http://doi.org/10.35784/iapgos.3258 *received: 8.11.2022 | revised: 9.12.2022 | accepted: 15.12.2022 | available online: 30.12.2022*

FREQUENCY-TO-CODE CONVERTER WITH DIRECT DATA TRANSMISSION

Piotr Warda
Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Automation and Metrology, Lublin, Poland

Abstract. *Signal frequency is one of the most popular information carriers in measurement technology. A circuit called a frequency-to-code converter is used to convert frequency into numerical values. Frequency-to-code converters usually operate in the mode of a recorder of a certain number of periods of the signal under test. Often it is a microcontroller-based circuit that uses built-in memory to collect measurement data. The size of the memory limits the measurement capabilities of the transmitter. The paper presents the development of a frequency-to-code converter that sends measurement data directly to the host computer without collecting the results in the RAM of the converter. The working algorithm of the transmitter is presented. The results of measurement experiments carried out for sample signals of constant and variable frequency are presented.. The metrological analysis of the results is presented.*

Keywords: frequency, digital period measurement, frequency-to-code converter, microcontroller

PRZETWORNIK CZĘSTOTLIWOŚĆ-KOD Z BEZPOŚREDNIĄ TRANSMISJĄ DANYCH

Streszczenie. *Częstotliwość sygnału jest w technice pomiarowej jednym z popularniejszych nośników informacji. Do konwersji częstotliwości w wartości liczbowe używa się układu zwanego przetwornikiem częstotliwość-kod. Przetworniki częstotliwość-kod zwykle pracują w trybie rejestratora określonej liczby okresów badanego sygnału. Często jest to układ oparty o mikrokontroler wykorzystujący wbudowaną pamięć do gromadzenia danych pomiarowych. Rozmiar pamięci ogranicza możliwości pomiarowe przetwornika. W artykule zaprezentowano opracowanie przetwornika częstotliwość-kod przesyłającego bezpośrednio dane pomiarowe do komputera nadrzędnego bez gromadzenia wyników w pamięci RAM przetwornika. Zaprezentowano algorytm pracy przetwornika. Przedstawiono wyniki przeprowadzonych eksperymentów pomiarowych dla przykładowych przebiegów o stałej i zmiennej częstotliwości. Przedstawiono analizę metrologiczną wyników.*

Słowa kluczowe: częstotliwość, cyfrowy pomiar okresu, przetwornik częstotliwości na kod, mikrokontroler

Introduction

Converting a physical quantity into frequency is one of the more common ways of converting information [4, 13]. A variable frequency signal is seen as an attractive transmission medium. It allows the transmission of information in an environment of increased electromagnetic interference [11]. In addition, due to its high resistance to signal amplitude attenuation, compared to a voltage signal, it allows information to be transmitted more easily over longer distances [10]. In the remainder of this paper, a signal in which information is transmitted by the value of its frequency will be referred to as a frequency signal.

Decoding the information transmitted by the frequency of the signal can be carried out by various methods. One of them is to apply a frequency-to-voltage converter [3, 7] and further transform the voltage signal using a standard measurement card containing an analog-to-digital converter (ADC). The result is a standardized data set obtained at uniform intervals, which can be analyzed using standard digital signal processing methods. However, the use of a data transmission channel with frequency carrier of information with intermediate additional processing with an ADC has some features that may be disadvantageous in some cases. The first is the need to verify that the introduced additional processing error will not have too great an influence on the total information processing error in the measurement system. The second feature is the increased complexity of the system structure. A larger number of components requires an increased amount of time to get the system up and running, and creates an additional risk of faults in the introduced intermediate data processing module.

One alternative to indirect conversion into voltage is to process the frequency of a signal by digitally measuring consecutive, information-carrying periods of the signal. Digital processing of period values [8] is widely known in the metrology literature. Correctly carried out measurement allows to obtain a very accurate result [14]. The attractiveness of the period measurement is also increased by the fact that the cost of the created system for direct analysis of the frequency signal is not high. The elements required to implement the measurement (counter, edge detection circuit, stable reference signal generator [1]) are contained in most microcontrollers, which are often used in control and measurement applications. Of course, it is not always possible to use any simple microcontroller for a particular purpose. What is important are the microcontrollers resources, its data transmission capabilities and the value of frequency of the microcontrollers clock signal.

