Barrière de parking Maquette programmable avec PICAXE Editor / Blockly pour PICAXE





Ressources disponibles pour le projet

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

Barrière de parking

- Fichiers **3D** (SolidWorks, Edrawings et Parasolid) de la maquette et de ses options.
- Dossier technique pour la mise en œuvre de la maquette.

Logiciels Picaxe Editor 6 / Blockly et App Inventor

- Procédure d'installation du driver pour le câble de programmation.
- Manuel d'utilisation Picaxe Editor 6.
- Notice d'utilisation App Inventor 2.

Activités / Programmation

C RESSOURCES

• Fichiers modèles et fichiers de correction des programmes pour Picaxe EDITOR 6 (organigrammes et blocs) 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 <u>http://creativecommons.fr/</u>

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 <u>www.a4.fr</u>



SOMMAIRE

Introduction	5
Barrière de parking	5
Les environnements de programmation graphique	5
Le dossier	5
Les fiches exercices	6
Prérequis	6
Caractéristiques techniques	6
Environnement de programmation graphique	7
Personnalisation des entrées/ sorties	7
Tableau d'affectation des entrées et sorties de Barrière de parking	8
Personnalisation du jeu d'instructions	9
Procédure de chargement d'un programme	9
Mode simulation	10
Programmation version de base niveau 1	11
Niveau 1 - A	12
Exercice niveau 1 - A 1 · Activer / désactiver un témoin lumineux	12
Exercice niveau 1 - A 2: Rénéter une action deux fois	13
Exercice niveau 1 - A 3 : Répéter une séquence indéfiniment	14
Niveau 1 - B	15
Exercice niveau 1 - B.1 : Maitriser la rotation du moteur	15
Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que	16
Niveau 1 - C	17
Exercice niveau 1 - C.1 : Instruction conditionnelle et bouton-poussoir	17
Exercice niveau 1 - C.2 : Instruction conditionnelle et barrière infrarouge	18
Exercice niveau 1 - C.3 : Contrôle moteur ET voyant lumineux	19
Programmation version de base niveau 2	24
Nivoau 2 - A	25
Exercise niveru 2 - A 1 : Ouverture/formature ontro fine de courses	25
Exercice niveau 2 - A.1. Ouverture/remeture entre fins de courses	20
Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture avec heuten et herrière infrareure	20
Exercice niveau 2 - A.S. Controle de l'ouverture et de la termeture avec bouton et bamere innarouge	21
Programmation niveau 3	28
Option : Module télécommande infrarouge	34
Télécommande RAX-TVR10	34
Télécommande TELEC-IR-UNIV	36
Exercice niveau 3 - B.1 : Contrôle d'ouverture/fermeture avec la télécommande IR	38
Exercice niveau 3 - B.2 : Ouvrir la barrière avec un code	39
Option : Module Bluetooth	40
Exercice niveau 3 - C.1 : Ouvrir/fermer avec application Bluetooth	43
Exercice niveau 3 - C.2 : Contrôle du portail par Smartphone	44
Exercice niveau 3 - C.3 : Envoyer des données vers un Smartphone	45
Exercice niveau 3 - C.4 : Envoyer et recevoir des données provenant d'un Smartphone	46



Option : Module capteur de courant	47
Exercice niveau 3 – D.1 : Utilisation du capteur de courant	51
Exercice niveau 3 – D.2 : Capteur de courant et variable	
Exercice niveau 3 – D.3 : Sécurité et réactivation	
Option : Module Capteur RFID	54
Exercice niveau 3 – E.1 : Utilisation du module capteur RFID	
Exercice niveau 3 – E.2 : Utilisation du module capteur RFID (2)	
Programmation niveau 4 : barrière double	58
Exercice niveau 4 – A.1 Mise en place de deux barrières	
Exercice niveau 4 – A.2 : Afficher le nombre de places disponibles	60





Barrière de parking

La maguette 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 : télécommande infrarouge, module Bluetooth, module capteur de courant et module capteur RFID.

Les environnements de programmation graphique

Tous les programmes correspondant aux activités menées autour de la maquette AutoAlarme ont été réalisés sous Editor 6 / Blockly. En effet, ce logiciel de programmation graphique présente plusieurs avantages :

- Gratuit
- Blocs et organigrammes (proche algorigrammes).
- Personnalisation des noms des entrées/sorties.
- Personnalisation du jeu d'instructions.
- Mode de simulation visuelle à l'écran pour mettre au point et débugger les programmes.

Vous pouvez aussi utiliser Blockly for Picaxe : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

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 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 Bluetooth, module capteur de courant et module 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 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 Editor 6 en blocs (extension .xml) et en organigrammes pour le niveau 1 uniquement (extension .plf).

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 organigrammes ou 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 : BP_N3_B2.xml

Correspond au niveau 3 avec le jeu d'instructions B, adapté aux objectifs « avancés » de ce niveau.

Prérequis

Pour la version de base :

- Installer le logiciel Editor 6 / Blockly : <u>http://www.picaxe.com/Software</u>
- **Maquette** Barrière de parking (Réf. BE-APARK).
- Câble de programmation Picaxe USB (Réf : CABLE-USBPICAXE).
- Interface programmable AutoProgX2 (Réf. K-APV2).
- des 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.

Caractéristiques techniques

Le guide de montage ainsi que les caractéristiques techniques des composants sont détaillés dans le dossier technique disponible sur le wiki.



Environnement de programmation graphique

Tous les programmes correspondant aux activités menées ont été réalisés sous **Editor 6**. En effet, ce logiciel de programmation graphique présente plusieurs avantages :

- Gratuit
- Blocs et organigrammes (proche algorigrammes).
- Personnalisation des noms des entrées/sorties.
- Personnalisation du jeu d'instructions.
- Mode de simulation visuelle à l'écran pour mettre au point et débugger les programmes.

Note : vous pouvez aussi utiliser **Blockly for Picaxe** : environnement de programmation par blocs simplifié (nombre de menus limité et personnalisation des entrées/sorties non disponibles).

Personnalisation des entrées/ sorties

Nous vous proposons le fichier **BP_BASE.xml** dans lequel les noms des entrées/sorties ont été personnalisés pour une utilisation avec la Barrière de parking. Tous les programmes et activités proposés dans ce document se basent sur cette liste. Celle-ci reste modifiable à tout moment.

A partir de Picaxe Editor 6, dans l'explorateur d'espace de travail cliquer sur Table d'entrées / sorties.



Une fenêtre apparaît à partir de laquelle vous pouvez modifier les noms de toutes les entrées et sorties dans la zone « Mon étiquette ».

Table d'ent	rées / sorties	
B.0	Moteur_A	•
B.1	Emetteur_IR_A	
B.2	Afficheur_A	
В.3	Voyant_Lumineux	
B.4	B.4	
B.5	B.5	
B.6	B.6	
		OK Annuler

Valider en cliquant sur OK.



Tableau d'affectation des entrées et sorties de Barrière de parking

ES	Module de communication pour entrées / sorties numériques	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX
6	Communication Bluetooth réception de données	C.6	BLTH_RX
EN	Modules capteurs pour entrées numériques		
5			
4			
3	Récepteur infrarouge	C.3	Recepteur_IR
2	Capteur de fin de course fermeture de la barrière	C.2	FDC_Fermeture_A
1	Capteur de fin de course ouverture de la barrière	C.1	FDC_Ouverture_A
0	Bouton-poussoir intérieur	C.0	Bouton_Poussoir_A
EA	Modules capteurs pour entrées analogiques		
3	(libre)	A.3	
2	(libre)	A.2	
1	(libre)	A.1	
0	(libre)	A.0	
SN	Modules actionneurs sorties numériques		
7	Connecté au moteur	B.7	Moteur_A
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	Module afficheur OLED (option)	B.2	Afficheur_A
1	Emetteur infrarouge	B.2	Emetteur_IR_A
0	Module voyant lumineux	B.0	Voyant_Lumineux





Personnalisation du jeu d'instructions

Vous pouvez personnaliser l'affichage du jeu d'instructions pour en limiter la quantité afin de faciliter la l'analyse et la correction des erreurs. Faire un clic droit sur la zone des blocs puis cliquer sur **Editer la boite d'outil**.



Une fenêtre s'ouvre à partir de laquelle vous pouvez sélectionner ou désélectionner les instructions de votre choix. Vous pouvez renommer le jeu d'instructions dans la zone « **Extension** ».



Valider en cliquant sur OK.

Procédure de chargement d'un programme

Commencer par relier le Loupiot à l'ordinateur avec le câble de programmation USB et le mettre sous tension. A partir du Picaxe Editor 6, ouvrir un programme.

	,	1 0	
📶 🔚 🕿 🦘 🔹 🔻			PICAXE Editor 6.0.9.3
Fichier Principal Simuler PICAXE			
Nouveau Organigramme Fichier	rir Enregistrer	Couper Coller Coller Couper Coller Pressepaires Impression rapide Coller Coller Coller Impression rapide Impression rapide Coller Impression rapide Impression rapide	Verifier Syrtaxe Tele Programmer le PICAXE (F5)
Configuration 👻 🕂	AL_N1_B5.xml ×		Télécharger le programme en cours dans le PICAXE
Configuration Type de PICAXE PICAXE-28X2 Afficher plus de VicAXE Verfiele types de PICAXE Verfiele types de PICAXE Table d'entrées / sotiles Pot de communication Configurer et tester Configurer et tester Gestionnaire les pots Configurer Simulation PICAXE-28X2 V Options de simulation	Blocs PICAX Sorties Entrées Délais Boucles Variables Maths Procédures Táches Moteurs	XE BASIC XML début répéter indéfiniment faire si entrée [LS_Porte = est désactivée = faire basculer Voyant_Lumineux = mélodie tune B.0, 3, (\$C0,\$44) sinon sortie Voyant_Lumineux = désactivée	
	Liaison série		

A partir du menu **Principal** ou du menu **PICAXE**, cliquer sur le bouton **Exécuter**. Vous pouvez également utiliser la touche **F5** de votre clavier.

Note : un programme téléchargé écrase le précédent.



Mode simulation

La simulation sur Picaxe EDITOR 6 permet de tester un programme avant de le téléverser dans la maquette. Pour lancer et contrôler une simulation, utiliser les boutons **Exécuter / Pause / Pas à pas / Arrêt** à partir du menu **Simuler**.

Fichier	Principal	Simuler	PICAXE								
Défaut	븛 Connecter 🔸	• 🔶 S er 👍 P 💥 E	uivant récédent ffacer tout	ABC Vérifier	Son activé Volume+ Volume-	Exécuter	D Pause	Pas à pas	O Arrêt	G Réinitialiser]
Panne	Simulation en dire	ect Po	ints d'arrêt	Syntaxe			Simul	er			Ī

La simulation surligne les blocs dans l'espace de travail pour vous montrer où en est le programme.



Programmation version de base niveau 1

Objectifs:

- Découvrir et maîtriser le matériel avec des exemples très simples pour débuter en programmation.
 - Appréhender les différentes fonctionnalités du matériel.

Ce niveau permet de découvrir toutes les fonctionnalités de base 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 « BP_N1_A1.xml », charger le programme modèle « BP_BASE.xml ».

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			
Niveau 1 A Fichier modèle	: BP_N1_A.xml				
BP_N1_A1	allumer le voyant lumineux pendant 3 secondes puis l'éteindre.	Fonctionnalité matérielle abordée : -Allumage/extinction du voyant			
BP_N1_A2	Répéter cette même action deux fois.	Notions de programmation abordées :			
BP_N1_A3	Répéter cette action à l'infini.	-séquence d'instructions -temps d'attente -boucle infinie			
Niveau 1 B Fichier modèle	: BP_N1_B.xml				
BP_N1_B1	Activer le moteur pendant 5 secondes.	Fonctionnalité matérielle abordé : -Gestion d'un moteur. -Utilisation d'un bouton-poussoir.			
BP_N1_B2	Activer le moteur jusqu'à l'appui d'un bouton-poussoir.	Notions de programmation abordées : -boucle qui dépend d'une entrée.			
Niveau 1 C Fichier modèle	: BP_N1_C.xml				
BP_N1_C1	Allumer le voyant lumineux à l'appui du bouton- poussoir.	Fonctionnalité matérielle abordé : -Gestion des modules infrarouges.			
BP_N1_C2	Activer le voyant lumineux lorsque la barrière infrarouge est franchie.	-Utilisation d'un bouton-poussoir.			
BP_N1_C3	Contrôler le moteur avec le bouton-poussoir et allumer le voyant au franchissement de la barrière infrarouge.	Notions de programmation abordées : -Le test d'une entrée (si/sinon).			
Niveau 1 D Fichier modèle : BP_N1_D.xml					
BP_N1_D1	Incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).				
BP_N1_D2	Incrémenter une variable au cours du temps faire un test sur celle-ci pour activer le voyant.	Notions de programmation abordées : -Définition de variable			
BP_N1_D3	Incrémenter une variable à l'appui d'un bouton- poussoir, la décrémenter à l'appui de l'autre bouton- poussoir.	-Test (si/sinon) de variable -Test (juste si) d'entrée -Débogage			
BP_N1_D4	Incrémenter une variable puis faire un test sur celle-ci pour contrôler l'état du voyant.				

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

Fichier modèle : BP_N1_A.xml

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 :

sortie A.O 🔹 activée 🔹	attendre pendant (500) ms

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

Fichier modèle : BP_N1_A.xml

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 :







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

Fichier modèle : BP_N1_A.xml

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

Notion abordée : la boucle infinie.

Instructions utilisées :

sortie A.O 🔹 activée 🔹	attendre pendant (500) ms
A répéter in faire	définiment

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 l'interface AutoProgX2.



Exercice niveau 1 - B.1 : Maitriser la rotation du moteur

Fichier modèle : BP_N1_B.xml

Objectif : activer le moteur pendant 5 secondes

Notion abordée : utilisation d'un moteur.

Instructions utilisées :







Exercice niveau 1 - B.2 : Utilisation d'une boucle tant que

Fichier modèle : BP_N1_B.xml

Objectif : Activer le moteur jusqu'à l'appui d'un bouton poussoir

Notion abordée : exécuter une boucle qui dépend de l'état d'une entrée.

Instructions utilisées :



Correction :



Remarque : Le programme ne peut sortir de la boucle qu'une fois le test sur le bouton-poussoir validé. Le test sur le bouton poussoir se fait qu'une seule fois en début de séquence, avant de commencer l'ouverture. Si un appui est effectué pendant la séquence, aucun effet n'aura lieu sur le programme. Afin de vérifier à tout moment le changement d'état d'une entrée dans une séquence, l'utilisation des interruptions est indispensable (voir ex sur interruption).

Si vous possédez un capteur RFID à la place du bouton-poussoir, veuillez-vous référencer dans un premier temps à l'exercice 1 de l'option Module Capteur RFID.



Exercice niveau 1 - C.1 : Instruction conditionnelle et boutonpoussoir

Objectif : allumer le voyant lumineux à l'appui du BP.

Notion abordée : utilisation des commandes conditionnelles (si/sinon).

Instructions utilisées :



Correction :



Remarque : les blocs de couleur bleu claires représente des commandes concernant l'utilisation des entrées.



Exercice niveau 1 - C.2 : Instruction conditionnelle et barrière infrarouge

Objectif : activer le voyant lumineux lorsque la barrière infrarouge est franchie.

Notions abordées : utilisation des commandes conditionnelles (si/sinon) et utilisation d'une barrière infrarouge.

Instructions utilisées :



Correction :



Remarque : l'entrée du récepteur IR est activée d'origine et se désactive lors de la réception du signal de l'émetteur IR.

Lorsque un obstacle franchi la barrière IR, le signal n'est plus transmis et l'entrée du récepteur IR devient active. Si le programme ne marche pas, mettre le cavalier MODE de l'émetteur IR sur B et le cavalier du code sur 127.



Exercice niveau 1 - C.3 : Contrôle moteur ET voyant lumineux

Objectif : contrôler le moteur avec le bouton-poussoir et allumer le voyant sur le franchissement de la barrière infrarouge.

Notion abordée : utilisation des commandes conditionnelles.

Instructions utilisées :





Exercice niveau 1 - D.1 : Utilisation des variables

Objectif : incrémenter une variable au cours du temps et observer sa valeur à l'aide du PC (débogage).

Notions abordées : la variable : définition et incrémentation, debug.

Instructions utilisées :



Correction :



Remarques : la commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.





Exercice niveau 1 - D.2 : Utiliser et tester une variable

Objectif : incrémenter une variable au cours du temps. Lorsque la variable est supérieure à 10, activer le voyant.

Notion abordée : boucle tant que dépendant d'une variable

Instructions utilisées :



Correction :



Remarque : cet exercice peut être utilisé comme un minuteur.



Exercice niveau 1 - D.3 : Contrôler la valeur d'une variable à l'aide des boutons poussoirs

Objectif : incrémenter une variable à l'appui d'un bouton-poussoir

Notions abordées : test sur entrées et incrémentation/décrémentation contrôlée d'une variable

Instruction utilisée :

▲ si entrée A.0 - est activée faire	Tixer varA ▼ à 0 10
	🔽 de C🚺
A répéter indéfiniment faire	décrémenter varA - de 1

Correction :

Organigramme	Blocs
	début fixer varA à 0 répéter indéfiniment faire si entrée Bouton_Poussoir_A • est activée • faire incrémenter varA • de 1 debug
Fichier organigramme PE6 : BP_N1_D3_Organigramme.plf	Fichier Blockly : BP_N1_D3.xml

Remarques : deux tests sont insérés l'un après l'autre. La vitesse d'exécution du programme donne l'impression que les commandes sont exécutées en même temps.

La commande « debug » est utilisée afin de retourner la valeur des variables à l'ordinateur. Il est donc indispensable de brancher le câble de programmation à l'ordinateur pour avoir un aperçu de leur valeur.

Exercice niveau 1 - D.4 : Tests /variables/ modules IR

Objectif : incrémenter une variable à chaque passage sur la barrière IR. Lorsque le compteur arrive à 10, activer le voyant lumineux 3 secondes et remettre la variable à zéro

Notion abordée : test dépendant d'une variable

Instructions utilisées :







Programmation version de base niveau 2

Objectifs :

- Utilisation concrète de la barrière de parking
- 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 mesures 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	: BP_BASE.xml	
BP_N2_A1.xml	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.	Notiona da programmation
BP_N2_A2.xml	Ouvrir ou fermer la barrière à l'appui du bouton- poussoir.	abordées : -Utilisation des fins de course
BP_N2_A3.xml	Ouvrir la barrière à l'appui d'un bouton-poussoir ou au franchissement de la barrière infrarouge, puis fermer la barrière après 5 secondes.	



Exercice niveau 2 - A.1 : Ouverture/fermeture entre fins de courses

Objectif : 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 abordées : utilisation des fins de course, procédures (sous-fonctions)



Correction :







Exercice niveau 2 - A.2 : Contrôle de l'ouverture et de la fermeture

Objectif : ouvrir ou fermer la barrière sur l'appui du bouton-poussoir.

Notions abordées : utilisation des fins de course, procédures (sous-fonctions).







Exercice niveau 2 - A.3 : Contrôle de l'ouverture et de la fermeture avec bouton et barrière infrarouge

Objectif : 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.

Notions abordées : utilisation des fins de course, procédures (sous-fonctions)





Programmation niveau 3

Objectif : Utiliser les modules plus complexes : pilotage à distance, contrôle par le courant...

Le niveau 3 n'intègre pas de nouvelles notions de programmation mais de nouveaux blocs permettant d'utiliser les modules options.

Nom du fichier	Description	Objectif			
Niveau 3 A – Afficheur OLED Fichier modèle : BP_BASE.xml					
BP_N3_A1.xml	Afficher un message sur les deux lignes de l'afficheur OLED.	Fonctionnalité matérielle abordée : Utilisation d'un afficheur OLED			
BP_N3_A2.xml	Afficher une variable qui changera à chaque incrémentation.	Notions de programmation			
BP_N3_A3.xml	Reprendre le programme BP_N2_A4 et afficher l'état de la barrière (en mouvement ou à l'arrêt) sur l'afficheur OLED.	Utilisation d'un block dédié à l'affichage sur OLED.			
Niveau 3 B – Té Fichier modèle	lécommande infrarouge : BP_BASE.xml				
BP_N3_B1.xml	Ouvrir et fermer la barrière à l'aide de la télécommande IR.	Fonctionnalité matérielle abordée : Utilisation de la télécommande IR Notions de programmation			
BP_N3_B2.xml	Ouvrir le portail en envoyant un code spécifique.	abordées : Utilisation d'un block dédié à la communication IR.			
Niveau 3 C – Mo Fichier modèle	Niveau 3 C – Module Bluetooth Fichier modèle : BP_BASE.xml				
BP_N3_C1.xml	Contrôler l'ouverture et la fermeture à l'aide de 2 boutons sur application Android.				
BP_N3_C2.xml	Contrôler l'ouverture et la fermeture à l'aide d'un seul bouton sur application Android.	Fonctionnalité matérielle abordée : - Module Bluetooth			
BP_N3_C3.xml	Jouer une sonnerie sur Smartphone à l'appui d'un bouton poussoir.	Notions de programmation abordées :			
BP_N3_C4	Lors de l'appui sur un bouton de la maquette, envoyer une information au Smartphone qui devra renvoyer une information à la maquette.	- Liaison serie (nserin/nserout).			
Niveau 3 D – Ca Fichier modèle	pteur de courant : BP_BASE.xml				
BP_N3_D1.xml	Lire la valeur du capteur de courant.	Fonctionnalité matérielle abordée : Utilisation de la télécommande IR			
BP_N3_D2.xml	Arrêter le moteur lors d'un blocage de la barrière.	Notions de programmation abordées :			
BP_N3_D3.xml	Implémenter le capteur de courant dans un vrai programme automatisé.	Utilisation d'un block dédié à la communication IR.			
Niveau 3 E – Ca Fichier modèle	pteur RFID : BP_BASE.xml				
BP_N3_E1.xml	Basculer le voyant lumineux lors de la détection d'une carte sur le capteur RFID.	Fonctionnalité matérielle abordé : Utilisation d'un capteur magnétique			
BP_N3_E2.xml	Ouvrir puis fermer la barrière lors du déclenchement du capteur RFID.	RFID.			



Option : Module afficheur OLED



Ces afficheurs permettent à son utilisateur de rentrer un texte dans un programme qui sera chargé d'afficher ce texte.

L'afficheur utilisé pour cette maquette possède 2 lignes sur lesquelles il est possible d'insérer au maximum 16 caractères (espaces compris).

Trois blocs sont déjà disponibles sur PICAXE Editor afin de simplifier l'utilisation de ce module :

envoyer sur LCD effacer 🔹 en Afficheur A 🛪 envoyer sur LCD (positionner le curseur à la ligne 1 🔹 en Afficheur A 🔹 afficher sur LCD Afficheur A « message)

Il est nécessaire d'effacer le texte déjà présent sur l'afficheur lorsqu'on veut y mettre un autre message. En effet, il garde en mémoire chaque caractère jusqu'à ce que tout soit effacé. Ce qui signifie que si nous écrivons sur 5 caractères là où il y en avait déjà 10 avant la programmation, nous verrons nos 5 caractères + les 5 derniers déjà présents auparavant, ou alors le message ne s'affichera pas au début de l'afficheur.

La commande positionner le curseur à la ligne 1 signifie simplement que nous écrivons sur la première ligne. Nous pouvons également choisir la deuxième ligne sur cet afficheur.

Ensuite il vous suffit seulement d'afficher le message souhaité, tant qu'il ne dépasse pas 16 caractères.

Pour plus d'informations, ou pour une méthode de mise en service, référez-vous à la documentation Afficheurs PICAXE OLED téléchargeable sur <u>www.a4.fr</u>

Tableau d'affectation des entrées et sorties

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	(libre)	C.5	
4	(libre)	C.4	
3	Capteur récepteur infrarouge	C.3	Recepteur_IR_A
2	Capteur de fin de course fermeture de la barrière	C.2	FDC_Fermeture_A
1	Capteur de fin de course ouverture de la barrière	C.1	FDC_Ouverture_A
0	Bouton poussoir intérieur	C.0	Bouton_Poussoir_A
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	(libre)	A.1	
0	(libre)	A.0	
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	(libre)	B.7	
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	Module voyant lumineux	B.3	Voyant_Lumineux
2	Module afficheur OLED (Option)	B.2	Afficheur_A*
1	Emetteur infrarouge	B.1	Emetteur_IR_A
0	Connecté au moteur	B.0	Moteur_A







Exercice niveau 3 - A.1 : Afficher un message

Objectif : Afficher un message utilisant les deux lignes sur l'afficheur OLED







Exercice niveau 3 - A.2 : Afficher une variable

Objectif : Afficher un message sur l'afficheur avec une variable. La variable augmente à chaque appui sur le boutonpoussoir.





Exercice niveau 3 - A.3 : Afficher un message en fonction de l'état de la barrière

Objectif : Reprendre l'exercice BP_N2_A3

Afficher un message dépendant de l'état de la barrière. Si elle est ouverte, il doit afficher « ouverte », si elle est fermée, il doit afficher « fermer », si elle se ferme, il doit afficher « fermeture » et « ouverture » lorsqu'elle s'ouvre.





Option : Module télécommande infrarouge

Télécommande RAX-TVR10

La télécommande universelle infrarouge associée à un capteur infrarouge approprié permet de piloter à distance une carte PICAXE.

Afin d'assurer la compatibilité de fonctionnement avec le système PICAXE il est nécessaire de l'initialiser avec le mode de fonctionnement au standard « Sony TV ».



Attention : L'émetteur infrarouge doit toujours être désactivé lors de l'utilisation de la télécommande infrarouge.

Procédure de mise en service pour PICAXE

1. Insérer 2 piles AAA dans le logement au dos de la télécommande.

- 2. Appuyer simultanément sur S et B.
- = La LED s'allume.

3. Taper le code 0 - 1 - 3 = La LED clignote brièvement à chaque appui des touches « 0 » et « 1 » puis s'éteint après l'appui sur la touche « 3 ».

4. Appuyer sur le bouton de mise en service.= La télécommande est opérationnelle.

Les touches suivantes risquent de déprogrammer :



Δ B C ▷ < F G S

Conseil : Si la télécommande ne fonctionne plus, appuyer sur **B** pour revenir à la configuration compatible PICAXE

<u>Remarque :</u> Le guide d'utilisation complet de la télécommande est disponible ici : <u>http://www.a4telechargement.fr/RAX-TVR010/RAX-</u> TRV10_Telecommande_InfraRouge.pdf





Tester la télécommande

Charger les programmes de test de la télécommande : « test_infra_bloc.xml » ou « test_infra_org.plf ». Respecter le plan de câblage vu précédemment dans le dossier.



Tableau de correspondance des touches

Diriger la télécommande vers le récepteur infrarouge et vérifier dans la partie « Variables » de Picaxe Editor que les données reçues sont correctes.

Ci-dessous, le tableau des valeurs renvoyées par les différents boutons de la télécommande :

Touche	1	2	3	4	5	6	7	8	9	0	
Code émis	0	1	2	3	4	5	6	7	8	9	21
		₿		\gg		^	+	×	-	+	
Code émis	16	17	19	18	96	54	37	20	98	11	

Télécommande TELEC-IR-UNIV

Procédure de mise en service pour PICAXE

- 1. Insérer 2 piles AAA dans le logement au dos de la télécommande.
- Appuyer simultanément sur les boutons Set et TV. Le bouton Power s'allume.
- Taper le code 0-7-7. Le bouton **Power** clignote brièvement à chaque appui, puis s'éteint.
- 4. Appuyer sur le bouton **Power**. La télécommande est opérationnelle.

ATTENTION !

La touche **DVB** risque de changer le mode. Appuyer sur **TV** pour revenir dans le bon mode.



<u>Conseil</u> : Si la télécommande ne fonctionne plus, appuyer sur **TV** pour revenir à la configuration compatible PICAXE.



Tester la télécommande

Charger les programmes de test de la télécommande : « test_infra_bloc.xml » ou « test_infra_org.plf ». Respecter le plan de câblage vu précédemment dans le dossier.







Tableau de correspondance des touches

L'appui sur une touche provoque l'émission d'un signal infrarouge qui véhicule un code correspondant à la touche. Par défaut : appui sur la touche 1 = envoi du code 0.

Pour simplifier l'utilisation de la télécommande infrarouge, il est possible d'activer la compatibilité entre le N° des touches et le code envoyé. Dans ce cas, appui sur la touche 1 = envoi du code 1.

Touche			2	3	4
Code émis standard	9	0	1	2	3
Compatibilité activée	10	1	2	3	4
Touche	5	6	7	8	9
Code émis standard	4	5	6	7	8
Compatibilité activée	5	6	7	8	9
Touche		ÂB	SCAN	AV	0
Code émis standard	12	37	16	37	56
Compatibilité activée	13	38	17	38	57
Touche				Power	
Code émis standard	63	74	29	21	76
Compatibilité activée	64	75	30	22	77
Touche	0			VOL+	- VOL
Code émis standard	77	78	79	18	19
Compatibilité activée	78	79	80	19	20
Touche		CH	CH		
Code émis standard	20	16	17		
Compatibilité activée	21	17	18		

Dans les exercices suivants, vous possédez déjà un récepteur infrarouge livré avec la maquette Barrière Parking simple.



Exercice niveau 3 - B.1 : Contrôle d'ouverture/fermeture avec la télécommande IR

Objectif : ouvrir et fermer la barrière à l'aide de la télécommande IR.

Notion abordée : gestion d'une liaison infra-rouge : télécommande/AutoProg à l'aide du bloc prévu à cet effet.

Correction :

Blocs	
début répéter indéfiniment fair debug Ile valeur infrarouge (Recepteur [R.A. et stocker dans consigne * Consigne * 10 Temps écoulé * é i consigne * 17 faire si entrée FDC_Ouverture A* est désactivée * faire si entrée FDC_Fermeture A* est désactivée * faire si entrée FDC_Fermeture A* est désactivée * faire appeler sous-fonction fermer faire appeler sous-fonction fermer faire appeler sous-fonction fermer	sous-fonction OLIVIII sortie Moteur_A a activée * répéter basculer Voyant_Lumineux * attendre pendant 300 ms jusqu'à entrée FDC_Ouverture_A est activée * sortie Moteur_A désactivée * sortie Voyant_Lumineux * désactivée * sortie Moteur_A a désactivée * sortie Moteur_A a désactivée * sortie Moteur_A a activée * épéter basculer Voyant_Lumineux * désactivée * sortie Moteur_A a activée * épéter basculer Voyant_Lumineux * intendre pendant 300 ms jusqu'à entrée FDC_Fermeture_A * est activée * sortie Moteur_A a désactivée * sortie Voyant_Lumineux * désactivée *
Tichiel Diockly . Di _N3	<u></u>

Remarque : Cocher la case +1 permet d'avoir une concordance entre la touche pressée et la consigne reçue.





Exercice niveau 3 - B.2 : Ouvrir la barrière avec un code

Objectif : ouvrir le portail dès la réception d'un code spécifique envoyée par la télécommande. Le fermer 5 secondes plus tard.

Notion abordée : gestion d'une liaison infra-rouge : télécommande/AutoProg.

	Blocs		
début fixer varG à â fixer appeler sous-fonction Recevoir_donnee S consigne = varG fixer appeler sous-fonction Recevoir_donnee S consigne = varG fixer appeler sous-fonction Recevoir_donnee S consigne = varG fixer appeler sous-fonction ouvrir attendre pendant 6000 ms appeler sous-fonction fermer	<pre>sous-onction @uvrif sortie Moteur_A activée intendre pendant 300 ms jusqu'à entrée FDC_Ouverture_A : est activée sortie Moteur_A : désactivée sortie Voyant_Lumineux : désactivée sortie Moteur_A : activée isortie Moteur_A : activée sortie Moteur_A : activée sortie Moteur_A : activée sortie Moteur_A : désactivée sortie Voyant_Lumineux : désactivée sortie Voyant_Lumineux : désactivée sortie Voyant_Lumineux : désactivée sortie : jusqu'à consigne isortie Voyant_Lumineux : activée sortie Voyant_Lumineux : activée attendre pendant : 500 ms sortie Voyant_Lumineux : désactivée</pre>		
Fichier Block	kly : BP_N3_B2.xml		



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 <u>http://a4.fr/wiki/index.php/Module_Bluetooth_-_K-AP-MBLTH_/_S-113020008</u>. 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 étre allumée pour allimenter le module). Le nom de votre module s'appelle : RNBT + les 4 derniers chiffres de l'adresse mac du module notés sur le composant. Selectionnez le et un message proposant de vous connecter à lui deARait s'afficher.



Une fois cette étape passée vous pourrez vous connecter au module à partir du programme AppInventor 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 en Bluetooth

ES	Modules de communication pour entrées / sorties numériques	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX
6	Communication Bluetooth réception de données	C.6	BLTH_RX
EN	Modules capteurs pour entrées numériques		
5	(libre)	C.5	
4	(libre)	C.4	
3	Capteur récepteur infrarouge	C.3	Recepteur_IR_A
2	Capteur de fin de course fermeture de la barrière	C.2	FDC_Fermeture_A
1	Capteur de fin de course ouverture de la barrière	C.1	FDC_Ouverture_A
0	Bouton poussoir intérieur	C.0	Bouton_Poussoir_A
EA	Modules capteurs pour entrées analogiques		
3	(libre)	A.3	
2	(libre)	A.2	
1	(libre)	A.1	
0	(libre)	A.0	
SN	Modules actionneurs sorties numériques		
7	Connecté au moteur	B.0	Moteur_A
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	Module afficheur OLED (Option)	B.2	Afficheur_A
1	Emetteur infrarouge	B.1	Emetteur_IR_A
0	Module voyant lumineux	B.3	Voyant_Lumineux

Câblage du module Bluetooth (K-AP-MBLTH)





Exercice niveau 3 - C.1 : Ouvrir/fermer avec application Bluetooth

Objectif : contrôler l'ouverture et la fermeture du portail à l'aide de 2 boutons présent sur l'application Android. **Notion abordée :** réception de données Bluetooth envoyées par un Smartphone.









Exercice niveau 3 - C.2 : Contrôle du portail par Smartphone

Objectif : ouvrir et fermer le portail à partir d'un seul bouton disponible sur l'application Android.

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



Correction :







Exercice niveau 3 - C.3 : Envoyer des données vers un Smartphone

Objectif : jouer une sonnerie sur le Smartphone à partir de l'appui d'un BP du portail.

Notion abordée : envoyer des informations à un Smartphone par Bluetooth.







Exercice niveau 3 - C.4 : Envoyer et recevoir des données provenant d'un Smartphone

Objectif : gérer la sonnette ainsi que le contrôle du portail à distance à l'aide de l'application Android.

Notion abordée : envoyer et recevoir des informations à l'aide du module Bluetooth à une application. **Correction :**







Option : Module capteur de courant

Ce module permet de sécuriser le fonctionnement d'un automatisme animé par un moteur à courant continu. En effet, lorsqu'un événement anormal se produit (ex : blocage de la barrière), la consommation de courant du moteur augmente. La détection de la surintensité au-delà d'un seuil permet de déclencher l'arrêt du moteur et ainsi mettre le système en sécurité.

Le module capteur de courant peut fonctionner selon deux modes.

Mode analogique :

Le cavalier S1 est en position ANA. La sortie MESURE renvoie une tension proportionnelle au courant circulant dans le module. Cette valeur est lue sur une entrée analogique du boîtier.

Mode numérique (tout ou rien) :

Le cavalier S1 est en position NUM. Le potentiomètre SEUIL permet de régler un seuil de courant au-delà duquel la sortie MESURE basculera. Si le courant circulant dans le module dépasse le seuil, la LED témoin et la sortie MESURE s'activent, sinon elles sont inactives.

Le potentiomètre INIT permet d'adapter la plage de mesure du module en fonction du contexte d'utilisation (consommation plus ou moins élevé de courant selon matériel utilisé).

La mesure de courant se fait au travers de de la connectique jack (AUTOPROG / ACTIONNEUR) ou bien directement via le bornier à vis ACTIONNEUR.

Mesure via connectique jack :

Le courant mesuré correspond au courant total qui circule dans le montage (courant consommé par l'interface AutoProg, les modules et le moteur).

Mesure via bornier à vis :

Le courant mesuré correspond à celui circulant dans l'élément connecté au bornier (moteur).

Dans la suite des exercices, le capteur de courant est utilisé en mode Analogique et la mesure est faite via la connectique jack. Il s'agit de détecter les surintensités dans le moteur ; on considère que la consommation de l'interface et des modules est négligeable par rapport à celle du moteur. Ce mode de d'utilisation facilite le câblage. L'interface AutoProg est alimenté par un bloc d'alimentation externe afin de garantir une tension constante de 5V dans l'ensemble du montage.

Mise en service du module







Réglage du potentiomètre d'initialisation :

Il s'agit de régler la tension de sortie du module de telle sorte qu'elle soit maximale lorsque le moteur et bloquée (consommation maximum) afin de travailler sur plage de mesure adaptée au moteur du portail.

En fonctionnement à vide (débrayé du portail) le moteur consomme jusqu'à 250 mA à vitesse moyenne. Si le moteur est bloqué, sa consommation monte au-delà de 500 mA.

Le réglage s'effectue en débrayant le moteur de la barrière (dérailler la barrière pour libérer le moteur de toute contrainte mécanique).

 Positionner l'interrupteur du module moteur sur OFF puis mettre le cavalier du moteur de Vext sur Vint pour que le moteur soit alimenté directement par l'interface.
 Positionner le cavalier S1 du module Capteur de courant sur ANA et le cavalier S2 sur la position 5V (Schéma)

Relier les embases jack A1 et A2 du module moteur respectivement sur les sorties B.6 et B.7 de l'interface AutoProg. Relier l'embase jack MESURE du module capteur de courant l'entrée analogique A.0 de l'interface AutoProg puis charger le programme de test **Test_Capteur_Courant.xml**

Ce programme permet de visualiser en direct à l'écran une valeur (varA) qui est proportionnelle à celle du courant consommé par le moteur.



- Positionner l'interrupteur du module moteur sur ON : le moteur doit tourner Ajuster le potentiomètre F-MOTA du module moteur à une position intermédiaire pour fixer la vitesse de rotation du moteur.
- 3) Ajuster le potentiomètre MESURE du module capteur de courant jusqu'à avoir une valeur de varA comprise entre 0 et 5.



4) Bloquer le moteur manuellement en retenant la roue dentée d'entrainement du portail. La valeur de varA doit augmenter de façon significative (augmentation brutale du courant consommé par le moteur).

Cette procédure permet de vérifier la bonne configuration du montage et de visualiser la valeur du courant circulant dans le moteur en vue de déterminer le seuil au-delà duquel on considère que celui-ci à consommation excessive (blocage).





Module pilotage moteurs (REF : K-AP-MMOT-M)



Module Capteur de courant (REF : K-AP-MAMP-M)



Tableau d'affectation des entrées et sorties

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	(libre)	C.5	
4	(libre)	C.4	
3	Capteur récepteur infrarouge	C.3	Recepteur_IR_A
2	Capteur de fin de course fermeture de la barrière	C.2	FDC_Fermeture_A
1	Capteur de fin de course ouverture de la barrière	C.1	FDC_Ouverture_A
0	Bouton poussoir intérieur	C.0	Bouton_Poussoir_A
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	(libre)	A.1	
0	Module capteur de courant	A.0	Capteur_courant
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	Connecté au moteur	B.0	Moteur_A
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	Module afficheur OLED (Option)	B.2	Afficheur_A*
1	Emetteur infrarouge	B.1	Emetteur_IR_A
0	Module voyant lumineux	B.3	Voyant_Lumineux





Exercice niveau 3 – D.1 : Utilisation du capteur de courant

Objectif : lire la valeur du capteur de courant en continu et déterminer le seuil (Valeur maximale de varA lorsqu'on bloque la barrière).

Notion(s) abordée(s) : lire une entrée analogique.

Correction :

Blocs
début
sortie Moteur_A activée
répéter indéfiniment
faire debug
lire valeur analogique en <u>Capteur_courant</u> et stocker dans <u>varA</u>
Fichier Blockly : BP N3 D1.xml

Remarque : Lorsqu'un effort est apporté au moteur, sa demande de courant augmente. Plus la résistance opposante au moteur sera grande, plus il va consommer de courant.

Lorsque d'un démarrage du moteur, il utilise pendant un bref instant plus de courant qu'en fonctionnement normal. Visualiser ce pic d'intensité sur varA à chaque changement de sens et considérer cette valeur comme le seuil bas, cela car le moteur ne doit pas s'arrêter au démarrage à cause de cette surintensité.

Voici une représentation de la surintensité de démarrage du moteur :



Celui-ci après le démarrage va se remettre à la valeur d'intensité normale de fonctionnement.

Il est préférable pour l'exercice suivant de mettre un seuil plus petit que le seuil maximum. Par exemple si lorsque vous bloquez la barrière vous trouvez un seuil max avec varA = 80, nous vous conseillons pour la suite de fixer ce seuil à 70 par exemple.



Exercice niveau 3 – D.2 : Capteur de courant et variable

Objectif : à partir de la valeur seuil établie dans l'exercice précédent, arrêter le moteur et faire clignoter le voyant lumineux lorsque le capteur de courant détecte un blocage.

Notion abordée : lire et interpréter une donnée d'une entrée analogique.

Correction :

Blocs				
début sortie Moteur A · activée · répéter indéfiniment faire debug lire valeur analogique en Capteur_courant · et stocker dans varA · si varA · > 30 faire appeler sous-fonction arret	sous-fonction arret sortie Moteur_A désactivée répéter indéfiniment faire basculer Voyant_Lumineux attendre pendant 500 ms			
Fichier Blockly : BP_N3_D2	2.xml			

Remarque : La valeur de seuil sélectionnée ici est 30, mais elle correspondra à celle que vous aurez établit dans votre programme.





Exercice niveau 3 – D.3 : Sécurité et réactivation

Objectif : Lorsque le bouton est activé ou qu'un obstacle franchit la barrière infrarouge, ouvrir la barrière puis la fermer après 5 secondes. En cas de blocage de la barrière (module capteur de courant), couper le moteur et le réactiver à l'appui du bouton poussoir

Utiliser une sous-fonction « programme » qu'on pourra rappeler en fin de boucle arrêt.



Cela afin de pouvoir rappeler le programme depuis la sous-fonction d'arrêt, et donc de relancer le programme sans avoir à relancer l'interface AutoProg.

Notion abordée : lire et interpréter une donnée d'une entrée analogique.

Correction :





Option : Module Capteur RFID



Module capteur RFID (ref : V-VM179)

Le module capteur RFID est une entrée tout ou rien répondant à une carte magnétique. Il agit donc comme un simple bouton poussoir lorsqu'on approche une carte magnétique spécifique.

Vous avez à disposition des cartes magnétiques permettant d'activer le module.

Il est nécessaire de fournir indépendamment une tension à ce module via une alimentation 12V à brancher sur la carte d'alimentation avec fusible et allumer le module grâce à l'interrupteur, la LED doit s'allumer.

Lorsque vous passez une carte devant le module, vous entendez un bruit signifiant qu'il a détecté quelque chose.

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	(libre)	C.5	
4	Module capteur RFID	C.4	Capteur_RFID
3	Capteur récepteur infrarouge	C.3	Recepteur_IR_A
2	Capteur de fin de course fermeture de la barrière	C.2	FDC_Fermeture_A
1	Capteur de fin de course ouverture de la barrière	C.1	FDC_Ouverture_A
0	Bouton poussoir intérieur	C.0	Bouton_Poussoir_A
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	(libre)	A.1	
0	Module capteur de courant	A.0	Capteur_courant
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	Connecté au moteur	B.0	Moteur_A
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	Module afficheur OLED (Option)	B.2	Afficheur_A*
1	Emetteur infrarouge	B.1	Emetteur_IR_A
0	Module voyant lumineux	B.3	Voyant_Lumineux





Exercice niveau 3 – E.1 : Utilisation du module capteur RFID

Objectif : Basculer le voyant lumineux lorsque le module détecte une carte magnétique.

Notion(s) abordée(s) : lire une entrée numérique

Correction :

	Blocs
début répéte faire	er indéfiniment si entrée Capteur_RFID v est activée v
	faire basculer Voyant_Lumineux attendre pendant 500 ms
	Fichier Blockly : BP_N3_E1.xml
	,

basculer Voyant_Lumineux •

Remarque : Le bloc

permet de changer l'état d'une sortie.



Exercice niveau 3 – E.2 : Utilisation du module capteur RFID (2)

Objectif : Ouvrir, attendre 5 secondes, et fermer la barrière lors du déclenchement du capteur RFID

Notion(s) abordée(s) : Condition dépendant d'une entrée numérique

Correction :

Blocs				
début répéter indéfiniment faire si entrée Capteur_RFID est activée faire si entrée FDC_Ouverture_A est désactivée faire appeler sous-fonction ouvrir attendre pendant 5000 ms appeler sous-fonction fermer	sous-fonction @uvrif sortie Moteur A a ctivée * répéter basculer Voyant Lumineux * jusqu'à entrée FDC_Ouverture A est activée * sortie Moteur A désactivée * sortie Voyant Lumineux * désactivée * sortie Moteur A a ctivée * répéter basculer Voyant Lumineux * désactivée * répéter basculer Voyant Lumineux * ittendre pendant 200 ms jusqu'à entrée FDC_Fermeture A est activée * sortie Moteur A désactivée * sortie Voyant Lumineux * désactivée * sortie Voyant Lumineux *			
FICHIER BIOCKIY : BP_IN3_E2.xml				





Tableau d'affectation des entrées et sorties

ES	MODULE DE COMMUNICATION POUR ENTRÉES / SORTIES NUMÉRIQUES	Broche Blockly	Etiquette Blockly
7	Communication Bluetooth envoi de données	C.7	BLTH_TX*
6	Communication Bluetooth réception de données	C.6	BLTH_RX*
EN	MODULES CAPTEURS POUR ENTRÉES NUMÉRIQUES		
5	(libre)	C.5	
4	Module capteur RFID	C.4	Capteur_RFID
3	Capteur récepteur infrarouge	C.3	Recepteur_IR_A
2	Capteur de fin de course fermeture de la barrière	C.2	FDC_Fermeture_A
1	Capteur de fin de course ouverture de la barrière	C.1	FDC_Ouverture_A
0	Bouton poussoir intérieur	C.0	Bouton_Poussoir_A
EA	MODULES CAPTEURS POUR ENTRÉES ANALOGIQUES		
3	(libre)	A.3	
2	(libre)	A.2	
1	(libre)	A.1	
0	Module capteur de courant	A.0	Capteur_courant
SN	MODULES ACTIONNEURS SORTIES NUMÉRIQUES		
7	Connecté au moteur	B.0	Moteur_A
6	(libre)	B.6	
5	(libre)	B.5	
4	(libre)	B.4	
3	(libre)	B.3	
2	Module afficheur OLED (Option)	B.2	Afficheur_A*
1	Emetteur infrarouge	B.1	Emetteur_IR_A
0	Module voyant lumineux	B.3	Voyant_Lumineux







Exercice niveau 4 – A.1 Mise en place de deux barrières

Objectif : Créer un programme permettant de faire fonctionner les deux barrières grâce aux entrées boutons poussoir ou capteurs RFID et barrières infrarouges. Les barrières doivent s'ouvrir indépendamment

Note : Dans ce premier exercice, il n'est pas obligatoire que les deux barrières puissent fonctionner en même temps.







Exercice niveau 4 – A.2 : Afficher le nombre de places disponibles

Objectif : Le parking possède 20 places maximum. A chaque fois qu'une voiture arrive (ouverture d'une première barrière), décrémenter le nombre de places. Incrémenter ce nombre lorsqu'une voiture sort du parking (Ouverture d'une seconde barrière).

Afficher le nombre de places sur les afficheurs LCD si vous possédez l'option. Bloquer l'entrée s'il n'y a plus de places dans le parking.







CONCEPTEUR ET FABRICANT DE MATÉRIELS PÉDAGOGIQUES