Jetson-Enabled Autonomous Vehicle

Matthew Haddad, Shem Cheng, Linda Thao, Justin Santos
Faculty Advisor: Dr. Hao Jiang, Graduate Advisor: Aaron Schlichting
School of Engineering, San Francisco State University, San Francisco, CA 94132, USA

MOTIVATION

- Learn how to use the Jetson TX2 with Robot Operating System to build a self-navigating autonomous robot.
- Gain experience working on the Jetson TX2 embedded development board

OBJECTIVE

- Build a vehicle that can navigate autonomously with the help of computer vision and sonar sensors
- Learn about modern robotics development in Linux through use of the Robot Operating System (ROS)

JETSON TX2

- NVIDIA's Jetson is the ideal solution for compute-intensive embedded applications such as image processing.
- The Jetson offers high-performance parallel processing power from onboard GPU, while consuming less than 10 watts of power.
- Supports vehicle components for mobility, such as Arduino, USB Camera, LiPo Battery Power.

	JETSON TX1	JETSON TX2
GPU	Maxwell	Pascal
CPU	64-bit A57 CPUs	64-bit Denver 2 and A57 CPUs
Memory	4 GB 64 bit LPDDR4 25.6 GB/s	8 GB 128 bit LPDDR4 58.4 GB/s
Storage	16 GB eMMC	32 GB eMMC
Wi-Fi/BT	802.11 2x2 ac/BT Ready	802.11 2x2 ac/BT Ready
Video Encode	2160p @ 30	2160p @ 60
Video Decode	2160p @ 60	2160p @ 60 12 bit support for H.265, VP9
Camera	1.4Gpix/s Up to 1.5Gbps per lane	1.4Gpix/s Up to 2.5Gbps per lane
Mechanical	50mm x 87mm 400-pin Compatible Board to Board Connector	

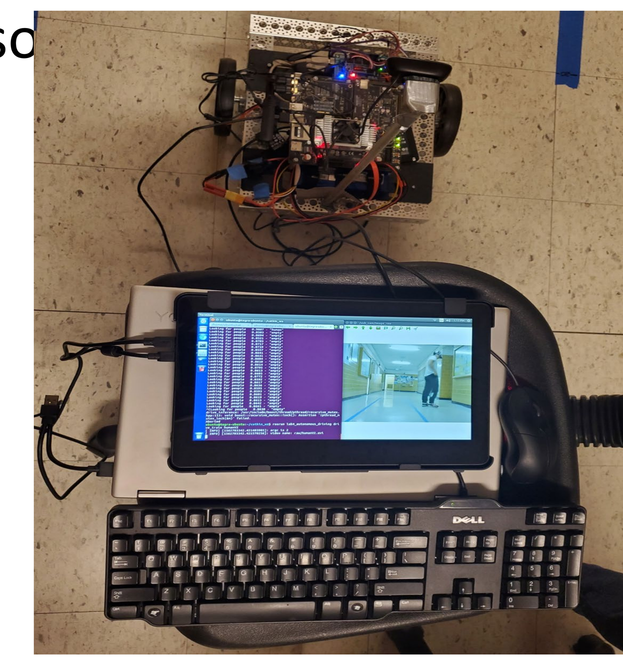
Fig 1. Jetson TX1 vs TX2 Specs

Robot Operating System (ROS)

- Robotics software framework for Ubuntu Linux that connects low-level device control with high-level programmed subroutines through a proprietary package manager and communication system
- Used to control vehicle components via messages sent to Arduino.
- Supports various packages for robot navigation and vision

COMPUTER VISION

- Worked with a computer engineering team to implement a deep learning algorithm based on the Caffe architecture to recognize humans
- Developed our own Computer Vision application based on the Tenso[illegible]OS_deep_learning node



