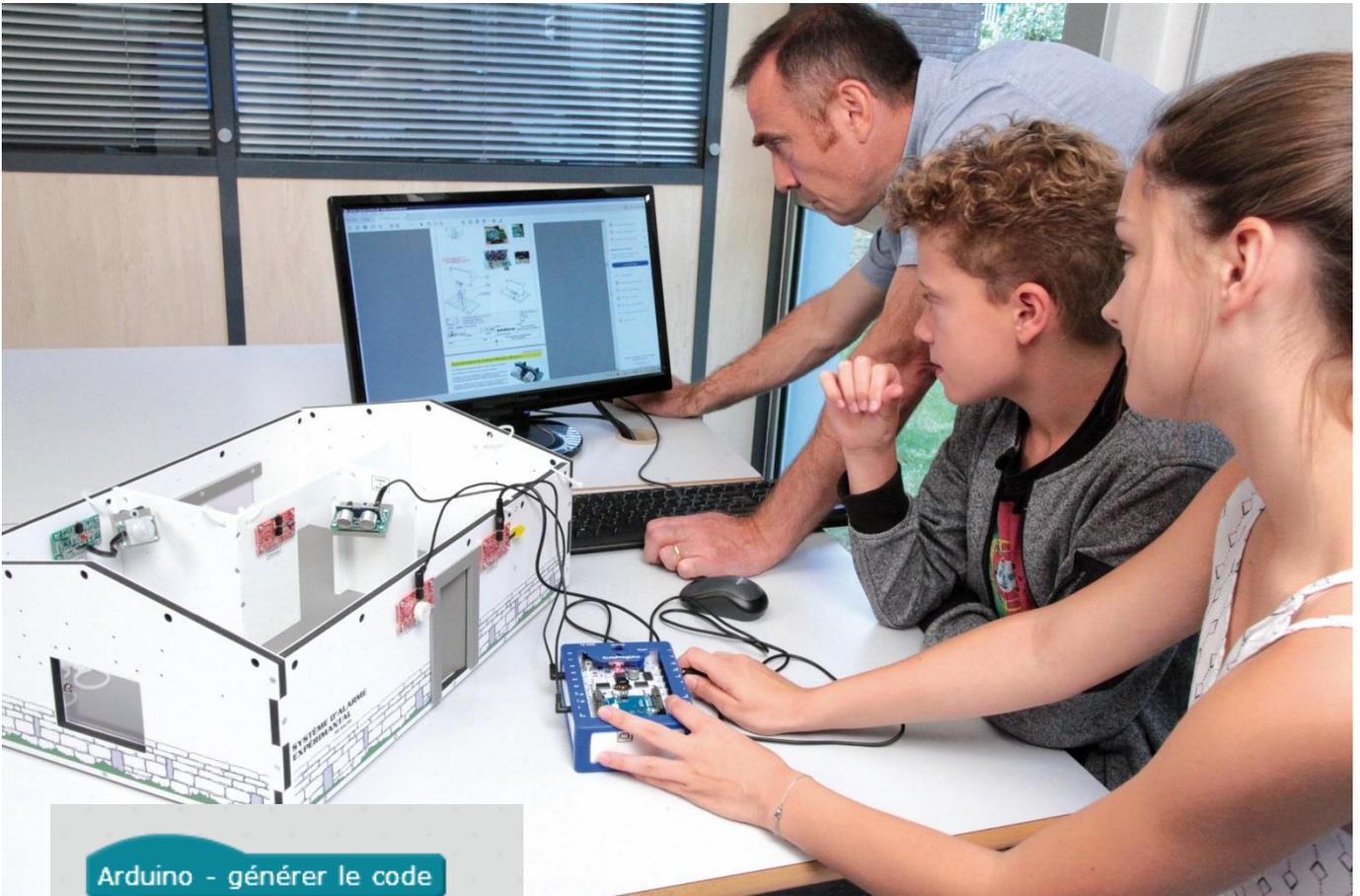


# AutoAlarme

## Maquette programmable avec mBlock



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



# Ressources disponibles pour le projet AutoAlarme

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

## AutoAlarme

- Fichiers **3D** (SolidWorks, Edrawings et Parasolid) de la maquette et de ses options.
- Dossier **technique** AutoAlarme 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>
AutoAlarme .....	2
Les environnements de programmation graphique .....	2
Le dossier .....	2
Les fiches exercices .....	3
Prérequis .....	3
Tableau d'affectation des entrées et sorties .....	4
<b>Programmation version de base niveau 1 .....</b>	<b>5</b>
<b>Niveau 1 - A.....</b>	<b>6</b>
Exercice niveau 1 - A.1 : Activer / désactiver un témoin lumineux.....	6
Exercice niveau 1 - A.2: Répéter une action deux fois.....	7
Exercice niveau 1 - A.3 : Répéter une séquence indéfiniment.....	8
<b>Niveau 1 - B.....</b>	<b>9</b>
Exercice niveau 1 - B.1 : Utiliser un capteur magnétique.....	9
Exercice niveau 1 - B.2 : Utiliser un buzzer .....	10
Exercice niveau 1 - B.3 : Première alarme .....	11
Exercice niveau 1 - B.4 : Activer l'alarme depuis plusieurs endroits .....	12
<b>Niveau 1 - C.....</b>	<b>13</b>
Exercice niveau 1 - C.1 : Utilisation d'un système émetteur/ récepteur infrarouge.....	13
Exercice niveau 1 - C.2 : Utilisation d'un capteur à ultrason .....	14
Exercice niveau 1 - C.3 : Utilisation d'un capteur PIR .....	15
<b>Programmation version de base niveau 2 .....</b>	<b>16</b>
<b>Niveau 2 - A.....</b>	<b>17</b>
Exercice niveau 2 - A.1 : Utilisation des sous-fonctions.....	17
Exercice niveau 2 - A.2 : Personne ne bouge .....	18
<b>Programmation niveau 3 - Option : Module Bluetooth .....</b>	<b>19</b>
Configuration .....	19
Mise en place des programmes et procédure de connexion.....	20
Tableau d'affectation des entrées et sorties.....	21
Exercice niv. 3 - A.1 : Ouvrir/fermer avec application Bluetooth .....	22
Exercice niveau 3 - A.2 : Détecter une intrusion via Bluetooth .....	23
Exercice niveau 3 - A.3 (difficile): Système d'alarme activable .....	25

# Introduction

---

## AutoAlarme

La maquette système d'alarme (BE-AALAR) est une reproduction homothétique d'un système d'alarme automatisé réel : plusieurs étages, capteurs fin de course, contrepoids, moteurs, 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 du système d'alarme :

- 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 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 AutoAlarme 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 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 :

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

Exemple : AL\_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\_Alarme** (fichier zip) dans mBlock.
- **Maquette** AutoAlarme (Réf. BE-AALAR).
- **Câble de programmation** USB (Réf : CABL-IMPUSB).
- **Interface programmable** AutoProgUno (Réf. K-AP-UNO).
- 13 **cordons de liaison** jack compatibles AutoProg pour établir les liaisons entre l'interface programmable et la maquette.

