NXP CUP CAR PROJECT - FALL 2019

Amrit Gautam & Jason Ramsden

Rochester Institute of Technology

ABSTRACT

Technology is becoming more advanced and has made lives easier with its new magical revolution. The revolution of technology can be seen in various ways. Autonomous technology is one of these big revolutions. Companies like Tesla are leading their way through the autonomous car market and projects like the NXP car cup provide insight into the modern real-world autonomous car technology. The goal of this project was to build, program and race to compete with other teams for the fastest time possible. The process provided some of the most important knowledge of modern engineering which includes problem solving, designing, planning, and eventually learning about components such as camera sensors and motor system control.

1. INTRODUCTION

The best way to learn something is when it is simple and effective. NXP car is simple and effective yet it's challenging. NXP cup provides real-world challenges and gives hands on experience on modern hardware and software. Embedded system software, motor control hardware combining that with camera sensor to sense and navigate through the proposed track is the approach taken through the process.

NXP cup competition is held around all over the world. Every semester RIT computer engineering students from Interface and Digital Electronic class work on the NXP project as one of the required class assignments. Students work in a team of two or three. nineteen teams competed for this semester competition which was held in the computer engineering IoT lab. Out of nineteen teams, eleven teams made it around the proposed track with some exciting and competitive cars pushing each other very closely for that top spot.

Each team had its own approach but the same goal, which was to complete the track in the fastest time possible. And each team chose their implementation based on their needs and what they thought was best. The line and edge detection algorithm were both vastly used among the teams though.

The line scan camera using algorithms and PID to determine when and how much to turn. This was done to keep the NXP car smoothly driving between the lines. It was also done with variable speed control to speed up and slow down for straightaways and turns.

The rest of the paper is organized as follows. After the introduction, Section 2 overviews the background and the basic strategy for the NXP car. Section 3 introduces the proposed method that was used for the completion of the NXP car in this project. Section 4 presents the results. Finally, Section 5 contains concluding remarks followed by references in Section 6.

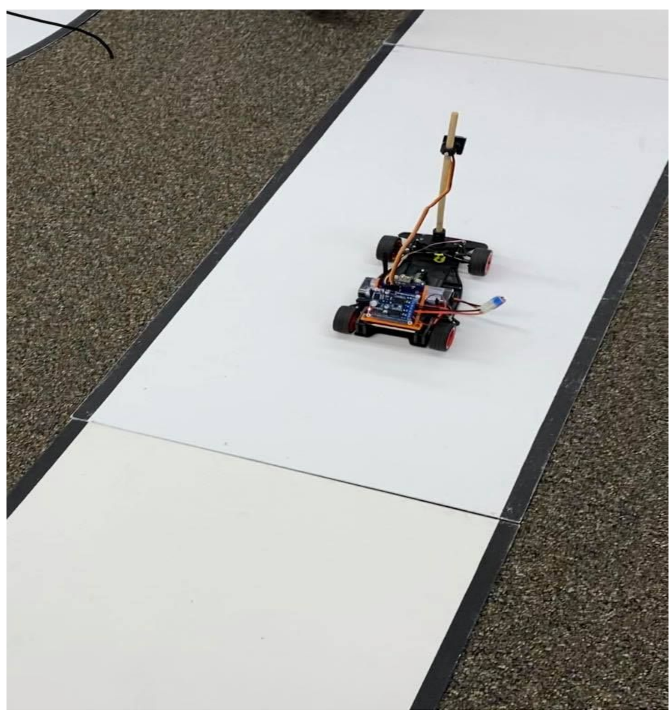

Fig. 1. NXP car doing a test run on a straightaway. It can drive smoothly between the lines without jerking back and forth because of the effectively implemented algorithms.

2. BACKGROUND

The NXP cup competition requires key concepts, ideas, and good planning to achieve the desired result. Testing is one of the key components alongside other technical strategies such as integrating proportional integral derivative controller (PID controller) for autonomous vehicles and interfacing flex timer.

The NXP cup provides students with important hands on experience to learn about modern automotive driving. It's a way for them to be as creative and to implement distinctive ideas in strive to better than other competitors. For this year's competition, almost every team priority was the idea of using a PID controller. Better camera mounting, focusing the camera lens to find the best possible images alongside coming up with the best possible algorithm to run the car as smoothly as possible were some procedures followed by each team.

