Group 2017.011

Modular Musical Instrument

Vincent Chung | Gitesh Dhir | Akshay Joshi | Hussein Salemwalla | Justin Wolfe

Consultant – Prof. David Wang

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Motivation

- Standard instruments can be overwhelming for a beginner musician, preventing them from being involved in musical sessions with peers
- There needs to be a way for a user to play a subset of an instrument with peers

Objective

- Design an instrument which can be customized for the user's musical ability
- Allow multiple instruments to be connected for beginner musicians to be involved with their peers and create music

Design Alternatives

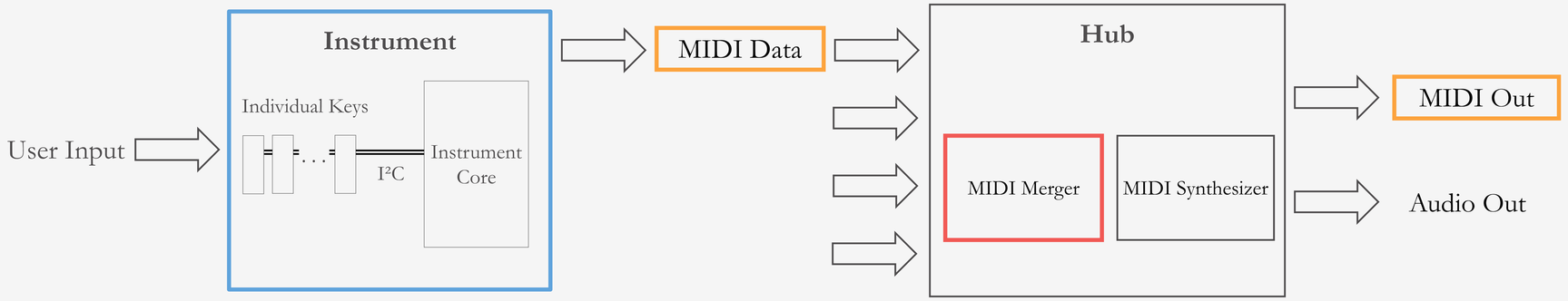
Keys
- Touch-sensitive screens which the user can tap on
- Virtual/augmented reality to present an instrument onto any surface

Music Data
- Produce an analog signal from each individual control

Data Bus
- Serial Peripheral Interface (SPI)
- Single-wire with arbitration
- Wireless instruments

Advantages

- Support for different type of controls: there is a uniform format of data packets sent from the control boards to the core, allowing easy integration of multiple types of controls
- Physical interaction: the feedback from physical keys is more natural than haptic feedback provided by a touch screen panel or the lack of feedback from a VR/AR alternative
- Variable number of controls: up to 4 cores supporting different instruments allows the user to easily access the desired notes on the instruments
- Portable: having modular keys makes the system portable and easy to carry around



Instrument Protocol

All connected keys must have a unique address for the instrument core to convert the key press into MIDI data.

A power latching circuit on each key prevents the next key from receiving power until the current one has received an address.

Data from each key is sent with the key's address, allowing the core to convert the data into the correct musical note.

Figure 1 illustrates the latency from a key press to the hub producing an audible note.

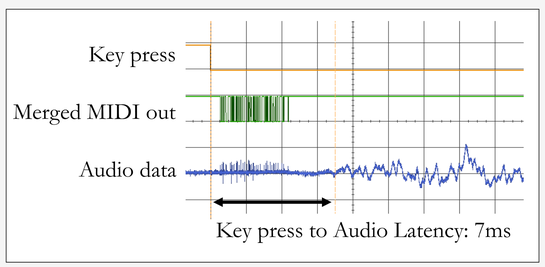


Figure 1: Key press latency to audio out

What is MIDI?

MIDI is an industry standard protocol for musical data. [1]

Musical "messages" are comprised of three bytes, a command (or status) byte, followed by up to two data bytes. [2]

Command bytes indicate actions (e.g. turning a note on), while data bytes containing information about the action (e.g. velocity of the key press). Figure 2 shows a sample MIDI message sent from an instrument.

A synthesizer can read this data and convert into sound.

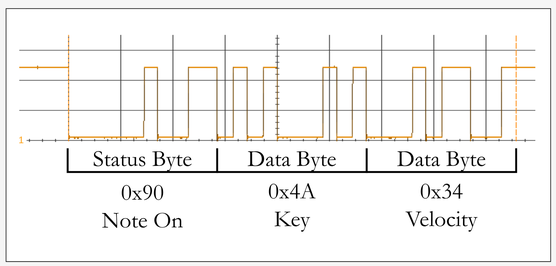


Figure 2: Sample MIDI note on message

MIDI Merger

To combine MIDI data, bytes from various inputs must be sequenced one after another.

The MIDI merger buffers data from each instrument, while simultaneously iterating through each input's queue and transmitting any information present.

The merged output can be sent to a MIDI synthesizer to generate audible sound.

Figure 3 shows two notes sent into separate inputs on the MIDI merger and the resulting merged output.

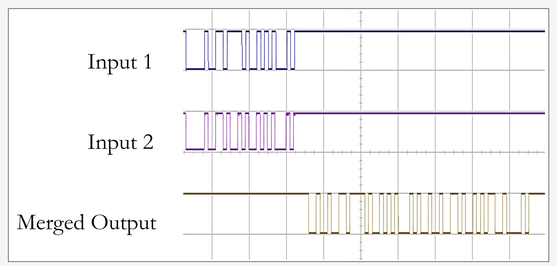


Figure 3: Multiple MIDI inputs merged into a single output

References

[1] Sapp, Craig Stuart. "Essentials Of The MIDI Protocol". *Ccrma.stanford.edu*. N.p., 2017. Web. 12 Mar. 2017.

[2] "MIDI-protocol - 03-MIDI Message", *Midi-protocol.wikispaces.com*, 2017. [Online]. Available: https://midi-protocol.wikispaces.com/03-MIDI+Message. [Accessed: 13- Mar- 2017].

Acknowledgements

Our group would like to thank the following people for their help with this project: David Bell, Chaitanya Dogra, Lutfia Kathrada, Kevin Michener, the course coordinator Prof. Dan Davison, and our consultant Prof. David Wang.