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 №6

**«Работа с ШИМ»**

по дисциплине «Техника микропроцессорных  
систем в коммутации»

Группа РИ-430801

Студент Ахтаров Д.Н.

Найков А.В.

Преподаватель Гусев А.В.

Екатеринбург 2016

# Цель работы

Научиться работать с ШИМ.

# Алгоритм

Все выполняется в цикле while. Мигают 4 светодиода: два от таймера, один от Systick и один от ШИМ.

# 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, CCLK, x=150, y=110, value=0;

12. void TIMER2\_IRQHandler();
13. void SysTick\_Handler();
14. void TIMER01\_IRQHandler();
15. **void delay(int);**
16. void port\_init();
17. void TIMER01\_init();
18. void TIMER2\_init();
19. void TIMER0\_init();
20. **void systick\_init();**
21. void display\_line();
22. void display\_color();
23. void ShIM\_init();
24. void InitADC ();
26. int main (void)
27. {
28. SystemInit ();
29. init\_display();
31. A[0] = (1<<6);
32. A[1] = (1<<5);
33. A[2] = (1<<4);
34. A[3] = (1<<3);
35. **A[4] = (1<<2);**
36. A[5] = (1<<31);
37. A[6] = (1<<29);
38. A[7] = (1<<28);
40. **port\_init();**
42. //TIMER01\_init();
43. TIMER0\_init();
44. TIMER2\_init();
45. **systick\_init();**
46. ShIM\_init();
47. InitADC();
49. display\_line();
50. **display\_color();**
52. while (1)
53. {
54. if((LPC\_ADC->ADGDR & (1<<31)) != 0)
55. **{**
56. value = (LPC\_ADC->ADGDR >> 4) & 0x0FFF;
57. LPC\_ADC->ADCR |= (1 << 24) | (1 << 2);
58. //GLCD\_DisplayU32(1, 1, 1, value);
59. ShIM\_init();
60. **}**
61. if (i!=ii){
62. i=ii;
63. LPC\_GPIO1->FIOPIN ^= (1<<31);
64. }
65. **if (j!=ij){**
66. j=ij;
67. LPC\_GPIO2->FIOPIN ^= (1<<5);
68. }
69. }
70. **}**
72. void TIMER2\_IRQHandler( void ) //18: Timer2
73. {
74. LPC\_TIM2->IR = 0x01; //clear the MR0 interrupt flag
75. **switching++ ;**
76. if (switching >= 1000)
77. {
78. switching = 0;
79. ii++ ;
80. **if (ii >= 8)**
81. {
82. ii = 0 ;
83. }
84. }
85. **}**
87. void SysTick\_Handler( void )
88. {
89. tick++;
90. **if (tick >= 500)**
91. {
92. tick = 0 ;
93. ij++;
94. if (ij >= 8)
95. **{**
96. ij = 0 ;
97. }
98. }
99. }
100. ***/\****
101. *void TIMER01\_IRQHandler( void ) //18: Timer0*
102. *{*
103. *LPC\_TIM0->IR = 0x01; //clear the MR0 interrupt flag*
104. *switching++ ;*
105. ***if (switching >= 1000)***
106. *{*
107. *switching = 0;*
108. *ii++ ;*
109. *if (ii >= 8)*
110. ***{***
111. *ii = 0 ;*
112. *}*
113. *}*
114. *}*
115. ***\*/***
116. void delay(int count){
117. for (i0 = 0; i0 < count; i0++);
118. }
120. **void port\_init(){**
121. LPC\_GPIO2->FIODIR |= (1<<2) | (1<<3) | (1<<4) | (1<<5) | (1<<6); // P2.2 - P2.6 - OUT
122. LPC\_GPIO1->FIODIR |= (1<<28) | (1<<29) | (1<<31); // P1.28, P1.29, P1.31 - OUT
123. }
125. **void TIMER01\_init(){**
126. // Program Timer 0 - Match Register 0
127. LPC\_TIM0->CTCR = 0x0; // mode - timer
128. LPC\_TIM0->PR = 0x0; // prescaler = 0, every clock
129. LPC\_TIM0->TCR = 0x2; // Disable
130. **LPC\_TIM0->MR0 = 9400-1; //10000 ; //202-1;**
131. LPC\_TIM0->MCR = 0x03; // interrup and Reset on Match MR0
132. //LPC\_TIM0->EMR = 0x31; // Toogle pin MAT0.0 P1.28,
134. //NVIC\_SetPriorityGrouping( 4 ); //sets PRIGROUP to 3:2 (XXX:YY)
135. **NVIC\_SetPriority( TIMER0\_IRQn, 2 ); //000:00 (bit 7:3) assign eint0 to group 0, sub-priority 0 within group 0**
136. NVIC\_EnableIRQ( TIMER0\_IRQn );
138. LPC\_TIM0->TCR = 0x1; // Enable timer 0
140. **//NVIC\_DisableIRQ( TIMER0\_IRQn );**
141. }