Speed is a major factor in any racing competition. The main goal of this project was to finish the race faster than other competitors. The primary approach taken was to go the highest speed possible in a straight line with an increased duty cycle and slow down around certain types of turns in the track. This was all done using a camera detection algorithm to control the duty cycle in turns.

The Proportional Integral Derivative (PID) controller was the most important method used in this project. PID uses a control loop feedback to make sure we get the result we want. The use of PID allows the error reduction between actual and desired value. The main function of the PID in this project was to maintain the smoothness of the NXP car’s speed based on the track. The loop control tells cars when to speed up or slow down. Things encountered every day such as temperature controllers or vehicle cruise controllers use PID control [1].

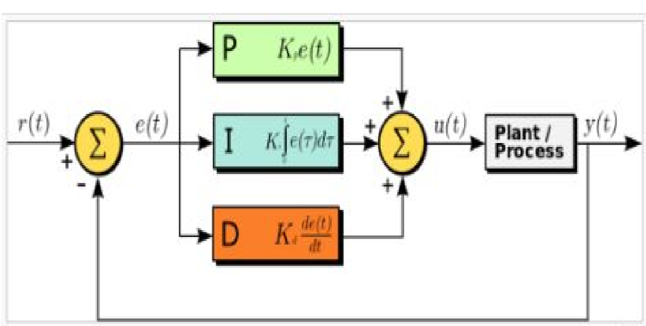

Fig. 2. PID block diagram, where r(t) is the desired value and y(t) is the measured value.

Another important component used in the project was The Flex Timer Module (FTM). FTM functions as a timer for motor control and can generate fixed cycle time interrupt for the system[2]. Using FTM modules, PWM outputs were created by interrupts which allowed the control of the desired duty cycle for the car.

3. PROPOSED METHOD

There were numerous components that went into the car shown in Fig. 3. A battery, the K64 itself, a motor shield, the body of the car and wheels, the DC and servo motors, a wooden dowel, and the 3D printed parts. All these parts went together to create the final car.

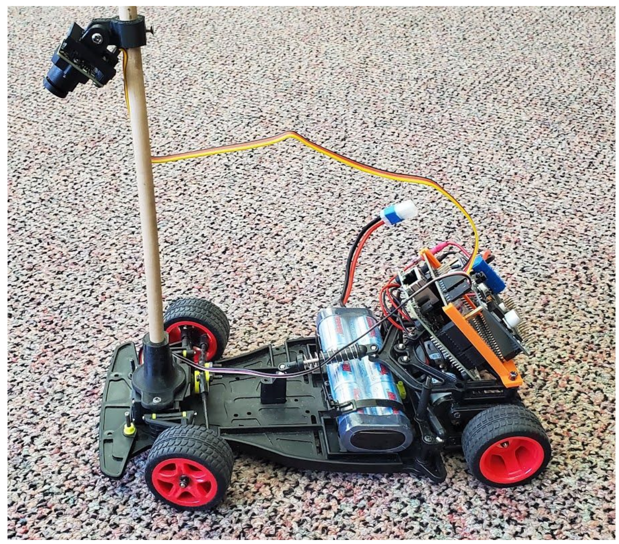

Fig. 3. NXP car that was created.

For this NXP car, there was a 3D printed camera mount, camera hinge, and camera base that made up the arm that would hold the camera on top of the wooden dowel. There was also a 3D printed control mount that was what held the K64 and motor shield assembly in place at the back of the car.

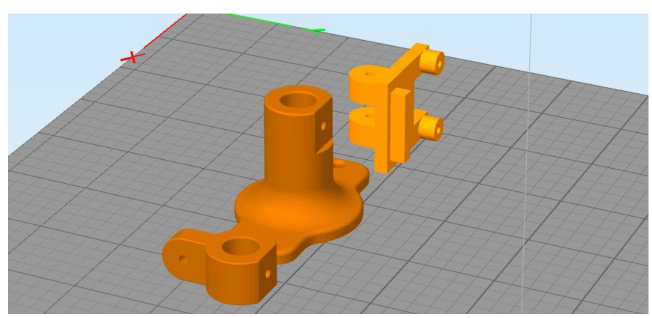

Fig. 4. 3D printed camera parts that allow the camera to screw into the mount (right), and then that mount connects to the hinge on a dowel (left). And that dowel base (middle). Note: these were printed in black filament for the NXP car.

