Getting Ready for College: Experience of a Secondary Student in Engineering Research

Jaden Gavenda[1], Kumar Yelamarthi[2]
[1]Ashley Community Schools, Ashley, MI
[2]Central Michigan University, Mt Pleasant, MI

Abstract- Knowledge inspires curiosity. The pre-college summer scholars program is designed to provide research experience and knowledge to secondary school students on various engineering topics. After a system process of submitting an application, review, and interview by a committee, I was selected to take part in this research program during summer 2017. During this time, I worked in the exciting field of Internet of Things (IoT). When dealing with the IoT, I gained knowledge of microcontrollers, programming, sensors, actuators, cloud computing, and conducted several experiments. Further, Arduino, Raspberry Pi, and Python programming were among the most used keywords through the experiences in the program. The focus, however, was not just on the participants, but also on those who couldn't participate in the program. Many participants contracted an infectious obsession with their work on various levels, and this was just due to the knowledge they had gained in the research program. The point of the program though was to have the participants exposed to the capabilities of the Science, Technology, Engineering, and Math (STEM) field, and to advocate for a broader dissemination in secondary schools. Through this knowledge gained, which is to be discussed in the paper, the support of participants was earned and in hope will spread their knowledge as I am to inspire the needed curiosity in the STEM field.

1. Introduction

Science, technology, engineering, and math (STEM) fields in the United States have a deficit, employers need more employees in some of those fields.[7] In order to have engineers, computer scientists, and mechanics, students need to become educated in STEM. In my school district a STEM class is offered for which I have completed. The class I had revisited the engineering design process and did many different hands-on projects to exemplify it. Two of the projects included were a "do-it-yourself grill" and water rocket. For the "do-it-yourself-grill" a group repurposed a fifty gallon drum and some scrap metal into a functioning grill. The water rocket project used pop bottles and other materials to have water and compressed air for propulsion. Our trashed items like water bottles soon became high-flying engineered works. Other rockets composed of straws and matches that were designed by students often fizzled out and never became successes such as the past water rockets. During failures and after successes, students kept the engineering design process in mind. The STEM class taught the virtues of the engineering design process like how to research, design, construct a prototype, test, and redesign. Despite the hands on activities provided in the district's current program, it lacks in the technological field. We did not code, master hardware, or design web pages in the STEM class. This is a disappointing reality because is predicted that much of the STEM field's job growth will be in computer sciences.[5] My school does not offer a distinct computer science program like many other schools in the state of Michigan. In fact, fewer than 50% of schools in the United States offer a computer science program.[5] The few schools that have these programs are

Proceedings of the 2018 ASEE North Central Section Conference

located in the states that require the programs to graduate like Texas, Florida, Indiana, New Jersey, and Georgia. Many other states, including Michigan, make STEM courses mandatory for graduation. This does not help much with the computer science field because schools do not offer a specific computer program.[6] Because of this, secondary schools are lacking in providing an education for students potentially or unknowingly interested in the STEM field.

I have recently participated in the Research Experience for Teachers (RET) program which expanded my knowledge in the entire STEM field. During my time in the RET program as a secondary student I worked and researched a wide variety of projects. I was one of five secondary students allowed into the program. As a secondary student I worked with student teachers, teachers, professors, and graduate students on many of the projects during the first six of twelve weeks of the program. The program which was funded by a grant from the National Science Foundation (NSF) has given me an opportunity to explore STEM fields in a greater depth that may not have been otherwise possible.

2. Program Implementation Experience

In the first week of the RET program not many people knew anything about programming. Participants such as I did not know what to do when handed an Arduino Uno, breadboard, wires, and sensors. The specific Arduino used during all of the tasks in the program was an Arduino Genuino Uno board that is larger for beginners. The Genuino Uno has 14 digital pins where 6 of them could be used for Pulse Width Modulation (PWM) pins and 6 analog pins. The board also has a USB type B connector, power jack, and reset button [4][14]. This type of Arduino has no on board Wireless Connection so a shield is necessary when connecting the the internet. Secondary school students were given that Arduino, a walkthrough paper, a partner, and a graduate student for help. With a lot of help they were able to understand how a loop, library, and General Purpose Input Output (GPIO) pin worked, and what all of them were. They learned those by doing a couple of labs, tutorials, and many graduate student explanations. Some labs covered in the program were wiring up and typing in the programs for a temperature and humidity sensor, a motion sensored alarm system, and a simple LED light. It does not take long to learn that commenting what is being tasked in each line of code is a must. One particular lab covered was where an ultrasonic sensor was hooked to an Arduino and if an object was within a certain distance, then the light would turn on. A similar lab covered the same thing but now had a motion sensor instead of a ultrasonic. In instances like those the C++ language needed an if-else statement where if a condition was met then an action would occur, otherwise another action would occur or nothing at all. To continually run that program where it would read the data from the sensor there had to be a while loop meaning while something was true then it would continually run all of the program in the loop. The schematic of Figure 2 shows how to wire an ultrasonic with LED to an Arduino.

