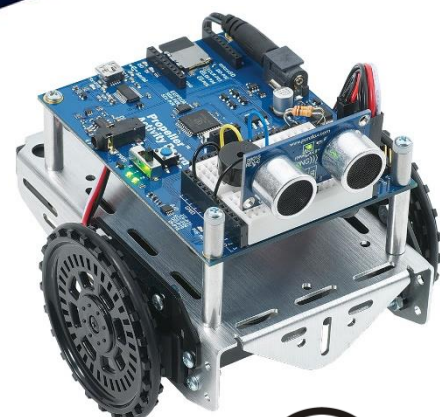




FORMATION

L'interactivité avec ARDUINO et ANDROID

Michel ESCAMEZ – Responsable Robotique



Sommaire

Introduction	1
--------------------	---

I – Découverte des différents outils

1. La carte Arduino	3
2. Les modules électroniques	5
3. App Inventor 2	7

II – Réalisation de petits projets

1. Découverte de l'Arduino	9
2. Programmation avancée	10

III – Bilan de la formation

Annexes	13
---------------	----

Faire clignoter une LED	14
Utilisation du port série	15
Contrôle de la LED avec le port série	16

Introduction

Dans le cadre de cette formation d'une durée de **4 heures**, nous allons aborder ensemble le thème de l'**interactivité**. En effet, l'**électronique** actuelle nous permet de plus en plus de réaliser des **projets collaboratifs** et en **interactivité** avec d'autres modules de communication. C'est dans cette optique que nous allons apprendre ensemble à réaliser de petits projets qui visent à mettre en évidence la communication entre un **Arduino** et une solution **Android**.

Cette formation se décompose en trois grandes parties. Dans un premier temps nous allons nous familiariser avec les différents outils qui nous accompagneront tout au long de la réalisation de nos petits projets. Ensuite nous approfondirons notre savoir-faire dans la conception d'applications visant à contrôler des modules **électroniques**. Nous finirons enfin par un temps d'échange sur ce que nous avons appris ensemble, nos interrogations mais aussi sur les perspectives d'avenir du projet.

Vous l'aurez compris, cette formation vise à vous donner tous les **outils** et les **compétences** pour mener à bien vos projets personnels et aborder le milieu de l'**électronique** avec plus de sérénité.



I - Première Partie

Découverte des différents outils

(Environ 30 min)

