fichiers modèles et fichiers de correction des programmes pour Picaxe EDITOR 6 (organigrammes et blocs) et Applinventor.

**NOTE**: Certains fichiers sont donnés sous forme de fichier.zip.



# Les documents techniques et pédagogiques signés A4 Technologie sont diffusés librement sous licence Creative Commons BY-NC-SA :

- BY: Toujours citer A4 Technologie comme source (paternité).
- ${\bf NC}$  : Aucune utilisation commerciale ne peut être autorisée sans l'accord préalable de la société A4 Technologie.
- SA : La diffusion des documents éventuellement modifiés ou adaptés doit se faire sous le même régime.

### Consulter le site <a href="http://creativecommons.fr/">http://creativecommons.fr/</a>

Note : la duplication de ce dossier est donc autorisée sans limite de quantité au sein des établissements scolaires, aux seules fins pédagogiques, à condition que soit cité le nom de l'éditeur A4 Technologie.

Logiciels, programmes, manuels utilisateurs téléchargeables gratuitement sur <u>www.a4.fr</u>

## **SOMMAIRE**

Introduction	3
Volet roulant	3
Les environnements de programmation graphique	3
Le dossier	3
Les fiches exercices	4
Prérequis	4
Caractéristiques techniques	4
Environnement de programmation graphique	5
Personnalisation des entrées/ sorties	5
Tableau d'affectation des entrées et sorties	6
Programmes de test	7
Personnalisation du jeu d'instructions	7
Procédure de chargement d'un programme	8
Mode simulation	8
Programmation version de base niveau 1	9
Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux	
Exercice niveau 1 - A.2: Répéter une action deux fois	
Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment	
Niveau 1 - B	13
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur	13
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que	14
Niveau 1 - C	15
Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir	15
Exercice niveau 1 - C.2 : Contrôle moteur ET voyant lumineux	16
Niveau 1 - D	17
Exercice niveau 1 - D.1 : Utilisation des variables	17
Exercice niveau 1 - D.2 : Utiliser et tester une variable	18
Exercice niveau 1 - D.3 : Contrôler la valeur d'une variable à l'aide des boutons poussoirs	19
Exercice niveau 1 - D.4 : Tests /variables/ modules IR	20
Programmation version de base niveau 2	21
Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses	22
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture	23
Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité	24
Exercice niveau 2 - A.4 : Contrôle d'ouverture/fermeture avec BP, signal de sécurité	25
Programmation version de base niveau 3 (OPTION)	26
Exercice niveau 3 - A.1 : Détection de lumière	27
Exercice niveau 3 - A.2 : Gérer l'ouverture et la fermeture en fonction de l'intensité lumineuse	28
Exercice niveau 3 - A.3 : Création d'un mode automatique	29
Exercice niveau 3 - A.4 : Aiout d'une sécurité	30

0	ption:Module Bluetoothption:	.31
	Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth	34
	Exercice niveau 3 - B.2 : Contrôle du volet par Smartphone	35
	Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone	. 36
	Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone	. 37
	Exercice niveau 3 - B.5 : Envoyer et recevoir des données provenant d'un Smartphone afin de contrô volet à distance	
0	ption : Module capteur de courant	. 40
	Exercice niveau 3 – C.1 : Utilisation du capteur de courant	. 44
	Exercice niveau 3 – C.2 : Capteur de courant et variable	. 45
	Exercice niveau 3 – C.3 : Sécurité du volet par contrôle du capteur de courant	. 46
0	ption : Module télécommande infrarouge	. 47
	Télécommande RAX-TVR10	. 47
	Télécommande TELEC-IR-UNIV	. 49
	Exercice niveau 3 - D.1 : Activer un voyant lumineux avec la télécommande IR	51
	Exercice niveau 3 - D.2 : Contrôle d'ouverture/fermeture avec la télécommande IR	. 52
	Exercice niveau 3 - D.3 : Contrôle d'ouverture/fermeture avec la télécommande IR	. 53
	Exercice complémentaire (bon niveau) : Gestion complète du Volet roulant à l'aide des boutons	. 54



## Introduction

### Volet roulant

La maquette volet roulant (BE-AVOL-ROUL) est une reproduction homothétique d'un volet roulant automatisé réel : roue dentée, crémaillère, capteurs fin de course, capteur de lumière, clignotant de sécurité, etc.

Programmable et piloté par les systèmes AutoProgX2 ou AutoProgUno, il permet une activité de programmation complète par rapport aux attendus de fin de cycle collège : l'algorithmique en maths, l'étude de scénarios, la programmation et la mise en œuvre en Technologie.

Vous trouverez dans ce document tout le nécessaire pour démarrer des activités de programmation autour du volet :

- La mise en œuvre de la maquette : câblage et configuration des modules.
- Différents scénarios de programmation, du plus simple au plus complexe, avec des exemples de programmes tout faits en langage par blocs.
- Des exercices complémentaires pour les différents modules en option : télécommande infrarouge, module Bluetooth, module capteur de courant.

### Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette ont été réalisés sous **PICAXE Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs **avantages** :

- Gratuit
- Blocs et organigrammes (proche algorigrammes).
- Personnalisation des noms des entrées/sorties.
- Personnalisation du jeu d'instructions.
- Mode de simulation visuelle à l'écran pour mettre au point et débugger les programmes.

Vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

Pour les activités menées avec un smartphone ou une tablette, les programmes et applications ont été réalisés sous **App Inventor 2**.

Il s'agit d'un environnement de développement pour concevoir des applications pour smartphone ou tablette Android. Il a été développé par le MIT pour l'éducation. Il est gratuit et fonctionne via internet avec Blockly.

### Le dossier

Ce document propose un parcours progressif pour découvrir et se perfectionner avec la programmation en se basant sur une série d'exemples ludiques autour de la maquette grâce à ses capteurs et actionneurs. Il est organisé en fonction des niveaux de programmation.

#### Niveau 1:

Découverte progressive du jeu d'instructions et des fonctionnalités de base de la maquette et maîtrise des principes fondamentaux pour concevoir un programme : séquences, boucles, structures conditionnelles (test) et variables.

#### Niveau 2:

Approfondissement des principes de programmation abordés dans le niveau 1 en concevant des programmes plus élaborés qui répondent à des cas concrets d'utilisation de la maquette (version de base).

#### Niveau 3:

Exemples d'utilisation des différentes options proposées : télécommande infrarouge, module Bluetooth.



### Les fiches exercices

Pour chaque niveau de programmation, nous vous proposons des fiches exercices avec :

- un objectif : ce que doit faire le programme ;
- un fichier modèle : un programme vide avec un jeu d'instructions limité (suffisant pour réaliser l'exercice) ;
- un fichier de correction qui propose un exemple de programme réalisé sous Picaxe Editor 6 en blocs (extension .xml) et en organigrammes pour le niveau 1 uniquement (extension .plf).

#### Intérêt du fichier modèle :

- il évite aux utilisateurs de se perdre dans une multitude d'instructions ;
- il limite les propositions possibles ;
- il facilite la correction et l'analyse des erreurs.

#### Deux approches:

- Avec les exemples de programmes, les utilisateurs découvrent les principes de la programmation graphique en organigrammes ou en blocs : chargement d'un programme, modification d'un programme et vérification sur le matériel (ex : modification des temps d'attente, etc.).
- Les utilisateurs conçoivent eux-mêmes le programme pour atteindre l'objectif proposé, en organigrammes ou en blocs (à partir du fichier modèle). Ils peuvent ensuite le comparer au fichier de correction.

### Principe de nommage des fichiers :

- VR pour Volet Roulant
- N: niveau de programmation 1-2-3
- A-B-C : jeu d'instructions du plus simple au plus avancé

Exemple: VR N1 C3.xml

Correspond au niveau 1 avec le jeu d'instructions C, adapté aux objectifs « avancés » de ce niveau.

### **Prérequis**

### Pour la version de base :

- Installer le logiciel Picaxe Editor 6 : <a href="http://www.picaxe.com/Software">http://www.picaxe.com/Software</a>
- Maguette Volet Roulant (Réf. BE-AVOL-ROUL).
- Câble de programmation Picaxe USB (Réf : CABLE-USBPICAXE).
- Interface programmable AutoProgX1 ou X2 (Réf. K-APV2).
- Cordons de liaison jack compatibles AutoProg pour établir les liaisons entre l'interface programmable et la maquette.

### Pour l'option Bluetooth :

- Tablette ou smartphone Android 5 ou + équipés de Bluetooth V3.
- Connexion internet pour accéder à App Inventor : http://ai2.appinventor.mit.edu/
- Compte Gmail requis.

### Caractéristiques techniques

Le guide de montage ainsi que les caractéristiques techniques des composants sont détaillés dans le dossier technique disponible sur le wiki.



# Environnement de programmation graphique

Tous les programmes correspondant aux activités menées autour du Volet roulant ont été réalisés sous **PICAXE Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs avantages :

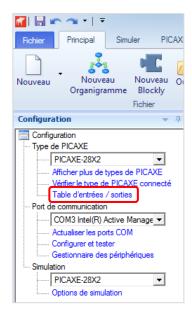
- Gratuit
- Blocs et organigrammes (proche algorigrammes).
- Personnalisation des noms des entrées/sorties.
- Personnalisation du jeu d'instructions.
- Mode de simulation visuelle à l'écran pour mettre au point et débugger les programmes.

Note : vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

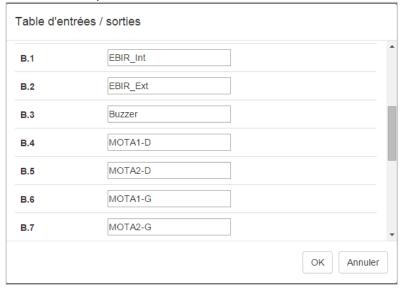
### Personnalisation des entrées/ sorties

Nous vous proposons le fichier **VR\_BASE.xml** dans lequel les noms des entrées/sorties ont été personnalisés pour une utilisation avec le Volet roulant. Tous les programmes et activités proposés dans ce document se basent sur cette liste. Celle-ci reste modifiable à tout moment.

A partir de Picaxe Editor 6, dans l'explorateur d'espace de travail cliquer sur Table d'entrées / sorties.



Une fenêtre apparaît à partir de laquelle vous pouvez modifier les noms de toutes les entrées et sorties dans la zone « Mon étiquette ».

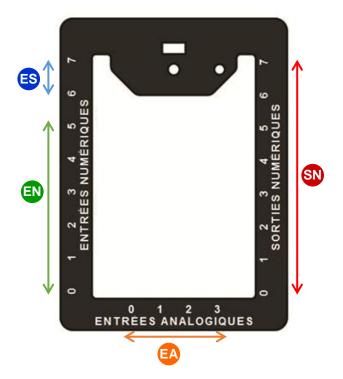


Valider en cliquant sur OK.



# Tableau d'affectation des entrées et sorties

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
	(livre)	C.7	
6	Récepteur infrarouge	C.6	Recepteur_IR
EM	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	Bouton poussoir Mode auto	C.5	BP_Auto
4	Capteur fin de course dépliement (Fermeture)	C.4	FDC_Deplie
3	Capteur fin de course repliement (Ouverture)	C.3	FDC_Replie
2	Bouton poussoir ouverture	C.2	BP_Bas
1	Bouton poussoir arrêt	C.1	BP_Arret
0	Bouton poussoir fermeture	C.0	BP_Haut
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	Capteur d'intensité lumineuse	A.1	Capteur_LDR
0	Capteur de courant analogique/numérique (option)	A.0	Capteur_courant*
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	(libre)	B.7	
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	LED	B.2	LED_Auto
1	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.1	Moteur_A2
0	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.0	Moteur_A1





### Programmes de test

Dans le but de s'assurer du bon fonctionnement des modules du volet, vous pouvez procéder à cette vérification en 10 étapes une fois votre maquette assemblée et câblée.

