International Journal of Information Engineering and Applications
2018; 1(2): 37-50
http://www.aascit.org/journal/information

RoboTown: Robots for Research and Outreach

Lee Boehm, Kellen Krammes, Shawn Mathes, Tyler Risinger, Emma Romig, Sridhar Ramachandran*

Department of Informatics, Indiana University Southeast, New Albany, USA

Email address

leeboehm@ius.edu (L. Boehm), kkrammes@ius.edu (K. Krammes), shmathes@ius.edu (S. Mathes), terising@ius.edu (T. Risinger), eromig@ius.edu (E. Romig), sriramac@ius.edu (S. Ramachandran)

*Corresponding author

Citation

Lee Boehm, Kellen Krammes, Shawn Mathes, Tyler Risinger, Emma Romig, Sridhar Ramachandran. RoboTown: Robots for Research and Outreach. *International Journal of Information Engineering and Applications.* Vol. 1, No. 2, 2018, pp. 37-50.

Received: January 22, 2018; **Accepted**: February 12, 2018; **Published**: March 10, 2018

Abstract: The focus of the research and development project discussed in this paper is to design, formalize, implement, and deliver a cost-effective, user oriented, useful and usable tool (called RoboTown) for easy use by middle school children and their teachers. The research team began with the goal of making a middle school student friendly research tool to seamlessly introduce the concepts of information systems to learners through robotics and electronics. The completed project includes a fully autonomous rover-style robot built using 3D printed material around an Arduino Mega microcontroller, along with a fully functional companion website that serves as the main source of the lessons and includes; instructional videos, interactive programming lessons, instructional resources, as well as links to download all the material for offline use in a computer laboratory.

Keywords: Information System, Informatics, Arduino, Robotics, Electronics

1. Introduction

With recent legislation and shifts in school standards in the State of Indiana, student's educational plans and educator's lessons are less in the hands of the educators and more at the mercy of state standards [6]. While this does have some positive effects on students and classrooms alike, this also takes away from elective courses, enrichment programs, and subjects that fall out of the state standards. This is currently the area where robotics is falling in with most middle and high school education programs. While some larger area schools may have the ability to offer a select few students the ability to pursue projects where STEM fields like robotics and electronics could be covered (because of the cost involved); other schools and their students may not be so privileged [6]. This is quite unfortunate for this select population of the student body, as they are unable to potentially be introduced into a subject they could come to enjoy. This is where the STEM (Science Technology Engineering and Math) researchers (authors of this paper) at Indiana University Southeast (SE) have chosen to focus, and have set sights on finding a way to bring to these students the opportunity to be introduced to robotics and electronics where none existed before. The authors believe that if students have an easily accessible and programmable robot that they can build and/or access with minimal to no cost; it can help level the playing field across all schools.

In order to ensure that these goals are met the authors as a collective research group chose to create a multilayered project. The project was broken into three deliverables; a companion website designed to be useful for instruction, a set of how-to videos for learning, and the physical robot as the end product. By breaking the project into these deliverables the authors could spread out the team in order to maximize efficiency and allow aspects of the project to be completed simultaneously as opposed to following a strictly linear progression. The website itself will be a welcome and first step for the school students when experiencing the program. It will introduce the students to the robot as well as give a bit of background information in order to ensure that every student starts with the same knowledge baseline. It will also house all of the lesson plans that the students will use when navigating through the course.

The lessons themselves will be broken down into approximately ten separate parts, progressing from one to the next, and building off of each other while increasing in difficulty until a final task is undertaken by the students to show their mastery of the material. Each of these lessons will be accompanied by video how-to guides ensuring a multimodal and multiliteracies approach to learning. These how-to guide videos will be short and to the point, and will accurately describe exactly the task the students are given for the set lesson as to allow the students to follow along step-by-step if need be. These lessons are not an attempt to force the students to learn by introducing them to the various concepts, but an effort to help walk them through a complex topic that the authors as the designers have enjoyed in hopes that the students themselves will gain a growing interest in the topic as well.

The original plan for these lessons was to have a read and response style interaction with the students, where they would be guided through an example and then be left to recreate a similar code on their own. However, the authors discovered (through trial and error and user testing) that this would result in the entirety of the program becoming extremely long, as well as hard to evaluate for self-correction because of the many different pathways that programmers can take to reach to the final result. So, instead the authors chose to show the students a step by step guide into how the authors created the robot, and walk them through hard coding everything in the loop of the program, followed by creating a method for that set of code. This allows for continued rehearsal of key concepts and in the end allows for the students to show their creativity by recreating the robot program whichever way they decide.

The robot that the students will be able to work with is a fully autonomous rover. This design was chosen for the ability to help gain student interest quickly, all while being able to utilize many common elements of programing and electronics. The body of the robot is entirely made in house at the Indiana University SE Library in the 3D Printing Lab. This became the method of choice in order to create a robot that would be strong and light enough to be handled by younger students, yet any part can be replaced easily and in a timely fashion (if the school students have access to a 3D printer at a makerspace[1]). The robot will be powered by a pack of five AA batteries in order to make sure that if and when the robot loses power it can be easily restored by the user or an instructional aid. As for the choice as to what type of hardware to add to the robot, the authors settled on using the Arduino Mega microcontroller board [4]. This choice was made for a multitude of reasons that includes – the Arduino board is open-source [12] that allows for making changes to it, the ease of the programming language that it uses, and the ability that the board offers to consistently change the overall function of the robot. Moreover, by design the Arduino Mega microcontroller board will be exposed on the top of the robot, along with the motor shield [3] and the breadboard [4], in order to allow the students to actually see how everything is put together and wired. This also allows free transformation and modification by the students, granted they feel comfortable to do so, as well as the simplicity of converting it back to its original form.

2. Research

2.1. Learning Standards

While there are not many standards in the State of Indiana that follow along and introduce robotics, there are areas where robotics can be linked back to said standards [14]. By analyzing the Arduino programming environment the authors determined that the most benefit (learning wise) did not necessarily derive from an in class instruction during the school day, so the authors decided to explore it as activity for an enrichment program that students can access on their own free time or assigned time during the school day that allows for students to reach out to a subject not covered in-class. The goal of the project (that the authors call RoboTown) is not to take away from the basic standards that educators are made to abide by, but to add on to these subject areas and help broaden the overall understanding of how robotics and electronics is relevant in other subjects and topics. The aspect of programming for example is closely intertwined with mathematics and logic, which are core concepts in the Indiana State Standards beginning in the sixth grade[2]. By taking concepts the students are already becoming familiar with, the authors can add more context to these subject in the hope of boosting the level of knowledge in both. The majority if not all of the programming that the project entails will directly be a reflection of the key concepts of mathematics and logical reasoning that these students will be seeing everyday in their current classroom settings. The authors hope that this will allow for a base level of understanding for every student, all while allowing every students who chose to partake in this course a smooth and successful learning experience.

2.2. Web Solutions

First, when looking for a web based solution to teach children aged 10-15, one must look at other websites designed for that age range. Some great examples of successful websites in this age range are ones based around kid's TV programming. Since a large portion of the RoboTown project is video lessons, websites like PBSKids.org, Nick.com and CartoonNetwork.com are all great examples of websites designed for kids. The primary challenge point for a website for children is navigation and usability; it must be simple to understand, while remaining engaging and entertaining. Video and animation is key to enabling this engagement with kids. Looking at the solutions applied by PBSKids.org, Nick.com and CartoonNetwork.com, the RoboTown team was able to find a successful method for navigation for kids. For the reader's benefit, each of inspirational website's solution is shown and explained.