1. Principle of measurement

The periods of a frequency signal are usually determined successively.

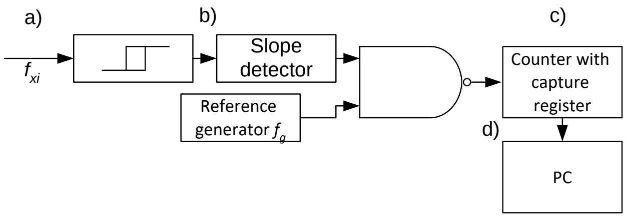

Fig. 1. An exemplary structure showing the processing of successive signal periods

Each period measurement is treated as a single, completely independent of the previous measurement [10]. Typically, the frequency signal, before being processed in the digital circuit, is formed in the Schmitt trigger [6] in order to adapt the signal amplitude to the requirements of the counter circuit used. The second reason for using a flip-flop is the need to form a square wave [5], allowing for precise detection of the boundaries of the processed period [3, 10].

An exemplary structure presenting the method of digital processing of the signal period is shown in figure 1. The waveforms showing the operation of the structure at more important points are shown in figure 2. The formed variable frequency waveform is fed to the signal edge detection circuit (Fig. 1b and Fig. 2b). At the selected edge (rising or falling), this system generates information about the end of the current period and the beginning of the next T_{xi} period of the frequency signal. In the defined limiting time moments (Fig. 1c and Fig. 2c), readings are made of the register of the counter continuously summing up the pulses from the reference signal generator. The difference between the successive read readings N_i of the counter is proportional to the length of a given period (Fig. 1d and Fig. 2d). The value of the difference is calculated from the dependence:

$$N_{xi} = N_{i+1} - N_i + C_m \cdot C_{Ov} \tag{1}$$

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
Utwór dostępny jest na licencji Creative Commons Uznanie autorstwa – Na tych samych warunkach 4.0 Miedzynarodowe.

where: C_m – maximum number of pulses that can be registered by the frequency-to-code converter counter and C_{Ov} – number of registered overflows of the counter.

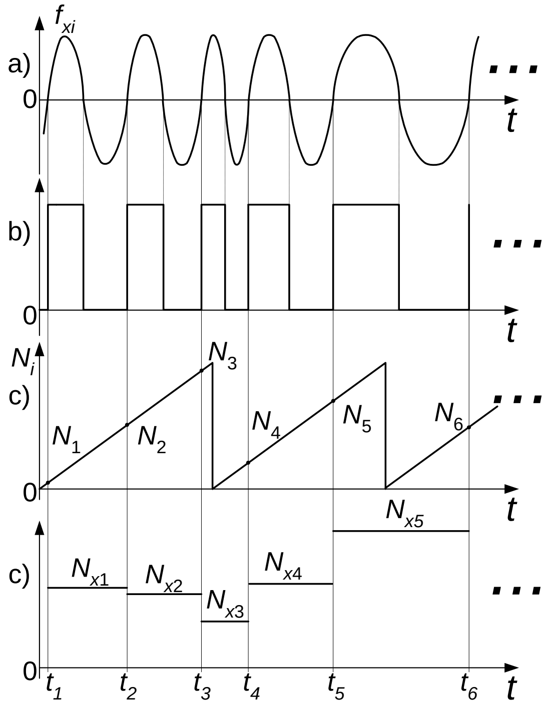

Fig. 2. Waveforms at selected points in the processing structure of successive periods of the signal

The period value is calculated as the product of the clock generator period T_g and the calculated difference of states for a given period T_{xi}:

$$T_{xi} = N_{xi} \cdot T_g \tag{2}$$

The system converting the signal frequency into numerical values is called the frequency-to-code converter [9] and marked with the symbol *f*/*N*.

