Todo

- Order starting materials from Amazon; give reimbursements to Matt
- ~~Update plan based on Brandon's feedback~~
- ~~Create detailed hour-by-hour workshop schedule~~
- ~~Create detailed materials list~~

Before leaving...

- Try out labs at MIT
- Create handouts when necessary (circuit diagrams/reference guides)
- Create template code; post on GitHub
    - https://github.com/skoppula/kis-workshop-code-2014

Summary

Learn fundamentals about software and hardware design and engineer robotic systems

Learning outcomes

- Have students think from a design, creative perspective.
- Applying concepts learned on paper to use in the real world

- Learn select topics in CS theory
    - Search algorithms
    - Basic programming (functions, variables, etc.)
- Learn to write code that implement algorithms
- Learn basic EE hardware/mechanical engineering topics
    - Circuits and circuit components
    - Torque, force, power, energy relations (time permitting)

Day One: Intro to Programming and Circuits

09:00-10:00am: Both

Introduce ourselves and where we're from, ice-breaker to learn people's names, grade, interests, past experience; introduce EECS and why it's cool, assign groups, and overview of entire week: this is hands-on class with competitions at end!

Demo our prototype robot here so that they get a picture of what they will be able to do by the end of the week.

10:00-11:00am: Juan

Get students on laptops. Explain how to write variables and do basic math operations in C. Write code to solve basic math operations. Introduce control flow. Write code to count to 100.

11:00-12:00am: Juan

More programming exercises. Check if a number is odd. Check if a number is prime. Introduce recursion. Output Fibonacci numbers.

We would teach modulo arithmetic, recursion, and functions in one hour, time permitting.

1:00-2:00pm: Skanda

Explain what a circuit is (directed flow of charge) (Would be cool to make a worksheet). Introduce wires, voltage source, resistors and LEDs. Demo how to solder. Hook-up an LED to wires (make sure LEDs don't burst!) and solder Arduino battery connectors. (It's just hooking up batteries to resistors and LEDs)

2:00-3:00pm: Skanda

Introduce $V=IR$. Introduce breadboards. Make LED and resistor light up on breadboard. Talk about two types of circuits: parallel and series. Make a series and parallel circuit. Relate series/parallel circuits to the brightness of an LED without introducing power.

Day Two: Microcontrollers and Making Things Move

09:00-10:00am Juan

Demo the line follower!

Explain what a servo, analog sensor, and Arduino are. Have them put Arduino onto chassis, run pre-written program, and watch motor move and lights turn on. Have them change values in program (guided by code comments/worksheet) and understand the program line by line. Play with LEDs and motor.

10:00-12:00am Juan (potentiometer) and Skanda (preset maze stuff)

Half the class: write a program to take in potentiometer values and move a motor and change lights. Test and debug. *Other half of class*: Finish a maze with preset movement distances (make multiple mazes, spread apart). Switch projects when done.

1:00-1:30pm: ***Line Following*** Skanda

Discuss with students what needs to be on our robot for it to follow a line. What would be the logic (algorithm)? Create an algorithm flowchart.

1:30-3:00pm Skanda

Attach the light sensors and read their values. Write line-following Arduino code, explain to Juan or Skanda how it works (hints on GitHub pre-written code). Run, test, and debug line follower.

Day Three: Remote Control a Lazy Arduino

09:00-9:30am Juan
Explain different types of E/M communication and uses in day-to-day life. Outline communication between Arduino and a remote.

09:30-10:30am Juan
Wire together IR sensor and 4 LEDs to Arduino. Explain basic functions in Arduino to access IR input.

10:30-12:00am Juan
Start and finish Arduino program to control 4 LEDs using remote control (template provided, if they want). Run, test, debug. Use any remaining time to finish projects from previous today.

1:00-2:00pm Skanda
Make maze for robot to traverse. Begin outlining code for traversing maze via IR sensors. Talk about code encapsulation/modularization.

2:00-3:00pm Skanda
Finish writing code to have the Arduino traverse a maze using remote control. Run, test, debug. Competition - who can get through maze the fastest!

Day Four: Autonomous Arduinos

09:00-10:30am Skanda
Demo a maze solver! Discuss how we could make an Arduino go through the maze by itself. What kind of sensors could we use and how many? Assume we use three ultrasound sensors. Have students in their group come up with a sequence of steps/algorithm flowchart to make the robot solve a maze.

10:30-12:00am Juan
Have students wire their ultrasonic sensors and read in input. Code their solution. If finished early, work with other groups to help them figure out the algorithm and code solution.

12:00-1:00pm
Run their code and test in maze. Discuss some of the problems. Does the robot complete the maze? How could we make it faster?

1:00-2:00pm
Implement points raised in discussion and get robot through the maze.

2:00-3:00pm
Competition in gym! Who can finish the maze fastest?

Day Five: Miscellaneous Topics

09:00-10:00am
Wrap up loose ends from previous day(s).

10:30-12:00am Skanda
Discuss what AI and machine learning are and why we need smarter computer thinking. Brief introduction to computer vision. Explain what the problem is and how we solve it. Have students try out a Python demo program on the computer. Show cool machine learning videos.

12:00-1:00pm
Programming contest: which team can code up solutions to a set of tricky problems?
Contest in computer lab. Awesome souvenir/candy prizes.

1:00-2:30pm
Robotics contest: who can design a robot and finish a maze with curved edges in the fastest time?
Contest in gym. Awesome souvenir/candy prizes.

2:30-3:00pm
De-brief and chill. Comments? Questions? Ask J&S anything.