THE GATE MODULATOR. AN EXPERIMENT IN DIGITALLY-CONTROLLED ANALOG SYNTHESIS

Andrea Valle
CIRMA/StudiUm, Università di Torino
andrea.valle@unito.it

Simone Pappalardo
Cons. N. Piccinni (Bari) - Dip. Nuove Tecn. e Linguaggi
Cons. O. Respighi (Latina) - Informatica Musicale
simpapp@yahoo.it

ABSTRACT

This paper describes the Gate Modulator, a device for computer-controlled analog audio processing using the Arduino board. The Gate Modulator is based on the 4066 quad bilateral switch, an integrated circuit typically used for control applications, that is here used for audio processing at different time ranges, providing a variety of possible effects and uses. In the paper, first we consider the increasing role of the Arduino platform in audio applications, then we introduce the design and implementation of the Gate Modulator and the benefits of using Arduino as a controller, finally we describe the *Otosimbionte* project, an installation that exploits the capabilities of the Gate Modulator.

1. ARDUINO AND AUDIO PROCESSING

Arduino is renown as a prototyping board for physical computing [1]. But it is gaining an ever increasing popularity in various forms of sound processing. Its main benefits are low cost, small size, prototyping easiness. If compared to computer based DSP, it has lowest computational resources, but symmetrically it has lowest requirements (e.g. operating system). Moreover, it can be easily embedded into small physical objects, thus becoming "transparent", i.e. it can be hidden to the user, who is not forced to take into account a specific interface layer. Various experiences involving the Arduino microcontroller platform and audio are available:

- some projects directly use the Arduino ADC input (8 bits) to sample the audio signal [1] . Those include granular synthesis by means of a 8-bit weighted resistor output (R2R) as a 8-bit DAC [2] ; a "bitcrusher" that introduces a weighted pin technique [3] ; reimplementations of the most famous sound effects [4] , as typically used in guitar pedals [5] ;
- real-time sound synthesis project have been conceived [6] [7] [8];
- dedicated audio shields are available. Some shields embed only passive components (connectors, resistors, capacitors, potentiometers) [9] and are specifically targeted at guitar pedals [10] : DSP is realised by software. Other shield includes various DAC/ADC modules [11] [12] ;
- Arduino has also been used as an analog audio effect controller, e.g. as a digital LFO [13] or as a driver for controlled resistors that modify some analog distortion effect parameters [14] .

2. ANALOG GATE MODULATION

Our approach does not involve Arduino as a DSP board, rather it focuses on Arduino as a computational controller for analog audio processing. In this sense, it is a form of hybrid system for synthesis [2]. Before introducing Arduino, we discuss some aspects of the use of the 4066 IC. The featured analog processing is very crude per se, but it still allows to produce interesting results and does not belong (if not very loosely) to previously existent techniques. Its design directly springs from technological exploration with integrated circuits. In particular, it uses the well-known 4066 quad bilateral switch integrated circuit (see Figure 1) as controllable pass/not pass gate [3]. The 4066 IC acts as a logical gate for the input signal, its internal state being determined by a control signal. The audio signal is fed into the input, and the gate is then open/closed by a control signal. When the control signal has an audio frequency, indeed the technique results in a form of synthesis/processing that belongs to the modulation family (hence the name "gate modulation"). From a theoretical perspective (and in a possible digital implementation) this kind of processing would result in a windowing function uncoupled from the input signal, with very sharp, noisy edges introduced by the abrupt change in signal amplitude.

