Conception d'une plateforme éducative de robot mobile pour Arduino programmable en scratch

Module d'Initiation à la Recherche Parcours ISTR

Encadrant : Berthou Pascal

Dovonou Jason
Grézit Daphnée
Ibanez Emmanuel

Table des matières

Remerciements

Nous tenons à remercier toutes les personnes qui ont contribué à l'élaboration et au succès de ce projet.

Nous adressons tout d'abord nos remerciements à notre encadrant, M. Pascal Berthou de l'Université Paul Sabatier, qui nous a aidé et orienté dans ce projet. Son écoute et ses conseils nous ont permis de mieux cibler et comprendre notre sujet.

Nous voulons également remercier M. Jérôme Rabayrol qui nous a aidé et conseillé dans la conception mécanique de notre robot. Il a été d'une aide précieuse dans les moments délicats.

I. Présentation du sujet

a. Objectifs du TER

Lors de ce TER, nous avons eu pour objectif de développer un kit dans le but de permettre l'initiation de jeunes enfants à la robotique. Nous avons commencé par nous approvisionner en matériaux de base afin de pouvoir visualiser un éventuel rendu. Nous avions à notre disposition un châssis, deux moteurs à courant continu comme actionneurs et différents capteurs. Par choix nous nous sommes équipés en plateformes Arduino qui sont un moyen peu cher et robuste de mettre en œuvre une commande. Cependant, le langage de programmation utilisé par cette carte est un peu trop complexe pour le public visé.

Afin de permettre une utilisation simple et éducative du codage appliqué en temps réel sur un robot, nous avons choisi d'utiliser le langage scratch, mis en place par le MIT, qui est un langage graphique adapté aux enfants. Basé sur des blocs à combiner, il permet la réalisation d'algorithmes simples et le pilotage d'objets graphiques de l'environnement. Toutefois, scratch n'est pas compatible avec l'environnement Arduino.

L'idée était de combiner les deux approches précédentes pour réaliser une plateforme de robot simple à monter et simple à commander. Nous avons donc réalisé un travail allant jusqu'à la réalisation du prototype et du scénario pour enfant. Il nous a alors fallu augmenter l'environnement Scratch développé en Java pour lui ajouter des briques de commande Arduino.

Séquencement du travail :
_ Etat de l'art arduino/scratch,
_ Développement des briques de commande pour moteurs à courant continu,
_ Développement de briques de commande pour capteurs,
_ Installation d'une alimentation électrique embarquée,
_ Mise en œuvre d'un scénario ludo-éducatif.

b. Mise en œuvre du scénario ludo-éducatif

Afin de faciliter la mise en œuvre de notre robot, il nous a fallu établir un scénario ayant pour but de définir les futures fonctions de celui-ci. Nous nous sommes projetés afin d'imaginer quel pourrait être son rôle et quels jeux les enfants pourront s'amuser à faire avec. Nous avions décidé de commencer par faire un suivi de route dans un labyrinthe or après réflexion nous avons choisi de faire en premier lieu un simple suivi de ligne pour que notre robot suive un chemin prédéfini sans s'éloigner de celui-ci. Puis, nous avons voulu intégrer un bouton poussoir pour démarrer notre robot, et quelques LED pour indiquer si oui ou non le suivi de ligne restait correct. Pour aller plus loin, nous aurions voulu rajouter un capteur pour éviter les obstacles.

II. Etat de l'art

Afin de mieux comprendre la raison pour laquelle nous avons travaillé avec certains logiciels au détriment d'autres, nous choisissons de commencer par présenter l'ensemble des travaux qui ont déjà été réalisés au niveau logiciel, tout en parlant des éventuels avantages, ou inconvénients de chacun. Puis nous détaillerons les raisons pour lesquelles nous avons choisi un logiciel en particulier.

Au départ, il nous était demandé de travailler en particulier avec Arduino et en parallèle avec Scratch, or nous avons réussi, grâce à des recherches sur le sujet, à trouver d'autres logiciels pouvant être utilisé comme alternative à Scratch.

a. Arduino

Arduino a été mis en œuvre par une équipe de développeur composée de Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti. Ce projet met en œuvre une carte électronique programmable et un logiciel multiplateforme, et a pour but de créer facilement des systèmes électroniques. Elle possède une communauté active et est un logiciel open-source.

Afin de passer directement à la partie programmation de notre projet, nous avons pu nous procurer une carte Arduino Uno prête à l'emploi. Nous nous sommes ainsi concentrés sur la mise en œuvre des blocs de commande et sur les différents tests à réaliser pour « prendre en main » notre carte.

Exemple de programme Arduino qui permet de faire clignoter une LED :

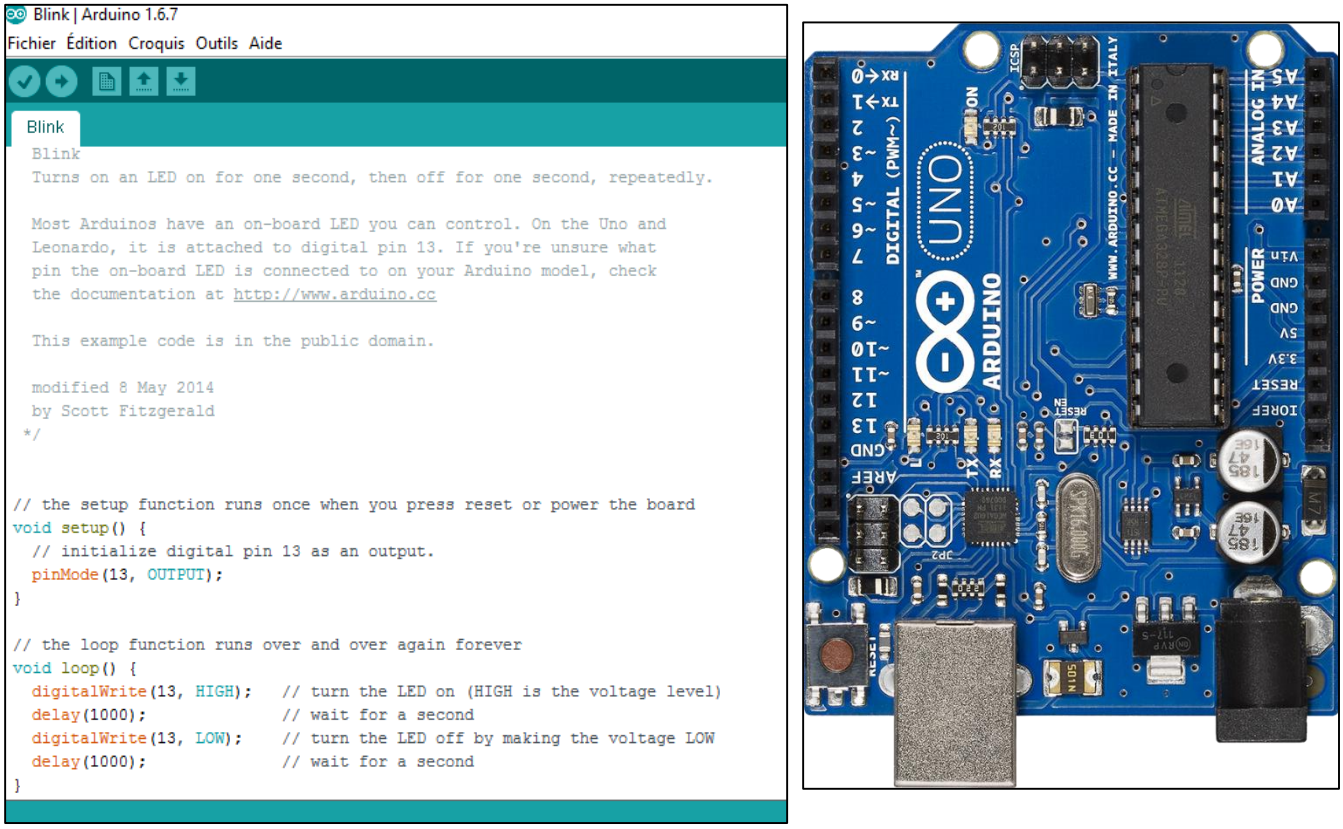


Figure 1 : Interface Arduino et Carte Arduino Uno

En quelques chiffres, on peut voir sur internet qu'une simple carte Arduino Uno ne coûte que 25 euros ce qui reste très abordable, le logiciel et l'assistance quant à eux sont gratuits.

Pour ce qui est du projet Arduino, on peut dire que l'idée est d'utiliser la carte comme un macro-composant dans des applications de prototypage électronique. Le concepteur n'a plus qu'à développer des interfaces et programmer le macro-composant pour réaliser son application.

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle.

On remarque qu'Arduino a de nombreux avantages : il est multiplateforme, c'est-à-dire qu'il tourne aussi bien sous Windows, que sur Mac OS ou Linux. Il possède de nombreuses librairies disponibles avec diverses fonctions implémentées. Il a également la chance d'être très bien documenté, on trouve beaucoup de conseils et tutoriels en ligne, qui permettent autant de prendre en main le logiciel et la carte que de résoudre d'éventuels problèmes. On sait qu'il existe des « shield » (cartes supplémentaires se connectant sur le module Arduino) qui sont très utile pour augmenter les possibilités de fonctionnement de la carte.

Pour ce qui est de la carte, elle s'interface au PC par l'intermédiaire de sa prise USB et s'alimente par le jack d'alimentation (utilisation autonome) mais peut être alimentée par l'USB.

Le développement sur Arduino est très simple, on commence par coder l'application en utilisant le langage Arduino, basé sur les langages C/C++. On relie ensuite la carte Arduino au PC et on transfère le programme sur la carte. Le logiciel de programmation des modules Arduino est une application Java multiplateformes servant d'éditeur de code et de compilateur. Il peut également transférer le firmware au travers de la liaison série.

Pour finir, voici une courte liste des applications possibles que l'on peut réaliser grâce à Arduino et qui nous permettrons d'intéresser les enfants au domaine de la programmation :

- Contrôler des appareils domestiques
- Commander un robot
- Permettre à un ordinateur de communiques avec une carte électronique et des capteurs
- Télécommander un appareil mobile
- …

b. Scratch

Scratch est un projet du groupe Lifelong Kindergarten développé auprès du laboratoire Media du MIT. Le projet est initié en 2003 et la première version date de 2006. Scratch est un logiciel libre créé pour initier les enfants au principe de l'informatique, il peut être utilisé sous Mac OS, Windows et Linux.

Scratch est un nouveau langage de programmation, c'est un langage graphique, qui facilite la création d'histoires interactives, de dessins animés, de jeux, de compositions musicales, de simulations numériques et leurs partages sur Internet. Scratch est spécialement conçu pour les enfants de 8 à 16 ans mais il est utilisé par des personnes de tout âge. Scratch est utilisé dans 150 pays différents et est disponible dans 40 langues. Il est utilisé dans des écoles pour permettre aux élèves d'appréhender les principes de bases algorithmiques et informatiques. Les élèves apprennent avec Scratch de l'école primaire à l'université.

Le slogan de Scratch est « Imagine-Programme-Partage ». En effet, le partage est un principe fondamental de la pédagogie de Scratch, il permet de réutiliser des projets existants pour les développer ou en inventer de nouveaux. Aujourd'hui, on recense plus de 13 millions de projets partagés et le nombre d'utilisateurs enregistrés est de 7,5 millions et ne cesse d'augmenter.

Ce logiciel est dynamique car permet de modifier le code du programme qui est en cours d'exécution. L'interface Scratch utilise 10 catégories de blocs de codage (cf Figure 2), chaque bloc

graphique permettant d'effectuer une action précise. La version 2.0 de l'éditeur Scratch est désormais disponible online et offline.

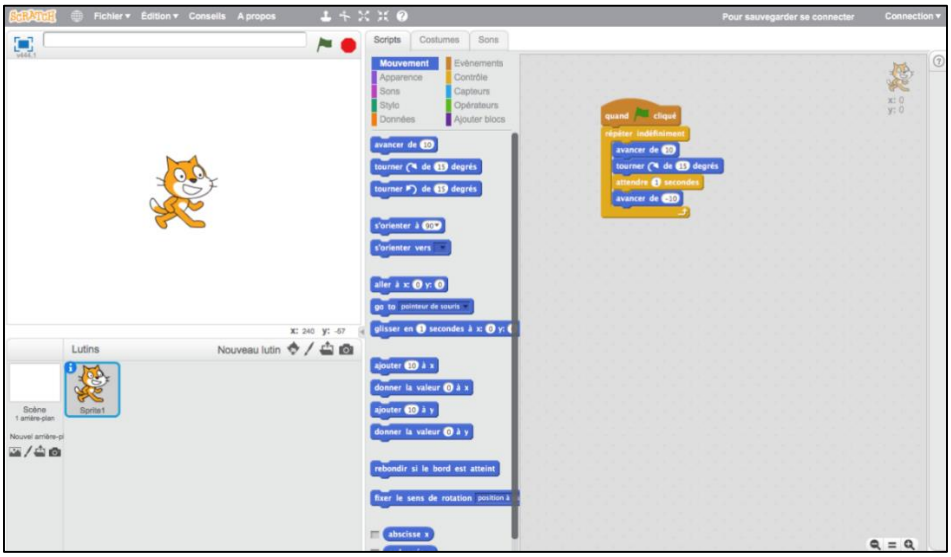


Figure 2 : Interface Scratch

c. Blockly@rduino

Blockly est un langage de programmation visuel et se présente sous la forme d'un puzzle dont chacune des pièces constitue une fonction pour former une action finale. Mis en place par Google, ce projet est accessible directement à partir d'une page web, où l'internaute peut glisser, déposer et assembler des blocs afin de constituer son programme. Le code généré est du JavaScript mais peut également être exporté en Python. Google explique que Blockly peut également être couplée à des applications existantes pour en augmenter les fonctionnalités. Il permet aussi aux codeurs débutants d'éviter les erreurs de syntaxe en générant automatiquement du code correct. L'outil reste cependant réservé aux petits scripts. Relativement ludique, Blockly s'inspire du projet App Inventor initié par Google et basé sur le langage Scratch.

Blockly@rduino est un programme web permettant la programmation graphique pour Arduino et sa traduction en code, directement à copier-coller dans l'IDE Arduino. Blockly@rduino possède une bibliothèque de blocs pour toutes les interfaces et modules, particulièrement adaptée à la découverte des microcontrôleurs et aux expérimentations de systèmes automatisés au collège ou au lycée. Il est également possible de l'héberger sur un serveur, et ainsi y faire accéder de nombreux postes en pointant vers ce fichier « index.html ».

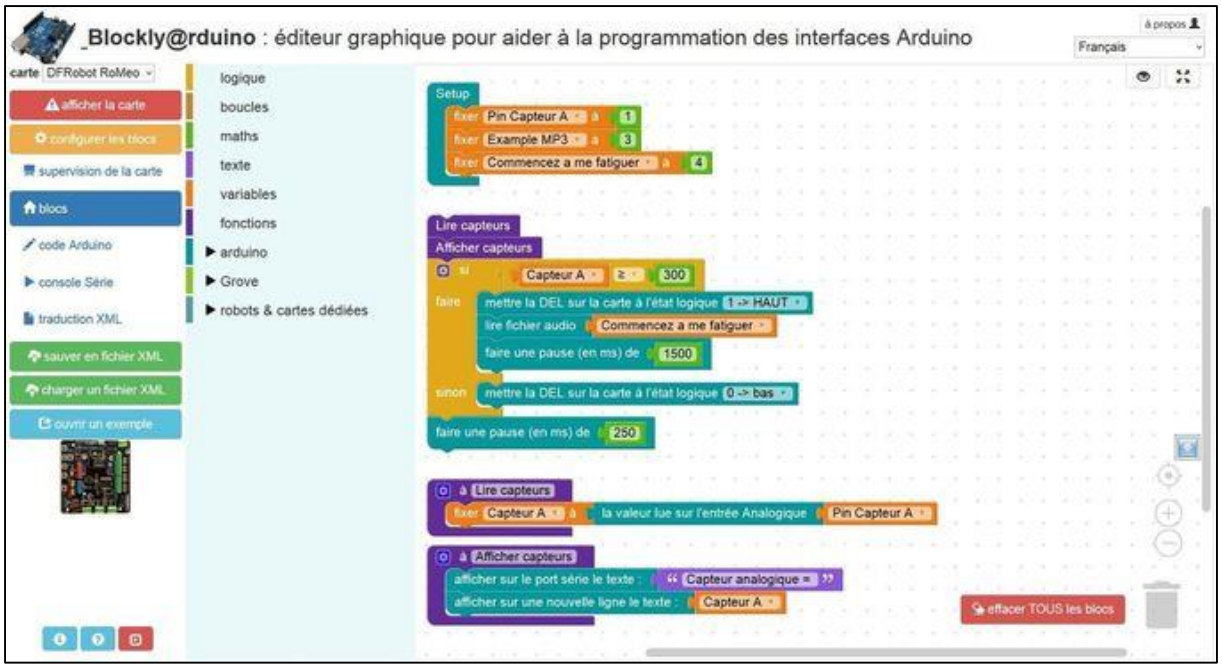


Figure 3 : Interface Blockly@rduino

d. S4A

S4A (Scratch For Arduino) est une modification de Scratch qui permet une programmation simple de l'IDE (Integrated Development Environment) Arduino. Cette extension a été développée par Media Lab Citilab en Espagne aidé par le Smalltalk Programming Group.

S4A permet de programmer des instructions que la carte d'interfaçage Arduino peut exécuter et ce depuis une GUI (Graphical User Interface) Scratch (cf Figure 3) à laquelle les créateurs ont ajouté des fonctions spécifiques à la carte Arduino.

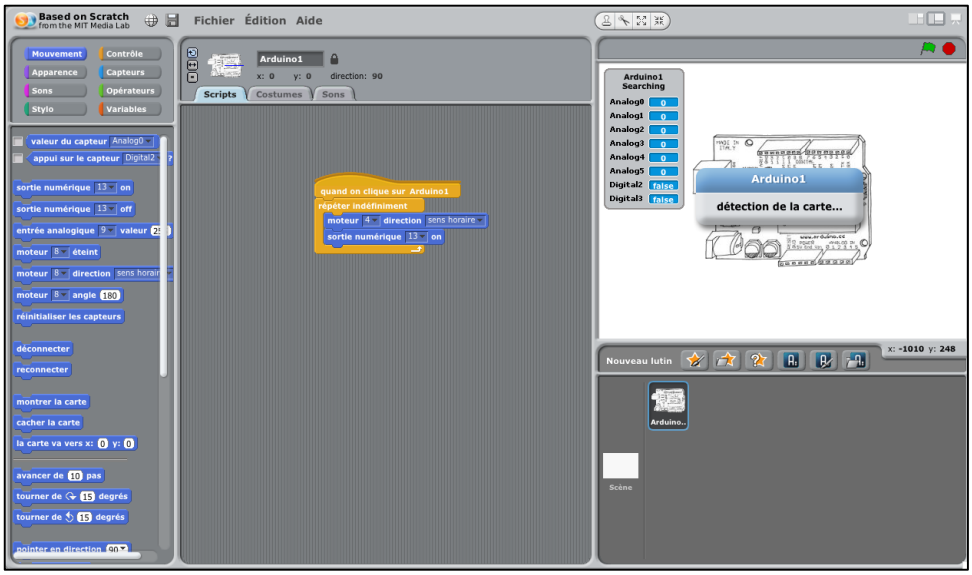


Figure 4 : Interface S4A

Le principe d'écriture d'un programme avec Scratch et S4A consiste à emboîter des blocs classés en code couleur qui correspondent à des instructions de programme : fonctions, variables, opérations, lecture de fichiers importés (images, sons). S4A présente donc de nouveaux blocs destinés uniquement à une programmation interactive entre les capteurs et les actionneurs via une carte Arduino. L'objectif principal de ce projet est d'attirer le plus de gens possible vers le monde de la programmation.

S4A fonctionne avec Arduino Diecimila, Duemilanove et Uno. Il nous permet aussi de contrôler autant de cartes qu'il y a de ports USB. L'utilisation d'un PicoBoard avec une carte Arduino est également prise en charge.

S4A est rétro-compatible avec Scratch, on peut donc l'utiliser pour ouvrir des projets Scratch. Cependant, nous ne serons pas en mesure de partager nos projets sur le site Web de la communauté Scratch étant donné que cela va à l'encontre des conditions d'utilisation. Il faut prendre en compte le fait que cette compatibilité ne fonctionne qu'en sens unique de sorte qu'il nous est impossible d'ouvrir un projet S4A à l'intérieur du Scratch originale.

Le logiciel communique avec une carte Arduino par une liaison série (via un câble USB) de manière à ce que l'ordinateur qui exécute S4A puisse lire les entrées ou écrire sur les sorties d'une carte Arduino. Il est nécessaire pour cela que la carte Arduino exécute un programme spécifique, le firmware de S4A, qui gère en permanence (toutes les 75 ms) cette communication avec l'ordinateur. La carte Arduino quant à elle, est gérée comme un simple périphérique d'entrée/sortie. Il reste possible de gérer une carte sans fil en connectant un module RF sur celle-ci, on prendra comme exemple Xbee.

e. S2A :

S2A (Scratch To Arduino) est une extension de Scratch 2.0. Il utilise la même interface que Scratch (cf Figure 2).

Figure 5 : Interface S2A

Pour piloter les entrées/sorties de l'interface Arduino par le logiciel S2A, il suffit de lancer simultanément le logiciel de base ainsi qu'un autre logiciel, nommé s2aio (cf Figure 6), qui va ainsi nous permettre d'aborder la programmation pour des publics de tous les niveaux.

Figure 6 : S2AIO

Le logiciel Scratch2 va activer des broches (ou PIN) et des cartes Arduino en entrée ou sortie pour nous permettre de contrôler nos montages. Le programme communique avec la carte à travers le logiciel s2aio, va tourner en tâche de fond. Il faut savoir que S2A fonctionne avec Arduino Léonardo, Mega (2560), Micro, Nano, Uno. Il ne permet malheureusement pas de faire fonctionner la carte électronique de manière autonome car il nécessite une connexion de la carte à l'ordinateur, afin de permettre la communication à travers s2aio.

f. Ardublock

Ardublock est un plugin qui s'ajoute à l'IDE d'Arduino qui offre la possibilité de programmer des blocs de fonctions. Il permet de téléverser vers l'Arduino les blocs de fonctions assemblés. Il est compatible avec tous les systèmes d'exploitation. L'interface d'Ardublock est facile à prendre en main mais toutefois elle peut l'être beaucoup moins pour des enfants.

L'interface d'Ardublock est assez basique mais facile à manipuler, les blocs sont classés par types. Tout comme scratch for Arduino le principe est d'emboiter des blocs correspondants à des composants (LED, capteur, moteur), des instructions (IF, SWITCH, etc) et des boucles (For, do while, while). Le but reste évidemment de réaliser un programme codé en Arduino par l'intermédiaire de ces blocs. Ainsi l'implémentation est facilitée pour les débutants.

Le logiciel nous permet de nous déconnecter de l'ordinateur une fois que le programme préalablement téléversé a été exécuté. Cette indépendance est le principal avantage d'Ardublock par rapport aux autres logiciels. En revanche l'interface d'Ardublock n'a jamais été manipulée par les enfants et n'est pas très attractive.

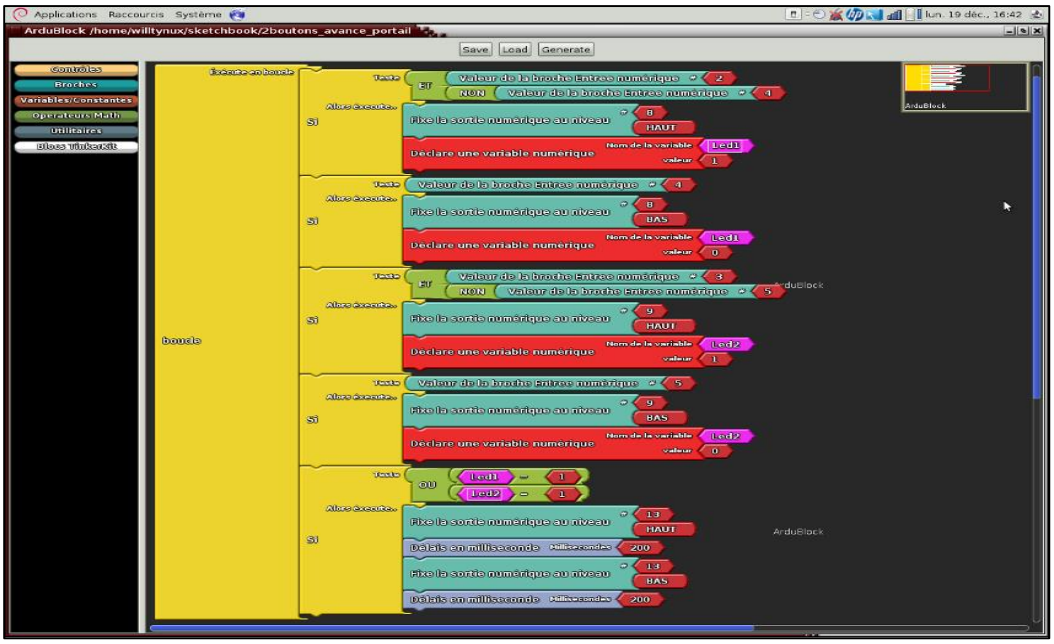


Figure 7 : Interface Ardublock

g. mBlock

mBlock est un logiciel de programmation basé sur Scratch 2.0, celui-là même qui, initié il y a 13 ans, a permis à de nombreux enfants de prendre leurs marques dans le développement informatique et de programmer des jeux vidéos et des animations interactives.

Le code Open Source Scratch a été conçu pour faciliter la programmation des cartes Arduino de manière graphique et interactive, et sa modification a permis à Makeblock de développer une nouvelle version du logiciel de base, appelé mBlock. Il est considéré comme une version personnalisée de Scratch qui peut facilement interagir avec des modules électroniques de l'environnement Arduino.

Pour programmer notre carte Arduino, il n'y a plus besoin de lancer d'application supplémentaire contrairement à S2A. On pourra directement téléverser notre programme dans la carte Arduino pour rendre le système autonome. En effet, on peut dire que la principale innovation de ce logiciel est la possibilité de générer le code Arduino de votre programme graphique. Mais il faut faire attention car on ne peut pas générer le code Arduino d'un programme s'il va contenir des éléments virtuels.

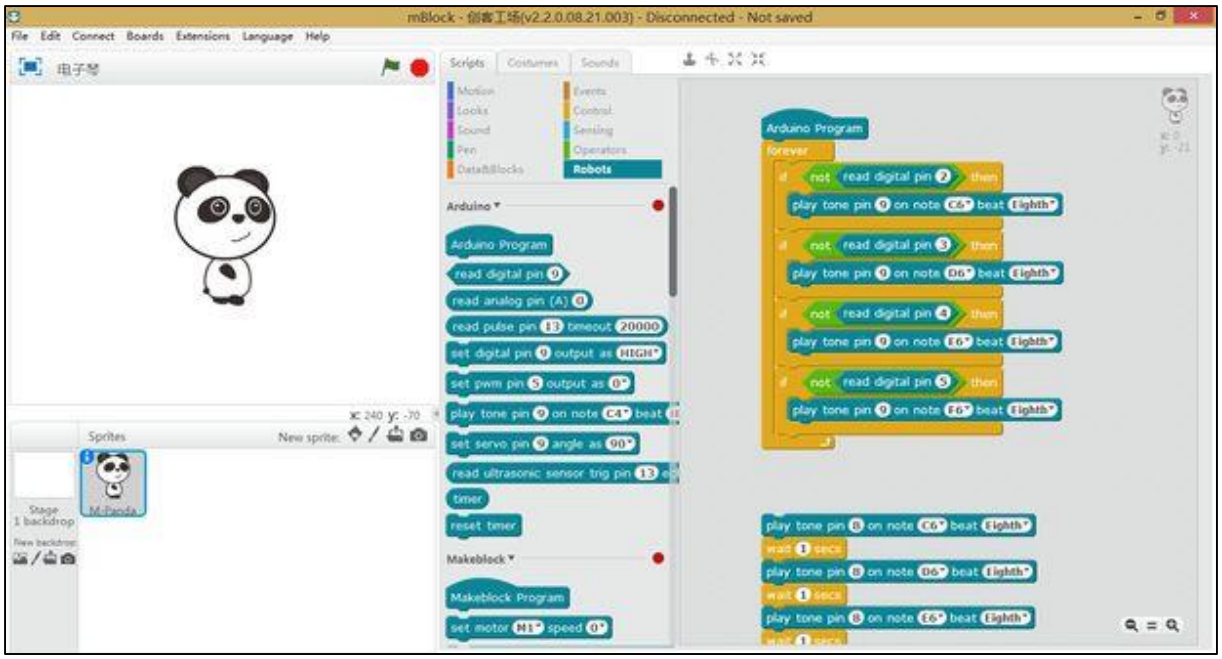


Figure 8 : Interface mBlock

h. Comparaison des logiciels

	Avantages	Inconvénients
S4A	- Rétro-compatible avec Scratch, - Contrôle de plusieurs cartes, - Interface un peu différente de Scratch mais grosse resemblance des blocs, - Fonctionne avec la carte Arduino Uno	- Intéraction S4A/Arduino toutes les 75 ms : pas de possibilité de projet autonome, obligation de connexion pc pour faire fonctionner la carte
S2A	- Interface identique à Scratch, - Fonctionne avec la carte Arduino Uno	- Pas de possibilité de projet autonome, obligation connexion pc pour faire fonctionner la carte
Ardublock	- Projet autonome, - Traduit les blocs en code Arduino, - Plug in à ajouter à Arduino, - Assez développé	- Interface compliquée pour les enfants - Obligation de faire des blocs simple, impossible de réaliser un code trop complexe
mBlock	- Même interface graphique que Scratch, - Possibilité de rendre la carte Aduino Uno autonome, car on peut envoyer le code directement sur l'IDE Arduino, - Très simple d'utilisation pour les enfants	- Très peu développé pour Arduino et peu d'information pour le développer
Blockly@rduino	- Interface directement en ligne, donc pas besoin d'installation - Possibilité d'obtenir un code à insérer dans l'IDE Arduino	- Obligation de faire des blocs simple, impossible de réaliser un code trop complexe

Etant dans l'obligation de réaliser un robot totalement autonome, c'est-à-dire sans aucune attache avec un ordinateur, nous avons été dans l'obligation de ne pas retenir les logiciels S4A et S2A qui ont pour spécifications d'avoir une interaction régulière (exemple, toute les 75ms) avec le pc d'où le fait qu'il est impossible de mettre en place un projet autonome avec ces deux logiciels. Dans un second temps, nous nous sommes intéressés à mBlock qui nous a semblé plutôt adapté à notre projet tout en restant en corrélation avec l'âge des futurs utilisateurs. Malheureusement, ce logiciel n'est que très peu développé pour Arduino et nous n'avons pas trouvé assez d'information pour continuer à le développer. Nous nous sommes donc tourné vers Ardublock et Blockly@rduino qui nous ont paru abordable, autant pour nous que pour des enfants.

III. Développement

Tout d'abord, nous avons commencé par faire fonctionner les différents composants que nous avions en langage C avec les librairies Arduino. Nous avons également pris en main le logiciel Scratch en réalisant des exemples de blocs afin d'en comprendre le fonctionnement. L'essentiel du travail consistait en la compréhension des diverses datasheet pour pouvoir utiliser ces mêmes composants. Une fois les datasheet comprises, il a fallu s'occuper de la connectique tout en faisant attention aux valeurs des résistances utilisées pour ne pas détériorer les composants.

Enfin nous avons réalisé le codage en C, cela a nécessité quelques recherches afin de connaître les fonctions spécifiques associées à la manipulation de chaque composant. Du fait notre familiarité avec la programmation en langage C, cette partie ne nous a pas véritablement posé problème.

Une fois la première partie de programmation terminée, nous nous sommes attaqués au développement de la programmation par blocs de fonctions. Nous avons naturellement commencé par la solution apportée par le logiciel Ardublock car elle nous semblait être la plus explicite pour le langage Arduino. Les blocs proposés nous ont alors permis de faire fonctionner tous les composants à l'exception des moteurs. En effet, les blocs permettant le contrôle des moteurs n'étaient pas directement disponibles. Les blocs proposés permettaient uniquement de commander les moteurs adaptés à la version précédente du Shield, c'est-à-dire le Shield V1, qui n'est plus disponible sur le marché et qui est différent du Shield en notre possession s'apparentant à un Shield V2. Par conséquent, nous avons dû développer nous-même nos blocs moteurs.

Nous avons donc effectué des recherches pour développer notre propre bloc moteur. Cependant les programmes Ardublock sont implémentés en Java et nous sommes novices dans ce langage. Il nous a fallu modifier des lignes de code corrigées qui ne fonctionnaient pas entièrement. La dernière étape pour obtenir nos blocs moteurs était de rajouter les lignes de code dans les programmes correspondant.

Pour ce faire nous avons installé Linux pour manipuler les programmes plus facilement. Nous avons installés Git, OpenJDK, Openblocks et Maven pour construire les blocs et compiler les programmes. Cela nous a pris beaucoup de temps car nous n'avions que quelques bases avec Linux, mais le résultat était à la hauteur de nos espérances. En effet après plusieurs compilations les nouveaux blocs moteurs dont nous avions besoin étaient intégrés dans Ardublock.

Après des recherches approfondies, nous avons trouvé plusieurs tutoriels nous détaillant la

méthode à utiliser pour réaliser nos fonctions. Le tutoriel que nous avons choisi d'utiliser était assez synthétique et destiné aux développeurs, d'où le fait qu'il devait être réalisé sous Linux. Les étapes majeures consistaient en :

- le téléchargement du code source,
- l'installation de Maven, un logiciel destiné à la production de projets logiciel,
- l'implémentation le code du nouveau composant et l'ajouter dans le répertoire,

ardublock-master/ardublock-master/src/main/java/com/ardublock/translator/block/

- le placement une image de taille 80x80 pixels et l'ajouter dans le répertoire,

ardublock-master/ardublock-master/src/main/resources/com/ardublock/block

- la déclaration le nouveau composant dans le fichier XML,

ardublock-master/ardublock-master/src/main/resources/com/ardublock/block/ardublock.xml

- la modification les fichiers de mapping, de description et de traduction,

ardublock-master/ardublock-master/src/main/resources/com/ardublock/block/block-mapping.properties
ardublock-master/ardublock-master/src/main/resources/com/ardublock/block/ardublock.properties
ardublock-master/ardublock-master/src/main/resources/com/ardublock/block/ardublock_fr.properties

- la compilation.

La réalisation fut assez difficile car, en plus du fait que c'était notre première utilisation de ce logiciel, il était nécessaire de télécharger diverses extensions pour réaliser la compilation. Après plusieurs tentatives de compilations, nous avons finalement réussi à générer le projet. Cela nous a permis d'obtenir un fichier JAR directement exécutable après la fin de la compilation. Ce fichier comporte les blocs de bases ainsi que les nouveaux blocs permettant de commander les moteurs.

En vue de rendre notre robot autonome, nous avons utilisé une pile 9 Volt, embarquée sur le robot afin de donner à celui-ci une capacité de mouvement indépendante d'une source de tension fixe tel qu'USB.

Afin de permettre à tout un chacun de réaliser notre robot, nous avons choisi de mettre en place un tutoriel détaillant chaque étape de notre travail, et allant de l'installation des logiciels aux montages tout en passant par la mise en œuvre des blocs sous différents logiciels.

IV. Tutoriels

Voici un court aperçu du travail à effectuer dans le but de mener à bien ce tutoriel :

- Commencer par installer les logiciels désirés,
- Se procurer les composants et faire les montages nécessaires,
- Tester des programmes simples sur vos montages pour valider ceux-ci,
- Mettre en œuvre les algorithmes de suivi de ligne et réaliser les blocs correspondant sur, au choix, Ardublock ou Blockly@rduino,

Votre suivi de ligne sera complet lorsque les blocs moteur, led, bouton poussoir et capteur infrarouge seront opérationnels et formeront un bloc uni qui commandera le robot. Pour faciliter la réalisation de ce projet nous mettons à la disposition de tous, les éléments que nous avons nous-même utilisé sur :

https://www.dropbox.com/sh/874z33q2d3hnyvk/AADGyUGZGZNAW2qU5qlKO_44a?dl=0

a. Installation des logiciels

1- Installation du logiciel Arduino

Pour télécharger le logiciel Arduino, allez sur la page officielle https://www.arduino.cc/en/Main/Software et téléchargez le logiciel en choisissant le système d'exploitation de votre machine ; Windows, Linux et Mac sont supportés par ce logiciel. Il est possible de choisir des versions plus anciennes du logiciel dans l'onglet Previous Releases.

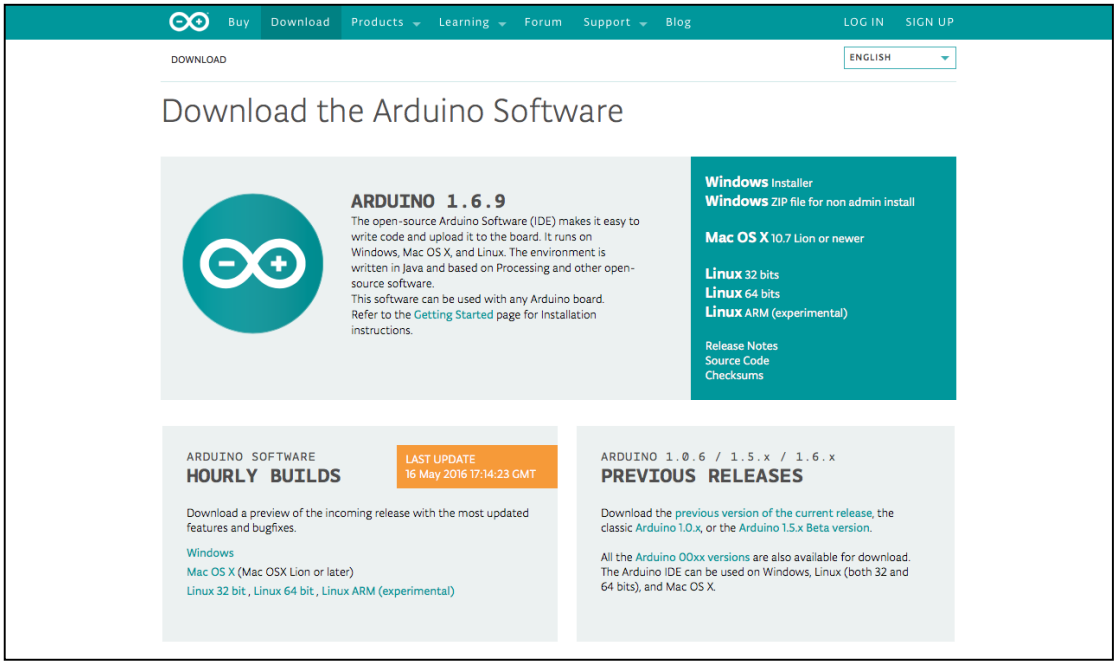


Figure 9 : Site internet d'Arduino

1) Sous Windows :

- Cliquez sur le lien Windows et le fichier apparaît. Enregistrez-le.
- Une fois le téléchargement terminé, exécutez le fichier et installer le logiciel Arduino.

2) Sous Mac OS :

- Cliquez sur le lien Mac OS X et un fichier **.app** apparaît. Enregistrez-le.
- Une fois le téléchargement terminé, vous pouvez déplacer l'application en la glissant dans le dossier « Applications ».

3) Sous Linux :

Vous avez le choix :

- Installez le logiciel en téléchargeant le fichier à partir du lien Linux correspondant à votre version.
- Installez le logiciel en recherchant "arduino" dans la logithèque.
- Installez le logiciel à partir de la ligne de commande suivante :

sudo apt –get install arduino

Vous pouvez modifier la langue dans le menu Preferences.

2- Installation du plugin Ardublock

Pour installer le plugin Ardublock, vous devez au préalable avoir installé le logiciel Arduino. Une fois l'IDE d'Arduino installé, téléchargez Ardublock à partir du lien :

https://www.dropbox.com/s/tzs2i6jogh0dlau/ardublock-all.jar?dl=0

Une fois le fichier .jar téléchargé, ouvrez le logiciel Arduino. Allez dans le menu « Préférences », où vous pouvez voir s'afficher une fenêtre avec l'emplacement du carnet de croquis.

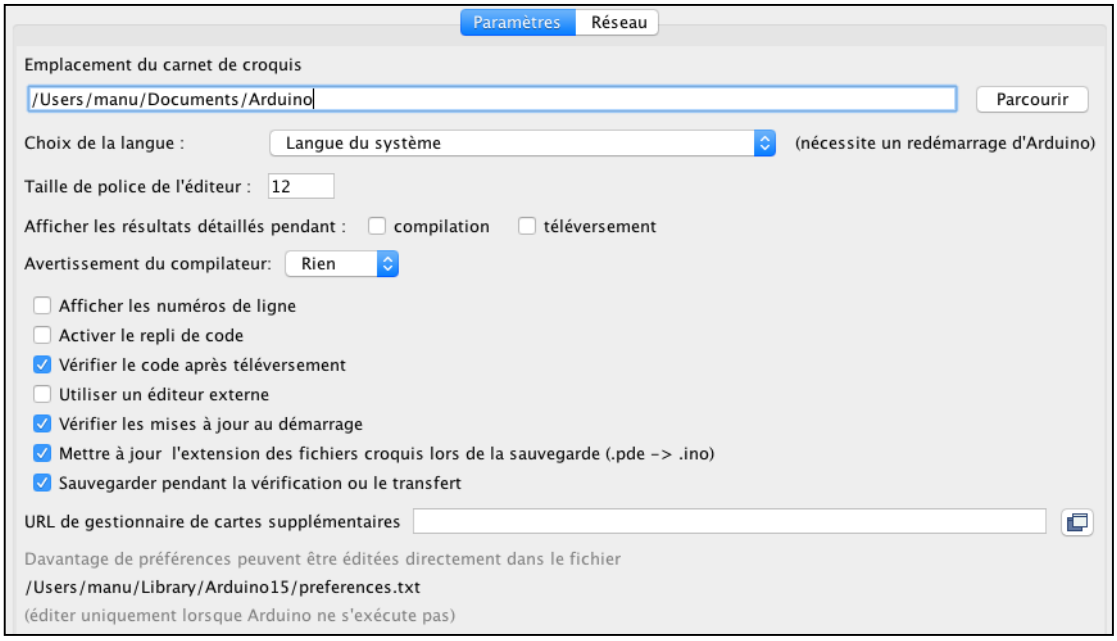


Figure 10 : Interface pour installer Ardublock

A partir de votre ordinateur, rendez-vous dans le dossier indiquez (ici : /Users/manu/Documents/Arduino). A l'intérieur de ce dossier, créer un dossier "tools" s'il n'y en pas. Puis, à l'intérieur du dossier "tools" créez un dossier "ArduBlockTool". A l'intérieur de ce dossier, créez un dossier "tool" et copiez à l'intérieur de "tool" le fichier .jar téléchargé.

Le fichier .jar a normalement été copié dans le dossier:

- Sous Windows : C:\Users\"nom_utilisateur"\Documents\Arduino\tools\ArduBlockTool\tool\
- Sous Mac OS : /Users/"nom_utilisateur"/Documents/Arduino/tools/ArduBlockTool/tool/
- Sous Linux : /home/"nom_utilisateur"/sketchbook/tools/ArduBlockTool/tool/

Vous pouvez à présent relancer le logiciel Arduino et constatez que Ardublock est bien installé et fonctionne depuis le menu « Outils ».

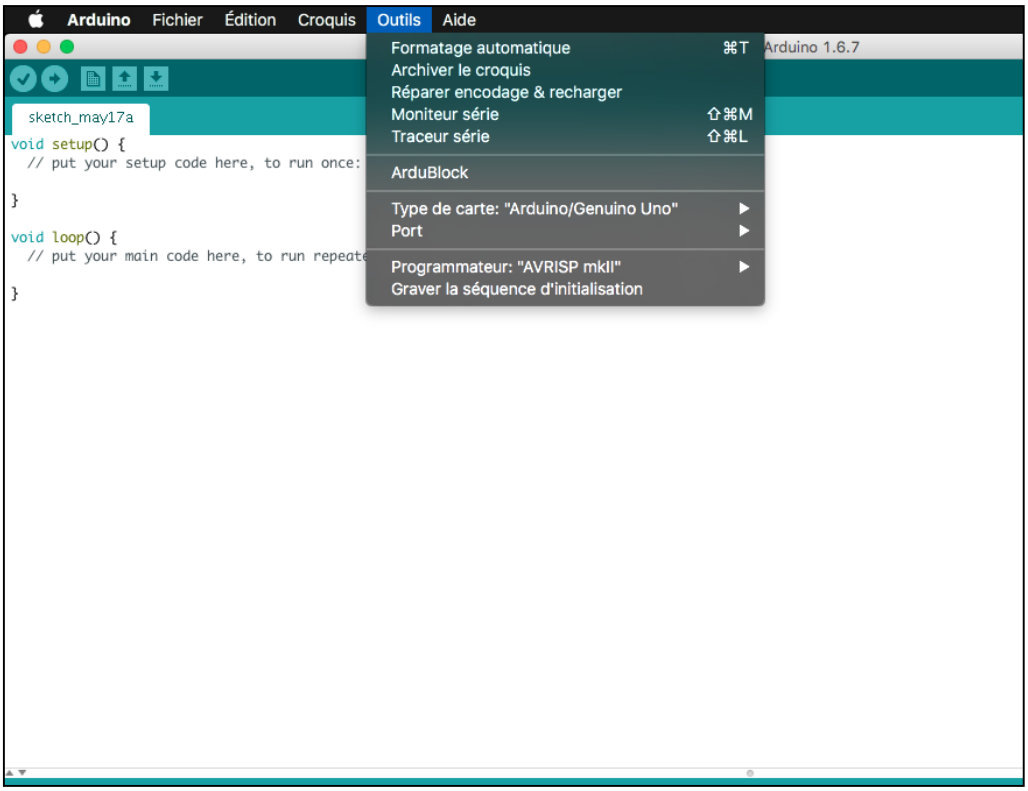


Figure 11 : Interface Arduino

3- Installation de Blockly@rduino

Blockly@rduino est disponible directement en ligne à l'adresse :
http://www.techmania.fr/BlocklyDuino/.

Afin de disposer des derniers blocs modifiés, il faudra vous rendre à l'adresse : https://github.com/technologiescollege/Blockly-at-rduino et télécharger tous les fichiers en cliquant sur « Download ZIP ».

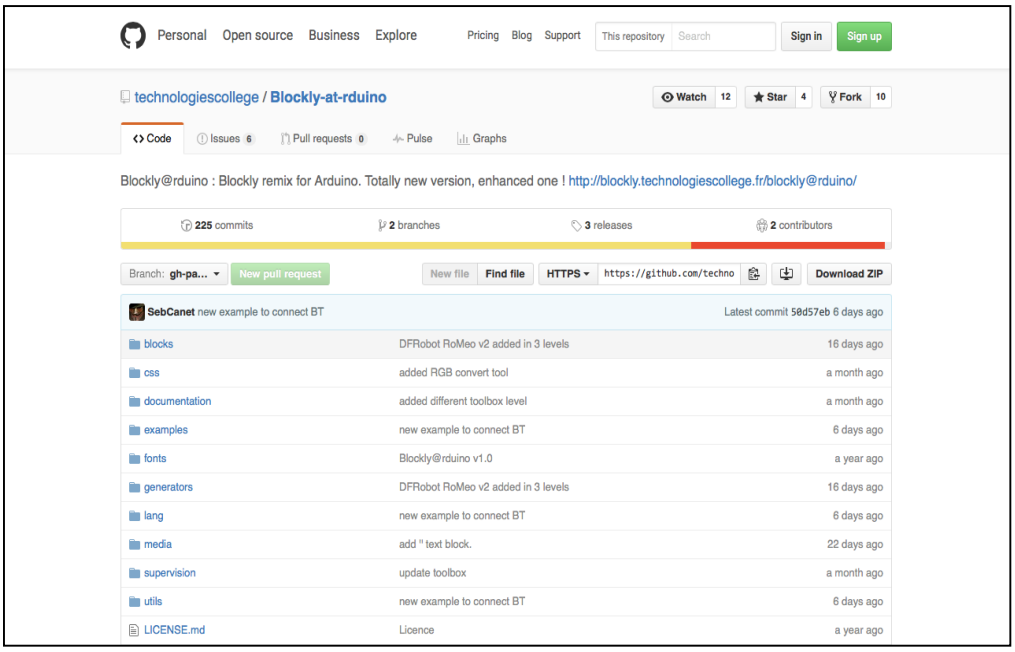


Figure 12 : Site internet pour le téléchargement des blocks mis en ligne

Une fois le fichier .zip téléchargé et décompressé, ouvrez le dossier ”Blockly-at-rduino-master” et double-cliquez sur le fichier **index.html**. L'interface Blocly@rduino va s'ouvrir. Vous pouvez modifier la langue en haut à droite et afficher tous les blocs disponibles en allant dans ”configurer les blocs” en haut à gauche et en sélectionnant toutes les catégories de blocs. Une fois cela fait vous devriez avoir une fenêtre similaire :

Figure 13 : Interface en ligne de Blockly@rduino

4- Utilisation d'Arduino

Vous avez fini l'installation du logiciel Arduino mais avant de réaliser vos programmes et les téléverser sur votre carte arduino :

- Branchez votre carte sur un port usb de votre ordinateur.
- Sélectionnez à partir du menu « Outils » votre type de carte.
- Sélectionnez à partir du menu « Outils » le port utilisé (le port usb où vous venez de brancher la carte électronique).

Vous pouvez à présent réaliser vos programmes sur l'IDE Arduino. Pour téléverser vos programmes sur la carte, cliquez sur la flèche en haut à gauche de l'IDE :

Figure 14 : Module de téléversement du logiciel Arduino

5- Installation des bibliothèques

Pour pouvoir utiliser certains composants sur l'environnement, il vous faudra installer une bibliothèque. Dans notre exemple du suivi de ligne, nous devons installer une bibliothèque pour les moteurs à courant continu.

Pour installer cette bibliothèque :

- Rendez-vous sur : https://www.dropbox.com/s/8wcnyxzt54namzh/Adafruit_Motorshield.zip?dl=0
- Téléchargez le fichier
- Déplacez le dossier "Adafruit_Motorshield" dans le dossier "libraries" dans l'emplacement du carnet de croquis (voir Tutoriel d'installation du plugin Ardublock). Si c'est la première bibliothèque que vous installez il vous faudra créer le dossier "libraries".
- Redémarrez le logiciel Arduino et vérifiez que vous avez bien le sous-menu :

Fichier->Exemples->Adafruit Motor Shield V2 Library->DCMotor Test

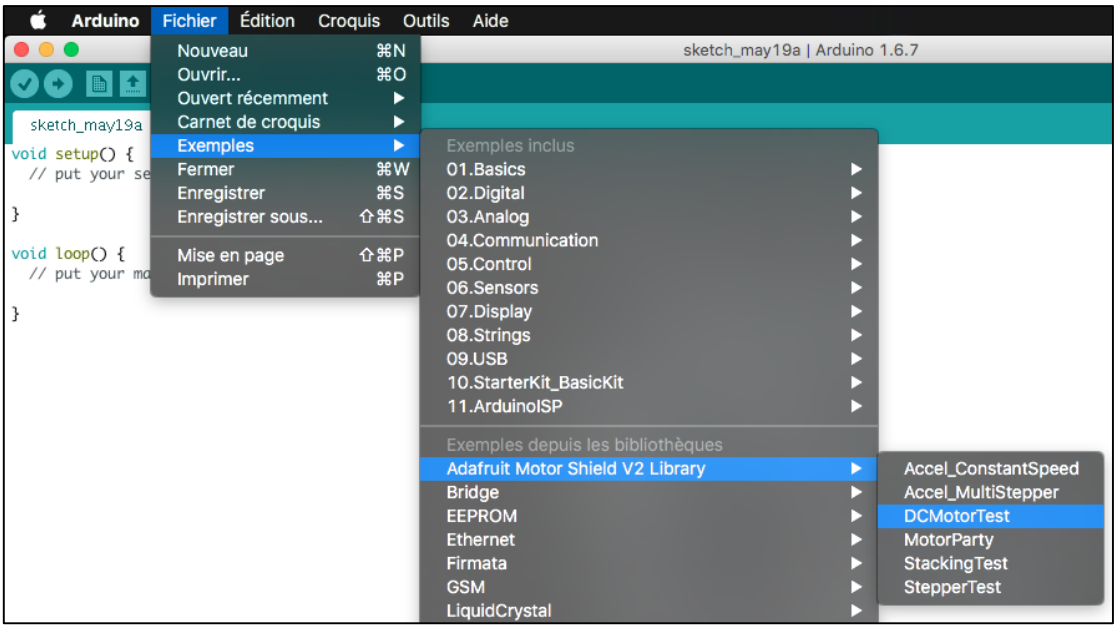


Figure 15 : Interface Arduino pour l'installation des bibliothèques

6- Utilisation d'Ardublock

Vous avez fini l'installation d'Ardublock et vous pouvez désormais réaliser vos programmes de manière graphique. Ouvrez Ardublock à partir de l'IDE Arduino et réalisez un premier programme en assemblant les différents blocs.

Avant de réaliser le téléversement, il ne faut pas oublier de choisir le type de carte et de sélectionner le bon port (voir Tutoriel d'utilisation d'Arduino). Pour téléverser vos programmes réalisés sur Ardublock, cliquez sur "Téléverser vers l'Arduino". En faisant cela, vos programmes graphiques seront traduits en langage Arduino sur l'IDE Arduino et téléversés vers votre carte.

7- Utilisation de Blockly@rduino

Vous avez fini l'installation de Blockly@rduino et pouvez désormais réaliser vos programmes de manière graphique. Lancez Blockly@rduino à partir du fichier index.html (voir Tutoriel d'installation de Blockly@rduino) et réalisez un premier programme en assemblant les différents blocs.

Pour téléverser vos programmes réalisés sur Blockly@rduino, cliquez sur "code Arduino" et copiez le programme sur l'IDE Arduino, puis téléversez le programme sur votre carte (voir Tutoriel d'installation d'Arduino). Or pour enregistrer au format .xml, il est impossible d'utiliser Safari, c'est pourquoi nous avons choisi utiliser Firefox.

b. Les composants

Afin de faciliter la mise en place des montages relatifs à notre robot, nous mettons à la disposition des futurs utilisateurs de ce tutoriel une liste des éléments utilisés ainsi que les sites où il est possible de les acheter (*voir VII. Annexes : Datasheet*) :

- **LED** : Leds 3mm et 10mm (http://snootlab.fr/10-leds)
- **Bouton poussoir** : Bouton poussoir FSM101 (http://fr.farnell.com/te-connectivity/fsm101/switch-spst-0-05a-24vdc-smd/dp/1813631)
- **Capteur infrarouge** : Capteur infrarouge TCRT5000 réfléchissant (https://www.amazon.fr/SODIAL-TCRT5000L-infrarouge-TCRT5000-reflechissant/dp/B00K67YKEO)
- **MCC** : Petit moteur SPK-10171 (http://snootlab.fr/sparkfun/352-petit-moteur-fr.html)
- **Shield** : Adafruit Motor Shield v2.3 (https://www.adafruit.com/product/1438)
- **Carte Arduino** : Seeeduino v4 (http://snootlab.fr/seeedstudio/909-seeeduino-v4-fr.html)

c. Les montages

- LED :

Afin d'ajouter des leds clignotantes à notre robot, il est possible de réaliser les branchements suivant à l'aide d'une carte Arduino ainsi qu'une résistance, dans le cas présent nous en avons utilisé une de 1k Ohms. On relie un port au choix (dans le montage ci-dessous le port 6), à la résistance puis on la fait rejoindre la led qui sera elle-même relié au port GND de la carte Arduino.

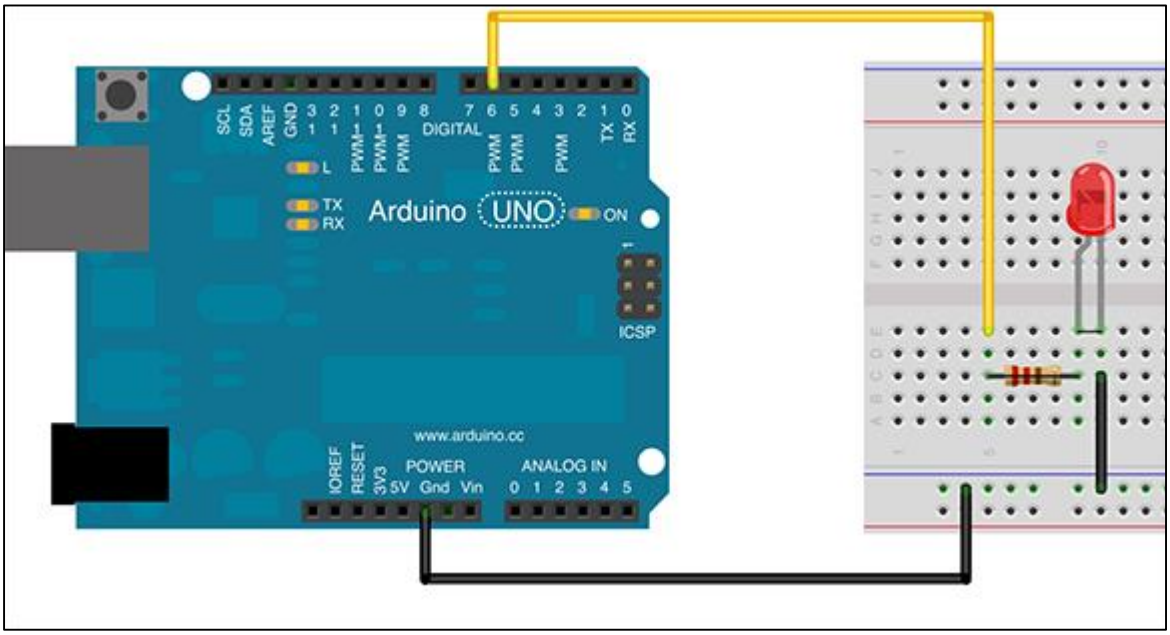


Figure 23 : Montage Arduino pour faire clignoter une led

Il suffit ensuite de coder le fonctionnement de la LED sur le logiciel Arduino dans le but de transférer ce code sur la carte Arduino. On utilise le pin 6 de notre carte pour y relier la LED, et ensuite on programme la carte pour qu'elle envoi l'information « mise en marche » à la LED, puis qu'elle attende une seconde, et que la carte envoi alors « mise hors service » à la LED.

```
void setup() {
  // initialize digital pin 6 as an output.
  pinMode(6, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(6, HIGH);   // turn the LED on (HIGH is the voltage
level)
  delay(1000);               // wait for a second
  digitalWrite(6, LOW);    // turn the LED off by making the voltage
LOW
  delay(1000);               // wait for a second
}
```

- Bouton poussoir :

On souhaite donner aux utilisateurs du robot la possibilité d'allumer celui-ci à l'aide d'un bouton poussoir. Pour cela, on montre dans l'exemple qui suit comment allumer et éteindre une LED avec un bouton, ce qui sera la même méthode utilisée pour l'allumage de notre robot. On a alors besoin d'une carte Arduino, d'une LED, d'une résistance (1k Ohms par exemple) et d'un bouton poussoir de type FSM101. Il ne reste plus qu'à câbler le montage suivant le modèle ci-dessous ; on branche directement la led sur la carte au niveau du port 13, ensuite on relie le port 2 à l'entrée du bouton poussoir. On alimente le bouton en le reliant au port 5V de la carte ainsi qu'au port GND de celle-ci tout en passant par la résistance.

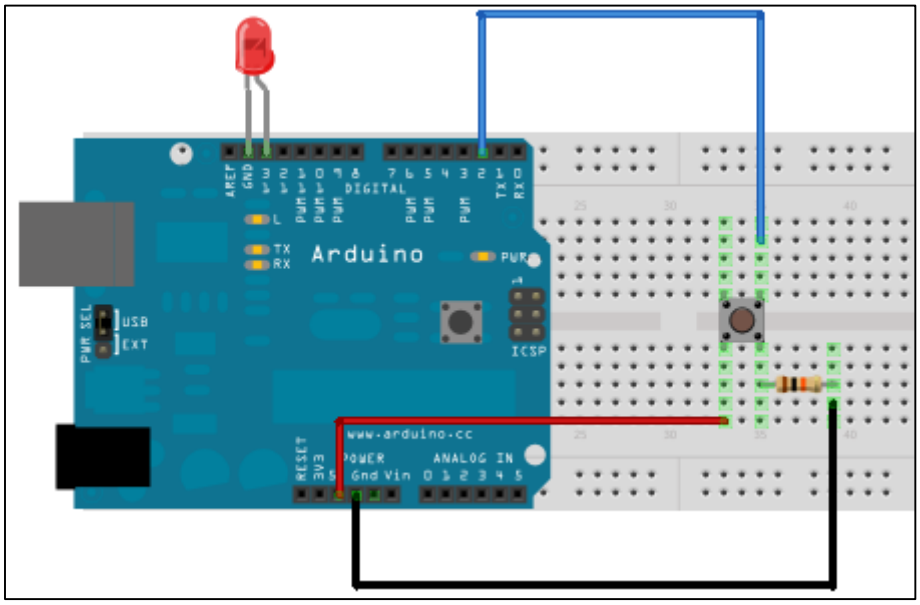


Figure 23 : Montage Arduino pour allumer et éteindre une led

Ensuite, il faut implanter le programme suivant sur la carte à l'aide du logiciel Arduino, ce qui permettra d'allumer la diode relié au port 13 lorsque le bouton relié au pin 2 de l'Arduino est pressé.

```
const int bouton = 2;
const int led = 13;
int etat;

// the setup function runs once when you press reset or power the board
void setup() {
      // initialize digital pin 1 as an output.
      pinMode(led, OUTPUT);
      pinMode(bouton, INPUT);
      etat = HIGH;
      digitalWrite(led, HIGH);
}
// the loop function runs over and over again forever
void loop() {
      etat = digitalRead(bouton); //Rappel : bouton = 2
      while(etat == HIGH)
      {
            etat = digitalRead(bouton);
      }
      digitalWrite(led, LOW);
      while(etat == LOW)
      {
            etat = digitalRead(bouton);
      }
      while(etat == HIGH)
      {
            etat = digitalRead(bouton);
      }
      digitalWrite(led, HIGH);
      while(etat == LOW)
      {
            etat = digitalRead(bouton);
      }
}
```

- Moteur à Courant Continu :

Notre robot faisant du suivi de ligne, il était nécessaire que celui-ci ai une capacité de déplacement. Nous avons donc mis en place un montage afin de relier notre moteur à courant continu au Shield ainsi qu'à la roue motrice du robot, nous assembleront ensuite le Shield et la carte Arduino pour assurer la communication. Nous avons suivi le montage suivant qui nécessite un petit moteur, des fils, un Shield et une source d'énergie. On branche le petit moteur au Shield sur le port M1 puis on ajout une source d'énergie (pile ou autre) en entrée Power. Il ne reste plus qu'à assembler le Shield et la carte Arduino pour que celle-ci lui envoi les ordres de fonctionnement.

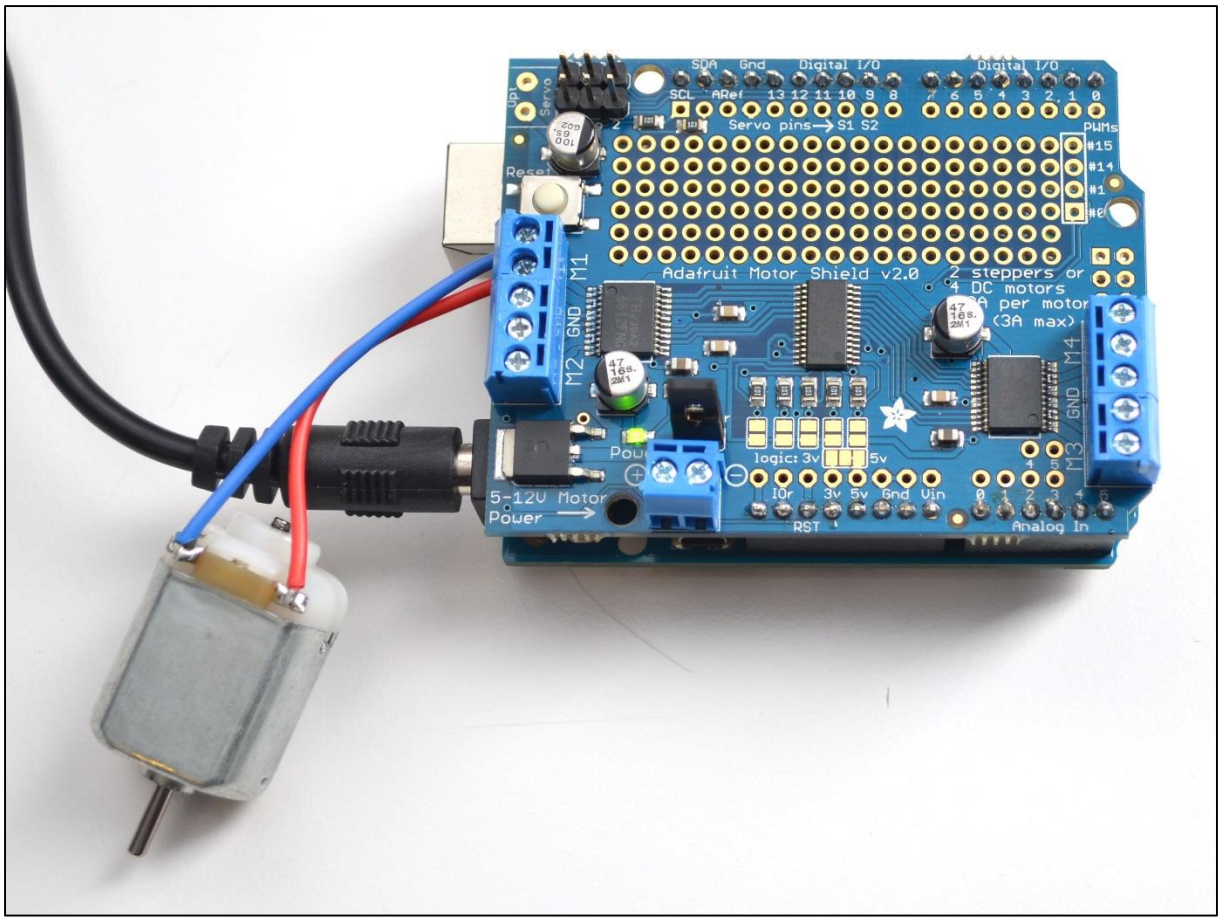


Figure 24 : Montage Arduino pour faire fonctionner un moteur à courant continu

Nous avons ensuite cherché plusieurs modèle de programmation afin d'utiliser au mieux les possibilités de la carte Shield. On règle donc la vitesse et le sens de rotation pour mettre en marche le moteur. Dans notre cas nous avons utilisé un Motor Shield v2 d'Adafruit et un petit moteur, trouvable chez Snootlab par exemple.

```
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Or, create it with a different I2C address (say for stacking)
// Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);

// Select which 'port' M1, M2, M3 or M4. In this case, M1
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
// You can also make another motor on port M2
//Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);

void setup() {
  Serial.begin(9600);           // set up Serial library at 9600 bps
  Serial.println("Adafruit Motorshield v2 - DC Motor test!");
```

```
  AFMS.begin();  // create with the default frequency 1.6KHz
  //AFMS.begin(1000);  // OR with a different frequency, say 1KHz

  // Set the speed to start, from 0 (off) to 255 (max speed)
  myMotor->setSpeed(150);
  myMotor->run(FORWARD);
  // turn on motor
  myMotor->run(RELEASE);
}

void loop() {
  uint8_t i;

  Serial.print("tick");

  myMotor->run(FORWARD);
  for (i=0; i<255; i++) {
    myMotor->setSpeed(i);
    delay(10);
  }
  for (i=255; i!=0; i--) {
    myMotor->setSpeed(i);
    delay(10);
  }

  Serial.print("tock");

  myMotor->run(BACKWARD);
  for (i=0; i<255; i++) {
    myMotor->setSpeed(i);
    delay(10);
  }
  for (i=255; i!=0; i--) {
    myMotor->setSpeed(i);
    delay(10);
  }

  Serial.print("tech");
  myMotor->run(RELEASE);
  delay(1000);
}
```

- Capteur infrarouge :

Dans le but de réaliser un suivi de ligne avec notre robot, on utilise un capteur infrarouge TCRT5000 que l'on pourra câbler avec des LED afin de pouvoir observer le bon fonctionnement de notre dispositif. Lorsque le blanc est détecté c'est une des deux leds qui s'allume, et quand c'est le noir qui est détecté, c'est la deuxième led s'allume. On peut ainsi suivre une ligne noire, et envoyer les informations aux moteurs pour qu'ils corrigent la direction de déplacement si la couleur noire n'est plus détectée. On commence par alimenter la plaquette en la reliant au port 5V de la carte Arduino, puis on rajoute deux résistances, dans le cas présent de 100 Ohms et 10k Ohms, afin de protéger le capteur infrarouge. On relie ensuite le capteur infrarouge à la pin A0 de la carte Arduino

pour que celui-ci lui envoi le fonctionnement que nous coderons ensuite. Enfin, il suffit de relier le capteur au port GND de la carte pour modéliser la masse du circuit.

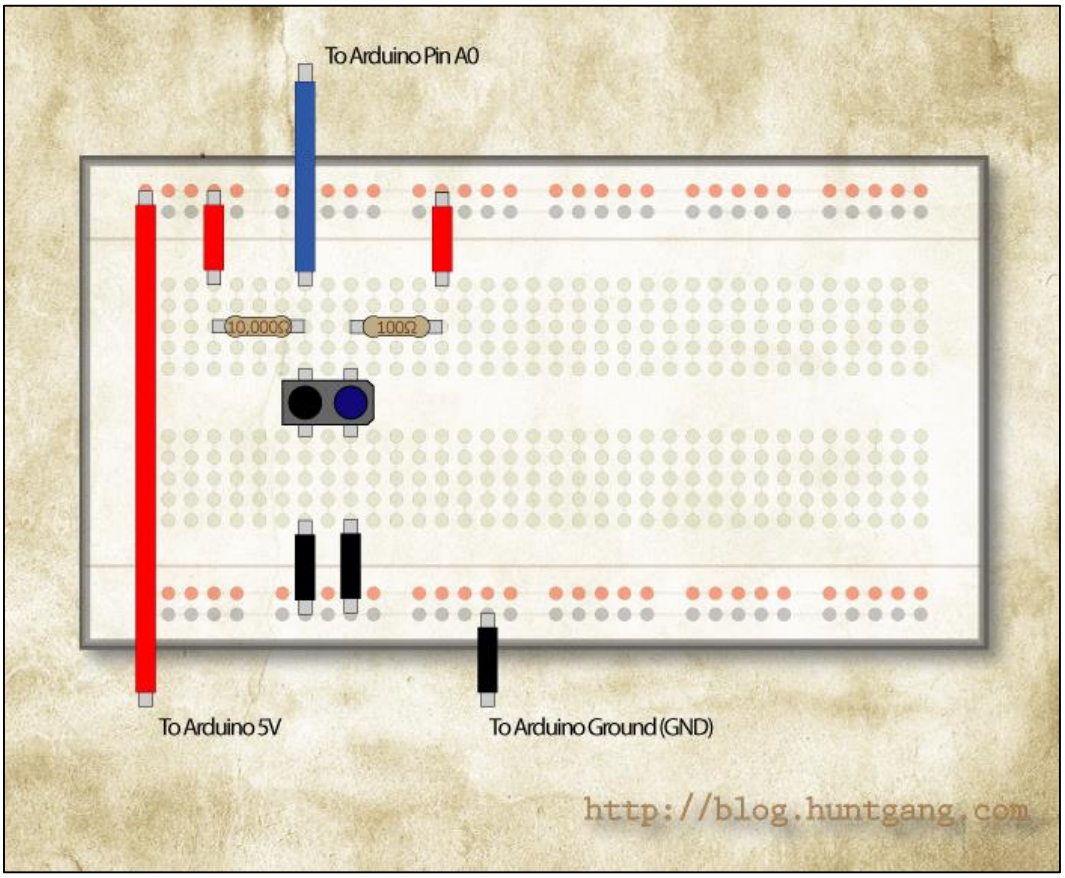


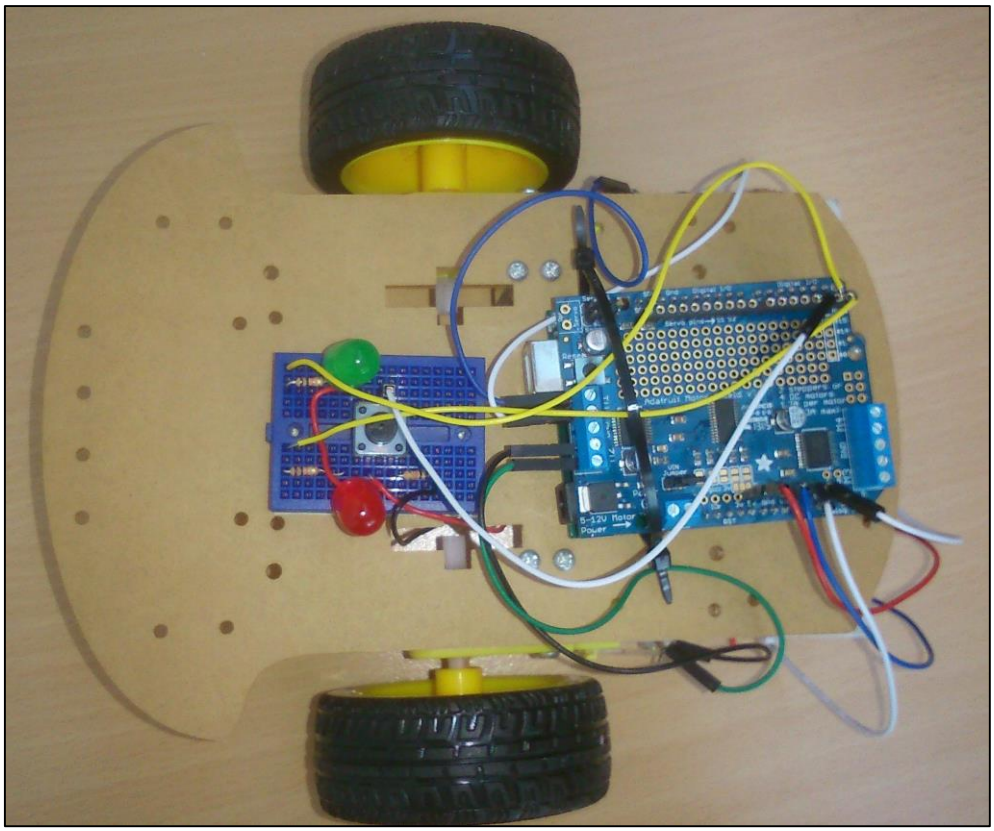
Figure 25 : Montage Arduino pour lire la valeur d'un capteur infrarouge TCRT 5000 analogique sur le moniteur série

Ensuite, il ne reste plus qu'à coder le fonctionnement du capteur sous Arduino et l'implémenter sur la carte. Pour cela, on utilise le code suivant :

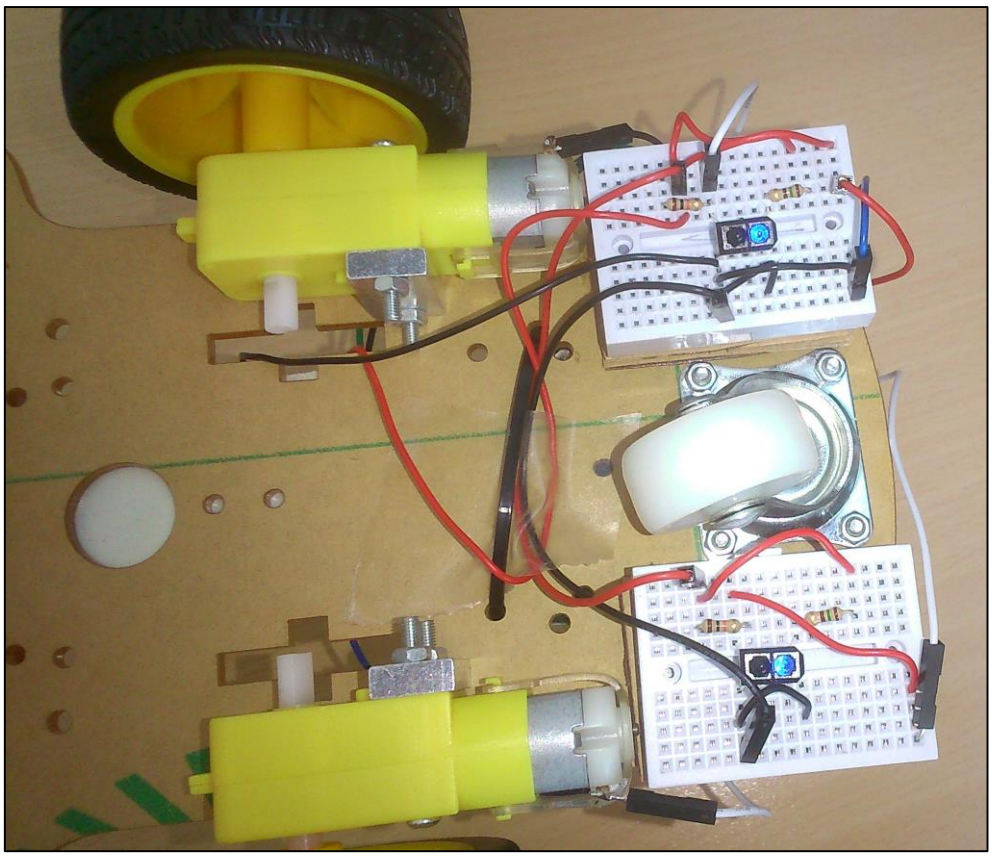
```
void setup() {
  pinMode(0, INPUT);
  Serial.begin(9600);
}
void loop() {
    Serial.println(analogRead(0));
    delay(1);
}
```

- **Montage final et complet du robot :**

- Vue de dessus :



- Vue de dessous :



d. Fonctionnement des logiciels

1- Ardublock

- Faire clignoter une led :

Fichier disponible sur : https://www.dropbox.com/s/66s2tvimfusqr77/led_ardublock.abp?dl=0

- Branchez votre led sur la sortie numérique 13 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme, située dans la catégorie ”Contrôle”.
- Ajoutez à l’intérieur de votre boucle un bloc pour fixer la sortie numérique, situé dans la catégorie ”Broches”. Fixez la sortie 13 au niveau haut (led éteinte).
- Ajoutez à l’intérieur de votre boucle une temporisation, située dans la catégorie ”Contrôle”. Choisissez un bloc delay en millisecondes et fixez la temporisation à 1000 (soit 1 seconde).
- Ajoutez à l’intérieur de votre boucle un nouveau bloc pour fixer la sortie numérique. Fixez la sortie 13 au niveau bas (led allumée).
- Ajoutez à l’intérieur de votre boucle une nouvelle temporisation. Choisissez un bloc delay en millisecondes et fixez la temporisation à 1000 (soit 1 seconde).
- Téléversez votre programme sur votre carte Arduino.

Algorithme :

```
Début
    Tant que vrai == vrai
        Eteindre la led ;
        Attendre 1 seconde ;
        Allumer la led ;
        Attendre 1 seconde ;
    Fin Tant que
Fin
```

Figure 16 : Bloc Ardublock pour faire clignoter une led

- Utiliser un bouton pour allumer et éteindre une led :

Fichier disponible sur : https://www.dropbox.com/s/pb43ipfl8yer1i6/led_bouton_ardublock.abp?dl=0

- Branchez votre led sur la sortie numérique 13 (*voir IV. d. Les montages*).
- Branchez votre bouton sur l'entrée numérique 2 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme, située dans la catégorie "Contrôle".
- Ajoutez un bloc Tant que de la catégorie "Contrôle". Dans le test, comparez la valeur de la broche Entrée numérique 2 (bouton) à l'état haut (bouton relâché). Le bloc "Valeur de la broche Entrée numérique" est situé dans la catégorie Broches.
- Après le bloc Tant que, fixez la sortie numérique 13 (led) au niveau bas (led allumée). Le bloc "Fixe la sortie numérique" est situé dans la catégorie Broches.
- Ajoutez un bloc Tant que. Dans le test, comparez la valeur de la broche Entrée numérique 2 à l'état bas (bouton appuyé).
- Après le bloc précédent, ajoutez un bloc Tant que. Dans le test, comparez la valeur de la broche Entrée numérique 2 à l'état haut (bouton relâché).
- Après le bloc Tant que, fixez la sortie numérique 13 au niveau haut (led éteinte).
- Ajoutez un bloc Tant que. Dans le test, comparez la valeur de la broche Entrée numérique 2 à l'état bas (bouton appuyé).
- Téléversez votre programme sur votre carte Arduino.

Algorithme :

```
Début
    Tant que vrai == vrai
        Tant que le bouton est relâché
            Ne rien faire ;
        Fin Tant que
        Allumer la led ;
        Tant que le bouton est appuyé
            Ne rien faire ;
        Fin Tant que
        Tant que le bouton est relâché
            Ne rien faire ;
        Fin Tant que
        Eteindre la led ;
        Tant que le bouton est appuyé
            Ne rien faire ;
        Fin Tant que
    Fin Tant que
Fin
```

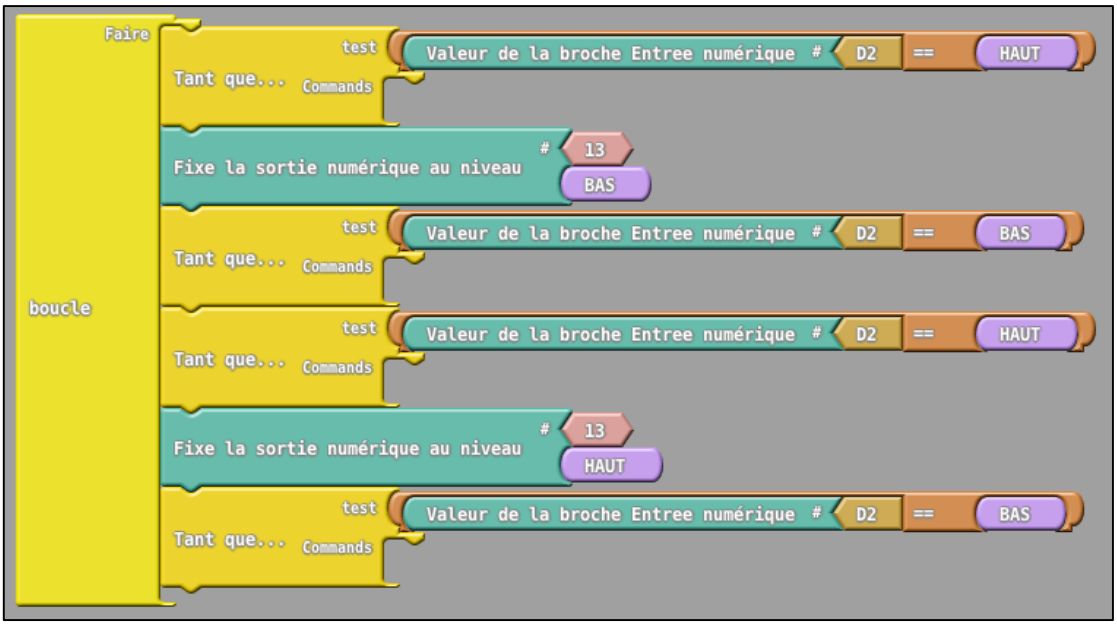


Figure 17 : Bloc Ardublock pour allumer et éteindre une led avec un bouton

- Faire fonctionner un moteur à courant continu :

Fichier disponible sur : https://www.dropbox.com/s/tx5glbjzcsi05ng/moteur_ardublock.abp?dl=0

- Branchez le moteur sur le shield en M1 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme, située dans la catégorie ”Contrôle”.
- Ajoutez à l’intérieur de votre boucle un bloc DC Motor **v2** Forward avec 1 pour le canal du moteur. Ce bloc est situé dans la catégorie ”Adafruit”.
- Ajoutez une temporisation. Choisissez un bloc delay en millisecondes, situé dans la catégorie ”Contrôle”, et fixez la temporisation à 5000 (soit 5 seconde).
- Ajoutez un bloc DC Motor **v2** Release avec 1 pour le canal du moteur. Ce bloc est situé dans la catégorie ”Adafruit”.
- Ajoutez une temporisation. Choisissez un bloc delay en millisecondes et fixez la temporisation à 5000 (soit 5 seconde).
- Ajoutez un bloc DC Motor **v2** Backward avec 1 pour le canal du moteur. Ce bloc est situé dans la catégorie ”Adafruit”.
- Ajoutez une temporisation. Choisissez un bloc delay en millisecondes et fixez la temporisation à 5000 (soit 5 seconde).
- Téléversez votre programme sur votre carte Arduino.

Algorithme :

```
Début
    Tant que vrai == vrai
        Faire tourner le moteur vers l’avant;
        Attendre 5 secondes ;
        Arrêter le moteur;
        Attendre 5 secondes;
```

Faire tourner le moteur vers l'arrière;
Attendre 5 secondes;
Fin Tant que
Fin

Figure 18 : Bloc Ardublock pour faire fonctionner un moteur à courant continu

- Lire la valeur d'un capteur infrarouge TCRT 5000 analogique sur le moniteur série :

Fichier disponible sur : https://www.dropbox.com/s/nhe8uj2davacamw/TCRT5000_ardublock.abp?dl=0

- Branchez le capteur sur l'entrée analogique 0 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme, située dans la catégorie "Contrôle".
- Ajoutez à l'intérieur de votre boucle un bloc pour écrire sur le port série, situé dans la catégorie "Communication".
- Pour ce bloc, dans la partie message remplacez le bloc présent par un bloc Coller finissant en triangle de la catégorie "Communication"
- A côté du bloc Coller, ajoutez un bloc Valeur de la broche Entrée Analogique, situé dans la catégorie "Broches", avec comme entrée 0.
- Téléversez votre programme sur votre carte Arduino.

Algorithme :

Début
Tant que vrai == vrai
Ecrire sur le port série (Valeur du capteur) ;
Fin Tant que

Fin

Figure 19 : Bloc Ardublock pour lire la valeur d'un capteur infrarouge TCRT 5000 analogique sur le moniteur série

2- Blockly@rduino

- Faire clignoter une led

Fichier disponible sur : https://www.dropbox.com/s/ib0sto1xl6zktu6/led_blockly.xml?dl=0

- Branchez votre led sur la sortie numérique 13 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme (bloc setup et loop), située dans la catégorie "arduino".
- Ajoutez à l'intérieur de votre boucle (partie loop) un bloc pour mettre la broche numérique à un niveau, situé dans la sous-catégorie "sortie" de la catégorie "arduino". Fixez la sortie 13 au niveau haut (led éteinte).
- Ajoutez à l'intérieur de votre boucle une temporisation, située dans la sous-catégorie "temps & durées" de la catégorie "arduino". Fixez la temporisation à 1000 (soit 1 seconde).
- Ajoutez à l'intérieur de votre boucle un nouveau bloc pour mettre la broche numérique à un niveau. Fixez la sortie 13 au niveau bas (led allumée).
- Ajoutez à l'intérieur de votre boucle une nouvelle temporisation. Fixez la temporisation à 1000 (soit 1 seconde).
- Copiez le code dans l'IDE Arduino et téléversez votre programme sur votre carte Arduino.

Algorithme :

```
Début
    Tant que vrai == vrai
        Eteindre la led ;
        Attendre 1 seconde ;
        Allumer la led ;
        Attendre 1 seconde ;
    Fin Tant que
Fin
```

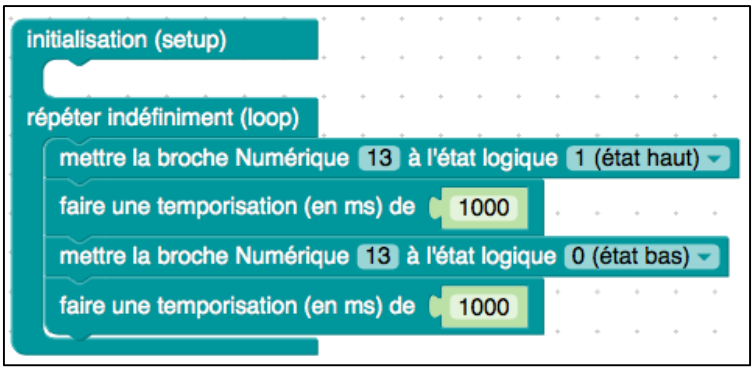


Figure 20 : Bloc Blockly@rduino pour faire clignoter une led

- Utiliser un bouton pour allumer et éteindre une led :

Fichier disponible sur : https://www.dropbox.com/s/iajsivtgza0fazv/led_bouton_blockly.xml?dl=0

- Branchez votre led sur la sortie numérique 13 (*voir IV. d. Les montages*).
- Branchez votre bouton sur l'entrée numérique 2 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme (bloc setup et loop), située dans la catégorie "arduino".
- Ajoutez, dans la partie loop, un bloc répéter tant que de la catégorie "boucles". Dans le test, comparez la valeur de la broche numérique 2 (bouton) à l'état haut (bouton relâché). Le bloc "l'état logique de la broche numérique" est situé dans la sous-catégorie "entrée" de la catégorie "arduino".
- Après le bloc répéter tant que, ajoutez un bloc pour mettre la broche numérique 13 (led) au niveau bas (led allumée). Le bloc est situé dans la sous-catégorie "sortie" de la catégorie "arduino".
- Ajoutez un bloc répéter tant que. Dans le test, comparez la valeur de la broche numérique 2 à l'état bas (bouton appuyé).
- Après le bloc précédent, ajoutez un bloc répéter tant que. Dans le test, comparez la valeur de la broche numérique 2 à l'état haut (bouton relâché).
- Après le bloc répéter tant que, ajoutez un bloc pour mettre la broche numérique 13 au niveau haut (led éteinte).
- Ajoutez un bloc répéter tant que. Dans le test, comparez la valeur de la broche numérique 2 à l'état bas (bouton appuyé).
- Copiez le code dans l'IDE Arduino et téléversez votre programme sur votre carte Arduino.

Algorithme :

```
Début
    Tant que vrai == vrai
        Tant que le bouton est relâché
            Ne rien faire ;
        Fin Tant que
        Allumer la led ;
        Tant que le bouton est appuyé
```

Ne rien faire ;
Fin Tant que
Tant que le bouton est relâché
Ne rien faire ;
Fin Tant que
Eteindre la led ;

Tant que le bouton est appuyé
Ne rien faire ;
Fin Tant que
Fin Tant que
Fin

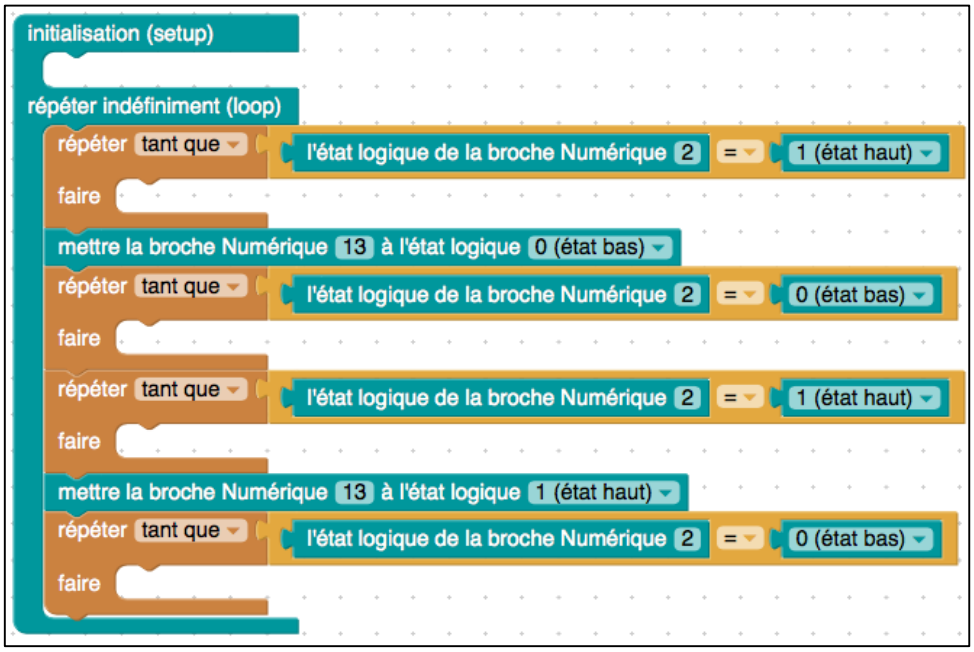


Figure 20 : Bloc Blockly@rduino pour allumer et éteindre une led avec un bouton

- Faire fonctionner un moteur à courant continu :

Fichier disponible sur : https://www.dropbox.com/s/wmzvdio3n040nxd/moteur_blockly.xml?dl=0

- Branchez le moteur sur le shield en M1 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme (bloc setup et loop), située dans la catégorie ”arduino”.
- Ajoutez, dans la partie loop, un bloc v2 Moteur CC situé dans la sous-catégorie ”Version 2” dans la catégorie ”cartes moteurs”. Choisissez la broche M1, direction Avant et vitesse 100.
- Ajoutez une temporisation, située dans la sous-catégorie “temps & durées“ de la catégorie ”arduino”. Fixez la temporisation à 5000 (soit 5 secondes).
- Ajoutez, dans la partie loop, un bloc v2 Moteur. Choisissez la broche M1, direction Stop et vitesse 0.

- Ajoutez une temporisation. Fixez la temporisation à 5000 (soit 5 secondes).
- Ajoutez, dans la partie loop, un bloc v2 Moteur CC. Choisissez la broche M1, direction Arriere et vitesse 100.
- Ajoutez une temporisation. Fixez la temporisation à 5000 (soit 5 secondes).
- Copiez le code dans l'IDE Arduino et téléversez votre programme sur votre carte Arduino.

Algorithme :

```
Début
    Tant que vrai == vrai
        Faire tourner le moteur vers l'avant;
        Attendre 5 secondes;
        Arrêter le moteur;
        Attendre 5 secondes;
        Faire tourner le moteur vers l'arrière;
        Attendre 5 secondes;
    Fin Tant que
Fin
```

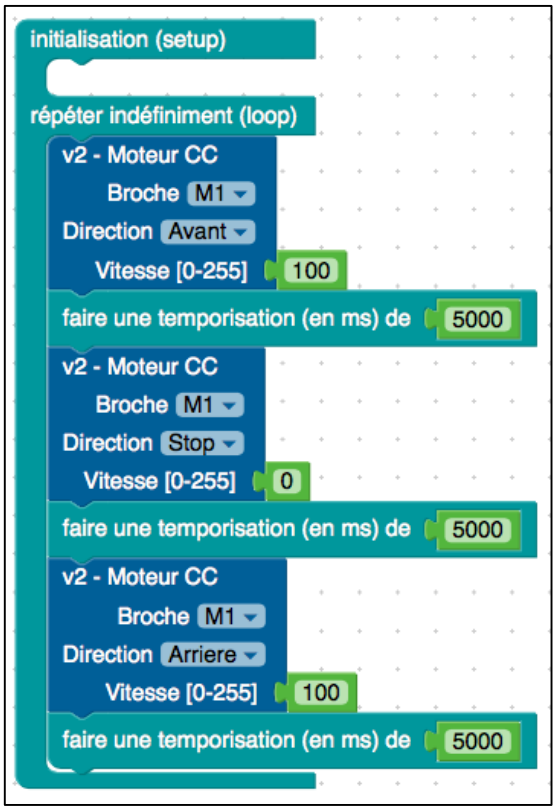


Figure 21 : Bloc Blockly@rduino pour faire fonctionner un moteur à courant continu

- Lire la valeur d'un capteur infrarouge TCRT 5000 analogique sur le moniteur série :

Fichier disponible sur : https://www.dropbox.com/s/89uvgwuqekah22h/TCRT5000_blockly.xml?dl=0

- Branchez le capteur sur l'entrée analogique 0 (*voir IV. d. Les montages*).
- Commencez par ajouter une boucle à votre programme (bloc setup et loop), située dans la catégorie "arduino".
- Ajoutez, dans la partie loop, un bloc affichez, puis saut, sur le port série, situé dans la sous-catégorie "communication série" dans la catégorie "arduino".
- Copiez le code dans l'IDE Arduino et téléversez votre programme sur votre carte Arduino.

Algorithme :

Début
    Tant que vrai == vrai
        Ecrire sur le port série (Valeur du capteur) ;
    Fin Tant que
Fin

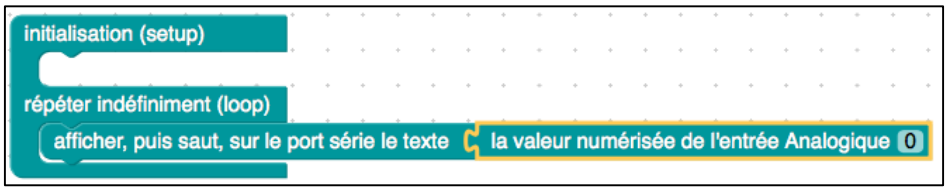


Figure 22 : Bloc Blockly@rduino pour lire la valeur d'un capteur infrarouge TCRT 5000 analogique sur le moniteur série

e. Algorithme de suivi de ligne

1- Ardublock

Nous mettons à votre disposition les algorithmes du suivi de ligne seul et ainsi que du suivi de ligne avec led et bouton, grâce au logiciel Ardublock.

- Algorithme de suivi de ligne :

Le fichier. abp est disponible sur :
https://www.dropbox.com/s/nx8263oc1yt1u95/suivi_ardublock.abp?dl=0

Début
    Tant que vrai == vrai
        Si capteur IR droit < val && capteur IR gauche < val
            Avancer moteur droit;
            Avancer moteur gauche;
            Sinon Si capteur IR droit > val && capteur IR gauche < val

Avancer moteur gauche;
Arrêter moteur droit;
Sinon Si capteur IR droit < val && capteur IR gauche > val
Arrêter moteur gauche;
Avancer moteur droit;
Fin Sinon
Fin Sinon
Fin Si
Fin Tant que
Fin

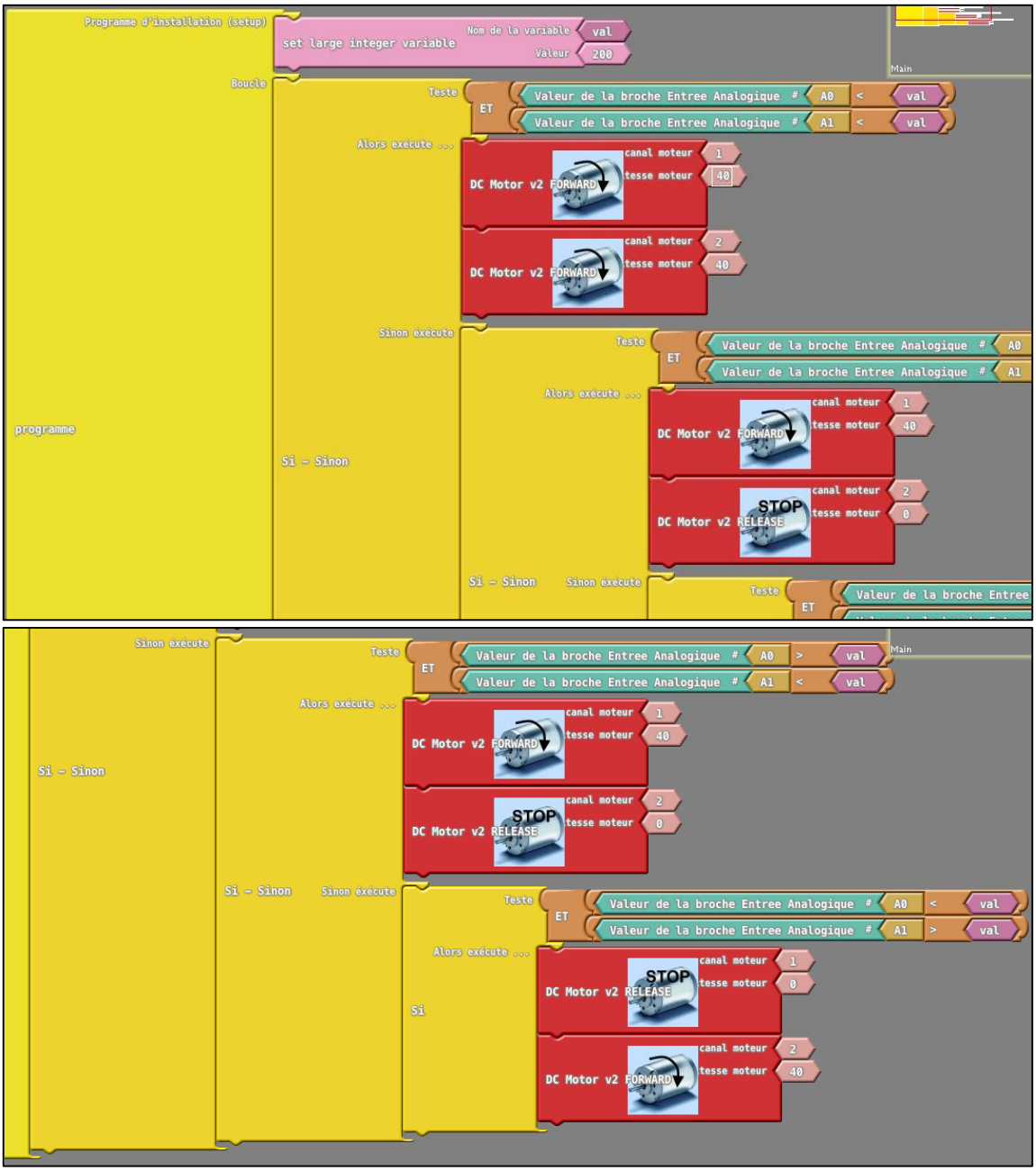
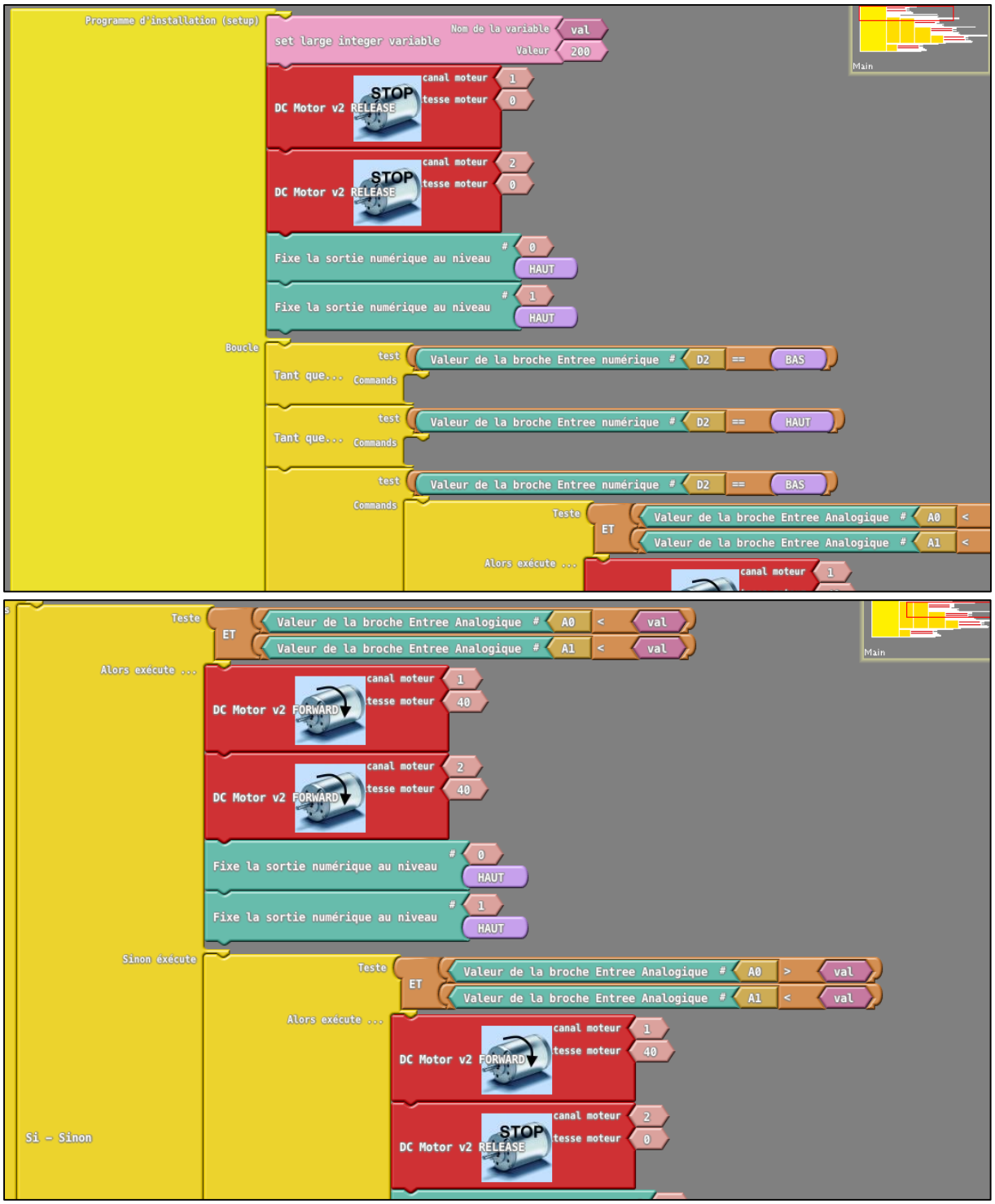


Figure 23 : Bloc Ardublock pour réaliser le suivi de ligne de notre robot

- Algorithme de suivi de ligne avec bouton et led :

Le fichier. abp est disponible sur :
https://www.dropbox.com/s/zgxr4zcf9buzsjk/suivi_ardublock2.abp?dl=0

```
Début
    Tant que vrai == vrai
        Tant que bouton relâché
            Ne rien faire;
        Fin Tant que
        Tant que bouton appuyé
            Ne rien faire;
        Fin Tant que
        Tant que bouton relâché
            Si capteur IR droit < val &&  capteur IR gauche < val
                Avancer moteur droit;
                Avancer moteur gauche;
                Eteindre led droite;
                Eteindre led gauche;
               Sinon Si capteur IR droit > val &&  capteur IR gauche <
        val
                    Avancer moteur gauche;
                    Arrêter moteur droit;
                    Allumer led droite;
                    Eteindre led gauche;
                    Sinon Si capteur IR droit < val &&  capteur IR
                gauche > val
                        Arrêter moteur gauche;
                        Avancer moteur droit;
                        Eteindre led droite;
                        Allumer led gauche;
                    Fin Sinon
                Fin Sinon
            Fin Si
        Fin Tan que
        Tant que bouton appuyé
            Arrêter moteur droit;
            Arrêter moteur gauche;
            Eteindre led droite;
            Eteindre led gauche;
        Fin Tant que
    Fin Tant que
Fin
```


Programme d'installation (setup)
set large integer variable
Nom de la variable
val
Valeur
200
DC Motor v2 RELEASE
STOP
canal moteur
tesse moteur
1
0
2
0
Fixe la sortie numérique au niveau
#
0
1
HAUT
Boucle
Tant que...
test
Commands
Valeur de la broche Entree numérique # D2 == BAS
Valeur de la broche Entree numérique # D2 == HAUT
Valeur de la broche Entree numérique # D2 == BAS
Teste
ET
Valeur de la broche Entree Analogique # A0 <
Valeur de la broche Entree Analogique # A1 <
Alors exécute ...
Main
DC Motor v2 FORWARD
40
Sinon éxécute
Valeur de la broche Entree Analogique # A0 > val
Valeur de la broche Entree Analogique # A1 < val
Si – Sinon

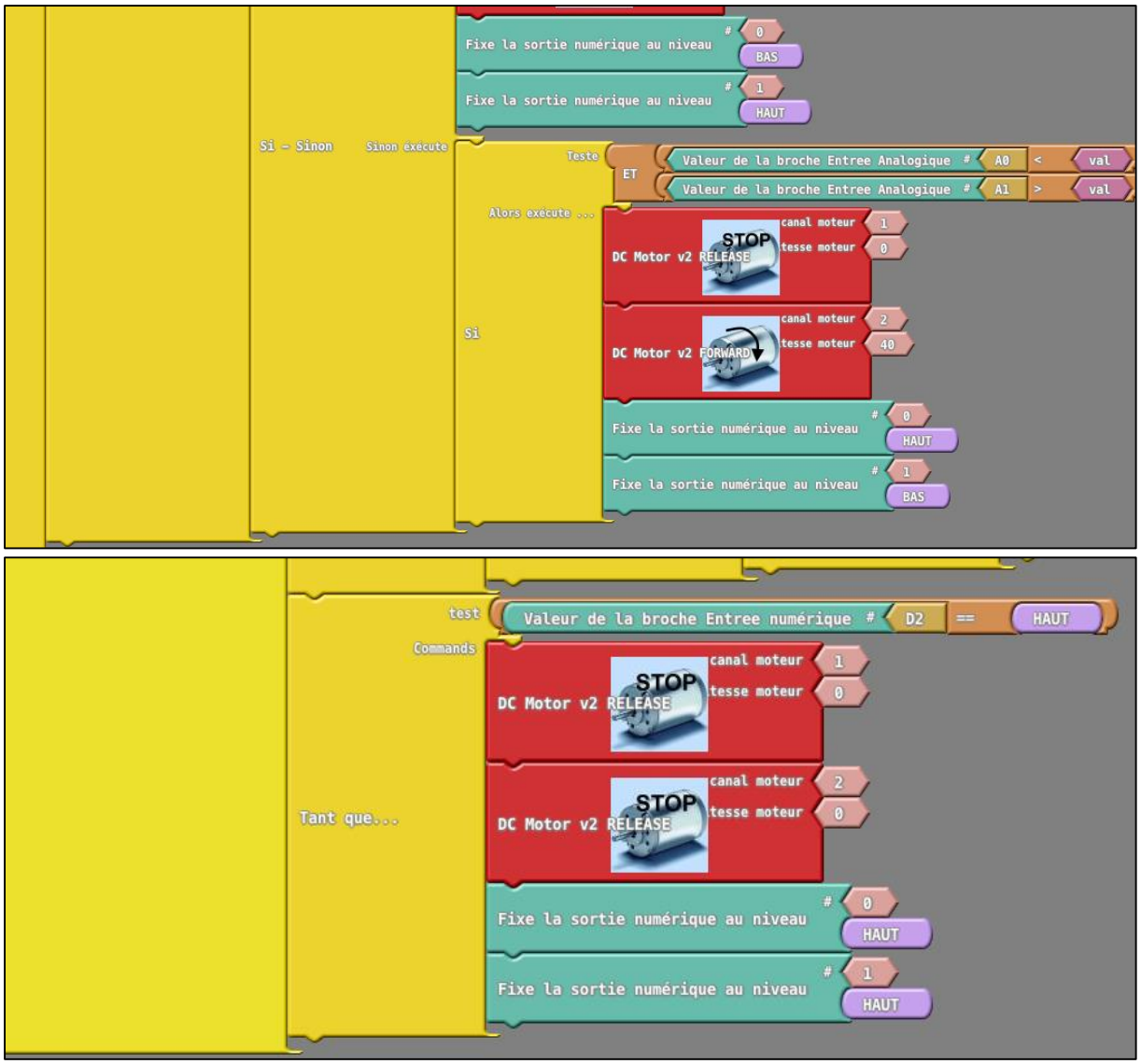


Figure 24 : Bloc Ardublock pour réaliser le suivi de ligne de notre robot avec allumage par bouton poussoir et leds

2- Blockly@rduino

Nous mettons à votre disposition les algorithmes du suivi de ligne seul et ainsi que du suivi de ligne avec led et bouton, grâce au logiciel Blockly@rduino.

- Algorithme de suivi de ligne :

Le fichier. xml est disponible sur :

https://www.dropbox.com/s/uttuvky27kcwmhb/suivi_blockly.xml?dl=0

Début

Tant que vrai == vrai

Si capteur IR droit < val && capteur IR gauche < val
Avancer moteur droit;
Avancer moteur gauche;
Fin Si
Si capteur IR droit > val && capteur IR gauche < val
Avancer moteur gauche;
Arrêter moteur droit;
Fin Si
Si capteur IR droit < val && capteur IR gauche > val
Arrêter moteur gauche;
Avancer moteur droit;
Fin Si
Fin Tant que
Fin

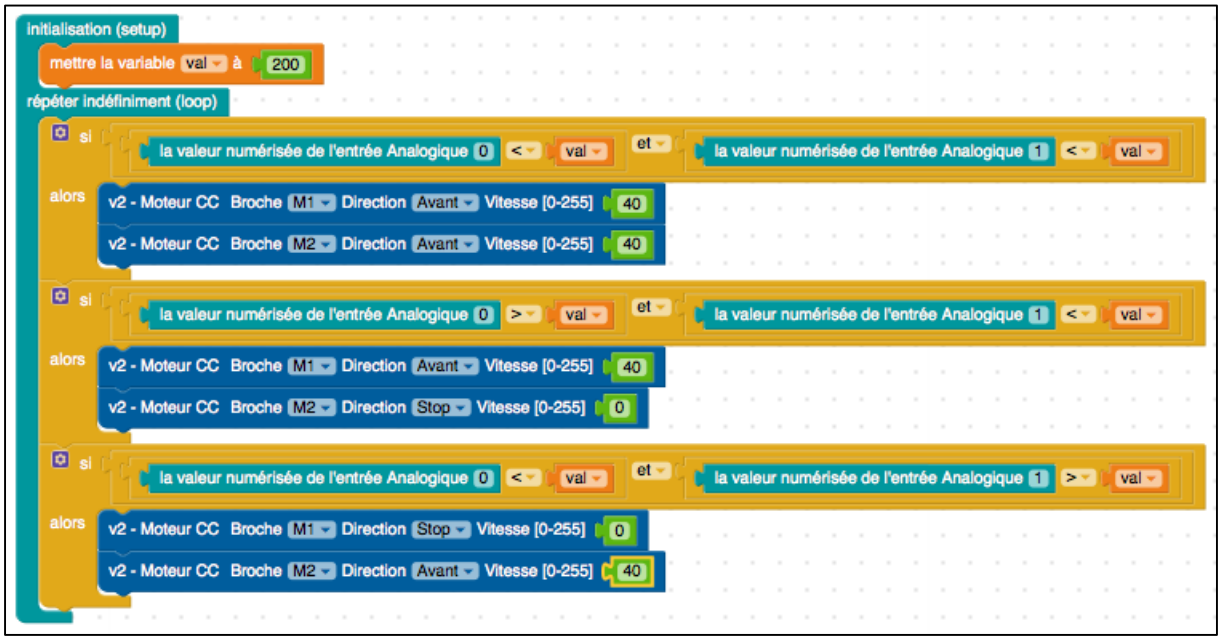


Figure 25: Bloc Blockly@rduino pour réaliser le suivi de ligne de notre robot

- Algorithme de suivi de ligne avec bouton et led :

Le fichier. xml est disponible sur :
https://www.dropbox.com/s/xvkciukxf56dits/suivi_blockly2.xml?dl=0

Début
Tant que vrai == vrai
Tant que bouton relâché
Ne rien faire;
Fin Tant que

Tant que bouton appuyé
Ne rien faire;

```
            Fin Tant que

            Tant que bouton relâché
                Si capteur IR droit < val &&  capteur IR gauche < val
                    Avancer moteur droit;
                    Avancer moteur gauche;
                    Eteindre led droite;
                    Eteindre led gauche;
                Fin Si

                Si capteur IR droit > val &&  capteur IR gauche < val
                    Avancer moteur gauche;
                    Arrêter moteur droit;
                    Allumer led droite;
                    Eteindre led gauche;
                Fin Si

                Si capteur IR droit < val &&  capteur IR gauche > val
                    Arrêter moteur gauche;
                    Avancer moteur droit;
                    Eteindre led droite;
                    Allumer led gauche;
                Fin Si
            Fin Tan que

            Tant que bouton appuyé
                Arrêter moteur droit;
                Arrêter moteur gauche;
                Eteindre led droite;
                Eteindre led gauche;
            Fin Tant que

        Fin Tant que
Fin
```

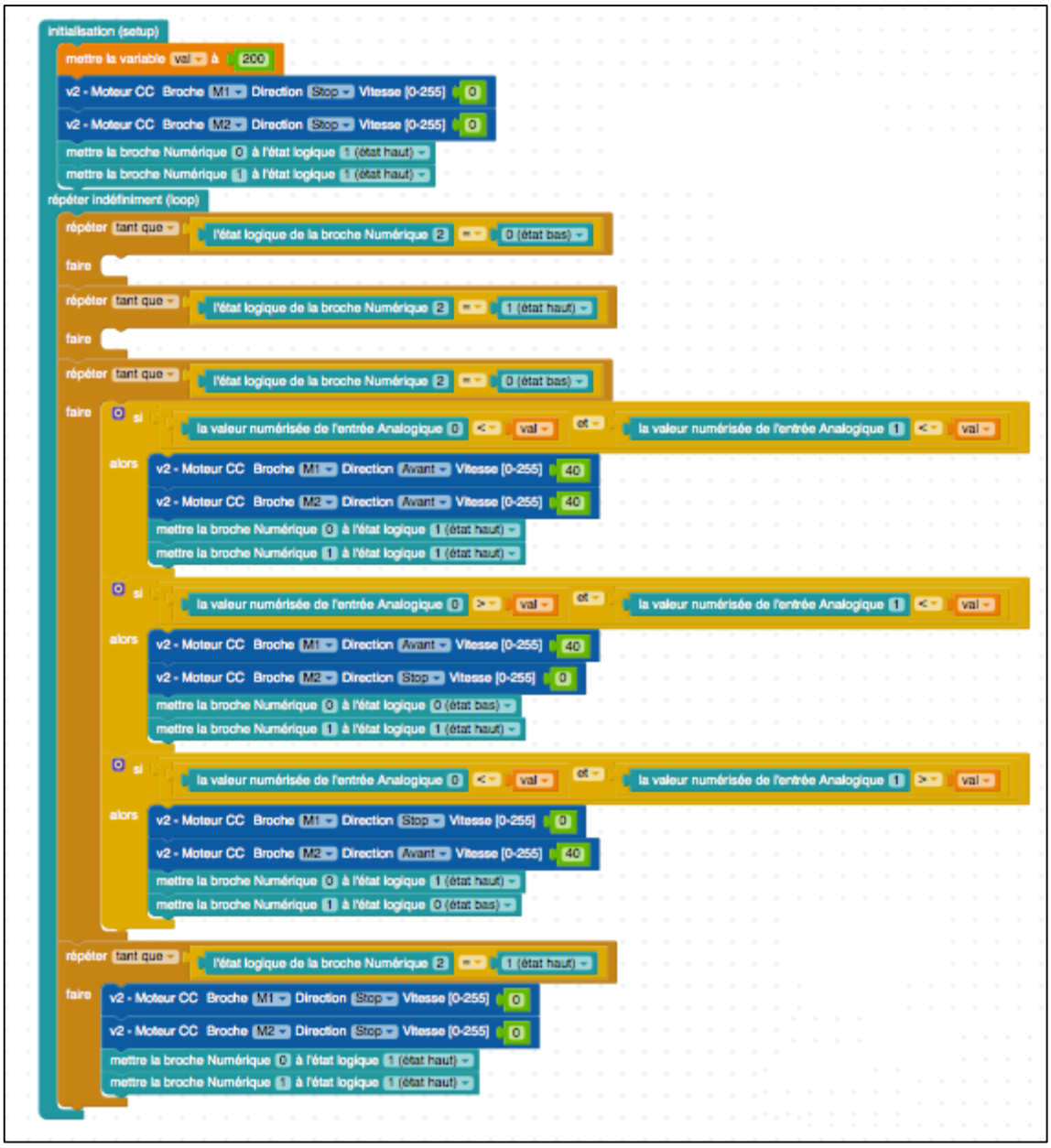


Figure 26 : Bloc Blockly@rduino pour réaliser le suivi de ligne de notre robot avec allumage par bouton poussoir et leds

V. Bilan du projet

Dans le cadre de notre projet nous avons dû réaliser la commande d'un robot destiné aux enfants de primaire. Cette commande devait être facile d'accès pour les enfants, dans le but de leur faire découvrir l'informatique d'une manière amusante. Nous espérons que notre application pourra ensuite profiter à tous du fait de l'utilisation de logiciels soumis aux critères de l'Open Source.

Ce projet s'est dans l'ensemble très bien déroulé, malgré quelques problèmes liés à un court-circuit. Cette expérience fût riche en enseignement dans la mesure où nous avons dû définir notre

travail en quasi-autonomie et qu'il nous a fallu mettre en place une bonne gestion de groupe. En effet, nous avons dû prendre des initiatives et coordonner notre travail d'équipe pour mener à bien le projet. De plus, il nous a permis d'appliquer et d'élargir nos connaissances en électronique et en programmation, ce qui sera très profitable dans notre parcours.

Certaines de nos compétences telles que la compréhension de schéma électrique ainsi que la mise en place des composants électroniques afin de réaliser des montages ont été indispensables quant au déroulement du projet. Cela nous a permis de nous remémorer des bases de Licence 3, qui nous seront toujours utiles dans la vie active.

La première partie du projet consistait en la prise en main de l'Arduino et des composants mis à notre disposition. Nous avons ainsi commencé par faire fonctionner les différents composants directement avec le langage C et les fonctions Arduino. Cette partie du projet ne nous a pas posé de problèmes majeurs. Les compétences nécessaires tel que la programmation en langage C étant déjà acquises.

Par la suite nous avons réfléchi à la mise en œuvre sous forme de blocs que les enfants pourraient programmer eux-mêmes. Toutefois ce ne fût pas aisé car nous avons rencontré des problèmes au niveau de la programmation. Certains langages nécessaires au développement du projet nous étaient alors inconnus. Nous avons dû effectuer de nombreuses recherches afin de contourner ces lacunes. Grâce à nos recherches respectives, nous avons pu réaliser l'implémentation requise à l'aide de quelques bases dans les langages jusqu'alors inconnus.

Lors de la dernière étape du projet, il nous a fallu développer une interface physique de notre robot, afin qu'il soit amusant pour les enfants qui étaient la cible visé. C'est pourquoi nous avons commencé par mettre en place un scénario ludo-éducatif portant sur un suivi de ligne où le robot se repèrerait au moyen de capteurs afin de suivre un chemin prédéfinit. Le principe étant d'analyser la couleur du sol et de changer de trajectoire en fonction de la couleur de la ligne pour toujours rester sur celle-ci.

Au final, le suivi de ligne complet, c'est-à-dire avec allumage par un bouton poussoir et leds indiquant si un roue est oui ou non bloqué, a bien fonctionné que ce soit avec les blocs Ardublock ou bien ceux réalisés sur Blockly@rduino. Il vous est possible d'observer le fonctionnement de notre robot sur :

https://www.dropbox.com/s/464mmdidmkqbq40/Suivi%20de%20ligne%20Ardublock%20avec%20pile%209V.mp4?dl=0

Nous devons cependant spécifier que lors de nos essais, plusieurs échecs ont pu être constaté car nous avons remarqué que parfois la carte fonctionnait et émettait un « bip » sans pour autant que le robot ne se mette en mouvement. Nous avons résolu ce problème car il s'apparentait à une vitesse trop lente des moteurs d'où l'absence de mouvement de la part du robot, il faut donc mettre une vitesse minimale de 70 et non de 40 comme il est indiqué dans certains blocs de notre tutoriel.

Le déroulement de ce projet s'est donc effectué sans gênes majeures, à l'exception de l'interface de programmation pour les enfants qui n'est peut-être pas assez abordable. Il nous est possible de réaliser la commande du robot sur différents logiciels, mais certain d'eux ne sont pas

destinés spécifiquement aux enfants. Cependant, il existe un robot mBot préalablement monté et câblé, que l'on peut contrôler avec mBlock et qui est facilement utilisable par des enfants. Nous avons essayé de développer notre commande sur un logiciel plus adapté aux enfants mais cela n'a pas était un succès, ces logiciels ne pouvant supporter qu'un code très simple. Nous avons donc choisi de privilégier l'avancement de la commande du robot pour obtenir la commande souhaitée s'accordant à au moins un logiciel en notre possession.

VI. Bibliographie

- **Arduino :**

Landrault, Simon ; Weisslinger Hippolyte. Arduino : Premiers pas en informatique embarquée. Le blog d'Eskimon, [19/06/2014]. 454p.

Landrault, Simon ; Weisslinger Hippolyte. Présentation d'Arduino [en ligne]. Zeste de savoir, 19/02/2016 [20/05/2016]. Disponible sur https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/742_decouverte-de-larduino/3414_presentation-darduino/

Reynier, Louis. C'est quoi Arduino ? [en ligne]. Prototypage électronique [20/05/2016]. Disponible sur http://www.louisreynier.com/fichiers/KesacoArduino.pdf

Arduino [en ligne]. 2016 [consulté le 20/05/2016]. Disponible sur https://www.arduino.cc

Jean-Noël Rousseau, Programmez vos premiers montages avec Arduino [en ligne]. Open Classrooms, 03/05/2016 [consulté le 20/05/2016]. Disponible sur https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino

- **Blockly@rduino :**

Belfiore, Guillaume. Blockly : Google simplifie la programmation web [en ligne]. Clubic, 12/05/2016 [20/05/2016]. Disponibilite sur http://www.clubic.com/pro/creation-de-site-web/langage-programmation/actualite-496422-blockly-langage-programmation-simplifiee-google.html

Canet, Sébastien ; Rat, Julien ; Olivier Métayer. Présentation et Initiation au concept Blockly@rduino [en ligne]. Technozone51 Documentation Center, 16/09/2015 [20/05/2016]. Disponible sur http://www.technozone51.fr/dokuwiki2/doku.php?id=documentation:initiation_et_presentation_au_concept_blocklyduino

Métayer, Olivier ; Picard, Frederic ; Pers, Daniel. Github [en ligne]. Technologiescollege/Blockly-at-rduino, 2016 [20/05/2016]. Multilingual version of Blockly@rduino. Disponible sur https://github.com/technologiescollege/Blockly-at-rduino

- **Ardublock :**

ressources en technologie college [en ligne]. scoop.it, [20/05/2016]. Disponible sur http://www.scoop.it/t/ressources-en-technologie-college/p/4029818947/2014/10/14/sti2d-apprenez-a-creer-vos-propres-blocs-pour-ardublock

Getting Started with Ardublock [en ligne]. Blog Ardublock [consulté le 20/05/2016]. Disponible sur http://blog.ardublock.com/engetting-started-ardublockzhardublock/

Julien Rat, Arduino : Présentation et traduction en Français de Ardublock [en ligne]. Semageek, 17/12/2011 [consulté le 20/05/2016]. Disponible sur http://www.semageek.com/arduino-presentation-et-traduction-en-francais-de-ardublock/

Pascal Pujades, Utilisation d'Ardublock : Programmation Arduino[en ligne]. Académie de Toulouse, octobre 2015 [consulté le 20/05/2016]. Disponible sur http://pedagogie.ac-toulouse.fr/technologie/doc-tutorial/didacticiel-95.pdf

Julien Rat, Ardublock créer des block [en ligne]. TIC Débrouillonet, 06/03/2014 [consulté le 20/05/2016]. Disponible sur http://blog.debrouillonet.org/TIC/index.php/post/2014/03/06/Ardublock-créer-des-block

Didier Carne, How to create a new block[en ligne]. Hack-E-Bot, 07/05/2014 [consulté le 20/05/2016]. Disponible sur http://www.hack-e-bot.com/how-to-create-a-new-ardublock/

Ardublock Adafruit MotorShield DC Motor [en ligne]. Google Groupes, 16/01/2016[consulté le 20/05/2016]. Disponible sur https://groups.google.com/forum/#!topic/ardublock/p8iY5vn5y4Y

- **Mblock :**

Mblock [en ligne]. 20/10/2015 [20/05/2016]. Disponible sur http://www.mblock.cc/

technologies et sciences des ingenieurs [en ligne]. Académie de Nante, [20/05/2016]. Disponible sur http://www.pedagogie.ac-nantes.fr/technologies-et-sciences-des-ingenieurs/documentation/didacticiels-tutoriels/mblock-videos-d-initiation-919548.kjsp?RH=1333492036996

- **Capteurs :**

Utilisation du module Ultrason HC-SR04 avec l'Arduino [en ligne]. iTechnoFrance, [20/05/2016]. Disponible sur https://itechnofrance.wordpress.com/2013/03/12/utilisation-du-module-ultrason-hc-sr04-avec-larduino/#more-429

Striebig, Benjamin. Capteur HC-SR04 [en ligne]. Tuto Arduino, 28/09/2013 [20/05/2016]. Disponible sur https://tutoarduino.com/capteur-hc-sr04/

Laetitia, [Tutoriel] Module à ultrasons HC-SR04 [en ligne].Snootlab, 11/02/2016 [consulté le 20/05/2016]. Disponible sur http://forum.snootlab.com/viewtopic.php?f=38&t=649

Dave, Introduction to Arduino TCRT5000 (IR Sensor) [en ligne]. Huntgang, 17/06/2014 [consulté le 20/05/2016]. Disponible sur http://blog.huntgang.com/2014/06/17/arduino-tcrt5000-build-ir-sensor/

- **Montages :**

Bertrand, Guillaume. Ex2-Connecter une LED [en ligne]. Processing & Arduino Workshops, [20/05/2016]. Disponible sur http://workshop.gui-aum.com/introduction-arduino/premier-sketch/ex-2-connecter-une-led/

Gaultier, Baptiste ; Hinault, Xavier. Atelier de découvert Arduino [en ligne]. Arduino Projects, 15/12/2011 [20 :05/2016]. Disponible sur http://redmine.labo4g.enstb.fr/projects/arduino-projects/wiki/Atelier_Arduino

Arduino reculator [en ligne]. La grotte du barbu, 09/07/2012 [20/05/2016]. Disponible sur http://www.lagrottedubarbu.com/2012/07/09/lagrottedubarbu-saison-04-episode-09-arduino-reculator/

Landrault, Simon. Tutoriel Arduino (ou pas) et articles divers [en ligne]. Le blog d'Eskimon, 12/12/2013 [20/05/2016]. Disponible sur http://eskimon.fr/285-arduino-601-le-moteur-courant-continu

Introductin to Arduino TCRT5000 (IR Sensor) [en ligne]. Blog.Huntgang.com, 17/06/2014 [20:05/2016]. Disponible sur http://blog.huntgang.com/2014/06/17/arduino-tcrt5000-build-ir-sensor/

- **Shield :**

Lady ada, Library install [en ligne]. Adafruit, 07/06/2015 [consulté le 20/05/2016]. Disponible sur https://learn.adafruit.com/adafruit-motor-shield/library-install

Lady ada, Adafruit Motor Shield V2 for Arduino [en ligne]. Adafruit learning system, 28/04/2016 [consulté le 20/05/2016]. Disponible sur https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield-v2-for-arduino.pdf

- **S2A :**

Piloter Arduino par Scratch2 [en ligne]. Académie de Nantes [consulté le 20/05/2016]. Disponible sur http://www.pedagogie.ac-nantes.fr/technologies-et-sciences-des-ingenieurs/documentation/didacticiels-tutoriels/piloter-arduino-par-scratch2-819284.kjsp?RH=1160751856953

- **S4A :**

S4A [en ligne]. Citilab, 2015 [consulté le 20/05/2016]. Disponible sur http://s4a.cat

- **Ardublock et S4A :**

Daniel Pers, Logiciels de programmation graphique pour le collège : Scratch pour Arduino (S4A) vs Ardublock [en ligne]. Académie de Poitiers, 22/02/2016 [consulté le 20/05/2016]. Disponible sur http://blogpeda.ac-poitiers.fr/techno-jean-mace/2015/04/12/scratch-pour-arduino-s4a-vs-ardublock/

VII. Annexes : Datasheet

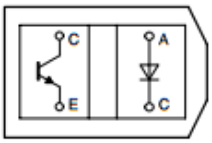
- **Capteur infrarouge TCRT5000 :** (http://www.vishay.com/docs/83760/tcrt5000.pdf)

VISHAY

TCRT5000, TCRT5000L

Vishay Semiconductors

Reflective Optical Sensor with Transistor Output



Top view 19156_1

FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 10.2 x 5.8 x 7
- Peak operating distance: 2.5 mm
- Operating range within > 20 % relative collector current: 0.2 mm to 15 mm
- Typical output current under test: I_C = 1 mA
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC

Pb-free

e2

RoHS COMPLIANT

DESCRIPTION

The TCRT5000 and TCRT5000L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light. The package includes two mounting clips. TCRT5000L is the long lead version.

APPLICATIONS

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

PRODUCT SUMMARY

PART NUMBER	DISTANCE FOR MAXIMUM CTR_{rel} (1) (mm)	DISTANCE RANGE FOR RELATIVE I_{out} > 20 % (mm)	TYPICAL OUTPUT CURRENT UNDER TEST (2) (mA)	DAYLIGHT BLOCKING FILTER INTEGRATED
TCRT5000	2.5	0.2 to 15	1	Yes
TCRT5000L	2.5	0.2 to 15	1	Yes

Notes

(1) CTR: current transfere ratio, I_{out}/I_{in}

(2) Conditions like in table basic charactristics/sensors

ORDERING INFORMATION

ORDERING CODE	PACKAGING	VOLUME (1)	REMARKS
TCRT5000	Tube	MOQ: 4500 pcs, 50 pcs/tube	3.5 mm lead length
TCRT5000L	Tube	MOQ: 2400 pcs, 48 pcs/tube	15 mm lead length

Note

(1) MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS (1)

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
INPUT (EMITTER)				
Reverse voltage		V_R	5	V
Forward current		I_F	60	mA
Forward surge current	$t_p \leq 10\ \mu s$	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25\ °C$	P_V	100	mW
Junction temperature		T_j	100	°C

- **Bouton poussoir :** (http://www.farnell.com/datasheets/921681.pdf)

Tyco Electronics

Switches

Tactile Switches

FSM . SM/JM Series

- Small PCB footprint

FSM . SM/JM LP Series

- Small PCB footprint
- Low profile
- SMT and SMT up pinning

NEW

FSM 6 x 6 Series

- Different actuator and mounting versions
- 105°C versions available

Tactiles			
Style	3.5 x 6	4 x 6	6 x 6
Mounting style	THT, SMT	SMT	THT, SMT
Right angle versions	No	No	Yes
Low profile versions	No	Standard	Yes
Flush actuator version	Yes	No	Yes
Different actuator styles	Yes	No	Yes
ESD grounding term./locating post	No / No	No / No	Optional / No
Sealed versions	Yes	No	Yes
Standard actuation force	1.3, 1.8 N	1.6, 2.6 N	1.0, 1.6, 2.6, 5.2 N
Electrical endurance	up to 100,000 ops.	up to 1,000,000 ops.	up to 2,000,000 ops.
Operating temperature range	-40...+85°C	-40...+85°C	-40...+85°C, (+105°C)
Dimensions horizontal ver. (LxWxH)	3.5 x 6 x 4.3...5 mm	4.2 x 6 x 2.5 mm	6 x 6 x 4.3...17 mm
Dimensions sealed ver. (LxWxH)	4.7 x 6.8 x 4.3...5 mm	-	-
Packaging	Tape and Reel, Bulk	Tape and Reel	Tape and Reel, Bulk

FSM 10 Series

- Different actuator and mounting versions

Tactiles	
Style	12 x 12
Mounting style	THT, SMT
Right angle versions	Yes
Low profile versions	No
Flush actuator version	Yes
Different actuator styles	Yes
ESD grounding term./locating post	No / Optional
Sealed versions	Yes
Standard actuation force	1.6, 2.6 N
Electrical endurance	up to 1,000,000 ops.
Operating temperature range	-35...+85°C
Dimensions horizontal ver. (LxWxH)	12 x 12.5 x 4.3...12 mm
Dimensions sealed ver. (LxWxH)	12 x 12.5 x 4.3...7.3 mm
Packaging	Tape and Reel, Bulk

Catalogue No. 1308111
Issued 03/2008

Dimensions are in mm and are shown for reference purposes only.

Specifications subject to change.

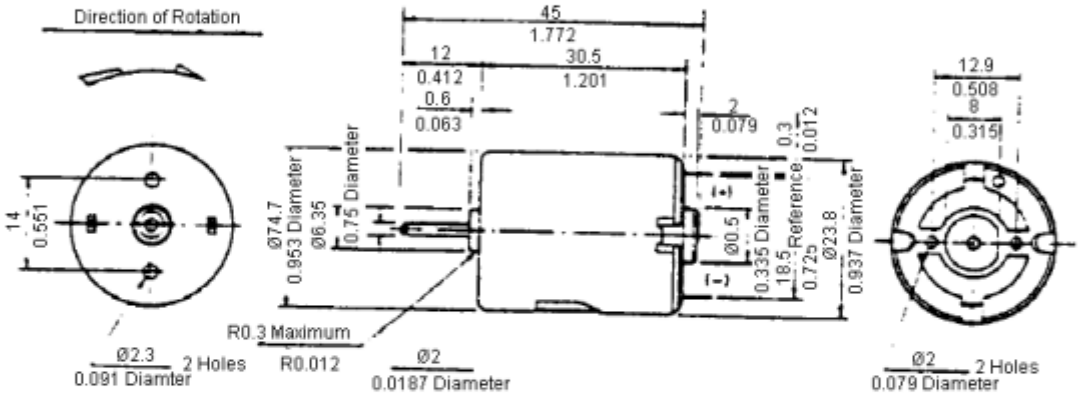
www.tycoelectronics.com

- **MCC :** (http://www.farnell.com/datasheets/1662763.pdf)

Miniature Motors

multicomp

Model MM28



Dimensions : Millimetres (Inches)

Specification Table

Model	Operating Range	Approximately Weight
MM28	3 - 6 V	42 g

Nominal	No Load		At Maximum Efficiency					Stall Torque g-cm
Constant V	Speed rpm	Current A	Speed rpm	Current A	Torque g-cm	Output W	Efficiency %	
3 V	9,600	0.22	8,000	0.99	20	1.64	55.2	112

- **Shield :** (https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield-v2-for-arduino.pdf)

- **Carte Arduino :** (http://www.seeedstudio.com/wiki/Seeeduino_v4.0)