

Développée au début des années 70, la technologie LCD (Liquid Crystal Display – écran à cristaux liquides) n'est réellement utilisée à grande échelle qu'à partir de la fin des années 90 pour les écrans des ordinateurs et au début des années 2000 pour les téléviseurs.

Problématique :

Comment de tels écran produisent-ils des images colorées ?



1. **Observer** à la loupe l'écran de l'ordinateur affichant une page blanche (dans Libre Office) puis **décrire** vos observations.

.....

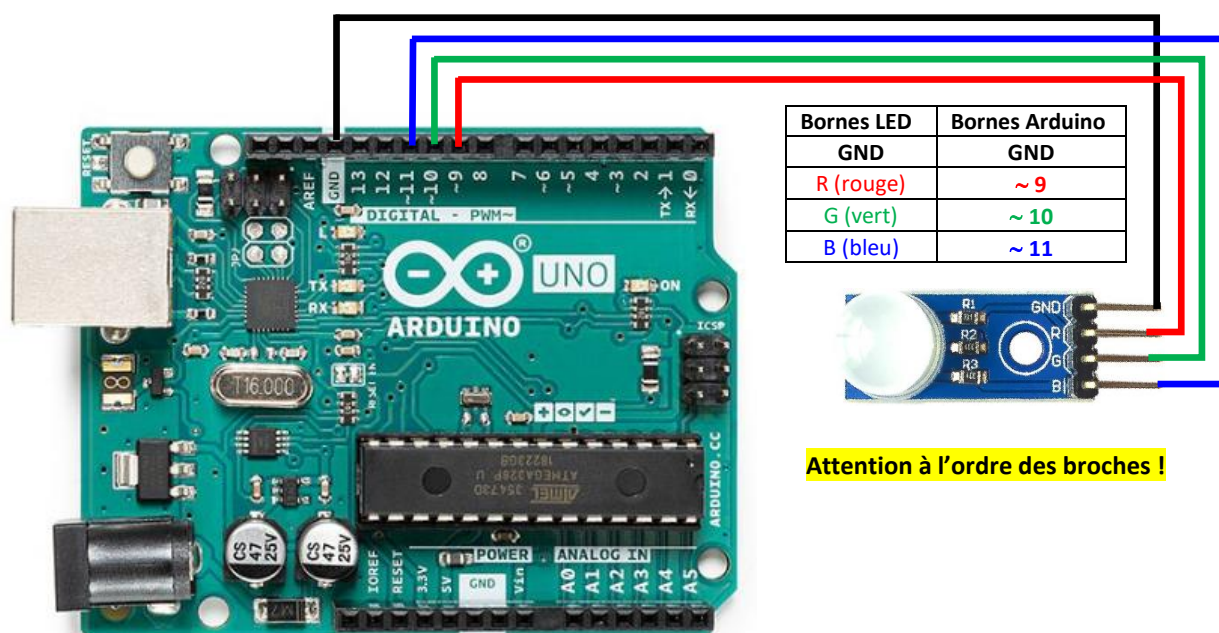
.....

.....

La lumière blanche est obtenue par la superposition des couleurs, c'est le **système RVB**.


Un écran LCD comporte une mosaïque de petits éléments lumineux appelés **pixels** (picture elements), eux-mêmes composés de trois sous-pixels correspondant aux trois couleurs primaires de la synthèse additive des couleurs.

2. **Réaliser** le montage ci-dessous qui permet de simuler le comportement de chaque sous-pixel grâce à une led RVB.



Lire le paragraphe 1 du document annexe.

3. **Ouvrir** le fichier **LED_RVB.ino** puis **procéder** aux réglages du logiciel (paragraphe 2 du document annexe).

4. **Téléverser** le programme dans la carte () puis **indiquer** la couleur de la led.

.....

La partie du programme qui sera à modifier est celle de la boucle principale :

```
void loop() {  
  affichage (255, 0, 0);  
}
```

5. **Proposer** une hypothèse sur la signification des valeurs 255, 0 et 0.

.....

.....

6. **Vérifier** l'hypothèse en modifiant les valeurs afin d'obtenir la couleur vert puis la couleur bleu. **Noter** les codages.

Couleur vert : affichage (..... , ,) Couleur bleu : affichage (..... , ,)

La fonction affichage possède 3 arguments notés r, v et b dans cet ordre qui peuvent varier de 0 à 255.

7. **Comparer** les instructions « affichage(0, 0, 5) », « affichage(0, 0, 100) » et « affichage(0, 0, 255) » puis **indiquer** l'influence, pour une couleur primaire donnée, de la variation de l'argument qui lui correspond.

.....

8. **Justifier** par un calcul que la led est capable de produire plus de 16 millions de couleurs.

.....

