Hydroelectric Generator for High-Rise Buildings

EPQ artefact report

Orlando Alexander
Berkhamsted School

Table of Contents

Acknowledgements

Matthew Knight (Akro Valves) is an ambassador for STEM in schools and very generously supported my project by donating an array of components for the turbine system.

Dr. Sam Williamson (Bristol University) is an expert in micro hydropower systems and generously gave up almost an hour of his time to discuss design considerations in hydroelectric systems.

Abstract

This report will detail the design process behind the development of my EPQ artefact with the project aim: 'To design and create a hydroelectric generator for apartment buildings, incorporating a microcontroller to enable automation and feedback'. Initially, I will provide a high-level summary of my design and why I chose it. Then, I will present a mathematical analysis of the viability of my system versus other proposed techniques for harnessing energy from the grey wastewater of high-rise buildings. I will then walk-through the technical details of each aspect of my design and explain the development process behind my design choices, citing both secondary and primary research to support my decisions.

Introduction

With climate change and diminishing resources making continued reliance on the existing, non-renewable energy sources, such as coal and oil, impossible (MacKay, 2013), we must increase the range of renewable energy sources we have available. While MacKay concluded in his book *Sustainable Energy – Without the Hot Air* that the UK has the theoretical capacity to produce up to 177.5kWh per day per person (kWh/d/p)[1], much of these energy solutions would be highly obtrusive and would tarnish Britain's countryside. Thus, considering the very probable widespread objection to these widespread renewable power plants, MacKay concluded that the UK would only manage to produce around 18kWh/d/p.

My hydroelectric generator for high-rise buildings provides a new, compact energy solution which uses existing infrastructure to generate electricity at a local level, avoiding the opposition towards most renewable energy technologies. The generator system is contained within a single wastewater pipe running down the side of a high-rise building and is powered by the kinetic energy of the grey water[2] as it passes across a Pelton water turbine located on the ground floor of the building. The system maximizes the energy harnessed from the wastewater by using several microcontrollers, electromagnet-operated barriers and integrated sensors which work together to redistribute the grey wastewater throughout the pipe before it is released across the turbine.

There are several existing proposals for suburban hydroelectric power systems. One such example was proposed in the International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET) by Ananthapadmanabhan G S and Dr. Shouri P V, a student and associate professor respectively at the Model Engineering College, Kochi, Kerala, India (Ananthapadmanabhan and Shouri, 2017). Their solution used a collection tank situated half-way up the building, which collected the grey wastewater from the floors above it and, when it became full, released this water across the turbine situated on the ground floor. However, I found two key issues with this design which limited its potential electricity generation: firstly, only half of the grey water from the building can be used for power generation, as wastewater released from floors below the tank does not pass across the turbine. Secondly, all the wastewater is stored at a constant head, equal to the height difference

[1] The average energy consumption per person per day in the UK is 125kWh, excluding the energy embedded in imported goods (MacKay, 2013).
[2] Grey water includes waste bath, shower, sink and washing water (The Green Age, 2012)

between the tank and the turbine, which 'wastes' the innate head of the water released from floors above the tank.

In my design, the use of dynamic barriers within the pipe serve to divide the grey water pipe into sections, with each section corresponding to the piping for one floor in the high-rise building. Doing so means that the entirety of the wastewater from the building passes over the turbine[3]. Moreover, the wastewater from each floor can be stored at or close to the height at which it is released from the building. This greatly increases the power potential from the wastewater, as a key component of hydropower is the water head[4].

[3] Note that the wastewater from the ground floor does not pass over the turbine as it is level with the water turbine. Instead, it runs directly into the main drainage pipe.
[4] Water head is the height of the water above the turbine.

Research Review

The research process for my artefact spanned a variety of methodologies and resource types and was integral throughout the design and development process. The two key methodologies were primary and secondary research. Primary research refers to the experiments I carried out to influence design decisions and to test prototypes. Secondary research refers to my research into the theory behind the principles and components I used in my design. The resources used in my secondary research included online articles and videos, books, online forums, MOOCs and interviews I conducted with experts in a range of fields. Each of these resource types served a different purpose, and thus, I chose my resource type carefully depending on the situation. Moreover, throughout the process I remained aware that no resource is completely free of any biases, even if these biases are unintentional. In this research review, I will illustrate the range of potential biases in the resource types I used and how I used each resource type to support my project.

Primarily, I was mindful to consider the credibility of the online articles and videos, books, MOOCs, and research papers which underpinned my research. Generally, this process involved examining the author's relevant qualifications and experience and the publication date of the source. This ensured both that the author has authority in the field they are writing or lecturing in and that the source's content is still relevant and accurate, as scientific theories and understanding can quickly become outdated. Depending on the credibility assurance provided by this initial analysis, I assessed whether it was necessary to cross-reference the source's content with that of other resources. Finally, for all sources, I checked that it was presented in an objective manner without spelling errors and with a comprehensive bibliography.

For example, I completed two MOOCs called *The Raspberry Pi Platform and Python Programming with the Raspberry Pi* and *Interfacing with the Raspberry Pi* from the University of California, Irvine (UCI) and instructed by Professor Ian Harris. While I was unable to verify the release date of these MOOCs, my research into Professor Harris found him to be a respected authority in the field of electronics and computing: he is the Vice Chair of Undergraduate Education in the Computer Science Department at UCI and has a PhD in computer science from the University of California, San Diego (Coursera, n.d.). Moreover, the content of the MOOCs, which explored how to use the Raspberry Pi as a microcontroller which interfaces with local circuits and back-end servers, was largely factual and widely researched, so any of the lecturer's personal agendas are unlikely to have permeated into the teaching.

During my initial analysis of the potential approaches to my aim of designing a hydroelectric power system, I referenced another secondary resource: David J.C MacKay's book, *Sustainable Energy – Without the Hot Air*. Despite MacKay's expertise in renewable energy as Chief Scientific Advisor to the UK Department of Energy and Climate Change and Regius Professor at Cambridge University, the book was published in over a decade ago, in 2008 (Wikipedia, 2021). Therefore, given the rapid changes in climate science and data, I deemed it necessary to cross-reference the figures cited by MacKay. This process was facilitated by MacKay's thorough referencing of his sources.

Another resource medium I used regularly were online articles and videos, as these provided pithy and accessible explanations for a range of topics. This was important as my artefact sits at the intersection between fluid dynamics, computing, electronics, mechanical engineering, and hydroelectric engineering, so it is not feasible to purchase a single book which covers all the theory and application know-how required by my project. However, although I remained vigilante in verifying the credibility of the authors of these online materials using the means identified earlier, I was conscious that the ease of falsifying information online meant this alone would not suffice. Therefore, for any online research, I always referenced at least three different sources to verify whether their explanations of a particular scientific topic were aligned with that of the original resource.

As aforementioned, I used several online forums to clarify certain specific questions that arose during the research and design process. Given the nature of online forums, I was incredibly cautious when considering the responses which I received. On certain platforms, such as *physics.stackexchange, engineering.stackexchange, raspberrypi.stackexchange, stackoverflow* and *quora*, it was often possible to view the accreditation and background of those who answered my questions. For example, in response to my question regarding the repulsion between electromagnets and permanent magnets, I received a detailed and very useful answer from Miguel Gonzalez, who is an electrical engineer and a Senior Developer at a medical engineering company called *AcuBase* (Miguel, 2021). Taking this information into account, alongside the responses from other members of the community to each answer, I made a case-by-case decision regarding the credibility of the answer. On the one hand, I was cautiously optimistic about the general accuracy of the answers I received on the online forums, as it seemed most of those answering were retired professors or life-long experts in the field. However, I was conscious that some members used the forums to promote a particular scientific agenda, which could lead to strong biases and inaccurate information.

I also interviewed several experts in a range of areas: Sam Williamson of Bristol University, Ashley Prescott of PowerPipe UK and Matthew Knight of Akro Valve. Due to their qualifications and influential positions, it was likely that each interviewee was a credible source. In particular, I was confident in the information prof. Williamson shared with me as he was a neutral stakeholder in my project, since he would not profit from any decisions I made, and as such there were no conflicts of interest which could interfere with his advice. However, as the latter two of the interviewees were senior representatives of a commercial business, it was vital to remain aware of marketing tactics that may impede the complete accuracy of their advice. These may include overstating or misrepresenting the capabilities of their product to fit my requirements or manipulating the presentation of data relating to their product.

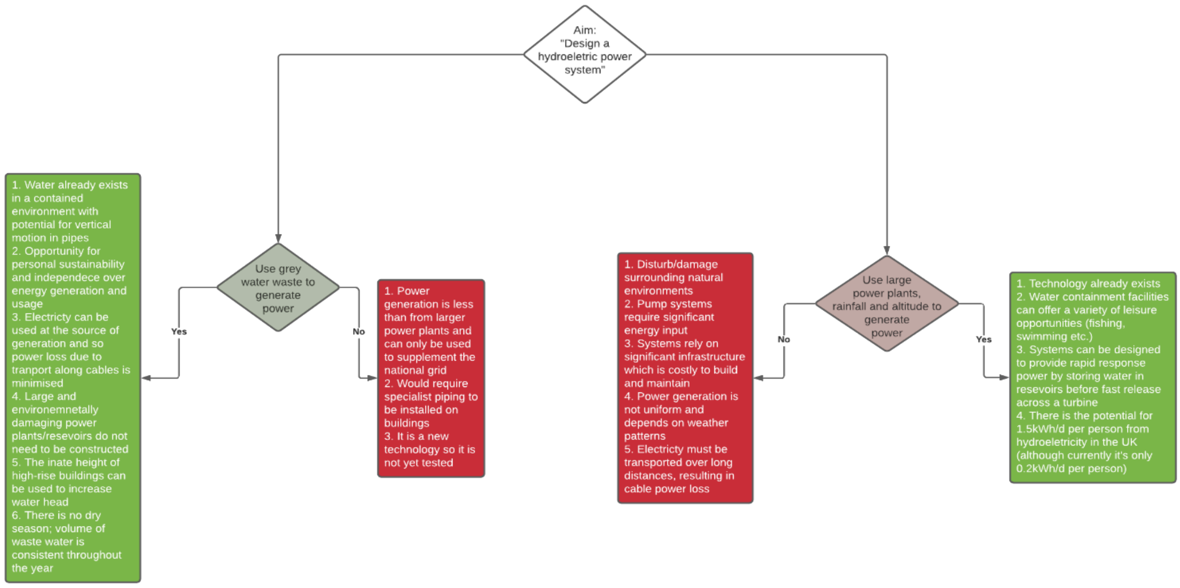
Finally, I carried out my own experiments to determine the requirements of my design and to test prototypes. For example, I simulated the weight of the water within the pipe to test the required strength of the pipe barrier mechanism. Throughout the development process, I regularly 3D printed and then tested prototypes for each aspect of the system. Following each experiment, I would make refinements to the design, and repeat the process until each aspect of the system worked as intended. Despite these experiments being primary research, I am aware that my desire to produce a certain result could have affected the impartiality of the conclusions I drew. However, throughout the iterative testing and development process I was conscious to avoid this by devising a clear method for each experiment before carefully realizing this method and recording the results.

Initial Design Analysis

I began the design process with an initial analysis of a range of possible solutions to the general engineering problem I had decided upon: *Design a hydroelectric power system*. I used a flow chart to analyze the viability of several solutions, scoring each one out of 40 with the following criteria:

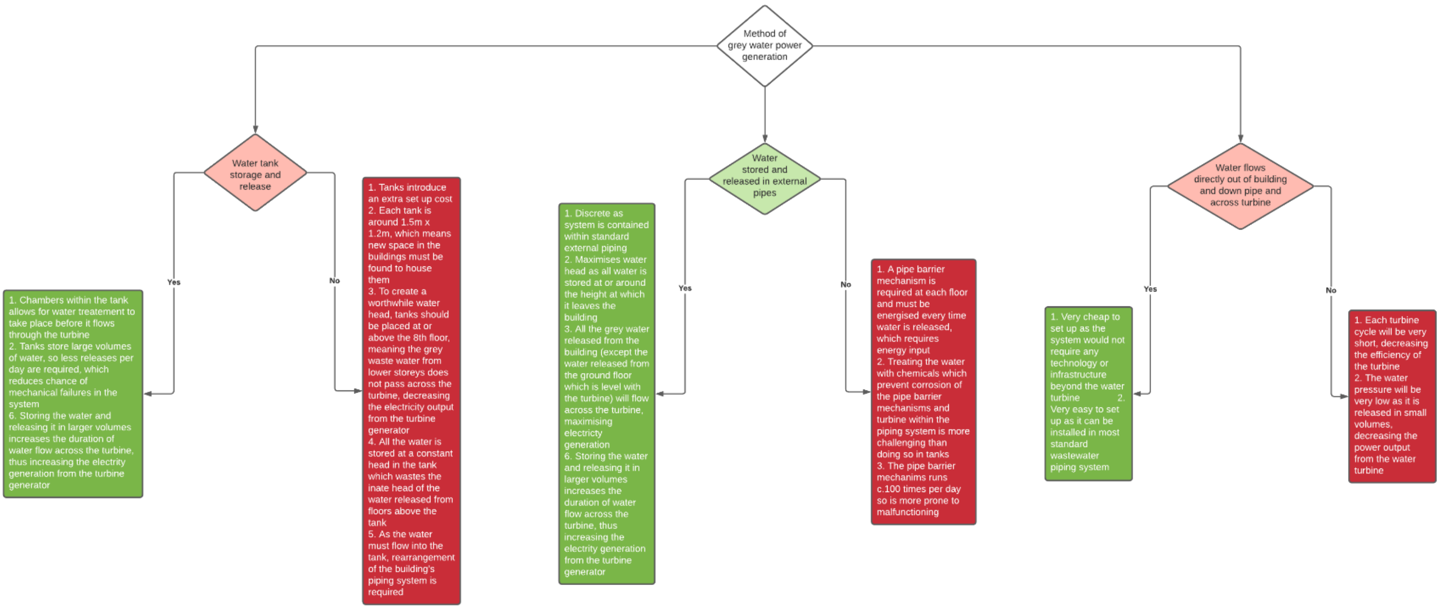
- *Personal interest*
- *Lack of existing solutions*
- *Electricity generation*
- *Feasibility*

Initially, I compared micro-hydropower systems, which harnessed energy from grey water in suburban environments, to large scale power plants, which harnessed energy from rainfall using reservoirs to store the water:



Criteria	Grey water	Power plant
Personal interest/10	8	6
Lack of existing solutions/10	9	3
Electricity generation/10	4	9
Feasibility/10	8	1
Total/40	29	19

Having decided that I would orientate my solution around generating electricity from *grey wastewater*, I compared several possible methods of doing so, and again scored each one out of 40 with the same criteria as aforementioned:



Criteria	**Water stored in tank**	**Water stored in pipe**	**Water released directly across turbine**
Personal interest/10	3	9	1
Lack of existing solutions/10	7	10	6
Electricity generation/10	6	8	0
Feasibility/10	8	8	10
Total/40	24	35	17

Based on the scores from this criteria-based analysis, it was clear that my hydroelectric generator design should utilise a method of storing the water in the building's grey water pipe[5]. On *page 24,* I also provide a quantative analysis of the electricity output from the tank-based system versus from the pipe based system to supplement and reaffirm the qualitive arguments presented above.

[5] This solution requires that the building's grey water pipe and black water pipe (which carries soiled water) are separate until they reach the drain. Older buildings are generally more readily suited for this system, as they tend to use separated grey and black water pipes. Newer buildings may require some configuration to their piping system as they often use a single soil stack pipe which both grey and black water flows down (Drainage Superstore, n.d.).

Mathematical Viability Analysis

Given the conclusion to my *Initial Design Analysis* on *page 9* and my subsequent decision to pursue a design for the hydroelectric generator which stored the water within the pipes, I conducted detailed mathematical viability analysis to identify the exact solution which yielded the most electricity generation. Firstly, I will determine how water *flow rate* (and therefore *water velocity*[6]) affects *head loss*, before exploring the impact of integrating a Waste Water Heat Recovery System (WWHRS) on the head loss. Finally, I will present a mathematical comparison of the electricity generation from a tank-based system versus my proposed system of storing the water within the grey water pipe.

Analysis is based on the installment of each system on a building with the following properties:

- *Building height: 84.3m*
- *No. floors: 26*
- *No. apartments: 150*
- *Circular grey water pipe height*[7]*: 81.2m*
- *External diameter of grey water pipe: 80mm*
- *Internal diameter of grey water pipe: 76mm*
- *Cross-sectional area of grey water pipe: 0.00454* m^2
- *Absolute roughness of grey water pipe*[8]*:* 0.0015×10^{-3} *m*

---

[6] Note that flow rate and velocity are *not* the same thing, but they are inextricably related by the equation Q=AV, where Q is flow rate, A is the cross-sectional area of the flow and V is the velocity of the flow. Therefore, for a constant cross-sectional area, such as a pipe, flow rate and velocity are directly proportional to each other.
[7] The uppermost floor (the 26th floor) would not require a grey water pipe to extend above its floor.
[8] Assumes pipe material is PVC (The Engineering Toolbox, 2019).

Effect of water flow rate on head loss

Before I explain how water flow rate impacts the head loss in a system, it is important to understand why it is vital to consider head loss when designing a hydroelectric system. The standard equation for calculating hydropower is:

$$P = Q \cdot g \cdot H_{net} \cdot \eta$$

- P = power, measure in Watts (W)
- Q = flow rate, measured in liters per second (lps)
- g = gravitational constant, which is 9.81 ms^{-2}
- H_{net} = net water head[9]
- n = product of all the component efficiencies[10]

From this equation, it is evident that an accurate understanding of the *net head* is necessary to determine the power output from a hydroelectric system. Water head is a measure of water pressure, and *gross water head* is given by the vertical height of the water above the turbine. During the water's motion through a system, a variety of factors, such as friction between the water and the pipe, can reduce this head; this phenomenon is called **head loss** (Engineering Library, n.d.).

During my initial analysis, I naively assumed that head loss would be minimal (c.10%) regardless of flow rate and velocity, and so would not be a significant determiner of my design choices. However, during the interview I conducted with *Dr. Sam Williamson*[11] of Bristol University, Dr. Williamson explained that "the faster the water's going through [the pipe], the increased amount of friction loss that you've got in your pipe". Following his suggestions, I decided to accurately determine the extent to which flow rate and velocity would affect head loss.

[9] The net is equal to the gross head minus any head losses.
[10] The components concerned are typically the turbine, drive system and generator (Renewables First, 2012).
[11] Dr. Sam Williamson is an expert in micro hydropower systems and generously gave up almost an hour of his time to discuss design considerations in hydroelectric systems.

Head loss *without* flow control

To determine the head loss created by the uncontrolled flow of the water in the pipe, I will use the following four equations:

1. **Darcy-Weisbach equation** for head loss[12]: $h_f = f_D \frac{L}{D} \frac{V^2}{2g}$

   - h_f = head loss
   - f_D = Darcy's friction constant
   - L = length of pipe
   - D = hydraulic diameter of pipe[13]
   - V = velocity of water
   - g = gravitational constant

2. **Bernoulli's equation**[14]: $h - h_f = \frac{V^2}{2g}$

   - h = gross head
   - h_f = head loss
   - V = velocity of water
   - g = gravitational constant

3. **Reynold's equation**: $Re = \frac{VD}{v}$

   - Re = Reynold's number[15]
   - V = velocity of water
   - D = hydraulic diameter of pipe
   - v = kinematic viscosity of water

4. **Relative pipe roughness** equation: $\text{Relative pipe roughness} = \frac{\text{Absolute roughness}}{\text{Hydraulic diameter}}$

5. **Flow rate and velocity** equation: $Q = A \cdot V$

   - Q = flow rate of water
   - A = cross-sectional area (of pipe)
   - V = velocity of water

---

[12] The Darcy-Weisbach is the most accurate equation for calculating head loss in pipes (The Engineering Toolbox, 2019).

[13] Hydraulic diameter of a circular pipe is equal to its diameter (Wikipedia, 2020).

[14] This equation is a derivative of Bernoulli's original equation where the atmospheric pressure change between the top and the bottom of the pipe is assumed to be negligeable (Physics Stackexchange, 2021).

[15] Reynold's number describes the flow type of water. A low Reynold's number indicates laminar flow and a high Reynold's number indicates turbulent flow (Labmate, 2014).

Calculations:

Note that these calculations all assume maximum gross head (81.2m).

1. Equate ***Bernoulli's equation*** with ***Darcy-Weisbach equation***:

$$\frac{V^2}{2g} = h - f_D \frac{L}{D}\frac{V^2}{2g}$$

2. Substitute *h* for *L*, as the height of a vertical pipe is equal to its length:

$$\frac{V^2}{2g} = h - f_D \frac{h}{D}\frac{V^2}{2g}$$

3. Rearrange the above equation to make *V* the subject:

$$V = \sqrt{\frac{2gh}{1 + \frac{f_D + h}{D}}}$$

4. Substitute the above equation into ***Reynold's equation***:

$$Re = \frac{D}{v}\sqrt{\frac{2gh}{1 + \frac{f_D + h}{D}}}$$

5. Substitute actual values into the above equation[16]:

$$Re = \frac{0.076}{1.004 \cdot 10^{-6}}\sqrt{\frac{2 \cdot 9.81 \cdot 81.2}{1 + \left(\frac{81.2}{0.076}\right) f_D}}$$

$$Re = \frac{3.021 \cdot 10^6}{\sqrt{1 + (1089.2 \cdot f_D)}}$$

[16] The values for *D* and *h* substituted into the equation are in accordance with the constant properties of the building upon which all the analysed systems are based upon, as outlined on *page 11* (0.076m and 81.2m respectively). The kinematic viscosity of water (*v*) at 20° is 1.004 x 10^{-6} ms^{-2} (Engineers Edge, 2014).

6. Determine ***relative pipe roughness***:

$$\text{Relative pipe roughness} = \frac{0.0015 \cdot 10^{-3}}{0.076}$$

$$\text{Relative pipe roughness} = 1.97 \cdot 10^{-5}$$

7. Use the ***Moody chart*** (figure 1), which relates the ***Darcy friction factor*** (f_D), ***Reynold's number*** and ***relative pipe roughness***, and an iterative approach to determine the correct value for f_D and Re :

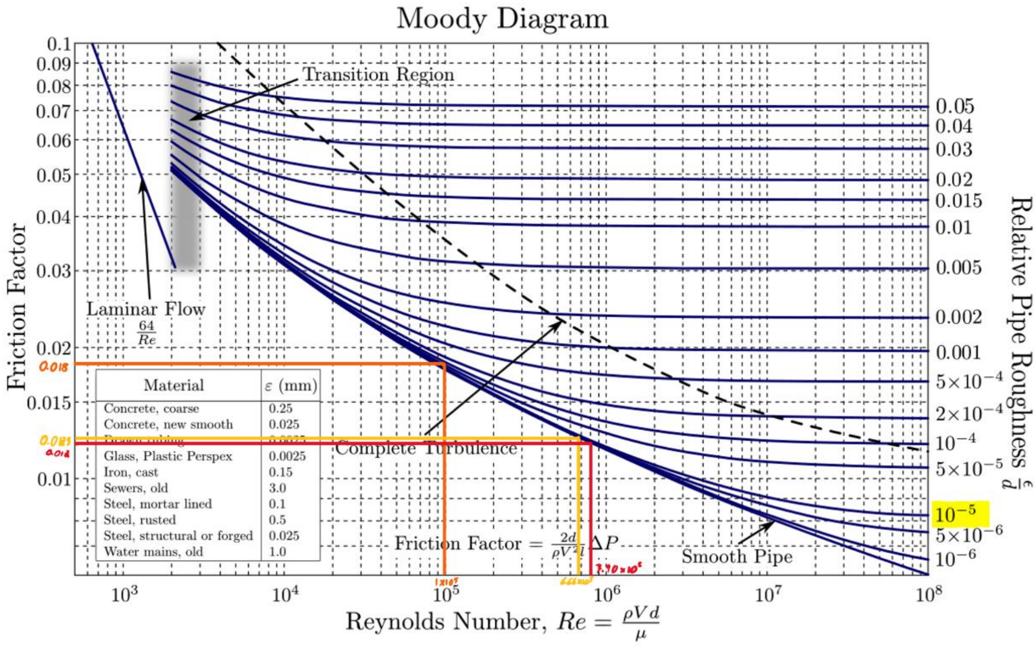


Figure 1, Mood diagram (Collins and Beck, 2016)

1st iteration:

$$\text{Let } Re = 10^5$$

$$\text{From chart, } f_D = 0.018$$

2nd iteration:

$$f_D = 0.018,\ Re = \frac{3.021 \cdot 10^6}{\sqrt{1 + (1089.2 \cdot 0.018)}}$$

$$Re = 6.66 \cdot 10^5$$

$$\text{From chart, } f_D = 0.0125$$

3rd iteration:

$$f_D = 0.0125,\ Re = \frac{3.021 \cdot 10^6}{\sqrt{1+(1089.2 \cdot 0.0125)}}$$

$$Re = 7.90 \cdot 10^5$$

$$\text{From chart, } f_D = 0.012$$

4th iteration:

$$f_D = 0.012,\ Re = \frac{3.021 \cdot 10^6}{\sqrt{1+(1089.2 \cdot 0.012)}}$$

$$Re = 8.05 \cdot 10^5 \approx 7.90 \cdot 10^5$$

From this iterative process, I found the ***Darcy friction factor*** (*f_D*) for the system = *0.012* and the ***Reynold's number*** = *7.90* x 10^5.

8. Rearrange ***Reynold's equation*** to make *V* the subject and substitute in the newly found value for ***Reynold's number***:

$$V = \frac{Re \cdot \nu}{D}$$

$$V = \frac{7.90 \cdot 10^5 \cdot 1.004 \cdot 10^{-6}}{0.076}$$

$$V = 10.44\ ms^{-1}$$

9. Use this value for velocity to determine the flow rate of the water:

$$Q = A \cdot V$$

$$Q = 0.00454 \cdot 10.44$$

$$Q = 0.047\ m^3 s^{-1}$$

10. Substitute the value for velocity into the ***Darcy-Weisbach equation*** for head loss to find the head loss due to friction in the pipe:

$$h_f = f_D \cdot \frac{L}{D} \cdot \frac{V^2}{2g}$$
$$h_f = 0.012 \cdot \frac{81.2}{0.076} \cdot \frac{10.44^2}{2 \cdot 9.81}$$
$$h_f = 71.2\ m$$

Further calculations[17]:

To recalculate the head loss using an alternative method, I will use the **Hazen-Williams equation** for head loss:

$$h_{100m} = \frac{0.2083 \cdot \left(\frac{100}{c}\right)^{1.852} \cdot Q^{1.852}}{D^{4.8655}}$$

- h_{100m} = head loss in meters of water per 100m of pipe
- c = Hazen-Williams roughness constant
- Q = flow rate of water
- D = hydraulic diameter of pipe

1. Use the ***Hazen-Williams equation*** to determine head loss due to friction in pipe[18]:

$$h_{100m} = \frac{0.2083 \cdot \left(\frac{100}{150}\right)^{1.852} \cdot 0.047^{1.852}}{0.076^{4.8655}}$$
$$h_{100m} = 95.2m$$
$$h_{81.2m} = 77.3m \approx 71.2m$$

[17] Given the extensive calculations required by the Darcy-Weisbach approach for determining head loss, I decided to use the value for flow rate which I had found using the Darcy-Weisbach approach to recalculate the head loss in the system using the Hazen-Williams equation for head loss. This would allow me to compare the results from each equation and determine whether the latter would provide a suitable approximation, as the Hazen-Williams equation is generally less accurate than the Darcy-Weisbach equation (The Engineering Toolbox, 2019).

[18] The value for Q substituted into the equation is in accordance with results from the previous calculations using the Darcy-Weisbach equation ($0.047 m^3 s^{-1}$) and the value for D substituted into the equation is in accordance with the constant properties of the building upon which all the analysed

The Hazen-Williams equation gave a value for head loss with difference of 8.6% from the value given by the Darcy-Weisbach approach. As this variation is fairly minimal, for future calculations of head loss I will use the more compact Hazen-Williams equation.

Conclusion:

Without any means of flow rate control, these calculations show that the head loss would be around *71.2*m in a system with a gross head of *81.2*m. This high level of head loss would drastically reduce the hydropower potential of the system and therefore is untenable. Moreover, as Dr. Williamson explained, "you want to have lower power for longer" in the turbine, as turbines increase in price as their output power capabilities increase. Below is an explanation of the physics behind this logic:

$$Energy = Power \cdot Time$$

- Therefore, electricity output can be increased either by increasing the power output or by increasing the operation time of the turbine;
- Increasing power output from the turbine is costly and requires an increased flow rate, which in turn increases the head loss in the system and thus, as discussed above, decreases the hydropower potential;
- Therefore, it is ideal to achieve a lower power output (between 500-1000W) for a sustained period;
- As $P = Q \cdot g \cdot H_{net} \cdot \eta$ power output can be reduced by decreasing the flow, which in turn decreases the head loss too.

Given the findings above, I realized the importance of controlling and reducing the flow rate in the system. Having spoken to Matthew Knight[19] of *Akro Valves*, I chose to use a **wafer valve** to control the flow rate in the system; further details about the wafer valve can be found on *page 61*. Using *PowerSpout's*[20] online advanced calculator (PowerSpout, n.d.), I found that, for my configuration, the maximum viable flow rate is 3.1 lps. Therefore, the wafer valve will be used to maintain a constant water flow rate of 3.1 lps ($0.0031m^3s^{-1}$).

systems are based upon (0.076m), as outlined on *page 11*. The Hazen-Williams roughness constant (*c*) for PVC pipe is 150 (The Engineering Toolbox, n.d.).

[19] Matthew Knight is an ambassador for STEM in schools and very generously supported my project by donating an array of components for the turbine system.

[20] *PowerSpout* is a manufacturer of micro-hydropower turbines. As is detailed on *page 18*, I have designed my hydroelectric system to use the *PowerSpout* Pelton HP turbine.

Head loss *with* flow control

The head loss due to the presence of the wafer valve in the pipe is taken to be 14m[21]. Therefore, considering only the head loss due to the wafer valve, the *net water head* in a pipe with a *gross water head* of 81.2m, is 67.2m.

To calculate the supplementary head loss due to friction, I will use the **Hazen-Williams equation** for head loss:

$$h_{100m} = \frac{0.2083 \cdot \left(\frac{100}{c}\right)^{1.852} \cdot Q^{1.852}}{D^{4.8655}}$$

- h_{100m} = head loss in meters of water per 100m of pipe
- c = Hazen-Williams roughness constant
- Q = flow rate of water
- D = hydraulic diameter of pipe

Calculations:

Note that these calculations all assume a net of 67.2m due the head loss caused by the wafer valve and a constant flow rate of $0.0031m^3s^{-1}$ controlled by the wafer valve.

1. Use the ***Hazen-Williams equation*** to determine head loss due to friction in pipe[22]:

$$h_{100m} = \frac{0.2083 \cdot \left(\frac{100}{150}\right)^{1.852} \cdot 0.0031^{1.852}}{0.076^{4.8655}}$$

$$h_{100m} = 0.6194m\ per\ 100m$$

$$h_{67.2m} = 0.416m$$

2. Sum head loss due to wafer valve and head loss due to friction in pipe to find total head loss:

[21] A head loss of 14m is equivalent to a pressure loss of 140kPa, which is the standard pressure loss indicated by the manufacturer of the wafer valve, *Maric Flow Control* (Maric Flow Control, n.d.) and the value advised by Matthew Knight of *Akro Valves*.

[22] The value for Q substituted into the equation is in accordance with the chosen flow rate which is set and controlled by the wafer valve ($0.0031m^3s^{-1}$) and the value for D substituted into the equation is in accordance with the constant properties of the building upon which all the analysed systems are based upon (0.076m), as outlined on *page 11*. The Hazen-Williams roughness constant (c) for PVC pipe is 150 (The Engineering Toolbox, n.d.).

$$h_{valve} = 14m$$
$$h_f = 0.416m$$
$$h_{total} = 14.416m$$

Conclusion:

Despite the inherent head loss caused by the presence of the wafer valve in the pipe system, when the flow rate is controlled, the head loss caused by the motion of the water in the pipe reduced by 81.4% compared with when flow rate is not controlled. **These calculations made very clear to me the necessity of controlling the flow rate of the water in the pipe.**

Impact of WWHRS on head loss

Intent on harnessing the maximum amount of energy from the grey water, I was very interested by the possibility of integrating WWHR into the hydroelectric generator system. WHHRS is a technology which can be installed below showers to recover up to around 68% of the heat from grey water and transfer this heat to incoming clean water running into showers (Recoup, n.d.). As Ashley Prescott[23] of *PowerPipe* explained to me, as the heated grey water falls through the WHHR pipe[24], the surface tension creates a thin film inside the pipe, allowing for efficient heat transfer with the cold mains water which runs through spiralling copper pipes on the outside of the pipe, as shown in *figure 2*:

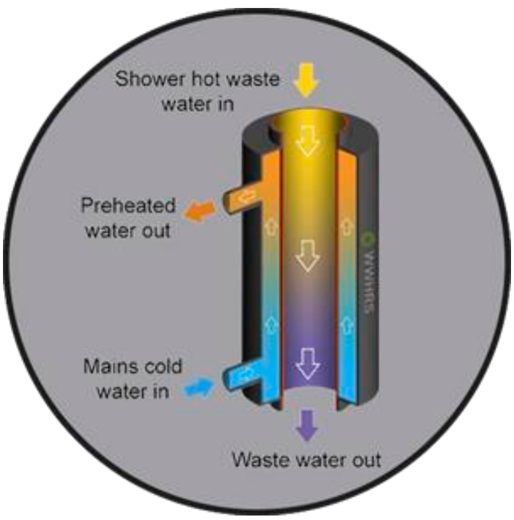


Figure 2, WWHR Pipe (Recoup, n.d.)

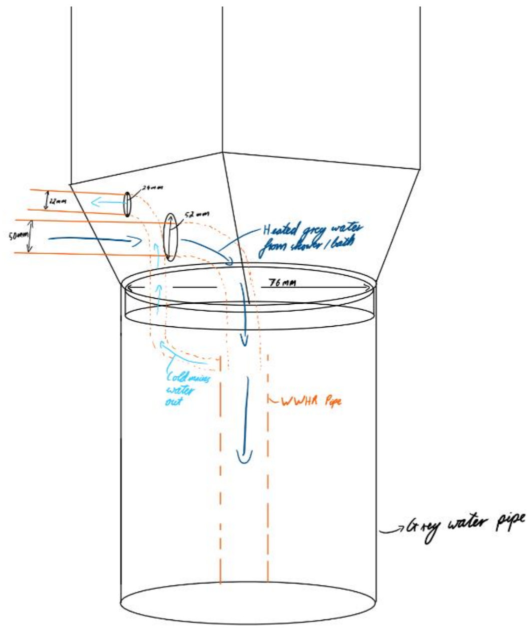


Figure 3, WWHR integration

Figure 3 shows my proposed design for the integration of a WWHR pipe into the hydroelectric generator system. A WWHR pipe with diameter of 0.05m[25] would be contained within each section of the main grey water pipe[26] to enable heat recovery from shower wastewater leaving apartments on each floor. The shower wastewater would flow directly into the WWHR pipe (as

[23] Ashley Prescott is the UK director at PowerPipe, a company who manufacture WWHR pipes. I spoke to Ashley as part of my research into the possibility of integrating WWHR into the my hydroelectric system.
[24] Each section of WWHR pipe varies in length from just under 1m to just over 2m (Powerpipe, n.d.).
[25] I have chosen to analyse the PowerPipe WWHR pipe with the smallest diameter to maximise the cross-sectional area of the region of the water flow and minimise the head loss due to friction in the pipe.
[26] As mentioned previously, each section of piping refers to a section of the main grey water pipe (equal to the height of one storey) which is temporarily separated from the rest of the pipe using a barrier mechanism.

indicated by the dark blue arrows in *figure 3*) and would leave through bottom of the WWHR pipe to rejoin the section of grey water pipe which the WWHR pipe is contained within. However, before committing to this implementation of a WWHR pipe, I assessed the impact it would have on the water head loss.

To calculate the head loss due to friction, I will use the following two equations:

1. **Hazen-Williams equation** for head loss: $h_{100m} = \frac{0.2083 \cdot \left(\frac{100}{c}\right)^{1.852} \cdot Q^{1.852}}{D^{4.8655}}$

   - h_{100m} = head loss in meters of water per 100m of pipe
   - c = Hazen-Williams roughness constant
   - Q = flow rate of water
   - D = hydraulic diameter of pipe

2. **Hydraulic diameter equation**[27]: $D_h = \frac{4A}{P}$

   - D_h = hydraulic diameter
   - A = cross-sectional area of water flow
   - P = 'wetted' perimeter of pipe

Calculations:

Note that these calculations all assume a net of 67.2m due the head loss caused by the wafer valve and a constant flow rate of $0.0031m^3s^{-1}$ controlled by the wafer valve.

1. Use the ***Hydraulic diameter equation*** to find the hydraulic diameter of the pipe annulus[28]:

$$D_h = \frac{4\left(\left(0.038^2 \cdot \pi\right) - \left(0.025^2 \cdot \pi\right)\right)}{0.05\pi + 0.076\pi}$$

$$D_h = 0.026m$$

[27] The hydraulic diameter is a characteristic length which is used to calculate the diameter of the pipe through which liquid is flowing. As mentioned previously, for circular tubes, the hydraulic diameter is equal to the standard diameter of a circle. However, the hydraulic diameter is an important measurement when considering rectangular pipes or pipes with an annulus shape (The Engineering Toolbox, n.d.).

[28] An annulus is the region between two concentric circles (Collins Dictionary, n.d.). Here, the annulus is the region between the grey water pipe and the WWHR pipe contained within in, as shown in *figure 3*. The value for A substituted into the equation is the cross-sectional area of the annulus and the value for P substituted into the equation is the combined perimeter of the internal circumference of the grey water pipe and the external circumference of the WWHR pipe.

2. Use the ***Hazen-Williams equation*** to determine head loss due to friction in pipe[29]:

$$
\begin{aligned}
h_{100m} &= \frac{0.2083 \cdot \left(\frac{100}{150}\right)^{1.852} \cdot 0.0031^{1.852}}{0.026^{4.8655}} \\
h_{100m} &= 114.42m \; per \; 100m \\
h_{67.2m} &= 76.89m
\end{aligned}
$$

Conclusion:

The results of these calculations show the impossibility of integrating a WWHR pipe into the system; the hydraulic diameter of the annulus created by the WWHR pipe contained inside the grey water pipe (0.026m) is such that the Hazen-Williams equation for head loss returns a great head loss than the gross head. The only possible means to increase the hydraulic diameter of the annulus whilst maintaining the integrated WWHR pipe would be to increase the diameter of the grey pipe. However, as Dr. Williamson emphasized, "a bigger pipe costs more money", which is undesirable in a system aiming to reduce electricity expenses. Moreover, as the diameter of the grey pipe is increased, the compact nature of the system is lost. For these reasons**, I decided to abandon the idea of integrating a WWHR into the hydroelectric generator system**.

[29] The value for *Q* substituted into the equation is in accordance with the chosen flow rate which is set and controlled by the wafer valve ($0.0031m^3s^{-1}$) and the value for *D* substituted into the equation is the newly calculated hydraulic diameter for a grey pipe with an integrated WWHR pipe (0.026m). The Hazen-Williams roughness constant (*c*) for PVC pipe is 150 (The Engineering Toolbox, n.d.).

Water tank system versus pipe water storage system

With the hope of reaffirming the qualitative comparison on *page 10*, I carried out a detailed quantitative analysis of the electricity generated by a system that uses a water tank situated half-way up the building to store the grey water *versus* my proposed method of storing the water within the grey water pipe. For both breakdowns, I have used the constant flow rate value determined on *page 18* (3.1 lps) and the value of hydraulic diameter is in accordance with the constant properties of the building upon which all the analysed systems are based upon (0.076m), as outlined on *page 11.*

Tank system

Note that analysis assumes a water tank is situated half-way up the building, meaning the grey water from half of the total number of apartments in the building (i.e. 75 apartments) will flow into and be stored in tank.

Constant values:

Number of bedrooms per apartment:	2	
Daily water demand per bedroom in 2 bedroom dwelling (litres)*:	130	litres
Grey water waste as a percentage of total water waste**:	76%	
Internal hydraulic diameter of grey water pipe (metres)***:	0.076	metres
Generator efficiency****:	0.93	%
Total daily grey water demand for apartement building (litres and kg):	14820	litres and kg
Total daily grey water demand for apartement building (cubic metres):	14.82	cubic metres
Total volume of water storage tank (litres)*****:	100	litres
Flow rate of water (lps)^:	3.1	lps
Velocity of water (metres/sec)^^:	0.683	metres/sec

Power output^^^:

Power output from turbine (kW)^^^^:	0.63	kW
Power output from generator (kW):	0.5850	kW
Volume per release (cubic metres):	0.100	cubic metres
Number of release per day:	148	
Time per release (sec)^*:	32.3	sec
Total turbine operating time per day (hours)^*:	1.33	hours
Electricty generated per day (kWh):	0.78	kWh

Assumptions:

+	Each floor/section of piping is 3.12m in height, but 0.329m of each section of piping is dedicated to the pipe barrier mechanism.
++	The grey waste water from the floors in the top half of the building is stored in a tank situated half-way up the building, before being released over the turbine when it becomes full. Therefore, the total maximum water head is equal to the height of the pipe (40.6).
+++	Net head takes into account head loss due to the wafer valve (taken to be 7m), friction caused by the pipe (0.6194m per 100m) and the 90 degree long radius pipe bend (0.0112m) at the bottom of the main downpipe before the water enters the turbine.
*	See Figure 1 (The Institute of Plumbing, 2002, as cited in The College of Estate Management, n.d.)
**	See figure 2 (Waterwise, 2012, as cited in The College of Estate Management, n.d.)
***	Internal hydraulic diameter is given by 4A/P, where A is the cross-sectional area of water flow and P is the wetted perimeter of the water flow.
****	(Renewables First, 2012)
^	Flow rate is maintained at a constant value of around 3.1lps (wafer valve is designed to allow up to 3.3lps) to reduce head loss due to friction in pipe and ensure no energy is wasted, as the turbine is limited to 3.1lps in the given configuration.
^^	Velocity of water is found using the equation Q=AV, where Q is the flow rate (3.1lps) and A is the cross-sectional area of the pipe
^^^	Net water head is a constant of 66.78 metres (net water head)
^^^^	Turbine in reference is the PowerSpout Pelton HP turbine. These values for power ouput are obtained from PowerSpout's online advanced calculator.
^*	Assuming flow rate of turbine has no effect on the velocity of the water's final velocity. Formula used to calculate operating time is T = V/Q (Lumen, 2019).

Pipe water storage system

Constant values:

Height of pipe (metres):	81.20	metres
Number of floors in apartment building:	26.00	
Height of each floor/section of pipe (metres)+:	3.12	metres
Max gross water head (metres)++:	81.12	metres
Min total water head (metres)+++:	3.12	metres
Max net water head (metres)++++:	66.78	metres
Min net water head (metres)+++++:	3.10	metres
Number of apartments with grey water involved in the system:	150.00	
Number of bedrooms per apartment:	2.00	
Daily water demand per bedroom in 2 bedroom dwelling (litres)*:	130.00	litres
Grey water waste as a percentage of total water waste**:	0.76	
Internal hydraulic diameter of grey water pipe (metres)***:	0.08	metres
Generator efficiency****:	0.93	%
Total daily grey water demand for apartement building (litres and kg):	29640.00	litres and kg
Total daily grey water demand for apartement building (cubic metres):	29.64	cubic metres
Total internal volume of each section of pipe (cubic metres)*****:	0.01	cubic metres
Total internal volume of grey water waste pipe (cubic metres):	0.33	cubic metres
Flow rate of water (lps)^:	3.10	lps
Velocity of water (metres/sec)^^:	0.68	metres/sec

Maximum power output^^^:

Power output from turbine (kW)^^^^:	1.06	kW
Power output from generator (kW):	0.9895	kW
Volume per release (cubic metres)^*:	0.329	cubic metres
Number of release per day:	90	
Time per release (sec)^**:	106.2	sec
Total turbine operating time per day (hours)^**:	2.66	hours
Electricty generated per day (kWh):	2.63	kWh

Minimum power output^***:

Power output from turbine (kW)^^^^:	0.03	kW
Power output from generator (kW):	0.0316	kW
Volume per release (cubic metres)^^*:	0.013	cubic metres
Number of release per day^^*:	2342	
Time per release (sec)^^**:	4.1	sec
Total turbine operating time per day (hours)^^**:	2.66	hours
Electricty generated per day (kWh):	0.084	kWh

Assumptions:

++	The water from each floor is stored in the pipe section of the floor below. Therefore, the total maximum water head is equal to the height of the pipe (81.2).
+++	The water flow from the ground floor apartment is level with the water turbine system, so the grey waste water from the ground floor is not passed across the turbine. Therefore, the first floor of the building is the first floor whose waste water is stored and passed across the turbine. The waste water from the first floor is stored in the pipe section which runs between the ground floor and the first floor, making the minimum total water head equal to the height of one section of piping (3.12m).
++++	Max net head takes into account head loss due to the wafer valve (taken to be 14m), friction caused by the pipe (0.6194m per 100m) and the 90 degree long radius pipe bend (0.0112m) at the bottom of the main downpipe before the water enters the turbine.
+++++	Min net head takes into account head loss due friction caused by the pipe (0.6194m per 100m) and the 90 degree long radius pipe bend (0.0112m) at the bottom of the main downpipe before the water enters the turbine.
*	See Figure 1 (The Institute of Plumbing, 2002, as cited in The College of Estate Management, n.d.)
**	See figure 2 (Waterwise, 2012, as cited in The College of Estate Management, n.d.)
***	Internal hydraulic diameter is given by 4A/P, where A is the cross-sectional area of water flow and P is the wetted perimeter of the water flow.
****	(Renewables First, 2012)
*****	Assumes maximum height of water in each pipe section is 2.79m. This is a conservative assumption based on the fact that each pipe section is 3.12m in height with 0.329m dedicated to the pipe barrier section.
^	Flow rate is maintained at a constant value of around 3.1lps (wafer valve is designed to allow 3lps) to reduce head loss due to friction in pipe and ensure no energy is wasted, as the turbine is limited to 3.1lps in the given configuration.
^^	Velocity of water is found using the equation Q=AV, where Q is the flow rate (3.1lps) and A is the cross-sectional area of the pipe
^^^	Assuming net water head is a constant of 66.78 metres (max net water head)
^^^^	Turbine in reference is the PowerSpout Pelton HP turbine. These values for power ouput are obtained from PowerSpout's online advanced calculator.
^*	Assuming entire pipe is full at the time of each release
^**	Assuming flow rate of turbine has no effect on the velocity of the water's final velocity and assuming that entire pipe is full at the time of each release. Formula used to calculate operating time is T = V/Q.
^***	Assuming net water head is a constant of 3.12m.
^^*	Assuming only lowermost section of pipe is full at the time of each release.

Conclusion:

The above calculations show that **my proposed method of storing the water within the grey water pipe can generate *up to* 3.4 times the electricity per day that a tank-based system would be able to generate in a day**[30], when installed on the same building with the same number of occupants. This finding reaffirms my conclusions stated on *page 10* and therefore I decided to pursue the pipe water storage design for my hydroelectric generator. It should be noted that the electricity generation capabilities of the pipe water storage system are dependent on the distribution of the floors from which the grey water is released; if water usage, and therefore grey wastewater release, is significantly greater towards the lower half of the building, then the electricity generated by the turbine will be less than the 2.63 kWh/day stated in the *Maximum power output* section of the *Pipe water storage system*. However, as detailed on *pages 48-54*, my system uses sensors and actuators to redistribute the water within the pipe to ensure that the electricity generation is always close to the maximum of 2.63 kWh/day.

[30] This factor of 3.4 assumes maximum power output from the turbine installed in the pipe water storage system for the entirety of the day.

Pipe Barrier Mechanism

Keywords:

- *Section of pipe: the grey water pipe is divided into sections by the pipe barrier mechanisms, with the height of each section equal to the height of one story of the apartment building.*
- *Extended state of sliding barrier: when the sliding barrier is at maximum extension, it is in contact with the edge of the barrier mechanism housing and the barrier mechanism is therefore in the closed position.*

Having decided to use a pipe water storage system for my hydroelectric generator based on my findings in the section titled *Mathematical Viability Analysis*, the first step was to design the mechanism to store the water within the pipe. There were four key criteria that this mechanism had to fulfill:

Dyanamic	• The key element of the pipe water storage system is its ability to redistribute the water within the pipe to achieve maximum water head and volume during each release of water across the turbine.
Reliable	• Given that the pipe is storing water, it is essential that this water can be released quickly when necessary to ensure there is always sufficient space within the pipe to accomodate incoming grey water, else severe flooding issues could arise. On the flip side, it is vital that the mechanism is reliable and quick at closing at the appropriate time to ensure optimal functioning of the hydroelectric generator system.
Watertight	• The mechanism must effectively seal when it stores the water within the pipe, else the hydroelectric generator system will not be able to work at maximum capacity as water head will be lost unecessarily.
Strong	• The mechanism must be capable of supporting the volume of grey water stored in one section of the grey pipe.

Given these conditions, I considered three fundamentally different designs for the pipe barrier mechanism. Below is a summary of the three different design concepts, with an explanation of the whether they met each criterion:

Fundamental design ideas

Sketches and notes of each design can be seen in the document titled 'Activity Log - Visual'

Mitre gate, 2 torsion springs, electromagnet, permanent magnet and ballcock (V1)

- Two leaves of a mitre gate each attached to a torsion (angular) spring. When the ballcock rises to a certain height, an energy to release electromagnet is energised and one leaf slides slides back, allowing the mitres gate to rotate downwards. When water flow rate past the mitre gate reaches a sufficiently low level, the torsion spring forces each leaf to rebound upwards into the closes position.
- Dynamic - miter gate can rotate upwards and downwards, allowing the barrier mechanism to open and close.
- Reliable - as the rebound motion of the mitre gate is dependent on two torsion springs, whose speed, force and time of motion is variable depending on the water flow across each leaf, it is highly likely each leaf may collide with each other or fail to return to their closed position. Moreover, - as the leaves are maintained in the closed position by a permanent magnet and an electromagnet, the presence of water in the pipe may damage these magnets and destroy the barrier
- Watertight - as the two leaves are maintained in the closed position by a permanent magnet and an electromagnet which must have face to face contact to ensure the barrier is watertight, it is very likely that during the rebound motion of the leaves, they won't reallign and water will be able to penetrate the barrier.
- Strong - the pivot for each rotating leaf creates a point of weakness that could break after many cycles of. However, the mechanism's miter gate shape is incredibly strong when it is in the closed position.

Rotating leaf, sliding leaf, torsion spring and electromagnet (V2)

- Rotating leaf attached to a torsion spring is maintained the closed position by a second leaf which can slide linearly. When the electromagnet is energised, sliding leaf retracts and rotating leaf swings downwards, allowing water to flow past. When water flow rate past the mitre gate reaches a sufficiently low level, the torsion spring forces the rotating leaf to rebound upwards into the closes position.
- Dynamic - rotating leaf can rotate upwards and downwards, allowing the barrier mechanism to open and close.
- Reliable - as the rotations of the rotating barrier are dependent on the force of the torsion spring, if the spring does not return to exactly the same position, the rotating leaf most probably won't line up with the sliding leaf, and so the barrier mechanism will not be able to return to its closed position. However, as the electromagnet which controls the retraction of the sliding barrier is removed from the water flow, the electromagnet is unlikely to malfunction, so the mechanism should remain sturdy. Moreover, the mechanism's miter gate shape is incredibly strong when it is in the closed position.
- Watertight - as the two leaves are miantained in closed position by two contact points, it is very likely that the rotating leaf will not make watertight contact with the sliding leaf as it rebounds to the closed position, allowing water to pentrate the barrier.
- Strong - the pivot for each rotating leaf creates a point of weakness that could break after many cycles of. However, the mechanism's miter gate shape is incredibly strong when it is in the closed position.

Sliding barrier, energy to release electromagnet, energy to hold electromagnet (V3)

- Sliding barrier is held in closed position by magnet attraction between an energy to release electromagnet and a permanent magnet attached to the sliding barrier. When the energy to release electromagnet is energised, the sliding barrier slides backwards, allowing water to flow. To close the barier an energy to hold electromagnet is energised, which repels the permanent magnet, moving the sliding barrier back to the closed position.
- Dynamic - sliding barrier can move linearly backwards and forwards to open and close the barrier.
- Reliable - the motion of the sliding barrier is controlled by two electromagnets whose behaviour is predictable and consistent regardless of water flow. Therefore, the sliding barrier will open and close reliably.
- Watertight - as the motion of the sliding barrier is controlled by electromagnets, contact and seal that it makes with the edge of the pipe when it is closed position will be consistent. However, it is necessary to ensure that this seal is secure enough to ensure no water leaks through the barrier.
- Strong - the sliding barrier is held in place by a sturdy structure which it retracts into and there are no pivots which could break after repeated use. However, the magnetic force between the enrgy to release electromagnet and the permanent magnet must be sufficient to hold the sliding barrier in closed position.

Refining the chosen design

Figure 4 shows an early version of my chosen design for the pipe barrier mechanism. The barrier is operated by a pair of electromagnets, one which becomes magnetic when it is energized and one which loses its magnetism when it is energized.

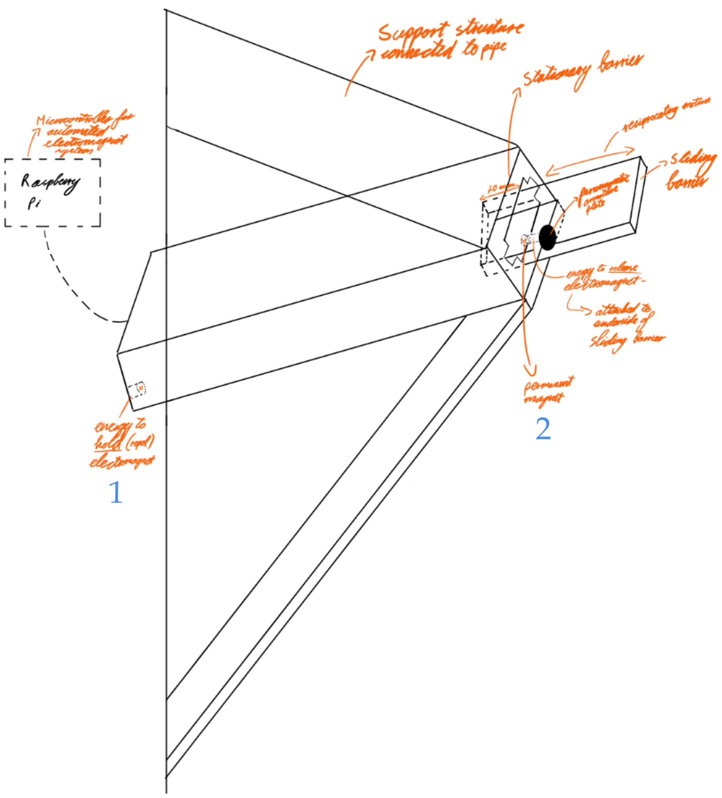


Figure 4, initial design of pipe barrier mechanism

The design shown in *figure 4*, contained one major issue: I misunderstood the internal design of the **energy to hold electromagnet**[31]. In my design, I assumed that energizing the energy to hold electromagnet (*figure 5*) would align the magnetic field of the soft[32] iron core (*1*) inside the electromagnet, such that the ferromagnetic core repelled the **permanent magnet** (*2*) attached to the sliding barrier. In turn, this would cause the sliding barrier to move into its extended state and put the barrier mechanism into the closed position. However, as the field of the permanent magnet is usually 'hundreds of times stronger than the [magnetizing effect of] the current in the coil' (Pasternak, 2015), the ferromagnetic core remained attracted to the permanent magnet attached to the sliding barrier regardless of the current in the electromagnetic coil. Therefore, using an electromagnet to repel a strong permanent magnet is impossible. The only possible solution was to use a coreless electromagnet (a solenoid)[33]. However, using the equation referenced in *footnote 33*, I found that a solenoid of a practical size for my application would be

[31] Details of and calculations relating to the design before this misconception was identified can be found in the document titled *'Activity Log - Visual'*

[32] Soft refers to the property of the iron core to become demagnetized once the external magnetic field from the current in the coil is removed.

[33] A coreless electromagnet has no ferromagnetic core, so its magnetic field comes only from the magnetic field created when current flows through the wire coil (solenoid) (Electronics Tutorials, 2018). The equation for the force of the electromagnet is $F = \frac{(N \cdot I)^2 \cdot \mu_0 \cdot A}{2g^2}$, where F = force of electromagnet, N = number of turns of wire, I = current through wire, u_0 = magnetic constant ($4\pi \times 10^{-7}$), A = cross-sectional area of solenoid and g = length of gap between solenoid and permanent magnet (Banas, 2018).

unable to generate a sufficient force to repel the permanent magnet attached to the sliding barrier.

Having spoken with an expert from *Magnet-Shultz*, I was advised to use a **linear solenoid actuator** instead of the energy to hold electromagnet to move the sliding barrier into the extended state. Two diagrams of the final design for the pipe barrier mechanism, which incorporates linear solenoid actuator, an energy to release electromagnet and a ferromagnetic armature, can be seen in *figure 6* and *figure 7* (further details on each element of the design will follow)[34].

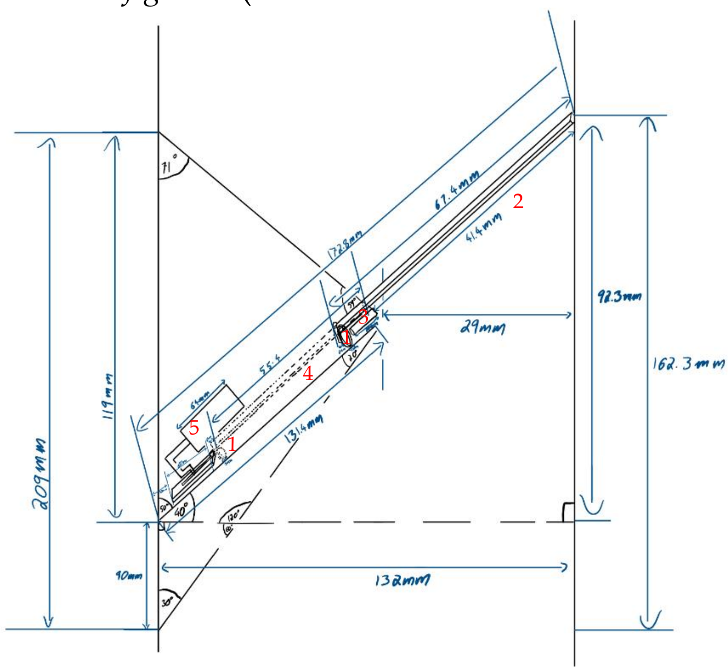


Figure 6, pipe barrier mechanism – final version (technical diagram)

Operation process

When pipe barrier mechanism is closed, the **ferromagnetic armature** (1) attached to the **sliding barrier** (2) is attracted to the **energy to release electromagnet** (3), holding the sliding barrier in its extended state.

To open the pipe barrier mechanism, the energy to release electromagnet[35] is energized and thus de-magnetized. As a result, the sliding barrier slides backwards under gravity into the open position.

To close the pipe barrier mechanism again, the **linear solenoid actuator** (*5*) is energized, pushing the sliding barrier into its extended state again with the actuator lever.

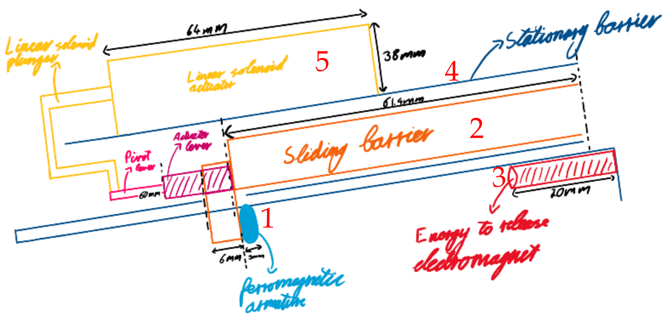
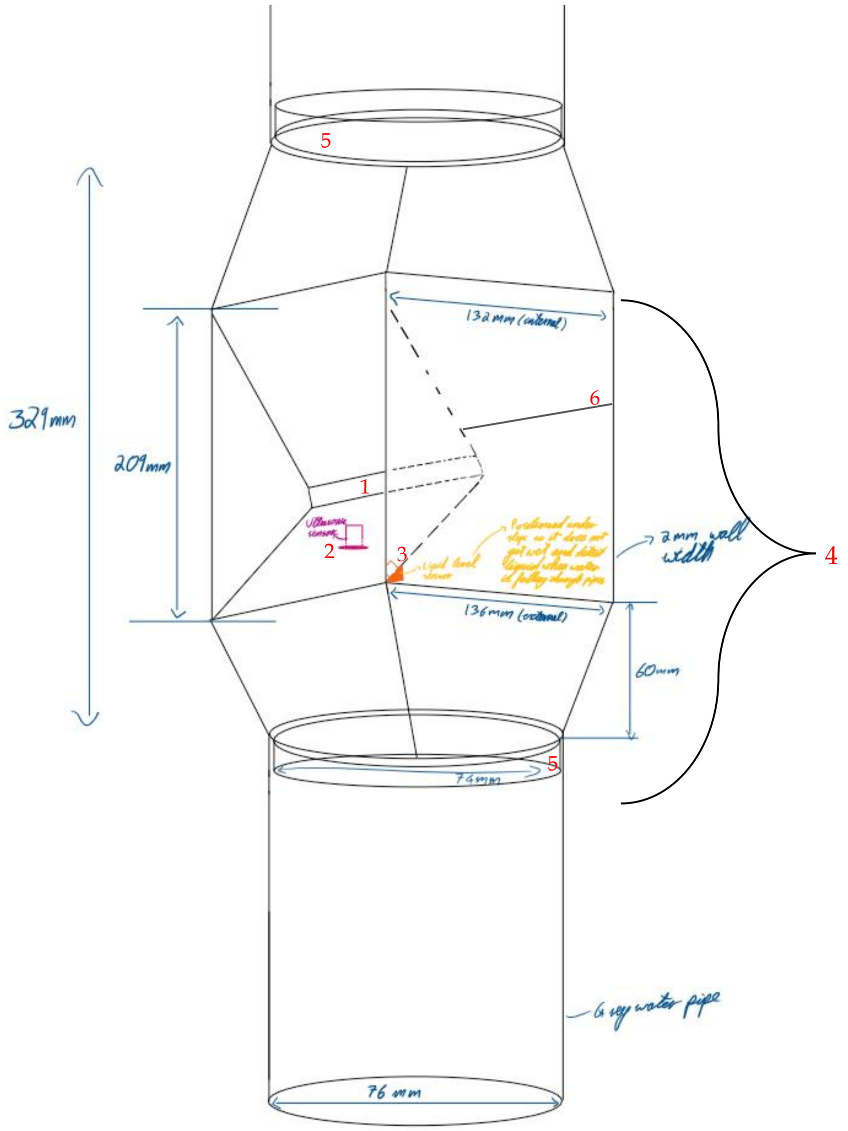


Figure 7, pipe barrier mechanism – final version

[34] The elevation of the pipe barrier mechanism (40°) was determined in Experiment 1 using a lead sash weight to simulate the mass of the water within each pipe section.
[35] The energy to release electromagnet is attached to the stationary barrier (*4*) rather than the sliding barrier, as otherwise it makes the sliding barrier too heavy for the ferromagnetic armature to move it into the extended state.

The integrated design of the pipe barrier mechanism into the grey wastewater pipe can be seen in *figure 7:*



Design details

The sliding barrier moves linearly through the slot labelled *1*.

The **ultrasonic distance module** (*2*) and the **liquid level sensor** (*3*) are both used in the redistribution of the grey wastewater within the pipe (details on both components will follow).

The grey water pipe used has an external diameter of 80mm and an internal diameter of 76mm – this is slightly wider than standard grey water pipes and is used to reduce the head loss due to friction between the water and the pipe.

The housing for the **pipe barrier mechanism** (*4*) is a cuboid as the sliding barrier must create a watertight seal with the edge of the housing, and thus the housing dimensions must be complementary to the sliding barrier.

Waterproof adhesive is used to join the barrier mechanism housing and the grey water pipe at (*5*).

Watertight sealant edge is used at (*6*) used to create seal between sliding barrier and the edge of the pipe barrier mechanism housing when the sliding barrier is in the extended state.

Stationary Barrier

Sketches and notes of each design can be seen in the document titled 'Activity Log - Visual'

The stationary barrier houses the **sliding barrier**, along with the **linear solenoid actuator** and the **energy to release electromagnet**. The sliding barrier must fit snugly into the stationary barrier, to ensure that it is sturdy and watertight when the barrier is in the closed mechanism. However, the sliding barrier must still be able to move freely within the stationary barrier, else the barrier mechanism could fail. *Figure 9* shows an initial design of the stationary barrier.

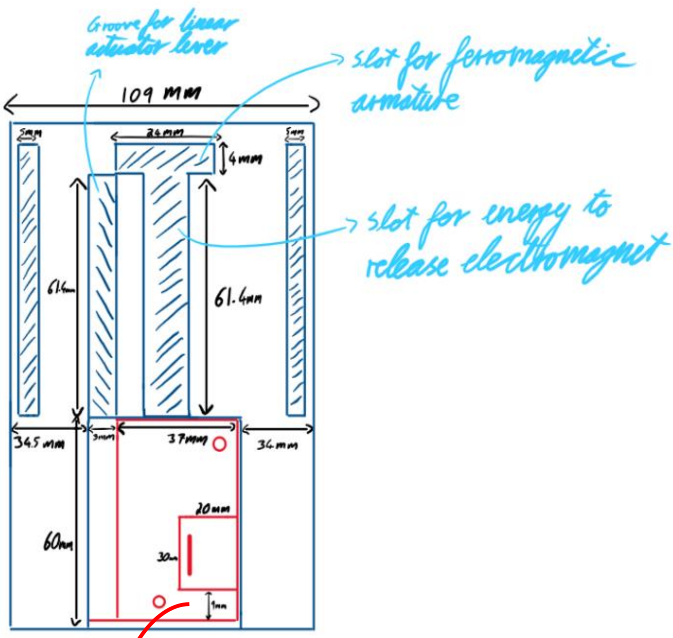


Figure 9, initial design of stationary barrier

The design shown in *figure 9* contains a **ferromagnetic armature** attached to the open end of the stationary barrier. This layout meant that the **energy to release electromagnet** had to be attached to the sliding barrier. However, in Experiment 15, it was found that the energy to release electromagnet made the sliding barrier too heavy for the **linear solenoid actuator** to push back into its extended state. In future designs, I switched the locations of the ferromagnetic armature and the energy to release electromagnet to solve this problem.

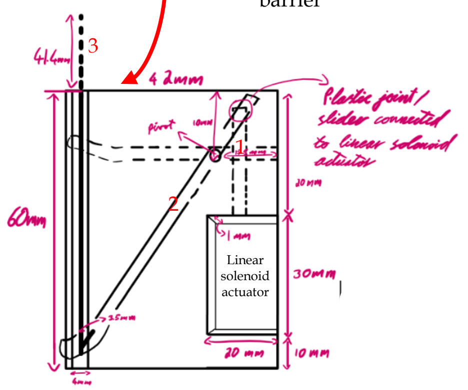


Figure 10, initial design of linear solenoid actuator mechanism

The design shown in *figure 10* shows the initial design for a mechanism to create rapid linear motion which returns the sliding barrier to the extended state. When the linear solenoid is energized, its **plunger** (*1*) is retracted by 9mm. The mechanism uses a **class 1 lever system** (the **pivot lever** is labelled as (2)) to translate this linear motion of the plunger into an extension of 41.4mm of the **linear actuator lever** (*3*).

This linear motion of the linear actuator lever drives the sliding barrier into the extended state. I used a pivot mechanism in conjunction with a short plunger stroke (rather than using a linear solenoid with a long plunger stroke alone), as this allows the design to be more compact since the linear solenoid actuator can be position alongside the actuator lever in the de-energized state[36].

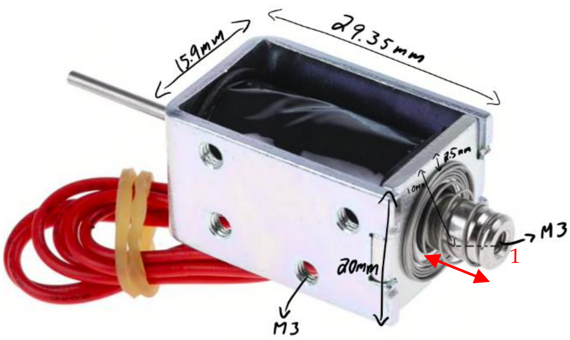


Figure 11, initial linear solenoid actuator used (RS Components, n.d.)

Figure 11 shows a 12V DC **linear solenoid actuator.** A plastic joint is attached to the female socket of the **plunger** (*1*), so that when the linear solenoid is energized and the plunger retracts, the **pivot lever** is rotated about the **pivot**, inducing linear motion in the **actuator lever**. However, in Experiment 14, I found that the force exerted by the stroke of the plunger of the linear solenoid in *figure 11* was too weak to move the sliding barrier to the extended state.

Sufficiently powerful linear solenoid actuators were also significantly larger than the one shown in *figure 11*. This meant that I had to rethink the design of the stationary barrier so that the linear solenoid actuator could still be incorporated without dramatically increasing the size of the stationary barrier. *Figure 12* shows the new design:

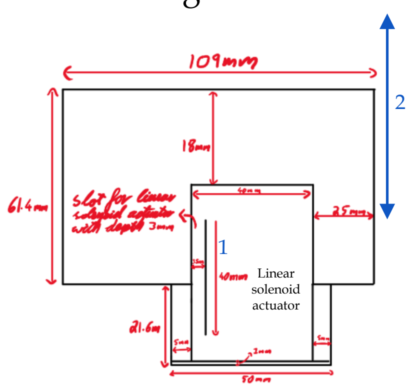


Figure 12, final design of stationary barrier cover

Figure 12 shows the **detachable cover**[37] of the stationary barrier mechanism. This new (and final) design locates the **linear solenoid actuator** in the cover of the stationary barrier, rather than being level with the lever mechanism. A **sleeve** attached to the **plunger** connects the plunger to the **pivot lever** (more details about this design can be found on *page 42*). The groove (*1*) slots into the linear solenoid actuator and retains it in the correct position. The arrow labelled (*2*) shows the motion of the **sliding barrier** induced by the stroke of the

[36] This would not be possible if a linear solenoid with a long plunger stroke was used, as the plunger would extend behind the linear solenoid actuator when it was in the de-energized state.
[37] The cover is detachable so that sleeve for the linear solenoid plunger can be connected to the pivot lever – this design feature is detailed further on *page 42*.

plunger. Note that, except for removing the slot for the linear solenoid actuator, the *base* of the **stationary barrier** remains as shown in *figure 10.*

Below are shown the final **CAD designs** of stationary barrier:

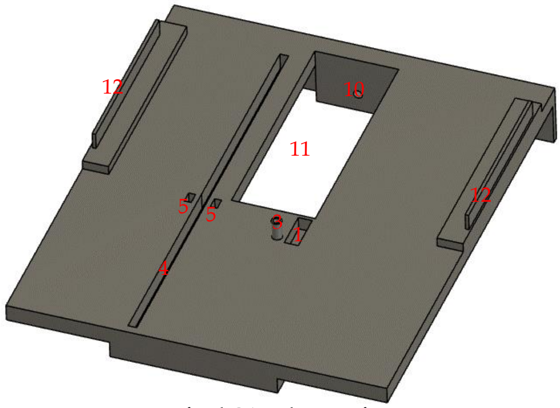


Figure 13, final CAD design of stationary barrier base

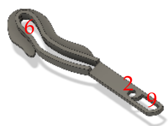


Figure 14, final CAD design of pivot lever

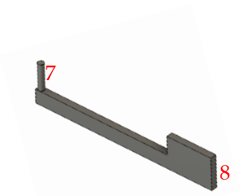


Figure 15, final CAD design of actuator lever

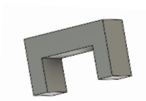


Figure 16, final CAD design of actuator lever cover

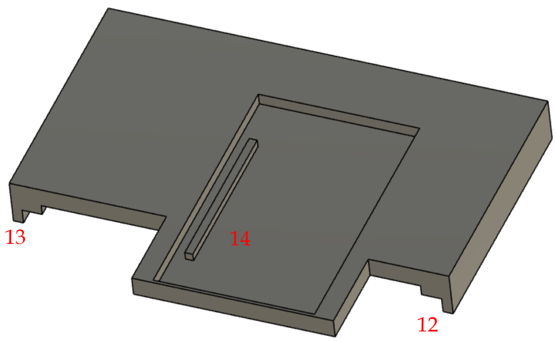


Figure 17, final CAD design of detachable cover of stationary barrier

Design details

The sleeve connected to the **plunger** slots into the **groove** (*1*), enabling the **pivot hole** (*2*) in the **pivot lever** (*figure 14*) to be rotated about the **pivot** (*3*) when the plunger retracts.

The **groove** (*4*) for the **actuator lever** (*figure 15*) ensures that the actuator lever's motion is linear and controlled. The actuator lever is also kept in place by the **actuator lever cover** (*figure 16*) which slots into the **grooves** (*5*).

The **slot** (*6*) in the **pivot lever** is connected to (*7*) on the **actuator lever**.

The **face** (*8*) on the **linear solenoid actuator** makes contact with and pushes the **sliding barrier**.

The sleeve attached to the **plunger** of the **linear solenoid actuator** connects to the **pivot lever** at (*9*).

The **energy to release electromagnet** attaches to the 2.5mm threaded hole at (*10*).

Figure 18, final CAD design of pivot cap

The **ferromagnetic armature** attached to the **sliding barrier** moves through the **slot** (*11*) when the sliding barrier is in motion.

The **detachable cover** (*figure 17*) is attached to the **base** of the barrier mechanism at (*12*) and (*13*) using plastic glue.

The **linear solenoid actuator** slots into the **detachable cover** at (*14*).

The **pivot cap** (*figure 18*) covers the **pivot** (*3*) to prevent the **pivot lever** (*figure 14*) from becoming detached during motion.

Linear Solenoid Actuator

Sketches and notes of each design can be seen in the document titled 'Activity Log - Visual'

The **linear solenoid actuator** (*figure 19*) serves to rapidly induce linear motion with a force large enough to move the sliding barrier into the extended state. As detailed on *page 34*, a **lever system** is used to reduce the space required by the linear solenoid actuator mechanism.

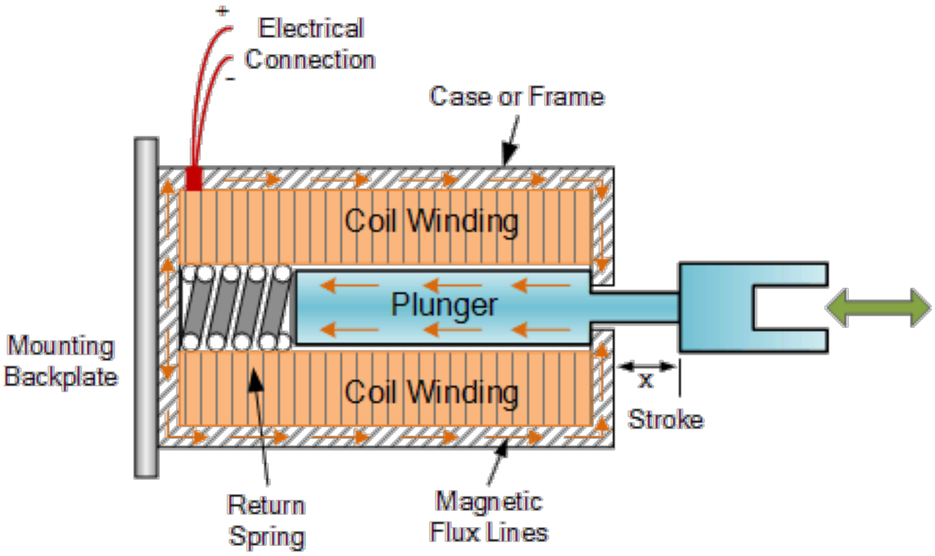


Figure 19, linear solenoid actuator (Electronics Tutorials, 2018)

A linear solenoid actuator comprises of a **solenoid** (a **coil** of wire) and a **plunger**. When a current passes through the solenoid, an **electromagnetic field** is induced and the plunger is attracted towards this electromagnetic field (Electronics Tutorials, 2018). This rapid linear motion is called the **stroke**[38] of the plunger. The force of the plunger's motion increases as the stroke distance decreases, since the magnetic flux[39] is greater towards the center of the solenoid. This relationship can be seen in *figure 20,* which is a standard graph[40] of Force (N) against stroke (mm) at 20°C.

[38] The stroke refers to the extension of the plunger from the body of the linear solenoid actuator.
[39] Magnetic flux is a measure of the total magnetic field in a given area. So, a higher magnetic flux indicates a higher magnetic field density (Supermagnete, n.d.).
[40] I have shown a standard graph rather than the graph specific for the linear solenoid actuator I used as the linear solenoid actuator I bought did not come with a data sheet.

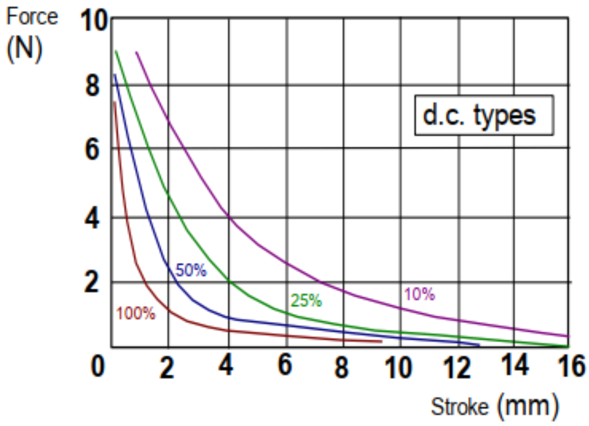


Figure 20, graph of Force (N) against Stroke (mm) for a linear solenoid actuator (Barleyman, 2019)

Note that the stroke force varies depending on the duty cycle[41]. At higher duty cycles, the linear solenoid is energized for more time and therefore the solenoid overheats. Therefore, resistance in the coil increases[42] and so the current drawn by the solenoid decreases in accordance with the *power* equation:

$$P = I^2R$$

The magnetic field generated by the linear solenoid is dependent on the current through the wire, since the equation for magnetic field is:

$$B = \mu_0 \left(\frac{N \cdot I}{l} \right)$$

- B = magnetic field
- u_0 = magnetic constant
- N = number of turns (of solenoid)
- I = **current through solenoid**
- l = length of solenoid

Therefore, if the magnetic field of the solenoid decreases, the force with which the plunger is attracted to this magnetic field will decrease. The opposite is the case for decreasing the duty cycle, since the solenoid has time to cool down in between it being energized (Pasternak, 2021).