PBS Kids (see Figure 1) employs a circular navigation

1 https://spaces.makerspace.com/

2 https://www.in.gov/sboe/files/2014-04-15_6-8_Math_draft_041414a.pdf

structure, with larger, more important navigation buttons separated from the main navigation. This allows for kids to find the important things easily, while still having a navigation menu that lets them find all their favorite shows. At the top of the page, there is also a link for parents, essentially separating the website into two sections. The colors here are all different and vibrant, and the background with animation, reinforcing the idea that video and movement engages children.

Figure 1. *PBS Kids Website (retrieved 01/01/2017).*

Nickelodeon's website has some similarities to PBS Kid's with some key differences. Nickelodeon's website is primarily targeted towards tablet-platforms; it has large buttons, a scrolling show menu at the top, and has the majority of the page taken up by a large carousel with very large menu buttons. Large buttons in combination with the bright colors allow for an engaging experience for kids. However, this page lacks animations, and while good for bandwidth, it is less engaging in that aspect for kids than the PBS site.

Cartoon Network utilizes some of the same features used by Nick's website including large buttons for navigation. A more traditional navigation menu here is indicative of a site that is aimed at older children, up into the later teens. This site is also designed aimed at tablet computer users. While the authors took inspiration from these sites, the website design for the RoboTown project ended up unique since the authors were designing for desktop users since kids will be learning how to code using a variety of platforms (desktops, tablets, mobile phones etc. 0 and so a desktop styled design will help account for these flexibilities. Interested readers can visit the project website at http://robotown.org/ to see the completed design.

2.3. Presentation as a Service

It was important to the authors that the RoboTown project is perceived with a singular and solid message. The project should appear as a brand and the things that contributed to this brand included consistent names, logos on all products (website, robot, transportation case, documentation) and even things like having a storyline or a reason for kids to program the robots. For example, the underlying original backstory behind RoboTown is a narrative about a town of robots building blueprints for a robot to defend their town from "rust bots."

2.4. Preparing for the Robot Build

Upon the inception of the project, the team decided to repurpose some old robots (Parallax Scribbler[3] robots) to produce programming lessons. The Scribbler was an interesting concept that would have enabled individuals to program the robot to take photos, draw pictures, and maneuver obstacles. However, the Scribblers that the authors procured were very old and plagued with several inconsistencies such as; failing to respond to commands that were coded into, freezing whenever ran against its native application, and its outdated serial port was adding to the complexity. Such issues required that the authors analyze additional options in order to ensure that they produce a better overall user experience.

The options for alternatives left the authors with either choosing from pre-built robotic kits, such as the newer Scribbler 3 robot[4], or they could instead purchase all necessary parts to design and build their own robot. Since the authors wanted to bring more than just the ability to learn programming to the students, they decided to build their own robotic kit. This option will enable the students to comfortably apply what they learn through the lessons in a more intuitive manner.

The first step to building the kit was to determine the microcontroller that will be used as the overall "brain" of the robot. Microcontrollers allow the designer to interface sensors and specialized control electronics together (along with anything else required for the project) and contain the overall logic of the robot [8] [11]. The research into microcontrollers led to discovering a wide array of options; boards for high-end robotic production, down to boards for the everyday, casual user. Therefore, the authors narrowed this result to three possible open-source based choices; Arduino[5], BeagleBone[6], and Raspberry Pi[7]. The authors decided to go with Arduino – here are some of the reasons for this decision. Arduino are popular microcontrollers that have been designated as one of the best starting points for beginners in both robotics and programmers [9] [1]. The high appraisal of the Arduino microcontroller is due to its large backing of the Arduino community and robust collection of models users can choose from. BeagleBone is very similar to Arduino's except that it fails to have as large of a community backing and isn't the most ideal solution for absolute beginners. The BeagleBone excels in projects that require interfacing with sensors and utilizing a network connection [7]. The last of available options, Raspberry Pi, is essentially a different breed between the two previously mentioned [13]. The Pi, which can act as a miniature computer, has become the board of choice for many individuals setting up theatre systems or even creating arcade machines loaded with nostalgia inducing classics.

3 https://en.wikipedia.org/wiki/Scribbler_(robot)
4 https://www.parallax.com/product/28333
5 https://www.arduino.cc/
6 http://beagleboard.org/bone
7 https://www.raspberrypi.org/

In the end, because of its large community and being aimed towards use by beginners, the authors chose the Arduino line of microcontrollers. During the research for the particular microcontroller, the authors identified two models from the Arduino line that would adequately suit the needs for this project; Arduino Mega 2560 and the Arduino Uno. The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button[8]; whereas the Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Therefore the biggest factor for the authors between the two was their overall size differences. Hence the authors decided to go with the Arduino Mega 2560 (explained below).

A central goal of the project is the ability for the school age kids to recreate a fully autonomous robot, so the microcontroller must have the capability to interface with various pieces of external hardware. Such hardware includes two DC motors, two HC-SR04 distance sensors[9], various LED lights, piezo speakers[10], and the Arduino Motor Shield Rev3 [3]. The Arduino Motor Shield Rev3 simplifies the approach to interfacing with various motors, such as DC motors, servo motors, step motors, etc. The overall size of this hardware when attached to the microcontroller, was in line with the Arduino Uno. Because the motor shield renders several pin connections unusable when DC motors are interfaced with it, the size distinction between the two boards would act as the deciding factor as to which would be chosen. Therefore, because this project could have the potential to expansions in the future, the authors wanted a board large enough to allow such expansions, leading to decision for the Arduino Mega 2560 [2].

3. Project Execution

3.1. Preparing for the Robot Build

Before beginning designing the look and feel of the robot, the authors knew that it has to be a simple design that the kids can edit and recreate for themselves if they wanted to. With that being said, the authors chose to 3D print the entire casing for the robot. This initial design phase had undergone several iterations before the authors truly felt that the authors had grasp of a solid body design for the Robot. While the authors had several issues with the functionality of the Scribbler robot, the design of its body was one that they felt could easily implement into this project. Iteration one, shown in Figure 2, was planned to have easy access to all essential hardware in the center of the body, while three ping sensors would be mounted in the front of the robot and one in the rear. Conceptually the idea would have worked great, but once printed, the authors found that the hardware was not fitting properly within the casing, nor would the lid fit into its intended location. On top of all of this, weight distribution on the two DC motors was causing too much pressure to be placed on them, causing no movement whatsoever.

