

# AVR272: USB CDC Demonstration UART to USB Bridge

## Features

- Supported by Windows® 2000 or later
- No driver installation
- Virtual COM Port Enumeration
- USB to RS232 Bridge with dynamic baudrate
- Bus powered

## 1. Introduction

The RS232 interface has disappeared from the new generation of PCs replaced by the USB interface. To follow this change, applications based on UART interface have to migrate to USB. Migration to USB can mean heavy development both on the PC and on the device side. To avoid this development, Atmel offers you a solution based on the CDC class (Communication Device Class) with the following advantages:

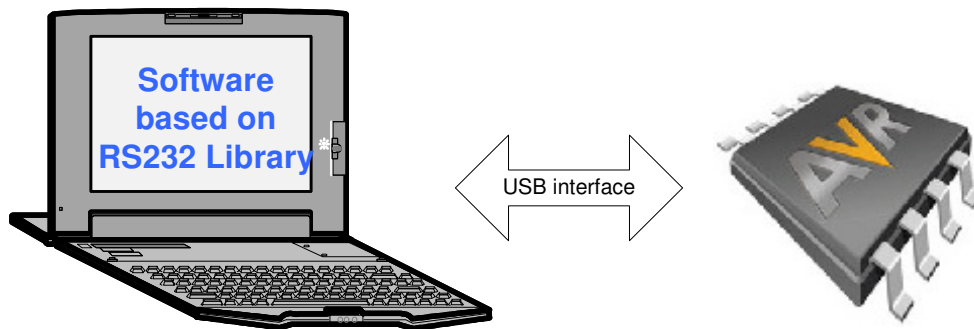
No need to change the PC application

Few modifications from the device side

The aim of this document is to describe how to start and implement a CDC (Virtual COM Port and UART to USB bridge) application using the STK525 starter kit and FLIP in-system programming software.

A familiarity with *USB Software Library for AT90USBxxx Microcontrollers* (Doc 7675, Included in the CD-ROM & Atmel website) and the CDC specification (<http://www.usb.org>) is assumed.

Figure 1-1. Virtual COM Port Application



8-bit AVR®  
Microcontrollers

Application Note

## 2. Hardware Requirements

The USB CDC application requires the following hardware:

1. AVR USB evaluation board (STK525, AT90USBKey, STK526...or your own board)
2. AVR USB microcontroller
3. USB cable (Standard A to Mini B)
4. RS232 crossed Cable (DB9 male to DB9 female)
5. PC running on Windows® (2000, XP) with USB 1.1 or 2.0 host

Note: An additional STK 52x and USB port are required if the PC has no RS232 interface.

## 3. In system programming and Device Firmware Upgrade

To program the device you can use the following methods:

- The JTAG interface using the JTAGICE MKII
- The SPI interface using the AVRISP MKII
- The USB interface thanks to the factory DFU bootloader and Flip software
- The parallel programming using the STK500 or the STK600

Please refer to the hardware user guide of the board you are using (if you are using Atmel starter kit) to see how to program the device using these different methods.

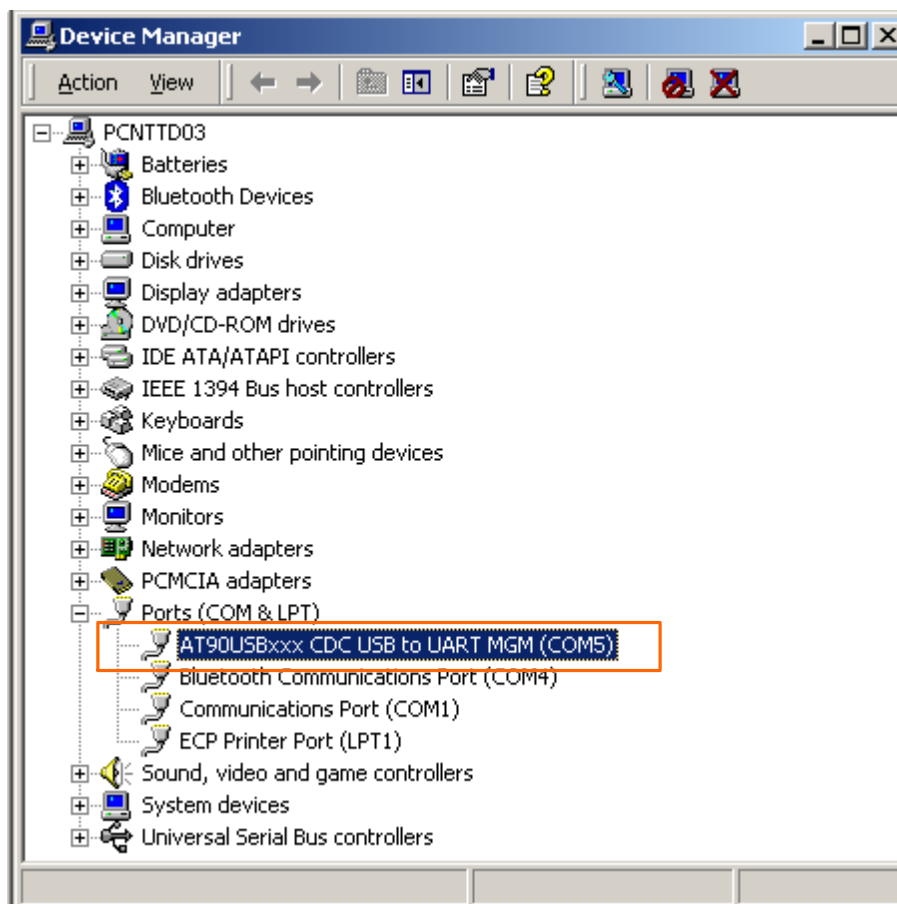
Please refer to Flip<sup>(1)</sup> help content to see how to install the USB driver and program the device through the USB interface.

Note: 1. Flip is a software provided by atmel to allow the user to program the AVR USB devices through the USB interface (No external hardware required) thanks to the factory DFU bootloader.

## 4. Quick Start

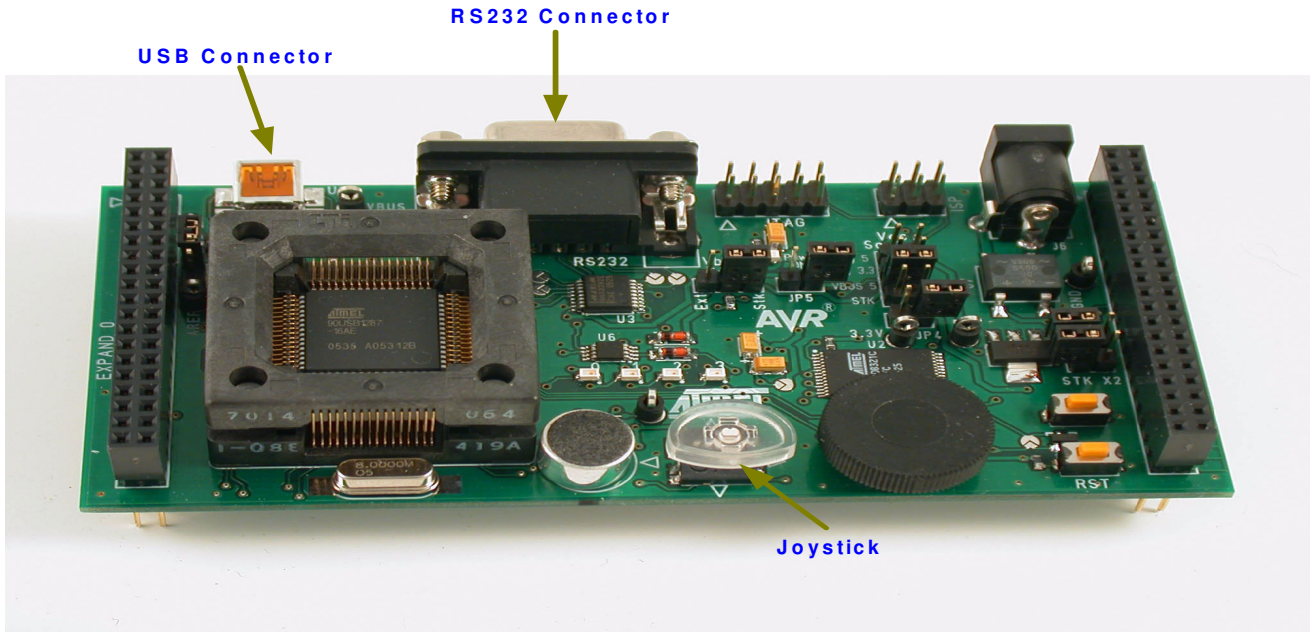
Once your device is programmed with `usb_cdc.a90` file, click on Start Application button on FLIP or push the reset button from the STK525 board to start the CDC demonstration. **A new device detection wizard will appear, point the wizard to the inf folder included in the CDC package.** Check that your device is enumerated as COM port (see [Figure 4-1](#)), then you can use the STK525 as a Virtual COM Port or USB to UART bridge.