Si le programme ne se charge pas dans l'AutoProg, référez-vous à

http://a4.fr/wiki/index.php/Probl%C3%A8me\_de\_chargement\_d%27un\_programme\_PICAXE.

Si à une instruction vos n'obtenez pas le comportement attendu, vérifiez votre câblage.

Enfin, si vous n'arrivez pas à résoudre le problème, vous pouvez contacter notre équipe d'assistance : http://www.a4.fr/conseils assistance/ .

Il est nécessaire d'effectuer ces tests dans un environnement éclairé. Pour cela, placez le **capteur de lumière** à côté d'une source lumineuse ou faites en sorte qu'il n'y ait pas d'ombre par-dessus.

**ATTENTION :** Avant de programmer, vérifiez que les capteurs de fin de course sont activés en fin et en début de parcours du volet. Dans le cas échéant, cela pourrait conduire à une détérioration du volet.

Etape	Instruction	Instruction Comportement attendu	
1	Ouvrir le programme VR_N3_A4		
2	Appuyer sur le bouton poussoir <b>ouverture</b> ou <b>fermeture</b>	Le volet s'ouvrira ou se fermera en fonction de son état initial	
3	Appuyer sur ouverture puis sur le bouton poussoir arrêt	Le volet va s'arrêter au moment de l'appui sur le bouton d'arrêt	
4	Appuyer à nouveau sur le bouton ouverture ou fermeture	Le volet reprendra sa course là où il en était	
5	Appuyer sur le bouton poussoir auto	Une <b>LED rouge</b> s'allume	
6	Appuyer sur le bouton ouverture ou fermeture	Rien ne se passe	
7	Appuyer sur le bouton d'arrêt	La LED s'éteint	
8	Appuyer sur le bouton auto puis passer votre main au-dessus du capteur de lumière (faire un ombrage)	Le volet se ferme et se referme lorsqu'il est à nouveau éclairé	
9	Lors d'un mouvement en mode automatique, appuyer sur le bouton d'arrêt	Le volet arrête sa course	
10	Appuyer sur ouverture ou fermeture	Le volet reprend sa course et continue en mode automatique	

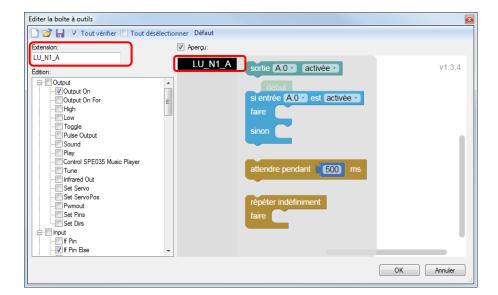
## Personnalisation du jeu d'instructions

Vous pouvez personnaliser l'affichage du jeu d'instructions pour en limiter la quantité afin de faciliter la l'analyse et la correction des erreurs. Faire un clic droit sur la zone des blocs puis cliquer sur **Editer la boite d'outil**.



Une fenêtre s'ouvre à partir de laquelle vous pouvez sélectionner ou désélectionner les instructions de votre choix. Vous pouvez renommer le jeu d'instructions dans la zone « **Extension** ».

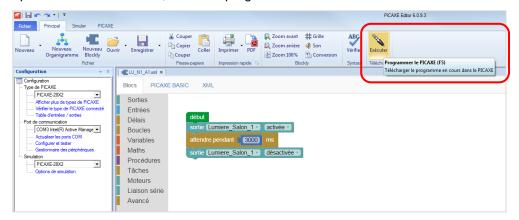




Valider en cliquant sur OK.

## Procédure de chargement d'un programme

Commencer par relier le Loupiot à l'ordinateur avec le câble de programmation USB et le mettre sous tension. A partir du Picaxe Editor 6, ouvrir un programme.



A partir du menu **Principal** ou du menu **PICAXE**, cliquer sur le bouton **Exécuter**. Vous pouvez également utiliser la touche **F5** de votre clavier.

Note : un programme téléchargé écrase le précédent.

## **Mode simulation**

La simulation sur Picaxe EDITOR 6 permet de tester un programme avant de le téléverser dans la maquette. Pour lancer et contrôler une simulation, utiliser les boutons **Exécuter / Pause / Pas à pas / Arrêt** à partir du menu **Simuler**.



La simulation surligne les blocs dans l'espace de travail pour vous montrer où en est le programme.



# Programmation version de base niveau 1

### Objectifs:

- Découvrir et maîtriser le matériel avec des exemples très simples pour débuter en programmation.
- Appréhender les différentes fonctionnalités du matériel.

Ce niveau permet de découvrir toutes les fonctionnalités de base du volet automatique, en apprenant les structures de base de la programmation. Et en particulier celles demandées dans les nouveaux programmes : séquences, boucles, structures conditionnelles et enfin les variables.

Nous vous conseillons pour chaque exercice d'essayer d'écrire le programme vous-même, en partant du modèle de base (fournit avec les exercices), avant de regarder la correction et l'explication de chaque programme. Par exemple pour le programme « VR\_N1\_A1.xml », charger le programme modèle « VR\_BASE.xml ». Dans chaque programme modèle du niveau 1 vous trouverez la liste de blocs nécessaires à la réalisation des exercices des sous niveaux A, B, C et D. Au fur et à mesure de l'avancement dans les sous niveaux, la liste de blocs s'agrandit jusqu'à retrouver tous les blocs nécessaires pour piloter complétement le volet roulant.

Nom du fichier	Description	Objectif	
Niveau 1 A (Modèle : VR_BASE.xml)			
VR_N1_A1.xml	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordé : - Allumage/extinction du voyant lumineux	
VR_N1_A2.xml	Répéter cette même action deux fois	Notions de programmation abordées : -séquence d'instructions -temps d'attente	
VR_N1_A3.xml	Répéter cette action à l'infini	-boucle infinie	
Niveau 1 B (Modèle	: VR_BASE.xml)		
VR_N1_B1.xml	Activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.	Fonctionnalité matérielle abordé : -Gestion du moteur -Utilisation de Bouton-poussoir	
VR_N1_B2.xml	Ouvrir et fermer le volet en continu jusqu'à l'appui d'un bouton-poussoir.	Notions de programmation abordées : -boucle qui dépend d'une entrée	
Niveau 1 C (Modèle	: VR_BASE.xml)		
VR_N1_C1.xml	Allumer le voyant lumineux à l'appui du BP.	Fonctionnalité matérielle abordé : -Gestion des modules infra-rouge -Utilisation de Bouton-poussoir	
VR_N1_C2.xml	Activer le voyant lumineux lorsque le capteur infrarouge est franchi.	Notions de programmation abordées : -Le test d'une entrée (si/sinon)	
Niveau 1 D (Modèle : VR_BASE.xml)			
VR_N1_D1.xml	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).		
VR_N1_D2.xml	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	Fonctionnalité matérielle abordé :  Notions de programmation abordées : -Définition de variable	
VR_N1_D3.xml	Incrémenter une variable à l'appui d'un bouton poussoir, la décrémenter à l'appui de l'autre bouton poussoir.	-Incrémentation de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage	
VR_N1_D4.xml	incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.		



### Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux

Fichier modèle : VR\_N1\_A.xml

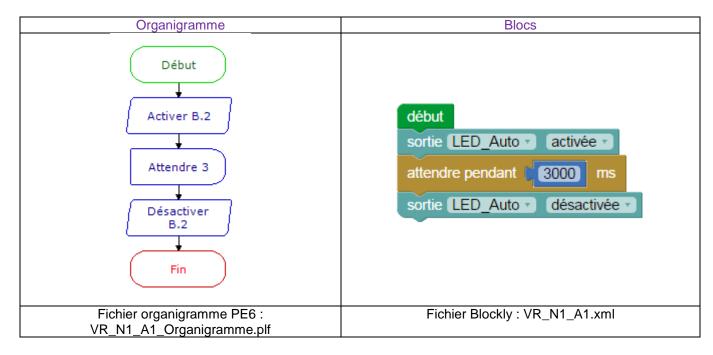
**Objectif:** allumer le voyant lumineux pendant 3 secondes puis l'éteindre.

Notions abordées: séquence d'instructions, activation / désactivation d'une sortie, temps d'attente.

Instructions utilisées :



### **Correction:**



Remarque : avec le langage de programmation par blocs la dernière instruction exécutée marque la fin du programme.



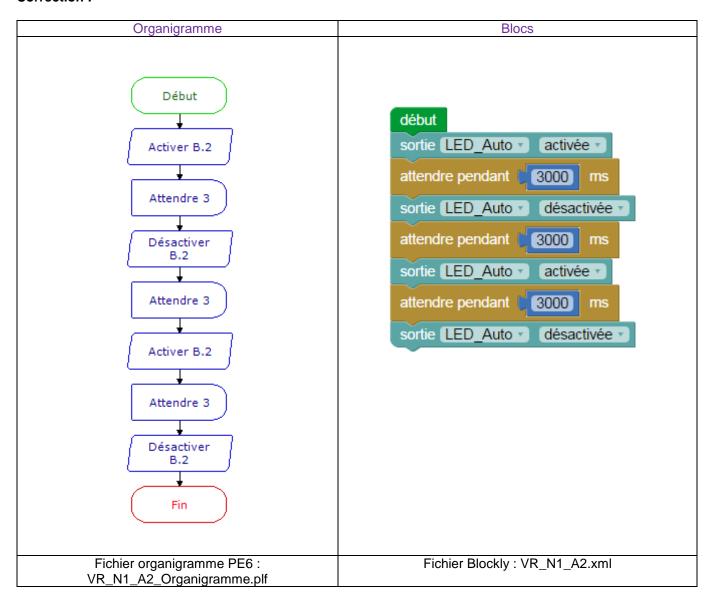
# Exercice niveau 1 - A.2: Répéter une action deux fois

**Objectif:** allumer le voyant lumineux pendant 3 secondes puis l'éteindre, recommencer.

Notions abordées: séquence d'instructions, activation / désactivation d'une sortie, temps d'attente.

Instructions utilisées :





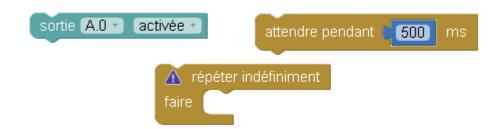


# Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment

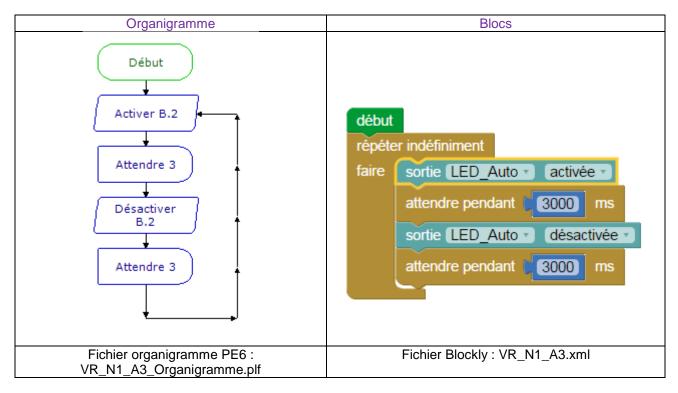
Objectif: faire clignoter le voyant lumineux avec une période de 6 secondes indéfiniment.

Notion abordée : la boucle infinie.

Instructions utilisées :



#### Correction:



**Remarque** : le programme ne peut s'arrêter lorsqu'il est dans une boucle infinie. Le seul moyen de sortir de la boucle est de faire un Reset ou d'éteindre et rallumer le boîtier AutoProg.



