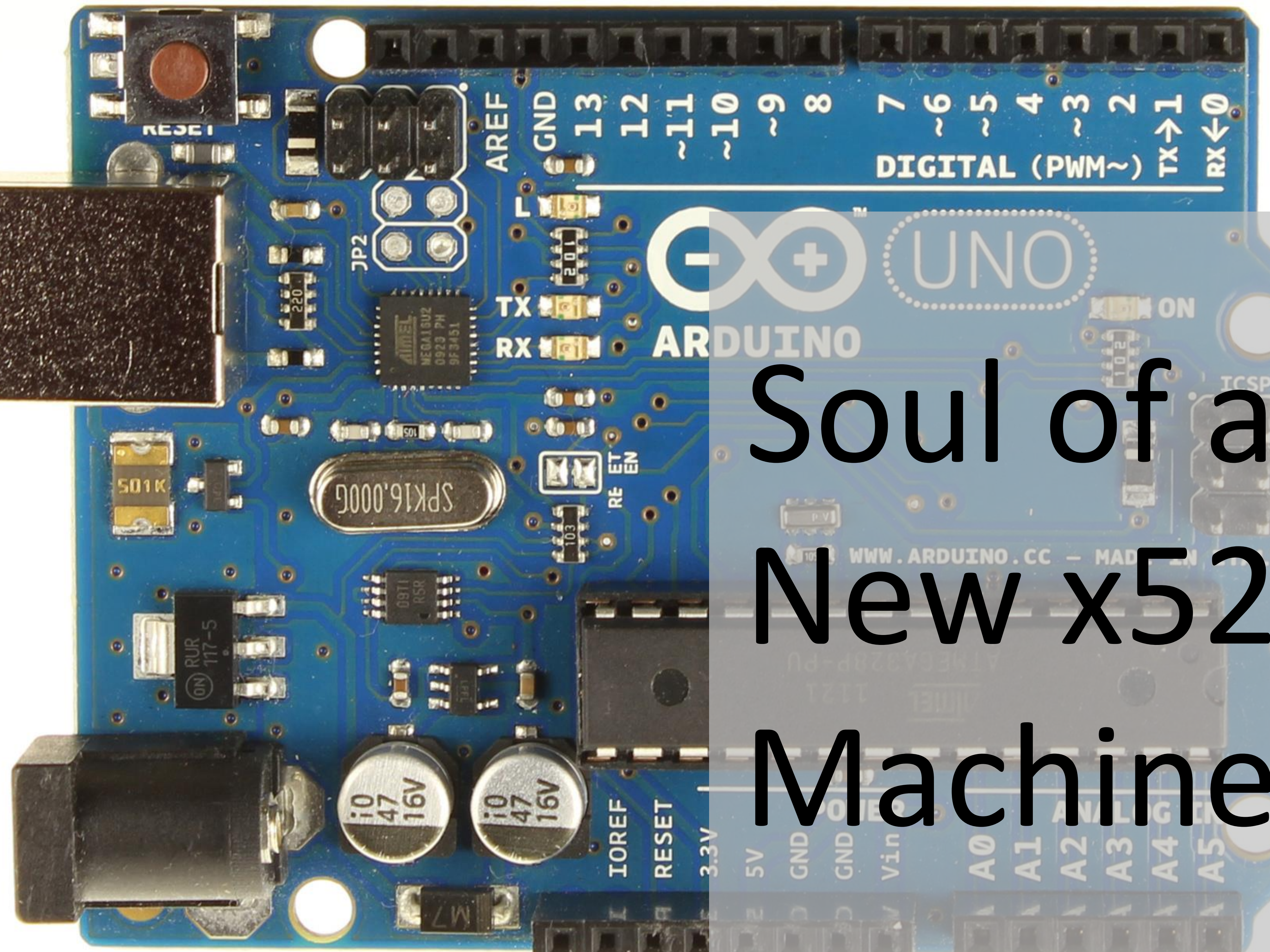
Soul of a New x52 Machine



Outline

- Background & overview of CS Dept and me
- Soul of a new Freshman CS252 Machine
  - Teaching Computer Science by Building Computers
  - Feedback and thoughts from Katie and Peter
- Soul of a new Senior CS552 Machine
  - Building the very chip used in Freshman year: Theo
- Connections back to research
  - Powering future datacenters with these ideas!
    Memory Processing Units: Theo

UW-CS 50 Years of Teaching & Research

- July 1964 founded as 2nd CS department
- Over 6,000 graduates who are flourishing in:
  - Companies: built, run and more: AOL, Autodesk, Epic, Microsoft, Oracle, Palo Alto Networks, Rocket Fuel Media, WebMD, and Yahoo!
  - Academia: Top-ten CS schools including: Berkeley, Carnegie Mellon, Cornell, Georgia Tech, Illinois, Stanford, Texas, and Washington.
- Research
  - Early Internet development, Microprocessor innovations w/ a billion shipped, Computing foundation for finding Higgs boson, Fundamental advances in graphics & approximation, principles of data management for "big data"

About me: 2007 - now

- Research: Building better microprocessors
  - 3.95 PhDs, 11 Masters students, 11 patents

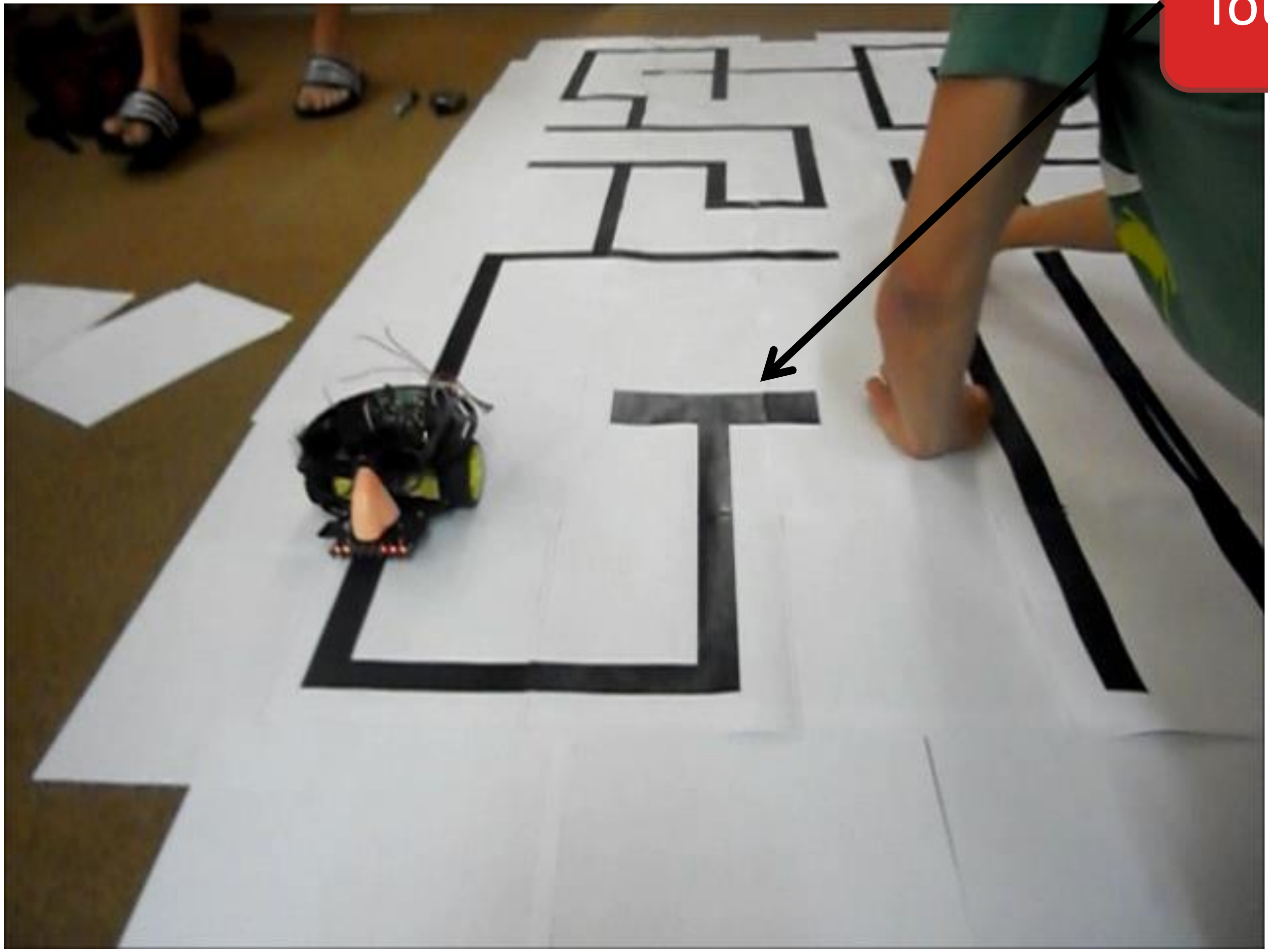
4th student is defending Oct 30th ☺

- Teaching: Freshman, senior undergrad, grad courses
- Select publications
  - Memory Processing Units, Hotchips 2014 Poster, Best Poster award, co-authored with **Theo Dahlen**
  - "A General Constraint-centric Scheduling Framework for Spatial Architectures", *PLDI Distinguished Paper award, CACM Highlights nomination (4 of about 400 papers awarded yearly)*, presented by under-grad **Michael-Sartin Tarm**
  - "Hands-on Introduction to Computer Science at the Freshman Level", SIGCSE, **4 under-grad student authors**

Outline

- Background & overview of CS Dept and me
- **Soul of a new Freshman CS252 Machine**
  - **Teaching Computer Science by Building Computers**
  - **Feedback and thoughts from Katie and Peter**
- Soul of a new Senior CS552 Machine
  - Building the very chip used in Freshman year: Theo
- Connections back to research
  - Powering future datacenters with these ideas!
    Memory Processing Units: Theo

Spring 2012: Freshman project



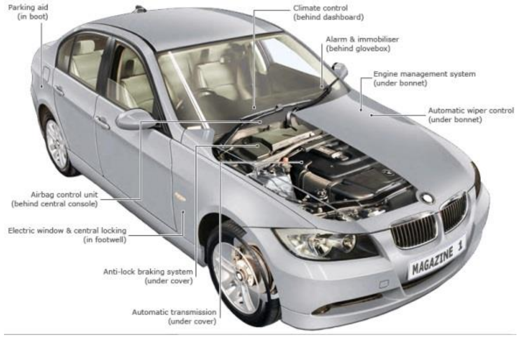
Touchdown!

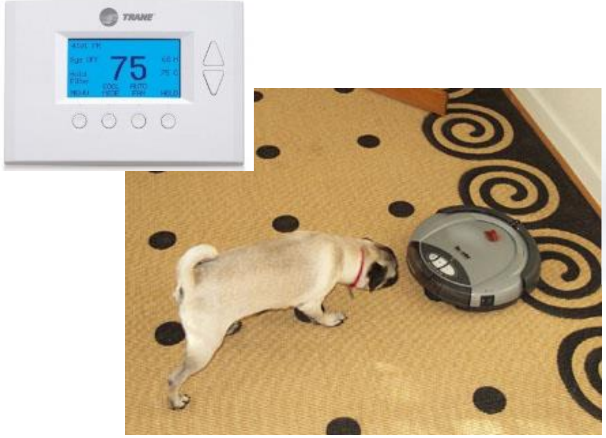
Hobbyist Computing in 80s

30 years ago, computers not ubiquitous, but…

building your own computer was cool, fun, educational, and common

Today, computers everywhere…





Can they learn by building a computer?

Better pedagogy and more fun

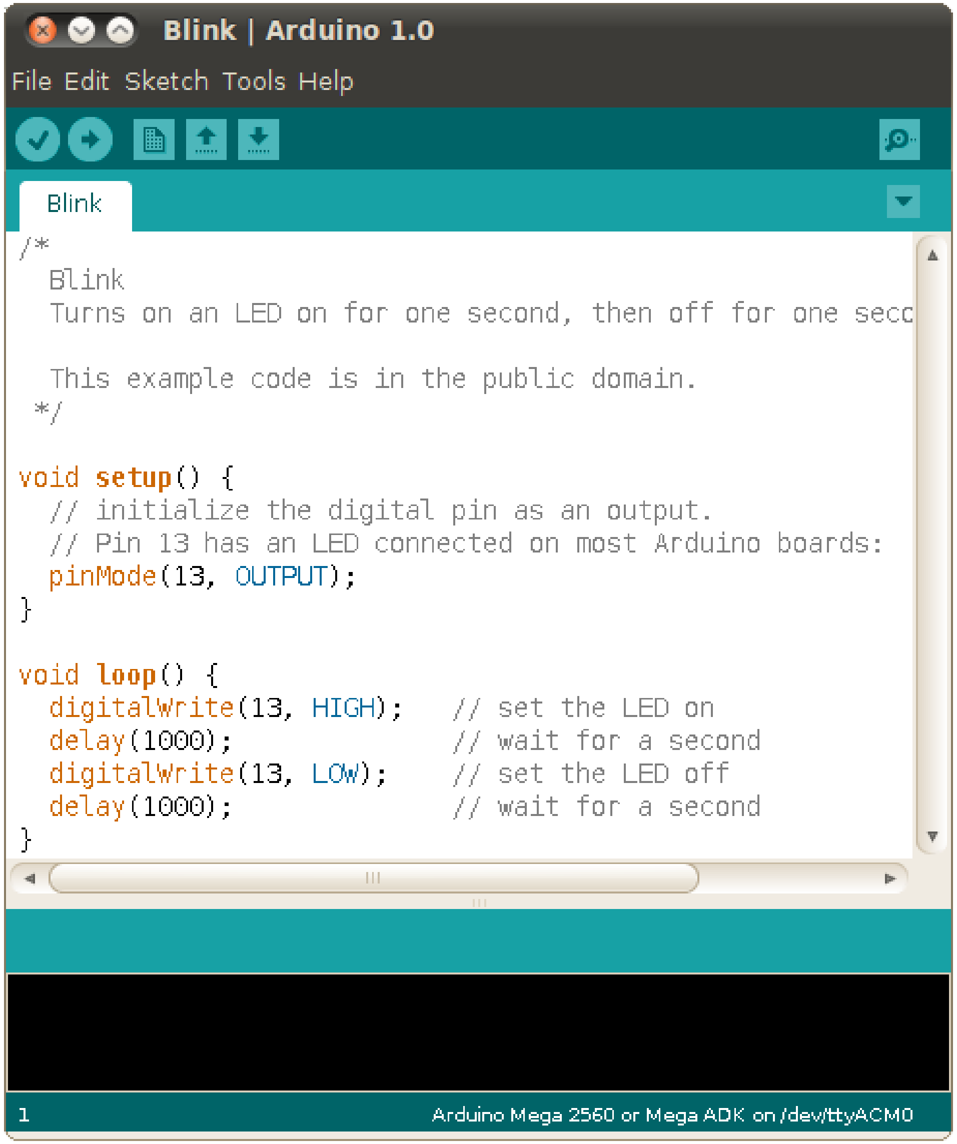
Arduino

Atmel chip, 14 digital input/output pins, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, 32KB Flash, 2KB SRAM

Costs $75

Family of Plugin Extensions

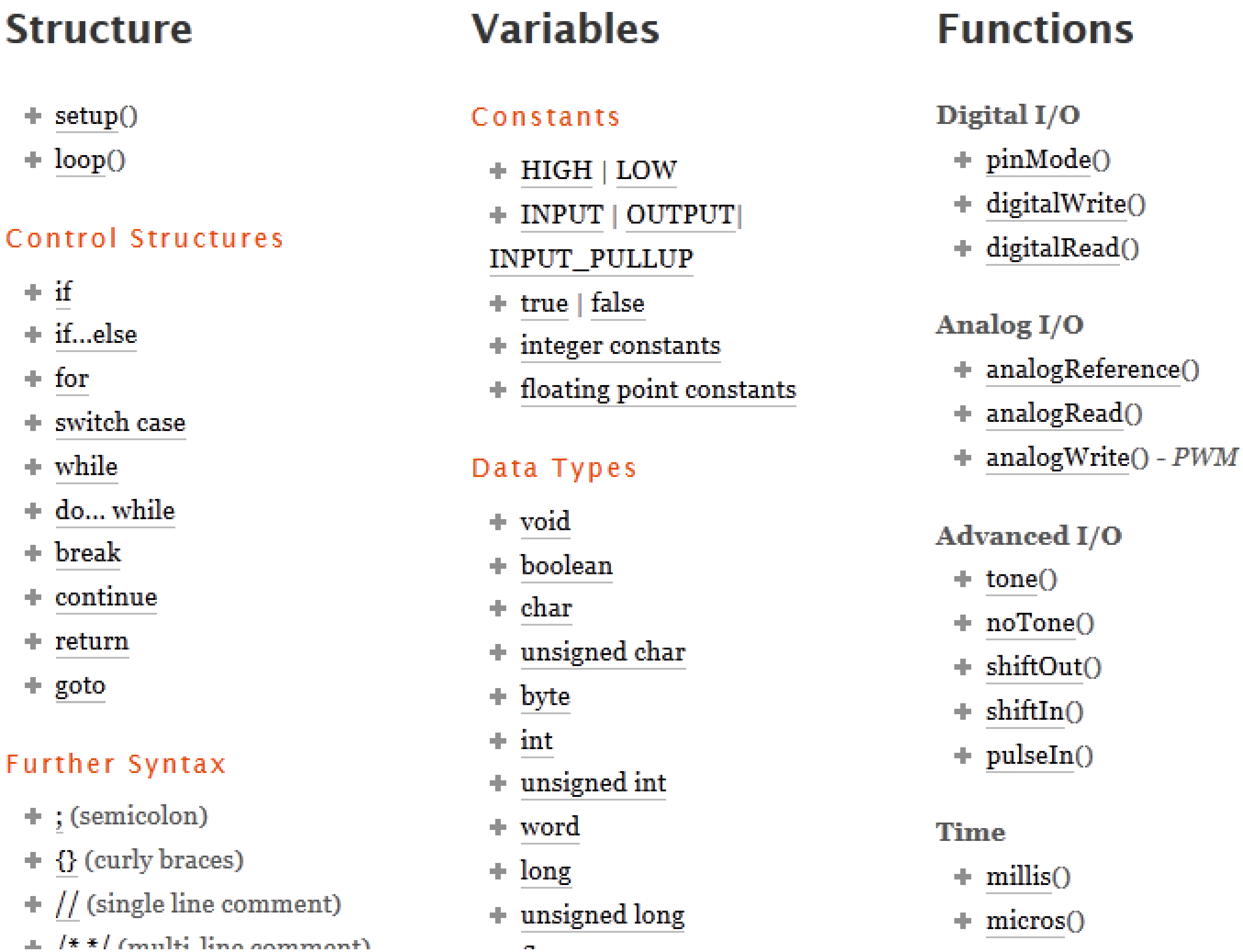
Intuitive programming IDE



```
/*
  Blink
  Turns on an LED on for one second, then off for one secc

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

1 Arduino Mega 2560 or Mega ADK on /dev/ttyACM0

Principles of Programming & Computing



Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /* */ (multi line comment)

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT| INPUT_PULLUP
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Advanced I/O

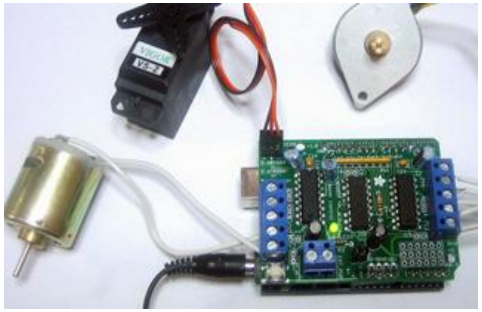
- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

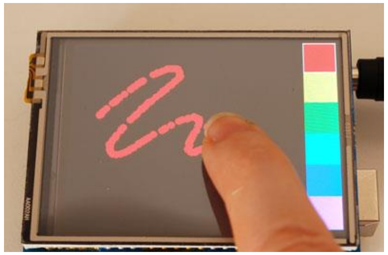
Time

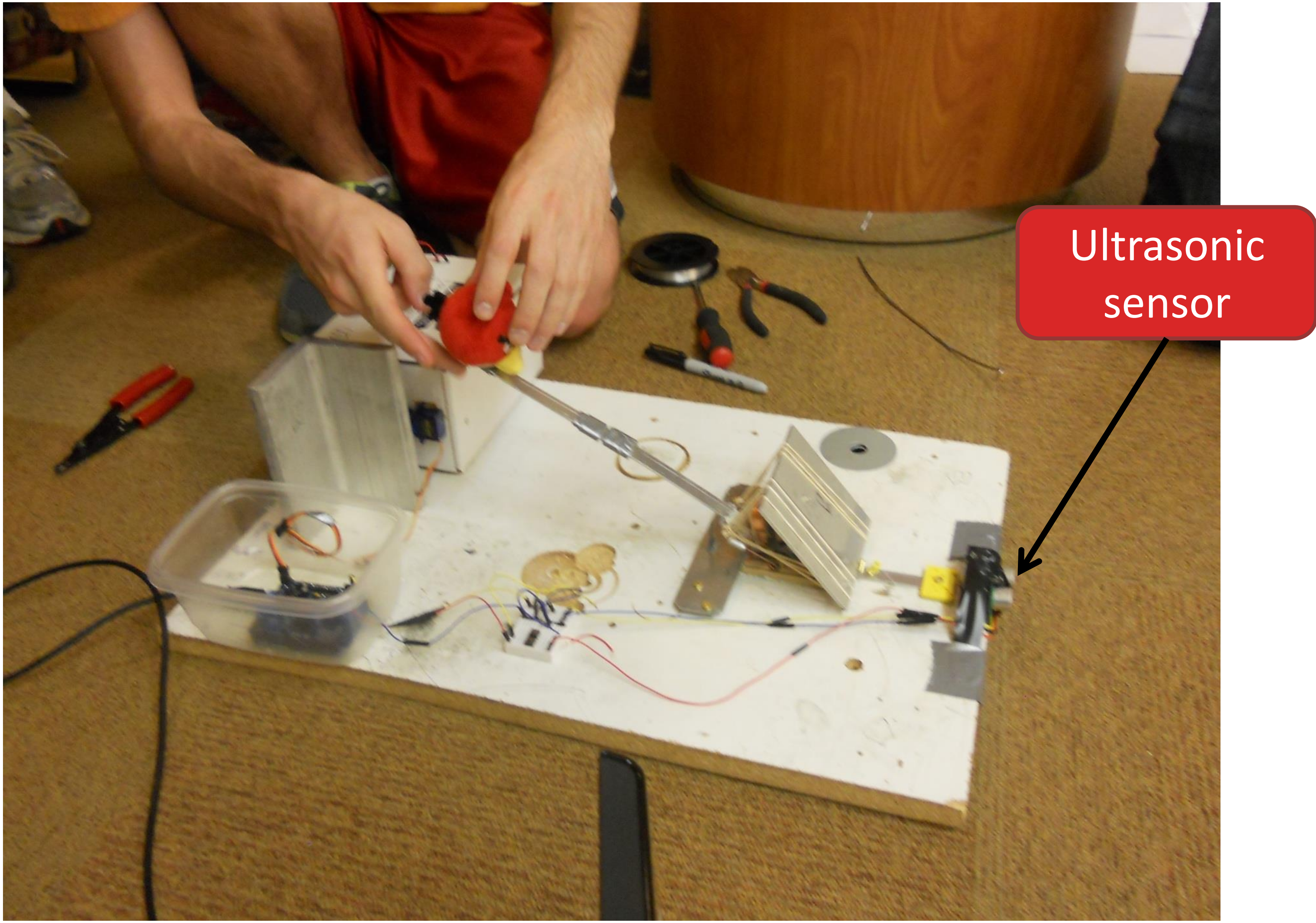
- millis()
- micros()

5 hands-on building projects to teach computer science

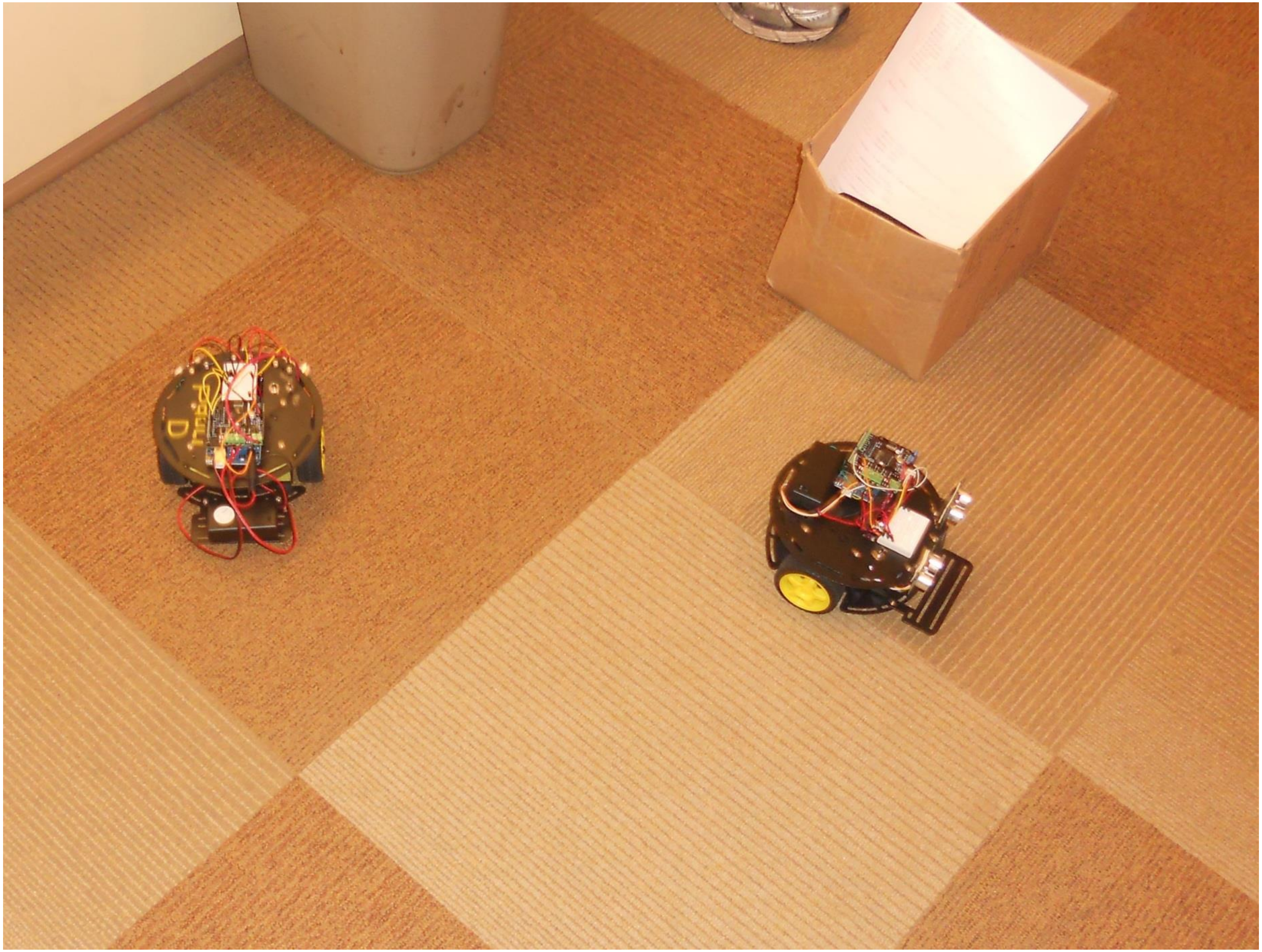
Freshman course: CS 252 Introduction to Computer Engineering

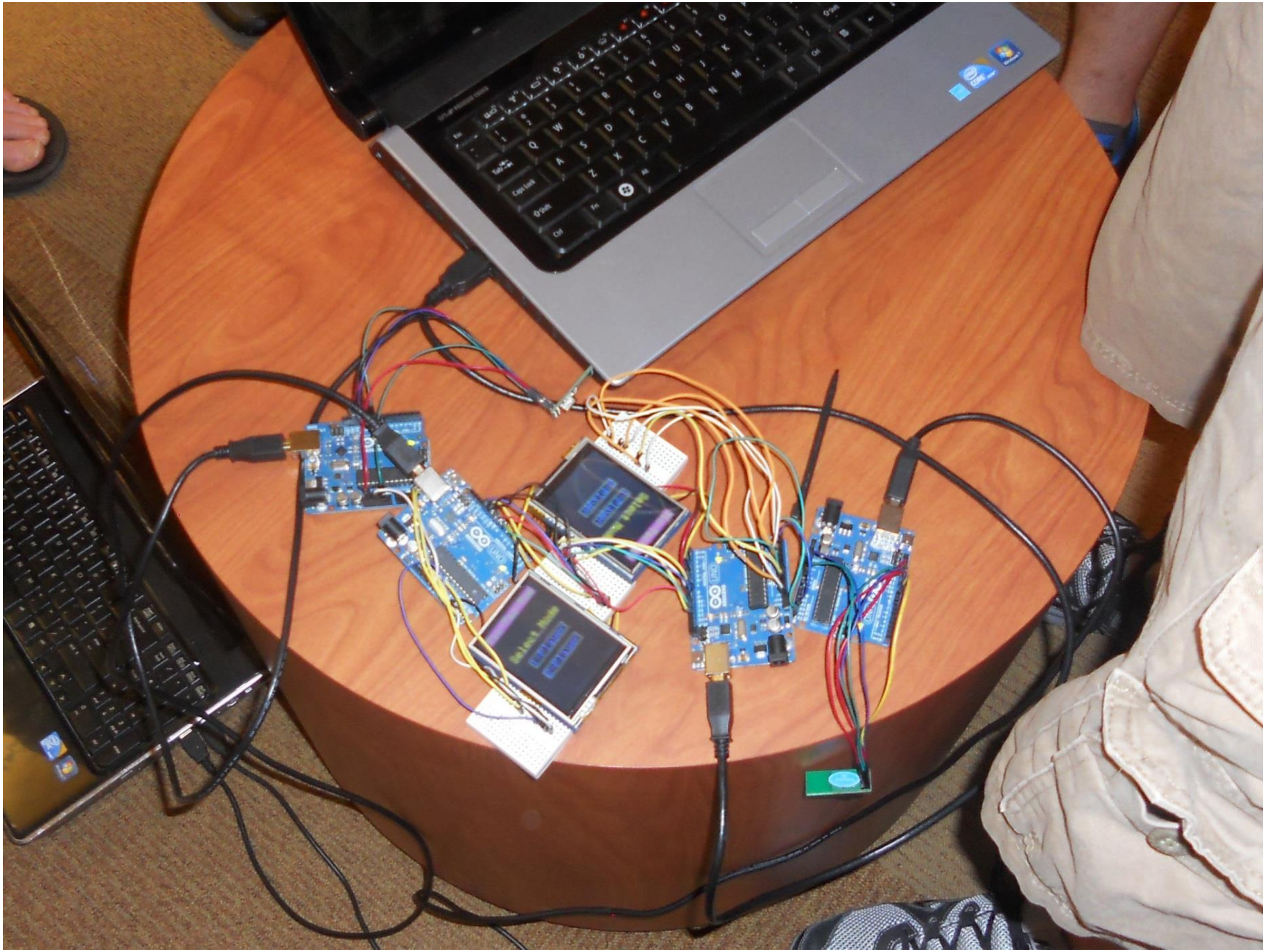


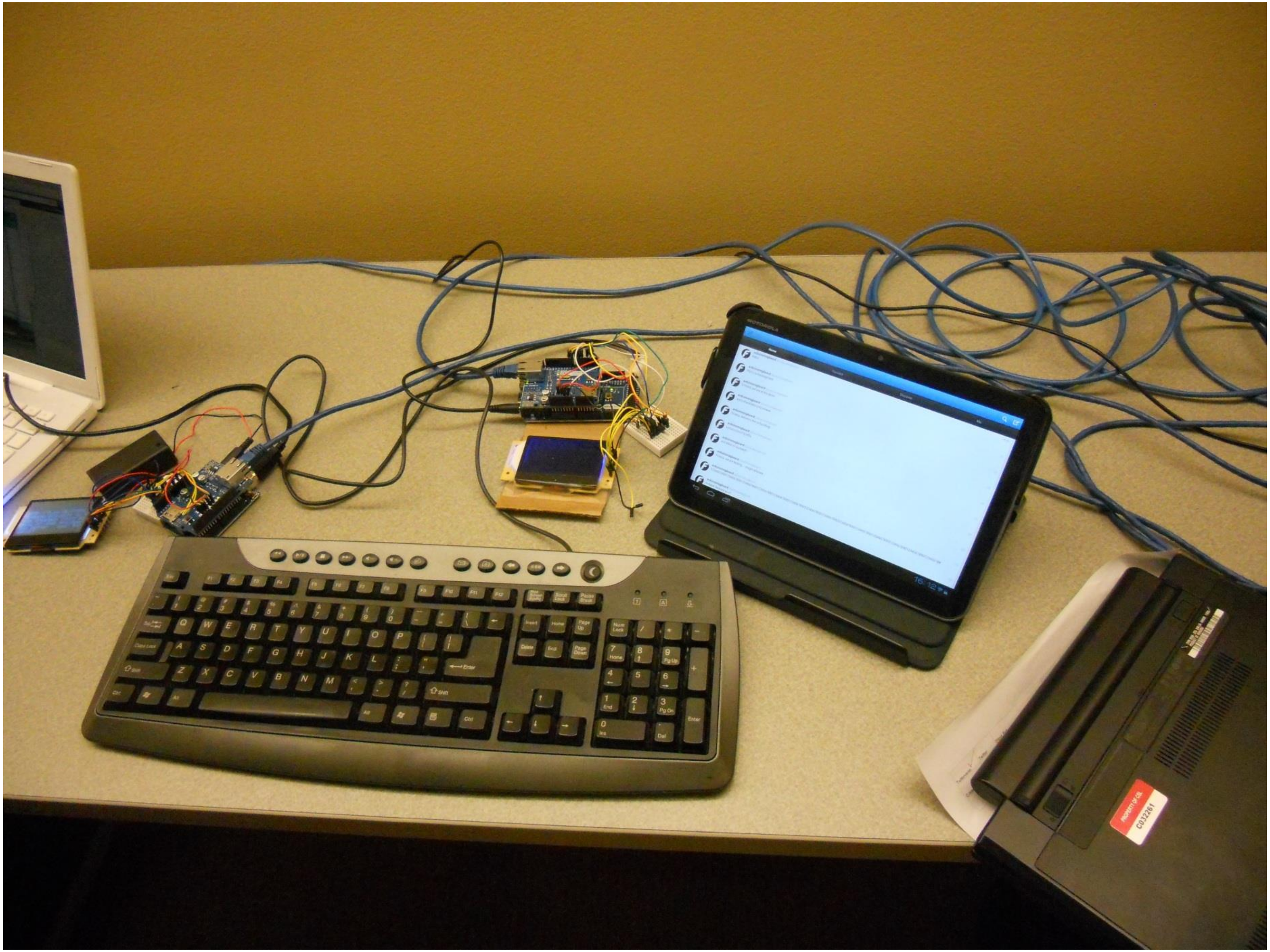




Ultrasonic sensor




Select Mode



Example Code

```
void loop()
{
  if (ultrasoundValue <=15 && ultrasoundValue >= 5
     && ultrasoundValueLeft > 10
     && ultrasoundValueRight > 10) { //spin clockwise;
      digitalWrite(E1,HIGH);
      analogWrite(M1,150);
      digitalWrite(E2,LOW);
      analogWrite(M2,150);
  } else if (ultrasoundValue <= 15 && ultrasoundValue >= 5
     && ultrasoundValueLeft <=10
     && ultrasoundValueRight > 10) { //spin clockwise
    digitalWrite(E1,HIGH);
    analogWrite(M1,150);
    digitalWrite(E2,LOW);
    analogWrite(M2,150);
  } else …
```

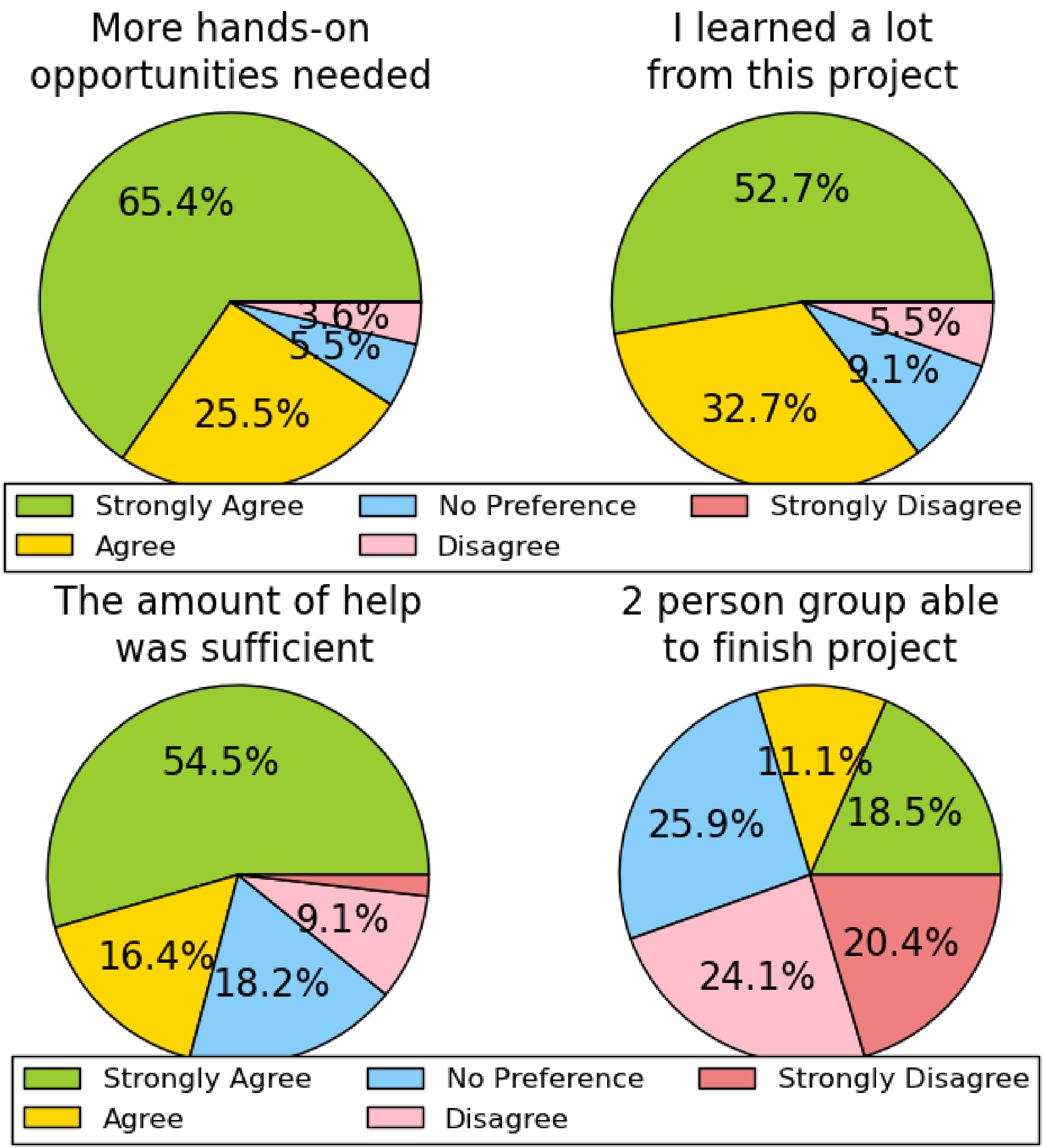
Learning Objectives

- Programming
  - Loops, conditionals, data-structures
- Systems
  - Notion of interrupts, concurrent programming, event-loop, device IO, wireless stack, **interference,** polling, **noise, overcoming noise,** Ethernet stack
- Algorithms
  - **Communication and hand-shake, maze traversal**
- Working with incompletely defined problems
- Working in a team, planning, asking for help
  - Proposal, revised proposal, 3 progress reports, final report

Instance 1 (Spring 2012)

- Extra credit – 5% of the course, Optional
- > 50% of the class participated
- 15 had no prior software experience
- Got them all hardware required
- Pointers to getting-started software
- **All but one team completed!**
- **2 teams went way beyond what we expected**

Instance 1 (Spring 2012)-Feedback


More hands-on opportunities needed

I learned a lot from this project

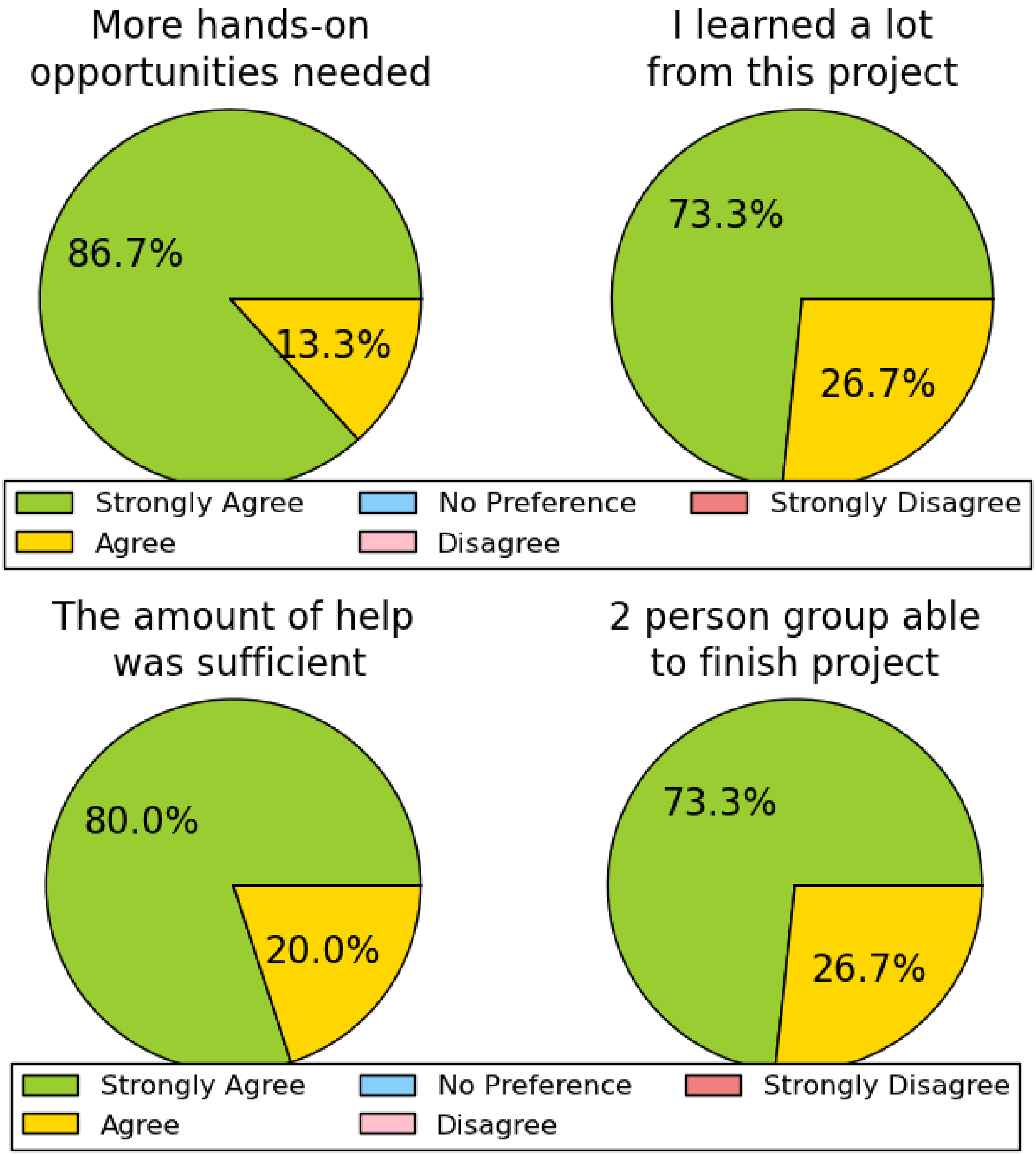
The amount of help was sufficient

2 person group able to finish project

Instance 2 (Spring 2013) - Improvements

- Instructional webpages
  - Detailed setup instructions
  - Demo videos
  - Step-by-step project plans
  - Intentionally open-ended!
- Support from multiple “Undergrad TA’s”
- Online platform for collaborative discussions

Instance 2 (Spring 2013) - Feedback



Student feedback

- My team put a lot of work into the project. If possible the **Arduino project could be used to form another credit for the class and in that case maybe the projects could be a little bit tougher.**
- This was **much more interesting than anything else** we did in class and I wish we could expand on it.
- **I thought it was great. It is a lot of fun,** and we are still making improvements on the robot.
- **Some step by step instructions** or more constructed demo.

Impact and Recognition

- SIGCSE paper
  - Premier publication venue for CS Education
- Matt Doran from instance 1 (undergrad freshman who created website for instance 2)
  - Astronaut Scholarship 1 of 40 offered nationally in all science disciplines
- Used in other offerings of 252
- Awards for me!
  - Emil H Steiger Distinguished Teaching award
  - Letters and Science Philip R. Certain - Gary Sandefur Distinguished Faculty Award in 2013

Lessons Learned

- Challenge : Diversity in student’s technical backgrounds
  - Projects of different complexity
- Challenge : Improving student enthusiasm and uptake
  - Instructional videos, open-ended projects
- Challenge : Too much information is bad!
  - Intentionally vague how-tos
- Challenge: Want more!

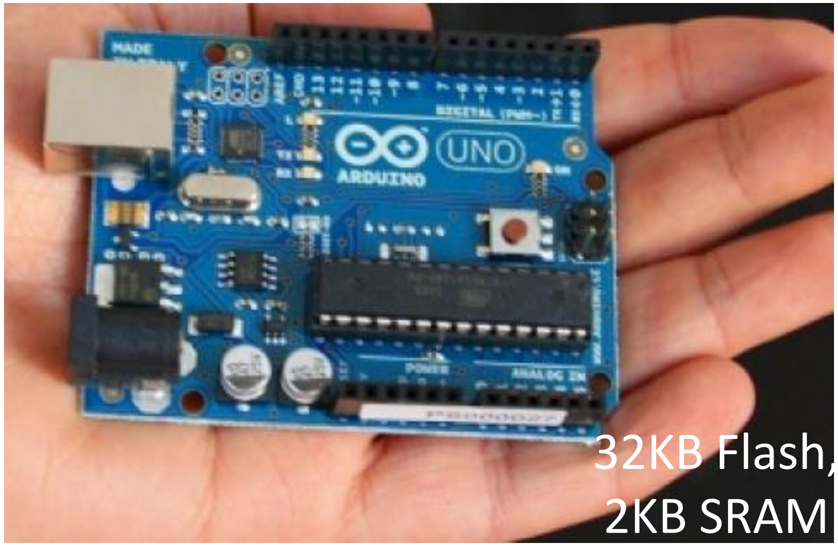
Can we extend and develop these hands-on projects through the entire curriculum?

A Hands-on Curriculum

- 202, 252, 352 : Overview of computing concepts
  - Arduino Lab with 2-person team projects
  - Core curriculum
- 536: Intro to Programming Languages and Compilers
  - Build compiler for Arduino’s language
- 537: Intro to Operating systems
  - Build Arduino OS and device drivers
- **552: Intro to Computer Architecture**
  - **Build Arduino processor, map to FPGA, drive shields**
  - **Run their Freshman project on their chip and software!**

Integration with Research

- Students gain exposure to research
  - Matt Sinclair (PhD at UIUC, Qualcomm Fellowship), Sam Wasmundt (PhD at UCSD)
- Realized Arduino processor is a great processor for data center!



32KB Flash, 2KB SRAM

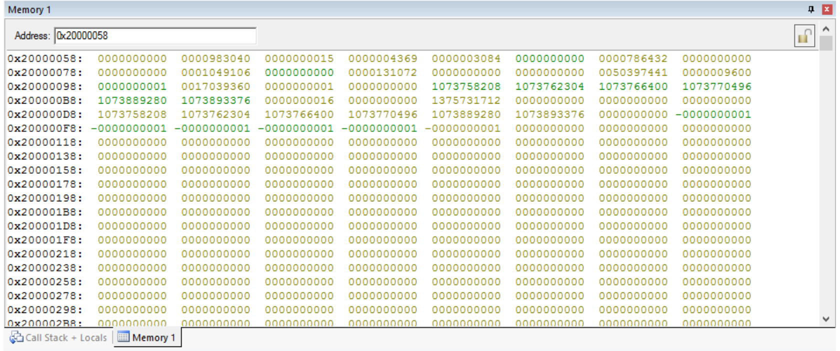
Outline

- Background & overview of CS Dept and me
- Soul of a new Freshman CS252 Machine
  - Teaching Computer Science by Building Computers
  - **Feedback and thoughts from Katie and Peter**
- Soul of a new Senior CS552 Machine
  - Building the very chip used in Freshman year: Theo
- Connections back to research
  - Powering future datacenters with these ideas! Memory Processing Units:Theo

Outline

- My experiences as a student
  - What I did, what I learned, impact on me
- My experiences as a TA
  - What I did, what I learned, impact on me

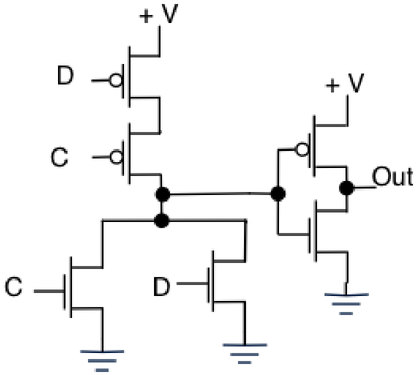
Background in ECE

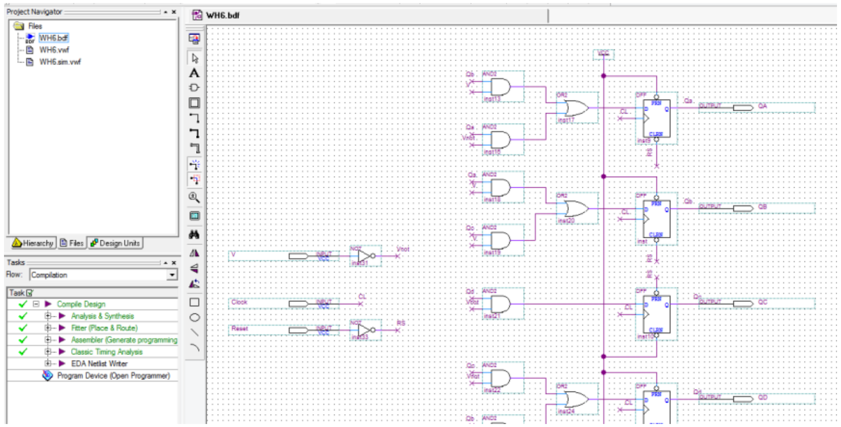


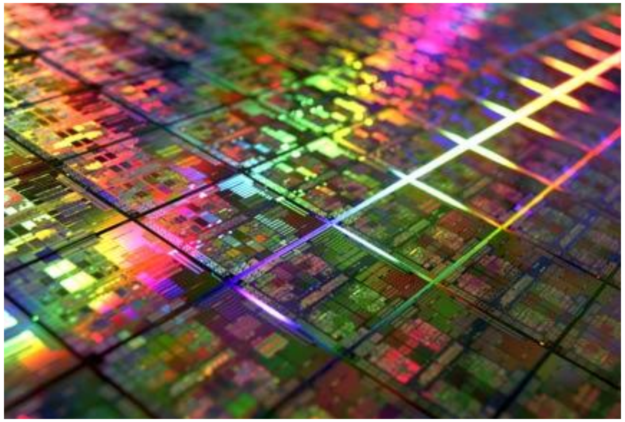
Memory 1

Address: 0x20000058

```
0x20000058:  0000000000  0000983040  0000000015  0000004369  0000003084  0000000000  0000786432  0000000000
0x20000078:  0000000000  0001049106  0000000000  0000131072  0000000000  0000000000  0050397441  0000009600
0x20000098:  0000000001  0017039360  0000000001  0000000000  1073758208  1073762304  1073766400  1073770496
0x200000B8:  1073889280  1073893376  0000000016  0000000000  1375731712  0000000000  0000000000  0000000000
0x200000D8:  1073758208  1073762304  1073766400  1073770496  1073889280  1073893376  0000000000 -0000000001
0x200000F8: -0000000001 -0000000001 -0000000001 -0000000001 -0000000001  0000000000  0000000000  0000000000
0x20000118:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000138:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000158:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000178:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000198:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x200001B8:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x200001D8:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x200001F8:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000218:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000238:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000258:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000278:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x20000298:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
0x200002B8:  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000  0000000000
```



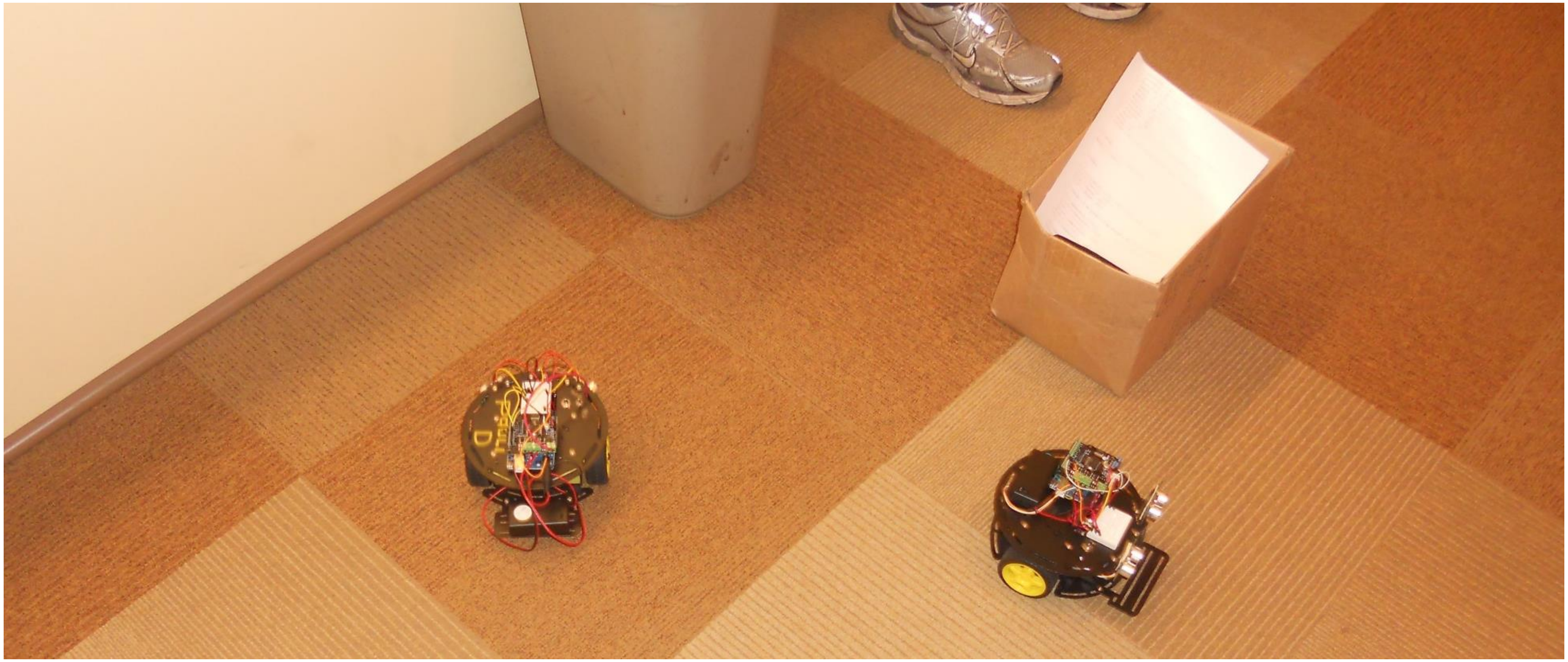




- ECE 252 first engineering course to introduce CMPE/CS concepts
- Had no prior experience in any ECE/CS topics

What I did

- Took opportunity for hands-on experience: joined project to create Arduino robots
- Created an ‘Obstacle Avoidance Robot’



What I learned

Make It So: Open Source, Arduino-Based Tricorder Nears Completion

- Introduced hardware/MCU programming
- Learned various hardware protocols

Impact on me

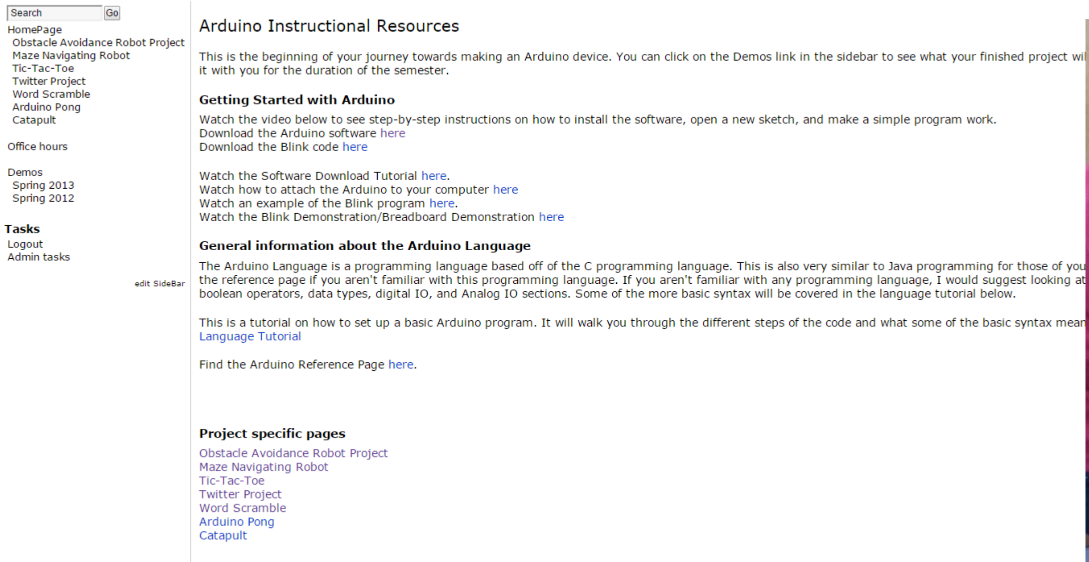
- Kick started interest in continuing work in the field of CMPE/CS
- Introduced me to branch of ECE that I am now most interested in: MCU & Internet of Things

Outline

- My experiences as a student
  - What I did, what I learned, impact on me
- **My experiences as a TA**
  - **What I did, what I learned, impact on me**

What I did

- 1st two weeks in Fall
  - Revise website
  - Assemble/disassemble projects
- During semester
  - Hold office hours
  - Trouble shoot
  - Email answers



Arduino Instructional Resources

This is the beginning of your journey towards making an Arduino device. You can click on the Demos link in the sidebar to see what your finished project wi
it with you for the duration of the semester.

Getting Started with Arduino

Watch the video below to see step-by-step instructions on how to install the software, open a new sketch, and make a simple program work.
Download the Arduino software here
Download the Blink code here

Watch the Software Download Tutorial here.
Watch how to attach the Arduino to your computer here
Watch an example of the Blink program here.
Watch the Blink Demonstration/Breadboard Demonstration here

General information about the Arduino Language

The Arduino Language is a programming language based off of the C programming language. This is also very similar to Java programming for those of you
the reference page if you aren't familiar with this programming language. If you aren't familiar with any programming language, I would suggest looking at
boolean operators, data types, digital IO, and Analog IO sections. Some of the more basic syntax will be covered in the language tutorial below.

This is a tutorial on how to set up a basic Arduino program. It will walk you through the different steps of the code and what some of the basic syntax mean
Language Tutorial

Find the Arduino Reference Page here.

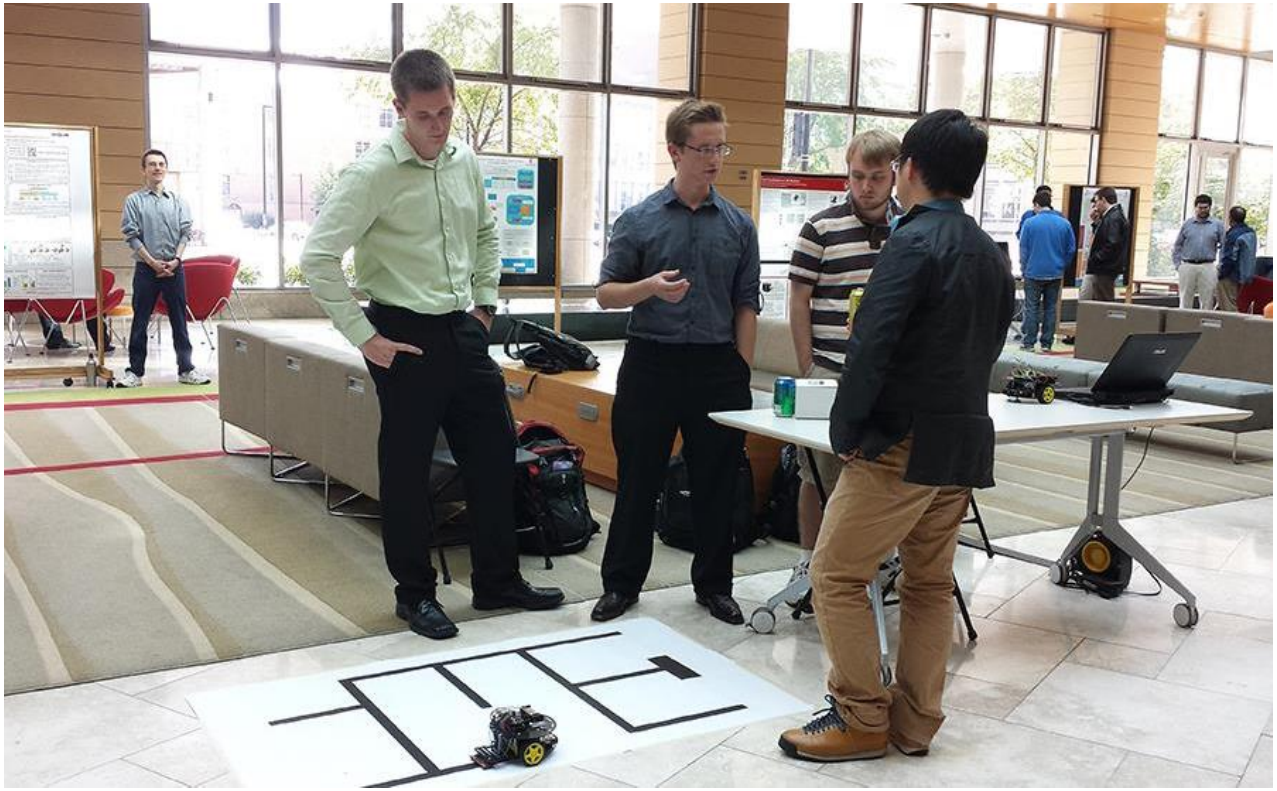
Project specific pages

Obstacle Avoidance Robot Project
Maze Navigating Robot
Tic-Tac-Toe
Twitter Project
Word Scramble
Arduino Pong
Catapult

What I learned?

- How to teach students hardware concepts on a basic but intuitive level
- How to effectively teach basic coding concepts
- How to gauge material/course based on feedback

Impact on me



- Teaching helped to reinforce CS/CMPE concepts
- Saw first hand student and faculty interest

Outline

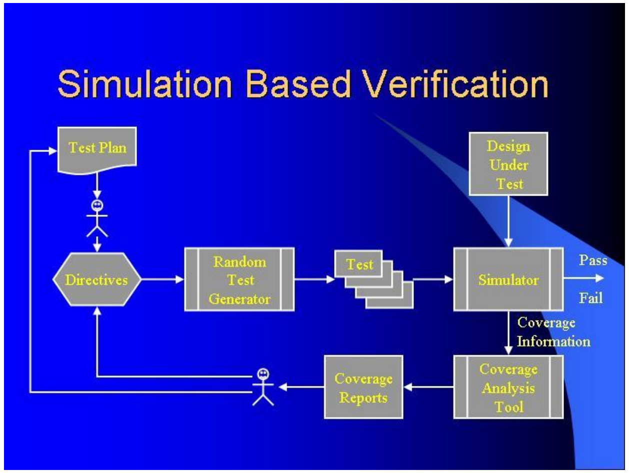
- Background & overview of CS Dept and me
- Soul of a new Freshman CS252 Machine
  - Teaching Computer Science by Building Computers
  - Feedback and thoughts from Katie and Peter
- **Soul of a new Senior CS552 Machine**
  - **Building the very chip used in Freshman year: Theo**
- **Connections back to research**
  - **Powering future datacenters with these ideas! Memory Processing Units:Theo**

Overview

- GROWL: extending Arduino projects into senior year
- What I learned
- Applied skills to MPU research

Background: Senior course

- "Build" a microprocessor
  - Very cool, but… "build" defined as run random programs, look at waveforms, verify correctness.



Simulation Based Verification

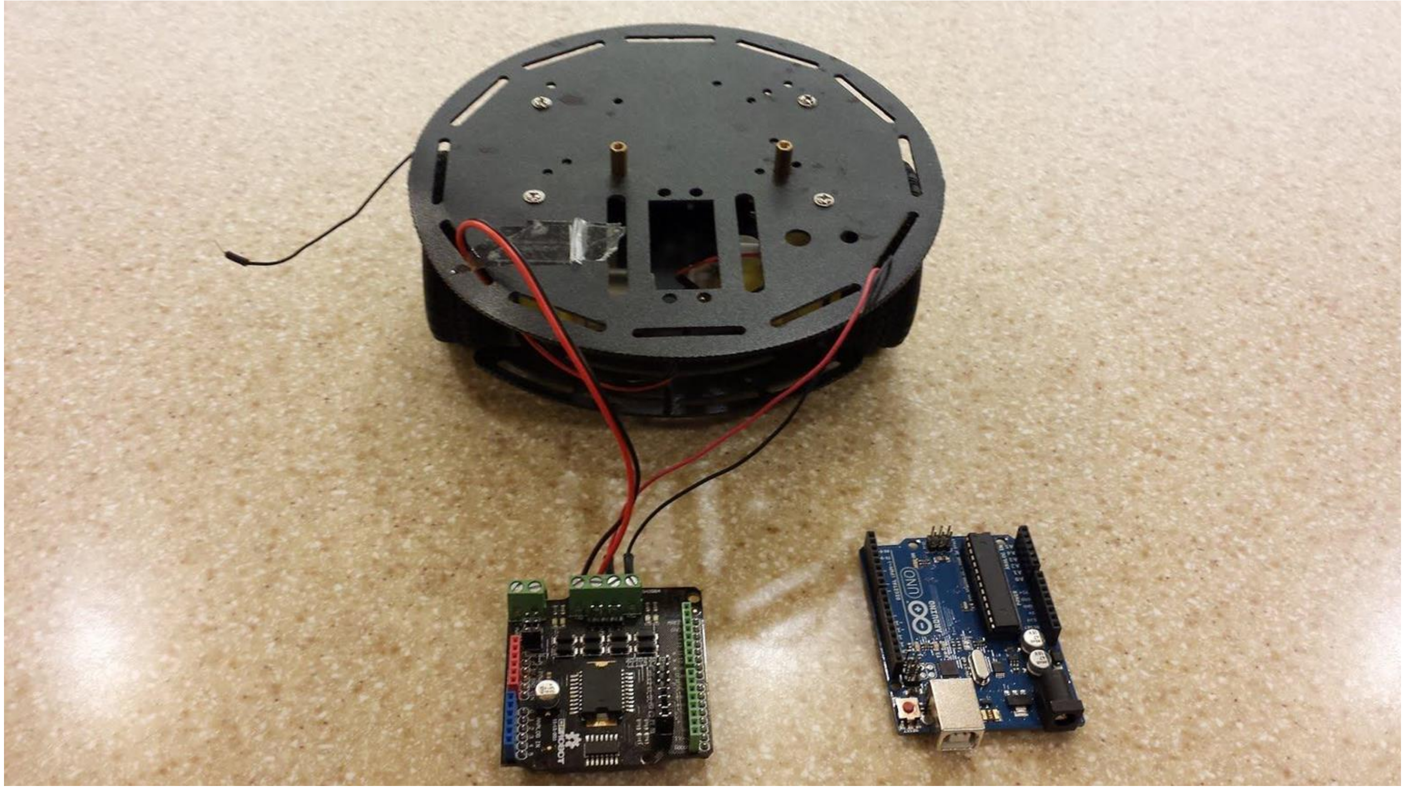
Mentor Graphics®

cadence®

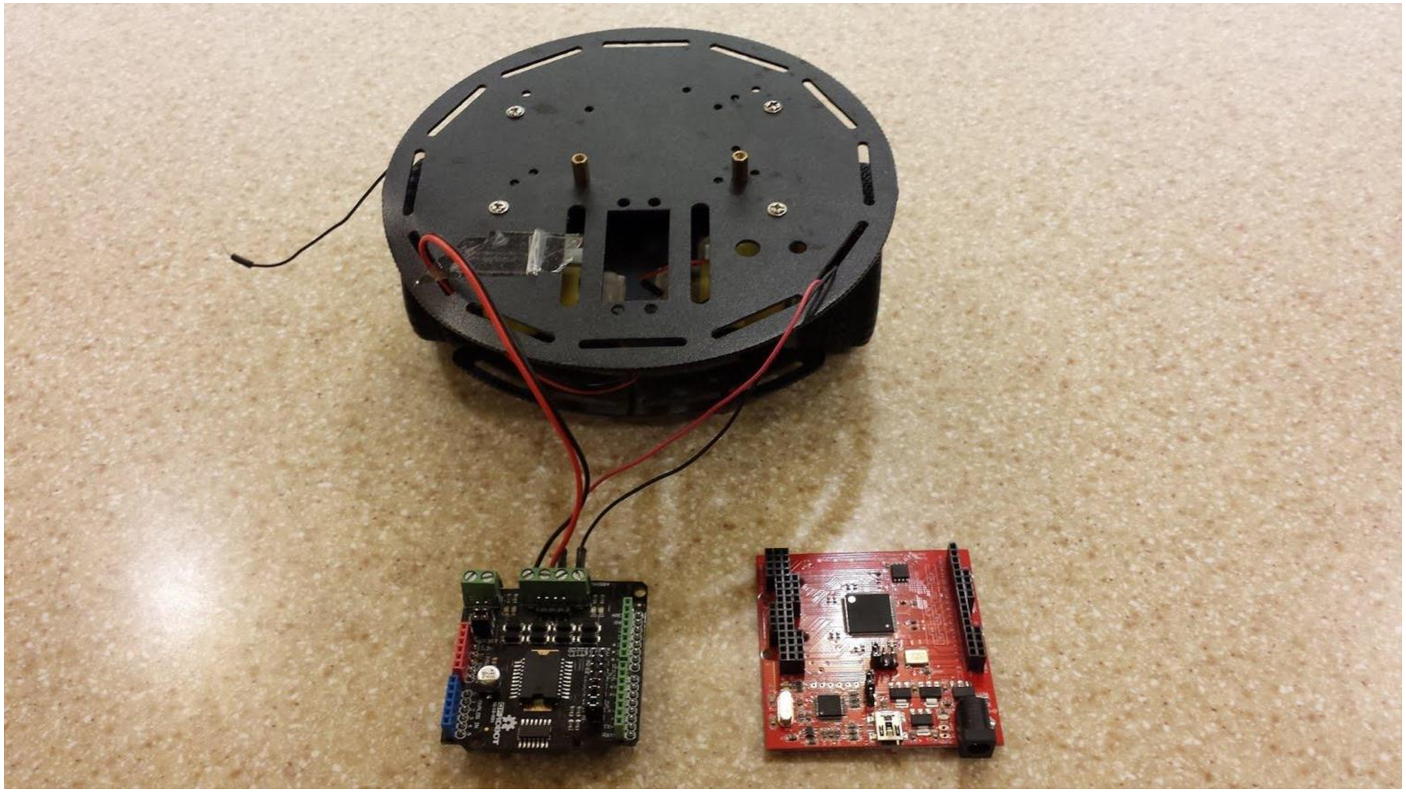
synopsys®

- **Nothing you can hold in your hands**
- **Can we build a real chip?**

$75 Gets You



$125 Gets You



Meet GROWL

How did I get there?

- ISA manual, FPGA board, design tools
- Six months of LOTs of work
- Internet

What we have done

- Designed and implemented Arduino processor
- Built testing infrastructure
- Verified implementation
- Loaded on FPGA

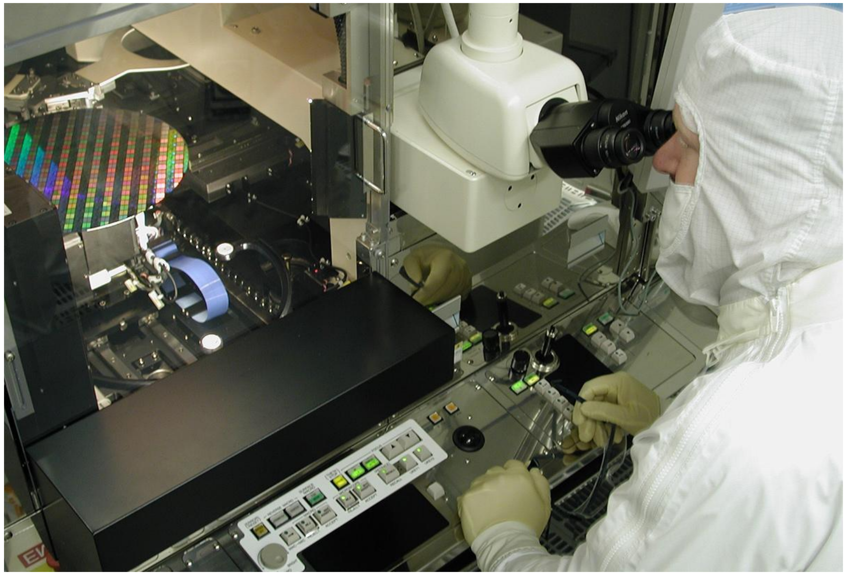
Design and Implementation

Verification Infrastructure

- Figure out how to test the processor
- What to test it with?
- How do I know if it is working as expected?

Verification

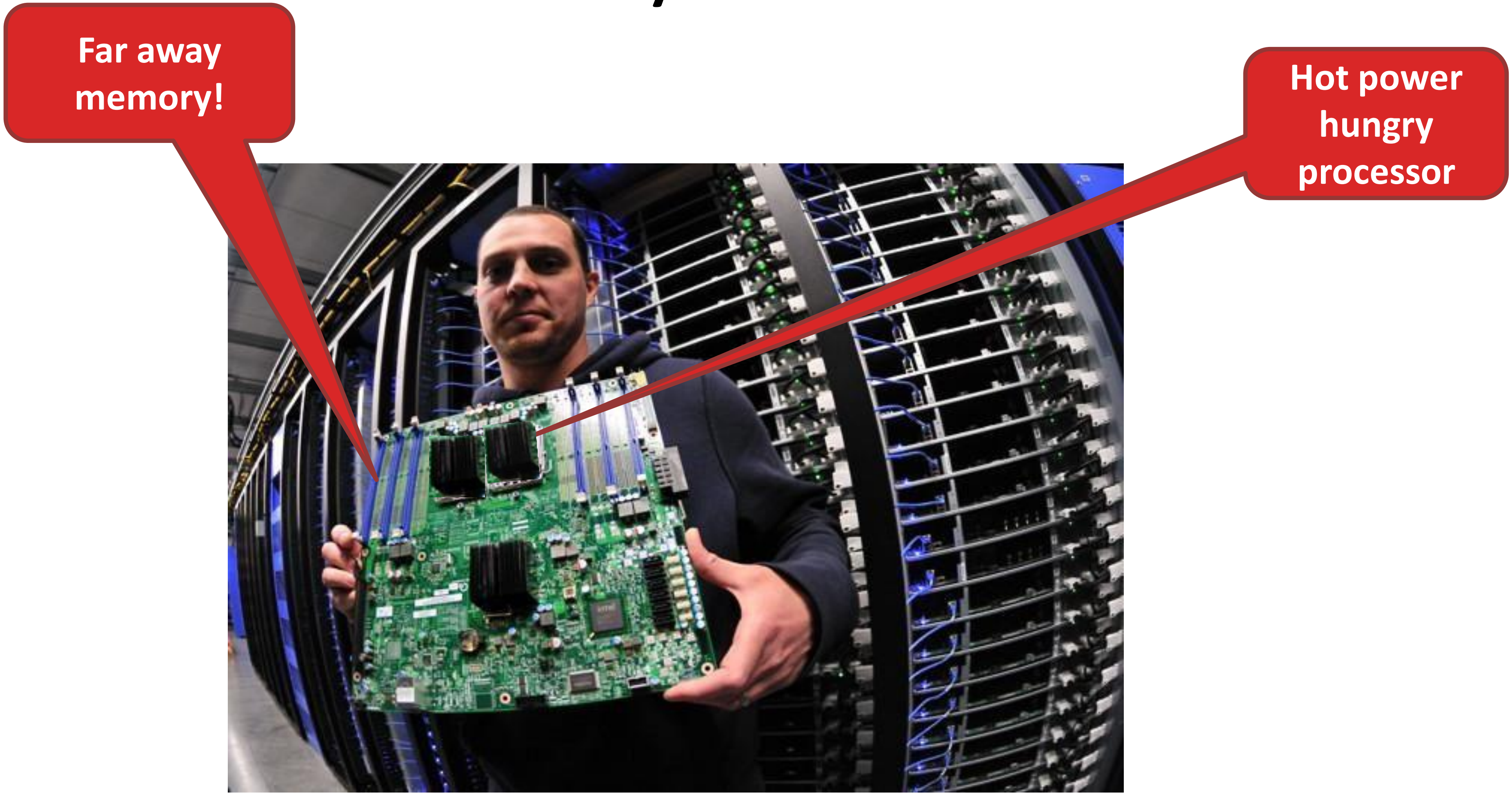
Map on FPGA: $20 million vs $125



What I learned

- Deeper understanding of how hardware works
- Technical skillset to work on hardware design problems
  - Tools
  - Languages
- Insight: this simple processor can play a role in big servers

Today's servers

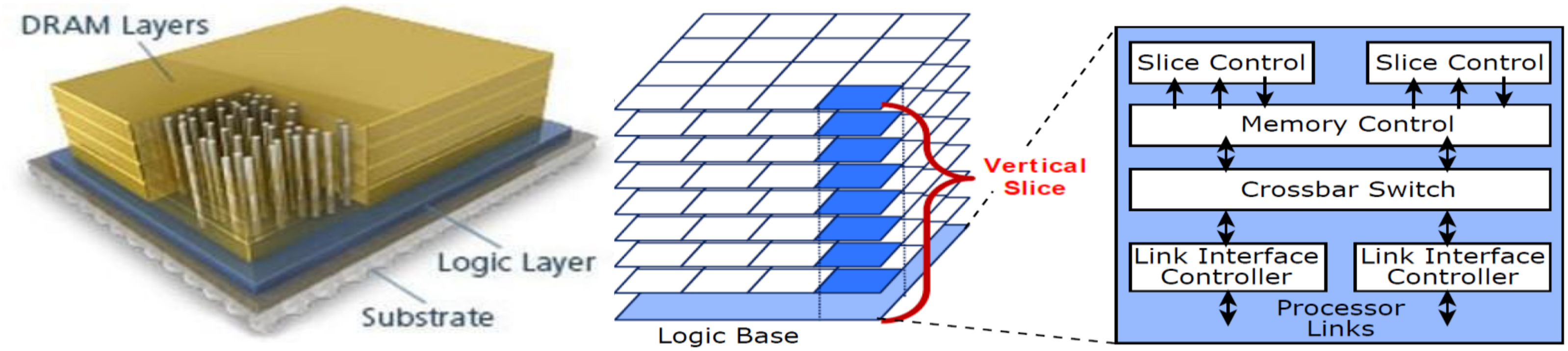


5pJ to do an operation but 360pJ to access memory

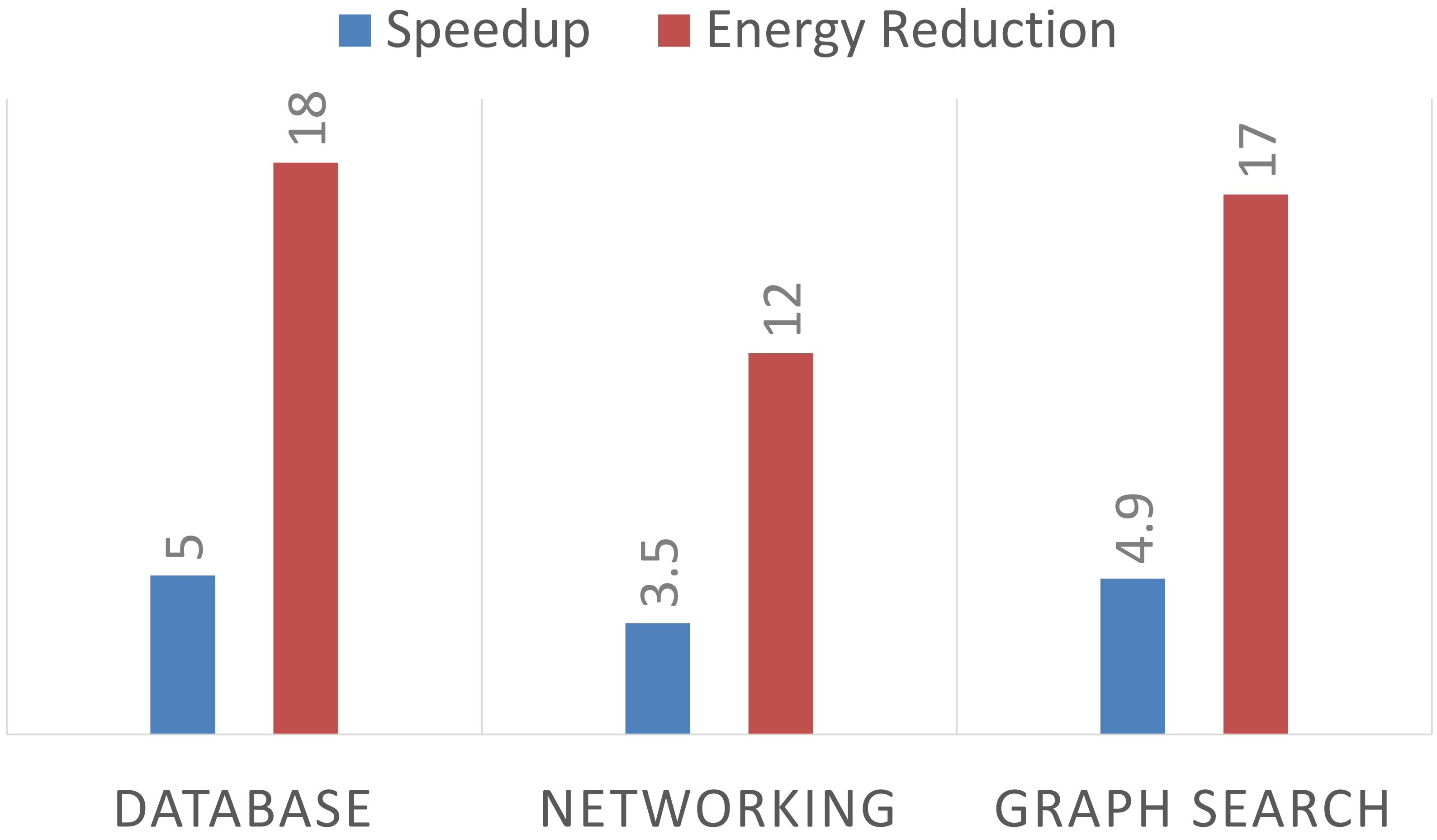
Runs for 1ns and waits for 40ns for memory

MPU Idea

- Processing in Memory with new 3D stacked memory
- Simple Arduino cores running at 250 MHz
- 5X to 10X faster and lower energy



MPU Results Relative to Today’s Server

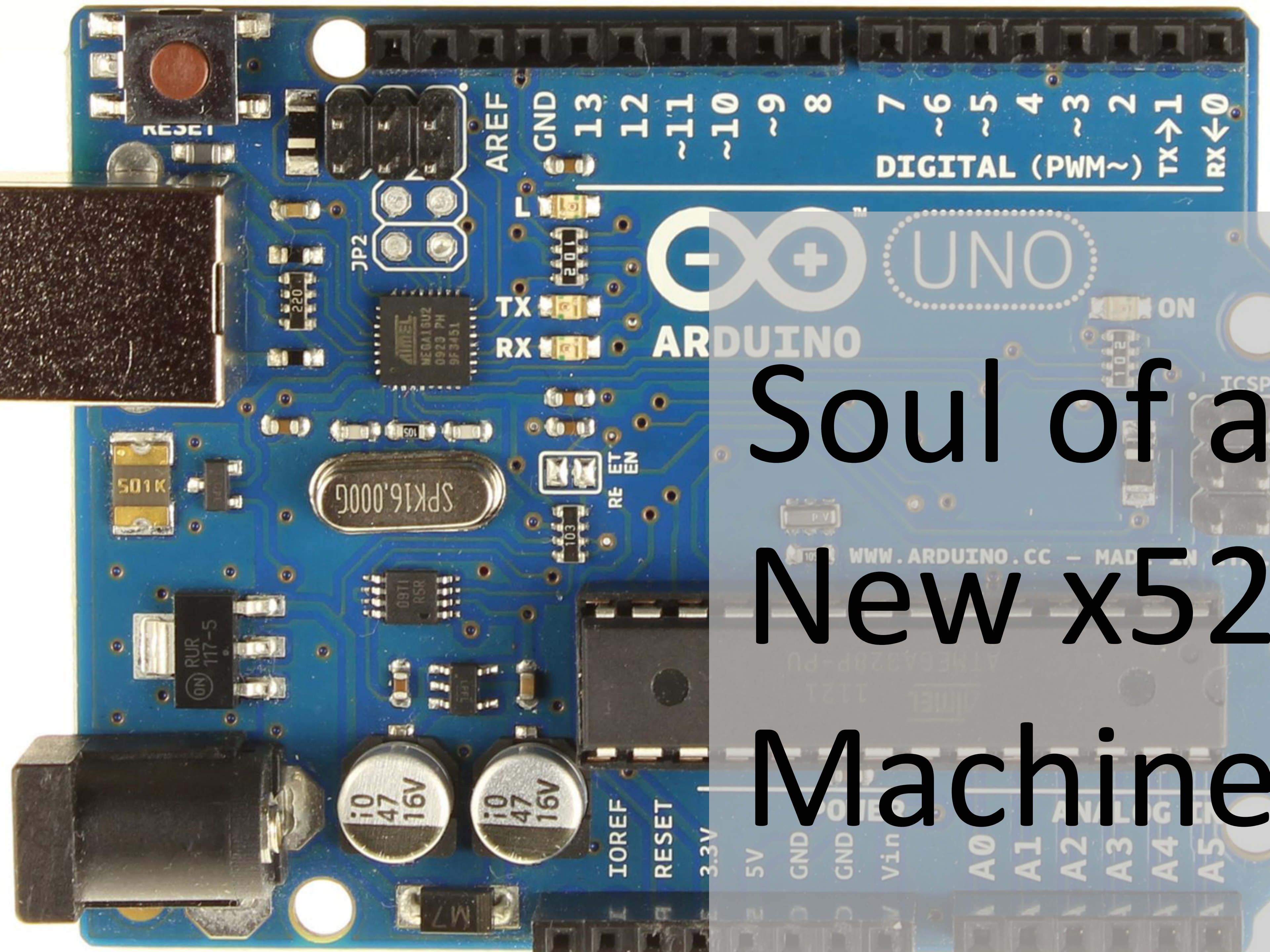


MPU Contributions

- Co-developed refinements to architecture
- Leading workload analysis
- Developing simulation infrastructure
- Prototype physical chip design using FPGA

Summary

- GROWL is ready for a trial with students
- Skills I learned from GROWL prepared me for MPU research



Soul of a New x52 Machine

Concluding Thoughts

- CS enrollments soaring
- These and other innovations are part of department's vision to:
  - Make CS major more accessible
  - Teach broadly applicable courses, CS certificate
  - Research expanding into exogenic (externally motivated) area