41 Duty cycle is the ratio of time a device is turned ON compared to the time the device is tuned OFF (Fluke, 2021).

42 Resistance is inversely proportional to current ($R = V/I$)

Plunger force calculations

To determine the plunger force required by the linear solenoid actuator to move the sliding barrier into the extended state, I performed several dynamics calculations:

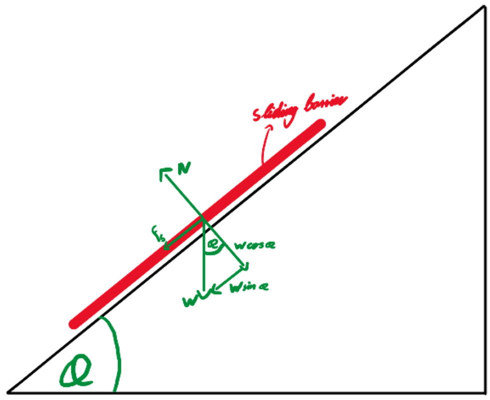


Figure 21, diagram of forces for calculating the coefficient of static friction

1. Determine **equation** for **coefficient of static friction** (u_s)[43]:

$$f_s = \mu_s \cdot N$$
$$f_s = W \cdot \sin\theta$$
$$N = W \cdot \cos\theta$$
$$W \cdot \sin\theta = \mu_s \cdot W \cdot \cos\theta$$
$$\mu_s = \tan\theta$$

2. Substitute results from Experiment 3 into above equation to find **coefficient of static friction** (u_s):

$$\mu_s = (\tan 22.3^\circ)$$
$$\mu_s = 0.41$$

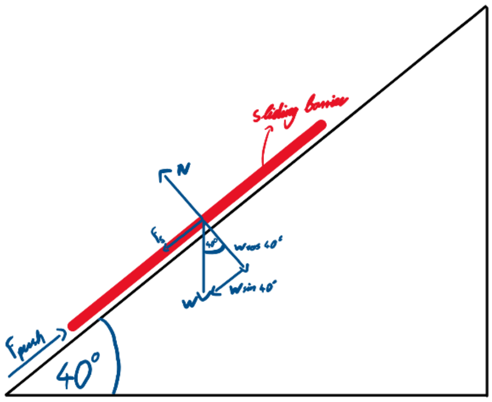


Figure 22, diagram of forces for calculating the static friction force and required force of plunger

3. Determine **static friction force** (f_s)[44]:

$$f_s = \mu_s \cdot N$$
$$f_s = 0.41 \cdot W \cos 40^\circ$$
$$f_s = 0.41 \cdot [mg \cdot \cos 40^\circ]$$
$$f_s = 0.41 \cdot [(0.044 \cdot 9.81) \cos 40^\circ]$$
$$f_s = 0.136\ N$$

[43] The coefficient of static friction (u_s) is the ratio of the maximum static friction force (f_s) to the normal force without achieving motion (Bijoy Kumar Behera, Hari and Woodhead Publishing, 2012). Note also that the angle θ is the minimum angle at which the sliding barrier slides back into the stationary barrier under gravity (shown in *figure 21*) and is calculated in Experiment 3.

[44] The static friction force (f_s) is the minimum force that must be applied to the sliding barrier to achieve motion (Holzner, n.d.). Refer to *figure 22* for a visual representation of the static friction force. Note also that m is the combined mass of the sliding barrier (29g) and the attached ferromagnetic armature (15g)

4. Determine required **force of plunger** (F_{push})[45]:

$$F_{push} = (W \cdot \sin 40^\circ) + f_s$$
$$F_{push} = (m \cdot g \cdot \sin 40^\circ) + f_s$$
$$F_{push} = (0.044 \cdot 9.81 \cdot \sin 40^\circ) + 0.136$$
$$F_{push} = 0.413\ N$$

The plunger of the **linear solenoid actuator** that I used initially (shown in *figure 11*) was unable to provide the required 0.413 N of force to move the sliding barrier into the extended state. Using the Force/Stroke graph shown in *figure 23* for the initial linear solenoid actuator I used, I drew the following conclusion as to why the plunger of this linear solenoid actuator was unable to provide sufficient force:

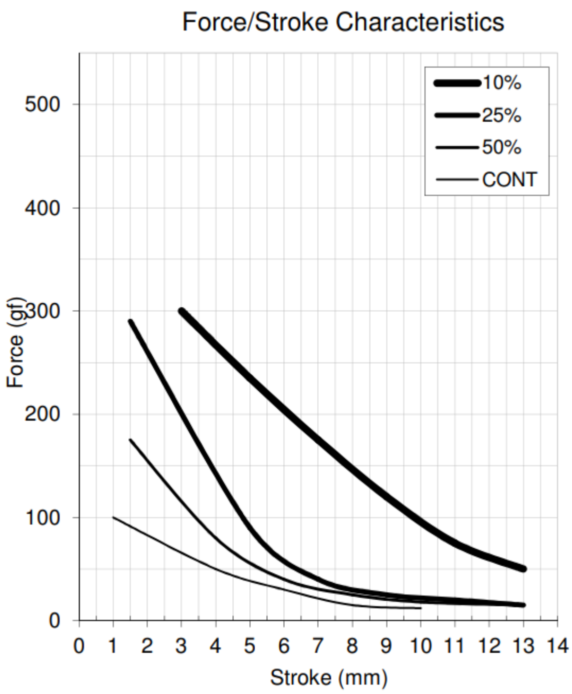


Figure 23, Force/Stroke diagram for the initial linear solenoid actuator used (RS Components, n.d.)

The units of force used in *figure 23* is *gram-force* (*gf*), which is equal to the force exerted by one gram of mass (Translators Cafe, n.d.). Therefore, on Earth, 1*gf* = 0.00981 N. Using this conversion and *figure 23*, it can be concluded that the force of the **plunger** where the stroke is in range 13mm-6mm is around 0.147-0.402 N[46]. As determined in Experiment 6, a stroke distance of 9mm is required to extend the **linear actuator lever** by 41.4mm and move the **sliding barrier** into the extended state. However, as the force of the plunger during the first 7mm of the stroke is less than the required 0.413 N[47] of force to move the sliding barrier into the extended state, the initial linear solenoid actuator I used was not powerful enough.

---

[45] Refer to *figure 22* for a visual representation of the forces concerned when calculating the total required force of the plunger (the sum of the static friction force and the component of the weight of the stationary barrier and attached ferromagnetic armature acting down the slope).
[46] Note that these values of plunger force assume a duty cycle of greater than 10%.
[47] The maximum force is c. 0.402 N at 6mm stroke.

Given this, I decided to incorporate a larger and more powerful **linear solenoid actuator** into the barrier mechanism, as further explained on *page 35.* An image and a technical diagram of this new **linear solenoid actuator** are shown in *figure 24* and *figure 25* respectively.

Figure 24, final linear solenoid actuator used (Amazon, n.d.)

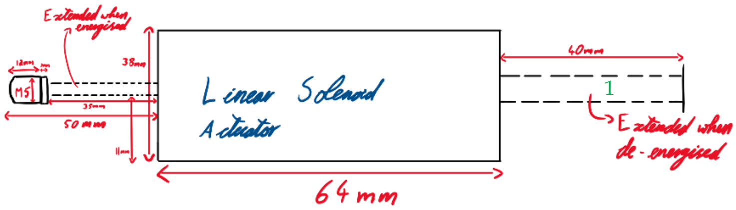


Figure 25, technical diagram of linear solenoid actuator shown in *figure 24*

The 12V DC **linear solenoid actuator** shown in *figure 24* is draws 8A current, which is significantly more than the initial linear solenoid actuator used. This explains why the plunger force of the new linear solenoid actuator is much greater; as explained on *page 39*, the plunger force has a positive correlation with the strength of the magnetic field created by the solenoid, which increases as current flow through the solenoid increases.

As shown in *figure 25*, the **stroke** of the new linear solenoid actuator's **plunger** is 35mm[48], which is significantly longer than the 9mm stroke required in my application. Therefore, I designed a **sleeve** which fitted tightly onto the plunger and capped the stroke at 26mm[49] (i.e. when the linear solenoid actuator is energized, the plunger can only retract by the required 9mm). A technical diagram of this sleeve can be seen in *figure 26*.

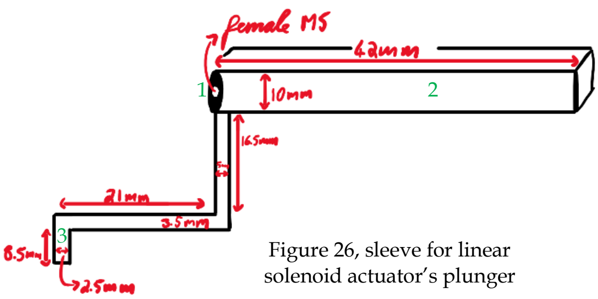


Figure 26, sleeve for linear solenoid actuator's plunger

The **channel** (labelled *1* in *figure 26*) extends throughout the **sleeve** (2) and houses the **plunger** (labelled *1* in *figure 24*). The bolt on the end of the plunger is male M5 and the complementary channel is female M5 to ensure that the connection is tight, and the sleeve remains in place when the linear

[48] Note that the maximum extension of the plunger is 50mm as the bolt and nut on the head of the plunger have a length of 15mm (as shown in *figure 25*), but the bolt and the nut do not retract into the linear solenoid actuator when it is energised, so they are not included in the measurement of stroke.

solenoid actuator is energized and the plunger is retracted rapidly. The sleeve connects to the **pivot lever** (shown in *figure 9*) at the point labelled (*3*) in *figure 26*.

When the linear solenoid actuator is energized and the pivot retracts by 9mm, the **rear plunger** (labelled *1* in *figure 25*) extends by 17mm. To ensure that this extension does not protrude outside of the barrier mechanism, the linear solenoid actuator is located 18mm away from the edge of the barrier mechanism from which the slide barrier emerges (this 18mm offset is shown in *figure 12*).

Below is shown the final **CAD design** of the sleeve for the plunger:

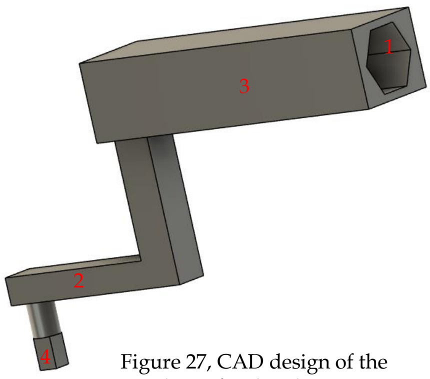


Figure 27, CAD design of the sleeve for the plunger

Design details

The **M5 channel** (*1*) tightly houses the plunger to cap the stroke to 9mm.

The shaft labelled (2) slides under the **detachable cover of the stationary barrier** (shown in *figure 17*) so that the **sleeve** (*3*) can connect to the **pivot lever** (*figure 9*) at the point labelled (*4*).

Figure 28, plunger sleeve connected to linear solenoid actuator and stationary barrier

Figure 28 shows how the **linear solenoid actuator** (*1*) and the **plunger sleeve** (*2*) fit onto the stationary barrier (*3*). From this image, it is clear why the **stationary barrier cover** must be detachable: the point labelled (*4*) which connects the **plunger sleeve** to the **pivot lever** (*5*) cannot be hooked into the pivot lever when the stationary barrier cover is in place. Therefore, without making the stationary barrier cover detachable, it would not be possible to assemble the barrier mechanism.

Energy to Release Electromagnet

Further details can be seen in the document titled 'Activity Log - Visual'

An **energy to release electromagnet** (otherwise known as a **permanent electromagnet**) behaves oppositely to traditional electromagnets: when it is de-energized, it has a magnetic field, but when it is energized, this magnetic field is neutralized (Magma, n.d.).

When the barrier mechanism is in the closed position, the **sliding barrier** must be held in position such that it extends between the mouth of the **stationary barrier** and the **barrier mechanism housing** as shown in *figure 29*:

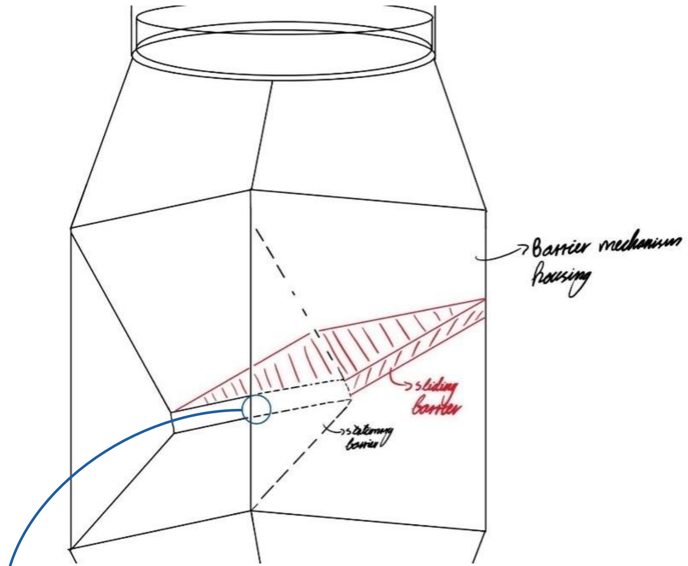


Figure 29, stationary and sliding barrier inside barrier mechanism housing

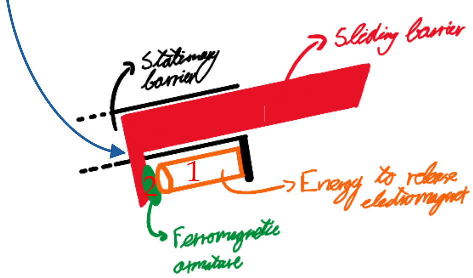


Figure 30, sliding barrier when the barrier mechanism is in closed position

As the **electromagnet** (labelled (*1*) in *figure 30*) serves to maintain magnetic attraction between itself and the **ferromagnetic armature** (labelled (2) in *figure 30*) for significantly longer periods than it should be de-magnetized[50], it is appropriate to use an **energy to release electromagnet** here. Energy to release electromagnets comprise of a strong permanent magnet surrounded by a solenoid. When the electromagnet is energized and current flows through the solenoid, a strong electromagnetic field is created which is in opposition to magnetic field of the permanent magnet. Therefore, the magnetic field of the

[50] The electromagnet should become de-magnetised when the barrier mechanism is to be opened and therefore the sliding barrier must no longer be held in the extended state shown in *figure 29*, and should instead slide back into the stationary barrier.

permanent magnet is **neutralized** and the electromagnet briefly becomes de-magnetized[51] (Upton, 2020). This removes the magnetic attraction between the **ferromagnetic armature** and the electromagnet and therefore allows the sliding barrier to slide back into the stationary barrier, opening the barrier mechanism.

As determined in Experiment 2, the holding force required to maintain the sliding barrier in the extended state (and therefore the barrier mechanism in the closed position) is 0.965kg. Therefore, I chose to use an **energy to release electromagnet** with a holding force of 2kg[52]. It was important to determine the minimum holding force required, as using an energy to release electromagnet with a surplus holding force in the de-energized state could jeopardize the functioning of the barrier mechanism: as the holding force in the de-energized state increases, the residual holding force in the energized state also increases (this relationship is shown in *figure 31*). Therefore, if the holding force in the energized state is too high, the energy to release electromagnet may remain attracted to the ferromagnetic armature and the sliding barrier may not be able to slide back into the stationary barrier, preventing the barrier mechanism from opening. Moreover, to counteract the residual holding force, the **linear solenoid actuator** (*page 38*) is also energized when the energy to release electromagnet is energized; the slight jerk created by the rapid retraction of the linear solenoid actuator's plunger jolts the sliding barrier sufficiently to overcome the residual magnetic attraction between the electromagnet and ferromagnetic armature.

Part No.	Dimensions		Holding Force		Threaded Hole Mountings	
	Diameter	Depth	Power Off (at 0mm)	Power On (at Working Voltage)	Diameter	Max Depth
XH1720	17mm	20mm	2kg	< 50g	M3	6mm
XH3530	35mm	30mm	10kg	< 500g	M4 x 2	6mm
XH4030	40mm	30mm	16kg	< 500g	M4 x 2	6mm
XH4942	49mm	42mm	40kg	< 2kg	M8	10mm
XH6045	60mm	45mm	60kg	< 5kg	M5 x 4	10mm
XH8050	80mm	50mm	80kg	< 4kg	M8 x 4	12mm
XH10060	100mm	60mm	150kg	< 5kg	M8 x 4	15mm

Part No.	Wire Length	Positive '+'	Negative '-'	Duty Cycle	Max Power ON time	Working Voltage	Rated Power
XH1720	200mm	Red	Black	10%	< 2 Seconds	6VDC	3.4W
XH3530	200mm	Red	Black	10%	< 2 Seconds	24VDC	11W
XH4030	200mm	Red	Black	10%	< 5 Seconds	24VDC	18W
XH4942	300mm	Red	Black	10%	< 10 Seconds	24VDC	20W
XH6045	200mm	Red	Black	10%	< 2 Seconds	24VDC	72W
XH8050		Red	Blue	5%	< 5 Seconds	24VDC	40W
XH10060		Red	Blue	5%	< 5 Seconds	24VDC	55W

Figure 31, datasheet for a range of energy to release electromagnets – the highlighted rows refer to the energy to release electromagnet I used (Bunting, n.d.).

Figure 32, energy to release electromagnet used (6V, 0.57A, 3.4W) (Bunting, n.d.)

[51] The qualifier 'briefly' is used since the electromagnet cannot remain de-magnetized for an extended period. As the current passes through the solenoid, it causes the wire to heat up, increasing resistance in the solenoid. Therefore, as resistance increases, the current drawn by the solenoid decreases (current is inversely proportional to resistance). The equation for the electromagnetic field strength of a solenoid is: $B = \mu_0 \cdot n \cdot I$, where B = magnetic field strength, u_0 = magnetic constant ($4\pi \times 10^{-7}$), n = coil turn density (number of wire turns/length of solenoid) and I = current through solenoid (Nave, 2019). Therefore, as the current drawn by the solenoid decreases, the electromagnetic field of the solenoid will decrease until it is too low to neutralize the magnetic field of the permanent magnet inside the solenoid.

[52] Note that in an actual application of this hydroelectric generator system, the electromagnet would have to be water-resistant, as greywater may enter the stationary barrier.

Sliding Barrier

Sketches and notes of each design can be seen in the document titled 'Activity Log - Visual'

The sliding barrier allows the **barrier mechanism** to move from the open position to the closed position. When the sliding barrier is retracted inside the stationary barrier, the barrier mechanism is in the *closed state*, and when the sliding barrier is extended out of the stationary barrier (by 41.4mm), the barrier mechanism is in the *open state*. The means of inducing this extension and retraction motion of the sliding barrier have been discussed earlier in the report.

Initial designs for the sliding barrier had the **energy to release electromagnet** attached to its underside, as shown in *figure 33*. However, in Experiment 15, I found that the mass of the electromagnet made the sliding barrier too heavy for the **linear solenoid actuator mechanism** to return the sliding barrier to its extended position. To solve this problem, the **ferromagnetic armature**[53] (shown in *figure 34*) was attached to the sliding barrier in place of the energy to release electromagnet[54], as shown in *figure 35*. Moreover, I chose to use a low infill density (25%) for the sliding barrier to further reduce its mass to 29g[55].

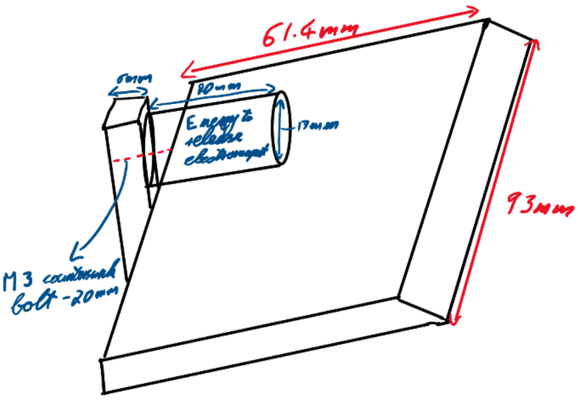


Figure 33, initial design of sliding barrier

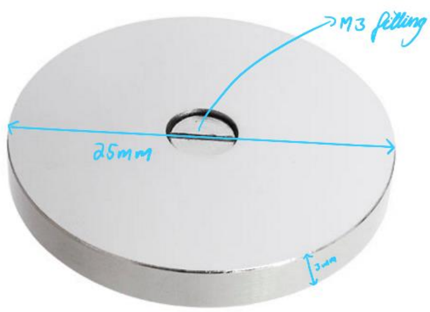


Figure 34, ferromagnetic armature (Eclipse magnets, n.d.)

[53] The ferromagnetic armature is a neodymium permanent magnet (Eclipse magnets, n.d.).
[54] As detailed on *page 34*, the energy to release electromagnet was attached to the stationary barrier.
[55] With the added mass of the attached ferromagnetic armature, the total mass of the sliding barrier is 44g.

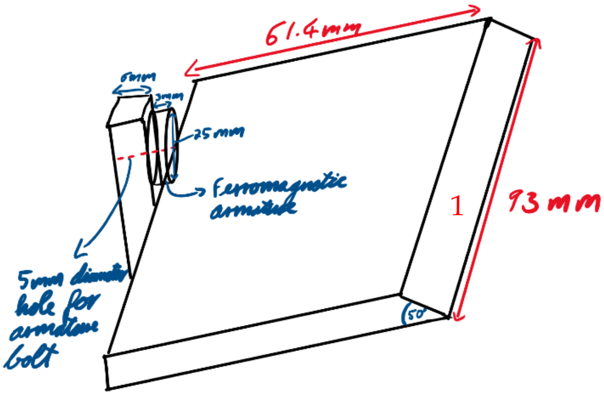


Figure 35, final design of sliding barrier

In Experiment 1, I found that 40° is the optimal angle for the **sliding barrier** to form with the **barrier mechanism housing**. Therefore, the edge of the sliding barrier (*1*) which makes contact with the barrier mechanism housing has a 40° chamfer to ensure the seal is watertight[56]. The sliding barrier has length 61.4mm, as it must extend 41.4mm out of the stationary barrier *and* pass over the energy to release electromagnet of length 20mm (these dimensions are visualized in *figure 6*).

Figure 36 shows the final **CAD design** of the sliding barrier:

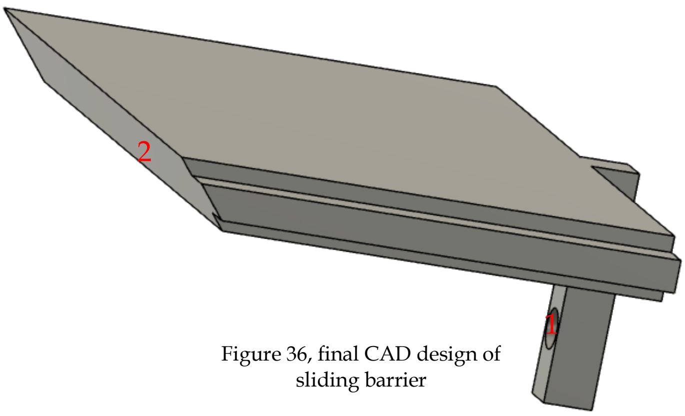


Figure 36, final CAD design of sliding barrier

Design details

The **ferromagnetic armature** is attached at (*1*).

The edge (*2*) makes contact with the edge of the **barrier mechanism housing.**

[56] As mentioned on *page 33*, a watertight sealant edge is also used above this contact point to reinforce this seal.

Liquid Level Sensor

Further details can be seen in the document titled 'Activity Log - Visual'

The **liquid level sensor** (shown in *figure 37*) is one of two types of sensors located at each barrier mechanism. Together with the **ultrasonic distance sensor** (detailed on *page* 52), it supports the intelligent hydroelectric system which redistributes water within the pipe to maximize electricity output from the turbine generator. The sensor detects when the pipe section below reaches full capacity of grey wastewater[57] and, using the Raspberry Pi microcontroller which is installed in each barrier mechanism, this information is fed-back to the server-side program to begin the water redistribution process.

Figure 37, liquid level sensor component (CQ Robot, 2021).

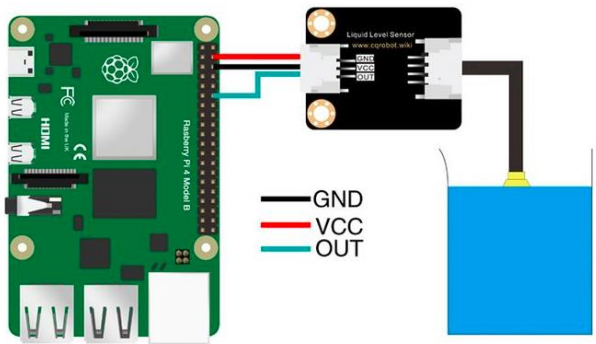


Figure 38, liquid level sensor connections with Raspberry Pi (CQ Robot, 2021).

As shown in *figure* 38, the liquid level sensor's control board has three connections with the Raspberry Pi[58]:

1. **GND** – ground connection
2. **V**$_{CC}$ – voltage supply to device with reference to the ground (5V DC)
3. **OUT** – the output channel is set to *high* (3.3V) when liquid is present and *low* (<0.1V) when no liquid is present

[57] The installation location of the ultrasonic distance sensor can be seen in *figure 8.* Note that it is positioned underneath the slope of the barrier mechanism so that doesn't get wet and falsely indicate that the pipe section below is at full capacity when water is falling through the greywater pipe.
[58] The circuit schematic for the liquid level sensor can be seen on *page 55.*

The theory

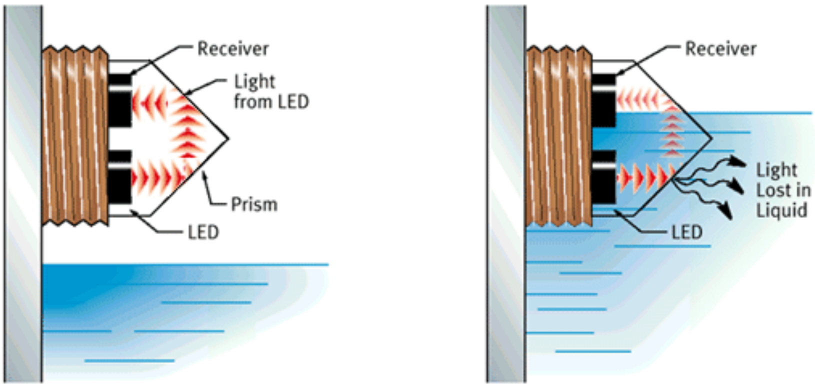


Figure 39, operation of liquid level sensor (SST Sensing, 2018)

No liquid present:

1. The **LED** (*1*) emits **infrared light**[59]
2. The infrared light undergoes **total internal reflection** at (*2*) as the plastic prism has a greater **refractive index** than air[60]
3. The infrared light again undergoes total internal reflection at (*3*) for the same reasons as above
4. The presence of infrared light is detected by the **infrared phototransistor** at (*4*)
5. The **output** from the liquid level sensor is set to *low*

Liquid present:

1. The **LED** (*5*) emits **infrared light**
2. Most of the infrared light does not undergo **total internal reflection** at (*6*) as the water now has a greater **refractive index** than the plastic prism[61]. Therefore, most of the light refracts across the plastic prism and is passes into the water.
3. More infrared light passes out of the plastic prism and into the water at (*7*)
4. The lack of presence of infrared light is detected by the **infrared phototransistor** at (*4*)
5. The **output** from the liquid level sensor is set to *high*

---

[59] Infrared light is used as humans cannot see it, so it does not interfere with any system design.
[60] Total internal reflection occurs when the infrared light is travelling across a boundary from a denser to a less dense medium with an angle of incidence greater than the critical angle (BBC Bitesize, 2019). As plastic has a greater refractive index than air (1.0003), the first condition is satisfied. To satisfy the second condition, the shape of the prism is such that the angle of incidence of the infrared light from the LED is greater than the critical angle.
[61] The first condition for total internal reflection to occur, as explained in *footnote 60,* is no longer satisfied, as the refractive index of water (1.333) is greater than the refractive index of plastic, so the infrared light is travelling from a less dense to a more dense medium.

Phototransistor

An **infrared phototransistor** detects the presence of infrared light, as it only conducts current when infrared photos are present. The actual design of a phototransistor is shown in *figure 40*; however, for the ease of explanation, I will consider the infrared phototransistor as the combination of a **photodiode** and a **transistor**, as shown in *figure 41*[62].

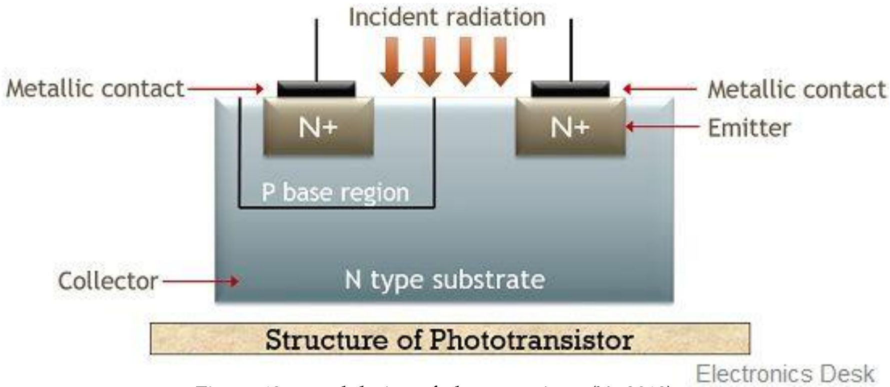


Figure 40, actual design of phototransistor (Y., 2019)

Photodiode

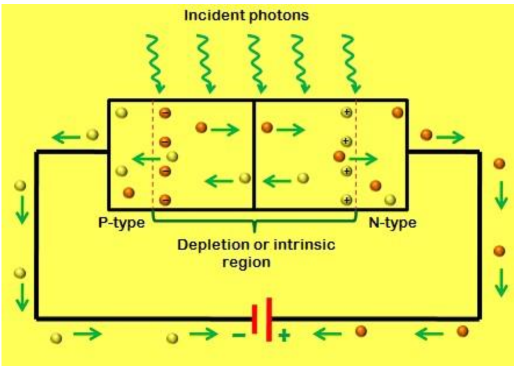


Figure 41, photo-diode circuit schematic (Ravi, 2018)

As shown in *figure* 41, the **p-n junction** created by the **positively doped** (**p-type**) and **negatively doped** (**n-type**) regions contains a **depletion region** which creates an **electric field**. As such, a **diode** is created so no charge can flow around the circuit in its natural state. However, if an **incident infrared (IR) photon** with sufficient energy strikes an atom in the depletion region, an **electron-hole pair** is created because of the **photo-electric effect**. The electric field will move the free electron towards the positive terminal and the free hole towards the negative terminal, creating a **photocurrent**.

[62] For more details on the working of a phototransistor, refer to the document title '*Activity Log – Visual*'

Transistor

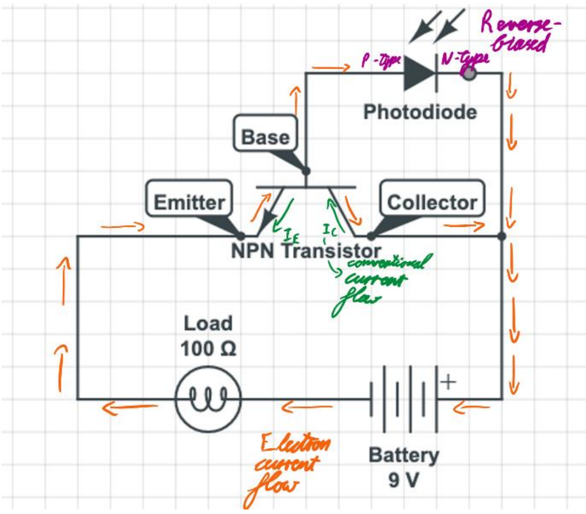


Figure 42, NPN transistor and photodiode schematic

The **photocurrent** generated by the photodiode shown in *figure 41* is very small. Therefore, an **NPN transistor** is used to amplify this current, as shown in *figure 42*. When no **IR photons** are present, no current can flow through the photodiode. Therefore, there is no **base current** and so the **electric field** created by the depletion zone between the **n-type emitter** and the **p-type base** of the NPN transistor prevents current flow between the **emitter** and the **collector**. However, when **IR photons** are present and the **photodiode** therefore allows current flow, a **base current** can now flow. As the base current flows, **positive ions** in the **emitter-base depletion region** are **neutralized** by electrons flowing from the emitter through the base and photodiode and **negative ions** are **neutralized** by holes flowing through the photodiode and base. As a result, the width of the depletion region at the **emitter-base junction** decreases and so the electric field opposing current flow disappears, allowing current to flow between the emitter and collector (Electronics Tutorials, 2018).

Ultrasonic Distance Sensor

Further details can be seen in the document titled 'Activity Log - Visual'

The **ultrasonic distance** (shown in *figure 43*) is the second of two types of sensors located at each barrier mechanism[63]. Along with the **liquid level sensor** (*detailed on page 48*), it supports the intelligent hydroelectric system which redistributes water within the pipe to maximize electricity output from the turbine generator. When the liquid level sensor detects that any one pipe section is at maximum capacity, the server-side program requests readings from all the ultrasonic distance sensors in the pipe. Each reading gives the distance to the water level in the pipe section below that barrier mechanism. Therefore, the server-side program uses this data to determine which barrier mechanisms should be opened to redistribute the water in the pipe, such that electricity output is maximized and any inlet of grey water into the apartment buildings is prevented.

The theory

Figure 43, ultrasonic distance sensor (Amazon, n.d.)

- Uses ultrasound waves, so humans cannot hear it[64]
- When the sensor is triggered, it emits an ultrasonic pulse and records the time for it to bounce back (*high level time*)
- Distance formula for ultrasonic sensor (Teach Engineering, 2021):

$$\text{Distance} = \frac{\text{High level time}}{\text{Velocity of sound in air}} \cdot 0.5$$

- Minimum distance[65] that can be measured is 25cm, so server-side computer program must adapt to erroneous results when distance to water level is less than 25cm (Amazon, n.d.)
- Transducer[66] is waterproof (Matt, 2016), so it can withstand exposure to the grey wastewater within the pipe.

[63] The installation location of the ultrasonic distance sensor can be seen in *figure 8.*
[64] Ultrasound waves have a frequency above the range of the human hearing (20-20,000Hz) (BBC Bitesize, n.d.).
[65] Maximum distance is 450cm, which is greater than the height of each pipe section (312cm) (Matt, 2016).
[66] The transducer is transmits and receives the ultrasound wave

As shown in *figure* 44, the ultrasonic distance sensor's control board has three connections with the Raspberry Pi[67] (Matt, 2016):

1. **5V** – voltage supply
2. **Trigger** - input pin which triggers sending of ultrasonic pulse
3. **Echo** – output channel is set to *high* (5V)[68] for the time it takes for the ultrasonic pulse to return to the transducer
4. **Ground** – ground connection

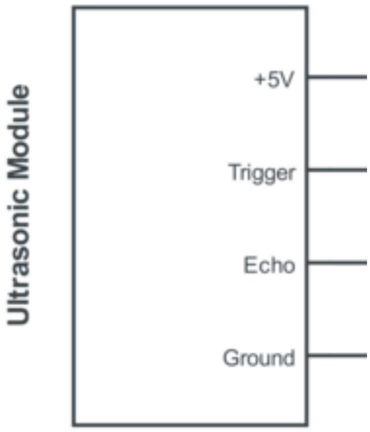


Figure 44, ultrasonic distance sensor connections with Raspberry Pi (Matt, 2016)

The theory

The **ultrasonic transducer** contains **piezoelectric crystals**. When the ultrasonic distance sensor is triggered and an electric current is applied to these crystals, they vibrate and undergo the piezoelectric effect, creating **ultrasound waves** that move outwards. When these ultrasound waves encounter a surface (in this case, the water in the pipe section below), they are reflected. The piezoelectric crystals convert these reflected ultrasound waves back into electric signals, which are processed by the sensor's control board (Absolute Medical Services, 2018).

[67] The circuit schematic for the liquid level sensor can be seen on *page 55*.
[68] As the output voltage of 5V is greater than the maximum 3.3V which the Raspberry Pi's GPIO input pins, a voltage divider is used to ensure the Raspberry Pi only receives 3.3V (circuit shown on *page 55*).

Piezoelectric effect

When piezoelectric crystals are 'hit' by the ultrasound waves and compressed slightly, they gain a charge, as the silicon and oxygen atoms in the silicon dioxide lattice of the crystals move slightly. As the oxygen is now slightly negatively charged and the silicon is slightly positively charged, an overall charged is induced in the crystal. This creates a voltage if the crystals are connected to a circuit (Mould, 2019).

When a voltage is applied to piezoelectric crystals, the inverse piezoelectric effect occurs and the crystals expand, creating ultrasound waves (Yang, 2016).

The piezoelectric effect is demonstrated in *figure 45* below:

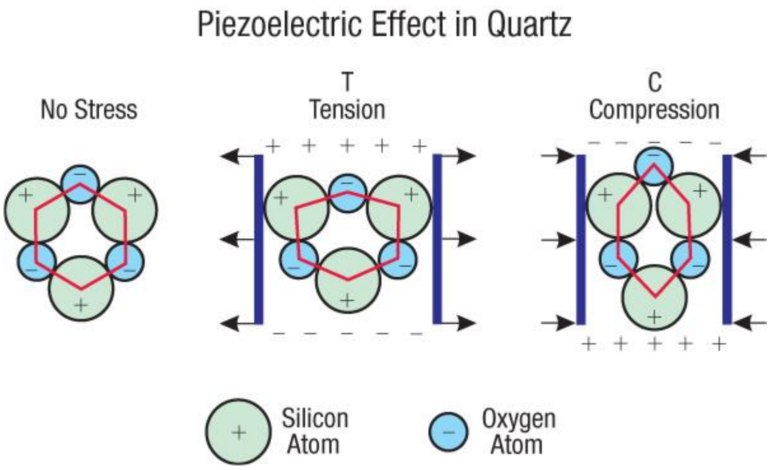


Figure 45, piezoelectric effect (Fleischer, 2018)

Circuit Schematics

Further details can be seen in the document titled 'Activity Log - Visual'

To operate the actuators (the **linear solenoid** and **energy to release electromagnet**) and receive data from the sensors (the **liquid level sensor** and **ultrasonic distance sensor**), I created a circuit which connected them to the power supply and the Raspberry Pi. Initially, I designed this circuit using *circuitlab* (shown in *figure 46*), so that I could visualize it and ensure that it worked as intended before physically constructing it. As such, I was able to verify if my calculations and design was correct as I could see what the current and voltage values were at different points in the circuit.

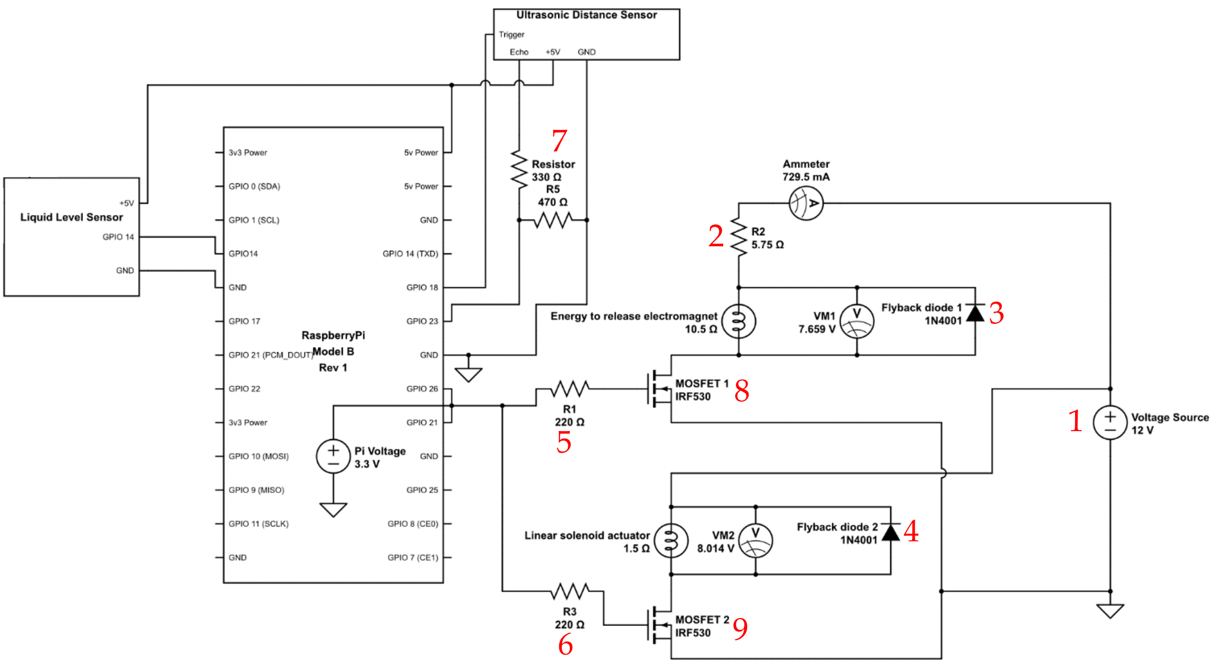


Figure 46, final circuit schematic

Circuit design theory

The Raspberry Pi is connected to each component via a **GPIO pin**, which can be sent to input or output depending on whether it is receiving data from or sending data to the component.

A 12V **power supply** (*1*) is used as the linear solenoid actuator requires 12V. Initially, I faced some issues with the operation of the linear solenoid, as my original power supply had a maximum current rate of 0.28A, yet the linear solenoid required 8A. Therefore, as the power supply was unable to supply 8A, the linear solenoid appeared to be broken. Using a 10A power supply solved this.

A **5.75 ohm resistor** (*2*) is connected in series with the energy to release electromagnet, to ensure that the 6V rated electromagnet does not receive 12V from the power supply, which could damage it. As the power through this 5.75 ohm resistor is 3W, initial, low-power resistor which I used started smoking due to the high power running through it. To solve this, I used two resistors in parallel with a power of 3W and resistance of 10hm.

A **flyback diode** (*3* and *4*) is placed in parallel with both the electromagnet and linear solenoid to protect them. When the coil in the electromagnet or linear solenoid is disconnected from the circuit, the drop in current causes the magnetic field to collapse, inducing a high current which creates a large, induced voltage across the coil in the opposite direction. As such, this voltage jump can cause a spark. However, as the flyback diode is forward-biased[69] when this induced voltage is created, it can conduct the current in a loop until the magnetic energy has disappeared (Cook, 2021).

A **220 ohm resistor** (*5* and *6*) is used to reduce the current flow through the GPIO pin when the MOSFETs are 'turning on', as the maximum current drawn by the MOSFET *Gate* 140mA, which is significantly greater than the maximum current flow through a GPIO pin of 16mA.

A **voltage divider** (*7)* is used to connect the *Echo* output on the **ultrasonic distance sensor** to the Raspberry Pi, as the HIGH value for *Echo* is 5V which is too high for the maximum 3.3V which can flow through the Raspberry Pi GPIO pin (Matt, 2016).

Two **MOSFETs** (*8* and *9*) are used to temporarily energize either the energy to release electromagnet or the linear solenoid actuator, depending on the output value of their associated GPIO pin. This GPIO pin is connected to the *Gate* on the MOSFET; a voltage across the *Gate* creates a path for current flow between the *Source* (connected to the component) and *Drain* (connected to 0V) of the MOSFET, energizing the component. The MOSFET I used has a gate threshold value[70] of 1.8V; the Raspberry Pi supplies 3.3V through the GPIO pin, which is sufficient to 'turn on' the MOSFET (The Pi Hut, n.d.).

[69] Forward-biased means that the flyback diode conducts in the same direction as the current flow.
[70] The gate threshold value is the minimum voltage across the *Gate* required to 'turn on' the MOSFET and allow current flow between the *Drain* and *Source*.

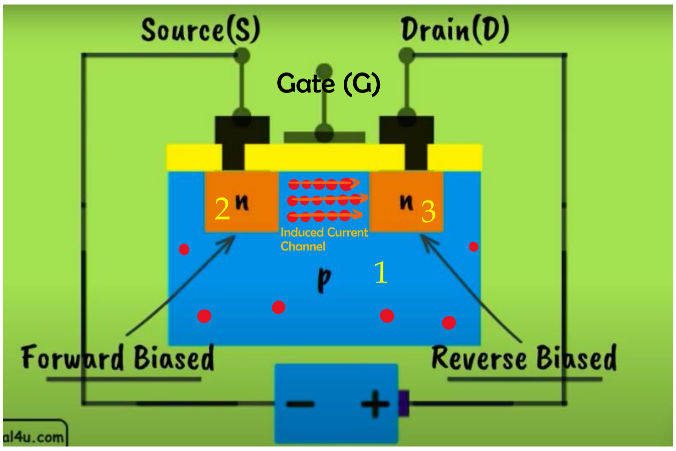


Figure 47, MOSFET diagram (Saurabh, 2019)

The ***Gate*** is a metallized layer which is insulated from the **p-type substance** (*1*). When no voltage is applied to the *Gate*, electrons can flow from the **n-type electrode** labelled (*2*) into the **p-type substance**, as the n-type substance is **forward-biased**. However, they cannot flow between the p-type substance and the *Drain,* as the **n-type electrode** labelled (*3*) is **reverse-biased**[71]. Therefore, when no voltage is applied to the *Gate*, current cannot flow between the *Source* and the *Drain*. When a voltage is applied to the *Gate*, it becomes positively charged[72], and therefore attracts the excess electrons in the p-type substance. As a result, an **induced conducting channel** is created between the two n-type electrodes (this phenomena is shown in *figure 47*), meaning current can flow between the *Source* and the *Drain* when a voltage is applied to the *Gate*.

Creating the circuit

During the prototyping phase, I used breadboards to create several temporary models of the circuit. This allowed me to rigorously test my circuit design, the behavior of the connected components and the associated code (detailed on *page XX*) without making permanent any aspect of the design. As such, when I ran into issues, it was easy to modify the circuit accordingly.

---

[71] Reverse-biased means that an n-type doped substance is connected to a positive terminal, meaning holes are formed at the barrier between the n-type and the p-type substance as electrons leave the n-type substance and flow towards the positive terminal. This creates a potential barrier between the p-type and n-type substance which opposes the flow of current/

[72] The delocalised electrons in the metal layer of the *Gate* will flow towards the positive potential of the Raspberry Pi GPIO pin which the *Gate* is connected to.

Figure 48 shows the breadboard version of the circuit schematic in *figure 46*.

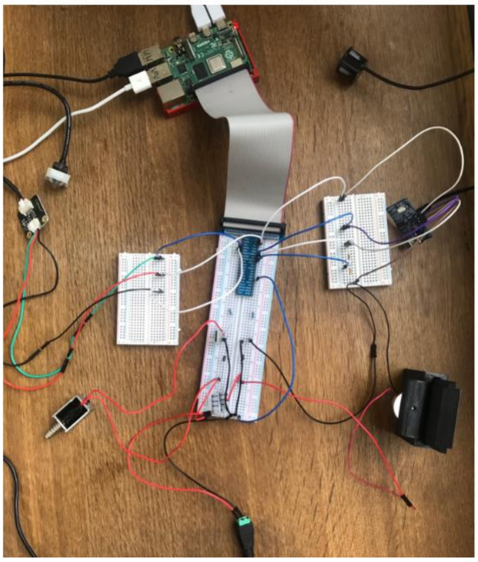


Figure 48, final breadboard model

The breadboard layout means that the circuit design is enlarged, with each section (i.e. the linear solenoid actuator, the energy to release electromagnet, the ultrasonic distance sensor and the liquid level sensor) having its own section. Moreover, each connection is easy to visualize. Therefore, I was able to easily isolate where any errors arose and use a **multimeter** to diagnose these.

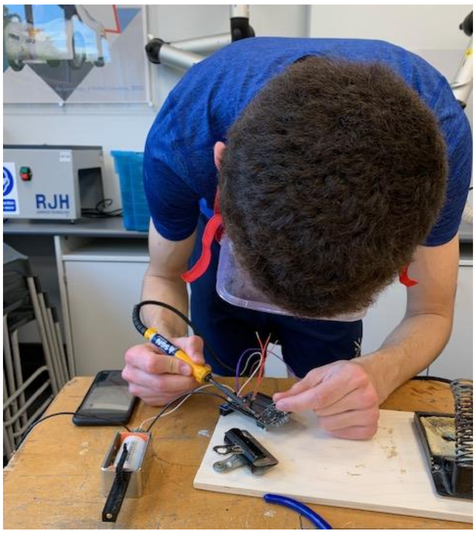


Figure 49, soldering the final circuit

Having finalized the circuit design using the three breadboards shown in *figure 48*, I used a **solderable breadboard**, a **soldering iron** and **solder wire** to permanently wire the circuit in a compact form (*figure 49*). Doing so required me to learn how to solder precisely to avoid any accidental connections being made, which could jeopardize the correct function of the circuit.

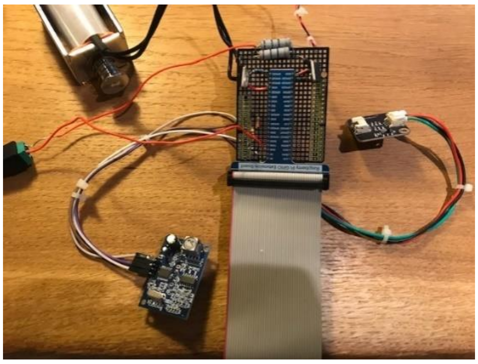


Figure 50, soldered circuit

Figure 50 shows the final soldered circuit which connects the sensor and actuators to the Raspberry Pi with the wiring detailed in the circuit schematic shown in *figure 46*.

Turbine and Generator

The turbine and generator are crucial elements of the hydroelectric system: as the grey wastewater flows across the turbine blades, it rotates the turbine rotor, inducing a current and thus generating electricity[73]. *Figure 51* shows a diagram of a turbine and generator.

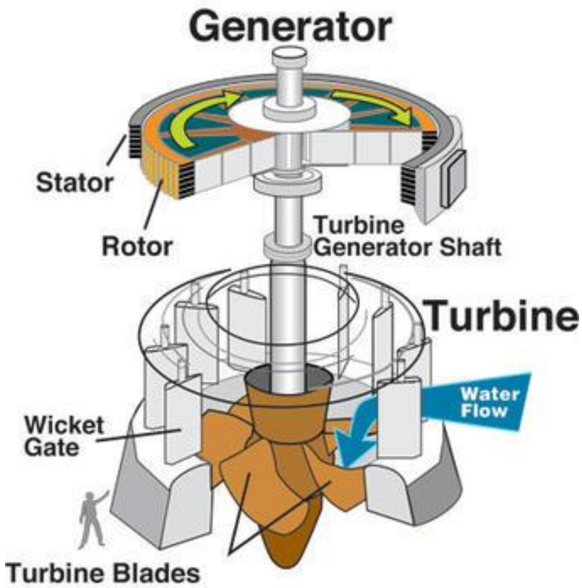


Figure 51, turbine and generator design (United States Geological Survey, 2018)

As turbine and generator systems cost upwards of £1,000, I decided to refrain from creating a complete and fully functional turbine and generator. However, I explored the technical specifications for the required turbine and constructed the **penstock**[74], **jet**[75] and container for the turbine.