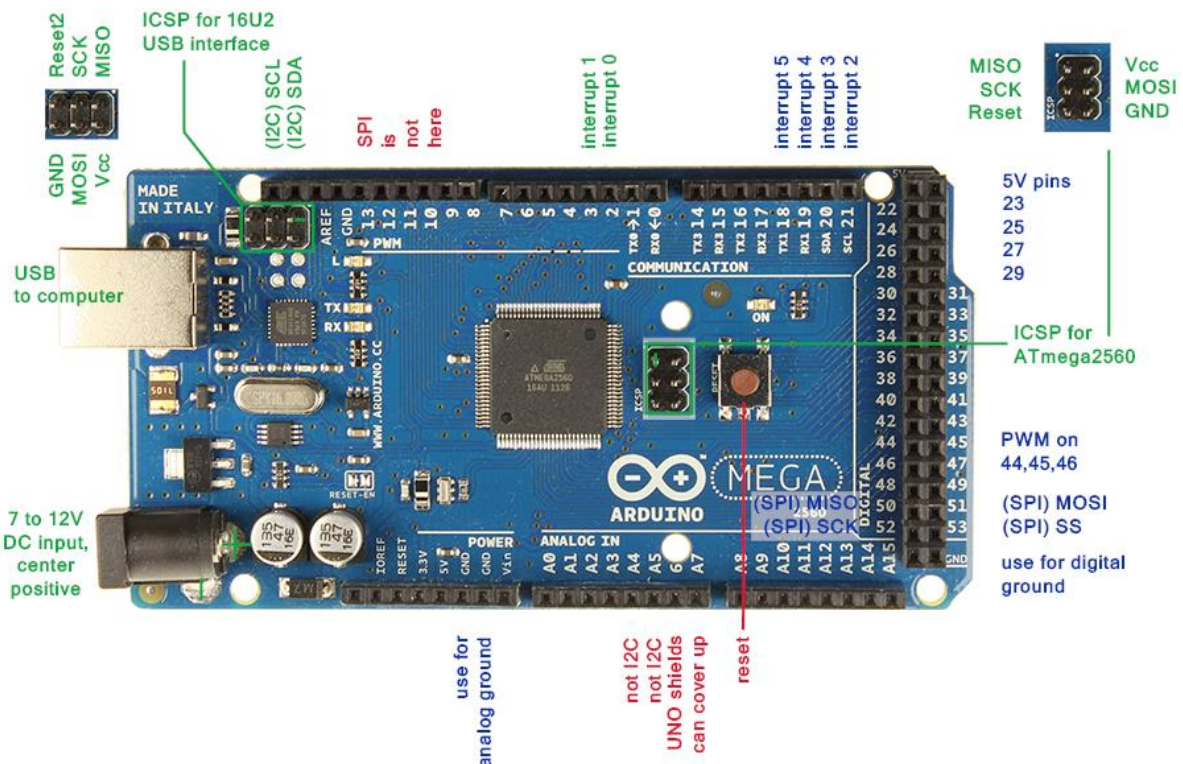


1. La carte Arduino

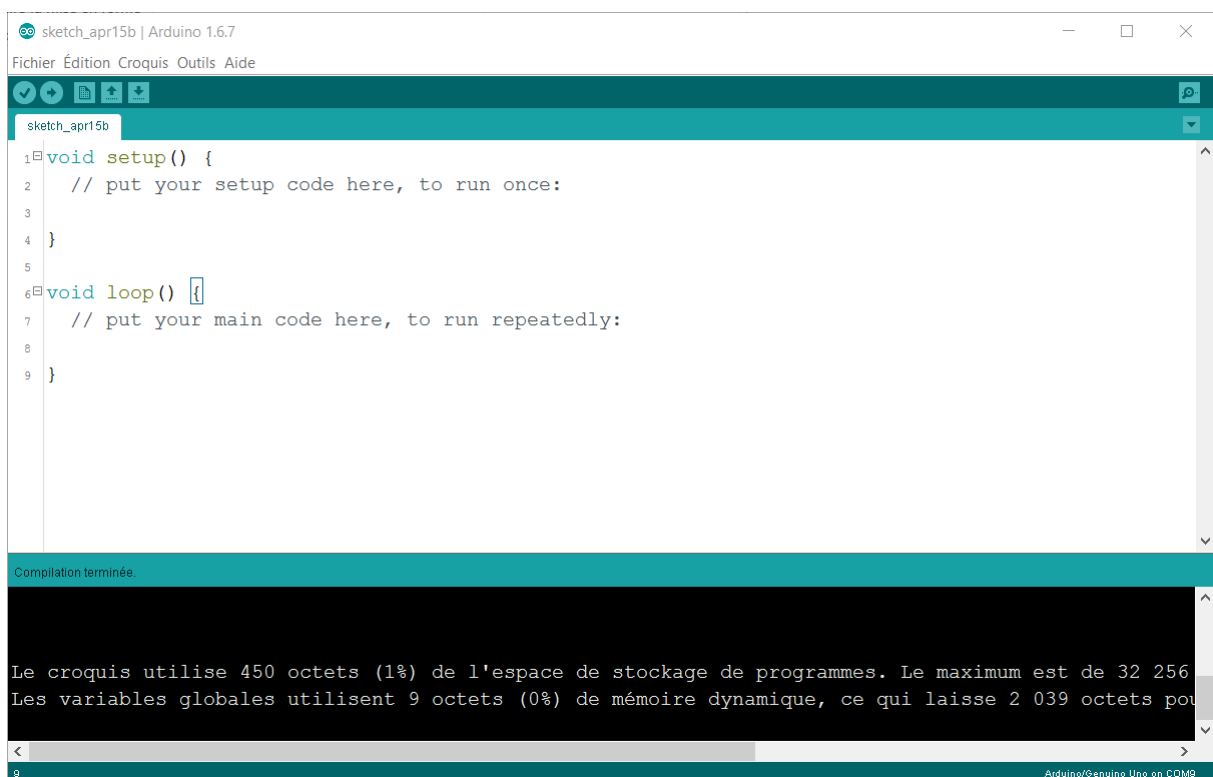
Présentation du microcontrôleur

L'**Arduino** est une carte **Open Hardware** sur laquelle se trouve un **microcontrôleur** avec une architecture de type **AVR**. Ce dernier est développé par le fabricant mondial de composants à semi-conducteur **ATMEL**. Les schémas de cette carte sont publiés en **licence libre**, cependant, certains composants, comme le **microcontrôleur** sont totalement privés.

La carte **Arduino** sert à créer des projets électroniques en simplifiant le processus de **programmation**. Il existe de nombreux modules **électroniques** qui s'adaptent à cette carte. La photo ci-dessous représente une carte de type **MEGA** :



Nous allons **programmer** la carte à l'aide d'un **IDE** créé par **Arduino**. Il existe de nombreux tutoriels pour apprendre la programmation en **C / C++**. Il est important de comprendre que cette carte fonctionne comme un petit **cerveau** auquel on vient ajouter des modules **électroniques** qui nous serviront de **muscles** afin de fournir de la puissance. Il faut voir la carte **Arduino** comme une liaison entre plusieurs parties **électroniques**. La photo ci-dessous présente l'**IDE** que nous allons utiliser afin de programmer la carte.



```
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Compilation terminée.

Le croquis utilise 450 octets (1%) de l'espace de stockage de programmes. Le maximum est de 32 256
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2 039 octets pou

9 Arduino/Genuino Uno on COM9

Il existe de nombreux types de carte **Arduino**, avec des prix allant de **15 à 80 €**. Il faut savoir que la **Chine** fabrique des copies à des prix défiant toute concurrence. Ces produits sont tout à fait satisfaisants pour la réalisation de prototypes.

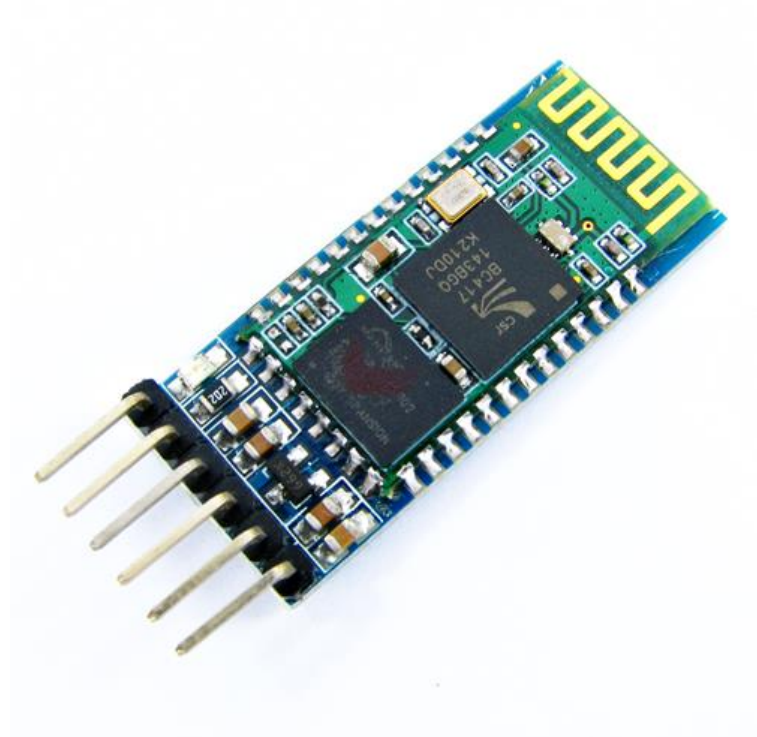


2. Les modules électroniques

Le module Bluetooth

Lors de cette formation, nous allons utiliser un module **Bluetooth** assez répandu dans le milieu **électronique** : le **HC-06**.

- Tension de **3.3** à **6V**
- Fréquence **2.4 GHz**
- Communication **UART**
- Paramétrable
- Portée de **10m**



Ce module va nous permettre de réaliser une **communication** en notre carte **Arduino** et un système supportant le protocole **Bluetooth**. La mise en place d'une **communication** est assez simple d'un point de vue matériel mais aussi en **programmation**.



Un robot Arduino 4 WD

Nous utiliserons comme support de formation un robot **Arduino** avec quatre roues motrices. Comme nous l'avons vu précédemment, la carte **Arduino** ne peut pas sortir de puissance, cependant, les roues du robot demandent un certain ampérage. C'est pour cela que nous utiliserons un **Shield Motor** qui viendra se placer dessus la carte de commande. La photo ci-dessous représente le robot que nous utiliserons pour illustrer la formation.

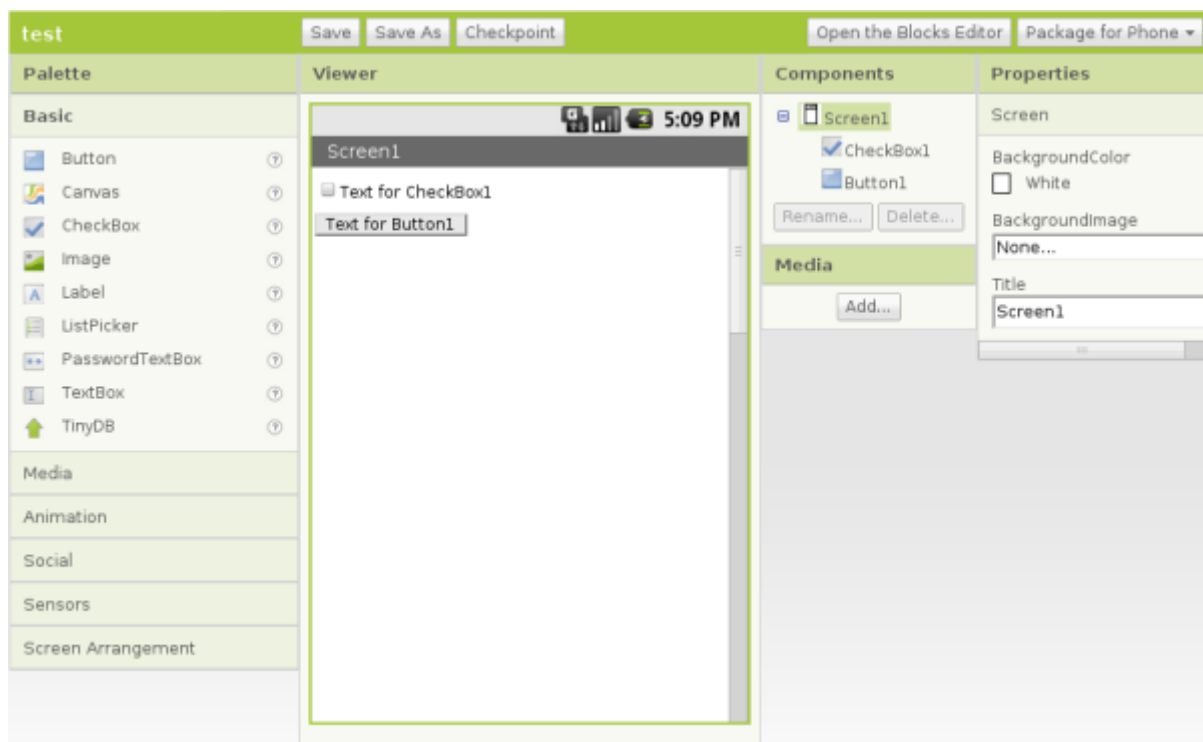


3. App Inventor 2

Outil de création d'application Android

Afin de faciliter la création d'**applications Android**, nous utiliserons une solution **Open Source** et disponible en ligne : **App Inventor 2**.

Cette solution est développée par **Google**. Le but de cette dernière est de **simplifier** la programmation d'applications sous **Android**. L'intérêt premier de **App Inventor** est de démocratiser la **programmation** à un grand nombre de personnes souhaitant **développer** des **applications** pour **smartphone** ou **tablette Android**. La photo ci-dessous représente l'application **App Inventor**.



II - Deuxième Partie

Réalisation de petits projets

(Environ 3 heures)



1. Découverte de l'Arduino

Clignotement d'une LED (5 min)

Pour notre premier programme, nous allons réaliser un simple clignotement d'une **LED** placée sur la carte **Arduino**. Il s'agit d'une **LED** de *debug*.

Le code se trouve en Annexe 1

Utilisation du port série (5 min)

Dans cet exemple, nous allons utiliser la fonction de base d'une communication : l'affichage de données sur le port série.

Le code se trouve en Annexe 2

Contrôler la LED avec le port série (15 min)

Ce programme va nous permettre de contrôler l'état de la LED placée sur la carte à l'aide de commande que nous enverrons sur le port série.

Le code se trouve en Annexe 3



2. Programmation avancée

Nous passons maintenant au niveau supérieur. Au vu de l'avancement de notre code **Arduino** ainsi que dans ce chapitre, de la programmation **Android**, les codes sources ne sont disponibles que sur un dépôt **GitHub** à l'adresse suivante :

<http://bit.ly/formation-arduino>

Se familiariser avec App Inventor (15 min)

Il s'agit de prendre en main la solution afin de programmer nos propres applications par la suite.

Création de notre première application (30 min)

Cette première application met en lien le programme **Arduino** sur le contrôle de la **LED** et un programme **Android** avec une communication **Bluetooth**.

Contrôler un robot Arduino (30 min)

Avec ce chapitre nous allons apprendre à contrôler un robot **Arduino** avec 4 roues motrices.

Contrôler un robot avec Android (1 heure 30 min)

Ce chapitre est le dernier de la partie programmation **Android**. Nous allons réaliser une application qui **contrôlera** en **Bluetooth** le robot **Arduino**. Le but est de comprendre le fonctionnement général d'une communication **Bluetooth**.



III - Troisième Partie

Bilan de la formation

(Environ 30 min)



Nous voilà maintenant à la fin de cette formation. Cette dernière s'est déroulée en **4 heures**. Il est maintenant temps de prendre quelques minutes afin de réfléchir sur ce que nous a apporté cette formation.

- **A-t-elle répondu à mes attentes ?**
- **Qu'est-ce que j'ai appris ?**
- **Je vois plus clair dans la suite du projet ?**
- **J'ai tout compris ?**
- **Je note combien de 0 à 5 ?**



Annexes

I – Faire clignoter une LED

II – Utilisation du port série

III – Contrôler la LED avec le port série



1. Faire clignoter une LED

```
#define LED 13

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}
```



2. Utilisation du port série

```
#define BAUDS 9600
```

```
void setup() {  
  Serial.begin(BAUDS);  
}
```

```
void loop() {  
  Serial.println("Mon premier programme avec le port série !");  
  delay(1000);  
}
```



3. Contrôler la LED avec le port série

```
#define LED 13

#define BAUDS 9600

void setup() {
    pinMode(LED, OUTPUT);
    Serial.begin(BAUDS);
}

void loop() {
    if (Serial.available() > 0) {
        byte etat = Serial.read();
        switch (etat) {
            case '1':
                digitalWrite(LED, HIGH);
                break;
            case '0':
                digitalWrite(LED, LOW);
                break;
        }
    }
}
```

