Master 1 - TER

GoodLua

Bovie Pierre-Edouard
Labourbe Loïc
Maslowski Antoine
Roche Julie

Enseignants : Couvreur Jean-Michel
Dabrowski Frederic

7 Février 2018 - 25 Mai 2018

Résumé du projet

L'objet du projet est la conception d'un système pour la création et le lancement à distance de programme Lua s'exécutant sur un robot. Le système matériel est composé d'une tablette Androïd et "un robot mobile à base pcDuino équipé de divers capteurs. Le logiciel de la tablette offrira la possibilité :

— de concevoir des programmes

— de charger à distance le programme sur le robot

— de télécommander le robot

Le système logiciel sur le robot permettra de recevoir et d'exécuter les programmes Lua transmis par la tablette.

Introduction au domaine

Besoins fonctionnels

- **exécuter du code**

  description : exécuter du code eLua sur le PcDuino

  priorité : très haute

  justification : cette fonctionnalité est la principale. Nous devons être en mesure de faire fonctionner le robot grâce à du code eLua avant toute autre chose.

- **connecter android au robot grâce au wifi**

  description : utiliser la carte wifi du PcDuino pour la connecter à un appareil androïd.

  priorité : haute

  justification : le code crée via l'application androïd doit être transmis au robot pour être exécuté. Une solution filaire étant peu pratique, nous le connecterons via wifi.

- **transformer blockly en eLua**

  description : traduire les blocs de la librairie Blockly dans le langage Lua, plus spécifiquement en eLua.

  priorité : moyenne

  justification : le robot étant "guidé" par du code eLua, il est nécessaire de procéder à une traduction du programme visuel dans ce langage avant de l'envoyer au robot pour être exécuté.

- **créer un programme**

  description : la création d'un nouveau programme blockly via l'application androïd.

  priorité : moyenne

justification : c'est la fonctionnalité de base de l'application, il faut pouvoir créer de nouveaux programmes en vue de les écrire et de les exécuter.

- **sauvegarder / charger des programmes**

  description : l'utilisateur pourra sauvegarder les programmes sur lesquels il travaille, et reprendre sa progression là où il était en chargeant un programme parmi ceux enregistrés.

  priorité : basse

  justification : cela permettra de travailler sur un programme en plusieurs fois sans risque de perdre son travail. Cette fonctionnalité était à la base en option, mais l'architecture de notre application nécessite de pouvoir l'utiliser.

Prototype - l'interface de l'application Androïd :
En ouvrant l'application, l'utilisateur se verra proposer de créer un nouveau programme, il devra alors lui donner un nom, ou alors de charger un programme déjà existant, en choisissant dans une liste. Une fois le programme sélectionné ou créé, il aura le choix de le modifier ou de l'exécuter. [Figure 1]

Dans le cas du choix de modifier le code, on ouvrira l'éditeur visuel contenant Blockly avec l'ensemble des blocs disponibles dans une barre latérale gauche. Un menu permettra également de sauvegarder le programme en cours d'édition ou de l'effacer. [Figure 2]

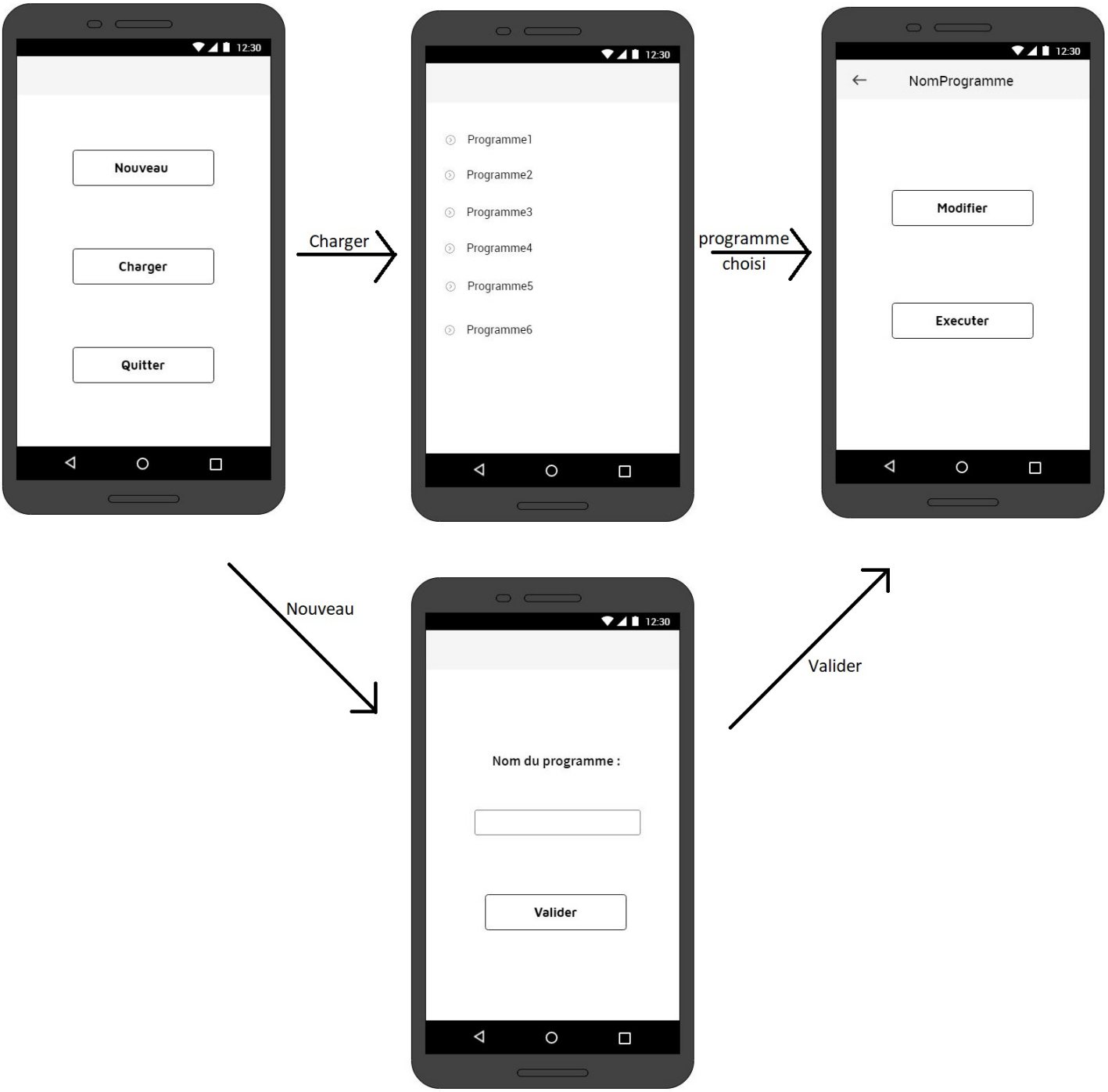

FIGURE 1 – Maquette 1ère partie

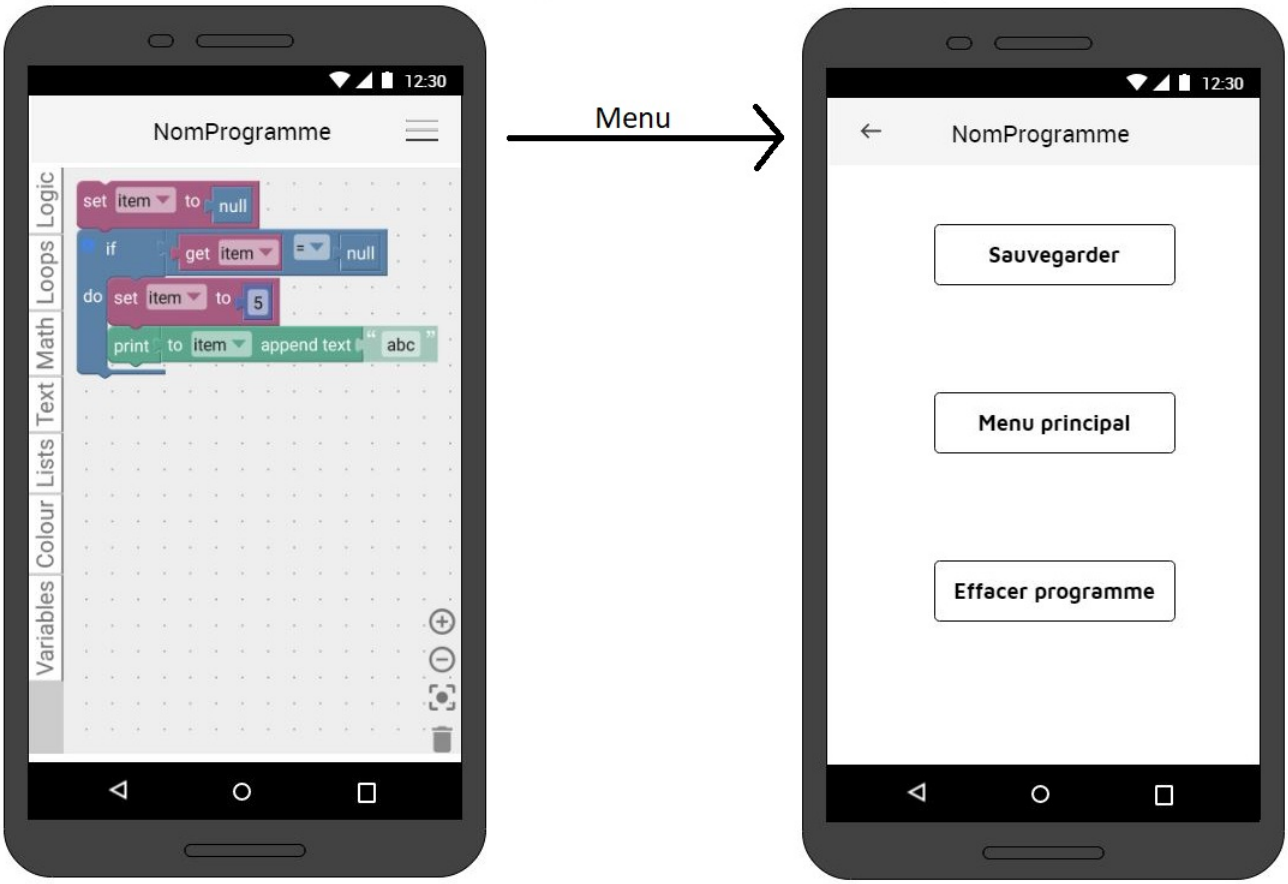

FIGURE 2 – Maquette 2ème partie

Besoins non fonctionnels

environnement de travail : Nous travaillons sous une distribution Ubuntu. Pour la partie programmation en eLua, il est possible d'utiliser n'importe quel éditeur de texte, puis de compiler le fichier.

exécution en temps réel / rapidité : Il serait préférable que l'exécution d'un programme se fasse de manière assez rapide, pour ne pas aller contre les attentes de l'utilisateur. Par exemple, si le programme utilise des capteurs, mais qu'une lenteur dans le programme fait qu'un obstacle n'est pas détecté à temps, cela nuira à l'expérience de l'utilisateur. De même, l'un des buts de l'application, est de pouvoir exécuter une séquence, ou de donner des ordres au robot en temps réel.

simplicité d'utilisation : L'application doit être utilisable par des utilisateurs sachant programmer, qui pourront créer un programme complexe à partir des blocs basiques, mais elle doit également être accessible à des utilisateurs débutants en programmation, qui pourront utiliser des blocs représentant des fonctions toutes faites concernant le robot (par exemple : allumer LED). Les blocs et l'interface devront donc être instinctifs et simples d'utilisation.

Analyse de l'existant

Nous avons reçu pour ce projet, un kit matériel qui forme le robot.
Pour nous aider à découvrir son utilisation, nous avons étudié le projet d'algorithmique répartie des étudiants de la promotion 2015 des master 1 informatique [1] dans lequel ils manipulent un robot semblable au notre, mais avec des des langages différent du notre (eLua) puisqu'ils l'ont contrôlé avec du python et du C++.

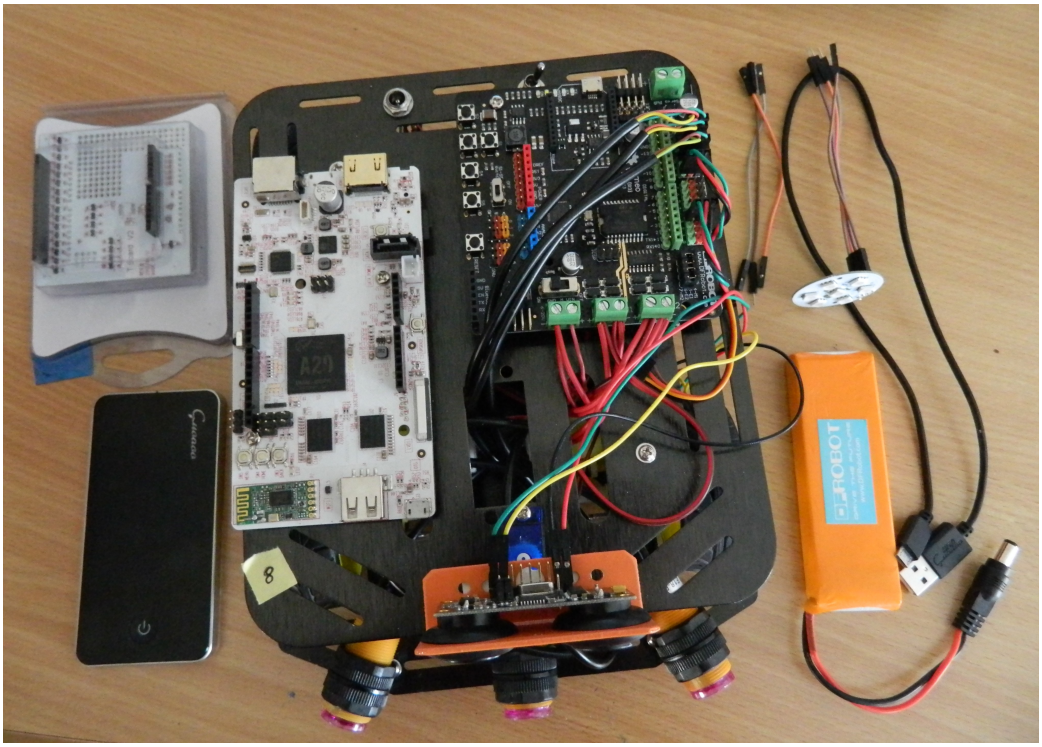

FIGURE 3 – Notre materiel

Voici la liste du matériel dont nous disposons et dont nous devrons nous servir pour ce projet :

Le pcDuino : La carte pcDuino V3 est un mini PC possédant un processeur dual core A20. Une fois branchée à un écran, un clavier et une souris, grâce à son port USB, elle s'utilise de la même manière qu'un PC sous une distribution Ubuntu. Elle possède un lecteur de carte micro-sd, car c'est dans cette carte que se situe le système d'exploitation du pcDuino. Le pcDuino posssède également un port ethernet ainsi qu'une carte wifi, ce qui permet de se connecter à un réseau facilement. Pour le faire fonctionner, il suffit de le brancher à un chargeur USB par son port micro USB. Il possède enfin des broches arduino, en plus de ses interfaces de communication, permettant de faire de la programmation arduino.[2]

L'arduino : L'arduino Romeo V2.2 (R3) [3] est un micro-contrôleur tout en un spécialement crée pour la robotique, qui fournit des interfaces de puissance pour contrôler des moteurs.

La plateforme mobile : Notre robot est composé d'une châssis Baron - 4WD auquel sont associées 4 moteurs reliés à 4 roues codeuses. [4]

La LED : Le disque contient 7 LEDs RGB [5], que l'ont peut allumer grâce à 3 pins, un pour chaque couleur, ainsi qu'avec un pin d'alimentation.

Les capteurs infrarouges : Les 3 capteurs possèdent une LED qui s'allume si un objet est dans le champs de détection. Ils ont une sortie binaire, selon si un obstacle est détecté ou non. Leur distance de détection peut être de 3 à 80 cm, et se règle manuellement avant l'utilisation. [6]

Le capteur ultrasons : Lié à l'arduino, il détecte des températures allant de -10 à environ 70 degrés celsius, à une distance de 4cm, jusqu'à 5m. [7]

Blockly : Nous utilisons la bibliothèque Blockly [8] pour avoir un éditeur de code visuel. Blockly se présente sous forme de pièces de puzzle, chacune représentant un concept de code, comme une fonction, une condition, une boucle ou une variable. L'utilisateur n'a donc qu'à sélectionner et faire glisser les pièces pour les placer et les imbriquer, afin de former un programme. Il n'a donc pas à s'occuper de la syntaxe, puisque les formes des pièces lui indique où il peut les placer ou non. Une fois le

programme formé, Blockly peut le transformer en différents langages (JavaScript, Python, PHP, Dart) dont celui qui nous intéresse pour ce projet, le Lua.
Grâce à cette bibliothèque, nous avons la possibilité de créer les blocs de notre choix, correspondant à des fonctions à appliquer sur le robot, d'intégrer l'éditeur Blockly à notre application Androïd, et de générer le code à exécuter.

Planning

Planning effectif jusqu'au rendu du présent rapport :

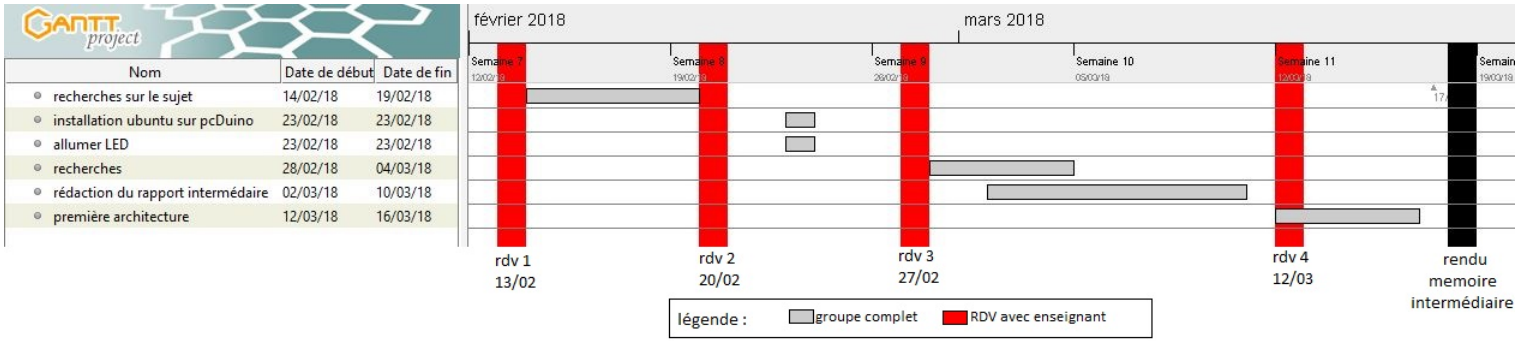

FIGURE 4 – Planning du 13 février au 18 mars

Planning prévisionnel pour la suite du projet :

Bibliographie

[1] M1 info promo 2015. Projet d'algorithmique répartie 2015, m1 info, université d'orléans, aug 2015. https ://github.com/blgatelierl2/robot.

[2] X. HINAULT. Le pcduino v3 (a20), jul 2014. http ://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php ?n=MAIN.PCDUINO.

[3] dfrobot. Romeo v2-all in one controller (r3), may 2017. https ://www.dfrobot.com/wiki/index.php/Romeo_V2-All_in_one_Controller_(R3)_(SKU :DFR0225).

[4] dfrobot. Baron-4wd arduino mobile robot platform with encoder). https ://www.dfrobot.com/product261.html.

[5] dfrobot. Light disc, june 2017. https ://www.dfrobot.com/wiki/index.php/Light_Disc_(SKU :DF

[6] dfrobot. Adjustable infrared sensor switch, may 2017. https ://www.dfrobot.com/wiki/index.php/Adjustable_Infrared_Sensor_Switch_(SKU :SEN001

[7] dfrobot. Urm37 v3.2 ultrasonic sensor, may 2017. https ://www.dfrobot.com/wiki/index.php/URM37_V3.2_Ultrasonic_Sensor_(SKU :SEN0001).

[8] Google for education. Blockly, apr 2015. https ://developers.google.com/blockly/.