Topics Blogs People Forum Shop Competition Help Enter or register

EasyElectronics.ru Community

All Collective Personal TOP

Good Bad

Поиск

Software delays for STM8

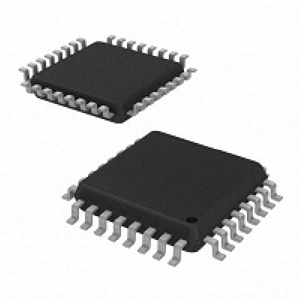
STM8

PCBWay HIGH-QUALITY PCB ONLY $5 FOR 10 PIECES
- Rogers, HDI, aluminum and rigid-flex PCB are available now
- Production time 24 hours

PCB ASS
Free shipping
ONLY
Compon
Quality a

Hello, All.

I'm tinkering with STM8 controllers and often come across the fact that some little things I'm used to are missing. This time, such a little thing was software delays. For example, AVR-LibC had delay.h for this.



It was its "interface" that I repeated for the animal I was learning. You can argue about the relevance of software delays, but it happens that there is still no alternative. For example, it is often easier than fiddling with a timer. Or when you need to form short time intervals: entering and exiting an interrupt require a fairly large number of clock cycles, which means we simply won't have time. There is another factor: when the time of entering an interrupt is poorly predictable and requires special attention.

Actually, the file can be taken from here .
11/27/2014 I didn't mention the compiler I tested it for - COSMIC. But now it works for SDCC too.

stm8 , stm8s , STM8L

+1 23 October 2014, 11:59 **Hoksmur**

Comments (31)

RSS Collapse / Expand

Mmmm, seriously? The downtime delay became the reason for the topic? -4

flaxirs
23 October 2014, 15:15

Yes and no. The reason is insignificant, you are right about that. I just got tired of calculating the number of cycles for different frequencies, and for different delays. And 0

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

here I transferred it to the compiler - it has silicon brains, they don't hurt... And I haven't seen a ready-made solution like that.

Hoksmur
October 23, 2014, 6:03 PM ↑

Good news, everyone!

+1

Fountain-G
October 23, 2014, 16:50

Bsd news, everyone. This thing has a conveyor belt, which means delays will be unpredictable...

0

king2
23 October 2014, 18:12 ↑

Yes, three-stage. There are many in the PM0044 Programming manual . I do not claim the accuracy of the excerpts, but it was enough for me. In theory, critical pulses are generated by the timer without the participation of the "brain", fortunately there are enough registers there.
On the conveyor - the execution time of commands is predictable, but I still have not fully mastered its loading according to the above document. More precisely - how long will it wait after the transition is performed. I expected an error in delays from the calculated values towards an increase, but in practice it turned out the opposite. Or I calculated something incorrectly.

+1

Hoksmur
October 23, 2014, 6:29 PM ↑

It is very difficult to predict the execution time, the same code can have very different execution times depending on its location in memory. When I was making a software USB on STM8 I had to tinker with this crap :(

0

ZiB
24 October 2014, 04:51 ↑

I read the result in the blog - starting the timer from the same output from which we catch the change of the pin - a beautiful solution!

0

Hoksmur
24 October 2014, 05:27 ↑

Спасибо.
Другого решения просто не смог найти, да и наверное его нет ;)

0

ZiB
24 октября 2014, 06:02 ↑

Как я понимаю, гадит не столь конвеер, сколь особенности доступа к флешу? Что-то вроде кеша или выравнивания команд?

0

Vga
24 октября 2014, 09:03 ↑

Если есть желание посмотреть самому — Подраздел 5.3, страница 23 выше по ссылке и дальше. Например, в 5.4.1:

> *In the example shown in Table 6, the code is stored in the Flash Program memory (32-bit bus). As a result, 3 cycles are needed to fill the 96-bit prefetch buffer. At each cycle, one word is loaded and stored in F1, F2 and F3. The next fetch operation can start only when all the instructions contained in one of the Fx word are decoded. In fact, at cycle 9, the last instruction contained in F3(SWAP A) is decoded, and a fetch operation can start to fill F3 word.*

0

Hoksmur
24 октября 2014, 09:11 ↑

0

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

Ну почему же, на мой взгляд он тоже не позволяет определить точное время выполнения команд.

ZiB
24 октября 2014, 09:15 ↥

Ну, не до такой степени, чтобы зависеть от расположения команд в памяти. По моему, более предсказуемая штучка.

0

Vga
24 октября 2014, 09:22 ↥

На худой конец, можно отсчитать задержку по таймеру, но поллингом, без прерываний.
Опять же, организация в виде отдельного модуля позволяет абстрагироваться от конкретной реализации

0

MrYuran
24 октября 2014, 17:26 ↥

Программные задержки- зло, хотя в некоторых простеньких проектах- их достаточно.

-2

Zlodey
23 октября 2014, 19:22

Каждый раз, когда ты пишешь в обработчике прерывания задержку, в мире умирает котенок(С)

+2

xar
23 октября 2014, 23:31 ↥

так я ж говорю зло

0

Zlodey
24 октября 2014, 08:13 ↥

Таки кто вас заставляет делать это? Задержки в прерываниях? «Месьё знает толк в извращениях!» (с)

+1

Hoksmur
24 октября 2014, 08:15 ↥

У AVR асмовская вставка объявлена как volatile, а у Вас нет. Это конечно зависит от компилятора, но теоретически оптимизатор может выбросить вашу задержку из кода.

+1

e_mc2
24 октября 2014, 11:34

работа полезная, но, увы, не работает.

0

```
$ make
sdcc -mstm8 -lstm8 --out-fmt-ihx   -o blinky.ihx blinky.c
In file included from blinky.c:9:
delay.h:20: syntax error: token -> 'void' ; column 18
make: *** [blinky.ihx] Error 1
```

особый синтаксис?

собираю этой версией (свежак):

```
$ sdcc -v
SDCC : mcs51/z80/z180/r2k/r3ka/gbz80/tlcs90/ds390/pic16/pic14/TININative/
```

Doka
26 ноября 2014, 19:34

0

Это под Cosmic писал. Похоже, что SDCC не переварил комбинацию *static inline void*.
Посмотрите в документации, как в используемом вами компиляторе оформить следует.

Hoksmur
26 ноября 2014, 20:08

Попробуйте оставить только **inline void**

0

Hoksmur
26 ноября 2014, 20:13 ↥

да-да… нарыл что нужно включить STD99, включил, также переписал ассемблерную вставку согласно синтаксиса, убрал const из объявления переменной передаваемой функции, теперь интереснее:

```
sdcc -mstm8 -lstm8 --std-sdcc99 -DF_CPU=8000000 --out-fmt-ihx   -o bli
delay.h:24: warning 85: in function _delay_cycl unreferenced function
blinky.asm:116: Error: <m> multiple definitions error
blinky.asm:116: Error: <p> phase error: label location changing betwee
```

0

подскажите как эти де строчки переделать в одну:

```
//ldw X, __ticks ; insert automaticaly
        __asm("nop\n $N:\n decw X\n jrne $L\n nop\n ", __ticks);
```

поскольку у инлайнового ассмблера такой синтаксис:

```
__asm__ (”; This is a comment\nlabel:\n\tnop”);
```

т.е. никаких запятых внутри скобок он не понимает.

Doka
26 ноября 2014, 20:19 ↥

У Cosmic компилятор перед асм-фрагментом кладёт __ticks в регистр X.
Получается:

nop
label_1:
decw X
jrne label_1
nop

I added nop at the beginning and at the end to align the bars. Can you leave only one or two nop inside and dump the resulting listing, hooking a few lines before it?

0

Hoksmur
November 26, 2014, 20:27 ↥

SDCC has a nearly ready recipe in the documentation in section **3.14.4 Use of Labels within Inline Assembler.**
Share the result? I'll add it to the file later.

0

Hoksmur
November 26, 2014, 20:32 ↥

```
;--------------------------------------------------------
; code
;--------------------------------------------------------
        .area CODE
;       delay.h: 20: static inline void _delay_cycl( unsigned s
;       -----------------------------------------
;        function _delay_cycl
;       -----------------------------------------
__delay_cycl:
;       delay.h: 24: __asm__("nop\n 00001$:\n decw X\n jrne 000
        nop
         00001$:
```

0

```
        decw X
        jrne 00001$
        nop
        ret
;       delay.h: 27: static inline void _delay_us( unsigned sho
;       ----------------------------------------
;        function _delay_us
;       ----------------------------------------
```

but check with the addition of "ldw X, __ticks":

```
;       delay.h: 22: static inline void _delay_cycl( unsigned s
;       ----------------------------------------
;        function _delay_cycl
;       ----------------------------------------
__delay_cycl:
;       delay.h: 25: __asm__("ldw X,_ticks\n nop\n 00001$:\n de
        ldw X,_ticks
        nop
         00001$:
        decw X
        jrne 00001$
        nop
        ret
;       delay.h: 28: static inline void _delay_us( unsigned sho
```

the syntax itself:

```
unsigned short ticks;

static inline void _delay_cycl( unsigned short ticks )
{
        ticks; // sdccman.pdf:  to avoid warning  "unreferenced
        __asm__("ldw X,_ticks\n nop\n 00001$:\n decw X\n jrne 0
}
```

Law
November 26, 2014, 21:16

There is another problem related to labels in assembler: if you call the delay procedure from several places, the same label is used in the final file, and as a result you get:

0

```
blinky.asm:188: Error: <p> phase error: label location changin
blinky.asm:188: Error: <m> multiple definitions error
blinky.asm:217: Error: <p> phase error: label location changin
blinky.asm:217: Error: <m> multiple definitions error
```

Law
November 26, 2014, 21:23

0

```
do {
                ticks--;
        } while (ticks);
```

is compiled by the compiler into

```
ldw     x, #0x007f
00104$:
        decw    x
        tnzw    x
        jrne    00104$
```

It's easier to recalculate the cycles, add nop if necessary to align the cycles, and correct the formula in _delay_us
But I'll only get to that tomorrow.

Hoksmur
November 26, 2014, 10:30 PM

0

Try replacing the body of **the _delay_cycl** function with the following fragment:

```
#if defined(__CSMC__)
/* COSMIC */
  #define T_COUNT(x) (( F_CPU * x / 1000000UL )-3)/3)
        // ldw X, __ticks ; insert automaticaly
        _asm("nop\n $N:\n decw X\n jrne $L\n nop\n ", __tic
#elif defined(__SDCC)
  #define T_COUNT(x) (( F_CPU * x / 1000000UL )-5)/5)
        __asm("nop\n nop\n");
        do {            // ASM: ldw X, #tick; lab$: decw X;
                ticks--;//      2c;                 1c;
        } while (ticks);
        __asm("nop"); // allign last cycle
#elif defined(__RCST7__)
/* RAISONANCE */
  #error "ToDo for RAISONANCE"
#elif defined(__ICCSTM8__)
/* IAR */
  #error "ToDo for IAR"
#else
 #error "Unknown Compiler!"          /* Compiler defines no
#endif
```

PS: It seems that IAR can't inline. No matter how much I tried, it always uses call.

Hoksmur
November 27, 2014, 06:42

0

And in _delay_us on

```
_delay_cycl( (unsigned short)( T_COUNT(__us) );
```

Hoksmur
November 27, 2014, 06:44

0

thanks, it seems to have started, here is the asm:

```

;---------------------------------------------------
; code
;---------------------------------------------------
        .area CODE
;       delay.h: 22: static inline void _delay_cycl(
;       -----------------------------------------
;        function _delay_cycl
;       -----------------------------------------
__delay_cycl:
;       delay.h: 33: __asm__("nop\n nop\n");
        nop
        nop
;       delay.h: 34: do {            // ASM: ldw X,
        ldw     x, (0x03, sp)
00101$:
;       delay.h: 35: ticks--;//      2c;
        decw    x
;       delay.h: 36: } while (ticks);
        tnzw    x
        jrne    00101$
;       delay.h: 37: __asm__("nop"); // allign last
        nop
        ret
;       delay.h: 49: static inline void _delay_us( u
;       ----------------------------------------
```

PS: according to sdcc syntax, asm inserts instead of __**asm** ("nop"); should be formatted as __**asm**__ ("nop");

Law
November 27, 2014, 14:57 ↑

The problem is different. I still haven't found how to pass a parameter to an asm insert in SDCC.

0

Hoksmur
November 27, 2014, 4:19 PM ↑

Only registered and authorized users can leave comments.

Design by — Студия XeoArt

© Powered by LiveStreet CMS