[1] http://www.uchobby.com/index.php/2007/11/11/arduino-sound-part-1/
[2] http://www.amandaghassaei.com/arduinodsp.html
[3] http://www.instructables.com/id/Lo-fi-Arduino-Guitar-Pedal/
[4] http://interface.khm.de/index.php/lab/experiments/arduino-realtime-audio-processing/
[5] http://wiki.leaflabs.com/Guitar_Audio_Effects
[6] http://playground.arduino.cc/Main/ArduinoSynth
[7] http://playground.arduino.cc/Main/PbSynthCode
[8] http://www.mycontraption.com/sound-effects-with-and-arduino/
[9] https://www.kickstarter.com/projects/2006771911/the-arduino-dsp-shield
[10] http://blog.arduino.cc/2014/01/28/program-sound-effects-for-your-guitar-with-pedalshield/
[11] https://learn.adafruit.com/adafruit-wave-shield-audio-shield-for-arduino
[12] http://174.136.57.214/html/rugged_audio_shield.html
[13] http://www.beavisaudio.com/Projects/DSWF/
[14] http://www.youtube.com/watch?v=_X0bL6WS-VY

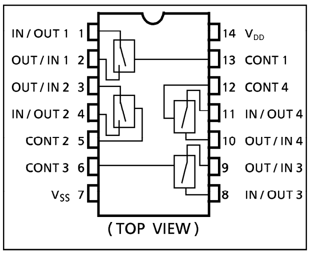

Figure 1. The 4066 quad bilateral switch.

The technique can be understood as a form of amplitude modulation, where the carrier is the input signal and the modulator (the signal resulting from the gating behaviour) can be thought as a square wave with varying frequency and duty cycle. While such a technique could prove interesting per se, its analog implementation by means of the 4066 IC introduces a relevant –analog-style so to say– smoothing due to nonlinearity in the onset phase. The IC requires a certain amount of time (which is variable from 15 to 70 nanoseconds, depending on the supply voltage and, in part, on the ratio from supply voltage and instantaneous amplitude of the audio signal) to change its state from open to close or vice versa, which results in a specific envelope applied to the input signal. A comparison between a digital implementation and the real 4066 IC signal is shown in Figure 2: in the second case, the same digital signal (a 1 Hz sinusoid) is fed into the IC and re-recorded, with the same gating parameters of the digital implementation.

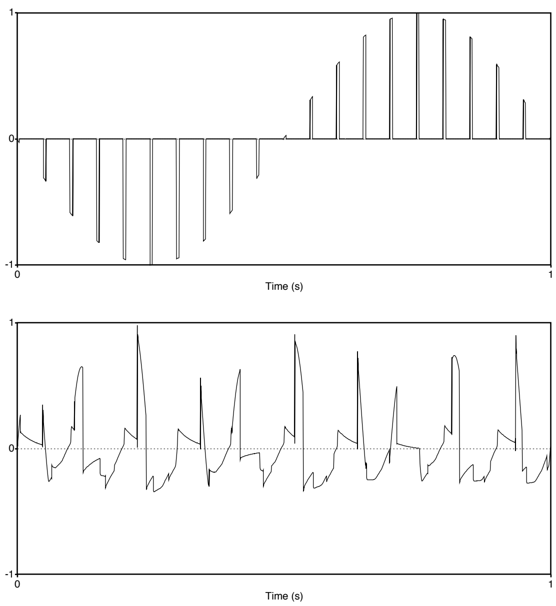

Figure 2. 1 Hz sinusoidal signal modulated by a gate signal: digital implementation (top) and 4066 analog implementation (bottom).

Even from these very first, empirical observations, nonlinearities are apparent [15] . When IC is on play, the modulator is no more a square wave, rather a sort of smoothed sawtooth. Complex spectral effects are thus obtained. The 4066 IC behaviour depends on the control signal, which can be thought as an unipolar square wave. Thus, if considering the wave resulting from an open/close cycle, two parameters are available: frequency and duty cycle. Frequency is related to the open/close period while duty cycle can be expressed as the open/close ratio in a period. Figure 3 shows a sonogram of a 1000 Hz sinusoidal signal fed into the 4066 IC with a fixed duty cycle (0.5) and variable frequency (frequency starts varying after some seconds). Modulation effects are apparent in the first 10 seconds, then, while frequency increases, time effects become relevant, i.e. a impulsive/granulating sound result. Harmonic frequencies related to the 1000 Hz fundamental gain progressively an increasing energy. Very low frequencies in

