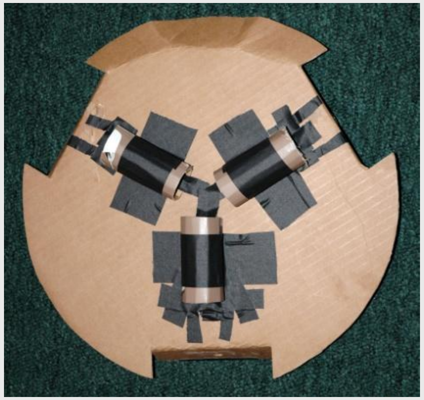
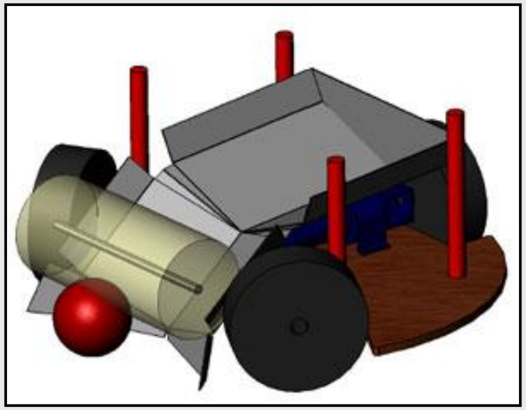
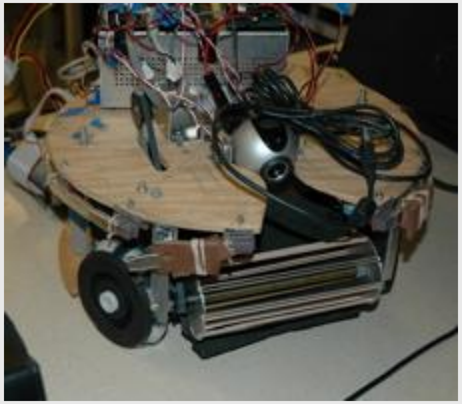
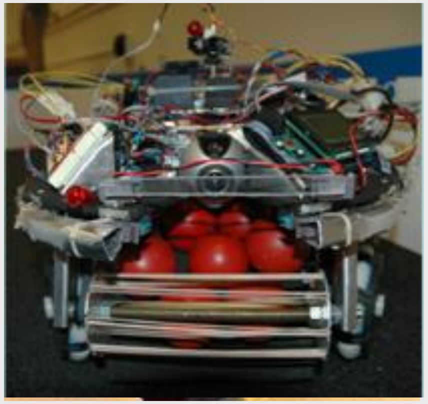
   

Strategy and Mechanical Design

Maslab IAP 2009

Ellen Yi Chen
Chris Celio
Kimberlee Collins

Agenda

- Strategy
  - Schedule
  - Systematic Strategy Selection
  - Case Studies
- Mechanical Design
  - Design Process
  - Mechanisms
  - Resources and Tools

Build Schedule

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
4	5 **Start!** Assignment 1	6 Assignment 2	7	8 Assignment 3	9 Assignment 4	10
11	12 **Mock # 1**	13 ITA Dinner	14	15 Assignment 5	16 **Mock #2**	17
18	19 MLK	20 Sponsor Dinner	21	22 **Mock #3**	23 Assignment 6	24
25	26 **Mock #4**	27 Assignment 7	28	29 **Impounding**	30 **Final Competition**	31 Cleanup Day

Mechanical

Design / Feasibility Tests / Prototype

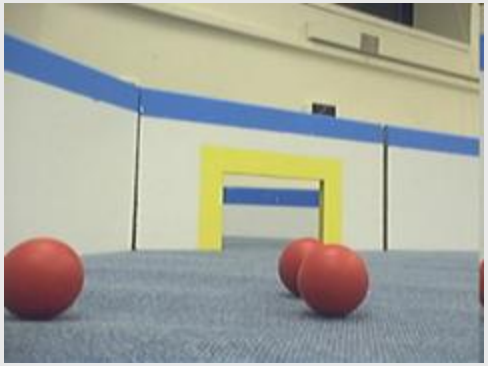
Build

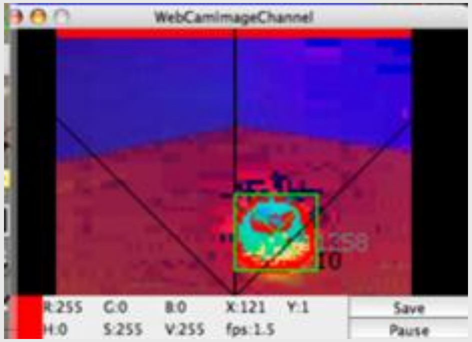
Debug

Fail Week!

Build Schedule

- Mock #1
  - Drive, maybe navigate
  - Test your color recognition / vision algorithm
  - Take lots of pictures of the field (lighting in 26-100)
  - Find field features your robot might have trouble with
- Mock #2
  - Navigate the field
  - Find balls and goals
  - Maybe pick up balls
- Mock #3
  - Pick up balls
  - Score Points
  - Mechanical feature freeze
- Mock #4
  - Dress Rehearsal
  - No big changes

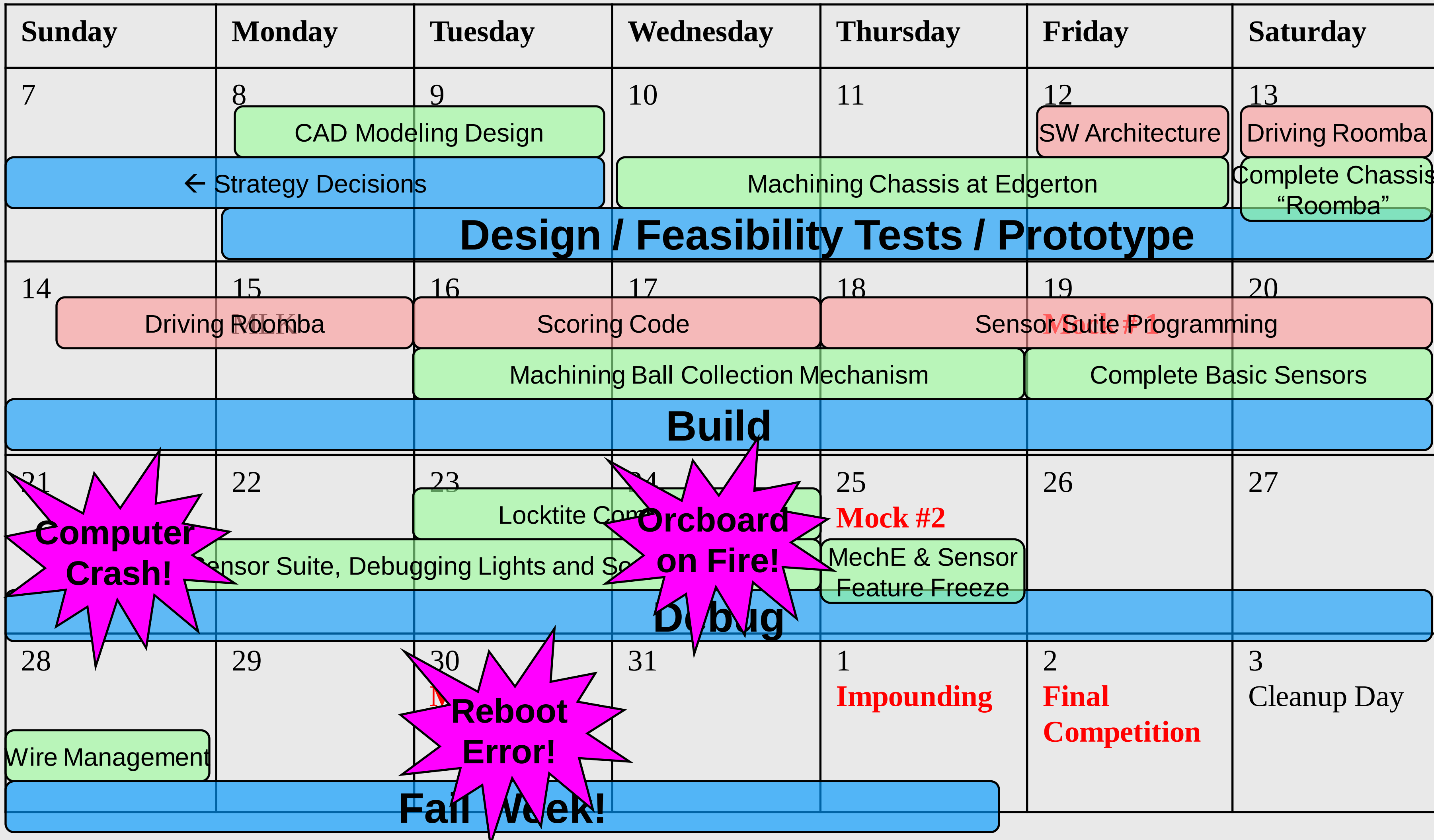




Detailed Schedule

(Based on Team 12AW12 in 2007)

- Design Stage
- MechE and Sensors
- Software



Scheduling Summary

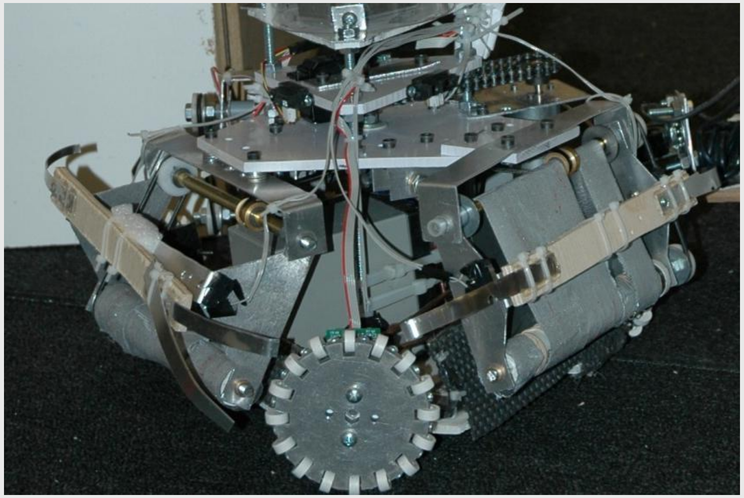
- Two weeks to build
- Keep your programmers happy!
  - Stagger work schedules:
    - Build by day (machine shops open)
    - Code by night
  - Get them food
  - Give them a working robot at all times!
- Expect failures so leave room in the schedule

Agenda

- Strategy
  - Schedule
  - Systematic Strategy Selection
  - Case Studies
- Mechanical Design
  - Design Process
  - Mechanisms
  - Resources and Tools

Strategy

- What do we mean by strategy?
  - Design (mechanical, software) chosen based on available resources to achieve a goal (hopefully to win Maslab)
- Systematic strategy selection*
  - Functional Requirements
  - Design Parameters
  - Analysis
  - References
  - Risks
  - Countermeasures



- Why spend time teaching this?
  - 52% of you have never built a robot
  - Don’t just build something because its cool! This goes for the MechEs especially!

