# LUMI: Live Performance Paradigms Utilizing Software Integrated Touch Screen and Pressure Sensitive Button Matrix

**Mingfei Gao**
CCRMA / Stanford University
Stanford, CA 94305
mikegao@ccrma.stanford.edu

**Craig Hanson**
CCRMA / Stanford University
Stanford, CA 94305
chanson9@ccrma.stanford.edu

## Abstract

This paper explores a rapidly developed, new musical interface involving a touch-screen, 32 pressure sensitive button pads, infrared sensor, 8 knobs and cross-fader. We provide a versatile platform for computer-based music performance and production using a human computer interface that has strong visual and tactile feedback as well as robust software that exploits the strengths of each individual system component.

**Keywords**: LUMI, live performance interface, pressure

## 1. Introduction

The LUMI is a new hardware/software performance interface optimized for live and studio musical expression (see Figure 1). The LUMI provides the performer with multi-dimensional control via a 10.4" touch screen, 32 pressure sensitive buttons, infrared sensor, 8 knobs and cross-fader.

## 2. Prior Works

After seeing a seminar by Adrian Freed at Stanford University, LUMI decided to adopt his method of achieving pressure sensitivity, as published in a NIME 2008 paper [1]. Arduinome buttons, originally designed to be binary on / off switches, are modified by placing resistive fabric underneath the buttons to detect pressure. The actual circuit and firmware code modifications to obtain pressure data on the LUMI's thirty-two buttons, however, were realized on our own, as Freed's code and circuitry were not released to the public. It was possible to rapidly prototype the interface in the short span of a month, by modifying Arduinome's serial to OSC interface and utilizing Adrian Freed's method of deriving pressure sensitivity. As a result, more attention could be given to the actual software that integrates the touch-screen with the buttons.



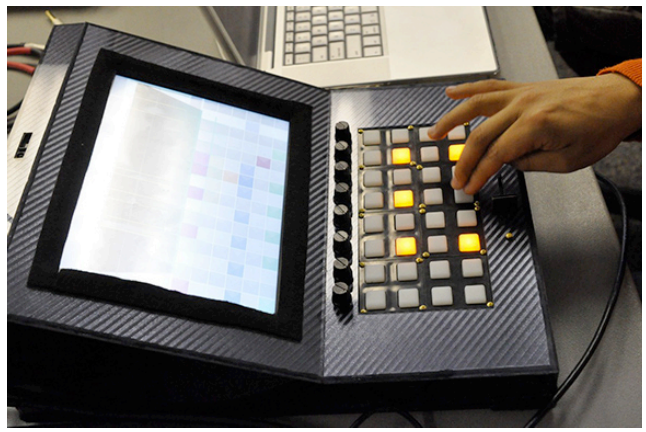

**Figure 1. LUMI Picture by Ben Keyserling**

## 3. Design

The design consists of a carefully laid out hardware interface brought to life by custom software that makes use of the hardware's best assets.

### 3.1 Design Configuration

Various hardware and software configurations were taken into account while prototyping the LUMI interface. The current hardware layout was designed to allow the user to physically manipulate controls while maintaining a clear view of the screen. The infrared proximity sensor is centered above the touch-screen to allow the performer to reach into the infrared path with either hand. The touch-screen is placed at a 25-degree angle to provide an optimal view of the screen. The remaining hardware elements (knobs, button matrix and cross-fader) are laid out to allow for a minimal movement to reach each user control.

### 3.2 Hardware Specification

The LUMI hardware design consists of a 10.4" touch screen, 32 pressure-sensitive buttons (providing velocity and polyphonic after-touch), a short-range infrared proximity sensor, 8 knobs, and a cross-fader. An Arduino with an Atmel micro-controller is used to interpret control data that is then sent to an external computer. This hardware profile integrates a variety of expressive controls that allow for manipulation of many musical parameters.

### 3.3 Software Specification

Our software is written in C/C++, utilizing libraries such as: RtAudio, OpenGL, libsndfile, OSCPack and FFTW. Our current software has three main sections: a general-purpose virtual control surface, a touch-screen sampler/waveform manipulator and an intelligent melodic instrument.

The virtual control surface consists of XY pads, buttons, knobs and sliders. These virtual control objects can be used to construct customized user interfaces tailored to the needs of specific applications. These objects are able to send MIDI or OSC messages to any live DAW (such as Ableton Live) or DJ software (such as Serato Scratchlive).

The second section of our software package is a sampler entitled "Wave Cut," which allows the user to manipulate sound buffers in real-time. Sound buffers may be obtained through microphone input or by dragging and dropping audio files directly into the program. The user is then able to physically "chop" a waveform into subsections using the touch-screen. This is accomplished by touching the screen, dragging to the position that one wish to place a cut, and releasing. This was found to be more accurate than placing the cut where the initial touch occurs. The closest zero-crossing is detected for the location of the cut. These subsections can then be assigned to the 32 buttons for immediate playback. The user can drag and drop subsections onto a 16x6 step sequencer, allowing users to quickly and intuitively take snippets from recordings and step sequence them, synchronized to a DAW via MIDI or OSC.

Finally, a melodic instrument allows the user to map and remap the button matrix to tonal scales. The user can change keys and key qualities (such as modes) on the fly. Networked communication via OSC facilitates a collaborative musical environment in which notes played by one user can be transmitted through a network connection to another user. Notes are displayed on the receiving user's button matrix via lighting the corresponding note within the current scale mapping, even if the two users' mappings are different.

## 4. Implementation

### 4.1 Hardware/Software Implementation

The LUMI firmware is built upon a low level C framework running on an Arduino. The C code is responsible for sending serial control data from the hardware to an external machine running LUMI software. This software is built in C/C++ with libraries that maximize platform compatibility. This was done so that in the future, the software can be ported and internalized on an onboard computer, rendering the current LUMI controller into its own stand-alone instrument.

### 4.2 User Feedback

Tactile and haptic feedback in music instruments have been shown to improve performance [2]. For this reason, the LUMI relies on physical sensors for important user interactions rather than only virtual touch-screen controls. The buttons, knobs and faders provide tangible surfaces for interacting with sound.

The touch-screen provides visual feedback to the user. It can be useful when controlling button mappings, tonal parameters and synthesis parameters. Textual labels are displayed on the touch-screen for explicit communication of data to the user. Other information can be communicated with color, such as in the virtual XY pads, upon which a gradient represents the current location of the pad. With both LEDs in each button and a touch-screen, many visual cues and symbolic representations can be displayed to the user to show what he or she is controlling.

### 4.3 Compatibility

LUMI software uses the MIDI and OSC protocols for communication with other applications. MIDI time synchronization is available with the LUMI acting as both slave and master. OSC is also supported for communication with applications such as Pd, Max/MSP, Reaktor, and ChucK.

## 5. Final Words

The LUMI is a rapidly prototyped device designed as the authors' first attempt at an HCI project, for a class at Stanford CCRMA. Many other software and hardware features are partially implemented, but cannot be discussed due to space limitations.

## 6. Acknowledgments

The authors gratefully acknowledge the following people: Michael Zeligs, a student in Stanford University's class on "Physical Interaction Design for Music" for his partnership and assistance during the initial LUMI conception and prototype; Adrian Freed of the University of California, Berkeley (CNMAT); everyone at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University.

## References

[1] A. Freed. "Application of New Fiber and Malleable Materials for Agile Development of Augmented Instruments and Controllers" in *Proc. of the Conf. on New Instruments for Musical Expression (NIME)*, 2008.

[2] B. Gillespie, "Introduction to Haptics," in Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics, P. R. Cook, Ed., pp. 229–245, MIT Press, London, UK, 1999.