ENEL300 Group Design & Build

UCFunKit Redesign

Jeremy Roberts – jro162 - 11284438

Benjamin Stewart - bst72 - 16055101

Kei Carden – kca77 - 75535091

Executive summary

The goal of this project was to redesign the UC Fun Kit (UCFK) used in ENCE260. Talking to ENCE260's course coordinator Richard Clare, we developed some key points that needed to be addressed in a redesigned version of the UCFK. Basically, he wanted to keep all the existing features, but he wanted to expand the boards usability beyond just his course.

We decided the best way to solve this problem would be to create a base board that would house the MCU and then create stackable shield like boards to create the functionality. After the base board was designed, we focused on creating two additional boards, an audio board and a games board. The games board captured all the features of the previous UCFK while adding some extra features to allow students to create more complex projects. The audio board was designed as an example of a way our project could be used outside of ENCE260, teaching students about signal processing like what they might learn in ENEL220.

After creating schematics and PCB layouts for these three boards, we were able to get prototypes developed. The first version (V1) of out designs had major bugs, causing us to make some minor changes to all the boards. We were able to fix the PCBs creating a fully working prototype showing the proof of concept.

Finally, we investigated what it would cost to manufacture our product. We looked at DigiKey to estimate the prices of our parts and then got a quote from JLPCB for each of the three boards we designed. At this point, we decided that if our product was to be mass manufactured, we would need to change our components to be surfaced mount rather than through hole mounted as it was in our prototype. These prices were based on a mass order of 200 boards as that is roughly the class size and the prices include assembly. It was found that the MCU board could be manufactured for roughly $11.50, the audio board for $5.68 and the games board for $5.95 excluding the three screens.

1. Introduction

In the University of Canterbury's course ENCE260, students use a microcontroller called the UC Fun Kit (UCFK) to teach students to code microcontrollers with the programming language C. The UCFK features a small LED matrix screen, some buttons and a joystick which allows students to code a game effectively. The problem with the UCFK is that outside ENCE260, the board has no use. Our team aimed to remedy this problem, by splitting the functionality of the UCFK across different removable shields. This report goes into details about the requirements and specifications of the project, the design process, and the evaluation of the design.

The original UCFK uses these components to teach concepts taught in ENCE260. This design of the UCFK was reported by students to be limited in functionality. The redesigned fun kit described in this report will still serve the same purpose as the existing UCFK but with an improved educational experience. This is achieved by using an alternative microcontroller and the addition of extension boards. These extension boards sort the existing peripheral components into related modules, with extra components to broaden the range of uses. The extension boards can be seen in figure 1. These extensions are interchangeable and do not require other extension boards to function, stacking extension boards simply adds functionality to the fun kit.

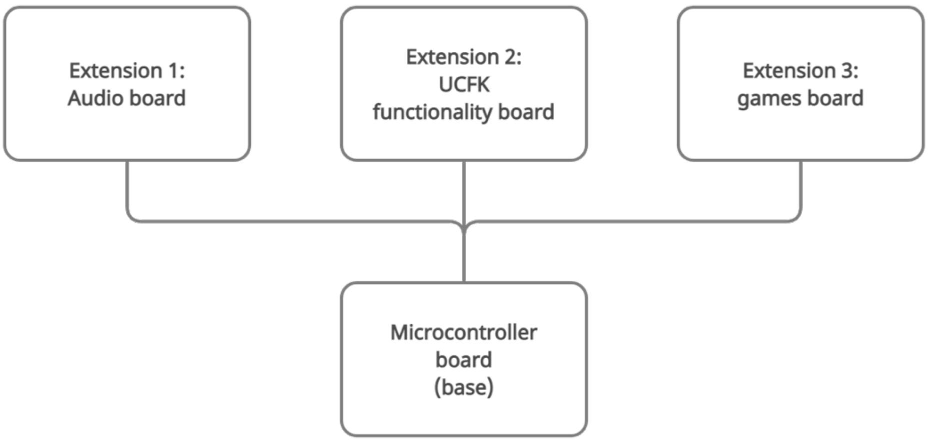


Figure 1: Block diagram describing the initial overview of the improved UCFK

2. Requirements

We created the requirements for this project based on our previous experience with the UCFK in ENCE260 and by speaking with lecturers who developed the original device. We also talked to the ENCE260 course coordinator Richard Clear about the short comings of the current design and what features he would like to see in an improved design. A few of the requirements obtained from the consultation with Richard Clare are as follows.

- Keep the current way of programming (using terminal)

- Allow the UCFK to be used for projects other than games (Reduce plagiarism)
- Upgrade the communication feature, using infrared communications.
- Cost (Might be hard with current shortage of microcontrollers would require a different micro which may make compatibility hard)
- Make more robust current device has issues with parts breaking off.

The main takeaway from our research was that the UCFK needed to be more versatile. To do this, we decided to focus our requirements on a modular design that would allow lecturers and individuals to make their own expansion boards for our product. This will allow our new product to be future proof as well as much more cost-effective as a complete redesign wouldn't be required when new features for different projects are desired. Taking this into account, the following requirements were developed, as shown in table 1.

No.	Section	Requirement	Hard/Soft
1	Financial	Total project cost under $50	Soft
2	Timeline	Submit requirement report by 8/8/2022	Hard
3		Order in parts for project by 25/8/2022	Soft
4		Submit PCB layout for manufacture by 26/8/2022	Hard
5		Start design report by 1/9/2022	Soft
6		Demonstrate prototype 22/9/2022	Hard
7		Submit Design report by 23/9/2022	Hard
8	Product	A modular design must be accounted for future proofing	Hard
9		Develop a board that can withstand 1 hour worth of typical use for 6 months.	soft
10		Develop a board that is mechanically able to withstand two year's use.	Soft
11		Demonstrate the ability to teach important embedded system concepts.	Hard
12		Provide a communication unit that is more less noisy than the current one on the used in the UC fun Kit.	Hard
13		The device should be backwards compatible with current UC games and software. The device should be able to able to run previous games developed in current software.	Hard
14		The device should be versatile and able to be used beyond ENCE260. The device should have at least two new features to ensure use beyond ENCE260 is possible.	Hard
15		Use the latest connection hardware for programming devices.	Hard
16		Use a faster microcontroller.	Hard
17		Can be programed with different programing IDEs.	Soft
18	Extension board 1		
19		The microcontroller should be able to take analog signal input through an ADC from the extension board's AUX connector.	Hard

19.1		The microcontroller should be able to produce an analog signal output with a DAC with an AUX connector	Hard
20		Ability to output more than one unique waveform.	Soft
21		Simple to use software module for students to use when sampling wave forms.	Soft
22		Simple to use software module for students to use when creating wave forms.	Soft
23	Extension board 2	Recreate the functionality of the current UCfunkit	Hard
24	Extension board 3	Ability to have higher resolution display for better games	Hard
25		Better ergonomics of the controls, user inputs have a clear intended function.	Soft
26		Ability to produce audible sound.	Hard
27		Ability to control the volume.	Soft
28		Ability to control the screen brightness.	Soft
29	Testing	The robustness will be met if components on the board do not break with average use.	Hard
30		Can the device be programmed in other programming languages.	Hard
31		Example of audio application running on the device.	Soft
32		Example of the expansion boards being used together.	Soft
33		An example of the backwards compatibly, running one of our last year's games.	Hard
34		The device must function after a drop of 1.5 metres.	Soft
35		The device should be able to run a program for at least 1hr without issues (i.e. not turn off).	Soft
36	Project	Must adhere to all University safety regulations	Hard
37		Must adhere to NZ wireless communication guidelines	Hard

Table *1*: Requirements categorized by section

3. Specifications

The following table outlines the specifications, these simply provide a more precise description of design choices that need to be followed when creating the prototype. Table 2 lists these specifications by section.

No.	**Section**	**Specification**
1	Product	Create stackable boards
2		Use transmitters to and achieve a noise level below XYZ
3		Should be able to run old ENCE260 games
4		Use the latest connection hardware for programming devices.
5	Extension board 1 (Audio)	Ability to construct waveforms using less than six pins on the microcontroller.

6		Students should be able to plug any signal in (0V to 5V & -2.5V to 2.5V).
7		Input to the analogue-to-digital converter (ADC) should use a 3.5mm Auxiliary connection as well as a screw terminal
8		Output from the Digital-to-analogue converter should be with a 3.5mm Auxiliary connection as well as a screw-terminal
9		Supply voltage for the DAC and ADC should be 5V from the microcontroller and should have minimal voltage - ripple less than 20mV.
10	Extension board 2 (Games)	OLED display for more complex projects.
11		8x8 LED matrix for learning the basics of a serial interface.
12		The original 5x7 display to keep the device backwards compatible.
13		A speaker to allow games with sound.
14		Volume control for the speaker.
15	Extension board 3 (MCU)	Use a USB-C connector to program microcontroller
16		New wireless transceiver that works without the devices needing to be next to one another (The original IR sensor had noise and was prone to breaking of the board).
17		A microcontroller that is compatible with AVR software. Would not require an overall in old games and programs.

Table 2: Specifications describing precise aspects of the design

Design Overview

Our design was focused around reducing the cost per unit for students as well as making the device a usable prototyping tool after being used in ence260. Unlike the original funkit which cost around $50 and once the course was over most people just have one laying around or have sold off at a large discount. The funkit was never usable after the course was over due to its already limited IO being taken up by the screen and navigation switch. To fix the flaws in the original design an expansion board system was devised. The concept was that students would only purchase the MCU board, and the university would loan out the expansion boards i.e games board audio board for the duration of the project similar to how course like ENCE361 handled their boards this year. This system also lends itself well to allowing more diverse projects in the future as new expansion boards can be designed.

IMIG of board stack

The main MCU board is to be brought by the students, like the UCFK. The extension boards, however, will be purchased by the university and loaned out to students when required for projects. That means that each course would have to front a big cost initially when buying the extension boards. However, in the long run, would save money while giving the students more freedom with their projects.

Our group has designed two extension boards for the MCU board, an audio-based board, and a games board. Using the MCU board as a base, the games board could allow students to create projects for a variety of screens. The main screen, the 8x8 display, allows the students to learn about shift registers and the basic serial interface. An OLED display has been incorporated to allow for more complex projects. Finally, the board also has a piezo speaker with a volume slider for playing music or making sounds.

To ensure backwards compatibility, this board also features a 5x7 display, the same as what would be found on the UCFK. This display would only need slight changes to the PIO and system header to work compared to the 8x8 main display. This display would allow for backwards compatibility with projects developed on the UCFK. The games board also features the same four-way navigation stick switch and the buttons found on the UCFK.

How the design satisfies the requirements and specifications

One of the main requirements that we encapsulated in our requirements and specifications was the need for multiple boards. In the requirements, we specified that we needed 3 different extension boards. These boards would recreate the functionality of the UCFK and add the extra functionality that was specified by the interested parties. The main advantage of using the extension board design is the modulation it provides. The modules separate unrelated peripherals, which allows for an easier understanding of each major concept taught in the embedded systems section of the course.

Extension board 1: Audio Board

One of the extension boards designed to attach to the main microcontroller board is the Audio board. The aim of this board is to teach students about simple signal processing concepts. This board features an analogue-to-Digital (ADC) and a Digital-to-analogue (DAC) converter. These features fulfil the requirements in section 19 to have signal processing components that generate and read signals. The design also fulfils the specification to operate the DAC and ADC with a supply voltage of 5V. The connectors in the final design also feature the specified connectors for input and output; 3.5mm auxiliary and 1x2 screw terminal connection.