After becoming a little familiar with the Arduino and the language C++ the next step was to begin researching for the following project. The reference articles read were able to demonstrate new information to the participants about the Internet of Things (IoT) and connected vehicles which turned out to be the main research for the big project of the program. Many of the journals read talked about vehicle-to-vehicle communication, vehicle-to-pedestrian communication, and vehicle-to-infrastructure communication[3]. Through that research the audience of the journals could figure out the basic idea of the IoT which is where any smart technology can gather information and send it up to a cloud for other smart technology in the cloud to bring down and use. The IoT connects all of the technology in a certain region allowing for a more advanced area. After the research the secondary school students joined forces with the student teachers, teachers, and professors to experiment with RC cars, ultrasonic sensors, and Raspberry Pi's. A Raspberry Pi is a single board computer similar to the Arduino Genuino Uno[13]. Throughout all of the program multiple Pi's were used with various SD cards. The SD cards were interchanged between the Pi's because some had Raspbian Jessie while others had Wyliodrin installed. Raspberry Pi's can operate on many open source Linux softwares such as Noobs and Raspbian. The Raspberry Pi 3's that we used had 40 pins total, 4 usb 2 ports, HDMI port, microsd slot, micro usb power supply, and a Quad Core 1.2 GHz Broadcom BCM 1837 64

Tasks	**Weeks**
Work with C++ and Arduino Uno to hook up sensors and work with if-then statements	1-2
Research for IOT and Connected Vehicles With Notes	2
Start with python basics(LED and Ultrasonic Sensor)	3
Start hooking up RC vehicle to Pi's	3
Start to explore what PuTTY is	4
Start working with Wyliodrin with other more experienced group members	4
Work on Wyliodrin by myself with new sensors of Grove kit	5
Intro to actually working with the IOT	5
Practice with Python on Pi in a little more depth than before	6
Arduino Vehicle building-coding	6-7

Figure 1. Schedule of Tasks

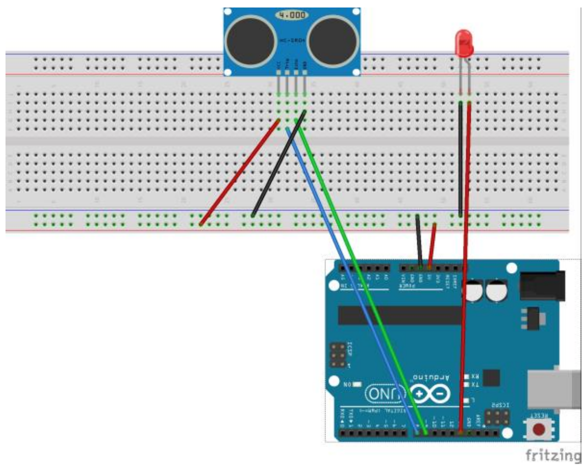

Figure 2. The lab demonstrated in this figure had the Arduino programmed with an if-else statement where if the ultrasonic sensor sensed an object closer than 20 centimeters it would give power to a pin to turn on the light, else it would do nothing to any other pin.

bit CPU [15]. That experiment turned into a challenge of having the cars stop before hitting an object look to schematic 2. In order to accomplish this challenge the sensor had to be tested to determine accuracy to a certain distance. The accuracy of the sensor is recorded in Figure 4 The Raspberry Pi was wired to sense information as the schematic in Figure 5 shows.

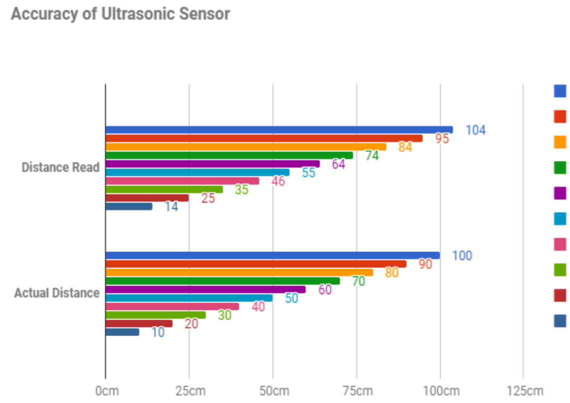

Figure 3. The graph shows the accuracy of the same ultrasonic sensors that were being used in the lab for almost all of the projects. The three sensors tested were consistently inaccurate. Other sensors used in a few other projects were very unreliable, unlike the sensors tested here.

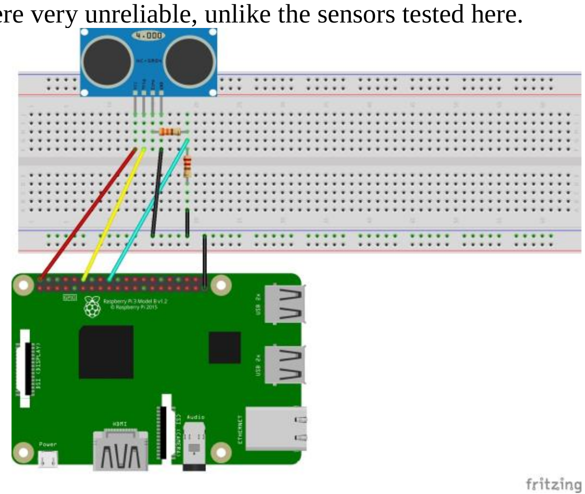

Figure 4. The method of wiring used in all projects throughout the program is pictured above.

The first attempt to have a Pi on the car and a Pi controlling the controller because there would be a client and server talking back and forth to each other. After a lot of trouble shooting it was determined the specific cars that were bought for the project had controllers that were not able to have a Pi control them. After having to get a different model of RC vehicle, the Pi had no problem controlling the vehicle and was able to communicate with the Raspberry Pi. Due to limitations on time the project had to be abandoned but at that time it partially work. It only worked at times because of a problem with the Raspberry Pi lagging, the limitations of the sensor, and an unreliable internet connection.

Because of an upcoming project the group was directed to start working on multiple sensors and motors to be controlled by a Raspberry Pi using Wyliodrin to test out that program. During testing, a few sensors worked with the program and more sensors may have worked but the purpose of the group using Wyliodrin was to determine how easy it is to use for younger students. The group had heard that Wyliodrin had a block program easy for younger students to plug and play with "puzzle pieces". The "puzzle pieces" were in normal English telling the user what they do while off the to the side it shows the real code being used to run the board. With the purpose of that software in mind the group determined it is not user friendly enough for those younger students they were planning to target. With Wyliodrin analog sensors and displays were capable of being used easily but no digital sensors. The goal of the larger project was to not have server and client Pi's but to have one Pi take in data from a sensor, send it to a cloud, be able to bring it back from the cloud, and access the data to determine how to operate or steer the motors as demonstrated in Figure 6.

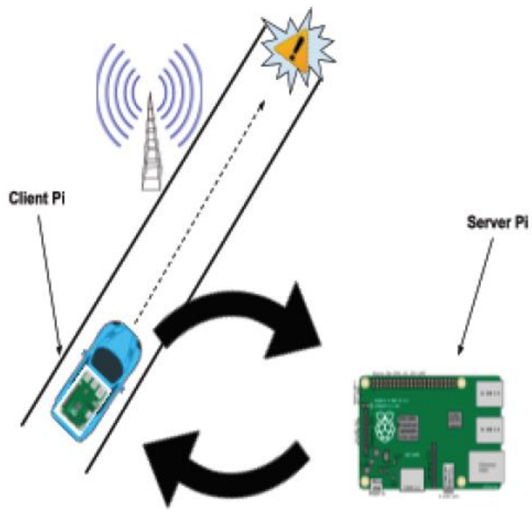

Figure 5. This figure shows the Server Pi communicating with a Client Pi through the internet. The Client Pi senses distance with the ultrasonic sensor connected to it and then sends the information to the Server Pi, the Server Pi then evaluates the information and continually sends back the decision it makes to the Client Pi. Based off the Server Pi's decision, the Client Pi actually controls the vehicle.

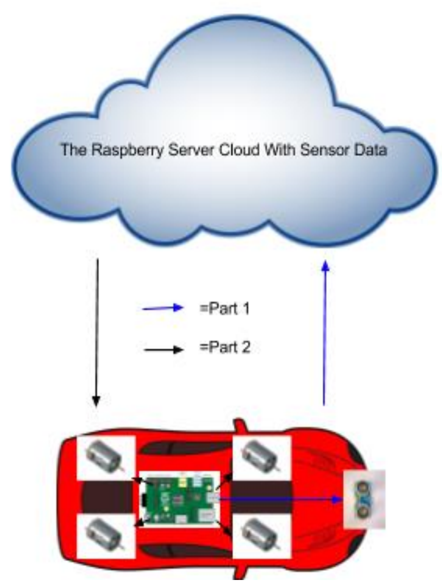

Figure 6. This is the first thought of how to use the internet of things to control the vehicle. First, the Pi is going to control the sensors then send information to the cloud, then bring it back down from the cloud when it knows how to use it, like how to control the vehicle.

That research and testing step was a struggle within itself. The program Wyliodrin is what was planned to use but it was not as user friendly as expected so a different program was necessary. In the meantime another group began assembling a SunFounder Pi Car V and Pi Car S for testing out programs on it to get it to accelerate as much as needed. When assembling and programming the car many participants had to put together and take apart certain components many times for multiple reasons. Some problems that had to be fixed were misplaced wires, having to recalibrate the steering, and reattach wheels that came off when calibrating the steering. Along with those hardware problems the group had the two motors operate independently. This occurred because the body of the Pi was off and connecting with pins in the back of the car. After that the next step was to get the Pi to operate on the vehicle when told to do so which was accomplished with a program the group was mildly familiar with called PuTTY. PuTTY is used to remotely access the Pi and tell it what files to run. Once the PuTTY program worked in this project the groups began working on getting a cloud for each Pi, constructing the programs for the cars, and for building a track for the car to navigate. It took a couple tries for some groups, and other groups didn't succeed, but eventually two Raspberry Pi vehicles that could sense data and make an appropriate reaction. The Pi was able to use a cloud service called ThingSpeak. And with about a 15 second delay the Pi was able to interpret data, send it to the cloud, and have the cloud send back what the reaction should be. In the program Thing Speak we had to set a condition to be met and we had to copy and paste MatLab Code for the Pi to Pi communication service to work. Once everything is all set up the Raspberry Pi sends information to the Primary Channel and that Channel evaluates it. Once the Primary channel senses the condition was met it sends that data to the Observation Channel for the person to see on a line graph what, when, and how many times the condition was met. The observation channel also sends the signal to the other Raspberry Pi telling it that it can do the action. The schematic below (Figure 6) shows how the Pi and other devices could communicate with ThingSpeak. This specific schematic pictures two Raspberry Pi 3s' with an Ultrasonic Sensor and LED light.

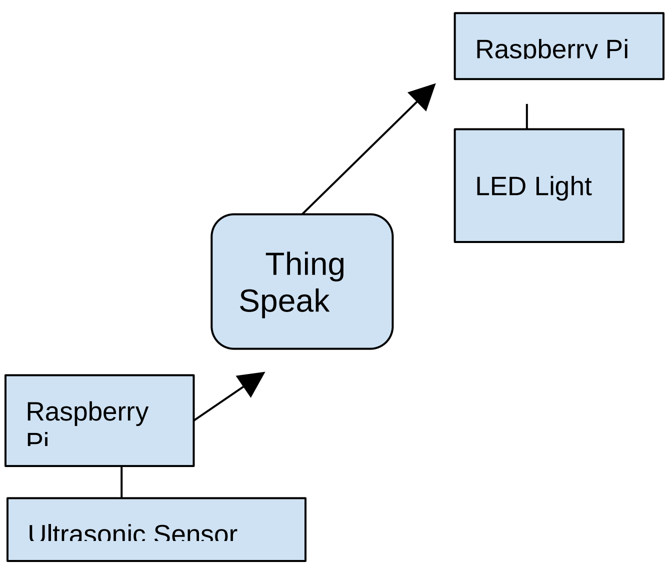

Figure 7. The schematic above shows the chain of events in how an Internet of Things program works. The ultrasonic sensor is the input while the LED light is the output. The schematic shows it with two Raspberry Pi's but it is possible to just use one board for it to work.

In week 6 and scattered throughout the program Raspberry Pi's were plugged into a monitor for Raspbian desktop to run. This allowed for participants to learn many different programming languages but most just focussed on Python 3 and Python 2. There were many similarities and differences between Python and C++. There were even some differences between Python 3 and Python 2.

