A Prototype Robot as an Example of Creative Repurposing of Accessible Technologies

Sadia Afrin
3rd Year Undergraduate Student
Manukau Institute of Technology
afri9@manukaumail.com

John Calder
Senior Lecturer
Manukau Institute of Technology
john.calder@manukau.ac.nz

ABSTRACT

This paper presents a prototype for a “robot” device to demonstrate that advanced functionality is achievable by utilising even a low-end smartphone’s capabilities. Through a standard client-server architecture implemented using HTML, JavaScript, .Net Core and SQL Server, the robot achieves five innovative functions: remote control, location tracking, video streaming, motion detection, and speech recognition. The robot program integrates commonly available Web APIs and phone functionalities through the browser on the smartphone. This paper also outlines a series of experiments to test whether the robot performed the new tasks to a usable level. The final prototype displayed performance that exceeded expectations for modest low-cost equipment and rivalled or even equalled that of high end devices. The solution’s front-end is based entirely on HTML and JavaScript running in-browser, and so it is platform independent and the smartphone stays unaffected. The project’s architectural simplicity and modular design is ideal for hobbyists and students to learn technological concepts.

Keywords: Repurposing, Smart Device, Smartphone, Accessible Technology, Location Tracking, Video Streaming, Motion Detection, Robotics, Speech Recognition, Android, Client-Server, .Net Core, Creativity

1. INTRODUCTION

Companies and individual hobbyists have devised several innovative solutions by which they can utilise these devices or their components to perform tasks beyond their design. They do this through “creative repurposing” which refers to utilising an object for a purpose different to its original intent. For example, a smartphone camera’s major purpose is to take photos. With creative repurposing, a user can use the focusing feature of the camera to measure proximity. This would allow the camera to be useful in a variety of applications such as environmental monitoring, volume measurement, and motion detection (Hoang & Niyato, 2016).

Among the different types of smart devices, smartphones have attracted the most interest from companies and enthusiasts intending to repurpose (Benton, Hazell, & Coats, 2015). This paper outlines the design and implementation of a mobile “robot”. It puts focus on utilising the major features of a relatively low-spec and low-priced smartphone. The paper aims to demonstrate that it is relatively easy to combine a variety of existing features present in a smartphone to create an original device with an array of innovative new functions.

2. WHAT IS ACCESSIBLE?

Early in this project we investigated the attractive idea of productive use of discarded obsolete smartphones as robot brains. We found that highly capable low cost smartphones outperform obsolete ones to a very large degree. Time spent with the old phones is time that is too much taken up with learning obsolete programming methods which do not give the learning and teaching spinoff benefits that we get from use of a new phone. The "Vodafone VFD-300 Smart Mini 7" has an official cost of $49 and appears in sales in NZ for as little as $29. It runs the Android 6 operating system which was the most recent OS version during this project. The VFD-300 can run current and latest beta versions of the programming environments of interest. Work created on the VFD-300 runs on a wide variety of related devices giving a standardised and repeatable approach. One may buy an online auction bargain which gives good prototyping value, but then when team members and peers want to build on the findings the process of shopping for the same or similar disappearing hardware becomes an unwelcome barrier. There are similar issues with sifting through waste and discarded materials for parts.

We had a similar experience with efforts to build robot bodies out of parts and materials found in common supermarkets and hardware stores. Progress was slow with remarkably few useful items discovered. Promising items like kitchen gadgets were subject to rapid stock turnover with design changes.

We suggest that recycling needs to be materials recycling and new developments need to use accessible new materials. Our progress accelerated when we changed to that approach which included items like model aircraft parts, electronic parts and mechatronic parts from suppliers servicing "maker" enthusiasts and hobbyists.

3. IDENTIFICATION OF ISSUE

This project takes the approach of practically demonstrating how the repurposing project can be creative and what to do to harness a device’s full potential. Utilising attributes of a smart device with a greater level of efficacy than is usual in repurposing projects can help to create such a prototype. Therefore, the design needs to:

- Utilise device processing power effectively.
- Utilise device /output capabilities to a large extent.
- Exploit device programmability.
- Harness already provided software and operating systems within the device.

This quality assured paper appeared at the 9th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2018) and the 31st Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, July 11-13 during ITx 2018.

4. PROPOSED SOLUTION

Easy availability, programmability, portability and low cost of a smartphone makes it the target device the project is based on. The most common/obvious features of a smartphone such as its network card, the operating system, installed apps, camera, and speakers are free to utilise to achieve the solution.

Furthermore, easy to obtain cheap motors and electronic components can allow the smartphone to become mobile, and the high availability of the internet would allow the system to be accessible from a distance.

This project takes on the name: "Creative Repurposing of Accessible Technologies" or "CRAT". The name refers directly to the three themes (Creativity, Repurposing and Accessibility) that were adhered to while creating the prototype. The word "Creative" encourages finding the solution with novel approaches. "Repurposing" refers to utilising components beyond what their original design dictates. The word "Accessible" implies using readily available components and features that are well within reach of the technically savvy student or hobbyist.

5. OBJECTIVES

This project aims to create a prototype of a device that a human commander can control to perform a number of useful functions. It must utilise many of the existing functionalities found in a typical smartphone. The original device will remain untouched at a low level and will be useable as a normal smartphone when taken out of the project's setting or hardware.

