Министерство образования и науки Российской Федерации

ФГАОУ ВПО «Уральский федеральный университет

имени первого Президента России Б.Н.Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ

Кафедра технологий и средств связи

ОТЧЕТ

по лабораторной работе №4

**«Работа с таймерными подсистемами»**

по дисциплине «Цифровые системы передачи»

Группа РИ-430801

Студент Ахтаров Д.Н.

Найков А.В.

Преподаватель Гусев А.В.

Екатеринбург 2016

# Цель работы

Научиться управлять светодиодами. Организовать переключение по нажатию кнопки.

# Алгоритм обработки нажатия кнопки

# Код программы

1. #include <LPC17xx.H>
2. #include "GLCD.h"
3. #include "LibraryDisplay.h"
5. **unsigned int data = 0;**
6. unsigned int data\_ready = 0;
7. unsigned int i0, tick=0, switching=0;
8. unsigned int A[8], B[3], C[3];
9. unsigned int i=0, j, ii=0, ij=0, djostic, dj, K, N, M, CCLK, x=150, y=110;
11. void TIMER2\_IRQHandler( void ) //18: Timer2
12. {
13. LPC\_TIM2->IR = 0x01; //clear the MR0 interrupt flag
14. switching++ ;
15. **if (switching >= 1000)**
16. {
17. switching = 0;
18. ii++ ;
19. if (ii >= 8)
20. **{**
21. ii = 0 ;
22. }
23. GLCD\_Bargraph(x, y, 5, 5, Blue);
24. }
25. **}**
27. void SysTick\_Handler( void )
28. {
29. tick++;
30. **if (tick >= 500)**
31. {
32. tick = 0 ;
33. ij++;
34. if (ij >= 8)
35. **{**
36. ij = 0 ;
37. }
38. }
39. }
41. void delay(int count){
42. for (i0 = 0; i0 < count; i0++);
43. }
45. **void port\_init(){**
46. LPC\_GPIO2->FIODIR |= (1<<2) | (1<<3) | (1<<4) | (1<<5) | (1<<6); // P2.2 - P2.6 - OUT
47. LPC\_GPIO1->FIODIR |= (1<<28) | (1<<29) | (1<<31); // P1.28, P1.29, P1.31 - OUT
48. }
50. **void TIMER0\_init(){**
51. // Program Timer 0 - Match Register 0
52. LPC\_TIM0->CTCR = 0x0; // mode - timer
53. LPC\_TIM0->PR = 0x0; // prescaler = 0, every clock
54. LPC\_TIM0->TCR = 0x2; // Disable
55. **LPC\_TIM0->MR0 = 9400-1; //10000 ; //202-1;**
56. LPC\_TIM0->MCR = 0x03; // interrup and Reset on Match MR0
57. //LPC\_TIM0->EMR = 0x31; // Toogle pin MAT0.0 P1.28,
59. //NVIC\_SetPriorityGrouping( 4 ); //sets PRIGROUP to 3:2 (XXX:YY)
60. **NVIC\_SetPriority( TIMER0\_IRQn, 2 ); //000:00 (bit 7:3) assign eint0 to group 0, sub-priority 0 within group 0**
61. NVIC\_EnableIRQ( TIMER0\_IRQn );
63. LPC\_TIM0->TCR = 0x1; // Enable timer 0
65. **//NVIC\_DisableIRQ( TIMER0\_IRQn );**
66. }

69. void TIMER2\_init(){
70. **// Program Timer 2 - Match Register 0**
71. //LPC\_TIM2->CTCR = 0; // mode - timer
72. LPC\_TIM2->TCR = 0x2; // Disable
73. LPC\_TIM2->PR = 0x0; // prescaler = 0, every clock
74. LPC\_TIM2->MCR = 0x03; // interrup and Reset on Match MR1
75. **LPC\_TIM2->MR0 = 12600-1; //10000 ; //202-1;**
76. //LPC\_TIM2->EMR = 0x31; // Toogle pin MAT0.0 P1.28,
78. //NVIC\_SetPriority( TIMER2\_IRQn, 2 ); //000:00 (bit 7:3) assign eint0 to group 0, sub-priority 0 within group 0
79. NVIC\_EnableIRQ( TIMER2\_IRQn );
81. LPC\_TIM2->TCR = 0x1; // Enable timer 0
83. //NVIC\_DisableIRQ( TIMER2\_IRQn );
84. }