*Also known as FRDPARRC by Professor Alex Slocum

Strategy

Functional Requirements

Win Maslab

- Drive Around
- Find Balls
- Collect Balls
- Find Goals
- Score Balls

Collect Balls →

	Claw	Roller	Active Gate	Passive Gate
Design Parameters	Grabs balls and deposits them in a bin above the robot	Rolls to collect and keep balls. Keep balls in a raised bin	Opens gate and drives forward to collect and keep balls	A bin area to hold balls, no mechanical parts
Analysis	How many balls? How many joints/actuators? Power needed to grab and lift?	How many balls? Motor torque needed to lift balls up ramp?	How many balls? Actuator power needed?	---
References	The DLB Maslab 2006	HMS Velociraptor Maslab 2008	Curious George Maslab 2007	Four Play Maslab 2005
Risks	Mechanically complex, collect only one at a time…	How to separate red balls from yellow?....	How to separate red balls from yellow?...	Could lose ball out the front, how to score?....
Counter Measures	•Use prebuilt servo claw kit	•Two rollers •One roller with separation in bin	•Collect 1 color only •Collect 1 color for first half and other color for second half	Only drive forward, drive backward only to get ball out

Pros/Cons to decide? Further Analysis Needed? More Brainstorming?

Strategy: Pugh Chart

- Used to select a strategy
- Choose a datum (simplest) strategy
- Choose criteria (time, difficulty, accuracy)
- Weight your most important criteria
- Score your strategies
  - "+" if its better
  - "-" if its worse
  - "0" if it is as good as your datum
- Add up the totals

Drive Strategies	Datum: Standard Two wheels and caster	Option2: Bigger wheels	Option3: Omniwheels
Coding difficulty	0	0	-
Building difficulty	0	-	--
Field Time (2x)	0	+	-
Navigating (2x)	0	-	+
Odometry	0	-	--
Total	**0**	**-2**	**-5**

System Strategy

- Design a system not parts!
  - Top Down
  - Bottom Up
- Resources
  - Time/People
  - Shop Access
  - Experience
- Areas
  - Navigation
  - Driving Around
  - Vision
  - Ball Capture/Deposit

Mechanical Strategy

People: 1 coder, 3 mechEs

Access: Maslab shop, Edgerton, CSAIL, Papallardo

Experience: Built and coded robots before

Strategies

Navigation: Bump sensors and wall following

Driving: omniwheels

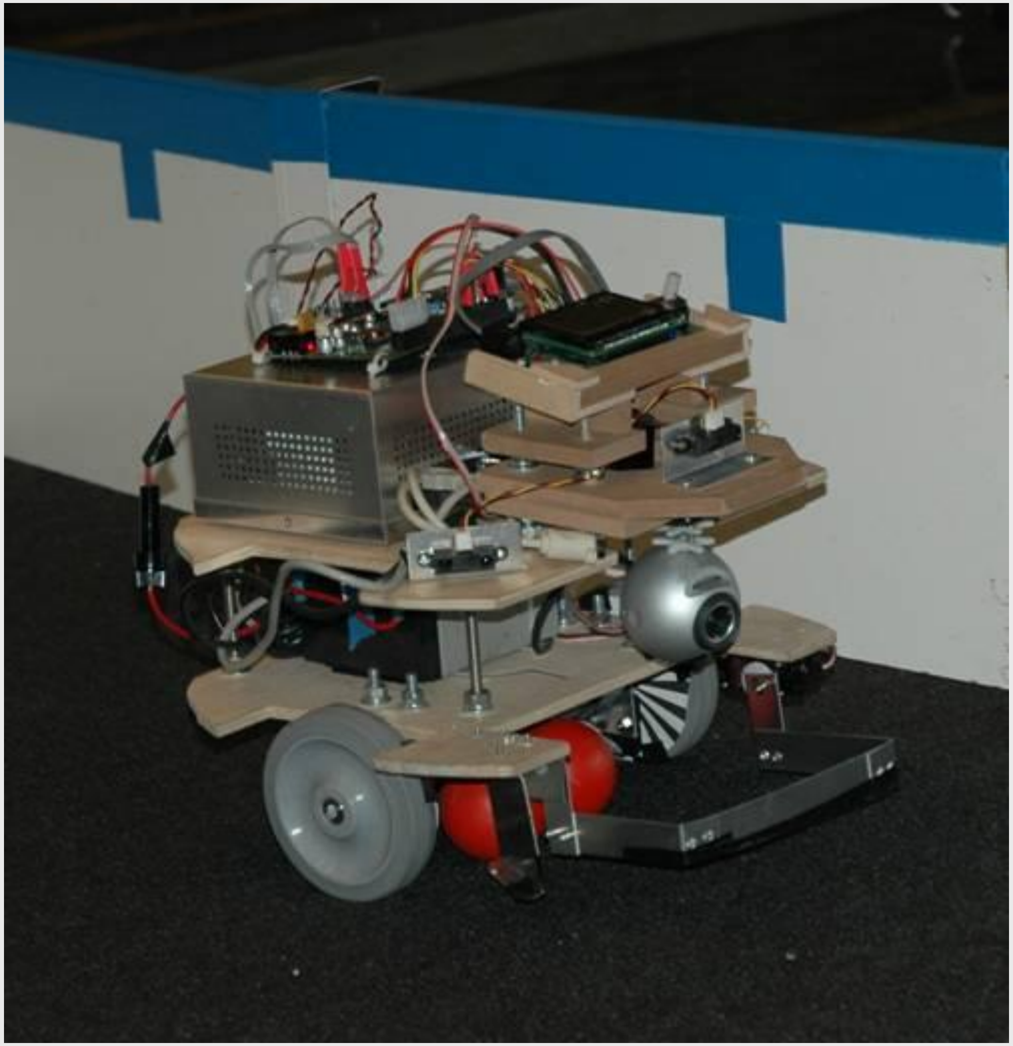
Vision: Rotating camera, Color detection only

Ball Handling: Two roller systems

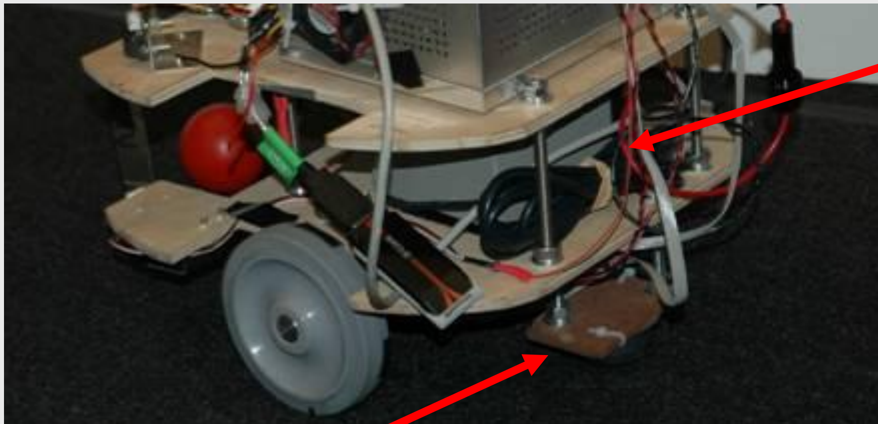
Agenda

- Strategy
  - Schedule
  - Systematic Strategy Selection
  - Case Studies
- Mechanical Design
  - Design Process
  - Mechanisms
  - Resources and Tools

