STM8 Microcontrollers. Introduction.

STM8

STM8 Microcontrollers. Introduction.

Hello,
Today I would like to start a series of articles about a fairly new and promising family of STM8 microcontrollers. To date, there is not much Russian-language information on this family, but we, with the joint efforts of the entire we.easyelectronics.ru community, will try to fill this gap.

So, the first STM8 microcontrollers were released by STMicroelectronics to replace the ST7 family in early 2008. At the same time, a sixfold increase in productivity was declared compared to the previous family. The main emphasis was on the high reliability of crystals, their protection from static voltage. At the same time, their use was implied primarily in the automotive industry, as well as in tasks that require high reliability with low power consumption. Somewhere in the middle of 2010, these controllers appeared on wide sale in our country, and forced attention to themselves with their price, wide range and completely unrealistic prices for Atmel products. Let's take a closer look at what STMicroelectronics offers us.
General overview of the architecture.
So, the core is built on the Harvard architecture, that is, the address space of commands and data is divided, but this division is internal. For the programmer, the processor is von Neumann and allows using a single address space. The maximum core frequency is up to 24 MHz for senior models. Flash memory size is from 4 to 128 KB, SRAM size is from 1 to 6 KB.
The core and peripherals of the microcontroller operate at a supply voltage of 1.8 V, while the power supply of the controller itself depends on the series. The STM8 family is divided into three subfamilies:
- STM8L - a family of low-power microcontrollers for battery-powered devices.

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

Gornist → 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 fly in the ointment 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

Gilaks → 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

The supply voltage range is from 1.65 to 3.6 V.
- STM8S - general-purpose microcontrollers. The supply voltage range is from 2.95 to 5.5 V.
- STM8A - highly reliable microcontrollers.
In the future, we will focus on general-purpose controllers as the most common and inexpensive.
The STM8 stuffing is basically the same as that in AVRs and PICs, so those who have mastered programming for these platforms will find it easy to switch to STM8 if they wish. Let's take the STM8S105C6T6C microcircuit as an example. The
microcircuit code can be decoded as follows:
STM8S – STM8S family.
The next three digits:
103, 903 – low-density devices.
105 – medium density devices.
207, 208 – high density devices. The 208 series differs from the 207 by the presence of hardware CAN.
The letter tells us about the number of case pins. K – 32; S – 44; C – 48; R – 64; M – 80.
Next, the volume of program flash memory is indicated:
3 – 8 KB;
4 – 16 KB;
6 - 32 KB;
8 — 64 KB;
B — 128 KB.
Package type: T — LQFP, B — SDPI, U — VFQFPN.
Operating temperature range: 6 — -40...+85 °C; 3 — -40...+125 °C.
And at the end, the pin pitch is indicated: if there is no letter, the pin pitch is 0.5 mm., letter B — 0.65 mm., letter C — 0.8 mm.
Accordingly, in our example, the STM8 microcontroller, family S, related to medium density devices, in an LQFP package with 48 pins, which has 32 kilobytes of Flash memory on board, with an operating temperature range of -40...+85 °C and a pin pitch of 0.8 mm is encrypted.
What peripherals does it have? To do this, go here , read, look at the block diagram and see what's inside:

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

Full broadcast | RSS

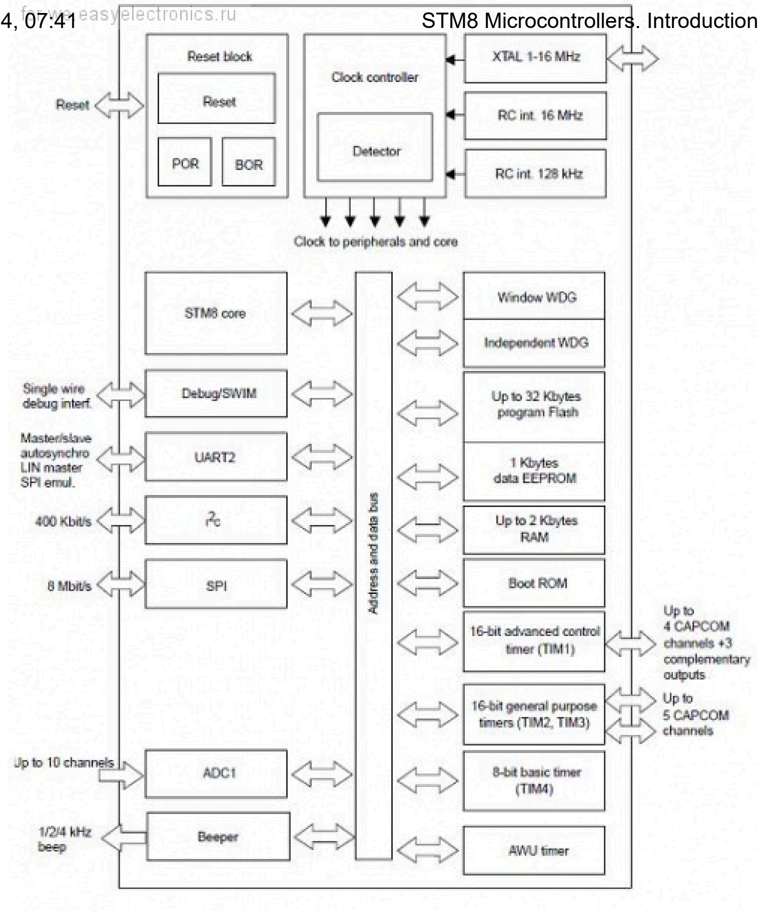
1-Wire Altera arduino ARM Assembler Atmel AVR C++ compel DIY enc28j60 ethernet FPGA gcc I2C IAR KEIL LaunchPad LCD led linux LPCXpresso MSP430 nxp PCB PIC pinboard2 RS-485 RTOS STM32 STM8 STM8L TI UART USB algorithm assembler ADC library power unit detail display idea tool contest competition2 LUT microcontrollers for beginners review Debug board soldering iron printed circuit board pay FPGA crafts purchases programmer programming Light-emitting diode software scheme circuit design Technologies smart House photoresist freebie crap Watch humor

Blogs

Top

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



- internal adjustable 16 MHz RC generator;
- internal low-speed, low-power 128 kHz RC generator;
- clock source security system;
- interrupt controller;
- two 16-bit general-purpose timers;
- one 16-bit special timer;
- two watchdog timers;
- 10-bit ADC;
- interfaces - UART, SPI, I2C;
- 38 input/output lines, 16 of which have increased load capacity.

In my opinion, not bad. Particularly impressive is the flexible clocking system and the ability to fine-tune interrupts. There is also a built-in speaker-tweeter control module, which can also be used as a frequency generator of about 1 kHz. Why it is there is a question for the developers.

Development and debugging tools.

At the moment, there are four environments for developing and debugging software for STM8: ST Toolset from STMicroelectronics, Ride7 from Raisonance, CXSTM8 from Cosmic software, IAR Embedded Workbench from IAR Systems.

To be honest, I personally hesitated for a long time and chose IAR simply because I have experience working in it. Accordingly, all further articles will go with examples in IAR. If someone wants to port examples to other environments, you are welcome.

The current version of IAR is 1.20, and we have two options - Kickstart, with a limit of 8 KB of code, and a 30-day trial version. To download, you need to go through a simple registration procedure, after which you will be given a link and an email with a key. Registration and installation are intuitive, so we will not dwell on this stage in detail.

Comparison with AVR.

Well, the most important question - is the game worth the candle? Is there any point in studying STM8 if you have Atmel AVR? So, let's see what advantages and disadvantages STM8 has compared to AVR.

Advantages:

- increased reliability and protection from static discharge are declared. AVR does not have this.
- flexible clocking system and no fuse bits. Sore points, especially for beginners. A huge plus.
— compatibility within the family. It is stated that the peripheral modules are practically the same in different models, which facilitates portability.
— lower or comparable cost compared to AVR. After last year's problems with the supply of Atmel controllers, it does not quite inspire confidence in the future.
— the presence of a single address space for code and data.
— more sophisticated, in my opinion, peripheral devices.
— the presence of a native firmware library.
— the presence of budget Discovery debug boards.
— low cost of the ST-Link programmer.
— a clock frequency of 16 MHz over the entire range of supply voltages, be it STM8L or STM8S
— the presence of PVD on STM8L — programmable voltage detector

Disadvantages:
— low prevalence.
— small amount of literature.
— a limited number of development environments.
— lack of models with less than 20 pins.
— difficulties in creating homemade programmers. There is no analogue of the "five wires" or Gromov programmer for STM8.
— no analog comparator (some models have an analog comparator COMP2)
Based on this comparison, we can conclude that STM8 controllers are at least as good as similar solutions from Atmel, and the main thing that holds back their widespread adoption is the lack of information for developers. But this is probably just a matter of time. That's

all for today, and in the next article we will look at the STM8S-Discovery debug board and create the first project.

Links:
STM website section dedicated to STM8
Reference manual for STM8S and STM8A
Datasheet for the STM8S105xx family
Firmware library
IAR for STM8

STM8 , contest , microcontrollers , review

+5 07 March 2011, 17:33 **Kalvenolt**

Comments (42)

RSS Collapse / Expand

I liked the fact that smt8 has a board for 300 rubles — stm8-discovery with a built-in debugger, which can then be connected to your circuit, and 3 wires are enough for debugging. I have not come across a cheap or easy-to-make debugger for AVR.
The disadvantage was how the port pins are scattered.
I advise beginners to pay attention to the fact that some pins can be open drain.
What I did not like about IAR for STM8 is that there are no _delay_ functions, which are sometimes very much lacking.

+1

citizen
07 March 2011, 20:19

For AVR, a cheap debugger is JTAGICE, which will cost 150 rubles in the minimum configuration. But it is not for all MKs.

0

Well, the lack of a do function is not a disadvantage. Just write your own, that's all.

DIHALT
07 March 2011, 20:25 ↑

> *Well, the lack of a do function is not a disadvantage. Just write your own, that's all.*

If I knew asm, I would have written it, but since I don't, I just had to put cycles. The developers themselves offer their own delay function in the library, but it is based on one of the timers.

0

citizen
07 March 2011, 20:41 ↑

```
for(i=1;1<10000;i++)
{
}
```

Is it no longer relevant?

0

DIHALT
07 March 2011, 21:53 ↑

typo **enemy!** -> infinite loop

0

XuMuK
07 March 2011, 22:52 ↑

oh, I definitely miscalculated
for(i=1;i<10000;i++)
of course.

0

DIHALT
07 March 2011, 23:38 ↑

Delay is cycles in 99% of cases. There are also problems with timers, but this is more exotic or for large and precise delays.

0

Ultrin
07 March 2011, 22:22 ↑

This is exactly the cycle I wrote, although I stuck nop in there. But in wingcc, as I noticed, the delay function contains a cycle in assembler, apparently to increase accuracy.

0

citizen
07 March 2011, 22:39

Thank you. Added.

0

Kalvenolt
08 March 2011, 00:58 ↑

It is necessary to add to the +
- 16 MHz in the entire range of supply voltages, be it STM8L or STM8S
- COMP2 comparator - analog (on STM8L15x, you can even supply the output from the DAC to the second input)
- PVD on STM8L - programmable voltage detector

, so we take STM8L15x - powered from two AA (AAA) batteries
PVD monitors the voltage of these batteries
and all this can be at a maximum frequency of 16 MHz

0

XuMuK
07 March 2011, 22:44

I answered in the wrong place) I still can't get used to it.

0

Kalvenolt
08 March 2011, 00:59

But the compilers for STM8 are still tight. There is no GCC port for this family. The only adequate option is IAR, the rest is so-so.

0

neiver
07 March 2011, 23:00

And is Cosmic crap?

0

DIHALT
07 March 2011, 23:39

I don't know, I haven't tried it. But there is no C++ there :)

0

neiver
08 March 2011, 10:43

and STM8L also has DMA.

0

citizen
07 March 2011, 23:01

and gcc and debugging tools for Linux for them have not been made yet? that's the saddest thing, but the stones are quite tasty

0

marvin_yorke
07 March 2011, 23:14

I'm afraid they won't do it for a long time.

0

DIHALT
07 March 2011, 23:39

What do you think is more of a hindrance - the complexity of implementation or the relatively low demand?

0

marvin_yorke
07 March 2011, 23:55

Free and open source. A compiler is a very complex thing and there are no fools to give out their product to everyone for free. And buying software for Linux is not accepted :)

0

DIHALT
08 March 2011, 10:11

Well, it's not just about Linux. There are gcc ports for AVR, ARM, MSP430 and a bunch of other things, both Linux and Windows. And for ARM, there are both arm-none-* for assembling Linux itself and arm-linux-* for assembling applications for ARM-Linux. I'm currently digging into Cortex-M3 and ChibiOS/RT — I assembled the toolchain from the sources: gcc-linaro, binutils, newlib, gdb, openocd

0

marvin_yorke
08 March 2011, 10:24

Ну это видать уже энтузиасты сделали. Когда нибудь и под STM8 будет. Но вот производитель меньше всего парится на счет этого — ему на линух сообщество пох. Лишняя трата ресурсов только.

0

DIHALT
08 марта 2011, 11:17

Стереотипное мышление это плохо… Не всегда пользователи линукс == халявщики. Но к сожалению не все производители ПО это понимают, а бОльшая часть линуксоидов этот стереотип лишь утверждает

0

marvin_yorke

08 марта 2011, 11:22

Кстати о платности, для ARM есть, например, Codesourcery G++, основанный на gcc и eclipse, и он платный. Вроде отличается от собранного ручками только какой-то спецовой сборкой Eclipse. Есть бесплатная версия, которая соержит только компилятор без среды разработки

0

marvin_yorke
08 марта 2011, 10:26

прояснил некоторые моменты

STM8L101
— нет ADC и PVD
— COMP1 и COMP2 имеют на "+" до 4 i/o на каждый
на "-" GND или i/o (но "-" общий для обеих компараторов)
STM8L15x
— есть и ADC и PVD и DAC
— COMP1 — логический
— COMP2 — на "-" можно подать вход i/o или выход DAC или с внутреннего резестивного делителя

STM8S
— ADC — есть
— PVD — нет
— компараторов нет совсем

0

XuMuK
08 марта 2011, 19:52

Поэтому мне больше поравились STM8L151K6 (LQFP32) по 45 рублей куил десяток, две уже ушли в девайсы.

0

ZiB
08 марта 2011, 19:58

Был у меня в арсенале STM8S-Doscovery.
Програматор в STM8S-Doscovery был благополучно спален при коротком замыкании на 24В (питание для шаговиков).

После куплена STM8L-Discovery. И тут без КЗ не обошлось — вылетел диод D2 (BAT60JFILM). На его место был поставлен аналогичный по размерам из имеющихся — 1N4148WS. Тот при случавйном КЗ (+3V3/GND) полетел еще быстрее.
На место D2(STM8L-Discovery) поставлен выпаянный из какого-то жесткого диска SMD-предохранитель.
Чисто ради эксперимента были замкнуты выводы питания +3V3/GND — предохранитель исправен. Замыкание +5V/GND — предохранителю кранты.

Тут терпение лопнуло, и на место D2 примостился самовосстанавливающийся предохранитель RXEF025 (Ic=0.25A, Umax=70V, Imax=40A) (http://kosmodrom.com.ua/table.php?name=ruefrxefntc&page=0). Преимущество решения — защита USB-порта от перегрузки по току (от маловероятных, но возможных 24..36V/1,5A). Очередное принудительное замыкание +5V/GND — сброс МК и всё работает дальше.

Различие STM8S-Discovery и STM8L-Discovery — в последнем хоть какая-то защита по линии питания от КЗ (диод шоттки, Umax=10V, Imax=3A).

Рекомендую в STM8S-Discovery сразу после покупки добавить защиту от КЗ в цепях питания как минимум между частью с програматором и частью с STM8S105C6 — будет хоть какая-то защита для програматора.

0

Rita
15 апреля 2011, 20:47

На очереди — предохранители на дорожки к выводам +3V3 и +5V в STM8L-Discovery. Благо там дорожки к выводам питания толстые и достаточно длинные, чтобы впаять диоды шоттки/предохранители.

0

Rita
15 апреля 2011, 20:51

Ссылка на IAR неправильная.

0

Krakozebl
19 ноября 2011, 10:22

SDCC для stm8 уже неплох, ещё не официально, но работает.

0

teplofizik
03 июля 2013, 10:19

Ну вот, наступило счастье на улице любителей stm8)

0

xterro
03 июля 2013, 12:59 ↑

Осталось к нему отладку прикрутить, тогда точно будет круть =D Но бесплатно компилировать уже можно

0

teplofizik
03 июля 2013, 13:14 ↑

А еще что-то опенсорсное для STM8 есть? Кажется, пилили бэкэнд для LLVM.

0

Vga
03 июля 2013, 16:04 ↑

Я больше ничего достойного не знаю. Видел ещё пару компиляторов, но они настолько сырые и не развивающиеся, что даже си не целиком поддерживают, фиг что соберёшь.

Для LLVM как-то новостей за последние два года нема, то ли заглохло, то на не афишируется…

Бранч SDCC с бэкендом stm8 до последнего времени тоже как-то был не лучше иного самопала, пока его не влили в основной репозиторий в июне сего года.

0

teplofizik
04 июля 2013, 11:15 ↑

Вот на сколько я знаю sdcc для stm8 разработчиком не поддерживается.
Он еще с год назад сказал, что: st7 я довожу до ума, а stm8 оставляю в том виде, в котором оно есть.
Разве что действительно с июня начали его пилить и можно на что-то расчитывать.

0

angel5a
04 июля 2013, 11:58 ↑

Да, тот, кто начал сие дело, всё давным-давно забросил. Его наработки перенесли в основную репу, которая на sdcc.sourceforge.net/.

0

teplofizik
04 июля 2013, 12:22 ↑

Как легко было портировать компилятор Си на новую платформу на заре становления этого языка. Компилятор преобразовывал код на Си в команды некоего абстрактного asm, оставалось только прописать файл(HAL) отображения этих абстрактных команд в команды asm конкретного CPU. Дубово и брутально, не оптимизированно — но это работало.

0

well-man2000
04 июля 2013, 12:45 ↑

В идеале же и сейчас так =) LLVM, gcc, sdcc — там тоже же идёт компиляция в промежуточный код, который потом переводится на нужный.

Другое дело, что я не смог так просто въехать, как свой

0

генератор кода (backend) к ним писать. Вроде, и информация есть, но там столько всего, что непонятно, откуда начинать.

teplofizik
04 июля 2013, 12:56 ↑

А TCC не смотрел? У него вроде довольно компактные кодогенераторы, килобайт 30 текста (сейчас есть x86, ARM, какой-то из TMS320). Правда, оптимизирует он паршиво.

0

Vga
04 июля 2013, 14:05 ↑

Смотрел, но там тоже как-то не очень прозрачно всё происходит. Да и код плотненький, и комментариев не очень

0

teplofizik
04 июля 2013, 14:26 ↑

Да, даже прерывания вроде работают =). Вон ребята уже даже скомпилили OggStreamer на STM8 с помощью SDCC

0

dotnot
28 января 2014, 00:16 ↑

Yes, I know =)

The only thing missing is debugging. I still don't understand if it's possible to somehow get elf with symbols, so that their stinky gdb.

0

thermophysicist
January 28, 2014, 00:22 ↑

> *These guys have even compiled OggStreamer on STM8*

What kind of animal is this?

0

Vga
January 28, 2014, 06:24 ↑

Only registered and authorized users can leave comments.

Design by — Studio XeoArt XeoArt

© Powered by LiveStreet CMS