

Barrière de parking

Maquette programmable avec mBlock



```
Arduino - générer le code
Oled sur Afficheur vider l'écran
mettre Comptage à 0
Oled sur Afficheur écrire regroupe Valeur = Comptage sur la ligne 1 colonne 0
répéter indéfiniment
si Bouton-poussoir Bouton_Poussoir_A appuyé alors
  ajouter à Comptage 1
  Oled sur Afficheur écrire regroupe Valeur = Comptage sur la ligne 1 colonne 0
  attendre jusqu'à non Bouton-poussoir Bouton_Poussoir_A appuyé
```

Introduction

Barrière de parking

La maquette Barrière de Parking (BE-APARK) est une reproduction homothétique d'une barrière de parking automatisée réelle : bouton d'activation, capteurs fin de course, barrière optique, clignotant de sécurité, moteur. Programmable et piloté par les systèmes AutoProgX2 ou AutoProgUno, il 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 barrière :

- 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 : afficheur, Bluetooth et capteur RFID.

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_Barriere** (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 : afficheur, Bluetooth et capteur RFID.

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 de correction qui propose un exemple de programme réalisé sous mBlock (extension .sb2).

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 :

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

Exemple : VR_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_Barriere** (fichier zip) dans mBlock.
- **Maquette** Barrière de parking (Réf. BE-APARK).
- **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.

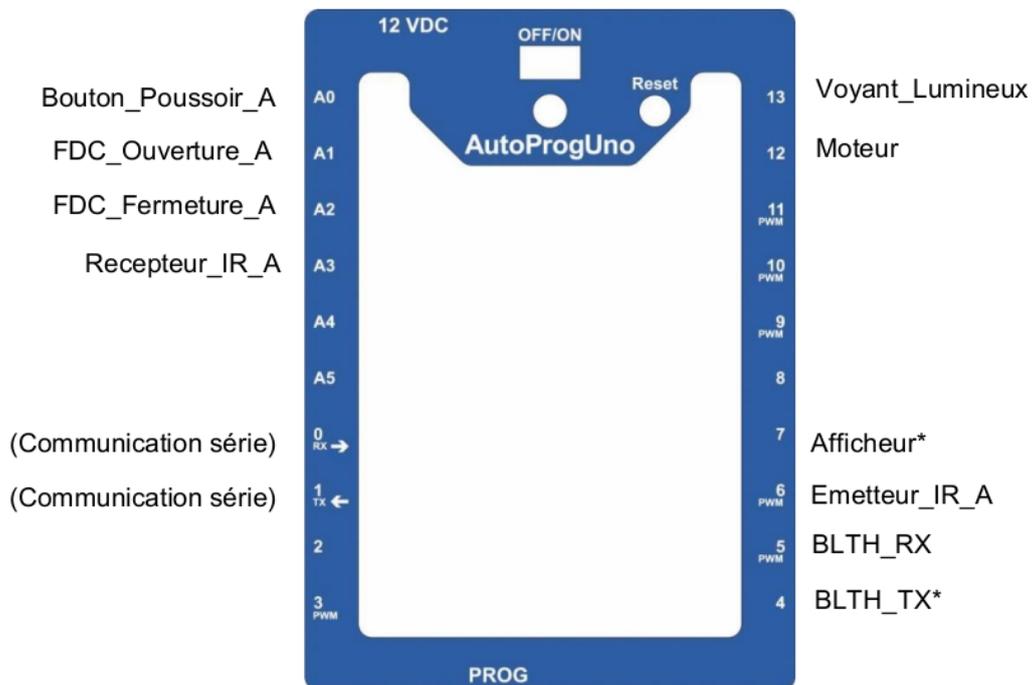
Pour l'option Bluetooth :

- **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	Maquette	Nom mBlock
MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
2		
3		
4	Module Bluetooth (option)	BLTH_TX*
5	Module Bluetooth (option)	BLTH_RX
MODULES ACTIONNEURS POUR SORTIES NUMÉRIQUES		
6	Emetteur infrarouge A	Emetteur_IR_A
7	Module afficheur OLED	Afficheur*
8		
9		
10		
11		
12	Module moteur	Moteur
13	Module lumineux jaune	Voyant_Lumineux
MODULE DE COMMUNICATION		
0	(communication avec ordinateur)	
1		
ENTRÉES / SORTIES LIBRES (A pour les analogiques)		
A0	Bouton poussoir A / Capteur RFID	Bouton_Poussoir_A
A1	Fin de course ouverture A	FDC_Ouverture_A
A2	Fin de course fermeture A	FDC_Fermeture_A
A3	Recepteur infrarouge A	Recepteur_IR_A
A4		
A5		



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		
BP_N1_A1.sb2	Allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordé : -Allumage/extinction du voyant lumineux Notions de programmation abordées : -séquence d'instructions -temps d'attente -boucle infinie
BP_N1_A2.sb2	Répéter cette même action deux fois.	
BP_N1_A3.sb2	Répéter cette action à l'infini.	
Niveau 1 B		
BP_N1_B1.sb2	Activer le moteur pendant 5 secondes.	Fonctionnalité matérielle abordé : -Gestion du moteur -Utilisation de Bouton-poussoir Notions de programmation abordées : -boucle qui dépend d'une entrée
BP_N1_B2.sb2	Activer le moteur jusqu'à l'appui d'un bouton poussoir.	
Niveau 1 C		
BP_N1_C1.sb2	Allumer le voyant lumineux à l'appui du BP.	Fonctionnalité matérielle abordé : -Gestion des modules infra-rouge -Utilisation de Bouton-poussoir Notions de programmation abordées : -Le test d'une entrée (si/sinon)
BP_N1_C2.sb2	Activer le voyant lumineux lorsque la barrière infrarouge est franchie.	
BP_N1_C3.sb2	Contrôler le moteur avec le bouton poussoir et allumer le voyant sur le franchissement de la barrière infrarouge.	
Niveau 1 D		
BP_N1_D1.sb2	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).	Notions de programmation abordées : -Définition de variable -Incrémentation de variable -Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage
BP_N1_D2.sb2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	
BP_N1_D3.sb2	Incrémenter une variable à l'appui d'un bouton poussoir, la décrémenter à l'appui de l'autre bouton poussoir.	
BP_N1_D4.sb2	Incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.	

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		
BP_N2_A1.sb2	Allumer le moteur et éteindre avec 2 secondes d'attente sur chaque capteur fin de course. Le voyant doit d'allumer lorsque le moteur tourne. Utiliser des sous-fonctions.	Notions de programmation abordées : -Utilisation des fins de course
BP_N2_A2.sb2	Ouvrir ou fermer la barrière sur l'appui du bouton poussoir.	
BP_N2_A3.sb2	Ouvrir la barrière sur l'appui d'un bouton poussoir ou sur le franchissement de la barrière infrarouge, puis fermer la barrière après 5 secondes.	