Teaching with Open Boxes: Datalogging in the Classroom

Daniel Bergey & Christalee Bieber
The Hacktory, Philadelphia PA
bergey@thehacktory.org, cbieber@thehacktory.org

To develop skills and confidence with technology, beginners need systems which they can take apart, modify, and occasionally break. Open hardware hackers can help by writing good documentation, designing systems which are simple to understand, and contextualizing our work in our communities. People of various levels of experience learn that technology is comprehensible & controllable when they form connections to their current activities and build on their strengths and prior knowledge.

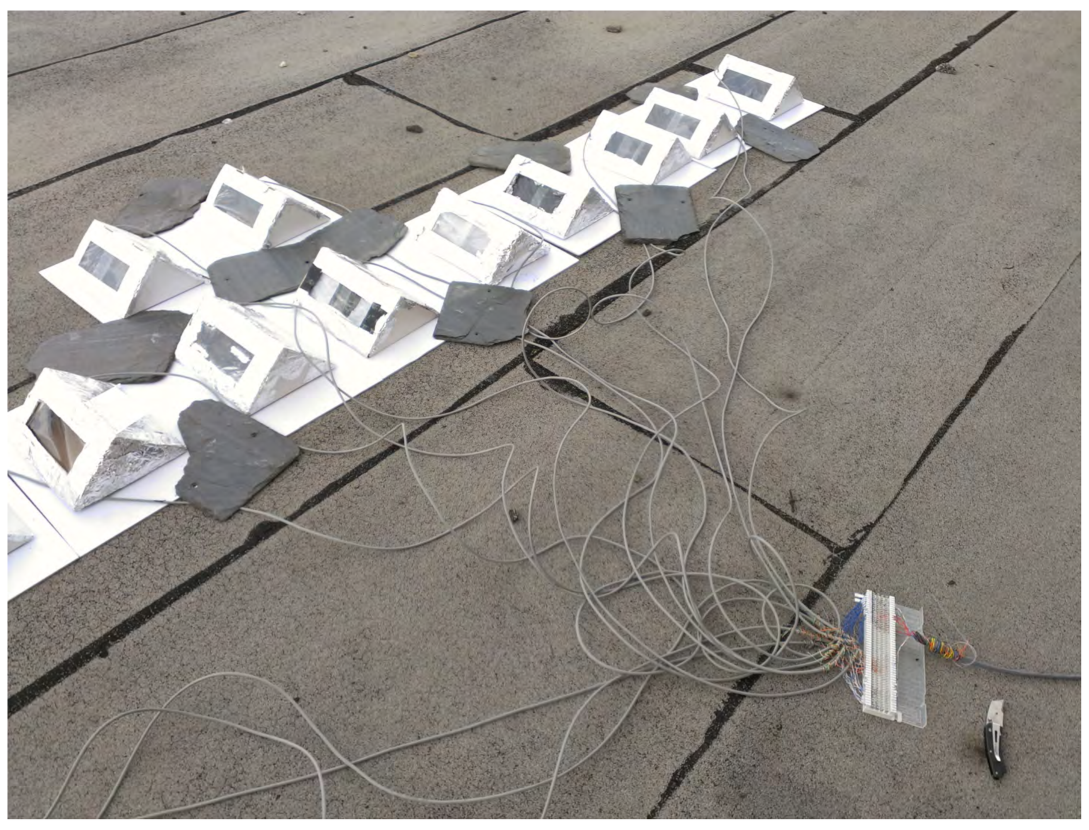
We teach about circuits, sensors, programming, and data visualization at our local hackerspace. To reach a wider audience, we are introducing open hardware into a nearby public school. Our project integrated with a unit on solar energy, where students build cardboard houses with features to reject or retain heat.

We built an Internet-connected datalogger for use by about 60 students in the 5th & 6th grade science classes, measuring temperature in 14 houses over 5 days. We installed the houses & sensors on the school roof, showed students the hardware & software involved, and facilitated class discussion & graphing of the data. Each class looked at the performance of their houses and discussed patterns caused by weather, daylight hours, and mass & window treatments of the houses.

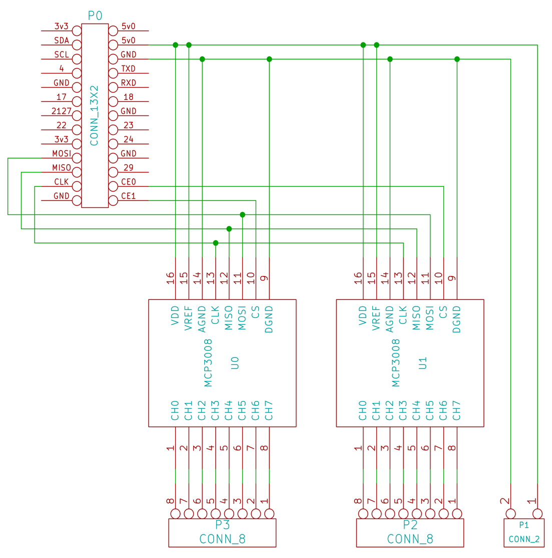
Real-world data, with all its imperfections, sparks engagement more easily than idealized graphs in textbooks. Replacing expensive black-box technology with open hardware increases the potential for student-driven learning and helps to demystify engineering practice.

Hardware

Groups of 3 or 4 students decided whether their design goal was to stay warm or stay cool, and built a solar house tuned to that goal. Each house started with the same cardboard frame, with a hole to act as a window. Students added cups of dirt, water, or sand to store heat, one or more layers of plastic wrap over the window, and aluminum foil & black paper to cover the exterior. One house frame was left unmodified to act as a control.

