Six Weeks to Coding

A six week program of 60 minute segments aimed at teaching the fundamentals of computers and coding to 10-12 year olds (grades 5-6). Some key goals for the program:

- Designed for students with absolutely no prior computing experience
- Focus on real world applications on computing, how they help us solve problems in the world around us
- Emphasizing hands on exercises supplemented with brief overview and discussion

Course given at Elgin Street Public School, January-March 2016, and April-May 2016

Week 1: Variables and Encoding

This week will introduce some very basic concepts that are heavily used in programming and computer science. You can leap straight into the activity and then circle back to talk about the concepts at the end.

Encoding: Using one thing to represent another thing. Written words and numbers are the most common examples of encodings used in everyday life. We encode a quantity of things as a number, and encode verbal communication with letters and words. Encoding is something that comes up over and over again in computer science. Computers are very stupid and can only understand “on” or “off”. This means programmers have to use a lot of techniques to encode words, numbers, etc as sequences of zeroes and ones.

Variables: Represents an encoded thing whose value can change over time. Examples of variables: “The amount of milk in my fridge”, “The time until lunch”. Variables are used all the time in computer programming

Activity: create a “spy decoder” ring. This helps them learn about encoding and the idea of a variable that can change. Each child can take one home at the end.

http://frugalfun4boys.com/2014/04/15/code-activity-kids-make-spy-decoder/

Give the students a secret message and see if they can decipher it. They should quickly realize they will need to know what wheel position to use. Give them the position and wait for them to decipher the message. How else can the wheel be used to make a code that is harder to crack? For example, the wheel could be turned after each encoded letter, or after each word…

As a final activity, have the students encode their own name or a secret word with a wheel position of their choice. The instructor can plug the names into this decoder program to illustrate how fast a computer can manipulate encodings:

https://github.com/jarthorn/little_bits/tree/master/decoder

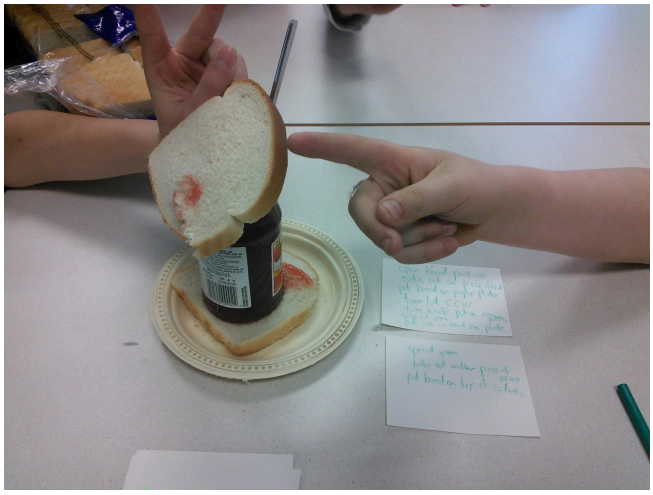
Historic trivia: During the second world war, Germany used a more complicated version of this ring to scramble their military communications (Enigma machine). They believed it was unbreakable. A group led by Alan Turing invented one of the world's first computers that was able to decode the ring, and this was one of the major factors in winning the war. Even today in computer theory, computers are called "Turing Machines" and a general purpose computer is said to be Turing Complete". The highest prize that can be awarded in computer science is called the "Turing Prize", which is something like the Nobel Prize in other fields.

Materials: paper, paper fasteners, sharpie/markers, scissors, computer with decoder problem and Ruby installed

Week 2: Algorithms

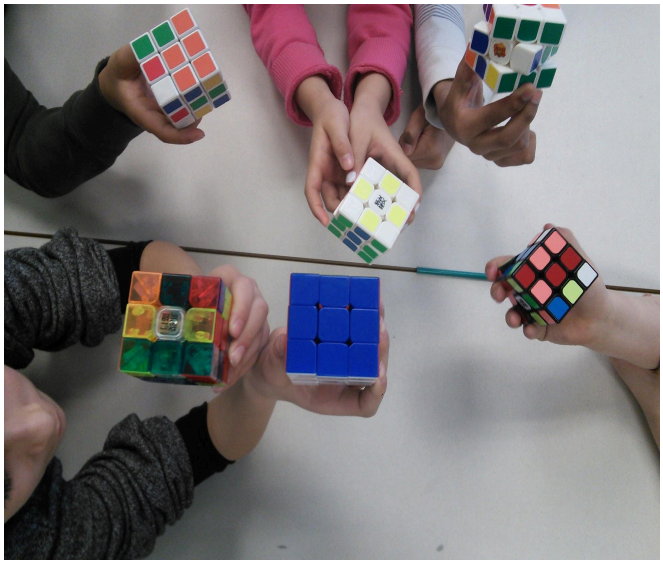
Learn about how computers execute sequences of steps called algorithms. You have to be very precise and pedantic because computers are completely literal.

Activity: Use "peanut butter robot" exercise (with school safe alternative to PB)



http://projectguts.org/files/pbjrobot.pdf

Activity: Bring Rubiks cubes, and show how specific cube configurations can be solved with known algorithms. This time the students are the “computers”, and they are given a carefully scrambled cube and an algorithm that will solve it. Have them implement the algorithm by following the instructions. Use simple 90 degree turns using only the top face (U) and right face (R). You can pick any scramble and then the solution is just inverting it. For example: Scramble 1: R, U, R back, U back. Solution: U, R, U back, R back.

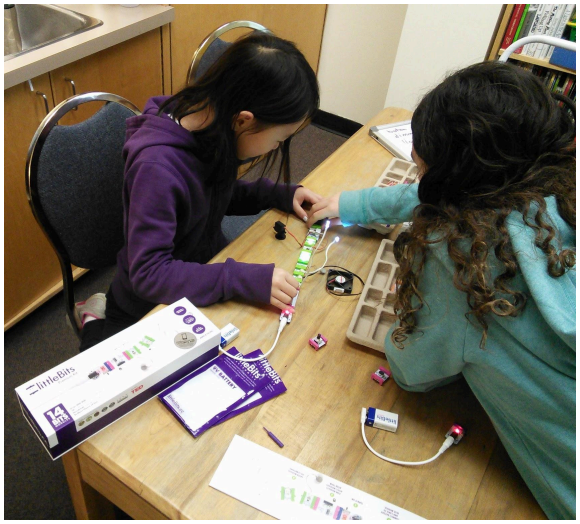


Materials: crackers, spreadable peanut butter alternative, knife, plate, Rubiks cubes

Week 3: Inputs and Outputs

Most computers take some kind of input, do an algorithm, and then give some kind of output. Have students list some examples of computer inputs (touchscreen, mouse, keyboard, buttons, switches, sensors), and outputs (screens, printers, speakers, networks). Bring some typical and unusual examples to show.

Explore the idea of different kinds of inputs and outputs with LittleBits. Introduce LittleBits. Start with only power (blue) and output (green). Have the students work in groups to figure out what each of them does and write it down. Then using only power and bargraph, explore each of the inputs (pink) and figure out what they do.



Week 4: Project with Electric Circuits

Introduce the concept of digital circuits, with the following terms:

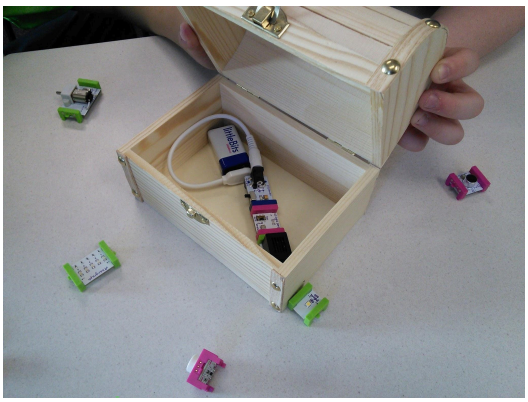
> **Circuit**: A closed path through which an electric current flows

There are two major kinds of circuits:

1. **Digital**: digital messages are encoded (ref week 1) as a sequence of “on” and “off” states
2. **Analog**: analog inputs/outputs have a varying range from a low point to a high point

Analog circuits have been around for over a hundred years, and are used all around us. For example the electricity in your home such as light bulbs and light switches are commonly analog circuits. Computers are not very clever so they prefer the simpler digital circuits.

A neat thing about littleBits is that they can act as both digital and analog circuits. Demonstrate examples of both analog (dimmer + LED) and digital (dimmer + bar graph) circuits. See how the bargraph is showing a number from 1 to 5, whereas the LED just gets brighter and dimmer. Show how each bit has three pins on each side. Two on the outside are for the analog electric circuit (negative and positive/ground), and the middle pin is for a digital signal.



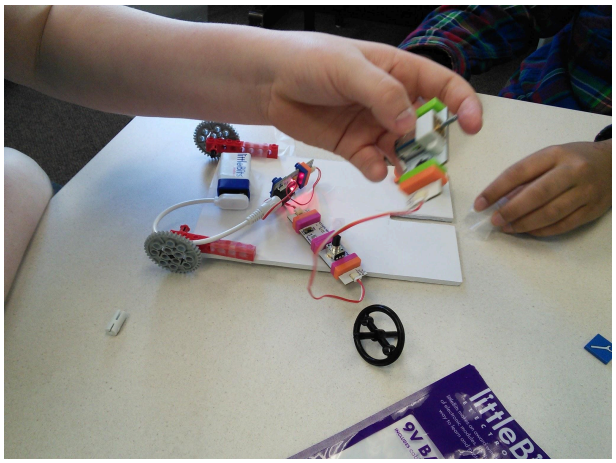
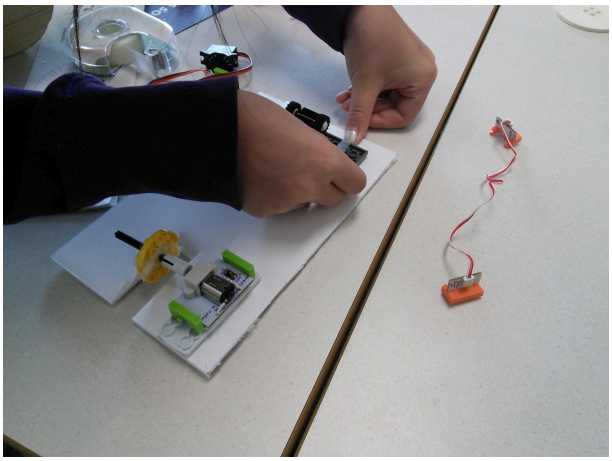
Give the children a mission or problem to solve using LittleBits, and let them figure out how to solve it on their own. Give them a box or treasure chest, and tell them to build an alarm system that will alert them when an intruder tries to open the box. This can be accomplished

most easily with the sound trigger bit, but can also be done with a roller switch or even a sound trigger.

Alternative ideas:

https://s3.amazonaws.com/littleBits_pdfs/littleBitsEducatorsGuide_FINAL.pdf
https://chowmiller.wordpress.com/2015/10/26/littlebits-first-lesson-a-success-with-digital-and-analog-circuits/

If they complete that quickly, let them build another project from the littleBits guide book. A popular choice is the three-wheeler car from the littleBits starter kit.

Materials: LittleBits, cardboard boxes, lego wheels, tape

Week 5: Simple Visual Coding

Do either the Star Wars or Frozen activity from hour of code:

https://code.org/frzn
https://code.org/starwars

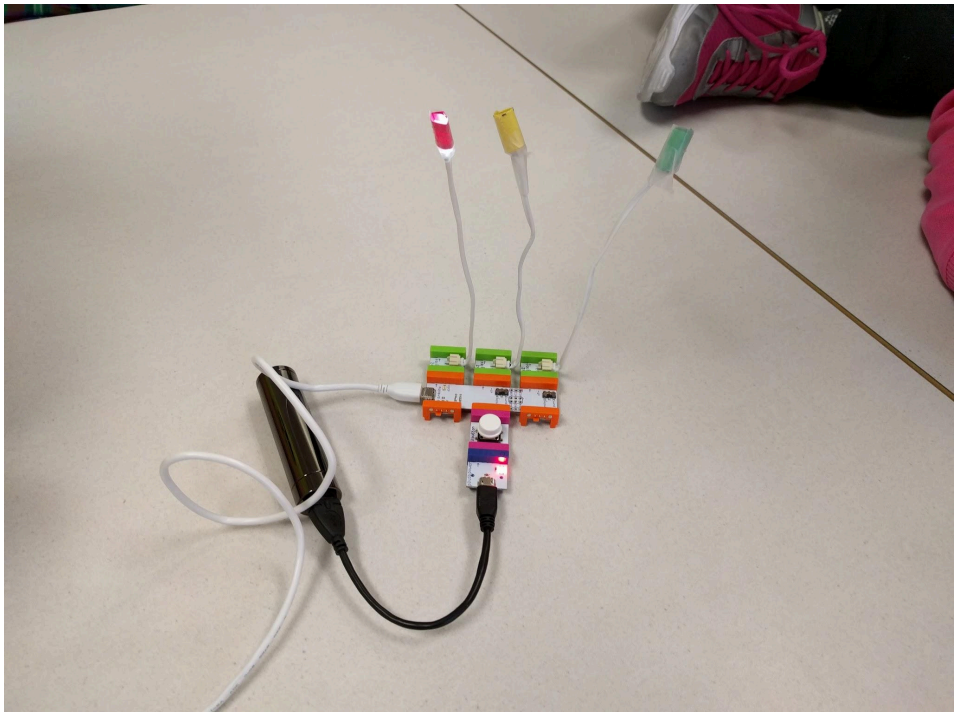
Materials: iPads, Chromebooks

Week 6: Project with Coding

Arduino coding with LittleBits. Have a basic Arduino program running and a project idea that the kids can assemble (both the circuits and the rest of the project). See how the program works on the littleBits design, and iterate a few code changes to the program as a group to see what effect it has.