144. void TIMER2\_init(){
145. **// Program Timer 2 - Match Register 0**
146. //LPC\_TIM2->CTCR = 0; // mode - timer
147. LPC\_TIM2->TCR = 0x2; // Disable
148. LPC\_TIM2->PR = 0x0; // prescaler = 0, every clock
149. LPC\_TIM2->MCR = 0x03; // interrup and Reset on Match MR1
150. **LPC\_TIM2->MR0 = 12600-1; //10000 ; //202-1;**
151. //LPC\_TIM2->EMR = 0x31; // Toogle pin MAT0.0 P1.28,
153. //NVIC\_SetPriority( TIMER2\_IRQn, 2 ); //000:00 (bit 7:3) assign eint0 to group 0, sub-priority 0 within group 0
154. NVIC\_EnableIRQ( TIMER2\_IRQn );
156. LPC\_TIM2->TCR = 0x1; // Enable timer 0
158. //NVIC\_DisableIRQ( TIMER2\_IRQn );
159. }

162. void TIMER0\_init(){
163. // Program Timer 0 - Match Register 0
164. LPC\_PINCON->PINSEL3 |= 0xC000000;
165. **LPC\_TIM0->CTCR = 0x0; // mode - timer**
166. LPC\_TIM0->TCR = 0x2; // Disable
167. LPC\_TIM0->PR = 0x0; // prescaler = 0, every clock
168. LPC\_TIM0->MR1 = 9400000-1; //10000 ; //202-1;
169. LPC\_TIM0->MCR = 0x10; // interrup and Reset on Match MR0
170. **LPC\_TIM0->EMR = 0xC2; // Toogle pin MAT0.0 P1.28,**
172. //NVIC\_SetPriority( TIMER0\_IRQn, 1 ); //000:00 (bit 7:3) assign eint0 to group 0, sub-priority 0 within group 0
173. //NVIC\_EnableIRQ( TIMER0\_IRQn );
174. LPC\_TIM0->TCR = 0x1; // Enable timer 0
176. //NVIC\_DisableIRQ( TIMER0\_IRQn );
177. }
179. void systick\_init(){
180. **SysTick->LOAD = 94000 - 1; */\* set reload register \*/***
181. SysTick->VAL = (0x00); */\* Load the SysTick Counter Value \*/*
182. SysTick->CTRL = (1 << 2)|(1<<0)|(1<<1);
183. }
185. **void display\_line(){**
186. GLCD\_Bargraph(5, 5, 5, 230, Purple);
187. GLCD\_Bargraph(5, 5, 310, 5, Purple);
188. GLCD\_Bargraph(310, 5, 5, 230, Purple);
189. GLCD\_Bargraph(5, 230, 310, 5, Purple);
190. **}**
192. void display\_color(){
193. GLCD\_SetTextColor(Navy);
194. //GLCD\_SetTextColor(Purple);
195. **}**
197. void ShIM\_init(){
198. LPC\_PINCON->PINSEL4 |= (0x01<<8) ; // P2.4 - out PWM1.5
199. LPC\_PWM1->PCR |= 0x01<<13 ; //Enable PWM1 3 output
200. **LPC\_PWM1->MCR = 0x00000002 ; //On match with timer reset the counter**
201. LPC\_PWM1->PR = 0x00000000 ; //Load prescaler
202. LPC\_PWM1->MR0 = 16000 ; //set cycle rate to one second
203. LPC\_PWM1->MR5 = 4\*value; //set duty cycle to 0%
204. LPC\_PWM1->LER = (1<<3)|(1<<0) ; //enable shadow latch for match 0
205. **LPC\_PWM1->TCR = 0x00000002 ; //Reset counter and prescaler**
206. LPC\_PWM1->TCR = 0x00000009 ; //enable counter and PWM, release counter from reset
208. }
210. **void InitADC (void)**
211. {
212. LPC\_PINCON->PINSEL1 = (1<<18);
213. LPC\_ADC->ADCR = (1 << 2) |
214. (0xf << 12) |
215. **(0xf << 8) |**
216. (1 << 21) |
217. (1 << 24) ;
218. }

# Вывод

В данной работе мы научились работать с модулем ШИМ.