View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Queen Mary Research Online

Queen Mary
University of London

Design and use of a hackable digital instrument

Zappi, V; McPherson, A; International Conference on Live Interfaces

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/jspui/handle/123456789/7204

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk

Design and Use of a Hackable Digital Instrument

Victor Zappi[1] and Andrew McPherson[2]

Centre for Digital Music, School of EECS
Queen Mary University of London
Mile End Road, London E1 4FZ, UK
[1] victor.zappi@qmul.ac.uk
[2] a.mcpherson@qmul.ac.uk

Abstract. This paper introduces the *D-Box*, a new digital musical instrument specifically designed to elicit unexpected creative uses and to support modification by the performer. Rather than taking a modular approach, the D-Box is a *hackable instrument* which allows for the discovery of novel working configurations through circuit bending techniques. Starting from the concept of *appropriation*, this paper describes the design, development and evaluation process lasting more than one year and made in collaboration with musicians and hackers.

Keywords: Hacking, Appropriation, DMI, Constraints, Embedded Hardware.

1 Introduction

The relationship between instrument designer and instrumental performer is more complex than these roles might suggest. Performers commonly develop playing techniques that were not part of the designer's original intentions. It is also common for performers to modify their instruments; regardless of the number of musical features and interaction techniques an instrument provides, its design will never fully satisfy the needs of every artist. Historical examples are plentiful, from Dizzy Gillespie's modified trumpet in the 1950s to the personalised electric guitars played by B. B. King and Eric Clapton to Keith Emerson's custom Moog modular synth (recently recreated by Moog Music).

Circuit bending (Ghazala 2005) is a practice by which musicians modify, repurpose and otherwise hack electronic devices. The origins of circuit bending date back decades (Collins 2008), but the practice is currently the focus of a vibrant online community which partly overlaps with open-source “maker” communities developing new musical interfaces. Curiously, however, it is rare to see a musician playing a hacked latest-generation DMI (other than those created by that musician). Many circuit benders prefer working on cheap electronics, including toys and other objects which are not designed for music-making, rather than more complex products of the DMI community.

The design of DMIs and software-based instruments can discourage hacking, especially by musicians without engineering training. Many DMIs are “black boxes” whose inner workings are difficult to understand, if they are accessible at all. High-speed digital circuits are more easily damaged by arbitrary rewiring than analog circuits of previous eras, and software is likewise fragile, where exploratory modifications are as likely to create a crash as to produce interesting sonic results. Therefore, while many new DMIs are created, future performers have limited scope to move beyond the original designer's specifications. Where conventional instrument designs pass from one musician to another, acquiring new creative meanings along the way, many DMIs exist only for a few performances and disappear to be replaced with completely new designs (Jordà 2004).

To investigate the relationship between instrument design and *hackability*, we created the *D-Box*, a self-contained digital instrument intended to be repurposed and rewired by the performer in unusual ways. Section 2 presents the initial investigations informing its design, including a study of unexpected use of a highly constrained instrument (Section 2.1) and interviews with instrument builders and circuit benders (Section 2.2). Section 3 presents the D-Box hardware and software, with a focus on features aimed at encouraging hacking. Sections 4 and 5 describe uses of the instrument in workshop and performance settings. Overall, the goal of the project is to create a DMI whose capabilities can be extended and modified in directions that we as the designers did not anticipate.

2 Instrument Appropriation

The process of developing a personal working relationship with an object is known as *appropriation*. It can be useful to consider appropriation in the design of human-computer interfaces: as Dix (2007) writes, "you may not be able to design for the unexpected, but you can design to allow the unexpected." Appropriation is common in musical performance, where the musician develops a personal approach to the instrument which might not fit the designer's original intentions. Hacking and circuit bending can be seen as extreme forms of appropriation, so our first step toward designing a hackable instrument consisted of understanding how design features influence exploration and appropriation of a simple DMI.

