

The Arduino: Computational Invention Made Easy

Todd W. Neller

What is an Arduino?

- A small, open-source, affordable, easy-to-program microcontroller (much like a CPU)
- 16K RAM
- Up to 20Mhz clock
- Inexpensive:
 - Arduino Diecimila board (\$35 @ SparkFun.com)
 - Replacement Atmel ATMEGA168 chip <\$5
 - Parts to build your own ~\$10

What is it used for?

- “Making Things Talk” – getting devices, sensors, actuators to communicate
- Adding a computational brain to a circuit for:
 - Control, i.e. directing circuit operations
 - Reasoning, “intelligent” decision making, etc.
- “Embodying” your code – making the transition from input text and GUI events to button presses, knob readings, etc.

Example: Arduino Drum Kit

- Part of Tod E. Kurt's Arduino tutorial at <http://todbot.com/blog/2006/10/29/spooky-arduino-projects-4-and-musical-arduino/>
- Piezoelectric sensors used to detect drum hits
- Arduino used to interpret sensor data and send as MIDI events

Example: Read Nunchuk Data

- <http://www.windmeadow.com/node/42>
- Nunchuk accelerometers reasonably affordable and easy to interface with
- What might you do with Wii nunchuk data?
 - Remote control
 - Gesture recognition → command
 - ???

Example: Arduino Plant Care

- Botanicalls Twitter:
<http://www.botanicalls.com/twitter/>
- Soil moisture sensor → circuit → Arduino which twitters plant status
- For each of these and many other 100s of Arduino projects documented online, how might you take them *one step further*?

How can I program it?

- Arduino programming language based on simple Wiring language (based on C/C++)
- Note the simplicity:
<http://www.arduino.cc/en/Reference/HomePage>
- CS 111 students can expect to understand most language elements
- Programming environment is *free*

Don't I need to be an Electrical Engineer?

- Let's turn this question around: Does a person need to be a Computer Scientist to do simple programming?
- An understanding of programming language basics opens the door to many useful applications.
- For simple circuit applications, Computer Scientists need an understanding of *basic building blocks of electronics*.

Building Blocks: LED

- From terminal window start Arduino software (“arduino &”)
- Plug Arduino into USB port
- File → Sketchbook → Examples → Digital → Blink
- Upload to I/O board (Control-U)
- Note small blinking LED on Arduino
- Disconnect Arduino, follow breadboard instruction with 5V, Ground, and LED, reconnect
- BlinkVariable – Blink with variable delay

Building Blocks: Pushbutton Switches

- Follow breadboard instructions for pushbutton switch
- File → Sketchbook → Examples → Digital → Button
- Upload to I/O board (Control-U)
- Change sketch to toggle LED on/off
- Try BlinkEcho

Example: The Four-Bit Maze

- So what can one make with an Arduino, LEDs, and buttons?
- <http://cs.gettysburg.edu/~tneller/mazes/oskar4bit/arduino.html>
- Goal: Turn off all four LEDs
- One button for each LED
 - Button by “off” LED has no effect
 - Button by “on” LED turns that LED off *and possibly changes others*
- Maze in the sense of navigating through LED states in search of goal state

Arduino Design Competition

- *If you could create any electronics project, what would it be?*

1. Look at what's out there already
2. Consider building on an existing project or taking a simple first step
3. Discuss with and get approval from a CS faculty member
4. Borrow an Arduino kit; we have/can get other components as well
5. Present your work at our colloquium before spring break
6. Presenters will compete for prizes (TBD) and all will get to keep their Arduinos

Extend your CS skills beyond the computer!