



## MODEL EKSTRAKTORA SYGNAŁÓW DYSKRETYCH *MODEL OF EXTRACTOR FOR DISCRETE SIGNALS*

Tomasz KUCZERSKI, Zbigniew LEWANDOWSKI  
Wojskowy Instytut Techniczny Uzbrojenia  
*Military Institute of Armament Technology*

DOI 10.5604/01.3001.0011.5825

**Streszczenie:** W artykule przedstawiono model ekstraktora sygnałów dyskretnych zrealizowany w postaci algorytmu zaimplementowanego w języku C++ na mikrokontroler jednocukładowy. Przeanalizowano możliwość wykorzystania proponowanego algorytmu do detekcji ciągów kodowych impulsowych sygnałów dyskretnych. Przedstawiono wyniki badań wstępnych zrealizowanego modelu ekstraktora z wykorzystaniem rzeczywistego sygnału impulsowych ciągów kodowych.

**Słowa kluczowe:** ekstraktor, sygnał dyskretny, bufor cykliczny, filtr medianowy

### 1. Wstęp

Na obecnym etapie rozwoju technologii cyfrowych stosowanie impulsowych ciągów kodowych znajduje zastosowanie w przypadku prac związanych z modernizacją istniejących rozwiązań na etapie przechodzenia do nowych technologii. Zastępując częściowo elementy układów nowymi rozwiązaniami często zachodzi konieczność współpracy z układami starszej generacji.

Prezentowane opracowanie przedstawia fragment toru odbiorczego impulsowych sygnałów analogowych zrealizowane w technologii cyfrowego układu programowanego w postaci algorytmu na mikrokontroler jednocukładowy. Zadaniem układu jest odwzorowanie w technologii cyfrowej oryginalnego układu odbiorczego wykonanego w technologii analogowej.

**Abstract:** A model of an extractor for discrete signals having the form of an algorithm implementing C++ in one-chip microcontroller is presented in the paper. A possibility of using the proposed algorithm for detection of strings of coded discrete impulse signals is analysed. Initial results of tests performed on the model of extractor by using the real signal of coded impulse strings are presented.

**Keywords:** extractor, discrete signal, cyclic buffer, median filter

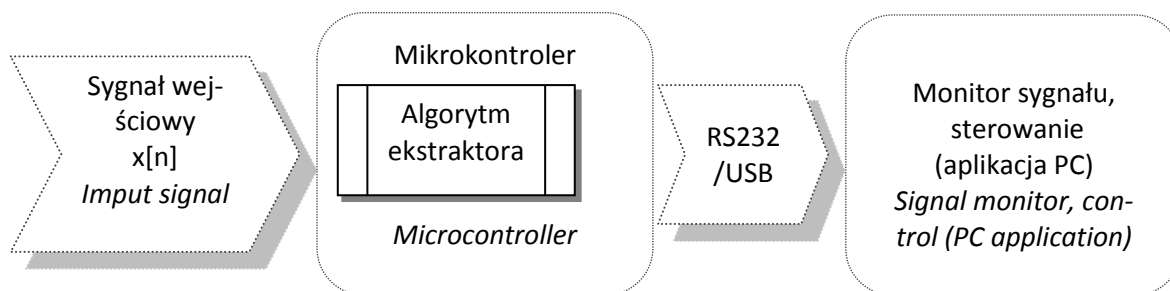
### 1. Introduction

Strings of coded impulses are used now to implement currently existing digital technologies for upgrading the older solutions to a modern level. Partial replacement of components by the new solutions requires they need to work together with older generation systems.

Presented elaboration shows a fragment of an impulse signals analogue receiving tract redesigned in the technology of a digital programmed system and having the form of an algorithm dedicated to one-chip microcontroller. The system is aimed to replicate in digital technology the original receiving system performed in analogue technology.

## 2. Model ekstraktora sygnału dyskretnego

Układ ekstraktora sygnału dyskretnego zrealizowany został w postaci algorytmu zaimplementowanego w języku C++ na mikrokontroler jednoukładowy. Schemat blokowy układu przedstawiono na rysunku 1.



Rys. 1. Schemat układu ekstraktora

Fig. 1. Diagram of extractor

Na wejściu układu występuje sygnał dyskretny w postaci impulsów binarnych  $x[n]$ , przykładowy sygnał (dla liczby impulsów w serii  $m=5$ ) przedstawiono na rysunku 2.

Sygnał wejściowy jest w postaci serii impulsów binarnych. Serie impulsów są powtarzane z określoną częstotliwością, okres powtarzania ustalonej sekwencji impulsów binarnych oznaczono symbolem  $T_p$ . Liczba impulsów  $m$  określa w tym przypadku rozmiar słowa danych binarnych i jest niezmienna w każdym okresie powtarzania.

Informacja zakodowana w sygnale jest zawarta w wartościach binarnych kolejnych impulsów w serii. Taki sposób kodowania informacji nie jest odporny na zakłócenia i może powodować po stronie odbiorczej występowanie przypadkowych błędów związanych z zanikiem pojedynczych impulsów. Dodatkowo, wykorzystanie przestarzałych układów nadawczych, z różnych przyczyn może powodować odchylenia od określonej częstotliwości powtarzania impulsów. Poprawny algorytm powinien wykazywać odporność na powyższe zakłócenia. Dla potrzeb niniejszego opracowania przyjęto następujące parametry sygnału wejściowego: liczba impulsów w okresie powtarzania  $m=5$ , liczba impulsów jest stała w każdym okresie powtarzania  $T_p$ .

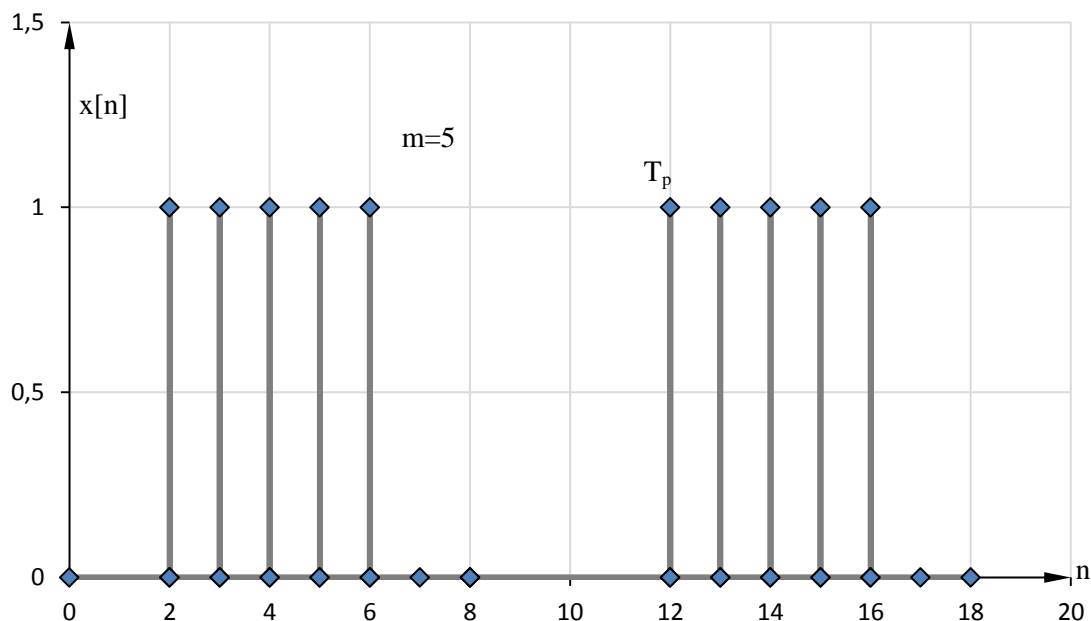
## 1. Model of Discrete Signal Extractor

The system of discrete signal extractor was made in the form of an algorithm implemented in C++ language for one-chip microcontroller. Block diagram of the system is shown in Fig. 1.

A discrete signal in the form of binary pulses  $x[n]$  in the system's input is shown in Fig. 2 as an exemplary signal (the number of pulses in the series  $m=5$ ).

Input signal has a form of a series of binary pulses. The series of pulses are repeated with a specific rate and the time of repetition for a set sequence of binary pulses is marked by  $T_p$ . In this case the number of pulses  $m$  identifies the size of binary data word and is unchangeable for each period of repetition.

Information coded in the signal is included in binary values of consecutive pulses within a series. Such way of coding is not resistant against interference and may cause the existence of casual errors connected with the missing of individual pulses. Moreover the use of obsolete transmission systems may cause for different reasons the deviation from the set specific frequency of pulse repetition. A correct algorithm has to be resistant against such interference. Following characteristics of the input signal were taken for the needs of present paper: number of pulses within a repetition time  $m=5$ , number of pulses is constant for each repetition time  $T_p$ .



Rys. 2. Impulsowy sygnał wejściowy  $x[n]$

Fig. 2. Input pulse signal  $x[n]$

W celu zwiększenia odporności układu odbiorczego na zaniki impulsów (zakłócenia sygnału) w pojedynczym okresie powtarzania zastosowano ekstraktor wykorzystujący bufor cykliczny z filtrem medianowym. Sygnał wejściowy w postaci serii impulsów z wyjścia odbiornika formowany jest w układzie kształtowania impulsów wejściowych. Uformowane impulsy o właściwej amplitudzie podawane są na wejście mikrokontrolera, w postaci zewnętrznych przerw sprzętowych. Tego rodzaju przerwy są określane jako warunek lub zdarzenie, które przerywa normalny ciąg instrukcji w programie mikrokontrolera. Przerwanie wywołuje między instrukcjami skok do procedury obsługi przerwania. Dzięki temu, można przyjąć, że procedura obsługi przerwania jest wykonywana synchronicznie z wywołującymi je impulsami wejściowymi. Niestety czas na wykonanie procedury obsługi przerwania jest ograniczony, powinna ona się zakończyć przed wystąpieniem kolejnego przerwania. Gdy procedura obsługi przerwania kończy się, wznowiany jest normalny ciąg instrukcji w programie. Ograniczony czas mierzony w liczbie kroków procesora (mikrokontrolera) wymusza zoptymalizowanie algorytmu, w taki sposób aby w procedurze obsługi przerwania wyko-

In order to increase the resistance of the receiving system to missing signals (signal interferences) within an individual time of repetition a cyclic buffer with median filter was applied in the extractor. Input signal in the form of series of pulses from the receiver output is formed in the system shaping input pulses. The formed pulses with proper amplitude are sent into the input of microcontroller in the form of external hardware interruptions. The interruptions of such type are defined as a condition or event interrupting the normal flow of instructions in the microcontroller program. The interruption makes the instruction jump to a procedure servicing an interruption. It may be accepted then that the interruption servicing procedure is executed in timing with input pulses which triggers it. Unfortunately the time for the execution of the procedure servicing the interruptions is limited as it has to be terminated before any next interruption. When the procedure for servicing an interruption ends then the normal flow of instruction in the program is renewed. The limited time measured by the number of processor's steps (microcontroller) enforces the optimisation of the algorithm to secure possibly low number of operations for servicing the procedure of

nywać jak najmniej operacji, tak aby zakończyć obliczenia przed przyjściem kolejnego impulsu powodującego ponowne wywołanie procedury obsługi przerw. Podczas kompilacji tego fragmentu oprogramowania mikrokontrolera napisanego w języku C++ należy dołożyć wszelkich starań aby kod wynikowy w postaci instrukcji asemblerowych był zwięzły i oszczędny w liczbie kroków procesora niezbędnych do jego realizacji. Z tego powodu, procedury obsługi przerw, często implementowane są bezpośrednio w asemblerze. W niniejszym opracowaniu algorytm działania ekstraktora sygnału dyskretnego został zaimplementowany w języku C++ w całości, przy czym procedura obsługi przerw została dodatkowo sprawdzona na poziomie instrukcji asemblerowych, które powstały w wyniku pracy kompilatora.

### 3. Algorytm

Badany układ ekstraktora realizuje algorytm detekcji ciągów kodowych impulsowych sygnałów dyskretnych. Zadaniem układu jest wypracowanie chwilowej binarnej wartości oznaczającej obecność na wejściu określonego ciągu impulsów lub ich brak. Wartości licznika zegara procesora są wpisywane do bufora cyklicznego w momencie wykonywania procedury obsługi zewnętrznych przerw sprzętowych, które występują w wyniku pojawiania się impulsów wejściowych. Algorytm pracy ekstraktora posiada wydzielony segment operacji sterowanych zewnętrznym przerwaniem sprzętowym.

Pozostała część algorytmu realizowana jest asynchronicznie, w czasie pomiędzy przerwami.

Ponieważ znana jest postać sygnału wejściowego, stosując zasadę filtracji dopasowanej [1] możemy wyznaczyć parametry bufora cyklicznego, który tworzy 'ruchome okno' detektora korelacyjnego i realizuje obliczenie korelacji sygnału  $x[n]$  z wzorcem. W procedurze wyrywania sygnału wyliczana jest wartość funkcji korelacji dyskretnych wartości w buforze kołowym. Funkcję autokorelacji  $\varphi_x(p)$  sygnału dyskretnego  $x(n)$  okresowego (o okresie  $T_p$ ) określa zależność (1):

interruptions to terminate the calculations before a next pulse triggering the procedure for servicing the interruptions may appear. During the compilation of this fragment of computer code for microcontroller prepared in language C++ the special care has to be applied to get the final computer code in form of assembler instructions which is short and saves the number of processor steps needed for its execution. For this reason the procedures for servicing the interruptions are often implemented directly into the assembler. In present work the algorithm of discrete signal extractor operation was implemented entirely in language C++ but the procedure of servicing the interruptions was additionally checked on the level of assembler's instructions which were created in effect of compiler's operation.

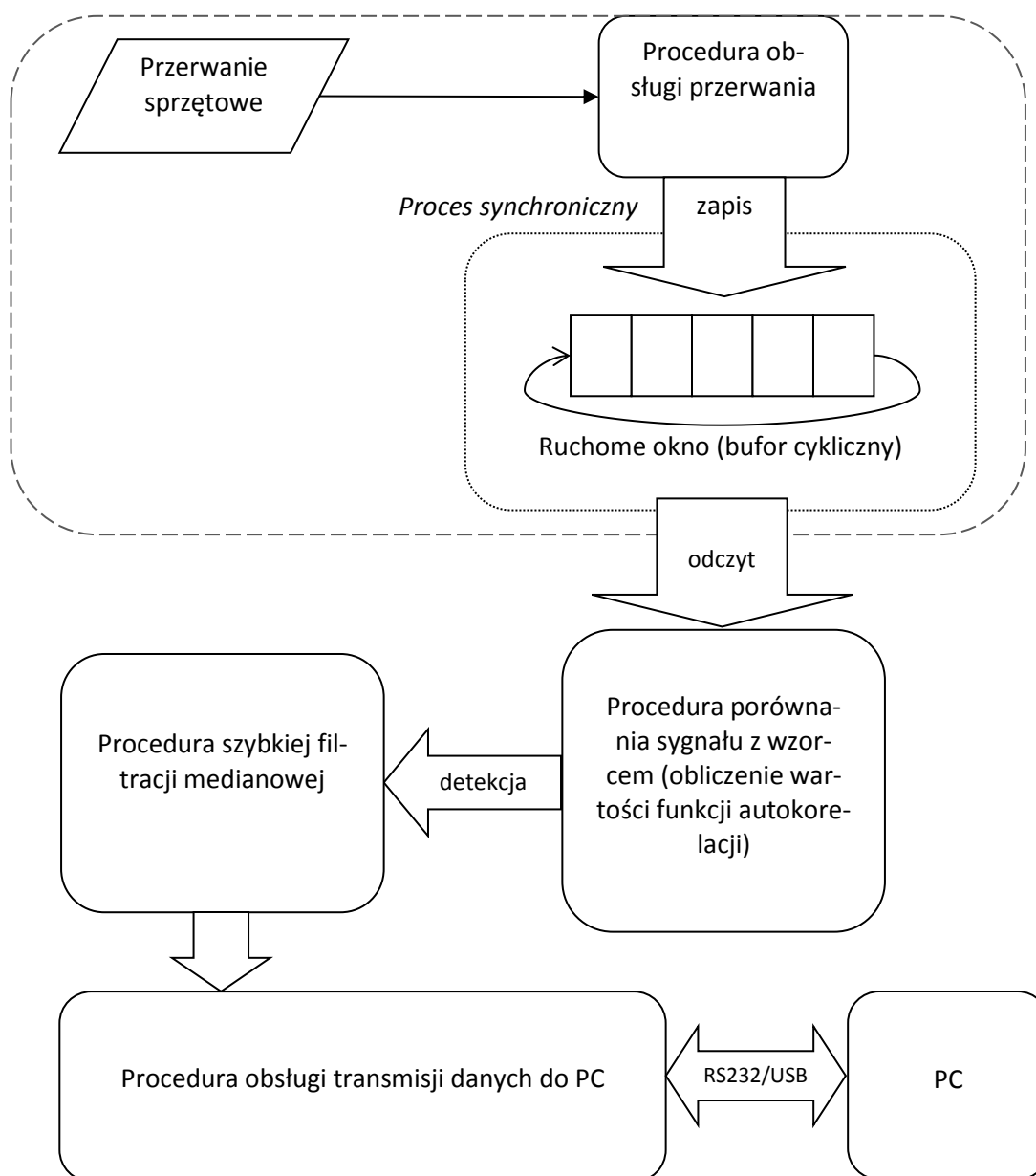
### 2. The Algorithm

Tested extractor system executes the algorithm of detection for the strings of discrete coded pulse signals. The system is designated to generate a current binary value indicating the presence or lack of a specific string of pulses in the input. The values of processor clock counter are written into the cyclic buffer in the moment of executing the procedure on servicing the external hardware interruptions existing in effect of input pulses presence. The algorithm of extractor operation has a separate segment of operations controlled by external hardware interruption.

The remaining part of the algorithm is executed in unsynchronised way between the interruptions.

As the form of the input signal is known then the parameters of the cyclic buffer may be determined by applying the principles of matched filtration [1] to create "a moving window" of correlation detector and perform the calculation of correlation between  $x[n]$  signal and the reference pattern. In the procedure of signal extraction the value of correlation function for discrete values is calculated in the cyclic buffer. The function of autocorrelation  $\varphi_x(p)$  for discrete periodic signal  $x(n)$  (with period  $T_p$ ) describes dependence (1):

$$\varphi_x(p) = \frac{1}{T_p} \sum_{t_0}^{t_0+T_p} x(n)x^*(n-p) \quad (1)$$



**Rys. 3. Algorytm działania układu**

**Fig. 3. System operation algorithm**

Przerwanie sprzętowe – *Hardware Interruption*

Procedura obsługi przerwania – *Interruption Servicing Procedure*

Zapis – *Recording*

Proces synchroniczny – *Synchronised Process*

Ruchome okno (bufor cykliczny) – *Moving Window (Cyclic Buffer)*

Odczyt – *Reading*

Procedura szybkiej filtracji medianowej – *Median Rapid Filtration Procedure*

Detekcja – *Detection*

Procedura porównania sygnału z wzorcem (obliczenie funkcji autokorelacji) – *Procedure Comparing the Signal to Pattern (Calculation of Autocorrelation Function)*

Procedura obsługi transmisji danych do PC – *Procedure Servicing Data Transmission to PC*

Kiedy sygnał dyskretny znajdujący się w buforze cyklicznym odpowiada postaci znanego sygnału wejściowego (dla określonego przesunięcia  $p$ ), wartość funkcji autokorelacji (1) osiąga maksimum, co oznacza wykrycie właściwej kombinacji impulsów dyskretnego sygnału wejściowego. Synchronizację impulsów w serii zapewnia, wspomniany, wyżej wydzielony segment operacji sterowanych zewnętrznym przerwaniem sprzętowym.

### 3.1. Algorytm - proces synchroniczny

1. Nadanie wartości początkowych
2. Oczekiwanie na wystąpienie przerwania sprzętowego
3. Obsługa procedury przerwania: odczyt wartości licznika zegarowego, zapis danych do bufora cyklicznego

### 3.2. Algorytm - proces asynchroniczny

1. Nadanie wartości początkowych
2. Odczyt danych z bufora cyklicznego
3. Porównanie zawartości bufora cyklicznego z wzorcem sygnału, korelacja, detekcja
4. Filtracja sygnału (filtr medianowy)
5. Procedura obsługi transmisji danych do komputera PC

## 4. Wyniki badań i wnioski

Celem badań układu, stanowiącego fragment toru odbiorczego impulsowych sygnałów analogowych, było sprawdzenie działania wybranego procesora wykonującego operacje zgodnie z algorytmem napisanym w języku C++.

Sygnał wejściowy, odpowiadający rzeczywistemu sygnałowi impulsowych ciągów kodowych (stosowanych w układach radio-kierowania), został doprowadzony do układu z wyjścia odbiornika. Badany układ został połączony z komputerem PC za pomocą konwertera RS232/USB. Na komputerze przy pomocy specjalnie opracowanej aplikacji obserwowano obecność sygnału odpowiadającego zadawanemu na wejściu sygnałom kodowym.

Wyniki badań wstępnych potwierdziły

When a discrete signal inside the cyclic buffer corresponds to the form of known input signal (for a specific shift  $p$ ) then the value of autocorrelation function (1) gets maximum what means that the relevant combination of discrete input signal pulses is detected. The synchronisation of pulses within a series is secured by the separate segment mentioned earlier for operations controlled by an external hardware interruption.

### 3.1. Algorithm - Synchronised Process

1. Setting initial values,
2. Waiting for hardware interruption event,
3. Servicing the interruption procedure: reading the clock counter value, recording the data into the cyclic buffer.

### 3.2. Algorithm-Unsynchronised Process

1. Setting initial values,
2. Reading the data from the cyclic buffer,
3. Comparing the contents of the cyclic buffer with the signal pattern, correlation, detection,
4. Signal filtration (median filter),
5. Procedure servicing data transmission to PC.

## 4. Results of Tests and Conclusions

The system being a part of a receiving device for analogue pulse signals was tested to check the operation of the applied processor according to the algorithm prepared in C++ language.

The input signal corresponding to a real signal of strings of coded pulses (used in radio-control systems) was connected to the system from the receiver's output. Tested system was connected with the PC computer via RS232/USB converter. A specially prepared application was used to observe on the computer the presence of signal corresponding to coded signals fed to the input.

Results of initial tests have confirmed the efficiency of operation for accepted

skuteczność działania zakładanego algorytmu. W czasie badań stwierdzono wrażliwość układu na zmiany okresu powtarzania  $T_p$ . Wynikało to z przyjętej stałej wartości częstotliwości powtarzania, co okazało się błędne, gdyż w realnych (zarejestrowanych) sygnałach analogowych występują odchylenia od zakładanej częstotliwości powtarzania. W czasie badań w układzie zastosowano adaptacyjną metodę pomiaru okresu powtarzania  $T_p$  ciągu kodowego. Zapewniło to zwiększoną odporność na przypadkowe zmiany lub zakłócenia  $T_p$ . Dzięki temu, układ dopasowuje się w pewnym zakresie do indywidualnych cech urządzenia nadawczego lub niedokładności jego zestrojenia. Badania z wykorzystaniem sygnałów rzeczywistych mają znaczący wpływ na rozwiązywanie problemów modernizacyjnych związanych, z 'cyfryzacją' układów analogowych.

algorithm. The tests have indicated the system is vulnerable to changes of the repetition period  $T_p$ . It was caused by the assumption of a constant frequency of repetition what has proved to be a fault as the real (recorded) analogue signals deviate from the assumed repetition rate. An adapting method for measurement the repetition period  $T_p$  of the coded string was used during the system tests. It has increased the resistance against casual changes or interferences of  $T_p$ . It makes the system match in certain degree to individual characteristics of a transmitting device or inaccuracies of its adjustment. Tests carried out with the use of real signals are significant for finding solutions at upgrading projects connected with "digitalisation" of analogue systems.

### **Literatura / Literature**

- [1] Kuczerski Tomasz, *Application of Neural Network to Radar Signals Matched Filtering, Proceedings of the XXIst National Conference on CIRCUIT THEORY and ELECTRONIC NETWORKS*, Poznań-Kiekrz, 1998r.
- [2] Kuczerski Tomasz, Tomczyk Mariusz. *Fuzzy Algorithm for Multidimensional Clustering and Elementary Detections Estimation, Microwaves, Radar and Wireless Communications, MIKON-2000 International Conference*, Wrocław, 2000 r.





