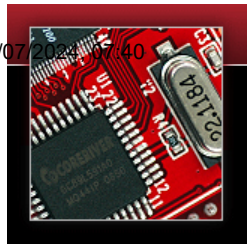


STM8 microcontrollers. First program.

STM8

STM8 Microcontrollers. First Program.
Hello,
Today we will talk about hardware development tools for STM8S microcontrollers and create the first project.

First, you should decide on the hardware platform, because the simulator is good, but, in any case, in the end everything will be implemented in hardware. For this, we need a programmer and a debug board. As I mentioned in the previous article, assembling a programmer for STM8 yourself is not an easy task, and requires serious knowledge and a lot of effort. There are serious debug kits on sale, like STM8/128-EVAL , and a universal ST-Link programmer, but for a quick start, ST has released two cheap debug kits STM8S-Discovery and STM8L-Discovery. They initially have a cut-down version of ST-Link built in, and thus, on one board we get a programmer and the target controller itself, ready for programming. And all this wealth costs very little money: for example, I got the STM8S-Discovery for 115 UAH (about $14). The difference between the STM8S-Discovery and STM8L-Discovery is, in addition to minor details, in the type of the installed controller, as well as the fact that the STM8L-Discovery has an energy-saving LCD indicator. We will take our first steps using the STM8S-Discovery, and then we will assemble our debug board. So, what do we have on this wonderful board?
STM8S-Discovery. Board review.
From the first moment, the board leaves a very pleasant impression. The package does not include anything except the board, but this is not surprising, given the price. The board is packed in a plastic box, familiar to many from the STM32VL-Discovery. You should take care of the cable for communication with the computer in advance - you need the most ordinary AB USB cable, which, for example, is used to connect a printer.

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

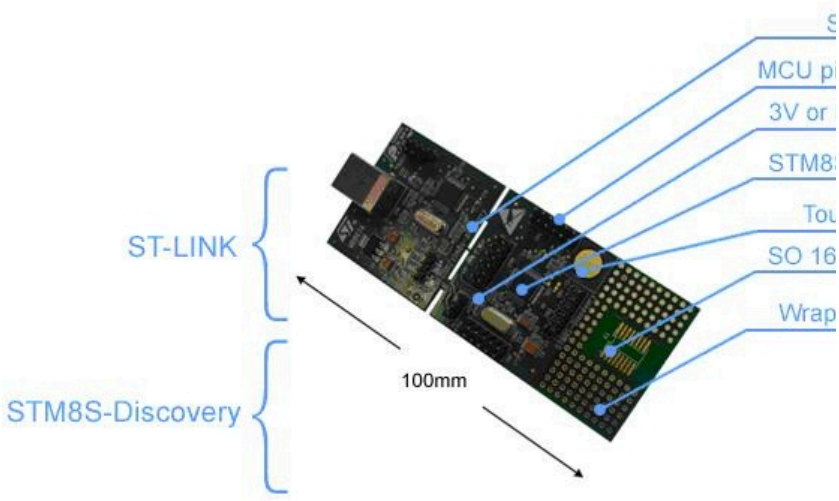
Gilaks → Expanding the capabilities of a simple MC up to an ADC on 2 or 1 pin. 8 → Theory, measurements and calculations

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

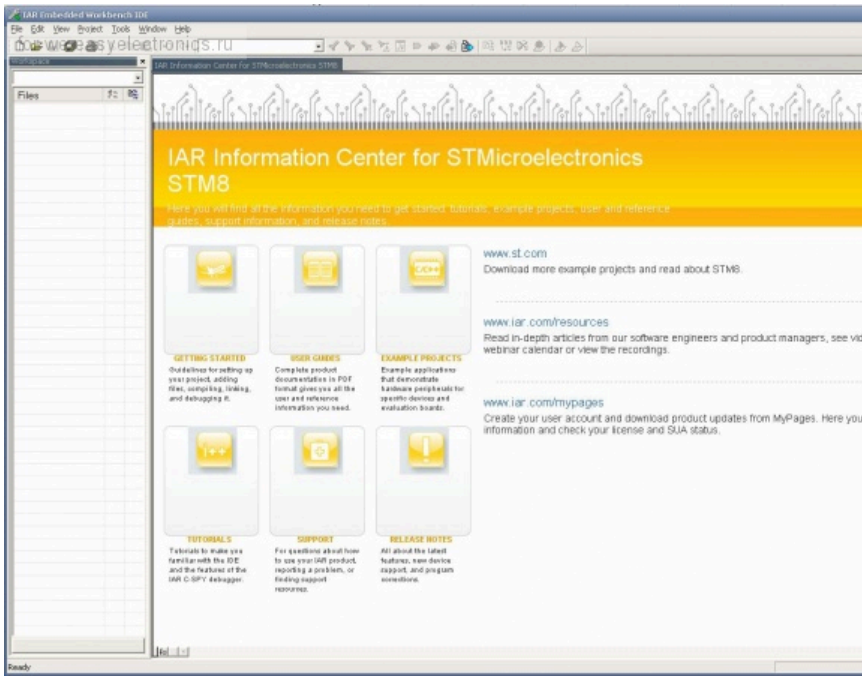
Looking at the board, we see that it is divided into two areas - the programmer and the debug board itself. The board is made so that it is convenient to break it into two parts ahead of time.
The programmer works via a two-wire SWD interface and allows you to flash not only the built-in controller, but also, when the corresponding jumpers are closed, external devices, which we will certainly use in the future. The programmer works with all families of STM8 microcontrollers, and there are rumors that it will soon be hacked, and it will be able to flash STM32 as well. The



debug board has an STM8S105C6T6 microcontroller, an external 16 MHz quartz, a touch button and an LED. All microcontroller pins are brought out to the connectors. At the bottom of the board there is a small breadboard area with a footprint for microcircuits in the SO 16 case and a small number of metallized holes.
Initially, a program is loaded into the controller, which allows you to change the LED blinking frequency by touching the touch button. But we will return to the touch button much later, and now we will create the first program for STM8.
Creating a project for STM8 is similar to creating projects in IAR for other platforms, but I will repeat myself so as not to interrupt the sequence of the course.

Creating the first project in IAR.

Let's launch the IAR Embedded Workbench for STMicroelectronics STM8 environment. The current version is 1.20. After loading, we see the following window:



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

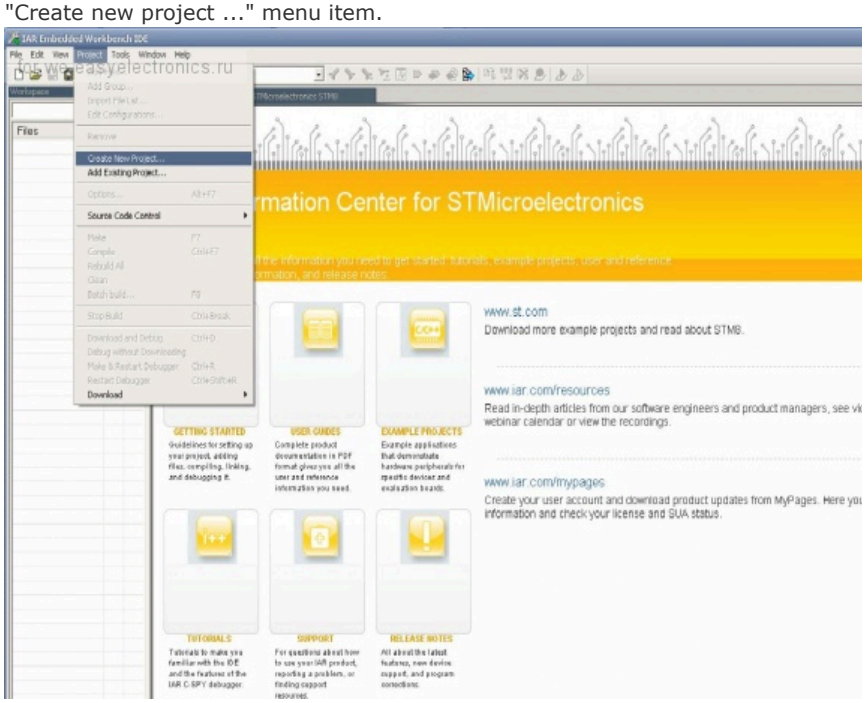
Blogs

Top

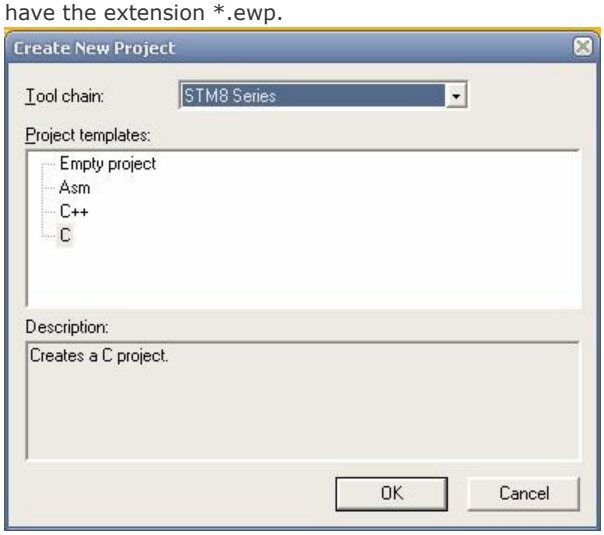
AVR	**38.98**
STM8	**37.92**
Garbage truck	**29.53**
STM32	**28.46**
Detail	**24.63**
Connection of hardware to the computer.	**24.04**
Circuit design	**18.15**
Smart House	**17.75**
MSP430	**17.13**
LPC1xxx	**14.79**

All blogs

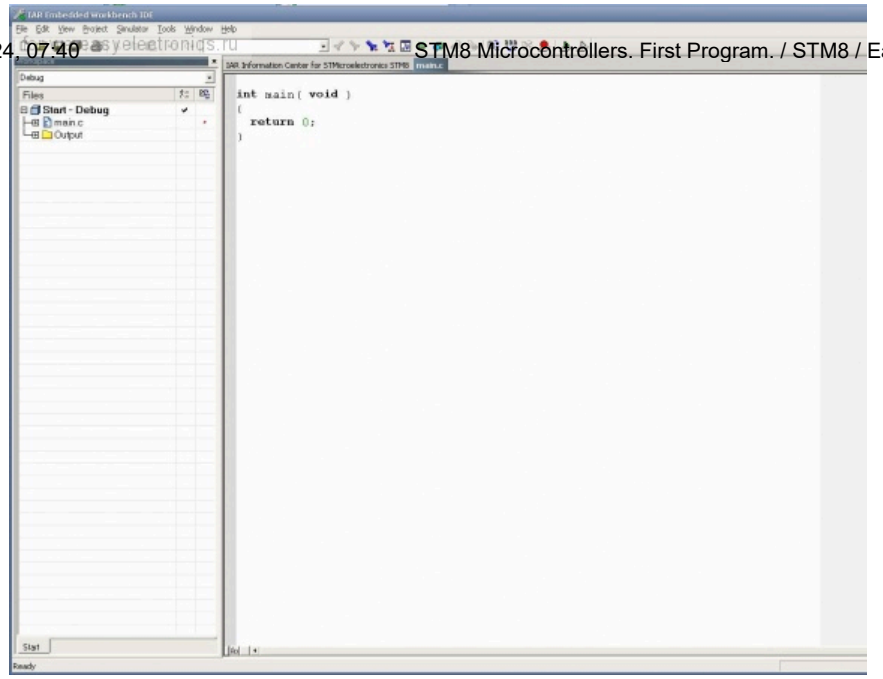
Do not neglect the ones located in the main program window on a yellow background. Particularly important are STM8 Getting Started and User Guide sections. They contain a lot of useful information about the development environment itself, as well as guides on C, C++ and assembler. In the Example Projects section, Captain Obvious has placed example projects.
Let's create a new project: go to the "Project" menu, where we select the "Create new project ..." menu item.



In the window that appears, select a template for the C language, after which we are asked to save our workspace - Workspace, with the extension *.eww. We save, a window pops up asking to save the project, we save it too. IAR projects have the extension *.ewp.



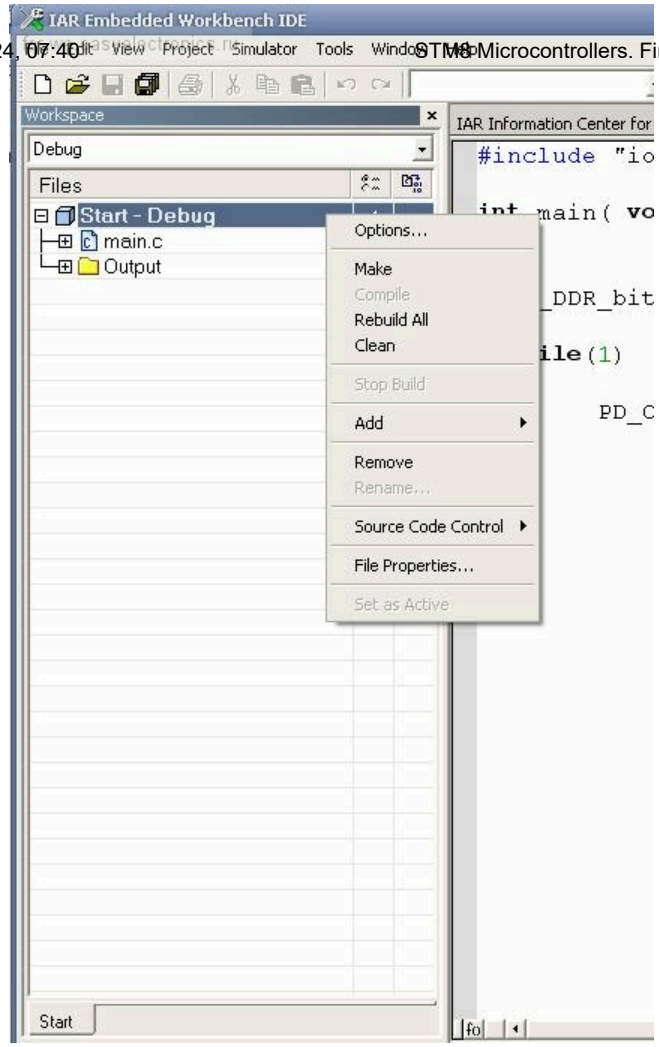
The following window appeared in front of us:



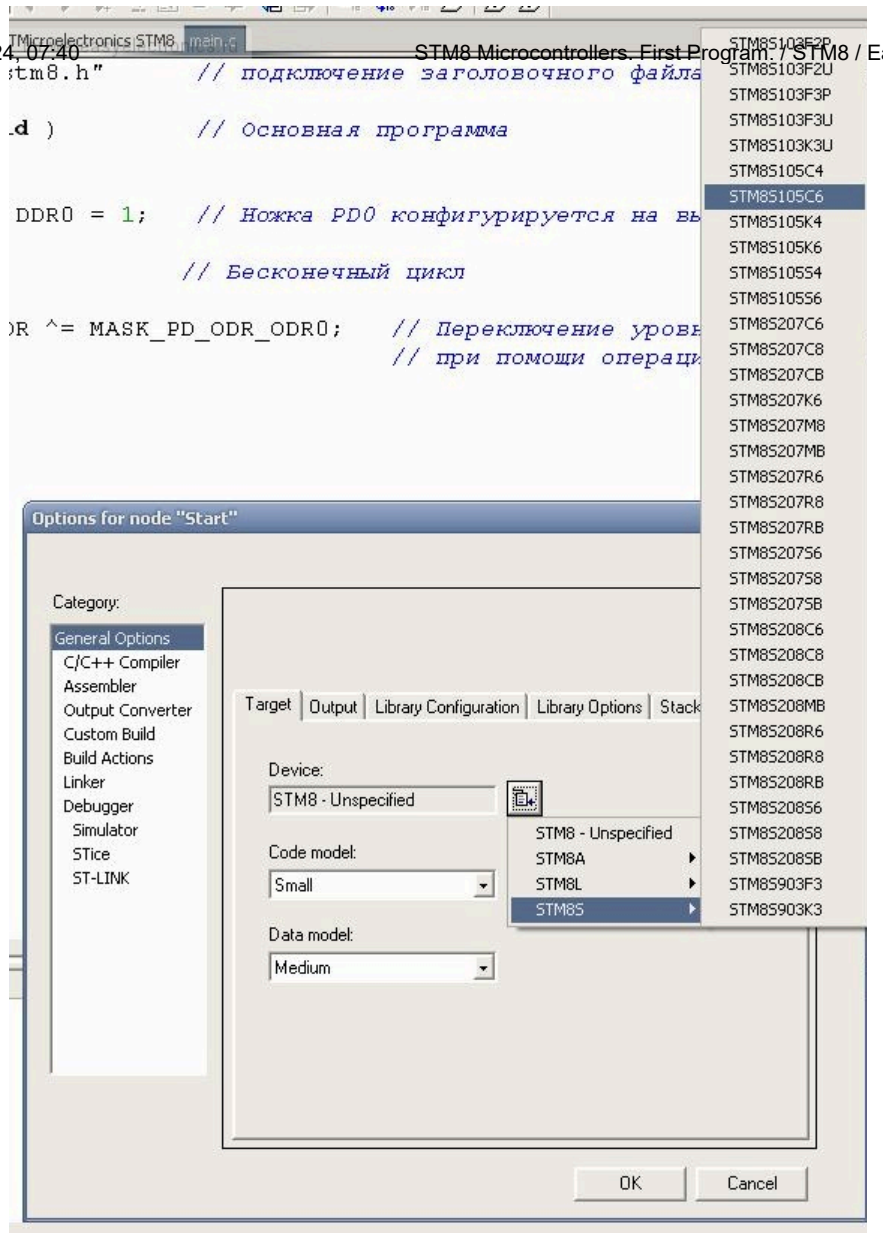
This is a template for an empty C file. The templates themselves are located in the folder %Path to IAR% IAR Systems\Embedded Workbench 6.0 Kickstart\stm8\config\template\project. If you have some templates repeated from project to project, you can create your own templates and select them when creating a new project.
Type the following code, then save the project:

```
#include "iostm8.h"      // подключение заголовочного файла с объявления

int main( void )         // Основная программа
{

  PD_DDR_bit.DDR0 = 1;   // Ножка PD0 конфигурируется на вывод

  while(1)              // Бесконечный цикл
  {
        PD_ODR ^= MASK_PD_ODR_ODR0;   // Переключение уровня напряжения
                                      // при помощи операции Исключающее
  }

}
```

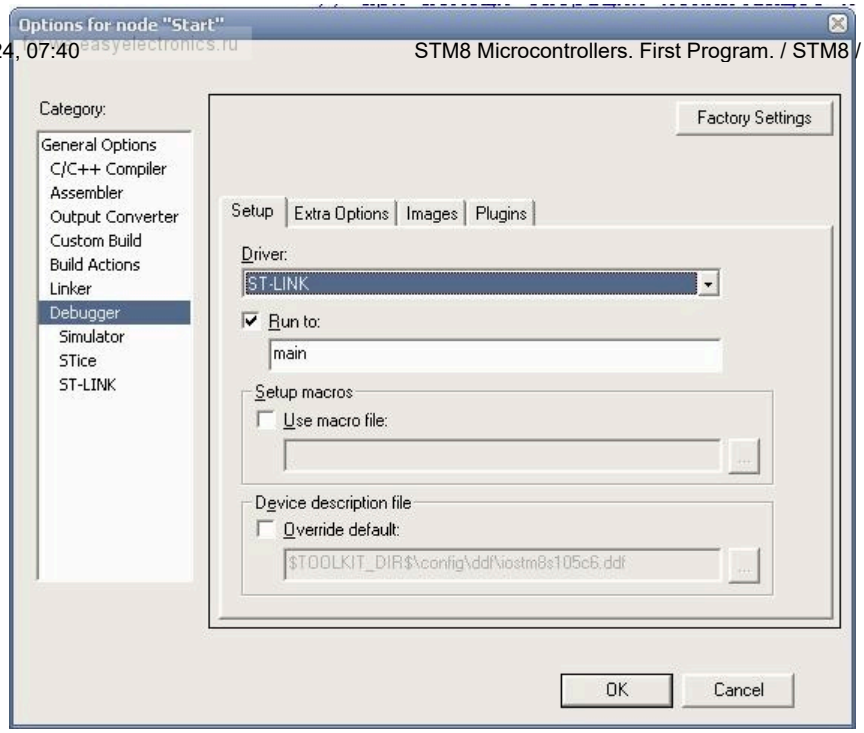
The program is short, each line is commented, so I will not provide additional explanations, if there are questions, I will answer them in the comments.
By default, our code will be launched in the simulator after compilation. Let's configure our project for execution in hardware. To do this, in the workspace window, select the context menu item "Options", and right-click on the project itself, and not on the files that are included in it. To achieve the same effect, you can press Alt + F7.



On the General page, Target tab, select the model of our controller: STM8S -> STM8S105C6.

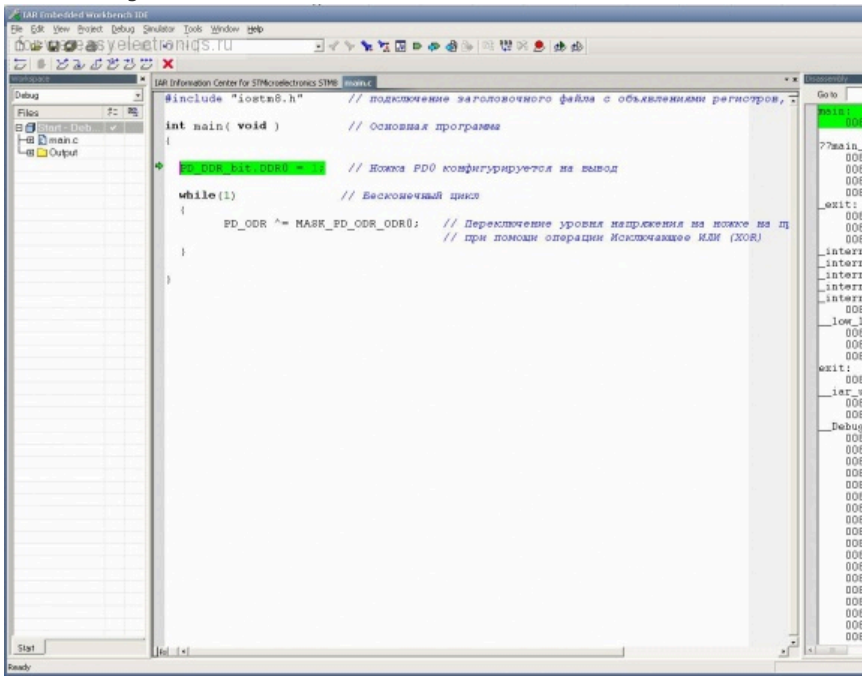


Select our debugger on the Debugger page, Setup tab: ST-Link.



Click OK. Now it's time to run our program.

On the main toolbar, click the "Download and Debug" button in the form of a green triangle, and if no problems arose, which IAR will not fail to inform us about, then our program will be written to the microcontroller and we will see the following window:



On the right we see the disassembler window, in the center - the main program window. The line highlighted in green will be executed in the next step. Step through the program using the "Step Over" button (F10), and we will see that the LED on the board is blinking.

Accordingly, the program is working correctly. You can exit the debug mode by clicking the "Stop Debugging" button in the form of a red cross. That's

all for today, and in the next article we will take a closer look at the input-output ports, as well as the Firmware Library for STM8.

Links:
STM website section dedicated to STM8S-DISCOVERY

STM8 , contest , microcontrollers , programming

+4 08 March 2011, 22:19 **Kalvenolt**

Comments (37)

RSS Collapse / Expand

> *The programmer works with all families of STM8 microcontrollers, and there are rumors that it will soon be hacked and will be able to flash STM32 as well.*

The opposite is more relevant) Many people already have STM32VL-Discovery.
By the way, is it really possible to break this board? At least part of the STM32VL-Discovery debug board gets into ST-Link territory.

0

Vga
08 March 2011, 23:44

> Tomorrow I'll post some photos - the board is milled in such a way that it's easy and convenient to break off, and the programmer and debug board are completely separated.

0

Kalvenolt
08 March 2011, 23:47 ↑

> As far as I understand, the STM32VL-Discovery board cannot be broken. The STM8L-Discovery board cannot be broken either.
> I will also say that the STM8L-Discovery is interesting because it has a 24-segment LCD installed.

0

citizen
09 March 2011, 00:36 ↑

> Yes, you can't break the board, but there are two jumpers, by removing which you can flash a third-party chip (I'm doing this now).
> I'm more interested in the board not for the indicator, but only because the installed controller consumes little.
> The 32nd one also can't be broken, but it also has jumpers, but I haven't tried flashing through it, I'll have to try. I have all three boards available...

0

ZiB
09 March 2011, 05:38 ↑

> As far as I can see, you can break the STM32 board, but the output will be STLink and a board fragment with parts that can be transferred to your design.

0

Vga
09 March 2011, 05:46 ↑

Thanks, I'm waiting for the continuation! Discovery and a couple of stones are lying around, but I haven't gotten around to putting them away anywhere yet

0

marvin_yorke
09 March 2011, 11:15

0

The continuation will be today or tomorrow. It all depends on the workload - today is a working day after all)

Kalvenolt
09 March 2011, 12:10

Hmm... Still, the crisis has had a strong impact on Atmel - people are seriously switching to other MCs... Or has the time of AVR really passed? 0

_YS_
09 March 2011, 19:35

People are thirsty for freebies! ST is aggressively climbing into this niche. And here there is only one type of struggle - price.

And judging by the IPO, Atmel shares have soared very high. Everyone should have such crises :))) The rise in prices for processors only shows this. 0

DIHALT
09 March 2011, 19:47

So the shares took off not because of the processors, but because of the memory and touchscreens, it seems. 0

Kalvenolt
09 March 2011, 19:50

Well, at their expense, yes. 0

DIHALT
09 March 2011, 19:53

I can't upload the program and debug it = ((In STVD + Resonans everything worked fine, I downloaded the 30-day IAR, after clicking "download and debug", the red LED on the STM8S-DISCOVERY lights up and gives the following errors: 0

1. An error occurred while retrieving GDI features: gdi-error [40201]: Can't access configuration database
2. Couldn't find STM8S105C6 in the list of supported MCUs.
3. Failed to load debugee:
D:\Users\Katbert\Desktop\STMicro\STM_8\IAR_Start\Debug\Exe\Pr_1.out

There are no errors in the program (building without debugging and flashing the stone is normal), the settings are all as in the article...

Katbert
April 13, 2011, 10:36 PM

Try to put IAR in another place, for example C:\IAR_STM8\ 0

Frolls
April 14, 2011, 5:04 PM

IAR version 1.20?
I would try reinstalling IAR first. 0

Kalvenolt
April 14, 2011, 5:54 PM

IAR Embedded Workbench for STMicroelectronics STM8, Evaluation version, v. 1.30

Installed on the disk where the OS is — D:\

Ok, I'll try. 0

Katbert
April 14, 2011, 20:52

No, it didn't work... I installed it on the C drive, where there is another OS (win7) from under the same OS. The same result. However, everything worked with XP, it 0

started up the first time, debugging started. But damn, I need IAR on win7.

Katbert
April 16, 2011, 10:32

Hurray! I had the same problem. It didn't take a day to solve it. You need to run IAR as an administrator))) 0

ISK
03 October 2011, 10:30

Thanks. I already thought that:
CFG->GCR |= CFG_GCR_SWD; /*disable SWIM interface*/
broke the debugger)) +1

velzor
November 21, 2014, 00:05

I don't understand where in IAR I can see the code size after compilation? 0

dcoder
July 22, 2011, 02:00

In the log, switching the filtering level to all. True, then a lot of stuff pours in there, including commands. 0

Vga
July 22, 2011, 02:58

And who will explain or point out to me "to read" why, having connected LEDs with 1 kOhm for "blinking", everything blinks on PD, but not on PB and PA (but it lights up). I don't have an oscilloscope at home, it's not clear 0

ltvch
September 11, 2011, 16:13

The question is partially removed. I messed up with the initialization. Delete the question))) 0

ltvch
September 11, 2011, 5:13 PM

Why can't you write in ASMA for STM? What about the code size in relation to AVR, no more than in assembler? 0

moon7
October 14, 2011, 6:31 PM

Because there is no point, you can write in C. 0

angel5a
October 28, 2011, 4:37 PM

It's being written little by little... There's an interesting assembler there, or rather the internal organization... +1

Deer
November 11, 2011, 16:19

That's the problem, little by little. I see the point in writing quickly, and then slowly optimizing as needed. But I don't see the point in writing any LED blinker little by little. 0

angel5a
December 20, 2011, 12:17

Is there any software for programming STM8 in Delphi? 0

titan
October 28, 2011, 4:00 PM

with ported VCL :D 0

angel5a
October 28, 2011, 4:37 PM

I have a stm8s-discovery board, installed IAR 1.3. I encountered a strange problem. The menu item "Project-Download and Debug" works, but I can't simply upload the firmware without debugging, because Project-Download-Download Active Application is not active. 0

mcsa
November 11, 2011, 12:11

Literal translation from IAR help: If the device does not support the commands of the Flash loader mechanism used in IAR Embedded Workbench, then these items will be inactive. Apparently, the only option left is to use debug for uploading. 0

Krakozebl
November 19, 2011, 11:47

Please give a link to normal drivers for stm8s-disovery, the system (seven) defines the programmer as "Unknown Device" and writes the error "This device has been stopped because it reported a problem. (Code 43)", and IAR writes "Wed Dec 21 11:40:56 2011: Failed to initialize communication with hardware: SWIM error [30200]: ST-Link connection error
". 0

jongame
December 20, 2011, 10:46

I suspect that the board is faulty. Try without drivers on another machine. The board is initially defined as a removable disk, on which a couple of URLs are located. 0

angel5a
December 20, 2011, 12:15

I won't look at STM8 until there is a GCC compiler for them. 0

rcdimon
December 20, 2011, 11:34

Write yourself, what does it cost you? For one thing, others will thank you too. :D 0

angel5a
December 20, 2011, 12:18

A couple of years of life?) 0

rcdimon
December 20, 2011, 12:21

does not work 0

selevo
07 November 2013, 16:45

It didn't work for me right away either, in the debugger the output honestly changed from 0 to 1 and back, but the LED still didn't light. The hardware was my own, not STM-Discovery. After AVR it never occurred to me that I needed to configure anything else, but it turned out I did: in the CR1 register specify that the output works in Push-pull mode. That is, for the given example add 0

```
PD_CR1_bit.C10 = 1; // Выход типа Push-pull
```

vkusleta
05 May 2016, 23:22

Only registered and authorized users can leave comments.