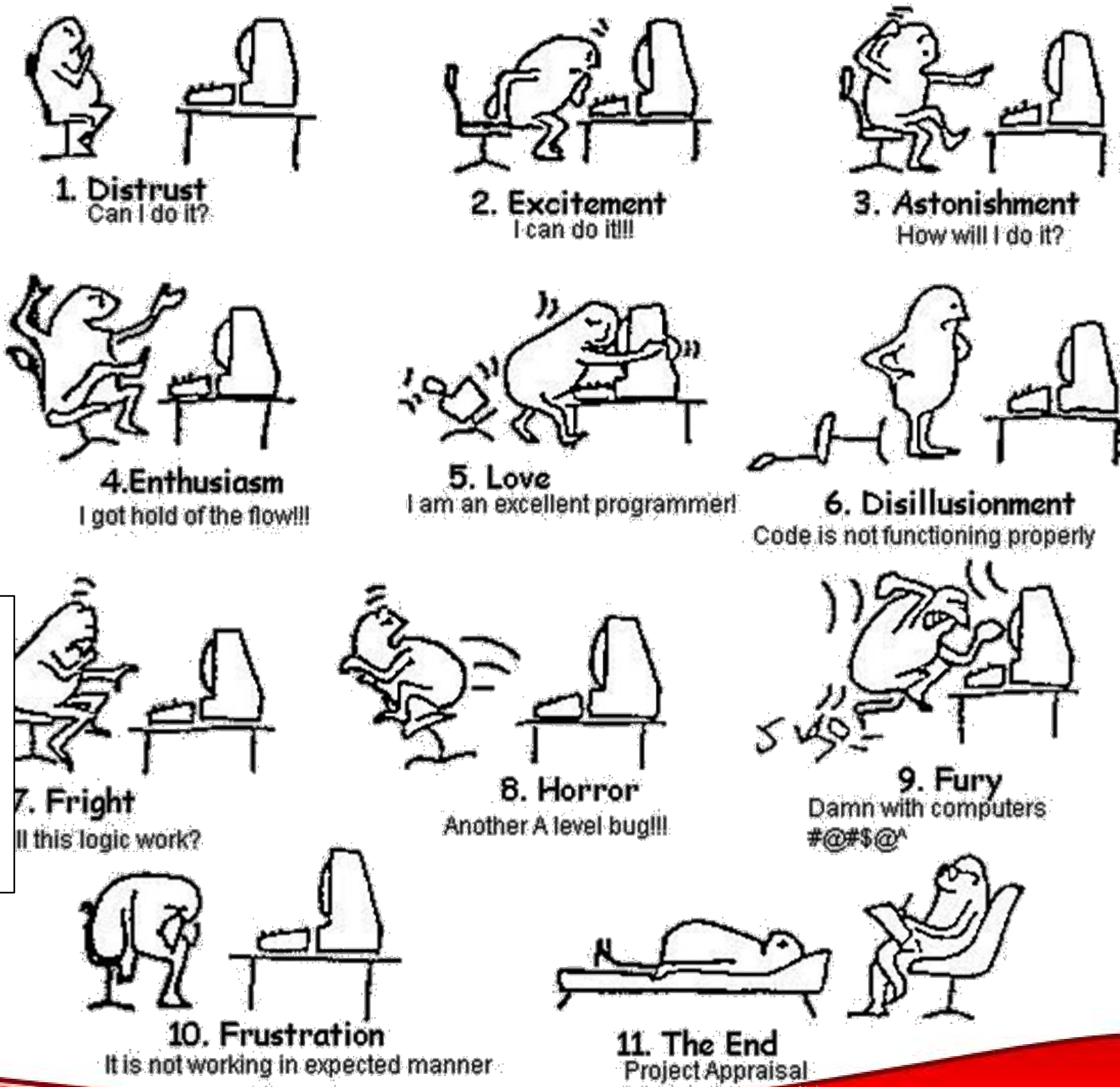
https://ajaywhiz.wordpress.com/2010/02/23/software-development-life-cycle-emotions/

ECE3400: Intelligent Physical Systems

Debugging and Evaluation

Classes of Interest:
ILRST 2100: Introductory Statistics
ENGRD 2700: Basic Engineering Probability & Statistics
ECE 3100: Probability and Statistics
MATH 4720: Statistics



Debugging Intelligent Physical Systems

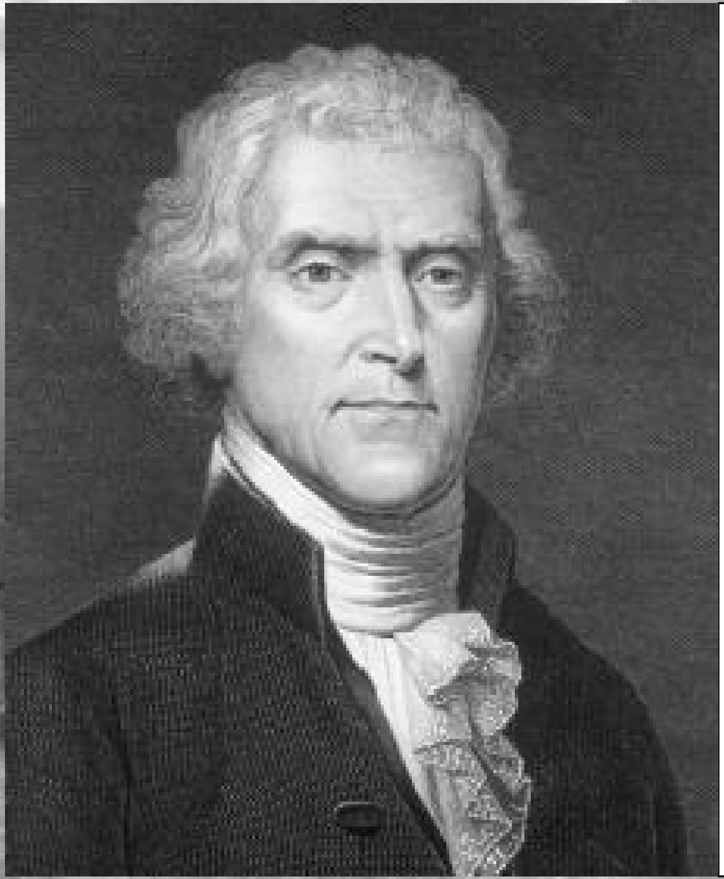
MARK I Computer, Harvard, by Howard Aiken in 1944

Admiral Grace Hopper, 1906-1992

Debugging Intelligent Physical Systems

MARK I Computer, Harvard, by Howard Aiken in 1944

Admiral Grace Hopper, 1906-1992



Thomas Jefferson 1878:

“'Bugs' -- as such little faults and difficulties are called -- show themselves after months of intense watching, study and labor are requisite before commercial success or failure is certainly reached.”

Debugging Intelligent Physical Systems

Debugging is more complex than ever!

- Electronics
- Software
- Mechanics
- Multiple connected devices
- Simulation

Worst bugs are intermittent

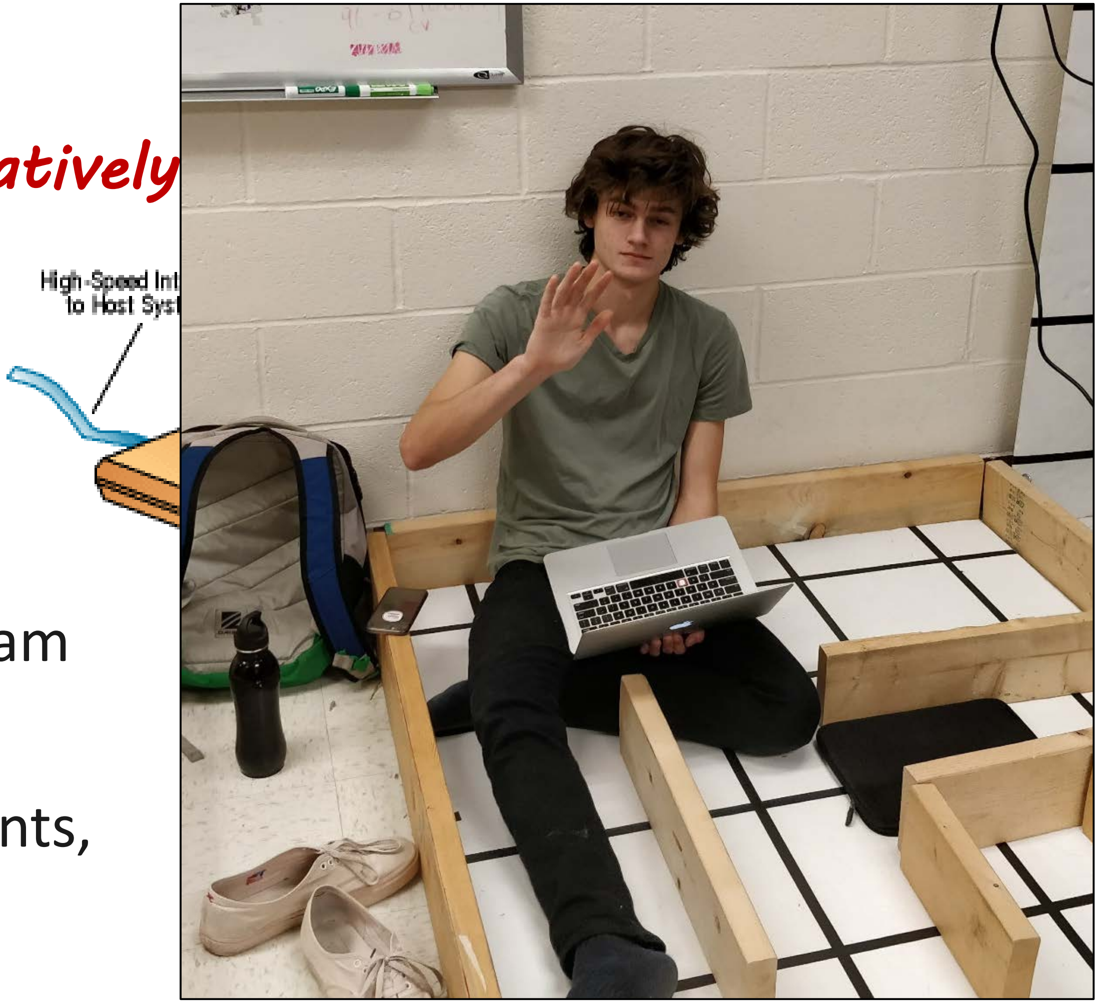
→ Apply a methodical and documented search

Software Debugging

- *How do you debug software?*
- Compilers
  - Syntax or typo errors
  - Exception handling
- Simulation environments
  - Allows you to monitor the execution of a program
  - Stop, restart, break points, etc.
    - Standard breakpoints, conditional breakpoints, breakpoints with counters
  - Change values in memory
  - AVR Studio / Visual Micro debugger for the Atmel processers