Figure 4-1. CDC enumeration



The figure below shows the STK525 used by the demo (you may use another kit: AT90USBKey, STK526, depending on the AVR USB product you are working with):

**Figure 4-2.** STK525



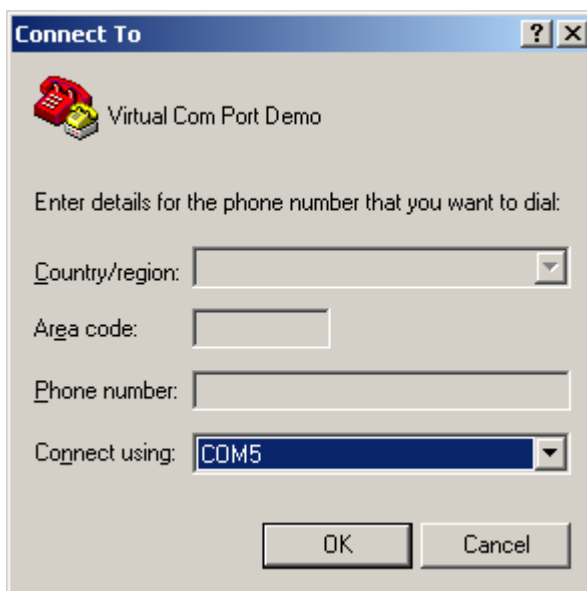
## 4.1 Virtual COM Port Demo

The purpose of the Virtual COM Port demonstration is to communicate with a RS232 PC application without any software modification.

Follow the instructions below to start the demo:

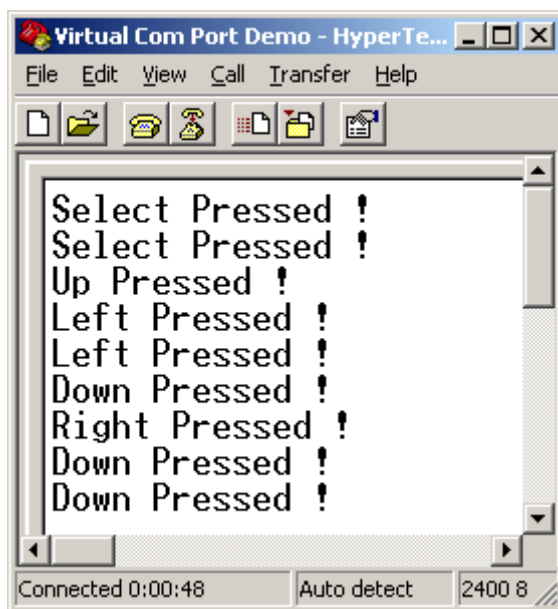
1. Launch the HyperTerminal application and select the correct COM port as indicated in the device manager.

**Figure 4-3.** COM Port selection



2. Press the joystick and you will see the selection or the direction status written on the HyperTerminal window.

**Figure 4-4.** Virtual COM Port Demo



## 4.2 USB to UART Bridge

The aim of the USB to UART bridge is to transfer data in full duplex mode between UART and USB interface. This application can be used to connect any RS232 device to a PC which does not have an RS232 interface.

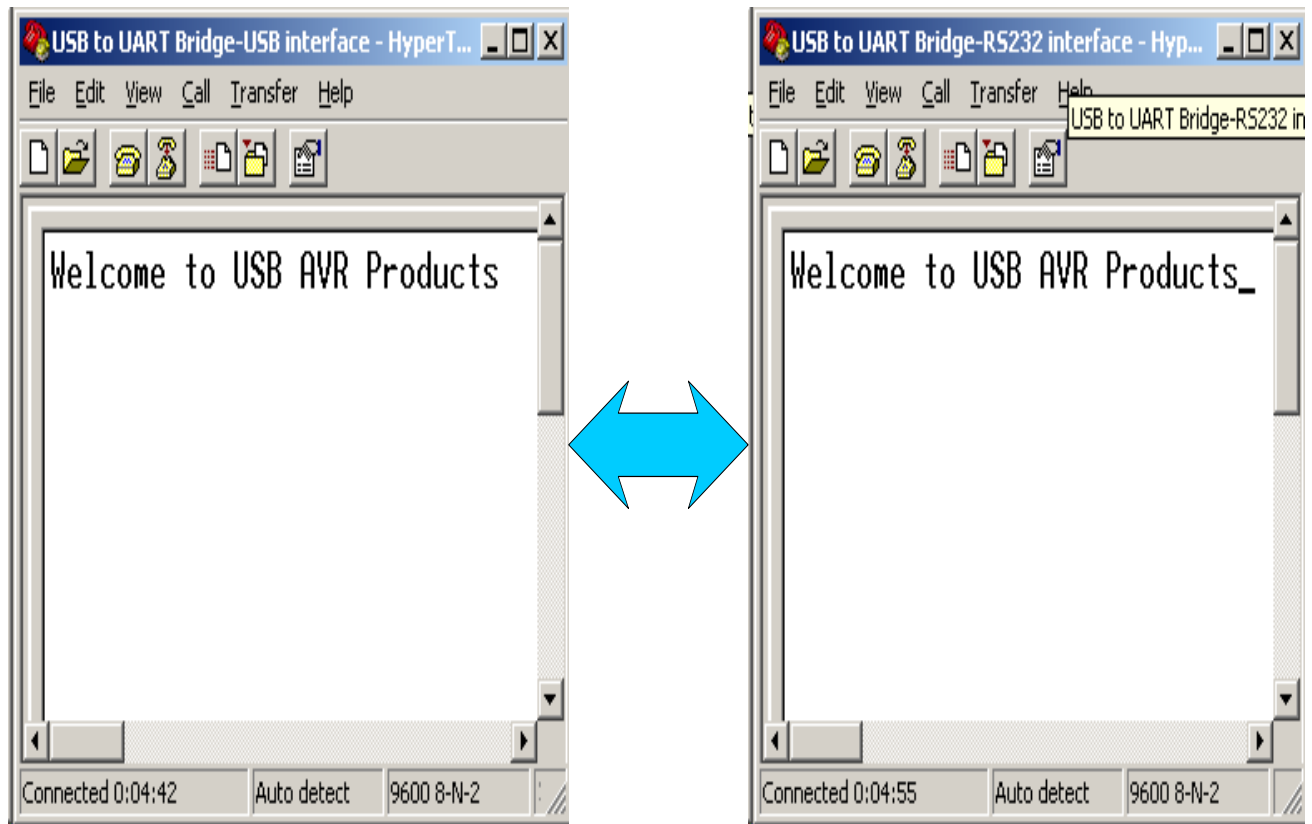
Follow the instructions hereunder to start the demo:

1. Connect the RS232 port of the STK 525 to The PC RS232 port.

Note: If the PC has no RS232 interface you can use another STK525: Connect the two boards with a RS232 crossed cable and connect each board to a USB port on the PC.

2. Launch two HyperTerminal applications (one with the RS232 port and the second with the Virtual COM port) with the same configuration (Baudrate, Data bits, Parity, Stop bits, Flow control).
3. Write something in one HyperTerminal, it will be displayed in the other.

**Figure 4-5.** USB to UART Bridge



## 5. Application Overview

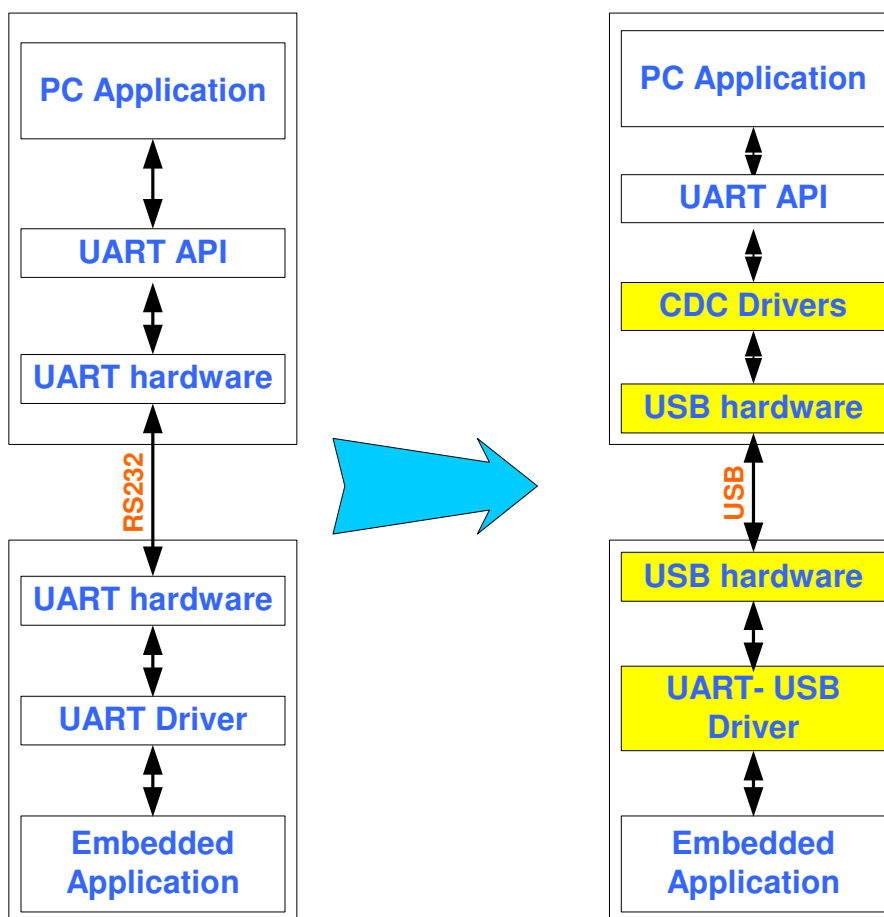
The CDC application allows the user to simulate an RS232 port using the USB hardware. The device appears as a COM port instead of USB device in the device manager. This allows the user to use his RS232 application without changing his PC application.

From the embedded side, the UART driver is replaced by the UART-USB Driver. The user has to use the UART-USB functions instead of the UART functions to communicate with the PC.

Once the device has enumerated, the application ensures a full duplex data transfer between the PC and the peripheral.

The figure below shows the structure:

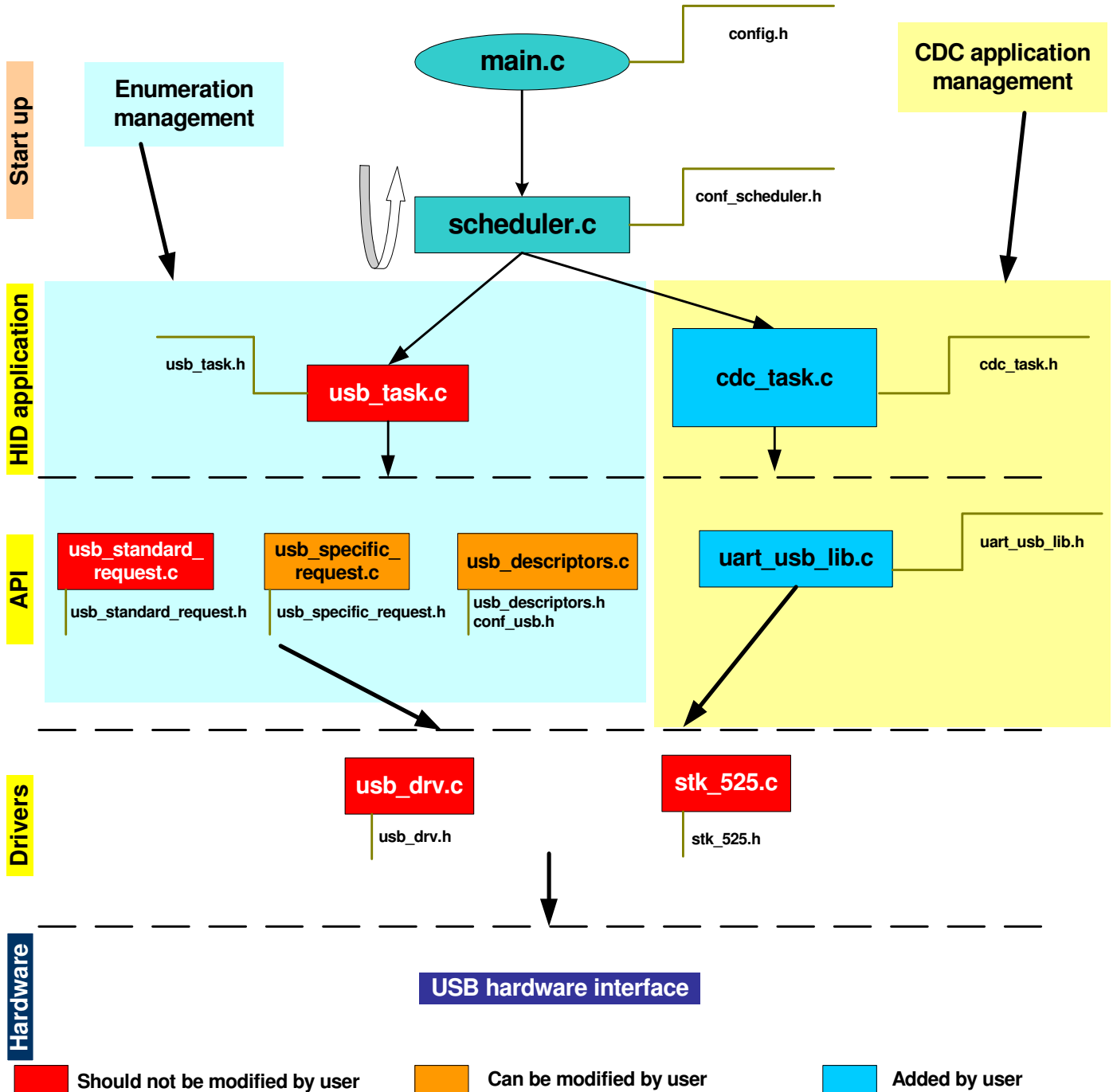
**Figure 5-1.** USB to UART migration



## 6. Firmware

As explained in the *USB Software Library for AT90USBxxx Microcontrollers*, all USB firmware packages are based on the same architecture (please refer to this document for more details).

Figure 6-1. USB CDC Firmware Architecture



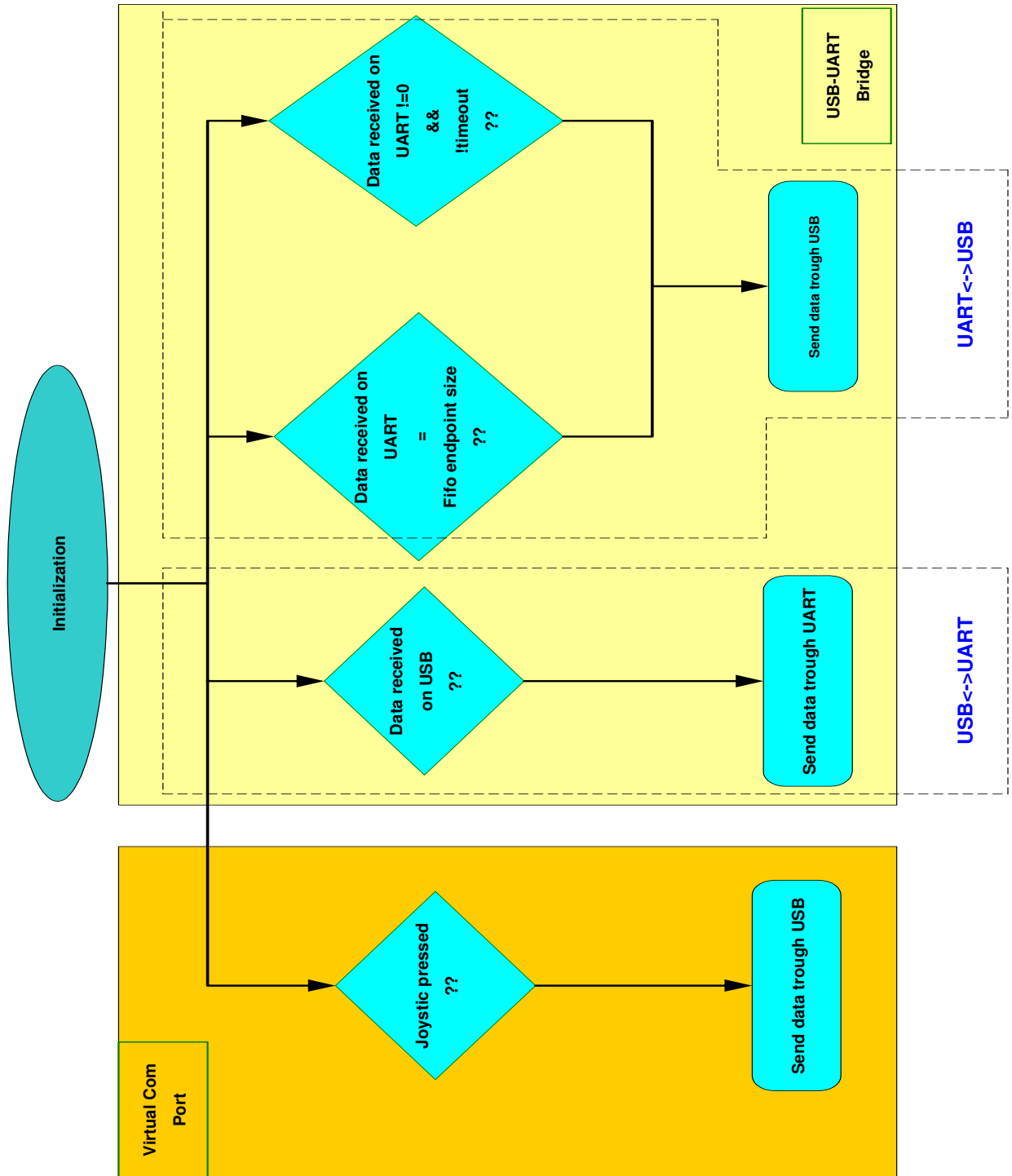
This section is dedicated to the CDC module only. The customization of the files described hereafter allow the user to build his own CDC Application.



6.1 cdc\_task.c

This file contains the functions to initialize the hardware which will be used by the application, collect data and transfer it.

Figure 6-2. CDC Application



### 6.1.1 `cdc_task_init`

This function performs the initialization of the CDC parameters and hardware resources (joystick...).

### 6.1.2 `cdc_task`

This function manages the data transfer for the two demonstrations (Virtual COM Port and UART to USB Bridge).

## 6.2 `uart_usb_lib`

### 6.2.1 `uart_usb_test_hit`

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

### 6.2.2 `uart_usb_getchar`

This function returns the byte received in the OUT endpoint FIFO.

### 6.2.3 `uart_usb_putchar`

This function writes the byte put in parameter into the USB IN endpoint FIFO. It also replaces the `putchar` function of the UART library. For example the `printf` will be based on `uart_usb_putchar` function instead of `putchar`.

### 6.2.4 `uart_usb_tx_ready`

This function checks if a byte can be written in the IN endpoint FIFO.

### 6.2.5 `uart_usb_flush`

This function sends data stored in the IN endpoint.

## 6.3 `stk_525.c`.

This file contains all the routines to manage the STK 525 board resources (joystick, potentiometer, temperature sensor, LEDs...).

## 7. PC Software

The CDC application uses the native Windows drivers. It requires only an INF file located in the `inf` folder from the CDC package.

## 8. Limitations

This application does not work with Windows 98 and ME (no native driver of CDC device).

This application can work with Linux OS, but support depends on configuration.

## 9. Related Documentation

- AVR USB Datasheet
- USB Software Library for AT90USBxxx Microcontrollers
- USB CDC class specification



## Headquarters

---

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

## International

---

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

**Atmel Europe**  
Le Krebs  
8, Rue Jean-Pierre Timbaud  
BP 309  
78054 Saint-Quentin-en-  
Yvelines Cedex  
France  
Tel: (33) 1-30-60-70-00  
Fax: (33) 1-30-60-71-11

**Atmel 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

## Product Contact

---

**Web Site**  
[www.atmel.com](http://www.atmel.com)

**Technical Support**  
[avr@atmel.com](mailto:avr@atmel.com)

**Sales Contact**  
[www.atmel.com/contacts](http://www.atmel.com/contacts)

**Literature Requests**  
[www.atmel.com/literature](http://www.atmel.com/literature)

---

**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.