AMS Lab Exercise 8
"Applied USB"

Purpose

To study USB and do experiments with practical USB interface for microcontrollers.

The focus will be to have "Hands On" some relatively simple methods for USB connectivity and AVR microcontroller – rather than to understand the details of USB.

After each demo, spend time to investigate USB further (by reading).

Literature

- The IT-AMS "Applied USB" lesson.
- USB in a Nutshell: http://www.beyondlogic.org/usbnutshell/usb1.shtml
- FTDI Chip: http://www.ftdichip.com
- V-USB: http://www.obdev.at/products/vusb/index.html
- V-USB Wiki: vusb.wikidot.com
- Demo program "HIDKeys_HH" (zipped at Blackboard).

Exercise, Part 1

This is to demonstrate the very simple way of USB enabling, by using a USB bridge hardware unit. In this case we will demonstrate the use of a USB-UART bridge from FTDI Chip (integrated in a cable). The UART signals will be TTL level signals, enabling us to connect directly to the Mega32 UART pins. The FTDI cable type is called TTL-232R-5V-WE (datasheet is available at Blackboard).

Start by studying the TTL-232R-5V-WE datasheet for specifications etc.
Notice the embedded USB-UART chip called FT232RQ. This chip is often used in embedded systems to implement a USB UART interface. Another advantage using USB is the option of powering the embedded system from the USB bus (removing the need for local power supply).

Test the USB bridge cable, by connecting it to the UART pins of Mega32 (at STK500) and the USB connected to the PC. Also remember to connect GND (optionally 5 volt).

Open "Device Manager" to setup Baud Rate etc.
Notice the maximum Baud Rate that the device is able to handle (close to 1 Mbit/s).

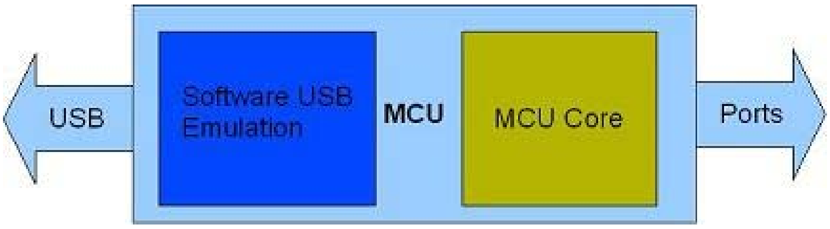
Test UART communication by using a terminal tool (for example "Tera Term") and a simple test program running at Mega32.

Important: When connecting TX and RX, remember to connect TX(Mega32) to RX(cable) and RX(Mega32) to TX(cable)!

End up by investigating FTDI's web page to see the many other USB solutions available:
http://www.ftdichip.com

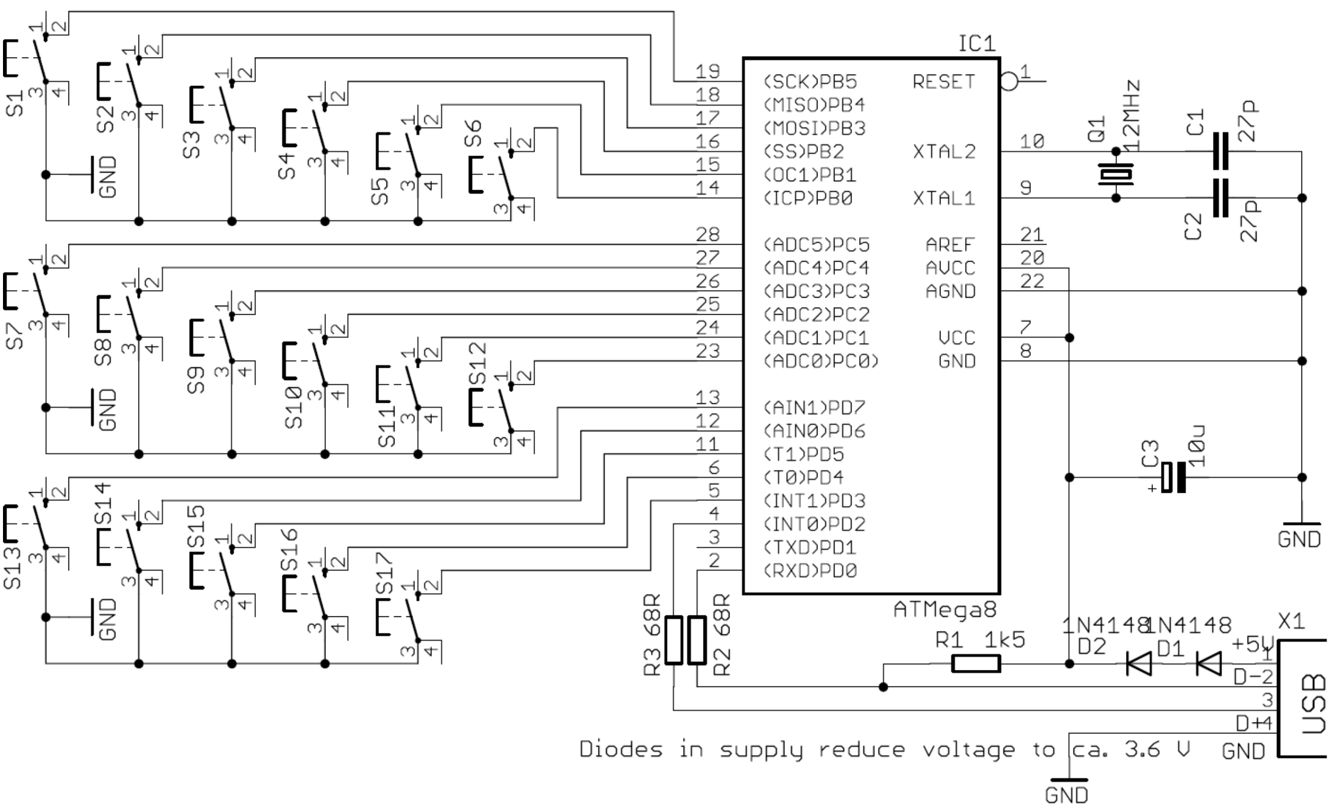
Exercise, Part 2

In this part of the exercise, we will USB enable the microcontroller by means of USB SW emulating using "Objective Development"'s V-USB:



From the V-USB web page http://www.obdev.at/products/vusb/index.html the driver can be downloaded and numerous AVR GCC project examples can be found.

Most examples use the 8-pin AVR microcontroller Tiny45 (or Tiny85), but for this exercise we will start focusing at an example using the Mega8 microcontroller. Originally this example is called "HIDKeys" and uses this hardware:



The device will present itself as a USB keyboard device (HID class) having 17 keys.

For easy test at STK500, the project has been ported to Mega32 and embedded into an Atmel Studio 6.0 project (now the project is called "HIDKeys_HH" and 24 keys are available, since PORTA, PORTB and PORTC all are used for keys).

The "HIDKeys_HH" is available for download at AMS Blackboard.
Start by downloading this and unzip to a proper folder at your PC.

Spend some time to study the project files and the V-USB documentation (plus the V-USB Wiki at vusb.wikidot.com).

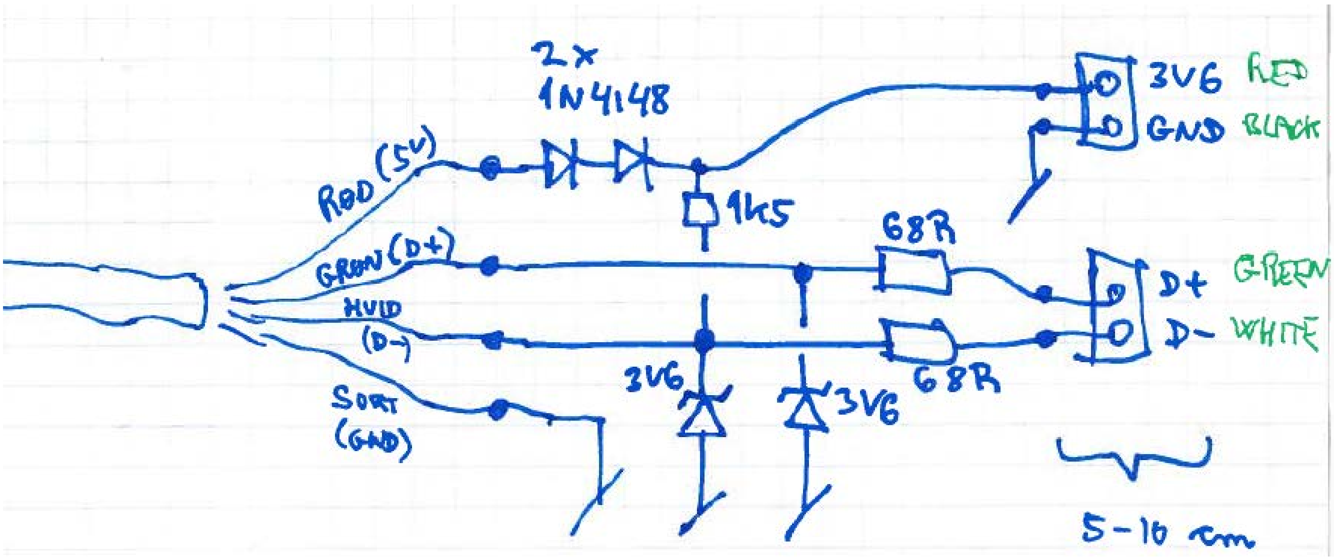
Files of most interest are:

- “HIDKeys.c”: The main function.
- “usbconfig.h”: Defines for configuring the driver.
- “usbdrv.h”: Header file for most driver functions (detailed explanations).

Then test the program (download to STK500).

Use the special designed USB interface cable to connect the PC USB port to STK500
(D-, D+, GND ,(optionally 5V)).

The cable implements the two 1N4148 diodes, the two 68 ohm resistors and the 1,5 kohm pull up resistor as shown at the diagram (prior page).
Moreover D- and D+ each are protected with 3,6 V zener diodes to GND:



A CPU frequency of 12 MHz is required. Therefore mount a 12 MHz crystal in the STK500 socket and change the setting of the jumper “OSCSEL”.

A simple way to test the program could be to open for example “notepad” and then press the STK500 keys. Then try to modify the program in some way and test the changes.

Extra: If you want to test some of the V-USB examples using a Tiny microcontroller, you can borrow a Tiny85 for that purpose. For testing and SPI programming using STK500 the Tiny85 has to use the socket SCKT3400D1, RST(PORTE) should be connected to PB5(PORTB) and XT1(PE) connected to PB3(PORTB). Also Mega8 controllers can be borrowed for experiments.

Exercise, Part 3

Using a microcontroller with an in-build USB controller is another way to USB enable.
Especially if you are to implement an embedded USB host, this would be the best choice.

Understanding the structure of in build USB controllers is far more than we have time for in this exercise !

However, spend some time to study two examples of evaluations boards using microcontrollers with on chip USB controller:

- The Atmel USBkey evaluation board (AVR microcontroller = AT90USB1287).
- The Atmel XPlained evaluation board (XMEGA microcontroller = XMEGA256A3BU).

Both evaluations board can be borrowed for experiments and user guides can be found at Blackboard.

Finally, spend a little time surfing the internet for stand-alone USB controllers (FTDI is one alternative, but many other companies produces stand-alone USB controllers).