Auto-Knight: The Autonomous Vehicle

Bruce Hardy (EE), Tyler Thompson (EE), Christian Theriot (CE), and Eduardo Linares (EE)

Dept. of Electrical and Computer Engineering University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Auto-Knight is a small-scale autonomous vehicle sponsored by the Networked Systems Labs here at the University of Central Florida. The vehicle is being created to aid the NSL research team in their goal of creating a vehicle to vehicle communication network for the emerging autonomous vehicle industry. This research is important to increase the reliability of these machines to keep the public safe by reducing automotive deaths. Our model aims to replicate real world vehicle mechanics to as to eventually be applied to a full-scale vehicle project from this laboratory. This vehicle will serve as a model that will be replicated in different environments for the purpose of cooperative autonomous driving and data transmission for networking and machine learning. This paper will discuss the methods and products used to achieve the specifications requested by our primary project sponsor.**

***Index Terms* — Arduino, Autonomous Vehicle, Image Processing, Jetson, Lane Following, LiDAR, Linux, Networked Systems Labs, Nvidia, OpenCV, Printed Circuit Board, PID Control, Serial Communication, Ultrasonic, UART, ZED.**

I. Introduction

Every year thousands of people die in preventable car accidents, and millions are injured. According to a study released by Intel in June of 2017, the self-driving vehicle market has the potential to become an 800-billion-dollar market by 2035 and a 7 trillion-dollar market by 2050[1]. Around the world, road crashes are the leading cause of death between the ages of 15-29. 1.3 million people are killed every year in car accidents, an average of 3,287 people every day, and 20 to 50 million are injured as a result. This grim description of the world's auto accidents shows a great need for increasingly automated cars capable of preventing crashes.

Currently many established companies in the auto-industry are engaged in a rat race to create a fully autonomous self-driving vehicle. Numerous partnerships have created between ride sharing companies, chip manufacturers, and auto-manufacturers to try to create strategic advantages over their competition. Any innovations with the field could bring prestige and funding to the designers.

The Auto-Knight team saw the opportunity to work with Dr. Yaser Fallah and the Networked Systems Laboratory at UCF to make strides in consumer safety by designing a small scale preliminary model of a self-driving vehicle that can communicate with other vehicles. This small-scale model with work as a proof of concept to help raise money to create a full-sized vehicle and also simultaneously learn the hardware and software necessary to create an autonomous vehicle.

II. Motivation And Description

Our motivation for this project is to increase public safety, aid the development of emerging technology and help to provide our University with quality research in this field. Our vehicle will serve as the beginning of a multistage project to eventually implement a platoon of fully autonomous vehicle with vehicle-to vehicle communication that can perform various scenarios

The primary objective was to use computer vision, LiDAR and other sensory inputs to design a useful vehicle for the University's researchers. We wanted our car to be able to lane follow, avoid collision through emergency stopping or possible evasive maneuvers and can be remotely controlled. When compared to similar projects at other universities. The Auto-knight will be able to move at faster speeds. Originally, the team wanted to implement mapping, localization, and LiDAR in our stage of the project, but time constraints made this impossible. The requirements set forth from the researchers through discussion meetings with our team were as follows:

1) Sufficient battery life for runs of up to twenty minutes using a maximum of two batteries.
2) A maximum speed exceeding 15 miles an hour.
3) A chassis size of 1/10-1/16 scale to accommodate for indoor operation and a rugged, durable design for outdoor operation.
4) Autonomous operation mode with lane following, and a remote PC operation mode through over a wireless network.
5) Sensory input from ultrasonic sensors for shorter distances and for rear detection.
6) A front facing stereo camera with 10-meter vision range.
7) Emergency stop function to prevent the vehicle from colliding with objects.

III. Project Design

To realize the outlined requirements, extensive research was done by our team to locate critical components, places they could be purchased, and their prices alongside comparable alternatives. The vehicle design was then divided into 4 categories: Physical, Electrical, Software, and Control.