# Exercice niveau 1 - B.1 : Maitriser la rotation du moteur.

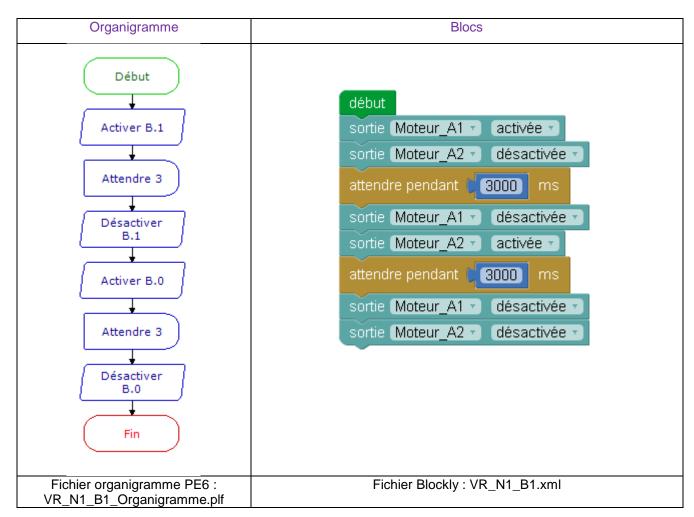
**Objectif:** activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.

Notion abordée: utilisation d'un moteur.

Instructions utilisées :



#### Correction:



**ATTENTION**: pour cet exercice il faut activer le moteur à l'aide de l'interrupteur (Une LED rouge indique si le moteur est allumé) et également activer l'interrupteur.

Il faut également placer le volet à mit hauteur pour éviter toute dégradation.

Il faut absolument éviter de faire sortir le volet de son rail. Couper l'AutoProg dans le cas où le volet sortirait du rail.



## Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que

Objectif: ouvrir et fermer le volet en continu jusqu'à l'appui d'un bouton-poussoir.

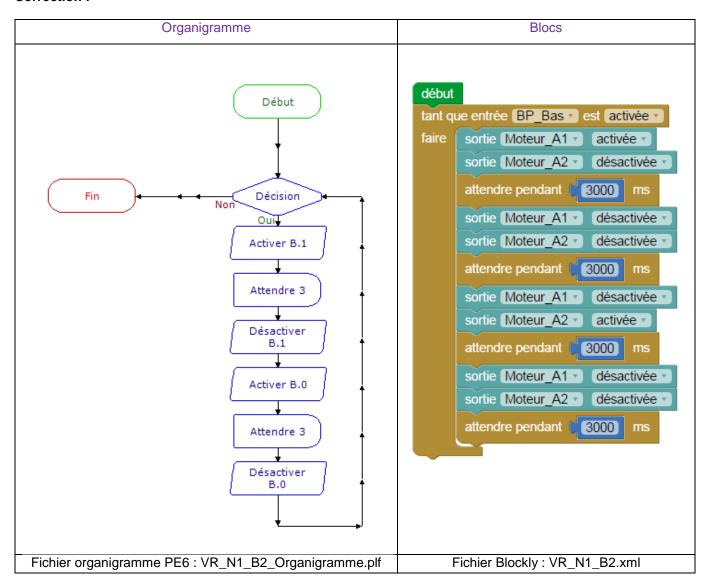
Notion abordée : exécuter une boucle qui dépend de l'état d'une entrée.

### Instructions utilisées :

```
sortie A.0 ▼ activée ▼ attendre pendant 500 ms

tant que entrée A.0 ▼ est activée ▼ faire
```

#### Correction:



**Remarque**: Le programme ne peut sortir de la boucle qu'une fois le test sur le bouton-poussoir validé. Le test sur le bouton poussoir se fait qu'une seule fois en début de séquence, avant de commencer l'ouverture. Si un appui est effectué pendant la séquence, aucun effet n'aura lieu sur le programme. Afin de vérifier à tout moment le changement d'état d'une entrée dans une séquence, l'utilisation des interruptions est indispensable (voir ex sur interruption).



# Exercice niveau 1 - C.1 : Instruction conditionnelle et boutonpoussoir

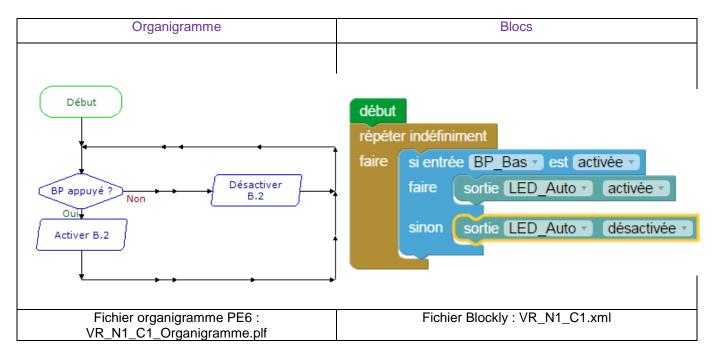
Objectif: allumer le voyant lumineux à l'appui du BP.

Notion abordée : utilisation des commandes conditionnelles (si/sinon).

### Instructions utilisées :



### Correction:



Remarque : les blocs de couleur bleu claires représente des commandes concernant l'utilisation des entrées.



# Exercice niveau 1 - C.2 : Contrôle moteur ET voyant lumineux

**Objectif**: contrôler le moteur avec les boutons poussoirs et allumer le voyant lors d'un mouvement du volet Ouvrir le volet avec le bouton d'ouverture, fermer le volet avec le bouton de fermeture et arrêter avec le bouton d'arrêt

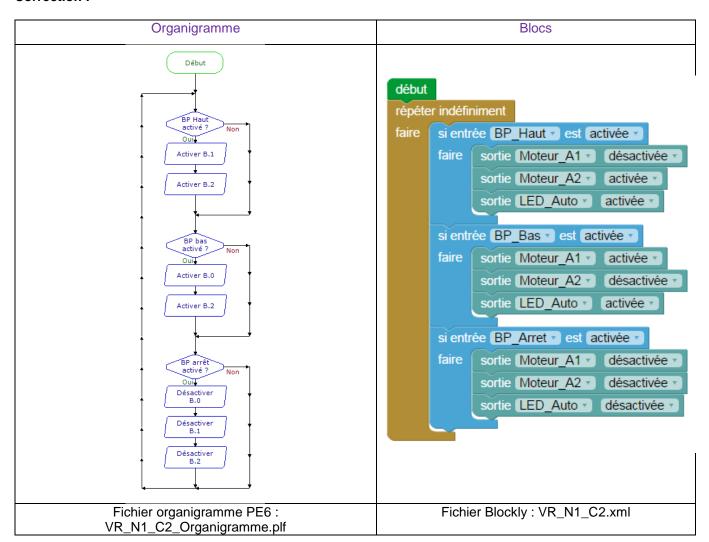
Allumer la LED lors d'un mouvement, l'éteindre lors d'un arrêt.

Notion abordée : utilisation des commandes conditionnelles.

### Instructions utilisées :



#### Correction:



**Remarque** : Attention à ne pas dépasser la limite d'ouverture ou de fermeture. Si les moteurs ne s'arrêtent pas dans un programme, arrêtez-les avec les interrupteurs.

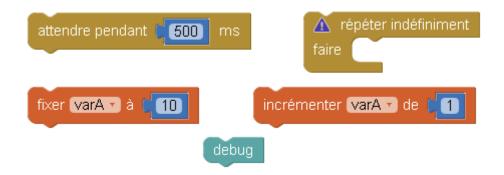


### Exercice niveau 1 - D.1 : Utilisation des variables

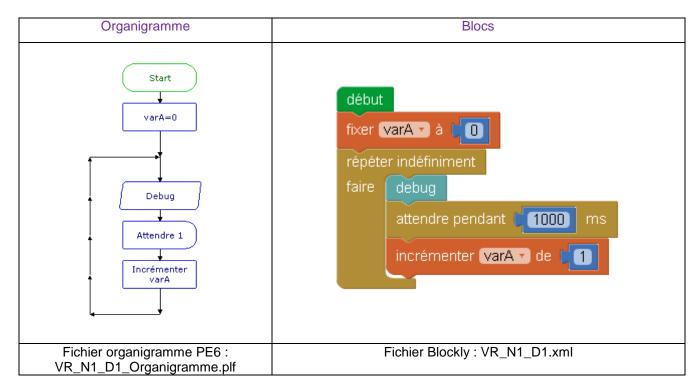
Objectif: incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).

Notions abordées : la variable : définition et incrémentation, debug.

### Instructions utilisées :



### Correction:



**Remarques : l**a commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.



# Exercice niveau 1 - D.2 : Utiliser et tester une variable

Objectif: incrémenter une variable au cours du temps. Lorsque la variable est supérieure à 10, activer le voyant.

Notion abordée : boucle tant que dépendant d'une variable

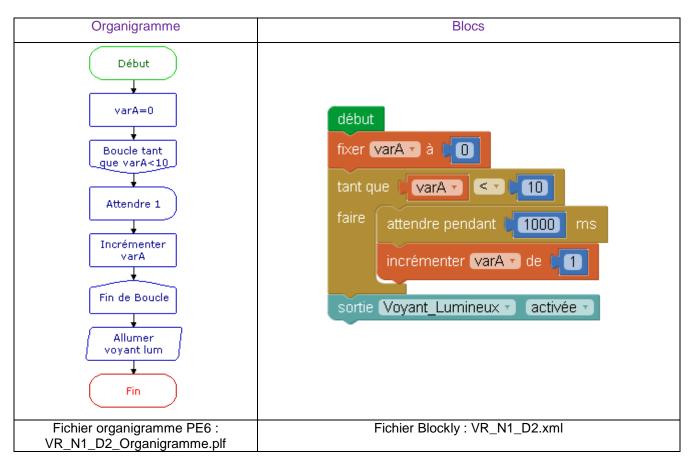
### Instructions utilisées :

```
sortie A.0 ▼ activée ▼ attendre pendant 500 ms

A tant que varA ▼ > ▼ 10 fixer varA ▼ à 10

faire incrémenter varA ▼ de 11
```

### **Correction:**



Remarque : cet exercice peut être utilisé comme un minuteur.



# Exercice niveau 1 - D.3 : Contrôler la valeur d'une variable à l'aide des boutons poussoirs

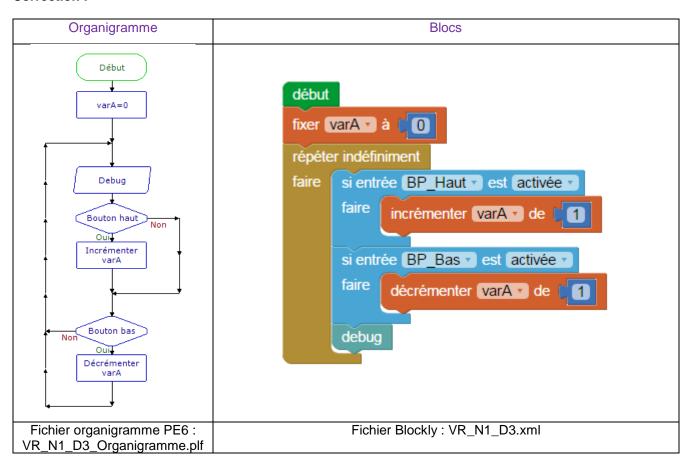
**Objectif**: incrémenter une variable à l'appui d'un bouton poussoir, décrémenter la même variable à l'appui de l'autre bouton poussoir.

Notions abordées : test sur entrées et incrémentation/décrémentation contrôlée d'une variable

#### Instruction utilisée:

```
incrémenter varA de la décrémenter varA de la decrémenter varA de la decrémenter varA de la decrementer varA de
```

#### Correction:



**Remarques :** deux tests sont insérés l'un après l'autre. La vitesse d'exécution du programme donne l'impression que les commandes sont exécutées en même temps.

La commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

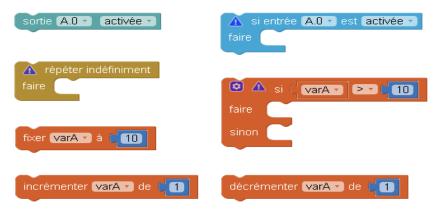


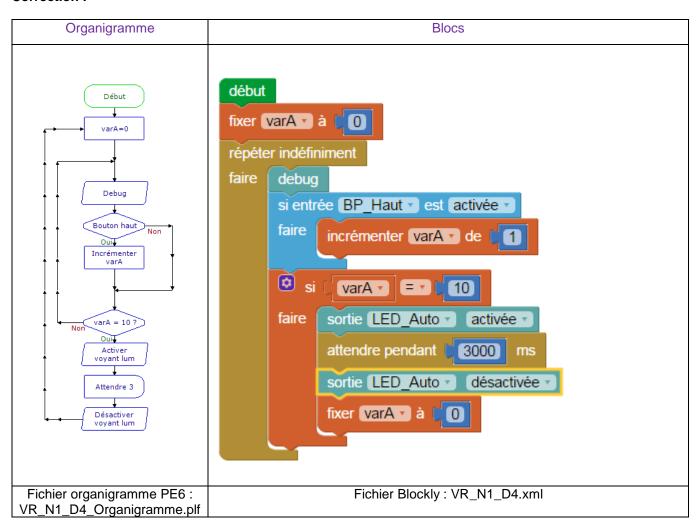
### Exercice niveau 1 - D.4: Tests /variables/ modules IR

**Objectif :** incrémenter une variable à chaque appui sur un bouton poussoir. Lorsque le compteur arrive à 10, activer le voyant lumineux 3 secondes et remettre la variable à zéro

Notion abordée : test dépendant d'une variable

### Instructions utilisées :







# Programmation version de base niveau 2

### Objectifs:

- Utilisation concrète du volet roulant
- Utilisation de tous les modules de la maquette
- Appréhension des différentes fonctionnalités du matériel ainsi que certaines notions de sécurité.

Ce niveau permet de mettre en œuvre le volet roulant, au fur et à mesures des exercices vous allez utiliser de plus en plus de modules et enrichir votre code pour obtenir à la fin du niveau un volet qui marche parfaitement et qui respecte une logique de fonctionnement calquée sur le réel.

Nom du fichier	Description	Objectif	
Niveau 2 A (Modèle : VR_BASE.xml)			
VR_N2_A1.xml	Ouvrir et fermer le volet avec 2 secondes d'attente entre chaque mouvement. Utiliser les capteurs fins de course pour contrôler l'ouverture et la fermeture.		
VR_N2_A2.xml	Montée du volet à l'appui sur BP_Haut. Descente du volet à l'appui sur BP_Bas.		
VR_N2_A3.xml	Ouvrir et fermer le volet à l'aide des BP sans distinction, faire en sorte que le voyant lumineux clignote lors d'une manœuvre du volet.	Fonctionnalité matérielle abordée : Utilisation des FDC	
VR_N2_A4.xml	Ouvrir et fermer le volet à l'aide des BP sans distinction, le voyant lumineux doit clignoter lors d'une manœuvre du volet. Inclure une gestion de sécurité lors la fermeture du volet.		



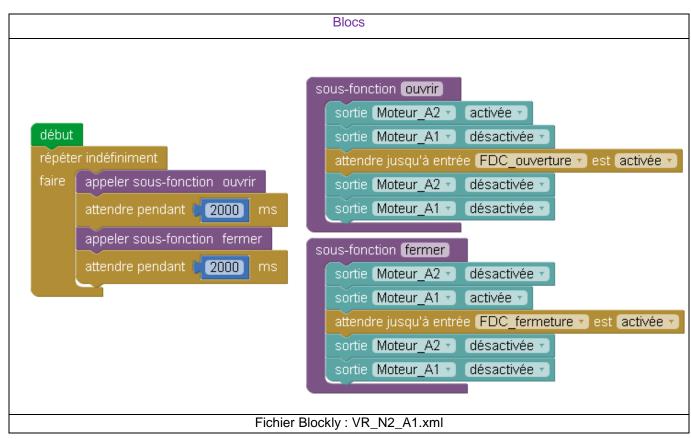
### Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses

**Objectif**: ouvrir et fermer le volet avec 2 secondes d'attente entre chaque mouvement. Utiliser les capteurs fins de course pour contrôler l'ouverture et la fermeture.

Notions abordées : utilisation des fins de course, procédures (sous-fonctions)



#### Correction:



**ATTENTION :** Avant de programmer, vérifiez que les capteurs de fin de course sont activés en fin et en début de parcours du volet. Dans le cas échéant, cela pourrait conduire à une détérioration du volet.



## Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture

**Objectif**: ouverture du volet à l'appui sur BP\_Exterieur. Fermeture du volet à l'appui sur BP\_Interieur

### Notions abordées :

```
Blocs
                                                                         sous-fonction ouvrir
                                                                            sortie Moteur_A2 activée
début
                                                                            sortie Moteur_A1 • désactivée •
répéter indéfiniment
                                                                            attendre jusqu'à entrée [FDC_ouverture • ] est [activée • ]
faire attendre jusqu'à entrée BP_exterieur v est activée v
                                                                            sortie Moteur A2 v désactivée v
     appeler sous-fonction ouvrir
                                                                            sortie Moteur_A1 • désactivée •
      attendre jusqu'à entrée BP_interieur v est activée v
                                                                         sous-fonction fermer
      appeler sous-fonction fermer
                                                                            sortie Moteur_A2 désactivée v
                                                                            sortie Moteur_A1 - activée -
                                                                            attendre jusqu'à entrée FDC_fermeture • est activée •
                                                                            sortie Moteur_A2 désactivée -
                                                                            sortie Moteur_A1 v désactivée v
                                           Fichier Blockly: VR_N2_A2.xml
```



# Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité

**Objectif**: ouvrir et fermer le volet à l'aide des BP sans distinction, faire en sorte que le voyant lumineux clignote lors d'une manœuvre du volet

Notions abordées: utilisation d'opérateur logique OU (+) Utilisation de la commande basculer pour une LED

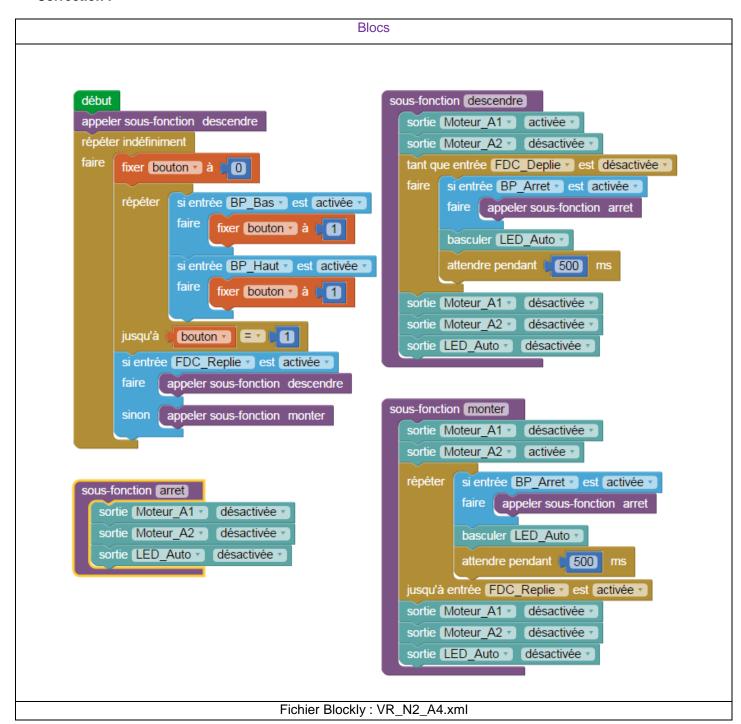
```
Blocs
                                                                               sous-fonction descendre
                                                                                  sortie Moteur_A1 v activée v
er indéfiniment
                                                                                  sortie Moteur_A2 v désactivée v
         fixer bouton · à
                             entrée BP_Haut v or v entrée BP_Bas v
                                                                                  tant que entrée FDC Deplie est désactivée
                                                                                  faire basculer LED Auto
        bouton = = 1
                                                                                        attendre pendant 500 ms
 fixer (bouton * à (0)
                                                                                  sortie Moteur_A1 désactivée
 si entrée (FDC_Replie v est activée v
                                                                                  sortie Moteur_A2 v désactivée v
      appeler sous-fonction descendre
                                                                                  sortie LED_Auto V désactivée V
 sinon appeler sous-fonction monter
                                                                               sous-fonction monter
                                                                                  sortie Moteur_A1 désactivée
                                                                                  sortie Moteur_A2 v activée v
                                                                                  répéter basculer LED Auto
                                                                                          attendre pendant 500 ms
                                                                                  jusqu'à entrée (FDC_Replie 🔻 est (activée 🕯
                                                                                  sortie Moteur A1 désactivée
                                                                                  sortie Moteur_A2 v désactivée v
                                                                                  sortie LED_Auto désactivée
                                       Fichier Blockly: VR_N2_A3.xml
```

# Exercice niveau 2 - A.4 : Contrôle d'ouverture/fermeture avec BP, signal de sécurité

**Objectif :** ouvrir et fermer le volet à l'aide des BP sans distinction, le voyant lumineux doit clignoter lors d'une manœuvre du volet. Inclure un bouton arrêt de sécurité

Notions abordées : utilisation d'une procédure de sécurité

Correction:



**Remarque :** Le bouton d'arrêt stoppe complètement le programme. Pour le relancer, éteindre et rallumer l'AutoProg.

La LED continuera de clignoter lors de l'arrêt de sécurité du programme.



# Programmation version de base niveau 3 (OPTION)

Objectif : Utiliser les modules plus complexes : pilotage à distance, contrôle par le courant.

Le niveau 3 n'intègre pas de nouvelles notions de programmation mais de nouveaux blocs permettant d'utiliser les modules options.

Nom du fichier	Description	Objectif		
Niveau 3 A : Capteur de lumière				
VR_N3_A1.xml	Lire une valeur envoyée par un capteur analogique.			
VR_N3_A2.xml	Gérer l'ouverture et la fermeture en fonction du niveau de luminosité.	Fonctionnalité matérielle abordée : Utilisation d'un capteur analogique		
VR_N3_A3.xml	Création d'un mode automatique et d'un mode manuel.			
VR_N4_A4.xml	Ajout d'un bouton d'arrêt du programme.			
Niveau 3 B : Modul	e Bluetooth			
VR_N3_B1.xml	Contrôler l'ouverture et la fermeture du volet à l'aide de 2 boutons présent sur l'application Android.	Fonctionnalité matérielle abordée :		
VR_N3_B2.xml	Ouvrir et fermer le volet à partir d'un seul bouton disponible sur l'application Android.	- module Bluetooth		
VR_N3_B3.xml	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du volet.	Notions de programmation abordées :		
VR_N3_B4.xml	Gérer la sonnette ainsi que le contrôle du volet à distance à l'aide de l'application Android.	- liaison série (hserin/hserout)		
VR_N3_B5.xml	Instaurer le mode automatique et le mode manuel pour contrôler le mode à distance.			
Niveau 3 C : Capted	Niveau 3 C : Capteur de courant			
VR_N3_C1.xml	Allumer le moteur et lire l'entrée capteur de courant.	Fonctionnalité matérielle abordée :		
VR_N3_C2.xml	Arrêter le moteur lors de la détection d'un blocage.	- capteur de courant		
VR_N3_C3.xml	Gestion complète du volet avec sécurité par capteur de courant.	Notions de programmation abordées : - lire une entrée analogique		



# Exercice niveau 3 - A.1 : Détection de lumière

Objectif : Prendre des valeurs d'intensité lumineuse à l'aide du capteur de lumière

Notion abordée : Gestion de variables analogiques

```
début
répéter indéfiniment
faire debug
lire valeur analogique en Capteur_LDR et stocker dans varA
```



# Exercice niveau 3 - A.2 : Gérer l'ouverture et la fermeture en fonction de l'intensité lumineuse

Objectif: Ouvrir le volet lorsqu'il y a beaucoup de lumière, le fermer lorsqu'il fait sombre.

