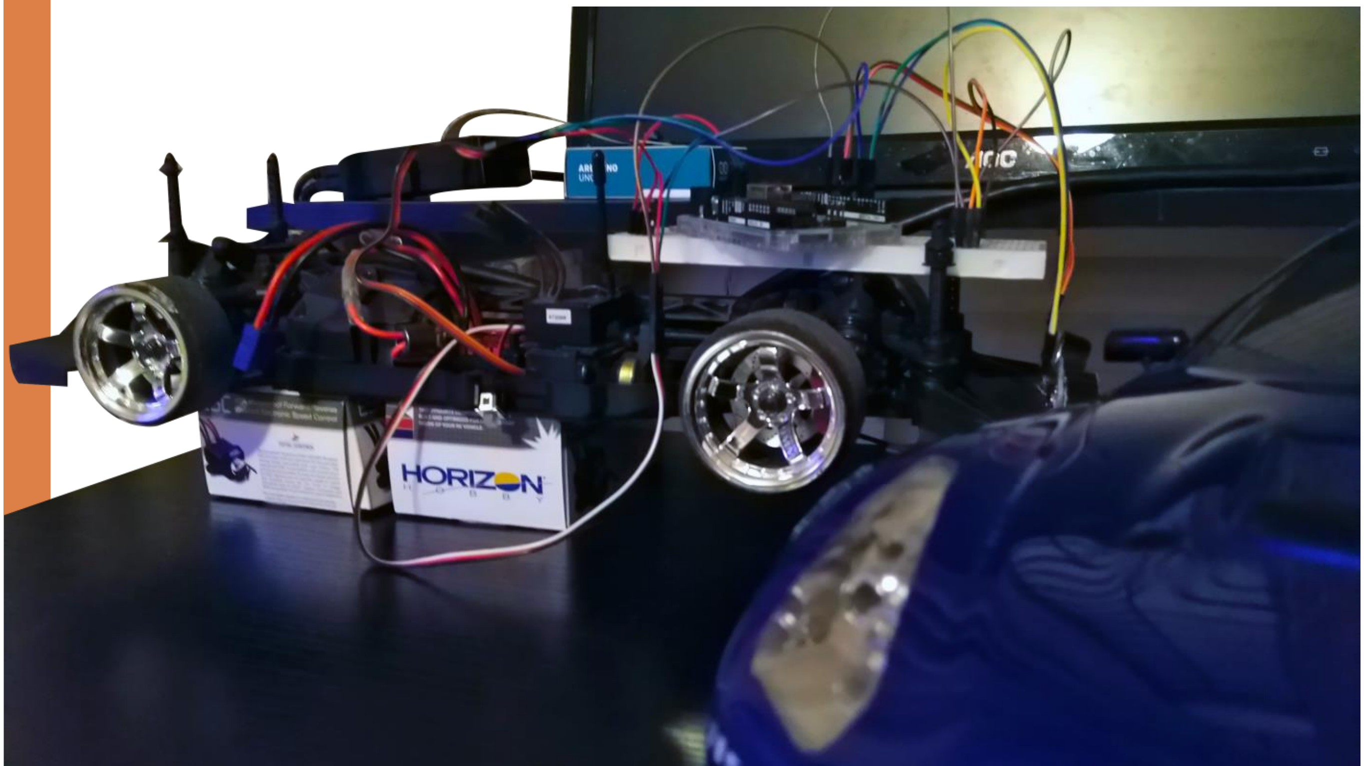

AUTONOMOUS RC VEHICLE

A LOOK INTO AUTONOMOUS VECHICLE TECHNOLOGY

CREATED BY: DINO CAJIC

CONTENTS

ABSTRACT

There is currently a race among the tech giants to develop the first fully autonomous vehicle. [1] This project explores autonomous features on a smaller scale by focusing on radio controlled (RC) vehicles. The Vaterra 2012 Nissan GTR Nismo GT3 V100-C RC car will be connected to both an Arduino Uno and a Raspberry Pi. The Arduino Uno R3 is connected to the Dynamite Waterproof 60A FWD/REV brushed ESC (Electronic Speed Controller), which in turn controls the speed of the Dynamite 540 brushed motor. For steering, the Arduino is also connected to the Spektrum RC S6170 standard digital surface steering servo. The Raspberry Pi 3 model B focuses on converting the Raspberry Pi Camera Module V2's input into steering instructions for the Arduino to execute; the Raspberry Pi will utilize the OpenCV 3.2.0 software for those calculations. Aside from servo controls, the Arduino Uno will contain five HC-SR04 ultrasonic sensors to help the vehicle avoid obstacles such as pedestrians and other vehicles.

SAFETY PRECAUTIONS

Numerous safety precautions will be implemented throughout the course of the project. When working with most electronics, a crucial safety precaution is to avoid the unnecessary transfer of built up static electricity. The RC vehicle, along with all the components that are connected to it, has been placed on a wooden table; the wooden table is stationed over hardwood floors. Prior to touching any parts, the antistatic wrist strap will be placed around the wrist and then connected to the metal portion of the table. All wires connected to the digital input/output, 5 volt and VCC pins of the Arduino and Raspberry Pi will be any color other than black; the black color is reserved for the ground connection. Datasheets will be referenced for each new component that's added to verify that an appropriate voltage is supplied to the component. The Arduino Uno, Raspberry Pi, and the RC vehicle will be disconnected from power before any work is performed on any part.

Safety precautions are not only in place for the safety of the electronics; they are in place for the safety of the person working on the electronics too. A ventilated, designated work area has been devoted for the course of the project. The soldering iron is only taken out when absolutely needed. The soldering iron stand is attached to the working surface so that the soldering iron cannot easily fall out. Upon soldering completion, the soldering iron is cooled down and is promptly stored away. A respirator and safety glasses will be worn while handling the soldering iron. The fire extinguisher is stored below the wooden table for easy access in case of emergencies. A smoke detector has been placed immediately above the working environment. Before leaving the work area, the power cord is removed from the outlet to prevent accidental overheating.

SIMILARITIES AND DIFFERENCES

Numerous organizations have succeeded in creating autonomous RC vehicles. In November of 2016, Adam Conway and William Roscoe from Make magazine attended the DIYRobocars hackathon, which was held by Chris Anderson, where attendees were introduced to the Donkey Self Racing Car platform. [2] The Donkey platform relies on training a neural network that will send commands to the servos. The user drives the RC vehicle around a track for 10 to 20 laps and then lets the autopilot take over. The training procedure is done on a PC while the autopilot commands are executed from a Raspberry Pi that's connected to the RC vehicle. [3]

The Self Driving RC Car that was developed by Zheng Wang uses a similar approach to the Donkey platform. It utilizes a Raspberry Pi, the OpenCV software and a neural network that help guide the RC vehicle along a track. The main difference is that the Raspberry Pi sends a video feed to a nearby PC via a WiFi module. The PC processes the feed and sends the servo commands back to the Raspberry Pi. [4]

Most autonomous RC vehicles will follow a similar approach, including the one proposed. A use of a microcontroller is necessary for servo control and camera input. The OpenCV software seems to be the standard video processing library for numerous applications and is especially useful for self-driving vehicles. Ultrasonic proximity sensors are also utilized in all three projects for collision avoidance. The main difference between this project and the other two mentioned is that the other two utilize a neural net where as this one does not. OpenCV software is used and calculations are performed on the Raspberry Pi itself to track the right shoulder-line on the road. Once the calculation is performed, the Raspberry Pi sends an integer value to the Arduino where the Arduino converts it into steering servo controls.

TIMELINE

The first phase of the project is to identify and understand the parts that make up the Vaterra RC car. A thorough understanding is needed of the ESC and servo motors that drive the vehicle. This phase was completed on February 9th, 2019. The ESC directly controls the speed of the motor. [5] The next phase involved connecting an Arduino to the Dynamite ESC and to the S6170 steering servo. [6] A correct connection was created on February 11th, 2019.

