Micromint, Inc	**AN700**
	Micro64/128
Using the RS485/RS422/RS232A Serial Port	11/3/04

Introduction: This application note demonstrates how to use the RS485/RS422/RS232A serial port (USART1) for the Micro64/128 using the CodeVision AVR C compiler.

Background: Micro64/128 can communicate with other serial devices at up to 230.4 kbps. Usart1 can be connected in one of three configurations: RS-232A, RS422, and RS-485. Please see the appendix of the Micro64/128 data sheet for further details.

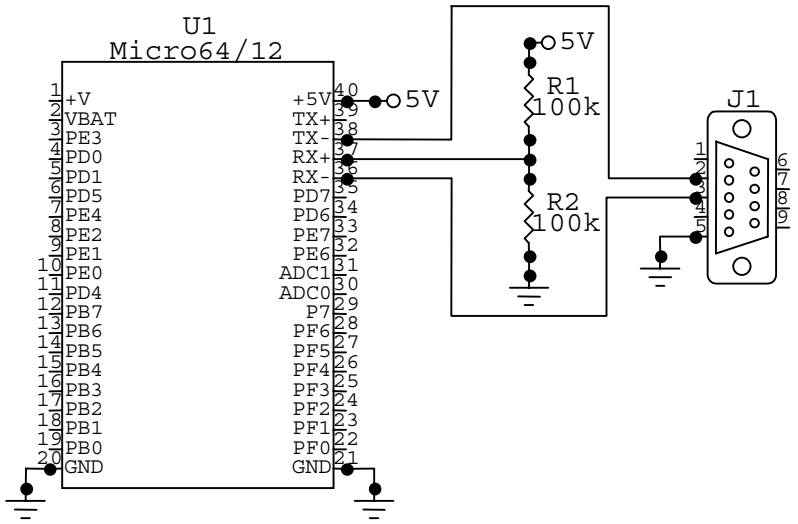

Figure 1: RS-232 Connection

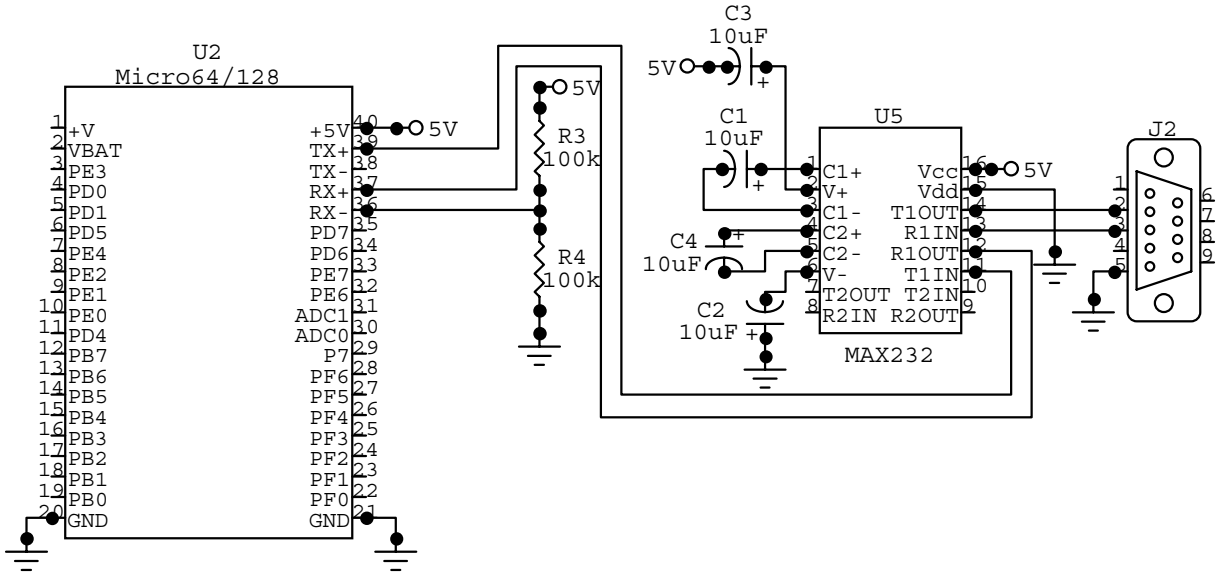

Figure 2: RS-232C Connection

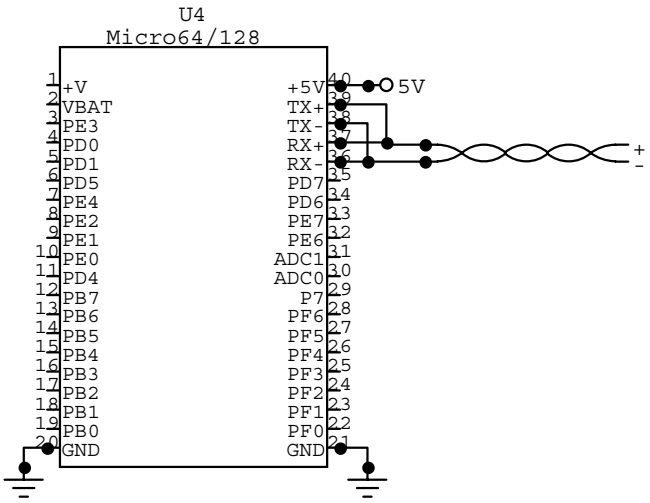

Figure 3: RS-485 Connection

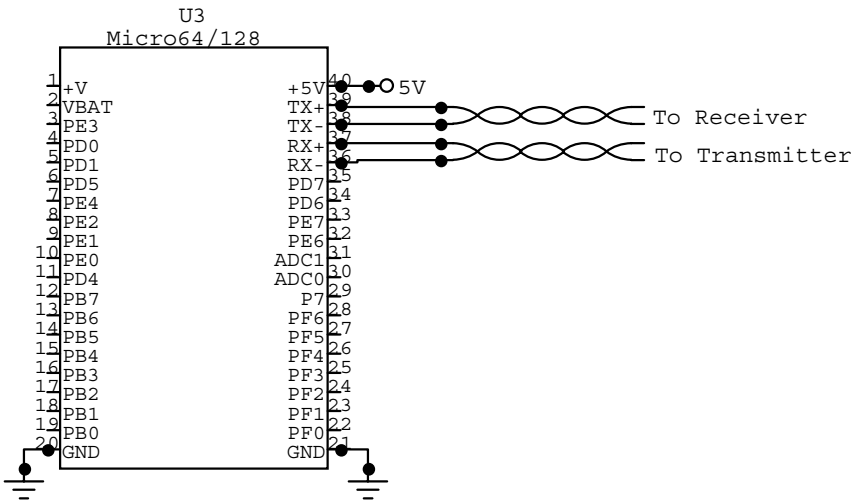

Figure 4: RS-422 Connection

How it works: Micromint, Inc. has written an alternate putchar and getchar library called MMRS485.lib. The library allows the user to swap between USART0 and USART1 by setting an integer variable called COM to 0 or 1. If COM is set to 0 then putchar and getchar will be used for USART0. If it is set to 1 then putchar and getchar will be used with USART1. All the user has to do is include the MMRS485.h file and declare the integer variable called COM at the beginning of the their program. The program listing demonstrates this. If the user is going to use USART1 then they must also make PORTD.6 an output and make it high in order to enable the transmitter and low to disable the transmitter. If the Micro64/128 is not on an RS-485 network then the user might want to just leave the transmitter enabled. The program in the Program Listing section enables the transmitter before it transmits and then disables it after it is done transmitting.

Program Listing:

```
/*****************************************
Program : USART1 (RS485 PORT) Example for Micro64
Company : Micromint, Inc
*****************************************/

#include <mega64.h>
#include <MMRS485.h>     // Micromints Library for using both USARTs
#include <stdio.h>       // Standard I/O library
#include <delay.h>       // Library for delays
#include <bcd.h>

// Declare your global variables here
int COM;         // if COM = 0 then use USART0 if it = 1 then use USART1
unsigned int x @0xFFE;
```

```
void(*UtilityVersion)(void)=0x7C19;

void main(void)
{
// Declare your local variables here

// Set up USART1's Baud rate at 9600 bps with a 11.0592 MHz Crystal
UCSR1A=0x00;      // RX EN, TX EN
UCSR1B=0x18;     // RX EN, TX EN
UCSR1C=0x06;     // 8N1
UBRR1H=0x00;    // Baud rate high - 9600
UBRR1L=0x47;   // Baud rate low


while (1)
     {
     COM = 1;                        // Use USART1
     DDRD.6 = 0;                                    // Make PORTD.6 an output
     PORTD.6 = 1;                                   // Enable the RS485 control line
     (*UtilityVersion)();
     printf("Utility Version = %02d\r\n",bcd2bin(x));
     delay_ms(500);          //Wait for th transmission to finish
     PORTD.6 = 0;                                   // Disable the RS485 control line
     };
}
```