

Parallel Development of Autonomous Robots with a Focus on Modularity

Brittany Hand, Jeffery Hay, Ross Heninger, Andrew S. Mandeville, Matthew McDonough,
Jordan Miller, Fredi Mino, Fiona Popp, Corey Pullium

Mechatronics Engineering
The University of North Carolina Asheville
One University Heights
Asheville, North Carolina 28804 USA

Faculty Advisor: Jeremy Brown

Abstract

The SoutheastCon conference, hosted annually by region three of the Institute of Electrical and Electronics Engineers (IEEE), invites students to engage in academic competitions at a collegiate level. As a way of representing the skills taught at the University of North Carolina at Asheville, a team of mechatronics engineering students participates most years in the Hardware Competition. For the 2017 competition, students were challenged to design and fabricate one or more autonomous robots that would navigate on a four foot by eight foot game board and complete four stages in under four minutes. Each stage required the completion of several unique electrical, mechanical, and computing tasks. In order to minimize the time required to complete the challenge, it was decided to develop a robot for each stage. The robots were designed to be modular, incorporating an interchangeable chassis to simplify the design and fabrication process. Since the completion of some stages was dependent on information from other stages, the robots were wired and programmed to send pertinent information to one another using a serial peripheral interface bus. Parallel development proved to be an efficient way to distribute the workload between team members, and modular design provided a suitable environment for rapid development. Furthermore, this approach allowed sub-teams to easily change their design without delaying or affecting the final project deadline.

Introduction

Competition Rules

General specifications

To qualify for the competition, the robot as a whole had to meet a set of requirements. The stipulations pertained to the physical size, behavior, and safety of the robot. Firstly, it had to be able to fit within a 12 inch cube before the start of every round. After the start of the round, it could expand beyond this size or split into multiple robots. During the round, it had to stay within 3 inches beyond the walls of the arena. There was no limitation on the robot's weight.

Most importantly, the robot had to be completely autonomous, and could not receive information or be controlled during the round. If multiple robots were used during the round, they could not communicate through any wireless means; all communication connections had to be wired. In order to start the round, the robot was required to have a single start button, even if there were multiple switches for powering each individual robot. It was recommended to have a emergency-stop button as well, so if there was a problem the robot could be quickly stopped before damaging itself or the arena. It could not contain any flammable liquids or gases, and any stored compressed air was limited to 30 psi. Lastly, it could not cause harm to the arena, or present any danger to the judges or spectators.

The arena floor was constructed from a 4 foot by 8 foot sheet of smooth plywood and walled on every side by nominal “2x4” lumber. The playing field was divided into five levels at varying heights, with the lowest being the largest and also including the starting area, as well as the first three stages. The lowest level extended 57 inches long from the far wall of the arena, and 43 inches wide between the two side walls. The starting area was a 15 inch square centered horizontally and longitudinally on the lowest level of the arena. Each of the four stages were mounted on the surrounding walls and their centers aligned with the center of the starting area. The four other levels created a staircase leading up to the stage four target. The first step was 0.5 inches tall, 12 inches deep, and 43 inches wide. The second step was 1 inch tall, and the same depth and width as the first. The third step was a duplicate of the second step. The fourth step was 1 inch tall, 12 inches deep, and 12 inches wide, and sat in the center of the third step. The position and dimensions of all components in the arena were not subject to change.

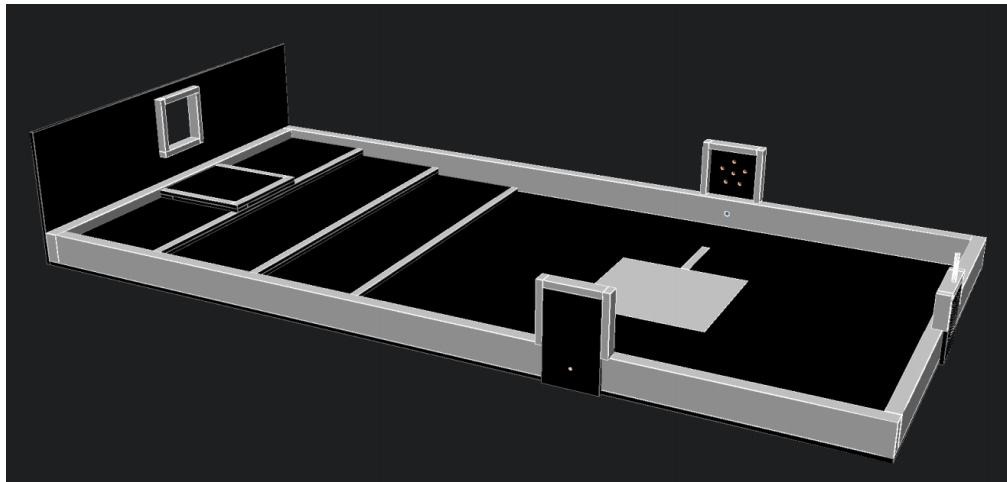


Figure 1. Arena [3]

stage one

In stage one, the robot had to accurately sense electrical components arranged randomly and determine a code. The setup of stage one was set up as described by the following excerpt from the SoutheastCon 2017 Hardware Competition Rules:

“This stage consists of six flat conductive copper pads arranged as shown in the diagram below. Each pad is attached to the back of the $\frac{1}{2}$ ” plywood and is accessible via a 0.5” diameter hole. The pads on the perimeter are arranged in a circle with a 1.5” radius around the center pad. The pads are only numbered in the following description; no number actually exists on the pad or on the stage). Pads begin with 1 at the top 0° position, with the next 4 counting off sequentially in a clockwise manner at 72° , 144° , 216° , and 288° . Between the common (center) pad and each of the surrounding pads are one of five components arranged in an order that is unknown to the robot when the match begins. Each component will exist once, and only once, but their order will be chosen randomly at the start of each match by a plug-in personality module. The five components are a wire, resistor, capacitor, inductor, and a diode. The message is encoded with each component representing a unique digit code value from as shown in the table below.”

Table 1: List of electrical components to be determined in stage one [3]

Code	Component type	Component value	Suggested vendor / part number
1	Wire	N/A	N/A
2	Resistor	10KΩ, 1/2W, 10% tolerance	Any
3	Capacitor	0.1uF, non polarized, ceramic	Any
4	Inductor	500mH, 30mA, 736Ω	Digikey, M10176-ND
5	Diode	IN4001–cathode/anode can be oriented in either direction	Any

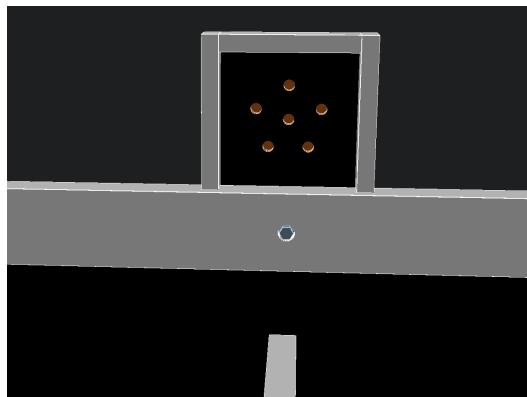


Figure 2: Stage one [3]

stage two

Stage two rose above the arena walls and consisted of a 3 inch tall by 6 inch wide plywood backboard inside a frame of 1 inch by 2 inch lumber, which created a one inch deep shadow box. This stage included an immobile cylindrical target rising approximately 7 inches above the frame, which was composed of a 3 inch section and 4 inch section.

