# The 2005 Region VI AGO Convention
# MIDI Workshop – MIDI 102
# An Introduction to Concepts and Terminology

### June 27, 2005 – 4:30 PM to 5:30 PM

### Sponsored by
### Rodgers Instruments and Church Organs of Colorado

## MIDI

- MI – Musical Instrument
  - Acoustic or electronic
  - Organ, piano, keyboard
  - Sound module, effects processor, sequencer
- DI – Digital Interface
  - Serial data transmitted from one device to another
  - Similar to data transmitted by a computer to a telephone modem
  - Not the audio or analog information that we hear

## MIDI DATA

- Messages (also called commands) that communicate events
- Messages or packets of data are transmitted as events occur
- If no events occur, no data is transmitted
- Examples of events
  - Note on
  - Note off
  - Program change
  - Expression change

## MIDI DEVICES OF MOST INTEREST TO ORGANISTS

- MIDI Sequencers - record and playback note, stop change and expression events
- Sound Modules - create sounds in addition to the sounds available on the organ

## THE MIDI CONNECTION BETWEEN MIDI DEVICES

- Typical connectors available on a MIDI equipped musical instrument
  - IN – Accepts data from another MIDI equipped device
  - OUT – Sends data to another MIDI equipped device
  - THRU – Passes data received at the IN connector onto another MIDI equipped device
- Each connector is circular and has 5 pins (called a 5-pin DIN connector)
- All MIDI cables are alike with male connectors on each end
  - One pin is connected to a shield inside the cable
  - Two pins are used for transmitting data
  - Two pins are not connected
- The MIDI cable carries only data - not sound
- Sound is carried on an audio cable - typically a cable with quarter inch phone jacks on each end

## MAKING A SOUND MODULE CONNECTION

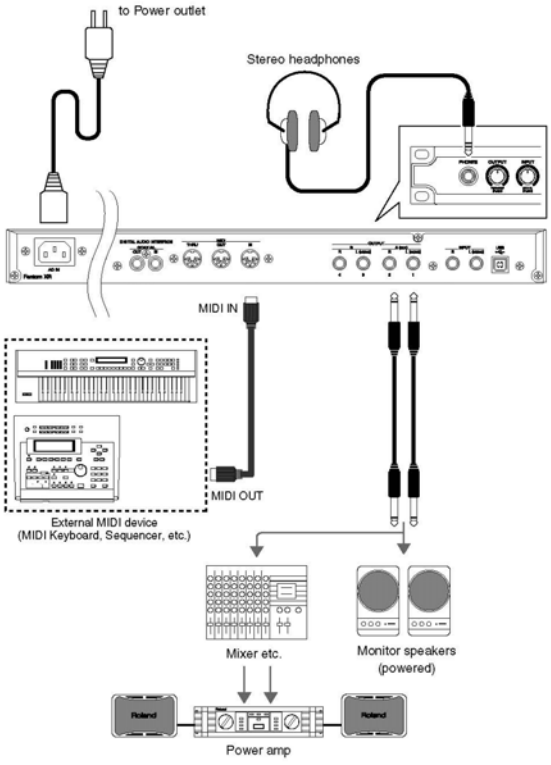
- Shown below is a page from the Roland Fantom FX instruction manual
- The Roland Fantom XR is a sound module
- The MIDI OUT from External MIDI device (MIDI Keyboard, Sequencer, etc.) is connected to MIDI IN on the Fantom XR with a MIDI cable
- This allows MIDI message data to flow from the keyboard or other device to the module
- The L & R (audio) OUTPUT from the Fantom XR are connected to whatever will serve to amplify the sound for speaker output with typically a quarter inch phone jack audio cable

# Getting Ready

## Connecting an Amp and Speaker System

Since the Fantom-XR contains no amplifier or speakers, you'll need to connect it to audio equipment such as a keyboard amplifier, monitor speaker system or home stereo, or use headphones to hear its sound.

1. **Before hooking anything up, make sure that the power on all of your gear is turned OFF.**
2. **Connect one end of the supplied power cable to the Fantom-XR, and the other end to a power outlet.**
3. **Connect the Fantom-XR to your amp/speaker system as shown in the diagram.**




NOTE
To prevent malfunction and/or damage to speakers or other devices, always turn down the volume, and turn off the power on all devices before making any connections.

