Creating a Robot that Models Asimov's Three Laws of Robotics

Aidan Hall

February 19, 2021

Introduction

In the modern world, we rely on computer-controlled systems to make our lives easier. Such systems can be incredibly complex, which can lead to oversights, resulting in failures that caused the tragic 737 plane crashes in 2019 (FAA 2019).

I believe that it could be helpful to create a simple model for how these systems could operate safely, which could potentially be used as a basis for more complex designs in the future.

This project is about creating a robot, and finding ways to integrate ethical decision making into its design, so it operates safely, as well as thinking about who is responsible for the decisions it then makes. The small scale of the project will make it possible to have a somewhat complete understanding of the full system, and so to create a basic model for how ethical (i.e. safe) computer systems can be designed.

I will use Asimov's Laws (Asimov 1950) to model the ethical system, since they are an appropriately simple model of ethics to complement my simple robot. I will also considering their limitations.

Project Aims and Objectives

1. Selecting and assembling the components of a robot capable of moving, sensing its environment, and providing feedback based on inputs.

2. Programming the robot to be able to detect and classify objects in its environment.
3. Researching robot ethics, with a focus on Asimov's Laws, their limitations, and ways of circumventing said limitations.
4. Applying this research when programming the robot, allowing it to make ethical decisions in response to commands from a human, and the things it senses in its environment.
5. Presenting and logging my work in a format that allows it to receive credit as an Extended Project Qualification.

Health and Safety

My artefact consists of a medium-size LEGO model with some electronics attached. As such, there are few real sources of danger:

- Blunt impact: The robot would be unlikely to cause injury by driving into someone due to its small size and low speed. The robot could cause harm if it fell on somebody from above. This could, at worst, cause a light bruise. I have only ever operated the robot at the height of a table or on the floor, so this risk was virtually nonexistent.
- Electronics: Electric currents can be hazardous to humans, causing electric shock or even death. However, according to the Health and Safety Executive (2021), 50V is needed to cause an electric shock. My battery pack has a series voltage of 4.8V (at most), so it is safe to use without significant precautions.

Help from Others

Aside from financial support and feedback on my presentation from my family, and advice from my supervisor (documented in the production log), I required no major help from others for the actual production of this project.

Using the Laws

Asimov's *Three Laws of Robotics*, as stated in *I, Robot* (Asimov 1950) are,

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Since their creation, the Laws have come under scrutiny due to numerous shortcomings. It is therefore important that I first establish what these shortcomings are, and then consider how I will use the Laws despite them.

A major criticism of the Laws is that they are too ambiguous. This is understandable since they are meant to be applicable to every robot, so they can't mention anything that would be unique to a particular domain of robotics. A commonly proposed solution is some form of extension to the laws. This may take the form of small (yet subtly significant) additions to or clarifications of the original 3, such as the 0^{th} Law (Asimov 1985) or Clarke (1994)'s extended set, referenced by McCauley (2007). However, I believe that these only add complication to the Laws, and do not greatly improve the ease with which they can be applied.

A maximal approach to extending the Laws can be found in the '10 Principles of Robot Law' (Field 2010). In this, I feel that some of the beauty of Asimov's Laws is lost, in that the specificity of some of these principles means that they simply won't be applicable to many varieties of robots.

Furthermore, it is unlikely that any such extension would be relevant to my robot, beyond the simple assertion that 'it would be completely impossible for my Robot to break a given Law, so it is acceptable', so I will not consider any Laws other than the original three for my project, since it is meant to be a simple, general model for all robot ethics.

Complaints are also raised about the fact that it is practically impossible for a robot to either understand whether its actions comply with the Laws, or have the required context to decide if this is the case, especially with the addition of the 0^{th} Law (Asimov 1985), which extends the 1^{st} Law to concern humanity as a whole. As a result, I believe that the 'Roboticist's Oath' proposed by McCauley (2007), which implies that manufacturers design robots to comply with the Laws within the context of their capabilities and awareness is the best solution; the robots themselves are not programmed directly to follow the Laws, but merely to act in pre-determined compliant ways. Due to the comparative simpleness of this approach, it is the one I used.

In the Robots series, the Laws are defined as mathematical equations which are hard-coded into the robot's brain, and the textual form we are familiar with is simply used to explain to humans how the robots behave as a result. As such, I need only demonstrate that the code of my robot will result in it obeying the Laws, rather than designing it to visually or structurally resemble their written expression.

Hardware Construction

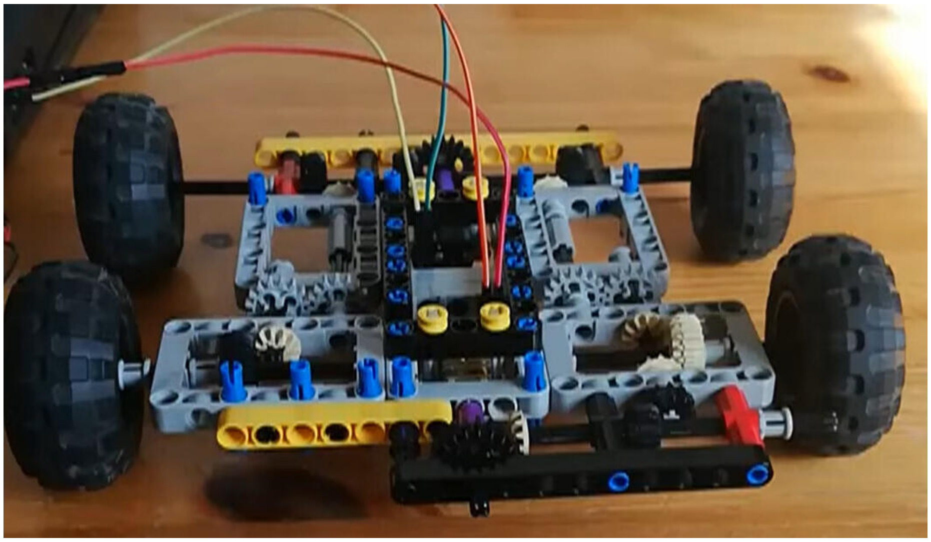

Figure 1: The first chassis, which was fragile and over-complicated.

Micro-controller

In the design of the robot, I have aimed, where possible, to use the simplest solution to any given problem.

First, I chose to use the Raspberry Pi 3 over a lower-level micro-controller such as an Arduino because it would need the ability not only to interact

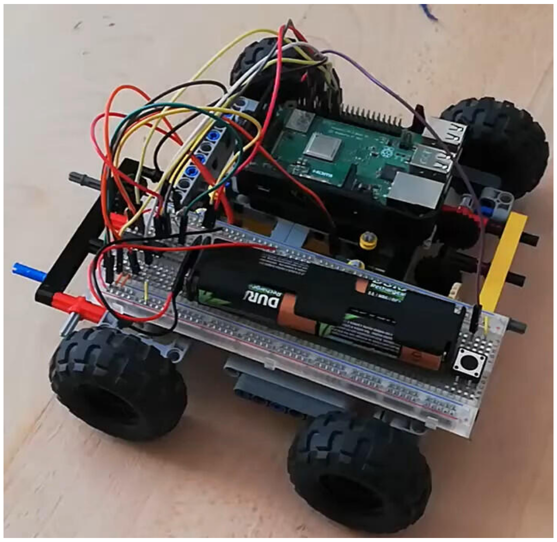

Figure 2: The robot running from an on-board power-supply.

with motor controllers and other low-level components, but also to interface with a camera and Bluetooth I/O devices, and to run processor-intensive object-detection code.

I have chosen not to use any Raspberry Pi 'hats' (small PCBs created to simplify interfacing with hardware) since using bare components and a breadboard gives me more control and allows the robot to be lighter and cheaper.

Power

The Raspberry Pi requires a 4.63V, 2.5A power supply (Raspberry Pi Foundation 2021). Most of the common methods for on-board power involve a USB power bank or some form of Raspberry Pi hat which regulates a source (Saville 2020). These were all too complicated, so I decided to use a battery attached across the GPIO pins, as demonstrated on the Raspberry Pi website (Barnes 2017), and shown in figure 2. Four 1.2V AA batteries in series provide 4.8V, which is sufficient to power not only the Raspberry Pi (being above 4.63V), but also the motors, which operate at 3–7V (Kingly Motor Co. 2014), and the rest of the components of the robot.

Parallel 'loops' of a circuit all have the same potential difference across them, so connecting the battery, the motor power input, and the Raspberry Pi in parallel allows the motors and Raspberry Pi to receive a sufficient voltage.

