

Arduino : fonction importante

+ pinMode (pin ; mode) : identifier l'état de la pin.
Le mode
Input Output

+ digitalWrite (pin, valeur) : change l'état de pin → 5V ou 0V
HIGH LOW

+ digitalRead (Pin) → Lire l'état de la pin → 1 ou 0

+ analogRead (pin) retourne 0 <= x < 1023 → 0 <= x < 5V
0 1023

+ analogWrite (pin, valeur) → 0 <= x < 5V ou (0°, 5V)

+ delay (valeur ms) → Mettre une pause
ms

* Serial monitor :

c'est une utile qui nous a facilité la communication avec Arduino en utilisant une interface de communication pour afficher les résultats tests les variables.

Serial.begin (debit) → débit de communication défaut 9600
Fixed
Serial.available ()

Serial.read () → lire débit Serial

* Composant Arduino : Serial.print(), Serial.println()

+ Microcontrôleur : le cœur de la carte

+ Bouton de remise à zéro

+ les pins d'entrées/sorties (0/13)

+ pins de sortie analogiques (3, 5, 6, 9, 10, 11) → 0 <= x < 255

+ Tx LED transmission + Reception

+ L : LED Lite avec Pin 13

+ A0 → AS pin d'entrée analogique

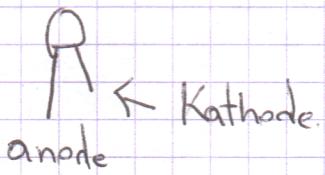
+ Broche SV, GND

// convert from type to type

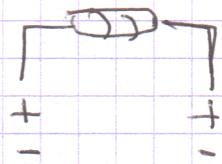
map (long x, long in min, long in max, long out min, long out max)

+ Composant

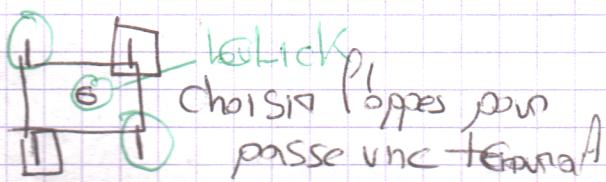
LFB



Resistor



button



RGB LEB



Red \ominus Blue Green

potentiometer



GND Input 5V

TMP

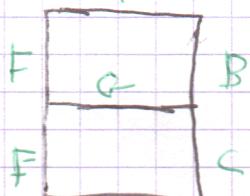


5V Input Ground

$$d = (e27.10) \times 10$$
$$c = (e27.100)$$
$$v = e27.10$$

7 Segment

A \leftarrow



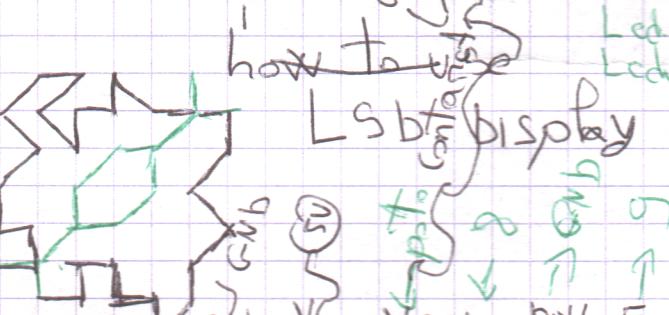
pin connection

EDC₄DP A
Ground

16x02

how to use

LSD₇ display



// the function of Lcd

#include <LiquidCrystal.h>

LiquidCrystal lcd(R₃, E, D₄, D₅, D₆, D₇)

| setup() :