A. Product Selection

To model accurate vehicle mechanics and to save time reconstructing one from scratch, many high-quality RC cars currently being sold on the market were examined. After narrowing the number of vehicles down to less than 5, the Traxxas Slash 4x4 Pro was the final decision. It is capable of exceeding 35 miles per hour, aesthetically pleasing, an active community forum that discusses vehicle adjustments/upgrades, and many customizable options offered by this Traxxas.

For control, our vehicle utilizes two processing units; the main CPU is NVIDIA's Jetson TX2. The TX2 utilizes a quad-core ARM CPU alongside an NVIDIA Pascal GPU with 256 CUDA cores. The quad core processor makes the TX2 extremely fast at performing multiple functions at once. Furthermore, the GPU provides significantly faster image processing speed than possible with a microcontroller. The TX2 also was provided with a development board containing GPIO pins, 8 GB's of RAM, HDMI port, 32 GB of storage, 4K max screen resolution, and a USB 3.0 port. The combined value of processing power and availability made it the obvious choice as a main processor. The secondary processing unit is a PCB consisting of two Atmega-328P. This PCB is used primarily to control the motor, steering, one ultrasonic sensor, and other peripherals that are not necessary to function, but are useful.

Power Specifications required the use of two separate batteries. One 7.2V Lithium Ion battery to power the Traxxas' car motors and a high capacity 50,000maH portable power bank to power our sensors, and computational components. The second battery must have several different outputs and the one purchased has 19V, 12V and 5V outputs. A USB hub was also purchased to add the ports necessary to process camera data, use a keyboard and mouse, and power other sensors that may be added in the future.

5 HC-SR04 ultrasonic sensors were purchased for the purpose of close range object detection. The sensors were advertised to work between 3 centimeters and 4 meters. 2 sensors were going to be used in the front of the vehicle and another 2 in the rear.

A Stereo Labs ZED Camera was purchased for image processing. The ZED was chosen due to its ability to sense depth using 2 lenses, 30 frames per second refresh rate, and comprehensive list of example code showing how to use the ZED camera. A Scanse Sweep LiDAR was also purchased due to its cost, documentation, supported libraries, and usefulness in mapping. As stated in the previous section, The LiDAR will unfortunately not be used in this stage of the project.

B. Platform and Mount AutoCAD Design

To secure all the components outlined in the previous section to our vehicle, custom mounts were designed using AutoCAD and 3D AutoCAD software. The intended goals of the mounts are to provide stability to each of the major components, provide enough room for each component to be accessible, have enough to add more mounts for future upgrades. and avoid a top heavy or mechanically unstable design. Multiple iterations of each mount were designed and manufactured using 3D printing or by laser cutting a large slab of material with the University's Innovation Lab equipment.

The final design utilizes 6 mounts designed with the purpose of providing either a base for all other mounts to attach to or to stabilize the LiDAR, Stereo Camera, USB Hub, a rear Wi-fi antenna, and battery bank. The main platform and fin are constructed from plexiglass and were laser cut using University provided lab equipment. The platform has five holes in total. The three oval holes are for the passing of cables between our system and add easy access and allow shorter cables to be used and reduce clutter. The other two square holes in the back and cuts in the front allow for the spring system of the vehicle shocks to have ample room during full compression while not reducing the full range of steering of the vehicle. The platform mounts to the top of the suspension mounts using existing screws and holes to reduce overall impact on the structure and make it easier to disassemble. The platform can be seen in Figure 1.

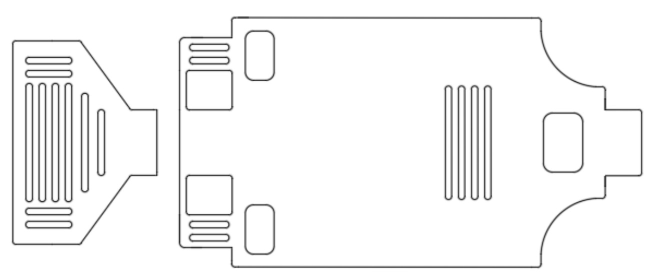

Fig. 1. Rear Fin – Left, Platform – Right

The LiDAR mount was designed to have the sensor mount to the rear platform and to top of the NVIDIA fan, without blocking it, to minimize vibration and allow for more consistent measurements. This mount was revised several times to allow wires to pass more easily and to prevent interference with the fan. The USB Hub had to be mounted to not only fit underneath the plastic covering of the car to but to reduce the arching of our component's USB cables. These mounts can be observed in Figure 2 below.

