**AACC ScuttleBot Robotics Project**

**A Technical Report**

**KEY WORDS:** robotics, autonomous robots, computer vision

**Authors:** Terry Slattery, Caden Annadale, Christopher Rader

**Faculty Advisor:** Prof. Tim Callinan

# Abstract

Robots are in use across many lines of business and government. Autonomous robots perform a wide variety of tasks, such as inventory checking in grocery stores like the Tally robot in BJ’s Wholesale Club (Say Hello to Tally 3.0), household vacuum robots (A Roomba home is a cleaner home), and lawn mowing robots (Delaney). A team of students working with Prof. Tim Callinan have identified a low-cost, extensible robot that’s designed for use in the education environment. This report details the development of an initial robot and future plans for additional versions that incorporate additional technologies.

# Introduction

The field of robotics offers students invaluable hands-on experience in mechanics, electronics, and software development. As part of an undergraduate proof of concept project, a group of AACC students formed an informal robotics team to build one or more instances of an affordable and versatile wheeled mobile robots. We selected a robot design developed by Texas A&M University--the SCUTTLE robot (an acronym from the primary design criteria: Sensing, Connected, Utility, Transport Taxi for Level Environments) (Welcome to the SCUTTLE Community). A variety of factors influenced our selection:

* Low cost, enabling multiple robots to be developed
* Availability as a complete kit to minimize the time spent on mechanical assembly
* Simple and versatile dictated a wheeled robot design that could use a variety of sensors
* Existing software code base to facilitate implementation of the fundamental robotic functions
* Adoptability by multiple higher level educational institutions
* Active development community

The SCUTTLE is an inexpensive mobile wheel-based robot made from commonly available or 3D-printed parts. The parts list, software, and course curriculum is open source, and several universities worldwide have adopted it for their robotics programs. In addition, there is an active support community on Discord for sharing information about problems and solutions. A benefit of the SCUTTLE is that the chassis is versatile, sturdy, and light-weight, providing a platform for a variety of modifications and sensors. A machine with text on it

AI-generated content may be incorrect.

Figure 1 The AACC SCUTTLE Robot and Its Components

# Methods and Materials

In the 2023-2024 academic year, the team acquired a SCUTTLE kit from Texas A&M University and assembled it according to the online documentation. (Figure 1) The green and purple plastic parts are 3D printed and measures 18 inches wide by 16 inches long by 6 inches high with a load capacity of 40 kg. The main components, identified in Figure 1, are the following:

1. Battery to provide 12 volts. A secondary battery provides longer mobile runtimes or an AC mains power supply may be connected for long duration software development access.
2. Battery monitor allows programmatic access to battery voltage and current draw.
3. A 12v to 5v power converter to provide enough power to the Raspberry Pi to drive all the USB components.
4. Raspberry Pi 5 (RPi5) single board computer with 256GB SD main storage, 8GB RAM, HDMI display output, wired Ethernet, and Wi-Fi Ethernet.
5. A dual motor controller performs pulse-width modulation of power to each motor under program control.
6. An LCD panel displays system status information, like battery voltage and current draw.
7. A forward-looking color camera is connected to the RPi5 via USB.
8. A wireless gamepad controller provides manual control via a USB interface.
9. Small speaker to facilitate a text-to-speech human interface.
10. Three IR sensors front-mounted for down-looking tape sensing.
11. Motor position encoder on each motor to provide wheel rotation feedback.

The SCUTTLE web site guided our mechanical assembly and wiring efforts. Some components are connected to the RPi5 via USB and others, like the motor position encoders and tape sensors use an Inter-Integrated Circuit (I2C) bus (Inter-Integrated Circuit Bus) via the general purpose interface pins on the RPi5.

We loaded Linux and downloaded the SCUTTLE software onto the RPi5. The low-level motor test function ran the robot forward 4 seconds, stopped, then reversed for 4 seconds and stopped, demonstrating successful fundamental operation.

Next, we tested the Gamepad manual control module and found that it also worked. This program read values from the Gamepad controller’s joystick and used them to control the robot motors. This essentially turned our robot into a remote-controlled vehicle, with forward, reverse, turning, and stop control.

Now that we had the basic system running, we could focus on experimenting with autonomous control.

## Autonomous Functions

One of our most important goals was the implementation of autonomous functions that the robot could perform without user input. We selected two functions: line tracking and colored ball tracking.

### Line Tracking

The first software that we developed was for line tracking, using three IR sensors mounted on the front of the robot (Figure 2).

A machine with wires and cables

AI-generated content may be incorrect.

Figure 2 Line Tracking

A table of sensor states and resulting robot actions was developed (Table 1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **State ID** | **Left Sensor** | **Middle Sensor** | **Right Sensor** | **Action** |
| 0 | Off | Off | Off | Search for tape |
| 1 | Off | Off | On | Turn right |
| 2 | Off | On | Off | Go straight |
| 3 | Off | On | On | Slight right turn |
| 4 | On | Off | Off | Turn left |
| 5 | On | Off | On | Error – Search for tape |
| 6 | On | On | Off | Slight left turn |
| 7 | On | On | On | Stop – End of tape marker |

Software was written to handle each of the eight states (0-7) of the three sensors. Some states were controlling tape tracking while other states initiated a *tape search* mode where the robot stopped forward motion and rocked side-to-side to find the tape. We also made the decision that putting an additional piece of tape across the end of the run would tell the robot to stop, thus the action for State 7 where all three sensors are *On*.

