

```
1  /*
2  *  uart0.c
3  *
4  *  Created: 16.1.2017 17:39:01
5  *  Author: atom2
6  */
7
8  #include <avr/io.h>
9  #include <stdio.h>
10 #include <string.h>
11 #include "common_defs.h"
12 #include "uart0.h"
13 #include "def_init.h"
14 #include <avr/interrupt.h>
15 #include "trinamic.h"
16
17
18 #define MACRO_BAUDRATE(BAUDRATE) (UART0_UBRR0L = (((F_CPU) / (BAUDRATE * 16UL)) - 1 + (F_CPU % (BAUDRATE * 16UL) > (BAUDRATE * 8UL) ? 1 : 0)))
19
20
21 struct tmp
22 {
23     uint8_t tmpData;
24     uint8_t tmpStatus;
25     uint8_t tmpTimer;
26 };
27
28 struct tmp tmpUart;
29
30
31 volatile uint8_t uart0_buf_rx[BUFFER_CHAR_PACKET];
32 volatile uint8_t uart0_buf_tx[BUFFER_CHAR_PACKET];
33
34 volatile uint8_t uart0_sum=0;
35 volatile uint8_t uart0_i=0;
36 volatile uint8_t uart0_ret=0, uart0_check_sum=0;
37
38 volatile uint8_t uart0_tx_flag=FALSE;
39 volatile uint8_t uart0_tx_ptr=0;
40 volatile uint8_t uart0_tx_iptr=0;
41
42 volatile uint8_t uart0_rx_flag=FALSE;
43 volatile uint8_t uart0_rx_ptr=0;
44 volatile uint8_t uart0_rx_iptr=0;
45
46 Trinamicpac TR_Buf_Out;
47 uint8_t uart1_buf_rx[BUFFER_CHAR_PACKET];
48 uint8_t uart1_rx_iptr;
49
50 uint8_t uart0_init(uint32_t MYUBRR0)
51 {
52     uint16_t UBRR0_COUNT = 0;
53     // Výpočet rychlosti
54     UBRR0_COUNT = (((F_CPU) / (MYUBRR0 * 16UL)) - 1 + (F_CPU % (MYUBRR0 * 16UL) > (MYUBRR0 * 8UL) ? 1 : 0));
```

```
55 // Nastavení UBRR0 pro rychlost
56 UART0_UBRR0L = (unsigned char) (UBRR0_COUNT);
57 UART0_UBRR0H = (unsigned char) (UBRR0_COUNT >> 8);
58 // Povolení RX a TX pinů
59 UART0_UCSRB |= BV(UART0_TXEN) | BV(UART0_RXEN);
60 // 8bit, 1stop, no parity
61 UART0_UCSRC |= BV(UART0_UCSZ00) | BV(UART0_UCSZ01);
62 return 0;
63 }
64
65 uint8_t uart0_interrupt_rx(uint8_t enable)
66 {
67     if (enable)
68         UART0_UCSRB |= BV(UART0_RXIE);
69     else
70         UART0_UCSRB &= BV(UART0_RXIE);
71
72     return 0;
73 }
74
75 uint8_t uart0_interrupt_tx(uint8_t enable)
76 {
77     if (enable)
78         UART0_UCSRB |= BV(UART0_TXIE);
79     else
80         UART0_UCSRB &= BV(UART0_TXIE);
81
82     return 0;
83 }
84
85 uint8_t uart0_ptr_ask()
86 {
87     return uart0_rx_ptr;
88 }
89
90 uint8_t xx;
91
92 void uart0_receive_char(uint8_t data)
93 {
94
95     xx = uart0_buf_rx[uart0_rx_ptr] = data;
96     uart0_rx_ptr++;
97     uart0_rx_iptr++;
98     //uart0_rx_ptr;
99     //uart0_rx_iptr;
100
101     switch(uart0_rx_ptr)
102     {
103         case 8:
104             asm("nop");
105             break;
106         case 9:
107             asm("nop");
108             break;
109         case 10:
110             asm("nop");
```

```
111         break;
112     case 11:
113         asm("nop");
114         break;
115     case 12:
116         asm("nop");
117         break;
118     case 13:
119         asm("nop");
120         break;
121     case 14:
122         asm("nop");
123         break;
124     }
125 }
126
127
128 ISR(UART0_RX_vect)
129 {
130     tmpUart.tmpData = UART0_UDR;
131     tmpUart.tmpStatus = UART0_UCSRA;
132     tmpUart.tmpTimer = DEFAULT_TIMEOUT;
133     uart0_receive_char(tmpUart.tmpData);
134 }
135
136
137 ISR(UART0_TX_vect)
138 {
139     if (uart0_tx_flag)
140     {
141         RS485_EN_INT_transmite;
142         //Odeslání 9 bytů dat
143         if (uart0_tx_iptr > 8)
144         {
145             // Vypnutí odesílání a povolení příjmu
146             uart0_tx_flag = FALSE;
147
148             RS485_EN_INT_receive;
149             uart0_tx_iptr=0;
150         }
151     }
152     else
153     {
154         //UART0_UDR = 0x01;
155         UART0_UDR = TR_Buf_Out.b[uart0_tx_iptr++];
156         //uart0_tx_iptr++;
157     }
158 }
159
160
161
162
163 uint8_t check_uart0(uint8_t data)
164 {
165     // Vnitřní čítač 9 příchozích Bytů
166     if (uart0_rx_iptr > 8)
```

```
167     {
168         uart0_rx_flag = TRUE;
169         uart0_rx_ptr;
170         uart0_rx_iptr;
171     }
172     // Vypnutí přerušení před kontrolou dat
173     cli();
174     if (uart0_rx_flag)
175     {
176         uart0_sum=0;
177         uart0_i=0;
178         for (uart0_i=9; uart0_i>1; uart0_i--)
179         {
180             uart0_sum += uart0_buf_rx[uart0_rx_ptr-uart0_i];
181         }
182         uart0_check_sum = uart0_buf_rx[uart0_rx_ptr-1];
183         if (uart0_sum == uart0_check_sum)
184         {
185             uart0_ret = 1;
186             uart0_rx_iptr=0;
187         }
188         else
189         {
190             uart0_rx_iptr=0;
191             uart0_rx_ptr=0;
192             uart0_ret=2;
193         }
194         // Vynulování crc
195         uart0_check_sum=0;
196     }
197     else
198     {
199         uart0_ret = 0;
200     }
201     sei();
202     uart0_rx_flag=FALSE;
203     return uart0_ret;
204 }
205 }
206
207
208 void uart0_transmit_char(uint8_t data)
209 {
210
211 }
```