

# Robotique

*Contenu des ateliers*

*Éléments de programmation*



Association de Cadres en recherche d'emploi  
et Ateliers Informatiques



# Présentation des ateliers Robotiques

## Objectifs

Les différentes composantes d'un robot  
Les éléments de détection ..... : les capteurs  
Les éléments de réflexion ..... : la programmation  
Les éléments de réaction ..... : moteurs et servomoteur  
Les ateliers conduisent à la construction d'un mobile mais ne la réalise pas.

## Le module Arduino UNO R3

Module électronique expressément préparé pour faire des expérimentations.  
Son fonctionnement global.  
Ses broches.

## L'EDI

Échange de **Données Informatisées** autorisant une programmation de l'Arduino  
Programmation du module  
Présentation des principales fonctions de l'EDI  
Les 3 parties d'un programme

## Les bases de la programmation

Initiation aux éléments fondamentaux :

- Script
- Variables
- Syntaxe
- Conditions
- Fonctions

## Les expériences de base

### Expérience 1 :

Allumer une diode à l'aide d'un bouton poussoir.  
Emploi d'un condensateur pour réduire les a coups du courant d'extra de rupture  
Sens d'une diode

### Expérience 2 :

Faire varier le régime d'un moteur avec un potentiomètre .  
Distinguer le circuit de commande faible intensité.  
du circuit de puissance forte intensité.  
Emploi d'un transistor interface entre les deux circuits.

### Expérience 3 :

Faire varier le régime d'un moteur avec les broches PWM.  
Définition de PWM.  
Explication de la formule.  
application avec modification des paramètres.

### Expérience 4 :

La direction avec le servomoteur.  
Définition d'un servomoteur.  
Rechercher la fonction Arduino correspondante.  
Recherche incontournable pour tous les composants utilisés.  
Application.

### Expérience 5 :

Le capteur de distance.  
Recherche de la fonction Arduino.  
Explication du fonctionnement.  
Mise en œuvre de la fonction.

### Expérience 6 :

Allumer successivement 3 diodes en fonction de l'éloignement d'un objet.  
Fin des ateliers.

# ^ Commandes de structure du programme

## Structure générale

- `void setup()` (configuration-préparation)
- `void loop()` (exécution)

## Contrôle et conditions

- `if` (si...)
- `if...else` (si...alors...)
- `for` (pour...)
- `switch case` (dans le cas où...)
- `while` (pendant que ...)

## Opérations de comparaison

- `==` (équivalent à)
- `!=` (différent de)
- `<` (inférieur à)
- `>` (supérieur à)
- `<=` (inférieur ou égal à)
- `>=` (supérieur ou égal à)

## Opérations booléennes

- `&&` (et)
- `||` (ou)
- `!` (et pas)

## Autres commandes

- `//` (commentaire simple ligne)
- `/* */` (commentaire multi-lignes)
- `#define` ( donner une valeur à un nom)

## Variables

### Type

- `char` (variable 'caractère')
- `int` (variable 'nombre entier')
- `long` (variable 'nombre entier de très grande taille')
- `string` (variable 'chaîne de caractères')
- `array` (tableau de variables)

### Niveaux logiques des connecteurs numériques

- `HIGH` (état 1)
- `LOW` (état 0)
- `INPUT` (configuré en entrée)
- `OUTPUT` (configuré en sortie)

## Fonctions

### Entrées-sorties numériques

- `pinMode(broche, état)` (configuration des broches)
- `digitalWrite(broche, état)` (écrire un état sur une broche num.)
- `digitalRead(broche)` (lire un état sur une broche num.)

- `unsigned long pulseIn(broche, état)` (lire une impulsion sur une broche num.)

