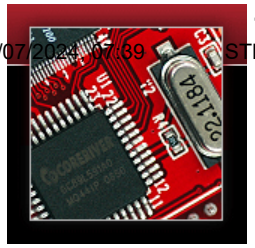


STM8L-Digital-to-Analog Converter (DAC) and Analog Signal Switching Unit (Routing interface)

STM8

After the input-output lines, the first thing I needed to try was the DAC. Since I don't need any complications, I will consider the simple mode of operation of the DAC.

Signal switch

Before moving on to the DAC block, it is necessary to consider one very important block – the signal switch (in my interpretation, in the original Routing interface (RI)).

This block switches signals between analog blocks and I/O lines.

I want to consider it first of all because the DAC output of microcontrollers in TQFP32 packages is not directly connected to the I/O line, and therefore before using the DAC, it is necessary to specify which line the DAC output should be connected to. Unfortunately, it is possible to connect only to one of three predefined lines.

Functional diagram of the block:

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

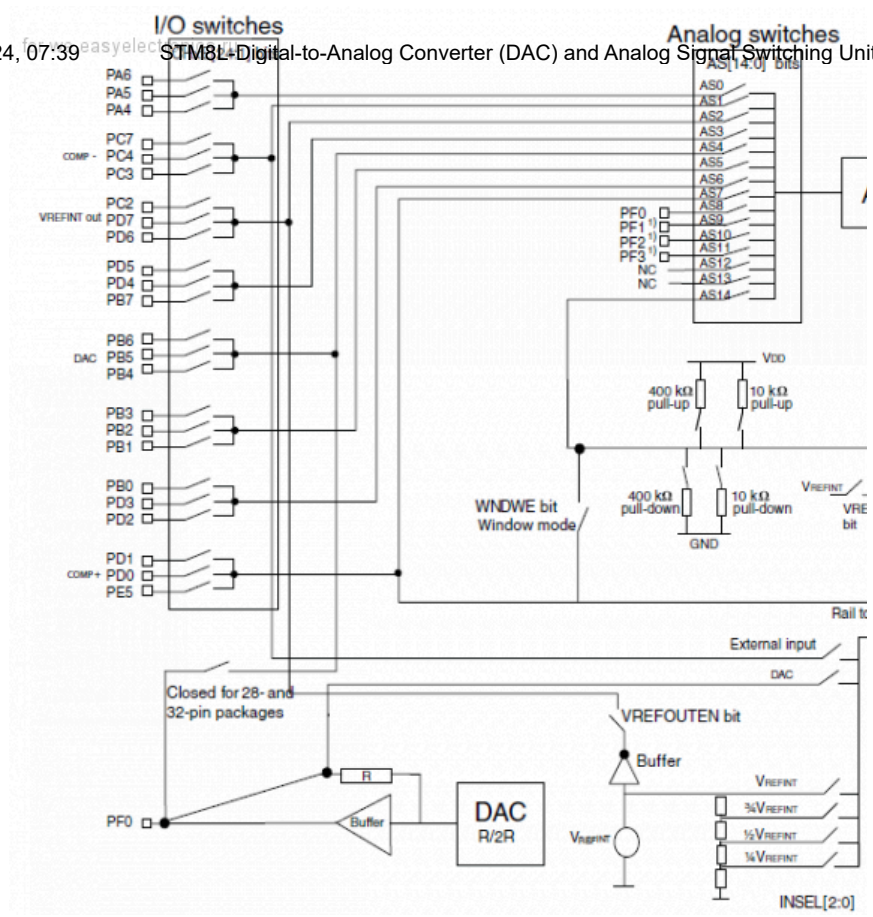
dmitrij999 → Capturing images from a USB camera using STM32 6 → STM32

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



According to the diagram, the I/O lines are grouped into eight groups of three lines each. Three lines PB4-PB6 are allocated for the DAC block.

On my board, I would like to use the PB4 line, it is very easy to do this, just turn on the corresponding switch:

```
RI->IOSR3 |= BIT(4);
```

However, I was unable to obtain an output signal. The first thought is that it is necessary to supply clocking, but in the Clock control (CLK) section describing the clock signal control unit, there is no word about this unit :(. After rereading the description several times, I thought that I had incorrectly configured the DAC itself and decided to check the switching unit on something simpler. The simplest was the reference voltage source. Having turned it on and connected it to the PD6 line (since the VREFOUTEN bit is located in the comparator register, I supplied a clock signal to it too), I was convinced that the switch was working and exactly as I thought. But the question arose as to why the DAC was not working, and here I was lucky. I mainly checked the DAC using a cyclic write to the DAC register of an incrementing eight-bit variable and did not comment out the code when checking the switching unit. How surprised I was when I decided to check what was at the DAC output, and there ... and there everything worked. As a result of the analysis, it turned out that for the switching unit to work, it is necessary to enable the comparator clocking. The first rake which I stepped on while studying, and I haven't really tried anything yet :(. Re-reading the documentation didn't help to solve this phenomenon. Therefore, before working with the switching unit, **it is necessary to enable the comparator clocking** :

```
CLK->PCKENR2 |= CLK_PCKENR2_COMP;
```

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

Well, now we can move on to the DAC.

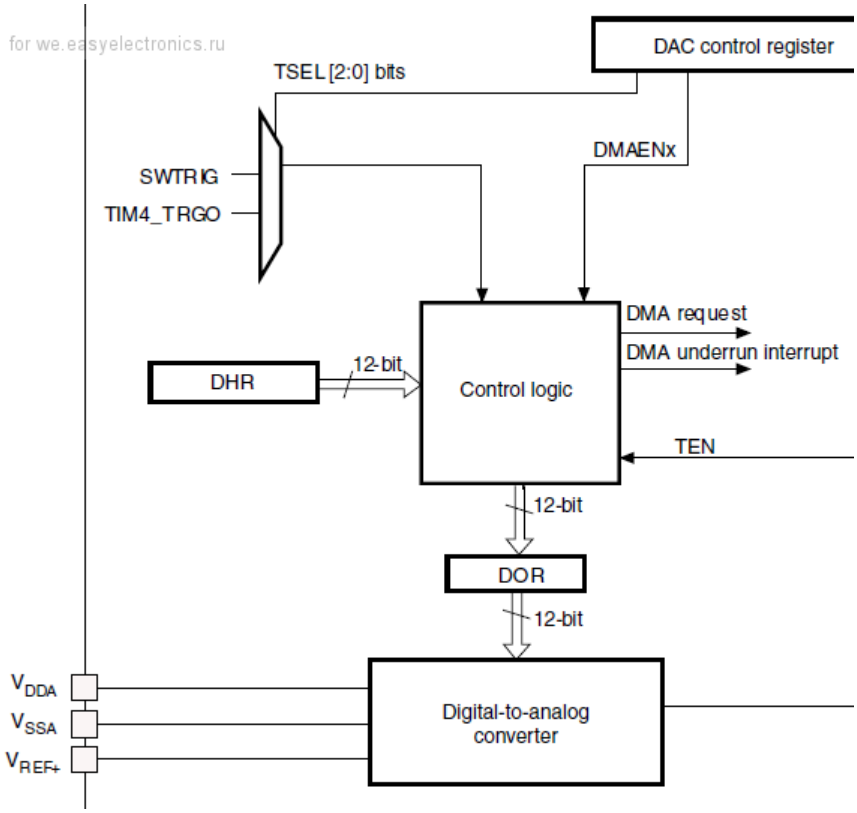
DAC

The STM8L15x microcontroller family contains one to two 12-bit digital-to-analog converters (DACs).

Key features:

eight- or twelve-bit operating mode
right or left justification in twelve-bit mode
synchronous refresh capability
direct memory access (DMA)
support multiple clock sources support
external reference voltage capability
auto-generation of triangular or "noise" signal (MD+ and HD only)
maximum signal settling time of 12 µs, with a small change in the output signal 1 µs

Functional diagram of the DAC block:



More details can be found in the RM0031 STM8L15x and STM8L16x microcontroller family manual.

For the DAC to work, you must:

enable clocking
configure the required operating mode

Enable module clocking:

```
CLK->PCKENR1 |= CLK_PCKENR1_DAC;
```

The output signal from the DAC can come not directly, but through a buffer amplifier. This allows you to use a lower-impedance load (5 kOhm), but reduces the output voltage range (from 0.2 to Vref-0.2 Volts), I am satisfied with these restrictions, therefore I will use a buffer amplifier.

At the moment, I am interested in software recording of values in the output register of the DAC, but it is impossible to directly write the desired value to the output register, you need to use one of the methods:

1) write the desired value to the corresponding storage register, and after one CPU cycle the value will be written to the output register

2) write the value and initiate the recording at the right time using a software trigger.

Such difficulties are only because the DAC module can operate in 8- and 12-bit modes, and in 12-bit mode, left and right alignment is possible, which are necessary for optimal operation with the direct memory access (DMA) block. Each of the modes has its own registers, i.e. if eight bits are enough for you, you write to just one 8-bit register.

I decided to start with the second mode, i.e. I will use a software trigger.

Well, based on the above, we configure the DAC:

```
DAC->CH1CR1 = DAC_CR1_TEN | DAC_CR1_BOFF | DAC_CR1_TSEL_SWTRIG;
```

and we allow work:

```
DAC->CH1CR1 |= DAC_CR1_EN;
```

It seems I haven't forgotten anything :)

A small demo:

```
/*
 * File: main.c
 * Date: 28.01.2011
 */

#include "main.h"

//-----------------------------------------------------------------------
void delay_ms(uint16_t time)
{
    volatile uint32_t i;

    while (time-- > 0)
    {
        i = 30;
        while (i-- > 0)
        {
        }
    }
}

//-----------------------------------------------------------------------
int main(void)
{
```

```
    uint8_t tDAC;
    CLK->PCKENR1 |= CLK_PCKENR1_DAC;
    CLK->PCKENR2 |= CLK_PCKENR2_COMP;

    RI->IOSR3 |= BIT(4);

    DAC->CH1CR1 = DAC_CR1_TEN | DAC_CR1_BOFF | DAC_CR1_TSEL_SWTRIG;
    DAC->CH1CR1 |= DAC_CR1_EN;

    tDAC = 0;

    while (1)
    {
        DAC->CH1DHR8 = tDAC++;

        DAC->SWTRIGR |= DAC_SWTRIGR_SWTRIG1;

        delay_ms(1);
    }

    return 0;
}
```

STM8 , DAC

+2 March 11, 2011, 06:51 **ZiB**

Comments (7)

RSS Collapse / Expand

You're talking about DAC, and here you already have "Functional diagram of the ADC block (medium density devices)" 0

Drunya
January 22, 2013, 12:26

Who doesn't? It's human nature to make mistakes. Corrected. Thank you. 0

ZiB
January 22, 2013, 12:41 ↑

Therefore, before working with the switching block, it is necessary to enable the comparator clocking: 0

Funny. But as far as I can see, the switching block is also used by the ADC. If you turn on the clocking of the ADC, and not the comparator, will it also turn on and output the DAC signal?

Vga
January 24, 2013, 10:55

As a result of the analysis, it turned out that for the switching unit to work, it is necessary to enable the comparator clocking. The first rake that I stepped on while studying, and I still haven't really tried anything :(. Re-reading the documentation did not help to solve this phenomenon. 0

Take the document RM0031 Reference manual
STM8L05xx, STM8L15xx and STM8L16xx microcontroller family
Open chapter 11 Routing interface (RI) and system configuration controller (SYSCFG)
Look at section
11.2.1 RI functional description
The RI registers can be accessed only when the comparator clock is enabled by setting the PCKEN25 bit in the CLK_PCKENR2 register. Refer to Section 9.14.5 on page 104.

Nothing personal. :) I just got the impression that ST documentation is quite informative, you just have to read it carefully.

Why and for what purpose it was done this way - I will not discuss. But everything is clearly described, in plain English.)

P.S. I have never worked with STM8L, I was just a little surprised by the problems with STM documentation. Finding the right place together with searching for RM took 8 minutes.

SeregaB
21 February 2013, 23:21

Katz beat me to it here . :) 0

SeregaB
21 February 2013, 23:39

I apologize to ZiB, I didn't look at the topic creation date. Perhaps this really wasn't in RM then. 0

SeregaB
22 February 2013, 00:41

Yes, it happens. The article itself was even earlier 28-01-2011:
ziblog.ru/2011/01/28/stm8l-tsifro-analogovyiy-preobrazovatel-tsap-i-blok-kommutatsii-analogovyih-signalov-routing-interface.html
And at that time it was not.
And their documentation is good in my opinion. Of course there are minor shortcomings. 0

ZiB
22 February 2013, 04:39

Only registered and authorized users can leave comments.