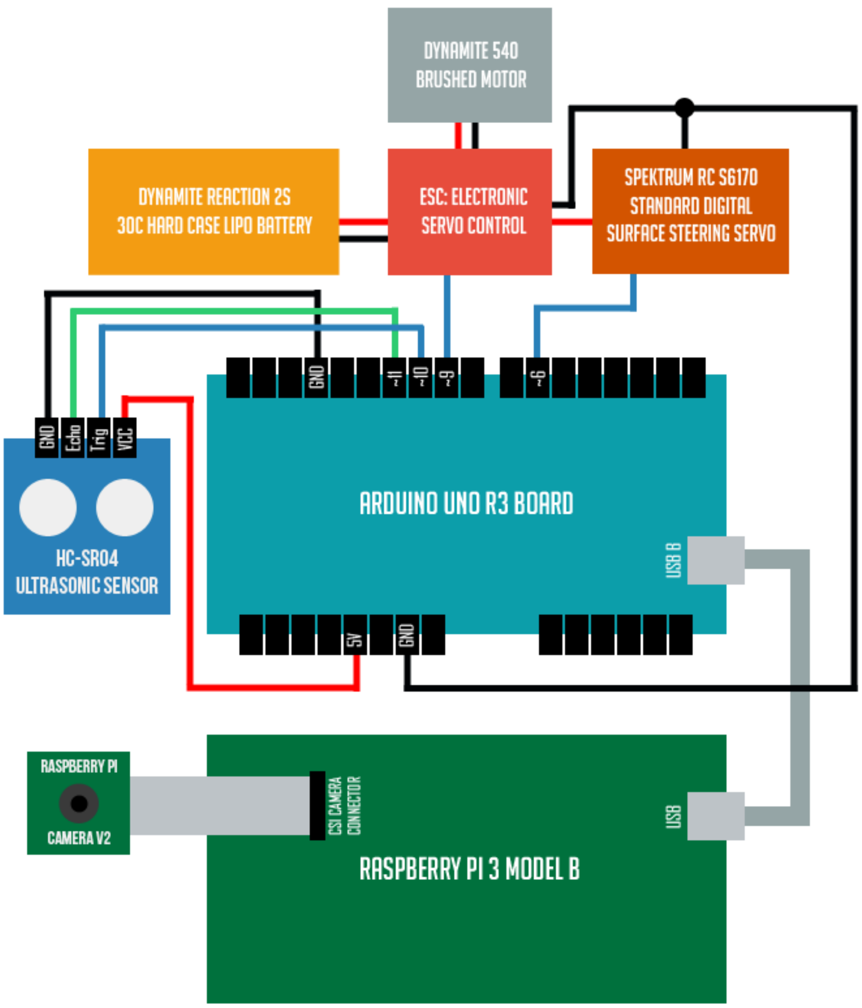

Figure 1 – The pinout showing the connection from the Arduino Uno R3 board to the Raspberry Pi 3, the HC-SR04 Ultrasonic Sensor, the Electronic Servo Control and the Spectrum Steering Servo. Diagram created by Dino Cajic.

Phase 3 involved connecting the Raspberry Pi to the Arduino. The Arduino program was installed on the Raspberry Pi and a test was created to send an integer value to the Arduino. [7] The installation consisted of opening the terminal on the Raspberry Pi and typing in the following command: sudo apt-get install arduino. [8] This phase was completed on February 12th, 2019. OpenCV installation was next on the list. To install the OpenCV software on the Raspberry Pi the following commands were executed on the terminal program running on the Raspberry Pi 3. The commands were specified by Sergio Canu. [9]

1. Make sure the Raspberry Pi's system is up to date
    a. sudo apt-get update
    b. sudo apt-get upgrade
2. Install OpenCV dependencies
    a. sudo apt-get install build-essential cmake pkg-config
    b. sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
    c. sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
    d. sudo apt-get install libxvidcore-dev libx264-dev
    e. sudo apt-get install libgtk2.0-dev libgtk-3-dev
    f. sudo apt-get install libatlas-base-dev gfortran
3. Install Python 3 and Pip3
    a. sudo apt-get install python3-dev
    b. sudo apt-get install python3-pip
4. Install the OpenCV software
    a. pip3 install opencv-python
5. Install other necessary OpenCV dependencies
    a. sudo apt-get install libqtgui4
    b. sudo modprobe bcm2835-v4l2

The installation was successfully completed on February 13th, 2019. The ultrasonic proximity sensor was attached to the Arduino board on February 13th, 2019 as well. The proximity sensor contains 4 pins: Ground, Echo, Trigger and VCC. The ground pin was attached to the ground pin on the Arduino. The Trigger and Echo pins were attached to the Arduino's digital input/output pins 10 and 11 respectively. Lastly, the VCC pin was connected to the 5V pin on the Arduino. [10]

Further research into autonomous vehicles is needed at this point to avoid any mistakes that were made by previous projects (February 22nd, 2019).

