nxt_movement.pl -- NXT Mindstroms - simple movement.

author
- Piotr Hołownia

license
- GNU General Public License

nxt_set_robot(*+WheelCircumference, +AxleLenght, +LeftMotor, +RightMotor, +Reverse, +TouchPort, +SoundPort, +LightPort, +UltrasonicPort*)
Changes the robot's settings. !! *Reverse* not implemented!

nxt_stop
Stops the robot.

nxt_is_stopped
Returns true if robot is stopped. Otherwise fails.

nxt_go(*+Speed*)
Moves the robot forward (if *Speed* is greater than 0) or backward (if *Speed* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels stop when nxt_stop predicate is called. Starts when the robot is stopped.

nxt_go(*+Speed, +Option*)
The same as nxt_go(+*Speed*), except it works immediately. *Option* is:

force

nxt_go(*+Speed, +Angle*)
Moves the robot forward (if *Speed* is greater than 0) or backward (if *Speed* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels stop after revolution of *Angle* (in degrees). Starts when the robot is stopped.

nxt_go(*+Speed, +Angle, +Option*)
The same as nxt_go(+*Speed*,+*Angle*), except it works immediately. *Option* is:

force

nxt_go_sec(*+Speed, +Time*)
Moves the robot forward (if *Speed* is greater than 0) or backward (if *Speed* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels stop after specified time (in seconds). Starts when the robot is stopped.

nxt_go_sec(*+Speed, +Time, +Option*)
The same as nxt_go_sec(+*Speed*,+*Time*), except it works immediately. *Option* is:

force

nxt_go_cm(*+Speed, +Distance*)
Moves the robot forward (if *Speed* is greater than 0) or backward (if *Speed* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels will stop, if the *Distance* (in cm) is reached. Starts when the robot is stopped.

nxt_go_cm(*+Speed, +Distance, +Option*)
The same as nxt_go_cm(+*Speed*,+*Distance*), except it works immediately. *Option* is:

force

nxt_go_in(*+Speed, +Distance*)
Moves the robot forward (if *Speed* is greater than 0) or backward (if *Speed* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels will stop, if the *Distance* (in inches) is reached. Starts when the robot is stopped.

nxt_go_in(*+Speed, +Distance, +Option*)
The same as nxt_go_in(+*Speed*,+*Distance*), except it works immediately. *Option* is:

force

nxt_go_cm_sec(*+Distance, +Time*)
Moves the robot forward (if *Distance* is greater than 0) or backward (if *Distance* is smaller than 0). Robot reaches the *Distance* (in cm) in *Time* (in seconds). Starts when the robot is stopped.

nxt_go_cm_sec(*+Distance, +Time, +Option*)
The same as nxt_go_cm_sec(+*Distance*,+*Time*), except it works immediately. *Option* is:

force

nxt_go_in_sec(*+Distance, +Time*)
Moves the robot forward (if *Distance* is greater than 0) or backward (if *Distance* is smaller than 0). Robot reaches the *Distance* (in inches) in *Time* (in seconds). Starts when the robot is stopped.

nxt_go_in_sec(*+Distance, +Time, +Option*)
The same as nxt_go_in_sec(+*Distance*,+*Time*), except it works immediately. *Option* is:

force

nxt_turn_degrees(*+Speed, +Degrees*)
Rotates the robot in place to its left (if *Degrees* is greater than 0) or right (if *Degrees* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels will stop, when specified revolution (*Degrees*) of the robot is reached. Starts when the robot is stopped.

nxt_turn_degrees(*+Speed, +Degrees, +Option*)
The same as nxt_turn_degrees(+*Speed*,+*Degrees*), except it works immediately. *Option* is:

force

nxt_turn(*+Speed, +Angle*)
Rotates the robot in place to its left (if *Angle* is greater than 0) or right (if *Angle* is smaller than 0). *Speed* is the rotational speed of the wheel in degrees per second. Wheels stop after revolution of *Angle* (in degrees). Starts when the robot is stopped.

nxt_turn*(+Speed, +Angle, +Option)*
The same as nxt_turn(+*Speed*,+*Angle*), except it works immediately. *Option* is:

force

nxt_turn*(+Radius, +Degrees, +Time)*
Makes robot turn with specified turning radius (*Radius*) moving forward (if *Degrees* is positive) or backward (if negative). Robot turns left (if *Radius* is positive) or right (if negative). Robot reaches the specified revolution (*Degrees*) in *Time* (in seconds). Starts when the robot is stopped.

nxt_turn*(Radius, Degrees, Time, +Option)*
The same as nxt_turn(+*Radius*,+*Degrees*,+*Time*), except it works immediately. *Option* is:

force

nxt_touch*(-Value)*
Gets touch sensor reading. Returns 1 if pressed, 0 otherwise. Starts when the robot is stopped.

nxt_touch*(-Value, +Option)*
The same as nxt_touch(-*Value*), except it works immediately. *Option* is:

force

nxt_sound*(-Value)*
Gets sound sensor reading. Starts when the robot is stopped.

nxt_sound*(-Value, +Option)*
The same as nxt_sound(-*Value*), except it works immediately. *Option* is:

force

nxt_light*(-Value)*
Gets light sensor reading. Starts when the robot is stopped.

nxt_light*(-Value, +Option)*
The same as nxt_light(-*Value*), except it works immediately. *Option* is:

force

nxt_light_LED*(+Setting)*
Sets the LED on if *Setting* is `activate` or off if `passivate`. Starts when the robot is stopped.

nxt_light_LED*(+Setting, +Option)*
The same as nxt_light_LED(+*Setting*), except it works immediately. *Option* is:

force

nxt_ultrasonic*(-Value)*
Gets ultrasonic sensor reading. Starts when the robot is stopped.

nxt_ultrasonic*(-Value, +Option)*
The same as nxt_ultrasonic(-*Value*), except it works immediately. *Option* is:

force