The robot had to detect the presence of an electromagnetic field and set off a vibration sensor when the target was active. The robot could not touch the upper four inch section of the target, since it housed a fragile vibration sensor and a thin strip of LEDs. The electromagnetic field was generated by an electromagnetic coil located in the middle of the wall and directly behind the stage two wall mounting bolt.

After the first strike, the field would be disabled and during the next thirty seconds, it would be randomly activated and deactivated a total of four more times.

The starting time of the four additional activations varied from match to match, with the following rules:

- The target would be considered active only when the electromagnetic field was on.
- The electromagnetic field would be active each time for two seconds, or until the vibration sensor was set off.
- Each deactivation would vary in time, with a minimum set time of one second.
- The final (fourth) activation would occur at exactly twenty-eight seconds.

Points would be awarded when the target was hit in its active state and at most only one hit would be counted. Points would be deducted when the target was hit in its inactive state and at most one negative hit penalty would be applied. Hits occurring during the first five tenths of a second of the deactivation would be considered neutral and would not count towards the final score.

stage three

Stage three was intended to be completed after stage one, determining electrical components, as information gathered from the earlier stage was critical to the completion of stage three. The construction of stage three was a 1 inch by 2 inch lumber frame enclosing a 1 inch deep shadow box, in the middle of which was a 0.5 inch diameter quadrature encoder knob. The goal of this stage was to successfully manipulate the encoder knob in a fashion similar to a combination lock. The knob was to be turned initially clockwise, then alternating in the opposing direction, until a series of five combinations had been achieved. Each rotation was supposed to be a full three hundred and sixty degrees. The codependence of this stage with stage one lies in the derivation of the correct combination. If the robot had successfully concluded the order of stage one's circuit components, it would hold the correct rotation combination for stage three's dial. The accuracy of each full rotation was to be within +/- fifteen degrees (or one measurement point on the twenty-four point encoder knob) in order to qualify as a valid input. Should a team's robot enter more than a string of five inputs, only the last five inputs would be recorded on the stage, allowing for corrective measures to take place, or multiple attempts by the robot if necessary.

final stage

In the final stage, the robot had to launch three foam darts through an opening in one of the arena's walls. The target consisted of a 6 inch by 6 inch square framed hole located in the horizontal center of the wall at the end of the arena which was farthest from the starting area. The bottom of the hole was located approximately 7.5 inches from the lowest level of the arena. The robot had to launch three foam darts through this target, and could do so from anywhere on the arena, even by climbing the platforms leading up to the target. The robot would receive points if at least one dart launch was attempted, and received points for every dart that made it through the target. Once all three darts had been fired, the round was considered over. Misses and misfires were acceptable, but if the judges suspected darts were intentionally being launched outside the arena, the robot would be disqualified.

Initial Design Concepts

The first design decision discussed was the number of robots to be used. A wide variety of geometric shapes were considered for the configuration of each chassis. For the idea of a single square robot, each of its sides would be dedicated to one of the challenges; the components for stage one and stage three would be on opposite sides - the components for stage two and the final stage would be opposite each other.

The alternative option was to use multiple robots. This approach would allow parallel development since each chassis would be independent. Furthermore, the use of multiple robots would allow the challenges to be completed more quickly. In a three robot layout (Figure 3.a), there would be two small robots with the same dimensions and one large robot. The small robots would complete stage two and the final stage, while the larger robot would complete stages one and three, eliminating the need for communication between these two stages. Triangular and square chassis were considered for a four-robot configuration. The triangular chassis (Figure 3.b) was not practical since it would make wheel placements difficult. The four square chassis design would dedicate one chassis to each challenge (Figure 3.c). In addition, this would allow for modularity since there would be a uniform chassis.

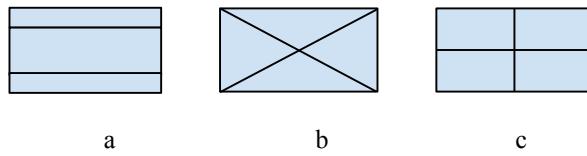


Figure 3. Configuration of chassis

Finally, it was decided to use the four square chassis setup. This was chosen so the challenges could be completed quickly and so the robots could be designed and tested in parallel. Each stage had a group tasked with designing the device that would be used to complete the stage's task. Each group had to mechanically design the actuator, complete the needed circuitry, and program the microcontroller to do the required tasks. Another choice made was to

make one of the four robots a stationary tower. This motionless robot would complete the final stage since it would be more consistent to aim the darts on a stationary platform than to correctly and consistently align a mobile robot.

Design

Modular Chassis

With the decision to use multiple robots in a square pattern, the challenge of designing chassis was left to a separate group, which allowed for better parallel development of all the tasks and a singular chassis design. The challenge that arose from this, was creating a chassis that could satisfy each challenge's need. Each chassis needed to contain: the navigation hardware, power management circuitry/hardware, and communication lines. The chassis top had to rest no more than 3.5 inches above the gameboard surface and the front-middle section needed to be left available for the stage two challenge.

Starting with a board up design idea, 54mm mecanum wheels, micro DC brush motors, and individual DC motor drivers (Adafruit DRV8871), were chosen due to their space efficiency. A circuit board was designed and prototyped to ensure proper function of the drive system. This was all attached to an aluminum lower chassis plate that would support the entire robot.

With the drive system complete, the location of the navigation system hardware needed to be considered. With the use of ultrasonic distance sensors from Parallax, it was determined two forward and one side sensors were needed to properly translate and rotate. The most efficient locations for the forward sensors were above the wheels at the edge of the chassis. This configuration allowed for the greatest distance between the sensors, which would result in more reliable distance sensing, since the further apart the ultrasonic sensors were more information about position would be known. The side sensor was chosen to be located in the middle of its respective side. These sensors were attached to an aluminum top chassis plate that was then, using spacers, mounted to the lower chassis plate. This brought the height of the chassis just under 3.5 inches.

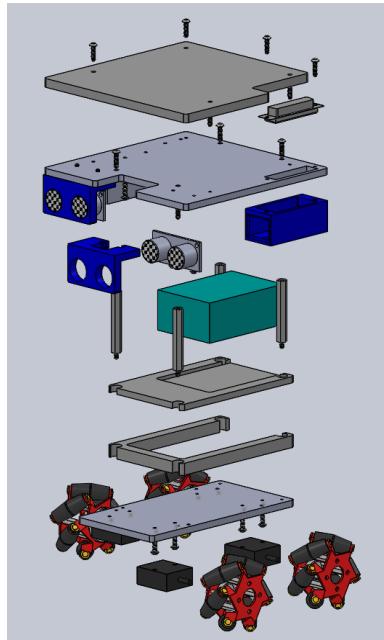


Figure 4. The chassis

The next step was to determine and install the battery, power management circuit and hardware. The battery was chosen after first determining the max current and max voltage needed in any of the stages. It was determined a 12V Tenergy 2000mAh NiMH battery would satisfy the power requirements of all the robots and easily fit within the chassis. With the battery known, a power management system needed to be designed to supply all components with proper supply voltage. It was determined only two voltages were needed, 5V and 12V. A voltage regulator and

protoboard were used to create 5V and 12V header rails supplying power to all components. The battery was ran through a kill switch, mounted to the upper chassis, before going to the voltage regulator to ensure safe operation of the chassis. The mounting of the battery was done between the spacers of the upper and lower chassis plates and the voltage regulator circuit was mounted to the upper chassis just off center of the battery.