Cheap, but slow…

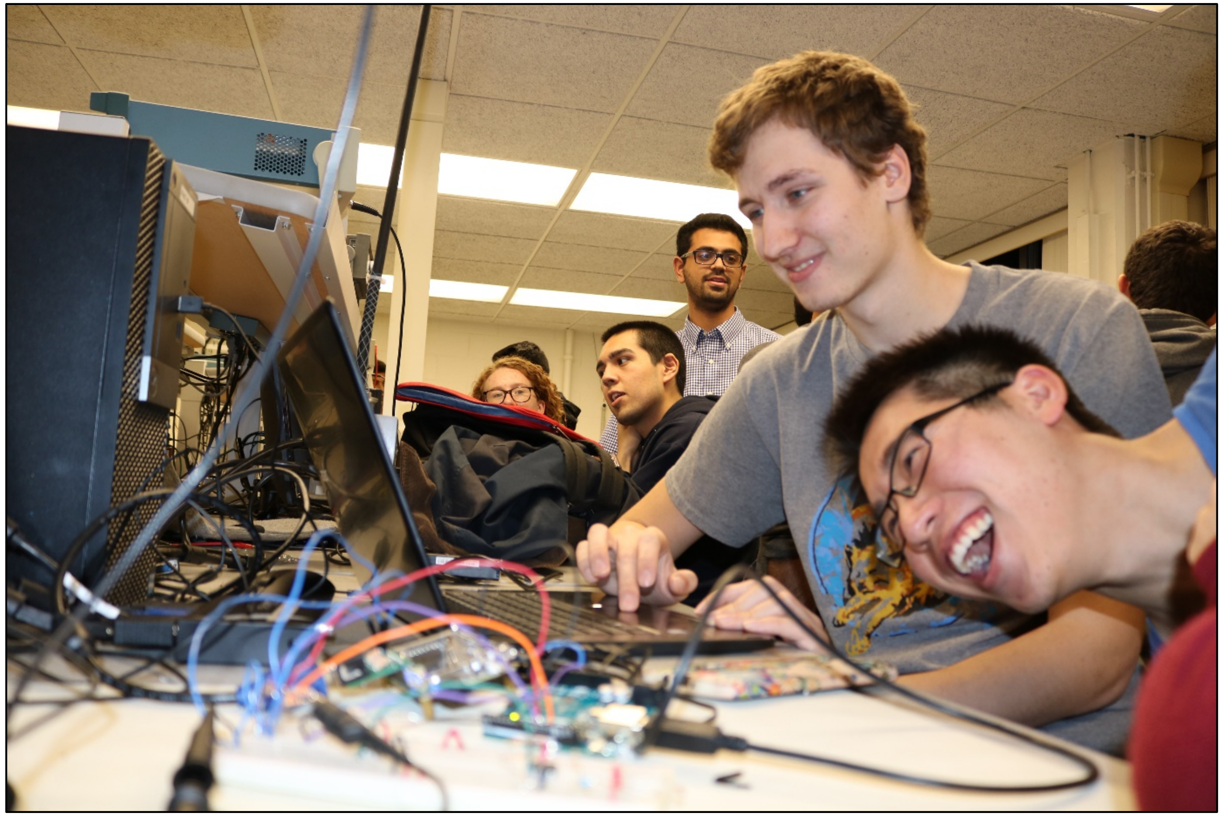
Alternatively



(AVR Dragon)

Circuit Debugging

- Know your components
- Unit tests
- Check wiring
- Ensure common ground

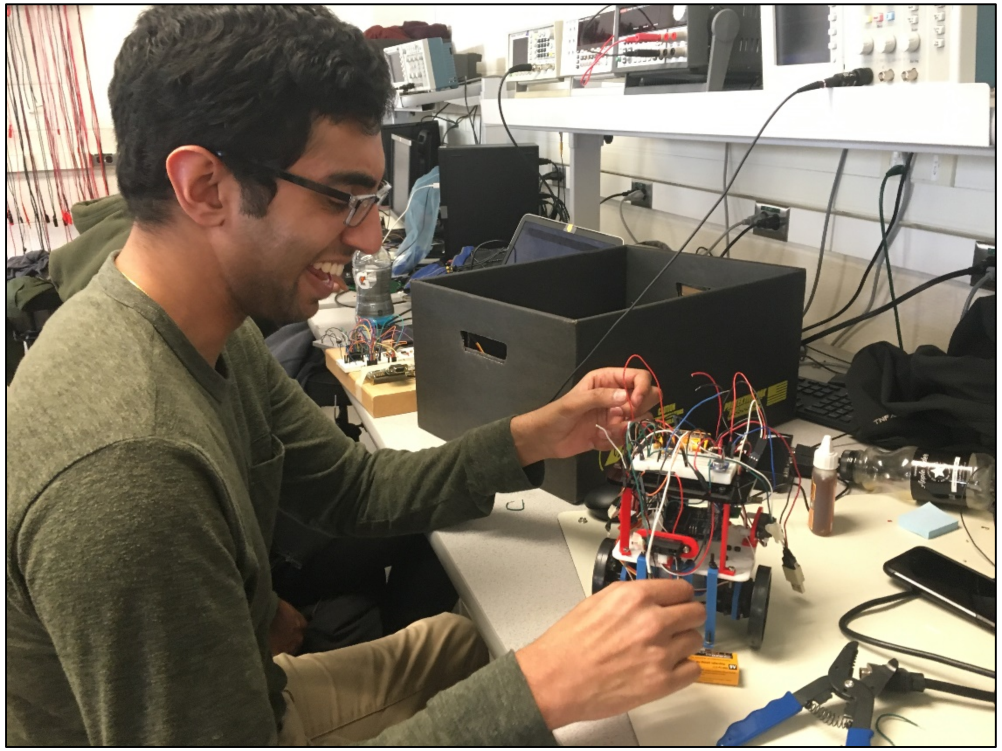


PCB Debugging

- Always test circuit beforehand!
- Add test points
- Make circuit dividers
- Check out class burn list:
  - https://cei-lab.github.io/ece3400-2017/tutorials/PCB/burnlist.html
- Visual inspection
- Unit tests

Mechanical Bugs?

- Typically related to friction or jamming
- Broken teeth/dirt in gears
- Broken axels
- Fallen/obscured sensors
- Broken wires

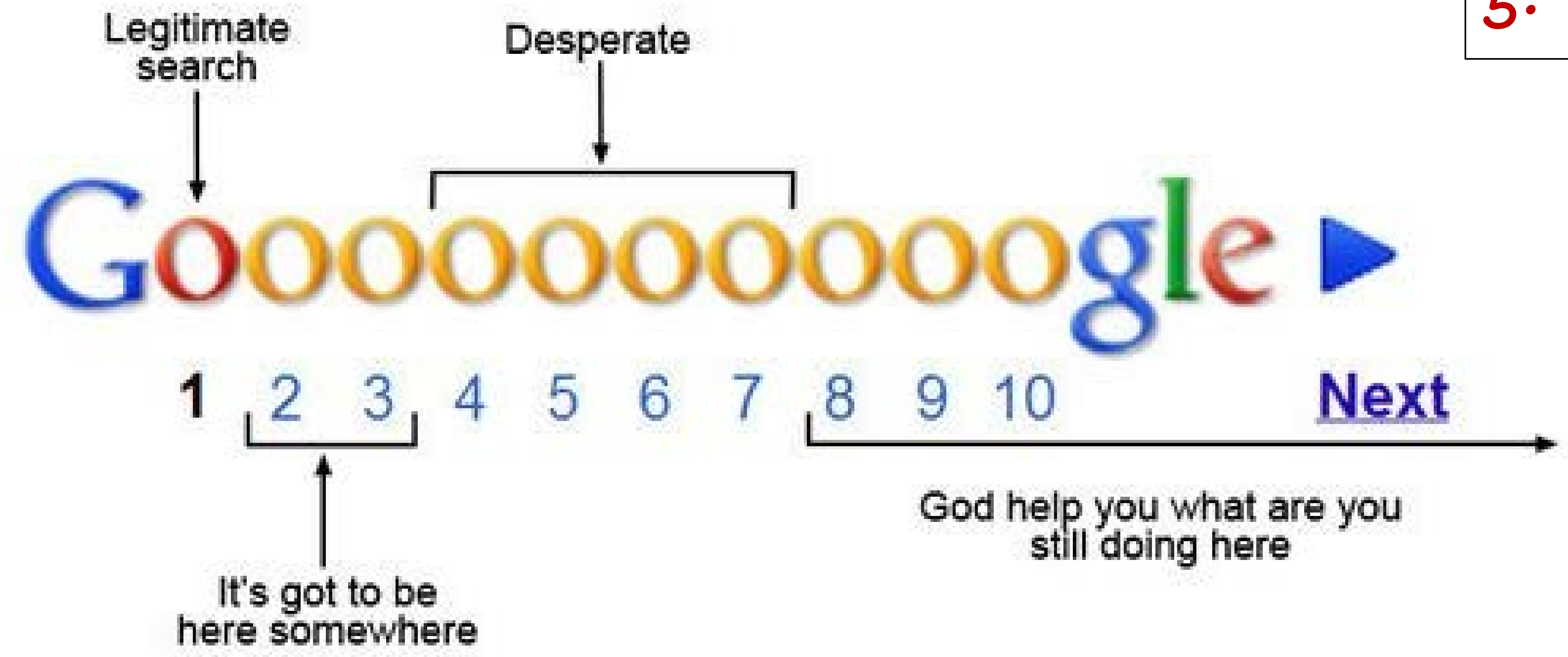


Errors leads to symptoms:

- Bad sensor values
- Slow/biased movement
- Jamming may cause power surges and reset conditions

