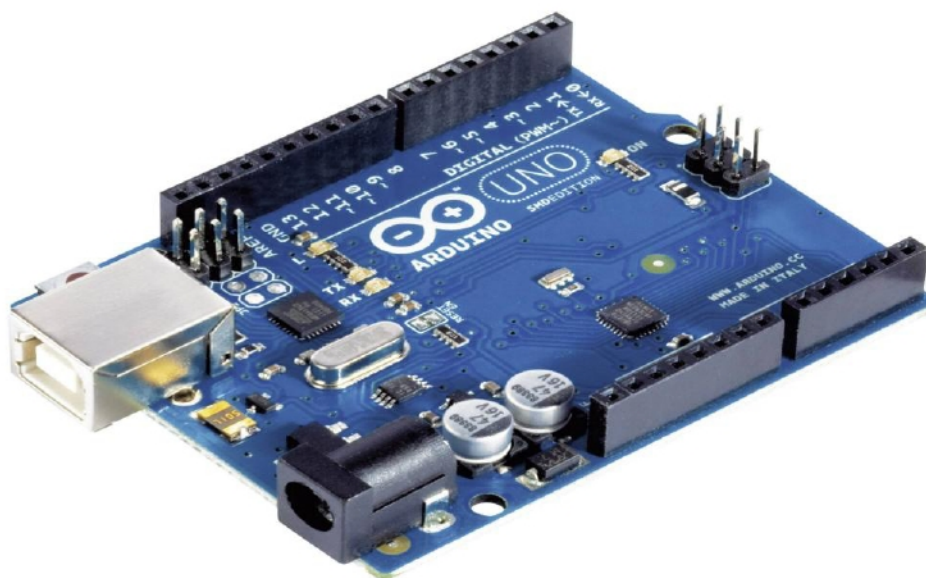


Description du matériel et environnement de travail



Groupe : LOFE

Membres : Lounas HADJALI
Ferroudja DJELLALI
Lotfi HAMICHE

Responsable de la matière :

Mr Aomar OSMANI

Chargé de TP :

Mr Massinissa HAMIDI

99 Avenue Jean Baptiste Clément, 93430 Villetaneuse

Sommaire

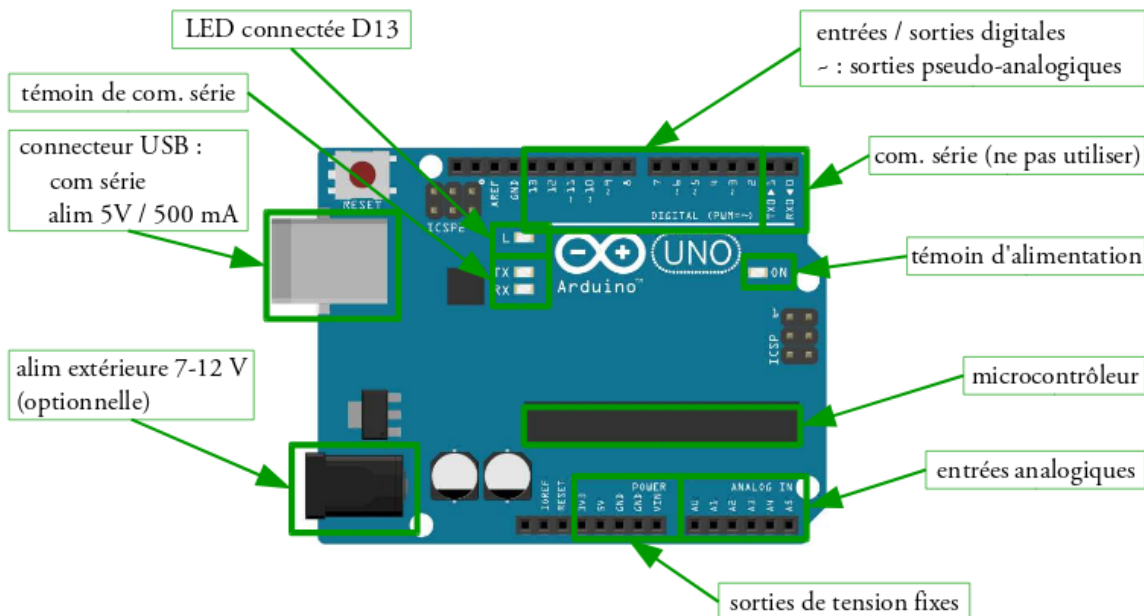
LE MICROCONTRÔLEUR ARDOUINO	3
LA PLATINE D'EXPÉRIMENTATION (BREADBOARD).....	4
LES ENTRÉES / SORTIES.....	4
LES ENTRÉES / SORTIES NUMÉRIQUES.....	5
LES ENTRÉES ANALOGIQUES.LES	5
TENSIONS DE RÉFÉRENCES.....	5
LE PORT USB	6
TRANSFÉRER UN PROGRAMME A LA CARTE	6
QUELQUES PRÉCAUTIONS.....	7
UNCOURANT TOTAL TROP FORT.....	8
INSTRUCTIONS LES PLUS IMPORTANTES EN PROGRAMMATION.....	9

LE MICROCONTRÔLEUR ARDUINO

Un microcontrôleur est un système qui ressemble à un ordinateur, il est composé de :

- Une mémoire.
- Un processeur .
- Des interfaces avec le monde extérieur.

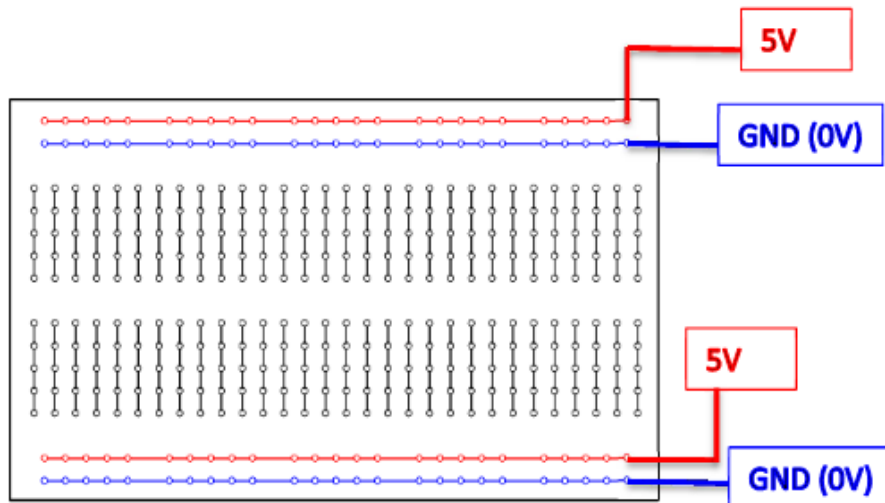
La carte Arduino se relie à un ordinateur par un câble USB. Ce dernier permet à la fois l'alimentation de la carte et la communication série avec elle.



LA PLATINE D'EXPÉRIMENTATION (BREADBOARD)

Cette platine appelée breadboard nous permet de planter des fils ou des composants sans avoir besoin de soudure, ce qui nous permet de faire des montages électriques rapidement.

Elle contient deux bandes latérales de chaque côté, qu'on réserve en général aux tensions d'alimentation (0 V et 5 V).



LES ENTRÉES / SORTIES

Afin que le arduino interagisse avec l'extérieur, on utilise les E/S, (en anglais I/O : input/output). Les sorties sont contrôlées par la carte, cela permet au programme du microcontrôleur de déclencher des actions (allumer ou éteindre une LED, un ventilateur, un moteur...).

Les entrées sont lues par le microcontrôleur, ce qui lui permet de connaître l'état du système auquel il est relié.

Il existe deux sortes d'I/O : les I/O numériques et les I/O analogues.

LES ENTRÉES / SORTIES NUMÉRIQUES.

Elles ne peuvent prendre que deux valeurs, la valeur LOW(~ GND, 0 V) et la valeur HIGH(~ 5 V).

La carte arduino comporte 14 I/O numériques(appelés DIGITAL sur la carte) numérotées de 0 à 13 et appelées D0,D1... D13. Chacun de ces ports peuvent être déclarés comme étant entrées ou sorties.

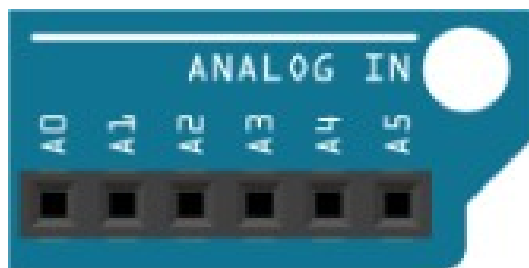
Les deux ports 0 et 1 sont réservés à la communication série.

Une LED s'allume quand le port est HIGH et s'éteint lorsque le port est LOW



LES ENTRÉES ANALOGIQUES.

Une entrée analogue est une sortie de voltmètre :la carte lit la tension qui est appliquée sur le port. Cependant le microcontrôleur ne travaille qu'avec des chiffres : il faut donc transformer la tension appliquée en sa valeur numérique. C'est le travail du convertisseur analogue/digital, dit « CAD ».



LES TENSIONS DE RÉFÉRENCES

La carte arduino fournit des ports permettant d'accéder à certaines tensions de référence.



GND est la référence de la carte arduino par rapport à laquelle toutes les différences de tension sont mesurées. Si la carte est reliée à un ordinateur par un câble USB, cette tension est celle de la terre.
Les ports 5V et 3V3 donnent accès aux tensions 5V et 3,3V.