5.1 Mobilisation and Remote Control

The smartphone must travel using readily available electronic and mechanical parts. The smartphone becomes open to new, previously unplanned use cases by allowing it to move, and making it controllable remotely. Since the smartphone has no feature that allows it to move, construction of a movable housing for the phone is necessary.

5.2 Location Tracking

The device must have the ability to provide a constant update on its location. This provides a level of confidence if the phone travels out of the controller's sight. The GPS (Global Positioning System) module of the phone can provide this information.

5.3 Live video streaming

The device must allow the controller to view the remote location through the phone's camera as a live video feed. The device should be movable while the controller views the feed, transforming the smartphone into a remote exploration tool.

5.4 Motion detection

The smartphone must be able to monitor an area for movement and record any motion with snapshots taken from the phone camera. The mobility of the device, combined with this detection feature, allows free positioning of the phone to monitor hard-to-reach places.

5.5 Speech recognition and vocal feedback

The smartphone must be able to communicate vocally at a rudimentary level using prebuilt responses to a predefined set of questions. This objective aims to demonstrate the potential of the device to serve as an entertainment/helper robot.

6. METHODOLOGY

John Calder, Senior Lecturer at Manukau Institute of Technology, at Manukau Institute of Technology laid the groundwork for this project by constructing a fully functioning robot housing circuitry and establishing the communication infrastructure. He created the housing to accommodate an inexpensive smartphone (Vodafone Smart Mini 7) and succeeded in making the robot move by controlling it remotely. He then initiated this project to enhance the existing hardware and software setup.

This paper mostly reports on work done as a student project by Sadia Afrin supervised by John Calder. Sadia Afrin had a timeframe of eight weeks for her investigations adding value to John Calder's existing prototype which was achieving Mobility and imprecise Remote Control. Taking into consideration the time it would take to create the project proposal and write other documentation, the feature implementation phase needed to have a cap of only four weeks. Independently tackling each requirement would give the project the best chance of success. This was to ensure that the project would have at least some requirements completed by the deadline, if not all of them.

6.1 Prototype Architecture

The robot unit needed to be controllable by a human operator and needed to send back videos, images, and other information. The human operator devices were laptops. To achieve distant remote control over the Public Internet, both devices need to run in "Client" mode. There are high barriers to running server software on the endpoint devices e.g. most if not all cell phone networks will block communications for server-side software running on smartphones. We know from our own trials that Vodafone NZ does this. Therefore, our infrastructure is set up so that the two endpoints can send and receive information by submitting and requesting messages to/from a server. Figure 1 illustrates this architecture. This pattern is commonly known as a publisher-subscriber pattern (Cao & Singh, 2004).

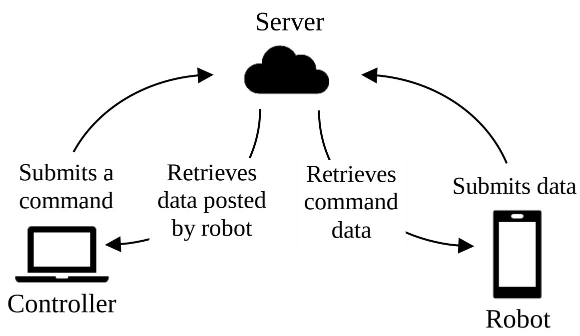

Figure 1: Client-server architecture diagram

The controller application sends a text command (e.g., to move the robot left/right, start or stop video streaming) to the server by using this pattern. The server stores this command in a relational database table. The server also sends back a response to the controller containing any data that the robot has previously posted (e.g., GPS location or video file). The controller then displays this data to the user (e.g., by playing a video or plotting GPS coordinates on the screen).

On the other side, the robot's smartphone application submits the data it has gathered (e.g., GPS coordinates or video captures), to the server. The server stores it and responds back with any commands that the controller had previously posted (e.g., move left/right). The robot then functions according to these commands (e.g., by moving left/right).

6.2 Research Approach

As mentioned previously, the emphasis on finding the simplest solutions to the requirements led to extensive research on readily available APIs and technologies. The quality of the APIs combined with the quality of the hardware of the smartphone should determine the quality of the final prototype.

6.3 Technology Stack Evaluation

The available technologies were freely or easily available and enabled high-level programming across the range of programming environments.

6.3.1 Backend: SQL Server

Microsoft SQL server allows sending messages and data *by* saving them in a relational format and enables more than one robot/commander pairing to happen through the same server. The choice of this database fits with the Microsoft stack of technologies that are utilised for this project. However, the data and scripts are portable to any other relational database management system supporting the SQL syntax (Turley & Wood, 2009).

6.3.2 Server: ASP .Net Core

ASP .Net Core is a programming framework used to create web applications to run on web servers. This project uses it to implement the logic of the publish-subscribe pattern and to retrieve/send information to and from the database. This framework has the added benefit of being able to run on any operating system, ensuring that it doesn't restrict potential future adopters to a Microsoft specific infrastructure (Chambers, Paquette, & Timms, 2016).

6.3.3 Robot Client: Android

Android devices are generally available at a low cost while providing satisfactory performance. It is the world's most popular mobile operating system, and therefore more technical assistance should be available for it than with other alternatives.

The major disadvantage is that due to the low price of the device, its battery may not last as long or the processor may not perform tasks as quickly as more expensive devices.