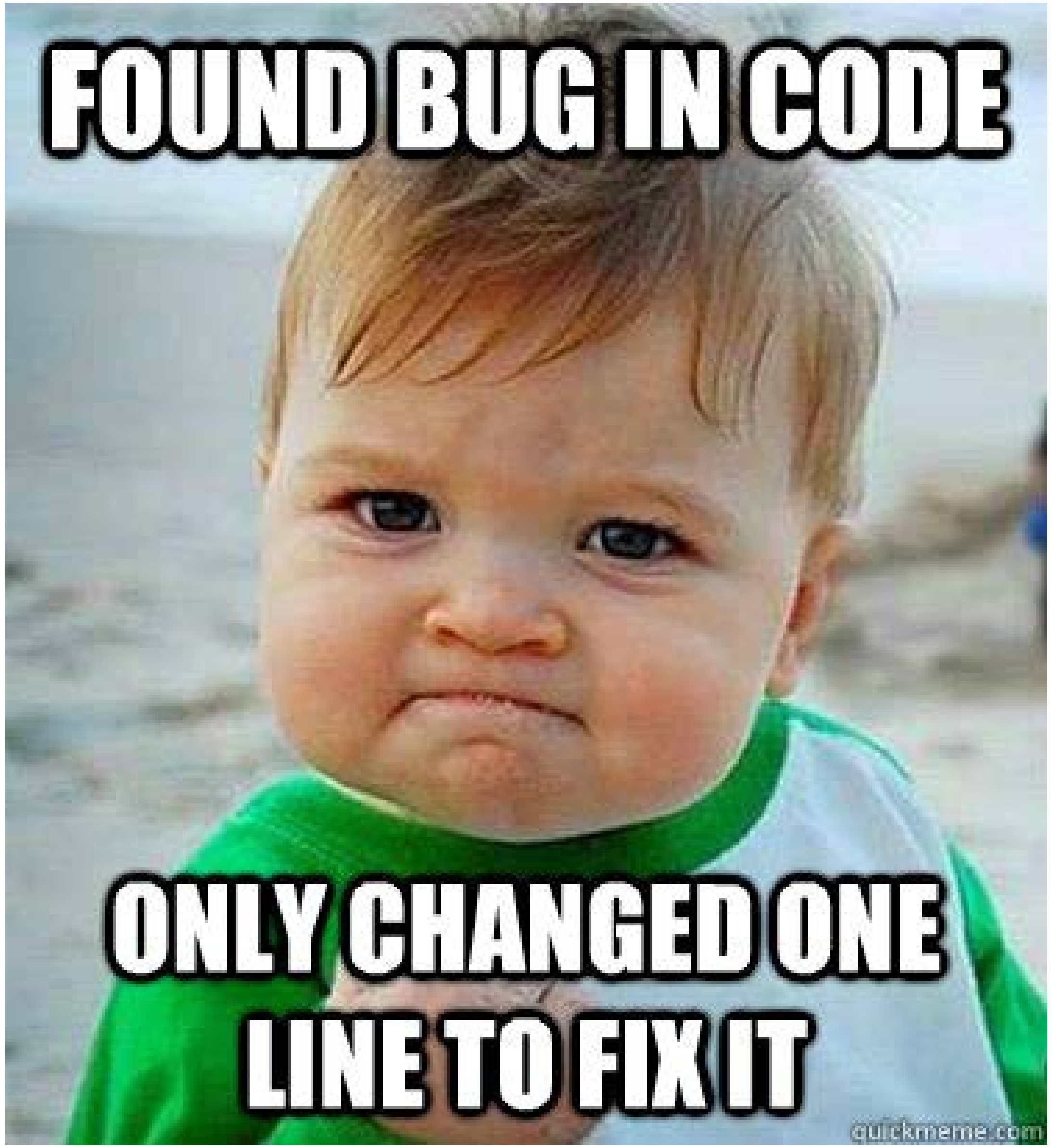
Browsing for help online

1. Try on your own
2. Try with a friend
3. Ask a TA
4. Ask Kirstin
5. Then what?


Legitimate search
Desperate
Goooooooooogle
1 2 3 4 5 6 7 8 9 10
Next
It's got to be here somewhere
God help you what are you still doing here

Debugging IPS

- STEP 1: Reproduce the symptom!
- STEP 2: Hunt down the bug *systematically*
  - Brute force debugging
  - Problem simplification
  - Backtracking (start from problem)
  - Tracing or print debugging
  - Binary Search
  - Bug clustering
  - *Scientific Method*: Form hypothesis and test it
- STEP 3: Solve the Problem
  - Assume simple error first
  - (Don’t look for complex solutions)



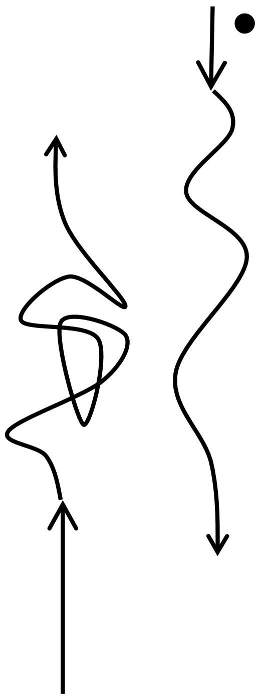
Debugging IPS

…Or try to prevent the bug in the first place

- Clean code, electronics, wiring, mechanics
- Incremental development: Compile/test often!
- Instrument program to log information
- Instrument program with assertions
  - Always add else-statements
  - Always add default to switch case statements
  - Add value checkers
  - Add visible feedback (LEDs?)

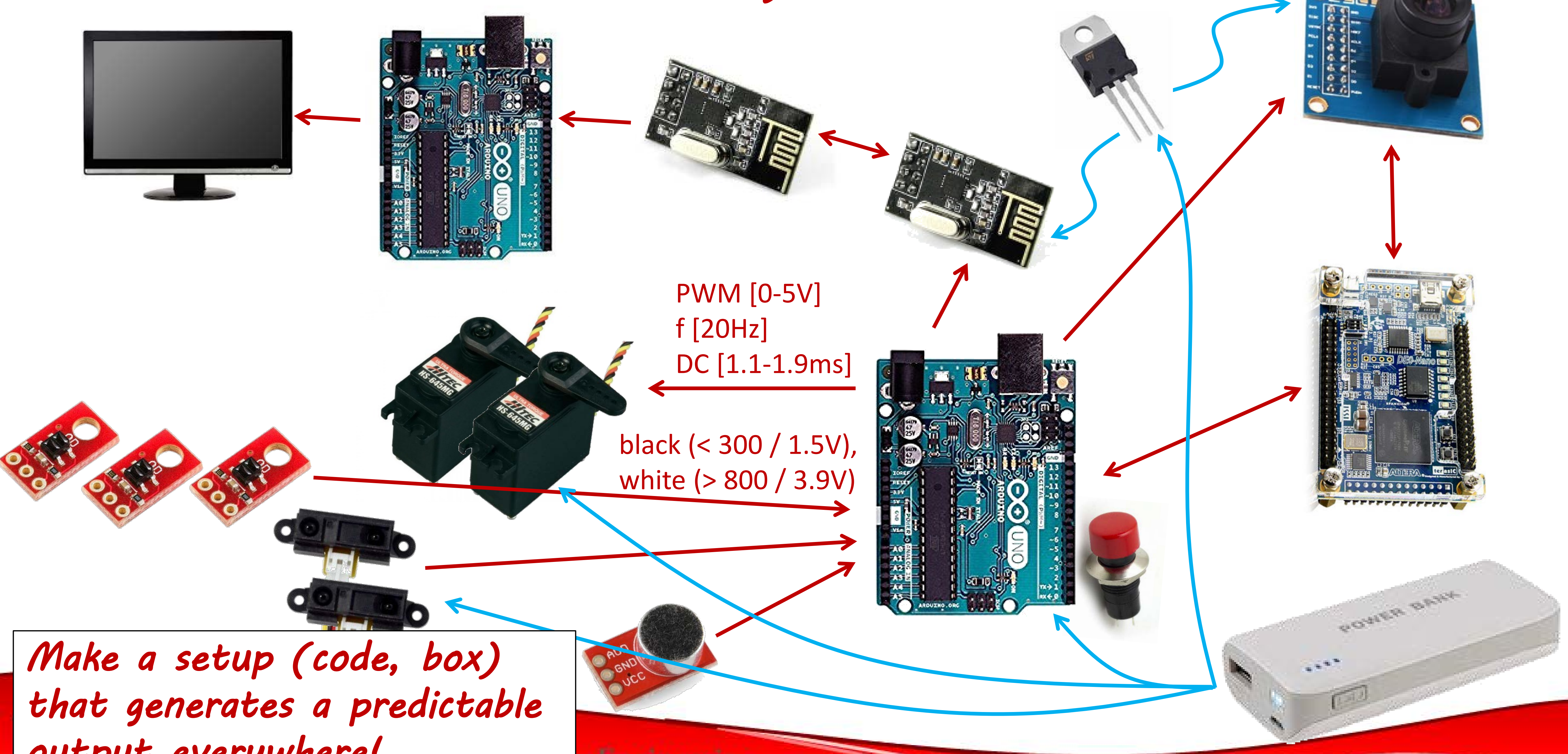
Developing Your System

- **Bottom-up development**
  - Unit testing
  - Faster initial progress



- **Top-down development**
  - Implement every thing to begin with
  - Add dummy functions as placeholders
  - Leads to more modular products
  - (Requires some familiarity with IPS)

Map Your System , *then describe your interfaces!*



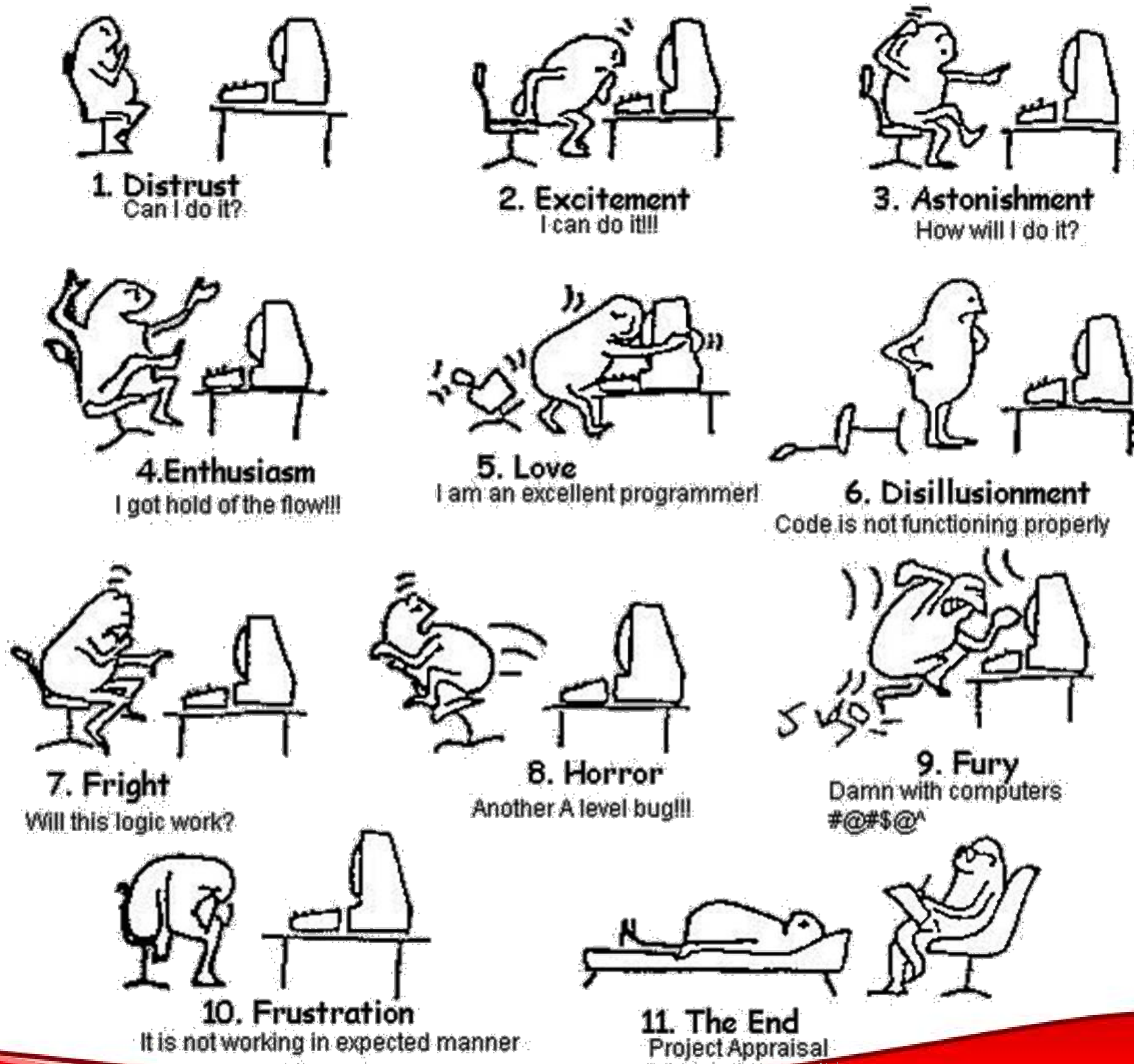
PWM [0-5V]
f [20Hz]
DC [1.1-1.9ms]

black (< 300 / 1.5V),
white (> 800 / 3.9V)

Make a setup (code, box) that generates a predictable output everywhere!

