Mathematical Keyboard Final Report

EE2 Project - Group 4

Imperial College London

Supervisor: Dr. Edward Stott

Koral Hassan (01096803)
Mariam Sarfraz (01069329)
Xavier Kearney (01044126)
Orion Mathews (01053855)
Antonio Enas (01070859)
Xu Wang (00932358)
Xiaoyu Ma (01049436)
Clive Toh Soon (01044264)

Date of submission: March 8, 2017

Contents

1 Abstract

2 Introduction

3 Project Management

3.1 Communication

3.1.1 Slack

The team employed various strategies to maximize group efficiency. To make communication among group members effective, the team decided to use Slack, a cloud-based, online application which is available across a range of devices and platforms. Channels are the key feature of Slack, which allow the team to separate discussion of different topics. A variety of discrete channels were created, 12 in total.

Four channels were made for project management purposes:

- General: for team-wide communication and announcements.
- Meetings: to decide on when and where the weekly meetings should take place.
- Minutes: where all the meeting minutes would be recorded. Meeting minutes were recorded to keep track of the weekly progress being made by the team. The work divided, decisions made and plans for the future were recorded.
- Random: where all non-work-related conversations would occur. These were kept out of more focused work-related-channels.

There were channels dedicated to the three distinct design areas of the mathematical keyboard.

- Hardware: to discuss the hardware implementation of the keyboard.
- Software: to discuss the software implementation of the keyboard.
- CAD: to discuss the industrial design of the keyboard.

For report writing, two channels were made:

- Report: where the report structure and content was discussed
- Citations: all useful cites were linked here

For the construction and design of the website, three channels were made:

- Website: to discuss the structure and content of the website.

- Video: it was decided that a video would be added in the website to effectively showcase the keyboard to potential customers. This channel discussed and shared the clips recorded and audio shortlisted for the video.
- Advert: to discuss how the product would be best advertised through the website.

In addition, Slack allowed the team to exchange one-on-one messages, to upload and share files and images and to notify particular team members on channels by typing their @username. Using Slack, the team could also set reminders for important meetings or to-do points. The team benefited from using Slacks search engine; documents and messages could directly be found.

3.1.2 Google Drive

All group documents were hosted on Google Drive to allow simultaneous collaboration and ensure everyone always had access to all the same information, including the full meeting minutes, project expenditure, and report sections.

3.1.3 Github

The software proportion of the project was managed using the online version control and collaboration system, Github. It enabled multiple members of the team to work on different parts of the software at different times, track and revert changes if necessary, and create branches to differentiate between code.

3.2 Team Roles

Tasks were distributed between group members under six focal divisions; Hardware, Software, Industrial Design, Website, Research and Administration. The responsibilities of each member are tabled below:

Role	Members	Role Description
Hardware Design	Xu Wang	Designing of the PCB. Construction and design of the other hardware required for the mathematical keyboard.
Software Design	Xavier Kearney, Koral Hassan	Analysis of methods used to output mathematical symbols. Implementation of the most efficient algorithm and programming language required for the mathematical keyboard. Write codes to emulate typing mathematical symbols within various software environments (e.g. Microsoft Word, LaTeX)/
Industrial Design	Antonio Enas, Clive Wong	CAD design of keycaps, case, switch holders and top plate. Exploring various manufacturing technologies and available facilities.
Website	Xavier Kearney, Clive Wong	Construction and design of the website. Recording and editing of the advertising video.
Research	Mariam Sarfraz, Orion Mathews, Xiaoyu Ma	Research of the distinct design areas of the mathematical keyboard.
Admin	Entire Team	Set the times and book the venues for the teams meetings. Collaborate with other divisions to get updates on progress and ensure the project is on schedule. Compile the report. Grammar check and submit the final report.

3.3 Gantt Chart

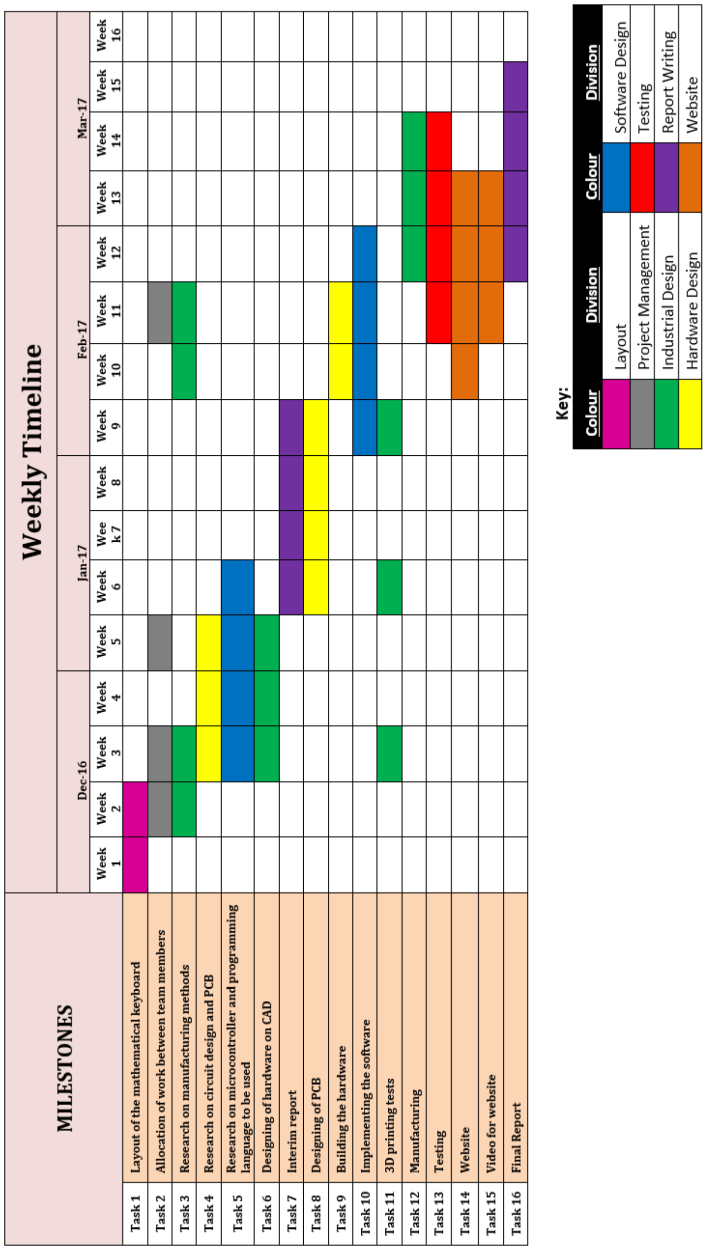


Figure 1: Project Gantt Chart

4 Design Concept

4.1 Problem

In the modern world, digitisation of documents is not only commonplace, but necessary. Students and professionals throughout the world regularly choose to write notes and reports with electronic devices. However, the standard computer peripherals were not designed for creating documents that include complex mathematical symbols and formulae. This means people waste time and effort digitising things that would otherwise be trivial to write on paper.

LaTeX is often employed when producing a professional documents, but the user can only see regular text (similar to code) before the file compiles and so it becomes difficult to refine the presentation before the final compilation produces a PDF. Microsoft Word is another popular word-processing application which allows users to insert equations and special symbols. However, the insertion process is not optimised for speed with authors having to find specific symbols by eye amongst a large collection. This leads to a fragmented writing process and slowed workflow.

The external mathematical keyboard envisaged by the team has the goal of smoothing the document producing experience, making users faster and more efficient. This is achieved with a small keypad that gives users access to the most commonly used mathematical expressions that are not possible on a typical QWERTY keyboard.

4.2 Market Research

The aforementioned problems were identified through personal experience of the team and confirmed by the survey that the team conducted of various engineering and mathematics students at various universities (full results available in Appendix #). The team then constructed a SWOT(Strength, Weakness, Opportunity, Threat) analysis to quantify the product potential. Finally, the team employed the MoSCoW method to define the specification for the product and clarify the necessity of the different requirements.

4.2.1 Strengths

The μBoard equips the user with fast access to the most commonly used mathematical symbols. It provides high flexibility with support for both LaTeX and Microsoft Word. In addition, μBoard does not require any additional software to be installed on the user's computer; it works immediately after being plugged in and doesn't require external power. The mathematical symbols are completely customisable, catering to the user's need. Due to its compact size, μBoard is also very portable.

4.2.2 Weaknesses

The initial prototype was comparable in manufacturing cost to a regular, full-size keyboard. However, this can be reduced by the economy of scale if the keyboard were to be mass produced. Additionally, a finite number of keys cannot cover the potentially infinite number of mathematical symbols; it only has most commonly used mathematical symbols. However, each key can be customised by the dedicated user to counter this weakness.

4.2.3 Opportunity

Based on the survey conducted, available in Appendix #, 73% of respondents were willing to pay more than £15 for an external mathematical keyboard. Therefore, there is a significant demand for a product similar to what the team has proposed. Also, the one of the key target groups for the product is Engineering/Mathematics students at university, who are heavily involved in the use of mathematical symbols in their reports. If the price of the keyboard can be brought down to a reasonable price due to the economy of scale, μBoard could fit their needs and fill the gap in the market.

4.2.4 Threats

There are software based mathematical keyboards that could potentially pose strong competition to the μBoard, but currently do not offer the same convenience and speed of a physical device. There is no physical mathematical keyboard available on the market, unless the user wishes to pay the premium for a fully customisable keyboard (£100+). Also, the two main platforms μBoard operates in are Microsoft Word and LaTeX. For compatibility reasons, μBoard operates best for users with the latest version of Microsoft Word (relative to the time of release of μBoard), hence it might not be compatible for older and future versions of Word, without updates to the firmware. In addition, LaTeX is taught at many universities which could reduce the number of people without the specialised knowledge to type mathematical symbols.

4.3 Specification

The team employed the MoSCoW method[1] to categorise the various requirements for the specification as detailed below.

4.3.1 Must Have

1. A diverse selection of the most commonly used mathematical symbols.
2. Fast, intuitive access with clearly printed keycaps.
3. A simple user interface with essentially zero learning curve.
4. A low enough cost to be justifiable to the user.
5. Standard USB interfacing to maximise compatibility with numerous different devices and operating systems.
6. Self-powering via USB.

4.3.2 Should Have

1. Multiple modes to support Microsoft Word, LaTeX and other text editors with ASCII symbols.
2. Include symbols from a range of different mathematical fields (Statistics, Calculus etc.)
3. Be ergonomic and comfortable to use, with keys arranged for maximal speed.
4. Be robust and durable, whilst remaining attractive.

4.3.3 Could Have

1. Support extra modes for other word-processing environments (e.g. Open Office).
2. Additional software to allow easy customisation of the key functions without having to edit code.
3. Bluetooth connectivity to enable wireless mode.
4. Support other devices, such as tablets.

4.3.4 Won't Have but Would Like

1. Dynamically display the function of the key (with small screens on the key caps). This technology is used in the Halo Keyboard on the Lenovo Yoga Book[2].
2. Crowd-sourced symbol layouts that people could download and install on their keyboard.

4.4 Design Choices

Several decisions had to be made whilst building the prototype in order to achieve an optimised outcome; financial, technical and practical limitations also had to be taken into consideration. Justification for each of these decisions is outlined below.

4.4.1 Hardware Design

1. The team settled on building a 4 x 5 key matrix including two modifier keys, after discussing many layouts. After testing a number of alternatives with rough skecthes, this number of keys was found to be the best compromise between providing a wide range of keys and portability. Together with the two large modifier keys, the user can input up to 64 unique symbols. The keyboard easily fits into the front pocket of a small bag. In terms of ergonomics, the keyboard can be used upright or sideways

comfortably according to preference (although upright is expected). The height and width were chosen intentionally for this purpose. Additionally, the keyboard can be used with either hand. The double-size modifier keys have been placed in such a way that they are in easy reach in any preferred position.

2. The team decided to implement a mechanical keyboard (with discrete mechanical key switches) rather than a membrane keyboard (with the key switches printed on a plastic membrane). Membrane keyboards require a customised circuit membrane and a specific silica-gel key layer, which would be incredibly difficult and expensive to prototype. Mechanical keyboards can be produced in an entirely discrete manner with the purchase of individual key switches, assembling them onto one small 2-layer PCB (Printed Circuit Board) as desired.

3. Initially the plan was to have 2 LEDs on the board as indicators. However, after consultation with the group supervisor Dr. Ed Stott, it was decided that the interface between the LED (5V) and the microcontroller (3.3V) would be less trivial than originally expected and could potentially drive the circuit into instability (especially as the circuit powers on). The group therefore introduced a robust digital LCD to indicate the working modes, which can be controlled easily using the microcontroller SPI (Serial Peripheral Interface) and the built-in Arduino LiquidCrystal library[3]. This LCD has the added benefit of being more user friendly and able to convey more information than LEDs, although at a slightly higher cost.

4. The PCB that was ordered had no solder mask. Without such a solder mask, the soldering process demands more time and expertise. However, the cost and speed of delivery of a PCB with a solder mask made it impossible to justify. The budget and time restrictions placed upon the team far outweighted the limitations in work capacity.

4.4.2 Software Design

1. The Teensy brand of microcontroller produced by PJRC[4] was chosen as the microcontroller to be used for a number of reasons - its small form factor, high clock rate (72MHz), large number of I/O pins (34), USB connectivity, USB power, compatibility with Arduino libraries and reasonable cost (£15). Moreover, it made sense to use the Teensy so that the team could take advantage of the multitude of online resources dedicated to it.

2. It was decided that one modifier key would function as a caps lock due to the regular necessity for both upper and lower case greek letters in mathematics. The functionality is identical to that of a traditional keyboard for maximum ease of use. The second modifier key functions as a shift key, making the buttons switch to alternative symbols which are logically and intuitively related to the original symbol each respective button

represents. To use the shift key, the user holds it down, as with a traditional keyboard. These decisions served to make the keyboard versatile but simple.

3. To switch modes (e.g. Word, Latex), the user must hold down both modifier keys for approximately 0.5 seconds. This is a simple way to switch and easily learned. It takes long enough to not be done accidentally, and short enough for it not to be cumbersome.

4. When a symbol key is pressed, the keyboard stops scanning for inputs until that key is released. The user can hold the key down for as long as they want without repeatedly typing the character. However, when the key is released, the software can continue scanning at a very rapid pace.

4.4.3 Industrial Design

1. A new type of keycap was designed from scratch for a number of reasons. Firstly, it holds onto the switches more reliably than the models found online. Secondly, the keycap was designed to mimic a laptop style flat minimalist shape. This ensures that the keys are comfortable to press no matter which direction you're holding the keyboard in. Furthermore, the flat tops make the engraving of the symbols on top of the keycaps much simpler.

2. Wooden tops were glued onto the keycaps, engraved with the symbols the keys produce. Having the symbols displayed on keys make the keyboard more user friendly and familiar. Engraving the symbols is not only easier to implement than putting stickers on the keys, but also gives the product a more professional look and feel. The team did consider the fact that if in the future the keys were customised by the user, this method might have to be revised to better support changing the icons on the keys.

3. The designs for the back and front panel of the keyboard were cut out of wood instead of printed with plastic. The four screws used in the design no longer pierce through the front panel but instead only hold the back panel and the PCB in place. The front panel is glued on; it has no screws or nuts protruding from it and interrupting the visual aesthetic, or potentially the ergonomics. If the device needs any servicing in the future then removing the back panel and PCB will suffice. The keyboard has a very unique and appealing look due to these choices, as shown in the Final Functionality section.

5 Design Development

The following section details the steps that the team has taken in order to meet the requirements laid out in the specification. The development is split up into three main sections: Hardware, Software and Industrial Design. The Final

Functionality section details the full functionality of the device produced by the team.

5.1 Hardware

The initial ideas behind the hardware component of the design were covered in the Interim Report. The circuit itself contains 3 interconnected parts: a microcontroller to detect and process key-pressed signals; an LCD screen to indicate the current operating mode (e.g. Word, LaTeX) and provide a user interface; and a custom 4 x 5 key matrix network with mechanical key switches.

The team initially tested the hardware design principles with a breadboard and smaller, 2 x 2 array of switches. In theory, the microcontroller would drive each row high for a short period whilst it read each column to determine if a key had shorted the node and therefore been pressed. It was discovered that the key-matrix columns did not return to their low (grounded) state after the keys were released due to the absence of pull-down resistors, and instead were left at an indeterminate, floating voltage. The PCB and schematic were easily modified to include these resistors (10kΩ) as shown in Figure 2.

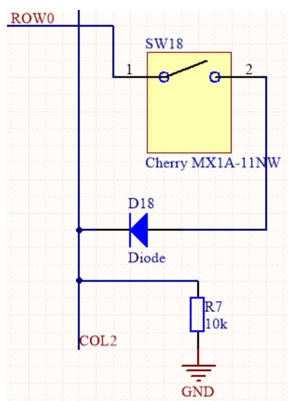


Figure 2: Circuit diagram for single section of the switch matrix

At this point, the team were confident in the PCB design and ordered the manufacture of a single prototype PCB from Newbury Electronics. The cost of this prototype was considerably higher than the team anticipated (£58.21), but could be reduced significantly in a mass-production scenario where the PCB could be produced abroad. Similar cost savings can be applied elsewhere in the project: the Teensy microcontroller is arguably overpowered for this application and, although it provides extensive customisability, could be replaced in a production model for a chip better cost-optimised such as an ATtiny series microcontroller.

The second challenge that the team faced was the interface between the LCD screen and the Teensy microcontroller, as detailed by Figure 3. The digital pins of the Teensy operate at a logic high level of 3.3V, whereas the LCD outputs

5V for a high signal. To utilise the fastest Arduino LCD library, the circuit required the data pins to be bidirectional so that data could be both read from and written to the LCD controller. In order to protect the microcontroller from over-voltage, research on the Teensy website suggested to place current limiting resistors between each of the data pins. These current-limiting resistors were originally set to 100kΩ but in testing were found to be too large compared to the pull-up resistors inside the microcontroller. Testing with a logic analyser showed that the logic levels were different either side of the resistors and this prevented the LCD from initialising correctly. Switching to 1k resistors quickly rectified this.

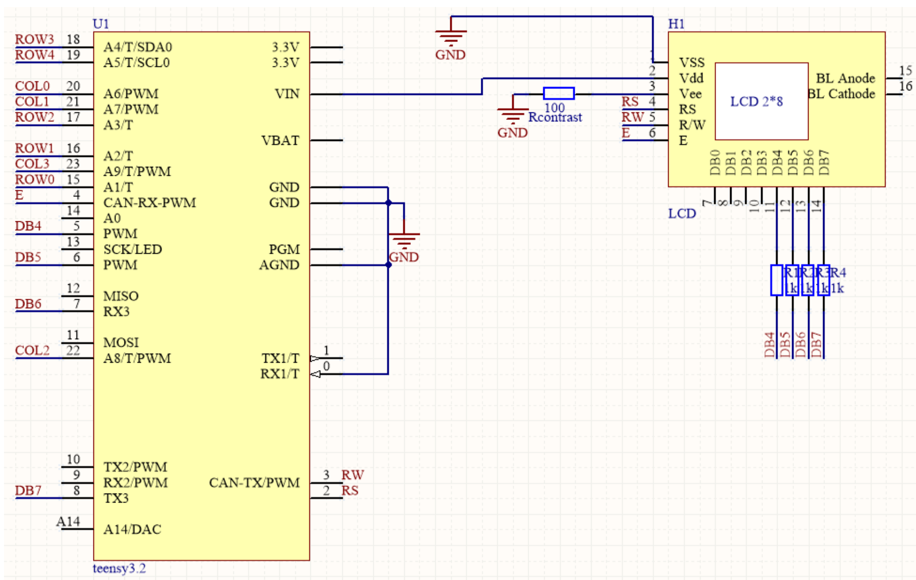


Figure 3: Circuit diagram of the Teensy - LCD interface

Finally, the team initially planned to set the contrast pin of the LCD to a high voltage to ensure maximum contrast. It was later discovered that the contrast was instead inversely proportional to the voltage applied and therefore the team had to modify the PCB to include a 100Ω resistor to ground. This value was suggested by one of the lab technicians as the standard value required by most LCDs and preliminary testing showed it provided a perfectly satisfactory contrast level. After all hardware issues had been resolved, the Hardware Design sub-team passed the board onto the Software Design team to complete the code and finalise the full functionality of the board. The schematic for the full circuit and the PCB design are both available in Appendix #.

5.2 Software

The software component of the device produced consists solely of the firmware installed on the controller module itself, whose purpose is to process input from the user (in the form of key presses) and transform it into the correct output which is then sent to the connected PC via USB. The specific microcontroller chosen by the team was the PJRC Teensy 3[4], as detailed in the Design Choices section.

A number of methods for processing the user input were considered by the software team, but it was decided that it would be most efficient to attempt to expand upon the existing open-source firmware framework available on Github, produced by the user JetpackTuxedo[5]. This framework provided the basic functions of scanning the key matrix for input and sending the corresponding keystroke. Unfortunately the framework did not support sending multiple keys with a single keypress and so the majority of the code has had to be rewritten by the team to incorporate this.

The team collaborated on the code using Github as detailed in the Communication section. The initial development focused entirely on the keyboard function of the device and only when that functionality was completed did the team write the code to interface with the LCD. To enable maximum compatibility, the keyboard functions in a number of user-selected modes: Microsoft Word, LaTeX and ASCII, as defined in the specification. Research was conducted to ascertain the required keypresses for each mode and these were tabulated in a multi-dimensional array that could be accessed programmatically. The specific program flow is shown in Figure 4.

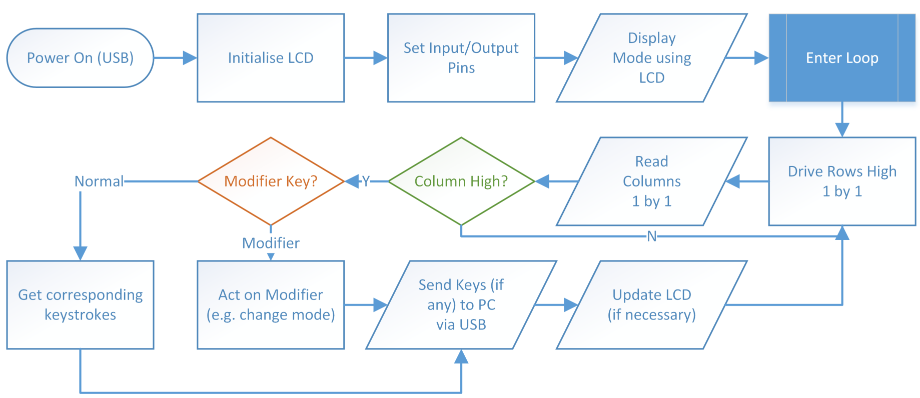


Figure 4: Simplified program flow diagram

Initial testing of the PCB required the hardware and software teams to work in tandem to diagnose various issues with the circuit. The software team devised a number of simpler test programs to isolate problems so that they could be analysed with an oscilloscope or a digital logic analyser. These specific tests enabled the team to quickly pinpoint the two main hardware issues with the board: incorrectly valued pull-down resistors for the key matrix, and incorrectly valued resistors for the interface between the Teensy and the LCD.

When programming the full firmware, a number of challenges were faced due to the various word processing environments. Firstly, the current state of the program running on the PC cannot be determined by the keyboard and so the keystrokes sent by the Teensy need to ensure that no unintended actions are performed if the user is in the wrong state (e.g. a menu). Secondly, the Arduino Keyboard library used to send the keystrokes is designed for the international (US) keyboard layout, which uses the same scan-code (key identifier) for backslash (\) as the UK keyboard layout uses for the hash key (#). This was resolved by editing the library to instead send the correct scan-code to the PC. Thirdly, with no input functions other than the keys themselves, the team had to decide how the user would intuitively change between the various modes; it was determined that holding both the SHIFT and CAPS buttons simultaneously (with no other keys pressed) for 0.5s was a comfortable, quick option that did not interfere with other keyboard functions. Finally, the delay between each program loop had to be optimised by trial and error, to determine the average length of a human key-press, so as not to inadvertently send multiple symbols with a single press.

The complete keyboard firmware code is available in Appendix # and also on the team's Github repository website[6]. Its exact functionality is detailed in the Final Functionality section. A user guide to operating the device is available at the team's website. The code has been written with an open-source philosophy in mind, and as such should be easily edited to modify its functionality or implement new features/layouts which are discussed in the Future Work section. The code is also valid for (almost) all Arduino compatible devices so could easily be reused for a larger project, or using a smaller (cheaper) microcontroller. Once the software was finalised, the board was passed to the Industrial Design team to fit the casing. Conveniently, the Teensy can be reset and reprogrammed purely using USB input and therefore sealing the case did not prevent further development.

5.3 Industrial Design

5.4 Final Functionality

The final product consists of a 4x5 keyboard matrix, including 16 symbol keys and 2 double-width function keys: CAPS LOCK and SHIFT. These keys can be combined in 3 ways to produce a possible 64 options for each user mode. A render of the final model is shown in Figure 5.

In reality, by utilising the 2 functions keys, it can provide up to 48 individual symbols using the modifier keys (SHIFT, CAPS). Figure 6 shows all the symbols that the board supports with the various modifiers.

The board also incorporates an LCD screen to indicate various things including the current working mode. The actual prototype PCB with all of the components soldered is shown in Figure 7. The realisation of this render is

Figure 5: 3D Render of the Complete Case

shown in Figure 8.

6 Future Work

In the Specification section the MoSCow method was used to discuss the team's objectives within the scope of the project.

All items in the 'Must' requirements have been completed, with a small exception. The prototype was more expensive than expected, and likely too expensive for a commercial product. However, the prototyping process is expected to be more expensive than mass production as it includes all testing, error corrections, and modifications. To mass produce the device would likely result in a huge cost saving (greater than 50%). For mass production, it may be financially viable to create a membrane keyboard instead of using mechanical switches and a PCB. Additionally, the Teensy was the perfect device for experimenting with the initial software but in the future could be replaced with a more cost effective chip with less (but still enough) functionality.

The requirements in the 'Should' section have also been successfully implemented. With future feedback from the product's users, the symbol layout could be rearranged and some keys even swapped out for other, potentially more useful ones. While the keyboard does satisfy the requirements in robustness, durability and aesthetics, it could be improved upon with more advanced manufacturing techniques. In particular, a finish on the wooden surfaces of the keyboard could be beneficial to durability.

There are several additional features the team would like to implement in the future. The board firmware could include even more modes fulfilling different needs. For example, a mode for Open Office would be useful by many users. The ultimate solution would be to allow users to choose their own custom layouts.

A	B	C	D
SYMBOL	SHIFT	CAPS	SHIFT_CAPS
d/d	∂ / ∂	-	-
∫	∫ with limits	-	-
√	n√	-	-
^2	^n	-	-
log_n	lim_m->n	-	-
σ	-	∑	∑ with limits
π	∠	Π	-
∞	Ø	-	-
±	∓	-	-
≃	≠	≅	-
≥	≤	-	-
∀	∃	-	-
θ	ω	Θ	Ω
δ	ϕ	Δ	Φ
∪	∩	⋃	⋂
μ	λ	M	Λ

Figure 6: All possible symbols with the standard keyboard layout

This could be achieved with a desktop app which has the ability to directly program the keyboard via USB. This elevates the capabilities of the product from helping mathematicians to being used in any and every field. Some fields that the team identified would benefit include Programming, Graphic Design, Animation, Composing and Gaming. The target market would be anyone whose repeated daily tasks are not catered for properly with traditional computer peripherals. Another possible future feature would be the introduction of wireless, Bluetooth connectivity. This would be relatively simple since there is a plethora of precedents and it is a highly standardised industry. Support for other devices such as tablets is definitely within the realm of possibility because this too has been done with keyboards in the past. Spreading to other devices would not drive the product cost much higher, but would help justify its price tag.

There also some features, which, although obviously advantageous, could only be made part of a significantly longer term plan. Some method to change the symbols printed on the keycaps dynamically, would arguably be the biggest improvement that can be made to the current device. This would require extensive research and development. The creation of this new keycap is possibly a bigger engineering challenge than the original board itself. One disadvantage of such a function would be that the keyboard would be sold at a higher price range, targeting a different, more niche market. Another impactful improvement would be the creation of an online library allowing users to share their custom layouts. Such crowd-sourcing and sharing of ideas has been massively successful with other software businesses in the past. It has been proven to be a very powerful tool, with lots of potential from little investment.

Figure 7: PCB with components soldered

7 Conclusion

Figure 8: Photo of the complete product prototype in its case

8 References

[1] Wikipedia. (2017). Moscow method for prioritisation, [Online]. Available: `https://en.wikipedia.org/wiki/MoSCoW_method`.

[2] Lenovo. (2016). Lenovo yoga book laptop, [Online]. Available: `http://shop.lenovo.com/us/en/tablets/lenovo/yoga-book/yoga-book-android/`.

[3] PJRC. (2017). Liquidcrystal teensy library, [Online]. Available: `https://www.pjrc.com/teensy/td_libs_LiquidCrystal.html`.

[4] ——, (2017). Teensy pjrc website, [Online]. Available: `https://www.pjrc.com/teensy/`.

[5] JetpackTuxedo. (2017). Jetpacktuxedo firmware on github, [Online]. Available: `https://github.com/jetpacktuxedo/keyboard`.

[6] G. Team. (2017). Controller github repository, [Online]. Available: `https://github.com/XavKearney/keyboard`.

9 Appendix

A PCB Design

Figure 9: 3D Render of the PCB

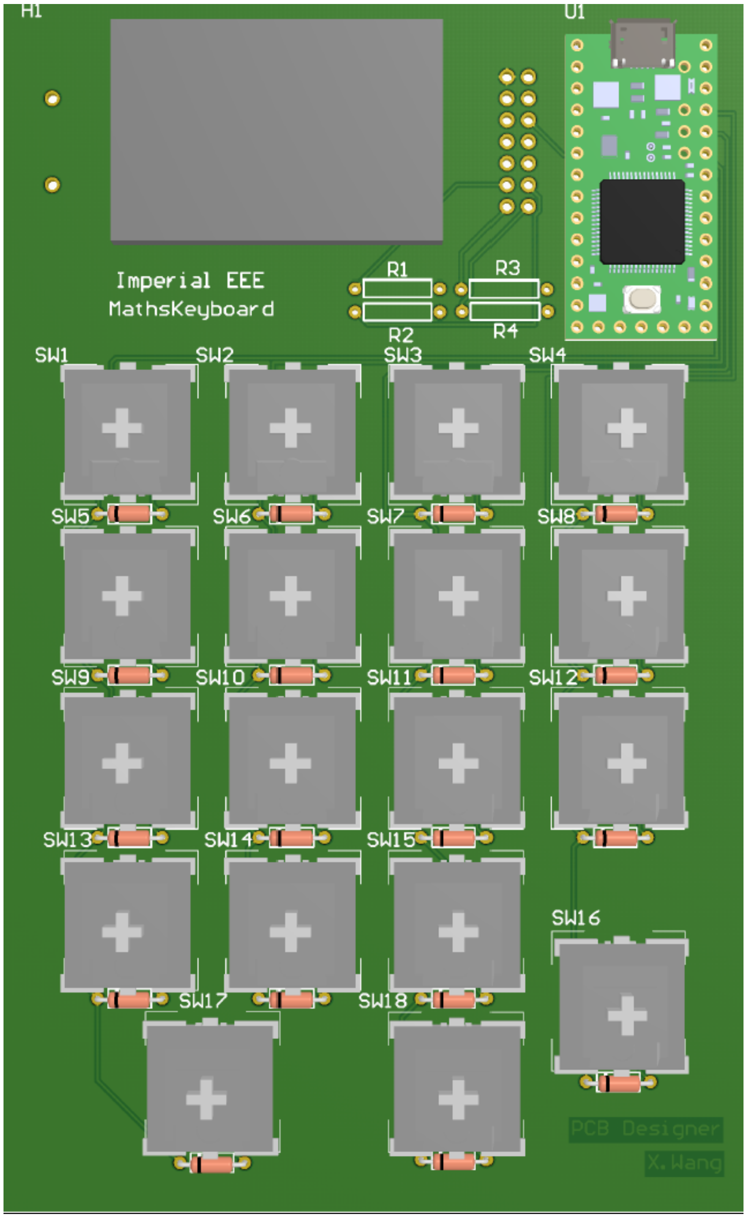


Figure 10: Technical 2D Render of the PCB Layers

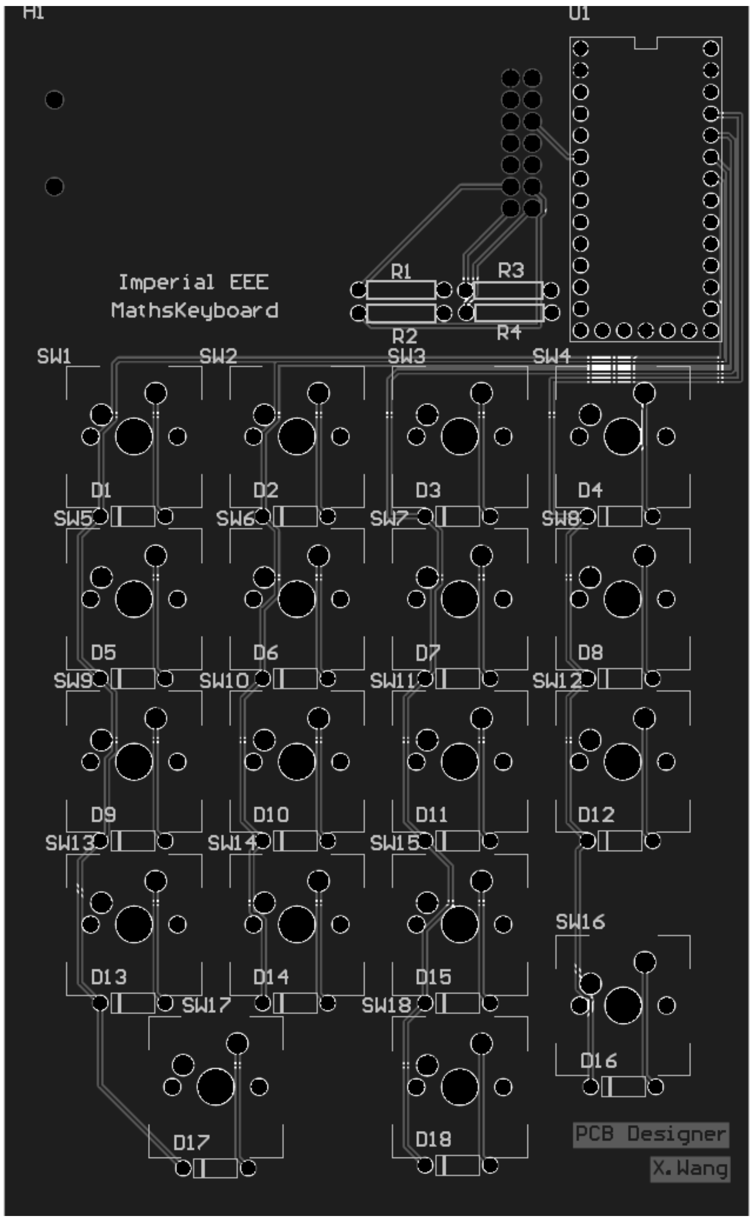
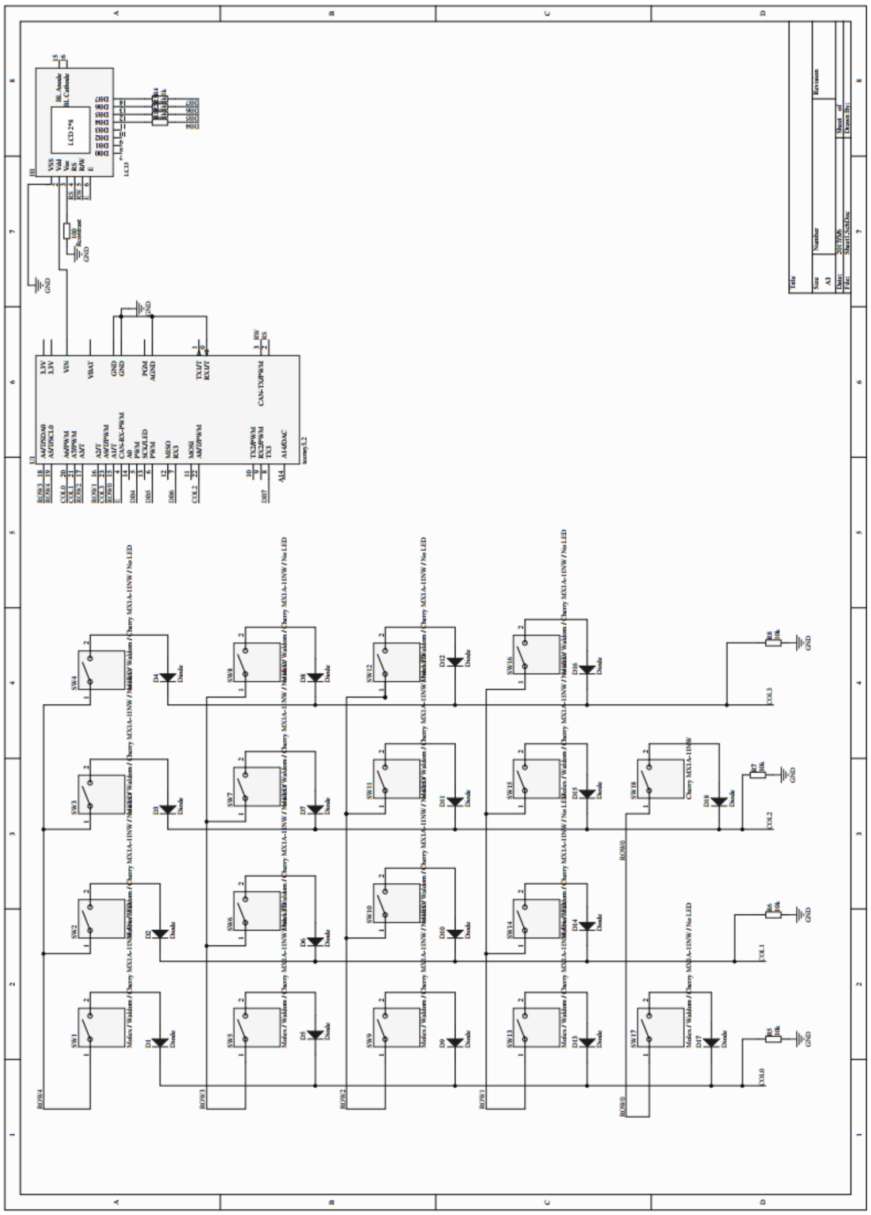


Figure 11: PCB Schematic



B CAD Renders

Figure 12: 3D Render of the Case

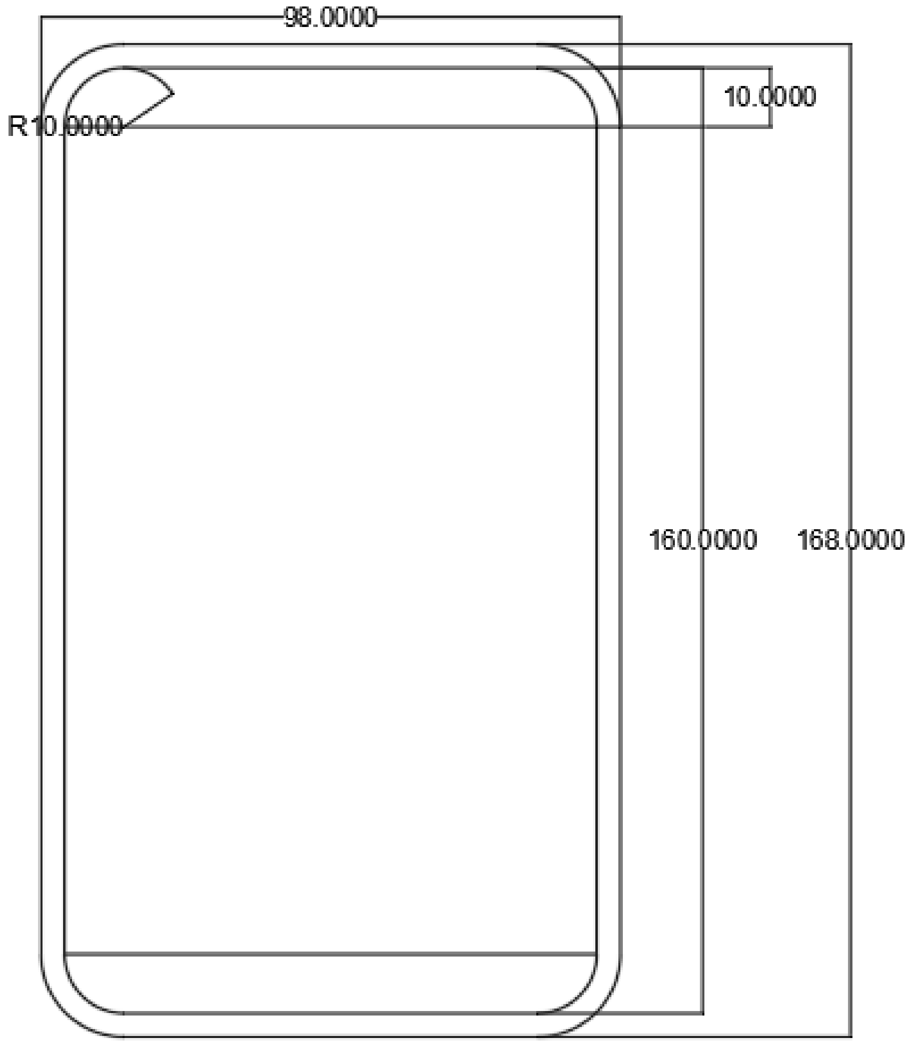


Figure 13: Dimensions for Custom Switch Holder Plate

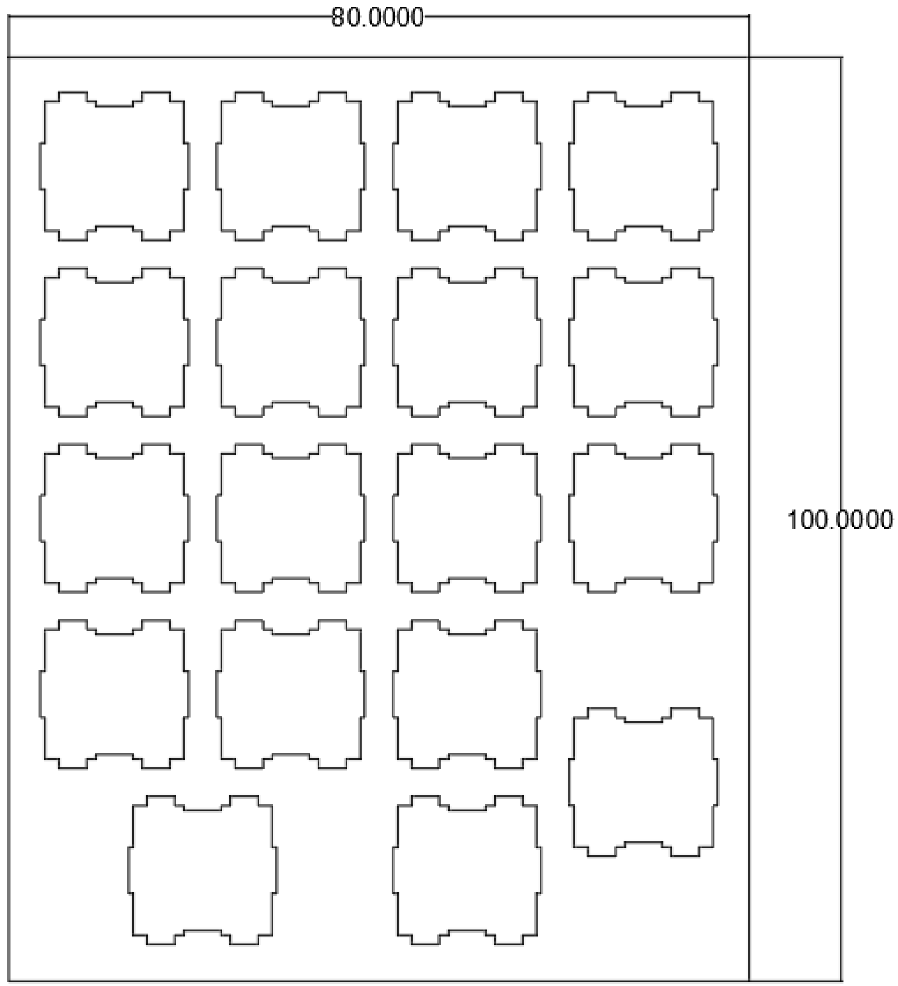


Figure 14: Full Dimensions for the Top Plate

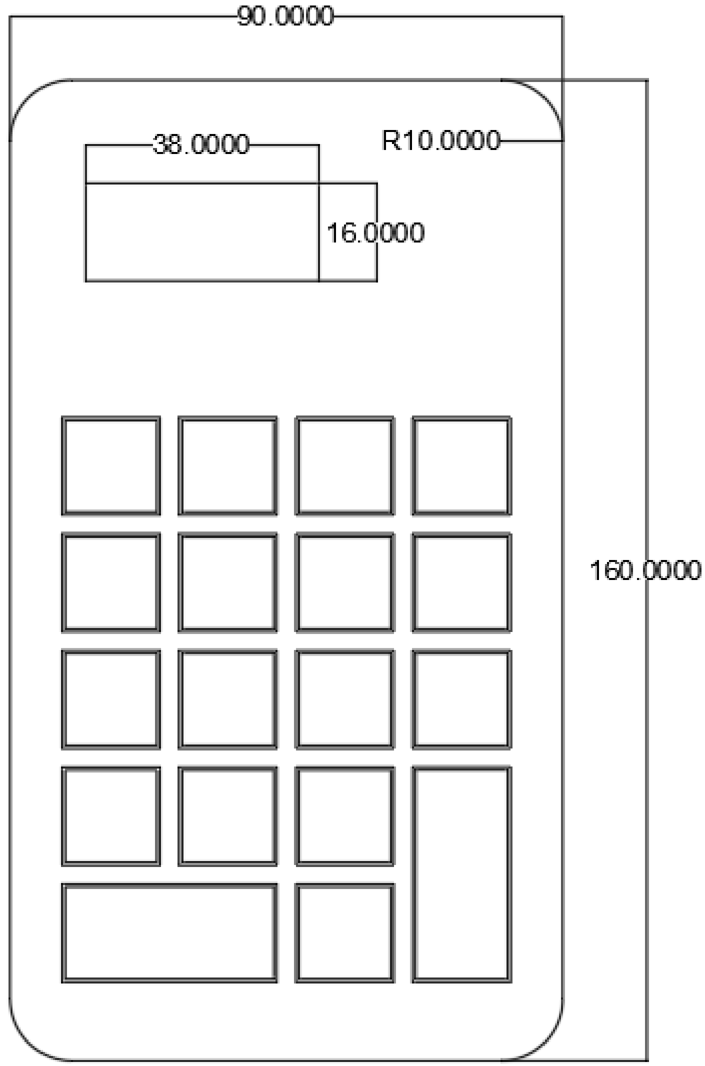
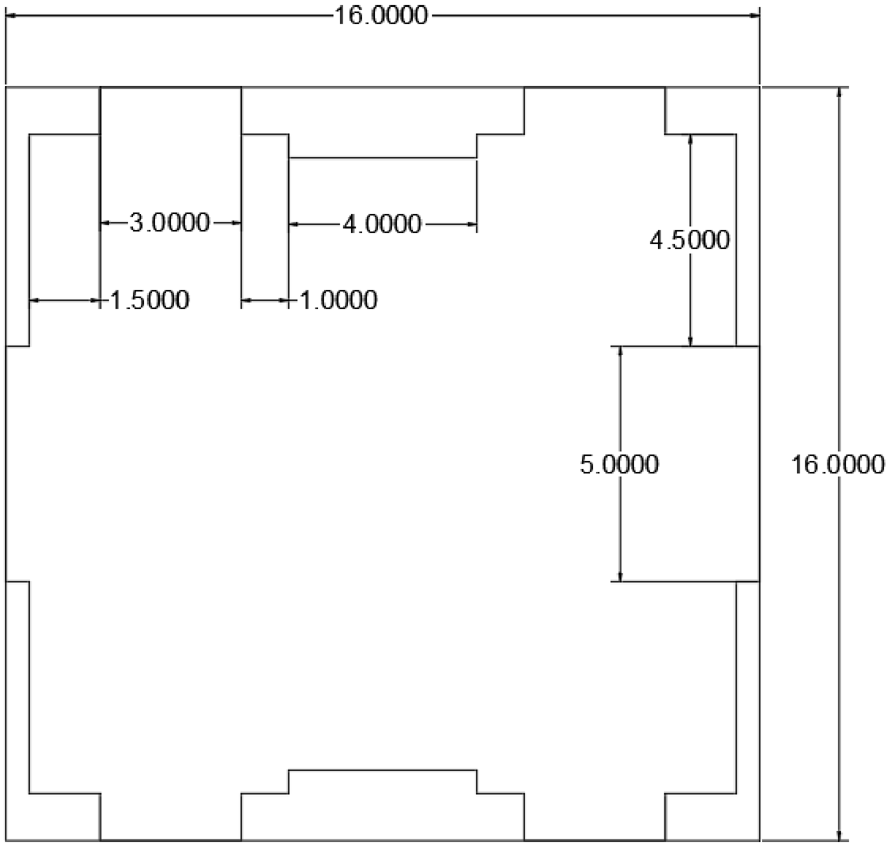
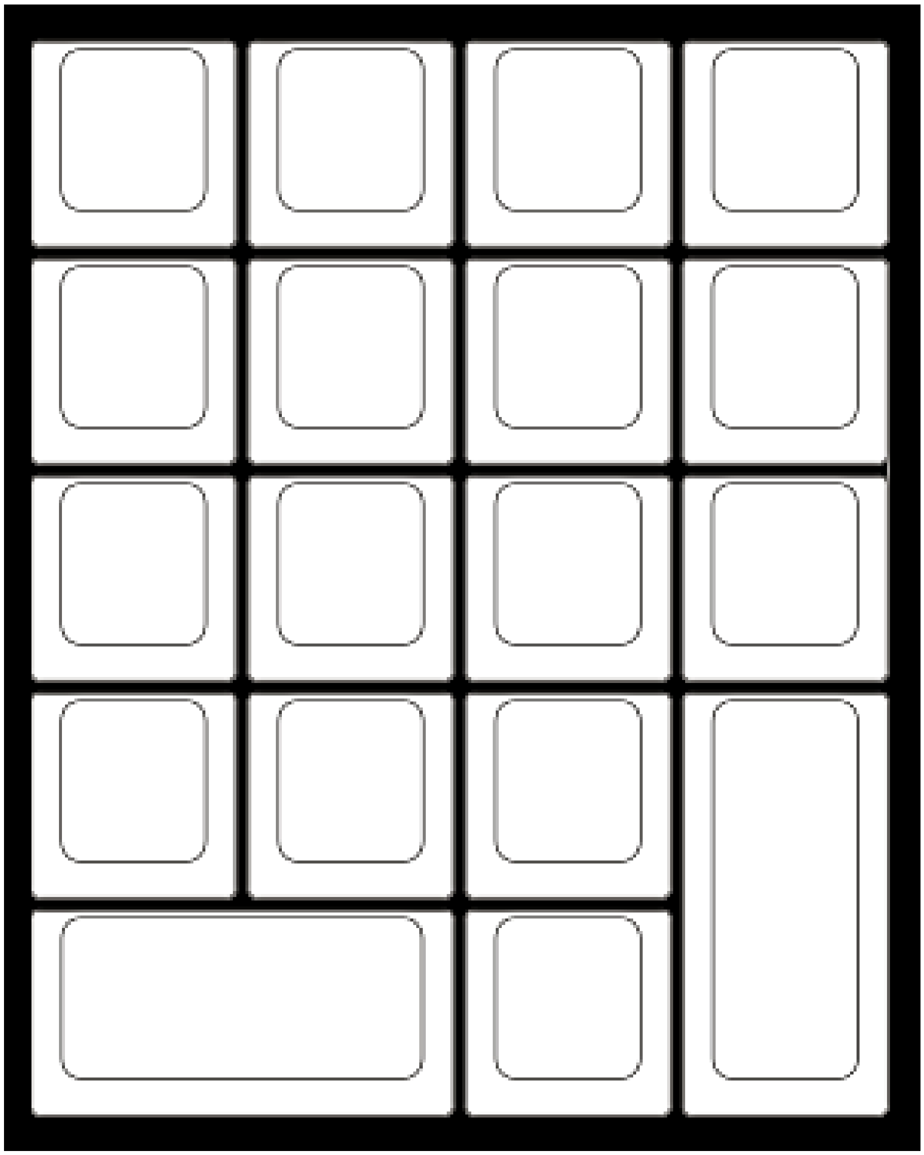


Figure 15: Dimensions for Individual Switch Holders



C Keyboard Layout

Figure 16: Visual Representation of the Key Layout



D Controller Code

```
#include <LiquidCrystalFast.h>

// Teensy 3.0 has the debug LED on pin 13
const int ledPin = 13;
const int LCD_RS = 2;
const int LCD_RW = 3;
const int LCD_EN = 4;
const int LCD_D4 = 5;
const int LCD_D5 = 6;
const int LCD_D6 = 7;
const int LCD_D7 = 8;

const byte ROWS = 5;
const byte COLS = 4;

int MODES = 3;
int currMode = 0;

bool shift_On = false;
bool caps_On = false;
bool caps_Lock = false;
int currLayer = 0;
int prevLayer = 0;
int mode_counter = 0;

int capsLayer = 1;
int shiftLayer = 2;
int caps_shift = 3;


  /* DEFINE MODIFIERS AS:
    ALT_GR = #
    ALT = $
    SHIFT = %
    ESC =
    ENTER = `
    LEFT_ARROW_KEY = <
    RIGHT_ARROW_KEY = >
    */
const char* modes[3] = {
  "Word_","LaTeX","ASCII"
};

const char* layout[][ROWS][COLS] = {
  {//layer 0 = word - normal
  {"caps","NULL","\\cup","NULL"},
  {"\\mu","\\delta","\\theta","shift"},
  {"\\forall","\\ge","\\simeq","+-"},
  {"\\infty","\\pi","\\sigma","$jem`  "},
  {"\\sqrt","^2","$jei`  ","d/d"},
  },
  {//layer 1 = word - caps
  {"caps","NULL","\\bigcup","NULL"},
  {"\\Mu","\\Delta","\\Theta","shift"},
  {"NULL","NULL","~=","NULL"},
  {"NULL","\\Pi","\\sum","NULL"},
  {"NULL","NULL","NULL","NULL"},
  },
  {//layer 2 = word - shift
  {"caps","NULL","\\cap","NULL"},
  {"\\lambda","\\phi","\\omega","shift"},
  {"\\exists","\\le","\\ne","-+"},
  {"\\emptyset","\\angle","NULL","$jem>>`  "},
  {"$jer>`","^","$jei>`","\\partial/<\\partial"},
  },
  {//layer 3 = word - shift+caps
  {"caps","NULL","\\bigcap","NULL"},
  {"\\Lambda","\\Phi","\\Omega","shift"},
  {"NULL","NULL","NULL","NULL"},
  {"NULL","NULL","$jeg>`  ","NULL"},
  {"NULL","NULL","NULL","NULL"},
  },



  {//layer 4 = latex - normal
  {"caps","NULL","\\cup","NULL"},
  {"\\mu","\\delta","\\theta","shift"},
  {"\\forall","\\geq","\\simeq","\\pm"},
  {"\\infty","\\pi","\\sigma","\\log"},
  {"\\sqrt{}<","^2","\\int\\mathrm{d}x<<<<<<<","\\frac{\\mathrm{d}}{\\mathrm{d}}<"},
  },
  {//layer 5 = latex - caps
  {"caps","NULL","\\bigcup","NULL"},
  {"\\Mu","\\Delta","\\Theta","shift"},
  {"NULL","NULL","\\cong","NULL"},
  {"NULL","\\Pi","\\sum","NULL"},
```

```
  {"NULL","NULL","NULL","NULL"},
  },
  {//layer 6 = latex - shift
  {"caps","NULL","\\cap","NULL"},
  {"\\lambda","\\phi","\\omega","shift"},
  {"\\exists","\\leq","\\neq","\\mp"},
  {"\\emptyset","\\measuredangle","NULL","\\lim_{m_\\to_\\n}"},
  {"\\sqrt[n]{}<","^","\\int_a^b_\\mathrm{d}x_","\\frac{\\partial}{\\partial}}"},
  },
  {//layer 7 = latex - shift+caps
  {"caps","NULL","\\bigcap","NULL"},
  {"\\Lambda","\\Phi","\\Omega","shift"},
  {"NULL","NULL","NULL","NULL"},
  {"NULL","NULL","\\displaystyle\\sum_{k=m}^n","NULL"},
  {"NULL","NULL","NULL","NULL"},
  },



  {//layer 8 = unicode - normal
  {"caps","NULL","\\cup","NULL"},
  {"\\mu","\\delta","\\theta","shift"},
  {"\\forall","\\ge","\\simeq","+-"},
  {"\\infty","\\pi","\\Sigma","$jem`  "},
  {"\\sqrt","^2","$jei`  ","#92"},
  },
  {//layer 9 = unicode - caps
  {"caps","NULL","\\cup","NULL"},
  {"\\mu","\\delta","\\theta","shift"},
  {"\\forall","\\ge","\\simeq","+-"},
  {"\\infty","\\pi","\\Sigma","$jem`  "},
  {"\\sqrt","^2","$jei`  ","capson"},
  },
  {//layer 10 = unicode - shift
  {"caps","NULL","\\cup","NULL"},
  {"\\mu","\\delta","\\theta","shift"},
  {"\\forall","\\ge","\\simeq","+-"},
  {"\\infty","\\pi","\\Sigma","$jem`  "},
  {"\\sqrt","^2","$jei`  ","shifton"},
  },
  {//layer 11 = unicode - shift+caps
  {"caps","NULL","\\cup","NULL"},
  {"\\mu","\\delta","\\theta","shift"},
  {"\\forall","\\ge","\\simeq","+-"},
  {"\\infty","\\pi","\\Sigma","$jem`  "},
  {"\\sqrt","^2","$jei`  ","shiftcapson"},
  }
};

byte row[ROWS] = {15,16,17,18,19};
byte col[COLS] = {20,21,22,23};
byte mu[] = {
  B00000,
  B10001,
  B10001,
  B10001,
  B10001,
  B11111,
  B10000,
  B10000
};
LiquidCrystalFast lcd(LCD_RS, LCD_RW, LCD_EN, LCD_D4, LCD_D5, LCD_D6, LCD_D7);

void setup() {
  lcd.begin(8,2);
  lcd.createChar(0, mu);
  lcd.setCursor(2,1);
  lcd.print("Board");
  lcd.setCursor(0,0);
  lcd.print("Power_On");
  lcd.setCursor(1,1);
  lcd.write(0);
  Serial.println("Printed...");
  // initialize the digital pin as an output.
  pinMode(ledPin, OUTPUT);
  for (int c = 0; c < COLS; c++){
    pinMode(col[c], INPUT);
  }
  for (int r = 0; r < ROWS; r++){
    pinMode(row[r], OUTPUT);
  }
  Serial.begin(9600);
  Keyboard.begin();
  delay(500);
  lcd.clear();
  lcd.print("Mode:");
  lcd.setCursor(0,1);
  lcd.print(modes[currMode]);
}
```

```
void setKey(char keypress){
  /* DEFINE MODIFIERS AS:
    CTRL =
    ALT = $
    SHIFT = %
    */
  // Catch Modifiers
  if(strcmp("#",&keypress) == 0){
    Keyboard.press(KEY_LEFT_ALT);
    Keyboard.press(KEY_NUM_LOCK  | 230 | 231);
  }
  else if(strcmp("$",&keypress) == 0){
    Keyboard.press(KEY_LEFT_ALT);
  }
  else if(strcmp("<",&keypress) == 0){
    Keyboard.press(KEY_LEFT);
    Serial.println("DETECTED");
  }
  else if(strcmp("  ",&keypress) == 0){
    Keyboard.press(KEY_ESC);
  }
  else if(strcmp("`",&keypress) == 0){
    Keyboard.press(KEY_ENTER);
  }
  else if(strcmp("%",&keypress) == 0){
    Keyboard.press(KEY_LEFT_SHIFT);
    Serial.print("Shift");
  }
  else if(strcmp(">",&keypress) == 0){
    Keyboard.press(KEY_RIGHT);
  }
  else if(strcmp("~",&keypress) == 0){
    Keyboard.releaseAll();
    Serial.print("Release");
  }
  else{
    Keyboard.write(keypress);
    Serial.print(keypress);
  }

}

// Macro sequence
void setKeyMap(const char* keypressed){
/* DEFINE MODIFIERS AS:
    CTRL =
    ALT = $
    SHIFT = %
    */
        if(strcmp("caps",keypressed) == 0){ // caps toggle
              currLayer = currLayer - 2 * (currLayer % 2) + 1;
    caps_Lock = !caps_Lock;
    if(caps_Lock){
        lcd.setCursor(7,1);
        lcd.print("C");
    }else{
      lcd.setCursor(7,1);
      lcd.print("_");
    }
        } else if(strcmp("shift",keypressed) == 0){
        }
        else {
    int len = strlen(keypressed); //get the length of the string
    Serial.print(len);
    int i = 0;
    for (i = 0; i < len; i++){ //iterate through each character in the string
      if(i>5){
        Keyboard.releaseAll();
      }
      setKey(keypressed[i]); //set the key equal to this character
    }

  Keyboard.press(KEY_SPACE);
  Keyboard.releaseAll();
        }

}

void loop() {

  for (int r = 0; r < ROWS; r++) {
    digitalWrite(row[r], HIGH); //drive each row high one by one
    for (int c = 0; c < COLS; c++){
      if (digitalRead(col[c])){ //check if each column is high, one by one
        if((strcmp(layout[currLayer][r][c],"shift") == 0) && !shift_On){
          currLayer = currLayer + 2;
          shift_On = true;
          lcd.setCursor(6,1);
          lcd.print("S");
        }
        if((strcmp(layout[currLayer][r][c],"caps") == 0)){
```

```
          caps_On = true;
          lcd.setCursor(7,1);
          lcd.print("C");
        }
          // Checks to see if the key pressed is defined in the layout
          if(strcmp(layout[currLayer][r][c],"NULL") != 0){
            setKeyMap(layout[currLayer][r][c]); // Work out what to send and send it
                .
          }
      }else if((strcmp(layout[currLayer][r][c],"shift") == 0) && shift_On){
        currLayer = currLayer - 2;
        shift_On = false;
          lcd.setCursor(6,1);
          lcd.print("_");
      }else if(strcmp(layout[currLayer][r][c],"caps") == 0){
        caps_On = false;
      }
    }
    digitalWrite(row[r], LOW); //reset the current column to zero
  }
  if(shift_On && caps_On){
    mode_counter++;
  }else{
    digitalWrite(ledPin,LOW);
    mode_counter = 0;
  }
  if(mode_counter > 4){
    currMode = (currMode + 1) % MODES;
    lcd.setCursor(0,1);
    lcd.print(modes[currMode]);
    digitalWrite(ledPin,HIGH);
    currLayer = (currLayer + 4) % (MODES * 4);

    currLayer = currLayer - 2 * (currLayer % 2) + 1;
    mode_counter = 0;
    delay(300);
  }
  delay(100);
}
```

E Survey Results

1/19/2017 Mathematical Keyboard Survey - Google Forms

Mathematical Keyboard Survey

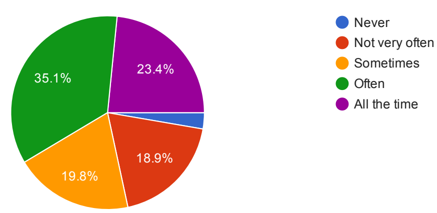
QUESTIONS RESPONSES 111

111 responses

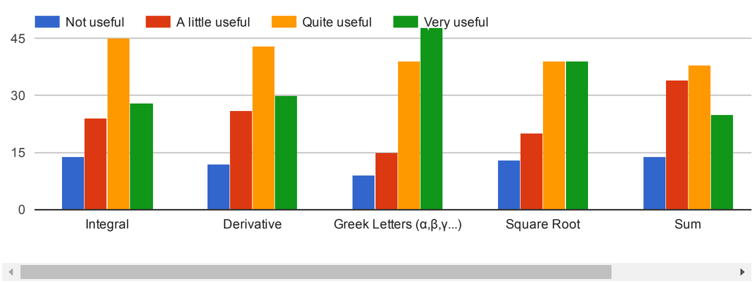
SUMMARY INDIVIDUAL

Accepting responses

How often do you need to include mathematical symbols not available on a traditional keyboard in your documents?
(111 responses)



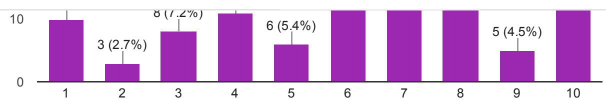
How useful would you find keys for these symbols?



https://docs.google.com/a/dizzyewok.com/forms/d/1O3OhTIS6uDDi1bfEMY-VlqJfWb1hUpT_acO2P0_1xkU/edit#responses 1/3

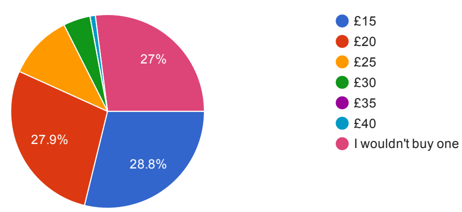
Mathematical Keyboard Survey

QUESTIONS RESPONSES 111



How much would you be willing to pay for a small external mathematical keyboard?

(111 responses)



Do you have any suggestions? (32 responses)

Subscript and superscript buttons could be very useful and probably easily bound.

Would a maths keyboard tie well into word? How would you ensure compatibility with lots of programs? How would this be better than latex? How would you ensure correct formatting for complex expression?

Use LaTeX, mate.

less than £15 actually -- i can get a whole second keyboard for £10

You can already do sum with a normal keyboard, no? But this must be very useful for teachers more than students :)

Generally typing maths in Latex is just fast enough with a regular keyboard.

Make it both LaTeX and Microsoft Word Equations compatible.

You should include a feature which auto types the latex code when enabled

F Meeting Minutes

20161201 Meeting

- 6pm @Central Library Group Study Room B
- Attendance: Orion, Xav, Chelle, Clive, Mariam, Burgess, Koral
- Recorder: Burgess
- General discussion on the ways of achieving the keyboard:

  Software part: need to know how to **programme** the **keyboard microcontroller** and the **app built in the OS**.

  Hardware part: **mechanical keyboard** is an easier way for prototyping our design (comparing to membrane keyboard). We need **3d-printed keyboard case**, **pcb**, **mechanical switches**, **key caps** as hardware.

  Start to design a mathematical keyboard initially. If possible, we can achieve multifunctionality.
- Milestones:

  1. **Research** on what the most commonly used math symbols are. (This week target)

  2. Decide the **layout** for our keyboard:(before Christmas beak)

  Size? Key numbers? Rectangle keys and square keys locations?

  There should be (ideally) no overlap between our keyboard and normal keyboard, since we will keep both. Therefore, we might have no numbers on our keyboard.

  3. **CAD** for hardware. (during Christmas break)

  4. **PCB** design and others. (next term after finishing the PCB lab)
- We start to use **Slack** as our online communication/discussion tool.

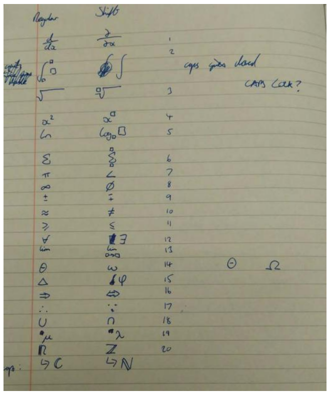
- Work for this week

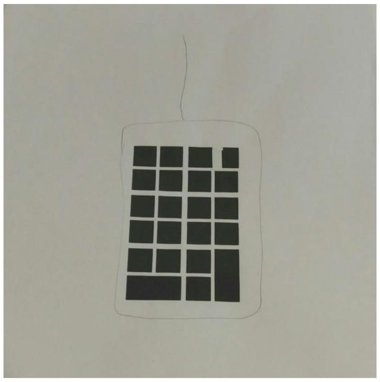
1. Research on what the most commonly used math symbols are.
2. Come up with ideas about the layout.

- Meeting finished @7pm

20161208 meeting

- 6pm central library group study room C.
- Attendance: everyone.
- Recorder: Burgess.
- New member: Antonio Enas.
- Summarizing the info and ideas we currently have:
  1. Based on the background research, we came up with a rough table of symbols we should have on our keyboard (below).
     Reference:
     http://www.tmrfindia.org/sutra/v2i17.pdf
     http://www.rapidtables.com/math/symbols/Basic_Math_Symbols.htm
  2. Decided the rough layout of our keyboard (below).





- We will consult Dr. Daniel Nucinkis about the symbols we need to keep and will consult Dr. Ed Stott, our supervisor, about the digital control core and software interface design of our prototype.
- We want to try 3D-print the keycaps, so that we can have symbols 'carved' on the keys. But we will as well buy blank key caps at this stage for testing.
- We decide to purchase keycaps and switches from China. It's cheap.
- We will set Christmas break missions on next meeting.
- Missions for this week:
  Antonio: figure out how to conduct 3-D printing at imperial.
  Koral: try to print one key.
  Clive: start to work on CAD.
  Burgess: decide where, which and how to buy key caps and switches.
  Orion: consult Nucinkis and conduct a survey for math symbols.
  Marian: consult Ed Stott.
  Keep researching and coming up with ideas!
- Next meeting shall be at next Wednesday 1pm.
- Meeting finished @7.35pm.

20161214 Meeting

- 1pm Central Library Group Study Room B.
- Absence: Mariam
- Recorder: Burgess
- Summarizing the progress so far:
    1. Resources about **cap 3-D printing** are collected and post on slack.
    2. We have consulted **Dr. Stott** about the keyboard designing. Help has been offered.
    3. **Online survey** for suggestions was built and post online.
    4. **Purchase list** was set. And we will have Mariam purchasing the components from China and shipped to Pakistan and will be carried back to London.
- Works to do during the Christmas break:
  Burgess: work on the **PCB** designing.
  Chelle, Burgess: work on the **circuit** designing.
  Xav: work on the **microcontroller** (teensy microcontroller) programming.
  Clive, Antonio: work on **CAD for case and frame**. (CAD here is for presentation and 3D-print.)
  Xav, Koral: work on **software** designing.
  Burgess, Mariam: **Order hardware** and components for PCB.
  Antonio: research on **manufacturing**.
  Orion: start **interim report.** (to cover: aims, specification, define structure, managerial stuffs: gantt chart, selection matrix…)
  Everyone: **share the survey** link to relevant parties.
- We will meet Ed Stott as a team in January.
- Next meeting time to be decided nearly at the start of next term.
- Meeting finished @2pm.

20170111 Meeting

- 11.30 AM Central Library Study Room B
- Recorder: Burgess Xu Wang
- Absence: Mariam
- Summarizing progress so far:
  1. PCB: References found on the web. Useful IP found on the Circuitmaker software's IP category.
  2. CAD and packaging: Initial design obtained. It is good for Demonstration and 3Dprinting. But detailed adjustment needs to be done to perfectly fit our circuit.
  3. Microcontroller and software: research done. We shall use Teensy microcontroller. And we are waiting for the Circuit to be connected and then we can programme the controller and software and do testing.
  4. Component purchase: no component was purchased during the winter, since we need to make orders via our department system, which is yet to open.
  5. Manufacturing: we have found way to manufacture (3D print) the keycaps.
  6. Interim report: good start has been made based on LaTex format.
  7. Online survey: spread out.
- Works to do before next meeting:
  1. **PCB** would be great to finish initial design before next week. And that should be fit for Cherry MX keys and Teensy Controller peripheral.
  2. **CAD and packaging**: the case waits for PCB.
  3. **Microcontroller and software**: wait for the circuit board to be complete.

     PCB→Components→Hardware→Programming→Packaging

  4. **Component purchase**: we really need to order components. We will consult Dr. Perea about it.
  5. **Manufacturing**: we shall try to 3D print the key caps.
  6. **Interim report and website**: Interim report waiting for the guideline lecture. And Xav is willing to build up the website.
- Burgess shall make a group appointment with Ed Stott next week for discussing the progress so far.
- Meeting finished @12.30.

20170116 Group Meeting with Supervisor

- 1 PM Dr. Ed Stott's office
- Recorder: Burgess Xu Wang
- Absence: Koral
- We discuss the progress thus far with our project supervisor: Dr. Ed Stott:
    1. Hardware: a complete PCB design is finished. That is compatible with all the physical peripherals and hardware-software interfaces. First Hardware order (25 keys, Teensy controller, 5 LEDs) has been made.
    2. Software: cooperative HW-SW interface reference has been provided by the HW designers. Much researches upon the controller programming and operation algorithms has been done.
    3. Packaging: 3D-printing induction organized by the imperial robotics soc was taken.
    4. Documentation: interim report has started to collect the info from HW, SW, Packaging design subgroups.
- Suggestions and advises given from Ed Stott:
    1. Hardware: familiarizing about the process for components purchasing, especially the PCB manufacturing. We decided to do sufficient pretest before eventually making the PCB, since to make the PCB is going to be expensive(≈£50 for our design) and irreversible. And the HW design shall be better proofread by Ed Stott before proceeding to manufacturer. During the talk, we found that to have several LED indicators on our board might be hard to handle referring to the circuitry. We decided to have a small black/white LCD display on our keyboard to show any information wanted. PCB needs to be adjusted according to this.
    2. Software: discussion upon functionality and the means to achieving that has been made. Dr. Stott suggested that we may simplify our user interface for simplicity, since the users may not be expecting to install a software app to manipulate the final product. Other ideas concerning the operation layers and display programming has been discussed. We will wait for the microcontroller to arrive this week to effectively start SW designing.
    3. Packaging: CAD and 3D printing methodology have been discussed. The CAD subteam may need to cooperate with HW subteam and pay more attention on the means for mounting the PCB and peripherals into the package.
    4. Documentation: yet to be discussed on Wednesday's meeting.
- It has been a helpful talk covering all the design fields with Ed Stott!
- Next meeting shall be on Wednesday this week.
- Meeting finished @2pm.

20170118 Meeting

- 2PM Central Library Study Room B
- Recorder: Burgess Xu Wang
- Attendance: All
- This meeting is important. We reflect on progress of each design field and discuss upon the interim report composition:
  1. Hardware: components ordered arrived whilst the controller is yet to come.
  2. Software: software team has done a lot of researches upon the implementation of controller programming. Since a display LCD screen shall be utilized, the software designing now faces new task to make keyboard **switches, display, computer interface** cooperate smoothly with each other.
  3. Packaging: initial 3D printed samples are manufactured. But due to the deficient resolution or design, the key cap and key frame do not fit well with our standard key switches. Packaging team shall adjust the CAD to manage the manufacture quality.

  4. Documentation: reflect on the online survey for design guidance. Several constructive suggestions are obtained. Majority of the feedbacks show the interesting of having a product like what we are working on.
- **Tasks at this stage:**
  1. **Hardware**: PCB design shall be adjusted to accommodate the LCD screen and microcontroller. A suitable type of LCD needs to be decided with software team to make sure the it is compatible with other peripherals.
  2. **Software**: research on controller coding.
  3. **Packaging**: optimize the CAD and find out a better means to improve the resolution of 3D printing.
  4. **Documentation**: we have discussed upon the interim report composition. According to the responsibility taken by every member so far, we decide to have specific people to take charge of different part of the report. Below attached a table for the task distribution.
     We decided to have part 3 Project Group Management containing 5&6 responsibilities & communication.
     We will have roughly 10 days to work on our own part of the report individually. A google drive shall be used to accumulate our words. If we have ideas to put in others' parts, we will talk to the person in charge.
     We want to have it finished 7 days before the official deadline so that we can pass it to Mrs. Perea for advises and suggestions.

interim report task distribution		
1	Abstract	Mariam
2	Intro	Mariam
3	Project Group Management	Clive, Orion
4	Specification	BurgessXuWang, Xav
5	Responsibilities	Clive
6	Communication	Orion
7	General Long Term Schedue	Chelle Ma
8	Research&Analysis	BurgessXuWang, Xav, Antonio
9	Summary	Mariam
	Appendix&overall	Koral

- Next meeting shall be in next week.
- Meeting finished @3pm.

20170121 Brief Talk

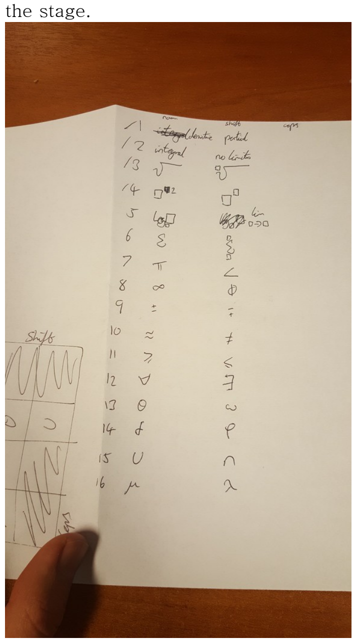
- 2PM online
- Recorder: BurgessXuWang
- Attendance: group members
- We realized that the interim report task distribution we have from last group meeting is not consistent with the **Assessment Guideline**. We therefore made adjustment based on our distribution so far and came up with an updated task distribution:

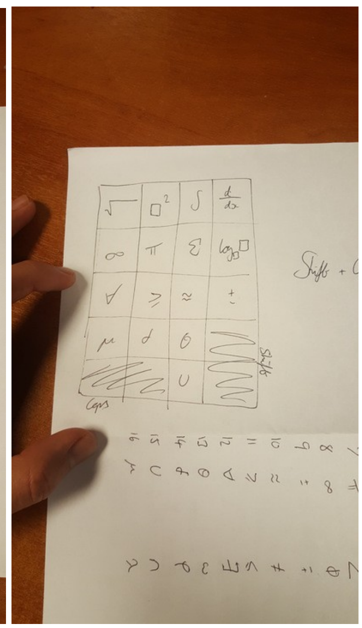
interim report task distribution		
1	cover page	Orion(already done)
2	contents page	Orion(already done)
3	abstract	Mariam
4	introduction/background	Clive,Orion(brief group introduction)/ Mariam(design intro and background)
5	design specification	BurgessXuWang, Xav
6	concept designs	BurgessXuWang, Xav, Antonio
7	discussion	Clive, Orion
8	conclusion and future work	Mariam(help with conclusion), Chelle Ma
9	references	Koral
10	Appendix	Koral
	Overall	Koral
Content of each section please refer to Interim Report Assessment Guidelines.		

- The new version is less 'group managemental' but more 'technical'.
- This new version shall be distributed to every member of the group
- Talk finished @2.30PM

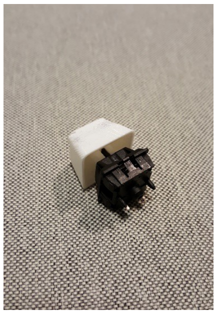
20170126 meeting

- 12PM @Central Lib Study Room B
- Recorder: Burgess Xu Wang
- Attendance: All
- Summary the progress:
  1. Hardware: teensy microcontroller received. In the meeting, the group decided 5x4 keyboard matrix with 18 keys, which are 16 symbol keys and 2 modifiers.
  2. Software: research on programming is continuously conducting. the group decided a list of key symbols on the keys. And a draft of the symbol layout at the stage.





  3. Industrial design: CAD for keycaps and circuit frame has been optimized. A suitable prototype for one key cap has been made. The quality is improved. Hardware team and ID team cooperated to make sure that the frame and PCB fit well together.



4. Documentation: each part of the interim report has been composed. And almost every section has initially finished.

- Tasks at the stages:
  1. **Hardware**: finish up the final design for PCB and PCB's routing. Come up with an entire SW-HD interface reference, which identify the pin assignment and interconnection among Controller core, Display and key matrix to put in the report. After report, the HW team will make up a simplified 2x2 key matrix and hardware circuitry on a breadboard for the SW team to start pretests.
  2. **Software**: continue research and programme buildup.
  3. **Industrial**: to compose the report. And keep on researching on the 3D-print method.
  4. **Documentation**: finish up the report by next week so that the group can save a week's time in advance for Mrs. Perea to do the formative assessment.

- Meeting finished @13pm
- Next meeting time to be decided.

20170201 meeting

- 2pm @EEE Room 508
- Absence: Mariam
- Recorder: Burgess Xu Wang
- Review on what we did so far:
    1. Hardware Design: we have all the components that we are going to have on our keyboard. PCB design refined. Pin assignment reference and refined HW design put in the interim report.
    2. Software Design: research and programming has been overtaken.
    3. Industrial Design: CAD refined.
    4. Documentation: initial report composed. Everyone made a contribution to his specific part. The report has sent to Mrs. Perea for formative assessment. The group is waiting for the feedback.
- Tasks this week:
    1. Hardware Design: to build up a mini circuit by 4 keys with a robust connection for SW team to do the before PCB programming test. Talk to Ed Stott to consult and improve the HW design.
    2. Software Design: implementing macros before testing with the 4 key breadboard.
    3. Industrial Design: improve the ID and 3Dprint one 2x2 mini frame for pretesting.
    4. Documentation: refine the report with respect to the feedback from Perea. We talked about the web designing as well.
- Next meeting next week.
- Meeting finished @3.30pm

G Expenditure

Figure 17: Table of Total Expenditure

Component List					
Component	Quantity	Price per item		Total	
Teensy 3.2 Microcontroller	1	£	19.00	£	19.00
Cherry MX Black keyboard Switch	25	£	0.48	£	12.00
Midas 8x2 LCD display (MC20805A6W-GPR-V2)	1	£	5.68	£	5.68
Fairchild/ON diode 1N4448TR	18	£	0.08	£	1.40
RS 10k resistor (Pull-down resistor)	4	£	0.01	£	0.05
RS 1k resistor (current limiting resistor)	4	£	0.01	£	0.05
RS 100 resistor (Contrast)	1	£	0.01	£	0.01
Newbury Electronics Customised Express PCB	1	£	58.21	£	58.21
White LED (NSPW500DS) (Not used)	5	£	0.50	£	2.49
Total				£	96.41