

# Volet roulant

## Maquette programmable avec mBlock



Arduino - générer le code

répéter jusqu'à Bouton-poussoir BP\_Haut appuyé

Monter Volet

attendre 3 secondes

Descendre Volet

attendre 3 secondes

Arrêter Volet

# 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 ;
- Une notice d'utilisation de l'**option Bluetooth** ;

## Logiciels mBlock et App Inventor

- Notice d'installation d'une extension dans mBlock.
- Notice d'utilisation App Inventor 2.

## Activités / Programmation

- Fichiers modèles et fichiers de correction des programmes pour mBlock et AppInventor.

**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).
- **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 <http://creativecommons.fr/>**

*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 [www.a4.fr](http://www.a4.fr)**

# SOMMAIRE

<b>Introduction .....</b>	<b>2</b>
Volet roulant.....	2
Les environnements de programmation graphique.....	2
Le dossier .....	2
Les fiches exercices .....	3
Prérequis .....	3
Tableau d'affectation des entrées et sorties.....	4
Programmes de test .....	5
<b>Programmation version de base niveau 1 .....</b>	<b>6</b>
<b>Niveau 1 - A.....</b>	<b>7</b>
Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux.....	7
Exercice niveau 1 - A.2: Répéter une action deux fois.....	8
Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment.....	9
<b>Niveau 1 - B.....</b>	<b>10</b>
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur.....	10
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que .....	11
<b>Niveau 1 - C.....</b>	<b>12</b>
Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir.....	12
Exercice niveau 1 - C.2 : Contrôle moteur ET voyant lumineux.....	13
<b>Niveau 1 - D.....</b>	<b>14</b>
Exercice niveau 1 - D.1 : Utilisation des variables .....	14
Exercice niveau 1 - D.2 : Utiliser et tester une variable.....	15
Exercice niveau 1 - D.3 : Tests /variables .....	16
<b>Programmation version de base niveau 2 .....</b>	<b>17</b>
Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses .....	18
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture.....	19
Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP et signal de sécurité .....	20
Exercice niveau 2 - A.4 : Contrôle d'ouverture/fermeture avec BP, signal de sécurité.....	21
<b>Programmation niveau 3.....</b>	<b>22</b>
<b>Option : capteur de lumière .....</b>	<b>23</b>
Exercice niveau 3 - A.1 : Détection de lumière .....	23
Exercice niveau 3 - A.2 : Gérer l'ouverture et la fermeture en fonction de l'intensité lumineuse .....	24
Exercice niveau 3 - A.3 : Création d'un mode automatique .....	25
Exercice niveau 3 - A.4 : Ajout d'une sécurité .....	26
<b>Option : Module Bluetooth.....</b>	<b>27</b>
Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth .....	29
Exercice niveau 3 - B.2 : Contrôle du volet par Smartphone .....	30
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone .....	31
Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone .....	32
Exercice niveau 3 - B.5 : Envoyer et recevoir des données provenant d'un Smartphone afin de contrôler le volet à distance.....	33

# 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 : capteur de lumière et module Bluetooth.

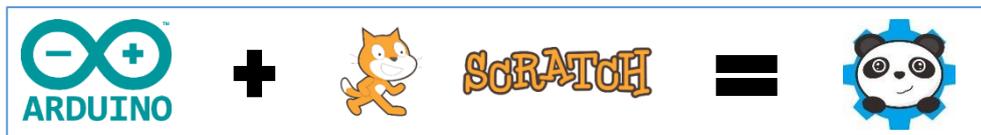
## Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette AutoAlarme ont été réalisés sous **mBlock**.



mBlock est un IDE développé par Makeblock, reprenant la base de Scratch avec l'ajout de blocs permettant le contrôle d'une carte Arduino.

mBlock permet également de créer ses propres blocs dans une extension **A4\_Alarme** (fichier zip), des blocs simples et intuitifs présents permettant de prendre en main la maquette rapidement.



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 mBlock.

## 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 : capteur de lumière et 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 de correction qui propose un exemple de programme réalisé sous mBlock (extension .sb2).

Deux approches :

- Avec les exemples de programmes, les utilisateurs découvrent les principes de la programmation graphique 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\_N3\_A1.sb2

Correspond au niveau 3 avec le jeu d'instructions A, adapté aux objectifs « avancés » de ce niveau.

## Prérequis

Pour la version de base :

- Installer le logiciel **mBlock**.
- Installer l'extension **A4\_Volet** (fichier zip) dans mBlock.
- **Maquette** Volet roulant (Réf. BE-AVOL-ROUL).
- **Câble de programmation** USB (Réf : CABL-IMPUSB).
- **Interface programmable** AutoProgUno (Réf. K-AP-UNO).
- **Cordons de liaison** jack compatibles AutoProg pour établir les liaisons entre l'interface programmable et la maquette.

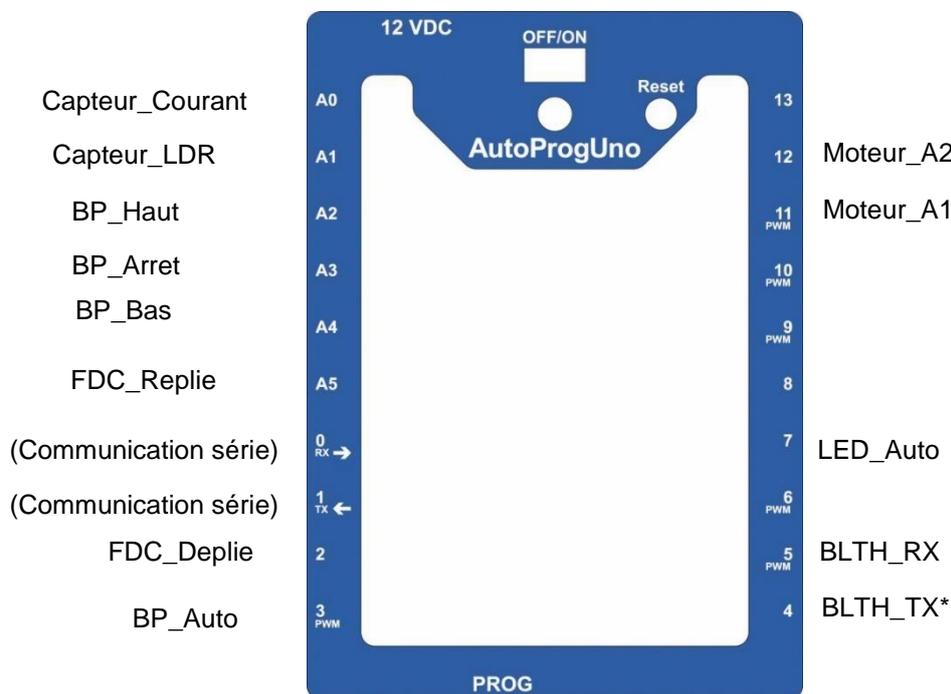
Pour l'option Bluetooth :

- **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.

