6-22-2020

Infusing Raspberry Pi in the Computer Science Curriculum for Enhanced Learning

Fitratullah Khan
The University of Texas Rio Grande Valley

Mahmoud K. Quweider
The University of Texas Rio Grande Valley

Ala Qubbaj
The University of Texas Rio Grande Valley

Emmett Tomai
The University of Texas Rio Grande Valley, emmett.tomai@utrgv.edu

Lei Xu
lei.xu@utrgv.edu

See next page for additional authors

Recommended Citation

Khan, F., & Quweider, M. K., & Qubbaj, A., & Tomai, E., & Xu, L., & Zhang, L., & Lei, H. (2020, June), Infusing Raspberry Pi in the Computer Science Curriculum for Enhanced Learning Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual On line . https://doi.org/10.18260/1-2--34828

Authors

Fitratullah Khan, Mahmoud K. Quweider, Ala Qubbaj, Emmett Tomai, Lei Xu, Liyu Zhang, and Hansheng Lei

Infusing Raspberry Pi in Computer Science Curriculum for Enhanced Learning

Dr. Fitratullah Khan, The University of Texas Rio Grande Valley

Professor Fitratullah Khan has been teaching computer science courses since 1992. His areas of expertise are computer architecture, networking, database systems, computing platforms and languages. As the director of Infrastructure, Telecommunications, and Networking (ITNet), and later as a Chief Technology Officer, at UT Brownsville, he implemented state of the art networking using campus wide fiber ring with redundant links. He established diskless computer labs to provide uniform computing platform across campus, and modernized classrooms to make them congenial to online learning. He was the PI on NSF funded BCEIL (Beowulf-based Curriculum Enrichment Integrated Laboratory) and Co-PI on NSF funded MCALL (Multimedia based Computer Assisted Learning Lab).

Dr. Mahmoud K Quweider, The University of Texas Rio Grande Valley

M K Quweider is a Professor of Computer & Information Sciences at the U. of Texas at UTRGV. He received his Ph.D. in Engineering Science (Multimedia and Imaging Specialty) and B.S. In Electrical Engineering, M.S. in Applied Mathematics, M.S. in Engineering Science, and M.S. in Biomedical Engineering all from the University of Toledo, Ohio. He also holds a Bachelor of English and a Masters of Business Administration from the University of Texas at Brownsville. After graduation, he was employed at several corporations including Pixera, a digital multimedia processing company in Cupertino, CA, 3COM, a networking and communication company in Schaumberg, IL, and Mercantec, an E-Commerce company in Naperville, IL. He has more than 40 publications in the field, and has served as a reviewer/moderator for several scientific and educational journals and conferences. He joined UTB in the Spring of 2000. His areas of interest include Imaging, Visualization and Animation, Networking and Cyber Security, Web Design, Computer Graphics, and Linguistics.

Dr. Ala Qubbaj, The University of Texas Rio Grande Valley

Ala Qubbaj, Ph.D. Dean of the College of Engineering & Computer Science The University of Texas Rio Grande Valley Dr. Ala Qubbaj is the Dean for the College of Engineering and Computer Science at the University of Texas Rio Grande Valley (UTRGV) and the Principle Investigator (PI) for the UTRGV's ADVANCE Institutional Transformation grant funded by the National Science Foundation (NSF); which focuses on increasing the representation and advancement of women in STEM fields. He is also the PI on an NSF INCLUDES (Inclusion across the Nation of Communities of Learners of Underrepresented Discoverers in Engineering and Science) project/Symposium for ADVANCING STEM Latinas in Academic Careers.

Prior to his Dean position, Dr. Qubbaj served as Senior Associate Vice President/Vice Provost for Faculty Affairs & Diversity at UTRGV. He is also a full professor in the department of Mechanical Engineering. Dr. Qubbaj received his Ph.D. from the University of Oklahoma with specialization in combustion and energy system. His research has been sponsored by NSF, the Department of Energy, and the Department of Defense.

Dr. Emmett Tomai, The University of Texas Rio Grande Valley

Emmett Tomai is an Associate Professor and Chair of the Department of Computer Science at the University of Texas Rio Grande Valley. He received his Ph.D. in Computer Science from Northwestern University in 2009. Before that he received his M.S. in Computer Science (2007), B.S. in Computer Engineering (1997), and B.S. in Electrical Engineering (1997) from Northwestern University. His research interests are in artificial intelligence, machine learning, natural language understanding, and video game development, with emphasis on learning models of decision-making to support commonsense behavior and storytelling. Dr. Tomai also pursues research in computer science education with emphasis on introductory problem-solving, critical thinking, and programming skills.

Lei Xu, University of Texas Rio Grande Valley

Lei Xu received his Ph.D. degrees in 2011 from the Chinese Academy of Sciences. He is an assistant professor in the Department of Computer Science, University of Texas Rio Grande Valley. His research interests include applied cryptography, cloud/mobile security, and decentralized systems.

Dr. Liyu Zhang, University of Texas Rio Grande Valley

Liyu Zhang is an Associate Professor in the Department of Computer Science Department of Computer Science at the University of Texas Rio Grande Valley. He received his Ph. D. in Computer Science from the State University of New York at Buffalo in September 2007. Before that he received his M. S. (2000) and B. S. (1997) from Fudan University, Shanghai , China, both in Computer Science. His research interests are in theoretical computer science and its applications, with emphasis on computational complexity, complexity-based cryptography, and design and analysis of algorithms. Dr. Zhang's research has been sponsored by National Science Foundation and U.S. Department of Education.

Dr. Hansheng Lei

Infusing Raspberry Pi in Computer Science Curriculum for Enhanced Learning

Abstract— With the advent of cloud computing, the Internet of Things (IoT), and mobile computing, CS faculty are continuously revamping the curriculum material to address such burgeoning set of technologies in practical and relatable ways. Raspberry Pi (RPi) devices represent an ideal hardware/software framework that embodies all these technologies through its simple architecture, small form factor (that minimizes the volume and footprint of a desktop computer), and ability to integrate various sensors and actuators that network together and connect to the Cloud. Therefore, one of the strategies of Computer Science Department, to enhance depth of learning concepts, has been to infuse Raspberry Pi (RPi) in computer science courses. RPi has been incorporated since 2017 in targeted courses, notably, Computer Organization & Assembly Language, Computer Architecture, Database Management Design & Implementation, Unix/Linux Programming, Internet Programming, and Senior Project. An inexpensive credit card sized computer, an RPi lends itself to allow depth of learning of concepts. From implementing firewalls, intrusion detection systems, scripting, client-server based computing, distributed computing, to interfacing with sensors and actuators, a student is guided to polish concepts taught in a class through RPi Project Based Learning (RPBL). Computer science curriculum already provides breadth of learning. The infusion of RPi in key courses provides depth in targeted concepts. There are peripheral desirable consequences as well, including a student learning prevalently used Linux environment even though a targeted course may have nothing directly to do with Linux. Furthermore, RPi provides an opportunity for students to realize that software programs can be interfaced with sensors and actuators to provide immersed experience in programming. From simply interfacing a switch and a Light Emitting Diode (LED) to getting data from sensors, buffering, and uploading to the cloud, a student already would have touched upon multiple disciplines in computer science. This paper provides a blueprint to infusing RPi in the targeted courses, and how each RPi based project provides depth to a targeted concept.

Index Terms—Raspberry Pi, Cloud Computing, Internet of Things (IoT), Mobile Computing, Linux.

I. Introduction, Background and Goal

The undergraduate computer science curriculum at The University of Texas Rio Grande Valley (UTRGV) was transitioned from its two legacy institutions, The University of Texas at Brownsville and The University of Texas Pan American, accounting for more than two decades of assessment and curriculum hardening. The computer science program is accredited by the Accreditation Board for Engineering and Technology (ABET) and, has the breadth covering important topics in computer science. However, some of the faculty, with electrical engineering background, felt that the breadth of the program could be complemented with Project Based Learning (PBL) in selected courses to provide depth in key concepts as well. In 2017, faculty started developing projects around Raspberry Pi (RPi) in key targeted courses and, assigned the developed projects as tutorials and assignments in the courses.

It should be emphasized here that infusion of RPi in the computer science curriculum is not being done to increase retention, and no related assessment is being done to show the efficacy of using Raspberry Pi Project Based Learning (RPBL) on retention. There is existing literature on the merits of using RPi platform [14,18].

In a nutshell, the purpose of RPBL is to enable students to study key concepts deeper so that retention of concepts can be increased as the students move up in the curriculum.

As will be addressed in Section IX, there have been favorable consequences of infusing RPi into the curriculum of computer science, such as spin off of a newly proposed cybersecurity program, however, the goal of infusing RPi into the curriculum of computer science is to strengthen key computer science concepts.

II. Traditional Approach for Enhancing Concepts

Traditionally, a curriculum has labs, mostly as separate 1-hour courses, to expose students to important concepts in depth. As an example, UTRGV computer science curriculum has *CSCI 1170 Engineering Computer Science I Laboratory* as a separate course to complement *CSCI 1370 Engineering Computer Science I*. The two courses are usually taught by different instructors, and essentially are two separate courses with no interactivity between the two. This approach becomes expensive in terms of the total number of hours required to complete a degree if one complements each course with its lab course.

Another traditional approach is to increase the number of hours of certain selected courses to 4 Semester Credit Hours (SCH). As an example, *PHYS 1401 General Physics I* at UTRGV is a 4-SCH course but has a separate 3-SCH lecture section and a separate 1-SCH lab section. Again, this approach quickly accumulates credit hours if one converts key courses in a curriculum to 4-SCH courses for including labs to enhance learning.

III. The Need for Lab Augmented Courses

The traditional approach of complementing courses with their corresponding lab courses has been practiced for decades. The purpose of such lab courses is to make sure that the students learn key concepts which are regarded to be of paramount value to the field of study. For example, it is common to have waveguide labs for complementing lecture courses in electromagnetic fields, or wind tunnel labs for augmenting fluid dynamics courses. So, the authors understand the value of lab courses, but at the same time have looked at alternative approaches to benefit students in more courses rather than depending on the available labs limited to one or two courses in the curriculum.

It is important to note that lab courses in a curriculum are meant as a supplemental instruction to make sure that key concepts are engrained into a student's understating of the subject matter. Expanding on the earlier example of *CSCI 1370 Engineering Computer Science I* courses and its complementary lab course *CSCI 1170 Engineering Computer Science I Laboratory*, what could be so different in the assignments of the two courses that necessitates two separate courses? There is no special instrumentation, such as a wind tunnel, lasers or waveguides, involved in the two courses that one needs to have separate experts teaching the two courses. Both courses use computers and C++ programming language to teach programming to students. Why aren’t the assignments and projects in *CSCI 1370* adequate so that the students don't have to take the complementary lab course? Apparently, the only justification to having a separate lab course in programming is to make students go through a wholesome project in each class.

IV. A New Computing Platform

All the courses in the core curriculum of computer science have one instrument in common: a familiar desktop computer in a computer lab or a student-owned laptop or tablet. A desktop computer is mostly good for pure computing since it is extremely difficulty to do interactive projects involving real world devices. The computing environment is made worse due to Information Security (IS) being involved in limiting what a student or an instructor may do on an institution's lab computer. Even interfacing a simple joystick to a computer would be a tall order given the security imposed on the lab computers.

Of course, exposing students to computing for modern systems such as embedded systems, vision systems, mobile systems, and cybersecurity is almost impossible given the secured lab computers as a computing platform. Furthermore, it does not pique students' interest in computing if projects are limited to pure computing, such as computing the average GPA of a class. That is why there are special programming environments, such as the legendary Karel The Robot [1], built to make programming interesting.

There are several major goals that the authors needed to achieve with a new computing platform, based on a flexible *computing node* (a computing device), to provide students with depth of learning of key concepts. These are summarized as follows:

1) A *computing node* in the new computing platform must easily interface to myriad of sensors and actuators.
2) A student must be responsible for security of his/her *computing node*, and not the Information Security department of the university. This is necessary to keep the computing node flexible, mobile and easily accessible.
3) A student must be able to use a *computing node* with other computing nodes to get exposed to networking and distributed computing learning.
4) A *computing node* must be able to accept different operating systems for students to learn different platforms.
5) A *computing node* must provide access to modern programming languages and packages.
6) A *computing node* must allow connectivity to cloud platforms.
7) A *computing node* must have sufficient computing horsepower and memory to allow students to do meaningful projects.
8) Lastly, a student should easily be able to afford a *computing node*. It is important for the student to purchase and own such a node, because IS of the university does not impose the same strict security profile on student-owned devices as it does on institution's computers.

There were several solutions available at the time, circa 2017, which included Arduino and Raspberry Pi. The former is more of a microcontroller while the latter, being a microcomputer, provides a true desktop computing environment. At the time of rolling out the project, a Raspberry Pi had the following features which satisfied the constraints on the computing platform that the authors were looking for:

- RISC ARM v7 quad core processor with 1GB of RAM
- Camera port for vision related projects

- 40-pin GPIO interface to connect sensors and actuators
- Easily swappable SD Card to switch between different operating systems
- Credit card size computer for mobile and embedded system projects
- Configurable in a small tower for distributed programming projects
- Wireless network for mobility
- Under US $35

Therefore, Raspberry Pi was selected as a *computing node* for the new computing platform to be used initially in key courses of computer science for providing in-depth learning of key concepts.

V. Key Targeted Computer Science Courses

There are six courses in the undergraduate curriculum of computer science targeted for infusing RPi since 2017:

1) *CSCI 2333 Computer Organization & Assembly Language*
2) *CSCI 2344 Unix/Linux Programming*
3) *CSCI 3342 Internet Programming*
4) *CSCI 4333 Database Management Design & Implementation*
5) *CSCI 4335 Computer Architecture*
6) *CSCI 4390 Senior Project*

Every targeted course uses RPi in several projects to hone skills involving important concepts, exception being *CSCI 4390 Senior project* where a student, choosing to do his/her senior project using RPi, completes a unique semester long project.

To emphasize further, not every student in *CSCI 4390 Senior project* is required to do an RPi based project. Whereas in the other five targeted courses, students need to purchase an RPi and use it in several projects given in each course throughout the semester. Some of the projects involve multiple RPi nodes to demonstrate multithreading client/server concepts. In other cases, students are required to leave their RPi at home while learning to allow access for the instructor from campus to check applications developed by the students.

VI. Key Concepts Targeted in Each Course

Key concepts in the targeted courses were identified over the years based on what worked in different courses. Tutorials were developed every time the assigned faculty taught a targeted course. More than twenty tutorials have been developed for the first five courses whereas a student taking the sixth course, *CSCI 4390 Senior Project*, has access to any of the tutorials if working on an RPi based semester long senior project. The tutorials are improved upon every semester based on the availability of sensors, a new concept being perceived as important, availability of improved RPi, or even to accommodate changes in operating system features. For example, recently RPi 4 is available with 4 GB RAM which has facilitated taking on computationally-intensive machine vision and cybersecurity projects.

Table I shows the key concepts targeted in each course. As mentioned earlier CSCI 4390 is an exception where students may choose to do an RPi based project, therefore, there are no established key concepts targeted for CSCI 4390. It should be noted that key concepts shown do not reflect all the topics covered in the course in which an RPi is used. Once RPi is part of a course, it is used by the student for most of the projects assigned in the course. Therefore, an RPi is used in a targeted course for many more topics than shown in the table.

TABLE I Key Concepts in Targeted Courses

Course #	Concepts through tutorials and projects
CSCI 2333	-RISC architecture -RISC assembly language -Interfacing sensors and actuators using assembly language
CSCI 2344	-Linux OS, CPU scheduling and process management -GREP, BASH scripts and CRON jobs -Multithreading client/server applications -Interfacing sensors and actuators using C -Firewall programming -Programming for real world devices
CSCI 3342	-Servers and browsers -Connecting RPi to the cloud -Interfacing cloud to sensors & actuators through RPi -Interfacing Android to sensors & actuators through RPi
CSCI 4333	-Embedded SQL -Interfacing to a backend database in the cloud -Collecting data from sensors and uploading to the cloud DB -Commanding actuators through RPi per cloud DB
CSCI 4335	-Detailed RISC architecture with multistage pipeline -Benchmarking -Effect of code arrangement on pipeline performance -Effect of code on cache performance
CSCI 4390	Different RPI-based capstone projects per student selection - some selected projects done by students are as follows: -Remote site real-time security and safety system -An embedded system based medical app for tracking health vitals and emergency issuer of notifications -An embedded system for real-time overhead clearance and collision detection for vehicles -An embedded system for real-time vehicle driving safety -Stress level detection applied to a joystick controller for various simulation systems -Network based real-time distributed parking lot management system -Embedded controller based smart self-sustaining greenhouse system -Embedded controller for multiple energy sources to minimize fossil fuel consumption

Furthermore, there are other courses being considered for adopting RPi as a computing platform. Courses in networking, cybersecurity, machine vision, operating systems, and digital forensics are some of the courses that will benefit from utilizing RPi, especially, now that RPi is available with 4GB of RAM providing a better computing platform. There are existing resources available to guide instructors to adopt RPi for variety of topics [4-13].

RPi has not been used in basic programming courses yet, such as CSCI 1370 and CSCI 1170. These are considered gateway courses, and the department is very cautious about using non-traditional computing platforms which could have enrollment consequences due to students not being mature enough to handle a device such as RPi with non-Windows based operating system. As a reference, there is existing literature regarding use of RPi in an introductory programming course [18].

VII. RPi Rollout and Usage

The process of infusing RPi into the undergraduate curriculum of computer science started in Spring 2017 with just one course, CSCI 4335, targeted. This was to make sure that the students are mature enough to handle a new computing platform, and at the same time that the instructors are getting used to idiosyncrasies of the new platform. From then on, RPi has been used in the targeted courses more than fifteen times or in more than 15 sections of the courses in different semesters. Figure 1 shows the RPi usage in the targeted courses since Spring 2017.

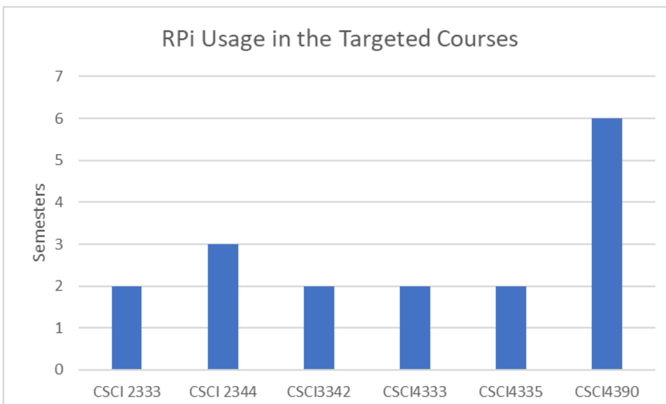

Fig. 1 RPi Usage in the Targeted Courses

The graph in Figure 1 shows consistent usage of RPi in *CSCI 4390 Senior Project course*, however, as mentioned earlier, not all the students are required to use RPi in their projects. The idea is to provide depth of key concepts to the students in targeted courses, and they can use RPi in their capstone project if they choose to do so. Figure 2 shows the head count in each targeted course to provide a comparison of RPi usage.

It should be noted that not all sections of the targeted courses use RPi. It depends on the instructor. If the instructor is well versed in RPi platform and is willing to use it to augment his/her course, the respective course section is RPi based. In other words, there is no departmental mandate to use RPi in every section of the targeted courses. Therefore, the head count shown does not reflect the computer science enrollment in the courses.

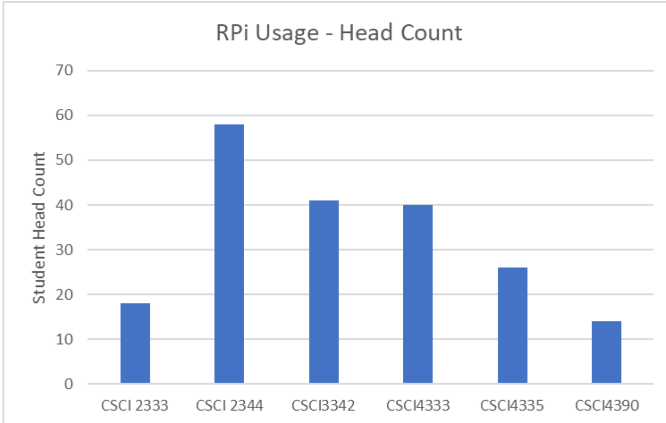

Fig. 2 RPi Usage in the Targeted Courses

VIII. Momentum and Limitations

There have been challenges in infusing RPi into different courses. It took two years to use RPi at least once in every targeted course. It takes significant amount of time to develop tutorials and vet them before using them in the targeted courses. This is irrespective of existing resources available for using RPi in an academic setting [15-17]. A useful approach found in the process of rolling out RPi, is to have a few Teaching Assistants (TAs) dedicated to test the developed tutorials, and to help improve upon the tutorials for next iteration. Of course, it is a bootstrap process where it takes a semester or two to get relatively knowledgeable TAs to be trusted with such a task.

Another important aspect is to have a group of faculty members vested in such a project. It does take significant amount of time to incorporate RPi in a course, where one could have just taught theory in class and assigned projects on traditional computing platform. Note that using RPi just for programming is not the idea here and, doing so will not benefit the students in learning deeper about the key concepts. Making meaningful tutorials to incorporate sensors and actuators to make projects interesting is what impacts the learning of key concepts the most. And this what takes time and, hinders instructors to incorporate RPi into the courses, which has been a major factor in gaining the desired momentum for rolling out RPi.

An RPi has its limitations for certain courses. For example, an RPi with 1GB of RAM is not adequate for certain class of real-time vision systems and cybersecurity applications. However, the new model of RPi offers 4GB of RAM easing the way for using RPi in additional courses.

IX. Usage in Other Programs

There have been favorable unintended consequences of infusing RPi into the undergraduate curriculum of computer science. The Department of Computer Science is in the process of proposing a baccalaureate degree in cybersecurity. RPi computing platform has been found ideal in most aspects of cybersecurity. It is well established that given the mobility of RPi with Kali

Linux on board [10], one can use the RPi platform for learning hacking aspects using ethical hacking. Especially, the new model of RPi, having 4GB of RAM, opens up new avenues to handle compute-intensive cybersecurity applications, such as parsing of voluminous logs in real-time to build an Intrusion Prevention System (IPS) or an Intrusion Detection System (IDS).

Recently, a seed funding of $15k was allocated which was used to purchase drones, robotic arms, mobile robots, RPi 4 units, and Internet of Things (IoT) equipment to establish a cybersecurity lab. Five Teaching Assistants have been dedicated to developing tutorials to be used in the cybersecurity courses of the proposed program. The Teaching Assistants are being guided by the instructors of the RPi based courses.

X. Results

There are six computer science courses targeted in the project to infuse RPi platform to strengthen the learning of key concepts. Table I tabulates the targeted courses and examples of key concepts covered in each one, *CSCI 4390 Senior Project* being an exception where a student chooses to do his/her capstone project in any area either using RPi platform or something else. This distinction is important to know to make sense of head count data in CSCI 4390 course.

The faculty started infusing RPi into different courses, via tutorials and projects, starting Spring 2017. Not all the six courses started using RPi at the same. Depending on the willingness of the instructor, the students of the targeted course are exposed to the RPi platform.

Figure 3 shows the student head count per Academic Year (AY) starting in 2017. However, the project started in Spring 2017, therefore, AY 2017 in the graph does not include Fall 2016. Also, the head count for AY 2020 does not include Summer 2020, when *CSCI 3342 Internet Programming* is scheduled to be taught using RPi platform.

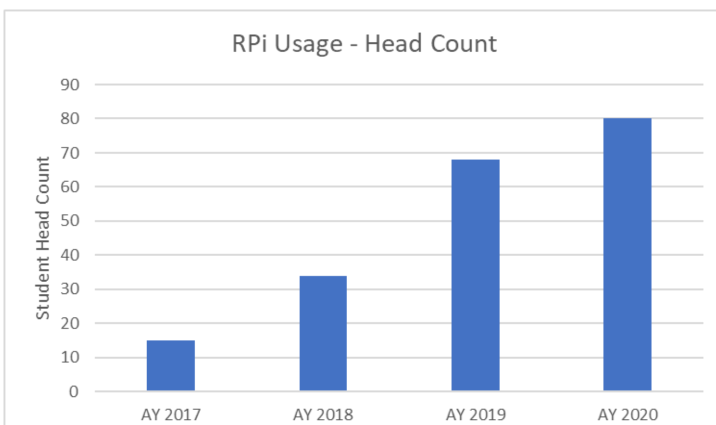

Fig. 3 Usage since Inception (AY 2020 excludes Summer 2020)

Besides learning key computer science concepts in different courses based on the RPi platform, students have been building sophisticated capstone projects involving software interacting with real world sensors and actuators. As an example, Figure 4 shows an RPi based capstone project titled *Network Based Real-Time Distributed Parking Lot Management System* [2]. The small size,

computing power, connectivity, mobility and affordability of RPi platform enables students to build such projects involving real world systems. Improvements in RPi platform, for example, availability of 4GB RAM in the newer model, has allowed instructors and students alike to use RPi in compute intensive applications such as AI, machine learning, deep learning, cybersecurity and vision systems.

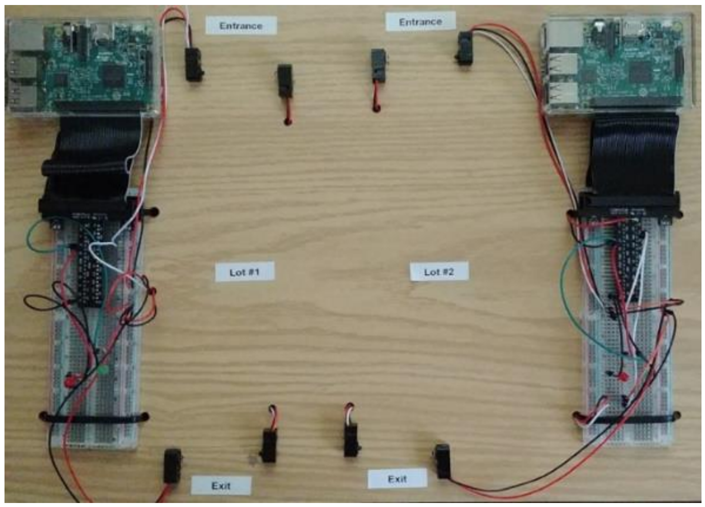

Fig. 4 Distributed Parking Lot Management System [2]

The credit card size footprint of RPi also lends itself conveniently for building distributed computing environments. Besides using such environments for solving challenging problems, one can use them to teach and learn both hardware and software aspects of distributed computing. Figure 5 shows a 5-node RPi tower built for teaching and learning distributed computing [3].

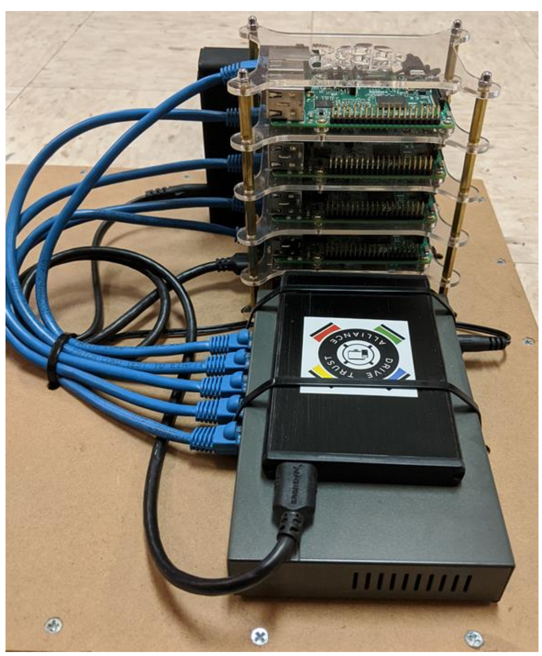

Fig. 5 Five-Node RPi Tower [3]

XI. Future Enhancements

The use of RPi in Senior Project has revealed that there are certain applications for which one needs a lot more GPU horsepower than is available in the RPi platform. This is true even in the newer model of RPi with 4GB of RAM. Even though the newer model, RPi 4, has a much faster GPU, it still poses challenges for certain real-time vision systems. Therefore, one must not limit oneself to RPi platform for every course rather one should consider other RPi-like platforms as well. For example, real-world vision systems have been built using NVIDIA Jetson Nano microcomputer which has 64-bit quad-core ARM processor with 4GB of RAM and a 128-core GPU, and still costs under US $100.

