Politechnika Lubelska, Wydział Elektrotechniki I Informatyki, Katedra Automatyki i Metrologii (1)

doi:10.15199/48.2018.07.31

# Rejestrator zmiennej częstotliwości sygnału

**Streszczenie.** W artykule opisano rejestrator zmiennej częstotliwości sygnału, zrealizowany z wykorzystaniem środowiska Arduino i komputera klasy PC, oprogramowanego w środowisku programistycznym LabVIEW. Omówiono strukturę rejestratora i algorytmy ważniejszych procedur aplikacji. Zaprezentowano oprogramowanie rejestratora. Przedstawiono efekty rejestracji przykładowych sygnałów.

Abstract. This paper presents an example of recorder of instantaneous frequency, which consists of PC computer with software designed in LabVIEW and a "frequency-to-code" converter with additional memory for data. The algorithm of the software of the "frequency-to-code" converter is discussed. The PC program that supports recording process of the measurement data by "frequency-to-code" converter is presented. Examples of recordings made using the presented equipment are presented. (Recorder of variable frequency signal).

Słowa kluczowe: przetwornik częstotliwość-kod, pomiar częstotliwości, Arduino, LabVIEW. Keywords: "frequency-to-code" converter , frequency measurement, Arduino, LabVIEW.

## Wstęp

Bezpośrednie przetwarzanie częstotliwości w kod jest jedną z metod odtwarzania informacji niesionej przez sygnał częstotliwościowy. Taki sygnał wytwarzany jest między innymi przez popularne przetworniki "napięcie – częstotliwość" [1]. W ofercie producentów przetworników pomiarowych znajduje się wiele modeli przetworników różnych wielkości fizycznych w sygnał o zmiennej częstotliwości (*X/f*). Można znaleźć opracowania gotowych konstrukcji realizowanych w postaci przetworników inteligentnych [2, 3], realizowane są również badania symulacyjne [4, 5] i eksperymentalne [6], ciągle powstają nowe konstrukcje przetworników [7, 8, 9].

Zastosowanie bezpośredniego przetwarzania częstotliwości w kod pozwala, dzięki rezygnacji pośrednich stopni przetwarzania, na znaczne Z uproszczenie toru pomiarowego oraz wykorzystanie w transmisji danych sygnału częstotliwościowego. Sygnał ten charakteryzuje się dużą odpornością na zakłócenia i spadek amplitudy sygnału oraz, co się wiąże z powyższymi, łatwością przesyłania informacji na duże odległości [3]. W pomiarach stałej częstotliwości sygnału wyjściowego przetwornika wielkości fizycznej częstotliwość, pomiaru w/ najprostszym sposobem częstotliwości i odczytu wartości mierzonej wielkości fizycznej jest zastosowanie częstościomierza z odpowiednio dobraną skalą, uwzględniającą stałą przetwarzania przetwornika.

W przypadku pomiaru zmiennej częstotliwości sygnału wyjściowego przetwornika, będącej wynikiem zmienności wartości mierzonej wielkości fizycznej *X*, powyższe rozwiązanie już nie będzie pracować poprawnie, ponieważ odczyt na bieżąco jest niemożliwy. Konieczna jest rejestracja kolejnych cykli fazowych [10] przebiegu częstotliwościowego, analiza i wizualizacja w jednostce nadrzędnej, którą może być np. komputer PC [11].

Praca prezentuje przykładową realizację rejestratora przetwornika czestotliwości sygnału złożonego z częstotliwość-kod, wykorzystującego moduł Arduino Mega 2560 z rozszerzoną pamięcią na dane. Rejestrator współpracował komputerem klasv PC z Z oprogramowaniem kontrolno-pomiarowym, opracowanym w środowisku programistycznym LabVIEW, pozwalającym na przeprowadzenie procesu pomiaru, rejestracji, prezentacji i archiwizacji uzyskanych danych.

### Bezpośrednie przetwarzanie częstotliwości w kod

Zasada przetwarzania informacji niesionej przez sygnał częstotliwościowy została pokazana na rysunku 1. Pozycja

(a) prezentuje przykładowy przebieg badanej wielkości fizycznej *X*, natomiast na pozycji (b) zamieszczono uproszczoną, odpowiadającą przebiegowi *X*, reprezentację impulsowo-częstotliwościową tego sygnału. Do wyznaczenia czasu trwania kolejnych wartości czasu  $T_{xi}$ wykorzystuje się ogólnie znaną, cyfrową metodę pomiaru czasu, polegającą na sumowaniu w badanym czasie  $T_{xi}$ okresów generatora sygnału wzorcowego  $T_g$  - pozycja (c).