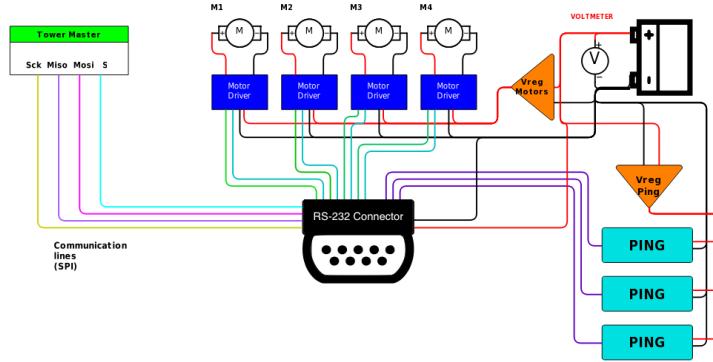


Figure 5. The electrical layout for the chassis

The communication line, 6-wire CAT5 telephone cable, was contained within the chassis using a custom 3D printed tray mounted to the upper chassis supports. The tray's lower section contained the cable while the top contained the battery. This allowed the cable to be folded and placed in the tray and would unfold when the chassis moved during competition.

Since each chassis was identical, a connection cable was added to allow the chassis to be fully modular, which allowed any chassis to be used with any challenge. A RS-232 D-Sub Connector was used to enable the chassis to interface with each challenge. Since each challenge contained its own microcontroller, all logic lines needed to be feed to the challenges as well as power and ground lines. The pinout diagram for the RS-232 D-Sub connector can be seen in Figure 6.

To be able to attach each stage's components to a chassis a challenge mounting plate was used. This allowed the stage's components to be built upon this plate and using a predefined set of mounting holes, attach the challenge mounting plate to the chassis. Spacers were to be used to adjust the height of the challenge plate to the height chosen to be appropriate for each stage.

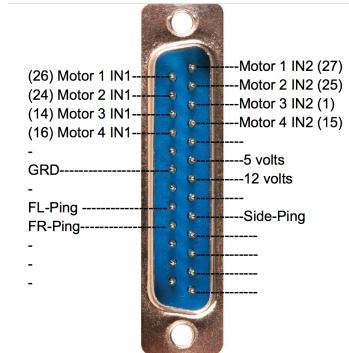


Figure 6. The pinout for the RS-232 connector
Navigation

An important component of having an interchangeable, modular chassis was to design a navigation code that could be easily transferred from challenge to challenge. A formulated benefit of having three chassis with the same lower hardware was the development of a single code to run them. From the top level, a single function could be called with the setpoints (fixed location) of any particular challenge and any one chassis could navigate to that point. The

only consideration after the development of such a code would be the initial orientation of all of the robots to avoid any competing paths between robots to their respective stages.

In order to prevent the route of the robots from being open loop, the onset of navigation code development was structured with PID tuning capabilities. The setpoints, in centimeters, of the desired location were passed to the navigation function within the code. After each chassis was given the signal to begin navigating, the three ultrasonic distance sensors would fire and record three values: distance from the front right corner of the robot to the forward wall, distance from front left corner of the robot to the forward wall, and distance from the side of the robot to the side wall. The error value from each setpoint would determine how much proportional gain was to be passed to influence that movement vector in the mecanum wheel array. A rotational error was derived to ensure the alignment of the robot would be square with the wall.

A variable sized error array also simultaneously stored and updated in order to reject any instantaneous spikes in readings. In addition to having a rolling value of error for smoothed data, referencing could be done with the most previous value for generating a derivative gain factor and an error summation factor for integral gain accumulation over time. Header files containing the macros for the quick tuning of not only the setpoints, but of the constant PID gains, and of the error array sizes, meant navigation for any stage could be passed from one robot to another with no issues of incompatibility.

Design of Individual Challenges

stage one

The first stage of the challenge was to acquire a coded sequence to be used in stage three. The challenge required the robot to sense five electrical components arrayed in an unknown parallel configuration with a common ground. Since each component had a unique combination of current and voltage values, it was determined the simplest way to identify them was to supply power to each component and measure voltage and current values. This alone was used to accurately identify three of the five components outright. The only components not able to be determined by this first test were the capacitor and the backwards diode, which both appeared as an open circuit. If the controller determined there was an open circuit, a second test was used to decide between the capacitor and the backwards diode. A relay was engaged to switch the component from power to ground. The voltage across a discharging capacitor decays exponentially while the voltage across a backwards diode drops to zero immediately. These properties were used to distinguish the two components from one another.

To sense the current, an INA219 current sensor was employed. The input to the sensor was provided from the 5 Volt output of the Teensy microcontroller. The input was placed in series with a 1K Ohm resistor to limit the current through the sensor. The output was connected to an array of five N-Channel MOSFETs, controlled from digital I/O pins on the Teensy microcontroller. Each MOSFET corresponded to a particular contact on the testing rig. When engaged, the MOSFET completed the circuit between the current sensor and its corresponding component. By engaging each MOSFET in sequence, the entire challenge was completed without the need to reposition mechanically. Furthermore, using MOSFETS, as opposed to relays, conserved space and allowed the challenge to be completed more quickly.

The physical connection between the challenge and the test circuit was achieved by using spring-loaded pogo pins. The pogo pins were supported with two 3D printed support rings. The first allowed the pins to pass through, with the secondary ring holding threaded brass pads to which the pogo pins were soldered. Wires were then connected to the brass pads in order to run the test circuit.

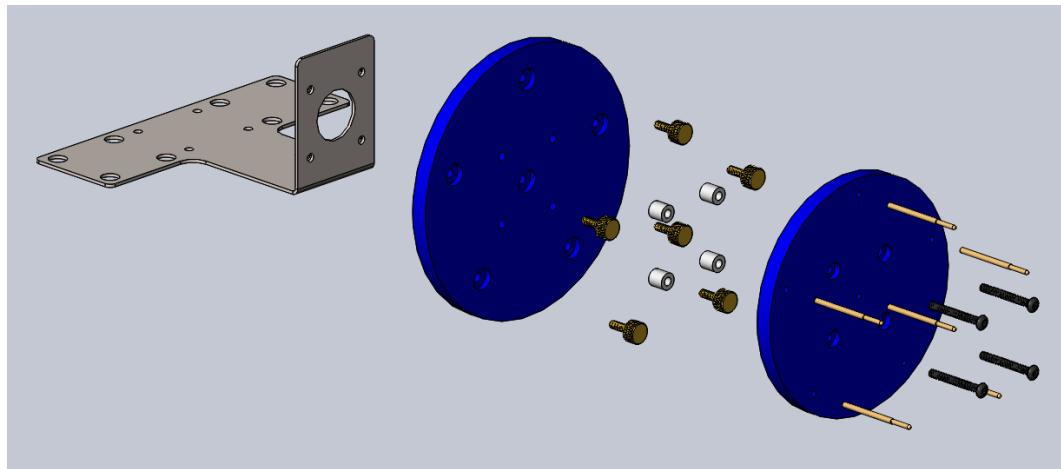


Figure 7. Pogo ring assembly

To be able to engage the challenge and remain within our initial chassis footprint of 5.5 inches x 5.5 inches, a mechanical system needed to be implemented. Initial ideas for generating linear motion were to use a linear actuator, a pantograph system, and a rack-and-pinion system. Due to implementation space constraints the rack-and-pinion was chosen. The pinion was driven by a small continuous servo motor, Tower Pro SG92R, with limit switches to determine the forwardmost and backmost travel positions. The linear system was composed of aluminum rails on a 3D printed support structure. Custom made rack and corresponding gear were 3D printed. Diagram of the linear rail system are shown in Figure 8.

