COSMOS UC San Diego
California State Summer School for Mathematics and Science

Light Jam$

By Jeremy Mui, Ramon Magana, Daniel Wehara

Source Code

Source code for the Arduino can be found here

Introduction

MIDI instruments do not create their own sound; they merely instruct a synthesizer to create a sound. Since MIDI keyboards can be very expensive, other ways of MIDI input were explored. One method is to program an Arduino microcontroller to read photoresistor inputs. Both Arduinos and photoresistors are very affordable and can be easily repurposed for other projects, making music creation more accessible to consumers.

Hardware

Arduino Micro
Controller for the instrument

2 16-Channel Analog Multiplexers
Changes one analog pin into 16 different pins

3 Breadboards
Connects power, buttons, and potentiometers

25 Photoresistors
Detects light intensity

27 330 Ohm Resistors
Needed for the photoresistors and multiplexers to work

Wiring
Connects everything

Heat Shrink Tubing
Reinforces wire connections

Acrylic
Makes up the body of the guitar

Laser Cutter
Cuts the Acrylic

Software

Arduino IDE
Used to program the Arduino

Hairless MIDI
Routes MIDI from Arduino to Pure Data

loopMIDI
Routes MIDI from Arduino to Pure Data

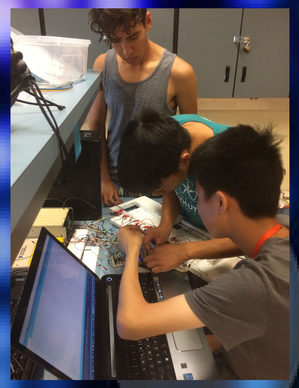
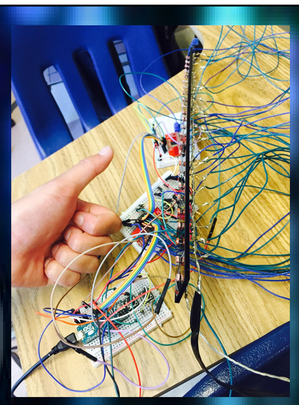
PureData
Used to program a synthesizer

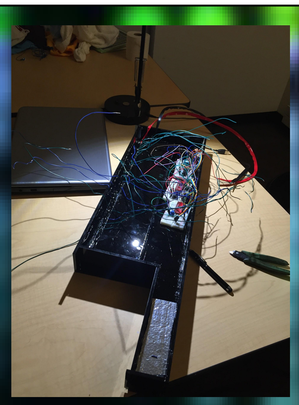
Abstract

This Light Keytar is a MIDI controller that can be played without touch. With a digital audio workstation, it can be used to play any software instrument. The Light Keytar has a layout similar to the keytar, but notes are played by obstructing light rather than pressing mechanical keys. The body is made from light but elegant, black laser-cut acrylic. To construct this instrument, 25 photoresistors were lined up on a strip of acrylic, creating two octaves of “keys.” The amount of voltage flowing through the photoresistors change based on the amount of light they are exposed to, so by reading the voltage, the relative amount of light can be determined. The Arduino does not have enough analog inputs to read from all 25 photoresistors, so multiplexers were used. Multiplexers serve as an avenue for multiple photoresistors to be read by only one analog input pin. The Arduino is programmed to send respective MIDI values whenever any sensor’s values change from a decrease in light. Hairless MIDI is used to route MIDI from the Arduino through a microUSB cable to loopMIDI in order for the keytar to be recognized as a MIDI device by the computer programs Pure Data. MIDI signals with note number and velocity are sent to a Pure Data synthesizer, which takes the MIDI values and converts them into audio. The timbre of the audio is modified by an ADSR envelope and a filter. With speakers, the Light Keytar can be played at performance volume with ease.

Extensions

The prototype can be improved by adding a Raspberry Pi in order to run the synthesizer, making the product completely mobile. Additionally, the addition of an analog ¼ inch output would make the instrument performance ready. Furthermore, each photoresistor can be reprogrammed to output chords for easy harmonization, or even algorithmically harmonize with the user.



Acknowledgements: We would like to thank Maurício de Oliveira, Kim Morris, and Kevin Haywood for their guidance in wiring, and Shlomo Dubnov for inspiring us to use technology in new ways to create sound. We extend our undying gratitude to Colin Zyskowski for providing us with code for the multiplexers and laser cutting the body for our instrument. We could not have completed this project without all of the aforementioned people, and we are truly grateful for their assistance!

Procedures

1. Wiring the photoresistors to the multiplexers.
2. Wiring the multiplexers, buttons, and potentiometers to the Arduino.
3. Writing code for detecting input from photoresistors, buttons, and potentiometers.
4. Writing code for sending MIDI messages from the Arduino.
5. Programming a synthesizer in Pure Data.
6. Configuring loopMIDI to create a loopback MIDI device.
7. Configuring Hairless MIDI for connecting the Arduino to the loopMIDI port.
8. Configuring PureData for receiving input from the loopMIDI port.

Product Features

The Light Keytar features a sleek, glossy back acrylic body. It has 25 photoresistors that serve as “keys.” Octave changes, recording, and playback can be achieved through buttons. Octave changes allow the Light Keytar to have a range from MIDI note 12 (C0) to MIDI note 120 (C9), which is greater than that of a piano. Dynamics can be inputted through a potentiometer that serves as a dial for MIDI notes’ velocities.

Implications

With the ability to program an Arduino to read photoresistor voltages, detect changes in their values, and generate MIDI output, musicians can affordably create their own MIDI instrument and program it to their liking. By making music creation tools more accessible to all using open-source, programmable hardware such as the Arduino, more people are able to express themselves through music without the need for expensive instruments or hardware.