9. **Ouvrir** le fichier **images_couleur_RVB** disponible dans votre dossier partagé puis **indiquer** pour chaque couleur (cyan, magenta et jaune) les sous-pixels qui ont permis de les obtenir.



Cyan	Magenta	Jaune
.....

10. **Reproduire** ces couleurs sur la led en plaçant judicieusement les valeurs 255 nécessaires. **Préciser** le codage utilisé.

Cyan : affichage (..... , ,) Magenta : affichage (..... , ,) Jaune : affichage (..... , ,)

11. **Répondre** à la problématique en complétant l'encadré.

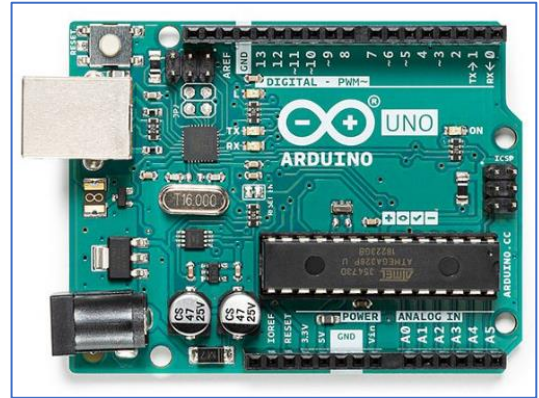
Les couleurs produites par les écrans LCD sont obtenues grâce à la **synthèse** **des couleurs du système**
Les **couleurs primaires** de cette synthèse sont et les couleurs
..... en sont les **couleurs secondaires**. Le blanc correspond à

Document annexe

1. La carte Arduino Uno

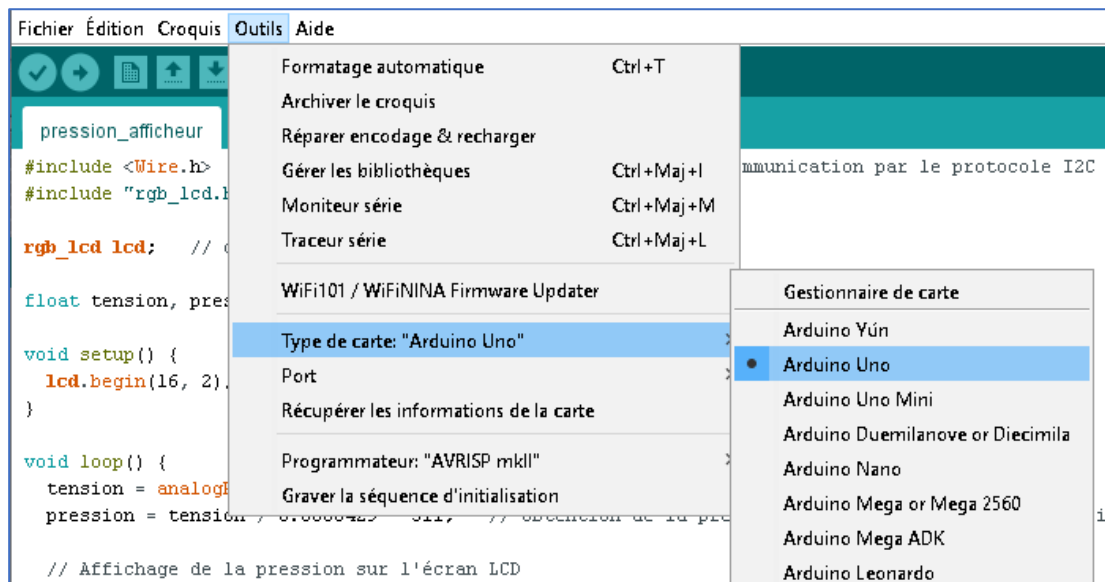
La carte Arduino est un microcontrôleur, c'est à dire une sorte de mini-ordinateur qui sert d'interface entre l'environnement (actions, mesures de grandeurs...) et un utilisateur. Elle se programme nativement dans un langage dérivé du C : le langage « Arduino ».

Pour cette activité on va utiliser les broches PWM notées 9, 10 et 11. Elles permettent de faire varier la tension qui alimente chacune des 3 leds incluses dans la led RVB.

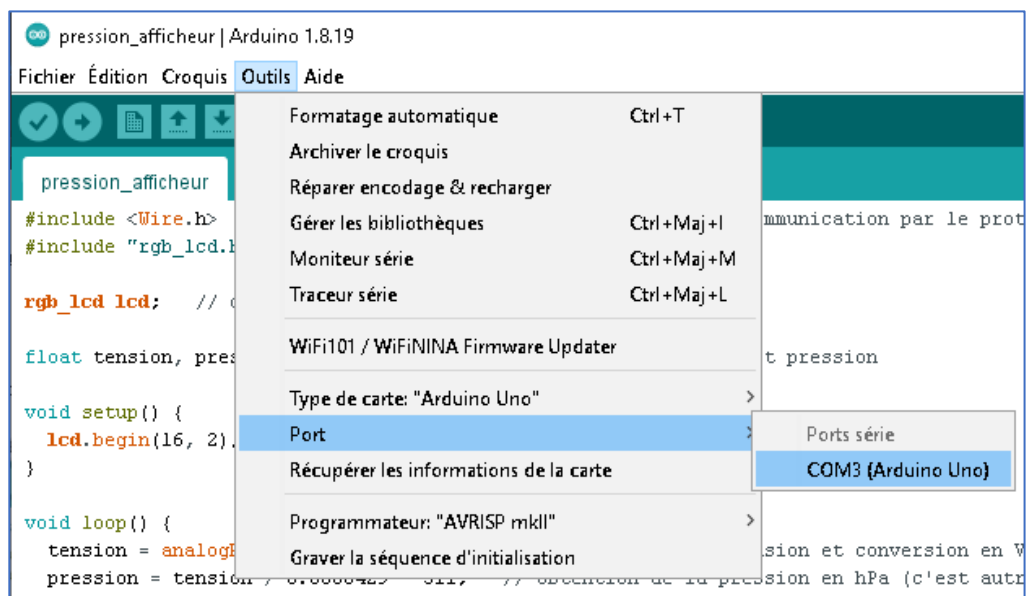


2. Le logiciel Arduino

Il est nécessaire de sélectionner la carte utilisée lors du TP. Aller dans le menu « Outils » et sélectionner « Arduino Uno ».



Il faut aussi choisir le port de communication, qui peut être différent selon l'ordinateur et le port USB utilisés :



3. Le programme du fichier `led_RVB.ino`

```
// Déclaration des noms des entrées Arduino reliées aux bornes de la led rgb
const int PIN_LED_R = 9;
const int PIN_LED_V = 10;
const int PIN_LED_B = 11;

// Durée d'éclairement de la led pour chaque couleur produite :
int duree = 2000;

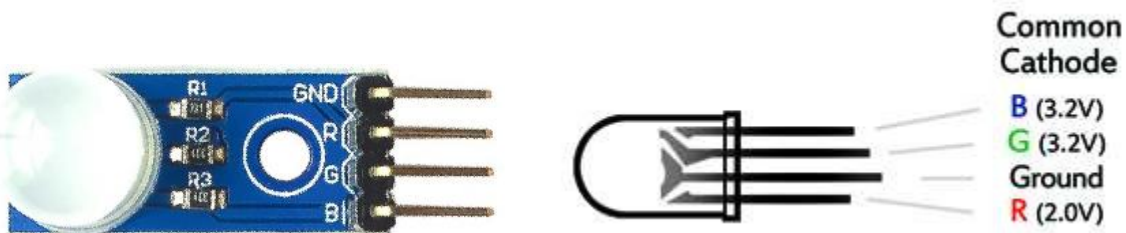
void setup() {
  // Initialisation des broches :
  pinMode(PIN_LED_R, OUTPUT); // Borne R de la led
  pinMode(PIN_LED_V, OUTPUT); // Borne G de la led
  pinMode(PIN_LED_B, OUTPUT); // Borne B de la led
}

// Fonction principale à modifier :
void loop() {
  affichage (255, 0, 0);
}

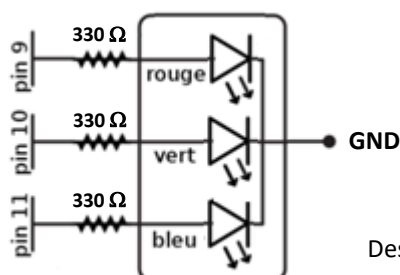
// Déclaration de la fonction affichage :
void affichage (int r, int g, int b) {
  analogWrite(PIN_LED_R, r);
  analogWrite(PIN_LED_V, g);
  analogWrite(PIN_LED_B, b);
  delay(duree); // La lumière est produite pendant une "durée" donnée (2000 ms = 2 s par défaut).
}
```

4. La led RVB

La diode est un modèle à cathode commune (remarquer la borne GND), le codage sera donc le même que celui du système RVB.



Il aurait été intéressant de travailler la loi des mailles et la loi d'Ohm car les pins 9, 10 et 11 délivrent une tension allant de 0 à 5 V (qui correspond au codage de 0 à 255 envoyé par la carte).



LED	I _{max}	U _{min}	U _{max}
Red	20mA	1.7V	2.3V
Green	20mA	3.0V	3.6V
Blue	20mA	3.0V	3.6V

Des mesures ont donné : $U_{\text{Red}} = 2,2 \text{ V}$, $U_{\text{Green}} = 2,7 \text{ V}$, $U_{\text{Blue}} = 2,8 \text{ V}$ sous une DDP totale maximum de 4,9 V (et pas 5 V !).