

## **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**

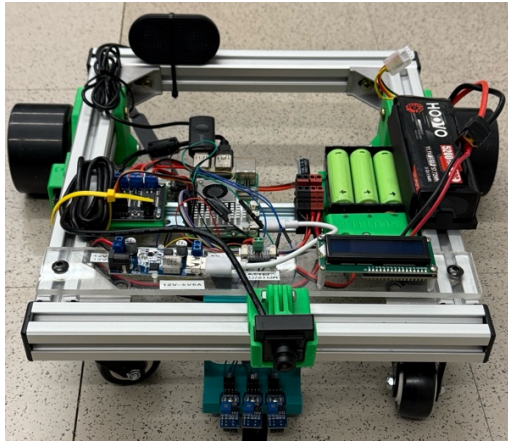
Robots are in use across many lines of business and government. Autonomous robots perform a wide variety of tasks, such as inventory checking in grocery stores like the Tally robot in BJ's Wholesale Club [1], household vacuum robots [2], and lawn mowing robots [3]. A team of students working with Prof. 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 affordable and versatile mobile robots. The selected platform is the SCUTTLE robot [4], developed by Texas A&M University. SCUTTLE is an acronym formed from the key characteristics: Sensing, Connected, Utility, Transport Taxi for Level Environments.

SCUTTLE is an inexpensive mobile wheel-based robot made from commonly available parts combined with 3D-printed parts. The parts list, software, and course curriculum is open source and several universities worldwide have adopted it for their robotics programs. There is an active support community on Discord. The chassis is versatile,

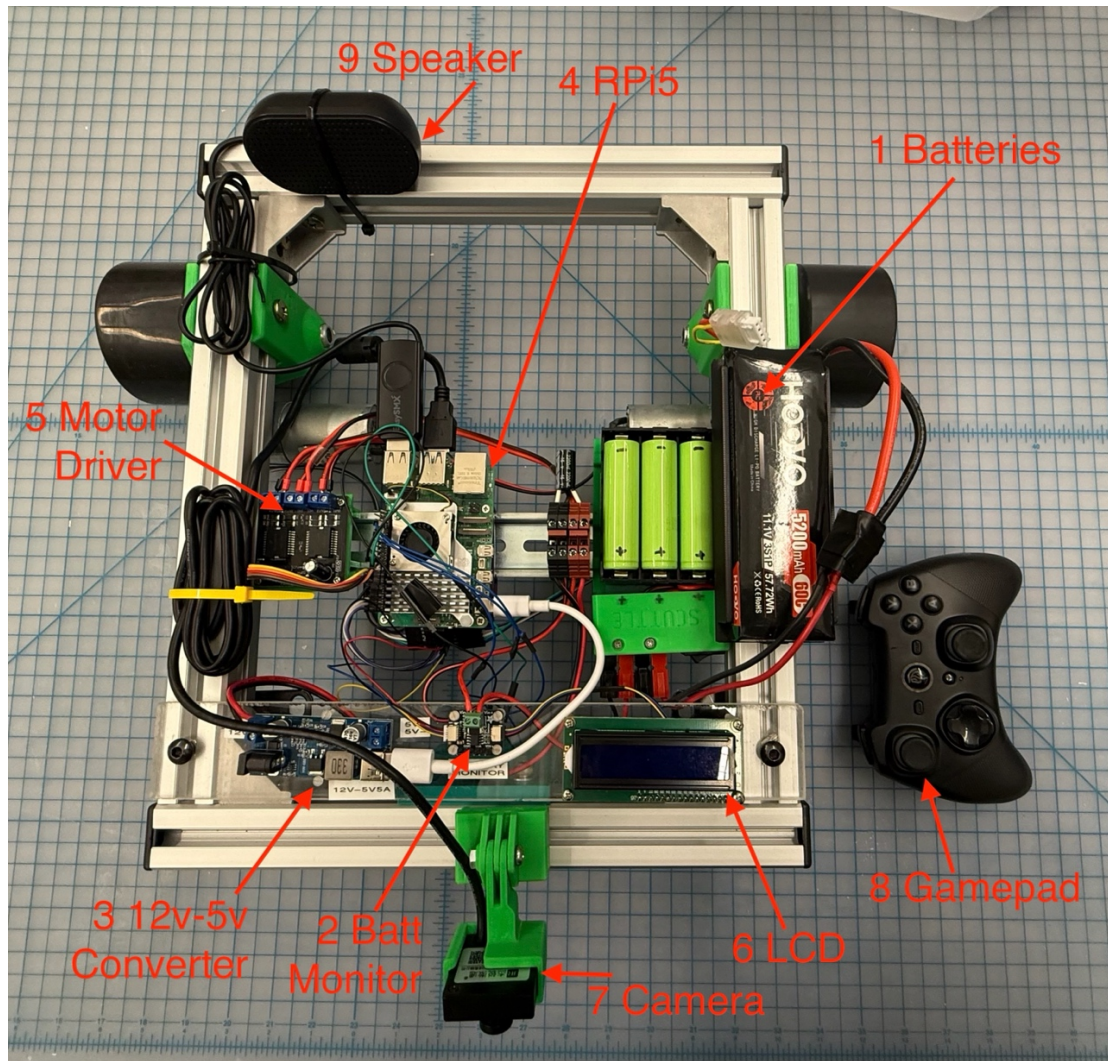
sturdy, and light-weight, providing a platform for a variety of modifications and sensors.



