



5.3 Timers

Name _____

Class _____

Date _____

What this lesson is about

Two things at once, remembered ticks, timeouts.

- Two things at once
- A timer is a remembered tick
- Timeouts

Questions

7 marks in all

1. Why can a program not blink the LED with `wait(0.5)` while it drives to the zone? [1 mark]

- ☐ A `wait(0.5)` holds up the whole loop, so the driving stops being updated
- ☐ B `wait` turns the LED off
- ☐ C The LED cannot change while the robot moves
- ☐ D `wait` only accepts whole numbers

2. What does this program print? [1 mark]

```
on = False
changes = 0
for ticks in range(20):
    if ticks % 5 == 0:
        on = not on
        changes += 1
print(changes, on)
```

3. The loop waits 0.1 s a tick. To toggle the LED every half second, every how many ticks should it toggle? [1 mark]

4. What does this program print? [1 mark]

```
started = None
for ticks in range(40):
    if started is None and ticks == 20:
        started = ticks
        print("start", ticks)
    elif started is not None and ticks - started >= 10:
        print("stop", ticks)
        started = None
```

5. In `started_turning = None`, what does `None` mean? [1 mark]
- ☐ A No timer is running
 - ☐ B The timer has run for 0 ticks
 - ☐ C The robot has stopped
 - ☐ D The turn failed
6. A search spins while `ticks < 30` and gives up if nothing is found. Why is this timeout important? [1 mark]
- ☐ A Without it, a search for a tag that is not there would spin forever
 - ☐ B The camera stops working after 3 seconds
 - ☐ C It makes the tag easier to see
 - ☐ D It saves the list of tags
7. What is the advantage of `clock() - started_at >= 1.0` over counting ticks? [1 mark]
- ☐ A It stays right even if a tick takes longer than planned
 - ☐ B It works without a loop
 - ☐ C It makes each tick shorter
 - ☐ D It stops the robot automatically

The task: blink and drive

Drive into the green zone while the LED blinks green and off every half second, at least eight changes.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

# the LED: green
led("green")
# drive forward at 60 for 70 cm, then stop
forward(60, distance=70)
# the LED: off
led("off")
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/5-3-timers/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Blink faster as the robot gets closer to the zone.
2. Drive for two seconds, stop for one, repeat, using one timer variable.
3. Print the total run time in seconds at the end from the tick count. Check it against the telemetry.