

```

1  #include "MDR32Fx.h" //Device Header
2  #define F_CPU 8000000 // указание тактовой частоты МК
3  // #include "milkites_delay.h" // подключение библиотеки задержек
4
5  // Нужно определить эту константу, чтобы не отваливался JTAG на PORTB
6  #define USE_JTAG_A
7
8  // Нужно определить эту константу, чтобы не отваливался JTAG на PORTD
9  // #define USE_JTAG_B
10
11
12 // Подключение SPL
13 #include <MDR32F9Qx_port.h>
14 #include <MDR32F9Qx_rst_clk.h>
15
16
17 int main()
18 {
19     //Настройка
20     MDR_RST_CLK->PER_CLOCK = 0xffffffff; // вкл. тактирование всей периферии
21
22     // -----PORTA (для демонстрации) ---
23
24     // Инициализация структуры для настройки порта
25     PORT_InitTypeDef PORTA_config_struct;
26     PORT_StructInit(&PORTA_config_struct);
27
28     // Настройка порта
29     PORTA_config_struct.PORT_OE = PORT_OE_OUT;
30     PORTA_config_struct.PORT_MODE = PORT_MODE_DIGITAL;
31     PORTA_config_struct.PORT_SPEED = PORT_SPEED_MAXFAST;
32     PORT_Init(MDR_PORTA, &PORTA_config_struct);
33
34     PORT_SetBits(MDR_PORTA, PORT_Pin_All);
35
36
37
38
39     // -----PORTB-----
40
41     // Инициализация структуры для настройки порта
42     PORT_InitTypeDef PORTB_config_struct;
43     PORT_StructInit(&PORTB_config_struct);
44
45     // Настройка порта
46     PORTB_config_struct.PORT_Pin = PORT_Pin_All & (~JTAG_PINS(MDR_PORTB)); // Чтобы не отваливался
JTAG,
47                                     //исключаем из инициализации JTAG-овские пины
48     PORTB_config_struct.PORT_OE = PORT_OE_OUT;
49     PORTB_config_struct.PORT_MODE = PORT_MODE_DIGITAL;
50     PORTB_config_struct.PORT_SPEED = PORT_SPEED_MAXFAST;
51     PORT_Init(MDR_PORTB, &PORTB_config_struct);
52
53     // Работоспособность разных способов записи в порт B:
54
55     // Здесь поведение JTAG-а соответствует ожидаемому:
56     MDR_PORTB->RXTX = 0xff & (~JTAG_PINS(MDR_PORTB)); // JTAG будет работать дальше
57     //MDR_PORTB->RXTX = 0xffff & (~JTAG_PINS(MDR_PORTB)); // JTAG будет работать дальше
58     //MDR_PORTB->RXTX = 1 << 7; // JTAG будет работать дальше
59     //MDR_PORTB->RXTX = 0xff; // JTAG отвалится
60
61
62     // А ВОТ ДАЛЬШЕ САМОЕ ИНТЕРЕСНОЕ:
63
64     //PORT_Write(MDR_PORTB, 0xff & (~JTAG_PINS(MDR_PORTB))); // JTAG будет работать дальше
65     //PORT_Write(MDR_PORTB, 0xff); // JTAG отвалится !!! Хотя в функции
PORT_Write есть защита от записи в пины JTAG
66
67     //PORT_WriteBit(MDR_PORTB, PORT_Pin_All, 1); // JTAG отвалится !!! Хотя
в самой функции защита есть от записи в пины JTAG
68     //PORT_WriteBit(MDR_PORTB, PORT_Pin_All & (~JTAG_PINS(MDR_PORTB)), 1); // JTAG отвалится !!! И
даже маска в аргументе не поможет !!!!!!!!!!!
69     //PORT_WriteBit(MDR_PORTB, PORT_Pin_7, 1); // JTAG отвалится !!!
70
71     //PORT_SetBits(MDR_PORTB, PORT_Pin_All); // JTAG отвалится !!!
Хотя в самой функции защита есть от записи в пины JTAG
72     //PORT_SetBits(MDR_PORTB, PORT_Pin_All & (~JTAG_PINS(MDR_PORTB)) ); // JTAG отвалится !!! И

```

```
72     тут тоже маска в аргументе не помогает !!!!!!!!!!!  
73     //PORT_SetBits(MDR_PORTB, PORT_Pin_7 );                               // JTAG отвалится !!!  
74 }  
75
```