Performer appropriation of a musical instrument relates to the instrument's *affordances* (possibilities for action) but appears to be even more strongly guided by the exploration of *constraints* (Magnusson 2010). On traditional acoustic instruments, constraints can be a powerful motivator for creativity and the development of personal style, but curiously, when playing DMIs, musicians often perceive constraints to be restrictive and frustrating (Magnusson et al. 2007). Sometimes constraints can also elicit diversity of style in the digital domain, as Gurevich et al. (2010) showed with a simple one-button instrument; however, many DMIs are highly complex and tailored to the specific needs of only a few musicians, typically the ones involved in its design.

2.1 The Cube Instrument Study

To better understand how affordances and constraints affect musicians approaching and exploring a DMI, we ran a user study with a deliberately limited instrument. Building on the investigation in (Gurevich et al. 2010), we explored the relationship between *dimensionality* (number of independent controls) and appropriation. Full details can be found in (Zappi and McPherson 2014); highlights are summarised below.

We created a novel DMI, simply called the *Cube Instrument*, which consisted of a wooden box containing a touch+force sensor, speaker and a BeagleBone Black[1] (BBB) embedded computer. It resembled no other familiar instrument to avoid suggesting any playing conventions. When the touch sensor was activated, a tone was produced, presenting a clear and simple metaphor. A user study was conducted wherein 10 musicians received an instrument. Although all were externally identical, 5 out of the 10 replicas were configured to support 2 Degree-of-Freedom (DoF) control, namely timbre and pitch, while the remaining 5 had only timbre control (1DoF). Participants were randomly assigned a 1DoF or a 2DoF instrument with no modification allowed; they were then asked to prepare two original solo performances over the following month.

Quantitative and qualitative analyses were run on audio/video recordings, sensor usage data logs (saved on the instrument), interviews and written questionnaires. As in (Gurevich et al. 2010), performers showed a remarkable variety of styles and techniques, linked to the exploration of both main and hidden affordances (i.e. those not explicitly designed into the instrument, such as scraping the sides of the box or filtering the speaker with the hand). Some performers said they turned to unconventional playing techniques after feeling overly limited by the constraints of the instrument; others found the constraints themselves to be conducive to exploring subtle musical variations.

[1] http://beagleboard.org/black

Participants who were assigned a 2DoF instrument showed a tendency to rely less on hidden affordances than those with a 1DoF instrument. It might be expected that 2DoF participants, having a richer instrument, would have explored a wider variety of main affordances, but this was not the case. Counterintuitively, higher dimensionality appeared to simply hinder the appropriation of the instrument, reducing the exploration of *both* main and hidden affordances. This was a striking result. While the 1DoF group tended to seek more unconventional ways of playing, which is one of the rationales behind musical hacking, the 2DoF group fixated more on perceived limitations, and they spontaneously described modifications they would have liked to have seen in the instrument design to overcome the constraints they perceived as most limiting.

2.2 Hacking Constraints

The findings of the Cube Instrument study informed our approach to designing a hackable DMI. To maintain the incentive to appropriate the instrument, the initial configuration of the D-Box needed to remain simple and clear to the performer. On the other hand, following the performers' requests for additional capabilities, we sought to give performers a way to overcome the initial constraints by modifying the instrument.

Modular approaches to electronic instrument design are common, including classic analogue synthesisers, interconnectable hardware blocks such as littleBits and Patchblocks[2], and software environments such as Max/MSP and PureData. However, modularity did not fit the purposes of our project. Modularity implies customisability within fixed boundaries: no matter how many blocks can be interconnected or how many parameters can be tweaked, the possible modifications are defined *a priori* by the designer. There is a risk of creating an over-determined design (Redström 2006) which neither includes all the features required by the musicians nor leaves them with enough space for creative "misuses." In a very large modular environment, the performer may not encounter its limitations, but this implies either a complex initial instrument which is hard to understand, or that the performer builds the instrument themselves from simpler blocks, which was not our goal.

Instead, we aimed to allow musicians to modify the instrument by *hacking* its constraints: extending and subverting the limits of the device by rewiring it and bending its circuitry to change its behaviour. Exposing the inner workings of the instrument allows the exploration of unplanned and unpredictable configurations, enabling new modes of creative expression. To understand more about hacking techniques, we individually interviewed three London-based music hackers. Two individuals were *instrument builders* who develop instruments from scratch and one was a *circuit bender* who modified existing devices. We also attended performances by two other circuit benders.