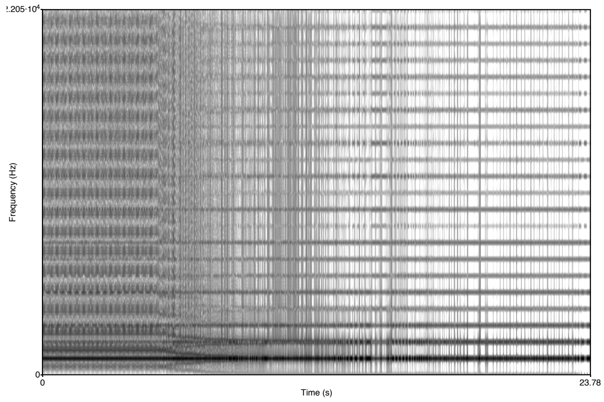

Figure 3. Sonogram of a 1000 Hz sine modulated with a variable frequency and fixed duty cycle.

the carrier (even in the sub-audio range) are still viable as the carrier acts like a "support" for various clicking effects introduces by modulation. In Figure 4, a 50 Hz sine is modulated with fixed frequency but decreasing duty cycle, that "dries" the signal introducing a stronger pulse/clicking effect.

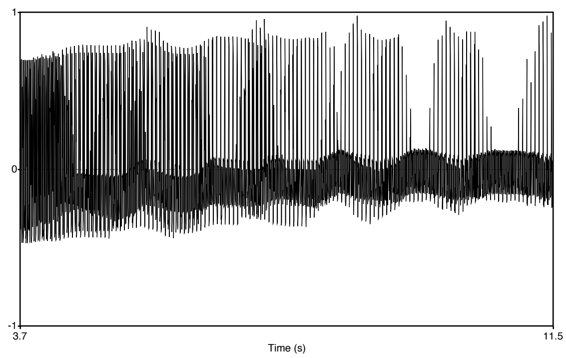

Figure 4. 50 Hz sine modulated with fixed frequency and decreasing duty cycle.

[15] Also, the gating process seems to introduce a certain amount of DC offset. Moreover, the Arduino's clock resolution may introduce some quantisation distortion.

3. USING ARDUINO AS A CONTROLLER

Gate modulation via 4066 IC can indeed be implemented in a complete analog fashion. In that case, a clock is needed to provide the control signals to the IC, that can be realised in various ways. Figure 5 shows a possible implementation by means of a 40106 Hex inverting Schmitt trigger [4], to which other diodes can be added in order to control the clock duty cycle. Another possible solution may include the notorious 555 timer IC. We will not deal into details of

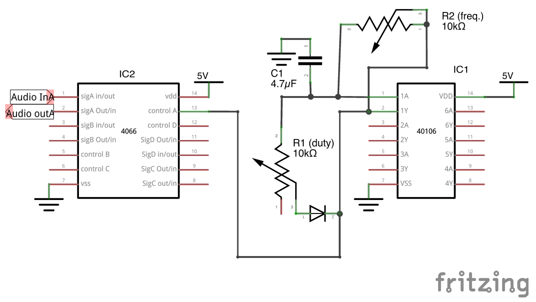

Figure 5. Analog implementation via 40106 Schmitt trigger.

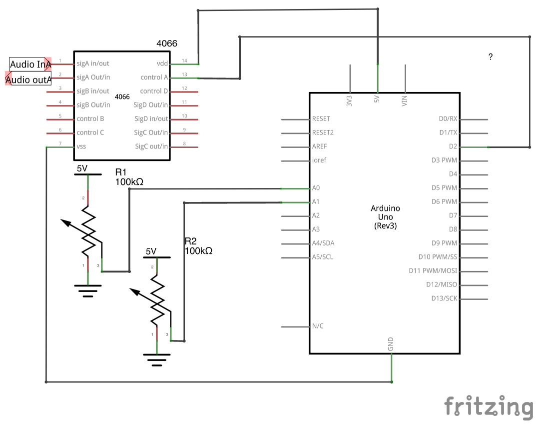

Figure 6. Gate Modulator circuit with Arduino.

