Develop lift interfaces

# Brief

You are to design and build a lift control system for your client who owns a 3-level house on the Port Hills. Her elderly mother will be moving in with her and she can’t navigate the stairs so need an easy way to get to the different levels of the house. For the purpose of proving the concept you need to be able to move the lift cart from the living area(which is the same level as the garage) to her room (highest level) and back. Eventually she would like to go to the spa level as well which is the lowest level in the house.

The lift should have a control system that interacts with sensors for example the call buttons and position sensors to manage the movement.

There must be additional features such as displays to show the lift position, sound at the end of an operation, operator override controls, smoke detection lock-out, automatic maintenance alerts, swipe card access, pin access, overweight protection. This is especially important for the Level 3 standards. Combining this with a network that sends maintenance alerts and other information to a central server, will enable you to achieve the network standard as well (Due date for this will be **8 October 2021**).

You are allowed to work in groups as part of developing the mechanical aspects of your lift system. Although aspects of the electronics interface may be similar due to the same mechanical design, each student creates and completes their own lift control system. Each student must have defined roles in this project. *Theoretically, depending on the complexity of the system, you could build one system but this needs to be negotiated with Mrs Dunn.*

# Completing the Task

As you perform the task, make notes and gather evidence for inclusion into your electronics portfolio.

## Project Brief

Start by clarifying what you are going to do by assessing the **client's needs** :

1. What is the **purpose** of the outcome? Why are you doing it?
2. Who are you making it for?
3. Explain all the **relevant implications** that needs to be considered for your lift and how these will be addressed, such as:
   1. Why software code needs to meet codes of practice
   2. Why the system needs to meet end-user specifications
   3. Why the system needs to comply with all relevant intellectual property.
   4. Why the system has to comply to health and safety requirements
4. List all the **specifications** for the outcome. *Remember specifications are measurable and it is what the client wants. Other things may look like specifications but are most likely part of your design phase.*

## Design

*Some of these areas will be designed in more detail within each key stage but you use the same ideas.*

1. Identify the inputs and outputs used in the tasks
2. Investigate various **conventions** and choose at least least three you will be using in your design for example: schematics, system block diagrams, naming conventions.
3. Generate and model a **range of design ideas** for each of the following:
   1. Sketch a circuit for your proposed electronic system, include correct symbols and labels for components you select. Use Yenka and/or Tinkercad Circuits for modelling your circuit. *At this stage it could just be pseudo code or a description of various functions et*c
   2. Sketch the 3D model of the lift. Decide on various options for motor and interfaces positions. A Fusion 360 model will be provided that you can modify.
4. **Select a design** from the ones above and explain the appropriateness of the design. Why did you choose it? Explain how design conventions were used in the ones you chose. *Link any choices to the digital part of your design.*
5. Describe/Explain various components and systems you will need
6. Break your design up in three functional areas using block diagrams
7. one to manage the motors
8. one to manage the sensors that detect the position of the lift
9. one to manage passenger requests and information.

**Plan the lift outcome** beforehand using block diagrams, flow charts and natural language. Use simulation programs (Yenka, Tinkercad Circuits etc) and calculations to ensure you meet the requirements.

## Processes

Use appropriate **project management tools** and techniques to plan the development of your lift. Use key stages, task lists, milestones, file naming and version control systems and collaboration tools. Regularly manage this.

When you break up your key stages think about what you will demonstrate to your client so they can see your progress. Key stage 1 may be motor movement and demonstrate how the lift will move up and down, key stage 2 may be adding the switch to control the lift movement etc.

As you develop, construct and interact the subsystems, record the work you are doing using photos and notes. Use these records to write a report, which you will submit as evidence for the assessment of this standard.

Provide a thorough evaluation at the end of each key stage of the results and current progress and use this as input to the next stage. You only need to list your tasks at the start of each key stage. Are there any changes you need to make?

There will be **formative assessment at each key milestone**. It is your responsibility to ensure you are ready to submit your work at these milestones.

| **Keystage** | **Due Date** |
| --- | --- |
| 1 Lift Movement (ex) | 4 June 2021 |
| 2 Call Buttons (ex) | 2 July 2021 |
| 3 Display (ex) | 11 August |
| Final Submission | 20 August 2021 |

## Construct your Lift System

As you work through your key stages, construct functional models for the lift subsystems (see Appendix A for examples). Write, test, and debug software (See Appendix B for examples) to control the subsystems and their interaction. Test all input and output interfaces separately. This include:

* Test all input interfaces on a range of operating conditions.
* Test all output interfaces on a range of input conditions.
* Evidence of modifying and debugging software code
* Show how you addressed the relevant implications within development

Make sure you use well-structured code, the outcome function as intended and is reliable.

Make sure that you:

* select the best type and value of component
* select the best arrangement of components
* modify hardware input and/or output parameters
* modify software parameters
* account for electronic noise and its filtering
* List all the tests you performed and any modification to hardware and software

## Trialling and Iterative Improvement of the Lift System

Iterative improvement in your system will show evidence of using documented cycles to refine functionality, reliability and fitness for purpose. Evidence needs to show the use of the design, build, test and evaluate process in various areas, such as:

* Software standards and maintainability
  + Choosing appropriate functions
  + Readable and understandable code
  + External software libraries
  + Appropriate docstring and comments
* Understandable user interface
* Hardware reliability - *Analyse and modify your initial interfaces until they are substantially improved and you are satisfied that they perform the functions required and meet specifications (controlling the motors, detecting the position of the lift, and responding to passenger requests and information). This may include switch debounce, improved regulation/stabilization of power supply.*
* 3D Modeling - How you have improved the lift design.

## Documentation

**Document** the process and techniques used to set up and test the specified lift system in the form of annotated photographs, diagrams, short video clips, or code snippets that demonstrate understanding, skills and knowledge and explain decisions made by you. The process you used to work through the project must be clear. Make sure you have the following evidence to demonstrate how you:

* Used the electronic components provided to produce a sensor(s) that can interact with the environment
* Used the electric motors provided so that the software on the lift can reliably vary the speed and position of the lift car
* Wrote, tested, and debugged well-structured, clearly annotated, readily understandable software to effectively manage the interface between the microcontroller and the sensors and actuators it controls
* Analysed, tested, debugged and modified the sensor subsystems and actuator subsystems. This is to substantially improve the way the lift subsystems work/interact and the quality of the data delivered by the interface.
* Evaluated and justified the choice of components and systems used
  + A photograph of your electronics system. Annotate your photographs with descriptions of each component and interface (what it does, how it functions),
  + Explain the behaviour and function of the electronics outcome
    - C++ Functions
    - Switches (Voltage Divider) and PWM
    - I2C communications and LCD interface
    - Serial communication between Arduino application and hardware. Explain use of serial monitor for debugging.
    - Switch debounce and software or hardware solutions to solve debouncing
    - Use of AND/OR logical functions to read multiple switches
    - Influence of various switches settings on motor control
  + Justify your choice of the components and systems used
  + Justify user Interface features used (controls used lift positions
  + and indicators)
  + Justify methods used to resolve switch debounce

## Presentation

**Present your work** to the client (teacher and others identified) at the key milestones. *Selling yourself and your ideas is an important part of Electronics and Computer Engineering. You must produce a professional and polished presentation, designed to ensure your client is confident that you can do it and make the effort you've put into developing a top-notch product.*

## Final submission

**Due 20 August (Except Networking)**

All relevant documentation, your lift and the presentations count towards your final grade. Presentations will be held throughout the assessment period in defined times. You should have at least three key milestones.

Complete the assignment checklists to ensure you have covered all the criteria in the assessments.

Hand in your lift control subsystems, your report and the assignment checklists.

# Appendix A

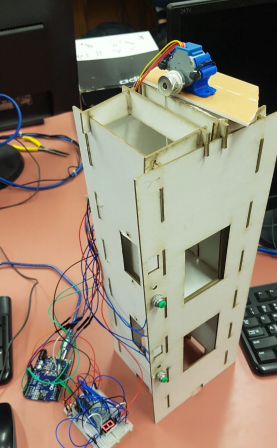
## Human interface

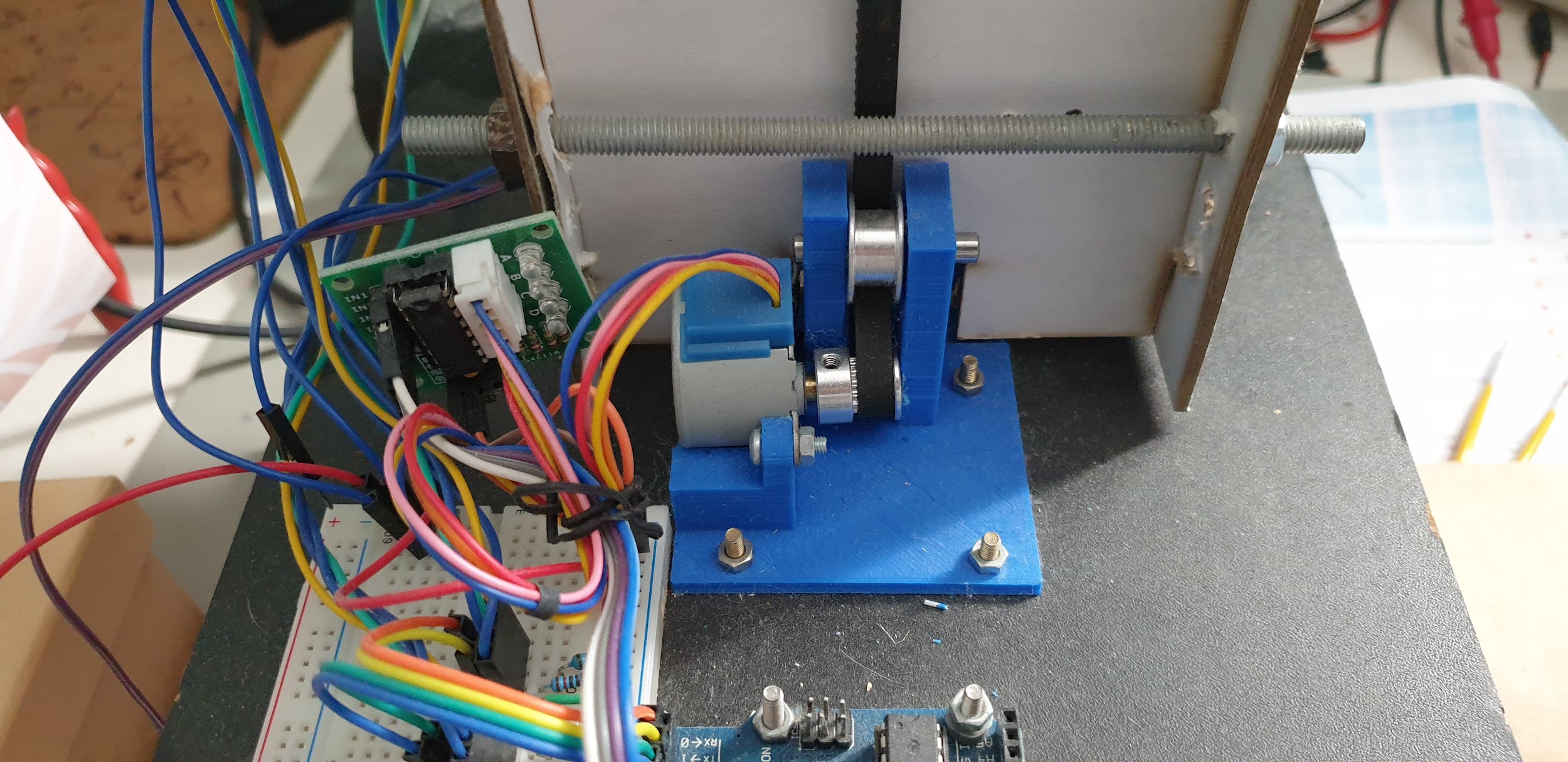
This photo shows the switch that the user can press to request a lift to go to the floor they are on. It also shows the LED that indicates that the request has been actioned. The seven segment LED will indicate the current floor the lift is on.

Below is the PCB used with a socket for a 4511 BCD to 7-segment latch decoder IC to drive the 7-segment displays.

## Motor Control Interfaces

Below are two examples of moving the lift up and down. One from the top using gravity. The other is using a pulley with a timing belt.





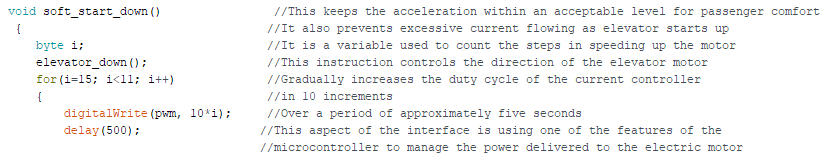
# 

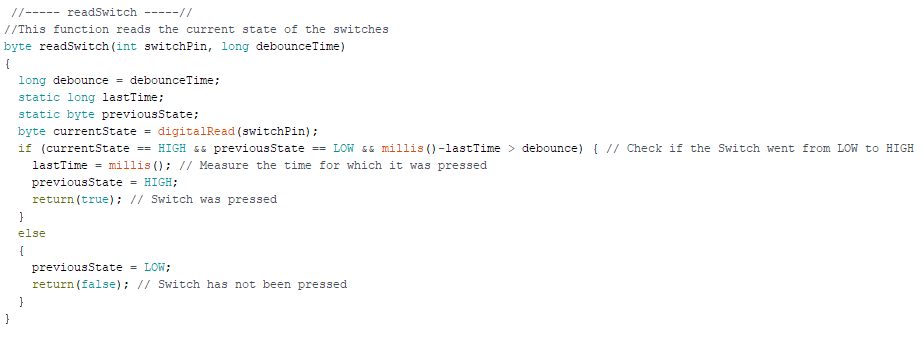
# Appendix B: Code snippets

The code shown in these snippets are examples of the way software is used to manage the interface hardware.

In this case, the hardware side of the interface is an Arduino that is capable of controlling the direction and power delivered to a motor. The software side of the interface is the code that is used to manage the hardware.

The snippet below is an example of this code:





# 