Though the artists came from different backgrounds and worked independently, each expressed a consistent set of opinions. Among this group, programming was described as "alienating" and "confusing" as opposed to hardware hacking, which was seen as more "rewarding" and conducive to "immediate" physical results. The search for compelling ways of obtaining unusual sounds was the main motivation for hacking. To achieve their goals, all the artists used similar techniques, shorting or cutting connections or assembling electronic components and sensors. Interestingly, none of the artists had an engineering background, but based their approach on experience and trial-and-error techniques. The fragility of a working device and its tendency to go silent when things went wrong were seen as deterrents to hacking. One artist described opaque, hard-coded software processes as the "biggest enemy" of hacking. Finally, all artists targeted open-endedness in their works; each hack was described as an ever-evolving instrument, stemming from a precise plan but in continuous development. Making music with the instrument inspired further modifications and hacks in an ongoing loop. This approach, based on the personal artistic usage of the device, was seen to account for the main difference between hackers and conventional instrument designers, as memorably summed up by one participant: "You know Bob Moog? He never made an album."

[2] http://littlebits.cc and http://patchblocks.com

3 D-Box Design

Guided by the results of the Cube Instrument study and interviews with circuit benders and instrument builders, we created the *D-Box*, a digital musical instrument specifically designed to elicit unexpected creative uses and to support modification and customisation by the performer.

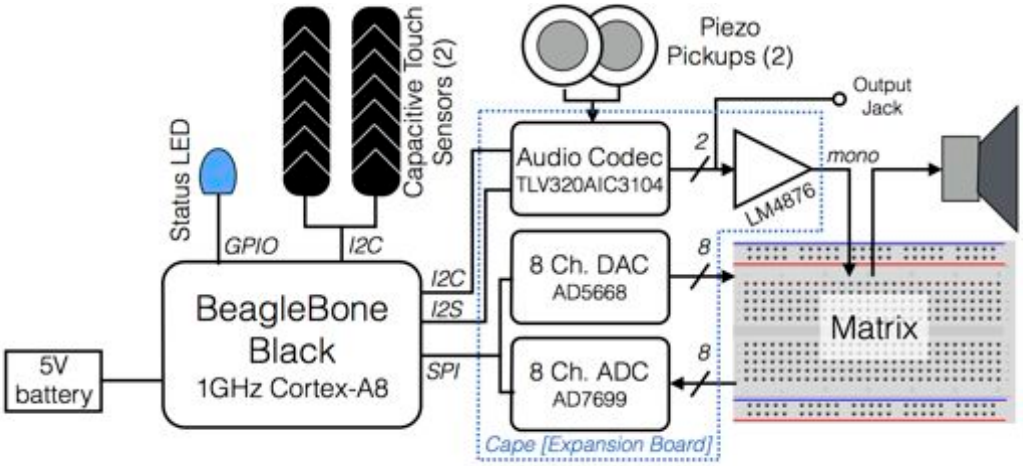

Fig. 1. Hardware for the D-Box. Hacking is focused on changing the matrix, a breadboard accessible through the side of the box, though other elements of the instrument can also be altered.

3.1 Hardware

Figure 1 shows a diagram of the D-Box. The core of the instrument is a BeagleBone Black single-board computer with a custom *cape* (hardware expansion board). The cape contains stereo audio input and output, a 1.1W monophonic power amplifier, an 8-channel 16-bit ADC and an 8-channel 16-bit DAC. The ADC and DAC are collectively termed the *matrix*, and these signals are brought out to a breadboard for the performer to modify as described in Section 3.3.

The D-Box is enclosed in a 15cm laser-cut wooden cube (Figure 2), identical in size to the original Cube Instrument (Section 2.1). Sound is generated by a 10cm full-range speaker. In contrast to the Cube Instrument, the top of the box contains two touch sensors, derived from (McPherson 2012); each is 10cm long and measures the location and contact area of up to 5 touches along a single axis. One touch sensor is stacked on top of a pressure sensor made out of resistive velostat material. Two piezo disc pickups amplify the acoustic vibrations of the box. The D-Box is powered by a 5V, 1000mAh rechargeable battery pack. A blinking status LED indicates when the box has booted and is ready to play (roughly 10 seconds after power-up).