2. The problem to be solved

During digital conversion of a period into numerical values, there is a problem of where to save large-volume measurement data. As it has already been written, microcontrollers [1] are often used for the construction of the *f*/*N* converter, which usually contain RAM memory in their structure. The advantage of the built-in memory chip is its direct accessibility and high speed of operation. Unfortunately, this memory is usually not very large, so it allows you to save only small sets of numerical data. Some microcontrollers allow you to add external memory. Unfortunately, also in this case the possibilities are small compared to the needs of using a possible longer measurement time. Much greater possibilities of data recording are offered by personal computers, usually controlling the measurement process.

The article presents an example of a frequency-to-code converter design using a personal computer for data storage.

3. Assumptions adopted for the transducer project being implemented

It was assumed that the developed *f*/*N* converter will be implemented with the use of the AVR family microcontroller. A standard UART module included in the microcontroller will be used for data transmission. The data will be received by a personal computer (PC) via the USB interface. A dedicated converter circuit will be used for the USB / RS232 conversion. It is obvious that the *f*/*N* converter system should not be burdened with unnecessary calculations that will slow down the measurement procedure. For this reason, instead of setting the measurement time, it was assumed that the number of T_{xi} period measurements to be performed by the *f*/*N* converter would be declared. It was assumed that the information about the number of measurements to be performed would be provided by the user via a PC.

4. Implementation

The ATmega32 microcontroller was chosen for the practical implementation of the *f*/*N* converter, which sends data to the computer via the FT232RL converter [16]. The integrated circuit 74LS132 [17] was selected to form the processed waveform of variable frequency into a rectangular shape with voltage levels adjusted to the requirements of the microcontroller. It was assumed that the test frequency signal would be produced in the Agilent 3322A generator [15]. The structure of the stand for the experimental verification of the developed frequency-to-code converter is shown in figure 3.

Fig. 3. The structure of the system for the implementation of the measurement experiment

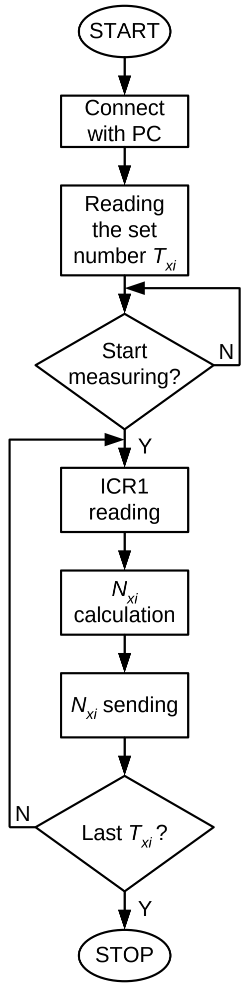

Fig. 4. The f/N converter work algorithm

The software of the *f*/*N* converter was created in the C language with the use of the AVR-GCC compiler [2]. The application work algorithm is shown in Fig. 4. After communicating with a PC, the transducer program waits for the determination of the required number of T_{xi} measurements. It then goes into a state of waiting for a frequency signal processing command. After receiving the expected command, the configuration of the T1 counter takes place, forcing the operation in the mode of capturing the instantaneous value of the totalizing register of the counter. The read states are stored in the ICR1 register at the moments of the rising edge of the frequency signal (Fig. 2a and Fig. 2b). After the configuration, the counter is started and subsequent readings of the T1 counter are registered. The stored numerical values are then used to compute numerical values N_{xi} (1) representing successive periods T_{xi}. The N_{xi} numbers are sent to a PC to be stored in a file with measurement data for further processing and analysis.

5. Results of experiments

The data obtained in the experiment was analyzed with the use of proprietary software developed in the LabVIEW [12] programming environment of the National Instruments company. It was assumed that for the initial evaluation of the *f*/*N* converter operation, the software will present the evaluation in a way depending on the nature of the signal. For a fixed frequency waveform, a histogram of the measured frequencies will be presented. For a variable frequency waveform, a portion of the stored waveform will be presented and an error curve showing the relationship between the two errors. The first error is the value obtained by theoretical considerations [10]:

$$\delta_{\Sigma} = \frac{T_g}{T_{xi}} + 0.5\left(1 - \frac{\sin \pi F_x T_{xi}}{\pi F_x T_{xi}}\right) \tag{3}$$

where: F_x – frequency of the frequency signal value change.

The second error is the value obtained from the experiment. It is the difference between the set frequency signal and the reconstructed frequency signal [12]:

$$\delta_F = 0.5\left|\sin 2\pi F_x t - \sin 2\pi F t\right| \tag{4}$$

where: F – frequency of the change in the value of the frequency signal reconstructed from the measurement data.

The lower range of frequencies processed by the *f*/*N* converter working in the basic operating mode limits the capacity of the counter of the frequency-to-code converter.

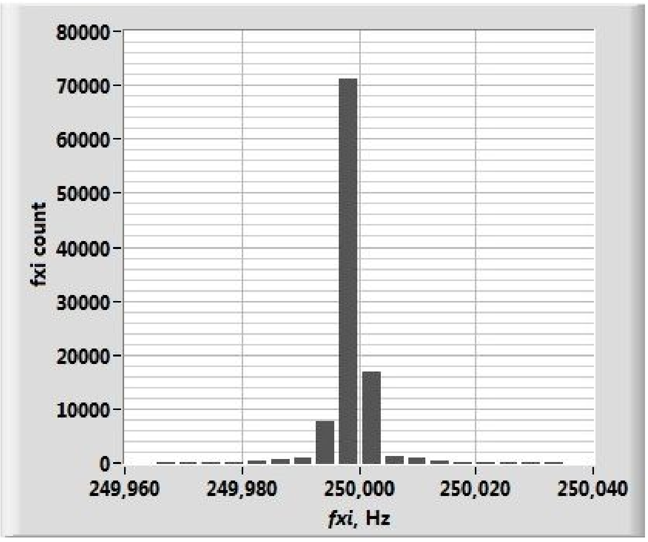

Fig. 5. Histogram of 100,000 frequency measurements of the frequency signal for a test signal with a frequency of 250 Hz

The discussed converter uses a 16-bit counter, and consequently the highest readable state is the number 65535. The period of the clock generator for the 16 MHz clock signal is 62.5 ns. As a result, the minimum measurable frequency, being the reciprocal of the product of the maximum counter state and the period of the clock generator, is approximately 244.144 Hz. To check the operation of the counter in the lower frequency range, a test signal with a frequency of 250 Hz was used. Figure 5 shows a histogram of the counter states after recording 100,000 periods of a frequency signal with a constant frequency of 250 Hz. The histogram shows that the read frequencies are very close to the set value in the test signal. The relative error of frequency measurement is within ± 0.014%.

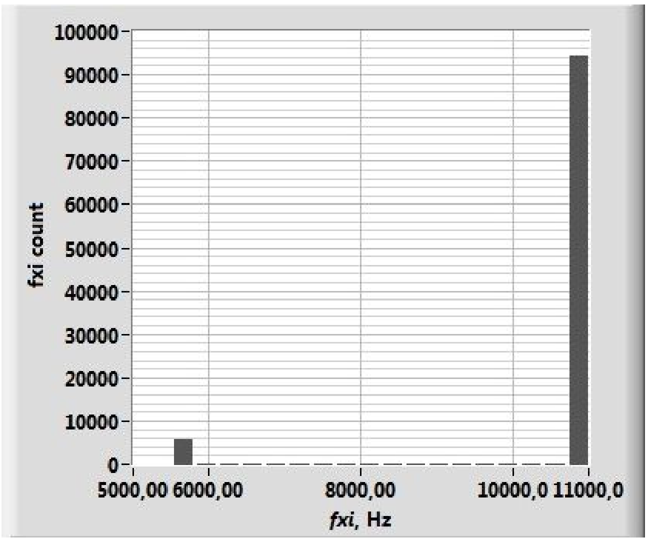

Fig. 6. Histogram of 100,000 frequency measurements of the frequency signal for a test signal with a frequency of 11 kHz

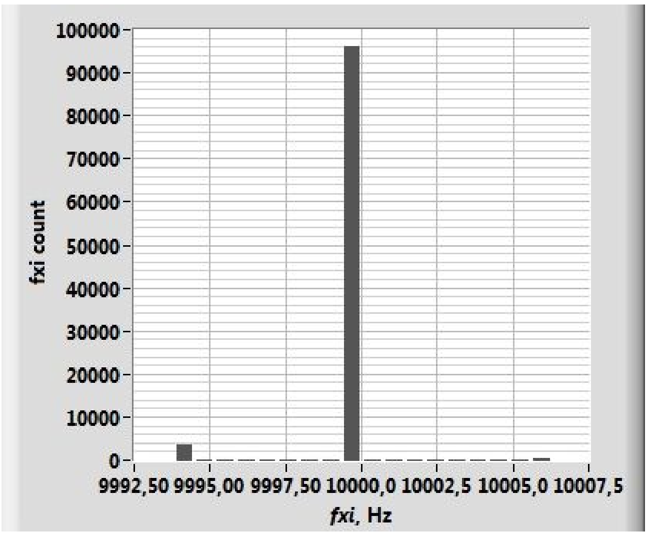

Fig. 7. Histogram of 100,000 frequency measurements of the frequency signal for a test signal with a frequency of 10 kHz

The frequency range for the top measuring range is limited by the calculation speed of the *f*/*N* converter and the data transmission baudrate through the interface. The histogram of the measurement of 100 thousand periods T_{xi} of the signal with the frequency of 11 kHz is shown in Figure 6. It can be seen that apart from the frequency of 11 kHz there is a frequency which is half of the set frequency. This is the effect of ineffective edge detection and recording of successive N_i states in the converter. Due to inefficiency problems some of the numbers N_{xi} read are actually the sum of two N_{xi}. The calculated frequency is then twice lower than the set frequency. Figure 7 shows a histogram of 100,000 period measurements of a 10 kHz signal. It can be seen that in this case the relative measurement error is ± 0.06%. The ability to process a variable frequency signal was also tested. Figure 8 shows a fragment of 1 million periods of a variable frequency test signal, the frequency of which varied according to the relationship:

$$\delta_F = 5250Hz + 5000Hz\left(\sin \pi F_x t\right) \tag{5}$$

Frequency F was set to 1 Hz. As can be seen, the reconstructed information allowed for the reconstruction of the given sinusoidal waveform with the parameters given in relation (5).

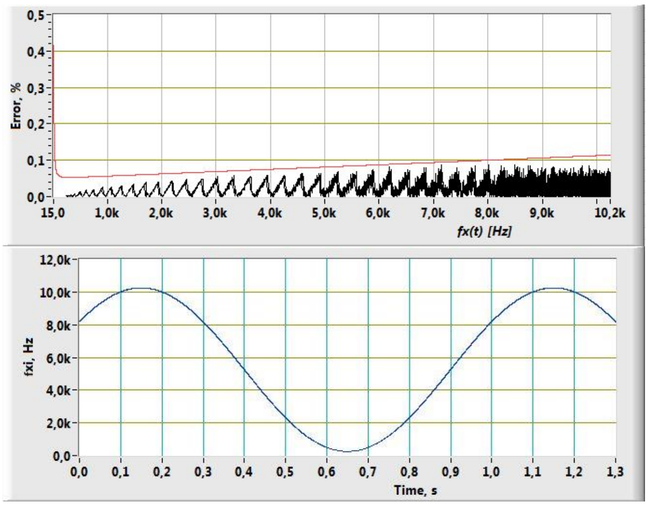

Fig. 8. The reconstructed fragment of the waveform with variable frequency (upper graph) and the error curve obtained experimentally for the processed frequency range (lower graph)

The graph of the theoretical information processing error curve in the *f*/*N* converter (3) with the generator error of 0.05% [15] is shown in red line. The course of the information processing error in the frequency-to-code converter obtained experimentally is shown in black line. As you can see, the real error introduced by the converter did not exceed the theoretical error curve at any point. It can be concluded that the developed *f*/*N* converter works properly.

6. Conclusions

The developed frequency-to-code converter allowed to confirm that there is a possibility of effective and useful direct data transmission from the digital signal period processing system to a personal computer. In the case of such a method of data archiving, microprocessor systems allowing for fast data transmission should be used. It is very important to develop an efficient data processing algorithm. As can be seen in the examples given, there are limitations in the speed of data processing in the microcontroller and they have a significant impact on the result of the processing. Of course, it is possible to increase the speed of code work, e.g. by the use of procedures created in assembler or the use of a microcontroller with higher performance. The developed device has demonstrated the ability to process a variable frequency signal in the range from 250 Hz to 10250 Hz.

References

[1] Barrett S. F., Pack D. J: Atmel avr microcontroller primer: Programming and interfacing. In Synthesis Lectures on Digital Circuits and Systems. Morgan & Claypool Publishers, Williston 2012.
[2] Francuz T.: Język C dla mikrokontrolerów AVR. Helion, 2011.
[3] Goswami J. C., Hoefel A. E.: Algorithms for estimating instantaneous frequency. Signal Process. 84, 2004, 1423–1427.
[4] Kirianaki N. V., Yurish S. Y., Shpak N. O., Deynega V. P.: Data Acquisition and Signal Processing for Smart Sensors. John Wiley & Sons, Chichester 2001.
[5] Murillo C. A., López B. C., Celma S.: Voltage-to-Frequency Converters: CMOS Design and Implementation. Springer, New York 2013.
[6] Pawłowski E., Świsulski D.: Problems with microprocessor voltage-to-frequency and frequency-to-voltage converters implementation. Przegląd Elektrotechniczny 91, 2015, 46–49.
[7] Pereira J. M., Postolache O. A., Girao P. S.: Analog to Digital Conversion Methods for Smart Sensing Systems. In: Sharma M. K. (Ed.): Advances in Measurement Systems. InTech Open, Rijeka 2010.
[8] Singh H. K., Bezboruah T.: Micro controller-based frequency to digital converter for interfacing frequency output sensors. International Conference on Electronic Design, Computer Networks & Automated Verification, 2015, 34–37.
[9] Swisulski D., Pawlowski E., Dorozhovets M.: Digital Processing of Frequency–Pulse Signal in Measurement System. Analysis and Simulation of Electrical and Computer Systems. Springer, Cham 2018.
[10] Warda P.: Adaptive correction of the quantization error in the frequency-to-code converter. 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), 2017, 226–231.
[11] Warda P.: Measurement Data Transmission in the Presence of Electromagnetic Field. Proceedings of the International Conference on Electromagnetic Devices and Processes in Environment Protection with Seminar Applications of Superconductors (ELMECO &AoS), 2017.
[12] Warda P.: Simulation of an Adaptive Method of Improving the Accuracy and Extending the Range of Frequency Signal Processing in a Frequency-to-Code Converter. Applied Sciences 11(21), 2021, 1–19 [http://doi.org/10.3390/app112110341].
[13] Yurish S. Y.: Sensors and transducers: Frequency output versus voltage output. Sens. Transducers 49, 2004, 302–305.
[14] Yurish S. Y., Reverter F., Pallàs-Areny R.: Measurement error analysis and uncertainty reduction for period-and time-interval-to-digital converters based on microcontrollers. Meas. Sci. Technol. 16, 2005, 1660.
[15] Agilent 33220A20 MHz Function /Arbitrary Waveform Generator, User's Guide, 2007 Agilent Technologies, Inc.
[16] FT232R USB UART IC Datasheet,Version 2.16
[17] SN54132, SN54LS132, SN54S132, SN74132, SN74LS132, SN74S132 Quadruple 2-input positive-nand Schmitt triggers, 2022, Texas Instruments Incorporated

Ph.D. Eng. Piotr Warda
e-mail: p.warda@pollub.pl

Graduate of the Faculty of Electrical Engineering, Lublin University of Technology (1995). Since 2000, a research and teaching employee of the Department of Automation and Metrology at Lublin University of Technology. He defended his doctoral dissertation in 2008. His areas of interest include digital metrology and measurement signal processing.

http://orcid.org/0000-0002-7525-4668