AME 3623: Embedded Real-Time Systems

Andrew H. Fagg
Symbiotic Computing Laboratory
Biomedical Engineering Center and
School of Computer Science
University of Oklahoma

Teaching Assistant: Uma Kaipa

What is an Embedded System?

What is an Embedded System?

- Computing system with a non-standard interface (often no keyboard or screen)
- Often involved in sensing and control (and may not even talk to a human)
- Typically a custom system for a very specific application

What is an Embedded System? (cont)

- Limited processing capabilities:
  - Can be extremely small
  - Can require a small amount of power
- Can have significant real-time constraints
  - Act on inputs very quickly
  - Generate outputs that can change quickly
- Often a higher expectation of reliability

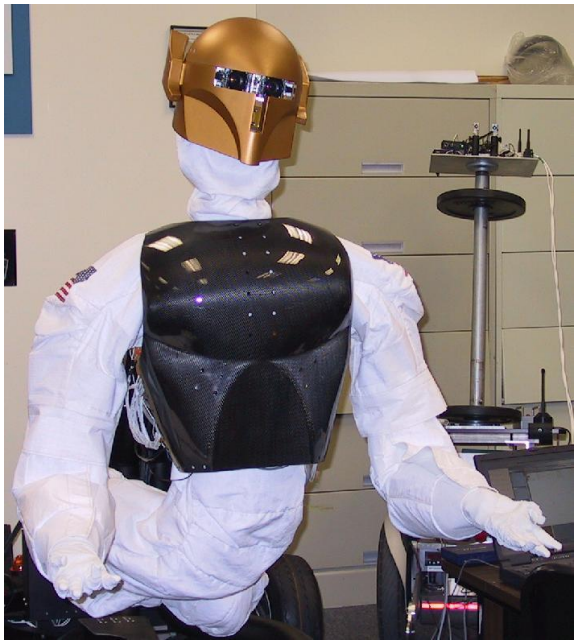
What is an Embedded System?

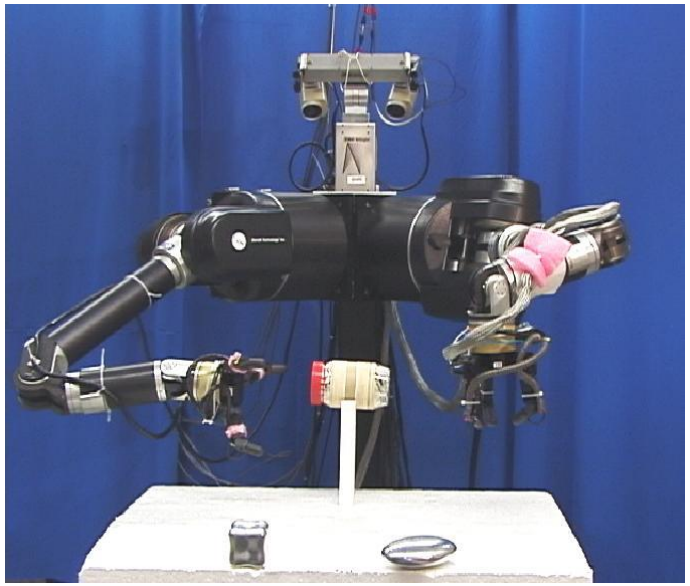
Fundamentally about the bit meeting the atom and the photon…

Examples of Embedded Systems

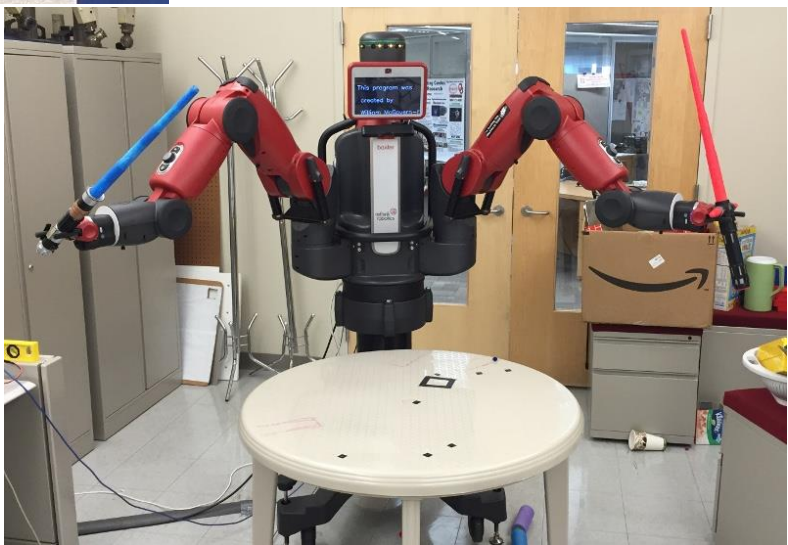
Humanoid Robotics

NASA/JSC Robonaut





UMass Torso



OU “Yatima”

Robotics

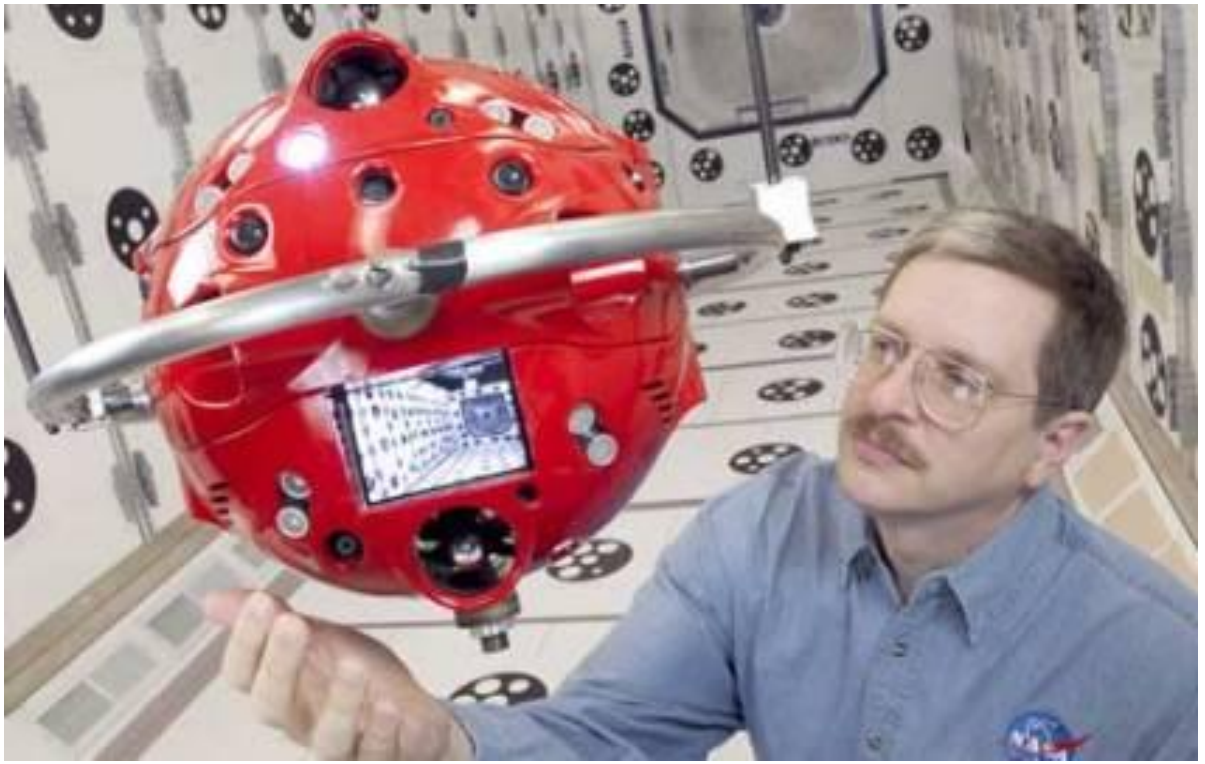
Mark Tilden

Los Alamos National Labs and Wowwee

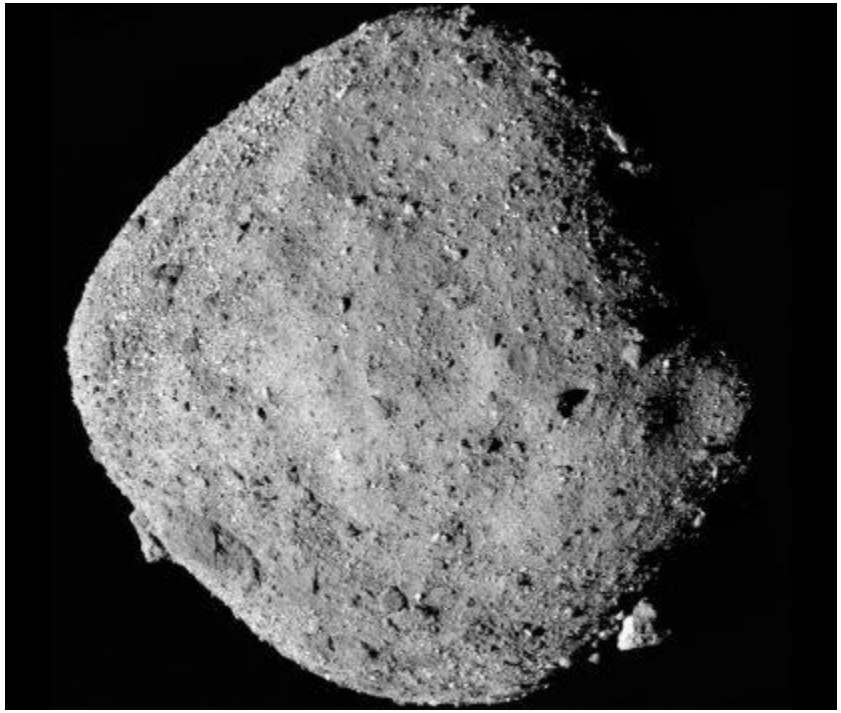
picture from *Robosapiens*

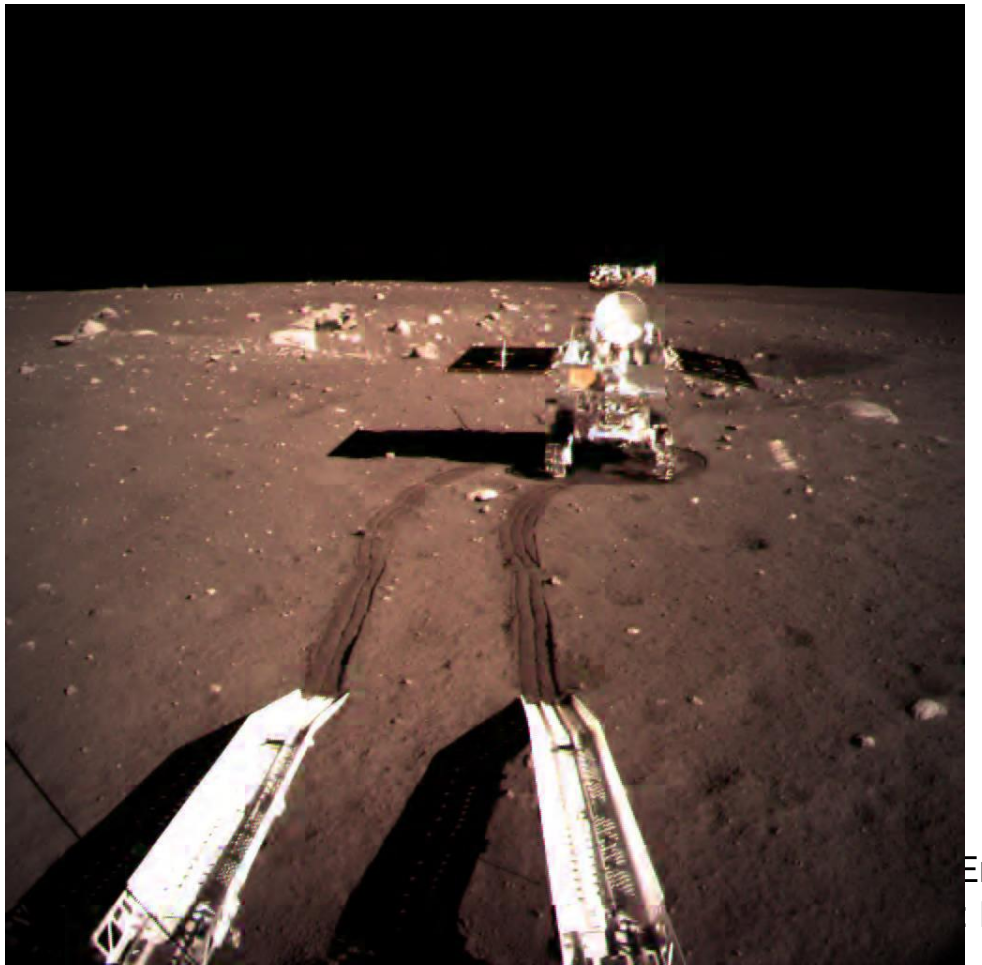
Personal Satellite Assistants

NASA Ames Research Center

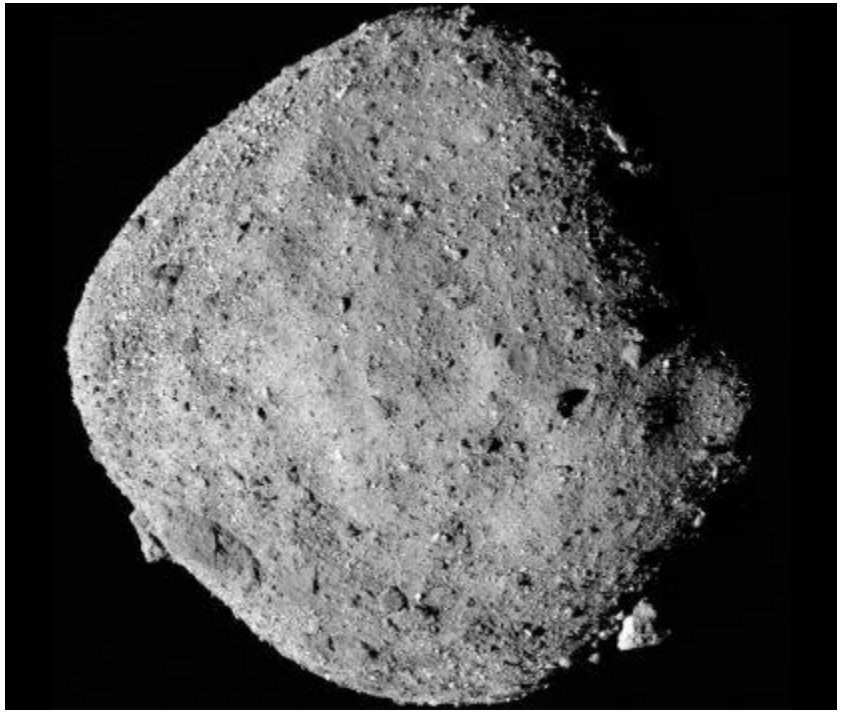


Space Missions

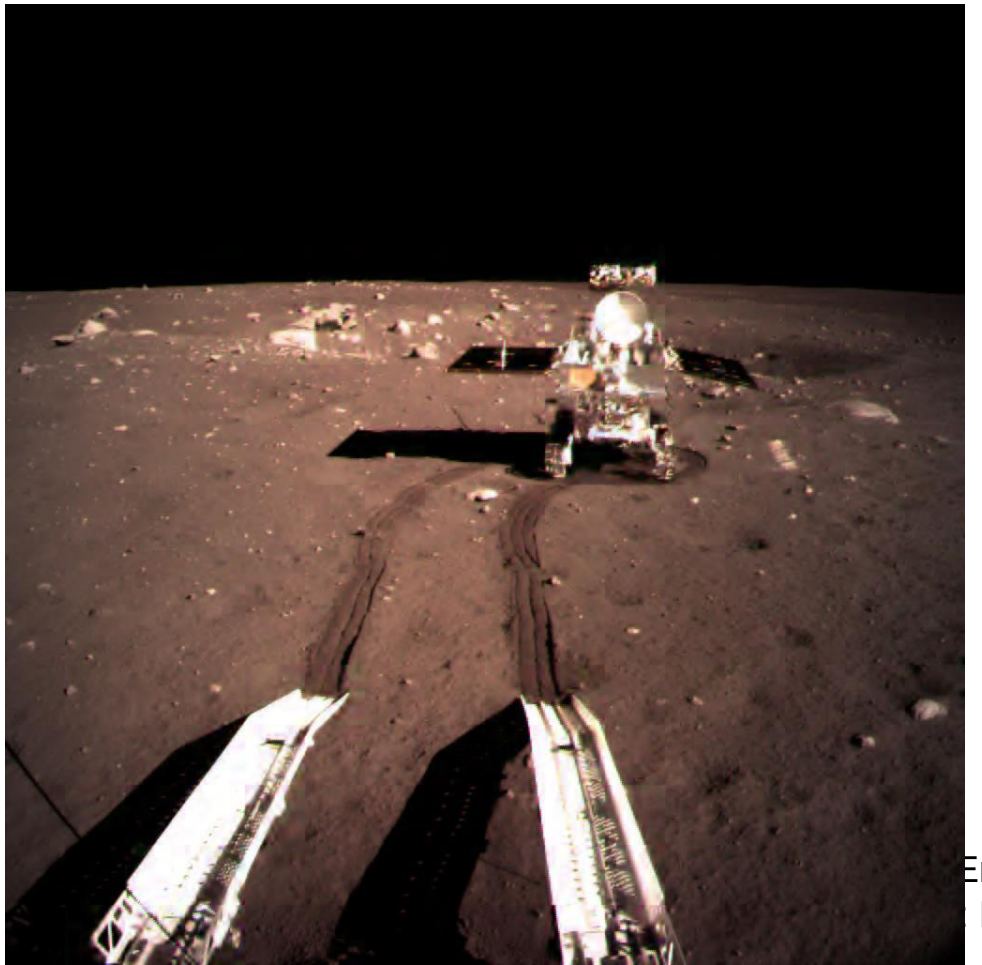




Space Missions



Bennu



Chang'e 4

Intelligent Prosthetics

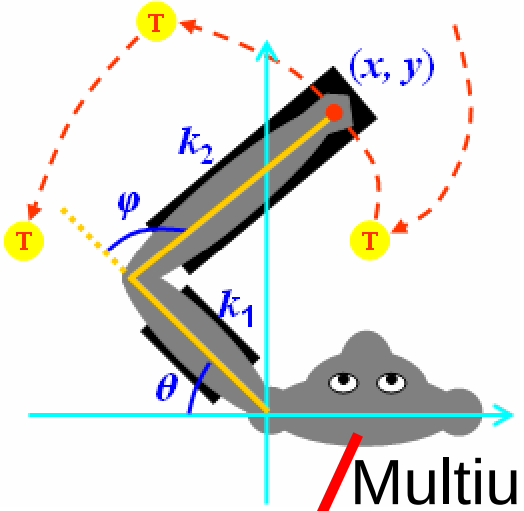
Hugh Herr
MIT Leg Lab

picture from
Robosapiens

Brain-Machine Interfaces

Estimate of intended movement

Command prosthetic arm



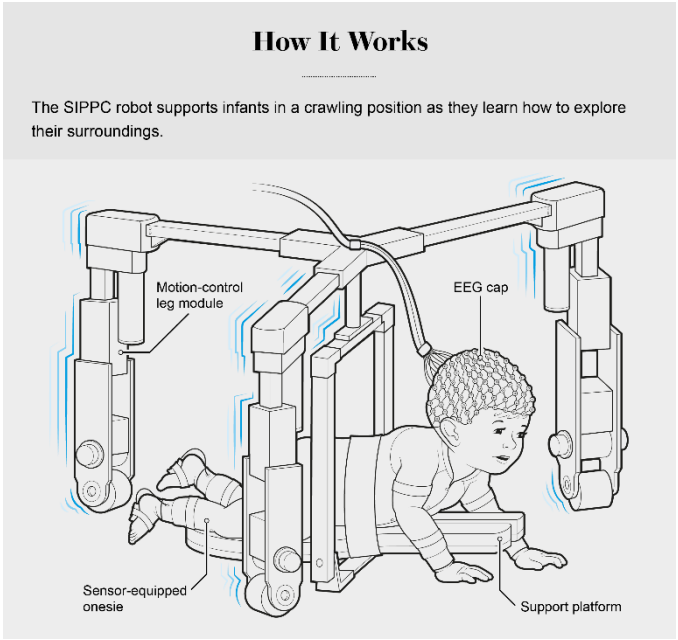
Multiunit recording

Predictive model

In collaboration with Nicholas G. Hatsopoulos and Lee E. Miller

Real-Time Activity Recognition for Assistive Robotics

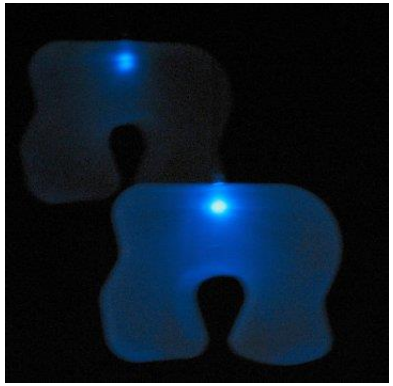
OU Crawling Assistant
(Kolobe, Fagg, Miller, Ding)

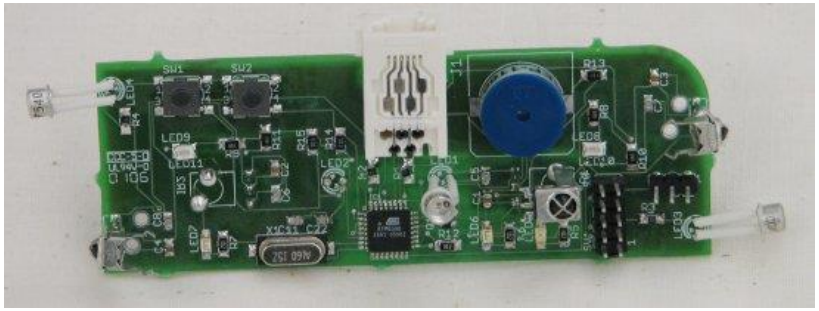


Scientific American (Oct 2016)

Sensor Networks

1000 sensor nodes





Embedded Systems Challenges

Embedded Systems Challenges

- Sensing the environment:
  - Sensors are typically far from ideal (noise, nonlinearities, etc.)
  - Sensors/subsystems can fail
  - Hard to get a ‘complete’ view of the environment
- Affecting the environment through “actuators”
  - Application can require fast, precise responses

Embedded Systems Challenges (cont)

- Testing/debugging can be very difficult:
  - Hard to identify and replicate all possible situations
  - Often involves the interaction of many different components
  - Often no standard user interface
  - Limited on-board resources with which to record system state

- Competing requirements of cost, complexity, design time, size, power…

Embedded Systems Challenges (cont)

- Lack of reliability can be a killer ….. literally

My Assumptions About You

- Circuits and sensors class (or equivalent):
  - Boolean logic
  - Analog circuits (in particular, resistive-capacitive circuits)
- One course background in programming
  - We will be using C for all projects
- Everyone has a functional laptop that can be used for the projects

How We Will Proceed

Embedded Systems is a mix of software, circuit, actuation, sensing and control

- Blend of theory and practice
- Practice requires practice:
  - Very hands-on
  - Don’t be afraid to try things (but care must be taken)

Course Goals

In this course, you will:

- Design embedded circuits,
- Use code design tools,
- Design, program and debug embedded sensing and control software,
- Work in teams, and
- Practice your communication skills.

Sources of Information

- Primary readings:
  - Book: *Programming Embedded Systems* from Zyante
  - Selected web pages
- Class web page:

  www.cs.ou.edu/~fagg/classes/ame3623
- Canvas: canvas.ou.edu

You are responsible for making sure that you have access to all of these resources


HOW TO SURVIVE A ROBOT UPRISING
TIPS ON DEFENDING YOURSELF AGAINST THE COMING REBELLION
Daniel H. Wilson

Preparing for Lecture

- Readings: you are responsible for reading these **before** our class time.
- The Zyante book includes a set of questions listed under “Participation Activities” in each chapter. Doing these questions is your homework (due by **8am** the day that the reading is assigned).
  - “Challenge Activities” are not required
- In class, we will address any questions that you have about the materials, expand on what you have already done, and do quizzes/in-class exercises (some of which will be graded)

Class Schedule

www.cs.ou.edu/~fagg/classes/ame3623/schedule.html

- Lecture plans
- Required reading
- Assignments
- Due dates

Note: this schedule can change

Channels of Communication

- Lecture
- Canvas announcements
  - You can configure Canvas to send you an email message or an alert on your phone every time a message is posted
- Canvas discussion group: you may post questions and answers
- Private email or office hours for non-public questions/discussions

Grading

- Components of your grade:
  - Midterm exam: 10%
  - Final exam: 20%
  - Zyante activities: 15% (Keep highest N-1)
  - In-class quizzes and exercises: 15% (Keep highest M-1)
  - Eleven small projects:
    - 25% the work by your group
    - 15% your personal contributions to programming
- Grades will be posted on Canvas
- Final letter grades boundaries will be selected based on the overall class distribution

Exams

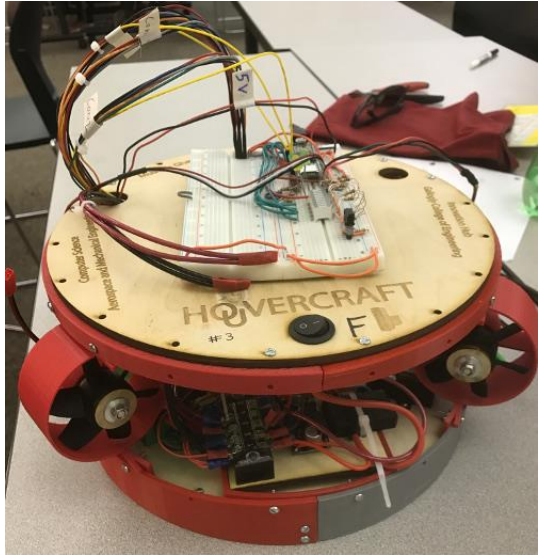
- Closed book/closed notes
  - Exception: you are allowed 1 page of your own notes
- No electronic devices
- Multiple choice
  - Preliminary grades are possible as you exit the exam
- Grading questions must be addressed within one week of return

In-Class Exercises

- Expand on readings
- Mixture of individual and group work
- Often graded

Group Projects

Focus: hovercraft control system

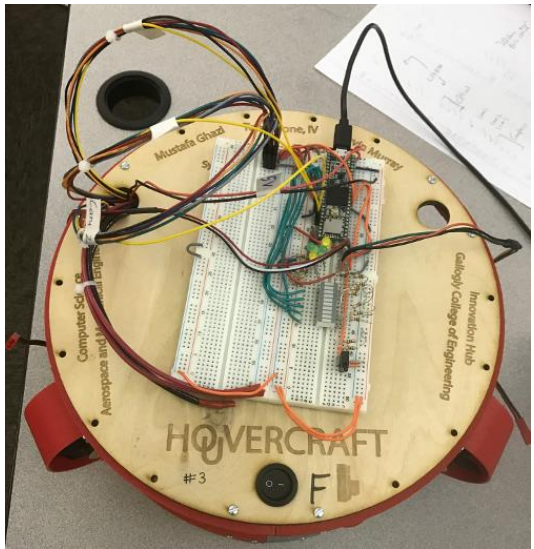


Each project:

- Discuss new project on Thursday
- In most cases: due the following Thursday (8:00am)
- Different components:

  circuit, mechanical, software,

  documentation



Project Topics

- Embedded processor programming (we are using “Teensies” this year)
- Analog processing and sensor models (distance sensing)
- Sensing lateral velocity
- Sensing heading and heading change
- Proportional-derivative control
- Finite state machines for mission-level control

Project Grading

Group grades are a function of:

- Code correctness and readability
- Documentation of code and circuits
- Demonstration

These are assessed during a short “code review” with me or the TA

Project Grading

Individual grades:

- Group grade scaled by the degree of your contribution to the group work (generally, this is balanced)
- Personal software contributions: must accumulate 3 software components over the course of the semester (1 available for each project)

Group Projects (cont)

- Lab space: Felgar Hall 300
  - Will work out room availability soon
- Groups will be of size 2-3 and will be assigned
- Be ready to demonstrate project by the due date
- Projects require more than a day to complete
- Code/documentation will be handed in through your group's Dropbox
- Projects may be late (but I do not recommend this): 0-24 hrs: 10% penalty; 24-48 hrs: 20% penalty; 48+ hrs: 100% penalty

Classroom Conduct

- Ask plenty of questions
- Contribute to the discussions

- No: cell phone use and laptop use (except for classroom exercises)

***More details in the syllabus

Proper Academic Conduct

Homework assignments (Zyante) :

- All work must be your own: no looking at or copying solutions from other students or from the net
- General discussion is okay (e.g., the fundamental skills that we are learning)

- When in doubt: ask me or the TA

Proper Academic Conduct

Projects:

- All work must be that of your group: no looking at, discussing or copying solutions from other groups or from the net
- General discussion is okay
- Software components: everyone must take primary responsibility for at least three

Secure your data

Next Time

- Zyante registration: access through Canvas
- Readings:
  - Introduction to Embedded Systems
  - Analog circuits review
  - Diodes
- Catme survey coming soon via email