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

## Tableau d'affectation des entrées et sorties

AutoProgUno	Monte-charge	Nom mBlock
<b>MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES</b>		
2	Fin de course dépliement	FDC_Deplie
3	Bouton poussoir mode auto	BP_Auto
4	Communication BT (Option)	BLTH_TX*
5	Communication BT (Option)	BLTH_RX
<b>MODULES ACTIONNEURS POUR SORTIES NUMÉRIQUES</b>		
6		
7	Voyant lumineux mode auto	LED_Auto
8		
9		
10		
11	Borne A1 du moteur	Moteur_A1
12	Borne A2 du moteur	Moteur_A2
13		
<b>MODULE DE COMMUNICATION</b>		
0	(communication avec ordinateur)	
1		
<b>ENTRÉES / SORTIES LIBRES (A pour les analogiques)</b>		
A0	Capteur de courant	Capteur_Courant
A1	Capteur de lumière	Capteur_LDR
A2	Bouton poussoir haut	BP_Haut
A3	Bouton poussoir arrêt	BP_Arret
A4	Bouton poussoir bas	BP_Bas
A5	Fin de course repliement	FDC_Replie



## 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](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/](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	Comportement attendu
1	Ouvrir le programme <b>VR_N3_A4</b>	
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 <b>arrêt</b>	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 <b>auto</b>	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

# Programmation version de base niveau 1

## Objectifs :

- Découvrir et maîtriser le matériel avec des exemples très simples pour débiter 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.

Nom du fichier	Description	Objectif
<b>Niveau 1 A</b>		
VR_N1_A1.sb2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordée : -Allumage/extinction du voyant lumineux Notions de programmation abordées : -séquence d'instructions -temps d'attente -boucle infinie
VR_N1_A2.sb2	Répéter cette même action deux fois.	
VR_N1_A3.sb2	Répéter cette action à l'infini.	
<b>Niveau 1 B</b>		
VR_N1_B1.sb2	Activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.	Fonctionnalité matérielle abordée : -Gestion du moteur -Utilisation de Bouton-poussoir Notions de programmation abordées : -boucle qui dépend d'une entrée
VR_N1_B2.sb2	Ouvrir et fermer le volet en continu jusqu'à l'appui d'un bouton-poussoir.	
<b>Niveau 1 C</b>		
VR_N1_C1.sb2	Allumer le voyant lumineux à l'appui du BP.	Fonctionnalité matérielle abordée : -Gestion des modules infrarouge -Utilisation de Bouton-poussoir Notions de programmation abordées : -Le test d'une entrée (si/sinon)
VR_N1_C2.sb2	Activer le voyant lumineux lorsque le capteur infrarouge est franchi.	
<b>Niveau 1 D</b>		
VR_N1_D1.sb2	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débugage).	Notions de programmation abordées : -Définition de variable -Incrémenter de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débugage
VR_N1_D2.sb2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	
VR_N1_D3.sb2	Incrémenter une variable à l'appui d'un bouton poussoir, allumer la LED au bout de 10 appuis.	

# Niveau 1 - A

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

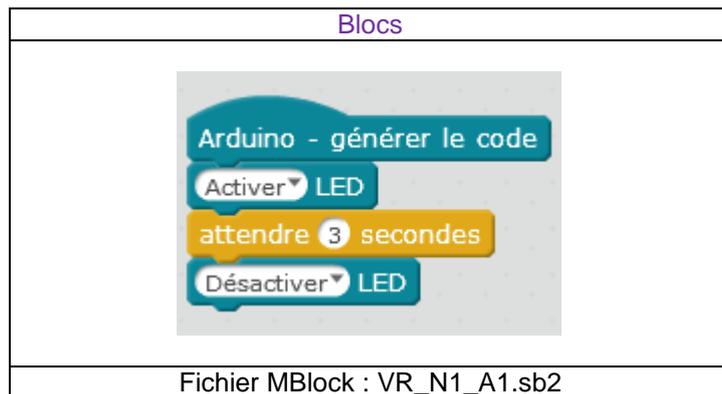
**Objectif** : allumer le voyant lumineux pendant 3 secondes puis l'éteindre.

**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** :



**Correction** :

Blocs

Le programme de correction est une séquence de blocs mBlock : 'Arduino - générer le code' (bleu), 'Activer LED' (bleu), 'attendre 3 secondes' (orange), 'Désactiver LED' (bleu), 'attendre 3 secondes' (orange), 'Activer LED' (bleu), 'attendre 3 secondes' (orange), et 'Désactiver LED' (bleu).

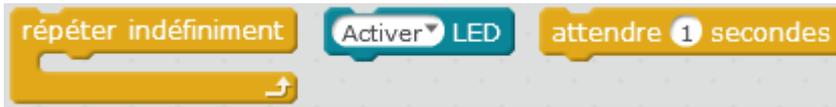
Fichier MBlock : VR\_N1\_A2.sb2

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

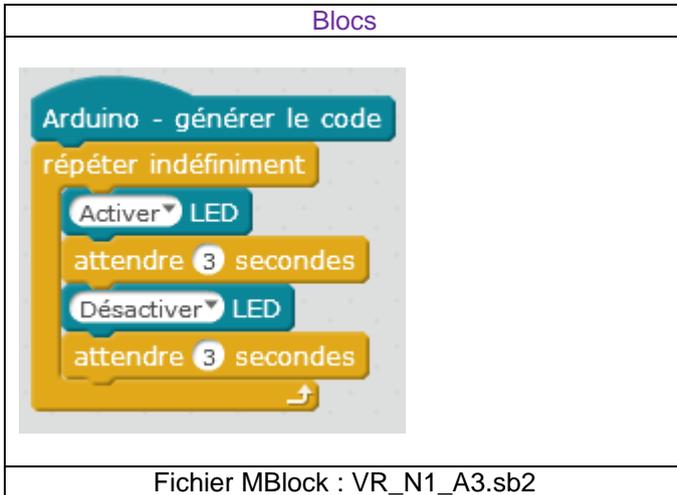
**Objectif** : faire clignoter le voyant lumineux avec une période de 6 secondes indéfiniment.

**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.

# Niveau 1 - B

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

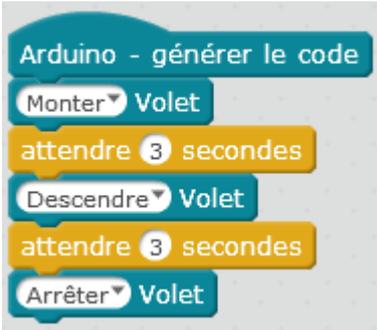
**Objectif** : activer un moteur dans un sens puis dans l'autre pour enfin s'arrêter.

**Notion abordée** : utilisation d'un moteur.

**Instructions utilisées** :



**Correction** :

Blocs

Fichier MBlock : VR_N1_B1.sb2

**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.

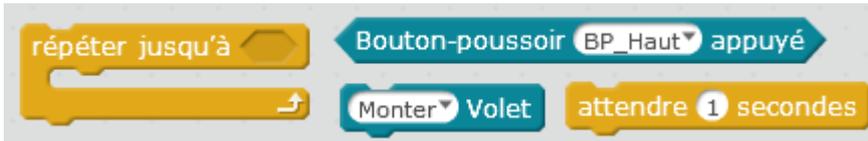
Il faut absolument éviter de faire sortir le volet de son rail. Couper l'AutoProg dans le cas où le volet sortirait du rail.