Movement

I decided to have my robot move similarly to a rover, with its wheels on fixed axles and the ability to turn on the spot, since this would be a simpler system to construct. I also wanted it to move moderately slowly, so that it would be easy to control and able to react to stimuli (such as the presence of a human) even if it took time to respond. Using these criteria, I chose a pair of 298:1 gear-ratio brushed DC motors, for their affordability, compactness and power delivery.

Motor control is a case where Raspberry Pi hats are commonly used. I researched motor controllers, and found that most hats came with features I was unlikely to use or need, such as IR receivers, that increased their cost. In addition, several of the ones I considered had the same motor controller chip (Corteil 2016, p. 17), the `TI SN754410`. As a result, I decided to buy one of these instead, since it was significantly cheaper than a hat and would provide all the functionality I needed.

Chassis

Since I only had two motors (and a control chip that was only capable of driving two), I had to design a gearing system that allowed each motor to drive both wheels on its assigned side. Due to my chosen drive system, only having two motors does not limit how my robot can move, beyond the power delivery obviously being half that of a four-wheel drive. This went through a few iterations which can be seen in my progress log here. Due to the unavoidable friction between LEGO axles and gears, I determined that the most effective design, seen in figure 3, would minimise the combined length of the axles and number of gears.

I mounted the components to the chassis so they stayed in place whilst in motion, with the camera attached to a raised stalk to increase its field of view and keep most of the chassis hidden from it.

The robot also has an LED which is used for feedback, representing a laser that it can use to ‘attack’ harmful things.

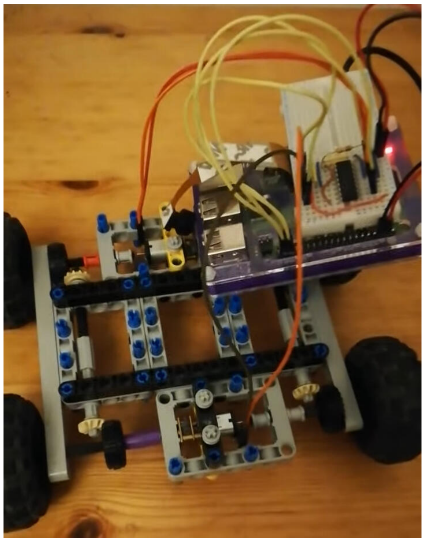

Figure 3: The final chassis, which was simpler and stronger.

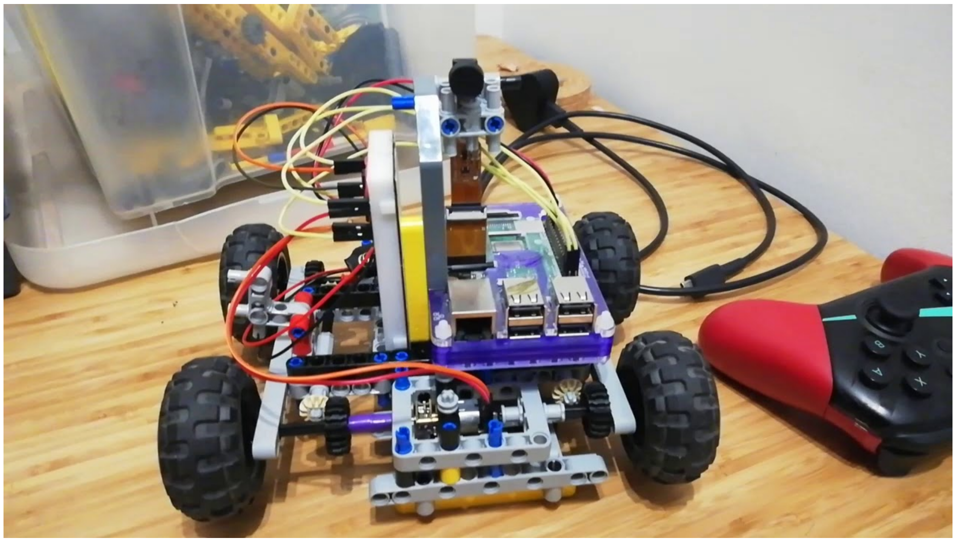

Figure 4: A near-complete version of the hardware, with the camera mounted.

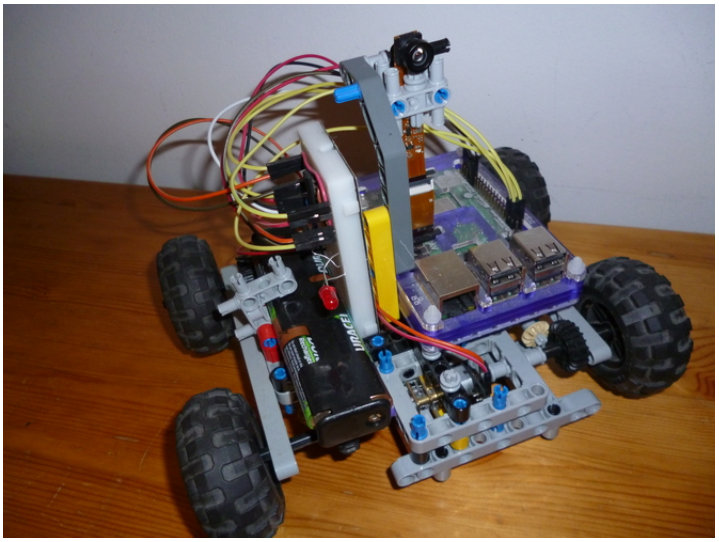

Figure 5: The finished robot, with the LED and battery pack.

Input and Environment

Manual Control

I had originally envisioned controlling the robot via voice commands, and having it autonomously navigate its environment. However, since this seemed overly complex and tangential to the focus of the EPQ, I instead elected to control the robot via a Bluetooth game controller, and to give it the ability to choose to act differently based on other stimuli.

Image Recognition/Object Detection

The only other input the robot receives comes from the camera. The video feed must be processed in order to be useful for my program. For the purpose of modelling the Laws, the robot needs to be able to identify:

- Sources of harm to humans (for the 1st Law).
- Humans (for the 2nd Law).
- Sources of harm to itself (for the 3rd Law).

I had a few choices for the Object Detection implementation, shown below.

Tool	Pros	Cons	Language
OpenCV	Fast and efficient	Too low-level	C++
TensorFlow	High-level	Too complicated	Python
TensorFlow Keras	Extremely high-level	Slightly restrictive	Python

