

Product Specification

Line-Follower Software
for Pololu Zumo32U4

Table of Content

1	Target Definition.....	3
2	Product Application	3
3	Product Overview	3
4	Product Functionality	4
4.1	Rules	4
4.2	Playfield	4
5	Product Data.....	5
6	Quality Requirements.....	5
7	Additions	5

Note: Product specification structure according to recommendation of Balzert, Helmut: Lehrbuch der Software-Technik, 2 Bände, 1998, Spektrum Akademischer Verlag

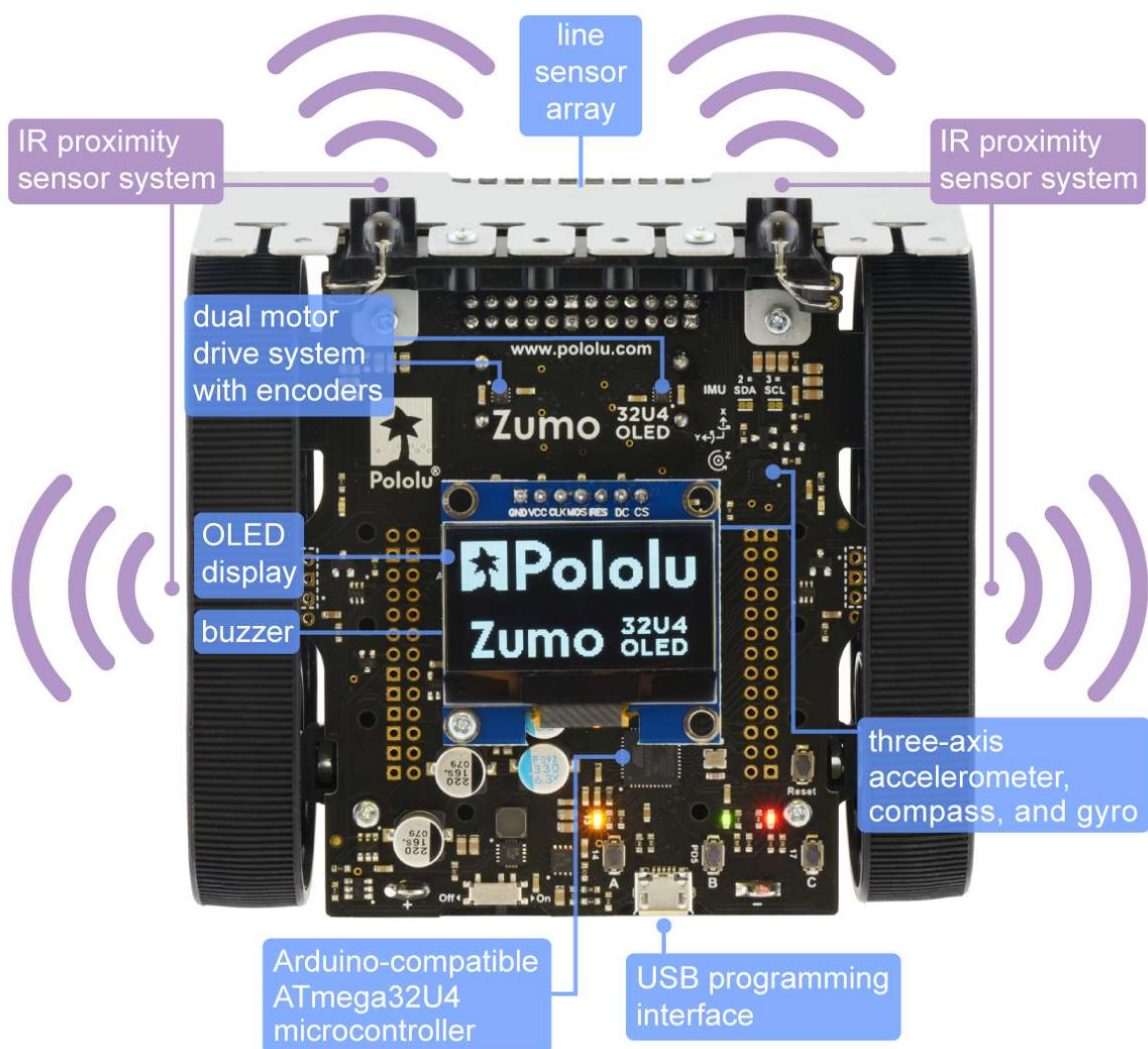
1 Target Definition

The Line-Follower software runs on the Zumo32U4 robot platform from Pololu. The robot drives a defined track as fast as possible from the start area to the finish area. The track is marked with a black line on white background.

2 Product Application

The Zumo32U4 robot from Pololu is used for line-follower competitions. The team with the robot that drove the fastest complete round wins.

3 Product Overview



4 Product Functionality

The main functions of the product derives from the competition rules and the playfield.

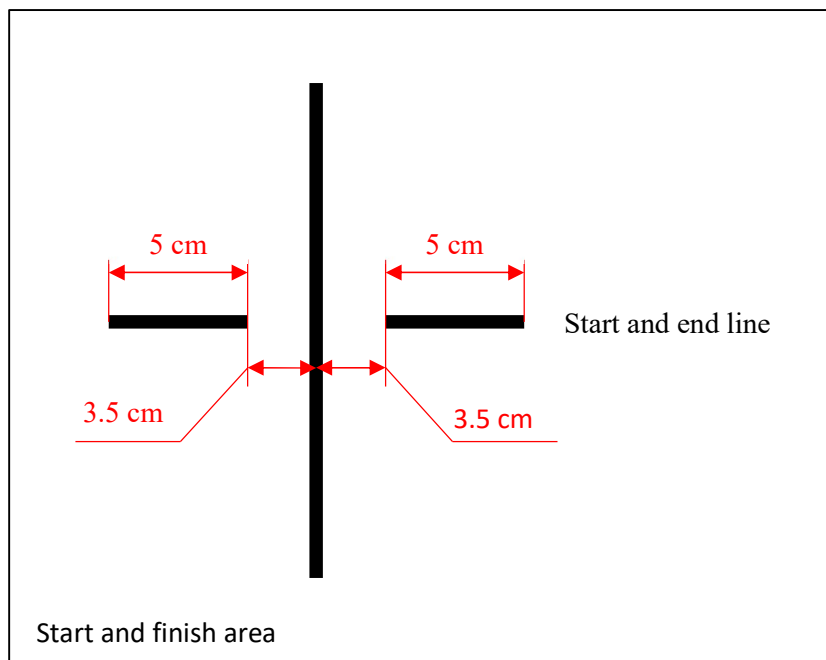
4.1 Rules

- The robot drives autonomously and it is not allowed to control it remotely via cable or wireless.
- The team is not allowed to make any modifications on the hardware during executing the competition, except changing the batteries or small repairs.
- Each team is able to test the robot on the track before the competition starts. During this phase it's allowed to change the software as well.
- In the competition phase it's only allowed to change pre-defined parameter sets.
- The robot must begin its run on the track and at least 1 cm from the start line.
- The start release is given via pushbutton on the robot, which will start driving after 3s.
- If the robot detects the start line, it will notify with a short beep and the time measurement for the lap starts.
- If the robot detects the end line, it will notify with a short beep and the time measurement stops. The measured time is displayed on the OLED.
- The robot automatically stops after the end line is detected.
- Each team has 3 attempts on the track. It counts the fastest complete round.
- The robot must finish one complete round on the standard track in maximum 20s, otherwise the run is considered as failed.
- If the robot leaves the track and is unable to re-detect the track within 5s, the run is considered as failed.
- If the robot detects an error, it will notify with an alarm tone and display the error reason on the OLED. Furthermore, it will stop.
- After power-on, the robot displays the team number for at least 2s.

4.2 Playing Field

- The playfield background is white.
- The track is marked with a black 1.5 cm wide line.
- The track does not cross itself.
- The track can contain gaps, which will continue after at least 10 cm in an opening angle of 30°.
- The start and end line are marked with 90° crossbeam. The crossbeam will be in a distance of 3.5 cm to the center line and has a length of 5 cm.
- The start and end line are minimum 30 cm away from any gap in the track.
- The minimum distance between playfield border and the track is 15 cm.
- The minimum curve radius on a track is 10 cm.

Graphic overview:



5 Non-functional Requirements

- The software must be designed in a way that allows to replace the hardware by e.g. a different robot or a hardware simulation. Unavailability of hardware features does not need to be considered.
- The software must work with normal daylight & workplace light conditions.
- The software must work independently of the charge status of the batteries, unless the charge status is below 50%.
- The software must be implemented in C.

6 Quality Requirements

The flash memory usage must be lower than or equal to 80%.

7 Documentation Requirements

- All documents must be written in English.
- Commit messages of version control system interactions must be in English.
- Commit messages of version control system interactions must be descriptive.
 - Non-descriptive commit messages like “Bug” or “Update” are not allowed.