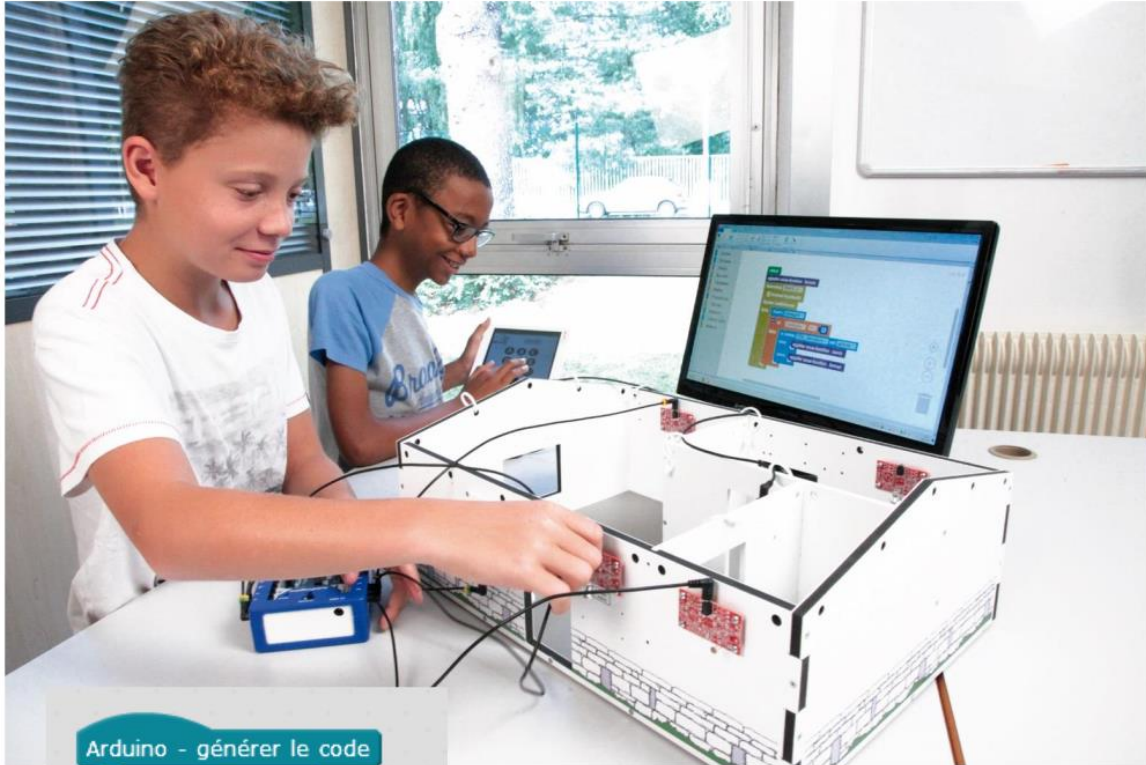


AutoLumi

Maquette programmable avec mBlock



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



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.

Prérequis

Pour la version de base :

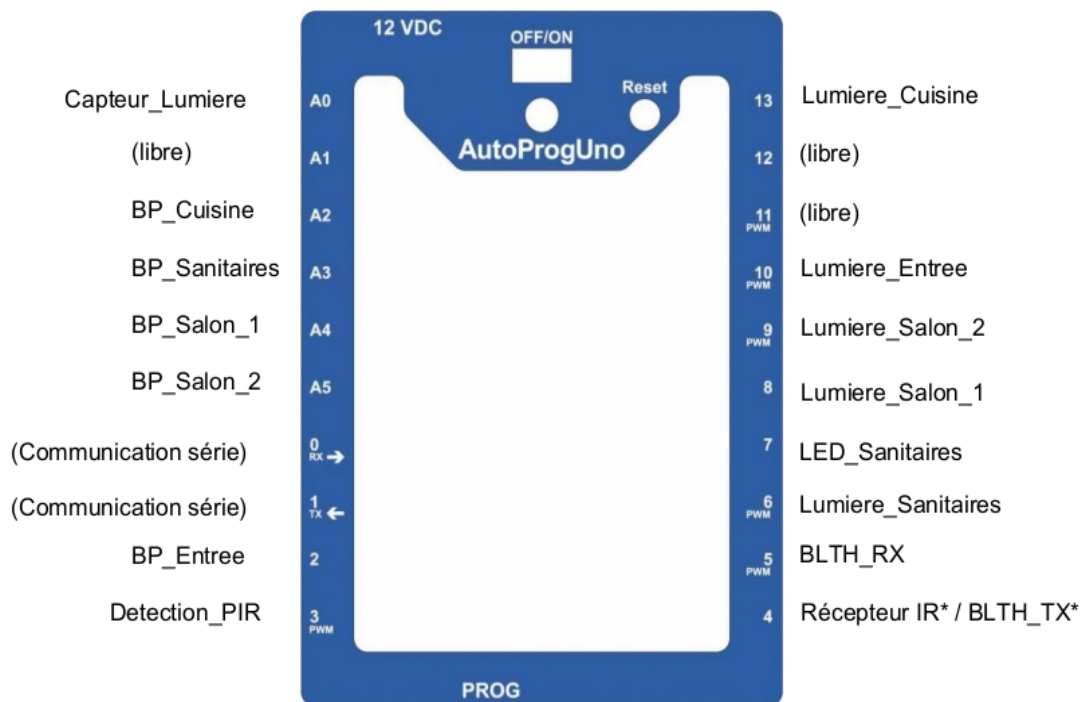
- Installer le logiciel **mBlock**.
- Installer l'extension **A4_Autolumi** (fichier zip) dans mBlock.
- **Maquette** AutoLumi (Réf. BE-ALUMI).
- **Câble de programmation** USB (Réf : CABL-IMPUSB).
- **Interface programmable** AutoProgUno (Réf. K-AP-UNO).
- 13 **cordons de liaison** jack compatibles 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.

Tableau d'affectation des entrées et sorties

AutoProgUno	AutoLumi	Nom mBlock
Modules capteurs pour entrées numériques		
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
Modules actionneurs pour sorties numériques		
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
Module de communication		
1	(communication avec ordinateur)	
2		
6		
Entrées / sorties libres (A pour les analogiques)		
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

Nom du fichier	Description	Objectif
Niveau 1 A Fichier modèle : 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.	
Niveau 1 B Fichier modèle : 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.	
Niveau 1 C Fichier modèle : 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
Niveau 1 D Fichier modèle : 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 - D

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

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.

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
Niveau 2 A Fichier modèle : 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.	
Niveau 2 B Fichier modèle : 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.	
Niveau 2 C Fichier modèle : 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

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

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.

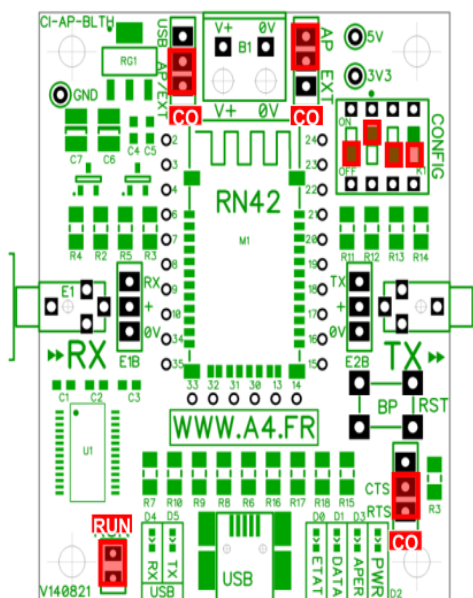
La documentation technique du module Bluetooth décrit en détail les fonctionnalités du module.

Elle est téléchargeable sur http://a4.fr/wiki/index.php/Module_Bluetooth_-_K-AP-MBLTH_/S-113020008.

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

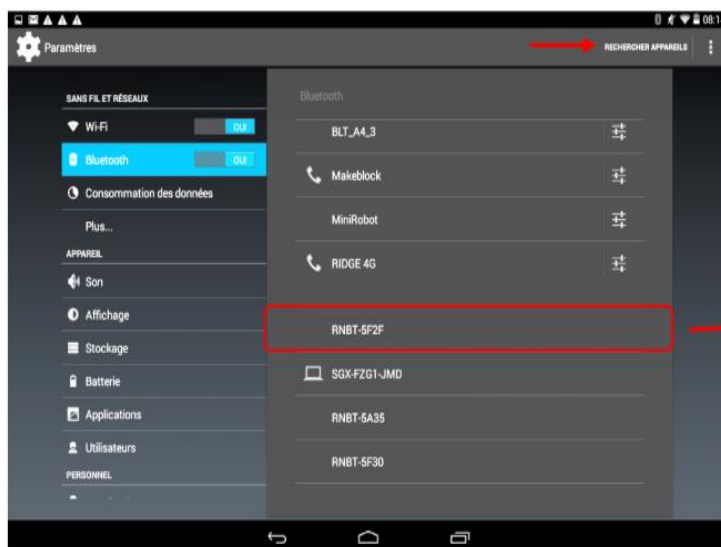
USB RX indique qu'il y a un flux de données sur la liaison USB du PC vers le module.

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.

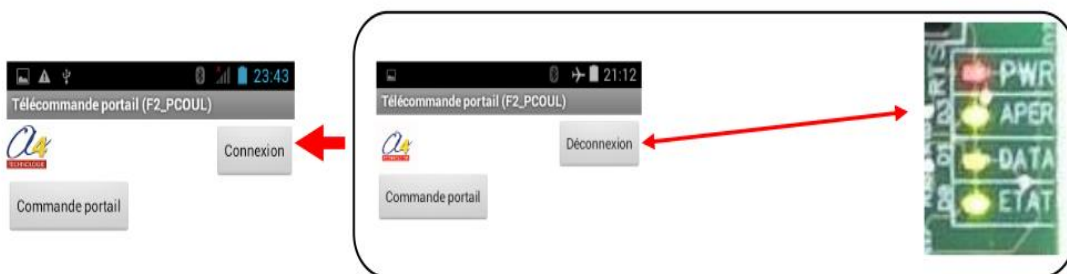


Tableau d'affectation des entrées et sorties

AutoProgUno	Monte-charge	Nom mBlock
Modules capteurs pour entrées numériques		
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
Modules actionneurs pour sorties numériques		
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
Module de communication		
1	(communication avec ordinateur)	
2		
6		
Entrées / sorties libres (A pour les analogiques)		
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		