3.2 Software environment

For the D-Box, we developed a new ultra-low-latency audio environment which improves on the latency and reliability of the standard ALSA Linux audio environment on embedded devices. The D-Box runs a Debian Linux OS with Xenomai real-time kernel extensions. Communication with the audio and matrix hardware is handled by the BBB PRU (Programmable Realtime Unit), which passes the data to the D-Box program running at a higher priority than the Linux kernel. Audio is sampled at 44.1kHz, and each of the 8 ADC and DAC matrix channels are sampled at 22.05kHz, synchronous with the audio clock. Where standard embedded Linux audio needs a hardware buffer of at least 128 samples for reliable performance (Topliss et al. 2014), the D-Box can run with hardware buffer sizes as small as 2 audio samples (1 matrix sample). For deployment, we chose a buffer size of 4 audio samples (2 matrix samples) as the optimal tradeoff between latency and processing overhead; in this state, the latency on the matrix is 182μs round-trip (ADC to DAC; 2 samples in and 2 samples out).

The D-Box program is written in C++ using the Xenomai real-time task API. The software uses an oscillator bank to reconstruct and transform sampled sounds. By using the NEON vector floating-point unit, up to 700 oscillators could be used at a time without underruns. This limit was not strongly dependent on the hardware buffer size. Analysis of sampled sounds is performed on a computer by SPEAR (Klingbeil 2006) and the resulting partial-based representation is stored on the D-Box via an SD card. 7 short sounds were loaded by default, and the user can also add their own.

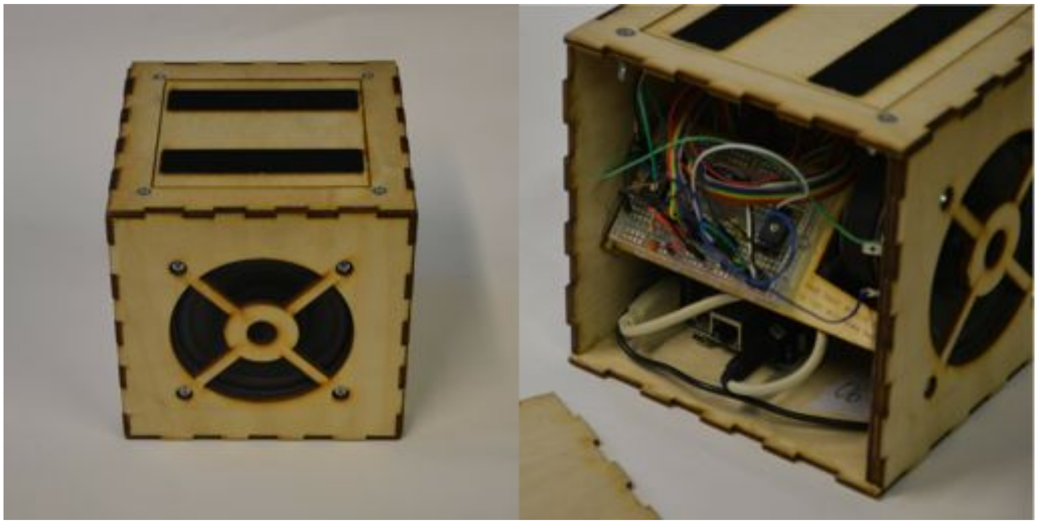

Fig. 2. *On the left: shot of the of the D-Box, showing the 2 sensors and the speaker. On the right: a detail of the matrix.*

Oscillator bank synthesis enables interesting transformations of the stored sounds, including pitch shift and time stretching independent from one another, change of amplitude/envelope, altering the waveform of each oscillator for timbre effects and detuning the oscillators to create inharmonic sounds. These effects are collectively controlled by the circuits attached to the matrix (Figure 3). Additionally, the piezo pickup inputs are amplified to accentuate the mechanical sounds of the box.

3.3 Hackable Hardware Features: Control on the Matrix

In this study, we focused on hardware hacking rather than software modification, inspired by the practices of circuit benders. The side panels of the D-Box open to reveal a breadboard to which all matrix inputs and outputs have been connected. In its standard configuration, the breadboard is pre-populated with simple circuits which define the instrument's basic metaphor. If left unmodified, the D-Box plays back the first of the default files every time the sensor opposite the speaker is touched, with pressure controlling volume. The original playback speed of the file is preserved, while the pitch can be altered with a range of 1.5 octaves by moving the finger along the touch sensor. A bandpass filter can be controlled using the second sensor, using up to 5 fingers to introduce 5 bands.

The basic principle of the matrix (8 ADCs and 8 DACs) is to create feedback loops between software and analog electronics. Analog signals are sent via the DACs, transformed through simple circuits on the breadboards, and read back into the ADCs. Each matrix channel has a separate function, as described in Figure 3. Though some of the matrix inputs function as simple control voltages (CVs), many of them have a dynamic behaviour which depends on the nature of the feedback from output to input. For example, the speed of playback is controlled by a hysteresis oscillator comprising a software-based comparator and a hardware-based RC network between output 1 and input 1 (red circuit in Figure 3). Changing the resistor and capacitor values affects the speed of playback and unusual effects can be obtained by removing these components or connecting wires to unrelated parts of the matrix. In comparison to a standard CV approach, feedback loops produced a wider variety of unexpected and time-varying behaviour when modified.

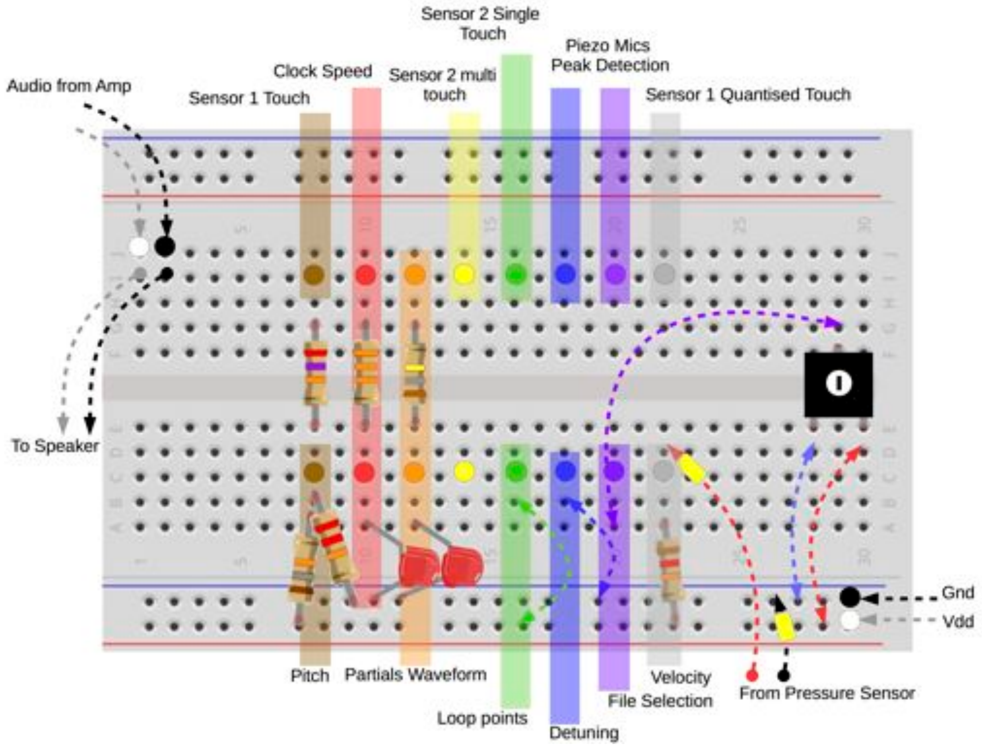

Fig. 3. *Matrix channels: the 8 analog outputs from the cape reach the breadboard on the top row of coloured dots. From the lower row, voltages are sent back to the 8 inputs of the cape.*

To ensure that no matrix wiring decision would damage the instrument, a 220 ohm resistor is placed in series with every input and output on the cape, before it arrives at the matrix. Likewise, the 5V (Vdd) supply rail, also present on the breadboard, has 100 ohms in series with it. The speaker output is also routed through the breadboard, and this signal is restricted to 0-5V range. No breadboard wires attach directly to the I/O pins of the relatively fragile BeagleBone. Any two breadboard signals can thus be shorted to one another without risk of damage.

Silence is perhaps the least interesting result of hacking an audio circuit, and the D-Box is designed so that no choice of wiring produces silence. Limits are placed on certain parameters, including playback speed and amplitude of oscillators, such that even extreme matrix settings produce some audible output. Other hackable features are present in the physical design of the D-Box: the top sensor panel can be removed or rotated in any direction; the pickups can be removed and repositioned; and the box can be played equally well with side panels open (wiring as a performance technique) or closed (wiring as preparation prior to performance). Because any electronic components can be used and wires can be left floating, the space of possibilities is wider and subtler than what could be achieved through a simple patch-cord approach to connecting inputs and outputs.

4 Pre-Study: Sónar Workshop

As preparation for the main D-Box user study, we ran two workshops at the Sónar electronic music festival in Barcelona, in June 2014. The workshops used a preliminary version of the D-Box; compared to Section 3, the main differences were fewer (3) preloaded sounds, no piezo pickups, and fewer matrix options. Each workshop had 15 participants drawn from the general public. During the 90-minute workshop, each participant received a D-Box to play and modify. Participants sat at a long table covered with various electronic parts to use. The session consisted of initial familiarisation with the instrument, explanation of a few specific modifications, then a short period of free hacking.

Feedback to the workshop was positive; it was seen as an engaging activity and demand exceeded available space. Following the workshop, we examined the wiring of each box and found a wide variety of configurations. Often the behaviour of the modified box would be surprising to us and would only be understandable on close evaluation of the wiring, supporting our goal of enabling performers to produce results we did not explicitly design for. Floating or "useless" wires going to unused columns on the breadboard were common, suggesting that participants tended to explore the wiring through arbitrary or empirical processes rather than strictly through a theoretical understanding of its function. This empirical approach would be seen again in our main study of experienced performers, though the performers tended to make more elaborate changes to the wiring (most likely due to longer timeframe and greater experience).

5 The D-Box User Study

The unpredictable hacks we saw at the Sónar workshop provided hints of the design's effectiveness, but further study was needed to assess how the D-Box features would affect musicians working on a piece over an extended time. We thus organized an extended user study observing how musicians appropriate and modify the box, and whether they perceive any of these activities as "hacks."

14 identical D-Boxes were built and given to musicians of varying background, including instrumentalists, electronic composers and circuit benders. Participants were asked to prepare 2 solo performances over the period of roughly a month (range 20-62 days on account of performer scheduling constraints). As a reward for their time and effort, participants could keep the instrument at the end of the study. In contrast to the Cube Instrument study, participants were allowed (but not required) to open the D-Box and modify the circuits inside; they were told that the matrix (breadboard) could be freely rewired while connections on the BBB itself were more fragile, but any sort of wiring and physical reconfiguration was permitted. Each participant was given an identical small bag of electronic components. Data gathering included written questionnaires, interviews, audio, video and sensor usage data directly saved on the instrument.

5.1. Style, Modifications and "Meta-hacking"

At the time of writing, 10 of the 14 participants have completed the study. Most numerical data remains to be analysed, but a first qualitative analysis highlights some interesting results. As expected, performers exhibited an extremely wide variety of styles and playing techniques. Much of this variety was attributable to functional modifications of the instrument, but we were also able to observe extensive exploration of its original features. As in the Cube Instrument study, the touch sensors were used in many different and sometimes unusual ways; techniques included multi-finger tapping to play melodies, rhythmically rubbing the wood panels and wetting the sensors to obtain infinite sustain. These similarities with the previous user study suggest that, as targeted, the unmodified design of the D-Box was broadly perceived as simple and clear, but still very constrained.

