

# Cafetière connectée commandée par Bluetooth

## Hybridation technologique

Culture Sciences  
de l'Ingénieur

Antony CHARBONNIERAS - Quentin HEBERT

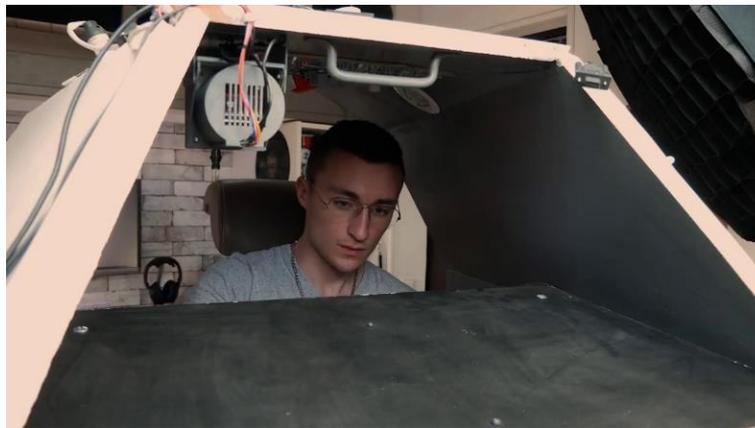
Édité le  
07/02/2022

école \_\_\_\_\_  
normale \_\_\_\_\_  
supérieure \_\_\_\_\_  
paris-saclay \_\_\_\_\_

### 1 – Introduction

Dans cet article nous vous proposons un guide succinct illustré par une vidéo de présentation générale, permettant d'avoir les connaissances élémentaires en programmation d'applications destinées aux Smartphones Android, et permettant, dans notre cas précis, de communiquer avec un système Bluetooth « homemade » adapté à un appareil déjà existant !

Quentin Hebert a 19 ans et est actuellement en dernière année de DUT GEII à l'IUT de Cachan. Depuis le collège il « bricole » différents supports et dispositifs, en particulier des projets autour de la technologie Arduino, qu'il présente sur sa chaîne Youtube « **Quentin Maker** ».



*Image extraite de la vidéo de présentation de Quentin Hebert*

**Retrouvez la vidéo de présentation de Quentin Hebert sur le site [Culture sciences de l'Ingénieur](https://www.culture-sciences-de-lingenieur.fr) ou sur sa chaîne Youtube : <https://www.youtube.com/watch?v=3w6G-McFG34>**

La première « victime » de cette hybridation technologique : une cafetière célèbre !

Une démarche similaire peut évidemment être adaptée sur différents modèles, en dehors bien sûr des versions italiennes à moka ! :)



*Image extraite de la vidéo de présentation de la cafetière commandée par Bluetooth*

Retrouvez la vidéo de présentation de la cafetière commandée par Bluetooth sur le site [Culture sciences de l'Ingénieur](http://Culture sciences de l'Ingénieur) ou sur Youtube : <https://www.youtube.com/watch?v=OJ8geif8pEw>

**ATTENTION !** Ce projet demande l'ouverture et la modification d'un appareil alimenté par le réseau domestique (230 V - 50 Hz) !  
Ainsi nous déconseillons la réalisation de ce projet à des novices.

Nous avons regroupé nos documents et fichiers complémentaires utiles pour la bonne réalisation de ce tutoriel dans un fichier « **CafetiereBluetooth.zip** » disponible sur ce lien : [https://profge2.iut-cachan.u-psud.fr/CSI\\_3EI/TutoAI/CafetiereBluetooth.zip](https://profge2.iut-cachan.u-psud.fr/CSI_3EI/TutoAI/CafetiereBluetooth.zip)

## 2 – Composants et circuit électrique

L'objectif n'est pas ici de refaire toute l'électronique de la cafetière mais d'en créer une nouvelle en adaptant en parallèle ses circuits initiaux, ce qui permettra de continuer d'utiliser, directement sur la machine, les fonctionnalités d'origine indépendamment de la commande à distance. Un exemple de prototype de la carte électronique est donné sur la figure 1.

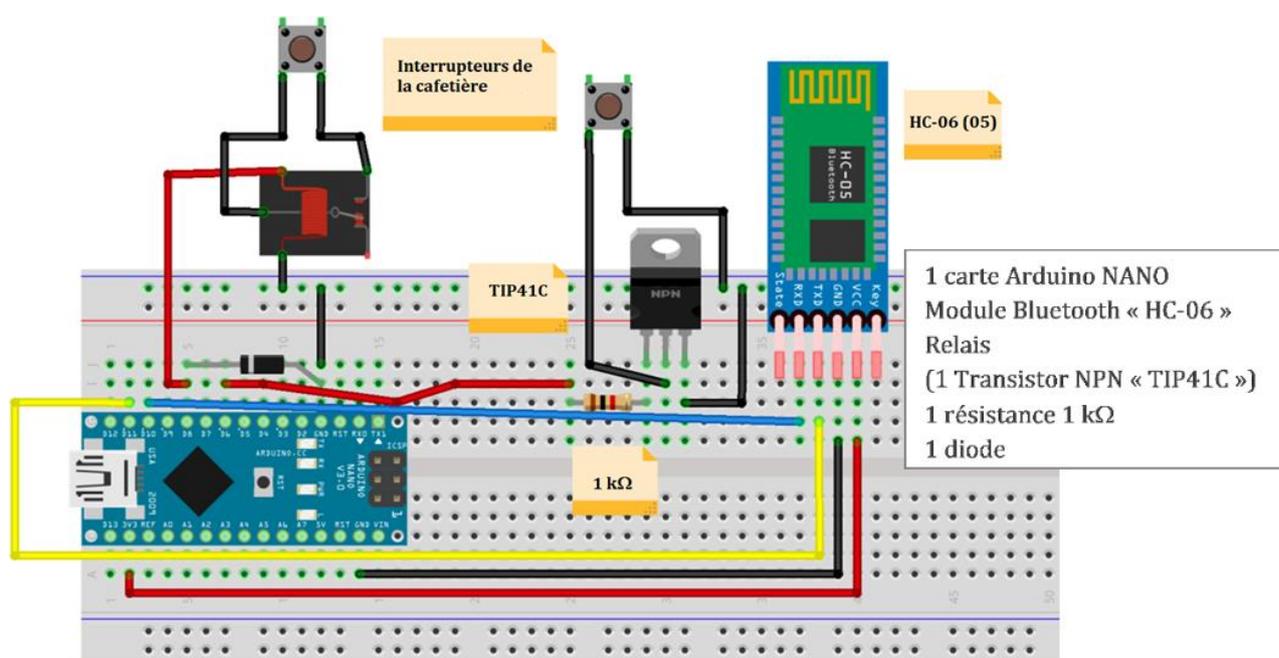


Figure 1 : Prototype sur plaquette Labdec

L'élément central de ce montage est la carte électronique programmable Arduino NANO équipée, entre autres, d'un microcontrôleur (ATmega328), le cerveau du circuit. Elle va nous permettre de communiquer avec le module Bluetooth HC-06 (souvent utilisé avec ce type de cartes...) afin de recevoir les instructions envoyées par l'utilisateur depuis l'application Android, et de commander les interrupteurs de la cafetière. Nous verrons que la programmation se rapproche beaucoup de celle en langage C.

Vous trouverez des compléments concernant les caractéristiques de la carte Arduino Nano sur le lien suivant : <https://www.arduino-france.com/review/arduino-nano-avantages-inconvenients-utilisation-et-fonctionnement/>

Pour commander les interrupteurs, il a suffi de leur ajouter deux composants en parallèle qui joueront également un rôle d'interrupteur, ce qui permet aussi de continuer à utiliser la cafetière normalement, directement depuis les boutons d'origine. En effet, rappelons que si deux

interrupteurs A et B sont en parallèle, la fermeture de A et/ou B permettra d'alimenter l'ampoule représentée sur la figure 2.

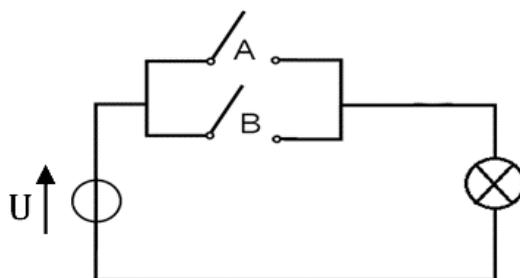


Figure 2 : Interrupteurs en parallèle, rappels...

Pour des raisons de compatibilité électronique, l'un des interrupteurs est un transistor NPN et l'autre un relais. Ces deux composants sont donc des interrupteurs commandés électriquement ayant chacun des avantages et des inconvénients (constante de temps, usure, sensibilité électromagnétique, isolation partie commande/partie puissance...), mais qui dans cette utilisation ne les mettent pas en concurrence. En revanche, le relais est un composant « indépendant » qui offre l'avantage de ne pas devoir être obligatoirement relié à la masse commune donc une grande souplesse quant à son implantation dans le circuit initial. Ce fut l'une des subtilités de ce montage.

- Le transistor TIP41C est connecté à la sortie D6 de la carte Arduino, il permet de commander la mise en marche de la résistance de chauffe de la cafetière ;
- Le relais est lui relié à la sortie D8 et permet d'activer la pompe de la cafetière, donc de « lancer le café » !

**La diode** aux bornes du relais est une diode dite de « roue libre » qui permet de limiter les pics de tension dus aux effets inductifs de la bobine du relais ; elle n'est passante que dans un sens, afin d'assurer la continuité du courant électrique lorsque la tension de commande du relais est coupée à la sortie du microcontrôleur (extinction de la cafetière par exemple) !

**La résistance** de 1 k $\Omega$  est en série avec la grille du TIP41C afin de limiter le courant de commande. Il est également possible d'ajouter une résistance en « pull-down » entre la grille et la masse afin d'assurer la décharge de la capacité parasite du transistor.

### 3 – Programmation de la carte Arduino

Le programme permet de traiter l'ordre envoyé à distance par Bluetooth depuis le Smartphone (Android), de commander la mise en marche de la cafetière (la chauffe) et la coulée du café.

Il existe une multitude de tutoriels divers et variés disponibles sur Internet concernant la programmation des cartes Arduino. Vous trouverez plus loin en webographie quelques exemples non exhaustifs pour les débutants et les plus confirmés !

Mais évidemment, qui dit « programmation » dit « environnement de programmation ». C'est le rôle du logiciel gratuit « Arduino IDE » que l'on télécharge directement sur le site officiel : <https://www.arduino.cc/en/software>.

Il suffit de suivre les indications données pendant le processus d'installation. Aucun piège, c'est ultra simple !

Une fois l'application installée, il suffit de l'ouvrir et de configurer le logiciel en lui indiquant votre type de carte.

- Cliquez sur [Outils], sur [Type de carte] ;
- Une fois la Nano sélectionnée, la cas « processeur » doit afficher « ATmega328 ».

Enfin, connectez la carte Arduino au port USB de votre ordinateur afin que ses pilotes s'installent puis, toujours dans [Outils], sélectionner le nouveau port qui est apparu correspondant à la carte fraîchement branchée.

Le programme final permettant de commander la chauffe de l'eau et le lancement du café, par Bluetooth via l'application Smartphone que nous allons créer dans la partie suivante, est directement disponible dans le fichier : Cafetière **Connectee-Programme-Arduino.ino** (dans CafetiereBluetooth.zip, téléchargeable sur ce [lien](#)).

Il suffit donc de le charger dans Arduino IDE depuis le menu [Fichier], de le compiler puis de le téléverser vers la cible.

Programmer des Arduino en 10 min !! Tuto débutant - Chaîne Abrège :

<https://www.youtube.com/watch?v=ihhID9Uvunk>

Mes premiers pas avec Arduino : tutoriel ou cours pour débutant en Arduino :

<https://www.robot-maker.com/ouvrages/tuto-arduino/>

Programmation Arduino : présentation pour les débutants :

<https://putaindecode.io/articles/programmation-arduino-presentation-pour-les-debutants/>

Cours complet pour apprendre à programmer un Arduino :

<https://arduino.developpez.com/tutoriels/cours-complet-arduino/?page=programmer-arduino>

## 4 – Programmation de l'application Android pour le Smartphone

Une fois le programme implémenté dans l'Arduino, il reste à créer l'application permettant de communiquer avec le module **Bluetooth HC-06**.

Une des méthodes les plus simples est d'utiliser l'outil de programmation en ligne « **App Inventor** » développé par Google. Cet environnement est équipé d'une interface entièrement graphique sans la moindre ligne de code à taper. La « programmation » consistera donc uniquement, dans la partie « **Blocks** » (figure 4), à assembler des briques en les agencant logiquement les unes par rapport aux autres afin de contrôler le comportement des « clics » ou actions choisies par l'utilisateur à partir de l'interface homme-machine dont nous créerons le visuel dans la partie « **Designer** » (figure 3) de l'application « App Inventor ». Il est donc parfaitement adapté pour l'initiation à la programmation. (Lien site App Inventor : <https://appinventor.mit.edu/explore/get-started>).

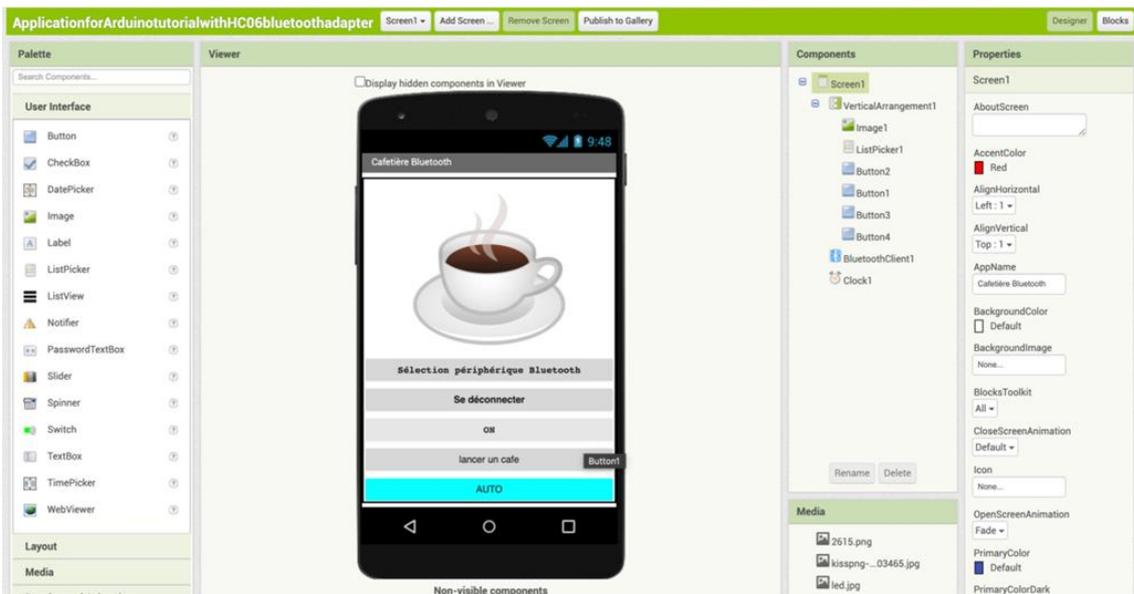


Figure 3 : Environnement graphique App Inventor, « mode Designer »

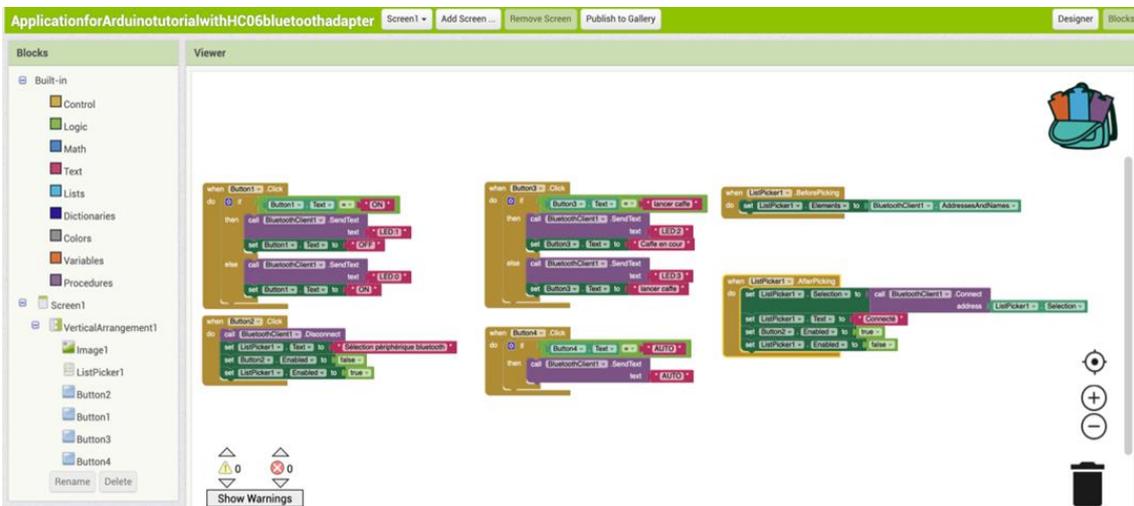


Figure 4 : Environnement graphique App Inventor, « mode Blocs »

En parallèle de la « programmation » en ligne, il est possible de visualiser au fur et à mesure le résultat soit directement grâce à un émulateur à installer sur votre ordinateur, soit directement depuis votre Smartphone en téléchargeant l'application adéquat depuis le Play Store.

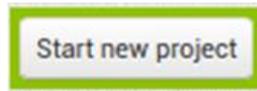
Encore une fois vous trouverez facilement plusieurs tutoriels de qualité pour compléter et approfondir les informations que nous donnons rapidement ici. Voici par exemple une vidéo très détaillée disponible sur la chaine Youtube « U=RI » utilisant également comme base de réflexion la communication entre un Arduino et un Smartphone Android via Bluetooth : <https://www.youtube.com/watch?v=jR4wX7GF9gQ>

#### 4.1 - Les étapes de conception : let's go now!

Pour démarrer avec « App Inventor », vous devez vous connecter avec votre compte Google. Si vous n'en avez pas, la création est très simple, rapide et gratuite ! Notez qu'il est possible de configurer la langue de l'environnement de travail, nous avons conservé ici la version originale qui est en anglais par défaut.

Lien vers le diaporama du tuto que nous avons fait : [https://profge2.iut-cachan.u-psud.fr/CSI\\_3EI/TutoAI/](https://profge2.iut-cachan.u-psud.fr/CSI_3EI/TutoAI/)

1. Pour créer votre application il suffit de cliquer sur



2. Donner un nom à votre projet : « CafetiereConnectee » par exemple

Une fenêtre s'ouvre comportant plusieurs parties, panneaux de travail, dont une centrale « Viewer » habillée d'un visuel de Smartphone, comme sur la figure 3, mais avec un écran totalement vierge ! Cette page va servir à concevoir l'esthétique de l'interface graphique homme-machine.

Le panneau de travail « Palette » propose les différents objets, fonctions que nous allons déposer au fur et à mesure sur le panneau central principal afin d'habiller visuellement notre application Smartphone.

Dans un premier temps, en sélectionnant sur « Screen1 » dans le panneau « Components », nous allons pouvoir renommer la scène principale dans laquelle nous pourrions déposer des boutons, des images, du texte... Bref, ce qu'on appellera de façon générale des objets ! Le nouveau nom choisi correspondra au nom de notre application. Choisissons par exemple « Cafetière Bluetooth », que nous écrivons dans le panneau « Properties », dans la zone de texte « Title » !

Vous pouvez vous amuser à modifier les autres paramètres qui correspondent en général à de la mise en page (couleur, police, taille, thèmes...).

Une fois votre espace de travail de base paramétré selon votre bon goût, nous allons commencer par définir un élément permettant de définir la façon dont les objets vont se positionner sur l'écran ; dans notre cas, un arrangement de haut en bas.

3. Cliquez sur l'onglet « Layout » dans le panneau « Palette ». Sélectionnez puis glissez l'icône « VerticalArrangement » sur l'écran vierge de l'illustration du Smartphone du panneau central principal, puis configurez les options de cet objet comme sur la figure 5.

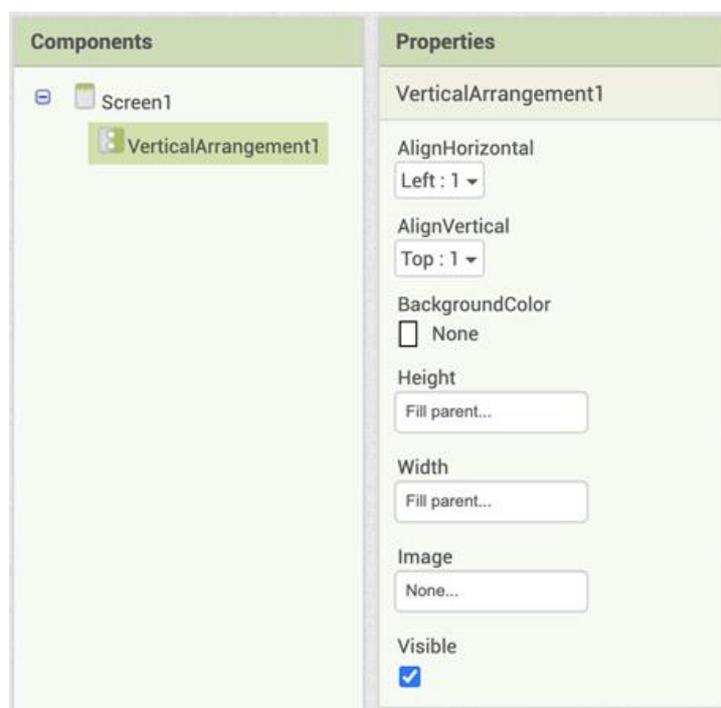


Figure 5 : Option de l'objet VerticalArrangement

Maintenant il nous faut ajouter 4 boutons : pour se déconnecter, lancer la chauffe, faire couler l'eau et un dernier pour lancer le mode automatique !

4. Dans le panneau « **Palette** », faites glisser l'icône « **Button** » sur l'écran du simili Smartphone ; répétez l'action 3 fois.
5. Nous allons ajouter un peu de déco' en plaçant une image, le logo par exemple, disponible dans le fichier dans CafetiereBluetooth.zip (téléchargeable sur ce [lien](#)). Vous pouvez évidemment utiliser l'image de votre choix !
  - Dans le panneau « **Palette** », faites cette fois glisser l'icône « **Image** » au-dessus du premier bouton par exemple !
  - Sélectionnez votre image dans le champ « **Picture** » du panneau « **Properties** », « Upload File » puis « Ok ».
  - Pour adapter l'image à l'écran sélectionnez « Fill parent » dans les champs « Width » et « Height ».
  - Vous pouvez ensuite la positionner verticalement comme vous le souhaitez sur l'écran fictif du Smartphone.
6. Renommez respectivement les 4 boutons, par exemple : **Déconnexion** (Button1), **Cafetière ON** (Button2), **Lancer le café** (Button3) et **Auto** (Button4)
7. Nous allons maintenant personnaliser l'icône de démarrage de notre logiciel, celle sur laquelle on clique pour le lancer.
  - Cliquez sur « Upload File » du panneau « **Media** », sélectionnez votre image puis validez ;
  - Cliquez sur « Screen1 » du panneau « **Components** », sélectionnez le champ « Icon » et choisissez l'image que vous venez de télécharger dans votre projet.

Il nous manque encore un élément important à ajouter, celui qui va permettre de connecter notre Smartphone au système Bluetooth de la cafetière, plus précisément au connecteur HC-06 qui « sera lu » par la carte Arduino Nano grâce à notre programme réalisé dans la partie précédente.

8. Cliquez sur l'icône « ListPicker » dans l'onglet « User Interface » du panneau « **Palette** », et glissez-la sur l'écran fictif. Ce 5<sup>ème</sup> bouton, particulier, permet lorsqu'on le sélectionne de dérouler une liste d'options de choix pour l'utilisateur, mais il est configurable comme les autres vus juste avant. Nous le nommerons par exemple « Module Bluetooth ».
9. Redimensionnez si besoin votre image « déco' » en modifiant les attributs « Height » et « Width ».

Comme pour les boutons, nous allons maintenant importer deux autres icônes sur le visuel fictif du Smartphone. Cette fois, elles ne seront pas graphiques. Ces fonctions apparaîtront seulement dans le panneau « **Components** », à la suite des précédents composants.

La première fonction est simplement un client Bluetooth qui lancera la recherche des périphériques Bluetooth disponibles autour du Smartphone, en particulier le Bluetooth du module HC-06 que nous avons connecté à notre carte Arduino !

10. Cliquez sur l'icône « BluetoothClient » dans l'onglet « Connectivity ». Faites glisser sur l'espace de travail.

Et la dernière fonction, une horloge, qui permet de déclencher un timer à intervalles réguliers et également d'effectuer d'éventuels calculs fonctions du temps, ou de synchroniser des actions. Par défaut la base de temps, la période, est de 1000 ms = 1s (1Hz).

11. Cliquez sur l'icône « Clock » dans l'onglet « Sensor », faites également glisser sur l'espace de travail.

Rien d'autre à configurer pour le moment, en revanche n'hésitez pas à modifier et jouer avec les paramètres visuels des différents composants (couleur, taille, police...).

En tout cas, la partie « graphique » est terminée !!!!

## 4.2 - Les étapes de programmation de cette application : « alive! it's alive! »

En cliquant sur le bouton « Blocks » en haut à droite, l'environnement passe en mode « programmation » !

Cette partie du logiciel « App Inventor » est basée sur la philosophie « Scratch », un logiciel de programmation visuelle destiné initialement aux enfants, dont le principal intérêt est cette simplicité de manipulation permettant de faire un programme sans se préoccuper de syntaxe, en utilisant des « blocs de commande » (figure 4).

Le menu de gauche contient deux parties très importantes pour la suite de notre aventure...

Dans la partie « Screen1 » apparaissent les différents éléments que nous avons positionnés précédemment dans le mode « Designer ». Un clic sur l'un des boutons affiche une liste de plusieurs blocs dédiés, spécifiques à cet élément permettant de configurer son apparence (message, couleur...) et les actions qui lui sont associées (lancer le café, se connecter au Bluetooth...).

La partie « Built-in » contient elle, plusieurs icônes de diverses couleurs permettant d'avoir accès aux instructions globales de programmation. Par exemple « Control » contient les blocs de conditions (if, else...), « Logic » les blocs permettant de gérer les conditions et « Text » de rajouter des zones de textes dans les blocs.

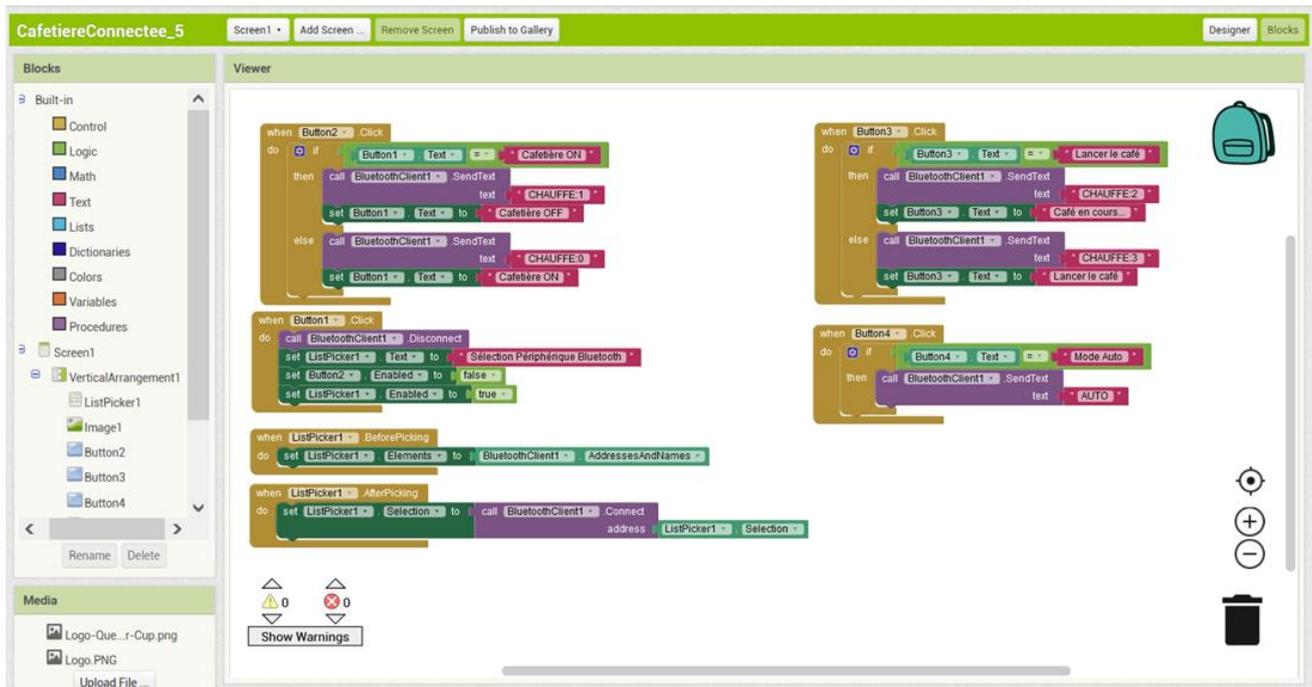


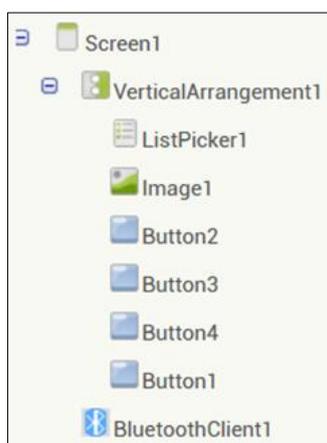
Figure 6 : Programme final en mode « Blocks »

L'idée de ce programme est très simple, quand on appuie sur un des boutons de l'application, celle-ci envoie un message au module HC-06 via le Bluetooth !

Rappelons les informations à transmettre en s'appuyant sur le cahier des charges que nous avons fixé dans la partie « programmation de la carte Arduino ».

- « **Button1** » correspond au bouton de Connexion/Déconnexion et permet à l'application de se connecter au module Bluetooth.
- « **Button2** » (initialement « ON ») : quand on clique sur « ON » l'application envoie le message "CHAUFFE:1" sinon "CHAUFFE:2". Ce message est récupéré par l'Arduino et est interprété par le programme (fichier .ino dans CafetiereBluetooth.zip, téléchargeable sur ce [lien](#)).
- « **Button3** » est le bouton de l'application qui va envoyer la demande « Lancer le café ». On remarquera que son bloc (*boucle When*) est le même que pour le « Button2 », seuls les messages envoyés changent.
- « **Button4** » n'est pas obligatoire mais l'option est pratique puisqu'elle permet d'activer le mode « Automatique » : mise en route de la cafetière (préchauffe) puis lancement du café !

Dans la partie « Screen1 » nous retrouvons des éléments et en particulier ceux que nous avons définis dans la partie « Designer ». Il suffit de cliquer sur l'un d'entre eux pour que des « Blocks » (associés à ces éléments) de différentes couleurs apparaissent.



Il suffit de cliquer dessus pour les positionner ensuite sur l'espace de travail. Les « Blocks » s'encastrent les uns dans les autres en fonction de leur spécificité (condition, texte, comparaison, instructions, appels, commandes...). Ils sont équipés de « Clés puzzle » qui permettent ou non de les associer entre eux !

Une fois un « Bloc » terminé, il est possible, comme tous les objets manipulables sur l'espace de travail, d'effectuer les copier-coller habituels de la bureautique traditionnelle.

Passons à la visite guidée à travers le tutoriel en vidéo disponible à partir de ce lien : <https://youtu.be/00gPa7AUzdQ>

Pour bien comprendre la programmation des blocs, il est nécessaire d'avoir le programme de l'Arduino sous les yeux afin de repérer les actions et les commandes.

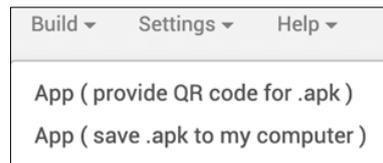
Rappelons que les deux blocs « ListPicker1 » permettent de lancer la détection des connexions Bluetooth disponibles dans l'environnement du Smartphone, de sélectionner et de se connecter à celui correspondant au HC-06, et de gérer l'affichage du « Bouton 1 » qui doit se griser (plus utilisable) une fois le Smartphone connecté avec succès.

Pensez à sauvegarder régulièrement vos modifications en créant des copies successives sur le site et à importer votre travail sur votre ordinateur en allant dans « Project » puis, pour récupérer le fichier de votre projet, cliquer sur « Importer... » au format **.aia**.

L'application est maintenant terminée ?! Il nous reste à la tester ! C'est très simple !

Il faut « compiler » le projet afin de générer un fichier **.apk**, qui sera interprétable et exécutable par votre Smartphone Android. Pour cela il suffit de cliquer sur « Build » puis « App (save .apk to my computer) ». L'explication en vidéo est sur ce lien : <https://youtu.be/l33yOkIzYac>

Une fois le fichier **.apk** (*cafetiereconnectee.apk*, par exemple) enregistré sur votre ordinateur, il reste à le copier dans votre Smartphone Android, dans le répertoire racine par exemple. Connectez pour cela votre téléphone en USB par exemple, ou toute autre démarche qui permet d'enregistrer ce fichier sur votre Android (mail, Cloud, utilitaire de messagerie...).



Puis depuis votre Smartphone, allez dans vos fichiers et enfin dans le répertoire racine. Cliquez sur le fichier : une procédure d'installation commence instantanément ! Des messages de sécurité peuvent apparaître pour confirmer votre intention d'installer cette application (pas forcément officiellement reconnue au départ par les services Google...).

Pensez évidemment à téléverser le programme **.ino** dans l'Arduino !...

Une fois que l'application est installée au milieu de celles que vous aviez déjà, vous n'avez plus qu'à cliquer sur la jolie icône « Tasse à café » par exemple (voir fichier *CafetiereBluetooth.zip*, téléchargeable sur ce [lien](#)), ou une autre que vous avez choisie dans la partie « Designer » !

Vous voilà devant l'interface, cette fois-ci interactive et fonctionnelle, que vous avez-vous-même infographiée dans cette même partie « Designer » !

Connectez-vous au Bluetooth de votre Arduino, vérifiez que votre Smartphone est également en Bluetooth !

Bon café ! ☺