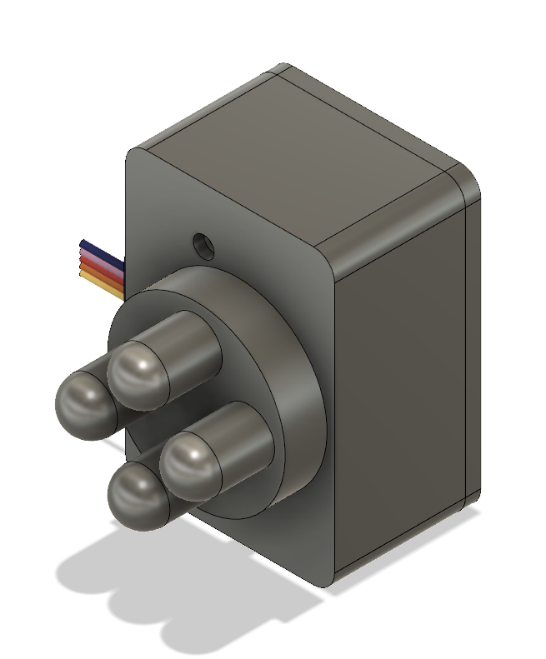
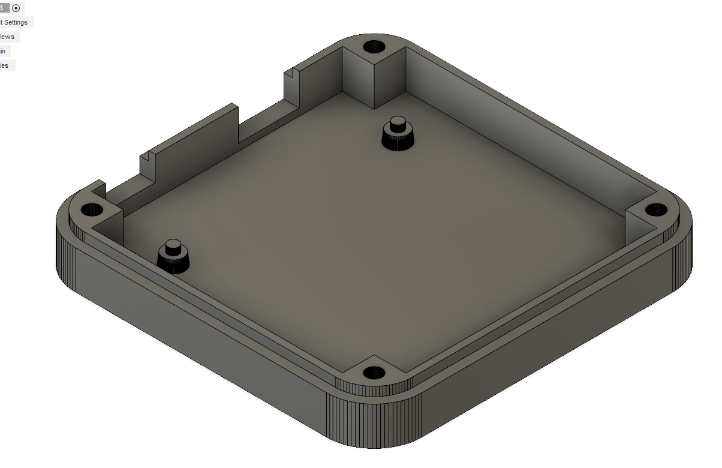
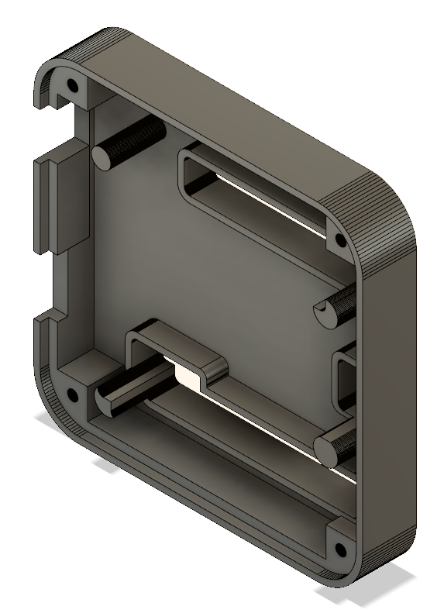
**Sprint 2 (No documentation for sprint 1)**

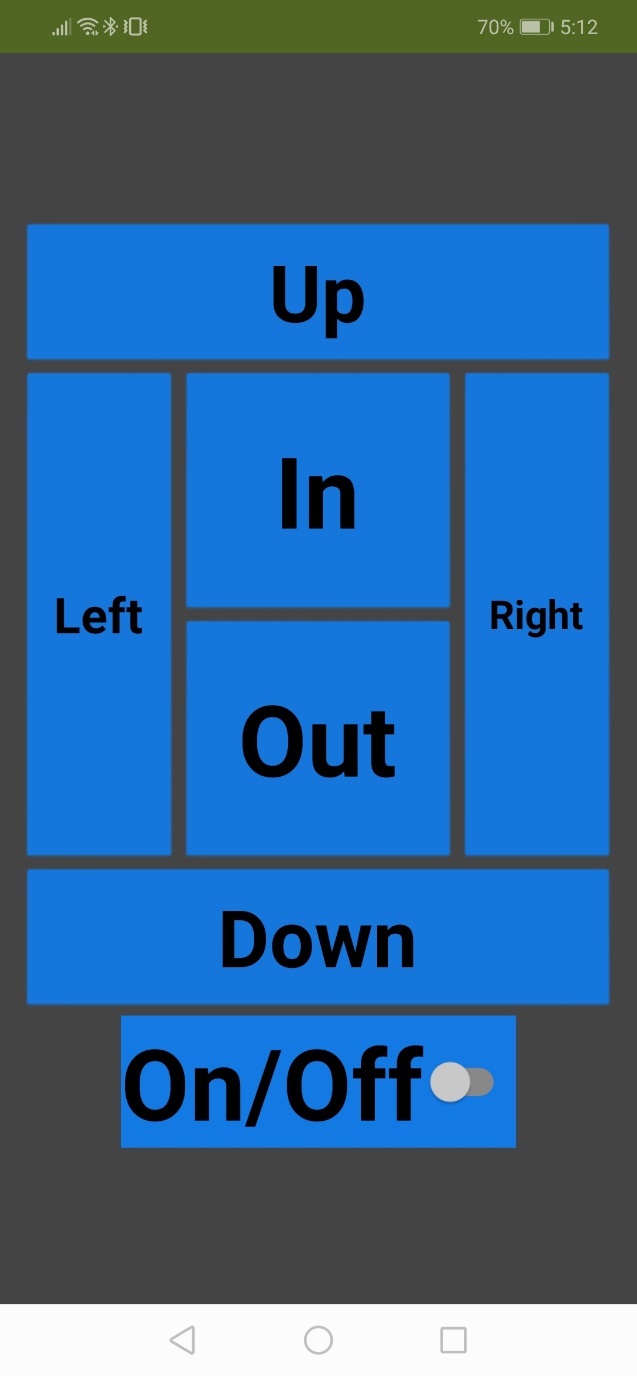
Body Mount (Modified Backpack), will be further modified later in development to allow access to have an open back region

Constructed Housing

3D Modelled Designs of Housing

3D Modelled Design of Head

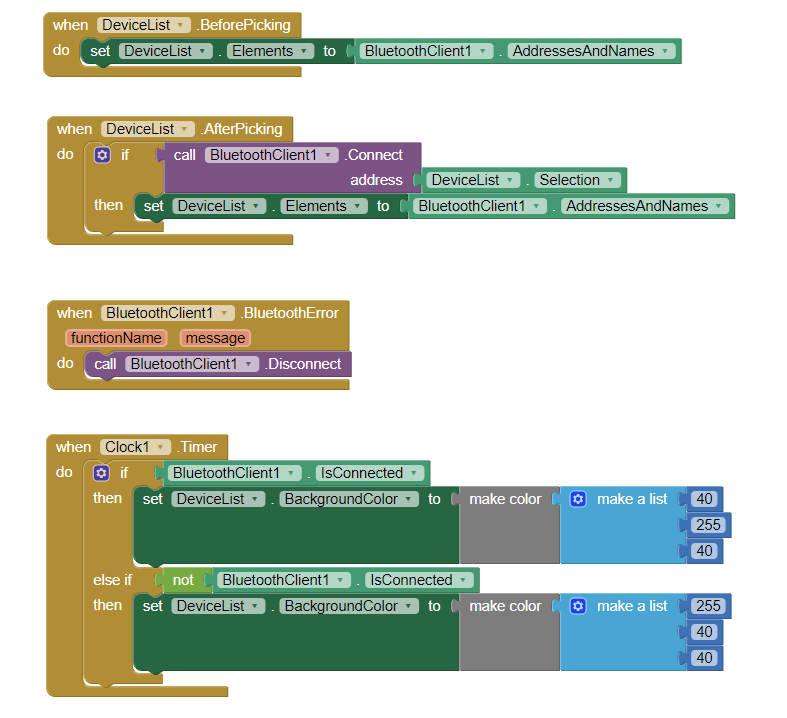
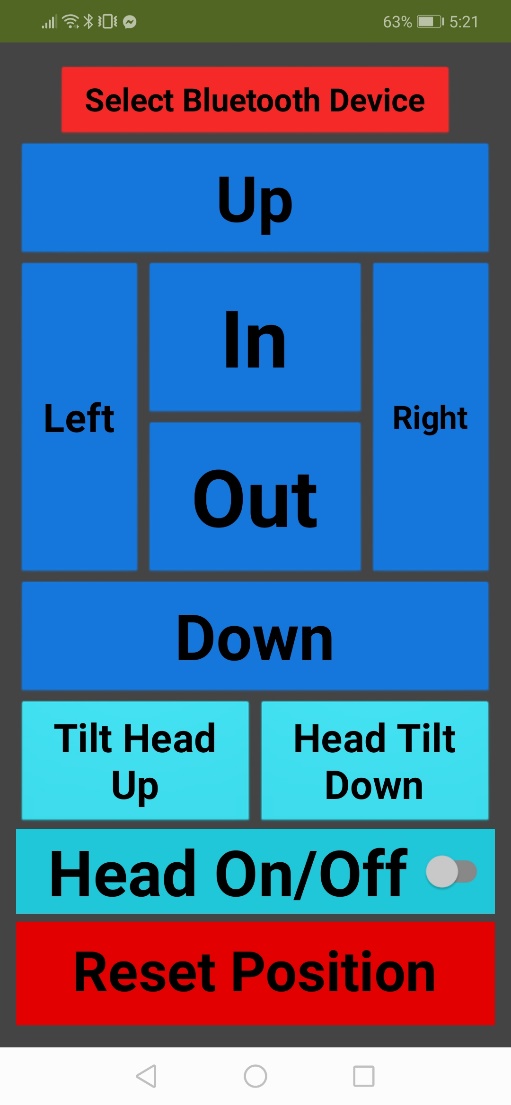
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type of test | What is being tested? | Expected result | Actual result | Action taken (if any) | Evidence | Notes |
| Expected | Under control of a test code, the head’s gyration should be controlled | Head gyration can be turned on and off on command |  |  |  | Postponed for next sprint as the related task was substituted out |
| Expected | The access of the Arduino board’s pins through the housing | Pins should be reachable by jumper cables through the holes in the housing | All pins I could foresee using are accessible |  |  |  |
| Expected | The micro USB port of the Arduino board’s accessibility through the housing | The housing should not obstruct the cable from reaching the port | No obstruction |  |  |  |

**Sprint 3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type of test | What is being tested? | Expected result | Actual result | Action taken (if any) | Evidence (if applicable) | Notes |
| Expected | Head’s ability to spin, using the provided controller and a temporary circuit and Arduino script | Head spins at a rate specified by the script | As expected |  |  |  |
| Expected | The connection between the backpack and the base of the arm | The base has minimal tilting out from the backpack | The base tilts out significantly but I think it’s reduced as much as possible. The connection (using zip ties) seems like it will degrade the bag however | From “consumer” suggestion (my dad) I printed a plate to sit on the inside of the backpack to reduce the zipties’ wear onto the bag |  |  |
| Expected | The physical connection between the base and its cap | The connection is strong enough to hold itself when the apparatus is its average position | As expected |  |  | The connection is not as tight as it could be, to reduce friction between the spinning cap and the base |
| Expected | The physical connection between the base cap and the first arm-stalk | The connection is strong enough to move successfully when guided and hold itself up well | As expected |  |  | As previously designed, there are rubber bands in place to assist the motor in holding the weight of the arm |
| Expected | The physical connection between the first and second arm-stalks | The connection is strong enough for the second arm-stalk to pivot in a study manner and hold itself well | The connection is sturdy, but the first stalk’s servo motor can’t freely the weight of the second stalk | A new set of stalks was printed with hooks to place rubber bands on, helping hold the stalks together |  |  |
| Expected | The physical connection between the second arm stalk and the head | The connection is strong enough for the head to pivot in a sturdy manner | As expected |  |  |  |
| Expected | The rotating base’s ability to rotate | Using a scrap Arduino script, the base can rotate to points within its range | As expected |  |  |  |
| Boundary | The rotating base’s ability to rotate to the ends of its range | Using a scrap Arduino script, the base can rotate to its max/min angles | Rotates successfully, though the base cap seems to tilt out at these extreme values | Base cap tightened |  |  |
| Invalid | The rotating base’s input | Using a scrap Arduino script, when given an input outside the motor’s range, the base either doesn’t move or moves as close as it can | Rotates to the closes point it can (maximum or minimum) |  | As above |  |
| Expected | The shoulder joint’s ability to rotate | Using a scrap Arduino script, the shoulder joint can rotate to angles within its range | As expected |  |  |  |
| Boundary | The shoulder joint’s ability to rotate to the ends of its range | Using a scrap Arduino script, the shoulder joint can rotate to its max/min angles | Shoulder rotates fine to its minimum angle but not to its maximum angle of 180 degrees(too much weight) however in practice no angle over 90 degrees will be necessary so this is acceptable |  |  |  |
| Invalid | The shoulder joint’s input | Using a scrap Arduino script, when given an input outside the motor’s range, the shoulder either doesn’t move or moves as close as it can | Joint moves as close as it can to the invalid angle |  | As above |  |
| Expected | The elbow joint’s ability to rotate | Using a scrap Arduino script, the elbow joint can rotate to angles within its range | As expected |  |  |  |
| Boundary | The elbow joint’s ability to rotate to the ends of its range | Using a scrap Arduino script, the elbow joint can rotate to its max/min angles | As expected, although the minimum angle was not tested as the arm is physically obstructed to this position (closest possible angle tested instead which encompasses all necessary angles) |  |  |  |
| Invalid | The elbow joint’s input | Using a scrap Arduino script, when given an input outside the motor’s range, the elbow either doesn’t move or moves as close as it can | Joint moves as close as possible (not tested on minimum angle for reasons above |  | As above |  |
| Expected | The wrist joint’s ability to rotate | Using a scrap Arduino script, the wrist joint can rotate to angles within its range | As expected |  |  |  |
| Boundary | The wrist joint’s ability to rotate to the ends of its range | Using a scrap Arduino script, the wrist joint can rotate to its max/min angles | As expected |  |  |  |
| Invalid | The wrist joint’s input | Using a scrap Arduino script, when given an input outside the motor’s range, the wrist either doesn’t move or moves as close as it can | Joint moves as close as possible |  | As above |  |
| Expected | The head motor can still operate, when connected to the other parts | Head spins at a rate specified by a scrap script | As expected |  |  |  |

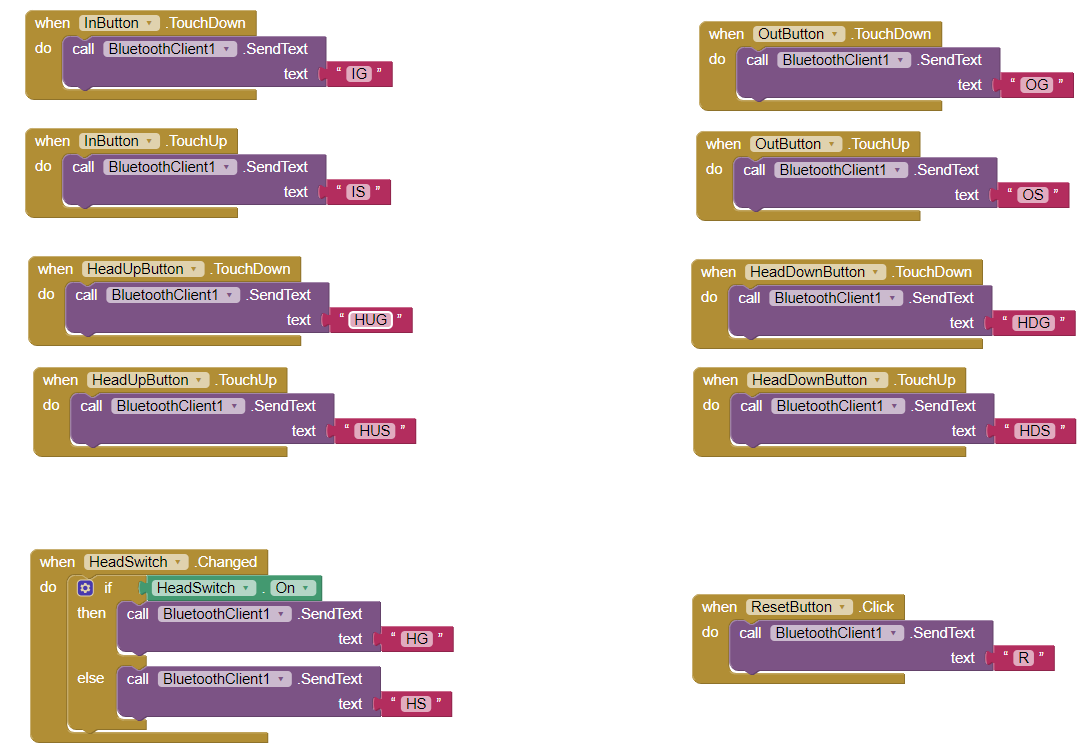
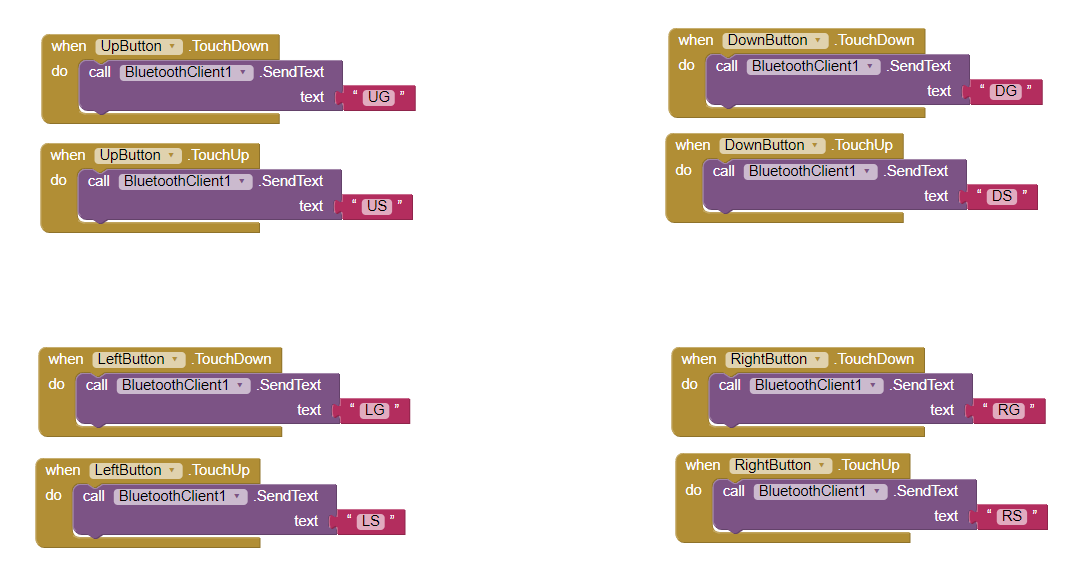
Discoveries too general to be applied to any one test:

* Upon the initial construction, motors were attached at random angles. It was required to synch the programmable range of the servos to the angles at which the connection components were mounted, and so a set of ranges was determined for each joint, to best encompass the possible required positions.
* The intended power supply, a double USB port power bank, stops outputting power after roughly 15 seconds of powering either the Arduino or the motors of the apparatus. It would seem the power bank has built-in current requirements so that circuits drawing low amounts of current are terminated, as my other lower quality power bank seems to work, however this only has one USB port so cannot power the apparatus in a fully portable manner. I will continue testing power sources in the next sprint but will most likely use two low quality ones without this current requirement.

******Sprint 4**

App Coding Blocks (Bluetooth connection)

Final App Design

****

App Coding Blocks (Output data)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type of test | What is being tested? | Expected result | Actual result | Action taken (if any) | Evidence | Notes |
| Expected | App commands reaching the Bluetooth module and subsequently the Arduino | Using a serial print function, the buttons pressed correspond to the printing of their text output | A combination of “x” and “?” characters are received instead, e.g. “xxxx??” or “xxx???xx?”, while these are different for each button they are clearly not the intended result | Research done on connecting Bluetooth modules, the module I am using requires a baud rate of 9600, so I used this instead of the previous rate of 38400, this gave the expected result |  | Abbreviated outputs were used as opposed to full words e.g. “UG” instead of “UpGo” because the processing of longer outputs could sometimes not be completed before the next loop starts (only at short loop periods like 10 milliseconds) |
| Expected | App commands being received at appropriate times | Using the same serial print method as above, each command is only received once per trigger and at the appropriate time | As expected, nothing is received when no button is pressed and the appropriate commands are received for the appropriate actions (typically pushing and releasing buttons) | None | Practically as above |  |
| Boundary | Range of button tap recognition | When tapping just outside the buttons on the app, no commands are received by the arduino | As expected |  |  |  |
| Invalid | Tapping outside of any buttons | When a non-button region is tapped, no commands are received | As expected |  |  |  |
| Expected | Bluetooth control of the base joint | Linking the motor to its corresponding buttons, does the motor respond and do so in the correct direction to the holding of the buttons | As expected | None | Not applicable in a word document (cannot be proved through pictures) | This is done in small increments as it equates to a large displacement |
| Expected | Bluetooth control of the shoulder joint | As above | Actions are triggered, but directions are backwards | Signs of the increment values (+1/-1) | Not applicable by image | As noted in sprint 3, the joint cannot rise higher than practical amounts, so manual limits will be put on its rotation in later coding |
| Expected | Bluetooth control of the elbow joint | As above | As expected | None | Not applicable by image | Manual limits will be put on this to prevent it from stretching itself out too far (this tangles the rubber bands in the cables) or folding in too far (this makes the attached stepper controller collide with the first stalk |
| Expected | Bluetooth control of the wrist joint | As above | The motor seems to be receiving the right commands, however they are executed in a jittery manner and sometimes not at all | After testing various sections of the wiring leading to the servo, I found that the problem was some wires which I folded onto themselves to save room, they were replaced and the result is now as expected | Not applicable by image |  |
| Expected | The synchronised movements of the up command | When the up button is held, the elbow shoulder and wrist move together in such a way that the head contracts directly towards the base, while remaining the same orientation in space | As expected |  | Not applicable by images | Initially my plan for this movement was to construct a mathematical relationship to have the head move the same distance up regardless of how extended the arm already is, however the current method (changing the angles regardless of arm extension) gives a reasonably constant speed, so that is not necessary |
| Expected | The synchronised movements of the down command | When the down button is held, the elbow shoulder and wrist move together in such a way that the head contracts directly out from the base, while remaining the same orientation in space | As expected |  | Not applicable by image |  |
| Expected | The toggling of the head’s scratching action | When the Head On/Off switch is pressed on the app, the scratching motion toggles its state, tested for both on to off and off to on | The scratching is triggered, but is slower than the max speed of the motor | The additional delay of the loop was causing periodic pauses in the scratching motion, the code has been modified to not have a delay when scratching is active. | Not applicable by image | This adjustment also caused scratching to be paused when the servos are moving, which is fine |
| Expected | The reset button and the subsequent movement of the servos | When the reset button is pressed, all joint revert to their original position | As expected, but when moving the servos after this they jolt back to near the previous positions | There was a mistake in the code, it was adjusted so that the stored angle variables were reset to the original angles, not just the servos | Not applicable by image |  |
| Boundary | The manually set lower limit of the elbow joint and therefore the up command | When attempting to move the head up when the elbow joint is already at 26 degrees (the lower limit) no movement is made and the elbowAngle variable is unchanged | As expected |  | Printed angle values over time: | The joints are ever so slightly unstable here due to servo strain however in practice there will be the user’s back to lean on, so this shouldn’t be a problem |
| Boundary | The manually set upper limit of the elbow joint and therefore the down command | When attempting to move the head down when the elbow joint is already at 140 degrees (the upper limit) no movement is made and the elbowAngle variable is unchanged | As expected |  | Printed angle values over time: | There is no invalid case for this test and the one above as the angle can never reach anything past boundary |
| Boundary | The manually set upper limit of the shoulder joint in the up command | When attempting to move the head up where the elbow joint is well within its range, but the shoulder joint is at its upper limit of 100 degrees due to overuse of the out command, no movements are made and the variables are unchanged | As expected |  | Printed angle values over time: |  |
| Boundary | The manually set upper limit of the shoulder joint in the out command | When attempting to move the head out when the shoulder joint is already at 100 degrees, no movements are made and the shoulderAngle variable is unchanged | As expected |  | As above |  |