Line tracking was not without its challenges. Varying floor reflectivity occasionally caused false positives or signal noise; calibrating sensor thresholds and mounting position was critical for reliable operation.

While testing the line tracking software we quickly identified the need to have a failsafe mechanism to take over the robot’s operation if it were to veer away from the tape. Our solution was to integrate the line tracking with the Gamepad control module, providing the robot operator full control. Line tracking is enabled by pressing and holding the Gamepad Right-Trigger (RT) button. Releasing the RT button stops line tracking and commands the robot to stop motion.

### Colored Ball Tracking

The SCUTTLE robot is outfitted with a forward-looking color camera that is connected to the RPi5 via USB. We loaded the OpenCV (Open Computer Vision) library (OpenCV) which provides numerous functions for image processing. The RPi5’s compute power makes it ideal for running OpenCV applications. Fortunately, the SCUTTLE robot software system included some modules that use OpenCV to track a colored ball, making it relatively easy to get started. The software grabs a camera image and filters it for the selected color, orange in our case. It then converts the image to black and white and identifies the largest object (actually a blob of pixels at this point).

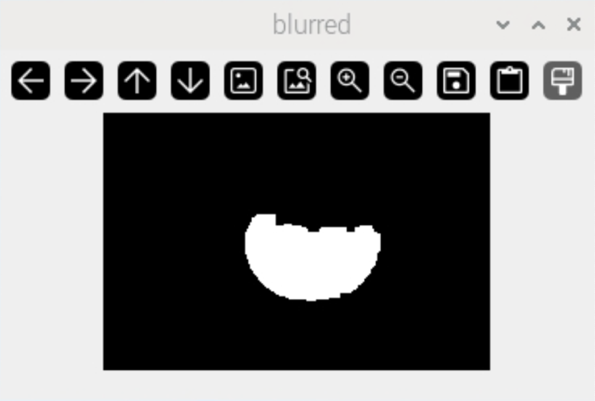


Figure 3 Blurred Ball Image

The blob image is rather jagged and contains holes where an individual pixel may have not matched the color threshold. Standard video processing practice is to run the image through blurring filters to smooth the edges (Figure 3). Finally, a centroid is identified that encloses the ball’s blurred image. This centroid can then be superimposed on the original image (Figure 4).

A close-up of a ball

AI-generated content may be incorrect.

Figure 4 Camera Image with Centroid Superimposed

The centroid is compared with the camera’s field of view and if it is not close to the center of the image (left-to-right), motor commands are given to turn the robot until it is centered. Then the size of the ball is compared with a range of pixel sizes to determine how close the robot is to the ball. The objective is to position the robot about 24 inches away from the ball. Therefore, if the ball’s centroid is too big, the robot is too close and motor commands are given to back away. Correspondingly, if the ball centroid is too small, the robot is too far away from the ball and the motors are driven to approach the ball.

The ball tracking system works very nicely but there are certainly challenges. The ball must be large enough to register in the image and not require that the robot get too close or too far away. A 9-inch diameter ball is about right. There are also challenges with ambient lighting that cause the ball’s color to fall outside the desired range. Reflected lights look more white while shadows look more dark. A reflective ball is more problematic than a matte surface ball. This imaging problem is apparent when comparing the camera image of Figure 4 with the blurred ball image in Figure 3. We also occasionally had problems with other objects in the background matching the color filter, such as wood doors that had an orange tint.

## Additional Functions

While developing the autonomous functions, we identified several additional functions that would be useful: battery monitoring and text-to-speech.

The SCUTTLE uses lithium-ion batteries that for longevity should not be discharged below 3 volts per cell, or 9 volts for the 3-cell battery system. To facilitate continuous monitoring, we added the battery monitor hardware module and software. The combined Gamepad and Line Tracking software was modified to check the battery status every few seconds and display it on the robot’s LCD panel. This addition convinced us that we needed to add additional battery capacity to extend the robot’s run-time.

A robot that just runs around on the floor is rather neat, but we felt that some human interaction would be fun, so we added a small USB speaker and integrated the Festival text-to-speech system from Carnegie Mellon University (Festvox). The battery monitoring software was augmented to trigger a voice message if the battery voltage gets too low and Gamepad button “B” was configured to allow the operator to request that battery status be spoken.

# Future Plans

We have just scratched the surface with what can be done with the SCUTTLE robot. The design is very flexible, allowing us to expand the number and type of sensors and to incorporate more advanced autonomous operation. The options range from adding hardware sensors to software-only enhancements that expand the use of the existing sensors. We outline these options below.

## Hardware

Our plans include adding additional sensors and building an additional robot. The additions are described below.

We will be adding an Inertial Measurement Unit (IMU) that will be used to improve autonomous navigation. It provides magnetic compass heading, velocity in X, Y, and Z directions, and acceleration in each axis, for a total of nine different navigational parameters. This sensor will allow for directional sensing and to provide feedback for executing precise turning angles.

