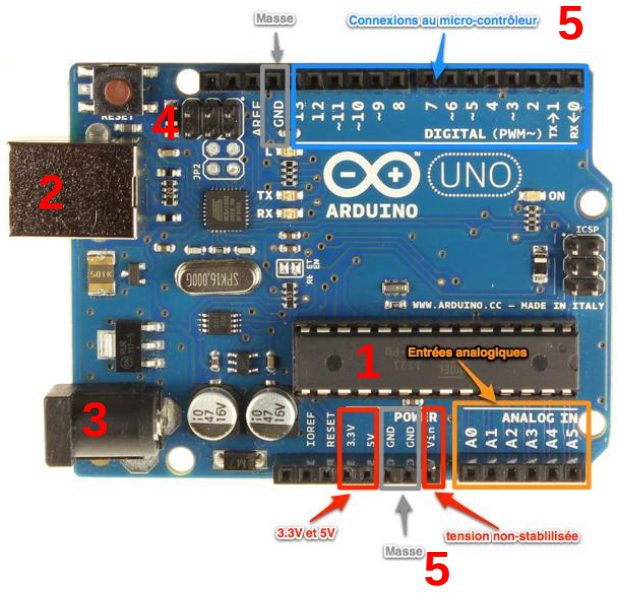
Spécialité SI

Présentation de la carte Arduino UNO et de son environnement de programmation

Présentation de la carte Arduino UNO :



1. Le cœur de la carte, un µcontrôleur Atmel ATMEGA328P

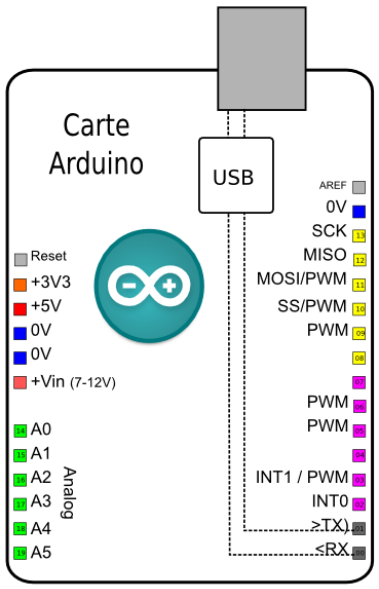
2. Le port USB, qui permet entre autre d'alimenter la carte en +5V, mais surtout de télécharger le programme de l'ordinateur vers le microcontrôleur et aussi de visualiser les échanges sur la liaison série.

3. Prise d'alimentation externe. L'alimentation doit être de type continue et avec une valeur comprise entre +7V et +15V. Il est parfois nécessaire d'utiliser cette alimentation lorsque l'on connecte une charge "importante" ou si l'on souhaite l'autonomie de la carte arduino.

4. DELS de visualisation. L'une (orange) permet de tester la carte, les 2 autres (vertes) permettent de visualiser la liaison série en émission Tx et en réception Rx.

5. Connecteurs permettant une liaison filaire avec des composants extérieurs (capteurs, boutons poussoir, Dels, transistor...)

Le processeur ATMEGA



Carte Arduino UNO

Atmega168 Pin Mapping

Arduino function				Arduino function
reset	(PCINT14/RESET) PC6	1	28 PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27 PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26 PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25 PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24 PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23 PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22 GND	GND
GND	GND	8	21 AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20 AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19 PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18 PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17 PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16 PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15 PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Correspondance entre les broches de l'Arduino et les ports de l'ATmega168.

