Design Journal

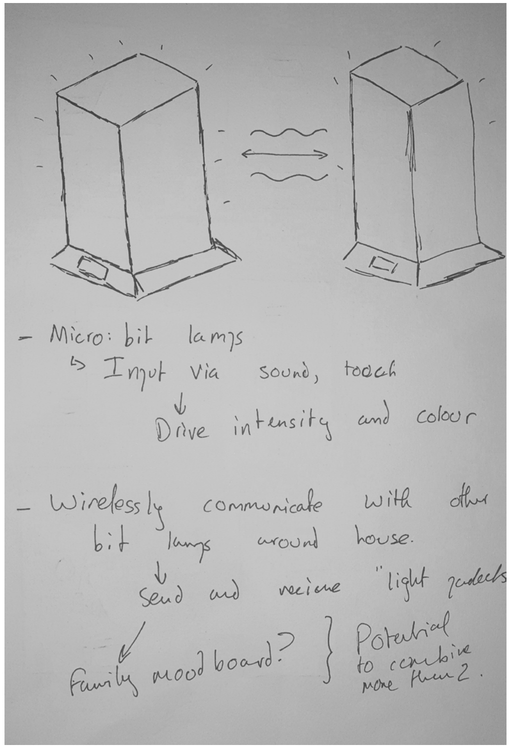
Soren Saville Scott
ssav7912

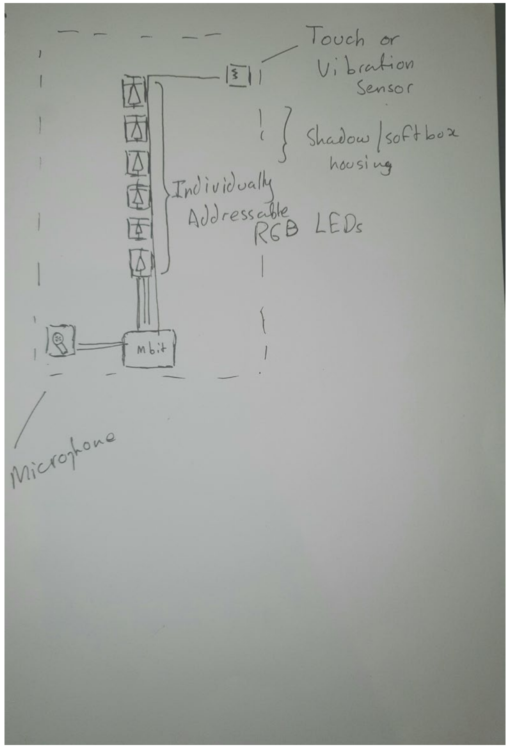
This journal documents the design iteration process for an interactive wireless lamp system that intends to address the problem of isolation within the household unit, by fostering interaction and communication between members of a given household. The design is implemented in accordance with ludic design principles, a design paradigm that reasons that curiosity and play - 'ludic' activities - are necessary aspects of human environments that can and should be intentionally designed for (Gaver et. al, 2004).

Research and Pre-production

The initial stages of the design process involved a depth and breadth of research that aimed to synthesise the briefs requirements of ludic design principles with the posed problem of isolation and interaction within households. One of the key takeaways from this research was that one of the key drivers of ludic activity was its social dimension, that such playful activities were most compelling or emergent among multiple people interacting with the design. As such, for the proposed design, it was reasoned that in order to encourage ludic interaction between household members, it should involve a distribution or slicing of affordances among multiple users, so that no single user on their own can utilise the entire functionality of the device. This helps encourage users to work collaboratively to construct a model of interaction for themselves, alongside obfuscating any purely ‘functional’ uses for the design, encouraging a more ludic mode of use.

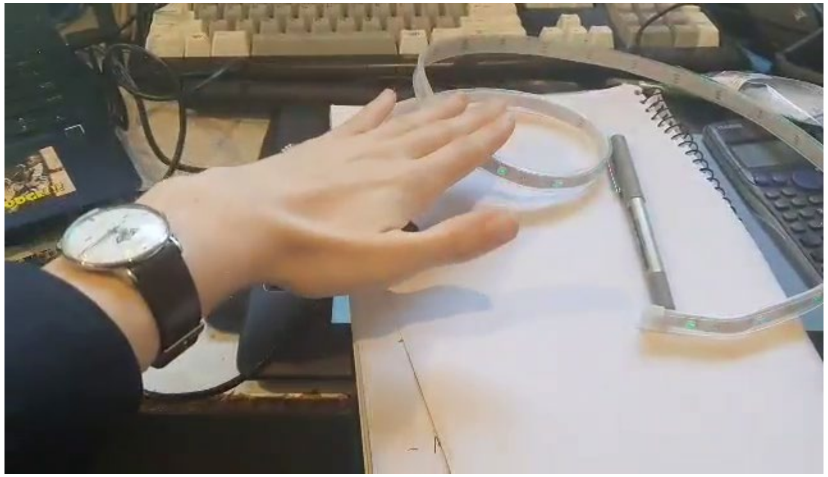
Conclusions from wider research showed that within household units, the large part of communication was concerned with coordinating household tasks (Crabtree et al., 2003), and that many attempts at digital devices that help facilitate this goal were ineffective due to their inability to overcome the ease and simplicity of face-to-face interaction(Hutchinson et al., 2003). As such, it was reasoned that a proposed design should, in order to encourage ludic interaction with it, restrict the resolution or granularity of the mode of communication between household members, encouraging users to again invent and explore models of interaction for themselves.

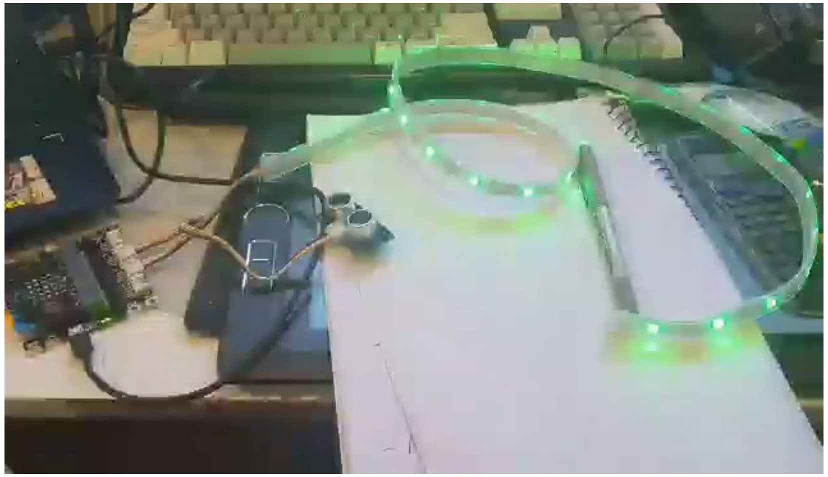




Some early sketches of the proposed design - a wireless lamp system that takes inputs from multiple users and communicates its state between itself, in order to facilitate communication between different areas of a household.

Early Prototyping





Initial prototypes made use of substitute components to test feasibility of design and experiment with proposed modes of interaction.

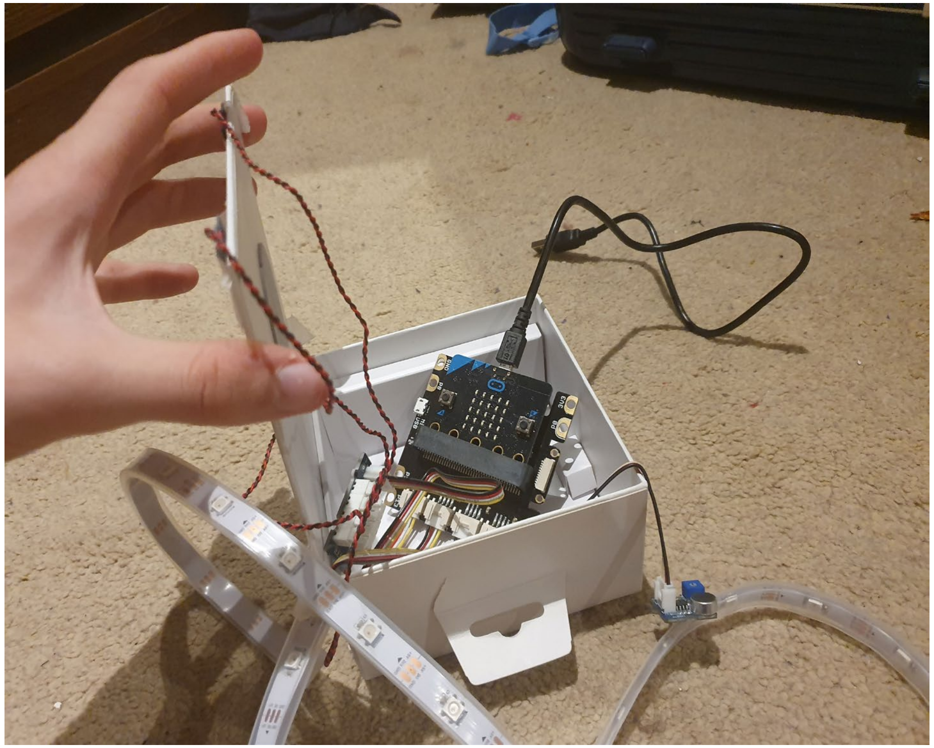
This made use of an ultrasonic ranger to modulate the intensity of pixels in a neopixel strip, demonstrating one of the basic proposed modes of interaction.