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

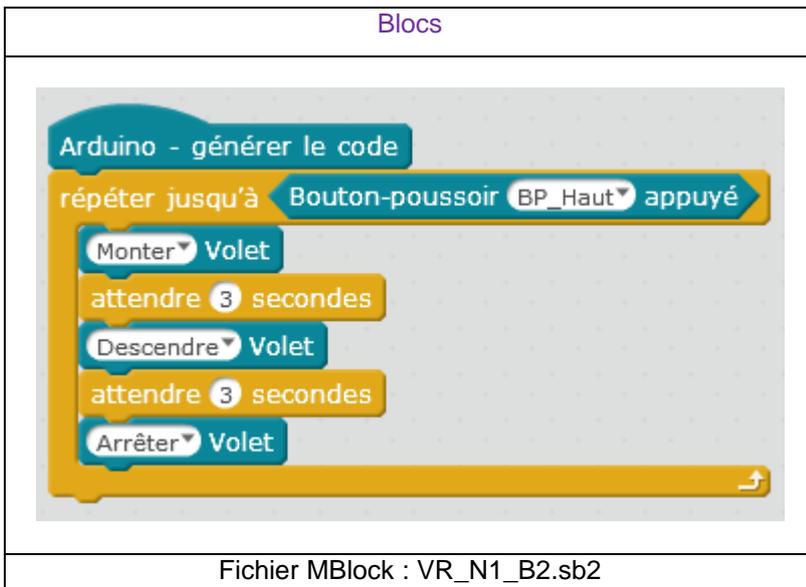
**Objectif** : ouvrir et fermer le volet en continu jusqu'à l'appui d'un bouton-poussoir.

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

**Instructions utilisées** :



**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).

# Niveau 1 - C

## Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir

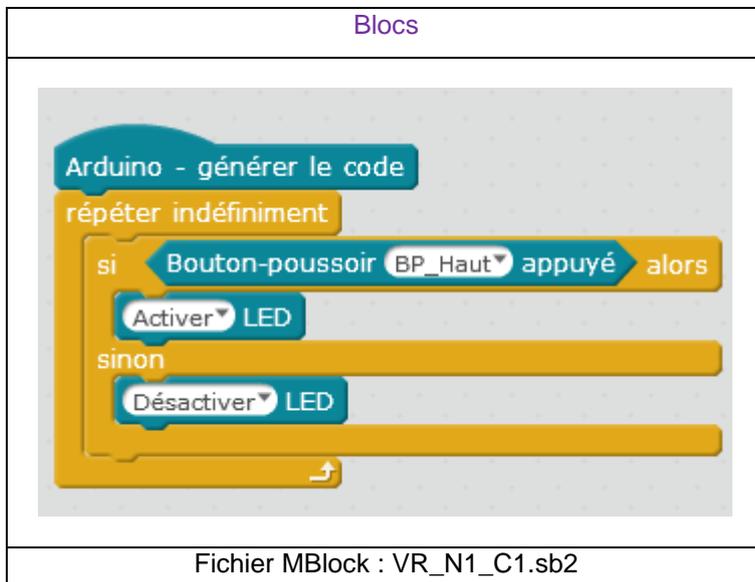
**Objectif** : allumer le voyant lumineux à l'appui du BP.

**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.

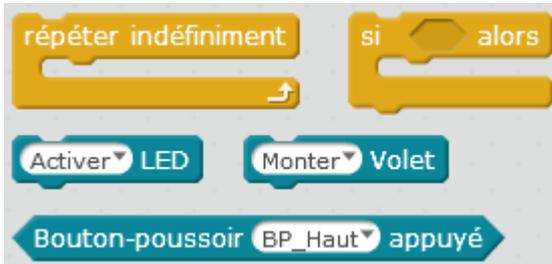
## Exercice niveau 1 - C.2 : Contrôle moteur ET voyant lumineux

