Parkside Montessori

LEGO®

May 9, 2024

Review

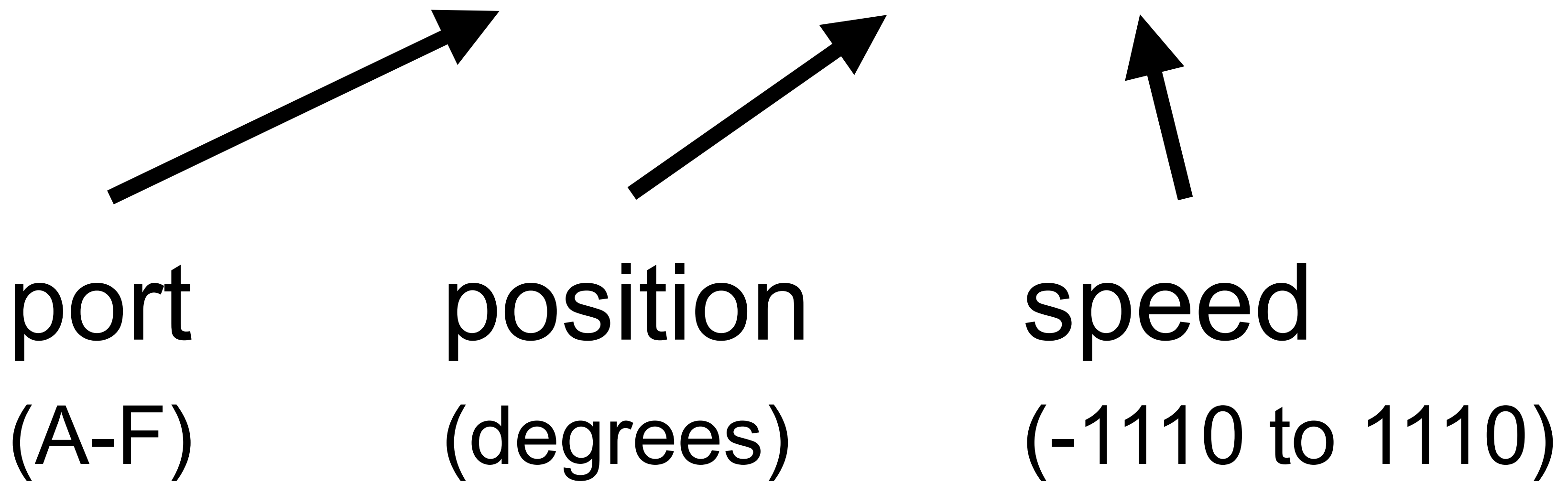
```python
from hub import port
import runloop
import motor

async def raise_arm():
    await motor.run_to_absolute_position(port.A, 45, 300)
    print("Arm raised")

async def main():
    # Why do we need to use "await" here?
   await runloop.sleep_ms(1000)
   await raise_arm()

runloop.run(main())
```

Review

Run to Absolute Position

- In the *Help* panel:
  API Modules > Motor > run_to_absolute_position

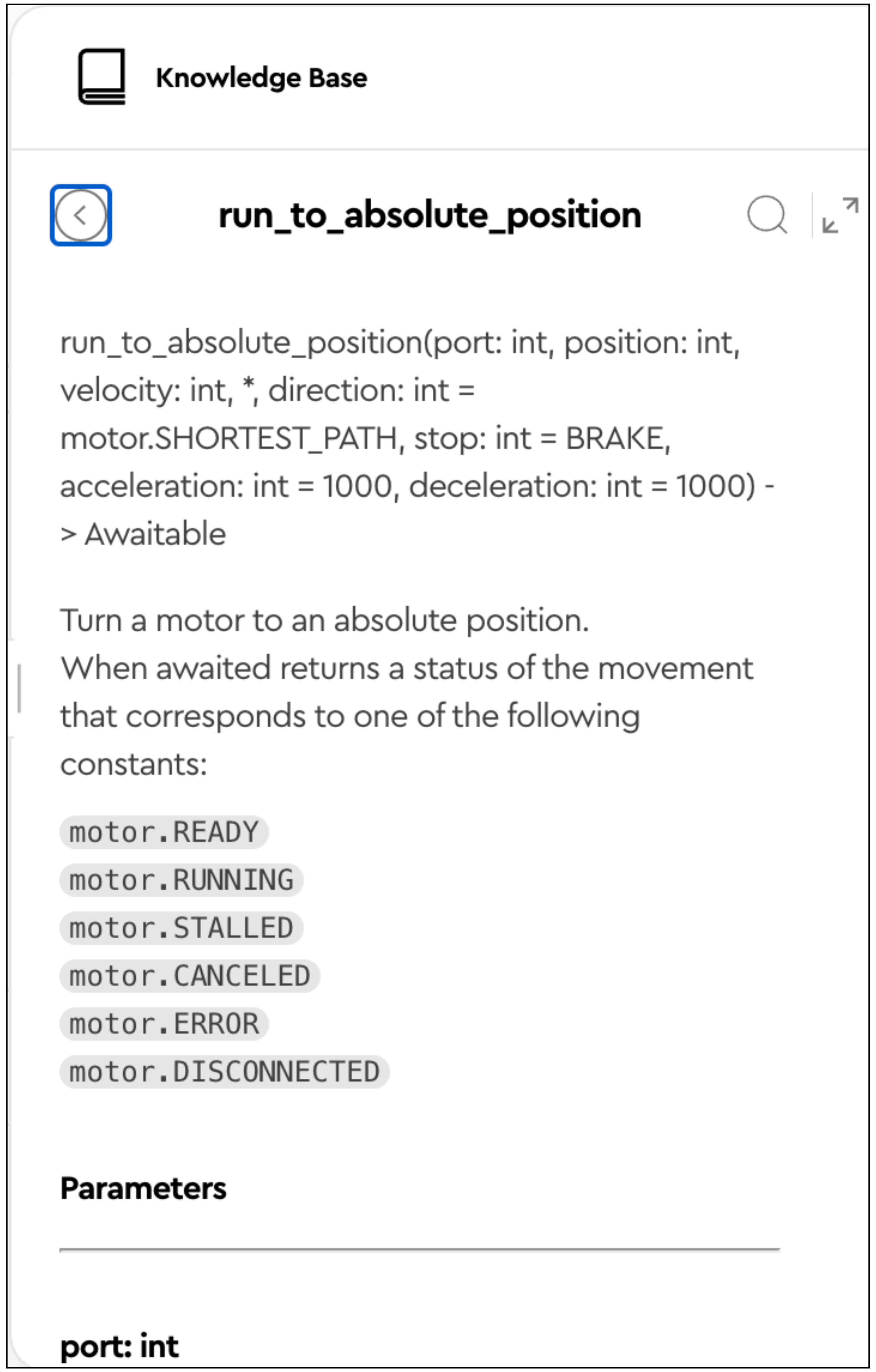
```
await motor.run_to_absolute_position(port.A, 45, 300)
```



port
(A-F)

position
(degrees)

speed
(-1110 to 1110)



run_to_absolute_position

run_to_absolute_position(port: int, position: int, velocity: int, *, direction: int = motor.SHORTEST_PATH, stop: int = BRAKE, acceleration: int = 1000, deceleration: int = 1000) -> Awaitable

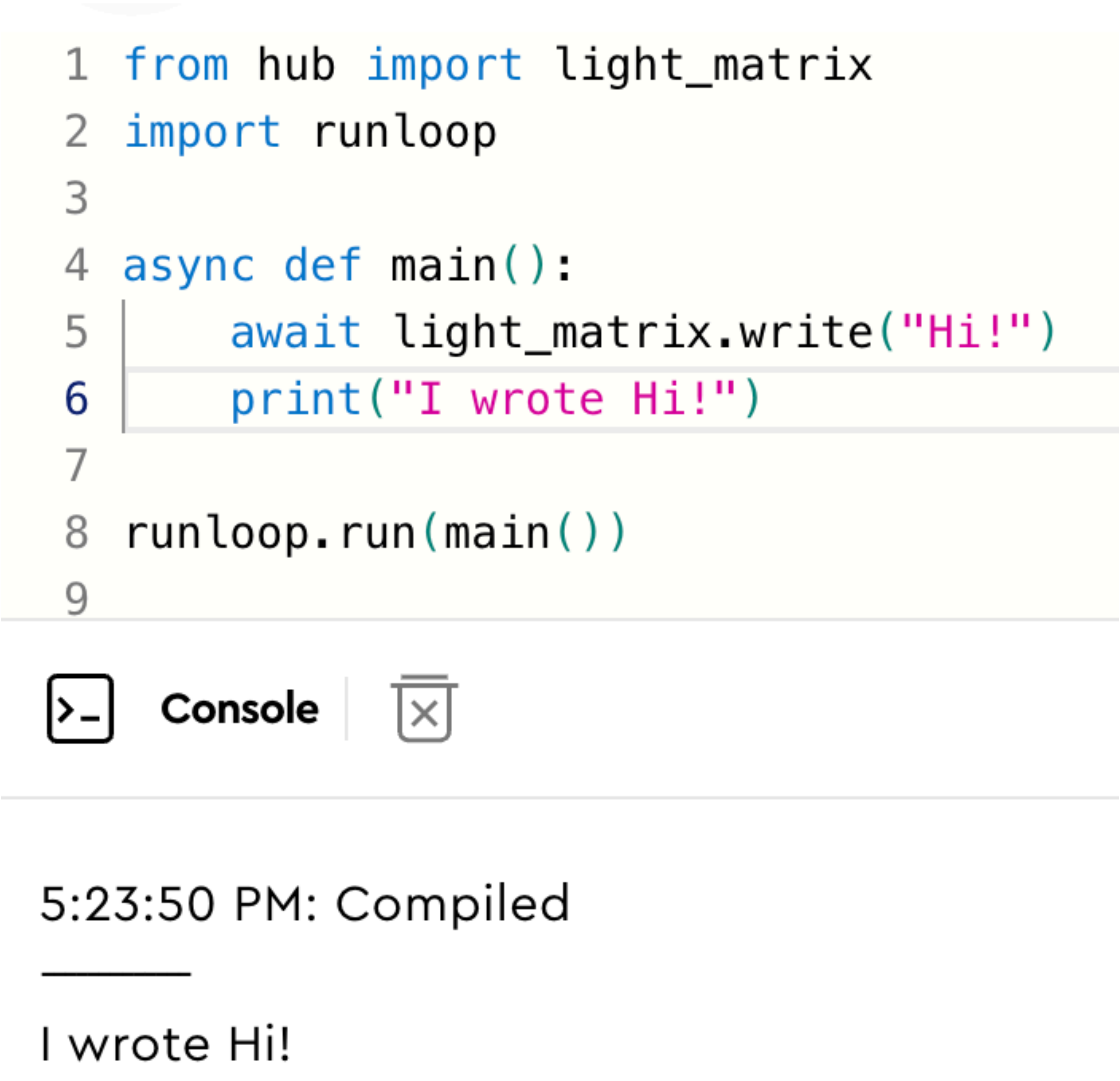
Turn a motor to an absolute position.
When awaited returns a status of the movement that corresponds to one of the following constants:

`motor.READY`
`motor.RUNNING`
`motor.STALLED`
`motor.CANCELED`
`motor.ERROR`
`motor.DISCONNECTED`

Parameters

port: int

Review



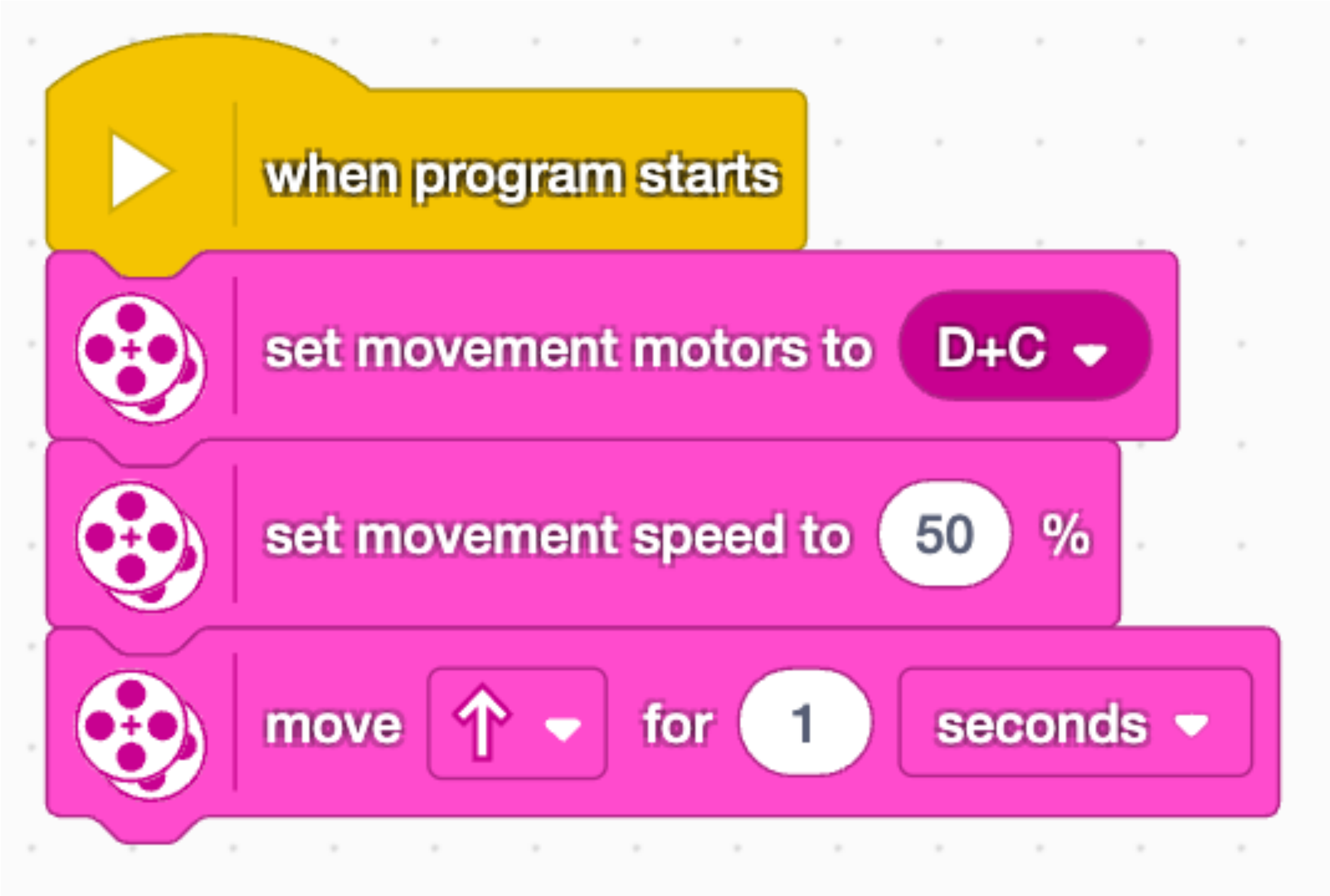
```python
from hub import light_matrix
import runloop

async def main():
    await light_matrix.write("Hi!")
    print("I wrote Hi!")

runloop.run(main())

```

```
5:23:50 PM: Compiled
________
I wrote Hi!
```

print()

- Use *print()* to send text from the LEGO controller back to your computer

Movement



when program starts

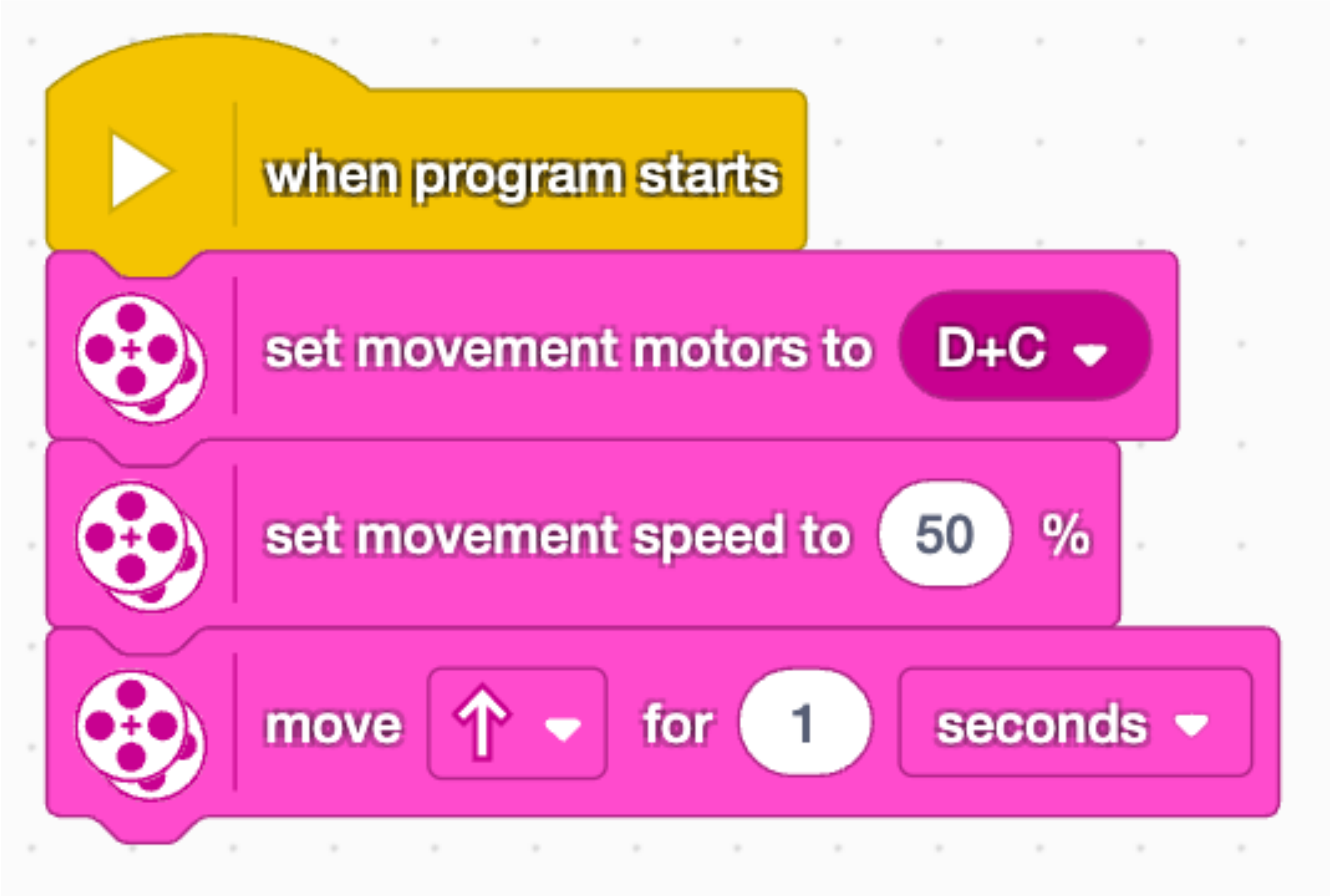
set movement motors to D+C

set movement speed to 50 %

move ↑ for 1 seconds

```
import motor_pair
import runloop

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 1000, 550, 550)

runloop.run(main())
```

Movement



when program starts

set movement motors to D+C

set movement speed to 50 %

move ↑ for 1 seconds

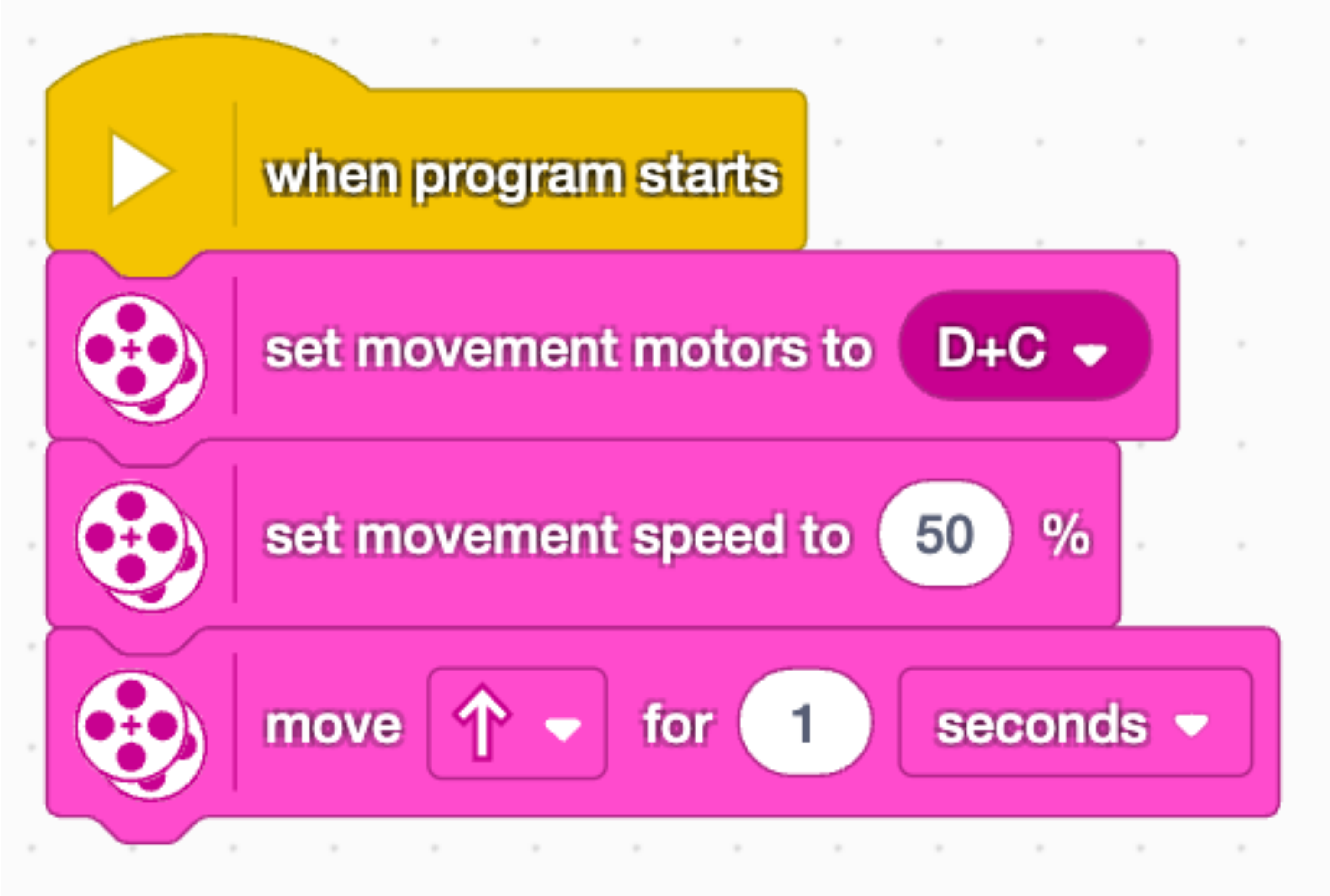
```
import motor_pair
import runloop

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 1000, 550, 550)

runloop.run(main())
```

motor pair name

left motor

right motor

Movement



when program starts

set movement motors to D+C

set movement speed to 50 %

move ↑ for 1 seconds

```
import motor_pair
import runloop

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 1000, 550, 550)

runloop.run(main())
```

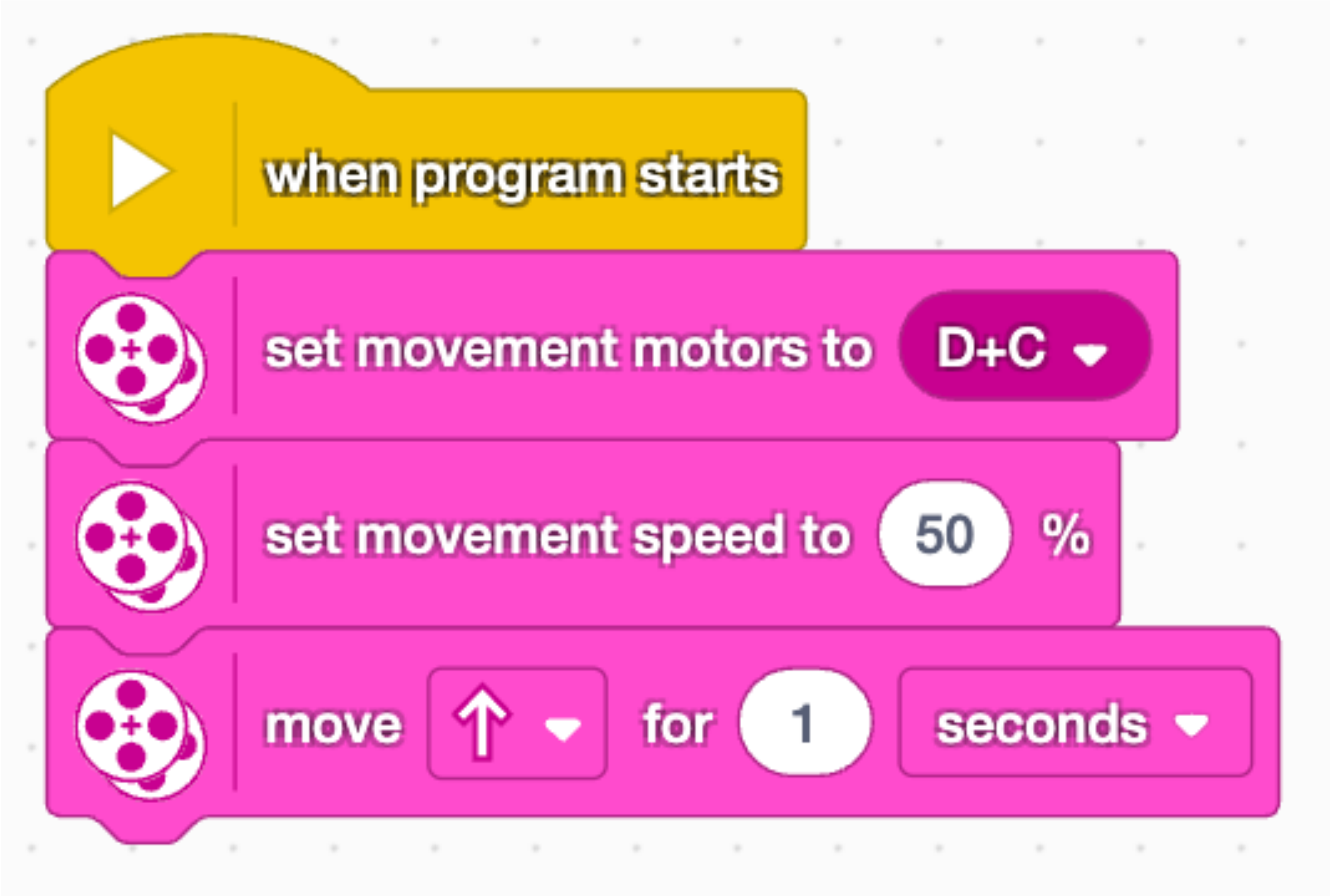
time in milliseconds
(1 sec = 1000 ms)

left motor speed
(-1100 to 1100)

motor pair name

right motor speed
(-1100 to 1100)

What Does This Do?



when program starts

set movement motors to D+C

set movement speed to 50 %

move ↑ for 1 seconds

```
import motor_pair
import runloop

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 1000, -550, 550)

runloop.run(main())
```

right motor speed

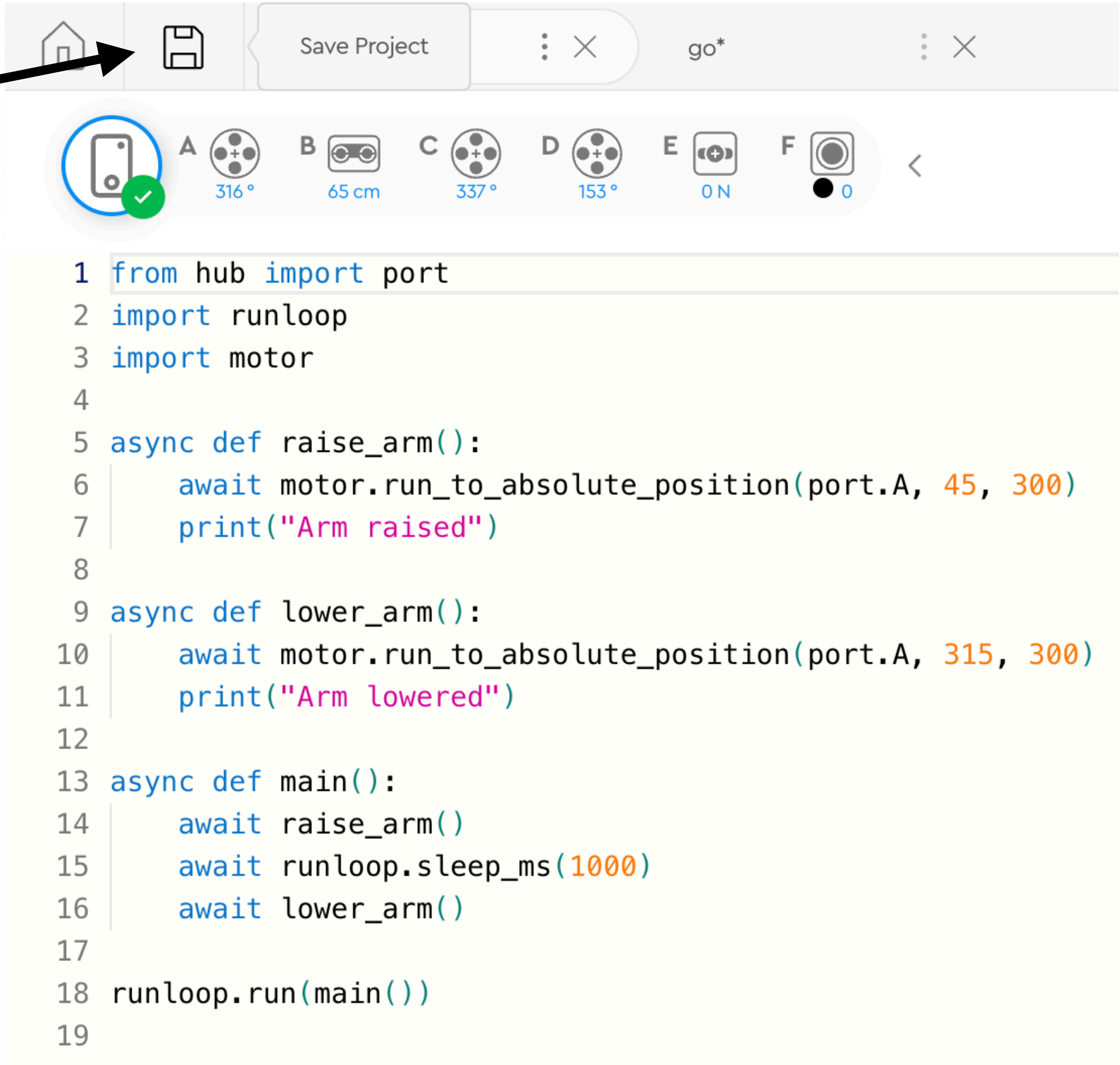
left motor speed

More Movement

```
# Tank motion by time
# move_tank_for_time(<pair>, <milliseconds>, <left motor velocity>, <right motor velocity>)
await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 5000, 550, 550)

# Tank motion by rotations
# move_tank_for_degrees(<pair>, <degrees>, <left motor velocity>, <right motor velocity>)
await motor_pair.move_tank_for_degrees(motor_pair.PAIR_1, 360*10, 550, 550)

# Arc motion by time
# move_for_time(<pair>, <milliseconds>, <steering -100 to 100>)
await motor_pair.move_for_time(motor_pair.PAIR_1, 5000, 0)

# Arc motion by rotations
# move_for_degrees(<pair>, <degrees>, <steering -100 to 100>)
await motor_pair.move_for_degrees(motor_pair.PAIR_1, 360*10, 0)
```

Save Your Program

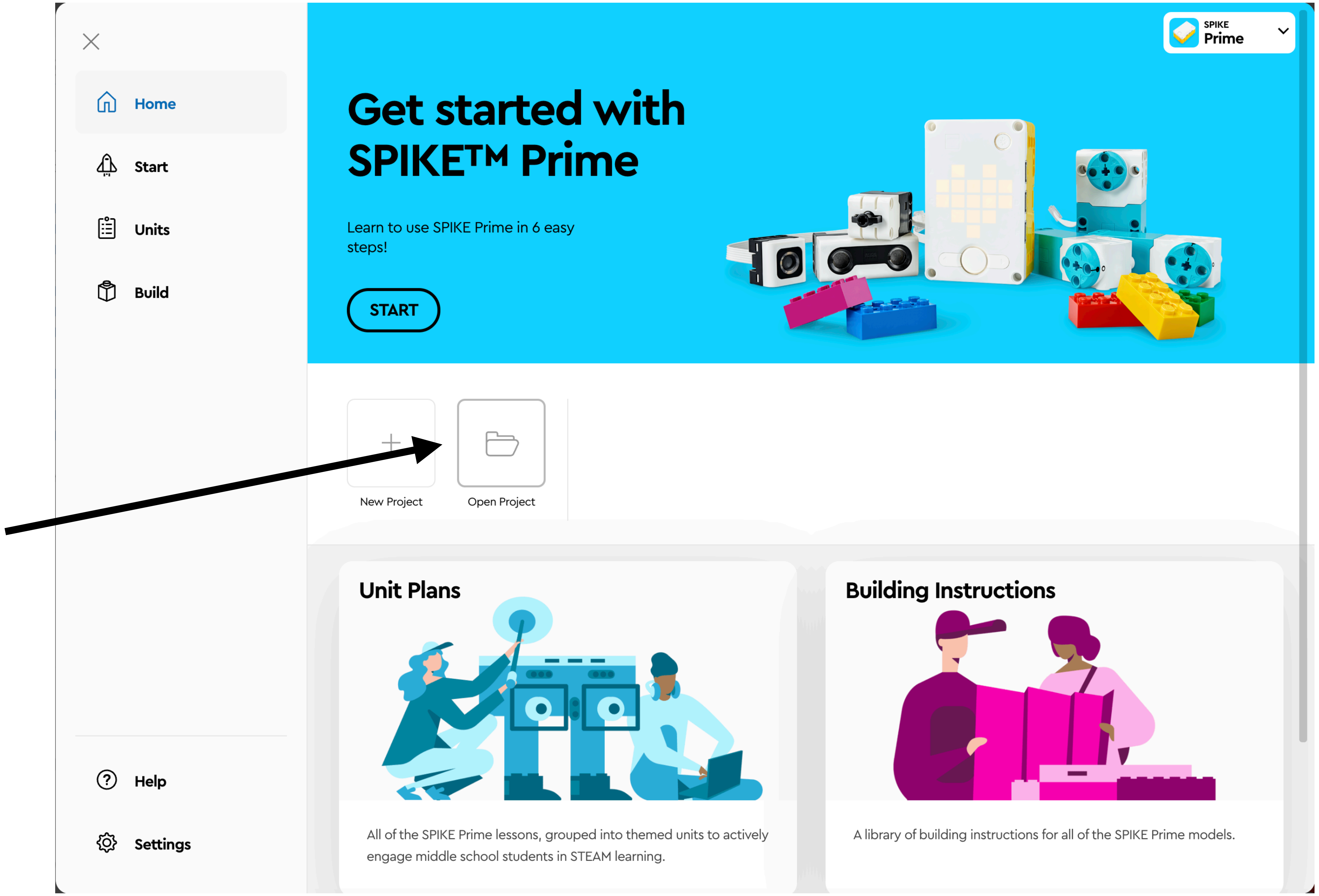
Save your program



```
from hub import port
import runloop
import motor

async def raise_arm():
    await motor.run_to_absolute_position(port.A, 45, 300)
    print("Arm raised")

async def lower_arm():
    await motor.run_to_absolute_position(port.A, 315, 300)
    print("Arm lowered")

async def main():
    await raise_arm()
    await runloop.sleep_ms(1000)
    await lower_arm()

runloop.run(main())

```

Open Saved Program

Open saved program



Get started with SPIKE™ Prime

Learn to use SPIKE Prime in 6 easy steps!

Unit Plans

All of the SPIKE Prime lessons, grouped into themed units to actively engage middle school students in STEAM learning.

Building Instructions

A library of building instructions for all of the SPIKE Prime models.