

# AP : Révision Arduino

Nom :

Prénom :

Classe

## Je prends connaissance du programme suivant

### 1. J'identifie sur le programme les différentes parties.

```
// du site www.mon-club-elec.fr
// Auteur du Programme : X. HINAULT - Tous droits réservés
// Programme écrit le : 16/1/2011.
// ----- Licence du code de ce programme -----
// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License,
// or any later version.
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
// You should have received a copy of the GNU General Public License
// along with this program. If not, see <http://www.gnu.org/licenses/>.

// ////////////////////////////////// PRESENTATION DU PROGRAMME //////////////////////////////////
// ----- Que fait ce programme ? -----
/* L'appui sur un bouton poussoir inverse l'état d'une LED */
// ----- Circuit à réaliser -----
// Broche 2 : Un BP connecté au 0V
// Broche 3 : Une LED et sa résistance en série connectée au 0V
// --- Déclaration des constantes utiles ---
const int APPUI=LOW; // constante pour tester état BP
// --- Déclaration des constantes des broches E/S numériques ---
const int BP=2; // Constante pour la broche 2
const int LED=3; // Constante pour la broche 3
// --- Déclaration des variables globales ---
int etatLED=0; // variable d'état de la LED
```

## AP : Révision Arduino

```
// ////////////////////////////////// 2. FONCTION SETUP = Code d'initialisation //////////////////////////////////
void setup() { // debut de la fonction setup()
    // ----- Broches en sorties numériques -----
    pinMode (LED,OUTPUT); // Broche LED configurée en sortie
    // ----- Broches en entrées numériques -----
    pinMode (BP,INPUT); // Broche BP configurée en entrée
    // ----- Activation si besoin du rappel au + des broches en entrées numériques
    digitalWrite (BP,HIGH); // Rappel au + activé sur la broche BP configurée en entrée
} // fin de la fonction setup()

//////////////////////////////// 3. FONCTION LOOP = Boucle sans fin = coeur du programme //////////////////////////////////
void loop(){ // debut de la fonction loop()
    if (digitalRead(BP)==APPUI) { // si appui sur le BP
        if (etatLED==0) etatLED=1; else etatLED=0; // inverse la variable etatLED
        delay(250); // pause anti-rebond
    }
    // met la LED dans l'état de la variable etatLED
    if (etatLED==1) { // si la variable vaut 1
        digitalWrite(LED,HIGH); // la LED est allumée
    }
    else { // sinon, càd si la variable vaut 0
        digitalWrite(LED,LOW); // la LED est éteinte
    }
}
```

### Je réponds aux questions

1. J'écris l'objet (le but) du programme :

L'appui sur le bouton poussoir inverse l'état de la LED

2. J'écris ce que fait le programme :

Le programme initialise les variables des broches et les affecte en entrée et sortie.

Si le bouton est appuyé, si etatLED est à 0, etatLED passe à 1 sinon il passe à 0.




Le programme attend 250 ms.

Si etatLED est à 1, le programme allume la led, sinon le programme éteint la led.\_

