Feasibility and Effect of low-cost Haptics on User Immersion in Virtual Environments

Daniel Wonnacott

MSc By Research

Interactive Media

University of York

Department of Theatre, Film Television and Interactive Media

December 2019

Abstract:

Since the later 1990s research into Immersion, Presence and Interactivity in the context of digital media has been steadily evolving into an exciting area of experimentation, fueled by advances in the visual, audio and tracking capabilities of Virtual Reality (VR) equipment, thanks to these improvements studies into the effectiveness of this equipment in producing an immersive experience are now possible. This is most commonly achieved by measuring the perceived level of Presence experienced by participants in virtual environments, with the higher the sense of Presence created, the more effective a VR system is deemed to be. However, due to the current limitations of Haptic interaction methods investigation into the role that touch plays in generating this sense of Presence is somewhat restricted. Following a structured process of design and research work, this project presents a new approach to creating Haptic Interaction by deploying a Haptic Prototyping Toolkit that enables Passive Haptic Interactions in Virtual Environments. The findings of this work provide the foundations for future research into the development of interaction methods of this type.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

CONTENTS:

INTRODUCTION (1) 7

(1)Introduction 7

LITERATURE REVIEW (2) 10

(2.1) Introduction: 10

(2.2) Immersion, Presence and Interactivity: 10

(2.3) Technology and the Human Senses: 19

(2.3.1) Hear, Taste, and Smell: 20

(2.3.2) Vision and Virtual Reality: 21

(2.3.3) Touch and Haptics in VR: 27

(2.4) Conclusions and Research Objectives: 33

DESIGN CONCEPT (3) 36

(3.1) Introduction: 36

(3.2) Initial Design Concept: 37

(3.3) Conclusion: 40

TECHNICAL RESEARCH (4) 42

(4.1) Introduction: 42

(4.2) Design Research: 42

(4.3) Feasibility Testing: 47

(4.4) Outcomes Of Feasibility Testing: 65

(4.5) Conclusions from Technical Research: 66

PROTOTYPE PHASE ONE (5) 68

(5.1) Introduction: 68

(5.2) First Prototype: Hand Development. 71

(5.3) Conclusions from Phase One: 74

PROTOTYPE PHASE TWO (6) 76

(6.1) Introduction: 76

(6.2) Developing Basic Static Interactions. 77

(6.3) Creation of Prototype Door Tracker: 79

(6.4) Conclusions from Phase Two: 81

PROTOTYPE PHASE THREE (7) 82

(7.1) Introduction: 82

(7.2) Creating External collision detection: 83

(7.3) Conclusions from Phase Three: 87

PROTOTYPE PHASE FOUR (8) 88

(8.1) Introduction: 88

(8.2) Designing and Building the Limpet prototypes: 88

(8.3) Full system Testing: 90

(8.4) Development of the Hand 2.0: 93

(8.5) Creating Position Tracking: 95

(8.6) Conclusions from Phase Four: 95

PROTOTYPE PHASE FIVE (9) 96

(9.1) Introduction: 96

(9.2) Design and Implementation of the Environment: 97

(9.3) Designing the Hardware Prototypes: 102

(9.4) Conclusions from the Evaluation Design: 104

EVALUATION (10) 105

(10.1) Introduction: 105

(10.2) User Study: 106

(10.3) Technical Evaluation: 111

(10.4) Conclusions from Evaluation: 114

CONCLUSION (11) 115

(11.1) Introduction: 115

(11.2) Successes, Achievements and Contributions: 115

(11.3) Future works 117

(11.4) Final Closing Statement 118

BIBLIOGRAPHY (12) 119

DESIGN APPENDIX (13) 127

EVALUATION INFORMATION SHEET 231

COPY OF I.P.Q SURVEY QUESTIONS 233

(1)Introduction

Virtual reality of a good enough quality to create an immersive experience for a user has been possible for several years. Since the Oculus Rift (Oculus, 2020) first went on sale in 2016. Followed by the HTC VIVE (Valve, 2017), PlayStation VR (‘PlayStation VR’, 2019) and a vast range of mobile phone powered headsets. Both commercial and academic interest in the field is steadily increasing. The Virtual Reality industry was valued at $11.5 Billion in 2019 and is predicted to grow to $85 Billion within the next five years. (MordorIntelligence, 2019).

The ability of the current generation of VR equipment to create an immersive experience comes from advances in several disciplines. Advances in the quality of small high-resolution screens and improved dedicated graphics cards have resulted in the creation of visually stunning virtual worlds, and lightweight head-mounted displays (HMD), for example, the HTC Vive. (Valve, 2017) Improved body tracking technology has resulted in the ability to move naturally around increasing areas of play. HTC Vive’s maximum play area is now 10x10 meters, for example. (Valve, 2017) Adoption of Ambisonics for the creation of 3D-sound has enabled crisp and realistic audio to be played back through most stereo headsets. (Horsburgh, Mcalpine and Clark, 2011).

The exact nature of the role these advances play in creating an immersive experience is the focus of a growing area of research. This research is based on three pillars of VR: Immersion, Presence and Interactivity(IPI). (Slater and Wilbur, 1997; Singer and Witmer, 1998; Mütterlein, 2018) Since the late 1990s researchers have been investigating the effects of VR equipment on user immersion, by measuring the perceived sense of Presence a participant experiences. (Singer and Witmer, 1998) The research can be classified into two approaches; one investigates IPI to further design of future technology. (Slater and Wilbur, 1997) The other examines IPI to find better real-world applications for VR. (Cheng and Cairns, 2005) The evidence presented in the form of a literature review chapter 2 of this thesis will expand on these principles in more detail. A particularly exciting area for research is the role that Haptic interactions play in creating an immersive experience, as the technology required to engage the touch sense is not as well developed as the systems for creating audio and visual stimulus. According to Michael Abrash speaking in 2015, while at the time chief scientist at Facebook Reality Labs (formerly Oculus VR). “There’s simply no existing technology or research that has the potential to produce haptic experiences on a par with the real world, so any solution will have to come from breakthrough research “(Abrash, 2015.) As the analysis of past and present research into Haptics presented in the literature review will show, there is still a great deal of potential for further research and design work into improving Haptic based Human-Computer Interaction in Virtual Reality.

Current approaches to Haptics can be classified into two fields; Active and Passive. Active Haptics aims to artificially stimulate the touch sense through augmentation of the hands through the use of gloves or an exoskeleton. (Bouzit *et al.*, 2002; Gu *et al.*, 2016) Passive Haptics focuses on engaging the natural touch sense through the use of hand-tracking technology. (Whitton *et al.*, 2005; Simeone and Velloso, 2015) As the evidence presented will show, both approaches currently have limitations. One proposed solution to Passive Haptics that has potential to be developed further is Substitutional Reality (Simeone, Velloso and Gellersen, 2015), the idea of overlaying digital content onto physical objects which act as proxies for their virtual counterparts. However, in its current form, the level of Interactivity offered by SR is limited to interactions with static objects.

The objective of this research project is to design and test a prototype system for creating more natural Interactivity in a Virtual Environment, by building on the pre-existing work conducted into Substitutional Reality and utilising hand-tracking technology and the advances in HMDs. The proposed method to achieve this will be through the creation of a development toolkit, that will provide a system for rapid prototyping of haptic hardware and allow for experimentation in the field of immersive experiences. This prototype system will then be used to conduct a comparison study measuring the impact on users’ immersion in virtual reality when using passive haptics over standard V.R. controllers.

The research, design and evaluation work that forms the main body of work of this thesis will be presented using the following structure—beginning with the presentation of evidence which supports the aim of the research objective in the form of a literature review establishing the motivation behind the problem space this work will address, looking at the history and recent advances of virtual reality and haptics, the importance of interaction with objects and engaging touch in creating a sense of presence in users in virtual environments.

Following this, is a chapter to explain how the initial design concept and brief where created from the evidence presented by the literature review and establishing a set of clear research objectives and design criteria.

The next two chapters present the design and prototyping process, which occurred concurrently during development. For ease of reading, the design research is presented first, followed by the prototyping phase. A concluding design chapter follows this, showing how the final prototype was developed in preparation for the evaluation phase.

Evaluation is presented in three parts, looking at how the test environment was created, analysing the outcomes of the experiment based on the technical performance of the system and analysis of statistical data collected and observations made. Concluding with a review of how well the final design of the hardware toolkit adheres to the design criteria set at the start of the project.

The thesis will conclude with a final overview of the outcome of the research project, highlighting possibilities for further development and future work, the successes and shortfalls in the design solution created and the possible impact the findings may have with regards to its value as a research tool.

End of Chapter, space left intently blank

Literature Review (2)

(2.1) Introduction:

The purpose of this literature review is to show evidence in support of the proposed research objective presented in the opening chapter. Highlighting the Haptic Interaction is a critical element of creating an immersive virtual experience and that current approaches have limitations, based on their cost, technical difficulty of implementation and portability, all of which present a barrier with regards to research into Immersion Presence and Interactivity in virtual environments.

In the subsequent sections of this chapter, academic and technical evidence that supports the argument for the development of an accessible haptic feedback solution will be analysed. Evidence is presented to outline a design problem space and to highlight the gaps in knowledge that must be addressed to create a viable design solution. This information is structured into sub-sections.

The first section contains a discussion on the concepts of Immersion, Presence and Interactivity in the context of Virtual Reality. The objective is to establish clear definitions for each term and highlight the wide range of approaches which influenced the field in the past twenty years. This section also emphasises the critical role that Haptics plays in Human-Computer Interaction. In the second section, a review of research and design work of the technology created to stimulate the senses in a virtual environment is presented. The focus is on the development of visual and haptic technology to further highlight the current gaps in knowledge and limitations of the approaches.

The conclusion to the literature review will highlight where the gaps in current research and development are. Which of these gaps, this thesis will aim to address and the intended approach to the research process to develop a design solution.

(2.2) Immersion, Presence and Interactivity:

Research into the role of Immersion, Presence and Interactivity in virtual environments (VE) is a relatively new field. The evidence presented in this chapter outlines the concepts and proposed definitions of each, highlighting that this is an evolving field with a great deal of opportunity for further research.

The publication of two opposing theories on the concepts and roles of Immersion, Presence and Interactivity in VE published in the late 1990s inspired a greater interest in and discussion of the field. In 1996 and 1997, researchers released two papers that presented Immersion as a measure of a technology's ability to engage the human senses. First, in 1996 an article titled "Immersion, Presence and Performance in virtual environments: An Experiment with Tri-Dimensional Chess" (Slater *et al.*, 1996), introduced the idea. The concept was then expanded on in a second paper in 1997, which posited that "Immersion is a description of a technology, and describes the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding, and vivid illusion of reality to the senses of a human participant." (Slater and Wilbur, 1997, p. 3) They present evidence to support the higher the quality of technology, the better its ability to engage the senses, and therefore, a higher sense of Presence is perceived by the user. Through the comparative study of this sense of Presence, Immersion can be measured objectively and subjectively and is, therefore, a means to assess the quality of the technology applied to a VE (Slater and Wilbur, 1997).

A second publication followed a year later from researchers at the US Army Institute for the Behavioural and Social Sciences, Measuring Presence in Virtual Environments: A Presence Questionaire, presenting a counter-argument. (Singer and Witmer, 1998). In this paper, they state, "Immersion is a psychological state characterised by perceiving oneself to be enveloped by, included in, and interacting with an environment that provides a continuous stream of stimuli and experiences." (Singer and Witmer, 1998, p. 227).

In their paper, Witmer and Singer not only presented an opposing argument to that of Slater and Wilber but very clearly state that they do not agree with the original concept that Immersion is a measure of the technical ability of the system through which a VE is experienced. (Singer and Witmer, 1998, p. 227) Their perspective says more components of Presence must be considered to measure Immersion citing Control, Sensory, Distraction and Realism as factors that contribute to a sense of Presence, which generates a sense of Immersion in a user. As this is a psychological response, they propose Immersion should, therefore, be measured subjectively(Singer and Witmer, 1998).

Singer, Slater, Wilber and Witmer offer different approaches to what constitutes the concept of Immersion in a VE; however, where they both agree is on the importance of Interactivity, definitions of Presence and to an extent the methodology in collecting data.

Both cases propose the need for activity-based stimulus within the VE, and the requirement of some form of a visual avatar, which combined can engage the user's senses, and facilitate interaction. Interactivity is shown as a crucial component in both theories of Immersion, as without it, you fail to create a sense of Presence in the user, which is critical as Presence is proposed as the metric for measuring Immersion in VE (Slater and Wilbur, 1997; Singer and Witmer, 1998).

Presence is defined in both theories as the ability of a VE to generate a sense of 'being there'; a sense of being present in a virtual location, even though physically you are situated in another(Slater and Wilbur, 1997; Singer and Witmer, 1998). In the paper by Slater and Wilbur, they also refer to earlier work by Steur that theorises "presence is the central goal of "virtual reality", perhaps a defining feature" (Steur 1992, cited Slater and Wilbur, 1997). There is a foundation of research that supports this (Heeter, 1992; Held and Durlach, 1992; Loomis, 1992; Sheridan, 1992; Steur, 1992; Barfield and Weghorst, 1993; Barfield et. al., 1995, cited Slater and Wilbur, 1997)(Sheridan (1992), Held and Dulach (1992), cited Singer and Witmer, 1998). Slater and Wilbur suggest from this research. "The fundamental idea is that participants who are highly present should experience the VE as the more engaging reality than the surrounding physical world, and consider the environment specified by the displays as places visited rather than as images seen" (Slater and Wilbur, 1997, p. 4).

Beyond the agreed definition of Presence, there is a unilateral agreement that Presence is a psychological state, that is subjective; therefore, it can be measured. Singer, Slater, Wilber and Witmer all propose using a questionnaire designed to obtain psychological responses on how engaging users found a particular experience or activity in a VE (Singer and Witmer, 1998; Slater, 1999). Slater, however, went on to explain in a follow-up paper in 1999. That in his proposed theory of Immersion, the external influences outlined by Singer and Witmer were personal responses of the user and should be excluded from consideration, as their subjective nature could lead to misdirection in evaluating the technology. He concludes by defining his term for Immersion as "system immersion" and theirs as "immersive response" to attempt to alleviate future confusion. (Slater, 1999)

Measuring personal response to the system is his criticism of the approach taken by Singer and Witmer. The issue he raises with measuring the immersive response is that if two people of different skill levels experience the same VE, a sports simulation is the given an example, one is likely to have a more enjoyable and therefore more immersive experience. This subjective nature of the data gathered is, Slater proposes, not an accurate measure of the Immersion of the VE as outside factors can influence it. Slater offers an alternative that "metrics can be established which are descriptions of the system, and not descriptions of people's responses to the system" (Slater, 1999, p. 1), allowing for the objective collection of data. In conclusion, Slater concedes that until the discovery of these metrics, Immersion has to be measured using questionnaires based on subjective responses to experience in VE (Slater, 1999).

These two opposing concepts of what constitutes measurable Immersion in the context of virtual reality have informed the foundation for the continued research into the field. On one side, there is a traditional scientific and technical approach that builds on the concept presented by Slater et al., "system immersion" (Slater, 1999): measuring Presence as a metric of Immersion by comparing the results to different experiences to virtual reality systems based on the applied technology. Drawing conclusions on which method generates a more immersive experience, based on user observation and feedback; identifying "important factors that contribute to presence" that can be used to "guide the future of the technology." (Slater and Wilbur, 1997, p. 8). This approach implies measuring Immersion as a design tool for improving future VR systems.

On the other is a social science approach based in part on the work of Singer and Witmer, immersive response (Slater, 1999). Where participants undergo an experience in a VE and using a control group for comparison, Immersion is assessed on the answers to a psychological set of questions. This approach can be seen as measuring Immersion as the psychological effects of experiencing a virtual environment. The implied purpose here being, gathering knowledge to understand the potential real-world applications of VR.

The twenty years since these theories first published have seen a wealth of research carried out following aspects of both approaches, contributing to the evolution of the field.

MIT issued a paper in 2000 studying the influence of haptic Interaction in VE, aiming to increase understanding on designing human-computer interfaces by investigating haptic feedback (Basdogan *et al.*, 2000).

In 2001, Schubert, Friedmann and Regenbrecht published an alternative to the presence questionnaire created by Singer and Witmer. Through the study of responses to over 500 online participants, they refined and created the IPQ, IGroup Presence Questionaire, consisting of 13 questions designed to measure Presence in between-group studies. Made up of three subcategories; "Spatial Presence- the sense of being physically present in the VE, Involvement - measuring the attention devoted to the VE and the involvement experienced, Experienced Realism - measuring the subjective experience of Realism in the VE" (*igroup.org*, 2001). The resulting data provides a presence profile of the application or technology being compared.

At the start of 2003, Slater published "A note on Presence Terminology" (Slater, 2003), in which he discusses further the different concepts of Immersion and Presence. He reinforces his original comments from 1999 (Slater, 1999), that Presence is a response to the immersiveness of a system. He concludes by proposing a range of areas for further research, giving examples as:- the relationship between Immersion and Presence, characteristics of an experience that will make it involving, requirements of a system to induce Presence – either through increased realism or better simulation of the human senses. (Slater, 2003)

In 2004 Brown and Cairns presented their findings on game immersion. By interviewing gamers for their perspective of Immersion, they draw conclusions that introduce levels of Immersion: engagement, engrossment and total Immersion, proposing each as further areas of research and the latter as being the principal goal of VR (Brown and Cairns, 2004)

In 2005 Mestre published a paper "Immersion and Presence", discussing further the notions of Presence and its usefulness in measuring whether users' experiences in VR have some validity in real life (Mestre, 2005). Also in 2005, Cheng and Cairns conducted a study to try and understand how Immersion works by attempting to break it, by lowering graphical realism. Their findings were the opposite of their expectations, concluding "we expected that inconsistencies in realism would have a negative impact on Immersion. Our studies show, however, that an immersive experience, once achieved, could, in fact, help to overcome other usability issues" (Cheng and Cairns, 2005, p. 4) opening up another area of a potential investigation. In November the same year, a literature review of the field from the DARWARS Training Impact group looked at the potential for immersive experiences as tools for training and learning opportunities. Concluding that a sense of Presence created in a PC-based environment, should lead to a more significant transfer of knowledge, but noting that there isn't enough research into the area to comment on definitively. "One would expect that these effects would increase learning and ensure transfer. However, there is precious little research that proves this. We recommend that researchers take on this mission". (Alexander *et al.*, 2005, p. 9)

In 2006 Cairns et al. published a paper positing a new hypothesis on how to measure Immersion objectively, through the study of the body and eye movement and task completion time recorded during an immersive gaming experience, rather than by capturing data after the fact. (Cairns *et al.*, 2006) A report published in 2008 by Jennett *et al.* discusses the findings of experiments to test these new approaches to collecting immersion data quantitatively. Their results showed that both task completion time and rate of eye movement could be used as metrics for measuring Immersion objectively. At the same time, data could also be collected by questionnaires subjectively to complement this. (Jennett *et al.*, 2008). Reinforcing Slater's earlier theory that metrics can be identified to measure Immersion objectively. (Slater, 1999)

The findings of a series of test experiments conducted by Spanlang et al. published in 2010 discuss incorporating a full-body avatar with haptic feedback in a VE (Spanlang *et al.*, 2010) The results showed that using full-body tracking opened up the opportunity to investigate "cognitive functions that are responsible for the representation of our human body" (Spanlang *et al.*, 2010, p. 50)

2011 and 2012 saw more papers published that reinforce the role of measuring Presence as a method of investigating Immersion. Immersion in computer games: The role of spatial Presence and flow (Weibel and Wissmath, 2011), Immersive environment: An emerging future of telecommunications (Abbasi and Baroudi, 2012) and Immersion in Digital Games: Review of Gaming Experience Research (Cairns, Cox and Imran Nordin, 2012). While these papers don't specifically look into Immersion in VR directly, they contribute to the knowledge within the field by reinforcing and building on previous concepts and theories of Presence and its relationship to an immersive experience.

In 2013 the Ilmenau University of Technology in Germany published the "The German VR Simulation Realism Scale" (Poeschl and Doering, 2013). A revised and updated version of the presence questionnaire proposed initially by Singer and Witmer (Singer and Witmer, 1998). Their work added four subcategories to the question set: "Scene Realism, realism of Audience Behavior and Audience Appearance, and Sound Realism" (Poeschl and Doering, 2013, p. 36) aimed at improving measuring simulated Realism in VE.

In 2015 a study published by the University of York looked at the effects of Immersion based on player perspective, 1st person v's 3rd person. (Denisova and Cairns, 2015) Concluding that users were more immersed when viewing the world first-person. The findings support the argument for using VR to study Immersion, as, by design, it offers a first-person perspective of VE.

A research paper investigating and aiming to provide a guide on collecting immersion data outside of a laboratory environment was published in 2016 by Stanford University. (Oh *et al.*, 2016) Their findings show that through the use of the newer more portable VR systems, the Oculus Rift DK2 for example, they could create pop-up research facilities at various public locations. They provide insight into considerations that should be adhered to when conducting this approach to data collection, notably health and safety recommendations on the need for close observation of participants when operating in a confined space, and the possibility of distractions caused by ambient noise. One critical point they also make is with regards to physical differences in the general population, observing that participants with visual or mobility difficulties found the experience more challenging. They state that "This was a truly eye-opening experience for our group as we realised the extent to which studies that focus solely on college students can inadvertently alienate a significant portion of the general population." (Oh *et al.*, 2016, p. 2) This research highlights the potential opportunity to collect more valid immersion data by holding experiments outside of the academic community, demonstrating that a more diverse research base provides more accurate real-world data for consideration in the design of future devices.

The Ludwig Maximilian University of Munich published a paper in 2018 which proposed that there are three pillars of virtual Reality: Immersion, Presence and Interactivity. (Mütterlein, 2018). Using another version of a presence questionnaire a survey of 294 visitors to a commercial VR centre aimed to assess the relationship between Immersion, Presence and Interactivity. The conclusions reinforce the theories presented previously, that, Interactivity contributes to Presence, and this, in turn, contributes to Immersion, and therefore by measuring Presence, you can assess levels of Immersion. Also published in 2018 another paper by Mel Slater further reinforces the argument that the better the VR technology is at engaging the senses, the more immersive the experience. (Slater, 2018) He also notes that hopefully the future of research "will be even more productive than the last, given the widespread availability now of immersive systems".(Slater, 2018, p. 433) implying that the influx of small scale and affordable VR equipment now available will lead to further insights into the relationship of Immersion, Presence and the advancement of VE systems. A final example paper from 2018 discusses findings of increasing user experience in a VE through the application of passive Haptics. (Cooper *et al.*, 2018) Using a real object as a substitute for a VR counterpart, concluding that their use "can enhance overall task performance as well as the users' perceived sense of presence". (Cooper *et al.*, 2018, p. 20)

In 2019 a further paper containing contributions again by Mel Slater, Using Presence Questionnaires in Reality (Usoh *et al.*, 2019) looked at the drawbacks of using IPQ surveys. In this paper, they conclude that while this approach is suitable for comparing two VR systems, it has problems when applied in" 'cross-environment' comparisons (virtual to real, immersive to desktop), which do not seem to be valid using this approach." (Usoh *et al.*, 2019) Their findings show that there is scope for further research into how to conduct these cross-system comparisons.

The evidence presented here is only a small highlight of the wealth of studies conducted. It aims to show the diversity in approaches and the different reasons for investigating the roles of Immersion and Presence in VE. Some projects were undertaken for the furtherment of system design (Basdogan *et al.*, 2000; Spanlang *et al.*, 2010), while others were focused on understanding the psychological effects for better application of the technology to real-world uses. (Cairns *et al.*, 2006; Freeman *et al.*, 2017) Furthermore, it shows that the field is still very much evolving, with more research and design work required to contribute to future experiments and a greater understanding of the roles of Presence and Immersion. This evidence supports the motivation for this research project.

Equally as valuable evidence in support of the proposal that haptic interaction is a vital part of creating a sense of Immersion in virtual reality can be gained from analysis of the research methodology applied in the papers reviewed. All the experiments conducted into Immersion involved some form of human-computer interaction. In each case, this interaction was facilitated by haptic devices. These ranged, for example, from the use of a simple mouse and keyboard input (Cairns *et al.*, 2006; Bracken and Skalski, 2009), joysticks and 3-D Mice (Singer and Witmer, 1998; Slater and Steed, 2000), force feedback steering wheel and pedals (Weibel and Wissmath, 2011), modern VR controllers (Oculus/HTC) (Oh *et al.*, 2016; Mütterlein, 2018) and the incorporation of the natural touch sense. (Basdogan *et al.*, 2000; Jennett *et al.*, 2008; Cooper *et al.*, 2018) This evidence clearly shows that Haptics is a vital part of the process of studying Immersion in VE Presented in more detail in the coming chapters is a further discussion on the development of haptic devices and research into the field.

The objective of this chapter was to provide an overview of the evolution of immersion and presence research over the past 20 years. To show that it is a rapidly advancing area of research and to present definitions for Immersion, Presence and Interactivity in the context of the study of virtual environments. The evidence has shown a definitive explanation of Immersion is yet to be agreed upon by the wider academic community, therefore, for this research work immersion is defined in line with the concepts of Slater et al.

- Immersion is a measure of the quality of the technologies ability to engage the human senses. referred to as System Immersion (Slater and Wilbur, 1997)
- Immersion is measured based on the sense of Presence experienced by the user in the VE, referred to as Immersive Response. (Slater, 1999)

Also, Presence can be defined as:

- The creation of a sense of "being there" in the user, a sense of being present in a virtual location. (Singer and Witmer, 1998; Slater, 2003; Abbasi and Baroudi, 2012)

Presence is the measurable metric of Immersion used to determine the immersiveness of a particular approach to creating or interacting with a VE.

And interaction, defined as:

- The ability to look around, move within and directly affect the virtual environment. (Slater and Wilbur, 1997; Singer and Witmer, 1998)

The research and discussions into Immersion and the roles of Presence and Interactivity outlined in this chapter were made possible by advances in technology. Specifically advances in the ability of technology to recreate or stimulate the human senses. The forthcoming sections of this chapter present a review of the research and design work that contributed to these advances, drawing attention to areas for further investigation and development.

(2.3) Technology and the Human Senses:

In the previous chapter, definitions of Immersion, Presence and Interactivity were established. It was shown that Immersion in the context of virtual environments(VE) could be considered to be a measure of the technology applied. Immersion can be used to assess the suitability of the technology for use in the creation of VE (Slater, 1999) The ability to measure Immersion is enabled by recording user responses to how present they felt during the VE experience. (Singer and Witmer, 1998; Schubert, Friedmann and Regenbrecht, 2001; Slater, 2018) This sense of Presence is created through a combination of visual stimulus and the ability to move and interact with the VE. Presented in this section is an overview of the history of research and technological developments that lead to the current state of virtual reality. The objective is to show how the field arrived at the point it is at today, and highlight the areas where gaps in knowledge still exist—providing further evidence in support of the motivation for the proposed design project.

See, touch, hear, taste, and smell: these are the senses we are equipped with to navigate and experience the world around us. Of these senses, sight and touch are vital components that combined allow us to interact with it physically. As the discussion so far has shown, to engage someone in a virtual environment, simulating and stimulating these senses should be considered paramount in creating a sense of Presence. The forthcoming analysis details the advances in technology that lead to the creation of virtual reality and the systems for haptic interaction, focusing on the human senses of vision and touch. Though, to truly create a fully immersive experience, the hearing, smell, and taste senses must be considered as well.

(2.3.1) Hear, Taste, and Smell:

The investigation into the taste sense, in the context of digital simulation, in particular, has only just begun; an example of this type of research comes from the National University of Singapore. Their work proposes a tongue mounted interface that digitally simulates taste, through the use of electrical stimulation. (Ranasinghe *et al.*, 2012, 2013) Alternatively, a paper from 2018 by Karunanayaka et al. presents the 'thermal taste machine,' using temperature to create a sensation of taste. (Karunanayaka *et al.*, 2018) In both cases, the technology is very much still in the experimental phase with further work required before any real-world applications can be considered.

Research into the inclusion of smell in VE has already found some real-world applications. An example of this is in training simulations for firefighters, where researchers developed a backpack that can recreate up to ten different odours to increase a sense of Immersion and realism in the simulations. (Cater, 1994) Other examples of work in this area include the development of synthetic smells for training in medical haptic simulations (Spencer, 2005) and an investigation into authentic smell diffusion in VE (Ramic-Brkic and Chalmers, 2010)

Digital audio for use in virtual reality is a much better-developed field. To do justice to this research would require a dedicated chapter and as this project focuses on looking at the role of Haptics and visual stimulation, included here is a short discussion to highlight the most significant contribution of the body of work available.

Initially conceived in 1974 Ambisonics, is an audio technique that allows for directionality in surround sound systems. (Fellgett, 1974) The main advantage of this system is that it can contain height information in the audio signal as well as the left/right channels and does not require multiple speakers to create a surround sound effect. (Ortolani, 2015) Initially, the technique was not deployed in commercial products until the realisation that it is particularly useful for virtual reality where three dimensional sound is necessary to produce a realistic audio experience. This incorporation into VE is well documented (Verron *et al.*, 2010; Horsburgh, Mcalpine and Clark, 2011; Poeschl, Wall and Doering, 2013; Ruotolo *et al.*, 2013; Kearney *et al.*, 2016) with findings from the research showing that the application of ambisonics to virtual reality environments allows the recreation of highly realistic directional sound.

(2.3.2) Vision and Virtual Reality:

The investigation into the technology used to create virtual reality is not a new field of research. Its origins can be traced back to the 1950s. They can be accredited in part to Morton L.Heilig, a philosopher, inventor, and filmmaker, who in 1957 filed a patent for a Stereoscopic-Television apparatus for individual use. (Fig.1,(Heilig, 1957)) This is the first iteration of what we would know today as a Head-mounted Display (HMD).

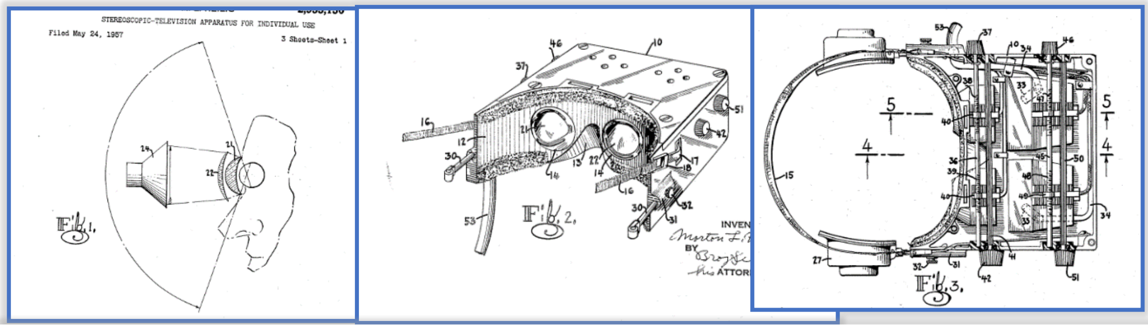


Figure 1: Morton L. Heilig Stereoscopic-Television apparatus for individual use

A few years later, in 1961, Heilig designed the Sensorama Simulator (Fig.2, (Heilig, 1961)), "which was intended to combine multiple technologies to give one to four people the illusion of being in a fully 3D immersive world." (Mortonheilig.com, no date)

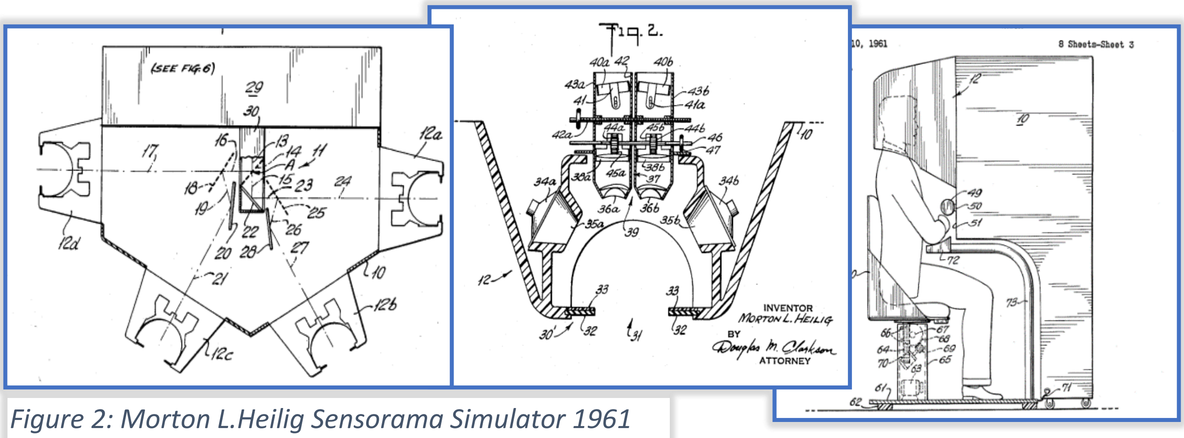


Figure 2: Morton L.Heilig Sensorama Simulator 1961

Heilig's' designs never entered production, and he did not use the term virtual reality (VR), the concepts bear a striking resemblance to the VR equipment available today and earned him the title of "Father of Virtual Reality" (Mortonheilig.com, no date).

After Heilig in 1968, Ivan Sutherland, who himself is regarded as "Father of Computer Graphics," proposed a concept for a kinesthetic

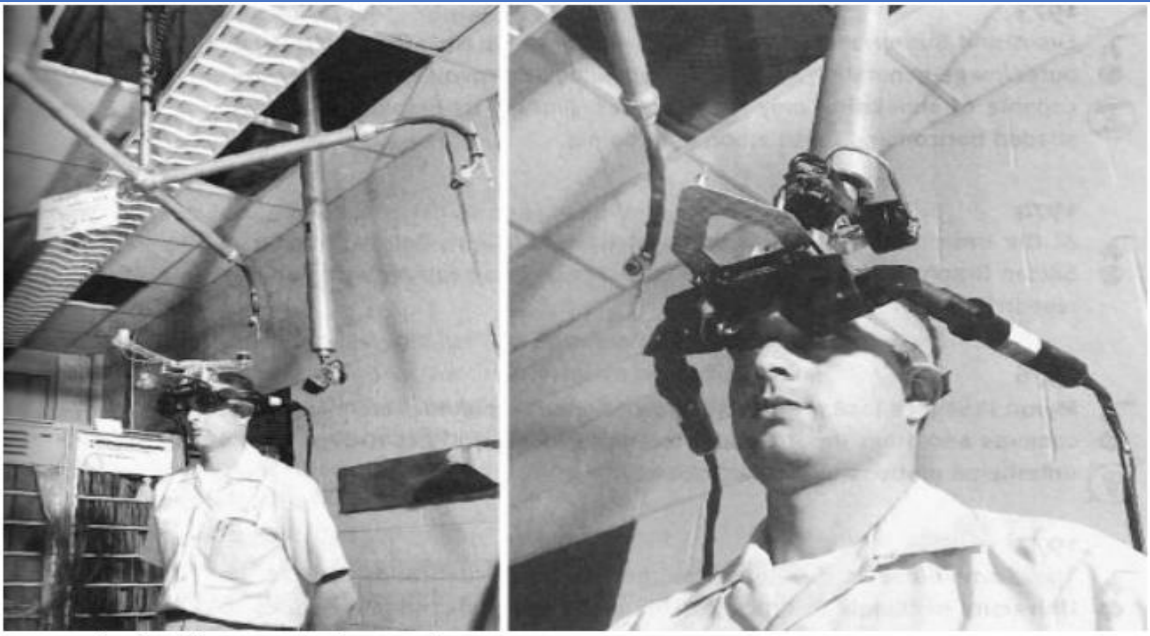


Figure 3: The Sword of Damocles: Ivan Sutherland, 1968, MIT

or ultimate display. In short, he describes "a room within which the computer can control the existence of matter," not only generating visual stimulus but creating objects with physical properties, where "A chair displayed,”" would be good enough to sit in," finishing by describing it as "the Wonderland into which Alice walked" (Sutherland, 1968). This ultimate display concept was immortalised in science fiction history by Gene Roddenberry in a 1988 episode of Star Trek The Next Generation, The Big Goodbye, with the introduction of the Holo-Deck. (Joseph L. Scanlan, 1988). For some, making this a reality can still be considered the ultimate goal of interactive media. (Steinicke *et al.*, 2008) Sutherland went on to create the "Sword of Damocles" in 1968 (Fig.3,(Turner, 2018)), which, as Heilig's design from 1957 was not built, is the first real practical example of a head-mounted display (HMD).

This technology being developed had still not directly been referred to as 'virtual reality'. This term was not popularised until the 1980s by another pioneer of the field Jaron Lanier a computer scientist and artist. He founded VPL Research in 1984, the first company to sell VR equipment and gloves, and they released the DataGlove, EyePhone HMD, and the Audio Sphere. (*VPL Research* | *C-SPAN.org*, no date). His work to bring what had previously only been dreamed of into reality gave birth to the first generation of virtual reality technology.

Throughout the remainder of the 1980s and into the early '90s, research and development continued across multiple fields. British AeroSpace developed a virtual cockpit (*BAE Systems*, 2019), NASA scientist developed VIEW, the Virtual Interface Environment Workstation (Rosson, 2014), and the University of Illinois developed the CAVE. (Cruz-Neira *et al.*, 1992)

Following this, the first commercially available virtual reality devices began to emerge. The Virtuality Group launched 'Virtuality,' and the 'CyberBae SU2000' range of arcade-style virtual reality set-ups (Giles and Kerry, 1994) and in 1994 Sega announced the SEGA VR (*SEGA*, 2019), followed by the ill-fated Virtual-Boy from Nintendo in 1995 (Iwata, 2019). Both devices were considered commercial failures. The Sega headset was plagued with technical difficulties and in fact, never made it out of the prototyping phase. The Nintendo offering could only produce red and black vector graphics, and this device was not released on the international market.

The main issue with the first generation of commercial VR technology was that it was hindered by the available computer and graphics power of the time. The Virtuality groups HMDs, for example, ran on the Amiga 3000, which was powered by a Motorola processor running at 16 or 25 MHz, had only 2MB of RAM and no dedicated graphics chip. (‘Amiga 3000’, 1991) The results were low-poly or vector graphics (Fig 5, (Rotberg, Rubin and Hector, 1983; W Industries, 1991), which were simply too inefficient to generate the required sense of Presence in the user.

Figure 5: Examples of first-generation V.R circa 1990s BattleZone (right) Dactyle Nightmare(left), Su2000 (center)

In the late 1990s and through the 2000s commercial and public interest in Virtual Reality has slowly started to grow again, with movies like the Wachowskis' The Matrix Trilogy(Wachowski and Wachowski, 1999), Joseph Kosinski's, Tron Legacy (Kosinski, 2010) and more recently Ernest Clines Ready Player One (Spielberg, 2018), all playing their parts to recapture interest in escaping into a virtual world.

In 2012 the Oculus Rift. (Oculus, 2020) was created by Palmer Lucky, an independent developer of custom gaming equipment. His concept was of an HMD with visuals only previously described in science fiction. (Fig 6, (Oculus, 2020)

Figure 6: Oculus Rift Dev Kit -1 And Scren shot of orginal game graphics

st demonstrated at the Electronic Entertainment Expo (E3) in 2012 and was able to raise $2.4 million in funding on Kick Starter, before eventually being bought and fully funded by Facebook (Clark, 2014). It would take four more years for the finished HMD to be completed, released on the market in October 2016, the rift has gone on to be the best selling VR device of contemporary times.

In the years between 2012 and 2016, other HMD systems were released. HTC, at the time a significant smartphone producer, partnered with Valve Studios, and announced the HTC VIVE (Valve, 2017), while Sony presented their offering, PSVR(‘PlayStation VR’, 2019). Samsung also released the "Gear VR" (*Samsung Gear VR*, 2019), and Google created "Google Cardboard" (*Google Cardboard*, 2019), both devices that operate using a smartphone to drive the experience.

With improved visuals and sounds, this new generation of VR technology was capable of creating a sense of Presence and, in turn, an authentic, immersive experience.

The current state of VR technology was made possible as a result of earlier development and creative innovation of may contributors to the field. This discussion so far has focused on the development of hardware which created the HMDs required for VR, but the technology also relies on advances in the field of tracking moving objects. As Jaron Lainer said in an interview on the current state of virtual reality, " VR's evolution came about not through advances in graphics, but advances in body-tracking sensors to faithfully recreate movement within virtual environments." (Rowley, 2018)

As established in previous research, accurate tracking of body position is a crucial component in creating a sense of Presence. (Slater and Wilbur, 1997; Singer and Witmer, 1998)

Tracking, in terms of virtual reality, is the ability to monitor and update location data on a person or object in virtual space, based on relative position to a fixed location or features in the real world, and in real-time. The technical term for this is Locomotion. This system uses a 3-dimensional representation of the cartesian co-coordinate system (x,y), with the addition of the Z-axis, for depth. Tracking using just the x/y/z axis data is known as 3 degrees of freedom, but VR also relies on tracking rotation information as well, adding three more degrees of freedom for 6 in total.

Extensive research has been carried out into this particular area, investigating techniques for achieving high accuracy in real-time tracking. These approaches have included:

- Computer Vision/SLAM (see below for definition) (Simeone and Velloso, 2015; Ramadasan and Pascal, 2015)(ODA, 2000)
- IR-Light based tracking with tags (Xu, Wang and Jiang, 2017)
- Visible Light tracking with LED's (Huynh and Yoo, 2016; Liang and Liu, 2017; Zhuang *et al.*, 2018)
- Hybrid Systems that use a combination of Visible and IR light (Kazikli and Gezici, 2018)

These approaches can be broken into two groups of systems those which track:

- A user from the outside (out-side in).
- The world from the users perspective (inside out).

Of the research carried out so far, two methods have proved promising and accurate enough to implement into commercial systems.

- Optical-based (out-side in): This is the system adopted by the HTC VIVE. Two lightboxes are placed into the target environment. They each emit a wide-angle, 2-dimensional array of IR laser beams which are swept across the space, on a single axis at a time, before this, they emit a flash of IR light, as each tracked object or device contains an array of photodiodes connected to an onboard chip, which can measure the time between the flash and the beam for each axis, this data can be used to calculate its relative position in virtual space. This utilises a technique known as time of flight (TOF)

- SLAM (Inside Out): Used in Microsofts Mixed Reality Headsets. For this approach, the tracking is achieved by observing the world from the HMD's perspective. In simple terms, this uses a specific type of algorithm referred to as Simultaneous Location And Mapping (SLAM). Advanced computer vision techniques are applied which allow onboard processors to recognise features in the environment allowing comparison of data from accelerometer and gyroscope sensors with how these features appear to move. From this, the position of the HMD can be calculated. (Spatial Scan)

There are advantages and limitations to both systems: Out-side in: offers a high level of accuracy in the tracking, but the need for lightboxes makes the set-up expensive to produce and implement. With Inside-out, there is a lower cost of setting up, and it does not need external equipment but will not work in the dark and is limited to tracking objects which are in the field of view of the HMD, therefore, tracking controllers behind your back or head is not possible. All of the computational processing has to take place inside the headset. So Outside-in tracking tends to produce more stable and low-latency experiences.

Despite Lanier's general claim that VR evolved due to these improved tracking methods, (Rowley, 2018) it is, however, the combination of tracking with contributions from improved graphics and sound developed for the PC gaming world and high-quality screens and small, powerful processers developed for mobile phones, that have resulted in the successes of the current generation of VR hardware.

As the evidence presented so far has shown research into virtual reality has found practically applicable approaches for tracking movement (Locomotion), producing high-quality 3-D sound (Ambisonics) and thanks to dedicated graphics processors, the creation of visually stimulating environments. However, these approaches only address two of the five human senses, vision and hearing. As established in the conclusions of section 2.1 of this chapter, engaging the sense of touch as well is an integral part of generating a sense of Presence, and therefore Immersion in a virtual environment. The research field dedicated to investigating the incorporation of touch is referred to as Haptics and forms the focus of the next section of this literature review.

(2.3.3) Touch and Haptics in VR:

The human touch sense has a unique ability to process both input and output sensing simultaneously. (Mackinlay, Card and Robertson, 1990; Jones, 2018) The ability to handle both in and output is where the touch sense differs from vision and hearing, which are both only capable of sensing input and it is this difference that makes touch such a vital part of human-computer interaction. The ability to sense touch comes from the largest organ in the human body, the skin, and is an extremely complex system, as detailed in the book Haptics published by the MIT Press. (Jones, 2018)

In the simplest terms the touch sense is made up of two separate systems:

- Kinesthetics or Proprioception that provides location data on where our limbs are positioned and forces applied to them. (Burdea, 1999; Jones, 2018)
- Tactile or Cutaneous, that provides data on the shape, texture, temperature and edges of objects. (Pacchierotti, Prattichizzo and Kuchenbecker, 2015; Jones, 2018)

The cutaneous element of touch allows us to detect interactions all over the body. However, due to the proportion of the brain dedicated to processing sensory data, certain parts of the body are more receptive to tactile inputs, as illustrated by the Sensory Homunculus (Fig 7,(Price, 2019)). First conceived by a neurologist Dr Wilder Penfield, the homunculus model is designed to represent how the body would look if our body parts grew in proportion to the size of the portion of the brain dedicated to handling particular senses.

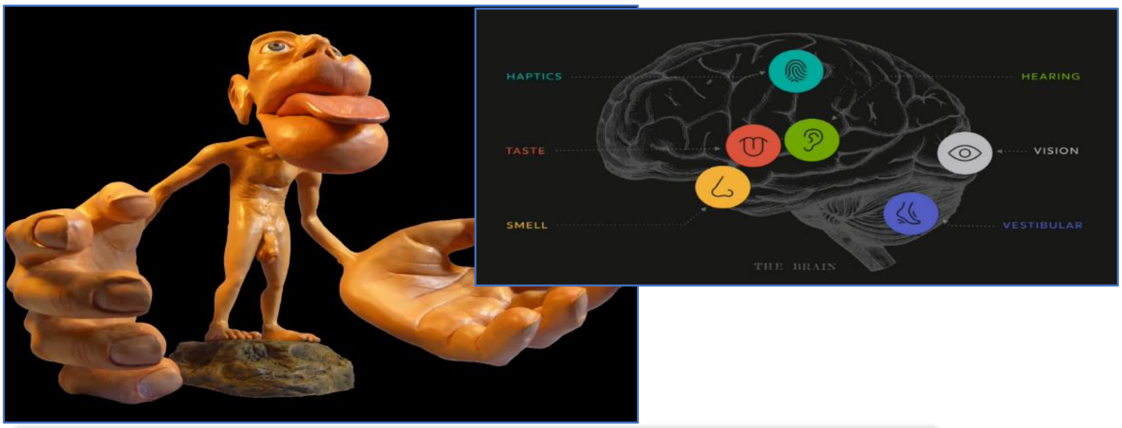


Figure 7: Sensory Homunculus and regions of the brain dedicated to each

As can be seen, our hands dominate this structure, as the portion of the brain dedicated to Haptics dominates over the other senses. It is the dominance of the Haptic sense that makes its inclusion in any VR system, a critical component, and as the previous research has shown facilitates the study of Presence and Immersion. (Singer and Witmer, 1998; Basdogan *et al.*, 2000; Slater and Steed, 2000; Cairns *et al.*, 2006; Weibel and Wissmath, 2011; Mütterlein, 2018)

Research into the field of integrating the human touch sense for Human-Computer Interaction (HCI), much like virtual reality (VR), is not a new field. The history of the development of the area is well documented in several books and journal publications. (Brewster, 2003; Jones, 2018; Parisi, 2018; Prattichizzo *et al.*, 2018) In which they discuss how the research and design into Haptics builds on outcomes of work completed in the field of robotics, dating back to the 1950s. It wasn't until the advent of the home computer in the later 1980s and early 1990s that the area became known as computer-haptics, more commonly, simply referred to as Haptics or Haptic interfaces. (Parisi, 2018, p. 216)

The first generation of haptic devices were input only; the two most commonly known examples of this are keyboard, the invention of which is accredited to Christopher Latham Sholes in 1868. (Sholes, Glidden and Soule, 1868) and the mouse created by Douglas Engelbart (Engelbart, 1970). These devices allowed for input control of computers but lacked any kind of output or feedback.

It was in the 1990s that research and development of haptic devices for HCI interfaces that also provided output as well as enabling input began to take off. (Mackinlay, Card and Robertson, 1990). In line with this, the first commercial products that offered a degree of feedback became available. These devices were targeted at the video gaming industry. Nintendo and Sega developed Rumble Packs for their N64(Nintendo, 1997) and Dreamcast consoles(*SEGA*, 2019), and Sony released the Dual Shock controller for the PlayStation (Playstaion, 2018). These devices offered a basic level of haptic feedback by creating vibrations to simulate taking damage or movement over rough surfaces. Microsoft developed a more advanced approach with their “SideWinder” range of force-feedback joysticks and steering wheels and pedals for PC based driving and flight simulators (Lee, 2007). The SideWinder products offered vibration feedback and force-feedback to enhance the gaming experience.

While hand-held or tethered devices such as the examples given above are suitable for traditional screen-based HCI, they lack the freedom of motion required for a VR experience. Similarly to the range of devices discussed so far, the current generation of VR controllers are hand-held and only provide minimal haptic feedback in the form of vibrations. Examples of these types of device are the controllers for the HTV Vive (Vive, 2018), Oculus Rift (Oculus, 2020)and PSVR ('PlayStation VR', 2019)

Figure 8: Examples of current V.R conrollers : Oculus (Left) /VIVE (Center)/PSVR (Right)

Controllers such as these (Fig 8, (Vive, 2018; 'PlayStation VR', 2019; Oculus, 2020)) allow for a full range of interaction in virtual experiences, by utilising the same tracking methods outlined in section 2.2.2. Still, they suffer from the disadvantage of being hand-held which limits the haptic feedback options to pulsed vibrations, and being hand-held cannot take advantage of the freedom of movement and complex sensitivity offered by the hands. The hands contain multiple internal systems for detecting and sensing interaction, as well as being able to rotate and flex in multiple directions. There are, for example, 23 degrees of freedom in the human hand (ElKoura and Singh, 2003; Jones, 2018) and as has been established current tracking systems are only capable of simulating six degrees of freedom.

Since the late 1990s and into 2000s onwards a great deal of work to develop systems for interaction that overcome the lack of freedom of movement in the current generation of VR controllers has been undertaken. (Srinivasan and Basdogan, 1997; Burdea, 2000; Hollerbach, 2000; Hayward *et al.*, 2004; Eid and Al Osman, 2016) This research and design work can be separated into two fields, Active, and Passive; the following sections look at the various approaches to these areas of research to highlight the advantages and current limitations of each.

Active Haptics:

The Active systems attempt to create artificial haptic feedback, for example, pressure, heat, force, texture and resistance by augmenting the hands. There are two main approaches to achieving this. One which uses gloves worn by the user to stimulate the cutaneous and kinesthetic systems. (Burdea, 2000; Bouzit *et al.*, 2002; Jeon and Choi, 2009; Hoang, Smith and Thomas, 2013; Pacchierotti, Prattichizzo and Kuchenbecker, 2015; Chagué and Charbonnier, 2016; Fani *et al.*, 2018) The other which uses an exoskeleton again worn on the hands (Gu *et al.*, 2016; Achibet *et al.*, 2017; Maisto *et al.*, 2017; Culbertson, Schorr and Okamura, 2018; Choi *et al.*, 2019) to stimulate fingertip touch sensations, force-feedback and resistance. Most of the approaches developed by this research are restricted to a single finger or have to be tethered to extensive and expensive systems to drive the sensations. However, this research work has produced several commercial products. For example, HaptX (Varga, 2019) and Dexmo (*Dexta Robotics*, 2019) gloves are available on the market. Still, prices are in the range of £5000 per pair (Robertson, 2019) which for small scale experimentation into the role Haptics plays on Immersion in VE is expensive, which has limited access to the technology to academic or commercial developers.

Passive Haptics:

Passive Haptics offers a low-cost alternative to the Active approach. It focuses on engaging touch naturally by tracking the hands, using advances in infrared light-based detection systems such as the Microsoft Kinect (Microsoft, 2017), which allows for full-body tracking; and the Leap Motion (Ultraleap, 2019) designed specifically to track the hands. Through the use of these types of systems, it removes the need for any hand-held devices as the hands or body become the controllers. There are many examples of this type of research approach that have shown its ability to enhance the user experience of VE (Insko, 2001; Viciana-Abad, Reyes-Lecuona and Cañadas-Quesada, 2005; Whitton *et al.*, 2005; Henderson and Feiner, 2007; Zadeh, Wang and Kubica, 2007; Steinicke *et al.*, 2008; Pacchierotti, Prattichizzo and Kuchenbecker, 2015). As with Active Haptics, this research has produced commercially viable products; an example of this comes from Bristol-based company Ultra Haptics (Ultraleap, 2019). Their approach uses the Leap Motion for hand tracking in combination with an ultrasonic interaction device that provides mid-air haptic feedback. This ultrasonic approach has been used to demonstrate touchless haptic output in several different scenarios, ranging from supernatural VR experiences (Martinez *et al.*, 2018) to rhythm-based games (Georgiou *et al.*, 2018). This Mid-air haptic approach has also been demonstrated to have applications for shape rendering (Long *et al.*, 2014) and tactile displays (Korres and Eid, 2016; Luzhnica, Veas and Pammer, 2016). There are, however, two significant drawbacks to this Passive approach to Haptics. First is that the systems for mid-air haptics are static, fixed in one location. So for creating objects which can be picked up and moved around in VE, it is simply not suitable. Secondly, the use of hand tracking engages part of the natural kinesthetic element of the touch sense, as the hands can be seen, but the curtaneous systems are not engaged as virtual content has no physical properties. It is possible to touch objects in VE with this approach but there is no tactile feedback and subsequently no force feedback either. These issues can be considered a barrier to the investigation of Immersion and Presence as they limit the Interactivity of the VE, which as previously discussed reduces the sense of Presence and therefore produces a less Immersive experience.

In 2012 a new concept for applying Passive Haptics began to gain ground in the research community referred to as 'Substitutional Reality.'(SR) (Suzuki, Wakisaka and Fujii, 2012) The approach taken with this idea is to design and create VE modelled to overlay virtual content onto real-world objects, for example, walls and furniture. By aligning the virtual content with real physical objects, it is possible to generate VE with physical properties, and when used in conjunction with devices such as the Leap Motion (Ultraleap, 2019), allows for more natural interactions to occur.

The concept of SR was developed further in 2015 by a joint team from Portsmouth and Lancaster Universities in which they clarified the definition of SR as "a class of Virtual Environments where every physical object surrounding the user has been paired to a virtual object." (Simeone, Velloso and Gellersen, 2015). Three further papers published in 2015 reinforced the approach (Simeone, 2015; Simeone and Velloso, 2015; Simeone, Velloso and Gellersen, 2015). The findings from this research were, that size and shape were more critical when dealing with smaller objects, and that weight and texture played a significant role in how convincing a proxy was at acting as its digital counterpart. The key take away they make for future design consideration is "that true 1:1 replicas are not mandatory to enjoy a

VR experience. Indeed, it is possible to substitute more common objects in place of those required by the VR experience." (Simeone and Velloso, 2015).

Following the publications from 2015, a series of research papers have been released which all show how this concept of using real-world objects as proxies for virtual counterparts has demonstratable merit to adding to the Immersive experience of a VE. (Azmandian *et al.*, 2016; Hettiarachchi and Wigdor, 2016; Sra and Schmandt, 2016; Estrada and Simeone, 2017; Suhail *et al.*, 2017; Garcia *et al.*, 2018; Ponraj and Ren, 2018) This approach of using proxy objects as substitutes for virtual content is also not without limitations. In each of the examples given, the objects being simulated have no secondary interactive features; this limits the range of real-world activities that can be reproduced. Examples of this would be if you wanted to replicate an environment which has doors (rotation around a fixed point), or light switches (input and response) or a weapon in a video game (trigger and freedom of motion) this approach in its current form would not be suitable. Simeone et al. in 2015 proposed that building S.R environments that were more interactive would "require new research on nonintrusive sensing devices" (Simeone, Velloso and Gellersen, 2015) which could be placed in the scene. These new sensing devices could enable a user to pick-up and interact with objects in the environment, and investigate the virtual scene, item in hand. For example, In a fantasy game this could be picking up a sword or wand, or in a training scenario operating a new tool, while in reality users could be holding a stick or broom handle or other suitable proxy objects, augmented with sensors that allow for input detection. Design and development of a prototype of this type of sensor can be considered a central objective of this research work.

As the evidence presented in this section of the literature review has highlighted, research into the incorporation of the touch sense in virtual reality is an evolving and rapidly developing field. It has also been shown that current approaches have limitation caused by the cost of systems, portability of the technology and the complexities of replicating the natural touch sense. A potential solution to these issues is provided by ongoing research into Substitutional Reality (SR). Which has demonstrated the potential benefits to Haptic Interaction of this approach to Passive Haptics(Simeone, Velloso and Gellersen, 2015); however, to fully utilise this approach development of new sensor technology is required, which facilitates HCI interactions that can provide input and output. The evidence discussed supports the proposed research work of this project, to develop a toolkit of sensors that allow for more interactive proxy objects to be used in an SR virtual environment.

(2.4) Conclusions and Research Objectives:

The objectives of this literature review were to provide an overview of the current approaches to research into Haptics and Virtual Reality technologies. To show how they are linked with studies into Immersion and Presence in Virtual Environments—highlighting current gaps in knowledge and technical capability of equipment used to investigate this area and the barries this causes to future work. Furthermore, the evidence presented demonstrated the importance of the role that Haptics plays in Human-Computer Interaction.

The evidence presented identifies several points that provide support for the motivations of this research project:

- From the discussions on research into Immersion, Presence and Interactivity definitions for each were presented. They are that Interactivity in a VE leads to a sense of Presence in the user, which in turn creates an Immersive Experience. (Slater and Wilbur, 1997)
- That by measuring this perceived sense of Presence, it is possible to assess the validity of a system used to create a VE, and this can be considered a measure of System Immersion. (Singer and Witmer, 1998; Slater, 1999) Through analysis of research data collected from the study of System Immersion factors that can be used to inform and improve future designs of VR technology can be gained. (Slater, 2018) The benefits of this better understanding of design are that more realistic and natural interactions can be created in VE, which leads to more potential real-world applications of VR technology.
- Haptic interactions form a critical part of research work into Immersion, Presence and Interactivity. (Cairns *et al.*, 2006; Weibel and Wissmath, 2011; Cooper *et al.*, 2018)
- From the evidence presented on engaging the senses through the application of technology it was shown that while Visual and audio technologies provide a suitable level of sensory engagement, further research and design work is required to incorporate the touch sense. Current limitations in the field of Haptics were identified to be, cost of the technology limiting

accessibility to research tools, portability of the equipment and the challenges of replicating the complexity of the natural touch sense.

- That Substitutional Reality offers a potential solution to the complications of Active Haptics by engaging the natural touch sense through the use of Passive Haptics. However, to fully realise this as a solution, further work is needed to develop a higher level of Interactivity, and this could be achieved through a system of sensors that enable both input and output detection in VE.
- That research into Haptics is currently limited to academic and industry-based research and design, and this could potentially be limiting the field. As was shown the breakthrough in VR headset design came from an independent developer (Clark, 2014), who utilised accessibility to domestically available components.

The proposed objective of this research and design project is to develop a prototype toolkit of mobile sensors. That allows the development of Haptic Interaction devices—building on the foundations of Passive Haptics and Substitutional Reality to facilitate increased levels of Interactivity. The proposed outcome of this research and design work will be a toolkit that facilitates rapid prototyping of devices that enable natural touch interactions in VE. Which could help further understanding of the role that Haptics plays in creating a sense of Immersion in Virtual Environments and potentially provide insight into improving future design work.

The system developed should adhere to the following primary research guidelines:

- Be constructed using affordable technology to ensure the solution is as accessible to the most comprehensive range of developers as possible, to open up research and design potential that comes from those working outside of academia or industry research.
- Be of modular design and adaptive so that it can be applied to a range of virtual reality equipment and employed on a wide range of systems, to continue to facilitate the full range of approaches to the investigation of Immersion, Presence and Interactivity.

In the forthcoming chapters of this thesis, details of the iterative design process undertaken in creating the prototype toolkit are explained. This is followed by a discussion on the design and evaluation of a comparison study of perceived Presence and Immersion in a VE when using the new toolkit v's using a traditional VR controller. The assessment is designed to test the feasibility of the proposed approach to Passive Haptic Interaction.

End of Chapter, space left intently blank

Design Concept (3)

(3.1) Introduction:

The discussions contained in the Literature Review chapter provided a background to the motivation for the proposed research and design project. The evidence presented emphasised the rapidly evolving nature of the research filed into Immersion, Presence and Interactivity. Highlighting a range of possibilities for new research directions. From the findings, a critical area of interest to this project was identified A low-cost way of engaging and experimenting with haptic interface design, that incorporates natural touch interaction, does not currently exist, this represents the initial design problem:

How to how to engage the touch sense, without relying on the presently available, complex and potentially inaccessible technologies and how to design a solution that adheres to the two primary design guidelines established in the previous chapter:

- Be constructed using affordable technology to ensure the solution is as accessible to the most comprehensive range of developers as possible.
- Be of modular design and adaptive so that it can be applied to a range of virtual reality equipment and employed on a wide range of systems.

The potential benefits of creating such a system can be considered as:

- Making the technology available to all developers and not just to those based in academia or industry research. Preventing large, well-established companies from monopolising the market.
- A more extensive participation base allows the opportunity for novel and innovative content creation.
- Accessibility to research tools may encourage small independent developers to move into creating content for this newly emerging research field.
- Access to new research tools may inspire new avenues of research and design.

The encompassing concept is to create a design solution that promotes the opportunity for innovation by being as accessible to as broad a development audience as possible. To guide in the design of any solution created, the initial primary guidelines were refined into the following research objectives:

- R1. Built using a technology that is based on passive haptics and does not need an expensive artificial feedback system.
- R2. Modular in design, for ease of adaption to new environments and control systems and creating of new hardware peripherals.
- R3. Incorporate a degree of technical and user testing to establish the feasibility of the proposed approach to Passive Haptic Interaction.
- R4. Follow a structured, iterative design process, to produce a design solution that has the potential to be an accessible means with which to study Immersion, Presence and Interactivity.

The remainder of this chapter outlines the process of how this initial concept idea was refined into a Design Brief—beginning with a review of the evidence from the literature review that provides the building blocks for the design specification.

(3.2) Initial Design Concept:

As established in the literature review, the principal idea of substitutional reality (SR) is the use of real-world objects as proxies for virtual counterparts (Simeone, Velloso and Gellersen, 2015). In an SR environment, a user can physically interact with the virtual surroundings using passive haptics, as the digital content is overlaid onto real objects. However, this approach is limited to interactions with static objects, tabletops or walls, for example, but what if it wasn't? What if you press a button in the virtual world and there is a real button there to touch physically, or if you pick up an object, say a weapon in a game, you can hold it and pull the trigger? , creating this ability to detect interactions in the real world and translate them into virtual responses is how this project intends to expand the interactive capabilities of Substitutional Reality.

The proposed approach to creating an Extended Substitutional Reality (ESR) environment is to design and develop a Haptic Prototyping Toolkit (HPT). The purpose of which will be to allow the use of passive haptics, to engage the touch sense with interactions that stimulate both the kinesthetic and cutaneous systems by providing both physical properties and input and response. The design solution for the HPT will then be used to identify indications that this accessible approach to haptics can potentially enhance a users’ perceived sense of Presence by increasing the level of interaction in a Substitutional Reality enhanced Virtual Environment.

Any solution developed should be constructed with accessibility to the HPT in mind. One approach to enable this is to design the solution based on as low a cost approach as possible, as evidence from the research into VR has shown the opportunity for studies conducted outside of a laboratory increases as the cost of equipment declines. (Slater, 2018)

Based on this idea of taking a low-cost approach to finding a solution and ethical considerations on the environmental impact of promoting rapid prototype design which has the potential to be wasteful with one-off designs being discarded etc.. The following design criteria were established :

- D1. Sourced from pre-established hardware where possible or constructed from easy to source pieces.
- D2. Built on a commonly accessible and where available free to use development platforms.
- D3. Where reasonable incorporate the use of upcycling of components from obsolete, 2nd hand or damage technology, to reduce cost and for the obvious environmental benefits.

With the research objectives and design criteria established the process of refining the initial concept idea continued by drawing on an analogy of the human nervous system. In the same way, the nervous system is a combination of sub-systems; the HPT developed would consist of components. A host PC (Brain), wired or wireless connectivity (Nerves), peripheral sensors for detecting interaction (Fingers) and the head-mounted display of the virtual reality rig providing sound and visuals (Eyes/Ears). The combination of the parts of the HPT will form the Core System Architecture. (CSA) of the prototype design solution.

This HPT system could then potentially operate by receiving signals from peripheral modules, attached to real-world objects in the target environment, that act as proxies for virtual content. The modules could be the sensors to detect interactions, for example, switching on a light or opening a door. An initial phase of ideation was conducted to provide concept drawings of a proposed solution, examples of this initial design work are shown in Fig.1-3

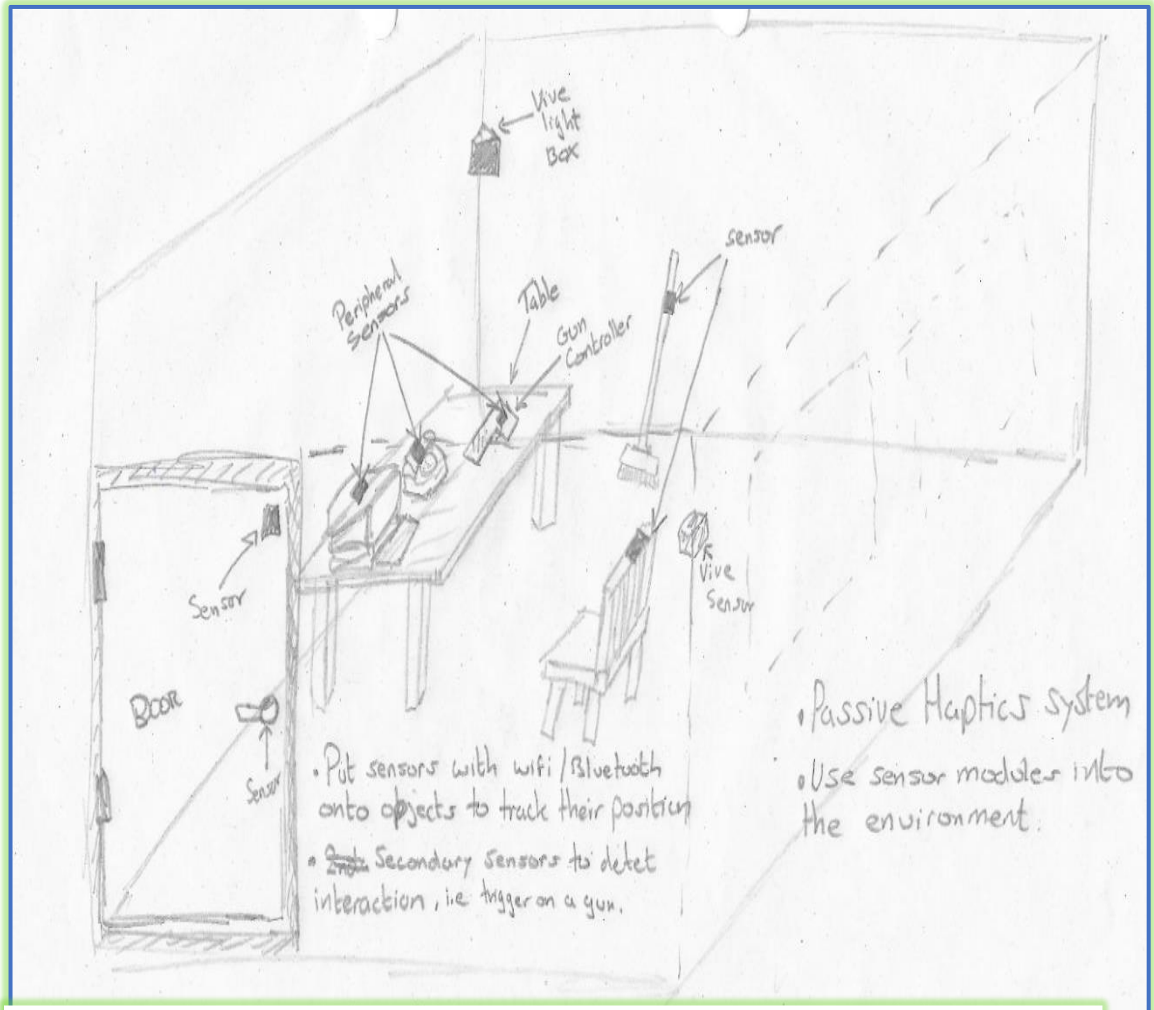


Figure 2: Orignal Concept Sketch For the use of the hardware prototyping toolkit.