Extension board 2: Games board

Another one of the extension boards that was designed was a games board. The aim of this board was to replicate the UCFK's functionality with some additional features. This device has all the peripherals found on the original UCFK such as a 5x7 display, a four-way navigation switch and buttons. This will allow old ENCE260 projects to be played on the new hardware. This board would fulfil the requirements listed for the games and for recreating the original UCFK functionality

The device also takes analogue sound waves generated by the MCU board and outputs it through the speaker that's included. The device also includes another two displays, the OLED display and the 8x8 display. This helps create more functionality and gives our device more functionality outside of one ENCE260 project.

4. Design Rationale

The following sub-sections provide rationale for the design choices made while constructing the next-generation UC fun kit prototype. This provides justification for component selection with reasoning and comparison to alternatives parts.

4.1. Microcontroller Board

USB Port

A USB-C port was chosen over USB-Micro or what was used on the original funkit, the USB-Mini. This decision was both related to cost and ease of use. With the original funkit, the port used was outdated and not commonplace anymore; as not many people had the cable to use with the funkit they were required to buy one. By using the USB-C connector, it is more likely that students would not need to purchase a cable, lessening the financial burden on students. The USB-C is also very versatile, being reversable, it is convenient for students and makes the connection more simplistic. USB-C having a data rate of 10Gb/s also meant that it would be more than suitable for any application needed by the microcontroller.

This port was chosen to have though holes for the shield as this reduces the chance of the port being ripped off the board due to the extra rigidity. The cost will be based on buying units of 100 as the number of students partaking in ENCE260 will be from 100 to 250 most likely.

Type	Individual Cost Per 100 – 249 units	Components Links
USB-C	$0.61770	https://www.digikey.co.nz/en/products/detail/gct/usb4105-gf-a/11198441
USB-Micro	$0.33340	https://www.digikey.co.nz/en/products/detail/amphenol-cs-fci/10118193-0001LF/2785388
USB-Mini	$0.29700	https://www.digikey.co.nz/en/products/detail/amphenol-cs-commercial-products/GMSB0530112C1HR/13683270

Table 3: table to compare USB connectors

Table 3 illustrates that it would be more economical to go with the USB-Mini port. But as described above USB-Mini are old and becoming outdated. By choosing a newer port, it is less likely the design will become obsolete. USB-Micro is a suitable choice considering both the cost and future perspective. However, The USB-Micro connection cable is less common than the USB-C cable. It is more likely that students would already have access to a USB-C as modern devices commonly use them. Therefore, using the USB-C port would prove to be the most cost-effective solution.

MCU

The selected microcontroller for our prototype was the Atmega32U4, which is similar to the MCU used on the funkit, an Atmega32U2. The main difference between these two is that the U4 has a built in 10bit ADC and can be used with a 16MHz crystal to achieve better programs. The U4 also supports more I/O. The U2 has 22 pins for I/O and the U4 has 26, this allows for more complex expansion boards in the future as well as allowing expansion

boards to be used simultaneously without interference from another board. Another benefit of both microcontrollers is that they come with built in USB support reducing the number of components needed on the board. Due to the shortage of electronic components world-wide, finding an available microcontroller that fulfilled our needs while being compatible with the original funkit was difficult. The following table shows different potential MCU's availability. This table uses prices from RS components and Digikey so that ICs in sock could be used for comparison.
This table was made on the 20th of September so the stock values might have changed.

MCU	Individual Cost Per 100 –249 units	USB - Serial	Stock	IO
ATMEGA32U4-AUR	$8.620 RS components	YES	2,344 0 Digikey	26
ATMEGA32-16PU	$6.750 Digikey	NO	7,158	32
ATMEGA328P-MMHR	$2.565 Digikey	NO	9,680	23
ATTINY3217-MFR	$1.185 Digikey	NO has UPDI	1,789	22
ATMEGA32U2-AU	$3.435 Digikey	YES	278	22

Table 4: *MCU comparison table*

Microcontroller prices retrieved from the following links

- ATMEGA32U4-AUR - https://nz.rs-online.com/web/p/microcontrollers/1772992p
- ATMEGA32-16PU - https://www.digikey.co.nz/en/products/detail/microchip-technology/ATMEGA32-16PU/739771
- ATMEGA328P-MMHR - https://www.digikey.co.nz/en/products/detail/microchip-technology/ATMEGA328P-MMHR/3046315
- ATTINY3217-MFR - https://www.digikey.co.nz/en/products/detail/microchip-technology/ATTINY3217-MFR/9477737
- ATMEGA32U2-AU - https://www.digikey.co.nz/en/products/detail/microchip-technology/ATMEGA32U2-AU/2187167

When considering a microcontroller for the project we had a few basic requirements. It had to have an ADC, more IO ports than the U2, could run at 16MHz, be low cost, and a package size that would make the board size not to large (Over 100mm x 100mm). From this set of requirements, a table of microcontrollers that were considered are illustrated in table 4. Highlighted in green is the microcontroller used on the current funkit.

One other thing we had to consider was the fact the I/O amount listed above did not reflect the number of I/O pins that could be utilized by the user. As some I/O are used by the crystal or reset. When considering the ATMEGA32, it was found that the package was too large, this MCU meets all requirements and would be suitable if available in a smaller package. The ATMEGA328P, which is the same used on the Arduino UNO, would also be a suitable choice for I/O, cost, availability, and adequate package size. However, it was not ideal because the SMD package for ATMEGA328P did not meet the requirements laid out in the brief, we were to only to use SMD components that could easily be soldered. We ended up choosing the ATMEGA32U4 as we had access to it at the university. It also has a built in USB to serial interface reducing the cost. If the U4 was not available due to stock shortages, the ATMEGA328P would be a suitable option in future revisions.

MCU configuration

Some of the component values that were selected for our design came for the microcontroller data sheet. The microcontroller board schematic can be seen in the appendix. The two components were two 22R for d+ and d- of the USB lines and a 1uF capacitor that was need for the internal voltage reference. This capacitor was chosen to be film in V1 due to it being available in the component store and due to films properties when it comes to low ESR. This capacitor helps keep the voltage reference stable and reduces noisy better than if an electrolytic capacitor was to be used. For V2 a ceramic was chosen as it is cheaper and comes in a smaller package as well as having many of the same benefits as a film capacitor.

The microcontroller chosen has an 8MHz internal RC oscillator. However, we aimed to have a faster clock speed to allow for more complex programs; an external 16MHz crystal was required. For the crystal to be used, capacitors must be placed between the crystal and ground. The capacitors chosen for this were 22pF this value was selected using the crystal data sheet as well as using the schematic for the Arduino Leonardo.

Capacitors

From the brief we were required to use though hole components for as much of the design as possible. We opted to uses the smallest capacitors packages sizes where it was possible to allow for easy routing and PCB layout. The type of capacitors was ceramic as they have a low ESR which is beneficial for reducing noise making the microcontroller more stable.

It was also necessary to have a small Bulk capacitor as to help with larger current requirements. This was just a standard 10uF electrolytic capacitor. This value was selected by observing other people’s designs to see what was required. For V2 all the capacitors were changed to ceramic and to SMD packages with all the 100nF being 0603 and the 10uF being 0805. This was done to allow the board to be mass manufactured.

Buttons

With most conventional development boards, the reset button is pointing up wards and is not located near the edge, this makes it difficult to reset the device when a shield or expansion boards is used. To remedy this issue a right-angle button located close to the edge was chosen. As the button will be receiving lots of use and force, a through hole mounted button was selected. **Figure *2*** demonstrates this feature.

Figure 2: right angle button on MCU board

Transmitter

The chosen transmitter is a separate board as it can be difficult to make a reliable RF circuit and they are cheap to buy depending on where they are sourced form (AliExpress, eBay). The transmitter we are using is the nrf24l01 which outputs at 2.4GHz and is interfaced with SPI. The main reason for the selection of this module over others was that we had some available to use and most others we looked at would need to be bought from places other than Digikey.

Filtering

To improve reliability and to help with making sure the MCU is not damaged, filtering on the input voltage was done by adding capacitors and a ferrite bead. This filtering is required as there can still be switching noise form the USB port on the computer. The ferrite bead is used to attenuate high frequency, it is able to do this as it acts as a resistor that changes with frequency. The bead we chose was the same one found on the Arduino Leonardo. This is 30R @ 100MHz. The performance of the ferrite bead can be seen in the graph in **Figure *3***.

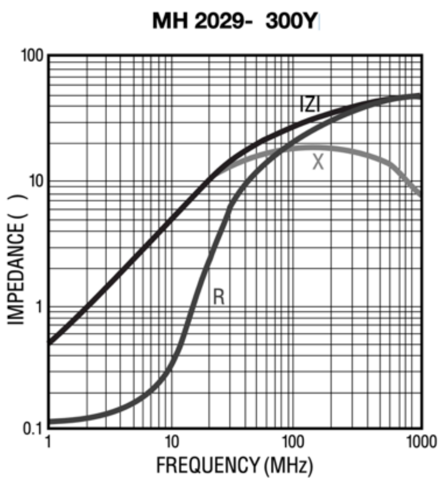


Figure 3: ferrite bead performance graph

Filtering was also added to the ADC voltage input as advised by the ATMEGA32U4 data sheet. This is done to help reduce noise levels further as any noise would result in inaccuracies in the ADC data. To save on cost the same ferrite bead was used with the same 100nF capacitors.

Board Layout

Component placement is quite important when it comes to microcontrollers. Microcontrollers current demands change constantly due to the switching taking place inside. Because of this, current is constantly being pulled from the power source and if it is too far away, the tracks act as inductors. This opposes these changing currents causing the voltage to no longer be stable on the microcontroller, which can lead to it not working as intended. To fix this issue, placing 100nF capacitors as close to the MCU VCC pins was done to supplement the microcontrollers current demands as well as reduce the stray inductance from the power source as this is now between the capacitors the MCU. To reduce the inductances further 1mm tracks were used for the voltage lines. The ground return path was made to be as large as possible by using the back side for ground only (Or as best we could). The empty space on the top was also used as a ground, this was stitched with many vias to reduce resistance and inductance. This ground plane also acted as a small capacitor helping to reduce noise. **Figure** shows correct decoupling capacitor configuration.

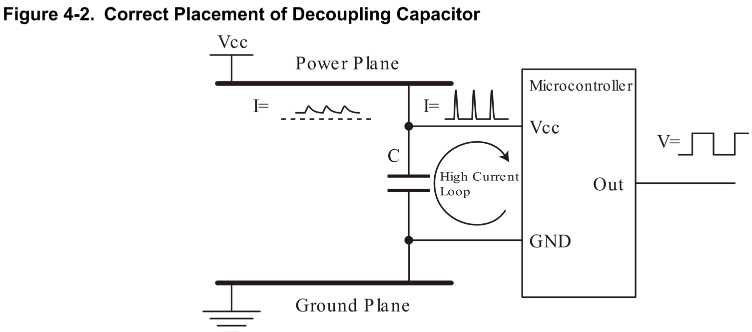


Figure 44: Decoupling Capacitor Placement from EMC Design Considerations *[2]*

4.2. Audio Board

The Audio board was kept simplistic in design only featuring one DAC and one buffer for the ADC as the course does not cover these components in much depth. The DAC and buffer take inputs/outputs through a 3.5mm Auxiliary cable or a 2x1 screw terminals for each. An MCP4921 integrated circuit is use for the DAC and a TL071 is used for the buffer functionality.

Digital-to-Analog configuration

A DAC was added to the design so that students could produce analogue signals with this extension board. The ATMEGA32U4 does not have any DAC output pins. So, an external DAC configuration was required to produce any analogue output signal. The component chosen was the MCP4921, and it was configured using the microcontrollers SPI pins. This DAC is suitable because the MCU can supply the needed voltage of 5V, its low noise properties, and a 12-bit resolution provides plenty of accuracy for its application. The price of the MCP4921 was $4.96

Filtering was added to the DACs voltage reference by using a LC low pass filter this was done to improve accuracy of the voltage output. This is not necessarily needed, as the DAC is most likely not going to be used in any projects that require high accuracy, however, doing this is good practices and the cost to add this is negligible.

Analog-to-digital configuration

The ADC was chosen to be part of the microcontroller selected to reduce cost and simplify the design. This design does not require a high-end ADC, the 10bit one that comes in most AVR microcontrollers were a suitable choice.

For the ADC an OP-AMP buffer was added to allow the student to connect any (with in the voltage range 0-5V) signal without needing to construct their own buffer externally. This buffer also protects the microcontroller from voltage spikes. The buffer chosen was created with the TL07 operational amplifier, this IC was chosen because of its cost (only $1.67) and low noise properties. A RC low pass filter has also been implemented on the output of the buffer to help reduce noise and make the signal more stable before being quantized.

In the V1 build there is no filtering on the ADC buffer input, this is not ideal as the ADC requires more attention to noise on its inputs. So, for V2 a filter has been added to the voltage input using the same style LC filter. The values for all the LC filters were chosen by using examples form other development boards as well as using the data sheets for the ferrite beads to determine whether the impedance would be suitable at the frequency of the noise.

4.3. Games Board

We decided to use three different displays to allow for complex and different programs to be made. OLED, 8x8 LED matrix, and the original UCFK display.

OLED

The choice for an OLED display was made due to it being relatively cheap and you get a much higher resolution allowing for projects such as a simple oscilloscope to be made.

5x7 Display and driver

Backwards compatibility with past games was an important requirement. Therefore, the 5x7 LED matrix found on the funkit was implemented. The configuration is mostly lifted form the original funkits schematic with component changes. The components that were changed were the values of the resistors. This was done as some of the individuals we talked to said that the original display brightness was to hash on the eyes. As such, the current was reduced using 220R rather than 100R resistors. The physical size of them was chosen to be as small as possible to allow for routing and getting all the other displays on the board. A normal resistor has a length of 6.3mm and a diameter of 2.5mm, the chosen resistors are

3.2mm long and have a diameter of 1.7mm. In version two these resistors would be changed to SMD eliminating this issue and making it more manufacturable.

The original design utilises MOSFETs for switching the columns on the matrix. As most MOSFETs that are though hole are not made for TTL or CMOS logic there was not much choice of transistors. We looked in to using BJTs but as they would require a base resistor, we choose to use the 2N7000 NMOS-FETs available in the component store. For V2 this is not an issue as there are many SMD logic FETs and layout is also a non-issue.

8x8 and driver

To add complexity and a larger space for graphics this size of display was chosen. To drive this display two 74hc595 8bit shift registers were chained together to give a 16bit shift register. This method was chosen as it only required three IO pins. This system would also allow students to learn a basic serial communication method as well as the normal raster display ideas taught. This shift register was chosen as we have experience with it as well as having two separate chips make the routing easer rather than one IC. The SMD package was chosen as it would make routing easier and take up minimal space on the board. For the same reasons these were placed on the back. The resistors selected were the same used for the 5x7 to save on costs. To connect the display to the board female headers were chosen as it would allow the display to be removed if not required.

Speaker

A speaker was chosen to be added so that students could play waveforms from the DAC or by just using PWM from the MCU. The requirements for a speaker were that minimal current was required to drive it as well as physically being small and easily mountable to the board as well as cost. From these requirements magnetic coil-based speakers were ruled out and piezo based one was chosen.

Volume

For the volume control a simple potentiometer-based approach was chosen. For the choice of potentiometer, it mainly came down to what would fit on the board and what looked best. One other thing to be considered was the choice between a logarithmic POT or a linear one. For audio, a logarithmic POT is usually used as this characteristic follows more similarly how our ears hear changes in sound. For the resistance value 10k was chosen as minimal current would be drawn by the piezo. And a linear one was chosen has this was not a critical audio device.

Navigation Switch

The selected navigation switch was chosen to being similar to the original funkits one, which was a 4-way switch with centre button. This allows for all directions as well as built in button. The options for navigation switches were not big, as such, **Table 54** is quite small. Other requirements for the switch were for it to be available to buy, easy to use and being able to handle rough use. The original switch on the funkit is quite similar to 5223 as it is part of the same family. The JS5208 was the only one that met all the requirements as it included some though hole clips as demonstrated in ***Figure*** . Other benefits were the size of the stick being 12.2mm tall (from PCB to top) this made it easy to use were as the stick on JS1400BFQ is about 4.5mm tall (from PCB to top). From this analysis the JS5208 was chosen.

Switch	Individual Cost Per 100 –249 units	Stock	Size L x W	Components Links
JS5208	$2.18310	622	12.4mm x 12.4mm	https://www.digikey.co.nz/en/products/detail/e-switch/JS5208/1739634
JS1400BFQ	$2.20430	4,911	9mm x 7.7mm	https://www.digikey.co.nz/en/products/detail/e-switch/JS1400BFQ/4028190
5223	$1.50000	12	10mm x 10mm	https://www.digikey.co.nz/en/products/detail/adafruit-industries-llc/5223/15277537 https://cdn-shop.adafruit.com/datasheets/SKQUCAA010-ALPS.pdf

Table 54: *Navigation Switch Comparison*

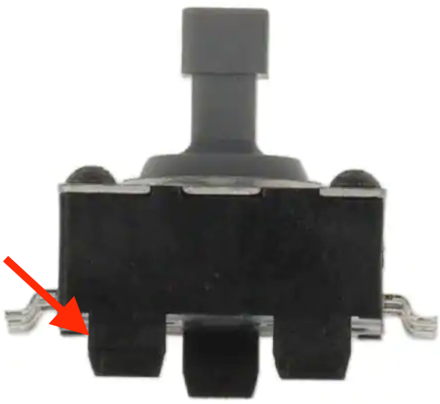


Figure 55: JS5208 Navigation Switch *[3]*

Games Board layout

The board was made to be larger than the MCU board as to make it more ergonomic for the user. Other ergonomic considerations were made by rounding the corners of the board with there being a larger radius on the bottom **Figure *6***. The position of the 4-way navigation switch was placed at the centre bottom of the board to make more space for the three displays as well as make it easy so use. To the right the piezo speaker and volume control are located to be out the way. The position of the screens has been made so that all three can fit next to one another. The location of the shift register for the 8x8 are located on the underside of the board to reduce the clutter on the top and as they are SMD they do not interfere with board connected **Figure *88***. With this board there was less consideration to the routing of tracks as all the lines were digital and not operating at high speed. However large ground planes were still used to help remove noise and reduce resistances and inductance in the return paths. The two ground planes were stitched via numerous vias to help keep the voltage differential across the board at ground potential.

As there are only so many IO pins available on the microcontroller utilising multiple pins for different tasks was used. This method was already used on the original funkit so that the display and navigation switch could be used. We were able to use the same for the displays as only one was expected to be used at one time. It would have been possible to use up all the pins but using less meant that expansion boards below could use these. Due to this the three pins used for the shift registers are used by the navigation/5x7 display.

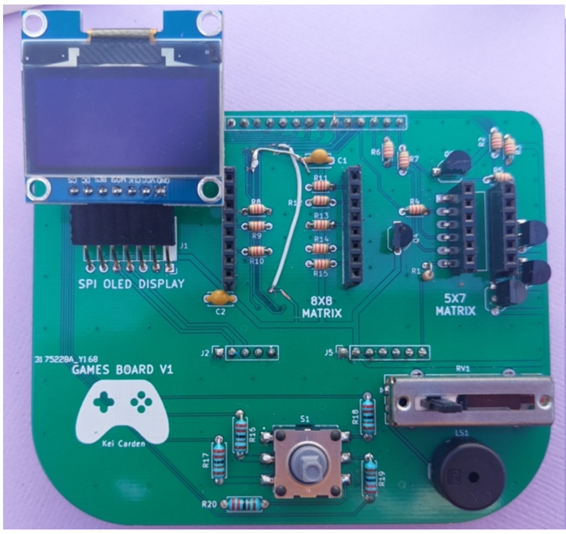


Figure 6: Games Board Top OLED

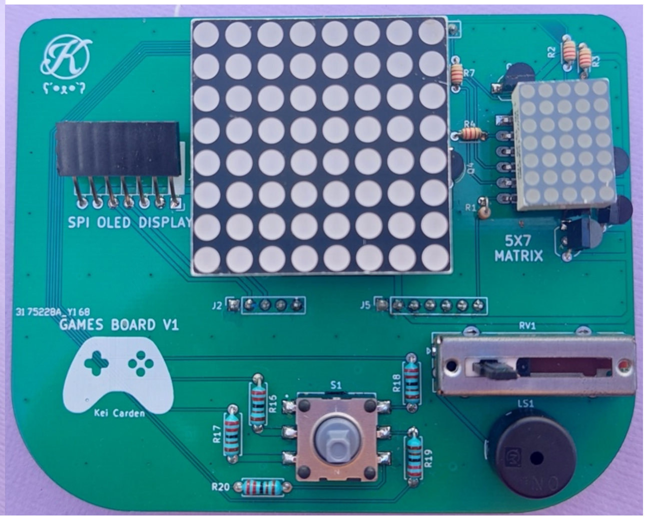


Figure 7: *Games Board Top 8x8 & 5x7*

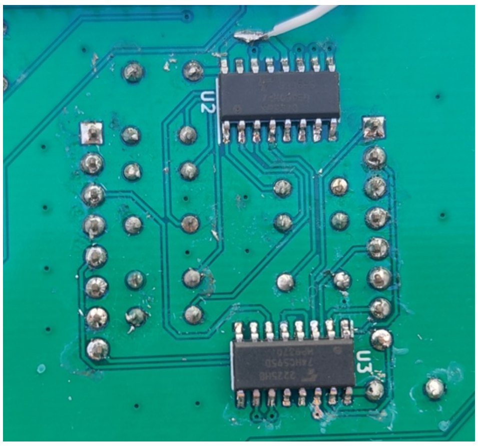


Figure 88: Games Board Bottom Shift Register

5. Recommendations and Lessons Learned

For this product to be viable for use at the university, changes to the design must be made so that it can be mass produced. The main change would be to switch from using though hole mounted components for resistors and capacitors as specified in our requirements to surface mounted.

Some basic research was done into the cost of manufacturing 200 MCU boards with JLpcb. A table has been constructed to demonstrate the cost per board of version 2 if the board was to be made with surface mount components (V2 has been made to use as many SMD components as possible as well as being changed to address the issues found in V1)

Component	Cost Per Unit	Cost Per 200 Units	Links to Components that could be used on V2
Atmega32u4	$8.62	$1,724.00	https://nz.rs-online.com/web/p/microcontrollers/1772992p
USB-C port	$0.6177	$123.54	https://www.digikey.co.nz/en/products/detail/gct/usb4105-gf-a/11198441
Ferrite Bead 2x	$0.0475	$19.00	https://www.digikey.co.nz/en/products/detail/bourns-inc/MH2029-300Y/2563323
16MHz Crystal	$0.15	$30.00	https://www.digikey.co.nz/en/products/detail/yic/16M20P2-49US/15648595
Right Angle Button	$0.4318	$86.36	https://www.digikey.co.nz/en/products/detail/apem-inc/MJTP1117/1795496
100n 0603 9x	$0.0083	$14.94	https://www.digikey.co.nz/en/products/detail/samsung-electro-mechanics/CL10B104KB8NNWC/3887593
22p 0603 2x	$0.0228	$9.12	https://www.digikey.co.nz/en/products/detail/kyocera-avx/06035A220JAT2A/563277
10u 0805	$0.033	$6.60	https://www.digikey.co.nz/en/products/detail/samsung-electro-mechanics/CL21A106KOQNNNE/3886754
1u 0603	$0.014	$2.80	https://www.digikey.co.nz/en/products/detail/samsung-electro-mechanics/CL10A105KA8NNNC/3886760
5k1 0603 2x	$0.01	$4.00	https://www.digikey.co.nz/en/products/detail/stackpole-electronics-inc/RMCF0201FT5K10/4439997
22R 0603 2x	$0.0044	$1.76	https://www.digikey.co.nz/en/products/detail/walsin-technology-corporation/WR06X220-JTL/13239156
1k 0603	$0.0044	$0.88	https://www.digikey.co.nz/en/products/detail/walsin-technology-corporation/WR06X102-JTL/13241138
10k 0603 2x	$0.0248	$9.92	https://www.digikey.co.nz/en/products/detail/stackpole-electronics-inc/RNCP0603FTD10K0/2240139
LED Green	$0.0656	$13.12	https://www.digikey.co.nz/en/products/detail/liteon/LTST-C191KGKT/386835
	Total Parts Cost	$2,046.04	
	Boards Cost	$159.50	

	Shipping Cost	$84.00	8-12 Business Days
	Total Cost	$2,289.54	
	Per Board	$11.45	This Cost can be rounded to $20 or $30 due to fluctuating prices

Table 5 SMD MCU Board Cost Analysis

The board cost was found using the JLpcb website to find a quote for 200 boards this also includes the cost for them to assemble it without components. From this basic look at getting 200 boards manufactured, the cost is around $20 to $30. This is considerably less than the current funkit which costs about $50 without the cable.

5.1. Audio Board Issues

DAC SPI

The Audio Board DAC implementation is incorrect in V1. The SDI pin on the MCP4921 was incorrectly connected to MISO pin on the microcontroller; It should connect to the MOSI pin. This issue was fixed with a manual correction by scratching off the MISO connection and soldering a wire to the correct pin in the back. This issue is fixed in V2 schematic.

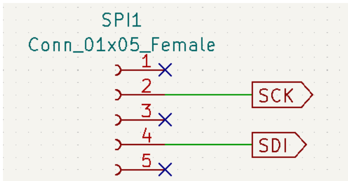


Figure 9: V1 SPI DAC Connection

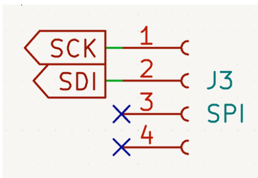
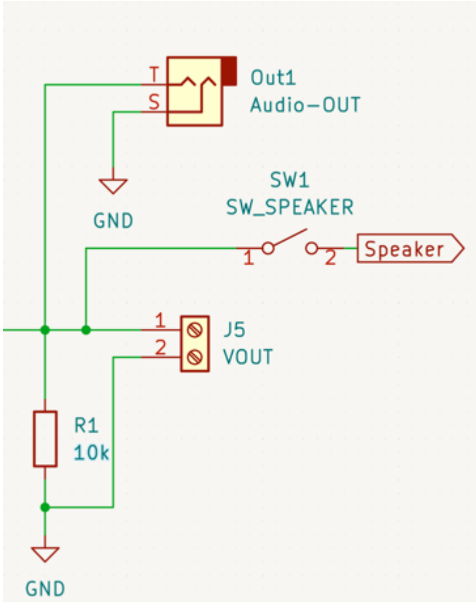


Figure 10: Revised SPI Header

DAC Out

With the original design V1 a switch was used so that a student could allow the DACs signal to pass to the piezo speaker located on the games board. Due to the piezo speaker chosen, this did not work as intended, the piezo best operates best at 4KHz so did not work well with lower frequencies that V1's DAC produced. So, in V2 this switch has been removed as changing the speaker to a better one would not be cost efficient. The piezo would remain because it works well with PWM based tunes produced by the microcontroller.
The new design shown in **Figure** 12. Has been adjusted so the jack can now be chosen as the output device or not. A capacitor has been used to remove the DC bias, allowing this signal to be connected to a device like a computer or amplifier. A value of 100uF was chosen as this large value would allow most frequencies to pass though without being attenuated.



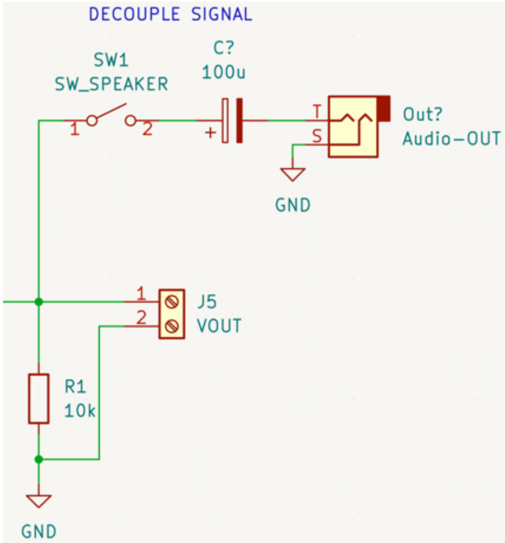


Figure *11*: V1 DAC Ouput Left

Figure *12*: V2 DAC Output Improved Right

Buffer

Audio board doesn't have negative rail. This is a problem for input waves that go into the negative voltage region. To fix this problem, a future design would lift up the signal with a resistor divider. This can be done on a breadboard to get the current version working.
This issue has been resolved for V2 as shown on in **Figure *13***. A capacitor has been placed in series with the signal coming from the jack as to decouple the signal from the DC side. After this capacitor a voltage divider has been setup to position the incoming wave form in the centre of the 5V rail thus making the signal operate entirely in the 0 to 5V region this allows input signals like audio from a phone (which operate from about -1V to 1V) to be quantised by the microcontrollers ADC (the signal would now be 0.5V to 3.5V). A switch has also been added to allow the student to choose between the AC bias input or the unbiased input depending on whether they are working with signal that require one or the other this also disconnects the bias voltage from the OP-AMP.

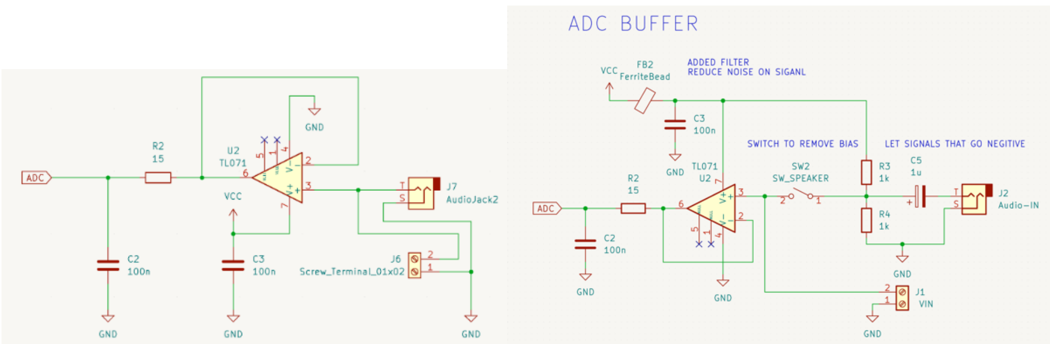


Figure 13: V2 New Buffer Input Design Right

Figure 14: V1 Original Buffer Design Left

5.2. Games Board Issues

OLED

Due to how the manual for the OLED shows the connections by using the same colour for the reset and VCC it was thought that pin 3 should be connected to VCC. The screen still works but for it to reset the power to the board must be fully cut off. Another minor mistake is the header is orientated incorrectly so for the screen to work it must be flipped 180deg.
Left is before right is how it should have been wired.

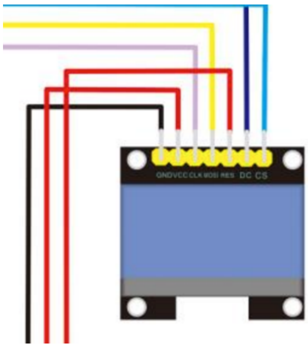
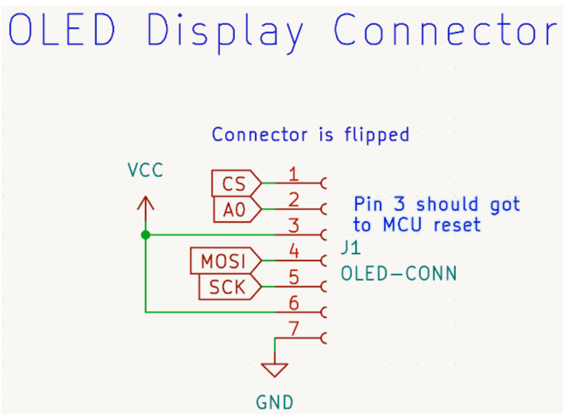


Figure 15: Reset Line in Red *[4]*



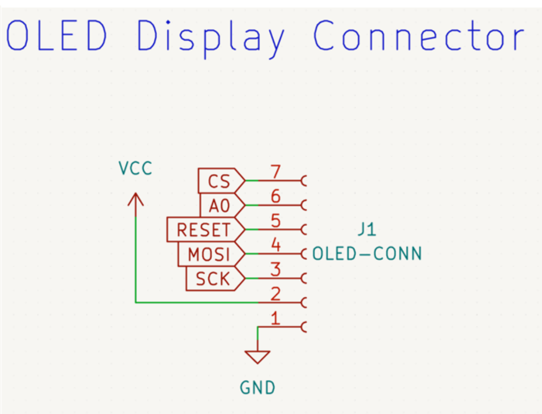


Figure *16*: V1 Incorrect OLED Wiring Right

Figure 17: V2 Corrected OLED Wiring Left

8x8 LED Matrix

When placing the connectors for connecting the 8x8 to the board they were placed two close together. This was due to not double-checking measurements taken from the display. A Veroboard adapter was made for testing it.

5x7 LED Matrix

This is a very simple issue the footprint selected was that of the wide variant resulting in the display not fitting and an adapter needing to be made. One other issue related to it's the driving circuit is that due to selecting NMOS-FETs it made it so the output of the MCU had to be inversed. The driver for V2 will utilise PMOS to fix this. It's also bad practises to use NMOS on the high side like we have done without additional boot strapping circuitry.

Shift Registers

When it came to testing each feature of the board it became clear that there was an issue with the shift register wiring. The Latch pin and Master Reclear pin were switched resulting in it not working. After cutting the tracks wiring it up correctly the screen works as intended.

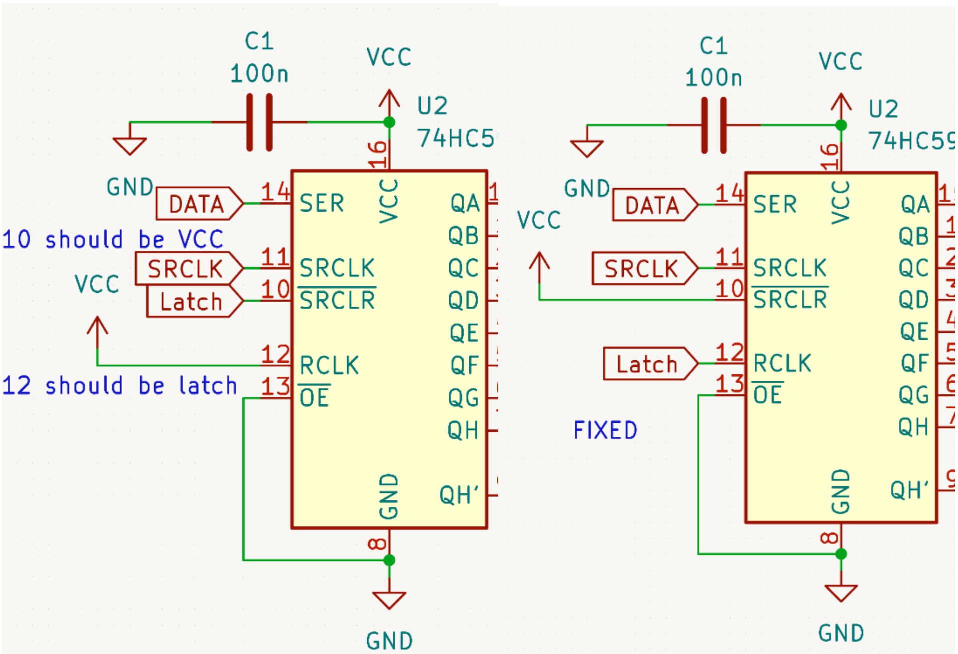


Figure 18: V1 Incorrect Shift Register Wiring Left

Figure 19: V2 Corrected Shift Register Wiring Right

Speaker

From testing it was found that the chosen piezo speaker was not suitable at reproducing frequencies below 4KHz as the sound output was hard to hear. But for sounds made via a PWM approach such as music made for the original funkit it worked quite well being very audible even with the volume set to ¼. For the V2 the same speaker will be used as it works well for PWM audio and the audio board has been reconfigured to not send its output to the speaker as we determined that the DAC would most likely not be used to produce music as that would be too complex to implement.

5.3. MCU Board Issues

USB-C

The port works well with a USB-A to USB-C cable however a USB-C to USB-C cable does not work with our board this is due to USB-C having the feature to not send power if not requested by the board. This has been fixed in V2 by adding 5k1 resistors on CC1 and CC2 as shown in **Figure *20***. This modification tells the computer to send power.

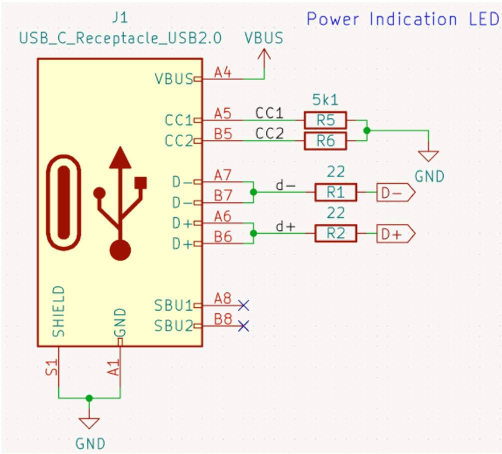


Figure 20: USB-C Port Reconfigured

One issue that was brought up during our demo was the robustness of the port. The chosen port was selected due to being though hole however the legs are not long enough to poke through a standard 1.6mm PCB as such when soldering solder might not flow down the hole enough to secure the port in place. In V2 a port with longer legs could be used but soldering the port better would be more ideal due to the costs of the port.

BOOT

After assembling one of the MCU boards it was found the button that we had added to pull this high was not required as such this has been pulled low via a 10k resistor. This also helps reduce the cost of the board due to the buttons being expensive per unit.

RESET

The original system for the reset flowed closely to how it was implemented on the original fun kit how ever after having issues with flashing the Arduino boot loader we did some research and found that this should be pulled high via a resistor. V2 has implemented this pullup resistor.

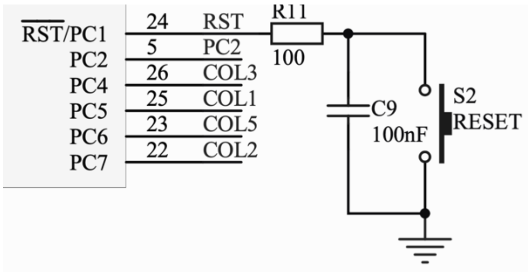


Figure 21: Original Funkit Reset

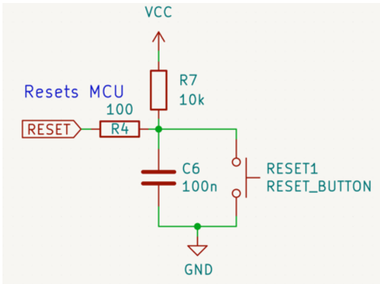


Figure 2222: V2 Reset Switch

ISP/SPI

Originally, we had decided that the only lines that would be important SPI and for ISP would be MOSI, SCK, MISO. But in the testing stage it was found that the MCU reset line would need to easy to access for ISP and was sometimes required for SPI devices such as the OLED display. So, in V2 this has been addressed by replacing the header that was made for SPI and some control signals with a four pin header as shown in **Figure 23:** *23* this was possible as the control signals that original went there were only for use with the transmitter so did not need to be sent up to other expansion boards.

Figure 23: 23 V2 New SPI Header

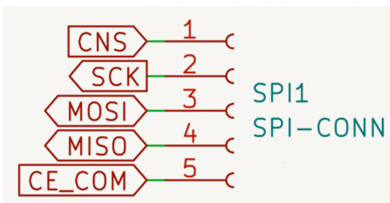


Figure 24: 24 V1 Old SPI Header

References

[1] Bourns, “Bourns® Inductive Components,” [Online]. Available: https://www.bourns.com/docs/Product-Datasheets/mh.pdf.

[2] Atmel, “AVR040: EMC Design Considerations,” [Online]. Available: https://ww1.microchip.com/downloads/en/Appnotes/Atmel-1619-EMC-Design-Considerations_ApplicationNote_AVR040.pdf.

[3] Digikey, “JS5208 Digikey,” [Online]. Available: https://www.digikey.co.nz/en/products/detail/e-switch/JS5208/1739634.

[4] Jaycar, “XC3728 Manual,” [Online]. Available: https://www.jaycar.co.nz/medias/sys_master/images/images/9677258489886/XC3728-manualMain.pdf.

6. Design evaluation

Completed tests

No.	Section	Specs tested	Description	Successful
1	**Games board**	1, 3, 4, 12, 15, 17	Backwards compatibility – Usability	✓
2		13, 14	Audio	✓
3		11	Games board display – 8x8 display	✓
4	**MCU board**	2, 16	Wireless communication using nrf24l01	✓
5	**Audio board**	10	Audio board – DAC sin wave	
6			Audio board - ADC	

Test 1: Backwards compatibility – Usability

How we tested

The first test we ran was the backwards compatibility test as this would allow us to test a few different features of the boards. Firstly, we stacked the games board onto the MCU board before running one of our old ENCE260 projects on the boards. Doing this would use the 5x7 display to show that our redesigned UCFK can replicate most of the functionality that the previous UCFK could do.

This test will be considered a success if the games board can display and run one of our ENCE260 projects from 2021 without having to change any functional code. We define functional code as any code essential to driving the game, we aren’t including pin configurations or refresh rates in this classification. We’re specifying this as in previous years the projects were required to incorporate the IR communication functionality. In our redesign of the UCFK, we have changed the way two boards can wirelessly communicate so to get previous years projects working some functional code would have to be changed.

Results of the test

This successfully proved we met 6 of our specifications. As seen below in Figure x, the game is running on the 5x7 Display.

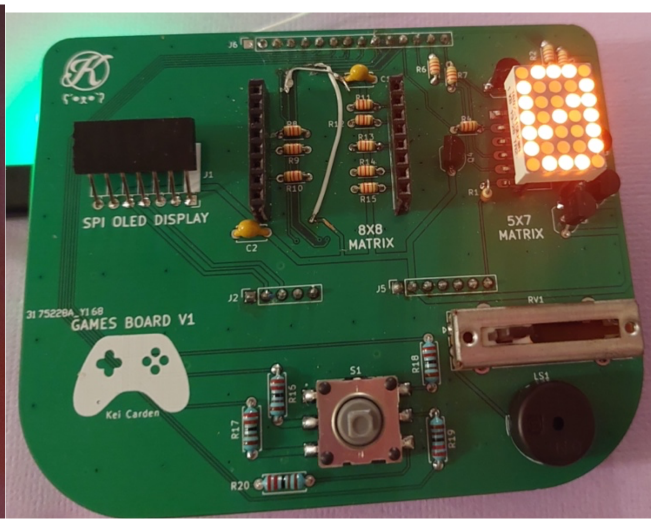



Figure 25: 25 Original FunKit Running Game
Figure 26: 26 Our FunKit Running the Same Game

No functional code was changed to get the game working on our project. However, some ports had to be changed as well as some timing related settings due to using a different microcontroller, as our project runs on a different refresh rate compared to the old UCFK. The joystick was able to manipulate the shapes and the game was running as expected. Therefore, our project can take old projects and get them working successfully.

Test 2: Backwards compatibility – Audio

How we tested

The game that was loaded up in test 1 also had an audio component to it. Therefore, using the speaker and volume slider on the games board the audio component could be tested by loading the previous project onto the board and turning on the sound component via code.

Results of the test

The test was a success. The volume slider allowed the music to change volume and the speaker outputted the music as expected.

Test 3: Wireless communication

How we tested

To test the wireless communication using the nrf24l01, two MCU boards were used. First, a basic Arduino wireless communication library was found as we just wanted to test the functionality rather than code a complicated module ourselves. Once an adequate library was found, we uploaded the transmitter code to one board and the receiver code to the other board. Finally, we tried to send a basic message from one computer to the other via the boards.

Results of the test

This test was a success. Using two MCU boards, we were able to program one of the boards as a transmitter and one of the boards as a receiver allowing us to send a basic message from one computer to the serial monitor of another computer. This can be seen below in Figure 27: *27* and Figure 28: *28* below. The code for this test was found online and the SPI pins and message were changed. The message being sent is 300 Project.

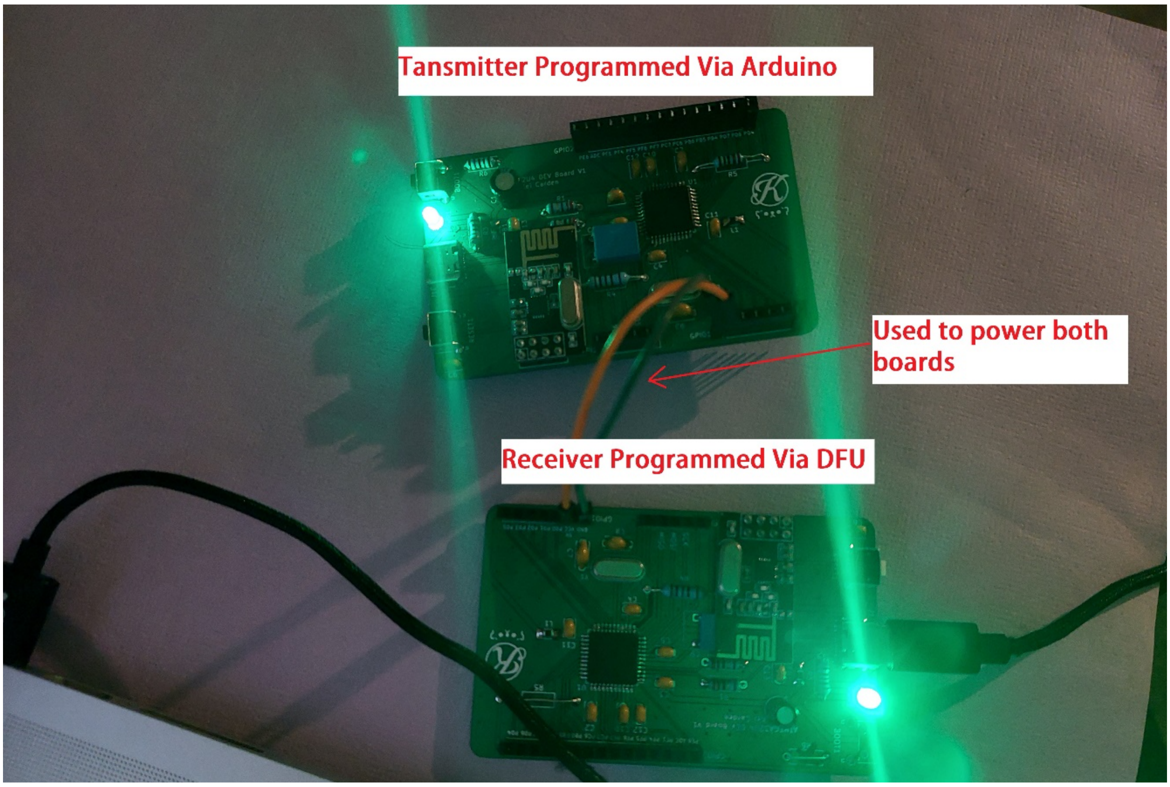


Figure 27: 27 Transmitter and Receiver

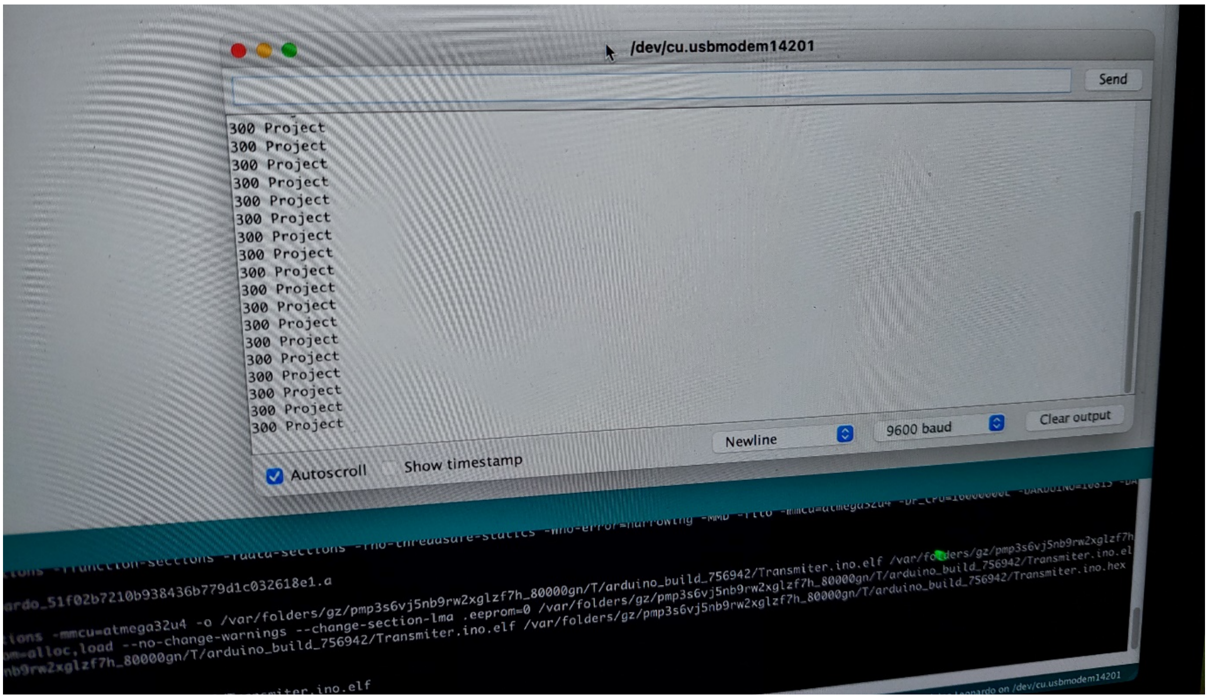


Figure 28: 28 Receiving Message

Showing can upload Arduino code
Shows up as Arduino Leonardo

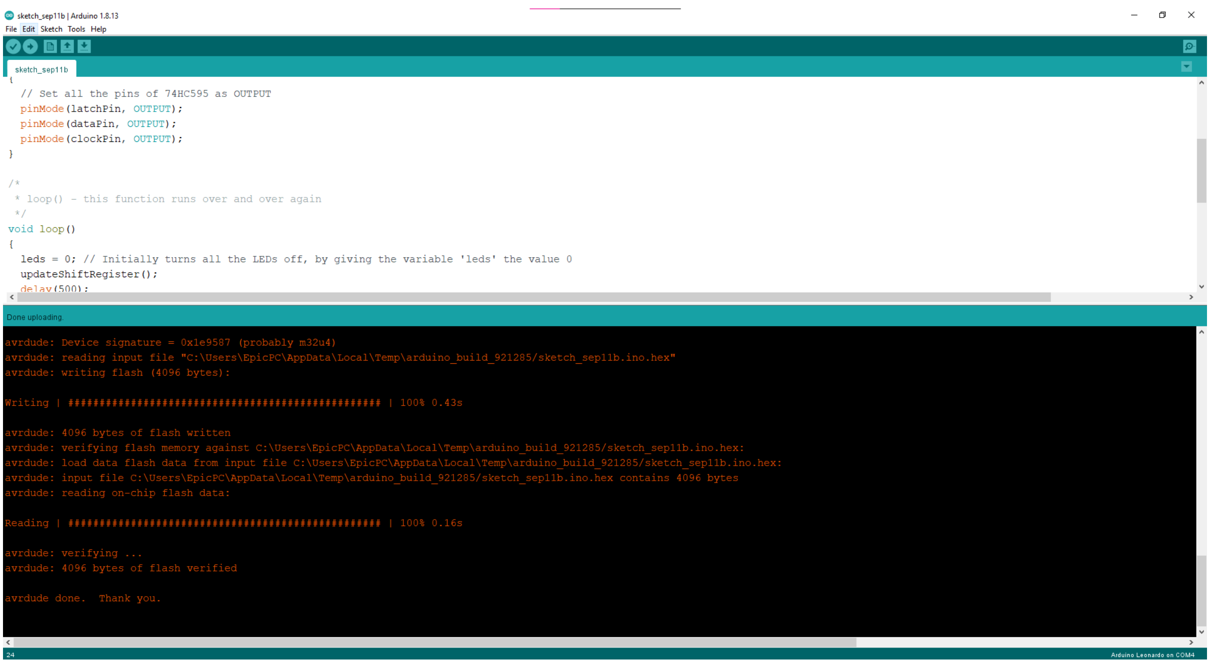


Figure 29: 29 Uploading Via Arduino IDE

Test 4: Games board - 8x8 display

How we tested

We coded an Arduino sketch with the basic goal of displaying some of the LEDs on the 8x8 display. This test would be a success if the correct LEDs displayed as intended.

Results of the test

Figure 30: *30* shows that this test was a success. We created a basic sketch in Arduino with the goal of displaying the four corner LEDs. This proves that the display works as intended and could be used for more complex problems.

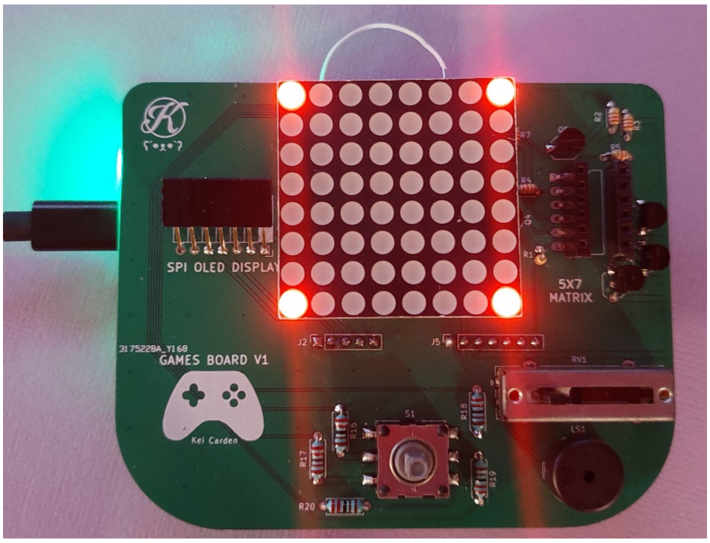


Figure 30: 30 8x8 Test

Test 5: Audio board – DAC and ADC

How we tested
The main parts of the audio board that need to be tested is the DAC and the ADC. To test the ADC, we generated a wave from an oscilloscope and input the signal into the ADC screw terminals. To confirm this test, we would display the results on the OLED screen to confirm that the correct signal was being read without distortion.

To test the DAC, a wave generator program was coded in the Arduino IDE and loaded onto the microcontroller. This wave was also measured with an oscilloscope.

Results of the test

As can be seen below in Figure 32: and Figure 33:, the ADC and DAC test was successful. The test successfully showcased two features. The OLED screen is displaying a the 2Hz sine wave coming from the signal generator. And the

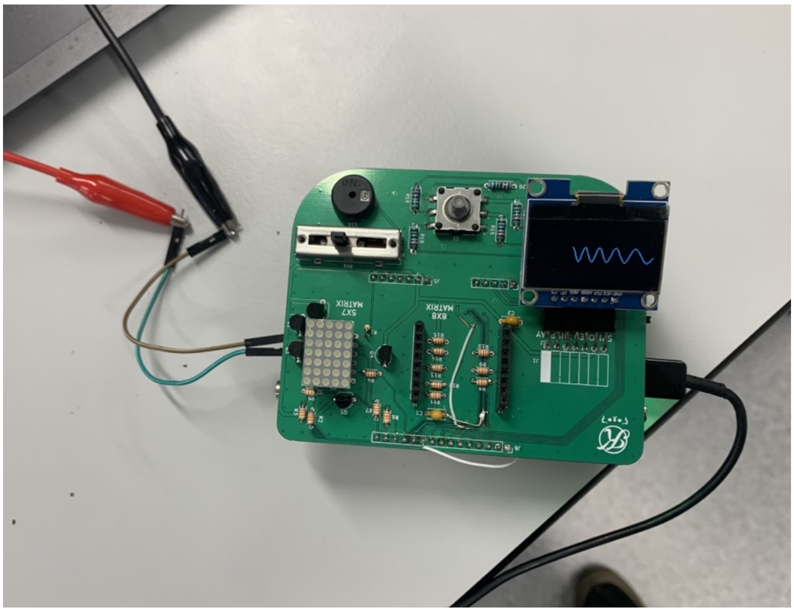


Figure 32: 31 Demo of ADC and OLED

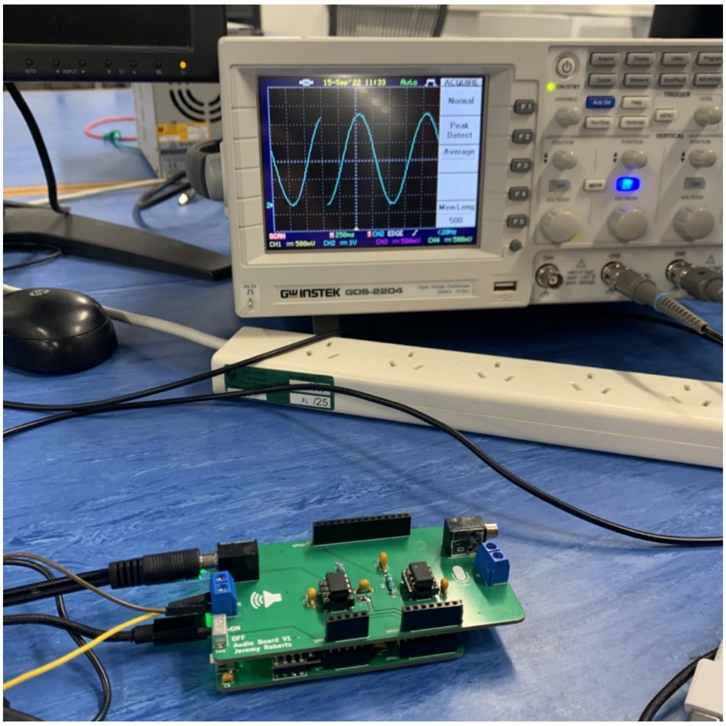


Figure 33: 32 Demo of DAC

Further testing plans

Further tests would revolve around creating demonstrations based on coding in C. Tests 1-5 are coded in Arduino. While this is good for showing the proof of concept, our mission is to redesign the UCFK. To achieve this, our redesign should be able to be coded in C so that our product can easily be swapped into courses like ENCE260. Some potential future tests are listed below.

Test 7: MCU board – Wireless communication using C

Test 8: Games board – 8x8 display using C

7. Conclusion

In conclusion, this document outlined the complete design process of creating a next generation UC fun kit. Through discussion with individuals involved with the course, various problems concerning the original design were identified. The original UCFK was said to have a limited educational experience. The new design took into account the robustness, cost and educational experience. From a defined list of requirements and specifications, the design we created. The design we created split the fun kit into one base microcontroller board with 2 optional extension boards, a games board to re-create the original fun kit functionality and an audio board to teach signal processing concepts. The first version of the board was compared with the requirements and specification, the issues were identified and hard requirements were fulfilled by version 2 of the design. While constructing the initial prototype, the first version contained a lot of issues. However, these could mostly be fixed through manual correction. These issues were noted and fixed in the design of version 2, these issues mostly consisted of incorrect pin configuration. A record of the cost was kept, it was calculated with mass production in consideration, the MCU board would cost $20 - $30, the audio board would $5.60 and the games board would cost $5.90. The final product was a relatively low cost which met all requirements and specification to be an improved UCFK.

References

[1] Bourns, “Bourns® Inductive Components,” [Online]. Available: https://www.bourns.com/docs/Product-Datasheets/mh.pdf.

[2] Atmel, “AVR040: EMC Design Considerations,” [Online]. Available: https://ww1.microchip.com/downloads/en/Appnotes/Atmel-1619-EMC-Design-Considerations_ApplicationNote_AVR040.pdf.

[3] Digikey, “JS5208 Digikey,” [Online]. Available: https://www.digikey.co.nz/en/products/detail/e-switch/JS5208/1739634.

[4] Jaycar, “XC3728 Manual,” [Online]. Available: https://www.jaycar.co.nz/medias/sys_master/images/images/9677258489886/XC3728-manualMain.pdf.

7. Appendix

Appendix: relevant source code and schematics

Games Board Schematic

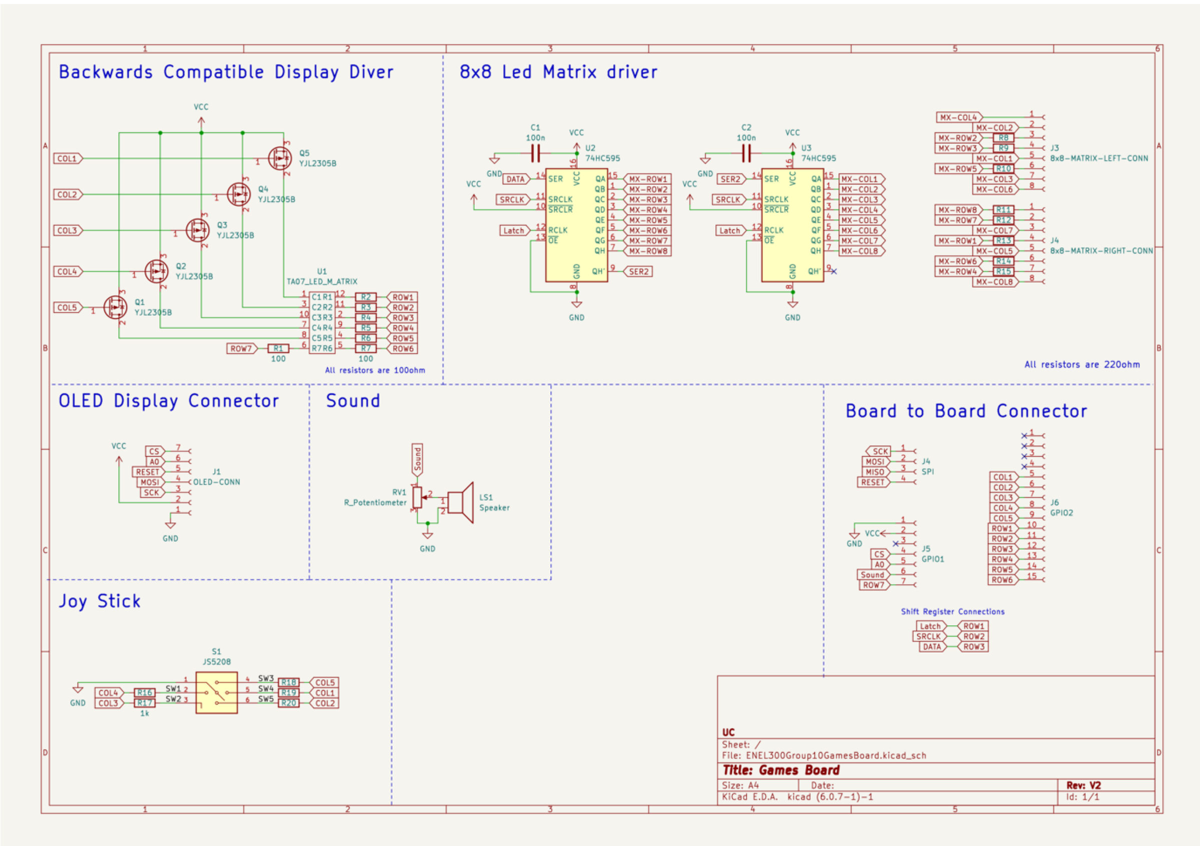


Figure 34: 33 Games Board Schematic V2

Audio Board Schematic

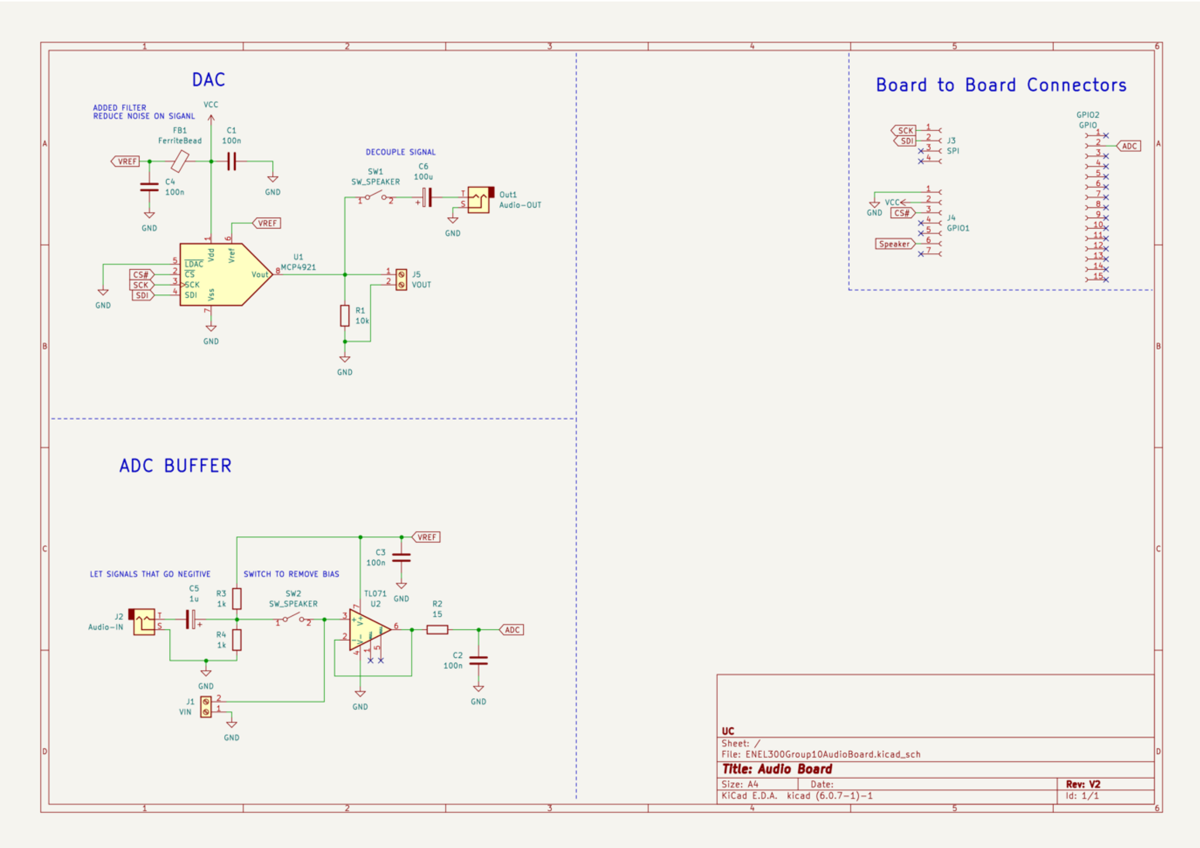


Figure 35: 34 Audio Board Schematic V2

MCU Board Schematic

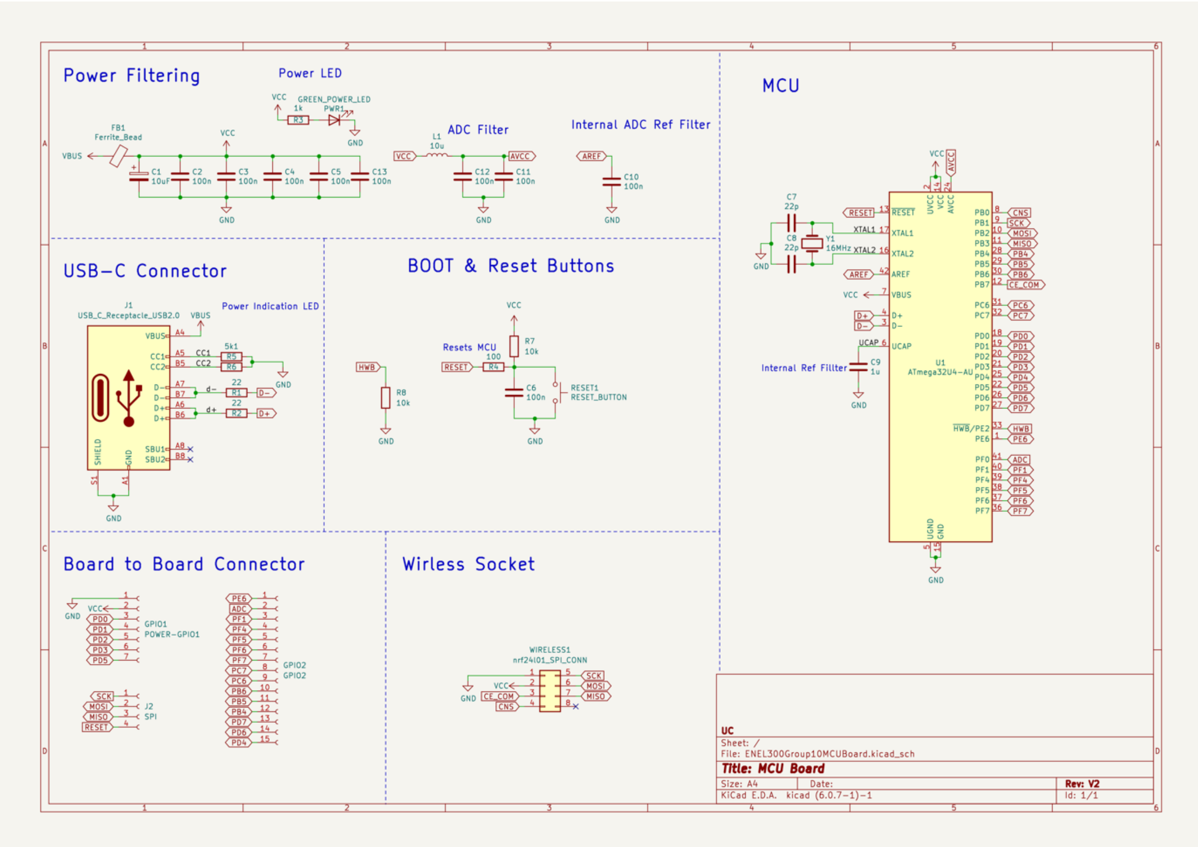


Figure 36: 35 MCU Board Schematic V2

8x8 LED Matrix Demo Code

```
int latchPin = 10;  // Latch pin of 74HC595 is connected to Digital pin 10
int clockPin = 9; // Clock pin of 74HC595 is connected to Digital pin 9
int dataPin = 8;  // Data pin of 74HC595 is connected to Digital pin 8

void setup()
{
  // Set all the pins of 74HC595 as OUTPUT
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}

void loop()
{
  updateShiftRegister();
  delay(500);
  for (int i = 0; i < 8; i++) // Turn all the LEDs ON one by one.
  {
    updateShiftRegister();
    delay(500);
  }
}

void updateShiftRegister()
{
   digitalWrite(latchPin, LOW);
   shiftOut(dataPin, clockPin, MSBFIRST, 129); //10000001
   shiftOut(dataPin, clockPin, MSBFIRST, 126); //01111110
   digitalWrite(latchPin, HIGH);
}
```

IO numbers for use with Arduino

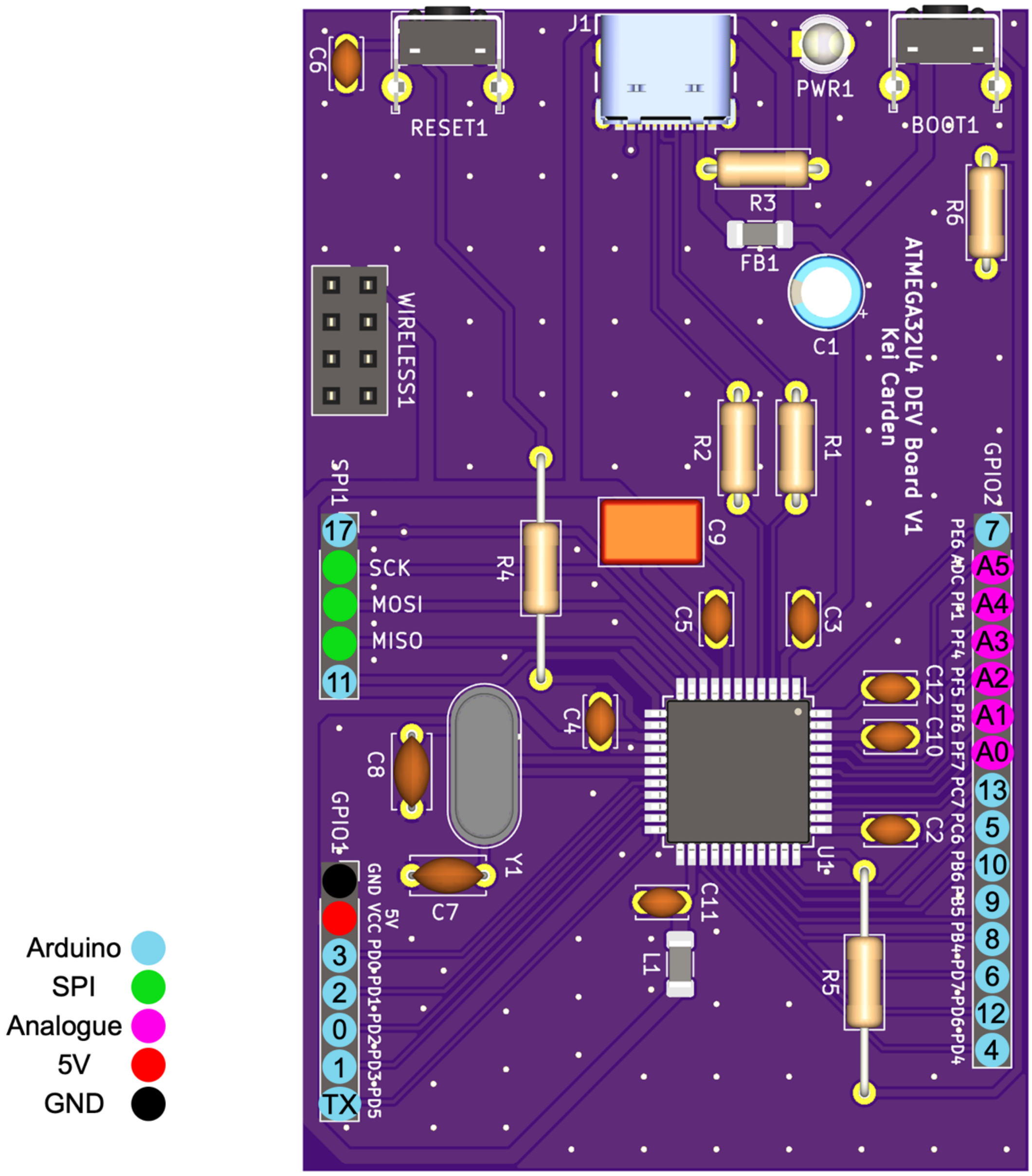


Figure 37: 36 V1 MCU Board Arduino Pin Out

Board and Components Cost

V2 Audio Board

Component	Cost Per unit	Cost Per 200 units	Links to Components that could be used on version 2
3.5mm Jack	$0.35	$140.40	https://www.digikey.co.nz/en/products/detail/globtek-inc/G-1360-N/16515806
Screw Terminal	$0.29	$114.80	https://www.digikey.co.nz/en/products/detail/w%C3%BCrth-elektronik/691137710002/6644051
100n 0603 3x	0.0083	$4.98	https://www.digikey.co.nz/en/products/detail/samsung-electro-mechanics/CL10B104KB8NNWC/3887593

1k 0603 2x	0.0044	$1.76	https://www.digikey.co.nz/en/products/detail/walsin-technology-corporation/WR06X102-JTL/13241138
Ferrite Bead	0.0475	$9.50	https://www.digikey.co.nz/en/products/detail/bourns-inc/MH2029-300Y/2563323
MCP4921 DAC	$1.98	$396.44	https://www.digikey.co.nz/en/products/detail/microchip-technology/MCP4921T-E-MS/716322
Right Angle Switch	$0.35	$140.80	https://www.digikey.co.nz/en/products/detail/te-connectivity-alcoswitch-switches/1825232-1/4021554
1u 0603	$0.014	$ 2.80	https://www.digikey.co.nz/en/products/detail/samsung-electro-mechanics/CL10A105KA8NNNC/3886760
100u CAP ALUM	$0.155	$31	https://www.digikey.co.nz/en/products/detail/w%C3%BCrth-elektronik/865080142007/5728067
15R 0603	$0.01	$2.46	https://www.digikey.co.nz/en/products/detail/te-connectivity-passive-product/CRGCQ0603F15R/8576268
MCP6486RT OPAMP	$0.24	$48	https://www.digikey.co.nz/en/products/detail/microchip-technology/MCP6486RT-E-OT/16624098
	Total Parts Cost	$892.94	
	Boards Cost	$ 159.50	
	Shipping Cost	$ 84.00	
	Total Cost	$1,136.44	
	Per Board	$ 5.68	

Table 6 SMD Audio Board Cost Analysis

Component	Cost Per unit	Cost Per 200 units	Links to Components that could be used on version 2
4-way switch	$2.18	$436.62	https://www.digikey.co.nz/en/products/detail/e-switch/JS5208/1739634
74hc595 2x	$0.23	$90.92	https://www.digikey.co.nz/en/products/detail/toshiba-semiconductor-and-storage/74HC595D/5879984
Piezo	$0.23	$45.04	https://www.digikey.co.nz/en/products/detail/murata-electronics/PKM13EPYH4000-A0/9859281
220R 0603	0.0044	$13.20	https://www.digikey.co.nz/en/products/detail/walsin-technology-corporation/WR06X221-JTL/13239083
POT 10k	$1.15	$230.28	https://www.digikey.co.nz/en/products/detail/bourns-inc/PTA1543-2010CIB103/3781161
PMOS 5x	$0.13	$125.40	https://www.digikey.co.nz/en/products/detail/yangzhou-yangjie-electronic-technology-co-ltd/YJL2305B/13911316

1k 0603 5x	0.0044	$4.40	https://www.digikey.co.nz/en/products/detail/walsin-technology-corporation/WR06X102-JTL/13241138
	Total Parts Cost	$945.86	
	Boards Cost	$ 159.50	
	Shipping Cost	$ 84.00	
	Total Cost	$1,189.36	
	per board	$ 5.95	

Table 7: Games Board Cost Analysis Excludes Screens

Websites used for design and demos

- https://cyberblogspot.com/how-to-use-mcp4921-dac-with-arduino/
- **https://www.arduino.cc/en/uploads/Main/arduino-leonardo-schematic_3b.pdf**
- http://arduino-er.blogspot.com/2015/04/display-waveform-on-mini-oled-with.html
- https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/?fbclid=IwAR2_p51OpKUbFSKHD9Aen4wMeOsUiCXmDdEOGM7pZG8rr8lmSTfOsSrWugY