3. Je complète le tableau avec les trois (3) principales parties du programme et j'écris leur utilité.

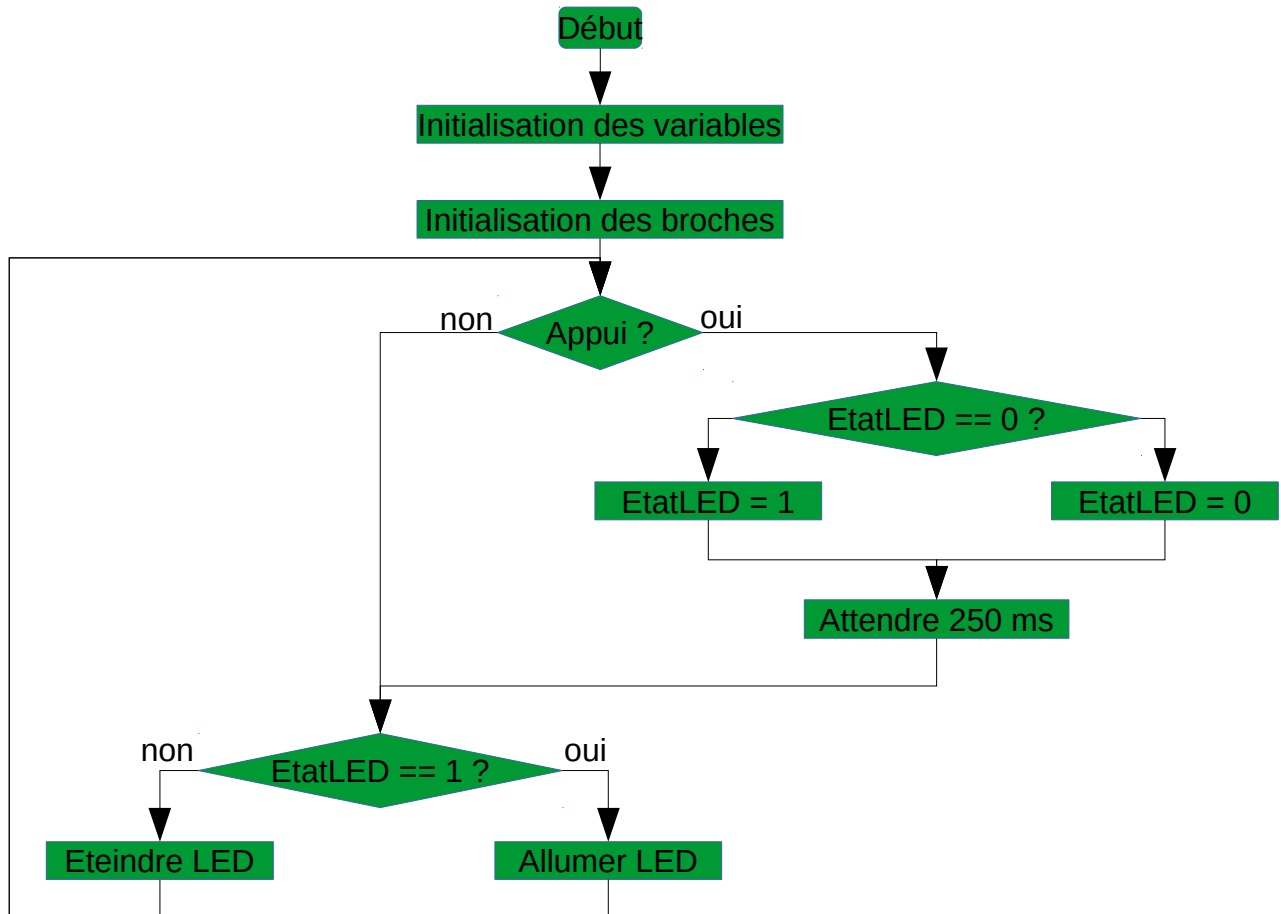
Les parties	A quoi elle sert
Initialisation des variables	A affecter des valeurs/états aux variables du programme.
La fonction Setup()	Sert à initialiser l'état des broches en entrée ou en sortie.
La fonction Loop	C'est le programme principal, il est répété indéfiniment.

4. Je rappelle les différents symboles d'un algorithme.

Je dessine la forme.	J'écris ce quelle représente.
	Début (et fin de l'algorithme).
	Traitement (en règle général un verbe d'action).
	Condition, en général se termine par un ?

## AP : Révision Arduino

5. Je réalise l'algorithme du programme.



6. Je rappelle le code couleur des résistances

Couleur	Noir	Marron	Rouge	Orange	Jaune	Vert	Bleu	Violet	Gris	Blanc
Valeurs	0 ( $\times 10^0 \Omega$ )	1 ( $\times 10^1 \Omega$ )	2 ( $\times 10^2 \Omega$ )	3 ( $\times 10^3 \Omega$ )	4 ( $\times 10^4 \Omega$ )	5 ( $\times 10^5 \Omega$ )	6 ( $\times 10^6 \Omega$ )	7 ( $\times 10^7 \Omega$ )	8 ( $\times 10^8 \Omega$ )	9 ( $\times 10^9 \Omega$ )

