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 №2

**«Работа с джойстиком»**

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 джойстиком и изменять частоту работы процессора.

# Алгоритм

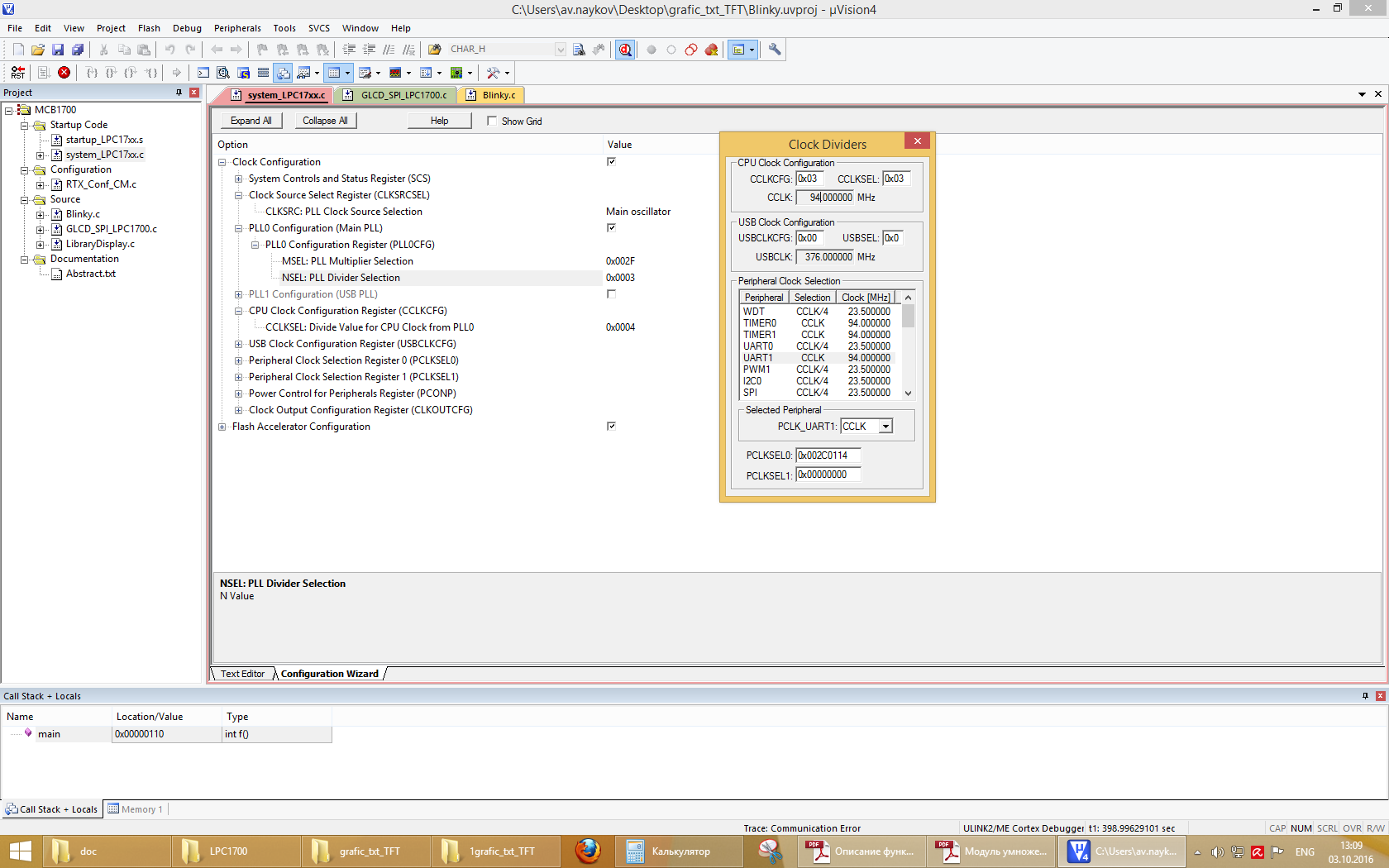
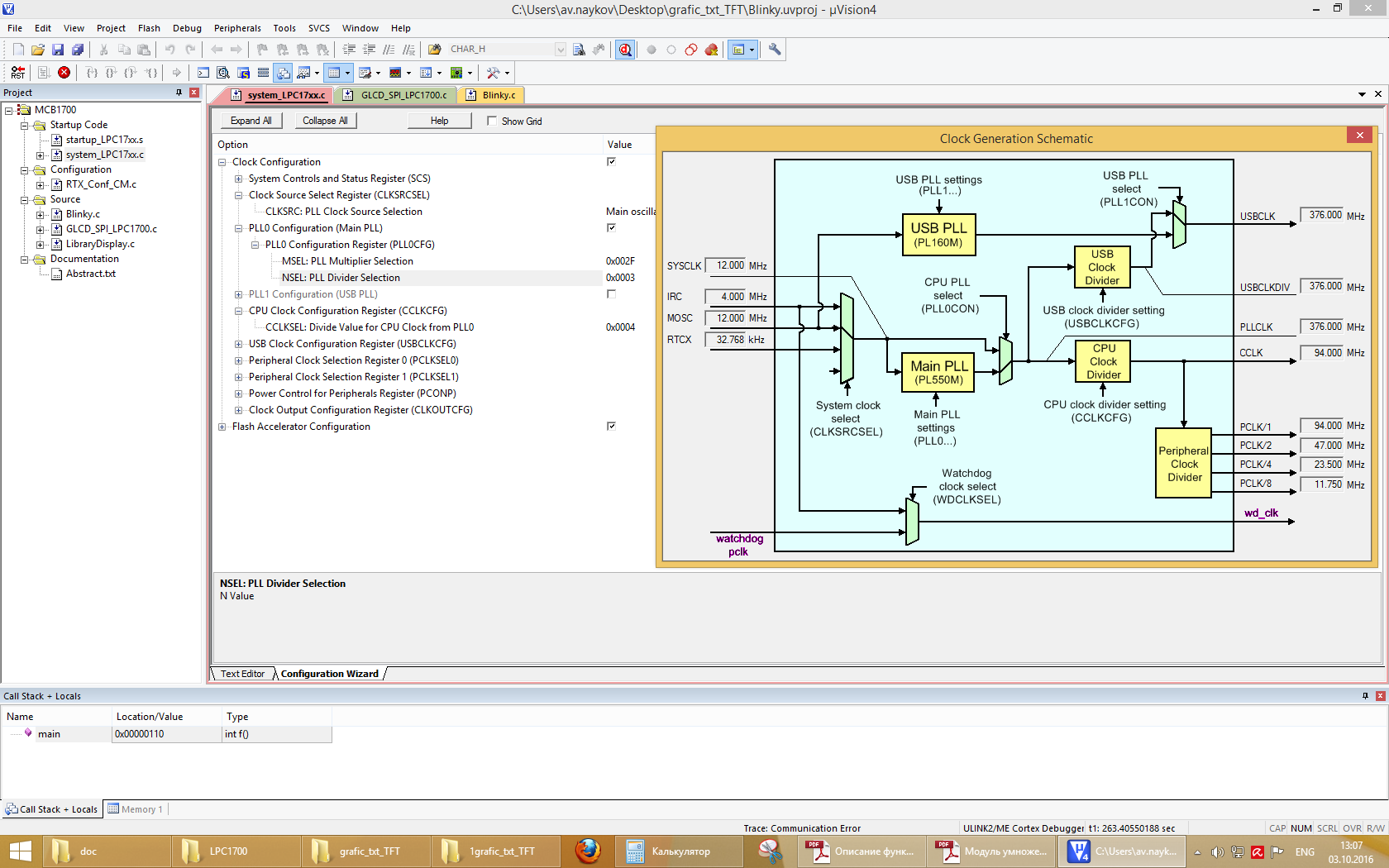
Задаем частоту процессора 94 МГц. Произвводим настройку в Wizard’е.

