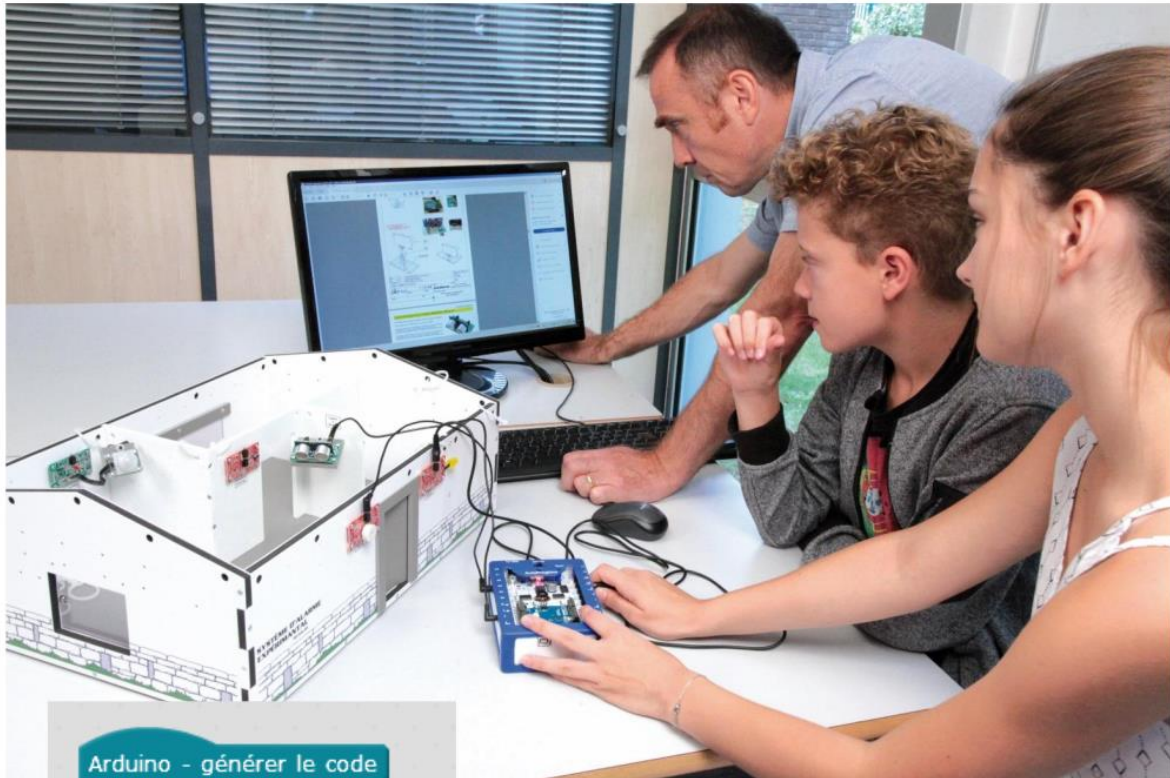


AutoAlarme

Maquette programmable avec mBlock



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.

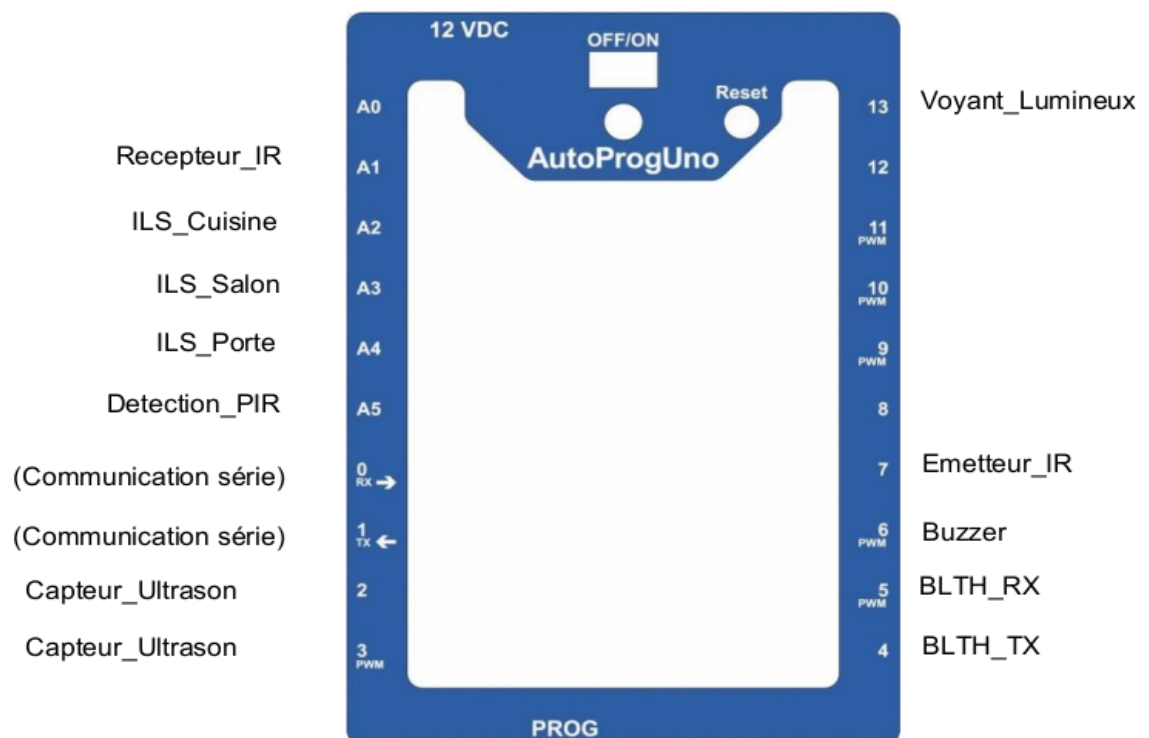
Pour l'option Bluetooth :

- **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
Modules capteurs pour entrées numériques		
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
Modules actionneurs pour sorties numériques		
9		
10		
11		
12		
13	Module voyant lumineux jaune	Voyant_Lumineux
Module de communication		
0	(communication avec ordinateur)	
1		
Entrées / sorties libres (A pour les analogiques)		
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

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

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

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
Niveau 2 A Fichier modèle : 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.	