Migrating from RS-232 to USB Bridge Specification

USB Microcontrollers

Application Note

Doc Control

Rev	Purpose of Modifications	Date
0.0	Creation date	24 Nov 2003
1.0	updates	22 Dec 2003

References

- Universal Serial Bus Specification, revision 2.0
- Universal Serial Bus Class Definition for Communication Devices, version 1.1
- USB CDC demo firmware

Abbreviations

- USB: Universal Serial Bus
- CDC: Communication Device Class
- ACM: Abstract Control Model
- VID: Vendor Identifier
- PID: Product Identifier

Rev. 4322A–USB–01/04

Introduction

Scope

In the PC world, the RS-232 COM port is about to disappear from most computers (especially from laptops) in the future, replaced by the USB connection. However, many applications still use the RS-232 standard.

One solution can be to use the USB bus as it is an RS-232. The advantages are:

- no need to change the PC application
- few hardware and software embedded modifications

This document has two main objectives.

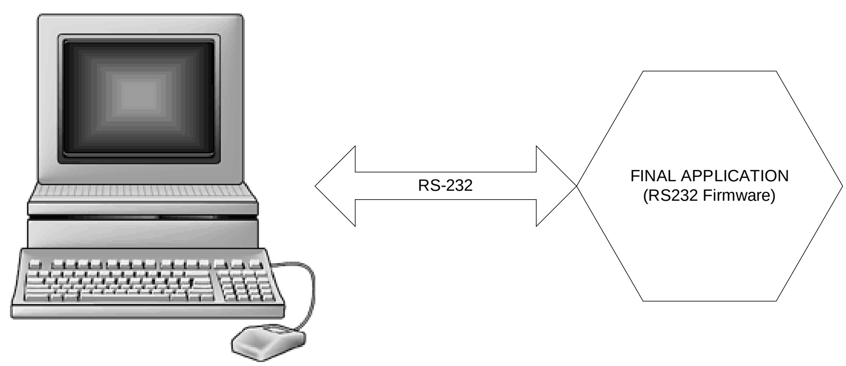
The first one is to describe how to migrate from a RS-232 to the USB into the firmware using the Atmel USB RS232 Virtual COM port library.

The second one is to describe how to build a USB <-> RS-232 bridge, using the Atmel library.

Overview

Many applications use the RS-232 communication port to communicate between the computer and the embedded application.

Figure 1. Initial Application



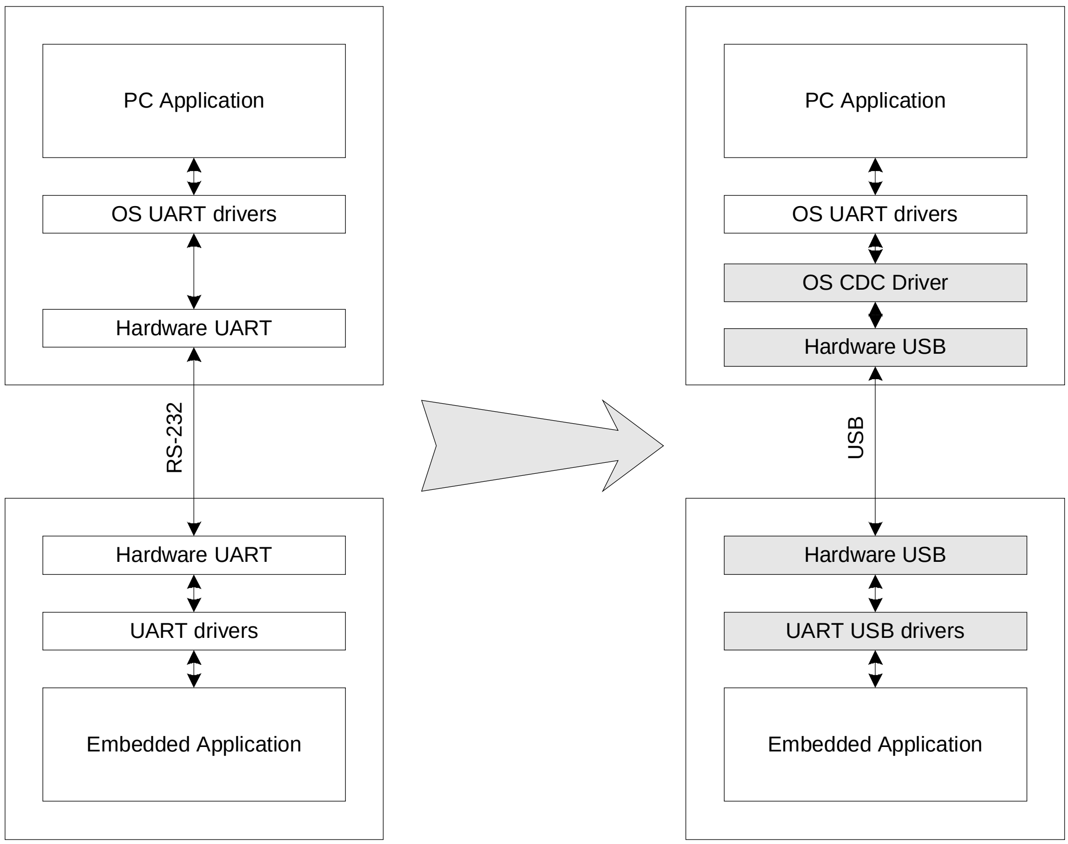
Switching to the USB port offers many advantages:

- USB is present on every new computer, not RS-232
- the final application can use the flexibility of the USB: data buffering, no data lost, automatic flow control, etc.
- USB link offers a power supply for the application.

Many advantages are available in switching from RS-232 to USB. But the aim is also to perform this switch with a minimum time and effort.

Software Architecture

The aim of a Virtual COM port implementation is to move from a real UART system to a UART-USB system:



This architecture induces very few modifications.

From the PC side, nothing to change! The USB device is seen as a COM port and the application doesn't need to be changed.

From the embedded side, the UART driver is replaced by the UART USB drivers. The embedded application calls UART-USB functions instead of UART functions.

USB CDC Class

The USB Communication Device Class document describes a way to implement devices such as ISDN modems, virtual COM ports, etc. Refer to this document for more details (http:\\www.usb.org).

In order to be considered a COM port, the USB device declares 2 interfaces:

- Abstract Control Model Communication, with 1 Interrupt IN endpoint
- Abstract Control Model Data, with 1 Bulk IN and 1 Bulk OUT endpoint

Loading the CDC Driver under Windows®

The CDC class is implemented in all releases of Windows, from Windows 98SE to Windows XP.

When plugging a new USB device, Windows checks all its INF files to load the appropriate driver. The INF file contains the Vendor ID, the Product ID or the USB Class definition. If the VID/PID or the USB Class Definition of the USB device matches with one INF file, Windows will load the driver described in this file.

As Microsoft does not provide standard INF file for the CDC driver, Atmel provides an INF file that allows to load this driver under Windows 2000 and Windows XP. When plugged for the first time, the user indicates to the Operating System which driver to use by selecting this INF file.

The application manufacturer has to use its own VID and PID. It has to modify these values in the embedded application (config.h) AND the INF file given in the example below.

Example of INF file

```
; Windows 2000 and XP setup File for AT89C5131 demo

[Version]
Signature="$Windows NT$"
Class=Ports
ClassGuid={4D36E978-E325-11CE-BFC1-08002BE10318}

Provider=%ATMEL%
LayoutFile=layout.inf
DriverVer=10/15/1999,5.0.2153.1

[Manufacturer]
%ATMEL%=ATMEL

[ATMEL]
%ATMEL_CDC%=Reader, USB\VID_03EB&PID_2009


[Reader_Install.NTx86]
;Windows2000

[DestinationDirs]
DefaultDestDir=12
Reader.NT.Copy=12

[Reader.NT]
CopyFiles=Reader.NT.Copy
AddReg=Reader.NT.AddReg

[Reader.NT.Copy]
usbser.sys

[Reader.NT.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,usbser.sys
```

```
HKR,,EnumPropPages32,,"MsPorts.dll,SerialPortPropPageProvider"

[Reader.NT.Services]
AddService = usbser, 0x00000002, Service_Inst

[Service_Inst]
DisplayName = %Serial.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 3 ; SERVICE_DEMAND_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %12%\usbser.sys
LoadOrderGroup = Base

[Strings]
ATMEL = "ATMEL, Inc."
ATMEL_CDC = "AT89C5131 CDC USB to UART"
Serial.SvcDesc = "USB Serial emulation driver"
```

UART USB Library

Elementary functions

uart_usb_test_hit()

```
bit uart_usb_test_hit (void);
```

This function checks if at least one character has been received.

uart_usb_getchar()

```
char uart_usb_getchar (void);
```

This function returns the byte received in the OUT Endpoint FIFO if there is at least one byte in the FIFO. If there is no byte in the stack, this function waits for a new USB receipt.

uart_usb_putchar()

```
char uart_usb_putchar (char);
```

This function writes the byte put in parameter into the USB IN Endpoint FIFO. If this Endpoint is full, this function sends the entire FIFO to the Host.

uart_usb_tx_ready()

```
bit uart_usb_tx_ready (void);
```

This function checks if the firmware can write a byte in the IN Endpoint FIFO.

uart_usb_flush()

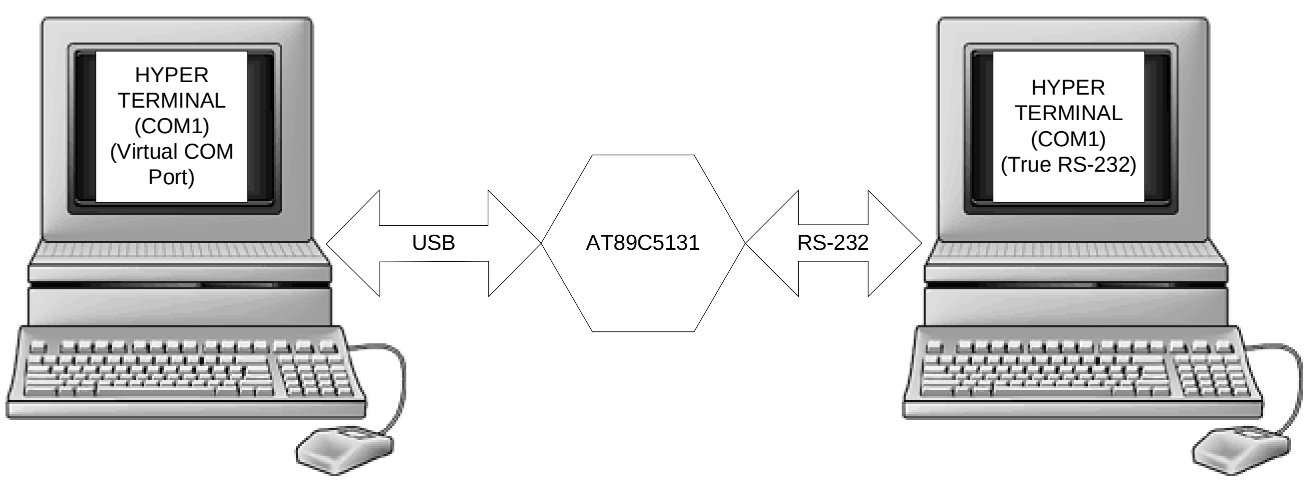
```
bit uart_usb_flush (void);
```

If there is data stored into the IN Endpoint FIFO, this function sends them.

This prevents waiting for the IN Endpoint FIFO to be full before it is sent to the Host.

