Design Report: The Automatic Parking System for DE2Bot

Soo Kwon Ha
Geon Woo Kang
Geun Tae Park
Woo Joo Shin

ECE 2031 – L11
Georgia Institute of Technology
School of Electrical and Computer Engineering

Submitted
April 25, 2017

Abstract

This document is a design report of an automatic parking system that allows the user to drive the DE2Bot manually and park the Bot autonomously. First, the system will have functionality to park the bot in perpendicular parking spaces fully and semi-autonomously. Second, it will allow the user to park the bot semi-autonomously in parallel parking spaces. To achieve the listed functions, the system utilizes the Simple Computer implemented on DE2-Board to control bot's hardware, such IR receiver and timer, movement API of DE2Bot, and an IR remote. The user can manually drive the bot using arrow buttons and power button on the IR remote. And, the user can make fully-autonomous parking in perpendicular spaces using number buttons from 1 through 7. Furthermore, after the user successfully position the Bot next to the intended parking space, the user can automatically park either in parallel or perpendicular parking spaces by pressing either number 0 button or number 9 button, respectively. To implement the semi and fully autonomous parking functions, the travel time was calculated, by measuring the travel distances and using the speed of DE2Bot of 10.4cm/s, then used to write subroutines to form previously determined automatic movement sequences. Finally, the subroutines were put together to form the automatic parking system.

Design Report: The Automatic Parking System for DE2Bot

Introduction

This document examines an implementation of automatic parking system on a DE2Bot. The requirements for the parking systems were:

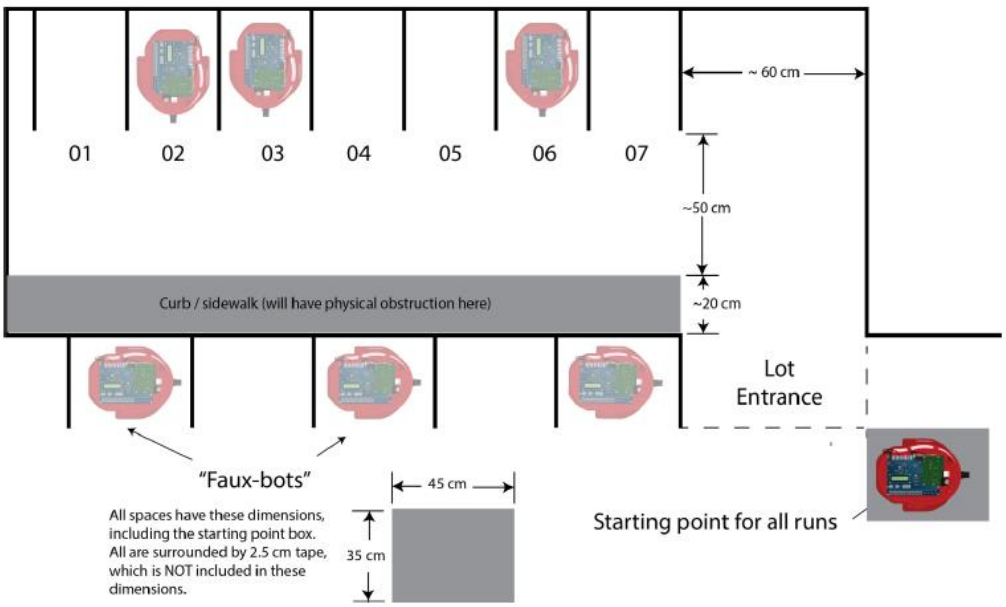


Figure 1. Layout of parking spaces and dimensions.

1. Manual drive mode which allows user to control the movement of the Bot using an IR remote.
2. Semi-autonomous parking function for perpendicular and parallel parking spaces.

   (Perpendicular parking spaces, numbered 1 through 7, are located above curb/sidewalk, and parallel parking spaces are located below curb/sidewalk in Figure 1 above.)
3. Fully-autonomous parking function for perpendicular parking spaces.

   (Parking spaces numbered 1 through 7 in Figure 1.)

The designed system utilizes Simple Computer (SCOMP) to communicate with hardware of DE2Bot, timer implemented in SCOMP, the Bot's movement API, and an IR remote to achieve the requirements listed above.

