Software USB on STM8 microcontrollers

STM8

At one point, the thought "is it possible to implement software USB on STM8 microcontrollers, similar to V-USB for AVR microcontrollers" got into my head.

And I don't know why, but I started working on this implementation (though I didn't have enough enthusiasm to finish it Sad face, but more on that later).

As far as I know, the first person to implement this protocol on AVRs was Engineer Igor Cesko , and later Atmel published the document AVR309: Software Universal Serial Bus (USB) (read the translation into Russian).

Igor implemented the reception of data in coded form with subsequent analysis, and in V-USB, NRZI decoding and sampling of insignificant bits occurs "on the fly".

(If I made a mistake somewhere, please don't kick me too much, I didn't go very deep into this topic).

Unfortunately, direct porting of code from AVR to ATM8 is not possible due to differences in architecture and command set.

The main STM8 problems I encountered:

- small number of registers (the main part of operations is available only for register A)
- three-level conveyor
- indefinite time to enter interrupt handler (9 cycles to save context and 1 to 6 to complete commands)

The pipeline causes the most problems, it is impossible to say for sure how many cycles a particular code will be executed.

The execution time depends on the state of the pipeline at the beginning of the code execution and the location of the commands in the memory.

Microcontroller

Live

Comments Publications

penzet → Sprint Layout in OS X 18 → Software for electronics engineer

Vga → EmBitz 6 → Software for electronics engineer

Vga → Rail-to-rail: ideal operational amplifier or a clever marketing ploy? 1 → Theory, measurements and calculations

Flint → A little more about 1-wire + UART 56 → Hardware connection to the computer.

penzet → Cross-platform terminal - SerIO 3.x 25 → Software for electronics engineer

From Gorn → PIP regulator 2 → Algorithms and software solutions

whoim → CC1101, Treatise on tracing 89 → Blog im. Khomin

OlegG → Clock on Bluetooth LE module 8 → Cypress PSoC

Technicum505SU → Nuances of PWM control of a DC motor by a microcontroller 3 → Theory, measurements and calculations

sunjob → DDS synthesizer AD9833 88 → Blog named after grand1987

dihalt → W801, LCD screen and tar spoon 2 → Detail

nictrace → New Arduino-compatible board. 20 → FPGA

sunjob → Connecting the ARM GCC compiler to CLion 1 → Software for electronics engineers

Vga → CRC32: on STM32 as on PC or on PC as on STM32. 58 → STM32

dmitrij999 → Capturing images from a USB camera using STM32 6 → STM32

Gilak → Expanding the capabilities of a simple MC up to an ADC on 2 or 1 pin. 8 → Theory, measurements and calculations

sva_omsk → Lithium ECAD - Russian PCB CAD 40 → Software for electronics engineer

x893 → W801 - budget controller with Wi-Fi 4 → Details

podkassetnik → Changing the standard instrument cluster lighting of Logan-like cars 5 → Automotive electronics

I chose the STM8S103F3 microcontroller as a test subject (I was aiming to use the STM8S003 Smile).

Although you can use any of the STM8L, STM8S series. Clocking from an "external quartz" at 12 MHz, the frequency is divided by a multiple of the USB clock (8 cycles per bit).

The USB lines are connected directly to PC7 and PC6.

Data transmission

It was wiser to start with signal generation (transmission), so that it would be easier to debug the receiver code by transmitting its data, rather than capturing data from a working USB bus.

To generate differential signals, I used two lines of port C, but the entire port is occupied, i.e. its other lines cannot be used during data reception and transmission. Initially I implemented the transmission without coding in NRZI and inserting empty bits, but later I managed to do it on the fly, without preliminary data preparation.

Signal generation algorithm: *I haven't drawn algorithms for a long time, maybe I drew some elements incorrectly.* Multiples of the algorithm:

2014-02-21-Block-diagram-of-transmission-algorithm

1. Register A is used only to count consecutively transmitted units.
2. The output signal is formed only by inverting the output states of all lines of port C at once.
3. After all data has been transmitted, the duration of the end-of-packet signal formation consists of the time of exiting the transmission subroutine and performing additional operations; it may be better to switch the lines to the output and form the duration of the "end-of-packet" signal with subsequent transfer of the lines to the input in this subroutine.

After writing the receiver code, I gained more knowledge and practice in STM8 assembler and some moments could have been made simpler and more beautiful, but at this stage I don't see any point in editing anything.

For the consistency of interval formation, the code should be located in a fixed place in the flash memory, I think it can be moved, but only with word alignment (I haven't tried it myself).

At the moment, the transmitter code takes up about 300 bytes. The

transmitter code has a lot of empty "nop" commands, some of them are used to align the code in the flash memory, the rest are used to form delays (the microcontroller works very quickly Smile), a total of almost 50% of the total number of commands.

That's all for today, describing everything at once is too much writing :(

SMT8, USB, Software

+10 21 February 2014, 08:17 **ZiB**

Comments (0)

RSS Collapse / Expand

Vga → ROPS (Rem Object Pascal Script) - embedded interpreter of the Pascal language. Plugin PSImport_Classes 3 → Algorithms and software solutions

Full broadcast | RSS

1-Wire The other arduino ARM Assembler Atmel AVR C++ compel DIY enc28j60 ethernet FPGA gcc I2C AND KEIL LaunchPad LCD led linux LPCXpresso MSP430 nxp PCB PIC pinboard2 RS-485 RTOS STM32 STM8 STM8L OF UART USB algorithm assembler ADC the library power unit detail display an idea tool competition competition 2 ANGRY microcontrollers for beginners review Debug board soldering iron printed circuit board salary FPGA crafts purchases programmer programming Light-emitting diode software scheme circuit design Technologies smart House photoresist it's free crap Times humor

Blogs

Group

AVR	**38.98**
STM8	**37.92**
Garbage truck	**29.53**
STM32	**28.46**
Detail	**24.63**
Connection of hardware to the computer.	**24.04**
Circuit design	**18.15**
Smart House	**17.75**
MSP430	**17.13**
LPC1xxx	**14.79**

All blogs

Only registered and authorized users can leave comments.