Towards Open Source Hardware Robotic Woodwind: an Internal Duct Flute Player

James Bennett **Bethan Moncur** **Kyle Fogarty** **Garry Clawson** **Charles Fox**
School of Computer Science, University of Lincoln, UK
25932340@students.lincoln.ac.uk

ABSTRACT

We present the first open source hardware (OSH) design and build of an automated robotic internal duct flute player, including an artificial lung and pitch calibration system. Using a recorder as an introductory instrument, the system is designed to be as modular as possible, enabling modification to fit further instruments across the woodwind family. Design considerations include the need to be as open to modification and accessible to as many people and instruments as possible. The system is split into two physical modules: a blowing module and a fingering module, and three software modules: actuator control, pitch calibration and musical note processing via MIDI. The system is able to perform beginner level recorder player melodies.

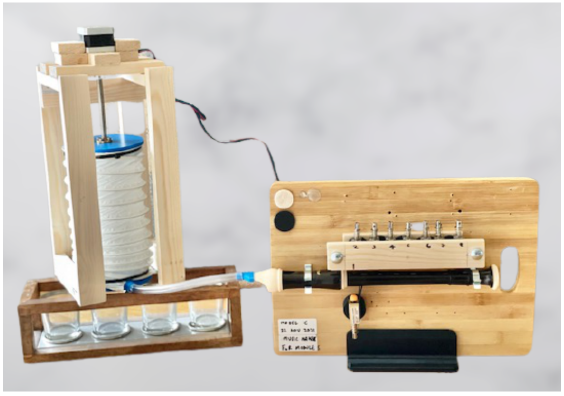

Figure 1. Photograph showing a build of the automated recorder player assembly with artificial lung and fingering module.

1. INTRODUCTION

Robotic performers have four major uses: As *assistive* technology, they may enable disabled users to participate in performances through semi-automation, such as automating the work of hands but leaving the user to use their own mouth or vice versa. Similar semi-automation can be used in *education* to enable students to focus on learning hand or mouth control individually. As *artistic* installations, fully automated robots can be used to perform music on physical instruments which is hard or impossible for humans to play, but retaining the live physical quality of their acoustic instruments. As *composition* tools, they can be used by humans who don't play an instrument themselves but would like to write for it, including for high quality, high accuracy studio recordings without the need for session musicians. The latter is especially important for orchestral film scores which are required to be edited and re-recorded many times at short notice to fit to new, sometimes daily, video edits. Synthesis of orchestral instruments remains unrealistic so physical automated instruments could cheaply and quickly enable these updates, beginning with a few lead instruments and growing via ensembles towards full orchestras.

While robotic performers have been developed for many instruments [1], they have mostly been developed independently of one another, and the research community lacks a method to build easily on one anothers' work. *Open Source Hardware (OSH)* is a recent movement [2] modelled on previous developments in Open Source Software to enable cumulative collaboration in hardware designs. Presenting a first research design as OSH enables it to be extended gradually by the community and develop into a robust and deployable solution. OSH also enables reuse of modules between related projects, for example a fingering module from a guitar could be modified and reused to finger woodwind. OSH does not simply mean reporting selected aspects of a design, rather it has a legal definition which requires the complete design to be available and buildable by anyone from commodity components. The CERN-OSH-W licence is a common legal definition of OSH and is used here. This licence ensures that any modifications made to the design are contributed back to the community, but also allows non-open products to use it unmodified as a subcomponent. Only a few robot musicians have been fully open sourced such as the guitar [3], violin [4], and percussive aerophone [5] family of instruments. There is a gap in the OSH robot orchestra for a woodwind family system.

Woodwind instruments are single pipes hosting a vibrating air column, whose harmonics are controlled by opening and closing finger holes along the pipe. Woodwind breaks down into reeds and flutes. Reeds use a vibrating reed, stimulated by the players mouth, to excite the air column. Flutes use the flow of air directly from the players mouth to excite the air column. Flutes are subclassified by how the air is blown: transverse flutes (such as the orchestral Western concern flute) have the air blow across a hole in the side; end-blown flutes (such as the middle eastern *ney*) have air blown *across* one end; and internal duct flutes (such as the recorder) have it blown *into* one end.

We would eventually like to automate the whole woodwind family. The sub-family of internal duct flutes, specifically the recorder, was chosen as this permits a standardisation of fingering actuation and air delivery. This provides a foundation for extension throughout the classification.

The contribution of this paper is the first open source hardware robotic system for playing the recorder (fig. 1) that non specialists can develop upon, and which is designed to support modular extension to other woodwind instruments. It consists of a mechanised lung to enable air flow regulation and allow breathing techniques to be seen during play (section 3.1), a fingering system for visual note actuation (section 3.2), a serial transceiver to interface utilising MIDI and an Arduino for system control and actuation (section 3.3). A calibration protocol that enables auto tuning of the system is also included (section 3.4).

1.1 Related Work

Flutes have been played since the stone age [6]. Internal duct flutes were originally made from bone or soft pithy wood commonly with six fingering holes and a speaker hole for the thumb [7]. The Banu Musa automated flute player of 9th century Baghdad [8] is important in computing history as it was both the first programmable machine of any kind. Automated internal duct flutes were popularised in 1736 by Jacques de Vaucanson [9] with a life-sized humanoid, 'The Flute Player,' playing twelve melodies using fingers as levers and a bellow system piping air into the mouth. Two centuries later, 'The Fluitist' extended upon this [10], mechanically reproducing the physiology of the organs involved during flute playing. Although full descriptions and general designs exist, both these early examples are too complex for non-specialists to build due to limited OSH design considerations.

The recorder playing robot [11] is an automated unsteady flow rate system, designed to enable the precise control of vibrato, specifically targeted towards music researchers of woodwind instruments. The systems uses a constant pressurised air supply with a quick response laminar flow sensor (QFS) to emulate natural sounds produced by a human player. Artificial mouths such as the single reed woodwind artificial blowing machine [12], implements an artificial tongue and actuated embouchure to study oscillation thresholds for various reed mouth pieces. Real time regulation of air pressure within an artificial mouth [13] enables a user to investigate the functioning of a recorder. Although extremely useful, previous recorder playing systems were developed as tools for realistic single note actuation and instrument fluid dynamic control analysis, rather than continual play robotic musical expressive purposes. Standardising an air delivery method will make it easier for users to obtain everything they need to use the system. A robotic bagpipe player [14] includes flute pipe controls. Each of the above include some public details of their designs which are used as inspiration here, but they predate the OSH concept so lack the licensing, build requirements, and instructions demanded by modern OSH.

OSH has been applied to instruments themselves as well as to robot performers. KeyWI [15] is a fully OSH/OSS electronic wind instrument. It is designed to emulate woodwind instruments via a keyboard for polyphonic pitch selection and a breath pressure sensor to create a dynamic range of sound. KeyWI is intended to improve upon previous limitations in breath sensors as well as providing a platform for development rather than enable the teaching of woodwind instruments themselves. The Open Woodwind Project [16] is an electronic woodwind instrument that is easy for users to build and develop. The breath system used, similar to the KeyWI is closed loop and requires a user to leak air out the side of their mouth to vary the pressure on the sensor to ensure pitch. Flow, a 3D printed recorder [17], is a socially responsible instrument designed to improve the psychological wellbeing of children with functional diversity in the upper limb. It requires only three fingers and a thumb to play. This is achieved via a piccolo key system, similar to a classical flute, to actuate the holes. Although simple in design and having many useful characteristics that would support our goal, the design is under patent so cannot be used here.

The OSH automated guitar [3] system includes fingering the frets using solenoids. This fingering is similar to what is needed for flutes, so – as it is OSH – parts of this design can be reused directly in the present work.

2. DESIGN CONSIDERATIONS

Multiple design considerations have been adopted from the automated guitar player, the Flow 3D printed recorder, as well as respecting the principles of CERN-OHL-W.

The automated guitar focuses on being *cheap* and *simple* to build (e.g., by a typical musician), broadening inclusivity and permitting configurable one-offs to be built to requirement at short notice. The Flow recorder emphasises *versatility* and *customisable* design, specifically enabling future modification to other woodwind instruments including orchestral traverse flutes and reeds. Where components cannot be self made, ensure that all *available components* within the system are available as per CERN-OHL-W. This means providing enough information to describe them and their interfaces or to enable them to be sourced.

For use as a composition tool, three further considerations were applied: The *capability* of the system will perform comparable to a basic human player. At minimum the system should seal all tone holes fully and move between notes with sufficient speed. The *level of autonomy* of the system should be fully autonomous other than if used as an assistive instrument. The *integration* of the system shall be via MIDI for best compatibility with existing products and industry standards. The physical system should be modularised where possible to support ease of sub-module or full system extension to other projects and instruments.

3. DESIGN

Our design is based upon, and forks from, a previous OSH guitar player [3] in which there is an Arduino microcontroller processing note instructions sent from a Python interface which actuates the hardware. The following important modifications have been made:

First, the design for the fingering module has been modified for the needs of the recorder. Second, an artificial lung is introduced to blow air into the recorder. Third, the communication protocol is redesigned to use MIDI. Fourth, a calibration procedure is implemented to calibrate the motor speed such that the correct air flow rate is obtained so each note is in tune.

The following sections give the detail of each design aspect. The full bill of materials, design, step by step build instructions as well as code and demonstration videos are available under the CERN-OSH-W licence from: `github.com/ICMC22-tmp/OSH_Automated_Recorder`

3.1 Blowing Module

An artificial lung was found to be the most appropriate means of blowing into the recorder over those of using a fan or pressure vessel type system.

The artificial lung (fig. 2) is constructed of a fixed end and an actuated end which are connected by a flexible concertina material. Pipes connect the lung to the recorder and atmosphere via one-way valves. The actuated end moves linearly via a lead screw and stepper motor which uses a DRV8825 driver and is controlled from the Arduino. This setup gives a near linear response of air flow rate to motor speed and readily facilitates setting different motor speeds to achieve the correct pitch of a note.

Figure 2. CAD model of bellow module.

3.2 Fingering Module

The fingering module (fig. 3) is an extension of the automated guitar fretting module. It is made of a five seat and single seat solenoid holding box, a bracket and clamp frame, and one four seat holding box. A base stand is also included to hold the instrument in an upright position.

The five seat holding box contains each of the solenoids to independently seal tone holes one through five along the top of the recorder middle joint. A single seat holding box is used to contain the solenoid for the thumb hole underneath the body at the start of the middle joint of the recorder. For the two double tone holes, a four seat holding box positions each solenoid back from each other and 90° from the centre line of the middle joint holes. Due to the close proximity of the smaller holes within each of the double tone holes, each solenoid has a fingerpad extension arm that enables close independent actuation without collision. These match the travel distance of the solenoids across the rest of the recorder. An M3 nut and bolt can be inserted into each box to secure each actuator in position.

Printed finger pads using a combination of rigid and flexible filament are attached as static end effectors to each solenoid to seal individual tone holes.

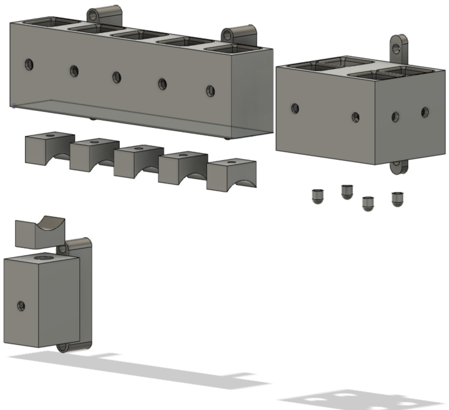

Figure 3. CAD model of fingering module.

3.3 Communication and MIDI Decoding

The Arduino (`arduino.cc`) controls the actuators via serial communication of MIDI messages [18]. The most common MIDI messages are note-on and note-off messages. They consist of three bytes each – a status byte, a byte to indicate which note to turn on or off, and a byte to indicate the velocity of the note (how loud or soft it should be played). The lowest note on the recorder, C5, is MIDI number 72 and each semitone is numbered consecutively. A MIDI message consisting of $90 72 60 translates to 'on channel 1, turn on note C5 at a medium volume'.

The Arduino responds to MIDI messages as they are received, thus it does not need to keep time. Timing is managed by the device sending the messages whether that is a MIDI controller or script on a computer.

Currently only data byte 1 is implemented due to the complicated nature of changing the volume of a note on a recorder – increasing air flow to make a note louder will also increase the frequency of the note, making it sound out of tune.

3.4 Calibration Protocol

The pitch of the recorder is controlled by the air flow rate produced by the blowing module; to ensure the recorder sounds as it should, the relationship between the stepper motor speed and the air flow rate must be calibrated. A high sensitivity sound sensor module was integrated into the Arduino system that can sample the sounds produced from the recorder and determine the dominant frequency via performing the Fast Fourier Transform (FFT) algorithm. The calibration procedure, which is only performed on initialisation of the system, is carried out as follows: 1) A note is selected and played while the sound sensor collects samples of the signal. 2) The signal is analysed via FFT, and the dominant frequency is identified 3) The difference between the measured frequency and a reference frequency for that note, Δf, is computed. 4) The motor speed is adjusted to try and minimise Δf via a simple proportional controller. 5) Steps 1 - 4 are repeated until Δf is within a predefined tolerance, then the new value of motor speed for that note is updated in the Arduino memory; this procedure is then repeated for each playable note of the recorder.

4. RESULTS

Video demos are provided in the GitHub repository including a) *Hot Cross Buns*, and b) *Rapid note playing*.

The system can be assessed by comparison with the abilities of a basic human player. This includes the ability to obtain the correct pitch at the correct time, without significant impact caused by mechanical constraints of the system, such as bellow capacity and excess system noise.

The ability to obtain the correct pitch was assessed by playing one note across the capacity of the bellow and measuring the variation in fundamental frequency as the note was played. For the note A5 (fundamental frequency 880 Hz) the variation is characterised in fig. 4.

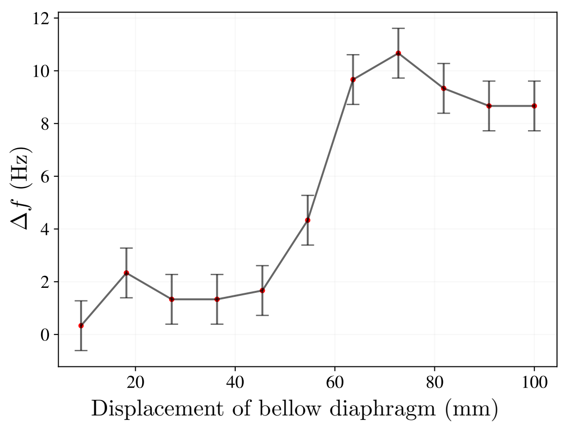

Figure 4. Plot of the difference in fundamental frequency ($\Delta f = f$ - 880) against bellow displacement; error-bars correspond to 2 standard deviations of the frequencies measured.

The ability to play a note at the correct time was assessed by measuring the time difference between the systems' note length and the desired note length as requested by the MIDI file. For a repeated A5 note, varying in length from 250 to 2000 ms, the average time difference was 110 $\pm$ 153 ms, with the lag increasing as the system progressed through the MIDI file. The artificial lung uses a four start, 2 mm pitch lead screw and is capable of playing four to eight breaths (depending on the note value: quaver to semibreve) per linear traversal. One full bellow inhale requires three seconds, however this is limited only by the speed of the motor and lead screw pitch length. The fingering module utilises ten solenoids with a 10 mm travel. Empirical testing has shown that a rate of eleven notes per second is achievable. An expert player, can achieve upwards of twelve notes per second but three notes per second is suitable for beginner player compositions.

5. DISCUSSION

Few examples of OSH robotic musical performers currently exist, but once they appear they may quickly evolve as the community is able to improve and modify them. Our OSH robot recorder player is a step towards this. Forking from a previous OSH guitar player, and using the recorder as an starting point for the whole woodwind family, the design is intended to be easily further developed, modified and extended upon. Initial results show the system is capable of replicating a beginner recorder player.

Further work could include a two-sided or twin lung system to allow notes to be played continuously without pausing for breath. A module for an artificial mouth, tongue and actuated embouchure, would allow for use in additional flute types (e.g., orchestral flute), and broader note and timbre characterisation. Interchangeable modules for reed instruments (e.g., oboe, clarinet) could also be implemented, utilising similar fingering and bellow designs.

Further software work includes looking ahead in the music, to enhance the timing and breathing control based on short-term planning as human players do. For example, how best to make use of finite air reserves until the next opportunity to take a breath. There is scope to incorporate musical dynamics in line with the MIDI 2.0 specification through adjustment of finger patterns to minimise pitch variation when increasing or decreasing air flow.

We invite interested members of the OSH community to contribute, extend, fork and build on the project, to higher accuracies and additional instruments.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [EP/S023917/1].

6. REFERENCES

[1] S. Kemper, "Locating Creativity in Differing Approaches to Musical Robotics," *Frontiers in Robotics and AI*, vol. 8, 2021.

[2] J. Bonvoisin, J. Molloy, M. Häuer, and T. Wenzel, "Standardisation of practices in open source hardware," *arXiv preprint arXiv:2004.07143*, 2020.

[3] A. Henry and C. Fox, "Open source hardware automated guitar player," in *International Computer Music Conference, ICMC; 2021 Jul 25-31*, 2021.

[4] D. Overholt and S. Gelineck, "Design & evaluation of an accessible hybrid violin platform," in *nime 2014: 14th International Conference on New Interfaces for Musical Expression 2014*. Goldsmiths, University of London, 2014, pp. 122–125.

[5] K. Sundberg, T. Sane, S. Barton, L. Baker, A. Walter, and A. O'Brien, "A robotic percussive Aerophone," in *18th International Conference on New Interfaces for Musical Expression*, 2018, pp. 15–19.

[6] J. Atema, "Musical origins and the Stone Age evolution of flutes," *Acoustics Today*, vol. 10, no. 3, pp. 26–34, 2014.

[7] A. Rowland-Jones, "Iconography in the history of the recorder up to c. 1430—Part 1," *Early music*, vol. 33, no. 4, pp. 557–574, 2005.

[8] T. Koetsier, "On the prehistory of programmable machines: musical automata, looms, calculators," *Mechanism and Machine theory*, vol. 36, no. 5, pp. 589–603, 2001.

[9] M. E. Moran, "Jacques de Vaucanson: The father of simulation," *Journal of Endourology*, vol. 21, no. 7, pp. 679–683, 2007.

[10] J. Solis, K. Chida, K. Suefuji, and A. Takanishi, "The development of the anthropomorphic flutist robot at Waseda University," *International Journal of Humanoid Robotics*, vol. 3, no. 02, pp. 127–151, 2006.

[11] T. Kato, Y. Kawamura, T. Funaki, K. Kawashima, and T. Kagawa, "Control of Blown Air For a Soprano-Recorder-Playing Robot Using Unsteady Flow Rate Measurements and Control Techniques," in *Proc. of 8th International Symposium on Fluid Power*, 2011, pp. 626–632.

[12] V. Chatziioannou, A. Hofmann, and M. Pàmies-Vilà, "An artificial blowing machine to investigate single-reed woodwind instruments under controlled articulation conditions," in *Proceedings of Meetings on Acoustics 174ASA*, vol. 31, no. 1 (035003), 2017.

[13] D. Ferrand and C. Vergez, "Blowing machine for wind musical instrument: toward a real-time control of the blowing pressure," in *2008 16th mediterranean conference on control and automation*. IEEE, 2008, pp. 1562–1567.

[14] R. B. Dannenberg, H. B. Brown, and R. Lupish, "McBlare: A robotic bagpipe player," in *Musical Robots and Interactive Multimodal Systems*. Springer, 2011, pp. 165–178.

[15] M. Caren, R. Michon, and M. Wright, "The KeyWI: An Expressive and Accessible Electronic Wind Instrument," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds. Birmingham, UK: Birmingham City University, Jul. 2020, pp. 605–608.

[16] J. Hopkins, "Open Woodwind project," https://github.com/jeffmhopkins/Open-Woodwind-Project, 2019.

[17] J. Esclapés, A. Gómez, and A. Ibañez, "Flow. A Socially Responsible 3D Printed One-Handed Recorder," *International journal of environmental research and public health*, vol. 18, no. 22, p. 12200, 2021.

[18] MIDI Manufacturers Association, "The Complete MIDI 1.0 Detailed Specification: Incorporating All Recommended Practices," Los Angelos, CA, 1996.