

## 1. Mise en place des outils / Ressources logicielles.

Nous allons travailler avec des microcontrôleurs *Atmel ATmega368P* sur plate-forme arduino.

La chaîne de développement est celle fournie par l'IDE arduino.

Elle est basée sur la chaîne libre avr-libc : <http://www.nongnu.org/avr-libc/>

Vous pouvez utiliser l'IDE arduino : <https://www.arduino.cc/en/Guide/HomePage>

ou l'IDE plus professionnel *eclipse* avec le plugin arduino :

<http://www.baeyens.it/eclipse/>

- Installer l'IDE de votre choix.
- Vérifier que vous savez compiler et télécharger un programme simple (blink).

(Si vous utilisez windows il sera certainement nécessaire d'installer le driver correspondant à votre carte).

Il n'y pas de "langage arduino", c'est simplement le langage C++ auquel se rajoutent des bibliothèque de classes et de fonctions spécifiques. La documentation de ces classes et fonctions particulières se trouve ici : <https://www.arduino.cc/reference/en/>  
Utilisez de préférence ce site et éviter les sites qui proposent des documentations traduites qui souvent incomplètes, pas à jour ou erronées.

Procurez vous un "shield" avec des leds et des boutons et relever son schéma pour connaître les broches associées aux différents éléments.

## 2. Leds , Boutons et affichage sur le port série.

Il faut savoir utiliser les fonctions `digitalWrite()`, `digitalRead()`, `delay()` et le port série (`Serial.begin()`, `Serial.print()`, `Serial.println()`).

Écrire un programme qui allume une Led lors de l'appuie sur un bouton et qui l'éteint lorsqu'on le relâche.

Ajouter au programme précédent l'allumage d'une autre led lors de l'appuie et éteindre cette nouvelle led par l'appuie sur un autre bouton.

Ajouter au programme précédent l'incrémentation d'un compteur à chaque appuie.

Afficher la valeur du compteur lorsqu'il varie. Attention un appuie ne doit provoquer qu'une incrémentation de'une unité de la valeur.

### 3. Générer des actions périodiques.

Il faut savoir déclencher plusieurs actions périodiques avec des périodes différentes dans un même programme.

- Étudier le programme exemple "blink without delay".
- Avec cette méthode faite clignoter 4 leds avec les périodes suivantes :  
1s, 15s, 4s, 9s. Écrire sur le port série l'état de chaque led à chaque changement.

### 4. Savoir utiliser la classe String.

La classe String permet de manipuler simplement les chaînes de caractères pour afficher des informations (sur le port série ou dans un fichier par exemple).

On peut également convertir des nombres en chaînes et inversement.

Voir les nombreuses possibilités ici :

<https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/>

```
#include "Arduino.h"
String msg = "";
String number="345";
String info = "heater:ON;fan:OFF" ;
int n = 12, p;
float x = 15.22458 , z;

void setup(){
  Serial.begin(115200);
  msg = "n= " + ..... ;
  Serial.print(msg);
  msg = ..... ;
  Serial.print(msg);
  p = ..... + 1 ; // augmenter number de 1
  Serial.println(p);
  z = ..... + 0.5 ; // augmenter number de 0.5
  msg = ..... ;
  Serial.println(msg);
  Serial.print("le chauffage est sur ");
  msg = ..... ; // utiliser la variable info
  Serial.print(msg);
  Serial.println(); // utiliser la variable info
}

void loop(){
}
```

Complétez le programme ci-dessus pour produire exactement l'affichage suivant :

n=12 x=15.2246

n vaut 1100 en binaire et c en hexa

346

z=345.50

le chauffage est sur ON le ventilateur est sur OFF

## 5. Passer des commandes par le port série.

Très souvent on doit "dialoguer" avec le programme par l'intermédiaire du port série. On passe une série de commande simple qui sont exécutée et qui retourne éventuellement une réponse.

Par exemple avec un shield comportant des Leds on peut imaginer le protocole très suivant :

Lorsque le programme peut accepter une commande il envoie le caractère \$.

Une commande est une suite de caractère terminée par le caractère '\n' (newline).

Certaines commandes peuvent posséder un ou plusieurs paramètres. Il est alors séparés de la commande par un espace.

\$ALLON↓	← allume toutes les Leds.
\$ALLOFF↓	← éteint toutes les Leds.
\$ON 2	← la led câblée sur la broche n°2 s'allume
\$OFF 4	← la led câblée sur la broche n°4 s'éteint
\$TEMP 1500	← le temps de clignotement est de 1,5 seconde (pour toute led)
\$CLI 4 5↓	← la led câblée sur la broche n°4 clignote 5 fois
\$MILIS	← affiche le nombre de milliseconde écoulées depuis le reset

variante :

\$LED 2 ON↓ ← la led câblée sur la broche n°2 s'allume

\$LED 5 OFF↓ ← la led câblée sur la broche n°5 s'éteint

\$CLI 4 5 1500 ← la led câblée sur la broche n°4 clignote 5 fois à 1,5 fois/sec.

Pour réaliser un tel programme, il faut bien comprendre l'exemple

"Communication → SerialEvent"

En analysant le contenu de la variable "inputString" lorsque la saisie est terminée (stringComplete vaut true) et en utilisant les méthodes de la classe String on peut implémenter ce dialogue.

## 6. Conversion analogique numérique.

L'arduino uno possède six broches utilisables en entrées analogiques (de A0 à A5).

La conversion se fait par la fonction `analogRead()` qui retourne la valeur numérisée sur 10bits selon la loi

$$N = \text{partie entière de } [V_e \cdot 5/1024]$$

- câbler un potentiomètre sur une entrée analogique de manière à pouvoir faire varier la tension entre 0 et 5V.
  - Afficher sur le port série, la valeur numérisée, la tension correspondante (en mV).
  - Faire une moyenne glissante de 10 dernières valeurs mesurées (voir l'exemple Analog → Smoothing)
- câbler une thermistance sur une entrée analogique.  
Afficher la température relevée par la thermistance.

## 7. Utilisation d'une librairie : Télécommande infrarouge

**Procurez vous une télécommande infrarouge. Il faut que la fréquence de modulation soit de 38kHz. La plupart des modèles de télécommande de téléviseur des marques courantes conviennent.**

Le récepteur de télécommande infrarouge de votre carte démodule le signal infrarouge de la télécommande et fournit le signal en bande de base.

La librairie arduino *IRRemote* ( <https://github.com/z3t0/Arduino-IRremote> ) permet le décodage de ces signaux pour la plupart des protocoles de télécommande grand public.

-Téléchargez la librairie, décompressez le dossier et renommez le IRremote. Copiez le dans le répertoire des librairies d'arduino (home/lpsarii/Arduino/librairie)

- Redémarrez le logiciel IDE arduino. Un nouveau dossier d'exemple « Irremote » doit apparaître.

- Étudiez le programme `Exemples->IRremote->IrrecvDemo` . Exécutez le après l'avoir modifié pour utiliser la broche `RECV_PIN` pour correspondre à votre câblage. <sup>1</sup>

- Relevez les codes de quelques touches de votre télécommande. Par exemple, pour les touche 1,2, + , - et POWER.

---

<sup>1</sup> :Attention si l'erreur « TKD2 » apparaît, il faut retirer le dossier nommé RobotIRRemote car il utilise les mêmes noms pour ses librairies.

Attention dans certains protocoles les codes d'une même touche changent une fois sur deux pour permettre la détection d'un nouvel appui sur une même touche (PHILIPS RC5,RC6). De plus sur les télécommandes de TV, le code d'une même touche peut changer en fonction du type d'appareil sélectionné (TV, DVD, SAT,...). Pour une description des différents protocoles voir :

<http://www.sbprojects.com/knowledge/ir/index.php>

Pour plus d'information sur les codes détectés vous pouvez utiliser le programme `Exemples->IRremote-> IRrecvDump` .

- Allumer une Led à l'appui de la touche 1 de la télécommande et l'éteindre à l'appui de la touche 2.

-Allumer et éteindre une autre Led à l'aide de la même touche (Power) .

- Allumer la Led câblée sur la broche n par les touches Vol+ suivie de la touche N et l'éteindre par les touches Vol- et N (N entre 2 et 9). Attention il faudra prévoir un "timeout" si l'utilisateur ne tape pas la séquence attendue.

## 8. Génération d'un signal PWM

La fonction `analogWrite()` permet de générer un signal PWM de rapport cyclique variable sur certaines broches (Voir documentation).

Dans un unique programme, faire varier la luminosité de trois leds câblées sur des broches sur lesquelles la génération de PWM est possible :

- en tapant une commande sur le port série `LUM=50` → 50 %,

- en faisant varier la position d'un potentiomètre,

- en utilisant les touches Vol+ et Vol- d'une télécommande infrarouge<sup>2</sup>.

Veillez à limiter la valeur de la PWM entre 0 et 255.

Vérifier le signal avec un oscilloscope.

---

<sup>2</sup> A cause d'un conflit d'utilisation des timers, ce n'est facile à faire que sur les broches 9 ou 10. Sur les autres broches il est nécessaire de modifier la configuration de la librairie `Irremote`.