

```
1  /*
2   * Firmware.c
3   *
4   * Created: 16.1.2017 17:16:21
5   * Author : atom2
6   */
7
8  /*
9   Nastavení fuse bits
10     High      0x91
11     Low       0xD7
12     Extended  0xFC
13     Bez Bootloader režimu
14  */
15
16
17
18 #include <avr/io.h>
19
20 #include "inc/def_init.h"
21 #include <util/delay.h>
22 #include <avr/interrupt.h>
23 #include <stdio.h>
24 #include <avr/eeprom.h>
25 #include <avr/wdt.h>
26 #include <string.h>
27 #include "inc/common_defs.h"
28 #include "inc/uart0.h"
29 #include "inc/uart1.h"
30 #include "inc/trinamic.h"
31
32
33 uint8_t uart0_error=0,uart1_error=0;
34
35 // Deklarování proměnných z uart0.c
36 uint8_t uart0_rx_ptr;
37 uint8_t uart0_buf_rx[BUFFER_CHAR_PACKET];
38 uint8_t uart1_buf_tx[BUFFER_CHAR_PACKET];
39
40 uint8_t uart1_rx_ptr;
41 uint8_t uart1_buf_rx[BUFFER_CHAR_PACKET];
42 uint8_t uart1_buf_tx[BUFFER_CHAR_PACKET];
43
44 uint8_t uart1_tx_flag;
45 uint8_t uart1_tx_iptr;
46
47 uint8_t uart1_rx_flag;
48 uint8_t uart1_rx_iptr;
49
50 uint8_t uart0_tx_flag;
51 uint8_t uart0_tx_ptr;
52 uint8_t uart0_tx_iptr;
53
54 uint8_t uart0_rx_flag;
55 uint8_t uart0_rx_ptr;
56 uint8_t uart0_rx_iptr;
```

```
57
58
59 Trinamicpac TR_Buf_In;
60 Trinamicpac TR_Buf_Out;
61
62
63 #define ADDRESS_EXT 100
64
65 uint32_t UB = 0;
66
67 int main(void)
68 {
69     UB = UART0_DEFAULT_BAUD;
70     uart0_error = uart0_init(UB);
71     UB = UART1_DEFAULT_BAUD;
72     uart1_error = uart1_init(UB);
73     uart0_interrupt_rx(TRUE);
74     uart0_interrupt_tx(TRUE);
75     uart1_interrupt_rx(TRUE);
76     uart1_interrupt_tx(TRUE);
77
78
79     //Pokud by nastal nějaký error případně rozšíření knihovny
80     //if uart0_error != 0
81
82     // Nastavení RS485 Enable
83     sbi(DDRD, DDD4);
84     // Nastavení RS485 Enable
85     sbi(DDRD, DDD5);
86     // UART0 RX
87     cbi(DDRD, DDD0);
88     // UART1 RX
89     cbi(DDRD, DDD2);
90     // UART0 TX
91     sbi(DDRD, DDD1);
92     // UART1 TX
93     sbi(DDRD, DDD3);
94     // LED1
95     sbi(DDRD, DDD6);
96     // LED2
97     sbi(DDRD, DDD7);
98
99     cbi(PORTD, PORTD7);
100    cbi(PORTD, PORTD6);
101
102    // Povolení příjmu dat
103    RS485_EN_INT_receive;
104    RS485_EN_EXT_receive;
105
106    // Povolení globálního přerušení
107    sei();
108    uint8_t j=0;
109
110    // Hlavní smyčka programu
111    while (1)
112    {
```

```
113 //uart1_transmit_char(0x01);
114 //RS485_EN_INT_receive;
115 // _delay_ms(500);
116 // RS485_EN_INT_transmite;
117 // uart0_tx_flag=true;
118 // UART0_UDR = 0x02;
119 //
120 // RS485_EN_EXT_transmite;
121 // uart1_tx_flag=true;
122 // UART1_UDR = 0x01;
123
124
125 j = check_uart0();
126 if (j==1)
127 {
128     // Naplní Buffer trinamic
129     TR_buf_fill_In();
130     // Kontrola adresy
131     if (TR_Buf_In.n.addr >= ADDRESS_EXT)
132     {
133         // Pokud je adresa externí - Odešlou se data na externí UART1
134         sbi(PORTD, PORTD7);
135         // Povolení odesílání dat na externí linku
136         uart1_tx_flag=TRUE;
137         RS485_EN_EXT_transmite;
138         UART1_UDR = TR_Buf_In.b[uart1_tx_iptr++];
139     }
140     else
141     {
142         cbi(PORTD, PORTD7);
143     }
144     //uart0_buf_rx[ptr-9]
145     uart0_rx_ptr;// = uart0_ptr();
146     //zpracuje trinamic příkaz
147 }
148 // Odeslání uart1 na uart0, nebo-li data z externí RS485 na Interní.
149 // check_uart1();
150
151 uint8_t jj=0;
152
153 if (uart1_rx_iptr > 8)
154 {
155     TR_buf_fill_Out();
156     uart1_rx_iptr=0;
157     // uart1_rx_flag = TRUE;
158     // RS485_EN_INT_transmite;
159     // uart0_tx_flag=TRUE;
160     // jj = TR_Buf_Out.b[uart0_tx_iptr++];
161     // UART0_UDR = jj;
162     // uart1_rx_iptr=0;
163 }
164
165 //
166 //
167 // if (uart1_rx_flag)
168 // {
```

```
169 //          //Odeslání 9 bytů dat
170 //          if (uart0_tx_iptr > 8)
171 //          {
172 //              // Vypnutí odesílání a povolení příjmu
173 //              uart1_rx_flag = FALSE;
174 //              RS485_EN_INT_receive;
175 //              uart0_tx_iptr=0;
176 //          }
177 //          else
178 //          {
179 //              RS485_EN_INT_transmite;
180 //              uart0_tx_flag=true;
181 //              UART0_UDR = uart0_buf_rx[uart0_tx_iptr++];
182 //          }
183 //      }
184
185
186 }
187 }
188
189
```