Notion abordée : Gestion de variables analogiques, application

Correction:

```
Blocs
début
                                                                        sous-fonction descendre
                                                                          si entrée FDC_Deplie v est désactivée v
      debug
                                                                          faire sortie Moteur_A1 activée
      lire valeur analogique en Capteur_LDR v et stocker dans varA v
                                                                                 sortie Moteur A2 désactivée
                                                                                     dre jusqu'à entrée FDC_Deplie 🔻 est (activée 🔻
      si (varA - > - 200)
                                                                                 sortie Moteur_A1 

désactivée 

           appeler sous-fonction monter
                                                                                 sortie Moteur_A2 v désactivée v
                                                                                sortie LED_Auto v activée v
      i varA - < 200
           appeler sous-fonction descendre
                                                                        sous-fonction monter
                                                                          si entrée (FDC_Replie v est désactivée v
                                                                          faire sortie Moteur A1 désactivée
                                                                                 sortie Moteur_A2 v activée v
                                                                                     dre jusqu'à entrée (FDC_Replie 🔻 est (activée 🔻
                                                                                 sortie Moteur_A1 désactivée v
                                                                                 sortie Moteur_A2 

désactivée
                                                                                sortie LED_Auto désactivée
                                           Fichier Blockly: VR_N3_A2.xml
```

**Remarque :** Sans les boucles Si dans les fonctions, lorsque le volet arrive en fin de course, il va sans cesse relancer une sous-fonction et donc activer et désactiver le moteur à chaque boucle.



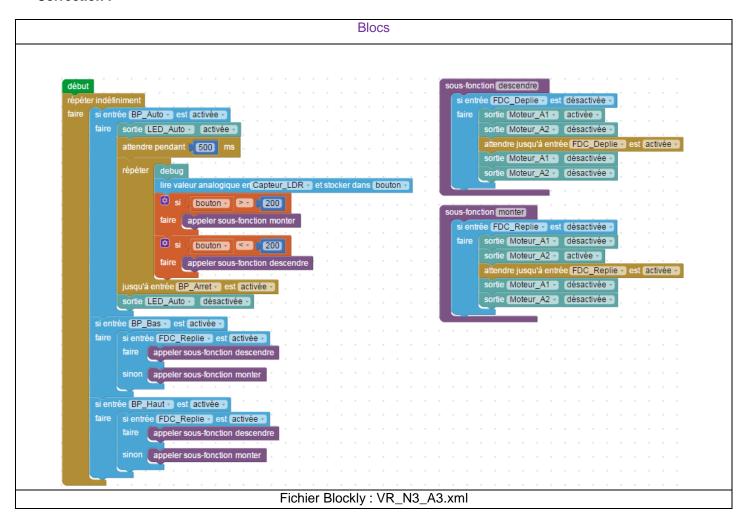
# Exercice niveau 3 - A.3 : Création d'un mode automatique

Objectif: Créer un mode automatique reprenant l'exercice A.2

En mode automatique : Une LED s'allume et le volet s'ouvre ou se ferme en fonction de l'éclairage En mode non-automatique : La LED est éteinte, montée ou descente du volet lors de l'appui sur un bouton Appuyer sur le bouton Auto pour lancer le mode automatique Appuyer sur le bouton d'arrêt pour sortir du mode automatique

Notion abordée : Gestion de variables analogiques, application

#### Correction:



**Remarque :** Sans les boucles Si dans les fonctions, lorsque le volet arrive en fin de course, il va sans cesse relancer une sous-fonction et donc activer et désactiver le moteur à chaque boucle.



## Exercice niveau 3 - A.4 : Ajout d'une sécurité

**Objectif :** Reprendre l'exercice précédent, lors d'un appui sur le bouton arrêt, que ce soir en mode automatique ou non, le volet doit s'arrêter jusqu'à l'appui du bouton d'ouverture ou de fermeture. A l'appui d'un de ces boutons, le programme reprendra la ou il était avant de s'arrêter.

Conseil: Améliorer les sous-fonctions

Notion abordée : Gestion de variables analogiques, application

```
Blocs
                                                                                                    onction descendre
                                                                                                 si entrée FDC_Deplie 🔻 est désactivée 🔻
si entrée BP_Auto 🔻 est activée 🔻
                                                                                                      sortie Moteur_A1 - activée -
faire sortie LED_Auto v activée v
                                                                                                        sortie (Moteur_A2 v désactivée v
       attendre pendant 500 ms
                                                                                                                 si entrée BP_Arret 🔻 est (activée 🔻
                                                                                                                 faire sortie Moteur_A1 v désactivée v
                                                                                                                         rtie Moteur_A2 v désactivée v
                ire valeur analogique er Capteur_LDR 🔻 et stocker dans bouton 🔻
               i | bouton - > 200
                                                                                                                 si entrée BP_Bas 🔻 est activée 🔻
                                                                                                                 faire sortie Moteur_A1 - activée -
                                                                                                                      sortie Moteur_A2 v désactivée v
                                                                                                                 si entrée BP_Haut 🔻 est activée 🔻
                                                                                                                      sortie Moteur_A1 v activée v
                                                                                                                      sortie Moteur_A2 v désactivée v
                 ée BP_Arret - est activée -
          ie LED_Auto - désactivée -
                                                                                                                    FDC_Deplie v est activée v
                                                                                                           tie Moteur_A1 - désactivée -
si entrée BP_Bas v est activée v
                                                                                                           tie Moteur_A2 v désactivée v
      si entrée (FDC_Replie v est activée
                                                                                                   fonction monter
                                                                                                 si entrée FDC_Replie 🔻 est désactivée 🔻
si entrée BP_Haut 🔻 est activée 🔻
                                                                                                       sortie Moteur_A1 v désactivée v
      si entrée (FDC_Replie » est activée
                                                                                                        sortie Moteur_A2 - activée -
                                                                                                                 si entrée BP_Arret 🔻 est (activée 🔻
                                                                                                                 faire sortie Moteur_A1 v désactivée v
                                                                                                                      sortie Moteur_A2 v désactivée v
                                                                                                                 si entrée BP_Haut • est activée •
                                                                                                                 faire sortie Moteur_A1 - désactivée -
                                                                                                                       sortie Moteur_A2 v activée v
                                                                                                                 si entrée BP_Bas v est activée v
                                                                                                                     sortie Moteur_A1 v désactivée v
                                                                                                                      sortie Moteur_A2 v activée v
                                                                                                                   e FDC_Replie - est activée -
                                                                                                           tie Moteur_A1 - désactivée -
                                                                                                          tie Moteur_A2 v désactivée v
                                                   Fichier Blockly: VR_N3_A4.xml
```



# **Option: Module Bluetooth**

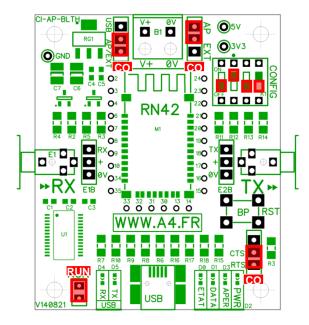
Le module Bluetooth développé par A4 Technologie permet de convertir le protocole Bluetooth en protocole de communication type Série qui est le mode de communication classique utilisé avec PICAXE ou Arduino. Ce module accepte différentes configurations.

En mode avancé, il peut être configuré au travers d'une liaison par connexion USB à un PC ou par l'envoi de commandes au travers de ses liaisons RX et TX.

La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module. Elle est téléchargeable sur <a href="http://a4.fr/wiki/index.php/Module\_Bluetooth\_-\_K-AP-MBLTH\_/\_S-113020008">http://a4.fr/wiki/index.php/Module\_Bluetooth\_-\_K-AP-MBLTH\_/\_S-113020008</a>. Les informations seront envoyées via un smartphone ou une tablette possédant la technologie bluetooth à l'aide d'une application développée sous Applnventor par l'équipe technique de A4.

### Configuration

Positionner les cavaliers et interrupteurs comme indiqué par les positions repérées en rouge ci-dessous.



- Le cavalier repéré RUN est utilisé lors de la mise au point de programmes avec Arduino.
   Il doit être ôté pour permettre le téléversement du programme puis doit être remis lors de l'utilisation.
- La mise au point de programmes avec **PICAXE** ne nécessite pas d'ôter ce cavalier pour transférer le programme.
- Les cavaliers **CO1** et **CO2** permettent de sélectionner le mode d'alimentation du module Bluetooth. Dans la configuration ci-dessus, son alimentation provient directement de l'interface AutoProg ou AutoProgUno au travers des cordons de liaison avec le module ; ils sont positionnés respectivement sur AP et sur AP/EXT.
- Le cavalier **CO3** est utilisé en mode avancé pour relier ou dissocier les signaux CTS et RTS nécessaires au fonctionnement du module Bluetooth. Ici, il est positionné sur CTS/RTS.
- Les interrupteurs CONFIG permettent de paramétrer le mode de fonctionnement du module Bluetooth. Ici, l'interrupteur n°2 est positionné sur ON pour sélectionner une vitesse de transmission des données à 9600 bauds.



#### **Témoins lumineux**

**PWR** indique que le module est sous tension.

**APER** indique que le module est associé avec un matériel Bluetooth.

DATA indique qu'il y a un flux de données entre le module et l'appareil avec lequel il est connecté.

**ETAT** indique que le module est opérationnel. L'affichage clignotant indique qu'il n'est pas opérationnel.

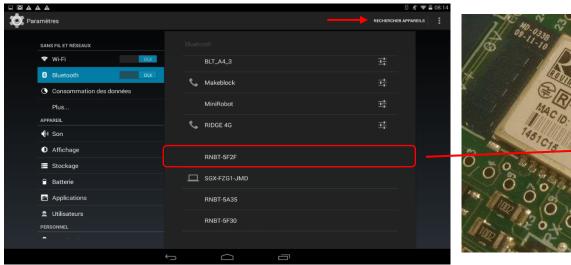
USB RX indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

USB TX indique qu'il y a un flux de données sur la liaison USB du module vers le PC.

### Mise en place des programmes et procédure de connexion

Avant de commencer à tester les programmes il faut d'abord appairer le smartphone ou la tablette au module bluetooth.

Pour cela rendez-vous dans les réglages bluetooth et lancer une recherche d'appareils (la maquette doit être allumée pour allimenter le module). Le nom de votre module s'appelle : RNBT + les 4 derniers chiffres de l'adresse mac du module notés sur le composant. Selectionnez le et un message proposant de vous connecter à lui devrait s'afficher.



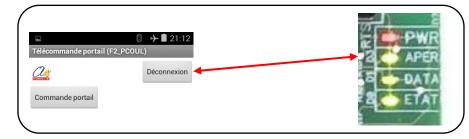


Une fois cette étape passée vous pourrez vous connecter au module à partir du programme Applnventor à chaque fois.

Lorsque la connexion est réalisée, le bouton **Déconnexion** apparaît dans l'application.

Le témoin vert **DATA** s'allume sur le module dès qu'une donnée est émise ou reçue par le module Bluetooth. L'appui sur le bouton d'envoi de données, dans cet exemple **Commande portail**, déclenche l'allumage fugitif de ce témoin.







