Interactive Computer Music with Graphical Control by Using Android Phone for Audiovisual Installation Art Applications

Tsung-Ching Liu[1] and I-Hon Lin.

Abstract Interactive music using Max/Msp has been extensively used. The Java environment allows for interface of Max/ Msp to other software and firmware applications much easier. In this paper we will demonstrate the integration of Android phones, Arduino micro-controller, Max/Msp and Processing under Java environment for controlling the music and graphic image simultaneously that can be used in installation art applications. The system uses a HTC Desire Z to control a pre-designed Max/Msp patch running on a PC through Arduino interface. The patch is functioning as a musical effect controller of three facets: *volume, reverb* and *tremolo*. These three parameters of Max/Msp are then linked to control instantly the graphic images developed by the Processing [1]. We detailed the how-to-do-it procedure. Among them, the Amarino [2] APP is used for linking the HTC phone and Arduino; the oscP5 [3] is used for linking the Max/Msp and Processing.

Keywords: Interactive Computer Music · Max/Msp · Android Phone · Installation Art.

[1] Tsung-Ching Liu

E.E. Department, Chinese Culture University, Taipei

tcliu@faculty.pccu.edu.tw

1 Introduction

The advancement of new computer music technology for the last decade has drawn extensively interest by people and created great impact in music education, performing art and gaming. Many new media art performance that had used interactive audio-visual elements to create un-limited imagination and innovation attracts many computer music performers to follow. Those days, more and more people from diverse fields have jumped into this area and created many amazing live performance as they can be seen in YouTube. It is not long to wait before the new type of music performing or acting will be fascinating prevailed.

In our previous paper [4] we have shown the interactive music control by using Android phone in conjunction with Arduino [5] interface module and Max/Msp [6] successfully. Similar strategy also tested by replacing Android phone to Theremin [7]. Here we will extend our previous design to include visual control that reflects the music content. Namely, the controlling apparatus must be able to drive the video or graphic patterns for a kind of new media art presentation while music is playing controlled by the Android phone or the like. Players can interact not only by the music stimulation but also with the video or graphic sensation. This feedback system could create many interesting impromptu pieces.

Our designed system is depicted in Fig. 1. We will explain our approach as following: In Section 2, the Amarino App provided by the MIT Media Lab is reviewed. We use their MultiColor Lamp App and its Bluetooth connection setup; Section 3 reviews in how to interface the Arduino microcontroller to the Max/Msp software. Abundant resources with tested program codes are provided by the popular Arduino official website; Section 4 briefly states our previous work on integration of the android phone, Arduino and Max/Msp for the interactive music purpose; In Section 5, as the main issue of this paper, we describe how to add in Processing to our previous system that allows the Max/Msp to control the computer graphs generated by Processing. We will then conclude our work in Section 6 and point out the future possibility in installation art applications.

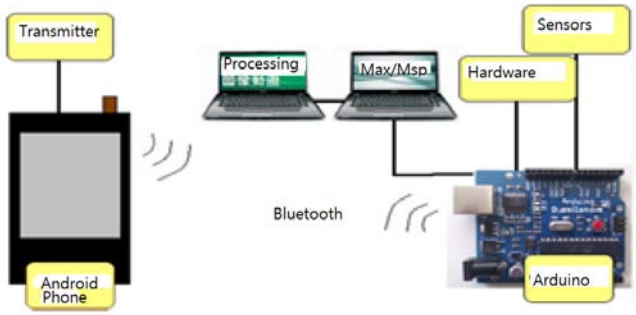

Fig. 1 An integrated system for interactive music with graphical control.

2 Amarino-Android phone +Arduino

In year 2007 Google announces the open source code Android platform for hand-held device. This platform is based on the Linux core and is adopted by the Open Handset Alliance to develop cell phone. The open source code platform allows users to develop various app based on the *Eclipse* [8] simulator.

Arduino is an open source micro-controller developed by Italian. The Processing is used for programming the code with the boot-loader design that allows software burn in. Many have used it to design interactive control applications since it is under an environment similar to the Java that facilitates its interface to many interactive computer music software such as Max/Msp, PureData, SuperCollider etc.

Amarino is a tool kits developed by the High-low Tech Group of M.I.T Media Lab which allows Android phone to transmit data and character to Arduino through Bluetooth. The installation of Amarino into cell phone is described in Fig. 2.

Fig. 2 Displays from the left: Amarino installed, search for the Bluetooth device, link to the Bluetooth module on Arduino, and monitor the data flow.

