

Plateforme élévatrice

Maquette programmable avec mBlock



Arduino - générer le code

répéter indéfiniment

```
si Bouton-poussoir BP_Haut appuyé alors
  Monter Plateforme
  attendre jusqu'à Fin de course Haut activé
  Arrêter Plateforme
si Bouton-poussoir BP_Bas appuyé alors
  Descendre Plateforme
  attendre jusqu'à Fin de course Bas activé
  Arrêter Plateforme
```

Ressources disponibles pour le projet

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

Plateforme élévatrice

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

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

SOMMAIRE

Introduction	2
Plateforme élévatrice	2
Les environnements de programmation graphique	2
Le dossier	2
Les fiches exercices	3
Prérequis	3
Tableau d'affectation des entrées et sorties	4
Programmation version de base niveau 1	5
Niveau 1 - A.....	6
Exercice niveau 1 - A.1 : Maitriser la rotation du moteur.....	6
Exercice niveau 1 - A.2 : Utilisation d'une boucle tant que	7
Exercice niveau 1 - B.2 : Contrôle moteur.....	8
Exercice niveau 1 - B.3 : Contrôle moteur.....	9
Niveau 1 - C.....	10
Exercice niveau 1 - C.1 : Utilisation des variables	10
Exercice niveau 1 - C.2 : Utiliser et tester une variable.....	11
Exercice niveau 1 - C.3 : Tests /variables	12
Programmation version de base niveau 2	13
Exercice niveau 2 - A.1 : ouverture/fermeture entre fins de courses	14
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture.....	15
Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP capteurs d'arrivée	16
Programmation niveau 3.....	17
Option : Module voyant lumineux	18
Exercice niveau 3 - A.1 : Activer / désactiver un témoin lumineux.....	19
Exercice niveau 3 - A.2 : Répéter une séquence indéfiniment.....	20
Exercice niveau 3 - A.3 : Allumer une LED à l'appui d'un bouton poussoir	21
Exercice niveau 3 - A.4 : Allumer une LED lors d'un déplacement de la plate-forme.....	22
Exercice niveau 3 - A.5 : Finalisation de la plate-forme élévatrice.....	23
Option : Module Bluetooth.....	24
Configuration	24
Mise en place des programmes et procédure de connexion.....	25
Exercice niveau 3 - B.1 : Monter/descendre avec application Bluetooth	26
Exercice niveau 3 - B.2 : Contrôle du portail par Smartphone	27
Exercice niveau 3 - B.3 : Envoyer des données vers un Smartphone	28
Exercice niveau 3 - B.4 : Envoyer et recevoir des données provenant d'un Smartphone	29
Exercice niveau 3 - B.5 : Gestion à distance d'une information	31

Introduction

Plateforme élévatrice

La maquette plateforme élévatrice (BE-AHANDI) est une reproduction homothétique d'un système d'une plateforme automatisée réelle : plusieurs étages, capteurs fin de course, moteur, plateforme, etc. Programmable et pilotée par les systèmes AutoProgX2 ou AutoProgUno, elle 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 de la plateforme :

- 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 : voyant lumineux 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_Plateforme** (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 : voyant lumineux 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 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 mBlock (extension .sb2).

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 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 :

- **PE** pour AutoAlarme
- **N** : niveau de programmation 1-2-3
- **A-B-C** : jeu d'instructions du plus simple au plus avancé

Exemple : PE_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_Plateforme** (fichier zip) dans mBlock.
- **Maquette** Plateforme élévatrice (Réf. BE-AALAR).
- **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.

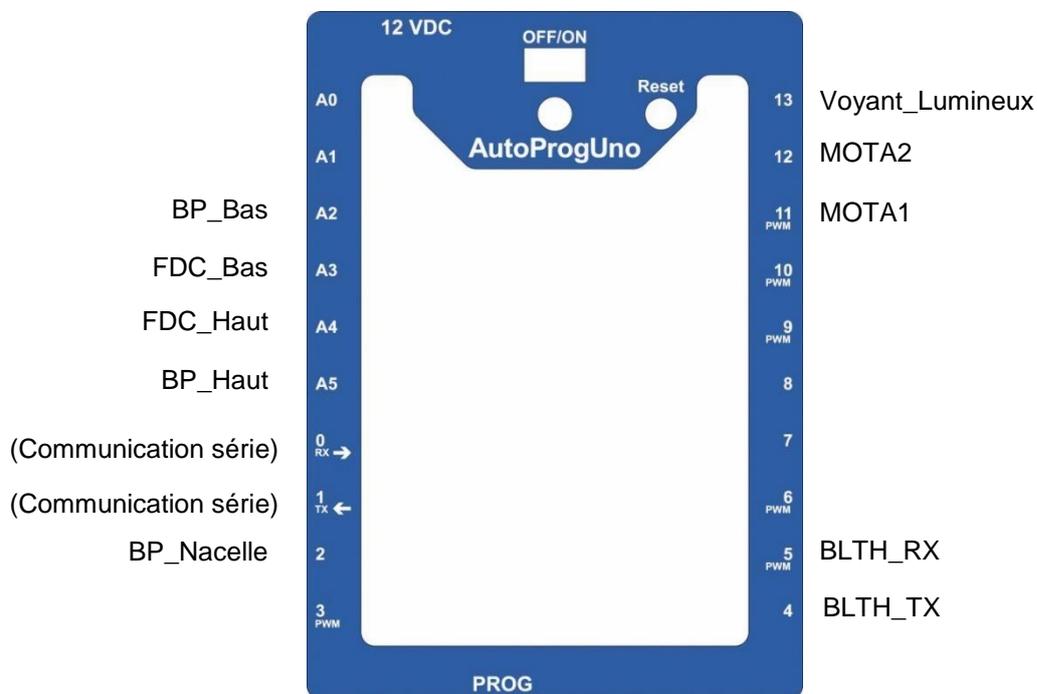
Pour l'option Bluetooth :

- **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	AutoAlarme	Nom mBlock
Modules capteurs pour entrées numériques		
2	Bouton poussoir sur la plateforme	BP_Nacelle
3		
4	Communication Bluetooth (option)	BLTH_TX
5	Communication Bluetooth (option)	BLTH_RX
Modules actionneurs pour sorties numériques		
9		
10		
11	Borne 1 du moteur	MOTA1
12	Borne 2 du moteur	MOTA2
13	Module voyant lumineux (option)	Voyant_Lumineux
Module de communication		
0	(communication avec ordinateur)	
1		
Entrées / sorties libres (A pour les analogiques)		
A0		
A2	Bouton poussoir bas	BP_Bas
A3	Capteur fin de course bas	FDC_Bas
A4	Capteur fin de course haut	FDC_Haut
A5	Bouton poussoir haut	BP_Haut



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
Niveau 1 A		
PE_N1_A1.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
PE_N1_A2.sb2	Ouvrir et fermer la plateforme en continu jusqu'à l'appui d'un bouton-poussoir.	
Niveau 1 B		
PE_N1_B1.sb2	Allumer un moteur à l'appui d'un bouton-poussoir.	Fonctionnalité matérielle abordée : -Utilisation de bouton-poussoir Notions de programmation abordées : -Le test d'une entrée (si/sinon)
PE_N1_B2.sb2	Allumer les moteurs jusqu'à l'arrivée sur un capteur de fin de course puis repartir dans l'autre sens à chaque fin de course.	
PE_N1_B3.sb2	Contrôler le moteur avec les boutons-poussoirs. Monter lors d'un appui sur le bouton du haut, descendre lors d'un appui sur le bouton du bas.	
Niveau 1 C		
PE_N1_C1.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 -Incrémentation de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débugage
PE_N1_C2.sb2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer un moteur.	
PE_N1_C3.sb2	Incrémenter une variable à l'appui d'un bouton-poussoir, la décrémenter à l'appui de l'autre bouton-poussoir.	

Niveau 1 - A

Exercice niveau 1 - A.1 : Maitriser la rotation du moteur.

Objectif : Monter la plateforme pendant 3 secondes puis la descendre 3 secondes et enfin l'arrêter

Notion abordée : utilisation d'un moteur.

Instructions utilisées :



Correction :

Blocs

Fichier MBlock : PE_N1_A1.sb2

ATTENTION : pour cet exercice il est recommandé de placer la plateforme à mi-hauteur pour éviter tout dommage. Il faut également activer le moteur à l'aide de l'interrupteur (Une LED rouge indique si le moteur est allumé).

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

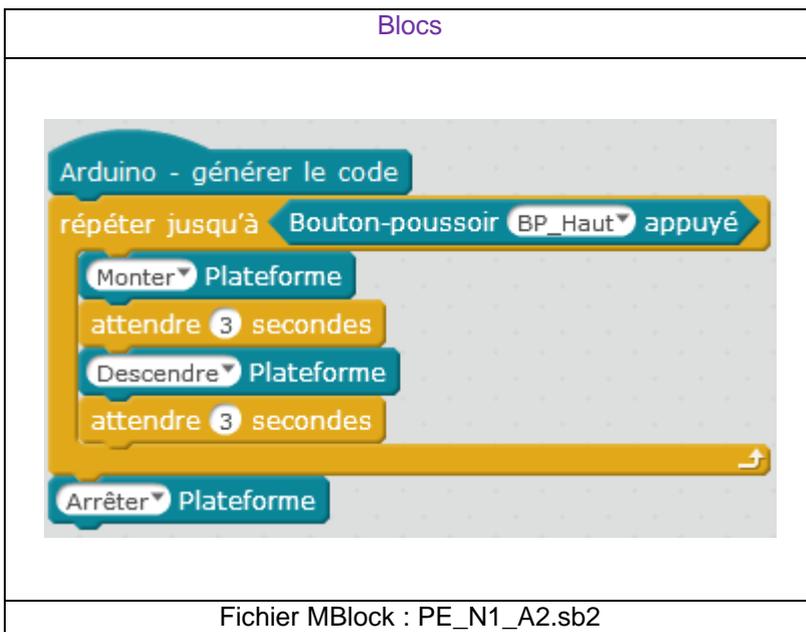
Objectif : monter et descendre la plateforme en continu jusqu'à l'appui d'un bouton-poussoir.

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).

Exercice niveau 1 - B.2 : Contrôle moteur

Objectif : contrôler le moteur avec les boutons poussoirs. Monter lors d'un appui sur le bouton du haut, descendre lors d'un appui sur le bouton du bas.

Notion abordée : utilisation des commandes conditionnelles.

Instructions utilisées :



Correction :

Blocs

Fichier MBlock : PE_N1_B3.sb2

Remarque : Ne pas surcharger le programmes de conditions si, le programme cherchera à vérifier toutes les conditions une à une et une condition pourrait en annuler une autre.

Le programme ne permettra pas deux montées successives.

Exercice niveau 1 - B.3 : Contrôle moteur

Objectif : contrôler le moteur avec les boutons poussoirs. Monter lors d'un appui sur le bouton du haut, descendre lors d'un appui sur le bouton du bas. Lors d'un appui sur le bouton de la nacelle, monter si la plate-forme était en bas et inversement.

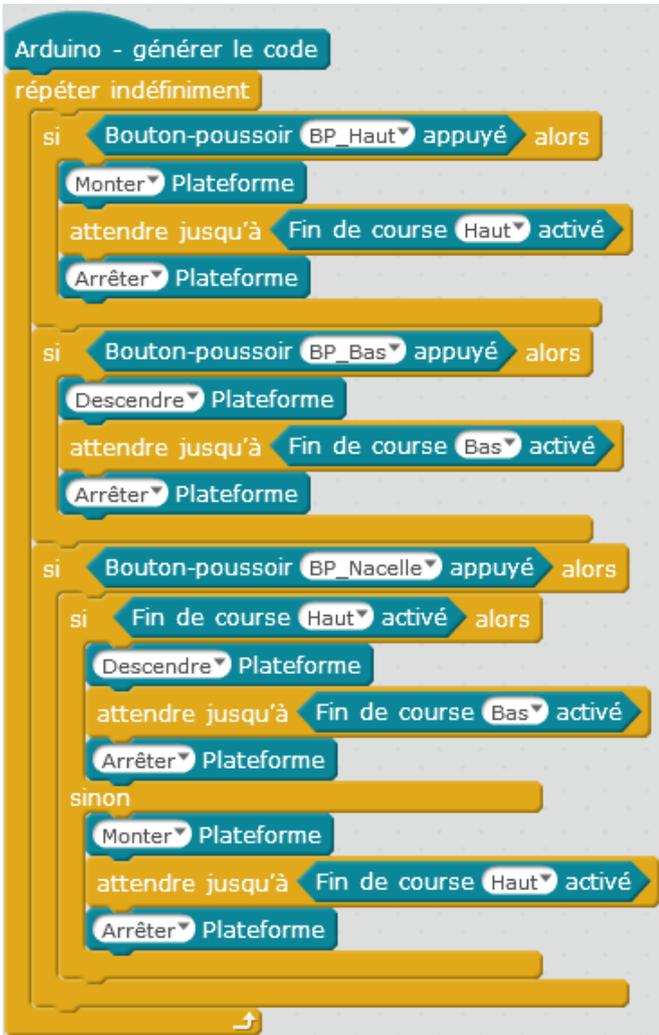
Notion abordée : utilisation des commandes conditionnelles.

Instructions utilisées :



Correction :

Blocs



Fichier MBlock : PE_N1_B4.sb2

Niveau 1 - C

Exercice niveau 1 - C.1 : Utilisation des variables

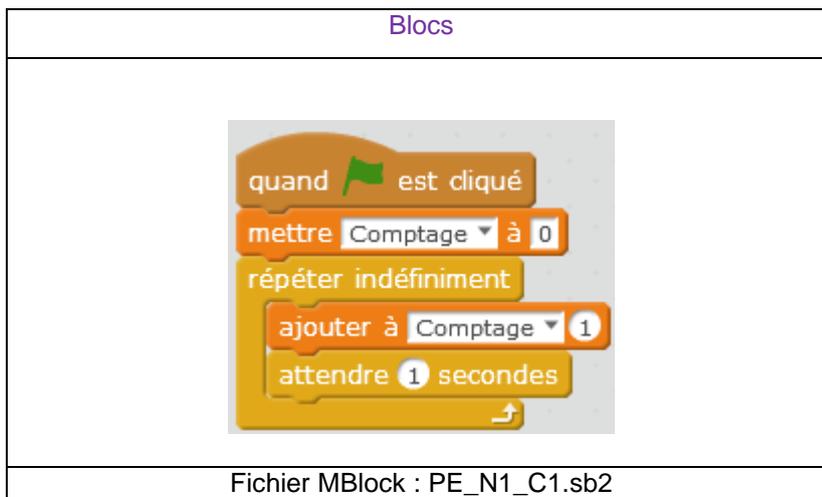
Objectif : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).

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

Instructions utilisées :



Correction :



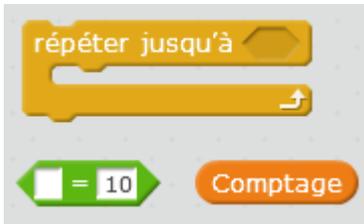
Remarques : la commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

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

Objectif : incrémenter une variable au cours du temps. Lorsque la variable est supérieure à 10, activer les moteurs et les désactiver

Notion abordée : boucle tant que dépendant d'une variable

Instructions utilisées :



Correction :

Blocs

Le code mBlock est le suivant :

- Arduino - générer le code
- mettre Comptage à 0
- répéter jusqu'à Comptage = 10
 - attendre 1 secondes
 - ajouter à Comptage 1
- Monter Plateforme
- attendre 1 secondes
- Descendre Plateforme
- attendre 1 secondes
- Arrêter Plateforme

Fichier MBlock : PE_N1_C2.sb2

Remarque : cet exercice peut être utilisé comme un minuteur.

Exercice niveau 1 - C.3 : Tests /variables

Objectif : incrémenter une variable à chaque appui d'un bouton poussoir. Lorsque le compteur arrive à 10, activer un moteur puis le désactiver après une seconde.

Notion abordée : test dépendant d'une variable

Correction :

Blocs

```
Arduino - générer le code
mettre Comptage à 0
répéter indéfiniment
  si Bouton-poussoir BP_Bas appuyé alors
    ajouter à Comptage 1
    attendre jusqu'à non Bouton-poussoir BP_Bas appuyé
  si Comptage = 10 alors
    Monter Plateforme
    attendre 1 secondes
    Descendre Plateforme
    attendre 1 secondes
    Arrêter Plateforme
    mettre Comptage à 0
```

Fichier MBlock : PE_N1_C4.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.

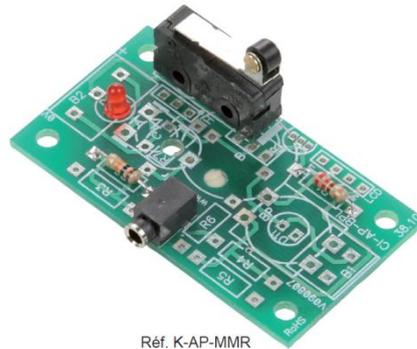
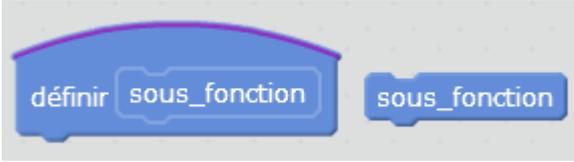
Nom du fichier	Description	Objectif
Niveau 2 A		
PE_N2_A1.sb2	Utilisation des fins de course	Notions de programmation abordées : -Utilisation des sous-fonctions
PE_N2_A2.sb2	Contrôler la montée et la descente avec des boutons-poussoirs.	
PE_N2_A3.sb2	Contrôler la montée et la descente avec des boutons-poussoirs et capteur d'arrivée.	

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

Objectif : Monter et descendre la plateforme 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 :



Réf. K-AP-MMR

Correction :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  monter
  attendre 2 secondes
  descendre
  attendre 2 secondes
  ↻

définir descendre
  Descendre Plateforme
  attendre jusqu'à Fin de course Bas activé
  Arrêter Plateforme

définir monter
  Monter Plateforme
  attendre jusqu'à Fin de course Haut activé
  Arrêter Plateforme
```

Fichier MBlock : PE_N2_A1.sb2

Remarque : l'utilisation des sous-fonctions « monter » et « descendre » facilite la lecture du programme.

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

Objectif : Montée de la plateforme à l'appui du Bouton haut. Descente de la plateforme à l'appui du Bouton bas

Notions abordées : Réutilisation des sous-fonctions pour un autre programme

Correction :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  attendre jusqu'à Bouton-poussoir BP_Bas appuyé
  descendre
  attendre jusqu'à Bouton-poussoir BP_Haut appuyé
  monter

définir monter
  Monter Plateforme
  attendre jusqu'à Fin de course Haut activé
  Arrêter Plateforme

définir descendre
  Descendre Plateforme
  attendre jusqu'à Fin de course Bas activé
  Arrêter Plateforme
```

Fichier MBlock : PE_N2_A2.sb2

Exercice niveau 2 - A.3 : Contrôle ouverture/fermeture avec BP capteurs d'arrivée

Objectif : Faire monter et descendre la plateforme à l'aide des BP sans distinction