The next phase involves reviewing books on OpenCV. An understanding of how the OpenCV software works is necessary to be able to use it (March 2nd, 2019).

Phase 6 focuses on connecting the Pi Camera Module V2 to the Raspberry Pi, capturing a video feed with the camera, and extracting meaningful information from it. The idea is to remove all unnecessary components from the video feed and only leave the edges that make up the lines on the road. A small segment of the right line will then be analyzed, and a formula will be created to translate the movement of the line to the movement of the steering servos. This is probably the most complicated phase of this project. The goal is to have this phase completed by March 22nd, 2019.

Phase 7 will be dedicated to controlling the forward and reverse directions of the servo. Since the ESC only controls the speed of the motor, a special servo shield will need to be installed that can swap the positive and negative lines of the servo motor. Phase 7 will be completed by March April 1st, 2019.

The remaining 4 ultrasonic proximity sensors will be installed, and interrupts will be programmed so that the RC vehicle can avoid obstacles (April 6th, 2019).

A review of scholarly articles and books on autonomous vehicles will help solidify any lose ends that might have occurred by the end of Phase 8. Phase 9 will be completed by April 14th, 2019.

The last phase involves testing the autonomous RC vehicle. A track will be constructed, and the RC vehicle will navigate it. Spontaneous obstacle avoidance will also be tested (April 21st, 2019).

MILESTONES

Milestone 1: Connect the Arduino Uno R3 to the Dynamite ESC and to the S6170 steering servo.

Milestone 2: Connect the Arduino Uno R3 to the Raspberry Pi model B and send a test signal from the Raspberry Pi to the Arduino Uno.

Milestone 3: Connect the Raspberry Pi Camera Module V2 to the Raspberry Pi and convert the video feed to servo instructions. Connect one HC-SR04 ultrasonic sensor and test it.

Milestone 4: Connect the servo shield to the Arduino Uno so that it can switch the Dynamite 540 brushed motor from forward to reverse and vice-versa.

Milestone 5: Install 4 more HC-SR04 ultrasonic sensors and program interrupts so that the RC vehicle can avoid obstacles.

Milestone 6: Create an obstacle course for the RC vehicle and test the vehicle on it.

REFERENCES

[1] "46 Corporations Working On Autonomous Vehicles," 4 September 2018. [Online]. Available: https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/. [Accessed 1 February 2019].

[2] A. Conway and W. Roscoe, "Build an Autonomous R/C Car with Raspberry Pi," 10 August 2017. [Online]. Available: https://makezine.com/projects/build-autonomous-rc-car-raspberry-pi/. [Accessed 10 February 2019].

[3] "Train an autopilot with Keras," [Online]. Available: http://docs.donkeycar.com/guide/train_autopilot/. [Accessed 10 February 2019].

[4] Z. Wang, "Self Driving RC Car," 2018. [Online]. Available: https://zhengludwig.wordpress.com/projects/self-driving-rc-car/. [Accessed 9 February 2019].

[5] "Introduction To Electronic Speed Control (ESC) Working and Applications," [Online]. Available: https://www.elprocus.com/electronic-speed-control-esc-working-applications/. [Accessed 7 February 2019].

[6] "https://www.instructables.com/id/ESC-Programming-on-Arduino-Hobbyking-ESC/," 2014. [Online]. Available: https://www.instructables.com/id/ESC-Programming-on-Arduino-Hobbyking-ESC/. [Accessed 10 February 2019].

[7] M. Seif, "Send data from Python to Arduino through serial port," 2 June 2017. [Online]. Available: https://raspberrypi.stackexchange.com/questions/67840/send-data-from-python-to-arduino-through-serial-port. [Accessed 11 February 2019].

