

Roboty Mobilne

Microsoft Robotics Developer Studio

dr inż. Tomasz Krzeszowski

2017-05-20

Spis treści

1	Przygotowanie do laboratorium.....	3
2	Cel laboratorium.....	3
3	Microsoft Robotics Developer Studio	3
3.1	Wprowadzenie	3
3.2	Przykład 1: Detekcja zderzenia z przeszkodą	3
3.3	Przykład 2: Ręczne sterowanie robotem.....	4
4	Zadania do samodzielnego wykonania.....	6
5	Literatura.....	7

1 Przygotowanie do laboratorium

Przed zajęciami należy się zapoznać z niniejszą instrukcją, w szczególności z funkcjonalnością poszczególnych bloków i serwisów (zob. 3.1).

2 Cel laboratorium

Celem laboratorium jest zapoznanie studentów z możliwościami oprogramowania Microsoft Robotics Developer Studio.

3 Microsoft Robotics Developer Studio

3.1 Wprowadzenie

Microsoft Robotics Developer Studio jest środowiskiem do tworzenia aplikacji na różne platformy robotów. Wszelkie informacje wraz z przykładami i samouczkami można znaleźć na stronie producenta [1].

Opis podstawowych aktywności: <https://msdn.microsoft.com/en-us/library/bb466256.aspx>

Opis serwisów użytkowych: <https://msdn.microsoft.com/en-us/library/dd126877.aspx>

W celu wyświetlenia pomocy należy wcisnąć przycisk F1.

3.2 Przykład 1: Detekcja zderzenia z przeszkodą

W przykładzie zostanie przedstawione:

- dodawanie i konfigurowanie nowych aktywności;
- łączenie aktywności;
- dodawanie i konfigurowanie okna powiadomienia ("Dialog Box");
- wybór rodzaju sensora;
- uruchamianie aplikacji.

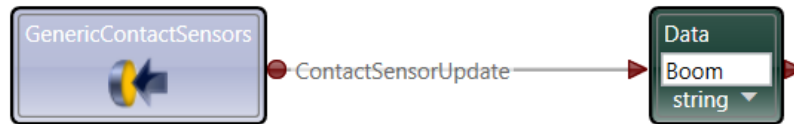
Krok 1: Utwórz nowy projekt.

Krok 2: Dodaj i skonfiguruj nowe aktywności:

- przeciągnij "Generic Contact Sensors" z panelu serwisów na główny panel;
- dodaj aktywność "Data" i ustaw jej typ na "string" oraz ustaw jej zawartość na komunikat, który ma się wyświetlić po zderzeniu z przeszkodą.

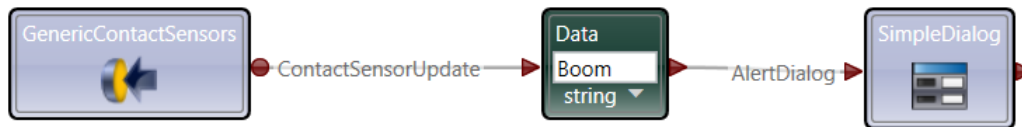
Krok 3: Połącz aktywności:

- połącz wyjście powiadomienia (ang. *notification*) "Generic Contact Sensors" z wejściem bloku "Data", powinno się wyświetlić okno "Connections". Należy wybrać opcję "ContactSensorUpdate". Połączenie to wysyła wiadomość "ContactSensorUpdate" z sensora kontaktu do akcji "Create" aktywności "Data". Aktywność "Data" wykorzystuje wiadomość z sensora do rozpoczęcia utworzenia danych (łańcucha) wprowadzonych do "Data".



Krok 4: Dodaj i skonfiguruj okno Dialog Box:

- dodaj serwis "Simple Dialog";
- połącz aktywność "Data" z "Simple Dialog", pojawi się okno "Connections". Należy wybrać "DataValue" w kolumnie "From" i "AlertDialog" w kolumnie "To". Powinno się pojawić okno "Data Connection", w którym należy wybrać wartość "value" do przesłania do formatki "Alert" wyświetlanej przez "Simple Dialog".



Krok 5: Określenie sensora kontaktu.

- Aktywność "GenericContactSensor" jest ogólną reprezentacją sensorów kontaktu dla różnych robotów, z tego powodu przed uruchomieniem programu należy określić, którego sensora chcemy użyć.
- W celu określenia sensora należy ustawić do serwisu plik manifestu, który zawiera sensor kontaktu.
- Należy jeszcze dodać serwis "SimpleDashboard", który umożliwi sterowanie robotem. Serwis ten nie musi być połączony z żadnym innym blokiem.

Krok 6: Uruchomienie aplikacji:

- w celu uruchomienia aplikacji wybieramy polecenie Start z menu Run, wciskamy przycisk F5 lub przycisk interfejsu oznaczony zieloną strzałką;
- sterujemy robotem tak aby uderzyć zderzakami w przeszkodę;
- po zderzeniu powinien się wyświetlić komunikat z łańcuchem wprowadzonym do bloku "Data".

3.3 Przykład 2: Ręczne sterowanie robotem

W przykładzie zostanie przedstawione:

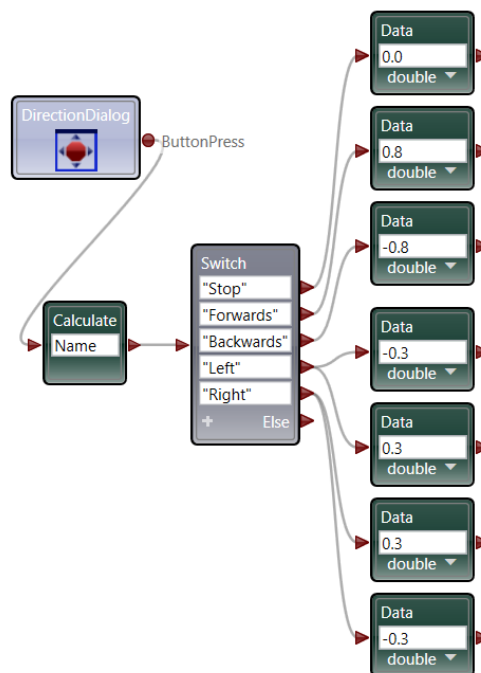
- definiowanie operacji do wykonania;
- stosowanie alternatywnych przepływów;
- uruchamianie aplikacji.

Krok 1: Utwórz nowy projekt.

Krok 2: Dodaj serwis "DirectionDialog". Serwis ten udostępnia proste okno z przyciskami do sterowania robotem.

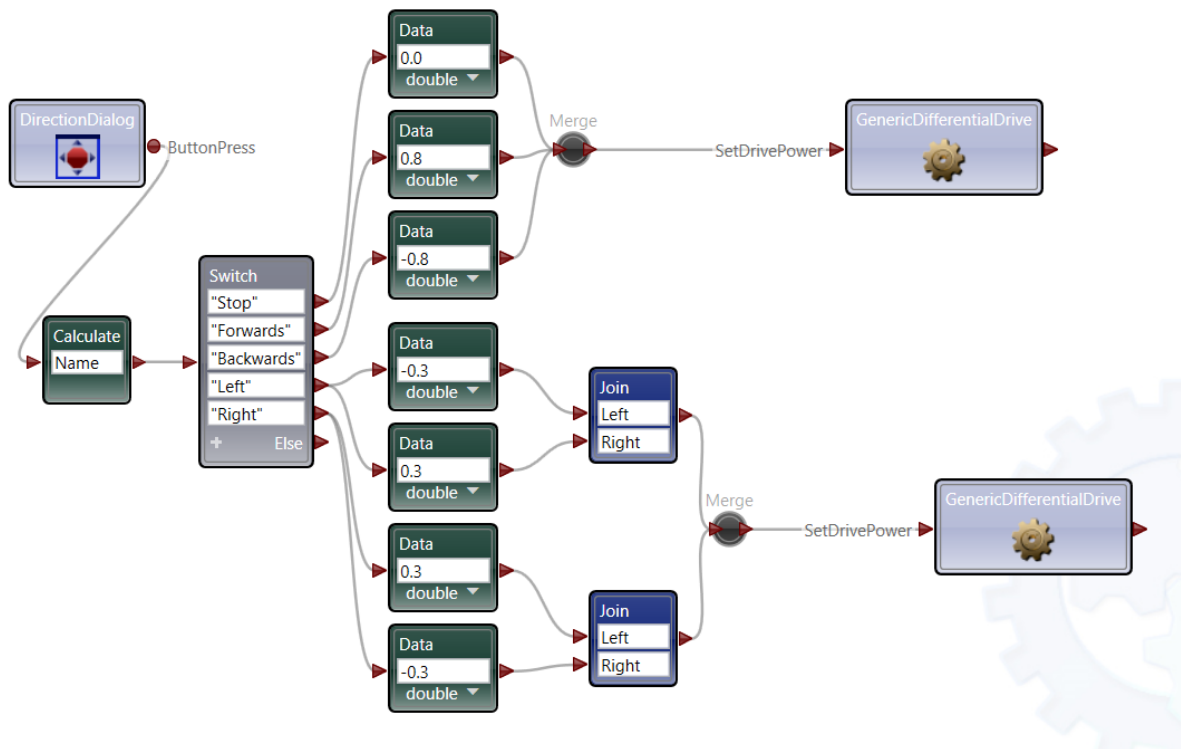
Krok 3: Definiowanie operacji do wykonania:

- dodaj serwis "DirectionDialog" oraz bloki "Calculate" oraz "Switch";
- połącz wejście bloku "Calculate" z wyjściem powiadomień (powiadomienie o wciśniętym przycisku - "ButtonPressed") serwisu "DirectionDialog", w bloku Calculate wprowadź wartość "Name" w celu pobrania nazwy wciśniętego przycisku;
- połącz wejście aktywności "Switch" z wyjściem "Calculate" i dodaj do niej następujące pola: "Stop", "Forwards", "Backwards", "Left", "Right", pola te odpowiadają nazwom przycisków "DirectDialog";
- dodaj 7 aktywności "Data" (aktywności te określają wartość mocy dla silników napędu) i połącz je z wyjściami aktywności "Switch":
 - 1 do wyjścia "Stop" o wartości 0.0
 - 1 do wyjścia "Forwards" o wartości 0.8
 - 1 do wyjścia "Backwards" o wartości -0.8
 - 2 do wyjścia "Left" o wartościach -0.6 i 0.6
 - 2 do wyjścia "Right" o wartościach 0.6 i -0.6



- dodaj dwie aktywności "Join" i ustaw etykiety ich lokalnych zmiennych na "Left" i "Right";
- dla każdego "Join" połącz dwa bloki "Data" idące od wyjścia "Left" do jednego bloku "Join" i dwa bloki "Data" idące od wyjścia "Right" do drugiego bloku "Join", aktywność "Join" pobiera dwie wartości wejściowe i przesyła je dalej jako jeden przepływ;
- dodaj dwa bloki "Merge", do jednego z nich połącz 3 pozostałe bloki "Data", a do drugiego wyjścia bloków "Join". Musimy mieć dwa bloki "Merge" ponieważ w pierwszym ustawiamy tą samą wartość mocy na oba silniki robota, natomiast w drugim przypadku nie;

- dodaj dwa serwisy "GenericDifferentialDrive" (oba muszą reprezentować ten sam silnik - tworzymy jedną instancję serwisu) i ustaw plik z konfiguracją (manifest) z obsługą napędu różnicowego;
- połącz wyjścia "Merge" z serwisem "GenericDifferentialDrive" (przy łączeniu wybierz "SetDrivePower" w celu ustawienia mocy dla poszczególnych silników);
- dodaj plik manifestu dla "GenericDifferentialDrive".



Krok 4: Zastosuj alternatywny przepływ:

- zastąp funkcjonalność przycisku "Stop" zdarzeniem "ButtonReleased" z serwisu "DirectDialog" (robot powinien się zatrzymać po puszczeniu przycisku).
- Wskazówka - dodaj dodatkowy "DirectionDialog" oraz "GenericDifferentialDrive".

4 Zadania do samodzielnego wykonania

Z wykonywanych zadań laboratoryjnych należy przygotować sprawozdanie. Sprawozdanie powinno zawierać:

- a) stronę tytułową;
- b) opis realizowanych zadań;
- c) wnioski.

Sprawozdanie należy przesłać za pomocą aplikacji internetowej znajdującej się pod adresem:
<http://sprawozdania.kia.prz.edu.pl/>

Na stronie należy założyć konto i zapisać się na odpowiednie zajęcia.

Zadania do wykonania:

1. Uruchomić i zapoznać się z symulatorem środowiska robota (ruch kamerą, ruch robota, edycja otoczenia):
 - uruchom scenę "Multiple Simulated Robots";
 - przetestuj możliwości poruszania się po scenie, rodzaje widoków (zakładka "Render"), dostępne kamery (zakładka "Camera");
 - przejdź do trybu edycji (Mode->Edit) i zmień położenie wybranego obiektu na scenie, spróbuj dodać nowy obiekt;
 - w trybie działania ("Run") połącz panel sterowania (Dashboard), a następnie przetestuj możliwości sterowania robota oraz symulację dalmierza laserowego.
2. Uruchom Visual Programming Language (VPL)
3. Zapoznaj się z aktywnościami i serwisami dostępnymi w VPL
4. Zrealizuj przykład 1
5. Zrealizuj przykład 2
6. Zaprojektuj prostą aplikację umożliwiającą omijanie przeszkód przez robota (gdy robot wykryje przeszkodę wykonuje skręt i jedzie dalej), należy wykorzystać dalmierz laserowy "SickLaserRangeFinder". Wskazówki:
 - w celu symulacji można wykorzystać symulator dalmierza "SimulatedLaserRangeFinder" jak i reprezentację rzeczywistego dalmierza "SickLaserRangeFinder" z konfiguracją ustawioną na "SimulatedLaserRangeFinder";
 - dane z dalmierza znajdują się w tablicy "DistanceMeasurements".
7. Udoskonal aplikację z punktu 6, tak aby robot rozpoznawał, z której strony jest przeszkoda i wykonywał skręt w przeciwnym kierunku.

5 Literatura

1. Strona projektu: <https://msdn.microsoft.com/en-us/library/bb881626.aspx>
2. K. Johns, T. Taylor, "Professional Microsoft Robotics Studio", Wiley Publishing, 2008
3. S. Morgan, "Programming Microsoft Robotics Studio", Microsoft Press, 2008