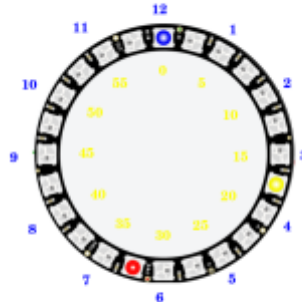


# Projet Horloge Neopixel arduino Brouillon GL



12 = Heures  
17 = Minutes  
xx = Secondes

[Projets sur thinkercad GL Brouillon](#)

[Projet avec neopixel arduino Module RTC et potentiometre](#)

[horloge\\_neopixel\\_Brouillon\\_GL\\_02062019.ino](#)

```
#include <Wire.h> // bibliotheque pour communiquer en I2C
#include <RTCLib.h> // bibliotheque pour module RTC DS1307
#include <Adafruit_NeoPixel.h> // Bibliotheque pour module Neopixel de
chez AdaFruit

#define PIN 6 // broche pour envoi de donnée sur Neopixel DI
#define NUMPIXELS 24 // nombre de Leds neopixels à piloter
#define Isecondes secondes*NUMPIXELS/60 // conversion pour afficher
60s sur un anneau de 24 pixels
#define Iminutes minutes*NUMPIXELS/60 // conversion pour afficher
60mn sur un anneau de 24 pixels
#define Iheures heures*NUMPIXELS/12 // conversion pour afficher
en mode 12h sur un anneau de 24 pixels

RTC_DS1307 rtc; // Initialise le module horloge DS1307 dans l'objet
rtc

/*Initialise le module neopixel avec le nombre de pixels utiliser , la
broche de raccordée sur l'entrée donnée, le mode utiliser , et la
vitesse du bus */

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);
```

```
int delayval = 800; // init le delais à 800 millis seconde
int secondes = 0;
int minutes = 0;
int heures = 0;

void AfSecondes(){
  pixels.clear();
  if (Iminutes == Isecondes && Iminutes == Iheures && Isecondes ==
Iheures){
    pixels.setPixelColor(Isecondes, pixels.Color(255,255,0));
    pixels.setPixelColor(Iminutes, pixels.Color(255,255,0));
    pixels.setPixelColor(Iheures, pixels.Color(255,255,0));
  }
  else if (Iminutes == Isecondes){
    pixels.setPixelColor(Isecondes, pixels.Color(255,255,0));
    pixels.setPixelColor(Iminutes, pixels.Color(255,255,0));
    pixels.setPixelColor(Iheures, pixels.Color(255,0,0));
  }
  else if (Isecondes == Iheures){
    pixels.setPixelColor(Isecondes, pixels.Color(255,0,255));
    pixels.setPixelColor(Iminutes, pixels.Color(0,128,0));
    pixels.setPixelColor(Iheures, pixels.Color(255,0,255));
  }
  else if (Iminutes == Iheures){
    pixels.setPixelColor(Isecondes, pixels.Color(0,0,255));
    pixels.setPixelColor(Iminutes, pixels.Color(192,192,192));
    pixels.setPixelColor(Iheures, pixels.Color(192,192,192));
  }
  else {
    pixels.setPixelColor(Isecondes, pixels.Color(0,0,255));
    pixels.setPixelColor(Iminutes, pixels.Color(0,128,0));
    pixels.setPixelColor(Iheures, pixels.Color(255,0,0));
  }
  pixels.show();
  delay(delayval);
}

void setup() {
  pixels.begin(); // demmarre le neopixel
  rtc.begin(); // demarre le moule rtc
  Serial.begin(57600);
  while (!Serial); // attends la communication serie
  pinMode(A1, INPUT);
}
```

```
void loop() {

pixels.setBrightness(map(analogRead(A1),0, 1023,0,255)); // ajuste la
brillance des neopixels entre 0 ( éteint ) et 255 (super brillant)

DateTime now = rtc.now(); // recupere la date et l'heure dans l'objet
now
heures = (now.hour());
if (heures >= 12){
    heures = heures - 12;
}
Serial.print(heures);
Serial.print(": ");
minutes = (now.minute());
Serial.print(minutes);
Serial.print(":");
secondes = (now.second());
Serial.println(secondes);

AfSecondes();

}

//https://www.rapidtables.com/web/color/RGB_Color.html
/*
Couleurs    Code (R,G,B
Black       (0,0,0)
White       (255,255,255)
Red         (255,0,0)
Lime        (0,255,0)
Blue        (0,0,255)
Yellow      (255,255,0)
Cyan        (0,255,255)
Fuchsia     (255,0,255)
Silver      (192,192,192)
Gray        (128,128,128)
Maroon      (128,0,0)
Olive       (128,128,0)
Green       (0,128,0)
Purple      (128,0,128)
Teal        (0,128,128)
Navy        (0,0,128)
*/
```

From:

<https://www.fablab37110.chanterie37.fr/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<https://www.fablab37110.chanterie37.fr/doku.php?id=start:arduino:horloge:brouillongl>

Last update: **2023/01/27 16:08**