ECE3400: Intelligent Physical Systems

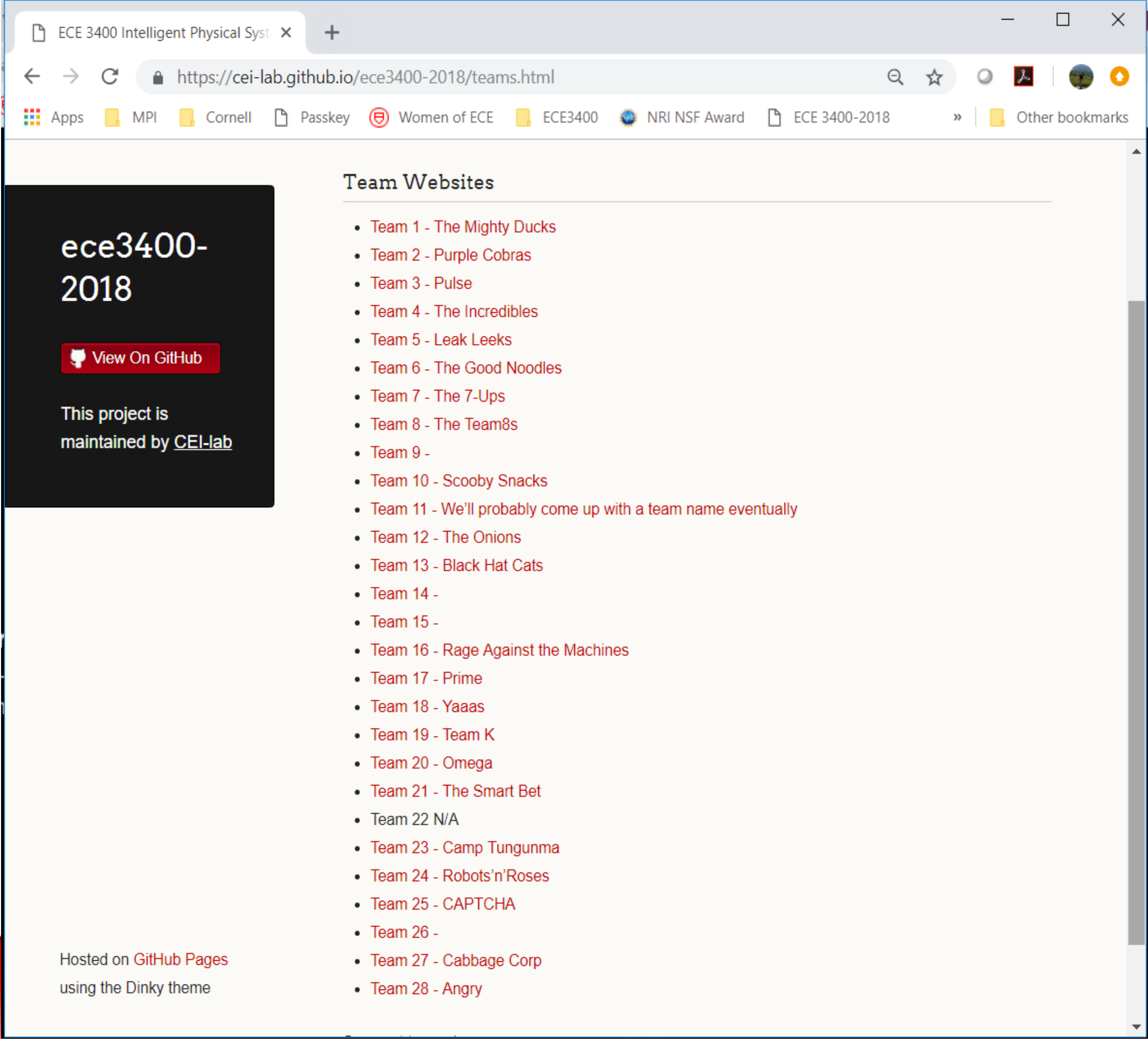
Debugging and Evaluation



https://ajaywhiz.wordpress.com/2010/02/23/software-development-life-cycle-emotions/

Pitch Your Product!

- There are 43 teams linked from the class webpage.
- **Users**
  - Assume no prior knowledge
  - And minimal attention span
  - First impression



ECE 3400 Intelligent Physical Syst
https://cei-lab.github.io/ece3400-2018/teams.html
Apps
MPI
Cornell
Passkey
Women of ECE
ECE3400
NRI NSF Award
ECE 3400-2018
Other bookmarks

ece3400-2018

View On GitHub

This project is maintained by CEI-lab

Hosted on GitHub Pages using the Dinky theme

Team Websites

- Team 1 - The Mighty Ducks
- Team 2 - Purple Cobras
- Team 3 - Pulse
- Team 4 - The Incredibles
- Team 5 - Leak Leeks
- Team 6 - The Good Noodles
- Team 7 - The 7-Ups
- Team 8 - The Team8s
- Team 9 -
- Team 10 - Scooby Snacks
- Team 11 - We'll probably come up with a team name eventually
- Team 12 - The Onions
- Team 13 - Black Hat Cats
- Team 14 -
- Team 15 -
- Team 16 - Rage Against the Machines
- Team 17 - Prime
- Team 18 - Yaaas
- Team 19 - Team K
- Team 20 - Omega
- Team 21 - The Smart Bet
- Team 22 N/A
- Team 23 - Camp Tungunma
- Team 24 - Robots'n'Roses
- Team 25 - CAPTCHA
- Team 26 -
- Team 27 - Cabbage Corp
- Team 28 - Angry

Pitch Your Product!

- *Pick a website from someone at your table (not your own), and find 1-2 positive first impressions*
  - **Mighty Ducks:** https://mb2372.github.io/ece3400-team1/
    - First thing you see is a video of their robot!
  - **Prime:** https://3400-17.github.io/Prime/#
    - Simple, positive message
  - **Camp Tugunma:** https://ece-3400-group.github.io/
    - Informative subheadings for labs and milestones
  - **YAAAS:** https://am2384.github.io/
    - Calls for action
  - **The Smart Bet:** https://ece3400-fa18-group21.github.io/
    - ‘Meet our Team of Roboticists’
  - **Captcha:** https://eldorbekpulatov.github.io/ece3400/index.html
    - Menu is easily accessible (about ‘motivated students’, #todo)
  - **Team 14:** https://ece3400-team14.github.io/Team-14-Website/
    - Updates

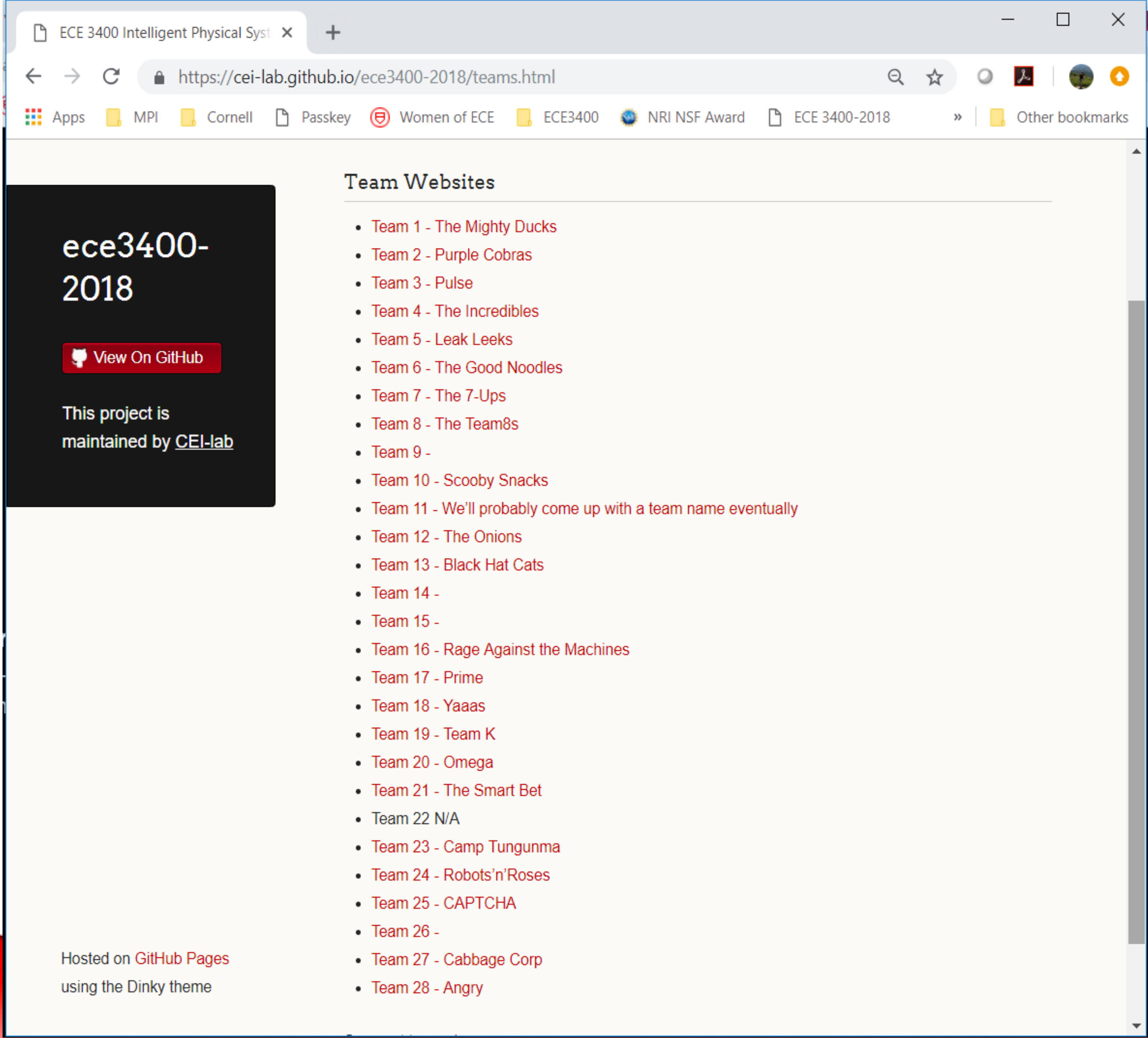
Pitch Your Product!

- *What issues did you notice?*
  - No team name
  - No introduction
  - No real photos
  - High quality photos that took a long time to load
  - Long code snippets without detailed explanations
  - etc.

Pitch Your Product!

- There are 43 teams linked from the class webpage.
- **Users**
  - Assume no prior knowledge
  - And minimal attention span
  - First impression

- ***Why should they read yours?***
  - *Brief* description
  - (LQ) Photos
  - Engaging



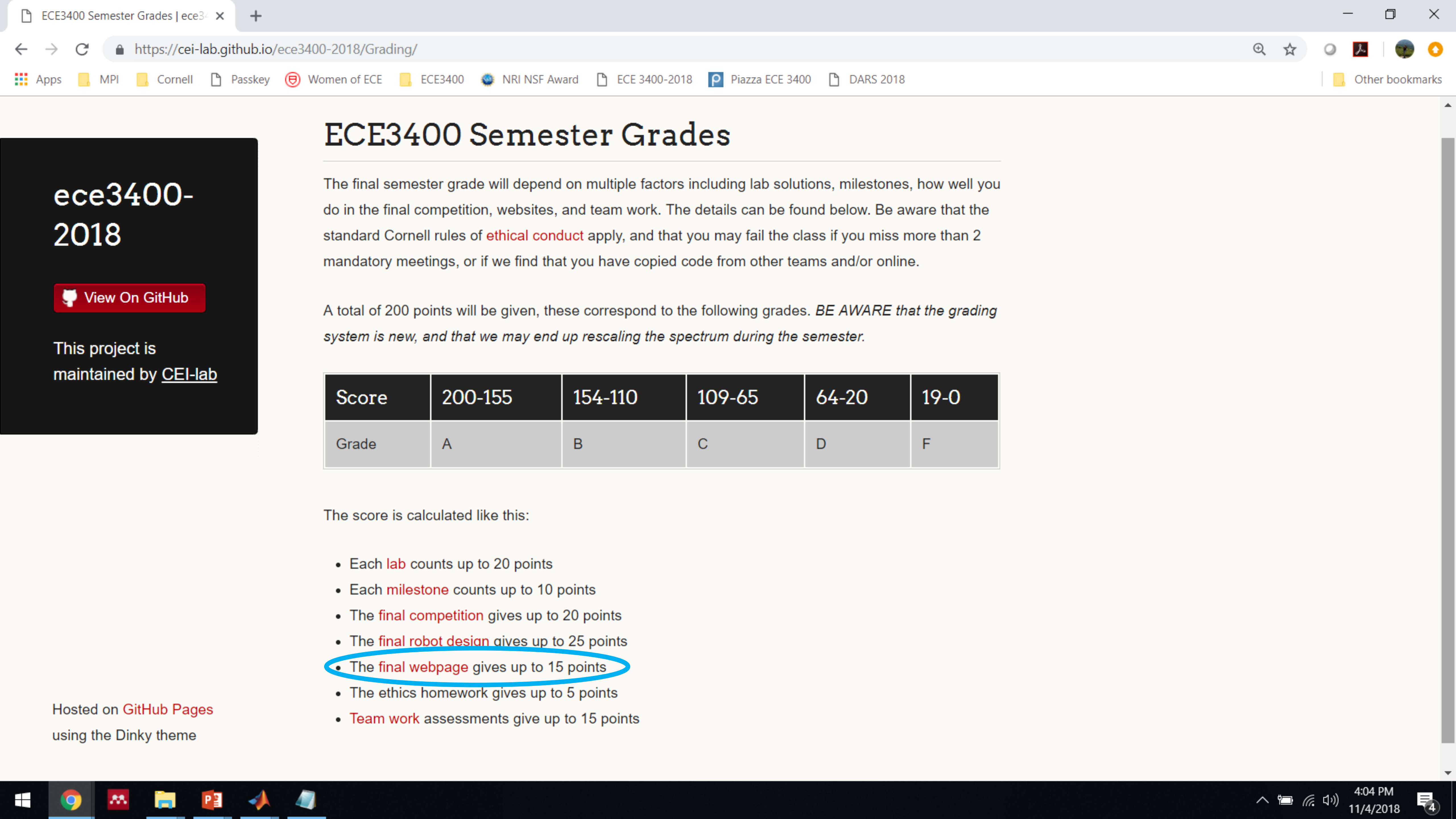
ece3400-2018

This project is maintained by CEI-lab

Hosted on GitHub Pages using the Dinky theme

Team Websites

- Team 1 - The Mighty Ducks
- Team 2 - Purple Cobras
- Team 3 - Pulse
- Team 4 - The Incredibles
- Team 5 - Leak Leeks
- Team 6 - The Good Noodles
- Team 7 - The 7-Ups
- Team 8 - The Team8s
- Team 9 -
- Team 10 - Scooby Snacks
- Team 11 - We'll probably come up with a team name eventually
- Team 12 - The Onions
- Team 13 - Black Hat Cats
- Team 14 -
- Team 15 -
- Team 16 - Rage Against the Machines
- Team 17 - Prime
- Team 18 - Yaaas
- Team 19 - Team K
- Team 20 - Omega
- Team 21 - The Smart Bet
- Team 22 N/A
- Team 23 - Camp Tungunma
- Team 24 - Robots'n'Roses
- Team 25 - CAPTCHA
- Team 26 -
- Team 27 - Cabbage Corp
- Team 28 - Angry



ece3400-2018

View On GitHub

This project is maintained by CEI-lab

ECE3400 Semester Grades

The final semester grade will depend on multiple factors including lab solutions, milestones, how well you do in the final competition, websites, and team work. The details can be found below. Be aware that the standard Cornell rules of ethical conduct apply, and that you may fail the class if you miss more than 2 mandatory meetings, or if we find that you have copied code from other teams and/or online.

A total of 200 points will be given, these correspond to the following grades. *BE AWARE that the grading system is new, and that we may end up rescaling the spectrum during the semester.*

Score	200-155	154-110	109-65	64-20	19-0
Grade	A	B	C	D	F

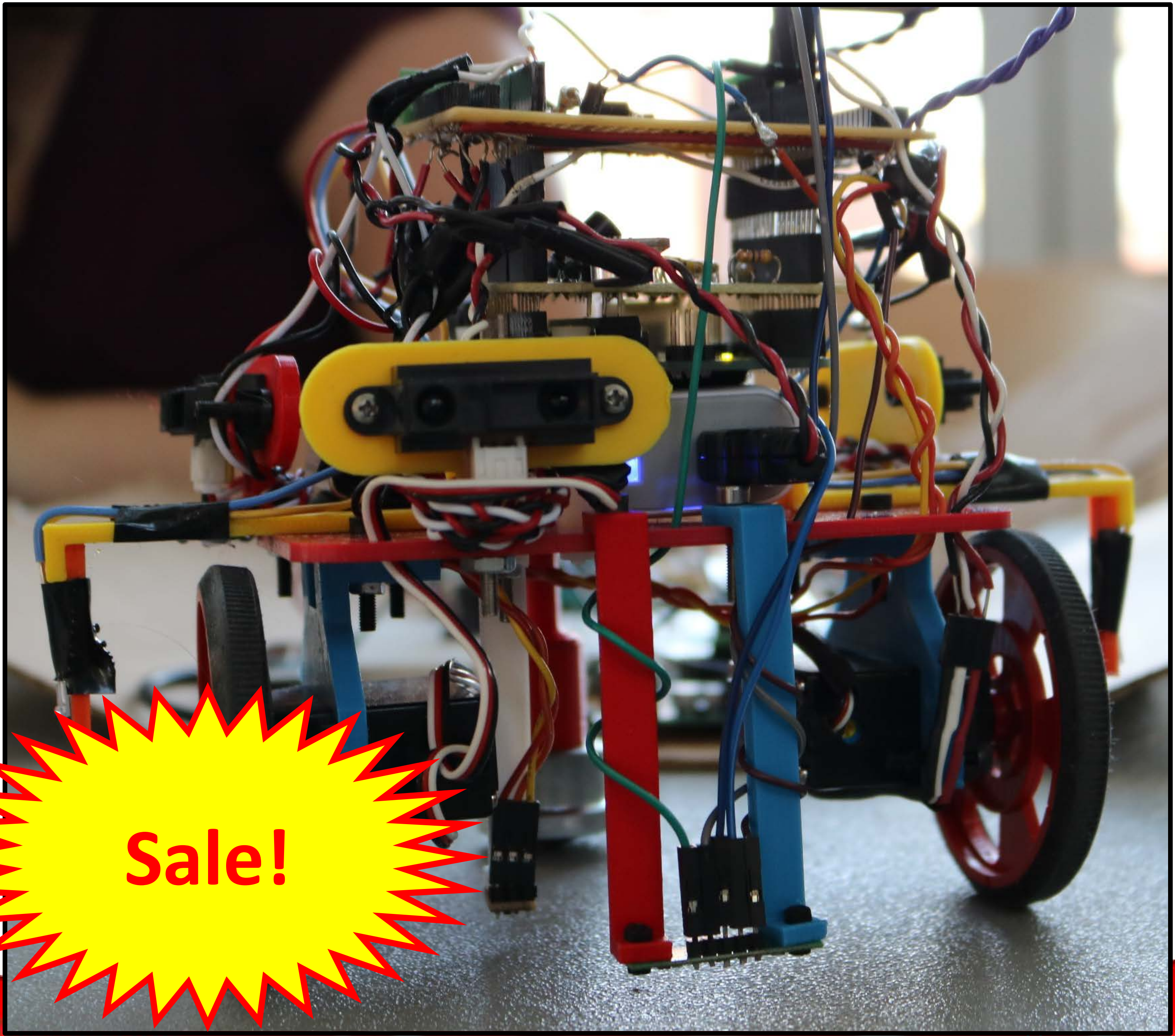
The score is calculated like this:

- Each lab counts up to 20 points
- Each milestone counts up to 10 points
- The final competition gives up to 20 points
- The final robot design gives up to 25 points
- The final webpage gives up to 15 points
- The ethics homework gives up to 5 points
- Team work assessments give up to 15 points

Hosted on GitHub Pages using the Dinky theme

Product Appraisal

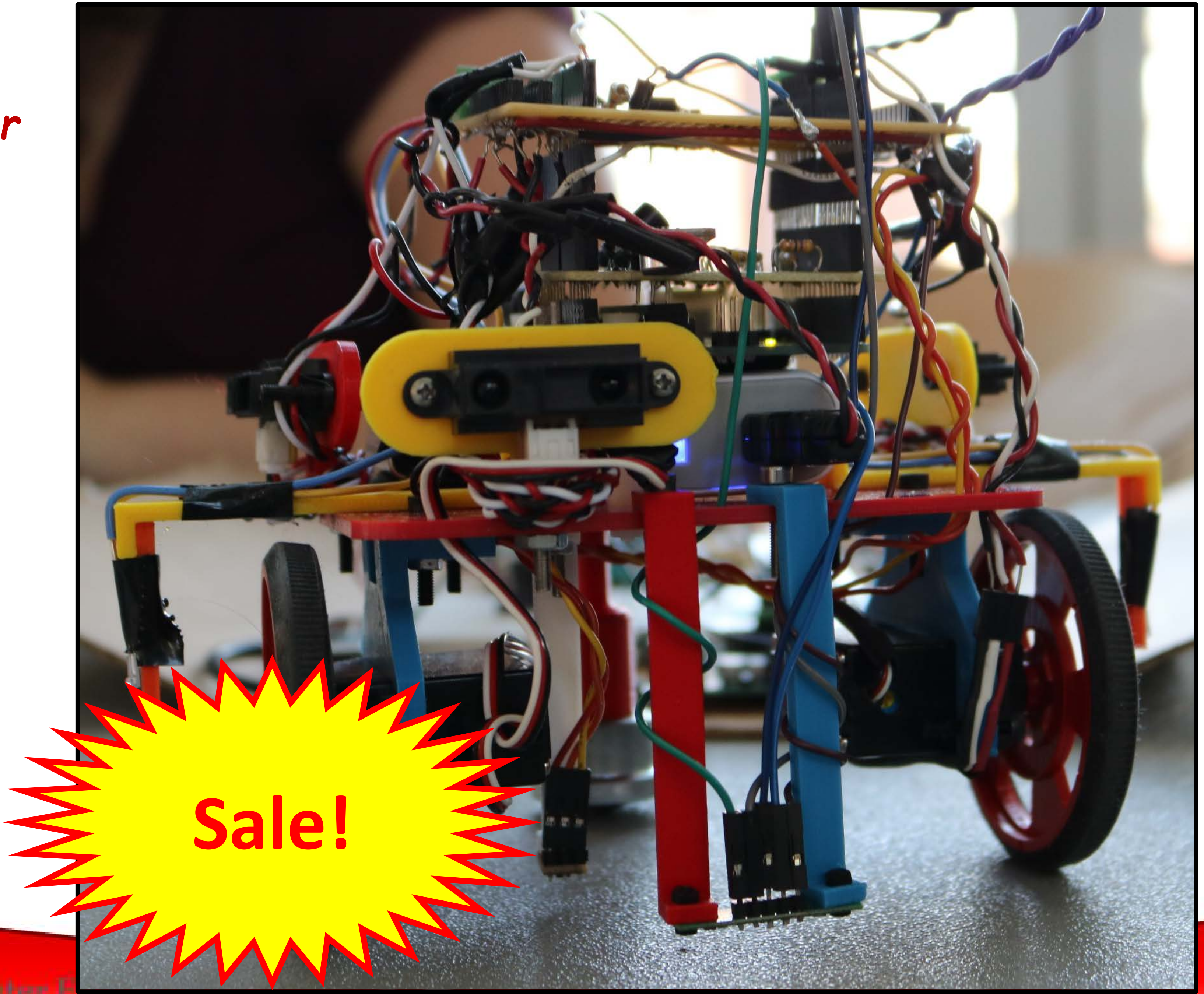
- *How do you spec out your robot?*
- **Capabilities:**
  - Mobile
  - Autonomous
  - Line following
  - Wall detection
  - Tone detection
  - Visual treasure detection
  - Robot detection
  - Maze mapping
  - Report to external screen



Sale!

Product Appraisal

- *How do you spec out your robot?*
- **Operating Conditions:**
  - Not impact resistant
  - Not water resistant
  - Minimum line size
  - Minimum grid size
  - Wall/line color
  - Treasure types
  - Light intensity
  - Audible noise level



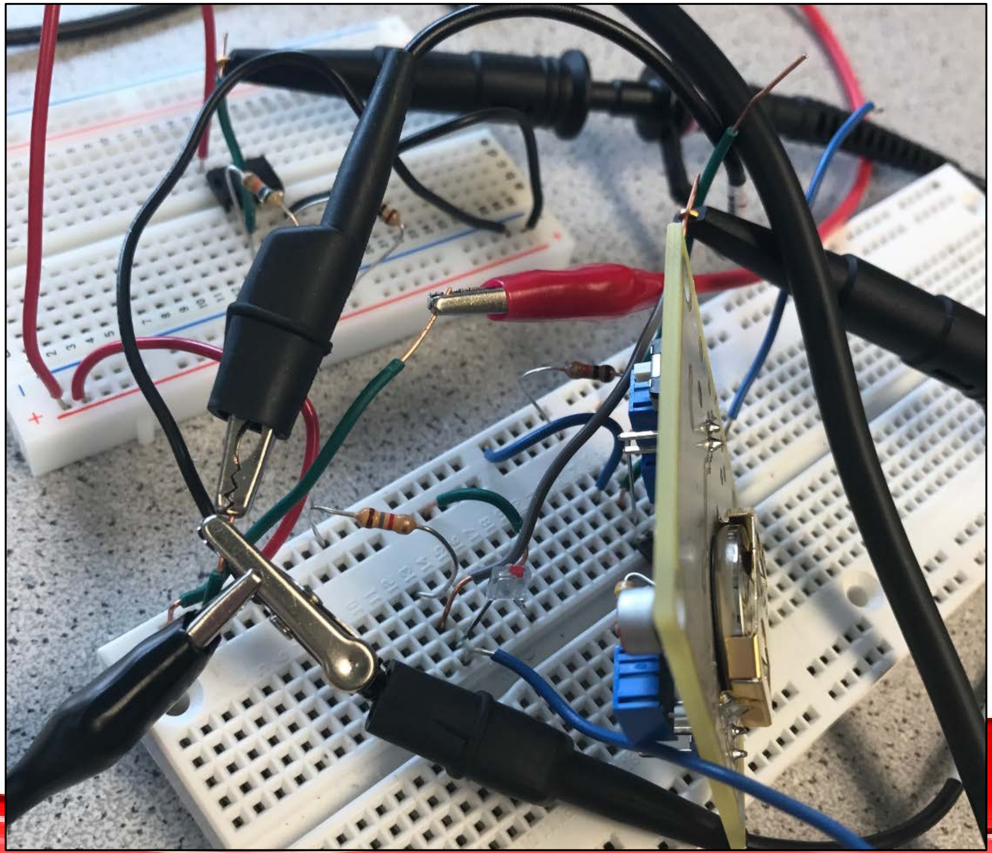
Quantifiable Metrics

Electronics

- Battery life time
  - (Under specific circumstance)
- Sensitivity of IR sensors
  - Output vs. distance
  - SNR
  - Resistance to ambient light
- Sensitivity of microphone
  - Output vs. distance
- Bandwidth of communication
- Computation speed/memory

- Filters…
- Multiplexers…
- etc.

The One-ders, 2017



Quantifiable Metrics

Software

- FFT
  - What is the Q of your filter?
- Search
  - How long does it take to find a path?
    - Worst case and best case scenarios
  - How does your implementation scale in time and memory with the size of the maze?

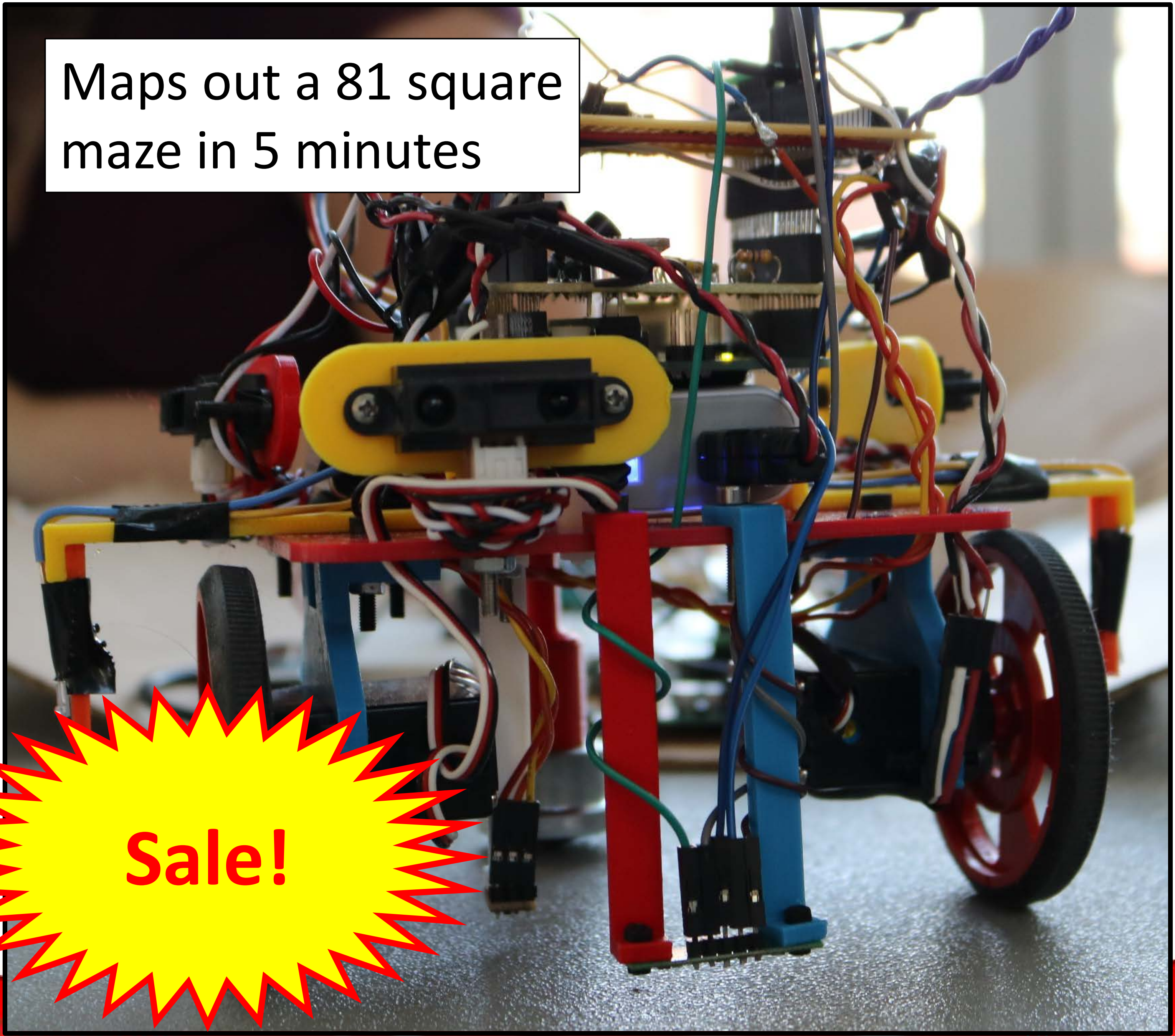
Simulation tools available at: https://cei-lab.github.io/ece3400-2018/tutorials/

Mechanics

- Speed/power of your servos
- Payload capability

Product Appraisal

- *How do you spec out your robot?*
- **How do you set yourself apart from the other 26 robots?**
  - Fast?

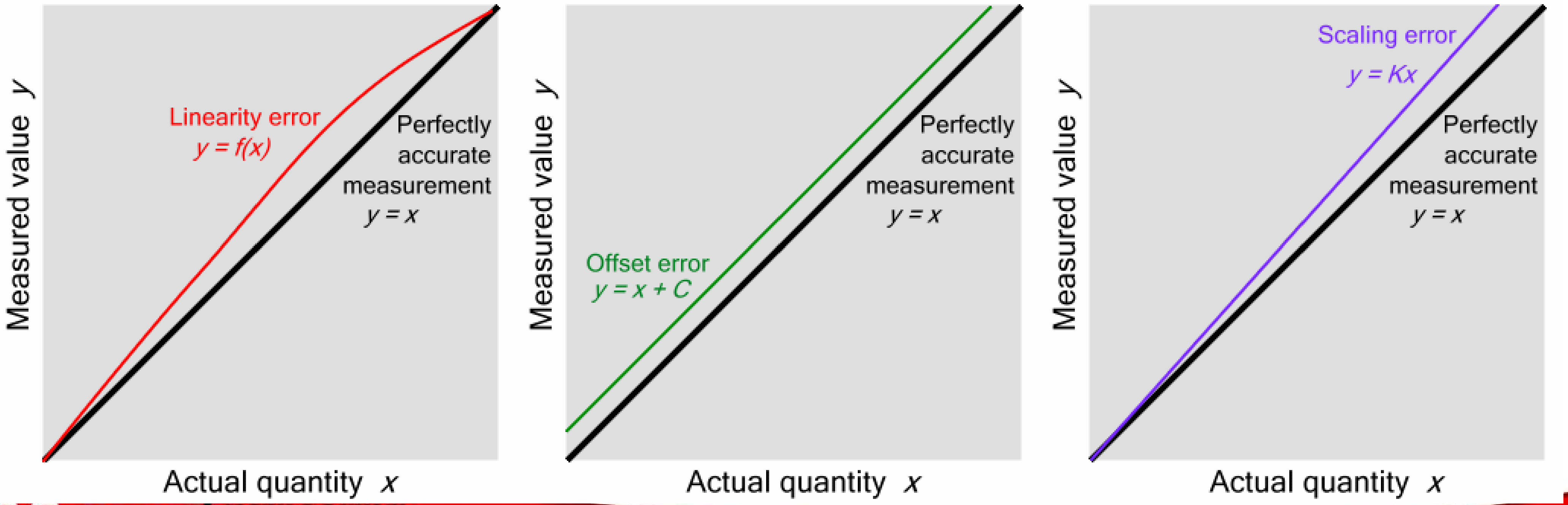


Maps out a 81 square maze in 5 minutes

Sale!

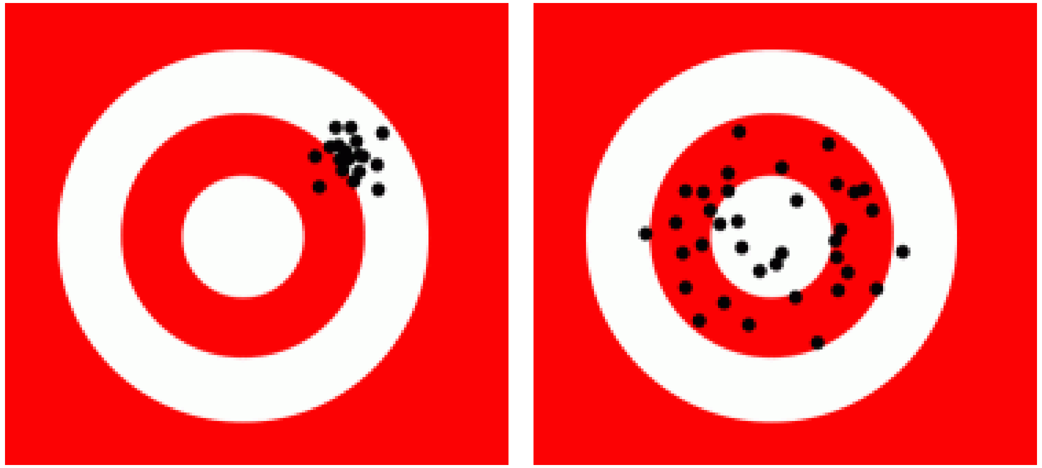
Accuracy vs Precision vs Resolution vs Sensitivity

- **Accuracy**
- The amount of uncertainty in the system with respect to an absolute standard.
  - Offset (independent of the amplitude of the input signal)
  - Gain (dependent on amplitude of the input signal)
  - Any biased noise
  - Accuracy is the sum of all of these, e.g. (0.1% of distance travelled +1cm)



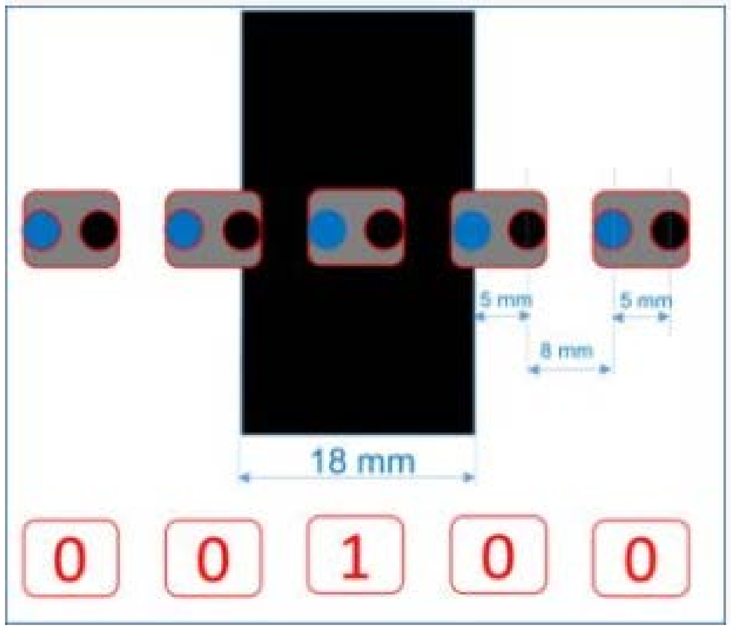
Accuracy vs Precision vs Resolution vs Sensitivity

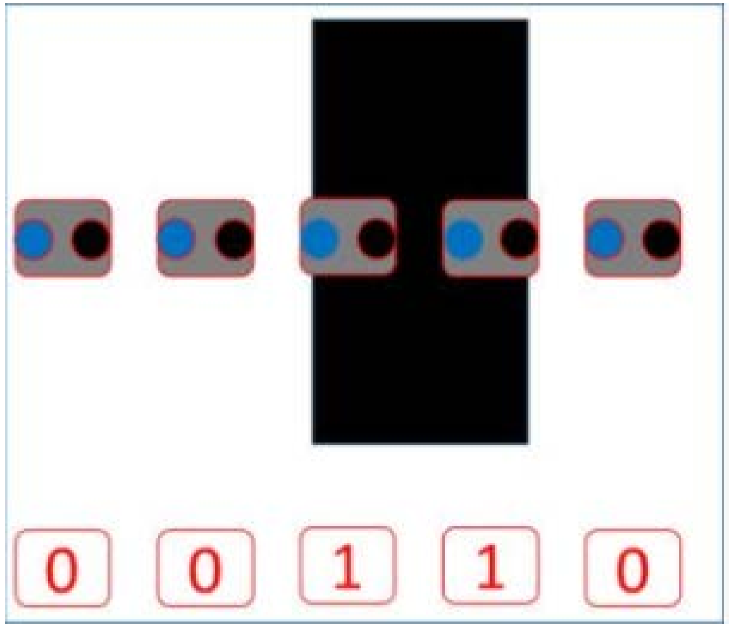
- **Precision**
- The reproducibility of the measurement.

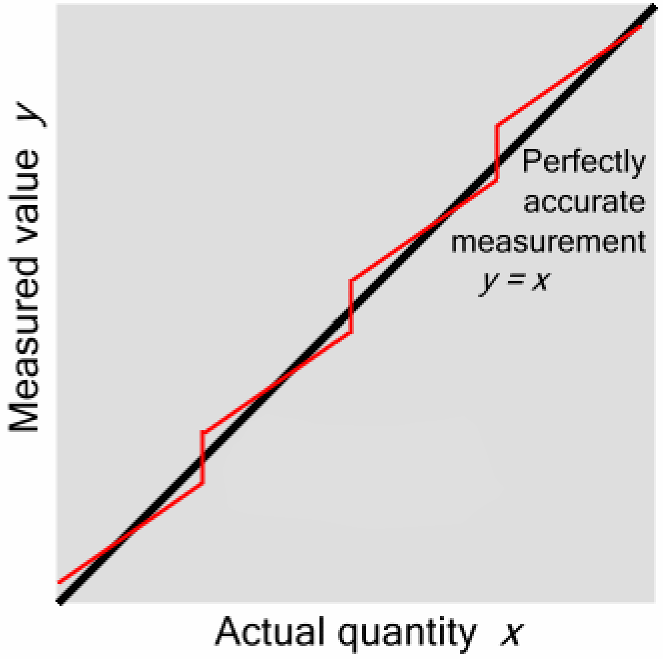


Accuracy vs Precision vs Resolution vs Sensitivity

- **Resolution**
- The ratio between the maximum signal measured to the smallest part that can be resolved
- (the degree to which a change can be theoretically detected)



