Robot projects

MQTT topics

MQTT messages are arranged hierarchically. The hierarchy levels are called topics. The top-level topic must be Zumo followed with your group number. You can check your group numbers in course workspace in OMA. For example, if your group number is 5 then the top-level topic must be “Zumo5” for all data that you send to the broker

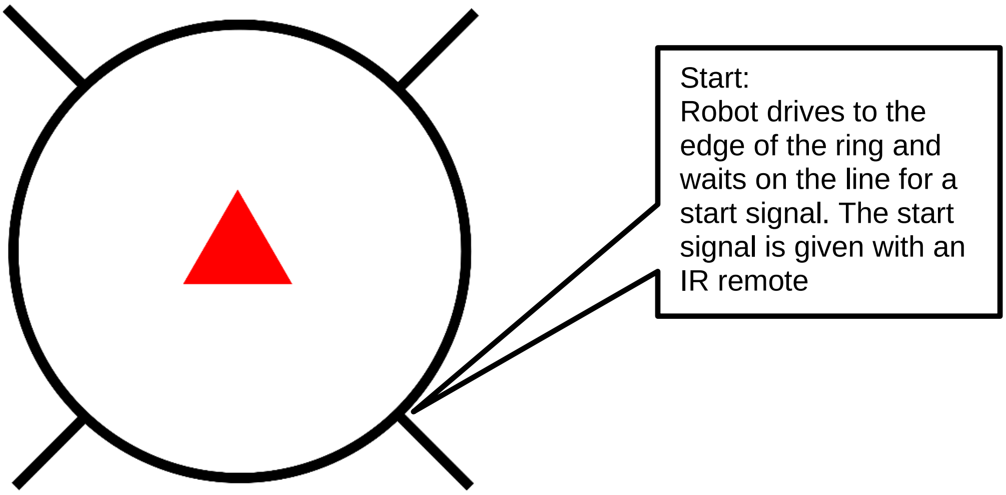
Do not use spaces in the top-level topic. Subtopics to use depend on the project in question. See project descriptions for details on what to send.

All data must be sent as text without linefeeds. If you send multiple values separate them with spaces. MQTT is a packet protocol so there is no need to send linefeeds with the data. **Sending linefeeds in the competition is considered an error that lowers your project grade.**

Sumo wrestling

Robot must drive along the line to the edge of the sumo ring and wait for start signal. The start signal is given with an IR remote. The robot drives around and avoids hitting the red triangle. Robot uses reflectance sensors to stay inside the ring. After entering the ring robot must stop when user button is pressed.

Robot is considered to be out of the ring if the center of the robot crosses the line. Instructors will judge who is out and who is not.



Start:
Robot drives to the edge of the ring and waits on the line for a start signal. The start signal is given with an IR remote

Robot must send the following data over MQTT connection:

- Ready indicator when robot stops on the start line and starts to wait for IR start signal:
    - Subtopic: ready
    - Data: “zumo”
- Start time (systick based time stamp = number of milliseconds since the robot code started)
    - Subtopic: start
- Stop time (when the user button is pressed robot stops and sends stop time stamp)
    - Subtopic: stop
- Run time (from start to finish, number of milliseconds between start and stop)
    - Subtopic: time
- Timestamp, number of milliseconds since boot, of each turn to avoid the obstacle
    - Subtopic: obstacle

For example, Zumo028 sends:

```
Zumo028/ready zumo
Zumo028/start 2301
Zumo028/obstacle 3570
Zumo028/obstacle 4553
Zumo028/obstacle 4988
Zumo028/stop 5210
Zumo028/time 2909
```

This is topic/subtopic. It will be displayed automatically in the log.

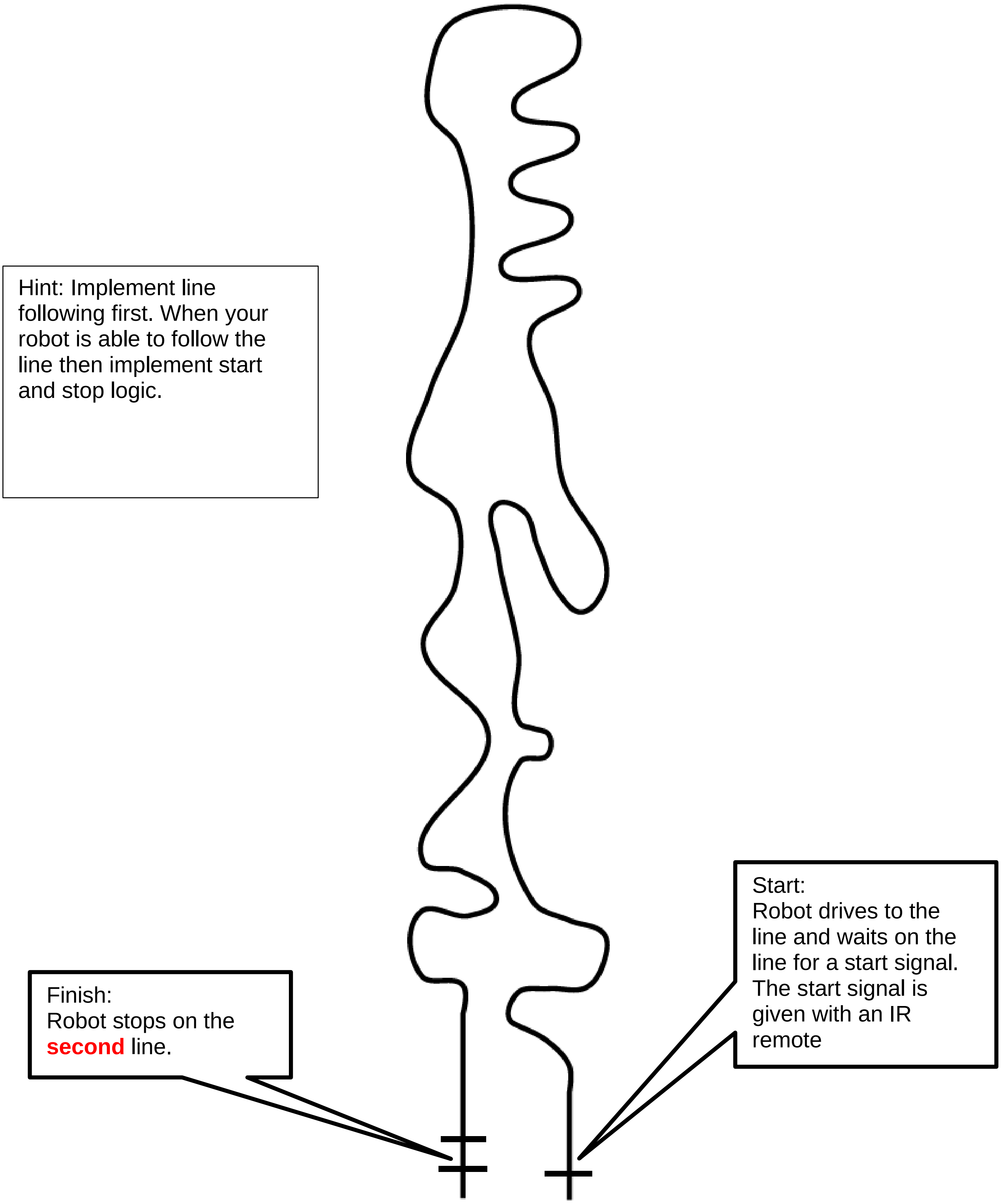
This is data. The space between topic and data is added automatically in the log. Use spaces only to separate data values if you send multiple values.

Line follower

Robot follows the black line using reflective sensors in the front of the robot. Track total length is about 10 meters. Robot has six reflective sensors. The center sensors are used mainly to keep the robot on the track and side sensors to detect intersections or tight curves on the track. The robot must follow the start and stop rules stated below. The intersections are in a 90-degree angle on a straight line.

The competition track is 2 cm wide. Make the robot run as fast as possible.

Hint: Implement line following first. When your robot is able to follow the line then implement start and stop logic.