analog implementation, as in our case we chose an hybrid approach, where the control signal is digitally generated and provided directly by the Arduino board. Our implementation is shown in Figure 6, where Arduino's digital outs are connected directly to the 4066 IC control ins. The use of a microcontroller results in various benefits with respect to a completely analog implementation:

- fast prototyping: in an experimental situation such as the one presented here, the relevance of the results lies only on the musical side. Thus, it must be assessed empirically by testing various control situations. While it is indeed possible to use well-known tools for electronic prototyping (e.g. by means of a breadboard), still the minimal setup Arduino/4066 is particularly light, as it shifts all the control logic on the Arduino language side. This allows the designer to work with a smooth trial-and-error methodology;
- multiple gate modulation: the 4066 IC contains four individually controllable switches, each one related to an input signal (audio, in our case). As it is trivial to connect four Arduino outs to the 4066 controls, four gate modulation processes are immediately available, that can operate on four different audio signal ("multiple ins" setup) or by applying four independent modulations to the same signal ("multiple modulator" setup). As this possibility is already provided by a standard Arduino (e.g. Uno), this expanded behaviour does not affect the hardware size.
- time independency: the scheduling of the control signals is handled by software. This means that it is not bound to particular upper time resolution. This allows to experiment with various modulating frequencies, with a time interval ranging from microseconds up to minutes. In this was, a continuum from audio synthesis to time scheduling is available. In our case a gate modulation device can easily become an audio signal distributor, a chopper, a tremolo, or a combination of these and other effects.
- hardware extension: the use of a microcontroller opens up other possibilities in hardware communication and control. The serial communication bus of the Arduino can be connected to a master software controller. In this way, the a gate modulation device can be controlled in real-time e.g. via USB port from a host computer, thus linking analog sound processing to algorithmic composition. Moreover, other features of Arduino can be exploited: as an example, by connecting the 4066 control ins with the Arduino PWM dedicated ports it is possible to generate control signals that can modulate the carrier with audio frequencies (typically, up to 980 Hz [16]). On the input side, it is easy to integrate the a gate modulation device in a physical computing framework, e.g. by connecting sensors to the Arduino and by associating them to a certain control logic for the modulation.

4. THE GATE MODULATOR

In this section we describe our implementation: the Gate Modulator. We chose a module-like design, that was inspired both by analog synth/processing modules and by guitar effect pedals. Also, it had to match, both functionally and aesthetically, the design of other devices that were meant to share the same hardware setup. Figure 7 shows three modules (see next section for an explanation), with the "Gate Modulator" on top (a). The design is indeed rooted into the DIY tradition in electronic music, that focuses on technological re-appropriation by means of low cost materials, hardware hacking, and knowledge sharing,

[16] http://arduino.cc/en/Reference/AnalogWrite

that is gaining a high momentum in recent years ([5], [6]). Following the "multiple ins" setup (see before), the Gate

Figure 7. The Gate Modulator (a) with other DIY modules (b and c).

Modulator features four parallel independent inputs and four related outputs for modulated signals [17] . As with the typical guitar pedal model, connectors are $\frac{1}{4}$ inch jack plugs. Modulation frequency for each input channel can be set by a dedicated knob. Knobs are connected to potentiometers read by Arduino's analog ins. In short, Arduino acts as the general controller for the whole modulation process, mediating between the user and the 4066 IC behaviour. Each knob has been programmed to exhibit a three-state behaviour. When a knob is set to minimum, its relative gate is always closed, so that the channel is "muted"; symmetrically, when it is set to maximum, the gate is always open, so that no modulation is applied; inside the range, gate modulation is active. This behaviour provides the user a mute/pass functionality without extra hardware interfaces. Two other knobs have been made available. The first (see the larger knob, right bottom, in Figure 7 a) allows to modify the duty cycle in the range $[0, 1]$. The second (smaller knob, Figure 7 a, right top) controls a scaling factor to be applied to all the modulations. In this, way it is possible to have a large range of frequency values while fine tuning the resulting values by means of each independent channel knob: as a result, time resolution varies from microseconds up to minutes, a range typically outside analog implementations. Figure 8 shows the internals of the Gate Modulator, with the 4066 IC in the foreground. An Arduino Uno is in use, that is placed in the box so that it shows externally the power inlet and the USB port. In this way it is possible to power the module externally, so that the Gate Modulator is in all respects an autonomous module. Moreover, by means of the USB port it is still possible to connect it to a computer and reprogram the Arduino so that the IC may be controlled from an external application bypassing the knobs.

[17] To independently manage the logical channels of 4066 the Arduino must be programmed for multitasking operations.

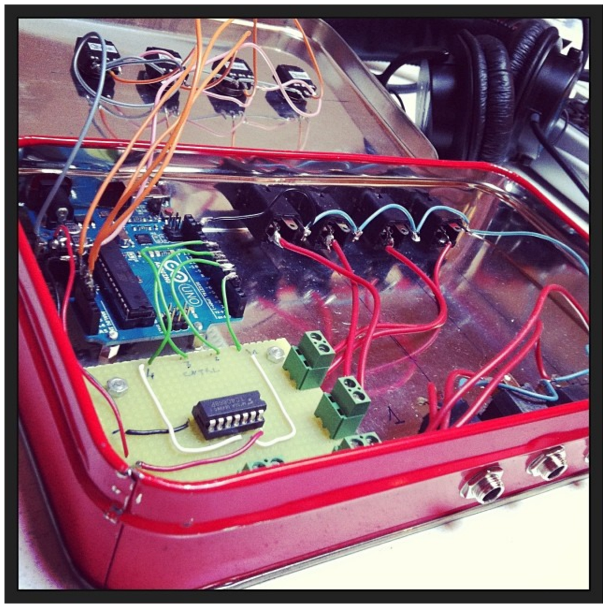

Figure 8. The Gate Modulator. Internals.

5. USAGE: THE OTOSIMBIONTE PROJECT

The *Otosimbionte* project started as a collaboration between the two authors in the occasion of the exhibition *Il sogno verde* curated by *Brecce per l'arte contemporanea* (March 22-23, 2014) at Villa Gregoriana, Tivoli. Otosimbionte is intended as a sound installation (see in general [7]) that, while focusing on sound production and symmetrically on aural perception, still retains a specific physical presence. The design results from the intervention of various elements:

- the DIY tradition in electronic music, that focuses on low-cost in order to assemble experimental sound devices, following the path of the American tradition ([5], [8]);
- the attention to reused, recycled, poor materials as a political way to challenge late Capitalism attitude towards the waste of resources, and as a way to recontextualize everyday objects (see also the Arte povera movement or the Nouveau Realism; for a discussion in the context of physical computing see [9]).
- the interest into feedback as a general, powerful expressive resource for generating content (see [10] for a discussion) that emerges directly from interaction between agents of variable nature (be them software or hardware), without prior definition.
- the collaborative attitude in artistic production that underlines the role of cooperation against competition and that blurs the distance between "pure" technical and artistic phases.
- the blurring of the installation/performance contexts, so that an installation can be performed, and vice versa a performance system can be turned into an installation.

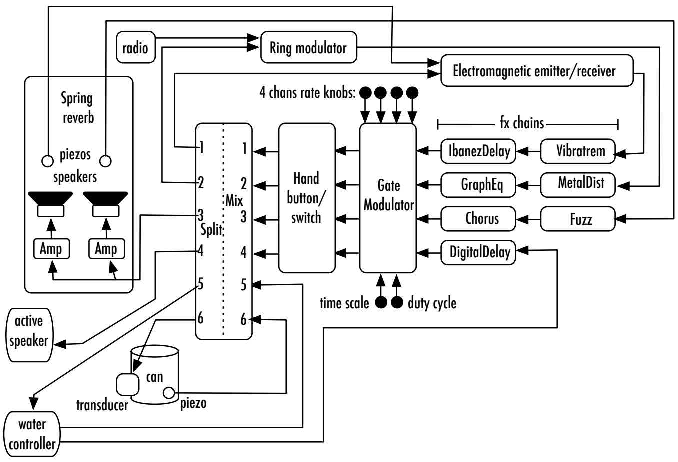

Figure 9. Diagram of Otosimbionte components.

Otosimbionte results from the hybridisation via feedback of each author' sound system. Hence the name, that refers to symbiosis and to listening. The whole setup is shown in Figure 10. The installation has been placed on a "raft" inside the ruins of an ancient Roman villa, so that it emerged from the context as an isolated organism. It has been conceived and designed at distance (the authors living in two different cities), and finally assembled and fine-tuned in loco, where it has taken its final shape. Components are detailed in Figure 9. We describe Otosimbionte as an example of the benefits of the Gate Modulator and as an extension to an artistic object of the principles at its basis. The whole system is based on a continuous circulation of audio material from its output to its input by means of six channels of audio streams. The effect chain block (right) is made by standard guitar pedals: they are connected to the Gate Modulator that operates by chopping them at variable time rates. The four outs are sent to a custom made hand button/switch module (Figure 7b, see later). The four audio channels are sent to a minimal DIY passive mixer/splitter (Figure 7c) that mixes all 6 the input into a single output and split this mono signal into 6 outputs. These outputs are sent to other custom devices. The spring reverb is assembled by connecting small loudspeakers to springs and by capturing their physical vibrations by means of piezo microphones. The ring modulator is a classical passive DIY effect made up of two small transformers and four diodes that allow to modulate the incoming signal from the splitter with a hacked radio. The electromagnetic emitter/transducer is a system of 4 modified coils. Three of these are used as electromagnetic emitters of signals coming respectively from the mixer/splitter and from one of the two channels of the spring reverb. The fourth coil captures the resulting signal.
Indeed, it is possible to derive from each point of the feedback chain an audio signal to be delivered to the audience. We chose two "extraction points". On one side, a large recycled metal can (visible in Figure 11, left) is used as a resonator by applying an amplified transducer to its surface, thus delivering the sound in the environment. A second sound output is provided by an active loudspeaker connected to one of the outputs of the mixer/splitter.
Even if it is mainly based on the constant, internal recirculation of audio signals, Otosimbionte gathers sound also from the external world: first, by integrating the radio signal in the ring modulator, secondly by providing four interaction points to the user.

- the hand button/switch module: by simply muting the desired streams, the whole sound output can be greatly affected;
- electromagnetic receiver/transmitter: its output varies in relation to the mobile coils, available to the user. By placing the receiving coil closer to one or the other transmitting coils it becomes possible to actively control the amplitude of the relative audio signal. The same signal may be modulated by inserting various electromagnetic noises, e.g. by introducing near the receiver small motors, fans, the same power supplies in use for Otosimbionte;
- spring reverb: the user can effectively intervene on the springs.
- water controller: an Arduino-based tangible system, it recognises various types of touch in the water (Figure 9, middle left). A copper wire is placed below the bowl –partly filled with water– that acts as a proximity sensor. The hardware and software configuration has been redesigned following an example from Disney laboratories [18] in order to make it compatible with Arduino [19]. While in the original design a computer was in use, in our case the whole control processing is entirely performed by the Arduino. The system basically distinguishes between two main types of touches: if the bowl is touched externally or the water is lightly perturbed, the signal is routed to the output A; rather, if the hand is immersed in the water, the signal is routed to the output B.

[18] http://www.disneyresearch.com/project/touche-touch-and-gesture-sensing-for-the-real-world/
[19] http://www.instructables.com/id/Touche-for-Arduino-Advanced-touch-sensing/
http://madlabdk.wordpress.com/2013/04/08/2-touche-meets-pd-and-maxmsp/

Figure 10. The *Otosimbionte* installation at Villa Gregoriana, Tivoli.

Both the knobs on the Gate Modulator and the hand button/switch module are not relevant to the installation per se, but allow to immediately turn the installation into a performance device, where the musicians operate on the various control points dramatically varying the sonic output. In fact, various improvisation performances on the "raft" have been realised by the authors as a part of the exhibition [20]. Figure 11 shows a live performance in a club where the raft has been "transported". In the context of Otosimbionte, the Gate Modulator has proven to be a crucial feature. A typical feature of analog feedback systems is to generate continuous, drone-like sounds ("sons homogènes", to speak with Pierre Schaeffer), making it difficult to obtain impulsive, granular, isolated sounds. The switching mechanism has been a key element in escaping this typical "feedback sound", rather providing a much more "event-oriented" sound perspective, from modulation to granulation, up to long sound events surrounded by silence. In relation to this aspect, by varying the duty cycle and the time interval, we were able to obtain a sort of "sound breathing" in the installation, that slowly evolved through a cycle of presence/absence of sound, further emphasised by the natural reverberation of the location.

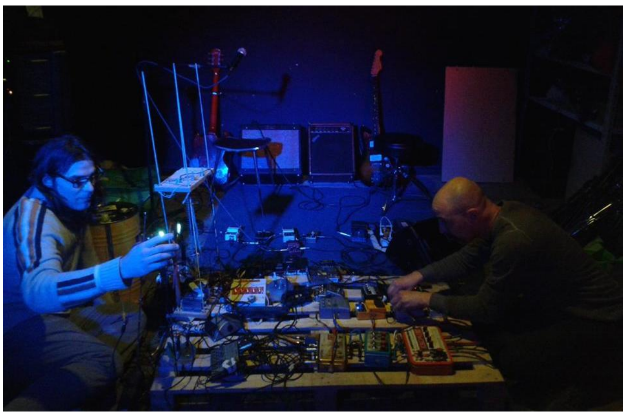

Figure 11. Improvising on the Otosimbionte's raft.

[20] A video can be seen here: https://vimeo.com/106018857

6. CONCLUSION

The Gate Modulator is intended both as an exploration of hybrid synthesis processes and as an experiment in a low-cost, unconventional hardware construction. Digitally-controlled gate modulation has proven to be an unusual yet interesting technique for sound generation and processing, in which the limited parameters (substantially, frequency and duty cycle) were functional to an immediate expressive usage. On the other side, the Arduino microcontroller allowed us to build a very versatile, portable hardware device (as seen in Figure 8, most of the case space is dedicate to connectors). Versatility is to be intended as a twofold feature: first, in relation to a large range of gating frequencies; second, in relation to hardware expansion (e.g. reprogrammability of the device, a feature that "softens" the hardware). Future work will focus on a more in-depth analysis of the acoustic distortion introduced by the analog gate, and on the expansion of the computer-controlled switching logic to a multiplexer for bridging synthesis processes with mixing.

7. REFERENCES

[1] M. Banzi, *Getting started with Arduino*. O'Reilly, 2009.

[2] C. Roads, *The Computer Music Tutorial*. Cambridge, MA, USA: MIT Press, 1996.

[3] Toshiba, "TC4066BP/BF/BFN/BFT." datasheet, 1997.

[4] NXP B.V., "HEF40106B Hex inverting Schmitt trigger." datasheet, Nov. 2011.

[5] N. Collins, *Handmade Electronic Music. The art of hardware hacking*. New York–London: Routledge, 2006.

[6] VV.AA., "Re-wiring electronic music," *Organised Sound*, vol. 18, 2013.

[7] B. LaBelle, *Background noise: perspectives on sound art*. New York–London: Continuum, 2006.

[8] VV.AA., "Composers inside electronics: Music after David Tudor," *Leonardo Music Journal*, vol. 14, 2004.

[9] A. Valle, "Making acoustic computer music: The Rumentarium project," *Organised Sound*, vol. 18, pp. 242–254, 2013.

[10] D. Sanfilippo and A. Valle, "Feedback systems: An analytical framework," *Computer Music Journal*, vol. 37, no. 2, pp. 12–27, 2013.