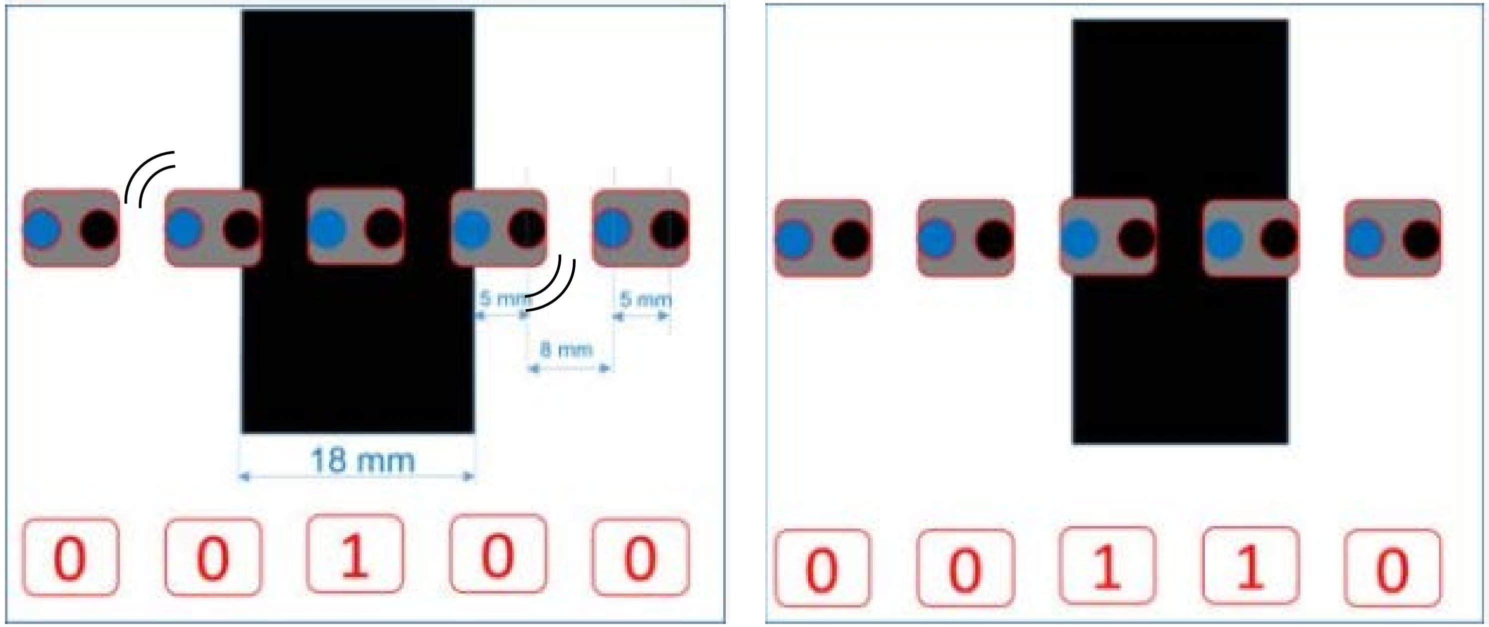




Accuracy vs Precision vs Resolution vs Sensitivity

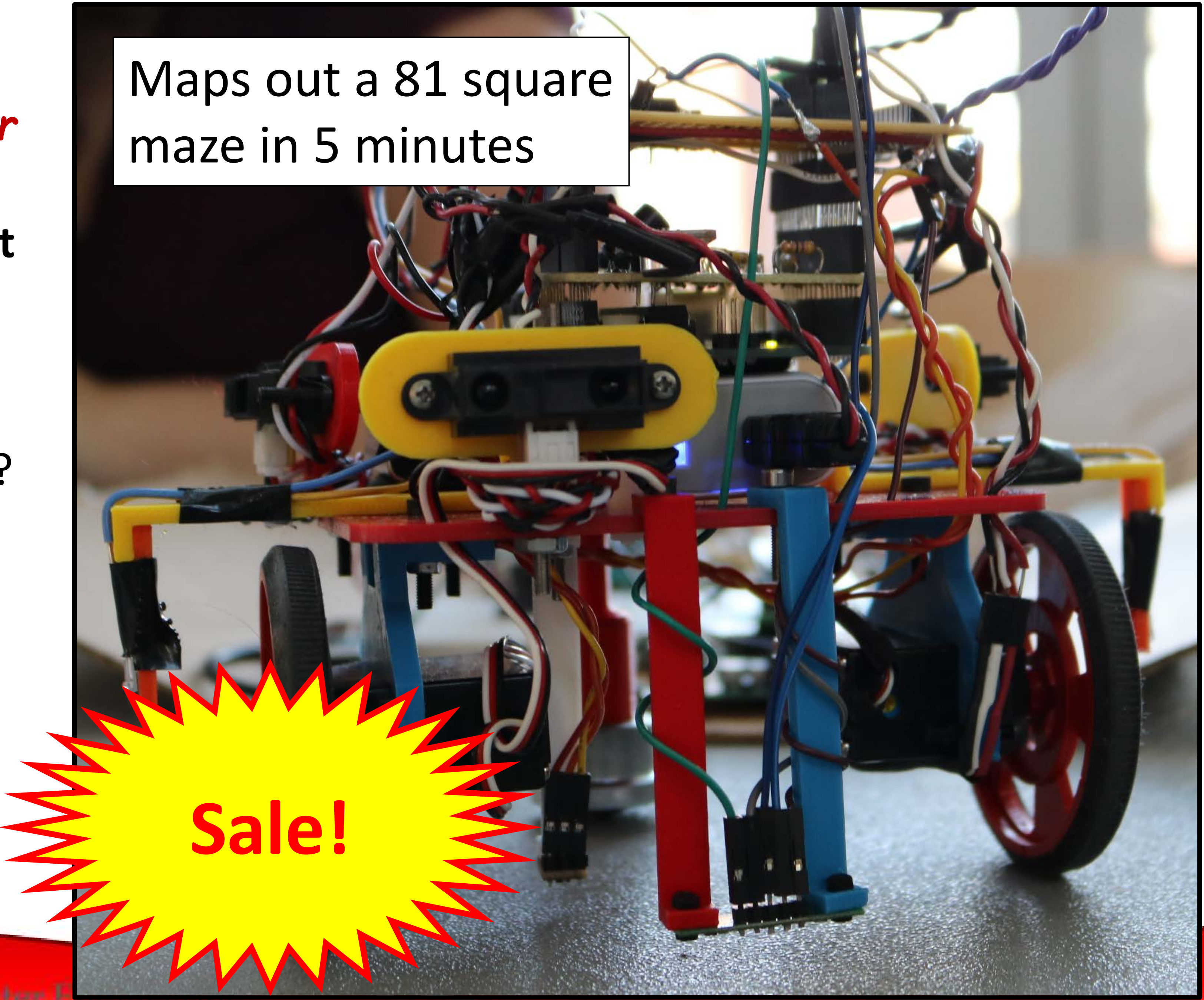
- **Sensitivity**
- The smallest absolute amount of change that can be detected by your robot.
- *What could cause resolution and sensitivity to be different in your line following?*

1. Noise picked up in the wires
2. Cross talk in the Mux
3. Mechanical vibrations
4. etc.



Product Appraisal

- *How do you spec out your robot?*
- **How do you set yourself apart from the other 26 robots?**
  - Fast?
  - Cheap? User friendly? Pedagogical? Entertaining?

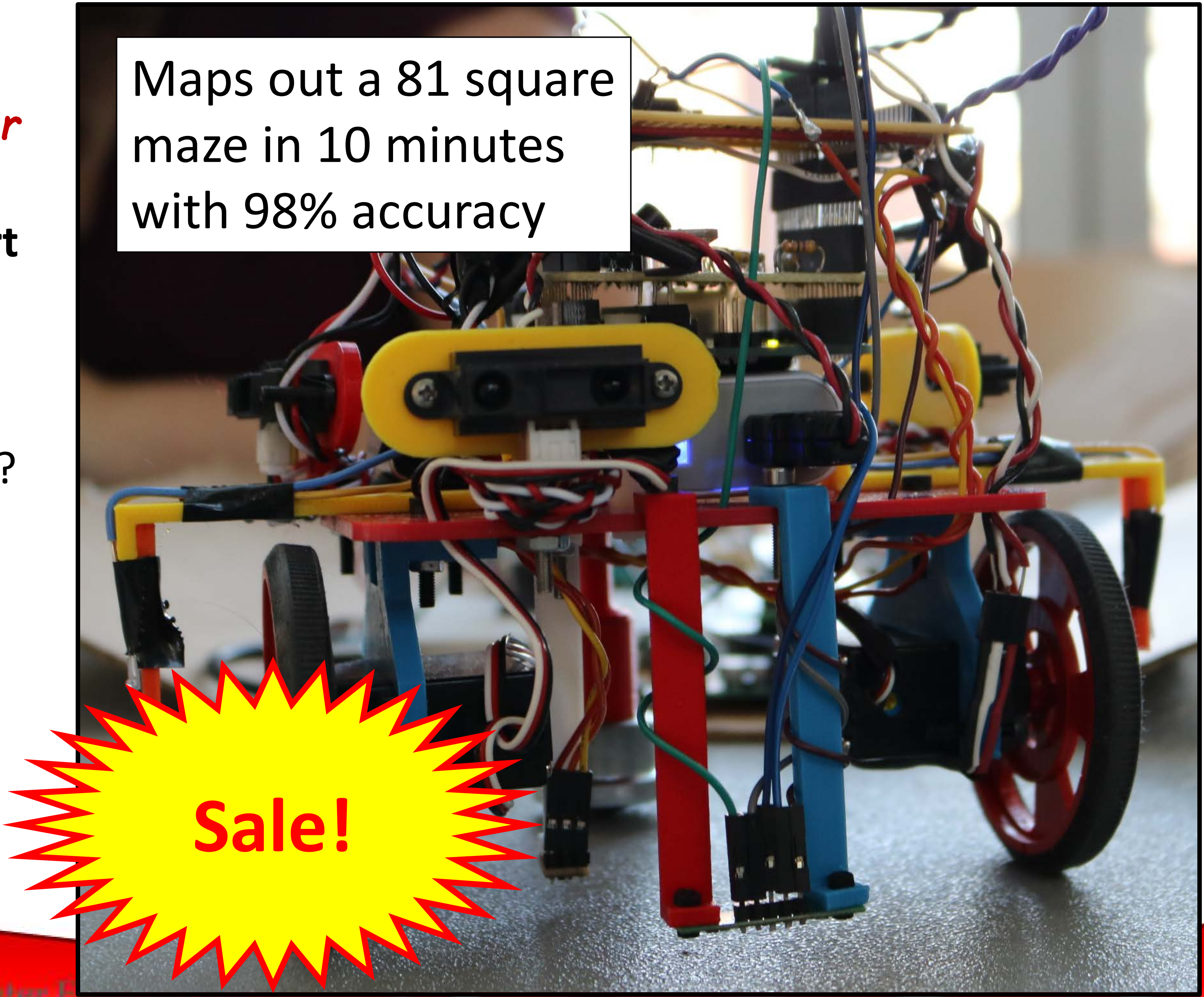


Maps out a 81 square maze in 5 minutes

Sale!

Product Appraisal

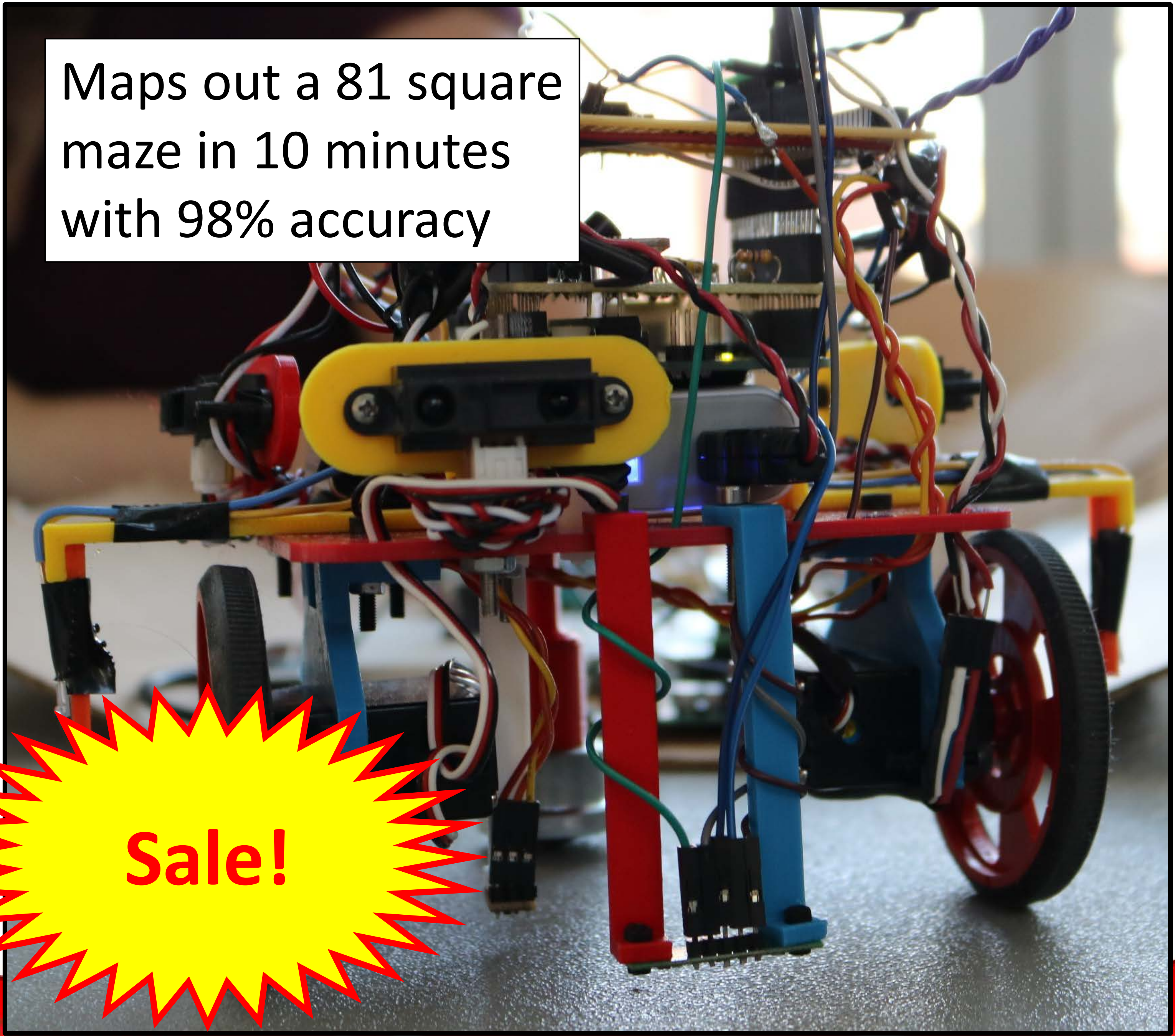
- *How do you spec out your robot?*
- **How do you set yourself apart from the other 26 robots?**
  - Fast?
  - Cheap? User friendly? Pedagogical? Entertaining?
  - Reliable?
    - Grid traversal
    - Grid turning
    - Wall detection
    - Treasure detection



Maps out a 81 square maze in 10 minutes with 98% accuracy

Product Appraisal

- **Engaging, thorough website**
- **Good robot specs**
- Capabilities
- Operating Conditions
- Goals
- Quantifiable Metrics
  - Speed
  - Reliability
  - Price
  - Competition: 18/20 points!
  - Award: Voted best team
  - Award: Voted best ethics
- = 15 points!



Maps out a 81 square maze in 10 minutes with 98% accuracy

Sale!

Reliability

- *How well does your robot go straight?*
- *How well does your robot turn?*
- *How well does your wall sensor detect walls?*
  - *...Are you really sure it works perfectly?*