87. void TIMER01\_init(){
88. // Program Timer 0 - Match Register 0
89. LPC\_PINCON->PINSEL3 |= 0xC000000;
90. **LPC\_TIM0->CTCR = 0x0; // mode - timer**
91. LPC\_TIM0->TCR = 0x2; // Disable
92. LPC\_TIM0->PR = 0x0; // prescaler = 0, every clock
93. LPC\_TIM0->MR1 = 9400000-1; //10000 ; //202-1;
94. LPC\_TIM0->MCR = 0x10; // interrup and Reset on Match MR0
95. **LPC\_TIM0->EMR = 0xC2; // Toogle pin MAT0.0 P1.28,**
97. //NVIC\_SetPriority( TIMER0\_IRQn, 1 ); //000:00 (bit 7:3) assign eint0 to group 0, sub-priority 0 within group 0
98. //NVIC\_EnableIRQ( TIMER0\_IRQn );
99. LPC\_TIM0->TCR = 0x1; // Enable timer 0
101. //NVIC\_DisableIRQ( TIMER0\_IRQn );
102. }
104. void systick\_init(){
105. **SysTick->LOAD = 94000 - 1; */\* set reload register \*/***
106. SysTick->VAL = (0x00); */\* Load the SysTick Counter Value \*/*
107. SysTick->CTRL = (1 << 2)|(1<<0)|(1<<1);
108. }
110. **void display\_line(){**
111. GLCD\_Bargraph(5, 5, 5, 230, Purple);
112. GLCD\_Bargraph(5, 5, 310, 5, Purple);
113. GLCD\_Bargraph(310, 5, 5, 230, Purple);
114. GLCD\_Bargraph(5, 230, 310, 5, Purple);
115. **}**
117. void display\_color(){
118. GLCD\_SetTextColor(Navy);
119. //GLCD\_SetTextColor(Purple);
120. **}**
122. void display\_const(){
123. GLCD\_DisplayString (1, 1, "Display data:");
124. GLCD\_DisplayString (2, 1, "LED:");
125. **GLCD\_DisplayString (3, 3, "K=");**
126. GLCD\_DisplayString (4, 3, "N=");
127. GLCD\_DisplayString (5, 3, "M=");
128. GLCD\_DisplayString (6, 3, "CCLK= MHz");
129. }
131. int main (void)
132. {
133. SystemInit ();
134. init\_display();
136. A[0] = (1<<6);
137. A[1] = (1<<5);
138. A[2] = (1<<4);
139. A[3] = (1<<3);
140. **A[4] = (1<<2);**
141. A[5] = (1<<31);
142. A[6] = (1<<29);
143. A[7] = (1<<28);
145. **port\_init();**
147. //TIMER0\_init();
148. TIMER01\_init();
149. TIMER2\_init();
150. **systick\_init();**
152. display\_line();
153. display\_color();
154. display\_const();
156. K = (LPC\_SC->CCLKCFG & 0xFF)%10 + 1;
157. N = (((LPC\_SC->PLL0CFG >> 16) & 0xFF)%10) + 1;
158. M = ((LPC\_SC->PLL0CFG & 0x7FFF) + 1) ;
160. **B[2] = M%10;**
161. B[1] = M/10%10;
162. B[0] = M/100%10;
164. CCLK = 2\*M\*12/N/K;
166. C[2] = CCLK%10;
167. C[1] = CCLK/10%10;
168. C[0] = CCLK/100%10;
170. **GLCD\_DisplayChar (3, 5, 0x30+K);**
171. GLCD\_DisplayChar (4, 5, 0x30+N);
172. for (j=0; j<3; j++)
173. {
174. GLCD\_DisplayChar (5, 5+j, 0x30+B[j]);
175. **}**
176. for (j=0; j<3; j++)
177. {
178. GLCD\_DisplayChar (6, 8+j, 0x30+C[j]);
179. }
180. **j=0;**
182. while (1)
183. {
184. if (i!=ii){
185. **i=ii;**
186. LPC\_GPIO1->FIOPIN ^= (1<<31);
187. //LPC\_GPIO2->FIOPIN ^= (1<<3);
188. GLCD\_DisplayChar (2, 5, 0x31+i);
189. }
190. **if (j!=ij){**
191. j=ij;
192. LPC\_GPIO2->FIOPIN ^= (1<<4);
193. }
194. }
195. **}**

# Вывод

В данной работе мы научились работать с портами ввода-вывода при помощи сдвига информации по разрядам. Так же реализовали переключение направления хода при нажатии на джойстик.