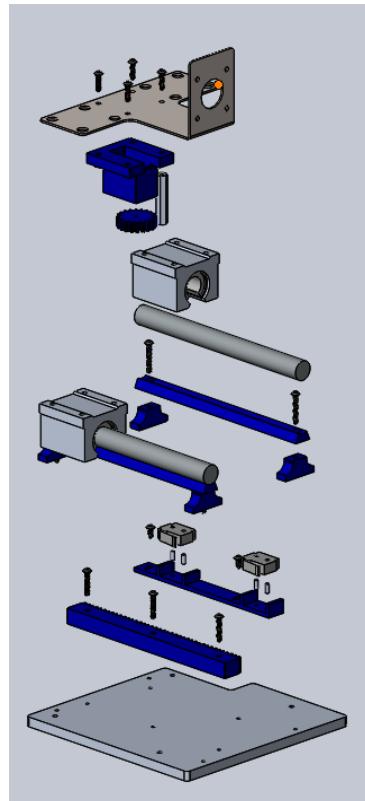


Figure 8. Linear slide assembly
stage 2

In order to sense a change in the electromagnetic field the Hitachi HM55B Compass Module was utilized. Before the testing of the electronic compass, a Single Axis TO-92-3 hall-effect sensor was researched and tested for measuring the magnetic field. Using this approach, the hall effect sensor would be employed to find reference values for when the electromagnetic field was off and when it was on. The magnetic field reference value when the field was off would be gathered when the robot was in the starting square. The magnetic field reference value when the field was on would be gathered once the robot had approached stage two, before the vibration sensor was activated. The electromagnetic field would remain active until the vibration sensor was triggered. The generated electromagnetic field was weak compared to sensitivity of the sensors which caused inaccuracy in their readings. The following algorithms were tested to try and combat the unreliability of the hall effect sensor: reading the hall-effect sensor values and comparing them to a magnetic field reference value when the field was inactive, reading the hall-effect sensor values and comparing them to a magnetic field reference value when the field was active, utilizing the magnetic field reference value and creating an array to check if a certain number of consecutive values were in a certain range, a running average on the electromagnetic field values, utilizing debounce to try to eliminate instability, and integrating a current sensor on the signal line.

The electric compass, Hitachi HM55B Compass Module, proved to be more reliable than the hall effect sensor. The coil which produced the electromagnetic field would act as a magnetic north pole when activated, which the compass would detect, unless the compass and the coil were in line with the Earth's north. For programming, the following algorithm was used. As done with the hall effect sensor, reference values would be read for when the magnetic field was on and off. An average of ten consecutive readings was used for both the inactive magnetic field and the active magnetic field values. Then, using the value for when the magnetic field was on, a range was calculated. If the readings were inside this range, the robot would determine the electromagnetic field was on, and it would strike the vibration sensor. Two parallel compasses were utilized, spaced out so when the robot was at the wall the coil would be in between the two sensors. This was done so if one was in line with north the other could still detect the change in magnetic field. Shown in Figure 9, are examples of the testing done to determine if the compass was in line with north or not.

Calibration done angle_off: 141 128 127 133 133 127 131 135 128 127 131 Calibration done angle_on: 142 range: 141 - 143 ***in line with north*** value discarded!!!	Calibration done angle_off: 147 121 120 121 121 119 118 119 118 121 123 Calibration done angle_on: 120 range: 119 - 121 good to go
--	--

Figure 9. Data from testing if compass was in line with north

Regarding the mechanical design, it was decided a blade parallel to the ground would hit the light saber every time the electromagnetic field was detected. A HS-645MG high torque servo motor was implemented in order to move the blade. This type of motor was chosen based on the need for high speed and torque. Additionally, the servo-motor did not require the use of a driver, making the mechanical design and circuit layout simpler.

The gear ratio was arranged with the intention of trading torque for speed.. A VEX sixty tooth gear with 2.5 inch pitch diameter gear was attached to the servo-motor and a VEX twelve tooth gear with 0.5 inch pitch diameter was connected to first gear. This resulted in a gear ratio of five, which caused the speed to be increased by a factor of five. It was decided to trade strength for speed because a quick reaction time was crucial for this challenge.

One of the most problematic parts of this challenge was to find flexible and malleable materials that could be used for the robot's blade. Initially, rigid materials were included in the design, however, when they were tested, it was revealed that these kind of materials would physically interfere with the motor settings.

The testing of the motion of the stage two robot's blade was the following. The bottom section of the stage two cylindrical target was centered at $(0,1)$. The robot's blade was centered at $(0,0)$ and started at $\theta = 0$. The rigid blade was programmed to travel from $\theta = 0$ to $\theta = \pi/2$ and then move back by $\theta = \pi/2$. Yet, the fact that the bottom section of the target occupied an area that included $\pi/2 > \theta \gg \pi/3$ was not taken in account. This situation made it impossible for the lightsaber to reach $\theta = \pi/2$. Instead, the motor would keep spinning trying to reach that position during the amount of time of the delay specified in the code. After the delay, the saber would go back by $\theta = \pi/2$. However, since it started to move back at $\theta < \pi/2$, the final position was $\theta < 0$. This state of affairs would repeat each time the electric field was triggered, and each time, the saber would start further back. By consequence, the strikes had a gradually decreasing contact force that would no longer trigger the sensor after the first two hits.

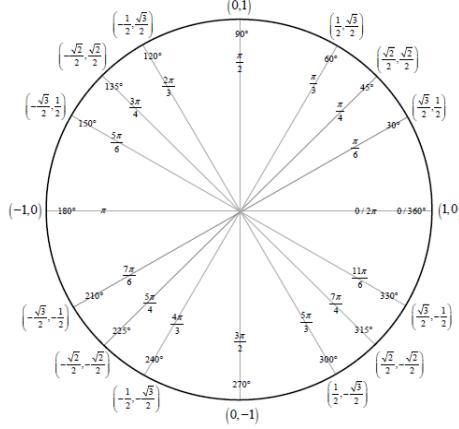


Figure 10. Map of coordinates for the blade test

In order to solve this issue, multiple deformable materials were tested. Doing this, the blade would be able to temporarily change its form so the motor could reach $\theta = \pi/2$ and the material would regain its original shape after losing contact with the hilt. Nevertheless, none of the tested materials had enough mass density to produce a significant momentum to trigger the vibration sensor.