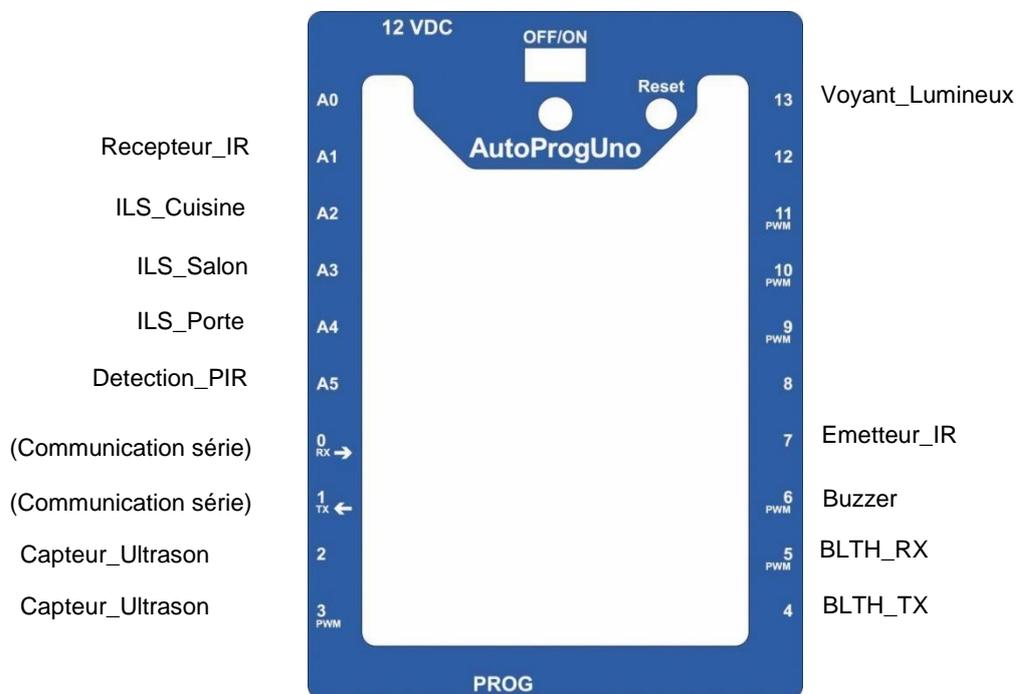
Pour l'option Bluetooth :

- **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
<b>Modules capteurs pour entrées numériques</b>		
2	Capteur à Ultrasons trigger	Capteur_Ultrason
3	Capteur à Ultrasons récepteur	Capteur_Ultrason
4	Module Bluetooth borne TX	BLTH_TX*
5	Module Bluetooth borne RX	BLTH_RX*
6	Module buzzer	Buzzer
7	Emetteur infrarouge	Emetteur_IR
<b>Modules actionneurs pour sorties numériques</b>		
9		
10		
11		
12		
13	Module voyant lumineux jaune	Voyant_Lumineux
<b>Module de communication</b>		
0	(communication avec ordinateur)	
1		
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0		
A1	Récepteur infrarouge	Recepteur_IR
A2	Capteur ILS Cuisine	ILS_Cuisine
A3	Capteur ILS Salon	ILS_Salon
A4	Capteur ILS Porte	ILS_Porte
A5	Module détecteur de présence	Detection_PIR



# 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.

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 « AL\_N1\_A1.sb2 », charger le programme modèle « AL\_BASE.sb2 ».

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 complètement la maquette.

Nom du fichier	Description	Objectif
<b>Niveau 1 A</b> <b>Fichier modèle :</b> AL_N1_A.sb2		
AL_N1_A1	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordée : -Allumage/extinction d'une lumière.
AL_N1_A2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre, recommencer.	Notions de programmation abordées : -séquence d'instructions -temps d'attente -boucle infinie
AL_N1_A3	Faire clignoter le voyant lumineux avec une période de 6 secondes indéfiniment.	
<b>Niveau 1 B</b> <b>Fichier modèle :</b> AL_N1_B.sb2		
AL_N1_B1	Allumer un voyant lumineux lorsque la porte de devant est ouverte.	Fonctionnalité matérielle abordé : -Gestion des capteurs oui ou non -Utilisation d'un buzzer
AL_N1_B2	Activer et désactiver un Buzzer toutes les 3 secondes pendant 1 seconde.	
AL_N1_B3	Créer une alarme qui sonne et allume le voyant lumineux toutes les 0,5 secondes.	Notions de programmation abordées : -boucle qui dépend d'une entrée
AL_N1_B4	Reprendre l'exercice précédent et rajouter les capteurs situés sur les fenêtres.	
<b>Niveau 1 C</b> <b>Fichier modèle :</b> AL_N1_C.sb2		
AL_N1_C1	Allumer le voyant lumineux lorsqu'il y a un obstacle entre le récepteur et l'émetteur infrarouge.	Fonctionnalité matérielle abordé : -Gestion de différents capteurs
AL_N1_C2	Récupérer la valeur de distance envoyée par un capteur à ultrason.	
AL_N1_C3	Allumer le voyant lumineux lorsqu'il y a un obstacle trop proche du capteur à ultrason.	Notions de programmation abordées : -Le test d'une entrée (si/sinon) -Lire une valeur
AL_N1_C4	Allumer le voyant lumineux lorsqu'il y a une détection sur le capteur PIR.	

