File "/CAROLIGHTPROJECT/main.cpp" printed from os.mbed.com on 04/06/2022

```
/*  Test DMXS12                                                          */
/***********************************************************************/
/*  LEnsE / Julien VILLEMEJANE       /   Institut d'Optique Graduate School */
/***********************************************************************/
/*  Brochage                                                             */
/*       TO COMPLETE                                                     */
/***********************************************************************/
/*  Test réalisé sur Nucléo-L476RG                                       */
/***********************************************************************/

#include "mbed.h"
#include "platform/mbed_thread.h"
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#define SAMPLES     512
Serial      debug_pc(USBTX, USBRX);

// Signaux DMX
Serial      dmx(A0, A1);
DigitalOut  out_tx(D5);
DigitalOut  start(D4);
DigitalOut  enableDMX(D6);
DigitalOut led (D4);


//Sorties associées au traitement des positions
InterruptIn   mvt(PC_1);

//Sorties associées au traitement des couleurs
PwmOut Strobe(D10); //Déclaration de la sortie pour le signal d'horloge
PwmOut Reset(D9);
PwmOut ClockPB(D11);
AnalogIn clr(PC_2);

InterruptIn ResetIn(PC_0);
InterruptIn StrobeIn (PC_3);

//AnalogIn    variationR(PC_0);
//AnalogIn    variationG(PC_2);
//AnalogIn    variationB(PC_3);


// DMX
char dmx_data[SAMPLES] = {0};
char mb = 0;

void initDMX();
void updateDMX();

char angle1[4] = {0, 29, 58, 29}; /*angle horizontal*/
char angle2[4] = {100, 0, 100, 0}; /*angle vertical*/


// MOSI, MISO, SCK, CSN, CE, IRQ
InterruptIn     my_bp(USER_BUTTON);


int compteur = 0; //permet de se déplacer dans la liste des positions (angles)

void majcompteur(){
    compteur = (compteur +1)%4;
    debug_pc.printf("test\n");

} //4 étant la longueur de la liste des positions

int ListeCouleurs[7] = {0,0,0,0,0,0,0}; // liste d e1 et 0 correspondant à la présence dans le signal des fréquences associées aux couleurs suivantes :
/*
ListeCouleurs[0] = violet (75%B 50%R)
ListeCouleurs[1] = bleu (100%B)
ListeCouleurs[2] = cyan (50%B 50%V)
ListeCouleurs[3] = vert (100%V)
ListeCouleurs[4] = jaune (50%V 50%R)
ListeCouleurs[5] = rouge (100%R)
ListeCouleurs[6] = magenta (75%R 25%B)
*/
int R=0;
int V= 0;
int B = 0;

void SimulCouleurs (){      //Simule un résultat de filtrage en fréquence
    int k;
    for (k=0; k<7; k++){
        ListeCouleurs[k] = (rand()%2);
    }
}

int compteur_clr = 0;

void maj_compteur_clr (){
    compteur_clr = compteur_clr+1;
    }

void DetectionCouleurs (){
    int compteur_clr = -1;
    while (compteur_clr < 6)
    StrobeIn.fall(&maj_compteur_clr);
    ListeCouleurs[6-compteur_clr] = clr.read();
    if (ListeCouleurs[6-compteur_clr] > 0.4){ //en % de 3.3
            ListeCouleurs[6-compteur_clr]=1;}
    else {
            ListeCouleurs[6-compteur_clr]=0;}
            }


 


 


 
```