Such a platform is considered ideal for embedded AI applications involving deep learning, and for embedded real-time vision systems applications such as inspection of manufactured parts.

XII. Conclusion

The project for infusing RPi into computer science curriculum started in 2017 and, is ongoing and has been gaining ground in terms of student head count exposed to RPi platform. The impetus for using RPi was its alignment with the constraints (Section IV) that the authors had on a new computing platform that would enable students to go deeper into key concepts. The efficacy of this project is measured in terms of how much of certain topics can be covered in depth with the aid of RPi platform to strengthen key concepts. The project was not initiated with any intention to impact retention. Therefore, based on the ability to cover topics leading to strengthening key concepts has been achieved in the targeted courses. An organized effort to measure, assess and quantize the impact of using such a platform on learning of key concepts still needs to be headed by the faculty since it will involve a team effort to cover similar topics and measuring tools between the control group/section and experimental group/section each semester. However, at this time this is not of concern since just having the RPi platform is enabling instructors to cover key concepts in depth, which was the original intent.

Even though unintended, there have been favorable spin offs from infusing RPi into the curriculum of computer science. This includes establishing an RPi based cybersecurity lab and newly proposed cybersecurity degree program. Nevertheless, the RPi project is intended to remain a niche for the willing instructors to use in their courses to enhance learning of key concepts. The benefit of its use has been demonstrated in five key computer science courses, in addition to its optional use by the students in their capstone projects via *CSCI 4390 Senior Project* course.

Acknowledgment

The authors are grateful to the administration of The College of Engineering and Computer Science (CECS) and the Department of Computer Science at The University of Texas Rio Grande Valley (UTRGV) for funding and allowing the implementation of RPi in the undergraduate curriculum of computer science.

References

[1] Karel the Robot: A Gentle Introduction to The Art of Programming by Richard E. Pattis, Jim Roberts and Mark Stehlik, 1995.

[2] A. Sierra, Network Based Real-Time Distributed Parking Lot Management System, Senior Project, UTRGV, May 2018.

[3] K. Recio, Raspberry Pi Based 5-Node Distributed Computing Tower, UTRGV, Nov 2018.

[4] R. G. Plantz, Introduction to Computer Organization: ARM Assembly Language Using the Raspberry Pi, 2019.

[5] Raspberry Pi Learning Resources, raspberrypi.org/learning/teachers-guide/, Feb 2020.

[6] D. J. Norris, Machine Learning with Raspberry Pi: Experiments with Data and Computer Vision, 2019.

[7] D. J. Norris, Beginning Artificial Intelligence with the Raspberry Pi, 2017

[8] J. Cicolani, Beginning Robotics with Raspbeery Pi and Arduino: Using Python and OpenCV, 2018.

[9] M. McPhee, J. Beltrame, Penetration Testing with Raspberry Pi, 2016.

[10] D. Dieterle, Security Testing with Raspberry Pi, 2019.

[11] C. Hosmer, Defending IoT Infrastructures with the Raspberry Pi: Monitoring and Detecting Nefarious Behavior in Real Time, 2018.

[12] M. Rao, Internet of Things with Raspberry Pi 3: Leverage the power of Raspberry Pi 3 and JavaScript to build exciting IoT projects, 2018.

[13] J. Newmarch, Raspberry Pi GPU Audio Video Programming, 2016.

[14] J. Steinmeyer, Project-based Learning with Single-Board Computers, 2015 ASEE Annual Conference & Exposition, June 2015.

[15] V. Galluzzi, C. Berry, Y. Shibberu, A Multidisciplinary Pilot Course on the Internet of Things: Curriculum Development Using Lean Startup Principles, ASEE, 2017.

[16] N. Husseini, I. Kaszubski, Incorporating the Raspberry Pi into laboratory experiments in an introductory MATLAB course, ASEE, 2017.

[17] G. Mullett, Teaching the Internet of Things (IoT) Using Universally Available Raspberry Pi and Arduino Platforms, 2016 ASEE Annual Conference & Exposition, June 2016.

[18] S. Secules, W. Lawson, Description and Mixed Methods Evaluation of a Novel Hardware-Based Introductory Programming Course, ASEE, 2019.