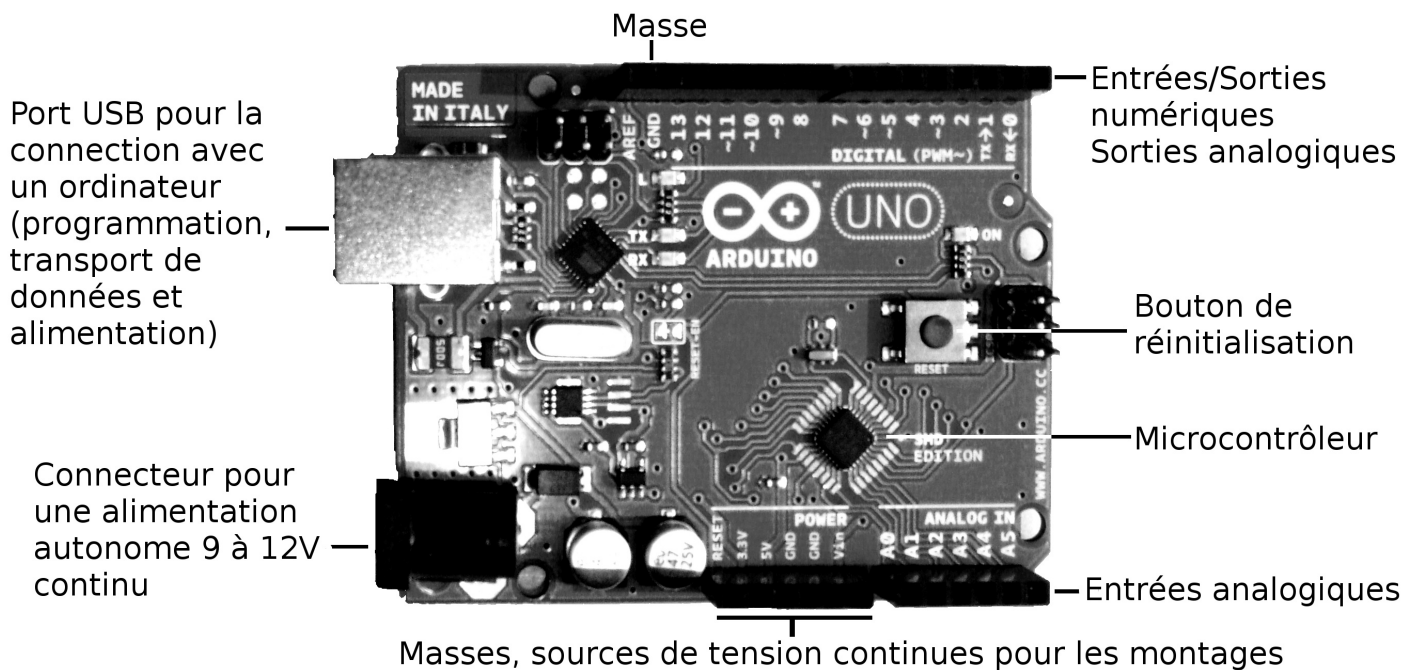


Utilisation d'un microcontrôleur

1^{ère} partie

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoire vive, mémoire morte et des interfaces d'entrées-sorties. Il est plus lent mais a une plus faible consommation électrique, une petite taille et un coût réduit par rapport aux ordinateurs. Nous pouvons ainsi disposer facilement d'un dispositif interactif autonome. Nous utiliserons pour cela une carte électronique Arduino.

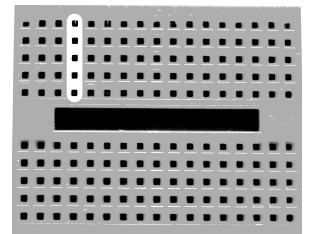
Présentation de la carte Arduino



La fréquence d'horloge de la carte est de 16 MHz, la mémoire vive de 2 kio et la mémoire morte pour stocker le programme est de 32 kio. La programmation se fait dans un langage de type C à partir du logiciel Arduino sur un ordinateur. Après débogage et compilation il est ensuite chargé dans la carte. La carte pèse moins de 24 grammes, ne consomme que quelques watt et peut délivrer jusqu'à 40 mA à chacune de ses sorties (200 mA maximum sur l'ensemble des broches).