*Figure 1 The AACCC SCUTTLE Robot*

## Accomplishments

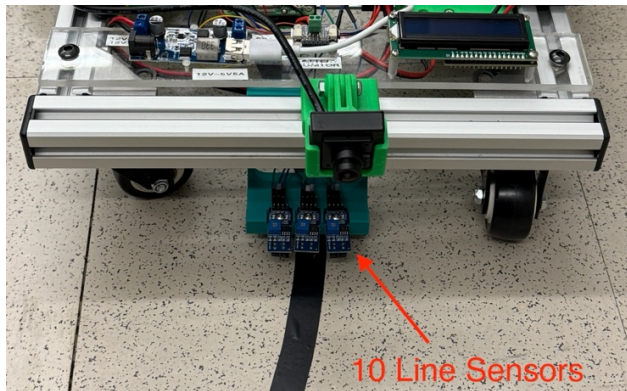
In the 2023-2024 academic year, the team acquired a SCUTTLE kit from Texas A&M University and assembled it according to the online documentation (Figure 1). The key components are identified in Figures 2, 3, and 4. The green and purple plastic parts of the robot are 3D printed.



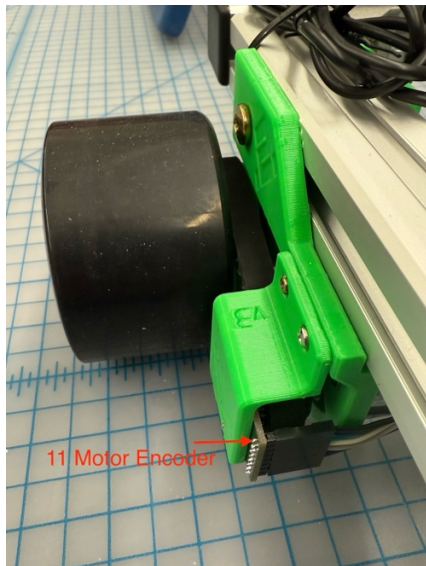
*Figure 2 SCUTTLE Robot Main Components*

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.



*Figure 3 Main Components*



*Figure 4 Main Components*

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 [5] via the general purpose interface pins on the RPi5.



## Initial Testing

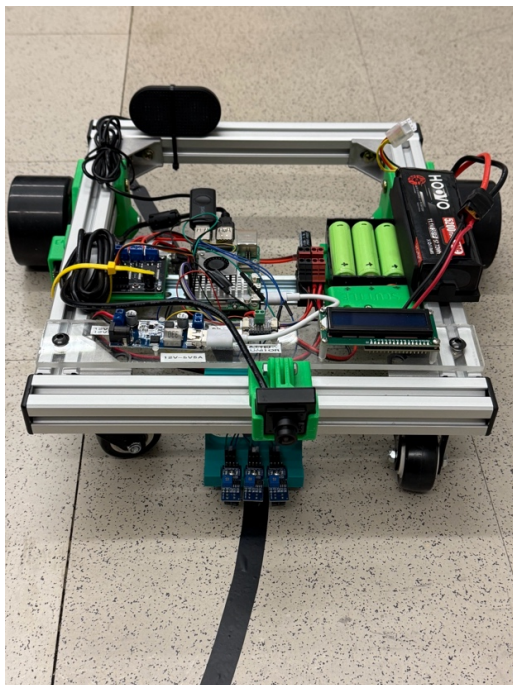
We loaded Linux and downloaded the SCUTTLE software onto the RPi5. It didn't take us long to find a low-level test function that ran the robot forward 4 seconds, stopped, then reversed for 4 seconds and stopped. It worked right away!

After studying the software some more, we identified the Gamepad manual control module and ran it. 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 Line Tracking

The first software that we developed was for line tracking, using three IR sensors mounted on the front of the robot. If the center sensor was active, then track straight. If the left or right sensor is activated, then turn towards that direction.



*Figure 5 Line Tracking*

Software was written to handle each of the eight states 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.