The original Arduino does not have Bluetooth built-in. In order to communicate Android phone we add the Bluetooth Mate Gold of Sparkfun to Arduino (Fig. 3). Detailed implementation can also be found in [9].

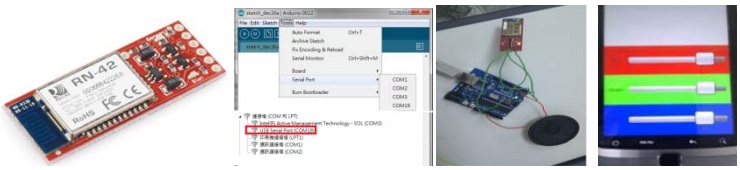

Fig. 3 Display from the left: Bluetooth Mate Gold, select the right com, Arduino with add on Bluetooth, MultiColorLamp in HTC Phone.

In order to use Android phone's faceplate image to control Arduino, a MultiColorLamp APP is added and is described below.

2.1 MultiColorLamp

There are two ways to install the *MultiCorlorLamp* APP into the Android phone. One is from *Eclipse* and the other is to use the packed APK. Under *Eclipse* we first create a new *Android Project* to import the *MultiCorlorLamp* program provided by the Amarino. Then identify the version of the cell phone used and set the parameter on *Eclipse*. We need also to be cautious that the series number of the Bluetooth module used in the Arduino is also typed in the setup. In this way the message sent by the phones then can reach to the Arduino. Last, change False to True in the Debuggable.

Now back to the cell phone setup. The phone we used is the HTC Desire Z with traditional Chinese display. Follow the steps: □ □ →□ □ □ □ →□ □ →USB□ □ □ . This will set the cell phone into the debug mode as the USB wire is connected (Fig. 4 left). After complete the setup in cell phone, we come back to *Eclipse* and select *Run As* with the left button of mouse (Fig.5 middle) and select Android Application (Fig. 5 right) with right button of mouse, then we have done the setup.

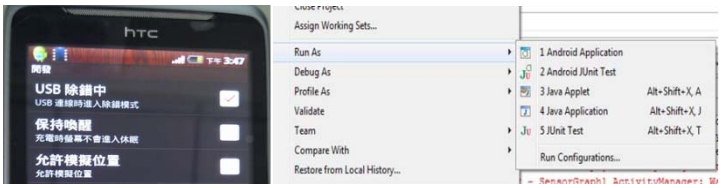

Fig. 4 Cell phone set to Debug mode (left). Open the USB debug mode (middle) and run the Android simulator (right).

The installation is much easier by using a packed APK program that can be loaded by using a HTC installation APP kit (□ □ □ □ □ □ □) (Fig. 5 left). This kit also allows us to revise the bluetooth series number much easier (Fig. 5 middle). But we have to first revise the three command lines of the Arduino program (Fig.5 right) by changing the o, p and q to r, g and b so as to send the correct message to the Arduino.

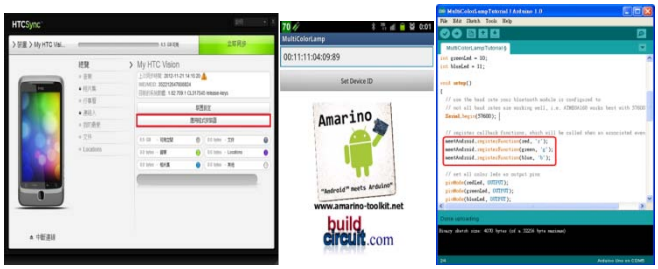

Fig.5 In HTC phone select the installation kits (left), type in the series number of the Bluetooth (middle) and revise the Arduino program (right).

Before we can use Amarino for the Bluetooth connection of the cell phone to the Arduino, we need to first use HTC installation kits to install two APK, Amarino and Amarino Plug-in Bundle, and also install the Amarinolibrary and MeetAndroid into the Amarino libraries.

In the sequel, we will then show our applications of using Android phone to control the lighting and music at the same time. We set the "Red" color to control the *volume* of the music, "Green" for *reverb* and "Blue" for *tremolo* on the touch screen bars in Fig.3 (last picture).

3 Arduino + Max/Msp

Using the SimpleMessage program we can connect the Arduino to Max/Msp and let Arduino to send messages to Max/Msp or vice versa.. There are six analog I/O ports to allow voltage values measured to convert to numerical values and sent to Max/Msp. The Arduino's SimpleMessage program contains three parts: Loop, Readpin and Writepin; Loop is used to build up Readpin and Writepin function, Readpin function is to read analog data and display in Max/Msp, Writepin functions as data sent by Max/Msp to control the Arduino's analog output. This SimpleMessage code can be downloaded from Arduino official Website [5].

The upper left part of Fig.6 in the next page shows the complete patch in Max/Msp environment to describe the interface between Max/Msp and Arduino. The Serial object in the middle decides the USB port, Baud rate (9600). While Max2Asciimessage object allows Max/Msp transmit messages to Arduino, conversely Asciimessage2Max let Arduino transmit message to Max/Msp. The Loadbang object drives Serial object to read one message per second.

4 Android+Arduino+Max/Msp

A complete Max/Msp patch is designed in Fig. 6 where the sound effects *volume, reverb* and *tremolo* of played music are controlled by the R,G,B bars in the cell phone screen. The Bluetooth module on the Arduino receives messages from the cell phone and relays it to the Max/Msp through USB. The three photo resistors sense the light intensity so as to convert them to the voltage changes by the voltage divider circuit appears in the input pins of the Arduino analog input ports.

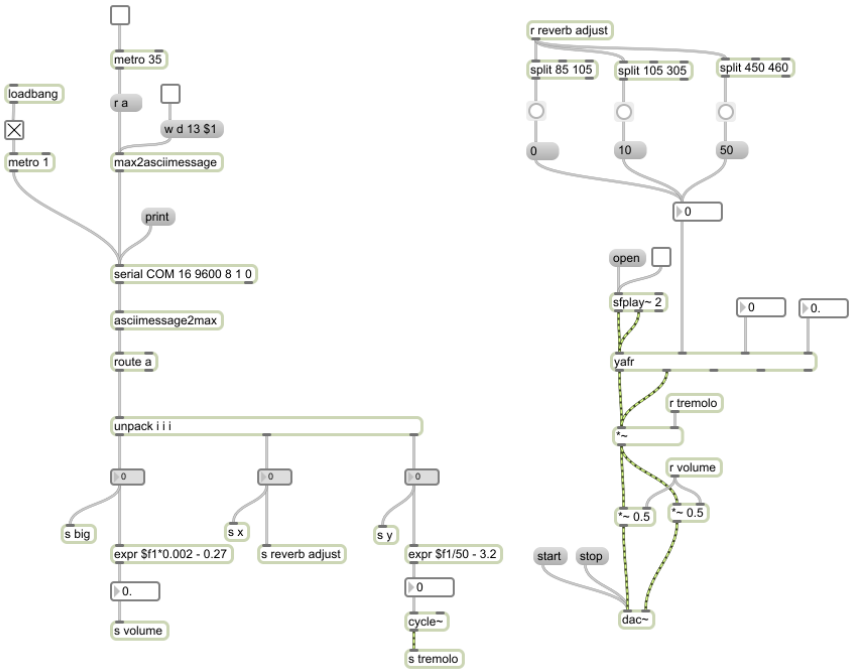

Fig.6 Max/Msp patch for controlling the sound effect of music by phone.

There remain questions on how to activate graphic image from variation of *volume, reverb* and *tremolo*? We proceed to the next section for the task of linking Max/Msp to Processing.

5 Max/MSP+Processing

Processing is a free programming language running on Java environment for interactive graphic image and animation. Since it is not built under Max, we need to find tools to link Processing to Max/Msp. Currently we have two ways to complete the task: Maxlink and oscP5. The Java environment makes the connectivity possible in the bottom layer. For limited pages, we will only explain oscP5 in our demonstration.

We need two plug-in: CNMAT for Max/Msp and oscP5 library for Processing. Under Max/Msp environment hit Options\File Preferences and add in CNMAT then we are able to use oscP5. Next is to install the oscP5 library to Mydocument \Processing \libraries.

OpenSoundControl patch in Max/Msp is the main body of connecting Max/Msp to Processing. We can use Max/Msp to control Processing and vice versa; *updreceive* is for receiving message and *udpsend* for sending message.

Fig. 7 shows sending data from Max/Msp to Processing. As described earlier the OpenSoundControl is used for interface between Max/Msp and Processing, *udp-send* for sending data to Processing, the argument 127.0.1 12000 defined the location of the graph in the Processing. Fig.8 shows the Processing code.

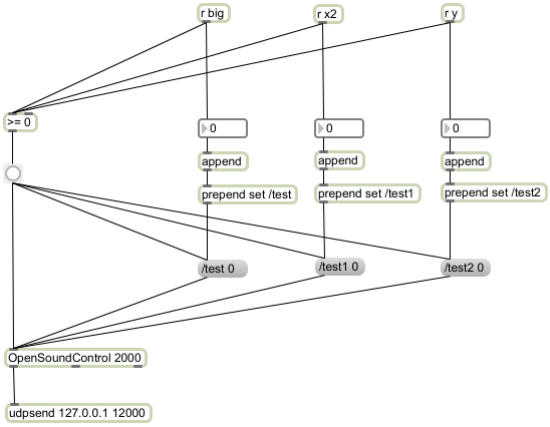

Fig. 7 The patch for Max/Msp sending data to Processing.

```
void setup() {
  size(640, 360, P3D);
  noStroke();
  colorMode(RGB, 1);
  oscP5 = new OscP5(this,12000);
  myRemoteLocation = new NetAddress("127.0.0.1",12000);

  oscP5.plug(this,"test","/test");
  oscP5.plug(this,"test1","/test1");
  oscP5.plug(this,"test2","/test2");
}
```

```
public void test(int value){
  a = + value;
  println(a);
}
public void test1(int value){
  b = + value;
  println(b);
}
public void test2(int value){
  c = + value;
  println(c);
}
```

Fig.8 Graphic image designed in Processing.

In Fig. 8, New NetAddress()is the most important command code with two arguments, the first argument defines the IP address, and the second argument is the port number. oscP5.plug() receives data from Max/Msp. "/test" is to match to the "/test" in Max/Msp patch. So does for for the /test1 and /test2. "test" is to deliver the received message to the public *void test()* of Processing so the message is printed in the Processing window. When the sending data is changed the parameters of the graphic image is changed in the Processing window. This is how Max/Msp is to control the graphic image.

We will now ready to integrate Android phone, Arduino, Max/Msp and Processing. We use finger to control the MultiColorLamp (three color bars) to create LED lighting connected to the Arduino. The light intensity is then detected by photo resistor and converted to voltage values sent to Max/Msp. Three sound effects, *volume, reverb* and *tremolo* can be controlled by fingering the Android phone screen, and in the same time these values also control the graphic images in Processing. Fig.9 shows that the *volume* data change directs to the size change of a square in Processing; the *reverb* data change directs to the rotation of a single cube in the x-axis and the *tremolo* data change directs to the rotation in the y-axis.

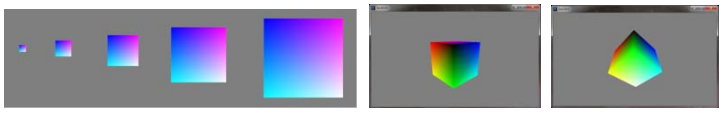

Fig.9 The three sound effects, *volume, reverb* and *tremolo* in Max/Msp control the graphical images, *volume* controls the size of a square-chain (left) and *reverb, tremolo* controls the rotation in the x-axis and y-axis separately.

6 Concluding Remark

We have successfully completed an interactive audiovisual system that utilizes Android phone, Arduino, Max/Msp and Processing for installation art performance. As controlled by the HTC phone's faceplate bars we are able to control audio effects in terms of *volume, reverb* and *tremolo* of the played music. These three terms are linked to change the image patterns shown in Processing window simultaneously. Our preliminary work here can be served as a basis to develop a more attractive (complex) performance scenario in the future.

References

1. Processing Software, http://processing.org/download/.
2. Amarino, Android Meets Arduino, http://www.amarino-toolkit.net/index.php
3. oscP5,http://www.cs.princeton.edu/~prc/ChucKU/Code/OSCAndMIDIExamples/VideoAction /oscP5/documentation/index.htm
4. Hsieh, Y.S., Liu, T.C., & Chen, Y.Y. (2011). The control of lighting and computer music using An Android phone. Hua Kang of Engineering, 27, 154-161.
5. Arduino Official Website, http://www.arduino.cc/.
6. Introduction of Max/Msp, http://www.maxmspjitter.com/index.html.
7. Liu, T.C., Chang, S.H., & Hsiao, C.Y. (2011). A modified quad-Theremin for interactive computer music control. ICMT2011, 6179-6182.
8. Eclipse system introduction, http://www.eclipse.org/
9. Kaufmann, B., & Buechley, L.(2010). A toolkit for the rapid prototyping of mobile ubiquitous Computing. Cambridge, USA, 2010.). (http://www.amarino-toolkit.net/tl_files/thesis/amarino_thesis_kaufmann_2010.pdf).