## Entrées analogiques

- `int analogRead(broche)` (lire la valeur d'une broche ana.)
- `analogWrite(broche, valeur)` (PWM : écrire une valeur analogique sur les broches 9, 10 ou 11)

## Gestion du temps

- `unsigned long millis()` (temps de fonctionnement du programme)
- `delay(ms)` (attente, en millisecondes)
- `delayMicroseconds(us)` (attente, en microsecondes)

## Nombres aléatoires

- `randomSeed(seed)` (aléatoire 'piloté')
- `long random(max)` (aléatoire à partir de telle valeur)
- `long random(min, max)` (aléatoire entre deux valeurs)

## Communications série entre Arduino et autres machines ou ordinateur

- `Serial.begin(speed)` (configuration de la vitesse de communication Série)
- `Serial.available()` (donne combien de caractères disponibles dans la zone tampon Série)
- `Serial.read()` (lit les données Série)
- `Serial.print(data)` (envoie des données Série)
- `Serial.println(data)` (envoie des données Série suivies de caractères spécifiques)

## Librairies additionnelles

### Matrix (gestion d'une matrice de LEDs par contrôleur MAX7219)

- `matrix(data, load, clock)` (Pour configurer des matrices de LEDs)
- `matrix.write(x, y, value)` (Pour envoyer des données aux matrices de LEDs)
- `matrix.write(x, y, sprite)` (Pour envoyer des données aux matrices de LEDs)
- `matrix.clear()` (efface l'écran de LEDs).
- `matrix.setBrightness(value)` (règle la brillance de l'écran)

### Sprite (gestion de formes sur matrice de LEDs)

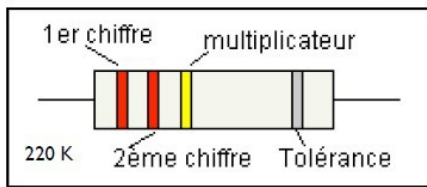
- `Sprite(largeur, hauteur, colonne1, colonne2...)` (Starts the LCD library.)
- `sprite.width()` (Returns the width in pixels of the sprite.)
- `sprite.height()` (Returns the height in pixels of the sprite.)
- `sprite.write(x, y, valeur)` (Writes data to an x, y position of the sprite.)
- `sprite.read(x, y)` (Returns the data stored on the x, y position of the sprite.)

**SimpleMessageSystem** ( librairie pour communiquer avec tous les logiciels à messages sous forme de caractères ASCII, comme PD ou Max), documentée ici <http://www.arduino.cc/playground/Code/SimpleMessageSystem>

**Wire**, une librairie pour connecter l'Arduino dans un réseau de capteurs en communication à deux fils, comme dans la carte Wiring

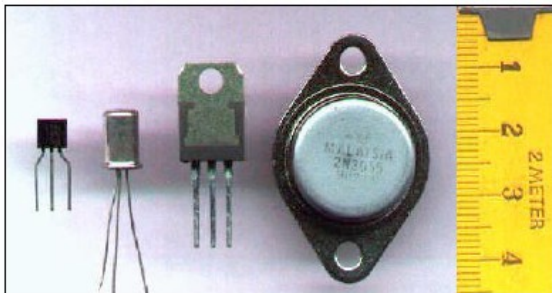
**PDuino**, un programme interne pour contrôler des fonctions avancées avec Pure Data <http://at.or.at/hans/pd/objects.html>

Code des couleurs des résistances



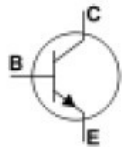
Chiffre	0	1	2	3	4	5	6	7	8	9	Or	Argent
Multiplicateur	1	10	$10^2$ 100	$10^3$ 1000	$10^4$ 10000	$10^5$	$10^6$					
Précision	20%										5%	10%

21



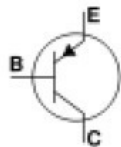
Le transistor

Le transistor est généralement utilisé comme une sorte de multiplicateur de puissance: lorsqu'on lui fait passer un courant faible, mais variable dans un de ses 3 pattes, il autorise proportionnellement le passage d'un gros courant dans une autre des 3 pattes.



transistor NPN

Symboles



transistor PNP

L'ensemble de ces données ont été récupérées sur Internet