Case Studies: Team Yellow Hat

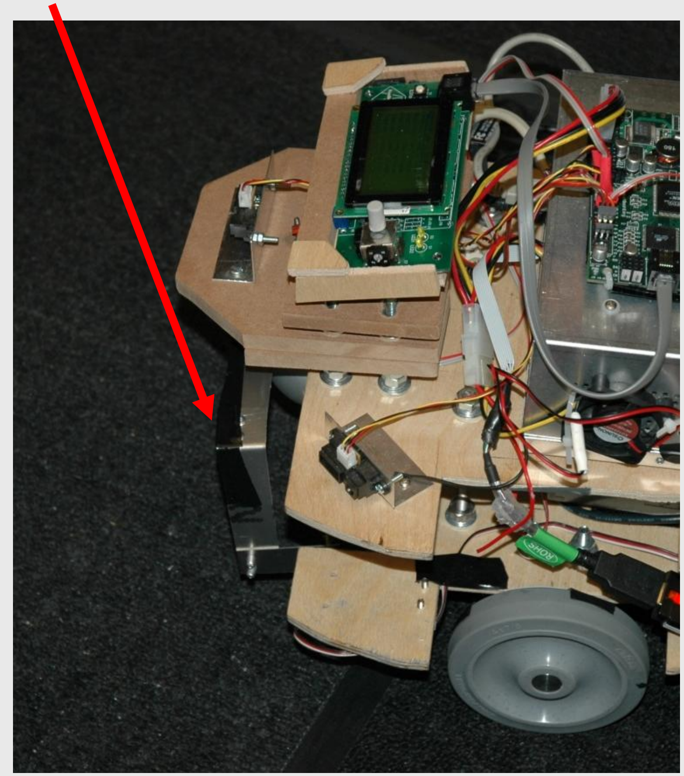


Team Yellow Hat - Features

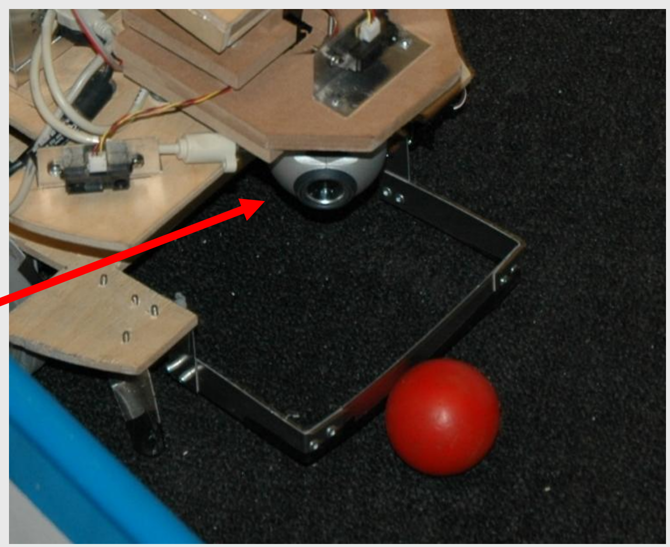


Battery – Center of Mass issues

Ball Gate



Back Caster



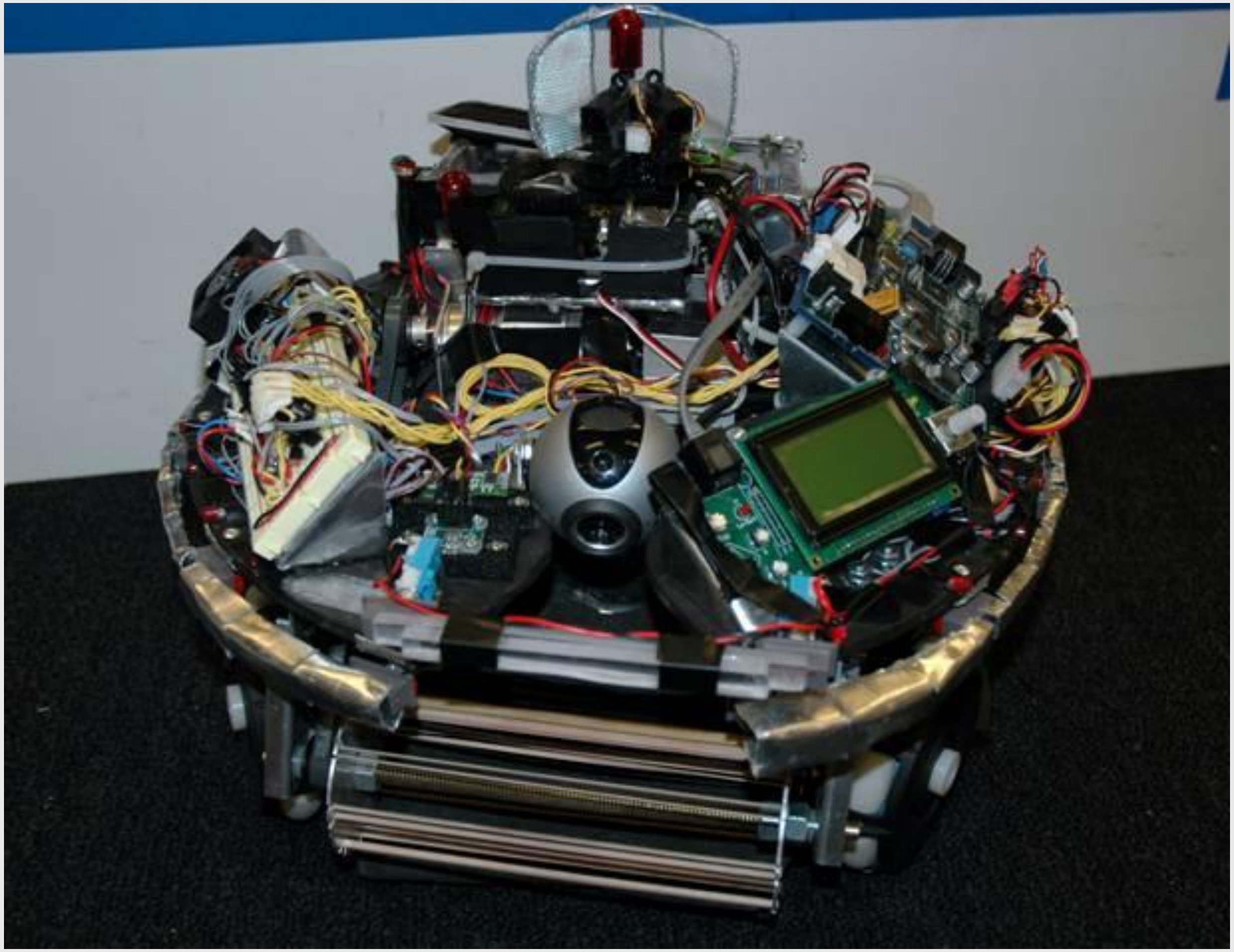
Rotating Camera

Team Yellow Hat's Advice

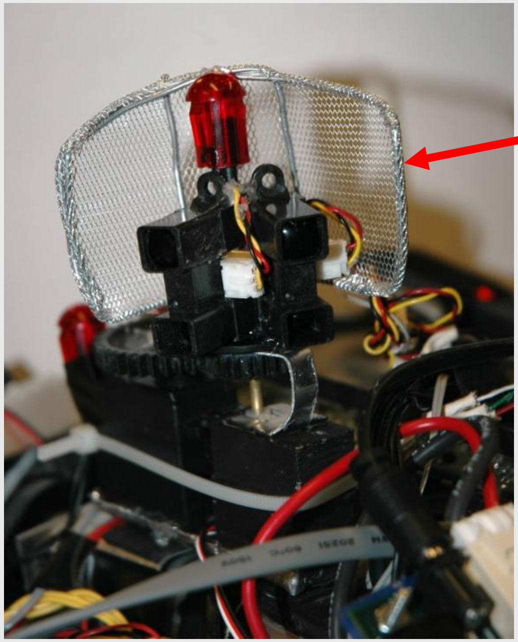
- Curious George was designed on the premise that simple behaviors are more reliable than complex ones. As such, any complex behavior, like mapping, must be backed up by simple behaviors, like wall following. **Our plan was to build a robot that could reliably wander, collect balls, and deposit in goals. Once this was complete, we would attempt to add mapping capabilities to provide more intelligence while wandering.** Alas, we never accomplished the latter, but our focus on reliability paid off with 19 points and a win at the final competition.

http://maslab.mit.edu/2007/wiki/Team_Fourteen/Final_Paper

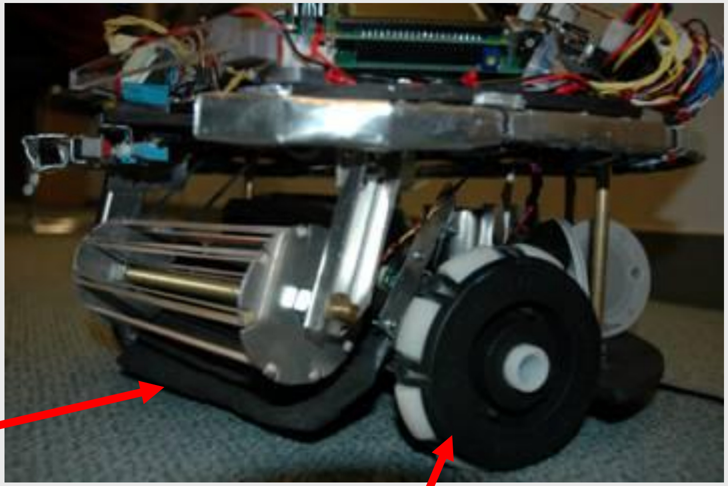
Case Studies: Team 12AW12



Team 12AW12 - Features

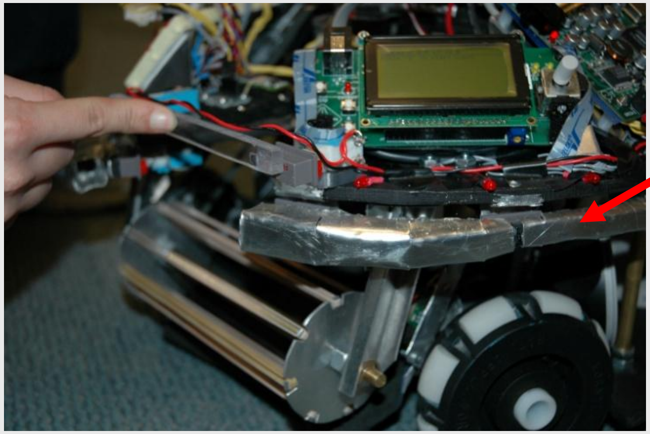


SSS
(Swiveling
Sensor
Suite)

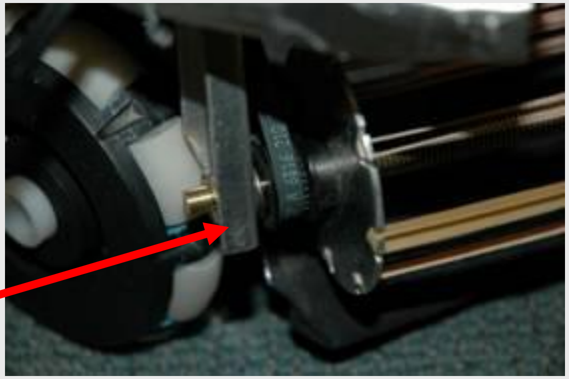


12 Ball
Capacity Bin

Omni-Wheels



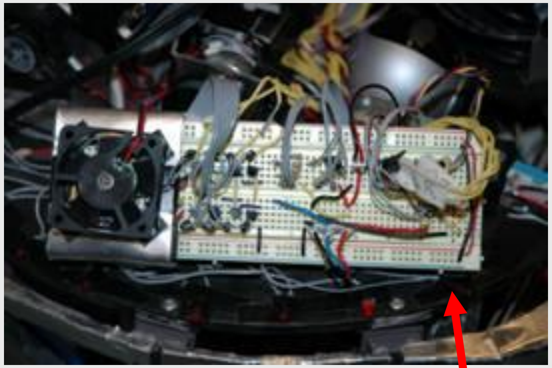
360° Bump
Sensor
Coverage



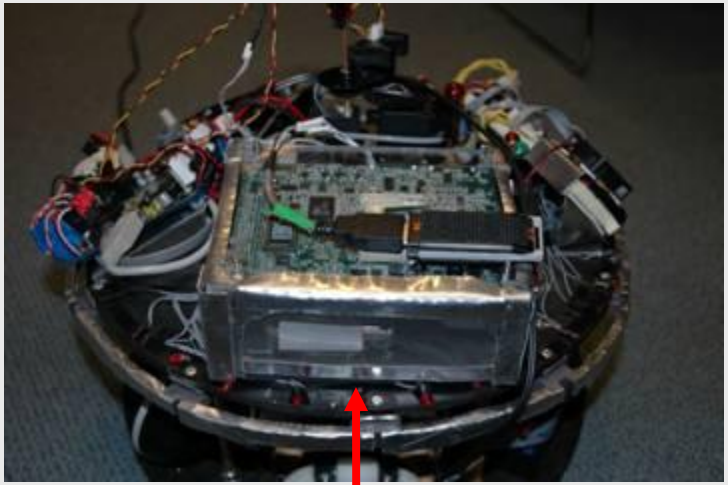
Roller and

Belt Drive

Team 12AW12 – Features

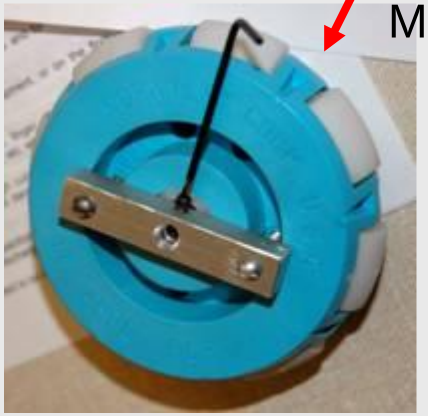


Breadboard Driving
LEDs and Fans



Small, Custom
Computer Box (the
stock box blocked the SSS)

Wheel
Mounts



Speakers (played R2D2
sounds. Useful for
debugging state changes).

