Squeak
IT group 4

Sarah Fjelsted Alrøe
201905161

Ida Fink Baun
201908654

Jakob Houmøller Pedersen
201907706

Abstract

This paper describes the Squeak toy project in detail with regards to its use, related work, design of the prototype and limitations and future work. Production of the PCB, and 3D printed parts are described, and the function of main components of the circuit are explained. Problems during creation of the prototype are described, and challenges with regards to commercial production are discussed.

Introduction

The Squeak prototype is a robotic toy made for children from age three and up. The toy is shaped like a big mouse with two wheels under its' base. This allows it to move on flat surfaces. It has three different tails, which activate the three different modes of play that Squeak can be in. The child can remove a tail and insert a new one to change the play mode. The play modes are:

- Maze
- Line follow
- Hide and Squeak

In the *maze* mode (as shown in figure 1), the child has to build a maze of objects for Squeak to navigate through. Squeak avoids bumping into the object and tries to find the

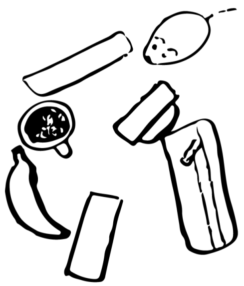


Figure 1: The Maze mode for Squeak.

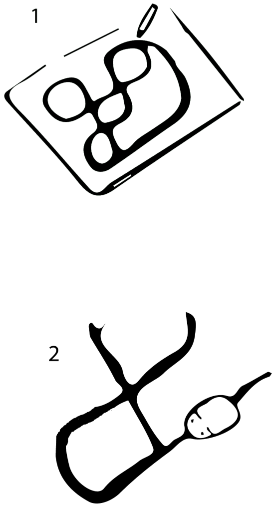


Figure 2: The Line mode for Squeak.

correct route out of the maze by trial and error. The tail for the *maze* mode is grey.

Another mode for Squeak, is *line follow* where Squeak follows a black line drawn by the child. Squeak has a wheel on each side of the line and moves with the line as shown in figure 2. If the line overlaps or an cross occurs, Squeak decides which line to follow. The tail for the *line follow* mode is black.

The last mode is the *hide and squeak* mode (this play mode is illustrated in figure 3). In this mode, Squeak tries to hide from the child by moving around until it finds a dark spot to hide. The tail for the *hide and squeak* mode is white.

The three different playmodes ensure that the child can train different types of learning. *Hide and Squeak* helps the child develop their gross motor skills, as they have to move around to try and find Squeak. *Line following* helps the child develop both fine motor skills and creative by requiring lines to be drawn for Squeak to track. Depending on the objects used and how the maze is put together, *Maze* can train both fine and gross motor skills and also creative play.

When a tail is plugged into Squeak, it is turned on and in the corresponding game mode. When a tail is removed, Squeak is not in any mode.

Use scenario
The scenario depicts Marius, a four year old boy, playing with Squeak.

Marius is sitting on the floor in his room and wishes to play with Squeak. Squeak has no tail connected, and is thus not in any play mode. Marius finds the white tail because he wants to play in the hide and squeak mode. He then connects the tail to his Squeak and puts it on the floor. The

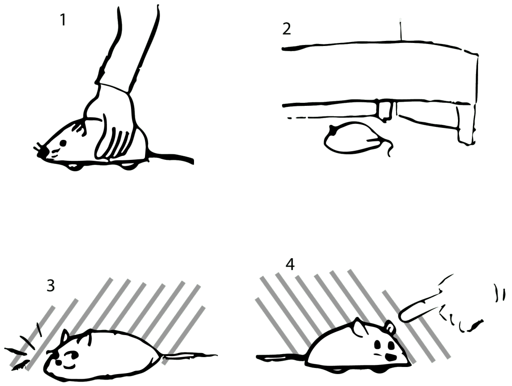


Figure 3: The Hide and Squeak mode for Squeak.

mouse turns on with a playful squeak, and first turns around and then begins to move around on the floor. Marius moves up into his bed and covers his eyes and waits for Squeak to squeak to signal it has found its hiding position. After he hears the squeak, he leaves the bed and begins to look for Squeak. First Marius searches under the bed and then his doll house but he can not find it. A small squeak sounds from under his bookcase, and Marius goes over to his bookcase and finds his Squeak under it. He grabs it and it squeaks, as it has been found.

Marius plays another round of hide and squeak by placing Squeak on the floor again. After Marius has found Squeak again, he wants to play the maze mode. He pulls the white tail from Squeak to turn it off while he build the maze. He arranges different toys from his room in a maze-like structure, then puts the grey tail into Squeak to activate the maze mode and places Squeak at the start of his maze. Squeak tries to navigate through the maze and does it without bumping into any of the objects. After Squeak has finished the maze, Marius pulls the tail out again.

He brings Squeak and the black tail into the living room and asks his dad for some paper and a black marker. After he has been given the materials he sits at the table and begins to draw a long, black line on the paper with an intersection. He then plugs the black tail into Squeak to activate the line follow mode. He places Squeak with a wheel on each side of the line, and it begins to follow the line, turning a bit in the correct direction whenever the line is not straight. When it reaches the intersection, it continues forward. When it reaches the end, Marius picks it up and it squeaks and stops moving. He pulls the tail as he is done playing with Squeak.

Related work

In this section, we look at commercial products with aspects similar to some of those of Squeak, relate Squeak and its interaction style to different papers detailing interaction with artefacts and look closer at the effects Squeak can have on training of fine and gross motor skills for children.

We have defined Squeak as a robotic toy, that can engage children in different play modes. In the article, *How do you play with a robotic toy animal?*[5], robotic toys are described as interactive, active artefacts that interacts directly with the world, and has a software component. The robotic toys should be *"[..]intended for basic leisure activities such as play, creativity, playful learning, entertainment, and relaxation"*. The main toy they discuss in the article is a toy shaped like a dinosaur called Pleo. In contrast to Squeak, Pleo has no predefined modes or activities the user can play in which gives it a very open-ended interaction. This open-ended interaction can make the Pleo seem more alive. This however also makes it very difficult to implement its behaviour. Pleo has a number of sensors and motors to make it move and react to the world around it. As with Squeak, it also has IR sensors to detect obstacles nearby. The article describes that many robotic toys today are shaped like animals, to awaken the child's feeling of having an artificial pet that can be played with. Pleo is seen as more pet like than toy like due to its behaviour, and thus the basis for Squeak and Pleo is different, as Squeak's behaviour is defined by different modes and the interaction is not as open-ended. This makes Squeak more toy like and less pet like.

Similar to the line-following mode of Squeak, a commercial robot toy[7] has a use case in which children draw out a track with black marker on a white surface for the toys to follow. As demonstrated on the product page, this toy is capable of crossing intersects, as Squeak also is. The product has similar limitations to Squeak: The lines have to adhere to a certain width, and turns must not be too sharp.

As mentioned in the article, *Characterization of fine motor development: Dynamic analysis of children's drawing movements*[6], drawing trains children's fine motor skills. They argue in the article, that drawing with a pen has a kinetic focus, as force needs to be applied to make the pen leave a trace. This gives a richer interaction and can contribute to learning more fine motor skills, than if the children had to draw on a non-force-sensitive interface, like a tablet. This means that the line-following mode of Squeak can train children's fine motor skills, as they have to draw lines on paper to interact with their Squeak in that play mode. And the fact that the children have to interact with the physical world (pen and paper) to create a basis for Squeak in its line following mode, also gives a richer interaction than it would if all the interaction took place on a tablet or computer.