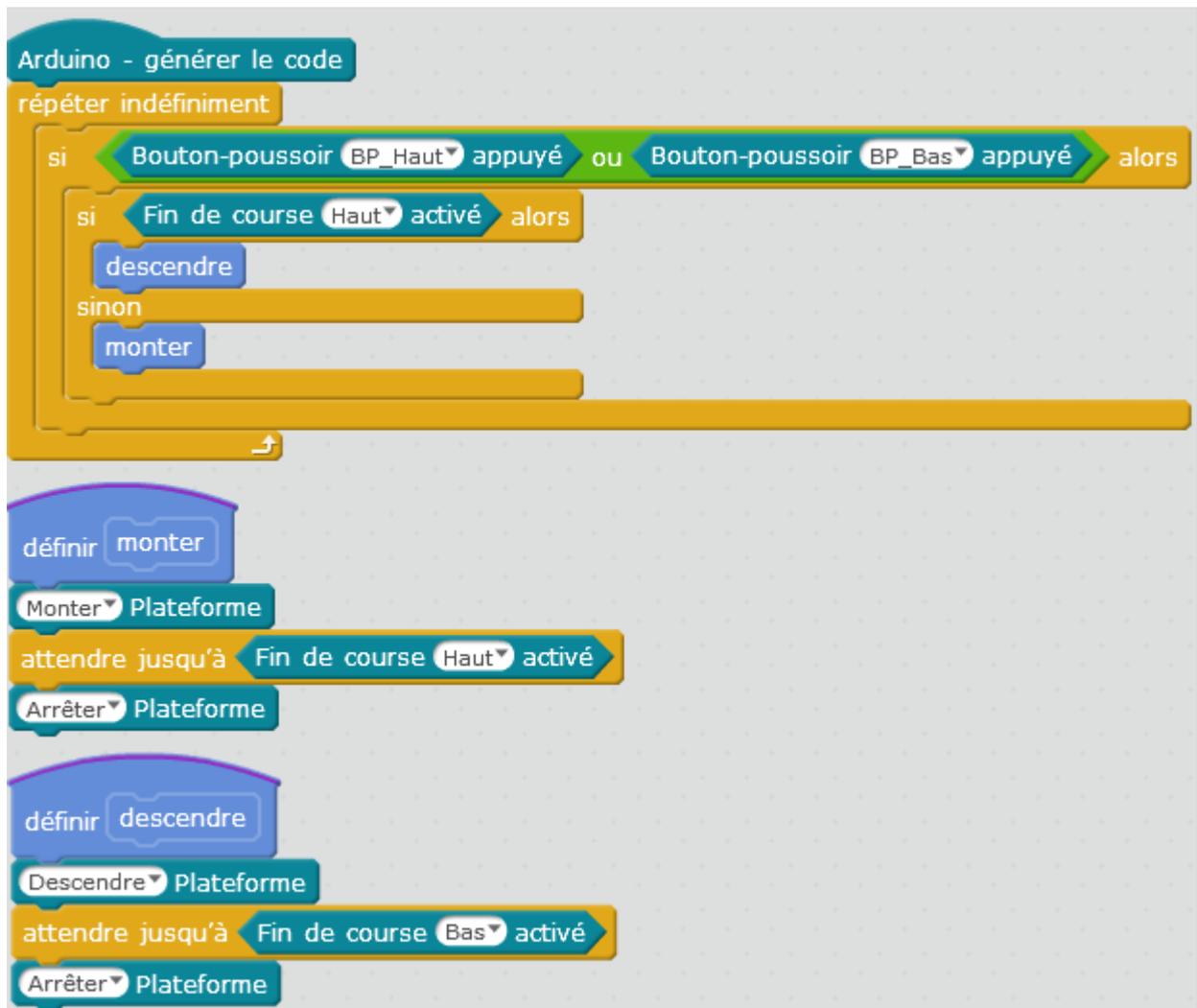
Notions abordées : utilisation d'opérateur logique OU (+)

Instructions utilisées :



Correction :

Blocs



Fichier MBlock : PE_N2_A3.sb2

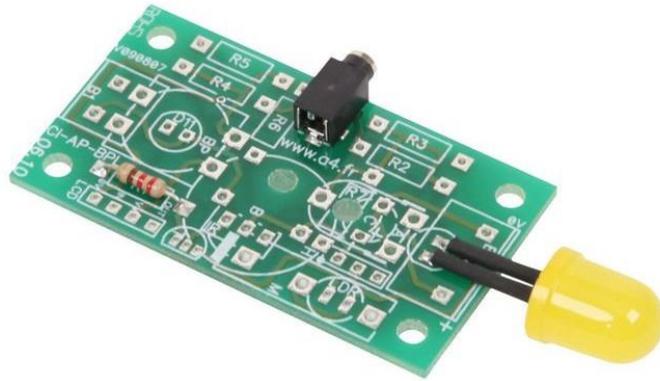
Programmation niveau 3

Objectif : Utiliser les modules plus complexes : pilotage à distance, voyant lumineux...

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 – Module voyant lumineux		
PE_N3_A1.sb2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordé : Utilisation de la télécommande IR Notions de programmation abordées : Utilisation d'un block dédié à la communication IR
PE_N3_A2.sb2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre de façon répétée.	
PE_N3_A3.sb2	Allumer le voyant lumineux à l'aide d'un bouton poussoir et l'éteindre lorsqu'on relâche ce bouton.	
PE_N3_A4.sb2	Reprendre l'exercice PE_N2_A2.sb2 Ajouter un clignotement de la LED pour chaque mouvement de la plate-forme.	
PE_N3_A5.sb2	Reprendre l'exercice précédent. Lors de l'appui du bouton sur la nacelle, la plate-forme descend ou monte en fonction de son étage d'origine.	
Niveau 3 B – Module Bluetooth		
PE_N3_B1.sb2	Contrôler la descente et la montée de la plateforme à l'aide de 2 boutons présent sur l'application Android.	Fonctionnalité matérielle abordée : module Bluetooth Notions de programmation abordées : liaison série (hserin/hserout)
PE_N3_B2.sb2	Monter ou descendre la plateforme à partir d'un seul bouton disponible sur l'application Android. La LED doit être activée lors d'un déplacement.	
PE_N3_B3.sb2	Jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.	
PE_N3_B4.sb2	Faire monter ou descendre la plateforme à partir de boutons sur une application Bluetooth. Jouer une sonnette lorsque la plateforme s'arrête à un étage.	
PE_N3_B5.sb2	Reprendre l'exercice précédent, lorsqu'on appuie sur le bouton poussoir de la balance, envoie une demande de mouvement au smartphone, qui peut décider ou non de mettre en marche la plateforme.	

Option : Module voyant lumineux



Le module voyant lumineux répond à l'activation d'une sortie.

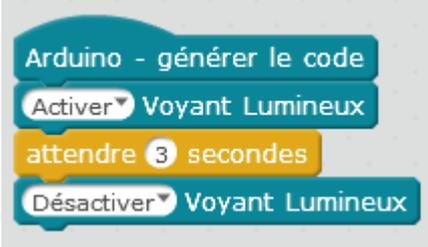
Si une sortie est activée, le voyant lumineux s'allume.

Exercice niveau 3 - 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.

Correction :

Blocs

Fichier MBlock : PE_N3_A1.sb2

Exercice niveau 3 - A.2 : Répéter une séquence indéfiniment

Objectif : allumer le voyant lumineux pendant 3 secondes puis l'éteindre de façon répétée.

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

Correction :

Blocs



```
Arduino - générer le code
répéter indéfiniment
  Activer Voyant Lumineux
  attendre 3 secondes
  Désactiver Voyant Lumineux
  attendre 3 secondes
```

Fichier MBlock : PE_N3_A2.sb2

Exercice niveau 3 - A.3 : Allumer une LED à l'appui d'un bouton poussoir

Objectif : allumer le voyant lumineux à l'aide d'un bouton poussoir et l'éteindre lorsqu'on relâche ce bouton.

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

Correction :

Blocs



```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Bas appuyé alors
    Activer Voyant Lumineux
  sinon
    Désactiver Voyant Lumineux
```

Fichier MBlock : PE_N3_A3.sb2

Exercice niveau 3 - A.4 : Allumer une LED lors d'un déplacement de la plate-forme

Objectif : Reprendre l'exercice PE_N2_A2.sb2 Ajouter un clignotement de la LED pour chaque mouvement de la plate-forme

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

Correction :

Blocs

The screenshot shows an mBlock code editor with the following blocks:

- Arduino - générer le code
- répéter indéfiniment
 - attendre jusqu'à Bouton-poussoir BP_Bas appuyé
 - descendre
 - attendre jusqu'à Bouton-poussoir BP_Haut appuyé
 - monter
- définir monter
 - Monter Plateforme
 - répéter jusqu'à Fin de course Haut activé
 - attendre 0.2 secondes
 - Activer Voyant Lumineux
 - attendre 0.2 secondes
 - Désactiver Voyant Lumineux
 - Arrêter Plateforme
- définir descendre
 - Descendre Plateforme
 - répéter jusqu'à Fin de course Bas activé
 - Activer Voyant Lumineux
 - attendre 0.2 secondes
 - Désactiver Voyant Lumineux
 - attendre 0.2 secondes
 - Arrêter Plateforme

Fichier MBlock : PE_N3_A4.sb2

Exercice niveau 3 - A.5 : Finalisation de la plate-forme élévatrice

Objectif : Reprendre l'exercice précédent. Lors de l'appui du bouton sur la nacelle, la plate-forme descend ou monte en fonction de son étage d'origine.

Correction :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Haut appuyé alors
    si non Fin de course Haut activé alors
      monter
  si Bouton-poussoir BP_Bas appuyé alors
    si non Fin de course Bas activé alors
      descendre
  si Bouton-poussoir BP_Nacelle appuyé alors
    si Fin de course Haut activé alors
      descendre
    sinon
      monter

définir descendre
  Descendre Plateforme
  répéter jusqu'à Fin de course Bas activé
    Activer Voyant Lumineux
    attendre 0.2 secondes
    Désactiver Voyant Lumineux
    attendre 0.2 secondes
  Arrêter Plateforme

définir monter
  Monter Plateforme
  répéter jusqu'à Fin de course Haut activé
    attendre 0.2 secondes
    Activer Voyant Lumineux
    attendre 0.2 secondes
    Désactiver Voyant Lumineux
  Arrêter Plateforme
```

Fichier MBlock : PE_N3_A5.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.

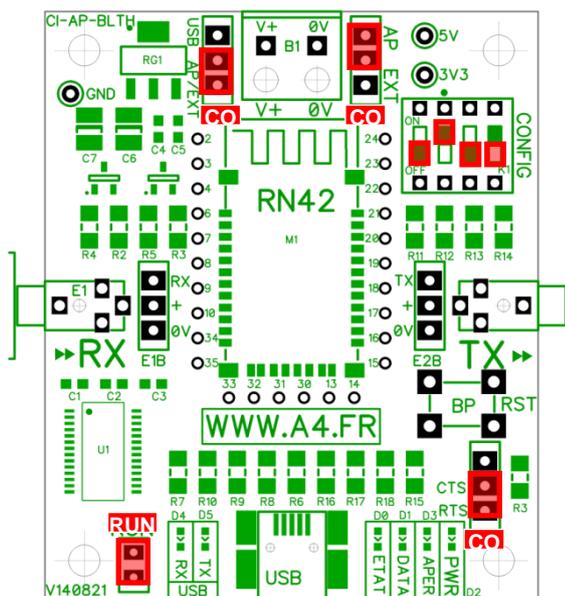
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

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.

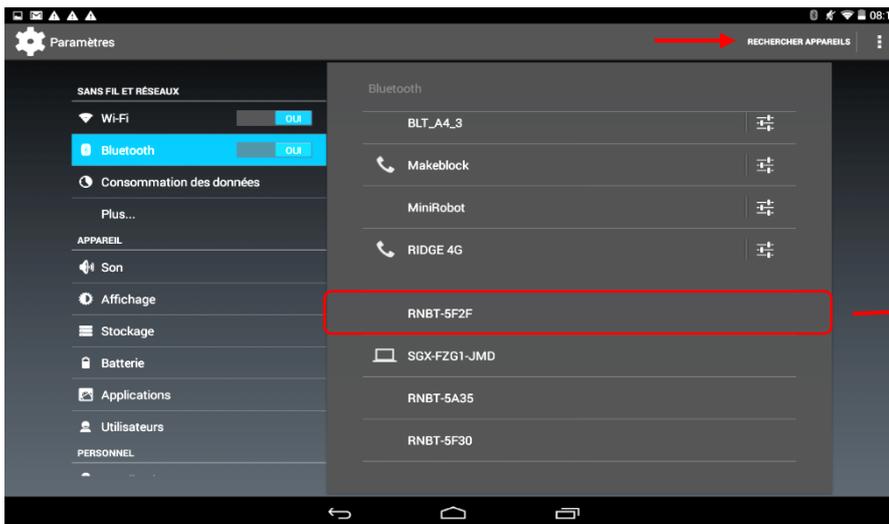
USB RX indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

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 : Monter/descendre avec application Bluetooth

Objectif : Contrôler la descente et la montée de la plateforme à l'aide de 2 boutons présent sur l'application Android.

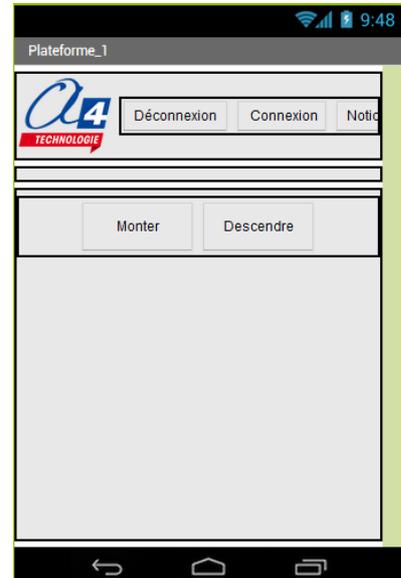
Notion abordée : réception de données Bluetooth envoyées par un Smartphone.

Application Android : Plateforme_1.apk

Fichier App Inventor : Plateforme_1.aia

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

quand descendre .Clic
faire
  appeler Bluetooth .Envoyer1Octet
  nombre 2
```



Correction :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    si non Fin de course Haut activé alors
      monter
  si Donnée Bluetooth reçue = 2 alors
    si non Fin de course Bas activé alors
      descendre

définir monter
  Monter Plateforme
  attendre jusqu'à Fin de course Haut activé
  Arrêter Plateforme

définir descendre
  Descendre Plateforme
  attendre jusqu'à Fin de course Bas activé
  Arrêter Plateforme
```

Fichier MBlock : PE_N3_B1.sb2

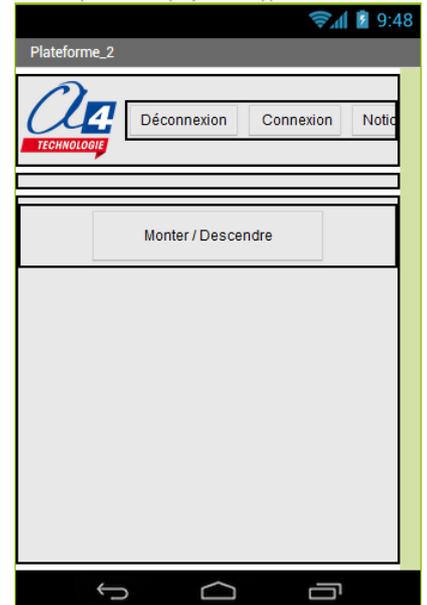
Exercice niveau 3 - B.2 : Contrôle du portail par Smartphone

Objectif : Monter ou descendre la plateforme à partir d'un seul bouton disponible sur l'application Android. La LED doit être activée lors d'un déplacement

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

Application Android : Plateforme_2.apk

Fichier App Inventor : Plateforme_2.aia



Correction :

Blocs

Fichier MBlock : PE_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 portail ou sur un bouton présent sur l'application

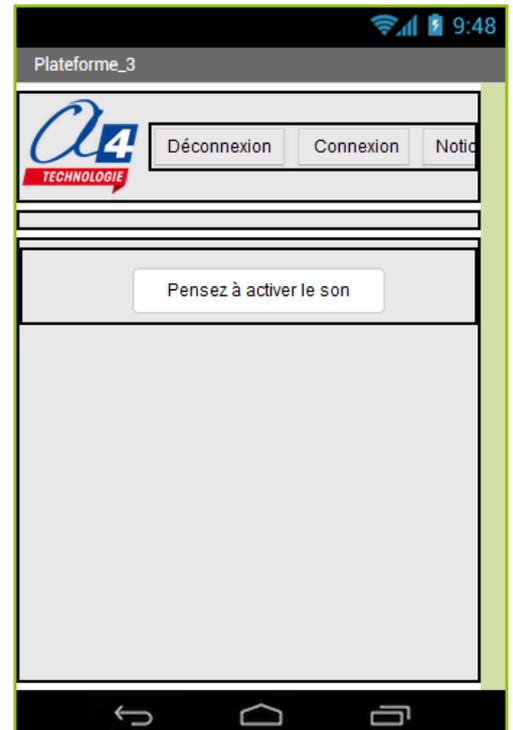
Notion abordée : envoyer des informations à un Smartphone par Bluetooth.

Application Android : Plateforme _3.apk

Fichier App Inventor : Plateforme_3.aia

