```
/***************************** C SOURCE FILE ********************************
**
**  Project:    UART0 Driver for CL2bm1
**  Filename:   Uart0_dr1.c
**  Version:    1.0
**  Date:       v17.12.2017
**  Modified    R.Oliva  - Include interrupt routines in C separate file (test)
**
*******************************************************************************
**
**
*******************************************************************************
**
**  VERSION HISTORY:
**  ----------------
**  initial Version:    1.0
**  Date:       17.12.2017
**  Revised by: R.Oliva
**  Description:
**      -
**      - Newer versions see top.
**
**
**
*******************************************************************************/

#include "../inc/Uart0_dr1.h"

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



/*******************************************************************************
**
** UART0 Global Variables declared in uart0_dr1.h
** MEMORY IS ASSIGNED
** HERE FOLLOWING RULE #5
**
*******************************************************************************/

char rx_buffer0[RX_BUFFER_SIZE0];

#if RX_BUFFER_SIZE0 <= 256
unsigned char rx_wr_index0,rx_rd_index0,rx_counter0;
#else
unsigned int rx_wr_index0,rx_rd_index0,rx_counter0;
#endif

// This flag is set on USART0 Receiver buffer overflow
bit rx_buffer_overflow0;


char tx_buffer0[TX_BUFFER_SIZE0];

#if TX_BUFFER_SIZE0 <= 256
unsigned char tx_wr_index0,tx_rd_index0,tx_counter0;
#else
unsigned int tx_wr_index0,tx_rd_index0,tx_counter0;
#endif



/*******************************************************************************
```

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

/*****************************************************************************
**
** Initializes the UART0
**
** Parameters: NONE
**
** Returns: NONE
**
*****************************************************************************/

// USART0_Init standard 19200,N,8,1 TxRx ISR support
void USART0_Init(void)
{
  // USART0 initialization - PWRC2
  // Communication Parameters: 8 Data, 1 Stop, No Parity
  // USART0 Receiver: On
  // USART0 Transmitter: On
  // USART0 Mode: Asynchronous
  // USART0 Baud Rate: 19200!
  UCSR0A=0x00;
  UCSR0B=0xD8;
  UCSR0C=0x06;
  UBRR0H=0x00;
  UBRR0L=0x2F;
}

/*****************************************************************************
**
** USART0 Receiver interrupt service routine
** Buffer Size 256 not considered..18.12.2017
*****************************************************************************/

interrupt [USART0_RXC] void usart0_rx_isr(void)
{
 char status,data;
 status=UCSR0A;
 data=UDR0;
 if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
  {
   rx_buffer0[rx_wr_index0++]=data;
   //#if RX_BUFFER_SIZE0 == 256     (commented out 18.12.2017)
   // special case for receiver buffer size=256
   // if (++rx_counter0 == 0)
   //   {
   //#else
   if (rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
   if (++rx_counter0 == RX_BUFFER_SIZE0)
      {
      rx_counter0=0;
   //#endif
      rx_buffer_overflow0=1;
      }
   }
  }


/*****************************************************************************
**
** Creates an Alternat Getchar() function using the USART0 ISR
** 18.12.2017
*****************************************************************************/
```

```
// Internal CVAVR compiler commands:
#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART0 Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter0==0);
data=rx_buffer0[rx_rd_index0++];
#if RX_BUFFER_SIZE0 != 256
if (rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
#endif
#asm("cli")
--rx_counter0;
#asm("sei")
return data;
}
#pragma used-
#endif

// ***************************************************************
// ** GetByte() Added for Modbus Inputs - transfer Bytes not chars..**
// ** v1.0 30-05-2012 - Used by MB_Serial() FromModbusTest2()       **
// ***************************************************************
unsigned char GetByte(void)
{
unsigned char data;
while (rx_counter0==0);
data=rx_buffer0[rx_rd_index0++];
#if RX_BUFFER_SIZE0 != 256
if (rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
#endif
#asm("cli")
--rx_counter0;
#asm("sei")
return data;
}


/*****************************************************************************
**
** USART0 Transmitter interrupt service routine
**
*****************************************************************************/

interrupt [USART0_TXC] void usart0_tx_isr(void)
{
if (tx_counter0)
   {
   --tx_counter0;
   UDR0=tx_buffer0[tx_rd_index0++];
#if TX_BUFFER_SIZE0 != 256
   if (tx_rd_index0 == TX_BUFFER_SIZE0) tx_rd_index0=0;
#endif
   }
}

/*****************************************************************************
**
** Creates an Alternate putchar() function using the USART0 ISR
** 18.12.2017
*****************************************************************************/

// Internal CVAVR compiler commands:
#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART0 Transmitter buffer
```

```
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter0 == TX_BUFFER_SIZE0);
#asm("cli")
if (tx_counter0 || ((UCSR0A & DATA_REGISTER_EMPTY)==0))
   {
   tx_buffer0[tx_wr_index0++]=c;
#if TX_BUFFER_SIZE0 != 256
   if (tx_wr_index0 == TX_BUFFER_SIZE0) tx_wr_index0=0;
#endif
   ++tx_counter0;
   }
else
   UDR0=c;
#asm("sei")
}
#pragma used-
#endif

// ***************************************************************
// ** PutByte() Added for Modbus output - transfer Bytes not chars..**
// ** v1.0 30-05-2012 - Used by FinaliseTransmit and ExceptionResp()**
// ***************************************************************
void PutByte(unsigned char txbyte)
{
while (tx_counter0 == TX_BUFFER_SIZE0);
#asm("cli")
if (tx_counter0 || ((UCSR0A & DATA_REGISTER_EMPTY)==0))
   {
   tx_buffer0[tx_wr_index0++]=txbyte;
#if TX_BUFFER_SIZE0 != 256
   if (tx_wr_index0 == TX_BUFFER_SIZE0) tx_wr_index0=0;
#endif
   ++tx_counter0;
   }
else
   UDR0=txbyte;
#asm("sei")
}



/*****************************************************************************
**
** EOF
**
*****************************************************************************/
```