

```

File : 'CAROLIGHTPROJECT/main.cpp' printed from os.mbed.com on 04/06/2022
1 /* Test DM512 */                                     */
2 #include "mbed.h"                                     */
3 #include "platform/mbed_thread.h"                      */
4 /* LienS / Juliet VILLEMEJANE / Institut d'Optique Graduate School */
5 #include "Brockage.h"                                */
6 /* TO COMPLETE */                                    */
7 #include "DMA.h"                                     */
8 #include "DMA512.h"                                  */
9 /* Test réalisé sur Nucleo-L476RG */                */
10 #include "mbed.h"                                    */
11 #include "SAMPLES.h"                                */
12 #include "DM512.h"                                   */
13 #include "platform/mbed_thread.h"                    */
14 #include "mbed.h"                                    */
15 #include "cstdlib.h"                                */
16 #include "ctime.h"                                   */
17 #include "math.h"                                    */
18
19 #define SAMPLES 512
20 Serial debug_pc(USBTX, USBRX);
21
22 // Signaux DMX
23 Serial dmxA(B, A1);
24 DigitalOut out_tx(D5);
25 DigitalIn in_rx(D6);
26 DigitalIn enableDMX(D6);
27 DigitalOut led(D4);
28
29 //Sorties associées au traitement des positions
30 InterruptIn metPC_1;
31
32 //Sorties associées au traitement des couleurs
33 PWMOut Strobe(D0); //Déclaration de la sortie pour le signal d'horloge
34 PWMOut Reset(D9); //Déclaration de la sortie pour le signal d'horloge
35 PWMOut my_hb(USER_BUTTON);
36 AnalogIn clr(PC_2);
37 AnalogIn in(PC_3);
38
39 InterruptIn ResinIn(PC_0);
40 InterruptIn StrobeIn(PC_3);
41
42 //AnalogIn variatione(PC_0);
43 //AnalogIn variationg(PC_2);
44 //AnalogIn variationb(PC_3);
45
46
47 // DMX
48 char dev_data[SAMPLES] = {0};
49 char nb = 0;
50
51 void initDMX();
52 void updateDMX();
53
54 char angle[4] = {0, 20, 50, 20}; //angle horizontal*
55 char angle[2] = {100, 0, 100, 0}; //angle vertical*
56
57 // MOSI, MISO, SCK, CSN, IRQ
58 //int MOSI, MISO, SCK, CSN, IRQ;
59 InterruptIn my_hb(USER_BUTTON);
60
61
62 int compteur = 0; //permet de se déplacer dans la liste des positions (angles)
63
64 void majCompteur(){
65     compteur += compteur +1*x;
66     debug_pcprintf("test");
67 }
68 // // etat la longueur de la liste des position
69
70 int listeCouleur[7] = {0,0,0,0,0,0,0}; // liste d et 0 correspondant à la présence dans le signal des fréquences associées aux couleurs suivantes :
71
72 listeCouleur[0] = violet (7550 serv)
73 listeCouleur[1] = bleu (7550)
74 listeCouleur[2] = cyan (7550 serv)
75 listeCouleur[3] = vert (7550)
76 listeCouleur[4] = jaune (7550 serv)
77 listeCouleur[5] = rouge (7550)
78 listeCouleur[6] = magenta (7550 255)
79
80 int R=0;
81 int G=0;
82 int B=0;
83
84 void simuleCouleurs () { //Simule un résultat de filtrage en fréquence
85     int i;
86     for (i = 0; i < 4; i++) {
87         listeCouleur[i] = (rand()>>1);
88     }
89 }
90
91 int compteur_clr = 0;
92 void maj_compteur_clr () {
93     compteur_clr = compteur_clr+1;
94 }
95
96 void detectionCouleur () {
97     int compteur_clr = 0;
98     while (compteur_clr < 6)
99     {
100         Strobe.period_us(1000); //déclaration de la fréquence du signal d'horloge en µs
101         Strobe.fall(); //compteur_clr++;
102         if (listeCouleur[0]-compteur_clr > 4) //on t de 3.3
103             listeCouleur[0]-compteur_clr++; //on t de 3.3
104         else {
105             listeCouleur[0]-compteur_clr=0;
106         }
107     }
108 }
109
110 void majcouleurs(){ // A refaire de la version définitive
111     /*SimuleCouleurs(); // A refaire de la version définitive
112     round((192*listeCouleur[0]+128*listeCouleur[1]+128*listeCouleur[2]+64*listeCouleur[3]+listeCouleur[4]+listeCouleur[5]+listeCouleur[6]);*/
113     R = round((192*listeCouleur[0]+128*listeCouleur[1]+128*listeCouleur[2]+64*listeCouleur[3]+listeCouleur[4]+listeCouleur[5]+listeCouleur[6]));
114     V = round((128*listeCouleur[0]+128*listeCouleur[1]+128*listeCouleur[2]+64*listeCouleur[3]+listeCouleur[4]+listeCouleur[5]+listeCouleur[6]));
115     B = round((128*listeCouleur[0]+128*listeCouleur[1]+128*listeCouleur[2]+64*listeCouleur[3]+listeCouleur[4]+listeCouleur[5]+listeCouleur[6]));
116 }
117
118 // Main
119 int main()
120 {
121     debug_pc_band(0x00);
122     debug_pcprintf("serial DM512\r\n");
123
124     mvt.siel(&majcouleurs); //Signal associé aux positions
125
126     while(1){
127
128         double t = 0.5; //rapport cyclique du signal d'horloge (portion du temps où le signal est en position haute)
129         Strobe.period_us(t); //déclaration de la fréquence du signal d'horloge en µs
130         Strobe.write(r);
131         double t = 0.5; //rapport cyclique du signal d'horloge (portion du temps où le signal est en position haute)
132         Reset.period_us(0.5); //déclaration de la fréquence du signal d'horloge en µs
133         Reset.write(r1);
134         double t = 0.5; //rapport cyclique du signal d'horloge (portion du temps où le signal est en position haute)
135         Clicks9.period_ms(1); //déclaration de la fréquence du signal d'horloge en ms
136         Clicks9.write(r2);
137
138         ResetIn.fall(detectionCouleur); //Signal associé aux couleurs
139
140         initDMX();
141         int entree;
142
143         majcouleurs();
144         dev_data[0] = angle[1]; //angle[1];
145         dev_data[1] = angle[2]; //angle[2];
146         dev_data[2] = angle[3]; //brillance (0 à 254)
147         dev_data[3] = R; // rouge (0 à 255)
148         dev_data[4] = V; // vert (0 à 255)
149         dev_data[5] = B; // bleu (0 à 255)
150         dev_data[6] = 0; //blanc (0 à 255)
151         /* dev_data[7] = variation.read()*25; */
152         dev_data[7] = 0;
153         updateDMX();
154
155         /* wait_us(10000); */
156     }
157 }
158
159
160 void initDMX(){
161     // Initialisation DMX
162     dev_data[0] = 0;
163     dev_data[1] = 0;
164     dev_data[2] = 0;
165     dev_data[3] = 0;
166     dev_data[4] = 0;
167     dev_data[5] = 0;
168     dev_data[6] = 0;
169     dev_data[7] = 0;
170     updateDMX();
171 }
172
173 void updateDMX(){
174     enableDMX = 1;
175     start = 1; // start
176     out_tx = 0; // break
177     wait_us(80);
178     out_tx = 1; // mb
179     wait_us(80);
180     out_tx = 0; // break
181     start = 1; // start
182     dev.putc(0); // Start
183     for(int k = 0; k < SAMPLES; k++){
184         dev.putc(dev_data[k]); // data
185     }
186     dev.putc(dev_data[1]); // data
187     wait_us(23000); // time between frame
188 }

```

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