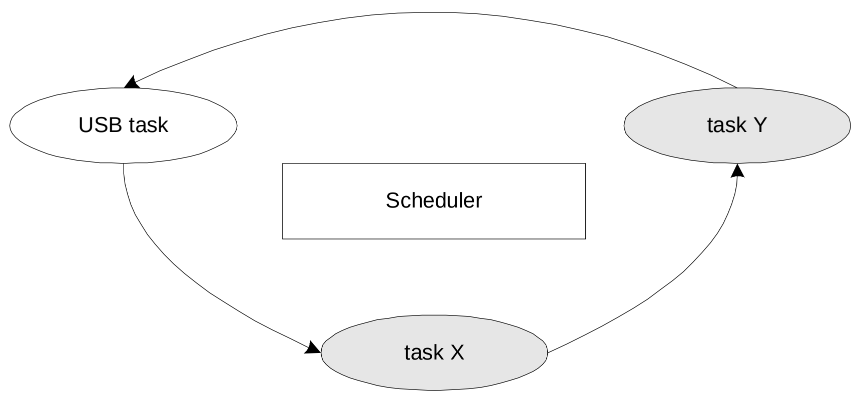
Example of use: USB to RS232 bridge

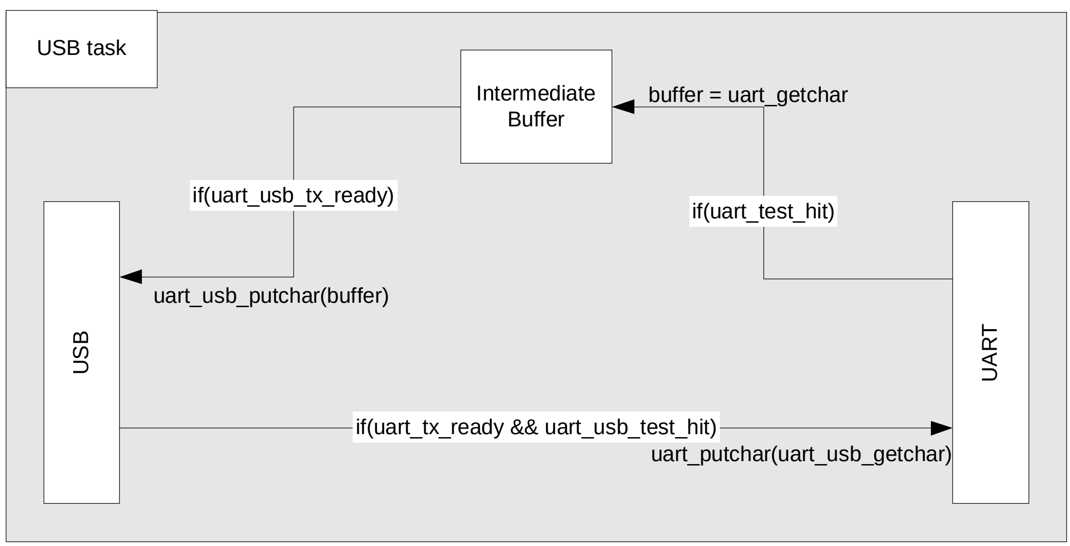
The aim of this example is to implement a very simple USB <-> RS-232 bridge:



Note: the UART configuration can't be changed in this example.

Embedded firmware architecture:





In this implementation, all the received characters from UART are stored in an intermediate buffer in order to avoid data lost when the USB is not able to handle the data.

```
if (uart_test_hit())
  { buff[uart_usb_index++]=uart_getchar();
    if (uart_usb_tx_ready())
    { for (j=0;j<uart_usb_index;j++) {uart_usb_putchar (buff[j]); }
    uart_usb_index=0;
    }
  }
if (Uart_tx_ready())
  { if (uart_usb_test_hit()) { uart_putchar(uart_usb_getchar()); } }
```

Example of use: Hello World

When the HELLO_WORLD_DEMO is defined in the config.h file, the standard STDIO functions are mapped on the USB instead of UART. The firmware can directly use the “printf()” function for example to write on the USB port.

The “HELLO WORLD” message is continuously sent when pressing on the INT0 button of the AT89C5131 demoboard.

All character typed and sent to the AT89C5131 through the Virtual COM port is echoed.

```
if (P3_2 == 0) { printf ("Hello World\r\n"); }
if (test_hit()) { putchar (_getkey()); }
```

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

4322A–USB–01/04 /xM