HINT
In order to fully experience the Fantom-XR's sound, we recommend using a stereo amp/speaker system. If you're using a mono system, however, make your connections to the Fantom-XR's OUTPUT A (MIX) jack L (MONO).

NOTE
Audio cables are not included with the Fantom-XR. You'll need to provide them.

For details on how to install a Wave Expansion Board (sold separately), refer to **"Installing the Wave Expansion Board"** (p. 166).

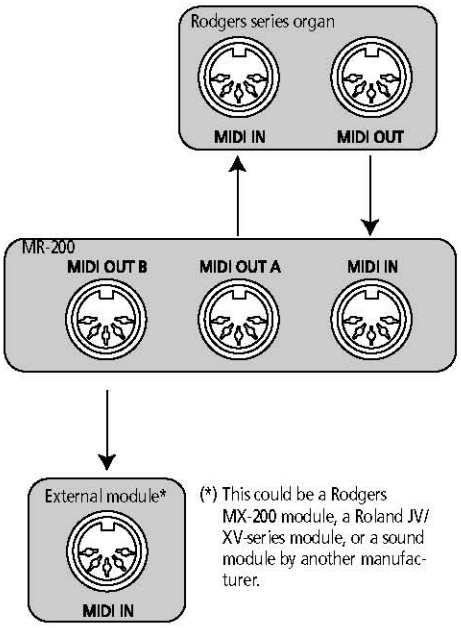
16

## MAKING A MIDI SEQUENCER CONNECTION

- Shown below is a page from Rodgers MR-200 instruction manual
- The Rodgers MR-200 is a MIDI sequencer
- The MIDI OUT from the sequencer is connected to the MIDI IN on the Rodgers organ
- The MIDI OUT from the Rodgers organ is connected to the MIDI IN on the sequencer
- When the organist is recording a series of MIDI events (i.e. playing a piece of music), data flows from the organ to the sequencer
- When someone wants to play back the sequence, data flows from the sequencer to the organ to recreate the events originally recorded

### MIDI connections

In order to use the MR-200's full potential, you need to connect its MIDI sockets as follows:




## THE MIDI DATA

- Serial data protocol which means data is transmitted one *bit* at a time
- A *Message* or *Command* consists of **one or more** *bytes*
- A *byte* consists of 10 *bits*
    - One start bit
    - Eight data bits
    - One stop bit
- A bit is an electrical state. Current-on or current-off (logical 0 or logical 1).
- Data is transmitted at 31,250 bits or 3,125 bytes per second (10 bits per byte)

## DATA NOTATION

- A 10 bit byte can be represented in *Binary* notation as in the following example.
  - 0100101011
- Since the first (start) bit and last (stop) bits are always the same (they frame the byte), it is customary to only refer to the eight variable data bits, i.e. or example again.
  - 10010101
- For ease of use, computer folks think in terms of the hexadecimal numbering system that deals with four bits at a time as follows.
  - 0000 = 0 0100 = 4 1000 = 8 1100 = C
  - 0001 = 1 0101 = 5 1001 = 9 1101 = D
  - 0010 = 2 0110 = 6 1010 = A 1110 = E
  - 0011 = 3 0111 = 7 1011 = B 1111 = F
- In the decimal system A = 10, B = 11, C = 12, D = 13, E = 14, F = 15
- Hence our example data byte above is 10010101 = 95H (the "H" stands for "Hex" as in hexadecimal notation)

## MIDI MESSAGE OR COMMAND STRUCTURE

- Recall from above that a *Message* or *Command* consists of one or more *Bytes*
- The first Byte is the command or status byte
- The bytes following the status byte are data bytes
- Bit 1 of each byte indicates whether the byte is a command or data
  - Bit 1 = 1 indicates byte is command
  - Bit 1 = 0 indicates byte is data
- If the byte is a command, the **first four bits** specify the type of command and the **last four bits** specify the MIDI CHANNEL to which the command applies.