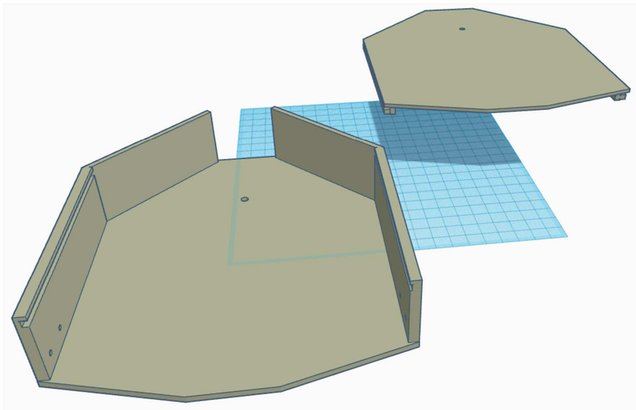


Figure 2. *Scribbler inspired design.*

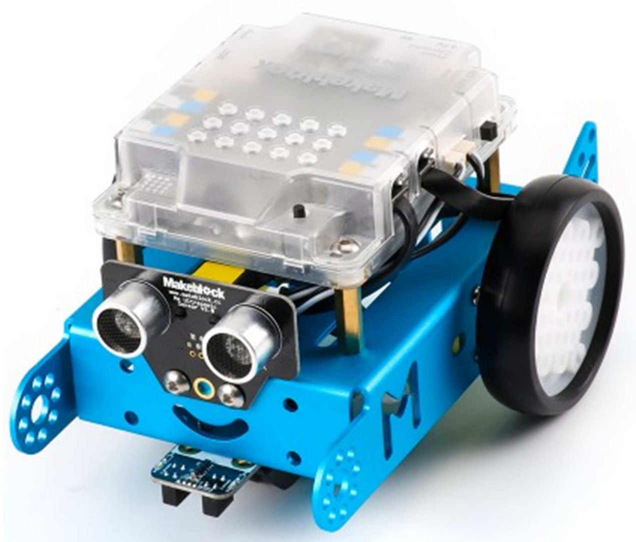


Figure 3. *MakeBlock mBot v1.1.*

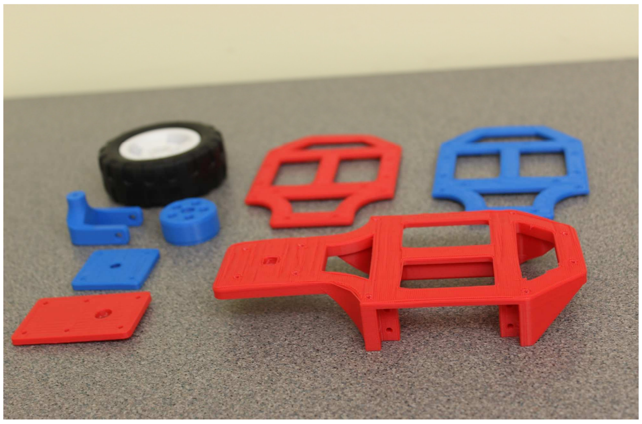


Figure 4. *Iteration 2 Design Side View.*

8 https://store.arduino.cc/usa/arduino-uno-rev3
9 https://www.sparkfun.com/products/13959
10 https://en.wikipedia.org/wiki/Piezoelectric_speaker

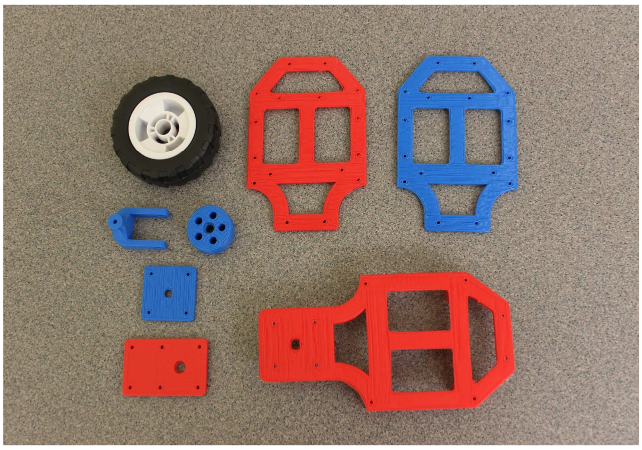


Figure 5. *Iteration 2 Design Top View.*

To fix this issue with the design of iteration one, the authors decided to take a step back and look into body designs being used by popular robotics kits designed for children. The research led to the finding of the Makeblock mBot v1.1[11] (see Figure 3), which is compact in its design and enables the children to access all components without much of a hassle. Because it was crucial to expedite the design process during the second iteration of printing, the authors utilized the Thingiverse.com online repository which contains projects that various members of the community have designed and distributed. The authors decided on a design that would be a bit larger than that of the mBot v1.1, since it needed to house a much larger microcontroller, and came across the design developed by Outcastrc[12] (see Figure 4 & Figure 5).

The idea of this project was to not only teach the kids how to program, but to also teach them to independently develop interesting projects with the Arduino microcontrollers. Thus, the project idea proposed was to build a fully autonomous robot that can navigate passed any potential obstacles. Figure 6, depicts the overall finalized design concept that the authors had originally sought to achieve with the completion of the robot. However, due to several glaring issues, such as, inconsistent color patterns, the motors being exposed to tampering, and the rear-mounted breadboard hanging too much, a third and final iteration had to take place.

The third and final revision to the body of the robot fixed the issues of varying colors, decreased the size of the back resting plate, and added a motor well (see Figure 7.). For this build the authors opted to go with a full red body to save the overall printing time. Decreasing the resting plate on the back of the robot enabled the authors to comfortably fit the smaller breadboard on it without much overhang. Lastly, the addition of the motor wells added stability for the position of the robot; this also prevents the users from accidently tampering with it, and potentially breaking the motor.

11 http://www.makeblock.com/mbot-v1-1-stem-educational-robot-kit

12 https://www.thingiverse.com/thing:248423

3.2. Programming Languages

In its default state, the Arduino IDE accepts programs written in C/C++ [4]. Within the Arduino community, many developers have written libraries that enable tinkerers/programmers to serialize their data through a range of other languages, including C#, Java, and Python [10]. Initially, since the Scribbler robot interfaced well with the Python programming language, the authors had considered to utilize this as the language of choice, since Python has been adopted universally as the introductory language to computer programming. Instead, the authors opted to cater the lesson plans around Arduino's native language, so that after the students have completed all the provided lessons and have a firm grasp of the basics, they would have no issues purchasing an Arduino of their own and diving right into their own independent projects.

Figure 6. *Iteration 2 of the design process.*

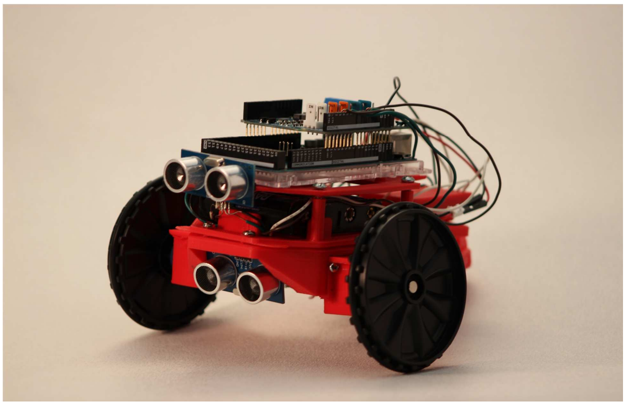
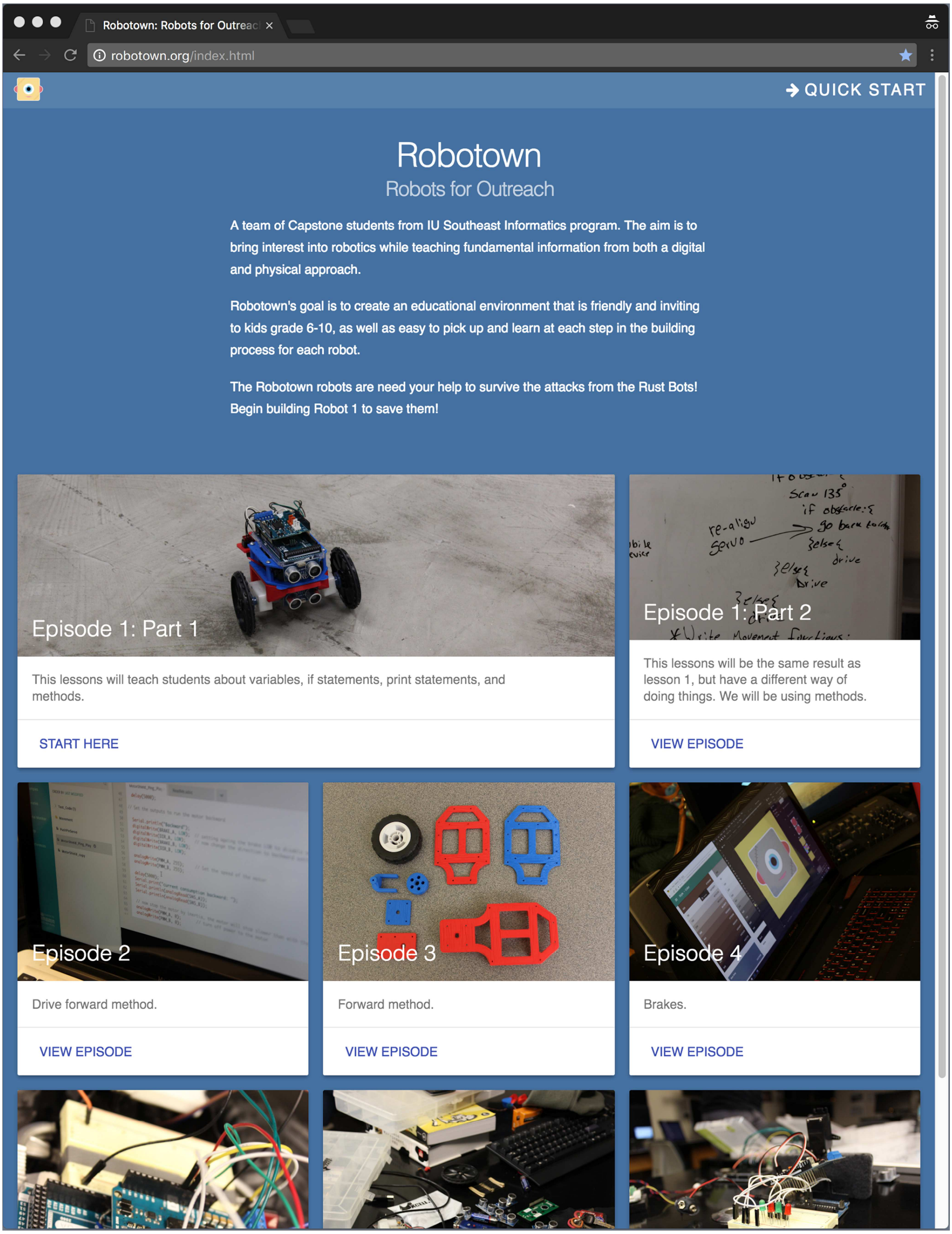


Figure 7. *Final design iteration.*

3.3. Website Implementation

When creating the website, the authors chose to stick with the original logo (that the authors designed – check the left corner of the project's website) and find colors that coordinated with it. The font chosen was white (Hex Code Color [HCC] #FFFFFF) and the background was a deep blue (HCC #4672A3). The contrast was not only easy to read but it was easier on the eyes when looking at the screen for a longer amount of time.


Robotown: Robots for Outreac
robotown.org/index.html
QUICK START
Robotown
Robots for Outreach
A team of Capstone students from IU Southeast Informatics program. The aim is to bring interest into robotics while teaching fundamental information from both a digital and physical approach.
Robotown's goal is to create an educational environment that is friendly and inviting to kids grade 6-10, as well as easy to pick up and learn at each step in the building process for each robot.
The Robotown robots are need your help to survive the attacks from the Rust Bots! Begin building Robot 1 to save them!
Episode 1: Part 1
This lessons will teach students about variables, if statements, print statements, and methods.
START HERE
Episode 1: Part 2
This lessons will be the same result as lesson 1, but have a different way of doing things. We will be using methods.
VIEW EPISODE
Episode 2
Drive forward method.
VIEW EPISODE
Episode 3
Forward method.
VIEW EPISODE
Episode 4
Brakes.
VIEW EPISODE

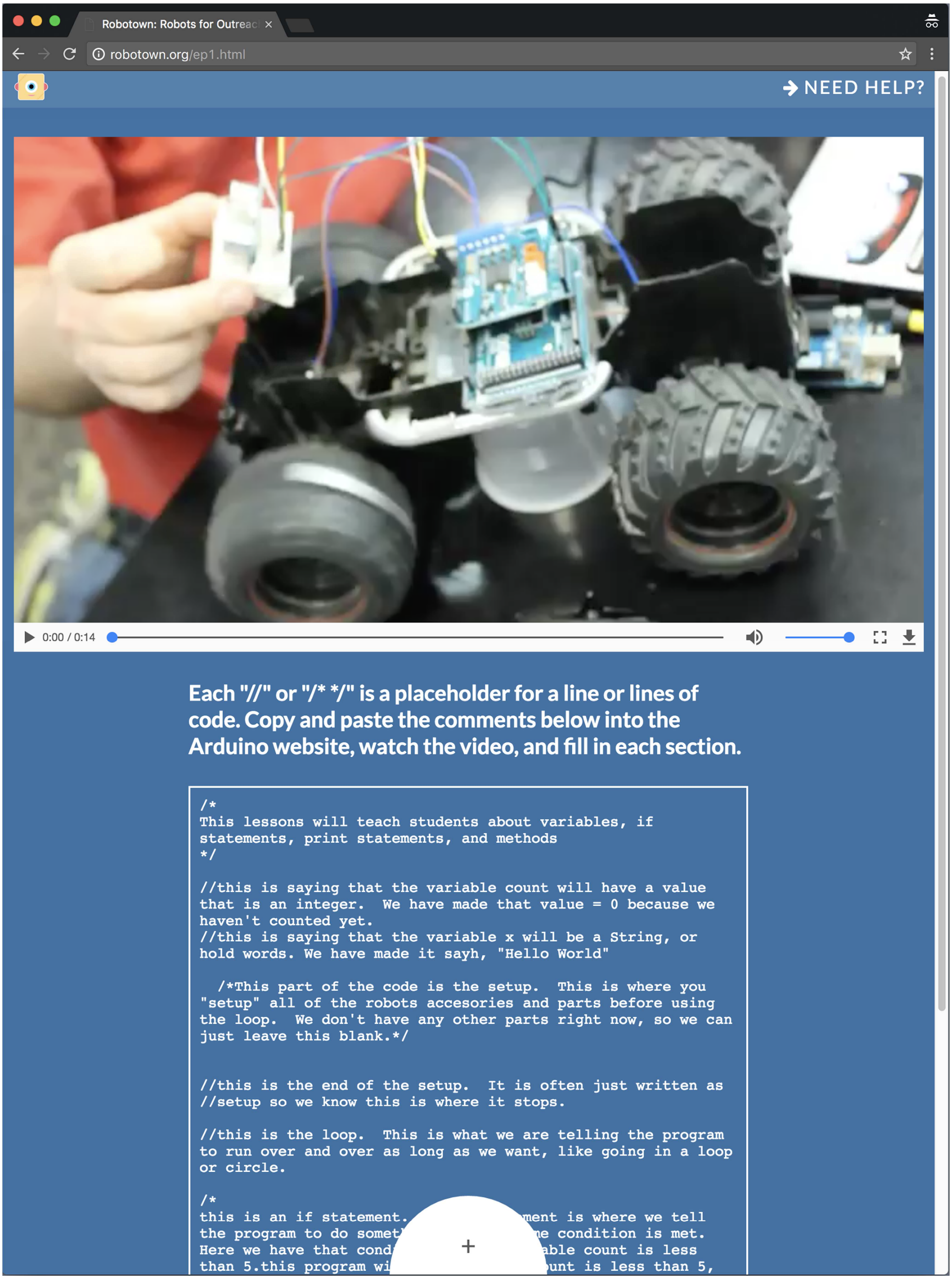


Figure 8. Images of the website.

For the home page (Figure 8) the authors implemented a grid pattern (instead of a simple text only screen). The authors utilized the Material Design Light card components[13] to show each Lesson on the homepage in the grid pattern. These cards could be coded by hand, but, Google MDL allows for quick implementation alongside a very robust responsive tool kit, since all devices with Google MDL are accounted for. All MDL elements are also built for optimal search engine optimization. Next, the authors decided to use a radial navigation (see Figure 9) for the website[14]. The authors then used the icons from the Font Awesome website[15] and which is available under the Creative Commons copyright. When thinking about color, the authors chose to use a slightly more vibrant yellow (HCC #FFDC89) than that within the logo, slightly darker accent yellow (HCC #F4CA4D) and white (HCC #FFFFFF) for the icons in the center. When the mouse hovers over one of the buttons, the colors of the icons and the background reverse, showing what has been selected by the user.

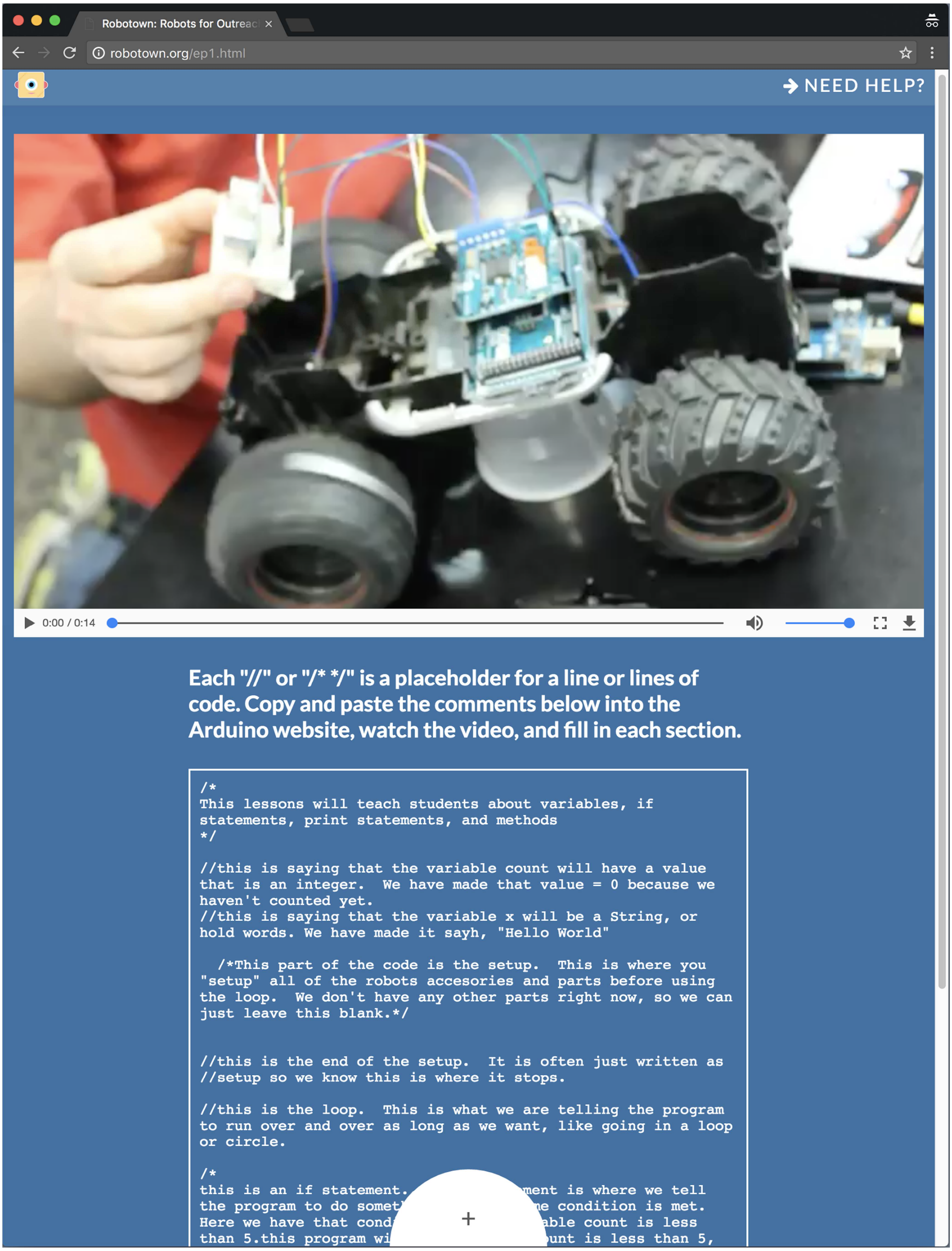
Figure 9. *The different hover effects within the video/lesson page's radial navigation.*

When the radial navigation is open, the background blue deepens (HCC #1C2D41) as well as the images and videos grow dim, not only to provide contrast, but to make the user more aware of this changed state. Under the video section, the authors have placed a box with comments about the code (Figure 10). This is to help a student follow along with the lesson by copying the code into the Arduino code creator.

13 https://getmdl.io/components/

14 Following the tutorial found at https://tympanus.net/Tutorials/CircularNavigation/

15 http://fontawesome.io/license/


Robotown: Robots for Outreac
robotown.org/ep1.html
NEED HELP?
0:00 / 0:14
Each "//" or "/* */" is a placeholder for a line or lines of code. Copy and paste the comments below into the Arduino website, watch the video, and fill in each section.
/*
This lessons will teach students about variables, if statements, print statements, and methods
*/
//this is saying that the variable count will have a value that is an integer. We have made that value = 0 because we haven't counted yet.
//this is saying that the variable x will be a String, or hold words. We have made it sayh, "Hello World"
/*This part of the code is the setup. This is where you "setup" all of the robots accesories and parts before using the loop. We don't have any other parts right now, so we can just leave this blank.*/
//this is the end of the setup. It is often just written as //setup so we know this is where it stops.
//this is the loop. This is what we are telling the program to run over and over as long as we want, like going in a loop or circle.
/*
this is an if statement.
ment is where we tell
the program to do somet
e condition is met.
Here we have that cond
able count is less
than 5.this program wi
unt is less than 5,
+

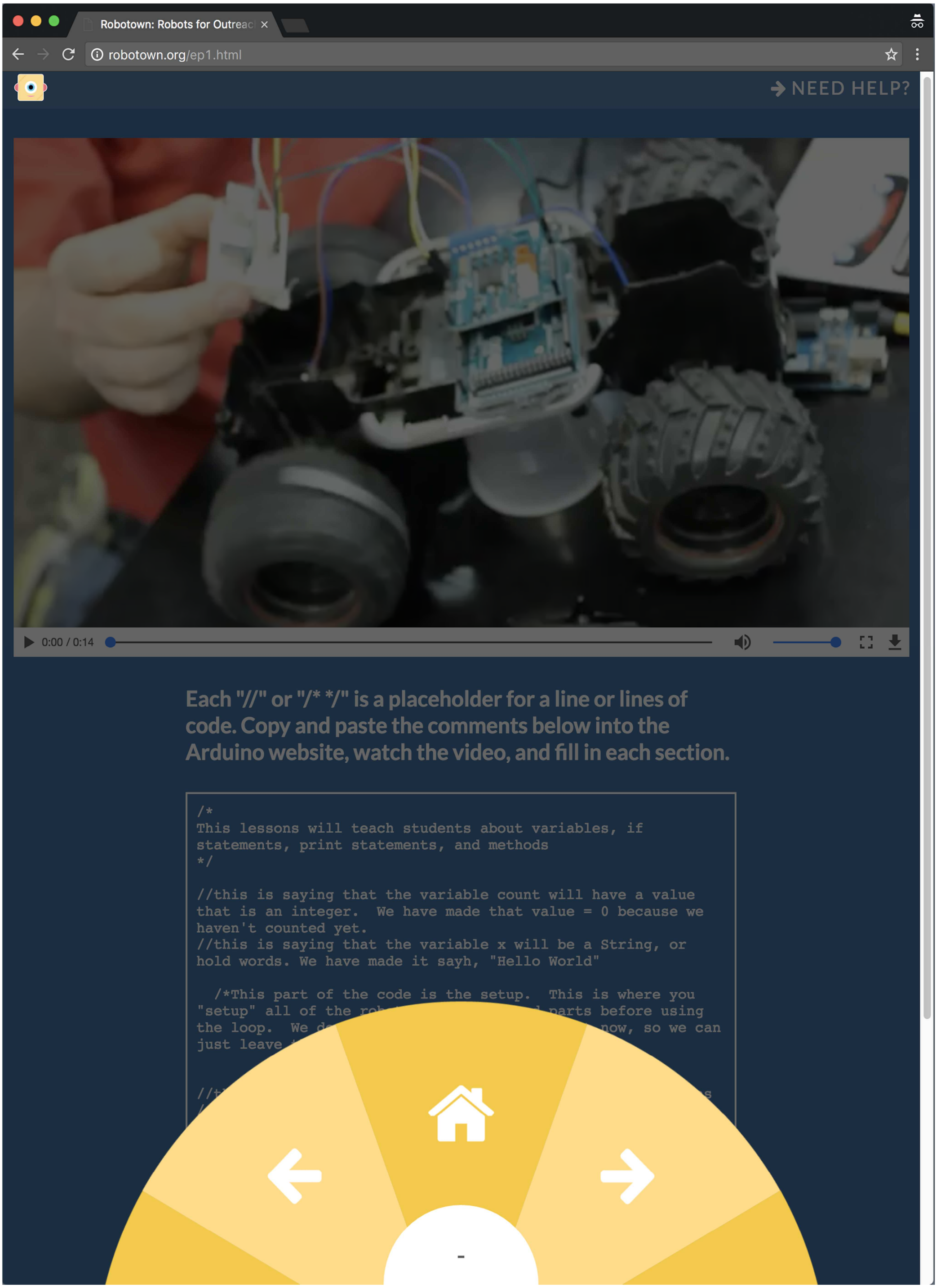
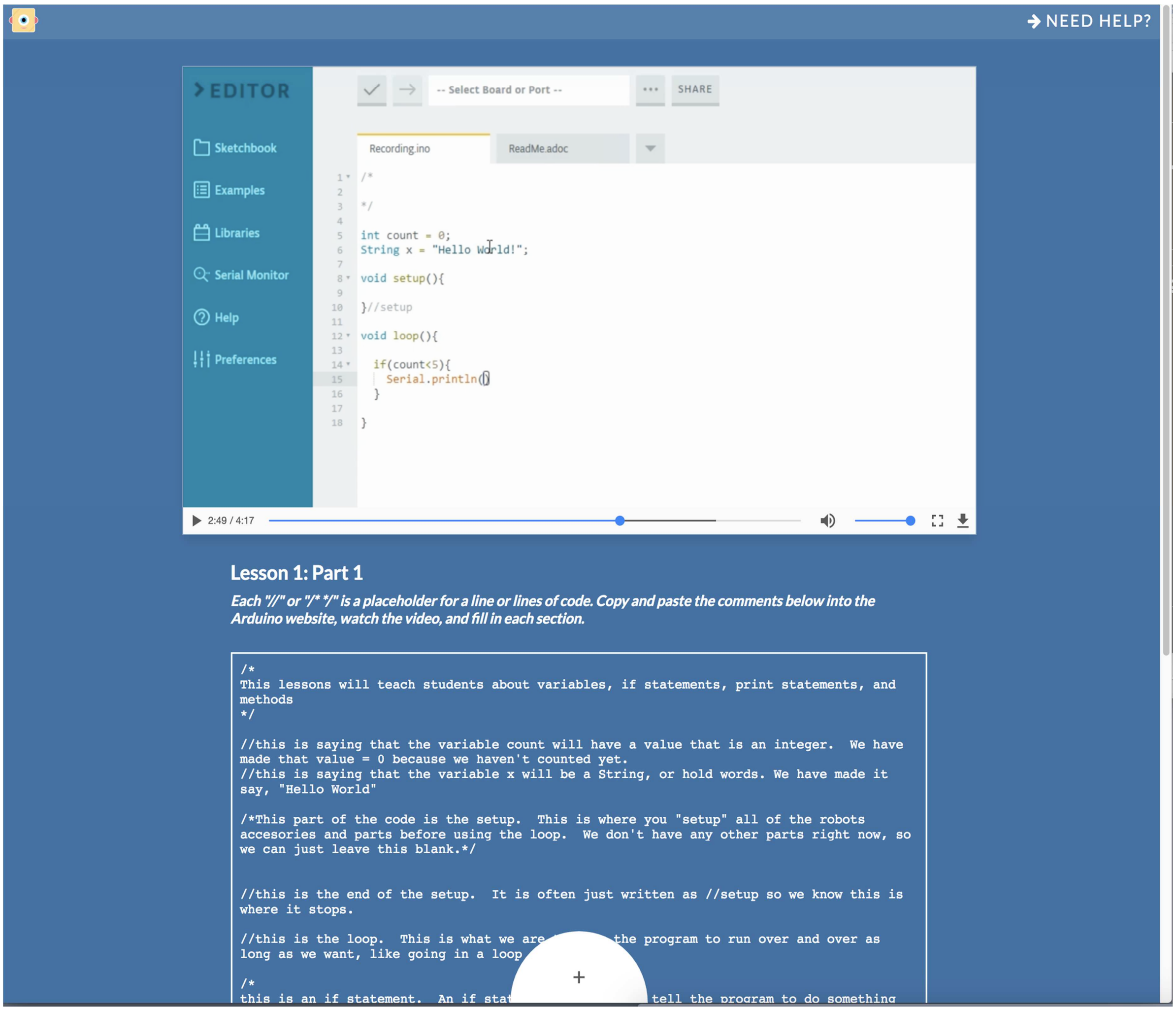


Figure 10. *Showing the dimming effect when the radial navigation is open as well as the box with code comments outlined.*

There is also a link to the editor below every comment box for easier access to students. By default, YouTube.com does not support responsive video. Also, embedding a YouTube video gives it a static width and height. To solve this problem when developing a responsive website, a JavaScript workaround (Figure 11) was used.

```
      <script>
        console.clear();
videos = document.querySelectorAll("video");
for (var i = 0, l = videos.length; i < l; i++) {
    var video = videos[i];
    var src = video.src || (function () {
        var sources = video.querySelectorAll("source");
        for (var j = 0, sl = sources.length; j < sl; j++) {
            var source = sources[j];
            var type = source.type;
            var isMp4 = type.indexOf("mp4") != -1;
            if (isMp4) return source.src;
        }
        return null;
    })();
    if (src) {
        var isYoutube = src && src.match(/(?:youtu|youtube)(?:\.com|\.be)\/([\w\W]+)/i);
        if (isYoutube) {
            var id = isYoutube[1].match(/watch\?v=|[\w\W]+/gi);
            id = (id.length > 1) ? id.splice(1) : id;
            id = id.toString();
            var mp4url = "http://www.youtubeinmp4.com/redirect.php?video=";
            video.src = mp4url + id;g
        }
    }
}
      </script>
```

Figure 11. *Showing the Javscript code for the responsive video workaround.*



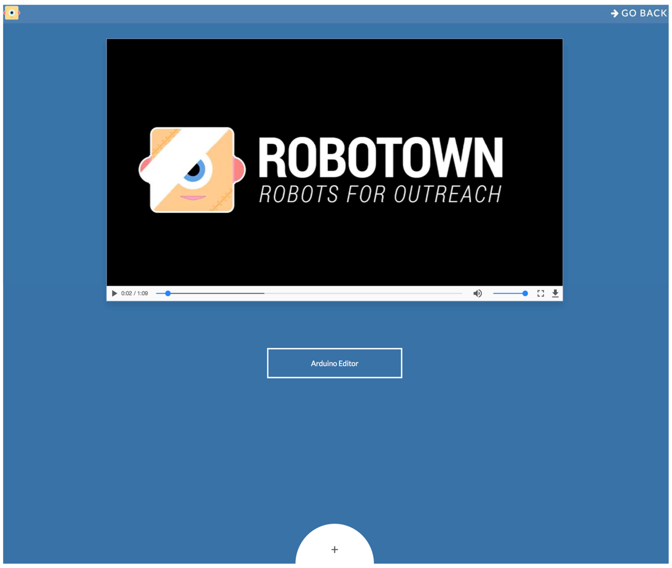
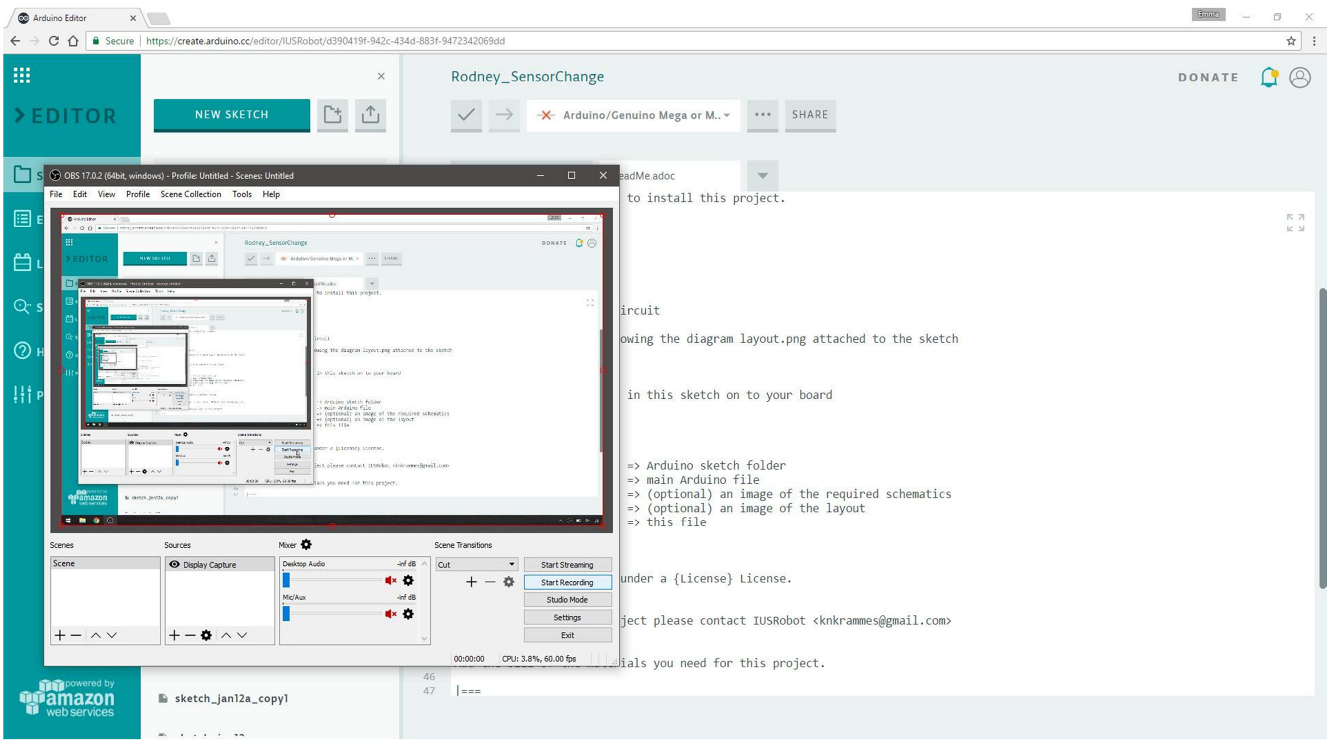


Figure 12. *Images of finished video.*

3.4. Video Lessons

The project's goal was to create engaging videos that have a personality the users can attach to and be comfortable learning from. The videos produced are comprised of a screen recording going through the code, a voice over explaining the code, and then footage of the robots running from the code that was input (Figure 12). The audio was recorded separately and then the screen recordings were made following the audio recordings. The authors used a Rode microphone[16] on a Nikon camera[17] to obtain a quality audio recording. Adobe Premiere 2017 was also used to edit the recordings. For the screens recordings, the authors used Open Broadcaster Software[18] (OBS), an open source and free recording software (Figure 13). The authors used a Canon camera[19] (Figure 14) and music from BenSound.com that sounds bright and happy. By doing this, the authors are hopeful that having the footage of the robot actually moving will help to keep the student's attention.



