MIDI Sequencer

Jacob Adlers (960331-2472)
Gustaf Pihl (861223-7555)

December 9, 2016

1 Objective and requirements

We turned the UNO32 Chipkit into a MIDI sequencer where we can use a MIDI keyboard to create sequences of MIDI data. The chipkit outputs musical data in MIDI form so that an external device can parse the output into sound. A sequencer is a electronic device for storing sequences of musical notes and rhythms. We think our project qualifies as an advanced project.

Features of our sequencer:

- Record polyphonic sequences from MIDI input. The sequence stored is 64 beats long.
- Output the stored sequence via MIDI
- Clear the stored sequence
- Undo function
- Play/pause
- Toggle record enable/disable
- Adjustable tempo
- Transpose function
- The LCD shows information about:
  - Saves (possible undos)
  - Current tempo
  - If the sequencer is paused or playing
  - If the sequencer is recording or not

2 Solution

The sequencer is implemented on the ChipKIT Uno 32 board along with the Basic I/O shield. In order to send and receive MIDI data we purchased a few electronic components[1]. We use the buttons and switches on the I/O shield to implement the user interface with the sequencer. A matrix (2D-array) is used to store input MIDI data on the ChipKIT, where the columns represent time intervals and rows represent stored MIDI events (note on or note off). A received MIDI event triggers an interrupt which, if the record switch is up, stores the event in the matrix. For playback a timer loops through the columns of the matrix and sends the MIDI data stored in each column to a MIDI connector which, for example, can be connected to a synth/piano. We use UART for both MIDI input and output.

[1] One optocoupler, one breadboard, a few resistors, two MIDI connectors, one diode and a bunch of cables.

3 Validation

Each time the code developed with a new feature or was refactored we ran it on the ChipKIT to test and verify that everything worked as intended. Most times when we developed some new feature we added printouts to the display to verify key parts of the code for that feature. We used a KORG R3 synthesizer with MIDI in and out in order to test recording and playing back sequences.

4 Contributions

We sat and worked together throughout the whole project and as a result we both contributed equally. At a typical session we would rotate who was doing the actual typing. As a result, roughly 50% of the code is written by Jacob and 50% by Gustaf.

5 Reflections

We are satisfied with the result of our project and that we managed to do pretty much everything we hoped to do while planning. The sequencer feels like a useful tool for creating music and has been a lot of fun to use. Some of the features we implemented were a lot harder to implement than we first thought, while others turned out to be easier.

A few of the optional goals we specified in the abstract draft were either not useful functions, or not feasible to be implemented on our limited user interface, and so were omitted.