Break Beam

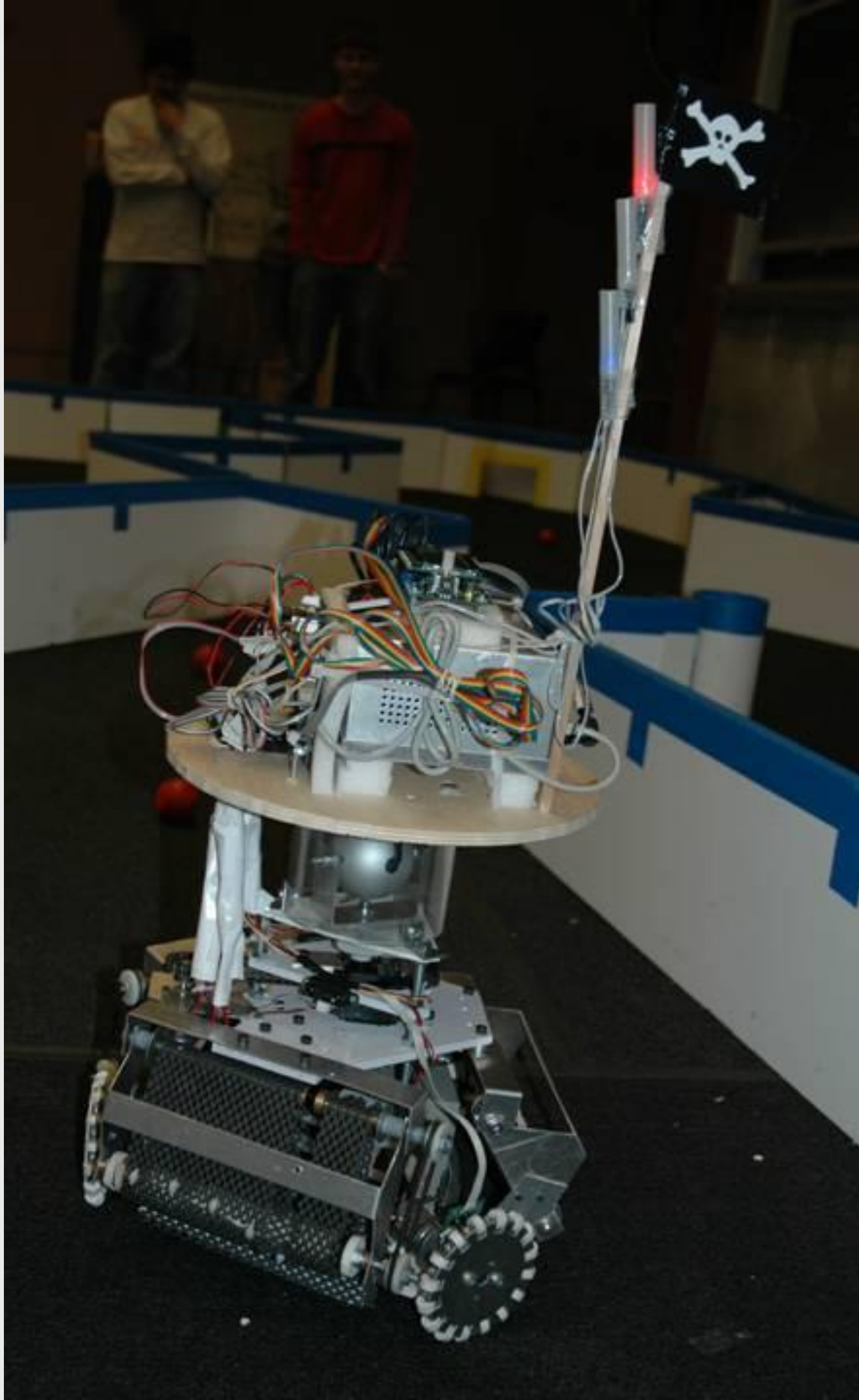
Team 12AW12's Advice

- **Modularity** is key! Design it to be easy to take apart and easy to take on new functionality.

- Invest some time (and man-power) into the mechanical side. Make a strong push to finish the robot ASAP (try to mostly finish it in the first week). **It is hard to program when you do not have a robot, so make some of the programmers build too.** Do not be afraid to make something a little more complicated mechanically if it will make things easier to code!

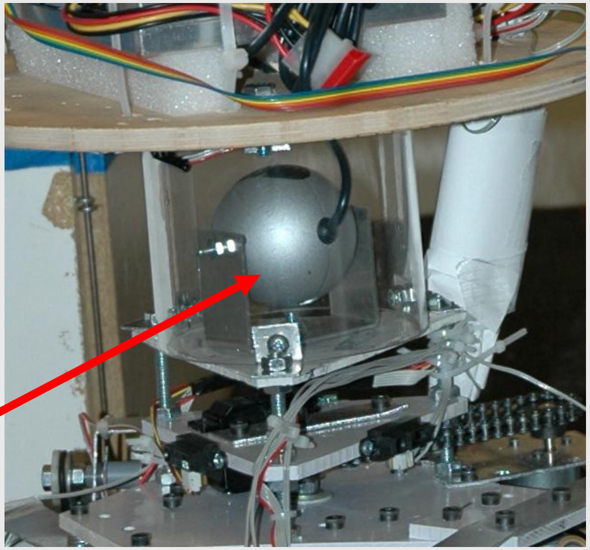
-http://maslab.mit.edu/2007/wiki/Team_One/Final_Paper

