98101001
Yukari Connelley

Tap Sensing BPM Counter

Design Context

This device calculates the tempo or BPM of a series of inputs by a user and outputs it as a coloured LED. This design was inspired by the game idea for our group project, which involves playing music that matches a user’s actions. This component will become a vital part of the prototype, since the BPM will be the key factor in selecting what music to play. This device can also be used by musicians to determine the tempo of any song they like.

To use the device, simply turn it on and press the push-button at your desired beat rate.

Components

1x Arduino Uno

4x LEDs

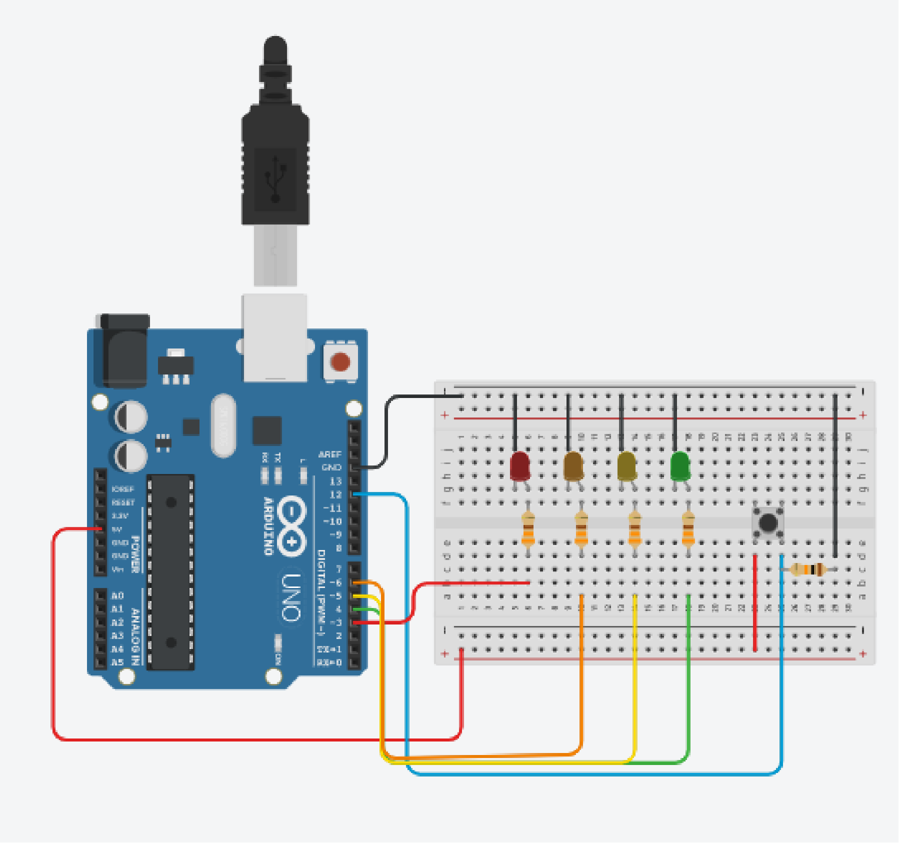
1x Push button

5x Resistors

1x Small Breadboard

13x Jumper wires

Hardware set-up



Arduino Code

```
//PINS
const int LED_RED = 3;
const int LED_ORN = 6;
const int LED_YLW = 5;
const int LED_GRN = 4;
const int pushButton = 12;

const long clickDelay = 50;
int BPM = 0;
unsigned long previous;
long lastClick = 0;
int buttonPush;
int lastButtonPush = LOW;
const int tapLength = 24;
unsigned long taps[tapLength];
int nextTap = 0;
int totalTaps = 0;
int timeSig = 4; //time signature


void setup() {
  Serial.begin(9600);

  pinMode(LED_RED, OUTPUT);
  pinMode(LED_ORN, OUTPUT);
  pinMode(LED_YLW, OUTPUT);
  pinMode(LED_GRN, OUTPUT);
  pinMode(pushButton, INPUT);

  previous = millis();
}

void loop() {
  int elapsed = millis() - previous;

  int reading = digitalRead(pushButton);


  if (reading != lastButtonPush) {
    // reset the clicking timer
    lastClick = millis();
  }

  if ((millis() - lastClick) > clickDelay) {
    if (reading != buttonPush) {
      buttonPush = reading;
    }
  }

     if (buttonPush == HIGH) {
     delay(250);
       if((millis() - lastClick) > 2000) {
                       reset_taps();
                 }
```

```
                taps[nextTap] = millis();
                BPM = bpms();
                Serial.println("BPM: " + String(BPM));
                nextTap = ++nextTap % tapLength; //stores last 24
taps
                totalTaps++;
         }

  lastButtonPush = reading;

      if(totalTaps < timeSig || BPM == 0) {
           if(BPM == 0) {
           return;
           }
      }


  if(BPM > 159) {
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_ORN, LOW);
    digitalWrite(LED_YLW, LOW);
    digitalWrite(LED_GRN, LOW);
  }

  if (BPM < 160 && BPM > 119) {
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_ORN, HIGH);
    digitalWrite(LED_YLW, LOW);
    digitalWrite(LED_GRN, LOW);
  }

  if (BPM < 120 && BPM > 79) {
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_ORN, LOW);
    digitalWrite(LED_YLW, HIGH);
    digitalWrite(LED_GRN, LOW);
  }

  if (BPM < 80){
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_ORN, LOW);
    digitalWrite(LED_YLW, LOW);
    digitalWrite(LED_GRN, HIGH);
  }

}


int bpms() {
  if(totalTaps < timeSig) {
    return 0;
  }

  unsigned long total = 0;
  for(int i=1; i<=timeSig; i++) {
    int tap = nextTap - i;

    if(tap < 0) {
      tap = tapLength + tap;
    }

    if(i>1) {

      total += (taps[tap+1%tapLength] - taps[tap]); //formula for
Beats
    }
  }

  return 60000/(total / (timeSig - 1)); //formula for BPM
}

    void reset_taps() {
  for(int i=0;i<tapLength;i++) {
    taps[i] = 0;
  }
  nextTap = 0;
  totalTaps = 0;
}
```

Video Tutorial

https://youtu.be/oKyzxnzku4o