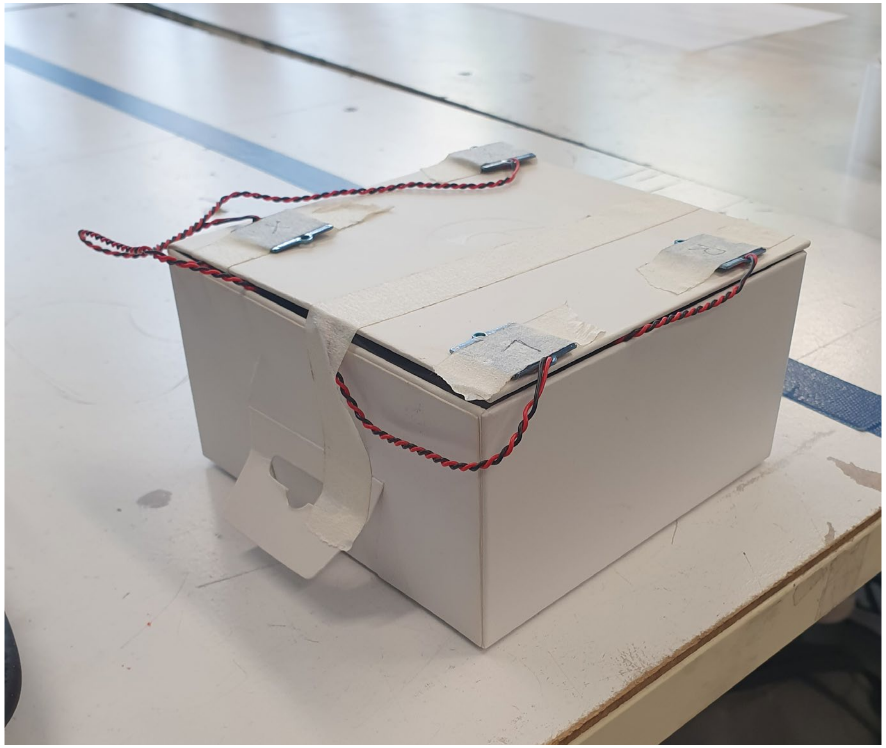
```
def hsvToRgb(hue, sat, val):
    # TO_OPTIMISE:
    # rewrite to use ints only? Assume 8 bits for HSV value...
    # Look into: https://www.vagrearg.org/content/hsvrgb

    magicnums = [5, 3, 1] #"magic" numbers to get correct values for each channel.
    RGB = [0, 0, 0]

    for i in range(3):
        n = magicnums[i]
        k = (n + (hue/60)) % 6
        RGB[i] = val - val * sat * max(0, min(k, 4 - k, 1))
        RGB[i] = trunc(remap(RGB[i], 0, 1, 0, 255)) #remaps and truncates floats into ints between 0 - 255, for compatibility and speed.
    return tuple(RGB)
```

This Implementation is based on an example from: https://en.wikipedia.org/wiki/HSL_and_HSV

The strip also continually shifted hues over time. Both the hue-shift and intensity modulation was accomplished with a custom function that translates between a Hue-Saturation-Value model of colour and the Red-Green-Blue model that the neopixel drivers expect. This would go on to form the interface for all human interaction with colour in this project, as the HSV model tends to be a more intuitive or perceptual representation of colour for humans, based on artistic concepts of hue, tint and shade (Smith, 1978).