Start:
Robot drives to the line and waits on the line for a start signal. The start signal is given with an IR remote

Finish:
Robot stops on the **second** line.

Robot must send the following data over MQTT connection:

- Ready indicator when robot stops on the start line and starts to wait for IR start signal:
  - Subtopic: ready
  - Data: “line”
- Start time (systick based time stamp = number of milliseconds since the robot code started)
  - Subtopic: start
- Stop time (when the robot stops on the finish line it sends stop time stamp)
  - Subtopic: stop
- Run time (from start to finish, number of milliseconds between start and stop)
  - Subtopic: time

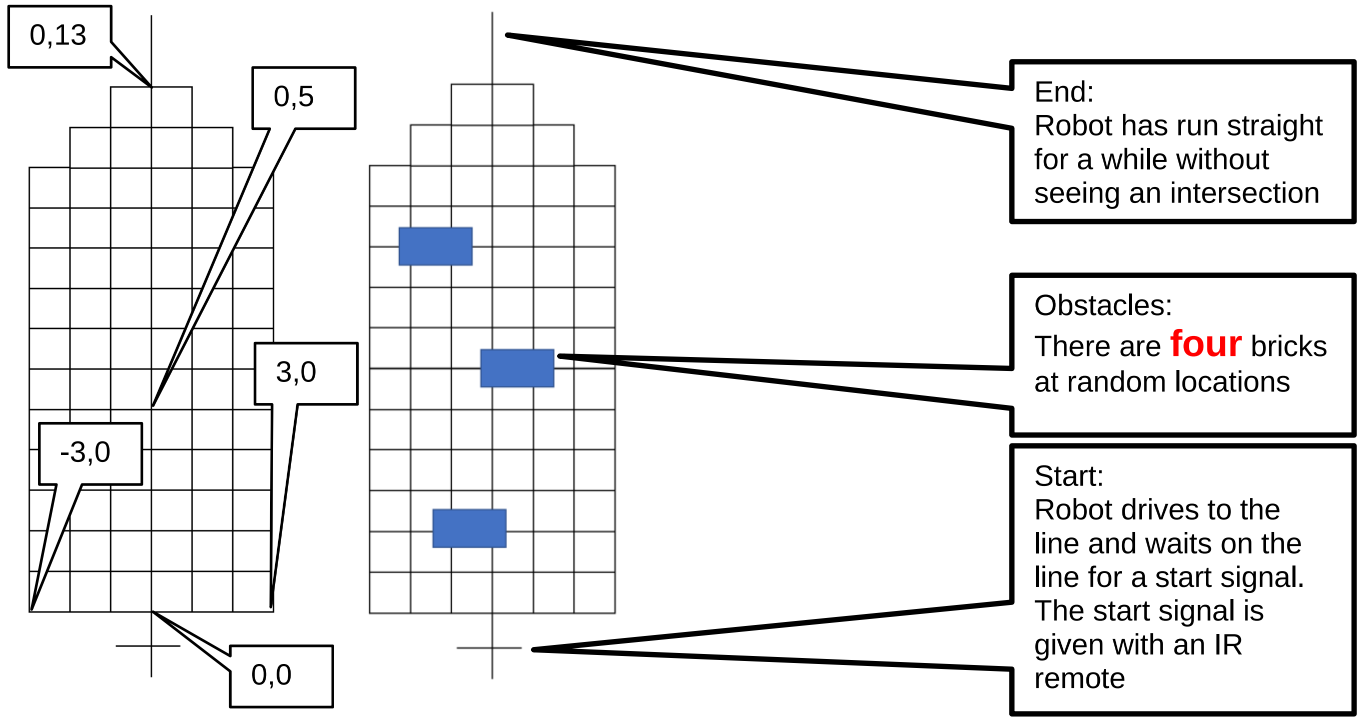
- Bonus (optional):
  - Timestamp if both centre sensors go off the track
    - Subtopic: miss
  - Timestamp when both centre sensors go back to the track after a miss
    - Subtopic: line
  - You must implement both timestamps to get a bonus

For example, Zumo028 sends:

```
Zumo028/ready line                    Zumo028/ready line
Zumo028/start 1210                    Zumo028/start 1210
Zumo028/stop 54290                    Zumo028/miss 7912
Zumo028/time 53080                    Zumo028/line 8012
                                      Zumo028/miss 32103
                                      Zumo028/line 32630
                                      Zumo028/miss 53223
                                      Zumo028/line 53933
                                      Zumo028/stop 54290
                                      Zumo028/time 53080
```

Maze

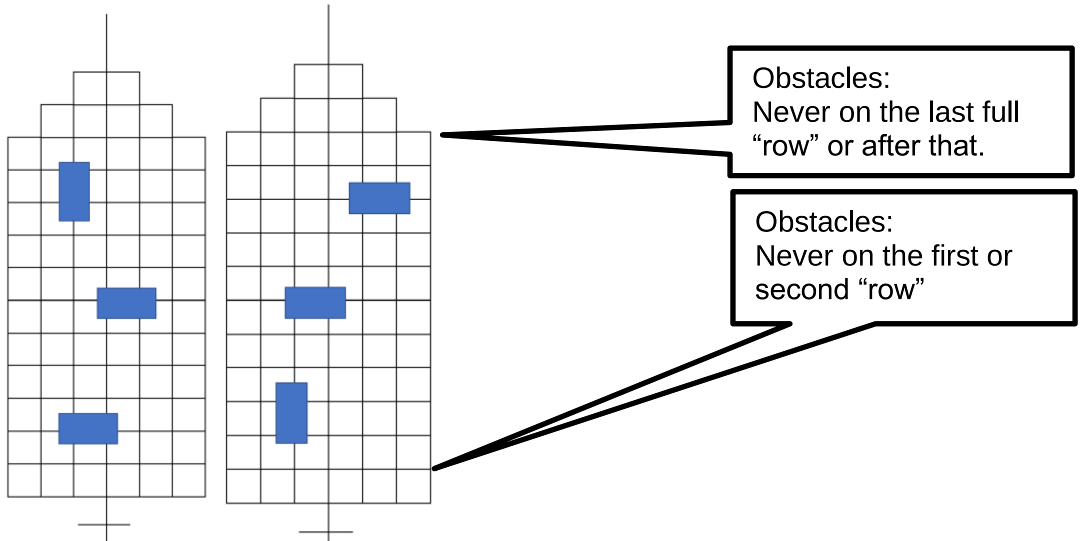
Robot finds a path through a grid by following lines and taking turns at intersections to avoid obstacles.



End:
Robot has run straight for a while without seeing an intersection

Obstacles:
There are **four** bricks at random locations

Start:
Robot drives to the line and waits on the line for a start signal. The start signal is given with an IR remote



Obstacles:
Never on the last full “row” or after that.

Obstacles:
Never on the first or second “row”

Bricks are never oriented diagonally.

Robot must send the following data over MQTT connection:

- Ready indicator when robot stops on the start line and starts to wait for IR start signal:
  - Subtopic: ready
  - Data: “maze”
- Start time (systick based time stamp = number of milliseconds since the robot code started)
  - Subtopic: start
- Stop time (when the robot stops on the finish line it sends stop time stamp)
  - Subtopic: stop
- Run time (from start to finish, number of milliseconds between start and stop)
  - Subtopic: time

- Bonus (optional):
  - send coordinates of each intersection the robot visits
    - Subtopic: position

For example, Zumo028 sends:

```
Zumo028/ready maze
Zumo028/start 1210
Zumo028/stop 24290
Zumo028/time 23080
```

```
Zumo028/ready maze
Zumo028/start 1210
Zumo028/position 0 0
Zumo028/position 0 1
Zumo028/position 0 2
Zumo028/position 0 3
Zumo028/position 0 4
Zumo028/position 0 5
Zumo028/position -1 5
Zumo028/position -2 5
Zumo028/position -2 6
Zumo028/position -2 7
…
Zumo028/position 0 13
Zumo028/stop 24290
Zumo028/time 23080
```