AME 3623:
Embedded Real-Time Systems

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

Teaching Assistant: Sean McDonough

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)

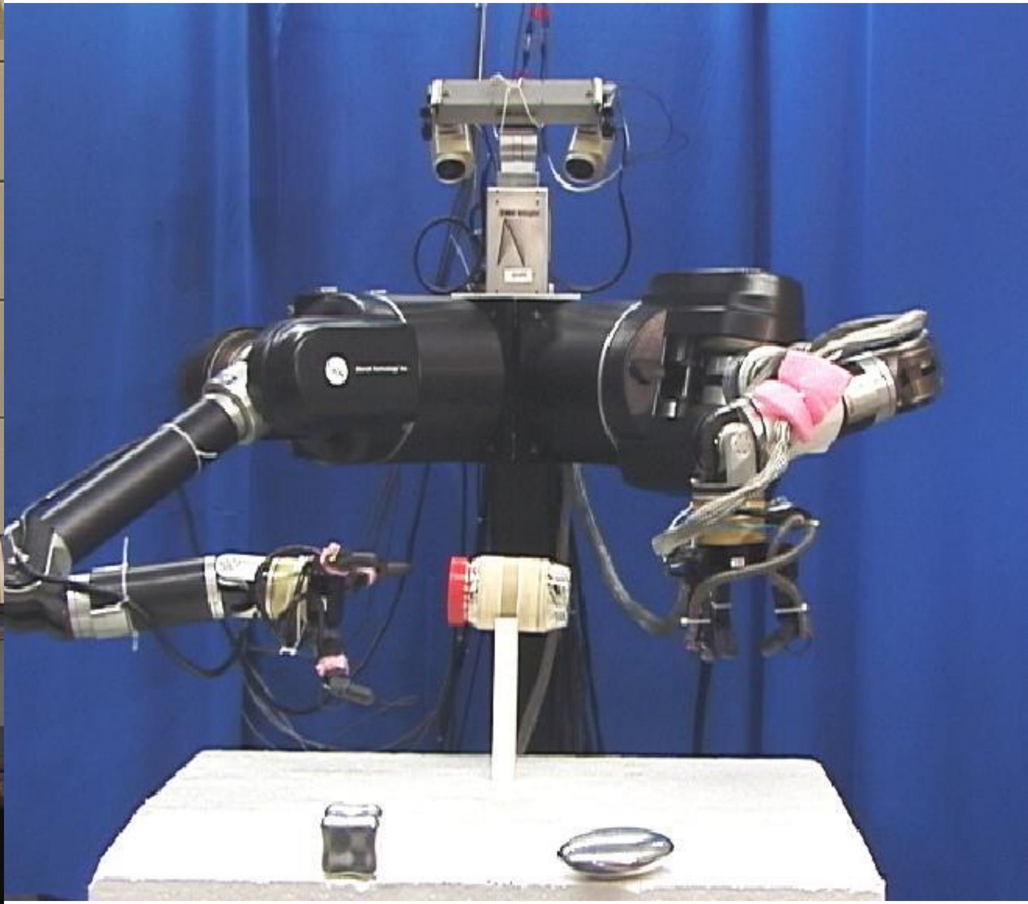
- Limited processing capabilities:
  - Can be extremely small
  - Can require a small amount of power
- Can have significant real-time constraints
  - Act on inputs very quickly
  - Generate high-frequency outputs
- Often a higher expectation of reliability

Examples of Embedded Systems

Humanoid Robotics

NASA/JSC Robonaut

UMass Torso



Robotics

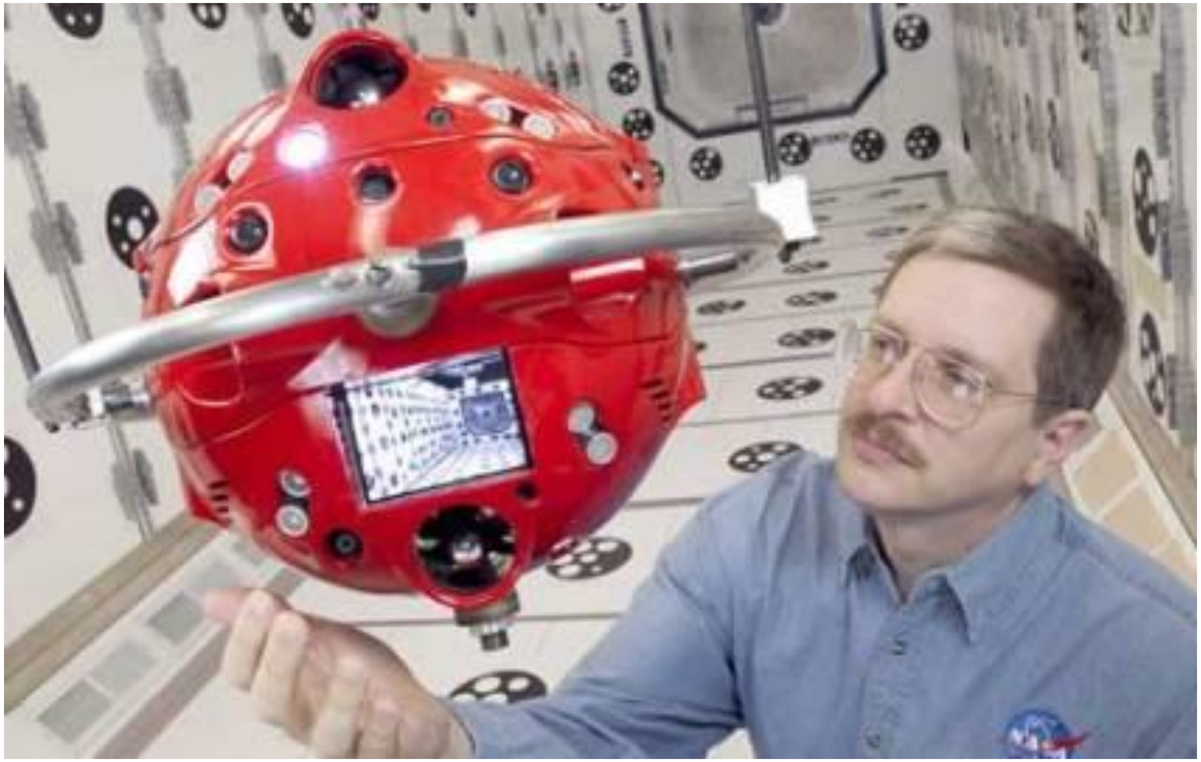
Mark Tilden

Los Alamos National Labs and Wowwee

picture from *Robosapiens*

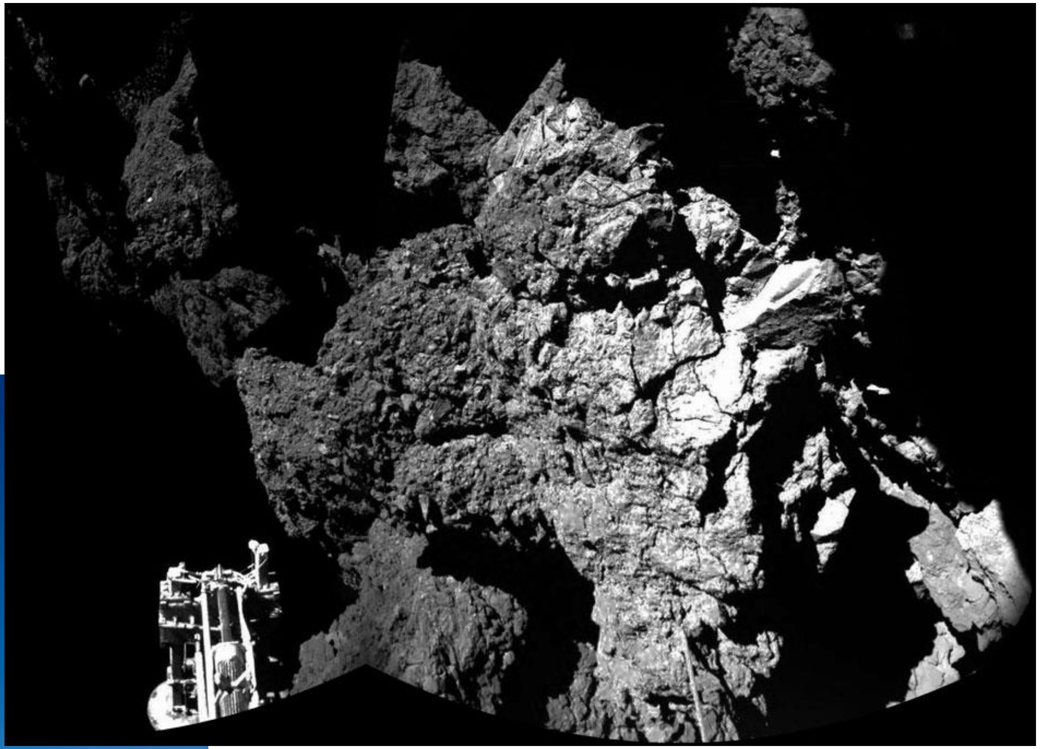
Personal Satellite Assistants

NASA Ames Research Center



Space Missions

Space Missions



Rosetta

Intelligent Prosthetics

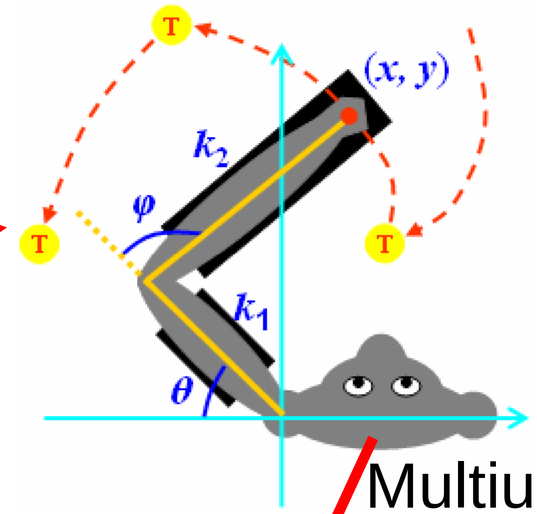
Hugh Herr

MIT Leg Lab

picture from *Robosapiens*

Brain-Machine Interfaces

Command prosthetic arm



Estimate of intended movement

Multiunit recording

Predictive model

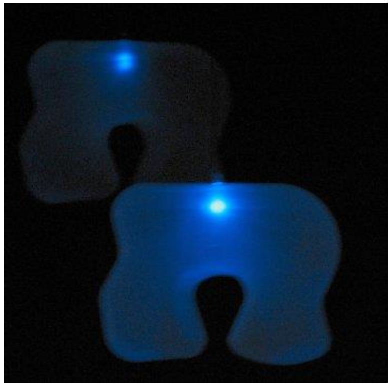
In collaboration with Nicholas G. Hatsopoulos and Lee E. Miller

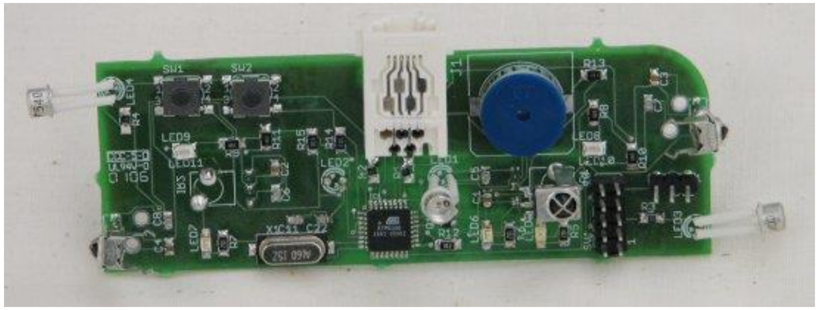
Real-Time Activity Recognition for Assistive Robotics

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

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 and circuits (AND/OR/NOT gates)
  - 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

Course Goals

By the end of this course, you will be able to:

- design and implement embedded circuits involving microcontrollers, sensors and actuators,
- use code and circuit design tools,
- design, program and debug embedded sensing and control software,
- work in collaborative teams to solve system design and implementation challenges, and
- communicate in both oral and written forms.

Sources of Information

- Primary readings:
  - Book: *Programming Embedded Systems* from Zyante
  - Selected web pages
  - Pencasts: recorded audio & writing
- Class web page: www.cs.ou.edu/~fagg/classes/ame3623
- Desire2Learn: learn.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

“Flipped” Class Structure

- Readings and pencasts: you are responsible for reading/viewing 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 & is due by **8am** the day of the assignment.
- In class, we will address any questions that you have about the materials, expand on what you have already done and do in-class exercises (some of which will be graded)

Class Schedule

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

- Lecture plans
- Required reading and pencast viewing
- Assignments
- Due dates

Note: this schedule can change

Channels of Communication

- Lecture
- Class email list: time-critical messages to the class
- Desire2Learn announcements
- Desire2Learn 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% (drop lowest)
  - In-class quizzes and exercises: 20% (drop lowest)
  - Eleven small projects: 35%
- Grades will be posted on the Desire2Learn
- Final letter grades boundaries will be selected based on the overall class distribution

Exams

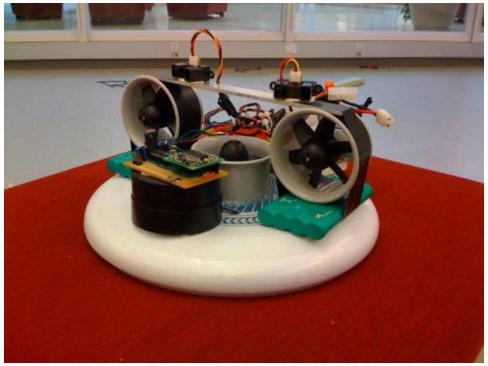
- Closed book/closed notes
  - Exception: you are allowed 1 page of your own notes
- Assigned seating
- No electronic devices
- Grading questions must be addressed before the returned exams leave the classroom

In-Class Exercises

- Expand on readings & pencasts
- Mixture of individual and group work
- Some will be graded

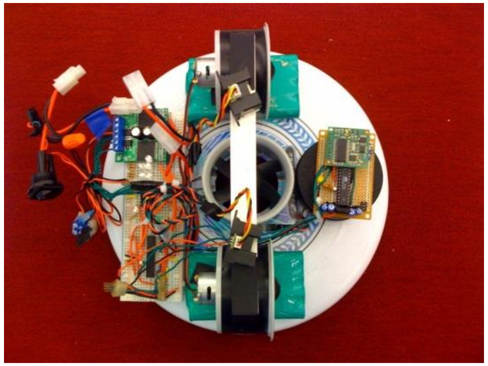
Expect short individual quizzes at the beginning of class

Group Projects



Focus: hovercraft control system

- Each project:
  - Reserve a lecture period to start and mostly complete the project
  - Generally one week to complete the rest
  - Different components: circuit, mechanical, software, documentation
- Eleven projects in total



Project Grading

Group grades are a function of:

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

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 handed in through your group's "subversion" tree
- 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 OK (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 (again) OK
- Software components: everyone must take primary responsibility for at least three

Secure your data

Next Time

- Complete Catme survey (see email)
- Tophat and Zyante registration
- Readings and pencasts:
  - Analog circuits review
  - Diodes