

Daniel RATAJ*, Krzysztof TOMCZEWSKI*

STEROWNIK AUTONOMICZNEGO ROBOTA MOBILNEGO NA BAZIE UKŁADU FPGA

W artykule przedstawiono koncepcję układu sterowania robota mobilnego, przeznaczonego do pracy jako platforma odkurzacza autonomicznego. W ramach projektu opracowano wyspecjalizowaną strukturę realizującą wszystkie podstawowe funkcje odkurzacza. Do budowy sterownika zastosowano układ FPGA Spartan 3E. W napędach robota zastosowano silniki prądu stałego z wbudowanymi enkoderami. Robot został wyposażony w czujnik odległości, czujniki zapobiegające upadkowi w przypadku dojazdu do schodów, czujniki mechaniczne wykrywające przeszkody. Robot może poruszać się autonomicznie lub zdalnie sterowany przez operatora z aplikacji w telefonie komórkowym za pośrednictwem interfejsu Bluetooth. Działanie robota testowano rejestrując trajektorie jego ruchu.

SŁOWA KLUCZOWE: układ programowalny, FPGA, robot mobilny, sterowanie.

1. WSTĘP

Największą grupę robotów sprzedawanych obecnie stanowią roboty mobilne do zastosowań domowych. Są to odkurzacze autonomiczne i kosiarki do trawy. Działanie robotów odkurzających nie wymaga żadnej ingerencji człowieka. Robot samodzielnie przemieszcza pomieszczenie w wyznaczonych godzinach sprząając podłogi. Przerwy w jego działaniu mogą być ograniczone do koniecznych ze względu na ładowanie akumulatorów. W wielu przypadkach również ta czynność nie wymaga ingerencji człowieka. Robot sam kontroluje poziom naładowania akumulatorów. Po ich rozładowaniu przerywa sprząatanie i przemieszcza się do stacji ładującej, do której samoczynnie podłącza do ładowania akumulatory. Po naładowaniu akumulatorów powraca do pracy.

Głównym zadaniem algorytmu sterowania jest realizacja takiej trajektorii ruchu, aby w jak najkrótszym czasie pokryła całą sprzątaną powierzchnię, zapewniając możliwie równomierny jej rozkład. Dodatkowo robot musi chronić się samodzielnie przed zniszczeniem, na przykład w wyniku upadku ze schodów [2].

W niektórych przypadkach taki sposób działania robota jest niewystarczający. Przydatną opcją jest możliwość przełączenia trybu pracy na sterowanie

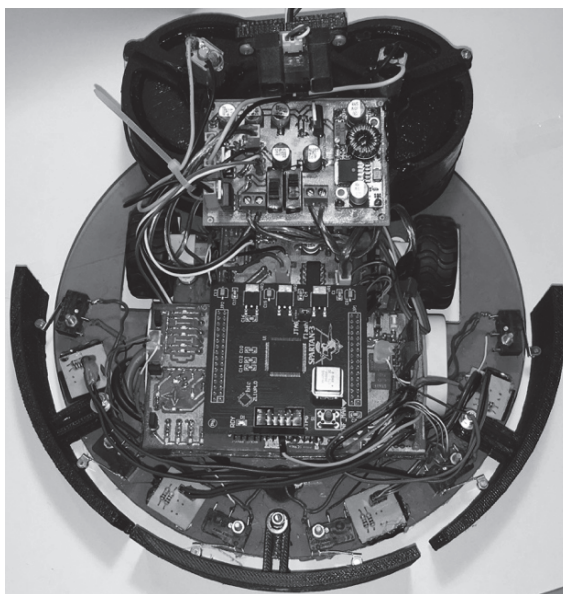
* Politechnika Opolska

zdalne przez operatora. Umożliwia to dodatkowe sprzątnięcie bardziej zabrudzonych miejsc lub posprzątanie miejscowego zabrudzenia, np. w przypadku przypadkowego rozsypania produktów sypkich, bez konieczności sprzątania całego pomieszczenia.

2. BUDOWA ROBOTA

2.1. Konstrukcja mechaniczna

Model fizyczny konstrukcji mechanicznej robota został opracowany w celu przeprowadzenia testów projektowanego układu sterowania. Platformę jezdną wykonano z laminatu szklano-epoksydowego, którego obróbkę wykonano na obrabiarce CNC. Podstawa robota ma kształt koła, w którym wycięto otwory montażowe. W przedniej części robota zamontowano dwa koła napędowe. Z tyłu zamontowano kulkę podporową. W przedniej części zamontowane zostały trzy zderzaki sprzężone z czujnikami wykrywającymi przeszkody. Przyjęto, że robot wykonuje obroty w miejscu, a ruchy postępowe tylko do przodu. Platforma ta służy jako podstawa nośna dla akumulatorów i układu sterowania oraz mechanizmu ssącego. Konstrukcję mechaniczną platformy z zamontowanymi układami napędowymi pokazano na rys. 1.



Rys. 1. Konstrukcja mechaniczna modelu robota mobilnego

2.2. Układ elektroniczny

Większość układów sterowania w robotach mobilnych i różnego rodzaju urządzeniach elektronicznych zawiera mikrokontrolery, stanowiące centralną jednostkę sterującą. Duże zapotrzebowanie rynku na tego typu układy przyczyniło się do szybkiego rozwoju mikrokontrolerów. Dzięki temu w ofercie rynkowej znaleźć można mikrokontrolery o strukturze dostosowanej do potrzeb różnego typu urządzeń. Zaletą tak wyspecjalizowanego sprzętu jest to, że dobór urządzeń peryferyjnych jest wykonany w sposób dostosowany do konkretnych zastosowań, przez firmę specjalizującą się w realizacji tego typu zadań, a struktura układu jest wielokrotnie testowana przed wdrożeniem układu do produkcji. Mikrokontrolery wyposażane są w szereg urządzeń peryferyjnych ułatwiających budowę układów sterowania takich, jak przetworniki analogowo-cyfrowe, interfejsy komunikacyjne, a nawet wewnętrzne czujniki wielkości nieelektrycznych. Liczba i rodzaj tych urządzeń zależy od serii i wersji układu. Wadą takiego rozwiązania jest brak możliwości zmiany jego struktury w celu przystosowania do nietypowych zastosowań co często wymaga stosowania struktur nadmiarowych.

W przypadku układów programowalnych sytuacja jest inna, ponieważ to projektant układu sterowania opracowuje własną strukturę sprzętową, a następnie implementuje ją w sterowniku. Do jego zadań należy zdefiniowanie portów wejść/wejść, którym następnie przypisuje się konkretne funkcje. Struktura układów programowalnych pozwala na dużą dowolność realizacji systemów sterowania. W przypadku układów programowalnych implementowane są tylko te elementy, które wymagane są w realizowanym projekcie. Wadą układów programowalnych jest natomiast brak większości gotowych układów peryferyjnych takich, jak interfejsy komunikacyjne, przetworniki analogowo cyfrowe. Projektant musi utworzyć te elementy wewnątrz struktury lub dołączyć w postaci zewnętrznych układów [1, 6].

Przewaga układów programowalnych nad mikrokontrolerami w wielu zastosowaniach polega na możliwości współbieżnej realizacji wielu zadań. Mikrokontrolery umożliwiają jedynie wykonywanie operacji szeregowo. Tylko wyspecjalizowane jednostki obliczeniowe wbudowane w procesorach sygnałowych umożliwiają realizację obliczeń w jednym cyklu, nie spowalniając realizacji algorytmu. Producenci dostosowują te struktury do konkretnych zadań, takich jak dekodowanie dźwięku, obrazu, sterowanie napędami. Z tego powodu mikrokontrolery takie dedykowane są tylko do typowych zastosowań.

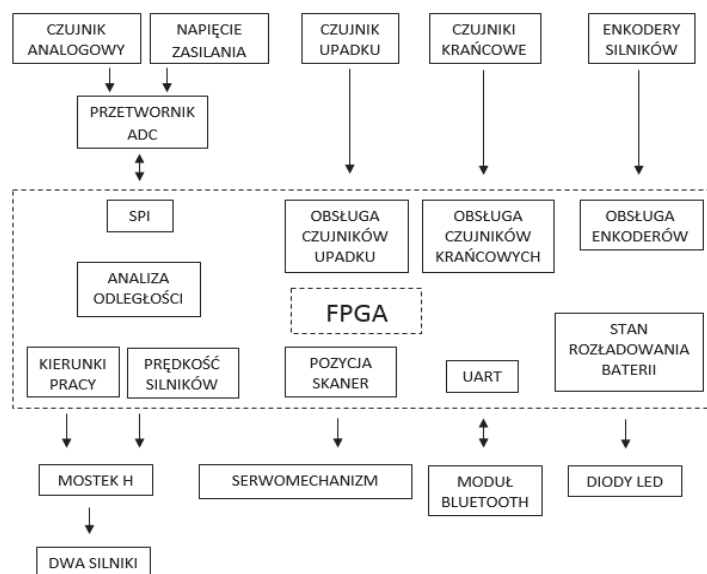
Pomimo implementacji takich wyspecjalizowanych struktur mikrokontrolery nie są w stanie w wielu przypadkach dorównać szybkością działania układom programowanym, w których większość zadań może być realizowana współbieżnie. Im więcej prostych, niezależnych od siebie zadań realizuje algorytm, tym bardziej rośnie przewaga układów programowalnych.

Układy FPGA, oprócz matrycy bramek logicznych i rozproszonej pamięci, zawierają struktury DSP złożone z mnożarek i sumatorów. Umożliwiają one tworzenie jednostek obliczeniowych. Ich liczba jest różna i waha się od kilkunastu do kilkuset w dużych układach FPGA. Daje to możliwość jednoczesnej realizacji wielu obliczeń. W związku z tym realizacja algorytmu regulacji prędkości jednego napędu zajmuje w przybliżeniu tyle samo czasu obliczeniowego, ile regulacja dwóch lub większej liczby napędów. [4]

Wykorzystując powyższe cechy układu programowalnego opracowano sterownik autonomicznego robota mobilnego, przeznaczonego do napędu odkurzacza automatycznego.

2.3. Implementacja algorytmu sterowania w układzie FPGA

Układ sterowania zaimplementowano w strukturze Spartan 3E firmy Xilinx. Opis funkcjonalny struktury układu wykonano w języku VHDL. W rezultacie analizy przyjęto strukturę przedstawioną na rys. 2.



Rys. 2. Struktura układu sterowania

Producent wykorzystanego do budowy sterownika układu FPGA udostępnił do projektowania środowisko ISE Design Suite. Za pomocą tego środowiska opracowano i zaimplementowano strukturę sterownika w wykorzystanym module FPGA ZL10PLD. W projekcie wykorzystano również programy iMPACT oraz PlanAhead, będące dodatkami do programu ISE.

W pierwszym etapie realizacji projektu wykonano opis modułów w języku VHDL zgodnie ze schematem blokowym uzyskanym w wyniku analizy układu. Projekt zawiera trzy bloki wykonawcze, cztery bloki sensoryczne oraz jeden moduł zarządzający, którego zadaniem jest przetwarzanie sygnałów otrzymanych z poszczególnych bloków. Poszczególne elementy projektu zaprojektowano w ten sposób, aby działały współbieżnie.

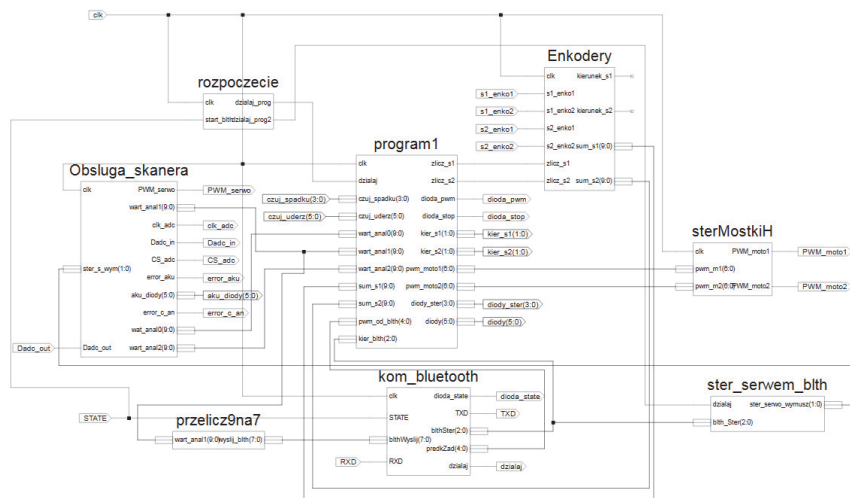
W skład elementów wykonawczych wchodzi następujące bloki:

- sterownik mostka H – do którego wyjść podłączono dwa napędy,
- sterownik serwomechanizmu - napęd skanera,
- dekodery do obsługi diod sygnalizacyjnych LED [3, 5].

Do elementów modułu sensorycznego należą bloki:

- przetwornika analogowo-cyfrowego - do obsługi czujnika odległości oraz pomiaru napięcia akumulatorów,
- czujników upadku oraz czujników odległości,
- obsługi enkoderów - do odczytu prędkości poszczególnych silników,
- obsługi modułu Bluetooth XM-15b.

W kolejnym etapie, wykorzystując opracowane moduły, dokonano syntezy układu w module graficznym. Schemat struktury opracowanego sterownika przedstawiono na rys. 3.



Rys. 3. Struktura główna sterownika zaimplementowanego w FPGA

Dla poprawy czytelności schematu trzy bloki zawierają zgrupowane zespoły połączonych bloków. Są to **obsługa_skanera**, a w nim zawarty **ADC_przetw** oraz **kom_bluetooth**. Struktura ta przystosowana została do pracy w trybie au-

tonomicznym oraz do pracy w trybie ręcznym. Przełączanie między trybami następuje w sposób automatyczny.

3. TRYBY PRACY UKŁADU

3.1. Tryb pracy autonomicznej

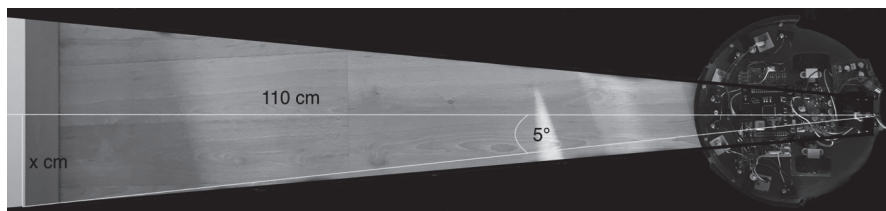
Podstawowym trybem pracy odkurzacza automatycznego tryb automatyczny. Realizacja tego zadania w układzie programowalnym jest najbardziej skomplikowaną częścią opracowanej struktury. Układy programowalne są przystosowane do szybkiego przetwarzania sygnałów, jednakże wykonywanie zadań typowych dla mikroprocesorów przez wyspecjalizowany układ logiczny bez implementacji w nim mikroprocesora jest skomplikowane. Działanie opracowanego sterownika kontrolowane jest przez blok „program_główny”. Blok ten oparto o konstrukcję *process*, dzięki czemu zawarte w niej instrukcje są wykonywane w sposób sekwencyjny. Na podstawie sygnałów wejściowych z czujników generowane są sygnały sterujące:

- prędkością silników,
- kierunkiem pracy silników,
- zliczaniem impulsów z enkoderów,
- diodami sygnalizacyjnymi LED.

Sygnały wejściowe doprowadzane są do jednostki sterującej z:

- czujników zderzeniowych,
- czujników upadku,
- czujnika odległości,
- enkoderów.

Po uruchomieniu sterownik rozpoczyna skanowanie przestrzeni przed robotem i przekazuje sygnały sterujące do modułu sterowania mostkiem H, ustalając ruch robota do przodu. Sterownik podaje ten sygnał tak długo, dopóki nie zdekoduje przeszkody. Sygnał wyjściowy z czujnika jest przetwarzany przez przetwornik analogowo-cyfrowy, a następnie za pośrednictwem interfejsu SPI przekazywany jest do układu FPGA. Sterownik na podstawie danych odebranych z przetwornika ADC ustala wartości współczynnika wypełnienia sygnału PWM o okresie 20 ms, ustawiającego serwomechanizm w odpowiedniej pozycji. Zakres skanowania uzależniono od odległości do przeszkody i obejmuje strefę od 90° do 90° w stosunku do kierunku jazdy. Zasięg czujnika wynosi od 26 cm do 120 cm. W przypadku braku przeszkody skaner pracuje w zakresie kątów od 5° do 5°, co przedstawiono na rys. 4.



Rys. 4. Strefa skanowania przestrzeni przed robotem w przypadku braku przeszkody

Szerokość skanowanej strefy odpowiada szerokość robota. W miarę zmniejszania się odległości do przeszkody wykrytej przez czujnik odległości następuje stopniowe zwiększanie zakresu kąтового pracy skanera. Dzięki temu, przy przeszkodzie znajdującej się około 10 cm przed robotem, skanowanie obejmuje pełny zakres pracy skanera, czyli 180° . Pozwala to na wyznaczenie kierunku dalszego poruszania się robota zależnie od ustalonych odległości w zakresie kątów od -90° oraz 90° . Przyjętą zależność strefy skanowania od odległości do przeszkody pokazano w tabeli 1.

Tabela 1. Zależność strefy i częstotliwości skanowania od odległości do przeszkody.

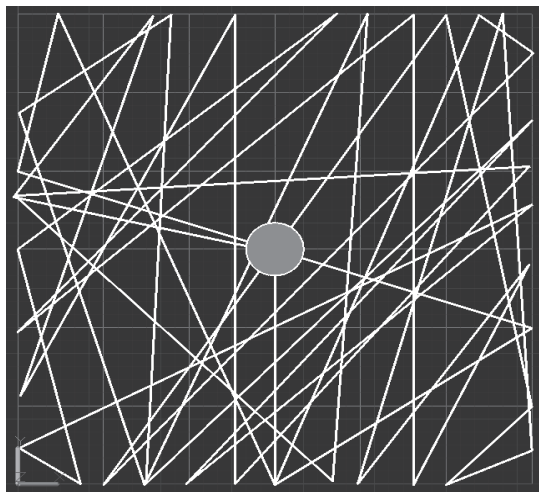
Odległość do przeszkody [cm]	Zakres skanowania [°]	Częstotliwość skanowania [Hz]
< 35	± 90	3,0
80 do 34	± 45	4,5
110 do 79	± 20	6,0
> 110	± 5	14,1

W tabeli 1. zawarto parametry pracy oraz częstotliwości pracy skanera w trybie automatycznym. Praca w tym trybie pozwala na wykonywanie pomiarów tej części przestrzeni, w stronę której porusza się robot. Na podstawie wyników pomiarów skanera ustalana jest również prędkość poruszania się robota. Im bliżej przeszkody znajduje się robot tym wolniej się porusza.

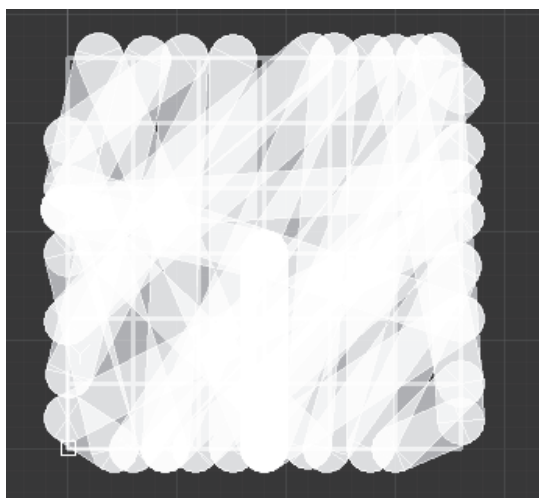
Drugi kanał przetwornika ADC wykorzystywany jest do pomiaru napięcia akumulatorów. Napięcie znamionowe ma wartość 6,6 V.

Działanie układu sterowania sprawdzono w pomieszczeniu o powierzchni około 4 m^2 . Trajektorię ruchu urządzenia zarejestrowano za pomocą kamery. Zarejestrowaną w pierwszym teście trajektorię ruchu przedstawiono na rys. 4.

Trajektoria z rysunku 4 została przebyta w czasie około 5 min 45 s. Zwiększając szerokość linii tworzących trajektorię do szerokości podstawy robota wynoszącej 21 cm otrzymano obraz pokrycia powierzchni (rys. 6), którą przy zamontowanym systemie czyszczącym robot byłby w stanie wyczyścić.



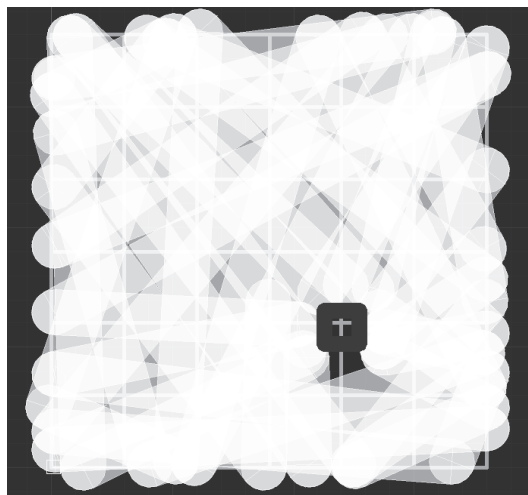
Rys. 4. Trajektoria robota w pierwszej próbie pomiarowej



Rys. 5. Pokrycie powierzchni testowej przez trajektorię robota

Trajektoria z rysunku 4. pokrywa prawie całą powierzchnię testową. W programie graficznym ustawiono ją jako przezroczystą (rys. 5). Pozwala to ocenić równomierność pokrycia powierzchni testowej. Odkurzacze automatyczne pracują z reguły minimum dwa razy dziennie. Pozwala to zmniejszyć prawdopodobieństwo pozostawienia niewyczyszczonych miejsc. W kolejnej próbie pokonanie trajektorii trwało 6 min 30 s. Po dwóch próbach pokrycie powierzchni było bardziej równomierne.

Kolejny test polegał na zbadaniu zachowania robota w pomieszczeniu, w którym umieszczono przeszkodę. Trajektoria ruchu robota w tej próbie pokazana została na rys. 6.



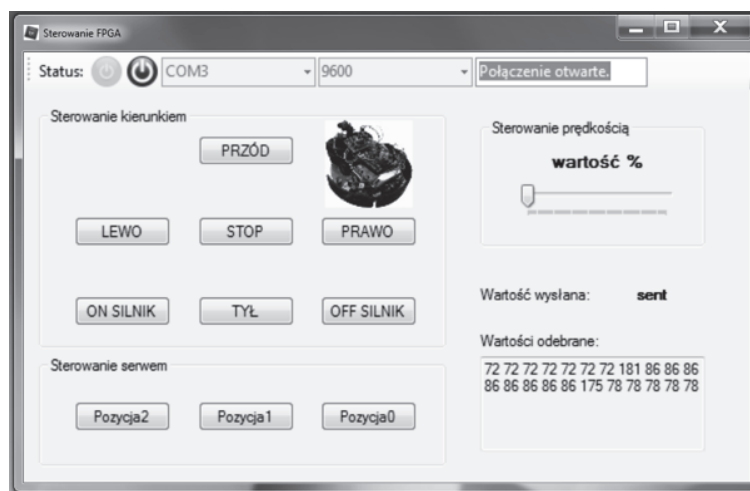
Rys. 6. Trajektoria ruchu robota w pomieszczeniu z przeszkodą

Jak widać na rys. 6. robot nie dotarł tylko do niewielkiego fragmentu powierzchni testowej przy przeszkodzie.

3.2. Tryb sterowania zdalnego

Model robota zgodnie ze swoim przeznaczeniem przystosowany został w głównej mierze do pracy w trybie automatycznym, jednak możliwe jest jego sterowanie zdalne przez operatora. Umożliwia to zaimplementowany interfejs komunikacyjny Bluetooth. Tryb manualny umożliwia przejęcie kontroli nad robotem przez użytkownika, za pomocą urządzenia wyposażonego w interfejs Bluetooth takiego, jak smartfon czy komputer PC. W ramach projektu opracowano aplikację umożliwiającą zdalne sterowanie robotem.

W celu nawiązania komunikacji z robotem należy sparować moduł Bluetooth podłączony do sterownika z komputerem wyposażonym w taki interfejs w wersji v 2.1. Po prawidłowym podłączeniu pojawia się on w menadżerze urządzeń w którym należy wybrać nowy port COM. Jego nazwę należy podać w aplikacji. Po nawiązaniu komunikacji należy uruchomić opracowaną aplikację *Sterowanie FPGA*. Interfejs programu przedstawiono na rys. 7.



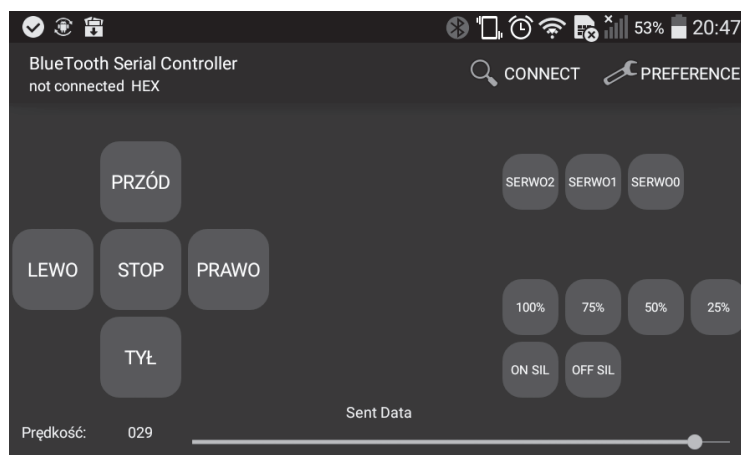
Rys. 7. Okno aplikacji sterującej robotem

Możliwość ręcznego sprzątania pozwala dotrzeć w miejsca, które wykazują zabrudzenie i nie wymaga sprzątania całej powierzchni w trybie automatycznym. Pozwala również na korekty związane z niedokładnościami sprzątania w trybie automatycznym.

Aplikacja oprócz standardowego sterowania kierunkiem oraz prędkością poruszania się robota umożliwia również uruchomienie dwóch silników napędzających turbiny zasysające powietrze pod robotem. Turbiny te zamontowane zostały w tylnej części modelu robota i zostały wydrukowane w technologii druku 3D. Uruchomienie silników następuje po naciśnięciu ikony „ON SILNIK” natomiast wyłączenie „OFF SILNIK”. Podczas pracy w trybie automatycznym robot uruchamia odkurzanie bezpośrednio po załączeniu.

W dolnej części aplikacji umieszczono przyciski służące do sterowania pozycją skanera. Za ich pomocą możliwe jest ustawienie czujnika w pozycji 0°, 90° oraz 180°. Odczytane wartości odległość od przeszkody wyświetlane są w prawej dolnej części.

Utrata połączenia lub zakończenie komunikacji czerwonym przyciskiem w części konfiguracyjnej aplikacji powoduje przełączenie robota w tryb pracy automatycznej i kontynuowanie sprzątania. Do sterowania robotem za pomocą urządzenia mobilnego z systemem Android wymagana jest instalacja aplikacji „Bluetooth Serial Controller”. Aplikację konfiguruje się przy pierwszym uruchomieniu, przypisując odpowiednim przyciskom wartości w systemie szesnastkowym.



Rys. 8. Okno aplikacji sterującej robotem w urządzeniu mobilnym

4. PODSUMOWANIE

Opracowany układ realizuje większość typowych funkcji odkurzacza automatycznego. W projekcie nie uwzględniono jedynie automatycznego wyszukiwania stacji ładującej.

Układ sterowania realizuje zadany algorytm przemieszczania się robota w pomieszczeniu. Wyposażenie robota w zestaw czujników: upadku, uderzenia oraz odległości, pozwala odpowiednio reagować na przeszkody znajdujące w pomieszczeniach, wytyczać trajektorię ruchu oraz chronić urządzenie przed zniszczeniem w wyniku upadku. Dodatkową funkcjonalnością opracowanego robota jest możliwość zdalnego sterowania bezprzewodowego. Funkcja ta umożliwi sterowanie robotem w celu usuwania lokalnych zabrudzeń. Jest to możliwe dzięki wyposażeniu robota w moduł Bluetooth, pozwalający na przejście nad nim kontroli przez aplikację komputerową.

Wykonanie modelu robota pozwoliło przetestować działanie opracowanego układu sterowania w obiekcie rzeczywistym. Przeprowadzone badania trajektorii ruchu pozwalają stwierdzić, że praca sterownika jest prawidłowa i robot realizuje swoje zadania. W trakcie pojedynczych testów robot pozostawiał jedynie małe fragmenty niesprzątej powierzchni.

Układy programowalne wykazują się dużą niezawodnością oraz szybkością działania. Realizują wiele czynności jednocześnie, co umożliwi implementację nawet rozbudowanych układów sterowania.

W opracowanym sterowniku układ FPGA realizuje funkcje odczytu stanu czujników upadku i uderzenia, odczytu sygnału z analogowego czujnika odległości poprzez interfejs SPI, sterowanie pracą serwomechanizmu ustalającego kierunek działania czujnika odległości, sterowanie napędami kół robota oraz

komunikację poprzez interfejs UART z modułem Bluetooth w celu zdalnego sterowania z aplikacji w telefonie komórkowym lub komputerze PC.

BIBLIOGRAFIA

- [1] Chakravarthy N., Xiao J., FPGA-based Control System for Miniature Robots. IEEE Intelligent Robots and Systems. Beijing, China, 2006.
- [2] Chongstitvatana P., Kongmunvattana A., A FPGA-based Behavioral Control System for Mobile Robot. IEEE Asia-Pacific Conference on Circuits and Systems, Chiangmai, Thailand, 1998.
- [3] Dinesh O. Dange, Rajan Mevekari, Dipankar D. Khartad, DC Motor Speed Control using LabVIEW FPGA Modeling, Control Algorithm Simulation & Implementation. IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 05, 2015 | ISSN (online): 2321-0613.
- [4] Fares Sassi, Mehdi Abbes, Abdelkader Mami, FPGA Implementation of PID Controller. International Conference on Control, Engineering & Information Technology, Proceedings- Copyright IPCO-2014, pp.1-13.
- [5] Manjunath U. S., Sandhya P., K. V. Devadas, FPGA Control of SPV Array Fed BLDC Motor Driven Water Pumping System Employing ILZ Converter. Electrical and Electronic Engineering 2017, 7(2): 11-17.
- [6] Shilpa Kale, S. S. Shriramwar, FPGA-based Controller for a Mobile Robot. International Journal of Computer Science and Information Security, IJCSIS July 2009, Vol. 3, No. 1, USA.

FPGA BASED CONTROLLER FOR AN AUTONOMOUS ROBOT

The paper presents a concept of a simple autonomous robot controller with an example application in an autonomous vacuum cleaner. During the project a specialized unit was designed, which performs all the basic functions of a vacuum cleaner. The implementation was based on the Spartan 3E FPGA board. For the drive system of the robot DC motors with built-in encoders were used. Additionally, the robot is equipped with a distance sensor, cliff sensors preventing fall of a staircase as well as mechanical sensors for recognizing obstacles. The robot can operate either autonomously or it can be controlled from a smartphone application via Bluetooth interface. The operation of the robot was tested by registering the routes chosen by the robot.

(Received: 06.02.2018, revised: 08.03.2018)