**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** :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Arret appuyé alors
    Arrêter Volet
    Désactiver LED
  si Bouton-poussoir BP_Bas appuyé alors
    Descendre Volet
    Activer LED
  si Bouton-poussoir BP_Haut appuyé alors
    Monter Volet
    Activer LED
```

Fichier MBlock : VR\_N1\_C2.sb2

**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.

# Niveau 1 - D

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

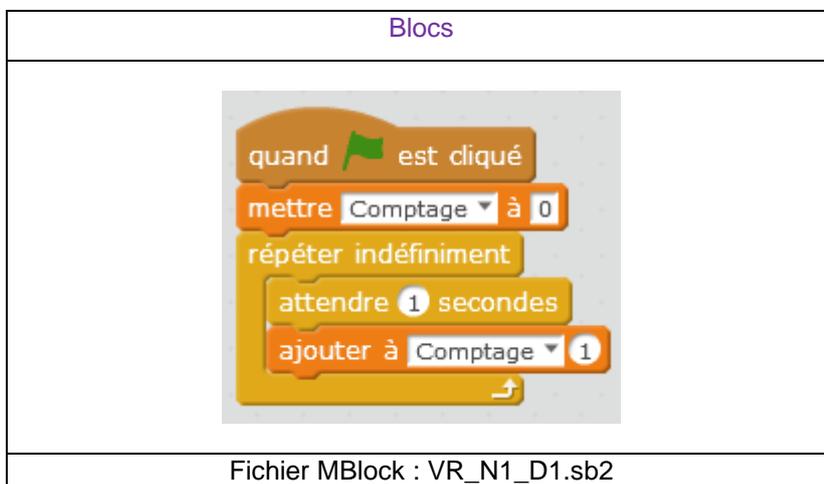
**Objectif** : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débugage).

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

**Instructions utilisées** :



**Correction** :



**Remarques** : 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.2 : Utiliser et tester une variable

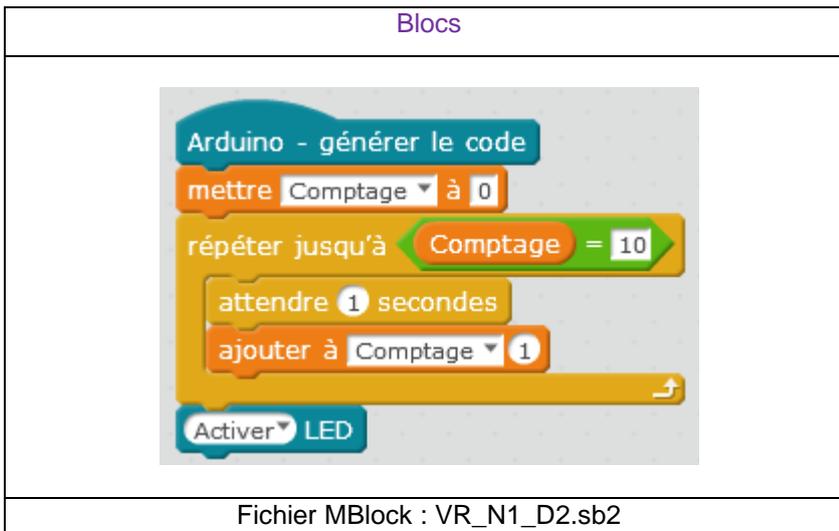
**Objectif** : incrémenter une variable au cours du temps. Lorsque la variable est supérieure à 10, activer le voyant.

**Notion abordée** : boucle tant que dépendant d'une variable

**Instructions utilisées** :



**Correction** :



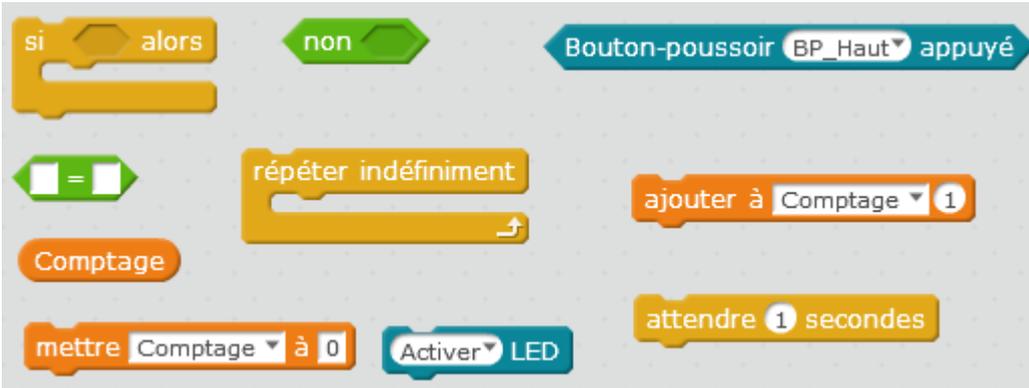
**Remarque** : cet exercice peut être utilisé comme un minuteur.

## Exercice niveau 1 - D.3 : Tests /variables

**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** :



**Correction** :

Blocs

Fichier MBlock : VR\_N1\_D4.sb2

# Programmation version de base niveau 2

## Objectifs :

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

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

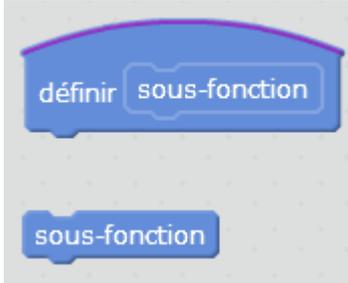
Nom du fichier	Description	Objectif
<b>Niveau 2 A</b>		
VR_N2_A1.sb2	Ouvrir et fermer le volet avec 2 secondes d'attente entre chaque mouvement. Utiliser les capteurs fins de course pour contrôler l'ouverture et la fermeture.	Notions de programmation abordées : -Utilisation des fins de course
VR_N2_A2.sb2	Montée du volet à l'appui sur BP_Haut. Descente du volet à l'appui sur BP_Bas.	
VR_N2_A3.sb2	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.	
VR_N2_A4.sb2	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.

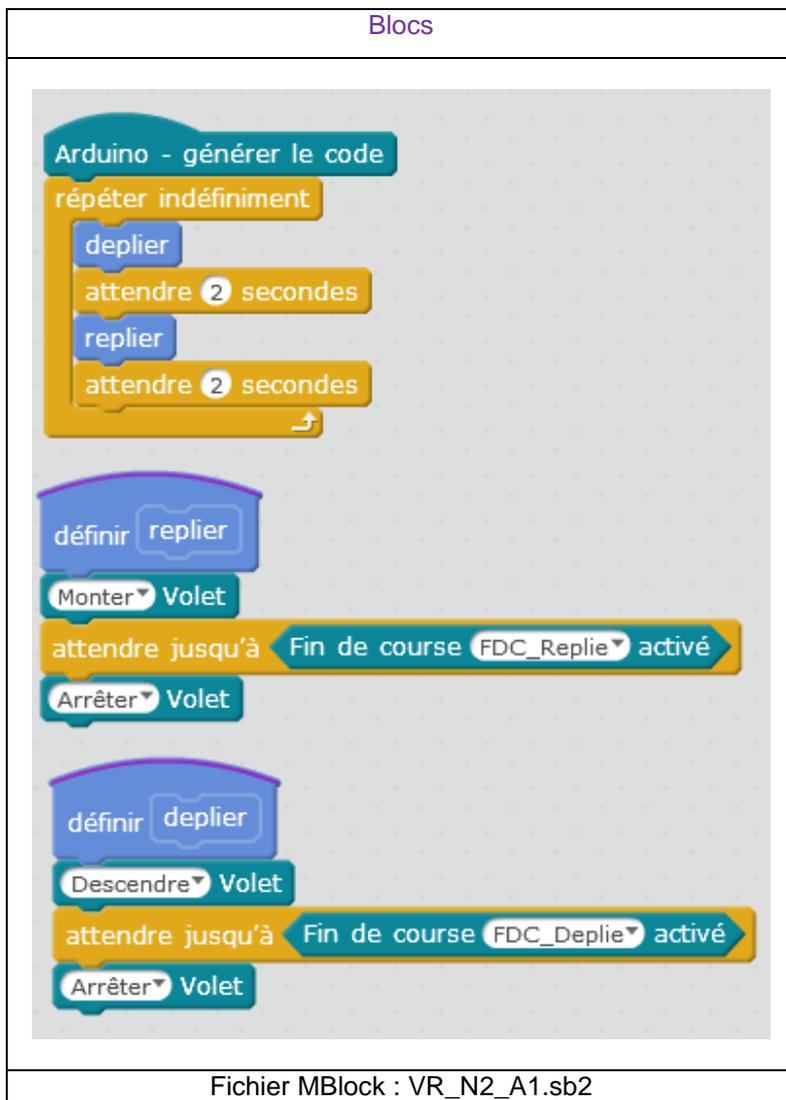
**Notions abordées** : utilisation des fins de course, procédures (sous-fonctions)

**Instructions utilisées** :



**Correction** :

Blocs



```
Arduino - générer le code
répéter indéfiniment
  déplier
  attendre 2 secondes
  replier
  attendre 2 secondes
  ↑

définir replier
  Monter Volet
  attendre jusqu'à Fin de course FDC_Replie activé
  Arrêter Volet

