# Development of a Digital Musical Instrument with Embedded Sound Synthesis

(Exposé)

Pascal Staudt

Exposé zur
Masterarbeit
im Fachgebiet

Audiokommunikation und -technologie
Institut für Sprache und Kommunikation
Fakultät 1 – Geisteswissenschaften
Technische Universität Berlin



| | |
|---|---|
| **vorgelegt von:** | Pascal Staudt |
| **Anschrift:** | Weserstraße 35, 12045 Berlin |
| **E-Mail:** | pascal.staudt@posteo.de |
| **Matrikelnummer:** | 338990 |
| **Erstgutachter:** | Prof. Dr. Stefan Weinzierl |
| **Zweitgutachter:** | Dr. Till Bovermann (3DMIN, UdK Berlin) |
| **eingereicht am:** | 07.07.2016 |

## Contents

# 1 Abstract

The availability of powerful and inexpensive computing hardware made performances with digital musical instruments (DMIs) commonplace nowadays. However, although a lot of new DMI's have been developed in the last decade, none of them have managed to become widespread in the performance scene or to succeed on a commercial scale. Indeed, new DMIs are brought to the market, but still it seems that the major audio hardware sellers stick to the traditional paradigm of keyboard controlled synthesis. Numerous alternative musical controllers have been developed by individuals or in academic context, yet most of them rely on an external computer or synthesizer to process audio. Whereas nearly all artists in contemporary electronic music use sound-synthesis processed on computers, when it comes to performing on stage, the role of the computer as a musical instrument causes issues both technically and aesthetically. Hardly any attempts have been made to overcome these issues with the development of new DMIs that on the one hand go beyond the design of pure controllers or interfaces and on the other hand take advantage of modern computing hardware by embedding it into the instrument itself. The thesis wants to examine –on the basis of a specific case of instrument development– to what extent recent microprocessor boards are suitable for the development of a new *self-contained* DMI.

# 2 Introduction

No doubt, we live in an era of technical revolutions. Our lives are dominated by new technologies which changed not only our every day lives but also creative processes in many fields. It provides us with a sheer unlimited number of possibilities for the prototyping and development of new ideas. When it comes to the field of digital musical instrument design, undoubtedly the resulting availability and easy use of computing hardware and sensors of all kinds has brought up a whole scene of instrument designers.

With the rise of the *Arduino*[1], microcontroller boards have been become easy to program. Devices like the *Raspberry Pi* (RPI)[2] or the *BeagleBone* (BB)[3] even bring full featured minicomputers at the size of a credit card that cost less than 100€. Community driven projects like the *Axoloti*[4] or the *Bela*[5] further refine these technologies for the specific use in audio applications. All these device are small, come as a single piece of hardware, are quite affordable and provide interfaces for sensor input. The possibility of processing program code to run sound-synthesises furthermore makes them interesting for the design of new musical instruments. Nonetheless these devices are mainly used within do-it-yourself (DIY) projects or in art school context. Hardly any attempts have been made to use them for dedicated DMI design. This raises the question, whether the mentioned devices are suitable for professional instrument design and what criteria have to be met. For that several aspects concerning models, design principles and paradigms of instrument design have to be addressed.

The thesis wants to examine these questions in a specific case of a professional DMI development: the PushPull, an instrument that has been designed and developed within the 3DMIN[6] project. The haptic of the instrument is mainly influenced by a bellows that is pushed and pulled to produce an airflow, which is captured by two microphones. The output of the microphones,

[1] https://www.arduino.cc
[2] https://www.raspberrypi.org/
[3] http://beagleboard.org/bone
[4] http://www.axoloti.com/
[5] http://www.bela.io
[6] Design, Development and Dissemination of New Musical Instruments, http://www.3dmin.org/

and data gathered from capacitive sensors in the hand piece as well as an accelerometer, is then used to control different sound synthesis processes. The main drawback on the current design is the need of an external computer with software that runs the sound synthesis. This makes it inconvenient and unreliable to use in live performances. Furthermore, without the computer, it has to be seen as a controller rather than a stand-alone instrument. This issue will be addressed within the thesis by using a microprocessor board to embed the sound-synthesis into the Push-Pull and thus make it independent from the use of an external computer. An evaluation of the finished instrument and the development process itself shall help to clarify the question to what extent recent microprocessor boards are suitable for the design of a new DMI that goes beyond the pure development of a new control interface.

# 3 State of the Art

A lot of research has been published in the last decades, dealing with the the design of (new) digital musical instruments (DMIs). Since 2001, a large variety of DMIs have been presented at the annual *International Conference on New Interfaces for Musical Expression* ( NIME Steering Committee (2016)). Although artistic and scientific research in this field dates back earlier — the first *International Computer Music Conference* (ICMC) took place in 1974 (International Computer Music Association, 2016)— the interest in new DMI's has grown enormously in the last decade due to the availability of high computing power at cheap prices and small hardware sizes.

From the initial idea to a ready-to-play instrument a lot of different design decisions have to be made. The choice of the sensors, the computing hardware and the synthesis software affects all stages of the design process right up to performance aspects. For this reasons several models, design principles, and paradigms have been discussed.

While some work deals with the principles of DMI design (Cook, 2001; Paine, 2013) in general, other work focuses on specific aspects like parameter mapping, i.e the connection between the controller parameters gathered from the instrument's sensor input and the synthesis parameters. Hunt et al. (2003) suggests that this is the essential part in designing DMI's and the fact that mapping and feedback are the main research topics of the NIME conference supports his suggestion (Lyons and Fels, 2012).

The choice of mapping parameters influences the dimensionality of the instrument (Zappi and McPherson, 2014) and strongly depends on designing constraints (Magnusson, 2010). User studies on dimensionality (Zappi and McPherson, 2014) indicate that constraints can even be beneficial for the musical creativity. Since dimensionality and the related gestural control of DMI's determines how the performer is playing the instrument and consequently the audiences perception of expressiveness (Arfib et al., 2005), multidimensionality often leads to the fact that it is hard to grasp the link between the sound output of a DMI and the related performance gesture (Gurevich and von Muehlen, 2001). This can be a problem for both the performer and the audience which wants to be satisfied not only by the musical outcome of a live performance but also by the visual sensation. The latter is highly dependent on the musical expression the instrumentalist is able to perform.

All of these aspects are not only connected to the physics of the instrument but also to the underlying hardware and software implementation of the sound synthesis. Therefore, it is important to have a closer look at the emerging hardware solutions for embedded sound synthesis.

The availability of powerful and inexpensive computing hardware, which expands the possibilities of software based sound synthesis, is one of the reasons why performances with new digital instruments are commonplace nowadays. But as Paine (2013) points out, although a lot of new DMI's have been developed in the last decade, none of them have managed to become widespread in the performance scene and there is a need to further investigate why they didn't succeed on a commercial scale. Indeed, new DMI's are brought to the market by the major audio hardware sellers but still it seems, that they stick to the traditional paradigm of keyboard controlled synthesis (Wessel and Wright, 2001).

Miranda and Wanderley (2006) devotes a whole book to the goal of overcoming control and interaction with the keyboard. But he does not deal with the question of building a stand-alone instrument, only with the design of control interfaces for musical expression. At this point, it is very important to refer to the inaccurate terminology in the field of DMI design. A distinction between *controller* or *interface* and a stand-alone instrument has to be made, since DMIs often are made up of a combination of controller and computer.

Although nearly all artists in contemporary electronic music use computers, many do not see them as musical instruments (Lyons and Fels, 2012). Midi Controllers or custom built devices usually are used to overcome the poor input and haptic feedback of keyboard and mouse while making use of the possibilities of a personal or laptop computer for audio processing. For decades, most of the emerging DMI's were (MIDI) interfaces for controlling sound synthesis processes ran on computers, rather than stand-alone instruments.

In his thesis, Marshall (Marshall, 2008, p. 53) presents a survey on types of DMI presented at the NIME conference from 2001 to 2008. Although he doesn't focus on sound synthesis, but on physical instrument body design, the categories which are used, (*Instrument-like controller*, *Instrument-inspired controller*, *Extended instrument*, *Alternate controller*), with alternate controller being still the most present type of instrument, underline the fact that most of the effort goes into designing controllers which then become part of a musical instrument consisting of control interface and computer.

To distinguish DMI's which don't need an external computer from pure control interfaces, terms like *self-contained musical instrument* (Smith and Michon, 2015) or *embedded musical instrument* (Berdahl, 2015) are used for the lack of a clear terminology.

There is little research on so called self-contained DMI's, which use embedded sound synthesis. Before naming some of them, it makes sense to have a look to the related embedded hardware. A lot of possibilities exist to implement embedded digital sound synthesis, but due to their easy use, their cheap prices and their open hardware/software development, the focus here is on so called *microprocessor boards* and *embedded linux boards*.

Microprocessor boards have become more convenient since the rise of the Arduino. While they were hard to program in the past and therefore somehow reserved for electrical engineers, they are now accessible for a considerable number of persons with basic programming skills (Smith and Michon, 2015). On the one hand, their user-friendliness and affordability makes them very popular for use in art installations and the control of sensor input in DMI's. On the other hand, due to the lack of fast digital-to-analog conversion (DAC) they don't come with out-of-the-box audio functionality and therefore, in the field of instrument design, they are mainly used for controller development.

Another category of hardware is embedded linux boards like the Raspberry Pi and the BeagleBone, which come as small sized full featured minicomputers with considerable processing power. This makes them interesting for the use in dedicated stand-alone DMI design.

A relatively new hardware device is the *Axoloti Core* (Taelman, 2016), a powerful standalone microcontroller board which is graphically programmable via a software that is similar to *Pure-*

*Data* [7]. The most important difference to the above mentioned hardware is the fact that the board provides integrated 24bit/96kHz capable stereo audio analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) as well as midi in- and outputs. Furthermore, unlike the other hardware devices, it is specially designed for audio usage and does not come with extra interfaces that aren't necessarily needed for DMI design.

In the following, some examples that make use of the mentioned hardware are given: The JamBerry (Meier et al., 2014), a standalone device for distributed network performance, uses the RPI with some hardware extensions and an embedded Linux system in conjunction with the *ALSA*[8] driver which allows the connection to external audio hardware. Although the RPI comes with enough computing power for sound synthesis, it lacks high-quality audio output and extensions are needed (Meier et al., 2014).

The *Black Box* (Michon et al., 2013) makes use of the *Satellite CCRMA* (Berdahl and Ju, 2011) system running on a BB in an interactive sound installation. Although it uses PureData for the sound-synthesis, no information is given on the implementation and the quality of the audio output.

With the Satellite CCRMA a "musical interaction design platform designed to support the creation of new instruments" is provided to overcome the constraints of usual laptop or desktop pc solutions for NIME designs. It is available for both *Raspberry Pi 2* (RPI2) and the BB.

A similar approach to the Satellite CCRMA is presented by Vega and Gómez (2012). A lightweight Linux Distribution is ran on the BB which runs PureData in conjunction with *JACK*[9] and ALSA.

Equipping these mini-computers with high quality audio outputs and or inputs is essential when using them for ambitious sound applications. One possibility is to attach a small USB Audio Interface. The more elegant solution is to use a Audio Cape like the *Durio Sound*[10] for the RPI or the Bela shield for the *BeagleBone Black* (BBB). For the *D-Box* (Zappi and McPherson, 2015; McPherson and Zappi, 2015) an extension cape with eight audio in- and outputs is used to create a hackable self-contained instrument, which can be modified by the musician.

Examples with a physical design that can be related to the PushPull , are the *SqueezeVox* (Cook and Leider, 2000), an accordion like controller for models of the human voice and the *Accordiatron* [Gurevich.2001], an accordion inspired instrument that uses a programmable microprocessor to output MIDI to a computer running Max/MSP.

# 4 Preliminary Work and Approach

**The Current Version of the PushPull** In autumn 2015 a version of the PushPull has been built according to the development stage at this moment. Various sound synthesis patches have been programmed to examine the haptic and the general possibilities and constraints of the instrument. The goal has been set to overcome the need of an external computer for the sound synthesis by using embedded hardware.

**Literature Research and Choice of Hardware** After exploring the requirements for an embedded design, an extensive literature research has been made. Various hardware solutions have been considered, and first tests were run with the chosen hardware.

[7] http://puredata.info/
[8] Advanced Linux Sound Architecture, http://www.alsa-project.org
[9] JACK Audio Connection Kit, http://www.jackaudio.org
[10] http://duriosound.org/

**Prove of Concept** A concept for the embedding of the hardware is made and extensive tests are carried out to prove that the hardware and the concept complies with the requirements of the instrument.

**Hardware Implementation** The hardware is being embedded and all components of the PushPull are being put together.

**Software Implementation** After the assembly, various components are programmed to allow proper communication through their interfaces and to provide some test and debugging functionality as well as switching between different sound-synthesis approaches.

**Sound Synthesis and Evaluation** Three different sound-synthesis/mapping concepts are implemented to test the functionality and evaluate the finished instrument based on the specified requirements. Furthermore the development process will be focused, considering the different aspects of DMI design, and a final assessment is given to whether the chosen hardware suits the needs of dedicated DMI development.

# 5 Time Schedule

| Step | Time |
|---|---|
| Building current version of the PushPull | Autumn 2015 |
| Literature research and choice of hardware | January–Mai 2016 |
| Prove of concept | June 2016 |
| Hardware implementation | July 2016 |
| Sound synthesis and evaluation | August 2016 |
| Finish writing | September 2016 |

Table 1: Time Schedule

# References


Arfib, Daniel; Jean-Michel Couturier; and Loïc Kessous (2005): "Expressiveness and Digital Musical Instrument Design." In: *Journal of New Music Research*, **34**(1), pp. 125–136.

Berdahl, Edgar (2015): *Embedded Musical Instruments with the Raspberry Pi 2*. Online. URL `http://edgarberdahl.com/tools/2015/09/22/Satellite-CCRMA-on-RPi-2.html`. Access 16.05.2016.

Berdahl, Edgar and Wendy Ju (2011): "Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform." In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Oslo, Norway, pp. 173–178.

Cook, P and C N Leider (2000): "SqueezeVox: A New Controller for Vocal Synthesis Models." In: *In Proc. of the International Computer Music Conference*. pp. 5–8.

Cook, Perry (2001): "Principles for designing computer music controllers." In: *Proc. New interfaces for musical expression (NIME '01)*. Singapore, Singapore: National University of Singapore, pp. 1–4.

Gurevich, Michael and Stephan von Muehlen (2001): “The Accordiatron: A MIDI Controller For Interactive Music.” In: *New Interfaces for Musical Expression, NIME-01, Proceedings, Seattle, April 1-2, 2001*. pp. 27–29.

Hunt, Andy; Marcelo M. Wanderley; and Matthew Paradis (2003): “The Importance of Parameter Mapping in Electronic Instrument Design.” In: *Journal of New Music Research*, **32**(4), pp. 429–440.

International Computer Music Association (2016): *ICMC*. Online. URL `http://www.computermusic.org/page/23/`. Access 20.05.16.

Lyons, Michael and Sidney Fels (2012): “Advances in New Interfaces for Musical Expression.” In: *SIGGRAPH Asia 2012 Courses*, SA ’12. New York, NY, USA: ACM, pp. 1:1–1:159.

Magnusson, T. (2010): “Designing Constraints: Composing and Performing with Digital Musical Systems.” In: *Computer Music Journal*, **34**(4), pp. 62–73.

Marshall, Mark T. (2008): *Physical Interface Design for Digital Musical Instruments*. Ph.D. thesis, McGill University, Department of Music Research, Schulich School of Music.

McPherson, Andrew and Victor Zappi (2015): “Exposing the Scaffolding of Digital Instruments with Hardware-Software Feedback Loops.” In: Edgar Berdahl and Jesse Allison (Eds.) *Proceedings of the International Conference on New Interfaces for Musical Expression*. Baton Rouge, Louisiana, USA: Louisiana State University, pp. 162–167.

Meier, Florian; Marco Fink; and Udo Zölzer (2014): “The JamBerry - A Stand-Alone Device for Networked Music Performance Based on the Raspberry Pi.” In: *Linux Audio Conference 2014*.

Michon, Romain; Myles Borins; and David Meisenholder (2013): “The Black Box.” In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Daejeon, Republic of Korea: Graduate School of Culture Technology, KAIST, pp. 464–465.

Miranda, E.R. and M.M. Wanderley (2006): *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Computer Music and Digital Audio Series. A-R Editions.

NIME Steering Committee (2016): *Archive of NIME Proceedings*. Online. URL `http://www.nime.org/archives/`. Access 20.05.16.

Paine, Garth (2013): “New Musical Instrument Design Considerations.” In: *IEEE MultiMedia*, **20**(4), pp. 76–84.

Smith, Julius and Romain Michon (2015): “Emerging Technologies for Musical Audio Synthesis and Effects.” In: *Linux Audio Conference 2015*. Mainz.

Taelman, Johannes (2016): *Axoloti*. Online. URL `http://www.axoloti.com/`. Access 20.05.16.

Vega, Rafael and Daniel Gómez (2012): “Using the BeagleBoard as hardware to process sound.” In: *Linux Audio Conference 2012*.

Wessel, David and Matthew Wright (2001): “Problems and Prospects for Intimate Musical Control of Computers.” In: *ACM Computer-Human Interaction Workshop on New Interfaces for Musical Expression*. Seattle, WA: Association for Computing Machinery’s Special Interest Group on Computer-Human Interaction.

Zappi, Victor and Andrew McPherson (2014): “Dimensionality and Appropriation in Digital Musical Instrument Design.” In: *14th International Conference on New Interfaces for Musical Expression, NIME 2014, London, United Kingdom, June 30 - July 4, 2014*. pp. 455–460.

Zappi, Victor and Andrew McPherson (2015): “The D-Box: How to Rethink a Digital Musical Instrument.” In: *Proceedings of the 21st International Symposium on Electronic Art*.