The Apple Barrier:
An open source interface to the iPhone

Tom Keene
Independent Artist
47 Hardel Walk, London SW2 2QG, United Kingdom
tom@theanthillsocial.co.uk

This paper presents the ongoing process, challenges and approach of integrating open source hardware with the iPhone. The aim of the project was to create software and an accompanying device using Arduino, an open-source electronics prototyping platform based on flexible, easy-to-use well documented hardware and software extensively used by artists. The iPhone was chosen primarily because of it ubiquitous presence but also because of creative possibilities due to computational power, networking functionality, inbuilt sensors and storage capabilities. However, restrictions and complexities to the way we can interface with those technologies mean many of those possibilities are lost. The ecology of open source tools available to digital artists make highly technical environments accessible to low technology users, yet the closed environment provided by Apple, used by the vast majority of owners, force corporate agenda onto the ways we choose to communicate. Users are actively discouraging from understanding how these tools work, be that through hardware interface, technical language, levels of knowledge or literal licensing restriction. The process of building an interface to these technologies reveals the restrictive mechanisms at play and provides insight into ways they may be challenged or subverted.

Apple iPhone. Ubiquitous technologies. Open source. Bio-sensors. Locative media. Hacking.

1. INTRODUCTION

This paper is an investigation into the corporate control mechanisms prevalent in ubiquitous mobile technologies. My interest in this issue arose during my involvement in a University of Sussex research project "Supporting Shy Users in Pervasive Computing". This is a cross-disciplinary EPSRC funded project, bringing together Informatics, Sociology, Human-Computer Interaction and Art.

As a digital artist in this project, I developed a hardware interface to the iPhone to measure changes in sweat level as a means to indicate psychological or physiological arousal. During the build process it became apparent that corporate control mechanisms specific to the iPhone blocked my attempts to interface open source technologies with this closed source device.

This paper is not an account of the use of this interface in the shyness project. The focus is on closed corporate control mechanisms at play within ubiquitous mobile technologies maintaining elitist control and access.

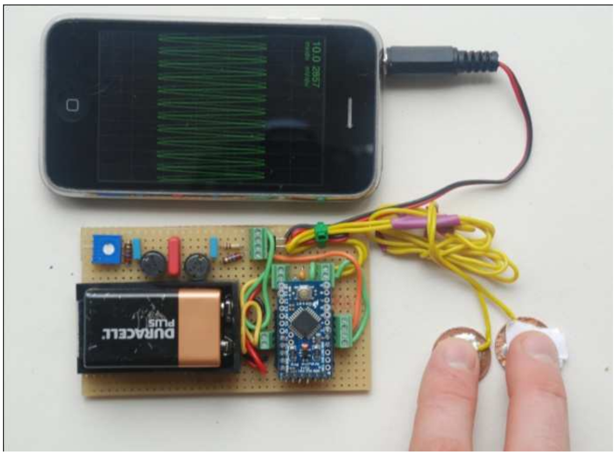


Figure 1: *Device attached to the iPhone*

Using an engineering exercise as a way of examining these issues, I illustrate and critique the role of restrictive licensing models in a creative process and examine future implications and ethical considerations that arise as these technologies evolve within a corporate landscape. The research was initiated with the following questions in mind:

- What creative possibilities emerge from building a distributed sensor network?
- What ethical considerations come into play?
- What technical difficulties arise?

To contextualize my approach it is helpful to outline my work as an artist. I write software code and use engineering techniques as part of my creative tool-set. I investigate technologies, their application, and social impact. I often act as a technological translator where I mediate between groups with wildly varying levels of technological understating. I have immersed myself in the language and practices of software engineers, electronic engineers and microbiologists. I take this knowledge into my work with youth and community groups of all ages, where my approach to technology is a study of what we are making as a culture, why we are making it, who has access to it and why.

Full instructions to build the hardware device have been provided at the end of this paper, including a circuit diagram and descriptive text.

This paper was inspired by a period of artistic research undertaken by artists Tom Keene, and Anna Dumitriu. This project is part of the University of Sussex Project "Supporting Shy Users in Pervasive Computing" an EPSRC funded project bringing together Informatics, Sociology, Human-Computer Interaction and Art.

2. THE APPROACH

The project arose as the result of a discussion between the artist Anna Dumitriu and myself. We were interested in seeing what would arise through the creation of a device that shared data from a Galvanic Skin Response (GSR) sensor across multiple users utilising their own mobile devices. A GSR sensor measures electrical conductance of the skin, which varies with the presence of sweat, indicating changes in stress level. Once the device had been built, our aim was to devise a series of workshops, performances and discussions that explored individual response and experience of these networked technologies.

