

DgTricolor

3 colors controlled by 2 bits

<table border="0"> <tr> <td style="padding-right: 10px;">0 0</td> <td style="border-left: 1px solid black; padding-left: 10px;">dark</td> </tr> <tr> <td style="padding-right: 10px;">0 1</td> <td style="border-left: 1px solid black; padding-left: 10px;">Blue</td> </tr> <tr> <td style="padding-right: 10px;">1 0</td> <td style="border-left: 1px solid black; padding-left: 10px;">Red</td> </tr> <tr> <td style="padding-right: 10px;">1 1</td> <td style="border-left: 1px solid black; padding-left: 10px;">Green</td> </tr> </table>	0 0	dark	0 1	Blue	1 0	Red	1 1	Green			<p style="text-align: center;">Breadboard options</p>
0 0	dark										
0 1	Blue										
1 0	Red										
1 1	Green										



Le DgTricolor vous permet accessoirement de voir une led couleur allumée. Le but est de comprendre comment on programme, et comment on peut commander les trois leds R G B par seulement 2 signaux grove.

Pour comprendre le schéma, il faut savoir qu'avec 2 bits on peut coder 4 états. Une table de vérité exprime le choix fait et le schéma résulte des équations logiques.

Le schéma logique utilise des portes Nand et pour définir les niveaux d'entrée si rien n'est connecté, des résistance "pull-down" imposent l'état par défaut 00, led éteinte.

Le logiciel commence par définir les états qui nous intéresse. Vous êtes mal à l'aise avec le #define? voir www.didel.com/coursera/LC1.pdf

```
#define Pin1 8
#define Pin2 9
#define BlueOn digitalWrite (Pin1,1); digitalWrite (Pin2,0)
#define RedOn digitalWrite (Pin1,0); digitalWrite (Pin2,1)
#define GreenOn digitalWrite (Pin1,1); digitalWrite (Pin2,0)
#define RGBOff digitalWrite (Pin1,0); digitalWrite (Pin2,0)
void setup () {
  pinMode (Pin1,1);
  pinMode (Pin2,1);
}
```

Ce groupe d'instructions se trouvera au début de tous nos fichier, mentionnés ici par ...déf, setup On a maintenant des programmes bien lisibles. Clignotons (les programmes se trouvent sous www.didel.com/digrove/DgTricolor.zip) et les recommandation de travail sous www.didel.com/coursera/GenererFichiers.pdf

```
//DgTricol1.ino
...déf, setup
void loop {
  BlueOn; delay (500);
  RedOn; delay (500);
  GreenOn; delay (500);
  BlueOn; delay (500);
  RGBOff; delay (1000);
}
```

Ce programme teste les définitions. On peut maintenant écrire des séquences d'allumage avec des timings variés, mélanger les couleurs, programmer des boucles. A noter que pour mélanger les couleurs, il faut commuter à plus de 50 Hz.

Pour une intensité variable, vous pensez naturellement au PWM Arduino. Il faut alors se brancher sur les pins 5 ou 6, ou 9 et 10. On a des mélanges de couleur intéressant avec des PWM de différentes valeurs. Essayez et comprenez.

Ce qui est intéressant du point de vue programmation, c'est de définir trois variables, et avoir les 3 couleurs prenant la valeur Pwm de ces variables. L'intensité maximale va toutefois être réduite de un tiers, puisque on ne peut avoir qu'une couleur à la fois.

La routine PWM pour une sortie est simple, plus de détails sous www.didel.com/diduino/PwmPfm.pdf

<pre> for (int j=1 ; j<16 ; j++) { delay (1) ; if (pwmR < j) BlueOn ; else RGBOff; } </pre>	<p>La valeur du PWM est entre 0 et 15. Le cycle est de 16 ms.</p>
---	---

On peut ajouter les autres couleurs dans la boucle for

<pre> for (int j=1 ; j<16 ; j++) { delay (1) ; if (pwmB < j) BlueOn ; else RGBOff; if (pwmR < j) RedOn ; else RGBOff; if (pwmG < j) GreenOn ; else RGBOff; } </pre>	
---	--



The DgTricolor is interesting to understand. One learn about logic gates, truth tables and software. How to control 3 colors with 2 bits and not 3? Applications use 3 bits, the interest of 2 is you learn about tricks and get a better understanding of decoding and timing.

The truth table on top shows that 2 bits give the 4 possibilities we need. NAND gates generate the 3 signals with the good polarity to light the leds. Pull-down resistors guarantee the state off if no connection are established.

You know how to write a dirty blink program with digital.Writes on the pins and delays.

<p>Let us make it clean with definitions of functional names that translate the truth table. We can now use the instruction RedOn; in the program and ignore how it is encoded.</p>	<pre> #define Pin1 8 #define Pin2 9 #define BlueOn digitalWrite (Pin1,1); digitalWrite (Pin2,0) #define RedOn digitalWrite (Pin1,0); digitalWrite (Pin2,1) #define GreenOn digitalWrite (Pin1,1); digitalWrite (Pin2,1) #define RGBOff digitalWrite (Pin1,0); digitalWrite (Pin2,0) void setup () { pinMode (Pin1,1); pinMode (Pin2,1); } </pre>
---	--

This set of definitions will be reused in all our TriCol programs. I can be a small library called by a #include <DrTricol.h>.

Let us blink one led after the other. (Program can be found in www.didel.com/digrove/DgTricolor.zip)

<pre> //DgTricol1.ino ...déf, setup void loop { BlueOn; delay (500); RedOn; delay (500); GreenOn; delay (500); RGBOff; delay (1000); } </pre>	<p>Limitation of this encoding is you can light one color at a time. Mixing is done using eye persistance. Replace the delay of 500 ms by 5ms and remove the last line. You will see the mix of 3 colors.</p>
---	---

In order to modify the intensity of the 3 colors, one think to the Arduin PWM, but it is not applicable due to the multiplexing.Anyway try to use the PWM on pins 5 ou 6, ou 9 et 10. Try and understand.