6.3.4 Programming Languages

The project adheres to only an essential set of languages to keep the language ecosystem for the project as minimal as possible. JavaScript handles all client-side programming needs (Github, 2017). HTML was the ideal choice for the frontend display language on both the robot smartphone and controller. Given the choice of ASP .Net Core as the server-side framework, the language used to program the server is C#.

7. IMPLEMENTATION

The unit's required features all rely on the client/server communication channel established with the help of the technologies mentioned. Despite there being an indirect connection, the latency should not be unsatisfactory given that average network latencies within New Zealand are less than 20 milliseconds (Truenet, 2017). Another advantage is that the smartphone and the human commander apps simply need to connect to the internet which allows them to use the most readily available network (such as mobile internet, Wi-Fi or Ethernet). As soon as either device connects to a network, it can make independent requests without relying on the status of the other device. This prevents the complexity of managing a real-time direct connection.

A limitation of this approach is the need for a polling mechanism in both apps to periodically grab information from the server. This could lead to a transfer of unnecessary network data if neither device has any useful information to send/receive. The polling period needs to be frequent (0.2 sec) to pass near-real-time movement and information updates between devices. We have coded a "long polling" solution where at times of no change detected the period can slow down to a maximum interval of 4 sec.

Both the human commander unit and the robot unit used simple HTML pages to send/receive commands. Due to the high availability and popularity of the languages that form the web ecosystem (HTML/CSS/JavaScript), they were the best choice for a project focusing on ease of accessibility to the technology.

Leveraging asynchronous HTTP POST requests allowed sending and receiving data to the server from each page. The Hypertext Transfer Protocol (HTTP) is a system of rules that allows transfer of messages digitally to a web server. The "POST" method carries data to the server through a response-request mechanism, where a client (the robot or controller page) sends a request with data, and the server then responds with a status update.

Our planned developments include changing from HTTP to the newer "Websockets" protocol which is designed to better meet real-time communication needs.

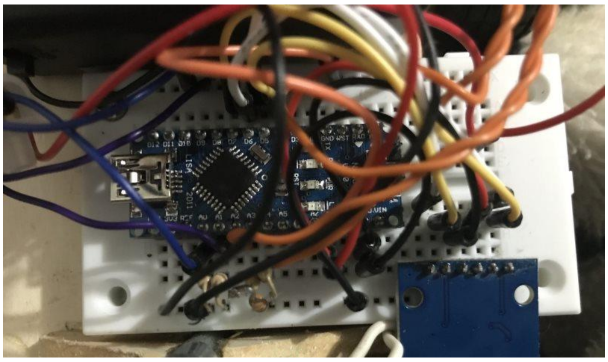

Figure 2: Robot body circuit closeup

Usually, the screen of a smartphone displays information to a human user for their consumption. This project utilises the smartphone screen as an output device. The circuit is an easily obtainable Arduino board connected to four LDR light sensors and a pair of motors. Each of the light sensors is adjacent to a light patch on the smartphone's screen. When the light patches activate (turn light or dark), the light sensors respond. When the Arduino board recognises this activation, it spins the motors accordingly. This moves the robot's wheels and allows it to travel. Figure 3 illustrates the flow of control from the light patches to the motors.

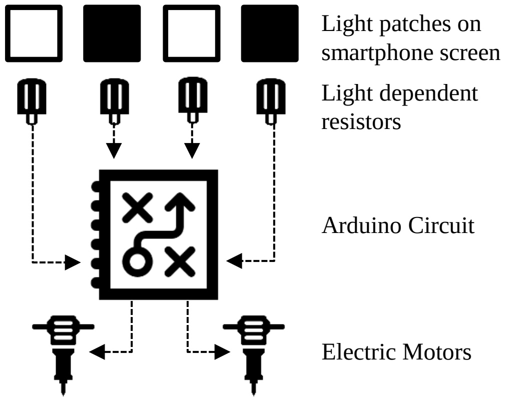

Figure 3: Robot body circuit diagram

Controlling the light patches on the smartphone is the responsibility of the human commander application (HTML page).

7.1 Movement with Game Controller

By connecting a commonly available Xbox game controller via USB to a laptop, and by leveraging the GamePad API provided in the Chrome Browser, the webpage was able to pick up the full array of button presses and joystick movements (Mills, 2017). The human commander page picked up only one of the joystick controls of the gamepad to keep the project simple. The output of the joystick was very sensitive to the slightest touch, and so a predefined threshold prevents the robot from moving unless the user explicitly moves the joystick by a significant amount (50% of the range of the joystick).

Also coded was a mouse interface for when the game controller was not available.

7.2 GPS

The phone's GPS unit allows access to the device's location by apps. In this project, the HTML5 Geolocation API provided a trivial way to access the device coordinates from the HTML page using JavaScript, without needing to create any native applications (W3Schools, n.d.).

The existing code of the robot page polled the server with back-to-back HTTP POST requests to obtain movement instructions from the controller. By leveraging this POST request with additional data containing the coordinates of the robot, the robot page successfully sent its location to the server. The server simply updated a pair of latitude and longitude records related to the robot in the database.

The human commander app, as part of its usual polling, picked up the additional coordinates recorded against the robot's record in the database and displayed them on a Google map on the page.

7.3 Motion Detection

Video: https://www.youtube.com/watch?v=uBGgNHqDlz8 (Afrin, CRAT Motion Detection, 2018)

The typical use of the smartphone camera is for a user to take photos and videos. This project utilises the camera differently, by receiving a live video feed and processing the frames to detect whether there was any motion.

Firstly, using the WebRTC (Web Real-Time Communication) API provided in the browser, the HTML page captures a video from the phone camera. As every frame of video arrives on the page, a third-party library called "DiffCamEngine" compares the current frame against the previous frame and checks for differences in pixel colour value (Boyd, 2016). This comparison yields a numerical result, and if this result is above a certain threshold, it triggers a photo upload to the server. The uploaded photo is essentially the captured frame of the video which caused the high numerical output.

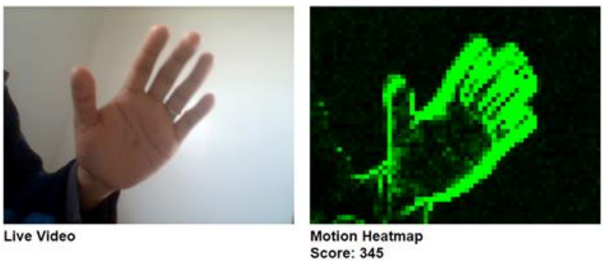

Figure 4: Visual demonstration of motion detection algorithm output

The uploading process saves the photo on the server's file system grouped by date and is viewable from the controller page.

The main challenge of this motion detection system is to be able to determine an optimal sensitivity threshold. A threshold which is too low will yield false positives, and a threshold which is too high will cause the robot to miss valid movement patterns.

7.4 Video Streaming

Recording videos is a very common use of a smartphone camera, but in this project, the robot utilises the camera and microphone to stream videos to the controller page.

The WebRTC API provides functionality for streaming video over a network. To allow video streaming to function with common connections to the public internet would require setting up a TURN server at extra cost. A TURN server allows two devices on different networks to connect directly and send data between each other (Hätönen, et al., 2010).

Our approach was to emulate video streaming by programming the robot to send chunks of video of approximately one-second length to the server and programming the controller to download them one after another. The WebRTC functionality provided the ability to record the video chunks, and they reached the server via HTTP POST requests.

This second approach adheres to the theme of simplicity, creativity and resorting to only readily available technology to achieve our project goals. This makes it the ideal approach to take for this project, even though it may not be as performant as the first.

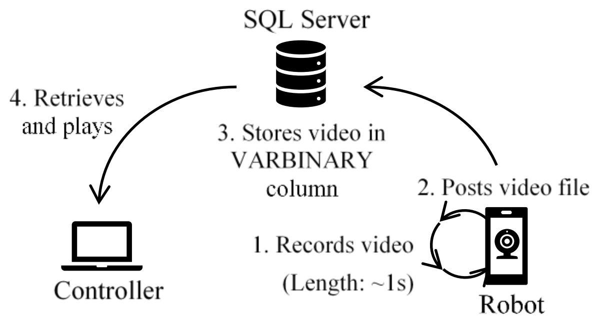

Figure 5: Video streaming data flow

7.5 Speech Recognition

Video: https://www.youtube.com/watch?v=QoeTP63kAfQ (Afrin, CRAT Speech Recognition, 2018)

The SpeechRecognition API provides the logic for our speech recognition trials. The SpeechSynthesis API provides the vocal output capability (abbycar, 2016). Both APIs utilise the native speech recognition and speech output capabilities of the device from within the browser. When the human speaks into the microphone, the SpeechRecognition API outputs an array of possible phrases in plain text (Ponomarev, 2017). By coding with JavaScript, we enable the robot to check for certain phrases that the API has recognised and chooses from a list of predefined matching responses. It then picks one response and calls the Speechsynthesis API with the chosen response. This makes the robot speak the response out loud. While the robot talks, the SpeechRecognition engine pauses so that it does not pick up its own speech.

8. SECURITY CONSIDERATIONS

There is no personal data collected from any device without the permission of the user. Additionally, the Chrome browser asks the user explicitly when it needs to access the location, camera or microphone.

9. EXPERIMENT PROCEDURE AND DATA

A series of experiments tested the capabilities of the unit under real world scenarios with various relevant factors. These experiments assessed each requirement individually with multiple trials and controls where possible. There was a restricted amount of time and effort placed in testing the device's performance, resulting in high error allowances and low trial numbers. Holding further trials under a greater variety of network and environmental conditions would reduce error rates and maximise test reliability.

9.1 GPS Accuracy Test

Placement of two other higher end smartphones (A Google Nexus 5X and an iPhone 7) alongside the robot unit, allowed comparison of the accuracy of the robot's GPS location. Then all three devices were moved to four different sites to compare their latitude and longitude readings at the same time and the same location. The human commander app remotely observed the robot's latitude and longitude readings, while the readings from the other phones were recorded manually from observing the phone's output.

Table 1 outlines the results of the test. The precise coordinate values fluctuate very rapidly due to a lack of dampening on the phones, but one random sample coordinate reading, from each device, at each location, was sufficient to show that all 3 devices were giving almost identical results. Figure 6 shows the coordinate readings from the three devices plotted on a map for each site. The coordinate readings from the three devices all fell within 2-3 meters of each other.

Table 1: GPS Coordinate Accuracy Test Results

Location	IPhone	Android	Robot
Site 1	-37.030528, 174.862701	-37.030549, 174.862686	-37.030533, 174.862646
Site 2	-37.030309, 174.864341	-37.030328, 174.864339	-37.030296, 174.864373
Site 3	-37.030285, 174.862107	-37.030295, 174.862093	-37.030289, 174.862114
Site 4	-37.030717, 174.862393	-37.030734, 174.862398	-37.030715, 174.862371

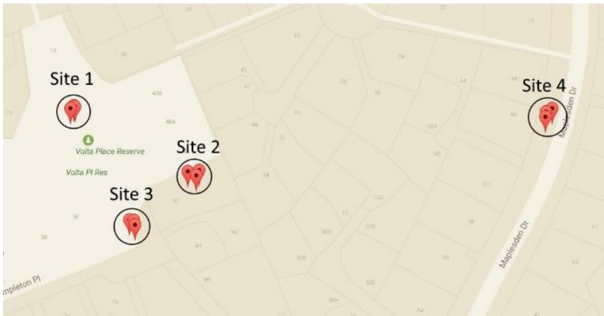

Figure 6: Aerial view of all geolocation test sites with 3 phone locations for each.

9.2 Controller Latency Test

The robot and the commander were set up to test the commander-to-robot latency under favourable local area network conditons. The robot smartphone and the human commander laptop were the only two devices on a home Wi-Fi network.

Table 2 outlines the results from the controller latency test. A built-in JavaScript function provided the logged times.

Table 2: Game Controller Latency Results

Trial	Command Time	Response Time	Delay (ms)
1	1510818903721	1510818904120	399
2	1510819160916	1510819161238	322
3	1510819405869	1510819406445	576
4	1510819423680	1510819424045	365
5	1510819441412	1510819441979	567
6	1510819459588	1510819460073	485

9.3 Video Streaming Latency Test

The video streaming latency test measured the time difference between an event happening in front of the phone camera and the same event displaying to the human commander. A stopwatch recorded the time difference and the recorded value has an error allowance of ±500ms due to accounting for reasonable human reaction time (Thorpe, Fize, & Marlot, 1996). The control devices for this experiment were a Google Nexus 6 and a Samsung Galaxy J5. Table 3 outlines the results from this test.

Table 3: Video Streaming Latency Results (Seconds)

Trial	Nexus 6	Galaxy J5	Robot Vodafone VFD-300
1	1.84	2.64	1.64
2	2.10	1.42	1.72
3	1.89	2.94	2.19
4	1.44	2.53	2.53
5	2.33	1.82	2.00

9.4 Motion Detection Test

The robot was set up to commence monitoring in three different environments ranging from daylight to dim light with minimal background movement to carry out testing on the motion detection feature. Then a yellow object of size 300mm x 300mm passed in front of the camera from varying distances ranging from 1m to 5m. There were ten trials for each distance for each device and the results yielded a success rate percentage. The control devices for this experiment were a Google Nexus 6 and a Samsung Galaxy J5.

Table 4 outlines the results of the motion detection experiment. The ambient light levels of the experiment were:

- Daylight: 3083.9 lux
- Bright: 42.8 lux
- Dark: 12.3 lux

The error allowance of this experiment is ±5% due to minor background movements.

Table 4: Motion Detection Test Results

Nexus 6 Successful Detection Percentage

Distance	Daylight %	Bright %	Dark %
1m	100	100	80
2m	100	100	70
3m	100	80	60
4m	100	70	70
5m	80	60	60

Galaxy J5 Successful Detection Percentage

Distance	Daylight %	Bright %	Dark %
1m	100	100	70
2m	100	100	40
3m	100	65	10
4m	40	50	0
5m	60	50	0

Robot VFD-300 Successful Detection Percentage

Distance	Daylight %	Bright %	Dark %
1m	100	100	60
2m	100	100	50
3m	70	90	20
4m	100	60	10
5m	80	50	0

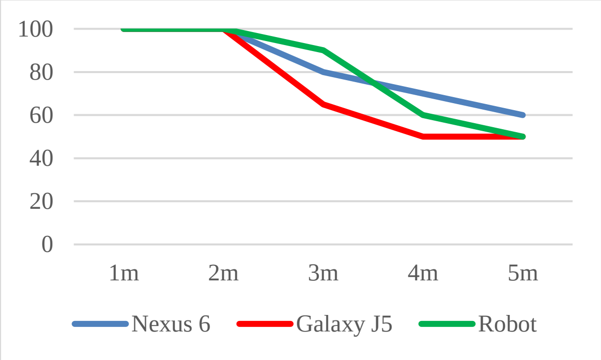

Figure 7: Motion detection success rate under bright conditions

9.5 Speech Recognition Test

To test the accuracy of the speech recognition engine, the phone was set in loud and quiet environments, then spoken to with a clear voice. There were 20 trials attempted from varying distances (1m – 5m) with the same voice and same predefined prompts. These trials yielded a success percentage for each distance. The control devices for this experiment were a Google Nexus 6 and a Samsung Galaxy J5.