[8] "Program and Arduino Uno with your Raspberry Pi," 2018. [Online]. Available: https://www.raspberrypi.org/magpi/program-arduino-uno-raspberry-pi/. [Accessed 8 February 2019].

[9] S. Canu, "Raspberry Pi 3 and OpenCV 3 Installation Tutorial," 31 October 2018. [Online]. Available: https://pysource.com/2018/10/31/raspberry-pi-3-and-opencv-3-installation-tutorial/. [Accessed 13 February 2019].

[10] D. Nedelkovski, "Ultrasonic Sensor HC-SR04 and Arduino Tutorial," 2015. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/. [Accessed 10 February 2019].

[11] V. Vantsevich and M. Blundell, Advanced Autonomous Vehicle Design for Severe Environments, Coventry: IOS Press BV, 2014.

[12] A. M. Lopez, A. Imiya, T. Pajdla and J. M. Alvarez, Computer Vision in Vehicle Technology: Land, Sea, and Air, West Sussex: John Wiley & Sons, 2017.

[13] Zulkarnain, Z. Hairi, A. M. H. M. and H. Umar Zakir Abdul, "Path Tracking on Autonomous Vehicle for Severe Maneuvre," *Telkomnika,* vol. 16, no. 4, pp. 1583-1589, August 2018.

[14] S. Noh, "Decision-Making Framework for Autonomous Driving at Road Intersections: Safeguarding Against Collision, Overly Conservative Behavior, and Violation Vehicles," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 4, pp. 3275-3286, April 2019.

[15] Y. Chaochun, L. Hui, S. Jie, C. Long and J. Haobin, "Design and Analysis of an Autobraking System Controller for Autonomous Vehicles Under the Influence of Perturbation," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 3, pp. 1923-1931, March 2018.

[16] A. Millard-Ball, "Pedestrians, Autonomous Vehicles, and Cities," *Journal of Planning Education & Research,* vol. 38, no. 1, pp. 6-12, 2018.

[17] S. Bok-Suk, "Vision-based navigation of an unmanned surface vehicle with object detection and tracking abilities," *Machine Vision & Applications,* vol. 29, no. 1, pp. 95-112, January 2018.

[18] S. Weijing, A. Mohamed Baker, L. Xin and Y. Huafeng, "Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey," *In Integration, the VLSI Journal,* pp. 148-156, September 2017.

[19] M. Guo, X. Zhang, D. Li, T. Zhang and L. An, "Traffic light detection and recognition for autonomous vehicles," *China Universities of Posts and Telecommunications,* vol. 22, no. 1, pp. 50-56, February 2015.

[20] F. Wael, "Recognition of traffic signs by convolutional neural nets for self-driving vehicles," *International Journal of Knowledge Based Intelligent Engineering Systems,* vol. 22, no. 3, pp. 205-214, 2018.

[21] X. Wang, J. Wen, Z. Nan, J. Shi, L. Xu and Z. Nanning, "A general and practical path planning framework for autonomous vehicles," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),* pp. 1397-1402, 2016.

[22] C. Ligia and R. J. F. Rosaldo, "Multivariate modelling for autonomous vehicles: Research trends in perspective," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),* pp. 83-87, 2016.

[23] A. J. Hamlet, P. Emami and C. D. Crane, "The cognitive driving framework: Joint inference for collision prediction and avoidance in autonomous vehicles," *2015 15th International Conference on Control, Automation & Systems (ICCAS),* pp. 1714-1719, 2015.

[24] J. Funke, M. Brown, S. M. Erlien and C. J. Gerdes, "Prioritizing collision avoidance and vehicle stabilization for autonomous vehicles," *2015 IEEE Intelligent Vehicles Symposium (IV),* pp. 1134-1139, 2015.

[25] M. Y. Abualhoul, P. Merdrignac, O. Shagdar and F. Nashashibi, "Study and evaluation of laser-based perception and light communication for a platoon of autonomous vehicles," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC),* pp. 1798-1804, 2016.

[26] K. Dawson-Howe, A Practical Introduction to Computer Vision with OpenCV, West Sussex: John Wiley & Sons, 2014.