```
 1  #ifndef NRF24
 2  #define NRF24
 3
 4  #ifndef F_CPU
 5  #define F_CPU 16000000UL
 6  #endif
 7
 8  #include "nRF24L01_Definitions.h"
 9  #include "Command_Handler.h"
10  #include <stdint.h>
11  #include <stdbool.h>
12  #include <avr/io.h>
13  #include <avr/delay.h>
14
15
16
17  #ifndef BIT_MANIPULATION_MACRO
18  #define BIT_MANIPULATION_MACRO 1
19  #define bit_get(p,m) ((p) & (m))
20  #define bit_set(p,m) ((p) |= (m))
21  #define bit_clear(p,m) ((p) &= ~(m))
22  #define bit_flip(p,m) ((p) ^= (m))
23  #define bit_write(c,p,m) (c ? bit_set(p,m) : bit_clear(p,m))
24  #define BIT(x) (0x01 << (x))
25  #define LONGBIT(x) ((unsigned long)0x00000001 << (x))
26  #endif
27
28  #define LOW 0
29  #define HIGH 1
30  #define nrf24_ADDR_LEN 5
31  #define nrf24_CONFIG ((1<<EN_CRC)|(0<<CRCO))
32  #define NRF24_TRANSMISSON_OK 0
33  #define NRF24_MESSAGE_LOST   1
34
35  #define AVAILABLE_COMMAND_BOARDS    2
36  #define CLEAR_FAULTY_RF_LED         bit_clear(PORTB, BIT(1))
37  #define FLIP_FAULTY_RF_LED          bit_flip(PORTB, BIT(1))
38
39
40  enum TransmissionMode {
41      RECEIVE,
42      TRANSMIT
43  };
44  typedef enum TransmissionMode TransmissionMode;
45
46  enum CommandsBoard {
47      MAIN_BOARD = 0,
48      POWER_BOARD = 1,
49      MOTORIZED_BOARD = 2
50  };
51  typedef enum CommandsBoard CommandsBoard;
52
```

```
53  extern void nrf24_initRF_SAFE(uint8_t boardIndex,TransmissionMode initMode);
54
55  void    nrf24_init();
56  void    nrf24_rx_address(uint8_t* adr);
57  void    nrf24_tx_address(uint8_t* adr);
58  void    nrf24_config(uint8_t channel, uint8_t pay_length);
59  bool    nrf24_checkRegister(uint8_t reg, uint8_t desiredValue, uint8_t len);
60  bool    nrf24_checkConfig();
61  bool    nrf24_checkAvailability();
62
63  void faultyRF_Alarm();
64
65
66
67  uint8_t nrf24_dataReady();
68  uint8_t nrf24_isSending();
69  uint8_t nrf24_getStatus();
70  uint8_t nrf24_rxFifoEmpty();
71
72  void    nrf24_send(uint8_t* value);
73  void    nrf24_getData(uint8_t* data);
74
75  uint8_t nrf24_payloadLength();
76
77  uint8_t nrf24_lastMessageStatus();
78  uint8_t nrf24_retransmissionCount();
79
80  uint8_t nrf24_payload_length();
81
82  void    nrf24_powerUpRx();
83  void    nrf24_powerUpTx();
84  void    nrf24_powerDown();
85
86  uint8_t spi_transfer(uint8_t tx);
87  void    nrf24_transmitSync(uint8_t* dataout,uint8_t len);
88  void    nrf24_transferSync(uint8_t* dataout,uint8_t* datain,uint8_t len);
89  void    nrf24_configRegister(uint8_t reg, uint8_t value);
90  void    nrf24_readRegister(uint8_t reg, uint8_t* value, uint8_t len);
91  void    nrf24_writeRegister(uint8_t reg, uint8_t* value, uint8_t len);
92
93  extern void nrf24_setupPins();
94
95  extern void nrf24_ce_digitalWrite(uint8_t state);
96
97  extern void nrf24_csn_digitalWrite(uint8_t state);
98
99  extern void nrf24_sck_digitalWrite(uint8_t state);
100
101 extern void nrf24_mosi_digitalWrite(uint8_t state);
102
103 extern uint8_t nrf24_miso_digitalRead();
104
```

```
105  #endif
106
```