Finally, instead of trying to create a completely inelastic collision, it was determined to create an elastic collision. For this purpose, a second axis was added at the base of the blade which allowed it to retreat after the impact with the hilt. The second axis utilized was a hinge. Furthermore, an aluminum plate was attached to the hinge as a blade. When tested, the new design effectively allowed the motor to reach $\theta = \pi/2$, while transmitting most of the kinetic energy to the vibration sensor.

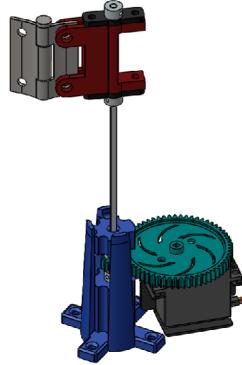


Figure 11. CAD representation of the mechanics of the stage two robot

stage 3

In order to rotate the encoder it was decided to use an end effector. The top-side, mechanical design of the stage three robot was centered around creating effective means to bring the end effector to the encoder knob, initially from a compact starting position. The manipulator was a Lynxmotion Little Grip Kit compressing claw with an attached HS-422 servo motor to drive the contraction. The rotational degree of freedom on the manipulator was generated by a single, bipolar stepper motor affixed to a NEMA 17 mounting bracket, and controlled by a Pololu A4988 stepper motor driver. In an effort to reduce the time it could theoretically take to complete the challenge, a custom interfacing slip ring was fabricated between the stepper motor and the manipulator.

A slip ring was designed to allow for continuous rotation of the manipulator during the challenge while still allowing full control for the manipulator's servo. The slip ring allowed power, ground, and a logic line to run from the manipulator's servo motor to the controller. This was achieved by using three radial axial bearings to transmit the signals over the rotating shaft. Wires were soldered inside the bearings using an aggressive flux and solder made for stainless steel. The bearings were then press fitted onto the motor shaft adapter. A 3D printed housing, with bearing spacers, was lined with copper tape to carry each signal to the edge of the housing where a wire was connected using a conductive washer and bolt. This allowed the stepper motor to turn at max revolutions per second without affecting the use of the manipulator. The CAD drawing of the slip ring system are shown in Figure 12.

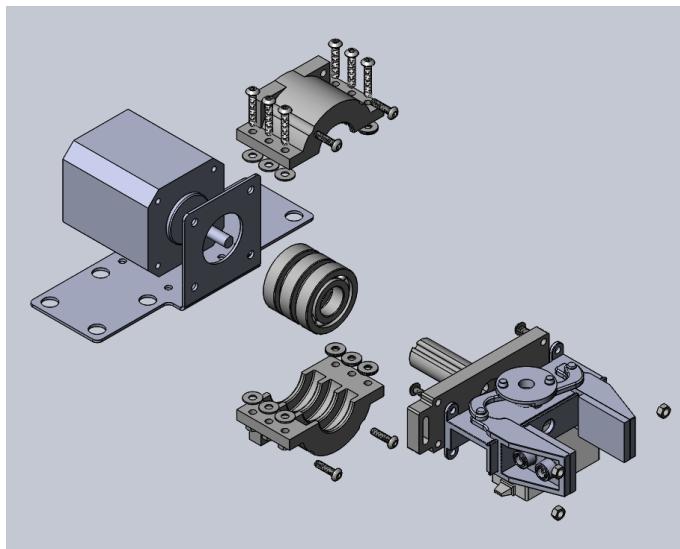


Figure 12. CAD drawing of slip ring

In order to adhere to the modular 5.5 inch x 5.5 inch top down uniformity of the robots, a linear slide was also incorporated into the dial robot. This was done to ensure the starting position of the robot would be confined well within overall size constraints and, after navigation had occurred, the manipulator could be extended to the challenge dial without coming into a compromising range of the forward ping sensors (roughly 1.97 inches away from the base wall). This linear slide was composed of aluminum rails on 3D printed supports, which operated on a 3D printed rack and pinion path. The pinion gear was driven by a Tower Pro SG92R micro servo.

Once the stage three robot had signaled to the final stage robot it had completed its navigation, the final stage robot would send a signal to the robot to exit its navigation state and enter into its challenge state, along with two byte sized messages containing information from the stage one robot. Before the robot would begin manipulation of the dial, it had to derive a combination from the two byte package in a process outlined in the Figure 13 below. An array of zeros was initialized in a function and two integer variables stored the each byte respectively. The operations in the Figure 13 were performed on the decimal values and the combination array was filled. Because the combination would have no repeating values and would always be between the integer values one through five, the last combination value could be assumed to be the only integer between one and five that was not previously used.

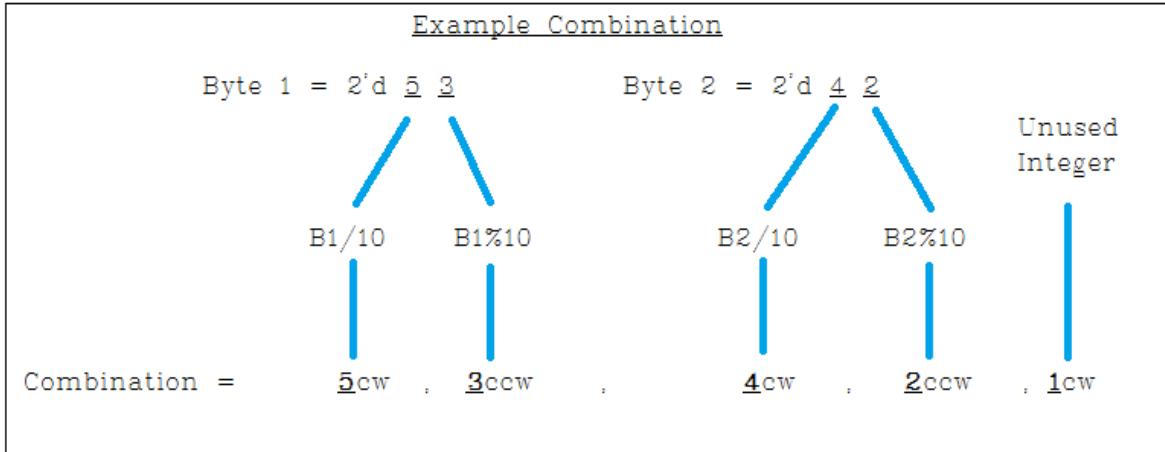


Figure 13 . Example combination

Once the combination array was filled, the robot proceeded with the challenge by extending to the encoder knob (stopping when the pinion gear approached a forward limit switch on the rack) and closing the gripper servo on the dial. Then, the stepper motor completed the series on the knob by moving through the combination array, multiplying the number of steps for a full rotation by the current array element. After the combination was entered, the robot retracted its arm (to a rear limit switch on the rack) and signaled to the tower that its challenge state had ended.

This point in the overall competition highlighted the advantage, with respect to time, of having a set of smaller robots communicating to each other, as opposed to one robot having to navigate multiple times, completing each challenge as it went.

final stage

As it was decided that the final stage robot would be stationary, the robot would need to be able to shoot the foam darts through the open target from a distance with consistent accuracy. For this apparatus, there were a few design options considered. A directly spring-powered dart launcher would have been consistent in its velocity, but might have been inaccurate and underpowered. A spring and piston design was considered; much like commercial foam dart guns, the piston would have compressed air in order to propel the dart forward. A third, and more direct option was to simply use compressed air, at the regulated 30 psi which was allowed, to propel the dart.

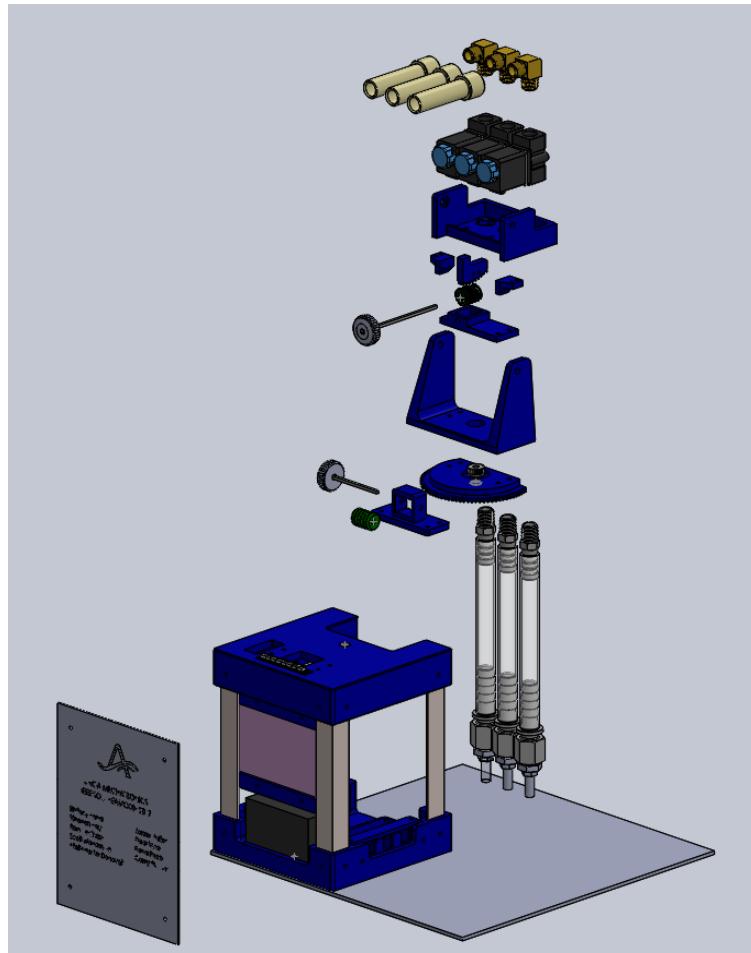


Figure 14. CAD drawing of final stage/ tower

It was determined to use a compressed air system, controlled by US Solid 12VDC solenoid valves. Because the pressure was limited to thirty psi, it was also decided each of the three darts would have their own independent air chamber, in order to ensure consistency, and to bypass effects of pressure loss that would be seen if a single chamber was used. This would also allow for greater control over when each dart could be fired. A prototype of one of the dart shooters was constructed using 0.5 inch CPVC as a barrel, as its inner diameter was a snug fit for the foam darts, various other pipe fittings, and 0.5 inch polyethylene hose as the air chamber. The chamber was pressurized to 30 psi during the tests using a gauged hand pump, and a 12 volt signal was applied to the solenoid valve to fire the dart. Before each individual test, a small amount of black acrylic paint was applied to the tip of the dart to mark the position it contacted. The results can be seen in Figure 15. Two tests were conducted: fifteen darts were fired with no delay between loading the dart and firing, and five were fired with a 1 minute delay. It was found with no delay, the darts would consistently hit within a 3 inch circle when fired 6 feet away from the target. The delay test indicated there were pressure leaks present in the air system, as the average contact point shifted downward. These leaks were easily fixed with better fittings and sealants.

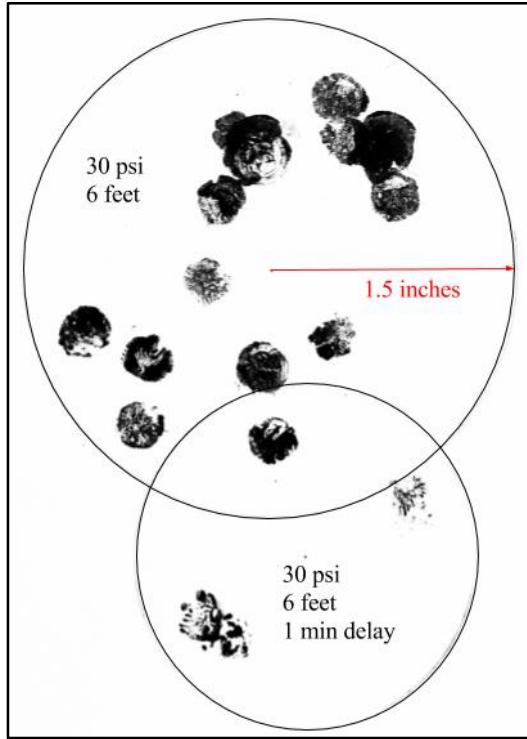


Figure 15. Dart shooter precision test data

From this information it was decided the dart shooter could be mounted to the fixed robot, and aimed before the challenge began. An apparatus was needed to allow the aim of the dart shooters to be adjusted for rotation and elevation. A mount was designed to hold the three dart shooters in place, and could be tilted and rotated via the movement of two respective worm and wheel gear assemblies. The worm and wheel would ensure, once adjusted, the aim of the dart shooters could not be altered unintentionally.

There were a few designs discussed for the fixed robot on which the dart shooter was to be mounted. One idea was to have a garage from which the three other robots would be housed under and would drive out from at the start of the match. The garage would not only have the dart shooter built in, but would also store the main electrical components, including the circuits and batteries, for each of the robots. Another idea for housing the electrical components was having a 12 inch by 12 inch base, with the circuitry underneath and the robots sitting on top at the start. The design decided upon was a tower with the dart shooter mounted on top and the electrical components stored on the inside.

Once the tower idea was chosen, the layout for the inside of the tower was designed so the hoses for the dart shooter and the electrical components would all fit within. In order to make the robot designs more modular, instead of the tower housing all electrical components, it was decided each robot chassis would house its own battery and circuitry, communicating with the tower through serial peripheral interface, using telephone cables. This modular design meant the tower only needed to store its own battery, protoboard, control buttons, and the three RJ11 jacks through which the robots plugged into and communicated with the tower. A circuitry drawer was considered in order to store the battery and circuit. However, one of the limiting factors for the inside layout was that the hoses for the firing mechanism curved far into the inside of the tower. For this reason, instead of having a drawer, the protoboard was mounted on the front inside of the tower as high as possible facing inside, and the battery was mounted immediately below. The control buttons consisted of a switch to give power to the tower, a start button to begin the match, and a button to stop all robots in cases of emergency. These buttons were mounted on the top plate of the tower for easy access. An LED strip was mounted in the top plate for debugging the tower code, and the RJ11 jacks were mounted onto the side of the bottom plate.

The top and bottom plates of the tower, the fire mechanism mounts, and the electrical component mounts were designed in Solidworks and 3D printed. The plates were held together by four angle irons. The design goal for the plates and mounts was to make everything as modular as possible so if a component broke, it could be easily replaced. The plates were designed so the corners would press fit onto the angle irons, and thus no screws were

needed to hold the tower together. In order to pressurize the firing chambers, the firing mechanism was removed from the tower by removing two screws and pulling it out the backside of the tower.