In the following article, *Gross and fine motor proficiency in preschoolers: Relationships with free play behavior and activity level*[2], it is discussed how different children have a different frequency of play, that stimulates either their fine or gross motor skills. This means it can be difficult to create an all-around toy that any child can use and wants to use, if it only stimulates either the fine or the gross motor skills. Because playing with Squeak can engage both fine and gross motor skills, a broader range of children can play with Squeak, and it can be engaging for a longer period of time to play with it, as the child can switch modes where new skills are in focus.

In Djajadiningrat's article [3] regarding tangible products and interaction design, he presents two different approaches, a semantic and a direct approach. Squeak has elements of both the semantic and direct approach: The tails semantically represents a play mode. Because the different coloured tails tell nothing about the play mode, there is no feed-forward in the tails. The direct approach is visible as the tails fit perfectly into the connector and when a tail is connected it looks more like a natural mouse (as mice have tails). So looking at Squeak without a tail connected can tell the user that something is missing. Djajadiningrat also presents the term *freedom of interaction* where a task can be completed by doing actions in a non-fixed order, and the user is not constrained in the interaction with the artefact. Squeak has elements of freedom of interaction, but not complete freedom, as the user still needs to insert a tail first, before doing anything else. However the user can draw any shape of a black line and build a maze using any objects. This means the user can interact in numerous non-constricted ways with Squeak.

Prototype design

Overall prototype

Squeak is a prototype of a toy mouse with two wheels and three different sensor combinations, that can collaborate to create different playmodes and give the mouse an overall behaviour. The sensor combinations used are two avoidance sensors, one lone LDR, and two LDRs combined with two LEDs.

The prototype also features a buzzer to make sounds similar to the squeaks of a real mouse.

To change the game modes, there is a 3,5 mm jack connector attached to the back of the mouse, with three different 'tails' that can be connected to the connector. The tails have a 3,5 mm jack and have either the tip or the ring or both shorted to the sleeve, so they can be distinguished.

The mouse can move using two DC motors with attached 3D printed wheels and tires.

In software, Squeak runs in a continuous loop, where various methods are called depending on the selected game mode. By separating game modes into methods, one game mode like hide and squeak can extend the functionality of another (maze, where obstacles are avoided), by performing its own actions and then calling the method of the other game mode. This way behaviour can be dynamically combined and modified.

The finished prototype of Squeak is shown in figure 4.

A drawing of the finished prototype seen from the side and the top is shown in the appendix "3D model drawings of Squeak from the side and top". More pictures of the outer

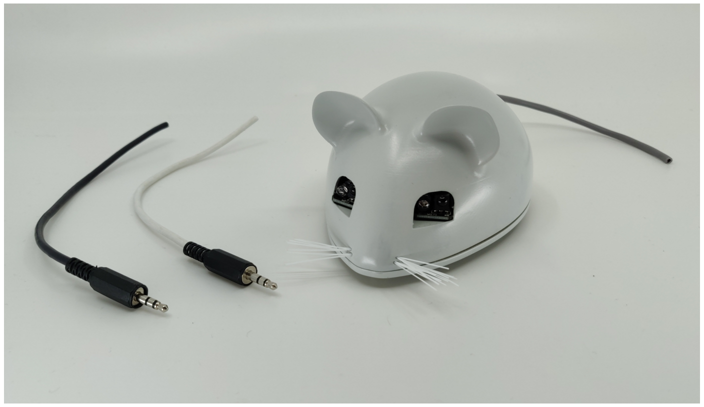


Figure 4: The finished Squeak prototype.

prototype can be seen in the appendix "Outer prototype gallery".

Circuit diagram
The circuit diagram has had minor changes since assignment 1, specifically changes to the pinout of the ATMEGA and addition of the mode-selecting jack-socket. The updated diagram with annotations matching the annotations in figure 5 can be found in the appendix "Annotated circuit diagram".

PCB
Squeak contains two custom made PCBs with through-hole components attached. The PCBs were made through a subtractive method. Photoresist covered FR-4 boards were exposed in UV light using a printed transparent mask and then developed and etched. In figure 6 the PCBs after the etching process are shown. After the boards were etched, holes were drilled in them. We have drilled both small holes for components, slightly larger holes for mounting the PCBs onto the base of the mouse and lastly two big, oblong holes on the largest PCB for the two LEDs and two LDRs of the line following sensor to stick through. The PCBs with holes and without components can be seen in figure 7.

The PCBs have been split into two, because the motors divide the mouse into two halves on the inside. All the components could have been added to a single PCB, but it would have an odd shape with a very thin, fragile middle strip, and would have wasted material and space. Instead, the two boards are interconnected by wires attaching to pin-headers on the boards. The annotated PCBs are shown in figure 5. A larger figure of the annotated PCBs is attached in the appendix "Annotated PCB".

The small PCB contains:

Ⓐ The motor control circuit

Ⓑ The jack tail connector

Ⓒ The power supply

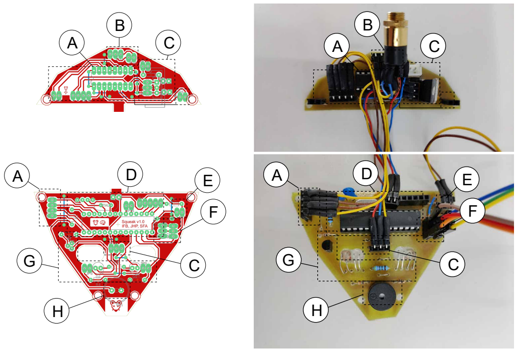


Figure 5: Annotated PCBs.

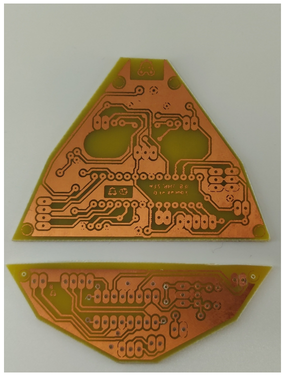


Figure 6: The PCBs for Squeak after the etching process.

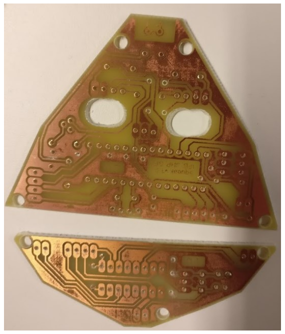


Figure 7: The PCBs for Squeak with drilled holes and without any components attached.

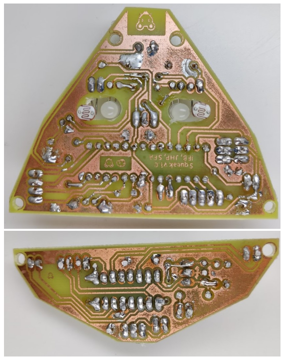


Figure 8: The PCBs for Squeak with components soldered on seen from the underside.

The large PCB contains:

Ⓓ The ATMEGA

Ⓔ The ambient sensing LDR

Ⓕ The avoidance sensors

Ⓖ The LED and LDR sensor pairs

Ⓗ The buzzer

As can been seen in the figures, we have chosen to have a ground plane on the PCBs. This is done to save some work of mapping out all the grounded wires. There are holes added to the PCBs to mount them onto the base where there are printed pins sticking up.

Because of the numbers and sizes of the components, the PCBs could not have been made much smaller. We have prioritized getting the smallest possible mouse, so this is also why we have some connecting jumper wires on the PCBs to connect components and the ground plane. If we had not included the jumper wires, we would have to make the PCB larger.

The PCBs with components attached are shown in figure 8. To make the wiring more robust, we have added heat shrink tubing to all the wires.

We had another set of functional PCBs with soldered on components before the current PCBs. But on the old PCBs the connector for the tails was not included. So when we changed the circuit diagram and PCB layout to incorporate the connector, we had to make new PCBs as well.

Estimated prototype price

We have made calculations to find the estimated total price of the prototype. The estimated price is 300,77 DKK. The full table of the individual costs for the components and materials used for the prototype is in the appendix "Price table for prototype". We have added both the price for the components and their shipping costs to the estimated full price of Squeak.

Main components

In the following subsections, the main components of Squeak are described with regards to their functionality, how they are integrated in the circuit, how they collaborate and how they are controlled.

Motors and control

To control the motors, an L293D quadruple half H-bridge IC [13] is used. The IC recieves 5v control signals on four pins from the ATMEGA to power the two motors in forwards and backwards directions at the battery voltage. The operation of the IC and its power calculations, based off its datasheet, is described in greater detail in assignment 1.

We are using 200 RPM N20 Micro Metal Gear Motors [9] to run Squeak. These are 12v motors, that are geared to run at 200 RPM when fully powered on. As the motor is geared, it is very fast to start and stop, an has an impressively high torque for its size. The motor shaft is 'D' shaped, which allows easy mounting of printed parts. The motors in use have been replaced since assignment 1. The reasons for this have been described in greater detail in the section "Finding the right motor".

In software, the motors are controlled by the Move class. Control of the two motors have been abstracted, such that movement is controlled by setting a direction (forwards, backwards), a speed, and an angle to turn at (ranging from

-2 to 2, with 0 being straight ahead). Additionally, a system has been added for performing 'canned' movements. This allows for a set of up to 16 combinations of direction, speed and angle to be scheduled to be performed in turn at various intervals. By scheduling the movements instead of making them blocking actions, other functionality can be maintained at the same time, like collecting sensor data or playing squeak sounds. The playback of squeaks use a similar task scheduling system for the same reasons. The largest blocking delay in the software is 1 ms (to allow the LEDs and LDRs to stabilise between reads).

Line tracking

Two tracking sensors are created discretely from a voltage divider using a light dependent resistor[11] and a $10k\Omega$ resistor next to a white LED[4], controlled by the MCU through a transistor.

The brightness of the surface underneath the sensor can be determined by measuring the voltage between the shared connection of LDR and fixed resistor, and ground. When the surface is bright, the resistance of the LDR will be low, and the voltage will therefore be high. When the surface is dark, the resistance will be high, and the voltage low. Additionally, by repeatedly turning the LEDs on and off and measuring the LDR voltages in each state, the amount of light reflected back onto the LDRs can be calculated, and used to sense when Squeak is picked up or put down onto a surface. Calculations relating to this part of the circuit are further detailed in assignment 1.

The LEDs and LDRs are bent down through the holes in the base, directly facing the surface which Squeak is driving on. As seen in other projects, infrared could have been used instead. However, through early prototyping we decided on this solution, as we found that the LDR and LED combination worked well, uses easily available components, and is a suitably small and simple solution for line tracking.

In software, line tracking is implemented in the LineFollower class. In the main loop, the observe() method of the object is called regularly, which measures the values of the LDRs with the LEDs on and the LEDs off, and then filters the results. When the line tracking mode is active, the filtered input values are used to calculate a fitting speed and direction, which is fed directly to the motor control. If light is sensed under the threshold, it means a black surface is sensed. If the black surface is sensed on one side, the motors will correct the direction back on track by turning that same direction. If both sensors sense a black surface, the motors slow down and the direction is turned slightly until a white surface is detected again, making it possible to cross intersections and choose a direction on a splitting path.

The threshold used for the sensor has been experimentally determined instead of calculated from a real world value. This is because we have not found a fitting datasheet for the specific LDRs we are using, and have not had access to a lux-meter or similar tool to measure luminosity. This is also the case for the ambient light sensor.

The input signals of the LDRs are filtered to reduce noise and to provide a decisive movement of the motors. To filter the signal, we use a running moving average avg_{now} calculated as such:

$$avg_{now} = \frac{(N-1) * avg_{prev} + in_{now}}{N}$$

Where avg_{prev} is the previous average and in_{now} is the new data point. The coefficient N determines how much influence a new data point has on the average. Through experimenting, we found N = 12 to be a good coefficient, meaning that any new input has 1/12 influence on the average. The selected coefficient results in a clean averaged signal. Increasing the coefficient would result in the movement being less responsive.

We opted to use running moving averages for filtering in this project as opposed to using a simple moving average, as it requires significantly less dynamic memory to calculate an average. This is important as the project currently has five averaged values in use. To compare the two types of averaging, they were graphed using simulated data. The result can be seen in the appendix "Comparative simulation of Running Moving Average and Simple Moving Average", and shows that the differences when configured right are negligible.

The same method is used for ambient light sensing, where N = 32 is used instead.

Object avoidance

For object avoidance, we use two KY-032 IR avoidance sensor modules[1]. There are two potentiometers on the modules, that can be manually turned to set the sensing distance and the frequency of the IR signal. The avoidance sensors for Squeak have had their distance sensing potentiometers adjusted so they detect flat, white obstacles at approximately a 5 cm distance.

The avoidance sensors work by sending out pulses of infrared light, and measuring how much of it is reflected back by objects in front of it, much like the line tracking sensors. Due to the inverse square law, the closer an object is, the larger amount of light is reflected back. By measuring the amount of light reflected, the presence of an object can be determined.

The avoidance sensor boards have a pinout of ground, power, and an output pin, and the module can run at 5V [1]. The output data from the module is binary and is used to maneuver around obstacles. In software, the two sensor inputs are simply read repeatedly while in a relevant mode. If any of the sensors register an obstacle, the motors stop, and wait a little while for everything to settle. Then, a canned motor movement and squeak is played depending on which one or if both of the sensors sense an obstacle. While a movement is being played, avoidance sensor data is temporarily ignored.

The modules offer very accurately what is desired for the prototype, however, they take up a lot of valuable space and they are difficult to adjust accurately.

We could have built avoidance sensors ourselves to make them smaller, but considering the time frame and how well the KY-032s work, we chose not to.

Power delivery

Squeak is powered using a three celled LiPo battery, which gives it a nominal voltage of 11.1 V. The battery has a rated capacity of 1500 mAh. The battery was selected to fit the requirements of the motors. Although the motors are rated for 12v, they run without problems at slightly lower voltages. Our decision then was between a regular 9v battery and a three celled LiPo battery. We did not consider AA or AAA batteries in series due to the limited available space, and having to put at least six batteries in series to meet the voltage requirements. We decided on using a LiPo battery for multiple reasons:

1. LiPo batteries have a significantly higher energy density compared to other battery types[10]. This allows us to not worry about the battery running out during testing or demonstration.
2. The voltage is higher compared to a 9V battery. This

means the motors will run faster and with a higher torque, which is desired.

3. LiPo batteries are rechargeable, while most 9V batteries are not.

The specific choice of battery was made simply because of availability, as it was the only easily accessible 3S LiPo battery. The current choice of battery is excessive, especially with the new motors which have a lower current draw. It could easily be replaced with a smaller 3S battery.

In line following mode, which has all aspects of the circuit running and the motor going at full speed, the whole circuit has been calculated to draw around 205 mA (see the appendix "Calculating current consumption"), and measured to draw 160 mA. With the current battery and measured consumption, this gives an on-time of at least 9 hours. With the expectation of a three hour running time, a 480 mAh battery would be sufficient.

With the current battery, the expected on-time can be calculated as $\frac{1500mAh}{160mA} = 9.4h$

To run the circuit continuously for three hours would require a battery with a capacity of $160mA * 3h = 480mAh$

The entire circuit is powered by the LiPo battery. Only the high side of the H-Bridge and the line following LEDs are driven directly by the 11.1 V source. All other components are driven at 5V, which are supplied by an L7805 [12] linear voltage regulator (LVR) connected to the 11.1 V rail. The choice and implementation of the LVR is described in detail in assignment 1.

3D Printed parts and assembly
All 3D printed parts have been printed using an FDM printer. A diagram of the exploded prototype can be seen in figure 9. The circled letters on the diagram indicate different elements corresponding to the elements with belonging letters on the following list:

Ⓐ 3D printed outer shell

Ⓑ 3S LiPo 11.1 V battery

Ⓒ 200 RPM N20 Micro Metal Gear motor

Ⓓ 3D printed wheel with 3D printed tire

Ⓔ PCBs with added components

Ⓕ KY-032 IR avoidance sensor

Ⓖ 3D printed platform for avoidance sensor

Ⓗ 3D printed base

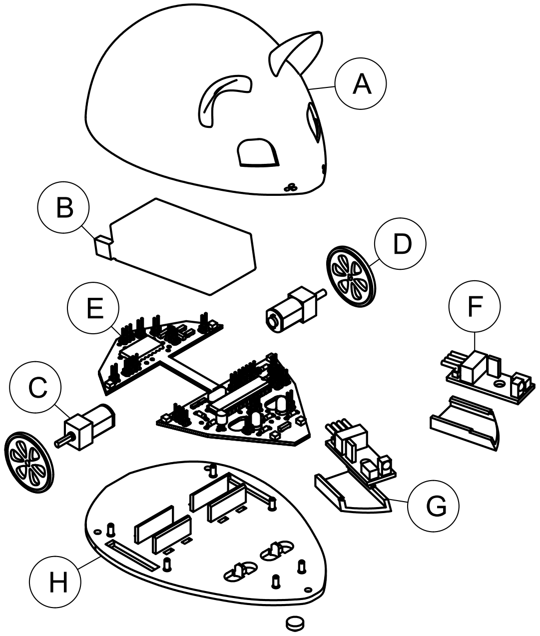


Figure 9: Exploded 3D model.

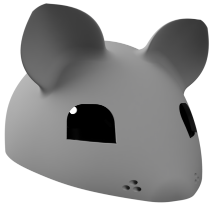


Figure 10: 3D model of the outer shell.

Shell
The shell is in terms of outer look the defining shape of Squeak. It also features screw holes for assembly onto the base. During assembly, M3 nuts are glued on top of the screw holes, and the base is screwed onto the shell using bolts. A screw hole with a nut can been seen in the appendix "Inner prototype gallery" in the lower right corner.

To give the shape more mouse-like characteristics, big ears have been modeled and printed on top of the shell. Additionally, small holes for brush hairs have been made to imitate whiskers. The 3D model of the shell can be seen in figure 10

The final shell for Squeak is printed using PLA filament as PLA is less time consuming to post process compared to PETG, which is what the shell was printed in before. We initially printed the shell using PETG filament as it is less brittle than PLA, and we wished for the shell to be able to be squished with force without breaking.

Base
The base of Squeak is a flat egg shaped 3D printed platform with different sized and shaped holes, pins and walls.

The base features are as follows:

Ⓐ Small pins for mounting the PCB.

Ⓑ Holes for the line tracking components. A small divider separates the LDR from the LED to protect the LDR from sensing light directly from the LED.

Ⓒ Walls for keeping the motors in place.

Ⓓ Holes for zip-ties to hold the motors tight against the base.

Ⓔ M3 screwholes to assemble with shell from underneath.

Ⓕ 0.5 mm insert for front contact point (placed under the base).

The annotated diagram of the 3D model for the base is shown in figure 11.

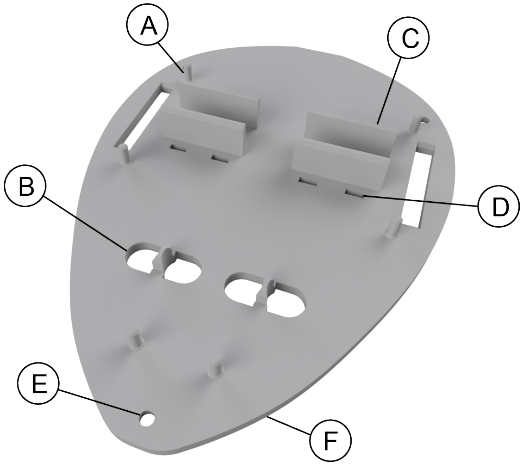


Figure 11: 3D model of the base.

The final base is printed in PLA as post processing prints in PLA is faster than post processing PETG prints.

Wheels and tires
To obtain a higher, rubberlike friction, NinjaFlex (TPU) filament has been used for the tires. The wheels have a slight indent around the side for the rubbery tire to slide onto. This also holds the tires better in place. The 3D model

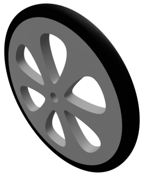


Figure 12: 3D model of wheel and tire.

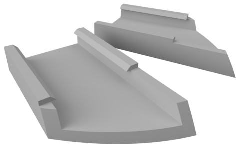


Figure 13: 3D model of the platforms for the avoidance sensors.

of the wheel and tire can be seen in figure 12. The wheels are printed in PETG because we decided not to post process them, as they are barely visible when looking at Squeak.

Eye platforms
To keep the avoidance sensors in place in the eye sockets of Squeak, we designed and printed small slide-fit platforms, which are inserted into and glued to notches on the inner-side of the shell. The 3D modelled platforms can be seen in figure 13. With these platforms, it also becomes easier to adjust the sensitivity on the potentiometer knob without having to put the sensor in the exact place every time.

The platforms are printed separately from the shell and glued together later, as the model would be significantly more difficult to print. The platforms would require support structures to be printed due to the large overhang, and at the same time either block the generation of support for the ceiling of the shell, or make cleaning the print significantly more difficult with generating support between the platforms and ceiling.

The final platforms are printed in PLA for the same reasons as previously mentioned.

Final prints and post processing
The final 3d printed parts were post processed by sanding them down until the layers were mostly invisible, priming them, and finally spraypainting them grey. The post processed shell can be seen in figure 14, and the base can be seen in figure 15.

Problems
Throughout the development process of the prototype, various technical problems were encountered. Some of these are described in the following subsections.

Figure 14: The finished, post processed shell.

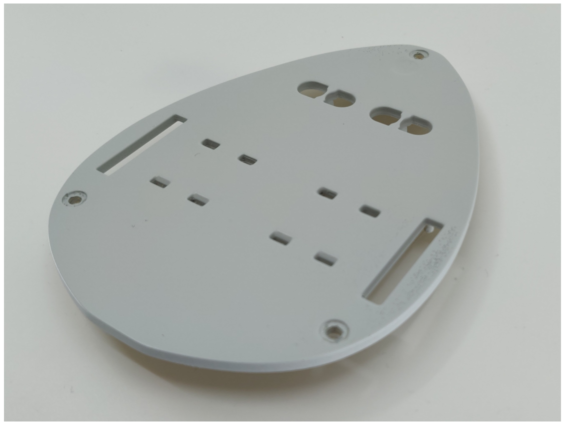


Figure 15: The finished, post processed base.

As the wheels have an outer circumference of ~88mm, the movement speed using the RF-370CA motors would be

$$88\frac{mm}{r} * 4840\frac{r}{min} =$$

$$425920\frac{mm}{min} = 7.10\frac{m}{s}$$

840mm is the long side of an A1 paper. We calculated the desired RPM as

$$\frac{840mm}{4s} * \frac{1r}{88mm} =$$

$$2.386\frac{r}{s} = 143\frac{r}{min}$$

Finding the right motor
When testing the running prototype, it became clear that the RF-370CA motors first used were not optimal for the speed and torque we needed for Squeak.

According to the datasheet[7], the motor is most efficient when running at 4840 RPM. This would result in a movement speed of more than 7m/s, which would be absolutely unmanageable. Furthermore, the torque of the motor is not particularly strong as it is not geared. This is compounded by having to run the motor at relatively low duty cycles, to manage the speed of the motor. The result was that the motors would often not provide sufficient torque to overcome the initial internal static friction of the motor bearings. This meant that Squeak would not move at first, and if given a push, it would suddenly move, and move too fast for the sensors to react.

To find a more fitting motor, we first calculated a desired motor RPM, from timing how long we would expect it to cross a set distance. We concluded that it should be able to cross 840mm in 4 seconds. From this we calculated a desired RPM of 143. From this we arrived at the formerly described motors with a gearing much more fitting. We decided to go with a slightly higher RPM, as we could always slow down the motors. The size is also smaller, while still running at 12V. The only downside to this change is the slightly noticeable noise from the gearing.

Scaling 3D models and PCBs
We went through a lot of work to make Squeak as small as possible. Firstly this meant making the PCBs take up as little space as possible. Then we had to model the shell, base and platforms around the PCB and battery, sensors and motors. To do this in a meaningful way, we made 3D mock-ups of all our components to assemble them into a base and shell in Fusion360.

We went through different iterations for both the shell and the base. The shell has been tested to both fit a 9V battery and a 3S LiPo battery. The shell for the 9V battery is slightly smaller than the version for the 3S LiPo battery, as the 9V battery is smaller.

We later added ears and more holes to the larger shell. So the shell has been through three iterations, and we have four different printed shells, as the first three were printed in PETG but we needed a PLA one for the final assembly. This means we have produced one small PETG shell, two big PETG shells where one has more details and lastly a big detailed shell printed in PLA. The first three shells are shown in figure 16.

The base has gone through four major iterations. The first iteration had a base with a guessed size (approximated based on how big the motors are) and many mounting holes because we assumed the PCB and component integration would look different. The holes for the motors are the correct size for the RF-370CA motors.

We found out that the base did not need as many mounting holes, so we made a new iteration with less holes. But the new iteration had no mounting possibilities, which meant the PCBs had to be lying directly on the base. A new base with poles that fit with the holes in the PCBs was then made. The motor holes in the third iteration are for the RF-370CA motors, so when we switched motors to the much smaller 200 RPM N20 Micro Metal Gear motors, we had to make a new base. On the newest base we also added walls to the keep motors in place, and made a small separation for the holes for the LED and LDR. So we have designed four different bases. The base used for the final prototype is identical to the fourth base iteration, but is printed in PLA instead of PETG which all the other iterations are. The bases printed in PETG are shown in figure 17.

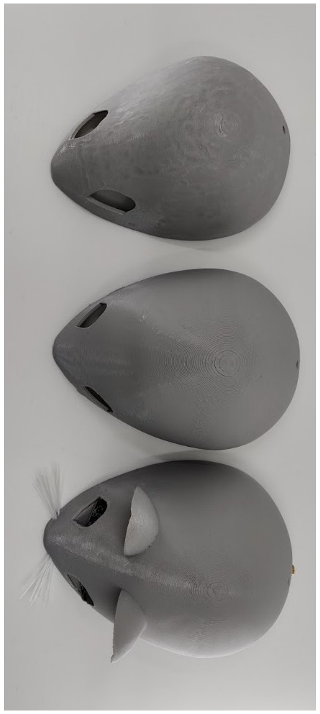


Figure 16: Printed shells not in use. The first iteration is at the top and the newer iterations downwards. The first shell looks a bit smudgy, as we tested post processing techniques on it.

As calculated in the appendix "Calculating current consumption", the LVR provides about $70.2mA$ and drops around $11.1V - 5V = 6.1V$, which results in a power dissipation of $0.0702A * 6.1V = 0.43W$.

Limitations and Future Work

In our work with designing, building and programming Squeak, we faced different issues and design challenges. Some of the found issues can be solved or improved upon in potential future work. The issues are framed from the perspective of being able to put Squeak into production as a commercial product.

Battery situation

The battery is currently completely enclosed in the plastic shell and requires complete disassembly to change. There is currently no way to recharge the battery while it is installed. Furthermore, the battery is currently not mounted rigidly to either the shell or base. This is a major limitation of the prototype, and must be rectified before beginning production. One solution would be to change out the current battery for a 9v battery or similar, and create a battery hatch in the model, so it can be replaced. This would require changing the motors for ones rated at a lower voltage. Finding a location for the battery hatch is also a challenge. There is no major space on the bottom of the model which is not covered by PCB or motors, and placing the hatch on the top would be an aesthetically bad choice. An alternative solution could be to add a charging circuit inside Squeak. This is discussed later.

There is currently no way to tell how much charge is left in the battery. This would be useful to know for when to replace the battery, or plug it in. The voltage of the battery could be sensed using a voltage divider across the battery. To communicate low battery, squeaks with a drawn out high to low frequency sweep could occasionally be made.

An additional concern with the power circuitry is that the LVR has a significant power dissipation, and therefore gets hot. To minimize this, a step-down circuit could be used.

Challenges and possibilities of the mode-changing tails

While the idea of using the tail for changing modes, it has both some challenges and untapped potential. During the development of the prototype, we suddenly discovered that one of the tails had gone missing. We found it again a while later on a table across the room. This clearly illustrated how kids could easily loose the tails, during or between play. As the tails are essential for the functionality of the mouse, this is a significant concern. This would have to be reconciled somehow before production.

We also recognise untapped potential in the tail. Currently when no tail is plugged in, the underneath LEDs and motors are turned off, making it inactive. All other parts of the circuitry remain on however. With some additional circuitry, unplugging the tail could be used to completely turn off major parts of the circuitry and put the MCU in a low power state, allowing the Squeak to not have any additional external switches.

As previously mentioned, there is currently no easy way to change or charge the battery. This could be reconciled by also using the tail socket as a charging connector. This would require additional circuitry with some intelligent switching, but would be an elegant solution to the battery problem. Combining these two possibilities would not be the best idea, as an irreplaceable battery with a tiny but not insignificant constant drain could have problems for long term storage.

Moving out of prototype territory

Currently the sensors use experimentally determined threshold values which are hard-coded into the software. This may be a problem for mass production, as different light settings and variations in sensors could result in unwanted or unexpected behaviour. Various methods can be used to compensate for this, like factory calibration, or

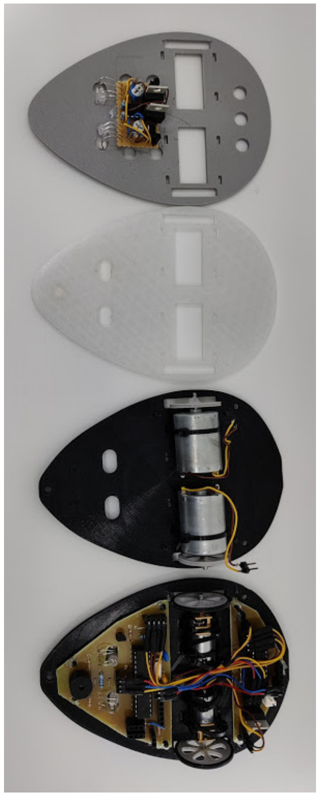


Figure 17: The printed bases. The first iteration is at the top, and then the newer iterations downwards.

automatic adjustment. Selecting the optimal solution would require further experimentation.

For mass production, the mouse could also be made smaller. Currently the size of the mouse is constrained by the size of its components:

- The battery as mentioned earlier is bigger than it needs to be. A 500 mA battery would require significantly less space.
- The avoidance sensor modules are very big compared to the other parts. By creating our own sensors, possibly using the same technique as the lift sensing, and mounting the components on the circuit board their size could be reduced. This would also allow for some smaller and better-fitting eyes, which could make the mouse look less like a prototype.
- In general, the size of the circuit boards could be minimized by using SMD components instead of through-hole components, and by using multi-layer boards.

By making these changes, the mouse could be scaled down to a smaller size.

The finish of the mouse is clearly still in the prototype state. Instead of producing the plastic part for the mouse using FDM and then post processing it, another manufacturing process could be used. If the Squeak was expected to be produced in very large volumes, injection moulding could be used. This would require a redesign of the assembly, as injection moulding requires the parts to be able to pass cleanly out of the mould. The current screw holes in the top part makes that difficult. One approach to this would be to extrude the holes into the shell, turning them into mounting bosses, and use self-tapping screws during assembly. Another approach would be to redesign the parts to be assembled using snap-fits.

Conclusion

Squeak is an interactive, moving toy mouse with different playmodes to engage different kinds of play. A prototype of the toy was created using FDM 3D printing and etched circuit boards. The prototype was developed in an iterative process through which the 3D models were refined, and choices of components like sensors, batteries and motors were discussed and decided. The process has been recorded in this paper. The result of this process is a fully functional prototype of a high fidelity. Finally, limitations of the prototype and areas of potential future work were identified with regards to making the project ready for commercial production.

Video demonstration

A video demonstration of the functioning prototype can be seen at the following link: https://youtu.be/37ZYqtFI65o.

Code

All the code for the Squeak project is available at the following git repo: https://gitlab.au.dk/au649483/squeak. Additionally a .zip of the code is provided alongside this document.

Division of labour

Each team member has contributed to an extend to every part of the work with Squeak. We have all been coding, 3D modelling, soldering, writing on the report, etc. But some team members have had the main responsibility for different areas, and has thus been in charge of making sure the tasks for their specific area have been completed on time.

Sarah has mainly been in charge of the code and assembly of the prototype. Jakob has mainly been in charge of the 3D modelling, 3D printing and physical design. Ida has mainly been in charge of designing and making PCBs and soldering components and wires.

REFERENCES

[1] ArduinoModules. 2020. KY-032 Infrared Obstacle Avoidance Sensor Module. (July 2020). `https://arduinomodules.info/ky-032-infrared-obstacle-avoidance-sensor-module/`

[2] J. Butcher and Warren Eaton. 1989. Gross and fine motor proficiency in preschoolers: Relationships with free play behavior and activity level. *Journal of Human Movement Studies* 16 (Jan. 1989), 27–36.

[3] Tom Djajadiningrat, Stephan Wensveen, Joep Frens, and Kees Overbeeke. 2004. Tangible products: redressing the balance between appearance and action. *Personal and Ubiquitous Computing* 8, 5 (Sept. 2004). DOI: `http://dx.doi.org/10.1007/s00779-004-0293-8`

[4] Everlight 2007. *5 mm Round White LED (T-1 3/4)*. Everlight. `https://ardustore.dk/error/Manuel%20-%205mm%20LED.pdf`

[5] Ylva Fernaeus, Maria Håkansson, Mattias Jacobsson, and Sara Ljungblad. 2010. How do you play with a robotic toy animal? a long-term study of Pleo. In *Proceedings of the 9th International Conference on Interaction Design and Children (IDC '10)*. Association for Computing Machinery, New York, NY, USA, 39–48. DOI: `http://dx.doi.org/10.1145/1810543.1810549`

[6] Qiushi Lin, Jianfei Luo, Zhongcheng Wu, Fei Shen, and Zengwu Sun. 2015. Characterization of fine motor development: Dynamic analysis of children's drawing movements. *Human Movement Science* 40 (April 2015), 163–175. DOI: `http://dx.doi.org/10.1016/j.humov.2014.12.010`

[7] Mabuchi Motor Company 2003. *RF-370CA*. Mabuchi Motor Company. `https://www.jameco.com/Jameco/Products/ProdDS/238473.PDF`

[8] Microchip 2020. *ATmega48A/PA/88A/PA/168A/PA/328/P*. Microchip. `http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf`

[9] Pololu 2019. *Micro metal gearmotors*. Pololu. `https://www.pololu.com/file/0J1487/pololu-micro-metal-gearmotors-rev-4-2.pdf` Rev 4-2.

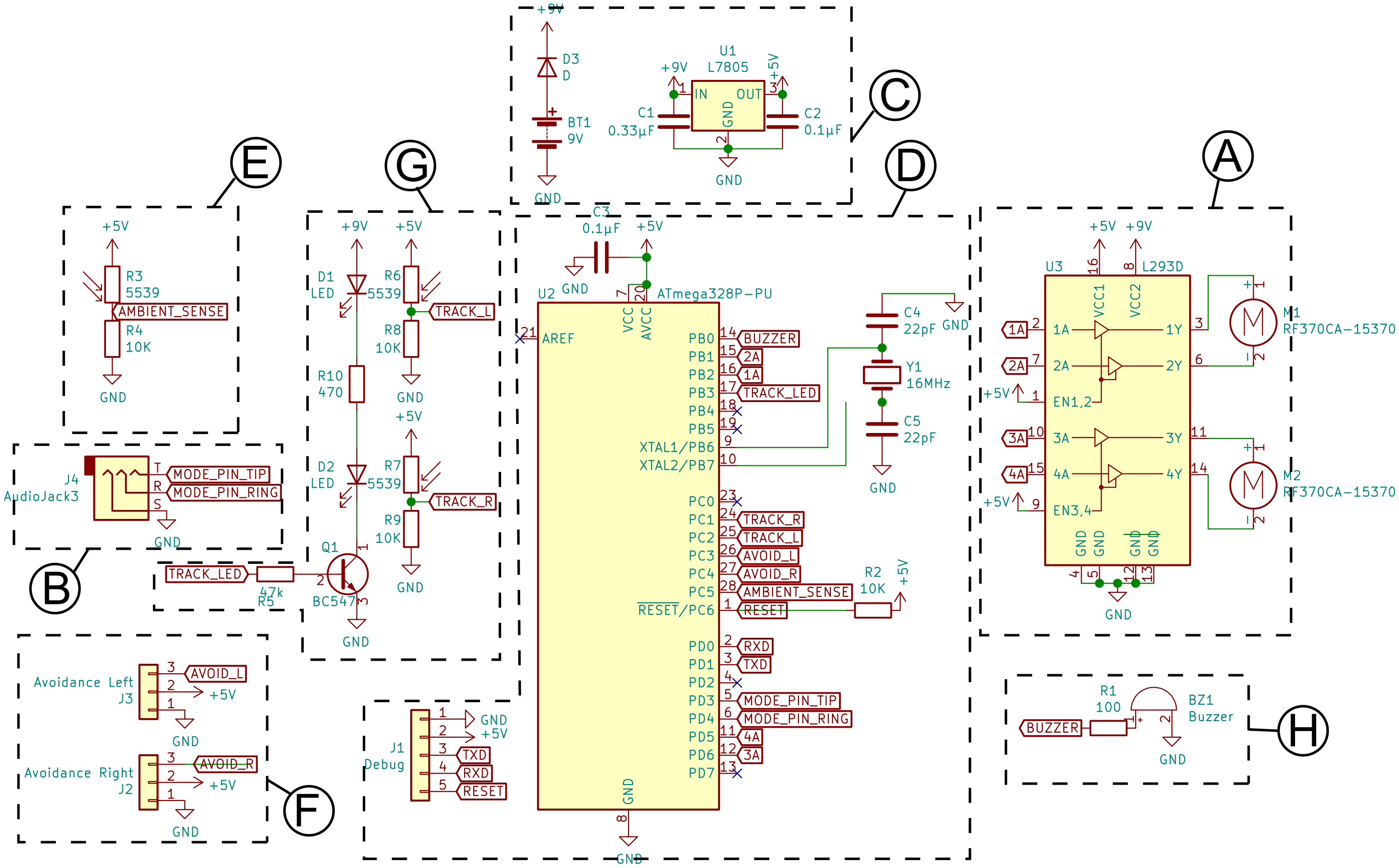
[10] Bruno Scrosati, K. M. Abraham, Walter A. van Schalkwijk, and Jusef Hassoun. 2013. *Lithium Batteries: Advanced Technologies and Applications*. Wiley & Sons, Incorporated, John.

[11] SENBA OPTICAL & ELECTRONIC CO.,LTD 2019. *GL55 Photoresistor*. SENBA OPTICAL & ELECTRONIC CO.,LTD. `https://datasheetspdf.com/pdf-file/756929/SENBA/GL5539/1`

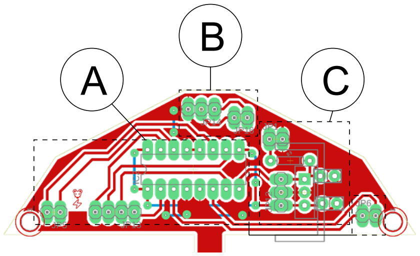
[12] STMicroelectronics 2018. *L78 Positive voltage regulator ICs*. STMicroelectronics. `https://www.mouser.dk/datasheet/2/389/l78-1849632.pdf` Rev 36.

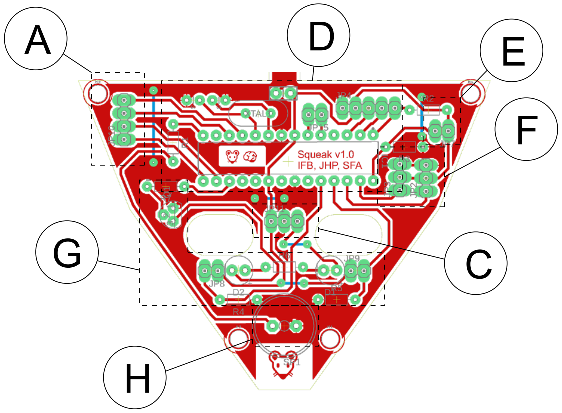
[13] Texas instruments 2016. *L293x Quadruple Half-H Drivers*. Texas instruments. `https://www.ti.com/lit/ds/symlink/l293d.pdf`

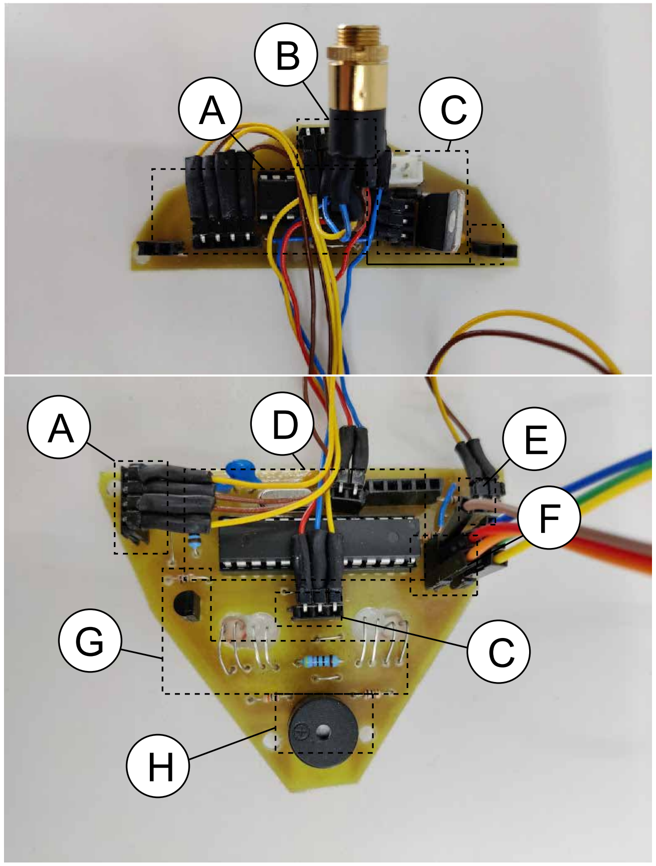
Annotated circuit diagram



Annotated PCB







Price table for prototype

Component	Amount	Individual cost	Shipping cost	Collective cost
Ceramic capacitors	5	0,06	7,7	8
LED white	2	0,04	5,18	5,26
3C LiPo 11.1 V battery	1	84,24	0	84,24
Atmega328	1	8,01	7,77	15,78
L293D	1	0,91	9,06	9,97
KY-032	2	4,19	10,36	18,74
200 RPM N20 Motors	2	12,26	7,77	32,29
LDR 5539	3	0,228	6,23	6,914
TO92 Transistor	1	0,57	7,77	8,34
Resistors	7	0,05	6,41	6,76
Buzzer	1	0,8	8,63	9,43
LVR L7805	1	0,3	8,14	8,44
16MHz Crystal	1	0,15	13,07	13,22
Filament (kg)	0,114	173	0	19,722
Pinheaders Male (40 PCS)	1	5,24	7,65	12,89
Pinheaders Female (40 PCS)	1	11,71	9,06	20,77
Other materials estimate (Solder, wire, heat shrink tubing, etc.)	1	20	0	20
SUM				**kr 300,77**

Calculating current consumption

As the motors will be running in a pretty low-torque environment, we will assume the motors run at close to a no-load condition. This puts the motors current consumption at $60mA$ each[9].

The L293D motor controller has both a significant logic supply current and output supply current. As the datasheet only provides information for static situations (all outputs on, all outputs off), we estimate an average consumption of about $10mA$ for the output supply, and $20mA$ for the logic supply[13].

At $5V$ and $12mHz$ oscillation, the ATMEGA328P[8] consumes about $7.5mA$. In addition, two internal and one external $10k\Omega$ pullup-resistors are used. Their current consumption is $0.5mA$ each.

According to the datasheet, the two avoidance sensors have a current consumption of up to $20mA$ each [1].

The three voltage dividers using LDRs have a combined $5V$ to ground resistance of at least $12k\Omega$. This puts their current consumption at $0.4mA$ (or less) each.

The two line following LEDs have an average $5mA$ power consumption, as explained by the previous assignment. As the components are in series, the current consumption should not be multiplied by the number of components. The transistor for the LEDs are controlled through a $47k\Omega$ resistor. The current through this transistor is so minuscule, it is ignored.

The buzzer does not have a massive power consumption, and as it is used very infrequently, it is also ignored.

This puts the power consumption for the 5v supply at:

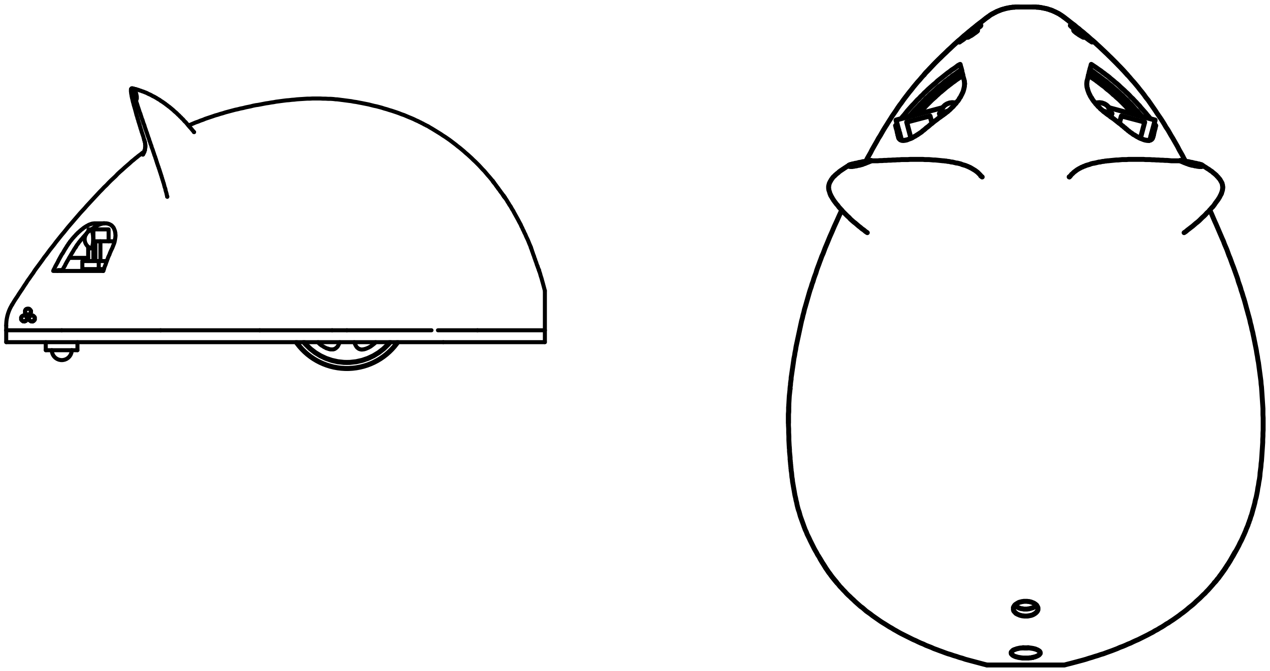
$$20mA+7.5mA+2*20mA+3*0.5mA+3*0.4mA = 70.2mA$$

In total, this puts the calculated current consumption at about:

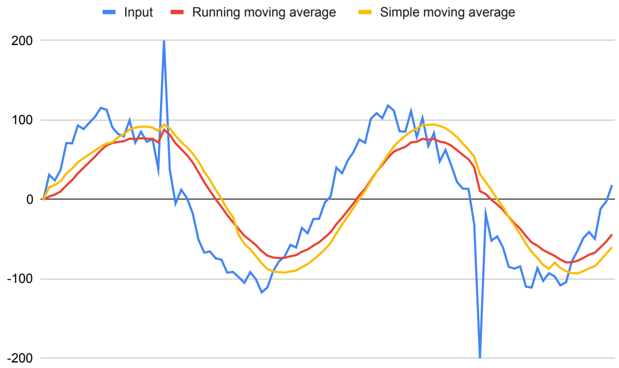
$$70.2mA + 2 * 60mA + 10mA + 5mA = 205.2mA$$

.

3D model drawings of Squeak from the side and top



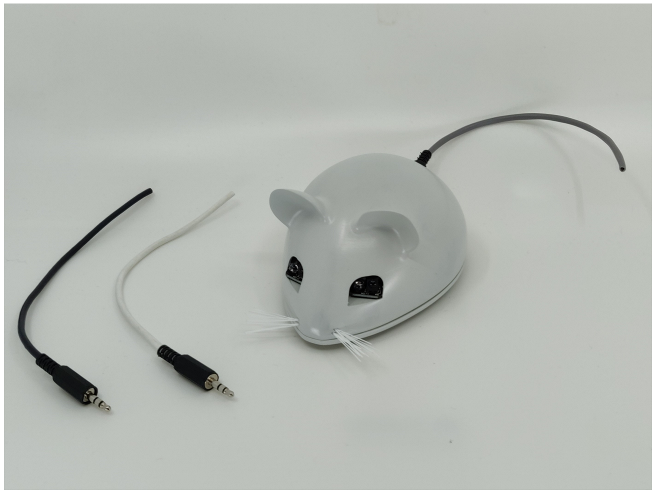
Comparative simulation of Running Moving Average and Simple Moving Average



The graph shows a sinus curve with added noise, and two large spikes to simulate an imperfect sensor signal. The blue line is the raw input. The red line is a running moving average, configured with a coefficient of 8. The yellow line shows a simple moving average, configured with a window size of 12. From the chart, we can see that the only major difference between the two methods are that the running moving average is not as good at following the peaks of the curve as the simple moving average. The running moving average however also seems to recover faster from the spikes in the data.

The used values and calculations for this chart are available at https://docs.google.com/spreadsheets/d/1O7_cTsW7JuG3KgfxRH0HxhHOGtW0-fpTdp25G_doUtI

Outer prototype gallery



Inner prototype gallery