```
when Sonnette Click
do
  call Bluetooth Send1ByteNumber
  number 1

when Clock1 Timer
do
  if Bluetooth IsConnected
  then
    if Bluetooth BytesAvailableToReceive > 1
    then
      if Bluetooth ReceiveUnsigned1ByteNumber
      then
        call Sonnette Play
```



Correction :

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Bas appuyé alors
    Envoyer donnée Bluetooth 1
    attendre 1 secondes
```

Fichier MBlock : PE_N3_B3.sb2

Remarque : penser à activer le son de votre appareil connecté

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

Objectif : Faire monter ou descendre la plateforme à partir de boutons sur une application Bluetooth. Jouer une sonnette lorsque la plateforme s'arrête à un étage.

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

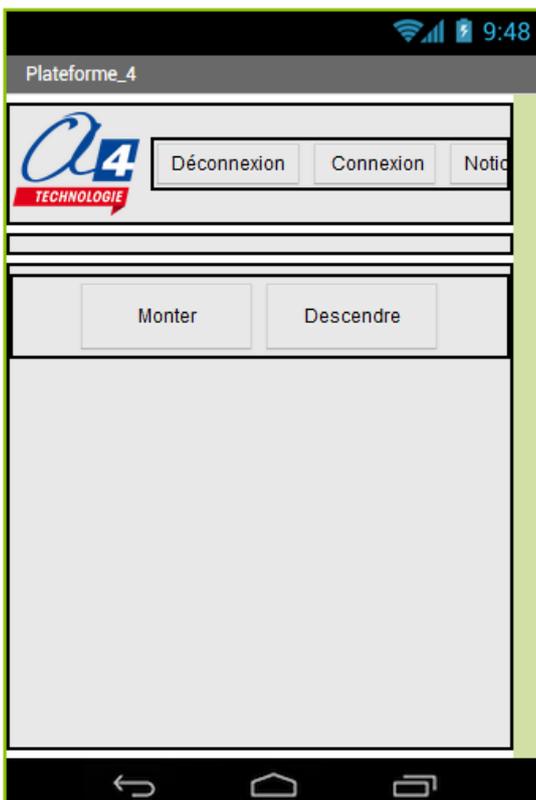
Application Android : Plateforme_4.apk

App Inventor : Plateforme_4.aia

```
quand Clock1 .Chronomètre
faire
  si Bluetooth .Est connecté
  alors
    si appeler Bluetooth .Octets disponibles pour le réception > 1
    alors
      si appeler Bluetooth .RecevoirOctetNonSignéNuméro1 = 1
      alors
        appeler ding .Jouer
```

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

```
quand descendre .Clic
faire
  appeler Bluetooth .Envoyer1Octet
  nombre 2
```



```
Arduino - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    si Fin de course Bas activé alors
      monter
      Envoyer donnée Bluetooth 1
      attendre 0.1 secondes
    si Donnée Bluetooth reçue = 2 alors
      si Fin de course Haut activé alors
        descendre
        Envoyer donnée Bluetooth 1
        attendre 0.1 secondes
```

```
définir monter
  Monter Plateforme
  attendre jusqu'à Fin de course Haut activé
  Arrêter Plateforme
```

```
définir descendre
  Descendre Plateforme
  attendre jusqu'à Fin de course Bas activé
  Arrêter Plateforme
```

Fichier MBlock : PE_N3_B4.sb2

Exercice niveau 3 - B.5 : Gestion à distance d'une information

Objectif : Reprendre l'exercice précédent, lorsqu'on appuie sur le bouton poussoir de la balance, envoie une demande de mouvement au smartphone, qui peut décider ou non de mettre en marche la plateforme.

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

Application Android : Plateforme_5.apk

App Inventor : Plateforme_5.aia

```
quand Clock1 .Chronomètre
faire
  si Bluetooth .Est connecté
  alors
    si appeler Bluetooth .Octets disponibles pour le réception > 0
    alors
      si appeler Bluetooth .RecevoirOctetNonSignéNuméro1 = 1
      alors
        appeler ding .Jouer
        appeler bouton_balance .Afficher fenêtre choix
          message " Quelqu'un souhaite utiliser la plateforme "
          Titre " Notification "
          Texte bouton 1 " Activer la plateforme "
          Texte bouton 2 " Ne rien faire "
          annulable faux

quand bouton_balance .Après choix
Choix
faire
  si obtenir Choix = " Activer la plateforme "
  alors
    appeler Bluetooth .Envoyer1Octet
      nombre 2

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

```
Arduino - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 ou Donnée Bluetooth reçue = 2 alors
    attendre 0.5 secondes
    si Fin de course Bas activé alors
      monter
    sinon
      si Fin de course Haut activé alors
        descendre
  si Bouton-poussoir BP_Nacelle appuyé alors
    Envoyer donnée Bluetooth 1
    attendre 0.5 secondes
```



```
définir monter
  Monter Plateforme
  attendre jusqu'à Fin de course Haut activé
  Arrêter Plateforme

définir descendre
  Descendre Plateforme
  attendre jusqu'à Fin de course Bas activé
  Arrêter Plateforme
```

Fichier MBlock : PE_N3_B5.sb2



CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES