imports

Chemical::*
Location::*

MicroControllerOPS

O changeDirection(l: Loc)

🖧 Movement

Ⓡ Operations
Ⓡ MicroControllerOPS
π lv: real, evadeTime: nat, stuckPeriod: nat, stuckDist: real, outPeriod: nat
✗ a: Angle, d0: real, d1: real, l: Loc
🕐 T

obstacle: Loc
odometer: real
resume
turn: Angle
stop
flag

F

Found

entry move(0, Angle::Front);
flag

stop
stop
stop
stop

resume

resume

GettingOut

entry shortRandomWalk();
wait (outPeriod)

[since(T)>=stuckPeriod∧d1-d0<=stuckDist]

AvoidingAgain

stop

#T[since(T)<stuckPeriod∨d1-d0>stuckDist]

obstacle?l/odometer?d1

ⓘ

turn?a

resume

Waiting

during randomWalk()

turn?a

Going

entry move(lv, a)

obstacle?l #T

Avoiding

entry odometer?d0;
changeDirection(l);
wait (evadeTime)

turn?a

TryingAgain

entry move(lv, a)

stop

turn?a

resume

turn?a

resume

resume