We have a LIDAR sensor (LIDAR) that uses a low-power eye-safe laser for two-dimensional, 360-degree object sensing. This will be the primary sensor for object sensing, object avoidance, and environment mapping. By creating a map of the environment, we plan to teach the robot how to avoid running into people and objects as well as avoiding stairs and elevators. The goal would be to safely map and navigate a large area like the third-floor hallway of AACC’s CALT building. If we find that the LIDAR operation needs more information for accurate object identification, we will add ultrasonic sensors.

The human interface would benefit from the addition of a microphone to allow verbal communication with the robot’s control system. This addition goes along with the existing speaker and might eventually allow for replacement of the Gamepad for override control.

We plan to install a graphical display panel on the pedestal that supports the LIDAR as an additional way to communicate robot status to the operator or individuals in its path. There are many things that could be done with a small display, such as emulating eyes that track where the robot is headed next, or to display robot status information.

Adding more cameras would open the possibility for stereo computer vision required for depth perception. Side- and rear-facing cameras would enhance the ability to detect objects that are outside of the forward-facing camera’s field of view.

We plan to build two additional robots based on the same basic frame design but with changes to simplify the drive system and component mounting system. An additional robot provides the team with another platform so that multiple avenues of software development can proceed in parallel. This approach also provides a learning environment for new students who join the team.

## Software

The robot software is the critical component in processing data from sensors and taking the appropriate actions. Safety is critical, so we have made the Gamepad capable of overriding the robot’s autonomous operation.

We would like the ability to monitor and control the robot via a web-based system that only requires a Wi-Fi connection. The current computer interface relies on pre-defined network connectivity to a cloud-based GUI interface, which could be compromised by poor network connectivity. Ideally, the robot should be controllable from a phone or tablet or a computer. We could then investigate using Bluetooth for the control and status connection.

We plan to use the TensorFlow AI system with camera input to identify objects and track them or avoid obstacles. This system is capable of identifying over 100 different types of objects and we are very curious how well it would work in conjunction with the color ball tracking system.

We are also considering the investigation of the recently-released DeepSeek AI (DeepSeek) system running on the RPi5.

An alternative to on-board AI is to send images or sound clips to a cloud-based AI system for more advanced processing than can be handled by the RPi5. Of course, this depends on the reliability of the robot’s Wi-Fi connectivity.

Simultaneous Localization and Mapping (SLAM) is a methodology for avoiding hazards, mapping the environment to facilitate autonomous robot navigation. This software technology would rely on various sensors like LIDAR, ultrasonic, and camera.

# Concluding Remarks

Our SCUTTLE Robot initiative has accomplished the goals that we initially defined and has laid a solid foundation for undergraduate students to gain practical experience in advanced robotics at AACC. The students gained experience in wiring, soldering, programming in Python, working with microcontrollers (Raspberry Pi), mechanical design, 3D modeling, and 3D printing. The project facilitated teamwork among students from different STEM backgrounds (e.g., mechanical design, electrical engineering, computer science).

Starting with simple line-following, then progressing to open-source computer vision for autonomous ball tracking has taught valuable lessons in electronics, coding, and teamwork. Looking ahead, implementing LIDAR for mapping and AI algorithms for autonomous decision-making will not only elevate the robots’ capabilities but also significantly enhance the educational value of the project.

An added bonus was the opportunity to take the SCUTTLE robot to the Annapolis Senior High School recruiting and science fair, raising awareness and interest in robotics and STEM curriculum at AACC.

# References

*DeepSeek Artificial Intelligence System* (Accessed Feb 2025) DeepSeek – An open source AI model <https://www.deepseek.com/>

*Delaney, J. (2024, Nov 20) The Best Robot Lawn Mowers for 2025* PC Magazine <https://www.pcmag.com/picks/the-best-robot-lawn-mowers>

*Festvox – A Computer Speech Software System (Accessed Feb 2025)* Carnegie Mellon University computer speech group’s text-to-speech system <http://festvox.org/>

*Inter-Integrated Circuit Bus* (Accessed Feb 2025) I2C Bus history and description <https://en.wikipedia.org/wiki/I%C2%B2C>

*LIDAR - Light Detection and Ranging* (Accessed Feb 2025) LIDAR description, history, and uses <https://en.wikipedia.org/wiki/Lidar>

*OpenCV – An open source computer vision library* (Accessed Feb 2025) OpenCV Home Page for computer vision <https://opencv.org/>

*A Roomba home is a cleaner home* (Accessed Feb 2025) iRobot product page for the Roomba vacuum <https://www.irobot.com/en_US/roomba.html>

*Say Hello to Tally 3.0* (Accessed Feb 2025) Simbe Robotics Autonomous Inventory Robot <https://simberobotics.com/platform/tally>

*SLAM - Simultaneous Localization and Mapping* (Accessed Feb 2025) SLAM description and uses <https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping>

*Welcome to the SCUTTLE Community* (Accessed Feb 2025) Home page for the SCUTTLE robot <https://www.scuttlerobot.org/>