

Diane Gayle

MA Commercial Music Producer

Level 7

Semester 2 (Development)

Module Code: ADAD7005_LON Discipline

Tutor: Ciarán Robinson

Student Number: 1808645

Assignment Content

Element 2b

4000 word essay

Code for Musical Instrument

INTRODUCTION

"Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board." (Sparkfun, 2019)

In this essay, I will discuss the making of a Granular Synthesizer, the principles behind it, I will also detail any issues encountered while making the Granular Synthesiser and how I addressed them. I will then discuss how it can be improved and how it compares to a commercial product of the same standing.

Making the Instrument

For my project, I decided to build a standalone version of an Arduino Granular Synthesiser. Originally designed by Peter Knight for tinker.it, this machine is targeted towards enthusiasts who are interested in electronic synthesis. (Volts, 2012)

The synthesizer is founded on the Arduino platform; this model uses a standalone Arduino that is positioned on a perma – board at the end of the creation process. "It uses granular synthesis techniques to generate a distinctive filter-sweep sound that had much more character than boring square waves. This synthesizer uses five controls to manipulate the sound that can be achieved. Peter Knight also explains that "Sound is generated by playing the same noise ('grain') repeatedly at very high speed. This merges into a tone that is an audible hybrid of the repetition rate and the original grain. It sounds quite similar to an oscillator with two resonating bandpass filters, although the different architecture means there are lots of additional interesting noises at parameter extremes." (Knight, 2013) The sound is made up of two triangular waves of adjustable frequency, and decay rate. Using the potentiometers, you can control the speed and repetition rates as well as manipulating the oscillators. This prototype uses 4.7 K ohm linear potentiometers.

Granular Synthesis is described as:

Many thousands of very *short* (usually less than 100 milliseconds) overlapping sound *bursts* or *grains*. The waveforms of these grains are often sinusoidal, although any waveform can be used. (One alternative to sinusoidal waveforms is to use grains of sampled sounds, either pre-recorded or captured live.) By manipulating the temporal placement of large numbers of grains and their frequencies, amplitude envelopes, and waveshapes, very complex and time-variant sounds can be created. (Burk *et al.*, 2019)

The grains collectively create clouds of sound, this method was achieved by composer Iannis Xenakis who recorded sound from an orchestra then cut up the tape and played it back in its remixed format. With this knowledge in mind, this machine sounded like a good instrument to make as I am always interested in noisemakers that I can use as opposed to using software and a mouse to make new sounds.

The purpose of following this tutorial was that I would learn how to build a circuit to run minimal code on a breadboard, without using an actual Arduino inside the enclosure I was about to build. The video provided step by step instruction to circuit construction. The lesson was provided by a man known only as Dave from www.notesandvolts.com.

Arduino on a Breadboard

Firstly, I connected some hook up wire connecting the positive rail, and then the negative as the breadboard is not internally powered. I made two small jumpers and connected one to the negative rail and one to the positive rail on the other side of the board. Next, I took a 10 uF powered capacitor and placed it on the rail, the part of the capacitor with the white stripe is the negative side, so that was placed in the negative rail and the other in the positive. I placed the 7805 voltage regulator in front of the jumper leads, and the 100 uF capacitor was put in front of the regulator, negative side first and the positive side to the left in front of pin one. A 1N4001 diode was then inserted to the pin on the 100 uF capacitor negative side first while the other side was put on the other side of the divide.

In the tutorial, the instructor had a powered jack that he was inserting into his breadboard and did not provide any instructions as to how to do it. As there were no instructions on how to do it, I found one on the www.learn.adafruit.com website and followed the instructions there. The tutorial demonstrated how to wire and solder a power adapter jack. (Brothers, 2018)

After following the tutorial on Adafruit I returned to the project and inserted the leads of my power connector into the breadboard, the live element was placed at the live leg of the diode, and the negative went to the negative rail at the end of the board. With the power plugged into my wall socket I tested the rails with my multimeter, the breadboard was indeed powered. I was getting 5 volts and was pretty happy that it was working as it should be.

The next stage of the tutorial involved placing the Atmega328 Microcontroller on the breadboard. The microcontroller is placed notch first, six pins down from the regulator. On pin seven of the microcontroller, I placed a jumper that went to the positive rail which is now tied to 5 volts on the rail. From pin eight I connected another jumper to the ground (negative) rail. Pins 20, 21 and 22 are also connected to the breadboard. The 16MHZ crystal is then placed in front of pins 9 and 10 of the Microcontroller. Then two 2uF capacitors are placed at both sides of the crystal to 9 and 10 and are connected to the negative rail. Next pin one of the microcontroller (which is the reset pin), is then connected to a 10K resistor which goes to the five volts which is positive on the breadboard rail. This setup is now the minimum circuit needed to run Arduino code on the breadboard.

To test the circuit, I placed the 220 Ohm resistor on pin 19 of the microcontroller and the other placed side just clear of it so that it is not connected to the microcontroller; then a lead was placed on the pin of the resistor and connected to the negative part of the rail. The breadboard was then supplied with power, on the video tutorial, the LED started to blink – he had already programmed his microcontroller with the blink

program. My LED, of course, was unlit, so then I had to figure out a way to get things going.

Arduino as ISP

The next thing to be done was to install the Arduino IDE program on my computer. Without having any knowledge of how the program worked, I went through a small tutorial (Thimbleio, 2017) and learned that in order to communicate with the microcontroller, you have to compile little programs called sketches. In this mode, you are using the Arduino as an In-Service Programmer. Using the Arduino as a testing ground, I uploaded the blink program to it. The LED on the Arduino Uno started blinking, I then tried to program the microcontroller on the breadboard as instructed on the www.notesandvolts.com tutorial; there was no response.

The Atmega328 microcontroller has memory called fuses that control how the chip communicates with the computer software. Burning the Bootloader sketch onto the microcontroller ensures that the fuses are set correctly. On the IDE Serial Monitor, I was getting a lot of error messages. Burning the Bootloader was proving to be difficult. I then spent many hours researching what it could be that was going wrong. I took to searching forums for the answers I was seeking. I googled the error message "avrdude stk500_getsync() not in sync resp=0x30" and found that it meant that there was a problem with the communication between the hardware on the breadboard and software on my computer.

I then found a website called DroneBot Workshop.com; there was an in-depth tutorial of how to get the code onto the microcontroller on the breadboard. There were three methods. Method one showed how to program the Atmega328 on the Arduino and then transfer it to the breadboard. I could not do this as my model of Arduino was what he described as the "Surface to Mount edition." (Workshop, 2018) Method two was the failing method I was trying to achieve. Method three gave details about using an FTDI adapter which I did not have.

After finding out that I may have the wrong model of Arduino Uno (SMD edition) I called up Pomironi where I purchased the Arduino from for some advice. Unfortunately, there was no one from the technical support team who could help. I was told to call back next week. Bearing all the problems that I was having in mind, I then set out to buy another Arduino in order to finish the project. With the new correct version of the Arduino Uno in hand, I continued with the making of my Granular Synth. After connecting it up to the breadboard and uploading the Blink sketch, the LED on the breadboard blinked. Unfortunately, I was still unable to upload to the Atmega328P microcontroller. Eventually, I programmed the Microcontroller on the Arduino Uno itself and replaced the one on the breadboard. At this stage, I had no way of knowing how this would affect the project as there was no communication with the breadboard and the Arduino software. I carried on regardless and soldered all the components to a Perma -Board as instructed by Dave at Notes and Volts.com.

MAKING THE ENCLOSURE

Using the drill guide provided by www.notesandvolts.com, I measured the holes needed for the potentiometers on the top of the enclosure. Using my drill and a step drill bit that I purchased from Screwfix, I drilled five holes into the top of the enclosure, measuring with a potentiometer to ensure I did not go too far and make the hole too big. The holes for the audio and power jack had to be measured out as there was no guide provided for them. By measuring the middle and height of the case, I divided the number in half and marked the centre. Then 35mm from the centre line as instructed by Dave divided it by half. It looked centred so I started drilling, again measuring the holes with the components to be used until I had a good fit. The enclosure is now ready to fit all the parts inside.

To connect the 1/4" audio jack, I had to first, identify which part of the jack was positive and negative. The tip is positive, and the sleeve is negative, with this in mind I soldered the audio jack. The positive wire was placed at pin 5 of the microcontroller and the negative wire placed in the negative rail beneath it.

The potentiometers came with little mounting tabs on the side, in order to lay flat in the enclosure, were cut off with side cutters and installed and tightened in the enclosure. The potentiometers were then all turned fully counter-clockwise and using a coin as a spacer, I installed the knobs. The potentiometers have three mounting terminals: positive which I have wired red, negative - green and the signal (middle) blue. The blue wire goes to the microcontroller. (volts, 2014)

On the microcontroller pins, 23 to 27 are the analogue inputs, on these pins the wires from the potentiometers will be connected. Firstly, I connected the positive terminals with 22-gauge wire, I did not have stranded wire, so I used solid core a little bit fiddly but otherwise okay. On the last potentiometer, a longer wire was connected, and this one will go to the positive rail on the Perm - Board. The negative terminals were connected and soldered in the Same way, with the last pin connecting to the negative rail on the Perma -Board. Next, each signal terminal is connected with longer wires as these will be connected to their respective pins on the microcontroller.

With the potentiometers wired up and soldered, I then fitted the audio jack into the side panel of the enclosure and repeated the same process with the power jack. I then covered the bottom of the board with double-sided tape making sure none of the soldering and parts underneath touched the enclosure. With that, I put it all together and plugged it in. I got nothing - the Synth had failed to launch.

Parts Used:

- 1 - 9v DC Power Adapter - 500ma or greater
- 1 - DC Power Socket (2.1mm or sized to fit the plug of your power adapter)
- 1 - LM7805 - 5 Volt regulator
- 1 - 1N4001 - rectifier diode
- 1 - 100uF (Micro-Farad) electrolytic capacitor - rated at 25v or more
- 1 - 10uF (Micro-Farad) electrolytic capacitor - rated at 25v or more
- 2 - 22pF ceramic capacitors

1 - ATMEGA328P microcontroller
1 - 16 MHz crystal
Hook-up wire (22 Gauge - Solid)

My granular synthesizer can be improved in the following ways:

Line level upgrade - The Arduino circuit in the Granular Synth has been designed with minimal parts that are needed to make it function. If there were to be an upgrade the 5 Volt signal coming from pin 5 on the microcontroller, the signal would be adjusted to a peak-to-peak level of 1 Volt. Adversely, this does affect the quality of the original sound, removing the character that was there in the first place. By "lowering the value of the 100nF capacitor, the high-frequency magic would re-appear." (Volts, 2015) Although a 33nF capacitor replacement seems to work well.

Extra parts needed are: 1x 33nF Film Capacitor.
1x 100nF Electrolytic Capacitor 25V
1x 2.2k Resistor 1/4 Watt

Midi upgrade - A midi upgrade will transform this noisemaker into an actual instrument. (Volts, 2015b) The Granular Synth now needs a modified program (links on www.notesandvolts.com) plus extra components. Making the Arduino Synth midi controllable means that it will now accept midi note commands and will also respond to pitch bend data. The new sketch is also programmed so that the middle button, will play a pentatonic major scale of the note that is being played. The line level upgrade should be done first before attempting the MIDI upgrade. The tutorial for this also comes with the drill guide with instructions for installing the midi Jack. The optional DIP switch should also be added as this is the way to change MIDI channels without having to reprogram the Arduino each time you use it. A small degree of adjustment has to be made with the code, but some instructions go with it.

Extra parts needed are:

1x 6n138 Optocoupler
1x 1n914 Diode
1x 5-pin DIN Jack (midi)
3x 220 Ohm Resistor (1/4 Watt)
1x 4-Position DIP Switch
2 x 1/8 Aluminium Rivets
Single Row Header (5 pins total)

Front panel overlay – There is a template already provided on www.notesandvolts.com. I want to take the dimensions and change the graphics to personalize it using a program such as Photoshop. The overlay would also benefit from lamination to keep it strong and durable. (Volts, 2014)

Rubber feet. - Already purchased, for aesthetic purposes only.

A further possibility would be to investigate 3D printing, considering the low costs of actually producing and manufacturing my own enclosures on my desktop, seems like a good idea.

But until that time, makerspaces have tools that you do not necessarily have at home; tools include laser cutters and 3D printers. It is also a fantastic place to learn to write code or build a robot and learning from another by sharing knowledge and tools is an advantage. There is a vast amount of great online projects such as Code Hero – a game that teaches you to make games and codes. (Peake 2019). Resources like <https://www.thingiverse.com/> have examples of music, electronics, robots, gadgets for audio, cameras, and computers. I am particularly impressed by the World's first 3D Printed, modular record player by Qeske Community. (Kickstarter, 2019) The Maker movement embraces everyone from all communities and all backgrounds, from educators, hobbyists and professionals alike. "Maker spaces promote learning through play; have the potential to demystify science, math, technology, and engineering; and encourage women and underrepresented minorities to seek careers in those fields."(Britton, 2012)

Contrasting My Machine with A Commercial Product

Bastl instruments are a company of instrument makers whose ethos is community-based. Established in 2013, this group of individuals are based in Brno, Czech Republic, their main focus is devoted to the development and manufacturing of electronic musical instruments. Because they own their own factory, they are able to manufacture the instruments they produce themselves – all sourced from local businesses in the area. They are very community focussed, facilitating events such as running workshops, organising music events, and delivering educational content that encompasses modular synthesis, electronic music, new media art and D.I.Y. (Instruments, 2019)

One of the instruments they make is the Granny 2 Monophonic Granular Sampler. The machine is Arduino based and is totally hackable it also comes as a kit that you can build it yourself, and it has a Micro SD card that is used for storing and recording samples and presets. It has a line input and microphone for recording 8-bit 20050HZ WAV files, houses 6 sounds and 60 presets. What is great about this machine is that it is upgradable soundbanks which are based on the WaveRP Library. The Granny 2 has full external control over all Midi parameters such as midi note, cc, clock and transposing. With a choice of how it is powered – by either battery or 9v plug, this machine does give you a bang for your buck which retails at €200.00 (approx. £173.00). Because the Granny 2 is Arduino based it also has an Atmega328 microcontroller housed inside of it as well as an Adafruit Wave Shield. Lastly, the Granny 2 was designed using Fritzing software.

The Arduino Granular Synthesiser I built is very simple in design and requires very few parts and like the Granny 2, it has the Atmega328 microcontroller at its heart. It does not have any sampling facilities on board but does use a triangle sound wave to generate the grain that it produces. Midi control is available through upgrade only, this would add midi note and pitchbend messages from an external midi controller only. A Major Pentatonic scale programmed on the middle button of the synth would also be added. The Granny 2 has buttons that perform different functions, in order to upgrade my synth, I would have to employ the use of multiplexers, with a chip like the 74HC4067 that has 16 inputs. Lastly, the Arduino Granular Synthesizer has a single mono output as opposed to the input and output on the Granny 2. In

conclusion, I will have to gain a lot of experience to build something like the Granny 2, and this gives me something to aspire to.

DISAPPOINTMENT

On reflection, my machine failed because I became too bogged down in trying to follow the tutorial at www.notesandvolts.com. Had I consulted the website with the original granular synthesizer, I would have seen what Peter Knight had created, I would have known that it was much easier than I thought. Dave at www.notesandvolts.com was expanding on the original creation, and although it was simple in principle, there was more of a technical aspect to it as Peter Knight, had not included the Arduino hardware inside of the enclosure. Lack of knowledge and inexperience was also a significant factor. I am not sure about which technical aspect of my Granular Synth failed as I did not have enough time to troubleshoot. Perhaps it could have been a failed component or faulty wiring and soldering. I will endeavour to find out once I get the machine back.

What was encouraging, was that I learned how to build a standalone Arduino on a breadboard and how to program the circuit using an Arduino board as an In - System Programmer. Even though I spent a lot of time banging my head against a wall in frustration, it was a positive experience that felt almost therapeutic as I was building my enclosure. I do feel that learning to use software like Fritzing may be beneficial, it appears that it might be the equivalent of Logic Pro X or Ableton Live for visually working with an Arduino. Fritzing is described as "The perfect tool for designers, inventors, hobbyists, and educators for creating a prototype or even making PCB" (E-radionica, 2019) There is also a lot of information and tutorials that come with Fritzing, and you can use it to upload code instead of using the Arduino IDE.

Conclusion

Making the Arduino Granular Synthesizer was quite an ambitious project to take on. On this occasion the machine did not work so I unfortunately could not participate in Beta testing of the product. I do have feedback regarding the instrument, if I can produce a working model, I have customers waiting to purchase it. Although there are plenty of resources out there, I found I could not get a definitive answer regarding all the error messages I was getting on the serial monitor of the Arduino IDE. The failure of the machine has in no way deterred me from pursuing the ultimate goal of having a fully functional noise maker sitting on my desk, and although it may never be as comprehensive as the Granny 2 Sampler, I am very much looking forward to upgrading my working machine with a Line Level and a Midi upgrade that will enable me to use it as a midi instrument in conjunction with my DAW.

Bibliography

Ada, Lady (2013) *Overview | Wave Shield | Adafruit Learning System*. Available at: <https://learn.adafruit.com/adafruit-wave-shield-audio-shield-for-arduino> (Accessed: 12 April 2019).

Arduino.cc (2019) *Arduino - ArduinoToBreadboard*. Available at: <https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard> (Accessed: 11 April 2019).

Arduino™ Sketches: Tools and Techniques for Programming Wizardry. 1st edn. 10475 Crosspoint Boulevard Indianapolis, IN 46256: John Wiley & Sons, Inc. Nussey, J. (2013)

Britton, L. (2012) *The Makings of Maker Spaces, Part 1: Space for Creation, Not Just Consumption*. Available at: <http://www.thedigitalshift.com/2012/10/public-services/the-makings-of-maker-spaces-part-1-space-for-creation-not-just-consumption/> (Accessed: 10 April 2019).

Brothers, R. (2018) *Wiring DC Jack | Crickit Lab Shaker | Adafruit Learning System*. Available at: <https://learn.adafruit.com/crickit-lab-shaker/wiring-dc-jack> (Accessed: 3 April 2019).

Burk et al. (2019) *Music and Computers*. Available at: http://sites.music.columbia.edu/cmc/MusicAndComputers/chapter4/04_08.php (Accessed: 12 April 2019).

Cook, M. (2015) *Arduino Music and Arduino Projects*. Springer + Science + Buisness.

Drymontis, A. (2015) *Digital Electronics for Musicians*. New York: Springer + Science + Buisness.
E-radionica (2019)

What is Fritzing, what is does and how to use it? / e-radionica.com LEARN english. Available at: <https://e-radionica.com/en/blog/what-is-fritzing-what-is-does-and-how-to-use-it/> (Accessed: 9 April 2019).

GitHub - bkibby/OptiLoader: *Arduino sketch for burning new bootloaders* (no date). Available at: <https://github.com/bkibby/OptiLoader> (Accessed: 5 April 2019).

GitHub - nickgammon/arduino_sketches: *Publicly-released sketches for the Arduino microprocessor* (no date). Available at: https://github.com/nickgammon/arduino_sketches (Accessed: 5 April 2019).

Instruments, B. (2019a) *microGranny 2 | Bastl Instruments*. Available at: <https://bastl-instruments.com/instruments/microgranny> (Accessed: 8 April 2019).

Instruments, B. (2019b) *We are... | Bastl Instruments*. Available at: <https://bastl-instruments.com/lifestyle/we-are> (Accessed: 8 April 2019).

Kaziunas, A. (2014) *Make: 3D Printing*. 2nd edn. Edited by B. Jepson. Sebastopol C.A.

Lang, D. (Inventor) and Demarest, R. (2013) *Zero to Maker : learn (just enough) to make (just about) anything*.

Langbridge, C. A. (2013)

Kickstarter (2019) *Lenco-MD: World's first 3D Printed, modular record player by Qeske Community — Kickstarter, Kickstarter PBC*. Available at: https://www.kickstarter.com/projects/1766465955/lenco-md-worlds-first-3d-printed-modular-record-pl?ref=user_menu (Accessed: 8 April 2019).

Kickstarter. 2019. *Code Hero: A Game That Teaches You To Make Games*. Available at: <https://www.kickstarter.com/projects/primerist/code-hero-a-game-that-teaches-you-to-make-games-he>. (Accessed: 8 April 2019).

Platt, C. (2014) *Make: More Electronics*. Edited by B. Jepson. Oreilley Maker Media. doi: 2014-04-28T18:06:00Z.

Predko, M. (2005) *Digital Electronics Demystified*. New York, Chicago: McGraw-Hill. doi: 10.1036/0071441417.

Roads, C. (2001) *Microsound*. Cambridge, Massachusetts: The MIT Press.

Sparkfun (2019) *What is an Arduino? - learn.sparkfun.com*. Available at: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all> (Accessed: 6 April 2019).

Thimbleio (2017) *Setting up the Arduino IDE on Mac OS X - YouTube*. Available at: <https://www.youtube.com/watch?v=9sUmuD86sQs&t=7s> (Accessed: 4 April 2019).

Thomson, B. (2018) *Failing to make a vinyl cutter - Glasgow Maker Faire*. Available at: <https://glasgow.makerfaire.com/2018/03/13/failing-make-vinyl-cutter/> (Accessed: 8 April 2019).

volts, notes (2014) *Notes and Volts: Build the Auduino Granular Synth - Part 1*. Available at: <https://www.notesandvolts.com/2014/07/build-auduino-granular-synth-part-1.html> (Accessed: 1 April 2019).

Volts, N. (2012) *Notes and Volts: Fun with Arduino - Arduino on a Breadboard*. Available at: <https://www.notesandvolts.com/2012/12/fun-with-arduino-arduino-on-breadboard.html> (Accessed: 1 April 2019).

Volts, N. (2013) *Notes and Volts: Fun with Arduino - The Arduino as ISP*. Available at: <https://www.notesandvolts.com/2013/01/fun-with-arduino-arduino-as-isp.html> (Accessed: 1 April 2019).

Volts, N. (2014) *Notes and Volts: Build the Auduino Granular Synth - Part 2*. Available at: <https://www.notesandvolts.com/2014/08/build-auduino-granular-synth-part-2.html> (Accessed: 1 April 2019).

Volts, N. (2014) *Notes and Volts: Fun with Arduino - Arduino on a Proto-Board*. Available at: <https://www.notesandvolts.com/2014/07/fun-with-arduino-arduino-on-proto-board.html> (Accessed: 1 April 2019).

Volts, N. (2015) *Notes and Volts: Build the Auduino Granular Synth - Line Level Upgrade*. Available at: <https://www.notesandvolts.com/2015/04/build-auduino-granular-synth-line-level.html> (Accessed: 1 April 2019).

Volts, N. (2015) *Notes and Volts: Build the Auduino Granular Synth - MIDI Upgrade*. Available at: <https://www.notesandvolts.com/2015/05/auduino-synth-midi.html> (Accessed: 1 April 2019).

WaveRP (2011) *Google Code Archive - Long-term storage for Google Code Project Hosting*. Available at: <https://code.google.com/archive/p/waverp/> (Accessed: 5 April 2019).

Workshop, D. (2018) *From Arduino to ATmega328 | DroneBot Workshop*. Available at: <https://dronebotworkshop.com/arduino-uno-atmega328/> (Accessed: 4 April 2019).