We decided to build a low cost device that plugged into an iPhone using Arduino, "an open-source electronics prototyping platform, intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments" (Arduino, 2011). We decided to use the iPhone because of its ubiquity, processing power, network and sensor capabilities. We then intended to distribute data with Pachube, a free web based service that allows users to "Store, share & discover real-time sensor data" (Pachube, 2011). All of these components promised rapid, low cost prototype development.

The essential software components were to consist of code residing on the Arduino that would interpret and send sensor data to the iPhone. The iPhone application would be written with OpenFrameworks, "an open source C++ toolkit for creative coding" (OpenFrameworks, 2011) that would receive data from the Arduino and distribute it to the Pachube web service.

For each of the components cursory research revealed there were readily available examples online for what we wanted to achieve. So, as I was tasked with building the device and writing software, I felt confident taking this route even though the iPhone was the least familiar component of what we wanted to achieve.

What was eventually discovered was that the open source and open standards tools provided by OpenFrameworks, Arduino and Pachube did rapidly enable us to achieve our goals, but the closed environments of the Apple iPhone and App Store thwarted and delayed almost every aspect of the technological endeavor. And while this was frustrating, the process clearly exposed key themes, issues and challenges that surround control of mobile technologies and personal data.

3. THE NARRATIVE

3.1 No hacking

It was decided that an approach would be investigated that did not require the iPhone operating system (iOS) to be hacked (otherwise known as "jailbreaking"), because the majority of iPhone users do not use hacked phones and I wished to use and investigate the same technologies that they use. It is also worth mentioning that in the United Kingdom and European Union, jailbreaking an iPhone remains a legally grey area, though it seems highly unlikely that legal proceedings would be undertaken for such an endeavour.

The decision to use the (not hacked) base operating system (IOS) meant that the approach would need to adhere to Apple's digital rights management (DRM) structure. DRM imposes restrictions on use of the operating system and access to hardware functionality. This includes a centralized approvals and distribution process for applications, in addition to restrictions on hardware and software interface.

If we wished to distribute our custom iPhone application to an audience beyond 100 users, it would need to pass an Apple implemented censorship process, as is permitted by the apple developer program. And while it was likely that the application would be passed, there was always a possibility that it would not, as has happened with other submissions to the App Store. We would also need to enroll on the iPhone developer program, costing $99 per year, which legally binds developers to draconian non-disclosure agreements. Without paying this fee you are not able to freely distribute applications to all users of the iPhone, nor are you able to voice any complaint should submissions to the App Store be rejected as rejection is bound by a non-disclosure agreement.

3.2 A web-based approach

Wishing to avoid the headache of non-disclosure agreements and the App Store, an initial prototype focused on developing a web-based approach utilising HTM5, a broad set of new technologies being implemented in web browsers allowing complex and standards compliant applications to be built within a browser interface. This approach would avoid developer registration or distribution via Apples App Store. Users would simply need to visit a web page in order to receive and transmit data. This was made possible because of a low cost add-on to the Arduino board called the “wi-shield” (Async_labs, 2011) a device that is able to create a wireless connection to the iPhone. The theory was that the iPhone would be able to simultaneously maintain two wireless connections, one to the custom device and another to the 3G-telephone network allowing data to be uploaded and re-distributed via Pachube.

Initial experiments were promising in that a wireless connection could easily be made to the sensor and that a purely web-based application could fulfill the requirements of the project. Advances in HTML5 and support for associated technologies on the iPhone meant that a web-based application could be made to download and look like a native iPhone application which would be a huge usability benefit to end-users. However, after extensive testing it became apparent that while data could be transferred this way, the simultaneous connections were not reliable, requiring continual re-setting of network preferences on the iPhone. These problems were compounded by an inability to effectively debug what was happening due to the locked down environment of the iPhone, where we are prevented from installing applications that would help diagnose the problems.

3.3 Direct connections

Back to the drawing board and it was clear that I would have to register with the developer program to achieve the aims of the project. Previous research had revealed an external accessories API (Application Protocol Interface), enabling the iPhone to communicate with external devices. This initially promised to be straightforward to implement utilising a minimal set of hardware and software components.