 16 2

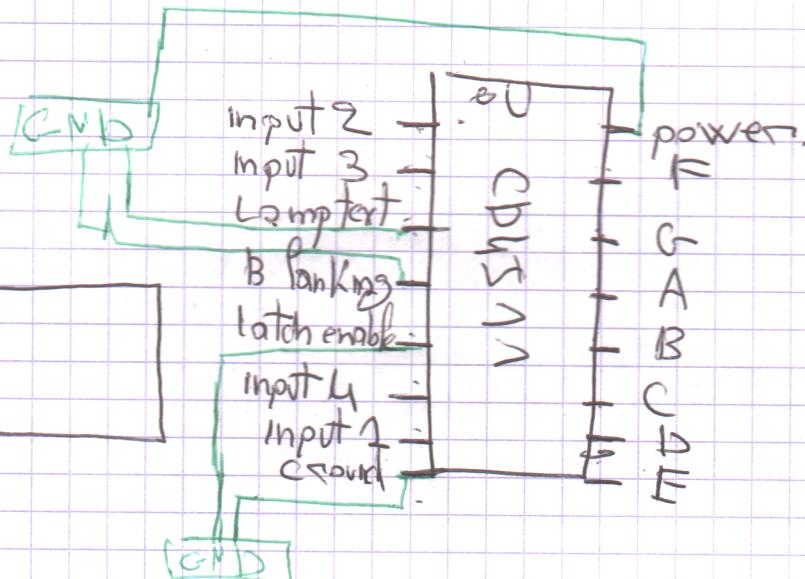
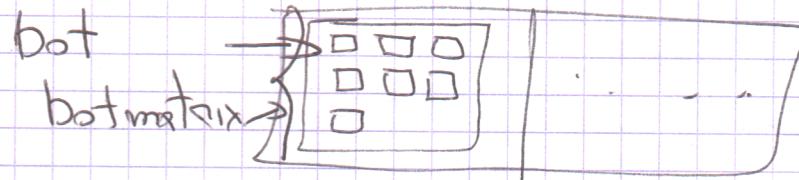
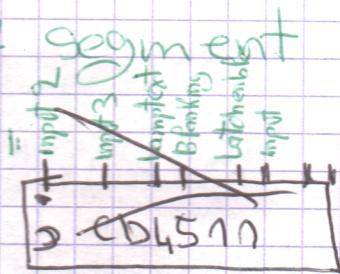
lcd.begin(16, 2) define the size.

lcd.print("Hello") print in the display

lcd.setCursor(positionColumn, positionRow)

lcd.clear(); clear the screen

// 7 segment



Input 1 bit plus faible

trouver la valeur de Cent x /100;

$$d = (e1/10) \% 10$$

e1/10

/10 deve le dernier nombre.

%10 retourne le dernier nombre.

Anatomie d'ARDUINO

(1) Microcontrôleur Atmega: Il représente le cœur de la carte Arduino.

(2) Bouton de remise à zéro: Il permet de remettre à zéro

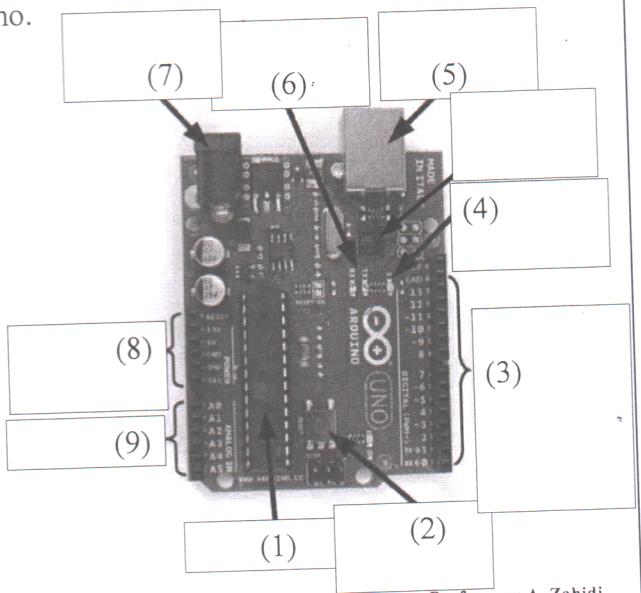
le microcontrôleur.

(3) Les pins d'entrées/sorties numériques (0-13): Ces pins peuvent être utilisés pour recevoir ou envoyer des signaux numériques "0" ou "1".

(3) Les pins de sorties analogiques (3, 5, 6, 9, 10 et 11): Ce sont des pins numériques qui peuvent être programmés pour délivrer des signaux analogiques.

(4) LED: Elle est soudée sur la carte Arduino et est reliée avec la Pin 13.