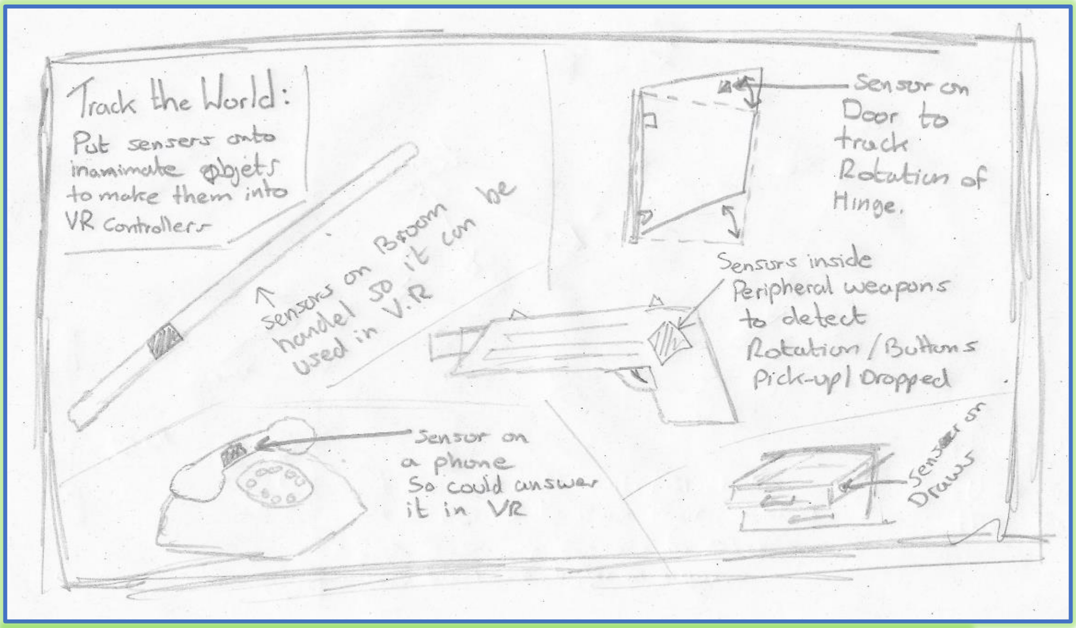


Figure 1: Track the world with modular sensors placed inside or onto real objects

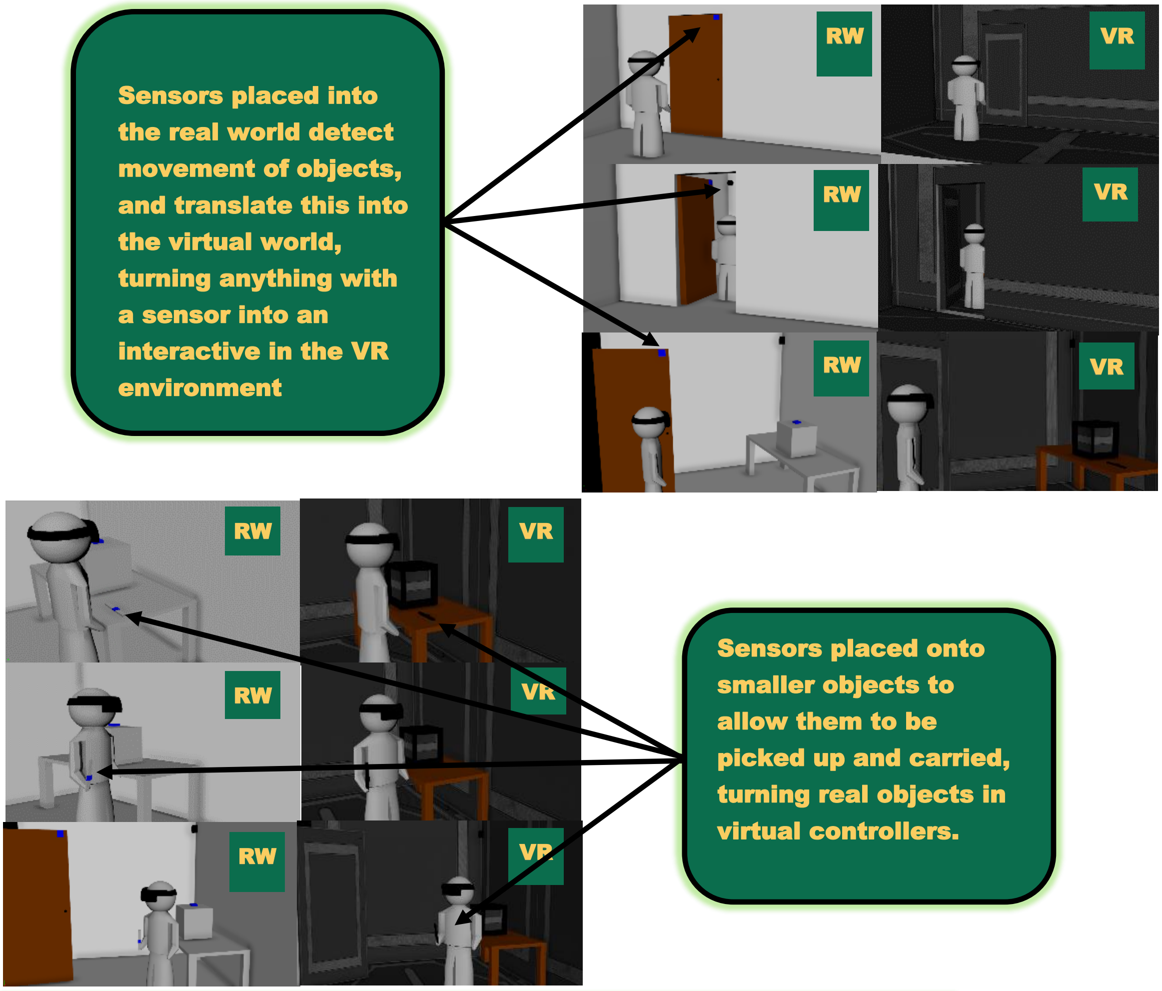


Figure 3: showing how sensors in the real-wolrd allow tracking in the virtual.

(3.3) Conclusion:

This objective of this chapter was to show how evidence generated from the literature review provided an original concept design for this project—outlining the design problem space and the building blocks for the commencement of the design process. The design synthesis conducted produced four research objectives (R1-4) and three design criteria (D1-3) which were established to guide the design process. From the outcome of this design work, it was possible to generate the design brief.

(3.3.1) Initial Design Brief:

To develop a Hardware Prototyping Toolkit (HPT), designed around a Core System Architecture (CSA) to facilitate the use of passive haptics in a virtual environment. By building on the main idea of Substitutional Reality (SR), by overlaying digital content onto real objects and enhancing this with a system that allows for interaction to be detected and translated into digital input and response. The proposed method to achieve this is to develop a network of discrete sensors that can be placed in a real-world environment.

In the subsequent chapters, this Design Brief will be refined through a concurrent process of iterative design research and prototyping. Integrating ideas presented by the design process with those generated by technical experimentation into a solution. Then to test the hypothesis that a low-cost approach to haptics can have a positive influence on user immersion, the prototype system will be put into practice, through user-based evaluation. The user study will also provide the method for technical appraisal of the system, highlighting any design problems for consideration in future work.

End of Chapter, space left intently blank

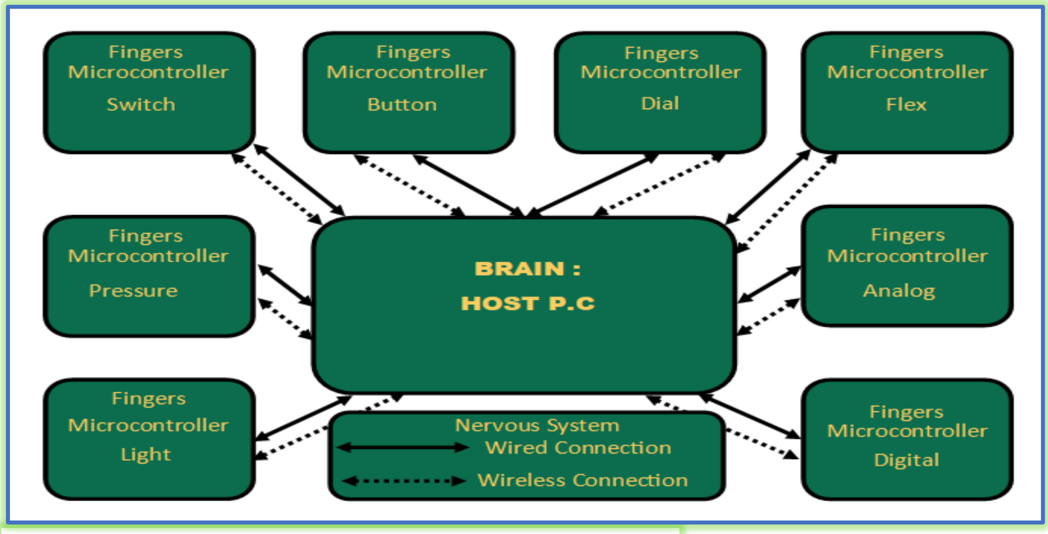


Figure 1: First Concept Design of Toolkit System

(4.1) Introduction:

The purpose of this chapter is to explain the essential technological research and design work that preceded the physical prototyping of the Haptic Prototyping Toolkit. (H.P.T) The objective is to illustrate the process of how each element of the Core System Architecture (C.S.A) was initially selected. In line with the criteria and research objectives outlined in the previous chapter (R1-4/D1-3), and how the design brief was adapted based on the results of the initial feasibility testing. This section is complimented by the Design Appendix, which is included after the main body of text in this document; this shows photographic details and notes from the design work that took place to inform the development process.

The original concept design for the C.S.A can be seen in Fig.1; it shows how the intention was to create a series of modules, placed into the real-world environment that act as sensors that feed interaction data back to the central control (Brain).

The technical research process began with addressing the elements of the C.S.A that would be developed on pre-existing hardware that would not require any design work to implement beyond that necessary for the control software interfaces.

(4.2) Design Research:

As established the starting point for the design process was first to address the components of the Core System Architecture (C.S.A) that would be built entirely upon pre-existing technology, the Eyes and the Brain.

(4.2.2) Eyes and Hand Tracking:

A review of the range of H.M.Ds and currently available hand tracking technology was conducted (details can be found in the design appendix P:5-7). From this information, The H.M.D to be used as the Eyes was selected the device chosen was the HTC VIVE, version one. The decision to use version one over one of the more recent updated version was made based on the cost per unit of the headset. As the overall aim is to produce a low-budget solution it makes sense to develop on an H.M.D which, while not as powerful, in terms of visual quality as newer (more expensive) models, will still perform more than adequately for this research. The secondary reason for selecting the HTC VIVE is that as it is an older model, it will run on less costly hardware, reducing the cost of the Brain (host P.C).

For hand tracking a Leap Motion controller would be fitted to the H.M.D. The Leap Motion allows for accurate hand tracking and is compatible with the USB port on the VIVE HMD. Support for its' integration with the VIVE and into software development environments is also readily available and straight forward to implement. The Leap Motion will provide the required method to access a passive interaction system by tracking the users' hands in real-time.

(4.2.3) Building the Brain:

To establish the Brain requires two parts, a host P.C to develop on, and a design environment to create within.

The host P.C selected was an HP-Omen 17 Laptop, with Intel® Core™ i7-7700HQ CPU 2.80Ghz, 16GB RAM and a GTX 1070 8GB providing the visuals. This device was selected as it offers a balance of cost, portability and reliable processing power, capable of supporting the HTC VIVE.

The Design Environment selected was Unity3d 2019.02. Unity was chosen as it is a widely available product which is free to use, simple to operate and has a full range of support for integrating custom toolkits, and virtual reality equipment via existing SDK packages.

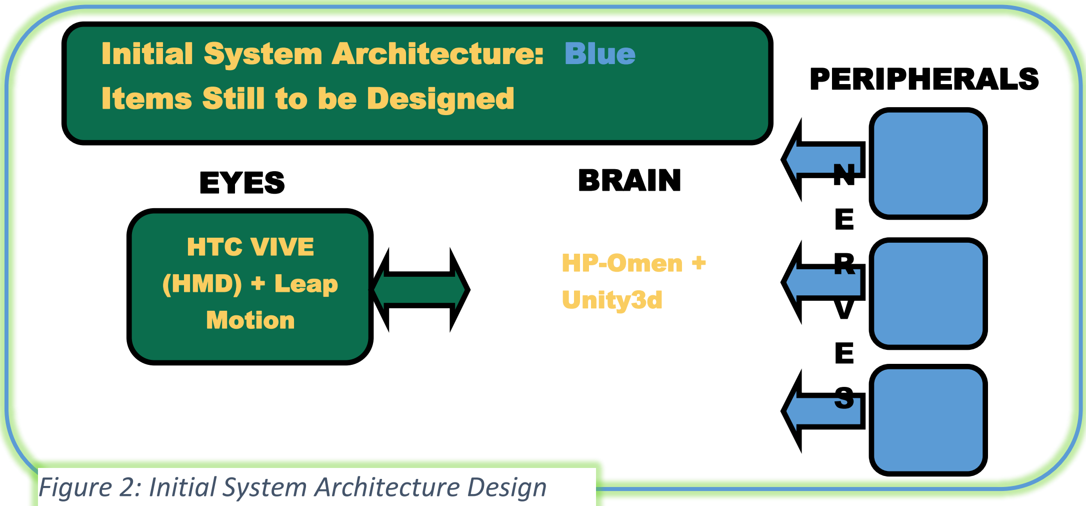


Figure 2: Initial System Architecture Design

With the initial structure of the Core System Architecture now in place (Fig.2) and the elements of the Eyes and Brain selected the next step in the process was to commence designing the peripheral devices. This began with looking at the types of interaction that the system would need to track (Fig.3)

(4.2.3) Interactions in Virtual Space:

The first step taken to identify the types of interaction that the system would need to be able to simulate was to consider examples of day to day activities and the kinds of objects we interact with.

Some examples of the interactions initially considered were:

- Switching a light on/off
- Turning a dial on a speaker, safe or radio
- Opening closing doors
- Operating door handles
- Picking up Cups and Cans
- Interacting with computers – keyboard/mouse
- Writing and holding pens
- Interacting with furniture
- Interactions with other people
- Driving, turning a steering wheel

The main types of interaction that occur in real-world situations were identified as pull, push, turn and pick up/put down. For comparison to these real-world interactions, a selection of video games from various genres was analysed to assess the types of interaction that are traditionally simulated. The games used for the analysis were, The Division 2 (Ubisoft, 2018), Tomb Raider (SquareEnix, 2018), Zelda (Nintendo, 2017)and Wolfenstein

(Bethesda, 2018), screen captures from these games can be seen in Fig.3.

Figure 3: Screen captures showing the types of action and object found in most video games

This analysis revealed the following interactions were critical to any simulation:

- Pulling: a lever
- Pick up: a sword
- Push: a button
- turn: a dial
- turn: a wheel
- Pick-up a gun

The findings from the comparison of interactions in the real-world and video games revealed that the majority could be grouped into specific types of activity.

Group 1: Interactions with objects that are fixed in location and require a switch or button mechanism.

Group 2: Interactions with inanimate objects that can be picked up and moved, but don't require any secondary tracking.

Group 3: Interactions with objects that can be picked up and moved that also will require additional secondary sensors.

Group 4: Interactions with Objects at a fixed location with a pivot point, sliding or rolling mechanism.

From further analysis of these groups, it became evident that to simulate most interactions required detecting four states: On/off, Collision(picked-up/dropped), position in 3D space and rotation. It was also noted that all groups could be separated into two parent groups, Static and Mobile.

- A Static interaction is defined as, an object in the virtual space that has a fixed location, with either a button/switch or a pivot point (hinge), that allows for interaction.
- The Mobile interaction component refers to objects which require position tracking in 3D space as their locations are not fixed.

To create the detection of static interactions the proposed solution was to use a selection of basic electrical components, for example, tactile switches, to detect on/off state, Rotary potentiometer to detect rotation, around a fixed point. To detect the mobile interactions would require the use of an IMU or inertial measurement unit (accelerometer/Gyroscope) and an as-yet-undetermined system for tracking location in 3d space.

For the design process, the development of the sensors would be split into these parent groups, static and mobile, and each one is addressed in a separate design section later in this thesis.

At this stage of the design research process, it was possible to create a basic first iteration of the core system to allow for initial feasibility testing of the selected parts. The objectives of this testing were to establish a basic understanding of how to implement each one and to highlight any unforeseen issues with the initial design concept that could form an impediment on further iterations of the design.

To begin the process of developing the prototype of the core system a series of feasibility experiments were devised to test the suitability of components, the details and findings of these experiments are documented in the following section.

(4.3) Feasibility Testing:

So far, the design process had looked at the elements that make up the whole system. To proceed further, the focus was shifted to looking specifically at the system that would facilitate the control of the peripheral devices, referred to from here on as the Core Nervous System. (Fig.4)

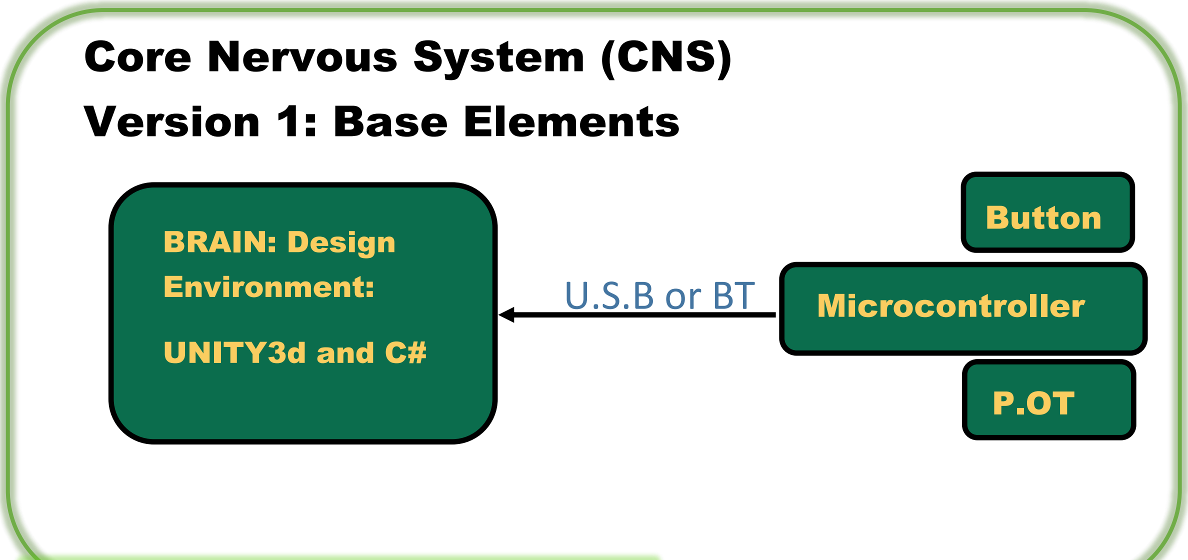


Figure 4: Initial system architecture Basic Elements

The key objectives of the feasibility testing carried out were to test the suitability of the basic elements that form the Core Nervous System (C.N.S.) of the haptic toolkit (Fig.1). Looking at, the microcontrollers' connection to the design environment, communication using Bluetooth modules and testing of a simple button, basic voltage and IMU variable data transfer and response. The outcomes will be to determine that the selected parts are functionally capable and to highlight any possible barriers to the design process that need to be addressed before further development can continue.

To achieve these objectives, five experiments will be conducted, with specific goals set for each one.

- One: To create stable Serial Communication via USB from Unity to Arduino Uno, to gain an understanding of the necessary steps involved, and then to test the effect of varying the data transfer rate has on the lag in the system.
- Two: To establish communication from the Arduino to Unity, To test using a tactile switch to trigger an event in the Unity environment and record any signal drops or loss.

- Three: To Test sending variable data as float or integer value as opposed to simple high/low state used in previous tests. And how to apply the data to control an event in Unity.
- Four: To test sending multiple variable data at once, by implementing an IMU. Also, to establish any limits to using the IMU.
- Five: To Configure Two Arduinos with Bluetooth and components from previous tests and establish communication and transfer of Data, Then assess how to connect to Unity wirelessly.

Full versions of the scripts and the Unity project files used for each of the experiments can be found in Digital Appendix Folder supplied with this document.

(4.3.1) Experiment One:

The purpose of the first experiment undertaken was to establish an understanding of the process of setting up serial communication between a microcontroller and Unity. Once a connection had been created, the effect of varying the baud rate of the serial port on delay in the transference of the signal between devices was observed and a recording of the time lag was taken. This was conducted to establish the optimum data transfer speed to use going forward.

In this test, the circuit was configured as shown in (Fig.5), an Arduino Uno Rev 3 was connected to the P.C via a U.S.B cable, and an L.E.D was connected to the Arduino ground and Digital I/O pin 2.

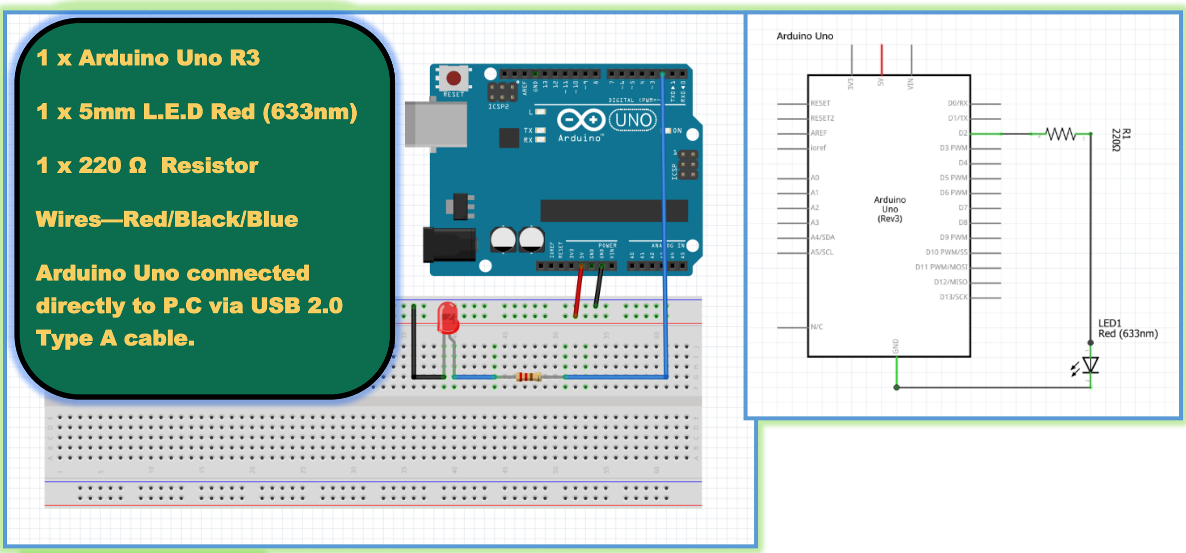


Figure 5 : Circuit 1.

```
void setup()
{
  Serial.begin(115200);
  pinMode(LED_Pin, OUTPUT);
}
```

Arduino C

```
SerialPort sp;

void Start()
{
    sp = new SerialPort("COM8", 115200);

            sp.Open();
 if (sp.IsOpen)
        sp.Write("Serial Port Connected");
```

C#

Figure 6: Serial Port Setup Arduino IDE and Unity

Scripts were then written using C# and Arduino C to allow serial port communication between the two devices. (Fig.6)

One script configured the Arduino to wait for a command from Unity3d, a specific ASCII character "H". The other controlled a basic scene in Unity3d, each time a key in the digital environment was pressed, the command character was transmitted, upon receiving the correct character the Arduino set the state of the L.E.D. to 'HIGH' (On), for a period of 2seconds then off again. Having made a visual confirmation that the system was working the next step was to investigate the effect of varying the Baud Rate of the serial ports and record any lag in data transfer, the aim is to select a speed which offers the lowest response time, as lag at this stage will only be compounded as the complexity of the system increases.

The baud rates (Bps), the rate of data transmitted via the serial port in bytes per second, of both the Arduino and Unity interface were initially set to 9600Bps (or 960 characters per second), then increased to 19200Bps and finally 115200Bps. For each case, ten recordings were made of the time delay between hitting a key in Unity and the

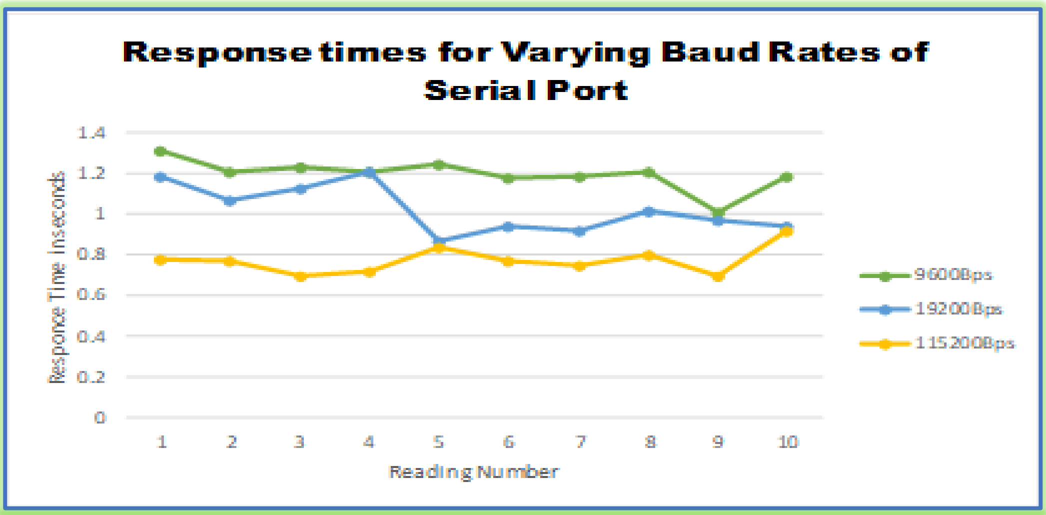


Figure 7: Graph showing the outcome of varying the Baud Rate of the Serial Port - response in sec

L.E.D lighting on the Arduino. Results have been plotted onto the graph shown in Fig.7

The method to record the time delay cannot be considered 100% accurate as they is an element of human error in the data recording. This is due to the fact that to record the delay in response the timmer was manually stopped when the L.E.D was observed to be on, and therefore the results will be affected by the observes personal reflexes. While this affects the overall accuracy in terms of time delay, it is not significant enough to undermind the purpose of the test, the outcome still highlighted the impact of changing the baud rate and as such the rate selected moving forward would be 115200Bps. While higher transfer rates are available due to the intention of transferring data via Bluetooth (B.T.), 115200Bps was chosen as this is the highest recommended transfer speed for B.T.

The average response time for the system was approximately seven-tenths of a second, which when you account for the assumed human error was determined to be sufficiently low enough to proceed with further design work. At this point, a stable connection between the microcontroller and Unity had been created, however, the transfer of data was in the wrong direction, from Unity to the Arduino, therefore, in the second experiment the objective was to establish sending data from the Arduino to Unity.

(4.3.2) Experiment Two:

The objective of the second feasibility test was to establish communication in the opposite direction to that created in experiment One, for the system to work the interactions in the real-world must be transmitted into the digital environment. Once the connection was established, a Unity scene was created in which a game objects visibility was toggled between true/false depending on the signal received from the Arduino. Messages sent from the Arduino were stamped with a count number and then received data was displayed in the Unity console, observations of the received sequence of numbers were then made to ensure that each time the button has pressed a response was detected in Unity. This was conducted to confirm the suitability of tactile switches being used in the toolkit to trigger events.

For this experiment, the circuit was configured, as shown in (Fig.8). The Arduino is once again connected to Unity via U.S.B. A tactile switch had then been added to the system . Connected to set to be active on logic low, the switch is held in a high state through the use of a pull-up resistor of 4.4 KΩ. When the switch is active, a signal is sent to Unity, which in turn triggers the dynamic state of a game object.

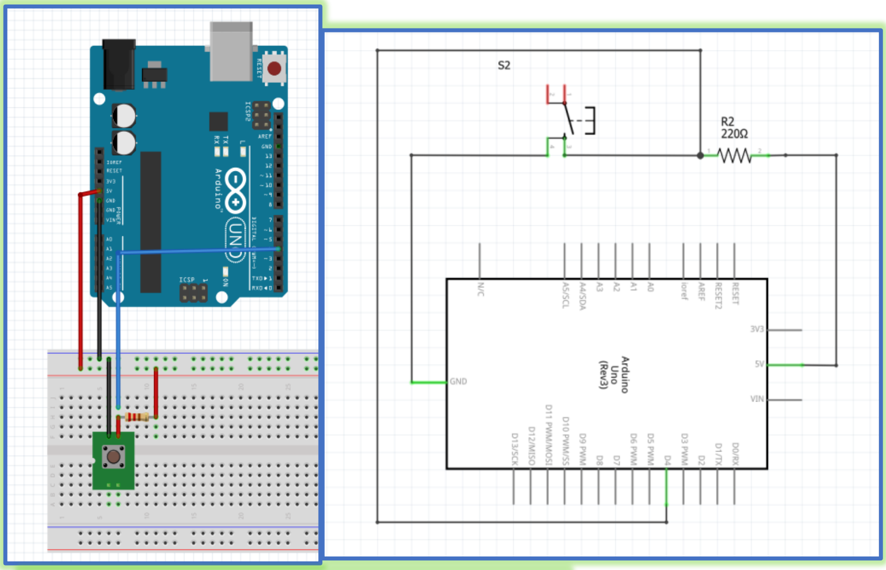


Figure 8: Circuit set up for Experiment two.

An Arduino script was created to read the digital input and send an 'H' for high/On or an 'L' for Low/Off, and a count number so that when received in Unity it was possible to identify each press of the switch and ensure that each signal was correctly received (Fig.9).

```
void loop()
{

  buttonState = digitalRead(buttonPin);
  if(buttonState == LOW && setActive == false){
  Serial.print("H");
  Serial.println(count);
  setActive = true;
  count++;
  delay(1000);
  }
  else if(buttonState == LOW && setActive == true){

    Serial.print("L");
    Serial.println(count);
    setActive = false;
    count++;
    delay(1000);
  }
```

Figure 9: Arduino Code to send trigger data to Unity

Each time the switch in the real-world was pressed, the resulting received message in Unity triggers the state of a simple game object from visible to not visible and displays the received message in the console window. To establish that the signal was being consistently correctly received the switch was triggered 30 times and each outcome was recorded, full details of the results can be found in the design appendix (P: 29-32)

For the system to correctly operate a new control script was also created for Unity, this script contains two crucial new functions. A method of reading the incoming data was created with a try and catch so that if no signal is received, the system does not hang up waiting for one. (Fig.10)

```
void receivedString() // method to read incoming serial data
{
    try
    {
        receivedDataLine = Arduino.ReadLine();
        Debug.Log("working 2");
        newData = true;
    }
    catch(TimeoutException e)
    {

    }
```

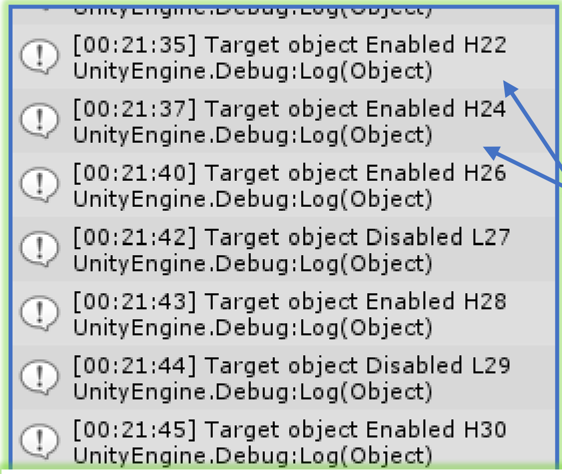
Figure 40 : Method to read incoming data in unity

Next of function to handle processing the incoming data was created, that triggered events in Unity when the correct data was received. (Fig.11)

```
void showNewData()  //Method to trigger events when new data receieved
{
    if (newData == true)
    {
      if (receivedDataLine.Contains("H"))
       {
           target.SetActive(true);
           Debug.Log("Target object Enabled " + receivedDataLine);
       }
       if (receivedDataLine.Contains("L"))
       {
           target.SetActive(false);
           Debug.Log("Target object Disabled " + receivedDataLine);
       }

       newData = false;
    }
```

Figure 11: Shows function for turning received signals into actions in the Unity

When a signal is received by Unity, the first method reads in the incoming data from the serial buffer. Then a Boolean variable is set to true, this, in turn, fires the ShowNewData method which checks to see what has been received and triggers events in Unity accordingly and displays the incoming data in the Unity console window.



Of the 30 signals sent from the Arduino to Unity only two were dropped, L23 and L24,

Confirming that this approach to sending data via a tactile switch would be sufficient for the purposes of this project

Figure 12: Results of triggering switch 30 times, shows two signals were dropped.

As the results in (Fig.12) show on two occasions, the signal was not received in Unity giving a success rate of 93%. Upon review of the Arduino code implemented, the drop in the signal can be accounted for as there is a coded delay of one second after each button press. Put in place to stop the system overloading, this resulted in the signal being dropped, as if the switch is triggered repeatedly in too quick a succession, then this would prevent it from being detected. In future iterations, this will be addressed by removing the delay.

Having now established communications in the desired direction Arduino to Unity. Confirmed that using tactile switches would be a suitable solution for the toolkit, and developed an appropriate set of scripts for sending and receiving data. The next step was to test the transmission of more complex data structure. As so far only a single fixed 'H' or 'L' value has been sent, but to create the remaining tools for the kit a way of transmitting variables that contain data stored as floats or integer values that can continuously change must be created. This is addressed in the next two feasibility tests.

(4.3.3) Experiment Three:

In the third feasibility experiment, the objective was to create an understanding of how to transmit dynamically changing variables. As so far, only a single fixed ASCII character has been sent. Since the overall design objective will involve more complex data structures being transferred, a system for enabling this process requires investigation. To test this, a new circuit was set up using a rotary potentiometer (P.O.T.) to send a variable data stream to Unity. When received, the data will be processed and then applied to a game object. In this instance, the received value of resistance from the P.O.T. is used to control the radius of a sphere so that the effect of changing the value can be seen. The received data will also be printed to the Unity console. The secondary objective is to test the reliability of the P.O.T. as an input detection tool, as the intention is to use P.O.T.'s to track the rotation of static interactions.

The circuit set-up is shown in (Fig.13), a 10KΩ rotatory variable resistor connected to GND and +5v of Arduino, with the slider (middle pin) connected to analogue pin 0 (A0) of the Arduino, and then this is connected to Unity Via USB.

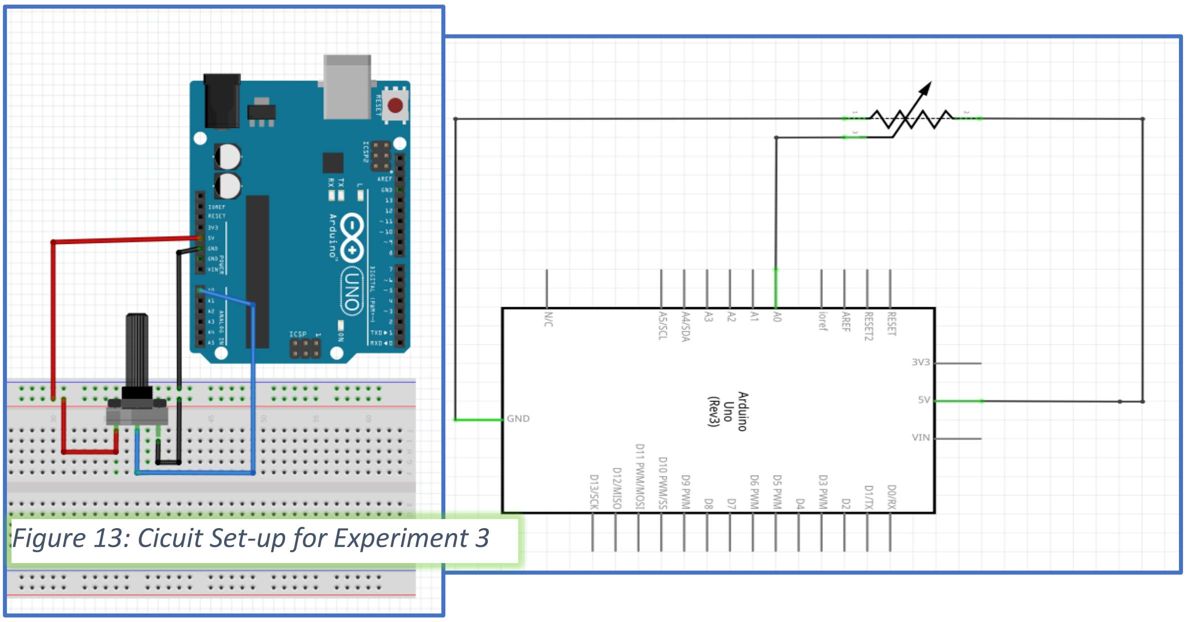


Figure 13: Cicuit Set-up for Experiment 3

With the circuit set-up and serial connection established the Arduino was configured with an updated script to send the value of the P.O.T. to Unity. Initially, this presented a new design problem. The value of the P.O.T. ranges from 0-1023Ωs, but the serial write function in Arduino can only send a single byte (0-256). There were several possible ways to handle this. The value of the Pot could be stored as an integer variable which Arduino handles in 16bit, two bytes; this data could then be split into low and high bytes, transmitted and then converted back into an integer when received. The downside to this is it adds calculations that must be computed by the microcontroller, which given that the eventual solution could involve multiple sensors that require splitting and transmitting could potentially add latency to the system. The second approach would use a conversion to scale the range of 0-1023 to 0-256; this would once again be reversed when received by Unity. However, this approach will suffer from the same issues as the first, with the added extra of rounding errors from the conversion process resulting in inaccurate data being received.

Thankfully, there was another method that could be applied instead of using the Serial.Write() function; the system could continue to use the Serial. Print () option, which as already seen transmits the data as strings. With this approach, the value read from the analogue sensors is split into its characters and sent as a string. This involves very little onboard computation. Allows for the full range of the P.O.T. to be used, as the data can easily be converted from a String to a Float or Int within the Unity environment, which benefits from far superior computational power and will not cause any lag/latency. The function to handle this can be seen in (Fig.14).

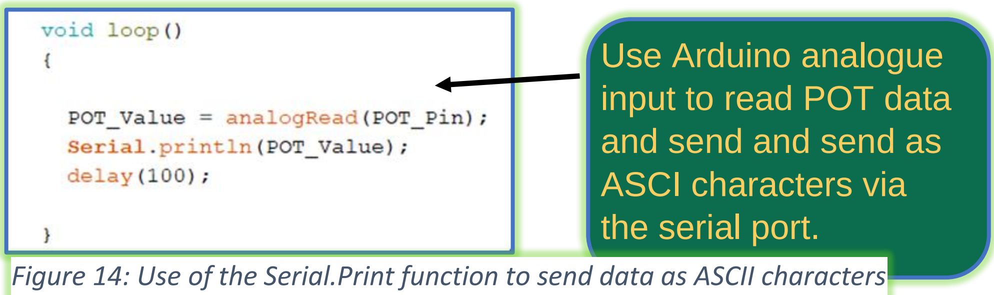


Figure 14: Use of the Serial.Print function to send data as ASCII characters

The advantage of taking this approach is also that it remains compatible with the previous method of sending data from the triggers. With a solution to transmitting the variable data in place, the ShowNewData function in Unity also needed to be updated to convert the received String into a useable numerical value again. (Fig.15)

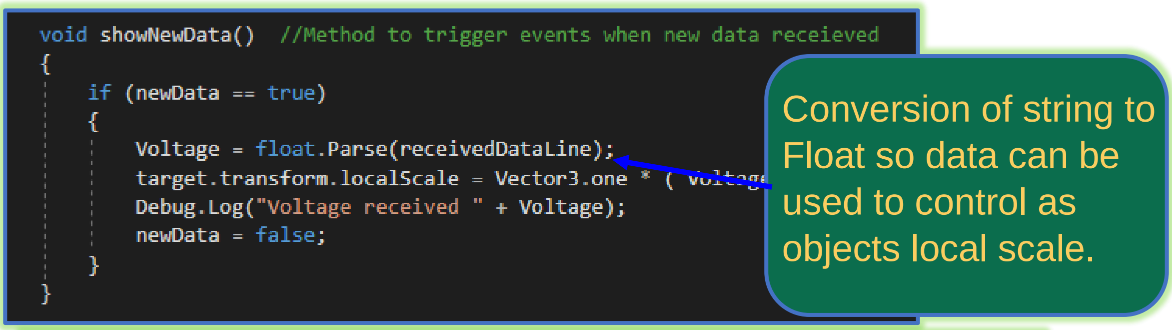


Figure 15 : Scirpt to convet Sting to a Flaot in Unity, using the buily in parse function.

This was achieved by utilising one of the many important builtin functions of Unity, Parse, which allows data to be passed from one variable type to another with ease.

With both the microcontroller and Unity set-up a test of the system was conducted, the value of the P.O.T. received by Unity, once converted, was applied to the transform of the local scale of a sphere game object in the environment, and as the value of the P.O.T. was variated as too was the size of the radius of the sphere. (Fig.16)

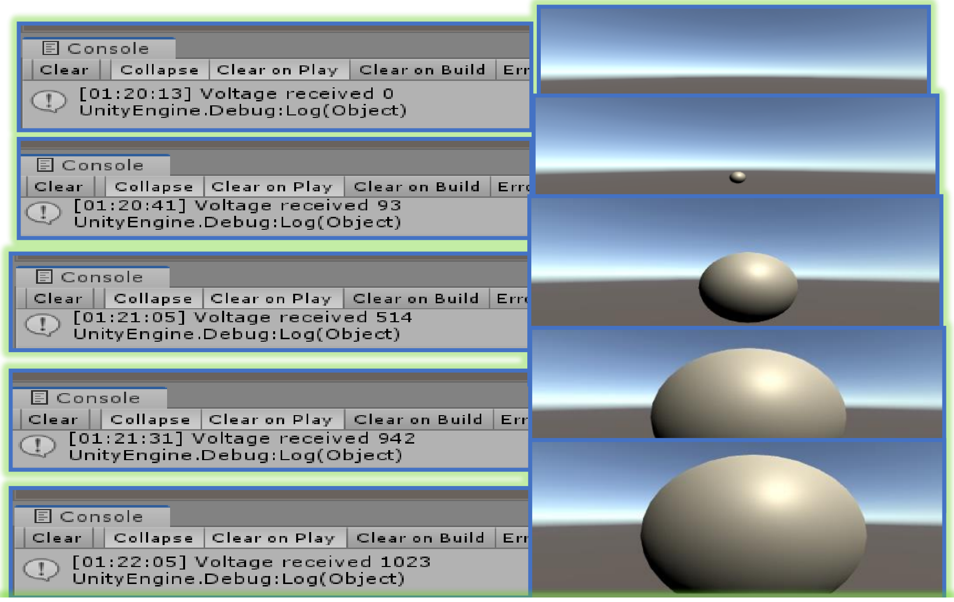


Figure 16: Outcome of varying resistance value and application of it to a game object in unity.

As can be seen in the outcome was as desired the value of the P.O.T. was successfully transmitted, received and converted into a usable value in Unity, and applied to the object.

This outcome was not achieved without some errors being created, a review of the data captured highlighted a recording of 11 miss reads in the process where the data received was not in the correct form for Unity to convert to a Float. (Fig.17)

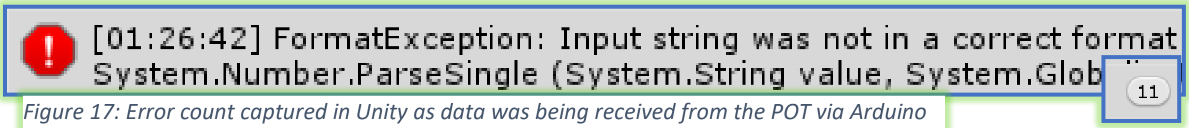


Figure 17: Error count captured in Unity as data was being received from the POT via Arduino

It was also observed that there is a small dead zone in the response of the P.O.T. the value stays at zero for the first few degrees of rotation. While neither issue caused severe problems in this experiment, both will need to be investigated further. As repeated miss reads will cause the system to hang or crash, and will generate another source of system Lag. Also, if the P.O.T. is to be used to track rotation as intended, the dead zone will need to be accounted for. Both problems are addressed and successfully corrected later in the design process.

(4.3.4) Experiment Four:

In the fourth experiment, the objective was to build on the previous test by introducing multiple dynamic variables at once. This was achieved through the use of an Inertial Measurement Unit (IMU), the option selected offers 9 degrees of freedom, provided by an onboard accelerometer and gyroscope, outputting data as X, Y, and Z values of rotation. The intention of the experiment carried out was to find a way of sending three variables at once to Unity, then splitting the data and applying each value to the corresponding x,y,z rotation of a game object in Unity. As a secondary objective, this experiment will also serve as a test of the stability of the IMUs output to see if the selected module will be sufficient for accurately relaying rotation information.

The set-up of this experiment can be seen in the circuit diagram (Fig.18). The IMU is connected to the Arduinos +5v and ground connection, the S.D.A. (Serial Data) is then connected to the S.D.A. port of the Arduino, and the S.C.L. (Serial Clock) to the S.C.L. of the Arduino.

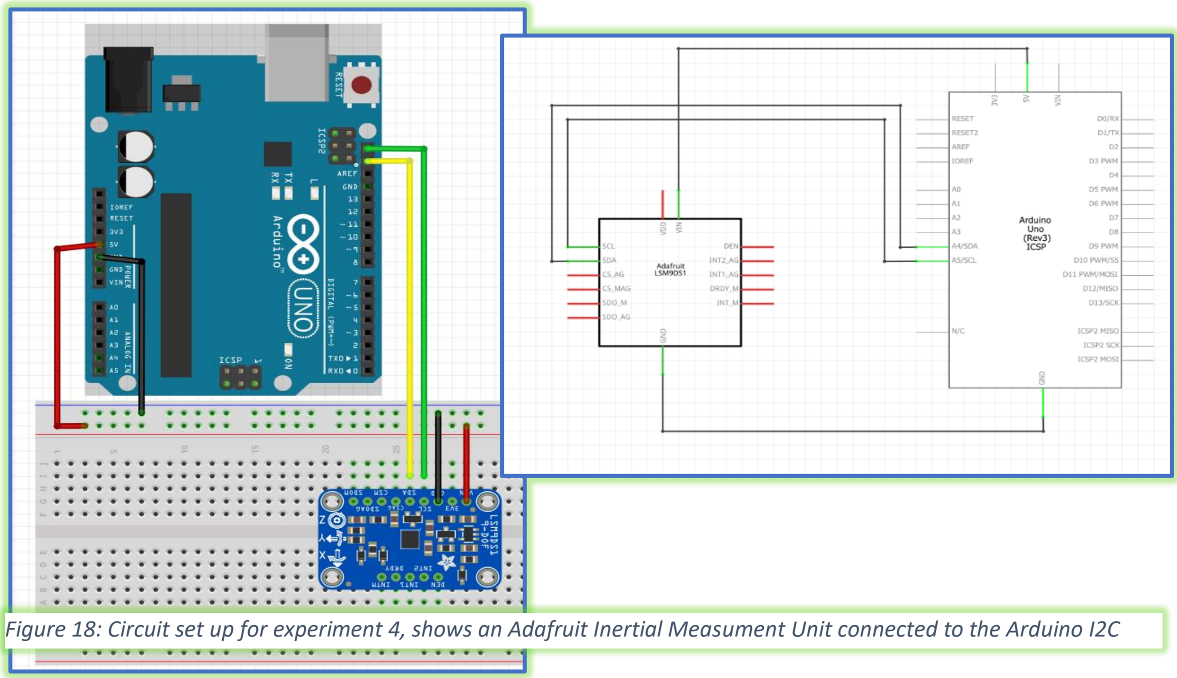


Figure 18: Circuit set up for experiment 4, shows an Adafruit Inertial Measument Unit connected to the Arduino I2C

The S.D.A. and S.C.L. ports are part of the Arduino I2C bus. I2C is a simple way of connecting single or multiple devices and sensors to microcontrollers, as the connection can be established with just two wires. The S.D.A. (Serial Data) carries the data to be transmitted. At the same time, the S.C.L. (Serial Clock) synchronises the data transfer between the devices on the I2C bus and is generated by the host device, in this case, the Arduino. To read the orientation data outputted by the IMU pre-configured software libraries are implemented. This is one of the main advantages of creating with the Arduino IDE as many/most manufacturer of components have also produced support libraries to simplify implementation. The specific libraries utilised in this experiment were, Adafruit_Sensor, Adafruit_LSM303, Adafruit_L3GD20 and Adafruit_9DOF. These provide pre-configured methods and functions to read the IMUs output.

Having constructed the circuit and installed the relevant libraries into the Arduino IDE, Three functions were created, to output each of the desired variables, X, Y and Z orientation. Fig.19 shows an example of these functions.

```
void xAxis(){
  sensors_event_t accel_event;
  sensors_vec_t   orientation;
  accel.getEvent(&accel_event);
  if (dof.accelGetOrientation(&accel_event, &orientation))
  {
    Serial.print("X");
    Serial.print(orientation.roll);
  }
```

Outputs the orientation data, preceded by the corresponding axis label

Figure 19: Example of function to output the X-axis rotation data, using adafruits Sensor Libraries

These functions provided the first part of the solution to output multiple variables at once, by adding the prefix of the axis label to each reading when the data is received in Unity it can be easily identified and applied to the desired target. However, as not, all the components in the toolkit will have clear labels and to counter issues with sending multiples of the same data, for example, more than one active IMU in a scene. The second level of separation was added in the form of start and end markers for each packet of data sent. (Fig.20).

```
void loop(void)
{
  Serial.print("<");
  xAxis();
  Serial.print(">");
  Serial.print("<");
  yAxis();
  Serial.print(">");
  Serial.print("<");
  zAxis();
  Serial.print(">");
}
```

The greater than > , and less than < are used a starting and ending markers

Figure 50: Example code showing the use of start and end markers to separate each variable.

To receive the data packets at the Unity side of the system, the ShowNewData script was modified again to reflect the new data structure as was the method for reading incoming data from the serial port, and a new function to handle incoming data was created.

The read method was converted from reading a line as a String to reading the individual characters received and storing them in a Char variable. (Fig. 21)

```
public Char receivedString() // method to read in
{

    char receviedData ;

    try
    {
        receviedData = (char)Arduino.ReadChar();
        Arduino.BaseStream.Flush();
        return receviedData;
```

Figure 21: Updated received data method, reads individual charaters as opposed to whole strings.

When data is received now, it is parsed into the new received with start and end markers function. (Fig.22)

```
void RecvWithStartEndMarkers() //method to split m
{
    bool recvInProgress = false;
    byte ndx = 0;
    char startMarker = '<';
    char endMarker = '>';
    char rc;
```

```
while (newData == false)
{

    rc = receivedString();

    if (recvInProgress == true)
    {
        if (rc != error)
        {
            if (rc != endMarker)
            {
                recivedChars[ndx] = rc;
                ndx++;
                if (ndx >= numChars)
                {
                    ndx = numChars - 1;
                }
            }
```

```
            else
            {
                recivedChars[ndx] = '\0';
                recvInProgress = false;
                ndx = 0;
                newData = true;

            }

        }
    }

    else if (rc == startMarker)
    {
        recvInProgress = true;
    }
```

Figure 22: example of the new function to split incoming data by a start and end marker

```
void showNewData()  //Method to trigger events when new data receieved
{

    char[] charToTrim = { 'X', 'Y', 'Z', '<', '>' };

    if (newData == true)
    {
        string display = new string(recivedChars);
        if (display.StartsWith("X"))
        {
            tmpX = display;
            tmpX = tmpX.Trim(charToTrim);
            try
            {
                xValue = float.Parse(tmpX);
            }
            catch
            {

                Debug.Log("string error X");

            }
            xValue = float.Parse(tmpX);
```

Figure 23: Updated Show New data fuction, which now spits incoming data based in the prefix character received.

This shows how the data stored in the char array is converted first into a String, then a series of if statements are used to check the prefix character than identifies which variable the data should be applied to. To reduce miss read or corrupted data in the array, the char trim function is used to remove the prefix character and any other stray characters that may be in the String. Once all non-numerical characters have been removed, it is once again converted to a float variable, this time the value is applied to the corresponding axis of a game objects rotation.

```
target.transform.rotation = Quaternion.Euler(xValue,yValue,zValue);
```

Figure 24: application of IMU data to a unity game object

The outcome of this experiment produced several further design considerations. Firstly the output from the IMU was not stable or consistent even when the IMU was perfectly flat and not in motion. This resulted in the game object jittering which in a V.R environment would affect the immersive experience of the user; also if this IMU were used for example on an object that needed to be aimed accurately, it would be impossible. Secondly, the issue seen in the previous experiment where Unity was unable to convert some data into strings as it was corrupted when it was received became much more apparent. More concerningly caused the system to freeze. It did not crash Unity, only froze the thread reading incoming data.

As the results from this experiment were not satisfactory, further investigation into why this was happening was required. To better understand where the issue was arising a graph of the output from the

IMU was produced to see if the problem was relating to the output itself, or in the way the data was being handled in Unity. (Fig.25)

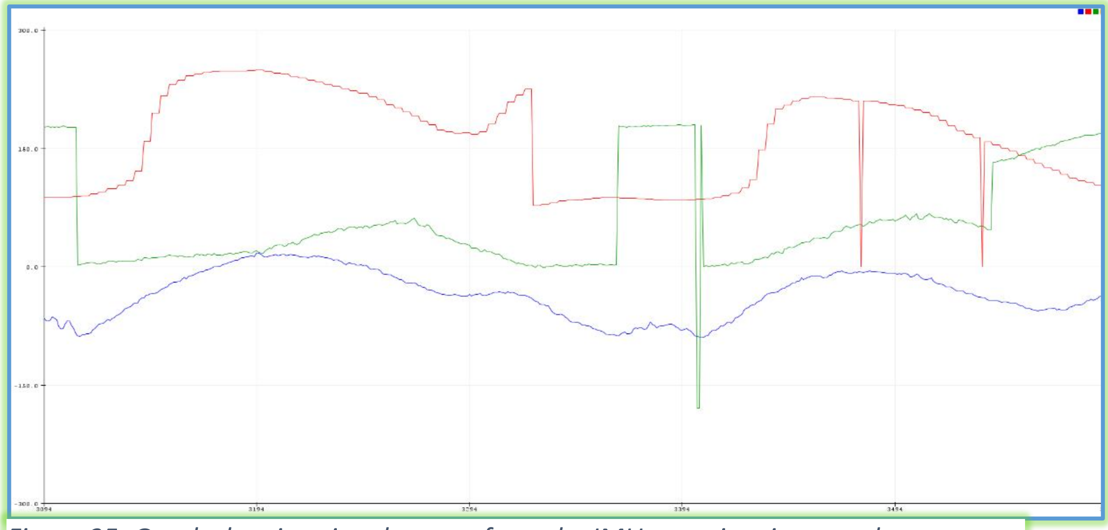


Figure 25: Graph showing signal output from the IMU over time in seconds

The graph shows a plot of the values outputted by the IMU for each axis as it was slowly rotated. The desired result would be three smooth and consistent waveforms, as can be seen, this is not the case for this IMU. The graph shows that the outputted values varied every few seconds, which accounts for the jittering seen in Unity, and the read errors detected that caused the system to freeze, which were caused by the drop-off in output data.

This presents a serious design issue that needed to be addressed if the continued use of IMUs in the toolkit was to be made possible. Initial attempts were made to alleviate the problem by applying smoothing to the signal from the IMU. High and low pass filters were applied to the output, however, to produce a stable signal required a broad range to be used with both filters. The results were that while the output signal was stable, there was also a dead zone that resulted in a lag in response when the IMU was rotated. As the initial attempts to apply smoothing were unsuccessful research into the issue was conducted using online forums. Information gathered online strongly suggested that perfecting an algorithm for accurate real-time smoothing would be complicated and very time-consuming. A suggested alternative to this approach was to change out the specific model of IMU, as the currently selected component was very low-cost, and models that boasted much more consistent results were available.

Sourcing a replacement IMU was the choice that was selected to reduce the amount of time spent trying to clean up the existing signal. The chosen new part was the Bosh BNO055 absolute orientation IMU.

The significant advantage of using this model of IMU is that it has a built-in microcontroller of its own. This onboard processor is explicitly designed to control the output signal. The result is a stable and consistent output of rotation data, solving the problem of the lower cost IMU.

This concluded the feasibility testing of the components to be used as peripheral sensors. These four experiments demonstrated the potential simplicity of using direct serial connections in sending data to Unity. However, there are some drawbacks to this approach. This configuration would require a significant amount of cables and USB ports to achieve control over multiple microcontrollers and would limit the adaptability of the system. Therefore, a wireless solution required investigation. The final experiment looks specifically at the process of configuring a wireless communication system based on Bluetooth.

(4.3.5) Experiment Five:

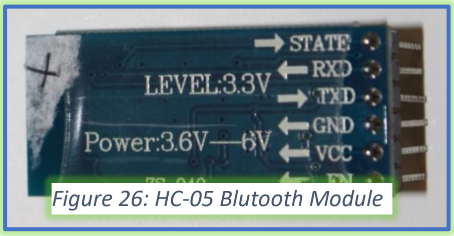


Figure 26: HC-05 Blutooth Module

In the final experiment, two circuits were configured using Arduino Uno's and HC-05 Bluetooth modules (Fig.26), to test their wireless transmission capabilities. The principle was to create a circuit that when a button on either controller was pressed the other opposing devices' L.E.D would light up, having received a signal via Bluetooth, demonstrating the establishment of two-way wireless communication.

The HC-05's were chosen based on their cost per unit, simplicity of implementation, both in terms of coding and circuit design, along with their ability to be configured in one of two modes. When set-up in Master mode, an HC-05 module can initiate a connection with another device. Whereas in Slave mode, they are only capable of receiving incoming signals. For example, a games console or tablet would be the master devices, and B.T. enabled headphones, speakers or controllers would be the slaves.

To begin this experiment, the HC-05 modules required pairing to establish communication. To do this, they needed setting to their AT (Attention Command) mode to allow access to their default command settings. This is enabled by adding a connection from the EN or

STATE pin of the HC-05 to the +5 volts of the microcontroller, pulling this connection HIGH and therefore activating the AT mode commands. Also, the links to the Arduino are reversed from their normal operating state. To transmit under regular operation, the R.X. (receiving) and TX (transmitting) pins are connected from the HC-05s to their opposites on the microcontroller. Still, to program them in AT mode, this is reversed. Once the device is in AT mode, it is possible to configure and retrieve several key operational variables using the Arduino IDE's or any other serial control software.

The primary information required for operating the two devices as a pair is their unique M.A.C. (Media Access Control) address, these comprise of six sets of two-digit, hexadecimal number separated by colons for example – 98:d3:81:fd:88:c8. This allows each device on a network or multiples of the same device to be easily distinguished from one another. For this experiment, the device NAME, ROLE, BIND and BAUD rate options were also accessed and where required altered.

- NAME: was reset to Master/Slave respectively, for ease of future identification.
    - AT+NAME = Master
- BIND: was set for the Slave unit to the Mac address of the master module.
    - AT+BIND = 98,d3,81,fd,88,c8
- BAUD Rate: was increased to 115200, allowing for a potentially more significant sized data packet to be sent in a single burst.
    - AT+UART = 115200
- ROLE: This was set to Master(1) or Slave (0) as required.
    - AT+ROLE = 1/0

In this instance, the Master BT was configured to an Arduino Uno with an L.E.D and a button attached; this was then synced to the

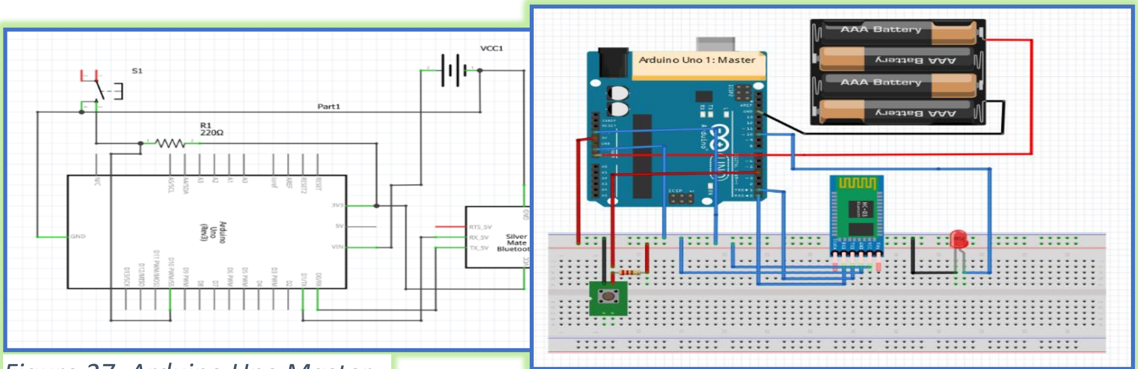


Figure 27: Arduino Uno Master

second Uno with the same configuration and the Slave HC-05. (Fig.27/Fig.28).

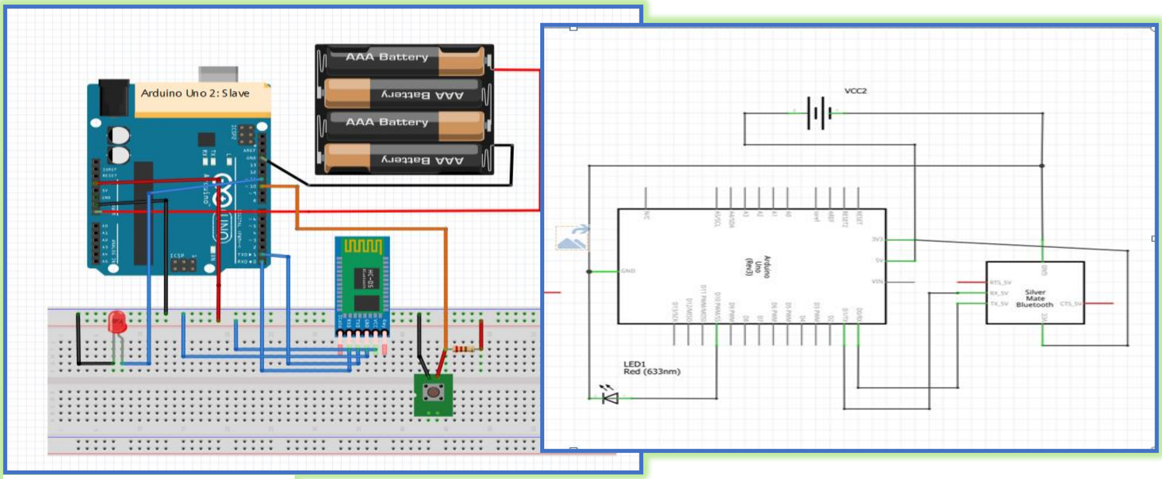


Figure 28: Arduino Slave

Once the Bluetooth was configured, and the relevant scripts were loaded onto the respective Arduino's, the outcome was as expected. A connection between the two devices was established, and it was possible to control the L.E.D's wirelessly. A further test was conducted to determine a maximum range, by simply increasing the slave's distance from the master unit until the connection was lost, this distance was consistently discovered to be approximately 4m in range.

The outcomes of this experiment outlined several further design considerations. Firstly, it is only possible to connect a single HC-05 slave to a master; multiple connections require disconnecting and switching BIND addresses, which, while possible, would add a substantial amount of computing time. Secondly, the Arduino Uno is limited to having only one hardware serial port, preventing the use of multiple HC-05's on a single board, which would hypothetically address the first problem. A potential solution to this would be software-based using virtual serial ports. However, this also would increase processing time. There is another hindrance to only having a single serial port that also needs consideration. That is that to update the software on the control board; you would be required to physically disconnect the Bluetooth controller from the Arduino as the serial connection is shared with the USB port. This, again, would impact the overall efficiency of the system.

As already mentioned increase in computing time directly translates to a potential lag/latency in the virtual environment, which would be detrimental to an immersive experience, therefore, ensuring any solution is optimised for efficiency is crucial. For this reason, an alternative approach to connecting multiple devices to a single host device (Brain) required investigating.

To establish control over multiple peripheral devices via Bluetooth (B.T.), several solutions were considered. Initially, the most evident resolution appeared to be to directly connect to the control devices internal B.T. controller of the P.C. While this approach would eliminate the requirements of multiple USB connections, it also raises a potential issue that severely limits the effectiveness of the system. The manufacturers' recommended number of simultaneous B.T. connections to any single P.C or Mac to avoid system instability, is 7. If you factor in that most users will potentially have B.T. enabled keyboards, mice, speakers, headsets and phone already connected, six connections are already spoken for, this leaves only one available for the new peripheral. As this would essentially render the proposed system functionally useless, finding a solution to this problem was an immediate priority.

The proposed solution to this was to develop an additional microcontroller layer to the original Core Nervous System architecture to handle the connection of B.T. devices externally and forward data to the Brain via a single serial link. The intended advantage of this would be two-fold. One that setting up serial Communication via USB in different design environments is more straight-forward than configuring multiple B.T. connections, thus ensuring the Core Nervous System remains adaptable, and implementation remains straight forward. Two, by creating an external control for the B.T., you eliminate any issues of conflicting with the limitations of the operating system.

The design and implementation of this proposed solution are detailed in the first section of the prototyping chapter that immediately follows on from this current chapter.

(4.4) Outcomes Of Feasibility Testing:

The objectives of the feasibility experiments carried out were to gain knowledge and understanding of how each of the primary components selected for the Core Nervous System (C.N.S) could be implemented as tools for the Haptic Prototyping Toolkit (H.T.P)—assessing each one, based on its suitability for use and highlighting any further design considerations. In the process of completing these tests, two of the initial concept design questions have now been addressed.

- D1. Sourced from pre-established hardware where possible or constructed from easy to source pieces
- D2. Built on a commonly accessible and where available free to use development platforms.

It has now been shown that building the H.T.P using pre-established hardware, i.e. Microcontrollers, tactile switches, rotary P.O.T.'s, IMUs and Bluetooth modules, will be possible. Also, the experiments demonstrated that creating the H.T.P control software can be entirely completed using free to use design and development platforms, i.e. Unity 3d and Arduino IDE.

The research and experimentation also highlighted the need for further design work to provide solutions to some of the issues discovered. These are:

- Creation of an intermediary control device to handle multiple peripheral devices.
- Refining a system to catch and control, or irradicate errors, relating to corrupted or partially received data packets.
- Further testing of the newly selected IMU module.

Finding design solutions to these problems will be addressed in the next phase of the development process.

(4.5) Conclusions from Technical Research:

The overall purpose of the technical research chapter was to show the process that was conducted to select the components that would make up the tools of the H.T.P. To gain an understanding of how to proceed into the prototyping design phase. From work carried out so far, there are now a set of components that have been proven to be suitable for use in the H.T.P, to move the design process forward these components required developing into their respective prototype states.

To begin the prototyping phase, a set of mini design briefs that compliment the main brief already established were created. These briefs would be used to form the structure of the prototyping process, and are as follows.

- **Phase One: Control Interface:**

  Address the issue of not being able to connect multiple devices using direct Bluetooth connections to the Brain (host P.C), and due to the limitations of available serial ports on the Arduino Uno, not being able to solve this by using multiple BT HC-05 modules. The proposed solution, as outlined in section (4.3.5), is to design and develop an intermediary control device that handles all incoming connections and forwards the received data to Unity.

- **Phase Two: Static Interactions:**

  To design a system that uses the tactile switches and P.O.T.'s to track and relay static interactions. As defined previously, a Static interaction is an object in the virtual space that has a fixed location, with either a button/switch or a pivot point (hinge), that allows for interaction.

- **Phase Three: Mobile Interactions:**

  To design a method of tracking mobile objects rotation in 3D space using the IMU and an as yet defined system for monitoring their position in 3D space. Mobile interaction is defined as interaction with objects that do not have a fixed location in the environment.

- **Phase Four: Creating a complete System:**

  Combing the prototypes generated in phases One to Three into the first iteration of a complete Haptic Prototyping Toolkit.

- **Phase Five: Design the Evaluation Environment:**

  Design and develop an evaluation scenario that will facilitate user testing of the system. To provide a platform for data collection to investigate any indications that this approach to Haptic Interaction has a positive or any kind of impact on the end-users sense of Immersion and Presence.

Prototype Phase One (5)

(5.1) Introduction:

The first phase of the prototyping process was to address the design problems surrounding the control of multiple peripheral devices. As defined in the Phase One design brief.

- Address the issue of not being able to connect multiple devices using direct Bluetooth connections to the Brain (host P.C), and due to the limitations of available serial ports on the Arduino Uno, not being able to solve this by using multiple BT HC-05 modules. The proposed solution is to design and develop an intermediary control device that handles all incoming connections and forwards the received data to Unity.

The original architecture of the Core Nervous System(C.N.S.) was designed around a direct wireless/wired connection from the peripheral devices to the Brain (Host P.C), however, as discovered in the feasibility testing, there are issues with this approach that affect the implementation of this system. As uncovered in the previous Chapter, these issues presented design problems that required resolving before the process of development could continue. In the technical research chapter (4.3.5), the concept of creating a new layer of the C.N.S was proposed, and the redesigned C.N.S. architecture accommodating these planned improvements can be seen in (Fig1).

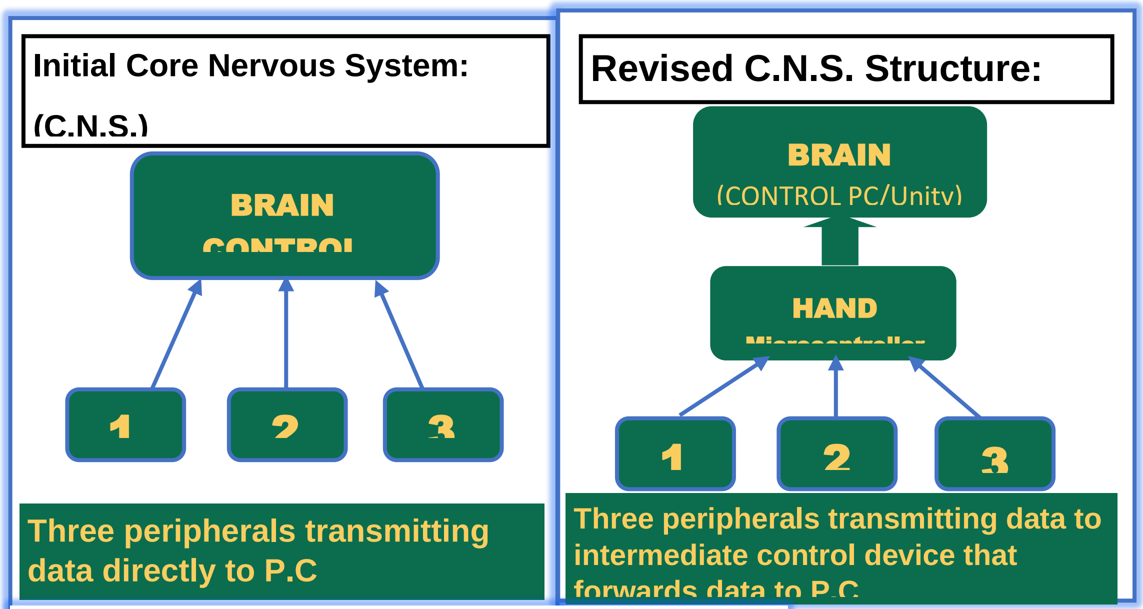


Figure 1: Initial and Revised Core Nervous System Architecture

As the infographic shows, the planned solution was to add an additional microcontroller layer to the original Core Nervous System architecture. To externally handle the connection of Bluetooth devices and forward data to the Brain (Host P.C) via a single serial port connection.

The Microcontroller that would be used for this new Hand element was selected from the research carried out of available boards (see Design Diary appendix (P.8-15)), and based on the following criteria:

- Low-cost board, that was readily available to all end users.
- Have a suitable level of processing power, to prevent the creation of a bottleneck in the data flow.
- Must have the option of multiple hardware serial ports.
- Have a small form factor, so that it would require minimal space in the end-users work environment.

The solution selected was the PJRC Robotics board Teensy 3.2. (Fig2.)

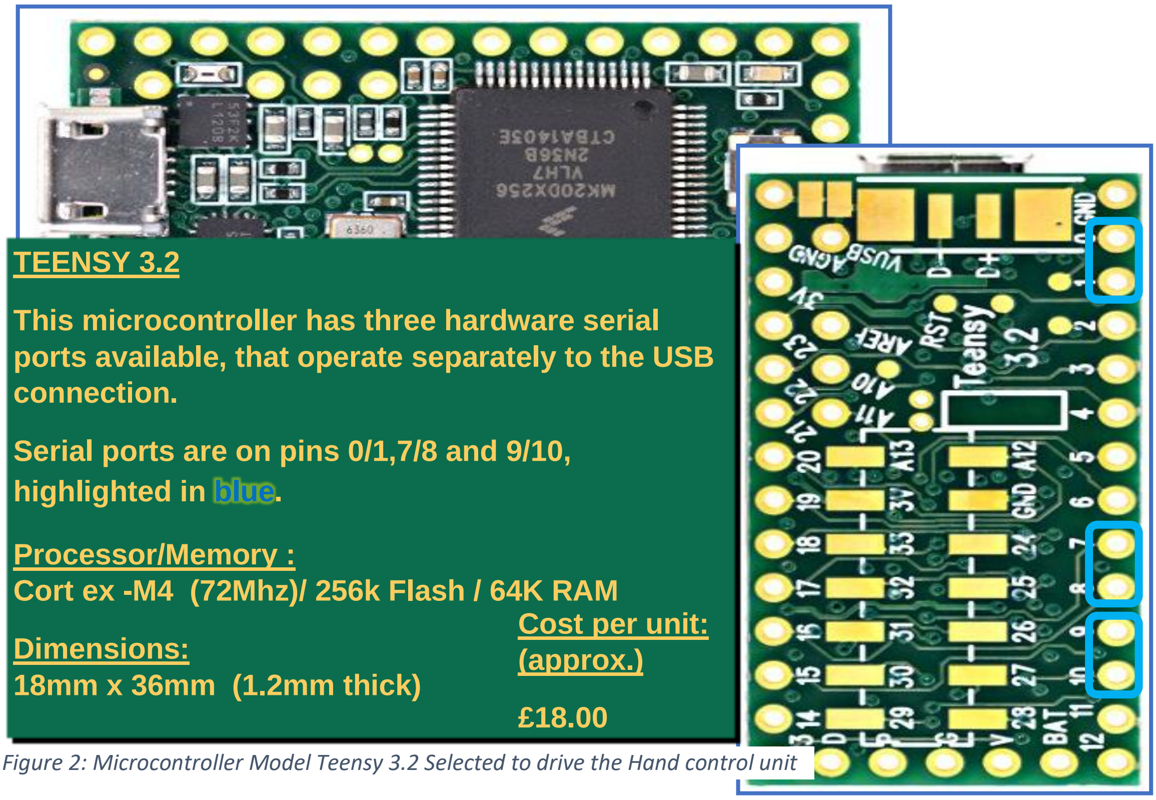


Figure 2: Microcontroller Model Teensy 3.2 Selected to drive the Hand control unit

By integrating the Teensy 3.2 into the Core Nervous System (C.N.S.), it potentially addresses all the questions raised in Phase One.

The Teensy 3.2 has significant advantages over the Arduino Uno, processor speeds of up to 96Mhz (overclocked) v's the 16Mhz of the Uno and four hardware serial ports, three that operate independently of the main USB interface. At the same time, the Teensy still benefits from the use of the Arduino IDE, with the addition of the TeensyDuino Library. (PJRC, 2019). This allows for its' deployment as a host controller that is compatible with the range of Arduino and Teensy boards, that can all be configured in a single programming environment

Access to independent serial ports allows for direct USB communication with the Brain (host P.C), while still allowing for separate Bluetooth connections. The fact that there are three ports also means that three master HC-05 modules could be connected and run concurrently, allowing for the control of three peripheral devices, while still only requiring a single USB connection to the host. This phase of prototyping aimed to test the solution by developing a prototype of the Hand Peripheral Control Device. Having developed an outline design criterion, built around the Teensy 3.2 and HC-05 modules, an initial circuit design was developed (Fig.3). The concept is to connect three Bluetooth (B.T.) HC-05 modules to the three serial ports and link them to three L.E.D.'s to signal when a connection with a peripheral has been established. Each of the HC-05's would also be paired to a unique Slave device. The Microcontroller will then act as a data forwarding hub, collecting incoming signals from each B.T. Slave peripheral, compiling them and sending this information on to the Brain (Host P.C) to be processed into interactions in the virtual environment.

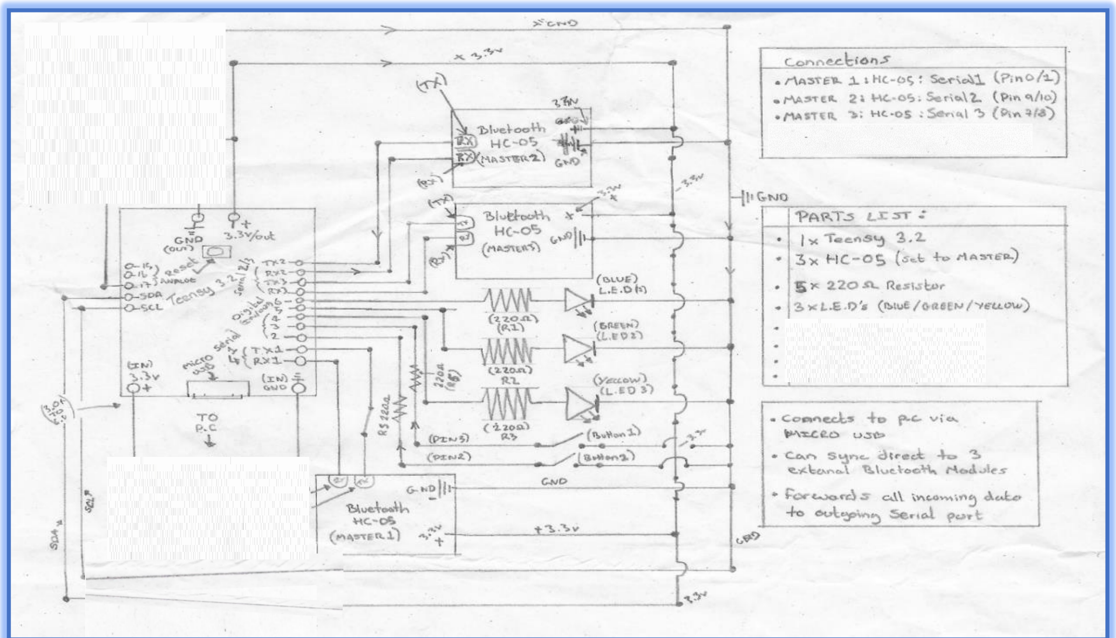


Figure 3: Fist Design of prototype Hand Device Circuit Layout

(5.2) First Prototype: Hand Development.

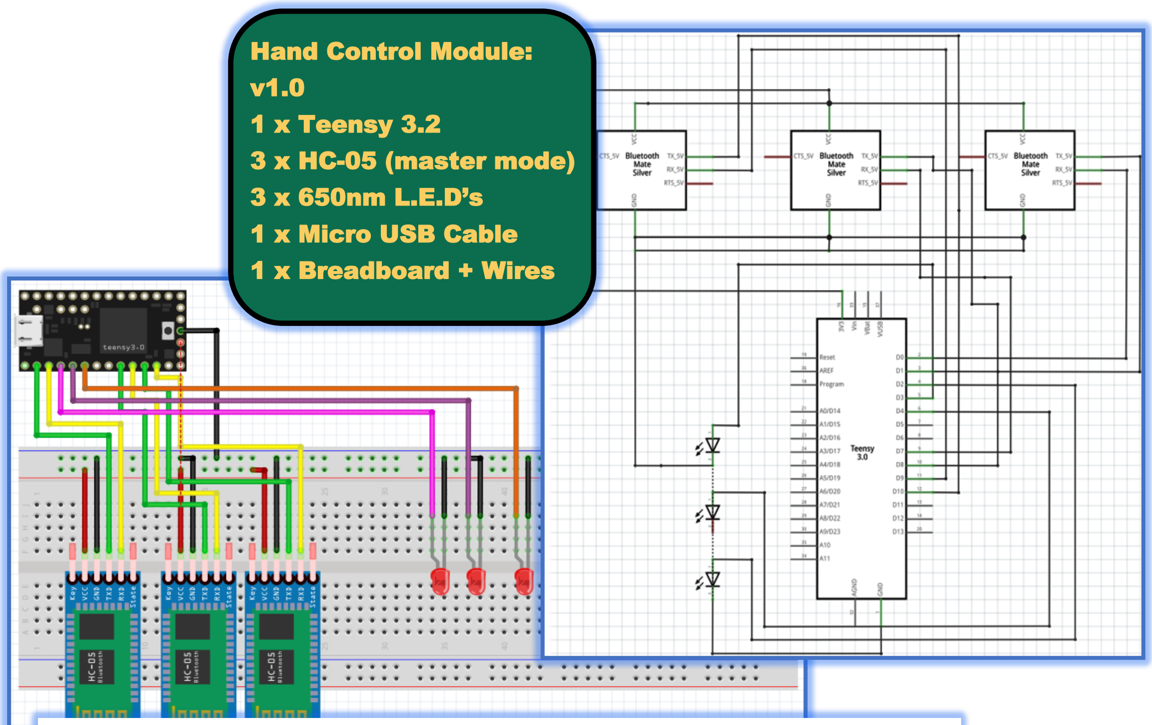


Figure 4: Refined Breadboard layout and circuit diagram for Hand Version 1.0

To create a functioning prototype system, a three-stage approach was adopted. In stage one a replication of the circuit design was built on a breadboard set up (Fig.4), and the components were configured with the relevant control software. The second stage involved testing the system's ability to forward incoming data efficiently. Then, once confirmation of the performance of the system had been confirmed, the final step was to build a physical prototype of the Hand device.

(5.2.1) Initial Circuit Prototype and Component Configuration:

The first step in the configuration process was to set up each pair of HC-05 modules into their respective Master and Slave modes. This was accomplished using the same methods outlined in Chapter (4.3.5). While each module was in its AT command mode they were renamed to HAND1/2/3, and Slave 1/2/3 as required, the Baud rates were set to the previously established speed of 115200Bps, and each pair was bound to each other using their unique Mac addresses. The Binding processing being applied here is to ensure that there is a level of control over the flow of data from the peripherals to specific Master devices. This control was included pre-emptively to ensure that later in development incoming signals of the same data structure, for example, multiple instances of the same type of interface module connected to individual cases of peripheral devices, could be easily identified and processed correctly. (Fig 5.)

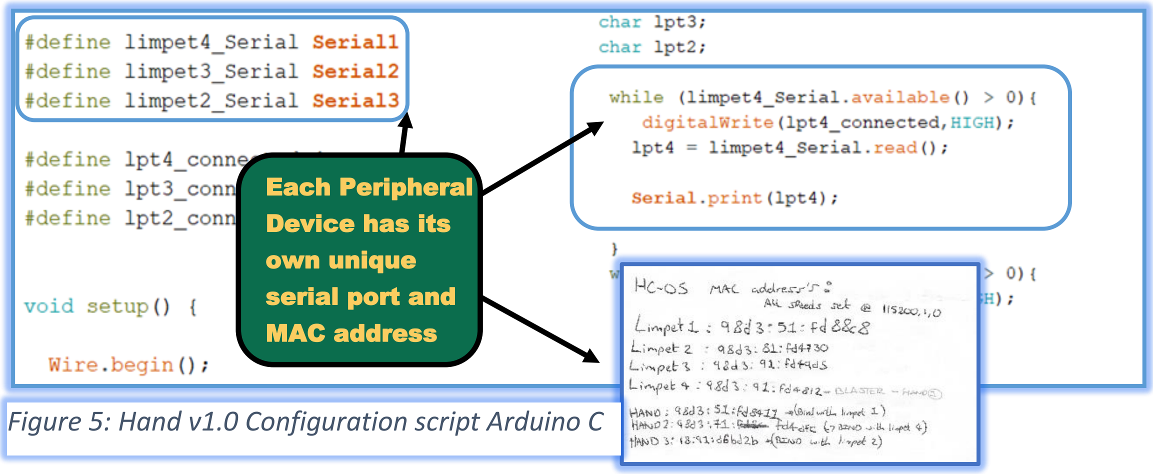


Figure 5: Hand v1.0 Configuration script Arduino C

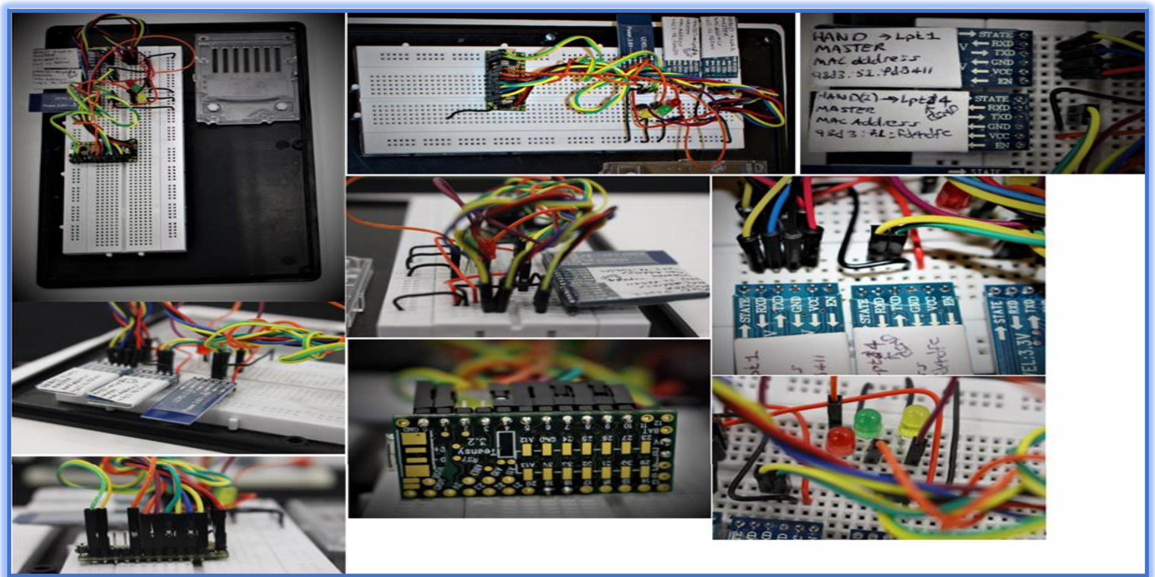


Figure 6: Primary Circuit Design Testing configuration

Once the Bluetooth modules were correctly configured, the prototype circuit was assembled on a breadboard, as shown in (Fig.6). The Teensy controller was then loaded with a new script to configure forwarding of the received data. An example of the forwarding function is shown in (Fig.7).

```
while (limpet4_Serial.available() > 0){
   digitalWrite(lpt4_connected,HIGH);
  lpt4 = limpet4_Serial.read();

  Serial.print(lpt4);

}
```

Figure 7: Example f the system used to forward incoming data to the Brain

The breadboard configuration was then connected to unity3d via USB serial connection, and the control script created during the feasibility testing was applied again. To demonstrate control from each peripheral device and to check that signals where not getting crossed, i.e. interactions occurring on the wrong object.

(5.2.2) Evaluating Prototype Setup:

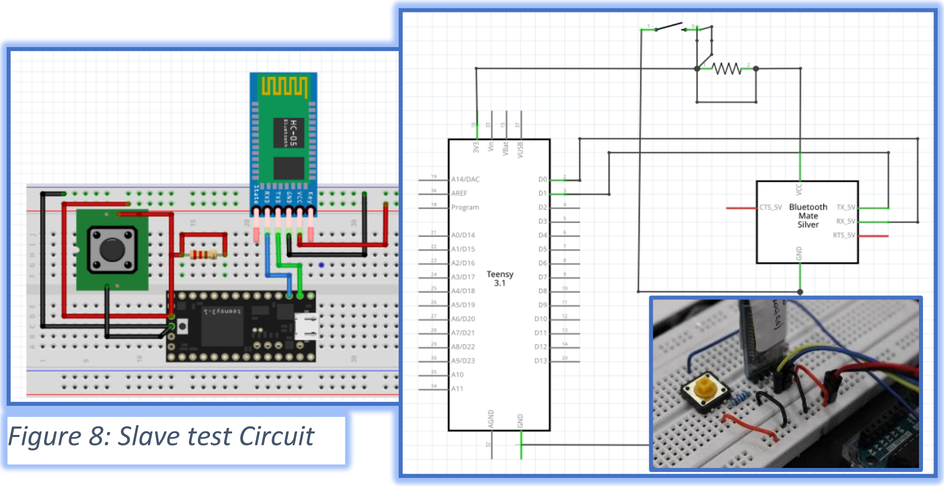


Figure 8: Slave test Circuit

To test the Hands capability to forward data effectively, three test peripherals were created (Fig.8). These used the Slave HC-05's paired to the Master modules on the Hand Control Device to transmit a single Char value that alternated depending on the state of the onboard button. With Unity configured to read incoming data from the serial buffer and then depending on the Character received triggered an event in the Unity environment, for this test this was to toggle the Boolean state of the SetAcitve() function for a game object, and register a Log in the console declaring which of the Slave devices had just been interacted with. (Fig.8)

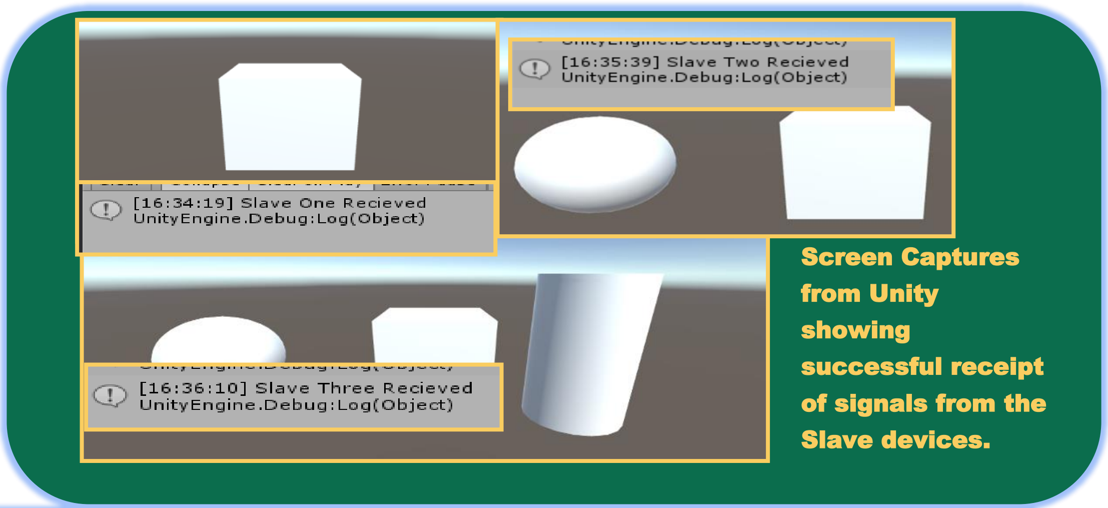


Figure 8: Unity – Hand Test 1 Screen Captures showing successful receipt of incoming signals from slave devices

The prototype system of the Hand circuitry performed well in the initial phase of testing, and the desired outcome in Unity was proven to be consistently repeatable, therefore, to move forward with the design process a more permanent iteration of the prototype was created.

(5.2.3) First Prototype: Construction of the Hand:

As the circuit for the system had already been developed (see 6.3 Fig.4), this design was transferred to a sheet of Perfboard to allow the soldering of components into position. (Fig.9)

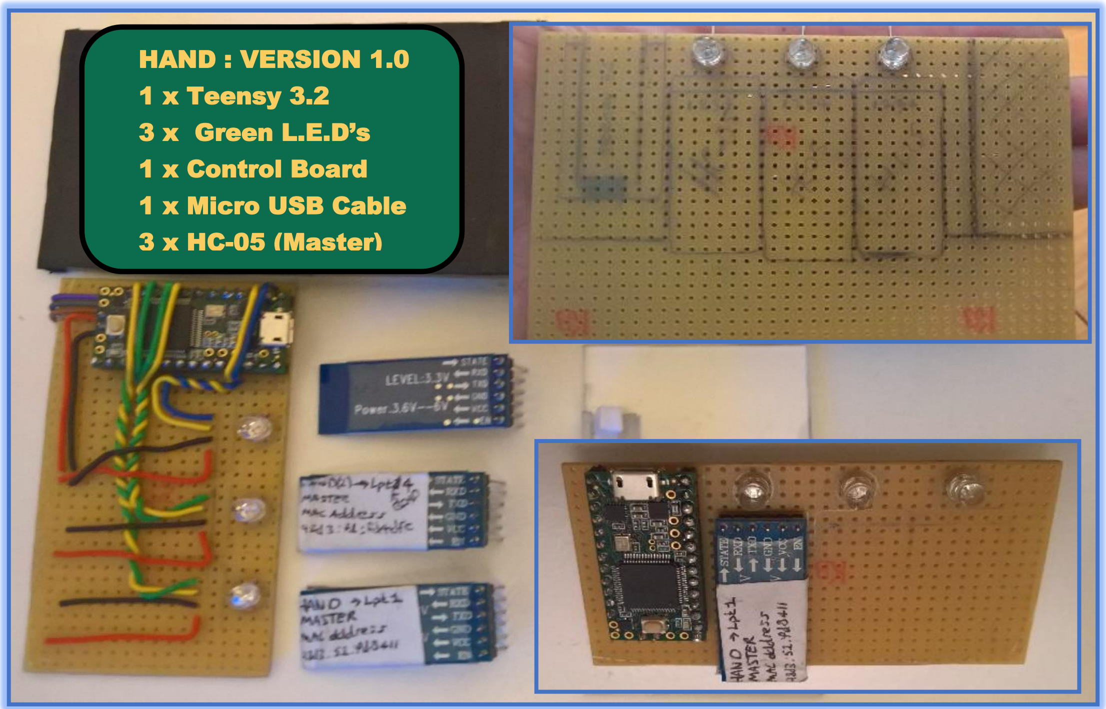


Figure 9: Circuit Board layout and Components used to construct the first Hand

With the main motherboard constructed the initial test were repeated to confirm that the build was stable, and the circuit was thoroughly tested to ensure that there were no dry joints in the soldering or short circuits.

As both tests proved successful, a connection with the peripherals was established, and data was correctly received in Unity, the completed motherboard was encased in a basic housing for protection. (Fig.10)

(5.3) Conclusions from Phase One:

The objectives of phase one were to show how the solutions to the design brief created in the technical research chapter were integrated into a functional prototype. Then, following the assembly of the Hand v1.0, to evaluate its capabilities to ensure the device was suitable for further use. Having confirmed the performance of the system and determined that the Hand would operate as an adequate control and data management device, it was determined that this new element of the Core Nervous System had been successfully integrated into its' architecture. Fig.10 shows the completed prototype build.

Figure 10: Final build of first Hand prototype Control Device

End of Chapter, space left intently blank

Prototype Phase Two (6)

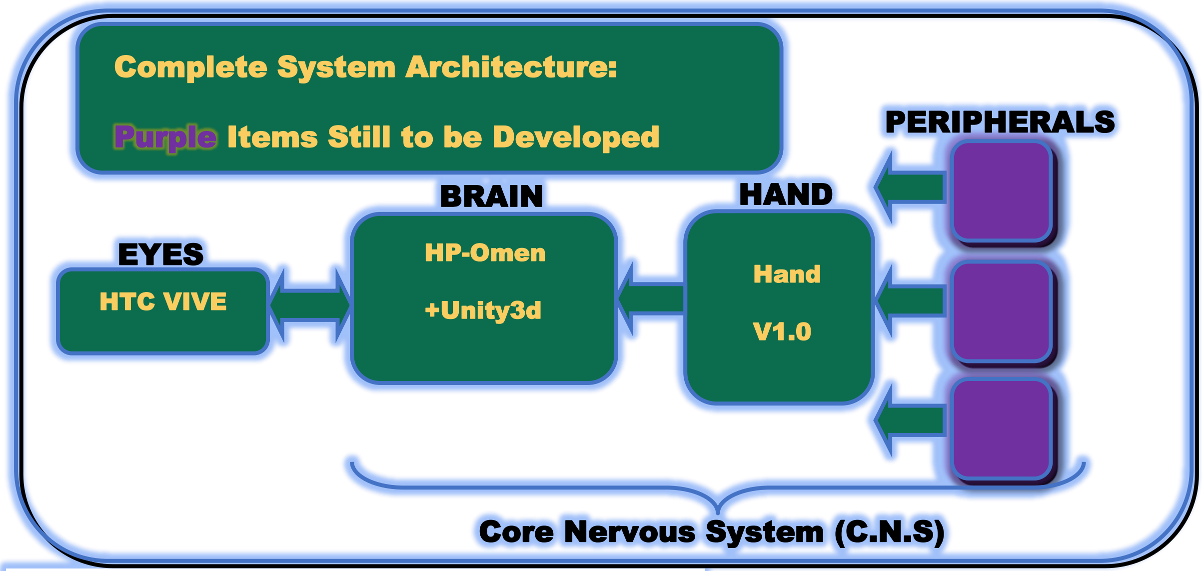


Figure 1: State of Full System Architecture as of the End of Phase Two.

(6.1) Introduction:

At this stage in the development of the Haptic Prototyping Toolkit, solutions to several of the design problems uncovered so far have been found. Communication between host P.C Brain and peripheral devices has been established along with the means to manage the flow of information from peripherals using the Hand. Yet to be addressed is the crucial component of how to translate real-world interactions into virtual reactions. This will be achieved through the development of peripheral devices (Fig.1).

The next step in the prototyping process will be to address the phase two brief.

- Design a system that uses the tactile switches and POTs to track and relay static interactions.

In this Chapter, the process of creating the tracking of Static based interactions will be covered. Static based interactions are assumed to be, interactions with an object with a fixed location in 3D space, but with a pivot point or switch/button, for example, a door that rotates on its hinge or a light switch.

(6.2) Developing Basic Static Interactions.


Figure 2: Tacktile Switch


Figure 3: 10kΩ Rotary

To create the system for tracking interactions of a static nature, modular elements for the microcontrollers will be developed. As established in the feasibility testing the components used will be a tactile switch for triggering events in the virtual environment, for example, pulling a trigger, switching a light or detecting the placing of an object (Fig2). A rotary potentiometer will be utilised to trace rotation, of dials, levers, or a combination on a safe, for example. (FIG.3)

(6.2.1) Triggers

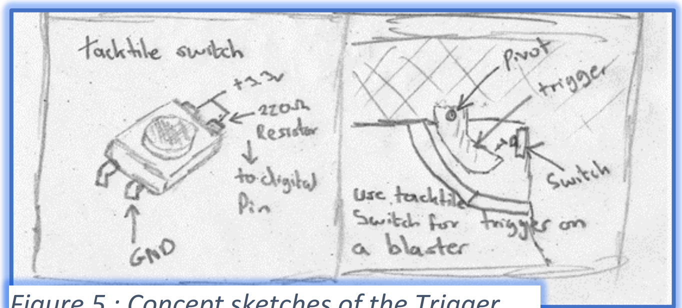


Figure 5 : Concept sketches of the Trigger

To create the trigger system the circuit configuration for the button from experiments carried out in (4.3.2) was converted into a finished prototype to create a single trigger and adapted to produce a double trigger (Fig.5).

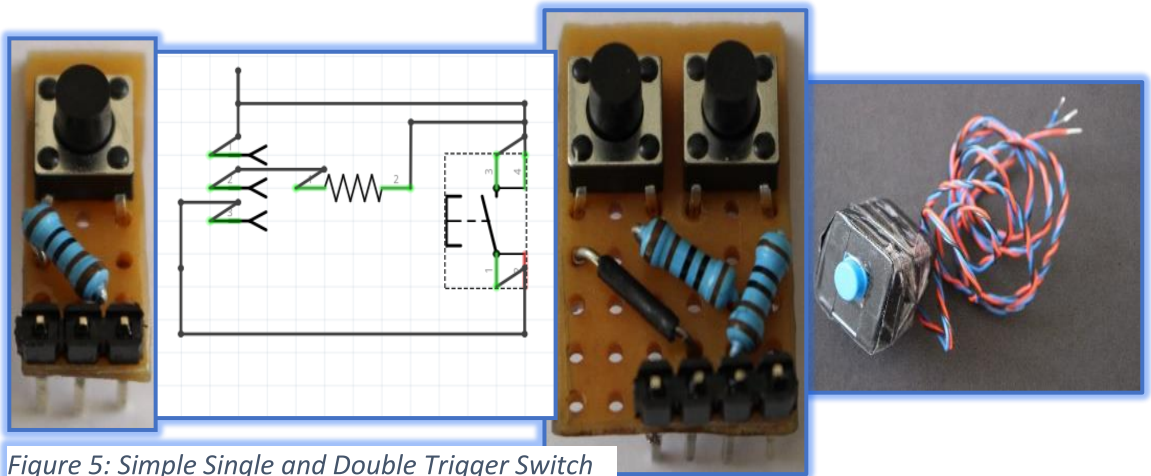


Figure 5: Simple Single and Double Trigger Switch

These Triggers were then tested for faults by repeating the experiment carried out in (4.3.2). Once stable functionality had been confirmed, duplicates were made in preparation for their deployment in conjunction with other modules to replicate real-world interaction. With a suitable solution created designed move onto looking at rotation tracking.

(6.2.2) Rotation Tracking:

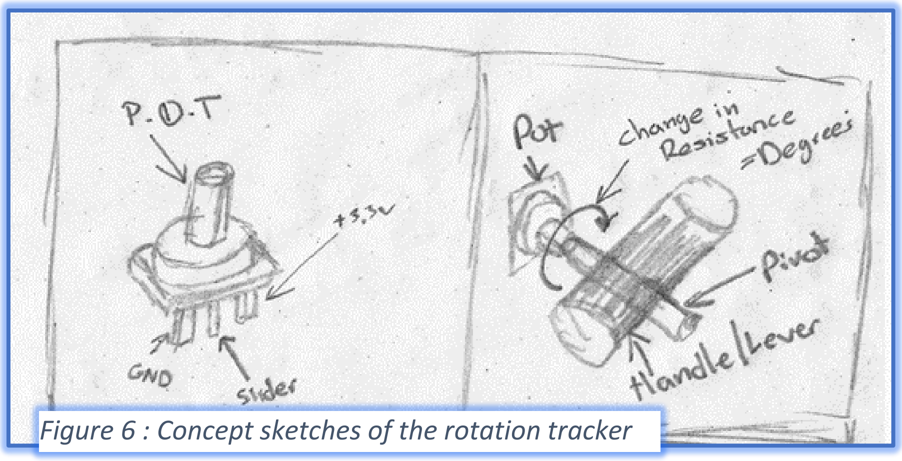


Figure 6 : Concept sketches of the rotation tracker

To create the rotation tracking, first, a test was conducted to determine the range of motion in the potentiometers, as they do not rotate a full 360 degrees.

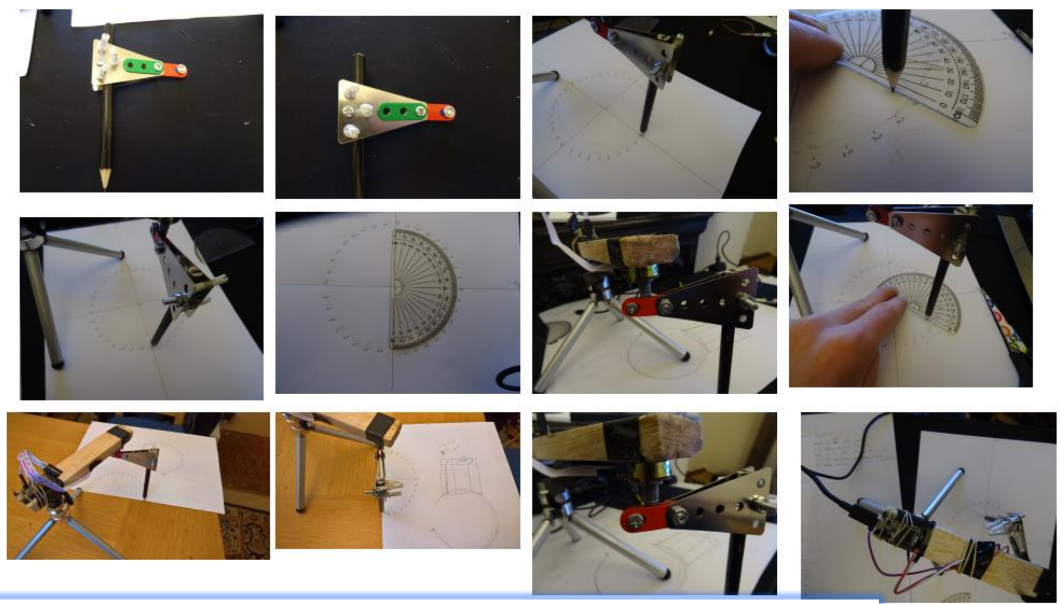


Figure 7: Pictures showing how the max angle of the POT was established

The POT was attached to a pivot, with a pencil attached, this was then used to draw an outline of the rotation, and a protractor was then used to calculate the min and max angles.

The tests carried out the established the range of the of POT to be from 0-295° degrees. However, to account for the dead zone identified in the feasibility test, this was reduced by 20° to 275°. While this does not offer a full 360° of rotation, it will be more than sufficient to track doors or levers, as their range of motion is usually less than 180°.

The data gained from the testing was used to create a script to convert the voltage value of the potentiometers into degrees of rotation (Fig.8)

```
void CalculateAngle(){
  max_ana = 275;
  min_ana = 0;
  max_vol = 1023;
  min_vol = 0;

dx = max_ana - min_ana;
dy = max_vol - min_vol;
m = dx / dy;
//val = analogRead(21);
//ana_raw =  1166-val;
ana_raw = analogRead(21);
ana_offset = ana_raw - min_ana;
vol_offset = ana_offset * m;
ang = vol_offset + min_vol;
```

Figure 8 : Voltage to rotation Conversion

With this script in place and the teensy set up to send data to the Hand via Bluetooth, the next step was to develop a prototype door tracker.

(6.3) Creation of Prototype Door Tracker:

To start the process of creating a peripheral device that could be attached to a door. To track its position in the real-world and transfer it to the digital environment the bracket used in the previous test was amended with a slider that fits over the top of the door, with the control unit mounted onto the frame (Fig.9)

Figure 9: Pic of the first Gen door tracker

The door tracker allows a real door to be used as a proxy for its virtual counterpart, as with substitutional reality, with the added advantage of being able to now interact with an object that moves.

To test the design and efficiency of the door tracking unit, a demo scene in Unity was created. In which a simple environment was built. With a door, that you can open, and thanks to setting the VIVE base stations up on either side of the door, you could also walk through and close it behind you. This system of tracking was enabled as the Vive base stations were connected directly using the supplied link cable. Screen captures from this demo scene can be seen in (Fig.10), and a video of capture taken from within the V.R environment and one shot in the real world is included for reference in the Digital Appendix Folder /Video Captures of Prototypes.

Figure 60: Images show the real-world on the left and what was seen in the Virtual world on the right

The outcome of this testing was positive. The door tracker worked well. Although after opening and closing the door several times, a discrepancy in the alignment became apparent. This was due to the POT having shifted slightly; the rotation was beginning in the dead zone; therefore, the first 10/20° of rotation of the door was not detected, this was noted for consideration in future iterations of the design.

(6.4) Conclusions from Phase Two:

The objective of this phase of the prototyping process was to address the implantation of tracking static interactions by developing on the knowledge gained in the previous feasibility testing.

Having completed this phase, two tools for the toolkit were now ready for deployment in the final evaluation environment. Details of which are contained in prototype phase Five.

End of Chapter, space left intently blank

Prototype Phase Three (7)

(7.1) Introduction:

Having so far completed the first two phases of prototyping, the Haptic Prototyping Toolkit was now on its way to being a suitable design tool.

The next step in the prototyping process will be to address the phase three brief.

- Create a method of tracking mobile objects rotation in 3D space using the IMU and an as yet defined system for monitoring their position in 3D space.

In this Chapter, the process of creating the tracking of mobile interactions will be covered. Mobile interaction is defined as interaction with objects that do not have a fixed location in the environment; this could be a, for example, a gun or weapon that would be used in a game.

Before prototyping could begin, the issue of how to track position in 3D space x,y,z had to be addressed as currently only a system for monitoring the rotation of an object has been tested.

The method of tracking position presents the biggest challenge in creating the HTP as this can not be achieved using basic components. As was discussed in the Literature Review, extensive research and development has been carried out in this area, and there as yet has not been a simple way designed that could be applied here. Options for tracking position considered were:

- Use of computer Vision and visible light-based tracking.
- Using a similar system to Augmented reality, where markers are used with computer vision to track an object.
- Use of IR-Camera and IR-based tracking, using a modified camera from a broken Wii remote
- Use of an external collision system using a technique to detect when an object is in hand by having the user's hand close a circuit and register a change in voltage. Each option was considered in turn, with the outcome of identifying which option to use being made based on how simply the system could be implemented.

The Visible light and other computer vision-based solutions would require expensive cameras, and a large amount of computer processing power, which could limit the accessibility of the HTP. Therefore they were deemed to be overly complex and expensive to implement into this project.

The IR-Light based system showed promise as it could be built using recycled faulty Nintendo Wii remotes, as they have a camera in with built-in IR-tracking that outputs the position of the four brightness IR dots in view. This approach was investigated and taken as far as a functioning prototype; details of this work are included in the Design Diary Appendix (P71-77). From investigatory work, it was found that while this system would allow for objects to be tracked. It was better suited to a desktop design environment and to implement it for this system was deemed to be beyond the scope of this research project but could be considered in future interactions of the HTP.

Therefore, it was decided to follow the path of creating an external Collison detection system.

(7.2) Creating External collision detection:

To create the external collision detection system, another pre-existing library for the Arduino IDE was utilised and adapted to suit this project's requirement. The library used is called FastTouch, and it allows you to turn any of a microcontroller GPI/O pins into capacitive touch sensors. The advantage of this is that by adding conductive pads to any object in the real world. A change in voltage will be detected each time it comes into contact with something that closes the circuit, i.e. a human hand, and in turn, this voltage shift can be used to send a signal to Unity that the object is currently in Hand or has been dropped.

(7.2.1) Creating the Prototype Limpet Controler:

To create the system, a broken PlayStation 2 peripheral gun was stripped out and modified with a teensy powered control board. The board developed also forms the prototype of the peripherals themselves, which as they are designed to latch onto real-world objects were dubbed "Limpets".

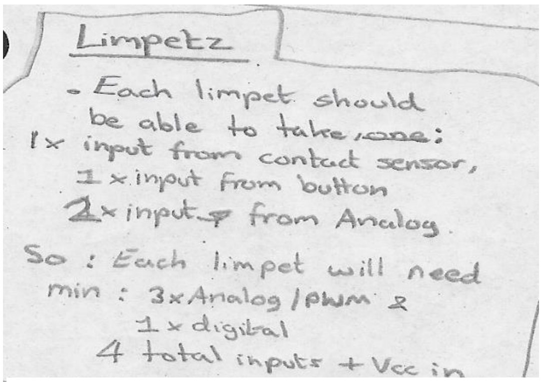


Figure 7

Fig.1 shows the 1st prototype design brief of the external tracking devices. The circuit board was et up with a Teensy 3.2, HC-05 Bluetooth, the new IMU BNO055 and LEDs, one in colour and one IR. These were included in the design to allow either the IR-based tracking system or a visible light system to be implemented in future iterations fo the toolkit. Fig.2 shows the circuit design for the Limpet module.

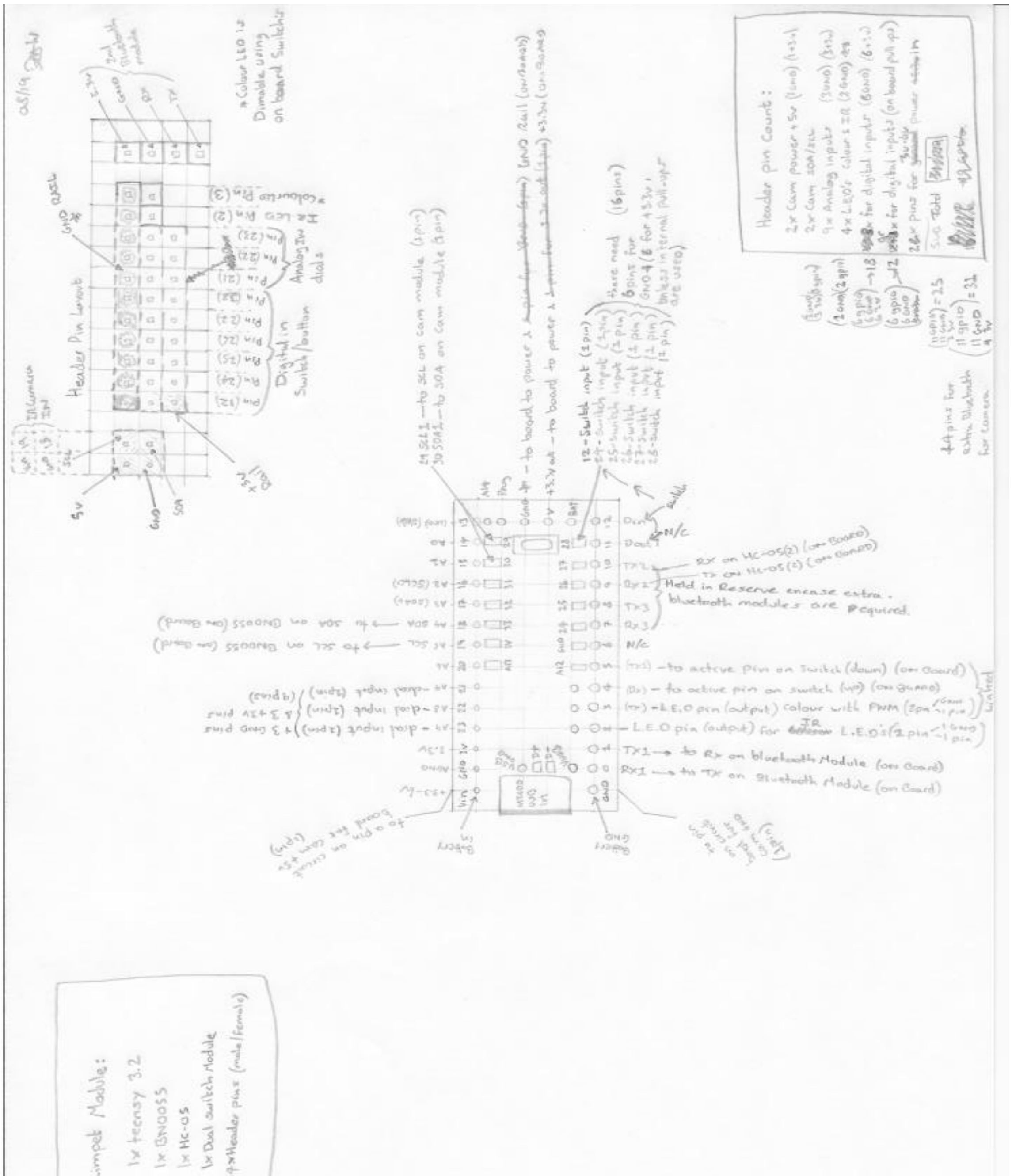


Figure 8 : Shows the prototype circuit diagram for the Limpet

As with the design of the Hand, the circuit was first set-up on a breadboard and tested using the feasibility tests developed in the technical research phase. (Fig.3)

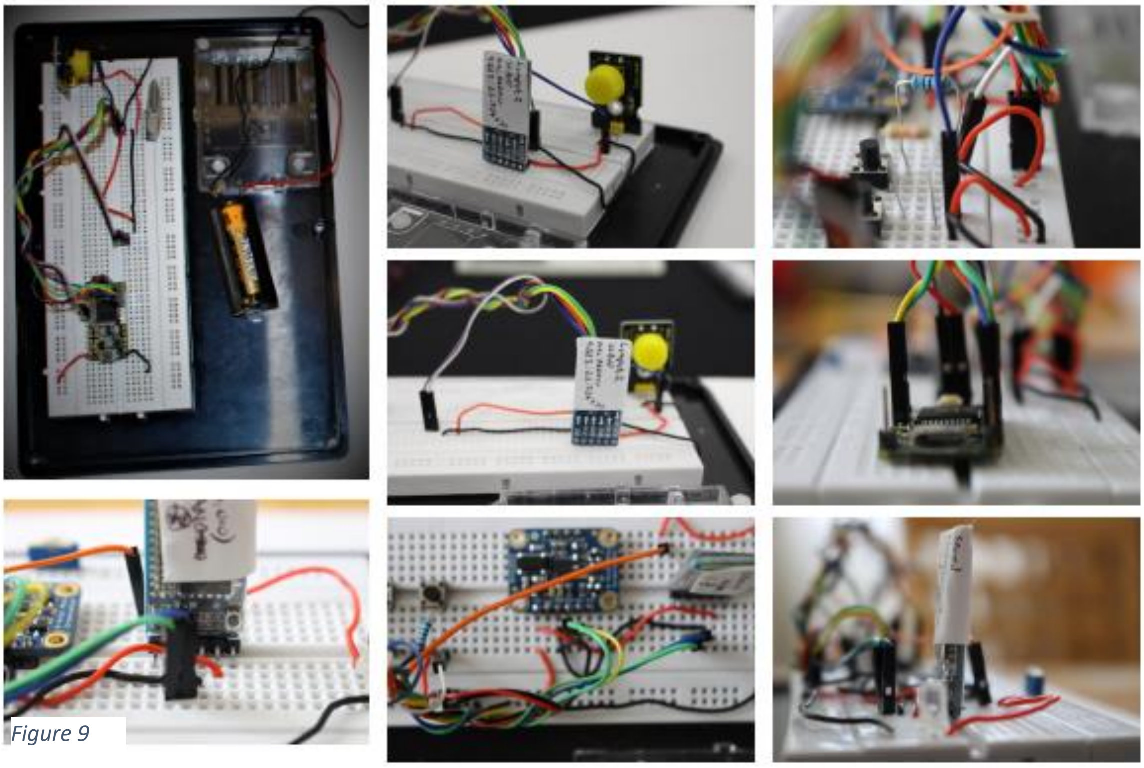


Figure 9

Once the design had been tested and deemed to be functional, the whole system was transferred to prototyping board, and the components were soldered into place. (Fig.4)

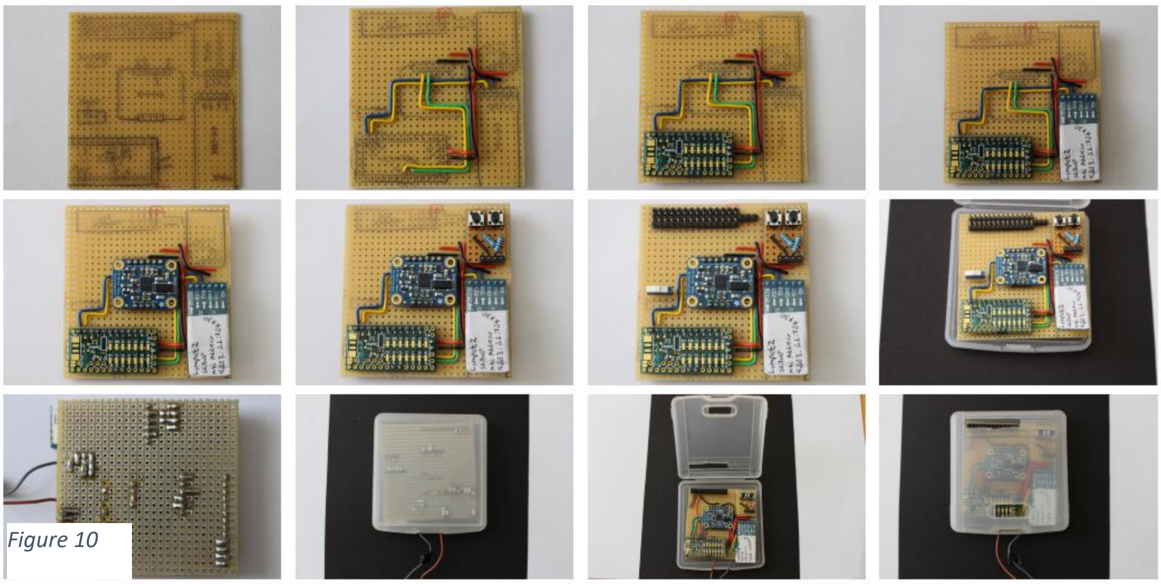


Figure 10

With the prototype Limpets controller constructed, and again tested for shorts and dry joints, the next step was to strip out and replace the electronics from the PlayStation Balster.

(7.2.2) Creating the First Game Controller:

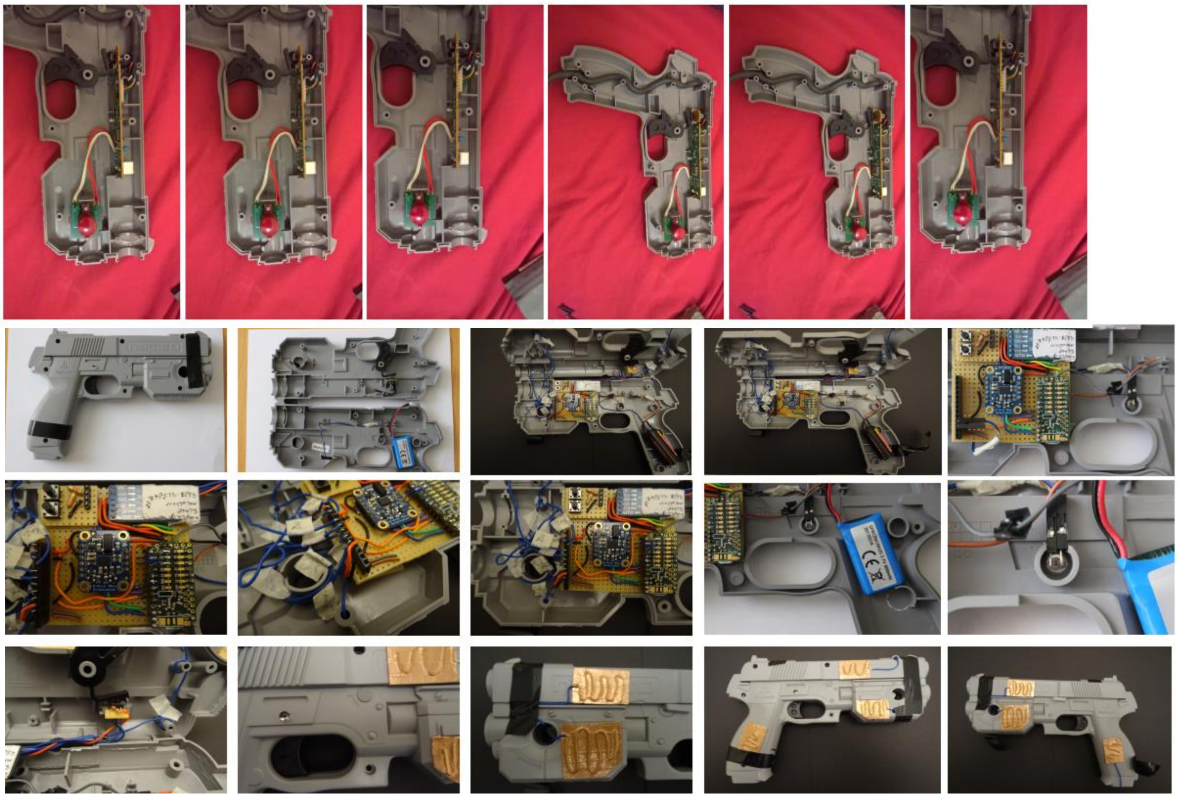


Figure 11

The images above (Fig.5) show the process of fitting out the recycled PS2 controller. The limpet controller was mounted inside the device with copper conduction pads added to the areas most likely to be held/Pick-up by the end-user.

The Control board was updated with version one of a master control script that combined all the functions created during the technical research phase. This meant that the Balster was outputting, the following Data to the Hand Device:

- Device Name: Blaster
- Rotation data: X/Y/Z
- Trigger Data: High/Low (1/0)
- Touch Data: High/LO. W (1/0)

As with previous iterations, the data was transmitted with a start and end marker, so that received information could be correctly assigned to the desired game object. However, instead of leading each value with a letter to identify it, this time, the individual variables between the start and end markers were separated by a comma. As the Hand would be compiling the data from three Limpets into One continues steam of output, the comma was added so that each part could be easily separated in Unity.

With all the relevant circuitry in place and code installed a 3D model of that would be the virtual proxy for the real Blaster was created with the Maya 2019 (Autodesk, 2013) modelling package. (Fig.6)

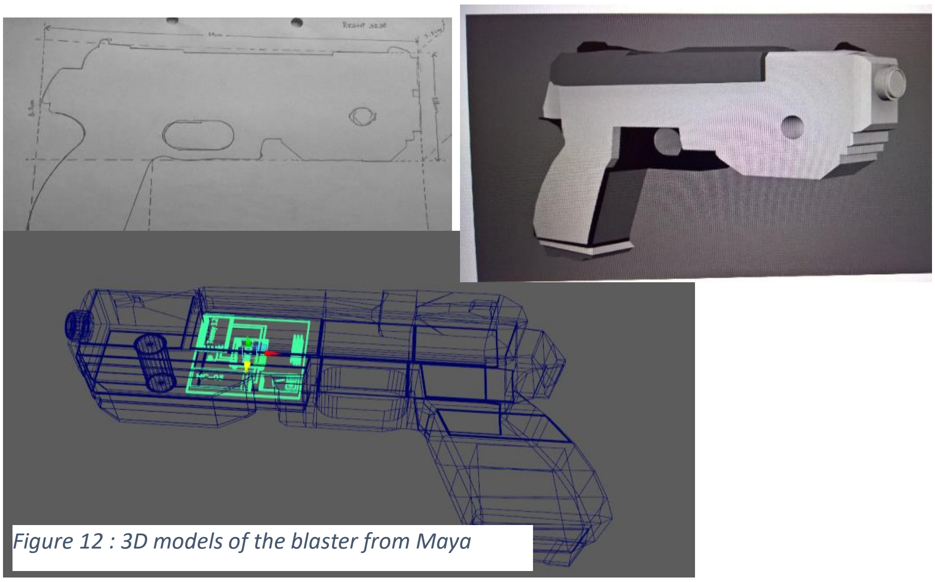


Figure 12 : 3D models of the blaster from Maya

To test the Blaster, the output stream from the Limpet was monitored, and the data being transmitted was then checked against what was received in Unity. The Blaster was them deemed ready for use in the final evaluation.

(7.3) Conclusions from Phase Three:

The objectives of this phase of prototyping were to develop a system for tracking mobile interactions. This was achieved by first designing and implementing a prototype control board (Limpet). Through the use of an Arduino Library, a system of capacitive touch was applied to act as an external collision system. The results were that the toolkit now had a prototype piece of hardware, that allowed Passive Haptic interaction.

A design consideration that still needed to be addressed is covered in the final prototyping sections. That is how the capacitive touch information was translated into a positioning system, in brief, this was achieved by parenting the Blaster to one of the virtual hand models created by the LEAP motion, and then the position was taken based on the Leap motions tracking.

Prototype Phase Four (8)

(8.1) Introduction:

In the final stage of the prototyping phase, the aim was to develop each of the prototypes created so far into a single complete and functioning first version of the Haptic Prototyping Toolkit. The objectives being to ensure that each of the prototype Limpet control boards was suitably housed so that they are ready to be used in a user evaluation. To achieve the objectives, the limpets were divided into two types of tracker, in line with the definitions created in chapter 4.2.3, they will be referred to as Mobile and Static.

- Mobile: Will be used for the mobile interaction detections and will have the BNO055 IMU, and capacitive touch system included.
- Static: Will be used for tracking Static Interactions, therefore, will only have digital and analogue input/output capabilities

This design choice was made as it reduces the cost of each unit, helping to ensure the solution adheres the overall objective of creating a low-cost solution. Also, the inclusion of the IMU in the static limpets would be a waste of computing power as they will not be used.

Once the competed system is in place, the design process will focus on creating the evaluation environment, details of this process are included in chapter 9.

(8.2) Designing and Building the Limpet prototypes:

To begin the creation of the final system design, first, prototype boards required producing, and plans for their housing were made.

(8.2.1) Mobile Limpets:

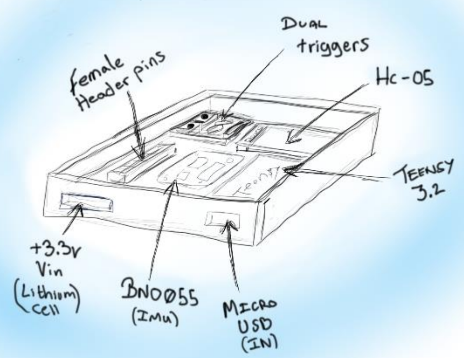


Figure 13: Mobile Limpet Module Case

Fig.1 shows the concept of art for the Mobile limpet modules; this design was then used to create the housing that would hold the active trackers, that were not being directly installed into hardware peripheral, such as the blaster.

Fig.2 shows the competed Limpet v1.0 Mobile tracker:

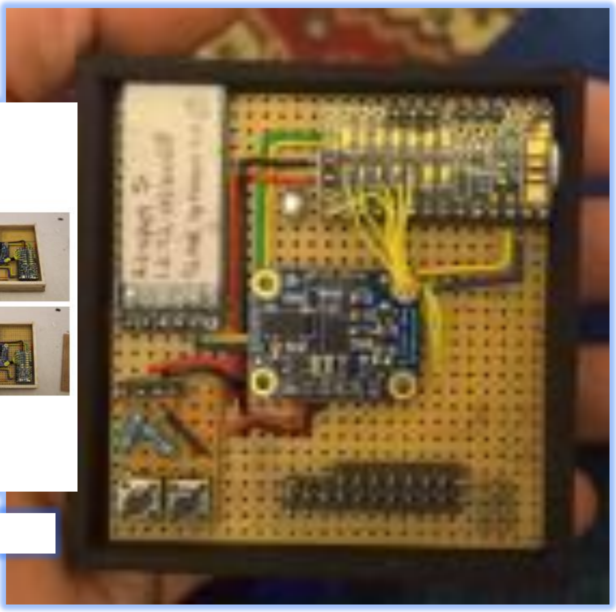
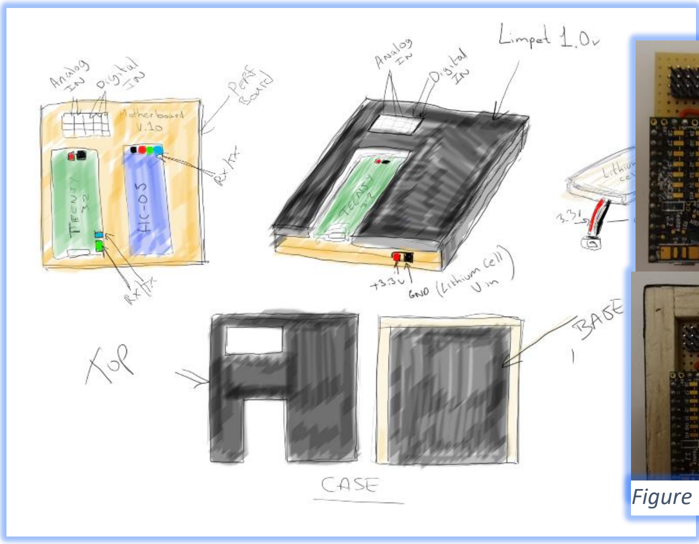


Figure 14: Production and Finished Mobile Limpet

The outcome produced a small, discrete tracking unit, which is capable of detecting six touch input, 3 Digital Inputs (as well as two onboard triggers, three analogue Inputs and control over either I.R. or visible light L.E.D's. L.E.D control is included for the future development of a more advanced position tracking system.

(8.2.2) Static Limpets:

To build the static limpets, the prototype board built in phase Two was first housed to give it protection and allow for ease of installation into a target environment. (Fig.3)



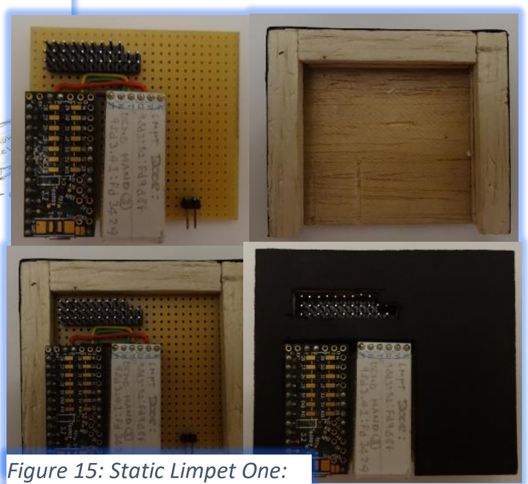


Figure 15: Static Limpet One:

Once this was complete, a second design was created to make another static tracker as the intention was to demonstrate a scene which uses three limpets, one mobile and two static. The number of peripheral is currently limited to three as this is how many serial ports were available on the teensy 3.2.

For the 2nd static tracker, the design was improved from the first iteration, an improved housing was developed, and analogue and digital inputs were separated to enable more straightforward attachment of the sensor devices. (Fig.4)

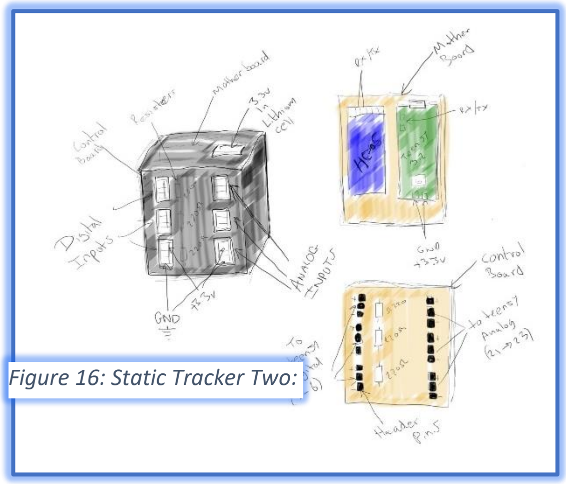
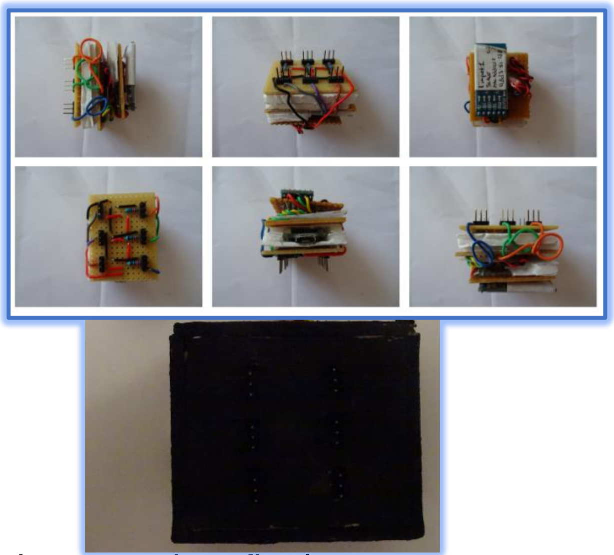


Figure 16: Static Tracker Two:



With the Three peripheral devices now in a final prototype stage, the next step was to test the full system in the demo scene created in chapter 7, with the addition of the blaster and a 2nd static tracker that acted as a trigger in the space.

(8.3) Full system Testing:

Now that a complete functioning prototype Haptic Toolkit had been developed the next step was to test if the entire system provided a consistent and low latency experience.

Due to the limits in computational power available on the Teensy 3.2 in the Hand device, all received data was being compiled into one continues steam of information, with the pre-configured scripts from the feasibility testing applied to split the data in Unity. While this approach had proved successful in the previous tests when the full system was put into place, the result was not satisfactory. It produced very high levels of latency, which in turn made the virtual world unpleasant to be in.

This latency was being caused as the system was hanging waiting for a correct read of the incoming data, as the system was still suffering from the initially observed system miss reads, where data could not be converted into the required float variables.

The first step to address this was to create a new function that used a Regex (Regular Expression) pattern to ensure that the system only attempts to process Strings received in the correct format and dump those that were wrong. (Fig.5)

```
private static readonly Regex rgx = new Regex(@"^[A-Z]\-?[0-9]?[0-9]?[0-9]\.?\d*$");
```

```
public static bool VerifyIncomingString(string incoming)
{
    return rgx.IsMatch(incoming);
}
```

Figure 17:Regular Expersion used to check the incoming data was in the correct format:

This solution reduced the number of errors in a 10 second period from the average of 4700 miss reads, to 520, but there was still significant lag in the system, so a second solution was required. The approach taken was to utilised Unity's' ability to operate muli-threading. Multi-Threading allows scripts to run on separate loops so that if one hangs, it does not cause the whole system to freeze, so a separate thread was created for the Hand and each of the Limpets incoming data.

To implement this new code was written the created a sperate thread to handle the incoming data from the Hand, with the rest of the program running on the main thread. A system of synchronised ques

```
public void Lmpt1Loop()
{
    while (IsLooping1())
    {
        ConvertLmpt2ToFl
        Debug.Log("2nd T
    }
} //code to use incoming
```

```
public void StartThread()
{
    BuildQueues();

    hand = new Thread(HandLoop);
    lmpt1 = new Thread(Lmpt1Loop);
    hand.Start();
    lmpt1.Start();
    Debug.Log("started threads");
}//creates and starts the thread;
```

```
public void BuildQueues()
{
    outputQueue = Queue.Synchronized(new Queue());
    inputQueue = Queue.Synchronized(new Queue());
    inputQueueX = Queue.Synchronized(new Queue());
    inputQueueY = Queue.Synchronized(new Queue());
    inputQueueZ = Queue.Synchronized(new Queue());
    inputQueueF = Queue.Synchronized(new Queue());
    inputQueueA = Queue.Synchronized(new Queue());
    inputQueueB = Queue.Synchronized(new Queue());
    inputQueueC = Queue.Synchronized(new Queue());
```

Figure 18: code to control muli-threading in Unity

was then also implemented to hold each variable as it was received until it was needed in the scene. Fig.6 shows examples of the new functions created to handle multi-threading. With the new system in place the demo scene was rerun, this time the latency had been removed and the scene ran smoothly, however, implementing four separate threads proved to be very computationally expensive and after around 5mins of testing the system overloaded and crashed the computer.

At this point, there seemed to be only two options to improve the efficacy of the system design, one would be to upgrade to a more powerful P.C, but this goes against the concept of designing for older models of computers to help keep development costs low. Two was to reconsider the microcontroller used in the Hand Device to improve the quality and consistency of the flow of data.

It was the second option that would provide the solution. As further online research into microcontrollers led to the discovery of anew model of the Teensy which had just been released by PJRC electronics. The Teensy 4.0, which offered computational power that was provided by an A.R.M. Cortex-M7 processor at 600 MHz, making not only the most powerful microcontroller currently available but offers processing speeds approximately 15 times faster than the currently used version 3.2. Other advantages are that it comes in the same form as the 3.2, so would fit into the current designs, but even more importantly it offered seven serial ports, which potentially meant that the H.T.P could be upgraded to control seven limpets. To develop four more peripherals was considered beyond the scope of the current aims of the project. Still, to enhance the evaluation, a fourth limpet was designed to provide a 2nd mobile tracker.

To incorporate the new microcontroller into the system version one of the Hand was redesigned. This upgrade to the system while providing more power also helps to demonstrate that the H.P.T designed is adaptable and will be upgradable as newer microcontrollers become available.

(8.4) Development of the Hand 2.0:

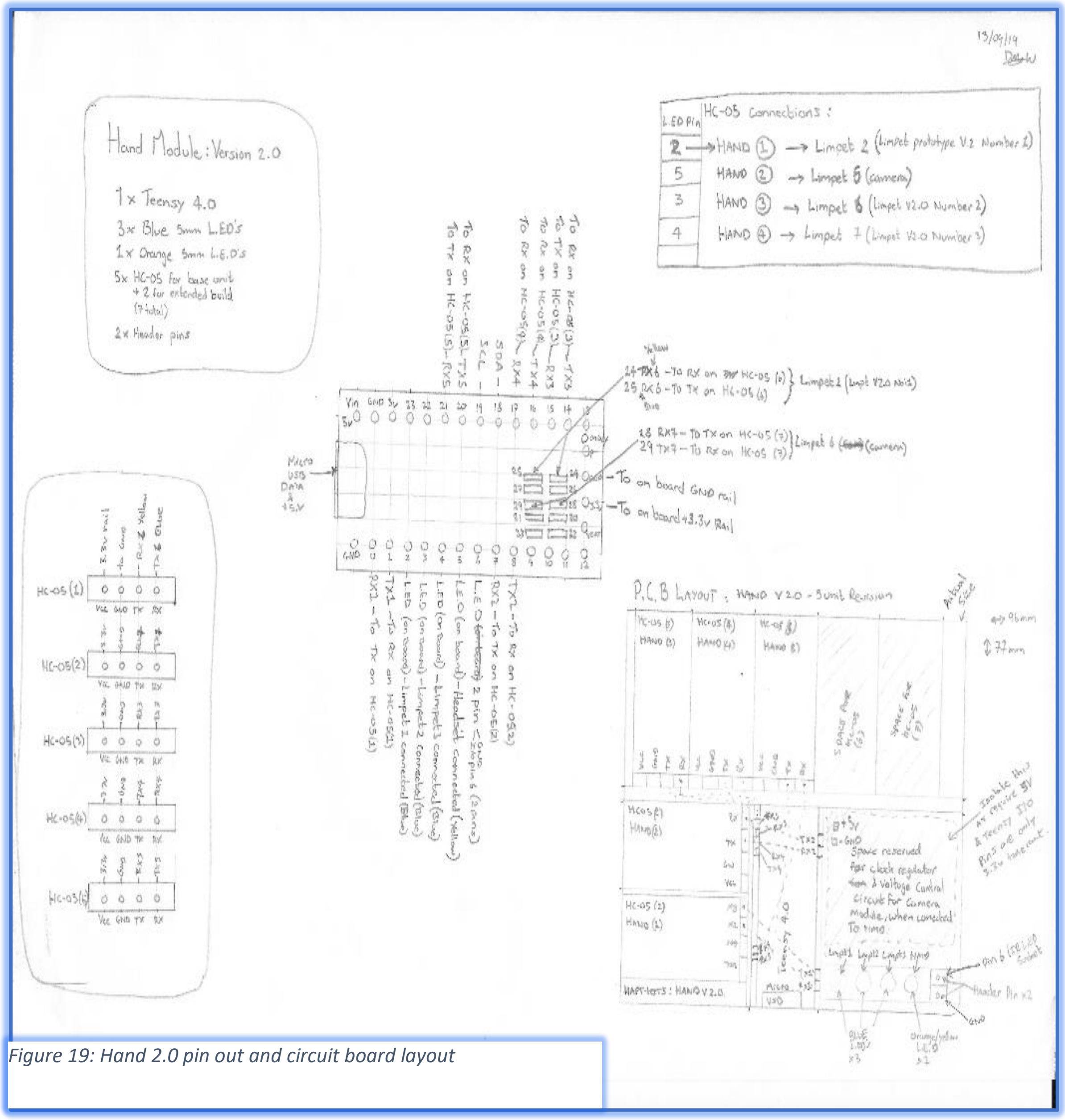


Figure 19: Hand 2.0 pin out and circuit board layout

Fig.7 shows the new layout and pin configuration designed for the 2nd iteration of the Hand Device. From this, a circuit board and casing were produced as shown in Fig.8.

Figure 20: Completed Hand 2.0

```
Serial.println("<");
Serial.print(receivedData4);
Serial.print(receivedData3);
Serial.print(receivedData1);
Serial.print(receivedData2);
Serial.println(">");
```

Figure 21: New Hand 2.0 sends all received data as one line

With a more powerful Hand Device, now handling the incoming signals from the Limpets, the method of transferring data was also upgraded to take advantage of the new processing power. To create a more stable output of data, the Hand device now complied all received data into a single line and sent the whole package in one go. (Fig.9)

```
Serial.println("<");
Serial.print(receivedData4);
Serial.print(receivedData3);
Serial.print(receivedData1);
Serial.print(receivedData2);
Serial.println(">");
```

Figure 9: New format for outputting Data from the Hand

Now all the data was being sent in one packet the scripts to handle the incoming data were upgraded and simplified to reduce the stress put on the host p.c and prevent the system from crashing. A single extra thread was created to control the incoming data and avoid any latency on the main thread. Boolean functions were designed to allow the triggers and touch sensors to fire events in the environment. (Fig.10)

```
public bool TriggerPulled()
{
    if (BlasterTrigger == "0")
    {
        return false;
    }
    else
    {

        return true;
    }
}
```

Figure 220: Simpe bool function returns true when the trigger is pulled

With the Hand 2.0 operational, the final step in developing the full system was to create a way to allow the mobile objects to use the positioning data from the Leap Motion.

(8.5) Creating Position Tracking:

To enable objects to be moved around in the virtual environment required a new script to be created in Unity. This script provides the means to attach an object to a fixed point of the hand model generated by the Leap Motion, transferring the transform data of the position in 3D space from the Hand to the object allowing its location to be tracked.

(8.6) Conclusions from Phase Four:

The objective of this phase of design was to test the full system capabilities, to identify any problems with the H.P.T design before the final user evaluation took place. As shown, there were several issues with the first iteration of the system designs, which have now been addressed and accounted for. Having established that a full system was now functional, the design of the Evaluation Environment could begin.

End of Chapter, space left intently blank

Prototype Phase Five (9)

(9.1) Introduction:

The final stage of the design process was to develop evaluation environments to conduct user research into the feasibility of the proposed approach to Passive Haptic Interaction.

The object was to look for indications that this approach to Passive Haptics interaction has a measurable effect on the perceived sense of Presence of users in the V.E. To achieve this Presence would be measured using the IGroup I.P.Q. (IGroup Presence Questionaire)(*igroup.org*, 2001). As previously mentioned in section 2.1, the IGroup questionnaire is designed to be used in comparative studies between groups. The results provide a presence profile of the application or technology being compared. For this evaluation process, the comparison study will be conducted between interaction using the H.T.C. Vive controllers (Valve, 2017) and the same interactions completed using the Haptic Prototyping Toolkit. The predicted outcome is that the perceived sense of Presence will be in higher when experiencing the Passive Haptic approach, which would be in line with previous findings on engaging the natural touch sense (Insko, 2001).

To begin creating the evaluation environment, certain design choices needed to be made. The first being what approach to take with regards to types of interaction to include in the environment design. From the analysis carried out in the literature review of the previous research into Immersion and Presence, there is evidence that video games are an engaging activity and can be used to facilitate this type of experiment. Some examples of the kinds of games used previously are Formula One, Half-Life and Unreal Tournament (Cheng and Cairns, 2005; Jennett *et al.*, 2008; Bracken and Skalski, 2009; Weibel and Wissmath, 2011)

Building on this idea of using video games, the decision was taken to create a virtual environment that had elements of puzzle-solving and some interactive gameplay. The types of interaction represented will be inline with the findings of the initial design work presented in chapter 3; therefore, they should include the following:

- Demonstration of rotation around a fixed point or points.
- Include an object which can be picked up and carried.

- Include an object that has a secondary feature of input as well as being moveable.

Refining this idea further by applying the principles of Substitutional Reality and taking inspiration from Ivan Sutherlands 'Ultimate Display' concept (Sutherland, 1968) and the Holodeck from Star Trek (Steinicke *et al.*, 2008) a design idea was developed.

The evaluation would be conducted in a virtual environment, that consisted of a simple scenario that includes four tasks:

- Step One: Users will be asked to pull a lever in the room, which will transform it from an office to 'another world'.
- Step Two: Users will need to pick-up and carry a 'key' and place it on a target across the room.
- Step Three: Users will be required to open a 'safe' and remove a 'blaster' from inside.
- Step Four: Users will be asked to shoot a series of targets using the blaster.

As there are physical objects in the virtual space there is potential for a participant to trip or otherwise injure themselves, so, for this reason, the decision was made not to include audio in the experiment to enable constant communication with the participant. There is a secondary advantage to the choice not to add sound as it removes any influence that an audio track my have on the sense of Presence felt by the participant, which could influence the findings of the experiment.

(9.2) Design and Implementation of the Environment:

The design process began by selecting a target space to provide the primary physical proxy for the virtual environment. Empty office space was chosen to test using the H.P.T to conduct Immersion and Presence investigations in a real-world setting, demonstrating the potential of increased accessibility to research into the field by showing that data can be collected outside of a laboratory environment.

The creation of the virtual environment began by taking pictures of the space from multiple angles to aid in the 3D-modelling process (Fig1).

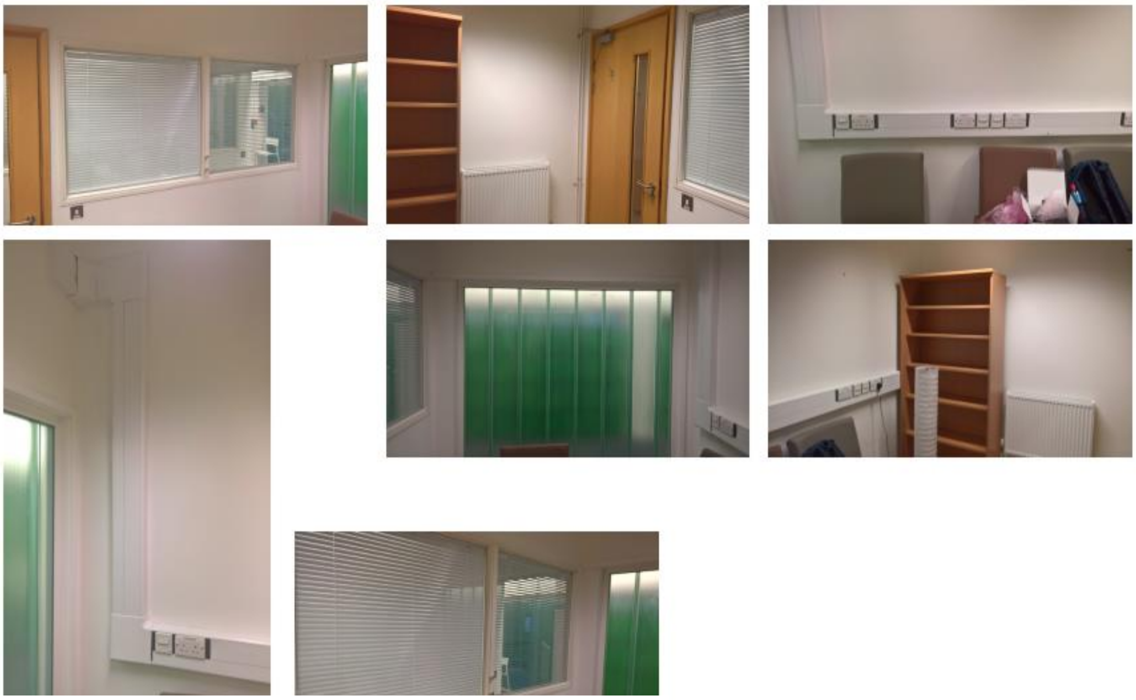


Figure 23; Office Spcce selected as the target environment

Measurements of the dimensions of each part of the space were recorded (Fig.2) to ensure a high level of accuracy when creating the virtual content to overlay.

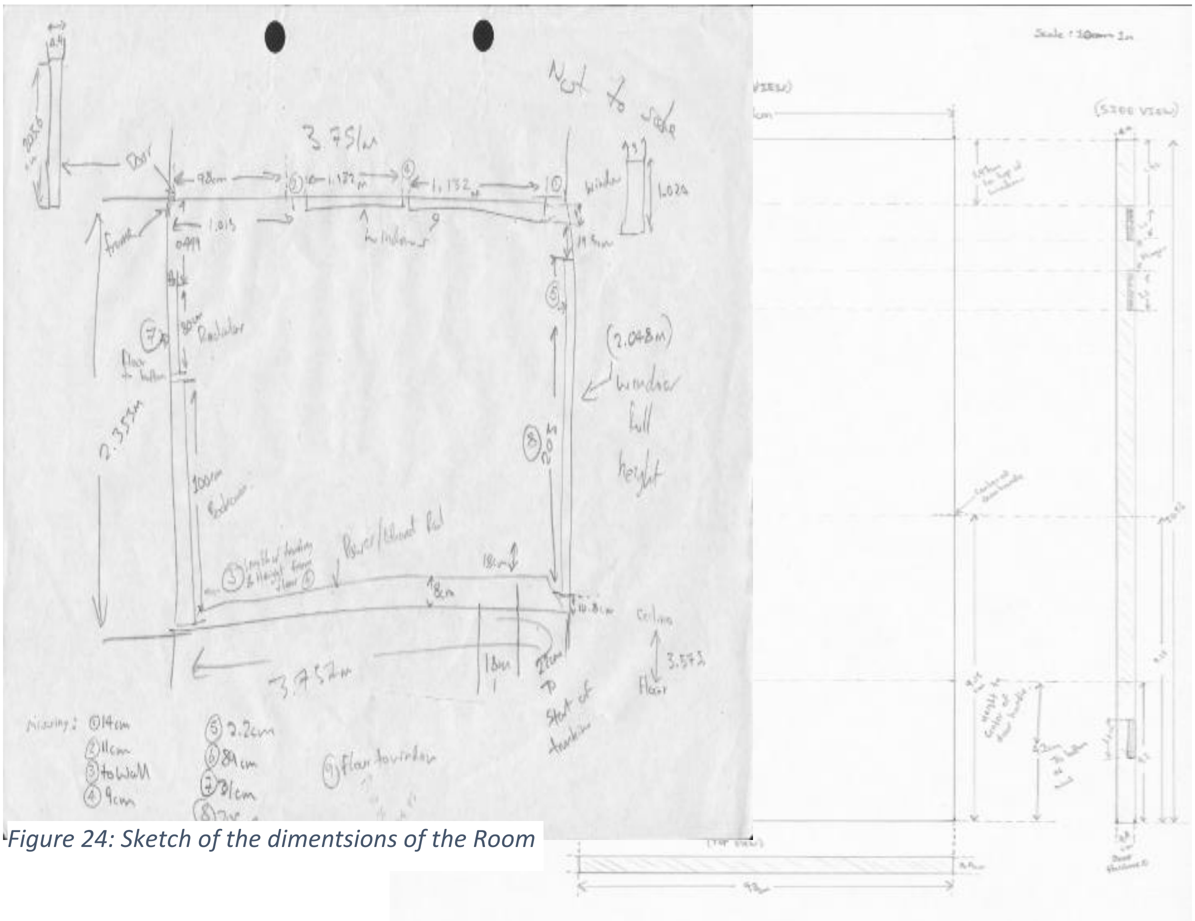


Figure 24: Sketch of the dimentsions of the Room

Using the data and pictures collected 3D models of the space were created using MAYA 2019 (Autodesk, 2013), Fig.3 shows the basic virtual overlay on the left and real-world on the right.

Figure 25 : Left – Virtual Overlay, Right – Real-World

The Models were then developed further, and colour textures were added to match as closely as possible to their real-world proxies.

See Fig.4.

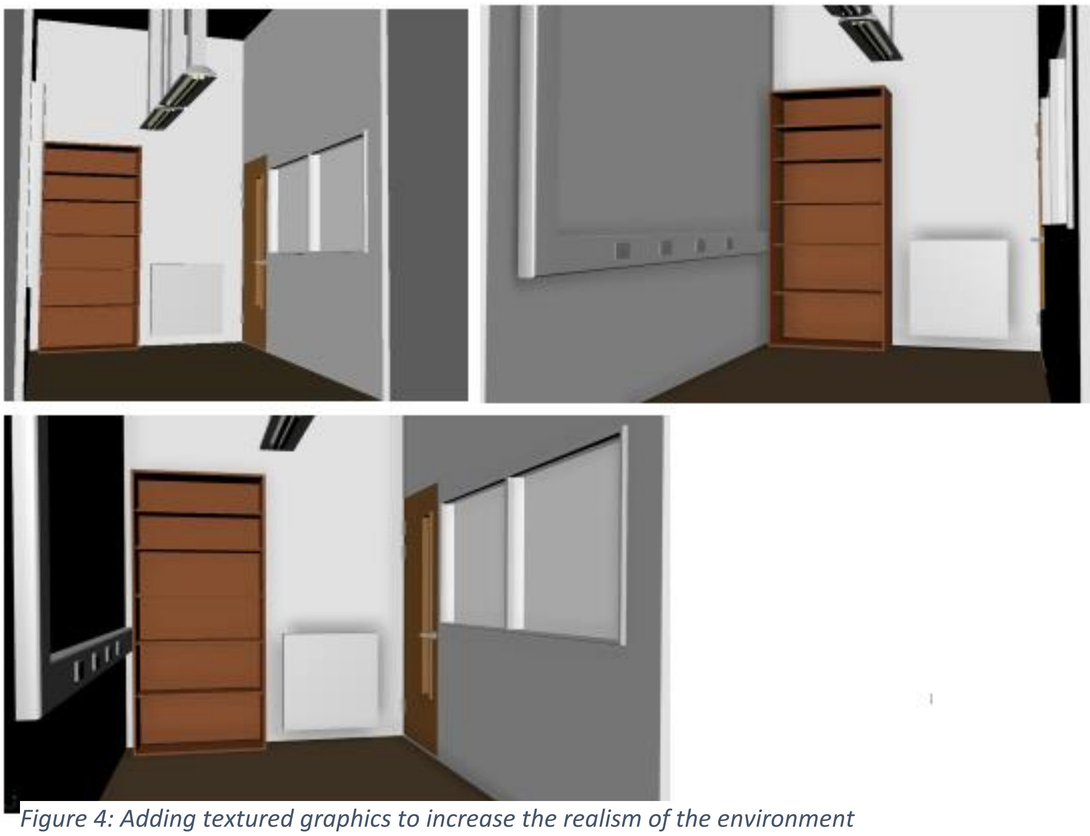


Figure 4: Adding textured graphics to increase the realism of the environment

Once the basic structure of the room was completed, development of the points of interaction began. Step one is a wall-mounted lever to facilitate the change in the environment. When pulled, the lever triggers an animation that caused the walls to drop away, and a new world to be revealed. (Fig.5)

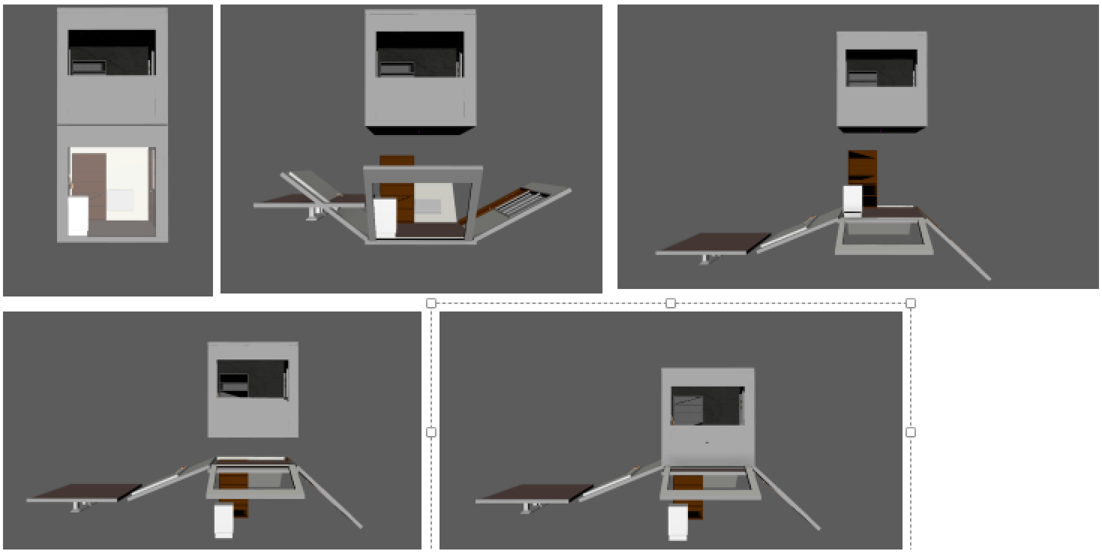


Figure 5: shows the animation that changes the environment : From left to right, the lever is pulled, the old walls fall away and the new environment drops into place.

Figure 6: Users View of environment before Lever is Pulled

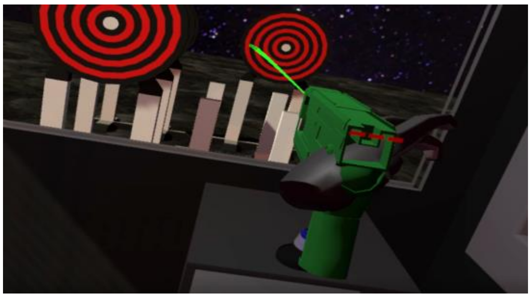
Figure 7: View of the new environment after the Lever

Fig 6 & 7 show the in users view in the scene before and after the lever is pulled. Once the user is presented with the new environment, there are three remaining steps to complete.

- First Pick-up a key and place it onto a “lock” on safe

- Second to open the safe and retrieve the blaster

- Thirdly to use the blaster to shoot a series of targets



Each step in the experiment is designed to test a feature of the H.P.T.

- Step one tests and demonstrates the ability to simulate a pull/push motion.
- Step two shows how the system can be used to create objects that can be picked up and carried.
- Step three shows rotation detection for hinges to make virtual doorways.
- Step four demonstrates interaction with objects that have an input feature as well as physical properties.

Once the experiment is complete, the results of the I.P.Q. will provide a Presence profile for the H.P.T, which will be used to assess the Systems Immersion. Also, by observing the participants while completing the tasks using the H.P.T modules, a visual assessment of how well they perform will be carried out, noting any unforeseen issues with the current prototype design. The findings will then be analysed, and the outcomes will be used to inform improvements to the system design as well as revealing any indications that this approach to Passive Haptics warrants further research.

With the digital content required for the Substitutional Reality created and a methodology for collecting data from the users completed, the next step in the design process was to develop the physical hardware that would provide the proxies for the points of interaction.

(9.3) Designing the Hardware Prototypes:

(Design Appendix P:98-108, contain further photographic evidence of the hardware design process)

To enable all of the required interactions in the evaluation scene, three new items of hardware needed to be developed, two using static versions of the Limpets and one more mobile type. The lever and the Hinge mechanic would use the static modules as they only need to output the rotation of one axis. The mobile Limpet would be used for the key as this requires the output of all three axes. For the blaster, the prototype developed in phase three would be used as it already had the required functionality.

To keep in line with the approach of using 2nd hand or upcycled materials to reduce the environmental impact of making new prototypes 2nd hand Meccano sets were selected to build the necessary parts. The advantage of using Meccano is that it can be adapted to suit a wide range of design solutions and can be acquired easily from 2nd hand shops.

The process of creating each interaction point followed similar steps to building the environment. A proxy object was selected for each, and measurements were taken to enable a 3D model to be created. For the lever, a section of broom handle was attached to a Meccano frame linked to a Limpet module, configured to detect the rotation of the lever around its pivot. A switch added at the base is depressed when the lever is pulled down, triggering animation in V.E. to start the scenario. Fig 8 shows the finished prototype lever.

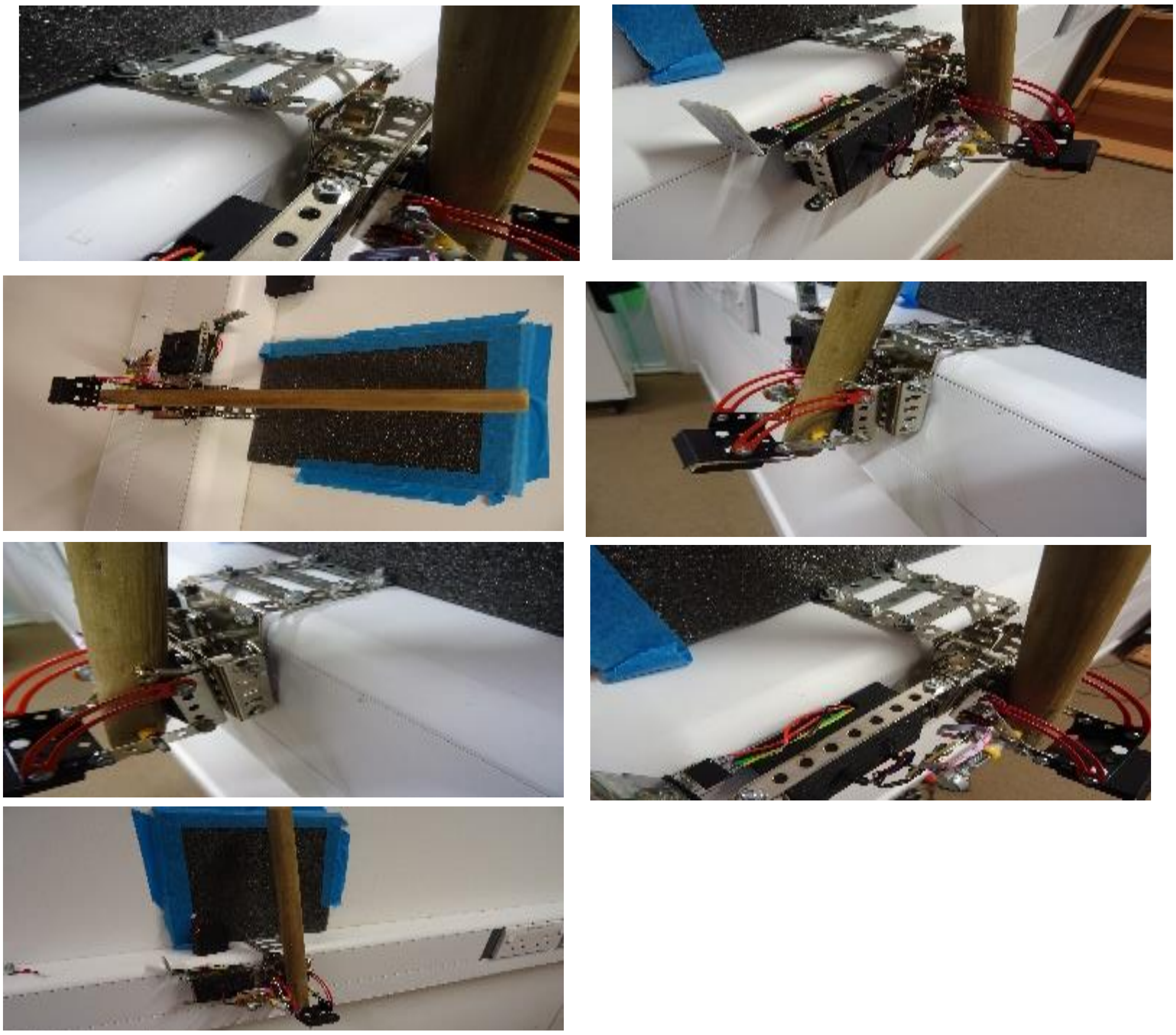


Figure 8: Interaction point One: Lever, constructed from Meccano and a Limpet Module : Shown it situ in the evaluation environement

For the key, an empty plastic container was fitted with an Mobile Limpet module, and the outside was wrapped in conductive copper

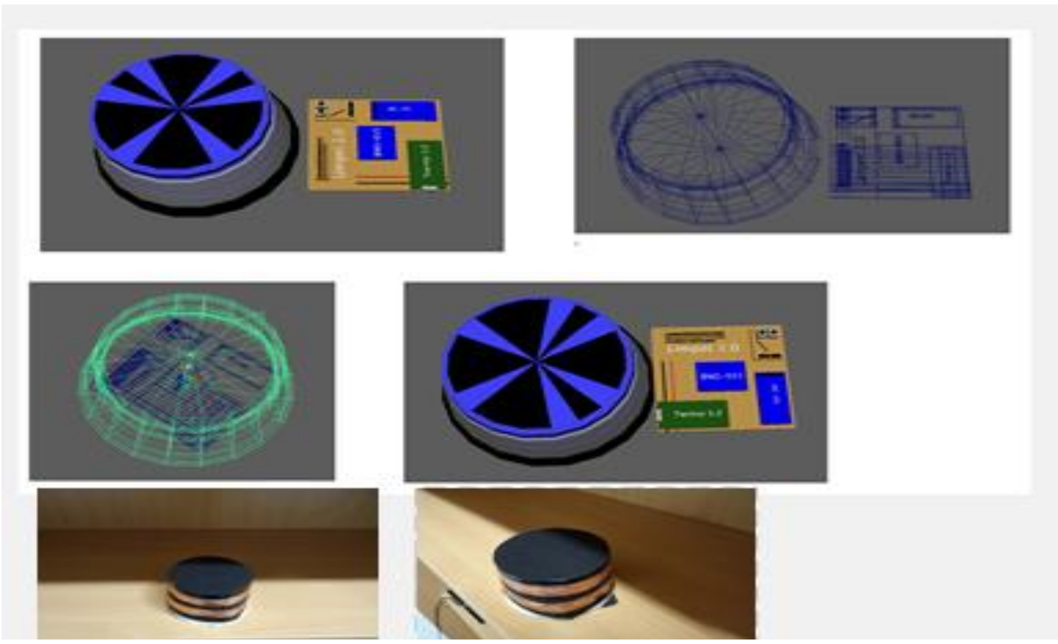


Figure 9: Mobile Tracker placed inside empty plastic container to create a movebale object as the 2nd interaction point

strips to enable touch detection. Once again, a scale model of the proxy object was created to act as the digital overlay. (Fig 9)

To build the 'safe' a small cupboard was chosen to be the proxy object, and Meccano was employed again to build the framework for the sensor. To track the rotation of the hinge, a modified version of the door tracker developed in Phase Two was created, adapted to fit the cupboard door. (Fig.10)

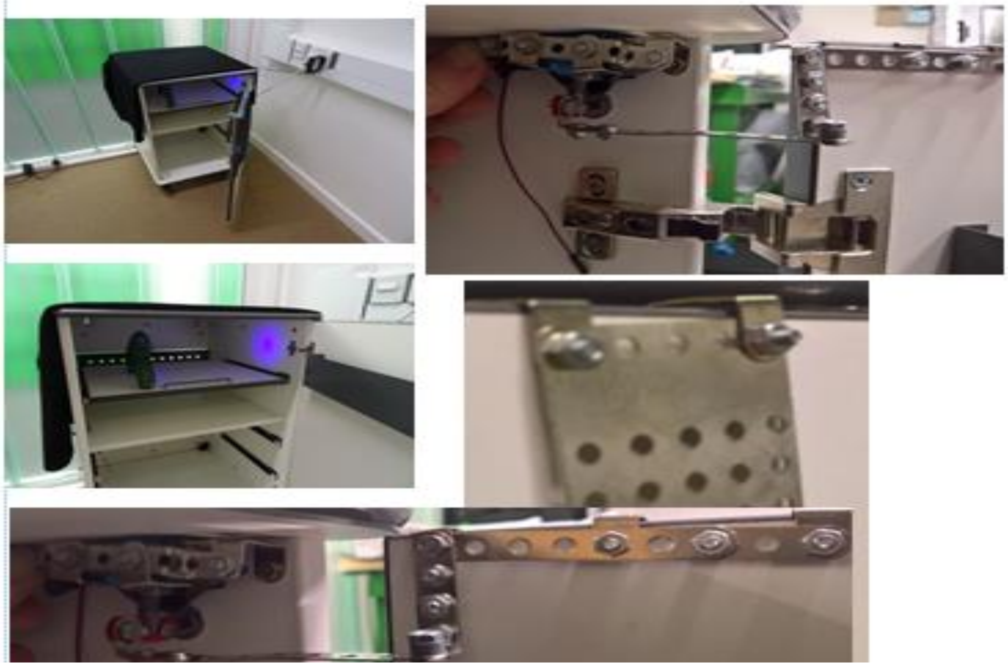


Figure 10: modded door tracker applied to cupboard to allow it to be tracked in VR

Once the design and implementation of the Haptic environment was completed, a second version was developed that uses the traditional H.T.V. Vive controllers for interaction. To achieve this, the same 3D scene was used, but the physical proxies were removed, and traditional collision detection methods were applied to detect user interaction.

(9.4) Conclusions from the Evaluation Design:

The objectives of this phase were to design and create two Virtual Environments to assess the feasibility of the proposed approach to Haptic Interaction. One built upon the principles on Substitutional Reality and enhanced to include passive haptic interactions enabled by the H.P.T. and the other built on the same environment but using traditional V.R. controllers for interaction, to enable a between-groups comparison study. The results and conclusions of this study are presented in the next chapter of this thesis.

Evaluation (10)

(10.1) Introduction:

Evaluation of the Haptic Prototype Toolkit (H.P.T) was designed to assess the technical capability of the system, validity of the system as a research tool and highlight indications that this approach to Haptics adds to the Immersive experience of a Virtual Environment (V.E), demonstrating the potential for further research and design work to further our understanding of Immersion, Presence and Interactivity.

As discussed in the previous chapter, the evaluation would be between groups, be conducted in the V.E. created in phase five, and involve the completion of 4 simple interactive tasks:

- Pull a lever.
- Pick-up and carry an object.
- Open a door.
- Pick-up an object and use it to interact with the environment.

These four tasks form the dependant variables of the study. The independent variable was the method of interaction, either the H.T.C. Vive controllers or Leap motion and the H.P.T. Participants were asked to complete the four tasks once with Vive controllers and then once with the Passive Haptic system. As an extra measure of comparison, the participants were split into two groups; the first group conducted the scenario with the Vive controllers first (Group1) and the second group used the H.P.T approach first (Group2). Participants were then required to complete the I.P.Q survey at the end of each session. They were asked to complete the I.P.Q while in isolation to minimise outside influences. It was also requested that they did not discuss their responses to the I.P.Q with any other participants. The findings from the I.P.Q were then used to calculate a presence profile for the two approaches to interaction. The predicted outcome being that the H.P.T would score a higher value and therefore indicate a positive result in support of this approach to Passive Haptic Interaction.

Presented in two sections below are the methodology and results of the user study and findings of the technical assessment. Conclusions are drawn from each separately, including a discussion of identified issues with the system and possible approaches to future works.

(10.2) User Study:

The user evaluation was conducted with twelve participants, selected from colleagues and faculty members of the University of York Department of Theatre, Film, Television and Interactive Media(TFTIM). The demographics of the group were: 60% Male, 30% Female, with 10% preferring not to answer. The age range was from 18 – 53; the mean age was 27.

Each participant was selected on the precondition that they had previous experience of Virtual Reality. This was done to alleviate the chance of people experiencing motion sickness, which can be a side effect of first experiencing V.R. They were also all given a participant information sheet which detailed: the motivation behind the investigation, clearly explained their participation was entirely optional, that they were free to exit the study at any time, that a video capture system would record their experience for technical referencing and outlined the details of how any personal data would be handled, ensuring it was clear this research was adhering to university guidelines.

The method of completion for the scenario followed these steps:

- Participants were introduced to the V.R equipment, and a brief explanation of how to safely remove the H.M.D was given.
- Participants were informed that a researcher would be present in the room with them at all times during the experiment. To observe the technical proficiency of the H.P.T and to assist in the event of an emergency.
- An explanation of the video capture system was given, outlining how the session would be recorded from their point of view, for further technical analysis and that they would not be identifiable from the footage.
- An explanation of required tasks to complete in the scenario was then provided; this was done before entering the V.E. to minimise the requirement for instruction during participation as outside distractions could affect the immersiveness of the experience.
- The process of how they would be interacting with the environment was then explained—detailing how to use the H.T.C. Vive controller and alternatively how the Leap Motion and H.P.T would allow them to use there hands as controllers.

- Once the scenario was completed the participants were given space in a separate room to complete the I.P.Q; this also acted as an opportunity to take a break from the V.R. equipment before they completed it for a second time.

The results of the user investigation were evaluated using the IGroup I.P.Q survey from 2001 (*igroup.org*, 2001). This version of the Presence questionnaire was selected over the earlier version created by Singer and Wilber (Singer and Witmer, 1998) as it has been developed explicitly for between-group studies of Virtual Reality systems. The I.P.Q contains a series of questions designed to measure Involvement (INV), Spacial Presence (S.P.) and Experienced Realism (REAL) to form a presence profile for each interaction method. Statistical analysis was carried out on the results of the I.P.Q., by conducting a one-way ANOVA on the means of the INV, S.P. and REAL values. This study assumes that: the null hypothesis is that there will be no implied improvement in perceived Presence when using the new HPT. The proposed outcome is that using the Passive Haptics approach offered by the HPT participants will perceive a greater sense of Presence. The calculations were conducted based on an alpha value of 0.05, which means that any calculated p-value less than or equal to 0.05 can be considered a statistical signification pointer towards rejecting the null hypothesis and in support of the proposed outcome.

The results can be seen as a measure of the System Immersion, which will be used to determine the effectiveness of the H.P.T approach to interaction and evaluate the feasibility of this approach to Passive Haptics.

(10.2.1) Results:

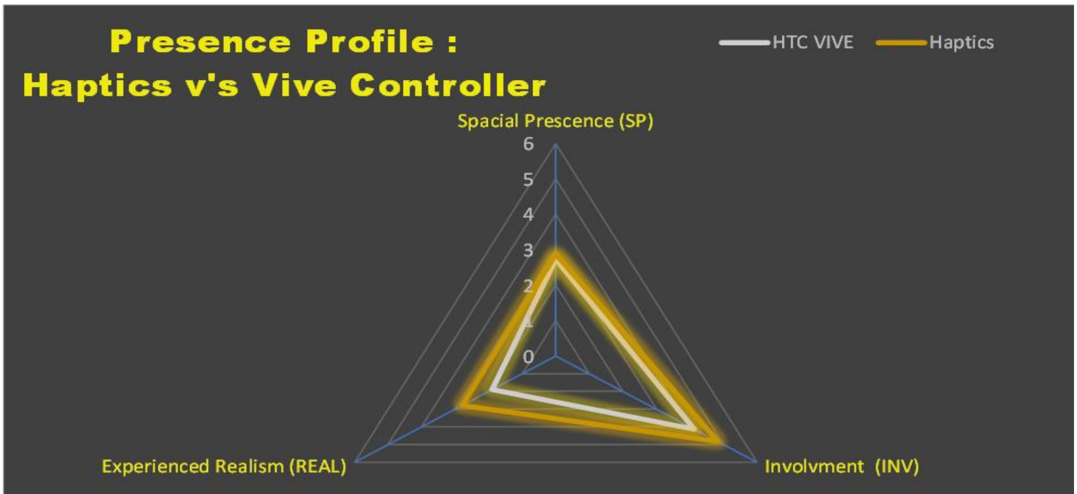


Figure 26: Presence Profiles for The HTC Vive Controller(Grey) and H.P.T interaction (Orange)

Fig.1 is a visual representation of the combined presence profiles for Group1 and Group2, for each method of interaction. It shows that overall the Passive Haptic Approach scored higher in all categories of the I.P.Q. The results from the one ANOVA confirm this and also show a statistically significant shift was seen in the Experienced Realism category (Fig.2). The Sig. or p-value for REAL is 0.024; this is less than the alpha value of 0.05, indicating a statistically significant increase in perceived realism for the Passive Haptic approach.

Combined Means	HTC VIVE	Haptics	Sig.
Involvement (INV)	2.72	2.87	0.817
Spatial Presence (SP)	4.13	4.83	0.089
Experienced Realism (REAL)	1.89	2.79	0.024

Figure 27: Table showing the outcome of the One ANOVA of the means for each interaction method

Further analysis of the results from each group of participants revealed that this increase in Experienced Realism when using the H.P.T was higher in Group1. Also, the results from Group1 showed a statistically significant increase in the Spatial Presence score.

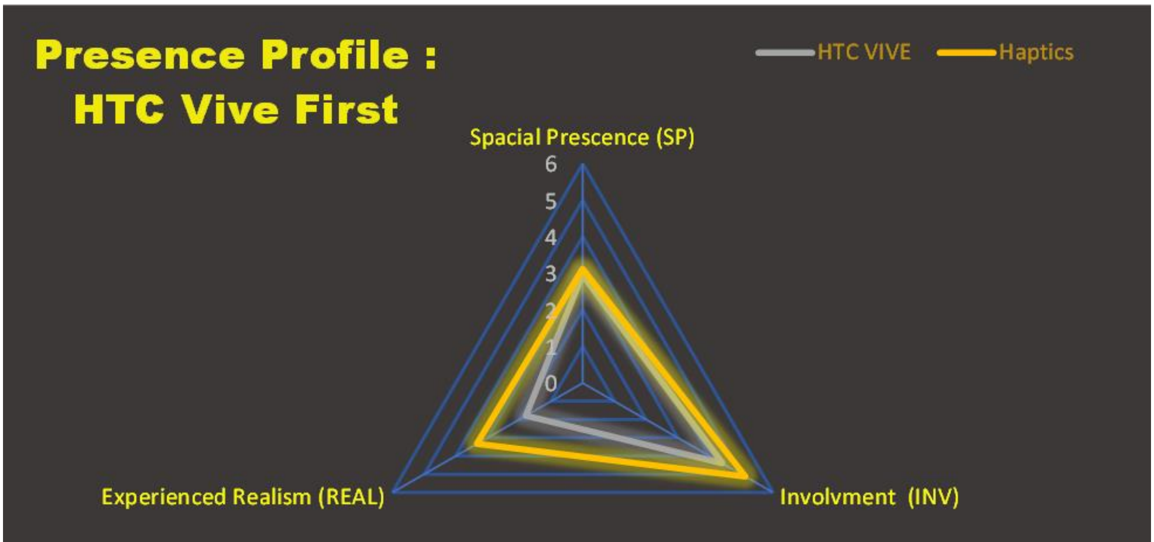


Figure 28: Presesnce Profiles of Participants who experienced the HTC Vive controller interaction method first

Fig.3 shows the Presence Profiles for each interaction method for Group1. The results of the one-way ANOVA comparison show that there is a statistically significant increase in Experienced Realism and Spatial Presence for the Passive Haptic approach (Fig.4)

Vive first	HTC VIVE	Haptics	Sig.
Involvement (INV)	2.958333	3.125	0.852
Spatial Presence (SP)	4.4	5.13	0.050
Experienced Realism (REAL)	1.79	3.33	0.007

Figure 29: Results of the one-way ANOVA for Group1: Using the H.T.C. Vive controllers first

In this case, the relevant p-values are 0.05 for S.P. and 0.007 for REAL, confirming their statistical significance. This demonstrates more positive reinforcement for the approach to using Passive Haptics offered by the H.P.T.

The analysis of the results from Group2, those who used the H.P.T approach first, showed no statistically significant difference in any of the measured fields, as the graphic in Fig.5 shows the two presence profiles are almost the same.

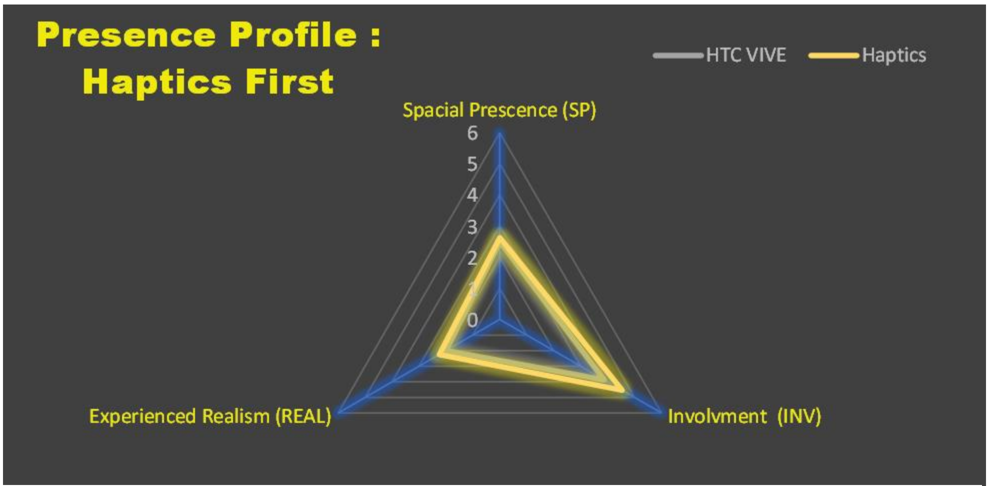


Figure 30: Presesnce Profiles of Participants who experienced the Haptic Prototype Toolkit interaction method first

The outcomes of the ANOVA confirmed these findings. However, the Passive Haptic approach does still score slightly higher in all three categories. (Fig.6).

Haptics first	HTC VIVE	Haptics	Sig.
Involvement (INV)	2.5	2.62	0.898
Spatial Presence (SP)	3.86	4.53	0.371
Experienced Realism (REAL)	2	2.25	0.636

Figure 31: Results of the one-way ANOVA for Group2: Using the Passive Haptic Approach first

(10.2.2) Conclusions from User Study:

The study demonstrated an indication that Passive Haptic Interactions conducted using the HPT result in a greater sense of Presence in the user. Which for this research work is interpreted as showing that the HPT method of interaction exhibited indications of a higher level of System Immersion than the traditional controllers.

Due to the limited size of the test group, the results do not provide enough data to prove this conclusively. However, they do provide enough information to show support for further research using this approach to Haptic Interaction and for the feasibility of the system itself. The outcome of the study highlights two opportunities for further investigation:

- There is an indication that the H.P.T method of interaction had a higher System Immersion than the current traditional controllers. Further research is needed to show if this trend continues in a more extensive and more diverse group of participants. As was noted in the literature review, restricting participants to students and academics may "inadvertently alienate a significant portion of the general population." (Oh *et al.*, 2016, p. 2). Which could negatively influence future design iterations of the Haptic Toolkit.
- There was little difference between the results of the presence profiles for Group2, participants who completed the scenario with the Passive Haptic method of interaction first. In contrast, the results were much more favourable for Group1. The implications from this could be interpreted as; any perceived increase in Presence resulted from having experienced the scenario with the Vive controllers first, which gave Group1 a base of reference on which to make a comparison. This suggests an interesting avenue for further research to determine the extent to which having the experience of the traditional interaction method to compare the H.P.T system to, influences how immersive the V.E. is perceived to be.

**Full results of the evaluation are included in the Design appendix (P:109-115). Samples of the Video Captures taken during the assessment are included in the digital appendix folder, and a copy of the I.P.Q survey and Participant Information sheet is included after the appendix at the end of this document. **

(10.3) Technical Evaluation:

The main objective of this part of the evaluation process was to assess the H.P.T based on the technical design. The outcome was to conclude the suitability of the first full system prototype and identify possible design improvements for future iterations of the system.

The assessment was carried out through observation of the participants' interactions with the hardware prototypes and review of the video captures taken during the user study. As outlined in phase five (chapter 9) each step of the user study was designed to test the Limpet modules ability to simulate a specific type of interaction.

- Step one tests and demonstrates the ability to simulate a pull/push motion. This interaction was represented by the lever. From the observations, one main issue with the current design was noted. There was an issue with the synchronisation of the motion of the physical proxy and the virtual lever. Which looked at first like system lag, but upon debugging the cause was found to be a replication of the issue discovered in the feasibility testing (4.3.3). Through repeated use, the P.O.T. used to detect rotation had moved so that rotation began in the dead zone causing the first 10-20degrees of rotation to be missed.
- Step two shows how the system can be used to create objects that can be picked up and carried. This interaction was represented by the key. Two main processes were being tested here. Firstly, the ability of the Mobile Limpet to relay rotation data in real-time: This was observed to be successful and highly responsive to changes in angle. Secondly, the efficiency of the external collision detection system developed in phase three (7.2) was assessed; this part of the system proved to be very buggy. The most common issue was the system detecting the object had been picked up before the participant had the proxy in hand, caused by accidental completion of the touch circuit; this resulted in the virtual object miss aligning with the proxy. This miss alignment was further increased as instead of aligning with the palm; held objects appeared to be attached to the centre of the index finger.

- Step three shows rotation detection for hinges to make virtual doorways. The safe door was used to represent this static interaction. The mechanism for detecting the position of the door worked to a high standard, although some slight alignment issues were discovered, due once again to the movement of the P.O.T.
- Step four demonstrates interaction with objects that have an input feature as well as physical properties. The blaster represented this interaction, and again this step had to features to assess. The collision detection performed better with this proxy object; this may be due to the position of the touch sensors, as the configuration on the handle of the blaster reduced the chances of accidental circuit completion. The virtual object also suffered from the previously identified alignment issues. However, the biggest problem with this prototype was due to the IMU unit malfunctioning at the start of the first user test, resulting in it having to be disabled for the remainder of the study. The outcome of this fault was that the blaster could only be moved on the x,y and z-axis, no rotation was possible, which provided a challenge to all participants when completing the final task.

As the technical evaluation has shown, there were several issues with the current prototype of the H.P.T, which will require addressing in future iterations of the design solution. However, the system performed well enough to establish there is merit to this approach to Passive Haptic Interaction. As despite the technical issues outlined above, they do not appear to have had an impact on the perceived realism of the virtual environment. This suggests support for the findings of earlier work by Cheng and Cairns that inconsistencies in realism do not appear to have a negative effect on Immersion. (Cheng and Cairns, 2005)

As a final measure of evaluation, the prototype system can be assessed based on a key objective of this research work, which was to ensure that any design solution was as accessible to as broad a range of developers as possible. With the intention being that through better access to research tools, this would create an opportunity for new directions in investigations into Immersion, Presence and Interactivity. The principle idea to achieve this was to ensure that the final system design was low-cost. To ensure that the design process adhered to this objective, three design criteria were developed, the final prototype design can be judged based on these criteria:

D1. Sourced from pre-established hardware where possible or constructed from easy to source pieces.

Answer: The system is built on standard electrical components that are affordable and easily sourced and using hardware that was already available but not applied in the way it is used here.

D2. Built on a commonly accessible and where available free to use development platforms.

Answer: The development platforms used were all free to use, Unity3d and the Arduino IDE, some 3D modelling work was completed in Maya as that was available to use, however, this is not a free software package, but free alternatives are available such as Blender.

D3. Where reasonable incorporate the use of upcycling of components from obsolete, 2nd hand or damage technology, to reduce cost and for the obvious environmental benefits.

Answer: The hardware prototypes created for the evaluation were all built from recycled materials or items acquired from 2nd hand stores.

As a result of designing with these guidelines in mind the final cost of the system prototype was calculated to be £307.40 (see Table.1)

Component	Number of units	Cost per unit	subtotal
Teensy 3.2	4	£18.00	£72.00
Teensy 4.0	1	£19.00	£19.00
HC-05 BT	8	£3.23	£25.84
BNO055 IMU	4	£34.95	£139.80
Small components	n/a	n/a	£15.00
Circuit Board	4	£3.94	£15.76
2nd Hand Materials	n/a	n/a	£20.00
		Total	£307.40

Table 1: Cost breakdown of the final system prototype

To give this figure some context, if we compare it to the active haptics systems currently commercially available it was previously established that they cost around £5000 per pair (Robertson, 2019), and a pair of H.T.C Vive controllers costs around £240 (HTC, 2017) so the proposed Haptic Prototype Toolkit can be considered a low-cost alternative approach to Haptic interaction.

Another clear indication that there is merit to the use of the Haptic Prototype Toolkit as a research tool is that its application here enabled the user study to be carried out. As, without the H.P.T, the type of comparison study conducted would not have been possible, without the use of expensive advanced Active Haptics. This highlights the opportunity for new avenues of research into Immersion, Presence and Interactivity offered by this approach to creating Passive Haptic Interactions.

(10.4) Conclusions from Evaluation:

The objective of the evaluation phase of this research project was to establish the feasibility of the H.P.T technical proficiency as a research tool and as a method of increased Interactivity when applying Passive Haptics to a virtual environment. The overall findings from the evaluation show positive indications that there is validity to this approach to Passive Haptic Interaction. The results suggest that there is an opportunity for new research and design work into the concept of Extending Substitutional Reality to further the overall understanding of Immersion, Presence, and Interactivity.

End of Chapter, space left intently blank

Conclusion (11)

(11.1) Introduction:

In the opening chapters of this thesis, it was proposed that the objective of this research and design project was to develop a prototype toolkit of mobile sensors. That allows the development of Haptic Interaction devices—building on the foundations of Passive Haptics and Substitutional Reality to facilitate increased levels of Interactivity and the method applied to enable this research would be an iterative design process. To support the motivation for this research and design work, literary evidence was presented that:

- Provided an overview of current approaches to investigations into Immersion, Presence and Interactivity (I.P.I) in Virtual Environments, highlighting its rapidly evolving nature and opportunities for new research.
- Demonstrated that Haptic Interaction plays a critical role in both Human-Computer Interaction and investigation into I.P.I
- Accessibility to the research equipment required to study Haptic Interaction is currently limited by cost, system portability and complexity of the set-up. As a result of this limited access research is presently restricted to academic institutions and large industrial researchers.
- Proposed a potential solution through the application of Passive Haptics, and Substitutional Reality with the requirement of further design work to increase levels of Interactivity offered by this approach.

From the analysis of the presented evidence, the initial concept idea was established and presented in chapter 3, following this was a detailed account of the design and prototyping work undertaken to refine this initial concept. The outcomes of the design process were the development of the Haptic Prototype Toolkit, which facilitated a user study that provided the means for the technical assessment of the system, as presented in the previous chapter. Now that the research and design project is completed, it is possible to reflect on the successes and achievements by reviewing the work undertaken against the original research objectives (R1-R4) proposed in chapter 3.

(11.2) Successes, Achievements and Contributions:

The first notable success of this research project can be considered to be that the prototype system designed can be used to develop new hardware prototypes and has potential as a research toolkit; this was demonstrated by the outcomes of the technical evaluation and supported by the findings of the user study as discussed in the previous chapter. However, to review the achievements of the overall project, the research objectives established at the start can be used for assessment.

- R1. Built using a technology that is based on passive haptics and does not need an expensive artificial feedback system.

Findings: The research work completed in chapter 4 provided evidence that the hand tracking system provided by the Leap Motion would enable the use of Passive Haptics and when applied in combination with the Haptic Prototype Toolkit it was demonstrated that it is possible to create an Immersive Interactive Experience without relying on expensive Active Haptic Systems.

- R2. Modular in design, for ease of adaption to new environments and control systems and creating of new hardware peripherals.

Findings: The final system design consists of a network of mobile sensors which are modular in design, and the control (HAND) is based on USB connectivity to the host P.C., which means that the H.P.T is potentially compatible with any hardware and software interfaces capable of transmitting and receiving data via a serial connection. The production of the evaluation environment outlined in phase five of the design process (chapter 9) highlights the simplicity of adapting a regular real-world space into a virtual environment through the use of the H.P.T, this chapter of the design phase also demonstrated how the system could be adapted to create a range of hardware prototypes.

- R3. Incorporate a degree of technical and user testing to establish the feasibility of the proposed approach to Passive Haptic Interaction.

Findings: The feasibility of each of the components used in the system were tested before the prototyping process began, as detailed in chapter 4. The design solution was evaluated by a limited user study which highlighted the design flaws in the current iteration of the prototype, but also provided positive indications in support of this approach to Passive Haptics.

- R4. Follow a structured, iterative design process, to produce a design solution that has the potential to be an accessible means with which to study Immersion, Presence and Interactivity.

Findings: There is a detailed account of the design process presented in chapters 5-9, that shows the iterative process taken to develop the prototype of the H.P.T. The outcomes of the user evaluation demonstrated that due to the low-cost nature of the design solution, it has the potential to be widely accessible, and there were also indications that the H.T.P could be deployed as a research toolkit.

The findings reviewed here present evidence that supports the success of this research project as a whole. The outcome of the research and design project is the development of the Haptic Prototyping Toolkit, which consists of a discrete network of mobile sensors. That allows the development of Haptic Interaction devices.

This research work contributes the foundations for a new approach to creating Passive Haptic Interactions in Virtual Environments and with further development has the potential to facilitate a range of future research work.

(11.3) Future works

The proposed ideas for the continuation of research into this projects approach to Haptic Interaction can be separated into two sections: Future Work to improve the system design and possible avenues for research directions once the technical issues highlighted in the evaluation have been addressed.

On the design side, there is potential for further development in several areas.

- Addition of extra limpets to utilise the full set of available serial ports offered by the Teensy 4.0
- Switch the Limpet modules themselves over to running on the Teensy 4.0 to allow for faster data processing, theoretically enabling more complex interactions.
- Integrate a system for force-feedback to engage the kinesthetic system further.
- Investigate the potential of developing a 3D tracking element for the system to facilitate more freedom of motion in the proxy objects.

In the case of potential directions for further, works once a more stable prototype system has been developed, the most informative next step in the design process would be to establish a new virtual environment and this time test the potential of the H.P.T system immersion in a comparison study with an Active Haptic Solution as this would help to prived conclusive data on the merits of engaging the natural touch sense.

(11.4) Final Closing Statement

This thesis represents the culmination of the research and design work that enabled the development of a new approach to haptic interaction that follows a low-cost method of engaging the human sense of touch. The work presented provides a foundation for future research into both the further development of technology that enables this type of Haptic Interaction and the role that haptics plays in creating a sense of Presence in a Virtual Environment.

Bibliography (12)

Abbasi, A. and Baroudi, U. (2012) 'Immersive environment: An emerging future of telecommunications', *IEEE Multimedia*. IEEE, 19(1), pp. 80–86. doi: 10.1109/MMUL.2012.7.

Abrash, M. (2015) *Oculus Connect 2 Keynote with Michael Abrash - YouTube*. Available at: https://www.youtube.com/watch?v=tYwKZDpsjgg (Accessed: 19 December 2019).

Achibet, M. *et al.* (2017) 'FlexiFingers: Multi-finger interaction in VR combining passive haptics and pseudo-haptics', in *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*, pp. 103–106. doi: 10.1109/3DUI.2017.7893325.

Alexander, A. L. *et al.* (2005) 'From Gaming to Training: A Review of Studies on Fidelity, Immersion, Presence, and Buy-in and Their Effects on Transfer in PC-Based Simulations and Games', *DARWARS Training Impact Group*, (1). doi: 10.1016/j.athoracsur.2004.02.012.

'Amiga 3000' (1991). Amiga.

Autodesk (2013) *Autodesk*. Available at: https://www.autodesk.co.uk/products/maya/overview?plc=MAYA&term=1-YEAR&support=ADVANCED&quantity=1 (Accessed: 14 November 2019).

Azmandian, M. *et al.* (2016) 'Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences', in *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1968–1979. doi: 10.1145/2858036.2858226.

BAE Systems (2019). Available at: https://www.baesystems.com/en/home (Accessed: 11 October 2019).

Basdogan, C. *et al.* (2000) 'An Experimental Study on the Role of Touch in Shared Virtual Environments', *ACM Transactions on Computer-Human Interaction*, 7(4), pp. 443–460. doi: 10.1145/365058.365082.

Bethesda (2018) *Wolfenstein: Youngblood | Bethesda.net*. Available at: https://bethesda.net/en/game/wolfenstein-faq (Accessed: 20 December 2019).

Bouzit, M. *et al.* (2002) 'The Rutgers Master II-ND force feedback glove', *Proceedings - 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2002*, (February 2014), pp. 145–152. doi: 10.1109/HAPTIC.2002.998952.

Bracken, C. C. and Skalski, P. (2009) 'Telepresence and video games: The impact of image quality and Skill Level', *PsychNology Journal*, 7(1), pp. 101–112.

Brewster, S. (2003) 'Haptic human-computer interaction', in *Proceedings of CHINZ 2003: The 4th Annual Conference of the ACM Special Interest Group on Computer-Human Interaction New Zealand Chapter*, pp. 3–4. doi: 10.1145/2331829.2331830.

Brown, E. and Cairns, P. (2004) 'A grounded investigation of game immersion', *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1297–1300. doi: 10.1145/985921.986048.

Burdea, G. C. (1999) 'Haptic Feedback for Virtual Reality', *Proceedings of International Workshop on Virtual prototyping, Laval, France*, (May), pp. 87–96.

Burdea, G. C. (2000) 'Haptics issues in virtual environments', *Proceedings of Computer Graphics International Conference, CGI*, (February 2000), pp. 295–302. doi: 10.1109/cgi.2000.852345.

Cairns, P. *et al.* (2006) ‘Quantifying the experience of immersion in games.’, *Cognitive Science of Games and Gameplay workshop at Cognitive Science 2006*, p. 7. Available at: http://discovery.ucl.ac.uk/149051/.
Cairns, P., Cox, A. and Imran Nordin, A. (2012) ‘Immersion in Digital Games: Review of Gaming Experience Research’, in *Handbook of Digital Games*, pp. 339–361. doi: 10.1002/9781118796443.ch12.
Cater, J. P. (1994) ‘SmelliTaste : Odors in Virtual Reality’, p. 7803.
Chagué, S. and Charbonnier, C. (2016) ‘Real virtuality: A multi-user immersive platform connecting real and virtual worlds’, in *ACM International Conference Proceeding Series*, pp. 5–7. doi: 10.1145/2927929.2927945.
Cheng, K. and Cairns, P. A. (2005) ‘Behaviour, realism and immersion in games’, *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1272–1275. doi: 10.1145/1056808.1056894.
Choi, H. *et al.* (2019) ‘A Wearable Hand Haptic Interface to Provide Skin Stretch Feedback to the Dorsum of a Hand’, *AsiaHaptics*, 1(Table 1), pp. 196–197.
Clark, T. (2014) ‘How Palmer Luckey Created Oculus Rift’, *Smithsonian Magazine*, p. 3. Available at: https://www.smithsonianmag.com/innovation/how-palmer-luckey-created-oculus-rift-180953049/?page=2 (Accessed: 18 December 2019).
Cooper, N. *et al.* (2018) ‘The effects of substitute multisensory feedback on task performance and the sense of’, *PLoS ONE*, 13(2), pp. 1–25. doi: 10.1371/journal.pone.0191846.
Cruz-Neira, C. *et al.* (1992) ‘The Cave: Audio Visual Experience Automatic Virtual Environment’, *Communication of the ACM*, 35(6). Available at: http://repositorio.unan.edu.ni/2986/1/5624.pdf.
Culbertson, H., Schorr, S. B. and Okamura, A. M. (2018) ‘Haptics: The Present and Future of Artificial Touch Sensation’, *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1), pp. 385–409. doi: 10.1146/annurev-control-060117-105043.
Denisova, A. and Cairns, P. (2015) ‘First person vs. Third person perspective in digital games: Do player preferences affect immersion?’, in *Conference on Human Factors in Computing Systems - Proceedings*, pp. 145–148. doi: 10.1145/2702123.2702256.
Dexta Robotics (2019). Available at: https://www.dextarobotics.com/en-us (Accessed: 15 October 2019).
Eid, M. A. and Al Osman, H. (2016) ‘Affective Haptics: Current Research and Future Directions’, *IEEE Access*, 4, pp. 26–40. doi: 10.1109/ACCESS.2015.2497316.
ElKoura, G. and Singh, K. (2003) ‘Handrix: Animating the human hand’, *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2003*.
Engelbart, D. C. (1970) ‘X-Y Position Indicator for a Display System’, *Pulse*, pp. 1–7.
Estrada, J. G. and Simeone, A. L. (2017) ‘Recommender system for physical object substitution in VR’, in *Proceedings - IEEE Virtual Reality*, pp. 359–360. doi: 10.1109/VR.2017.7892325.
Fani, S. *et al.* (2018) ‘W-FYD: A Wearable Fabric-Based Display for Haptic Multi-Cue Delivery and Tactile Augmented Reality’, *IEEE Transactions on Haptics*. IEEE, 11(2), pp. 304–316. doi: 10.1109/TOH.2017.2708717.
Fellgett, P. B. (1974) ‘Ambisonic reproduction of directionality in surround-sound systems’, *Nature*, pp. 534–538. doi: 10.1038/252534b0.
Freeman, D. *et al.* (2017) ‘Virtual reality in the assessment, understanding, and treatment of mental health disorders’, *Psychological Medicine*, 47(14), pp. 2393–

2400. doi: 10.1017/S003329171700040X.
Garcia, J. F. *et al.* (2018) ‘Inside looking out or outside looking in? An evaluation of visualisation modalities to support the creation of a substitutional virtual environment’, in *Proceedings of the Workshop on Advanced Visual Interfaces AVI*. doi: 10.1145/3206505.3206529.
Georgiou, O. *et al.* (2018) ‘Touchless Haptic Feedback for VR Rhythm Games’, *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings*, (March), pp. 553–554. doi: 10.1109/VR.2018.8446619.
Giles, M. and Kerry, B. (1994) ‘Cyberbase SU2000 Technical Manual’. Available at: http://arianchen.de/su2000/files/su2000tech.pdf.
Google Cardboard (2019). Available at: https://arvr.google.com/cardboard/ (Accessed: 12 October 2019).
Gu, X. *et al.* (2016) ‘Dexmo’, in, pp. 1991–1995. doi: 10.1145/2858036.2858487.
Hayward, V. *et al.* (2004) ‘Haptic interfaces and devices’, *Sensor Review*, 24(1), pp. 16–29. doi: 10.1108/02602280410515770.
Heilig, M. L. (1957) ‘Stereoscopic-Television’. USA.
Heilig, M. L. (1961) ‘Sensorama Simulator’, *Sensorama*. USA. doi: 10.1093/acprof:oso/9780198732655.001.0001.
Henderson, S. J. and Feiner, S. (2007) ‘Opportunistic Controls: Leveraging Natural Affordances as Tangible User Interfaces for Augmented Reality’, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, p. 7.
Hettiarachchi, A. and Wigdor, D. (2016) ‘Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality’, in *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1957–1967. doi: 10.1145/2858036.2858134.
Hoang, T. N., Smith, R. T. and Thomas, B. H. (2013) ‘Passive Deformable Haptic glove to support 3D interactions in mobile augmented reality environments’, in *2013 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013*. IEEE, pp. 257–258. doi: 10.1109/ISMAR.2013.6671793.
Hollerbach, J. M. (2000) ‘Some current issues in haptics research’, *Proceedings - IEEE International Conference on Robotics and Automation*, 1(April), pp. 757–762. doi: 10.1109/robot.2000.844142.
Horsburgh, A. J., Mcalpine, K. and Clark, D. F. (2011) ‘A Perspective on the Adoption of Ambisonics for Games’, (January 2011), pp. 1–6.
HTC (2017) ‘VIVE | Controller’. Available at: https://www.vive.com/uk/accessory/controller/.
Huynh, P. and Yoo, M. (2016) ‘VLC-based positioning system for an indoor environment using an image sensor and an accelerometer sensor’, *Sensors (Switzerland)*, 16(6). doi: 10.3390/s16060783.
igroup.org (2001). Available at: http://www.igroup.org/pq/ipq/index.php (Accessed: 7 October 2019).
Insko, B. E. (2001) ‘Passive haptics significantly enhances virtual environments’, *Computer*, p. 100.
Iwata, S. (2019) *Shigeru Miyamoto Talks Virtual Boy*. Available at: https://www.nintendo.co.uk/Iwata-Asks/Iwata-Asks-Nintendo-3DS/Vol-1-And-That-s-How-the-Nintendo-3DS-Was-Made/2-Shigeru-Miyamoto-Talks-Virtual-Boy/2-Shigeru-Miyamoto-Talks-Virtual-Boy-229419.html (Accessed: 12 October 2019).
Jennett, C. *et al.* (2008) ‘Measuring and defining the experience of immersion in games’, *International Journal of Human Computer Studies*, 66(9), pp. 641–661. doi: 10.1016/j.ijhcs.2008.04.004.
Jeon, S. and Choi, S. (2009) ‘Haptic augmented reality: Modulation of real object stiffness’, in *Proceedings - 3rd Joint EuroHaptics Conference and Symposium on*

Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2009, pp. 384–385. doi: 10.1109/WHC.2009.4810907.
Jones, L. A. (2018) *Haptics*. 1st edn. Cambridge,MA: The MIT Press.
Joseph L. Scanlan (1988) '"Star Trek: The Next Generation" The Big Goodbye'. Paramount Domestic Television.
Karunanayaka, K. *et al.* (2018) 'New thermal taste actuation technology for future multisensory virtual reality and internet', *IEEE Transactions on Visualization and Computer Graphics*. IEEE, 24(4), pp. 1496–1505. doi: 10.1109/TVCG.2018.2794073.
Kazikli, E. and Gezici, S. (2018) 'Hybrid TDOA/RSS Based Localization for Visible Light Systems', pp. 2–27. Available at: http://arxiv.org/abs/1802.03991.
Kearney, G. *et al.* (2016) 'Design of an Interactive Virtual Reality System for Ensemble Singing', *Proceedings of the Interactive Audio Systems Symposium*. Available at: https://www.york.ac.uk/sadie-project//IASS2016/IASS_Papers/IASS_2016_paper_1.pdf.
Korres, G. and Eid, M. (2016) 'Haptogram: Ultrasonic Point-Cloud Tactile Stimulation', *IEEE Access*, 4(September), pp. 7758–7769. doi: 10.1109/ACCESS.2016.2608835.
Kosinski, J. (2010) *Tron : Legacy*. United States: Walt Disney Studios Motion Pictures.
Lee, K. (2007) 'Microsoft Hardware Timeline'.
Liang, Q. and Liu, M. (2017) 'Plugo: a VLC Systematic Perspective of Large-scale Indoor Localization', pp. 1–11. Available at: http://arxiv.org/abs/1709.06926.
Long, B. *et al.* (2014) 'Rendering volumetric haptic shapes in mid-air using ultrasound', *ACM Transactions on Graphics*, 33(6). doi: 10.1145/2661229.2661257.
Luzhnica, G., Veas, E. and Pammer, V. (2016) 'Skin reading: Encoding text in a 6-channel haptic display', *International Symposium on Wearable Computers, Digest of Papers*, 12-16-Sept(October 2017), pp. 148–155. doi: 10.1145/2971763.2971769.
Mackinlay, J., Card, S. K. and Robertson, G. G. (1990) 'A Semantic Analysis of the Design Space of Input Devices', *Human-Computer Interaction*, 5(2–3), pp. 145–190. doi: 10.1080/07370024.1990.9667153.
Maisto, M. *et al.* (2017) 'Evaluation of wearable haptic systems for the fingers in augmented reality applications', *IEEE Transactions on Haptics*. IEEE, 10(4), pp. 511–522. doi: 10.1109/TOH.2017.2691328.
Martinez, J. *et al.* (2018) 'Touchless Haptic Feedback for Supernatural VR Experiences', *25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings*, (March), pp. 629–630. doi: 10.1109/VR.2018.8446522.
Mestre, D. R. (2005) 'Immersion and Presence', *Movement & Perception*.
Microsoft (2017) *Kinect hardware*, *developer.microsoft.com*. Available at: https://developer.microsoft.com/en-us/windows/kinect/hardware (Accessed: 23 October 2017).
MordorIntelligence (2019) *Virtual Reality (VR) Market | Growth, Trends, and Forecast (2020-2025)*. Available at: https://www.mordorintelligence.com/industry-reports/virtual-reality-market (Accessed: 2 January 2020).
Mortonheilig.com (no date) *index*. Available at: http://www.mortonheilig.com/ (Accessed: 18 December 2019).
Mütterlein, J. (2018) 'The Three Pillars of Virtual Reality? Investigating the Roles of Immersion, Presence, and Interactivity', in *Proceedings of the 51st Hawaii International Conference on System Sciences*, pp. 1407–1415. doi: 10.24251/hicss.2018.174.

Nintendo (1997) *Nintendo*. Available at: https://www.nintendo.co.uk/Corporate/Nintendo-History/Nintendo-64/Accessories/Accessories-627128.html (Accessed: 10 November 2019).
Nintendo (2017) *The official home for The Legend of Zelda - Home*. Available at: https://www.zelda.com/ (Accessed: 20 December 2020).
Oculus (2020) *Oculus| VR Headsets and Equipment*, *Facebook Technologies, LLC.* Available at: https://www.oculus.com/ (Accessed: 2 June 2020).
ODA, N. (2000) 'A Visual Motion Control Strategy of Multi-Degrees-of-Feedom Manipulators using Virtual Image Space'.
Oh, S. Y. *et al.* (2016) 'Immersion at scale: Researcher's guide to ecologically valid mobile experiments', in *Proceedings - IEEE Virtual Reality*, pp. 249–250. doi: 10.1109/VR.2016.7504747.
Ortolani, F. (2015) 'Introduction to Ambisonics A tutorial for beginners in 3D audio', *Excerpt from the technical papers written during the development of Ambisonic Auralizer*, pp. 1–25. Available at: http://www.ironbridge-elt.com/downloads/FrancescaOrtolani-IntroductionToAmbisonics.pdf.
Pacchierotti, C., Prattichizzo, D. and Kuchenbecker, K. J. (2015) 'Displaying Sensed Tactile Cues with a Fingertip Haptic Device', *IEEE Transactions on Haptics*, 8(4), pp. 384–396. doi: 10.1109/TOH.2015.2445770.
Parisi, D. (2018) *Archaeologies of Touch*. 1st edn. Minneapolis: University of Minnesota Press.
PJRC, R. (no date) *Teensyduino: Download and Install Teensy support into the Arduino IDE*. Available at: https://www.pjrc.com/teensy/td_download.html (Accessed: 10 December 2019).
Playstaion (2018) *DUALSHOCK wireless controller*, *Playstation*. Available at: https://www.playstation.com/en-us/explore/accessories/gaming-controllers/dualshock-4/ (Accessed: 12 November 2019).
'PlayStation VR' (2019). Sony. Available at: https://www.playstation.com/en-gb/explore/playstation-vr/ (Accessed: 15 October 2019).
Poeschl, S. and Doering, N. (2013) 'The german VR simulation realism scale - psychometric construction for virtual reality applications with virtual humans', *Annual Review of CyberTherapy and Telemedicine*, 11(June), pp. 33–37. doi: 10.3233/978-1-61499-282-0-33.
Poeschl, S., Wall, K. and Doering, N. (2013) 'Integration of spatial sound in immersive virtual environments an experimental study on effects of spatial sound on presence', *Proceedings - IEEE Virtual Reality*, (March), pp. 129–130. doi: 10.1109/VR.2013.6549396.
Ponraj, G. and Ren, H. (2018) 'Sensor Fusion of Leap Motion Controller and Flex Sensors Using Kalman Filter for Human Finger Tracking', *IEEE Sensors Journal*. IEEE, 18(5), pp. 2042–2049. doi: 10.1109/JSEN.2018.2790801.
Prattichizzo, D. *et al.* (2018) *Haptics : Science , Technology , 11th International Conference, EuroHaptics 2018 ,*.
Price, S. (2019) *The Original Homunculus Company*. Available at: https://www.sharonpricejames.com/the-original-homunculus-company.html (Accessed: 10 November 2019).
Ramadasan, D., Chevaldonne, M. and Chateau, T. (2015) 'Real-time SLAM for static multi-objects learning and tracking applied to augmented reality applications', in *2015 IEEE Virtual Reality Conference, VR 2015 - Proceedings*, pp. 267–268. doi: 10.1109/VR.2015.7223398.
Ramic-Brkic, B. and Chalmers, A. (2010) 'Virtual smell: Authentic smell diffusion in virtual environments', *Proceedings of AFRIGRAPH 2010: 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in*

Africa, 1(212), pp. 45–52. doi: 10.1145/1811158.1811166.
Ranasinghe, N. *et al.* (2012) ‘Tongue mounted interface for digitally actuating the sense of taste’, *Proceedings - International Symposium on Wearable Computers, ISWC*. IEEE, pp. 80–87. doi: 10.1109/ISWC.2012.16.
Ranasinghe, N. *et al.* (2013) ‘Simulating the sensation of taste for immersive experiences’, *ImmersiveMe 2013 - Proceedings of the 2nd International Workshop on Immersive Media Experiences, Co-located with ACM Multimedia 2013*, pp. 29–34. doi: 10.1145/2512142.2512148.
Robertson, A. (2019) ‘Teslasuit’s new VR gloves let you feel virtual objects and track your pulse’, *The Verge*. Available at: https://www.theverge.com/2019/12/26/21037855/teslasuit-glove-vr-haptic-feedback-glove-announce-pricing-release-date-ces-2020 (Accessed: 27 December 2019).
Rosson, L. (2014) *The Virtual Interface Environment Workstation (VIEW), 1990*. Available at: https://www.nasa.gov/ames/spinoff/new_continent_of_ideas/ (Accessed: 18 December 2019).
Rotberg, E., Rubin, O. and Hector, R. (1983) ‘BattleZone’. Atari Inc. Available at: https://videogamehistory.fandom.com/wiki/Battlezone.
Rowley, P. (2018) *What does Jaron Lanier, godfather of virtual reality, think of VR?*, *PHD Media Worldwide*. Available at: https://www.phdmedia.com/what-does-jaron-lanier-godfather-of-virtual-reality-think-about-vr-phil-rowley-phd-global-business/ (Accessed: 18 December 2019).
Ruotolo, F. *et al.* (2013) ‘Immersive virtual reality and environmental noise assessment: An innovative audio-visual approach’, *Environmental Impact Assessment Review*. Elsevier Inc., 41, pp. 10–20. doi: 10.1016/j.eiar.2013.01.007.
Samsung Gear VR (2019). Available at: https://www.samsung.com/global/galaxy/gear-vr/ (Accessed: 16 October 2019).
Schubert, T., Friedmann, F. and Regenbrecht, H. (2001) ‘The experience of presence: Factor analytic insights’, *Presence: Teleoperators and Virtual Environments*, 10(3), pp. 266–281. doi: 10.1162/105474601300343603.
SEGA (2019). Available at: https://www.sega.co.uk/ (Accessed: 11 October 2019).
Sholes, C. L., Glidden, C. and Soule, S. W. (1868) ‘Type-Writing Machine’. USa.
Simeone, A. L. (2015) ‘Substitutional reality: Towards a research agenda’, in *2015 IEEE 1st Workshop on Everyday Virtual Reality, WEVR 2015*, pp. 19–22. doi: 10.1109/WEVR.2015.7151690.
Simeone, A. L. and Velloso, E. (2015) ‘Substitutional reality: Bringing virtual reality home’, *XRDS: Crossroads, The ACM Magazine for Students*, 22(1), pp. 24–29. doi: 10.1145/2810044.
Simeone, A. L., Velloso, E. and Gellersen, H. (2015) ‘Substitutional reality: Using the physical environment todesign virtual reality experiences’, in *Conference on Human Factors in Computing Systems - Proceedings*, pp. 3307–3316. doi: 10.1145/2702123.2702389.
Singer, M. J. and Witmer, B. G. (1998) ‘Measuring Presence in Virtual Environments : A Presence Questionnaire’, *Prescence*, 7(3), pp. 225–241.
Slater, M. *et al.* (1996) ‘Immersion, presence and performance in virtual environments : An Experiment with Tri-Dimensional Chess’, (January 2013), pp. 163–172. doi: 10.1145/3304181.3304216.
Slater, M. (1999) ‘Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire’, *Presence: Teleoperators and Virtual Environments*, 8(5), pp. 560–565. doi: 10.1162/105474699566477.
Slater, M. (2003) ‘A note on presence’, *Presence Connect*, 3(January 2003), pp. 1–5. Available at: www.cs.ucl.ac.uk/staff/m.slater.

Slater, M. (2018) ‘Immersion and the illusion of presence in virtual reality’, *British Journal of Psychology*, 109(3), pp. 431–433. doi: 10.1111/bjop.12305.
Slater, M. and Steed, A. (2000) ‘A virtual presence counter’, *Presence: Teleoperators and Virtual Environments*, 9(5), pp. 413–434. doi: 10.1162/105474600566925.
Slater, M. and Wilbur, S. (1997) ‘A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments Mel’, *Presence: Teleoperators & Virtual Environments*, 6, pp. 603–616.
Spanlang, B. *et al.* (2010) ‘A first person avatar system with haptic feedback’, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, pp. 47–50. doi: 10.1145/1889863.1889870.
Spencer, B. S. (2005) ‘Incorporating the sense of smell into haptic surgical simulators’, *Studies in Health Technology and Informatics*, 114(1), pp. 54–62.
Spielberg, S. (2018) *Ready Player One*. USA: Warner Bros.
SquareEnix (2018) ‘Shadow of the Tomb Raider’. Available at: https://tombraider.square-enix-games.com/en-gb (Accessed: 20 December 2020).
Sra, M. and Schmandt, C. (2016) ‘Bringing real objects, spaces, actions, and interactions into social VR’, in *2016 IEEE 3rd VR International Workshop on Collaborative Virtual Environments, 3DCVE 2016*, pp. 16–17. doi: 10.1109/3DCVE.2016.7563561.
Srinivasan, M. A. and Basdogan, C. (1997) ‘Haptics in virtual environments: Taxonomy, research status, and challenges’, *Computers and Graphics (Pergamon)*, 21(4), pp. 393–404. doi: 10.1016/s0097-8493(97)00030-7.
Steinicke, F. *et al.* (2008) ‘The holodeck construction manual’, in *SIGGRAPH’08 - ACM SIGGRAPH 2008 Posters*. doi: 10.1145/1400885.1400988.
Suhail, M. *et al.* (2017) ‘Redirected reach in virtual reality: Enabling natural hand interaction at multiple virtual locations with passive haptics’, in *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017 - Proceedings*. IEEE, pp. 245–246. doi: 10.1109/3DUI.2017.7893363.
Sutherland, I. E. (1968) ‘The Ultimate Display’, *Information Processing Techniques Office, ARPA, OSD We*.
Suzuki, K., Wakisaka, S. and Fujii, N. (2012) ‘Substitutional reality system: A novel experimental platform for experiencing alternative reality’, *Scientific Reports*, 2, pp. 1–9. doi: 10.1038/srep00459.
Turner, J. (2018) *ImmotionVR*, *IMMOTIONVR*. Available at: https://immotionvr.com/blog/the-history-of-virtual-reality/ (Accessed: 10 June 2020).
Ubisoft (2018) *Tom Clancy’s The Division 2*. Available at: https://store.ubi.com/uk/tom-clancy-s-the-division-2-tm-/5b06a3984e0165fa45ffdcc5.html?ncid=287-3---1-pdsc-6-11-Store_Search_TCTD2_SUMMER_0620--1-1-7-0620-4----ID_89253-----&maltcode=ubisoftstore_convst_pdsc_google___UBISOFT____ (Accessed: 20 December 2020).
Ultraleap (2019) *Tracking* | *Leap Motion Controller* | *Ultraleap*. Available at: https://www.ultraleap.com/product/leap-motion-controller/ (Accessed: 2 June 2020).
Usoh, M. *et al.* (2019) ‘Using Presence Questionnaires in Reality’, in *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–16. doi: 10.1145/3290605.3300590.
Valve (2017) *VIVE™* | *Vive Tracker*, *Steam*. Available at: https://www.vive.com/uk/vive-tracker/ (Accessed: 9 August 2018).
Varga, S. (2019) ‘HaptX | Haptic gloves for VR training, simulation, and design’,

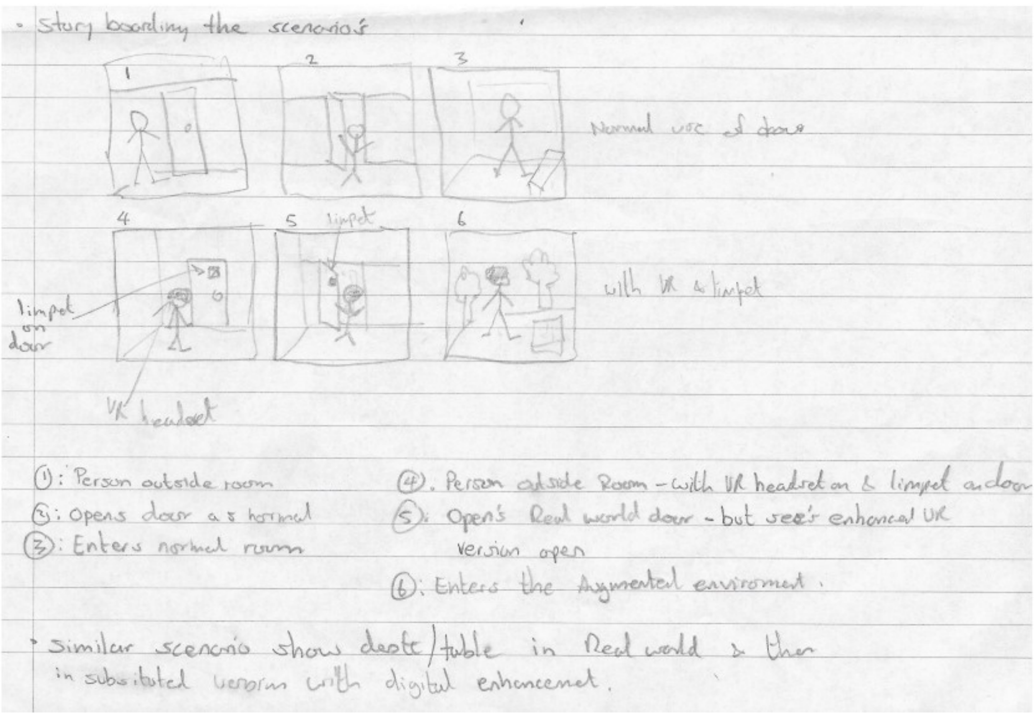
HaptX. Available at: https://haptx.com/ (Accessed: 15 November 2019).
Verron, C. *et al.* (2010) 'A 3-D immersive synthesizer for environmental sounds', *IEEE Transactions on Audio, Speech and Language Processing*, 18(6), pp. 1550–1561. doi: 10.1109/TASL.2009.2037402.
Viciana-Abad, R., Reyes-Lecuona, A. and Cañadas-Quesada, F. J. (2005) 'Difficulties Using Passive Haptic Augmentation in the Interaction within a Virtual Environment', in *Presence*, pp. 287–292.
Vive (2018) 'VIVE™ | Discover Virtual Reality Beyond Imagination'. Available at: https://www.vive.com/uk/ (Accessed: 2 June 2020).
VPL Research | C-SPAN.org (no date). Available at: https://www.c-span.org/organization/?112710/VPL-Research (Accessed: 18 December 2019).
W Industries (1991) 'Dactyl Nightmare'. W Industries. Available at: http://www.arcadianvr.com/SU_2000_SOFTWARE_DACTYL2.htm.
Wachowski, Lana and Wachowski, Lilly (1999) *The Matrix*. UK: WarnerBros.
Weibel, D. and Wissmath, B. (2011) 'Immersion in computer games: The role of spatial presence and flow', *International Journal of Computer Games Technology*, 2011. doi: 10.1155/2011/282345.
Whitton, M. *et al.* (2005) 'Intergrating Real and Virtual Objects in Virtual Environments'.
Xu, W., Wang, B. and Jiang, Y. (2017) 'Multi-target indoor tracking and recognition system with infrared markers for virtual reality', in *Proceedings of 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2017*, pp. 1549–1553. doi: 10.1109/IAEAC.2017.8054273.
Zadeh, M. H., Wang, D. and Kubica, E. (2007) 'Human factors for designing a haptic interface for interaction with a virtual environment', in *HAVE 2007 - The 6th IEEE International Workshop on Haptic, Audio and Visual Environments and Games, Proceedings*, pp. 21–26. doi: 10.1109/HAVE.2007.4371580.
Zhuang, Y. *et al.* (2018) 'A Survey of Positioning Systems Using Visible LED Lights', *IEEE Communications Surveys and Tutorials*, (January). doi: 10.1109/COMST.2018.2806558.

Design Appendix (13)

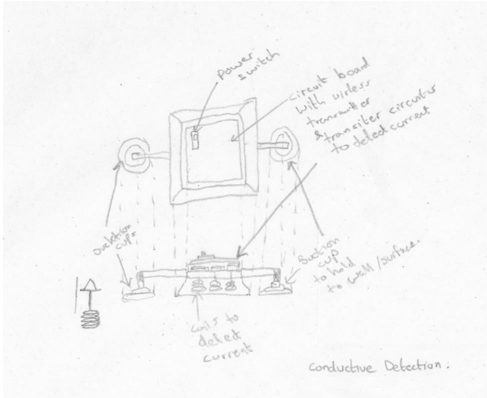
Contained in this document are notes and photographic evidence that accompany the design work detailed in the main body of text.

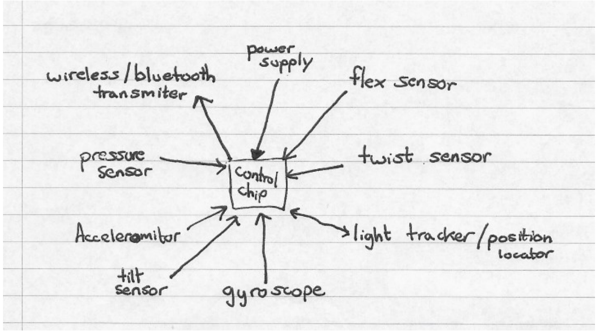
Initial Design Brief:

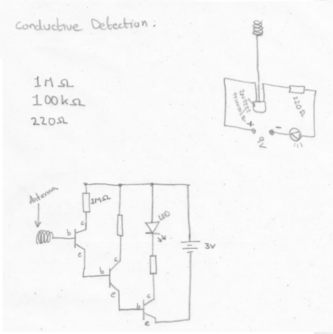
To develop a Hardware Prototyping Toolkit (H.P.T), designed around a Core System Architecture (C.S.A) to facilitate the use of passive haptics in a virtual environment. By building on the principal idea of Substitutional Reality (S.R), by overlaying digital content onto real objects and enhancing this with a system that allows for interaction to be detected and translated into a digital response.

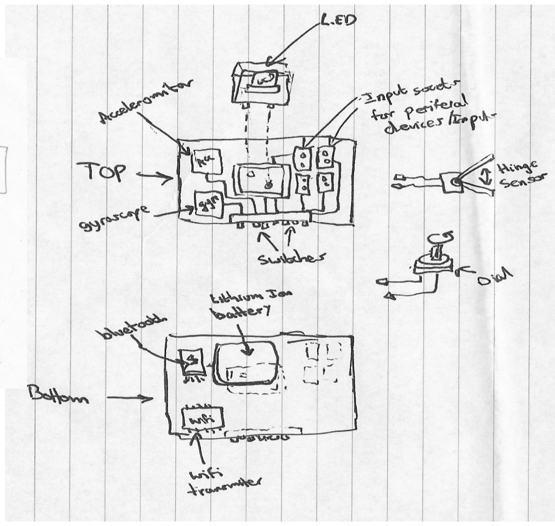


Concept Ideas:



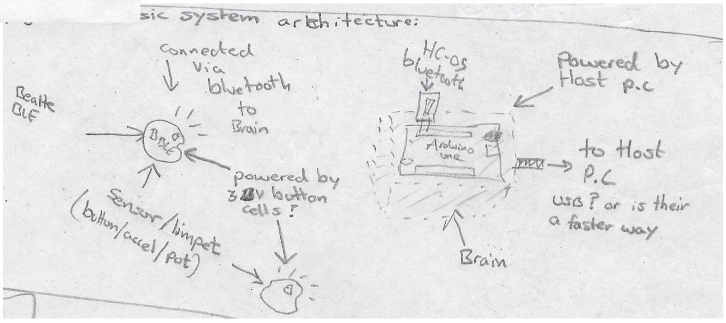




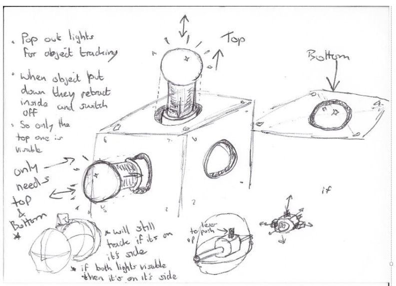


- Create a standard table of coded signals
- record the time precisly between pulses
- Use time of flight calculations to mesure the difference in time between recived pullses and the database, see if you can calculate location.

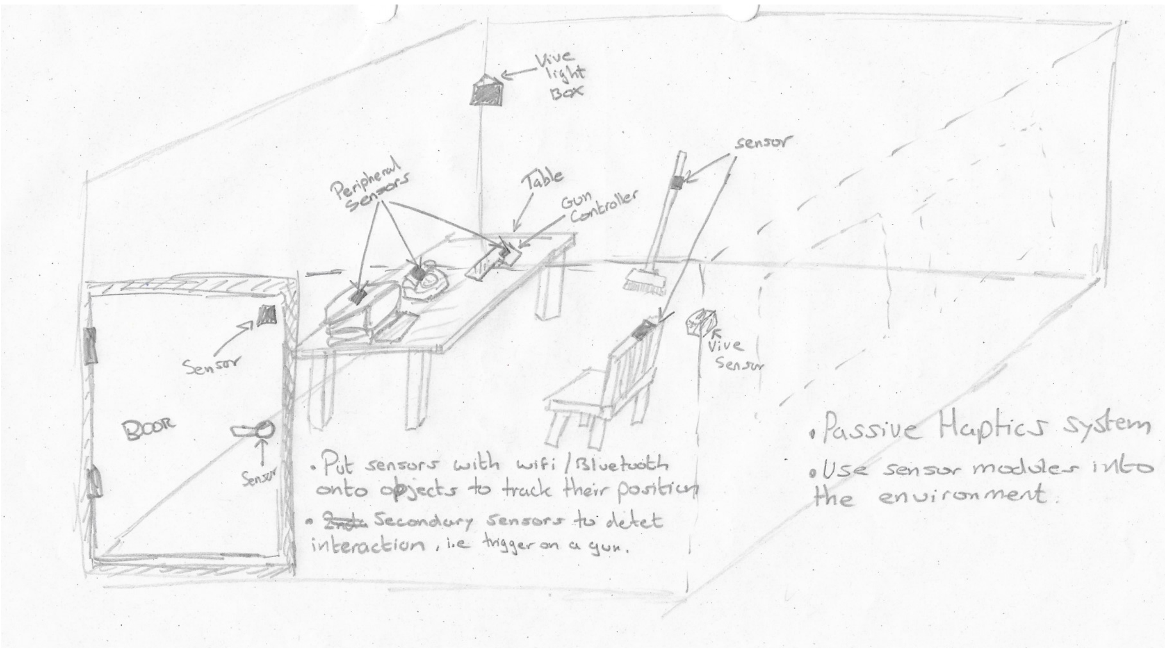
- Flashing from a fixed location will always arrive in the same time between pullses, if you move loction up or down then the time will change and you can cross refrence this to see if ~~the~~ object is moving up or down.

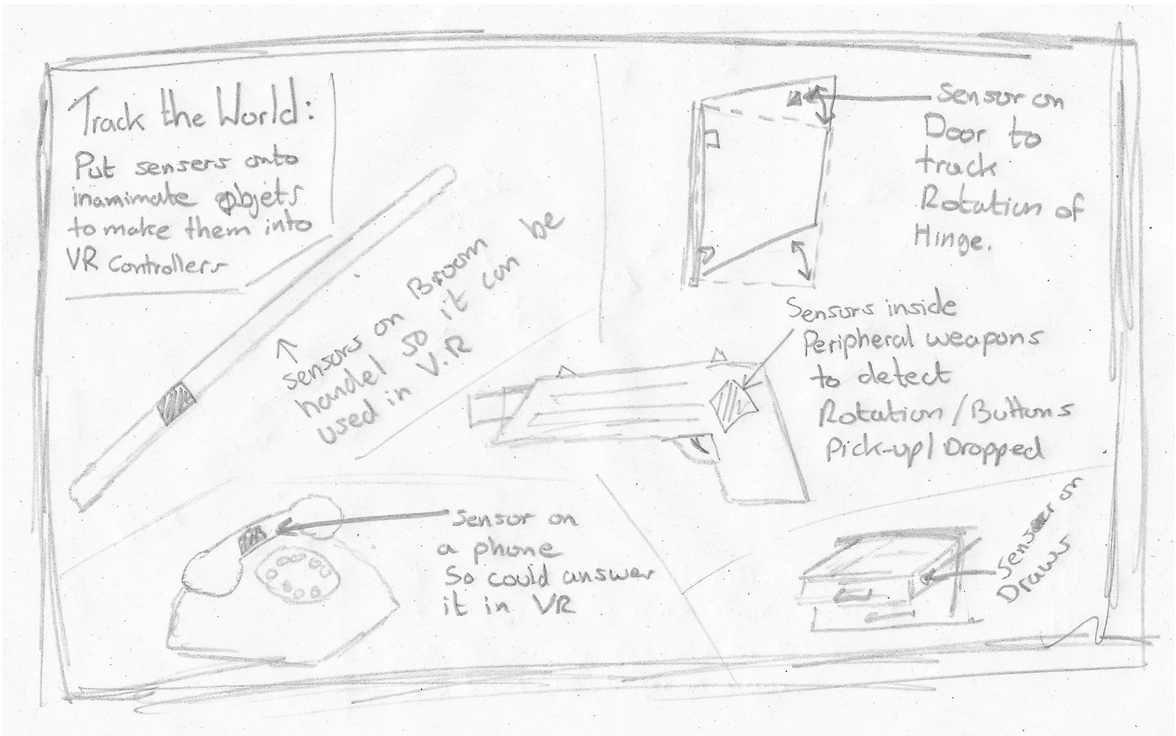


- central Control station
- bluetooth controlled periferals / or hardwired to control station
- Control unit comunicates via wifi with main server



Concept Idea:

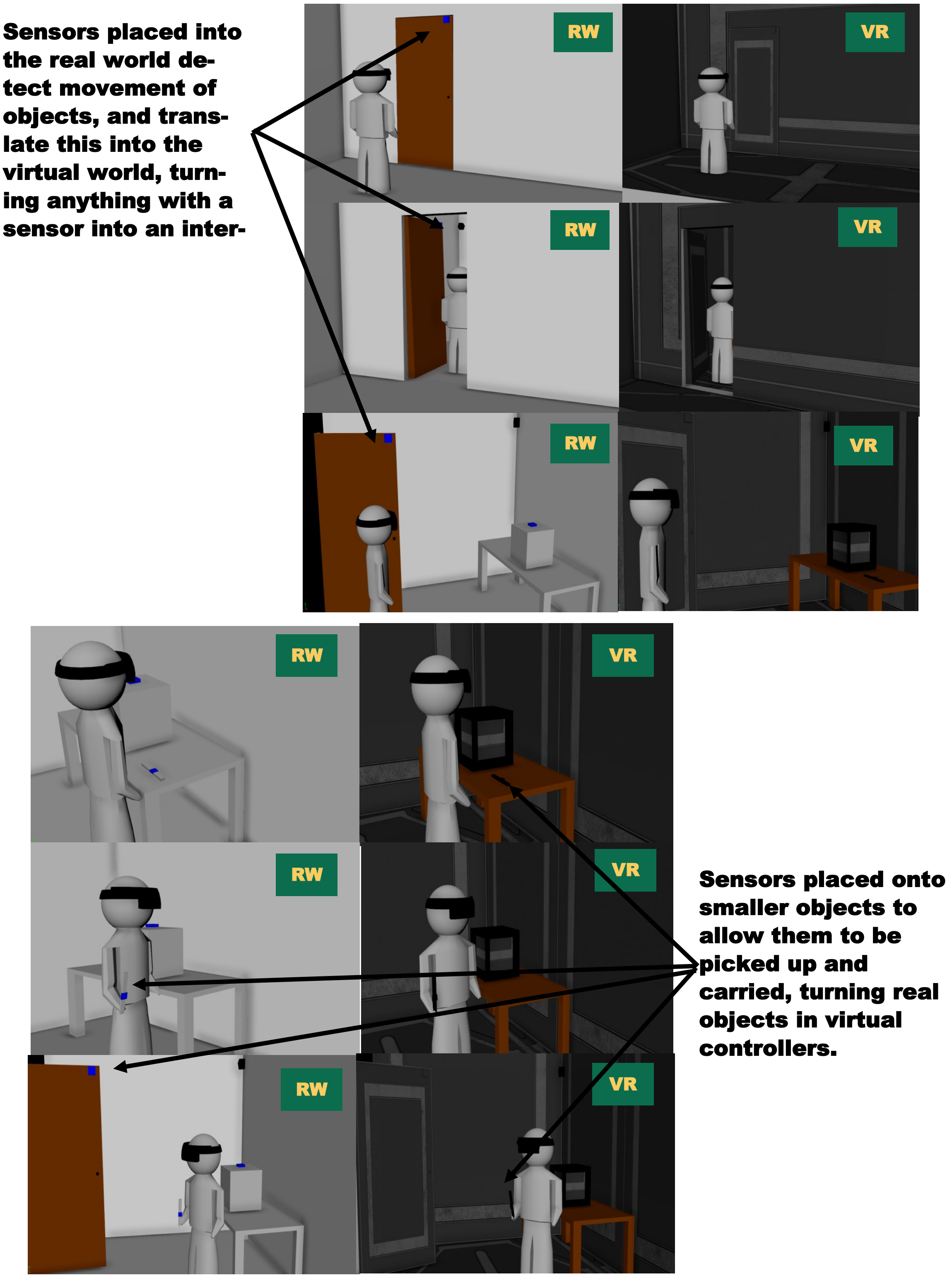




Concept Idea: cont..

Comic strip below outlining how the concept of the system will work:

By tracking the real-world can use the physical environment to create objects in the virtual space that a user can physically interact with.



Development Equipment: VR Headsets:

- Oculus Rift S Headset
- Oculus VR
- Retail Price £400.00
- Display: LCD 2560*1440 (1280x1440 per eye)
- Input: 2nd Gen Oculus Touch Controller
- Connectivity: Display port 1.2, USB 3.0

- **HTC VIVE V1.0**
- **HTC/VALE**
- **Retail Price £350.00 (complete)**
- **Display: OLED 2160x1200 (1080x1200 per eye)**
- **Input: Steam VR controller**
- **Connectivity: HDMI 1.4, USB 3.0**

- HTC VIVE PRO
- HTC/VALE
- Retail Price £1299.00 (complete)
- Display: AMOLED 2880x1600 (1440x1600 per eye)
- Input: 2nd Gen Steam VR controller/ plus hand tracking
- Connectivity: Display Port 1.2, USB 3.0

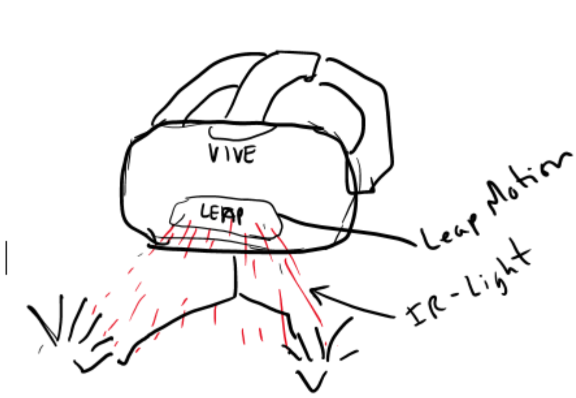
- OSVR HDK 1.4
- Razor
- Retail Price £350.00 (HMD only)
- Display: OLED 1280 x 1080 per eye
- Input: works with custom con-trollers
- Connectivity: 2 x USB 3.0, HDMI 1.4 and audio out.

Development Equipment: VR Headsets: cont...

Of the headsets currently available the PSVR has been omitted from selection as it requires a PlayStation to operate and the range of Windows Mixed reality HMDs due them being already obsolete. The remining headsets are all completable with open source or free to use dev platforms so have potential for use in this project.

Selected the HTV VIVE v1.0 , as it

- ◊ offers the best combination of screen, functionality and price.
- ◊ Has a vast online support network of tutorial docs, SDK's and software library's etc..
- ◊ HMD has available USB for addition of peripheral devices.
- ◊ Has built in camera for potential hand/object tracking
- ◊ Can operate tethered or wirelessly (with adapter)



Development Equipment: Hand Tracking:

implementing hand tracking using computer vision, but as support for this on the HTC Vive v1.0 is only currently available in Beta form and not reliable through the built in camera in HMD, to get this working properly will require an external camera, and if I'll need to add another devices to make it work it makes more sense to use one the has a proven record of high accuracy and reliability.

Features

- Browse the web, read articles, flip through photos, and play music just by lifting a finger.
- Draw, paint, and design with your fingertip. You can even use a real pencil or paintbrush.
- Slice falling fruit and shoot bad guys with your finger. Steer cars and fly planes with your hands.
- Sculpt, mold, stretch, bend, and build 3D objects. Take things apart and put them back together.
- Reach into the universe, grab the stars, and soar around the sun. It's a whole new way to learn.
- Pick, pluck, strum, and drum. Play air guitar, air harp, air everything. For real.

Specifications

- Minimum System Requirements
- Windows 7 or 8 or Mac OS X 10.7
- AMD PhenomTM II or Intel® CoreTM i3 / i5 / i7 Processor
- 2 GB RAM
- USB 2.0 port
- Internet connection
- Size: 0.5" (H) x 1.2" (W) x 3" (D)
- Weight: 0.1 pounds

Includes

- Controller
- USB Cables

As the aim is to engage the natural touch sense then the best option is to use the Leap Motion controller. It has

◊ support for integration into the HTC VIVE/PRO and Oculus HMDs.

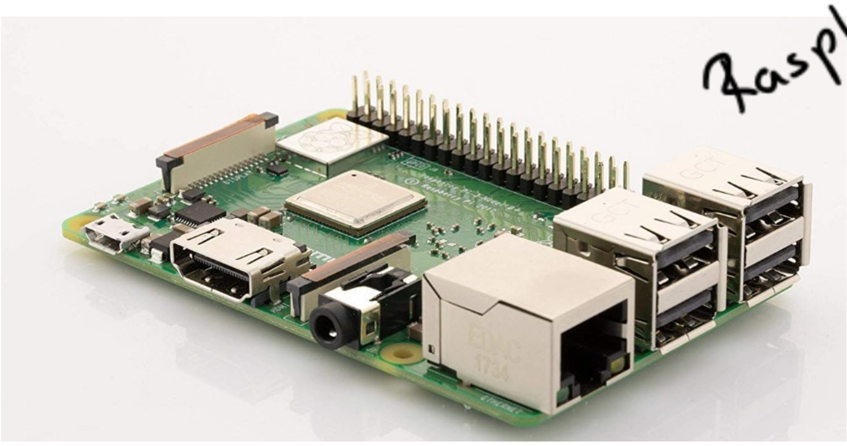
◊ developer portal and support software for free and open source dev environments.

◊ 120fps capture rate suing IR cameras so accurate

◊ 135 degree Filed of View

Development Equipment: Microcontrollers:

As the intention is to create a modular system of peripherals , they will need to be powered by some form of microcontroller :



Raspberry Pi 3

£ 35.00

Brand name	Raspberry Pi
Item Weight	86.2 g
Product Dimensions	9 x 6 x 2 cm
Item model number	Raspberry Pi 3 Model B+
Series	RPI 3 Model B +
Color	Green
Screen Size	0.01
Processor Brand	Broadcom BCM2837B0
Processor Speed	1.4 GHz
Processor Count	1
RAM Size	1 GB
Computer Memory Type	DDR2 SDRAM
Graphics Coprocessor	Broadcom
Graphics Card Description	Broadcom Graphics Chipset
Graphics Card Interface	Integrated
Wireless Type	802.11bgn, 802.11ac
Number of USB 2.0 Ports	4
Voltage	5 volts

Pro's

- Powerful
- capable of controlling screens as well as Periferals

Con's

- Big in relation to other options
- Over kill for this Project.

Development Equipment: Microcontrollers: cont..

BeaglBone Black £65.00

Brand name	Waveshare
Item Weight	9.07 g
Product Dimensions	18 x 10 x 9 cm
Item model number	BB BLACK
Color	BB BLACK
Processor Socket	LGA 775
RAM Size	512 MB

Has similar spec's to the Raspbery Pi but again for purpose of this projet maybe over kill, as can run full operating system - not needed.

Development Equipment: Microcontrollers: cont..

ARDUINO UNO REV 3

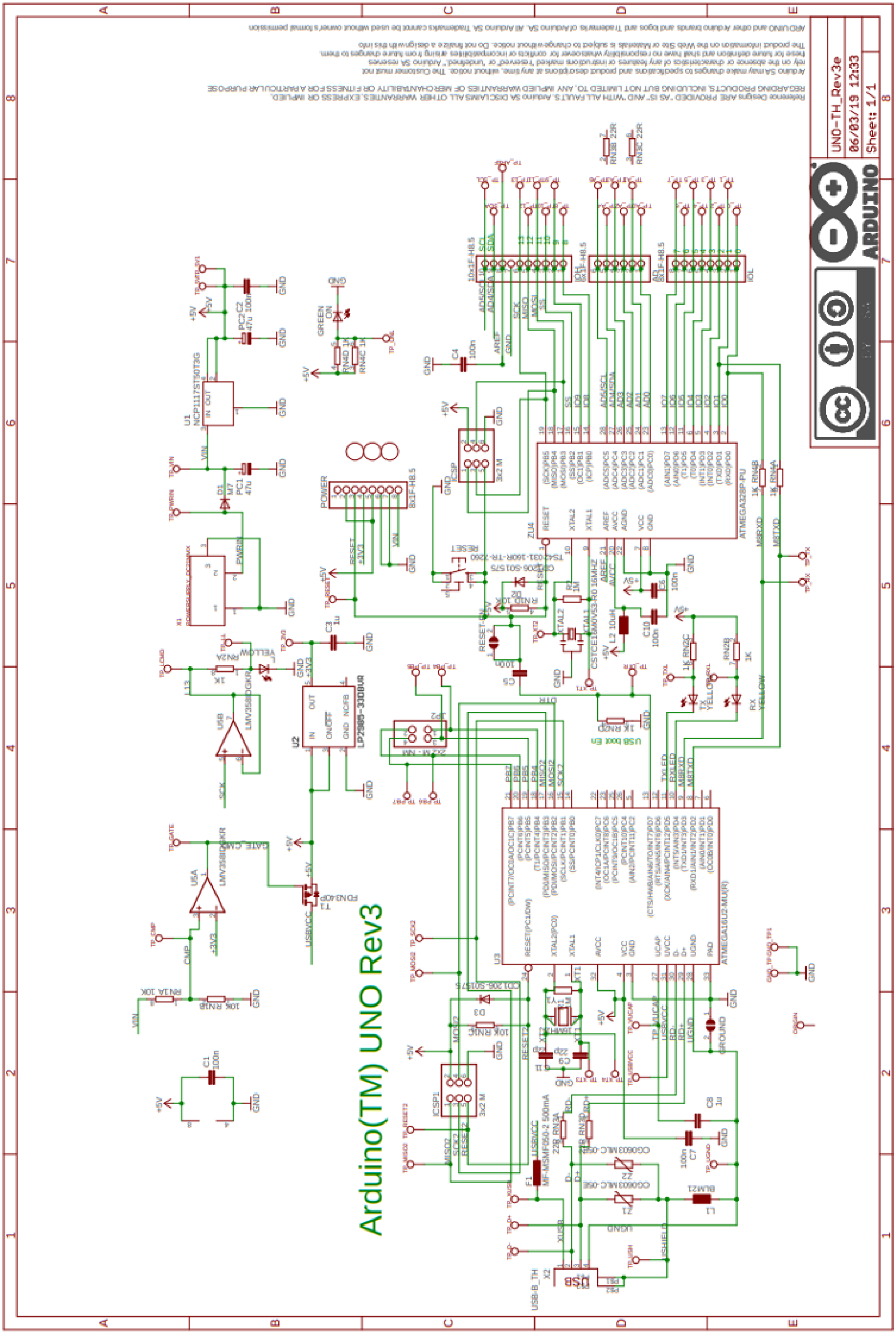
£19.95

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

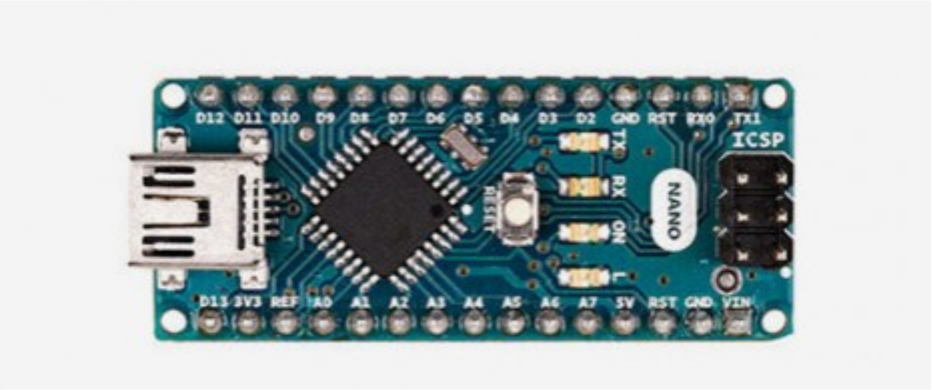
- Very well supported

- CAN support multiple input/output Analog & Digital.
- Easy to use IDE that also works with other boards
- NO built in BT
- cheap

Development Equipment: Microcontrollers: cont..

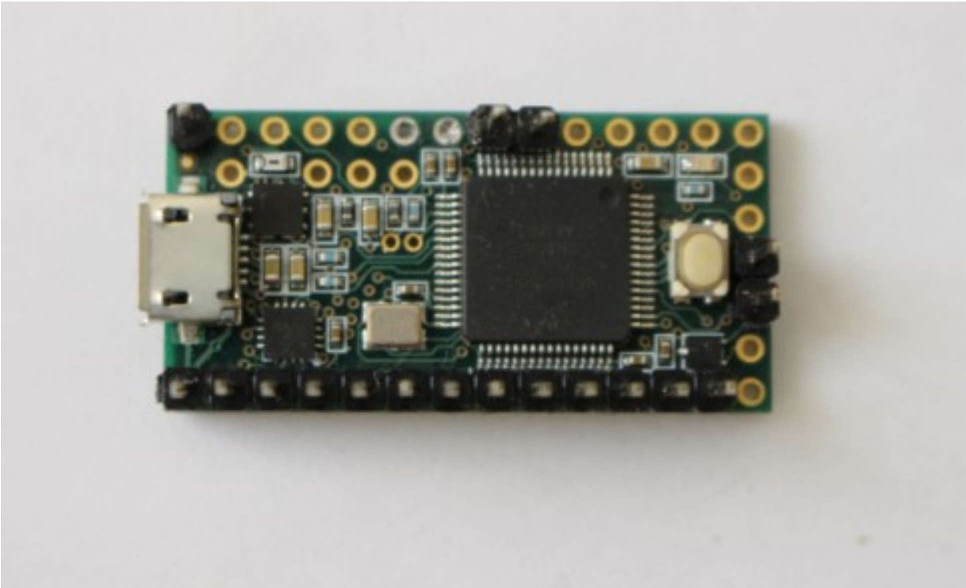


Development Equipment: Microcontrollers: cont..



Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

Development Equipment: Microcontrollers: cont..

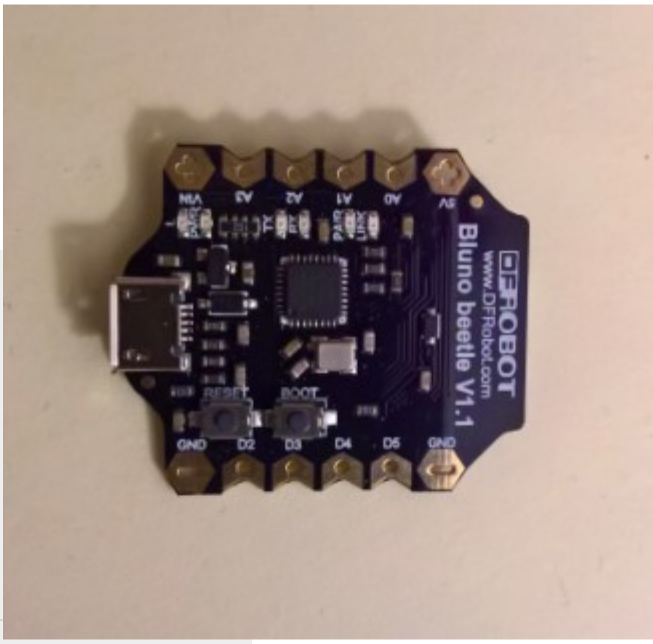


- 32 bit ARM Cortex-M4 72 MHz CPU (M4 = DSP extensions)
- 256K Flash Memory, 64K RAM, 2K EEPROM
- 21 High Resolution Analog Inputs (13 bits usable, 16 bit hardware)
- 1x 12 bit DAC Analog output
- 34 Digital I/O Pins (16 shared with analog)
- 12 PWM outputs
- 7 Timers for intervals/delays, separate from PWM
- USB with dedicated DMA memory transfers
- 3 UARTs (serial ports)
- SPI, 2x I2C, I2S, IR modulator
- I2S (for high quality audio interface)
- CAN Bus
- Real Time Clock (with user-added 32.768 crystal and battery)
- 16 general purpose DMA channels (separate from USB)
- Touch Sensor Inputs
- Micro-USB connector
- **Dimensions:** 1.4 x 0.7" (~35 x 18 mm)

Development Equipment: Microcontrollers: cont..

FEATURES

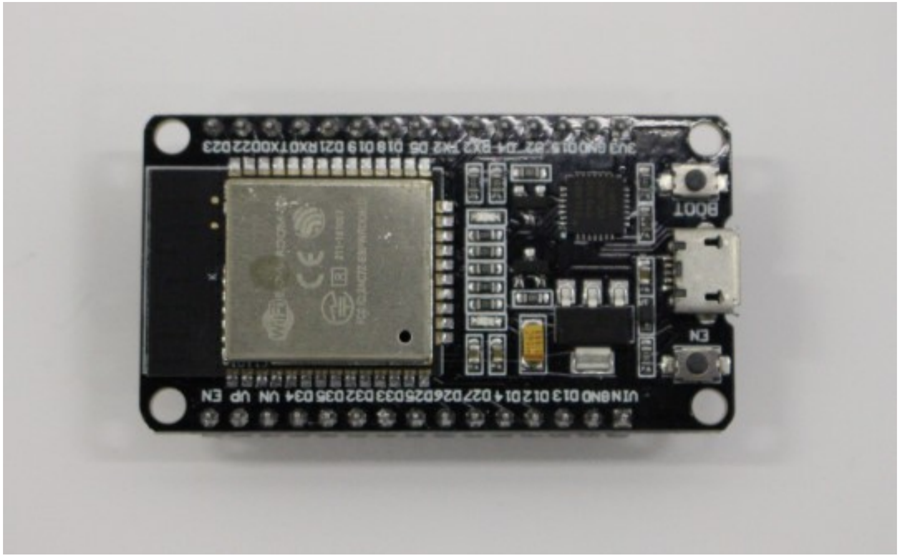
- ATmega328@16MHz
- Bluetooth Low Energy (BT 4.0)
- Micro USB port
- Super Compact Size
- Support Bluetooth HID and ibeacon
- Comaptible with all DFRobot Bluno Series
- Support Wireless Programming



SPECIFICATION

- Bluetooth Chip: CC2540
- Sensitivity: -93dBm
- Working Temperature：-10 ℃ ~ +85 ℃
- Maximun Distance：50m(1968.50")(Open field)
- Microcontroller: ATmega328
- Clock frequency: 16 MHz
- Operating voltage: 5V DC
- Input voltage: <8V (Vin < 8V)
- Digital Pin x4
- Analog Pin x4
- PWM Output x2
- UART interface x1
- I2C interface x1
- Micro USB interface x1
- Power port x2
- Size：28.8mm X 33.1mm(1.13" x 1.30")
- Weight：10g

Development Equipment: Microcontrollers: cont..



ESP32 Peripherals Features

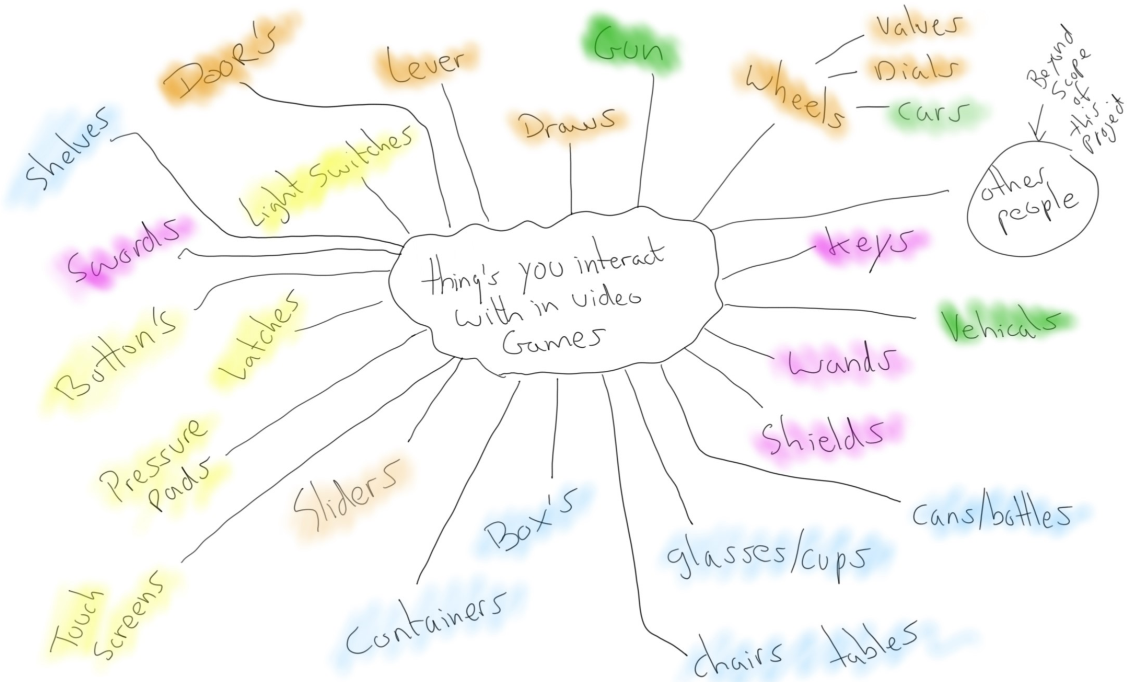
- 18 Analog-to-Digital Converter (ADC) channels
- 10 Capacitive sensing GPIOs
- 3 UART interfaces
- 3 SPI interfaces
- 2 I2C interfaces
- 16 PWM output channels
- 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces

Development Equipment: Design Environment

unity

- ◊ Easy to use interface
- ◊ Designed to be adaptive - editor is fully editable, can add custom menu’s, setting etc..
- ◊ Free to use unless your making over £1000,000
- ◊ Full support for all VR HMDs and the Leap Motion
- ◊ Excellent online support and pre-existing code repositories

Peripheral Design: Types of Interaction: cont...



Brainstorm of interactions likely in a video game environment.

Colours highlight groups of type of interaction:

Green: Objects that require tracking in 3d space and will also need some kind of input (trigger on a gun etc)

Blue: Moveable inanimate objects who will only need either an initial location setting or will require position tracking, but no secondary triggers etc..

Yellow: Static interactions with fixed location, but need a trigger/button/switch.

Purple: inanimate objects that can be pick-up and carried in the world, and may need to be able to interact with other objects through collision

Orange: Objects with fixed position but have a pivot point or some form of rotating interaction.

Can be split into three basic categories of interaction:

Pick-up/Put-down Push/Pull

Rotate/Turn

Peripheral Design: Types of Interaction: cont...

Can be split into three basic categories of interaction:

Pick-up/Put-down On/Off

Rotate/Pull/Push

Pick-up/Put Down : This is fundamental in nearly all forms of interaction (*may pose the greatest challenge with regards to tracking*) will require further research to implement.

Rotate/Push/Pull: For doors, levers, draws etc.. Will require a method of tracking rotation.

On/Off : For light switches, button triggers etc. Will need a way to track state.

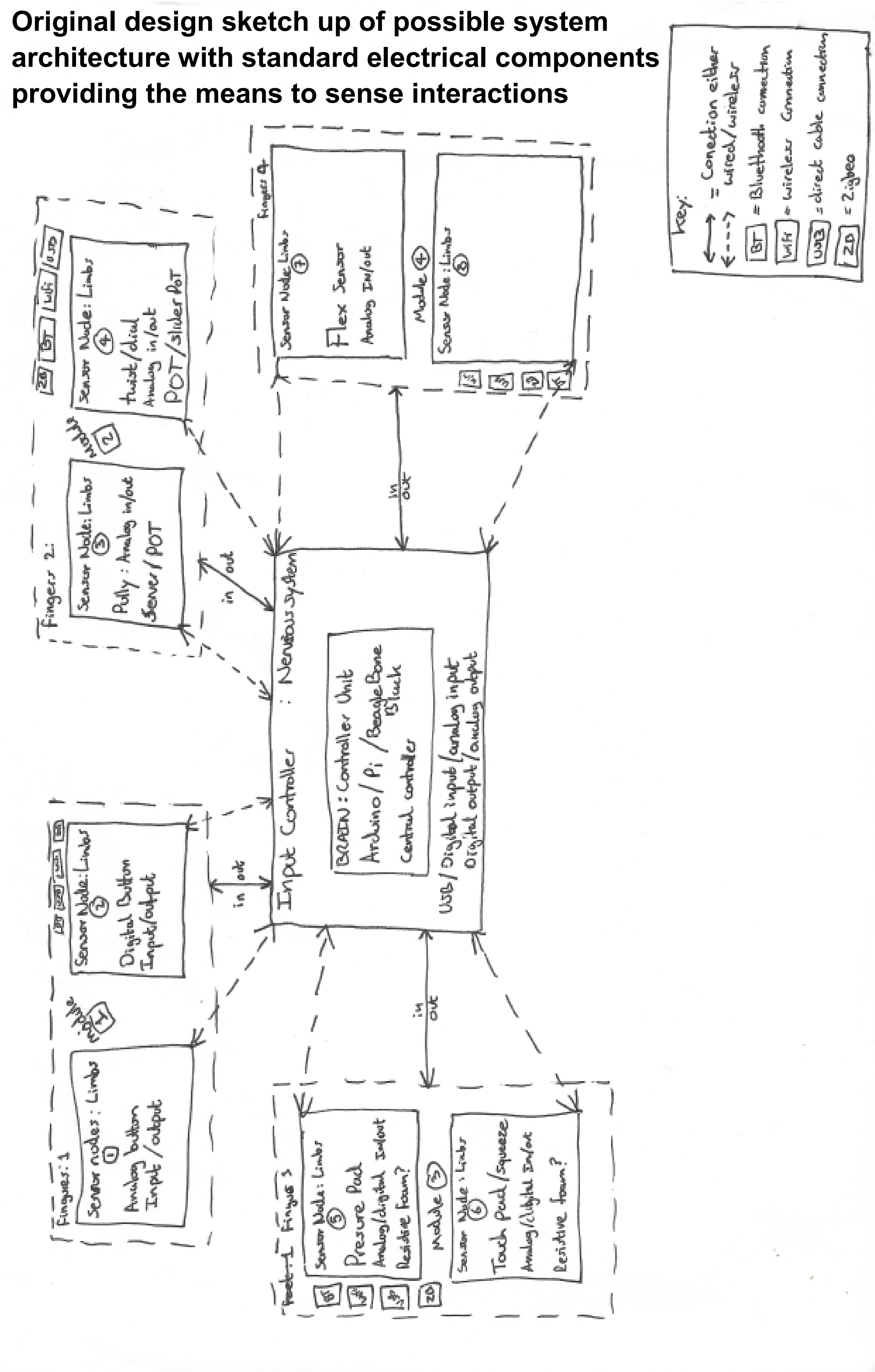
Can be separated into two groups for design purposes:

Static : based interactions are assumed to be, interactions with an object with a fixed location in 3D space, but with a pivot point or switch/button: Door. Lever , light switch , push button

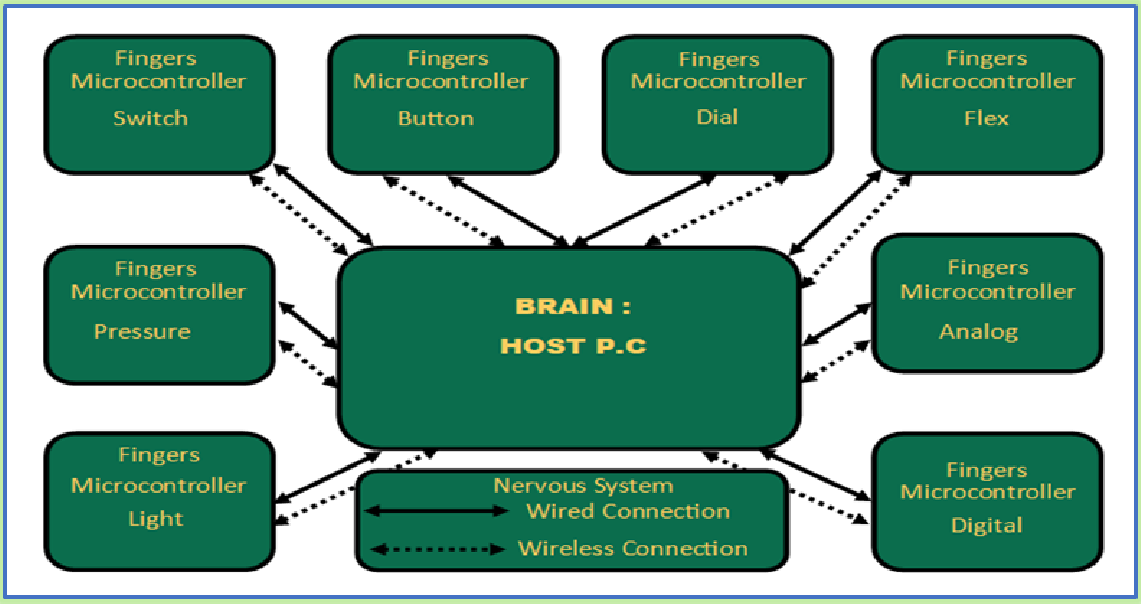
Mobile: Interactions with objects that can moved around freely in the virtual space, either inanimate only requiring position/location tracking (chair tables etc..), or animated which will also require 2nd or 3rd method of tracking, triggers on a gun etc..

Peripheral Design: Concept idea for sensing Interactions

Original design sketch up of possible system architecture with standard electrical components providing the means to sense interactions



Peripheral Design: Concept idea for sensing Interactions cont..



Refined design of initial concept idea.

Showing how sensors will detect interactions and send data to central controller.

Sensing interactions can possibly be achieved through the use of basic electrical components and specialised sensors.

Analog inputs: Potentiometer or Rotary Trackers could provide solution to static rotation, turning a dial or dimming a light , or pressure sensing

Digital Inputs: Switches/Buttons , tilt sensors etc. used to as they are for buttons and switches etc…

Specialist Sensors: Accelerometers, gyroscopes could be used to track rotation of moving objects

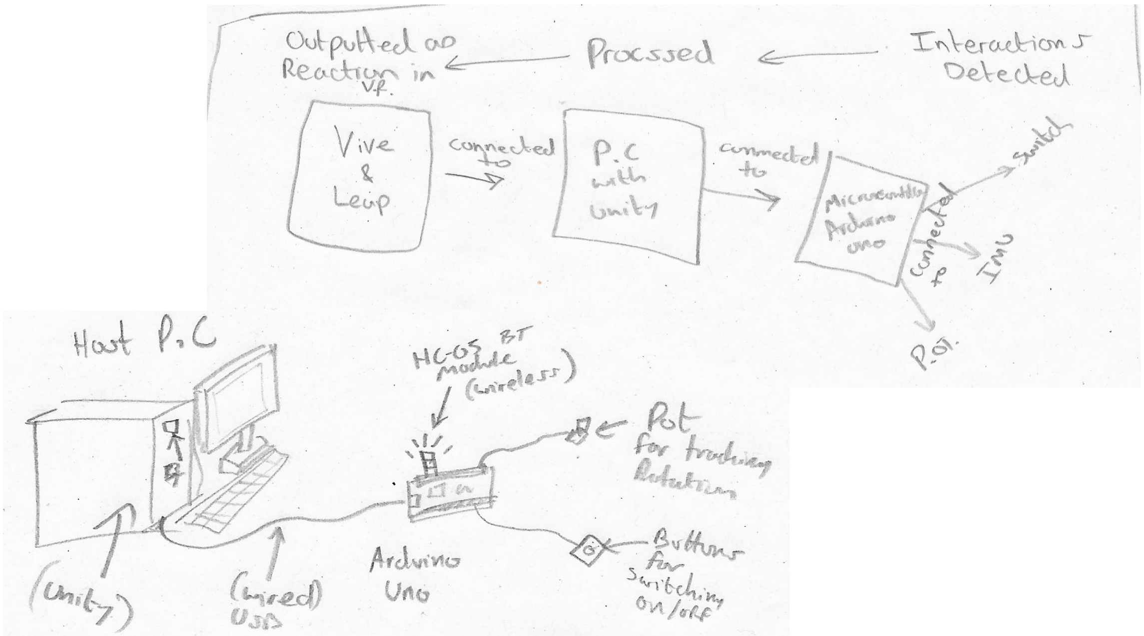
Wireless Controllers: Bluetooth modules or Bluetooth low ener-gy modules.

Wired Control: through ethernet or U.S.B. cables

Peripheral control: Achieved through installing a microcontroller in the device to handle incoming sensor data

Initial Concept System Design:

From the design work so far an concept high level system



Concept:

Peripheral devices placed into the real-world act as detectors of interaction, interaction detected and transmitted to P.C to design environment, were they are turned into a digital reaction.

Initial components :

Design Environment: Unity 3d

Peripheral Controller: Arduino Uno Rev3

Wired Connection: Micro USB cable

Wireless Connection: Bluetooth HC-05

Trigger: Tactile Switch

Rotation (static): Rotary Potentiometer

Rotation (moving) : Inertial Measurement Unit (accelerometer/Gyro)

Feasibility Testing of Base Components:

Next step in design process was to test out the components to see if the selected parts would be sufficient , and identify any issue that need addressing before design continues.

Experiment One:

Concept/Aims/ Circuit Set-up:

1 from Unity to Arduino to test Serial Com's gain understanding of how it works & baud Rate effect

Limits One way

~~[illegible]~~

9600	Time	19200	Time	115200	time
1	1.31		1.14		0.78
2	1.28		1.07		0.77
3	1.23		1.13		0.76
4	1.21		1.21		0.72
5	1.25		0.87		0.84
6	1.16		0.94		0.77
7	1.19		0.52		0.75
8	1.21		1.02		0.80
9	1.01		0.97		0.70
10	1.19		0.94		0.92

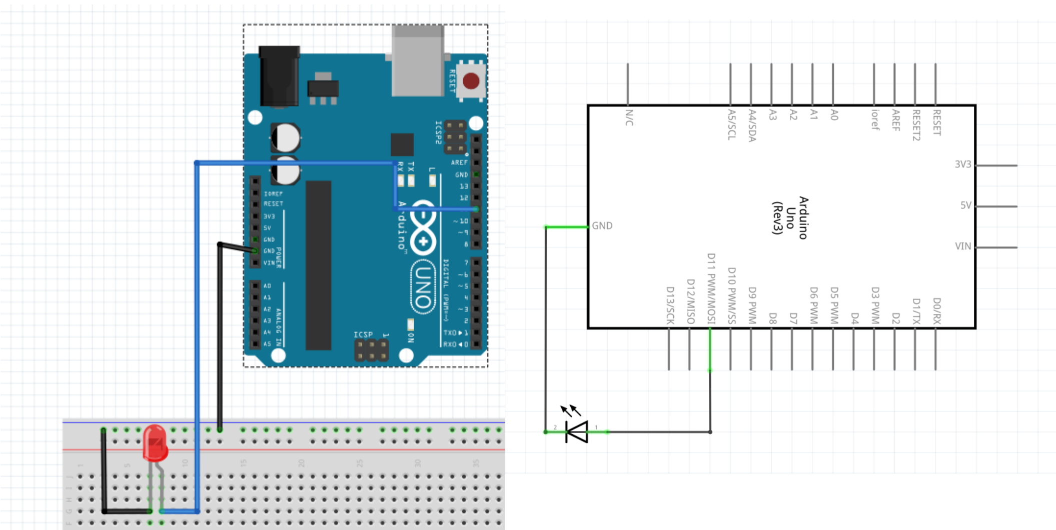
outcome

established simple com's but one way and measured effect of varying baud rate settled on 115200, as this is max recommended Speed for Bluetooth Comm's.

Aimb: To establish how to connect to Unity via a serial port and control an L.E.D, also to investigate correct baud rate in Bps to operate on.

L.E.D : connected to Pin 2 of Arduino and GND

Set up as below.



Feasibility Testing of Base Components: Cont...

Scripts and Method Outline:

ExperimentOne | Arduino 1.8.9

File Edit Sketch Tools Help

ExperimentOne

```
//Experiment One:
//Simple receive activation signal from Unity/Serial Port to switch on L.E.D .

#define LED_Pin 2

void setup()
{
  Serial.begin(115200);
  pinMode(LED_Pin, OUTPUT);
}

void loop()
{


  if(Serial.available() > 0)
  {
    String incomming = Serial.readString();
    if(incomming == "High")
    {
      digitalWrite(LED_Pin, HIGH);
      delay(2000);
      digitalWrite(LED_Pin, LOW);
    }

  }

}
```

Scripts created in Arduino IDE in C, and Unity (Visual Studio) C#.

Arduino: waits for received signal from Unity , when received turns led

Feasibility Testing of Base Components: Cont...

Scripts and Method Outline:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO.Ports;
using System.Linq;

public class ExperimentOne : MonoBehaviou
{

    private float timerStart;
    private float timeElapsed;
    private bool timerStarted = false;



    SerialPort Arduino;
    string sendSignal = "High";
    public string ComPort = "";

    // Start is called before the first f
    void Start()
    {
        Arduino = new SerialPort(ComPort,
        Arduino.Open();
        if (Arduino.IsOpen)
        {
            Arduino.Write("Unity Online");
            Debug.Log("Serial Connection Active");
        }


    }
```

```
    void Update()
    {

        if (Input.GetKeyDown(KeyCode.S))
        {

            Arduino.Write(sendSignal);
            Debug.Log("SENT MESSAGE");
            timerStart = Time.time;
            timerStarted = true;

        }
        if (timerStarted && Input.GetKeyDown(KeyCode.T))
        {
            timeElapsed = Time.time - timerStart;
            Debug.Log(timeElapsed);


        }

    }
}
```

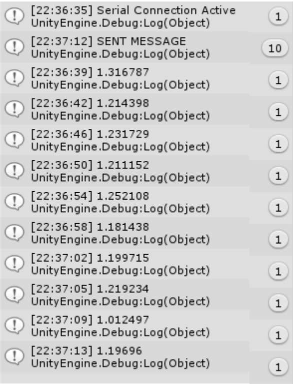
Unity: sends a signal when the S key is pressed, starts a timer, when T key pressed, stops timer and records time, T pressed each Time light observed in ON state.

Method:

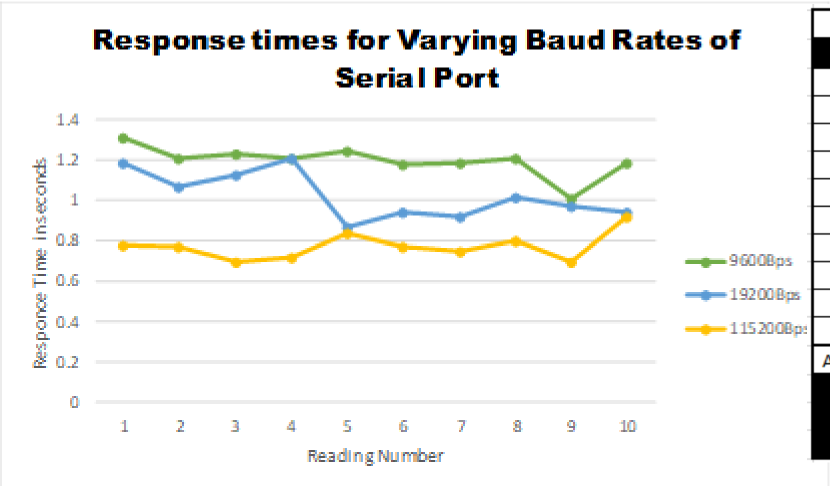
Ten recordings taken for three target baud rates, slow (9600), mid (19200) , and fast (115200) (all in Bytes per second) , assessed based on how much lag in response, as lag in VR had very bad effect on experience.

Feasibility Testing of Base Components: Cont…

Results and Findings:



RAW out put data from unity console, shows aprrox lag in signal for three baud rates, left to right 9600Bps, 19200Bps and 115200Bps



Results of Varying Baud Rate			
	1	2	3
1	1.31	1.19	0.78
2	1.21	1.07	0.77
3	1.23	1.13	0.7
4	1.21	1.21	0.72
5	1.25	0.87	0.84
6	1.18	0.94	0.77
7	1.19	0.92	0.75
8	1.21	1.02	0.8
9	1.01	0.97	0.7
10	1.19	0.94	0.92
Average	1.199	1.026	0.775
	1= 9600Bps		
	2=19200Bps		
	3=115200Bps		

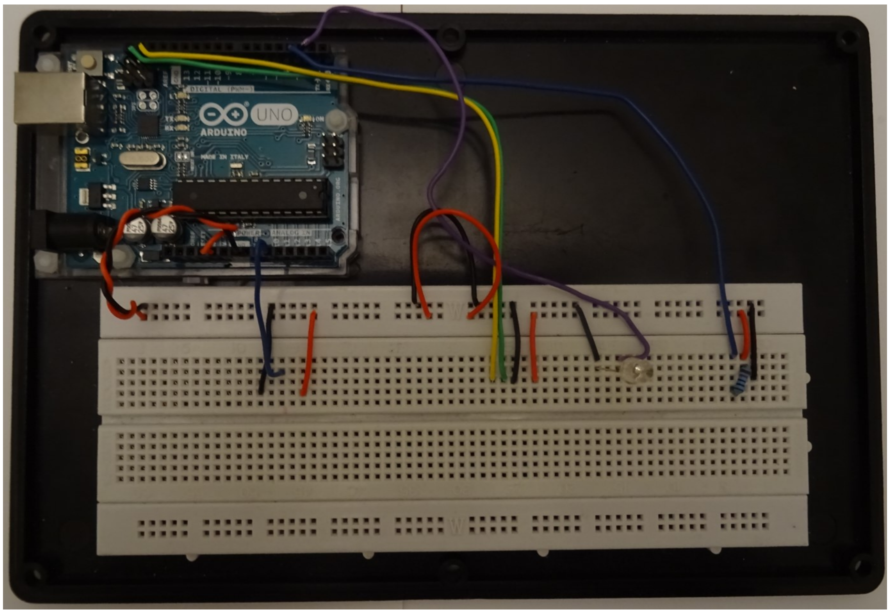
Objective: To establish communication between the Arduino Uno and Unity , when key pressed in Unity turns on light in real-world attached to Arduino. When Light Visibly On second key press stops timer and records time delay, only approx. as have to account for my own reaction time, but sufficient to demonstrate the effect of the baud rate.

Out Comes:

- Successfully established connection to unity, but coms limited to one way from unity out, needs to be the other way around or two way.
- 100% success it L.E.D response
- Identified that the optimum speed for operation is 115200Bps, higher speeds available but may cause transfer issue when using the Bluetooth Modules, slower speeds cause too much lag.

Feasibility Testing of Base Components: Cont...

Pictures of Experiment :



Feasibility Testing of Base Components: Cont..

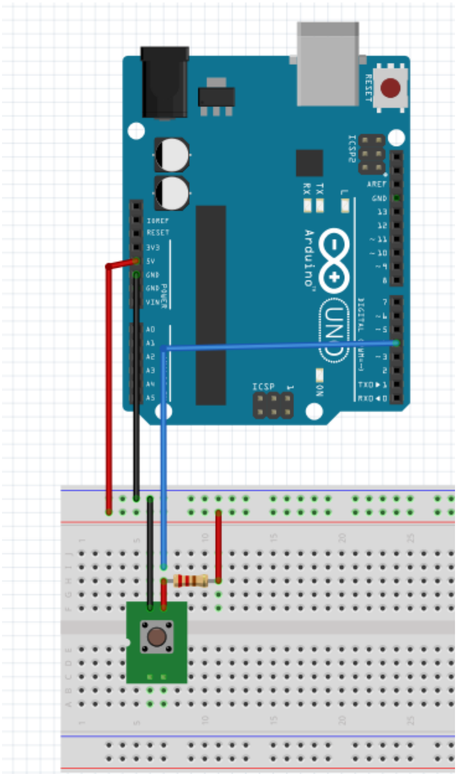
Experiment Two:

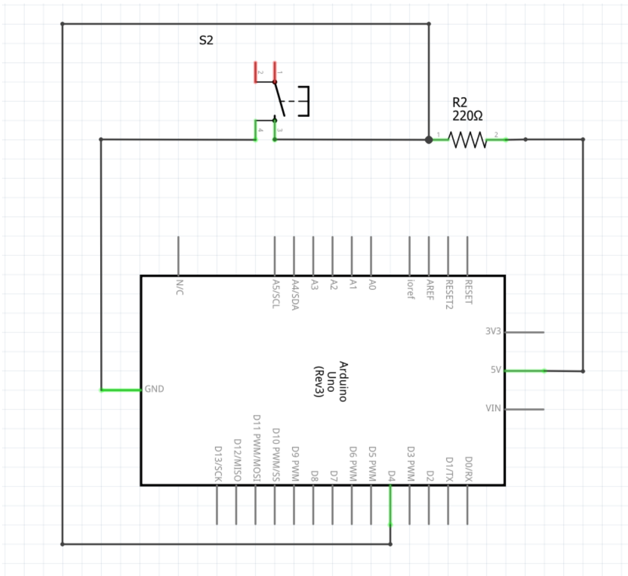
Concept/Aims/ Circuit Set-up:

2 Arduino to unity
Send input from button to Unity & set an object to Active ~~+ Record~~
Pressed button 30 times only dropped the signal once / no visable lag
tacktile switches will be suitable for triggers.

Aim: To establish comms from Arduino to Unity, and then send data via USB from a interaction with a tactile switch , used to control the active state of an object in the Unity environment.

Circuit set up: switch set to active on logic Low, connected to GND and +5v via a pull-up 4.4KΩ resistor, with the active pin connected to Arduino Digital Pin 3. When switch pressed puuls signal Low and transmits data to Unity





Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

ExperimentTwo | Arduino 1.8.9

File Edit Sketch Tools Help

ExperimentTwo

```
//Experiment 2:
//Send data from arduino to unity via usb, received singal toggles set active property of unity game object.



unsigned int count = 0;
int buttonPin = 3;
int buttonState = 0;
bool setActive = false;

void setup()
{
  Serial.begin(115200);
  pinMode(buttonPin, INPUT);
}

void loop()
{

  buttonState = digitalRead(buttonPin);
  if(buttonState == LOW && setActive == false){
  Serial.print("H");
  Serial.println(count);
  setActive = true;
  count++;
  delay(1000);
  }
  else if(buttonState == LOW && setActive == true){

    Serial.print("L");
    Serial.println(count);
    setActive = false;
    count++;
    delay(1000);
  }
}
```

Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

```
void showNewData()  //Method to trigger events when new data receieved
{
    if (newData == true)
    {
        if (receivedDataLine.Contains("H"))
        {
            target.SetActive(true);
            Debug.Log("Target object Enabled " + receivedDataLine);
        }
        if (receivedDataLine.Contains("L"))
        {
            target.SetActive(false);
            Debug.Log("Target object Disabled " + receivedDataLine);
        }

        newData = false;
    }
}
```

```
void receivedString() // method to read incoming serial data
{
    try
    {
        receivedDataLine = Arduino.ReadLine();
        Debug.Log("working 2");
        newData = true;
    }
    catch(TimeoutException e)
    {

    }
```

```
public void ControlSerialPort() // method t
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        Arduino.Close();
        Debug.Log("Arduino Disconnected");
    }
}
```

Feasibility Testing of Base Components: Cont..

Results and Findings:

```
[00:21:06] Target object Enabled H0
UnityEngine.Debug:Log(Object)
[00:21:09] Target object Disabled L1
UnityEngine.Debug:Log(Object)
[00:21:12] Target object Enabled H2
UnityEngine.Debug:Log(Object)
[00:21:14] Target object Disabled L3
UnityEngine.Debug:Log(Object)
[00:21:16] Target object Enabled H4
UnityEngine.Debug:Log(Object)
[00:21:17] Target object Disabled L5
UnityEngine.Debug:Log(Object)
[00:21:18] Target object Enabled H6
UnityEngine.Debug:Log(Object)
[00:21:19] Target object Disabled L7
UnityEngine.Debug:Log(Object)
[00:21:21] Target object Enabled H8
UnityEngine.Debug:Log(Object)
[00:21:22] Target object Disabled L9
UnityEngine.Debug:Log(Object)
[00:21:23] Target object Enabled H10
UnityEngine.Debug:Log(Object)
```

```
[00:21:24] Target object Disabled L11
UnityEngine.Debug:Log(Object)
[00:21:25] Target object Enabled H12
UnityEngine.Debug:Log(Object)
[00:21:26] Target object Disabled L13
UnityEngine.Debug:Log(Object)
[00:21:27] Target object Enabled H14
UnityEngine.Debug:Log(Object)
[00:21:28] Target object Disabled L15
UnityEngine.Debug:Log(Object)
[00:21:29] Target object Enabled H16
UnityEngine.Debug:Log(Object)
[00:21:30] Target object Disabled L17
UnityEngine.Debug:Log(Object)
[00:21:31] Target object Enabled H18
UnityEngine.Debug:Log(Object)
[00:21:32] Target object Disabled L19
UnityEngine.Debug:Log(Object)
[00:21:33] Target object Enabled H20
UnityEngine.Debug:Log(Object)
[00:21:34] Target object Disabled L21
UnityEngine.Debug:Log(Object)
```

```
[00:21:35] Target object Enabled H22
UnityEngine.Debug:Log(Object)
[00:21:37] Target object Enabled H24
UnityEngine.Debug:Log(Object)
[00:21:40] Target object Enabled H26
UnityEngine.Debug:Log(Object)
[00:21:42] Target object Disabled L27
UnityEngine.Debug:Log(Object)
[00:21:43] Target object Enabled H28
UnityEngine.Debug:Log(Object)
[00:21:44] Target object Disabled L29
UnityEngine.Debug:Log(Object)
[00:21:45] Target object Enabled H30
UnityEngine.Debug:Log(Object)
```

Of the 30 signals sent from the Arduino to Unity only two were dropped, L23 and L24,

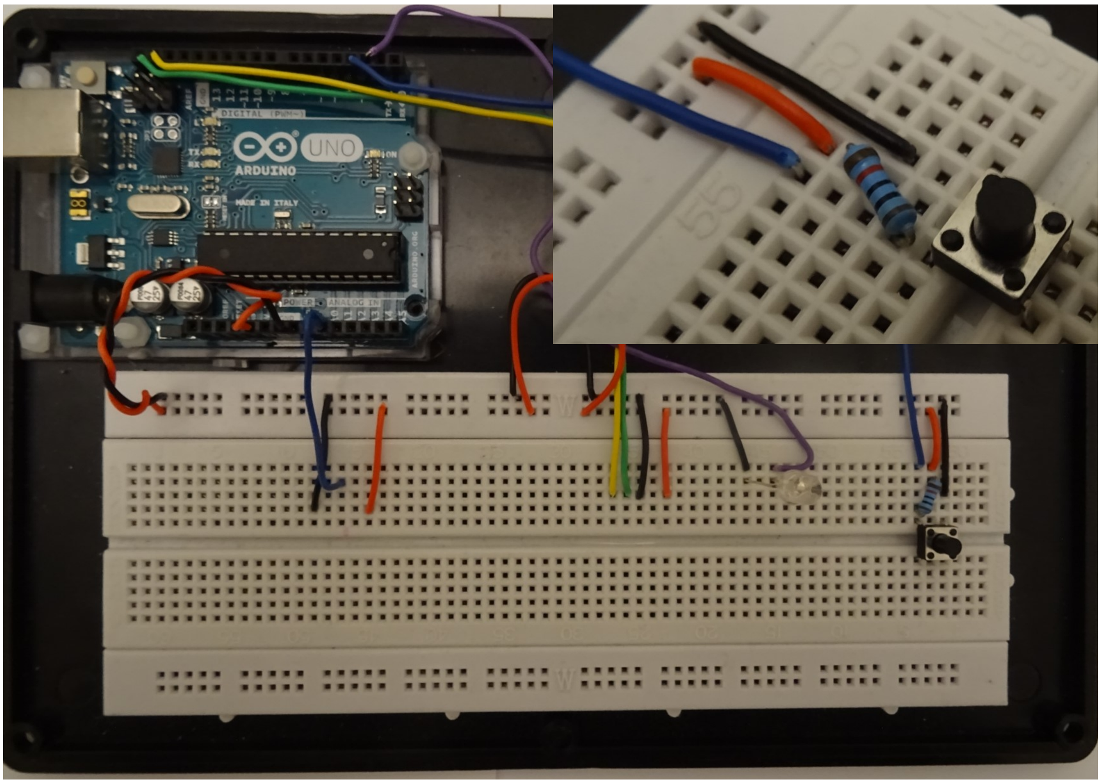
Confirming that this approach to sending data via a tactile switch would be sufficient for the purposes of this project

Objectives: To establish data transfer from the Arduino to Unity, once connected to send a signal from the real-world into the digital environment via USB, the resulting received signal is used to set the active state from true to false, demoing the desired outcome of interaction and response.

Observations: The signal was received successfully on 93% of the times the button was pressed, signal dropped twice, this is believed to be due to pressing the switch in too quick succession, as there is a 1 second delay built into the code, this will need to be considered when taking the design further and either removed or reduced to eliminate the issue.

Feasibility Testing of Base Components: Cont..

Pictures of Experiment :



Feasibility Testing of Base Components: Cont..

Experiment Three:

Concept/Aims/ Circuit Set-up:

③ pot to Unity

Aim work out how ~~best~~
to send float/int's ~~vat~~ via with value
serial port & test responsiveness
of the pot.

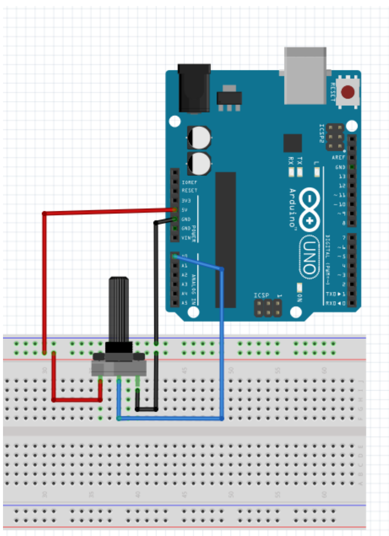
** greater than 256, solution send as
~~individ~~ string & convert to
float in unity as fast
and efficient. low computational
time on microcontroller.

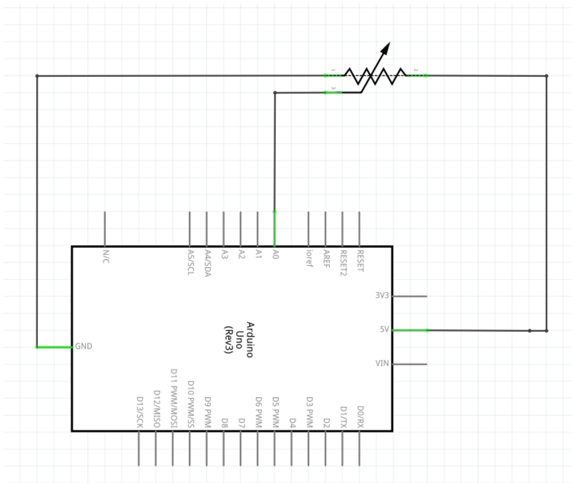
By transmitting this way get full
range of voltage i flagged " error that
string in wrong format *

to unity

Aim:

Circuit set up: 10KΩ rotatory variable resistor connected to GND and +5v of Arduino, slider connected to analogue pin 0 (A0) of the Arduino, and connected to Unity Via USB.





Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

ExperimentThree

```
//Experiment 3:
//Send potentiometer data from arduino to unity via usb, received singal adjusts

int POT_Pin = A0;

float POT_Value;
unsigned int count = 0;
int buttonPin = 3;
int buttonState = 0;
bool setActive = false;

void setup()
{
  Serial.begin(115200);
  pinMode(buttonPin, INPUT);
}

void loop()
{

  POT_Value = analogRead(POT_Pin);
  Serial.println(POT_Value);
  delay(100);

}
```

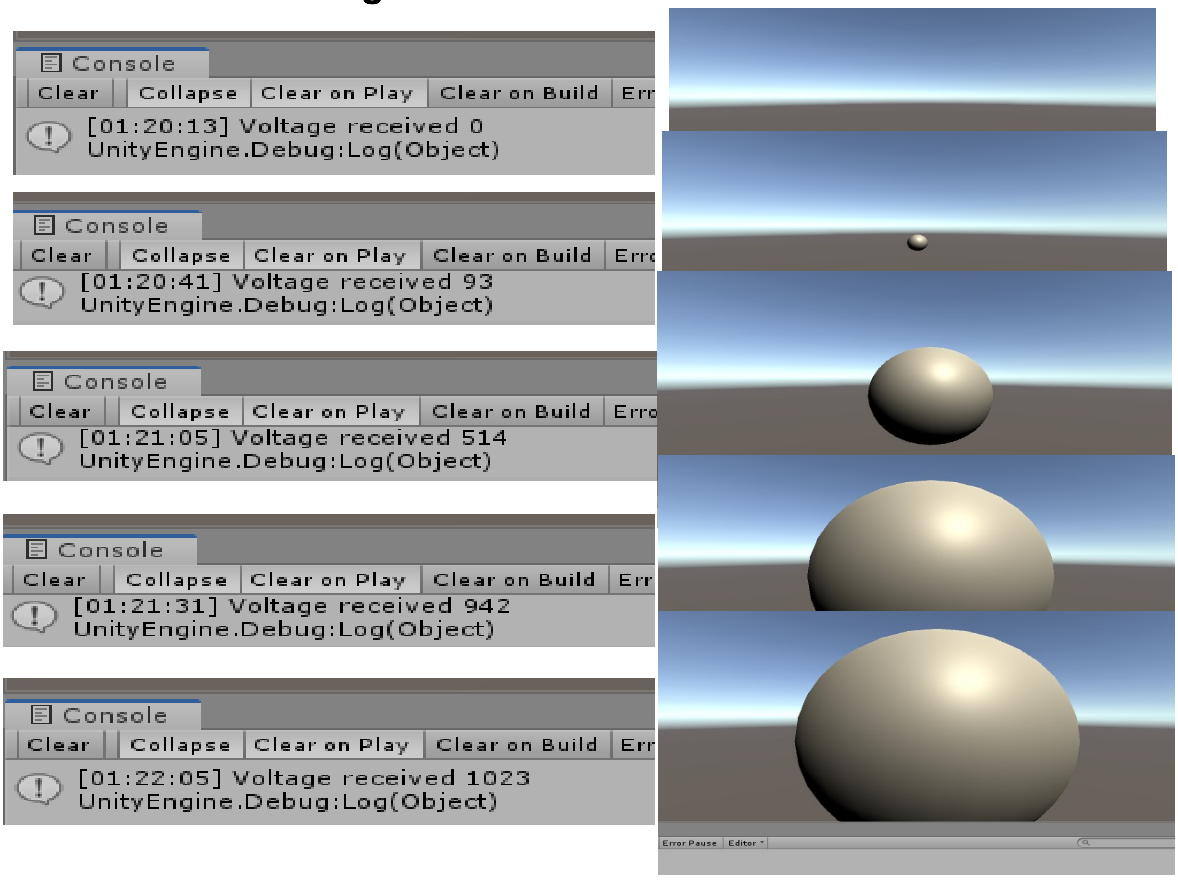
Use Arduino analogue input to read POT data and send and send as ASCI characters via the serial port.

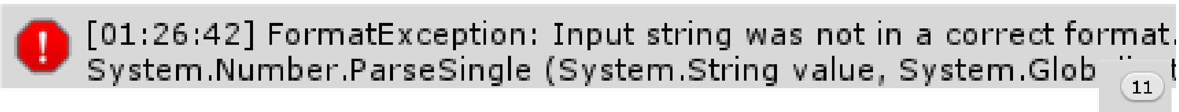
Conversion of string to Float so data can be used to control object local scale.

```
void showNewData()  //Method to trigger events when new data receieved
{
    if (newData == true)
    {
        Voltage = float.Parse(receivedDataLine);
        target.transform.localScale = Vector3.one * ( Voltage/100.0f);
        Debug.Log("Voltage received " + Voltage);
        newData = false;
    }
}
```

Feasibility Testing of Base Components: Cont..

Results and Findings:





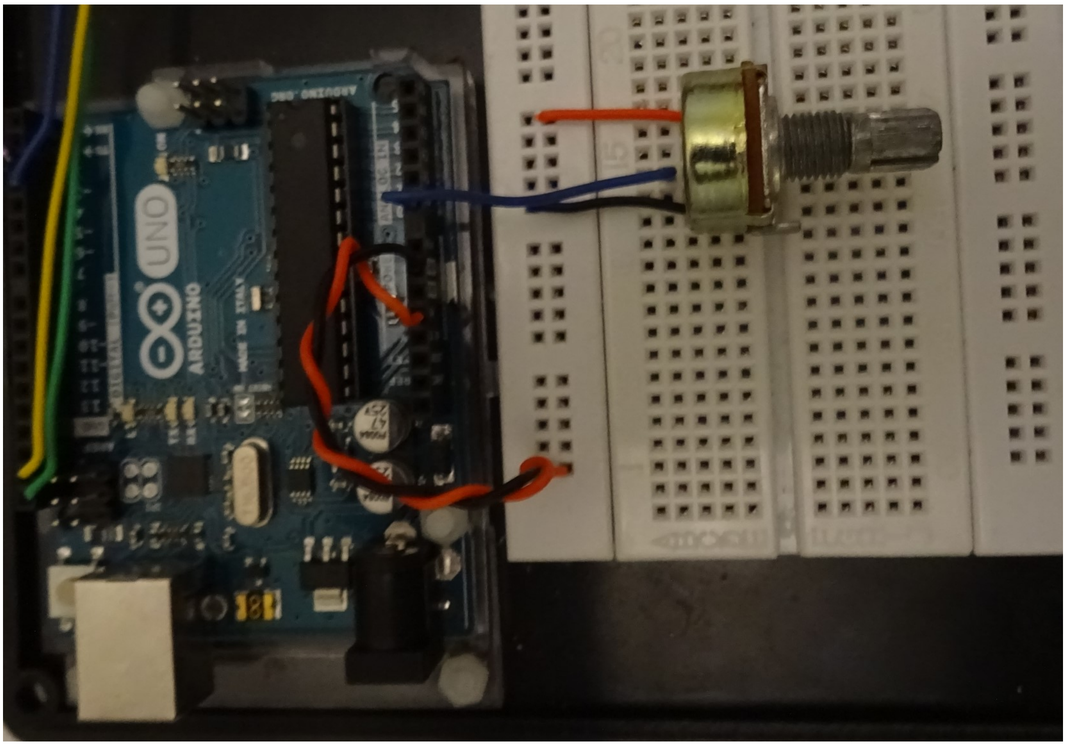
Objectives: To establish a method of transmitting variable data as opposed to a single fixed value, to be achieved by connecting a POT to the Arduino and sending the value to Unity to control the radius of a sphere. Also to check the reliability of the POT as a sensor for the tool kit.

Observations:

- Value needed to be transmitted as chars , not as a number , as only transmit single bytes so number limited to 0-256, by sending as ASCII characters this was resolved. The subsequent String received in Unity was then converted to a Float and applied to the size of the sphere.
- Noted that there is a dead zone at the start of rotation that will need to accounted for if accurate tracking is to be achieved
- Some miss reads as received data not in correct format, not an issue here but maybe later, solution to be investigated later.

Feasibility Testing of Base Components: Cont..

Pictures of Experiment :



Feasibility Testing of Base Components: Cont..

Experiment Three:

Concept/Aims/ Circuit Set-up:

(4) Send IMU data to unity
to test multiple variables being sent
& potential of IMU for stable
object Rotation tracking.

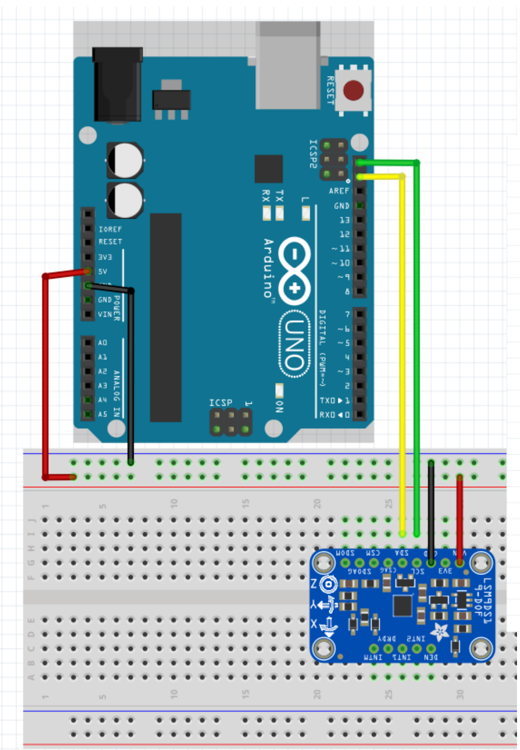
x, y, z values used to set orientation
of Game object in Unity.

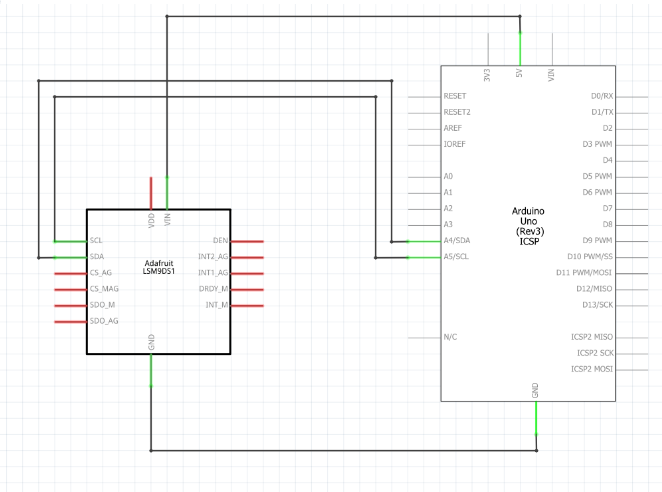
* IMU output not stable will need to
be addressed further

* String in wrong format error again caused
System to hang.

Aim: To establish a system for transmitting multiple variables in one go, to unity, converting them to usable values and applying the results to a game object . Also to test stability and suitability of the selected IMU as a device to track rotation in 3D space.

Circuit set up: Adafruit 9DOF IMU, connected to +5v, and GND. SDA of IMU to SDA of Arduino, SCL of IMU to SCL of Arduino. Arduino connected to P.C via USB.





Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

ExperimentFour

```
#include <Adafruit_Sensor.h>

#include
<Adafruit_LSM303_U.h>

#include
<Adafruit_L3GD20_U.h>
```

Built in libraries for Arduino IDE that make implementing the use of IMU very straight forward

```
void loop(void)
{
  Serial.print("<");
  xAxis();
  Serial.print(">");
  Serial.print("<");
  yAxis();
  Serial.print(">");
  Serial.print("<");
  zAxis();
  Serial.print(">");

  delay(100);
}

void xAxis(){
  sensors_event_t accel_event;
  sensors_vec_t   orientation;
  accel.getEvent(&accel_event);
  if (dof.accelGetOrientation(&accel_event, &orientati
  {
    Serial.print("X");
    Serial.print(orientation.roll);
  }
}
```

Outputs the orientation data, preceded by the corresponding axis label

```
void yAxis(){
 sensors_event_t mag_event;
 sensors_vec_t   orientation;
mag.getEvent(&mag_event);
  if (dof.magGetOrientation(SENSOR_AXIS_Z, &mag_event, &orientation))
  {
    Serial.print("Y");
    Serial.print(orientation.heading);
  }
}
void zAxis(){
  sensors_event_t accel_event;
  sensors_vec_t   orientation;
  accel.getEvent(&accel_event);
  if (dof.accelGetOrientation(&accel_event, &orientation))
  {
    Serial.print("Z");
    Serial.print(orientation.pitch);
  }
}
```

Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

```
void RecvWithStartEndMarkers() //method to split m
{
    bool recvInProgress = false;
    byte ndx = 0;
    char startMarker = '<';
    char endMarker = '>';
    char rc;

    while (newData == false)
    {

        rc = receivedString();

        if (recvInProgress == true)
        {
            if (rc != error)
            {
                if (rc != endMarker)
                {
                    recivedChars[ndx] = rc;
                    ndx++;
                    if (ndx >= numChars)
                    {
                        ndx = numChars - 1;
                    }
                }
                else
                {
                    recivedChars[ndx] = '\0';
                    recvInProgress = false;
                    ndx = 0;
                    newData = true;

                }

            }
        }

        else if (rc == startMarker)
        {
            recvInProgress = true;
        }
    }
}
```

Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

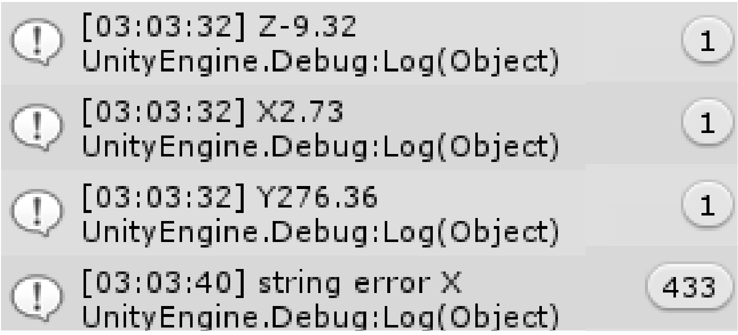
```
public Char receivedString() // method to read in
{

    char receviedData ;

    try
    {
        receviedData = (char)Arduino.ReadChar();
        Arduino.BaseStream.Flush();
        return receviedData;

    }
    catch (TimeoutException e)
    {
        return error;
    }



}
```

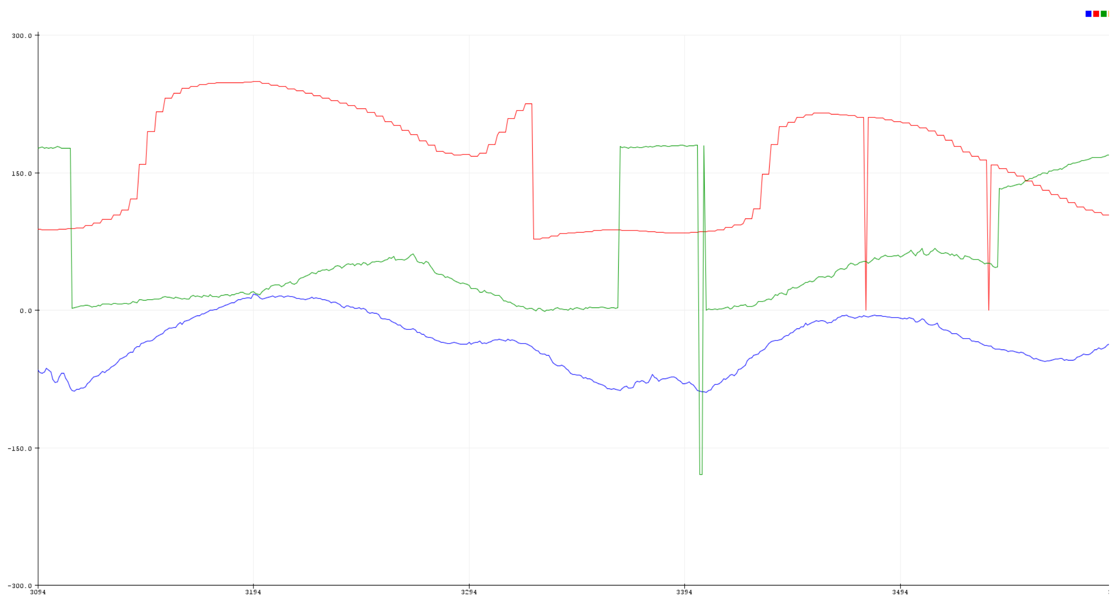
```
void showNewData()  //Method to trigger events when new data receieved
{

    char[] charToTrim = { 'X', 'Y', 'Z', '<', '>' };

    if (newData == true)
    {
        string display = new string(recivedChars);
        if (display.StartsWith("X"))
        {
            tmpX = display;
            tmpX = tmpX.Trim(charToTrim);
            try
            {
                xValue = float.Parse(tmpX);
            }
            catch
            {

                Debug.Log("string error X");

            }
            xValue = float.Parse(tmpX);
```

```
target.transform.rotation = Quaternion.Euler(xValue,yValue,zValue);
```

Feasibility Testing of Base Components: Cont..

Results and Findings:



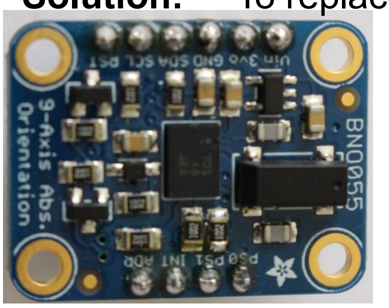


Objectives: By connecting an IMU to Arduino and to Unity , test the ability of the system to transfer multiple variables, and to test the efficacy of the IMUs signal stability.

Observations:

- This IMU not going to be suitable as the output signal is not consistent enough, signal drops and constant variations in value resulted in jittering in the Unity environment.
- Signal drops caused the system to freeze as Unity could not convert the data into the desired format

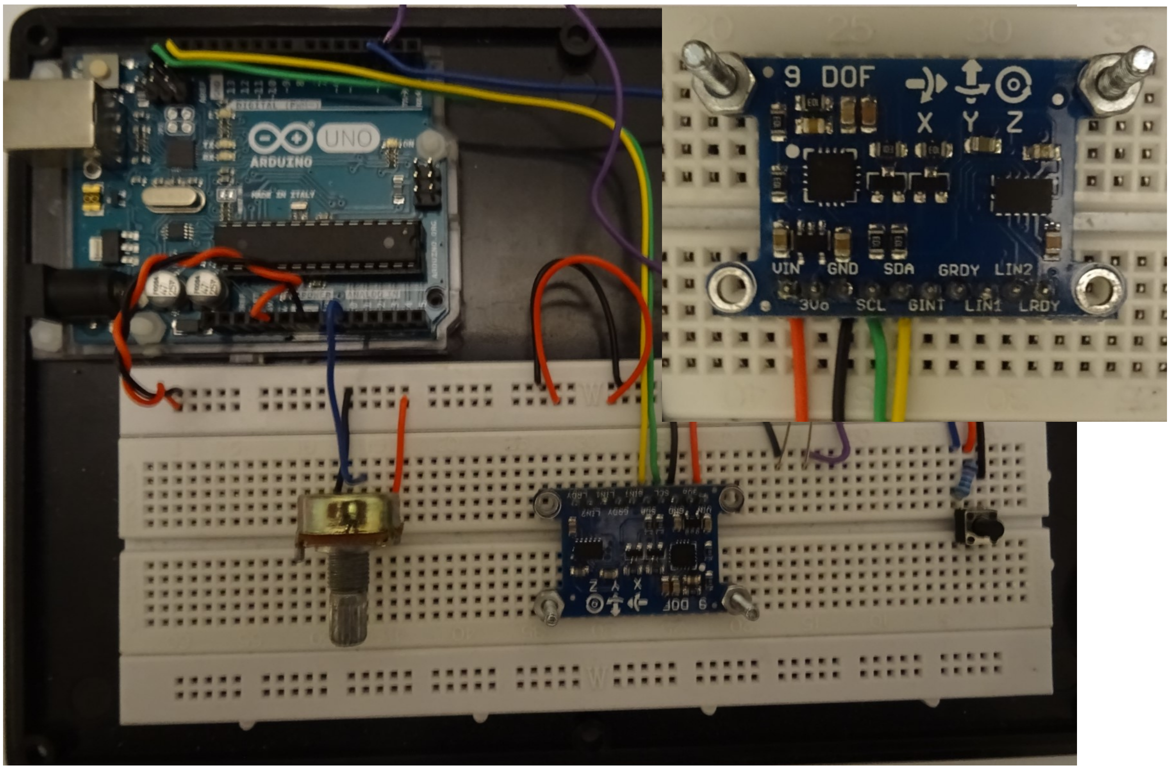
Solution: To replace the current IMU with a more expensive and



reliable model, component selected is the BNO055 absolute orientation unit, as this model has an onboard microcontroller that handles smoothing out the output signal. RRP £30.00

Feasibility Testing of Base Components: Cont..

Pictures of Experiment :



Feasibility Testing of Base Components: Cont..

Experiment Three:

(5) Aim: to establish two-ways comm's with Arduino's, using two HC-05 BT Modules

- to get understanding of how they work & are set up

- connect the Rx of HC/05 to Arduino TX & Rx Arduino to ~~R~~Tx of HC-05

Reverse this for AT Mode

work out if there are any issues with using these cheap BT units

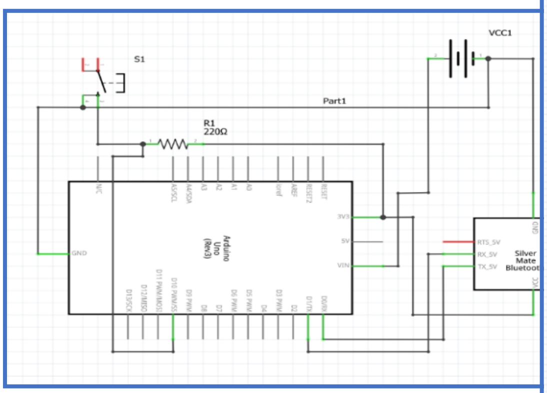
HC-05

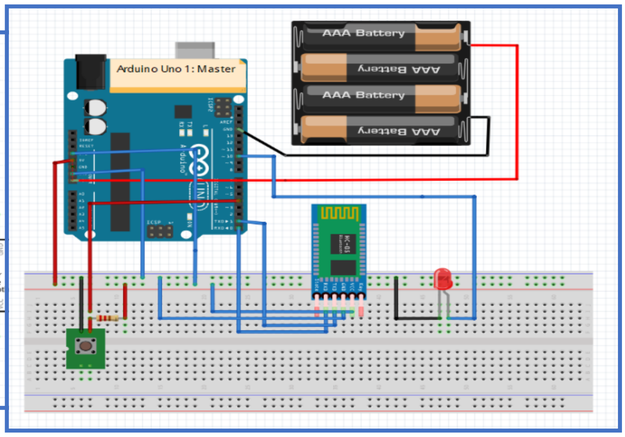
Rx/Tx

Aim: To set-up two Arduino Uno with HC-05 Bluetooth to establish two-way comms, and identify any limitations to this approach

Circuit set up (MASTER):

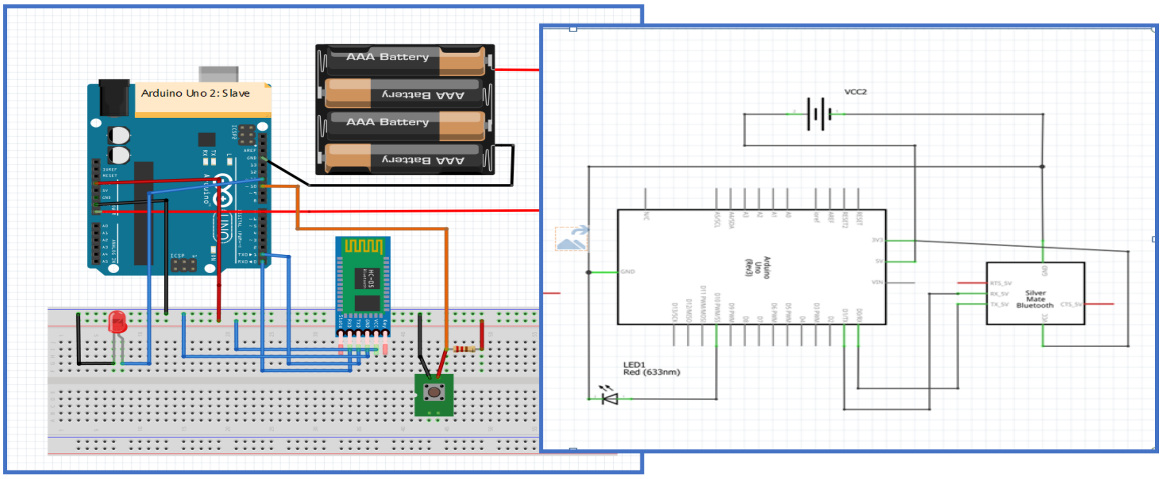
Tactile switch connected to Arduino as in exp2, and L..E.D set up as in EXp1, with a HC-05 BT module connected to the RX/TX pins of the Arduino, RX(BT) to TX(UNO) and TX(BT) to RX(UNO). Need to be disconnected when uploading software from IDE as shares the port with USB. Also when configuring in AT mode reverse the connections. HC-05 set to MASTER Mode and bound to Slave





Feasibility Testing of Base Components: Cont..

Concept/Aims/ Circuit Set-up: cont..



Circuit set up (SLAVE): . Tactile switch connected to Arduino as in exp2, and L..E.D set up as in EXp1, with a HC-05 BT module connected to the RX/TX pins of the Arduino, RX(BT) to TX(UNO) and TX(BT) to RX(UNO). Need to be disconnected when uploading software from IDE as shares the port with USB. Also when configuring in AT mode reverse the connections. HC-05 set to SALVE Mode and bound to Master module.

Feasibility Testing of Base Components: Cont..

Scripts and Method Outline:

Arduino_set_hc05_to_master

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

arduino_withButton_connect_Master

```
int data;
int blueLed = 2;
void setup()
{
  Serial.begin(115200);
  pinMode(blueLed, OUTPUT);

}

void loop()
{
  if(Serial.available())
  {
    data = Serial.read();
    if(data == '1'){

      digitalWrite(blueLed, HIGH);
    }
    if(data == '0'){
      digitalWrite(blueLed, LOW);
    }
  }
}
```

Feasibility Testing of Base Components: Cont..

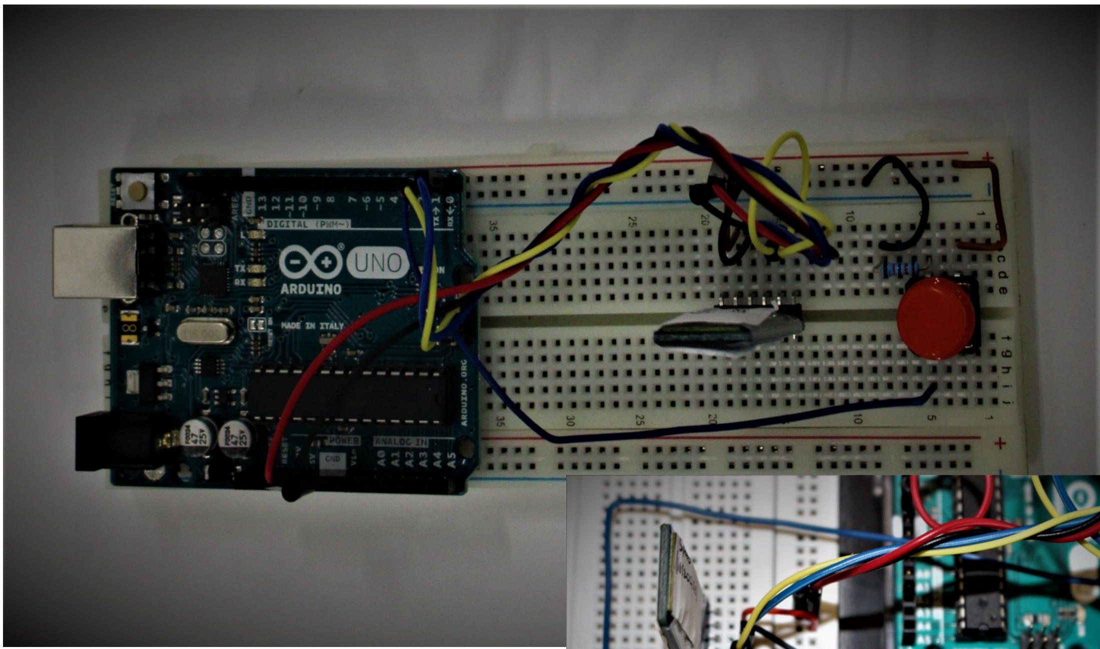
Scripts and Method Outline:

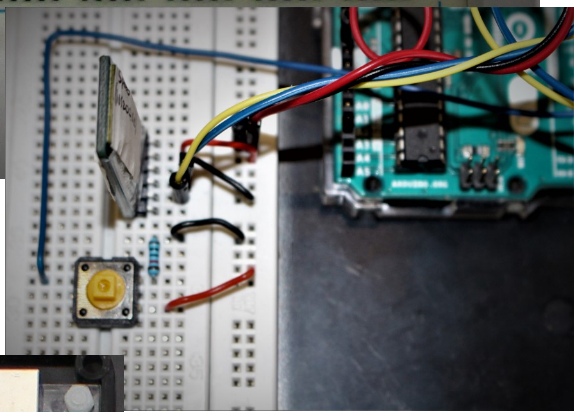
arduino_withButton_connect_Slave

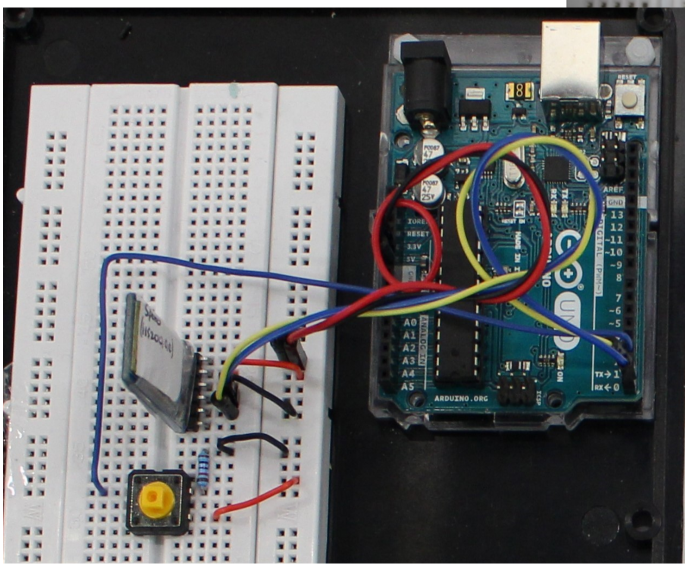
```
int buttonPin = 4;

bool lightState = true;

void setup()
{
  Serial.begin(115200);

  pinMode(buttonPin, INPUT);
}

void loop()
{
  buttonState = digitalRead(buttonPin);


  if (buttonState == HIGH  ) {


    Serial.println("1");

    delay(10);

  } else if (buttonState == LOW) {
    Serial.println(0);
    delay(10);
  }




}
```

Feasibility Testing of Base Components:

Pictures of Experiment :







Development Equipment: Wireless Comms

Development Equipment: Wireless Comms

Zigbee is for low-data rate, low-power applications and is an open standard. This, theoretically, enables the mixing of implementations from different manufacturers, but in practice, Zigbee products have been extended and customized by vendors and, thus, plagued by interoperability issues. In contrast to Wi-Fi networks used to connect endpoints to high-speed networks, Zigbee supports much lower data rates and uses a mesh networking protocol to avoid hub devices and create a self-healing architecture.

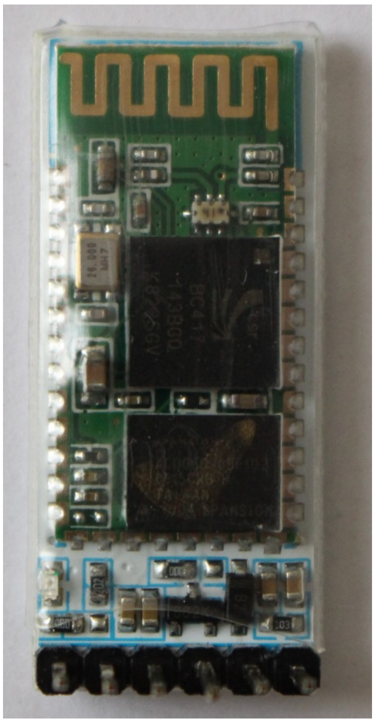
https://internetofthingsagenda.techtarget.com/definition/ZigBee

Development Equipment: Wireless Comms

Bluetooth is an open wireless technology standard for transmitting fixed and mobile electronic device data over short distances. Bluetooth was introduced in 1994 as a wireless substitute for RS-232 cables.

Bluetooth communicates with a variety of electronic devices and creates personal networks operating within the unlicensed 2.4 GHz band. Operating range is based on device class. A variety of digital devices use Bluetooth, including MP3 players, mobile and peripheral devices and personal computers.

https://www.techopedia.com/definition/26198/bluetooth



Development Equipment: Wireless Comms

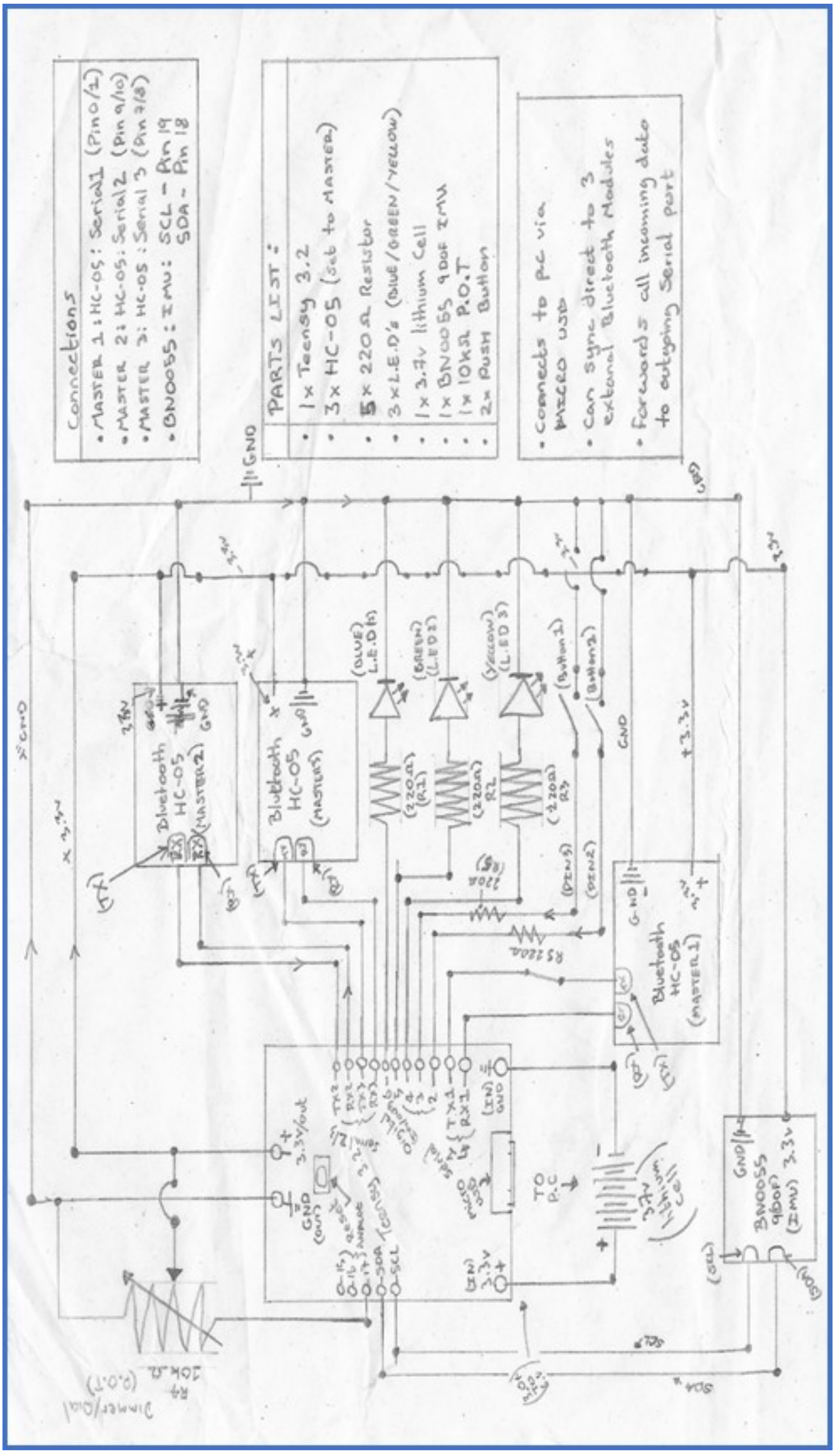
Stands for "Bluetooth Low Energy." BLE (also "Bluetooth LE") is a variation of the Bluetooth wireless standard designed for low power consumption. It was introduced by the Bluetooth Special Interest Group (Bluetooth SIG) in December 2009 as part of the Bluetooth 4.0 specification.

Some Bluetooth applications, such as audio streaming and data transfers, require a strong consistent signal and large bandwidth. Other applications do not need a strong signal and require less electrical power. BLE is was developed for these types of applications. Examples include:

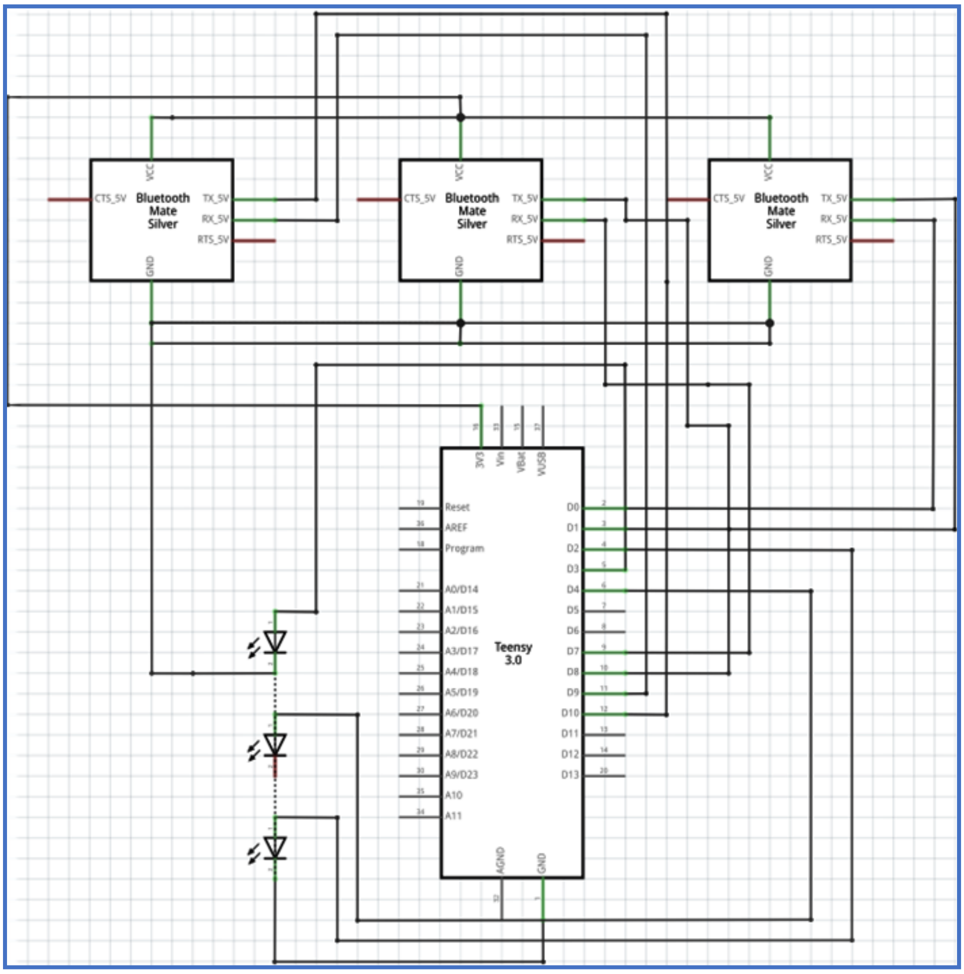
- wearable devices, such as fitness trackers
- smart home appliances
- proximity sensors

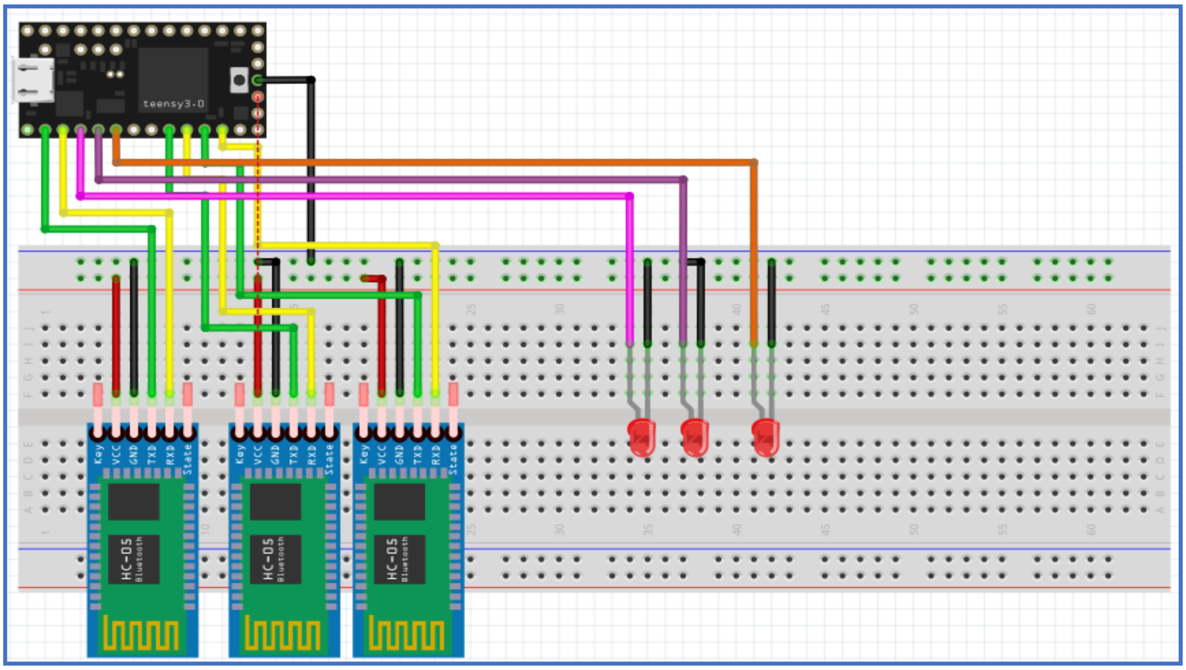
https://techterms.com/definition/ble

Prototype Phase ONE: Data Control Device

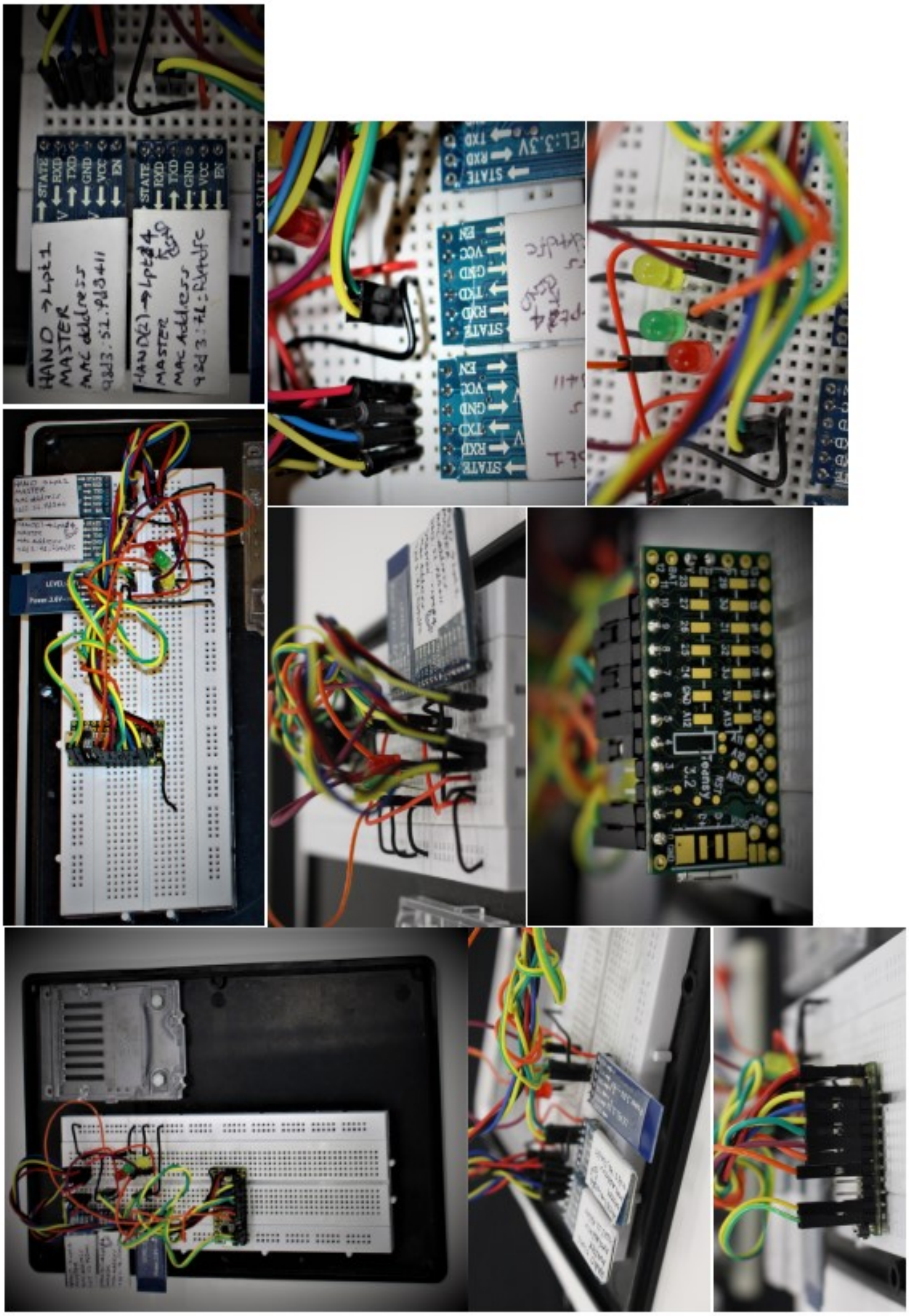


Prototype Phase ONE: Data Control Device Cont..





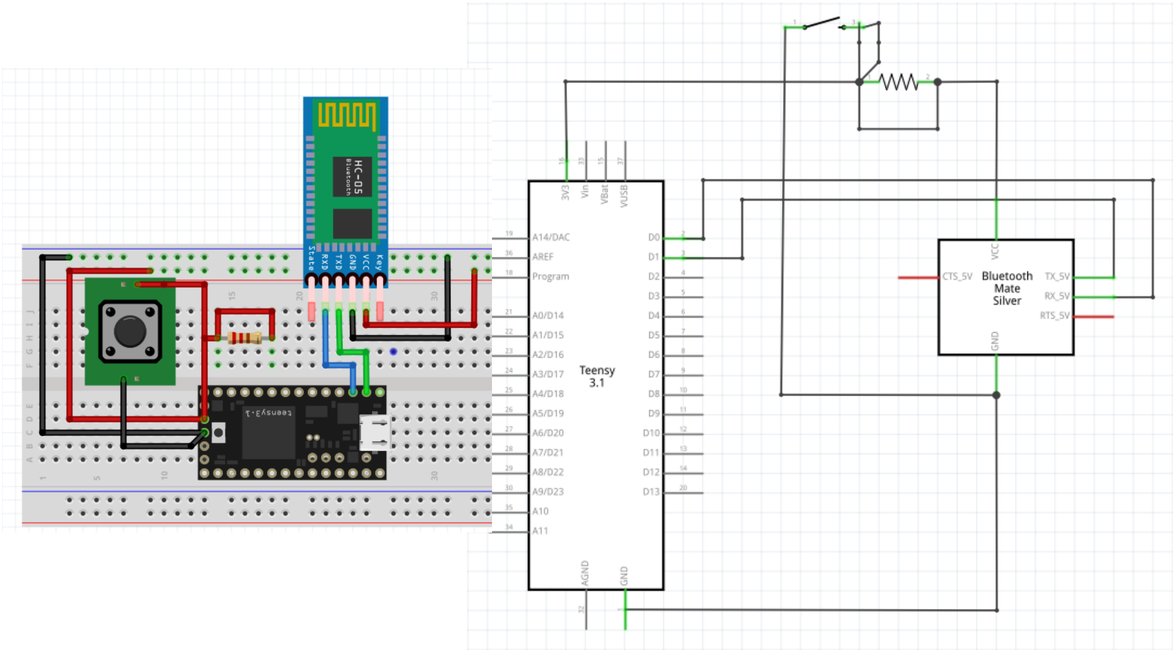
Prototype Phase ONE: Data Control Device Cont..

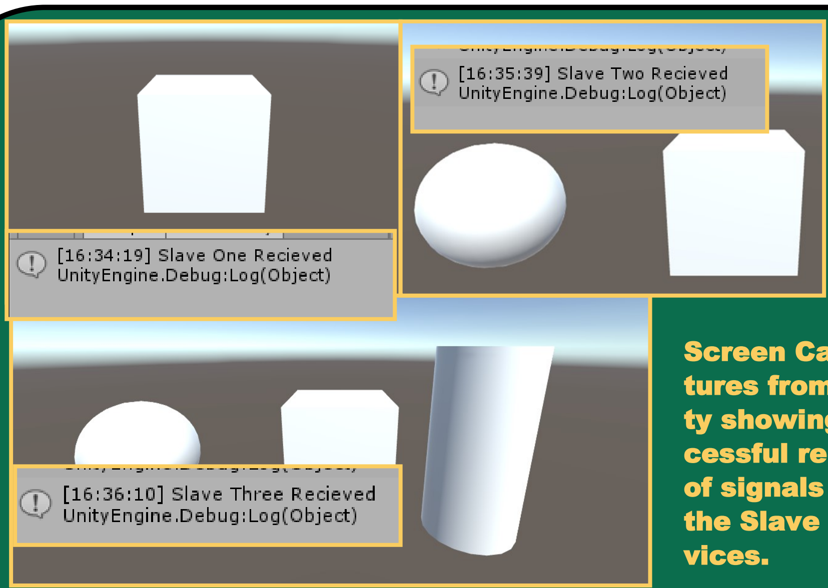


Prototype Phase ONE: Data Control Device Cont..

```
void forwardData(){
   char lpt4;
   char lpt3;
   char lpt2;

    while (limpet4_Serial.available() > 0){
       digitalWrite(lpt4_connected,HIGH);
      lpt4 = limpet4_Serial.read();

      Serial.print(lpt4);

    }
    while (limpet3_Serial.available() > 0){
       digitalWrite(lpt3_connected,HIGH);
      lpt3 = limpet3_Serial.read();

      Serial.print(lpt3);
    }
    while (limpet2_Serial.available() > 0){
       digitalWrite(lpt2_connected,HIGH);
      lpt2 = limpet2_Serial.read();

      Serial.print(lpt2);
    }

        digitalWrite(lpt4_connected,LOW);
        digitalWrite(lpt3_connected,LOW);
        digitalWrite(lpt2_connected,LOW);



  delay(10);
}
```

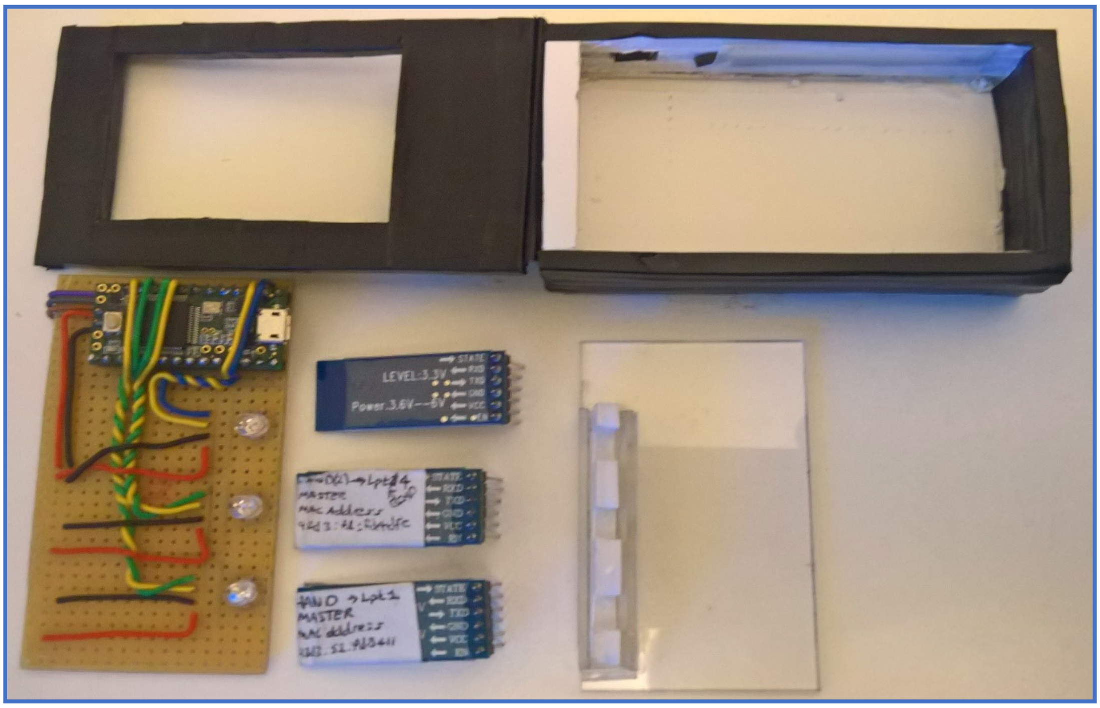
Prototype Phase ONE: Data Control Device Cont..

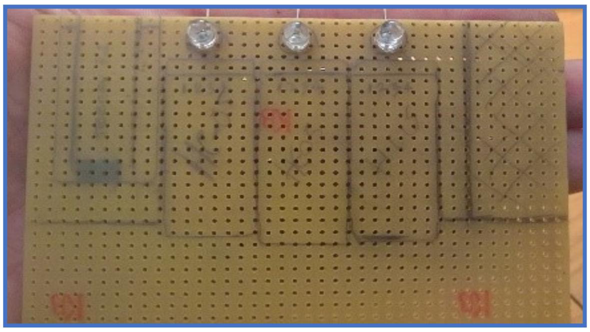


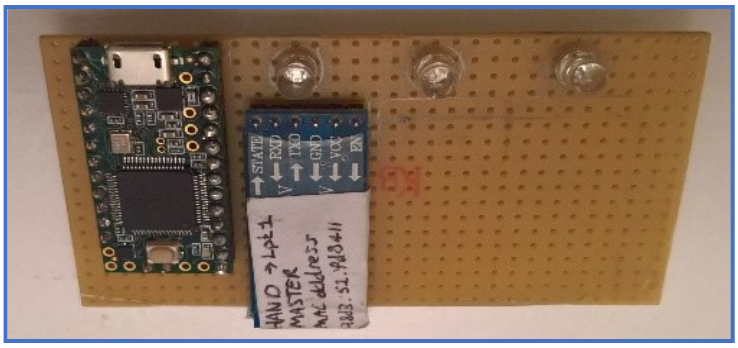


Screen Captures from Unity showing successful receipt of signals from the Slave devices.

Prototype Phase ONE: Data Control Device Cont..

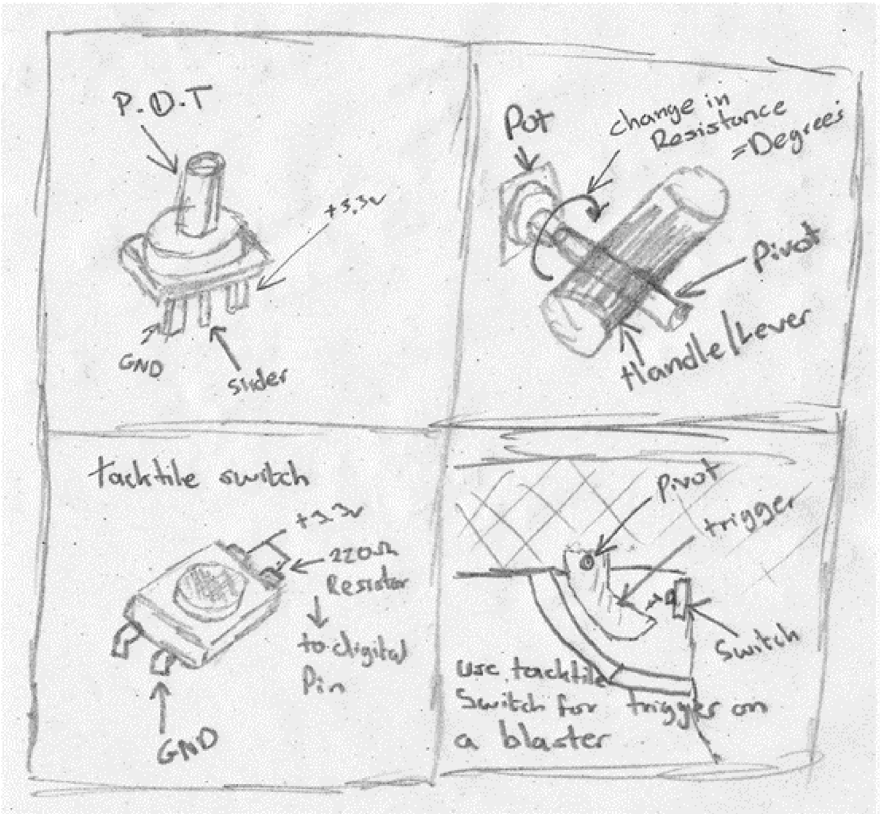






Prototype Phase ONE: Data Control Device Cont..

Prototype Phase Two : Static Interactions

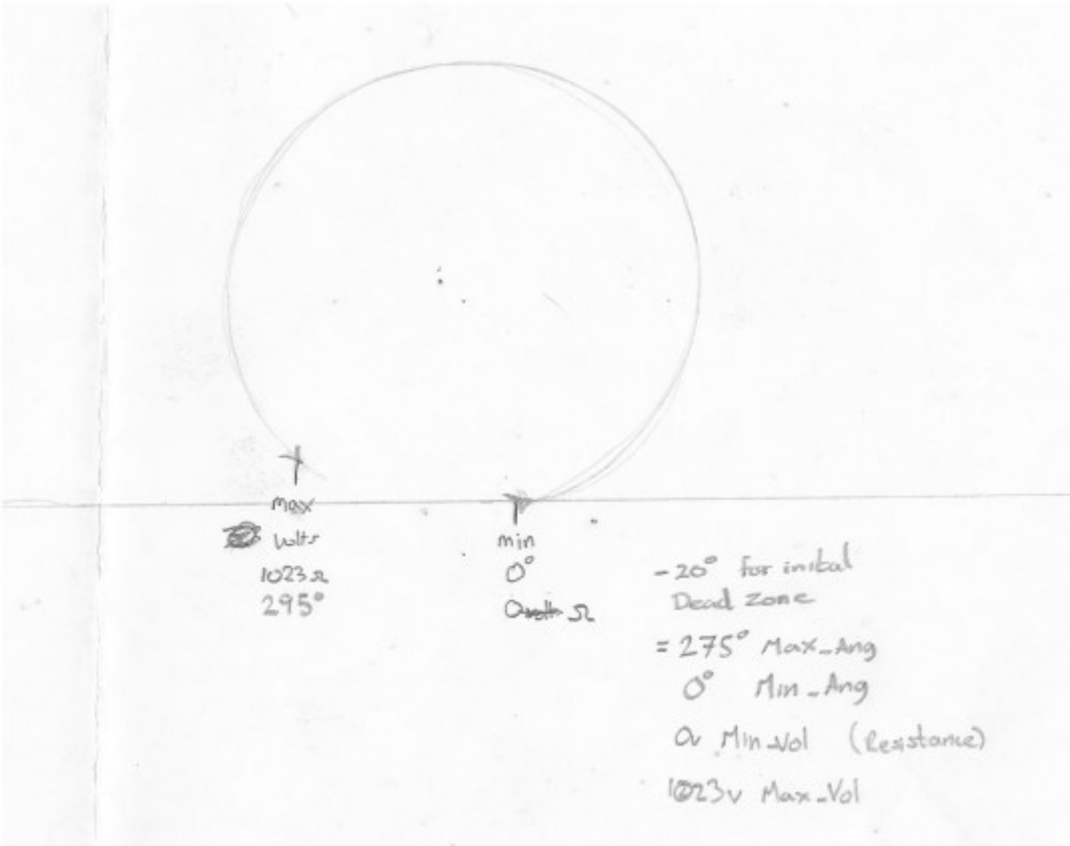


Prototype Phase Two : Static Interactions Cont..

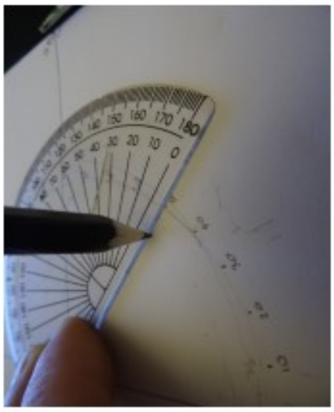
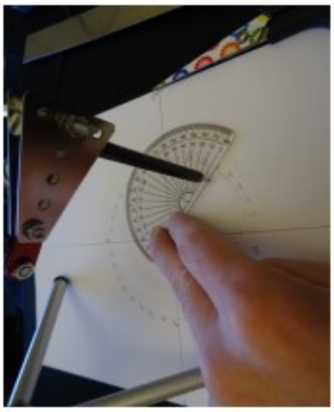
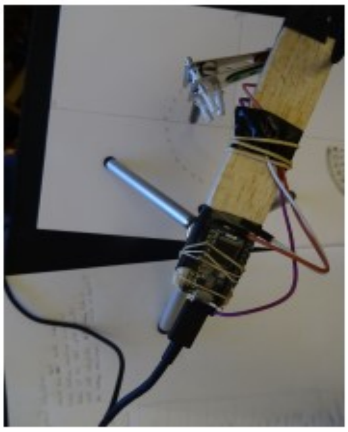
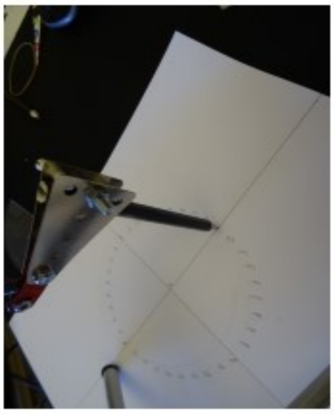
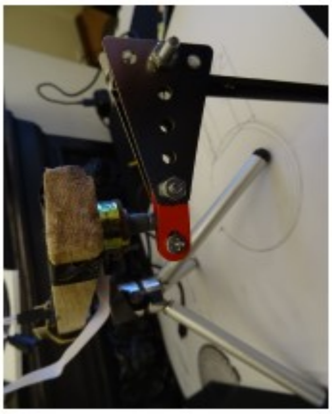
```
void CalculateAngle(){
  max_ana = 314;
  min_ana = 0;
  max_vol = 1023;
  min_vol = 0;

dx = max_ana - min_ana;
dy = max_vol - min_vol;
m = dx / dy;
//val = analogRead(21);
//ana_raw =  1166-val;
ana_raw = analogRead(21);
ana_offset = ana_raw - min_ana;
vol_offset = ana_offset * m;
ang = vol_offset + min_vol;


}
```

```
void sendBluetoothHandVersionOne(){
  Limpet_Door.print("<");
  Limpet_Door.print("Door");
  Limpet_Door.print(",");
  Limpet_Door.print(ang);
  Limpet_Door.print(",");
  Limpet_Door.print(t);
  Limpet_Door.print(">");
}
```

Prototype Phase Two : Static Interactions Cont..



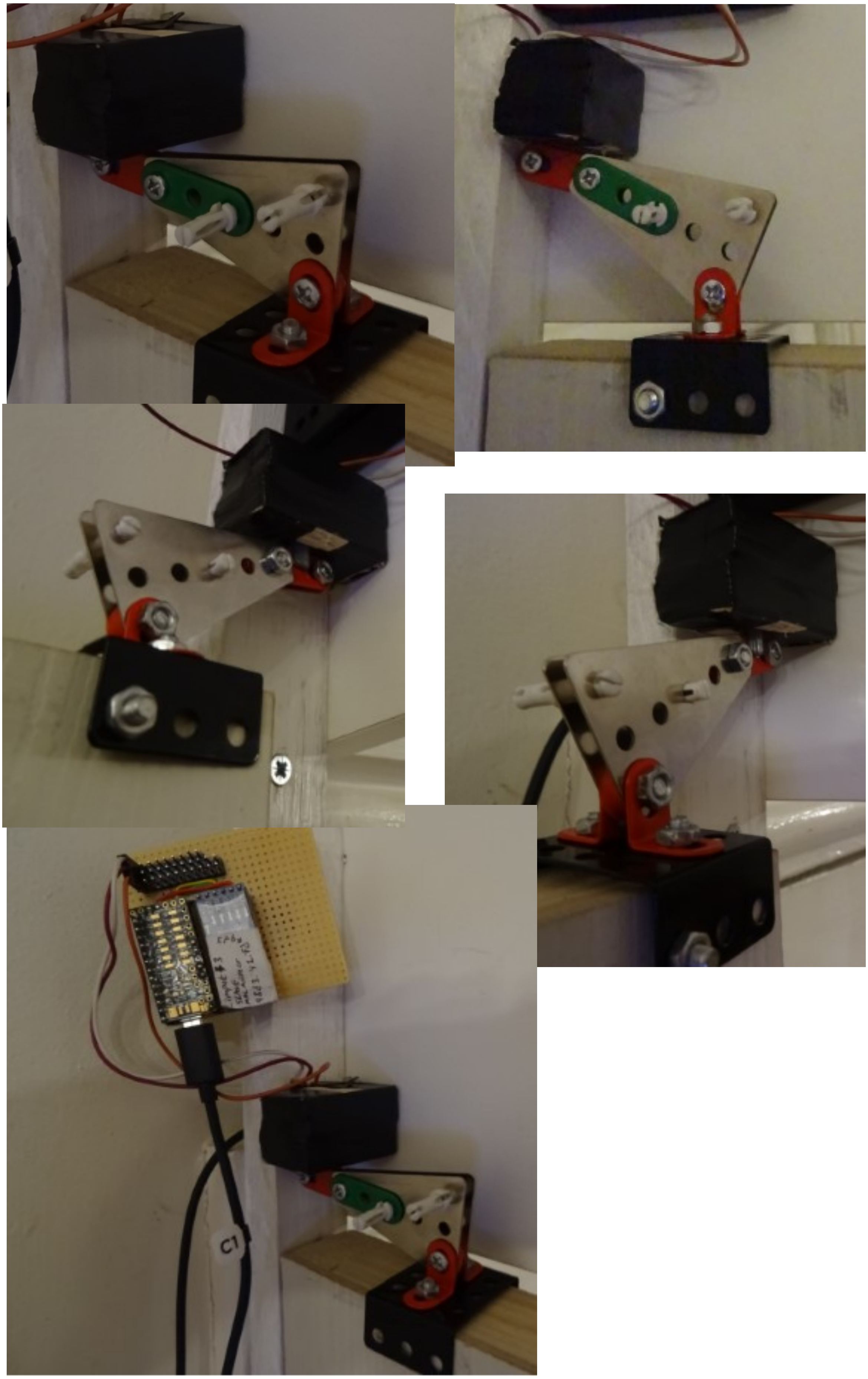
Prototype Phase Two : Static Interactions Cont..

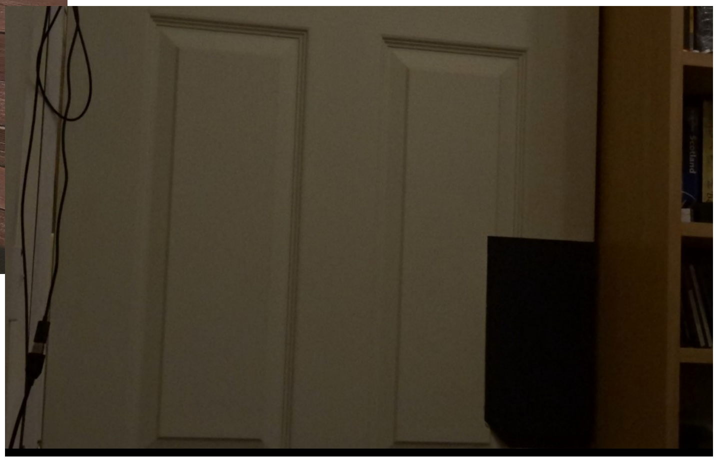




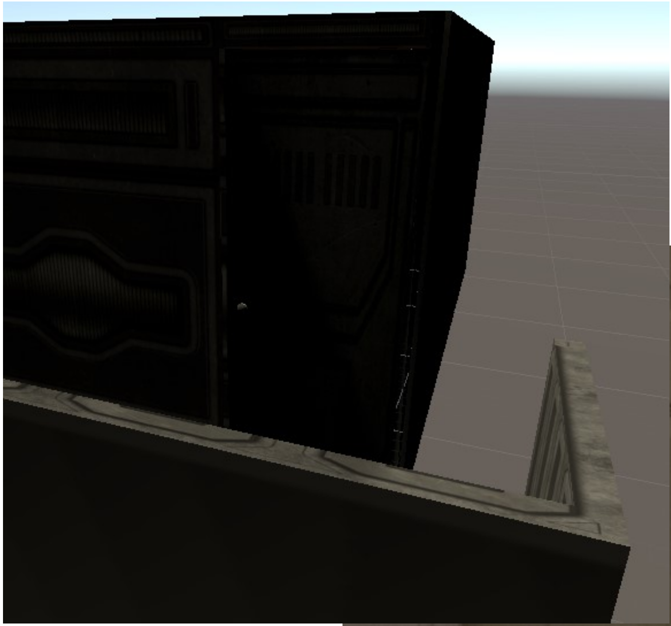

Prototype Phase Two : Static Interactions Cont..

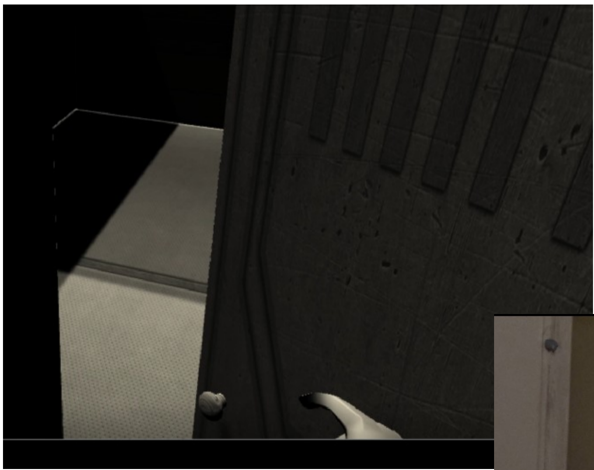


Prototype Phase Two : Static Interactions Cont..

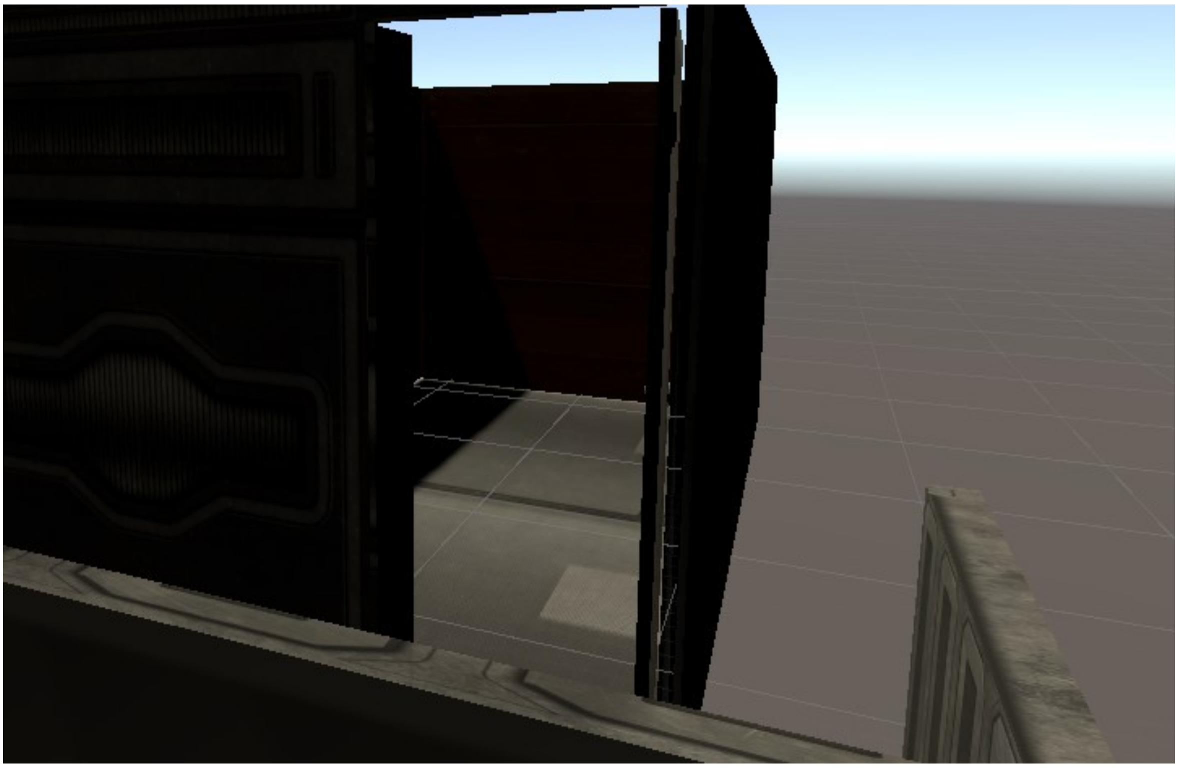


Prototype Phase Two : Static Interactions Cont..

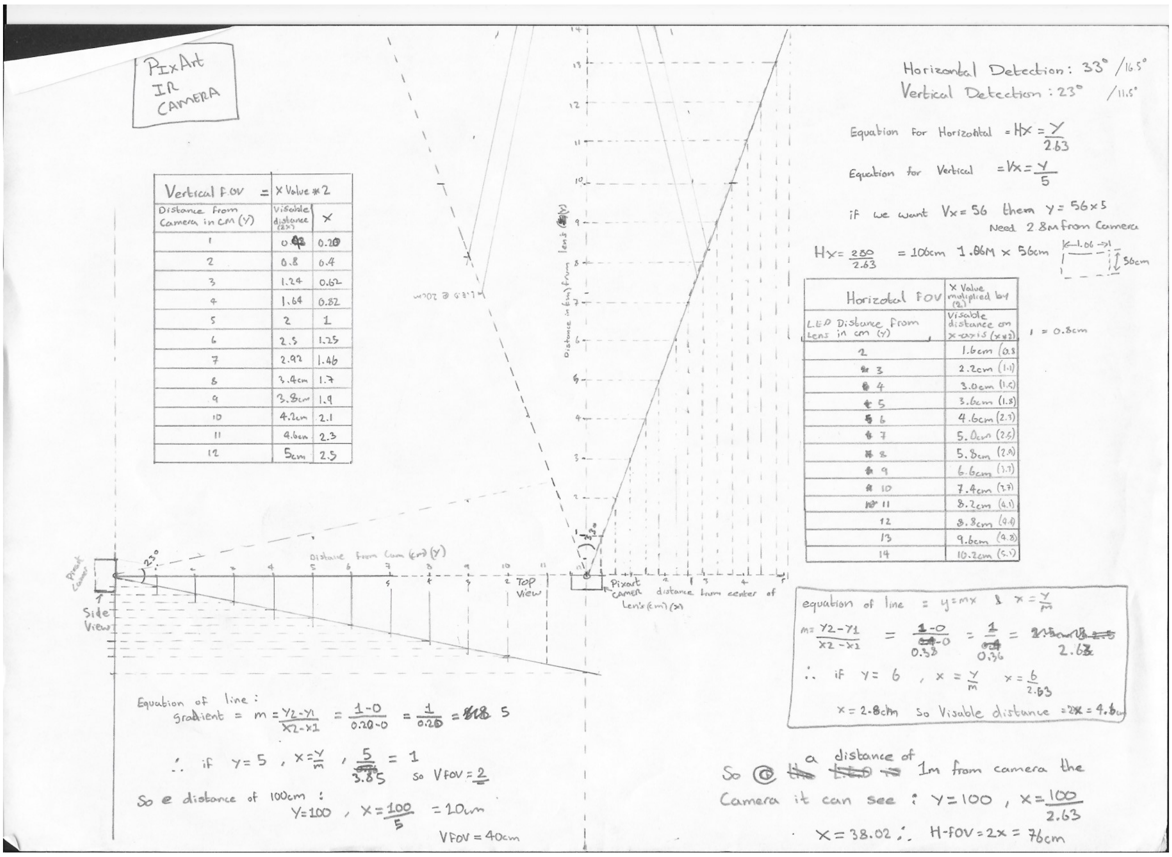




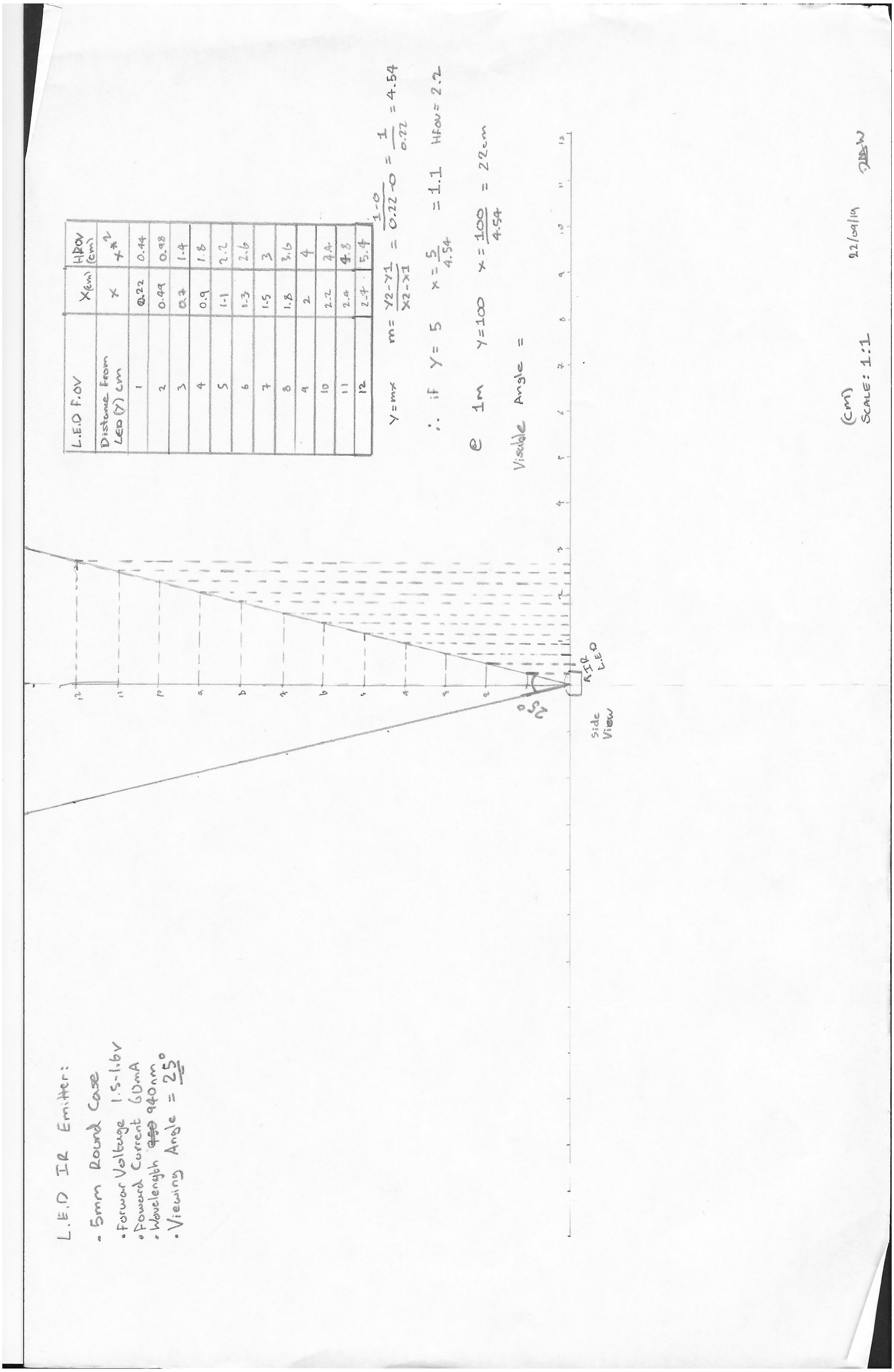
Prototype Phase Two : Static Interactions Cont..



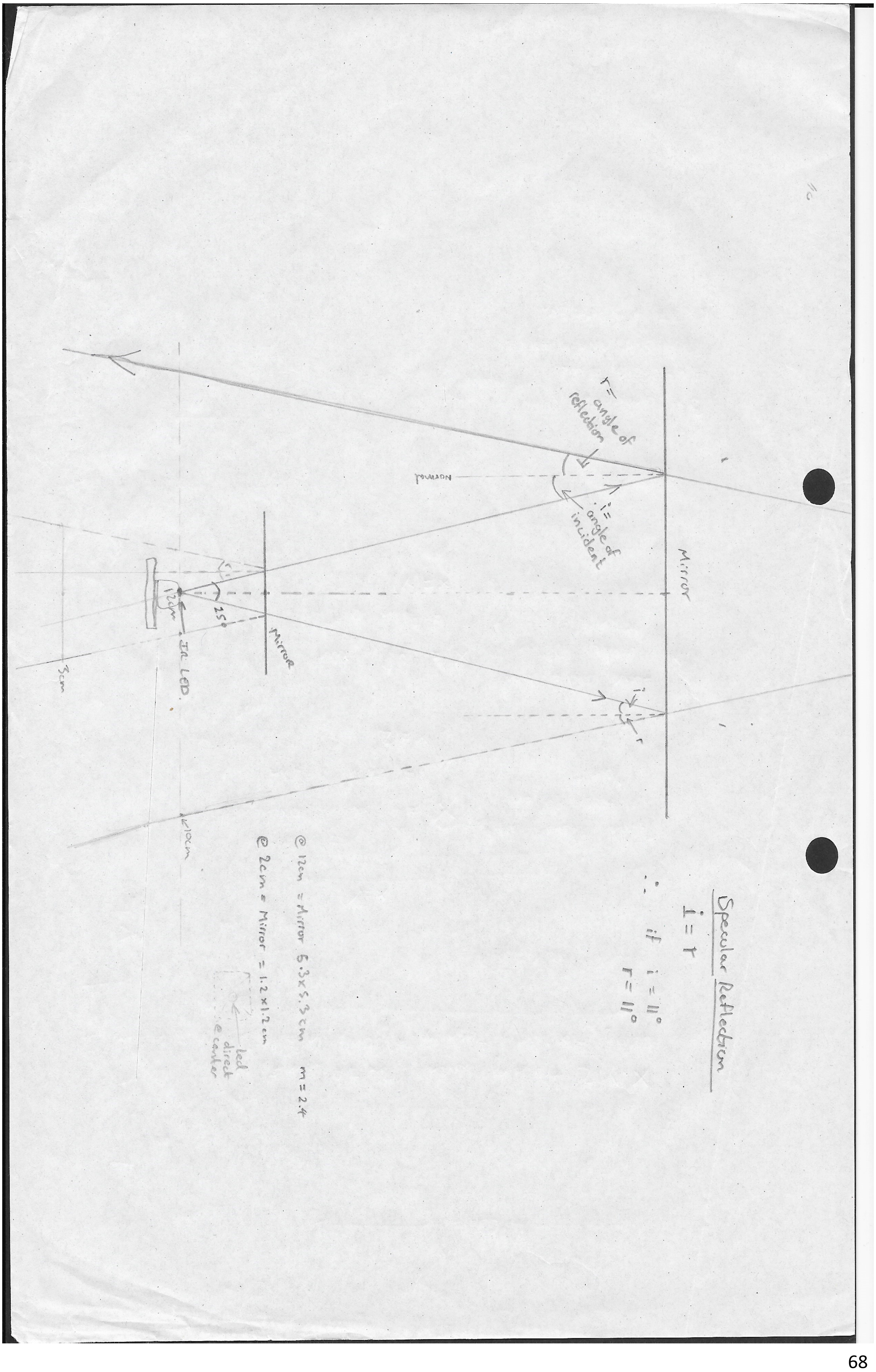
Prototype Phase Three : Mobile Interactions



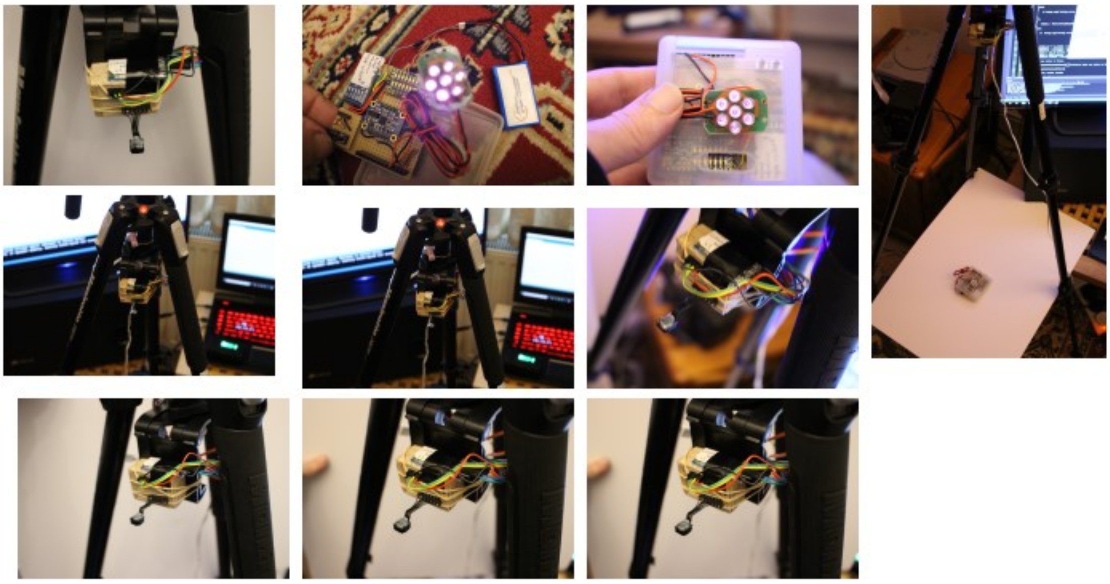
Prototype Phase Three : Mobile Interactions Cont..



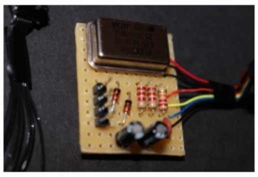
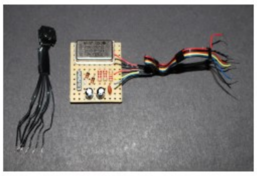
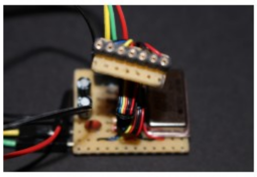
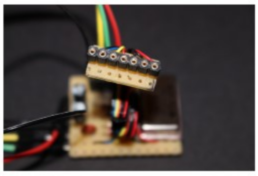
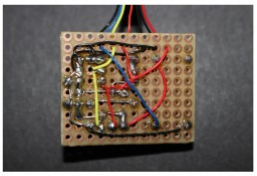
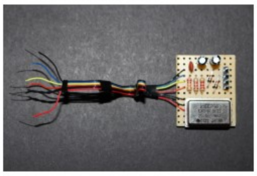
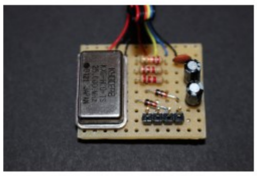
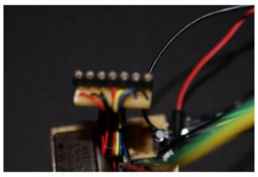
Prototype Phase Three : Mobile Interactions Cont..

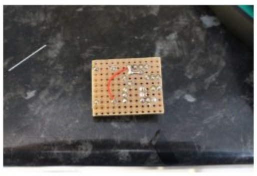


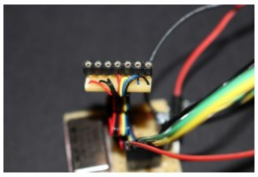
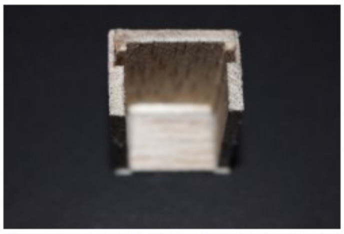
Prototype Phase Three : Mobile Interactions Cont..

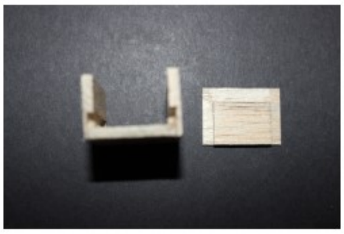


Prototype Phase Three : Mobile Interactions Cont..

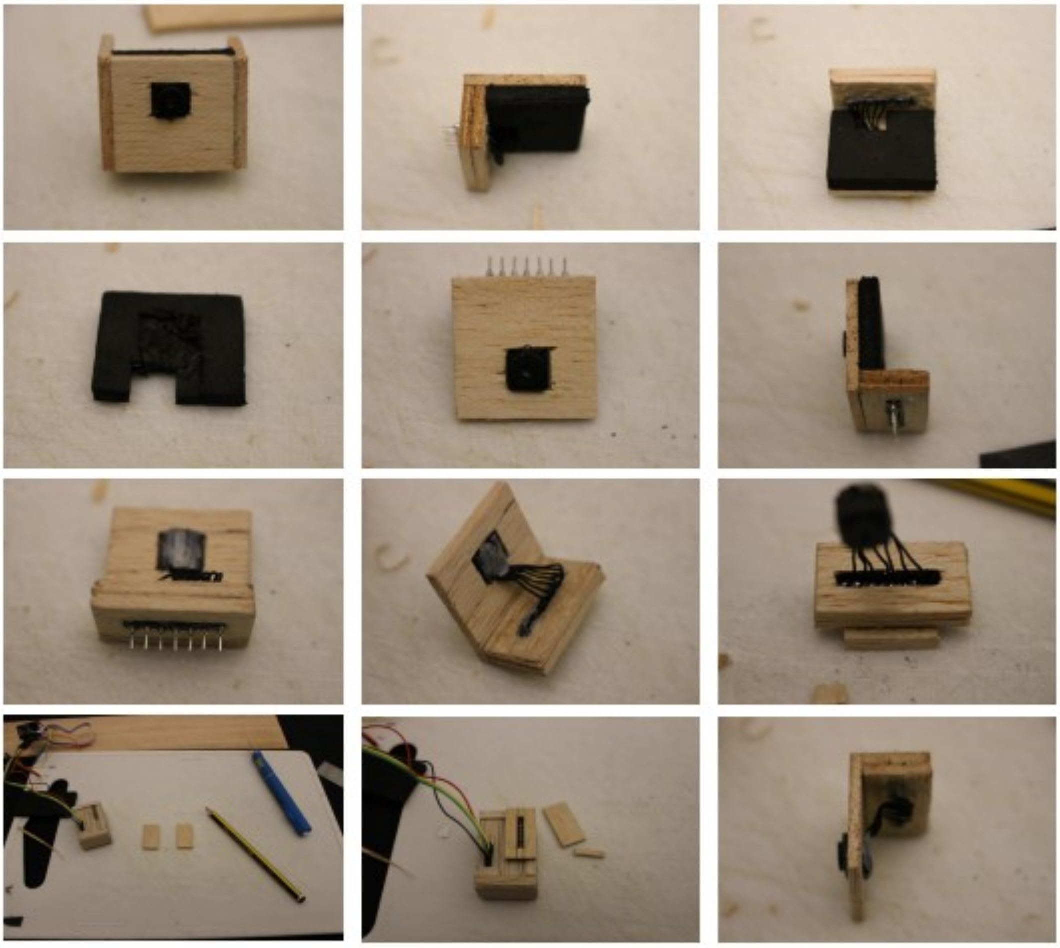


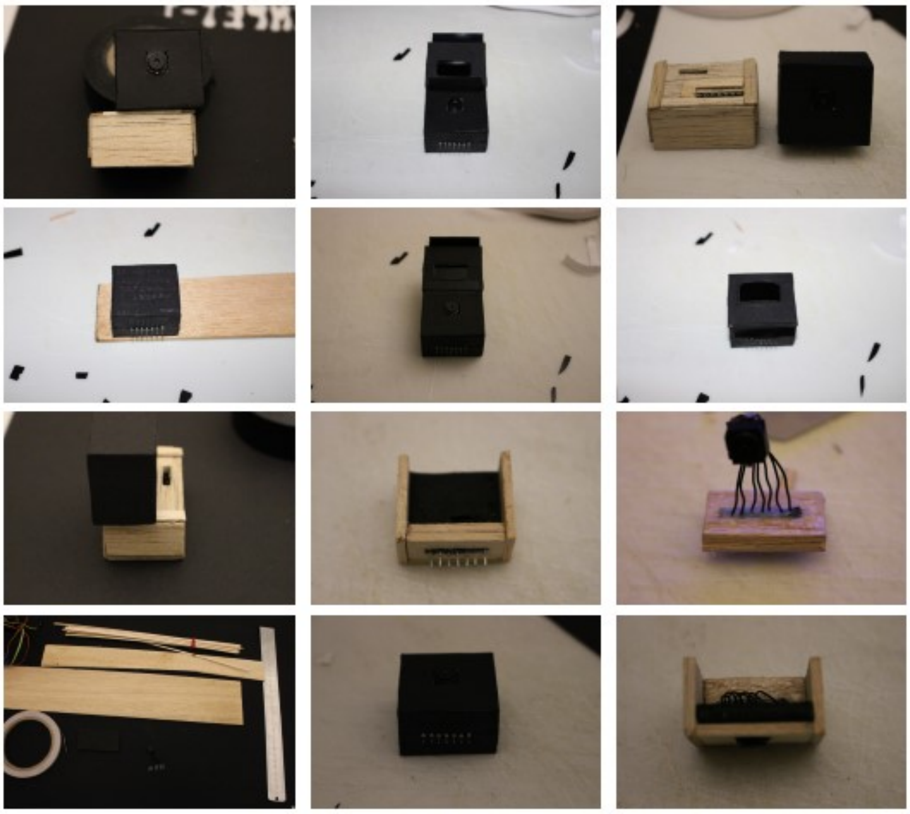





Prototype Phase Three : Mobile Interactions Cont..

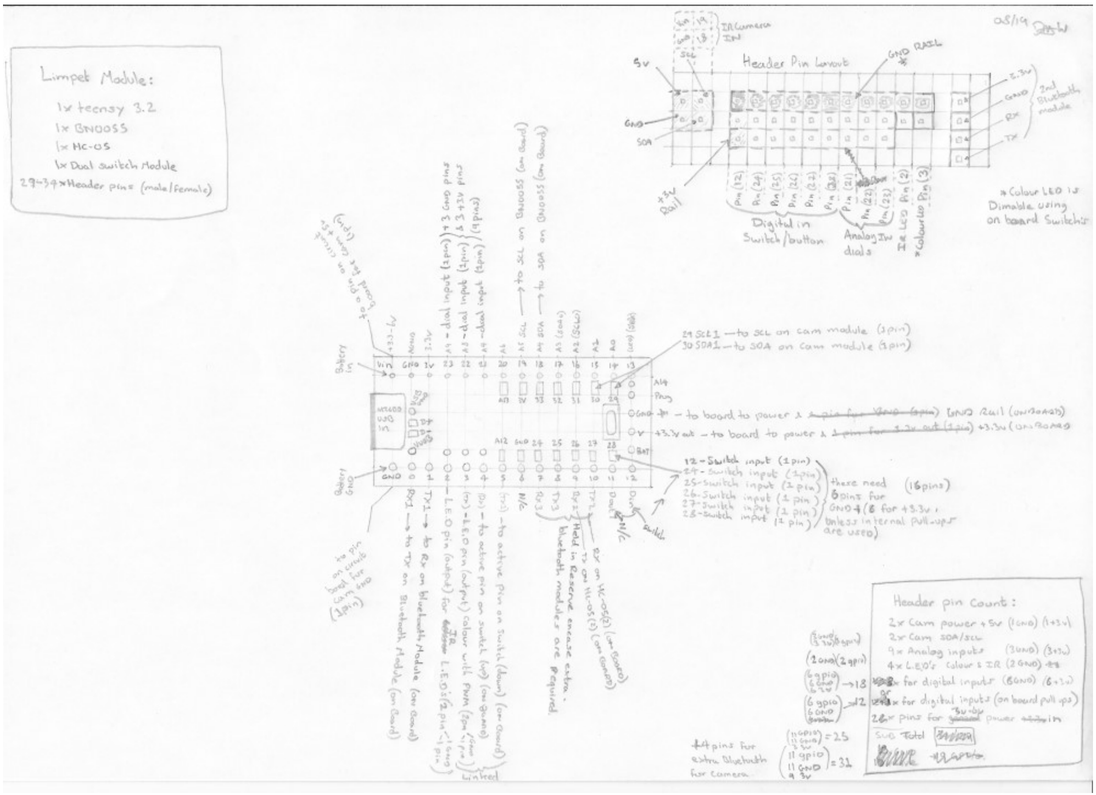




Prototype Phase Three : Mobile Interactions Cont..

- Each limpet should be able to take ~~one~~:
1x input from contact sensor,
1x input from button
2x input from Analog

So : Each limpet will need min : 3x Analog/PWM & 1x digital
4 total inputs + Vcc in



Limpet -4

Pin	colour	Pin	Colour
0	Blue	11	
1	Green	12	
2		13	
3	orange	14	
4	Purple	15	
5	Blue 2	16	
6		17	
7		18	GREEN
8		19	YELLOW
9		20	
10			

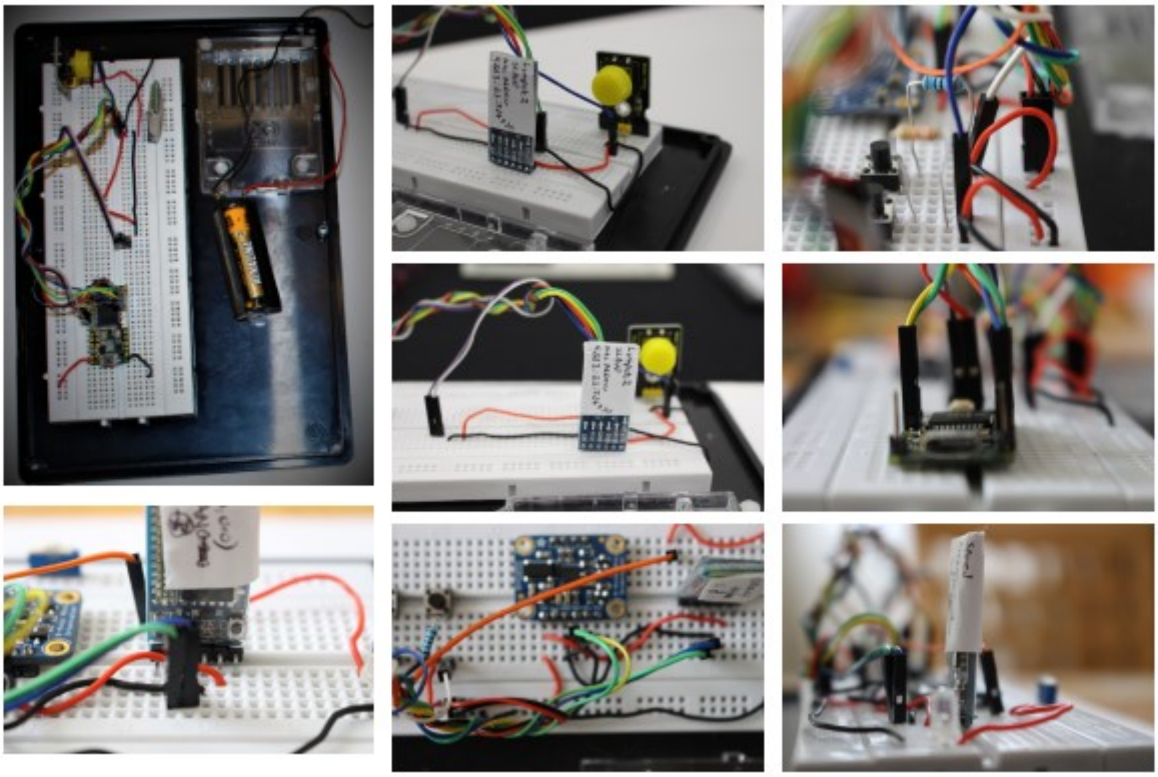
Pin	Connected to
0	Bluetooth TX
1	Bluetooth RX
3	+ Led
4	up switch
5	down switch
18	IMU SDA
19	IMU SCL
Vout	Vin IMU
GND	GND IMU
GND/IMU	to GND Switch Module
Vin/IMU	to + Switch Module
Vin/IMU	to Vcc Bluetooth
GND/Switch	to GND Bluetooth
GND/BIT	to GND LE.D.

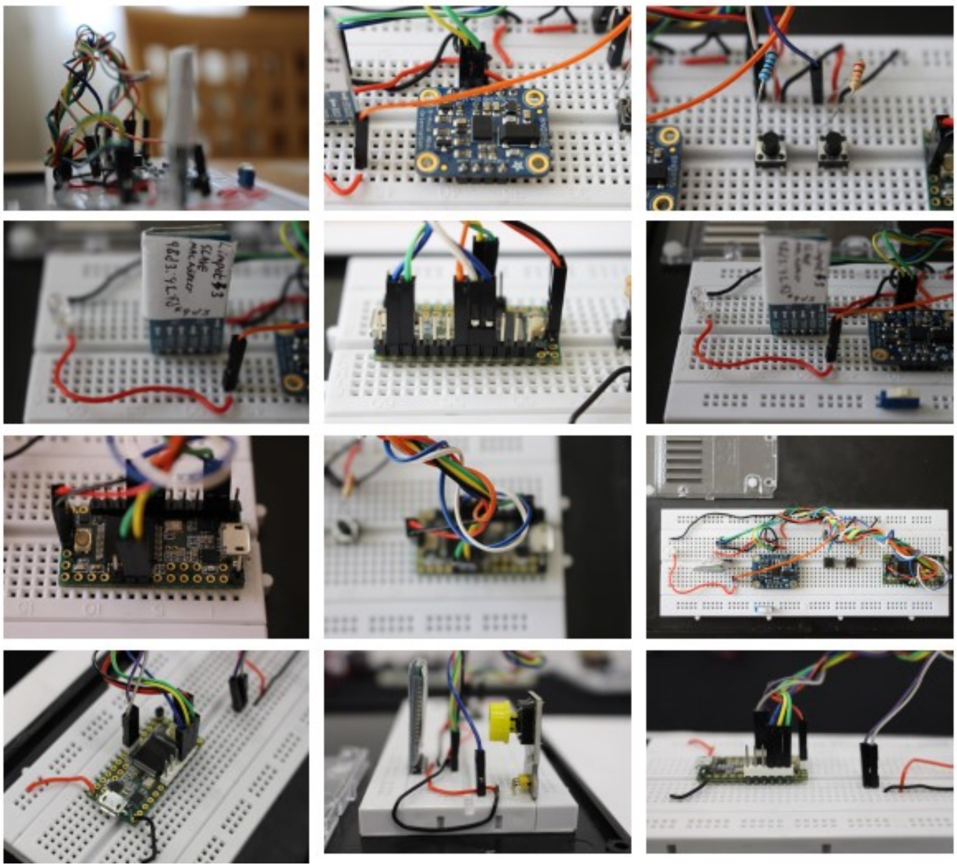
Pin	
(PWM) → 6 7 8	Reserved for Aditional Digital inputs

Pin	
20 21 22	for Analog inputs

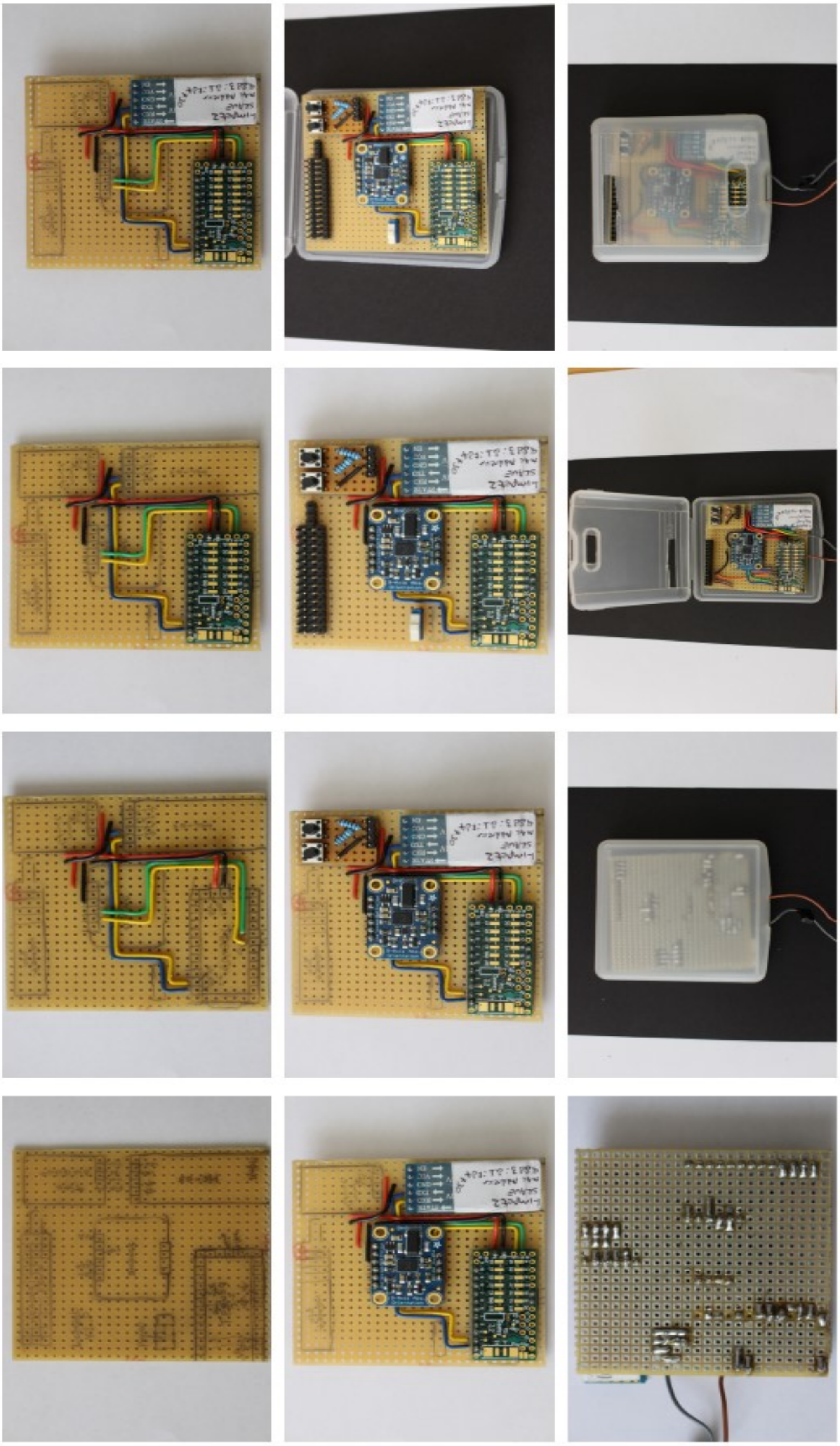
VBAT in	Brown	to + on Battery
GND	Grey	to − on Battery

Prototype Phase Three : Mobile Interactions Cont..

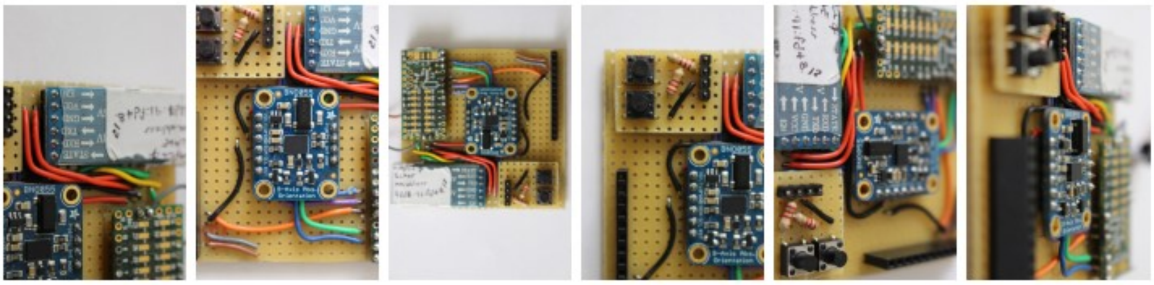


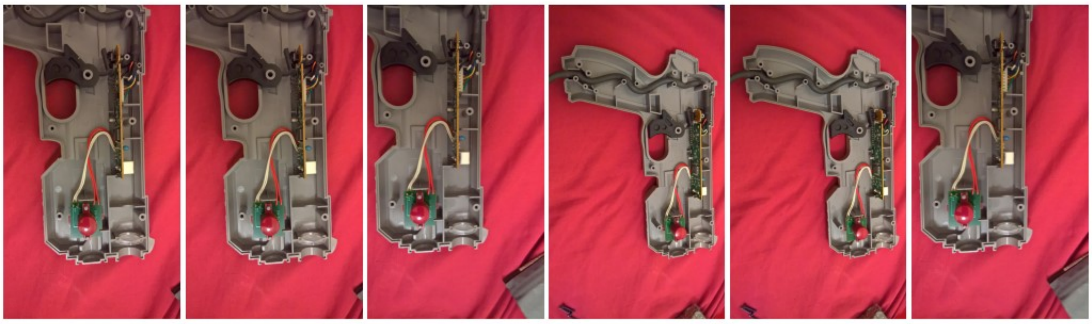


Prototype Phase Three : Mobile Interactions Cont..

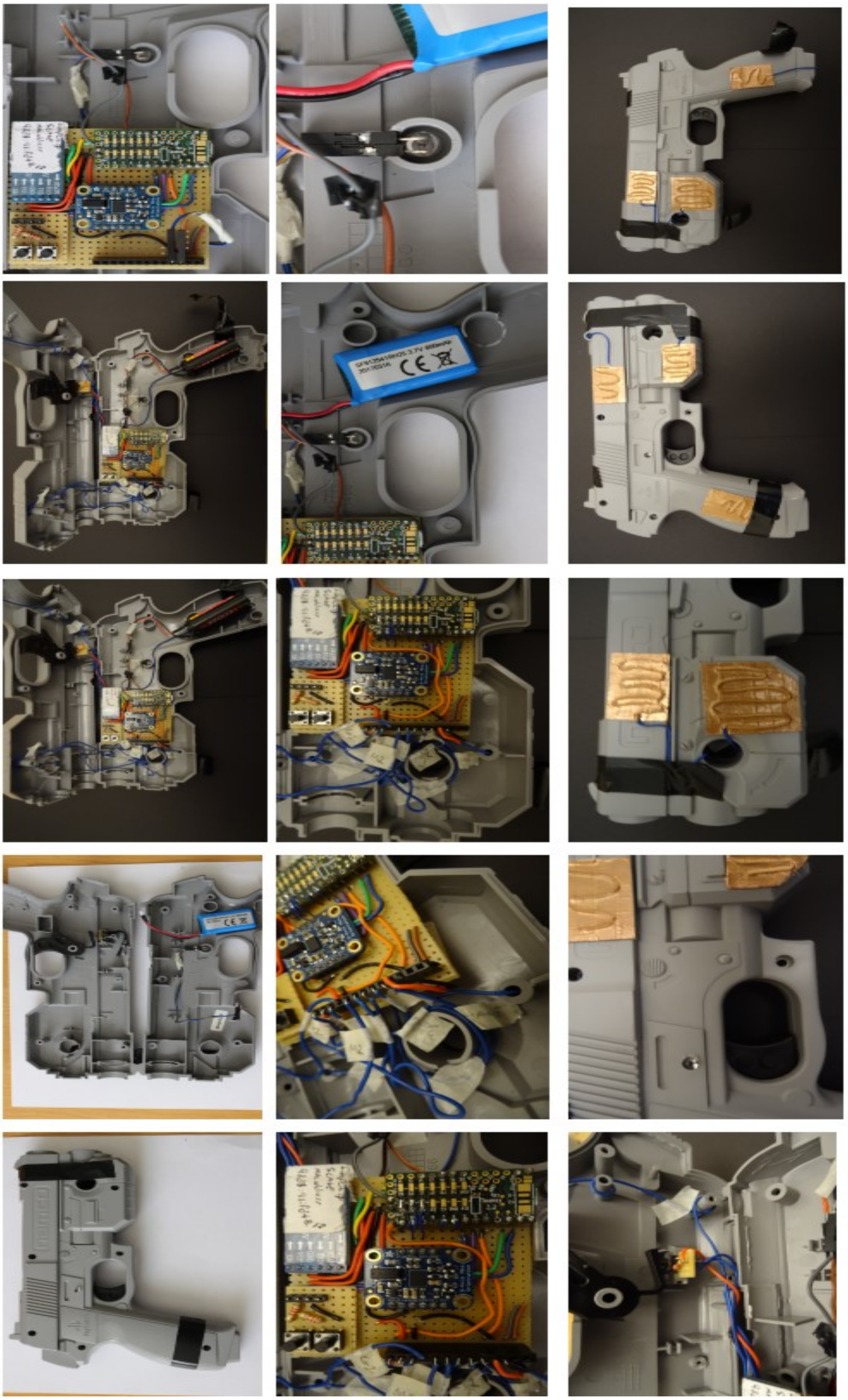


Prototype Phase Three : Mobile Interactions Cont..

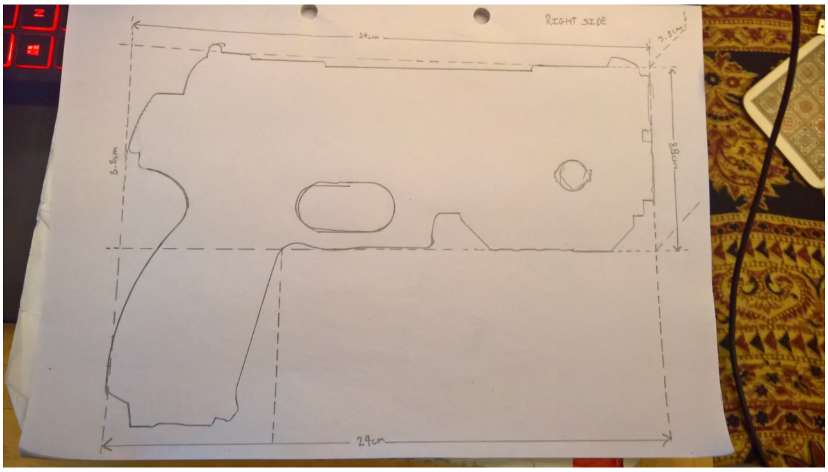




Prototype Phase Three : Mobile Interactions Cont..

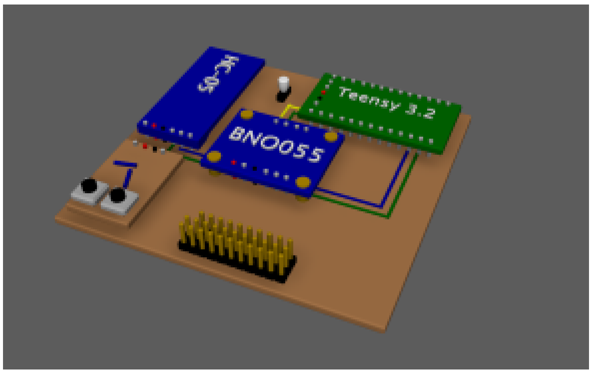


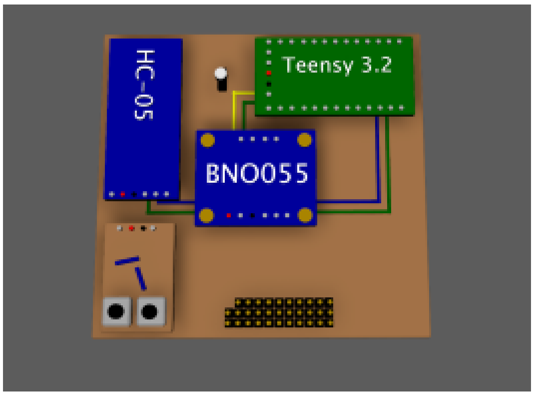
Prototype Phase Three : Mobile Interactions Cont..

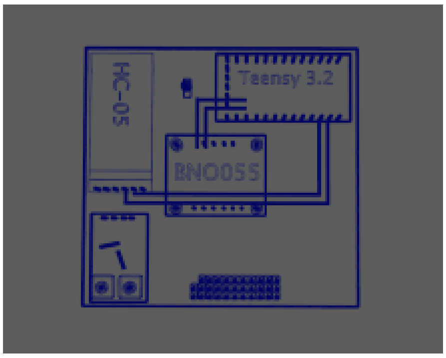


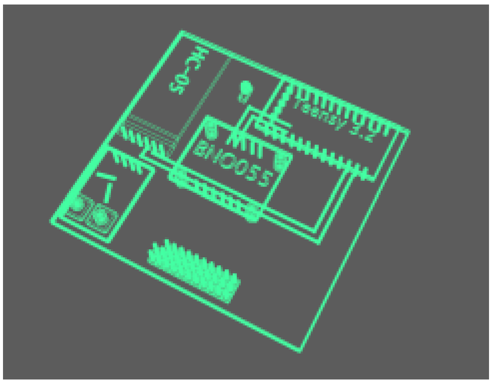
Prototype Phase Three : Mobile Interactions Cont..

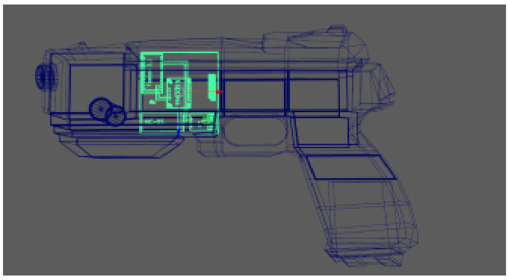
Prototype Phase Three : Mobile Interactions Cont..

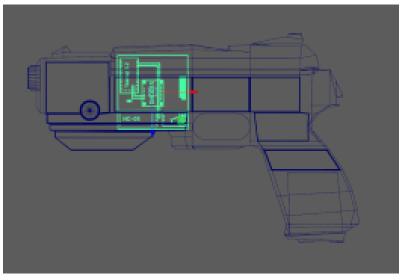


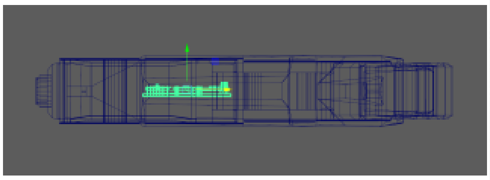


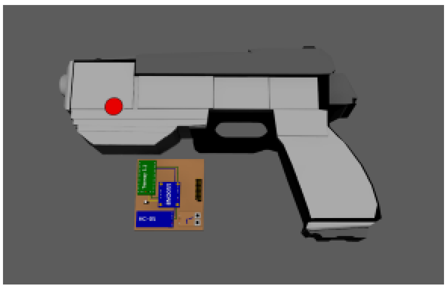


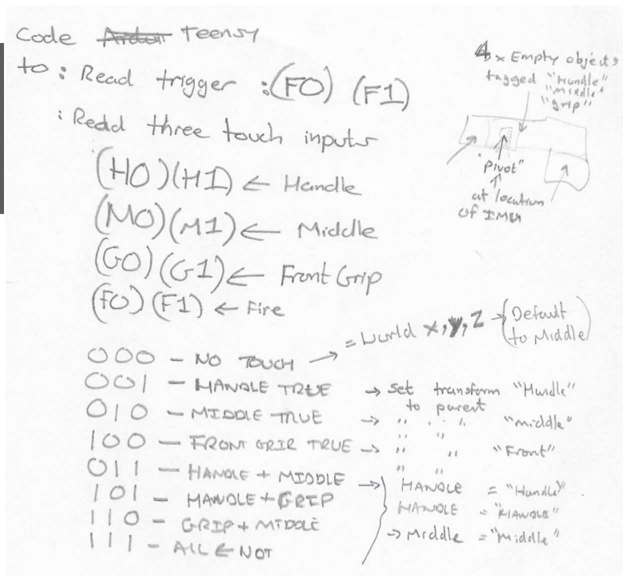




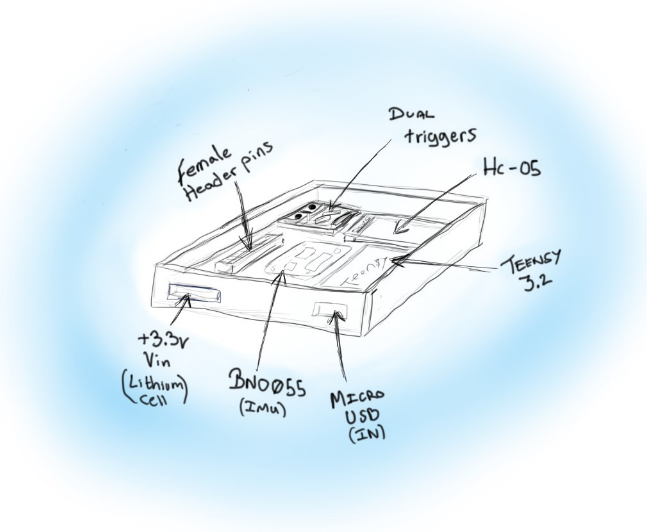


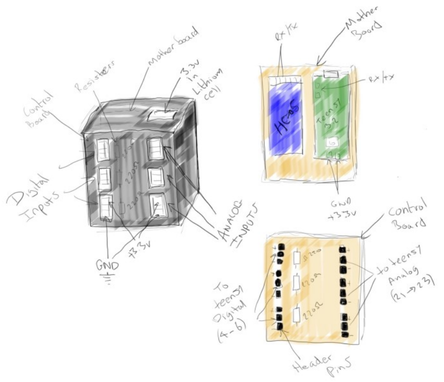


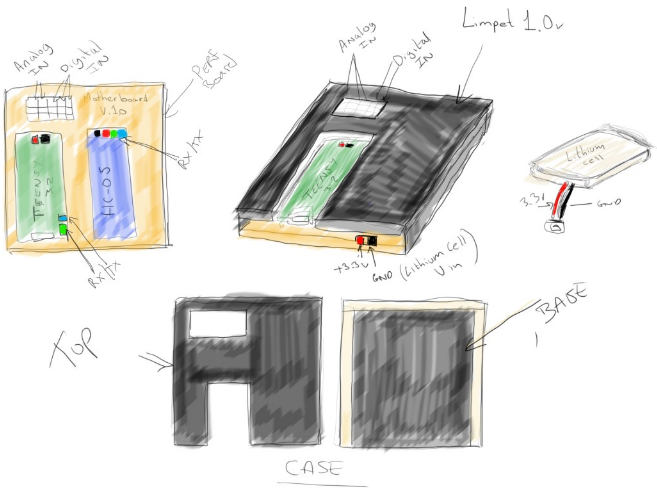




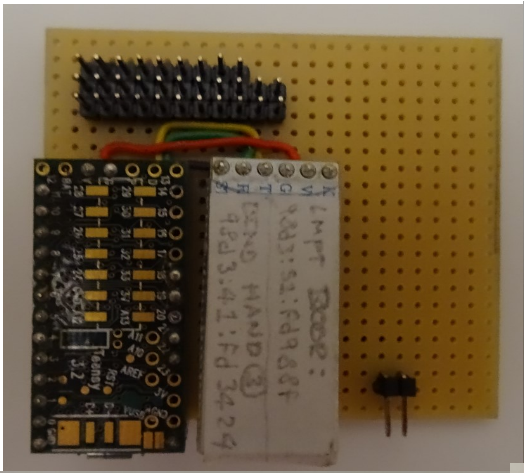
Prototype Phase Four: Full System:

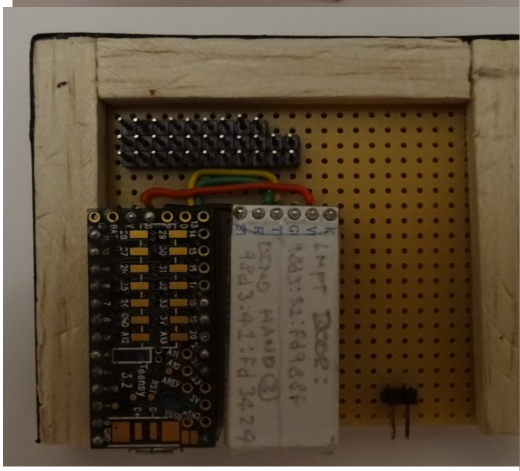
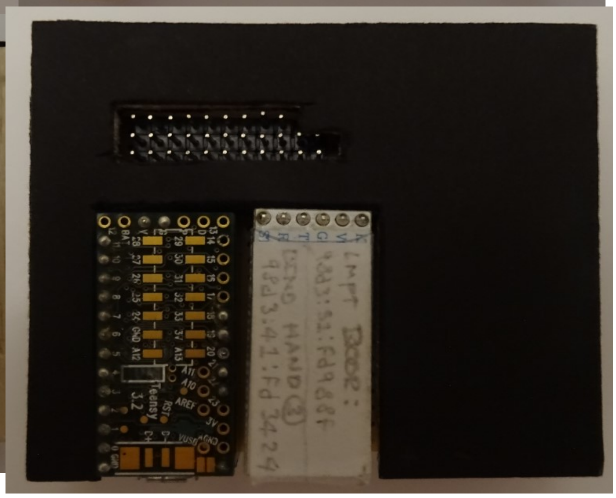






Prototype Phase Four: Full System:



```
private static readonly Regex rgx = new Regex(@"^[A-Z]\-?[0-9]?[0-9]?[0-9]\.?\d*$");
```

```
public static bool VerifyIncomingString(string incoming)
{
    return rgx.IsMatch(incoming);
}
```

Prototype Phase Four: Full System:

```
public void Lmpt1Loop()
{
    while (IsLooping1())
    {

        ConvertLmpt2ToFl
        Debug.Log("2nd T


    }

} //code to use incoming
```

```
public void StartThread()
{
    BuildQueues();

    hand = new Thread(HandLoop);
    lmpt1 = new Thread(Lmpt1Loop);
    hand.Start();
    lmpt1.Start();
    Debug.Log("started threads");

}//creates and starts the thread;
```

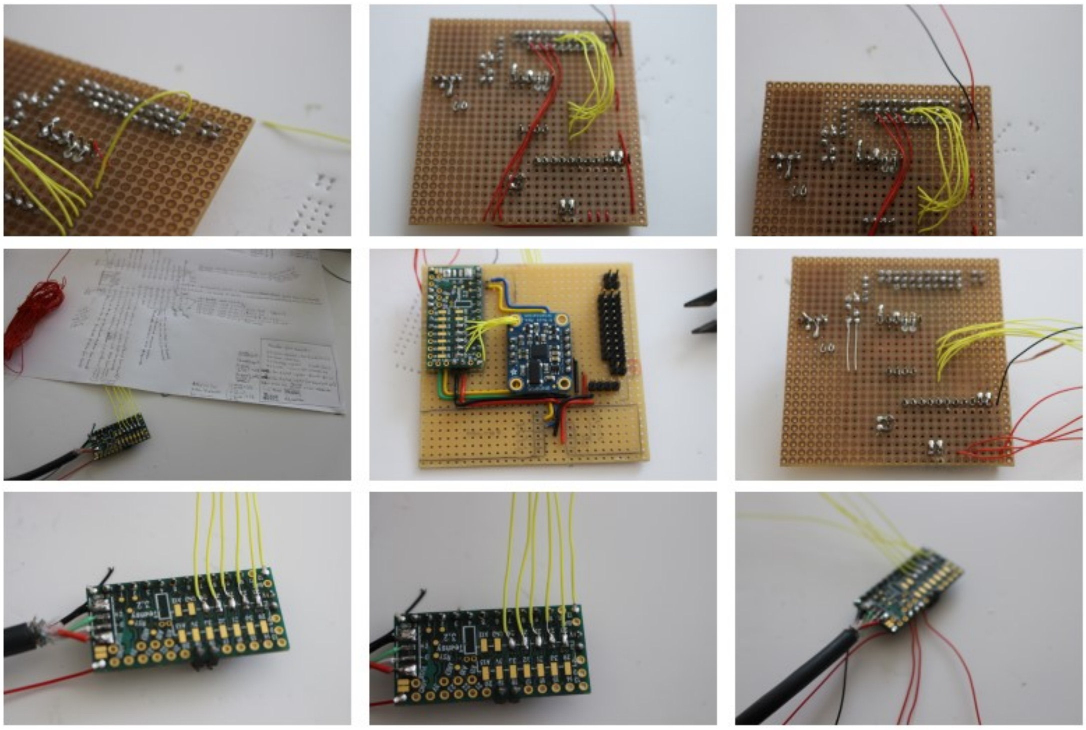
```
public void BuildQueues()
{
    outputQueue = Queue.Synchronized(new Queue());
    inputQueue = Queue.Synchronized(new Queue());
    inputQueueX = Queue.Synchronized(new Queue());
    inputQueueY = Queue.Synchronized(new Queue());
    inputQueueZ = Queue.Synchronized(new Queue());
    inputQueueF = Queue.Synchronized(new Queue());
    inputQueueA = Queue.Synchronized(new Queue());
    inputQueueB = Queue.Synchronized(new Queue());
    inputQueueC = Queue.Synchronized(new Queue());
} // creates the input and output threads
```

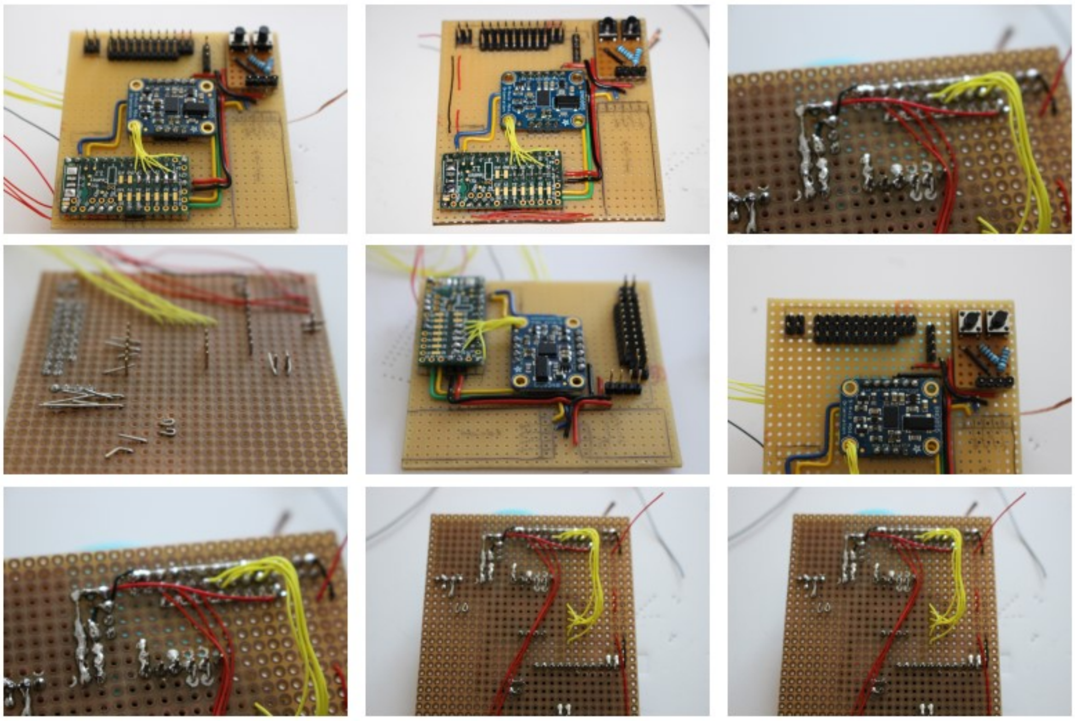
```
public string ReceiveX()
{
    if (inputQueueX.Count == 0)
        return null;

    return (string)inputQueueX.Dequeue();
} // read X data from queue and dequeue
```

Prototype Phase Four: Full System:

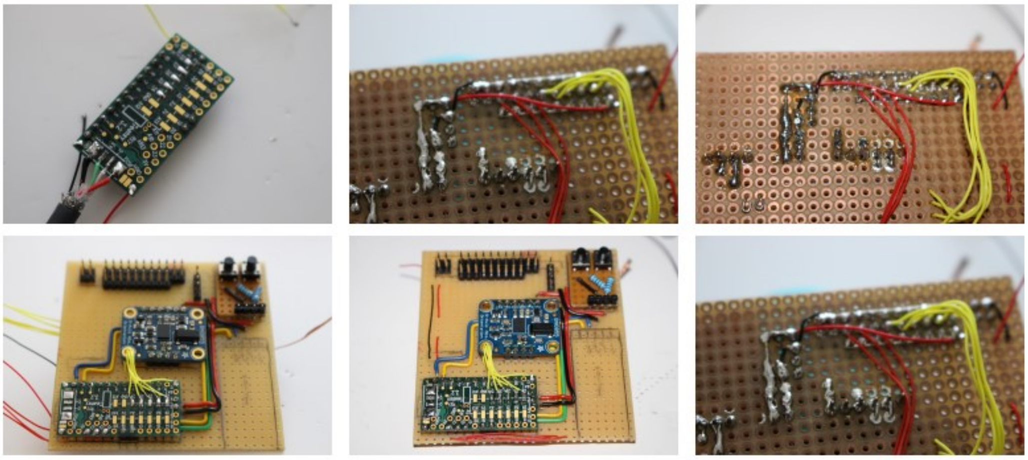
```
public string ReceiveX()
{
    if (inputQueueX.Count == 0)
        return null;

    return (string)inputQueueX.Dequeue();
} // read X data from queue and dequeue
```

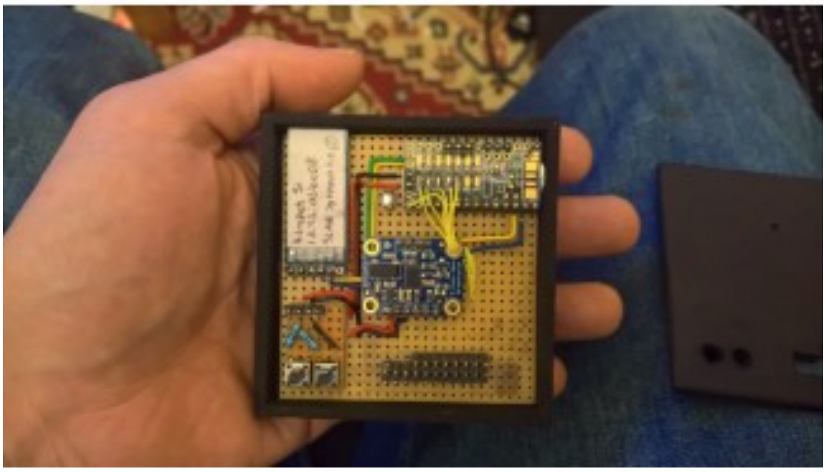
Prototype Phase Four: Full System:



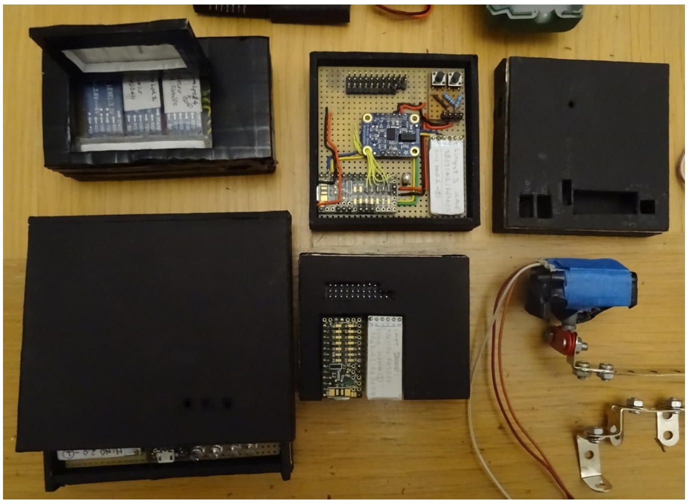


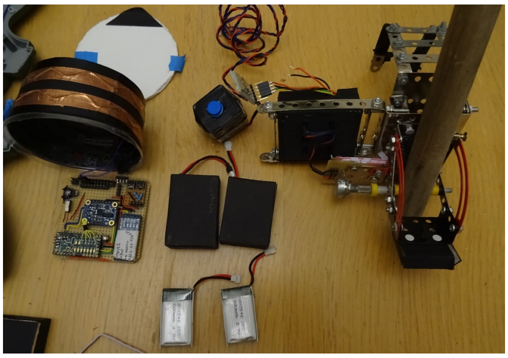
Prototype Phase Four: Full System: Cont...

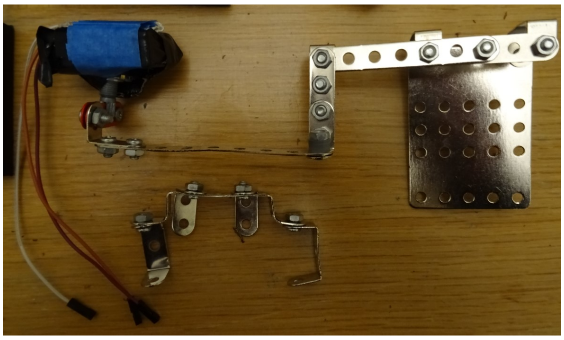




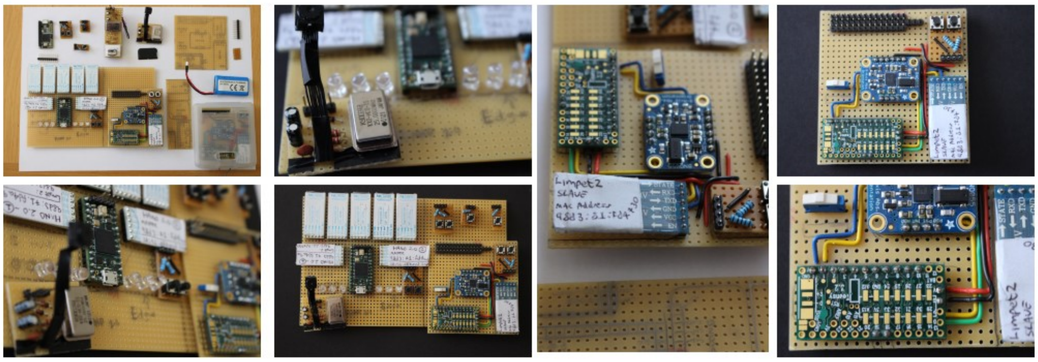
Prototype Phase Four: Full System: Cont...

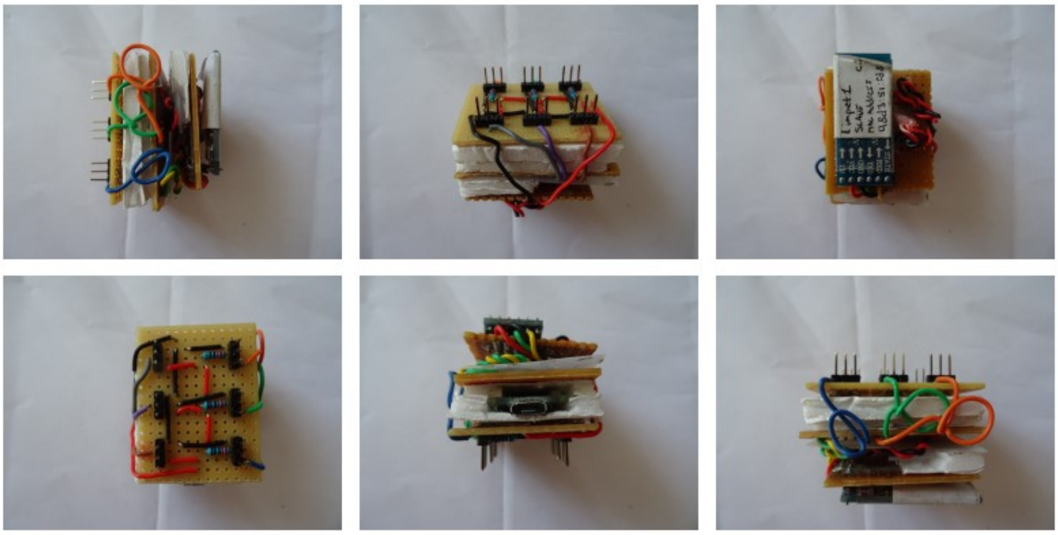


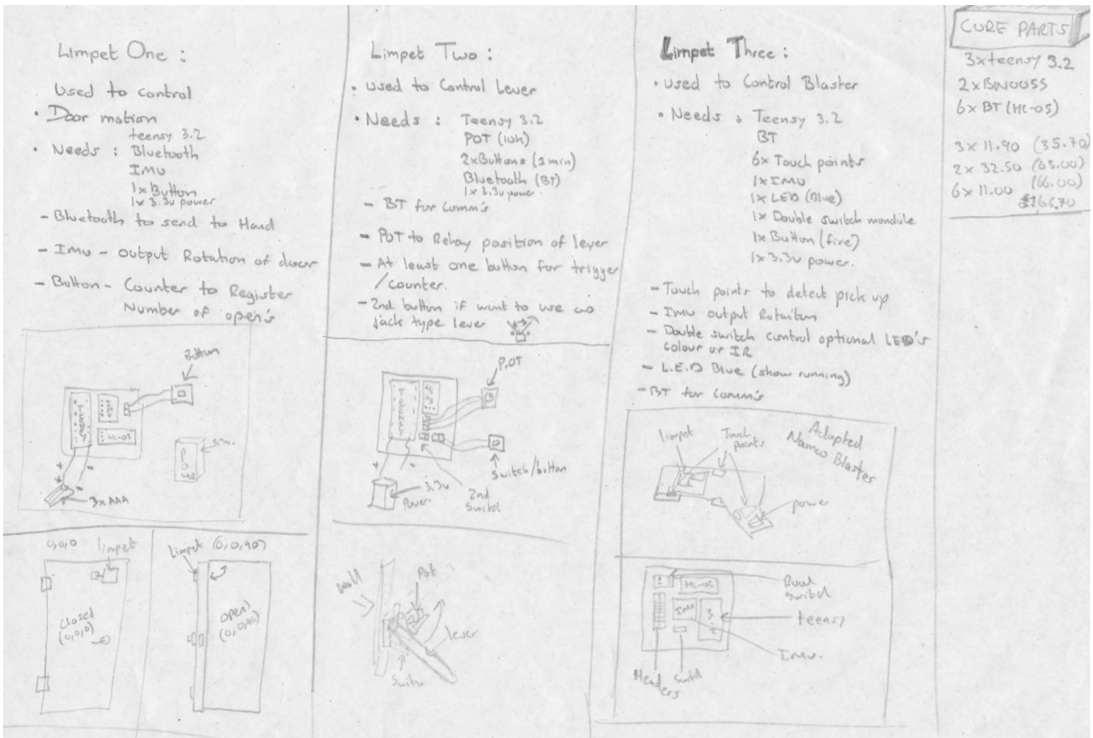




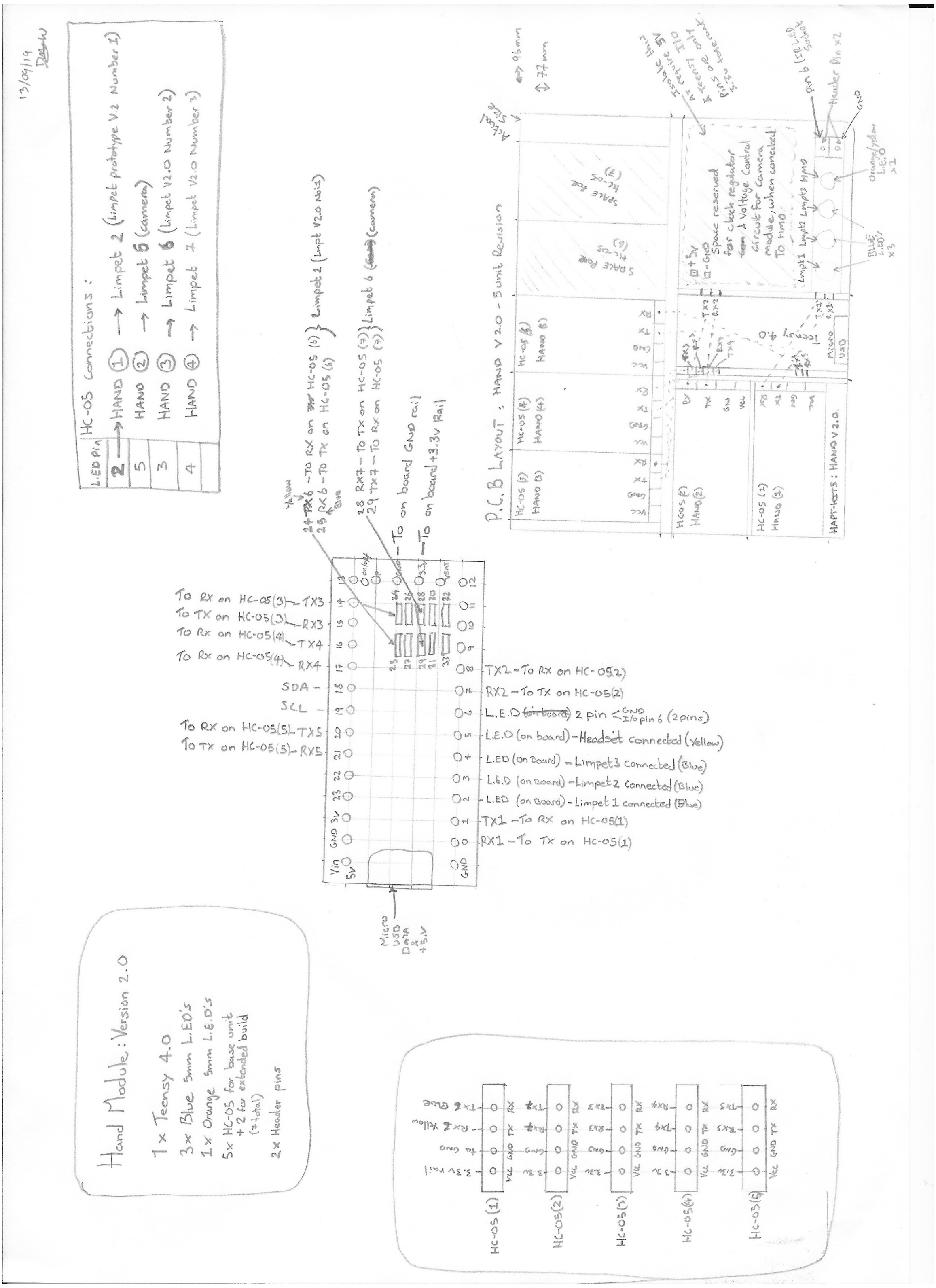
Prototype Phase Four: Full System: Cont...





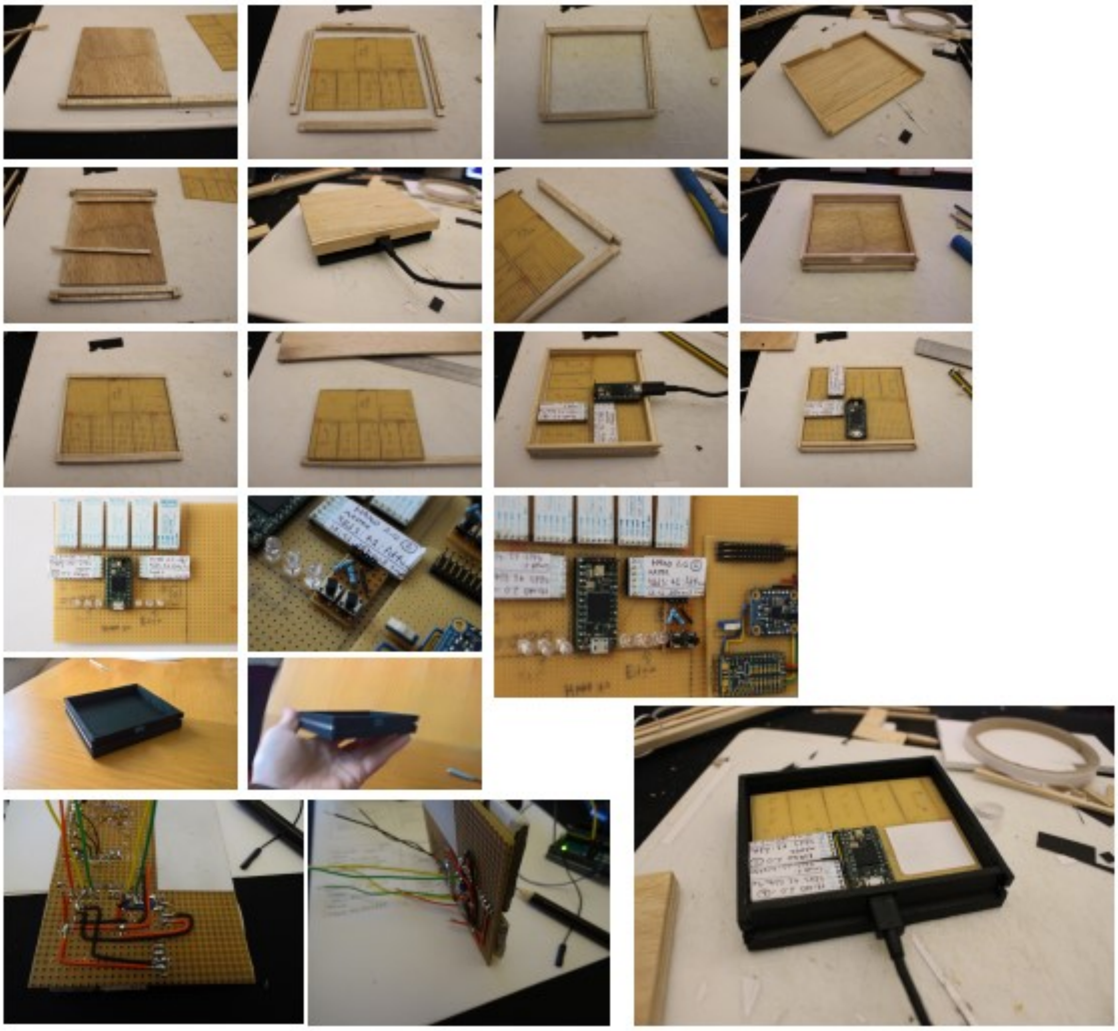


Prototype Phase Four: Full System: Cont...



Prototype Phase Four: Full System: Cont...

Prototype Phase Four: Full System: Cont...



Prototype Phase Four: Full System: Cont...

teensy.master_hand_V4_Current_4.0.ino §

```
while (Hand1_Serial.available() > 0){

  digitalWrite(lpt1_connected,HIGH);

  lpt1 = Hand1_Serial.read();

  if(recvInProgressOne == true){
  if (lpt1 != endMarker) {
            receivedData1[ndxOne] = lpt1;
            ndxOne++;
            if (ndxOne >= numChars) {
                ndxOne = numChars - 1;
            }
        }
        else {
            receivedData1[ndxOne] = '\0'; // terminate the string
            ndxOne = 0;

            lpt1Data = true;
        }
  }
  else if(lpt1 == startMarker){
    recvInProgressOne = true;
  }
}
```

```
Serial.println("<");
Serial.print(receivedData4);
Serial.print(receivedData3);
Serial.print(receivedData1);
Serial.print(receivedData2);
Serial.println(">");
```

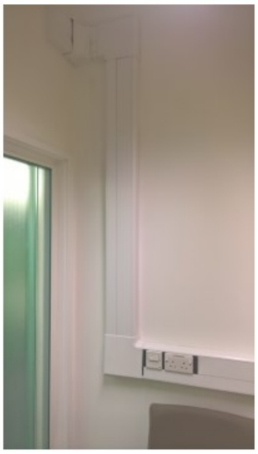
Prototype Phase Four: Full System: Cont...

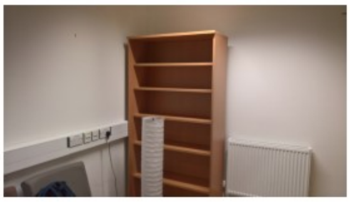
```
if (VerifyIncomingString(receivedDataLine))
{

    Words = receivedDataLine.Split(',');
          (field) string[] HandDataManager.Words
    ObjectX = float.Parse(Words[0]);
    ObjectY = float.Parse(Words[1]);
    ObjectZ = float.Parse(Words[2]);

    ObjectButton = Words[3];
    ObjectHeld = Words[4];

    LeverAng = float.Parse(Words[5]);
    LeverButton = Words[6];


    BlasterTrigger = Words[7];
    BlasterHeld = Words[8];

    DoorAng = float.Parse(Words[9]);
    DoorButton = Words[10];


}
else
{
    print("String read error");
```

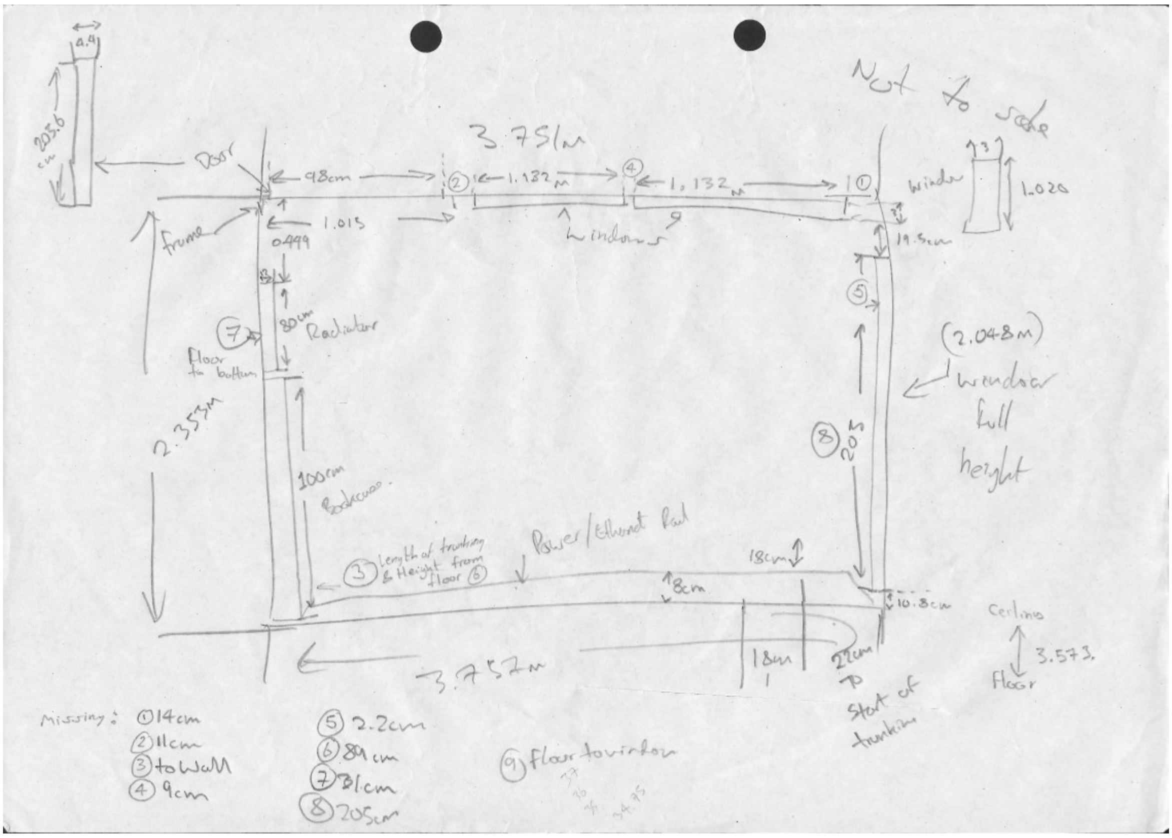
```
public bool TriggerPulled()
{
    if (BlasterTrigger == "0")
    {
        return false;
    }
    else
    {

        return true;
    }
}
```

Prototype Phase Five : Evaluation Design:

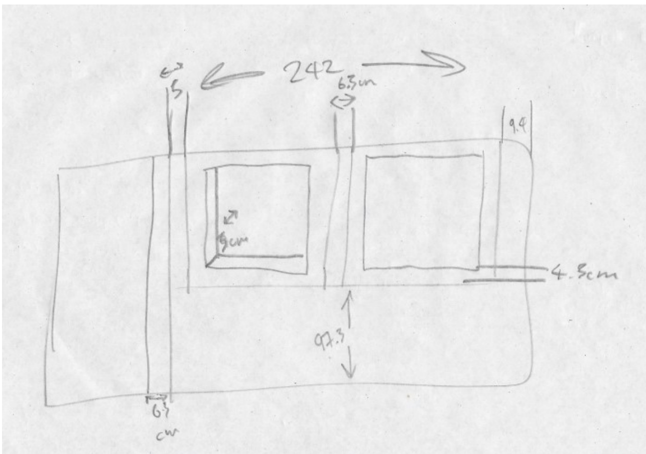


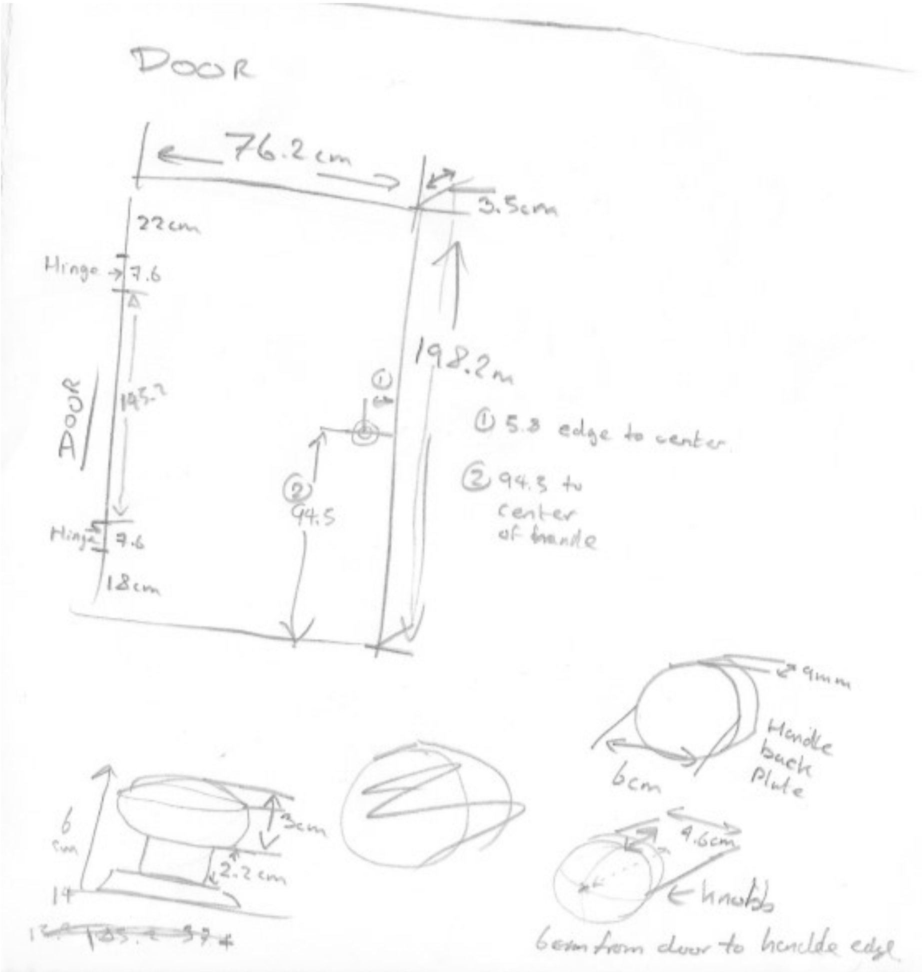





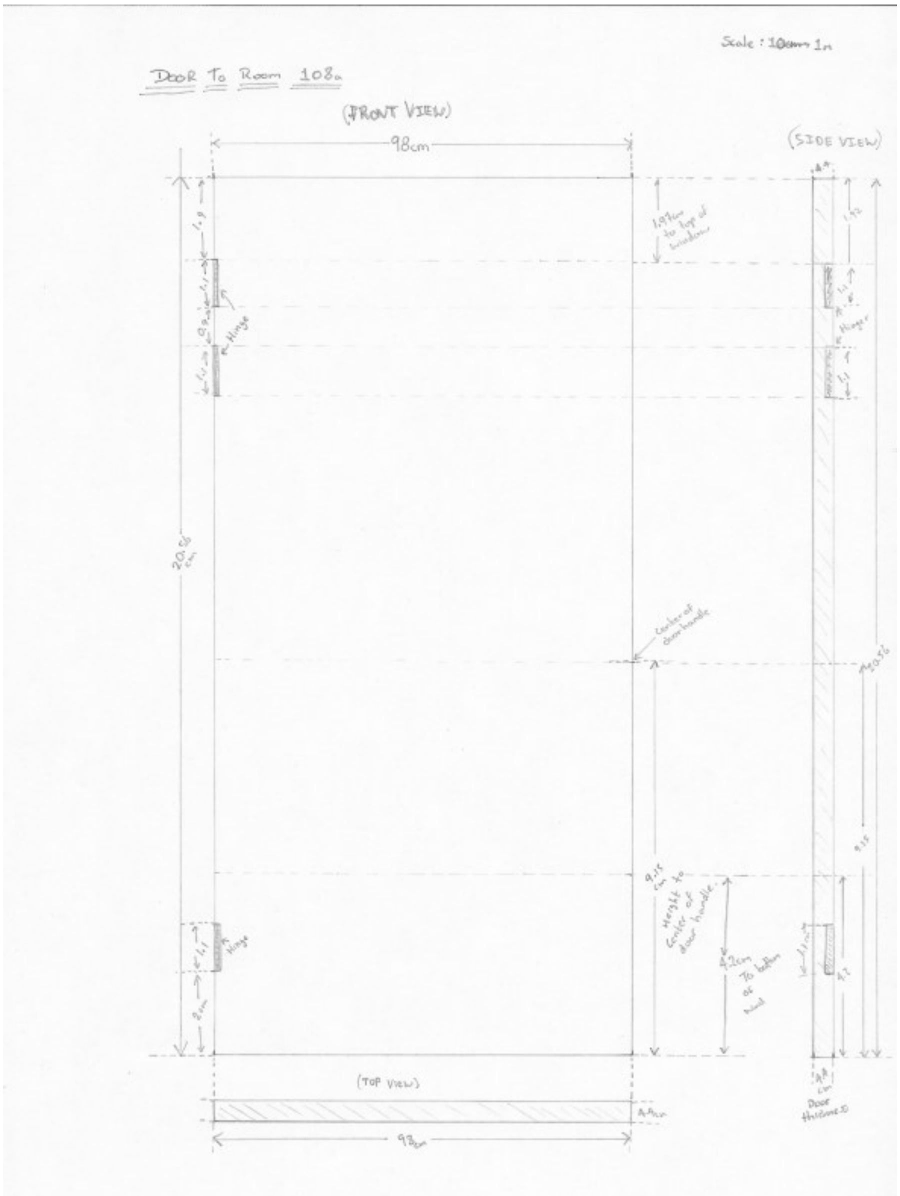



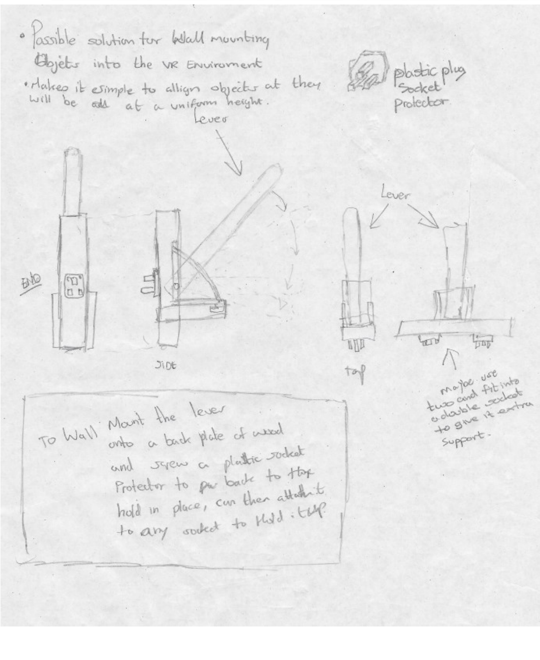
Prototype Phase Five : Evaluation Design:



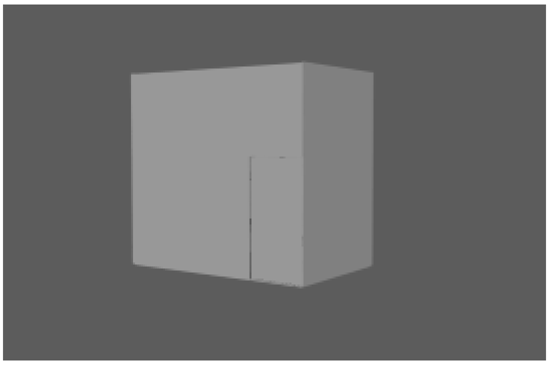


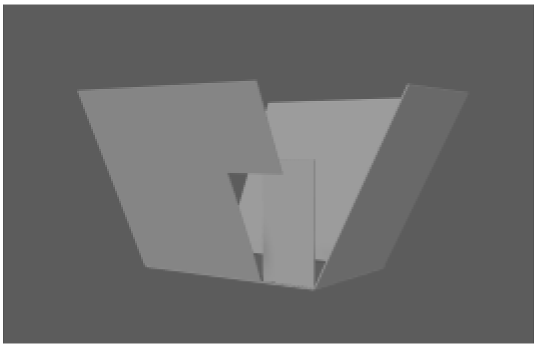
Prototype Phase Five : Evaluation Design:

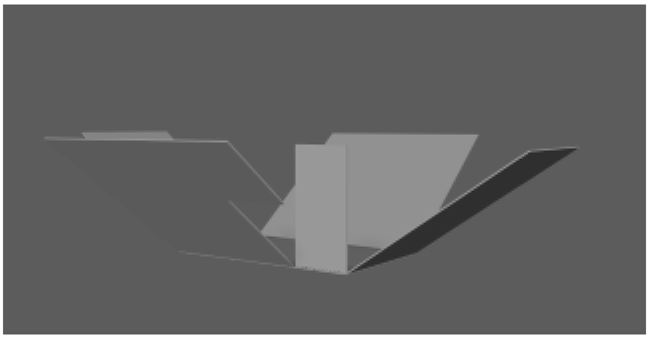


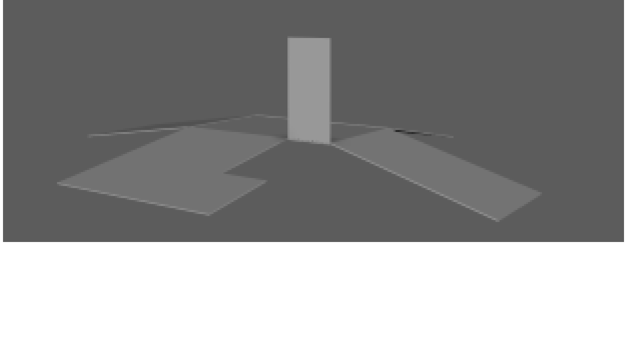


Prototype Phase Five : Evaluation Design:

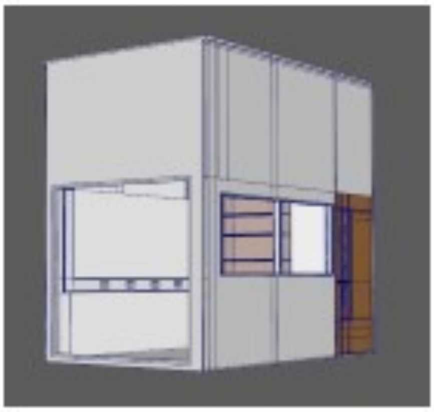


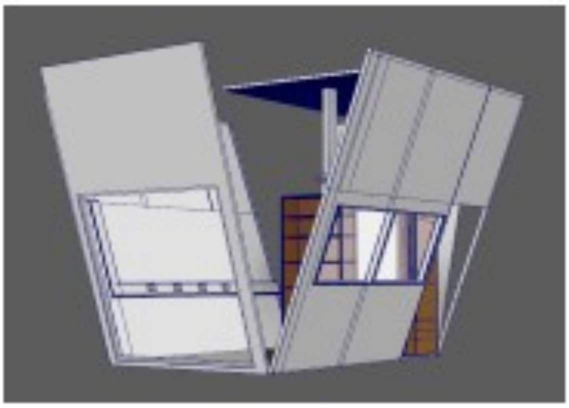


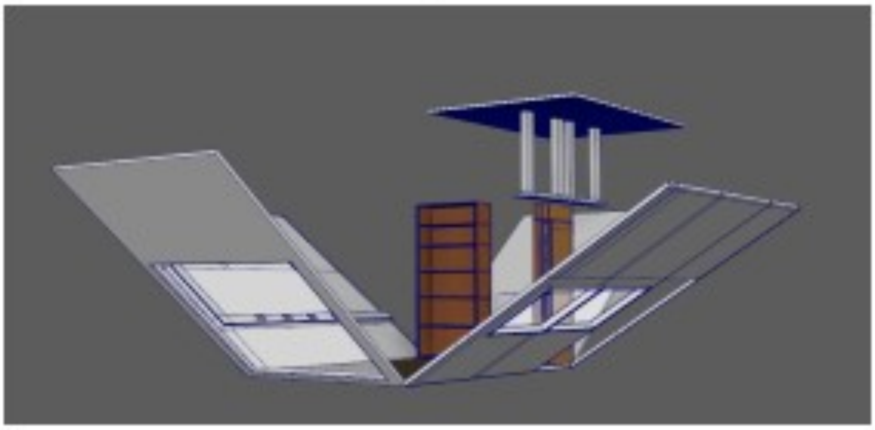


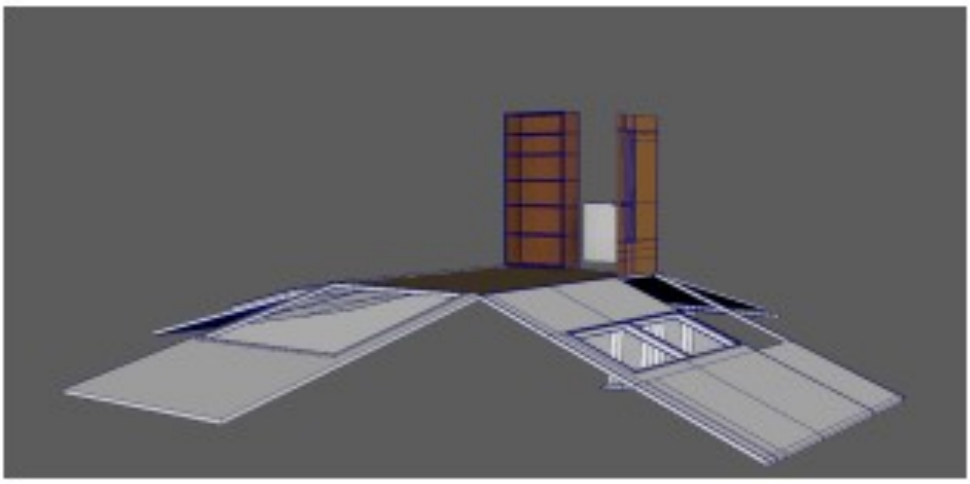


Prototype Phase Five : Evaluation Design:



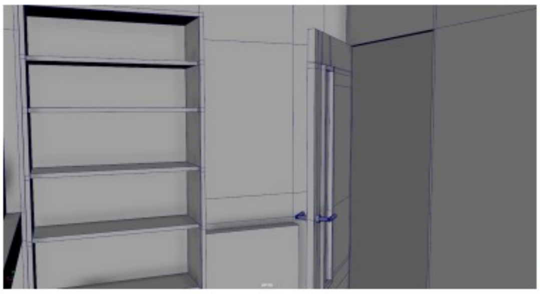
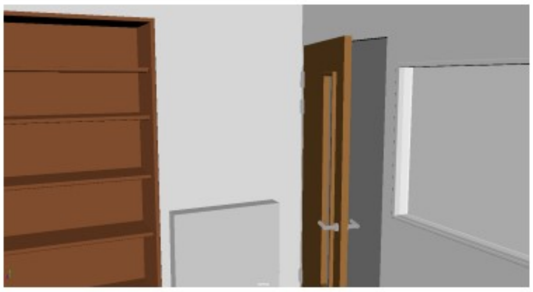
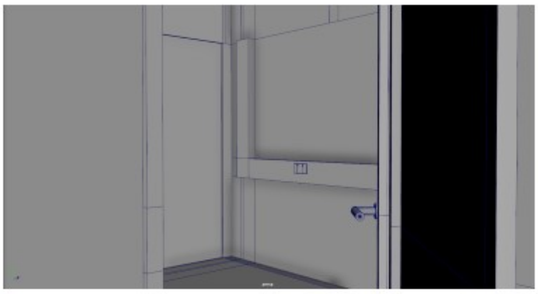
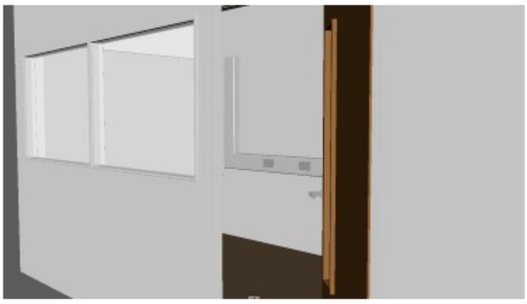




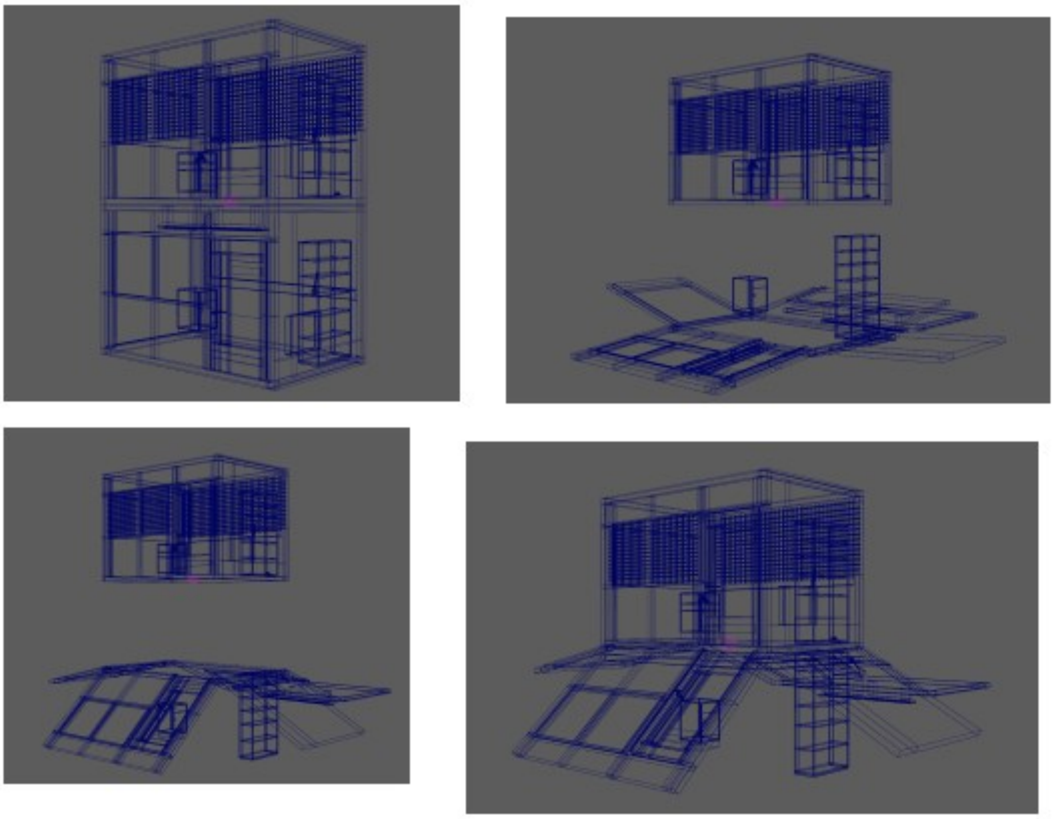


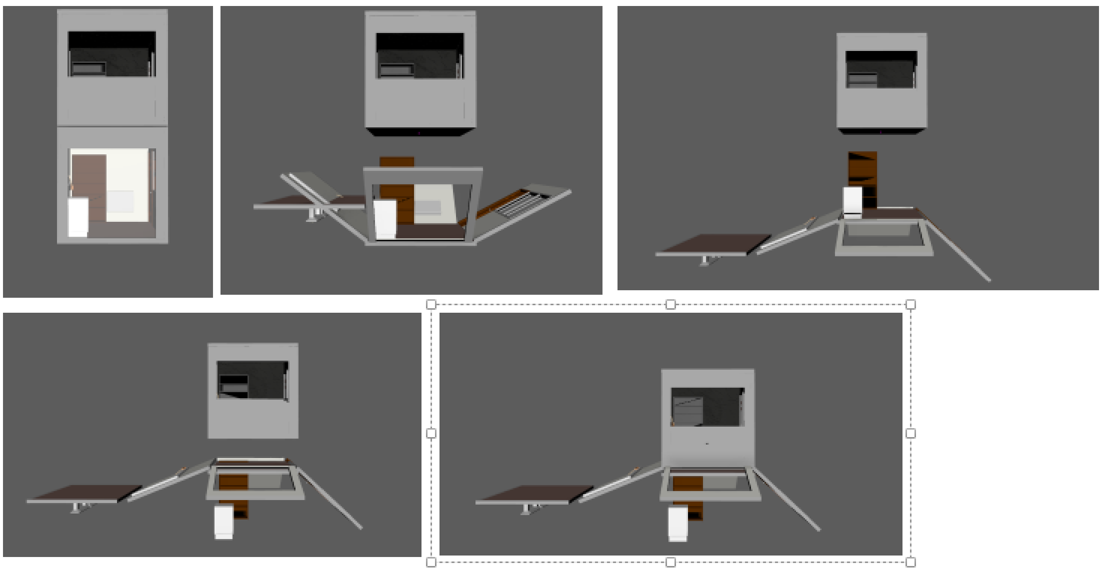
Prototype Phase Five : Evaluation Design:



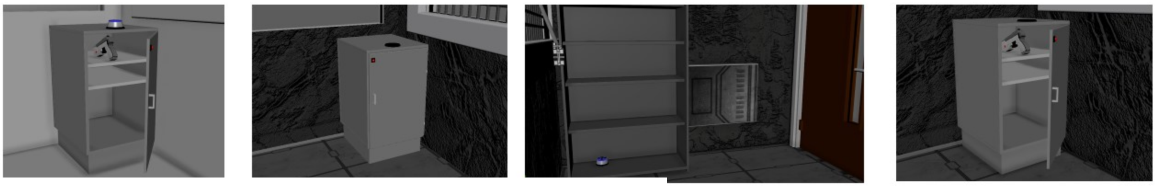





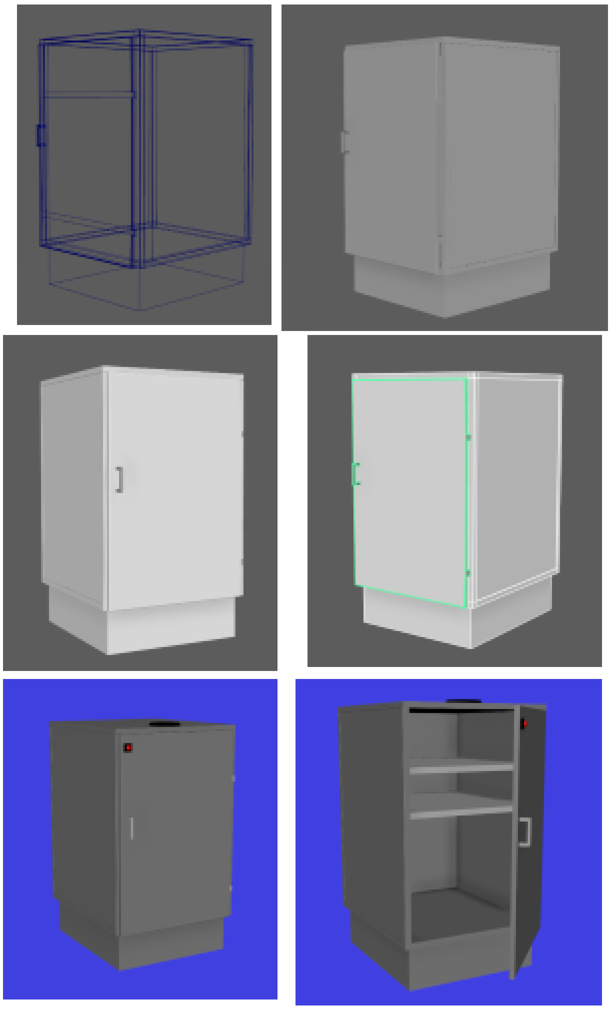

Prototype Phase Five : Evaluation Design:

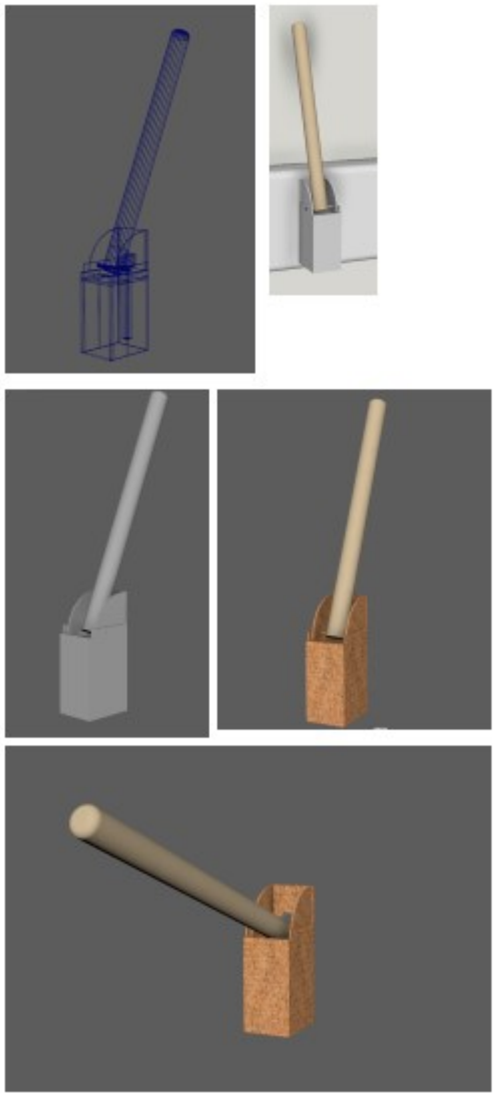


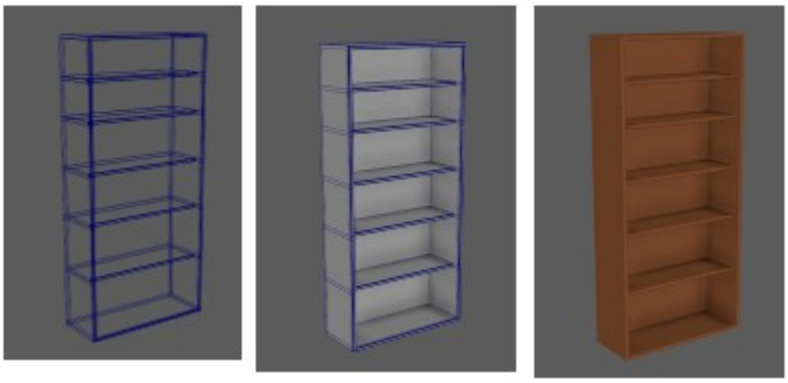


Prototype Phase Five : Evaluation Design:

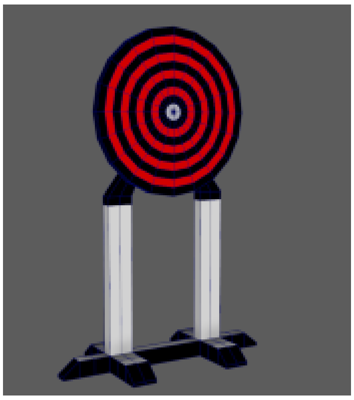


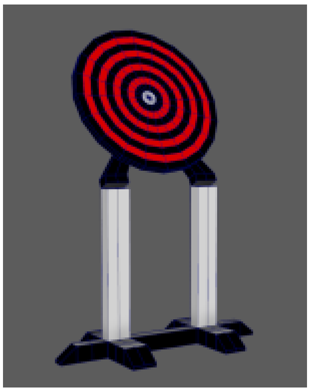


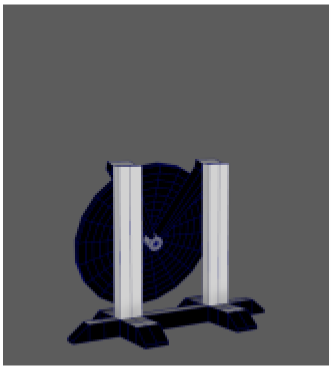


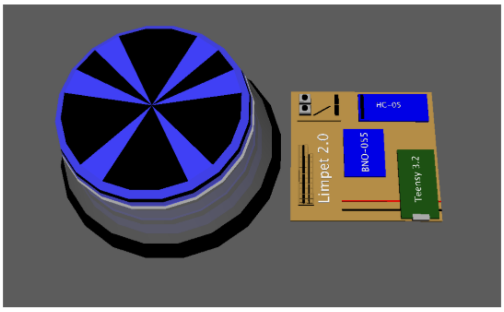


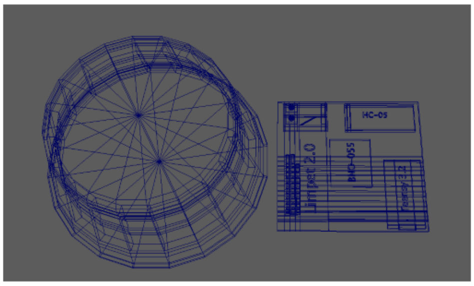
Prototype Phase Five : Evaluation Design:

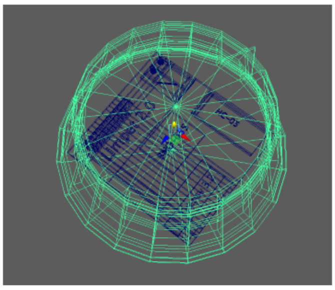


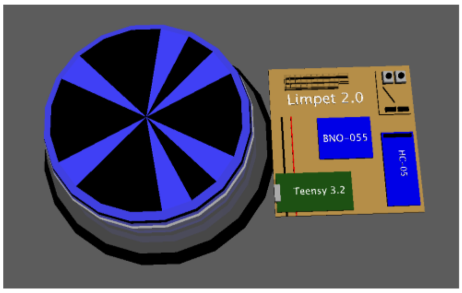




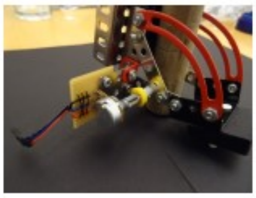
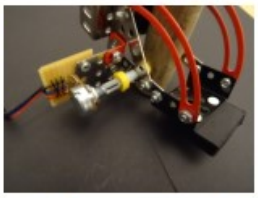
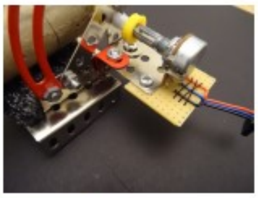
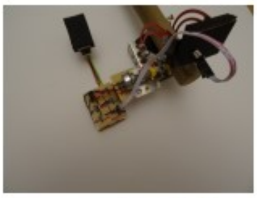




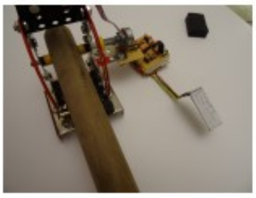


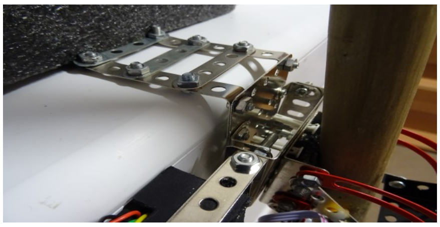
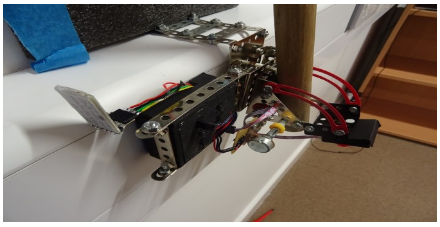
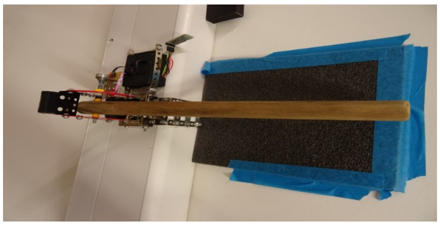
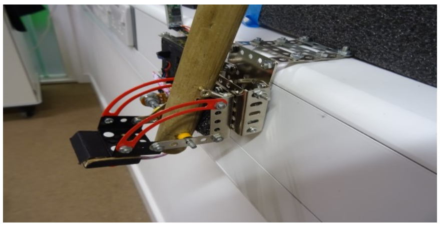
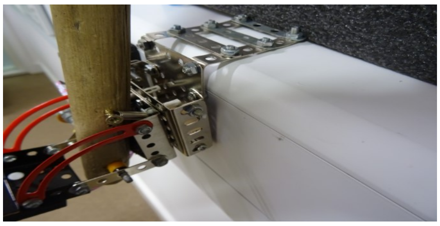
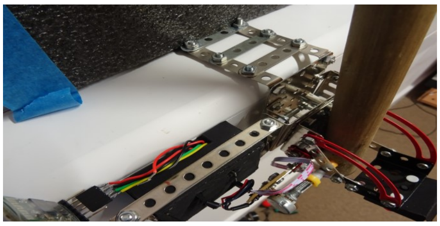
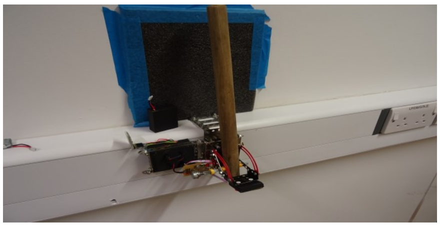


Prototype Phase Five : Evaluation Implemented:

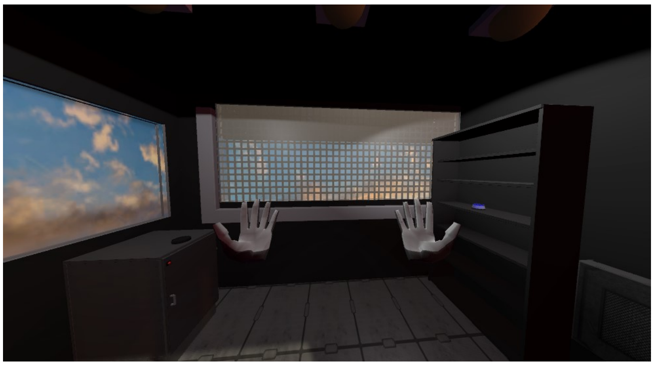






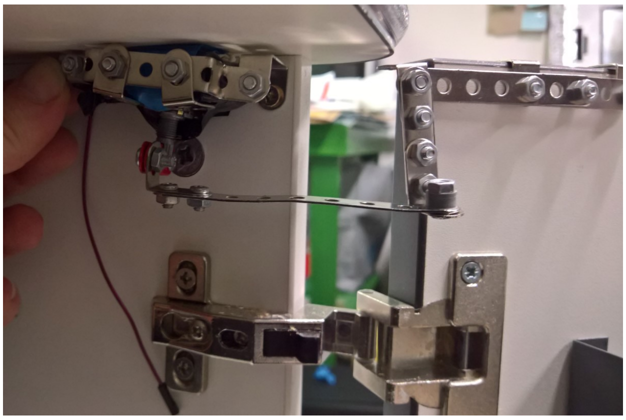
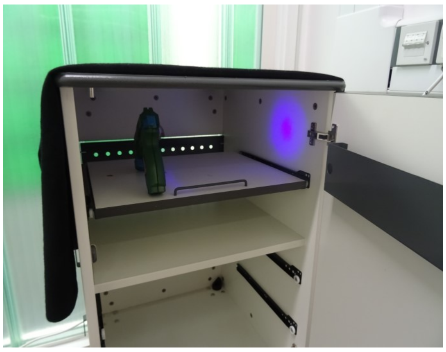


Prototype Phase Five : Evaluation Implemented:



Evaluation : Outcomes and results:

			INV1			REAL1		SP4		REAL2
Participant Number	Which Virtual Experience did you participate in first ?	Which Virtual Experience have you just participated in ?	1/How aware were you of the real world surroundings while navigating in the virtual world ? (i.e. sounds, room temperature, other people etc..)	INV1 Score	INV2 Inverted	2/How real did the Virtual World seem to you?	REAL1 Score	3/ I had a sense of acting "in" the virtual space, rather than operating something from "outside" ?	SP4 Score	4/How much did your experience in the virtual world seem consistent with your real world experience of similar activities ?
1	ITV Vive Controlle	HTV Vive Controlle	-3	0	6	1	2	-1	2	-2
2	ITV Vive Controlle	HTV Vive Controlle	2	5	1	-2	5	1	4	-2
3	ITV Vive Controlle	HTV Vive Controlle	-2	1	5	2	1	-1	2	-2
4	ITV Vive Controlle	HTV Vive Controlle	-1	2	4	-1	4	1	4	-1
5	HTV Vive Control	HTV Vive	-1	2	4	-1	4	1	4	-2
6	HTV Vive Control	HTV Vive	3	6	0	-1	4	3	6	-1
1	ITV Vive Controlle	Hap-tics System	-2	1	5	-1	4	3	6	2
2	ITV Vive Controlle	Hap-tics System	2	5	1	-1	4	2	5	1
3	ITV Vive Controlle	Hap-tics System	-1	2	4	-1	4	2	5	2
4	ITV Vive Controlle	Hap-tics System	-1	2	4	-2	5	2	5	1
5	HTV Vive Control	Hap-tics System	-2	1	5	-2	5	2	5	2
6	HTV Vive Control	Hap-tics System	3	6	0	-2	5	2	5	2
7	Hap-tics System	Hap-tics System	2	5	1	-2	5	3	6	2
8	Hap-tics System	Hap-tics System	-2	1	5	0	3	0	3	3
9	Hap-tics System	Hap-tics System	-3	0	6	1	2	3	6	0
10	Hap-tics System	Hap-tics System	1	4	2	2	1	1	4	-1
11	Hap-tics System	Hap-tics System	-2	1	5	2	1	0	3	-3
12	Hap-tics System	Hap-tics System	2	5	1	-1	4	0	3	-1
7	Hap-tics System	HTV Vive	-2	1	5	-1	4	2	5	2
8	Hap-tics System	HTV Vive	2	5	1	-1	4	0	3	-2
9	Hap-tics System	HTV Vive	-2	1	5	1	2	-1	2	0
10	Hap-tics System	HTV Vive Controlle	-1	2	4	-1	4	1	4	-2
11	Hap-tics System	HTV Vive Controlle	-1	2	4	2	1	0	3	0
12	Hap-tics System	HTV Vive Controlle	2	5	1	2	1	3	6	-3

Evaluation : Outcomes and results: Cont...

	REAL3		SP3		INV2		G1		SP1		SP5		INV3	
REAL2 Score	5/How real did the Virtual World seem to you?	REAL3 Score	6/ I did not feel present in the virtual space ?	SP3 Score	7/ I was not aware of my real environment ?	INV2 Score	8/ In the Computer Generated World I had a Sense of "Being There" ?	G1 Score	9/ Somehow I felt that the Virtual world surrounded me ?	SP1 Score	10/I felt present in the virtual space ?	SP5 Score	11/I still paid attention to the real world environment ?	INV3 Score
1	-2	1	1	4	-2	1	2	5	2	5	1	4	2	1
1	-2	1	1	4	-1	2	1	4	1	4	1	4	-1	4
1	-2	1	1	4	-2	1	1	4	1	4	1	4	1	2
2	-2	1	1	4	-1	2	-1	2	1	4	1	4	1	2
1	0	3	1	4	-1	2	2	5	2	5	2	5	2	1
2	-3	0	2	5	3	6	2	5	1	4	3	6	-3	6
5	-2	1	2	5	2	5	2	5	2	5	2	5	2	1
4	-1	2	2	5	-1	2	2	5	1	4	2	5	1	2
5	-2	1	2	5	-2	1	2	5	1	4	2	5	-2	5
4	-1	2	2	5	-2	1	2	5	3	6	2	5	2	1
5	2	5	2	5	-3	0	3	6	3	6	2	5	3	0
5	2	5	3	6	3	6	3	6	3	6	3	6	-3	6
5	-1	2	2	5	-1	2	3	6	3	6	3	6	-2	5
6	-3	0	3	6	-3	0	1	4	3	6	3	6	3	0
3	-1	2	2	5	-1	2	2	5	3	6	2	5	-1	4
2	1	4	3	6	-1	2	2	5	3	6	3	6	1	2
0	2	5	-3	0	-3	0	0	3	0	3	0	3	3	0
2	-2	1	-3	0	3	6	2	5	1	4	3	6	0	3
5	0	3	2	5	-2	1	1	4	2	5	2	5	2	1
1	-1	2	2	5	-2	1	0	3	1	4	1	4	2	1
3	0	3	1	4	-2	1	1	4	1	4	1	4	1	2
1	0	3	1	4	-2	1	0	3	-1	2	-1	2	0	3
3	0	3	-3	0	-2	1	2	5	1	4	1	4	3	0
0	-2	1	1	4	3	6	-2	1	1	4	2	5	0	3

Evaluation : Outcomes and results: Cont...

12/The Virtual World seemed more realistic than the real world ?	REAL4 Score	13/I felt like I was just perceiving pictures?	SP2 Score	14/I was completely captivated by the virtual world.	INV4 Score	INV MEAN	SP MEAN	REAL MEAN	INV MEAN with Inverted Q1
-3	0	-3	6	2	5	1.75	4.2	1	3.25
-3	0	-1	4	-1	2	3.25	4	1.75	2.25
-2	1	-2	5	2	5	2.25	3.8	1	3.25
-2	1	-2	5	1	4	2.5	4.2	2	3
3	6	-3	6	1	4	2.25	4.8	3.5	2.75
-3	0	-3	6	2	5	5.75	5.4	1.5	4.25
-2	1	3	0	2	5	3	4.2	2.75	4
-2	1	-3	6	1	4	3.25	5	2.75	2.25
-1	2	-3	6	1	4	3	5	3	3.5
-1	2	-3	6	1	4	2	5.4	3.25	2.5
-2	1	-3	6	2	5	1.5	5.4	4	2.5
-1	2	-3	6	3	6	6	5.8	4.25	4.5
0	3	-3	6	2	5	4.25	5.8	3.75	3.25
-3	0	-3	6	-2	1	0.5	5.4	2.25	1.5
-2	1	-2	5	1	4	2.5	5.4	2	4
-1	2	-2	5	2	5	3.25	5.4	2.25	2.75
-3	0	1	2	-3	0	0.25	2.2	1.5	1.25
-3	0	1	2	3	6	5	3	1.75	4
-1	2	-2	5	2	5	2	5	3.5	3
-3	0	-3	6	1	4	2.75	4.4	1.75	1.75
-2	1	-2	5	1	4	2	3.8	2.25	3
-2	1	0	3	-2	1	1.75	3	2.25	2.25
-3	0	0	3	0	3	1.5	2.8	1.75	2
-3	0	1	2	3	6	5	4.2	0.5	4

Participants 1-6 experienced the Vive First.
Participants 7-12 experienced the Haptics First

HTC VIVE (ENV1)
HAPTICS (ENV2)

Spacial Prescence	SP
Involvment	INV
Experienced Realism	REAL

result	score
-3	0
-2	1
-1	2
0	3
1	4
2	5
3	6

V.R Immersion Evaluation: Information Sheet

Participant Information Sheet –Anonymous Research

Project background

The University of York would like to invite you to take part in the following project: Evaluation of Low-cost Haptics for Virtual Environments

Before agreeing to take part, please read this information sheet carefully and let us know if anything is unclear or you would like further information.

What is the purpose of the project?

This project is being performed by Dan Wonnacott who is a postgraduate student studying MRES MSc Interactive Media at the University of York. This research is being conducted to form part of the evaluation section of his thesis research.
The research for this project is being conducted according to restrictions that have been subject to approval by the TFTI Ethics committee. The Chair of the TFTI Ethics committee can be contacted on TFTI-ethics@york.ac.uk.
For this research project, I am interested in assessing the impact of using a handheld controller v's natural touch sense on participants perceived immersion and sense of presence in a VR environment. Your participation in this project will involve you playing a simple VR test scenario using the HTV VIVE. You will be asked to complete the environment twice, once interacting using the VIVE controllers and once using leap motions hand tracking technology. Within the scenario you will be required to simply interact, opening and closing draws, doors and cupboards, and push and pull buttons and levers. The game will automatically collect data regarding how you have played the game, and screen and audio capture you are playing through it. Whilst playing you may be asked to pause so that I can photograph a specific interaction, please note I will only be recording hand position and pose, you will not be identifiable from these pictures. Upon completion of the VR experience you will then be asked to complete an anonymous online survey, that aims to asses your level of immersion in the experience.

Please note that to comply with the approved Ethics requirements of this work, we do not intend to discuss sensitive topics with you that could be potentially upsetting or distressing. If you have any concerns about the topics that may be covered in the research study, please raise these concerns with the researcher.
Your participation in this project is voluntary. If you wish, we will provide you with access to the edited film and/or the report that we submit after our marks have been confirmed. If you would like to receive access to these, you can indicate as such on the consent form.

Why have I been invited to take part?

You have been invited to take part because we are aiming to recruit a diverse group of participants, and we hope you might be interested in this work.

Immersion Presence Questionnaire: part 1

Please ask me if I've forgotten to give you a participant number!!

*Required

1. Participant Number *

____

2. Which Virtual Experience did you participate in first ? *

Mark only one oval.

- ( ) HTV Vive Controller
- ( ) Hap-tics System

3. Which Virtual Experience have you just participated in ? *

Mark only one oval.

- ( ) HTV Vive Controller
- ( ) Hap-tics System

4. 1/How aware were you of the real world surroundings while navigating in the virtual world ? (i.e. sounds, room temperature, other people etc..) *

Mark only one oval.

- -3 Extremely Aware
- -2
- -1
- 0
- 1
- 2
- 3 Not Aware At All

5. 2/How real did the Virtual World seem to you? *

Mark only one oval.

- -3 Completely Real
- -2
- -1
- 0
- 1
- 2
- 3 Not Real At All

6. 3/ I had a sense of acting "in" the virtual space, rather than operating something from "outside" ? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Full Agree

7. 4/How much did your experience in the virtual world seem consistent with your real world experience of similar activities ? *

Mark only one oval.

- -3 not consistent
- -2
- -1
- 0
- 1
- 2
- 3 very consistent

8. 5/How real did the Virtual World seem to you? *

Mark only one oval.

- -3 about as real as an imagined world
- -2
- -1
- 0
- 1
- 2
- 3 indistinguishable from the real world

9. 6/ I did not feel present in the virtual space ? *

Mark only one oval.

- -3 Did not feel
- -2
- -1
- 0
- 1
- 2
- 3 Felt Present

10. 7/ I was not aware of my real environment ? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Fully Agree

11. 8/ In the Computer Generated World I had a Sense of "Being There" ? *

Mark only one oval.

- -3 Not at all
- -2
- -1
- 0
- 1
- 2
- 3 Very Much

12. 9/ Somehow I felt that the Virtual world surrounded me ? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Fully Agree

13. 10/I feet present in the virtual space ? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Full Agree

14. 11/I still paid attention to the real world environment ? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Fully Agree

15. 12/The Virtual World seemed more realistic than the real world ? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Fully Agree

16. 13/I felt like I was just perceiving pictures? *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Fully Agree

17. 14/I was completely captivated by the virtual world. *

Mark only one oval.

- -3 Fully Disagree
- -2
- -1
- 0
- 1
- 2
- 3 Fully Agree