Case Studies: Team "Pieces of" Eight

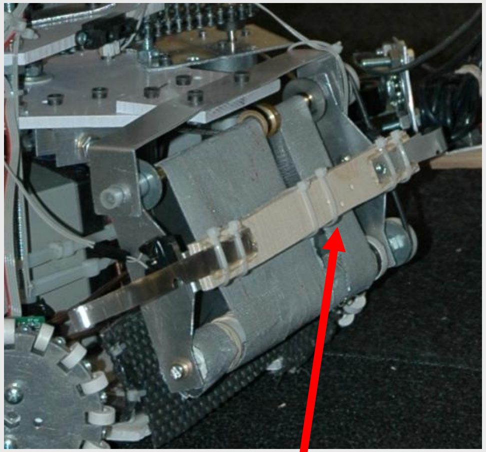


Team “Pieces of” Eight - Features

LED Lights for Debugging

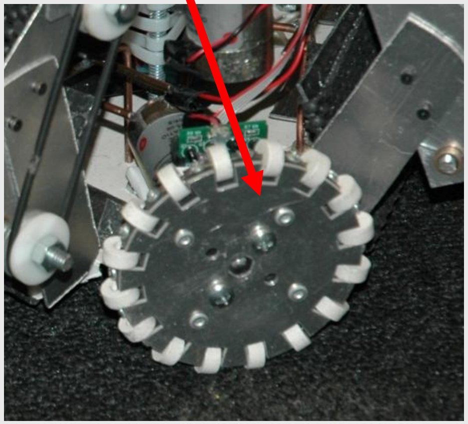


Rotating Camera

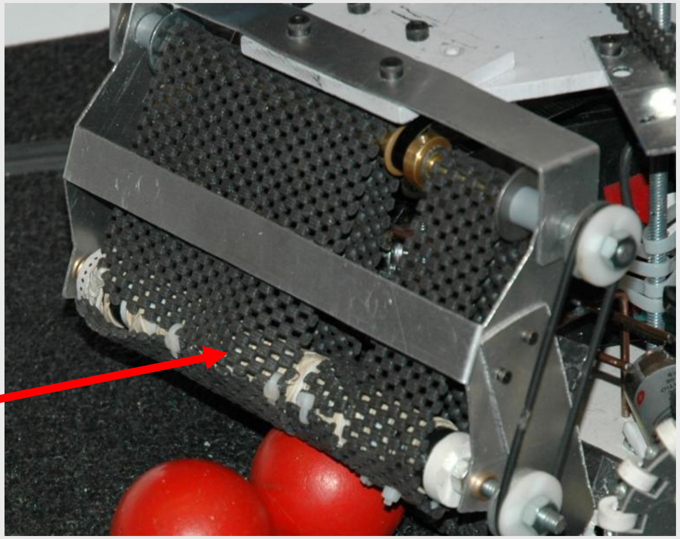


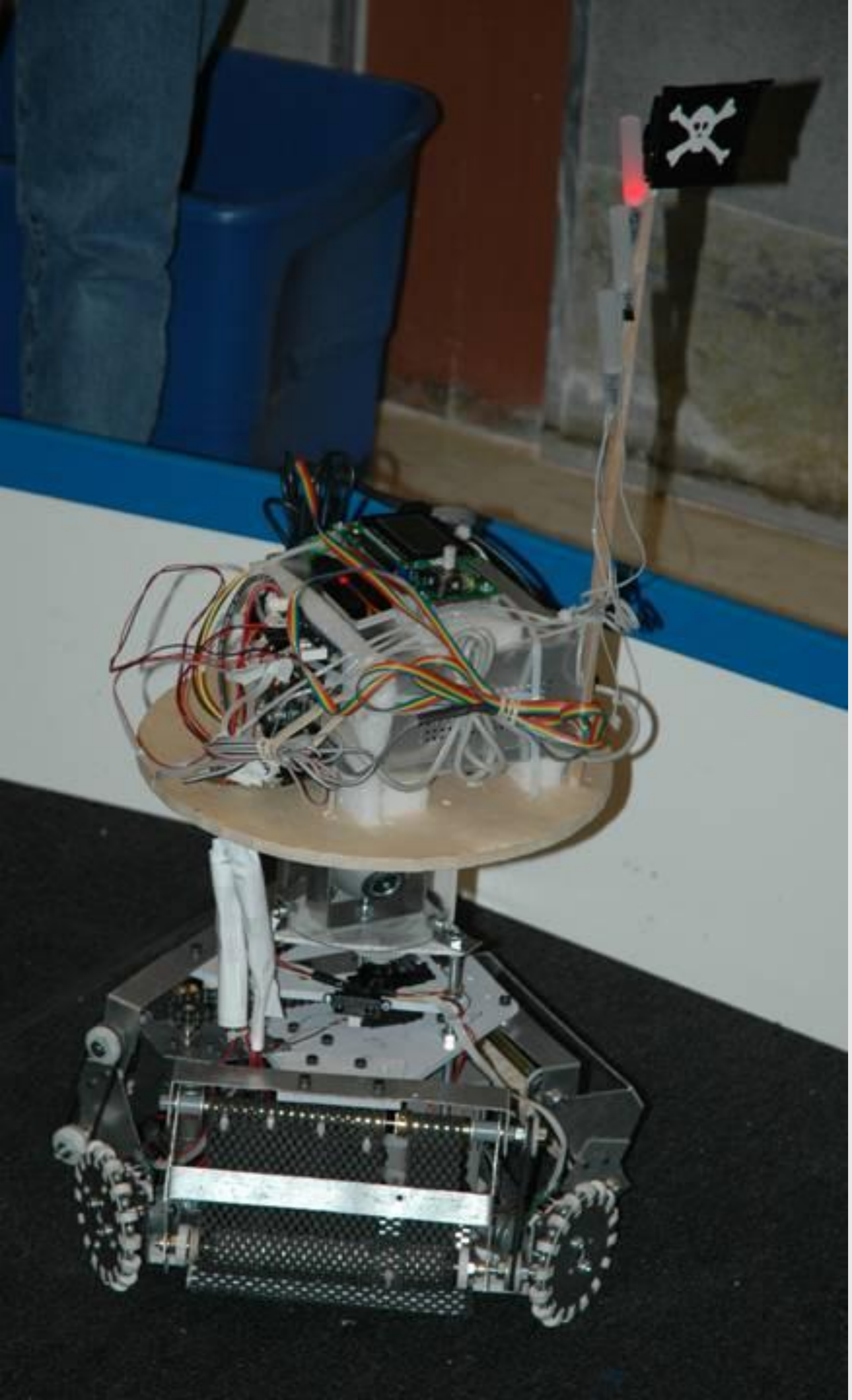
Bump Sensors

Omni Wheels



3 Rollers Powered by a Single Motor



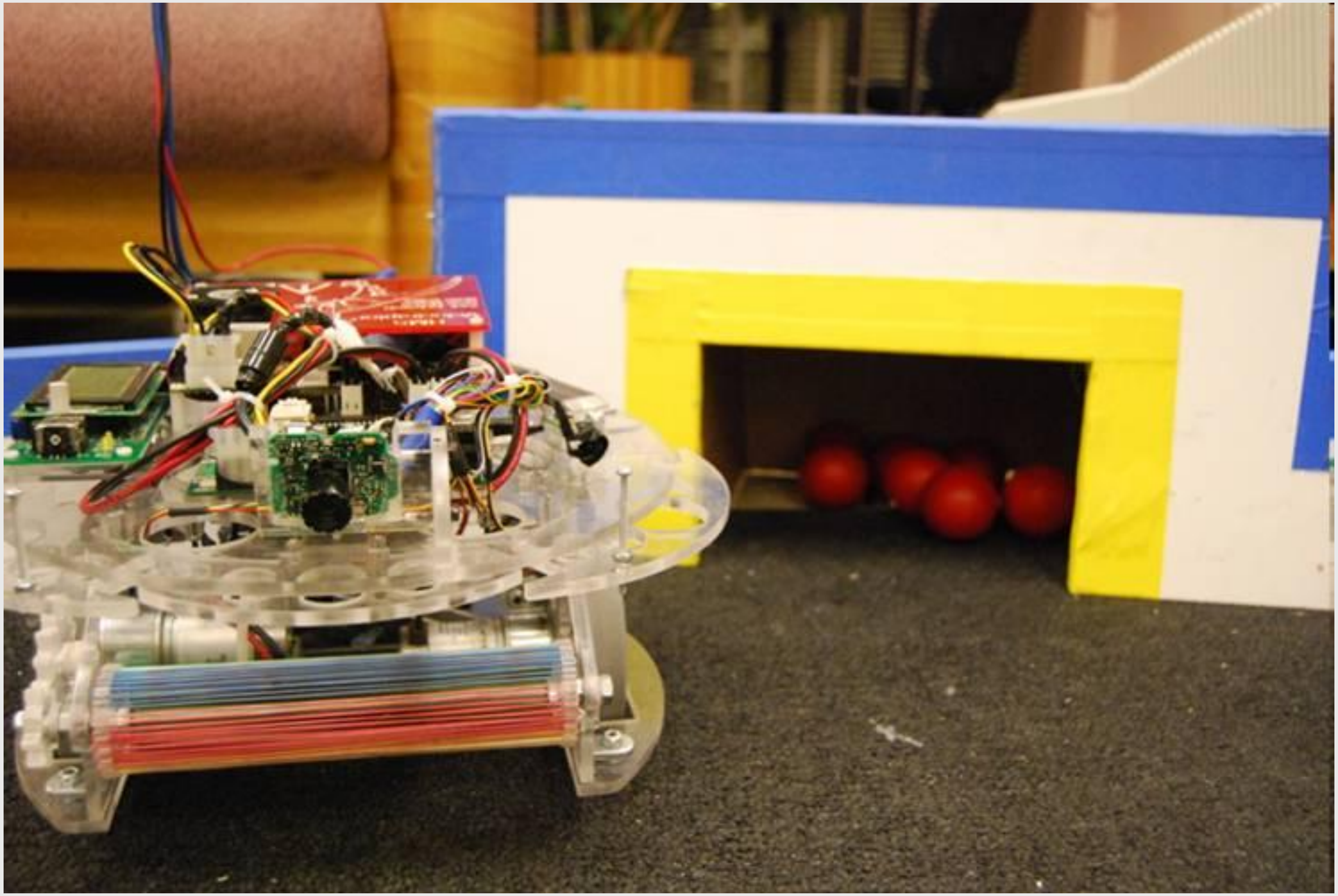


Team "Pieces of" Eight's Advice

- "When we tested the robot on the field, I feel like we were mostly testing individual pieces of the robot. It wasn't until the very end that we combined everything, and **we found that they didn't come together as seamlessly as we'd hoped.** It would have probably been better to throw everything together at the beginning and fine-tune each thing later on."

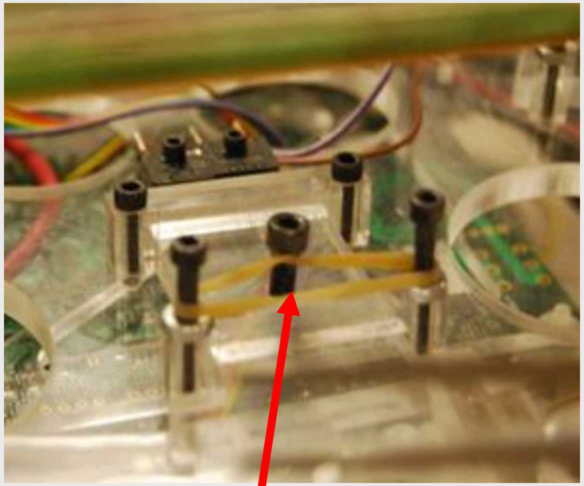
-Email from Daniel Torres, Jr., answering questions for a Maslab magazine article

Case Studies:
Team HMS Velociraptor

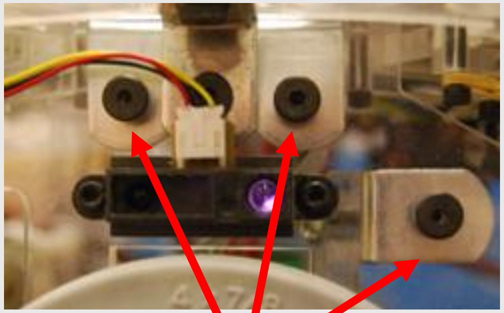


Team HMS Velociraptor - Features

Computer Cover – Prevent Short Circuits!

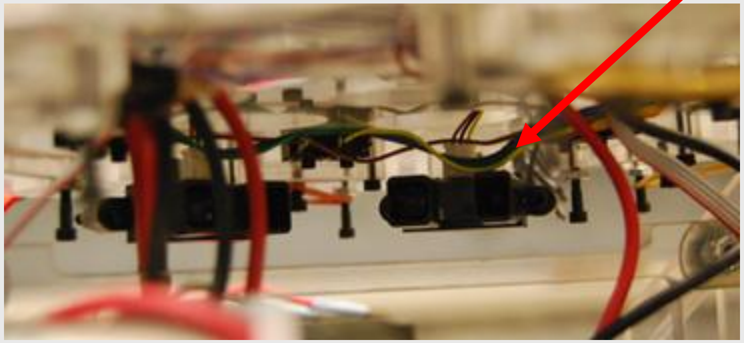


Bump sensor “return spring”

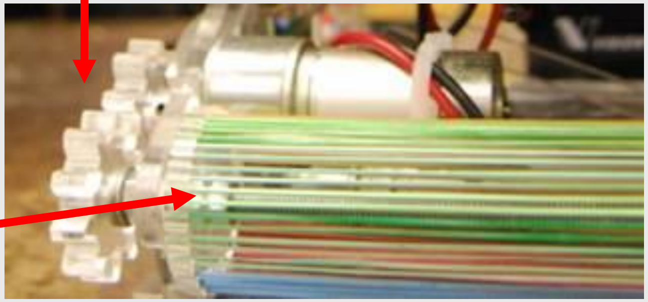


Simple standardized brackets

Compact sensor placement – also compensates for IR deadzone

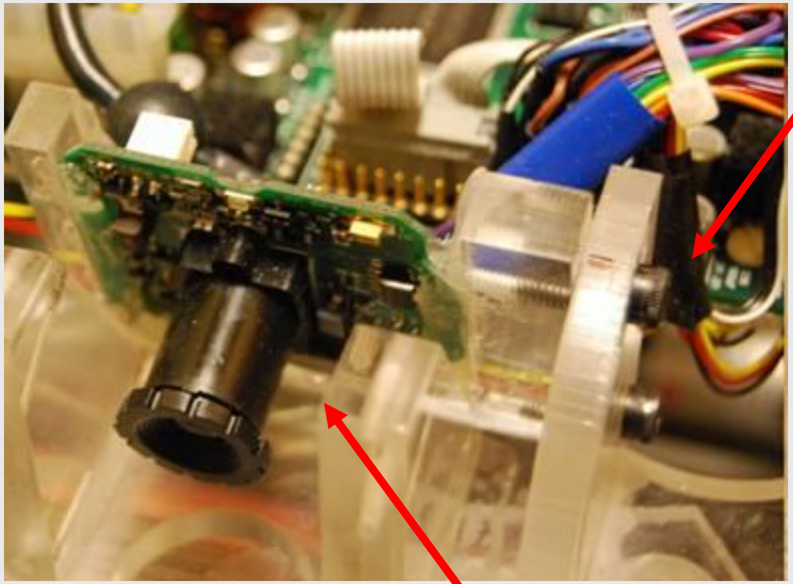


Geared drive – Note the wide gear, bearing length, and tolerance



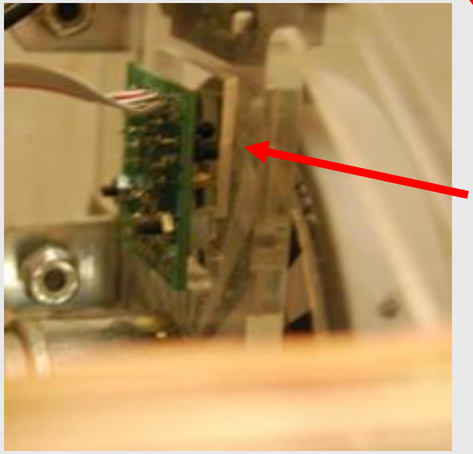
Compliant Roller

Team HMS Velociraptor - Features



Adjustable Camera Angle

Camera



Quadrature Encoder – Note the simple mounting

Staggered Bump Sensor Design

Team HMS Velociraptor - Advice

- Prioritize your development! If a feature is taking too much time, is it necessary? The goal is to win MASLAB, but that's impossible without a robot that can move around the table looking for balls and goals. **Don't lose sight of the goal for unnecessary pet features.**
- Get a robot up and running immediately. You can't debug your software without running hardware. We had the first version of our final competition bot built by the end of the first week.
- Use robust mechanical design principles. **Do as much as you can to abstract away the hardware, good software design is impossible without this.**

Strategy Summary

- Finish building as early as possible
- Plan for unexpected downtime
- Make design choices systematically
- Design a system, not parts.

Agenda

- Strategy
  - Schedule
  - Systematic Strategy Selection
  - Case Studies
- Mechanical Design
  - Design Process
  - Mechanisms
  - Resources and Tools

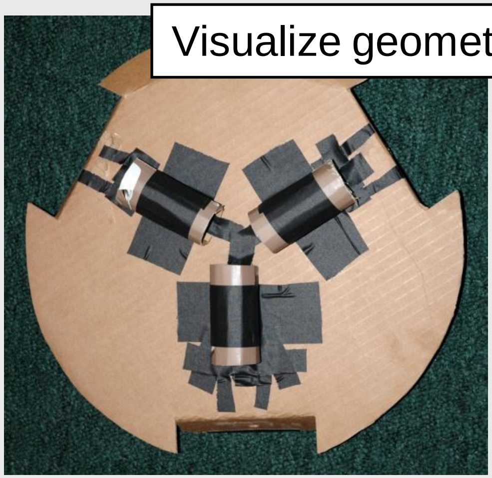
Design Process: Physical Model

<u>But Remember when going to your final machine….</u>

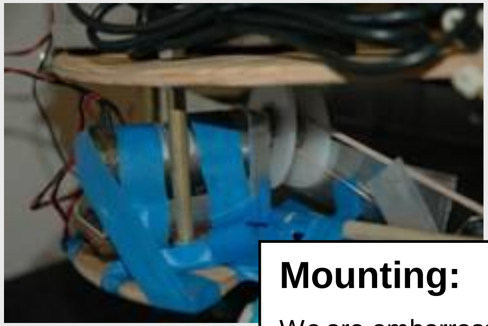
- Trial & error should occur in the design stage - not the build stage!

<u>A Slice of 2.007</u>

- Friction
- St. Venant’s Principle
- Abbe Error
- Reciprocity
- Structural Loop



Visualize geometry



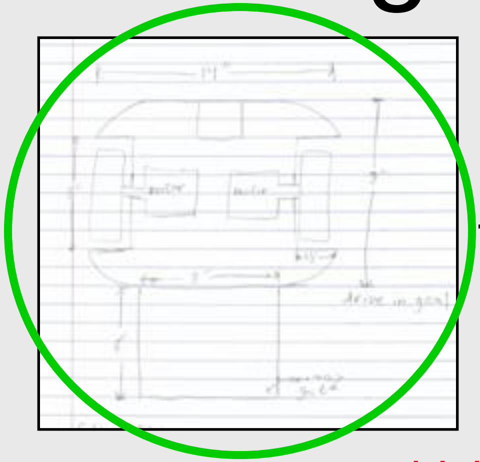
Mounting:

We are embarrassed to admit that this is our robot ^_^

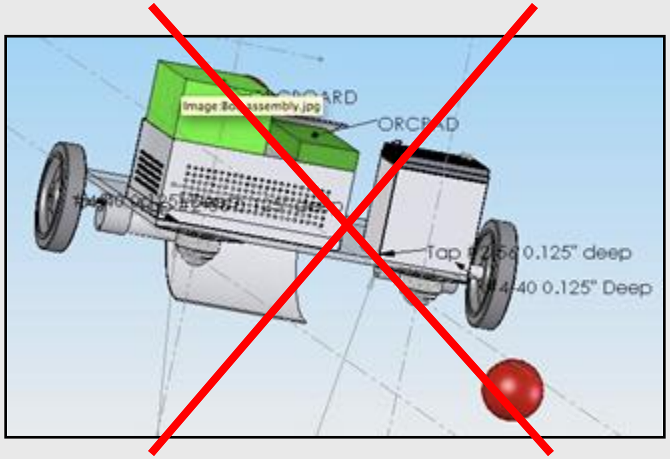
Design Process: Bench Level Testing

- Use simple physical model to test mechanism concept. Does it work like you think?
- Don’t take anything for granted, test it first!
- Key situations (and learnings)
  - Rubber Band Roller - Roller materials
    - Material must be radially compliant, tangentially high frictional constant (good grip).
  - Swiveling Camera - Servo speed
    - Servos don’t reach a desired position instantaneously!
  - Omniwheel Robot - Not all motors are created equal
    - Motors with high gear ratios have more torque but run slow
    - Motors with lower gear ratios run fast but don’t have enough torque to move your robot!
  - Etc…

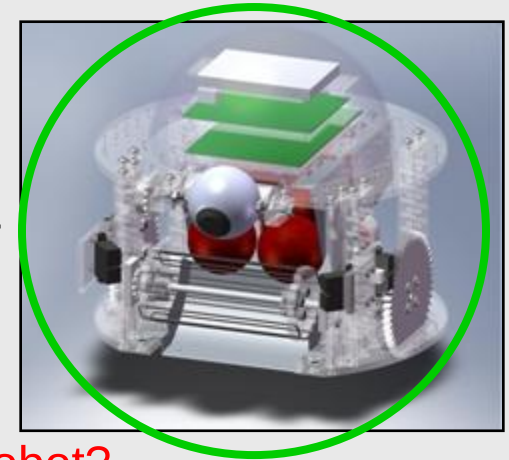
Design Process: CAD Modeling



-OR-



-OR-

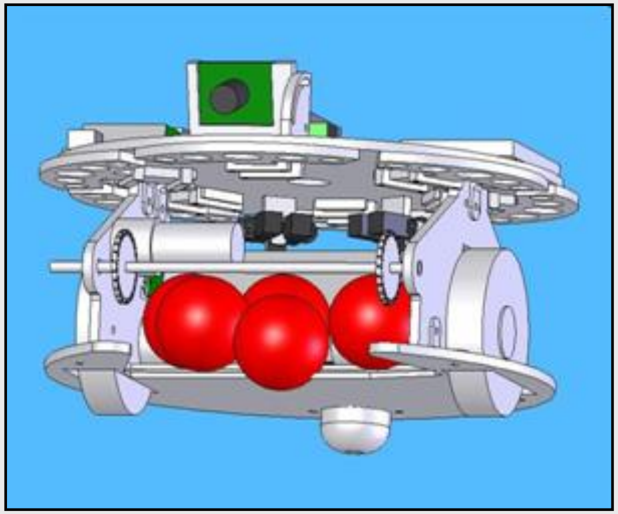


Which one is the wrong way to plan your final robot?

- Benefits
  - –Easy to visualize complex ideas
  - –Fast feasibility checks of designs
- Must CAD when:
  - –Laser cutting/Water jetting/Milling
  - –Deriving dimensions for complex geometry
  - –Does all this stuff fit in 3D?
- Don’t CAD when:
  - –You can confidently dimension it out on a single sheet of paper → Your time is better spent elsewhere! Like building!

Design Process: CAD Modeling

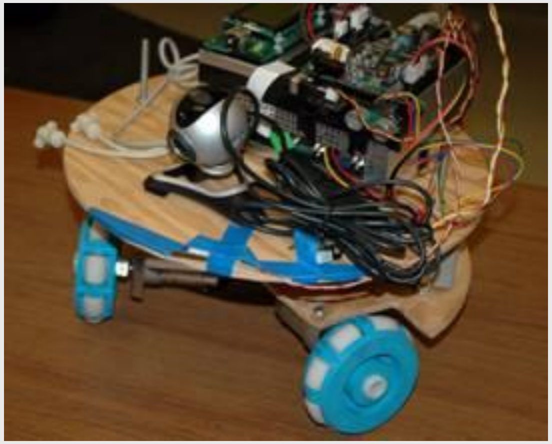
- Software
  - Solidworks*
  - ProEngineer*
  - Rino
  - Google SketchUp
  - And many more!

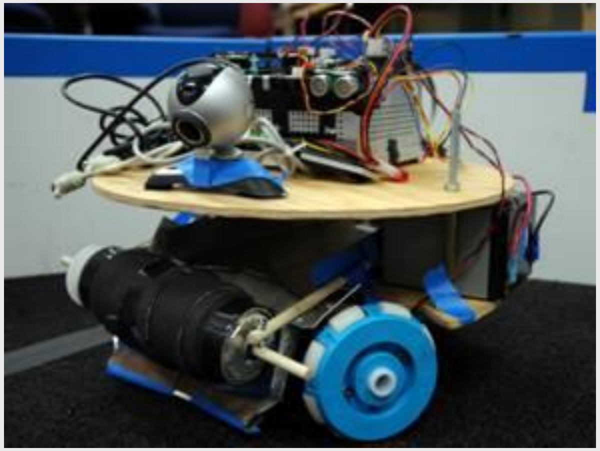


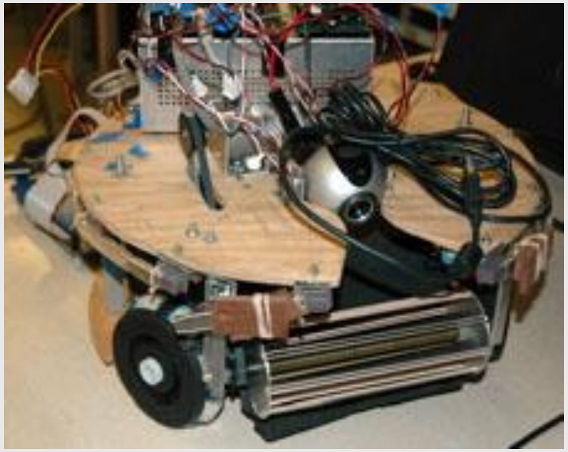
*https://meche.mit.edu/resources/computing/software/

(MIT certificates required)

Modular Design







Driving Robot

Coders can practice driving

-Day 3-

Collection Mechanism

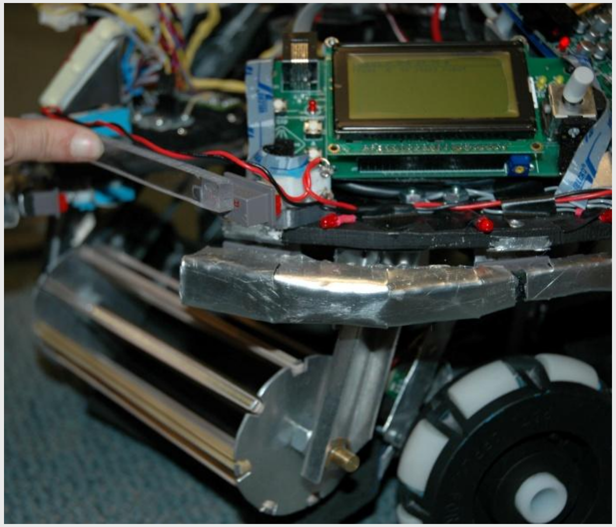
Coders can practice driving and collection

-Day 5-

Refined Collection Mechanism and Prototype Bump Sensors

-Day 11-

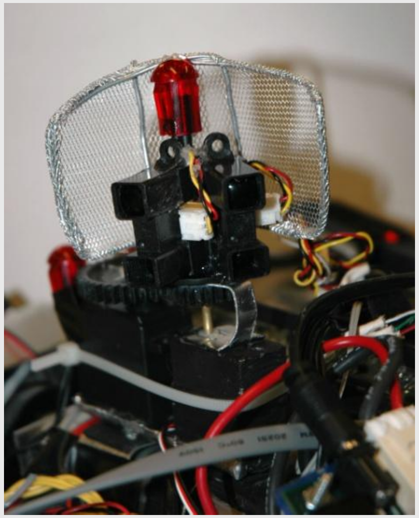
Modular Design



Bump Sensor Module

Coders can practice navigating with sensors

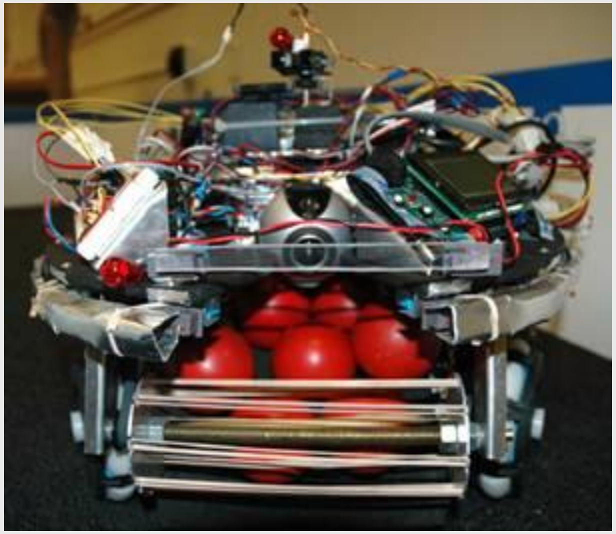
-Day 13-



SSS Module

Coders can practice navigating with sensors

-Day 15-



Full Functionality!

-Day 18-

Design

- Modularity
  - Each feature does not depend on others
  - Interchangeability → use standardized parts
- **Goal:** Working robot at all times
  - Minimize downtime
  - Identify design issues early on
  - Maximize sanity!

Design for Assembly

- Why?
  - You will take your robot apart many times when building and fixing
  - Your partners will need put it back together when you are asleep
- What?
  - Make fasteners accessible
  - Avoid Glue
  - Standardize (screws, brackets)
  - Phillips or hex head screws. No flathead!
  - Label things when you’re bored (wires, left/right motor)

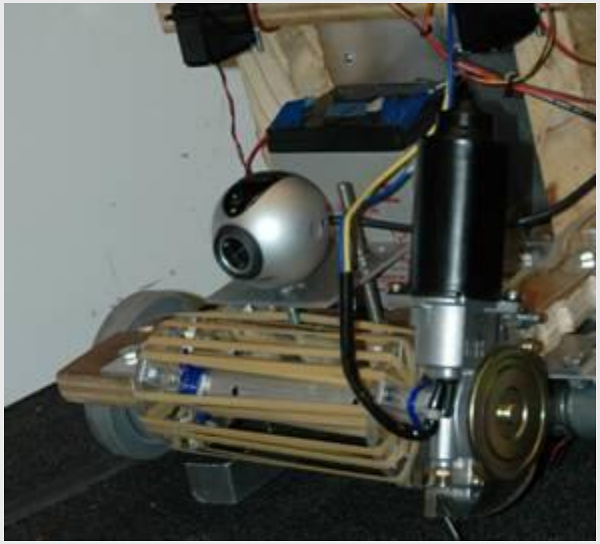
Agenda

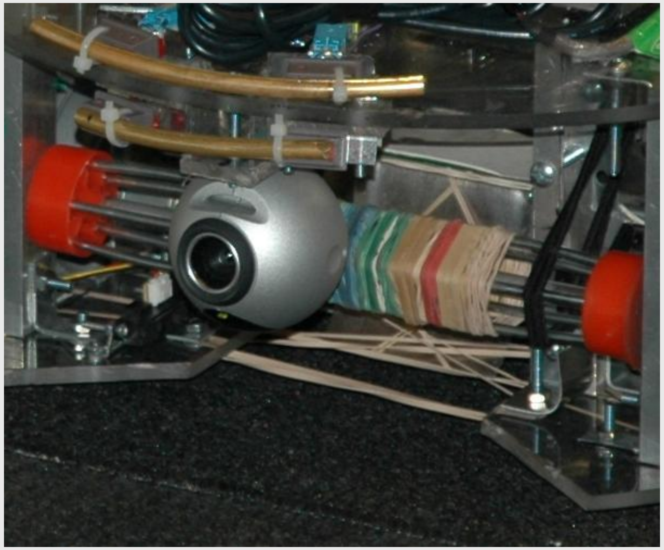
- Strategy
  - Schedule
  - Systematic Strategy Selection
  - Case Studies
- Mechanical Design
  - Design Process
  - Mechanisms
  - Resources and Tools

Mechanisms

- Most Common Mechanisms
  - Roller
  - Mounting
    - Servos
    - IRs
    - Cameras
    - Electronics
    - Battery

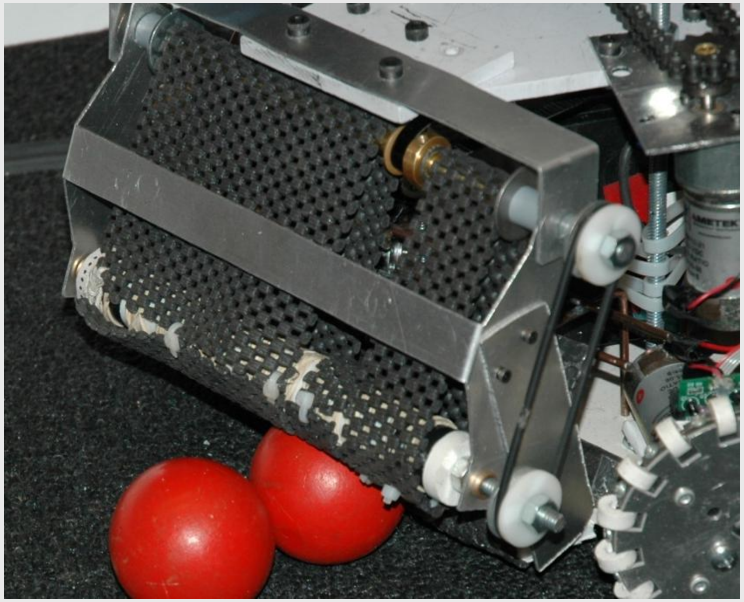
How to build a Roller

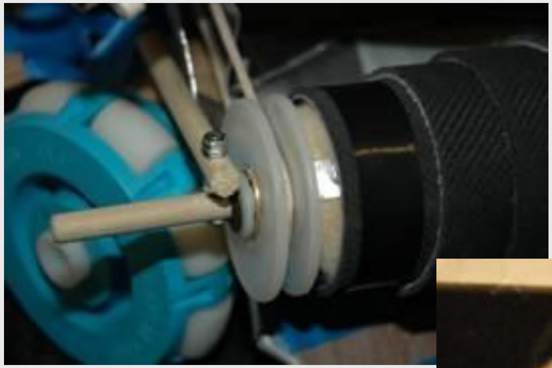




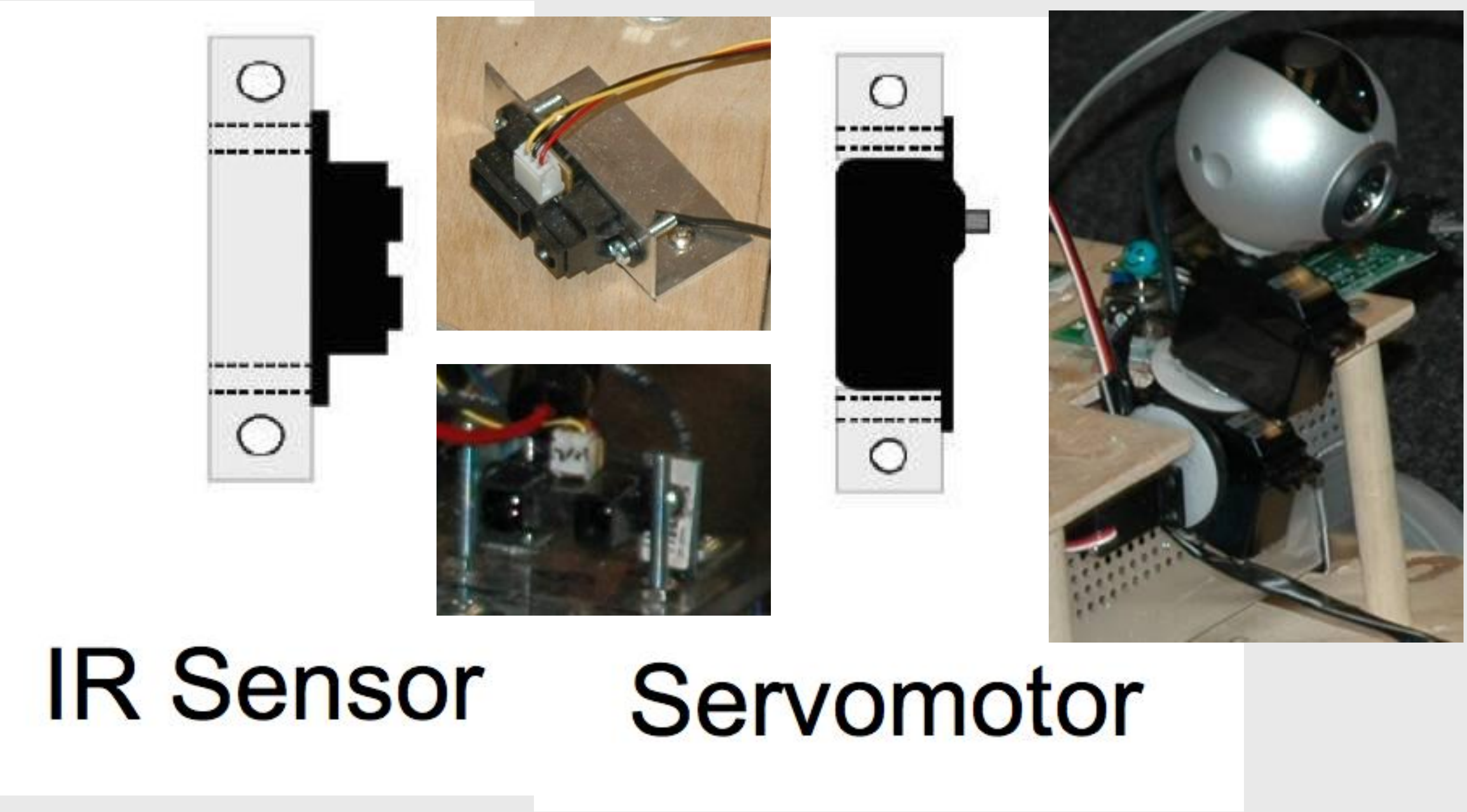
<u>Roller Material:</u>
Radially Compliant
Tangentially high grip

<u>Roller Drive:</u>
Belt Drive
Gear Drive (Use a gear box!)

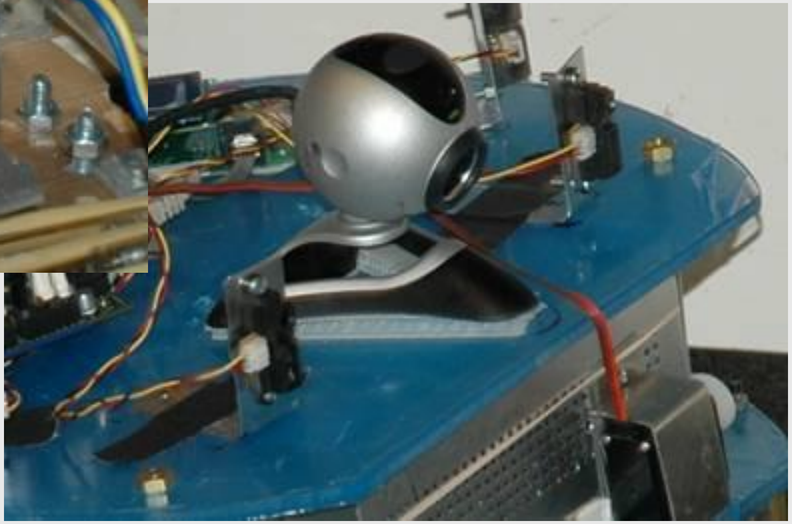


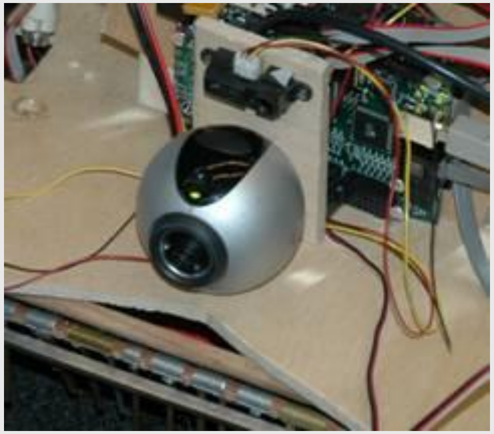


Mounting: Sensors & Servos



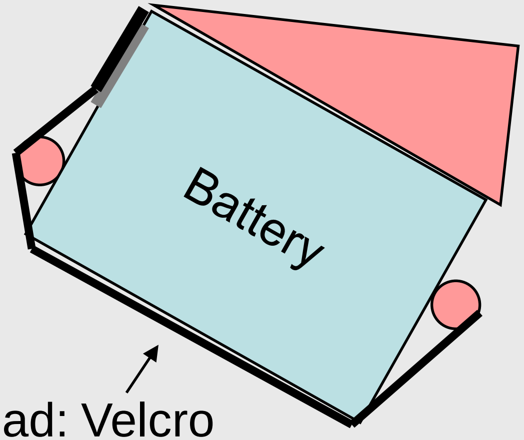
Mounting: Camera





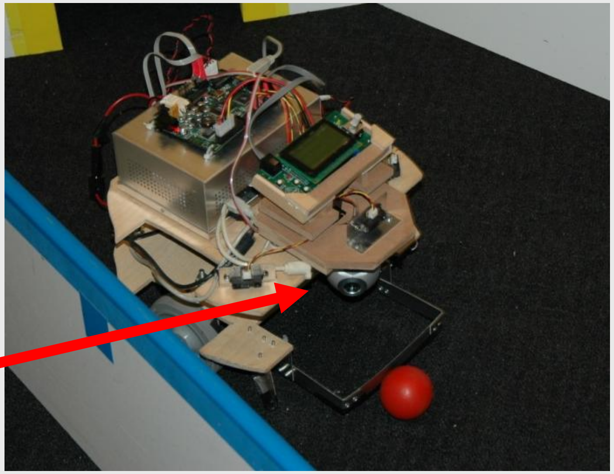
Mounting: Computer, Orcboard, and Battery

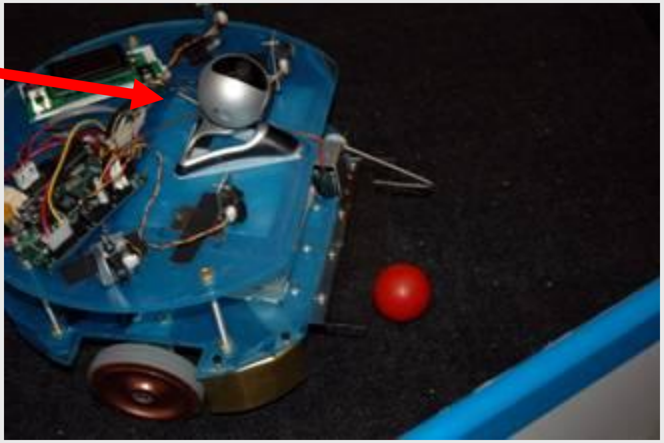
- Velcro
- Zip ties
- Kinematic Constraint
  + Preload



Good Mechanical Design Simplifies Coding

- **Wide collection bin** requires less driving precision
- **Large bin capacity** eliminates need for keeping track of number of balls collected
- **Rotating camera** for vision without turning
- **Camera position** so that it does not see the blue line on tall walls
- **Bump sensors** give reliable navigation inputs
- **Modularity** allows for programmers and builders to work in tandem





Agenda

- Strategy
  - Schedule
  - Systematic Strategy Selection
  - Case Studies
- Mechanical Design
  - Design Process
  - Mechanisms
  - Resources and Tools

Resources and Tools

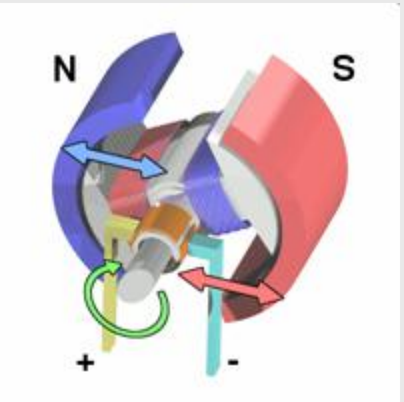
- Maslab 6.002 Lab
  - Available tools:
    - Drill press, scroll saw, arbor press, thin sheet metal bender and shear, hand tools…
    - Hand tools, hand drills
  - Maslab saw special notes:
    - Use on wood and thin plastics. Do not cut acrylic (will crack), thick aluminum, or other metals
- Edgerton Shop
  - Band saw
    - Cutting wood, metal, acrylic
  - Drill press
    - Great for drilling holes
  - Milling machine
    - Accurate machining
  - Lathe
    - Building motor couplings, axels, pulley wheels (anything round)

Never cut metals that are thinner than the distance between saw teeth on any type of saw – Use shears or sheet metal cutter

Where to find materials?

- On campus
  - Central Machine Shop (basement building 36)
  - Pappalardo/Edgerton/other various shops
  - Reuse/Basements
- Online
  - McMaster Carr: any mechanical hardware you could need with overnight delivery http://www.mcmaster.com
  - Grainger (similar to McMaster) http://www.grainger.com
  - http://www.hobbyengineering.com/
  - http://www.onlinemetals.com/
  - http://sabest.org/links/partslinks.html

DC Motors



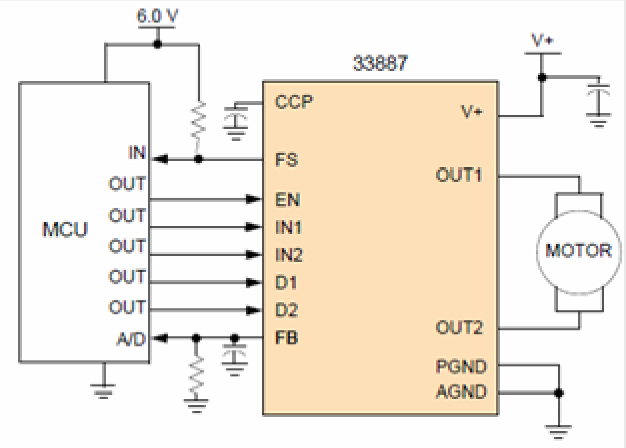
- $P_{in}=P_{out}$
- $P_{in}=IV_{bemf}$
- $I = (V_{applied}-V_{bemf}) / R_{armature}$
- $P_{out}=T\omega$
- How do I….?
  - Increase motor power
    - Increase applied voltage
    - Be careful of thermal limit, voltage limit
  - I can't increase voltage. Need more torque!
    - Increase gear ratio, sacrifice speed
  - My motor is too slow
    - Decrease gear ratio, sacrifice torque
  - I'm still not getting enough torque/speed
    - Change motor armature -> get a new motor!
- Read the Datasheet
- Check that your motor has the power/torque/speed you need

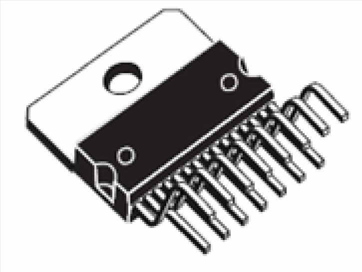
Characteristic	Value	Unit
M12V200		
Operating voltage	4.5-12	V
No load current	111	mA
No load speed	200	RPM
Startup torque	3.6	KG•CM
Gear ratio	30:1	

Special Consideration: Battery Power, Omniwheels

Motor Controller

- uOrc provides 3 motor ports, you may need more
- Additional Motor Controller Options
  - External Package
  - Design your own board
    - uORc uses MC3887 ($5.60)
- Consider
  - Does your motor need current sense?
  - Does your motor need speed control or is it on/off?
    - Yes: Use PWM (Digital IO or I2C, or SPI control chip)
    - No: Use an H-bridge (Digital IO)
  - Does your motor need to be bidirectional?
    - Yes: Use Full bridge driver (L298 $6.05)
    - No: Use Half or full bridge driver
- When adding motors, do a power budget





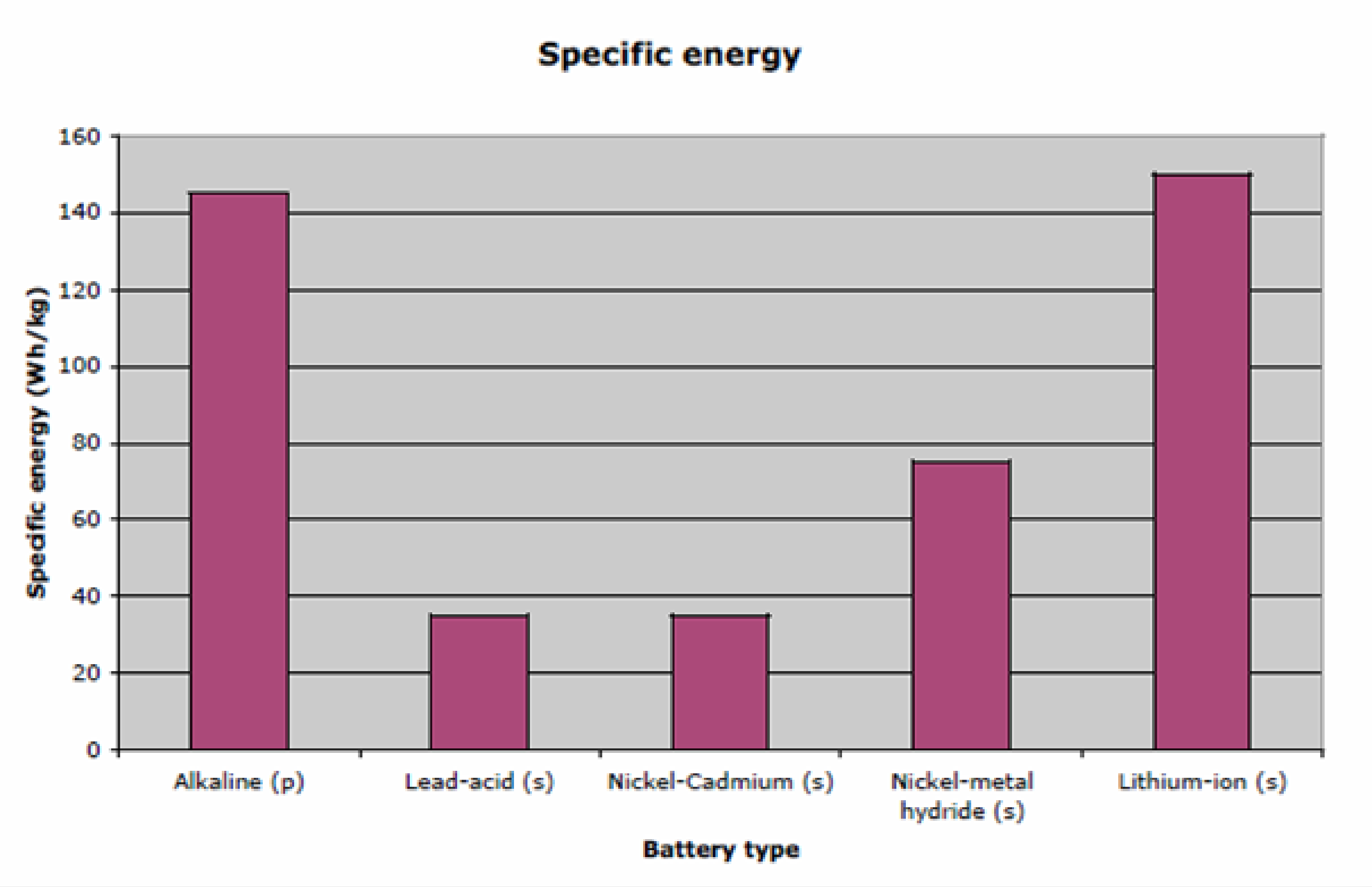
Batteries

- Sealed Lead Acid Battery
- Battery
  - 1.84 kg (4.06 lb)
  - 12V 5AH
  - 67.5 A max
- Charger
  - Smart Charger 1.0A at 12V
- AmpHours * V = P*t
  - You determine time (t)
  - Minimum time = 4.4 min
- Alternate Batteries
  - Check uOrc and component compatibility
  - More motor torque → Higher Voltage
  - Longer Runtime → Higher AH
  - Continuous Run (2 batteries) → Charger
  - Weight/Space

2008 Team 9

Batteries

Power Density



Take Aways

- Do a quick BLE (Bench Level Experiment)
  - Things don’t always work like you think
  - Will save you hours of debugging later!
- Modular design
  - Have a base robot built ASAP for the programmers to use.
  - Make it easy to take apart to fix or replace pieces
- Design for assembly (and <u>disassembly</u>)
  - Avoid glue when possible
  - Avoid screws when velcro will do
- Design pieces to be adjustable
- Building takes 3 times longer then you think it will.
- Design by night, build by day (when shops are open).
- No serious, building takes *forever*.

References

- Chris Celio, “Mechanical Engineering: Design, Strategy, and Building,” 2008
- 2006 Maslab Wiki, http://maslab.mit.edu/2006/Wiki
- 2007 Maslab Wiki, http://maslab.mit.edu/2007/Wiki
- 2008 Maslab Wiki, http://maslab.mit.edu/2008/Wiki
- Batteries: http://www.batteryspace.com/index.asp?PageAction=VIEWPROD&ProdID=2145
- Battery Charger: http://www.batteryspace.com/index.asp?PageAction=VIEWPROD&ProdID=2518
- 2.009 Battery Primer: http://web.mit.edu/2.009/www/resources/mediaAndArticles/batteriesPrimer.pdf
- Motors: http://www.solutions-cubed.com/solutions%20cubed/Products%20Page/Downloads/ER_DS_8.pdf
- 2.009 Design Resources: http://web.mit.edu/2.009/www/resources/resourceIndex.html