# Niveau 1 - A

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

Fichier modèle : AL\_BASE.sb2

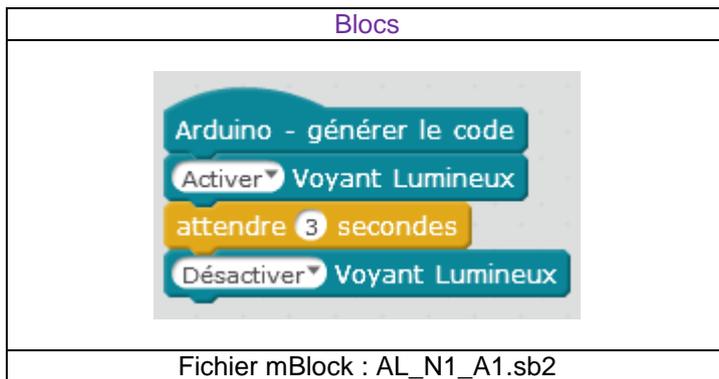
**Objectif** : allumer le voyant lumineux pendant 3 secondes puis l'éteindre.

**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

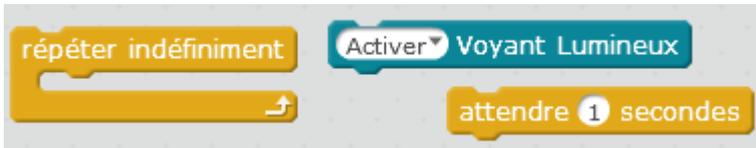
Fichier mBlock : AL_N1_A2.sb2

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

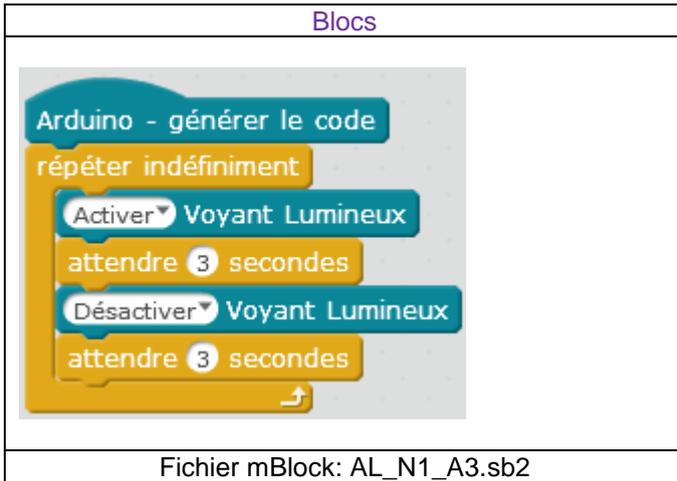
**Objectif** : faire clignoter le voyant lumineux avec une période de 6 secondes indéfiniment.

**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 : Utiliser un capteur magnétique

**Objectif :** Allumer un voyant lumineux lorsque la porte de devant est ouverte

**Notion abordée :** utilisation d'un capteur magnétique

**Instructions utilisées :**



**Correction :**

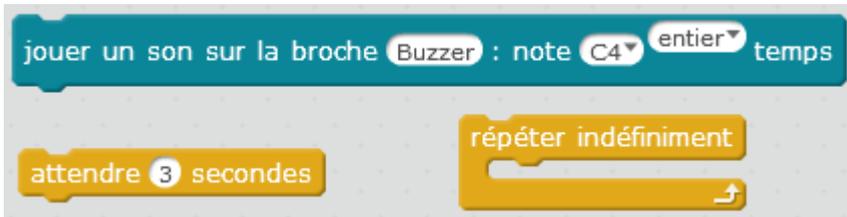
Blocs
<p>Le diagramme de blocs mBlock illustre une logique conditionnelle plus complexe. Il commence par un bloc 'Arduino - générer le code', suivi d'un bloc 'répéter indéfiniment'. À l'intérieur de ce bloc, il y a un bloc 'si...alors...sinon'. Dans la section 'alors', il y a un bloc 'Désactiver Voyant Lumineux'. Dans la section 'sinon', il y a un bloc 'Activer Voyant Lumineux'. Le bloc 'si...alors...sinon' est conditionné par le bloc 'Bouton-poussoir ILS_Porte appuyé'.</p>
Fichier mBlock : AL_N1_B1.sb2

## Exercice niveau 1 - B.2 : Utiliser un buzzer

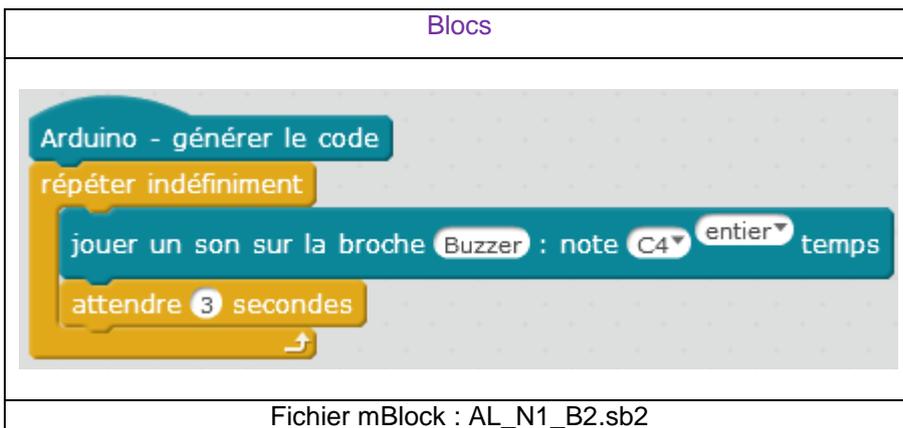
**Objectif :** Activer et désactiver un Buzzer toutes les 3 secondes pendant 1 seconde

**Notion abordée :** utilisation du buzzer

**Instructions utilisées :**



**Correction :**



## Exercice niveau 1 - B.3 : Première alarme

**Objectif :** Créer une alarme qui sonne et allume le voyant lumineux toutes les 0.5 secondes lorsque la porte est ouverte

**Correction :**

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir ILS_Porte appuyé alors
    Désactiver Voyant Lumineux
  sinon
    Activer Voyant Lumineux
    jouer un son sur la broche Buzzer : note C4 Un demi temps
    Désactiver Voyant Lumineux
    attendre 0.5 secondes
  fin
```

Fichier mBlock : AL\_N1\_B3.sb2

# Exercice niveau 1 - B.4 : Activer l'alarme depuis plusieurs endroits

**Objectif :** Reprendre l'exercice précédent et rajouter les capteurs situés sur les fenêtres

**Instructions utilisées :**

**Correction :**

Blocs

```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir ILS_Porte appuyé alors
    Désactiver Voyant Lumineux
  sinon
    Activer Voyant Lumineux
    jouer un son sur la broche Buzzer : note C4 Un demi temps
    Désactiver Voyant Lumineux
    attendre 0.5 secondes
  si Bouton-poussoir ILS_Cuisine appuyé alors
    Désactiver Voyant Lumineux
  sinon
    Activer Voyant Lumineux
    jouer un son sur la broche Buzzer : note C4 Un demi temps
    Désactiver Voyant Lumineux
    attendre 0.5 secondes
  si Bouton-poussoir ILS_Salon appuyé alors
    Désactiver Voyant Lumineux
  sinon
    Activer Voyant Lumineux
    jouer un son sur la broche Buzzer : note C4 Un demi temps
    Désactiver Voyant Lumineux
    attendre 0.5 secondes
```

Fichier mBlock : AL\_N1\_B4.sb2

**Remarque :** Vous pouvez utiliser plusieurs boucles Si, nous verrons par la suite qu'il existe un moyen plus simple pour ne pas surcharger visuellement le programme.

# Niveau 1 - C

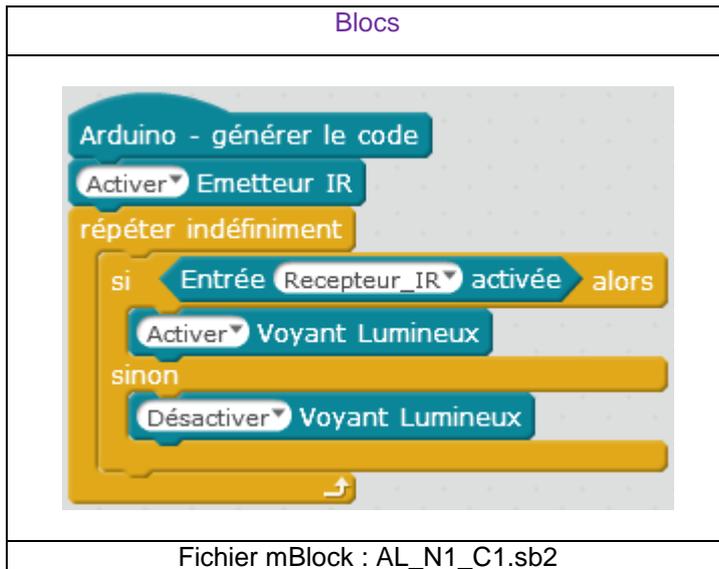
## Exercice niveau 1 - C.1 : Utilisation d'un système émetteur/récepteur infrarouge

**Objectif :** Allumer le voyant lumineux lorsqu'il y a un obstacle entre le récepteur et l'émetteur infrarouge

**Instructions utilisées :**



**Correction :**



**Remarque :** Le récepteur est activé lorsqu'il n'y a rien, c'est à dire qu'il est désactivé lorsqu'il est en face d'un émetteur.

## Exercice niveau 1 - C.2 : Utilisation d'un capteur à ultrason

**Objectif :** Allumer le voyant lumineux lorsqu'il y a un obstacle trop proche du capteur à ultrason (5cm)

**Instructions utilisées :**



**Correction :**

Blocs

