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 №3

**«Systick»**

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

# Алгоритм

# 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;
8. unsigned int A[8], B[3], C[3];
9. unsigned int i=0, j, ii=0, djostic, dj, K, N, M, CCLK, x=150, y=110;
11. void SysTick\_Handler( void )
12. {
13. tick = tick + 1 ;
14. if (tick >= 500)
15. **{**
16. tick = 0 ;
17. ii++;
18. if (ii >= 8)
19. {
20. **ii = 0 ;**
21. }
22. }
23. }
25. **void delay(int count){**
26. for (i0 = 0; i0 < count; i0++);
27. }
29. void port\_init(){
30. **LPC\_GPIO2->FIODIR |= (1<<2) | (1<<3) | (1<<4) | (1<<5) | (1<<6); // P2.2 - P2.6 - OUT**
31. LPC\_GPIO1->FIODIR |= (1<<28) | (1<<29) | (1<<31); // P1.28, P1.29, P1.31 - OUT
32. }
34. void systick\_init(){
35. **SysTick->LOAD = 94000 - 1; */\* set reload register \*/***
36. SysTick->VAL = (0x00); */\* Load the SysTick Counter Value \*/*
37. SysTick->CTRL = (1 << 2)|(1<<0)|(1<<1);
38. }
40. **void display\_line(){**
41. GLCD\_Bargraph(5, 5, 5, 230, Purple);
42. GLCD\_Bargraph(5, 5, 310, 5, Purple);
43. GLCD\_Bargraph(310, 5, 5, 230, Purple);
44. GLCD\_Bargraph(5, 230, 310, 5, Purple);
45. **}**
47. void display\_color(){
48. GLCD\_SetTextColor(Navy);
49. //GLCD\_SetTextColor(Purple);
50. **}**
52. void display\_const(){
53. GLCD\_DisplayString (1, 1, "Display data:");
54. GLCD\_DisplayString (2, 1, "LED:");
55. **GLCD\_DisplayString (3, 3, "K=");**
56. GLCD\_DisplayString (4, 3, "N=");
57. GLCD\_DisplayString (5, 3, "M=");
58. GLCD\_DisplayString (6, 3, "CCLK= MHz");
59. }
61. int main (void)
62. {
63. SystemInit ();
64. init\_display();
66. A[0] = (1<<6);
67. A[1] = (1<<5);
68. A[2] = (1<<4);
69. A[3] = (1<<3);
70. **A[4] = (1<<2);**
71. A[5] = (1<<31);
72. A[6] = (1<<29);
73. A[7] = (1<<28);
75. **port\_init();**
77. systick\_init();
79. display\_line();
80. **display\_color();**
81. display\_const();
83. K = (LPC\_SC->CCLKCFG & 0xFF)%10 + 1;
84. N = (((LPC\_SC->PLL0CFG >> 16) & 0xFF)%10) + 1;
85. **M = ((LPC\_SC->PLL0CFG & 0x7FFF) + 1) ;**
87. B[2] = M%10;
88. B[1] = M/10%10;
89. B[0] = M/100%10;
91. CCLK = 2\*M\*12/N/K;
93. C[2] = CCLK%10;
94. C[1] = CCLK/10%10;
95. **C[0] = CCLK/100%10;**
97. GLCD\_DisplayChar (3, 5, 0x30+K);
98. GLCD\_DisplayChar (4, 5, 0x30+N);
100. **for (j=0; j<3; j++)**
101. {
102. GLCD\_DisplayChar (5, 5+j, 0x30+B[j]);
103. }
104. for (j=0; j<3; j++)
105. **{**
106. GLCD\_DisplayChar (6, 8+j, 0x30+C[j]);
107. }
109. while (1)
110. **{**
111. if (i!=ii){
112. i=ii;
113. djostic = LPC\_GPIO1->FIOPIN & (1<<20);
114. if(djostic != 0)
115. **{**
116. if (i <= 4){
117. LPC\_GPIO2->FIOPIN = A[i];
118. LPC\_GPIO1->FIOPIN = 0;
119. } else {
120. **LPC\_GPIO1->FIOPIN = A[i];**
121. LPC\_GPIO2->FIOPIN = 0;
122. }
123. }
124. else{
125. **if (i <= 4){**
126. LPC\_GPIO2->FIOPIN = ~A[i];
127. LPC\_GPIO1->FIOPIN = ~0;
128. } else {
129. LPC\_GPIO1->FIOPIN = ~A[i];
130. **LPC\_GPIO2->FIOPIN = ~0;**
132. }
133. }
134. GLCD\_DisplayChar (2, 5, 0x31+i);
135. **}**
136. }
137. }

# 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.