Arduino Workshop – Robotics

LEARNING GOALS: After the completion of this workshop, students will understand:

1. How the devices that kids use every day include programmable microcontrollers
2. How do you customize microcontrollers (hardware) with your own programs
3. How to use various electronic components and how to use the Arduino controller to control or communicate with those components
4. How to build and program an autonomous robot using a microcontroller

CONCEIVE – What do I wish to accomplish through this project?

This stage involves guiding students in defining the goals of the project, then helping them develop conceptual, technical and action plans to meet those goals while considering the technology, knowledge, and skills that apply. This guidance is provided in the form of Essential Questions that use student's preconceptions, and misperceptions then move them toward a deeper and more realistic understanding of the process and skills needed to complete the project.

ESSENTIAL QUESTIONS:

1. Introduce students to electronics and how to program the microcontroller to produce a designed result
2. The components that make up everyday devices.
3. Hardware design (hardware hacks) vs software design (software hacks)

DESIGN - How will I accomplish the project?

This stage focuses on creating the plans, drawings and algorithms that describe the product, process or system that will be implemented.

What is Arduino and how is being used. Link to an interesting and artistic project http://youtu.be/Eze8lt7jDnY - Twitter
World Mood Lamp
This workshop lends itself well to open inquiry with differentiated levels of engineering challenges. All students start with a simple Blink program then harder challenges are given to students to accomplish throughout the workshop.

NOTES: Project ideas can be found in many reference books such as: http://www.amazon.com/Beginning-Arduino-Michael-McRoberts/dp/1430232404/ref=sr_1_5?s=books&ie=UTF8&qid=1338837208&sr=1-5

IMPLEMENT - From an idea to a product!
This stage refers to the transformation of the design into a product. It includes hardware, manufacturing, software coding, testing and validation.

**DAY 1 – 2 hours**

**Depending on the number of students it is best to have students work in groups of 2**

1. After describing and discussing the Arduino microcontroller and its varied uses we discussed the programming and how to compile and download a program to control certain inputs and outputs of the microcontroller. A diagram of the board can be found on the attached power point slide.
2. The first build is the Blink sketch. This can be found on the Arduino software → File → Examples → Basics → Blink
   a. The Blink sketch will open, press the upload button (arrow). The pin 13 LED should begin blinking
   b. Go through the sketch comments (included in the sketch) as an overview for what the commands mean
3. Place an LED with the long leg (positive) in pin 13 and the short leg (negative) on one of the gnd pins
   a. Students will observe the LED flashing at the same rate as the onboard LED
   b. **CHALLENGE #1**: challenge the students to change the rate of blink. Don’t show them how to do it, let them figure it out. Some will do a rapid flashing which will make the LED appear solid. Find out where students can start to see a blink.
   a. **CHALLENGE #2**: challenge students to build a circuit that powers the LED from pin 13 to the breadboard and still blinks (hardware hack) [http://www.arduino.cc/en/Tutorial/BlinkingLED](http://www.arduino.cc/en/Tutorial/BlinkingLED)
   b. **CHALLENGE #3**: challenge students to power more than 1 LED (hardware hack)
   c. **CHALLENGE #4**: challenge students to now power more than 1 LED but make them blink at different rates (software hack)
   d. **CHALLENGE #5**: challenge students to make a “Knight Rider” style chase sequence (software hack)
5. Introduce the analogWrite command using the potentiometer build (#2 in Inventor’s Kit)
6. Depending on the amount of time available and to in order to make the workshop more open inquiry (rather than guided inquiry) have the students do some research on different uses of the Arduino microcontroller online and see if they can design their own circuit and program or re-create and modify and project from an online source.

**DAY 1 – 1.5 hours**

1. Build chassis using Magician’s Chassis instructions (attached)
2. Begin breadboarding the motor shield using the Fritzing diagrams (attached)

**DAY 2 – 2 hours**

1. Finish breadboarding the motor shield and test motors
2. Coding – introduce motor pin coding and motor control (directions and power)
3. Complete construction – fasten breadboard and Arduino on chassis
4. Students hack code to make the motors run for set amounts of time then turn

**DAY 2 – 2 hours**
1. Ultrasonic sensor – serial read, and print commands
2. Coding for the sensor
3. Testing autonomous robot

**DAY 2 – afternoon**
1. Robot challenges and navigation contest (all groups)

- Set-up computer with projector for powerpoint
- Provide above parts. If you are short on time you can pre-package the necessary parts
- Make sure that .pde files for the projects are installed on the computers that students will be using

---

### Massachusetts Technology Literacy Standards and Expectations: 6-8\textsuperscript{th} Grade

<table>
<thead>
<tr>
<th>Standard 1. Demonstrate proficiency in the use of computers and applications, as well as an understanding of the concepts underlying hardware, software, and connectivity.</th>
<th><strong>Basic Operations</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.1</strong> Use features of a computer operating system (e.g., determine available space on local storage devices and remote storage resources, access the size and format of files, identify the version of an application).</td>
<td><strong>1.1</strong> Identify successful troubleshooting strategies for minor hardware and software issues/problems (e.g., “frozen screen”).</td>
</tr>
<tr>
<td><strong>1.2</strong> Independently operate peripheral equipment (e.g., scanner, digital camera, camcorder), if available.</td>
<td><strong>1.3</strong> Independently operate peripheral equipment (e.g., scanner, digital camera, camcorder), if available.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standard 3. Demonstrate the ability to use technology for research, critical thinking, problem solving, decision making, communication, collaboration, creativity, and innovation.</th>
<th><strong>Problem Solving</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3.4</strong> Independently use appropriate technology tools (e.g., graphic organizer) to define problems and propose hypotheses.</td>
<td><strong>3.5</strong> Use and modify databases and spreadsheets to analyze data and propose solutions.</td>
</tr>
</tbody>
</table>

---

### Massachusetts Technology Literacy Standards and Expectations: 9-12\textsuperscript{th} Grade

<table>
<thead>
<tr>
<th>Standard 1. Demonstrate proficiency in the use of computers and applications, as well as an understanding of the concepts underlying hardware, software, and connectivity.</th>
<th><strong>Basic Operations</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.1</strong> Identify the platform, version, properties, function, and</td>
<td><strong>1.1</strong> Identify the platform, version, properties, function, and</td>
</tr>
</tbody>
</table>

---

concepts underlying hardware, software, and connectivity. interoperability of computing devices including a wide range of devices that compute and/or manage digital media.  

1.5 Explain criteria for evaluating hardware and software appropriate for a given task (e.g., features, versions, capacity).

Standard 3. Demonstrate the ability to use technology for research, critical thinking, problem solving, decision making, communication, collaboration, creativity, and innovation.

Problem Solving  
3.5 Explain and demonstrate how specialized technology tools can be used for problem solving, decision making, and creativity in all subject areas (e.g., simulation software, environmental probes, computer-aided design, geographic information systems, dynamic geometric software, graphing calculators, art and music composition software).

The supply costs included are the up front cost of purchasing the equipment. After equipment is purchased, this workshop can be run at no cost!

<table>
<thead>
<tr>
<th>Supply</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino microcontroller [<a href="http://www.sparkfun.com">www.sparkfun.com</a>]</td>
<td>$20-$25.00 a piece</td>
</tr>
<tr>
<td>Arduino starter kit – includes microcontroller, USB cable, breadboard, jumper wires, flex sensor, softpot, photocell, thermistor, LED (Tri-color), Basic LEDs, Linear trin pot, buzzer, 12mm button, resistors</td>
<td>$53.96</td>
</tr>
<tr>
<td>USB cord, breadboard, resistors, LEDs, jumper wires</td>
<td>$15.00</td>
</tr>
<tr>
<td>10 computers/laptops</td>
<td>Borrowed computer lab</td>
</tr>
<tr>
<td></td>
<td>$350 for ten or $539.60</td>
</tr>
</tbody>
</table>

*Costs based on workshop with 20 students. Cost/student= $17.50 – 26.98  2 staff recommended