General Methodology

To implement the required functions of automatic parking system, the designing process was broken down to five major parts:

- Path mapping for DE2Bot's movements
- Determination of movement sequences required for DE2 Bot using mapped out path
- Assignment of IR remote buttons for functions
- Implementation of manual drive mode using IR remote
- Implementation of automatic movements

The listed parts for design process was broken down in that manner to distribute work load among team members to allow parallel processing of system implementation.

Path Mapping

The purpose of path mapping process was to determine the dimension of paths that DE2Bot will take for autonomous parking.

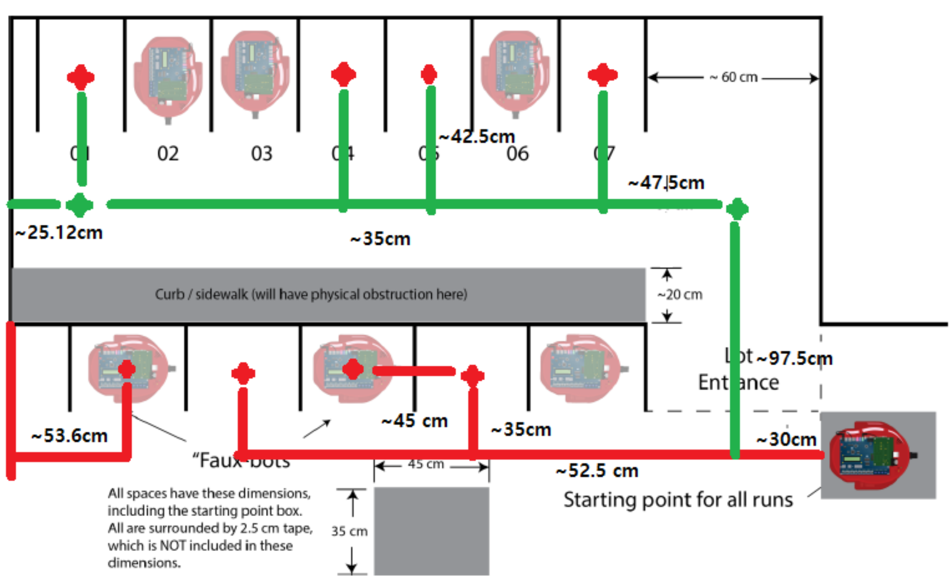


Figure 2. Dimensions of parking lot, green path is for both semi and fully autonomous parking in perpendicular spaces, red path for semi-automatic parallel parking.

The figure 2 above show paths that bot will take for the semi and fully autonomous parking mode which are represented in green and red lines. Both red and green paths were measured precisely to align the Bot with center of allowed path and intended parking space. For the fully-autonomous parking mode, the green path was used to determine the travel distance and required movements for the Bot. Furthermore, the green path was used to guide the user to align the Bot correctly with the intended perpendicular parking space in semi-autonomous mode. Lastly, the red path was used to determine the required alignment and movements of DE2Bot for semi-autonomous parallel parking function. Overall, this path mapping process was a crucial part of design process since it will be used to determine the required movements for the Bot and possible subroutines that will be shared between functions.

Determination of Movement Sequences

After mapping the path for DE2Bot, necessary movement sequences were determined for each function. The movement sequence for fully-autonomous perpendicular parking function was determined to be:

Fully Automatic Perpendicular Parking Actuation Sequences

1) Move bot forward by 30cm
2) Turn right (90° clockwise)
3) Move forward by 97.5cm
4) Turn left (90° counter-clockwise)
5) Move forward by the determined distance to align with center of parking space (each space will have specific forward distance, increment of 35cm)
6) Turn right (90° clockwise)
7) Move forward by 42.5cm
8) Stop and wait for new IR code

Figure 3. Steps for fully autonomous perpendicular parking.

