the orange pi:

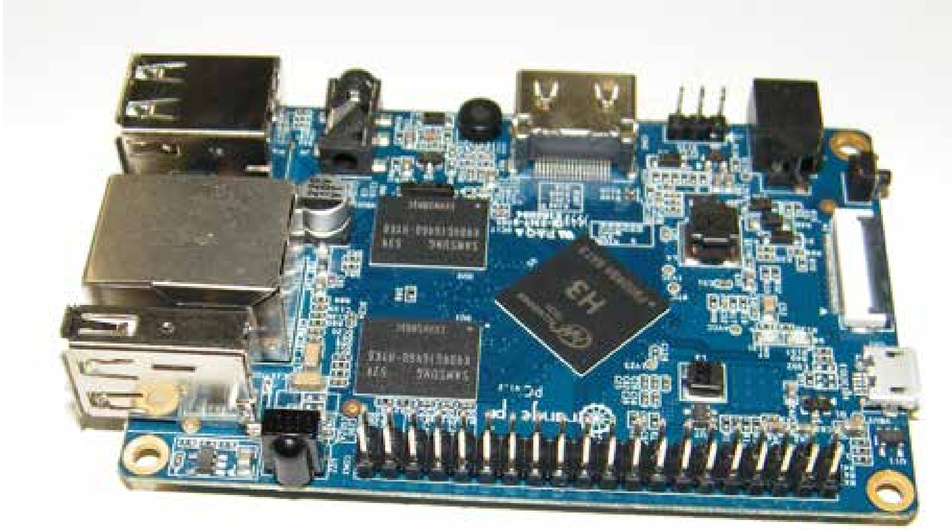
integrating programming through electronic technology

The Orange Pi is an inexpensive resource to teach an array of electronics and programing concepts from an engineering design focus.

by
Tyler S. Love,
Joel Tomlinson,
and Derrek
Dunn

Introduction

Electronic technology plays an important role in developing technological literacy as evidenced by its significance throughout Standard 16 of *Standards for Technological Literacy* (*STL*) (ITEA/ITEEA, 2000/2002/2007) and its designation as one of the nine core technologies (Smith & Gray, 2010). Electronics lessons provide an excellent opportunity for integrating science, technology, engineering, and mathematics (STEM) concepts while addressing a number of the *STL* standards, *Next Generation Science Standards* (*NGSS*), and *Common Core State Standards* (*CCSS*). Traditionally, electronics lessons have been taught via modules, electronic circuit stations, or circuit kits. Technological advances have allowed technology and engineering (T&E) educators to further engage students by teaching design-based STEM lessons in the context of the latest electronic technologies such as mobile phones, robotics, and drones.



Orange Pi.

In the state of Maryland, electronic technology plays an increasingly important role due to recent changes regarding what courses count for T&E education credit. Since 1993 Maryland has been one of the only states requiring students to complete a T&E education course to graduate high school. State regulations, otherwise known as the Code of Maryland (COMAR), specify that T&E education courses shall help students demonstrate an understanding of the nature of technology, the engineering design process, the core technologies, and the designed world. In 2016 the Maryland State Department of Education (MSDE) broadened its list of approved T&E education courses that could fulfill this graduation requirement to include computer science classes (MSDE, 2016). This has allowed school systems to enroll students in computer science classes in lieu of technological-literacy-focused T&E education courses. Additionally, this has broadened the breadth of content to be covered in pre- and in-service training to prepare T&E educators to teach computer science concepts. Engineering design challenges utilizing microcomputers, such as those presented in this article, can help school systems teach programming from a technological literacy lens rather than a computer-literacy focus. They also address an array of standards while requiring students to apply their STEM skills to solve engaging engineering design problems.

Comparing Raspberries, Bananas, and Oranges

The Raspberry Pi, Banana Pi, and Orange Pi are examples of microcomputers, an electronic device with a microprocessor as its central processing unit (CPU) (Encyclopaedia Britannica, 2016). The Raspberry Pi has a rich history and was the first of these three microcomputers commercially available (Gawande & Deschmukh, 2015). Since their release they have grown in popularity and serve as excellent resources for teaching technological and engineering design applications. One example of an authentic application is the FarmBot project, which utilized the Raspberry Pi 3 to yield more efficient crop growth using technology that operates similarly to that of a 3D printer (FarmBot Inc., 2016). Additionally, the Raspberry Pi has been used for various robotic design challenges (RaspberryPi.org, 2016) and remotely operated underwater (ROV) vehicle research (Rocha et al., 2014).

The Raspberry, Orange, and Banana Pi each feature a unique set of onboard capabilities. The Raspberry Pi 3 is the most recent to hit the market and comes from a successful line of predecessors. The Orange Pi PC is a second-generation product that has gained popularity due to its affordability and range of operating systems that can be installed. The Banana Pi is a first generation microcomputer and was developed to be an alternative option for consumers.

The Raspberry Pi 3 is a very proficient microcomputer that is the result of design improvements from previous pi generations. It offers new hardware options such as a wireless chip, antenna, and higher-level graphics card. Unlike previous generations, all of these additional features have to be purchased and added externally via USB. Since the wireless chip on the Raspberry Pi 3 offers a 2.4 GHz 802.11n wireless LAN, it has built-in Bluetooth, FM radio, and wireless internet capabilities. The CPU on the Raspberry Pi 3 operates with a 1.2 GHz quad core processor, which is impressive considering its size, price, and capabilities. The Raspberry Pi 3 runs a Linux-based operating system, but has an array of features to help first time pi users become acclimated with this system.

The Orange Pi PC has some competitive capabilities at a low cost. This generation of the Orange Pi does not have an integrated wireless chip, but it does possess a quad core CPU, 1.6GHz of processing power, and is capable of handling 1080P video content (Xunlong Software Co., Limited, 2016). Although the Orange Pi PC does not offer wireless internet connection, it does have a standard ethernet port to support wired connection. This Orange Pi can run a variety of operating systems such as Android, Linux-based systems, and the Raspberry Pi image.

The Banana Pi is a notable competitor of the two previously stated microcomputers. It offers a CPU with a 1GHz dual core processor (Banana Pi, 2014), which is excellent considering its small size and weight of only 48 grams. The Banana Pi has two sets of 26 GPIO (General Purpose Input/Output) pins that provide the capability to customize with user-friendly external hardware like a micro LCD screen. Although the Banana Pi does not have some of the same hardware features as the Raspberry Pi 3, nor does it offer the versatility of the Orange Pi PC, it is still a feasible option for educators seeking to conduct engineering design-based projects with a Linux based operating system.

Getting Started

When selecting a microcomputer for engineering design challenges, there are a few things to consider before purchasing. Cost is a big concern for educators with limited resources. The prices stated in Table 1 only account for the cost of the unit itself. In addition, to use the unit one must buy the properly rated power supply, video output such as a HDMI cable, USB mouse and keyboard to communicate with the computer, and at a minimum an eight gigabyte memory card to install the operating system and additional programs. Each manufacturer has a variety of affordable starter packages that can be purchased with all of the necessary start-up items and accessories. The University of Maryland Eastern Shore (UMES) Raspberry and Orange Pi webpage (UMES, 2016) provides recommendations for starter packages and vendors.

After purchasing the pi unit, accessories, and components, there are a few additional factors to consider before teaching a pi engineering design lesson. The majority of projects will require a Linux-based program to operate the pi. There are many online communities with free tutorials and step-by-step instructions for installing an operating system, programming python (the programming language for Linux based operating systems), and educational activity ideas. Examples of these online communities are also provided on the UMES (2016) webpage. It is important to remember that these microcomputers were developed for educational purposes, and the manufacturers have developed resources to assist individuals with limited computer programming experience.

Classroom Application

In this design challenge the authors opted to use the Orange Pi PC for its low cost yet broad range of capabilities. Students were tasked with designing a home security system that used the Orange Pi and a motion sensor to control the power to an LED. This interested students because most were familiar with motion sensor floodlights, and they valued the concept of home security/safety. To prepare for the design challenge the authors had to prepare all necessary cabling for video output and power, the mouse and keyboards, and memory cards for the operating systems. In addition, the following accessories had to be purchased: breadboards, GPIO pin adapters, jumper wires, and electronic components.

The Orange Pi PC requires a number of accessories and peripheral devices. The first is a 2 amp micro USB charger to supply adequate power for operating the pi and peripherals such as a keyboard and mouse. HDMI cabling is recommended for video output to sources such as a computer monitor, television with HDMI input, or projector with an available HDMI port. Peripherals are critical for the initial install and allowing the user to communicate with the Orange Pi PC after the initial installation is complete. A memory card with a minimum of eight gigabytes of memory is needed for storing the operating system, additional software, and programming code. Additional accessories such as an adapter to convert the GPIO pins to an output for a bread-

Table 1

Comparison of the Raspberry, Orange, and Banana Pi

	Raspberry Pi 3	**Orange Pi PC**	**Banana Pi**
Cost	$36.00	$21.99	$35.00
CPU	ARM Cortex-A53 Quad-Core 1.2 GHZ	H3 Quad-Core Cortex-A7, 1.2 GHZ	ARM Cortex A7 Dual-Core 1GHz
RAM	1GB	1GB	1GB
Storage	Up to 64 GB	Up to 64 GB	Up to 64 GB
Storage Type	Micro SD Card	Micro SD Card	SD Card
Onboard Network	Wireless 802.11n and Ethernet	Ethernet	Ethernet
Video Output	HDMI and RCA	HDMI and Integrated CVBS	HDMI and Composite
Audio Input	Additional hardware via USB	Built In Mic	Built In Mic
Audio Output	3.5mm jack and HDMI	3.5mm jack and HDMI	3.5mm jack and HDMI
USB Ports	4 USB 2.0	3 USB 2.0	2 USB 2.0, 1 Micro USB
Power Supply	5.0 V, 2.5 A	5.0 V, 2.0 A	5.0V, 2.0A
GPIO Pins	40 pin	40 pin	Two sets of 26 pin

board can be purchased for easier configuration. Electronic components recommended for purchase include: a 330 Ohm resistor, LED, and motion detection sensor (see Table 2 for additional sensors that can be purchased).

After preparing for this design challenge, the authors began teaching students about basic python programming language via online tutorials and mini lessons (UMES, 2016). It is important for students to understand that Linux-based operating systems are different than Windows and OS X operating systems. One of the benefits of a Linux-based system is that it is open-source software offering a multitude of readily available freeware. Open-source software allows users to modify programs, code, and operating systems to the user's specified preferences. Viewing tutorials and examples of how to insert, modify, and run programming code for different design projects helped familiarize students with python programming basics. The pi communities listed on the UMES (2016) webpage provide a vast amount of free instructional resources including step-by-step tutorials, videos, projects, blogs, and lesson plans.

After foundational instruction about python programming and operating system installation, students were tasked with installing a basic program on their Orange Pi PC. They worked to input the program using Idle, an application allowing the user to check the code before outputting the program to the GPIO pins. Idle is important because it can identify input code errors and detect if the GPIO pin location is correct. After checking the code with Idle, students assembled the circuit on a breadboard using the schematic provided by the instructor (Figure 1). A 40 pin adapter

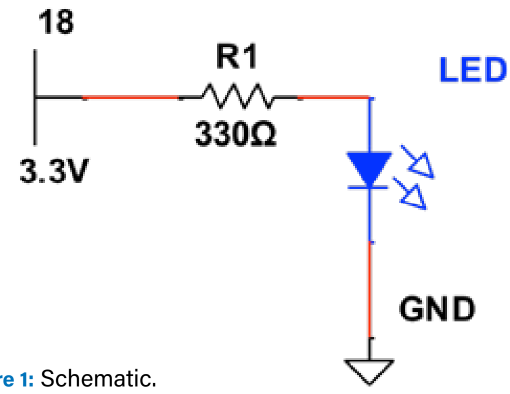


Figure 1: Schematic.

was used to connect the breadboard to the GPIO pins on the Orange Pi PC. Lastly, students verified the electronic component layout to ensure the circuit was wired correctly to the breadboard's GPIO pins adapter. This preliminary verification process was critical for checking students' work and providing more efficient instruction.

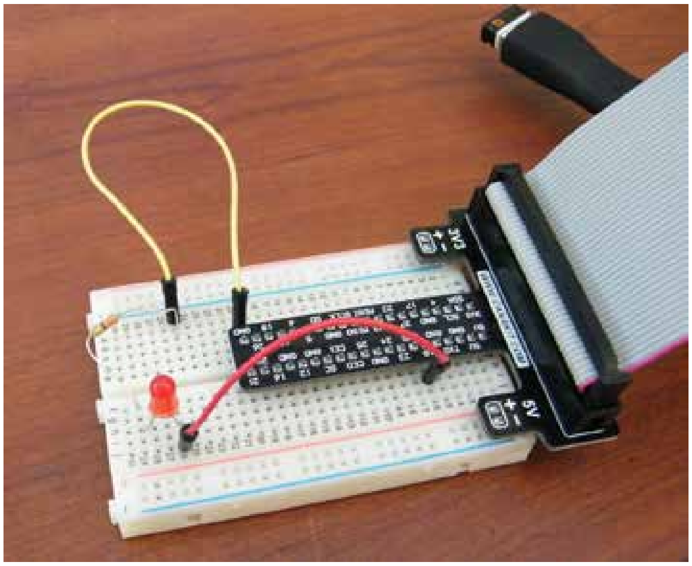
Once the correct code and schematic verification were completed, the students were ready to execute their code. When executing a code in Idle, the application exports the code to another application called the Shell. Before executed, the code remains in Idle and is similar to most Notepad or text editor applications in Windows operating systems. After the code is exported to the Shell, it verifies and allows it to run, and outputs the voltage to the GPIO pins defined in the program. Students tested their designs by placing an object or their hand within close proximity of the motion detection sensor. Their design solutions had to

Table 2

Pi Accessories and Extension Activities

Accessory	Cost	Potential Design Challenge/Applications
5 megapixel camera board	$20.00	Motion-activated surveillance system, baby monitor, vehicle back up camera, trail camera
Accelerometer	$15.00	Control the maximum speed of a robotic vehicle
Ball tilt sensor	$2.00	Safety feature to shut off power to a device that tips over, rotate cell phone screen orientation
Force sensitive resistor (FSR)	$7.00	LED dimmer upon touch, design a musical keyboard
Gas sensor	$15.00	CO_2 detector used to detect Carbon Dioxide in the air
Hall effect sensor	$2.00	Open and closed door/window magnetic sensor
Infrared (IR) sensor	$2.00	Remote controls for televisions and radios, point-to-point wireless communication systems
Motion sensor	$10.00	Security system, automatic door opener
Piezo buzzer	$1.50	Sound for various security system alarms
Photo cell sensor	$1.00	Break beam alarm with a laser pointer, dusk to dawn light.
Temperature sensor	$1.50	Power a fan based on temperature, biofeedback (body temperature) system, safe coffee/tea drinking temperature sensing coaster

Note: Additional sensors, product information, and design challenges/applications can be found at www.adafruit.com.



Breadboard.

activate an LED when an object passed close to the sensor. Also, their code had to allow the LED to stay lit for 10 seconds after no further motion was detected.

STEM Integration

Numerous mathematics and science practices were integrated throughout this electronic technology design challenge. Ohm's law was used to verify the correct resistor selection for students' designs. The LED provided to students had a forward operating current of 10 mA. The instruction in the design challenge called for the use of GPIO pins that were able to output 3.3V. Using Ohm's law, students had to calculate the resistance needed to supply the proper current to the LED. Additionally, this design challenge helped students have a better understanding of electron flow and electricity. This is often a difficult science phenomenon for students to conceptualize, but writing a code and using sensors to control the current helped them see this concept in action.

Students also learned computer programming concepts from integrated engineering design and electronic technology applications. They were tasked with developing a system that used electronic circuit components and a microcomputer to communicate with the user via external sensors. Students created their programs from examples and tutorials and then executed programming commands using a Linux-based operating system. Each student developed a basic program to control an LED, which served as a foundation for developing more advanced motion-detection programs.

