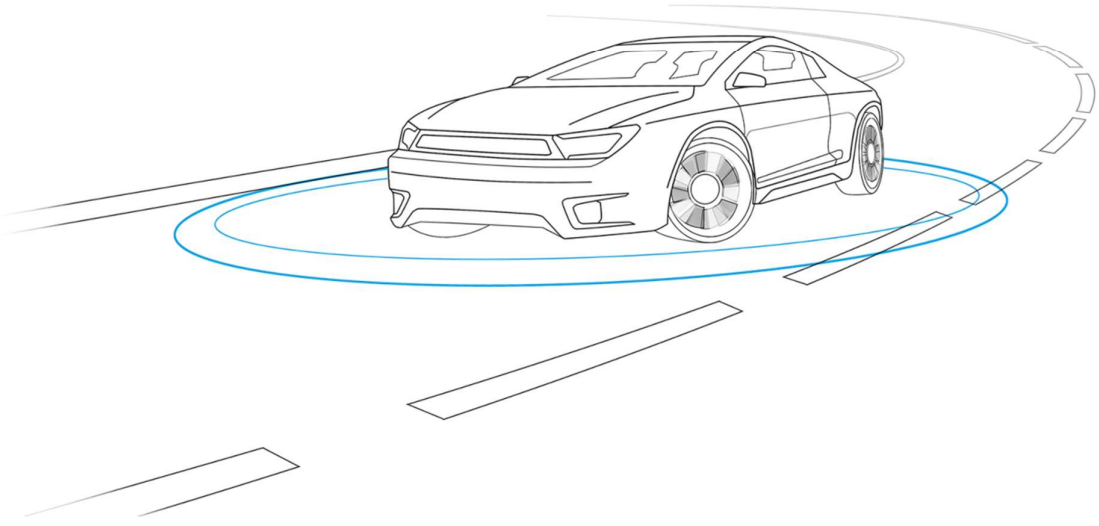
Automotive Safety Technologies GmbH
Lilienthalstraße 11, D-85080 Gaimersheim | +49 841 89 95000

Data of the document

A. Manolica, V. Chernukhin, V. Muresan	**Author**
5.0	**Document version**
released	**Release state**
16/04/2020	**Release date**

ASTech Student Tournament 2020
Regulations



Document history

Version	Date	Author (department)	Description of change
1.0	26.09.2019	Alexandru Manolica Victor Muresan Victor Chernukhin	Initial Version
2.0	31.10.2019	Alexandru Manolica Victor Muresan Victor Chernukhin	Initial version reviewed and final version drafted
3.0	19.12.2019	Alexandru Manolica Victor Muresan Victor Chernukhin	Version 2 reviewed and Section 5.3 updated
4.0	01.02.2019	Alexandru Manolica Victor Muresan Victor Chernukhin	Version 3 reviewed. Milestones updated.
5.0	16.04.2020	Alexandru Manolica Victor Muresan Victor Chernukhin	Version 4 reviewed. Circuit draws added to Appendix

Table of Content

1 Overview

1.1 Objectives

The "ASTech Student Tournament 2020" is a competition between student teams to get involved with the conceptualization and implementation of automated model vehicles. The goal of this tournament is to develop a model vehicle that performs tasks derived from requirements imposed by realistic environments. This annual competition offers the participating students the opportunity to present their know-how in front of a jury from ASTech and the University “Gheorghe Asachi” Iasi.

1.2 Tasks

Each student team is responsible for developing, assembling and demonstrating a cost efficient concept of an automated model vehicle. For the ASTech student tournament 2020, the tasks involve lane detection, lane keeping assistance and Car-to-X communication. In the last milestone “Final Tournament”, each team should present the results in front of the jury and fellow students.

2 General Information

The participation to this competition is allowed for teams composed of students from Automatic Control and Computer Engineering Faculty. Every participant should take into consideration following information.

2.1 Team Members

The initial number of participating teams is not limited. However, the number of members per team is limited to three students.

2.2 Registration

In order to register for the competition, each team must send a letter of motivation and the CVs of the team members via E-mail to lpa@ac.tuiasi.ro until 15th November 2019. Additional information can be also found on http://lpa.ac.tuiasi.ro/.