### Tableau d'affectation des entrées et sorties

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	Bouton poussoir Mode auto	C.5	BP_Auto
4	Capteur fin de course dépliement (Fermeture)	C.4	FDC_Deplie
3	Capteur fin de course repliement (Ouverture)	C.3	FDC_Replie
2	Bouton poussoir ouverture	C.2	BP_Bas
1	Bouton poussoir arrêt	C.1	BP_Arret
0	Bouton poussoir fermeture	C.0	BP_Haut
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	Capteur d'intensité lumineuse	A.1	Capteur_LDR
0	Capteur de courant analogique/numérique (option)	A.0	Capteur_courant
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	(libre)	B.7	
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	LED	B.2	LED_Auto
1	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.1	Moteur_A2
0	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.0	Moteur_A1

Remarque : Changement de connectique sur les modules de communication C.6 et C.7 pour cet exercice

### Câblage du module bluetooth (K-AP-MBLTH)





# Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth

Objectif: contrôler l'ouverture et la fermeture du volet à l'aide de 2 boutons présent sur l'application Android.

Notion abordée : réception de données Bluetooth envoyées par un Smartphone.

Application Android: Volet\_1.apk

Fichier App Inventor: Volet 1.aia



```
Blocs
début
                                                                  sous-fonction monter
hsersetup B9600_8 T
                                                                     sortie Moteur_A2 activée
 Inverser la polarité
                                                                     sortie LED_Auto activée
répéter indéfiniment
                                                                     attendre jusqu'à entrée (FDC_Replie v est activée v
     hserin consigne -
                                                                     appeler sous-fonction arret
      si (consigne = 111
      faire appeler sous-fonction monter
                                                                  sous-fonction descendre
      sinon si ( consigne - 2 2
                                                                     sortie Moteur_A1 v activée v
                                                                     sortie LED_Auto Tactivée T
      faire appeler sous-fonction descendre
                                                                     attendre jusqu'à entrée FDC_Deplie v est activée v
      si entrée BP_Haut v est activée v
                                                                     appeler sous-fonction arret
      faire appeler sous-fonction monter
      si entrée BP_Bas ▼ est activée ▼
                                                                   sous-fonction arret
      faire appeler sous-fonction descendre
                                                                     sortie Moteur_A1 désactivée v
                                                                     sortie Moteur A2 désactivée
      fixer consigne à (0
                                                                     sortie LED Auto désactivée
                                       Fichier Blockly: VR_N3_B1.xml
```

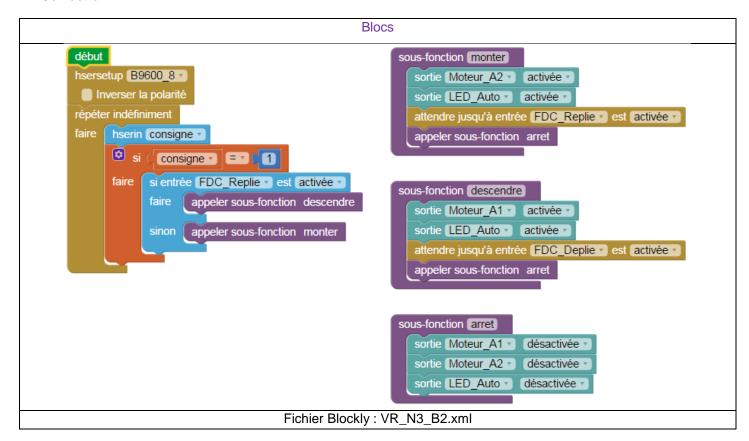


# Exercice niveau 3 - B.2 : Contrôle du volet par Smartphone

Objectif: monter et descendre le volet à partir d'un seul bouton disponible sur l'application Android.

Notion abordée : réception de données Bluetooth envoyées par un Smartphone.







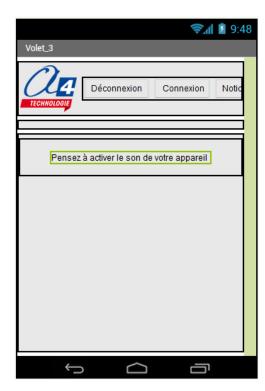
# Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone

Objectif: jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du volet.

Notion abordée : envoyer des informations à un Smartphone par Bluetooth.

Application Android : Volet\_3.apk
Fichier App Inventor : Volet\_3.aia

```
quand Clock1 v. Chronomètre
faire si Bluetooth v. Est connecté v
alors si papeler Bluetooth v. Octets disponibles pour le réception v. Octets disponibles pour le réception si papeler Bluetooth v. RecevoirOctetSignéNuméro1 v. 1
alors appeler ding v. Jouer
```



```
Blocs
début
hsersetup B9600_8 *
 Inverser la polarité
répéter indéfiniment
faire
      si entrée BP_Haut v est activée
       faire
              hserout |
              attendre pendant
                                500
       si entrée BP_Bas v est activée
              hserout
       faire
              attendre pendant
                                500
                 Fichier Blockly: VR N3 B3.xml
```

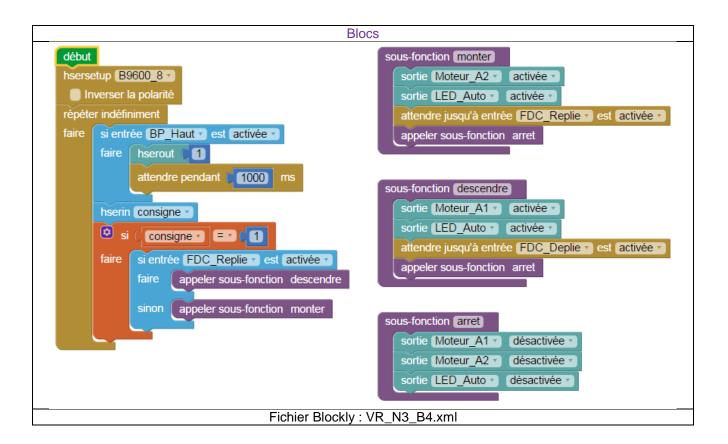


# Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone

Objectif: gérer la sonnette ainsi que le contrôle du volet à distance à l'aide de l'application Android.

**Notion abordée :** envoyer et recevoir des informations à l'aide du module Bluetooth à une application. **Correction :** 

🖘 🛮 9:48 Application Android: Volet\_4.apk Télécommande portail 4 App Inventor: Volet\_4.aia 4 Déconnexion ıand Clock1 ▼ Bluetooth . Est connecté . appeler Bluetooth . Octets disponibles pour le réception > (0) Attention pensez à activer le son de votre appareil alors si appeler Bluetooth v .RecevoirOctetSignéNuméro1 alors appeler ding .Jouer appeler Sonnerie . Afficher fenêtre choix Un bouton a été activé, que voulez-vous faire? ding " Monter/Descendre " Ne rien faire uand Sonnerie .Après choix si obtenir Choix v = v Monter/Descendre " alors appeler Bluetooth .Envoyer1Octet





# Exercice niveau 3 - B.5 : Envoyer et recevoir des données provenant d'un Smartphone afin de contrôler le volet à distance

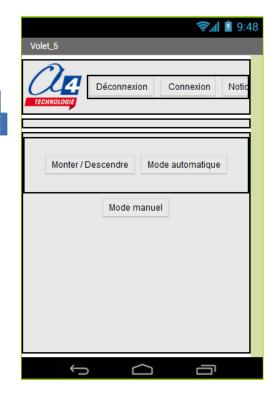
**Objectif :** Reprendre l'exercice A4 du niveau 3, permettre l'activation du mode automatique, de l'ouverture et de la fermeture à distance, ainsi qu'une désactivation du mode automatique

Notion abordée: envoyer et recevoir des informations à l'aide du module Bluetooth à une application.

Application Android: Volet\_5.apk
App Inventor: Volet\_5.aia

quand Clock1 Chronomètre
faire si Bluetooth Est connecté 
alors si appeler Bluetooth Octets disponibles pour le réception
alors appeler Bluetooth RecevoirOctetSignéNuméro1 
quand Mode\_auto Clic
faire appeler Bluetooth Envoyer1Octet
nombre 1

quand Mode\_manuel Clic
faire appeler Bluetooth Envoyer1Octet
nombre 3



```
Blocs
hserin consigne •
si consigne = 1
    sortie LED_Auto → activée →
     hserout [ 1
         dre pendant (500) ms
             hserin consigne •
                             e en Capteur_LDR • et stocker dans bouton •
             si bouton - < 200
            consigne - = 3
        rout 🖟 🚹
      sortie LED_Auto - désactivée -
     fixer (consigne • à (0)
sinon si Consigne - 2
      si entrée FDC_Replie est activée
     fixer consigne - à 🚛
```



```
sous-fonction descendre
                                                                        sous-fonction monter
  si entrée (FDC_Deplie - ) est désactivée -
                                                                            si entrée (FDC_Replie v est (désactivée v
  faire sortie Moteur_A1 v activée v
                                                                            faire sortie Moteur_A1 v désactivée v sortie Moteur_A2 v activée v
          sortie Moteur_A2 v désactivée v
                    si entrée BP_Arret v est activée v
                                                                                             si entrée BP_Arret v est activée v
                    faire sortie Moteur_A1 v désactivée v
                                                                                             faire sortie Moteur_A1 v désactivée v
                         sortie Moteur_A2 v désactivée v
                                                                                                   sortie Moteur_A2 v désactivée v
                    si entrée BP_Bas • est activée •
                                                                                             si entrée BP_Haut • est activée •
                    faire sortie Moteur_A1 v activée v sortie Moteur_A2 v désactivée v
                                                                                             faire sortie Moteur_A1 v désactivée v
                                                                                                   sortie Moteur_A2 v activée v
                    si entrée BP_Haut • est activée •
                                                                                             si entrée (BP_Bas - ) est activée -
                    faire sortie Moteur_A1 v activée v
                                                                                             faire sortie Moteur_A1 v désactivée v sortie Moteur_A2 v activée v
                         sortie Moteur_A2 v désactivée v
         jusqu'à entrée FDC_Deplie • est activée • sortie Moteur_A1 • désactivée • sortie Moteur_A2 • désactivée •
                                                                                  jusqu'à entrée FDC_Replie • est activée • sortie Moteur_A1 • désactivée •
                                                                                  sortie Moteur_A2 - désactivée -
                                       Fichier Blockly: VR_N3_B5.xml
```

# Option : Module capteur de courant

Ce module permet de sécuriser le fonctionnement d'un automatisme animé par un moteur à courant continu. En effet, lorsqu'un événement anormal se produit (ex : blocage du volet), la consommation de courant du moteur augmente. La détection de la surintensité au-delà d'un seuil permet de déclencher l'arrêt du moteur et ainsi mettre le système en sécurité.

Le module capteur de courant peut fonctionner selon deux modes.

#### Mode analogique:

Le cavalier S1 est en position ANA. La sortie MESURE renvoie une tension proportionnelle au courant circulant dans le module. Cette valeur est lue sur une entrée analogique du boîtier.

#### Mode numérique (tout ou rien) :

Le cavalier S1 est en position NUM. Le potentiomètre SEUIL permet de régler un seuil de courant au-delà duquel la sortie MESURE basculera. Si le courant circulant dans le module dépasse le seuil, la LED témoin et la sortie MESURE s'activent, sinon elles sont inactives.

Le potentiomètre INIT permet d'adapter la plage de mesure du module en fonction du contexte d'utilisation (consommation plus ou moins élevé de courant selon matériel utilisé).

La mesure de courant se fait au travers de de la connectique jack (AUTOPROG / ACTIONNEUR) ou bien directement via le bornier à vis ACTIONNEUR.

#### Mesure via connectique jack :

Le courant mesuré correspond au courant total qui circule dans le montage (courant consommé par l'interface AutoProg, les modules et le moteur).

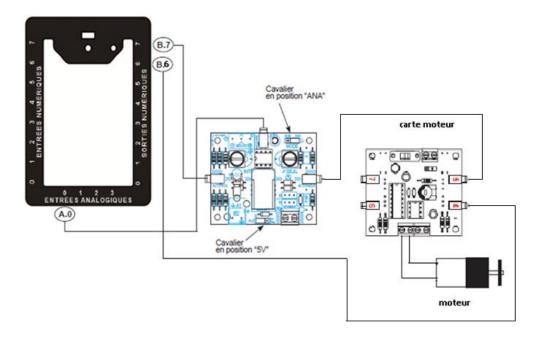
#### Mesure via bornier à vis :

Le courant mesuré correspond à celui circulant dans l'élément connecté au bornier (moteur).