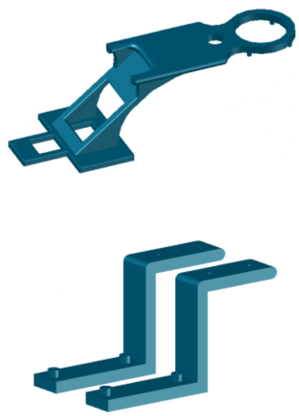

Fig. 2. LiDAR Mount – Left, USB Mount- Right

The stereo camera mount was made to fit in a manufactured hole in the front axle of the vehicle. With some small modification to the front bumper, the stereo camera and its mount fit perfectly in the front of the car. The mount being attached to the front axle that is connected to the shock system reduces the vibration of the camera which will reduce errors during image processing. To mount our Wi-Fi antennas and still be able to use the protective outer body, another mount was designed to attach to the rear bumper of the vehicle. The mount also features an engraving of the University's Pegasus logo. These mounts are depicted in Figure 3.

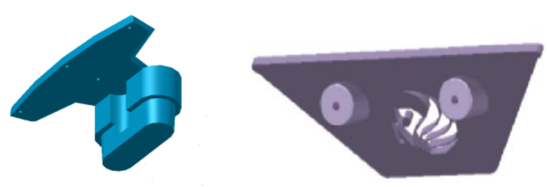

Fig. 3 Camera Mount – Left, Antenna Mount – Right

IV. Testing and Electrical Design

The electrical design for the project consisted of Servo/Electronic Speed Controller/motor testing, Tachometer testing, custom circuit board design, and custom wiring and soldering (USB, Dupont, and power connectors). Using the results of these tests, the commands to drive and steer the vehicle alongside the accuracy of speed measurement could be determined.

A. Determining Vehicle Dynamics

Despite a thriving online community, The Traxxas car came with minimal documentation for motor control. Traxxas' engineering department was also not willing to release details about what signals control the vehicle. As a result, a series of tests was created to determine how to control the vehicle. Using Arduino's servo library, the signals for moving our steering servo were determined to be based on the pulse width of a 5 V rectangle wave (Shown in Figure 4). To determine the maximum and minimum steering angles, a function generator was attached to the servo pins. A ruler was attached to the front wheel of the vehicle and the duty cycle of the produced wave was increased by 0.01%. In between increments, A photo was taken of the wheels position from above and using geometry from the photo the actual steering angle was calculated. A maximum range of 37 degrees to either side of the wheel was determined to be the range.

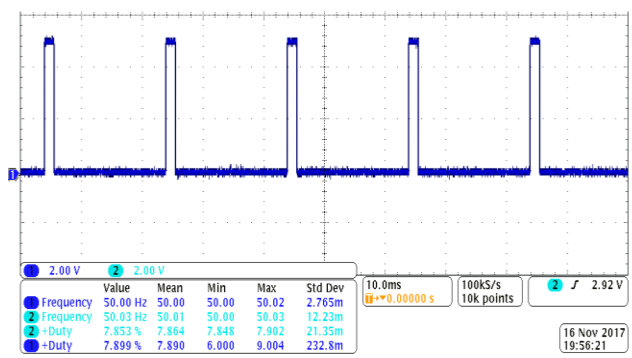

Fig. 4 Arduino Servo function pulse oscilloscope output

The ESC was programmed using Arduino's Servo library and the instructions shown in Traxxas' online instruction manual. The range of servo commands and the resulting motor reactions is shown below in Table 1.

