Kiilo logo?

dusjagr labs

# The Art of Hardware Hacking

## Physical Computing, Arduino and DIY Sound @ playaround workshop, Taiwan 2008

***Authors***

*Marc R. Dusseiller, aka dusjagr, Swiss Mechatronic Art Society, marc@dusseiller.ch*

*Tobias Hoffmann, aka kiilo, kiilo@kiilo.org*

version 0.2, 07/24/08

## The Origins of Hardware Hacking

If we take a look into the history of computing, we have to remember the times when a computer was an apparatus for doing scientific mathematical and logical operations. The computing machines were usually filling whole rooms in some science laboratory and only specific scientist were allowed to use them. But the history of computing has always to be put together with that of hacking. These early so called hackers, were students or other staff around the lab, who tried to get some time on these computing machines, without a specific scientific goal in mind, just to playaround on these modern and awesome machines. So without the scrutiny of their senior supervisors they started to program just for fun, because they had time to play around on the nice pieces of technology. And look at what these hackers were doing, they invented first graphics, word processors and visuals or even computer games on oscilloscopes (the game “tennis for two” is famed to be the first computer game, made in 1954). The playful abuse of the hardware, which usually was only available to some scientists, spawned completely new branches of industry, new art forms and even new cultural movements, such as the open source communities.

In general we want to state that many inventions and new art forms come from the fact that you do not want to except an established or commercial product as it comes from the developers. You want to tailor it to your own creative needs, you want to change it, you want know what is inside and you want to be able to share it.

## Open Hardware

Most people know already about open source software and the movement of open source communities. Various open source software (FLOSS) are widely used by students and artists alike. The major benefit of virtual software or code is that on a digital computer you can always create a completely identical copy of it. This gets much more difficult when we talk about physical hardware, which is the topic of our workshop. There has also been attempts recently to be able, similarly to the open source communities, to share knowledge about the hardware, change what others have been developed and communicate all the changes and improvements.

One example, which has been widely used now in many media art schools and other media labs, is the arduino platform. The arduino is not only a piece of electronics, its a whole framework and all the information about it can be openly accessed. The arduino framework consists of the following: First the hardware, an atmega microcontroller-board containing several inputs and outputs, a usb-to-serial converter and programmer, a voltage converter and some other electronic parts. The whole design and information is open access, meaning you don't have to buy it, but yo can easily build one yourself, change it to fit your own needs. Second, the programming environment, to write the code and compile and burn it onto the microcontroller. And finally the community and the web platform, where many instructions, developments, how to's or educational workshop syllabus are shared and discussed.

## Creative Control of Physical Objects

The area where we are teaching, physical computing, is wide, and being creative in that area is challenging for good reasons: have a good idea, see what is possible, find a solution for something what isn't possible.

It is for us with the experience with other students always obvious, without understanding of the *material* and *form* creativity isn't possible. The *material* in physical computing is dealing with electronics and mechanics (sensors and actuators), computer programming (basic languages such as c, assembler and processing or advanced graphical languages like puredata) and the interfacing/connecting between the two. The *form* results from the open minded and artistic approach to the field, with the innate creativity and the personal background. As creative persons we have to bridge the gap between calculus and creativity, something which is on one hand exact and a rigid technology and on the other hand combine it with the undefinable, blurred field of aesthetics and (hopefully) new forms.

The Physical Computing workshop was done within 3 days, albeit the area is big, we could extend this to 4 weeks or one semester. What was provided in the 3 days is almost only a small peek into an exploding field of activities. Our personal goal is to transcend the status of people from users of applications and technical gadgets to inventors of their personal ideas and purposes they have. Let us address this with a term which is always a purpose of all artistic activity: it is about inventing new forms. Form is not to mix up with activities giving clay, paper, stone or images another condition. I want to point out: form is making these conditions possible at all.

## Impressions during the workshop

First we started with a personal introduction, who we are, and showed video examples and a small real toy to show the students what could be achieved by physical computing. We were mainly using the arduino platform and puredata for our experiments, so we had to make sure to get the software running on all the computers. To get in touch with some basic electronic parts, we spent the evening playing around with a small oscillator based on a NAND CMOS integrated circuit chip. This gave a physical feeling of what electrical current, voltage, resistance or capacity are doing. The evening ended up in a noisy loud cloud of 1bit square wave chirping, beeping and screaming.

On the next day we kept the circuit and tried to connect it to arduino. With the famous “hello world!” example, a blinking LED, the same code could also used to control the sound device. We tried to give the students a general understanding of how to connect things and how to control physical objects. Just by replacing the LED output with the input of the NAND gate or even to power the breadboard. The next step was how to control this by the computer. Pduino is a small ready made “LEGO brick stone like” chunk of software, which converts the arduino board to a simple input/output device and be able to control it with puredata. One small button and our sound device was controlled by the mouse. One pure data object added and our sound device was screaming on key-press. Already here the possibilities are endless. Things to control! It could be an LED, a sound device, a light-bulb or a 110V plug and everything could connected to it. Also with pduino already running, we where showing how to sense data from the outside world – by using a potentiometer or a light dependent resistor (LDR). We tried also to understand programming the arduino – not only use it as a ready made simple I/O-board. Because more complex sensors or actuators, such as servos or ultrasonic ping sensors, could not yet be controlled by this simple approach. We where providing a software library to send data to arduino from puredata and to receive data from arduino within puredata. The code could then be extended by the students own functions.

On third day we were finally hacking some toys. How to use the motors and LEDs inside – some toy worms were crawling on command. Some sound devices in the toys we tried to hack into – by electronics or by the use of the arduino. In the afternoon, we spent some time to get into touch with student's group projects by getting an overview of their ideas and giving them tips of how things could be done.

## Conclusion

The time was very short and we would have liked to spend more time for creative playing around. Creativity cannot be forced and final results on a short call can hinder the creativeness of students. Detours in the path of problem solving are necessary to achieve a deeper understanding of the tools used and of course the idea itself could be changed along the way.

Our recommendations for a next course would be the following: First for ourselves provide more brick stone like examples to provide a feeling of “It's possible, I can do this and thats fun!” Give the students more time and more freedom to explore and to concentrate. Do not take brick stones as the only option to create something – point out that brick stones are just one convenient way to build something, but the possibilities are predetermined by their form. Instead create your own tools and knowledge and share it with others. We see this as a way of how students can actively participate in society as artists, designers or engineers.

And also we would like to thank here all the people who helped making this happen: First a big thanks to all the students, who where really great, innovative and worked so hard they didn't even sleep! And of course to all our teaching assistants, who always helped us translating and making sure the students get it right. And finally the other teachers and all the organizers from ITRI, U19 and National Chiao Tung University and most of all thanks to microplayground, who made the workshop possible.