Before and during the time of that project a more specific group were building and testing out a new type of Sun Founder vehicle that was controlled by an Arduino Uno. The vehicle was similar to the Pi vehicles but the group still experienced some difficulties. First, a big mistake was made due to lack of attention by accidently having two wires touch which burnt their battery pack. After that the process wiring continued and was done just as the pictures had them shown. Well, some of the pictures had the certain modules turned around and upside down misleading the group and the vehicle was wired wrong. When nothing would work it was suspected that it was the battery pack which was bad since the wires were glowing and the plastic melted. An extra battery pack was found to replace the burnt battery pack. All was well and the vehicle had power for a while until the it started to smoke again. The wiring was carefully examined with graduate student and he found some wiring mistakes and fixed those. Once again nothing worked after the new wiring. The graduate student looked over the vehicle with the group and they tested one wire after another, plugging them into different spots. After a lot of educated guesses a simple switch on the vehicle was thought to be the problem. The group hooked up the battery pack straight to the vehicle without a switch so the vehicle was always on (the group thought this was fine but it turned out not to be ideal.) The switch was fixed with a switch laying around the lab. A wire was simply wrapped around each end and wrapped with electrical tape for a temporary fix. After all of the wiring problems the Arduino Car worked with code provided by the company from which the car was purchased from. The codes were retrieved and able to work but also a couple codes had problems. The end of the program came up soon and the arduino group only got to get the line follower sensor and the obstacle avoidance working. The group did not get enough time for the Arduino vehicle to work with ThingSpeak at that time like they had hoped to.

3. Conclusion

RET participants of my group took home plentiful information about programming, microcontrollers, and simple electronics. Throughout the program the engineering design process had been crucial. Some instances when using the process was during the wiring of the Arduino vehicle and nothing worked or when testing out sensors on Wyliodrin. These moments bring what have been learned in the past to a different field. I say this because in my past STEM class we did a lot of hands-on-activities of a different nature. Some areas of engineering that were worked on in the RET Program are things that my school and many other schools don't have to offer yet.[5] For example, my school has not offered the level of computer science or types of hands on activities which I covered during my time at Central Michigan University in the RET program. By learning more about these topics everyone in the program has become more fascinated with STEM as a whole. When working alongside student teachers, graduate students, teachers, professors, and other students, they all gave me an insight on the education of STEM like I have not experienced before. They also showed me how anyone has engineering capabilities. Working with this technology was eye opening because many people don't realize what microcontrollers even are. If my school receives a program similar to this it will be interesting to see other students in my area get an opportunity comparable to mine. When speaking to others in the program, everyone was in agreeance that it was a great experience to have and everyone had a positive outlook on engineering.

Acknowledgements

This work has been sponsored by the National Science Foundation under Grant no. 1542368.

References

[1] Inclusion by Design Engineering Inclusive Practices in Secondary Schools. http://journals.sagepub.com/doi/pdf/10.1177/004005990904100302

[2] Secondary student research projects in engineering: optical fibre Bragg grating sensor applications. http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1780&context=ecUworks2011

[3] WPA-Autonomous-Drive.ppt by DENSO Doug Patton

[4] Arduino Uno Rev 3 https://store.arduino.cc/usa/arduino-uno-rev3

[5] Support K-12 Computer Science Education in Michigan https://code.org/advocacy/state-facts/MI.pdf

[6] States Move towards Computer Science Program http://www.nasbe.org/wp-content/uploads/Tilley-Coulson_Computer-Science-Final.pdf

[7] STEM crisis or STEM surplus? Yes and Yes https://www.bls.gov/opub/mlr/2015/article/stem-crisis-or-stem-surplus-yes-and-yes.htm

[8] Friends of Fritzing (2016). Fritzing (Version 0.9.3b) [Software] Available from http://fritzing.org/download/?donation=0

[9] GitHub (2017). Arduino IDE (Version 1.8.3) [Software] Available from https://www.arduino.cc/en/Main/Software#source

[10] Alexandru R., Ovidiu S., Ioana C., & Cristian R. (2016) Wyliodrin (Version Hobbyist) [Software] Available from https://www.wyliodrin.com/prices/

[11] MathWorks(2017). ThingSpeak (Version N.A.) [Software] Available from https://www.mathworks.com/help/thingspeak/

[12] What is Arduino? (n.d.). Retrieved August 15, 2017, from https://www.arduino.cc/en/Guide/Introduction

[13] Raspberry Pi FAQs - Frequently Asked Questions. (n.d.). Retrieved August 15, 2017, from https://www.raspberrypi.org/help/faqs/

[14] Rochelle salt integrated PIR sensor arduino based intruder detection system (ABIDS) http://ieeexplore.ieee.org/document/7918228/

[15] Raspberry Pi 3 Model B Specifications https://www.raspberrypi.org/products/raspberry-pi-3-model-b/