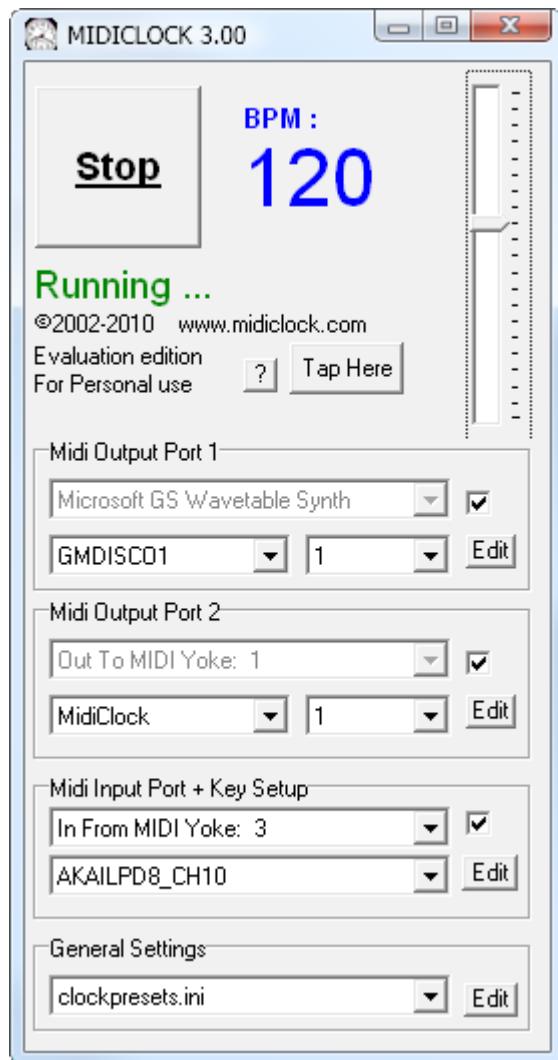


MIDIClock 3.00



Quick Guide by Serge

Rel 1

25 Sep 2010

www.midiclock.com

Introduction

- What is MIDIClock ?
 - A Program that can be used to synchronize synths, effects, sequencers, arpeggiators, ...
 - Its main function is generating a midi beat clock signal on a midi output device at a selectable bpm rate. Other output signals are also possible through custom output drivers
 - The software has no installer, simply unzip to where you need it. It is portable : it can be run from a removable disk/stick without installing anything.
 - The latest trial version of MIDIClock can be downloaded from midiclock.com.

What is the midi clock protocol?

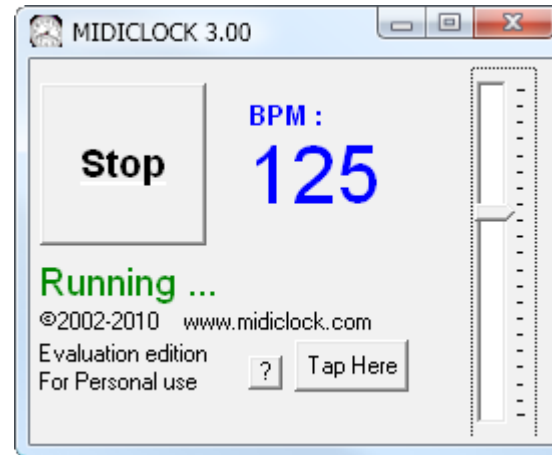
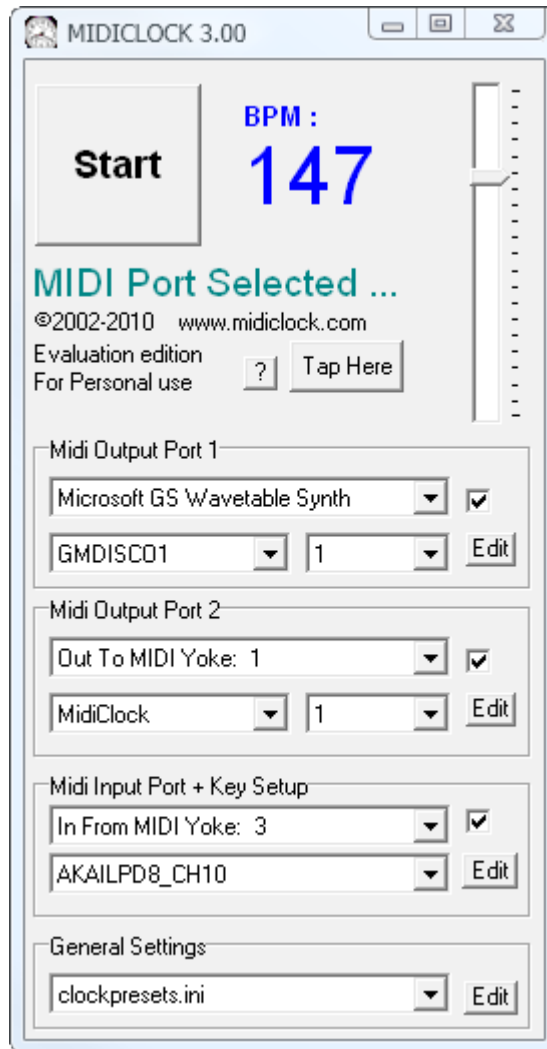
From wikipedia :

- MIDI beat clock is a clock signal that is broadcast via MIDI to ensure that several synthesizers stay in synchronization. It is not MIDI timecode.
- Unlike MIDI timecode, MIDI beat clock is sent at a rate that represents the current tempo, at 24 ppqn (pulses per quarter note). It is used to maintain a synchronized tempo for synthesizers that have BPM-dependent voices and also for arpeggiator synchronization. It does not transmit any location information (bar number or time code) and so must be used in conjunction with a positional reference (such as timecode) for complete sync.
- MIDI beat clock defines the following real time messages:
 - * clock (decimal 248, hex 0xF8)
 - * start (decimal 250, hex 0xFA)
 - * continue (decimal 251, hex 0xFB)
 - * stop (decimal 252, hex 0xFC)
- All the above real-time messages are supported by MIDIClock
- * tick (decimal 249, hex 0xF9) : this obsolete command is currently not supported.
If you need this command please let me know.

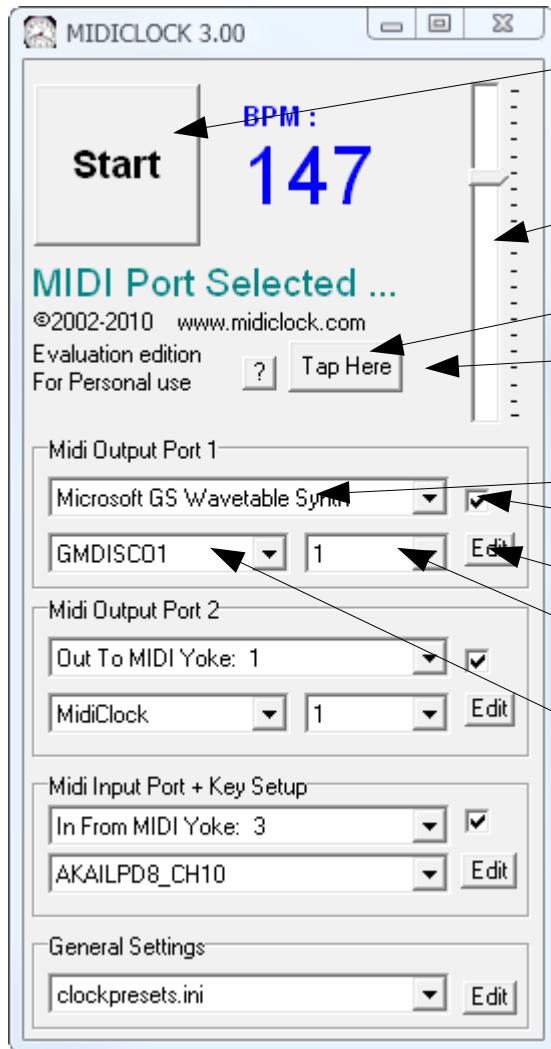
MIDIClock Features :

- Midiclock Registered :
 - Use for gigs, studio, live shows,...
 - 2 MIDI output drivers supported
 - Resizable Window
 - Up to 32 midi input action mappings
 - Registration & Support : midiclock.com/upgrade

MIDIClock Shapes

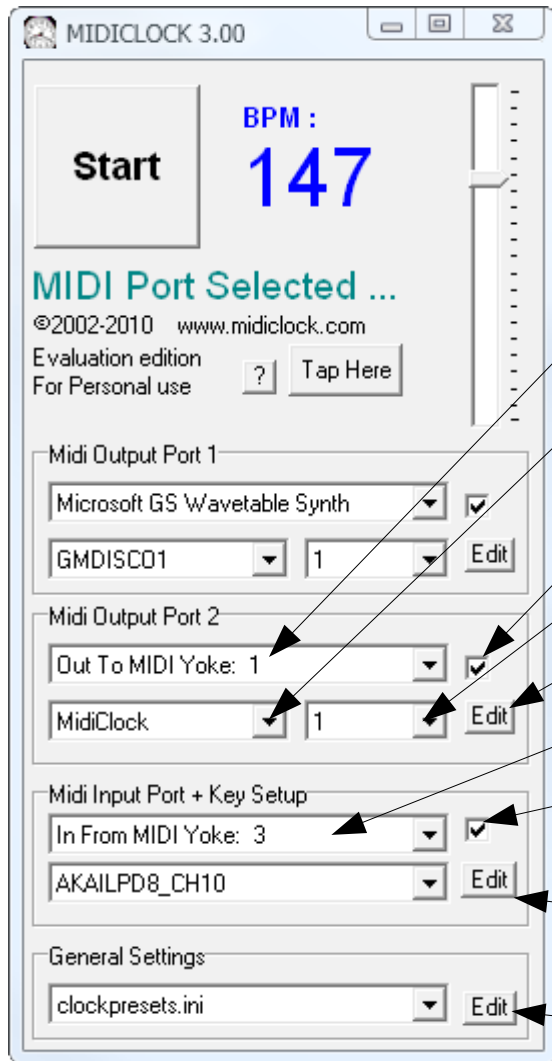


Main Screen functionality (1/2)



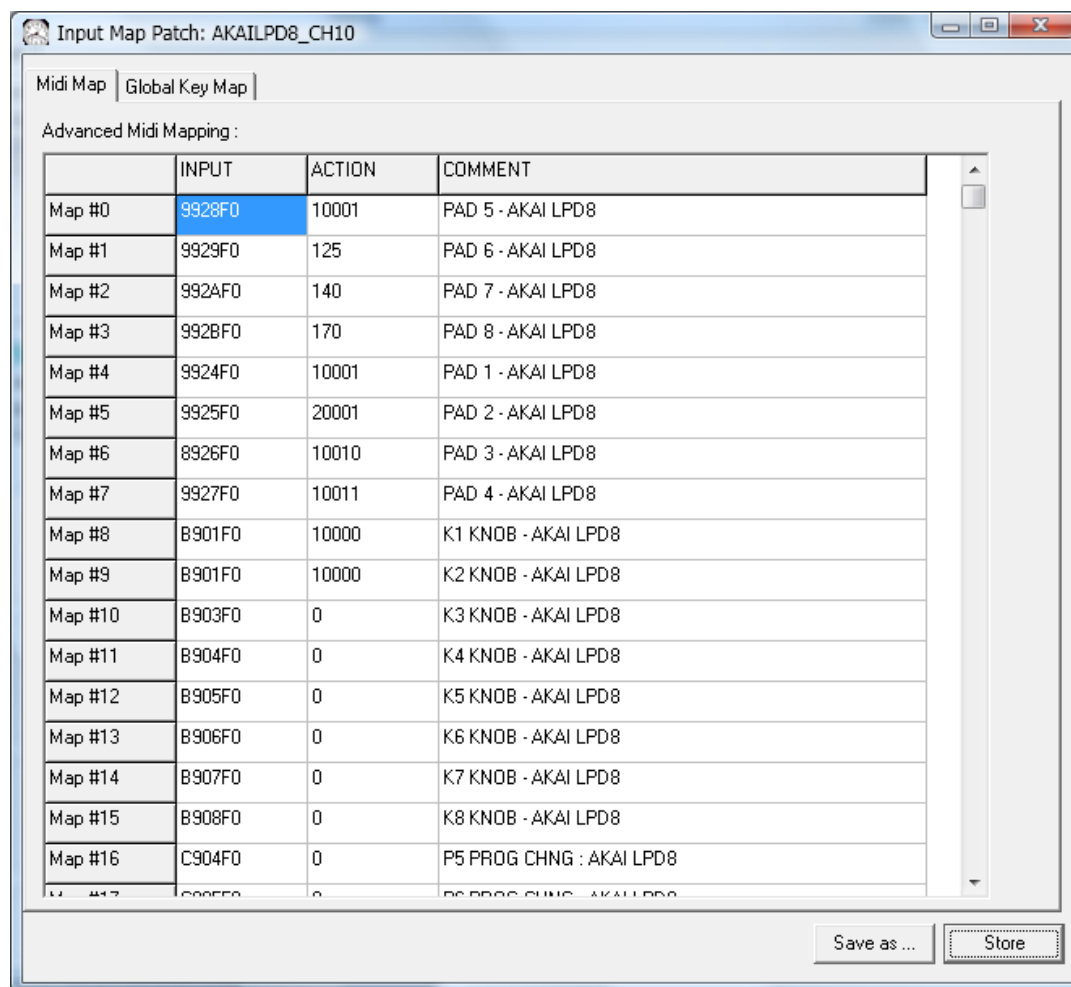
- Start /stop the timer
- Set the BPM rate
- Tap tempo input button
- Small form factor [Pro]
- Device #1 Selection
- Device #1 Enable/Disable
- Device #1 Config editing
- Device #1 Tempo divider
- Device #1 Output driver

Main Screen functionality (2/2)



- Device #2 Selection [Pro]
- Device #2 Output driver [Pro]
- Device #2 Enable/Disable [Pro]
- Device #2 Tempo divider [Pro]
- Device #2 Config editing [Pro]
- Input MIDI Device Selection
- Input MIDI Device Enable/Disable
- Input Map Editing
- Preset patch selection/edit

Input Patch Editor



- MIDI command to action mapping
- Click on any line to edit the mapping

Editing Input Map #0 (1/2)

Preset Edit : Map #0

Input
Hex : 99 28 F0
Status : 0x9c=Note On
Channel : 9
Par1Label : 40= E2
Par2Label : 240=0xF0: Accept All>0
Status : 9 ; Channel : 9; Bytes : 28 F0

Action
10001
10001=Start/Stop

Comments
PAD 5 - AKAI LPD8

Input Learning
Learn Mode Enable : ☐
☐ Status Filter
☐ Channel Filter
☒ Mask Range
☒ Auto Enter Hex

OK

• Status byte

• Midi channel

• Parameter 1 of midi message

• Parameter 2 of midi message

• Learning mode
--> updates map to incoming midi data

• Filter on status byte in learning mode

• Filter on selected channel for incoming data in learning mode

• Apply the mask range when data comes in, in learning mode

• Automatically update the hex code in the editor on input

Editing Input Map #0 (2/2)

Preset Edit : Map #0

Input
Hex : 99 28 F0
Status : 0x9c=Note On
Channel : 9
Par1Label : 40= E2
Par2Label : 240=0xF0: Accept All>0
Status : 9 ; Channel : 9; Bytes : 28 F0

Action
10001
10001=Start/Stop

Comments
PAD 5 - AKAI LPD8

Input Learning
Learn Mode Enable : ☐ ?
☐ Status Filter
☐ Channel Filter
☒ Mask Range
☒ Auto Enter Hex

OK

- Action mapped to the defined input midi code (for a list of actions see the special commands description at the end of this document)
- Descriptive comment for this map
- Press ok to apply the settings

Keyboard Shortcuts

- CTRL+ALT+S : save current settings to clockpresets.ini. It creates the file if it does not yet exist.
- F1/RETURN : Start/Stop timer toggle
- F2 : Pause timer
- F3 : Continue/Resume timer
- F5 : set BPM to Preset #5
- CTRL+F5 : Store current BPM rate at preset #5
- F6/F7/F8 : set BPM to Preset #6/7/8
- CTRL+F6/F7/F8 : Store BPM at preset #6/7/8
- SPACE : tap tempo trigger

File Overview

- **MIDIClock.exe** : The application binary. Clicking this application starts up midiclock.
- **clockpresets.ini** : Startup application presets. These are saved by pressing CTRL+ALT+S.
- **MidiclockRegistration.ini** (optional) : contains the key for registered users.
- ***.mcp** files : alternative application preset patches. Same file format as clockpresets.ini.
- **outdrv subdirectory** : contains output drivers
 - GMBASSDRUM.ini : a GM bass metronome
 - More output drivers available at midiclock.com
- **indrv subdirectory** : contains input drivers
 - AKAILPD8_CH10.ini : example driver for AKAI LPD8

Clockpresets.ini layout (1/3)

- **[APPLICATION_PRESETS] ; Section indicating general application presets**
 - MAINSCREENPOSITIONTOP=155 ; Main screen top pixel position at program startup
 - MAINSCREENPOSITIONLEFT=565 ; Main screen left pixel position at program startup
 - DEFAULTSPEED=120 ; BPM Rate used at program startup
 - SHOWFLASHING=0 ; 0 = disable window color flashing ; 1=enable window flashing at ¼ beat
 - FULLFLASHING=0 ; 1 = enable full window flashing ; 0 = disable
 - FLASHINGDURATION=1000 ; Color duration (unit = 1/24th of a beat). Set to 1000 for normal use.
 - FLASHINGPERIOD=24 ; Cycle period for the flashing, define in 1/24th of a beat units
 - IGNOREMOUSEWHEEL=0 ; 1=Disable mouse wheel influence on bpm rate
 - IGNOREKEYBOARDTAP=0 ; 1=Ignore keyboard 'SPACE' for tapping and 'RETURN' for start/stop
 - ALWAYSONTOP=0 ; 1= keep application window always in front of other applications
 - AUTOSTARTONTAP=0 ; 1= automatically start the midi clock signal at beat 1 after tapping beat 4
 - TAPTEMPOTIMEOUT = 5000 ; Tap tempo input state reset timeout (in milliseconds)
 - TAPTEMPOCOUNT=2 ; number of taps needed to trigger a tempo change (2, 3 or 4)

Clockpresets.ini layout (2/3)

- **[MIDIOUTPORT1] ; section containing the midi port #1 application settings**
 - MIDIPORTINDEX=1 ; Selection Order number of the midi port #1
 - MIDIPORTNAME=Out To MIDI Yoke: 1 ; Device name of the midi port #1
 - DEVICETYPE=MidiClock ; Output driver selected for this midi port #1
 - CLOCKDIVIDER=1 ; Divider used on the clock signal on port #1
- **[MIDIOUTPORT2] ; section containing the midi port #2 application settings (same structure)**
 - MIDIPORTINDEX=9 ; Selection Order number of the midi port #2
 - MIDIPORTNAME=E-MU Xmidi1X1 ; Device name of the midi port #2
 - DEVICETYPE=MidiClock ; Output driver selected for this midi port #2
 - CLOCKDIVIDER=1 ; Divider used on the clock signal on port #2
- **[MIDIINPORT1] ; section containing the midi input port application settings**
 - MIDIPORTINDEX=2 ; Selection Order number of the midi input port
 - MIDIPORTNAME=In From MIDI Yoke: 3 ; Device name of the midi input port
 - DEVICETYPE=AKAILPD8_CH10 ; Input driver selected for this midi port

Clockpresets.ini layout (3/3)

- **[BPM_PRESETS]** ; section containing BPM rate preset values.
 - F5BPM=100 ; pressing F5 will set the BPM rate to 100
 - F6BPM=120 ; pressing F6 will set the BPM rate to 120
 - F7BPM=140 ; pressing F7 will set the BPM rate to 140
 - F8BPM=160 ; pressing F8 will set the BPM rate to 160

Input driver layout

- **[SPECIAL_COMMANDS]** ; Powerful description of remote controlling midi clock by MIDI messages
 - INPUT0=933C40 ; MIDI Message to be parsed. (replace 933C40 with your midi message in hex)
 - ACTION0=150 ; Action to be taken when receiving that MIDI message.
 - ...up to INPUT31 : 32 INPUT/ACTION pairs can be defined.
- INPUT specifier :
 - Specifies the 3 midi bytes in hex (status byte + 2 data bytes) on which midiclock reacts
 - For example the midi message 933C40
 - The keyboard plays middle C (note number 60, hexadecimal 3C) on channel 4 (the zero nibble of 94), at half the full velocity (velocity 64, hexadecimal 40).
 - INPUT0=933C40 ; this would define the above midi message as input signal
 - ACTION0= 150 ; the action would be to set the bpm rate to 140 bpm
 - What about velocity sensitive keyboards ?
 - Use masking : using F0 as data byte 2 will make midiclock trigger on all non zero velocities
 - Example : INPUT0=933CF0 : for every middle C played (note on event), the ACTION0 will be run
 - More advanced options and usage for INPUT/ACTION will be described on the next pages.

Special Commands – Actions

- ACTION0=130 ; set BPM RATE to 130 when the INPUT0 MIDI Message is received
- The following target action codes can be executed on incoming midi commands :
 - 0 = do nothing
 - 1 --> 200 = set this specific BPM value
 - 10000 = perform midi value to tempo transformation (see further on for the input masking options)
 - 10001 = start-stop
 - 10002 = pause
 - 10003 = continue
 - 10004 = start
 - 10005 = stop
 - 10006 = decrease tempo
 - 10007 = increase tempo
 - 10008 = tempo - 2
 - 10009 = tempo + 2
 - 10010 = double tempo
 - 10011 = halve tempo
 - 20001 = tap tempo trigger
 - 20101 = FCB1010 native tap tempo control message support (experimental)

Special Commands – Tempo Transformation Input Masks

- In case of an action = 10000 --> several midi value to tempo transformation can be selected by input midi data byte 2
- This is useful for mapping a specific controller range onto a tempo range of choice.
- Midiclock will transform the incoming data byte 2 according to these formulas :
 - Mask byte 0xF0: From 0x00 to 0x7F --> multiplier = 1.58 and offset = 0
 - Mask byte 0xF1: From 0x00 to 0x7F --> multiplier = 1 and offset = 0
 - Mask byte 0xF2: From 0x00 to 0x7F --> multiplier = 2 and offset = 0
 - Mask byte 0xF3: From 0x00 to 0x7F --> multiplier = 3 and offset = 0
 - Mask byte 0xF4: From 0x00 to 0x7F --> multiplier = 4 and offset = 0
 - Mask byte 0xFA: From 0x00 to 0x3F --> multiplier = 2 and offset = 0
 - Mask byte 0xFB: From 0x40 to 0x7F --> multiplier = 2 and offset = 0
 - Mask byte 0xFC: From 0x00 to 0x3F --> multiplier = 2 and offset = 50
 - Mask byte 0xFD: From 0x40 to 0x7F --> multiplier = -2 and offset = 306
 - Mask byte 0xFF: From 0x01 to 0x7F --> multiplier = 2 and offset = 0

Any Questions ??

- Visit www.midiclock.com for more information...