Teaching the aforementioned concepts to develop design solutions helped address standards in the following content areas: (a) *CCSS* in math and English language arts, (b) *NGSS* in physical science and engineering, technology, and science applications, and (c) *STL* in the relationships among technologies, engineering design, energy and power technologies, and information and communication technologies. Other concepts and standards can be covered with the purchase of additional pi accessories.

Extension Activities

It is easy for teachers to create their own engineering design challenge scenarios by purchasing accessory sensors for their pi unit (Table 2). Accessories can be purchased individually or in kits at a relatively low cost. One example of designing an engineering challenge would be to task students with creating a model-sized smart house. There are a number of accessories students could use to design more efficient methods for controlling lights, fans, and other integral components of a house. Advanced design challenges like this should have a systems focus, requiring multiple accessories to work together. Another example of a systems-focused design challenge includes creating a security surveillance system that sounds a buzzer and saves a photo to a memory card when motion is detected.

Conclusion

There are a number electronic technology and computer science concepts that can be integrated using different resources; however, the authors highlighted the activities in this article for their low cost, minimal learning curve, ease of use, and potential to situate within an engineering design context. UMES uses the Orange Pi to teach pre- and in-service T&E educators how to integrate computer programming concepts through engineering design-based electronic technology lessons. School systems should consider purchasing inexpensive resources like the Orange Pi and providing training for their teachers to learn how to incorporate them into their curricula.

References

Banana Pi. (2014). *Banana pi.* Retrieved from www.bananapi.org/

Gawande, S. V. & Deschmukh, P. R. (2015). Raspberry pi technology. *International Journal of Advanced Research in Computer Science and Software Engineering, 5*(4), 37-40. Retrieved from www.ijarcsse.com/docs/papers/Special_Issue/ITSD2015/9.pdf

International Technology Education Association (ITEA/ITEEA). (2000/2002/2007). *Standards for technological literacy: Content for the study of technology.* Reston, VA: Author.

Encyclopaedia Britannica. (2016). Microcomputer. In *Encyclopaedia Britannica.* Retrieved from www.britannica.com/technology/microcomputer

FarmBot Inc. (2016). *FarmBot.* Retrieved from https://farmbot.io/

Maryland State Department of Education (MSDE). (2016). *Maryland technology education standards: Grades 6-12*. Baltimore, MD: Division of Career and College Readiness. Retrieved from http://marylandpublicschools.org/programs/Documents/CTE/MDTechnologyEducationStandards.pdf

RaspberryPi.org. (2016). *Raspberry pi robots blog.* Retrieved from www.raspberrypi.org/blog/tag/robots/

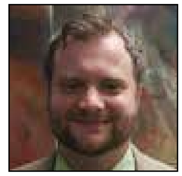
Rocha, C. R., Branco, R. M., da Cruz, L. A., Scholl, M. V., Cezar, M. M., & Bicca, F. D. (2014). *Design aspects of an open platform for underwater robotics experimental research.* In L. J. De Vin & J. Solis (Eds.), The Proceedings of the 14th Mechatronics Forum International Conference. Karlstad, Sweden (pp. 318-325).

Smith, W. H., & Gray, R. C. (2010). *The core technologies: Why are they important to technology educators?* Retrieved from http://bmi.servcent.com/attachment_file/file/139/Core_Technology_Resources.pdf

University of Maryland Eastern Shore (UMES). (2016). *The raspberry and orange pi: Recommended resources.* Retrieved from www.umes.edu/Tech/raspberrypi.html

Xunlong Software Co., Limited. (2016). *Orange pi.* Retrieved from www.orangepi.org/

Tyler S. Love, Ph.D. *is Coordinator and an assistant professor of Technology and Engineering Education at the University of Maryland Eastern Shore. He can be reached at tslove@umes.edu.*



Joel Tomlinson *is a lecturer in Electrical Engineering Technology at the University of Maryland Eastern Shore. He can be reached at jmtomlinson@umes.edu.*

Derrek Dunn, Ph.D. *is a professor and Chairperson of the Department of Technology at the University of Maryland Eastern Shore. He can be reached at ddunn@umes.edu.*