

Marek Grobelny
Sebastian Jarosiński
Marcin Pilat
Daniel Pieniak
Łukasz Sobaszek

Instytut Technologicznych Systemów Informacyjnych

Wydział Mechaniczny, Politechnika Lubelska

20-618 Lublin, ul. Nadbystrzycka 36

marek.grobelny92@gmail.com

sebastian.jarosinski2912@gmail.com

marcin.pilat@gmail.com

daniel.pieniak@gmail.com

l.sobaszek@pollub.pl

Projekt aplikacji komputerowej umożliwiającej sterowanie robotem przemysłowym Kawasaki RS003N

1. Wstęp

We współczesnym przemyśle roboty znajdują coraz szersze zastosowanie [1, 5]. Popularne staje się tworzenie zrobotyzowanych gniazd obróbczych czy montażowych, wykorzystanie robotów do spawania, bądź malowania. Kluczowym celem takich rozwiązań jest zastąpienie ludzi w pracy na stanowiskach uciążliwych, bądź charakteryzujących się niebezpiecznymi warunkami [7]. Roboty są w stanie wykonywać pracę, która jest ryzykowna dla ludzi, albo wymaga dużej siły fizycznej lub precyzji.

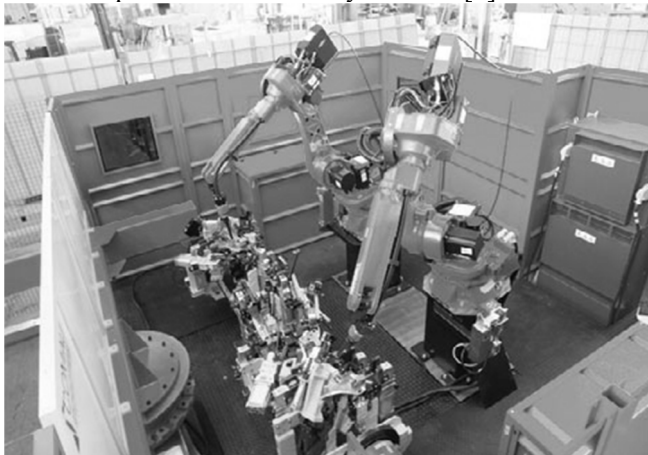
Niemniej jednak, nowym wyzwaniem w dziedzinie robotyki staje się współpraca robota z człowiekiem – jest to zauważalny i popularny trend w tej dziedzinie. Coraz częściej myśli się o tym, aby nie tylko tworzyć autonomiczne stanowiska zrobotyzowane, ale także zapewnić pracę robota „ramię w ramię” z człowiekiem [14, 17]. Zmienia to między innymi podejście do bezpieczeństwa – usuwane są wygradzenia i bariery bezpieczeństwa, a ich działanie zastępują się coraz nowszymi rozwiązaniami, włącznie z inteligentnym oprogramowaniem sterującym robotem [14, 18].

2. Zastosowanie robotów w przemyśle

Współczesny rynek produkcyjny wymusza na przedsiębiorcach ciągle doskonalenie procesów produkcyjnych [4]. Głównym kryterium celu staje się czas realizacji zleceń. Zakład produkcyjny chcąc zaspokoić potrzeby konsumentów musi sprawnie realizować procesy produkcyjne [3, 4]. Dlatego też coraz częściej w przedsiębiorstwach zastosowanie znajdują roboty przemysłowe. Jako główne korzyści wynikające z ich wdrożenia i wykorzystania podaje się [2, 5]:

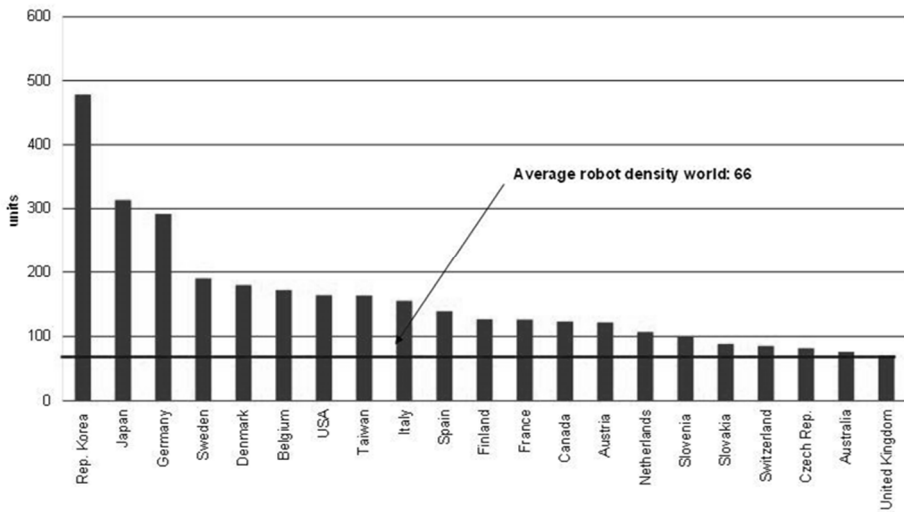
- szybkość – robot potrafi pracować szybciej od człowieka, a przy tym nie potrzebuje czasu na przerwy,
- precyzję i powtarzalność – współczesne roboty osiągają dokładność sięgającą nawet do tysięcznych części milimetra, a powtarzalność pozycji może wynosić $\pm 0,02$ mm.
- niezawodność – szacunkowy czas niezawodnej pracy robota to kilka lat,
- zwiększenie wydajności – jest to wynikiem szybkości pracy robota oraz wydłużeniem czasu pracy (praca bez przerw),
- pracę w trudnych warunkach – roboty do prac specjalnych mogą pracować bezpiecznie w warunkach szkodliwych dla człowieka (wysoka temperatura, duże ciężary, wysokie zapylenie, hałas, środki chemiczne).

Klasycznie roboty wykorzystywane są w zakładach produkcyjnych do wykonywania jednej, wyspecjalizowanej w wykonywaniu zazwyczaj jednej czynności (rysunek 1). Przykładami takich czynności są: paletyzacja, spawanie, lakierowanie, klejenie, montowanie, sortowanie. Zastąpienie człowieka robotem zwiększa wydajność na danym stanowisku, ponieważ robot może pracować nawet 24 godziny na dobę. Robot wykonujący swoją pracę, praktycznie za każdym razem wykona ją w taki sam sposób i w takim samym czasie [7].



Rys. 1. Zrobotyzowana stacja spawalnicza [1]

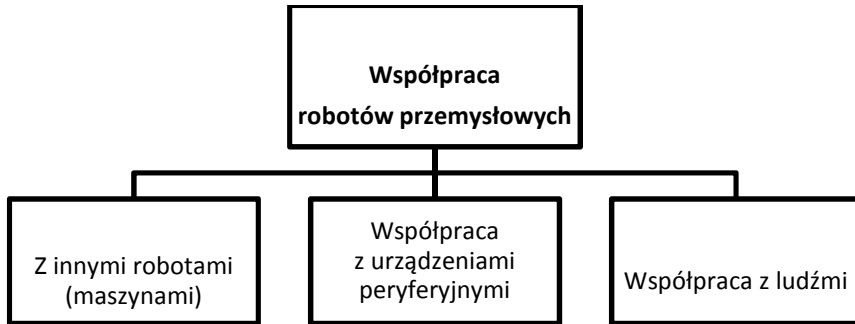
Międzynarodowa Federacja Robotyki opublikowała dane, z których wynika, że w roku 2015 nastąpił wzrost sprzedaży robotów przemysłowych o 8% (jest to prawie 240 000 maszyn) [11]. W roku 2014 zostały także przedstawione dane dotyczące najbardziej zrobotyzowanych przemysłów na świecie (wyrażonych w liczbie robotów przypadających na jednego pracownika przemysłu). Z opublikowanych danych wynika, iż najbardziej zrobotyzowany przemysł posiada Korea Południowa (478 maszyn na pracownika), następną jest Japonia (314 maszyn na pracownika), a kolejne są Niemcy (292 maszyn na pracownika). Średnia światowa wynosiła 66 robotów na pracownika (rys. 2) [12].



Rys. 2. Państwa z najwyższym stopniem zrobotyzowania przemysłu [12]

3. Współpraca robotów z ludźmi

Pojęcie „współpracy” w odniesieniu do robotów przemysłowych zazwyczaj rozumiane jest jako wymiana informacji pomiędzy poszczególnymi maszynami w zrobotyzowanym gnieździe wytwórczym. Niemniej jednak wyróżnić można inne kluczowe obszary współpracy robotów przemysłowych (rys. 3) [17].



Rys. 3. Rodzaje współpracy robotów przemysłowych [17]

Współpraca pomiędzy robotami występuje zazwyczaj w zrobotyzowanych gniazdach wytwórczych. Wówczas niezbędne jest zapewnienie odpowiedniej komunikacji, która realizowana jest przez typowe, przemysłowe protokoły komunikacyjne (PROFIBUS, Interbus S, MODBUS [8]). Czasami kilka robotów może być obsługiwanych za pomocą jednego kontrolera – wówczas nie jest konieczne stosowanie zewnętrznych protokołów [15]. Niejednokrotnie wyposażenie stanowiska zrobotyzowanego obejmuje także inne maszyny posiadające odpowiednie porty komunikacyjne. Wówczas robot może także współpracować z takimi urządzeniami. Mowa tu o stołach pozycjonujących, obrabiarkach CNC, czy maszynach dedykowanych dla konkretnych procesów (np. owijarki do palet).

Komunikacja robota przemysłowego z urządzeniami peryferyjnymi może być także rozpatrywana w aspekcie współpracy. Różnego rodzaju czujniki czy specjalne oprzyrządowanie wymaga niejednokrotnie komunikacji z robotem. Ponadto elementy chwytne robota (mechaniczne, pneumatyczne czy elektryczne) sterowane są z kontrolera robota, a więc wymagają odpowiedniej komunikacji [17].

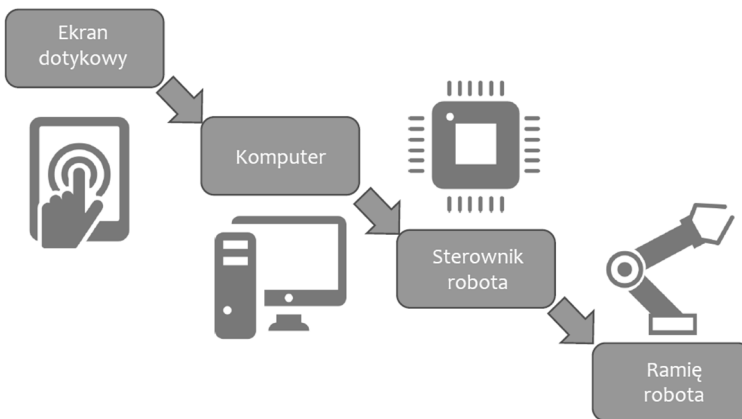
Trzeci rodzaj współpracy (kooperacja z człowiekiem) jest najnowszym kierunkiem rozwoju robotyki. Pomysłodawcy chcą odejść od barier bezpieczeństwa, zastępując je czujnikami wbudowanymi w roboty, systemami wizyjnymi oraz odpowiednim oprogramowaniem. Niektóre roboty można programować metodą uczenia przez demonstrację, co znacznie przyspiesza i ułatwia ich programowanie. Roboty tworzone z myślą o współpracy z człowiekiem często są mniejsze gabarytowo od typowych robotów przemysłowych. Ich kompaktowa wielkość pozwala na przenoszenie robota pomiędzy fragmentami linii produkcyjnej. Istnieje wiele gotowych rozwiązań proponowanych przez licznych producentów robotów. Przykładem mogą tu być roboty: ABB's YuMi, Baxter czy UR3 Universal Robots. Jednak roboty współpracujące z człowiekiem mają też wady, jak na przykład mniejsza maksymalna prędkość ruchów (podyktowana kwestiami bezpieczeństwa).

Dlatego też coraz częściej analizowane są możliwości adaptacji typowo przemysłowych robotów do pracy z ludźmi [1, 17, 18].

Sugerując się najnowszymi kierunkami rozwoju robotyki, autorzy prezentowanej pracy postanowili wykonać aplikację komputerową, która zapewni komunikację oraz elementy współpracy typowego robota przemysłowego Kawasaki RS003N z użytkownikiem.

4. Sterowanie robotem przemysłowym za pomocą dedykowanej aplikacji

Celem prezentowanej pracy było stworzenie aplikacji komputerowej, która pozwalałaby na sterowanie typowym robotem przemysłowym. Ponadto, w celu weryfikacji działania proponowanego rozwiązania, zbudowano odpowiednie stanowisko wyposażone w: komputer z ekranem dotykowym, router (zapewniający komunikację pomiędzy komputerem i robotem poprzez złącze Ethernet) oraz robota przemysłowego Kawasaki RS003N. Schemat działania proponowanego rozwiązania został przedstawiony na rys. 4.



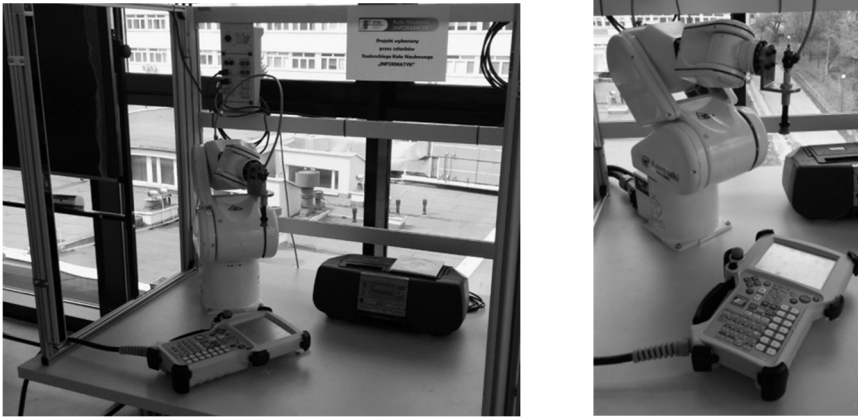
Rys. 4. Schemat działania proponowanego rozwiązania [opracowanie własne]

Użytkownik poprzez wybór polecenia na ekranie dotykowym komputera aktywował odpowiednią funkcję opracowanej aplikacji. Następnie z wykorzystaniem bibliotek DLL oraz łącza Ethernet do robota wysyłane było odpowiednie polecenie, które po przetworzeniu przez sterownik robota, realizowało odpowiednie ruchy ramienia (odpowiedni program robota). Całość projektu została przygotowana z myślą o Dniach Otwartych Politechniki Lubelskiej. Działanie robota polegało na wybraniu odpowiedniej płyty CD, umieszczeniu jej w odtwarzaczu oraz uruchomieniu muzyki. Aplikacja umożliwiała zatrzymanie odtwarzania oraz zmianę

muzyki. Dodatkowo każdy z operatorów mógł otrzymać od robota upominek – zdarzenie to również realizowane było za pomocą opracowanej aplikacji.

4.1. Opis zastosowanego robota

Prace przedstawione w niniejszej publikacji zostały przeprowadzone na stanowisku laboratoryjnym wyposażonym w robota przemysłowego Kawasaki RS003N wyposażonego w narzędzie pneumatyczne (przyssawkę). Jest to stanowisko wykorzystywane podczas typowych zajęć dydaktycznych (rys. 5). Ponieważ wykonana aplikacja miała zapewniać pełną interakcję z użytkownikiem – świetnie bariery bezpieczeństwa w które wyposażone jest stanowisko nie były podłączone do kontrolera robota.



Rys. 5. Stanowisko badawcze wyposażone w robota przemysłowego Kawasaki [opracowanie własne]

Robot Kawasaki RS003N jest przykładem manipulatora, a więc ma za zadanie odwzorowywać ruchy kończyny górnej człowieka. Manipulator przemysłowy jest to stacjonarnym układ ramion połączonych ze sobą przegubami, umożliwiającymi ich ruch [7]. Ruch wykorzystanego robota można programować za pomocą ręcznego programatora (Teach Pendant'a) lub przy pomocy dedykowanego oprogramowania K-ROSET. Teach Pendant ma charakter panelu operatorskiego, który umożliwia sterowanie oraz programowanie robota (za pomocą zapamiętywania pozycji lub poleceń języka AS). K-ROSET jest natomiast zaawansowanym oprogramowaniem, które pozwala symulować trajektorię ruchu robota, analizować czasy cykli, wyszukiwać potencjalne kolizje oraz analizować pozycje umieszczenia robota na stanowisku. K-ROSET podobnie jak inne programy tego typu umożliwia programowanie robota w trybie off-line. Odbywa się ono w języku AS. AS jest językiem programowania robotów stworzonym przez firmę Kawasaki. Jest on zaimplementowany w programie K-ROSET w postaci zbioru poleceń ruchu, obsługi

sygnałów, sterowania chwytakiem, itp. W aplikacji został zastosowany emulator kontrolera robota, co pozwala na praktycznie 100% zgodność wyników [16]. Za sterowanie ramieniem robota odpowiedzialny jest kontroler, który jednocześnie odpowiada za możliwość komunikacji z robotem poprzez odpowiednie protokoły komunikacyjne. Specyfikacja robota oraz kontrolera została przedstawiona w tabeli 1.

Tab. 1. Specyfikacja robota oraz kontrolera [10]

Specyfikacja robota	Specyfikacja kontrolera
Model: RS003N-A Typ: robot przegubowy Liczba stopni swobody: 6 Powtarzalność: $\pm 0,05$ mm Maksymalne obciążenie: 3 kg Maksymalna prędkość: 6000 mm/s Jednostka zasilająca: bezszczotkowy serwonapęd zasilany prądem zmiennym	Model: E70/E71 Konstrukcja: samo-wspierający Liczba sterowanych osi: 6 Typ sterowania: teach/repeat Sposób nauczania: język AS Napięcie zasilające: 200-240 V (50/60 Hz)

Proces programowania robota Kawasaki może odbywać się poprzez naukę „krok po kroku” lub napisanie programu za pomocą poleceń języka AS. Pierwsza z metod polega na wprowadzeniu do pamięci kontrolera danych dotyczących takich parametrów jak: współrzędne kartezjańskie położenia końcówki ramienia (lub wartości kątów wychylenia poszczególnych osi), prędkość ruchu robota, sygnału włączenia/wyłączenia narzędzia, precyzja ruchu ramienia, ewentualny czas oczekiwania na kolejny ruch. W rzeczywistości proces ten polega na zapisywaniu kolejnych położenia robota. Zapisane punkty łączą się w trajektorię ruchu robota. Programowanie robota za pomocą języka AS przypomina klasyczne programowanie wysokopoziomowe. Wykorzystując szereg poleceń (rozumianych przez kontroler) programista określa kolejne zachowania robota. Podobnie jak w większości języków programowania dane mają postać: stałych, zmiennych lokalnych lub globalnych. Standardowe są również typy danych: dane numeryczne, ciągi znaków alfanumerycznych w kodzie ASCII oraz wartości logiczne. Instrukcje języka AS obejmują polecenia sterujące wykonaniem programu, definiowania pozycji robota, ruchowe, sterujące prędkością i dokładnością osiągnięcia pozycji docelowej oraz narzędziem [10]. Możliwe są następujące rodzaje interpolacji:

- liniowa (LMOVE),
- kołowa (C1MOVE),
- liniowa w przestrzeni konfiguracyjnej manipulatora (JMOVE).

Instrukcje sterujące prędkością i dokładnością umożliwiają ustawienie: prędkości (SPEED) czy dokładności osiągnięcia pozycji (ACCURACY). Sterowanie narzędziem odbywa się za pomocą instrukcji uruchomienia (OPEN) i wyłączenia (CLOSE) narzędzia. Do sterowania wykonaniem programu wykorzystuje się również standardowe instrukcje wyboru (IF, THEN, ELSE), pętle (FOR, DO, WHILE) i instrukcję skoku (GOTO). Na rys. 6 został przedstawiony przykład prostego programu polegającego na rozpoczęciu ruchu z punktu #start, dojechaniu do punktu pobrania #p001 i pobraniu elementu za pomocą chwytaka.

```
.PROGRAM point1()

JMOVE #start
JAPPRO #p001,50
JMOVE #p001
CLAMP 1
JAPPRO #pb_1,50

.END
```

Rys. 6. Fragment kodu napisanego w języku AS [opracowanie własne]

Autorzy niniejszej pracy wykorzystali metodę programowania robota za pomocą ręcznego programatora. Dla każdego ze zdarzeń (np. „weź płytę nr 1 i umieść ją w odtwarzaczu”, „podaj element dla operatora”) za pomocą Teach Pendant’a został opracowany program robota zapisany w pamięci kontrolera.

Ważnym elementem programowania ruchu robota jest weryfikacja wykonanego programu. Ma ona na celu sprawdzenie poprawności pracy robota, wprowadzeniu ewentualnej korekty, uniknięciu kolizji lub zoptymalizowaniu ruchy robota. Proces ten realizowany jest za pomocą ręcznego programatora, który umożliwia zmianę sterowania pomiędzy trybami „TEACH/REPEAT” (z ang. „naucz/powtórz”). Testowanie programu może odbywać się krokowo lub płynnie. Ponadto możliwe jest wykonywanie opracowanego programu w pętli. Podczas realizowanych prac funkcja ta była bardzo pomocna, gdyż precyzyjna praca robota na przedstawionym stanowisku wymagała niejednokrotnie wprowadzania wielu korekt w trakcie programowania ramienia.

4.2. Środowisko programistyczne

W celu stworzenia aplikacji sterującej robotem wykorzystano środowisko Microsoft Visual Studio 2010 w wersji Express. Visual Studio to wszechstronne zintegrowane środowisko programistyczne firmy Microsoft, przeznaczone do tworzenia zróżnicowanych aplikacji, pozwalające na kompleksową realizację

projektu (w ramach jednego zagadnienia istnieje możliwość realizowania kilku odrębnych zadań). Środowisko to pozwala na wykonanie zintegrowanych projektów, będących kombinacją aplikacji okienkowych, internetowych, wykonywanych z linii poleceń czy web serwisów [6]. Cechy charakterystyczne Visual Studio to [13]:

- Jeden IDE (zintegrowany interfejs użytkownika) dla kilku języków i różnych typów projektów,
- pozwala na połączenie z przeglądarką Internetową,
- dostarcza narzędzi do debugowania,
- posiada dostosowany do oczekiwań użytkownika interfejs.

Aplikacja została wykonana w języku Visual Basic, który charakteryzuje się względnie prostą składnią, a ponadto posiada wsparcie ze strony firmy Kawasaki.

W aplikacji zastosowano bibliotekę DLL dedykowaną dla robotów Kawasaki. Jest to biblioteka AsKCT składająca się z odpowiednich funkcji. W opracowanej aplikacji zastosowanie znalazły [9]:

- funkcje AsKCTInit() i AsKCTDestroy() – inicjujące oraz kończące komunikację pomiędzy robotem a aplikacją,
- funkcje AsKCTConnect() i AsKCTDisconnect() – pozwalające nawiązać połączenie za pomocą łącza Ethernet,
- funkcja AsKCTCommand() – umożliwiająca przesyłanie do robota odpowiedniego polecenia (uruchomienie odpowiedniego programu/polecenia).

4.3. Budowa programu sterującego robotem

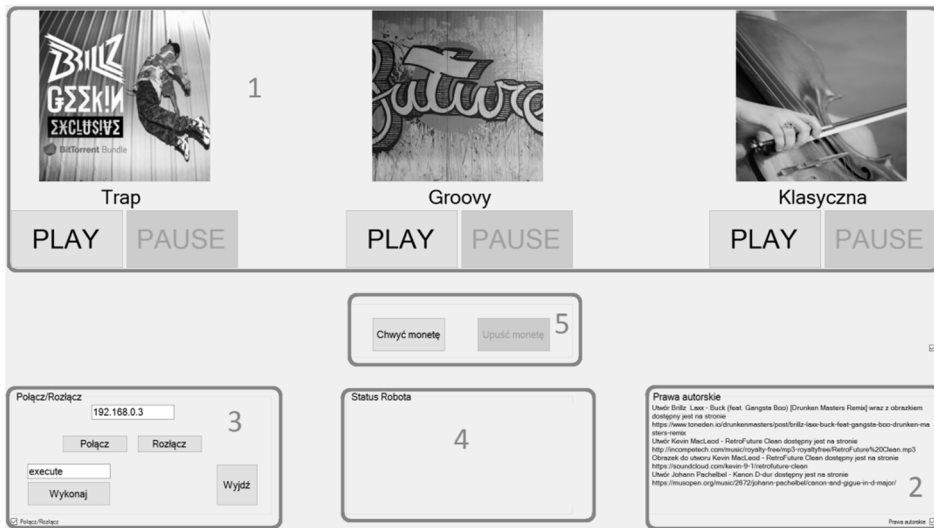
Wygląd aplikacji został opracowany tak, aby ułatwić jego wyświetlanie na dużym ekranie dotykowym oraz zapewnić łatwy i prosty dostęp do poszczególnych funkcji programu (rys. 8). W górnej części ustawiono obrazki przedstawiające dany styl muzyczny oraz przyciski do wydania poleceń – uruchamiania, bądź zatrzymywania muzyki (1). Pod każdym z przycisków umieszczono odpowiednie polecenia języka Visual Basic oraz biblioteki AsKCT (rys. 7)

```
Dim ret As Integer
ret = AsKCTCommand(0, "execute cd1")

If ret <> 0 Then
    MessageBox.Show("Connect Error : " & ret)
Else
    MessageBox.Show("Odtwarzam CD1", "STATUS")
End If
```

Rys. 7. Fragment kodu – uruchomienie programu odtwarzania płyty nr 1 [opracowanie własne]

W prawym dolnym rogu umieszczone zostały informacje dotyczące wykorzystanych materiałów źródłowych (obrazów oraz muzyki) (2). W lewym dolnym rogu zamieszczono przyciski przeznaczone do użytku przez osobę nadzorującą pracą robota (3) – jest to m.in. obszar do manualnego przekazywania poleceń oraz okno wprowadzania adresu IP robota i ustanawiania połączenia. W dolnej części menu (4) znajduje się kontrolka GroupBox w której wyświetlany jest obecny status robota. W centralnej części aplikacji znajdują się dwa przyciski odpowiadające za program do podawania upominków w czasie prezentacji (5). Obszary 2, 3 oraz 5 są domyślnie ukryte i mogą zostać pokazane przez naciśnięcie odpowiedniej kontrolki CheckBox.



Rys. 8. Wygląd programu do sterowania robotem przemysłowym Kawasaki [opracowanie własne]

Ważnym aspektem pracy programu była dezaktywacja jego przycisków w trakcie wykonywania poleceń przez ramie robota. Zbyt duża liczba przesyłanych danych powoduje zawieszenie połączenia z robotem, a także błędy pracy kontrolera. Zastosowano zatem czasową blokadę wybranych przycisków poprzez zastosowanie kontrolerek typu Timer, które czasowo dezaktywowały większość przycisków (rys. 9).

```
Private Sub TPlay1_Tick(ByVal sender As System.Object, e As
System.EventArgs) Handles TPlay1.Tick

    i += 1
    If i = 55 Then
        STOPl.Enabled = True
        TPlay1.Stop()
        i = 0
    End If
End Sub

Private Sub TStop1_Tick(ByVal sender As System.Object, e As
System.EventArgs) Handles TStop1.Tick

    i += 1
    If i = 50 Then
        PLAY1.Enabled = True
        PLAY2.Enabled = True
        PLAY3.Enabled = True
        TStop1.Stop()
        i = 0
    End If
End Sub
```

Rys. 9. Fragment kodu – zastosowanie kontrolki Timer [opracowanie własne]

5. Podsumowanie

Roboty przemysłowe znacząco zwiększają produktywność oraz elastyczność produkcji zakładów przemysłowych, więc prognozowany wzrost ich sprzedaży w kolejnych latach nie jest bezzasadny. Czynnikiem ludzki jest i dalej będzie potrzebny w każdym przedsiębiorstwie, niezależnie od jego zrobotyzowania, ponieważ każdy robot potrzebuje operatora. Znajomość języków programowania oraz umiejętne sterowanie robotem pozwala na zaprojektowanie oraz optymalizację aplikacji dedykowanych dla konkretnych procesów (takich jak spawanie, montaż, czy sortowanie). Odpowiednio wykonana aplikacja umożliwia intuicyjną, sprawną i bezpieczną komunikację pomiędzy operatorem, a robotem.

Autorzy w niniejszej pracy zaprezentowali wykorzystanie typowego środowiska programistycznego do stworzenia aplikacji sterującej dla robota przemysłowego Kawasaki RS003N. Przedstawione oprogramowanie realizowało tylko kilka z wielu dostępnych zastosowań robota, jednak niewątpliwie istnieją możliwości rozbudowy proponowanego rozwiązania o chociażby możliwość sterowania robotem za pomocą urządzeń mobilnych z wykorzystaniem połączenia WiFi czy budowę układów sterowania opartych o platformę Arduino, bądź Raspberry PI.

Literatura

1. AutomatykaOnline.pl: Robotyzacja i automatyzacja procesów wytwórczych – <http://automatykaonline.pl/Artykuly/Robotyka/Robotyzacja-i-automatyzacja-procesow-wytworczych>
2. EVOTEC: Dlaczego robot? – <http://evotec.com.pl/dlaczego-robot.html> [data dostępu: 17-11-2015]
3. Gola A., Sobaszek Ł., Świć A.: Selected Problems of Modern Manufacturing Systems Design and Operation. Robotics and Manufacturing Systems, Lublin, 2014, p. 56–68.
4. Gola A.: Economic Aspects of Manufacturing Systems Design. Actual Problems of Economics, 156, 6 (2014), pp. 205–212.
5. Hajduk M., Koukolová L.: Trends in Industrial and Service Robot Application. Applied Mechanics and Materials, Vol. 791, 2015, pp. 161–165.
6. Halvorson M.: Microsoft Visual Basic 2010 Step by Step, Microsoft Press, Washington, 2010.
7. Honczarenko J.: Roboty Przemysłowe. Budowa i zastosowanie, Wydawnictwa Naukowo-Techniczne, Warszawa, 2004.
8. Kaczmarek W.: Elementy Robotyki Przemysłowej. WAT, Warszawa, 2008.
9. Kawasaki Robot Materials, KCwinTCP-dll – Instruction Manual.
10. Kawasaki Robot Materials, RS003NFE70 – Robot Specification.
11. Materiały Międzynarodowej Federacji Robotyki: <http://www.ifr.org/news/ifr-press-release/president-s-report-807> [data dostępu: 15-05-2016]
12. Materiały Międzynarodowej Federacji Robotyki: <http://www.ifr.org/news/ifr-press-release/survey-13-million-industrial-robots-to-enter-service-by-2018-799> [data dostępu: 15-05-2016]
13. Pilat M.: Projekt multimedialnego serwisu internetowego programowania w C#. Politechnika Lubelska, 2016.
14. Portal gospodarczy WNP.PL: Współpraca człowiek-robot w procesie produkcyjnym Audi – http://motoryzacja.wnp.pl/wspolpraca-czlowiek-robot-w-procesie-produkcyjnym-audi,244424_1_0_0.html
15. Semjon J., Baláž V., Vagaš M.: Project multirobotic systems with KUKA robots in cooperation with VW Slovakia, in: L. Koukolová, A. Świć (Eds.), Robotics and manufacturing systems, Lublin, 2014, pp. 33-38.
16. Sobaszek Ł., Gola A., Varga J.: Virtual Designing of Robotic Workstations. Applied Mechanics and Materials, Vol. 844, pp. 31-37.

17. Sobaszek Ł., Gola A.: Perspective and methods of human – industrial robots cooperation. *APPLIED MECHANICS AND MATERIALS*, 2015, vol. 791, pp. 178-183.
18. The Engineer: Robot revolution: Humans and droids, working together – <http://www.theengineer.co.uk/manufacturing/automation/robot-revolution-humans-and-droids-working-together/1019569.article>

Abstrakt

Roboty znajdują szerokie zastosowanie w dzisiejszym przemyśle. Najczęściej wykorzystuje się je, aby zastąpić ludzi w pracy na stanowiskach niebezpiecznych, bądź uciążliwych. Niemniej jednak, coraz bardziej popularnym trendem w robotyce staje się współpraca robota z człowiekiem. Zagadnienie to stanowi nowe wyzwanie w tej dziedzinie – usuwane są bowiem bariery bezpieczeństwa, a ich działanie zastępowane jest inteligentnym oprogramowaniem robota.

Celem niniejszego artykułu jest przedstawienie projektu zespołowego dotyczącego stworzenia aplikacji komputerowej, która umożliwi sterowanie robotem przemysłowym. W pracy omówiono także ideę współpracy robotów z ludźmi, przedstawiono stanowisko badawcze wyposażone w robota Kawasaki RS003N na którym realizowano prace, zaprezentowano aplikację opracowaną w języku Visual Basic, jak i metodykę programowania robota.

Słowa kluczowe: robot przemysłowy, programowanie robotów, Visual Basic, współpraca człowiek-robot

Abstract

Nowadays, robots are widely used in the industry. Mainly purpose of its implementation is to replace human at the danger and oppressive work stations. However, an increasingly popular trend in robotics is a human-industrial robots cooperation. This issue represents a new challenge in this area.

The paper presents the results of group project on creating software which helps to industrial robot control. First of all, the idea of a human-industrial robots cooperation was presented. Moreover, the laboratory station equipped with a Kawasaki RS003N robot was outlined. In the final part of the paper the authors discussed the prepared software and industrial robot programming methodology.

Keywords: industrial robot, robots programming, Visual Basic, human-robot cooperation