Remarque: quand la carte est branchée sur un port USB sans alimentation externe, le port 5V ne provient plus de la carte arduino mais plutôt directement du câble USB.

VIN est la tension de l'alimentation externe, quand il y en a une.

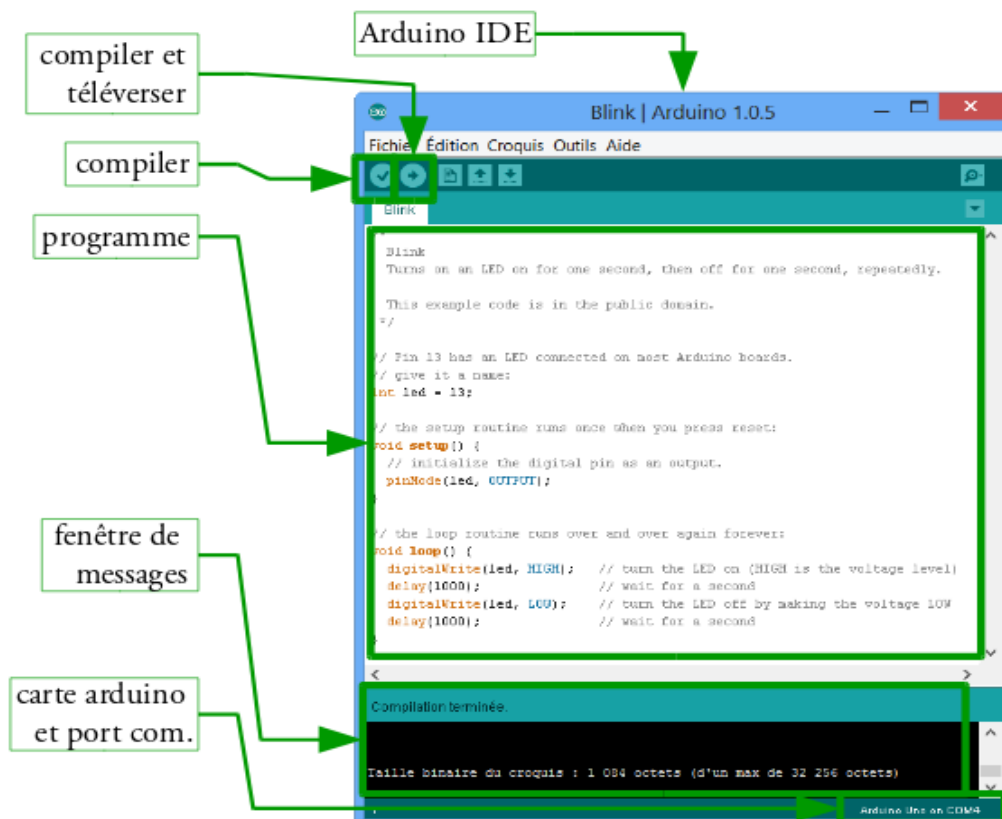
LE PORT USB

Le port USB permet à la fois l'alimentation de la carte Arduino et la communication série entre la carte et l'ordinateur. Une fois connectée, la carte Arduino apparaît dans le gestionnaire de matériel de notre ordinateur, connecté à un port série (COM1, COM4, ...).

TRANSFÉRER UN PROGRAMME A LA CARTE

La façon dont le microcontrôleur gère ses entrées / sorties est fixée par un programme, contenu dans le microcontrôleur. Ce programme doit être écrit par l'utilisateur. En pratique, l'utilisateur écrit le programme en langage C, en utilisant un environnement de développement spécialisé (IDE) installé sur un ordinateur. Ce programme est ensuite compilé et téléversé dans le microcontrôleur par liaison série (USB).

Dans le cadre de ce cours, nous utiliseront l'ide standard Arduino

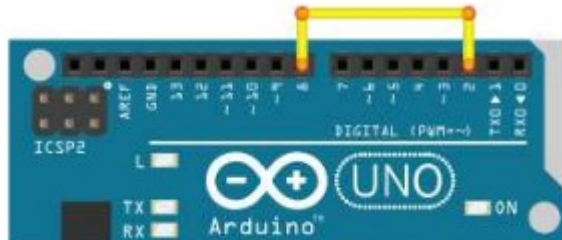


QUELQUES PRÉCAUTIONS

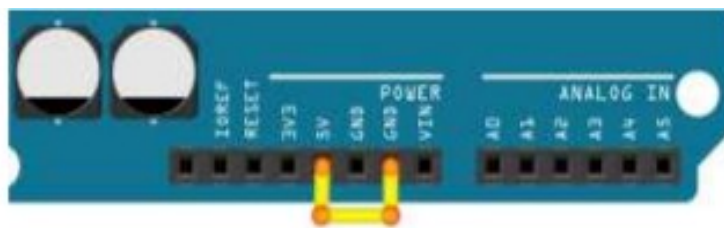
Afin de préserver notre matériel , on ne doit éviter :

Un court-circuit :

- Si on relie deux ports numériques l'un à l'autre, tous deux définis en sortie, l'un étant basculé en HIGH, l'autre en LOW. Un court-circuit s'établit entre les deux ports, et la carte brûle.

