Mathematical Keyboard Interim 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: February 7, 2017

Contents

1 Abstract

Professionals working in diverse fields face trouble digitising their notes due to the weaknesses of traditional keyboards, which have barely advanced as technology has evolved. By introducing an external mathematical keyboard with keys representing numeric symbols, scientists, mathematicians and engineers will be able to focus on their work instead of struggling with entering symbols like π repeatedly. Initially, the team invested a considerable amount of time in exploring various keycaps, switches and controllers which would be suitable for use. The keyboard case and keycaps have been designed and will be 3D printed. The PCB is designed and the microcontroller (Teensy 3.2) is ordered. The team is currently looking into how various different software environments will handle USB input signals from the external keyboard. This report comprises of extensive details on the research conducted, design specifications and alterations made to date.

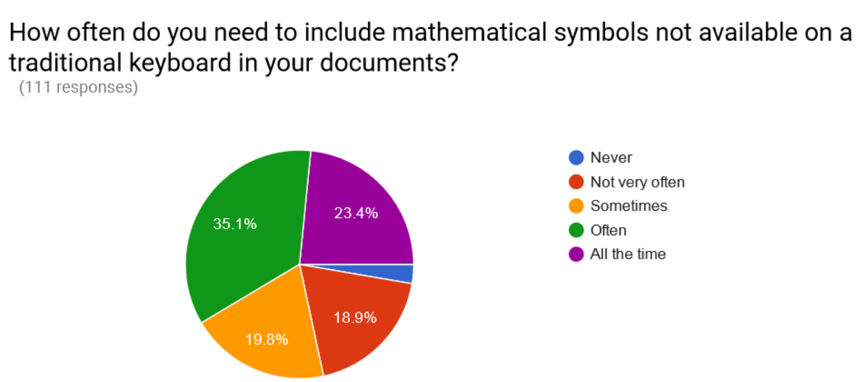
2 Introduction

2.1 Background

Mathematics is a diverse and widely used tool in society, especially in all fields of science and engineering. However, professionals and students alike have trouble digitising their work due to the generalised nature of traditional keyboards. It is often inconvenient to type mathematical symbols which do not appear on current keyboards.

To find out if there was a reasonable market for a mathematical keyboard, an extensive online survey was conducted. Figure 1 shows the survey results for one of the questions (See Appendix E for the full survey questions and results).

Figure 1



As illustrated by Figure 1, only 2.8% of the people who took the survey said they would never need to use mathematical symbols that are not on traditional keyboards. A total of 78.3% said that they at least sometimes needed to use mathmatical symbols in thier documents. This emphasises the weaknesses of using a traditional keyboard and shows that a mathematical keyboard could be extremely useful to many people working in Engineering, Science and Mathematics.

2.2 Project Aim

The team's objective is to design and prototype an External Mathematical Keyboard. It will be designed with a familiar form-factor to ensure that it is intuitive and easy to use, with as little learning curve as possible. This will be an additional keyboard that can be plugged into a computer USB port and used to type mathematical symbols that are not on traditional keyboards. Its aim is to make typing mathematical equations more convenient for users.

3 Specification

In order to strictly define the goals for the project and the product to be prototyped, the team set out to develop a complete design specification. The team made decisions based upon the information gained in performing the survey (Appendix E), as well as online research. Given that the team consisted of those who might find the product beneficial, it was possible to discuss during the meetings the relative importance of specifications such as size, keys and compatibility. There are no commercially available mathematics-focused external keyboards; the only competitors are expensive (£100+) fully-programmable multi-purpose keyboards[1]. The exact symbols and number of keys to be included on the keyboard were decided based on both the team's online research of the most commonly used symbols[2], and the keys on a standard scientific calculator.

The MoSCoW method[3] was used to assign priority to the different elements of the specification, clearly highlighting which elements must, should and could be implemented. It was decided that the design must:

1. Allow the user to type mathematical symbols and equations not easily typed with a traditional keyboard.
2. Function in the presence of any other standard keyboard without interference.
3. Be simple and user-friendly to operate.
4. Be of a reasonable cost ($\leq$£40, otherwise the expense outweighs the benefits for most cases)
5. Use the standard USB interface to maximise compatibility.
6. Not use external power (other than via USB) in order to allow mobile use on a laptop.

It was decided that the design should:

1. Cover the requirements of different mathematical fields (Statistics, Calculus etc.)
2. Be of a reasonably small size (similar to that of the standard keyboard numpad).
3. Work with any operating system and not require additional drivers to be installed.
4. Be ergonomic and comfortable to use, with the keys arranged to maximise speed of typing.
5. Be robust and durable, as most keyboards last years of extended use.

And finally it was decided that the design could (depending on available time and feasibility):

1. Allow the user to switch between different operating modes (e.g. to work with Microsoft Word, LaTeX etc.)
2. Be reprogrammable by the user to enable custom key layouts, even ones related to specialist fields other than mathematics.
3. Use Bluetooth connectivity to allow wireless usage.

The team initially decided on a variation of a traditional num-pad layout which can be seen in Figure 2.

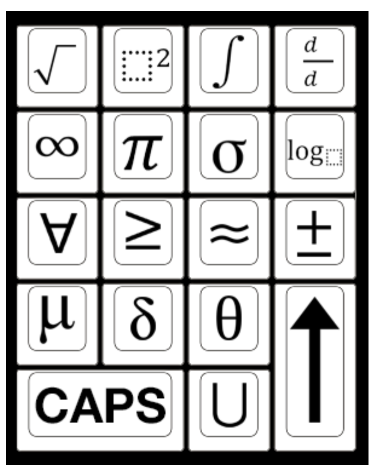


Figure 2: Mock-up of the keyboard layout

4 Project Organisation

The team employed a number of different strategies for enhancing group efficiency. All communications were handled through Slack, an online platform which allows teams to collaborate and communicate in discrete channels to separate information so that it can be accessed and referred to more easily. 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 (Appendix F) and project expenditure (Appendix G) thus far.

To enhance group efficiency, tasks were distributed between four main divisions; Hardware, Software, Industrial Design and Organisation/Administration. The responsibilities of each member are tabled below:

Assignment	Team Members involved	Role Description
Hardware	Burgess Wang, Xavier Kearney	Find and design the hardware required for the keyboard.
Software	Xavier Kearney, Koral Hassan	Discover all alternative methods to output mathematical symbols, and determine which will be used. Research the most efficient algorithm and programming language. Write code to emulate typing mathematical symbols within various software environments (e.g. Microsoft Word, LaTeX).
Industrial Design (CAD)	Antonio Enas, Clive Toh Soon	CAD Design of keycaps, case, switch holders and top plate. Research manufacturing technologies and available facilities.
Organisation/ Administration	The entire team	Set the times and book the venues for the team's 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.

5 Design Research and Analysis

According to the project proposal, the goal is to implement a computer-peripheral (keyboard) with keys that map to the most commonly used mathematical symbols (determined as a result of online research[2]). As the specification for the project was well defined, the design process involved making specific decisions related to the individual modules involved rather than producing a number of vastly different concept designs. This section is divided into the three distinct design areas that the team split up into: Hardware Design (HD), Software Design (SD), and Industrial Design (ID). It separately explains the research and analysis the group has undertaken, and the design decisions the group has made as a result of this research.

5.1 Hardware Design

Hardware Design refers to the projects physical configuration: its circuits, components, micro-modules etc. The packaging and physical realisation of this design is introduced in the ID subsection. After the first group meeting, the HD subgroup decided to achieve the design specification by implementing a mechanical keyboard (with discrete mechanical key switches) rather than a membrane keyboard (with the key switches printed on a plastic membrane). This was a fairly simple decision as membrane keyboards require a customized circuit membrane and a specific silica-gel key layer, which would be incredibly difficult to prototype. Mechanical keyboards can be produced in an entirely discrete manner by purchasing individual key switches and assembling them onto one small 2-layer PCB (Printed Circuit Board) as desired.

The group utilised CircuitMaker software[4] to design the PCB as it was used in the spring term PCB lab. The circuit itself contains 3 interconnected parts: a microcontroller to process key-pressed signals, an LCD display to indicate the operation mode and provide a user interface, and a custom 4 x 5 key matrix network with the mechanical key switches. The 3-D and schematic circuit can be seen in Appendix A. By pressing a key in the matrix, the switch short-circuits the node connecting a row and a column. The microcontroller drives a single row at a time and therefore can detect when a column is short-circuited. Diodes connected in series with each key switch to guarantee that the signals flow from row sources directly to column sinks without backward interference, and isolate each key to allow pressing multiple keys simultaneously. There are 18 keys on the board including 2 double-size keys on the edges. One double-key acts as a "Shift" button that when held down, makes the keyboard switch to another set of symbols (using the microcontroller). The other double-key essentially functions as a "Caps Lock" for all the greek letters available. Pressing both double-keys simultaneously causes the keyboard to switch between operation modes (e.g. Microsoft Word, LaTeX). The 8 x 2 LCD display is predominantly used to indicate the mode in which the user is operating. Initially the HD subgroup planned 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 display to indicate the working modes, which can be controlled easily using the microcontroller SPI (Serial Peripheral Interface) and the built-in LiquidCrystal library[5].

The PCB layout is simple and requires no jumper wires. It measures 87mm x 142.25mm. The 2-D circuit can be seen in Appendix A. The key matrix keeps a precise spacing of 19.02mm between the keys. The full list of components purchased and their quantities/prices is available in Appendix G. Tests are currently being conducted with a smaller, 2x2 matrix to ensure that the hardware setup interfaces correctly with the software and that the specification can be met. After sufficient preliminary testing, the PCB will be checked by Dr. Stott and then ordered via the EE Stores. Finally, the board will be handed over to the SD subgroup for programming.

5.2 Software Design

When the user presses a key it short-circuits a line in the switch matrix as described above. This user input is processed by a microcontroller module running custom firmware which transforms the key press into an output that can be recognised by the computer.

Research on the internet revealed an almost overwhelming amount of information about controlling a keyboard with a custom layout. A lot of introductory information was found on Reddit[6] which revealed a number of open-source options for firmware we could repurpose. The Teensy brand of microcontroller produced by PJRC[7] is universally accepted in this online keyboard community to be the microcontroller of choice 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). Although the design would include far fewer keys than a traditional full keyboard, it made sense to use the Teensy so that the team could take advantage of the multitude of online resources dedicated to custom keyboards.

With the exact microcontroller chosen (specifically the Teensy 3.1 due to availability), the team had to decide on the specific open-source firmware framework that would be expanded. From the plethora available, two possibilities arose: the simple JetpackTuxedo firmware[8] based on Arduino, or the more complicated but more customisable Kiibohd Controller[9].

The preference would be to use the simpler firmware framework if possible, to simplify and speed up the design process, so the software team were tasked with determining whether it was possible to achieve everything in the specification with this framework. For clarification the team contacted the author (JetpackTuxedo) to ascertain whether various features were possible. It was found that the code would not fulfil the requirements without modification but the core operating principle could be harnessed to achieve with the desired specification. The software development team set to work implementing these changes and are now in the process of producing a prototype firmware for the controller which will be tested first on a breadboard and then on the PCB. It is likely that changes will need to be made to accommodate for unforeseen bugs, but this should be of little consequence given that the physical design need not change.

The full code is available both in the Appendix D, and also in the team's Github repository[10] which has been used to track changes and collaborate easily without overlap.

5.3 Industrial Design

AutoCAD 2017[11], a professional Computer Assisted Design (CAD) software, has been employed to design a customised keyboard case and keycaps compliant to the project specifications, adaptable and of high precision.

The purpose of the case is to protect the internal electronic components, as well as to provide a customer oriented and aesthetically pleasing design for the peripheral. The dimensions of the case to accommodate the internal circuitry, LCD display and microcontroller were set at 90mm x 160mm x 25mm. From the dimensions in the datasheet[12] of the Cherry MX an appropriate switch holder has been designed. This module has been included in a switch holder plate, to be mounted over the PCB as a support for the switches and to protect the board. A top plate to fill the space between the keys, cover the microcontroller and provide an opening for the LCD has been designed. The case, switch holder and plate designs are available in the Appendix B.

Several 3D keycap models for Cherry MX switches are available in online libraries. Three different designs have been 3D printed and tested. The final choice due to manufacturing feasibility has been the Cherry MX Keycap published by the user bombtree on Thingiverse.[13] Alternatives to place symbols on the keycaps have been considered as follows:

Technique	Pros	Cons
Handwritten symbols on keycaps	Very easy to achieve; good for prototyping; free.	Poor aesthetical result; symbols could fade with use.
Symbols on stickers applied on keycaps	Easy to achieve; limited cost; good aesthetic result.	Stickers might not last intensive use.
Embossed/engraved symbols on keycaps	Good aesthetic result; no added cost to the keycap manufacturing process; longer lasting solution.	Hard to achieve with available 3D printers due to the process of layer print and support structures required; time consuming as it requires multiple CAD designs.
Use laser-cutting to engrave symbols on wood surfaces glued on top of plastic keycaps	Optimal aesthetic effect with original design; customisable to shapes not achievable with 3D printing; permanent result.	Requires mastering of laser cutter operation and safety issues; higher cost compared to the other solutions.

The group agrees in pursuing further research in the 3D printing and laser cutting options, and in the case of failure to use stickers.

Case and keycaps will be 3D printed as no other option offers the same cheap, customisable and fast manufacturing features. The process of 3D printing is simple: a software slices the CAD designed object into thin horizontal layers, an extruder melts the filament of the plastic material and deposits it over a heated plate recreating layer after layer. However, some attention has to be paid to the printers settings, the infill percentage (how dense the object is), and the presence of a raft (a flat plastic surface printed in order to avoid the filament to not stick to the bottom surface). Moreover, there are some restrictions to the design process as well. It has to be watertight and specific support structures have to be added in the case of empty spaces. Figure 3 shows a 3D render of the potential product.

Figure 3: 3D Render of the Complete Case

3D printers mainly use two materials of similar properties: ABS (Acrylonitrile butadiene styrene; thermoplastic)[14] and PLA (Polylactic Acid; a biodegradable thermoplastic obtained mainly from corn starch.).[15] The printers available at this stage offer only the PLA option, therefore this is the material of choice for prototyping purposes.

The first attempt has been held at the Imperial College Robotic Society lab with a Makerbot Replicator 2[16] printer combined with MakerBot Desktop software[17] for object slicing and printer settings. The minimum achievable layer thickness is of 0.2mm. A working switch holder has been printed but the keycap is not of a high enough resolution to be used. The second attempt has been held at the Imperial College Advanced Hackspace[18], using a Ultimaker 2[19] printer set at 0.1mm layer thickness with the software Cura 2.3.1[20]. A perfectly working keycap and switch holder have been produced with this equipment. Predicted time and material quantity by Cura 2.3.1:

Component	Infill	Time	Material (m)
Keycap (1)	Dense	0h 13min	0.15
Top Plate	Light	5h 13min	4.37
Case	Light	9h 54min	8.71

6 Future Work

There are a few complications that the team is currently facing and various solutions are being explored. The main complication threatening the achievement of the full design specification is the universal compatibility of the software; it is possible that the team will require that the user installs a certain module on their PC before being able to use the keyboard's full functionality. The main goal from now onwards will be for the team to implement the full design on the PCB and perform extensive testing in a number of different software environments. The team will also begin development of its website to showcase the product and its features.

The team recognise that the potential applications of external programmable keyboards are numerous, ranging from social sciences to engineering. If given the time to make further improvements,the team hopes to use the framework of this project to further implement the team's technical knowledge and research into building external keyboards for people from all fields, whether individual or a single reprogrammable device. Wireless (potentially Bluetooth) connectivity is also being considered, although may not be feasible to implement in the given time period.

7 Conclusion

The team has finalised the design of a PCB model file with a functioning circuit matrix which is ready for printing. The team has evaluated 3D printing techniques required to print high quality keycaps and the keyboard case. The software team has conducted research in programming of the microcontroller and are prototyping a complete, user-friendly system to interface with the PC.

The main focus now is on building and testing the keyboard for efficient and accurate performance. Programming the keyboard, and working on improving the compatibility between the USB input signals and different operating modes will be prioritised. The final packaging will take place after the keyboard circuitry prototype is tested and approved.

8 References

[1] X-Keys. (2017). X-keys usb keypad, [Online]. Available: http://xkeys.com/xkeys/xk24.php.

[2] A. S. Isaiah Lankham Bruno Nachtergaele. (2007). Some common mathematical symbols and abbreviations (with history), [Online]. Available: https://www.math.ucdavis.edu/~anne/WQ2007/mat67-Common_Math_Symbols.pdf.

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

[4] CircuitMaker. (2017). Circuitmaker website, [Online]. Available: https://http://circuitmaker.com/.

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

[6] M. Authors. (2017). Reddit /r/mechanicalkeyboards firmware wiki, [Online]. Available: https://www.reddit.com/r/MechanicalKeyboards/wiki/firmware.

[7] PJRC. (2017). Teensy pjrc website, [Online]. Available: https://www.pjrc.com/teensy/.

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

[9] kiibohd. (2017). Kiibohd controller firmware on github, [Online]. Available: https://github.com/kiibohd/controller.

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

[11] Autodesk. (2017). Autocad software, [Online]. Available: http://www.autodesk.com/products/autocad/overview.

[12] Cherry. (2017). Cherry mx keycap datasheet, [Online]. Available: http://cherryamericas.com/wp-content/uploads/2014/12/mx_cat.pdf.

[13] bombtree. (2017). Cherry mx 3d printing template, [Online]. Available: http://www.thingiverse.com/thing:1289777.

[14] Wikepedia. (2017). Abs, acrylonitrile butadiene styrene, [Online]. Available: https://en.wikipedia.org/wiki/Acrylonitrile_butadiene_styrene.

[15] —, (2017). Pla, polylactic acid, [Online]. Available: https://en.wikipedia.org/wiki/Polylactic_acid.

[16] Makerbot. (2017). Makerbot printer, [Online]. Available: https://www.makerbot.com/.

[17] —, (2017). Makerbot desktop, [Online]. Available: https://www.makerbot.com/download-desktop/.

[18] I. C. A. Hackspace. (2017). Imperial college advanced hackspace, [Online]. Available: https://www.imperial.ac.uk/advanced-hackspace.

[19] Ultimaker. (2017). Ultimaker 2 printer, [Online]. Available: https://ultimaker.com/en/products/ultimaker-2-plus.

[20] —, (2017). Cura desktop 2.3.1 software, [Online]. Available: https://ultimaker.com/en/products/cura-software.

9 Appendix

A PCB Design

Figure 4: 3D Render of the PCB

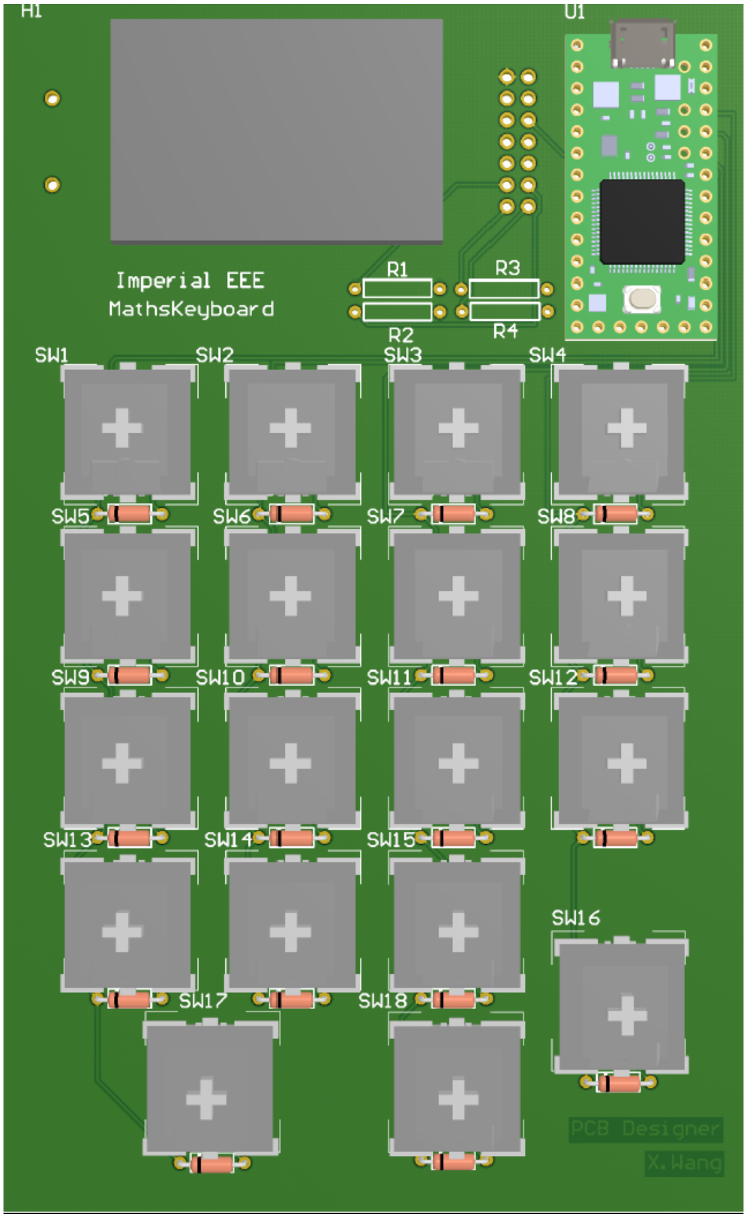


Figure 5: Technical 2D Render of the PCB Layers

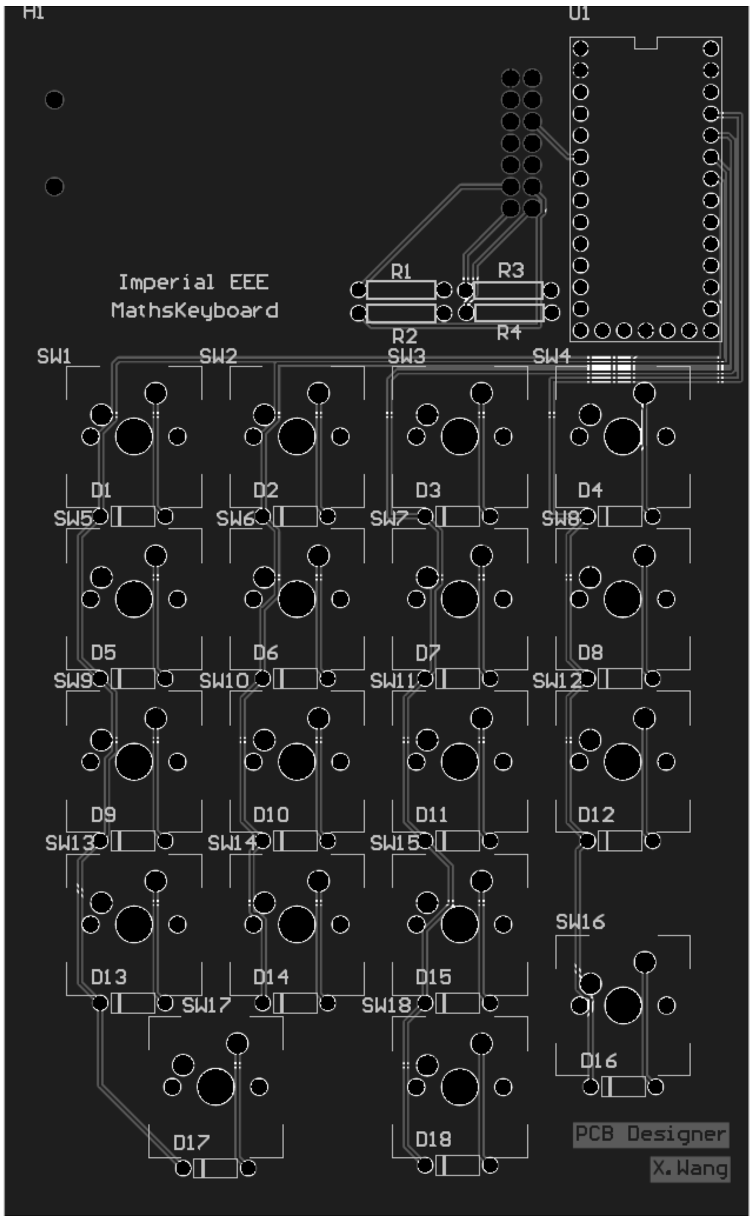
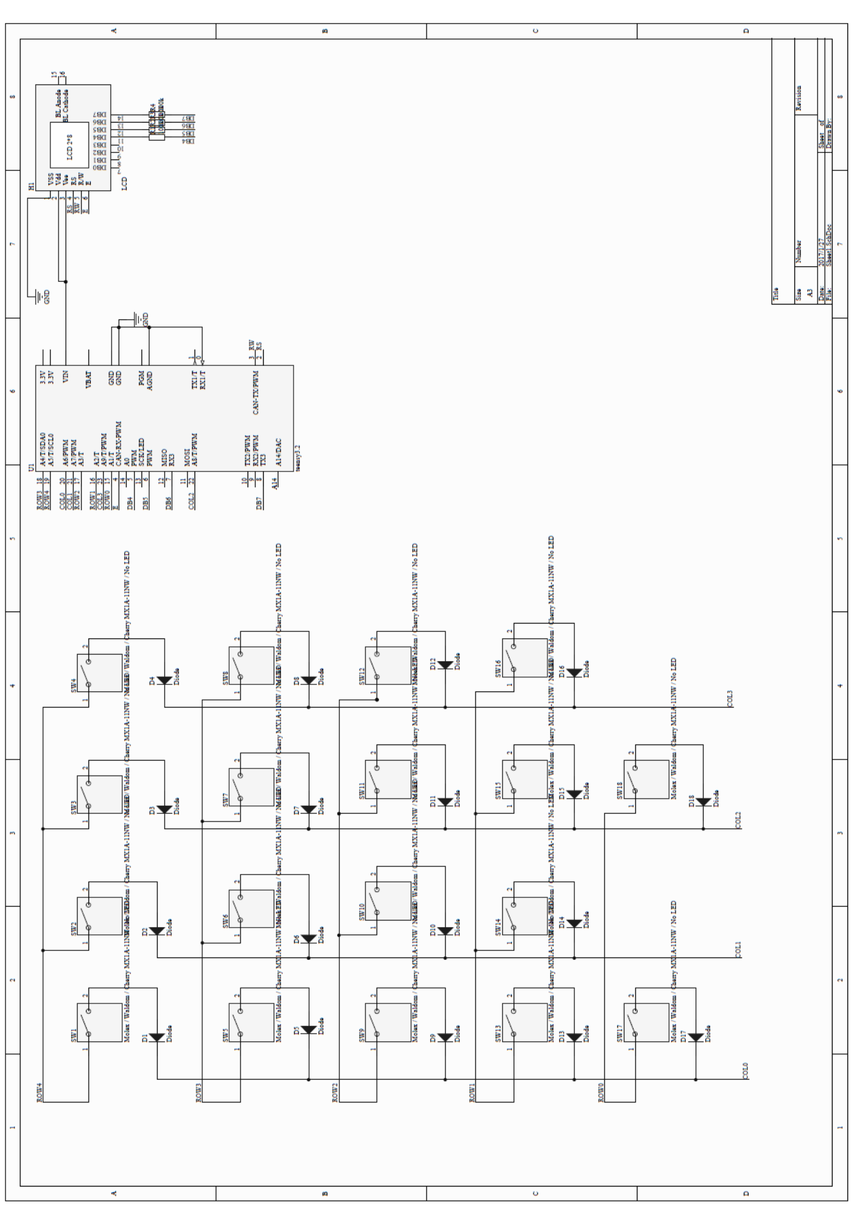


Figure 6: PCB Schematic



B CAD Renders

Figure 7: 3D Render of the Case

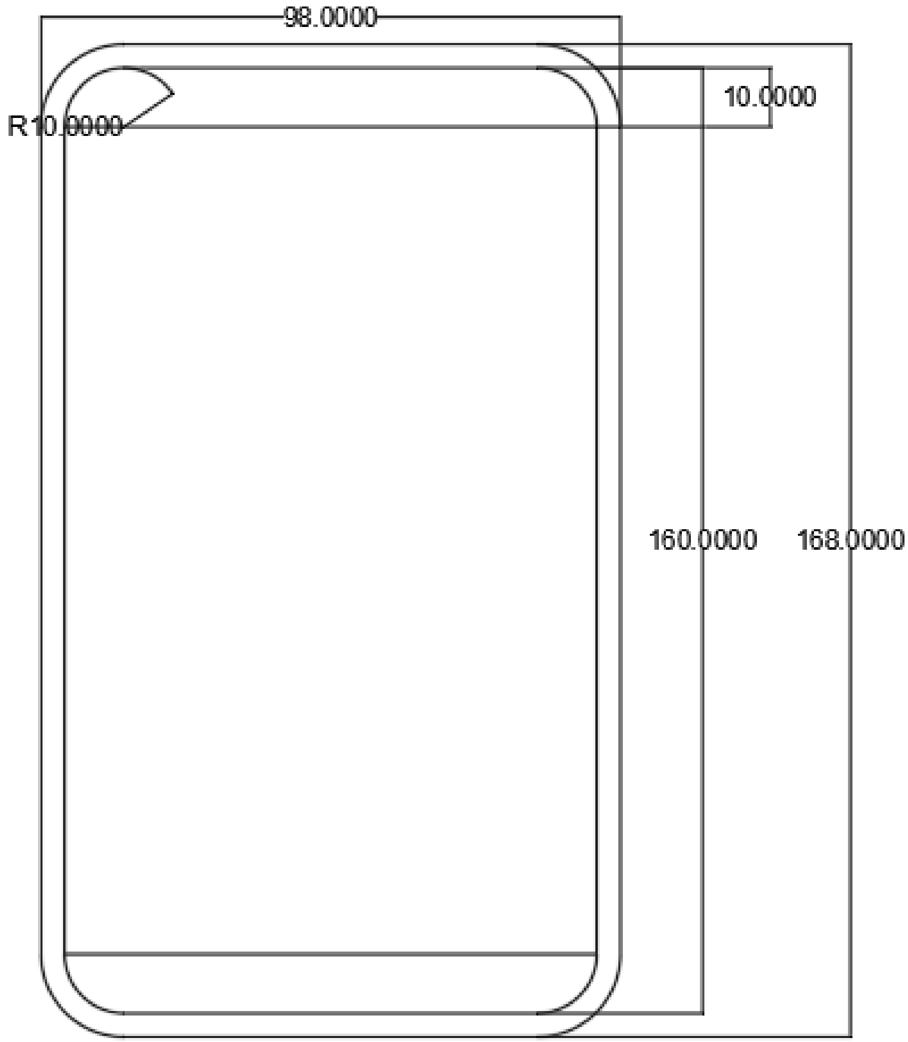


Figure 8: Dimensions for Custom Switch Holder Plate

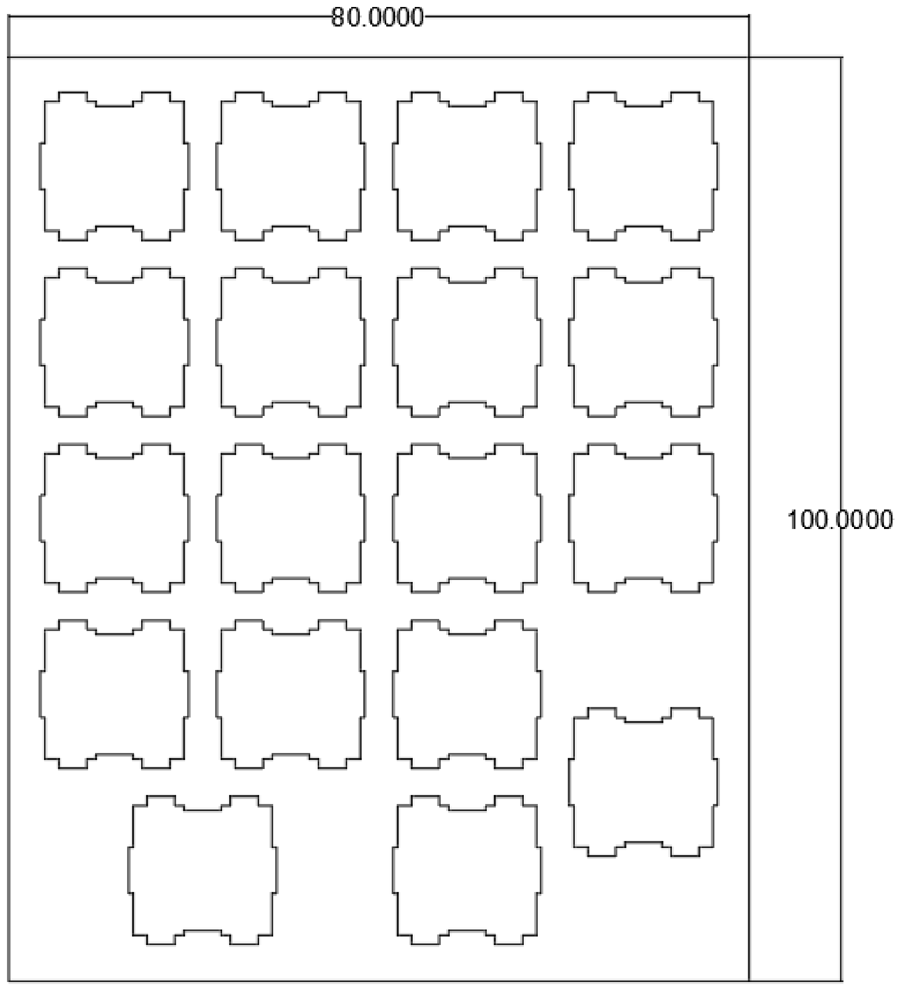


Figure 9: Full Dimensions for the Top Plate

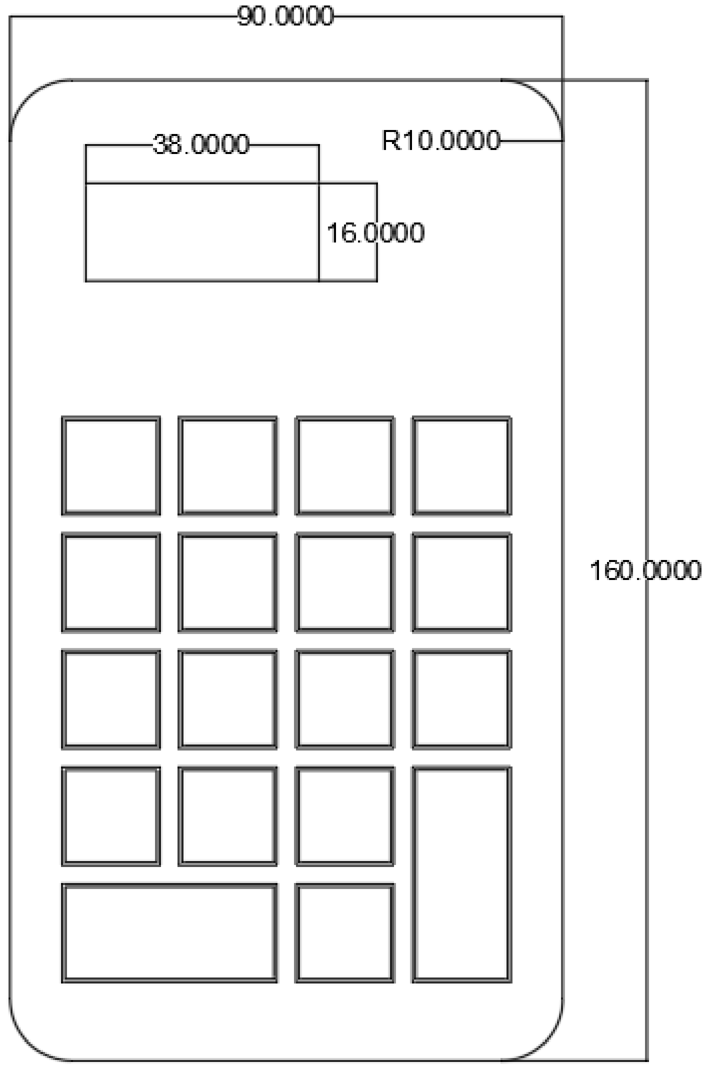
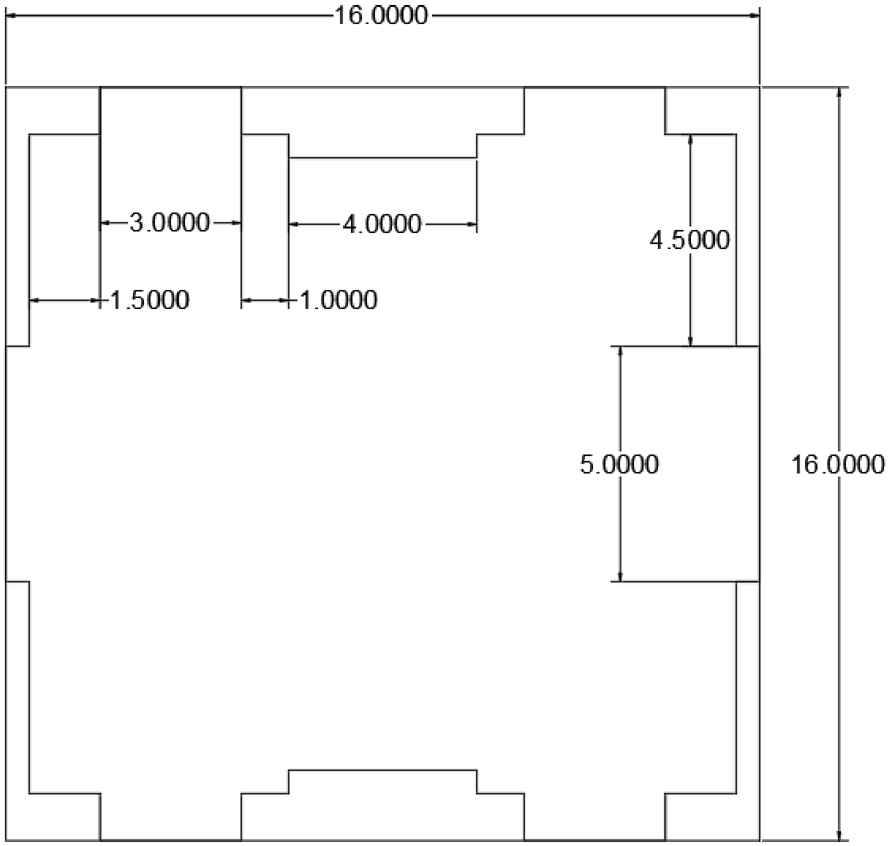
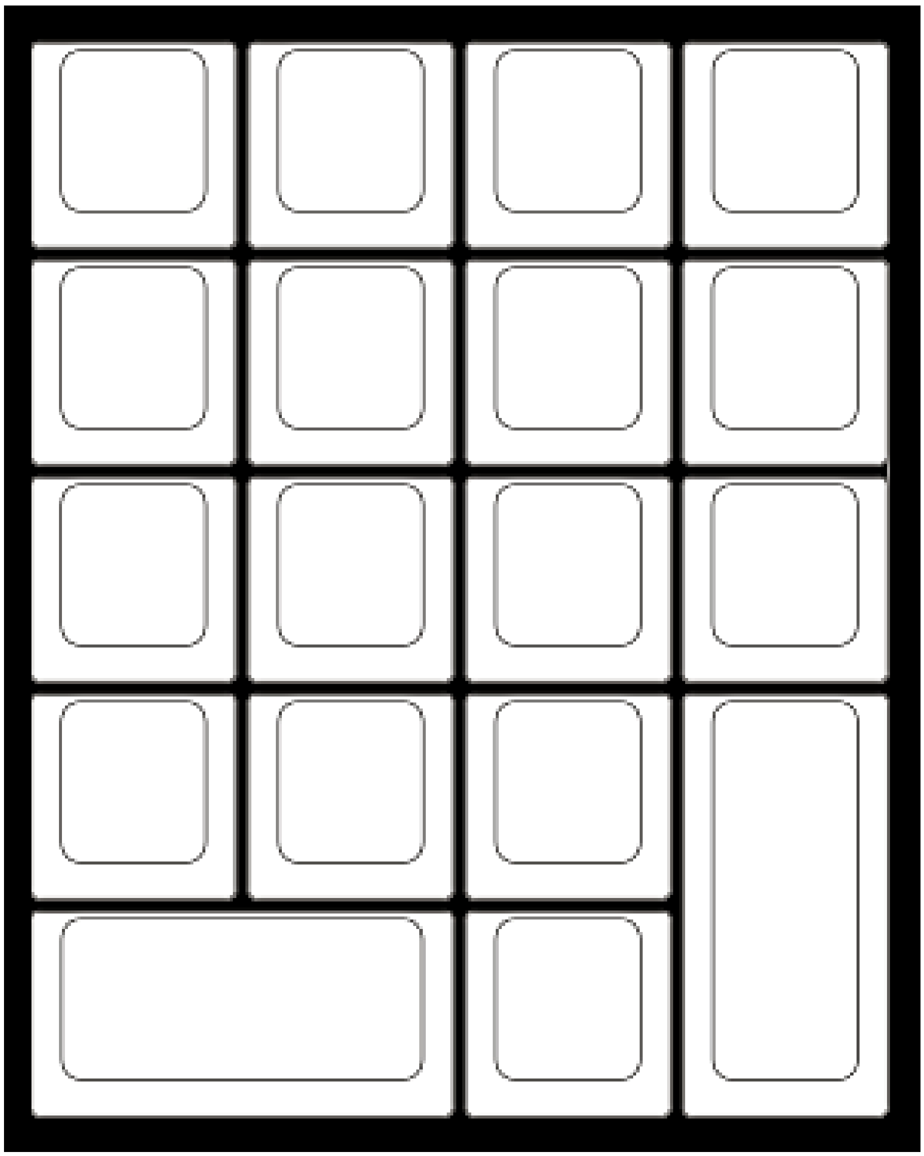


Figure 10: Dimensions for Individual Switch Holders



C Keyboard Layout

Figure 11: Visual Representation of the Key Layout



D Controller Code

```
// Teensy 3.0 has the debug LED on pin 13
const int ledPin = 13;
const int capsLedPin = 4;
const int powerLedPin = 3;

const byte ROWS = 5;
const byte COLS = 4;
int LAYERS = 12;
int MODES = 3;

bool toggleBind = false;
int currLayer = 0;
int prevLayer = 0;

//int wordLayer = 0;
//int latexLayer = 4;
//int asciiLayer = 8;

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

int currMode = 0; //side note: (always) int currMode == (currLayer % 4) / 4; <--
    this isn't true?

const int maxStrLen = 12; //define the maximum length of the strings in the layout
    array

const char* layout[][ROWS][COLS] = {
  {

  //layer 0 = normal
  {"sqrt","power","int","derivative"}, // '^' is defined as fn layer key, when held
      the the layer goes to the desired layer
  {"infinity","pi","sum","log"},
  {"forall","gtrthan","equals","plusminus"},
  {"mu","delta","theta","NULL"},
  {"caps","NULL","union","shift"}
  },{

  //layer 1 = shift layer
  {"NULL","^"},
  {"NULL","#"}
  }

};

byte row[ROWS] = {15,19};
byte col[COLS] = {20,23};

int key[] = {0,0,0,0,0,0};
char mod[] = {0,0};



void setup() {
  // initialize the digital pin as an output.
  pinMode(ledPin, OUTPUT);
  for (int c = 0; c < COLS; c++){
    pinMode(col[c], OUTPUT);
  }
  for (int r = 0; r < ROWS; r++){
    pinMode(row[r], INPUT);
  }
}

// This function will take keypresses passed to it (in the form of a char, for no
    particular reason)
// and add them to set of six keys that will be passed to the computer when Keyboard
    .send_now() is called.

// Basically, this collects the currently pressed keys and stores them until they
    can be passed to the computer.
void setKey(char keypress){

  // Look for unused keys in the buffer
  int i, j;
  for(i = 0; key[i] != 0; i++){}
  for(j = 0; mod[j] != 0; j++){}

  // Catch Modifiers
  if(keypress == 176){
    mod[j] = KEY_LEFT_CTRL;
  }
  else if(keypress == 177){
    mod[j] = KEY_LEFT_ALT;
  }
```

```
  else if(keypress == 178){
    mod[j] = KEY_LEFT_SHIFT;
  }
  else{
    key[i] = keypress;
  }

  if(holdKey('^')) // Prevent setting layer key into set_key or set_modifier
    return;

  // Hold keypresses in buffer
  Keyboard.set_modifier(mod[0]|mod[1]);
  Keyboard.set_key1(key[0]);
  Keyboard.set_key2(key[1]);
  Keyboard.set_key3(key[2]);
  Keyboard.set_key4(key[3]);
  Keyboard.set_key5(key[4]);
  Keyboard.set_key6(key[5]);

}

// This method sends the depressed keys and clears the buffer.
void sendKey(){

  Keyboard.send_now();
  clearBuffer();

  Keyboard.set_modifier(mod[0] | mod[1]);
  Keyboard.set_key1(key[0]);
  Keyboard.set_key2(key[1]);
  Keyboard.set_key3(key[2]);
  Keyboard.set_key4(key[3]);
  Keyboard.set_key5(key[4]);
  Keyboard.set_key6(key[5]);
}

// Helper function to clear the buffer
void clearBuffer(){

  for(int x = 0; x < 6; x++){ key[x] = 0; }
  for(int x = 0; x < 2; x++){ mod[x] = 0; }

}

// Detects when a key is held down, returns true if held down, false if not
bool holdKey(char keypress){

  if(key[0] == keypress ||
     key[1] == keypress ||
     key[2] == keypress ||
     key[3] == keypress ||
     key[4] == keypress ||
     key[5] == keypress){
    return true;
  }

  return false;
}

// Toggles between two layers, the curret layer and desired layer
void toggleMode(){

        //if both shift and caps held
        currLayer = (currLayer + 4) % 12;
}

// Macro sequence
void setKeyMap(const char* keypressed){
  // Modifiers:
  // KEY_LEFT_CTRL = 176
  // KEY_LEFT_ALT = 177
  // KEY_LEFT_SHIFT = 178
        if(strcmp("caps",keypressed) == 0){ // caps toggle added to setKeyMap
                currLayer = currLayer + 4 * (currLayer % 2) - 2; //if already on a
                     caps layer, AKA an even layer, caps toggles off, otherwise on
        } else {
    int len = strlen(keypressed); //get the length of the string
    int i = 0;
    for (i = 0; i < len; i++){ //iterate through each character in the string
      if(i>5){ //can only send 6 keys at once
        sendKey();
        clearBuffer();
      }
      setKey(keypressed[i]); //set the key equal to this character
    }
        }
        sendKey();
}

// Goes to desired layer when keyHeld is pressed, returns to previous layer when
     released
```

```
void holdLayer(char keyHeld, int desLayer){

  if(holdKey(keyHeld)){

    if(!toggleBind){ // Saves the previous layer, using boolean to prevent updating
        prevLayer more than once
      prevLayer = currLayer;
      toggleBind = 1;
    }

    currLayer = currMode + desLayer; // Desired layer
  }

  else{

    if(toggleBind){
      toggleBind = !toggleBind; // Resets boolean
    }

    currLayer = prevLayer; // Returns to previous layer
  }
}

void loop() {

  for (int c = 0; c < COLS; c++) {
    digitalWrite(col[c], HIGH); //drive each column high one by one
    for (int r = 0; r < ROWS; r++){
      if (digitalRead(row[r])){ //check if each row is high, one by one

          // 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
                .
          }
      }
    }
    digitalWrite(col[c], LOW); //reset the current column to zero
  }
  /*TODO: FIX THIS
   * if (holdKey(shiftKey) && holdKey(capsKey)){
          toggleMode();
  }else{
        holdLayer('shift', shiftLayer); // Checks if shift is held and if so, moves
            to shiftLayer
  } */
  delay(5);
}
```

E Survey Results

Mathematical Keyboard Survey

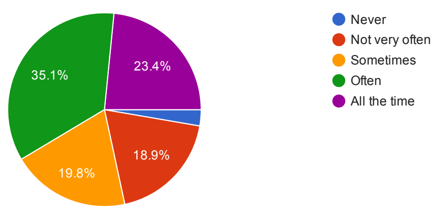
QUESTIONS RESPONSES 111

111 responses

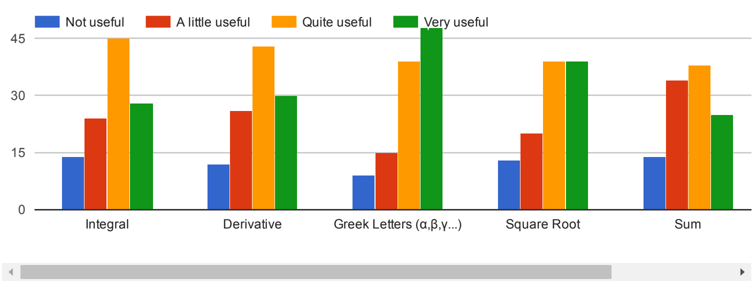
SUMMARY INDIVIDUAL

Accepting responses

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

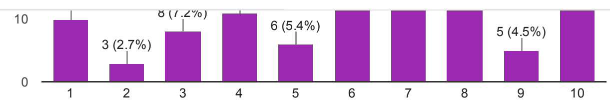


How useful would you find keys for these symbols?

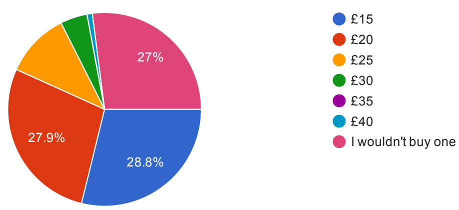


Mathematical Keyboard Survey

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.

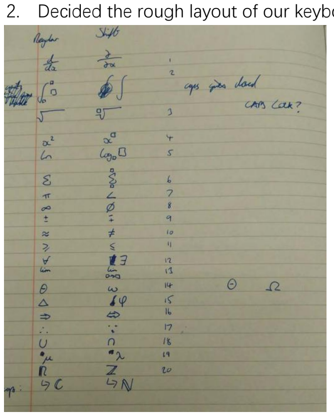
- Work for this week

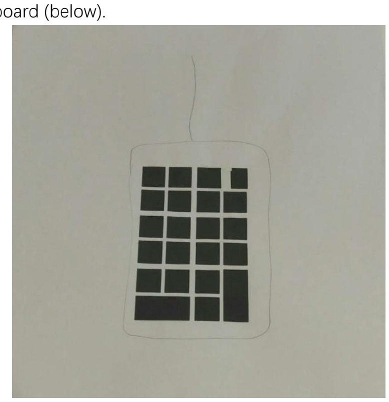
1. Research on what the most commonly used math symbols are.
2. Come up with ideas about the layout.

- 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

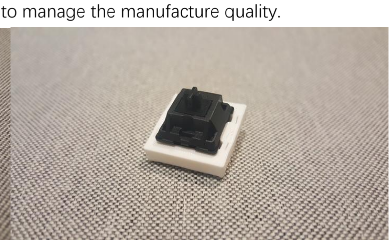
  4. **Component purchase**: we really need to order components. We will consult Dr. Perea about it.
  5. **Manufacturing**: we shall try to 3D print the key caps.
  6. **Interim report and website**: Interim report waiting for the guideline lecture. And Xav is willing to build up the website.
- Burgess 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

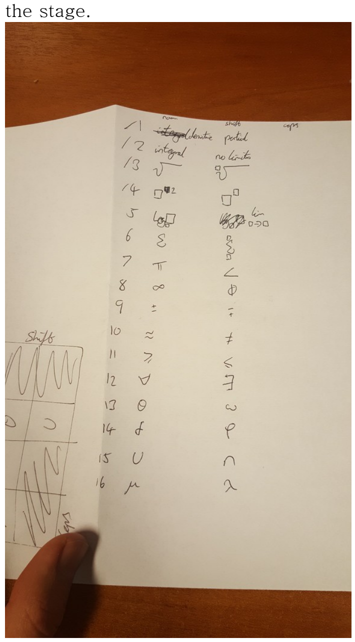
- 2PM online
- Recorder: BurgessXuWang
- Attendance: group members
- We realized that the interim report task distribution we have from last group meeting is not consistent with the **Assessment Guideline**. We therefore made adjustment based on our distribution so far and came up with an updated task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

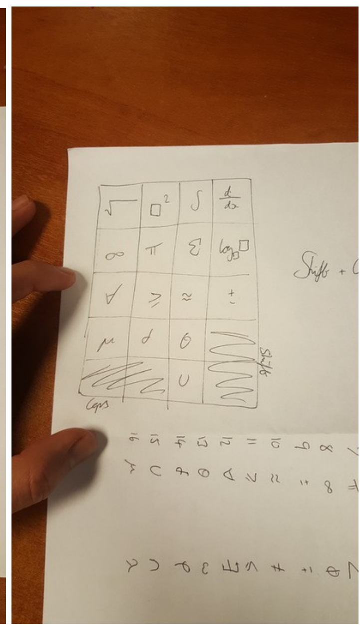
interim report task distribution		
1	cover page	Orion(already done)
2	contents page	Orion(already done)
3	abstract	Mariam
4	introduction/background	Clive,Orion(brief group introduction)/ Mariam(design intro and background)
5	design specification	BurgessXuWang, Xav
6	concept designs	BurgessXuWang, Xav, 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

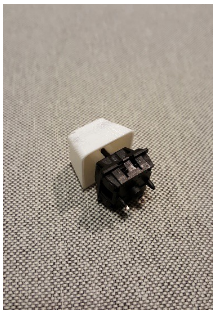
20170126 meeting

- 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 12: Table of Total Expenditure

Order History			
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
White LED (NSPW500DS)	5	£ 0.50	£ 2.49
Midas 8x2 LCD display (MC20805A6W-GPR-V2)	1	£ 5.68	£ 5.68
Total			£39.17