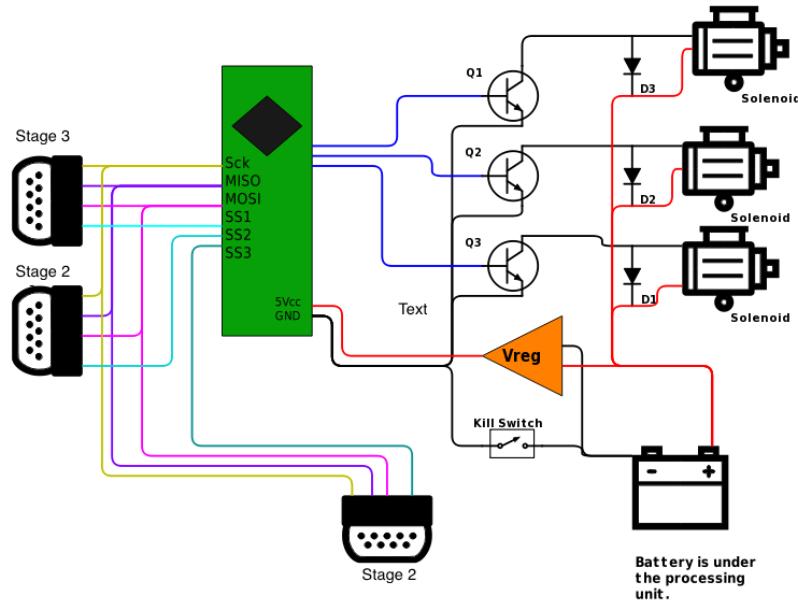
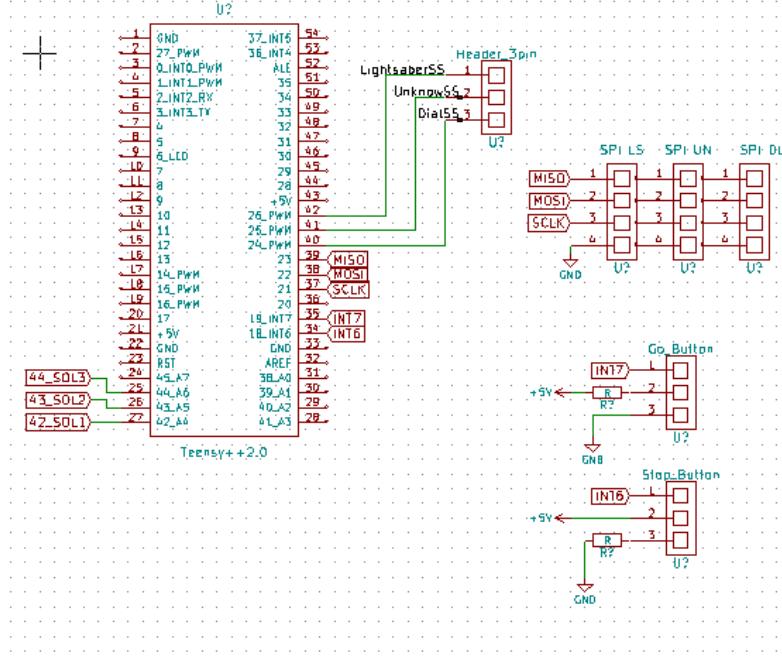


Figure 16. The electrical layout for the tower

Communication

The specifications required each match to begin with the push of a single “kill switch”, and any communication between robots be done via a wired connection. To meet these specifications serial peripheral interface (SPI) was chosen as the communication protocol between robots. Three CAT5 cables were used to connect MISO, MOSI, clock, slave select lines, and a common ground between the tower and the stage one, two, and three robots. The stationary final stage tower robot acted as the master, dictating the timing and sequence of each message.



design was based on four robots, there were six critical components that could be tested in parallel: the modular chassis, the stage one electrical component probe, the stage two electromagnetic pulse response, the stage three encoder manipulator, the stage four dart shooter, and the serial peripheral interface communication.

The modular chassis had several goals set to prove it would be effective. Upon completing the fabrication of a chassis, each voltage regulator, motor, motor driver, and distance sensor was checked to ensure they behaved as expected. In addition, the connections between the wires and the RS-232 connector were examined for proper continuity. Once those tests were proven successful, the code for PID navigation was able to be tested. It was determined that in order for the chassis to be a successful solution to traversing the challenge board, it would need to be able to navigate to a set point on the board with a tolerance of plus or minus a quarter of an inch. With some fine tuning, the PID navigation was able to meet that tolerance consistently enough for the design to be proven as a success.

The stage one electrical component probe was tested in two separate increments. First, circuitry was set up on a breadboard and an arduino was used to test the ability to distinguish the different components based on the current and voltage response. Once the electrical analysis was proven to be a successful way of decoding the components, the mechanical design became the focus of testing. Using a rack and pinion to force contact between the pogo pins and the brass pads and then running the electrical analysis was the final test proving this design to be successful. The stage two robot's success was based on its ability to sense the change in electromagnetic field on the board and respond accordingly. Once the electromagnetic field changed, there was a short time frame where the response was valid. When the design was proven to consistently respond within that timeframe it was considered successful. The stage three encoder manipulator was proven to be successful when it could successfully enter the code in the encoder. In order to prove the success of the design, the manipulator demonstrated its ability to complete five full rotations of the encoder knob with an accuracy of plus or minus 15 degrees. The final stage's dart shooter's success was based off of a consistent ability to maintain air pressure and shoot the darts accurately. Once accuracy was proven, the ability to maintain the needed air pressure was checked based on a time delayed shooting test.

The serial peripheral interface was able to be developed independently because it was one of the few objects in the critical path of development that had no mechanical dependencies. Matches were able to be simulated without any of the robots being completed. Success was proven when the state machine for each robot was simulated and byte packages were communicated between them allowing each state machine to simulate its role in the competition.

The final tests came from putting all of the pieces together and running each robot with navigation, its challenge, and communication and testing everything in unison. Parallel development was essential in getting to this point. Being able to test different pieces without having to worry about unnecessary dependencies was a primary reason for the timely completion of each goal.

Results

Modifications During The Testing Runs At The Competition

The day before the competition was spent debugging the robots and testing them on the practice game boards set up at the conference. The navigation values for the stage one and stage three robots were adjusted to better align with their challenges. Throughout the day, twelve motor drivers were replaced on the robots. The motor drivers were breaking due to the load of the robot exceeding what the drivers were capable of carrying. However, due to the modular design of the chassis, these drivers were easily replaceable.

The final stage tower and stage two robot worked consistently throughout the day and were ready for the competition, so they were not continually tested. In the worst case scenario, these two robots, due to the modular design, could have competed by themselves and completed two of the challenges. If only one robot had been designed to do every task, one problem could have meant not being able to compete at all.

For some unknown reason, the stage three robot stopped working correctly the night before the competition. After a few hours of checking the wiring, the code, and the motor drivers of the stage three robot, the problem was still not discovered. So, it was decided the backup chassis would be used. In less than an hour the backup chassis was functional and the challenge portion of the robot was mounted. Due to the backup chassis and the interchangeable design of the chassis, on which any of the challenge portions could have been mounted, all the robots were able to compete.

