Single chip software defined instrumentation for educational purposes

Jiří Hladík

Dept. of Measurement, Czech Technical University, Technická 2, 166 27 Praha, Czech Republic

Abstract. The aim of this paper is to show features of modern microcontroller units since they can be used in education and effectively can replace some of the laboratory instrumentation. Software defined instrumentation is often discussed topic but with capability of nowadays microcontrollers it can be done on single chip maintaining the software facility and keeping the performance good enough. One of the real implementations is described in text.

Keywords

Single chip, software defined instrumentation, signal generation, oscilloscope

1. Introduction

Electro technical students need to have pretty good background of usage of laboratory instrumentation. Unfortunately it is not always possible to have even oscilloscopes for everyone. So the basic idea is to bring simple platform allowing students to work on their own, take the instrument home and deepen their electro-technical knowledge.

This kind of instrument should embed all the laboratory instrumentation which can come in mind from oscilloscope, generator, logic analyzer, counter and others to pretty complex devices like impedance analyzations, regulators or PWM controllers. All described can be a part of software defined instrument (SDI).

2. State of the art

In the past, there was tendency to do virtual instrumentation using classical standalone laboratory instruments or some other embedded instruments controlled from PC to make instrument for special purpose. It is still necessary for high performance measurement. But for common measurement it is possible to embed everything into one device.

Many manufactures already offer their combined instruments or SDIs. The well-known are RedPitaya, LabNation, Analog discovery. Their products contains mainly FPGA, DAC and ADC. These product are achieving quite good parameters and works out of the box. Similar approach was used for definition of single chip SDI.

As the single chip SDI has to run on single MCU taking care about all things it is necessary to pay attention to implementation of all of the parts not to waste processing time. It is also good keep easy transferability between different MCUs.

Before we go next to describe two main features of such instrument let’s see figure 1 which is showing what is inside of MCU and which blocks are commonly used.

3. Oscilloscope

The ADC in MCU are not really fast so the instrument is more for measurement in low frequencies. But the ADC is not the only part of oscilloscope. Trigger and its implementation has much bigger impact for speed and detection of the events. To achieve really single chip SDI trigger must be implemented in SW.

All MCUs are able to use timers and DMA to transfer data so the only payload is finding the trigger event. This is not necessary to do all the time and check every sample but it can be done once per several samples thanks to circular buffer with margins as show in figure 2.

Trigger event is checked time by time and exact trigger event is found when all data are sampled. This is simply done by checking the data in post-process and shifting the
pointer of the trigger event from the time when it was detected to the right time of the event. It is necessary to have margins of buffer big enough to store data for the shift. The only issue of this implementation might be missing of short events, however events shorter then roughly 10 samples are not properly sampled and we cannot clearly recognize what has happened.

Fig. 2. Circular buffer composition and trigger

Some of the MCUs embeds analog watchdog (AWG) which can be used to trigger interrupt when analog value is out of some range. Thanks to this it is possible to achieve zero MCU payload while sampling. On the other hand the time from generating interrupt to stop sampling can be even longer then in pure SW implementation so the margins of circular buffer are still needed.

4. Generator

The state of the art waveform generators are using DDS generators. Such a generators usually use external HW with long signal in flash, phase counter and many other things. Thanks to it DDS can set frequency very precisely but requires a lot of computational. For sure DDS cannot be done in single chip SDI because of huge consumption of computational power.

There are some other alternative solutions how to deal with generation of the signal. One of them which was implemented in this case can achieve zero payload and precise frequency setting. It uses DMA and timers similarly like in oscilloscope but in reverse order. Unfortunately timer can divide MCU clocks only by whole numbers therefore setting of the frequency is very inaccurate. To improve the resolution of frequency some other method has to be employed. One of them can be select right length of the buffer. This method enables you to achieve fraction division of clock source. Used signal flow shown in figure 3.

![Signal flow in generator](image)

From above shown signal flow we can find equation for output frequency $f_{out}$

$$f_{out} = CLK \cdot \#SMP / DIV$$

Where $CLK$ is clock frequency of MCU which is in this case set to 72MHz. $DIV$ is divider of clock and $\#SMP$ is number of samples.

To achieve target output frequency we need to minimize frequency error which is given by:

$$e = \left| f_{target} - \frac{CLK \cdot DIV}{\#SMP} \right|$$

Now the problem of setting the right frequency has split into problem of finding two numbers (signal length and clock divider). The issue of finding two parameters is usually not a simple task so some optimization method has to be used. The heuristic function for this optimization is the absolute value of the frequency error which we are trying to minimize. Optimization method is described in pseudocode below.

```
while error > 0.01Hz or bufferLength > 1/4 of maxBufferLength
    calculate divider and error
    if error < lastError
        remember divider and buffLength
    end
    decrease bufferLength
end
set bufferLength, samplingFreq and signal
```

Fig. 4. Optimization pseudocode for buffer length

In the beginning maximum signal length is used and appropriate clock divider is estimated. From the clock divider, we can calculate real generating frequency and error. If the error is higher than highest acceptable error then signal length is decreased and process starts again. If the exact numbers are not found then the best of them is used achieving the best possible frequency setting.

Some cut off conditions must be used to have reasonable results. It make no sense to set the frequency with accuracy >50ppm when 50ppm clock oscillator is used. We also don’t want to reduce signal length to e.g. 5 samples since it will not be the signal we want anymore. System implemented in this project have clock source 72 MHz and signal length 1k samples. The cut of conditions are frequency error <0.01 Hz and minimal signal length 250 samples. The outputs from the optimization algorithm and frequency error can be found at figures 4-7.
You can see different impact of used optimization for different frequencies. Change of divider has small impact for very low output frequencies therefore algorithm can achieve error 0.01 Hz and maintain the buffer relatively long. In our case this is valid for frequencies below 500 Hz. For mid frequencies both cutoff conditions become important and sampling frequency with buffer length start to vary over the full range of set boundaries. This is for frequencies up to 2 kHz. For higher frequencies it is not possible to use maximal length of buffer due to limited sampling frequency however the error is still acceptable. Frequencies over 10 kHz uses very small number of samples and optimization is not able to do its job properly and error is similar to error without optimization.

In contrast with DDS, the method described above is not introducing any additional jitter. It is thanks to deterministic signal generation.

5. Conclusion

Single chip software instrument was defined in this paper. Some constrains were discussed to achieve best possible performance. One of the possible implementation was described and this device is currently used by students at many laboratory projects. Complete system shown realized on Nucleo board with analog expansion board to interface signal so MCU and GUI for PC are shown in figures 9-11.
Fig. 9. Window with oscilloscope GUI

Fig. 10. Window with generator GUI

Fig. 11. Realization of single chip software defined instrument

Acknowledgements

Research described in the paper was supervised by doc. Ing. Jan Fischer, CSc., FEE CTU in Prague and supported by laboratory of videometry.

References


About Authors...

Jiří HLADÍK was born in Turnov, Czech Republic in 1990. He received his master’s degree in 2015 from Cybernetics and robotics at Czech Technical University. At present he continues in his PhD studies at the same university focusing on precise measurement of position.