Dans la suite des exercices le capteur de courant est utilisé en mode Analogique et la mesure est faite via la connectique jack. Il s'agit de détecter les surintensités dans le moteur ; on considère que la consommation de l'interface et des modules est négligeable par rapport à celle du moteur. Ce mode de d'utilisation facilite le câblage.

L'interface AutoProg est alimenté par un bloc d'alimentation externe afin de garantir une tension constante de 5V dans l'ensemble du montage.

### Mise en service du module





### Réglage du potentiomètre d'initialisation :

Il s'agit de régler la tension de sortie du module de telle sorte qu'elle soit maximale lorsque le moteur et bloquée (consommation maximum) afin de travailler sur plage de mesure adaptée au moteur du volet.

En fonctionnement à vide (débrayé du volet) le moteur consomme jusqu'à 250 mA à vitesse moyenne. Si le moteur est bloqué, sa consommation monte au-delà de 500 mA.

Le réglage s'effectue en débrayant le moteur de la barrière (dérailler la barrière pour libérer le moteur de toute contrainte mécanique).

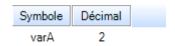
 Positionner l'interrupteur du module moteur sur OFF puis mettre le cavalier du moteur de Vext sur Vint pour que le moteur soit alimenté directement par l'interface.
 Positionner le cavalier S1 du module Capteur de courant sur ANA et le cavalier S2 sur la position 5V (Schéma)

Relier les embases jack A1 et A2 du module moteur respectivement sur les sorties B.6 et B.7 de l'interface AutoProg. Relier l'embase jack MESURE du module capteur de courant l'entrée analogique A.0 de l'interface AutoProg puis charger le programme de test **Test\_Capteur\_Courant.xml** 

Ce programme permet de visualiser en direct à l'écran une valeur (varA) qui est proportionnelle à celle du courant consommé par le moteur.



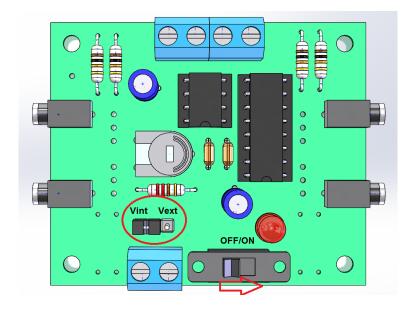
- 2) Positionner l'interrupteur du module moteur sur ON : le moteur doit tourner Ajuster le potentiomètre F-MOTA du module moteur à une position intermédiaire pour fixer la vitesse de rotation du moteur.
- 3) Ajuster le potentiomètre MESURE du module capteur de courant jusqu'à avoir une valeur de varA comprise entre 0 et 5



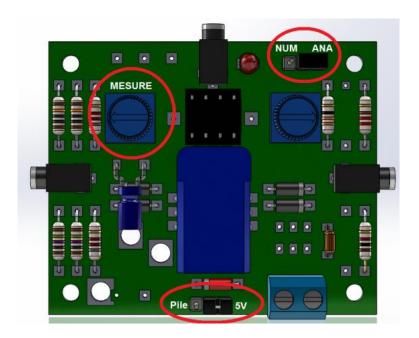
4) Bloquer le moteur manuellement en retenant la roue dentée d'entrainement du volet. La valeur de varA doit augmenter de façon significative (augmentation brutale du courant consommé par le moteur).

Cette procédure permet de vérifier la bonne configuration du montage et de visualiser la valeur du courant circulant dans le moteur en vue de déterminer le seuil au-delà duquel on considère que celui-ci à consommation excessive (blocage).





Module pilotage moteurs (REF: K-AP-MMOT-M)

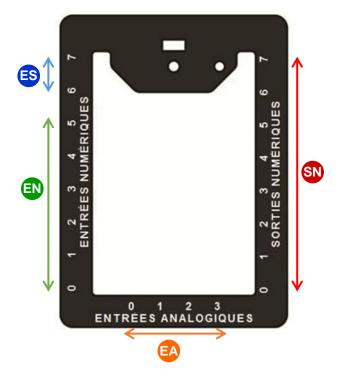


Module Capteur de courant (REF : K-AP-MAMP-M)



## Tableau d'affectation des entrées et sorties

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	Bouton poussoir Mode auto	C.5	BP_Auto
4	Capteur fin de course dépliement (Fermeture)	C.4	FDC_Deplie
3	Capteur fin de course repliement (Ouverture)	C.3	FDC_Replie
2	Bouton poussoir ouverture	C.2	BP_Bas
1	Bouton poussoir arrêt	C.1	BP_Arret
0	Bouton poussoir fermeture	C.0	BP_Haut
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	Capteur d'intensité lumineuse	A.1	Capteur_LDR
0	Capteur de courant analogique/numérique (option)	A.0	Capteur_courant
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	(libre)	B.7	
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	LED	B.2	LED_Auto
1	Connecté à la broche MOTA-2 de la carte contrôle moteur	B.1	Moteur_A2
0	Connecté à la broche MOTA-1 de la carte contrôle moteur	B.0	Moteur_A1





## Exercice niveau 3 - C.1: Utilisation du capteur de courant

Objectif : allumer le moteur et lire l'entrée capteur de courant

Notion(s) abordée(s) : lire une entrée analogique.

**Sécurité**: Si le moteur ne s'arrête pas, il faut débrancher le fil de communication relié au moteur. Cela est nécessaire en fin de course afin de ne pas abîmer la maquette. Remonter ou redescendre ensuite le volet à le main à l'aide des roues.

### Correction:

```
début
répéter indéfiniment
faire debug
lire valeur analogique en Capteur_Courant vet stocker dans varA voice Moteur_A1 vactivée v

Fichier Blockly: VR_N3_C1.xml
```

**Remarque**: Bloquer légèrement le moteur. Lorsqu'un effort est apporté au moteur, sa demande de courant augmente. Plus la résistance opposante au moteur sera grande, plus il va consommer de courant. Trouver la valeur de seuil correspondant à un blocage excessif sur le volet (fenêtre de debogage avec varA).



# Exercice niveau 3 - C.2 : Capteur de courant et variable

Objectif: arrêter le moteur lors de la détection d'un blocage.

Notion abordée : lire et interpréter une donnée d'une entrée analogique.

**Correction:** 

```
début
répéter indéfiniment
faire debug
sortie Moteur_A1 v activée v
lire valeur analogique en Capteur_Courant v et stocker dans varA v
si varA v > v 80
faire sortie Moteur_A1 v désactivée v
```

**Remarque**: La valeur de seuil sélectionnée ici est 80, elle peut varier entre 0 et 255 mais il est conseillé de régler le potentiomètre du gain afin d'avoir une valeur de courant compris entre 50 et 150 pour une consommation à vide du moteur.



# Exercice niveau 3 – C.3 : Sécurité du volet par contrôle du capteur de courant

Objectif: reprendre le programme VR\_N2\_A4 et y ajouter le capteur de courant.

Notion abordée : gérer une séquence d'instruction complète.

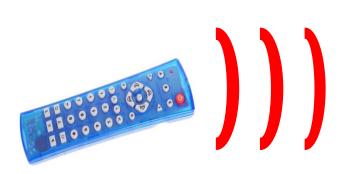
```
Blocs
                                                         sous-fonction descendre
début
appeler sous-fonction descendre
                                                            sortie (Moteur_A1 - activée -
répéter indéfiniment
                                                            sortie Moteur_A2 v désactivée v
faire fixer (bouton - à ( 0
                                                            tant que entrée (FDC_Deplie - est désactivée -
                                                                  si entrée BP_Arret est activée
               si entrée BP_Bas - est activée -
                                                                  faire appeler sous-fonction arret
                     fixer bouton - à [1]
                                                                  basculer LED_Auto -
                                                                  lire valeur analogique er Capteur_Courant vet stocker dans varA v
               si entrée BP_Haut v est activée v
                                                                  si (varA → ) > → ( 80)
                     fixer (bouton - à [1]
                                                                  faire sortie Moteur_A1 - désactivée -
                                                                         sortie Moteur_A2 - désactivée
      jusqu'à bouton - = 1
      si entrée FDC_Replie v est activée v
                                                                  attendre pendant 500 ms
      faire appeler sous-fonction descendre
                                                            sortie (Moteur_A1 - désactivée -
      sinon appeler sous-fonction monter
                                                            sortie Moteur_A2 désactivée
                                                            sortie LED_Auto - désactivée
sous-fonction arret
                                                         sous-fonction monter
                                                            sortie Moteur_A1 - désactivée -
   sortie Moteur_A1 - désactivée -
   sortie Moteur_A2 - désactivée -
                                                            sortie (Moteur_A2 - activée -
   sortie LED_Auto - désactivée -
                                                                    si entrée BP_Arret - est activée -
                                                                    faire appeler sous-fonction arret
                                                                    basculer LED_Auto -
                                                                     lire valeur analogique en Capteur_Courant - et stocker dans varA -
                                                                     🔯 si 💮 varA 🕶 🔼 🔞 🗑
                                                                          sortie Moteur_A1 - désactivée -
                                                                           sortie Moteur_A2 v désactivée v
                                                                     attendre pendant (500) ms
                                                            jusqu'à entrée FDC_Replie v est activée v
                                                            sortie Moteur_A1 désactivée
                                                            sortie Moteur_A2 désactivée
                                                            sortie LED_Auto - désactivée -
                                            Fichier Blockly: VR N3 C3.xml
```

# Option: Module télécommande infrarouge

## Télécommande RAX-TVR10

La télécommande universelle infrarouge associée à un capteur infrarouge approprié permet de piloter à distance une carte PICAXE.

Afin d'assurer la compatibilité de fonctionnement avec le système PICAXE il est nécessaire de l'initialiser avec le mode de fonctionnement au standard « Sony TV ».





Attention: L'émetteur infrarouge doit toujours être désactivé lors de l'utilisation de la télécommande infrarouge.

## Procédure de mise en service pour PICAXE

- 1. Insérer 2 piles AAA dans le logement au dos de la télécommande.
- 2. Appuyer simultanément sur S et B.
- = La LED s'allume.
- 3. Taper le code 0 1 3
- = La LED clignote brièvement à chaque appui des touches « 0 » et « 1 » puis s'éteint après l'appui sur la touche « 3 ».
- 4. Appuyer sur le bouton de mise en service.
- = La télécommande est opérationnelle.



Les touches suivantes risquent de déprogrammer :











Conseil : Si la télécommande ne fonctionne plus, appuyer sur B pour revenir à la configuration compatible PICAXE

Remarque : Le guide d'utilisation complet de la télécommande est disponible ici : http://www.a4telechargement.fr/RAX-TVR010/RAX-TRV10\_Telecommande\_InfraRouge.pdf

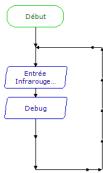




### Tester la télécommande

Charger les programmes de test de la télécommande : « test\_infra\_bloc.xml » ou « test\_infra\_org.plf ». Respecter le plan de câblage vu précédemment dans le dossier.





## Tableau de correspondance des touches

Diriger la télécommande vers le récepteur infrarouge et vérifier dans la partie « Variables » de Picaxe Editor que les données reçues sont correctes.

Ci-dessous, le tableau des valeurs renvoyées par les différents boutons de la télécommande :

Touche	1	2	3	4	5	6	7	8	9	0	G
Code émis	0	1	2	3	4	5	6	7	8	9	21
		$\bowtie$	<b>(</b>	<b>&gt;&gt;</b>		<u>\</u>	1	×	-	+	
Code émis	16	17	19	18	96	54	37	20	98	11	



## **Télécommande TELEC-IR-UNIV**

## Procédure de mise en service pour PICAXE

- 1. Insérer 2 piles AAA dans le logement au dos de la télécommande.
- 2. Appuyer simultanément sur les boutons **Set** et **TV**. Le bouton **Power** s'allume.
- 3. Taper le code 0-7-7. Le bouton **Power** clignote brièvement à chaque appui, puis s'éteint.
- Appuyer sur le bouton **Power**.
   La télécommande est opérationnelle.