[73] The rotor is a disk with electromagnets attached to its perimeter. When the rotor is rotated by the turbine shaft, the motion of these electromagnets induces current in the external, static conductor, generating electrcity (United States Geological Survey, 2018).
[74] The penstock pipe delivers the water from the grey wastewater pipe to the turbine (Renewables First, 2012).
[75] The jet is connected onto the end of the penstock pipe and it has conical shape which serves to increase the pressure of the water as it hits the turbine buckets.

The turbine

I chose to use a **Pelton turbine,** as they are the preferred choice for high water head applications (over 30m) which have relatively low flow rates, as is the case for my project. *Figure 52* represents this relationship between water head, flow rate and turbine type. The Pelton turbine (shown in *figure 53*) is installed vertically and uses discrete buckets to harness the kinetic energy of the water, so it is also fairly compact in design, which ensures minimal disruption to the external view of the apartment building on which it installed.

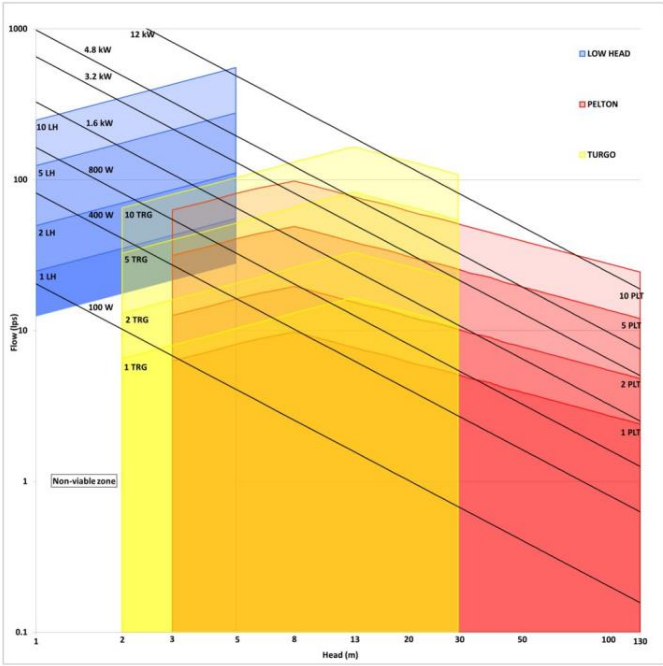


Figure 52, diagram showing relationship between flow rate, water head and turbine type (PowerSpout, n.d.)

Figure 53, model of Pelton turbine in container

Figure 52 shows my model of the **Pelton turbine** (*1*) in its **container** (*2*). Water flows into the container through the **jet** (shown in *figure 54*) at (*3*), impinging the buckets (*4*) on rotor and 'transferring around 97% of its kinetic energy into the rotational energy of the rotor' (Renewables First, 2012). The buckets have a central ridge which splits the water flow between the two halves of the bucket. Moreover, their design is such the water rebounds from the bucket by almost 180°, ensuring the maximum momentum change of the water, and therefore the greatest possible force is exerted on the rotating turbine. The water then falls to the floor of the turbine container and leaves through a **discharge pipe** (Renewables First, 2012).

As I concluded in my analysis on *page 26*, the power output of a turbine with a water flow rate of 3.1lps is 1.06kW. Therefore, the turbine in the system should have a power rating of 1kW. As mentioned on *page 18*, I have used *PowerSpout's*[76] online advanced calculator (PowerSpout, n.d.) in several calculations relating to water flow rate and the turbine; therefore, I suggest that the *PowerSpout Pelton HP* be used for my hydroelectric system to ensure optimal functioning.

In accordance with the recommendations from the *PowerSpout* online advanced calculator, the **jet** diameter should be 8.1mm at its narrowest point (which occurs as the jet enters the turbine container). As shown in *figure 54*, the cross-sectional area of the water flow is decreased as the water flows into the turbine container. Therefore, as $Q=AV$ (where Q = water flow rate, A = cross-sectional area of water flow and V = volume of water), the jet increases the flow rate, and therefore pressure, of the water at the instance prior to the water impinging on the turbine, maximizing the power output from the turbine.

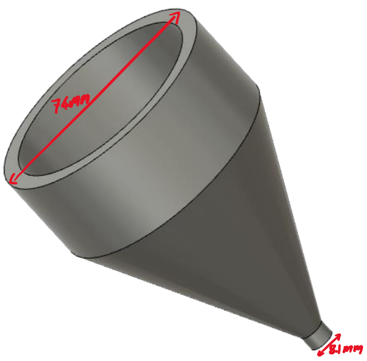


Figure 54, CAD design of turbine jet

Figure 55, wafer valve (Maric Flow Control Valves, n.d.)

As concluded on *page 20*, a **wafer valve** is used to limit the water flow rate to 3.1lps, reducing the head loss due to friction between the water and the pipe. The control rubbers (*1*) flex as the change in pressure across it varies, ensuring the flow rate remains constant. A **flange** (*2*) is used to integrate the wafer valve into the **penstock** pipe, shortly before the water flows into the turbine container.

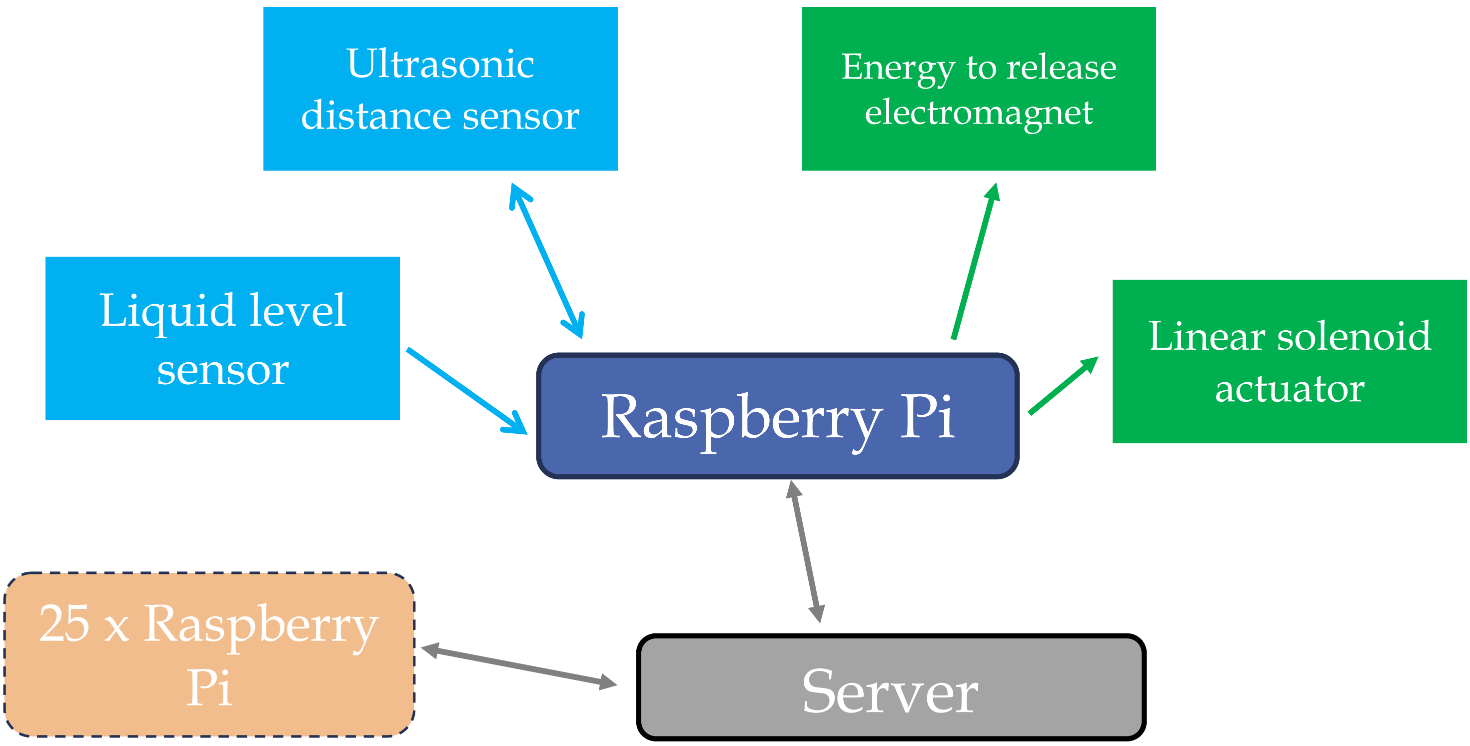
[76] *PowerSpout* is a manufacturer of micro-hydropower turbines. I have designed my hydroelectric system to use the *PowerSpout* Pelton HP turbine.

Software

In my project title, I emphasized that my design should have an element of *intelligence*, which maximizes the electricity generated by the system by minimizing the head loss. Therefore, when a particular section of piping is full of grey wastewater, the system should open the minimum number of barrier required to successfully redistribute the water within the downpipe and prevent the water from overflowing. To achieve this, the volume of water-free space in the pipe section(s) below the pipe section which is full of water must be found. As such, it is necessary to have a means of interfacing between the Raspberry Pi associated with the pipe section which is full of water, and the Raspberry Pi(s) associated with the pipe section(s) below.

I created a server program to act as this *interface* which receives feedback from the Raspberry Pis, processes this feedback and coordinates the appropriate action in response.

The relationship between the sensors, actuators, Raspberry Pis and server is shown here:



Communication protocols

- The communication between the **actuators** (*green*) and the Raspberry Pi is uni-directional. The actuators are connected to the Raspberry Pi via GPIO pins, which are set to HIGH or LOW voltage depending on whether they are to be energized or de-energized.
- The communication between the **sensors** (*blue*) and the Raspberry Pi is both uni-directional and bi-directional, depending on whether the sensor requires a trigger signal to read data. The ultrasonic distance takes a reading when a trigger signal is sent from the Raspberry Pi, and then returns the result of this reading to the Raspberry Pi via a GPIO pin. The liquid level sensor doesn't require a trigger, as it simply sends a signal to a Raspberry Pi GPIO pin at the instance water is detected.
- The communication between each Raspberry Pi and the **server** uses a framework called MQTT. For each area of communication required, a topic is created, to which messages/data can be published to and read from. The distribution of these messages is performed by a broker called *cloudmqtt*.

Code – Raspberry Pi

There are two programs which run on each Raspberry Pi:

1. *MQTT_Pi.py – see appendix 1*
   The main program which waits for message from the central server which are posted to the *actions* topic. When a message is received, the Raspberry Pi triggers the appropriate actuator. The possible actions are to obtain ultrasonic distance measurements or open or close the barrier mechanism.

2. *MQTT_Pi_Liquid.py – see appendix 2*
   A supplementary program which infinitely loops to detect for the presence of water (using the liquid level sensor) below the barrier mechanism, and notifies the central server when water is detected.

Every time either of these programs is run on the Raspberry Pi, SSH must be used to access and run the file on the Raspberry Pi from a remote device (such as a desktop or laptop). The **command line input** used to SSH into the Raspberry Pi is shown in *figure 46*:

```
C:\Users\aorla>ssh pi@192.168.0.16
pi@192.168.0.16's password:
Linux raspberrypi 5.10.11-v7l+ #1399 SMP Thu Jan 28 12:09:48 GMT 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep 18 11:54:44 2021
pi@raspberrypi:~ $ cd Desktop/EPQ/EPQ/
pi@raspberrypi:~/Desktop/EPQ/EPQ $
```

Figure 46, command line input to SSH into Raspberry Pi

Code – Server

The server manages the system, receiving feedback data from all 26 Raspberry Pis and processing this data, before coordinating the appropriate actions from sensors and actuators connected to a particular Raspberry Pi. The MQTT framework is used to communicate between the server and Raspberry Pis due to its lightweight nature, so it can be implemented with low latency[77] and little band-width requirements. Whilst developing the server code, I ensured that it was sufficiently robust to deal with all possible scenarios by running several lengthy tests to verify that the program did behave correctly in each possible scenario.

As well as creating the server program for a complete system, I have also written two other programs which run on the server to emulate the behavior of the other 25 barrier mechanisms and Raspberry Pis which would exist in a fully functioning system:

1. *mqtt_server.py – see appendix 3*
   The main program which runs on the server and coordinates which barrier mechanisms should open and which should close depending on the data it receives from the liquid level and ultrasonic distance sensors in each barrier mechanism.

2. *mqtt_all_pi.py – see appendix 4*
   A program which simulates the behavior of all the other programs titled '*MQTT_Pi.py*' running on the other 25 Raspberry Pis.

3. *mqtt_liquid_loop.py – see appendix 5*
   A program which simulates the behavior of all the other programs titled '*MQTT_Pi_Liquid.py*' running on the other 25 Raspberry Pis.

[77] Latency is the time delay between a message being sent and the message being received

Conclusion

In this project I designed all the aspects of a hydroelectric generator system for apartment buildings, using scientific theory, advice from experts and my own experiments to support my design decisions. I also produced a working prototype of the barrier mechanism, which is my original design that makes my system around 3.4 times as effective as existing equivalents. Throughout the report, I have explained the design decisions which make the project robust and appropriate for high-rise apartment buildings. Moreover, I have also thoroughly explored the ‘intelligent’ aspect of the design, using sensors in collaboration with a server which coordinates the entire system to maximize the water head and optimize the electricity generation. Therefore, my project has successfully fulfilled its aim and developed my love for electronics such that I am now planning to study electronics at University!

Bibliography

Absolute Medical Services (2018). *How does an Ultrasound Transducer work?* [online] Absmed. Available at: http://www.absolutemed.com/medical-equipment-blog/how-ultrasound-transducer-work.htm [Accessed 17 Feb. 2021].

Amazon (n.d.). *Module Sensors, DC Distance Sensor 5V Module Ultrasonic Sensor Waterproof Distance Measurement Board : Amazon.co.uk: Business, Industry & Science*. [online] www.amazon.co.uk. Available at: https://www.amazon.co.uk/Sensors-Distance-Ultrasonic-Waterproof-Measurement/dp/B07TS28BQW [Accessed Apr. 2021].

Ananthapadmanabhan, G.S. and Shouri, P.V. (2017). Pico Hydro Power Generation from Grey water in High Rise Buildings: Feasibility and Economic Analysis. *International Journal of Innovative Research in Science, Engineering and Technology*, [online] 6(6). Available at: http://www.ijirset.com/upload/2017/nftcos/21_paper%2021.pdf [Accessed 30 Aug. 2021].

Banas, T. (2018). *How to Calculate the Force of an Electromagnet*. [online] Sciencing. Available at: https://sciencing.com/calculate-force-electromagnet-5969962.html [Accessed 11 Feb. 2021].

Barleyman (2019). *Solenoid stroke vs force*. [online] Electrical Engineering Stack Exchange. Available at: https://electronics.stackexchange.com/questions/438784/solenoid-stroke-vs-force [Accessed 19 Sep. 2021].

BBC Bitesize (2019). *The total internal reflection of waves – WJEC - Revision 1 - GCSE Physics (Single Science) - BBC Bitesize*. [online] BBC Bitesize. Available at: https://www.bbc.co.uk/bitesize/guides/zctmh39/revision/1 [Accessed 19 Sep. 2021].

BBC Bitesize (n.d.). *Ultrasound - Sound and ultrasound - Edexcel - GCSE Physics (Single Science) Revision - Edexcel*. [online] BBC Bitesize. Available at: https://www.bbc.co.uk/bitesize/guides/zwf92nb/revision/2 [Accessed Mar. 2020].

Bijoy Kumar Behera, Hari, P.K. and Woodhead Publishing (2012). *Woven textile structure : theory and applications*. Oxford: Woodhead Publishing.

Bunting (n.d.). *Permanent Electro Holding Magnets - Energise To Release*. [online] Available at: https://e-magnetsuk.com/wp-content/uploads/2020/11/Electromagnets-Energise-to-Release-Data-Sheet.pdf [Accessed 19 Sep. 2021].

Bunting Magnets (n.d.). *Round Electromagnets 12 Volt*. [online] Bunting - eMagnets. Available at: https://e-magnetsuk.com/product/round-electromagnets-12-volt/ [Accessed Jan. 2021].

Collins Dictionary (n.d.). *Annulus definition and meaning | Collins English Dictionary*. [online] www.collinsdictionary.com. Available at: https://www.collinsdictionary.com/dictionary/english/annulus [Accessed 1 Sep. 2021].

Collins, R. and Beck, S. (2016). *Moody Diagram*. *Wikipedia*. Available at: https://commons.wikimedia.org/wiki/File:Moody_EN.svg#/media/File:Moody_EN.svg [Accessed 31 Aug. 2021].

Cook, J. (2021). *How Flyback Diodes Work – Snubber Diodes Explained*. [online] Arrow.com. Available at: https://www.arrow.com/en/research-and-events/articles/flyback-protection-diodes [Accessed 20 Oct. 2021].

Coursera. (n.d.). *Ian Harris, Instructor*. [online] Available at: https://www.coursera.org/instructor/ianharris [Accessed 21 Sep. 2021].

CQ Robot (2021). *Contact Water/Liquid Level Sensor SKU: CQRSENYW002 - CQRobot-Wiki*. [online] www.cqrobot.wiki. Available at: http://www.cqrobot.wiki/index.php/Contact_Water/Liquid_Level_Sensor_SKU:_CQRSENYW002 [Accessed 15 Feb. 2021].

Drainage online (n.d.). *80mm Round Downpipe x 4m | Floplast | Drainage Online*. [online] www.drainageonline.co.uk. Available at: https://www.drainageonline.co.uk/above-ground-drainage/guttering/floplast-guttering/floplast-80mm-round-plastic-downpipe/80mm-round-downpipe-x-4m-rph4 [Accessed 20 Feb. 2021].

Drainage Superstore (n.d.). *How does a soil pipe differ from a waste pipe?* [online] Drainage Superstore Help & Advice. Available at: https://www.drainagesuperstore.co.uk/help-and-advice/product-guides/plumbing-heating/how-does-a-soil-pipe-differ-from-a-waste-pipe/.

Eclipse magnets (n.d.). *Armature plates for electromagnets*. [online] www.eclipsemagnetics-shop.com. Available at: https://www.eclipsemagnetics-shop.com/armature-plate-c2x23313417 [Accessed 11 Feb. 2021].

Electronics Tutorials (2018a). *Electromagnet, Electromagnetic Coil and Permeability*. [online] Basic Electronics Tutorials. Available at: https://www.electronics-tutorials.ws/electromagnetism/electromagnets.html [Accessed Nov. 2020].

Electronics Tutorials (2018b). *Linear Solenoid Actuator Theory and Tutorial*. [online] Basic Electronics Tutorials. Available at: https://www.electronics-tutorials.ws/io/io_6.html [Accessed Apr. 2021].

Electronics Tutorials (n.d.). *NPN Transistor*. [online] www.physics-and-radio-electronics.com. Available at: https://www.physics-and-radio-electronics.com/electronic-devices-and-circuits/transistors/bipolarjunctiontransistor/npntransistor.html [Accessed 20 Sep. 2021].

Electronics Tutorials (2018c). *PNP Transistor Tutorial - The Bipolar PNP Transistor*. [online] Basic Electronics Tutorials. Available at: https://www.electronics-tutorials.ws/transistor/tran_3.html [Accessed Nov. 2020].

Engineering Library (n.d.). *Head Loss | Engineering Library*. [online] engineeringlibrary.org. Available at: https://engineeringlibrary.org/reference/head-loss-fluid-flow-doe-handbook [Accessed 31 Aug. 2021].

Engineers Edge (2014). *Water - Density Viscosity Specific Weight | Engineers Edge | www.engineersedge.com*. [online] Engineersedge.com. Available at: https://www.engineersedge.com/physics/water__density_viscosity_specific_weight_13146.htm [Accessed 31 Aug. 2021].

Fleischer, C. (2018). *How Piezoelectricity Works | EAGLE | Blog*. [online] Eagle Blog. Available at: https://www.autodesk.com/products/eagle/blog/piezoelectricity/ [Accessed 20 Sep. 2021].

Fluke (2021). *What is Duty Cycle?* [online] www.fluke.com. Available at: https://www.fluke.com/en-gb/learn/blog/electrical/what-is-duty-cycle [Accessed 19 Sep. 2021].

Holzner, S. (n.d.). *Calculating the Force Needed to Move an Object Up a Slope*. [online] dummies. Available at: https://www.dummies.com/education/science/physics/calculating-the-force-needed-to-move-an-object-up-a-slope/ [Accessed 30 Oct. 2020].

Labmate (2014). *What is the Reynolds Number Used For?* [online] Labmate Online. Available at: https://www.labmate-online.com/news/news-and-views/5/breaking-news/what-is-the-reynolds-number-used-for/32219 [Accessed 31 Aug. 2021].

Lumen (n.d.). *Electric Power and Energy | Physics*. [online] courses.lumenlearning.com. Available at: https://courses.lumenlearning.com/physics/chapter/20-4-electric-power-and-energy/ [Accessed 19 Sep. 2021].

Lumen (2019). *Flow Rate and Its Relation to Velocity | Physics*. [online] Lumenlearning.com. Available at: https://courses.lumenlearning.com/physics/chapter/12-1-flow-rate-and-its-relation-to-velocity/.

Mackay, D.J.C. (2013). *Sustainable energy - without the hot air*. Cambridge: Uit Cambridge.

Magma (n.d.). *Permanent Electromagnets*. [online] Magma Magnetic Technologies. Available at: https://www.magmamagnets.com/electromagnets-solenoids/permanent-electromagnets/ [Accessed 29 Oct. 2020].

Magnet Shultz (n.d.). *Linear: Enclosed, Tubular Solenoids*. [online] Magnet Schultz Ltd. Available at: https://www.magnetschultz.co.uk/products/solenoids/linear-enclosed-tubular [Accessed 19 Sep. 2021].

Maric Flow Control (n.d.). *WaterMark Quality Assured Biggest Range of Flow Rates Best Accuracy 20 Years Life Expectancy Maric Constant Flow Valves*. [online] Available at: https://z6j8k6w4.stackpathcdn.com/wp-content/uploads/Maric-Catalogue-V919.pdf [Accessed 1 Sep. 2021].

Maric Flow Control Valves. (n.d.). *Wafer Valve Type intended for flange mounting*. [online] Available at: https://www.maric.com/product-range/wafer-valve-type/ [Accessed 20 Oct. 2021].

Matt (2016). *Waterproof Ultrasonic Distance Measuring Module*. [online] Raspberry Pi Spy. Available at: https://www.raspberrypi-spy.co.uk/2016/10/waterproof-ultrasonic-distance-measuring-module/ [Accessed 17 Feb. 2021].

Miguel, G. (2021). *Will an electromagnet repel a permanent magnet? The top answer in the link states that a minimum current is required, but does the magnet...* [online] Quora. Available at: https://www.quora.com/Will-an-electromagnet-repel-a-permanent-magnet-The-top-answer-in-the-link-states-that-a-minimum-current-is-required-but-does-the-magnetic-field-of-the-permanent-magnet-have-to-also-be-significantly-less-than-that-of/answer/Miguel-Gonzalez-115 [Accessed 6 Oct. 2021].

Mould, S. (2019). *Piezoelectricity - why hitting crystals makes electricity*. *YouTube*. Available at: https://www.youtube.com/watch?v=wcJXA8IqYl8 [Accessed 20 Sep. 2021].

Nave, R. (2019). *Solenoids as Magnetic Field Sources*. [online] Gsu.edu. Available at: http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/solenoid.html [Accessed 19 Sep. 2021].

Pasternak, E. (2015). *Why is repulsion force weak between an electromagnet and a permanent magnet?* [online] Quora. Available at: https://www.quora.com/Why-is-repulsion-force-weak-between-an-electromagnet-and-a-permanent-magnet/answer/Eli-Pasternak [Accessed 29 Oct. 2020].

Pasternak, E. (2021). *Why does increasing the duty cycle of a linear solenoid decrease the potential force of the solenoid? The graph in the link shows that th...* [online] Quora. Available at: https://www.quora.com/Why-does-increasing-the-duty-cycle-of-a-linear-solenoid-decrease-the-potential-force-of-the-solenoid-The-graph-in-the-link-shows-that-the-force-at-any-point-in-the-stroke-is-less-if-the-duty-cycle-is-increased/answer/Eli-Pasternak [Accessed 19 Sep. 2021].

Physics Stackexchange (2021). *fluid dynamics - Bernoulli's principle for water in vertical pipe*. [online] Physics Stack Exchange. Available at: https://physics.stackexchange.com/questions/658226/bernoullis-principle-for-water-in-vertical-pipe/658244#658244 [Accessed 31 Aug. 2021].

Powerpipe (n.d.). *Power-Pipe Price and Sizes - Power-Pipe UK | Waste Water Heat Recovery Systems*. [online] powerpipehr.co.uk. Available at: https://powerpipehr.co.uk/price-and-size [Accessed 31 Aug. 2021].

PowerSpout (n.d.). *Pelton & Turgo Advanced Calculator*. [online] PowerSpout. Available at: https://www.powerspout.com/pages/advanced-plt-and-trg-series-calculator?model=TRG [Accessed 31 Aug. 2021].

PowerSpout. (n.d.). *Advanced Calculators*. [online] Available at: https://www.powerspout.com/pages/advanced-calculators [Accessed 20 Oct. 2021].

Ravi (2018). *What is a Photodiode? Working, Characteristics, Applications*. [online] Electronics Hub. Available at: https://www.electronicshub.org/photodiode-working-characteristics-applications/ [Accessed 20 Sep. 2021].

Recoup (n.d.). *Introduction to Waste Water Heat Recovery for Showers (WWHRS) | Recoup Learning*. [online] Recoup WWHR. Available at: https://recouplearning.co.uk/courses/waste-water-heat-recovery-for-showers-wwhrs/ [Accessed 31 Aug. 2021].

Recoup (n.d.). *Recoup WWHRS | Waste Water Heat Recovery (WWHRS) for showers*. [online] Recoup WWHRS. Available at: https://recoupwwhrs.co.uk/ [Accessed 31 Aug. 2021].

Renewables First (2012). *How much hydropower power can I get - Renewables First*. [online] Renewables First - The Hydro and Wind Company. Available at: https://www.renewablesfirst.co.uk/hydropower/hydropower-learning-centre/how-much-power-could-i-generate-from-a-hydro-turbine/ [Accessed 30 Aug. 2021].

RS Components (n.d.). *SD0630 (D-Frame Solenoid) Data Sheet*. [online] RS Online. Available at: https://docs.rs-online.com/c4ff/0900766b815dfd71.pdf [Accessed 28 Apr. 2021].

RS Components (n.d.). *SD0630 12V | Solentec Limited Linear Solenoid, 12 V dc, 29.5 x 15.9 x 20 mm | RS Components*. [online] uk.rs-online.com. Available at: https://uk.rs-online.com/web/p/linear-solenoids/9059931/ [Accessed 18 Aug. 2021].

Saurabh (2019). *What Is A MOSFET? | Basics, Working Principle & Applications*. [online] Electronics For You. Available at: https://www.electronicsforu.com/technology-trends/learn-electronics/mosfet-basics-working-applications [Accessed 20 Oct. 2021].

SST Sensing (2018). *Optical Liquid Level Sensors: How they work | SST Sensing*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=ZByijUX_TDY [Accessed 19 Feb. 2021].

Supermagnete (n.d.). *What does "magnetic field" & "magnetic flux" mean? - supermagnete.de*. [online] www.supermagnete.de. Available at: https://www.supermagnete.de/eng/faq/What-does-magnetic-field-and-magnetic-flux-mean [Accessed 19 Sep. 2021].

Switch Plan. (n.d.). *Average energy consumption: how many kWh is normal?* [online] Available at: https://www.switch-plan.co.uk/energy/consumption/#:~:text=The%20average%20UK%20household%20uses.

Teach Engineering (2021). *Measuring Distance with Sound Waves - Activity*. [online] TeachEngineering.org. Available at: https://www.teachengineering.org/activities/view/nyu_soundwaves_activity1 [Accessed 15 Feb. 2021].

The College of Estate Management (n.d.). *GREY WATER FOR UK HOUSING*. [online] pp.23–24. Available at: https://www.ucem.ac.uk/wp-content/uploads/2013/10/grey-water-final.pdf [Accessed 30 Aug. 2021].

The Engineering Toolbox (2019a). *Darcy-Weisbach Pressure and Major Head Loss Equation*. [online] Engineeringtoolbox.com. Available at: https://www.engineeringtoolbox.com/darcy-weisbach-equation-d_646.html [Accessed 31 Aug. 2021].

The Engineering Toolbox (n.d.). *Hazen-Williams Coefficients*. [online] www.engineeringtoolbox.com. Available at: https://www.engineeringtoolbox.com/hazen-williams-coefficients-d_798.html [Accessed 1 Sep. 2021].

The Engineering Toolbox (2019b). *Hazen-Williams Equation - calculating Head Loss in Water Pipes*. [online] Engineeringtoolbox.com. Available at: https://www.engineeringtoolbox.com/hazen-williams-water-d_797.html.

The Engineering Toolbox (n.d.). *Hydraulic Diameter*. [online] www.engineeringtoolbox.com. Available at: https://www.engineeringtoolbox.com/hydraulic-equivalent-diameter-d_458.html [Accessed 1 Sep. 2021].

The Engineering Toolbox (2019c). *Roughness & Surface Coefficients*. [online] Engineeringtoolbox.com. Available at: https://www.engineeringtoolbox.com/surface-roughness-ventilation-ducts-d_209.html [Accessed 31 Aug. 2021].

The Green Age (2012). *Greywater Recycling - TheGreenAge*. [online] TheGreenAge. Available at: https://www.thegreenage.co.uk/tech/greywater-recycling/ [Accessed 30 Aug. 2021].

The Pi Hut. (n.d.). *N-channel power MOSFET (30V / 60A)*. [online] Available at: https://thepihut.com/products/n-channel-power-mosfet [Accessed 20 Oct. 2021].

Translators Cafe (n.d.). *Convert gram-force [gf] to newton [N] • Force Converter • Common Unit Converters • Compact Calculator • Online Unit Converters*. [online] www.translatorscafe.com. Available at: https://www.translatorscafe.com/unit-converter/en-US/force/21-1/gram-force-newton/ [Accessed 19 Sep. 2021].

United States Geological Survey (2018). *Hydroelectric Power: How It Works*. [online] Usgs.gov. Available at: https://www.usgs.gov/special-topic/water-science-school/science/hydroelectric-power-how-it-works?qt-science_center_objects=0#qt-science_center_objects [Accessed 20 Oct. 2021].

Upton, P. (2020). *How do permanent electromagnets work? How does the current flow neutralise the magnetic field of the permanent magnet?* [online] Quora. Available at: https://www.quora.com/How-do-permanent-electromagnets-work-How-does-the-current-flow-neutralise-the-magnetic-field-of-the-permanent-magnet/answer/Peter-Upton-2 [Accessed 26 Oct. 2020].

Wikipedia (2021a). *Ferromagnetism*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Ferromagnetism#:~:text=Ferromagnetic%20materials%20can%20be%20divided [Accessed 18 Feb. 2021].

Wikipedia. (2020). *Hydraulic diameter*. [online] Available at: https://en.wikipedia.org/wiki/Hydraulic_diameter [Accessed 31 Aug. 2021].

Wikipedia. (2021b). *David J. C. MacKay*. [online] Available at: https://en.wikipedia.org/wiki/David_J._C._MacKay [Accessed 21 Sep. 2021].

Y., R. (2019). *What is a Phototransistor? Definition, Construction, Working, Characteristics Curve of Phototransistor*. [online] Electronics Desk. Available at: https://electronicsdesk.com/phototransistor.html [Accessed 19 Sep. 2021].

Yang, C.E. (2016). *StackPath*. [online] Electronicdesign.com. Available at: https://www.electronicdesign.com/power-management/article/21801833/what-is-the-piezoelectric-effect [Accessed 20 Sep. 2021]

Self-Evaluation

As displayed in my EPQ presentation, I successfully created a functional prototype of the barrier mechanism which can be rapidly opened and closed in response to signals sent by the central server. Experiment 2 showed that, when in the closed position, the barrier is held in place with sufficient force to support 12kg of water (the mass of water in each section of pipe). I also achieved the 'intelligent' element of the system by incorporating a liquid level sensor and ultrasonic distance sensor which feedback data to the server via the Raspberry Pi. The server code which I wrote (*shown in the appendices*) processes this data and coordinates which are barriers are to be opened. Moreover, I developed the code so that it can deal with all possible scenarios, making it a robust system. This is an incredibly important consideration when designing systems for use in public spaces (such as apartment buildings), so I ran many lengthy simulation tests to verify robustness of the server code.

As well as producing a successful artefact, I am also very proud of my development report. I faced many challenges throughout the design and development stages of my project, but I always persevered through them. As a result, every element of the project was a steep, and fascinating, learning curve; prior to completing my EPQ, I had never used a microcontroller or CAD software, built circuits, or worked on an actual engineering project. Writing the report allowed me to tie together all the technical knowledge I have gained, and improved understanding of how to approach an engineering project. For example, from my experiences, I have concluded that the optimal engineering development cycle is as follows: *Idea, Research, Design, Test Prototypes* and *Evaluate & Refine*. Interestingly, this is very closely aligned with typical development cycles given online.

Although I found my system to be 3.4 times more effective at generating electricity than similar tank-based systems, I am not convinced that any type of hydroelectric generator system for apartment buildings is commercially viable. By my calculations, my system has a with a maximum energy output of 2.63 kWh/day. To put that value into context, globally, the average energy consumption is 58kWh/day (Switch Plan, n.d.). However, this does not take away from the inherent success of the barrier mechanism prototype I designed and created, and all other elements of the system which I designed in great detail. Ideally, I would have liked to assembly all elements of the system, including the barrier mechanism, grey wastewater pipe, the wafer valve, the turbine and the generator. However, due to the cost and impracticality of building a fully-functional prototype, I believe I chose the correct approach to only create a prototype of the pipe barrier mechanism, as it is the element of the hydroelectric generator system which is completely my original design.

Throughout the process I regularly evaluated my designs and recorded any observations and sketches in the document titled *Visual Activity Log*. This organized approach allowed me to maintain an overview of all the elements of my project, which was important since I designed and created all aspects of the system in parallel. As such, I ensured that I was able to develop the system cohesively, with all individual designs contributing to the overall success of the project. However, while I was consistent in my evaluation of my designs, at times I was reluctant to interrogate the potential flaws with a design. For example, I spent a significant amount of time perfecting the design of the original linear solenoid actuator mechanism without having tested whether the linear solenoid actuator was powerful enough to move the sliding barrier into the closed position. Upon reflection, I believe I was initially reluctant to test the linear solenoid actuator, as I didn't want to find out that it didn't work. However, it is impossible to avoid issues in a project by ignoring them, as this will simply lead to the issues being discovered after further time has passed. Therefore, with the intention of minimizing time wasted working on a design which is inherently flawed, going forward, I will investigate all possible issues in a design before proceeding to further develop it.

Throughout the process, I have realized how challenging it can be to physically create a mechanical and/or electrical product which *actually* works. There are so many factors which must be exactly correct to ensure the whole system behaves as expected, so it is important to identify these factors in the initial designs. It is easy to disregard the design details when drawing a technical diagram or using CAD. However, I now appreciate the importance of considering each factor *before* you begin making complex prototypes, as it is costly and time-consuming to rely on prototypes to identify all the issues and to risk invalidating a large prototype because of one minor error. As I developed my affinity for engineering project development, I started isolating the areas where issues could arise before 3D printing a complete prototype. This meant I could produce small and specific designs for those individual areas. I could then quickly print these designs, evaluate their success, and either modify them and repeat the cycle, or print the entire product with confidence that each individual element would work.

Appendices

Appendix 1

MQTT_Pi.py

```
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import time


def get_ultrasonic_distance():
    # Speed of sound in cm/s at temperature
    temperature = 25
    speedSound = 33100 + (0.6*temperature)

    print("Ultrasonic Measurement")
    print("Speed of sound is",speedSound/100,"m/s at ",temperature,"deg")

    # Use BCM GPIO references
    # instead of physical pin numbers
    GPIO.setmode(GPIO.BCM)

    GPIO_TRIGGER = 18
    GPIO_ECHO    = 23
    GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
    GPIO.setup(GPIO_ECHO, GPIO.IN)

    # Set trigger to Low
    GPIO.output(GPIO_TRIGGER, GPIO.LOW)

    # Allow module to settle
    time.sleep(0.5)

    # Send 10us pulse to trigger
    GPIO.output(GPIO_TRIGGER, GPIO.HIGH)
    # Wait 10us
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, GPIO.LOW)
    start = time.time()

    while GPIO.input(GPIO_ECHO)==0:
        start = time.time()

    while GPIO.input(GPIO_ECHO)==1:
        stop = time.time()

    # Calculate pulse length
    elapsed = stop-start

    # Distance pulse travelled in that time is time
    # multiplied by the speed of sound (cm/s)
    distance = elapsed * speedSound

    # That was the distance there and back so halve the value
```

```
    distance = distance / 2

    # Converts the distance to metres and formats the distance to two decimal
places
    distance = "{0:.2f}".format(distance/100)
    if float(distance) > 5: # if the distance is less than 25 cm, the
ultrasonic distance sensor will give an invalid and large result
        distance = "0"
    print("Distance : {}".format(distance))

    # Reset GPIO settings
    GPIO.cleanup()

    return distance

def publish_message_distance(client_sensor):
    # function to publish data about the water level height in the pipe
section to the topic 'sensors/ultrasonic_distance'
    distance = get_ultrasonic_distance() # gets the distance to the water
level below the barrier of this client (which is client 4)
    print("Get distance below barrier 4")
    client.publish("sensors/ultrasonic_distance",
("{},{}").format(client_sensor, distance), qos=2) # publishes the client name
(floor number) and the distance to the water level in the pipe to the topic
'sensors/ultrasonic_distance'

def publish_message_next_distance(client_sensor):
    # function to publish data about the water level height in the pipe
section to the topic 'sensors/next_ultrasonic_distance'
    distance = get_ultrasonic_distance() # gets the distance to the water
level below the barrier of this client (which is client 4)
    print("Get (next) distance below barrier 4")
    client.publish("sensors/next_ultrasonic_distance",
("{},{}").format(client_sensor, distance), qos=2) # publishes the client name
(floor number) and the distance to the water level in the pipe to the topic
'sensors/ultrasonic_distance'

def publish_message_barrier_status(status):
    # function to notifies the server that the barrier is now closed by
publishing this 'Closed' status to the topic 'status/barrier_status'
    client.publish("status/barrier_status", status, qos=2) # publishes
'Closed' the topic 'status/barrier_status'

def on_message_get_distance(client, userdata, msg):
    # callback function which is called when the server publishes to the
topic 'sensors/ultrasonic_distance'
    msg = msg.payload.decode() # converts msg from bytearray to string
    if msg == clientNum: # if the server is asking for data from client 4
(the ID of the Raspberry Pi running this file), then the function
'publish_message_distance()' is called
#         print("Get distance!")
        publish_message_distance(clientNum) # calls the function
'publish_message_distance()'

def on_message_get_next_distance(client, userdata, msg):
    # callback function which is called when the server publishes to the
topic 'sensors/next_ultrasonic_distance'
    msg = msg.payload.decode() # converts msg from bytearray to string
    if msg == clientNum: # if the server is asking for data from client 4
(the ID of the Raspberry Pi running this file), then the function
'publish_message_next_distance()' is called
```

```
        publish_message_next_distance(clientNum)

def on_message_open_barrier(client, userdata, msg):
    # callback function which is called when the server publishes to the
topic 'actions/open_barrier'
    msg = msg.payload.decode() # converts msg from bytearray to string
    barrier = msg.split(",")[0] # the msg string contains both info about the
barrier to be opened and the distance to the water level below. These data
items are seperated by a ',' so the split function is used to seperate the
two data items; the first item is assigned to the variable 'barrier'
    distance = msg.split(",")[1] # the msg string contains both info about
the barrier to be opened and the distance to the water level below. These
data items are seperated by a ',' so the split function is used to seperate
the two data items; the second item is assigned to the variable
'ultrasonic_distance'
    # Use BCM GPIO references
    # instead of physical pin numbers
    GPIO.setmode(GPIO.BCM)
    GPIO_ELECTROMAGNET = 21
    GPIO_SOLENOID = 26
    GPIO.setup(GPIO_ELECTROMAGNET, GPIO.OUT)
    GPIO.setup(GPIO_SOLENOID, GPIO.OUT)
    GPIO.output(GPIO_ELECTROMAGNET, GPIO.LOW)
    GPIO.output(GPIO_SOLENOID, GPIO.LOW)

    if barrier == "All": # if all barriers are to be opened
        print("All barriers are open")
        # open barrier
        GPIO.output(GPIO_ELECTROMAGNET, GPIO.HIGH)
        GPIO.output(GPIO_SOLENOID, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(GPIO_ELECTROMAGNET, GPIO.LOW)
        GPIO.output(GPIO_ SOLENOID, GPIO.LOW)
        time.sleep(20)
        GPIO.output(GPIO_SOLENOID, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(GPIO_SOLENOID, GPIO.LOW)
        print("All barriers are closed")
        # closes all barriers
    elif barrier == clientNum: # if the barrier to be opened is this
Raspberry Pi's barrier (barrier 4)
        print("Barrier 4 is open")
        # open barrier
        GPIO.output(GPIO_ELECTROMAGNET, GPIO.HIGH)
        GPIO.output(GPIO_SOLENOID, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(GPIO_ELECTROMAGNET, GPIO.LOW)
        GPIO.output(GPIO_ SOLENOID, GPIO.LOW)
        if distance != "None":
            while float(distance) > 0.3:
                distance = get_ultrasonic_distance()
        else:
            time.sleep(5) # replace with time to empty bottom section of pipe
        # close barrier
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(GPIO_SOLENOID, GPIO.OUT)
        GPIO.output(GPIO_SOLENOID, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(GPIO_SOLENOID, GPIO.LOW)
        print("Barrier 4 is closed")
        publish_message_barrier_status("Closed") # calls function to notify
```

```
server that barrier is closed

def on_connect(client, userdata, flags, rc):
    if rc == 0: # if connection is successful
        print("Connected")
        client.subscribe("actions/get_distance") # subscribes to topic
'actions/get_distance'
        client.subscribe("actions/open_barrier") # subscribes to topic
'actions/open_barrier'
        client.subscribe("actions/on_message_get_next_distance") # subscribes
to topic 'actions/get_next_distance'
        client.message_callback_add("actions/get_distance",
on_message_get_distance) # creates callback for topic 'actions/get_distance'
        client.message_callback_add("actions/open_barrier",
on_message_open_barrier) # creates callback for topic 'actions/open_barrier'
        client.message_callback_add("actions/on_message_get_next_distance",
on_message_get_next_distance) # creates callback for topic
'actions/get_next_distance'

    else:
        # attempts to reconnect
        client.on_connect = on_connect
        client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
        client.connect("hairdresser.cloudmqtt.com", 18973)

clientNum = "4"
client = mqtt.Client()
client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
client.on_connect = on_connect # creates callback for successful connection
with broker
client.connect("hairdresser.cloudmqtt.com", 18973) # parameters for broker
web address and port number

client.loop_forever()
```

Appendix 2

MQTT_Pi_Liquid.py

```
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import time


def on_connect(client, userdata, flags, rc):
    if rc == 0: # if connection is successful
        print("Connected")
    else:
        # attempts to reconnect
        client.on_connect = on_connect
        client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
        client.connect("hairdresser.cloudmqtt.com", 18973)

client = mqtt.Client()
client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
client.on_connect = on_connect # creates callback for successful connection
with broker
client.connect("hairdresser.cloudmqtt.com", 18973) # parameters for broker
web address and port number


# Use BCM GPIO references
# instead of physical pin numbers
GPIO.setmode(GPIO.BCM)

# Define GPIO to use on Pi
GPIO_LIQUID = 4
GPIO.setup(GPIO_LIQUID, GPIO.IN)

while True:
    if GPIO.input(GPIO_LIQUID)==1:
        client.publish("sensors/liquid_level","5", qos =2)
        print("There's liquid!")
        start_time = time.time()
        while GPIO.input(GPIO_LIQUID)==1:
            if (time.time() - start_time) > 4:
                client.publish("sensors/liquid_level","All")
                print("There's still liquid - open all barriers!")
                time.sleep(5)
                break
            time.sleep(0.1)
    client.loop()
```

Appendix 3

mqtt_server.py

```
import paho.mqtt.client as mqtt

def open_barrier_get_distances(client_message):
    # function which coordinates the requests to the appropriate clients to
carry out and return ultrasonic distance measurements of the water level in
the pipe section below
    global ultrasonic_distances # accesses the global variable
'ultrasonic_distances'
    global ultrasonic_distance # accesses the global variable
'ultrasonic_distance'
    global barrier_open_range # accesses the global variable
'barrier_open_range'
    global active_client # accesses the global variable 'active_client'
    active_client = client_message
    ultrasonic_distances = []
    barrier_open_range = []
    if client_message == 1: # if the client which notifies the server that
its pipe section is full is the client with the number '1' (first floor
client), then only the ground floor client must open its barrier, so no
ultrasonic distance measurements are required
        barrier_open_range.append(0) # only the ground floor barrier (client
0) must be opened
        ultrasonic_distances.append("None") # as the ground floor barrier
runs directly ito the penstock pipe, there are no clients which must measure
ultrasonic distances in their pipe section
        ultrasonic_distance = 0 # the distance below the ground floor barrier
is set to '0' as it runs directly into the penstock pipe
        open_barrier()  # calls the function to open all the barriers and
release all the water in the pipe over the turbine
    for client in range((active_client-1), 0, -1): # iterates through all the
clients below the pipe section which is full of liquid
        publish_get_distance(client) # calls the function
'publish_get_distance' for each client elow the pipe section which is full of
liquid

def open_barrier():
    # function which opens one barrier at a time in order to redistribute the
liquid from the pipe which is full of liquid
    global barrier_open_range # accesses the global variable
'barrier_open_range'
    global ultrasonic_distance # accesses the global variable
'ultrasonic_distance'
    global ultrasonic_distances # accesses the global variable
'ultrasonic_distances'
    global barrier_status # accesses the global variable 'barrier_status'
    barrier = barrier_open_range[0] # the first element in the list
'barrier_open_range' is the next barrier to be opened
    barrier_open_range.pop(0) # the first element in the list
'barrier_open_range' is removed as this barrier has now been opened
    ultrasonic_distances.pop(0) # the first element in the list
'ultrasonic_distances' is removed as this distance has now been measured
    client.publish("actions/open_barrier", "{},{}".format(barrier,
ultrasonic_distance), qos=2) # publishes the barrier which is to be opened
and the water level below that barrier to the topic 'actions/open_barrier'
```

```
    barrier_status = "Open" # sets the global variable 'barrier_status' to
'Open' to signify that a barrier is currently open
    print("\nOpened barrier",barrier)
    print("Water level below
barrier",barrier,"is",ultrasonic_distance,"metres")

def open_barriers():
    # function which opens all the barriers
    global open_all_barriers
    open_all_barriers = True # sets the global variable 'open_all_barriers'
to True to signify that all barriers are currently open
    client.publish("actions/open_barrier", "All, None", qos=2) # publishes
'All' to the topic 'actions/open_barrier' so that all clients open their
barrier

def barrier_coordination():
    # function which coordinates the opening and closing of the barriers
    global barrier_open_range # accesses the global variable
'barrier_open_range'
    global ultrasonic_distances # accesses the global variable
'ultrasonic_distances'
    global ultrasonic_distance # accesses the global variable
'ultrasonic_distance'
    global active_client # accesses the global variable 'active_client'
    counter = 0
    print("\nActive client:", active_client)
    for i in ultrasonic_distances: # iterates through all the measurements of
water level below clients
        barrier = int(i.split(",")[0]) # each element in the list
'ultrasonic_distances' stores the barrier number and the water level below
it. The value before the ',' is the barrier number and is assigned to the
variable 'barrier'
        ultrasonic_distance = (i.split(",")[1]) # each element in the list
'ultrasonic_distances' stores the barrier number and the water level below
it. The value after the ',' is the barrier number and is assigned to the
variable 'ultrasonic_distance'
        if counter == 2:  # if the iteration through two measurements of
water level has not found any pipe sections which are sufficiently empty,
then all barriers are opened and all the water in the pipe is released across
the turbine
            print("\nOpened all barriers - checked 2 pipe sections below pipe
section",active_client,"and both are too full")
            ultrasonic_distance = ""
            open_barriers()  # calls the function to open all the barriers
            break
        elif float(ultrasonic_distance) < 1: # if the height above the water
level in the pipe section is less than 1 metre, then it is deemed to be an
isufficient amount of space to store the excess liquid on its own, so the
iteration continues
            if len(ultrasonic_distances) == 1: # if the active client is
client 2, then there will only be one available measurement of ultrasonic
distance (below client 1), as client 0 is on the ground floor and leads
directly into the penstock pipe
                barrier_open_range = [0,1] # only the ground floor barrier
(client 0) and first floor barrier (client 1) must be opened
                ultrasonic_distances = ["0", ultrasonic_distances[0]] # the
distance below the ground floor barrier (client 0) is 0 metres as it runs
directly into the penstock pipe
                ultrasonic_distance = 0
                open_barrier()  # calls the function to open all the barriers
and release all the water in the pipe over the turbine
```

```
                print("Barriers to be opened:", (", ".join(str(barriers) for 
barriers in barrier_open_range)))
                break
            else:
                counter+=1 # the value of counter is incremented by a value 
of '1' to indicate that the height of the water in another pipe section has 
been measured and has been deemed to have an isufficient amount of space to 
store the excess liquid on its own
        else:
            barrier_open_range = [item for item in range(barrier, 
active_client)] # creates a list which stores all the barriers which need to 
be opened, starting with the barrier furthest down the pipe
            ultrasonic_distances = 
(ultrasonic_distances[:len(barrier_open_range)]) # creates a list of all the 
client numbers and the water height in their pipe section below (i.e. their 
ultrasonic distances) required to coordinate the opening and closing of the 
barriers in 'barrier_open_range'
            ultrasonic_distances.reverse() # reverses the list 
ultrasonic_distances as the barriers are opened from the barrier furthest 
down the pipe upwards
            print("Barriers to be opened:", (", ".join(str(barriers) for 
barriers in barrier_open_range)))
            open_barrier() # calls the function 'open_barrier()' to open the 
first barrier which is to be opened
            break

def publish_get_distance(client_action):
    # function which publishes to the topic 'actions/get_distance' with the 
client numbers from which the server is request water level measurements
    client.publish("actions/get_distance", client_action, qos=2)

def publish_get_next_distance(client_action):
    # function which publishes to the topic 
'actions/on_message_get_next_distance' with the client numbers from which the 
server is request water level measurements
    client.publish("actions/on_message_get_next_distance", client_action, 
qos=2)

def on_message_barrier_status(client, userdata, msg):
    # callback function which is called when a client publishes to the topic 
'status/barrier_status'
    global ultrasonic_distances  # accesses the global variable 
'ultrasonic_distances'
    global ultrasonic_distance  # accesses the global variable 
'ultrasonic_distance'
    global barrier_status  # accesses the global variable 'barrier_status'
    global open_all_barriers  # accesses the global variable 
'open_all_barriers'
    global client_queue  # accesses the global variable 'client_queue'
    global busy  # accesses the global variable 'busy'
    msg = msg.payload.decode() # converts msg from bytearray to string
    busy = False # the global variable 'busy' is set to False as a client has 
indicated that the barrier(s) which were open is/are now closed, so the pipe 
system and the server is no longer busy coordinating the opening and closing 
of pipes
    if msg == "Closed" and open_all_barriers == False and 
len(barrier_open_range)>0: # if the barrier that was open has has now been 
closed, but there are still more barriers to be opened to redistribute the 
water that is in the pipe section which is full (as indicated by the fact 
that there are still elements in the list 'barrier_open_range)
        print("\nFinding water level in next pipe section")
```

```
        busy = True # the global variable 'busy' is set to True as the pipe
system and server are now busy again
        next_distance = int(ultrasonic_distances[0].split(",")[0]) # the
first element in the list 'ultrasonic_distances' is the next client whose
distance to the water level in its pipe section must be measured
        publish_get_next_distance(next_distance) # calls the function
'publish_get_next_distance()'
    elif msg == "Closed" and open_all_barriers == False and
len(barrier_open_range)==0: # if the barrier that was open has has now been
closed and there are no more barriers to be opened (as indicated by the fact
that there are no more elements in the list 'barrier_open_range)
        barrier_status = "Closed" # the global variable 'barrier_status' is
set to 'Closed' as all the barriers are now closed
        print("\nAll barriers are closed")
        print("Barrier status is now:", barrier_status)
        if len(client_queue) == 1: # if there is a client which is full of
liquid and is waiting for the water in its pipe section to be
released/redistributed
            print("\nNew barrier(s) to be opened from queue")
            busy = True # the global variable 'busy' is set to True as the
pipe system and server are now busy again
            client_message = client_queue.pop(0) # the first element in the
list 'client_queue' is the client which has been waiting for the water in its
pipe section to be released/redistributed
            open_barrier_get_distances(int(client_message)) # calls the
function 'open_barrier_get_distances()'
        elif len(client_queue) > 1: # if there is more than one client which
is full of liquid and is waiting for the water in its pipe section to be
released/redistributed, all barriers are to be opened to avoid any issues
            print("\nOpened all barriers - more than 1 client in queue")
            ultrasonic_distance = ""
            open_barriers() # calls the function 'open_barriers()'
    elif msg == "All closed": # if all the barriers were open and all the
barriers have now been closed
        open_all_barriers = False # the global variable 'open_all_barriers'
is set to False as all the barriers are no longer opened
        barrier_status = "Closed" # the global variable 'barrier_status' is
set to 'Closed' as all the barriers are now closed
        print("\nAll barriers are closed")
        print("Barrier status is now:", barrier_status)


def on_message_liquid_level(client, userdata, msg):
    # callback function which is called when a client publishes to the topic
'sensors/liquid_level'
    global ultrasonic_distances # accesses the global variable
'ultrasonic_distances'
    global ultrasonic_distance # accesses the global variable
'ultrasonic_distance'
    global barrier_status # accesses the global variable 'barrier_status'
    global open_all_barriers # accesses the global variable
'open_all_barriers'
    global client_queue # accesses the global variable 'client_queue'
    global busy # accesses the global variable 'busy'
    global barrier_open_range # accesses the global variable
'barrier_open_range'
    client_message = msg.payload.decode() # converts msg from bytearray to
string and assigns the data from the message (which is the client who sent
the message) to the variable 'client_message'
    print("\nPipe section",client_message,"is full")
    if client_message == "All": # if the client which was too full of liquid
```

```
has been waiting for too long for the water in its pipe section to be
released/redistributed, then all barriers are opened to avoid any issues
        print("Opened all barriers - waiting too long to open individual
barriers")
        client_queue = [] # clears all the clients from the queue as all the
barriers have now been opened
        open_barriers() # calls the function 'open_barriers()'
    elif open_all_barriers == True: # if the global variable
'open_all_barriers' is set to True, then all barriers are already open so
nothing needs to be done
        print ("All barriers already open")
        pass
    elif len(client_queue) == 1: # if there is already a client in the queue
waiting for the water in its pipe section to be released/redistributed
        if busy == True: # if the pipe system and server is still busy
opening barriers to redistribute water from another client whose pipe section
is full of liquid        if len(barrier_open_range) > 0: #
            print("Opened all barriers - server is busy and one client
already in the queue")
            client_queue = [] # clears all the clients from the queue as all
the barriers have now been opened
            open_barriers() # calls the function 'open_barriers()'
        else:
            print("Added pipe section", client_message, "to queue")
            print("New barrier(s) to be opened (from queue)")
            client_queue.append((client_message)) # the client which is too
full of liquid is added to the queue of clients waiting for the water in its
pipe section to be released/redistributed
            client_message = client_queue.pop(0) # the first element in the
list 'client_queue' is the client which has been water in its pipe section to
be released/redistributed
            open_barrier_get_distances(int(client_message)) # calls the
function 'open_barrier_get_distances'
    elif len(client_queue) > 1: # if there is more than one client in the
queue waiting for the water in its pipe section to be released/redistributed
- note: this situation should never arise!
        print("Opened all barriers - more than 1 client in queue")
        client_queue = [] # clears all the clients from the queue as all the
barriers have now been opened
        open_barriers() # calls the function 'open_barriers()'
    elif busy == True and len(client_queue) == 0: # if the pipe system and
server is busy but there are no clients in the queue waiting for the water in
its pipe section to be released/redistributed
        print("Added pipe section",client_message,"to queue")
        client_queue.append(client_message) # adds the client whose pipe
section is too full of liquid to the queue of clients waiting for the water
in their pipe section to be released/redistributed
    else:
        print("New barrier(s) to be opened")
        busy = True  # the global variable 'busy' is set to True as the pipe
system and server are now busy again
        open_barrier_get_distances(int(client_message)) # calls the function
'open_barrier_get_distances()'

def on_message_ultrasonic_distance(client, userdata, msg):
    # callback function which is called when a client publishes to the topic
'sensors/ultrasonic_distance'
    global ultrasonic_distances # access the global variable
'ultrasonic_distances'
    global barrier_open_range # access the global variable
'barrier_open_range'
```

```
    global active_client # access the global variable 'active_client'
    msg = msg.payload.decode() # converts msg from bytearray to string
    ultrasonic_distances.append(msg) # appends the msg data which includes
the client number (which is a Raspberry Pi) and the water level below this
client
    if len(ultrasonic_distances) == active_client-1: # if all the required
clients have returned their values for the distance to the water level in
their pipe section
        ultrasonic_distances.sort(key=lambda tup: int(tup[:tup.index(",")]),
reverse=True)  # sorts the list 'ultrasonic_distances' by the first element
in each tuple (which is the Raspberry Pi client number) so that the values of
the distances to the water level are in descending order, starting with the
client number below the clien which is too full of liquid
        barrier_coordination() # calls the function 'barrier_coordination()'
    else: # if not all the required clients have returned their values for
the distance to the water level in their pipe section yet
        pass

def on_message_next_ultrasonic_distance(client, userdata, msg):
    # callback function which is called when a client publishes to the topic
'sensors/next_ultrasonic_distance'
    global ultrasonic_distance # access the global variable
'ultrasonic_distance'
    print("Next distance: ",msg.payload.decode())
    ultrasonic_distance = msg.payload.decode().split(",")[1] # converts msg
from bytearray to string
    open_barrier() # calls the function 'open_barrier'

def on_connect(client, userdata, flags, rc):
    # callback function which is called when the server successfully connects
to the MQTT broker
    if rc == 0:
        # placing subscription command in the connection callback ensures
that subscription to a topic occurs after a successful connection to the
broker
        client.subscribe("sensors/liquid_level") # subscribes to topic
'sensors/liquid_level'
        client.subscribe("sensors/ultrasonic_distance") # subscribes to topic
'sensors/ultrasonic_distance'
        client.subscribe("sensors/next_ultrasonic_distance")  # subscribes to
topic 'sensors/next_ultrasonic_distance'
        client.subscribe("status/barrier_status") # subscribes to topic
'sensors/barrier_status'
        client.message_callback_add("sensors/liquid_level",
on_message_liquid_level) # creates callback for topic 'sensors/liquid_level'
        client.message_callback_add("sensors/ultrasonic_distance",
on_message_ultrasonic_distance) # creates callback for topic
'sensors/ultrasonic_distance'

client.message_callback_add("sensors/next_ultrasonic_distance",on_message_nex
t_ultrasonic_distance)  # creates callback for topic
'sensors/next_ultrasonic_distance'
        client.message_callback_add("status/barrier_status",
on_message_barrier_status) # creates callback for topic
'sensors/barrier_status'
    else:
        # attempts to connect again
        client.on_connect = on_connect
        client.username_pw_set(username="raspberrypi",
password="RaspberryP1")
        client.connect("node02.myqtthub.com", 1883)
```

```
ultrasonic_distances = []
barrier_open_range = []
client_queue = []
ultrasonic_distance = ""
barrier_status = ""
active_client = 0
busy = False
open_all_barriers = False

client = mqtt.Client()
client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
client.on_connect = on_connect # creates callback for successful connection
with the MQTT broker
client.connect("hairdresser.cloudmqtt.com", 18973) # parameters for the MQTT
broker web address and port number

client.loop_forever() # program loops indefinitely
```

Appendix 4

1. *mqtt_all_pi.py*

```
import paho.mqtt.client as mqtt
import random
import decimal
import time

def publish_message_distance(client_sensor):
    randomNum = decimal.Decimal(random.randrange(0, 297))/100
    client.publish("sensors/ultrasonic_distance",
("{},{}").format(client_sensor, randomNum), qos=2)

def publish_message_next_distance(client_sensor):
    randomNum = decimal.Decimal(random.randrange(0, 297)) / 100
    client.publish("sensors/next_ultrasonic_distance",
("{},{}").format(client_sensor, randomNum), qos=2)

def publish_message_barrier_status(status):
    client.publish("status/barrier_status", status, qos=2)

def on_message_get_distance(client, userdata, msg):
    msg = msg.payload.decode()
    if int(msg) != 4:
        publish_message_distance(msg)

def on_message_get_next_distance(client, userdata, msg):
    msg = msg.payload.decode()
    if int(msg) != 4:
        publish_message_next_distance(msg)

def on_message_open_barrier(client, userdata, msg):
    msg = msg.payload.decode()
    barrier = msg.split(",")[0]
    ultrasonic_distance = msg.split(",")[1]
    if barrier == "All":
        print("All barriers are open")
        # open all barriers
        time.sleep(20)
        print("All barriers are closed")
        # closes all barriers
        publish_message_barrier_status("All closed")
    elif barrier != "4":
        print("Barrier", barrier, "is open")
        # open barrier
        # call ultrasonic sensor to loop until (2.97 - ultrasound result -
0.2) == ultrasonic_distance
        # close barrier
        time.sleep(5)
        print("Barrier", barrier, "is closed")
        publish_message_barrier_status("Closed")


def on_connect(client, userdata, flags, rc):
    if rc == 0:
        client.subscribe("actions/get_distance")
```

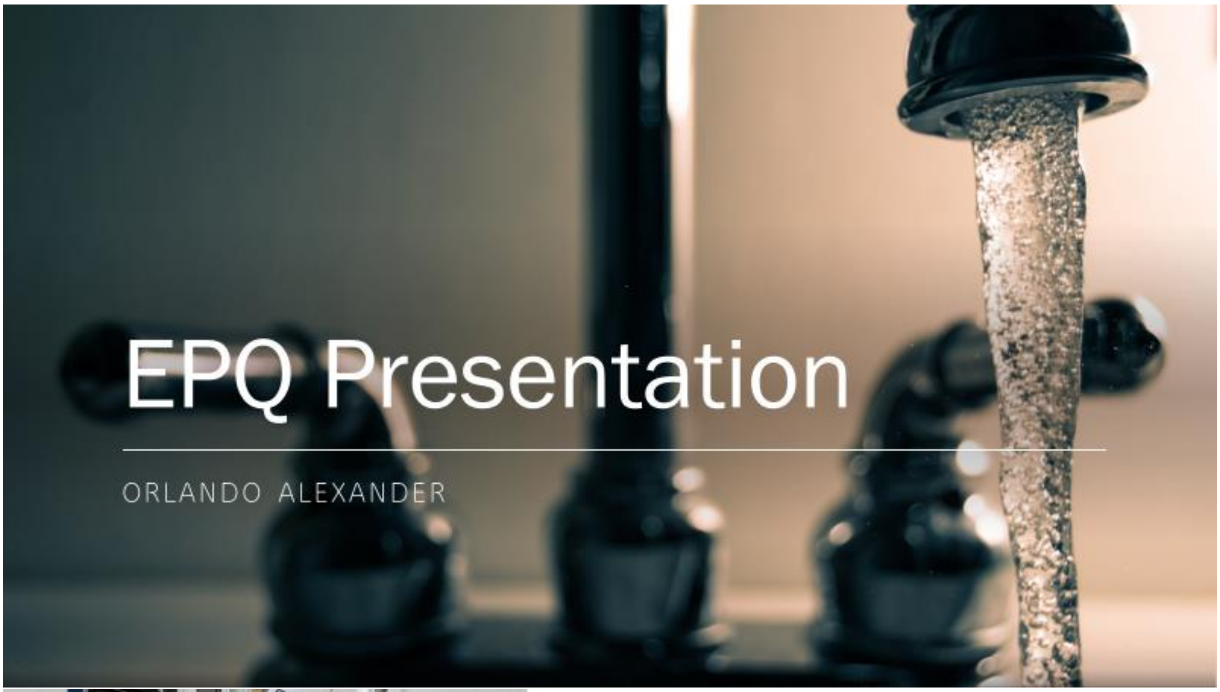
```
        client.subscribe("actions/open_barrier")
        client.subscribe("actions/on_message_get_next_distance")
        client.message_callback_add("actions/get_distance",
on_message_get_distance)
        client.message_callback_add("actions/open_barrier",
on_message_open_barrier)
        client.message_callback_add("actions/on_message_get_next_distance",
on_message_get_next_distance)
    else:
        # attempts to connect again
        client.on_connect = on_connect
        client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
        client.connect("hairdresser.cloudmqtt.com", 18973)




client = mqtt.Client()
client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
client.on_connect = on_connect
client.connect("hairdresser.cloudmqtt.com", 18973)

client.loop_forever()
```

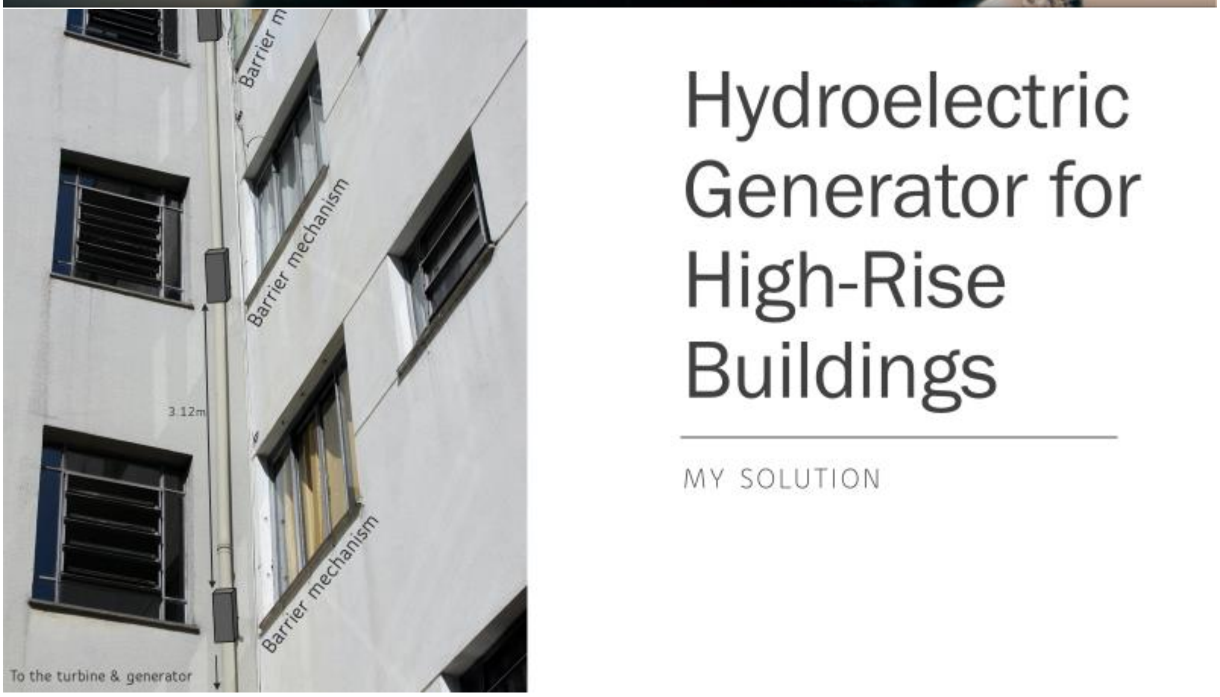
Appendix 5

2. *mqtt_liquid_loop.py*

```
import paho.mqtt.client as mqtt
import random
import time

clientNums =
[1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26] # the
first client (Raspberry Pi) which has sensors is on the first floor, as the
ground floor barrier mechanism is just a barrier mechanism on its own with no
sensors. The last client is number 26 and only has an ultrasonic distance
sensor and liquid level sensor, as there is no water above it

client = mqtt.Client()
client.username_pw_set(username="yrczhohs", password="qPSwbxPDQHEI")
client.connect("hairdresser.cloudmqtt.com", 18973)

while True:
    randomNum = random.choice(clientNums)
    client.publish("sensors/liquid_level", ("{}").format(randomNum))
    randomTime = random.randint(7,20)
    time.sleep(randomTime)
```

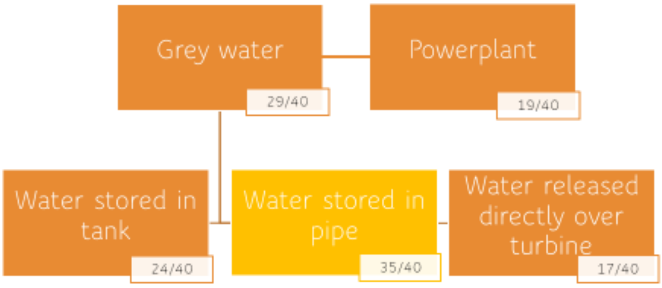
Presentation Slides





Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing

Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing

Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing

1. Equate *Bernoulli's equation* with *Darcy-Weisbach equation*:

$$\frac{V^2}{2g} = h - f_D \frac{L}{D}\frac{V^2}{2g}$$

2. Substitute h for L as the height of a vertical pipe is equal to its length:

$$\frac{V^2}{2g} = h - f_D \frac{h}{D}\frac{V^2}{2g}$$

3. Rearrange the above equation to make V the subject:

$$V = \sqrt{\frac{2gh}{1 + \frac{f_D \cdot h}{D}}}$$

4. Substitute the above equation into *Reynold's equation*:

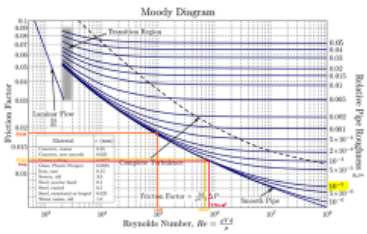
$$Re = \frac{D}{v}\sqrt{\frac{2gh}{1 + \frac{f_D \cdot h}{D}}}$$

5. Substitute actual values into the above equation[34]:

$$Re = \frac{0.076}{1.004 \cdot 10^{-6}}\sqrt{\frac{2 \cdot 9.81 \cdot 81.2}{1 + (\frac{81.2}{0.076}) f_D}}$$

$$Re = \frac{3.021 \cdot 10^{6}}{\sqrt{1 + (1089.2 \cdot f_D)}}$$

Calculating Reynold's number of water flow



Moody Diagram relates friction constant of system, Reynold's number and pipe roughness

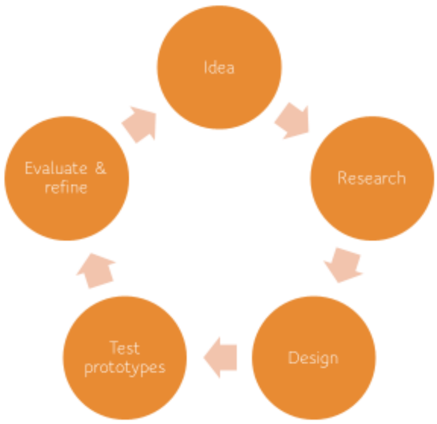
Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing

Wafer valve

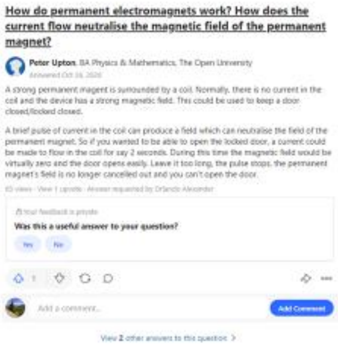
Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



How do permanent electromagnets work? How does the current flow neutralise the magnetic field of the permanent magnet?

Peter Upton, BA Physics & Mathematics, The Open University

A strong permanent magent is surrounded by a coil. Normally, there is no current in the coil and the device has a strong magnetic field. This could be used to keep a door closed/locked closed.

A brief pulse of current in the coil can produce a field which can neutralise the field of the permanent magnet. So if you wanted to be able to open the locked door, a current could be made to flow in the coil for say 2 seconds. During this time the magnetic field would be virtually zero and the door opens easily. Leave it too long, the pulse stops, the permanent magnet's field is no longer cancelled out and you can't open the door.

Was this a useful answer to your question?

Add a comment...

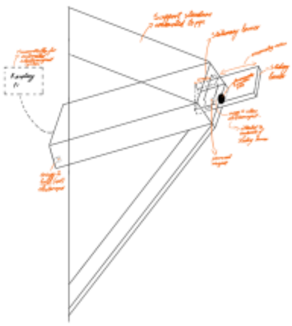
Add Comment

View 2 other answers to this question >

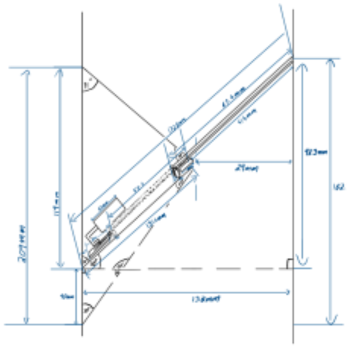
Quora question

Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



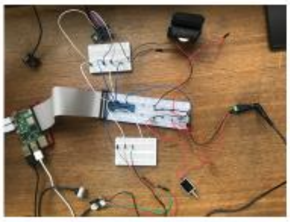
Initial sketch



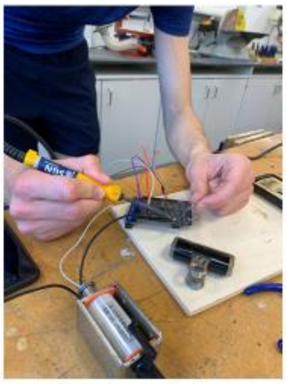
Technical drawing

Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



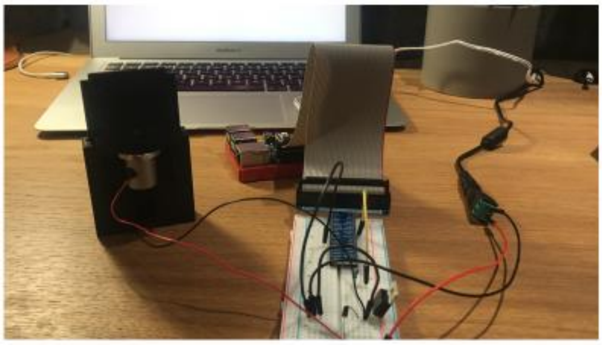
Basic prototype



Complex prototype

Development Process

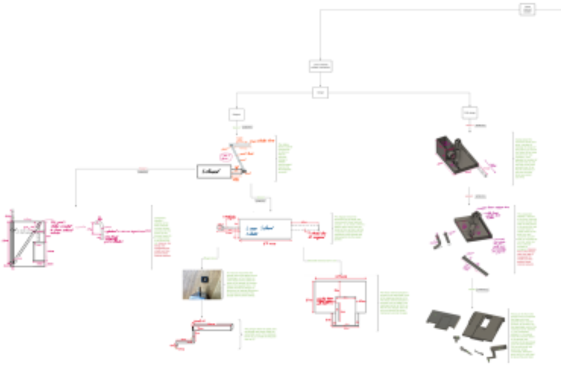
1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



Testing energy to release electromagnet

Development Process

1. Background research
   - Context
   - MOOCs/book
   - Investigate possible solutions
2. Mathematical analysis
   - Head loss
   - Wafer valve
3. Iterative development
   - Research
     - Articles/videos, interviews, forums
   - Design
   - Prototyping
   - Testing



Design flowchart

Challenges

1. Intricacy of mechanical motion in a working model
2. Simulating water weight
3. Interfacing between mechanics, electronics and software

Challenges

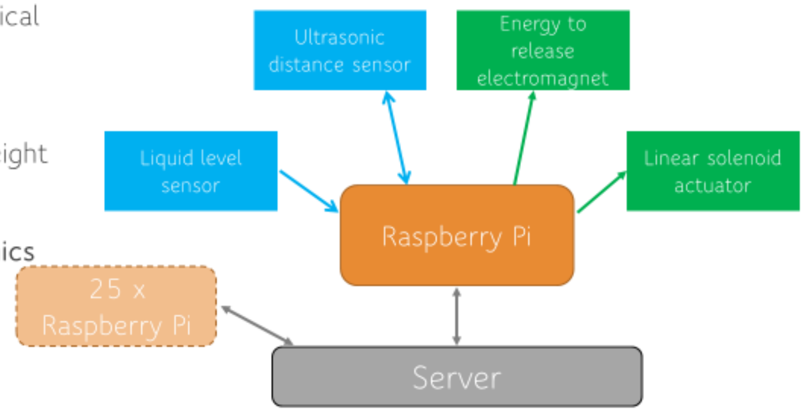
1. Intricacy of mechanical motion in a working model
2. Simulating water weight
3. Interfacing between mechanics, electronics and software

Experiment setup

Experiment video

Challenges

1. Intricacy of mechanical motion in a working model
2. Simulating water weight
3. Interfacing between mechanics, electronics and software



My favourite discovery...

Liquid level sensor

1. Based on principles learnt in A level physics
2. Rewarding to see theory learnt in the classroom be used for practical application

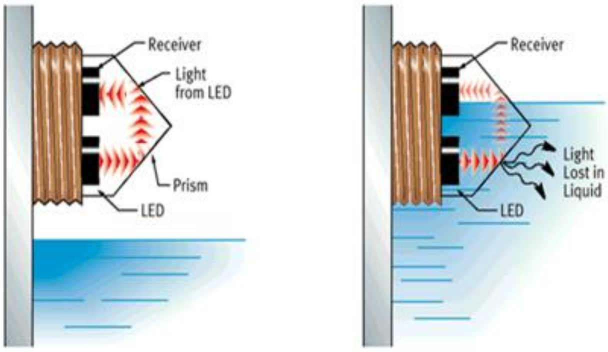


Figure 39, operation of liquid level sensor (SST Sensing, 2018)

Evaluation

1. If I had more time..
   - Watertight
   - Assemble system into single unit
   - Explore optimisations of the turbine and generator
2. Challenges of physically creating a mechanical and/or electrical product
3. Spotting issues early on
4. Learnt loads and discovered my passion for electronics!