Table 1. Motor Commands

Servo Command Values	Motor Direction
1000-1550	Backward
1551-1649	Neutral
1650-2000	Forward

Testing also determined that to implement a braking scenario, the command to the motor would need to switch from one direction to the other without ever resetting to the neutral range e.g. sending a command of 1650 and then sending 1550 later. The vehicles direction can only be changed from neutral. A flowchart of this is shown below in Figure 5. Braking becomes progressively stronger as you approach the opposite end of the range e.g. sending a command of 1650 and then sending 1000 implements a more powerful brake than sending 1500.

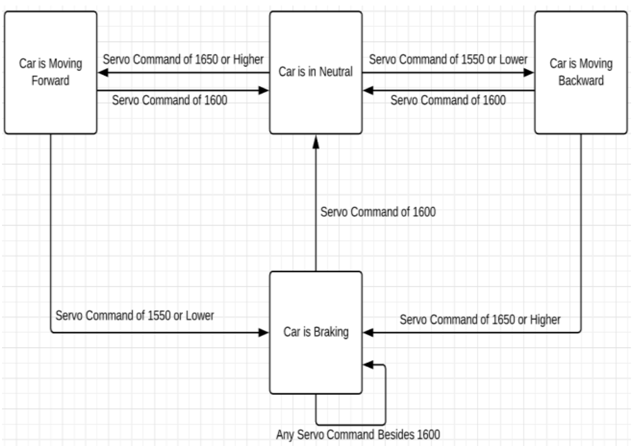

Fig. 5 State Flow Diagram

To aid in emergency stopping calculations, odometry data would need to be collected. A Hall sensor from Traxxas was purchased and installed in a slot near the spur gear of the vehicle and a magnet was attached to the spur gear. The sensor's 3 pins are then attached to a voltage source, ground, and one GPIO pin of the TX2. The sensor's VCC and signal are also connected via a 1000-ohm resistor. Whenever the spur gear passes in front of the sensor, the signal pin connects to ground and all the voltage that was previously on the pin is consumed by the resistor. These values are then filtered using software to calculate rotations of the spur gear. Using the gearing ratio of the center differentials in the back of vehicle, this value can be translated to a linear speed. A graph from motor speed testing can be seen in Figure 6.

Three dimensional models of the vehicle were also created to further aid our sponsor in accurate computer simulation of the vehicle for future research. With the added weight of all our components to improve vehicle dynamics were purchased and added springs to the shock system. The auxiliary battery added significant weight to the rear end. Using an additional spring in parallel to the manufacturer shock system spring corrected the error effectively.

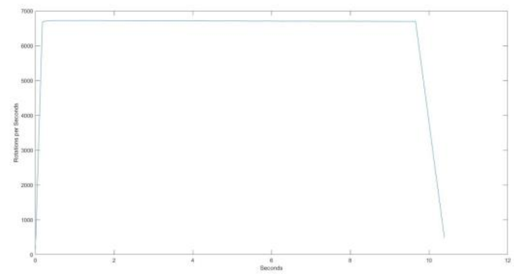

Fig. 6 Motor Speed Graph

B. Printed Circuit Board Design

There is a total of 3 boards that were designed for this project, two microcontroller-based designs and a logic-level converter board. The original MCU board used an Atmega2560 controller for data processing and peripheral control. This board has GPIO connected header pins to accommodate ultrasonic sensors, drive motor control, steering servo control, a horn to alert pedestrians in the hallways and outdoors during operation and testing, LCD display for data and error alerts, a low battery level detector to alert when charging is required, temperature sensor input and a fan for temperature management, LED headlights and taillights, USB serial connection for data transfer, over-current fuse protection, and auxiliary connections for back up transmit/receive lines on the MCU, USB programming and power. The board design for this iteration can be seen below in Figure 7.

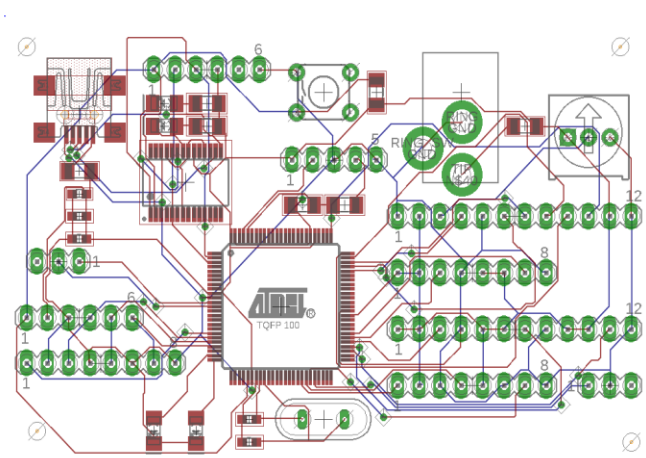

Fig. 7. Atmega2560 based PCB board layout design

Due to time constraints and challenges boot loading the MCU, an additional board was designed for easier programming to ensure the project could work before the demonstration. This board still accomplishes LCD and LED display, motor control, servo control, horn, battery monitor and USB serial communication. Other features were removed so the board remained compact despite the use of 2 Atmega-328P integrated circuits. The new PCB does include headers to create the tachometer circuit and preserve space. The board layout for this design can be observed in Figure 8.

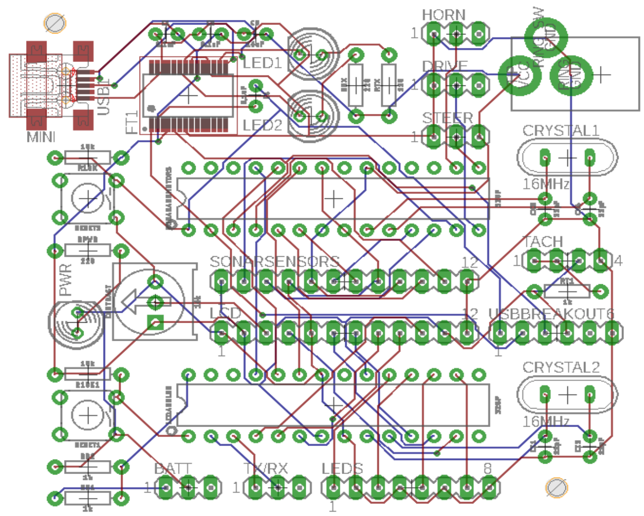

Fig. 8. Atmega328P based PCB board layout design

To decrease the latency in readings from our sonar sensors, we connected most them to the TX2, but the GPIO pins of the TX2 function read and write data at 3.3 volts and the ultrasonic sensors read and write with 5 volts. Furthermore, we need a 5-volt pin for each of the sensors VCC pins.

To eliminate both problems simultaneously, a 3rd PCB was designed to be used to convert the signals and work as a 5-volt power source. The board amplifies the trigger signal from the TX2 to enable sonar triggering using several non-inverting operational amplifier circuits. The echo signal is then reduced using a voltage divider. The board contains an overcurrent fuse, power LED, and 20 header 5V and 0V pins to supply sonar sensors and any additional future components with power. This design's board layout is shown in Figure 9.

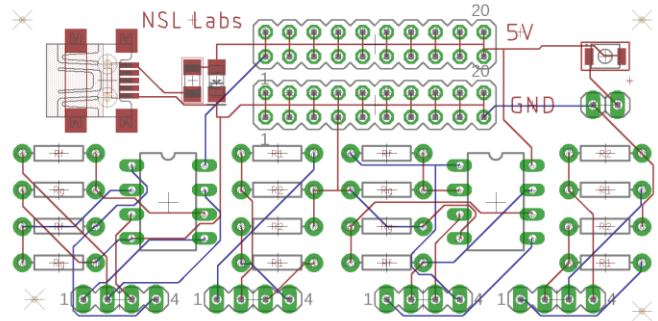

Fig. 9. Logic-Level converter board layout

V. System Design

The overall systems were each designed by a member of the team in effort to distribute tasks for a timely completion of the project. Software and customized functions were written by Christian Theriot. Image processing, object detection, and lane following algorithms were designed and written by Tyler Thompson. Motor and sensor calibration and testing, as well as PID control was handled by Eduardo Linares. Electrical design, soldering, customized cable and printed circuit board design was done by Bruce Hardy.

A. General Software Flow

Figure 9 depicts the simplified software flow of this project. The lane detection program provides input to the PID controller, which then provides an angle for the steering servo to orient the wheels toward detected lanes. A main program is run on the TX2 which handles the communication between the lane following algorithm, the motor controller, the collision avoidance algorithm, and the instrumentation nodes. The main program also determines whether teleoperation is engaged or not; if so, all commands from teleop will take precedence over the lane following algorithm.

The instrumentation nodes consist of a tachometer, a software odometer, and four ultrasonic sensors. The tachometer only reports a value when the spur gear is rotating; after a timeout of 2 seconds, it reports 0 rotations per minute (RPM). To keep a more accurate track of distance traveled, the software keeps track of an odometer within the tachometer object. Every time the tachometer records one rotation, it increases the odometer by the circumference of the wheel. Even if the wheels are rotating too slowly, the actual distance travelled will be correct. Each sonar runs in its own thread which checks the distance to objects in its surroundings at 10 Hz. The main thread has access to each sonar object's last recorded distance, and thus is thus able to make emergency stopping decisions at a rate of approximately 10 Hz.

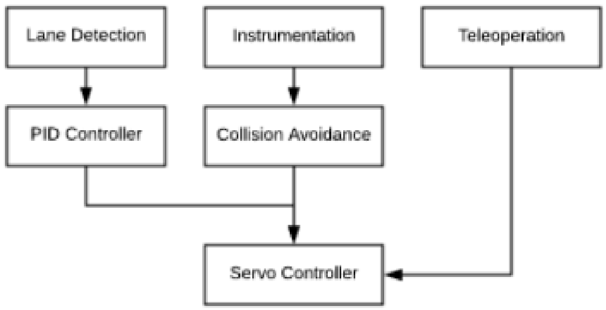

Fig. 9. Software Flow Diagram

The main form of inter-process communication has been the use of text files. The values for motor speeds and servo angles are stored in their own text files, which the main program reads and then sends to the motor controller via serial UART. Due to the limited throughput of serial UART, and the high frequency at which the motor controller receives messages, the use of a single byte as a message was used.

Each byte sent to the motor controller is either greater than-or-equal to zero, or negative. If positive or zero, the message is treated as a motor command and converted to the proper pulse width for the motor's ESC. If negative, the message is treated as a servo angle command and sent to the steering servo.

There is a fifth, isolated ultrasonic attached to the motor controller which is used for emergency collision avoidance. If this ultrasonic reads a distance below a threshold, it tells the motor controller to brake by setting the value of the servo command to 1000 and then ignores any commands telling it to throttle. The braking distance for this ultrasonic sensor was determined manually by having the vehicle brake at different commands and hard coding an acceptable stopping distance that could not be triggered too early due to sensor read errors or one that was too short that would lead to collisions. The stopping distances for each motor command are shown in Table 2. The HC-SR04 sensors purchased, while advertised to work at 4 meters, only worked for about 1 meter and were very inaccurate after 80 centimeters. As a result, the highest speed the sensors could work for is at the 1700 servo command. Upgrading the ultrasonic sensors should be one of the main priorities in the next stage of the project.

Table 2. Stopping Distances for Brake Function

Motor command	Stopping Distance (cm)
1650-1700	40
1700	60

When operating, it was noticed that the TX2 would sometimes power off due to an insufficient current draw from the battery, however the motor controller would remain functional and continue operating the last command. Due to this, an "emergency signal" was used to connect the TX2 to the motor controller via a GPIO-GPIO connection. When the main program is run, the TX2 sends a signal to the motor controller that it is ready to transmit data. The motor controller then only allows operation of the motor and steering servos when this pin is set high. When the pin is set low, either due to an operator sending the "emergency" signal via the main program, or the TX2 powering off, the motor controller then sets the motor to neutral and discontinues operations until the pin is set again.

B. Image Processing

The OpenCV computer vision library was utilized to handle live image processing using Python 3.5. Object detection and motion tracking were implemented as tests at the beginning of this project, but the main objective lies is to give the vehicle the ability to detect and localize itself with respect to a lane(s). While there exist several methods for advanced lane detection, the flowchart in Figure 10 delineates the procedure that has been implemented in this project. The program collects image pixel data, applies a perspective transform to achieve a bird's-eye view, creates a binary image with the desired color being the positive results, and finally analyzes the lane pixel data with respect to the lens' center. The deviation from that center is then sent to the PID controller.

To reduce the latency of the program, the image quality of the camera was lowered to increase the framerate from 60 frames per second to 100 frames per second. The program originally included features that allowed the camera to calculate radius of curvature as well as yaw rate, but the latency was still too high to accurately track the line without losing it.

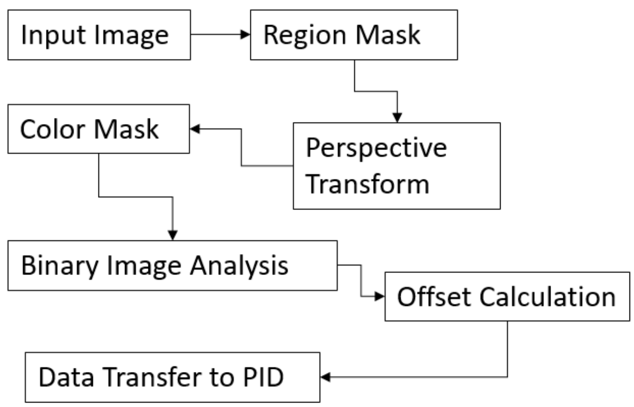

Fig. 10. Line follow program flow

C. PID And Control

For the vehicle to accurately follow lanes or lines detected by the camera, a PID controller was designed to control the steering angle of the vehicle. The goal is to reduce deviation from the line or lanes to a preset value defined in the code. Steering angle was chosen to be the controlled variable due to the simplicity of implementation and to reduce the latency between steering angle adjustments.

Since the PID controls only one of the variables necessary to follow the line or lane, traditional tuning methods like the Nicholas-Ziegler method do not work. Simulations were also somewhat cumbersome due to the need to re-run and re-tune PID values at different vehicle speeds and for different starting scenarios e.g. starting much farther away from the line. Instead, the PID was tuned manually finding an appropriate P value for each speed and then adjusting D and I to control overshoot and overall error as necessary.

The controllers P, I, and D values are based on the current speed of the vehicle due to the vehicle's yaw rate being controlled by steering angle, speed, and orientation of the vehicle at any instantaneous moment. Due to limitations in the test environment, only values for lower speeds were obtained.

D. Hardware Diagram

The hardware connections can be seen illustrated below in Figure 12. The TX2 uses the USB Hub as its data source for the Stereo Camera and PCB while powering the Ultrasonic sensors. The PCB receives data from the TX2 about the current direction for display purposes while utilizing a front facing ultrasonic sensor for emergency object detection. The LEDs connected to the PCB are also used as a visual indicator of the current state of the vehicle. Like most Li-Po batteries, the Traxxas battery retains its voltage level until it becomes critically low and needs to be charged again. Using an analog input pin and a voltage divider, the PCB will display a battery critically low display on the LCD when it senses a voltage drop of 0.5 volts. 0.5 volts was chosen because every cell of a Li-Po battery should never go below 3 volts and the battery contains 2 cells that run at 3.6 volts.

The PCB sends a byte of data to either the steering servo or the ESC/motor of the vehicle. The camera sends its data through the USB hub to the TX2 which is used for line processing images.

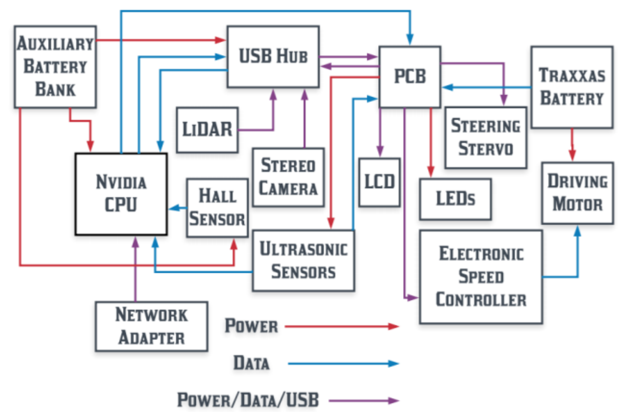

Fig. 12. Hardware connections diagram

VI. Conclusion

The Auto-Knight proved to be an ambitious undertaking for our team and unfortunately the original ambitious goals were not achieved. Despite this, the team has provided a very valuable base platform containing lane following algorithms, forward collision avoidance, PID control, a tachometer, and remote-control capability. Alongside these software implementations, the team also provided physical mounts, 2 functioning PCBs, and a manual explaining what signals drive the vehicle.

Future groups working on this project should focus on the following:

- Using neural networks to train the vehicle to follow lanes, recognize signs, etc.
- Implementing communication using an Ad-Hoc Network.
- Adding a GPS to aid in localization
- Implementing LiDAR
- Converting Python code to C++ for greater efficiency.
- Using a visual encoder instead of a hall sensor for the tachometer for faster updates.

- Upgrading ultrasonic sensors for better range and faster processing times.

VII. Acknowledgement

First and foremost, the Auto-Knight team would like to thank Dr. Fallah and his team of researchers, especially Behrad Toghi, for their guidance and confidence in us. It was a pleasure to work with the Networked Systems Lab and all their researchers. Next, we would like to thank our review committee for taking time out of their schedules to see our project. Lastly, the team would like to thank the students working in the Texas Instruments Labs for helping us cut and print all our parts quite a few times.

VIII. Biographies

Bruce Hardy is a a senior enrolled in the Electrical Engineering Bachelor's program at the University of Central Florida. Bruce has a passion for Power Systems and alternative energy expansion. With intern experience from SECO Energy, Bruce has accepted an Engineer I postion in Black and Veatch's Energy Department. Bruce has plans to further his education in seeking a Master program in either Power Systems or Engineering Management with no planned start date.

Christian Theriot is a a senior enrolled in the Computer Engineering Bachelor's program at the University of Central Florida. He has accepted a job offer from Lockheed Martin, who he has been working for since 2016 through UCF's CWEP program. His preferred programming language is C++.

Eduardo Linares is a a senior enrolled in the Electrical Engineering Bachelor's program at the University of Central Florida. He has a passion for microelectronics, controls, system design, and audio. Eduardo is currently applying for graduate programs in Industrial and Electrical Engineering and entry level positions so he can maximize his options post-graduation.

Tyler Thompson is a a senior enrolled in the Electrical Engineering Bachelor's program at the University of Central Florida. Currently a co-op employee at Orlando Utilities Commission, he is eagerly anticipating his official start date as a Electrical Distribution Engineer immediately following graduation.

IX. References

[1] Recode. (2017). *Ford's partnership with Lyft finally gives it a clear plan for self-driving cars*. [online] Available at: https://www.recode.net/2017/9/27/16374060/lyft-ford-self-driving-cars-partnership [Accessed 8 Oct. 2017].

[2] W. H. Cantrell, and W. A. Davis, "Amplitude modulator utilizing a high-Q class-E DC-DC converter," *2003 IEEE MTT-S Int. Microwave Symp. Dig.*, vol. 3, pp. 1721-1724, June2003