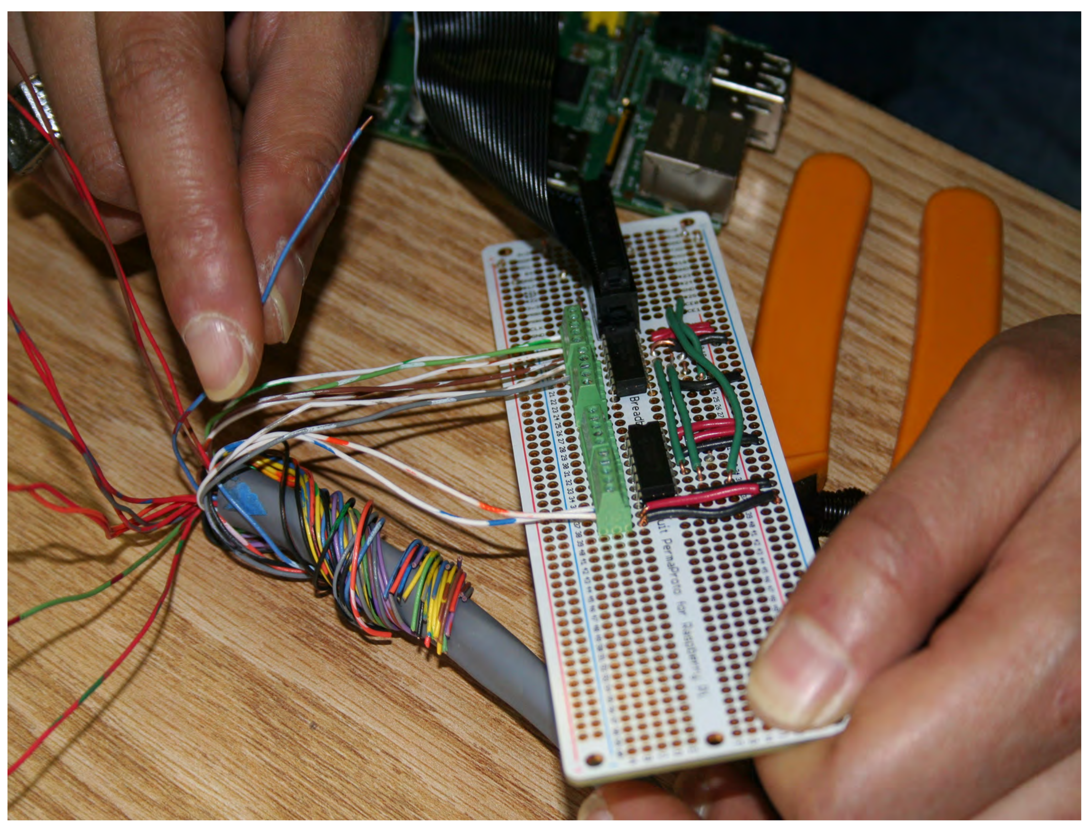


We assembled a simple sensor for each house by soldering a thermistor and matched precision resistor in series to one end of a CAT-5 cable. A punchdown block on the roof connected power, ground, and signal lines to a trunk cable running to the school computer lab, where the ADC board was installed.



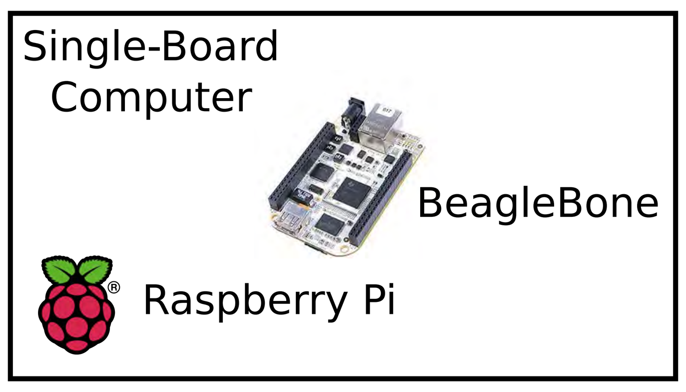
A previous version of this project used commercial dataloggers each with its own battery, temperature sensor, and flash memory. Populating 14 houses with these loggers would be prohibitively expensive. From prior experience collecting data in buildings, we knew we wanted hardware which could:

- Accept arbitrary analog-voltage sensors
- Expand to more input channels
- Be programmed in a popular language



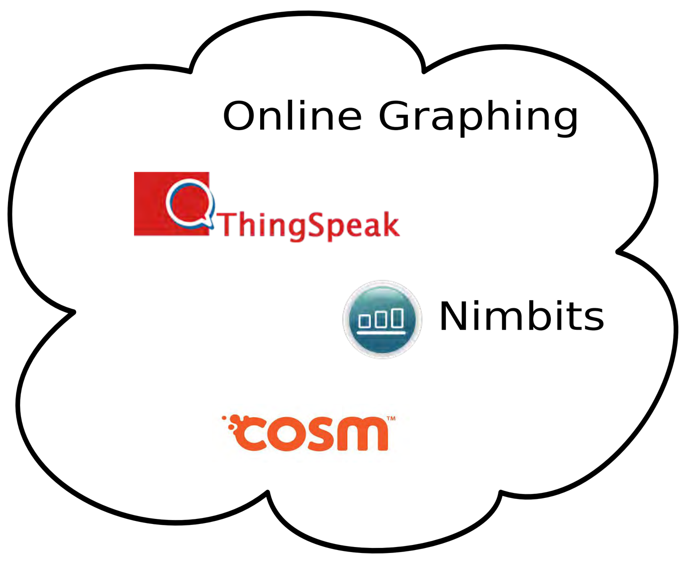
Our board connects two 8-channel ADC chips (MCP3008) to the Raspberry Pi over SPI. We picked this chip because it had already been used in several well-documented open hardware projects. Without much prior SPI experience, it was easy to modify code from these projects to handle two chips instead of one.

Software

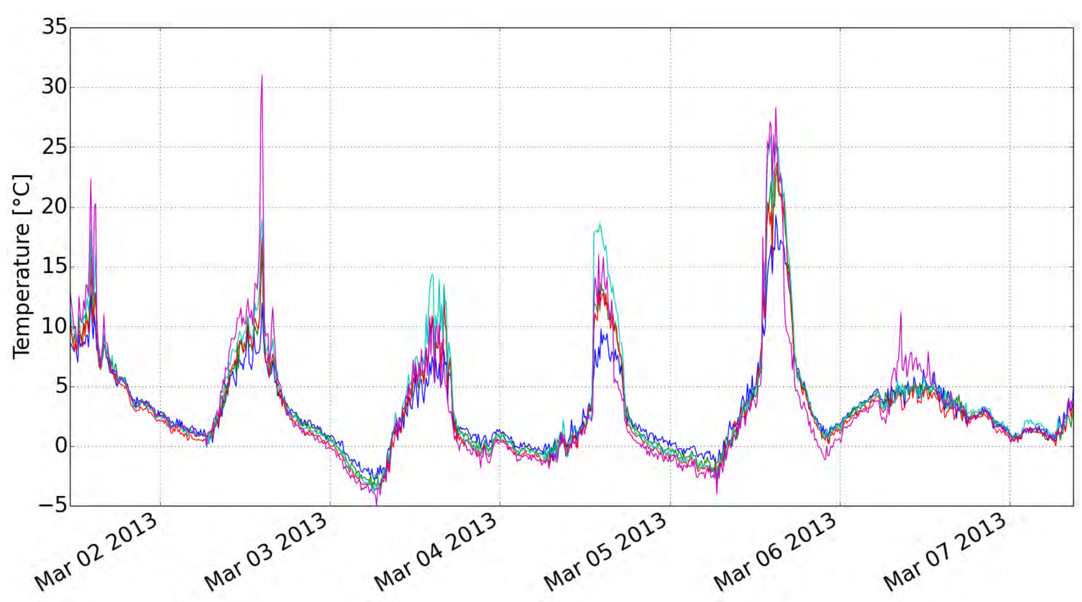



```
def readadc(chip, channel):
    """read channel ADCNUM of the MCP3008 chip"""
    if ((channel > 7) or (channel < 0)):
        return -1
    r = chip.xfer2([1,(8+channel)<<4,0])
    adcout = ((r[1]&3) << 8) + r[2]
    return adcout
```

We wrote a Python script to sample the temperature of each house every 10 minutes, write the result to a local CSV file, and upload the data for online display. Several online services offer APIs for embedded devices to append data to a feed, for easy display & retrieval. This allows students to view their houses' temperature while the experiment is in progress, and to show family & friends. It also allows early detection of problems with the logger installation.

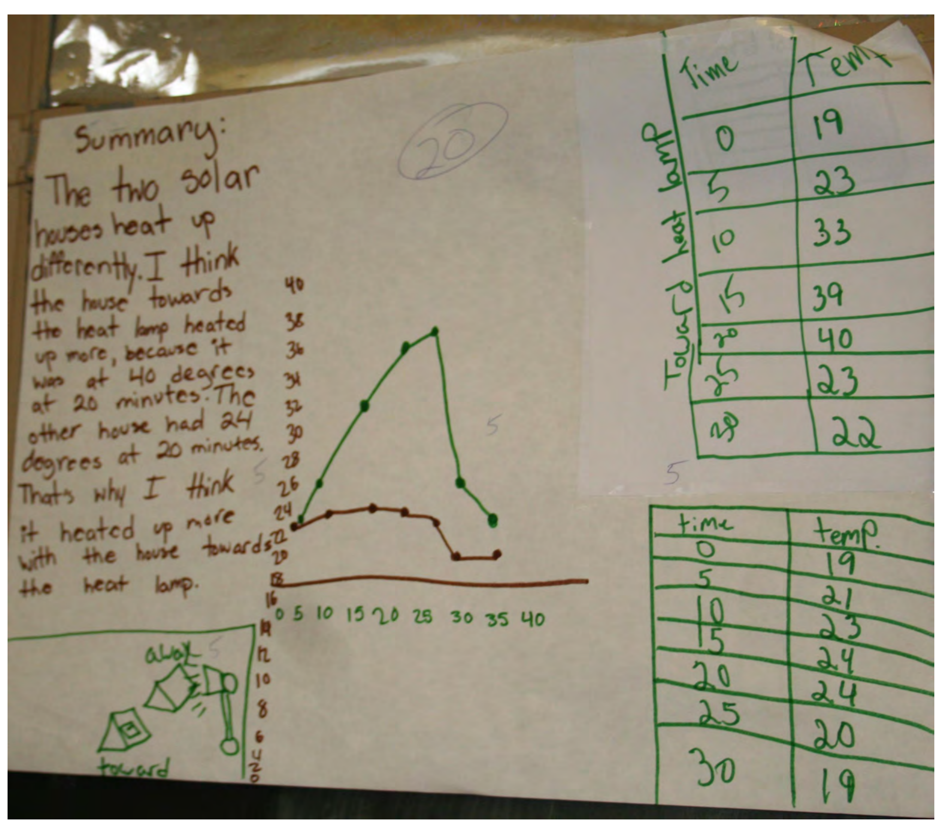


The graph below shows temperatures in 4 houses constructed by one class, and the control (blue line). Daily peaks fall in the early afternoon, except on the 6th, when clouds mostly blocked direct radiation. All houses had at least one layer of plastic over the window, which caused them to heat up faster than the control. We graphed the data for each class in Excel and projected it in the classroom. This allowed zooming in when the discussion required a closer look at one day of data. Students compared the houses' performance to each other and to original design goals.



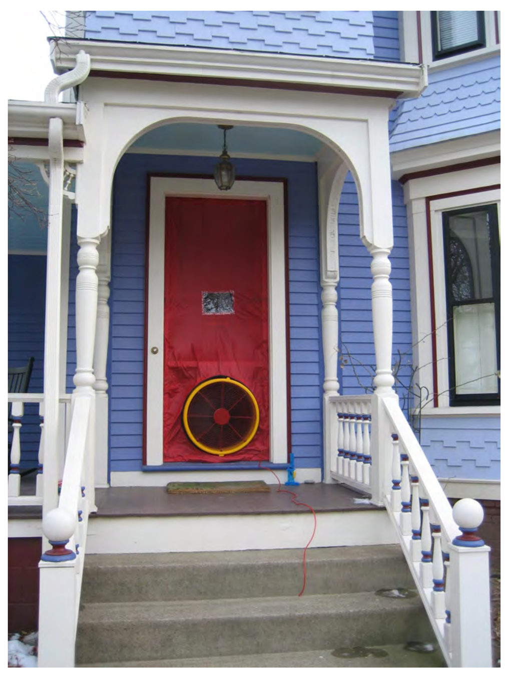
Visual representation and interpretation of data are key literacy skills. Using real data and computer graphing for this exercise encourages students to view their environment as something that they can model.

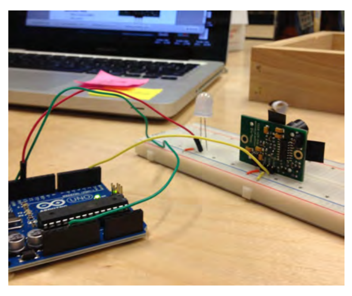
Prior to designing their houses for the rooftop experiment, students performed more controlled experiments in the classroom, with electric lamps substituting for sunlight. They measured temperature and prepared graphs manually.