Значение M=47, N=3, K=4.

CCLK = 2\*M\*12/N/K.

Считываем значение джойстика: если нажат, то светодиоды мигают.

Каждое направление джойстика перемещает точку на экране.



# C:\Users\av.naykov\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана (4).png

# Код программы

1. #include <LPC17xx.H>
2. #include "GLCD.h"
3. #include "LibraryDisplay.h"
5. **void TIMER0\_IRQHandler( void );**
7. unsigned int data = 0;
8. unsigned int data\_ready = 0;
9. unsigned int i0 ;
10. **unsigned int A[8], B[3], C[3];**
11. unsigned int i=0, j, djostic, dj, K, N, M, CCLK, x=150, y=110;
13. void delay(int count){
14. for (i0 = 0; i0 < count; i0++);
15. **}**
17. void display\_line(){
18. GLCD\_Bargraph(5, 5, 5, 230, Purple);
19. GLCD\_Bargraph(5, 5, 310, 5, Purple);
20. **GLCD\_Bargraph(310, 5, 5, 230, Purple);**
21. GLCD\_Bargraph(5, 230, 310, 5, Purple);
22. }
24. void display\_color(){
25. **GLCD\_SetTextColor(Navy);**
26. GLCD\_SetTextColor(Purple);
27. }
29. void display\_const(){
30. **GLCD\_DisplayString (1, 1, "Display data:");**
31. GLCD\_DisplayString (2, 1, "LED:");
32. GLCD\_DisplayString (3, 3, "K=");
33. GLCD\_DisplayString (4, 3, "N=");
34. GLCD\_DisplayString (5, 3, "M=");
35. **GLCD\_DisplayString (6, 3, "CCLK= MHz");**
36. }
38. int main (void)
39. {
40. **SystemInit ();**
41. init\_display();
43. A[0] = (1<<6);
44. A[1] = (1<<5);
45. **A[2] = (1<<4);**
46. A[3] = (1<<3);
47. A[4] = (1<<2);
48. A[5] = (1<<31);
49. A[6] = (1<<29);
50. **A[7] = (1<<28);**
52. LPC\_GPIO2->FIODIR |= (1<<2) | (1<<3) | (1<<4) | (1<<5) | (1<<6); // P2.2 - P2.6 - OUT
53. LPC\_GPIO1->FIODIR |= (1<<28) | (1<<29) | (1<<31); // P1.28, P1.29, P1.31 - OUT
55. **display\_line();**
56. display\_color();
58. K = (LPC\_SC->CCLKCFG & 0xFF)%10 + 1;
59. N = (((LPC\_SC->PLL0CFG >> 16) & 0xFF)%10) + 1;
60. **M = ((LPC\_SC->PLL0CFG & 0x7FFF) + 1) ;**
62. B[2] = M%10;
63. B[1] = M/10%10;
64. B[0] = M/100%10;
66. CCLK = 2\*M\*12/N/K;
68. C[2] = CCLK%10;
69. C[1] = CCLK/10%10;
70. **C[0] = CCLK/100%10;**
72. display\_const();
73. GLCD\_DisplayChar (3, 5, 0x30+K);
74. GLCD\_DisplayChar (4, 5, 0x30+N);
75. **for (j=0; j<3; j++)**
76. {
77. GLCD\_DisplayChar (5, 5+j, 0x30+B[j]);
78. }
79. for (j=0; j<3; j++)
80. **{**
81. GLCD\_DisplayChar (6, 8+j, 0x30+C[j]);
82. }
84. while (1)
85. **{**
86. for (i = 0; i < 8; i++)
87. {
88. djostic = LPC\_GPIO1->FIOPIN & (1<<20);
89. if(djostic != 0)
90. **{**
91. if (i <= 4){
92. LPC\_GPIO2->FIOPIN = A[i];
93. LPC\_GPIO1->FIOPIN = 0;
94. } else {
95. **LPC\_GPIO1->FIOPIN = A[i];**
96. LPC\_GPIO2->FIOPIN = 0;
97. }
98. }
99. else{
100. **if (i <= 4){**
101. LPC\_GPIO2->FIOPIN = ~A[i];
102. LPC\_GPIO1->FIOPIN = ~0;
103. } else {
104. LPC\_GPIO1->FIOPIN = ~A[i];
105. **LPC\_GPIO2->FIOPIN = ~0;**
107. }
108. }
109. GLCD\_DisplayChar (2, 5, 0x31+i);
111. dj = LPC\_GPIO1->FIOPIN;
112. if ((dj & (1<<23)) == 0)
113. {
114. y=y-5;
115. **}**
116. if ((dj & (1<<24)) == 0)
117. {
118. x=x-5;
119. }
120. **if ((dj & (1<<25)) == 0)**
121. {
122. y=y+5;
123. }
124. if ((dj & (1<<26)) == 0)
125. **{**
126. x=x+5;
127. }
128. GLCD\_Bargraph(x, y, 5, 5, Purple);
129. delay(1000000);
130. **GLCD\_Bargraph(x, y, 5, 5, Blue);**
131. }
132. }
133. }

# Вывод

В данной работе мы научились работать с джойстиком и изменять частоту работы процессора.