7. Je complète le tableau des opérateurs.

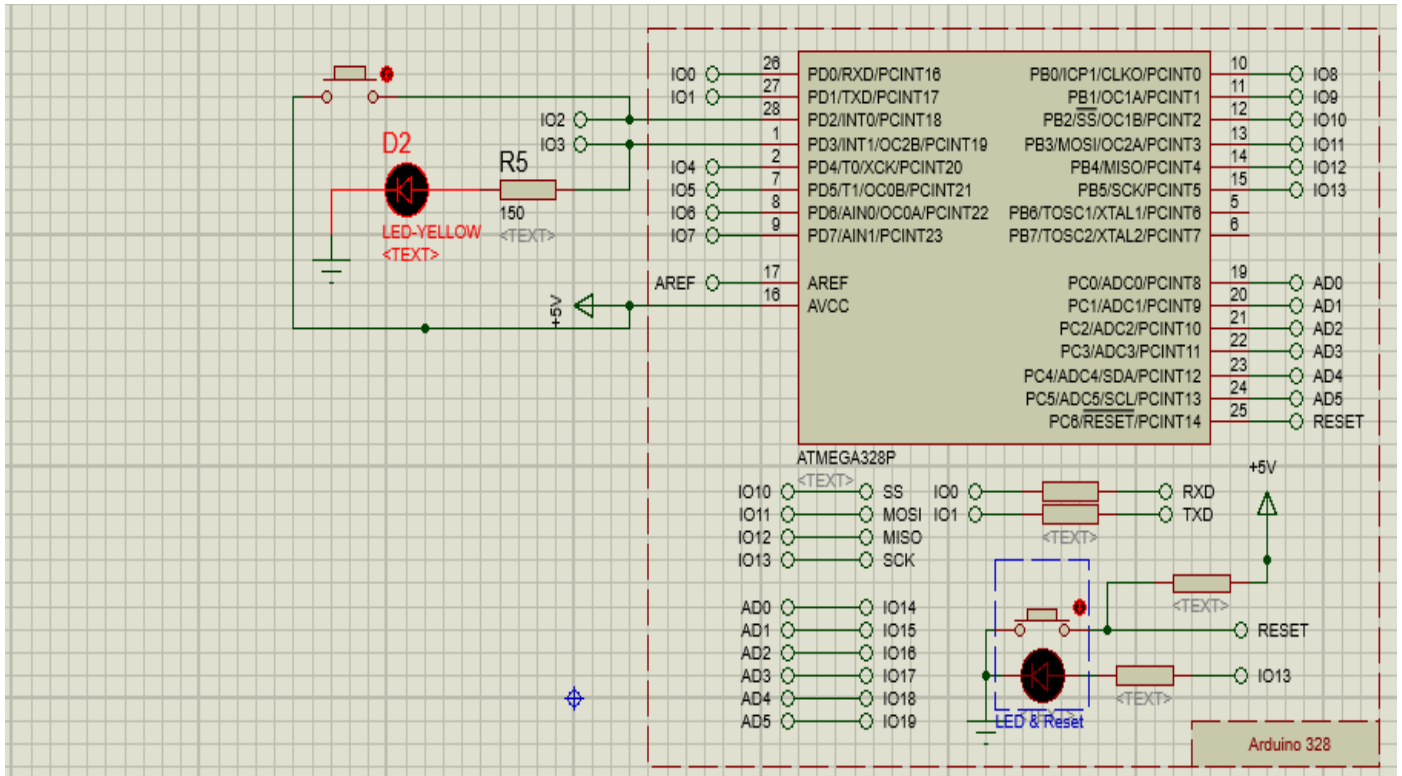
Opérateur	Explication	Exemple
+=	Additionne dans la variable de gauche la somme des deux termes.	Si a=1 et b=2, a+=b donne a=3
!=	Est vrai si le terme de gauche est différent du terme de droite.	1 != 2 est vrai 1 != 1 est faux

## AP : Révision Arduino

*	Multiplication	$2 * 3 = 6$
!	Inverse la valeur d'un booléen.	Si b est Vrai alors $b != b$ rend b à Faux

# AP : Révision Arduino

Je représente mon circuit



## Dimensionnement résistance

- Je rappelle l'expression de la résistance en fonction de la tension et du courant.

$U = RI$  donc  $R = U/I$

- En tenant compte des tensions seuil des LED suivantes, je calcule la valeur de la résistance nécessaire à leur fonctionnement. La carte Arduino fournit un courant de 10 mA.

Couleurs	Tension seuil (V <sub>f</sub> )	Courant I <sub>f</sub> mA	Résistance Ω
Rouge	1,6 V à 2 V	6 à 20	$R = (5 - 2) / 10 \cdot 10^{-3} = 3/10^{-2} = 3 \cdot 10^2 = 300 \text{ } \Omega$
Jaune	1,8 V à 2 V		
Vert			
Bleu	2,7 V à 3,2 V		$R = (5 - 3,2) / 10 \cdot 10^{-3} = 1,8 / 10^{-2} = 1,8 \cdot 10^2 = 180 \text{ } \Omega$