The Competition

During the day of the competition, only minor alterations were made to the robots, mainly to adjust the navigation values. The stage one robot had trouble aligning accurately with the challenge, but was still able to decipher two of the numbers each time. The stage two robot received all possible points for its challenge in all but its second competition run, when it double hit the lightsaber at the start. The stage three robot had smooth navigation in all but the second run, when it was bumped off course possibly due to another robot or the CAT5 cable being taught. In this run the stage three robot never exited the navigation function. In all other runs, the stage three robot turned the dial correctly based on the code it received. The final stage robot received all possible points for its challenge in all three competition runs.

In the first run, the team received 585 points, the second run 320 points, and the third run 605 points. The team received fifth place overall (one team away from competing in the finals) out of the forty-two teams in the collegiate division.

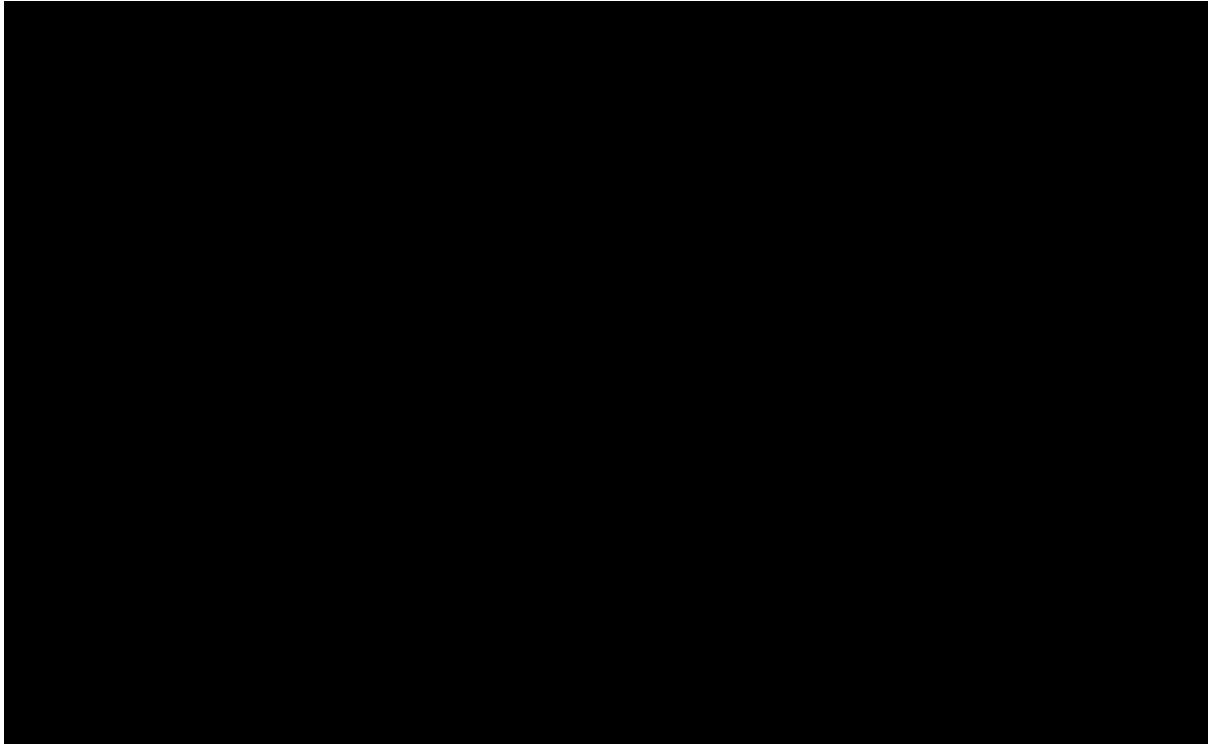


Figure 18. The ranking of the teams before the finals

Figure 18 The top four teams would go on to compete during the banquet.

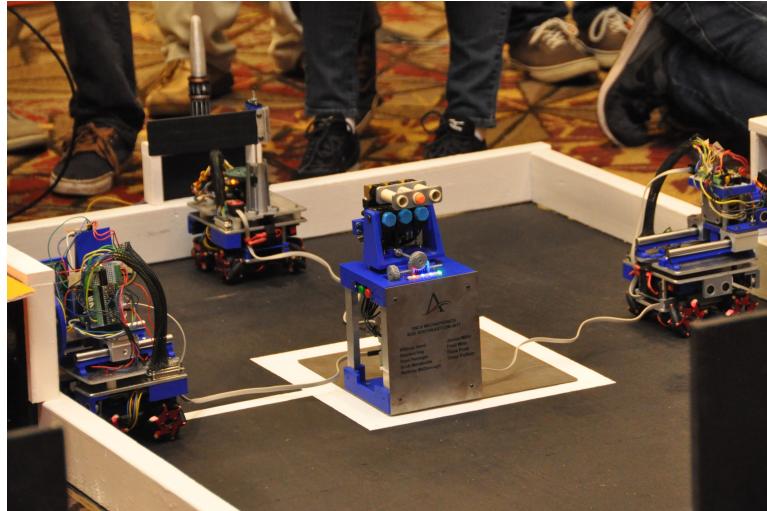


Figure 19. The robot during the competition

Figure 19 The photo credits go to James Willamor who was the official photographer of the IEEE SoutheastCon Hardware Competition.

Conclusion

Due to the modular design of the robots, parallel development was made possible and proved to be an effective method for design, fabrication, and testing. Through parallel development, the workload was efficiently distributed between team members, allowing for sub-teams to be formed and holding each responsible for making their robot work by the competition. Each sub-team was able to design, build, test, and make alterations as needed on their own timeline, within the overarching deadlines, without affecting the progress of the other robots. The goal of modularity also meant the robots were designed to have easily replaceable parts. One group also designed and built the interchangeable chassis on which the challenge portions of the robots were mounted. The interchangeable chassis made it so the stage one, two, and three sub-teams only needed to focus on the creation of the top portion of the robot used to complete the challenge. The interchangeable chassis proved to be beneficial in the fabrication process, and during the competition, when a backup chassis was utilized to allow the stage three robot to compete. All communication between the robots was through serial peripheral interface (SPI) using telephone cables. This was an effective method in which the tower handled all communication: when to start the match, immediately stop all the robots if needed, the code from the stage one robot, or the current status: navigating, completing the challenge, or finished. Through the parallel development of the autonomous robots with a focus on modularity, the team received fifth place overall out of forty-two collegiate teams in the 2017 SoutheastCon Hardware Competition.

Acknowledgements

The authors wish to express their appreciation to the University of North Carolina at Asheville Mechatronics Engineering faculty. In addition, the authors wish to express their appreciation to Jeremy Brown, who was the faculty advisor.

References

- [1] "Hitachi HM55B Compass Module (#29123)." <http://users.ece.utexas.edu/~valvano/Datasheets/HM55BModDocs.pdf>.
- [2] "Parallax HM55B Compass Sensor." Arduino Playground - HM55B. <http://playground.arduino.cc/Main/HM55B>.
- [3] Radford, Rodney. "Southeastcon 2017 Hardware Competition Rules." IEEE SoutheastCon 2017. <http://sites.ieee.org/southeastcon2017/student-program/student-hardware-competition/>.