Project 1: Traffic Light

Circuit: Power -> Button -> Arduino -> Three lights of any kind on the three output pins
Code: https://github.com/jarthorn/little_bits/tree/master/examples/traffic_light_mar09a



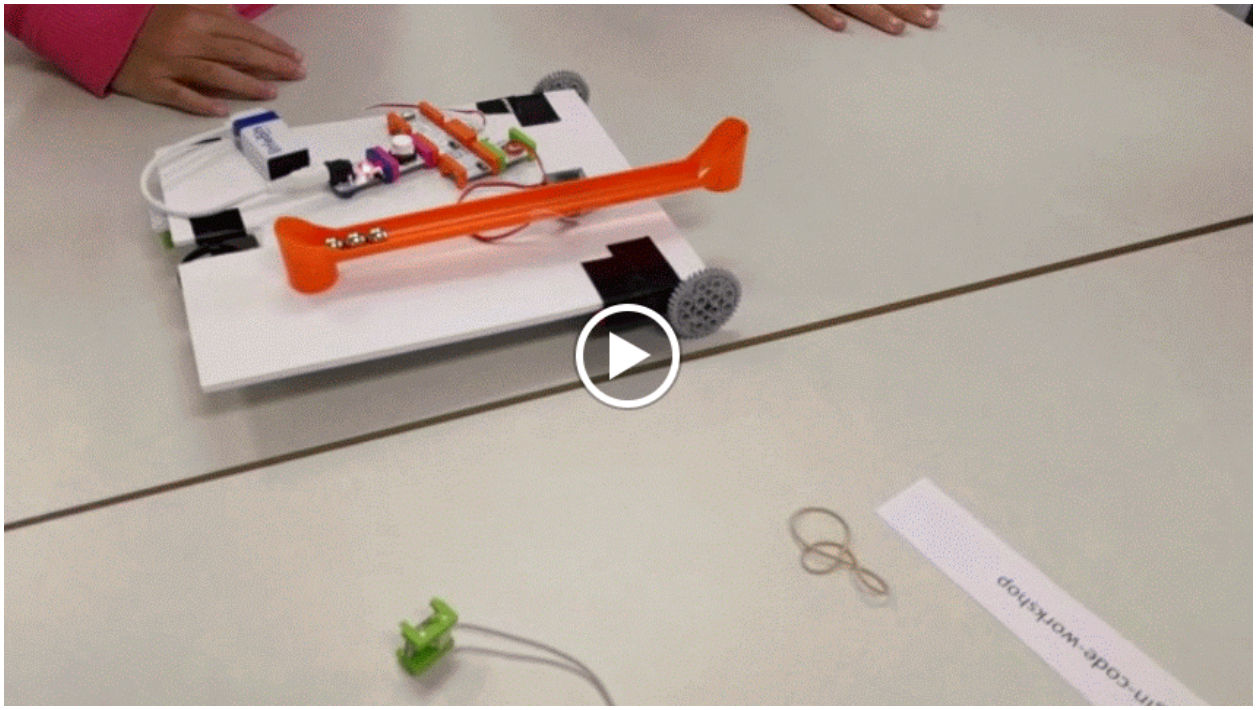
Use green, amber, and red paper or other semi-translucent cover over three long LEDs. Improvise as needed depending on available bits. The idea is that this is a traffic light with a pedestrian crosswalk button. It starts with the green light on. Pushing the button will change the light to amber, then red. The instructor can do the coding with input from the children on what to write. Experiment with changing delays to get something that feels right.

Project 2: Dump Truck

Circuit: Power -> Button -> Arduino -> Servo (D5), DC Motor (D9)
Code: https://github.com/jarthorn/little_bits/tree/master/examples/dump_truck_mar05a

This starts with the “three wheeler” project design from the basic LittleBits kit. On top, use a servo motor with a paperclip through the motor arm to lift/lower a tube (you could use a paper towel roll). This is the “dumper” part of the truck. Close one end of the tube and put a marble inside it. The servo is programmed to lower the tube and “dump” the marble. As a challenge, place a shallow container on the floor and see if the children can help program the dump truck to back up and tip the marble into the container from some fixed start position.

The button can be replaced with a sound or light trigger to make the truck move based on sound/light trigger.



Watch a video of the robotic dump truck the class built

http://littlebits.cc/lessons/arduino-intro-workshop-girls-who-code
http://superawesomebook.com/riff/
http://littlebits.cc/lessons/experiment-the-pendulum

Materials: LittleBits kits, LittleBits Arduino, wheels, stiff cardboard platform, marble, paper tube, paper clip, colored paper, laptop setup for Arduino programming

The Learning Continues

Here are some resources to look into for students wanting to continue exploring technology and coding:

https://code.org/ - A web site full of learning materials for children learning to code
http://carleton.ca/vv/ - Summer camps and workshops on technology run by Carleton Engineering Department
http://superawesomebook.com/ - Another web site with tech/code material for children
http://csunplugged.org/ - A web site with computer science topics you can learn at home without a computer

http://barefootcas.org.uk/ - teaching materials for primary grades