définir déplier
  Descendre Volet
  attendre jusqu'à Fin de course FDC_Deplie activé
  Arrêter Volet
```

Fichier MBlock : VR\_N2\_A1.sb2

**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.

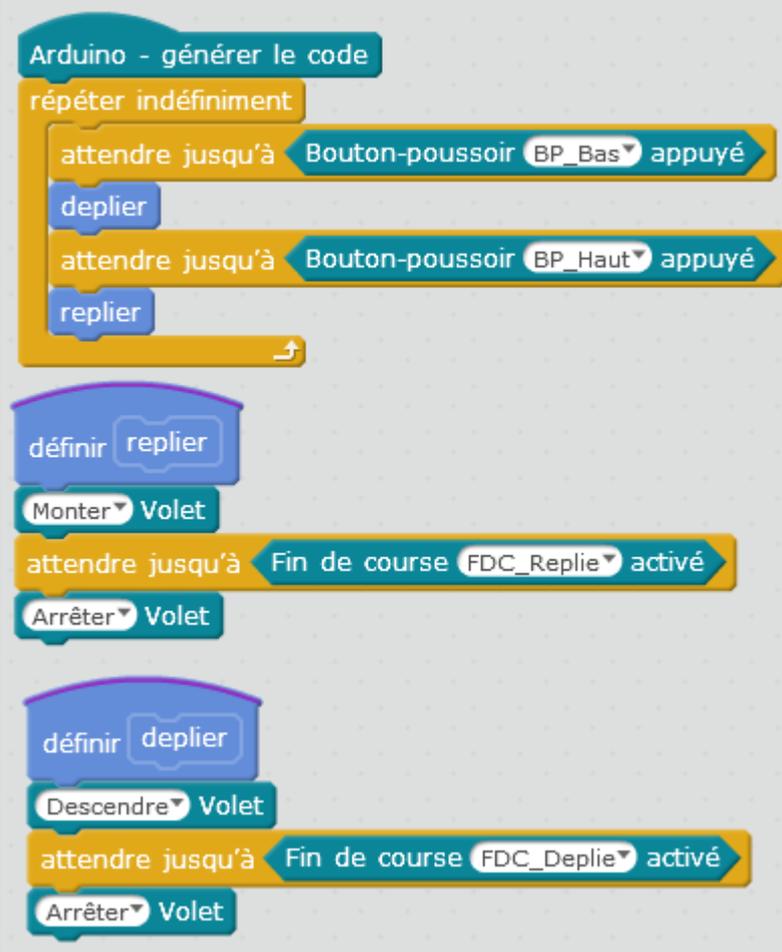
## Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture

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

**Notions abordées :**

**Correction :**

Blocs



The image shows an mBlock code editor window titled 'Blocs'. The code is organized into three main sections. The first section is an 'Arduino - générer le code' block containing an infinite loop. Inside the loop, there are two conditional blocks: one for 'Bouton-poussoir BP\_Bas appuyé' which triggers 'deplier' (unfold) and another for 'Bouton-poussoir BP\_Haut appuyé' which triggers 'replier' (fold). The second section is a 'définir replier' block, which includes 'Monter Volet' (raise blind), 'attendre jusqu'à Fin de course FDC\_Replie activé' (wait for limit switch), and 'Arrêter Volet' (stop blind). The third section is a 'définir déplier' block, which includes 'Descendre Volet' (lower blind), 'attendre jusqu'à Fin de course FDC\_Deplie activé' (wait for limit switch), and 'Arrêter Volet' (stop blind).

Fichier MBlock : VR\_N2\_A2.sb2

## 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

**Instructions utilisées** :



**Correction** :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Haut appuyé ou Bouton-poussoir BP_Bas appuyé alors
    si Fin de course FDC_Replie activé alors
      déplier
    sinon
      replier
  ↵

définir replier
  Monter Volet
  répéter jusqu'à Fin de course FDC_Replie activé
    Activer LED
    attendre 0.2 secondes
    Désactiver LED
    attendre 0.2 secondes
  ↵
  Arrêter Volet

définir déplier
  Descendre Volet
  répéter jusqu'à Fin de course FDC_Deplie activé
    Activer LED
    attendre 0.2 secondes
    Désactiver LED
    attendre 0.2 secondes
  ↵
  Arrêter Volet
```

Fichier MBlock : VR\_N2\_A3.sb2

## 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, 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** :

### Blocs

```
Arduino- générer le code
répéter indéfiniment
  si Bouton-poussoir BP Haut appuyé ou Bouton-poussoir BP Bas appuyé alors
    si Fin de course FDC Replié activé alors
      déplier
    sinon
      replier
  fin

définir replier
  répéter jusqu'à Fin de course FDC Replié activé
    si Bouton-poussoir BP Arrêt appuyé alors
      arrêt
    fin
    Monter Volet
    Activer LED
    attendre 0,2 secondes
    Désactiver LED
    attendre 0,2 secondes
  fin
  Arrêter Volet

définir déplier
  répéter jusqu'à Fin de course FDC Déplié activé
    si Bouton-poussoir BP Arrêt appuyé alors
      arrêt
    fin
    Descendre Volet
    Activer LED
    attendre 0,2 secondes
    Désactiver LED
    attendre 0,2 secondes
  fin
  Arrêter Volet

définir arrêt
  répéter jusqu'à Bouton-poussoir BP Haut appuyé ou Bouton-poussoir BP Bas appuyé
    Arrêter Volet
    Désactiver LED
    si Bouton-poussoir BP Haut appuyé alors
      replier
    fin
    si Bouton-poussoir BP Bas appuyé alors
      déplier
    fin
  fin
```

Fichier MBlock : VR\_N2\_A4.sb2

**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 niveau 3

Objectif : Utiliser les modules plus complexes : pilotage à distance, capteur de lumière...

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
<b>Niveau 3 A : Capteur de lumière</b>		
VR_N3_A1.sb2	Lire une valeur envoyée par un capteur analogique.	Fonctionnalité matérielle abordé : Utilisation d'un capteur analogique
VR_N3_A2.sb2	Gérer l'ouverture et la fermeture en fonction du niveau de luminosité.	
VR_N3_A3.sb2	Création d'un mode automatique et d'un mode manuel.	
VR_N4_A4.sb2	Ajout d'un bouton d'arrêt du programme	
<b>Niveau 3 B : module Bluetooth</b>		
VR_N3_B1.sb2	Contrôler l'ouverture et la fermeture du volet à l'aide de 2 boutons présent sur l'application Android.	Fonctionnalité matérielle abordé : - module Bluetooth  Notions de programmation abordées : - liaison série (hserin/hserout)
VR_N3_B2.sb2	Ouvrir et fermer le volet à partir d'un seul bouton disponible sur l'application Android.	
VR_N3_B3.sb2	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du volet.	
VR_N3_B4.sb2	Gérer la sonnette ainsi que le contrôle du volet à distance à l'aide de l'application Android.	
VR_N3_B5.sb2	Instaurer le mode automatique et le mode manuel pour contrôler le mode à distance.	

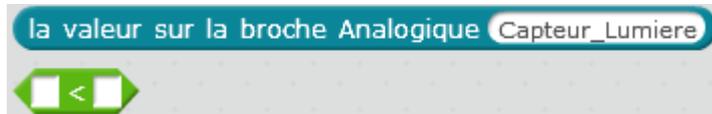
# Option : capteur de lumière

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

**Objectif** : Allumer la LED lorsqu'il y a peu de lumière

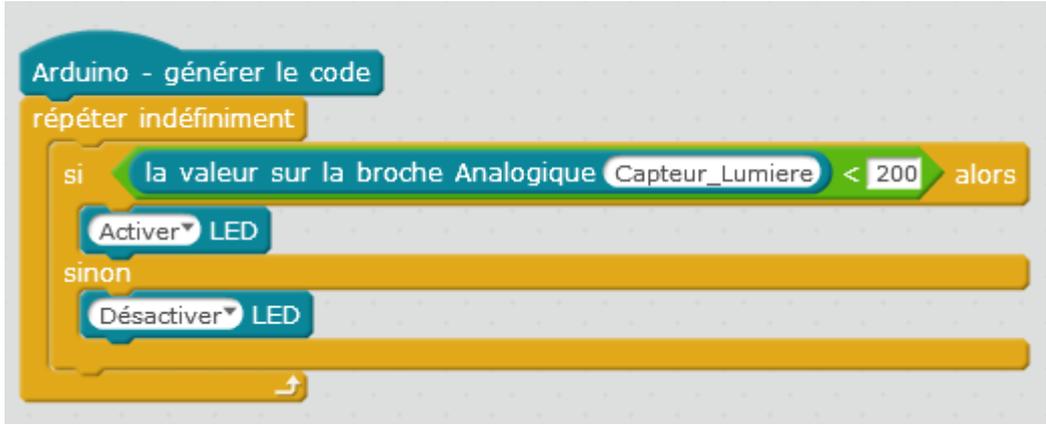
**Notion abordée** : Gestion de variables analogiques

**Instructions utilisées** :



**Correction** :

Blocs



Fichier MBlock : VR\_N3\_A1.sb2

**Remarque** : La valeur de 200 est choisie de telle sorte à ce que le programme marche, vous pouvez utiliser cette valeur. Si la LED ne s'allume pas, augmentez la valeur, si au contraire elle reste allumée même sous la lumière, diminuez cette valeur.

## 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

The screenshot shows the mBlock code for controlling a roller blind. The main code is an 'Arduino - générer le code' block containing an 'infiniment répéter' loop. Inside the loop, there is a 'si' block: 'si la valeur sur la broche Analogique Capteur\_Lumiere < 200 alors'. This 'si' block has two branches: 'si non Fin de course FDC\_Deplier activé alors' followed by a 'deplier' block, and 'sinon' followed by 'si non Fin de course FDC\_Replier activé alors' followed by a 'replier' block. Below the main code, there are two function blocks. The first function is 'définir replier', which contains 'Monter Volet', 'attendre jusqu'à Fin de course FDC\_Replier activé', and 'Arrêter Volet'. The second function is 'définir déplier', which contains 'Descendre Volet', 'attendre jusqu'à Fin de course FDC\_Deplier activé', and 'Arrêter Volet'.

Fichier MBlock : VR\_N3\_A2.sb2

**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

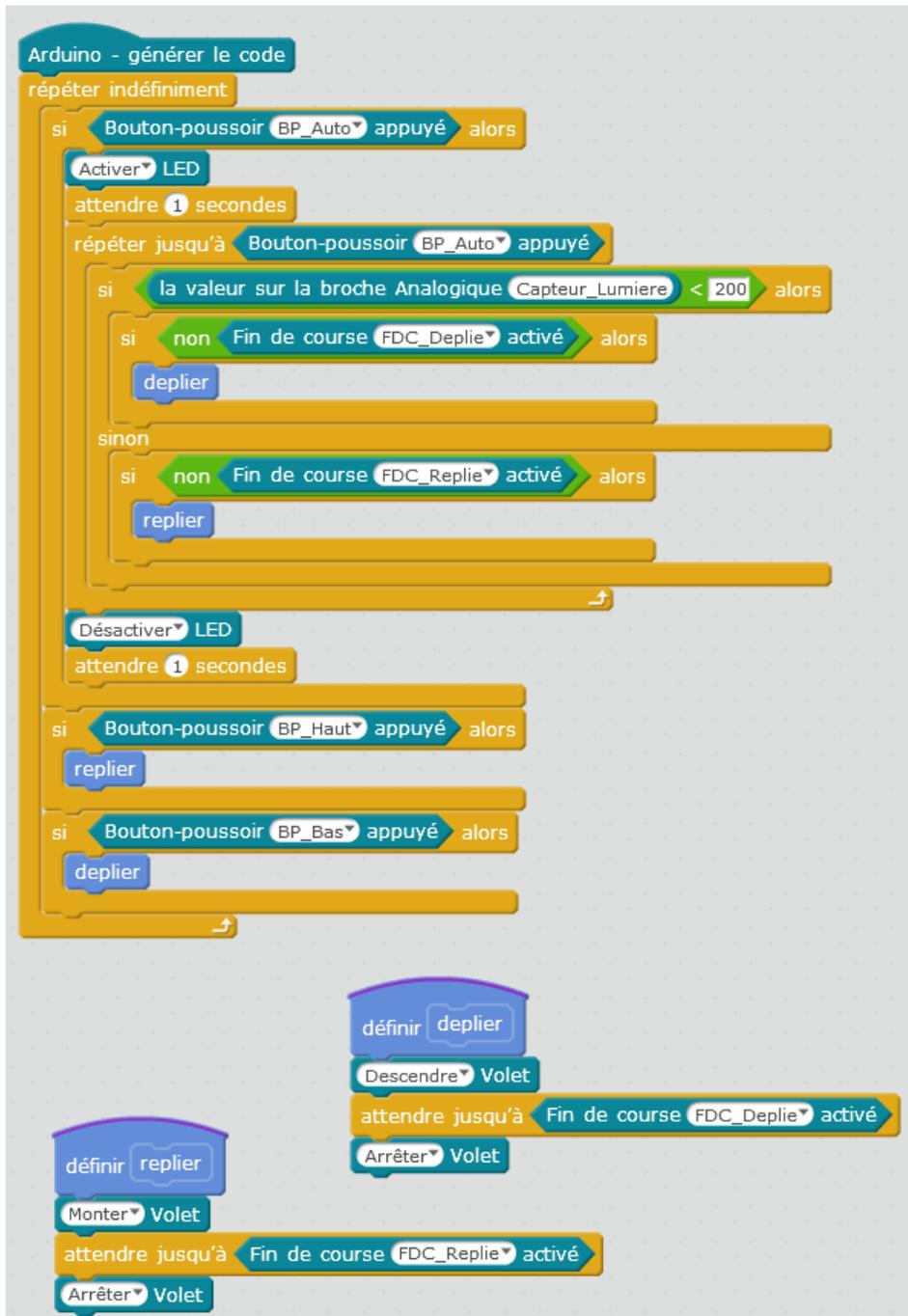
En mode non-automatique : La LED est éteinte, montée ou descente du volet lors de l'appui sur un bouton

Appuyer sur le bouton Auto pour activer ou désactiver le mode automatique

**Notion abordée** : Gestion de variables analogiques, application

**Correction** :

Blocs



Fichier MBlock : VR\_N3\_A3.sb2

**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 soit 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

**Correction :**

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP Auto appuyé alors
    Activer LED
    attendre 1 secondes
    répéter jusqu'à Bouton-poussoir BP Auto appuyé
      si la valeur sur la broche Analogique Capteur Lumier < 200 alors
        si non Fin de course FDC Déplié activé alors
          déplier
        sinon
          si non Fin de course FDC Replié activé alors
            replier
      fin
    Désactiver LED
    attendre 1 secondes
  si Bouton-poussoir BP Haut appuyé alors
    replier
  si Bouton-poussoir BP Bas appuyé alors
    déplier

définir déplier
  Descendre Volet
  attendre jusqu'à Fin de course FDC Déplié activé ou Bouton-poussoir BP Arrêt appuyé
  Arrêter Volet
  attendre jusqu'à Bouton-poussoir BP Haut appuyé ou Bouton-poussoir BP Bas appuyé ou Bouton-poussoir BP Auto appuyé

définir replier
  Monter Volet
  attendre jusqu'à Fin de course FDC Replié activé ou Bouton-poussoir BP Arrêt appuyé
  Arrêter Volet
  attendre jusqu'à Bouton-poussoir BP Haut appuyé ou Bouton-poussoir BP Bas appuyé ou Bouton-poussoir BP Auto appuyé
```

Fichier MBlock : VR\_N3\_A4.sb2

# 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.

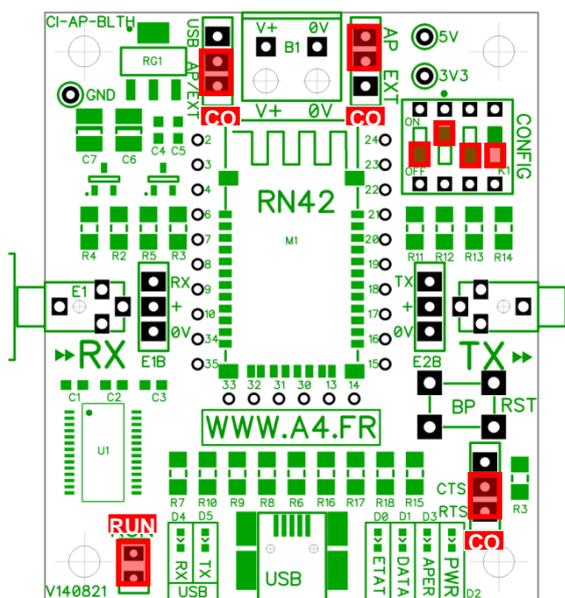
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

Elle est téléchargeable sur [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).

Les informations seront envoyées via un smartphone ou une tablette possédant la technologie Bluetooth à l'aide d'une application développée sous Applinventor 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.

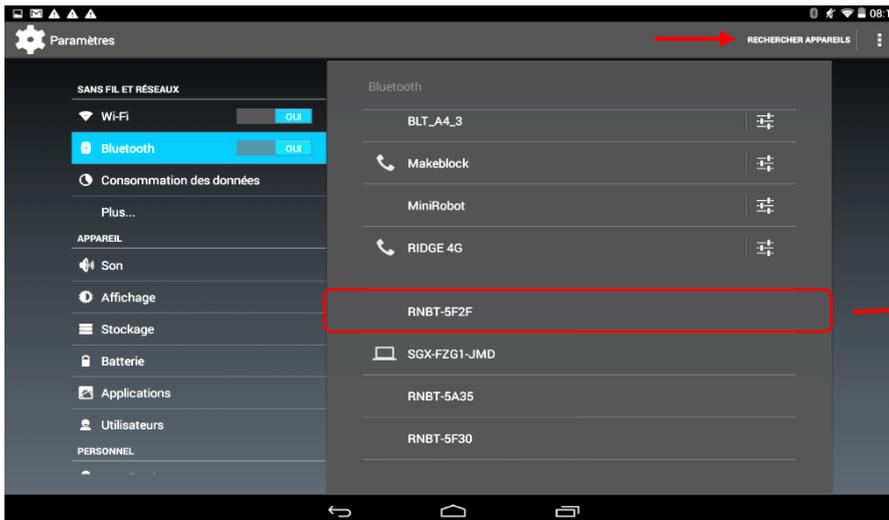
**USB RX** indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

**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 alimenter le module). Le nom de votre module s'appelle : RNBT + les 4 derniers chiffres de l'adresse mac du module notés sur le composant. Sélectionnez 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 Applinventor 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.



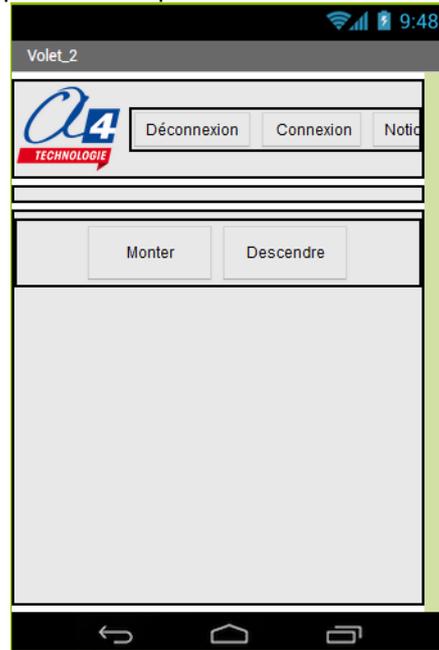
# Exercice niveau 3 - B.1 : Ouvrir/fermer avec application Bluetooth

**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



**Correction :**

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Haut appuyé alors
    replier
  si Bouton-poussoir BP_Bas appuyé alors
    déplier
  si Donnée Bluetooth reçue = 1 alors
    replier
  si Donnée Bluetooth reçue = 2 alors
    déplier

définir replier
  Monter Volet
  attendre jusqu'à Fin de course FDC_Replie activé
  Arrêter Volet

définir déplier
  Descendre Volet
  attendre jusqu'à Fin de course FDC_Deplie activé
  Arrêter Volet
```

Fichier MBlock : VR\_N3\_B1.sb2

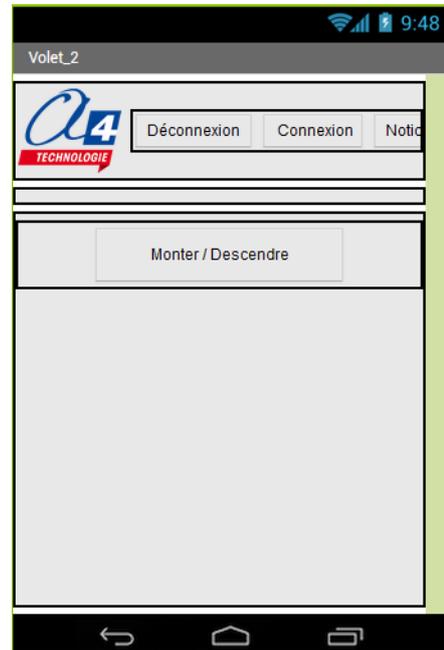
# Exercice niveau 3 - B.2 : Contrôle du volet par Smartphone

**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.

**Application Android :** Volet\_2.apk

**Fichier App Inventor :** Volet\_2.aia



```
quand monter_descendre .Clic
faire appeler Bluetooth .Envoyer1Octet
      nombre 1
```

**Correction :**

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Haut appuyé alors
    replier
  si Bouton-poussoir BP_Bas appuyé alors
    déplier
  si Donnée Bluetooth reçue = 1 alors
    si Fin de course FDC_Deplie activé alors
      replier
    sinon
      déplier
  fin

définir replier
  Monter Volet
  attendre jusqu'à Fin de course FDC_Replie activé
  Arrêter Volet