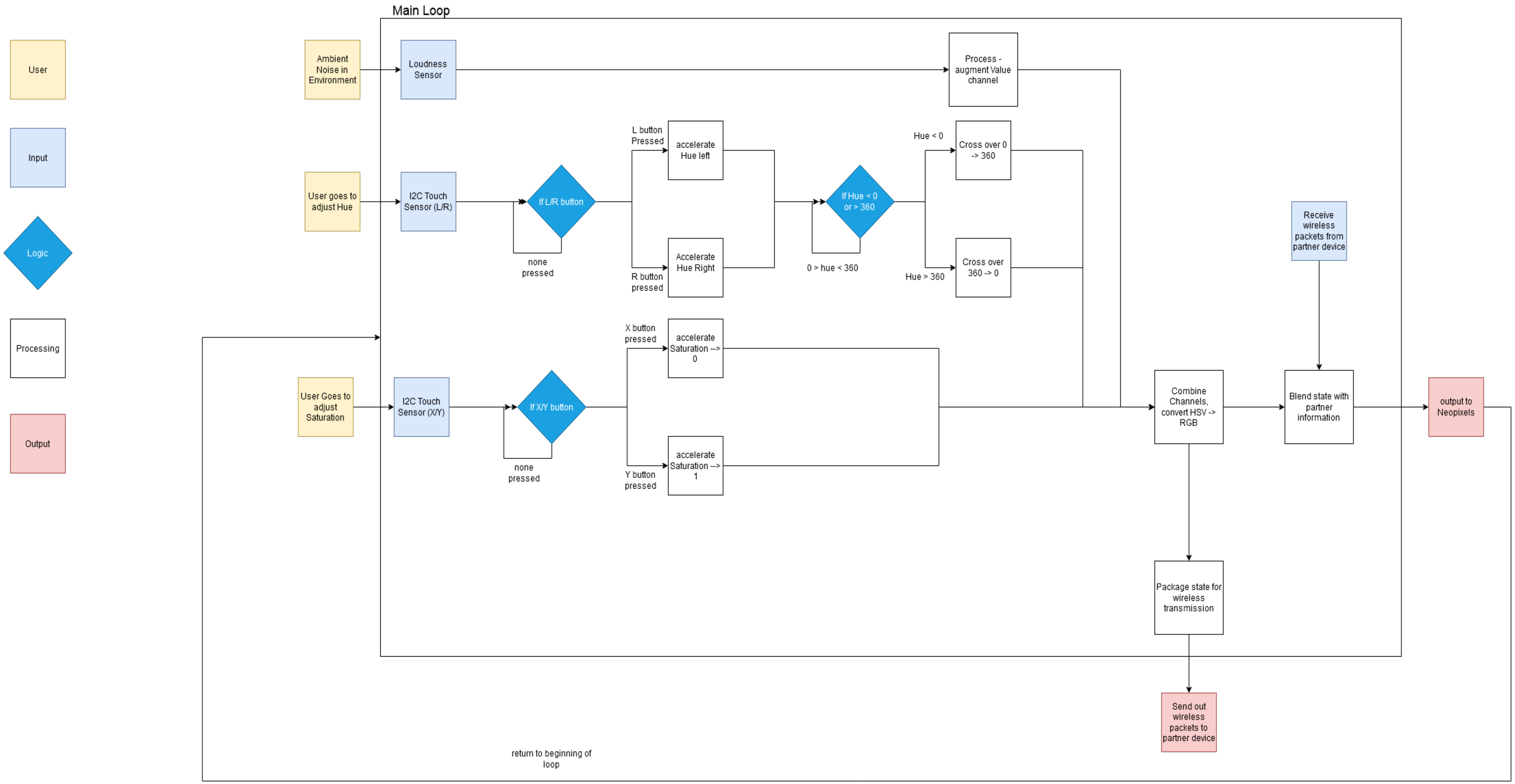
Real Component Prototypes





In this stage, substitute components were swapped for their actual counterparts, and a more complete model of interaction was developed. These new components included a 4-pad I2C touch sensor array and a microphone. The microphone was intended to modulate the brightness of pixels, as an ambient component to the design that can be used to automatically transmit information like human presence, which in turn could function as the equivalent of a “handshake” (IETF, 1981) and signal to other users a readiness to begin communicating/interacting. The touch sensors, in turn, would be used to control cylinder positions of the hue and saturation channels respectively, allowing for a full 3 dimensional traversal of the colour model.

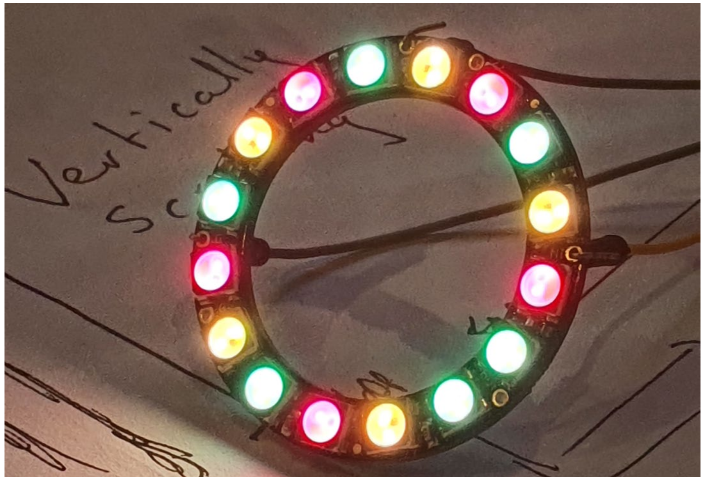
In order to make the touch sensors function a custom library had to be written, ported from the Adafruit arduino implementation. This port can be found here: https://github.sydney.edu.au/ssav7912/deco1013-microbit/blob/master/lib/i2ctouchsensor.py



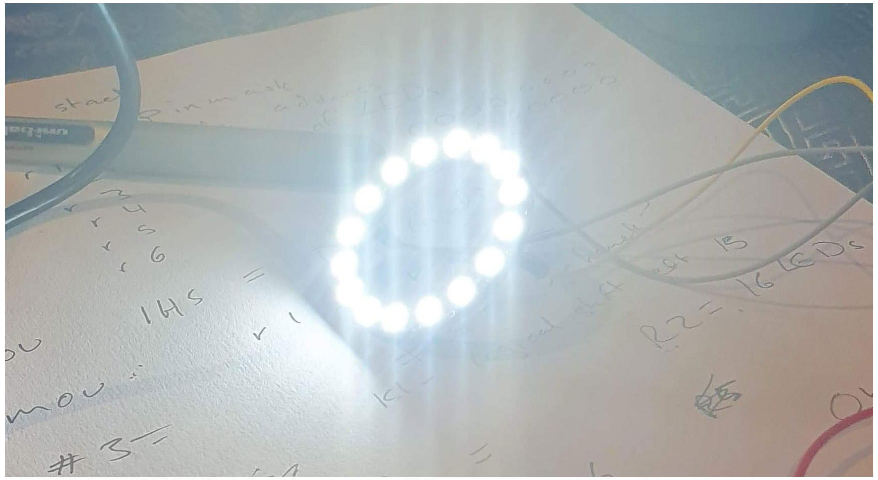
The proposed interaction model at this particular stage of the project.

NeoPixel Ring Modification

It was found that one of my intended components - a 16 pixel WS2812 neopixel ring - was incompatible with the implementation of the neopixel library in the default micropython runtime. This was due to it having an extra channel for reproduction of white colours, while the neopixel library assumed only 3 channels per pixel. This led to over/undershoot errors when writing data to the array, causing strange patterns and an essentially unusable component.



Strange colours produced by the channel mismatch.



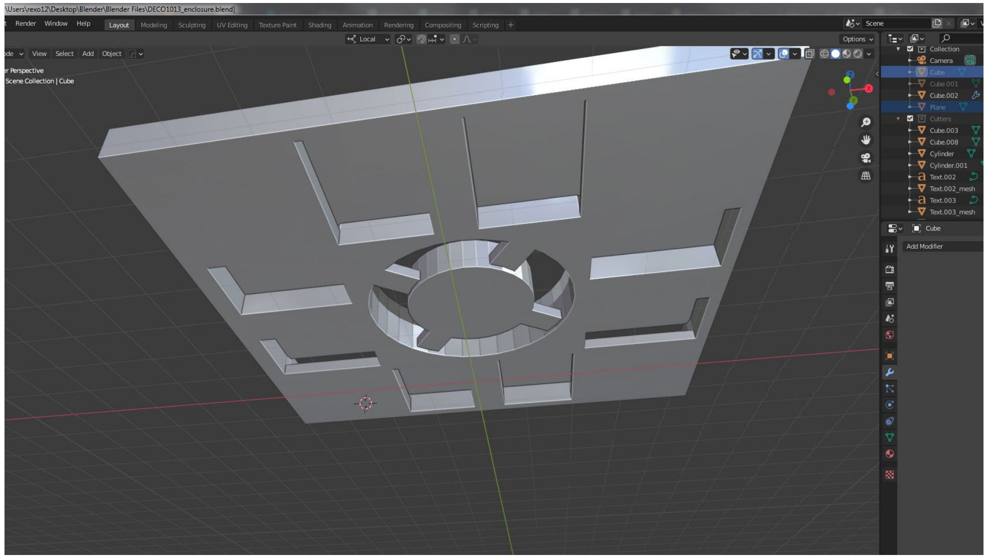
The Neopixel working as intended

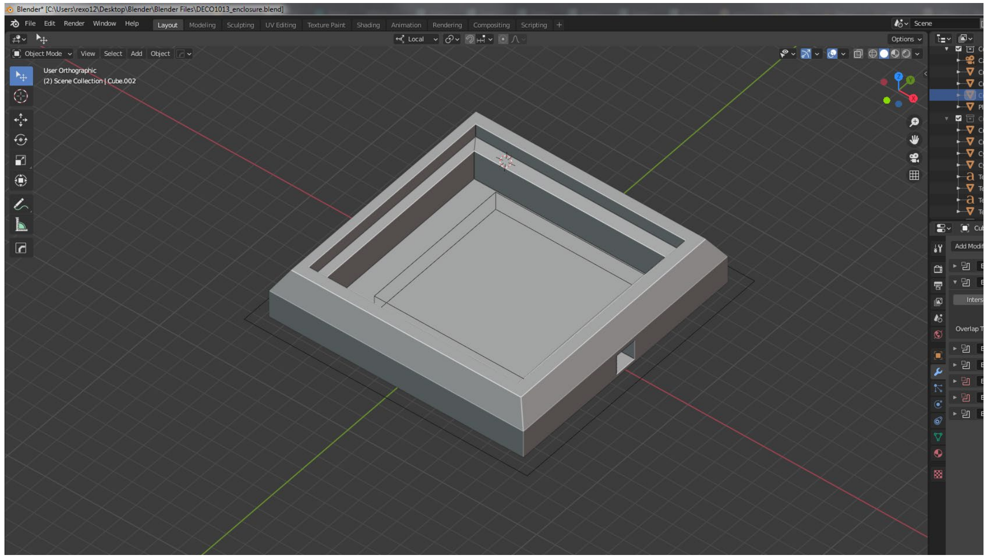
Fixing this involved patching the micro:bit micropython runtime source code to support RGBW pixels and bring the library in-line with the full WS2812/B specification. This patch added an extra channel argument to the NeoPixel constructor that allowed programmers to specify how many channels their neopixels used, which was passed to the assembly function that writes data to the WS2812 chip. The ARM x32 Procedure Call Standard reserves registers 0-3 as scratch/argument registers, and so register 3, unused by the assembly code, was used to pass this channel argument in. This new value is used to dynamically calculate how many bytes need to be written before the function returns, thus accommodating for both RGB and RGBW neopixels. The patch can be found here.

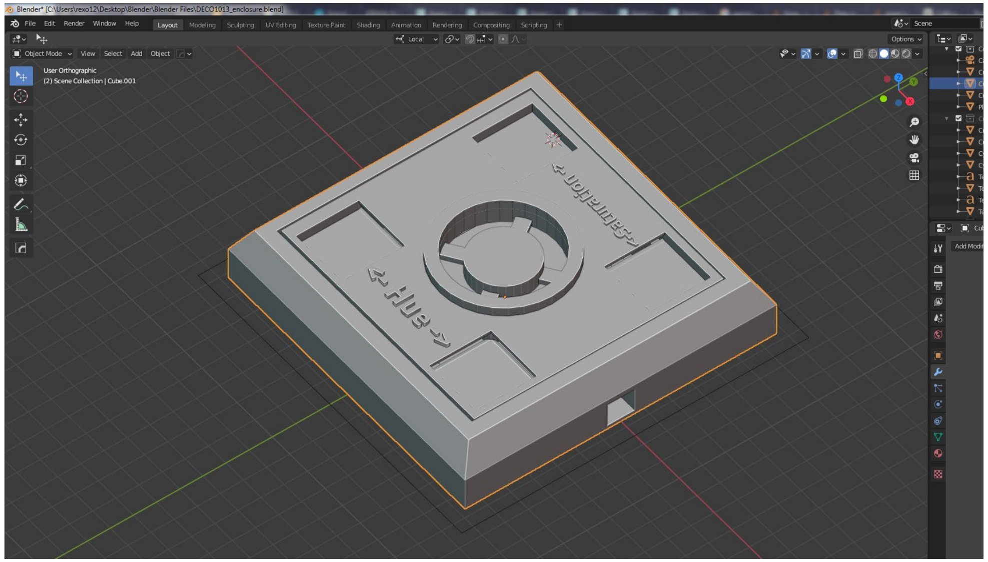
At this stage it was also decided to do away with the microphone component of the design, as it was found in testing that it obfuscated the functionality and interactivity of the design far too much, showing oscillations and seemingly random responses to user input, which made it too frustrating for users to build an understanding of what the design was doing.

Housing Design

Housing was designed in Blender and 3D printed. The body makes up 2 printed components - A base shell to house the micro:bit, shield and other components, and a faceplate where input/output is situated. The central ring houses the neopixel ring, while the 4 inset pads house the touch sensors. Lettering on the faceplate tells the user which sensors control what, with the hue and saturation pairs placed opposite each other to communicate their grouping - e.g. The left hue control shifts the hue ‘left’, or towards zero, and the right to 360. In line with the interaction model, these values loopback once they hit either extreme, so which direction one chooses is largely irrelevant, contributing to user ease-of-use.

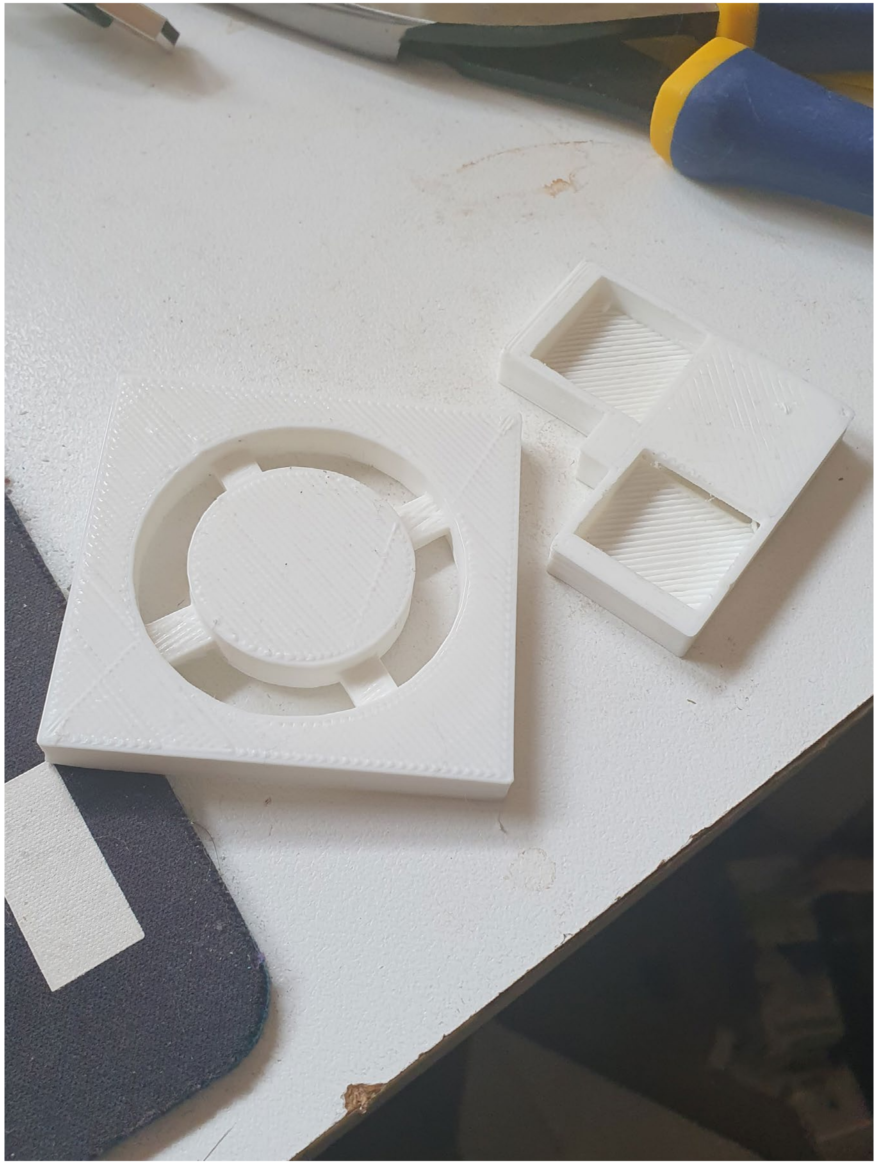






Housing Tests

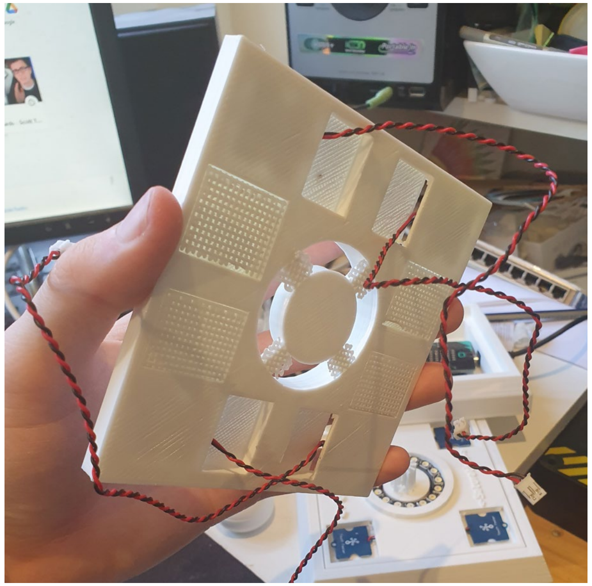
Small test pieces were printed to check fits for different parts of the housing. Pictured here are the supports for the neopixel ring (left) and inserts for the touch pads (right). The pads are inserted through the slits from underneath and settled in the inset.

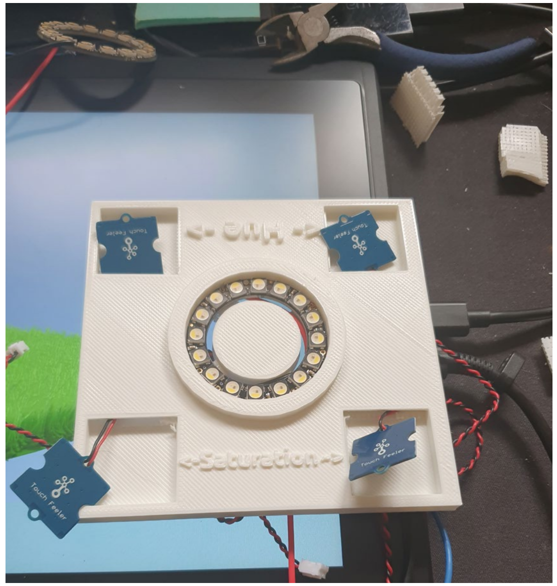


Final Fabrication

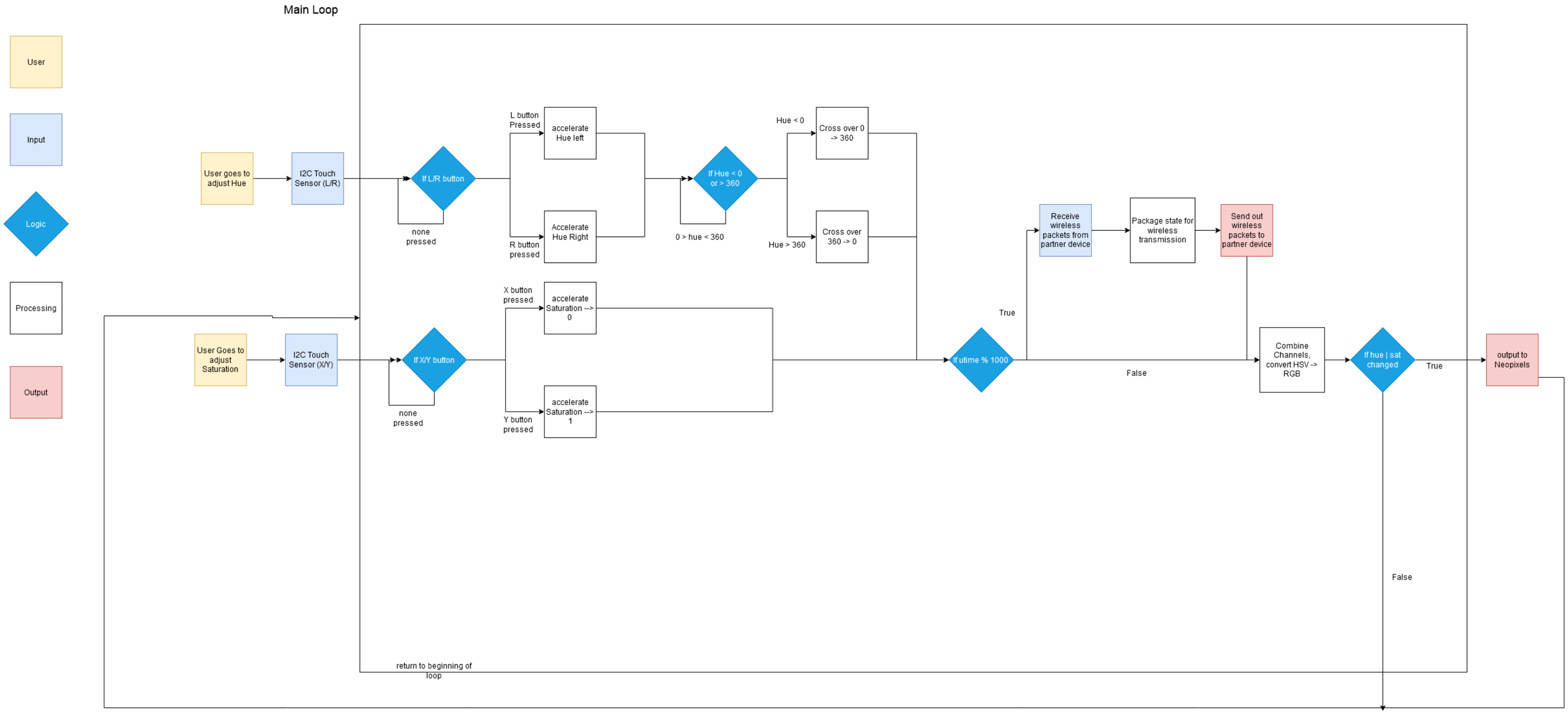
Finished Fabricated Modules

Because there were two housings to be fabricated, this stage doubled as a final, minor prototyping stage as well. As such, there are some slight variations between the individual housings, such as a taller power cable port and smaller touch pad insets, based on findings and optimisations from the initial print.



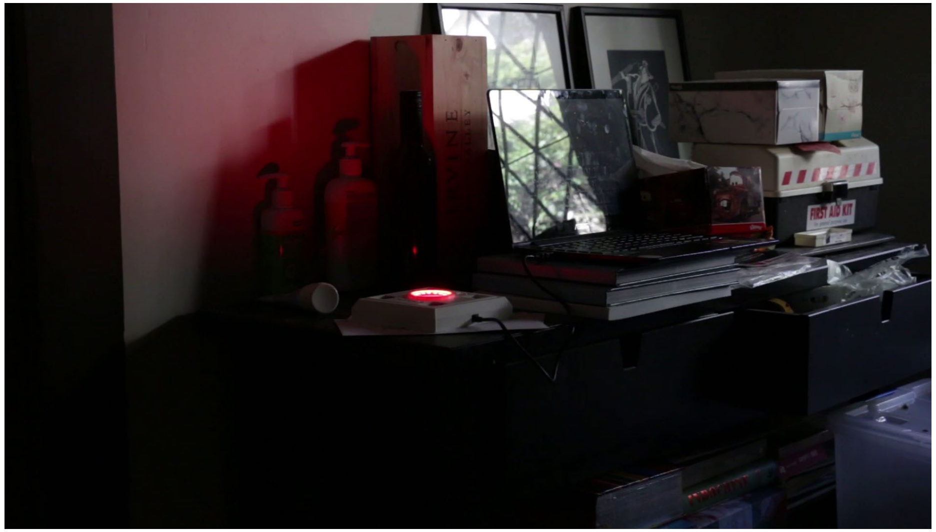


Cabling for the touch sensors threaded through the fabricated faceplate. While 2 entrance slits were designed per inset, I only removed the supports for one. These secondary slits were only designed to allow flexibility in case tolerances did not allow the primary to be used for whatever reason. The neopixel ring fit snugly in the face-plate. Leads were soldered to the PCB connectors on the ring and spliced onto one of the grove 4pin connectors.

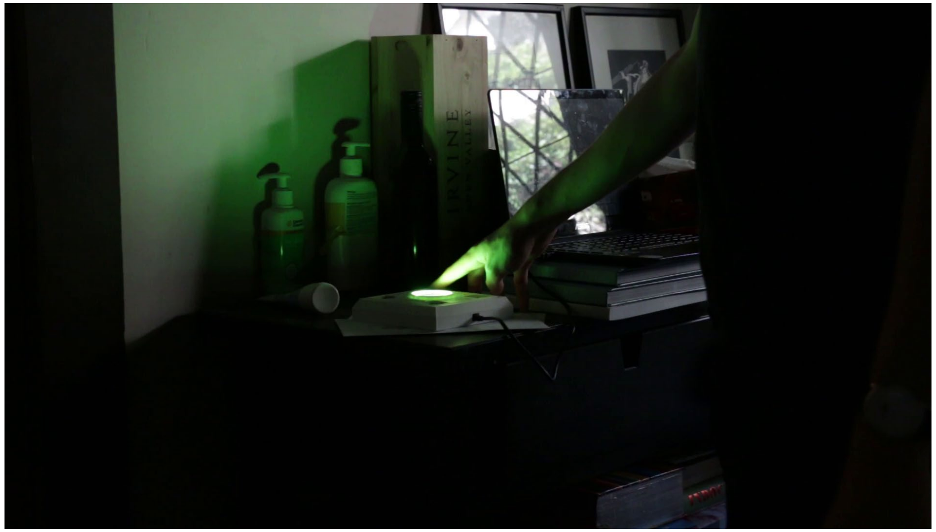


Revised final interaction model.

Storyboard Use



Lamp in home thoroughfaire



Users shifts hue on way past

Lamp in private space.

Lamp receives message.

Appendix

1. Design Proposal Code and source files repository: https://github.sydney.edu.au/ssav7912/deco1013-microbit

2. Patched RGBW neopixel micropython runtime repository: https://github.com/ssav7912/micropython/tree/RGBW-Support

References

Gaver, W., Bowers, J., Boucher, A., Gellersen, H., Pennington, S., & Schmidt, A. et al. (2004). The drift table: designing for ludic engagement. In 2004 Conference on Human Factors in Computing Systems. Vienna, Austria. Retrieved 13 September 2020, from https://www.researchgate.net/publication/221516001_The_drift_table_designing_for_ludic_engagement.

Crabtree, A., Rodden, T., Hemmings, T., & Benford, S. (2003). Finding a Place for UbiComp in the Home. Retrieved 17 September 2020, from http://www.cs.nott.ac.uk/~pszaxc/work/UC-1-03.pdf

Hutchinson, H., Bederson, B., Druin, A., Plaisant, C., Mackay, W., & Evans, H. et al. (2003). Technology Probes: Inspiring Design for and with Families. Retrieved 15 September 2020, from https://www.researchgate.net/publication/228438101_Technology_Probes_Inspiring_Design_for_and_with_Families.

Wikipedia. (2020). HSV and HSL. Retrieved 8th, October 2020, from https://en.wikipedia.org/wiki/HSL_and_HSV.

Vagrearg. (2019). Fast HSV to RGB Conversion. Retrieved 8th, October 2020, from https://www.vagrearg.org/content/hsvrgb.

Smith, A (1978, August, 12-19). *Color Gamut Transform Pairs* [Paper Presentation]. SIGGRAPH '78: Proceedings of the 5th annual conference on Computer Graphics and interactive techniques, Atlanta, GA. Retrieved 23rd of November 2020, from: https://www.researchgate.net/publication/233784968_Color_Gamut_Transform_Pairs

Internet Engineering Task Force. (1981). *Transmission Control Protocol - DARPA Internet Program - Protocol Specification* (RFC: 793). Retrieved 23rd of November 2020, from: https://tools.ietf.org/html/rfc793#section-2.7

Seeed Studio (2015) *Grove I2C Touch Sensor Library* (e958653) [Library Source Code] https://github.com/Seeed-Studio/Grove_I2C_Touch_Sensor.

micropython-on-micro:bit (2018) *Micropython for the BBC micro:bit* (v1.0.1) [Language Runtime Source Code] https://github.com/bbcmicrobit/micropython

ARM. (2012). *Procedure Call Standard for the ARM® Architecture* (IHI 0042E). Retrieved 4th November 2020 from: https://static.docs.arm.com/ihi0042/e/IHI0042E_aapcs.pdf