Présentation de la plaque multiconnexion

La carte est associée à un montage qui comporte des capteurs et/ou des actionneurs. La plaque permet de réaliser le montage sans soudure et celui-ci peut être simplement modifié. Le principe : tout ensemble de cinq plots verticaux, un ici est représenté en clair, sont connectés entre eux et correspondent à un même point d'un circuit.



Notre premier montage !

Pour aller loin il faut commencer en douceur : nous allons simplement faire clignoter une diode électroluminescente (DEL).

Réalisation du montage (éléments du kit Arduino) :

- Placez une DEL sur la plaque multiconnexion avec chacune de ses bornes sur deux lignes verticales adjacentes. Patte courte à gauche.
- Placez en série avec la diode une résistance de protection de 330 Ohms (code couleurs orange-orange-marron). Nous la plaçons sur la droite, une borne de la résistance sur la série où il y a la patte longue de la diode et l'autre plusieurs colonnes plus loin.
- Placez l'extrémité d'un fil noir sur un plot de la série où est enfiché la patte la plus courte de la diode (il y a aussi un méplat de ce côté). L'autre extrémité du fil noir est mise sur un plot GND de la carte (GND pour *ground*, la masse en anglais). Pour clarifier les montages on prendra toujours des fils noirs pour relier les masses entre-elles.
- Prendre un autre fil non-noir pour relier l'autre borne de l'ensemble diode / résistance au PIN 13.
- Relier en USB la carte à l'ordinateur.

Lancement du logiciel Arduino :

- Allumer l'ordinateur sous Ubuntu
- Lancement du logiciel Arduino : Applications > Programmation > Logiciel Arduino

The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main text area contains the following code:

```
// Partie du programme qui ne se réalise qu'une fois :
void setup() {
  // Initialisation du pin 13 en tant que sortie numerique.
  // Pin 13 est connecté a une diode
  pinMode(13, OUTPUT);
}

// Partie du programme qui se répète en boucle :
void loop() {
  digitalWrite(13, HIGH); // met la sortie 13 à 5V
  delay(1000);           // attente d'une seconde
  digitalWrite(13, LOW); // met la sortie 13 à 0V
  delay(1000);           // attente d'une seconde
}
```

At the bottom of the IDE, a status bar indicates "Done compiling." and a message box shows "Binary sketch size: 1018 bytes (of a 32256 byte maximum)".

Réalisation du programme :

Nous allons utiliser un programme déjà réalisé dans les exemples du logiciel Arduino :

File > Examples > 1.Basics > Blink.

Le programme est commenté (*//*), les commentaires servent à expliquer le programme, ils sont nullement pris en compte lors de la compilation et ne prendront aucune place mémoire dans la carte. Vous pouvez en mettre autant que vous voulez.



La fonction *pinMode* permet de spécifier si les broches numériques de 2 à 13 seront utilisées en tant qu'entrées (INPUT) ou sorties.

En sortie, la fonction *digitalWrite* permet de fixer la tension d'un pin : HIGH (haut) à 5V, comme un générateur de tension, LOW (bas) à 0V, comme une masse.

La fonction *delay* réalise une pause dans l'exécution du programme. La durée est indiquée en millisecondes.

Toutes les fonctions sont expliquées sur le site suivant :

<http://arduino.cc/fr/Main/MiniReference>

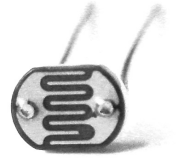
Le programme réalisé vous le compilez . Si le résultat est positif vous pouvez ensuite le charger dans la carte : . Et c'est parti !