(5) Port USB: Il sert à alimenter la carte Arduino, transférer le programme et établir la communication entre Arduino et le moniteur série.



Professeur: A. Zahidi

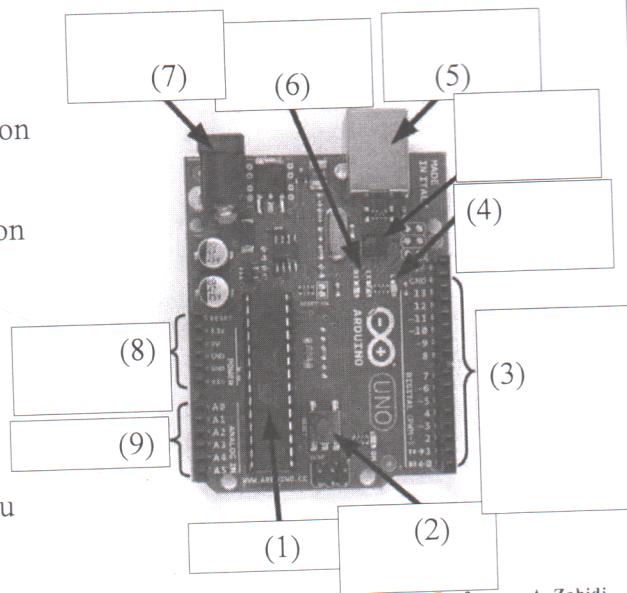
Anatomie d'ARDUINO

(6) LED de transmission et réception: Elles indiquent la communication entre la carte Arduino et l'ordinateur.

(7) Connecteur d'alimentation: Il permet de connecter une alimentation à la carte Arduino quand le port USB n'est pas utilisé. Ce connecteur accepte une tension entre 5V-12V.

(8) Broche 5V et GND: Elles fournissent 5V et la masse au circuit réalisé.

(9) Les pins d'entrées analogiques: Elles sont utilisées pour mesurer une tension variable (entre 0 et 5V) qui peut provenir des capteurs ou d'interfaces diverses (potentiomètre, etc..)



Professeur: A. Zahidi

1. Cahier des charges : Les étapes de réalisation d'un cahier des charges comprennent l'analyse des besoins, la spécification fonctionnelle, la spécification technique, la conception et la rédaction du cahier des charges.

2. Différence entre microprocesseur et microcontrôleur :

- Un microprocesseur est principalement responsable du traitement des instructions, tandis qu'un microcontrôleur intègre un processeur ainsi que des ressources telles que la mémoire et les interfaces d'entrée/sortie.

- Les microprocesseurs sont généralement plus complexes et puissants, tandis que les microcontrôleurs sont conçus pour des applications spécifiques avec des exigences réduites en termes de taille, de consommation d'énergie et de coût.

- Les microprocesseurs offrent une plus grande flexibilité de programmation, tandis que les microcontrôleurs sont souvent préprogrammés pour des tâches spécifiques.

3. Carte ARDUINO : C'est une plateforme matérielle open-source basée sur un microcontrôleur, conçue pour faciliter le prototypage et le développement de projets électroniques interactifs.

4. Différence entre ARDUINO Mega et Uno :

- ARDUINO Mega a plus de broches d'entrée/sortie, plus de mémoire programme et plus de mémoire SRAM que l'ARDUINO Uno.

- ARDUINO Uno est une carte plus basique, utilisée pour des projets plus simples avec moins de ressources.

5. Langage ARDUINO : C'est un langage de programmation simplifié basé sur C/C++, spécialement conçu pour la programmation des cartes ARDUINO, permettant le contrôle des entrées/sorties, la gestion des capteurs et des actionneurs, ainsi que d'autres fonctionnalités spécifiques aux cartes ARDUINO.

6. Fonctions ARDUINO :

- `'pinMode(pin, mode)'`: Configure le mode d'une broche (INPUT ou OUTPUT).
- `'digitalRead(pin)'`: Lit l'état d'une broche numérique (HIGH ou LOW).
- `'analogWrite(pin, valeur)'`: Génère un signal PWM sur une broche spécifique.
- `'delayMicroseconds(valeur)'`: Met en pause le programme pendant une durée spécifiée en microsecondes.