```
Arduino - générer le code
répéter indéfiniment
  attendre 0.1 secondes
  si distance mesurée par ultrason : broche TRIG 2 , broche ECHO 3 < 5 alors
    Activer Voyant Lumineux
  sinon
    Désactiver Voyant Lumineux
```

Fichier mBlock : AL\_N1\_C2.sb2

**Remarque :** Mettre un temps de 0.1 seconde entre chaque mesure

## Exercice niveau 1 – C.3 : Utilisation d'un capteur PIR

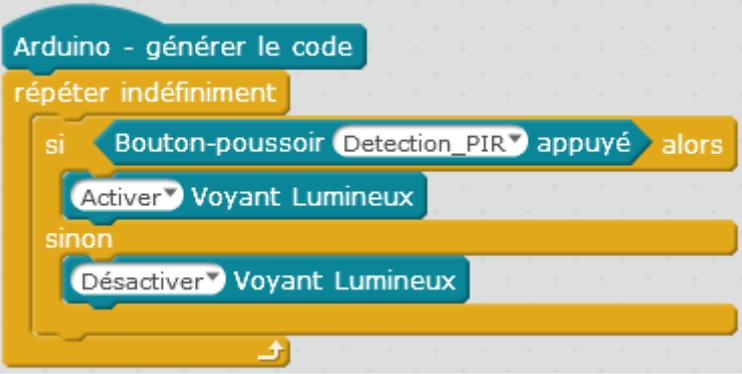
**Objectif :** Allumer le voyant lumineux lorsqu'il y a une détection sur le capteur PIR

**Instructions utilisées :**

Bouton-poussoir Detection\_PIR appuyé

**Correction :**

Blocs



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

Fichier mBlock : AL\_N1\_C3.sb2

**Remarque :** Le récepteur est activé lorsqu'il n'y a rien, c'est à dire qu'il est désactivé lorsqu'il est en face d'un émetteur.

Le module PIR est équipé d'un capteur pyroélectrique. Il réagit aux faibles variations de température et permet de détecter la présence (mouvement) d'une personne jusqu'à 5 m. Son champ de détection est de 60° jusqu'à 2,5 m et 20° à 5 m.

Le capteur réagit comme un bouton poussoir actif lors d'une détection d'un mouvement. Son activation est retardée d'environ 20 secondes après la mise sous tension afin d'éviter les détections intempestives.

Par ailleurs, le capteur est sensible aux variations de températures brutales, aux vibrations ou aux chocs importants. Il ne faut pas l'exposer à la lumière directe du soleil, à l'air pulsé d'un radiateur ou d'un climatiseur. Il est conçu pour une utilisation en intérieur ; pour une utilisation en extérieur, une protection anti humidité est nécessaire.



# Programmation version de base niveau 2

---

## Objectifs :

- Utilisation concrète d'AutoAlarme
- 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 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> <b>Fichier modèle</b> : AL_N2_A.sb2		
AL_N2_A1	Utilisation des sous-fonctions.	Notions de programmation abordées : -Utilisation des sous-fonctions
AL_N2_A2	Personne ne bouge.	

# Niveau 2 - A

## Exercice niveau 2 - A.1 : Utilisation des sous-fonctions

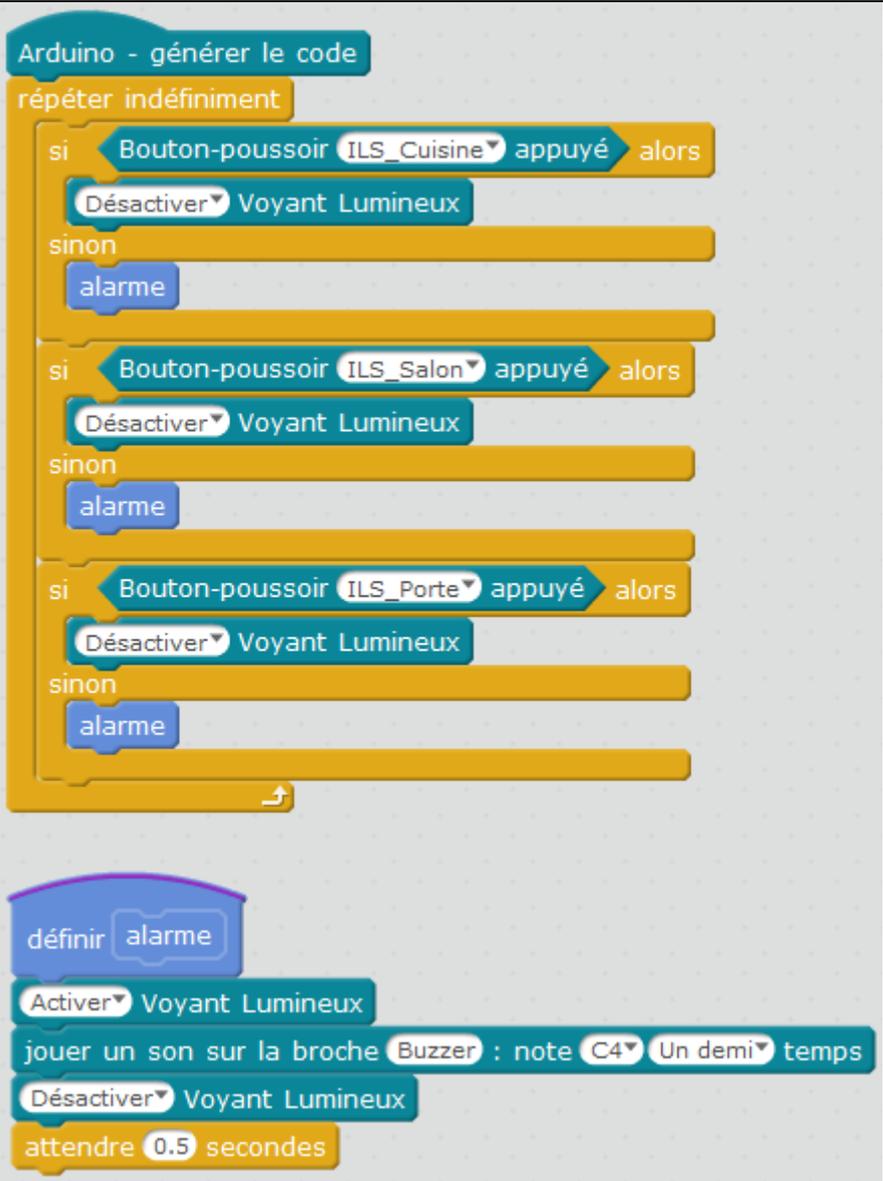
**Objectif** : reprendre l'exercice AL\_N1\_B4.sb2, remettre le programme d'alarme dans une sous-fonction qu'on devra appeler.

**Instructions utilisées** :



**Correction** :

Blocs



