

```
1
2 #define F_CPU 16000000UL
3
4 #define bit_get(p,m) ((p) & (m))
5 #define bit_set(p,m) ((p) |= (m))
6 #define bit_clear(p,m) ((p) &= ~(m))
7 #define bit_flip(p,m) ((p) ^= (m))
8 #define bit_write(c,p,m) (c ? bit_set(p,m) : bit_clear(p,m))
9 #define BIT(x) (0x01 << (x))
10 #define LONGBIT(x) ((unsigned long)0x00000001 << (x))
11
12 #include "nrf24.h"
13 #include "Command_Handler.h"
14
15 #include <avr/io.h>
16 #include <string.h>
17 #include <stdlib.h>
18 #include <util/delay.h>
19
20 bool initRF();
21 void initIO();
22 void faultyRF_Alarm();
23
24 int main(void)
25 {
26     initIO();
27     initRF();
28
29     while (1)
30     {
31         if(nrf24_dataReady())
32         {
33             bit_clear(PORTB, BIT(0));
34
35             nrf24_getData(command_buffer);
36
37             bit_set(PORTD, BIT(7));
38             _delay_ms(500);
39             commandType currentCommand;
40             bool success = decomposeCommand(command_buffer, &currentCommand,
41                                             parameter);
42             if (success) { currentCommand.handlerFunction(); }
43             bit_clear(PORTD, BIT(7));
44         }
45         if (nrf24_checkAvailability()==false) { while(initRF()==false); }
46     }
47 }
48
49 void initIO(){
50     /*
51     Input/Output pin initialization
```

```

52      1 : OUTPUT | 0 : INPUT | 0b76543210 Bit order
53      ATTACHMENTS
54          RELAY 0      : PD3          |      OUTPUT
55          RELAY 1      : PD2          |      OUTPUT
56          RELAY 2      : PD6          |      OUTPUT
57          RELAY 3      : PD5          |      OUTPUT
58          RED LED      : PD7          |      OUTPUT
59          GREEN LED    : PB0          |      OUTPUT
60      nRF24L01
61          CE   : PC0          |      OUTPUT
62          CSN  : PC1          |      OUTPUT
63          MISO : PD0 (MSPIM MISO ATMEGA) |      INPUT
64          MOSI : PD1 (MSPIM MOSI ATMEGA) |      OUTPUT
65          SCK  : PD4 (MSPIM XCK)        |      OUTPUT
66      */
67      DDRD = 0b11111110;
68      DDRB = 0b00101001;
69      DDRC = 0b11011111;
70
71      PORTD = 0b00000000;
72      PORTC = 0b00000000;
73      PORTB = 0b00000000;
74  }
75
76  bool initRF(){
77      uint8_t tx_address[5] = {0xD7,0xD7,0xD7,0xD7,0xD7};
78      uint8_t rx_address[5] = {0xE7,0xE7,0xE7,0xE7,0xE7};
79
80      initliazeMemory();
81
82      /* Power down module */
83      nrf24_powerDown();
84
85      nrf24_init();
86
87      /* Channel #112 , payload length: 32 */
88      nrf24_config(112,32);
89
90      /* Check module configuration */
91      if (nrf24_checkConfig()==false) { faultyRF_Alarm(); return false; }
92
93      /* Set the device addresses */
94      nrf24_tx_address(tx_address);
95      nrf24_rx_address(rx_address);
96
97      /* Power up in receive mode */
98      nrf24_powerUpRx();
99
100     return true;
101 }
102
103 void faultyRF_Alarm(){

```

```
104     bit_clear(PORTD, BIT(7));
105     for (uint8_t x = 0; x < 6; x++)
106     {
107         bit_flip(PORTD, BIT(7));
108         _delay_ms(125);
109     }
110 }
111 }
112
113
114
115
```