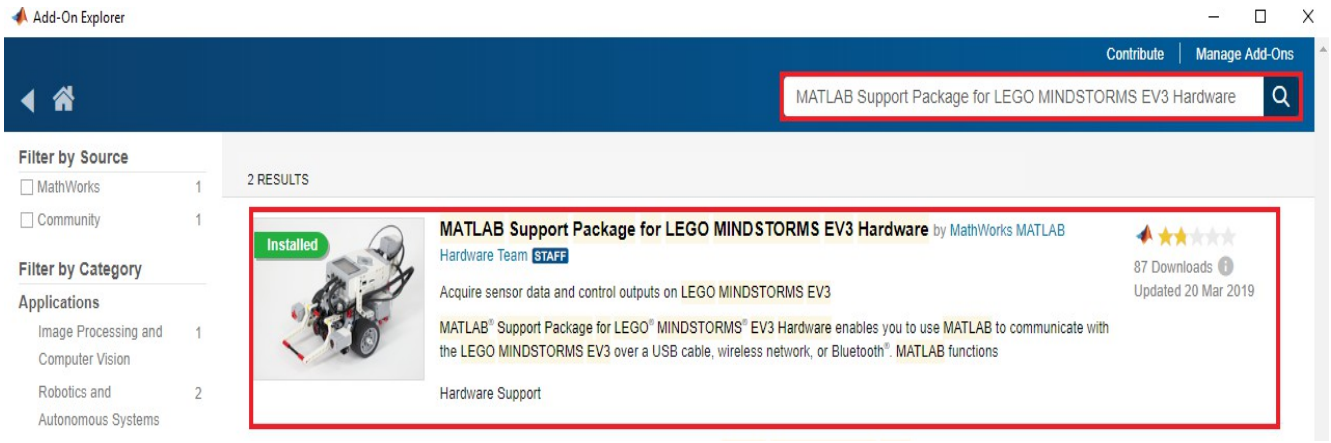
Robot Learning With LEGO

Einführung in:
LEGO MINDSTORMS EV3 Support from MATLAB

Installation



Initialisierung und Verbindung

```
%% USB Connection
ev3_usb = legoev3('usb');
% Do stuff
clear ev3_usb % after finishing the program the variable has to be
              % cleared in order free the connection for later usages
```

```
%% Bluetooh Connection
ev3_bt = legoev3('bluetooth', 'COM10');
%                             Serial Port name
% Do stuff
clear ev3_bt  % after finishing the program the variable has to be
              % cleared in order free the connection for later usages
```

```
%% WiFi Connection
ev3_wifi = legoev3('WiFi','192.168.1.2','00165340e49b');
%                          IP adress   ID of ev3 brick
% Do stuff
clear ev3_wifi  % after finishing the program the variable has to be
                % cleared in order free the connection for later usages
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/legoev3.html

Motor

```
ev3 = legoev3();
```

```
%% Init motors
motorA = motor(ev3, 'A'); % possible ports ('A', 'B', 'C', 'D')
```

```
%% set speed
motorA.Speed = 50; % forward at 50%
```

```
%% start motor
start(motorA);
pause(2); % wait to seconds
```

```
%% read and reset rotation of motor (in degree)
rot1 = readRotation(motorA);
resetRotation(motorA);
```

```
%% change speed and direction
motorA.Speed = -25; % backwards at 35%
pause(2);
```

```
%% Stop motor
stop(motorA);

clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/motor.html

Ultrasonic Sensor

```
%%
ev3 = legoev3();
```

```
%% Init ultra sonic sensor
sonic_sensor = sonicSensor(ev3);  %take ultra sonic sensor at port with lowest ID
sonic_sensor = sonicSensor(ev3, 2); % possible ports: 1, 2, 3, 4
```

```
%% read sensor value (in meters)
% range: 0m to 2.55m
distance = readDistance(sonic_sensor);
```

```
%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/sonicsensor.html

Touch Sensor

```
%%
ev3 = legoev3();
```

```
%% Init touch sensor
touch_sensor = touchSensor(ev3);  %take touch sensor at port with lowest ID
touch_sensor = touchSensor(ev3, 2); % possible ports: 1, 2, 3, 4
```

```
%% read sensor value
is_pressed = readTouch(touch_sensor);
% is_pressed == 0 -> sensor is not pressed
% is_pressed == 1 -> sensor is pressed by an object
```

```
%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/touchsensor.html

Gyro Sensor

```
%%
ev3 = legoev3();
```

```
%% Init gyro sensor
gyro_sensor = gyroSensor(ev3);  %take gyro sensor at port with lowest ID
gyro_sensor = gyroSensor(ev3, 2); % possible ports: 1, 2, 3, 4
```

```
%% read the total amound of rotation (in degrees)
alpha = readRotationAngle(gyro_sensor);
```

```
%% reset rotation angle measurement
resetRotationAngle(gyro_sensor);
```

```
%% read the rate of rotation of the sensor (in degrees per second)
d_alpha = readRotationRate(gyro_sensor);
```

```
%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/gyrosensor.html

Color Sensor

```
ev3 = legoev3();

%% Init color sensor
color_sensor = colorSensor(ev3);  %take color sensor at port with lowest ID
color_sensor = colorSensor(ev3, 2); % possible ports: 1, 2, 3, 4

%% read color values
color = readColor(color_sensor);
% return values:
% none, black, blue, green, yellow, red, white, brown

%% read ambient light intensity
% values form 0 to 100
amb_light = readLightIntensity(color_sensor);
%or
amb_light = readLightIntensity(color_sensor, 'ambient');

%% read reflected light intensity
% values from 0 to 100
refl_light = readLightIntensity(color_sensor, 'reflected');

%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/colorsensor.html

LCD Display

```
%%
ev3 = legoev3();
```

```
%% LCD
% clear LCD
clearLCD(ev3);

% write at LCD near the center
writeLCD(ev3, "test message");

% write at LCD in the first row starting at column 2
writeLCD(ev3, "test message", 1, 2);
% number of rows: 9
% number of clumns: 19
```

```
%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/writelcd.html
https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/clearlcd.html

Sound

```
%%
ev3 = legoev3();
```

```
%% Play tones
% play tone at 500Hz for one second at volume 10(default values)
playTone(ev3);
% play tone at 450Hz, volume 50 for 2.5 second
playTone(ev3, 450, 2.25, 50);
% freq: 250 to 10000 Hz
% duration: 0.0 to 30.0
% volume: 0 to 100
```

```
%% Beeb
% play a beep for one-tenth of a second at 500 Hz
beep(ev3);
%play a beep for 1 second
beep(ev3, 1)
% duration: 0 to 30 seconds
```

```
%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/playtone.html
https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/beep.html

Buttons and Status Lights

```
%%
ev3 = legoev3();

%% Buttons
% read the status of the center button
btn_center = readButton(ev3, 'center');
% btn_center == 1 -> button is pressed
% btn_center == 0 -> button is not pressed

% button posisionts:
% up, down, left, right, center

%% Status lights
% set status lights to green in pulsing mode
writeStatusLight(ev3, 'green', 'pulsing');
% colors: off (default), green, red, orange
% modes: solid (default), pulsing

% set status light to off
writeStatusLight(ev3, 'off');

%%
clear ev3
```

Weitere Informationen: https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/readbutton.html
https://de.mathworks.com/help/supportpkg/legomindstormsev3io/ref/beep.html