6 December 2007

Distributed wireless duplex audio networks in a travelling art work: the Megaphone Project

Jesse Stevens, Madeleine Flynn and Dr Tim Humphrey, Melbourne Australia

Abstract

The megaphone project is a freestanding and portable sound and visual installation with flexible, interactive and changeable sound content. A field of red "megaphones" on simple stands recall an iconography of historical sound diffusion and reinforcement, while inviting physical and emotional responses from those who encounter it.

To facilitate a distributed and portable full-duplex sound network, we have incorporated PDa (PureDataAnywhere), running on HP hx4700 PDA's and *Gumstix* computers that are mounted underneath each "megaphone". This enables a stable network of portable devices capable of interlinking the entire field.

Each node of this field then has the capability to create a higher level of interactive sophistication, but also sends and receives audio from a central PD patch that stores and re-broadcasts sound inputs across the field, with the option for electro-acoustic realisations related to the immediate acoustic and cultural context.

Keywords: *Pda* (Pure Data Anywhere) - *Gumstix – Familiar Linux* - Portable installation - Multiple soundcards - duplex wireless audio network – inexpensive devices – acoustic field

Introduction

The megaphone project arises from the artists' concept that allows audio input from audiences of all ages to be communicated across a portable field, and also stored temporarily or permanently as a record of interaction and encounter in various contexts. From this simple premise, we created acoustic "megaphones" and mounted these on stands, allowing free play in the acoustic realm as they "broadcast" speech, song and all other manner of utterance.

Concurrently with the visual and acoustic audio play that the piece creates, we have placed portable devices in the base of each "megaphone", together with an external speaker and microphone mounted within the top of the megaphone stand. Thus a full-duplex network of wireless audio is created that is stable and also flexible in both siting and in operation.

We have installed the megaphone project in several Australian cities, most recently appearing at the Australian Performing Arts Market (APAM) in February 2008, and previously at the FINA World Swimming Championships Cultural Festival Melbourne, and the Awesome Arts Festival Perth in 2007.

The work creates a spatial-audio "cultural field" that invites those who encounter it to express themselves gesturally and verbally. In this sense it approaches the kind of spatial musical instrument discussed by Blesser and Salter [Ble01], although these authors are generally speaking about the hyper-instrumental character of enclosed spaces of different kinds. In our case, we create a virtual acoustic space, constituted by those who encounter it as much as the characteristics of its ambient acoustic ecology as determined by its siting.

As we have not encountered similar distributed and full-duplex audio networks, we continue to explore various inexpensive and flexible options for implementating our idea.

Audio and WIFI architecure: asserting control over the audio field

Our desire to install the megaphones in many contrasting public spaces has meant that we have encountered a wide range of different environmental conditions for WiFi as a result of interference from microwave transmission, cordless phones, and other WiFi networks on the 2.4ghz range.

The existing installation features a centralised design, with a central laptop running *Debian Etch* and *Pure Data Extended* [Puc01], that is connected by ethernet to WIFI access points (at this stage the D-link DWL-2100AP. At each portable "megaphone site" embedded linux devices, that are a combination of *Gumstix* computers and PDAs (generally the HP IPAQ

range), the WiFi access points give coverage to, and depending on the configuration of the site, load-balancing across the field.

Earlier incarnations of the work using hacked versions of cordless phones over a variety of broadcast frequencies (2.4 to 5.8 gHz ranges), led to a somewhat mercurial audio performance that most commonly resulted from the regulation of input and output in our telephones. This common feature in communication devices allows the user some audio feedback of his/her own voice while hearing the voice on the other end.

The acoustic effect of wildly-varying outdoor sound environments is exacerbated by the acoustics of the megaphone objects which effectively concentrate the ambient sound reaching the microphone. As a result, many of the cordless phones are unable to transmit audio from our matrixed network, as they reasonably interpreted that an uninterrupted audio stream is reaching their microphones, and so attenuating any output.

A sometimes pleasing phase and unpredictable audio re-broadcast operates in the system that can be tweaked by adjusting the threshold levels in our "control" (*sic*) *PureData* software environment that records from the megaphones into temporary and more permanent buffers samples of varying lengths.

Even though the mercurial performance of our hacked telephone system has a certain artistic appeal, we do want to assert control over the performance of the audio, enabling a range of implementations of the project, including indoor and outdoor installation, and moving towards the creation of an interactive "megaphone instrument" that is capable of performing over a wide range of dynamic and frequency spectra.

To re-assert our control, we have now moved towards linux embedded intelligent devices (PDAs and *Gumstix* microcomputers) that give us customised control over the audio performance of the work.

Component features of the architecture

The architecture of the work features a number of component features that we have found to be necessary due to our choice of multiple and very inexpensive consumer soundcards, and the possibilities of *Pure Data anywhere (PDa)* [Gei01] and *Pure Data (PD)*:

1. Scripting. To ensure ease of startup and stability in varying environmental conditions(eg microwaves, cordless phones, other wifi networks – 2.4ghz) we have created scripts that start multiple instances of *PD* on the central laptop. The *Gumstix* are scripted to begin *PDa* as a system daemon on powerup. The PDAs need to manually start *PDa* using the touchscreen.
2. When operating the hacked cordless phone system we need multiple channels of audio at the central laptop. For this we require the multiple instances of *PD* due to it's limitations on how many soundcards it can handle per instance (4 individual in and 4 individual out regardless of channels per card). Depending on the situation we either start the multiple instances in X with a script or start them in ttys 2-6 leaving tty1 open for control and tty7 (X) open for graphical interaction with the patches. Message and audio passing is then done through use of the loopback adaptor (127.0.0.1). This also allows the processing load to be spread over multiple processes and a neat simplistic graphic interface.
3. The multiple instances of *PD* then can enable a large number of audio nodes, and being limited by both what cards are available and our budget (as well as our philosophical concerns!), we have created an array of consumer soundcards. We use inexpensive USB 2.0 hubs and extremely inexpensive USB sound cards to create a "web" of soundcards, rather than opting for the often-assumed solution of an expensive multi-channel soundcard that may or may not provide linux driver support at this time.
4. Wireless access points. Depending on the size of the field it is necessary at times to use multiple access points to provide enough throughput for the streaming audio. There are two methods; multiple access points with the same SSID so that clients roam between them depending on signal strength, or multiple access points with different SSIDs and

clients being grouped manually to each access point for more control. Each has strengths and weaknesses but for the sake of having more control we usually opt for different SSIDs.

The final components of the architecture of the work are the embedded linux audio devices.

Embedded linux audio devices

The embedded linux audio devices represent a focus of our develoment efforts in the megaphone project. We have used two different devices:

1. *Gumstix*

The Gumstix is a very small (size of a stick of gum), lightweight, low power consumption, Intel *ARM* processor-based device. Linux is well supported by the company that builds them as well as a large community. They range in speed and storage/RAM capacities, as well as having a unique click together expansion card setup to extend capabilities (Wifi, bluetooth, audio, GPS, SDcard storage). *PDa* has been ported without GUI to the *Gumstix* along with many other tools and useful utilities but more support is needed so that it can be added to the buildroot environment.

The buildroot environment is basically similar to those for many other embedded devices. It builds the toolset to cross-compile code into suitable binaries for the target processor which it then uses to build the rest, which include *UClinux*, *Busybox*, basic scripts and other required utilities for the *Gumstix*. Once it has completed this it goes on to to build the created filesystem into a *jffs2* (a suitable image filesystem for flash memory storage) filesystem image. The whole process can take a few hours from scratch, but once the toolchain is built (cross compiling gear) the process is much faster.

From here we then need to use a means of uploading this new image to the *Gumstix*. This is easily achieved via a serial (RS232) connection, and we are left with a new image built by hand.

The hard part is that the *Gumstix* by default doesn't include a package manager to install and remove programs/packages on the *Gumstix* itself, and because of size constraints there are no tools onboard to build packages on the running *Gumstix*. So you need to know what you want to use your particular build for before you use the *Gumstix* itself unless you are prepared to build and move binaries over to the *Gumstix* post-upload.

The lack of a GUI in the current *Gumstix PDa* port is also a major limitation given that *Puredata* is fundamentally a graphical programming environment. However we are currently working on a full *Gumstix* port of *Pda* that will enable us to access the GUI via forwarded X over SSH that will run on a network-connected device.

2. *Familiar* on many PDAs

Familiar [Fam01]is a linux distribution built for many different PDAs with the main support being in the HP IPAQ range. This distribution includes the *ipkg* package management system [Fru01] which allows easy installation of many useful tools including *PDa*. It can run a small GUI with either OPIE or GPE as the window manager. Simple and easy to use, this platform is an ideal companion with interactive works such as the Megaphone Project. In our model we use it both as a base for the megaphones as well as a roaming interface for the work allowing simple changes to be made in the *PDa* patches in use throughout the field.

Both of the above boast 802.11b/g capabilities and enough processor speed to allow streaming of audio in and out over the network. Along with the above – we also utilise the netsend~ and netreceive~ objects in *PD* [Mat01] to allow passing of the audio throughout the network with ease.

Of the two embedded devices, we have encountered some frustrations using the *Gumstix,* related to engineering a power source and figuring out charging methods as well as a stable method to mount the units to the required level of robustness to deal with a public moving and standing on each portable megaphone stand. So we have found that the PDA route is an easier task. They are mass-produced items that are cheaper in total cost, not requiring external

batteries or custom mounts, just a small cradle that held them in place and a plug to reach the speaker and microphone.

They can also then double as handheld units should we change the ratio of megaphones to PDAs which allow us to carry out a different style of performance/installation whereby we can create a higher-level interactive audio experience from a megaphone for the audience.

Conclusion

The Megaphone project is a continuing exploration into wide and local acoustic and electro-acoustic horizons. Each individual megaphone is a discrete device granting permission to make a public vocal utterance. They may also be encountered as a visual and sonic sculptural object that contains some unexpected broadcast audio content, whether an echo of their own vocalisations, or a "re-mix" of the accumulations of others.

We aim to develop the audio so that they can operate as "choral" instruments in a manner inspired by the beautiful work*, 40-part motet* created by Janet Cardiff [Car01]. We see the megaphones as a visually-beautiful field, encountered by passers-by as an embedded field sculpture that also comments on the acoustic ecologies of the modern world.

As independent artists, working outside of institutions or large corporations, we investigate inexpensive and low-powered options for creating sound works, which partly forms, or is formed, by our aesthetic of democratic and de-centralised cultural production.

Many of the technical tools that we need to develop our work are only to be discovered within the open-source community. In addition, the reliability and open-ended character of *Pure Data* and its *PDa* and *PD-Extended* flavours have allowed us to test a range of embedded devices and to customise our computing platforms to achieve the flexibility and portability required. We hope that our work with portable linux devices will help to re-invigorate activity in this area which has been comparitively quieter than in previous years.

Acknowledgement:

The authors would like to thank Jim Sosnin for his essential early advice for this project, and also for his inspirational work and support over many years. Thanks also to the many people in the open source community who have built the tools that make our own modest contributions possible.

References:

[Ble01] Blesser, Barry and Linda-Ruth Salter *spaces speak, are you listening? Experiencing aural architecture.* Cambridge, Massachusetts: The MIT Press, 2007. Chapter 5.

[Car01] Cardiff, Janet. *The Forty-Part Motet: a reworking of "Spem in Alium" by Thomas Tallis 1573.* Installation. 2001.

[Fam01] the Familiar Project. **http://familiar.handhelds.org/**

[Fru01] Frullani, Pierluigi, Carl Worth and Steve Ayer *ipkg – Itsy Package Management System*
http://handhelds.org/moin/moin.cgi/Ipkg

[Gei01] Pure Data anywhere - Gunter Geiger
http://gige.xdv.org/pda/

[Mat01] Matthes, Olaf. netsend~ and netreceive~ objects
http://www.nullmedium.de/dev/netsend~/

[Puc01] Miller Puckette, Harry Castle, Krzysztof Czaja, Mark Danks, Christian Feldbauer, Guenter Geiger, Kerry Hagan, Trevor Johnson, Fernando Lopez-Lezcano, Adam Lindsay, Karl MacMillan, Thomas Musil, Toshinori Ohkouchi, Winfried Ritsch, Vibeke Sorensen, Rand Steiger, Hans-Christoph Steiner, Shahrokh Yadegari, David Zicarelli, Iohannes Zmoelnig and many others. *Pure Data.*
http://puredata.info/