# Introduction

Le but de ce document est de présenter les éléments qui ont été développés au cours de ce projet afin que les personnes le souhaitant puisse continuer le développement. Nous y détaillerons aussi bien les choix techniques que les algorithmes utilisés.

Dans cette introduction nous détaillerons la liste des fonctionnalités sur lesquelles nous avons travaillé, leur priorité ainsi que leur état d’avancement.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Priorité | Etat | Dernière modification |
| Bibliothèques pour les éléments natifs du robot : moteur | Haute | Fait | 16/12/2016 |
| Bibliothèques pour les éléments natifs du robot : capteurs | Haute | Fait | 16/12/2016 |
| Bibliothèques pour les éléments natifs du robot : LED | Haute | Fait | 16/12/2016 |
| Bibliothèques pour les éléments natifs du robot : boutons | Haute | Fait | 16/12/2016 |
| Gestion des combinaisons de bouton | Bas | Fait |  |
| Bibliothèques pour les éléments natifs du robot : buzzer | Bas | Fait | 16/12/2016 |
| Communication Wifi avec le robot | Haute | En cours | 16/12/2016 |
| Programmation du robot en Scratch | Haute | En cours | 16/12/2016 |
| Bibliothèque Scratch : Primaire | Haute | En cours | 16/12/2016 |
| Bibliothèque Scratch : Collège | Haute | En cours | 16/12/2016 |
| Environnement de développement basé sur Scratch | Moyenne | Attente | 16/12/2016 |
| Programmation du robot en Arduino | Haute |  |  |
| Programme de démonstration : Suiveur de ligne | Basse | Fait | 16/12/2016 |
| Programme de démonstration : Evitement d’obstacle | Basse | Fait | 16/12/2016 |
| Programme de démonstration : Joystick (Boutons / Wifi) | Basse | En cours | 16/12/2016 |
| Programme de démonstration : Menu pour programme de démonstration | Basse | Fait | 16/12/2016 |
| Modules MakeBlock externes | Basse | En cours | 09/01/2017 |
|  |  |  |  |

L’ensemble du code du projet est disponible sur GitHub au lien suivant :

<https://github.com/mthSmr/ISEN_m2-2016_robotique>

# Scénario d’utilisation

L’utilisation du robot a été pensée en deux partie distinctes :

* Première utilisation ou mode démonstration :

Les programmes de démonstrations sont chargés dans le robot, permettant ainsi à l’utilisateur de découvrir les possibilités que lui offre le robot. Les différents programmables de démonstration sont accessibles par l’intermédiaire d’un menu dans lequel on circule grâce aux boutons haut et bas du robot. Les programmes de démonstrations sont disponibles en ligne en copier-coller ou en téléchargement sur le site ou le GitHub du robot.

Une fois le robot initialisé l’utilisateur accède à un menu défilant dont chaque élément est distingué par une couleur qui lui est propre. L’utilisateur circule dans le menu grâce au flèches gauche et droite du robot. Il peut ensuite valider son choix en appuyant sur le bouton central du robot. Arrivé au dernier élément, l’utilisateur peut retourner au début du menu en continuant le défilement des options dans le même sens. Actuellement, le robot comporte les options [liste des options] ;

* Mode programmation en Scratch ou en Arduino :

Pour programmer le robot en Scratch, un fichier Arduino utilisant la librairie Firmata pour Arduino doit être télé-versé dans le microcontrôleur [à modifier] avant de lancer le code scratch. L’interface de programmation en Scratch intègre une liste de blocks Scratch spécifiques au robot dans un nouvel onglet intitulé « more blocks » en plus des blocks Scratch présentent par défaut. Cette liste de blocks peut être complétée au besoin. L’interface de programmation Scratch est organisée de la même manière que la version originale développée par le MIT afin de rendre la prise en main facile. Le principe de la programmation Scratch reste la même : cliquer-glisser des blocks est les assembler entre eux pour créer des instructions. Le lancement du programme Scratch se fait au click du drapeau vert.

Pour programmer le robot en Arduino, il suffit de créer un nouveau fichier « .ino », de l’éditer puis de le télé-verser dans le microcontrôleur du robot. Cela peut être fait grâce à l’IDE Arduino ou grâce à d’autres interfaces permettant de développer en Arduino comme l’extension Scratch pour l’IDE Visual Studio, ou l’extension Arduino pour l’IDE Eclipse par exemple selon les préférences de l’utilisateur.

# Les bibliothèques :

Les bibliothèques du robot sont disponibles au téléchargement sur le site du robot.

Une première partie sera une bibliothèque agencée sous forme de classe pour chaque composant présent dans le robot. Cela comprend les classes Sensors, Motor, Led, Button, Buzzer. Chacune de ces classes contient les fonctions nécessaires à l’utilisation des différents éléments du robot (initialisation, envois d’ordre, lecture des valeurs…)

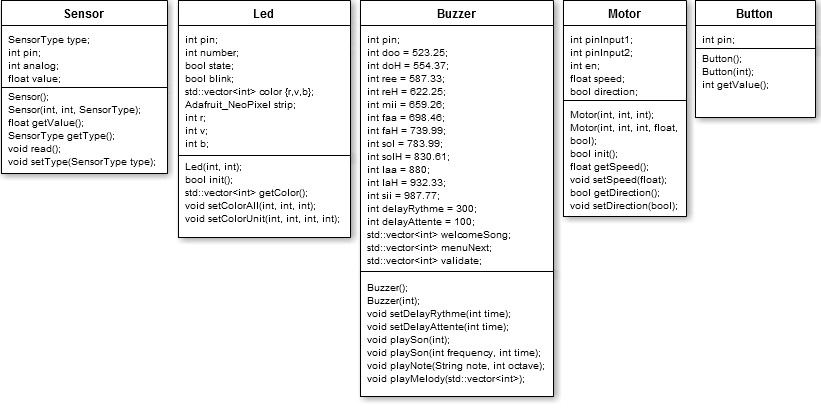


Diagramme de classe 1 : Bibliothèque du robot

La deuxième partie correspond à la démonstration des capacités du robot ainsi que des librairies développées. Ces éléments sont également organisés sous forme de classe.

Cela comprends la classe ButtonPanel dont le but est de gérer si besoin les différentes combinaisons de boutons et de lier les combinaisons à une action, un menu (etc.); La classe Program qui lie entre eux les différents éléments du robot et liste les programmes de démonstration tels que le joystick, le suiveur de ligne ou encore l’évitement d’obstacle;La classe Menu qui est une interface pour choisir et circuler entre les différents programmes de démonstration de la classe Program;

Au lancement du robot, le programme main\_robot.ino se lance. Ce fichier suit l’organisation par défaut des fichier Arduino (fonctions setup() et loop()).

L’ensemble des composants utilisés par le robot (moteurs, leds, buzzer...) sont dans un premier temps déclarés comme variable globales puis sont initialisés dans la fonction setup(). On y spécifie le numéro de pin, leurs valeurs clés (ex: les seuils de lecture pour les capteurs), on y utilise également les fonctions de la bibliothèque Arduino nécessaire à leur initialisation.

Le programme robot\_main subdivisé en deux parties:

* Le setup() dans lequel sont déclarés chaques composants du robot (moteurs, leds, buzzer...) en leurs attribuant un numéro de pin ainsi que les diverses valeurs qui leurs sont propres (ex: les seuils de lecture pour les capteurs). Les différents éléments sont transmis à une instance de la classe programme afin de les lier entre eux.
* Le loop() dans lequel on appelle le menu pour la démonstration. Il est également possible d’y faire des appelles directs de bibliothèques diverses.

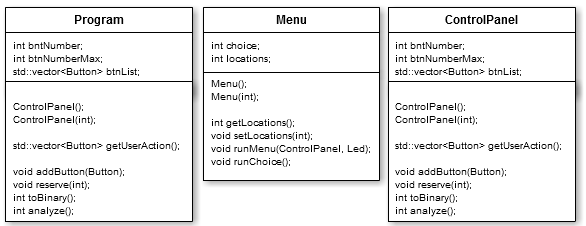


Diagramme de classe 1 : Démonstration

# La Programmation en Scratch

## Programmation du robot en Scratch**:**

Un fichier Arduino permettant de transcrire les blocks Scratch en code Arduino doit être télé-versé au préalable dans le microcontrôleur. Ce programme utilise une série de bibliothèques en C++/Arduino développée pour le robot.

## Les bibliothèques Scratch :

La forme des blocks Scratch sont créés en deux étapes.

Dans un premier temps, leur forme graphique est déclarée dans le fichier *IHM\Partie\_PC-JavaScript\Block\_description\s2a\_functions.s2e* .

Dans un second temps, les blocks sont décrit pour Firmata dans différents fichiers JavaScript dans le dossier *IHM\Partie\_PC-JavaScript\Fonction\_extension\_librairie\Lib\_fonctionnel* . Ce Dossier regroupe l’ensemble des bibliothèques utilisées par le robot.

Bibliothèque Scratch Primaire :

Les descriptions des fonctions Scratch pour primaire sont dans le fichier *lib\_primaire.js*.

Liste des block ajoutés à Scratch pour la librairie primaire :

* Avancer à une vitesse donnée
* Reculer à une vitesse donnée
* Avancer d’une distance donnée
* Reculer d’une distance donnée

Bibliothèque Scratch Collège :

Les descriptions des fonctions Scratch pour primaire sont dans le fichier *lib\_college.js*.

Liste des block ajoutés à Scratch pour la librairie collège :

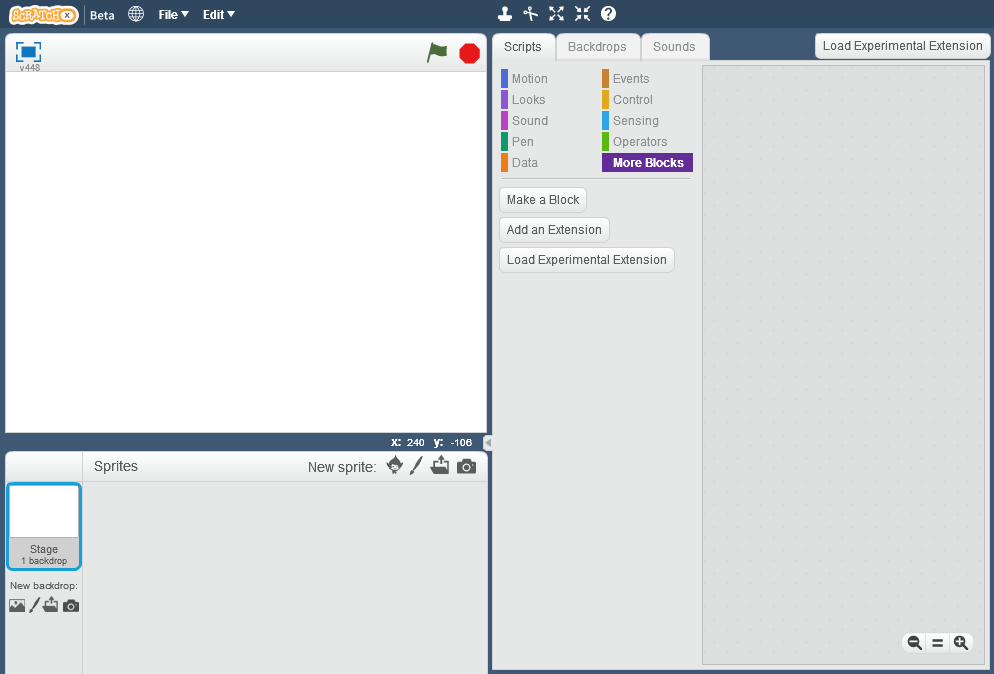
* Modification de l’état d’un pin digitale
* Modification de l’état d’un pin analogue
* Jouer un son sur un pin donné
* Arrêter le son sur un pin donné
* Bouger un servo-moteur sur un pin donné à un degrés donné
* Lecture d’un pin digital donné
* Lecture d’un pin analogue donné

## Environnement de développement basé sur Scratch :

Maquette :

L’environnement de développement est une interface basée sur Scratch. Cette interface est développée en Flash [version]. Son organisation est identique au Scratch officiel.

arborescence

Vue 1 : Ecran à l’ouverture de l’application – Ecran de base de Scratch

Licences et copyright :

# La programmation en Arduino