```
Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir ILS_Cuisine appuyé alors
    Désactiver Voyant Lumineux
  sinon
    alarme
  si Bouton-poussoir ILS_Salon appuyé alors
    Désactiver Voyant Lumineux
  sinon
    alarme
  si Bouton-poussoir ILS_Porte appuyé alors
    Désactiver Voyant Lumineux
  sinon
    alarme
définir alarme
  Activer Voyant Lumineux
  jouer un son sur la broche Buzzer : note C4 Un demi temps
  Désactiver Voyant Lumineux
  attendre 0.5 secondes
```

Fichier mBlock : AL\_N2\_A1.sb2

**Remarque** : On peut appeler des sous-fonctions dans une sous-fonction.

## Exercice niveau 2 - A.2 : Personne ne bouge

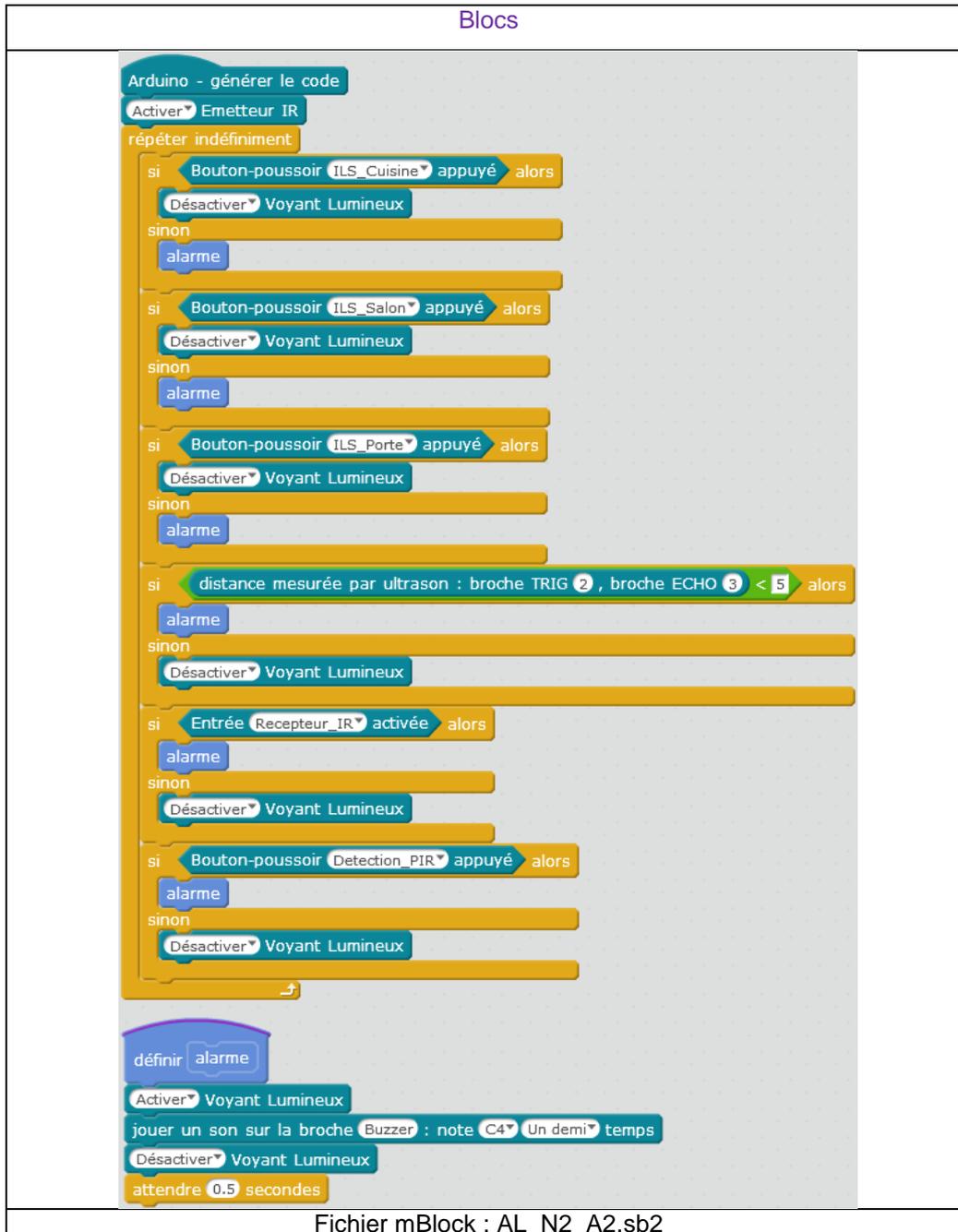
**Objectif :** Créer un programme qui active l'alarme lorsqu'il y a intrusion par une porte ou une fenêtre, ou lorsqu'un des capteurs est activé.

**Instructions utilisées :**



**Correction :**

Blocs