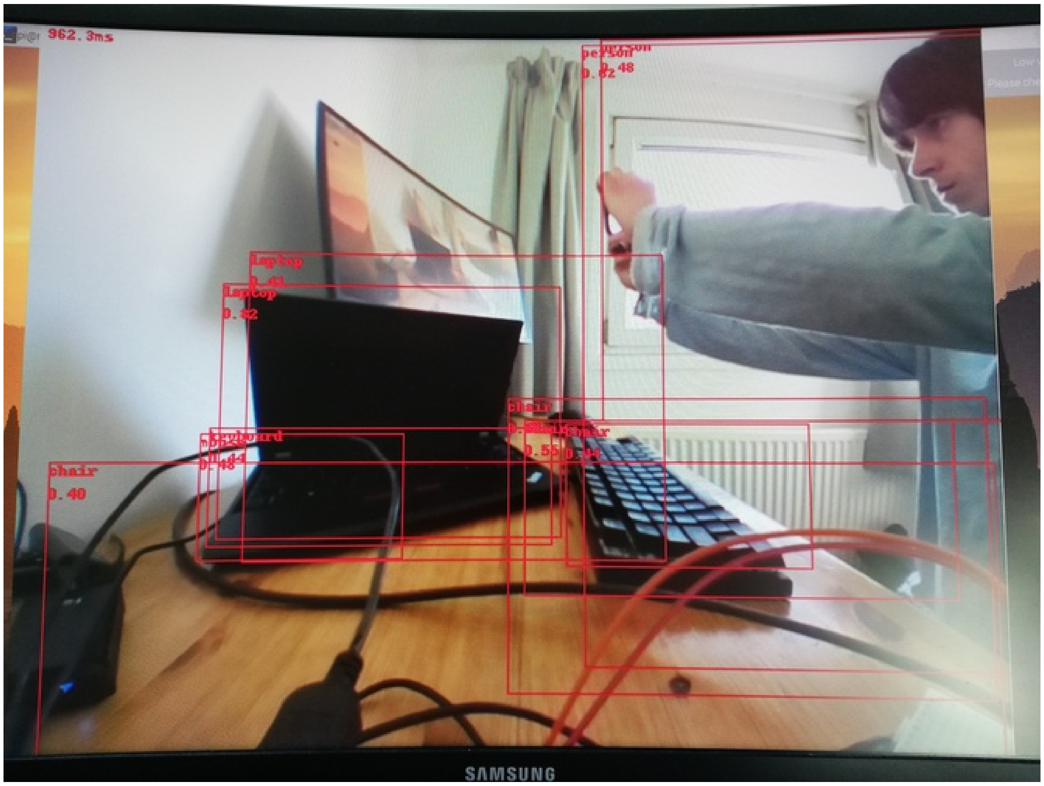

Figure 6: The object detection example identifying my laptop and me.

I attempted to create an image classification model for LEGO pieces using TensorFlow's Keras library, as this seemed like the easiest and simplest approach to take, but even that proved too challenging so I have instead used a Raspberry Pi object detection model example created by TensorFlow and hosted here. It is capable of identifying humans and common household objects, so it suits my needs perfectly. I have decided to designate electronic devices (such as phones and laptops) as potential sources of harm to humans *and* robots for this project, since the model can easily identify these.

Programming

The robot's program is large and complex, so its development merits description. Since the Raspberry Pi 3B+, the computer I used as the controller for the robot, has a quad-core processor (Raspberry Pi Foundation 2020), I designed the program to make use of three threads of execution, with the fourth being left for the operating system.

1. Object detection: The pre-made example. Produces a list of objects that are visible, in the form of an array of string identifiers.[1]

2. Visibility tracking and classification: The list of currently visible objects is compared to a previous one to determine which objects have moved in or out of view. If any of the visible objects are deemed to be harmful or human, an appropriate variable is set to indicate whether one is visible.

3. Decision making and robot control: Using the current input from the game controller, and the indicator variables for humans and sources of harm, this thread determines how the robot should move and whether it should 'fire its laser'.

The robot's program follows a pipeline model, where the data produced by one thread is pushed into a queue, which the next thread then reads from the front of. This structure allows data to be transferred between threads whilst avoiding the nightmares of ***shared mutable state.***

Ethical Decisions and Responsibility

The robot I have produced is very different to the ones in Asimov's stories, mostly because of its near complete lack of autonomous capabilities. As such, since it is clearly not conscious, we cannot ascribe any responsibility for its behaviour to the robot itself.

[1] In reality, the object-detection example creates many program threads itself, so this tidy model is just for ease of explanation.

Instead, as I explained in Section 2, all responsibility must be placed upon the designer, creators and person controlling it (me, me and me). Given its lack of autonomy, any actions it takes under the command of a human (whilst obeying the Second Law) are entirely the responsibility of the human in question. The responsibility is then shifted to the creators when its ethical programming is supposed to activate.

For example, if it detected a potential source of harm to humans and failed to act, within its own contextual awareness that could be a violation of the First Law. In this case, the one who could be deemed responsible would depend on the cause of the failure to recognise the danger. If the object detection model failed, it would partially be the fault of the model's creator, but also my own for using the inadequate model. If, however, the robot's program did not include a suitable response for that danger source, I would be entirely responsible as the sole creator of that program.

Ethics in Action

It is hard to demonstrate the robot in action through images and text so I refer the reader to the last few videos (especially number 11 and number 12) in my EPQ Robot playlist.

Instead, this table enumerating the behaviour of the robot according to a human's commands and the objects it detects will have to suffice:

Human Visible	Danger Visible	Command	Action	Law(s)
Yes	Yes	None	Doesn't fire.	1^{st}
No	-	Any	Obeys command.	2^{nd}
No	Yes	None	Retreats	3^{rd}
Yes	Yes	Retreat	Stops and shoots.	$1^{st} > 2^{nd}$
Yes	Yes	None	Stops and shoots.	$1^{st} > 3^{rd}$
-	Yes	Advance	Advances.	$2^{nd} > 3^{rd}$

Conclusion

From the decision-making table, it can be determined that the robot's decisions do model Asimov's Laws. As a result, the robot will attempt, under the modelling assumptions of this project, to operate in a way that keeps itself and humans safe.

This model could now be built upon, by adding more commands for the robot to respond to, or more objects and categories of objects for it to recognise, to create a system that could potentially operate in the real world. This could be done whilst making sure that each new behaviour added still complied with the Laws. As a result, with Asimov's Laws as a basis for how it operated, regardless of how complex it became, its behaviour would still follow the predictable patterns the Laws impose. Consequently, it would be more likely that the system would operate safely, since the basis for its operation would remain simple enough to understand that any errors could be quickly detected and alleviated.

In conclusion, my robot models Asimov's Laws, and could be used as a basis for designing safe robots in the future.

Objective Satisfaction

Here I will attempt to demonstrate how my project satisfies my objectives.

1. The robot is capable of moving with the motors. It can sense its environment using its camera and object-detection model. It can receive commands or inputs from a human via a game controller. It can provide feedback either by moving or deciding whether to 'fire its laser'.

2. As demonstrated in my development videos, the robot can *detect* a variety of common household objects, and can *classify* them as either human or harmful (technology) as appropriate.

3. I have researched limitations of Asimov's Laws, such as how they are vague and hard to comply with, and have been able to circumvent them in a few ways:

   - Difficult Compliance: By making myself responsible for compliance, the robot itself doesn't need to 'understand' the abstract concepts the Laws involve, since it can simply be programmed in a way that results in compliant behaviour.

     This is also overcome by the observation that the robot only needs to be designed to comply with the Laws within the limits of its capabilities and awareness. Since my robot has such a limited degree of awareness of its environment, that being just what it can currently see, it is comparatively easy to ensure that it complies.

- Vagueness: By restricting my robot to a simple model capable of only a few actions (move and shoot), I can consider each specific action individually, rather than having to implement a general algorithm to apply to all eventualities that would be more subjective and potentially inaccurate.

4. By using those previously specified constraints, I have been able to write a decision-making program for the robot that considers the *presence* of humans and sources of harm using the results of the object-detection algorithm, as well as the *commands* of humans from the game controller inputs, when determining which actions to take.

5. Since this objective directly concerns the success of my project, I am not the one to judge this.

Bibliography

This is why I used LaTeX; this bibliography was generated automatically!

Asimov, I. (1950). *I, Robot.* Harper Voyager, p. 1. ISBN: 978-0-00-827955-4. London.

Asimov, I. (1985). *Robots and Empire.* Doubleday Books.

Barnes, R. (2017). *Power your Raspberry Pi: expert advice for a supply.* (Accessed: 14/04/2020). URL: https://magpi.raspberrypi.org/articles/power-supply.

Clarke, R. (1994). “Asimov’s laws of robotics: Implications for information technology.” In: *Computer.* URL: http://www.rogerclarke.com/SOS/Asimov.html.

Corteil, B. (2016). “Build a Remote Control Robot”. In: *The MagPi Magazine.* (Accessed: 27/11/2019), pp. 14–27. URL: https://magpi.raspberrypi.org/issues/51.

FAA (2019). *Continued Airworthiness Notification to the International Community.* (Accessed: 26/08/2020). URL: https://www.faa.gov/news/updates/media/CAN_2019_03.pdf.

Field, C. (2010). *Japan's "Ten Principles of Robot Law"*. (Accessed: 11/08/2020). URL: https://akikok012um1.wordpress.com/japans-ten-principles-of-robot-law/.

Health and Safety Executive (2021). *Electrical injuries.* (Accessed: 18/02/2021). URL: https://www.hse.gov.uk/electricity/injuries.htm.

Kingly Motor Co., Ltd (2014). *JL-12FN20-298-0675 motor datasheet.* (Accessed: 06/04/2020). URL: https://cdn.shopify.com/s/files/1/0174/1800/files/JL-12FN20-298-0675_XG2014111403-Model.pdf.

McCauley, L (2007). "AI Armageddon and the Three Laws of Robotics". In: *Ethics and Information Technology* 9.2, pp. 153–164. URL: https://doi.org/10.1007%2Fs10676-007-9138-2.

Raspberry Pi Foundation (2020). *Raspberry Pi 3 B+ Product Brief.* Tech. rep. (Accessed: 17/02/2021). URL: https://static.raspberrypi.org/files/product-briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf.

Raspberry Pi Foundation (2021). *Power Supply - Raspberry Pi Documentation.* (Accessed: 15/02/2021). URL: https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md.

Saville, R. (2020). *10 Ways to Power Your Raspberry Pi.* (Accessed: 21/09/2020). URL: https://www.lifewire.com/ways-to-power-your-raspberry-pi-4092246.