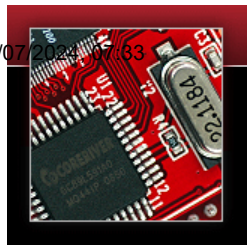

Software USB on STM8 Microcontrollers (Part 2)

STM8

We continue the description of software USB based on STM8 microcontrollers (read the first part).

Today I will tell you about the receiver code and about the first turn on of the device:

image

Data transfer

In the process of "connecting" the device to the PC, it was necessary to correct the transmitter code, now the address (pointer) to the data array is transferred to the subroutine.

Data reception

Unfortunately, at the moment it was not possible to cut out insignificant bits on the fly, only NRZI decoding is implemented.

Both lines are also used during reception, this allows decoding NRZI with literally two commands.

Here is a brief algorithm of the receiver (decoder) operation: Using a test signal is redundant, and the check can be done differently, for example, decoding and checking the end of the packet can be done like this:

2014-02-22-Block-diagram-of-reception-algorithm

```
srl             a
xor             a,(y)
tnz (y)
jreq    L_End_Rx
```

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 worked with this code last time when I tried to implement the cutting of insignificant bits.

An important part in decoding is the capture of signals always at the same time.

Unfortunately, STM8 commands can be executed for a long time from 1 to 6 cycles, to eliminate this drawback, I had to use one timer.

At the first signal edge, timer 1 is started, after entering the interrupt handler, the counting register is read and a transition is performed to the number of cycles (nop commands).

Thus, it turns out that the greater the delay in entering the interrupt, the further the transition according to the code.

This approach allows you to "hold" the capture point within one microcontroller cycle.

This is perhaps the key point, without it it is impossible to ensure the start of signal capture at the same time.

Here is how it looks in assembler:

```
_usb_rx:

        ldw x, #L_Delay_Begin ; address of the beginning of the delay ta
        addw x, 21086 ; + offset (timer value)
        cpw x, #L_Delay_End ; table bounds check (protection!)
        jruge   L_Error
        jp(x) ; jump through the delay table

L_Error:

        iret ; emergency exit

L_Delay_Begin:
        nop
        nop
        nop
        nop
        nop
        nop
```

Please excuse me for such a chaotic description of the implementation.

I implemented reading the descriptor from the protocol, and not completely.

I will not describe it, since I myself do not understand well what is happening there. Download Smile

the source code of the project **. Attention!!! The code is RAW, without formatting and careful checking. The code is compiled by the Cosmic compiler version 4.3.7, unfortunately I do not use IAR.** You can use the code without restrictions, just one request to link to my site, and indicate the author, i.e. me :)

STM8 , USB, Software, STM8S103F3

+10 22 February 2014, 17:28 **ZiB** 1

Files in topic: 2014-02-STM8S-USB.zip

Comments (11)

RSS Collapse / Expand

Cool, is it possible to do this in IAR, partially with C code? 0

KT3012
22 February 2014, 18:06

I think it's possible, probably just to adjust the asm to fit the IAR style, but unfortunately I'm not strong in this area. 0

ZiB
22 February 2014, 18:10 ↑

Why not attach the code to the topic? And the video can be inserted here. 0

Vga
22 February 2014, 18:24

I don't know how to video :(0

ZiB
22 February 2014, 18:37 ↑

Made :) 0

ZiB
22 February 2014, 18:41 ↑

It is also advisable to redirect the links to the first part and "download" to a local copy and to the attached file, respectively. 0

Vga
22 February 2014, 18:45 ↑

made :) 0

ZiB
22 February 2014, 18:48 ↑

A miracle happened!!! This is fantastic! We've been waiting for this for so long! :)

ZiB, I'm ready to donate, tell me where and how! You can send me a PM, or better yet, by email.

Azzzooo@gmail.com

0

Oh my!
04 April 2014, 13:57

There is still a lot of work to do before USB, I just showed the possibility of implementation.

0

ZiB
04 April 2014, 17:27 ↑

The Chinese have also already taken notice:
www.amobbs.com/thread-5585137-1-1.html

0

Exeland
June 26, 2014, 5:01 PM

Yes, they have been bombarding me with messages for a long time. But it seems none of them could get it to work :(Although I asked two independent people to check it for themselves. No one had any problems, my demo worked as expected.

0

ZiB
June 26, 2014, 5:23 PM ↑

Only registered and authorized users can leave comments.