Table 5 outlines the results of the speech recognition accuracy test carried out under different background noise conditions. The "loud" environment had a decibel reading of 70.4dB whereas the quiet environment had a decibel reading of 38.5dB. The speaker tried to use a consistent voice level despite the background noise, but there may have been an element of human error as the human voice automatically adjusts to the loudness of the background noise (Lindstrom, Waye, Södersten, McAllister, & Ternström, 2011). The allowance for this error was ±20%.

Table 5: Speech Recognition Accuracy Percentage in Loud and Quiet Environments

	Nexus 6		Galaxy J5		Robot	
Distance	Loud	Quiet	Loud	Quiet	Loud	Quiet
1m	90	95	80	85	70	85
2m	30	95	15	85	20	75
3m	15	90	10	70	10	80
4m	20	85	0	80	10	80
5m	0	75	0	70	0	70

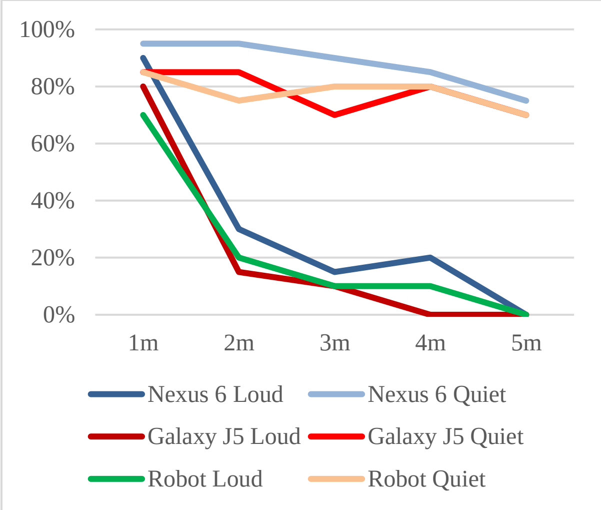

Figure 8: Speech recognition success rate vs distance under loud and quiet conditions

10. RESULT FINDINGS

The GPS unit's accuracy was satisfactory because the coordinate readings from the three devices all fell within 2-3 meters of each other. The close clustering of readings at each site from all devices proves that the robot can successfully transmit its location as per high-end smartphone standards.

As shown from the experiment to investigate the lag between the game controller command and the robot's response, the unit was able to respond within 452ms on average. According to the findings from Claypool and Claypool (2006), the average lag in online games should be under 500ms - the threshold of a third-person sport/RPG game. This is a comparable use case for the movement of the unit.

The practical, real-world value of this latency, however, will depend on the network speeds for both the commander and the robot. The test would require more trials under a variety of network conditions to claim a reliable average response time.

The results of the speech recognition test demonstrates that the robot picks up human speech better than expected under very loud background noise conditions. However, it is not able to perform effectively from a long distance. This is likely to be due to the smartphone's microphone capability. This means that the robot, as it stands, does not perform to the quality of home automation or Amazon Alexa-like devices (Yan, Castro, Cheng, & Ishakian, 2016) but it still has the potential to carry out tasks which involve the user being near the device. One possible solution for our robot is for the human commander to move the robot closer to the speaker.

The motion detection test results indicate a heavy reliance on bright light for the feature to work satisfactorily. This means that the feature will not work at night without satisfactory lighting. However, the robot can act as a typical home security device during the daytime, as it can monitor an area over a large distance. One caveat to using the device over a large distance though is that it has the potential to pick up false positives due to random movements from the environment (due to a larger field of view) (Harrouss, Moujahid, & Tairi, 2015). Therefore, the ideal place to station it is in a room (with no people) or a location without much external movement.

11. DISCUSSION

The solution prototype exists to answer the three research questions outlined before:

- "How can the repurposing process be *creative*?"

- “What are the benefits of repurposing smart devices creatively?”
- “What is required to harness the full potential of smart devices?”

This paper responds to the first research question with:

“The repurposing process can be *creative* by intentionally aspiring to find a unique way to utilise any component that is available.”

This approach yielded the designs and solutions present in the project. Among several innovations, some highlights are as follows:

- The project utilised the phone’s web browser to simplify app development and provide a cross-platform solution. This allows the client-side code to remain maintainable and reduces deployment effort.
- The phone’s camera, while normally simply used to take photos, can detect motion and still serve its original purpose (by capturing the motion event in a photo).
- MS SQL Server stores the chunks of video sent during video streaming rather than the server’s file system. This enables fast data transfer during a time-critical scenario like video streaming. MS SQL Server was never designed to perform video streaming, but it performs in this project without fault.
- The screen of the robot serves a completely different purpose to what it would have been designed for. Its purpose is to allow humans to view the phone’s screen to consume information. In this project, however, a set of light dependent resistors consume the information on the screen. This in turn sets the entire unit into motion through a very simple circuit.

The second research question asks to outline the benefits of repurposing smart devices creatively. The largest benefit of this prototype is that it has the potential to serve as an educational tool. The target audience of this robot are tech-savvy teachers, students or hobbyists. Its low cost and simple architecture allows educational institutions to replicate the design, using it as a tool to teach how a client-server system can be utilised in a unique and interesting way. This project’s focus on using readily available programming languages and a popular software platform makes it easy to integrate into most software or robotics courses. For a beginner, the gentle learning curve of HTML and JavaScript can allow them to become familiar with important technical concepts used throughout the development industry.

The third research question seeks the gateway to access the potential of a smart device. For this project, this gateway was the factory installed Chrome browser. Additionally, some easy to understand and well-documented JavaScript APIs allowed easy interaction with the smartphone’s capabilities.

Interacting with the device hardware through the browser using JavaScript provides a few major benefits as opposed to using a low-level language:

- It makes the functionality cross-platform and easily deployable to any device. Any smartphone, tablet, laptop or desktop computer from any manufacturer, even when not placed in the mobile robot housing, can be utilised, simply by visiting a link (URL).
- The device does not need any special configuration or apps to unmask its potential. Even when there is an update to the program, there is no software deployment necessary. This is a simpler user experience than even a standalone app can provide.
- The current version of the Chrome browser (Version 62.0.3202.84) fully supports the features and APIs that are necessary for the prototype to function. The device used was a very low-end smartphone with storage and memory issues. Despite these shortcomings, the Chrome browser successfully performed all tasks required by this project.

11.1 Implications

To answer the second research question from a real-world benefit viewpoint, the potential applications of this project include:

- An all-rounder robot companion.
- A toy or entertainment robot.
- Security monitor for an empty home.
- Monitor babies and toddlers remotely.
- Helper to the elderly or disabled by being personable with additional voice capabilities.
- Viewer and explorer for remote locations for wildlife and environment monitoring.

All these features are obtainable for a very low price as the parts are readily available and are based on free and open source software.

A study into inserting smartphones into toys reveals a creative approach to utilising device features in an entertaining way (Fan, Shin, & Choudhury, 2014). In a similar way, the speech recognition system from this project can interact with a child to teach them vocabulary and mathematics.

Interested parties can utilise the lessons learned from a project such as this since it combines familiar technology in a unique configuration to create something powerful and interesting. Around the world, students drop out of engineering and science courses because they do not get enough exposure to real-world applications and use cases of their study material. This project demonstrates a very practical way to utilise essential modern technology and covers important concepts in programming, robotics, and network communication. This real-world, result oriented approach should relate to students in such courses and peak their interest in the field of science and technology (Cortés, Gaviria, Garcia, & Rueda, 2017).

11.2 Future Work

The architecture supports multiple robot and controller pairings to allow the same server to serve multiple concurrent users. This may be possible, but the project does not include tests with multiple concurrent controller/robot pairings. The server may need to use asynchronous methods or store the data in a different storage system such as an in-memory database to improve performance.

Depending on demand for the video streaming service, users can purchase a TURN server to allow real-time video streaming capability. The project is a prototype which is not ready for commercial release because commercial grade testing needs to take place before releasing it. The speech recognition prompts and responses may need improvement or customisation to provide more intelligent output to the user.

12. CONCLUSION: BETTER REPURPOSING, A BETTER TOMORROW

There is generally a lack of innovation when repurposing devices because the industry tends to observe the same routines and procedures. It is possible to construct a feature-packed repurposed device by being creative. By utilising the original device’s functionalities to a large degree, this approach prevents wastage of the resources spent on creating the original device. It also infuses the device with value beyond the threshold of planned obsolescence.

The learning experience that a hobbyist or student can gain from a repurposing endeavour such as this project is valuable because it teaches a set of real-world skills using well known and popular languages and platforms. A powerful solution is possible to achieve even with a low-end device, by creatively utilising components such as a cheap Android phone, a standard browser, easy to learn programming languages, a simple client/server setup, and pre-existing JavaScript libraries and APIs.

By trying to utilise a piece of functionality in a different way to its original purpose, reveals a large number of hidden potentials. This can give new life to a device that its owner would otherwise choose to discard. Additionally, this project proves that even a low-end smartphone has similar capabilities to devices several times its price.

Despite the awareness of the potential that smart devices have, they are still underutilised. The greatest potential of this project lies in its ability to inspire these people by revealing the power of readily available components. Coupled with a simple architecture and easy to learn programming languages and APIs, even low end and obsolete smart devices can perform tasks that are beyond typical expectations.

13. REFERENCES

abbycar. (2016). *SpeechSynthesis - Web APIs | MDN*. Retrieved November 20, 2017, from https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis

Afrin, S. (2018). *CRAT Motion Detection*. Retrieved from https://www.youtube.com/watch?v=uBGgNHqDlz8

Afrin, S. (2018). *CRAT Speech Recognition*. Retrieved from https://www.youtube.com/watch?v=QoeTP63kAfQ

Benton, D., Hazell, J., & Coats, E. (2015). *A circular economy for smart devices.* London: Green Alliance.

Boyd, W. (2016). *Motion Detection wtih JavaScript*. Retrieved October 24, 2017, from http://codersblock.com/blog/motion-detection-with-javascript/

Cao, F., & Singh, J. (2004). Efficient event routing in content-based publish-subscribe service networks. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 929-940). Hong Kong, China: IEEE.

Chambers, J., Paquette, D., & Timms, S. (2016). *ASP.NET Core Application Development: Building an application in four sprints.* Redmond, WA: Microsoft Press.

Claypool, M., & Claypool, K. (2006). *On Latency and Player Actions in Online Games.* Retrieved November 19, 2017, from http://digitalcommons.wpi.edu/computerscience-pubs/57

Cortés, B., Gaviria, F., Garcia, S., & Rueda, S. (2017). Development of a platform for teaching basic programming using mobile robots. *Revista Facultad de Ingeniería, 26*(45), 61-70.

Eubanks, A.-M., & Strader, R. (2012). Introduction to programming with the Finch robot. *Journal of Computing Sciences in Colleges, 27*, 5-5.

Fan, S., Shin, H., & Choudhury, R. (2014). Injecting life into toys. *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications.* Santa Barbara, California: ACM New York.

Github. (2017). *The State of the Octoverse 2017.* Github.

Harrouss, O., Moujahid, D., & Tairi, H. (2015). Motion detection based on the combining of the background subtraction and spatial color information. *2015 Intelligent Systems and Computer Vision (ISCV)*, (pp. 1-4). Fez.

Hätönen, S., Nyrhinen, A., Eggert, L., Strowes, S., Sarolahti, M., & Kojo, M. (2010). An experimental study of home gateway characteristics. *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (pp. 260-266). Melbourne, Australia: ACM.

Hoang, D., & Niyato, D. (2016). Information service pricing competition in Internet-of-Vehicle (IoV). *2016 International Conference on Computing, Networking and Communications (ICNC)*, (pp. 1-5). Kauai, Hawaii.

Hodoň, M., Húdik, M., Tóth, Š., & Kochláň, M. (2017). Smart Screen System for Smart Buildings Made of Tablets. *International Conference on Innovations for Community Services* (pp. 185-190). Darmstadt: Springer, Cham.

Kelly, T. (2001). Moon Lander: How We Developed the Apollo Lunar Module (Smithsonian History of Aviation and Spaceflight (Paperback)). Washington D.C.: Smithsonian Institution Press.

Lambrecht, J., Chemnitz, M., & Krüger, J. (2011). Control layer for multi-vendor industrial robot interaction providing integration of supervisory process control and multifunctional control units. *2011 IEEE Conference on Technologies for Practical Robot Applications* (pp. 115-120). Woburn, Massachusetts: IEEE.

Lindstrom, F., Waye, K., Södersten, M., McAllister, A., & Ternström, S. (2011). Observations of the Relationship Between Noise Exposure and Preschool Teacher Voice Usage in Day-Care Center Environments. *Journal of Voice, 25*(2), 166-172.

Marcus, D. (2017). *Insanely Simple WebRTC Video Chat Using Firebase (With Codepen Demo)*. Retrieved October 25, 2017, from https://websitebeaver.com/insanely-simple-webrtc-video-chat-using-firebase-with-codepen-demo

Mills, C. (2017). *Using the Gamepad API*. Retrieved October 25, 2017, from https://developer.mozilla.org/en-US/docs/Web/API/Gamepad_API/Using_the_Gamepad_API

Ponomarev, N. (2017). *SpeechRecognition - Web APIs | MDN*. Retrieved November 20, 2017, from https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition

Rehu, C., & Al-Ali, F. (2017). Internet of Things: Survey, Observations and Future Trends. *Proceedings of the 8th Annual Conference of Computing and Information Technology Education and Research in New Zealand* (pp. 28-36). Napier: CITRENZ.

Sparkfun. (n.d.). *Rover 5 Robot Platform - ROB-10336 - SparkFun Electronics*. Retrieved from https://www.sparkfun.com/products/10336

The Guardian. (2016). *Turning old smartphones into anti-burglary devices and baby monitors*. Retrieved October 20, 2017, from https://www.theguardian.com/sustainablebusiness/2016/aug/07/old-smartphones-security-cameras-baby-monitors-e-waste

Thorpe, S., Fize, D., & Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 520-522.

Truenet. (2017). *NZ Latency - September 2017 | Truenet*. Retrieved November 20, 2017, from https://truenet.nz/nz-latency

Turley, P., & Wood, D. (2009). *Beginning T-SQL with Microsoft SQL Server 2005 and 2008.* Indianapolis, IN: Wiley Publishing, Inc.

Villas-Boas, A. (2017). *One of the original signature features in Android phones is all but dead*. Retrieved November 17, 2017, from https://www.businessinsider.com.au/removable-batteries-android-smartphones-are-dead-2017-2?r=US&IR=T

W3Schools. (n.d.). *HTML5 Geolocation*. Retrieved October 25, 2017, from https://www.w3schools.com/html/html5_geolocation.asp

Xun, L., Ortiz, P., Browne, J., Franklin, D., Oliver, J., Geyer, R., . . . Chong, F. (2010). Smartphone Evolution and Reuse: Establishing a More Sustainable Model. *39th International Conference on Parallel Processing Workshops*, (pp. 476-484). San Diego, CA.

Yan, M., Castro, P., Cheng, P., & Ishakian, V. (2016). Building a Chatbot with Serverless Computing. *Proceedings of the 1st International Workshop on Mashups of Things and APIs.* Trento, Italy: ACM New York .

Zallio, M., & Berry, D. (2017). Design and Planned Obsolescence. Theories and Approaches for Designing Enabling Technologies. *The Design Journal*, *20*, S3749-S3761.

Zink, T., Maker, F., Geyer, R., Amirtharajah, R., & Akella, V. (2014). Comparative life cycle assessment of smartphone reuse: repurposing vs. refurbishment. *The International Journal of Life Cycle Assessment, 19*(5), 1099–1109.