

STARTERKIT COLLÈGE

N2



Thème : serrure codée



OpenSource



Simple



Évolutif

Votre premier starterkit OpenSource pour collège !
Document librement diffusable

Version document : BETA 0.8

SOMMAIRE

[Ouvrir comme un document](#)

INFOS00	C'est quoi ce kit ?	03
ÉTAPE01	Petit tour d'horizon du matériel	04
ÉTAPE02	Installer son matériel	06
ÉTAPE03	Découverte de la programmation graphique	10
	Programme test 01	13
	Programme test 02	15
	Programme test 03	18
EXEMPLE01	RFID - Lecture simple d'un code RFID	22
EXEMPLE02	RFID - Ajoutons un afficheur LCD	27
EXEMPLE03	RFID - Gestion de plusieurs codes & couleurs	32
EXEMPLE04	RFID - Enregistrement des codes à la volée	36
EXEMPLE05	KEYPAD - Gestion d'un keypad et code	42
EXEMPLE06	KEYPAD - Exemple complet avec keypad	46

Merci d'avoir choisi notre tout premier StaterKit OpenSource pour collègue.

Vous trouverez dans ce petit dossier tout ce dont vous avez besoin pour démarrer avec Arduino en collègue. Mais ce n'est pas tout ! Les exemples proposés sont directement liés à une problématique classique en technologie collègue plutôt adaptée au niveau 4ème (même si rien ne vous empêche de travailler sur ce thème en 3ème). Vous devriez arriver rapidement au résultat attendu sans avoir à passer des heures à rechercher des solutions. Vous pourrez ensuite passer à des solutions plus évoluées et plus puissantes si vous en avez envie sans devoir tout changer : toutes nos solutions sont évolutives et libres.

La plate-forme Arduino est très large et peut très facilement s'adapter à vos besoins futurs. De plus, tous les logiciels utilisés sont libres et parfaitement adaptés au milieu scolaire. Comme il n'est pas prévu que les élèves aient à saisir du code, nous vous proposons de construire vos programmes par assemblage d'éléments graphiques. Rien ne vous empêche cependant d'essayer de comprendre le langage qui se cache derrière.

Si malgré nos efforts vous avez des problèmes pour installer, manipuler ou développer sur Arduino, nous sommes à votre disposition pour vous aider. N'hésitez pas à nous contacter sur notre messagerie.



Comment obtenir ce kit

>>

<http://duinoedu.com/store1/starter-kits-college/215-starterkitn2.html>



Besoin d'aide ?

>> <http://duinoedu.com/contacterduinoedu.html>

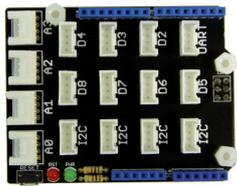
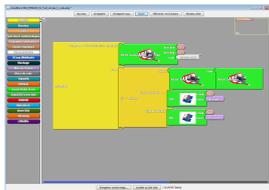
Arduino n'est pas simplement un produit mais aussi une communauté. N'hésitez pas à partager vos connaissances et votre expérience.

Nous vous souhaitons bonne lecture et bonnes expérimentations dans le monde fabuleux d'Arduino !

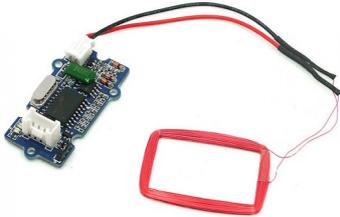
Commençons par vérifier notre matériel et voyons à quoi servent les différents éléments.

Ce kit (dans sa version complète) contient tout ce dont vous avez besoin pour réaliser 2 serrures codées (code de votre kit N2) utilisables directement avec vos élèves sans modification.

Votre kit doit contenir les éléments suivants :

NOM	PHOTO	UTILITÉ / REMARQUES
<p>Carte Arduino UNO (X1) Sauf si vous avez choisi le kit sans carte Arduino</p>		<ul style="list-style-type: none"> - Doit contenir votre programme - Doit exécuter votre programme - Dispose d'entrées / sorties - Se branche sur l'ordinateur - Contient un microcontrôleur
<p>Câble USB A-B (X1) Sauf si vous avez choisi le kit sans carte Arduino</p>		<ul style="list-style-type: none"> - Permet de brancher la carte Arduino à l'ordinateur - Peut être remplacé par un câble d'imprimante ou autre...
<p>Shield de connexions Grove UNO (X1) Sauf si vous avez choisi le kit sans carte shield de connexions Grove</p>		<ul style="list-style-type: none"> - Permet de connecter facilement les modules - Se monte sur la carte Arduino UNO - N'a pas d'intelligence
<p>Arduino augmenté =</p> <ul style="list-style-type: none"> ● Le logiciel Arduino + ● Les bibliothèques déjà installées + ● Ardublock augmenté + ● Des exemples 		<ul style="list-style-type: none"> - Permet de programmer nos modules en mode graphique ou avec le code. >> Téléchargeable ici (vous le téléchargerez plus tard)
<p>Câbles Grove (X2)</p> <ul style="list-style-type: none"> ● 1 pour la LED ● 1 pour l'afficheur LCD 		<ul style="list-style-type: none"> - Permettent de brancher les modules Grove sur le shield de connexion Grove
<p>Clavier matriciel Dupont (X1)</p>		<ul style="list-style-type: none"> - Permet de saisir les codes - Est passif (ne contient pas d'électronique) - Doit être branché avec des adaptateurs Grove / Dupont

		- Ce n'est pas pas un module Grove
--	--	------------------------------------

NOM (suite)	PHOTO (suite)	UTILITÉ / REMARQUES (suite)
Adaptateurs mâle Grove / Dupont (X5) <ul style="list-style-type: none"> ● 4 pour le clavier ● 1 pour le servo 		- Permettent de brancher le clavier matriciel au shield de connexions Grove - Nous adapterons 1 peu ces câbles plus tard
Lecteur RFID ATTENTION FRAGILE ! (X1)		- Permet de lire le code des tags RFID (chaque tag a un code unique) - Les tags que nous vous proposons sont des cartes RFID et des badges RFID. Vous pourrez plus tard compléter avec des puces, des pièces...
Lot de 5 tags (X1) <ul style="list-style-type: none"> ● 2 cartes blanches ● 1 badge rouge ● 1 badge bleu ● 1 badge jaune 		- Permet de disposer de 5 codes RFID uniques. - Le code unique est inscrit sur la carte ou le badge et ne peut pas être changé. - Seuls les 8 derniers caractères sont utiles (nous y reviendront plus tard)
Afficheur LCD (X1)		- Permet d'afficher des messages simples - Vous disposez de 2 lignes de 16 caractères
Module LED verte (X1)		- S'allume ou s'éteint pour signaler l'ouverture de la porte - Vous disposez de 2 lignes de 16 caractères
Module servo Dupont (X1)		- Vous permet éventuellement d'ouvrir ou fermer un élément de maquette type porte - Se branche par un adaptateur mâle Grove / Dupont

Switch
(X1)



- Permet de choisir entre 2 options par programme
- Nous l'utiliserons pour choisir entre : lire un code RFID ou enregistrer un code RFID

Cette partie est consacrée à toute la partie téléchargement et installation de votre matériel.

La carte Arduino intègre un microcontrôleur. C'est un composant programmable qui pour l'instant ne sait rien faire ou presque. Pour lui apprendre à réaliser des tâches utiles, il va falloir le programmer. Le plus simple pour nous est d'utiliser un ordinateur et quelques logiciels - tous gratuits !

Pour vous simplifier la tâche, nous avons déjà préparé pour vous une archive (un dossier compressé) nommée "Arduino augmenté" avec tout ce qu'il vous faut pour programmer vos montages présents et futurs. Cette archive - après décompression - est directement utilisable par vous-même ou vos élèves et intègre les dernières versions d'Arduino, d'Ardublock et des bibliothèques utiles. Elle sera maintenue à jour par nos soins à chaque évolution de l'un des outils mais il n'est pas indispensable de toujours disposer de la dernière version. Si vous souhaitez par la suite installer vous-même ces outils un par un, rien ne vous en empêche - bien entendu.

02.A - Nous allons télécharger Arduino augmenté

02.A.1 - Cliquez sur le lien ci-dessous



Arduino augmenté se trouve ici :

http://duinoedu.com/dl/logiciels/arduino/arduino_augmente/

02.A.2 - Choisissez ensuite le dossier **BETA_0.11**

En choisissant la version BETA 0.11, vous aurez exactement la même version que celle utilisée pour construire cette aide. Plus tard, vous pourrez choisir une version plus complète et éventuellement plus performante. Mais pour suivre ce tutoriel, nous vous conseillons d'utiliser cette version. Ainsi, les copies d'écran seront rigoureusement identiques à ce document.

02.A.3 - Cliquez ensuite sur :



Parent Directory

[arduino-1.0.5-r2.zip](#)

19-Aug-2014 15:30 103M

02.A.4 - Enregistrez votre fichier sur le bureau (ou ailleurs)

Le téléchargement peut prendre 1mn. Ensuite, vous devriez obtenir une archive, c'est-à-dire un dossier compressé avec tous les outils déjà installés.



Attention, il est possible que votre navigateur vous signale un risque suite à ce téléchargement. L'archive que nous vous proposons ne présente pas de virus connu.

02.A.5 - Décompressez l'archive précédente

Généralement, il suffit de cliquer avec le bouton droit et de choisir une option du type : "Extraire ici", "Extraire le fichier", "Décompresser"... ou équivalent. Vous devriez obtenir un dossier tout à fait classique appelé arduino-xxxxxx



Oui, il n'y a pas de réelle installation !

La version d'Arduino que nous vous proposons dans ce kit est portable. Vous n'avez donc pas réellement besoin de l'installer.

02.B - Nous allons installer le port virtuel (si besoin)

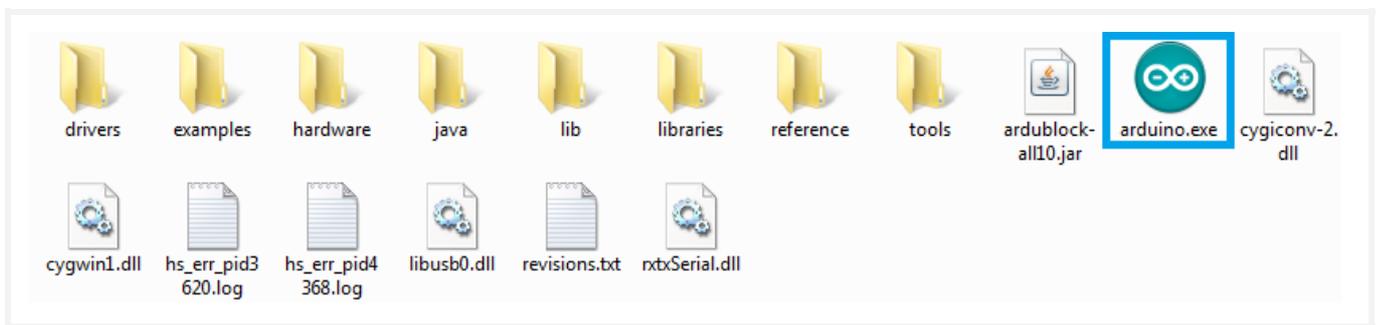
02.B.1 - Ouvrez votre dossier arduino

Vous devriez voir nombre de dossiers dont certains ne sont pas utiles pour vous. Nous verrons un peu plus tard ce qu'il faut connaître.

02.B.2 - Cliquez 2 fois sur arduino.exe

Comme vous l'aurez compris, arduino.exe est votre logiciel Arduino ! Vous pouvez éventuellement créer un raccourci vers ce fichier et cacher ce dossier.

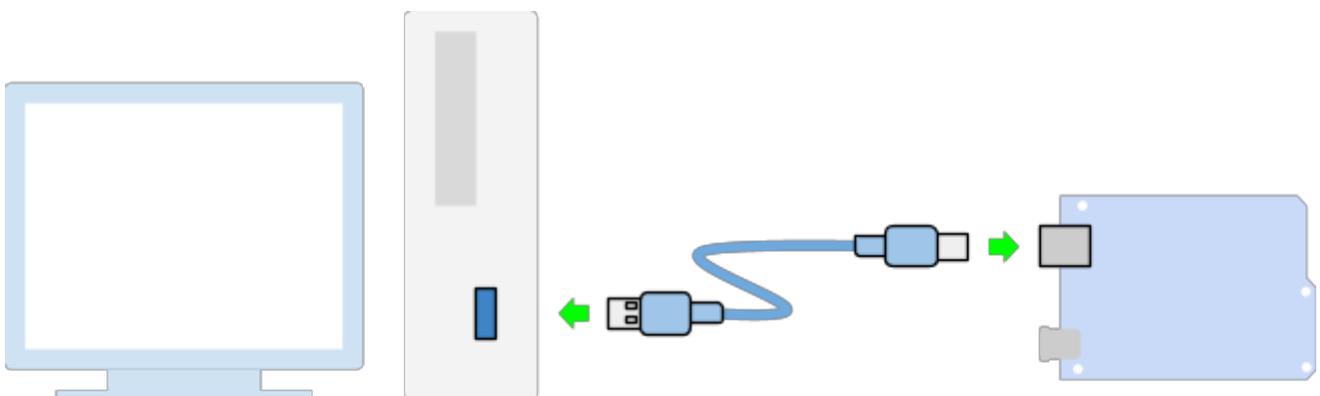
Pour lancer votre logiciel Arduino



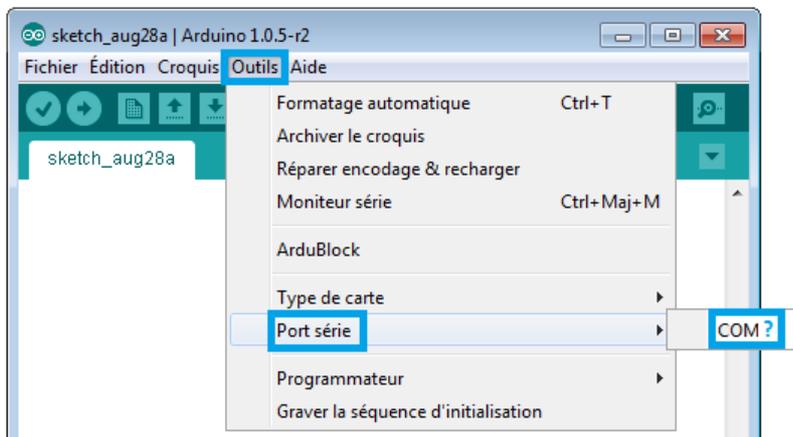
VOTRE LOGICIEL ARDUINO OUVERT	REMARQUES
	<ul style="list-style-type: none"> - C'est ici que doit être placé le fameux code. -Mais nous n'écrivons rien car c'est Ardublock augmenté - notre plug'in graphique - qui le fera pour nous. -Nous l'avons déjà mis en français pour vous mais normalement, c'est l'anglais la langue par défaut. -Plus tard, c'est ici que vous pourrez vous essayer à la rédaction de votre programme en langage EDU ou Arduino (voire d'autres langages).

02.B.3 - Branchez votre carte Arduino sur votre ordinateur

Suivant le type d'ordinateur et sa version, il est possible qu'un pilote soit nécessaire. Les ordinateurs récents ont généralement déjà les pilotes pour les cartes Arduino UNO et MEGA. Mais les ordinateurs plus anciens peuvent vous demander ces fameux pilotes.



02.B.4 - Dans le logiciel Arduino puis Outils puis Port série, vérifiez la présence de COMxx :



Si en branchant la carte Arduino :

- un port COMxx apparaît = la carte est installée => allez au **B.5**
- aucun port COMxx n'apparaît = la carte n'est pas installée => allez au **B.4**

02.B.5 - Si aucun COMxx n'apparaît, vous devriez :

- avoir un [message](#) vous invitant à installer les pilotes
- obtenir une [fenêtre](#) vous demandant l'[emplacement des pilotes](#)
- [sélectionnez](#) Bureau puis [arduino-xxxxxx](#) puis [drivers](#) comme emplacement
- laissez l'installation se terminer
- retournez au **B.4**



Les pilotes de la carte Arduino se trouvent ici :

- votre dossier [arduino-xxxxxx](#) puis [drivers](#)

A ce stade, les pilotes de votre carte Arduino doivent être installés. Il est inutile de continuer si votre carte n'est pas installée. Cette étape est la plus délicate et peut nécessiter un redémarrage de votre ordinateur dans quelques rares situations.

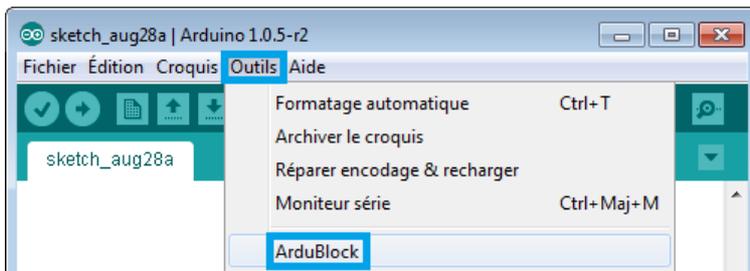
Cette partie est consacrée aux rudiments de la programmation graphique avec le plug'in libre Ardublock.

Arduino doit être programmé avec un langage appelé langage Arduino. C'est un dérivé des langages C/C++ un peu simplifié. Même si ce langage n'est pas très compliqué, il est possible de faire beaucoup plus simple ! C'est là qu'intervient le Plug'in Ardublock. C'est un petit programme s'intégrant directement dans Arduino. Rassurez-vous, nous avons intégré Ardublock pour vous et l'avons même augmenté pour faciliter son utilisation pour certains modules.

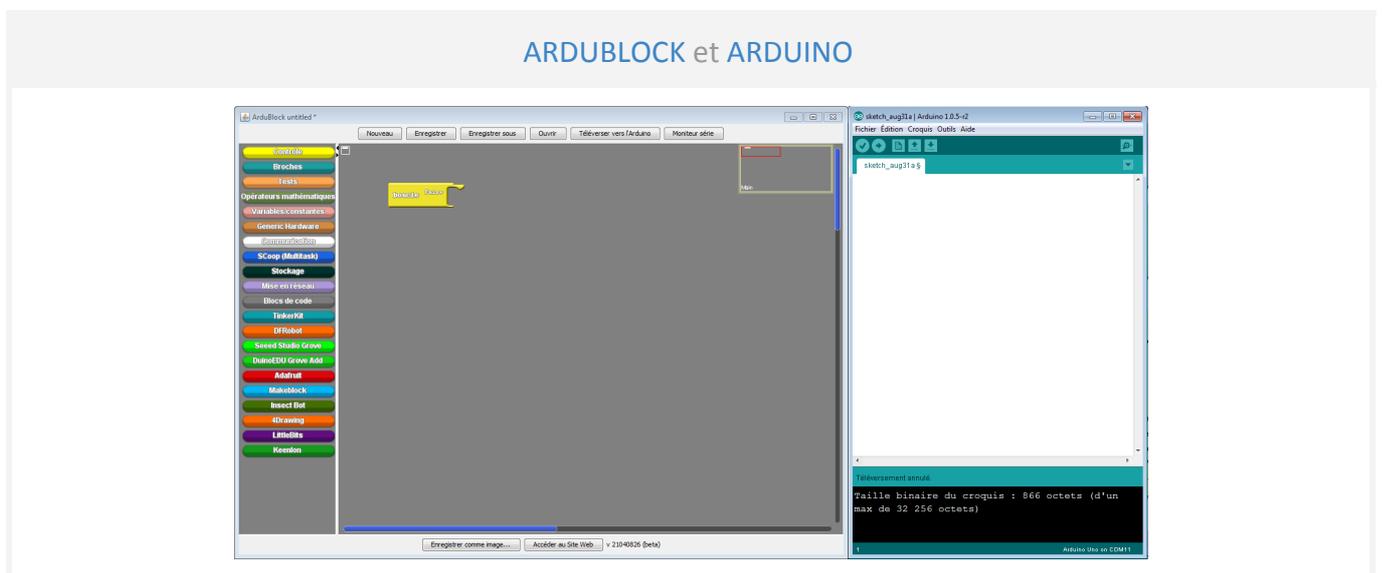
Retenez que le rôle d'Ardublock est de générer le code Arduino à votre place. Si vous souhaitez un jour programmer en langage Arduino, le passage pourra se faire très progressivement.

03.A - Nous allons lancer Ardublock et tester notre premier programme

Cliquez sur Outils puis Ardublock

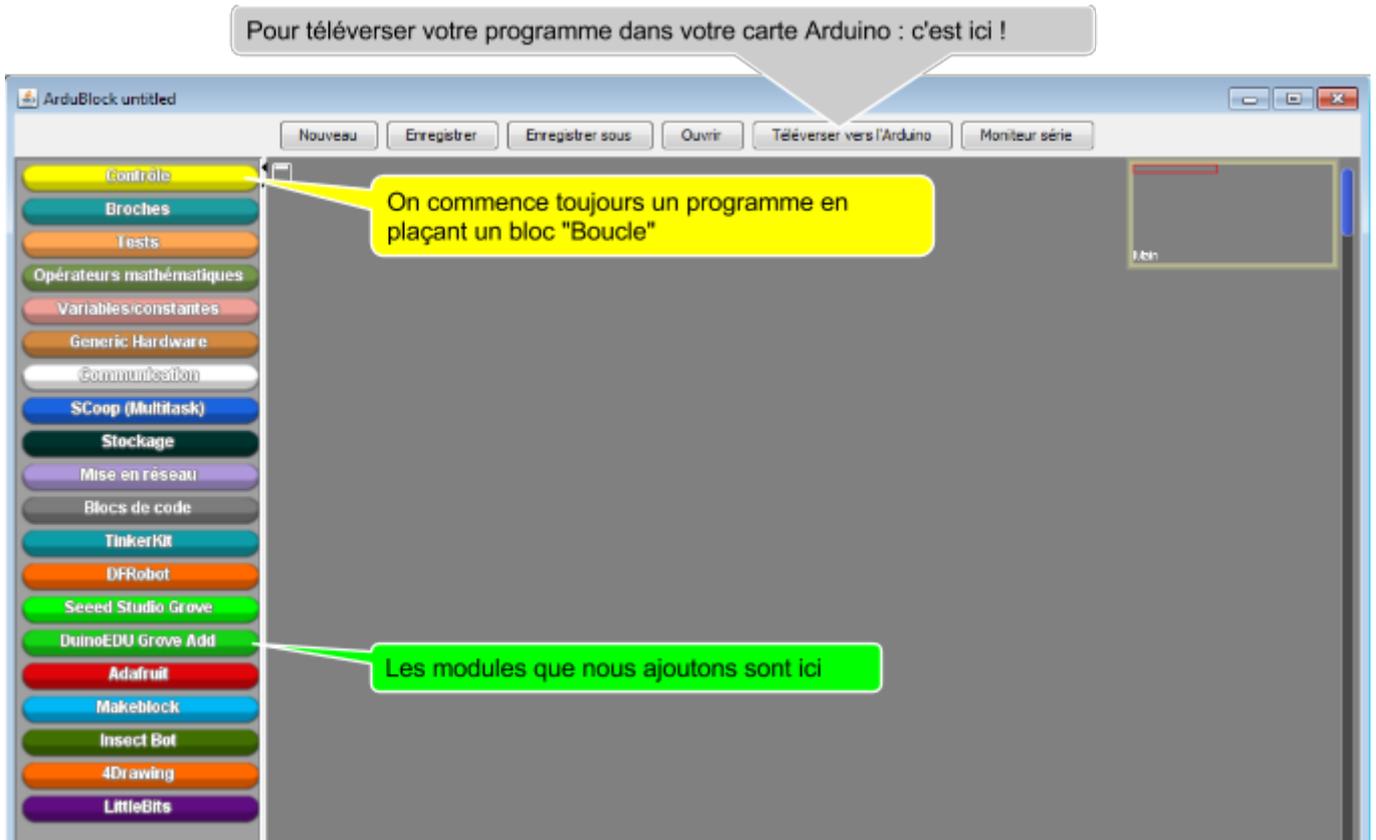


Attendez quelques instants, Ardublock devrait se lancer et vous obtiendrez :



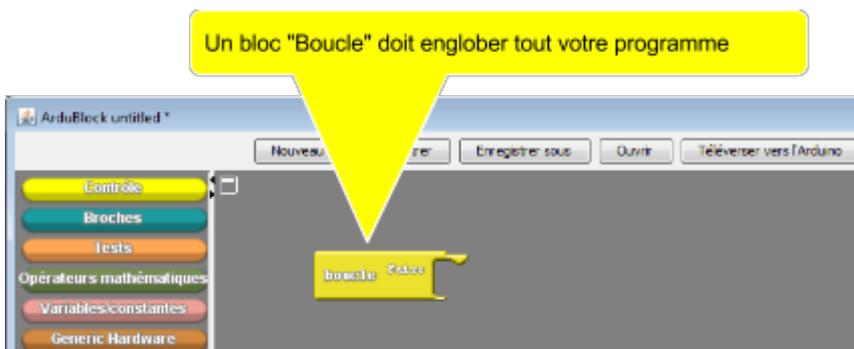
03.B - Nous allons tester le fonctionnement de toute la chaîne (Ardublock, Arduino, pilotes)

03.B.1 - Veillez noter que :



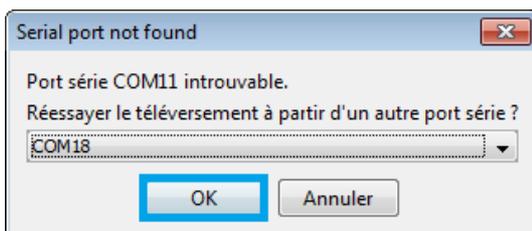
03.B.2 - Regardez bien, un bloc "Boucle" a été placé automatiquement

Le bloc "Boucle" englobe tout le programme. Il existe 2 types de bloc "Boucle" mais le plus simple est représenté ci-dessous. Retenez que tout votre programme (sauf exception) doit être inclus dans ce bloc !



03.B.3 - Cliquez ensuite sur "Téléverser vers l'Arduino"

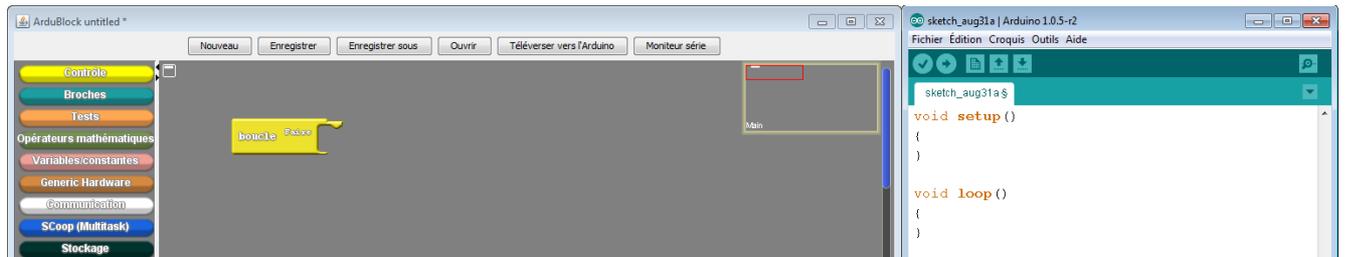
1. Le programme s'écrit tout seul dans le logiciel Arduino
2. Le premier téléversement se fera peut-être après confirmation du n° de port. Cliquez sur "Ok"



Si tout se passe bien :

1. le programme doit apparaître automatiquement dans le logiciel Arduino
2. il doit compiler (un message vous le confirme)
3. le programme compilé doit ensuite se téléverser dans la carte Arduino
4. ensuite, le message " Téléversement terminé" doit apparaître
5. le programme s'exécute dans la carte Arduino aussi longtemps que la carte Arduino est alimentée.

ARDUBLOCK a généré automatiquement le programme dans ARDUINO



Le logiciel Arduino vous dit ce qui se passe.

1. Téléversement terminé = votre programme est opérationnel !
2. Compilation terminée = votre programme est correct mais n'est pas dans la carte
3. Vous avez un message orange = il y a 1 erreur de syntaxe dans Arduino

Les outils maintenant installés, nous pouvons enfin programmer. La démarche classique est la suivante.



Démarche classique pour programmer avec Ardublock

1. Réfléchir au résultat attendu
2. Branchez les modules sur le shield de connexion par des câbles
3. Placez vos blocs (image de votre programme)
4. Cliquez sur "Téléverser vers l'Arduino"

03.C - PROGRAMME TEST 01

03.C.1 - Le résultat attendu est le suivant :

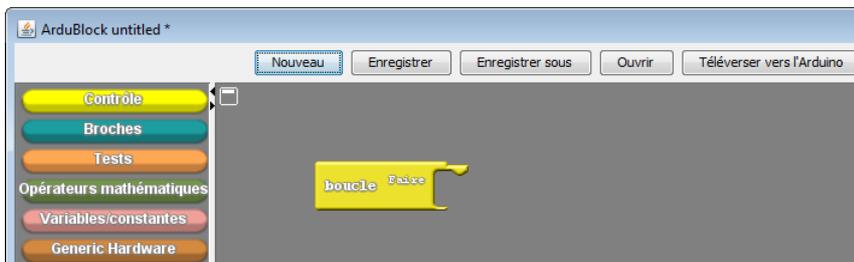


Objectif du programme TEST01

1. Allumer simplement 1 LED
2. et rien d'autre pour le moment

03.C.2 - Nous allons commencer par injecter le programme minimal.

Pour cela, il faut vous arranger pour **ne garder que le bloc "Boucle"**. Comme ceci :



Si vous avez suivi à la lettre ce tutoriel, cela devrait être le cas. Sinon, retenez ceci :



Pour supprimer un bloc :

- faites-le glisser vers le menu de gauche (il retournera d'où il vient)
- pour placer un bloc "Boucle" allez dans Contrôles"

C'est fait ? Vous n'avez plus qu'un bloc "Boucle" ? Passez alors à la suite !

Maintenant, cliquez sur le bouton : "**Téléverser vers l'Arduino**".

Cela permet de compiler et d'injecter le programme minimal, c'est-à-dire, un programme totalement neutre pour vos modules. Même si cela n'a pas d'importance pour le moment, prenez l'habitude d'injecter le programme minimal avant de changer les branchements de vos montages.



Pourquoi injecter le programme minimal avant un nouveau montage ?

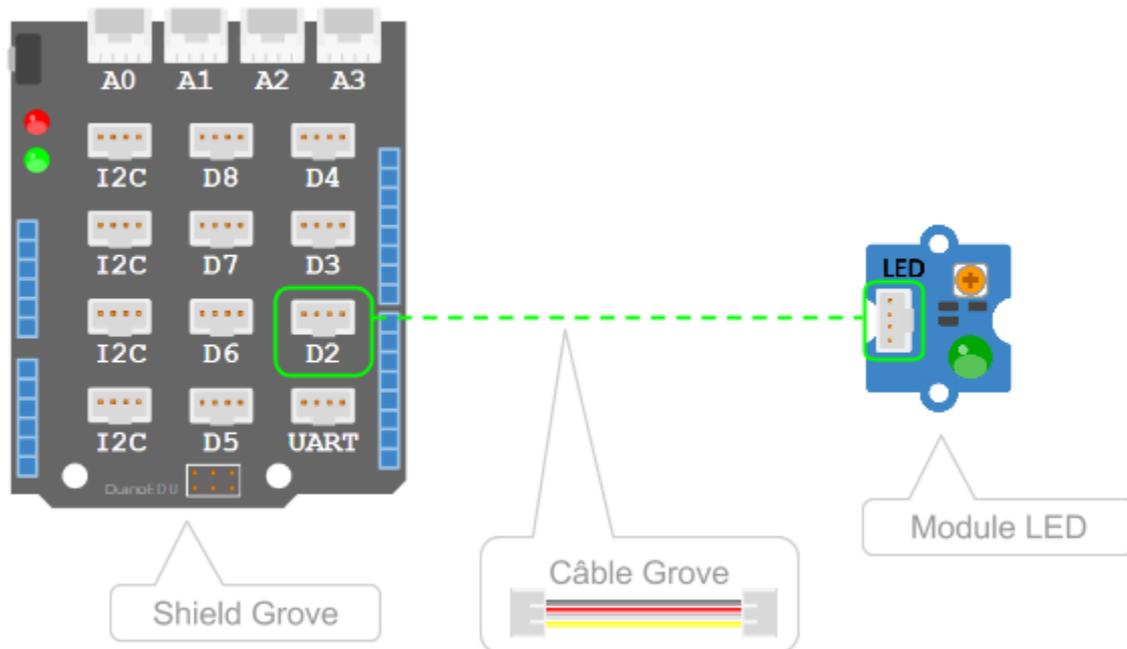
Imaginons que vous branchez un capteur (qui donne donc une information) sur une prise qui alimentait précédemment une LED : il y aura un court-circuit. En général, il n'y a pas de conséquence fâcheuse avec des modules élémentaires mais certains modules évolués (souvent beaucoup plus chers) n'apprécient pas ce genre d'erreur.

03.C.3 - Nous allons ajouter un module LED

Le branchement est très simple : prenez votre module LED, prenez un câble Grove et réalisez le branchement suivant.

REM : branchez sans crainte ! le programme que nous avons injecté est totalement neutre !

Reliez votre module LED au shield de connexion par un câble Grove



03.C.4 - Nous allons ajouter un module LED dans Ardublock

Pour cela, c'est assez simple :

1. cliquez sur le bouton DuinoEDU (menu de gauche) => un menu apparaît
2. faites glisser l'objet LED du menu vers votre espace de travail
3. emboîtez ensuite votre bloc LED dans la mâchoire boucle (ou loop)
4. écrivez ensuite au clavier "2" pour l'option "Broche" *1

*1 : rappelez-vous, c'est le n° de la prise sur laquelle vous avez branché votre LED !

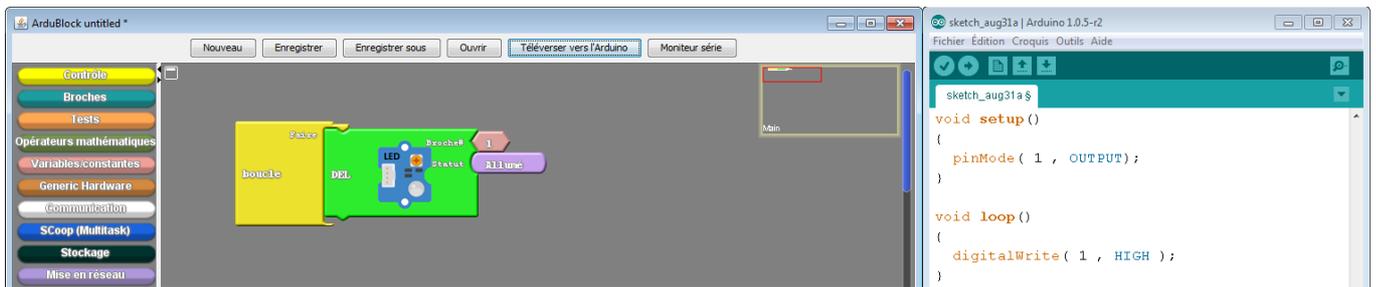
Vous devriez obtenir ceci :

VOTRE PROGRAMME ARDUBLOCK	EXPLICATIONS
<p>The screenshot shows the Ardublock programming environment. A yellow 'boucle' block contains a green 'LED' block. The 'LED' block has a 'Broche#' input set to '2' and a 'Statut' input set to 'Allumé'.</p>	<p>Broche : permet de choisir le n° de la prise concernée.</p> <p>Allumé : la LED doit se trouver dans l'état Allumé</p> <p>DEL : le nom du bloc pour piloter les LED</p> <p>Boucle : le programme principal (oui ! il peut aussi y avoir des sous-programmes)</p>

03.C.5 - Cliquez maintenant sur "Téléverser vers l'Arduino"

Le programme correspondant à vos blocs est automatiquement créé et se téléverse dans la carte Arduino. Dans quelques instants, votre LED branchée sur D2 devrait s'allumer.

Vous devriez obtenir ceci :



Ce petit programme ne fait pas grand chose. Mais notez que la LED s'est allumée par programme ! Passons à un exemple un peu plus complet.

01.C - PROGRAMME TEST 02

02.C.1 - Le résultat attendu est le suivant :



Objectif du programme TEST02

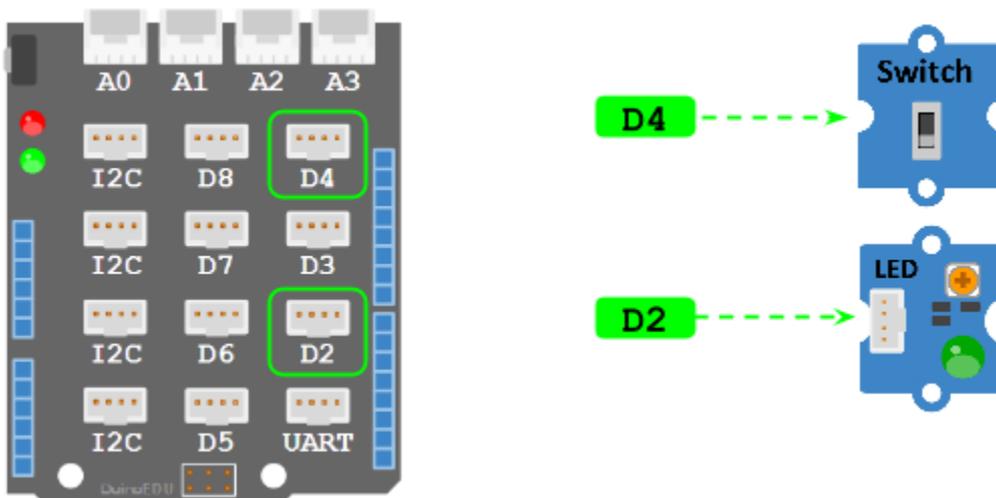
3. Si l'interrupteur est fermé, la LED doit être allumée
4. Sinon, la LED doit s'éteindre

Nous arrivons à un point essentiel en programmation : les tests. Ici, notre test est très simple mais permet déjà de faire nombre de programmes.

02.C.3 - Nous allons donc ajouter un module Switch

Le branchement est aussi simple que pour la LED. Mais comme vous débutez, nous allons prendre une habitude. Nous n'allons brancher nos modules que sur les prises paires pour l'instant. Il est tout à fait possible d'utiliser aussi les prises impaires, mais comme certains modules utilisent 2 prises en même temps sans que cela soit très visible et que vous ne savez pas encore les différencier, nous allons nous contenter des prises paires dans un premier temps.

Reliez votre module Commutateur au shield de connexion par un câble Grove



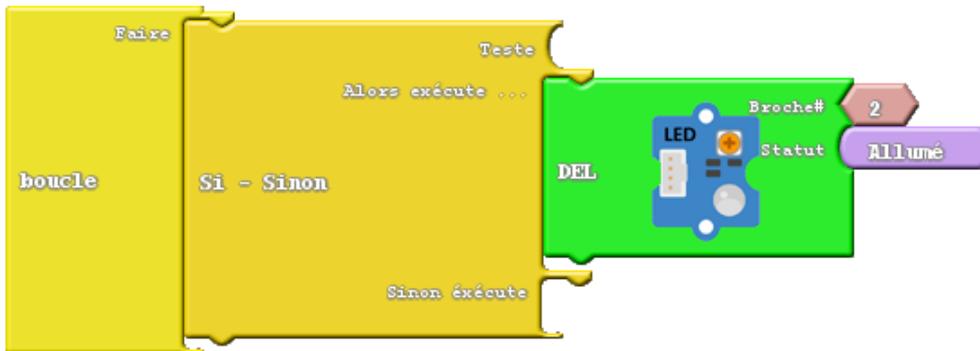
02.C.1 - Nous allons maintenant ajouter un bloc "Si - Sinon"

Le bloc "Si - Sinon" est justement fait pour introduire une condition du type : s'il pleut, ouvre ton parapluie, Sinon, ferme ton parapluie.

Procédez comme ceci :

1. cliquez sur le bouton "**Contrôles**" (menu de gauche)
2. faites glisser le bloc "**Si - Sinon**" du menu vers votre espace de travail
3. **intercalez-le** entre le bloc "Boucle" et le bloc "DEL"

Vous devriez obtenir ceci :



C'est très bien, mais que doit-on mettre dans la mâchoire "test" ? Cherchez un peu...Oui, c'est effectivement le commutateur ! C'est assez logique. C'est lui qui d'une certaine façon "fait la pluie ou le beau temps".

Plus précisément, c'est le commutateur qu'Arduino devra tester pour savoir s'il faut :

- allumer la LED (ou)
- éteindre la LED.

C'est donc bien lui qu'il nous faut.



Attention, certains modules occupent 2 prises en même temps.

Seeed Grove a choisi de brancher tous ses modules avec un seul câble. C'est très pratique mais impose d'être un peu vigilant. Si vous regardez les connecteurs de près, vous verrez qu'ils ont 4 pointes. Deux sont utilisées pour alimenter le module. Une correspond à la prise en question (D2,D3...). Mais à quoi sert la 4ème ? C'est là que réside l'astuce. Quand vous branchez une prise sur D2, le module peut utiliser la quatrième pointe pour avoir aussi accès à la prise D3 sans que cela vous oblige à brancher un autre câble. Rassurez-vous, nous y reviendrons. Mais retenez déjà cela.

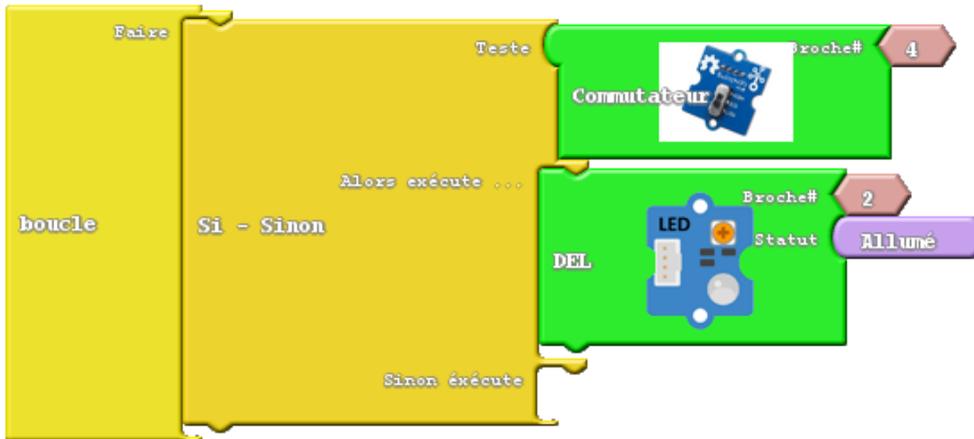
02.C.1 - Nous allons donc ajouter un bloc Commutateur

Branché sur une entrée Arduino, le commutateur permet de choisir entre 5V (état haut) et 0V (état bas). En d'autres termes, vous donnez à l'utilisateur la possibilité de choisir entre 2 options. Vous donnez des instructions à la carte Arduino qui peut agir en conséquence.

Procédez comme ceci dans Ardublock :

1. Cliquez sur le bouton **Seeed Studio Grove**
2. faites glisser l'objet **Commutateur (ou switch)** dans l'espace de travail (il faut descendre un peu dans le menu)
3. **emboîtez** ensuite votre **bloc Switch** dans la mâchoire "Test"
4. écrivez ensuite au clavier **"4"** pour l'option "Broche" (comme prise n°4)

Vous devriez obtenir ceci :



Nous voyons au premier coup d'oeil que le programme allumera la LED, mais ne l'éteindra jamais.

02.C.1 - Nous allons cloner le bloc LED et modifier son état

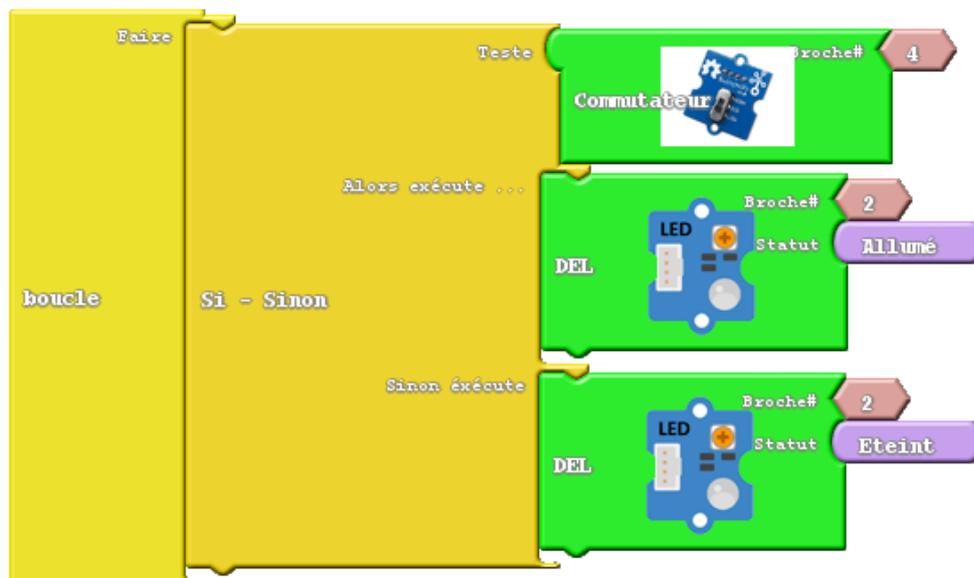
Procédez comme ceci :

5. placez votre souris au-dessus de votre bloc LED
6. cliquez avec le bouton droit de la souris => un menu apparaît
7. sélectionnez tout simplement cloner => le bloc a été dupliqué
8. placez ensuite le bloc cloné dans la mâchoire "Sinon exécute"

Pour changer l'état de la LED, procédez comme ceci :

1. placez votre souris au-dessus de l'étiquette "Allumé" => une petite flèche apparaît
2. cliquez sur la petite flèche => un menu apparaît
3. sélectionnez "Eteint"

Vous devriez obtenir ceci :



02.C.1 - Cliquez ensuite sur "Téléverser vers l'Arduino"

=> Votre programme devrait se téléverser tout seul dans la carte Arduino.

Effectivement, notre programme n'est pas des plus intéressants. Mais il est possible de remplacer le commutateur par un fin de course pour détecter, par exemple, une ouverture de porte, et, d'une manière générale, il est possible de brancher n'importe quel capteur pouvant délivrer une information numérique.

Imaginons maintenant que nous souhaitions piloter une porte sur 90°. Nous gardons notre interrupteur et notre LED mais en plus nous allons piloter ce que l'on appelle un servomoteur : un moteur capable de faire tourner son axe sur un certain angle. C'est justement l'objet du programme test 03.

03.C - PROGRAMME TEST 03

03.C.1 - Le résultat attendu est le suivant :

Objectif du programme TEST03



5. Si l'interrupteur est fermé,
 - a. la LED doit être allumée
 - b. la porte doit être ouverte à 90°
6. Sinon,
 - a. la LED doit s'éteindre
 - b. la porte doit se refermer

La seule chose qui manque à notre montage est un servomoteur.



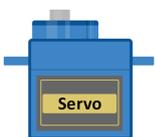
Pour rendre ce test plus visuel :

- nous vous invitons à monter un des palonniers fournis sur l'axe du servomoteur
- vous pouvez aussi fixer sur ce palonnier un élément qui matérialisera la porte

03.C.3 - Nous allons donc ajouter un servomoteur

Le branchement est un peu différent. Grove est un système très ouvert acceptant nombre de modules d'autres marques. Nous allons vous montrer comment brancher un servo qui n'est pas Grove sur un shield de connexion Grove. Il existe bien sûr des servos Grove mais ils sont un peu plus chers.

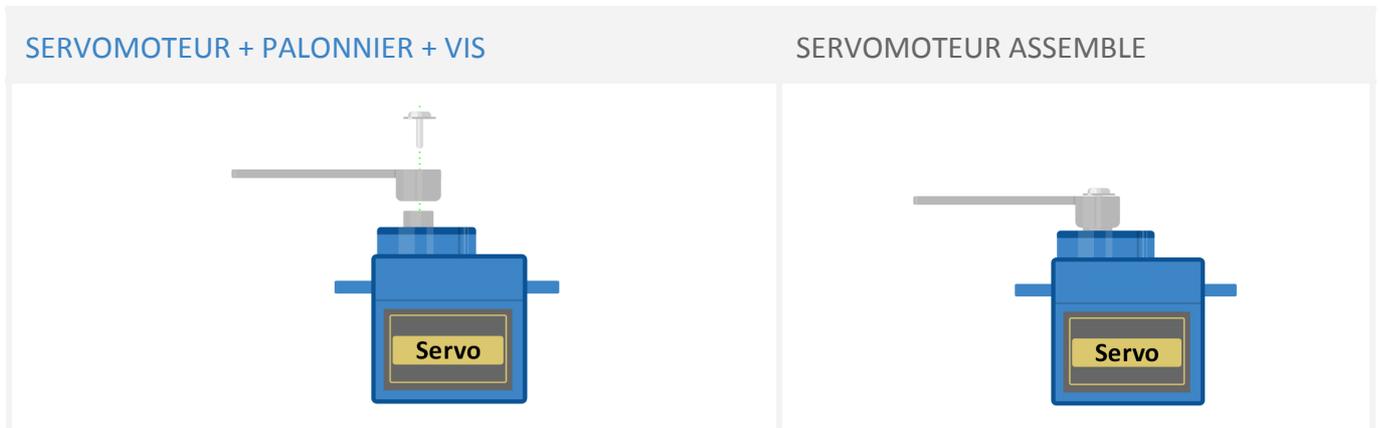
Prenez votre servomoteur



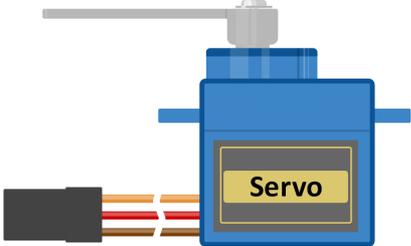
Montez un palonnier fourni sur l'axe de votre servomoteur

Pour cela, vous pouvez simplement l'emboîter, il n'est pas absolument nécessaire de le visser.

Vous devriez obtenir ceci :



Reliez un adaptateur Grove à la prise du servomoteur

ADAPTATEUR	BRANCHEMENT	SERVOMOTEUR DUPONT
	<p>Procédez comme ceci :</p> <ol style="list-style-type: none">1. fil noir Grove => fil marron servo2. fil rouge Grove => fil rouge servo3. fil noir Grove => fil orange servo	



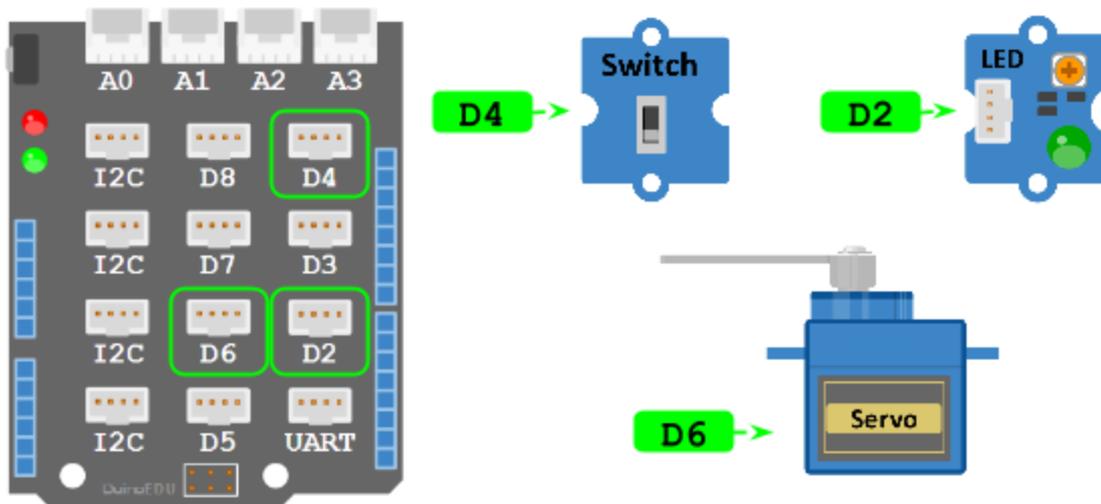
Il existe aussi des servomoteurs Grove !

Mais en procédant de cette façon, vous découvrez qu'il est aussi possible de brancher des servomoteurs de tous types même s'ils ne sont pas de marque Grove !

Il ne reste plus qu'à le brancher avec la prise.

Reliez votre servomoteur au shield de connexion par l'adaptateur branché précédemment.

Encore une fois, nous avons branché le servomoteur sur une prise paire même si tous ces modules pourraient très bien être branchés les uns à la suite des autres. Mais rappelez-vous : certains modules évolués utilisent 2 prises en même temps et nous ne savons pas encore les différencier.



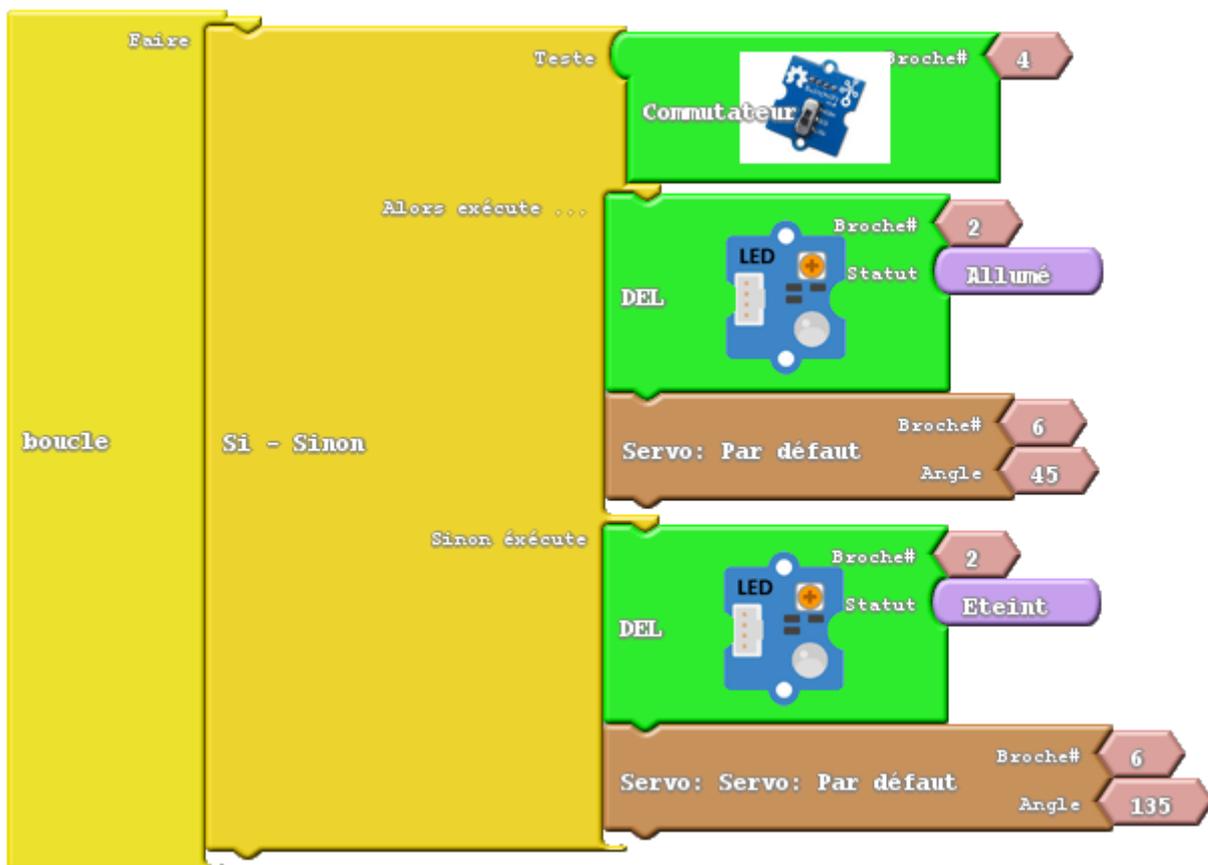
03.C.4 - Il reste maintenant à **placer le bloc "Servomoteur" dans Ardublock**

Le principe est toujours le même mais il y a une étiquette qui va nous intéresser tout particulièrement. C'est l'angle. Il suffit d'écrire un angle au clavier pour que l'axe du servomoteur le respecte dans la mesure du possible. Car comme pouvez vous en douter, il y a des limites. L'axe de ce type de servomoteur ne peut pas faire plusieurs tours. D'ailleurs, il ne peut faire théoriquement qu'un demi-tour, voire un peu moins dans les faits.

Procédez comme ceci :

4. cliquez sur le bouton "Generic" (menu de gauche)
5. faites glisser le bloc "Servo" du menu vers votre espace de travail
6. placez-le sous une LED
7. écrivez "6" dans l'étiquette Broche
8. écrivez "45" dans l'étiquette Angle
9. clonez-le
10. placez le clone sous l'autre LED
11. écrivez "135" dans l'étiquette Angle
12. téléversez l'ensemble en cliquant sur "**Téléverser vers l'Arduino**"

Vous devriez obtenir ceci :





Attention aux angles pour les servos

- en théorie, un servomoteur angulaire peut décrire un angle de 0° à 180°
- en réalité, c'est un peu moins et c'est variable d'une marque à l'autre
- pour être tranquille, **restez dans la fourchette : 15° à 165°**



Pour obtenir une ouverture sur 90° :

- le plus sûr est de choisir les angles : **45° / 135°**
- mais rien ne vous empêche de choisir d'autres valeurs

Nous allons maintenant passer à des exemples plus complets. Vous avez déjà une idée du fonctionnement global des blocs dans Ardublock et des modules Grove.

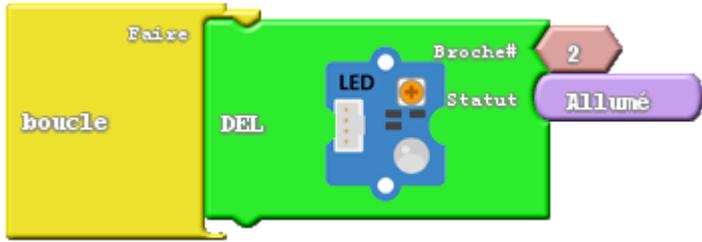
03.C.5 - Nous allons ajouter un module LED dans Ardublock

Pour cela, c'est assez simple :

5. cliquez sur le bouton DuinoEDU (menu de gauche) => un menu apparaît
6. faites glisser l'objet LED du menu vers votre espace de travail
7. emboîtez ensuite votre bloc LED dans la mâchoire boucle (ou loop)
8. écrivez ensuite au clavier "2" pour l'option "Broche" *1

*1 : rappelez-vous, c'est le n° de la prise sur laquelle vous avez branché votre LED !

Vous devriez obtenir ceci :

VOTRE PROGRAMME ARDUBLOCK	EXPLICATIONS
	<p>Broche : permet de choisir le n° de la prise concernée.</p> <p>Allumé : la LED doit se trouver dans l'état Allumé</p> <p>DEL : le nom du bloc pour piloter les LED</p> <p>Boucle : le programme principal (oui ! il peut aussi y avoir des sous-programmes)</p>

Dans cet exemple, vous allez utiliser un module lecteur RFID Grove. Celui-ci sera utilisé pour tester la validité d'un code déjà préalablement enregistré.

Il est important de comprendre que les tags (Badges, Cartes...) ont un code RFID unique inscrit en dur. Il existe des tags RFID enregistrables mais qui ne seront pas traités dans ce dossier car ils ne présentent pas d'intérêt pour ce que nous voulons faire.

Ce premier exemple reste simple mais va vous permettre d'utiliser votre premier module évolué. Il s'agit d'un module série 2 fils. Cela signifie qu'il occupe 2 connecteurs.

Nous espérons que vous n'avez rien débranché car nous allons nous contenter de compléter le montage précédent. Si vous avez tout débranché, remplacez vos modules comme précédemment.

■ EX01 - PROGRAMME EXEMPLE 01

■ EX1.1 - Le résultat attendu est le suivant :

Objectif du programme EXEMPLE01

1. état initial de notre serrure :
 - a. un identifiant RFID est pré-enregistré
 - b. la porte est initialement fermée
2. l'utilisateur saisit un mot de passe faux :
 - a. rien ne change
3. l'utilisateur saisit le bon mot de passe :
 - a. la LED s'allume pendant 4 secondes
 - b. la porte s'ouvre pendant 4 secondes

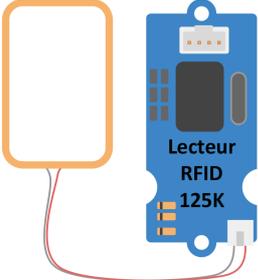


Le module RFID utilise une librairie pour fonctionner. Rassurez-vous, nous l'avons déjà installée mais ne perdez pas de vue que chaque module évolué a sa propre librairie disponible dans le dossier *libraries* d'Arduino.

Passons maintenant aux choses sérieuses.

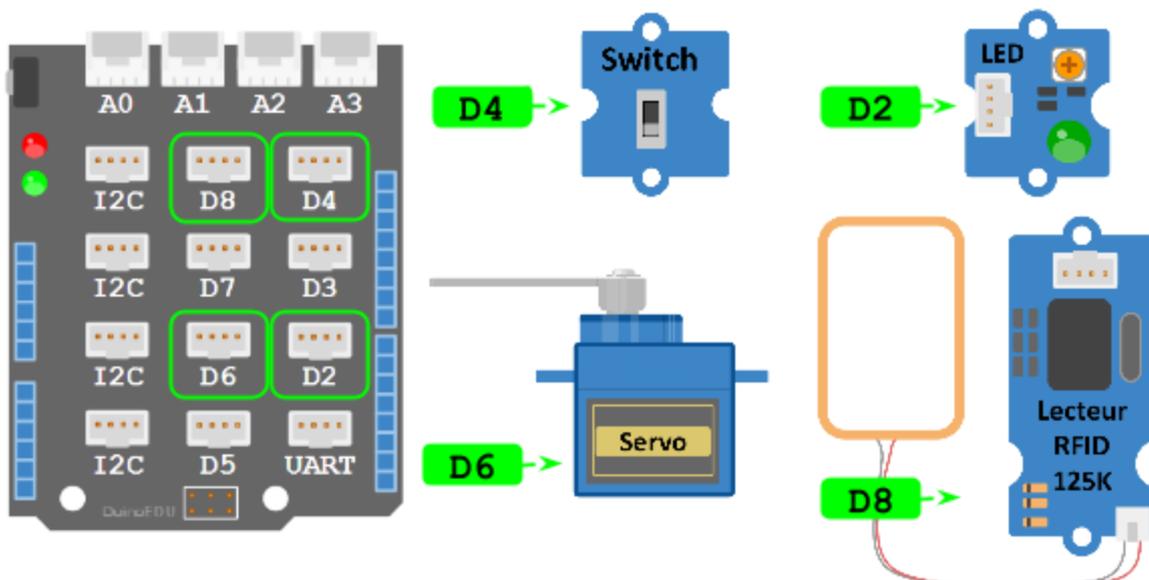
EX01.2 - Découvrons le nouveau matériel

Le matériel nouveau concerne le RFID. Il existe plusieurs types de RFID notamment liés à la fréquence de fonctionnement. Pour que 2 éléments RFID puissent communiquer, ils doivent être sur la même fréquence. Une fréquence très classique est le 125K (ou 125 kHz). Si vous devez faire des expérimentations avec d'autres tags RFID, assurez-vous qu'ils soient de type 125K pour fonctionner sur ce montage.

MATÉRIEL NOUVEAU		BRANCHEMENT
Lecteur RFID 125K		<ul style="list-style-type: none">- Communication : Série 2 fils- Se branche avec : 1 câble Grove- Occupe réellement : 2 prises  X1
Badges RFID 125K		<ul style="list-style-type: none">- Type : RFID- Fréquence : 125 kHz- Identifiant : unique
Cartes RFID 125K		<ul style="list-style-type: none">- Type : RFID- Fréquence : 125 kHz- Identifiant : unique

EX01.3 - Nous allons maintenant brancher notre lecteur RFID

Ce lecteur RFID est de marque Seeed Grove, il se branche donc avec un seul câble. Branchez-le sur le connecteur **D8**.

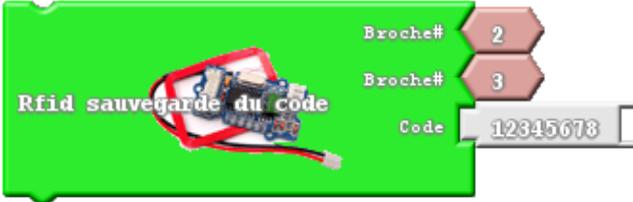
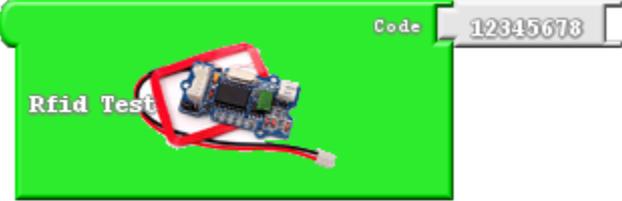


EX01.4- Nous allons maintenant ajouter les blocs RFID

Ce qui serait intéressant, ce serait de remplacer le test précédent par un test de code RFID. Comme cette tâche n'est pas aussi simple qu'elle en a l'air, nous avons ajouté pour vous des fonctions aux blocs RFID permettant de gérer cela pour vous. Il suffit maintenant de savoir s'en servir. Mais vous verrez que nous allons nous en sortir assez facilement.

Découvrons les fonctions des blocs RFID

Prenez un moment pour comprendre à quoi servent les différents blocs.

BLOCS RFID	UTILITÉ
 A green block with a photo of an RFID module. On the right, there are three input fields: 'Broche#' with '2', 'Broche#' with '3', and 'Code' with '12345678'.	<p>Permet d'initialiser le lecteur RFID</p> <p>-Broche# : si vous branchez ce module sur : D8, alors vous devez choisir :</p> <ul style="list-style-type: none">● Broche# 8● Broche# 9 <p>Rappelez-vous, ce module occupe 2 prises !</p> <p>-Code : permet d'enregistrer des codes. (séparez les codes par un espace s'il y en a plusieurs)</p>
 A green block with a photo of an RFID module and a 'Code' input field containing '12345678'.	<p>Permet de tester un code RFID</p> <p>-Code : le code à comparer aux codes enregistrés</p>
 A green block with a photo of an RFID module.	<p>Permet de lire le code présenté</p> <p>Ce bloc peut être emboîté directement dans la mâchoire Code du précédent bloc.</p>
 A green block with a photo of an RFID module.	<p>Permet d'effacer tous les codes enregistrés</p>

Ajoutons maintenant les blocs RFID

Procédez comme ceci :

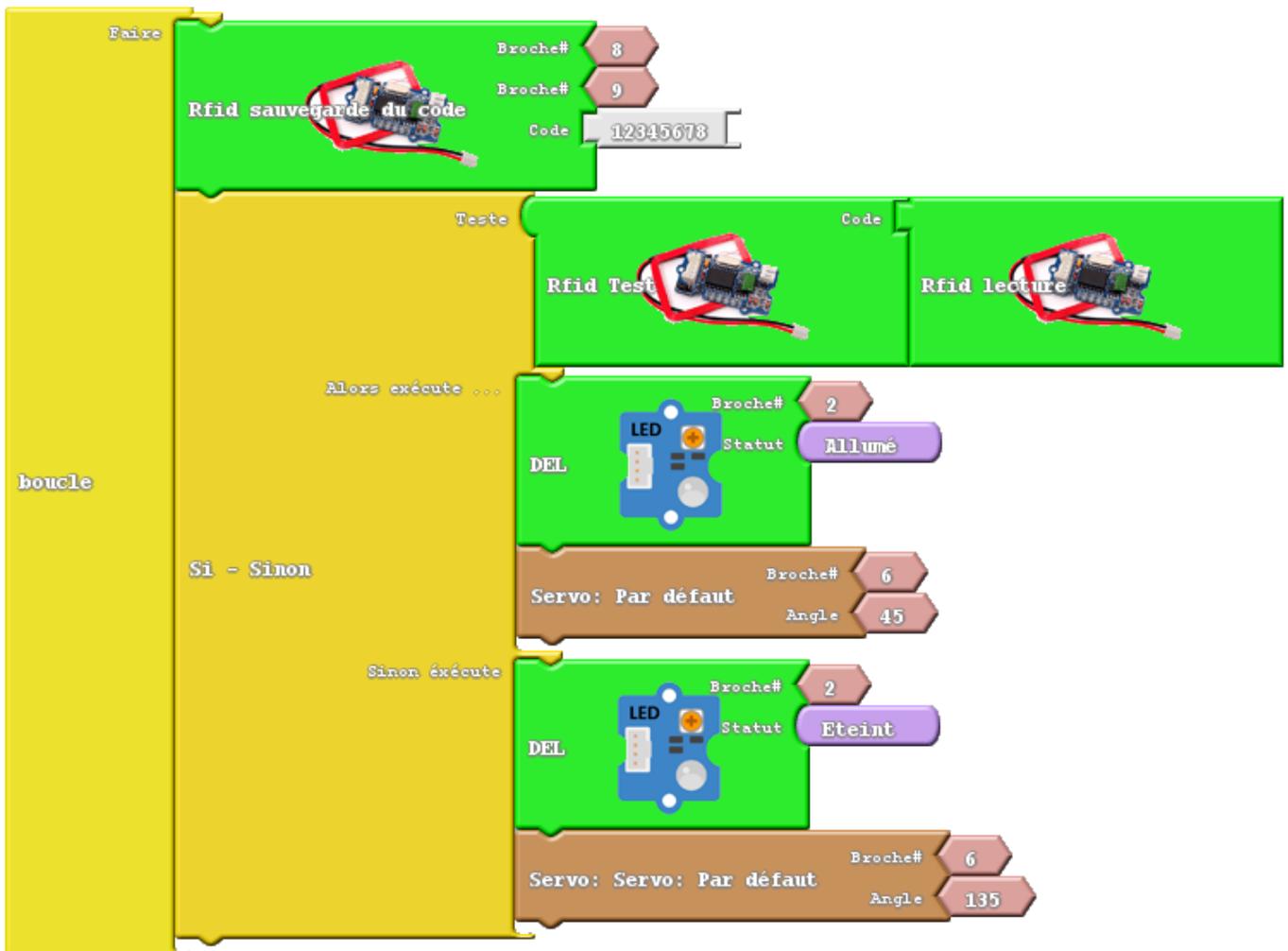
1. Placez le bloc "RFID sauvegarde du code" directement en haut dans la boucle Faire
 - a. broche# = 8
 - b. broche# = 9
 - c. Code = les 8 derniers chiffres du code inscrit sur le tag (carte ou badge)
2. Placez le bloc "RFID Test" à la place du commutateur (ou switch)
3. Placez le bloc "RFID lecture" dans la mâchoire du bloc précédent (supprimez ce qui s'y trouve)
4. Injectez le programme dans la carte Arduino

Personnalisons du "code RFID valide"

Bien entendu, le code RFID valide 12345678 doit être remplacé par les 8 derniers caractères d'un de vos tags RFID.

Pour une carte et badges Grove, vous devriez vous débarrasser normalement des 2 premiers 0.

Vous devriez obtenir ceci :



Comment cela peut-il fonctionner ? Il n'y a pas de temporisation !

Si vous avez analysé le programme, peut-être avez-vous été surpris par l'absence de temporisation. Cela vient du fait que nous l'avons ajoutée au niveau de la fonction de lecture du code. Ainsi, lorsqu'un code autorisé est détecté, le test est bon seulement pendant une durée déterminée (nous avons choisi 4 secondes).

C'est ce qui se passe dans la réalité avec les digicodes. Un code bon n'ouvre la porte que pendant un certain temps. Nous avons choisi de faire ainsi dans un souci de simplicité d'utilisation.

■ EX01.5 - Testez votre montage

Procédez comme ceci :

1. Placez la carte ou le badge dont le code est correct à quelques centimètres de la bobine
 - a. le lecteur doit lire ce code et le comparer à celui que vous avez saisi dans Ardublock
 - b. la porte s'ouvre et la LED s'allume
 - c. la porte doit s'ouvrir
2. Cette situation doit rester en l'état pendant 4 secondes environ
3. Après 4 secondes (temporisation)
 - a. le code lu devient périmé
 - b. la LED s'éteint et la porte se ferme
 - c. la porte se ferme et la LED s'éteint
4. Placez un tag non autorisé, la LED doit rester éteinte.

Nous allons maintenant compliquer un peu les choses avec l'exemple 02. Nous allons ajouter un afficheur LCD.

Ne débranchez rien, l'exemple 02 ne fait qu'ajouter un élément d'affichage au montage précédent.

Le seul composant que nous allons ajouter est un afficheur I2C. C'est un très bon exemple de module avec communication type "bus".

Jusqu'ici, nous avons toujours associé un module à un n° de prise bien précis. Mais avec une communication type bus comme l'I2C, plusieurs modules peuvent être connectés sur une même prise. C'est un grand avantage. Pour reconnaître le module, il faut une adresse. Heureusement, vous n'avez pas à connaître l'adresse de votre module I2C. Mais sachez qu'elle existe.

Nous espérons que vous n'avez rien débranché car nous allons nous contenter de compléter le montage précédent.

■ EX02. - PROGRAMME EXEMPLE 02

■ EX02.1 - Le résultat attendu est le suivant :

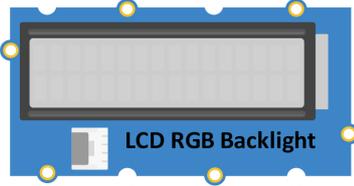
Objectif du programme EXEMPLE02

- 
4. état initial de notre serrure :
 - a. un identifiant RFID est pré-enregistré
 - b. la porte est initialement fermée
 - c. l'afficheur affiche : "saisir code" sur la 1ère ligne
 5. si l'utilisateur saisit un mot de passe faux :
 - a. rien ne change
 6. si l'utilisateur saisit le bon mot de passe :
 - a. l'afficheur affiche : "code bon"
 - b. la porte s'ouvre pendant 4 secondes

Le module afficheur LCD I2C utilise lui aussi une librairie. Comme pour le module I2C, nous l'avons déjà intégrée à Arduino augmenté. Mais si un jour vous devez utiliser Arduino en version classique, il faudra l'ajouter vous-même.

EX02.2 - Découvrons le nouveau matériel

Le matériel nouveau concerne l'afficheur LCD I2C. Comme vous le savez, la communication est un peu particulière puisqu'elle est du type I2C. Même si le shield de connexion est équipé de plusieurs prises I2C, en réalité, elles vont toutes au même endroit. La carte Arduino UNO n'a en effet qu'un seul bus I2C. Vous pouvez donc brancher votre module sur la prise I2C que vous voulez sans distinction.

MATÉRIEL NOUVEAU	BRANCHEMENT
<p>Afficheur LCD I2C</p> 	<p>- Communication : I2C - Se branche avec : 1 câble Grove sur I2C</p>  <p>X1</p>

EX02.3 - Nous allons maintenant ajouter les blocs LCD dans Ardublock

Encore une fois, plusieurs blocs sont associés à cet afficheur LCD. En plus, ce dernier est équipé d'une fonctionnalité originale puisqu'il est possible de changer la couleur de fond - d'où le nom de LCD RGB Background. Mais nous verrons cela un peu plus tard.

Découvrons les fonctions des blocs RFID

Prenez un instant pour comprendre à quoi servent les différents blocs.

BLOCS RFID	UTILITÉ
	<p>Permet d'écrire du texte</p> <p>-Imprimer : le texte à écrire</p> <p>-n° de ligne :</p> <ul style="list-style-type: none">● 0 = la première ligne● 1 = la deuxième ligne
	<p>Permet d'ajuster la couleur du fond</p> <p>-Rouge : 0 à 1000 (1000=maximum de rouge)</p> <p>-Vert : 0 à 1000 (1000=maximum de vert)</p> <p>-Bleu : 0 à 1000 (1000=maximum de bleu)</p>

Notez aussi que le pilotage de cet écran se fait ligne par ligne. Vous pouvez par exemple modifier la 2ème ligne sans toucher à la première. Le présent exemple va justement illustrer ceci.

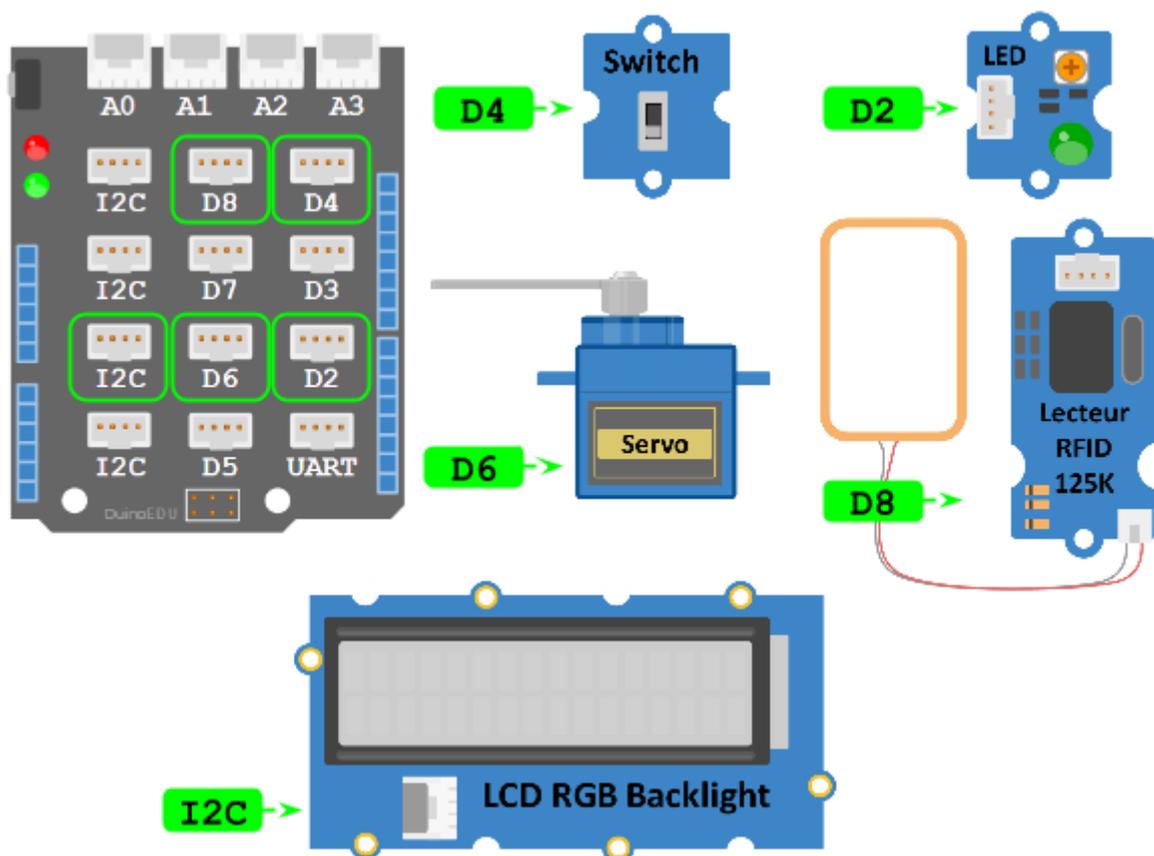
EX02.4 - Nous allons maintenant brancher notre afficheur LCD

L'afficheur LCD est un nouveau type de module. Il utilise un bus I2C. Très concrètement, il vous suffit de brancher ce dernier sur une des prises I2C et la connexion sera établie.

Retenez quand même qu'un bus I2C peut recevoir de très nombreux modules. Si par exemple il vous manquait une prise I2C, il vous serait tout à fait possible de brancher une multiprise ! C'est très pratique et cela permet de brancher plusieurs choses sur une seule prise...

Rappelez-vous, les modules précédents étaient liés à un n° de prise, les modules I2C, eux, ne sont pas liés à un numéro.

Réalisez le branchement ci-dessous.

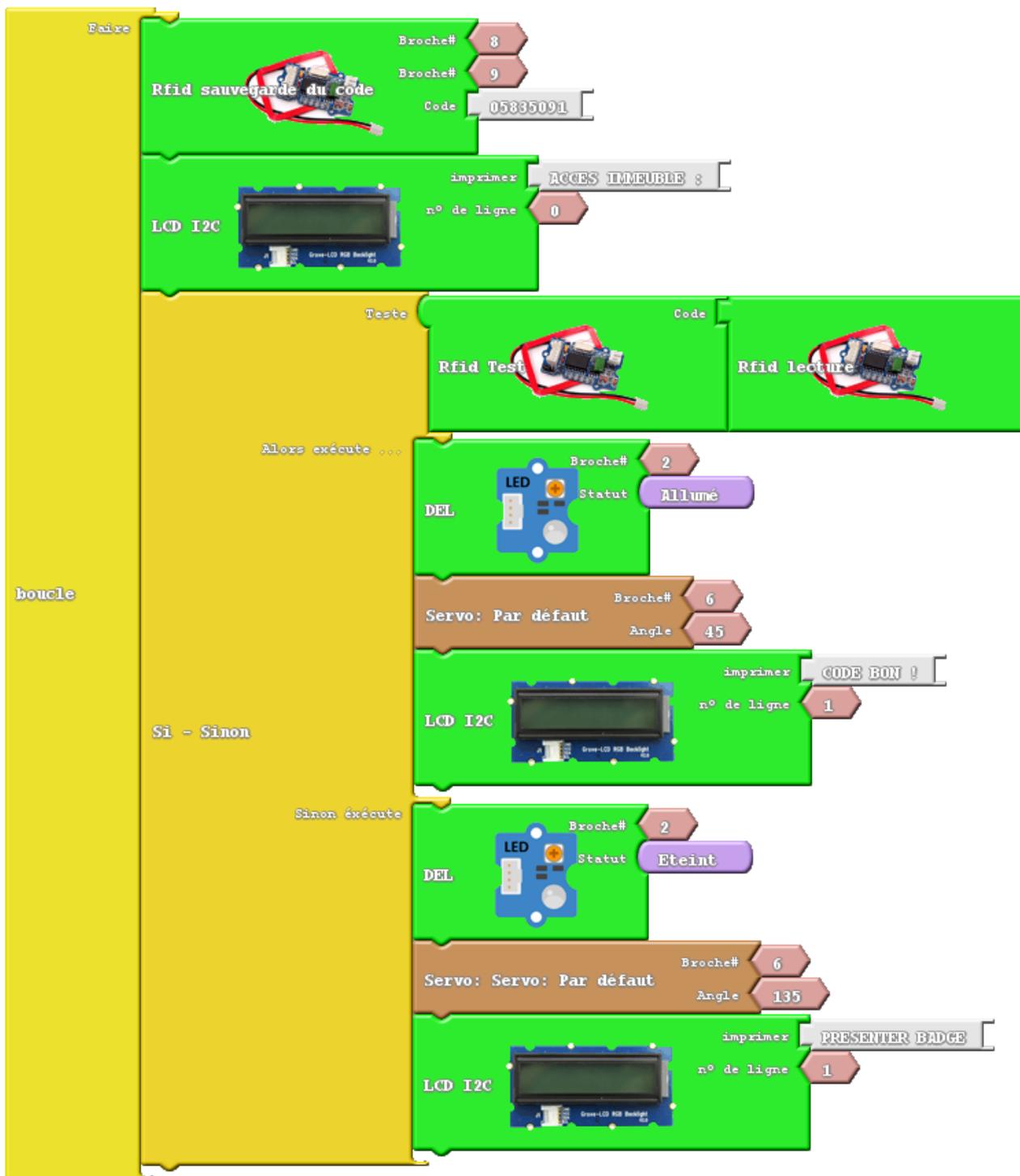


Ajoutons maintenant les blocs LCD dans Ardublock

Procédez comme ceci :

1. Placez un bloc "LCD I2C" sous le 1er bloc RFID
 - a. Imprimer = ACCES IMMEUBLE :
 - b. n° de ligne = 0 (= la 1ère ligne)
2. Placez un bloc "LCD I2C" en dessous du 1er bloc servo
 - a. Imprimer = CODE BON !
 - b. n° de ligne = 0
3. Clonez un bloc "LCD I2C" au-dessus du 2ème bloc servo
 - a. Imprimer = PRESENTER BADGE !
 - b. n° de ligne = 0

Vous devriez obtenir ceci :



Comment éviter les variations de luminosité de l'afficheur

Si votre montage est exclusivement alimenté par la prise USB de votre ordinateur, il est possible que la luminosité de l'afficheur baisse quand le servomoteur s'actionne. Cela vient du fait que les servomoteurs sont très gourmands en énergie. Pour remédier à ce problème, il faut brancher une alimentation externe sur la prise noire de la carte Arduino. Lien vers un produit qui pourrait vous convenir.

EX02.5 - Testez votre montage

Procédez comme ceci :

1. Au départ s'affiche :

ACCES IMMEUBLE

PRESENTER BADGE

2. Placez la carte ou le badge dont le code est correct à quelques centimètres de la bobine

- a. le lecteur doit lire ce code et le comparer à celui que vous avez saisi dans Ardublock
- b. la porte s'ouvre et la LED s'allume
- c. l'afficheur doit indiquer :

ACCES IMMEUBLE

CODE BON !

- d. la porte doit s'ouvrir

3. Cette situation doit rester en l'état pendant 4 secondes environ

4. Après 4 secondes (temporisation)

- a. le code lu devient périmé
- b. la LED s'éteint et la porte se ferme
- c. l'afficheur doit indiquer :

ACCES IMMEUBLE

PRESENTER BADGE

5. Placez un tag non autorisé

- a. la LED doit rester éteinte
- b. l'afficheur doit indiquer à nouveau

ACCES IMMEUBLE

PRESENTER BADGE

Nous allons maintenant ajouter 2 options :

- une liste de codes
- le rétro-éclairage rouge / vert

Passons à l'exemple suivant.

Ne débranchez rien, l'exemple 03 ne fait qu'ajouter un élément d'affichage au montage précédent.

Dans cet exemple, nous allons ajouter 2 choses différentes : la gestion de la mémoire des codes (utile dans les exemples qui suivent) et la gestion de la couleur du LCD.

EX03 - PROGRAMME EXEMPLE 03

EX03.1 - Le résultat attendu est le suivant :

Objectif du programme EXEMPLE03



- 7. état initial de notre serrure :
 - a. plusieurs identifiants RFID sont pré-enregistrés
 - b. la porte est initialement fermée
 - c. **l'afficheur affiche : "saisir code" sur la 1ère ligne en rouge**
- 8. si l'utilisateur saisit un mot de passe faux :
 - a. rien ne change
- 9. si l'utilisateur saisit le bon mot de passe :
 - a. **l'afficheur affiche : "code bon" en vert**
 - b. la porte s'ouvre pendant 4 secondes

Le module LCD dispose d'un rétro-éclairage RGB (ou Red Green Blue). La seule chose à comprendre est que le bloc n°2 de cet afficheur pilote indépendamment les 3 couleurs. Vous pouvez donc régler la quantité des 3 couleurs.

Petit rappel à lire

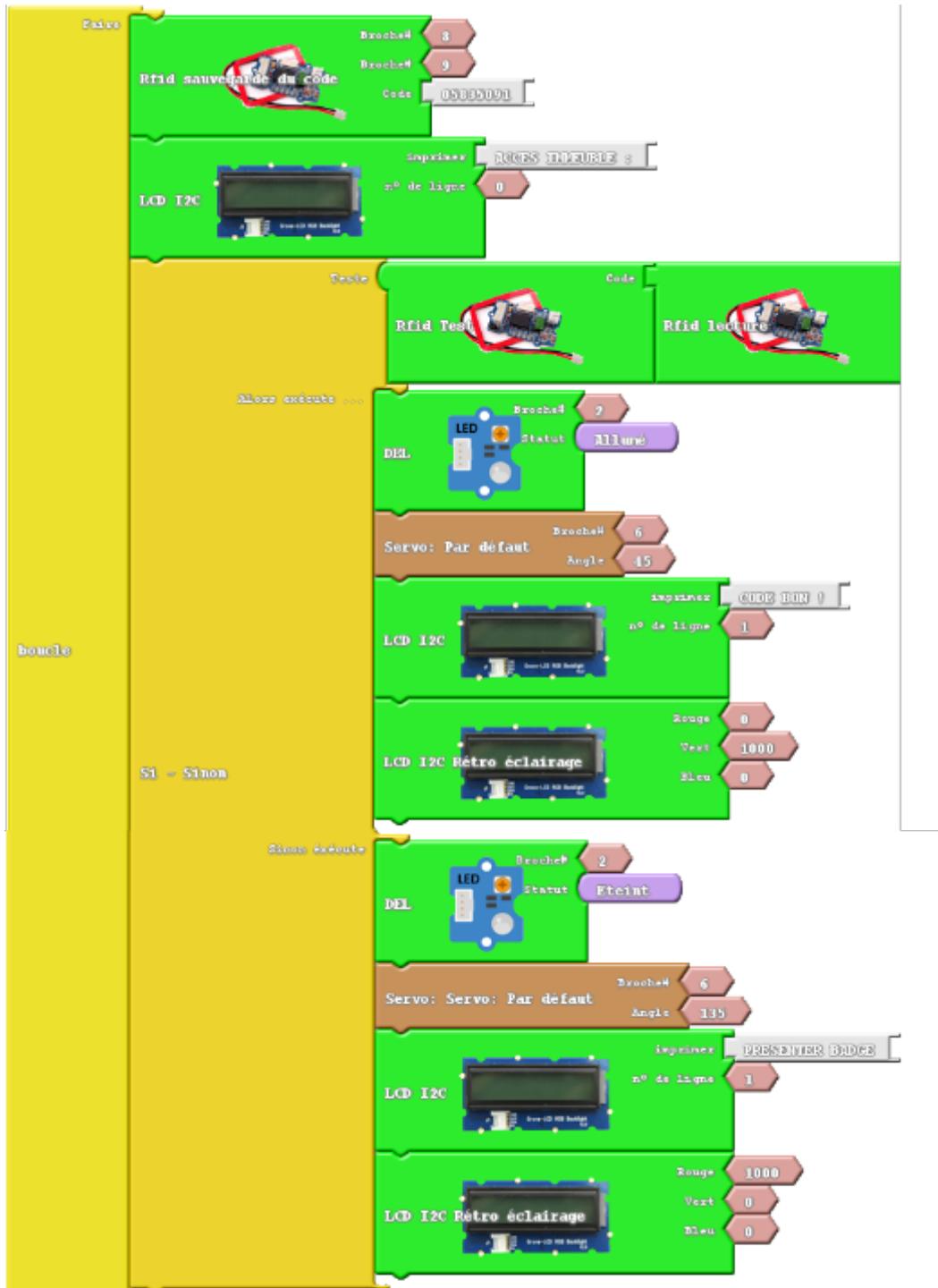
BLOCS RFID	UTILITÉ
	<p>Permet d'ajuster la couleur du fond</p> <ul style="list-style-type: none"> -Rouge : 0 à 1000 (1000=maximum de rouge) -Vert : 0 à 1000 (1000=maximum de vert) -Bleu : 0 à 1000 (1000=maximum de bleu)

Ajoutons maintenant les blocs de gestion du rétro-éclairage du LCD dans Ardublock

Procédez comme ceci :

1. Placez un bloc "LCD Rétro éclairage" sous le 2ème "LCD I2C"
 - a. Rouge = 0
 - b. Vert = 1000
 - c. Bleu = 0
2. Placez un bloc "LCD Rétro éclairage" sous le 23ème "LCD I2C"
 - a. Rouge = 1000
 - b. Vert = 0
 - c. Bleu = 0

Vous devriez obtenir ceci :



EX03.2 - Testez votre montage

Procédez comme ceci :

6. Au départ s'affiche :

ACCES IMMEUBLE

PRESENTER BADGE

7. Placez la carte ou le badge dont le code est correct à quelques centimètres de la bobine

- le lecteur doit lire ce code et le comparer à celui que vous avez saisi dans Ardublock
- la porte s'ouvre et la LED s'allume
- l'afficheur doit indiquer :

ACCES IMMEUBLE

CODE BON !

- la porte doit s'ouvrir

8. Cette situation doit rester en l'état pendant 4 secondes environ

9. Après 4 secondes (temporisation)

- le code lu devient périmé
- la LED s'éteint et la porte se ferme
- l'afficheur doit indiquer :

ACCES IMMEUBLE

PRESENTER BADGE

10. Placez un tag non autorisé,

- la LED doit rester éteinte
- l'afficheur doit indiquer à nouveau

ACCES IMMEUBLE

PRESENTER BADGE

Nous allons maintenant ajouter 2 options :

- une liste de codes
- le rétro-éclairage rouge / vert

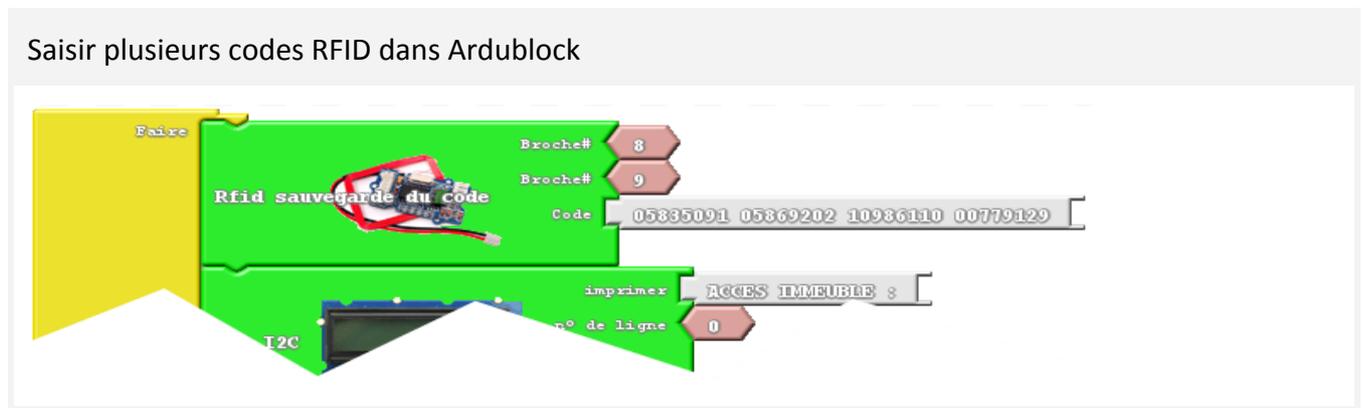
Passons à l'exemple suivant.

Avant de passer à l'exemple suivant, nous allons apprendre à créer une liste de codes RFID autorisés. En réalité, c'est assez simple dans Ardublock. Il suffit d'ajouter les codes les uns après les autres séparés par un espace. C'est tout !

Ajoutons maintenant 3 autres codes à la liste

Dans Ardublock, ajoutez les codes RFID des 3 autres tags qui vous sont fournis. Il s'agit toujours des 8 derniers caractères directement lisibles sur les cartes ou badges.

Vous devriez obtenir ceci :



Peut-être vous demandez-vous s'il est possible d'enregistrer ces badges à la volée ? La réponse est oui. Non seulement vous pouvez le faire à la main comme vu précédemment, mais vous pouvez aussi les enregistrer en mémoire RAM. Nous développerons un peu plus tard un enregistrement en ROM (EEPROM) si des demandes sont exprimées. [Pour faire une demande de modification, cliquez ici.](#) Pour l'instant, les codes sont perdus à la coupure de l'alimentation. C'est, semble-t-il, le plus pratique dans une démarche expérimentale en collège.

C'est l'objet de l'exemple suivant. Attention, il est un peu plus compliqué.

Cette partie est consacrée à l'enregistrement des codes RFID à la volée.

Enregistrer des codes RFID à la volée nécessite un changement de mode de fonctionnement. Il y aura maintenant 2 modes :

- mode **enregistrement** des codes dans la mémoire RAM
- mode **lecture et test** des codes présentés au lecteur

Le changement de mode se fera par l'intermédiaire du switch (ou commutateur). Ainsi, le système pourra fonctionner d'une manière plus proche de la réalité :

- Le **responsable** pourra passer en **mode enregistrement** pour ajouter des codes autorisés
- L'**utilisateur** restera en mode lecture et test et sera autorisé à entrer suivant son badge

EX04 - PROGRAMME EXEMPLE 05

EX04.1 - Le résultat attendu est le suivant :

Objectif du programme EXEMPLE05

1. état initial de notre serrure :
 - a. des codes RFID peuvent déjà être enregistrés (facultatif)
 - b. La porte est fermée
 - c. La LED est éteinte
2. Si le switch (commutateur) est en position 1
 - a. état initial de ce mode
 - i. Message ligne 0 : ENREGISTREMENT :
 - ii. Message ligne 1 : PRESENTER BADGE
 - iii. si un badge est présenté au lecteur
 - iv. si le code est déjà autorisé
 1. Message ligne 0 : ENREGISTREMENT :
 2. Message ligne 1 : BADGE CONNU
 3. si le code est inconnu
 4. Enregistrement du code puis
 5. Message ligne 0 : ENREGISTREMENT :
 6. Message ligne 1 : BADGE CONNU
3. Si le switch (commutateur) est en position 2
 - a. Etat initial de ce mode
 - i. Message ligne 0 : ACCES IMMEUBLE :
 - ii. Message ligne 1 : PRESENTER BADGE
 - b. si un code autorisé est présenté au lecteur
 - i. Message ligne 1 : ACCES IMMEUBLE :
 - ii. Message ligne 1 : CODE BON
 - iii. La porte s'ouvre pendant 4 secondes
 - iv. La LED s'allume pendant 4 secondes
 - c. si un code non autorisé est présenté au lecteur
 - i. pas d'action



EX04.2 - Nous allons commencer par enfermer le programme de lecture dans un sous-programme

Vous avez la possibilité de créer des sous-programmes dans Ardublock. Cela permet d'avoir un ensemble plus lisible. Dans notre cas, nous allons avoir 2 gros sous-programmes :

- un sous-programme lecture (qui reprend ce que nous venons de faire)
- un sous-programme enregistrement (ce que nous ajoutons de nouveau)

Découvrons les blocs liés à la gestion des sous-programmes

Il y a toujours 2 blocs lors de l'utilisation d'un sous-programme :

- L'étiquette qui correspond au nom du sous-programme
- Le sous-programme lui-même qui n'est à écrire qu'une seule fois

BLOCS SOUS-PROGRAMMES	UTILITÉ
<p>MENU Contrôles</p> 	<p>Permet d'appeler un sous programme</p> <p>-Cette étiquette doit juste porter le même nom que le sous-programme qu'elle doit appeler</p>
<p>MENU Contrôles</p> 	<p>Permet de construire son sous-programme</p> <p>-Tout ce qui se trouvera entre les mâchoires de ce bloc sera appelé par l'étiquette portant le même nom</p>

Prenez un bloc sous-programme

Il est dans le menu Contrôles

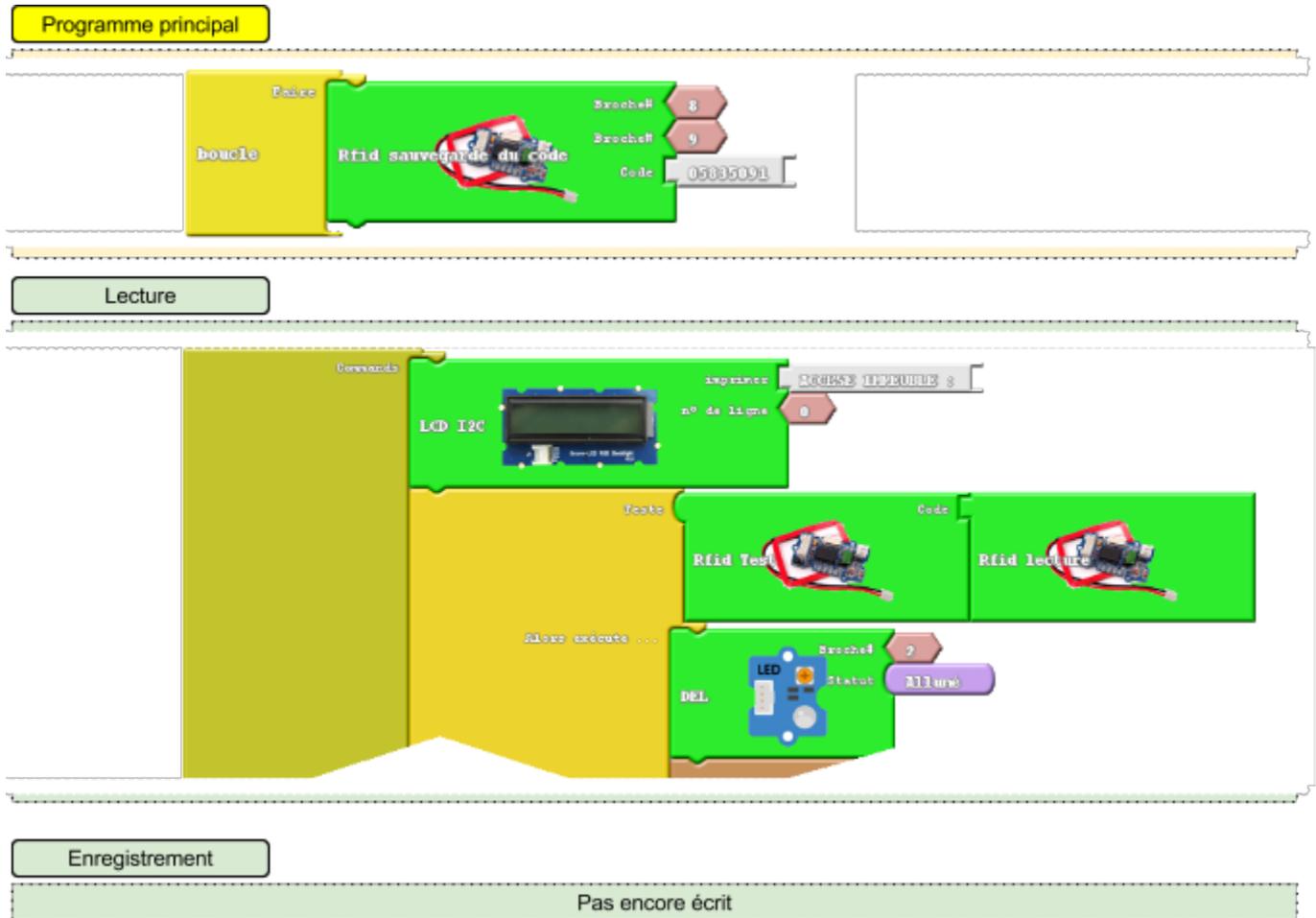
Placez-le dans une zone vide

Un sous-programme se place toujours en dehors du bloc Boucle. C'est justement son intérêt. Il est effectivement extérieur au programme principal.

Faites maintenant glisser l'ensemble de vos blocs (sauf le bloc RFID) dans la mâchoire sous-programme

Le premier bloc RFID est un bloc permettant d'initialiser le lecteur RFID (avec éventuellement des codes RFID par défaut). Il est donc commun à l'enregistrement et à la lecture. Il est donc assez logique de le lancer en haut dans le programme principal. C'est ce que l'on appelle un bloc d'initialisation ou setup. Si vous avez eu la curiosité de regarder le code généré par Ardublock, vous avez certainement remarqué un bloc de code commençant par `setup(){...}`. C'est un bloc de code qui sert justement à l'initialisation. Pour l'instant, nous n'avons pas à nous en soucier mais gardez à l'esprit que cela existe.

Vous devriez obtenir ceci :



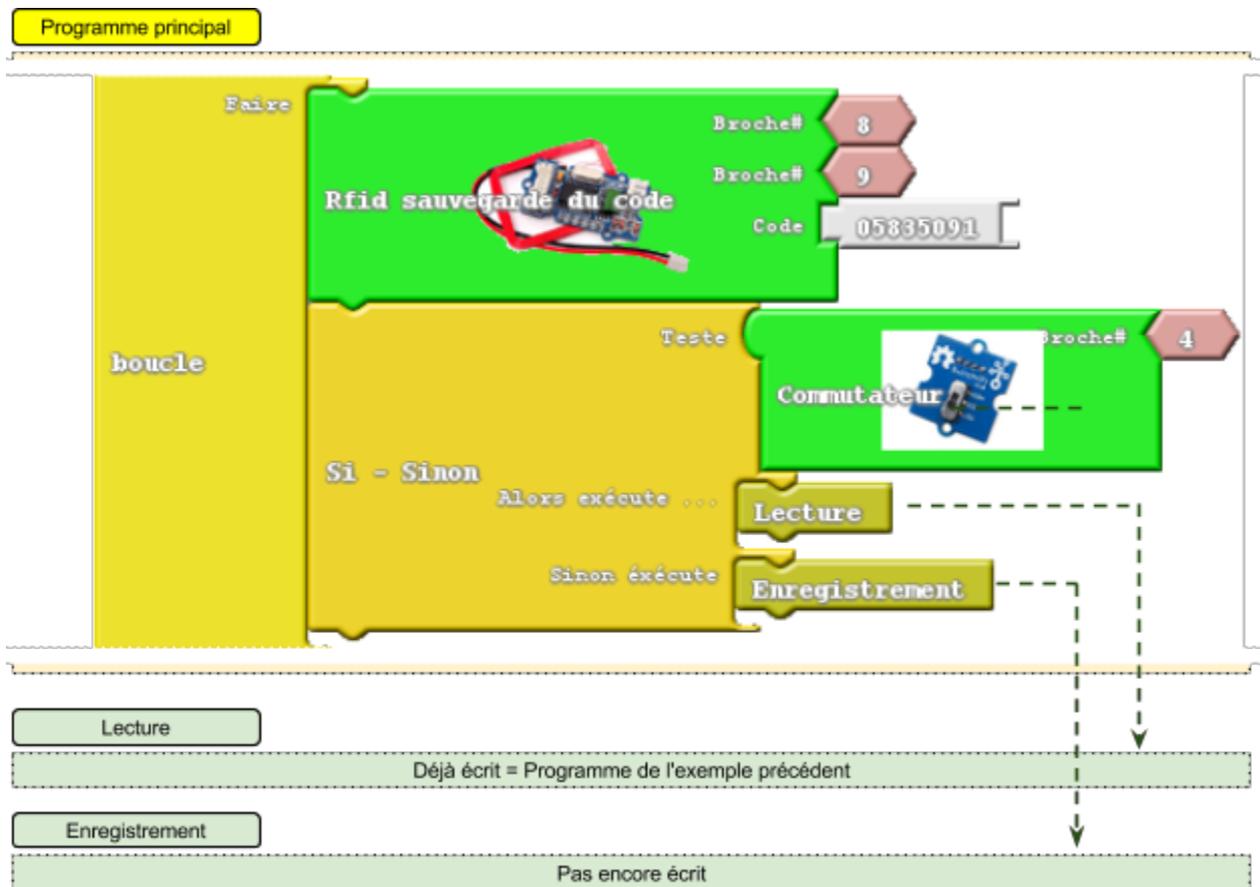
EX04.3 - Nous allons créer le nouveau programme principal

Reproduisez l'assemblage de code ci-dessous

Il s'agit d'un assemblage connu puisque nous l'avons déjà vu au moment de la lecture du commutateur (switch) pour allumer ou éteindre une LED.

Cette fois, nous allons l'utiliser pour appeler un sous-programme ou un autre. Si vous avez suivi, vous savez déjà que seules 2 étiquettes seront nécessaires :

- une étiquette **Lecture** => lance le sous-programme lecture de codes RFID
- une étiquette **Enregistrement** => lance le sous-programme enregistrement de code RFID



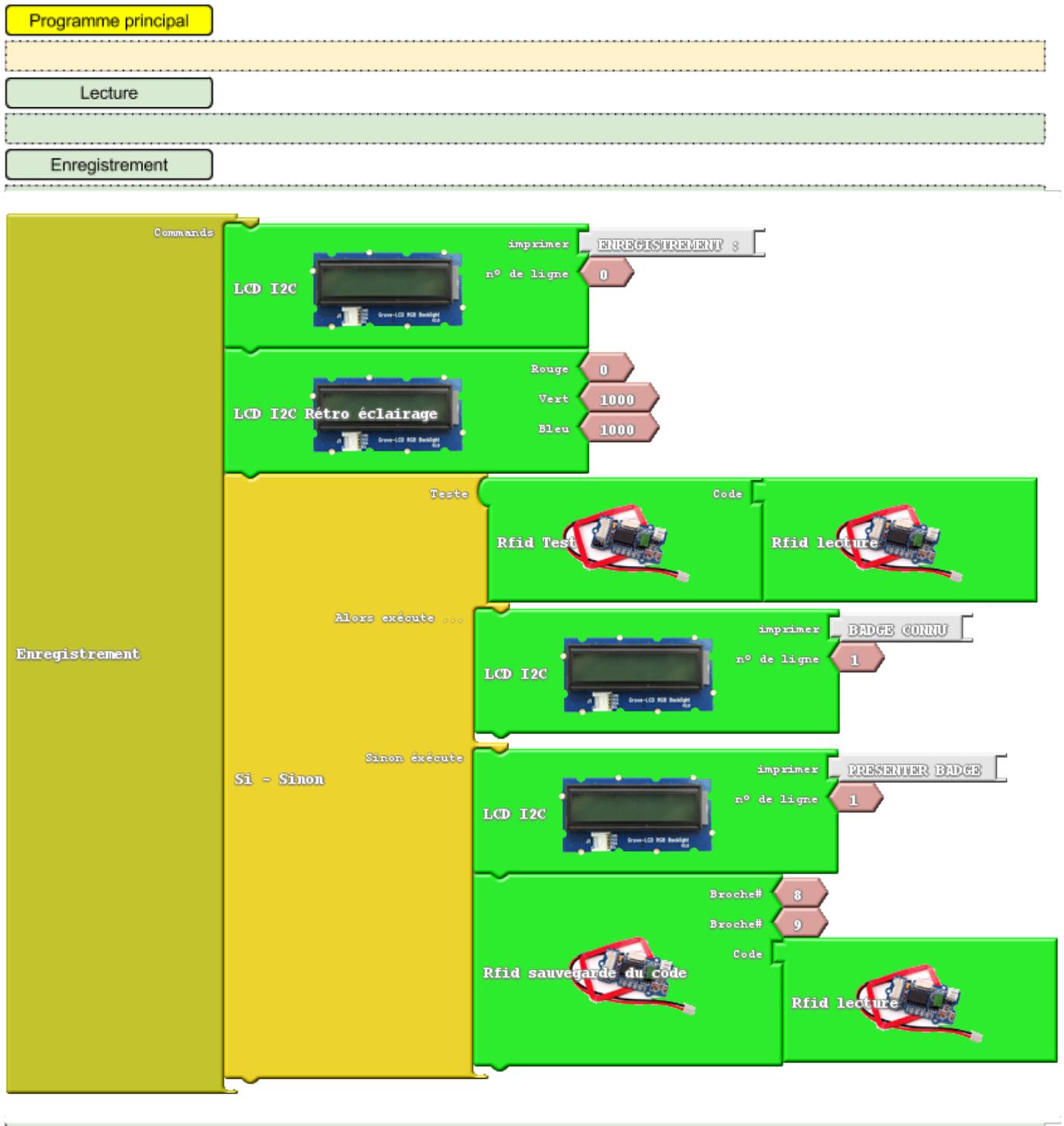
Si vous n'avez pas débranché votre commutateur (switch), il devrait déjà se trouver sur la bonne prise. Sinon, vous savez quoi faire...

EX04.4 - Nous allons créer le sous-programme Enregistrement

Reproduisez le sous-programme ci-dessous

La seule subtilité ici est que l'on emboîte un bloc de lecture RFID directement dans la mâchoire "Code" du bloc "Sauvegarde code". Ainsi, le code lu est directement enregistré dans la liste des codes autorisés.

Votre sous-programme enregistrement devrait ressembler à ceci :



Vous pouvez télécharger le fichier Ardublock ici :

<http://duinoedu.com/dl/exemples/ardublock/STARTERKITS/N2/exemple05/>

Voici une petite explication du fonctionnement du sous-programme enregistrement

Rappelons pour commencer que le sous-programme Enregistrement est appelé par l'étiquette "Enregistrement". Vérifiez que vous avez bien nommé votre sous-programme de la même façon. Si vous décidez d'un nom différent, ne mettez pas d'espace dans le nom. Prenez cette habitude.

Lors de l'appel du sous-programme le premier bloc affiche :

- "ENREGISTREMENT :" sur la première ligne

Puis le deuxième bloc produit un mélange entre le vert et le bleu :

- le fond passe en bleu-vert

Nous arrivons ensuite au test d'existence du code :

- On commence par une tentative de lecture de code avec le bloc RFID Lecture
- Ce code est ensuite envoyé au bloc tester
 - Si le code est déjà connu => On affiche BADGE CONNU
 - Si le badge est inconnu => On passe à l'enregistrement

L'enregistrement du code est la partie la plus intéressante ici. Le bloc de lecture RFID envoie le résultat (donc le code RFID) au bloc Sauvegarde du code RFID. Ce dernier code est en réalité lié à une fonction de sauvegarde du code. Cette fonction s'assure bien entendu que le code n'est pas déjà enregistré. Le test "Si... Sinon" dont il est question ici ne sert donc pas à éviter les doublons mais sert à gérer 2 affichages différents :

- PRESENTER BADGE et
- BADGE CONNU

Sans ce test il ne nous était pas possible de gérer cet affichage.

EX04.5 - Testez votre montage

Procédez comme ceci :

1. Placez votre switch en position "LOW" = "En bas" => vous êtes en mode enregistrement
 - a. placez le badge à enregistrer sur le lecteur
 - b. dans tous les cas le message BADGE CONNU doit apparaître
2. Placer votre switch en position "HIGH" = "En haut" => vous êtes en mode lecture
 - a. placer le dernier badge sur le lecteur
 - b. vous devriez retrouver le fonctionnement de l'exemple précédent

Nous arrivons au terme des exemples liés à la gestion des tags RFID. Vous pouvez diversifier vos tags en y ajoutant des pièces RFID. Une pièce RFID peut être collée sur différents supports pour créer une collection d'objets RFID pour des projets originaux.

Nous allons maintenant passer à la gestion d'un clavier type Keypad passif. C'est un module assez difficile à gérer car il est totalement passif et c'est avec pas moins de 8 fils qu'il faut le connecter. Il en existe bien sûr des actifs mais ils sont beaucoup plus chers et la gestion d'un tel module est l'occasion d'aborder de nouveaux aspects d'Arduino qu'il est bon de connaître.

Dans cet exemple, vous allez utiliser un keypad 16 touches pour saisir un code.

Le keypad que nous allons utiliser est un keypad passif. En d'autres termes, il n'a pas d'électronique et peut être comparé à un ensemble de poussoirs placés les uns à côté des autres et connectés en lignes et colonnes. Ce keypad n'est pas un produit Grove mais vous allez voir qu'il est possible de l'utiliser sans problème à condition de comprendre comment sont organisés les connecteurs Grove.

Si vous avez déjà regardé votre keypad, vous avez certainement constaté qu'il faut 8 fils pour le connecter. C'est beaucoup ! Mais là où il est imbattable, c'est sur son coût. Il est un bon exemple de module purement passif.

EX05 - PROGRAMME EXEMPLE 05

EX05.1 - Le résultat attendu est le suivant :

Objectif du programme EXEMPLE05



10. état initial de notre serrure :
 - a. un mot de passe de 4 chiffres est pré-enregistré : "1234"
 - b. l'afficheur indique qu'il faut saisir un mot de passe
 - c. le voyant (LED) est éteint
11. l'utilisateur saisit un mot de passe faux :
 - a. rien ne change
12. l'utilisateur saisit le bon mot de passe :
 - a. la LED s'allume pendant 4 secondes
 - b. on revient à l'état initial

EX05.2 - Découvrons le nouveau matériel

Tous les modules Grove utilisent un seul connecteur ! Mais Grove ne propose pas encore de clavier matriciel. D'ailleurs, aucun fournisseur ne propose tous les modules. C'est l'occasion idéale pour vous montrer comment utiliser un matériel qui non seulement n'est pas de marque Grove et qui en plus ne respecte pas le standard Grove (standard Grov : un module = un connecteur).

MATÉRIEL NOUVEAU		BRANCHEMENT
Keypad 16 touches		<ul style="list-style-type: none"> - Communication : parallèle passif - Se branche avec : 4 adaptateurs Grove - Occupe réellement : 8 prises  <p style="text-align: right;">X4</p>

EX05.3 - Débranchez le matériel précédent

EX05.4- Nous allons maintenant brancher notre keypad

Grove propose des adaptateurs de toutes sortes. Il en existe des mâles, des femelles et même des adaptateurs à souder. Pour le Keypad, c'est des adaptateurs mâles qu'il nous faut. Comme le keypad se pilote avec 8 fils et qu'un connecteur Grove contient toujours 2 signaux, il nous en faudra 4 !

Retenez ceci :



Les modules simples ont cette particularité :

1. ils ont l'inscription "NC" en face du fil blanc (Non Connecté)
2. ils n'occupent qu'un seul connecteur
3. il est possible de brancher un module sur le connecteur suivant
4. les servomoteurs se pilotent toujours avec un seul fil

Les modules évolués ont cette particularité :

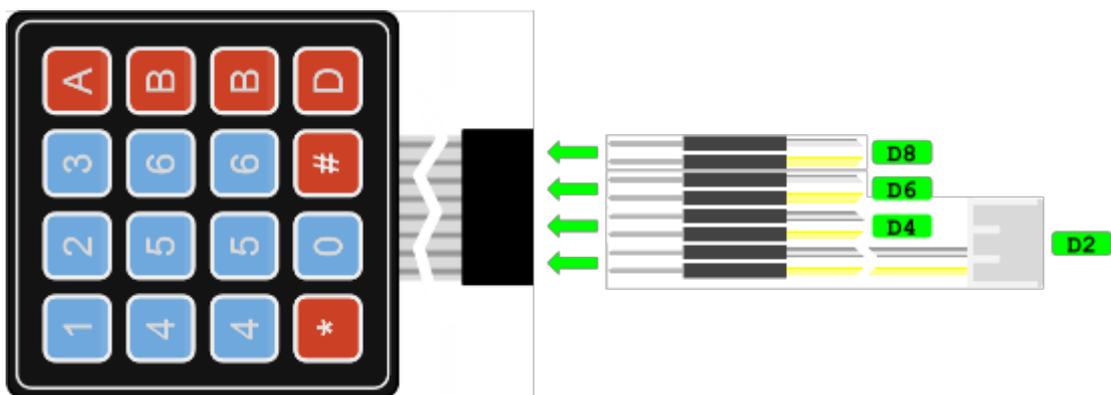
1. ils n'ont pas l'inscription "NC" en face du fil blanc
2. ils occupent 2 connecteurs (même si cela ne se voit pas)
3. il faut sauter un connecteur

Branchez les 4 adaptateurs mâle Grove comme ceci

Ce n'est pas très compliqué. Le premier signal est toujours le fil jaune. Le 2ème signal est toujours le blanc. Branchez donc votre keypad en respectant l'alternance : jaune / blanc / jaune / blanc...
Rappelez-vous, "NC" est inscrit lorsque le fil blanc n'est pas utilisé.

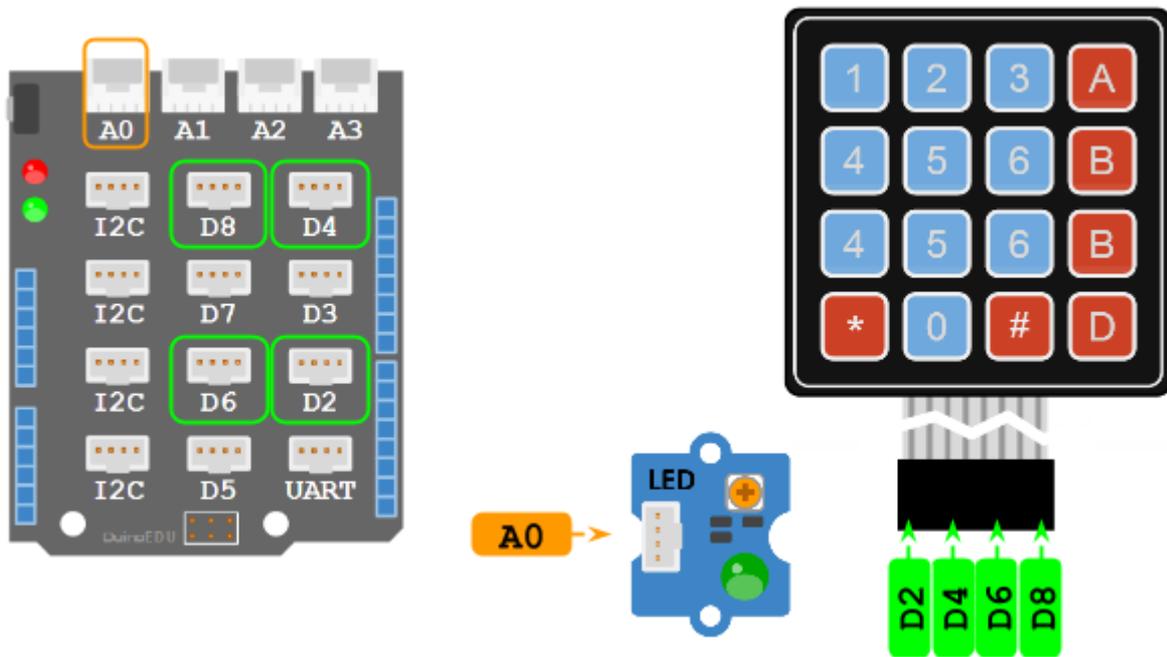
Vous devriez obtenir ceci :

VOTRE KEYPAD ÉQUIPE DE SES 4 ADAPTATEURS



Branchez ensuite votre keypad et une LED

Si ce keypad était un keypad Grove, il n'aurait pas l'inscription "NC" en face des fils blancs. Vous en concluez certainement qu'il faut le brancher en sautant une prise à chaque fois. Pour la LEDE, vous n'avez pas le choix, il faut passer par les prises Ax.



Je pensais que A0, c'était de l'analogique ?

- Oui, A0 peut être utilisée comme entrée analogique !
- Mais c'est 1 **entrée numérique** comme une autre !



D2, c'est l'entrée 2, D3, c'est l'entrée 3 mais A0, c'est quoi ?

- Retenez ceci :
 - **D2** = entrée/sortie **02**
 - **D3** = entrée/sortie **03**
 - ...
 - **D8** = entrée/sortie **08**
 - ...
 - **A0** = entrée/sortie **14**
 - **A1** = entrée/sortie **15**
 - **A2** = entrée/sortie **16**
 - **A3** = entrée/sortie **17**
 - **A2** = entrée/sortie **18**
 - **A3** = entrée/sortie **19**

EX05.5 - Nous allons construire notre programme dans Ardublock

Vous en avez maintenant l'habitude. Il faut rechercher dans les menus les modules nécessaires et les placer dans la zone de travail. Il faut ensuite modifier les options nécessaires.

Découvrons les fonctions du bloc keypad

Prenez un instant pour comprendre à quoi servent les différents blocs.

BLOCS RFID	UTILITÉ
	Permet de saisir des codes -StartPin : première prise utilisée -EndPin : dernière prise utilisée -Affichage port série : affichage d'infos de debugage sur le port série (permet de suivre les saisies clavier...)
	Permet de lire la touche saisie -Il est ainsi possible d'enregistrer des touches dans une variable...
	Permet de tester un mot de passe à la volée -Pwd : mot de passe (doit contenir seulement les caractères disponible sur le keypad)

Assemblez maintenant les blocs dans Ardublock

L'assemblage est assez simple. On place le bloc d'initialisation en haut en prenant garde de bien définir les prises du keypad. Le bloc Mot de passe est justement fait pour s'emboîter dans la mâchoire Teste d'un bloc "Si... Sinon...". Ensuite, on allume la LED ou non. Il faut aussi faire attention au n° de broche. A0 est aussi le trou 14 sur la carte Arduino.



Dans cet exemple, nous allons simplement compléter le programme précédent afin qu'il soit complet.

Si vous avez testé le programme précédent, vous avez certainement remarqué que le code reste valide tant qu'une nouvelle touche n'est pas appuyée. Ce n'est pas le fonctionnement habituel d'un accès d'immeuble mais il est possible de faire mieux. Dans un premier temps, nous allons simplement ajouter un afficheur LCD. Ensuite nous verrons une astuce pour gérer une temporisation.

EX06 - PROGRAMME EXEMPLE 06

EX06.1 - Le résultat attendu est le suivant :

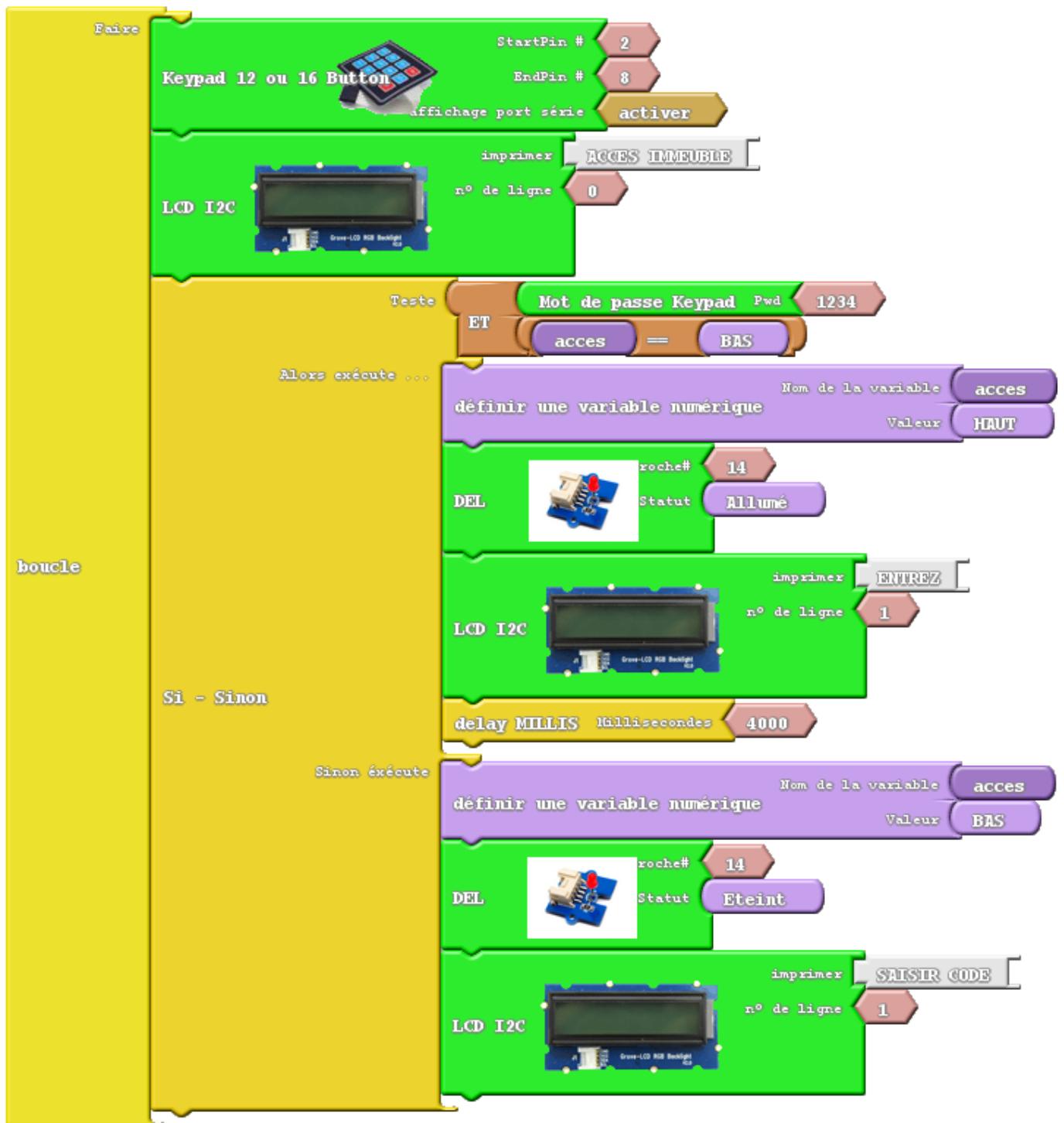
Objectif du programme EXEMPLE06

13. état initial de notre serrure :
 - a. un mot de passe de 4 chiffres est pré-enregistré : "1234"
 - b. l'afficheur indique qu'il faut saisir un mot de passe
 - c. le voyant (LED) est éteint
14. l'utilisateur saisit un mot de passe faux :
 - a. rien ne change
15. l'utilisateur saisit le bon mot de passe :
 - a. la LED s'allume pendant 4 secondes
 - b. l'afficheur indique : ENTRER
 - c. cet état reste pendant 4 secondes
16. on revient à l'état initial



EX06.2 - Voici un exemple plus complet

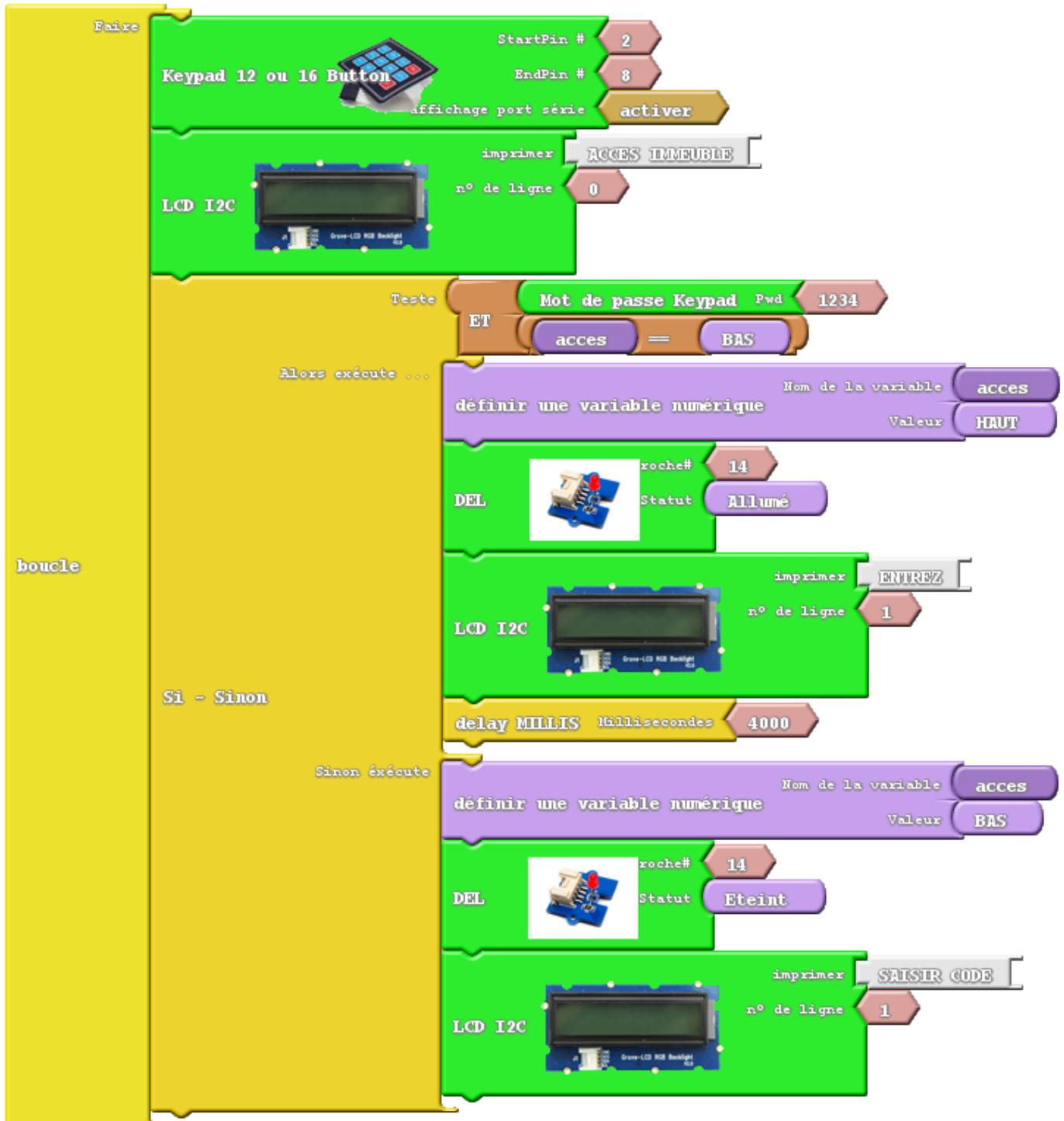
Nous n'avons fait qu'ajouter le bloc Afficheur LCD. Comme nous l'avons déjà utilisé, nous ne donnons pas plus de précisions. Vous pouvez bien sûr ajouter la gestion de la couleur d'arrière plan.



EX06.3 - Nous allons ajouter une gestion de temporisation

Il y a encore beaucoup de choses à découvrir dans Ardublokc. Notamment les variables (assez peu utilisées en collège) ; les opérations logiques ; les temporisations... Nous vous proposons de compléter le programme précédent avec une gestion de temporisation qui justement utilise tout cela à la fois.

Voici un exemple de gestion de la temporisation



Vous devriez être en mesure d'en comprendre le principe général. Il s'appuie sur l'utilisation d'une variable nommée "acces" dont la valeur doit être "BAS" pour justement permettre l'accès. Comme cette variable est mise dans un état "HAUT" lors d'une saisie correcte, le programme sort tout seul de la boucle et ne reste plus bloqué dans la partie "Alors exécute..." comme c'était le cas précédemment.

Le bloc "**Delays MILLIS**" est une temporisation pour laisser le temps à l'utilisateur de passer la porte. Un autre bloc utile est le bloc "**ET**". Il permet d'obliger les 2 tests à être vrais en même temps.

Voici en résumé l'utilité des nouveaux blocs :

BLOCS RFID	UTILITÉ
	<p>Permet de définir une variable</p> <ul style="list-style-type: none"> -Ici, son nom est accès -Sa valeur est HAUT
	<p>Permet de faire appel à une variable</p> <ul style="list-style-type: none"> -La variable doit avoir été définie
	<p>Permet de tester si une variable à la bonne valeur</p> <ul style="list-style-type: none"> == (égale) != (différent)
	<p>Permet de tester simultanément 2 conditions</p> <p>Exemple de test simultané</p> <ul style="list-style-type: none"> -Il faut que le mot de passe soit correct ET que la variable accès soit égale à BAS pour exécuter la partie haute du bloc
	<p>Fait une pause</p> <ul style="list-style-type: none"> -4000 = 4 secondes

Nous arrivons à la fin de ce petit tutoriel. Nous espérons qu'il vous aura permis de vous faire idée concrète de ce que permet Ardublock. N'oubliez pas de consulter le code Arduino. Ardublock n'est qu'un générateur de code, peut-être un jour passerez-vous à l'écriture du code Arduino. Dans ce cas précis, le passage se fera en douceur...

■ EX06.4 - Vous avez un problème ?

N'hésitez-pas à nous contacter en cas de problème. Vos remarques permettront aussi d'apporter des améliorations à ce document et à Ardublock.

N'hésitez-pas aussi à partager vos découvertes et vos créations. La force de l'OpenSource est le partage.

Demander une amélioration dans ce document

http://duinoedu.com/mail_modif_document.html

Demander de l'aide

<http://duinoedu.com/contacterduinoedu.html>

Bon Arduino à Vous !

