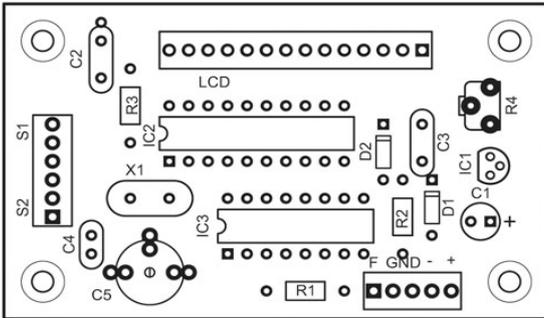


# An ATtiny2313 Frequency meter

Written by Vladimir Mitrović - Last Updated Tuesday, 04 June 2013 13:37

```
(function(i,s,o,g,r,a,m){['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)})(window,document,'script','//www.google-analytics.com/analytics.js','ga'); ga('create','UA-41457682-1','svet-el.si'); ga('send','pageview');
```



The ATtiny2313-based frequency meter shown in Figure 1 can measure frequencies from 1 Hz to almost 10 MHz with a 4 to 5 digit accuracy. The accuracy depends primarily on the exact frequency of the X1 crystal, which can be fine-tuned by the variable capacitor C5. A 74HCT4520 (IC3) is used as a digital low-pass filter, which signals the microcontroller if the measured frequency exceeds the upper limit. A two line LCD display is used to show the result. Therefore, the frequency and the period can be displayed simultaneously.



a)



b)



c)

The input signal should be TTL or CMOS compatible, with an amplitude of at least 2 Vpp. Diodes D1 and D2 limit the input voltage if its amplitude exceeds 5 Vpp. S2 is used as a display-hold button.

## How we can measure high frequencies

We can measure high frequencies by counting input pulses within a precisely-defined time interval. In its simplest form, it can be done as shown in this example:

# An ATtiny2313 Frequency meter

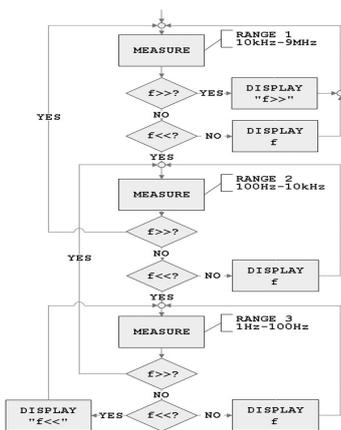
Written by Vladimir Mitrović - Last Updated Tuesday, 04 June 2013 13:37

---

Counter1 = 0 Config Timer1 = Counter , Edge = Falling Wait 1 Stop Timer1

The 16-bit Timer/Counter1 is configured as a counter, counting pulses on the T1 input pin. The counting starts as soon as the counter has been configured and stops after 1 second. If the initial value in the Counter1 register is 0, as is the case here, its value after one second will match the frequency of input pulses. Very simple indeed, but such a simple program has two limitations:

-



Counter1 is a 16-bit register and can only count up to 65535, at which point it overflows and starts counting again from 0;

- The Wait command is not accurate enough for precise measurements.

Therefore, an assembler routine has been written which lasts exactly one second. Apart from being accurate, this routine also counts the number of Counter1 overflows. The final result is a 32-bit number and is accurate to +/-1 digit:

$$\text{frequency} = \text{number\_of\_overflows} * 65536 + \text{Counter1}$$

The highest frequency that could be measured as described above is 232, i.e. about 4 GHz. Such a high frequency cannot be measured with this microcontroller. With a 20 MHz crystal, the microcontroller can measure frequencies up to 10 MHz. It should be noted that the microcontroller itself cannot accurately measure frequencies that are beyond the upper frequency limit. For example, a frequency of 10.2 MHz will be measured as 9.8 MHz. This error is caused by the way the microcontroller detects level changes at the input pin and it cannot be detected by software.

## An ATtiny2313 Frequency meter

Written by Vladimir Mitrović - Last Updated Tuesday, 04 June 2013 13:37

---

An ATtiny2313 Frequency meter

2012\_AVR\_UK\_185



[Continue reading](#)

---

## Shop area