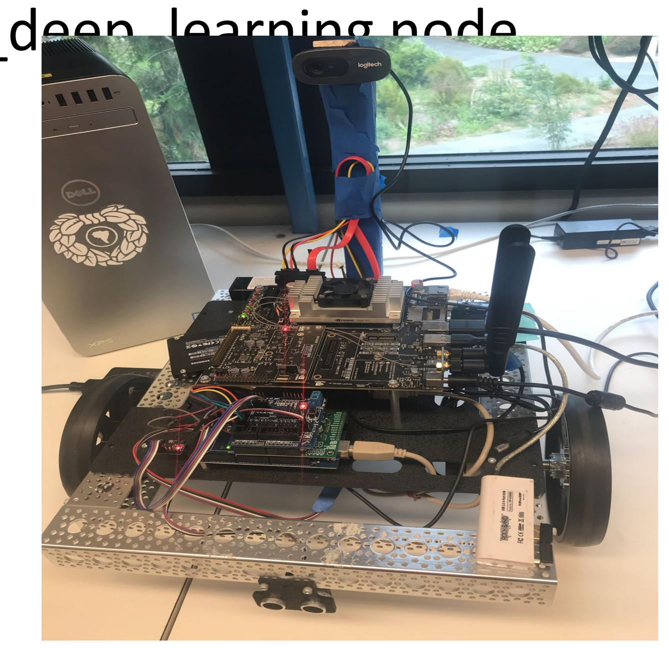


Fig 2. Jetson TX2 being tested with driving algorithms derived from TensorRT
FIg 3. Jetson TX1 configured for running computer vision scripts with Caffe

ARDUINO AND HARDWARE

- To control the motors, we use the Arduino Motor Shield Rev3, a driver module created specifically to allow efficient control of DC and Servo motors through the Arduino IDE.
- For sensing its environment, the robot used two kinds of sensors: HC-SR04 sonar sensor and a Logitech 720p camera

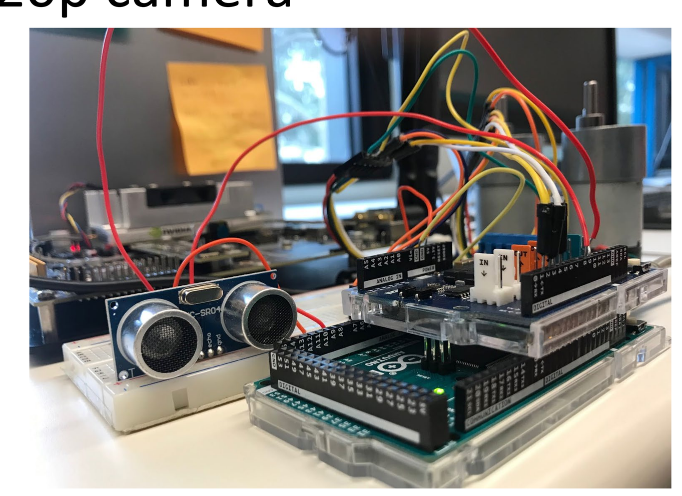
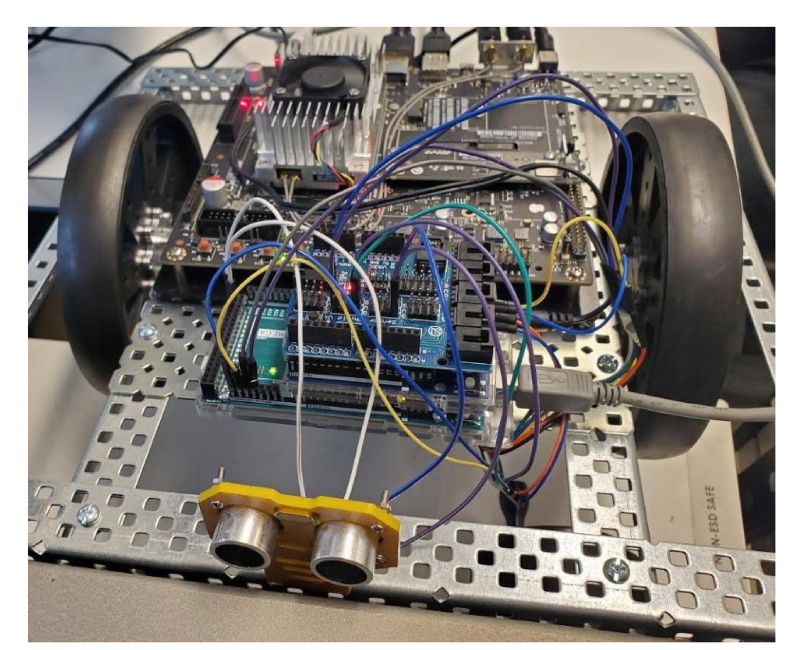


Fig 4. HC-SR04 Sonar Sensor (left)
Fig 5. Two Brushed DC motors and HC-SRO4 sonar sensor connected to Arduino Arduino motor shield Rev3 and Arduino Mega