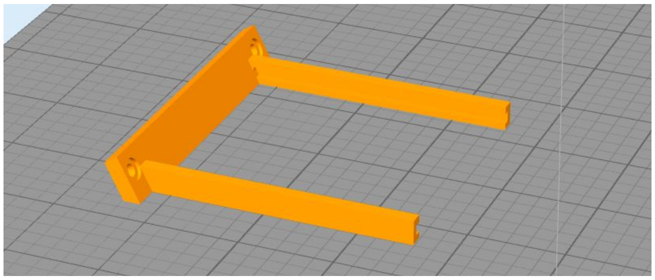

Fig. 5. 3D printed K64 mount that holds the K64 and motor shield assembly to the back of the car. Note: this was printed in orange filament for the NXP car.

The line scan camera was the first part that was set up for the NXP car. The camera needed to be focused and verified with an oscilloscope output or MatLab to make sure that the waveform that was being outputted from black lines on white paper looked like sharp rises and falls.

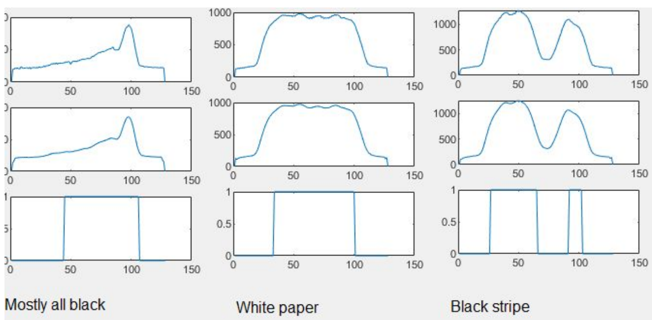

Fig. 6. Camera output: raw data (top row), filtered with a low pass filter output (middle row), and binary high low edge detection (bottom row).

The camera data (raw) was sent back to the K64 via the wires that were connected to the motor shield which routed the connector to Vss power and Darlington transistor switches [3]. This would let the K64 see in real-time where the car was going and what the NXP car's camera was seeing. With the filtered camera data (low pass/5 point average), edge detection (with a difference filter) could be done. This was very important because the edge detection is what determined where the edge of the track was. And that is what was used to determine where and when the NXP car should turn.

One algorithm for getting peaks and determining what to do with them was implemented such that it would find the two peaks (left and right). The goal was to have the NXP car be equidistant between the edge of the track. This would mean that the peaks were equally spaced. There was a hardcoded calibrated center point. This would be compared to the peaks and the center point. If the calculated value was off of the hardcoded calibrated value, the NXP car would need to turn in order to center itself. To prevent very jerky steering back and forth, there were thresholds set so that after the center point was off a certain amount the NXP car would turn a little, and then the further that it was off of center, the more that the NXP car would turn to get itself centered. This algorithm could also detect turns. And that is what would make the NXP car slow down and prepare for turns and this was important for variable speed. Also, the Proportional Integral Derivative (PID) was implemented to provide smooth speed and turning. The PID helped prevent random or unexpected changes from happening which is how the NXP car was given a smooth look for all turning and acceleration/deceleration.

Variable speed was a very important part of this NXP car. As with all races, going fast is important. But the key is to be able to slow down in turns specifically so that the car would not slide off the track or go too fast and the K64/camera wouldn't be able to process the next turn quick enough. As discussed more in the motor discussion up next, the inside wheel (right wheel on a right turn) would be slowed down significantly, and the outside wheel would be slowed down a very small amount if it was a big turn. This allowed for variable speed on turns. Also on straightaways, there was logic implemented that would allow the car to accelerate to max speed.

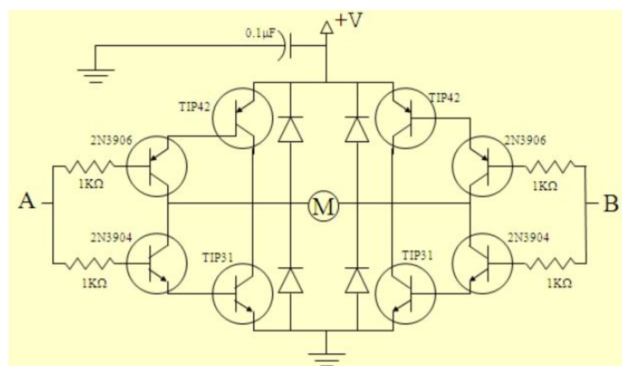

Fig. 7. DC-Motor Control H-Bridge [1]

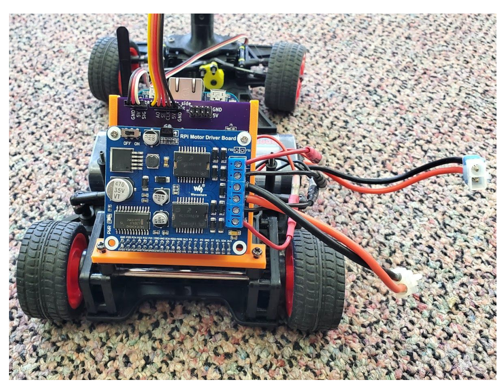

Fig. 8. Motor shield on the NXP car with the DC-motor wires on the top two and bottom two spots on the right, and the battery connection in the middle two spots on the right.

The motors used in the NXP car were one DC motor for each wheel and a servo motor for the front wheels' steering control. The DC motors were connected using the motor shield using an H-Bridge. There is a controllable PWM duty cycle to control the speed of the motor. Each wheel had its own motor to control it so the speed was set independently. This allowed for enhanced turning as well. The inside wheel on a turn would be slowed down significantly so that the outside wheel would push around and this improved turning functionality greatly. This slowed the car down since one wheel was pushing less, but it also acted like a little extra steering through the back wheels. The H-Bridge is necessary as well because it is what amplifies the K64 signal into something that the motors can use. Also, the battery is important for the H-bridge. The battery also powers everything on the NXP car. The servo motor is used to turn the front wheels left or right to control the steering. In software, the max for left and right was defined so that the servo motor would not try to turn further than it could physically turn with wheels and the car body in the way.

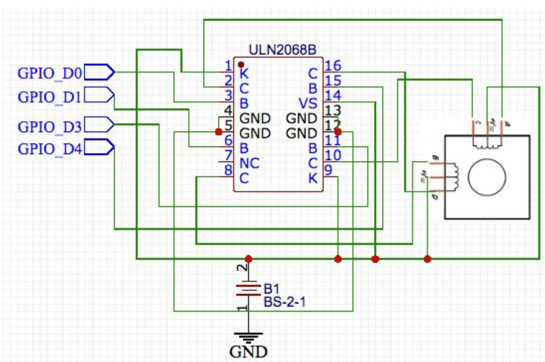

Fig. 9. Stepper motor

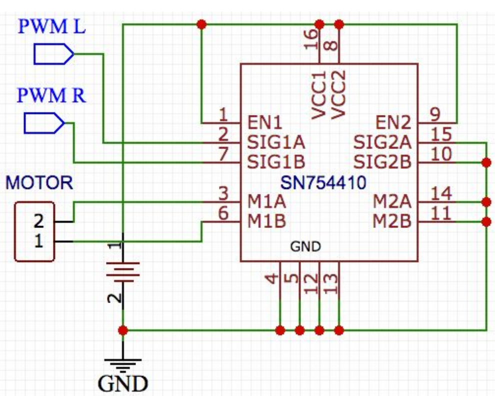

Fig. 10. DC-motor

Calibration and testing was a very instrumental part in creating a successful NXP car. Fine-tuning all of the hardcoded values in the code was very important to get right. The max left and right, as well as the straight direction for the wheels, was tested. Also, the center value of the car needed to be tested thoroughly in order to ensure that the algorithms and steering would work properly. The car taking averages for camera values worked very well to help with calibration as well for finding the center because it would remove the impact that outliers or bad data would have on the calculated center point. Testing all of the code frequently was important to do to make sure that there were no negative effects of any code changes. Also to ensure that hardware was still functioning as expected since the camera was sensitive and could be bumped and taken out of focus. Thorough testing was done to determine what speed reductions should be made for a turn and what acceleration should be done on a straightaway. Also determining what defined a straightaway was important to test. If the straightway code was triggered too quickly the car could boost into a turn and go off the track, and if the straightaway code took too long to trigger, then the car would never speed up and the race time would be negatively impacted.

4. RESULTS

With the final calibrated version of the NXP car ready, the PID and steering and all other algorithms were done. As well as hardcoded values being set and no longer being tuned or changed. After a lot of on-track testing with various turns and track setups, the values for the code were adequately tested and the NXP car was in good working condition for a race.

There were a few parts that were more difficult to get tuned and set in a satisfactory way. One of those values that were hard to tune was the center distance of the car so that the entire turning algorithm could work properly. Also, the PID controller took a lot of work to get tuned properly in a way that would successfully smooth the car and help remove outliers and bad data without negatively affecting the performance of the NXP car. For a while the simple if statement based hard left and right was more effective than the PID at first because the jerky movements worked better than a poorly tuned PID controller. A properly tuned algorithm with a PID controller made the NXP car have significantly faster times around the track.

One big challenge at the beginning was finding a properly set camera angle. This angle of the camera is what would have all of the edge detection algorithms built around so it was important to get it right. The camera needed to be able to see far enough ahead to slow down for turns and see enough of the track in front of the NXP car, but also be looking down enough to see the track directly in front of the car and not lose the track when going around turns. There was a balance in the angle of the camera that needed to be achieved. After the camera was set to a satisfactory angle, the screw was tightened down so it could not be moved.

The results of the race were surprising. Some cars that were expected to do well did not finish at all. The average time to complete the course was 25.02 seconds. The fastest time was 19.77 seconds, the second fastest was 20.93, and the third fastest was 21.55. The slowest time was 29.34 the second slowest time was 28.57 seconds, and the third slowest was 28.09 seconds. Nine cars did not finish. Eleven cars finished.

Fig. 11. NXP race day track being assembled.

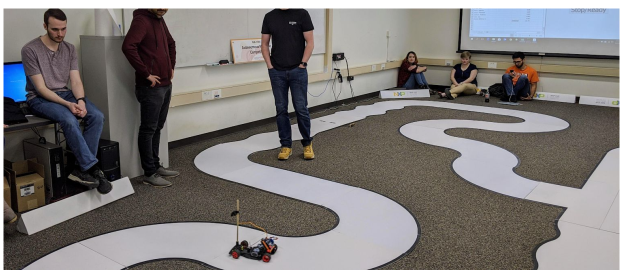

Fig. 12. The NXP car is going around a turn on race day.

To improve the NXP car in the future, giving out new cameras every time when receiving the kit would be good. Some cameras focus much better and have better raw data than others and that gives a large advantage to the team with the better cameras. Another thing to improve would be to have had multiple modes with a button push so that speed could be lowered on race day. Also, one thing that worked well on the NXP car was that the car drove very smoothly. The turns and speed changes were all very smooth and did not jerk the car around at all.

5. CONCLUSION

The NXP car project provided a real-world learning experience. It provided experience with autonomous cars. The project provided some of the most basic skills needed for modern engineering. Designing, problem solving, planning and management were some of the most helpful skills achieved throughout the process. Working with components such as camera sensors and motor system control alongside microcontrollers served as a useful learning experience for future projects. Designing filters, working with hardware, creating software, working with motors, and a line scan camera were all done in this exercise and it was a success.

6. REFERENCES

[1] "What Is a PID Controller?" *RealPars*, 9 July 2019, realpars.com/pid-controller/.

[2] "Features of the FlexTimer Module." *Features of the FlexTimer Module*, Freescale Semiconductor, Inc, 2015, www.nxp.com/docs/en/application-note/AN5142.pdf..

[3] *Interface & Digital Electronics Laboratory Manual*, Rochester Institute of Technology Department of Computer Engineering, Rochester, NY, Fall 2019.

[4] *K64 Adapter Documentation*. Rochester Institute of TechnologyDepartment of Computer Engineering, Rochester, NY, Fall 2019.

[5] *K64 Sub-Family Reference Manual Rev 2*. Freescale, January 2014.

[6] *FRDM-K64F Freedom Module User's Guide rev 1*. NXP Semiconductors, August 2016.