Keynetic

D2: Sun A Cho, Katherine Dettmer, and Lance Yarlott
18-500 Capstone Design, Spring 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

A mechanically actuated keyboard, under the management of a microcontroller and OpenCV, offers an equitable solution for music playing. Primarily designed for simple notes and chords, as anticipated, this keyboard boasts a playing range of 14 keys. However, its capabilities can be further extended through MIDI or additional hardware expansions. Moreover, this innovative keyboard can store up to 8 measures of generated music. It allows users to improvise and create music in real-time. In essence, this mechanically actuated keyboard addresses the need for hands-free music options for people who may not have the traditional physical ability to create music.

System Architecture

Figure 1: Block Diagram of the Integrated System

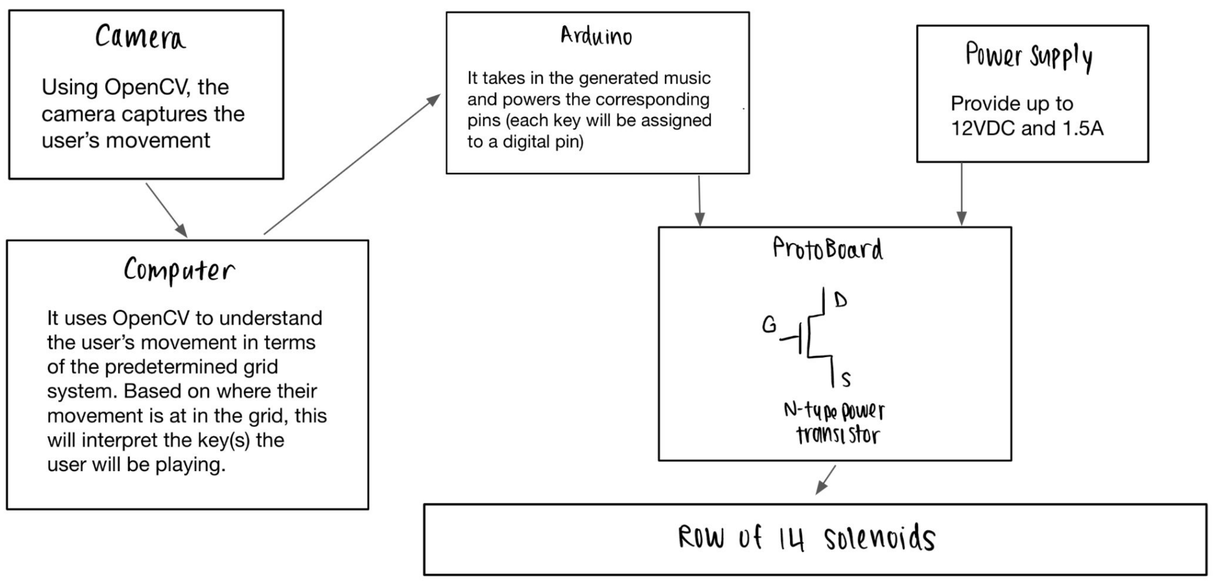
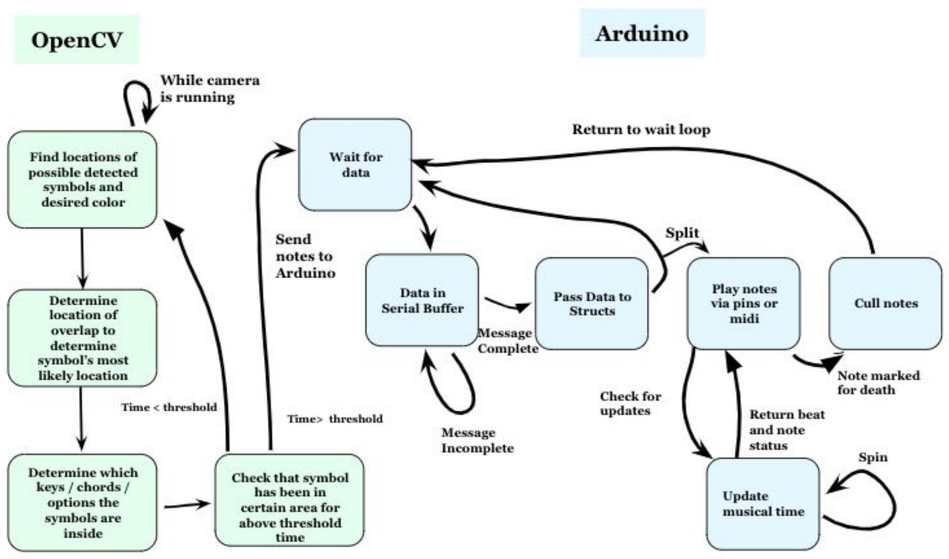
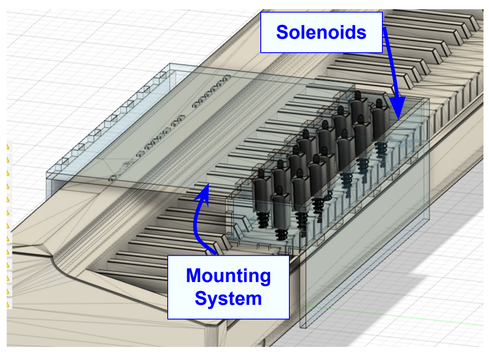


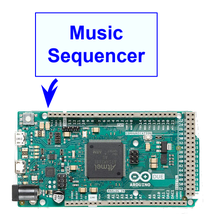
Figure 2: Block Diagram Describing Software Logic Flow

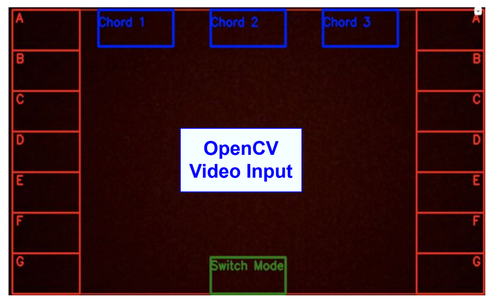


System Description

Computer Vision Software: Use custom Haar Cascade and color detection to determine location of symbols and what keys, chords, or options the user is signaling.
Music Generation Software: Take in input from CV in numeric or string format, then generate melodies and phrases from that with 8th note subdivisions.
Arduino: Reads in byte-data over serial. It can accept a multitude of message formats. Includes error handling. Has a music sequencer for solenoid activation.
Actuators: 14 solenoids are contained inside a box that sits above the keyboard and controlled by Arduino's digital pins. Maximum of 4 solenoids will be turned on at each time. The actuators will use its push function to actually play the piano.







System Evaluation

OpenCV SW: Latency was tested by time taken to detect symbol. Accuracy testing was done to determine how often the symbol was correctly detected.
Music SW: Musical generation was tested by generating phrases and subjectively analyzing their content.
Serial SW: Testing was done by sending single byte streams and having the Arduino return the same bytes. Pin testing was done with a series of LEDs mimicking the solenoids.
HW: from testing our system as part of the integration process, we noticed that solenoids were "aging out" and showing increased latency. Just to be safe, we added MIDI functionality to our system in case a solenoid loses functionality.

Figure 3: Testing Metrics and the Measured Results

Description	Goal	Measured
Hand Recognition Accuracy (using color)	> 90%	92%
Latency (from when a hand is placed in the box to when software recognizes)	< 1 second	0.6 seconds
Latency (from SW to when key is played)	< 1 second	0.3 seconds
Power to run the actuator system	Less than 30 V / 3 A	20V / 2A

Conclusions & Additional Information

Solenoids are known to be sensitive and prone to aging (shown as latency) with prolonged use. In the context of a musical instrument, their reliability becomes crucial. Our system aims to detect movement and position, with each representing a key or a chord. The system relies on serial interfacing between Python and Arduino, enabling software-hardware communication. This method produces latency due to the handling of single bytes, it offers a viable and simple solution for capturing and interpreting user input. Nevertheless, with careful design and implementation, this computer vision system has shown itself to be a valid proof of concept for intuitive, dexterity-independent musical performance.

http://course.ece.cmu.edu/~ece500/projects/s23-teamd2/

Electrical & Computer ENGINEERING
Carnegie Mellon