```
/***************************** C SOURCE FILE ********************************
**
**  Project:    UART1 Driver for CL2bm1 (UART1A)
**  Filename:   Uart1_dr1.c
**  Date:       21.02.2018
**  Version:    1.2
**  Modified    R.Oliva  - Include interrupt routines in C separate file (test)
**
*******************************************************************************
**
**
*******************************************************************************
**
**  VERSION HISTORY:
**  ----------------
**  initial Version:    1.1
**  Date:       21.02.2018
**  Revised by: R.Oliva
**  Description:
**      -
**      - Newer versions see top.
**
**
**
*******************************************************************************/

#include "../inc/Uart1_dr1.h"

/*******************************************************************************
**
** DEFINITIONS
**
*******************************************************************************/



/*******************************************************************************
**
** UART1 Global Variables declared in uart1_dr1.h
** MEMORY IS ASSIGNED
** HERE FOLLOWING RULE #5
**
*******************************************************************************/

char rx_buffer1[RX_BUFFER_SIZE1];

#if RX_BUFFER_SIZE1 <= 256
unsigned char rx_wr_index1,rx_rd_index1,rx_counter1;
#else
unsigned int rx_wr_index1,rx_rd_index1,rx_counter1;
#endif

// This flag is set on USART1 Receiver buffer overflow
bit rx_buffer_overflow1;

char tx_buffer1[TX_BUFFER_SIZE1];

#if TX_BUFFER_SIZE1 <= 256
unsigned char tx_wr_index1,tx_rd_index1,tx_counter1;
#else
unsigned int tx_wr_index1,tx_rd_index1,tx_counter1;
#endif

/*******************************************************************************
**
** EXPORTED FUNCTIONS
**
*******************************************************************************/

/*******************************************************************************
**
** Initializes the UART1 - version UART1A with parameter
**
** Parameters: uint8_t pbaud, can take values:
** #define PBAUD_9600  0
** #define PBAUD_19200 1
```

```
77      ** #define PBAUD_38400 2
78      ** Then for each option, considering CLK=14.7456E06 Hz
79      **  38400 -> u2x=0 ->UBRR=23dec = 0x17
80      **  19200 -> u2x=0 ->UBRR=47dec = 0x2F
81      **  9600  -> u2x=0 ->UBRR=95dec = 0x5F
82      **
83      ** Returns: NONE
84      **
85      *****************************************************************************/
86
87      // USART1_Init modified for pbaud parameter
88      void USART1_Init(unsigned char pbaud)
89      {
90        switch(pbaud){
91          case PBAUD_9600:
92              // USART1 initialization 9600 baud
93              // Communication Parameters: 8 Data, 1 Stop, No Parity
94              // USART1 Receiver: On
95              // USART1 Transmitter: On
96              // USART1 Mode: Asynchronous
97              // USART1 Baud Rate: 9600
98              UCSR1A=0x00;
99              UCSR1B=0xD8;
100             UCSR1C=0x06;
101             UBRR1H=0x00;
102             UBRR1L=0x5F; // ==95 dec for 9600, U2X=0
103             break;
104         case PBAUD_19200:
105             // USART1 initialization 19200
106             // Communication Parameters: 8 Data, 1 Stop, No Parity
107             // USART1 Receiver: On
108             // USART1 Transmitter: On
109             // USART1 Mode: Asynchronous
110             // USART1 Baud Rate: 19200
111             UCSR1A=0x00;
112             UCSR1B=0xD8;
113             UCSR1C=0x06;
114             UBRR1H=0x00;
115             UBRR1L=0x2F; // ==47 dec for 19200, U2X=0
116             break;
117         case PBAUD_38400:
118             // USART1 initialization 38400 baud (PWRC2 - V22.3.2012)
119             // Communication Parameters: 8 Data, 1 Stop, No Parity
120             // USART1 Receiver: On
121             // USART1 Transmitter: On
122             // USART1 Mode: Asynchronous
123             // USART1 Baud Rate: 38400
124             UCSR1A=0x00;
125             UCSR1B=0xD8;
126             UCSR1C=0x06;
127             UBRR1H=0x00;
128             UBRR1L=0x17; // ==23 dec for 38400, U2X=0
129             break;
130         default:
131             printf("\n\r Parametro COM1 Incorrecto! (9600, 19200 o 38400)");
132             break;
133         }
134     }
135
136     /*****************************************************************************
137     **
138     ** USART1 Receiver interrupt service routine
139     ** Buffer Size 256 not considered..30.1.18
140     *****************************************************************************/
141
142     interrupt [USART1_RXC] void usart1_rx_isr(void)
143     {
144     char status,data;
145     status=UCSR1A;
146     data=UDR1;
147     if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
148        {
149        rx_buffer1[rx_wr_index1++]=data;
150        // #if RX_BUFFER_SIZE1 == 256
151        // special case for receiver buffer size=256
152        // if (++rx_counter1 == 0)
```

```
153      //    {
154      // #else
155      if (rx_wr_index1 == RX_BUFFER_SIZE1) rx_wr_index1=0;
156      if (++rx_counter1 == RX_BUFFER_SIZE1)
157         {
158         rx_counter1=0;
159      //#endif
160         rx_buffer_overflow1=1;
161         }
162      }
163   }
164
165
166   /*****************************************************************************
167   **
168   ** Creates an Alternat Getchar1() function using the USART1 ISR
169   ** 30.1.18
170   *****************************************************************************/
171
172   // Get a character from the USART1 Receiver buffer
173   #pragma used+
174   char getchar1(void)
175   {
176   char data;
177   while (rx_counter1==0);
178   data=rx_buffer1[rx_rd_index1++];
179   #if RX_BUFFER_SIZE1 != 256
180   if (rx_rd_index1 == RX_BUFFER_SIZE1) rx_rd_index1=0;
181   #endif
182   #asm("cli")
183   --rx_counter1;
184   #asm("sei")
185   return data;
186   }
187   #pragma used-
188
189
190
191   /*****************************************************************************
192   **
193   ** USART1 Transmitter interrupt service routine
194   **
195   *****************************************************************************/
196
197   interrupt [USART1_TXC] void usart1_tx_isr(void)
198   {
199   if (tx_counter1)
200      {
201      --tx_counter1;
202      UDR1=tx_buffer1[tx_rd_index1++];
203   #if TX_BUFFER_SIZE1 != 256
204      if (tx_rd_index1 == TX_BUFFER_SIZE1) tx_rd_index1=0;
205   #endif
206      }
207   }
208   /*****************************************************************************
209   **
210   ** Creates an Alternate putchar1() function using the USART1 ISR
211   ** 30.1.18
212   *****************************************************************************/
213
214   // Write a character to the USART1 Transmitter buffer
215   #pragma used+
216   void putchar1(char c)
217   {
218   while (tx_counter1 == TX_BUFFER_SIZE1);
219   #asm("cli")
220   if (tx_counter1 || ((UCSR1A & DATA_REGISTER_EMPTY)==0))
221      {
222      tx_buffer1[tx_wr_index1++]=c;
223   #if TX_BUFFER_SIZE1 != 256
224      if (tx_wr_index1 == TX_BUFFER_SIZE1) tx_wr_index1=0;
225   #endif
226      ++tx_counter1;
227      }
228   else
```

File: C:\cvavr328\Work3\CL2\CL2_Drivers\UART1A\TST_MENU_PRJ\src\Uart1

```
229       UDR1=c;
230     #asm("sei")
231     }
232     #pragma used-
233
234
235
236     /*****************************************************************************
237     **
238     ** EOF
239     **
240     *****************************************************************************/
241
242
```