16 https://www.musicmatter.co.uk/rode-videomicro
17 http://www.nikonusa.com/en/nikon-products/product/dslr-cameras/d3300.html
18 https://obsproject.com/
19https://www.canon.co.uk/for_home/product_finder/cameras/digital_slr/eos_600d/

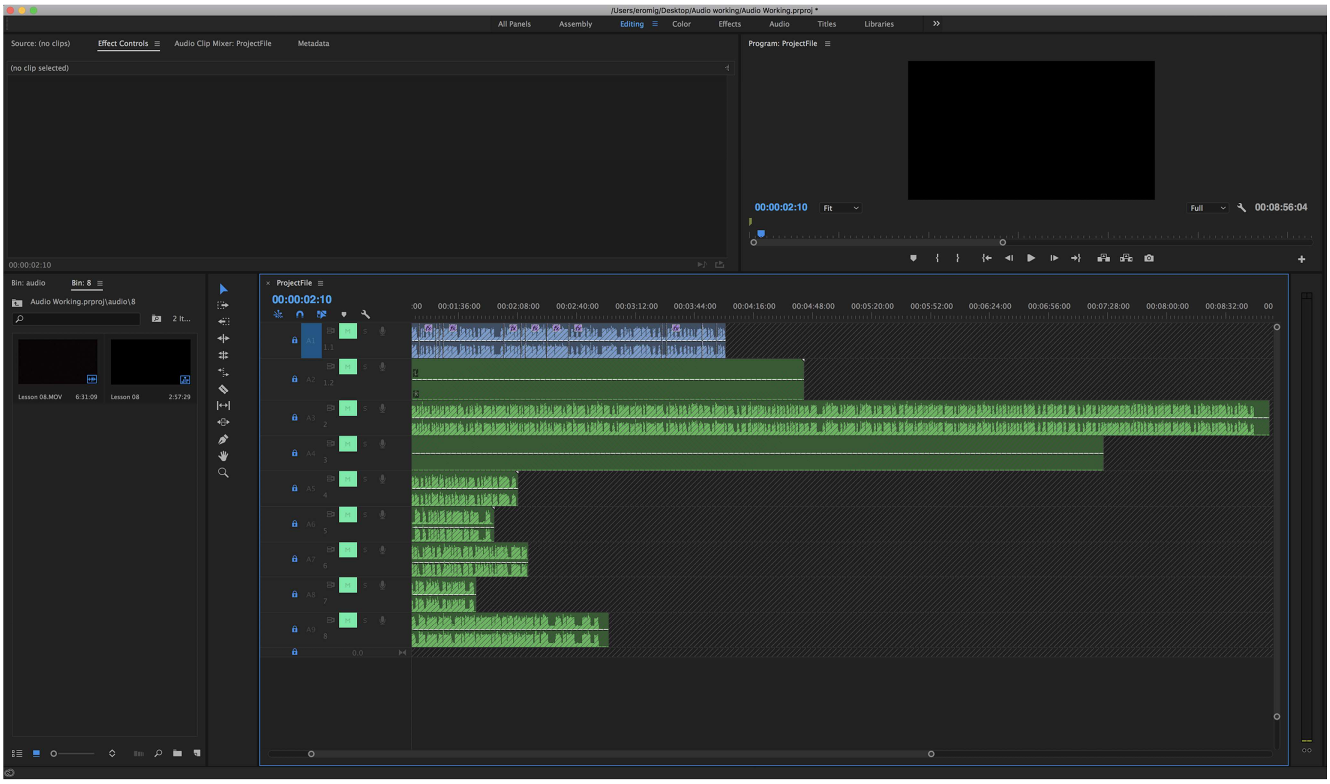


Figure 13. *Example of screen recording setup and audio file.*

4. Focus on Overall User Experience

The end product of this research venture contains a built robot, code to run the robot, lessons to show how to build and code the robot and a website to access all the material and source code for free. The robot is a cumulation of data that will allow middle schoolers to test code that they have written. The code that they will start with is included in the lessons that are provided. Moreover, sub lessons are also included where individual facets of making the robot function a certain way is also included. For example one of the early lessons teaches students how to control a LED with code. The teaching of the program code will be accomplished by the use of videos that will show the interested student what the robot will do once the coding phase is completed. Then the following video lessons will break down the code into sub lessons that will teach the students about the individual parts of the code. Taken together, the authors believe that they have made available a robust platform for robotics and electronics research and outreach.

Figure 14. *The cameras and microphone that was used for recording the robot and the voice over.*

5. Conclusion

In this research paper the authors have presented the design and implementation of an educational environment called RoboTown that was designed to be friendly and inviting to middle school children interested in learning about information systems, robotics and electronics. As discussed in the paper, the focus through-out the design process was to make the learning environment cost-effective, user oriented, useful and usable so that middle school children and their teachers can adapt and use this resource. The completed project includes a fully autonomous rover-style robot built using 3D printed material around an Arduino Mega microcontroller, along with a fully functional companion website that serves as the main source of the lessons and includes; instructional videos, interactive programming lessons, instructional resources, as well as links to download all the material for offline use in a computer laboratory.

References

[1] Ahmad, A. & Zaki, I. (2017). Automated Color Sensor System Using Ldr And Rgb Leds Controlled By Arduino. *Journal of Engineering and Applied Sciences*, *12* (19), pp. 5532-5537.

[2] Arduino Mega 2560: Retrieved November 4, 2017, from https://store.arduino.cc/usa/arduino-mega-2560-rev3

[3] Arduino Motor Shield: Retrieved November 4, 2017, from https://store.arduino.cc/usa/arduino-motor-shield-rev3

[4] Bramhe, P., Sonkusle, A., Baranwal, C., Tekade, K., Aware, P., & Jasutkar R. (2017). Coin Based Mobile Charger using Solar System. *International Journal of Engineering Science and Computing*, 7 (3), pp. 5533-5534.

[5] Breadboard: Retrieved November 4, 2017, from https://store.arduino.cc/usa/mini-breadboard-white

[6] Carmichael, Courtney C., "A State-by-State Policy Analysis of STEM Education for K-12 Public Schools" (2017). *Seton Hall University Dissertations and Theses (ETDs)*. 2297.

[7] El-Abd, M. (2017). A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions. *International Journal of Engineering Pedagogy*, *7* (2), pp. 79-93.

[8] Lokesh, G. & Sivakumar, V. (2015). Design of Robotic Exosuit For Gait Assistance In Paraplegics. *Journal of Engineering and Applied Sciences*, *10* (7), pp. 2822-2826.

[9] Murizah, K., Siti, S. & Shahrani, S. (2016). Adaptive Distance Alert Safety System (Adass) On Child Tracking Using Arduino. *Journal of Engineering and Applied Sciences*, 11 (23), pp. 14014-14020.

[10] Norris, D. (2015) The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi, and BeagleBone Black. McGraw-Hill Education, New York, USA.

[11] Nurul, H., Mohd, M. & Nurul, S. (2015). Implementation Of Speech Recognition Home Control System Using Arduino. *Journal of Engineering and Applied Sciences*, *10* (23), pp. 17492-17498.

[12] Oreski, P. & Varga, M., "*Free and Open Source Software and Raspberry Pi 3 Computer in Teaching and Learning the Coding Skills and Acquiring Digital Competences*" in Proceedings of the 9th International Conference on Education and New Learning Technologies, July 2017, pp. 150-156.

[13] Ray, P. P. (2017) A Survey on Visual Programming Languages in Internet of Things. Scientific Programming. https://doi.org/10. 1155/2017/1231430.

[14] Rose, M. A., Carter, V., Brown, J., & Shumway, S (2017). Status of Elementary Teacher Development: Preparing Elementary Teachers to Deliver Technology and Engineering Experiences. *Journal of Technology Education*, *28* (2), 2-18.