Line tracking wasn't 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.

## Battery Monitoring

The lithium-ion batteries that the robot uses 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.

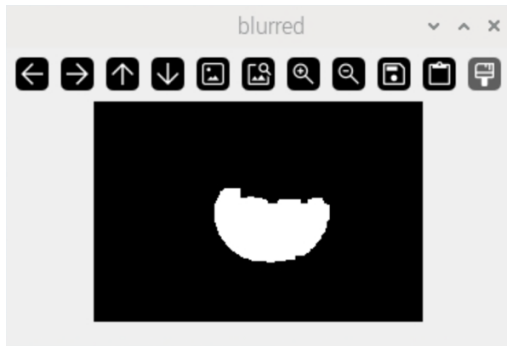
## Text-to-Speech

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 [6]. The battery monitoring software was then modified 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.

## Autonomous Colored Ball Tracking

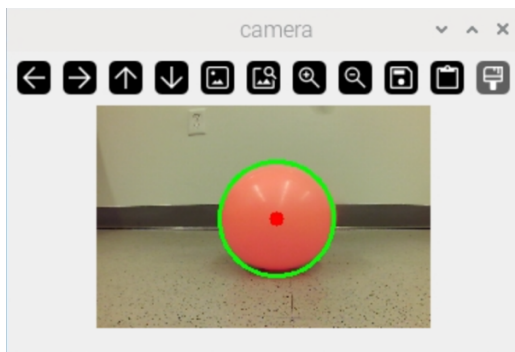
The SCUTTLE robot is outfitted with a forward-looking color camera that is connected to the RPi5 via USB. We loaded the OpenCV (Open Computer Vision) library [7] which provides numerous functions for image processing. The RPi5's compute power makes it ideal for running OpenCV applications. Fortunately, 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 6 Blurred Ball Image*

The blob image is run through blurring filters to smooth the edges (Figure 6) and a centroid is identified that encloses the ball's blurred image. This centroid can then be superimposed on the original image (Figure 7).



*Figure 7 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 slew 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. If the ball's image is too big, the robot is too close and motor commands are given to back away. Correspondingly, if the image 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. <pic showing light and dark sections of a ball> We also occasionally had problems with other objects in the background, such as wood doors that had an orange tint.

## Educational Impact

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).

The SCUTTLE robot was taken to the Annapolis Senior High School recruiting and science fair, raising awareness and interest in robotics and STEM curriculum. We were able to demonstrate Gamepad and ball tracking operations to high school students.

## 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.

## Hardware

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

### Inertial Measurement Unit

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.

### Light Detection and Ranging (LIDAR) Sensor

A pedestal will be installed for mounting a LIDAR sensor [8] 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 the third floor hallway.



## Ultrasonic Sensors

Ultrasonic sensors can be added to provide additional object detection to augment the LIDAR's operation.

## Microphone

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.

## Graphical Display Panel

We plan to install a graphical display panel on the pedestal that supports the LIDAR as an additional human interface element. 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.

## Additional Cameras

Adding more cameras would open the possibility for stereo-scopic computer vision. Side- and rear-facing cameras would enhance the ability to detect objects that are not right in front of the robot.

## Building an Additional Robot

We plan to build at least one additional robot that uses the same basic frame design but with changes to the drive system and component mounting system.

## 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.

## Web Status and Control

We would like the ability to monitor and control the robot via a web-based system. The current computer interface relies on 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.

## On-Board Artificial Intelligence

The existing camera system can be utilized with the TensorFlow AI software system available from Google. 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 [9] system running on the RPi5.

## Cloud-based Artificial Intelligence

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)

Simultaneous Localization and Mapping [10] 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 laid a solid foundation for undergraduate students to gain practical experience in robotics. By starting with simple line following and open-source computer vision for ball tracking, the project 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.

## References

1. Tally robot <https://simberobotics.com>
2. Rumba Robot vacuum [https://www.irobot.com/en\\_US/roomba.html](https://www.irobot.com/en_US/roomba.html)
3. Robot lawn mowers <https://www.pcmag.com/picks/the-best-robot-lawn-mowers>
4. SCUTTLE robot <https://www.scuttlerobot.org/>
5. Inter-Integrated Circuit (I2C) bus <https://en.wikipedia.org/wiki/I%C2%B2C>
6. Carnegie Mellon University computer speech group <http://festvox.org/>
7. OpenCV open source computer vision library <https://opencv.org/>
8. Light Detection and Ranging (LIDAR) <https://en.wikipedia.org/wiki/Lidar>
9. DeepSeek AI <https://www.deepseek.com/>
10. Simultaneous Localization and Mapping [https://en.wikipedia.org/wiki/Simultaneous\\_localization\\_and\\_mapping](https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping)