

Realizacja sterowania zespołem manipulatorów za pomocą wybranych mikrokontrolerów AVR oraz PSoC

Rafał Kociszewski, Szymon Kochanowski

Politechnika Białostocka, Wydział Elektryczny, ul. Wiejska 45A, 15-351 Białystok

Streszczenie: W pracy przedstawiono problematykę realizacji sterowania zespołem manipulatorów za pomocą wybranych mikrokontrolerów 8-bitowych. Wykorzystano w tym celu układ nowej serii ATXmega rodziny AVR oraz układ należący do programowalnej rodziny PSoC. Pokazano, że generowanie synchronicznego (takiego samego) ruchu przez dwa manipulatory za pomocą impulsów sterujących serwomechanizmami jest możliwe do realizacji z poziomu jednego 8-bitowego mikrokontrolera. Warunkiem jest dysponowanie odpowiednią liczbą bloków peryferyjnych mogących generować niezależne sygnały PWM.

Słowa kluczowe: manipulator, mikrokontroler AVR, mikrokontroler programowalny PSoC, sygnał PWM

Rozwój przemysłu powodował przez lata pojawianie się coraz większych komponentów produkcyjnych. Od początku najłatwiejszym sposobem ich przemieszczania była siła człowieka. Transportowane ładunki z racji gabarytów stały się jednak zagrożeniem dla pracownika zakładu. Wobec tego opracowano manipulator, czyli urządzenie, które pozwala na osiągnięcie dwóch celów: z jednej strony odciąża pracownika, a z drugiej – zwiększa bezpieczeństwo jego pracy [9].

Manipulator łączy zalety wielu urządzeń czy maszyn, takich jak suwnica, wciągarka, czy żuraw. Jego zadaniem jest podnoszenie ładunku za pomocą chwytaka, zgodnie z poleceniami operatora wydawanymi z pulpitu sterowniczego lub zgodnie z automatycznie realizowanym algorytmem zapisanym w pamięci sterownika. Manipulatory przydają się zwłaszcza tam, gdzie czynności wykonywane przez operatora są powtarzalne, a ponadto wymagają dużej dokładności. Wykorzystaniu manipulatorów sprzyjają także krótkie czasy pomiędzy kolejnymi czynnościami wykonywanymi na tym samym stanowisku. Często spotykaną sytuacją jest przenoszenie określonego detalu jednocześnie przez dwa manipulatory. W takim przypadku urządzenia muszą cechować się zsynchronizowaną i niezawodną pracą, tak aby komponent był w tym samym czasie podniesiony, przeniesiony i osadzony w danym miejscu.

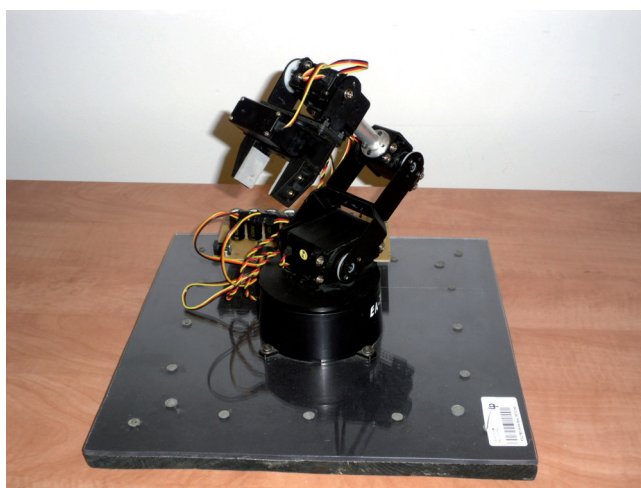
Niniejsza praca dotyczy problematyki implementacji sterowania zespołem manipulatorów za pomocą wybranych mikrokontrolerów 8-bitowych. Kryterium wyboru układów zakłada

możliwość swobodnego generowanie wielu, niezależnych sygnałów PWM sterujących serwomechanizmami dwóch manipulatorów, zapewniając w ten sposób ich dowolnie przyjęte przez użytkownika zsynchronizowane ruchy.

1. Zespół manipulatorów

Badanym obiektem jest zespół manipulatorów złożony z dwóch urządzeń. Pierwszy to gotowy produkt firmy Lynxmotion o oznaczeniu handlowym AL5A [8] (rys. 1). Drugi manipulator [5] pracuje w układzie z modelem AL5A (rys. 2).

Manipulatory mają pięć stopni swobody (liczba osi robota) [10]: pierwszy – określający ruch względem podstawy, drugi – zgięcie barku, trzeci – zgięcie w ramieniu, czwarty – zgięcie nadgarstka i piąty – obrót nadgarstka. Ruch poszczególnych elementów manipulatorów jest generowany za pomocą serwomecha-



Rys.1. Manipulator 1
Fig. 1. Manipulator 1

Autor korespondujący:

Rafał Kociszewski, rafko@pb.edu.pl

Artykuł recenzowany

nadesłany 10.11.2014 r., przyjęty do druku 30.03.2015 r.

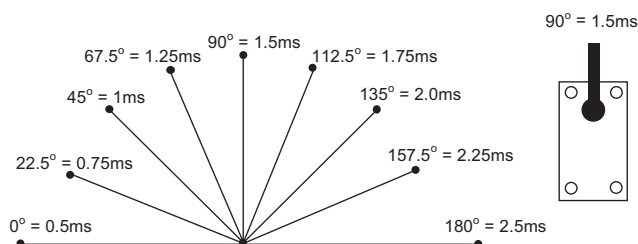


Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0



Rys. 2. Manipulator 2
Fig. 2. Manipulator 2

nizmów typu standard firmy Hitec o takich samych parametrach odnośnie zakresu kąta obrotu wału w zależności od parametrów impulsów sterujących. W manipulatorze 2, ze względu na większą powierzchnię przestrzeni roboczej, wykorzystano przy podstawie serwomechanizmy o zwiększonej wartości momentu obrotowego bez wpływu jednak na wymienione wcześniej parametry. Konfiguracja wewnętrzna serwomechanizmu zawiera komutatorowy silnik DC, przekładnię mechaniczną, potencjometr sprzężony z silnikiem oraz wyspecjalizowany scalony układ sterujący [7]. Użytkownik ustawiając parametry impulsu na linii sygnałowej wyznacza pozycję, w której powinien się znaleźć wał serwomechanizmu zaś serwomechanizm musi tą pozycję osiągnąć i utrzymać. Czas trwania impulsu (stan wysoki) w zastosowanych serwomechanizmach zawiera się w przedziale 0,5...2,5 ms, co zapewnia obrót wału o kąt odpowiednio 0°...180° w stosunku do położenia neutralnego. Osiągnięcie położenia neutralnego, tj. 90° zapewnia impuls o czasie trwania 1,5 ms (rys. 3).



Rys. 3. Wartości wychyleń wału dla impulsów określonej długości
Fig. 3. Values of inclination of shaft for predetermined length pulses

Okres sygnału sterującego w serwomechanizmach zawiera się w przedziale 10...20 ms (bez wpływu na dokładność uzyskania zadanej pozycji wału). Sygnał musi się składać z ciągu takich impulsów, gdyż pojedynczy impuls nie zapewnia obrotu wału. W użytych manipulatorach sygnały o opisanej charakterystyce wychodzące z mikrokontrolerów sterujących do poszczególnych serwomechanizmów zostały odseparowane galwanicznie za pomocą transoptorów CNY74-4H.

Problem sterowania synchronicznego zespołem manipulatorów o 5 stopniach swobody sprowadza się do wygenerowania łącznie 10 sygnałów PWM (ang. *Pulse-Width Modulation*) o różnym współczynniku wypełnienia, zależnym od ustalonej pozycji wału. Sygnały docierające do serwomechanizmów umieszczonych w każdym punkcie swobody danego manipulatora muszą względnie posiadać identyczne parametry (taki sam czas trwania stanu wysokiego). Opóźnienie pojawiające się na drodze sygnału

do danego serwomechanizmu w jednym z manipulatorów może wygenerować zwłokę w ruchu danej części manipulatora. Zjawisko to jest wysoce niepożądane i skrajnym skutkiem takiego opóźnienia może być nierównomierność ruchu zespołu manipulatorów i zniszczenie komponentu np. przy jego odkładaniu. Wśród najpopularniejszych na rynku 8-bitowych mikrokontrolerów rodziny AVR serii ATmega (firmy Atmel) lub rodziny PIC (firmy Microchip) nie jest możliwe wygodne wygenerowanie większej liczby sygnałów PWM. Ograniczeniem jest liczba dostępnych układów czasowo-licznikowych (timerów), w ramach których można takie sygnały uzyskać. Możliwe są także inne wersje realizacji sterowania manipulatorem oparte na strukturach programowalnych FPGA [4, 11] lub 32-bitowych mikrokontrolerach z rdzeniem ARM-Cortex zaprezentowane w [3] oraz cytowanej tam literaturze. Urządzenia będące tematem wymienionych publikacji nie pracują w zespole, są bardziej zaawansowane i mają inne cele w stosunku do prezentowanego w niniejszej pracy układu manipulatorów.

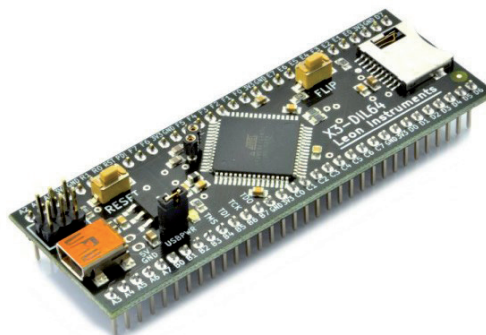
Zasadniczym celem pracy jest przedstawienie sposobu synchronicznego sterowania zespołem manipulatorów (każdy z manipulatorów wykonuje tą samą sekwencję w tym samym czasie) za pomocą mikrokontrolera nowej serii Xmega rodziny AVR, a także interesującego, konfigurowalnego mikrokontrolera rodziny PSoC. Wymienione układy, wyposażone w rdzeń 8-bitowy pozwalają na swobodne generowanie niezależnych sygnałów PWM w liczbie znacznie większej niż potrzeby analizowanego układu.

2. Sterowanie zespołem manipulatorów

2.1. Realizacja z zastosowaniem mikrokontrolera rodziny AVR Xmega

Rodzina 8-bitowych mikrokontrolerów AVR Xmega została niedawno wprowadzona do sprzedaży przez firmę Atmel i stopniowo zyskuje na popularności. Mikrokontrolery Xmega wyposażono w funkcjonalności dotychczas sporadycznie spotykane wśród rdzeni 8-bitowych (np. w porównaniu z poprzednią generacją ATmega). Innowacją jest system obsługi zdarzeń, który umożliwia współpracę bloków funkcjonalnych bez udziału CPU, przy jednoczesnym zachowaniu gwarantowanego czasu reakcji pomiędzy wystąpieniem zdarzenia a uruchomieniem procedury jego obsługi. Tego rodzaju funkcja nadaje się do użycia w aplikacjach, w których kluczowe znaczenie ma zapewnienie reakcji układu w czasie nie dłuższym, niż wymagany (np. w automatyce przemysłowej lub motoryzacji). Praca systemu obsługi zdarzeń jest wspierana przez kontroler DMA czy też wielopoziomowy system przerwań [2].

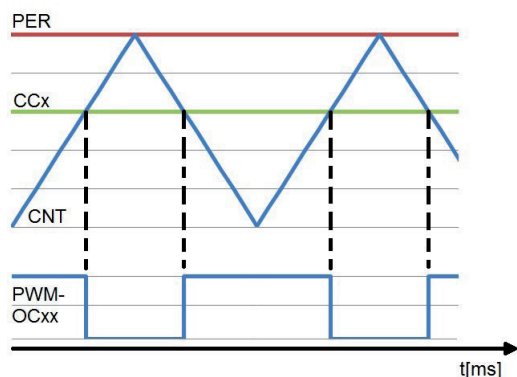
Do realizacji sterowania zespołem manipulatorów wykorzystano gotową płytkę z układem ATXmega128A3U (rys. 4), którego najważniejsze parametry są następujące: pamięć Flash 128 kB, SRAM 8 kB, EEPROM 2 kB, 4-kanalowy kontroler



Rys. 4. Płytkę z mikrokontrolerem ATXmega128A3U
Fig. 4. Board with ATXmega128A3U microcontroller

DMA, 8-kanalowy system zdarzeń, 7 programowalnych timerów, kontroler Full-speed USB, 7 interfejsów USART, 3 interfejsy TWI, 3 interfejsy SPI, moduł kryptologiczny AES i DES, Generator CRC-16, dwa 16-kanalowe 12-bitowe przetworniki A/C, 2-kanalowy 12-bitowy przetwornik C/A, 4 komparatory analogowe, przerwanienia INT na wszystkich wyprowadzeniach I/O procesora, taktowanie do 36 MHz.

Jak wspomniano wcześniej, sygnały sterujące poszczególne serwomechanizmy można wytwarzać korzystając z układów czasowo-licznikowych. W mikrokontrolerze ATXmega128A3U zaimplementowano 16-bitowe liczniki typu 0/1, które można zamienić na liczniki 8-bitowe typu 2, z których każdy ma 4 kanały. W efekcie można uzyskać 32 niezależne kanały PWM. Timer generujący sygnał PWM może pracować w trybie *single slope* – zliczając zawsze w tym samym kierunku, lub w trybie *dual slope* – zliczając naprzemiennie w górę do wartości maksymalnej, a następnie w dół do 0. W ramach realizacji sterowania zespołem manipulatorów wykorzystano tryb *dual slope*, którego idea została zilustrowana na rys. 5.

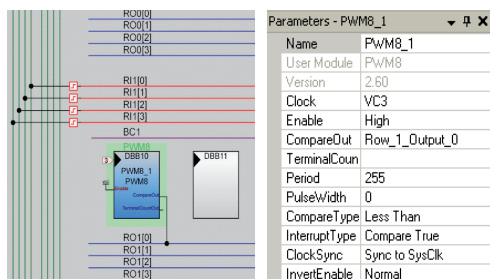


Rys. 5. Idea generowania sygnału PWM z wykorzystaniem trybu dual slope

Rys. 5. The idea of generating of PWM signal with dual slope mode

Oznaczenie PER odnosi się do rejestru, który odpowiada za wartość, do której zlicza licznik. W zależności od rozdzielczości, może to być 255 (dla 8-bitowej) lub 65 535 (dla 16-bitowej). CCx (x jest numerem kanału) to rejestr, którego wartość wpływa na wypełnienie przebiegu (gdy $CNT = CCx$, przy czym CNT to rejestr, który jest porównywany z PER aby określić wartość jego przepełnienia) zaś PWM-Cxx to kanał dostępny na danym pinie mikrokontrolera, na którym jest generowany sygnał PWM. Przykładowo realizacja podanego niżej fragmentu programu napisanego w środowisku programistycznym IDE Atmel Studio zgodnie z ideą pokazaną na rys. 5, odpowiada za generację przebiegu o okresie 50 ms i szerokości impulsu wynoszącym 1 ms.

```
TCC0.CTRLB = TC_WGMODE_DSBOOTH_gc |
TC0_CCAEN_bm;
TCC0.PER = 10000;
TCC0.CCA = 10;
TCC0.CTRLA = TC_CLKSEL_DIV2_gc;
```

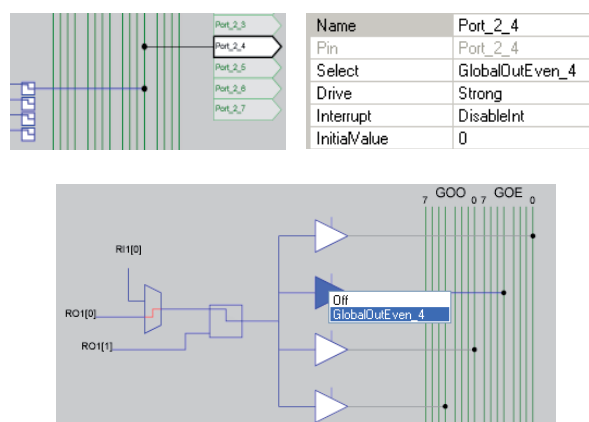


Rys. 6. Konfiguracja rozdzielczości oraz wypełnienia sygnału
Fig. 6. Resolution and pulse width configuration of signal

2.2. Realizacja z zastosowaniem mikrokontrolera rodziny PSoc

Mikrokontrolery PSoc (ang. *Programmable System on Chip*) produkowane przez firmę Cypress stanowią unikalną rodzinę mikrokontrolerów na rynku komponentów elektronicznych. Cechą charakterystyczną architektury PSoc jest obecność swobodnie konfigurowalnych przez użytkownika bloków analogowych i cyfrowych, które mogą pełnić funkcję dowolnych, wymaganych w danej chwili, bloków funkcjonalnych [1]. Mogą to być np. kontrolery PWM, timery, kontrolery magistral szeregowych, przetworniki A/C i C/A, wzmacniacze, filtry itd. Kolejną cechą są konfigurowalne połączenia wewnętrzne, co powoduje, że poszczególne bloki funkcjonalne mikrokontrolera nie są przyporządkowane na stałe do konkretnych wyprowadzeń. Projektant ma możliwość wykonania swobodnej konfiguracji sprzętowej układu PSoc, wybierając te zasoby, które zostaną w nim umieszczone. Dzięki temu PSoc może zostać optymalnie dostosowany do konkretnej aplikacji. Na możliwości konkretnego zastosowania mają wpływ zasoby danego układu PSoc. Do sterownika manipulatorów wybrano układ CY8C29466-24PXI należący do 8-bitowej rodziny PSoc1. Jest on opisany następującymi właściwościami: pamięć Flash 32 kB, SRAM 2 kB, emulacja pamięci EEPROM w pamięci programu, 16 bloków cyfrowych, 12 bloków analogowych, 24 piny wejścia/wyjścia, taktowanie do 24 MHz (wewnętrzny oscylator lub zewnętrzny rezonator kwarcowy), napięcie zasilania od 3 V do 5,25 V, SMP (ang. *Switch Mode Pump* – wewnętrzna pompa ładunkowa pozwalająca na zasilanie od 1 V). Mikrokontroler jest dostępny w obudowie DIP 28. Do realizacji sterownika z tym układem wykorzystano typową prototypową płytę stykową.

Sterownik zespołu manipulatorów musi generować 10 sygnałów PWM. Do realizacji 8-bitowego PWM w układzie CY8C29466 należy zastosować tylko jeden blok cyfrowy [6], co



Rys. 7. Konfiguracja połączeń wewnętrznych wyjścia PWM do wybranego pinu mikrokontrolera (P2[4])

Fig. 7. Configuring internal connections of PWM output into selected microcontroller pin (P2[4])

przy dostępnej liczbie bloków jest wystarczające. Proces konfiguracji kanału PWM (na przykładzie pojedynczego sygnału) w dedykowanym środowisku IDE PSoc Designer polega na ustawieniu parametrów globalnych jak wybór napięcia zasilania, częstotliwość taktowania rdzenia czy ustawienia preskalera linii zegarowej. Następnie dokonywany jest wybór bloku PWM8 (rozdzielczość 8-bitowa) i realizacja połączeń.

Skonfigurowaną graficznie aplikację uzupełniono o kod w języku C, odwołujący się de facto do opracowanych przez producenta podprogramów assemblerowych realizujących inicjalizację danego modułu PWM i odpowiadający za generowanie

sygnału o pożądanym parametrach. Przykład takiego odwołania jest pokazany niżej.

Język C (fragment kodu):

```
PWM8_1_Start();
PWM8_1_WritePeriod();
PWM8_1_WritePulseWidth();
```

Asembler (podprogramy):

```
PWM8_1_Start:
_PWM8_1_Start:
or REG[PWM8_1_CONTROL_REG],
bfCONTROL_REG_START_BIT
ret

PWM8_1_WritePeriod:
_PWM8_1_WritePeriod:
tst REG[PWM8_1_CONTROL_REG],
bfCONTROL_REG_START_BIT
jnz .CounterRunning
.CounterStopped:
push X
mov X, A
M8C_SetBank1
mov A, REG[PWM8_1_INPUT_REG]
push A
and REG[PWM8_1_INPUT_REG], F0h
M8C_SetBank0
mov A, X
mov REG[PWM8_1_PERIOD_REG], A
pop A
M8C_SetBank1
mov REG[PWM8_1_INPUT_REG], A
M8C_SetBank0
pop X
ret

.CounterRunning:
mov REG[PWM8_1_PERIOD_REG], A
ret
```

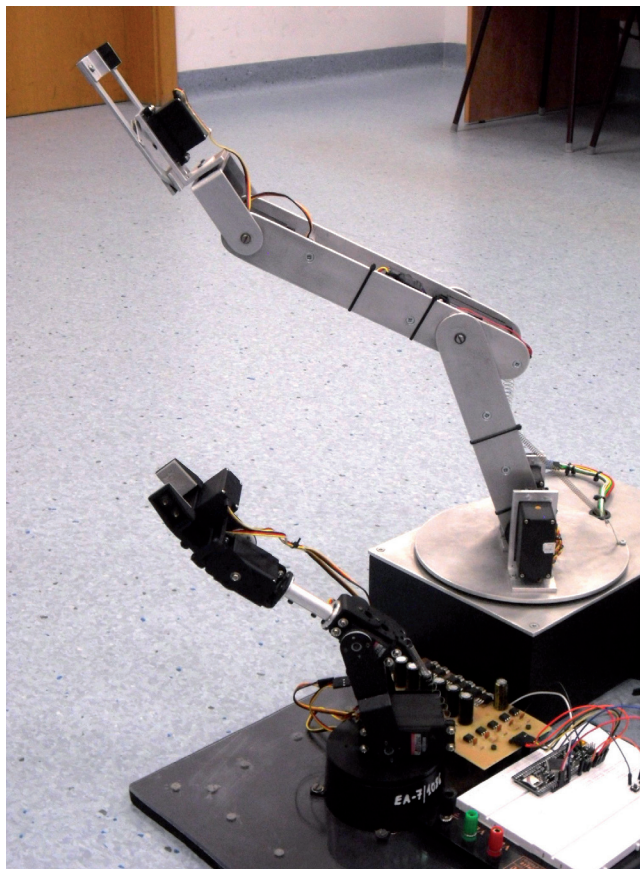
PWM8_1_WritePulseWidth:

```
_PWM8_1_WritePulseWidth:
mov REG[PWM8_1_PWIDTH_REG], A
ret
```

3. Wybrane wyniki badań układu sterowania

W niniejszym punkcie zaprezentowano oscylogramy wybranych sygnałów sterujących serwo mechanizmami dwóch manipulatorów, generowane za pomocą mikrokontrolera ATXmega128A3U.

W układzie pokazanym na rysunku 8 sprawdzano synchroniczne wykonywanie zadanych ruchów, realizowanych za pomocą zmiany szerokości impulsu sygnałów sterujących serwo mechanizmami w przedziale 0,5...2,5 ms z krokiem 0,1 ms. Przy czym, ze względu na konstrukcję manipulatorów polegającą na określonym zamocowaniu serwo mechanizmów, nie wszystkie serwo mechanizmy mogły być sterowane sygnałem o takich parametrach. Przykładowo, serwo me-



Rys. 8. Przykładowa orientacja zespołu manipulatorów
Fig. 8. Example orientation of team manipulators

chanizm generujący ruch względem podstawy ma zakres 0,8...2,2 ms.

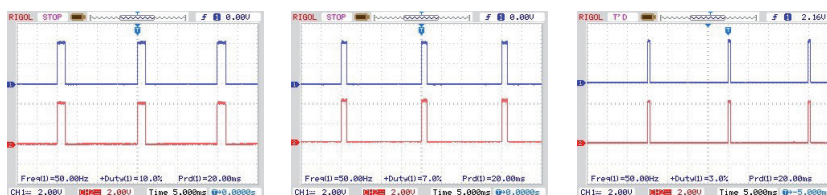
Wybrane oscylogramy prezentujące wygenerowane sygnały sterujące są pokazane na rys. 9.

Przebiegi zaznaczone kolorem niebieskim (kanał 1) sterują serwo mechanizmy (chwytak, nadgarstek, łokieć) manipulatora 1, zaś kolorem czerwonym (kanał 2) – manipulatora 2. Szerokość impulsu sygnałów PWM wynosi odpowiednio 2 ms, 1,4 ms, 0,6 ms i zawierają się one w przedziale pokazanym na rys. 3. Podczas badań nie zaobserwowano żadnych różnic w sygnałach generowanych przez opisywane wcześniej mikrokontrolery, które mogłyby wpłynąć na nieprawidłowy zakres ruchu jednego z manipulatorów. Analogiczne badania przeprowadzono także z zastosowaniem układu CY8C29466. W tym przypadku uzyskano również powtarzalne synchroniczne ruchy zespołu manipulatorów.

4. Uwagi końcowe

W pracy wykazano, że można zaimplementować synchroniczne sterowanie ruchem dwóch manipulatorów w jednym 8-bitowym mikrokontrolerze. Manipulatory wykonują dokładnie ten sam ruch bez opóźnień. Sprawdzone w tym celu zasadność

zastosowania dwóch różnych układów: ATmega128A3U z rodziny AVR oraz CY8C29466 należący do rodziny PSoC1. W analizowanym przypadku możliwości pod względem zasobów peryferyj-



Rys. 9. Wybrane przebiegi sterujące serwo mechanizmami manipulatorów
Fig. 9. Selected waveforms servo control of manipulators

nych zastosowanych mikrokontrolerów są w zupełności wystarczające. Należy zauważyć jednak, że zwiększając dwukrotnie rozdzielczość sygnału PWM struktura mikrokontrolera PSoC1 wymaga użycia dwóch bloków cyfrowych. Z jednej strony stanowi to niewątpliwie zaletę (taka rozdzielczość jest niemożliwa do uzyskania w układzie Xmega), ale również pewne ograniczenie i redukcję kanałów PWM.

Na obecnym etapie prac nie założono realizacji określonego cyklu pracy, np. przenoszenia komponentu z położenia początkowego do końcowego. Wynika to z faktu zastosowania urządzeń z różnymi przestrzeniami roboczymi. Zadanie wykonywania takiego cyklu jest możliwe do zrealizowania, biorąc pod uwagę np. zasoby pamięci zaimplementowanej w zastosowanych mikrokontrolerach. Można w niej zapisać wartości kątów obrotu poszczególnych serwomechanizmów, które będą następnie cyklicznie odtwarzane. Wymaga to jednak zastosowania dwóch identycznych urządzeń, np. modeli AL5A. Taka problematyka będzie analizowana w następnym etapie badań.

Podziękowanie

Publikację zrealizowano w ramach pracy S/WE/1/11 finansowanej przez Ministerstwo Nauki i Szkolnictwa Wyższego.

Bibliografia

1. Ashby R., *Designer's Guide to the Cypress PSoC*, Elsevier 2005.
2. Francuz T., *AVR praktyczne projekty*, Wyd. Helion 2013.
3. Khan A.A., Kumar P., Reddy K., *Design and Implementation of Electronic Gesture Recognition Unit Using Accelerometer to Control Robotic Arm Powered With Cortex-M3 Core*, "International Journal of Engineering Research and Development", Vol. 2, Issue 11 (August 2012), 14–18.
4. Kung Y-S., Shu G-S., *Development of a FPGA-based motion control IC for robot arm*. "IEEE International Conference on Industrial Technology (ICIT'05)", Hong Kong 2005. DOI: 10.1109/ICIT.2005.1600854.
5. Maciejewicz R., *Projekt i realizacja manipulatora dydaktycznego sterowanego za pomocą mikrokontrolera ARM*. Praca dyplomowa, Politechnika Białostocka, Białystok 2012.
6. Murphy R., *Getting Started with PSoC® 1, Cypress Semiconductor*, 2012.
7. Raabe Z., *Sterownik serwomechanizmów*. „Elektronika dla wszystkich”, Nr 4, 1997, 14–17.
8. Robotic Arms [www.lynxmotion.com/c-124-al5a.aspx] (dostęp: 6 listopad 2014).
9. Schilling R., *Fundamentals of Robotics – Analysis and control*, Prentice-Hall 1998.
10. Spong M.W., Vidyasagar M., *Dynamika i sterowanie robotów*, WNT 1997.
11. Thweny F.A., Alhammad A.L., *Implementation of FPGA Based PSO PID Controller for Feedback IVAX SCARA Robot Manipulator*, International "Journal of Computer Science Engineering & Technology", Oct 2013, Vol. 3, Issue 10.

Control of Team Manipulators Using Selected AVR and PSoC Microcontrollers

Abstract: The paper presents a problem of control of team manipulators. It has been shown that this task can be solved by the use of 8-bit microcontrollers with suitable numbers of PWM channels. The steering strategy (control servo drive) has been successfully implemented in the ATmega128A3U microcontroller and in Programmable System on Chip CY8C29466.

Keywords: manipulator, AVR microcontroller, programmable microcontroller, PWM signal

dr inż. Rafał Kociszewski

rafko@pb.edu.pl

Absolwent Wydziału Elektrycznego Politechniki Białostockiej (2001 r.). Obecnie adiunkt w Katedrze Automatyki i Elektroniki na Wydziale Elektrycznym Politechniki Białostockiej. Zainteresowania naukowe autora są skoncentrowane na syntezie optymalizacyjnych metod sterowania oraz wykorzystaniu techniki mikroprocesorowej do realizacji algorytmów sterowania.



inż. Szymon Kochanowski

Szymon.Kochanowski90@gmail.com

Student II stopnia kierunku Elektrotechnika (specjalność Automatyka przemysłowa i technika mikroprocesorowa) na Wydziale Elektrycznym Politechniki Białostockiej. Aktualnie pełni funkcję przewodniczącego Koła Naukowego Automatyki i Robotyki. Zainteresowania zawodowe: robotyka, elektronika, aplikacje z zastosowaniem mikrokontrolerów.

