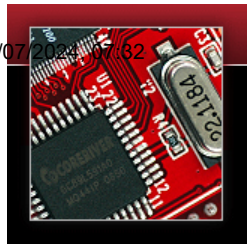




# STM8L Monitoring Supply Voltage Without Using ADC

STM8



**STM8L** microcontrollers are designed primarily for battery-powered devices. Therefore, it is often necessary to monitor the battery discharge level. This can be done using the ADC . If the ADC is fully occupied with the main task, then power monitoring can be entrusted to the **Programmable voltage detector (PVD)** system .

Usually, **PVD** is used to determine the low battery threshold, after which certain actions are taken until the power is completely lost.
We will use **PVD** to display the current battery charge on the LED scale in the range of **1.7 ... 3.05 V** .

For work, we will take the STM8L-Discovery board with the **STM8L152C6T6** microcontroller .
Carefully remove the LCD indicator and put it aside.

To connect the LED scale, we will need the entire port **B** . Instead of the LED scale, you can simply take 8 LEDs. We will connect their cathodes to the common wire, and their anodes to the outputs **PB0 ... PB7** of port **B** through 2K resistors.

We will solder a socket for connecting an external power source to **the 3.3V** and **GND** pins on the Discovery board. (We will not connect the external power source yet!) That's all the preparations.

## 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


**The program is written in IAR without using standard libraries:**

We will test **PVD** at a maximum frequency of 16 MHz;
We configure port **B** to output;
We switch port **B** to push-pull mode
We start **PVD by setting the PVDE** bit in the **PWR_CSR1** register :

```
CLK_CKDIVR = 0x00;
PB_DDR = 0xFF;
PB_CR1 = 0xFF;
PWR_CSR1_bit.PVDE = 1;
```

Inside the infinite loop we start measuring the supply voltage:

```
while (1)
{
     uint8_t Port_B = 0x01;  // 0 бит порта B всегда в "1" - показывает

     for (uint8_t n=0; n<=6; n++)
     {
       PWR_CSR1_bit.PLS = n;  // последовательно устанавливаем пороги  P
       Delay(4);      // задержка > 50 uS перед чтением  PWR_CSR1_bit.PV
                      // флага готовности, к сожалению, нет
       if ( PWR_CSR1_bit.PVDOF == 0 )  // порог ещё не достигнут
       {
         Port_B |= (1<<(n+1));   // записываем "1" в соответствующие пор
       }
       else break; // когда порог превышен, прерываем цикл
     }
```

**Vga** → ROPS (Rem Object Pascal Script) - compiled 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

```
    PB_ODR = Port_B;  // загораются светодиоды, соответствующие порогу
```

First of all, we write "1" to 0 bit of port **B.** When the supply voltage rises to 1.7 V, the microcontroller will start working and the LED in the least significant digit will light up.

Now in the cycle, we will sequentially raise the threshold from 0 to 6 in the **PWR_CSR1 register PLS** bit . The thresholds correspond to the following voltages:

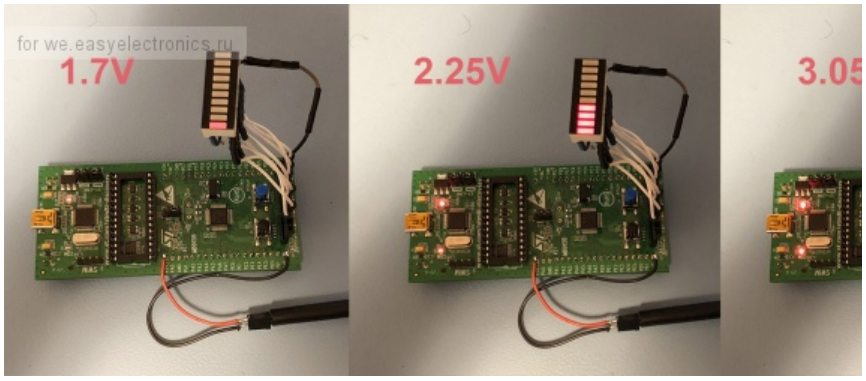
**PLS[2:0]**: PVD level selection

These bits are set and cleared by software.
000: PVD threshold = 1.85 V typ.
001: PVD threshold = 2.05 V typ.
010: PVD threshold = 2.26 V typ.
011: PVD threshold = 2.45 V typ.
100: PVD threshold = 2.65 V typ.
101: PVD threshold = 2.85 V typ.
110: PVD Threshold = 3.05 V typ.
111: Threshold = PVD_IN input pin

If the **PVDOF** bit in the **PWR_CSR1** register is equal to "0", this means that the supply voltage is above the threshold and "1" is written to the corresponding bit of port **B.** When the threshold is above the supply voltage, **the PVDOF** bit in the **PWR_CSR1** register will be equal to "1" and the cycle will be interrupted. Thus, the enabled bits of port **B** will show the current supply voltage. Before reading the **PVDOF** bit , a delay of > 50 uS is required so that the threshold has time to set. Unfortunately, there is no ready flag. The delay of > 50 uS is set empirically.
In order not to slow down the program, the delay can be organized on the timer. The full text of the program for IAR is attached.

We flash the program - 7 LEDs should light up, which corresponds to a level of **2.85 ... 3.05V** . This voltage comes to the microcontroller when powered from USB. In this example, we do not touch the remaining bits in the **PWR_CSR1** register , responsible for interrupts.

Disconnect the USB cable and connect an external power source. Smoothly increase the voltage from 0 to 3.3V. The photo clearly shows how the circuit works:




STM8L, STM8L-Discovery, battery discharge , indicator

+6 13 March 2019, 00:09 **CreLis** 1

Files in the topic: PVD_STM8_L.zip

## Comments ( 2 )

RSS Collapse / Expand

This will probably sound like "found something to find fault with", but it hurts me to look at a piece of code with completely chaotic indents. I would like to at least see normal code formatting in articles

0

**anper**
March 16, 2019, 7:41 p.m

Trimmed it. Thank you!

0

**CreLis**
March 16, 2019, 10:12 p.m

Only registered and authorized users can leave comments.