Slugs of the Caribbean - Autonomous Robot  
*UCSC, CE118 Final Project*

**Introduction**

The Slugs of the Caribbean project consisted of designing, constructing, programming, and testing an autonomous robot to meet the minimum specifications for the CE118 final project:

* + Navigate to the opposing island and back within two minutes
  + Hit the opponent with at least two ping pong balls
  + Stay within the field demarcated by two inch thick black tape
  + Detect and resolve collisions with an immovable object
  + The robot must be fully autonomous and cannot be controlled remotely

The project was assigned and completed within a five week period, and proved to be very challenging and time consuming. It taught us many new skills that drew from a variety of disciplines including: mechanical, electrical and computer engineering. Our goal was to develop the simplest robot possible, and later improve its capabilities as much as the remaining time allowed. Not everything worked out exactly as we had planned, but the end result successfully completed the minimum specification trial run with fifteen seconds to spare.

**Designing and Planning**

The three of us began the project by contributing five unique designs each, for a total of fifteen designs. The final design we chose to pursue was a combination of each of our best ones. However, we simplified the ping pong ball launcher into a box with a hinged gate that sprung open when released by an RC servo. Originally, we were designing a modular solution that would allow us to swap out the box for a rotating turret if we had the time to do so, which was not the case.

Both designs can be viewed below in figure 1.

<Minspec and turret schematics from first week>

A good portion of our time was spent in SolidWorks creating a comprehensive CAD model of our robot. Although we prototyped our design with foamcore, the final product was cut from MDF, and joined together using the tab and slot method. The CAD design of our robot is attached and can be seen in appendix A. A good amount of research and planning went into the actuators and sensors on our robot.

Drive motors

In order to make the robot mobile we purchased two 12 V, 140 RPM gearhead DC motors, which were screwed in and fixed to the body platform by motor mounts that we designed. These mounts consisted of a thin acrylic faceplate and MDF top and sidewalls. The motor shafts were attached to two wheels, which were made from thick acrylic, while the back and front of the robot was supported by skid plates created from ¼” thick bolts.

Collision sensors

Our durable collision sensors (bumpers) were made from thick acrylic plates that we cut and bent at ninety degree angles. The flat bumper plates were attached to the bent pieces with machine screws and nuts, and tension was maintained by short springs. Small push buttons were attached behind the bumpers to allow our robot to detect collisions.

Gate release

We used a servo motor as a gate release. When the servo was at the neutral position the gate would remain closed. Two springs were mounted behind the gate, and when the servo head was sent to the -45° position the gate was free to spring open. A motor mount was made for the servo out of MDF and attached just behind the gate.

Infrared Sensors

The infrared sensors that we designed and soldered in lab 2 proved to be robust enough to use on our final robot. We experimented with switching between two photodiodes on one circuit we elected to use two separate circuits detecting signals from two photodiodes oriented at an angle to allow for both long and short range detection. Each circuit was modified slightly to take an output from the peak detector allowing for software hysteresis. Both photodiodes were placed in MDF shielding to block light entering from the sides and give good directionality to our main beacon.

Tape Sensors

We used 7 of the TCRT5000 tape sensors provided in our lab kit for navigation of the course. Our arrangement allowed for high resolution coast following on the right side as well as detection of corners and the enemy base, with one sensor in the back to prevent reversing onto land. The ULN2003A board provided to us proved invaluable for simplification in driving our tape sensors, allowing for all 7 sensors to be switched easily using 3 ports on the PIC32. A perfboard with all our inputs and outputs as well as power and loading resistors for each LED/PD on the tape sensor had to be carefully soldered (twice) in order for our sensors to work properly.

Once a majority of the design work was completed we began by testing our design and creating prototypes of the actuator and sensor elements individually.

**Prototyping and Modeling**

To prototype our actuators and sensors we used foamcore instead of the more expensive MDF. Aside from prototyping the launcher and ‘geneva wheel’ loader for the turret design, which were never implemented, the first element we tested were our collision sensors.

us over the course of this project The microcontroller used by a PIC32MX320F128H donated by MicroChip.