It quickly became apparent that to use this API you have to sign up and get accepted to an additional developer program titled “Made for iPhone” (MFi), which was also protected with non-disclosure agreements. Acceptance on the program gives permission to purchase a proprietary authentication chip sold exclusively by apple. This chip would need to be embedded on any custom device. The MFi program is clearly aimed at commercial organizations and actively blocks open hardware integration, meaning this author was rejected from the application process.

3.4 An alternative approach

With avenues for experimentation closing down, it was clear that alternative methods would need to be sought. A number of others had also come to the same conclusion and people had started to publish alternative strategies for connecting hardware to the iPhone without having to be accepted as part of the MFI program. The solutions centered on a similar technique called frequency shift keying (FSK) which utilized the iPhones ability to record and generate audio, functionally that developers are allowed to use under the standard iPhone developer program terms and conditions. The FSK technique, first discovered in the early 1900s, encodes digital signals within an audio signal.

A paper originating from the university of Michigan (Kuo et al., 2010), clearly indicated this method would work. A book entitled “iPhone Hacks” (Jurick et al., 2009) provided instructions for building hardware and software, though unfortunately, both those resources did not illustrate how to implement the solution using Arduino hardware. A Japanese language website (Burogumappu, 2011) translated into English via ‘Google Translate’, revealed instructions for building a device with Arduino. I followed these instructions and subsequently built the device but communication between devices was erratic and the limit of my understanding of electronic engineering and C++ programming techniques had been reached.

3.5 The simple solution

At a point where it looked like all possible solutions had failed I came across a paper produced by “lab3: The Laboratory for Experimental Computer Science” in Cologne (Nawrath, 2011). They outlined a simple method to generate a very precise sine wave tone using the Arduino. It mentioned that this tone could be used as the basis for FSK or simply for generating an audio tone. I also discovered example code for detecting the frequency of an audio tone within the OpenFrameworks coding environment, which could be used as a basis of the iPhone application. From these examples I was able to build a prototype hardware interface and associated iPhone application that read a sensor value as mapped to a simple rising or falling tone. High resistance in the skin generates a high frequency tone and lower resistance a low frequency tone. This technique did not need to use the more complex FSK approach, though the hardware would facilitate this more powerful approach as my skill level increases or further solutions became apparent.

4. THE BARRIERS

My attempt to interface with an object and culture was barred on grounds of experience, money, access to knowledge, influence and technological understanding. However, most importantly, my freedom to cooperate, share and learn from others was also being barred. The technologies themselves were not the main barrier, but rather the licensing structures, distribution mechanisms and non-disclosure agreements that served to control my interaction with these technologies and other users. Had I attempted to connect to a laptop rather than an iPhone, construction of a sensor device would have been a much easier task. It was not possible to connect the iPhone because Apple blocks this in order to maintain a monopoly. This authoritarian control is concealed by Apple's rhetoric, that keeping these technologies 'locked down' facilitates consistency of design and maintains high technological standards. What it actually seeks to maintain is an effective marketing system.

4.1 Creative barriers

The creative tools of media technologies have expanded to include any combination of electronics, engineering, software, APIs, data and telephone networks. The promise of Apples marketing strategy is that their technological tools inspire creativity, yet new forms of creativity are stifled. Legal language is used to discourage experimentation and hardware features are made redundant to protect consumer markets. A strong message is received that creative endeavors need to be of a particular standard as judged by a corporate organization. If we do not live up to this standard or sets of rules, then anything created outside of the given structure is deemed wrong, with possible legal implications.

This project could have been implemented using alternative technologies, but rather than take the easy option, I saw an opportunity to analyze and experience the control structures inherent within a mainstream technology. This analysis has revealed potential ways of subverting these control mechanisms, enabling alternative networks and data routes. No better example of the importance of this is with the recent uprisings in Egypt, where citizen net access dropped almost to zero. In response, alternative means of communication were made available via international dial-up telephone numbers and offers to forward voice and Morse code messages sent via ham radio.

4.2 Dystopian futures

Mobile networks now cover a huge proportion of the planet allowing connectivity from almost any location on the globe. Changes in Internet protocols, specifically IPv6 have increased the maximum number of devices able to simultaneously connect via the Internet, paving the way for huge numbers of sensors and mobile technologies to share data. Understanding the implications of this explosion of connectivity is paramount. It is not hard to imagine what could happen when biological data and devices (heart rate monitors, breathing, electrocardiograph, pacemakers) mediated by mobile devices are accessed for economic gain. The reputation of private medical industry diminishes confidence to in trusting companies with our biological data and devices that physically connect to our bodies. Yet the private medical industry is precisely the market that Apple has identified for the development of external hardware for the iPhone. A comparable trajectory can be seen with the evolution of energy sensors in the home. For example The Electronic Frontier Foundation has highlighted potential surveillance and civil rights issues in response to companies having access to something as seemingly benign as energy usage in the home:

> Without strong protections, energy data can and will be used in ways that will hurt consumers. Marketing companies will desperately want to access this data to get intimate new insights into your family's day-to-day routine, and it's not hard to imagine an insurance company interpreting the data in a way that allows it to penalise you. Our privacy rights should be strongest in our home (Jeschke, 2010).

These problems of civil rights exponentially increase as networked sensors move from the home into new territory: the body. If a wide proportion of society is blocked from accessing or engaging with these technologies then this is cause for concern, as elite groups then have ultimate control of how we communicate at the most intimate level.

5. CONCLUSION

We exist in a new age of mobile technologies. Small powerful computers able to produce all forms of digital media and instantly distribute data across network, both with or without our knowledge. The doors are open for private medical industry and larger corporate organizations with access to huge resources to develop interfaces to these technologies. It is extremely difficult for smaller independent organizations and individuals to produce experimental hardware that interfaces with the mass consumer version of these devices. It is vital to investigate and experiment with these technologies precisely because they are so personal. They live in our pockets and are next to our skin. They are capable of sharing the most intimate of information, from whispered voices, to a sensor attached to our bodies revealing a heartbeat or changes in our stress levels. There is an imperative to open up means of experimentation with the underlying technologies and structures that facilitate these new forms of communication.

That privilege should not be handed solely to the engineers of corporations or even technological 'geek' elites.

> Engineers tend to design platforms for other engineers, not for artists, weirdos, or kids who want to connect stuff up in a simple way to share an idea (Torrone, 2011).

Therefore, access to technology goes beyond merely the ability to afford a device. Access is about providing literal hardware interface and sets of tools and solutions that people at all levels can experiment with and understand.

6. THE CONSTRUCTION

The following instructions have been amalgamated from examples that have been posted across many websites online. The construction, outlined in Figure 2 consists of four elements:

a. The iPhone.
b. The Arduino microprocessor.
c. A filter circuit.
d. A galvanic skin response (GSR) sensor.

The sensor connects to an Arduino board that generates an audio tone that is mapped to value of electrical resistance in the skin. A filter removes a 32KHz sampling frequency contained within the output signal, which is sent to the microphone input of the iPhone, which then processes the signal.

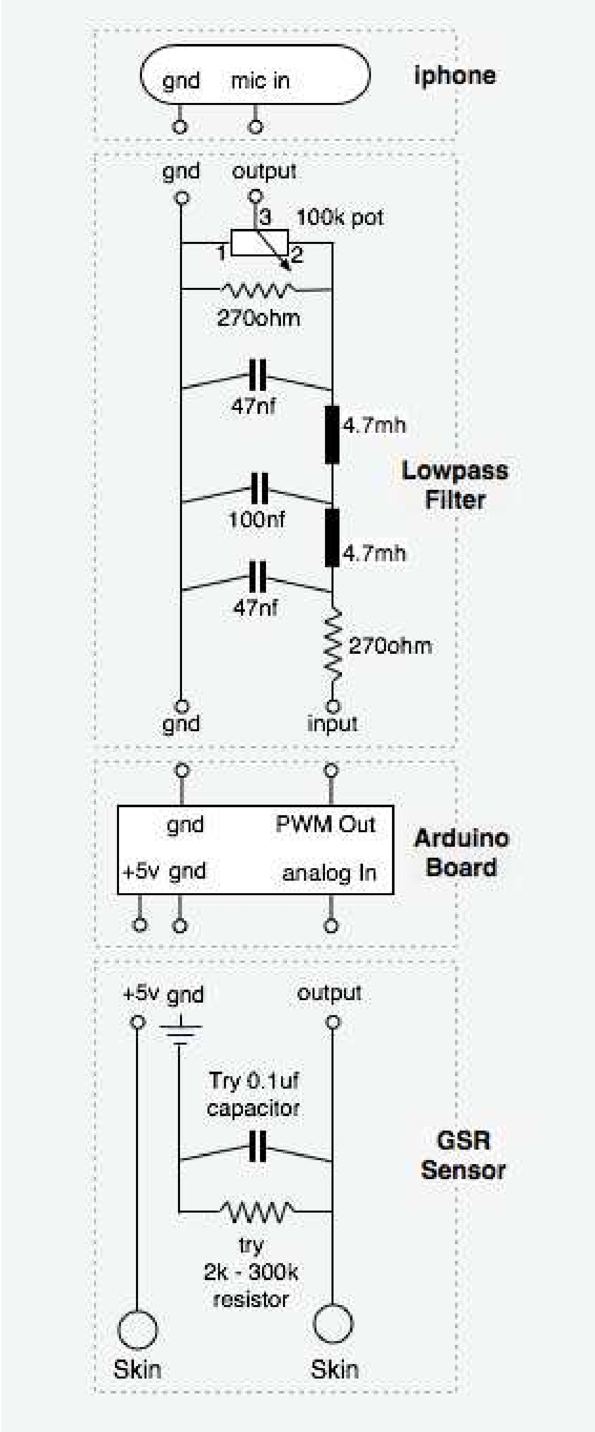


***Figure 2**: Circuit diagram*

6.1 Tone generation and filter

The filter is a design outlined by lab3, at The Academy of Media Arts Cologne. Taken from their website:

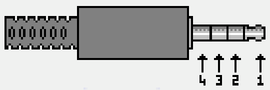
> ... we describe how to generate sine waves with an Arduino board in a very accurate way... The frequency range reaches from zero to 16 KHz ... useful for music and sound generation another range of application is ... in telecommunication [where] the DDS Method is useful for instance in frequency of phase modulation (FSK PSK). (Nawrath, 2011).

Filter parts list

2X 47nF 100V 5mm Capacitors
1X 100nF 63Vdc Capacitor
2X ELC coil inductor 4.7mH 0.12A
2X 270Ω resistors
1X 100K resistor
1X 100K Trim Pot

Table 1: *iPhone headphone connections*

Pin	*Name*	*Description*
1	*Tip*	*Left audio*
2	*Ring*	*Right audio*
3	*Ring*	*Common/Ground*
4	*Sleeve*	*Microphone*



4 pin 3.5mm (2.5mm) plug connector

6.1 Galvanic Skin Response Sensor

A simple proven design, though the contact points, tend to corrode if used over long periods of time so need to be refreshed with a quick rub of sandpaper. To create the skin contacts, I simply sanded the pennies then soldered wires to them, sticking them to the skin with medical tape.

I have seen many examples of this kind of GSR sensor, which have recommended any resistor value between 2k and 300k. My experiments performed best with a 10k resistor. I used copper pennies as the skin contacts, as these performed better than other solutions that had been suggested such as foil tape. better than other solutions that had been suggested such as foil tape.

Parts list for GSR sensor

2 X 1 pence pieces
Sand paper to refresh contacts.
Plasters to fasten sensors to skin.
1 X 2k-300k resistor.
0.1uf Capacitor.

Figure 3: *holding the device*

6. REFERENCES

Arduino (2011). http://www.arduino.cc (2.3.2011).

Async_labs (2011). http://www.asynclabs.com (2/2/2011).

Burogumappu (2011) Play with Arduino – iPhone audio jack for use through, *FC2*. http://arms22.blog91.fc2.com/blogentry-350.html (1/4/2011).

Jeschke, R. (2010) EFF Advises California PUC on Smart Grid Privacy Protections, *EFF*. http://www.eff.org/deeplinks/2010/10/eff-advises-california-puc-smart-grid-privacy (1/4/2011).

Jurick, D., Stolarz, A., and Stolarz, D. (2009) *iPhone Hacks*. O'Reilly Media / Make. USA, 4 September 2009.

Kuo, Y.-S., Schmid, T., and Dutta, P.. (2010). Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface, 2 September 2010.

Nawrath, M. (2011) *Arduino DDS Sinewave Generator*. Lab III, Cologne, Germany. http://interface.khm.de/index.php/lab/experiments/arduino-dds-sinewave-generator/ (29/5/2011).

OpenFrameworks (2011) Open Source C++ Toolkit. http://www.openframeworks.cc (2/1/2011).

Pachube (2011) Connected Environments. http://www.pachube.com (2/3/2011).

Torrone, P. (2011) Why the Arduino Won and Why it's Here to Stay, *Make: Technology on your time*. http://blog.makezine.com/archive/2011/02/why-the-arduino-won-and-why-its-here-to-stay.html (1/4/2011).