

RWTH - Mindstorms NXT Toolbox for MATLAB

1. Увод

Овај додатни пакет за рад у *MATLAB* окружењу у циљу остваривања комуникације између робота и рачунара развијен је на Универзитету у Ахену (CP Немачка) у оквиру пројекта под називом *Matlab meets LEGO Mindstorms*. У развоју овог софтверског пакета учествовало је више од 300 студената основних академских студија (BSc) и више од 60 наставника и сарадника.

Званична интернет презентација пројекта је: <http://www.lfb.rwth-aachen.de/mindstorms> или www.mindstorms.rwth-aachen.de/.

Основни вид комуникације робота са рачунаром је преко *wireless Bluetooth* протокола или *USB* кабла.

2. Инсталирање софтвера

Пре инсталирања софтверског пакета неопходно је инсталирати оригинални *LEGO NXT* апликативни софтвер. Након тога потребно је инсталирати додаток основном софтверу *LEGO® MINDSTORMS® NXT Firmware V1.05* (тзв. *patch*). Инсталацију је потребно спровести на следећи начин:

1. Отпаковати датотеку *Firmware V1.05* (пратити следећу путању ...\\Program Files\\LEGO Software\\LEGO MINDSTORMS NXT\\engine\\Firmware);
2. Стартовати *LEGO NXT* апликативни софтвер;
3. Притиснути "Tool" падајући мени и изабрати "Update NXT Firmware...".
4. *LEGO MINDSTORMS NXT Firmware v1.05* треба да се појави као "Available Firmware Files". Селектовати одабрани фајл и притиснути "Download".
5. Крај инсталације.

Након успешног инсталирања додатка основном софтверу *LEGO® MINDSTORMS® NXT* инсталација *MATLAB* апликативног софтвера је веома једноставна.

1. Отпаковати директоријум који садржи датотеке софтвера *RWTH - Mindstorms NXT Toolbox for MATLAB*;
2. Структура датотека (фајлова) у оквиру поменутог директоријума мора бити одржана!
3. Стартовати *MATLAB*;
4. Притиснути "File" и са падајућег менија изабрати "Set Path...". У оквиру новог прозора који ће се појавити одабрати "Add Folder...", и изабрати датотеку *RWTHMindstormsNXT*. Поновити овај корак и изабрати фолдер *demos*, који се налази у оквиру претходно додатог директоријума *RWTHMindstormsNXT*. Притиснути "Save".
5. Крај инсталације.

3. Рад у *MATLAB* окружењу са *RWTHMindstormsNXT*

Да би се стартовао програм потребно је иницијализирати конекцију између рачунара и робота путем УСБ кабла. У том смислу, треба спровести следећи код:

```
COM_CloseNXT('all')
clear all
```

ПЛ-1 .: Лаб. вежба #1 (*Matlab* апликативни софтвер и робот *LEGO Mindstorms NXT*)

```
close all
handle = COM_OpenNXT('USB.ini', 'check');
    COM_SetDefaultNXT(handle);
    OpenUltrasonic(SENSOR_4);
    distance = GetUltrasonic(SENSOR_4)
CloseSensor(SENSOR_4);
```

Првих пет командних линија је стандард и **увек** ћемо их користити за стартовање програма које будемо писали.

У овом примеру смо у последње три командне линије кода иницирали читавање сонара. У последњој линији смо извршили „затварање“ сензора. За неки други сензор, нпр. светлосни сензор, поступак је идентичан али с том разликом што би последња команда била:

```
OpenLight(SENSOR_3, 'ACTIVE');
    light = GetLight(SENSOR_3)
CloseSensor(SENSOR_3);
```

Обратити пажњу да увек можемо променити излазни порт за одговарајући сензор, што је веома битно

Управљање моторима

Роботски комплет има три серво мотора. Приступ мотору или моторима треба спровести на следећи начин:

```
COM_CloseNXT('all')
clear all
close all
% this function moves the robot untill the sonar reading is gretaer than 20
% it works fine
handle = COM_OpenNXT('USB.ini', 'check');
    COM_SetDefaultNXT(handle);
    % Motor Control
    SetMotor(MOTOR_B); % accesses the motor b
    SyncToMotor(MOTOR_C); % provides same controls to motor c
    SetPower 20 % power is 50 (0-100)
    SetTurnRatio 0 % straight forward
    SetAngleLimit 360 % how much to rotate motor shaft;
    %0 for endless rotation of shaft, it could be stopped with StopMotor
SendMotorSettings
% StopMotor all off
```

Дакле, прво успостављамо комуникацију између рачунара и робота. Као што се види на примеру, првих неколико команди је идентично за сваки појединачни програм који се развија у овом окружењу. Након тога позивамо функцију `SetMotor(MOTOR_B)` која „приступа“ моторима роботског комплета. `SyncToMotor(MOTOR_C)` прослеђује идентичне функције ка другом мотору. `SetPower 20` дефинише којом брзином ће се ротирати вратило мотора. Максимална вредност је 100. `SetTurnRatio 0` одређује да угао промене курса. На крају, команда `SetAngleLimit 360` одређује за који угао ће се ротирати излазно вратило мотора. Последња команда је `SendMotorSettings` и она прослеђује дефинисане вредности ка моторима. На крају, препорука је да се команда `StopMotor all off` **увек** држи у приправности!

Следећи пример се односи на симултани (истовремени) рад мотора и читавање сензора. „Очекујемо“ од робота да се креће све док сонар не региструје препреку на одређеној

ПЛ-1 .: Лаб. вежба #1 (*Matlab* апликативни софтвер и робот *LEGO Mindstorms NXT*)

удаљености. У примеру кода који следи, дистанца до препреке је 10 [cm]. у наставку је дат Матлаб код.

```
COM_CloseNXT('all')
clear all
close all
% this function moves the robot untill the sonar reading is gretaer than 10
% it works fine
handle = COM_OpenNXT('USB.ini', 'check');
COM_SetDefaultNXT(handle);
OpenUltrasonic(SENSOR_4);
distance = GetUltrasonic(SENSOR_4)
while distance > 10
distance = GetUltrasonic(SENSOR_4)
% Motor Control
SetMotor(MOTOR_B); % accesses the motor b
SyncToMotor(MOTOR_C); % provides same controls to motor c
SetPower 20 % power is 50 (0-100)
SetTurnRatio 0 % straight forward
SetAngleLimit 360 % how much to rotate motor shaft;
% 0 for endless rotation of shaft, it could be stopped with StopMotor
SendMotorSettings
end
StopMotor all off
CloseSensor(SENSOR_4);
```