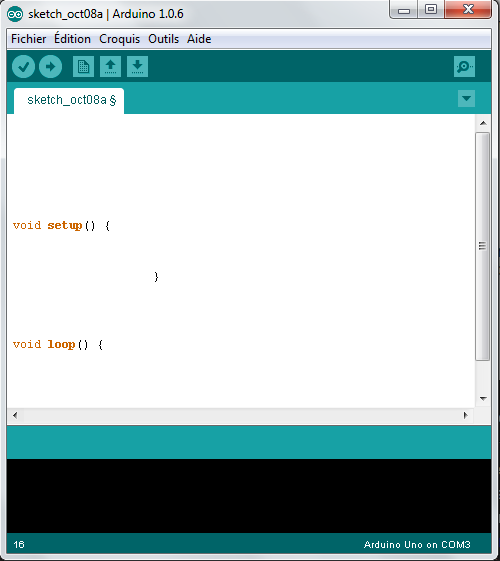
**Réalisation d’un programme recevant les ordres de l’application pour allumer et éteindre une LED**

Pour commencer, je dois me familiariser avec l’environnement dans lequel je vais réaliser le programme :

Lorsque j’ouvre le logiciel Arduino, je tombe sur la fenêtre suivante :



Fonction void loop ()

Fonction void setup ()

Déclaration

Résultat de compilation

**Explication :**

* Déclaration
* La fonction void setup () { } permet d’initialiser la carte, (s’exécute en premier)

Entre les accolades de cette fonction je déclare le contenu de l’initialisation ou de la fonction.

Mais c’est quoi ce contenu ?

Après explication de la fonction void loop (){ } et du résultat de compilation , je reviendrai pour expliquer le contenu de la fonction void setup () et d’ailleurs aussi le contenu de la fonction void loop ()

* La fonction void loop () { } se répète et s’exécute à l’infinie après le setup, plus précisément elle répète ce qu’il y a entre ses accolades (contenu) à l’infinie.
* Le résultat de compilation : Il permet de faire une vérification du contenu des fonctions setup et loop, il donne des indications sur les erreurs faites dans le programme.

**Le contenu des fonctions setup et loop :**

En effet le contenu de ses fonctions est appelé instructions.

Dans ce paragraphe j’expliquerai juste les instructions qui vont nous permettre d’allumer et d’éteindre une LED

* Instructions : pinMode(pin,INPUT) ;
* Instructions : pinMode(pin,OUTPUT) ;

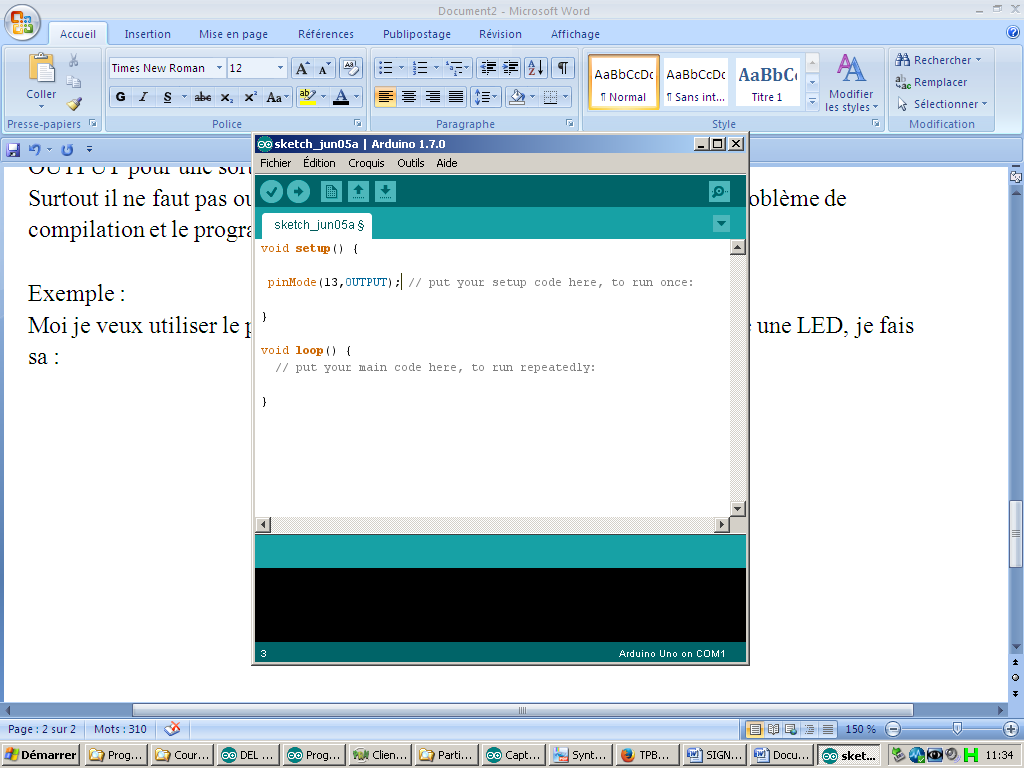
Avant d’utiliser un pin de la carte, il faut mettre dans la fonction setup () les instructions vues dessus, sachant que si je veux que le pin déclaré soit une entrée alors j’utilise la première instruction « pinMode(pin , INPUT) », par contre si je veux que mon pin soit une sortie alors j’utilise l’instruction « pinMode(pin,OUTPUT) ».

Surtout il ne faut pas oublier le « ; » (point virgule) à la fin de l’instruction sinon tu auras un problème de compilation et le programme ne fonctionnera pas.

Quoi ? Tu n’as rien compris ☹ ? Avec un exemple ça ira mieux.

**Exemple :**

Je veux utiliser le pin 13 de la carte Arduino afin d’allumer et éteindre une LED, donc j’écris entre les accolades de la fonction setup l’instruction correspondant :



*Dans ce cas je dis à la carte Arduino que je veux utiliser le pin 13 comme sortie*

*Mais pourquoi comme sortie et pas comme entrée ?*

*Tout bêtement parce que le pin 13 va envoyer de la tension (+5V) à la LED (sortie de la tension du pin 13 vers la LED)*

*Alors que si au contraire je veux alimenter le pin 13 en tension (entrée de la tension dans le pin 13), dans ce cas je la déclare comme entrée (c’est à dire à la place de OUTPUT je mets INPUT).*

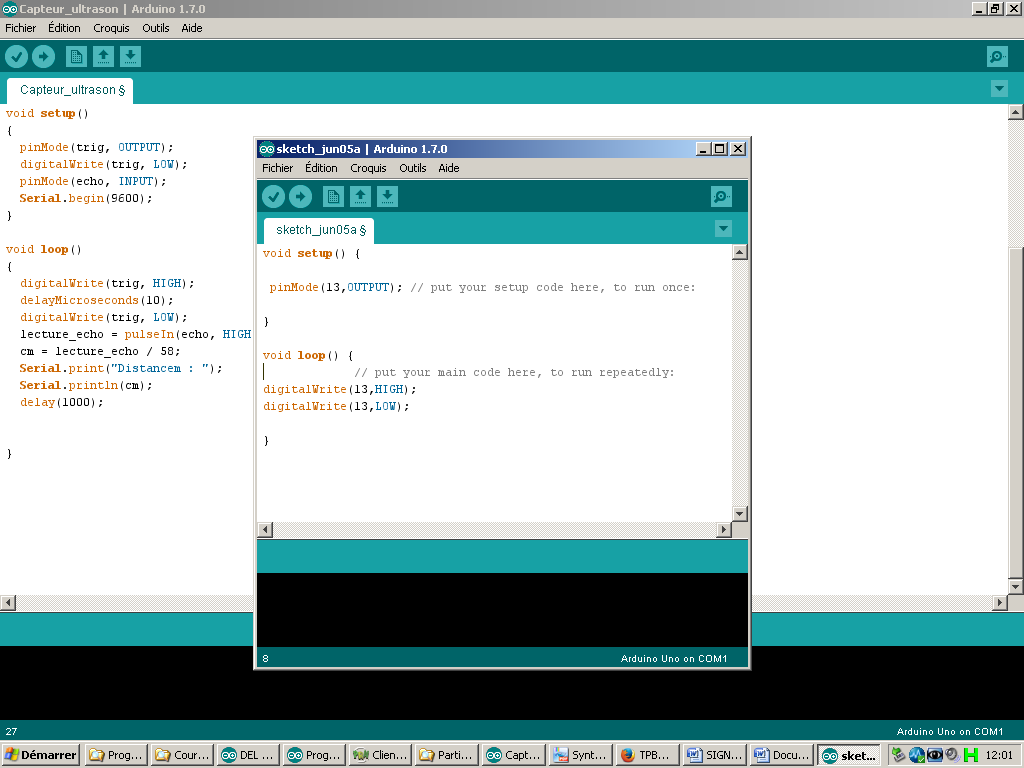
* Instructions : digitalWrite(pin,HIGH) ;
* Instructions : digitalWrite(pin,LOW) ;

La première instruction va me permettre d’envoyer de la tension du pin concerné (dans notre cas le 13) vers la LED.

La deuxième instruction va me permettre de ne pas envoyer de la tension du pin vers la LED.

Ah c’est bien donc possible d’écrire ces instructions dans la fonction void loop() pour allumer et éteindre la LED.

**Je le fais :**

****

Ma question est : cela fonctionnera t’il vraiment lors de l’envoi du programme à la carte Arduino ?

Autrement dit : ce programme me permettra t’il d’allumer et éteindre la LED avec l’application ?

Bien sûr que non, je fais un programme dans lequel je demande à la carte de faire allumer et éteindre une LED à l’infini (fonction loop qui se répète à l’infinie), et dans ce programme il n y a aucune relation avec les informations envoyées par l’application.

Il me faut donc utiliser des instructions qui vont me permettre de lire les ordres reçus de l’application ?

* Instruction : Serial.read()

Si je traduis cette instruction en français, cela veut dire « Lire série ».

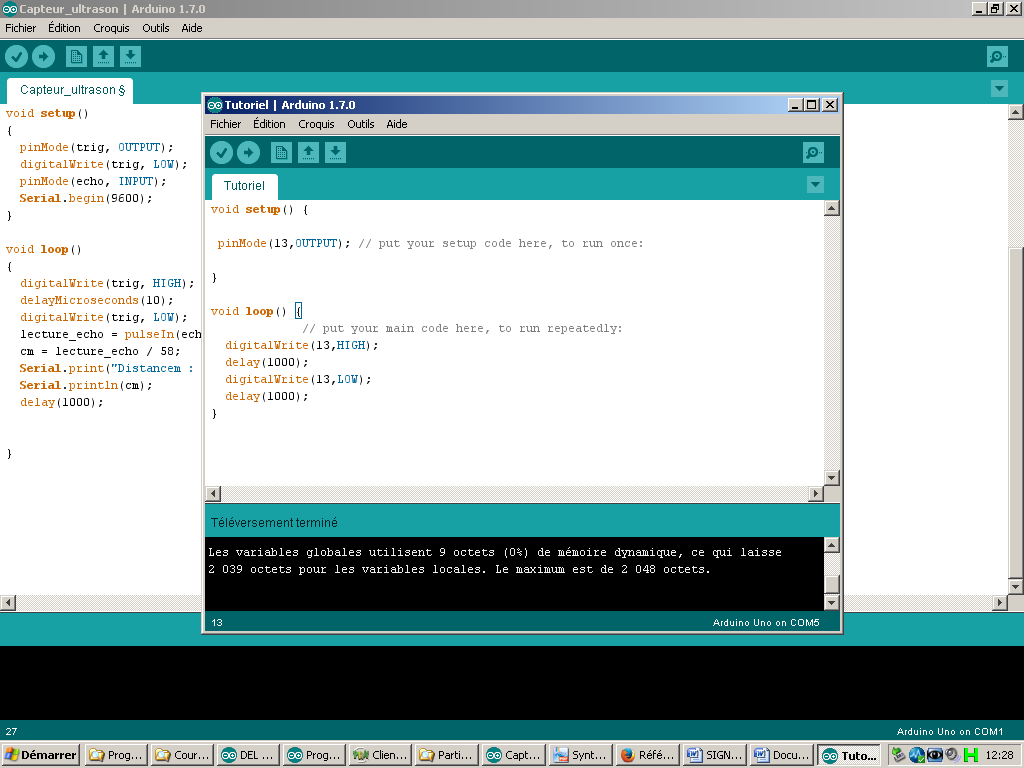
Cette instruction me permettra de reconnaitre et lire les ordres reçus de l’application via la liaison série entre la carte Arduino et le module Bluetooth.

Si vous vous rappelez lors de la réalisation de l’application, nous avons programmé les boutons de sorte que lorsque l’on appuie sur le bouton Allumer, l’application envoi la lettre « A » à la carte Arduino, la même chose pour le bouton éteindre qui envoi la lettre « E ».

Donc cette instruction me permettra de récupérer et lire ces lettres.

et de faire ceci :

* Si je reçois la lettre A j’allume la LED
* Si je reçois la lettre E j’éteins la LED



Instruction permettant d’attendre 1000ms (1s) avant de revenir au début du programme et répéter ce qu’il y a dans le void loop.

Instruction permettant d’attendre 1000ms (1s) avant d’exécuter la prochaine instruction.

Instruction permettant d’allumer la LED.

Instruction permettant d’éteindre la LED.