```
Arduino - générer le code
Activer Emetteur IR
répéter indéfiniment
  si Bouton-poussoir ILS_Cuisine appuyé alors
    Désactiver Voyant Lumineux
  sinon
    alarme
  si Bouton-poussoir ILS_Salon appuyé alors
    Désactiver Voyant Lumineux
  sinon
    alarme
  si Bouton-poussoir ILS_Porte appuyé alors
    Désactiver Voyant Lumineux
  sinon
    alarme
  si distance mesurée par ultrason : broche TRIG 2 , broche ECHO 3 < 5 alors
    alarme
  sinon
    Désactiver Voyant Lumineux
  si Entrée Recepteur_IR activée alors
    alarme
  sinon
    Désactiver Voyant Lumineux
  si Bouton-poussoir Detection_PIR appuyé alors
    alarme
  sinon
    Désactiver Voyant Lumineux
définir alarme
Activer Voyant Lumineux
jouer un son sur la broche Buzzer : note C4 Un demi temps
Désactiver Voyant Lumineux
attendre 0.5 secondes
```

Fichier mBlock : AL\_N2\_A2.sb2

# Programmation niveau 3 - 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.

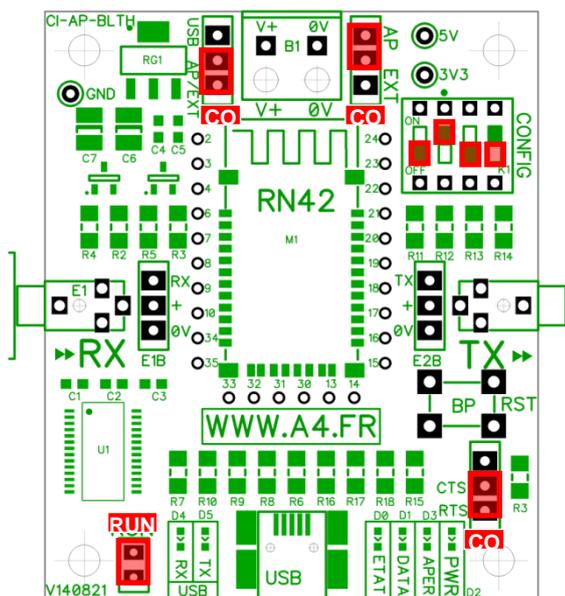
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

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.

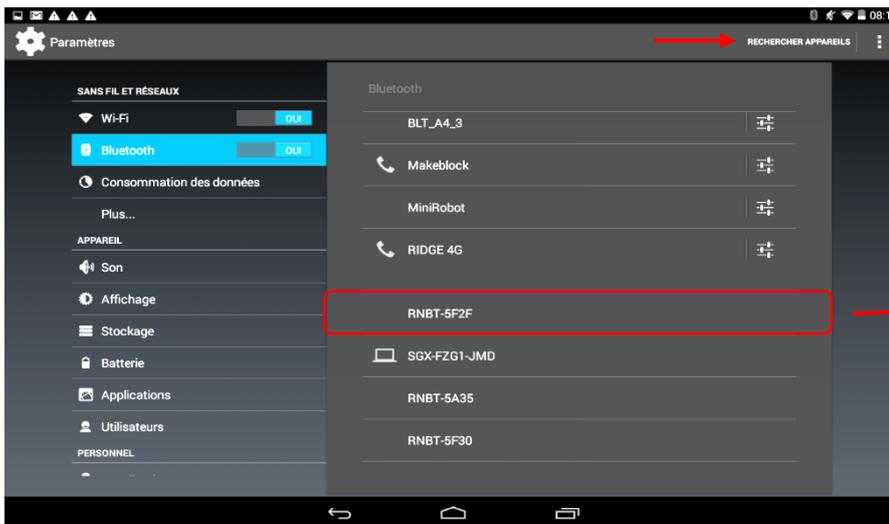
**USB RX** indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

**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.

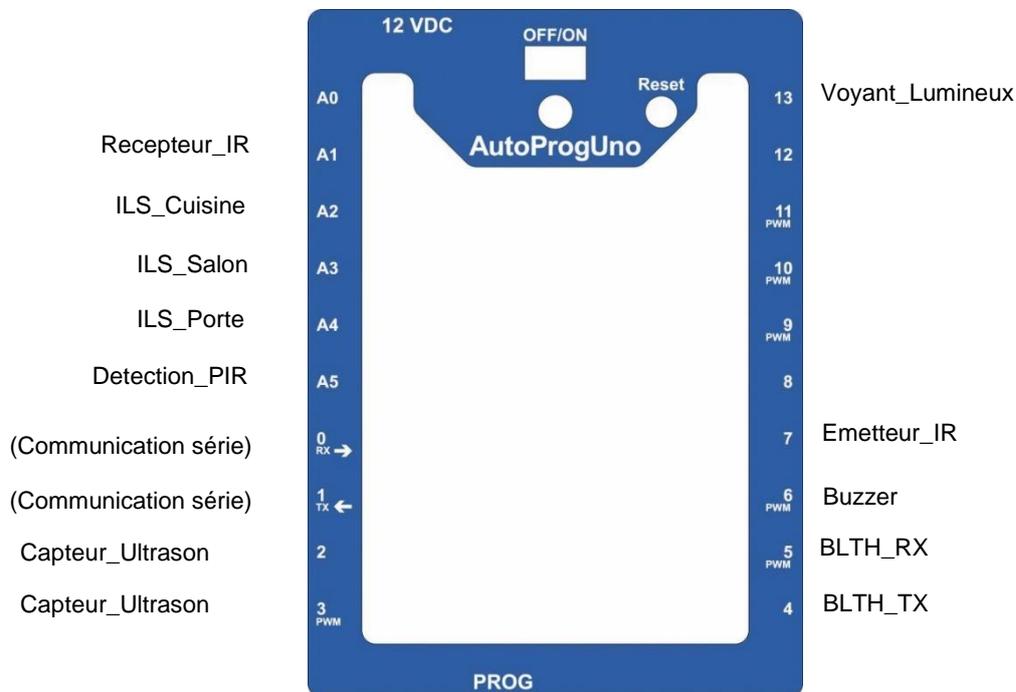
Le témoin vert **DATA** s'allume sur le module dès qu'une donnée est émise ou reçue par le module Bluetooth.

L'appui sur le bouton d'envoi de données, dans cet exemple **Commande portail**, déclenche l'allumage fugitif de ce témoin.



## Tableau d'affectation des entrées et sorties

AutoProgUno	Monte-charge	Nom mBlock
<b>Modules capteurs pour entrées numériques</b>		
2	Capteur à Ultrasons trigger	Capteur_Ultrason
3	Capteur à Ultrasons récepteur	Capteur_Ultrason
4	Module Bluetooth borne TX	BLTH_TX*
5	Module Bluetooth borne RX	BLTH_RX*
6	Module buzzer	Buzzer
7	Emetteur infrarouge	Emetteur_IR
8		
<b>Modules actionneurs pour sorties numériques</b>		
9		
10		
11		
12		
13	Module voyant lumineux jaune	Voyant_Lumineux
<b>Module de communication</b>		
0	(communication avec ordinateur)	
1		
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0		
A1	Récepteur infrarouge	Recepteur_IR
A2	Capteur ILS Cuisine	ILS_Cuisine
A3	Capteur ILS Salon	ILS_Salon
A4	Capteur ILS Porte	ILS_Porte
A5	Module détecteur de présence	Detection_PIR



## Exercice niv. 3 - A.1 : Ouvrir/fermer avec application Bluetooth

**Objectif :** contrôler l'alarme via Bluetooth à partir d'une application pour Smartphone.

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

Donnée Bluetooth reçue

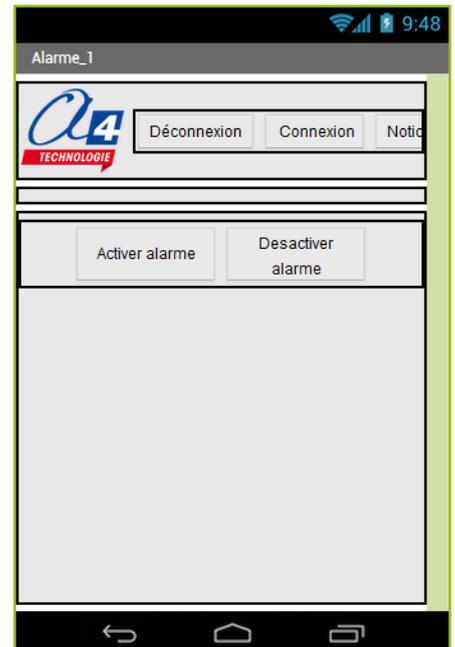
Envoyer donnée Bluetooth 0

**Application Android :** Alarme\_1.apk

**Fichier App Inventor :** Alarme\_1.aia

```
when activer .Click
do call Bluetooth .Send1ByteNumber
   number 1

when desactiver .Click
do call Bluetooth .Send1ByteNumber
   number 2
```



**Correction :**

```
Blocs

Arduino - générer le code
répéter indéfiniment
  si Donnée Bluetooth reçue = 1 alors
    alarme

définir alarme
  répéter jusqu'à Donnée Bluetooth reçue = 2
  Activer Voyant Lumineux
  jouer un son sur la broche Buzzer : note C4 Un demi temps
  Désactiver Voyant Lumineux
  attendre 0.5 secondes
  Désactiver Voyant Lumineux