2.3 Prizes

The winners of the tournament will be invited to Ingolstadt to visit ASTech GmbH and the Audi AG headquarters. During their visit, the students will get an insight into the automotive world. By visiting the Audi assembling lines or insights into ASTech projects, the winners will experience automotive development from ideas to final products.

2.4 Additional Info

This document can be modified and will be updated when necessary. Old versions of the document are invalidated, as soon as a new version is published. Any updates will additionally be announced to registered teams. The updates and other information will be available at: lpa.ac.tuiasi.ro/ast.php and LPA Facebook page.

Exchanging ideas and open discussions among the teams is fully encouraged. The students are encouraged to do research and discuss their issues with the jury composed of engineers from ASTech and supervisors aside the university. The teams are encouraged to contact the jury early in case of doubts or questions about a particular component.

3 General Requirements

The model vehicle of each team will be designed based on the provided kit. This development kit consists of 3D-printed chassis and parts, as well as Raspberry Pi 3b and Arduino Uno boards, sensors and connectors.

Language for all documents, code or other form of text (such as in pictures) must be English. Moreover, templates for the documents that need to be created (requirements, test specification etc.) will be made available.

The operating system for Raspberry Pi should be Ubuntu and the Software Environment should be Robot Operating System (ROS). Testing of software should be done using a simulation in Gazebo.

All software must be written in C, C++ or Python and version control in form of a git repository must be used. The software modules on the Raspberry Pi must be ROS nodes. This means you are not allowed to bypass ROS and run your functions within another infrastructure or middleware. For the software modules on the Arduino board and for the communication between Raspberry Pi and Arduino it is not necessary to use ROS nodes.

All algorithms have to be self-developed. If the participants want to use already available libraries, this can be done only with the prior approval of the jury and usage should be fully described in the final documentation.

The software on the vehicle should be able to handle all scenarios during the final tournament.

4 Vehicle Requirements

The students are free to create their own model vehicle using the development kit with respect to the following requirements.

4.1 Drivetrain

The vehicle must be equipped with an electric motor.

4.2 Energy Supply

Batteries are used to supply the model vehicle with energy. Changing and charging the batteries during the final tournament is permitted.

4.3 Physical Dimensions

The width of the vehicle, including possible extensions and bodywork, must not exceed 150 mm and must be at least 80 mm.

The length of the vehicle, including possible extensions and bodywork, must not exceed 300 mm and must be at least 200 mm.

The height of the vehicle, including fixed installations, must not exceed 300 mm above the track surface.

The vehicle must have four wheels and two axles. At least one axis must be steerable.

These dimensions will be checked bevor the competition and deviations from the requirements will lead to deduction of points.

4.4 Lights (optional)

As in real traffic situations, the model vehicle can be equipped with lights to signal different driving maneuvers. These requirements are optional and their implementation will bring additional points.

4.4.1 Headlamps

The vehicle must be equipped with white headlamps. The headlamps must be on during the drive. White LEDs can be used as headlamps.

4.4.2 Braking Lights

Three red LEDs must be installed at the rear of the vehicle to signal braking maneuvers.

4.4.3 Direction Indicators

Each corner of the vehicle must be equipped with a yellow or orange LED to signal overtaking, turning. After an emergency braking, the lights must flash for 2 seconds at 50% duty-cycle.

4.5 The Development Kit

The basic components for the model vehicle are provided in the development kit.

Each team is free to decide which sensors will be used to fulfill the tasks specified in Section 5, as long as the functionality of the vehicle is assured.

The use of other sensors or more sensors than those provided is allowed only with prior approval of the jury.

Component
Raspberry Pi 3b board
Arduino Uno board
Magnetic encoder
Motor driver
Servo motor nano
Infrared sensors
Ultrasonic sensor
Raspeberry Pi Camera
XBee module

5 Functional Requirements

5.1 Lane Keeping Assistant

The model vehicle must detect its driving lane and move forward within it. Each lane has a nominal width between 300-400 mm.

Colored adhesive strips mark the sides of the road, i.e. they represent the boundaries of the lane.

The road can be a one way road and/or can have several lanes in one direction.

A straight lane may be interrupted. The vehicle must continue to keep the lane during and after this gap.

Another lane may intersect the vehicle's lane, i.e. the vehicle may drive through crossroads. The vehicle must not follow an intersecting lane, but must continue to follow its original lane.

A perpendicular continuous white line in vehicle's path represents the demarcation line in an intersection. When a demarcation line is detected, the vehicle must stop for 2 seconds and only afterwards continue driving.

5.2 Emergency braking

While the model vehicle drives within the lane, it must not collide with obstacles laying in its path. In this case, the vehicle must perform an emergency braking or an overtaking maneuver.

The braking lights must be triggered for 2 seconds after an emergency braking (optional requirement). After an emergency braking, the direction indicators must flash for 2 seconds at 50% duty-cycle (optional requirement).

If the obstacle appears suddenly in front of the vehicle at a distance between 200 mm and 800 mm, the vehicle must break to standstill. The front bumper of the ego-vehicle must not touch the obstacle. The vehicle will continue its movement after the obstacle is removed from the track.

If the obstacle appears or is standing still in front of the vehicle at a distance between 800 mm and 1200 mm, the vehicle must perform an overtaking maneuver, i.e. the vehicle must change the lane and return to its original lane after this maneuver.

The dimensions of the obstacle will be between 150 and 250 mm.

5.3 Car-2-X Communication

All vehicles will receive a specific message broadcasted by the master. The master in the Car-2-X communication will be the computer and all the vehicles will act as slaves.

The frame of a broadcasted message is composed of 5 bytes. The start and end bytes, i.e. the first and fifth byte in the message frame, have a fix value in order to identify new incoming messages. The second byte in the frame represents the number of the car. Hence, each model vehicle will be assigned a specific number before start of Milestone "Implementation on the Model Car".

The third byte in the frame addresses functional commands such as: *no restrictions, driving, lane, speed* and *overtaking*. The fourth byte in the frame represents the value of the function. A more detailed overview is given in the following table.

Byte 1 Start of Frame	**Byte 2** Car number	**Byte 3** Functionality		**Byte 4** Value of function		**Byte 5** End of Frame
		Value	Meaning	Value	Meaning	
AA (hex) = 1010 1010 (binary)	0 = All cars [1, …] = Car *X*	0	No restrictions	x	Not applicable	55 (hex) = 0101 0101 (binary)
		1	Driving	0	Not allowed	
				1	Allowed	
		2	Lane	0	No restrictions, i.e. driving allowed on all lanes	
				1	Driving allowed only on lane 1	
				2	Driving allowed only on lane 2	
		3	Speed	0	Speed adapted according to the situation	
				1	Low speed driving	
				2	Nominal speed driving	
				3	High speed driving	
		4	Overtaking	0	Not allowed	
				1	Allowed	

The installation steps and configuration of XBee are described in the “XCTU_config” document. Please refer to this document for further information.

6 Competition Milestones and Scoring

During the competition, various milestones have to be reached. An overview can be found in Section 7.3. For each milestone, you will score points. Your final score will be the sum of all points gathered in each milestone.

The work progress of each team will be checked not only after each milestone but also in the middle of the milestone. During this check, each team will receive a feedback from the jury. The feedback should be used by the teams to improve their approach in order to achieve maximum of points in the milestone.

If a team does not reach the minimum number of points required for the milestone, it will be disqualified from the competition.

After the milestone “Gazebo Simulation” the five teams that achieved the highest scores will receive the development kit for the final tournament. The other teams that achieved the minimum number of points in the milestone and are interested in remaining in the competition must acquire the development kit on their own costs.

6.1 Milestone "System Specifications"

The teams must document following specifications using the provided template or similar one:

- System architecture / block diagram
- Input - Output description
- General description of each functional block (e.g. target detection, controller etc.)
- Test specification - description of the tests that will be performed in order to verify that the functional requirements (see Section 5 "Functional Requirements") are fulfilled.

The documents must be send to the jury until 6th of January 2020 via lpa@ac.tuiasi.ro.

If necessary, all the above documents can be further updated during the competition. Additional points can be achieved if the documents are improved.

6.2 Milestone "Gazebo Simulation"

Within this milestone, the functional requirements shall be implemented and tested in form of a Gazebo simulation.

The Gazebo simulation and the implemented tests must be sent to the jury until mid of March 2020 via lpa@ac.tuiasi.ro. The exact date will be announced in time on lpa.ac.tuiasi.ro/ast.php.

6.3 Milestone "Implementation on the Model Car"

The final software and the final test results must be sent to the jury until mid of May 2020 via lpa@ac.tuiasi.ro. The exact date will be announced in time on lpa.ac.tuiasi.ro/ast.php. This must contain all git repositories, which were used during development, including the git history.

After the milestone “Implementation on the Model Car”, no further modifications of the vehicle or its software are allowed.

6.4 Milestone "Final Tournament"

The final tournament will take place end of May 2020. Exact date will be announced in time on lpa@ac.tuiasi.ro

The students will have to present their results in form of a PowerPoint presentation or a poster.

During the final tournament the vehicle has to perform a course with various tasks as described in Section 5: "Functional Requirements". An exemplary representation of the course will be provided during the tournament.

7 Appendix

Within this section, some useful information, tutorials and technical documentation are provided.

7.1 Circuit

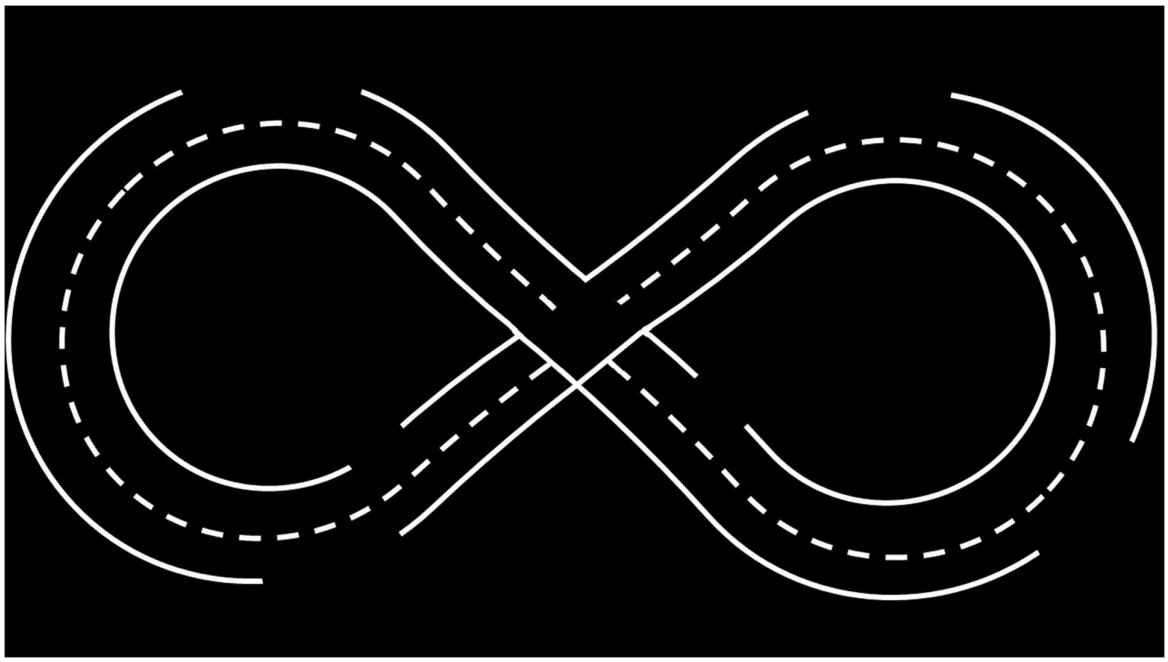


Figure 1: Official circuit.