8 of the 10 musicians chose to modify their instruments. Comments and interviews suggested two primary motivations for hacking. Some participants modified the matrix to overcome limitations they perceived as encumbering during the composition phase. For example, some of them had to dynamically modify the pitch range of the instrument to play their piece; others felt constrained by the fixed playback speed, which did not allow proper syncing with other sound sources, and decided to modify the circuit to make it adjustable. Other participants, mainly skilled circuit benders and instrument builders, took an attitude to hacking related mainly to their musical background. They explored their D-Box focusing mostly on the matrix. Most of their hacks discarded the original capabilities of the instrument and, so far, it is still unclear to us exactly how many of them work.

In a group discussion, participants were asked whether modifying the D-Box constituted "hacking". Musicians with circuit bending background pointed out that the design of the matrix allowed them to directly apply their usual hacking techniques. In general, the fact that the circuits were not soldered but arranged on a modifiable breadboard helped speed up the bending process; however, some participants felt that the absence of a fixed circuit to "poke" diminished that sense of subversion that characterises hacking. All participants agreed that hacking includes the misuse of a device, enabling features that go beyond the original purpose of the instrument ("something you are not supposed to do"). From this perspective, some participants didn't consider changing the matrix to be "hacking", since the matrix design was meant to be subverted. Still, these participants identified modifications which they considered to be beyond even the hackable features of the D-Box. One participant referred to this process of modifying a hackable instrument in ways you are not supposed to as "meta-hacking." Identified examples included feeding the speaker output back into the matrix and creating feedback by touching the piezo microphones to the speaker.

6 Conclusion

The D-Box is a self-contained musical instrument whose design is intended to be modified by the performer. Rather than take a modular approach to building and modifying the instrument, a simple and apparently limited interface is presented to the performer in the beginning, but the internal mechanics of the instrument are exposed inside the box for rewiring in arbitrary ways.

The user study revealed a wide variety of style and two general motivations for modification: as a means of overcoming limitations of the device, and as an expression of personal performance technique. The approach based on overcoming limitations extends the original Cube Instrument study in showing the role of constraints in encouraging creative (mis)uses of technology, where the second approach shows that the design is flexible enough to accommodate the needs of experienced circuit benders. Finally, the fact that many of the novel behaviours were unknown even to the original designers suggests that the space of possibilities is ultimately determined by the creativity of the performer rather than strict limitations imposed by the designer.

Acknowledgements. This work was funded by the UK Engineering and Physical Sciences Research Council under grant EP/K032046/1 (2013-14).

References

Ghazala, Reed. *Circuit Bending: Build Your Own Alien Instruments.* Wiley, 2005
Collins, Nicholas. *A Solder's Tale: Putting the "Lead" back in "Lead Users".* Pervasive Computing, 7 2008.
Jordà, Sergi. *Instruments and Players: Some Thoughts on Digital Lutherie.* Journal of New Music Research, 33:321–341, 2004.
Dix, Alan. *Designing for Appropriation.* In Proc BCS-HCI, 2007.
Magnusson, Thor. *Designing Constraints: Composing and Performing with Digital Musical Systems.* In Proc. NIME, 2010.
Magnusson, Thor and Mendieta, Enrike Hurtado. *The Acoustic, the Digital and the Body: a Survey on Musical Instruments.* Proc. NIME, 2007.
Gurevich, Michael, Stapleton, Paul and Marquez-Borbon, Adnan. *Style and Constraint in Electronic Musical Instruments.* In Proc. NIME, 2010.
Zappi, Victor and McPherson, Andrew. *Dimensionality and Appropriation in Digital Musical Instrument Design.* In Proc. NIME, 2014.
Redström, Johan. *Towards User Design? On the Shift from Object to User as the Subject of Design.* Design Studies, 27:123–139, 2006.
McPherson, Andrew. *TouchKeys: Capacitive Multi-touch Sensing on a Physical Keyboard.* In Proc. NIME, 2012.
Topliss, James, Zappi, Victor and McPherson, Andrew. *Latency performance for real-time audio on BeagleBone Black.* In Proc. Linux Audio Conference, 2014.
Klingbeil, Michael. *Software for spectral analysis, editing, and synthesis.* In Proc. ICMC, 2006.