

1 **nxt_movement.pl** – NXP Mindstorms - simple movement.

author Piotr Hoownia

license GNU General Public License

nxt_set_robot(+*WheelCircumference*, +*AxleLenght*, +*LeftMotor*, +*RightMotor*, +*PincerMotor*, +*Reverse*, +*TouchPort*, +*SensorPort*)
Changes the robot's settings. !! *Reverse* not implemented! Set sensor port to 'S0' if there is no sensor.

nxt_open

Opens connection with default TTL (Time To Live) of 4, which means that in case of connection failure there will be 3 additional retries.

nxt_close

Closes connection with the robot and terminates all its threads, triggers and timers.

nxt_stop

Stops the robot by sending a stop command to all motors.

nxt_is_stopped

Returns true if robot is stopped. Otherwise fails.

nxt_motor(+*Motor*, +*Speed*)

nxt_motor(+*Motor*, -*Speed*)

Reads absolute value of speed of the specified motor or rotates it at specified speed. Starts when the robot is stopped. If *Speed* is 0, a robot will stop.

nxt_motor(+*Motor*, +*Speed*, +*Spec*)

Rotates motor forward if *Speed* is positive, backward if *Speed* is negative. Starts when the robot is stopped. Character of this rotation depends on *Spec*. *Spec* is one of:

time(+*Time*)

Rotates motor at specified speed. *Motor* will stop after specified time in seconds.

angle(+*Angle*)

Rotates motor at specified speed. *Motor* will stop, when specified revolution (Angle in degrees) is reached.

+ *Angle*

The same as **angle**(+*Angle*)

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 wheels in degrees per second. Maximal speed is 900 degrees per second (actual speed may vary depending on voltage). 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 wheels in degrees per second. Maximal speed is 900 degrees per second (actual speed may vary depending on voltage). 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 wheels in degrees per second. Maximal speed is 900 degrees per second (actual speed may vary depending on voltage). 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 wheels in degrees per second. Maximal speed is 900 degrees per second (actual speed may vary depending on voltage). 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_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_rotate(+Speed, +Degrees)

Rotates the robot in place to its right (if *Degrees* is greater than 0) or left (if *Degrees* is smaller than 0). *Speed* is the rotational speed of wheels in degrees per second. Maximal speed is 900 degrees per second (actual speed may vary depending on voltage). Wheels will stop, when specified revolution (*Degrees*) of the robot is reached. Starts when the robot is stopped.

nxt_rotate(+Speed, +Degrees, +Option)

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

force

nxt_turn(+Speed, +Angle)

Rotates the robot in place to its right (if *Angle* is greater than 0) or left (if *Angle* is smaller than 0). *Speed* is the rotational speed of wheels in degrees per second. Maximal speed is 900 degrees per second (actual speed may vary depending on voltage). 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(+Speed, +Degrees, +Radius)

Makes robot turn with specified turning radius (*Radius*) moving forward (if *Speed* is positive) or backward (if negative). Robot turns right (if *Degrees* is positive) or left (if negative). Robot reaches the specified revolution (*Degrees*) and stops. *Speed* is a speed of an outer wheel.

nxt_turn(+Speed, +Degrees, +Radius, +Option)

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

force

nxt_pincer(+Option)

Moves pincer when a robot is stopped. *Option* is one of:

open

close

nxt_pincer(+Option1, +Option2)

Moves pincer immediately. *Option* is one of:

open

close

Option2 is:

force

nxt_pincer(+Option, +Speed)

Moves pincer when a robot is stopped with specified rotational motor speed. *Option* is one of:

open

close

nxt_pincer(+Option1, +Speed, +Option2)

Moves pincer immediately with specified rotational motor speed. *Option1* is one of:

open

close

Option2 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 (0-100). 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 (0-100, 0 meaning complete darkness). 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 (0-255). Starts when the robot is stopped.

nxt_ultrasonic(-Value, +Option)

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

force