```
#include <asf.h>
#define F_CPU 8000000L                    // set the CPU clock
#include <avr/io.h>
#include <util/delay.h>

#define BAUD 9600                           // define baud
#define BAUD_PRESCALLER 51    // set baudrate value for UBRR
#define _ASSERT_ENABLE_
#include <string.h>
#include "compiler.h"
#include "comms.h"

/**
 * \def BUFFER_SIZE
 * \brief The size of the UART buffer
 */
#define BUFFER_SIZE 100



#include "ring_buffer.h"

//buffers for use with the ring buffer (belong to the UART)

//buffer for sending characters
uint8_t out_buffer[BUFFER_SIZE];

//buffer for receiving characters
uint8_t in_buffer[BUFFER_SIZE];

// the string we send and receive on UART
// This strings are used for testing
const char test_string[] = "Received:[";
const char test_string2[] = "]\n";
const char test_string3[] = "LALALALALAL";



extern void uart_init(void)
{
    //Load upper 8-bits of the baud rate value into the high byte of the UBRR0H register
    UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8);

    //Load upper 8-bits of the baud rate value into the high byte of the UBRR0L register
    UBRR0L = (uint8_t)(BAUD_PRESCALLER);

     // Turn on the transmission and reception circuitry
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);

    UCSR0C = (3<<UCSZ00)|(1<<USBS0);

    // enable RX and TX and set interrupts on rx complete
    UCSR0B = (1 << RXEN0) | (1 << TXEN0) | (1 << RXCIE0);

    // 8-bit, 1 stop bit, no parity, asynchronous UART
    UCSR0C = (1 << UCSZ01) | (1 << UCSZ00) | (0 << USBS0) |
            (0 << UPM01) | (0 << UPM00) | (0 << UMSEL01) |
            (0 << UMSEL00);

}

/**
 * \brief Function for putting a char in the UART buffer
 *
 * \param data the data to add to the UART buffer and send
 *
 */
static inline void uart_putchar(uint8_t data)
{
        while(!(UCSR0A & (1<<UDRE0)));
        UDR0 = data;
}

/**
 * \brief Function for getting a char from the UART receive buffer
 *
 * \retval Next data byte in receive buffer
 */
static inline uint8_t uart_getchar(void)
```

```c
{
        // Do nothing until data have been received and is ready to be read from UDR
        while(!(UCSR0A & (1<<RXC0)));

        // return the received character
        return UDR0;
}


extern void send_confirmation_msg(uint8_t data)
{
    if((data > 32) && (data < 126)) {
        send_str(test_string);
        uart_putchar(data);
        send_str(test_string2);
    }
}

extern void send_str(char* StringPtr)
{
        //loop over all characters
        while(*StringPtr != 0x00) {
            //Put the character in the buffer to send it
            uart_putchar(*StringPtr);
            //increment the pointer
            StringPtr++;
        }
    //Send return character after sending the string
    uart_putchar('\r');

}

extern uint8_t get_char(void)
{
    //Declare data character
    uint8_t data = '\0';
    //Get the received character
    data = uart_getchar();
    //Return the received character
    return data;
}
```