



2.2 The distance sensor

What this lesson is about

distance() and loops that react to it.

- distance()
- A loop that reacts
- Slowing down as you get close
- Following the wall's distance

Questions

7 marks in all

1. How does the time-of-flight sensor measure distance? [1 mark]

- ☐ A It fires invisible light and times how long the reflection takes to come back
- ☐ B It counts how far the robot has driven
- ☐ C It listens for an echo of a sound
- ☐ D It looks at how big things appear in the camera

Answer: A. Light travels at a known speed, so the time for it to bounce back tells you how far away the thing is.

2. distance() reads 70 cm. The robot runs forward(60, distance=20) towards the wall. About what does distance() read now? Give a whole number. [1 mark]

Answer: 50 (accept within 2). Driving 20 cm towards the wall takes about 20 off the reading.

3. This loop stops the robot. Where does it end up? [1 mark]

```
while distance() > 30:  
    forward(40)  
    wait(0.1)  
stop()
```

- ☐ A A little closer than 30 cm, because it coasts after stop()
- ☐ B Exactly 30 cm from the wall
- ☐ C A little further than 30 cm
- ☐ D Touching the wall

Answer: A. The loop ends once the reading is 30 or less, and then the robot still slides for a moment after the motors switch off.

4. Why is there a `wait(0.1)` inside the loop? Choose all that apply.

[1 mark]

Tick every answer that is true.

- ☐ A It gives the robot time to move before the next look
- ☐ B It keeps the safety timer fed
- ☐ C It makes the sensor read in millimetres
- ☐ D It makes the robot drive at a lower speed

Answer: A, B. Without it the loop would ask the sensor again before the robot had moved, and no time would pass at all. The wait also counts as the program still talking to the robot, which keeps the safety timer fed.

5. In a "drive until the wall is close" loop, how does the program know how far the wall is?

[1 mark]

- ☐ A It does not; it asks the sensor before every repeat
- ☐ B You type the distance in at the top
- ☐ C It works it out from the speed
- ☐ D It counts how many loops have run

Answer: A. Nothing in the program knows where the wall is. The robot finds out by looking, and that is what makes it a robot.

6. What does this program print?

[1 mark]

```
for d in [80, 45, 30]:
    if d > 40:
        print(d, "fast")
    else:
        print(d, "slow")
```

Answer:

```
80 fast
45 fast
30 slow
```

Only 80 and 45 are more than 40. The same test in a driving loop means fast while far, slow when close.

7. What does this program print?

[1 mark]

```
readings = [60, 52, 44]
for i in range(3):
    print(f"step {i}: wall in {readings[i]} cm")
```

Answer:

```
step 0: wall in 60 cm
step 1: wall in 52 cm
step 2: wall in 44 cm
```

`i` counts 0, 1, 2, and `readings[i]` picks the reading at that position, so each step shows its own number.

The task: stop before the wall

Drive up to the wall and stop in the green band in front of it, without touching it. Adjust the number and the speed until the robot stops inside the band every time.

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

# as long as the thing ahead is further than 30 cm
while distance() > 30:
    # drive forward at 50 (keeps going until the next command)
    forward(50)
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
```

The hint students can ask for: Drive until distance() says the wall is close, then stop. The green zone is the band just before the wall.

A solution

```
from bugbot import *
connect()
while distance() > 20:
    forward(50)
    wait(0.1)
stop()
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.