

Ćwiczenie 2

Podstawowe sterowanie robotem Mindstorms NXT w bibliotece NXT++



Zadanie do wykonania

- 1) Tworzymy na pulpicie katalog w formacie Imię.Nazwisko,
- 2) Rozpakowujemy plik nxtp0-6-3-STARTER.rar znajdujący się na dysku D: do utworzonego katalogu. Jeżeli na dysku D: brakuje pliku ściągamy alternatywnie z linku:
`http://wmii.uwm.edu.pl/~artem/nxt-starter.rar`
- 3) Wybieramy przy asyście wykładowcy, wybrany model robota Mindstorms NXT,
 - sprawdzamy poziom baterii, w razie potrzeby podłączamy do ładowania,
 - rozpoznajemy sensory podłączone do kostki sterującej i zapisujemy numery portów, do których są podłączone,
 - zapisujemy numery portów, do których podłączone są serwo motory,
 - zdobyte informacje będą potrzebne do ustawienia oprogramowania.
- 4) Zapoznajemy się z możliwościami sterowania serwomechanizmami robota Mindstorms NXT i używania sensorów w bibliotece NXT++, a następnie przechodzimy do kolejnych poleceń. Podstawy sterowania załączone są w sekcji Biblioteka NXT++ - wprowadzenie.
- 5) Uruchamiamy plik re4.sln w katalogu,

`Imię.Nazwisko\nxtp0-6-3-STARTER\FINAL\examples\vs10\re4`

- 6) Załączone oprogramowanie w pliku źródłowym main solucji re4, pozwala na podstawowe sterowanie zdalne robotem Mindstorms NXT. Oprogramowanie ma charakter filtru, pętli nieskończonej, w której następuje nasłuch wciśnięcia klawisza. Wzorując się na istniejącym sterowaniu należy napisać obsługę czujnika koloru (lub

światła) oraz sonaru, w celu użycia czujników w zadaniach opisanych w kolejnych podpunktach.

7) Implementujemy program śledzenia linii z prawej i lewej strony. Nasycenie światła linii i tła pobieramy zdalnie sterując robotem. W laboratorium mamy dostępne dwa typy czujników wykrywających nasycenie światła, prosty czujnik światła oraz czujnik koloru, który może być użyty jako czujnik nasycenia światła. Przed rozpoczęciem programowania należy sprawdzić w jakiego typu czujnik wyposażony jest używany robot.

8) Implementujemy program symulujący zachowanie zawodnika sumo, robot nie może wyjechać poza czarną obręcz, gdy wykrywa przeszkodę sonarem, stara się przesunąć przeszkodę poza czarną linię.

Wprowadzenie do biblioteka NXT++

Podstawowe moduły i przestrzenie nazw

```
#include "NXT++.h"
#include <iostream>
#include <conio.h>
using namespace std;
using namespace NXT;
```

Nawiązywanie komunikacji z robotem

```
za pomocą USB,
if (NXT::Open(&comm)) //initialize the NXT and continue if it succeeds
{
}
NXT::Close(&comm); //close the NXT

za pomocą Bluetooth ,
if (NXT::OpenBT(&comm)) //initialize the NXT and continue if it succeeds
{
}
NXT::Close(&comm); //close the NXT
```

Ramy programu NXT++

```
NXT::StartProgram(&comm,"program1");
kod programu,
NXT::StopProgram(&comm);
```

Wyświetlanie poziomu baterii i ilości dostępnej pamięci

```
cout<<NXT::BatteryLevel(&comm);
cout<<NXT::GetAvailableFlash(&comm);
```

Używanie generatora dźwięków

```
for (int i=1; i<1000; i=i+10){  
    NXT::PlayTone(&comm, i, 1);  
    Wait(1);  
}
```

Nasłuchiwanie wciśnięcia klawisza - przydatne w odświeżaniu parametrów sterujących

```
if (kbhit()==true){  
    decyzja=getch();  
}
```

Uwaga ogólna do użytkowania serwomechanizmów

Zalecane jest jednorazowe włączanie servo i po wielokrotnym użyciu wyłączenie na końcu.

Porty serwomechanizmów

-OUT_A
-OUT_B
-OUT_C

motory jezdne podłączamy najczęściej do portów B,C, a dodatkowy servo do portu A.

Skręcanie w lewo przy $x < y$, skręcanie w prawo przy $x > y$, przy założeniu, że servomotor lewego koła jest podłączony do portu OUT_B, prawego koła do OUT_C

```
NXT::Motor::SetForward(&comm, OUT_B, x);  
NXT::Motor::SetForward(&comm, OUT_C, y);
```

Alternatywnie skręcanie w miejscu,

```
NXT::Motor::SetForward(&comm, OUT_B, x);  
NXT::Motor::SetBackward(&comm, OUT_C, x);
```

Jazda na wprost z prędkością x

```
NXT::Motor::SetForward(&comm, OUT_B, x);  
NXT::Motor::SetForward(&comm, OUT_C, x);
```

Jazda do tyłu z prędkością x

```
NXT::Motor::SetReverse(&comm, OUT_B, x);
NXT::Motor::SetReverse(&comm, OUT_C, x);
```

Przykładowe sterowanie szczypcami w modelu NXT Tribot, przy założeniu, że szczypce podłączone są do portu OUT_A

```
if (decyzja=='C' || decyzja=='c'){
    if (pointer==0){
        NXT::Motor::SetForward(&comm, OUT_A, 10);
        Wait(1000);
        NXT::Motor::Stop(&comm, OUT_A, 0);
        pointer=1;
        continue;
    }
    if (pointer==1){
        NXT::Motor::SetForward(&comm, OUT_A, -20);
        Wait(1000);
        pointer=0;
        continue;
    }
}
```

Przykładowe zwiększenie prędkości robota przy jeździe na wprost

```
if (decyzja=='M' || decyzja=='m'){
    x+=10;
}

if (decyzja=='W' || decyzja=='w' || (xtemp!=x && last_value=='w')){
    xtemp=x;
    last_value='w';
    NXT::Motor::SetForward(&comm, OUT_B, x);
    NXT::Motor::SetForward(&comm, OUT_C, x);
}
```

Zwalnianie serwomechanizmów

```
NXT::Motor::Stop(&comm, OUT_B, 0);
NXT::Motor::Stop(&comm, OUT_C, 0);
```

Uwaga ogólna do użytkowania sensorów

Zalecane jest jednorazowe włączanie sensorów i po wielokrotnym użyciu wyłączenie na końcu, w ten sposób czujniki będą działały stabilnie.

Porty sensorów

-IN_1,
-IN_2,
-IN_3,
-IN_4.

Przykładowe uaktywnienie czujnika dotyku w porcie 1 i demonstracja użycia

```
NXT::Sensor::SetTouch(&comm, IN_1);
```

Aby sprawdzić czy sensor dotyku został użyty, należy nasłuchiwać w pętli nieskończonej używając następującego kodu,

```
if (NXT::Sensor::GetValue(&comm, IN_3)==1){  
    NXT::Motor::Stop(&comm, OUT_B, 0);  
    NXT::Motor::Stop(&comm, OUT_C, 0);  
    std::cout<< "\n\nWykryta przeszkoda - silniki zatrzymane";  
}
```

Uaktywnienie sonaru w porcie 4, pobranie odległości od przeszkody i wyłączenie czujnika

```
NXT::Sensor::SetSonar(&comm, IN_4);  
NXT::Sensor::GetSonarValue(&comm, IN_4);  
NXT::Sensor::SetSonarOff(&comm, IN_4);
```

Obsługa czujnika KOLORU: uaktywnienie czujnika w porcie 2 w trybie odczytu intensywności światła 'n', pobranie poziomu światła, dezaktywacja czujnika

```
NXT::Sensor::SetColor(&comm, IN_2, 'n');  
Wait(50);  
color=NXT::Sensor::GetValue(&comm, IN_2);  
Wait(50);  
NXT::Sensor::SetColorOff(&comm, IN_2);
```

Inne funkcje czujnika koloru opisane są w instrukcji dostępnej pod linkiem <http://wmii.uwm.edu.pl/~artem/HowToUseColorSensor.pdf>

Obsługa czujnika ŚWIATŁA: Uaktywnienie czujnika w porcie 3, pobranie poziomu światła i dezaktywacja czujnika

```
NXT::Sensor::SetLight(&comm, IN_3, 1);  
Wait(50);  
value=NXT::Sensor::GetValue(&comm, IN_3);  
Wait(50);  
NXT::Sensor::SetLight(&comm, IN_3, 0);
```

Użycie czujnika dźwięku

```
NXT::Sensor::SetSound(&comm, IN_2);  
Wait(50);  
value=NXT::Sensor::GetValue(&comm, IN_2);
```