The step 5 of actuation sequences above varied for each parking space furthest being 257.5cm (parking space 1) and closest being 47.5cm (parking space 7). The steps 1 through 4 and 6 through 8 were written as subroutines to be shared for fully-autonomous parking for space 1 through 7. Lastly for the semi-autonomous perpendicular and parallel parking functions, following actuation sequences were used:

Semi-Automatic Perpendicular Parking Actuation Sequences

*Center of the bot must be 47.5cm away from the center of perpendicular parking space

1) Turn right (90° clockwise)
2) Move forward by 42.5cm
3) Stop and wait for new IR code

Semi-Automatic Parallel Parking Actuation Sequences

*Center of the bot must be 35cm away from the center of perpendicular parking space

1) Turn right (90° clockwise)
2) Move forward by 35cm
3) Turn left (90° counter-clockwise)
4) Stop and wait for new IR code

*For the automatic parking to work correctly, the parking space must be on the right side of bot, and the bot must be aligned with the center of parking space

Figure 4. Steps for semi-autonomous parking modes.

The listed actuation sequence for semi-autonomous parking functions are highly dependent of correct alignment of Bot with intended parking space. Also, the semi-autonomous perpendicular parking actuation sequences were determined to be identical to steps 6 through 8 of fully-autonomous function. Therefore, the previous written subroutine was used for semi-automatic perpendicular parking.

Assignment of IR Remote Buttons

Once all the required movements were determined, the buttons on IR remote were assigned to do specific functions. To make the assignments possible, first, the hexadecimal codes of IR remote buttons were determined, listed in figure 5 below.

IR_LOW Values of IR Remote	
POWER	10EF
PREV CHANNEL	58A7
MUTE	906F
VOLUME UP	40BF
VOLUME DOWN	C03F
CHANNAL UP	00FF
CHANNAL DOWN	807F
NUMBER BUTTONS	
0	08F7
1	8877
2	48B7
3	C837
4	28D7
5	A857
6	6897
7	E817
8	18E7
9	9867

Figure 5. IR_LOW values of buttons in hexadecimal.

These hexadecimal values were obtained in TV mode on IR remote. Also, there was change in hex code value for previous channel button after consecutive presses. Therefore, it was later eliminated from available buttons.

Once the IR_LOW values of buttons were determined, each button was assigned a specific function. The flowchart below shows overall algorithm and assignments of buttons for the automatic parking system:

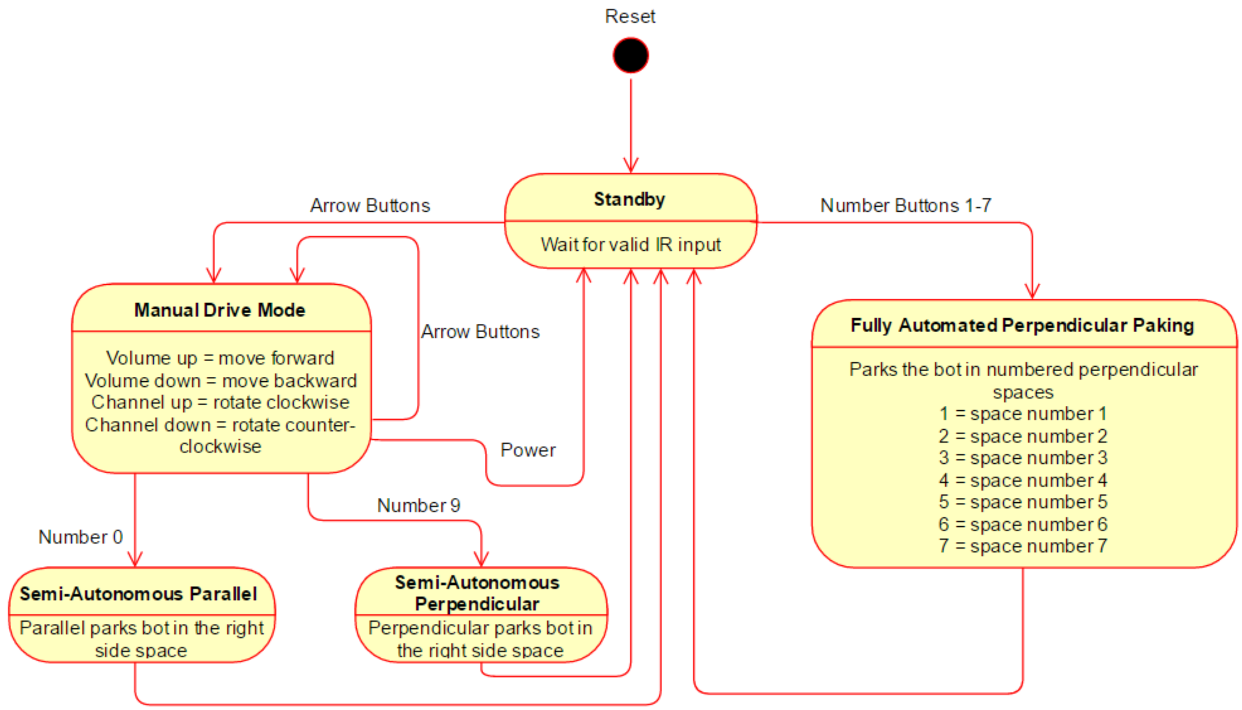


Figure 6. Flowchart of automatic parking system showing two distinct modes, semi-auto parking on the left and full-auto parking on the right. Semi-parallel is activated by number 0, semi-perpendicular is activated by number 9, and fully automated perpendicular is activated by number 1-7 buttons on the remote.

As described in the flowchart, once the arrow buttons on the remote (volume and channel buttons) are pressed, manual drive mode will be activated. And, although the autonomous parking functions can be activated at any time (by pressing number buttons 1 through 7, 9 and 0), the flowchart was made to fit the intended purpose of the system (i.e. final demo). Also, it should be noted that once the Bot goes into fully-autonomous function, it cannot be interrupted due to design limitation.

Implementation of Manual Mode

By using the assigned buttons for manual movements, manual drive mode was implemented with the movement API of DE2Bot. The figure 7 shows detailed flowchart of Bot in manual drive mode.

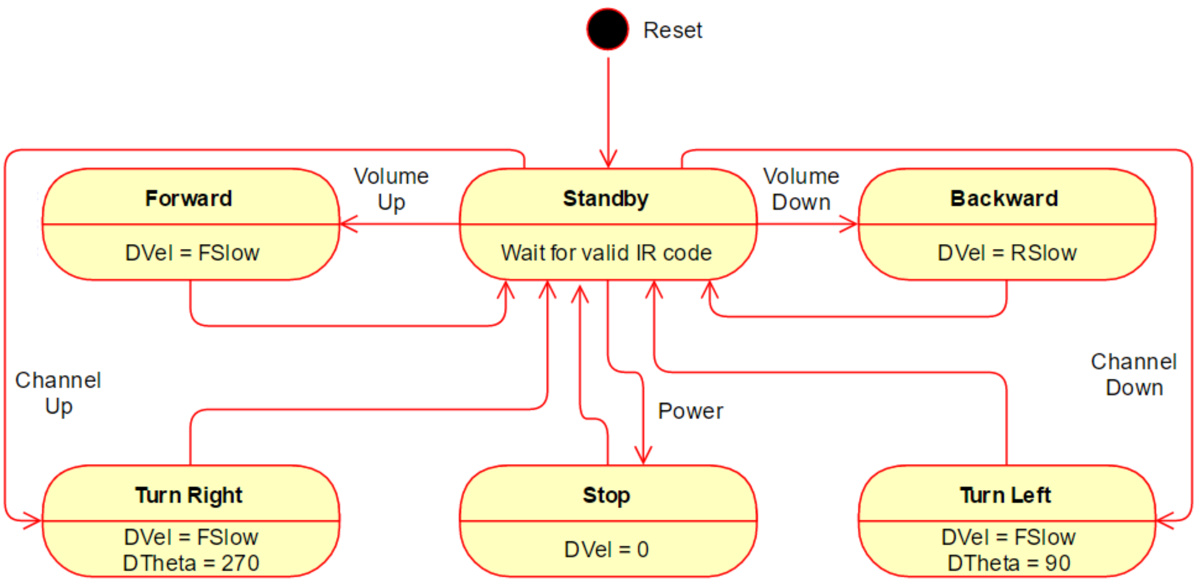


Figure 7. Flowchart of manual drive mode.