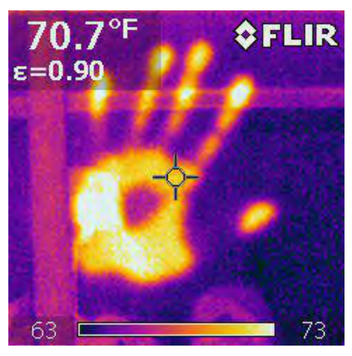


Connections

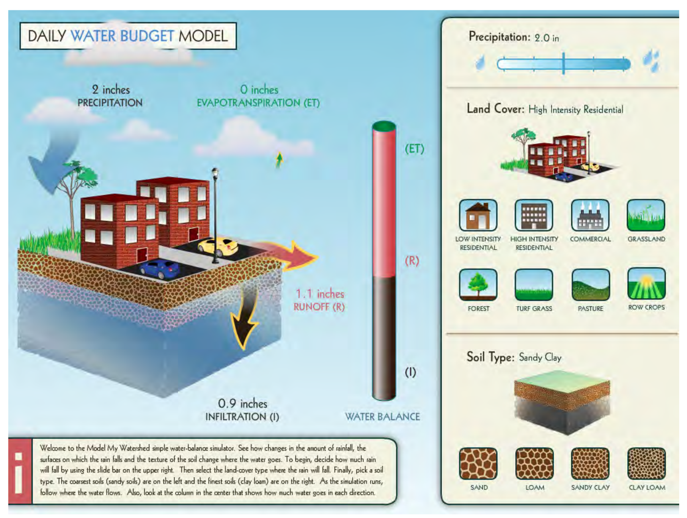
The students building solar houses studied circuits in 4th grade, so they are well-prepared to understand the analog (temperature-sensing) part of the circuit. For classes that have studied programming, the python script is simple for beginners while demonstrating use of CSV, JSON, HTTP, and SPI libraries. In-class exposure to open hardware may inspire some students to learn practical electronics skills like breadboarding & soldering.







The 5th & 6th grade science curriculum at this school is designed to connect broadly applicable scientific ideas with specific examples taken from the local urban environment. Erosion & runoff are illustrated with quantitative examples from the city about rainfall and surface permeability. We used blower doors and infrared cameras to help students identify air leaks around windows, radiators, and soffits.



References

Raspberry Pi <http://www.raspberrypi.org/>

Adafruit Pi T-Cobbler Breakout Kit for Raspberry Pi <https://www.adafruit.com/products/1105>

Adafruit Perma-Proto Raspberry Pi Breadboard PCB Kit <https://www.adafruit.com/products/1135>

Microchip MCP3008 10-bit Analog-to-Digital Converter <https://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010530>

Stroud Water Research Center. "Model My Watershed." <http://www.stroudcenter.org/mmw/mini/>

Ladyada. "Adafruit Rasberry Pi Educational Linux Distro." <http://learn.adafruit.com/adafruit-raspberry-pi-educational-linux-distro> Adafruit Learning System, August 3, 2012.

doceme. "py-spidev." <https://github.com/doceme/py-spidev> GitHub, June 11, 2012.

Blythe, Jeremy. "Raspberry Pi hardware SPI analog inputs using the MCP3008." <http://jeremyblythe.blogspot.com/2012/09/raspberry-pi-hardware-spi-analog-inputs.html> Jeremy's Blog, September 5, 2012.

petervizi. "python-eeml: A python package for generating eeml documents." <https://github.com/petervizi/python-eeml> GitHub, April 26, 2013.

bergey. "berrybasket: Raspberry Pi logging infrastructure (Cosm & MCP3008)." <https://github.com/bergey/berrybasket> GitHub, February 25, 2013.

verizon

Graduate School of Education Penn GSE

This work was supported by the Verizon Foundation and the University of Pennsylvania Graduate School of Education. The instructional content of this project was conceived and led by NancyLee Bergey, Associate Director of Teacher Education & Ed.D candidate at Penn GSE.

THE HACKTORY

http://thehacktory.org

http://bergey.github.io/berrybasket/