

15-323/623 Spring 2019

Homework 1

Due Jan 22

This homework is based on the reading (MIDI specification). Copy the questions, add clear short answers, and submit your work as a .pdf file to Autolab.

- 1.) Write your name and your Andrew ID.
- 2.) Explain the difference between Channel Aftertouch (aka Channel Pressure) and Polyphonic Aftertouch (aka Key Pressure) messages in terms of semantics (what they mean) as opposed to syntax (what are the bits).
- 3.) Describe the format of a MIDI controller change message (including the status byte). Account for every bit.
- 4.) Suppose you want to play different instruments at once using a single MIDI connection to a synthesizer. For example, you want to select piano, violin, viola, cello, bass, flute, clarinet, oboe, bassoon, etc. How many different instruments can you play at once? (This is not a trick question, if you include drums, count “drums” of all types to be one instrument.)
- 5.) How long does it take to send one 3-byte MIDI message? How long does it take to play a 5-note chord in each of the following two cases? (Show the math and label the units, e.g. 7 bananas/second * 0.1 monkeys/banana * 2 minutes * 60 seconds/minute = 84 monkeys.)
 - 5A.) Assume each note is encoded using a 3-byte NoteOn message. What is the transmission time?
 - 5B.) Assume the first note is encoded using a 3-byte NoteOn message, and the remaining notes are encoded using “running status.” What is the transmission time?

Note: How can we evaluate these timing errors introduced by speed limitations? We are sensitive to timing errors of about 10ms in a sequence of steady pulses; we perceive notes as simultaneous chords when they occur within about 50ms of one another; we can hear that onset delays are different when the delays change by 1ms or even less, depending on the sounds (even if we cannot detect which onset was first); sound travels about 1 foot/ms, so musicians separated by normal distances have difficulty synchronizing more closely than 10ms; typical tapping of beats by hand has a standard deviation of around 30ms of timing error; the MIDI standard has not changed even though much faster hardware is now very inexpensive (a marketing argument that MIDI is “fast enough”). 30ms of timing error; the MIDI standard has not changed even though much faster hardware is now very inexpensive (a marketing argument that MIDI is “fast enough”).

Homework 1, due Jan 22

6.) Human muscles are limited to about 100Hz, meaning that sampling positions at 200Hz should be enough to capture human gestures. Let's be generous and capture 300 data points per second from sensors. (Let's think about this in a little more detail: Can humans really be that slow? If the number were higher, we'd be able to make sound by waving our arms really fast, so yes, we're really that slow. But is a 300Hz sample rate really fast enough to capture all gestures? Tap on something hard with a fingernail. You should definitely hear audio, at least in part because the deceleration happens at audio rates. So if you glued an accelerometer onto your fingernail and tapped on things, sampling at 300Hz would *not* capture everything. We'll ignore this and stick with the 300Hz number.)

6A.) Suppose we build a microcontroller to capture pressure from a sensor and send the pressure as MIDI to our laptop using Channel Aftertouch messages. How many messages can we send per second?

6B.) How many bits of data can we send per message? I.e. how many bits can we use to encode pressure samples?

6C.) Suppose we now want to capture multiple pressure sensors. We need to sample each sensor 300 times per second. Assuming the only limitation is MIDI bandwidth, and we limit ourselves to one standard MIDI cable connection, how many pressure sensors can we have, with each sensor sending 300 Channel Aftertouch messages per second on the same MIDI cable?

6.) Write 8 bytes in hexadecimal that will

- Select an oboe sound using General MIDI on Channel 3 (where channels are numbered 1 to 16)
- Play an F# above middle C with a velocity of 110
- Stop playing the F# above middle C (using a NoteOn status byte, not NoteOff)
- You could encode this sequence in only 7 bytes. How?