The manual drive mode is solely implemented by setting velocity and heading of the Bot to desired values. Also, to achieve car like movement, the desired velocity for right and left turns were set to FSlow. Again, the buttons were assigned according to the flowchart.

Implementation of Automatic Movements

To move the Bot certain distance, the system used the timer available through SCOMP. First, the movement speed was set to FSlow; the slow speed was chosen due to the drift of DE2Bot in high speed even with the control movement subroutine. The FSlow had value of 10.4cm/s. By using the numerical value of FSlow and measured distances in path mapping process, the travel time of each subroutine described in movement sequence determination process was calculated.

For example, the travel time for step 1 in figure 3 was:

$$30\text{cm} \times \frac{1\ sec}{10.4\ cm} = 2.88\ sec$$

And, for the turning sequences, the subroutines written in manual mode were used.

Once the initial travel time was calculated, there were series of testing and adjustment to meet the requirement. During this process, slight differences in movement speed of DE2Bots were detected. However, it had manageable impact on implemented system during the testing and debugging processes which required more adjustment in travel time (max 0.5 sec) and turning angles (max 5 degrees).

Project Management

The design schedule had to be adjusted due to requiring two extra days to implement automatic movements. This reduced final testing period by two days, which was ok since testing was also done during the debugging period. The final schedule is tabulated in figure 8 below.

Tasks	Start Date	Duration (Days)	End Date
Path Mapping	3/27	2	3/28
Manual Drive Mode	3/27	6	4/1
Automatic Movements	3/27	8	4/3
Testing and Verification of Movements	4/3	5	4/7
Debug and Finalize Solution	4/7	9	4/15
Final Testing	4/15	3	4/17

Figure 8. Tabulated final project schedule.

The updated Gantt Chart is provided in Appendix A.

Technical Results

During the final testing period, the automatic parking system had reliable success rate. The figure 9 below is the tabulated result during the testing period.

Type of Parking	Number of Successes / Number of Trials	Success Rate
Fully-Autonomous Perpendicular	18/20	**90%**
Semi-Autonomous Perpendicular	17/20	**85%**
Semi-Autonomous Parallel	19/20	**95%**

Figure 9. Tabulated success rate of automatic parking system during the final testing period.

However, it should be noted that the semi-autonomous parking functions are highly dependent on the user's skill of aligning the Bot correctly.

For the demo, four trials were taken:

Trials	Type of Parking	Results
1	Semi-Autonomous Parallel	Collision with wall
2	Semi-Autonomous Perpendicular	Not centered in space
3	Fully-Autonomous Perpendicular	**Perfect**
4	Semi-Autonomous Parallel	**Perfect**

Figure 10. Results of demonstration.

The tabulated demo results indicated that the semi-autonomous parking functions were not reliable since they only succeed one out of three trials. The cause of this outcome can be narrowed down to user error since the functions required correct alignment of Bot. And, the cause of user error may be either driver's performance decrease due to nervousness or the driver not having a driver's license.

Conclusions

Overall, the implemented system met the required specifications by being able to perform the fully-autonomous and semi-autonomous parking under the control of user who is familiar with the manual drive capability of the system. The strengths and weaknesses of the implemented parking system are:

Strengths:

- Mimics car movements
  - Velocity is not zero when turning
- Simple concept applied
  - Simple calculations of speed, distance, and time

Weaknesses:

- Requires skilled driver
  - Semi-autonomous functions require precise positioning of bot
- Limited to slow speed
  - Change in travel speed would require new set of travel times
- Requires many testing for adjusting travel distance
  - A lot of trials to fine tune the automatic movements

If there is an opportunity for modification in the system, the usage of timer will be replaced with the odometer system of DE2Bot movement API. As one of the weaknesses, the use of timer limited the Bot's velocity to preset value (FSlow in the designed system) since change in speed will require whole new set of travel times. On the other hand, the odometer system is not effected by speed change, only the input of distances.

