The Flynth: A Digital Musical Instrument

By Dinis Tavares

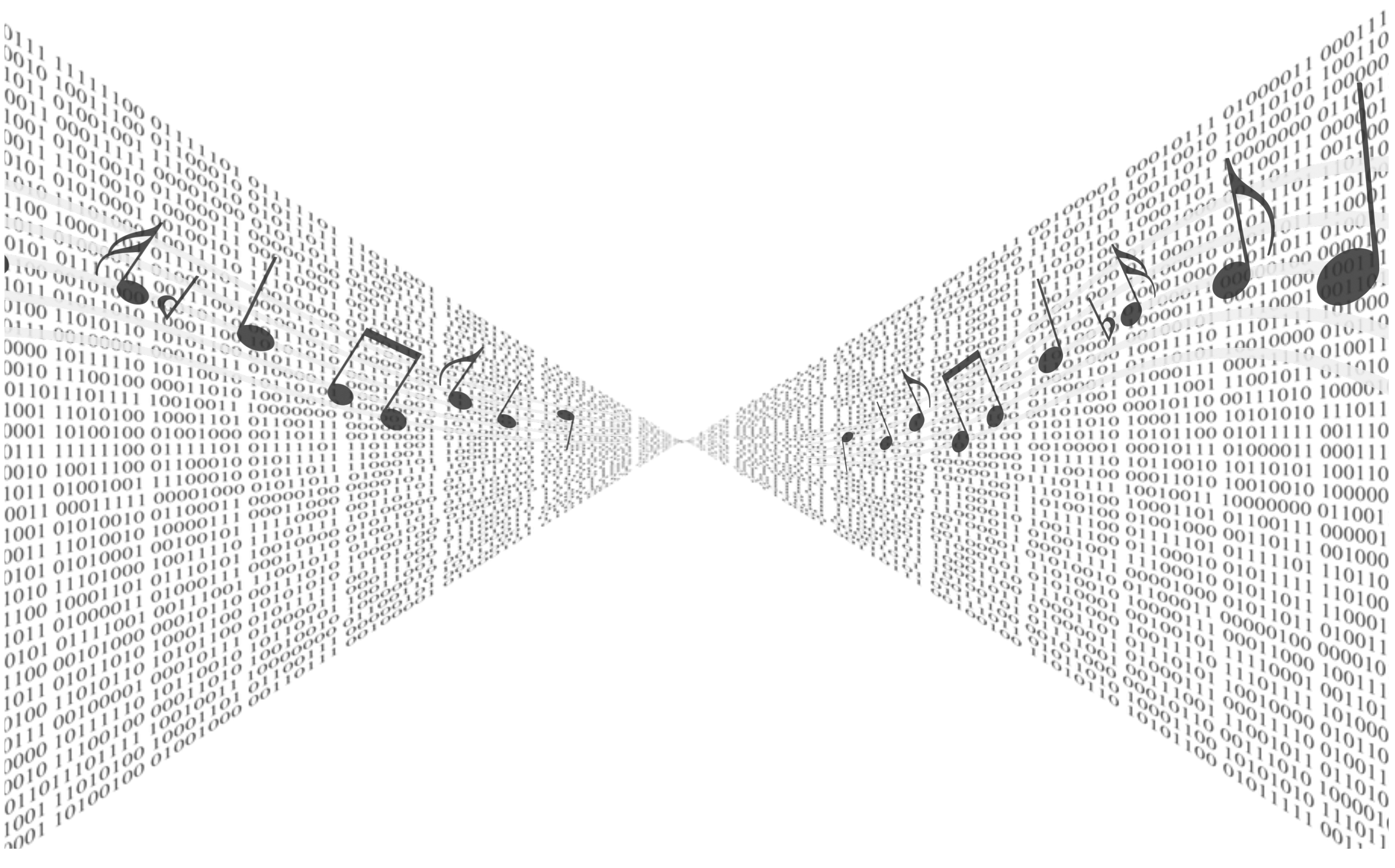


Table of Contents

Introduction

The report is based on the development of a new Digital Musical Instrument (DMI) called 'The Flynth', using the physical inputs similar to existing traditional wind instruments. The dominated electronic music interface since the 1960's has been the keyboard (Snyder and Ryan, 2014) and many researchers have explored designs to harness the expressive potential of wind players for electronic music. The goal of the DMI is to enable artists to expressively perform electronic music, where the user has full control of the sound using a synthesiser. In addition, the performer will have the ability to control parameters of the synthesiser whilst playing the instrument through the use of colour tracking.

Digital Musical Instruments

A musical instrument is defined to be a device which is used to create music and up to the 19th century, the design of instruments relied upon the mechanical systems and acoustic properties of tubes, strings and membranes (Miranda and Wanderley, 2006). The availability of electricity in the 1920s (NPS, 2015) inevitably led to instrument designers to experiment and create new instruments, such as the Theremin which was built in 1920 by Leon Theremin (Mirando and Wanderly, 2006). Digital musical instruments have allowed the separation of gestural control and sound synthesis, allowing performers to use the same gestures incorporated in traditional acoustic instruments to achieve completely different sounds. Mirando & Wandery claim there are four steps to creating a new DMI, summarised as:

- Deciding on which gestures will control the system and how the gestures will be captured and translated to electrical signals.
- Define sound synthesis algorithms to create sound or use existing music software to be controlled by the captured gestures.
- Mapping the sensor outputs to the synthesis inputs.
- Decide on feedback modalities apart from the sound generated by the system, such as: visual, tactile or kinaesthetic outputs.

Prior to incorporating the four steps into the designing stage for the Flynth, the definition of gesture needed to be understood. According to Jensenius et al., there has not been an unequivocal definition of gesture, however most authors seem to agree it involves body movement and meaning (2009). The following section will discuss which gestures have been used for the Flynth and determine the tools used to translate gestures into electrical signals.

Capturing Gestures

The Flynth uses three inputs from the performer, being: breath control, fingering, and movement of the instrument. The inputs are measured using an open-source electronic platform, based on hardware and software which read inputs and convert them into outputs (Arduino, 2018). The output messages are read by Max 8, a visual programming software, where the mapping of the inputs take place. In-order to send values from the Arduino to Max 8, an existing Max patch and some Arduino code created by Daniel Jolliff (2012) has been used and adapted to suit the Flynths needs. The Arduino code checks if values have been received by the hardware through the 'serial' object within Max 8, where if the letter 'r' is received all the analogue and digital pins will be read and consequently, the Arduino code returns the pins values to the Max patch. Digital pins read if values are on or off (0 or 1) where the values of the analogue pins are in the range of 0 and 1024. The code has been modified to use the first analogue pin as a digital pin, as 13 digital pins were needed for the Flynth.

Breath Control

The breath of the performer is measured by using the Freescale MPXV4006 (Figure 1) pressure sensor, measuring pressure levels between 0 to 6kPa and output voltage levels of 0.2 to 4.8 V (NXP, 2009). The pressure sensor consists of 8 pins, however only pins 2, 3 and 4 are in use and are connected to the Arduino as follows; Pin 2(Vs) to 5 Volts, Pin 3(GND) to Ground and Pin 4 (Vout) to Analog 2. The Arduino reads the voltage output of the sensor in values from 0 to 1024 and returns the values to Max.

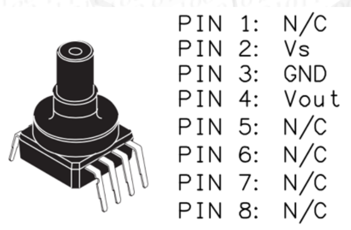


Figure 1 Freescale MPXV4006 Pressure Sensor

Fingering

The fingering will be read using tactile switches consisting of 4 pins (Figure 2), where T1 is routed to 5 volts and T4 is routed to the ground and a digital pin with a 10.2k Ohm resistor for each switch. When the switch is pressed down, the voltage connects to the ground and the digital pin (sending the value 1 to the digital pin) and disconnects once the switch is released (sending the value 0 to the digital pin). Seven buttons are used to determine the pitch of the note played and are connected to digital pins 2 to 8 on the Arduino board.

Colour Tracking

The movement of the instrument is recognised within Max by tracking a colour using an RGB LED attached to the end of the Flynth, where the user can switch on by pressing one/or a combination of the three separate tactical switches located at the bottom of the interface (Appendix A). A WP154A4SEJ3VBDZGW/CA RGB LED has been used, sharing a common anode and 3 grounding pins for the red, green and blue diodes Kingbright, 2018.), where each grounding pin is resisted by 220 Ohms. Digital pins 9 to 11 are used for the tactile switches to control the LED and grounding pins of the LED are connected to the two remaining digital pins and the first analogue pin on the Arduino board. The colour of the LED is determined within the Arduino code, which are set when a letter is received from between t and z from the 'port' object in Max.

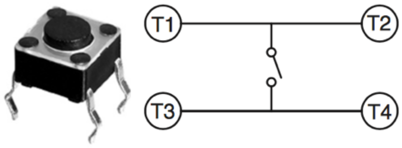


Figure 2 Tactile switch for fingering and LED control.

Mapping

The input for wind instruments require pressure from the lungs, where the amount of pressure needed can determine the duration of a note on a single breath (Wolfe et al., 2015). Wolfe et al. further outline that wind instruments requiring a higher-pressure level and low airflow, allows the performer to hold a note for a longer duration in one breath, where high pressures result in louder and higher notes. The design of the Flynth aimed to allow a small amount of airflow in-order to replicate the feel of playing an ordinary wind instrument and will pitch the note up depending on how hard the performer is blowing into the pressure sensor, enhancing the expression of the DMI. The vst~ object includes a 'Pitch Bend' input receiving a value between 0 and 127 where a scaled value from 64 and 127 will be received by the pressure sensor in-order to only pitch the note up in relation to higher pressures. In addition, the object includes an 'After Touch' input, receiving the same value from the pressure sensor scaled between 0 and 127. The user will have the freedom to control both attributes within Massive, enabling the performer to change the amount the notes are pitched up and which attributes are affected by the after touch, such as the vibrato of the synth. Finally, the pressure sensor values are scaled down into a float value, between 0 and 1, and multiplied by the signal output from the vst~ object to control the loudness of the instrument.

The Flynth allows the performer to play all notes within the chromatic scale across 8 octaves by bit shifting each value from the digital pins to the left, except for the first pin. Bit shifting is an operation in which the order of a series of bits is moved, either to the left or right, to effectively perform a mathematical operation (Hope, 2017). Therefore, if a button is pressed, the binary value will be sent to the bit shifter and depending on how many bits it is shifted, a new number will be produced. The numbers of all the active buttons are summed up allowing a maximum value of 127, providing possibilities of the entire midi range. However, in-order to make the DMI easier to play, the midi notes have been scaled down in a range from 21 to 108 providing the note range of a piano, as the notes above and below this range are seldom used. Additionally, the 'flush' object is triggered 5 milliseconds before a new note is sent to the vst, turning off all previous notes before a new note is played.

The colour tracking of the LED on the Flynth uses 'Jitter' within Max 8, utilising the webcam of a computer to control parameters in the Synthesiser Massive. Bit shifting is used once again to select the colour of the LED, by opening a gate to select and send a lettered value to the Arduino. Tracking the colour of the LED is achieved by using the 'jit.findbounds' object in Max and is set by tacking a minimum and maximum RGB value. The Flynth interface provides a tolerance dial to increase or decrease the minimum and maximum colours being tracked by the camera, and visual aid is provided to the performer of which colours are being tracked. Furthermore, the user has the option to recalibrate or change any colour to track by simply selecting which colour option to change on the dropdown menu, and simply clicking on the video output. The 'suckah' object is placed above the video output, creating a clickable area that returns an RBG value to the 'jit.findbounds' object to track the new colour. Once the colour is being tracked, the maximum and minimum coordinates of both the x and y axis are given by 'jit.finbounds' where each value is averaged to provide the centre coordinates. The coordinates are used to control any parameter within Massive, by scaling down the coordinates between 0 and 1, and are sent to the VST prepended with the parameter index.

To allow the performer full control, the width and height of each colour has the ability to change any attribute in Massive. A menu system has been implemented where all attributes are divided into submenus and an attribute needs to be selected on each colour in-order to control the attribute. By selecting an attribute on the 'unmenu', a specific gate is opened which allows either the x or y coordinates to control the specified VST attribute. The macros are able to control many attributes within Massive, where the macros can be controlled by the colour tracking creating a many-to-many mapping scheme found to engage and inspires the creativity of performers (Hunt et al., 2003). The attributes will only change if the LED buttons are pressed, releasing the colour buttons will turn off the LED, and the attribute setting will remain on the last value received.

Feedback and Expression

According to Bongers, the interaction between a human and a system is a two-way process: control and feedback (2000). The interaction takes place through an interface, translating real world actions into signals in the digital domain. The system is controlled by the user and feedback is returned to help the user articulate the control. In the case of the Flynth, the performer does not only receive sonic feedback, but visual feedback too, such as: the video output on the software interface highlights the colour being tracked by drawing an oval around it and when a note is played, an oscilloscope outputs the waveform at the top of the interface.

A common definition to be expressive is to effectively convey thought or feeling. In music, performers convey feeling by using a set of conventional signifiers such as tempo, timbre, vibrato, tone attacks, tone decays and pauses, which are understood by both the performer (Dobrian and Koppelman, 2006). These attributes can be controlled by the performer with the Flynth, by controlling the breathing techniques into the pressure sensor as well as ensuring attributes within massive are mapped to further enhance the expression of the instrument. An example of mapping massive to strengthen the expression could be achieved by setting the range of the pitch bend value in massive from 0 to 1 and adding the after-touch control to the rate and depth of the synth vibrato, allowing the performer to subtly control the expression of a note.

Conclusion

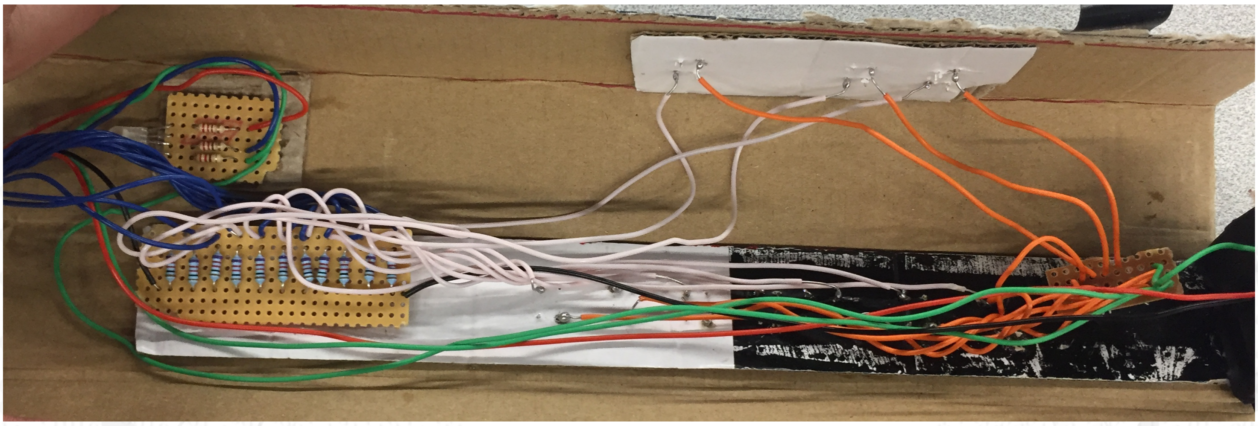
The Flynth has been built to allow a performer to have control over the creatively and expression of the sounds produced by the synthesiser. The pressure senor allows the user to control the dynamics and loudness of the notes and the ability to map the after-touch attribute will further improve the expression of the DMI. Adding more sensors, such as an embouchure sensor, would further increase the performers control over the expression and is seen in a digital musical instrument called the Bril (Snyder and Ryan, 2014). However, using the colour-tracking to control attributes in Massive, allow a performer to use gestures of movement to control the expression of the instrument in a different method of traditional wind instruments.

Appendixes

Appendix A

Appendix A The Flynth

Appendix B



Appendix B The inside of the Flynth

References

- Arduino, 2018. *Arduino - Introduction* [online]. Arduino.cc. Available from: https://www.arduino.cc/en/Guide/Introduction [Accessed 12 Jan 2019].
- Bongers, B., 2000. Interaction Theory and Interfacing Techniques for Real-time Performance. *In*: Wanderley, M. and Battier, M. *Trends in Gestural Control of Music* [online]. Paris: IRCAM, 41-70. Available from: http://www.musanim.com/pdf/PhysicalInteractionBongers.pdf [Accessed 11 Jan 2019].
- Dobrian, C. and Koppelman, D., 2006. The E in NIME: Musical Expression with New Computer Interfaces. *In*: *New Interfaces for Musical Expression* [online]. Paris: NIME, 277- 282. Available from: http://www.nime.org/proceedings/2006/nime2006_277.pdf [Accessed 12 Jan 2019].
- Hope, C., 2017. *What is a Bit Shift?* [online]. Computerhope.com. Available from: https://www.computerhope.com/jargon/b/bit-shift.htm [Accessed 13 Jan 2019].
- Hunt, A., Wanderley, M. and Paradis, M., 2003. The Importance of Parameter Mapping in Electronic Instrument Design. *Journal of New Music Research*, 32 (4), 429-440.
- Jensenius, A., Wanderley, M., Godøy, R. and Leman, M., 2009. Musical Gestures: concepts and methods in research. *Musical Gestures: Sound, Movement, and Meaning.*, 12-35.
- Jolliff, D., 2012. *Arduino2Max*. Max MSP, Arduino. Arduino. Available from: http://playground.arduino.cc/interfacing/MaxMSP
- Kingbright, 2018. *WP154A4SEJ3VBDZGW/CA* [online]. ebook. Mouser. Available from: https://www.mouser.co.uk/datasheet/2/216/WP154A4SEJ3VBDZGW-CA-1145236.pdf [Accessed 11 Jan 2019].
- Miranda, E. and Wanderley, M., 2006. *New digital musical instruments*. Middleton, Wis.: A-R Editions.
- NPS, 2015. *The Electric Light System - Thomas Edison National Historical Park (U.S. National Park Service)* [online]. Nps.gov. Available from: https://www.nps.gov/edis/learn/kidsyouth/the-electric-light-system-phonograph-motion-pictures.htm [Accessed 11 Jan 2019].
- NXP, 2009. *Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated* [online]. ebook. Freescale Semiconductor. Available from: https://www.mouser.co.uk/datasheet/2/302/MPXV4006-1127024.pdf [Accessed 11 Jan 2019].
- Snyder, J. and Ryan, D., 2014. The Birl: An Electronic Wind Instrument Based on an Artificial Neural Network Parameter Mapping Structure. *In*: *Proceedings of the international conference on new interfaces for musical expression* [online]. London: NIME14, 585–588. Available from: http://www.nime.org/proceedings/2014/nime2014_540.pdf [Accessed 3 Nov 2018].
- Wolfe, J., Fletcher, N. and Smith, J., 2015. The Interactions Between Wind Instruments and their Players. *Acta Acustica united with Acustica*, 101 (2), 211-223.