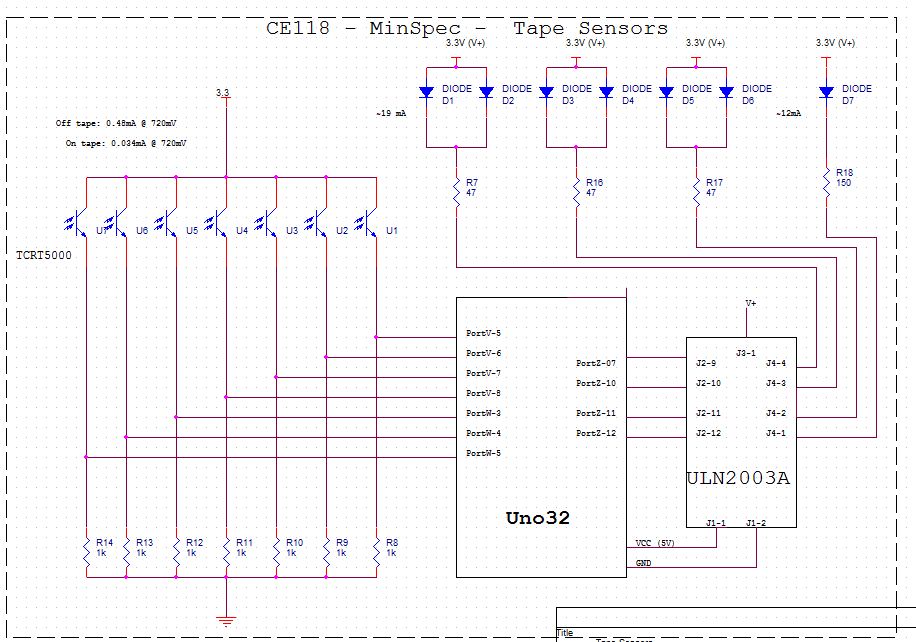
…

**The Tape Sensors**

Our tape sensor implementation is fairly simple on its surface; however once we began to solder up all of the components its complexity became apparent. Each tape sensor has 4 leads, cathode and anode for the LED as well as collector and emitter for the photodiode. The LED anode and phototransistor cathode leads went into a board which provided 3.3V power from a TIP122 voltage regulator. Each LED needed a 150ohm loading resistor and was grounded with the ULN2003, capable of sinking current from two LEDs at each port. The photodiodes were placed in a sourcing configuration with a 1K resistor to ground with output was taken at the emitter using the ‘W’ analog port on the PIC32. The LEDs can be switched off and on by switching the ports on the ULN using the ‘Z’ ports and we then cascade through our analog inputs to take tape sensor readings.

As we soldered together our tape sensor board it quickly turned into a rats nest of wires leading into a small perfboard. 4 leads for each sensor \* 7 sensors + 4 leads to the ULN meant we had 42 wires coming out of our perfboard, not to mention resistors and power regulation. This board was made even denser by the addition of bump sensor circuitry to the same perfboard. Despite the complexity our first assembled board worked reasonably well and we were able to use our sensors to design and implement our tape sensor code. Unfortunately with only a few days before min spec two of our tape sensors started to behave erratically, with no obvious software or hardware solution. After probing our tape sensor circuit for several hours we determined that there was no recourse but to resolder the whole board. Thankfully professor Elkaim was able to step in help us solder up the new board in record time.

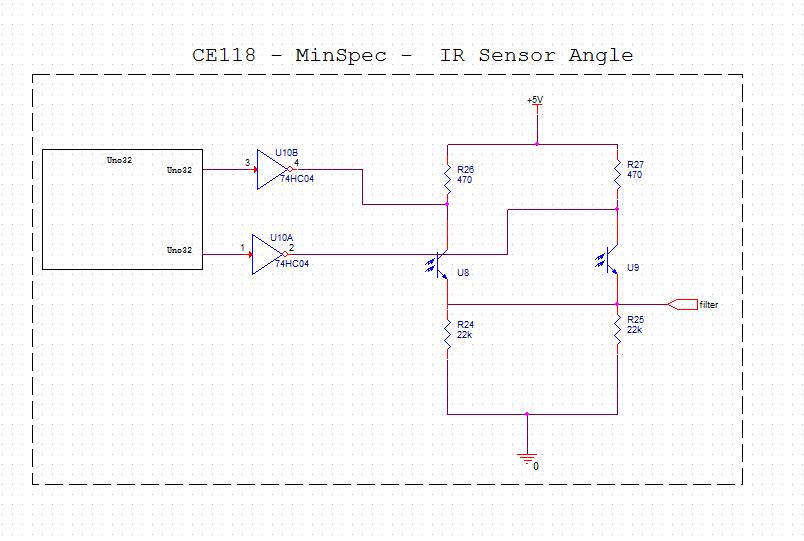
The redone tape sensor board worked great, and all of our sensors gave readings close enough together that we did not have to deal with the additional complexity of separate thresholds for each sensor.



The software we used for the tape sensor made use of a simple state machine and a **[synchronous?]** sampling counter to ensure we had accurate readings at all times. Our state machine turned the leds on/off, took four readings, averaged them, repeated the process, and takes the difference of ‘ON’ and ‘OFF’ states to get a reading. High values correspond to being off tape, and low on tape. In practice we ended up taking only two samples during each state to speed up our sampling rate which in turn allowed us to navigate the course slightly faster. <insert state machine diagram?>

**The Infrared Sensors**

From very early on we wanted a two infrared sensor arrangement to allow for both long and short range detection of enemies, the idea being that we lock on with the main beacon and a second one placed at an angle notifies us when we are within range to dump / charge ahead over tape. Our original idea to save space on our bot and (in theory) reduce complexity was to switch two photodiodes on/off on the same circuit using a hex inverter. While we were able to implement a circuit that switched between two detectors on the same circuit we had nothing but trouble with interference between the two detectors. After working with the circuit trying to determine which capacitors weren’t discharging or where our initial voltage spikes were coming from we elected to cut our losses and place an additional sensing circuit on our board. While this detracted from the simplicity of our design it did not cost us any I/O ports and worked well with some minor alterations.

We first thought that we could get by with taking output from the comparator of the two circuits, saving us some software complexity, but quickly abandoned this idea in favor of a software comparator. The hardware comparator did not allow us much in the way of accuracy and our final navigation made use of a find max IR signal function which would not have been possible without taking output from our peak detectors. In order to get good resolution off our peak detector we had to alter it’s time constant for a faster discharge with a smaller resistor.

<switch schematic>

<David & Justin’s Schematics in appendix>

**The Collision Sensors**

In theory we wanted very robust bump sensors to allow us to ram into the other robots without sustaining any structural damage ourselves. In practice we ended up with some exquisitely designed bumpers that were perhaps somewhat overkill for our slow-moving robot.

Our durable collision sensors (bumpers) were made from thick acrylic plates that we cut and bent at ninety degree angles. The flat bumper plates were attached to the bent pieces with machine screws and nuts, and tension was maintained by short springs. Small push buttons were attached behind the bumpers to allow our robot to detect collisions.