RESULTS

- Successfully implemented sonar and computer vision for autonomous navigation
- Control the hardware of the robot with ROS nodes

```
void setup() {
  md.init();
  nh.getHardware()->setBaud(115200);

  nh.initNode();

  nh.subscribe(motor_right_speed_sub);
  nh.subscribe(motor_left_speed_sub);

  nh.advertise(motor_right_current_pub);
  nh.advertise(motor_left_current_pub);

  nh.advertise(encoder_left_pub);
  nh.advertise(encoder_right_pub);

  for(uint8_t i = 0; i < SONAR_NUM; i++) {
    nh.advertise(sonar_pub[i]);
  }
```



DIFFERENTIAL DRIVE

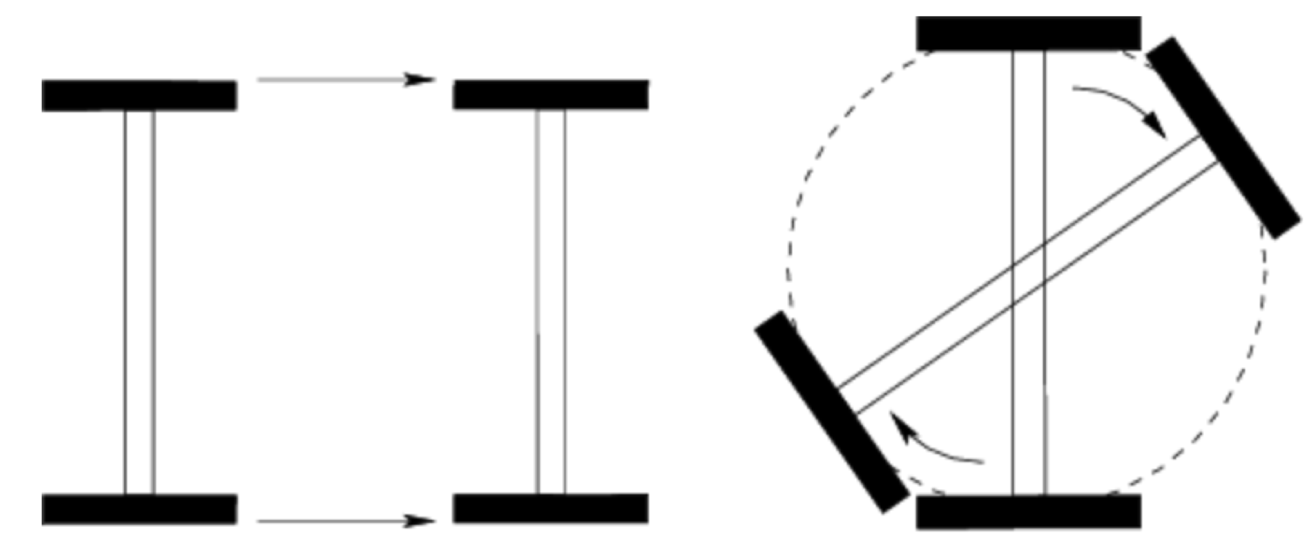


Fig 6. Pure translation (left) occurs when both wheels move at the same angular velocity; pure rotation (right) occurs when the wheels move at opposite velocities.

- ROS communicates data over rostopics by sending messages to different nodes
- For the robot to navigate it has to be sent messages on a topic called /cmd_vel which takes an input of the data type geometry_msgs/Twist.
- This message contains two vectors for linear and angular speeds, we took advantage of the linear.x and angular.z components to control linear motion and angular rotation
- The ROS package diff_drive_controller was used to convert the /cmd_vel message into voltage signals to be sent to the individual motors

CONCLUSION

- Built an autonomous vehicle that can change directions based on its sensor reading using the Jetson TX2.
- Implemented self-navigation algorithms based on computer vision and sonar
- Used ROS as a framework to to connect high-level instruction subroutines with low-level device control.

FUTURE WORK

- Implement deep learning, image recognition programs for autonomous behavior.
- Expand on movement and feedback routines to accomplish complex tasks.

ACKNOWLEDGEMENTS

This project is supported by the US Department of Education through the Minority Science and Engineering Improvement Program (MSEIP, Award No. P120A150014); and through the Hispanic-Serving Institution Science, Technology, Engineering, and Mathematics (HSI STEM) Program, Award No. P031C110159.