Rys.1. Przebiegi obrazujące przetwarzanie informacji w torze pomiarowym z bezpośrednią konwersją czasów  $T_{xi}$  sygnału częstotliwościowego w wartości liczbowe

W wyniku pomiaru uzyskuje się liczbę  $N_i$  reprezentującą sumę zarejestrowanych przez licznik przetwornika częstotliwość-kod (*f/N*) okresów  $T_g$ . Obliczenia przeprowadza się według zależności:

(1) 
$$N_{xi} = N_{i+1} - N_i + PN_{max}$$

gdzie  $N_{xi}$  – liczba reprezentująca dany  $T_{xi}$ , P – liczba przepełnień licznika.

W celu uzyskania informacji o długości kolejnych  $T_{xi}$ należy w sposób ciągły rejestrować kolejne wartości  $N_i$  – pozycja (d). Wówczas, znając wartość  $T_g$ , można odtworzyć kolejne  $T_{xi}$  posługując się zależnością

$$(2) T_{xi} = T_g N_{xi},$$

w konsekwencji częstotliwość  $f_{xi}$  – pozycja (e), obliczana jest ze wzoru:

$$f_{xi} = \frac{1}{T_{xi}}.$$

Uwzględniając czułość *S* przetwornika *X/f* można odtworzyć przebieg wielkości *X* – pozycja (f):

(4) 
$$x_{xi} = \frac{f_{xi}}{S} = \frac{1}{T_g S(N_{i+1} - N_i + PN_{\max})}$$

Błąd przetwarzania informacji przez przetwornik *f/N*, zawartej w sygnale częstotliwościowym opisuje się zależnością:

(5) 
$$\delta_{\Sigma} = \left[ \frac{T_g}{T_{xi}} + \left[ \left( 1 - \frac{\sin\left(\pi T_{xi}F\right)}{\pi T_{xi}F} \right) + \left( 1 - \frac{\sin\left(\Omega T_{xi}\right)}{\Omega T_{xi}} \right) \right] \frac{1}{\frac{X_0}{X_m} + 1} \right] \cdot 100\%$$

Wartość błędu zależy od parametrów sygnału *X*, składowej stałej *X*<sub>0</sub>, amplitudy *X*<sub>m</sub>, częstotliwości *F*, a co za tym idzie, również pulsacji  $\Omega$ =2 $\pi$ *F*, od długości czasu *T*<sub>xi</sub> i okresu generatora sygnału wzorcowego *T*<sub>g</sub>.

## Realizacja rejestratora

Strukturę rejestratora pokazano na rysunku 2. Zamieszczono literowe odniesienia do przebiegów czasowych z rysunku 1. Są w niej zawarte trzy główne elementy: przetwornik X/f, przetwornik f/N i pełniący rolę jednostki nadrzędnej komputer PC. Ze względu na zależną od potrzeb wymienność przetwornika X/f, nie będzie on w bieżącej pracy rozpatrywany. Zostanie natomiast omówiona realizacja pozostałych dwóch elementów urządzenia.



Rys. 2. Schemat blokowy układu przetwornika częstotliwość-kod

#### Przetwornik "częstotliwość-kod"

Do realizacji przetwornika *f/N* celowym jest zastosowanie mikrokontrolera [11]. Wyboru układu dokonano, biorąc pod uwagę:

- liczbę linii komunikacyjnych,
- posiadanie w strukturze 16-bitowego timera/licznika, mogącego pracować w trybie sprzętowego

przechwytywania stanu bez wstrzymywania pracy, niezbędnego do pomiaru kolejnych  $N_i$ ,

- pracę z sygnałem zegarowym o wysokiej częstotliwości,
- możliwość dołączenia zewnętrznej pamięci RAM na dane pomiarowe.

Powyższe kryteria spełnił mikrokontroler Atmega 2560 [12] stosowany w module Arduino Mega 2560 [13]. Moduł Arduino Mega 2560 posiada 54 cyfrowe porty wejścia/wyjścia, 16 analogowych wejść, 4 interfejsy UART. Mikroprocesor jest sterowany zewnętrznym generatorem kwarcowym o częstotliwości 16 MHz.

Oprogramowanie przetwornika *f/N* zostało napisane w środowisku Atmel Studio 6 [14]. W mikrokontrolerze do pomiaru kolejnych odcinków czasu użyto 16-bitowy timer/licznik nr 4 (dalej będzie nazywany licznikiem T4), wykorzystujący między innymi linię ICP4, którą można użyć do sprzętowego zapisu stanu licznika w rejestrze przechwytującym ICR4. Do konfiguracji licznika stosowane są dwa rejestry TCCR4A i TCCR4B [12]. Do poprawnej pracy w układzie rejestratora należało zaprogramować wartości pięciu komórek rejestru TCCR4B:

- ICNC4 ustawienie tego bitu na 1 włącza funkcję eliminacji zakłóceń. W tym przypadku mikrokontroler, przy wystąpieniu stanu wysokiego, będzie oczekiwać przez cztery cykle zegara. Dopiero wówczas stabilny stan wymusi rejestrowanie wybranego zbocza sygnału częstotliwościowego na linii ICP4.
- ICES4 rejestr pozwala na wybór zbocza sygnału częstotliwościowego wymuszającego wpis stanu T4 do rejestru ICR4. Przy ustawieniu 0 wpis nastąpi przy zboczu opadającym, 1 – przy narastającym.
- Ustawienie kombinacji bitów CS42, CS41, CS40 służy do ustawienia dzielnika częstotliwości sygnału generatora wzorcowego. Przy wyłączonym dzielniku (tryb pracy przy  $f_g=1/T_g$ ), przetwornik może pracować w zakresie od 122 Hz do 160 kHz, maksymalny błąd kwantowania będzie wówczas wynosił 1%. Istnieje możliwość ustawienia podziału źródła częstotliwości. Przykładowo podział przez 64 powoduje, że zakres częstotliwości wynosi od 1,9 Hz do 2,5 kHz przy błędzie kwantowania również 1%. W oprogramowaniu przetwornika zawarto procedury umożliwiające przetwarzanie częstotliwości w obydwu podanych zakresach.

Po konfiguracji i wysłaniu sygnału "start" następuje uruchomienie przetwarzania w konwerterze f/N kolejnych odcinków czasu  $T_{xi}$ . Czas trwania pomiaru zależy od częstotliwości sygnału wejściowego i ilości mierzonych  $T_{xi}$ .

Sygnał wejściowy przetwarzany jest wstępnie przez przerzutnik Shmidta. Jego działanie dostosowuje poziomy stanu wysokiego i niskiego do standardu TTL obowiązującego w zastosowanym module.

Zarejestrowanie przez mikrokontroler na linii ICP4 narastającego zbocza sygnału częstotliwościowego powoduje, że w rejestrze ICR4 zostanie zachowany aktualny stan licznika T4 przechowywany w rejestrze TCNT4. Po wykryciu kolejnego narastającego zbocza powyższa operacja jest powtarzana. W trakcie trwania pomiaru kolejnego  $T_{xi}$  mikrokontroler musi zapisać aktualną zawartość rejestru ICR4, będącą kolejną wartością  $N_i$ , do pamięci RAM przetwornika f/N. W przeciwnym przypadku dane z tego pomiaru zostaną utracone. Algorytm procedury pomiaru  $T_{xi}$  i zapisu danych do pamięci RAM pokazano na rysunku 3.

Wpisanie do bitu SRE rejestru MCUCR [12] wartości 1 spowoduje włączenie interfejsu XMEM, pozwalającego na wykorzystanie zewnętrznej pamięci SRAM. Pamięć rozszerzono za pomocą dwu elementów: układu Alliance AS7C4096A [15] o pojemności 512 kB i szybkiego zatrzasku 74AHC373. Zastosowany w module Arduino mikrokontroler [12] może adresować maksymalnie 64 kB pamięci. Z tego powodu w przetworniku zastosowano podział rozszerzonego obszaru adresowego 512 kB na osiem tzw. banków po 64 kB każdy. Do przełączania banków służą linie PD7, PL6 i PL7 modułu Arduino.



Rys. 3 Algorytm pomiaru kolejnych stykających się odcinków czasu

Po osiągnięciu zadanej przez obsługę liczby mierzonych  $T_{xi}$  przetwornik *f*/*N* przechodzi do przesyłania danych do komputera nadrzędnego.

## Oprogramowanie komputera PC

Oprogramowanie komputera nadrzędnego opracowano w środowisku programistycznym LabVIEW, pozwalającym w efektywny sposób tworzyć złożone aplikacje kontrolno -pomiarowe.



Rys 4. Panel konfiguracji rejestratora

Rolą komputera nadrzędnego w pierwszej fazie procesu rejestracji jest konfiguracja przetwornika f/N do pracy. Użytkownik ustawia (rys.4) parametry komunikacji, liczbę stanów  $N_i$  do zmierzenia (wartości z zakresu od 1000 do 200 000) oraz zakresu pomiarowego, tj. stopnia podziału częstotliwości sterującej licznikiem. Ma do wyboru, jak już wcześniej opisano, możliwość sterowania bezpośrednio

okresem sygnału głównego  $T_g$  = 62,5 ns lub okresem po podziale przez 64, co daje  $T_g$  = 4 µs.

Po procesie konfiguracji następuje uruchomienie przetwarzania sygnału częstotliwościowego przez przetwornik *f*/*N* według opisanych wcześniej zasad.

Dane przesłane do PC wymagają dalszej obróbki. Na rysunku 5 przedstawiono algorytm przetwarzania kolejnych stanów licznika  $N_i$  w wartości  $N_{xi}$  odpowiadające poszczególnym odcinkom czasu  $T_{xi}$ .



Rys. 5. Algorytm przetwarzania i zapisu danych pomiarowych

Przyjęto założenie, że zakres pomiarowy rejestratora zostaje dobrany tak, aby umożliwić co najwyżej jedno przepełnienie licznika T4 wykorzystywanego do pozyskiwania kolejnych liczb  $N_i$ . Stąd dopuszczalne jest wystąpienie liczby  $N_i$  mniejszej od liczby o indeksie niższym, czyli  $N_{i-1}$ . Oczywiście istnieje możliwość rozszerzenia zakresu sumowanych okresów  $T_g$ . Wymaga to jednak zastosowania kolejnego licznika.

Po odbiorze danych z modułu przetwornika f/N następuje obliczanie kolejnych wartości  $N_{xi}$  reprezentujących odpowiednie  $T_{xi}$ . Realizację algorytmu obliczania kolejnych liczb  $N_{xi}$  w LabVIEW pokazano na rysunku 6.

Obliczone  $N_{xi}$  są zestawiane w tablicę i przekazywane do podprogramu wyliczającego wartości  $T_{xi}$  i  $f_{xi}$  wg zależności odpowiednio (3) i (2). Rysunek 7 prezentuje podprogram realizujący powyższe czynności.



Rys. 6. Konwersja stanów  $N_i$  na liczby  $N_{xi}$ 



Rys. 7. Obliczanie przedziałów czasu  $T_{xi}$  i wartości częstotliwości  $f_{xi}$ 

Na rysunkach 8 i 9, zamieszczono dwa przykładowe wyniki eksperymentalnej rejestracji sinusoidalnego przebiegu testowego.



Rys. 8. Wynik rejestracji przebiegu testowego o parametrach  $f_0 = 50 \text{ kHz}, f_m = 49 \text{ kHz}, F = 10 \text{ Hz}.$ 



Rys. 9. Wynik rejestracji przebiegu testowego o parametrach  $f_0$  = 5 kHz,  $f_m$  = 4,9 kHz, F = 10 Hz.

Uzyskane przebiegi potwierdzają teoretyczną analizę przetwarzania danych w torze pomiarowym z częstotliwościowym nośnikiem informacji. W obszarach minimum sinusoidy widać widoczne zniekształcenia powodowane rosnącym wpływem błędu uśredniania [16]:

(6) 
$$\delta_a = 0.5(1 - \frac{\sin \pi \ T_{xi}F}{\pi \ T_{xi}F}) ,$$

natomiast szczyty sinusoidy są zniekształcone typowymi załamaniami powodowanymi występowaniem błędu kwantowania [16]:

$$\delta_k = \frac{T_g}{T_{xi}} 100\% \cdot$$

Rejestrator umożliwia zapis wyników pod wybraną nazwą na dysk, co pozwala na późniejsze wykorzystanie ich do analizy z wykorzystaniem innego, zewnętrznego oprogramowania np. Microsoft Excel.

# Podsumowanie

Opracowany model rejestratora przebiegu częstotliwościowego pozwala na przetwarzanie sygnału częstotliwościowego z błędem kwantowania mniejszym niż 1% w zakresie 1,9 Hz do 160 kHz. Aktualne oprogramowanie pozwala zarejestrować do 200 tysięcy stanów licznika przetwornika "częstotliwość-kod".

Wyniki pracy opracowanego urządzenia uznano za poprawne i zgodne z oczekiwaniami. Zostały potwierdzone teoretyczne analizy błędu przetwarzania w przetworniku częstotliwość-kod. Dla wyższych częstotliwości widoczny jest wpływ błędu kwantowania, natomiast dla niższych wpływ błędu uśredniania.

Opracowany rejestrator zmiennej częstotliwości daje szerokie możliwości prowadzenia dalszych prac badawczych. Opcja archiwizacji uzyskanych danych do dalszego wykorzystania w zewnętrznym oprogramowaniu dodatkowo uniezależnia od zastosowanych w rejestratorze rozwiązań. Zastosowany moduł Arduino pozwala na swobodną wymianę oprogramowania, co daje możliwości testowania nowych metod przetwarzania sygnału częstotliwościowego.

Autorzy: dr inż. Piotr Warda, Politechnika Lubelska, Wydział Elektrotechniki i Informatyki, Katedra Automatyki i Metrologii, ul. Nadbystrzycka 38A, 20-618 Lublin, E-mail: p.warda@pollub.pl; dr inż. Eligiusz Pawłowski, Politechnika Lubelska, Wydział Elektrotechniki i Informatyki, Katedra Automatyki i Metrologii, ul. Nadbystrzycka 38A, 20-618 Lublin, E-mail: e.pawlowski@pollub.pl

#### LITERATURA

- Murillo C.A., López B.C., Celma S., Voltage-to-Frequency Converters: CMOS Design and Implementation, Springer (2013), New York
- [2] Meijer G. C. M., Smart Sensor Systems, Wiley (2008), New York
- [3] Kirianaki N. V., Yurish S. Y., Shpak N. O., Deynega V. P. Data acquisition and signal processing for smart sensors, John Wiley & Sons, Ltd (2001), Baffins Lane
- [4] Pawłowski E., Spectrum analysis of measuring signals in sensors circuits with frequency output, *Proceedings of SPIE*, 4516 (2001), 181-186
- [5] Warda P., Voltage-to-frequency converter simulation in LabVIEW, *Przegląd Elektrotechniczny*, 90 (2014), nr 11, 133-136
- [6] Warda P., Generator with frequency modulation as a source of testing signal for a slotted line with a frequency data carrier, *Przegląd Elektrotechniczny*, 86 (2010), nr 8, 235-238
- [7] Pawłowski E., Digital processing of pulse signal from light-tofrequency converter under dynamic condition, *Proc. SPIE*, 929102, August 19, (2014)
- [8] Cheng-Ta Ch., Chi-Shen W., Yu-Ch. Huang, A CMOS Integrated Capacitance-to-Frequency Converter with Digital Compensation Circuit Designed for Sensor Interface Applications, Proc. of IEEE Sensors, (2007), 954-957
- [9] Aragones R., Oliver J., Ferrer C., A System Clock Precision Frequency to Code Converter for Low Power Supply Dependence ROIC," IEEE 4th Latin American Symposium on Circuits and Systems, LASCAS (2013)
- [10] Jasik J.R., Przetwarzanie sygnałów pomiarowych w systemach z częstotliwościowym nośnikiem informacji, Wydawnictwo Politechniki Lubelskiej, Lublin (2003)
- [11] Świsulski D., Cyfrowa rejestracja sygnałów impulsowych z częstotliwościowym nośnikiem informacji, Wydawnictwo Politechniki Gdańskiej, Gdańsk (2006)
- [12] Atmega 2560 8-bit Microcontroller with 16/32/64K Bytes In-System Programmable Flash, Atmel Corporation (2014)
- [13] Evans M., Hochenbaum J., Arduino w akcji, HELION (2014)
- [14] Francuz T., Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji, Helion, Gliwice (2011)
- [15] AS7C4096A, Alliance Semiconductor, February (2006)
- [16] Warda P., Error of conversion of 'physical quantity-tofrequency' converter output signal, *Proceedings of SPIE*, ISSN 0277-786X, nr 10161 (2016)