Exercices :

1. Faire clignoter la diode dix fois plus vite.
2. Nous définissons une variable *duree*. Cette variable est un entier (*integer* en anglais). Pour cela on ajoute en début de programme la ligne *int duree = 1000;* et nous remplaçons *1000* par *duree* dans les instructions *delay*. Vérifiez que le programme fonctionne alors de la même manière.
3. Branchez une pile 9V sur le connecteur d'alimentation puis débrancher le câble USB. Vérifiez que le fonctionnement autonome est opérationnel.
4. Réalisez le programme pour le fonctionnement suivant : la diode reste allumée pendant 5 secondes puis, après une pose de 3 secondes, clignote indéfiniment.
5. Faire clignoter deux diodes dont l'une deux fois plus vite que l'autre.
6. Faire clignoter une diode 10 fois, de plus en plus vite, puis la laisser allumée. Pour cela nous utiliserons une boucle. En début de programme *int i;* puis, par exemple, *for (int i=0; i<100; i=i+1) { suite d'instructions }*. Ici la suite d'instructions est réalisée en boucle 100 fois, *i* allant de 0 à 99.

Capteur de luminosité

Nous utilisons une photo-résistance (GL5528). Il s'agit d'une résistance dont la valeur varie en fonction de la luminosité. En utilisant un ohmmètre constatez la variation de la résistance R avec l'éclairement.



Placez la résistance R en série avec une résistance R_0 de $10k\Omega$. L'ensemble est placé sous une tension E continue de 5V. Nous avons ainsi un montage diviseur de tension. Nous appelons U la tension aux bornes de la photo-résistance, exprimez U en fonction de R, R_0 et E. Avec un voltmètre notez U suivant l'éclairement.


Nous nous proposons de réaliser un montage qui allume une diode quand il fait sombre et l'éteint quand il fait jour.

Notions à utiliser :

- Nous allons cette fois utiliser une entrée analogique pour lire la tension aux bornes de la photo-résistance. La fonction *analogRead(n)* renvoie une valeur entière de 0 à 1023, correspondant à une tension de 0V à 5V sur l'entrée analogique An (tension lue par rapport à la masse). Nous avons sur la carte six entrées de ce type de A0 à A5.
- Fonction *if* : si une condition est vérifiée alors on réalise une suite d'instructions. Exemple : *if (a < 0) { b=3; }* , si a est négatif alors b vaut trois. On peut aussi réaliser une autre suite d'instructions dans le cas contraire : *if (a = 0) { b=b+2; } else { b=8; delay(100); }* , si a est nul alors j'ajoute deux à b, sinon b vaut huit et je fais une pause d'un dixième de seconde.

Ça marche ?

Dans un second temps nous affichons les valeurs de *analogRead(0)* en temps réel dans une fenêtre de l'ordi. Pour cela on utilise le moniteur série :

- Ajouter *Serial.begin(9600);* dans le setup. On obtient ainsi un débit de communication de 9600 bits par secondes de la carte vers l'ordinateur (port USB branché).
- Dans la loop : *Serial.print(*, DEC);* pour afficher la valeur de * en base 10 dans la fenêtre du moniteur sur l'écran de l'ordinateur. *Serial.println();* pour passer à la ligne. *Serial.print(",");* pour écrire un point virgule, etc.
- Important : pour éviter que la carte ne se bloque ajouter un *delay(1000);* dans la boucle.
- Après avoir chargé le programme, cliquez sur  pour ouvrir le moniteur.

Copiez les données du moniteur et collez-les dans un tableur. Tracez la courbe de la luminosité en fonction du temps.

Liste du matériel pour les parties 1 et 2 :

Par groupe :

- Kit Arduino
- Pile et connecteur pour alimentation autonome
- Multimètre MX 579
- Ordi avec le logiciel Arduino

Pour la classe :

- Imprimante déplaçable