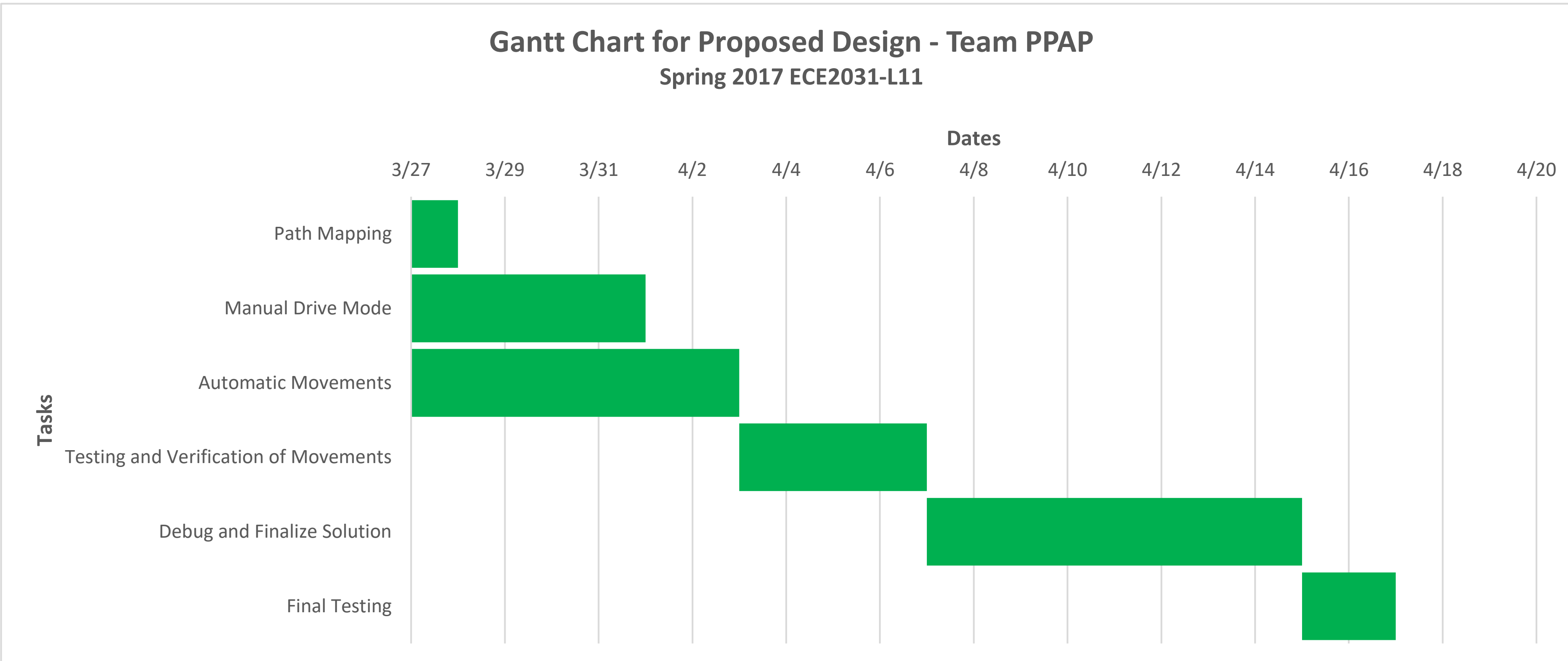
Finally, the recommendations for future engineers undertaking this task are:

- Utilize the odometer system of DE2Bot

    - A lot simpler to implement than timer approach
- Use sensors that can detect obstacle (e.g. ultrasonic sensors)
    - Use to detect how close the Bot is to surrounding obstacle and stop the bot if too close

These features will make the system more reliable. Especially, the use of sensors will allow the bot to avoid collisions with surrounding obstacle.

Appendix A: The Final Gantt Chart for The Automatic Parking System Design Project


Gantt Chart for Proposed Design - Team PPAP
Spring 2017 ECE2031-L11
Dates
3/27
3/29
3/31
4/2
4/4
4/6
4/8
4/10
4/12
4/14
4/16
4/18
4/20
Tasks
Path Mapping
Manual Drive Mode
Automatic Movements
Testing and Verification of Movements
Debug and Finalize Solution
Final Testing