

Programowanie Autonomicznych Robotów Mobilnych

Lab. 3 – Sterowanie robotem FIRA z wykorzystaniem akcelerometru

wersja. 3.2, 2022.09.28



1 Przygotowanie do zajęć

W sali laboratoryjnej dostępny jest sprzęt komputerowy z potrzebnym oprogramowaniem.

Planując pracę na własnym laptopie, proszę zainstalować:
Android Studio 2020.3.1:

<https://developer.android.com/studio/archive>

Przed zajęciami proszę skutecznie zainstalować środowisko oraz SDK 21 dla Androida 6.

Proszę też skutecznie zbudować aplikację FiraAccelerometer:

- trzeba zaimportować aplikację do IDE,
- zainstalować wszystkie wymagane przez IDE elementy i narzędzia,
 - UWAGA! - nie należy aktualizować wersji Gradle,
- projekt powinien budować się bez błędów.



Na zajęciach będzie dostępny telefon, ale można też użyć własnego. W takiej sytuacji proszę przetestować uruchamianie aplikacji na telefonie.

2 Uruchamianie i programowania robota

2.1 Robot FIRA

FIRA to organizacja zajmująca się standaryzowaniem lig robotów mobilnych grających w piłkę. Roboty ligi Mirosot to sześciangy o boku 75mm napędzane dwoma niezależnymi silnikami. W drużynie gra 3, 5 lub 11 robotów, które są sterowane zdalnie przez jeden komputer.

<https://www.youtube.com/watch?v=s5y-0SPdnVQ>

Stworzenie skutecznej drużyny wymaga rozwiązania kilku ważnych problemów robotyki, jak lokalizacja na podstawie barwnych znaczników, planowanie, koordynacja, przewidywanie, sterowanie i komunikacja bezprzewodowa.

Nasze roboty typu Miabot są sterowane przez interfejs Bluetooth. Podstawowa komenda przekazywana do robotów to: $[=15, 30]$, gdzie 15 to prędkość koła lewego, a 30 prawego.

2.2 Aplikacja na system Android sterująca robotem FIRA

Aby uruchomić i przetestować aplikację na telefonie:

1. Podłączyć telefon do komputera kablem USB.
 - Telefon musi mieć włączone opcje developerskie i debugowanie przez USB
2. Włączyć Miabota przełącznikiem.
3. Sparować telefon z Miabotem przez Bluetooth w ustawieniach telefonu (pin: 1234).
4. Uruchomić aplikację z poziomu IDE.
5. Wybrać jako docelowe urządzenie podłączony telefon.
6. Po uruchomieniu na telefonie powinien pojawić się czarny ekran.
W menu należy wybrać opcję *Connect a device* i ponownie podać pin.
7. Aby sprawdzić czy połączenie się udało należy nacisnąć przycisk pogłaśniania i zaobserwować czy robot się porusza.

W katalogu *app/java/* znajdują się klasy tworzące aplikację. Klasy *BluetoothCommandService* i *DeviceListActivity* odpowiadają za komunikację między telefonem a Miabotem przez bluetooth, oraz za wykrywanie dostępnych urządzeń – nie będziemy ich zmieniać.

Główną klasą odpowiadającą za działanie aplikacji jest **klasa *MainActivity***. Zawiera ona szereg niezbędnych metod callbackowych odpowiadających za poprawną inicjalizację, funkcjonowanie, wstrzymywanie aplikacji itp. (*onCreate()*, *onStart()*, *onResume()*, *onPause()*...). Wykonanie ćwiczenia zasadniczo nie będzie wymagało modyfikacji tych metod.

Ważniejsze elementy aplikacji:

onSensorChanged() - metoda wywoływana przy każdej zmianie wartości używanego czujnika.

<https://developer.android.com/reference/android/hardware/SensorListener.html>

W ćwiczeniach użyjemy akcelerometru. W początkowej wersji metoda wypisuje do logu aktualne przyspieszenie w trzech kierunkach.

Tutaj należy umieścić główną część kodu realizującą ćwiczenia.

onKeyDown() - metoda wywoływana w momencie wciśnięcia konkretnych przycisków telefonu.

<https://developer.android.com/reference/android/view/KeyEvent.Callback.html>

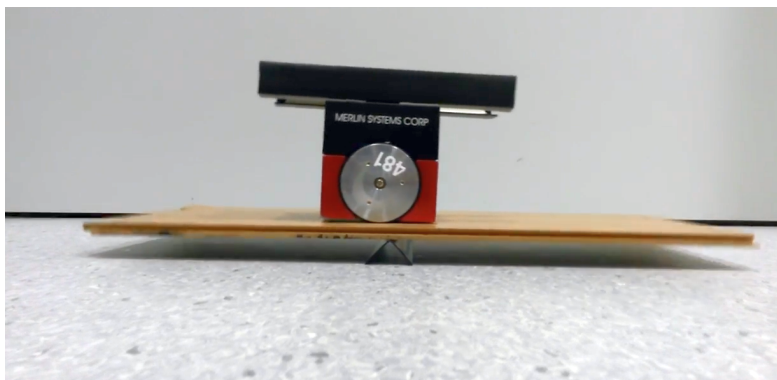
W wersji początkowej zawiera kod pozwalający na poruszanie się robota przy wciśnięciu przycisku volume up.

sendSpeedCommand() - Metoda pozwalająca wysłać rozkaz do Miabota przez bluetooth. Jako argumenty należy podać wartości prędkości odpowiednio dla lewego i prawego koła. Robot będzie poruszał się z zadaną prędkością aż do momentu otrzymania kolejnego rozkazu.

3 Balansowanie na platformie

Pierwszym zadaniem jest napisanie autonomicznego sterownika, który zrealizuje sterowanie balansowania na platformie. Telefon leżąc płasko na robocie jest w stanie zmierzyć stopień wychylenia robota i pokierować go „pod górę”.

Przykładowe zachowanie robota: <https://www.youtube.com/watch?v=acRvO2eLUo>



4 Telefon jako zdalna kierownica

Akcelerometr telefonu można wykorzystać do odczytania decyzji człowieka. Jest to wygodna metoda uzyskania płynnego i intuicyjnego sterowania robotem. Program powinien:

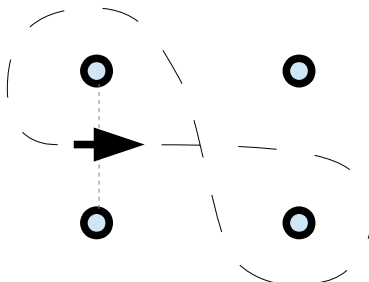
- Aktywować sterowanie przy wciśnięciu przycisku *Volume-up*.
- W chwili aktywowania sterowania zapamiętywać położenie telefonu – będzie to pozycja w której robot jest zatrzymany.
- Wychylenia telefonu należy przełożyć na sterowanie kołami.

<https://www.youtube.com/watch?v=MEoLx3vgxYk>

Celem dostosowania parametrów programu jest uzyskanie możliwie *intuicyjnego* sterowania, czyli takiego, które nie wymaga długiej nauki i pozwala na szybkie i precyzyjne pokonywanie tras.

5 WYŚCIG!

Jakość opracowanej mej metody sterowania zweryfikujemy w Wielkim Wyścigu !
Oba zespoły jednocześnie ścigają się na torze wokół nóg krzesła. Wyścig ma 3 okrążenia.
Zespół lepszy w turnieju 3 wyścigów wygrywa.



Zakończenie zajęć - proszę skasować aplikację FiraAccelerometer z telefonu.