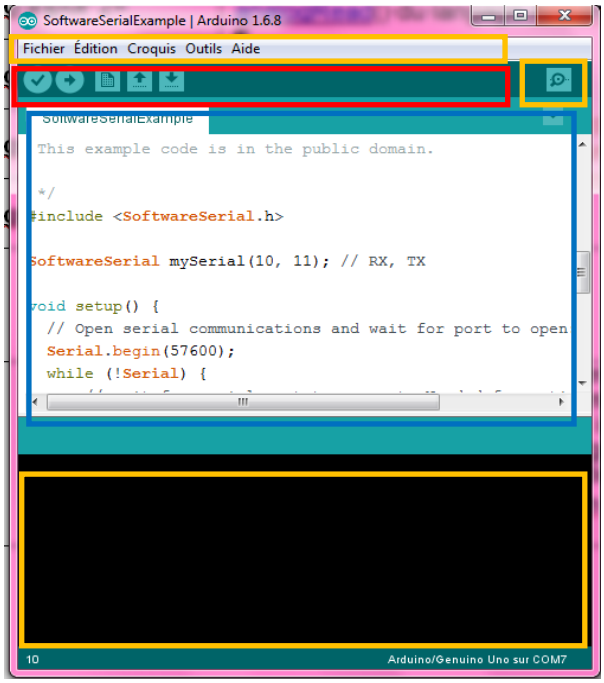
Entrées et sorties de la carte Arduino UNO

Digital Pins		
Broche 0	• **Communication série RX**	**Communication Serie**: Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL.
Broche 1	• **Communication série TX**	
Broche 2	• **Interruption externes**	Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt() pour plus de détails.
Broche 3	• **Interruption externes** • **Impulsion PWM (largeur d'impulsion modulée)**	
Broche 4	• **Impulsion PWM (largeur d'impulsion modulée)** • **I2C (SDA)**	Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().
Broche 5	• **Impulsion PWM (largeur d'impulsion modulée)** **I2C (SCL)**	
Broche 6	**Impulsion PWM (largeur d'impulsion modulée)**	
Broche 7	• **Entrée/sortie digitale**	
Broche 8	• **Entrée/sortie digitale**	
Broche 9	**Impulsion PWM (largeur d'impulsion modulée)**:	
Broche 10	• **Impulsion PWM (largeur d'impulsion modulée)** **SPI (Interface Série Périphérique) (SS)**	
Broche 11	**SPI (Interface Série Périphérique) (MOSI)**	
Broche 12	**SPI (Interface Série Périphérique) (MISO)**	
Broche 13	• **SPI (Interface Série Périphérique) (SCK)** **Broche LED**	

Analog Pins		
Broche A0 (14)	• **Analog Input 0**	Broches pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino Note : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19.
Broche A1 (15)	• **Analog Input 1**	
Broche A2 (16)	• **Analog Input 2**	
Broche A3 (17)	• **Analog Input 3**	
Broche A4 (18)	• **Analog Input 4**	
Broche A5 (19)	• **Analog Input 5**	

Présentation de la carte Arduino UNO et de son environnement de programmation

Le logiciel de programmation



1 L'ensemble de ces onglets permettent principalement de choisir ou de définir les options de configuration du logiciel.

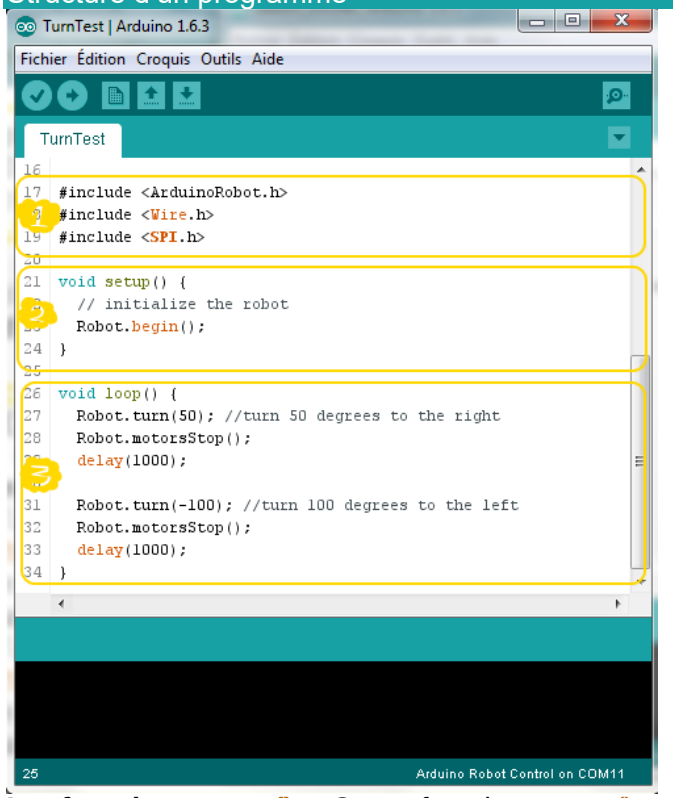
2 Les icônes permettent de contrôler les fichiers programmes, de télécharger les programmes.

3 Le moniteur série

4 La partie ou l'on écrit le programme.

5 L'écran du débugger (indication d'erreurs de compilation)

Structure d'un programme



1- La partie déclaration des bibliothèques et des variables
2- La partie initialisation et configuration des Entrées / Sorties void setup()
3- Le programme principal qui s'exécute en boucle void loop()

Coloration syntaxique

En orange, apparaissent les mots-clés reconnus par Arduino comme des fonctions existantes ;
En bleu, ce sont les constantes.
En gris, apparaissent les commentaires. Ils ne sont pas exécutés par le programme. Ils sont obligatoires pour se retrouver dans le code, notamment dans le cadre de projets collaboratifs.
« // » tout ce qui se trouve après ce symbole sera interprété comme commentaire, dans une ligne donnée
Entre « /* » et « */ », on peut encadrer plusieurs ligne de commentaires.

La fonction setup() : Cette fonction setup() est appelée une seule fois lorsque le programme commence. On y écrit le code qui n'a besoin d'être exécuté qu'une seule fois. On appelle cette fonction : "fonction d'initialisation". On y retrouvera la configuration des principales fonctions que l'on souhaite utiliser, la configuration des broches en entrée ou en sortie...
La fonction loop() : C'est dans cette fonction loop() que l'on écrit le contenu du programme. Cette fonction s'exécute en boucle infinie, lorsque la dernière ligne de programme est exécutée, le programme reprend de la première ligne.

Syntaxe

Le code est structuré par une ponctuation stricte

- Toute ligne de code se termine par un point virgule
- Le contenu d'une fonction est délimité par des accolades
- Les paramètres d'une fonction sont contenus par des parenthèses

Les variables

Une variable est un espace réservé dans la mémoire de l'ordinateur. C'est comme un compartiment dont la taille n'est adéquate que pour un seul type d'information. Elle est caractérisée par un nom qui permet d'y accéder facilement.
Il existe différents types de variables identifiés par un mot-clé dont les principaux sont :

- nombres entiers (int)
- nombres à virgule flottante (float)
- texte (String)
- valeurs vrai/faux (boolean).

Un nombre à décimales, par exemple *3.14159*, peut se stocker dans une variable de type float. Notez que l'on utilise un point et non une virgule pour les nombres à décimales. Dans Arduino, il est nécessaire de déclarer les variables pour leurs réserver un espace mémoire adéquat. On déclare une variable en spécifiant son type, son nom puis en lui assignant une valeur initiale (optionnel).
Exemple :

```
int ma_variable = 45;  // int est le type, ma_variable le nom et = 45 assigne une valeur.
```

Les fonctions

Une fonction (également désignée sous le nom de procédure ou de sous-routine) est un bloc d'instructions que l'on peut appeler à tout endroit du programme.
Le langage Arduino est constitué d'un certain nombre de fonctions, par exemple analogRead(), digitalWrite() ou delay().
Il est possible de déclarer ses propres fonctions par exemple :

```
void clignote(){
digitalWrite (brocheLED, HIG H) ;
delay (1000) ;
digitalWrite (brocheLED, LO W) ;
delay (1000) ;
}
```

Pour exécuter cette fonction, il suffit de taper la commande :

```
clignote();
```

On peut faire intervenir un ou des **paramètres** dans une fonction :

```
void clignote(int broche,int vitesse){
digitalWrite (broche, HIG H) ;
delay (1000/vitesse) ;
digitalWrite (broche, LO W) ;
delay (1000/vitesse) ;
}
```

Dans ce cas, l'on peut moduler leurs valeurs depuis la commande qui l'appelle :

```
clignote(5,1000); //la sortie 5 clignotera vite
clignote(3,250); //la sortie 3 clignotera lentement
```