#### ATTENTION !

La touche **DVB** risque de changer le mode. Appuyer sur **TV** pour revenir dans le bon mode.



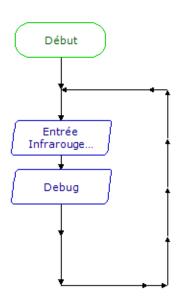
<u>Conseil</u> : Si la télécommande ne fonctionne plus, appuyer sur **TV** pour revenir à la configuration compatible PICAXE.



### Tester la télécommande

Charger les programmes de test de la télécommande : « test\_infra\_bloc.xml » ou « test\_infra\_org.plf ». Respecter le plan de câblage vu précédemment dans le dossier.







## Tableau de correspondance des touches

L'appui sur une touche provoque l'émission d'un signal infrarouge qui véhicule un code correspondant à la touche. Par défaut : appui sur la touche 1 = envoi du code 0.

Pour simplifier l'utilisation de la télécommande infrarouge, il est possible d'activer la compatibilité entre le N° des touches et le code envoyé. Dans ce cas, appui sur la touche 1 = envoi du code 1.

Touche	0	1	2	3	4
Code émis standard	9	0	1	2	3
Compatibilité activée	10	1	2	3	4
Touche	5	6	7	8	9
Code émis standard	4	5	6	7	8
Compatibilité activée	5	6	7	8	9
Touche		A·B	SCAN	AV	
Code émis standard	12	37	16	37	56
Compatibilité activée	13	38	17	38	57
Touche				Power	
Code émis standard	63	74	29	21	76
Compatibilité activée	64	75	30	22	77
Touche				VOL+	- VOL
Code émis standard	77	78	79	18	19
Compatibilité activée	78	79	80	19	20
Touche	(ME)	ĈH	CH		
Code émis standard	20	16	17		
Compatibilité activée	21	17	18		



# Exercice niveau 3 - D.1 : Activer un voyant lumineux avec la télécommande IR

Objectif: Allumer ou éteindre la LED en fonction d'une consigne envoyée

Notion abordée : gestion d'une liaison infra-rouge : télécommande/AutoProg à l'aide du bloc prévu à cet effet.

Correction:

```
début
répéter indéfiniment
faire lire valeur infrarouge Recepteur_IR v et stocker dans consigne v
v télécommande infrarouge (+1)
Temps écoulé
si consigne v = v 2
faire sortie LED_Auto v désactivée v
sinon si consigne v = v 1
faire sortie LED_Auto v activée v
```

**Remarque :** Cocher la case +1 permet d'avoir une concordance entre la touche pressée et la consigne reçue. La consigne 17 correspond à la flèche vers le haut, tandis que la consigne 18 correspond à la flèche vers le bas.

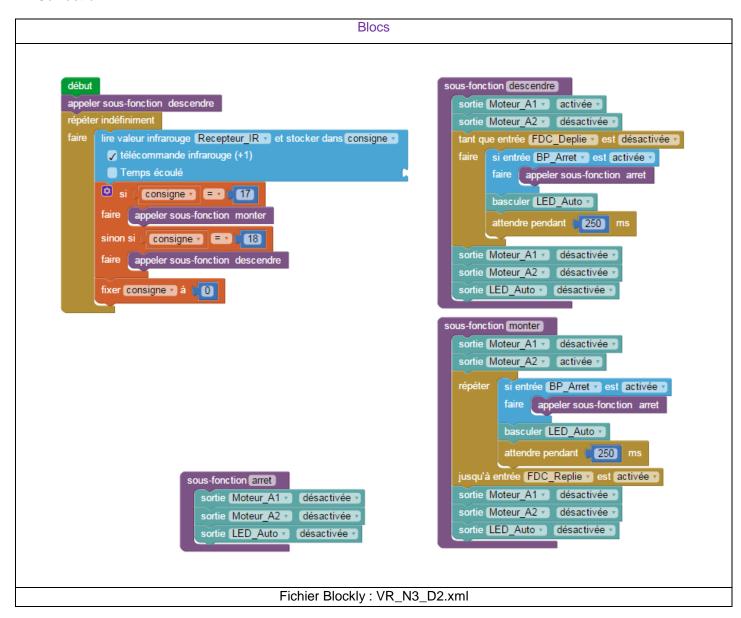


# Exercice niveau 3 - D.2 : Contrôle d'ouverture/fermeture avec la télécommande IR

Objectif: ouvrir et fermer le volet à l'aide de la télécommande IR.

Notion abordée: gestion d'une liaison infra-rouge: télécommande/AutoProg à l'aide du bloc prévu à cet effet.

Correction:



**Remarque :** Cocher la case +1 permet d'avoir une concordance entre la touche pressée et la consigne reçue. La consigne 17 correspond à la flèche vers le haut, tandis que la consigne 18 correspond à la flèche vers le bas.

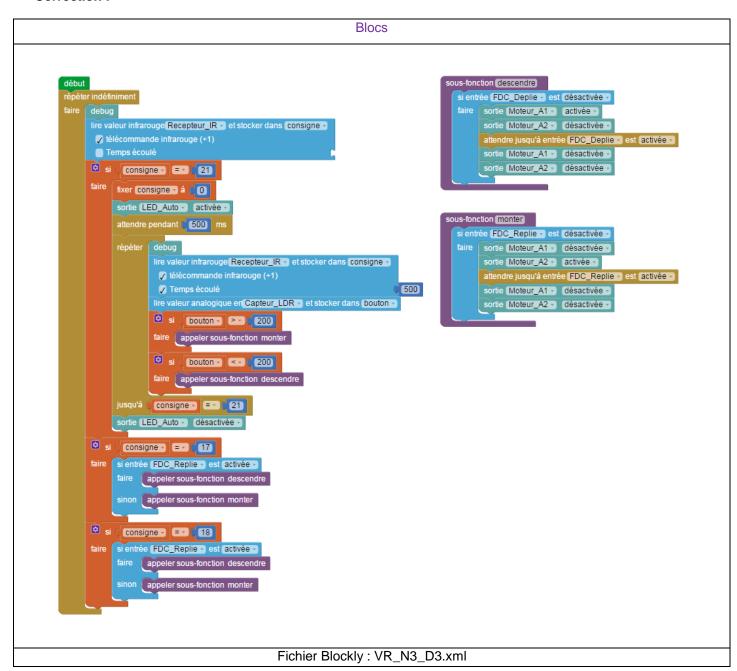


# Exercice niveau 3 - D.3 : Contrôle d'ouverture/fermeture avec la télécommande IR

**Objectif :** Reprendre l'exercice VR\_N3\_A3.xml, un bouton doit pouvoir activer/désactiver le mode automatique, les flèches haut et bas doivent monter ou descendre en fonction de sa position d'origine.

Notion abordée: gestion d'une liaison infra-rouge et introduction du timeout (Temps écoulé)

Correction:



**Remarque :** Le temps écoulé donne un temps pour activer la télécommande. Sans celui-ci, le programme restera bloqué sur la fonction jusqu'à ce qu'une commande soit envoyée. Le temps écoulé permet donc ici de récupérer les valeurs du capteur toutes les 500ms tout en pouvant changer la consigne afin de sortir du mode automatique.



# Exercice complémentaire (bon niveau) : Gestion complète du Volet roulant à l'aide des boutons

Objectif: reprendre le programme VR\_N3\_A4.

Après un appui sur le bouton d'arrêt, le volet pourra d'ouvrir ou se fermer en fonction du bouton sur lequel on appui. Le programme pourra changer de sens si on appui sur le bouton opposé.

Indice 1: Utiliser les sous fonctions

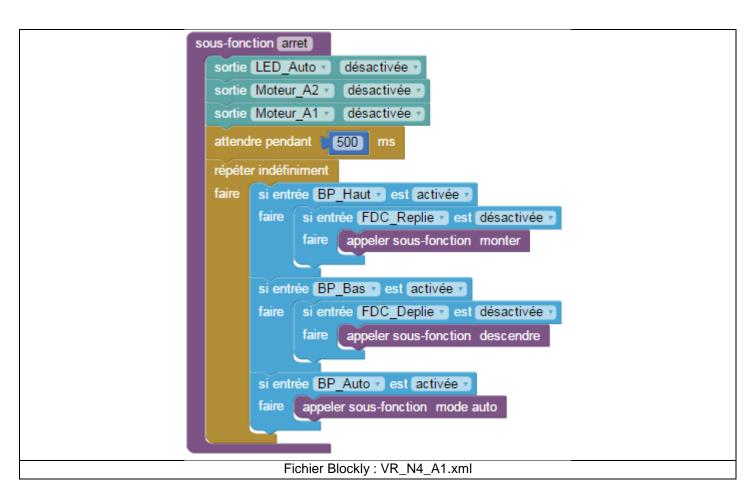
Indice 2 : Créer une sous-fonction pour l'arrêt

Indice 3 : Créer une sous-fonction pour le mode automatique

Indice 4 : La fonction arrêt doit pouvoir appeler les trois autres fonctions

```
Blocs
                                                                                  sous-fonction descendre
                                                                                     si entrée FDC_Deplie v est désactivée v
      si entrée BP_Auto v est activée v
                                                                                     faire sortie Moteur A1 activée
                                                                                            sortie Moteur_A2 v désactivée v
      faire appeler sous-fonction mode auto
                                                                                            attendre pendant 500 ms
      si entrée BP_Bas ▼ est activée ▼
            si entrée [FDC_Replie v est désactivée v
                                                                                                     si entrée BP_Arret v est activée v
             faire appeler sous-fonction monter
                                                                                                     faire appeler sous-fonction arret
                                                                                                     si entrée BP_Haut v est activée
      si entrée BP_Haut v est activée v
                                                                                                     faire appeler sous-fonction monter
           si entrée (FDC_Deplie v est désactivée v
             faire appeler sous-fonction descendre
                                                                                                     si entrée BP_Auto v est activée v
                                                                                                     faire appeler sous-fonction mode auto
                                                                                                       e FDC_Deplie v est activée v
sous-fonction monter
                                                                                                 Moteur_A1 ▼
  si entrée FDC_Replie ▼ est désactivée ▼
                                                                                                               désactivée
  faire sortie Moteur_A1 v désactivée v
                                                                                            sortie Moteur_A2 v désactivée v
         sortie Moteur_A2 v activée v
         attendre pendant 🄰 500 ms
                                                                                  sous-fonction mode auto
                  si entrée BP_Arret v est activée v
                                                                                     sortie LED Auto v activée v
                       appeler sous-fonction arret
                                                                                     attendre pendant $\int_{000}$ ms
                  si entrée BP_Bas 🔻 est activée 🔻
                       appeler sous-fonction descendre
                                                                                              lire valeur analogique en Capteur_LDR v et stocker dans varA v
                  si entrée BP_Auto v est activée v
                                                                                              ② si varA → > → (200)
                       appeler sous-fonction mode auto
                                                                                                    appeler sous-fonction monter
         jusqu'à entrée [FDC_Replie v est activée v
                                                                                                      varA - < - 200
         sortie Moteur_A1 v désactivée v
                                                                                                     appeler sous-fonction descendre
         sortie Moteur_A2 désactivée
                                                                                          LED_Auto • désactivée •
```





**Remarque :** Si on reste appuyé sur le bouton d'ouverture ou de fermeture, le moteur fonctionnera par acoups car il entrera sans cesse dans la boucle d'arrêt. (à éviter en fin de course)



**CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES**