

Educational Videos to Teach Parallel Computing on the Raspberry Pi

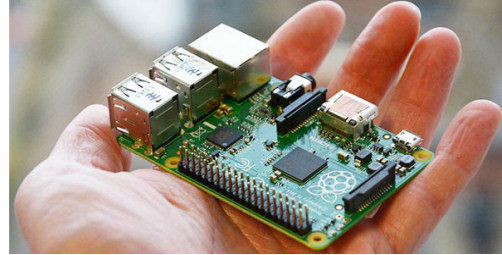
Margaret Zimmermann, Jesus Caballero Jr., Richard Brown
Department of Mathematics, Statistics, and Computer Science
St. Olaf College
Northfield, MN 55057
zimmer1@stolaf.edu, caball1@stolaf.edu, rab@stolaf.edu

Abstract

The Raspberry Pi is a credit-card sized single-board computer (SBC) designed for teaching, learning, and applying computation. We have created educational videos for teaching and learning about parallel and distributed computing (PDC). We have created feedback forms to assess the videos for teaching parallel computing with the Raspberry Pi as well as the benefits of using the videos as a teaching tool. These videos come in series for Macintosh and Windows computers that teach the user how to image a microSD card, connect the Raspberry Pi to a computer, connect the Raspberry Pi to a monitor, as well as how to run parallel programs with the quad-core CPU of the Raspberry Pi. Modules from the CSinParallel project (csinparallel.org) were used to teach parallel programming concepts. We describe the freely available Raspberry Pi software image developed for this project, the video development process, and assessment results.

Introduction

The Raspberry Pi 3 is the latest in a series of single-board computers designed by the Raspberry Pi Foundation in Britain, intended for teaching computer science in schools and by hobbyists around the world. About the size of a credit card and costing about \$35, the Pi has a quad-core processor, meaning that it can physically execute up to four programs at the same time, and provides many hardware interface connections for network communications, sensors, and devices such as a speaker and a camera. Linux-based system software and utilities, together with well-developed and accessible documentation and support resources, have made the Pi series the best-selling British computer, with over 10 million units sold. [Gibbs 2015, Raspberry Pi 2016]



The Pi's low cost, wide accessibility, multicore processor, and hands-on appeal make it an attractive platform for teaching parallel computation through brief in-class exercises using CSinParallel teaching materials. The CSinParallel project (csinparallel.org) produces modular teaching materials for students to learn principles and practices of parallel and distributed computing (PDC) through hands-on exercises that require only one to three days of class time. When carried out on a Raspberry Pi, we assert that PDC concepts can become reinforced not only by the software exercises but also by hardware interaction.

Our project seeks to demonstrate that assertion through the use of specially configured Pi system software that supports direct connection to a laptop, and educational videos that enable students to set up, connect to, and operate a Pi using their laptop. The laptop connection has proven simpler and quicker to assemble than the usual attachment of a keyboard, monitor, and mouse to a Pi, leaving more time for hands-on exercises focused on learning computer science principles. The use of videos makes it possible for students to understand how to connect to the Pi using their laptop, and to set up the custom software configuration as a homework assignment. The combination of laptop interaction and video support has made it possible for students to obtain a parallel computing learning experience on the Raspberry Pi 3 within a single class period or less.

Background

Raspberry Pi Single-Board Computers

This project emerged in the Parallel and Distributed Computing Course at St.Olaf College. We sought to produce an interactive platform for teaching parallel and distributed computing using Raspberry Pi 3's. The Raspberry Pi is a credit card sized single-board computer designed for teaching learning and applying computation. Single-board computers (SBCs) are miniature complete computers including processor, memory and storage, requiring only monitor and keyboard/mouse for full functionality.

Raspberry Pi 3 Model B	
Introduction Date	2/29/2016
CPU	Quad Cortex A53 @ 1.2 GHz
Instruction Set	ARMv8-A
GPU	400MHz VideoCore IV
RAM	1GB SDRAM
Storage	micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
GPIO	40
Price	\$35

Figure 1. Raspberry Pi 3 Model B specs

The Raspberry Pi comes equipped with 1GB of RAM and Quad core Arm Processor running at 1.2 GHZ. A core is a circuit for carrying out instructions having 4 makes it able to carry out multiple instructions at once. This aspect makes the raspberry pi a good candidate for running parallel programs. The Pi sells for \$35 dollars and features a 40 pin GPIO (General Purpose Input and Output) circuit board that enables the user to control the behavior of the 40 pins. A breadboard may be mounted to make additional circuits, for example running LEDs, or using sensors, and other interfaces include a full-sized HDMI port, camera port, video display, and audio jack. To power up your Pi you only need a micro USB B cable and a power source that provides 5 volts, this makes the Raspberry Pi particularly power efficient.

A system image is a special type of file that file which contains an operating system and software to operate a Raspberry Pi. To use a system image on a Pi, one must install that special file on a microSD card, insert that microSD card on that Pi, and boot the Pi by connecting it to power. We made our own custom system system image from a fresh copy of the Raspbian operating system for Pis to serve our needs. As usual, our



Figure 2. microSD Size Comparison

system image supports connections using a monitor, keyboard and mouse. In addition, one can connect to a Pi using a Windows or Macintosh laptop using our system image. This is accomplished by operating a DHCP server on the Raspberry Pi, which provides a local IP address to the user's laptop when connected via Ethernet. Assuming availability of a laptop, this approach requires fewer hardware components and making it more convenient.



Figure 3. Connecting to a Raspberry Pi using monitor, keyboard, and mouse.



Figure 4. Connecting to a Raspberry Pi using a laptop.

Teaching with Videos

Educational videos engage students of all knowledge levels and allow professors to cover all the necessary material in a short course period. The goal of teaching videos involves providing a more comprehensive classroom that tailors to the needs of all students at any level. Videos designed for education settings contribute to effective learning in several ways. These videos benefit students by providing them an alternate form of media for learning. This gives students the option to stop and rewatch the videos as much as they want, until they fully understand the concepts. No matter what level of computer science knowledge, these videos allow students to study at their own pace in their own learning environment. This guarantees that all students receive the in depth explanation on a course topic if they need that extra support. In addition to students having more time to fully understand the concepts, professors can make sure that they cover all the topics necessary in the videos. During normal class periods, professors find it difficult to fit all the necessary material into the limited class hours. With the use of videos, professors do not have to worry about having enough time for the whole lecture and thus can spend the class hour working in groups. [Sadaghiani 2012]

Educational videos can be used to “flip the classroom” in CS courses. Instead of professors lecturing during class time then students carrying out homework assignments outside of class, the “flipped classroom” approach assigns video presentations for students to watch as homework before the class period. During that class meeting, students then work on group problems. If any questions arise, the professor can step in to help. This gives students greater depth of understanding on the topics because of the peer collaboration and the different types of problems that can be used in class instead of just for homework. [Prof. Gary Muir, private communication, 2017. Studies have shown that students not only perform better on tests, but have a higher retention rate as well. [Stelzer, 2009] Videos can either be customly created for the course or found through other sources. If professors produce their own videos, they do not have to worry about extraneous material. By using videos, professors can animate concepts in a way unattainable in the typical classroom setting. For example, in one of our PDC videos, we showed students step by step how to image a microSD card on both a Macintosh and Windows computer, a process impossible to show during one class time.

Learning about Parallel Computing

Parallel and distributed computing (PDC) is the technology of using multiple computational circuits and/or multiple computers in order to perform more than one computer operation at the same time. Virtually all computers and computing services now provide PDC capabilities, but not all CS curricula include the education in PDC that students need to program with those technologies.

The CSinParallel initiative (csinparallel.org) provides brief hands-on classroom modules designed for professors at any undergraduate institution to add a wide range of PDC content to their courses, in order to educate their students in these essential technologies, with only modest changes to a course syllabus.

Project Description

During this project we wanted to see whether students learned well using education videos in a CS course to learn about parallel and distributed computing. It originally started in the core course in Parallel and Distributed Computing (PDC) at St. Olaf

College. PDC is the technology of using multiple computational circuits and/or multiple computers in order to carry out more than one computer operation at the same time. Virtually all computers and computing services now provide PDC capabilities, but not all CS curricula include the education that students need to solve problems with those technologies. This project seeks to create videos that allow anyone, even those who may know nothing about coding, to setup and run parallel computing modules on their own Raspberry Pi and monitor or computer. For these videos, we created a custom Raspberry Pi system image that can operate as a standalone or as a Raspberry Pi cluster. It can also readily use a Macintosh or Windows laptop to serve as keyboard, monitor, and mouse. This last fall we presented our work to two computer science classes (“CS 300, *Parallel and Distributed Computing* (PDC)” and “CS 241, *Hardware Design* (HD)”), with students viewing videos ahead or following along in class. The content of the five videos included how to put a custom Raspberry Pi image onto a microSD (on Macintosh or Windows), how to connect a Raspberry Pi to a monitor/keyboard, to a Macintosh computer, to a Windows computer, and how to run and understand parallel learning modules on a Raspberry Pi.

For both *PDC* and *HD*, we followed a similar procedure in how we presented the Raspberry Pi materials. When presenting the material we first asked the students to fill out a pre-survey that asked questions about their level of CS experience, their prior experience with Raspberry Pis, and their prior knowledge about parallel and distributed computing. The professor required that all students completed the survey before they began the videos. For the *PDC* students, who had more experience with parallel and distributed computing, they had to watch the videos (one for a Macintosh and one for a Windows computer) and then image their own microSD card they received during the previous class period. In addition to the video on imaging the microSD card, they also watched the introductory video about how to connect the Raspberry Pi to their personal computer. Students did not have the Raspberry Pis yet because the activities described in the videos formed the basis for the in class activities the next day. After watching the videos, students came to class with the knowledge of how to image a microSD card and what steps they needed to perform in class to connect a Raspberry Pi to their personal computer. We did not have the *PDC* students watch the videos about running a parallel program on the Raspberry Pi because students learned all the same content presented in the video throughout the semester. During class, students worked in pairs or by themselves, depending if they correctly imaged their microSD card, and connected their Pi to their computer. After this they ran the OpenMP parallel program, which computed the area under a trapezoid using different numbers of threads and teaches the concept of a race condition. In this instance, students had already encountered that parallel programming exercise earlier in the term on computers other than Raspberry Pis.

In the *HD* class, students did not have the full flipped classroom experience because we could not assign the videos as homework due to homework constraints. They did have to complete the pre-survey that asked the same questions as the *PDC* students. They also did not watch the imaging a microSD video because we used the pre-imaged microSD cards from the *PDC* students. Before the *HD* class, we checked all the microSD cards to make sure that the *PDC* students did not modify them. When students came to class, we gave them the materials to connect the Raspberry Pi to a computer. Then as a group we watched the two introductory videos about connecting the Pi to their personal computer. After watching the video, students followed the steps (they could rewatch the video online as many times as they wanted if they needed a refresher on the steps), and ran the same parallel program as the *PDC* students. This completes the whole process used to test the videos in the *PDC* and *HD* classes.

To accomplish our goal we had many resources already available to produce these videos and perform the in class exercises. One of them included the above mention custom Raspberry Pi system image. We decided to use our own image because we wanted to include all the teaching modules from the CSinParallel site to allow users easy setup. This way all the modules are on the image and they can simply move right into running code after installing the system image onto their microSD card. Another included premade documentation from prior work that described how to setup a Raspberry Pi. From this documentation we based the descriptions about connecting the Raspberry Pi to a monitor/keyboard. In addition, the modules used in the videos came from the CSinParallel website (csinparallel.org). The overall scope of the videos also came from a presentation done by the CSinParallel group at the Tapia 2016 conference.

Class Testing Results

Before and after leading the in class demonstrations, students filled out surveys describing their previous knowledge about the Pis as well as their levels of experience with parallel computing. Analyzing the qualitative and quantitative data provided by twenty-two students, it showed the benefits of using videos as well as hands on learning in the classroom. Some quantitative data, as shown below in Figure 5, shows that 90% of students “learn well using videos,” 75% felt confident at the tasks described, and 85% stated that they wanted to do more with the Pi in later courses. All of this feedback demonstrated that students learned well from the custom videos as well as the from the

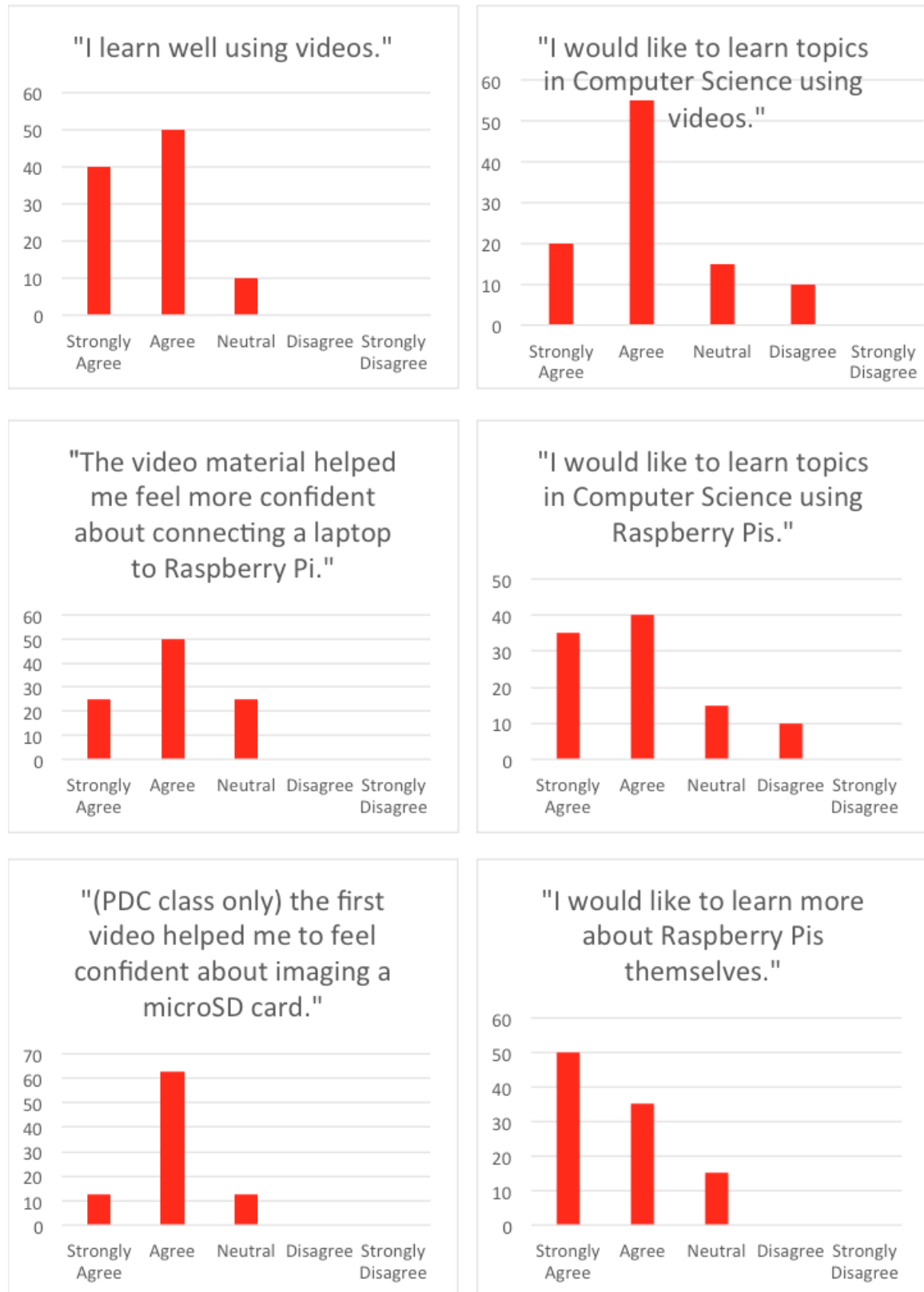


Figure 5. Combined responses to survey questions from *Hardware Design* and *Parallel and Distributed Computing* students (20 respondents).

flipped classroom technique. In addition to all the positive feedback, we saw no negative feedback from students about either using the videos or working with the Raspberry Pis. Studies from the University of Calgary and Carleton University concluded that after taking a course designed around using the Raspberry Pi, the majority of their students also agreed that the Pi motivated and enhanced their learning throughout the course. [Kawash 2016] Overall we found that our project enhanced the learning of students in two different CS courses, as well as gave them hands on experience that most of them never had encountered before in classes.

Experiences

One aspect that made this project particularly difficult included writing scripts. This task requires extensive moderation to insure effectiveness, lack of detail can lead the viewer into a confused state. To avoid confusion for beginners we wrote scripts that require little to no hardware or software experience. This process involves many steps, write the script, edit the script, record finished parts of the script, record dictation of the scripts, edit the video, repeat until finished, watch the video, make final edits, done.

Since these two in class sessions represented the first time presenting to a classroom, we learned some helpful tricks along the way. We learned the benefit of bringing backup images because some students did not image their microSD cards properly. Students then worked together in pairs if they had a bad image or a different type of computer that a Macintosh or Windows computer. The same computer problem occurred in HD, when students did not have the proper computer, but they simply used their partner's computer instead. In HD we did the entire session, including watching the videos as well as working on the Raspberry Pis, in 25 minutes. We deemed this as a huge measure of success.

Future Work

We plan to continue this ongoing project by producing videos to setup a cluster of Pis. A cluster is a group of networked computers that work all together as one to solve bigger problems. Our system image comes preloaded with configurations and packages required for a Head node. A Head node manages the rest of the computers on the network. The new version of our system image, currently being tested, has custom software to modify

the configurations to make it into a worker node. Initially we planned to have 3 system images, one system image for using the Pi regularly, a second for connecting remotely, and a Head node system image for making clusters. We also plan to make new modules that utilize MPI(Message Passing Interface). Our next teaching videos will include step-by-step instructions for teaching parallel computing using CSinParallel materials. After that, we will focus on making videos for other platforms such as Linux.

Conclusion

We have presented a custom system image for the Raspberry Pi 3 that makes it convenient for students to connect to and compute with a Pi using a Macintosh or Windows laptop, as well as a series of teaching videos that describe how to install that system image on a Pi, and how to connect a laptop to a Pi running that custom image. We have also reported on classroom testing of these materials in two courses. In a core-level course in parallel and distributed computing, students followed the system setup videos as homework in order to install the custom system image on a Pi, then proceeded to carry out an in-class exercise of connecting their laptops (without videos) and carrying out a parallel computation. In a lower-level computer organization course, students used videos to connect their laptops to Pis running the (provided) custom system image, then replicated a parallel computation exercise they had previously performed on a shared multicore server. Assessment results indicate that students agree or strongly agree with statements that they would like to learn more computer science using the Raspberry Pi platform and using videos, that the videos helped them to feel competent about the hardware tasks they undertook, and that they would like to learn more about the Raspberry Pi. Since these in-class exercises required only 25-50 minutes of class time each, we believe that teaching parallel computing concepts using Raspberry Pis with our supports makes it feasible to teaching PDC concepts through short, in-class interventions that are effective, convenient, and appealing to students.

Acknowledgement

Stephen Akers '17 and Eric Oseid '17 helped with the development of the Raspberry Pi 3 system image that supports convenient laptop connection.

Prior work for this project was funded by the Minnesota LSAMP and CSinParallel, which are both NSF-funded projects, and by St. Olaf College.

References

- [Gibbs 2015] Gibbs, Samuel (2015-02-18). *Raspberry Pi becomes best selling British computer*. The Guardian. Retrieved 2016-12-28.
- [Kawash 2016] Jalal Kawash, Andrew Kuipers, Leonard Manzara, and Robert Collier. 2016. Undergraduate Assembly Language Instruction Sweetened with the Raspberry Pi. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16). ACM, New York, NY, USA, 498-503.
- [Sadaghiani 2012] Sadaghiani, H. R. (2012, April). Online Prelectures: An Alternative to Textbook Reading Assignments: The Physics Teacher: Vol 50, No 5. Retrieved March 17, 2017, from <http://aapt.scitation.org/doi/full/10.1119/1.3703549>
- [Stelzer 2009] Stelzer, T., Gladding, G., Mestre, J. P., & Brookes, D. T. (2009, January). Comparing the efficacy of multimedia modules with traditional textbooks for learning introductory physics content: American Journal of Physics: Vol 77, No 2. Retrieved March 17, 2017, from <http://aapt.scitation.org/doi/figure/10.1119/1.3028204>
- [Raspoberry Pi 2016] Ten millionth Raspberry Pi, and a new kit - Raspberry Pi. *Raspberry Pi*. 2016-09-08. Retrieved 2017-02-01.