

Under the Hood:

Using Raspberry Pis and WiFis to Do More with Data

By Lisa Hardy and Matt Lewandowski



Lisa Hardy
(lhardy@concord.org)
is a research associate.



Matt Lewandowski
(mattkain.lewandowski@gmail.com)
is a software developer who has worked on many Concord Consortium projects as a consultant.

For decades students have been using sensors in science classrooms to collect real-time data. In this age of the “Internet of Things” (IoT), sensors are now commonplace, cheap, and amenable to DIY projects. Beyond just creating more data, how can we improve the ways that students interact with and understand data?

The goal of the InSPECT (Integrated Science Practices Enhanced by Computational Thinking) project is to help students do more than simply look at graphs of sensor data. We want them to do things with data: to build dynamic systems that can both sense changes in the environment and respond to them—whether it’s turning on a fan when a room gets too stuffy or stabilizing temperature using a Peltier cooler. With our partners at Manylabs, we’re developing Dataflow, a browser-based application that (unlike other IoT platforms) not only lets students collect and view sensor data, but also allows them to write programs that transform that data to control physical actuators.

To do this we needed to develop WiFi-connected devices that could execute students’ programs and control actuators based on sensor data. Since we want students to understand how Dataflow works, they also need a real-time display of all the collected sensor data and how that data flows through their programs to determine actuator states.

Dataflow uses Raspberry Pis as WiFi-enabled devices that can execute these control programs because they are inexpensive but powerful computers. The Pis run a client program that reads data from all

USB-connected sensors, runs student programs, and sets the states of USB-connected actuators. When a Pi boots up, it is available to respond to requests for sensor data and to run students’ programs.

In the browser, students connect onscreen “blocks” and “wires” to create the Dataflow programs that are serialized and sent to a Raspberry Pi, where they are run (Figure 1). The client software de-serializes the program as a set of linked, executable “blocks.” Each block has:

- other blocks linked as inputs,
- a function to execute when those inputs are defined, and
- a single output value.

To run these programs, the Pi client first fills all sensor-type blocks with values from the connected sensors, then loops over the entire set of blocks, requesting each to attempt to execute. In each loop iteration, the sensor values “flow” through the program and are processed by each block, until each block in the program has a defined output value. The client then uses any actuator-type block values to set the states of connected actuators and any storage-type block values to send data to the server to be persisted in a database and accessed later.

To display a real-time representation of the “flow” of data through those programs as they run, we implemented bi-directional communications between the server and the Raspberry Pis. We send students’ control programs to the Raspberry Pis over a websocket, and in response have the Pis send back (once per second) the current state of each block in the running student program. We then feed those states into the real-time representation of the student’s blocks-and-wires program in the browser.

Students are learning about computation and control systems as they create control programs and see how the sensor data flows through those programs to determine actuator states. We’re adapting low-cost IoT technologies to give students new ways of doing things with data—beyond just looking at it.

Note: Find the open-source Dataflow code on GitHub at the link below.

LINKS

InSPECT
<https://concord.org/inspect>

Manylabs
<https://www.manylabs.org>

Dataflow repo
<https://github.com/concord-consortium/flow-server>

This material is based upon work supported by the National Science Foundation under grant DRL-1640054. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Figure 1. A Dataflow program to turn on a relay when the temperature is above 25 degrees.