

5. Exercice 4 – Effet de fondu d'une LED

5.1. Objectifs

Le programme ci-après permet la décoloration d'une LED. Un bouton poussoir est relié sur la broche 3 de l'Arduino (configuré comme une entrée). Une LED est connectée à la broche 9 de l'Arduino (configuré en sortie). L'objectif est de créer un effet de fondu de luminosité de la LED lors d'un appui sur un bouton. Afin de créer cet effet de fondu, on utilise un signal PWM (Pulse Width Modulation) permettant de moduler le signal de sortie (plus de détail dans la partie 5.3 de ce chapitre).

5.2. Schéma de câblage

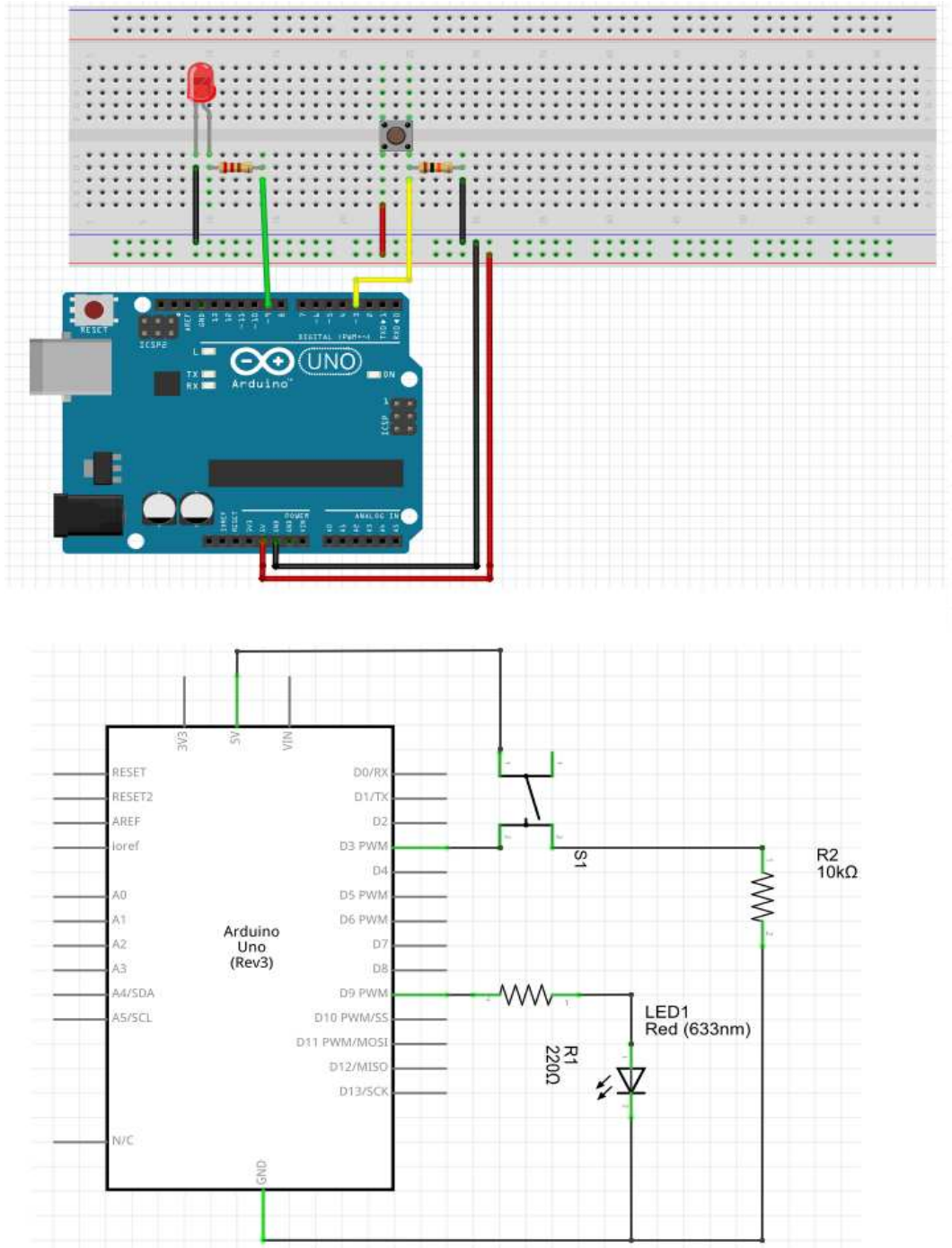


Figure 14 Schéma de câblage Exercice 4

5.3. Technologie des composants

Définition d'un signal PWM (définition reprise du site Arduino)

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off.

The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to `analogWrite()` is on a scale of 0 - 255, such that `analogWrite(255)` requests a 100% duty cycle (always on), and `analogWrite(127)` is a 50% duty cycle (on half the time) for example.

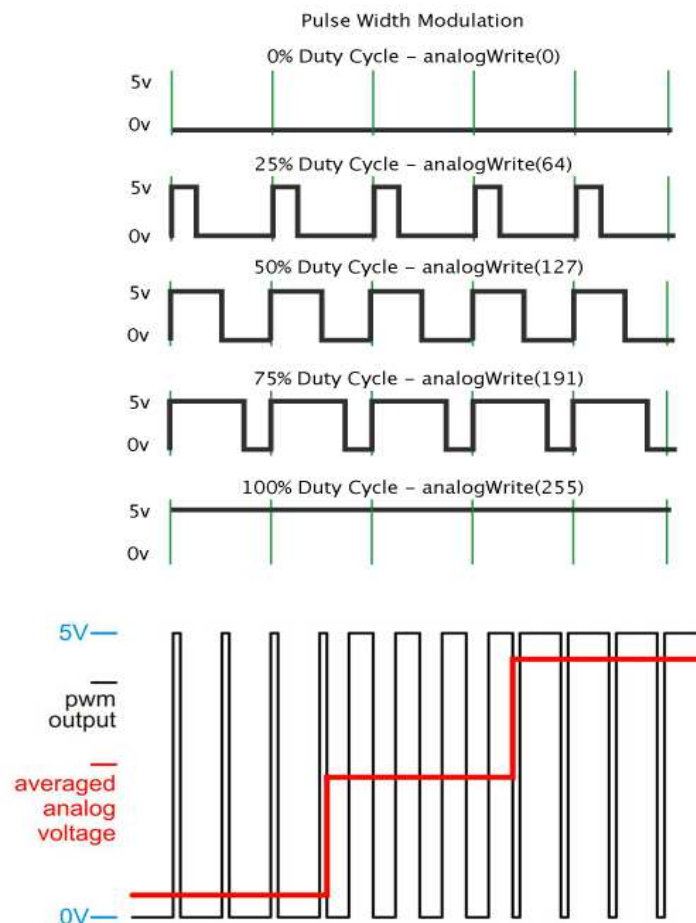


Figure 15 Technologie Signal PWM – Exercice 4

5.4. Principales fonction à découvrir

`digitalRead(pin)` : Lit la valeur de la broche numérique spécifiée « pin ». Cette fonction retourne une valeur HIGH (niveau binaire 1) ou LOW (niveau binaire 0).

`analogWrite(pin,value)` : Génère une tension analogique « (signal PWM) sur la broche « pin »

5.5. Le programme

```
int led = 9;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by
int button = 3;        //the pin that the button is attached to

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop()
{
  if (digitalRead(button) == HIGH)
  {
    // set the brightness of pin 9:
    analogWrite(led, brightness);
    // change the brightness for next time through the loop:
    brightness = brightness + fadeAmount;
    // reverse the direction of the fading at the ends of the fade:
    if (brightness == 0 || brightness == 255) {
      fadeAmount = -fadeAmount ;
    }
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

Figure 16 Exemple de programme pour l'Exercice 4

5.6. Les écueils prévisibles

Problème	Causes	Remèdes
La LED ne s'éclaire pas	La LED n'est pas bien connectée	Brancher correctement la LED (sens + pin Arduino)
	Vous utilisez des points de la breadboard non reliés entre eux électriquement	Analysez bien le schéma de câblage tel qu'il vous est donné, ainsi que le schéma électrique de la breadboard, et corrigez si nécessaire
	Le bouton n'est pas bien connecté	Brancher correctement le bouton poussoir (vérifier le sens)
La compilation génère une erreur	La compilation génère une erreur	Vous avez fait une erreur de syntaxe, n'avez-vous pas oublié un point-virgule à la fin de chaque ligne