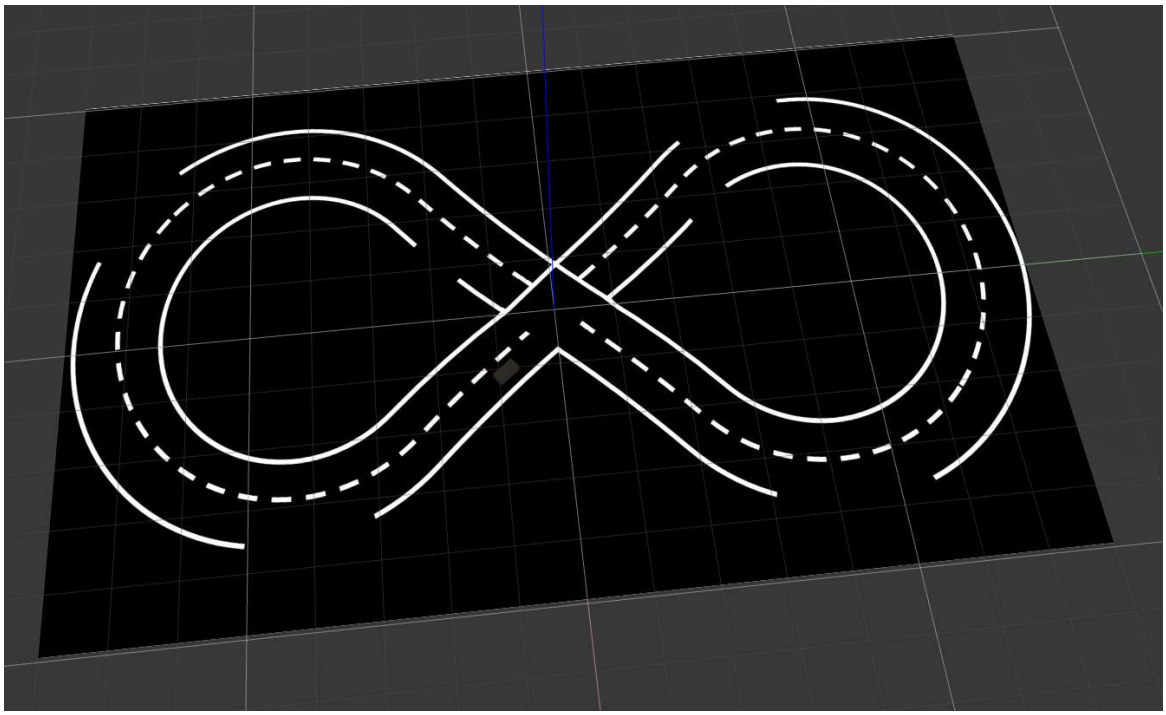


Figure 2: Circuit representation in Gazebo.

7.2 Technical Specifications

- Camera module: https://www.raspberrypi.org/products/camera-module-v2/
- Legacy Digi XBee: https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-modules/xbee-802-15-4#specifications-legacyxbees1802154
- Ultrasonic sensor: http://anleitung.joy-it.net/wp-content/uploads/2017/11/SEN-US01-Datasheet.pdf
- Analog distance sensor: https://www.pololu.com/product/2474
- Digital distance sensor: https://www.pololu.com/product/1134
- Gearmotor: https://www.pololu.com/product/3065
- DC motor driver carrier: https://www.pololu.com/product/2961
- Magnetic encoder: https://www.pololu.com/product/3081
- Logic level shifter: https://www.pololu.com/product/2595

7.3 Tutorials

- Robot Operating System (ROS): http://wiki.ros.org/ROS/Tutorials
- Gazebo: http://gazebosim.org/tutorials
- Git: https://www.atlassian.com/git

7.4 Tournament Timeline

The following table gives a general overview of the tournament's timeline.

Due date	Action
01 – 15.11.2019	Registration period
20.11.2019	Announcement of participating teams / Start of competition
11 - 18.12.2019	Work progress check – "System Specifications"
06.01.2020	Milestone "System Specifications"
27-28.02.2020	Work progress check – "Gazebo Simulation"
23-27.03.2020	Milestone "Gazebo Simulation"
27-30.04.2020	Work progress check – "Implementation on the Model Car"
13-15.05.2020	Milestone "Implementation on the Model Car"
18-19.05.2020	Milestone "Final tournament"
To be announced	Visit of ASTech GmbH and Audi AG in Ingolstadt