## MIDI CHANNEL

- The MIDI specification provides for a maximum of 16 *MIDI Channels*
- The concept of a MIDI Channel provides the following features.
  - Allows multiple devices to share the same data stream and selectively react to the data. For example, a drum machine on one channel and a keyboard on another driven by a single MIDI data source.
  - Allows a single device capable of producing multiple sounds to respond to desired commands specific for each sound. For example a split keyboard (or in the case of an organ, the Swell and Great) issuing commands to a single sound module for a flute solo accompanied by strings.
  - On a Rodgers Organ, as an example, the Great manual transmits and receives MIDI messages on MIDI Channel 12, the Swell on Channel 13, the Pedal on Channel 14, the Choir on Channel 15 and the Solo on Channel 16
  - Also on a typical three manual Rodgers Organ MIDI Channels 1 through 8 are assigned to the eight available MIDI couplers (two for each manual) as follows: 1 - GT A, 2 - SW A, 3 - PD A, 4 - CH A, 5 - GT B, 6 - SW B, 7 - PD B, 8 - CH B
  - MIDI Channel 10 is generally reserved for percussion (or drum) sets
- Hence to achieve desired results, it is important that the transmitting MIDI device and the receiving MIDI device are creating and responding to data on the same intended MIDI Channel.

## OVERVIEW OF THE MIDI COMMAND SET

Specific MIDI Channel Commands

| Command or Status Name | Byte 1 (Hex Value) | | Number and type of additional data bytes |
|---|---|---|---|
| | Bits 1 – 4 Command | Bits 5 – 8 Channel | |
| **Note Off** | 8 | 0 - F | 2 – note number, velocity value |
| **Note On** | 9 | 0 – F | 2 – note number, velocity value |
| Key Pressure (polyphonic) | A | 0 – F | 2 – note number, pressure value |
| **Control Change** | B | 0 – F | 2 – controller number, controller value |
| **Program Change** | C | 0 – F | 1 – program number |
| Channel Pressure (mono) | D | 0 – F | 1 – pressure value |
| Pitch Bend | E | 0 – F | 2 – (LSB, MSB) |

System Messages (i.e. not associated with a specific MIDI Channel)

| Command Name | Byte 1 (Hex Value) | Number and type of additional data bytes |
|---|---|---|
| **System Exclusive** | F0 | Variable as defined by manufacturer<br>Data byte 1 = manufacturer code<br>Additional data bytes as defined |
| System Common | | |
| . MIDI Time Code | F1 | 1 – time code, nibble |
| . Song Position Pointer | F2 | 2 – (MSB, LSB) |
| . Song Select | F3 | 1 – song number |
| . Tune Request | F6 | 0 |
| **System Exclusive End** | F7 | 0 |
| System Real Time | | |
| . Timing Clock | F8 | 0 |
| . **Start** | FA | 0 |
| . Continue | FB | 0 |
| . **Stop** | FC | 0 |
| . Active Sensing | FE | 0 |
| . **System Reset** | FF | 0 |

Controllers (Control Change Commands)

- Total of 128 possible controllers per MIDI channel (00H thru 7FH)
    - First 64 (00H - 3FH) are double-precision controllers (i.e. controller commands always sent in pairs)
    - Next 64 are single-precision controllers
- Examples of double-precision controllers
    - **Bank Select** – discussed later
    - Modulation – device is a wheel or lever, affects LFO depth
    - Foot Pedal – device is foot pedal, affects loudness or timbral change
    - Portamento Time – device is slider, affects glide from one note to the next
    - **Volume** – device is slider or pedal, affects volume
    - **Expression** – device is foot pedal, affects expression
        - Volume can be thought of as a fader on a console
        - Expression can be thought of as the pedal on an organ
- Examples of single precision controllers
    - Sustain, Sostenuto, Soft Pedal

## FOCUS ON THE IMPORTANT COMMANDS

- **Note On** – the command is sent when a key is depressed
  - Command — 2nd Byte — 3rd Byte
    9*n*H — *kk*H — *vv*H
    *n* = MIDI Channel number. 0H – FH (ch 1 – 16)
    *kk* = note number. 00H – 7FH (0 – 127)
    *vv* = note on velocity. 00H – 7FH (0 – 127)
  - MIDI can address 127 notes (compare with the piano's 88 keys)
  - Note 0 (or 00H) is five octaves below middle C (equivalent to a 64' organ stop)
  - Note 127 (or FFH) is over five octaves above middle C
  - Velocity is a measure of how fast a key is depressed
  - A low velocity value results from a key depressed slowly
  - A high velocity value results from a key depressed quickly
  - A velocity value of zero is the same as a note off command
  - Velocity can be used to control character of sound as well as volume level
  - A velocity sensitive keyboard is required to measure velocity
  - Most "MIDI" keyboards detect the full range of velocity values
  - Most organ keyboards do not detect velocity at all
  - Some organ manufacturers claim velocity sensitive keyboards but their keyboards do not sense all 127 potential levels of velocity
- **Note Off** – the command is sent when a key is released
  - Command — 2nd Byte — 3rd Byte
    8*n*H — *kk*H — *vv*H
    *n* = MIDI Channel number. 0H – FH (ch 1 – 16)
    *kk* = note number. 00H – 7FH (0 – 127)
    *vv* = note off velocity. 00H – 7FH (0 – 127)
  - Note off velocity is not always supported by keyboards or sound modules
- **Program Change**
  - Command — 2nd Byte
    C*n*H — *pp*H
    *n* = MIDI Channel number. 0H – FH (ch 1 – 16)
    *pp* = Program number. 00H – 7FH (prog 1 – 128)
  - Selects the sound (also called patch) or instrument (i.e. flute, trumpet, etc.)
  - Allows for only 128 distinct values hence the later addition of Bank Select Controller (see next command) to provide for more sound selection options
- **Bank Select** (Controller number 0, 32)
  - Command — 2nd Byte — 3rd Byte
    B*n*H — 00H — *mm*H
    B*n*H — 20H — *ll*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *mm* = MSB (Most Significant Byte). 00H – 7FH (bank 1 – 128)
    *ll* = LSB (Lest Significant Byte ). 00H – 7FH (bank 1 – 128)
  - Provides for 128 x 128 (or 16,384 possible) banks of program selections
  - Hence the number of possible programs selections is 16,384 x 128 (or 2,097,152)
  - Bank Select parameter important in working with most contemporary MIDI sound modules to allow proper and complete selection from all available sounds
  - Many organ manufacturers MIDI systems do not provide a complete implementation of the Bank Select feature
  - Program Change and Bank Select values are selectable and displayed in the **Rodgers** organ menu as PRG MSB LSB for each of the eight MIDI couplers

## CONTINUE FOCUS ON IMPORTANT COMMANDS

- **Volume** (Controller number 7)
  - Command 2nd Byte 3rd Byte
    B*n*H 07H *vv*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *vv* = Volume. 0H – 7FH (0 – 127)
  - Generally used as an overall volume control like a fader on a sound console
  - Command generally issued by a **Rodgers** organ when expression shoe is moved
- **Expression** (Controller number 11)
  - Command 2nd Byte 3rd Byte
    B*n*H 0BH *vv*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *vv* = Expression. 0H – 7FH (0 – 127)
  - Generally for temporary changes in volume like a pedal on an organ
  - Alternate command issued by a Rodgers organ when the expression shoe is moved. Option selected and displayed in the **Rodgers** organ menu for each of the eight MIDI couplers. **Rodgers** expression options are VOL, EXP, OFF.
- **Sustain** or Hold 1 (Controller number 64)
  - Command 2nd Byte 3rd Byte
    B*n*H 40H *vv*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *vv* = Control value. 0H – 7FH (0 – 127)
    0 – 63 = Off, 64 – 127 On
  - Used to imitate the sustain pedal on a piano
- **Sostenuto** (Controller number 66)
  - Command 2nd Byte 3rd Byte
    B*n*H 42H *vv*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *vv* = Control value. 0H – 7FH (0 – 127)
    0 – 63 = Off, 64 – 127 On
  - Used to imitate the sostenuto pedal on a piano
- **Soft** (Controller number 67)
  - Command 2nd Byte 3rd Byte
    B*n*H 43H *vv*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *vv* = Control value. 0H – 7FH (0 – 127)
    0 – 63 = Off, 64 – 127 On
  - Used to imitate the soft pedal on a piano
  - The **Rodgers** organ has a single foot switch attached to the left most expression shoe. The function of this switch is selectable and displayed in the **Rodgers** organ menu for each of the eight MIDI couplers as Sustain, Sostenuto or Soft.
- **Pan** (Controller number 10)
  - Command 2nd Byte 3rd Byte
    B*n*H 0AH *vv*H
    *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    *vv* = Control value. 0H – 7FH (0 – 127)
    00H – 40H – 7FH (Left – Center – Right)
  - Used to control pan position of the sound in stereo field between left and right
  - This value can be independently set and displayed in the **Rodgers** organ menu for each of the eight MIDI couplers

## CONTINUE FOCUS ON IMPORTANT COMMANDS

- **Reverb Depth** (Controller number 91)
  - Command: B*n*H; 2nd Byte: 5BH; 3rd Byte: *vv*H
    - *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    - *vv* = Control value. 0H – 7FH (0 – 127)
  - Used to control amount of sound module reverberation effect added to the sound
  - This value can be independently set and displayed in the **Rodgers** organ menu for each of the eight MIDI couplers
- **Chorus Depth** (Controller number 93)
  - Command: B*n*H; 2nd Byte: 5DH; 3rd Byte: *vv*H
    - *n* = MIDI channel number. 0H – FH (ch 1 – 16)
    - *vv* = Control value. 0H – 7FH (0 – 127)
  - Used to control amount of sound module chorus effect added to the sound
  - This value can be independently set and displayed in the **Rodgers** organ menu for each of the eight MIDI couplers
- **System Exclusive** – Example for Rodgers Allegiant 677 stop change

| Byte | Value | Function |
|---|---|---|
| 1 | F0H | Begin System Exclusive |
| 2 | 41H | Manufacture code for Roland/Rodgers |
| 3 | 10H | Device ID |
| 4 | 30H | Generic organ data |
| 5 | 12H | Data Set Command |
| 6 | 00H | Offset byte |
| 7 - | *vv*H | Data bytes, followed by checksum byte and End System Exclusive (F7H). Data bytes indicate which stops are changing state. |

## ADDITIONAL ITEMS AVAILABLE IN RODGERS MIDI COUPLER MENUS

- **MIDI Channel**
  - Value is displayed for all couplers
  - Value is selectable on Great MIDI A only
  - Used often to select MIDI Channel 10 for the various drum kits in a sound module
- **Octave**
  - Displays and allows user to select octave level of sound for each of the eight MIDI couplers
  - Options are Down 2, Down 1, Normal, Up 1, Up 2

## THE ORGAN CONSOLE IS THE IDEAL SOUND MODULE CONTROL WORK STATION

- Multiple keyboards – even one to play with the feet
- Convenient layout without having to stack keyboards and equipment
- One or more expression or volume control pedals
- Presets that control the entire set up (all keyboards at the same time)

## WHAT IS MISSING FROM THE AVERAGE ORGAN CONSOLE TO BE THE IDEAL SOUND MODULE CONTROL WORK STATION

- Velocity sensitive keyboards
- Ability to issue complete Bank Select commands
- Ability to set and store key voice control parameters for each available MIDI channel in presets (i.e. octave, pan, reverb, chorus, level and the like)
- Availability of sustain, sostenuto and soft controls with a foot switch
- Ability to control volume level of each MIDI part independently
- Current **Rodgers Masterpiece Trillium** organs offer each of the above standard in each console

## MAIN USES OF MIDI FOR ORGANISTS

- Sequencing
    - A device called a "sequencer" is used to capture a sequence of events for later playback
    - Data can be saved internally to sequencer or saved to disk or diskette
    - Possible uses
        - Record the string part of the Handel Organ Concertos to play back sequence while playing organ part
        - Record service prelude so the organist can warm up with choir
        - Record accompaniment so organist can play another instrument or conduct
- Control of one or more sound modules to create non-organ or additional organ sounds
- Control of other MIDI devices such as video, lights, etc.
- Entering notes in a music notation program like Finale or Sibelius

## ISSUES TO BE AWARE OF WITH MIDI

- The MIDI data protocol provides no error control
    - If a message transmission is corrupted in data transfer, the receiving device will either do something unexpected or ignore the command
    - Typically data transmission problems are the result of a bad MIDI cable or bad MIDI cable connection
    - A typical example is a cipher. The note on message is received by a device but the note off message is not.
- MIDI data is transmitted serially
    - If you play a four note chord (pressing all four notes at the same time), the notes are transmitted one after the other from one MIDI device to the next.
    - The subtle time difference is not usually detected by the human ear
    - If however a lot of parameters are changing at the same time, the MIDI transmission path can become overloaded and cause audible delays in the intended musical performance (i.e. a recorded sequence may not replay exactly as originally performed)

## RESOURCES USED IN PREPARING PRESENTATION

- MIDI for the Professional by Paul D. Lehrman and Tim Tully
- Roland and Rodgers Owner's Manuals for various devices