Fichier mBlock : AL_N3_A1.sb2
```

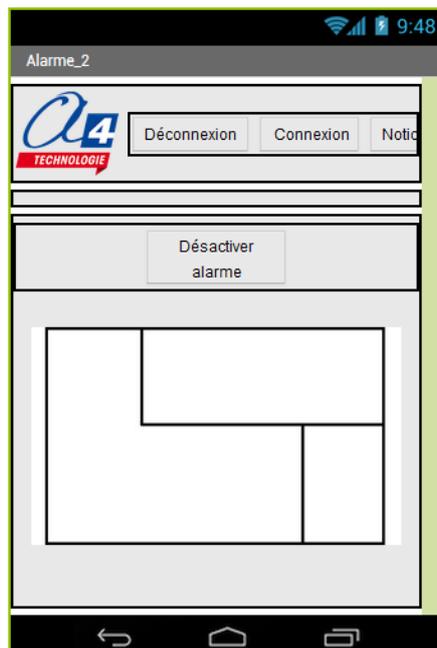
## Exercice niveau 3 - A.2 : Détecter une intrusion via Bluetooth

**Objectif** : détecter où se situe l'intrusion. Activer l'alarme en cas d'intrusion, celle-ci pourra être désactivée grâce à un bouton sur l'application.

**Notion abordée** : réception de données Bluetooth envoyées par un smartphone.

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

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



```
quand desactiver .Clic
faire
  appeler Bluetooth .Envoyer1Octet
  nombre 3
  mettre maison .Image à "maisonvide.jpg"
```

```
initialise global piece à créer une liste
  "infrarouge"
  "pir"
  "ultrason"
  "porte"
  "salon"
  "cuisine"
```

```
quand Clock1 .Chronomètre
faire
  si appeler Bluetooth .Octets disponibles pour le réception > 0
  alors si appeler Bluetooth .Octets disponibles pour le réception = 1
  alors mettre maison .Image à joint choisir liste élément liste obtenir global piece
  index appeler Bluetooth .RecevoirOctetSignéNuméro1
  appeler alerte .Jouer
```

Correction :

Blocs

```
Arduino - générer le code
Activer Emetteur IR
attendre 1 secondes
répéter indéfiniment
  si non Bouton-poussoir ILS_Cuisine appuyé alors
    Envoyer donnée Bluetooth 6
    alarme
    attendre 1 secondes
  si non Bouton-poussoir ILS_Salon appuyé alors
    Envoyer donnée Bluetooth 5
    alarme
    attendre 1 secondes
  si non Bouton-poussoir ILS_Porte appuyé alors
    Envoyer donnée Bluetooth 4
    alarme
    attendre 1 secondes
  si Bouton-poussoir Detection_PIR appuyé alors
    Envoyer donnée Bluetooth 2
    alarme
    attendre 1 secondes
  si Entrée Recepteur_IR activée alors
    Envoyer donnée Bluetooth 1
    alarme
    attendre 1 secondes
définir alarme
répéter jusqu'à Donnée Bluetooth reçue = 3
  Activer Voyant Lumineux
  jouer un son sur la broche Buzzer : note C4 Un demi temps
  Désactiver Voyant Lumineux
  attendre 0.5 secondes
Désactiver Voyant Lumineux
```

Fichier mBlock : AL\_N3\_B2.sb2

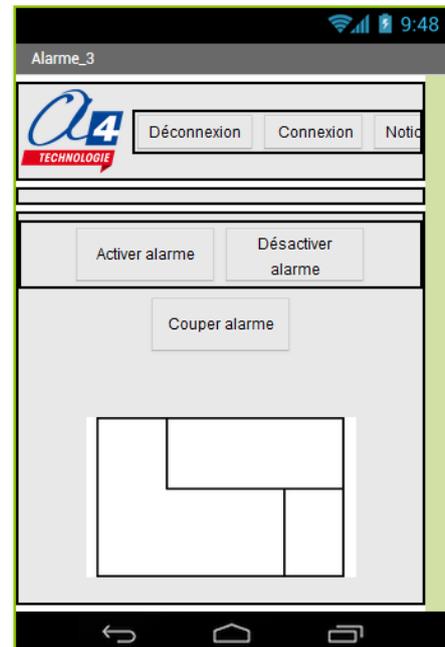
# Exercice niveau 3 - A.3 (difficile): Système d'alarme activable

**Objectif** : détecter où se situe l'intrusion. Activer l'alarme ou non en cas d'intrusion.

**Notion abordée** : créer un programme permettant à l'alarme d'être activée ou désactivée. C'est-à-dire qu'il va ou non chercher s'il y a une intrusion. Ajouter également un bouton pour couper l'alarme.

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

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



```
quand couper .Clic
faire
  appeler Bluetooth .Envoyer1Octet
    nombre 3
  mettre maison .Image à "maisonvide.jpg"
```

```
initialise global piece à créer une liste
  "infrarouge"
  "pir"
  "ultrason"
  "porte"
  "salon"
  "cuisine"
```

```
quand Clock1 .Chronomètre
faire
  si appeler Bluetooth .Octets disponibles pour le réception > 0
  alors si appeler Bluetooth .Octets disponibles pour le réception = 1
  alors mettre maison .Image à joint choisir liste élément liste obtenir global piece
    index appeler Bluetooth .RecevoirOctetSignéNuméro1
    "jpg"
  appeler alerte .Jouer
```

```
quand activer .Clic
faire
  appeler Bluetooth .Envoyer1Octet
    nombre 1
  mettre maison .Image à "maisonvideactive.jpg"
```

```
quand desactiver .Clic
faire
  appeler Bluetooth .Envoyer1Octet
    nombre 2
  mettre maison .Image à "maisonvidedesactive.jpg"
```

## Correction :

Blocs

Fichier mBlock : AL\_N3\_B3.sb2

**Remarques :** Il faut couper l'alarme avant de la désactiver  
Le bouton « couper l'alarme » éteint simplement l'alarme, mais elle pourra se déclencher à nouveau.



CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES