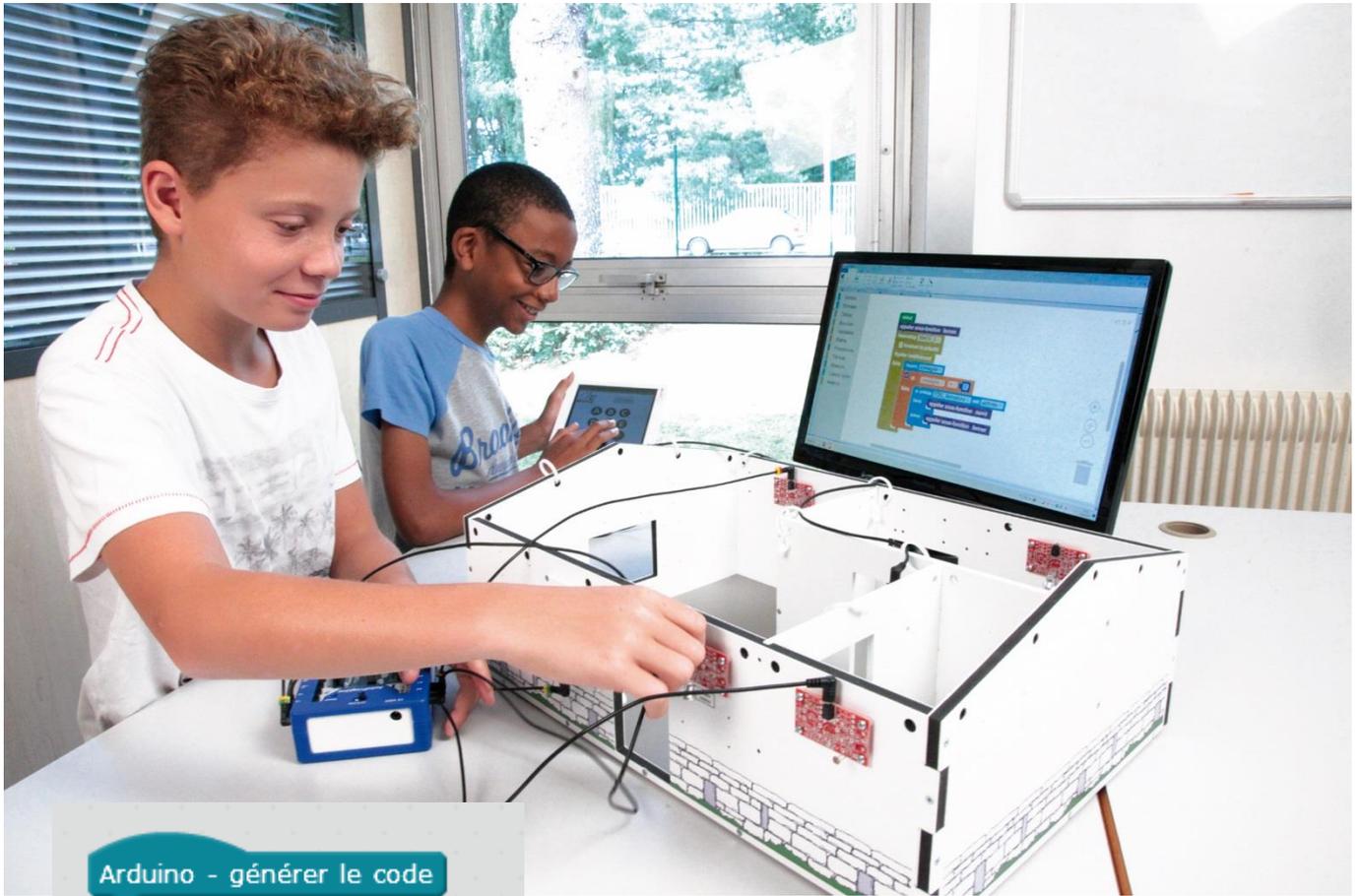


# AutoLumi

## 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 AutoLumi

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

## AutoLumi

- Fichiers **3D** (SolidWorks, Edrawings et Parasolid) de la maquette et de ses options.
- Dossier **technique** AutoLumi 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>
AutoLumi.....	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 niv. 1 - A.1 : Allumer / éteindre une lumière .....	6
Exercice niv. 1 - A.2: Allumer / éteindre une lumière deux fois .....	7
Exercice niv. 1 - A.3 : Allumer / éteindre une lumière indéfiniment .....	8
<b>Niveau 1 - B.....</b>	<b>9</b>
Exercice niv. 1 - B.1 : Allumer une lumière manuellement.....	9
Exercice niv. 1 - B.2 : Allumer/éteindre une lumière manuellement.....	10
Exercice niv. 1 - B.3 : Allumer une lumière tant que le bouton est actionné .....	11
<b>Niveau 1 - C.....</b>	<b>12</b>
Exercice niv. 1 - C.1 : Allumer / éteindre toutes les lumières successivement .....	12
Exercice niv. 1 - C.2 : Allumer / éteindre toutes les lumières successivement avec un bouton .....	13
<b>Niveau 1 - D.....</b>	<b>14</b>
Exercice niv. 1 – D.1 : Allumer/éteindre la lumière en fonction de la luminosité.....	14
Exercice niv. 1 – D.2 : Allumer la lumière si une présence est détectée et en fonction de la luminosité .....	15
<b>Programmation version de base niveau 2 .....</b>	<b>16</b>
<b>Niveau 2 - A.....</b>	<b>17</b>
Exercice niv. 2 – A.1 : Utiliser une sous-fonction pour allumer la lumière .....	17
Exercice niv. 2 – A.2 : Utiliser une sous-fonction pour allumer la lumière lorsqu'une présence est détectée.....	18
<b>Niveau 2 - B.....</b>	<b>19</b>
Exercice niv. 2 – B.1 : Utilisation de variables.....	19
Exercice niv. 2 – B.2 : Utilisation de variables(2) .....	20
<b>Niveau 2 - C.....</b>	<b>21</b>
Exercice niv. 2 – C.1 : Automatiser l'éclairage de la maquette .....	21
<b>Programmation niveau 3 - Option : Module Bluetooth .....</b>	<b>22</b>
Configuration .....	22
Mise en place des programmes et procédure de connexion.....	23
Tableau d'affectation des entrées et sorties.....	24
Exercice niv. 3 - B.1 : Allumer/éteindre une lumière avec votre smartphone (deux boutons) .....	25
Exercice niv. 3 - B.2 : Allumer/éteindre une lumière avec votre smartphone (un bouton) .....	26
Exercice niv. 3 - B.3 : Envoi de données sur le smartphone .....	27
Exercice niv. 3 - B.4 : Envoyer et recevoir des données sur un smartphone.....	28

# Introduction

---

## AutoLumi

La maquette AutoLumi (BE-ALUMI) est une reproduction homothétique d'une maison éclairable automatisable : lumières, interrupteurs, capteur de lumière, capteurs de présence. 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 avec AutoLumi :

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

## Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette AutoLumi 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\_Autolumi** (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 Blockly.

## Le dossier

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

Il est organisé en fonction des niveaux de programmation.

### Niveau 1 :

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

### Niveau 2 :

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

### Niveau 3 :

Exemples d'utilisation des différentes options proposées : télécommande infrarouge, module de détection de mouvement PIR, 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 :

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

Exemple : LU\_N2\_C1.sb2

Correspond au niveau 2 avec le jeu d'instructions C, adapté aux objectifs « avancés » de ce niveau.

## Prérequis

Pour la version de base :

- Installer le logiciel **mBlock**.
- Installer l'extension **A4\_Autolumi** (fichier zip) dans mBlock.
- **Maquette** AutoLumi (Réf. BE-ALUMI).
- **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.

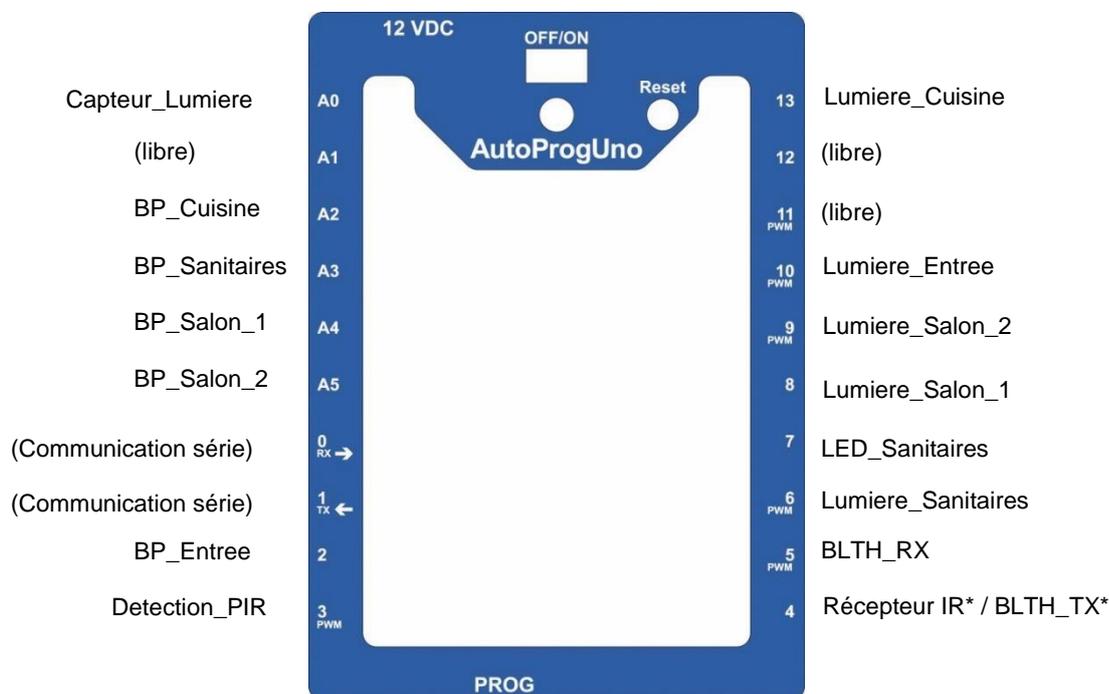
Pour l'option Bluetooth :

- **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	AutoLumi	Nom mBlock
<b>Modules capteurs pour entrées numériques</b>		
2	Bouton poussoir entrée	BP_Entree
3	Module détecteur de présence	Detection_PIR
4	Communication IR ou BT (Option)	Récepteur IR* / BLTH_TX*
5	Module Bluetooth entrée (RX)	BLTH_RX
<b>Modules actionneurs pour sorties numériques</b>		
6	Lumière des sanitaires	Lumiere_Sanitaires
7	LES des sanitaires	LED_Sanitaires
8	Lumière à l'entrée du salon	Lumiere_Salon_1
9	Lumière au fond du salon	Lumiere_Salon_2
10	Lumière devant la porte d'entrée	Lumiere_Entree
11		
12		
13	Lumière de la cuisine	Lumiere_Cuisine
<b>Module de communication</b>		
1	(communication avec ordinateur)	
2		
6		
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0	Capteur de lumière	Capteur_Lumiere
A1		
A2	Bouton-poussoir cuisine	BP_Cuisine
A3	Bouton sanitaires	BP_Sanitaires
A4	Bouton entrée du salon	BP_Salon_1
A5	Bouton fond du salon	BP_Salon_2
12		



# 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 d'AutoLumi, 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 (fourni avec les exercices), avant de regarder la correction et l'explication de chaque programme.

Par exemple, pour le programme « LU\_N1\_A1.sb2 », charger le programme modèle « LU\_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> LU_N1_A.sb2		
LU_N1_A1	Allumer une lumière pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordée : -Allumage/extinction d'une lumière.
LU_N1_A2	Répéter cette action deux fois.	Notions de programmation abordées : -séquence d'instructions -temps d'attente -boucle infinie
LU_N1_A3	Répéter cette action à l'infini.	
<b>Niveau 1 B</b> <b>Fichier modèle :</b> LU_N1_B.sb2		
LU_N1_B1	Activer une lumière sur l'appui d'un bouton-poussoir.	Fonctionnalité matérielle abordée : -utilisation de bouton-poussoir -utilisation d'un capteur tout ou rien
LU_N1_B2	Activer/désactiver une lampe à l'aide d'un bouton-poussoir.	
LU_N1_B3	Utiliser une boucle tant que.	Notions de programmation abordées : -boucle qui dépend d'une entrée
LU_N1_B4	Utilisation d'un capteur tout ou rien.	
<b>Niveau 1 C</b> <b>Fichier modèle :</b> LU_N1_C.sb2		
LU_N1_C1	Activer une à une les lumières de la maison. Quand une lumière s'allume, une autre doit s'éteindre.	Fonctionnalité matérielle abordée : -Gestion de sorties
LU_N1_C2	Activer ou désactiver chaque lampe à l'aide de son bouton-poussoir.	Notions de programmation abordées : -Le test d'une entrée
<b>Niveau 1 D</b> <b>Fichier modèle :</b> LU_N1_D.sb2		
LU_N1_D1	Activer une lumière lorsque la salle est éteinte.	Fonctionnalité matérielle abordée :
LU_N1_D2	Lorsque la salle est éteinte et que le capteur PIR détecte une personne, allumer la lumière de l'entrée pendant 5 secondes.	Notions de programmation abordées : -Définition de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage

# Niveau 1 - A

---

## Exercice niv. 1 - A.1 : Allumer / éteindre une lumière

Fichier modèle : LU\_BASE.sb2

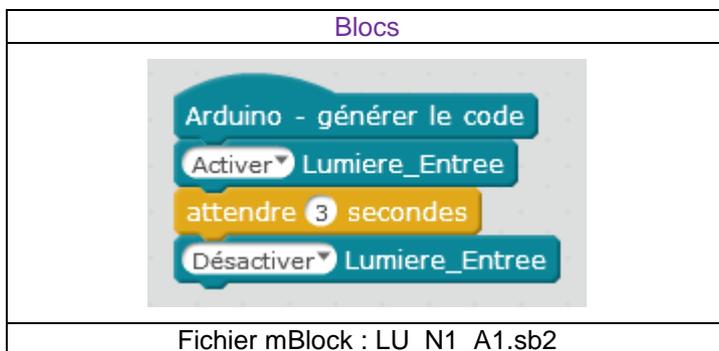
**Objectif** : allumer une lumière pendant 3 secondes puis l'éteindre.

**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 niv. 1 - A.2: Allumer / éteindre une lumière deux fois

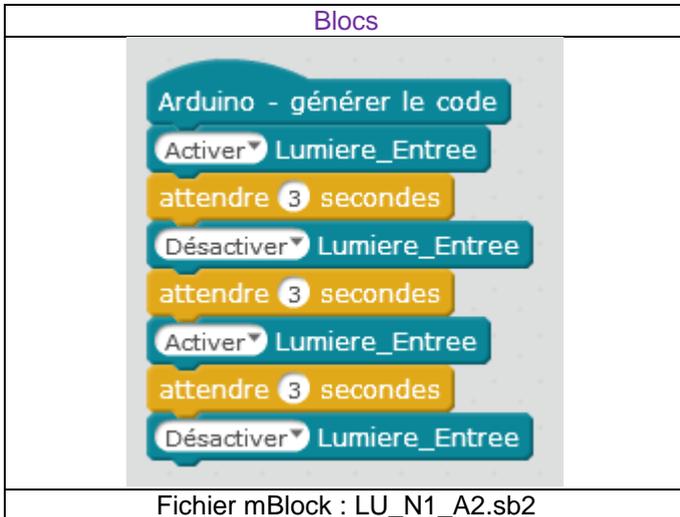
**Objectif** : allumer une lumière pendant 3 secondes puis l'éteindre, recommencer.

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

**Instructions utilisées** :



**Correction** :



## Exercice niv. 1 - A.3 : Allumer / éteindre une lumière indéfiniment

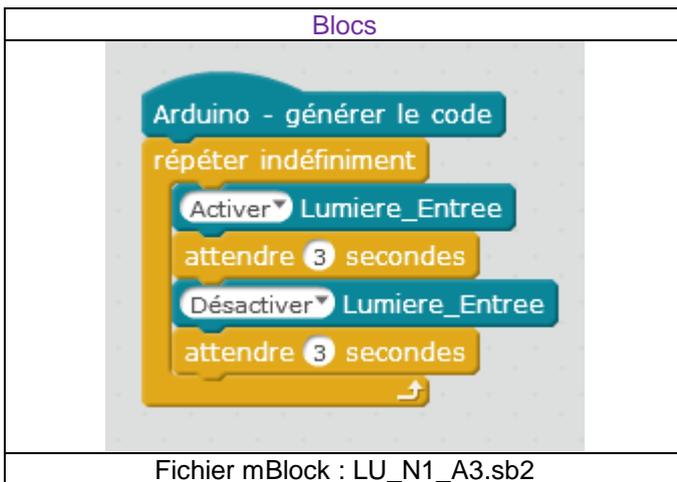
**Objectif** : faire clignoter une lumière toutes les 3 secondes indéfiniment.

**Notion abordée** : la boucle infinie.

**Instructions utilisées** :



**Correction** :



**Remarque** : le programme ne peut pas 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 l'interface AutoProg.

# Niveau 1 - B

## Exercice niv. 1 - B.1 : Allumer une lumière manuellement

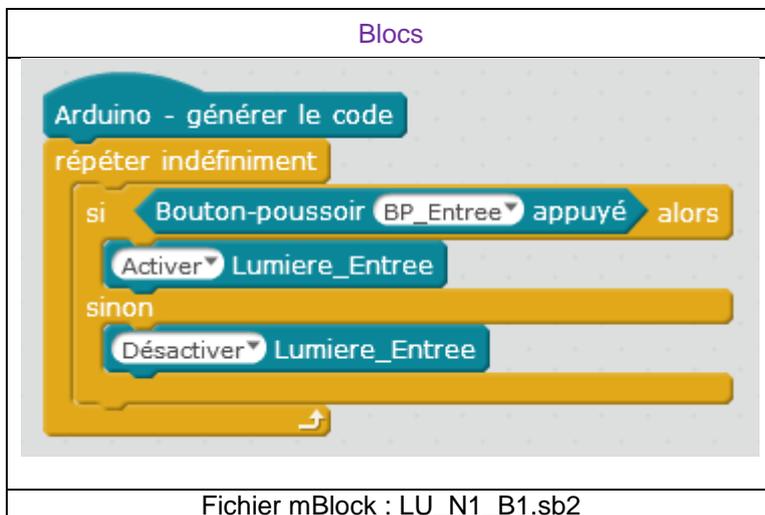
**Objectif** : Activer une lumière en appuyant sur un bouton-poussoir.

**Notion abordée** : utilisation d'une entrée bouton-poussoir et d'une condition.

**Instructions utilisées** :



**Correction** :

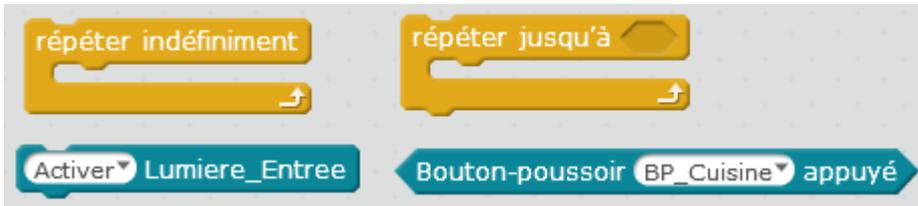


## Exercice niv. 1 - B.2 : Allumer/éteindre une lumière manuellement

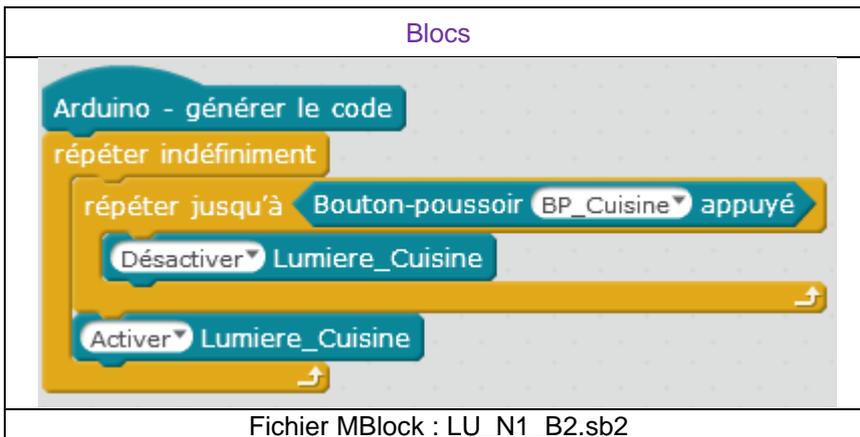
**Objectif :** Activer une lampe à l'aide d'un bouton-poussoir. Pour l'éteindre, il faut réappuyer sur le bouton.

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

**Instructions utilisées :**



**Correction :**



## Exercice niv. 1 - B.3 : Allumer une lumière tant que le bouton est actionné

**Objectif :** Utiliser une nouvelle boucle

**Notion abordée :** utilisation d'une boucle tant que.

**Instructions utilisées :**



**Correction :**

Blocs



Fichier mBlock : LU\_N1\_B3.sb2

# Niveau 1 - C

## Exercice niv. 1 - C.1 : Allumer / éteindre toutes les lumières successivement

**Objectif :** Activer une à une les lumières de la maison. Quand une lumière s'allume, une autre doit s'éteindre.

**Notion abordée :** succession de consignes.

**Instructions utilisées :**



**Correction :**

Blocs



```
Arduino - générer le code
répéter indéfiniment
  Activer Lumiere_Cuisine
  attendre 1 secondes
  Désactiver Lumiere_Cuisine
  Activer Lumiere_Sanitaires
  attendre 1 secondes
  Désactiver Lumiere_Sanitaires
  Activer Lumiere_Entree
  attendre 1 secondes
  Désactiver Lumiere_Entree
  Activer Lumiere_Salon_1
  attendre 1 secondes
  Désactiver Lumiere_Salon_1
  Activer Lumiere_Salon_2
  attendre 1 secondes
  Désactiver Lumiere_Salon_2
```

Fichier mBlock : LU\_N1\_C1.sb2

## Exercice niv. 1 - C.2 : Allumer / éteindre toutes les lumières successivement avec un bouton

**Objectif :** Activer ou désactiver chaque lampe à l'aide de son bouton-poussoir.  
La LED des sanitaires doit s'activer en même temps que la lumière des sanitaires.

**Notion abordée :** boucles multiples.

**Instructions utilisées :**



**Correction :**

Blocs

Le code de programmation mBlock est structuré comme suit :

- Arduino - générer le code
- répéter indéfiniment
  - si Bouton-poussoir BP\_Cuisine appuyé alors
    - Activer Lumiere\_Cuisine
  - sinon
    - Désactiver Lumiere\_Cuisine
  - si Bouton-poussoir BP\_Sanitaires appuyé alors
    - Activer Lumiere\_Sanitaires
    - Activer LED\_Sanitaires
  - sinon
    - Désactiver Lumiere\_Sanitaires
    - Désactiver LED\_Sanitaires
  - si Bouton-poussoir BP\_Salon\_1 appuyé alors
    - Activer Lumiere\_Salon\_1
  - sinon
    - Désactiver Lumiere\_Salon\_1
  - si Bouton-poussoir BP\_Salon\_2 appuyé alors
    - Activer Lumiere\_Salon\_2
  - sinon
    - Désactiver Lumiere\_Salon\_2
  - si Bouton-poussoir BP\_Entree appuyé alors
    - Activer Lumiere\_Entree
  - sinon
    - Désactiver Lumiere\_Entree

Fichier mBlock : LU\_N1\_C2.sb2

# Niveau 1 - D

## Exercice niv. 1 – D.1 : Allumer/éteindre la lumière en fonction de la luminosité

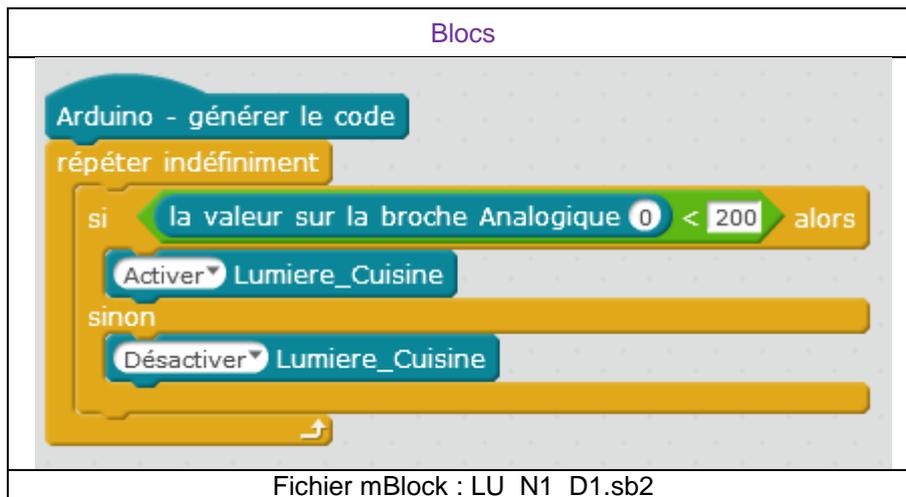
**Objectif** : Activer une lumière lorsque la salle est éteinte.

**Notion abordée** : conditions sur des valeurs analogiques.

**Instructions utilisées** :



**Correction** :



**Remarque** : un capteur analogique envoie une valeur entre 0 et 255. On considère que le capteur donne 0 dans un environnement totalement plongé dans le noir, et augmente avec le niveau de lumière. On considère qu'une salle est sombre lorsque la valeur passe en dessous de 200, nous fixerons donc la valeur analogique à 150.

La broche Analogique 0 correspond à la broche A0 de l'interface AutoProgUno.

## Exercice niv. 1 – D.2 : Allumer la lumière si une présence est détectée et en fonction de la luminosité

**Objectif** : Lorsque la salle est éteinte et que le capteur PIR détecte une personne, allumer la lumière de l'entrée pendant 5 secondes. La lumière ne doit s'allumer que s'il fait sombre.

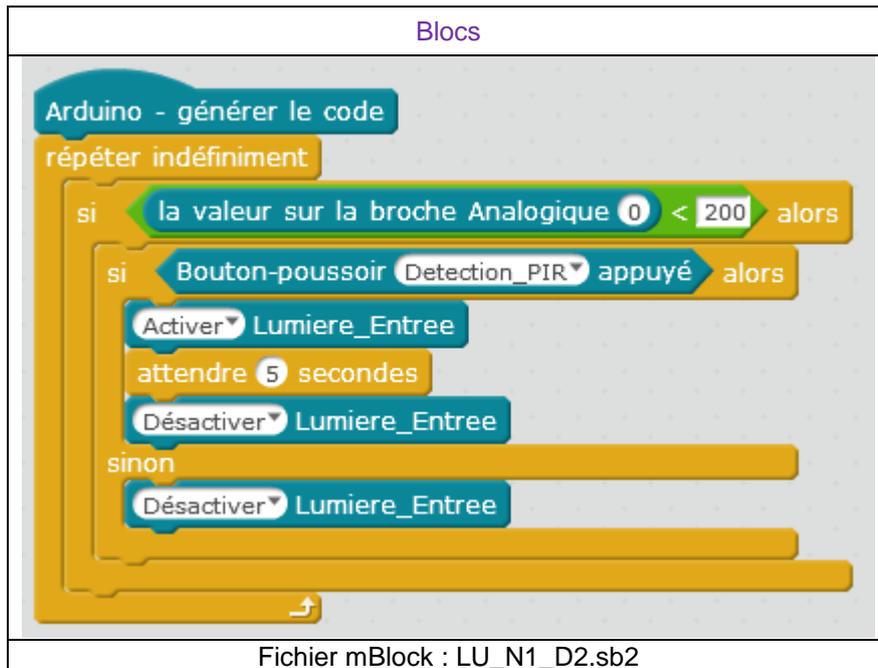
**Notion abordée** : conditions sur des valeurs analogiques.

**Instructions utilisées** :



**Correction** :

Blocs



Fichier mBlock : LU\_N1\_D2.sb2

# Programmation version de base niveau 2

## Objectifs :

- Utilisation concrète d'AutoLumi
- 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> LU_N2_A.sb2		
LU_N2_A1	Créer une sous-fonction permettant d'allumer une lumière à l'appui d'un bouton-poussoir.	Notions de programmation abordées : -Utilisation des sous-fonctions
LU_N2_A2	Créer une sous-fonction qui allume la lumière pendant 5 secondes lorsque le capteur de présence détecte quelqu'un.	
<b>Niveau 2 B</b> <b>Fichier modèle :</b> LU_N2_B.sb2		
LU_N2_B1	Créer une boucle qui augmente la valeur d'une variable toutes les secondes.	Notions de programmation abordées : -Manipulation de variables
LU_N2_B2	Créer une boucle qui augmente la valeur d'une variable toutes les secondes et à 10 secondes allume une lumière 3 secondes puis remet la variable à 0.	
<b>Niveau 2 C</b> <b>Fichier modèle :</b> LU_N2_C.sb2		
LU_N2_C1	Créer un programme chargé de rendre possible tout l'éclairage de la maison.	Automatisation de la maquette

## Niveau 2 - A

### Exercice niv. 2 – A.1 : Utiliser une sous-fonction pour allumer la lumière

**Objectif** : Créer une sous-fonction permettant d'allumer une lumière à l'appui d'un bouton-poussoir.

**Notion abordée** : sous-fonction.

**Instructions utilisées** :



**Correction** :

Blocs

Fichier mBlock : LU\_N2\_A1.sb2

**Remarque** : Le programme principal (sous le bloc début) ne doit contenir que la boucle de répétition et l'appel de la sous-fonction.

## Exercice niv. 2 – A.2 : Utiliser une sous-fonction pour allumer la lumière lorsqu'une présence est détectée

**Objectif** : Créer une sous-fonction qui allume la lumière pendant 5 secondes lorsque le capteur de présence détecte quelqu'un. La lumière doit aussi pouvoir s'activer et se désactiver grâce à un bouton-poussoir.

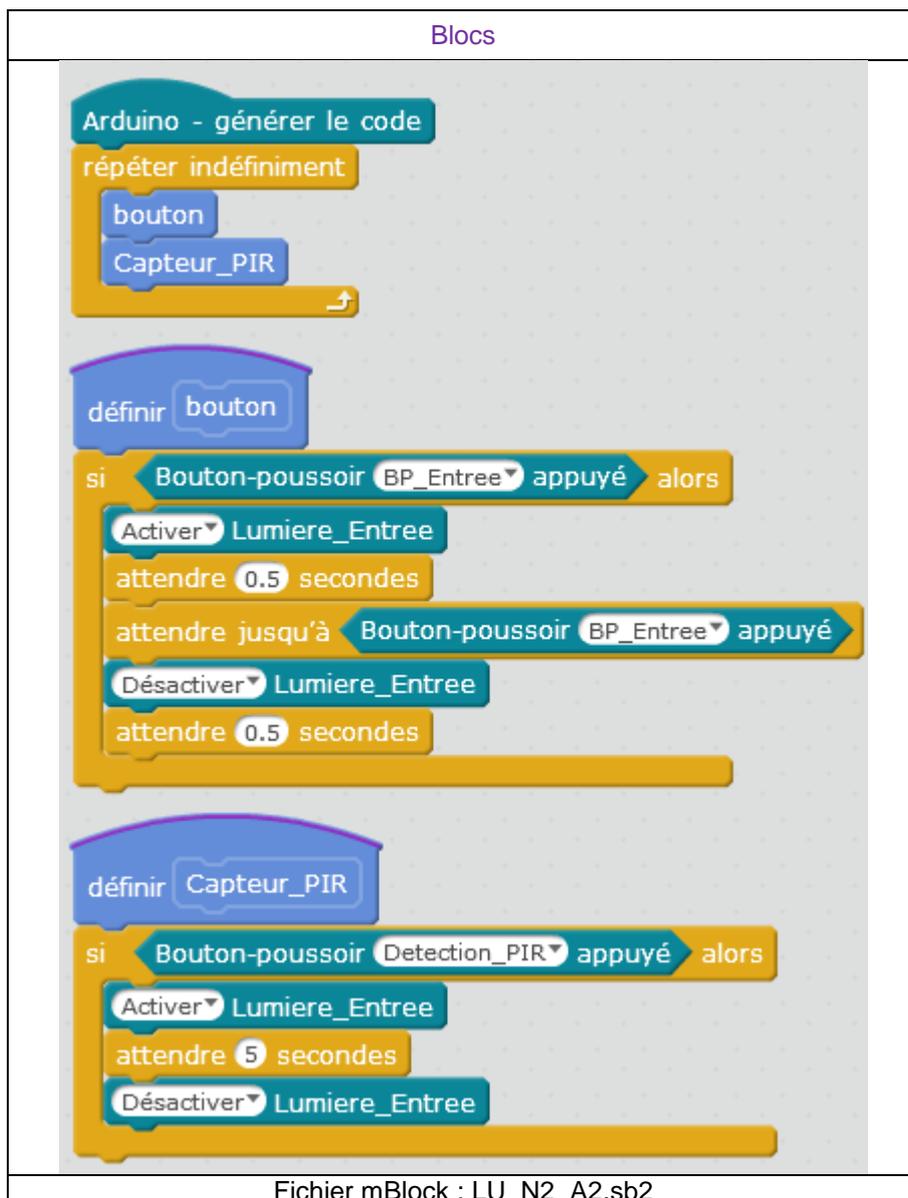
**Notion abordée** : Gestion de sous-fonctions.

**Instructions utilisées** :



**Correction** :

Blocs



Fichier mBlock : LU\_N2\_A2.sb2

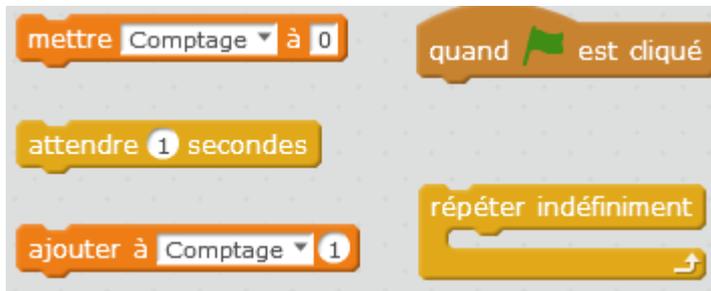
# Niveau 2 - B

## Exercice niv. 2 – B.1 : Utilisation de variables

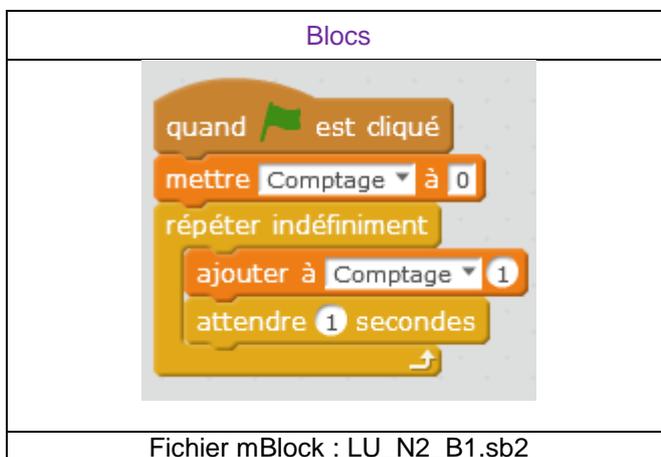
**Objectif** : Créer une boucle qui augmente la valeur d'une variable toutes les secondes.

**Notion abordée** : gestion de variables.

**Instructions utilisées** :



**Correction** :



## Exercice niv. 2 – B.2 : Utilisation de variables(2)

**Objectif** : Créer une boucle qui augmente la valeur d'une variable toutes les secondes et à 10 secondes allume une lumière pendant 3 secondes puis remet la variable à 0.

**Notion abordée** : gestion de variables.

**Instructions utilisées** :



**Correction** :

Blocs

Arduino - générer le code  
mettre Comptage à 0  
répéter indéfiniment  
ajouter à Comptage 1  
attendre 1 secondes  
si Comptage = 10 alors  
Activer Lumiere\_Entree  
attendre 3 secondes  
Désactiver Lumiere\_Entree  
mettre Comptage à 0

Fichier mBlock : LU\_N2\_B2.sb2

# Niveau 2 - C

## Exercice niv. 2 – C.1 : Automatiser l'éclairage de la maquette

**Objectif :** Créer un programme chargé de rendre possible tout l'éclairage de la maison.

- L'appui sur un bouton allume ou éteint la lumière correspondante (comme un vrai interrupteur).
- Un passage sur le capteur de présence PIR allume la lumière puis l'éteint après quelques secondes.
- Si la lumière extérieure est trop forte, toutes les lumières s'éteignent ou doivent rester éteintes sauf les sanitaires.

**Indice :** utiliser des sous-fonctions et une variable pour le capteur de présence.

**Correction :**

Blocs

```
definiere boutons
si la valeur sur la broche Analogique 0 < 200 alors
  si Bouton-poussoi BP Cuisine appuyé alors
    Active Lumiere_Cuisine
  si Bouton-poussoi BP Salon 1 appuyé alors
    Active Lumiere_Salon_1
  si Bouton-poussoi BP Salon 2 appuyé alors
    Active Lumiere_Salon_2
  si Bouton-poussoi BP Entree appuyé alors
    Active Lumiere_Entree
  si Bouton-poussoi BP Sanitaires appuyé alors
    Active LED_Sanitaires
    Active Lumiere_Sanitaires
  sinon
    Désactive Lumiere_Cuisine
    Désactive Lumiere_Salon_1
    Désactive Lumiere_Salon_2
    Désactive Lumiere_Entree
    Désactive Lumiere_Sanitaires
    Désactive LED_Sanitaires

Arduino - générer le code
répéter indéfiniment
  boutons
  Detection_PIR

définir Detection_PIR
si la valeur sur la broche Analogique 0 < 200 alors
  si Bouton-poussoi Detection_PIR appuyé alors
    Active Lumiere_Entree
  sinon
    Désactive Lumiere_Entree
sinon
  Désactive Lumiere_Entree
```

Fichier mBlock : LU\_N2\_C1.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.

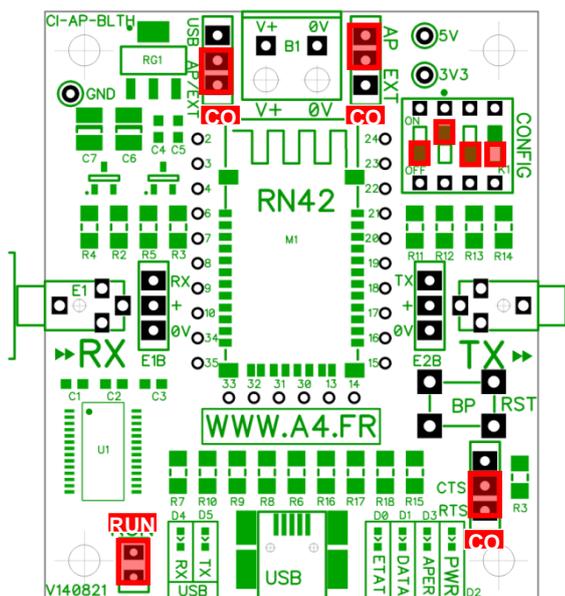
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

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.

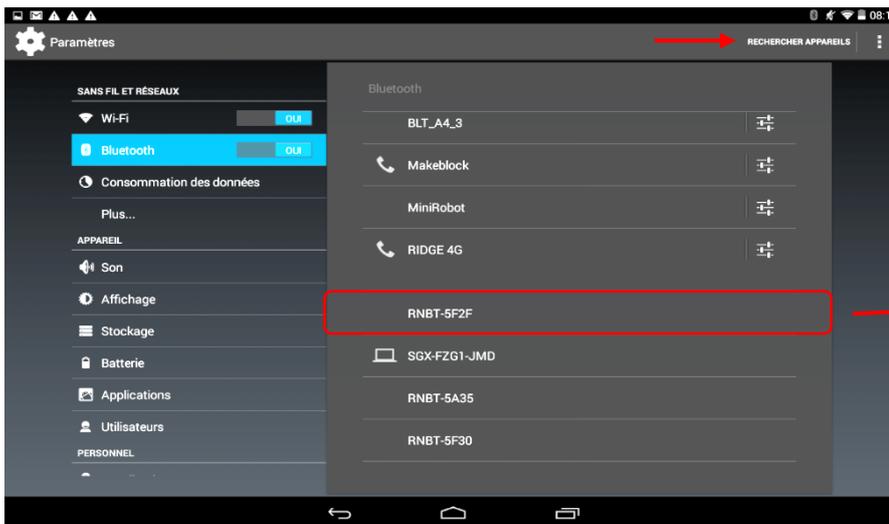
**USB RX** indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

**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.

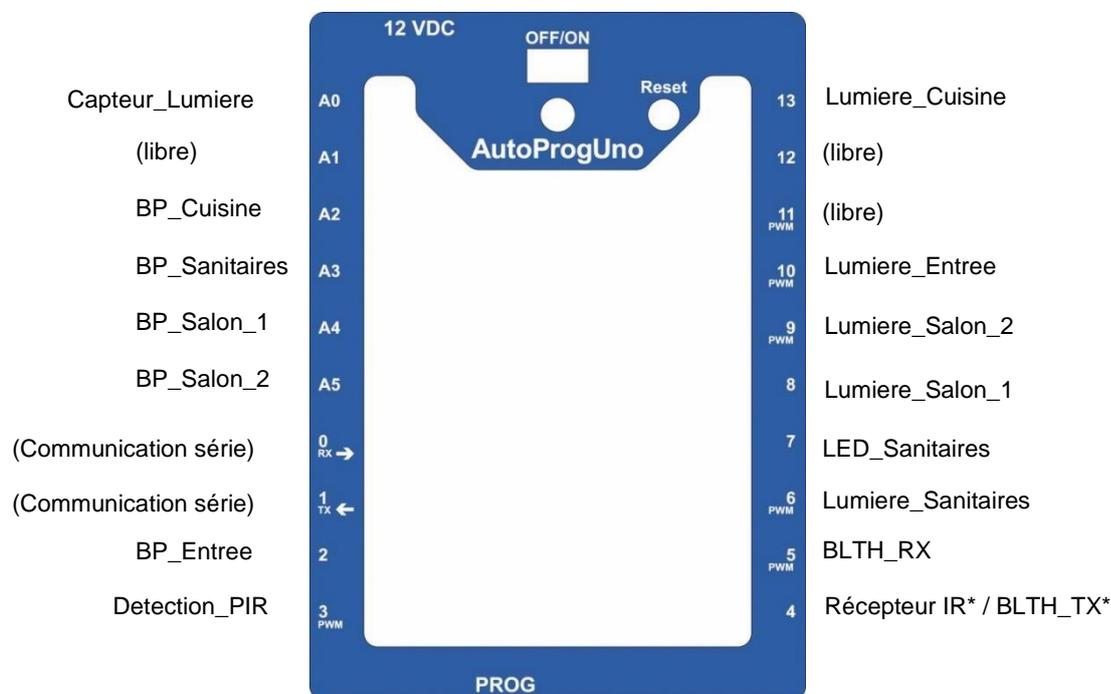
Le témoin vert **DATA** s'allume sur le module dès qu'une donnée est émise ou reçue par le module Bluetooth.

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

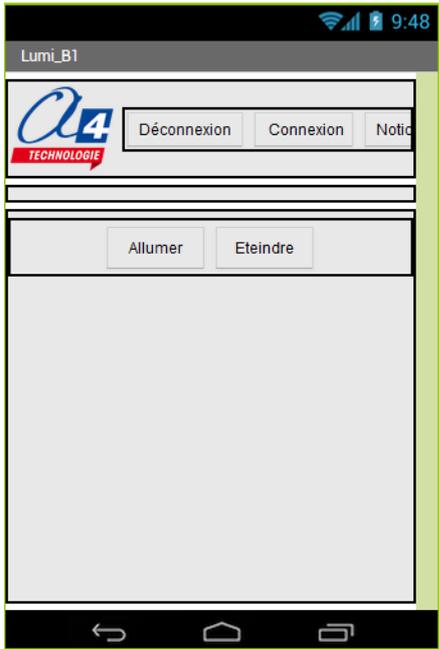
AutoProgUno	Monte-charge	Nom mBlock
<b>Modules capteurs pour entrées numériques</b>		
2	Bouton poussoir entrée	BP_Entree
3	Module détecteur de présence	Detection_PIR
4	Communication IR ou BT (Option)	Récepteur IR* / BLTH_TX*
5	Module Bluetooth entrée (RX)	BLTH_RX
<b>Modules actionneurs pour sorties numériques</b>		
6	Lumière des sanitaires	Lumiere_Sanitaires
7	LES des sanitaires	LED_Sanitaires
8	Lumière à l'entrée du salon	Lumiere_Salon_1
9	Lumière au fond du salon	Lumiere_Salon_2
10	Lumière devant la porte d'entrée	Lumiere_Entree
11		
12		
13	Lumière de la cuisine	Lumiere_Cuisine
<b>Module de communication</b>		
1	(communication avec ordinateur)	
2		
6		
<b>Entrées / sorties libres (A pour les analogiques)</b>		
A0	Capteur de lumière	Capteur_Lumiere
A1		
A2	Bouton poussoir cuisine	BP_Cuisine
A3	Bouton sanitaires	BP_Sanitaires
A4	Bouton entrée du salon	BP_Salon_1
A5	Bouton fond du salon	BP_Salon_2
12		



# Exercice niv. 3 - B.1 : Allumer/éteindre une lumière avec votre smartphone (deux boutons)

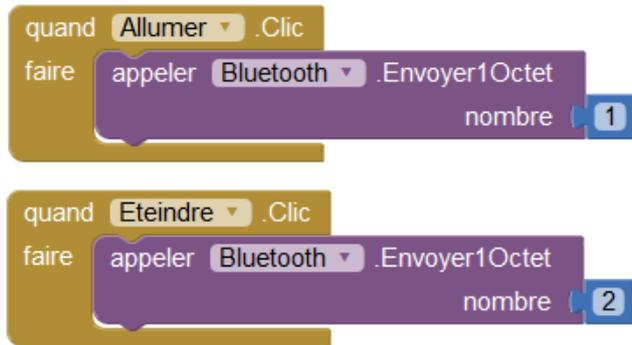
**Objectif :** contrôler une lumière à l'aide de 2 boutons présent sur l'application Android **Lumi\_B1** développée sous AppInventor.

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

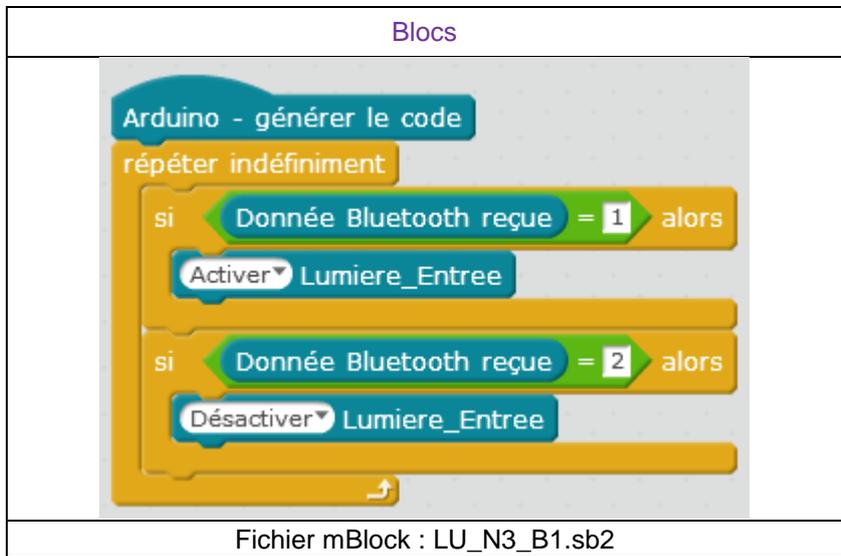


**Application Android :** Lumi\_B1.apk

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



**Correction :**



## Exercice niv. 3 - B.2 : Allumer/éteindre une lumière avec votre smartphone (un bouton)

**Objectif** : contrôler une lumière à l'aide de 1 bouton présent sur l'application Android **Lumi\_B2** développée sous AppInventor.

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

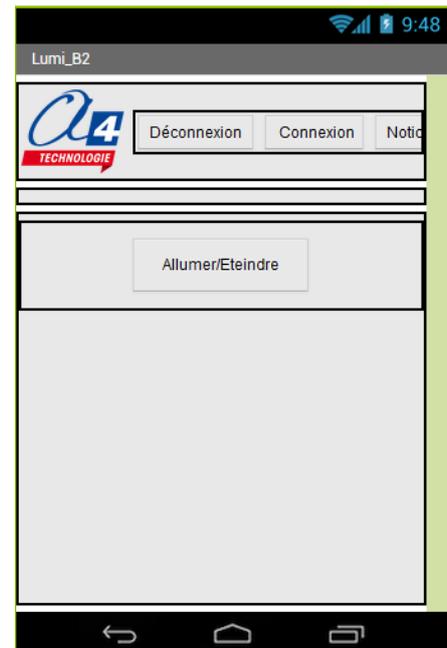
Donnée Bluetooth reçue

Envoyer donnée Bluetooth 0

**Application Android** : Lumi\_B2.apk

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

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



**Correction** :

```
Blocs
Arduino - générer le code
mettre Comptage à 0
répéter indéfiniment
  si Comptage = 0 alors
    si Donnée Bluetooth reçue = 1 alors
      mettre Comptage à 1
      Activer Lumiere_Entree
    si Comptage = 1 alors
      si Donnée Bluetooth reçue = 1 alors
        mettre Comptage à 0
        Désactiver Lumiere_Entree
```

Fichier mBlock : LU\_N3\_B2.sb2

## Exercice niv. 3 - B.3 : Envoi de données sur le smartphone

**Objectif** : jouer un son sur le smartphone lors d'un appui sur un bouton-poussoir de la maquette.

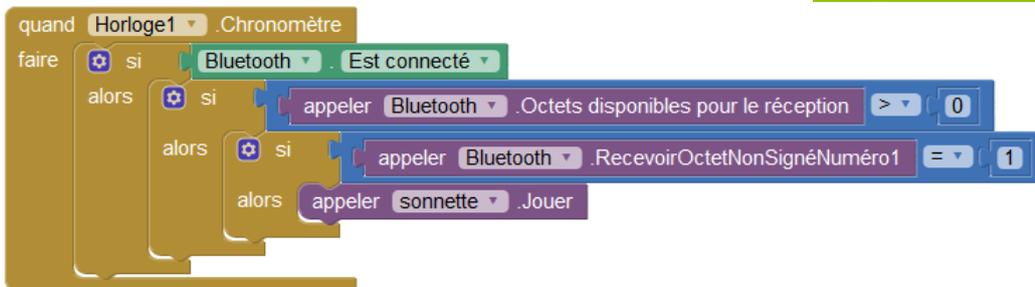
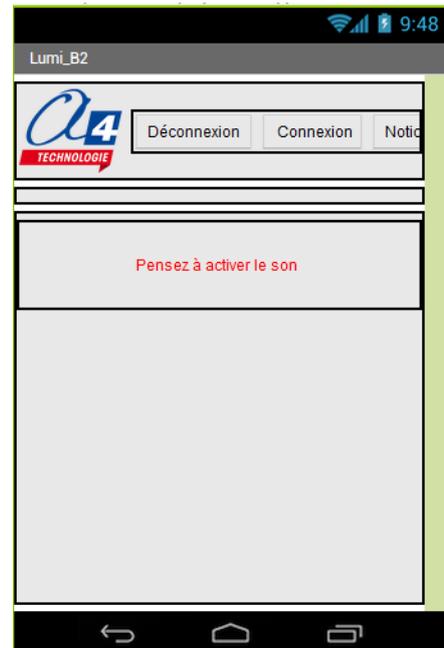
**Notion abordée** : Envoi de données Bluetooth vers un Smartphone.

Donnée Bluetooth reçue

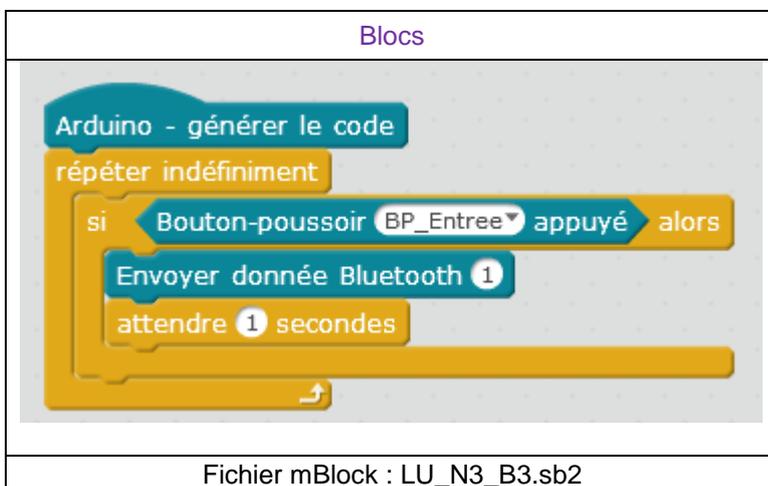
Envoyer donnée Bluetooth 0

**Application Android** : Lumi\_B3.apk

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



**Correction** :



Fichier mBlock : LU\_N3\_B3.sb2

# Exercice niv. 3 - B.4 : Envoyer et recevoir des données sur un smartphone

**Objectif :** Envoyer une information à un smartphone et la traiter sur celui-ci.

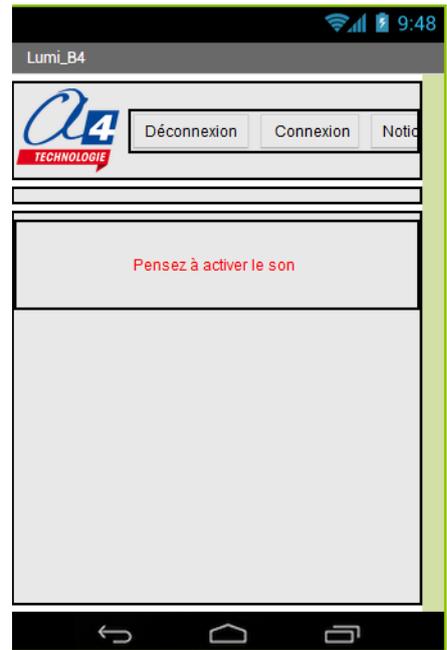
**Notion abordée :** Envoi et réception de données Bluetooth sur un Smartphone.

Donnée Bluetooth reçue

Envoyer donnée Bluetooth 0

**Application Android :** Lumi\_B4.apk

**Fichier App Inventor :** Lumi\_4.aia



```

quand Horloge1.Chronomètre
faire
  si Bluetooth.Est connecté
  alors
    si Bluetooth.Octets disponibles pour le réception > 0
    alors
      si Bluetooth.RecevoirOctetNonSignéNuméro1 = 1
      alors
        appeler sonnette.Jouer
        appeler Evenement.Afficher fenêtre choix
          message "Souhaitez-vous allumer la lumière ?"
          Titre "Bouton poussoir activé"
          Texte bouton 1 "Oui (Allumer)"
          Texte bouton 2 "Non (Eteindre)"
          annulable faux
  quand Evenement.Après choix
  Choix
  faire
    si obtenir Choix = "Oui (Allumer)"
    alors
      appeler Bluetooth.Envoyer1Octet nombre 1
    sinon
      appeler Bluetooth.Envoyer1Octet nombre 2
  
```

**Correction :**

Blocs

```

Arduino - générer le code
répéter indéfiniment
  si Bouton-poussoir BP_Entree appuyé alors
    Envoyer donnée Bluetooth 1
    attendre 1 secondes
  si Donnée Bluetooth reçue = 1 alors
    Activer Lumiere_Entree
  si Donnée Bluetooth reçue = 2 alors
    Désactiver Lumiere_Entree
  
```

Fichier mBlock : LU\_N3\_B4.sb2



CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES