



6.8 Project: the delivery

Seeing more · Robot club · about 30 min

Name _____

Class _____

Date _____

What this lesson is about

Follow a line to a marker and back through traffic.

- The plan, drawn
- Two detectors, one camera
- Turning round
- The whole program
- Where next

Questions

6 marks in all

1. Why does the delivery program keep calling `set_cv` inside the loop? [1 mark]

- ☐ A The camera runs one detector at a time, and it needs both lines and tags
- ☐ B `set_cv` refreshes the picture
- ☐ C Each detector only works once
- ☐ D It makes the robot drive straighter

2. Put the delivery states in order. [1 mark]

Number the lines 1 to 4 to put them in the right order.

___ turn
___ done
___ home
___ out

3. The give way check comes after the state logic in each tick. Why there? [1 mark]

- ☐ A Its `stop()` overrides whatever the state asked for
- ☐ B It has to run before `set_cv("line")`
- ☐ C The state logic would undo the stop otherwise
- ☐ D Checks at the end of a loop run faster

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

```
def facing_back(h, h0):  
    return abs((h - h0 - 180 + 360) % 360 - 180) < 20  
  
for h in [0, 90, 170, 185, 200]:  
    print(h, facing_back(h, 0))
```

5. What is `h0` for?

[1 mark]

- ☐ **A** It remembers the heading when marker 7 was found, so the robot knows when it faces back
- ☐ **B** It is the heading at the start of the program
- ☐ **C** It stores the line's angle
- ☐ **D** It counts ticks in the turn state

6. A student reads the tags with `set_cv("apriltag")` but forgets to switch back to `set_cv("line")`. [1 mark]
What will `follow_step` do?

- ☐ **A** Return `False` every tick, because the line detector is off
- ☐ **B** Follow the line as normal
- ☐ **C** Follow the tags instead of the line
- ☐ **D** Raise an error on the first tick

The task: the delivery

Follow the line to marker 7, print `found 7`, and follow it home without touching the red robot.

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

def follow_step(speed=60, gain=0.4):
    # [cx, angle] of the line ahead, or [] if none
    seen = line()
    if not seen:
        return False
    cx, angle = seen
    # forward, sideways, rotation: -100 to 100 each, until the next command
    drive(speed, 0, (cx - 160) * gain)
    return True

# camera: the line detector
set_cv("line")
# again and again, for ever
while True:
    if not follow_step(55, 0.4):
        # leave the loop
        break
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
```

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



Do it on the robot

www.bugbotlab.com/learn/6-8-project-delivery/

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

Challenges

1. Make the give-way rule use the prediction from lesson 6.7 instead of stopping for every close robot.
2. Blink the LED green on the way out and blue on the way home.
3. Time the whole delivery and print it. Then make it faster on the straights.