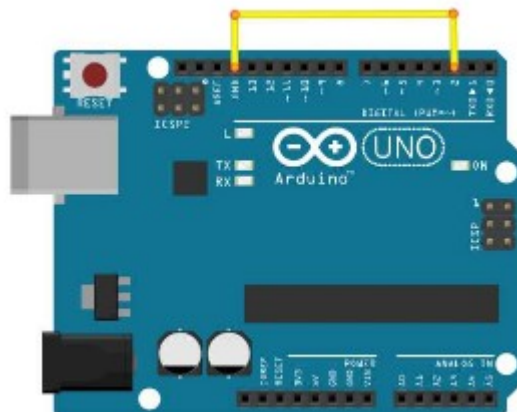


- Relier le port 5V(ou le port 3.3V) directement au port GND, ou relier le port 5V au port 3.3V produit un court-circuit.



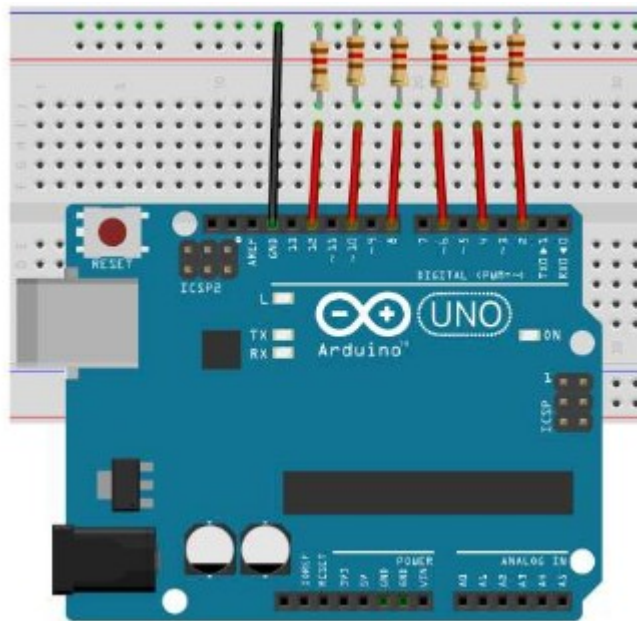
Un courant trop fort

Une résistance inférieure à 100 Ω entre le port numérique et la masse (port GND) demandera un courant supérieur à 50 mA si le port bascule en HIGH. Un fil, qui a une résistance quasi-nulle, fera un court-circuit et imposera un courant énorme.



UN COURANT TOTAL TROP FORT

Une façon un peu plus compliquée de détruire les sorties numériques est de respecter la limite des 40 mA par ports, mais de dépasser la limite des 200 mA que peut débiter la carte Arduino. Si une résistance de 125 Ω relie un port numérique à la masse, 40 mA circulent quand le port est basculé en HIGH, ce qui n'abîme pas le port (même si il vaut mieux ne pas trop s'approcher des 40 mA en pratique). Mais si six ports numériques basculent en même temps, le courant total délivré par la carte dépassera les 200 mA, et le microcontrôleur sera endommagé.



- Toujours mettre une résistance en série à une diode branchée à une sortie digitale pour ne pas cramer la diode.
- Ne jamais relier directement une sortie digitale à la masse.
- Ne jamais relier directement deux sorties digitales .

INSTRUCTIONS LES PLUS IMPORTANTES EN PROGRAMMATION

digitalWrite(3, HIGH);

Impose la valeur HIGH au port 3 défini comme OUTPUT (autrement dit, envoie 5 V à la sortie du port D3). On peut mettre LOW et dans ce cas, c'est 0 V.

analogWrite(6,100) ;

Impose la valeur 100 au port D6 défini comme OUTPUT (il s'agit d'une sortie analogique PWM).

digitalRead(4);

Lit la valeur HIGH ou LOW du port D4 défini comme INPUT. **analogRead(5);**

Lit la valeur analogique au port 5 défini comme INPUT.

delay(1);

Attendre en millisecondes.

Serial.print(" toto ");

Envoi d'une chaîne sans retour à la ligne

Serial.println(" toto ");

Envoi d'une chaîne avec retour à la ligne.

Nous utiliserons le site : <https://www.arduino.cc/reference/en/> pour plus de fonctions que nous utiliserons dans notre projet.