définir déplier
  Descendre Volet
  attendre jusqu'à Fin de course FDC_Deplie activé
  Arrêter Volet
```

Fichier MBlock : VR\_N3\_B2.sb2

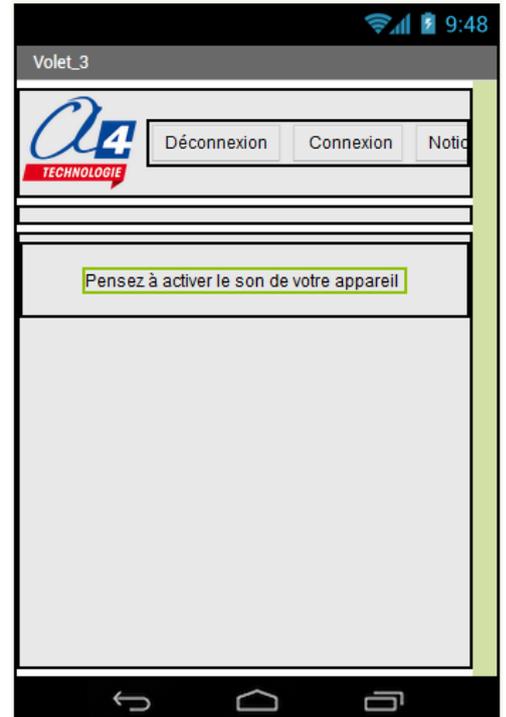
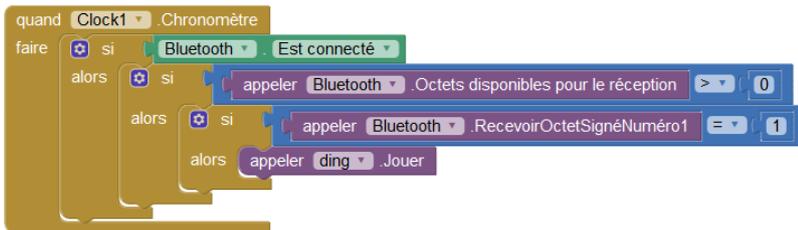
## Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone

**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.

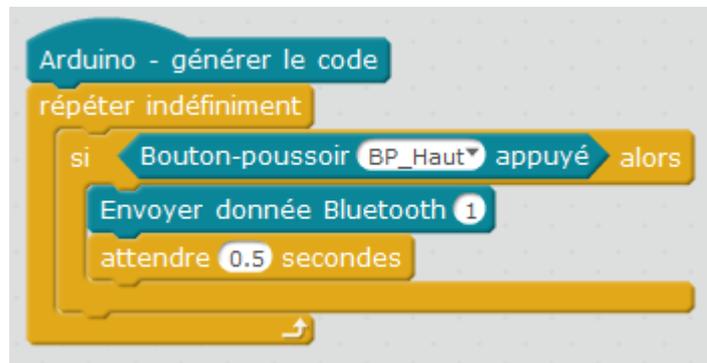
**Application Android** : Volet\_3.apk

**Fichier App Inventor** : Volet\_3.aia



**Correction** :

Blocs



Fichier MBlock : VR\_N3\_B3.sb2

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

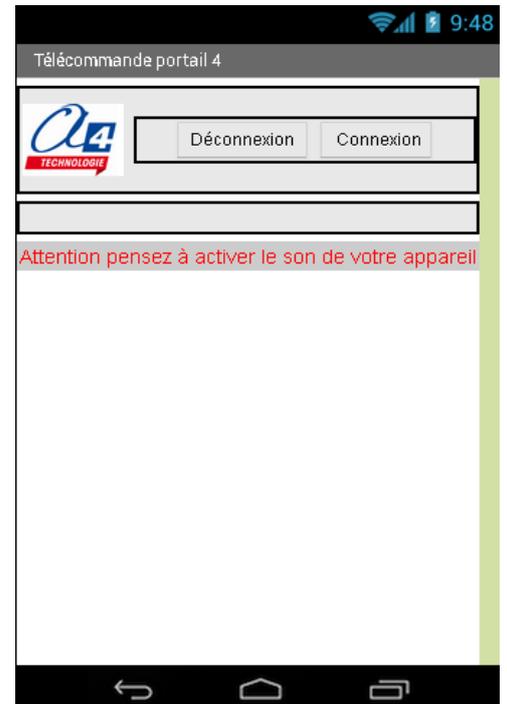
**Objectif :** gérer la sonnette ainsi que le contrôle du volet à distance à l'aide de l'application Android.

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

**Application Android :** Volet\_4.apk

**App Inventor :** Volet\_4.aia

```
quand Clock1 Chronomètre
faire
  si Bluetooth Est connecté
  alors
    appeler Bluetooth .Octets disponibles pour la réception > 0
    appeler Bluetooth .RecevoirOctetSignéNuméro1 = 1
    alors
      ding .Jouer
      Sonnerie .Afficher fenêtre choix
      message "Un bouton a été activé, que voulez-vous faire?"
      Titre "ding"
      Texte bouton 1 "Monter/Descendre"
      Texte bouton 2 "Ne rien faire"
      annulable faux
  quand Sonnerie .Après choix
  Choix
  faire
    si obtenir Choix = "Monter/Descendre"
    alors
      appeler Bluetooth .Envoyer1Octet nombre 1
```



## Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Haut appuyé alors
    Envoyer donnée Bluetooth 1
    attendre 1 secondes
  si Donnée Bluetooth reçue = 1 alors
    si Fin de course FDC_Deplie activé alors
      replier
    sinon
      deplier
  si Fin de course FDC_Deplie activé alors
    deplier
```

Fichier MBlock : VR\_N3\_B4.sb2

# 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

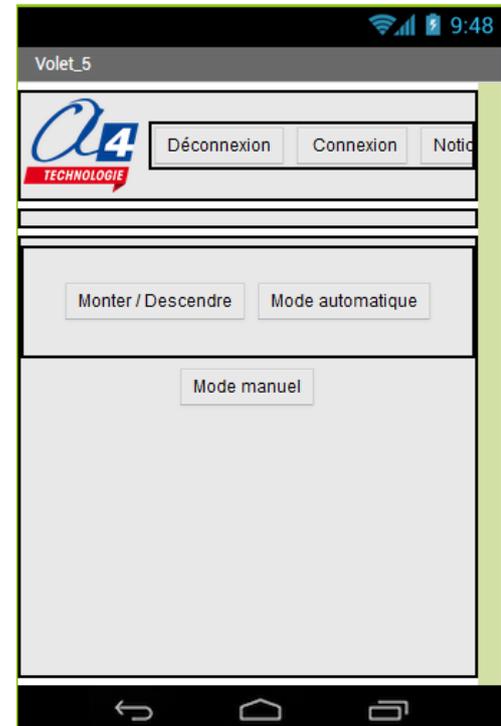
App Inventor : Volet\_5.aia

```
quand Clock1 .Chronomètre
faire
  si Bluetooth .Est connecté
  alors
    si Bluetooth .Octets disponibles pour le réception > 0
    alors
      si Bluetooth .RecevoirOctetSignéNuméro1 = 1
      alors
        appeler ding .Jouer

quand Mode_auto .Clic
faire
  appeler Bluetooth .Envoyer1Octet
  nombre 1

quand Monter_Descendre .Clic
faire
  appeler Bluetooth .Envoyer1Octet
  nombre 2

quand Mode_manuel .Clic
faire
  appeler Bluetooth .Envoyer1Octet
  nombre 3
```



```

Arduino - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    Activer LED
    attendre 1 secondes
    répéter jusqu'à Donnée Bluetooth reçue = 3
      si la valeur sur la broche Analogique Capteur Lumier < 200 alors
        si non Fin de course FDC Déplié activé alors
          déplier
        sinon
          si non Fin de course FDC Replié activé alors
            replier
      ↵
    Désactiver LED
    attendre 1 secondes
  si Donnée Bluetooth reçue = 2 alors
    si Fin de course FDC Déplié activé alors
      replier
    sinon
      déplier
  ↵

définir déplier
  Descendre Volet
  répéter jusqu'à Fin de course FDC Déplié activé
    si Donnée Bluetooth reçue = 1 alors
      Arrêter Volet
    si Bouton-poussoir BP Arrêt appuyé alors
      Arrêter Volet
    si Bouton-poussoir BP Bas appuyé alors
      Descendre Volet
  Arrêter Volet

définir replier
  Monter Volet
  répéter jusqu'à Fin de course FDC Replié activé
    si Bouton-poussoir BP Haut appuyé alors
      Monter Volet
    si Bouton-poussoir BP Arrêt appuyé alors
      Arrêter Volet
    si Donnée Bluetooth reçue = 1 alors
      Monter Volet
  Arrêter Volet
  
```

Fichier MBlock : VR\_N3\_B5.sb2



CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES