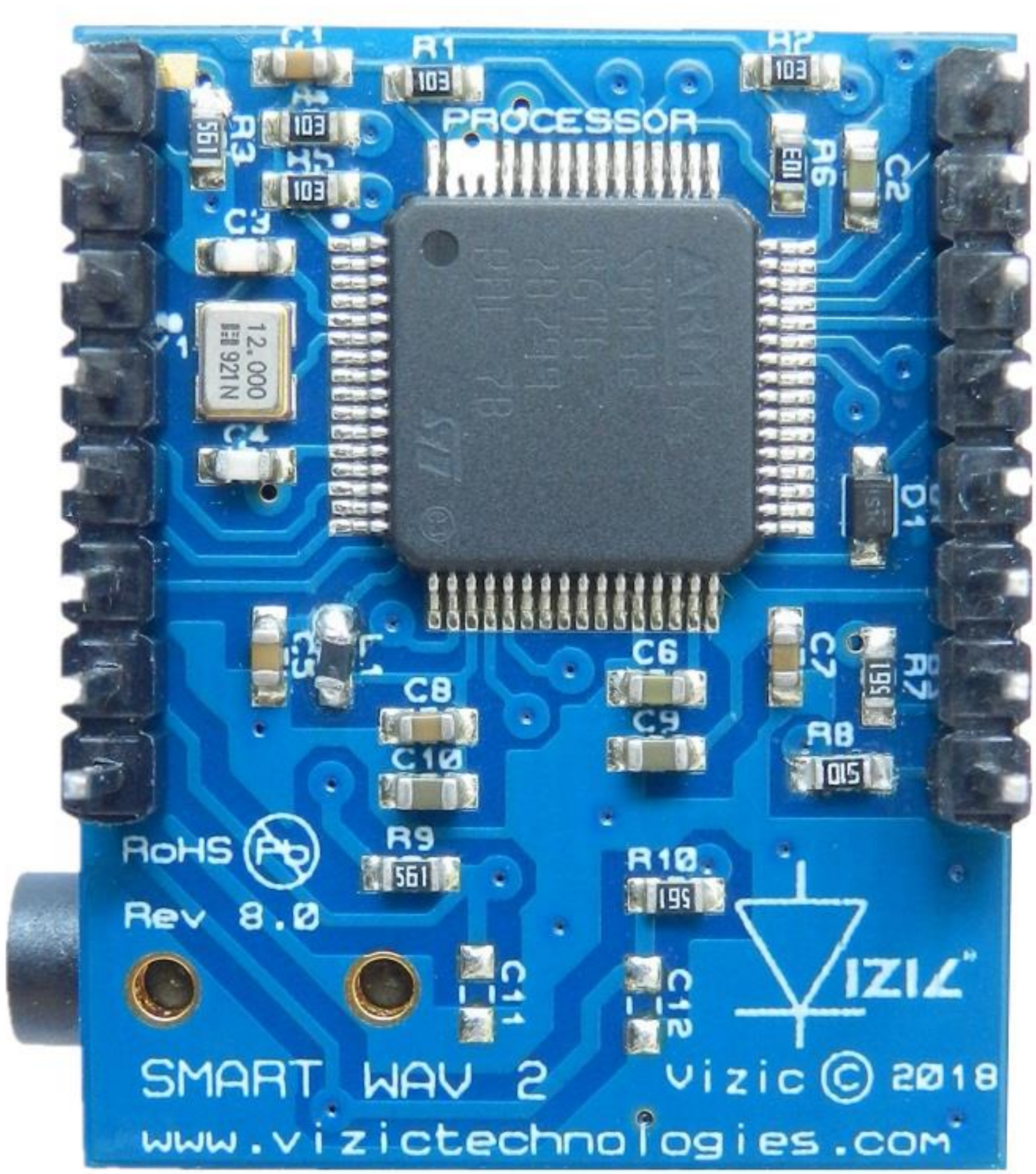
VIZIC TECHNOLOGIES

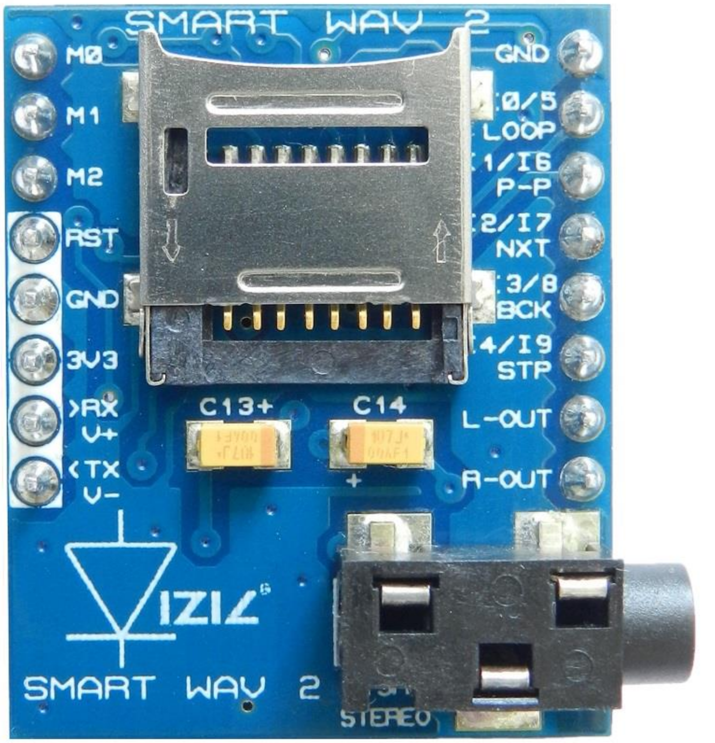
SMARTWAV 2

MIDI Mode----Rev 4.0

SmartWAV2 – Intelligent Embedded Audio Processor



SmartWAV2 Top View



SmartWAV2+MIDI Adapter Connections

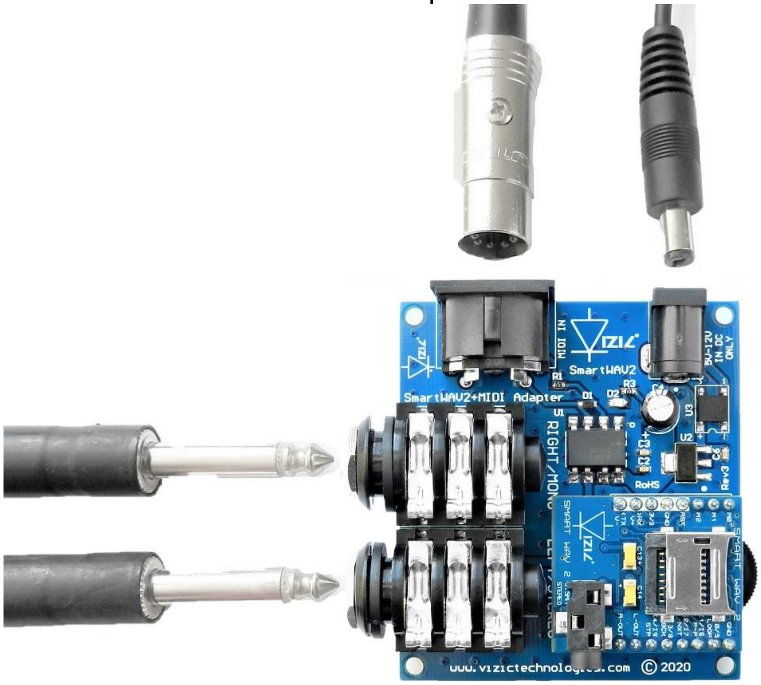


Table Of Contents:

1- SmartWAV2 4

1.1 Introduction...4

1.2 Features.…...5

1.3 Typical Applications...5

2- SmartWAV2 System Explained ………...6

3- SmartWAV2 Pinout...10

3.1 MIDI Serial Mode Pinout...11

4- SmartWAV2 MIDI Hardware Connections…...13

4.1 MIDI Serial Mode 4 Hardware Typical Connection.....................14

4.2 MIDI Serial Mode 4 Hardware Channel Selection.......................15

5- SmartWAV2 MIDI Supported Commands 17

5.1 Note OFF..17

5.2 Note ON..18

5.3 Polyphonic Key Pressure(Aftertouch)...18

5.4 Control Change…...18

5.5 Program Change……………………………………………..…………22

5.6 Channel Pressure……………………………………………………….22

5.7 Pitch Bend Change….…………………………………………………..23

5.8 System Common Messages……………………………………….23

6- SmartWAV2 MIDI Internal Handling...24

6.1 MIDI Notes/Tracks Structure..24

6.2 MIDI Program Change/Folders Structure.......................................25

7- SmartWAV2 MIDIConfig.txt File...28

8- MicroSD File/Folder Organization...31

9- Proprietary Information..32

10- Disclaimer of Warranties & Limitation of Liability.....................................32

1- SmartWAV2:

1.1- Introduction:

The SmartWAV2 is an Intellectual Property smart high-end audio processor running on a state-of-the-art ARM Cortex M4 chip. The processor is mounted on a development board for easy and fast development.

SmartWAV2 is polyphonic: can play up to 14 channels/voices at the same time with high quality stereo sound from a microSD card with universal **FAT/FAT32** format. The processor supports **8/16bit**, **8Khz - 48Khz, mono / stereo,** WAVE (**.WAV**) files CD Quality.

The main goal of the SmartWAV2 it's to bring a very easy way to add polyphonic CD high quality stereo audio to any application or project without the user having experience in handling, mixing, sampling or decoding audio.

SmartWAV2 is not a programmable device, no IDEs, programmers, nor debuggers are needed, it is a fully featured end device ready to plug and play, this document covers **MIDI mode**: SmartWAV2 acts as a slave MIDI receiver device that process standard MIDI commands.

SmartWAV2 supports polyphonic, velocity layer, looping, pitch bend, panning, and some other functionalities, enabling unlimited possibilities.

1.2- Features:

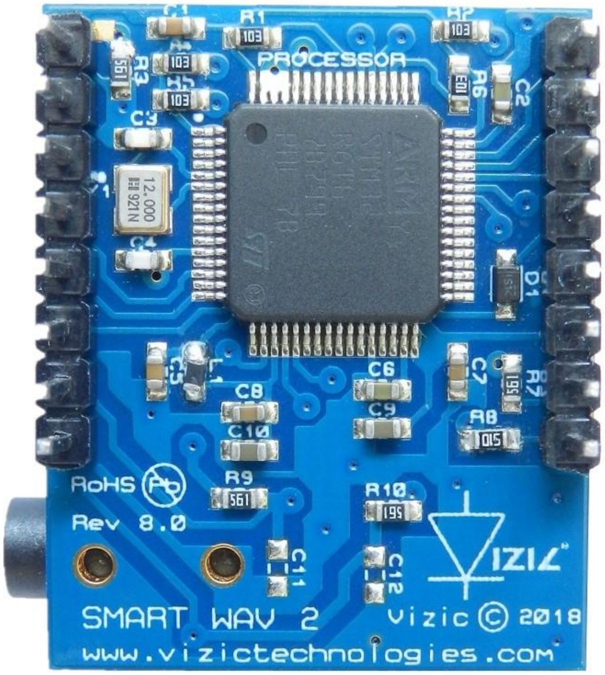
- Pre-amplified dual channel stereo with 8/16bit, stereo/mono, and up to 48khz sampling rate, CD quality.
- Full polyphonic 14 channel playback / auto mixing.
- Fully support Program Change messages from 0 to 127.
- Master gain digital volume control with 1/100 steps.
- MIDI mode compliant with channel selection and Omni mode.
- On board stereo 3.5mm plug line-out level.
- On-board uSD/uSDHC memory card socket with FAT/FAT32 support up to 32GB for storing thousands of tracks/audio **WAVE** files. No need of special/rare file formatting.
- MIDI note velocity layer support.
- Pitch bend, loop, release time and more.
- 5V and 3V3 I/O compatible, 3V3 powered.

1.3- Typical Applications:

- Embedded polyphonic audio/sound systems.
- Synthesizers.
- Samplers.
- Drum machines, drum modules.
- Loop stations, rhythm machines.
- Sequencers.
- All voice annunciator systems.
- Battery powered audio systems.
- Toys, learning tools, talking books, gaming sound.

2- SmartWAV2 System Explained:

Main Processor



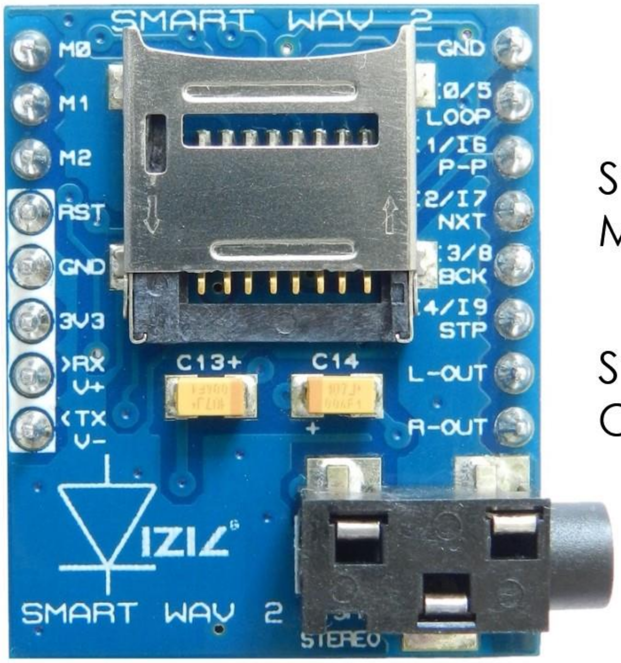
Stand-Alone Modes Inputs

Stereo Audio Outputs

Mode Selection Inputs

Serial Interface Inputs

MicroSD Card Socket



Mode Selection Inputs

Serial Interface Inputs

Stand-Alone Modes Inputs

Stereo Audio Outputs

3.5mm Stereo Output

SmartWAV2 processor is a sophisticated multi-task audio processor, it can playback up to 14 individual .wav tracks/songs/voices at the same time, triggering, decoding, sampling, mixing are automatic processes that can be controllable.

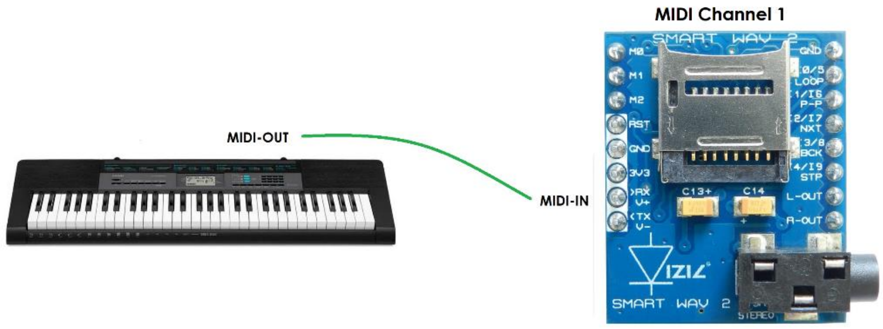
The module offers **8(eight)** modes of operation; selection between modes is by setting the external pins M0, M1 and M2, **this document covers only the Polyphonic Slave Standard Serial Mode 4(MIDI Mode)**, for more operation modes please refer to the document **SmartWAV2-Datasheet.pdf**.

It is strongly recommended to be familiar with the MIDI protocol, be sure to understand some basics of the MIDI messages structure.

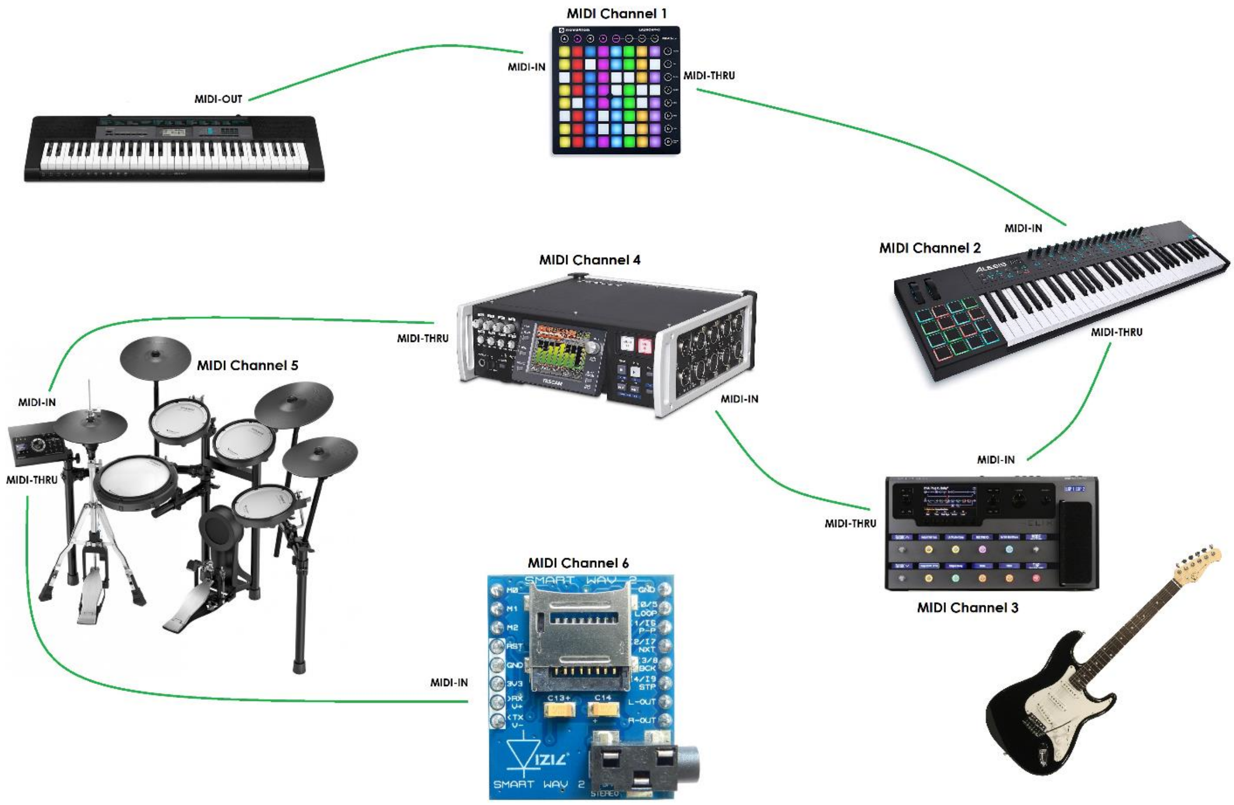
MIDI is a protocol primarily designed to communicate musical instruments using commands or messages, MIDI does not send any actual physical audio data or sounds like RCA cables, ¼" Jacks or even wireless Bluetooth connections, MIDI messages are just instructions/directions that are processed by other MIDI devices.

In MIDI protocol there is always one MIDI transmitter device and one or more MIDI receivers at the time, MIDI is designed to connect up to 16 different devices/instruments, each instrument/device must be assigned to a different **MIDI channel (1 to 16)** to not interfere with other devices connected in the chain (MIDI channel is similar to the address number of a house). MIDI devices only process MIDI messages that match their selected MIDI channel and non-matching channel messages are just ignored.

There are MIDI devices that can act only as MIDI-OUT transmitters, some other devices can only work as MIDI-IN receivers(SmartWAV2), and some other devices can do both, but only one device can be the transmitter at the time while the other 15 devices must act as MIDI receivers, the next illustration shows a typical chain of MIDI devices:



Simple MIDI interconnection



Multiple MIDI Interconnection

In the illustration, the top right keyboard is the MIDI transmitter that will be sending MIDI messages to each device, note how each MIDI receiver device has an assigned MIDI channel to only receive their matching channel messages.

MIDI commands are size defined bytes of data, this document won't cover the binaries of how those are created, but basically all MIDI commands are divided in 8 main commands:

- Note OFF.
- Note ON.
- Polyphonic Key Pressure (Aftertouch).
- Control Change.
- Program Change.
- Channel Pressure.
- Pitch Bend Change.
- System Common Messages.

Always all MIDI commands/messages contain at least the following information:

-Command Type (One of the above 8 main commands).
-MIDI Channel (1-16)(For the receiver channel).
-MIDI parameters (1 to n)(Size depends on the command type).

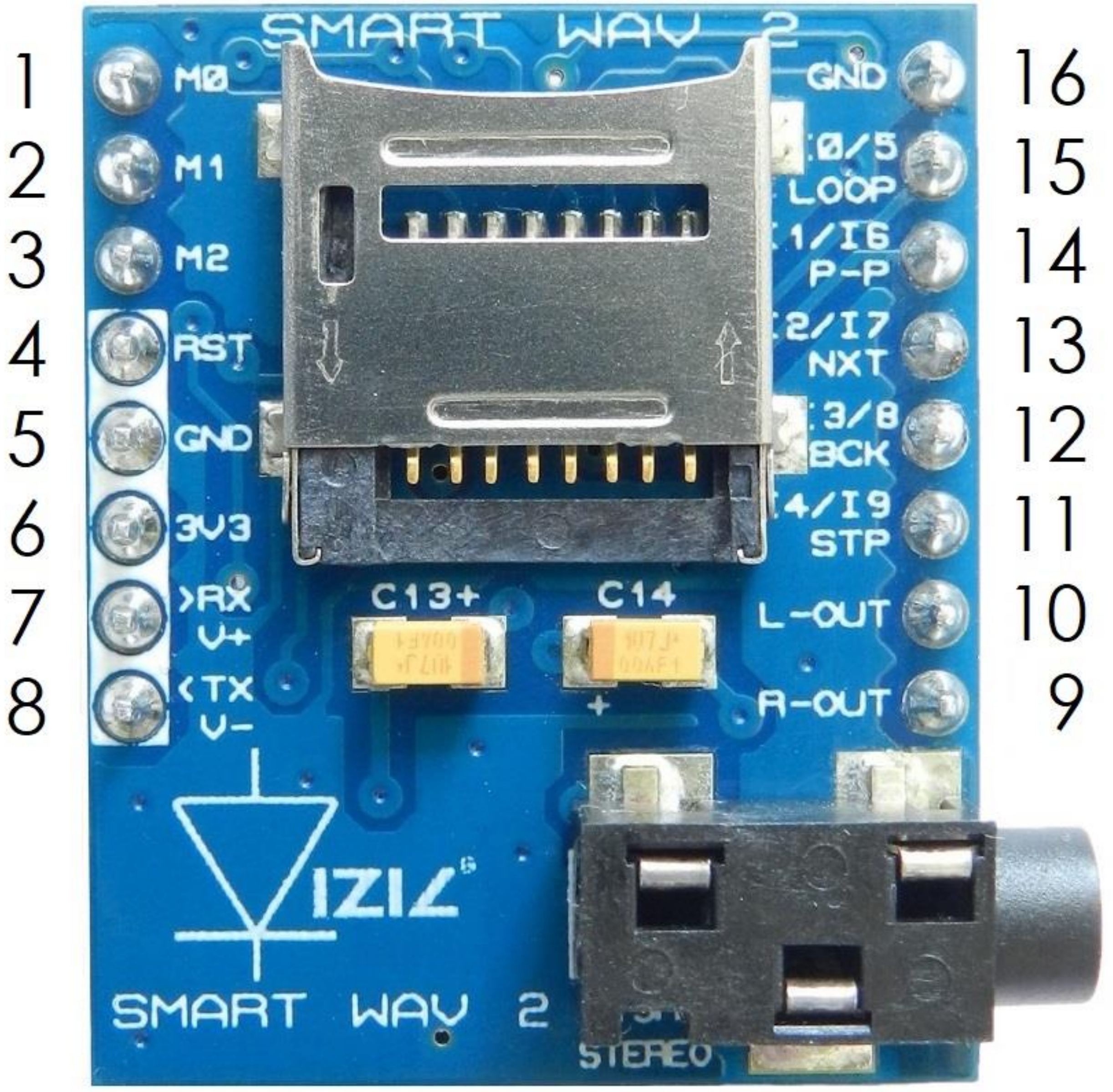
The most important and used commands are **Note ON**, **Note OFF**, **Program Change** and **Control Change**. In general, the commands that trigger notes/voices/sounds are **Note ON** and **Note OFF**.

The command that changes the type of voice/instrument is **Program Change**.

The **Control Change** commands are the one that contain configuration parameters like release time, master gain volume, loop, foot controller, etc.

The full MIDI protocol is very extensive and an official MIDI association specification of several pages is available on internet, however there is no need for the standard user to study this full specification as not all the commands are broadly used, neither not all MIDI devices support all the full specification commands, this is the case of SmartWAV2, it supports several commands, but not all of the full specification.

3- SmartWAV2 Pinout:



3.1- MIDI Serial Mode 4 Pinout:

Pin	Symbol	Function	Description
1	**M0**	**INPUT**	Digital input power ON/reset mode selection pin, the combination of M0, M1 and M2, determines the working mode of the processor. Internally pulled-down to ground via a 40K resistor.
2	**M1**	**INPUT**	Digital input power ON/reset mode selection pin, the combination of M0, M1 and M2, determines the working mode of the processor. Internally pulled-down to ground via a 40K resistor.
3	**M2**	**INPUT**	Digital input power ON/reset mode selection pin, the combination of M0, M1 and M2, determines the working mode of the processor. Internally pulled-down to ground via a 40K resistor.
4	**RESET**	**INPUT**	Digital input reset pin, an active low pulse greater than 100ns will reset the processor. Internally pulled-up to 3.3V via a 40K resistor. 5V tolerant input.
5	**GROUND**	**POWER**	Supply ground, be sure to have a well-grounded connection to avoid noise over the audio outputs.
6	**3.3V**	**POWER**	Main supply voltage: 2.7v-3.5v. Be sure to use a 10uF and 100nF coupling capacitors to GND as close as possible to avoid noise over the audio outputs.
7	**Rx - V+**	**INPUT**	Digital input pin MIDI IN. Refer to MIDI to TTL converter for connections. Internally pulled-up to 3.3V via a 40K resistor.

8	**TX - V-**	**NC**	Not used, leave unconnected.
9	**R-OUT**	**OUT**	Pre-amplified right channel audio output pin. Connect this pin instead of the 3.5mm plug connector, those are internally connected. (*Do not use both plug and this pin at the same time).
10	**L-OUT**	**OUT**	Pre-amplified left channel audio output pin. Connect this pin instead of the 3.5mm plug connector, those are internally connected. (*Do not use both plug and this pin at the same time).
11	**I4/I9 - STP**	**INPUT**	Digital input pin MIDI omni-channel enable/disable. Internally pulled-down to ground via a 40K resistor.
12	**I3/I8 - BCK**	**INPUT**	Digital input pin MIDI channel selection bit 3. Internally pulled-down to ground via a 40K resistor.
13	**I2/I7 - NXT**	**INPUT**	Digital input pin MIDI channel selection bit 2. Internally pulled-down to ground via a 40K resistor.
14	**I1/I6 - P/P**	**INPUT**	Digital input pin MIDI channel selection bit 1. Internally pulled-down to ground via a 40K resistor.
15	**I0/I5 - LOOP**	**INPUT**	Digital input pin MIDI channel selection bit 0. Internally pulled-down to ground via a 40K resistor.
16	**GND**	**POWER**	Supply ground, be sure to have a well-grounded connection to avoid noise over the audio outputs.

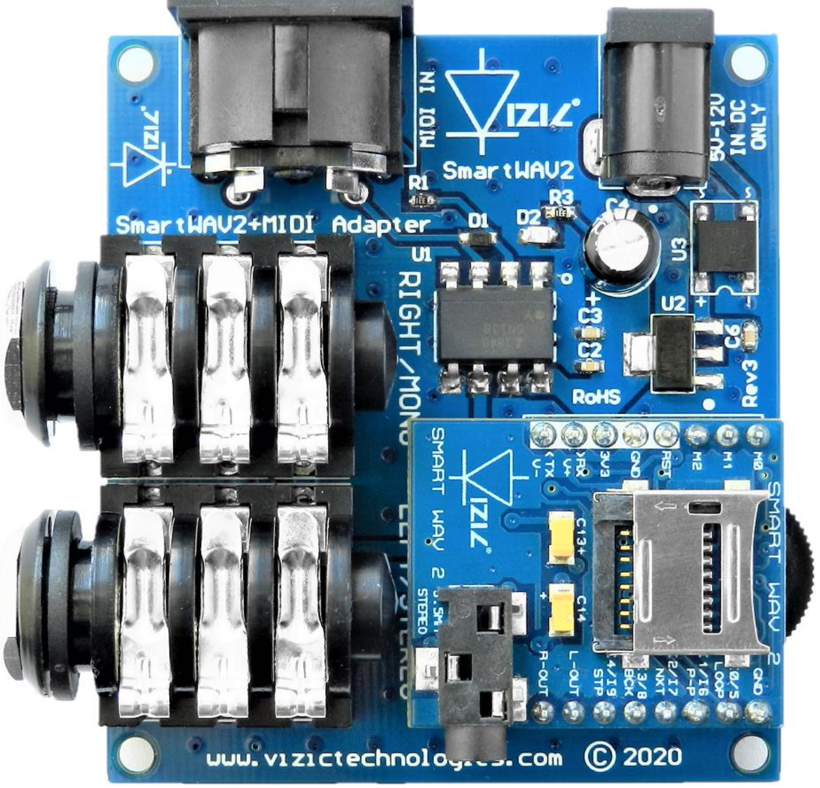
*__NC__ means no connect.

4- SmartWAV2 MIDI Hardware Connections:

Those next sections detail the required physical electrical connections to operate SmartWAV2 in MIDI mode: MIDI channel selection, audio outputs and MIDI-IN connection.

The **SmartWAV2+MIDI Adapter** is a simple board that simplifies all the required connections to operate the SmartWAV2 in MIDI mode. For users that does not have any or basic experience in electronics, the adapter exposes all the required **inputs** (MIDI-IN and DC power jack) and **outputs** (¼" Jacks) at the level of a MIDI musical instrument for plug and play, this adapter is an optional hardware but strongly recommended for MIDI applications.

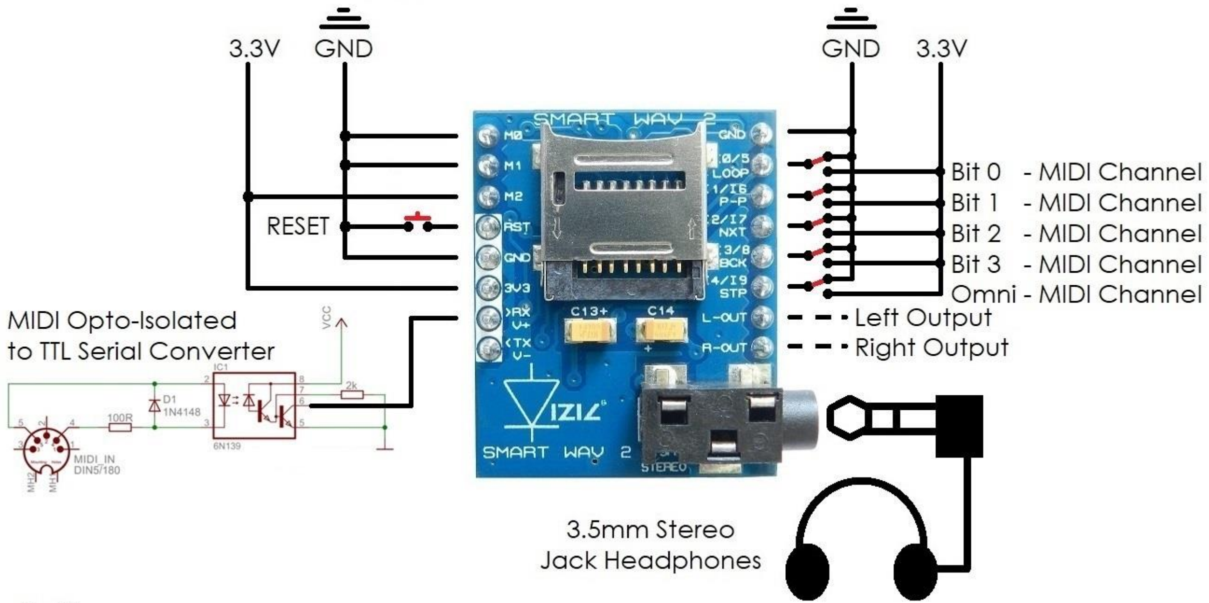
SmartWAV2+MIDI Adapter



SmartWAV2 is directly soldered to the adapter, just a standard DC power supply, a MIDI cable and the ¼" audio jacks are needed to operate the system.

4.1- MIDI Serial Mode 4 Hardware Typical Connection:

*The SmartWAV2 must always be powered with 2.7V-3.5V (Reset is 5V tolerant).

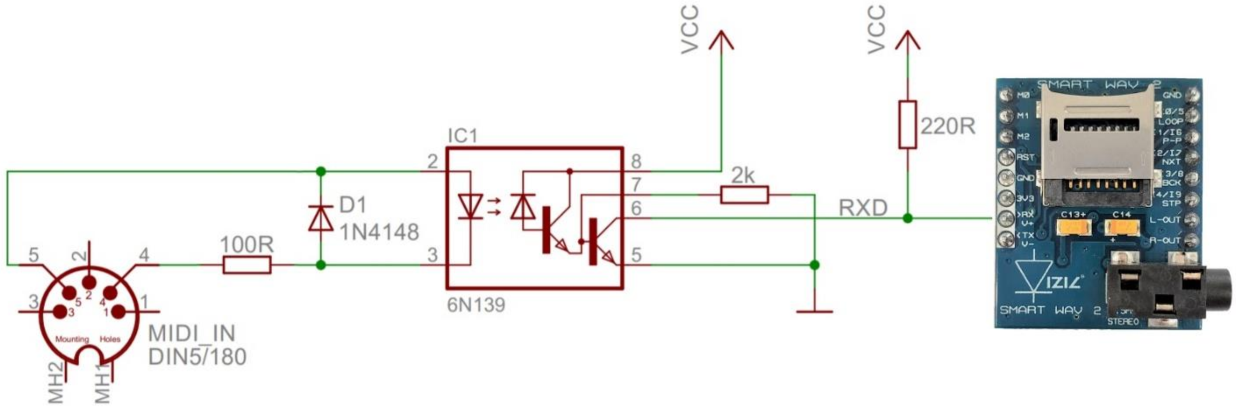


* Switch Symbol
* Push-Button Symbol

*Note: use 3.5mm plug or the L-Out & R-Out output pins, but not both at the same time!

To receive midi commands the SmartWAV2 uses it RX pin (MIDI-IN), MIDI works on a standard serial protocol (31250bps, 8bit, 1stop, no parity), however the hardware interface of a standard serial TTL USART/UART device is different from the hardware current-loop interface of MIDI(opto-isolated), in order to connect a MIDI-OUT to SmartWAV2 RX pin(MIDI-IN), a MIDI to serial TTL(0V-3.3V) converter is required:

MIDI to Serial TTL converter.



The MIDI(opto-isolated) to TTL(0V-3.3V) converter is a simple circuit just to convert the electrical current signals from the MIDI loop to the standard voltage TTL signals of the serial protocol.

The design of the **SmartWAV2+MIDI Adapter** is fully based on the electrical connections presented in this section, the adapter does integrate the MIDI to serial TTL adapter, the 3.3V power management and the audio outputs, therefore this section can be skipped if a SmartWAV2+MIDI Adapter is being used.

4.2- MIDI Serial Mode 4 Hardware Channel Selection:

The MIDI channel selection is via the I0-I3 input pins, those pins can be directly connected to VCC or GND, or using 2-way switches, this allow a logical binary selection of channel 0(0000b) to 15(1111b), however inside the processor this is handled as channel: n+1, resulting in a **1 to 16 channel** selection. The STP pin acts as a MIDI omni-channel enable/disable pin, when enabled (tied to VCC) it overrides the I0-I3 input pins, and therefore it accepts commands from all 16 MIDI channels.

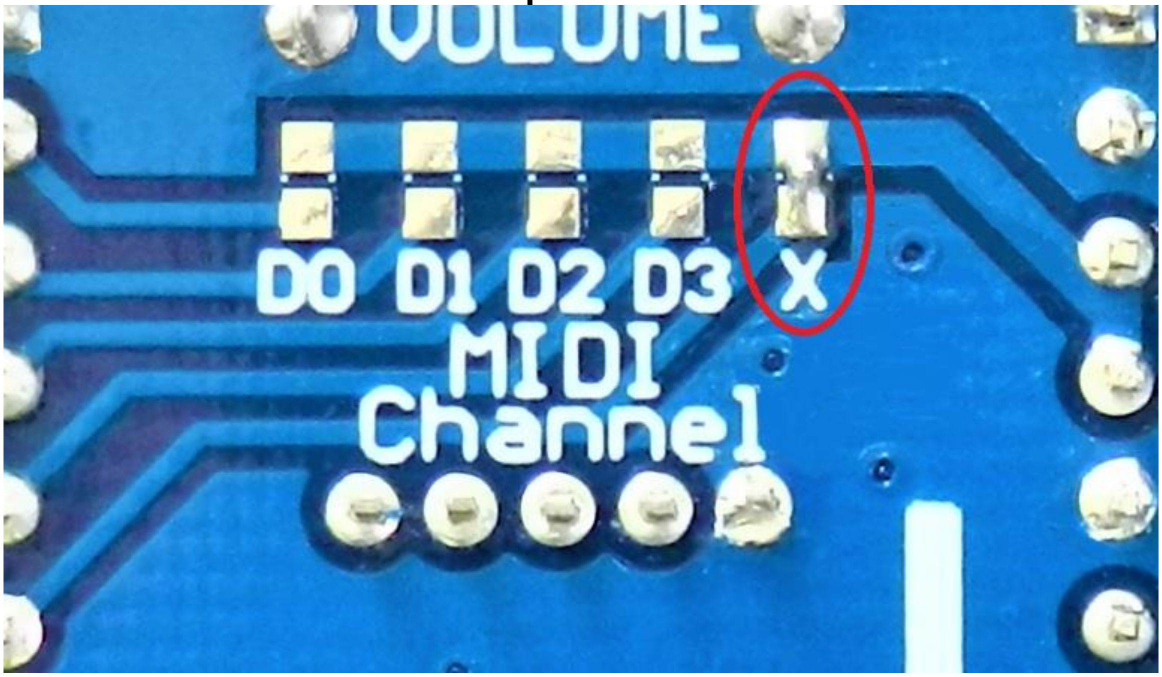
SmartWAV2 MIDI Channel Selection:

Channel #	I3/I8 BCK Pin	I2/I7 NXT Pin	I1/I6 P-P Pin	I0/I5 Loop Pin
Channel 1	GND	GND	GND	GND
Channel 2	GND	GND	GND	VCC
Channel 3	GND	GND	VCC	GND
Channel 4	GND	GND	VCC	VCC
Channel 5	GND	VCC	GND	GND
Channel 6	GND	VCC	GND	VCC
Channel 7	GND	VCC	VCC	GND
Channel 8	GND	VCC	VCC	VCC
Channel 9	VCC	GND	GND	GND
Channel 10	VCC	GND	GND	VCC
Channel 11	VCC	GND	VCC	GND
Channel 12	VCC	GND	VCC	VCC
Channel 13	VCC	VCC	GND	GND
Channel 14	VCC	VCC	GND	VCC
Channel 15	VCC	VCC	VCC	GND
Channel 16	VCC	VCC	VCC	VCC

Internally the smartWAV2 has pull-down resistors, so when any of the I0-I3 channel selection pins are not connected to VCC, those are automatically/internally connected to GND. As an example if no pin I0-I3 is connected and left floating, the resulting combination is 0000b, this means that smartWAV2 is configured to channel 1 and will only receive commands for this channel.

The design of the **SmartWAV2+MIDI Adapter** is fully based on the electrical connections presented in this section, the adapter contains a "MIDI Channel" part to binary select the channel via the **D3**, **D2**, **D1** and **D0** jumper pads, note that by default the **X** jumper (Marked in red) is soldered allowing the SmartWAV2 to be on omni-channel, meaning that the processor will receive MIDI messages from all the 1 to 16 MIDI channels.

SmartWAV2+MIDI Adapter: MIDI Channel Selection



When a jumper pad is soldered: the corresponding pin is connected to **VCC**, when the pad is left unsoldered: the pin is connected to **GND** (Refer to SmartWAV2 MIDI Channel Selection table for details).

5- SmartWAV2 MIDI Supported Commands:

SmartWAV2 acts as a MIDI Receiver device, providing a unidirectional serial interface to a MIDI controller via its MIDI-IN pin. Any MIDI compliant device can communicate with SmartWAV2 using standard MIDI commands, note on, note off, program change, etc.

The serial protocol over MIDI is universal:
Serial Data Format: 8 Bits, No Parity, 1 Stop Bit.
BaudRate: 31250 bps(default).
Serial data is true and not inverted.

This section details the supported commands that SmartWAV2 can process in MIDI mode.

MIDI commands are basically divided in 8 main commands:

- Note OFF.
- Note ON.
- Polyphonic Key Pressure (Aftertouch).
- Control Change.
- Program Change.
- Channel Pressure.
- Pitch Bend Change.
- System Common Messages.

Those commands will not be binary detailed in this document for simplicity purposes, if user need this information can refer to several online MIDI documentations.

5.1 Note OFF

The MIDI Note OFF command is supported; it basically stops the sound of the received note parameter based on a decreasing volume ramp defined by the set **Release Time**, this can be adjusted from 0ms up to 5,080ms. The velocity parameter for this command is always ignored and does not have some internal functionality.

5.2 Note ON

The MIDI Note ON command is supported; it basically triggers the sound of a received note/track; the velocity parameter determines the amount of volume for the track.

5.3 Polyphonic Key Pressure(Aftertouch)

This command type is not supported by SmartWAV2 and all incoming Polyphonic Key Pressure commands are ignored.

5.4 Control Change

Control Change commands can modify and set several functionalities on the run of a MIDI system, there's not a perfectly defined table of this commands that all manufactures follow and respect, but there are many commands that could be called standard among the most manufacturers of MIDI devices.

Control Change commands consist of 2 parameters: the command number(type), and the command value/parameter.

The next table shows the most standard MIDI commands and the ones in bold black are the ones that SmartWAV2 support, all other command numbers(types) are ignored:

Command Number(Type)	Function	Command Value
0	Bank Select	0-127
1	Modulation Wheel or Lever	0-127
2	Breath Controller	0-127
3	Undefined	-
4	Foot Controller	0-127
5	Portamento Time	0-127
6	Data Entry MSB	0-127
7	**Channel Volume(Master Gain)**	**0-127**
8	**Balance(Same as Panning)**	**0-127**
9	Undefined	-
10	**Panning(Same as Balance)**	**0-127**
11	Expression Controller	0-127
12	Effect Control 1	0-127
13	Effect Control 2	0-127

14-15	Undefined	-
16	General Purpose Controller 1	0-127
17	General Purpose Controller 2	0-127
18	General Purpose Controller 3	0-127
19	General Purpose Controller 4	0-127
20	**Tracks Loop Enable/Disable**	**<=63 Off, >=64 On**
21	**Layer Velocity Enable/Disable**	**<=63 Off, >=64 On**
22	**Note OFF Enable/Disable**	**<=63 Off, >=64 On**
23-29	Undefined	-
30	**Audio Error Messages Enable/Disable**	**<=63 Off, >=64 On**
31	Undefined	-
32	LSB for Control 0 (Bank Select)	0-127
33	LSB for Control 1 (Modulation Wheel)	0-127
34	LSB for Control 2 (Breath Controller)	0-127
35	LSB for Control 3 (Undefined)	-
36	LSB for Control 4 (Foot Controller)	0-127
37	LSB for Control 5 (Portamento Time)	0-127
38	LSB for Control 6 (Data Entry)	0-127
39	LSB for Control 7 (Channel Volume)	0-127
40	LSB for Control 8 (Balance)	0-127
41	LSB for Control 9 (Undefined)	-
42	LSB for Control 10 (Panning)	0-127
43	LSB for Control 11 (Expression Control)	0-127
44	LSB for Control 12 (Effect Control 1)	0-127
45	LSB for Control 13 (Effect Control 2)	0-127
46-47	LSB for Control 14-15 (Undefined)	-
48	LSB for Control 16 (Gen Purpose Ctrl 1)	0-127
49	LSB for Control 17 (Gen Purpose Ctrl 2)	0-127
50	LSB for Control 18 (Gen Purpose Ctrl 3)	0-127
51	LSB for Control 19 (Gen Purpose Ctrl 4)	0-127
52-63	LSB for Control 20-31 (Undefined)	-
64	**Damper Pedal On/Off (Sustain)**	**<=63 Off, >=64 On**
65	Portamento On/Off	<=63 Off, >=64 On
66	Sustenuto On/Off	<=63 Off, >=64 On
67	Soft Pedal On/Off	<=63 Off, >=64 On
68	Legato Footswitch	<=63 Off, >=64 On
69	Hold 2	<=63 Off, >=64 On
70	Sound Variation	0-127
71	Timbre/Harmonic Intens.	0-127
72	**Release Time**	**0-127**
73	Attack Time	0-127
74	Brightness	0-127
75	Decay Time	0-127
76	Vibrato Rate	0-127
77	Vibrato Depth	0-127
78	Vibrato Delay	0-127
79	Undefined	-
80	General Purpose Controller 5	0-127
81	General Purpose Controller 6	0-127
82	General Purpose Controller 7	0-127

83	General Purpose Controller 8	0-127
84	Portamento Control	0-127
85-87	Undefined	-
88	High Resolution Velocity Prefix	0-127
89-90	Undefined	-
91	Effects 1 Depth	0-127
92	Effects 2 Depth	0-127
93	Effects 3 Depth	0-127
94	Effects 4 Depth	0-127
95	Effects 5 Depth	0-127
96	Data Increment	-
97	Data Decrement	-
98	Non-Registered Parameter Number LSB	0-127
99	Non-Registered Parameter Number MSB	0-127
100	Registered Parameter Number LSB	0-127
101	Registered Parameter Number MSB	0-127
102-119	Undefined	-
120	**All Sound Off**	**-**
121	**Reset All Controllers(System Reset)**	**-**
122	Local Control On/Off	0 off, 127 on
123	**All Notes Off**	**-**
124	**Channel Omni Mode Off**	**-**
125	**Channel Omni Mode On**	**-**
126	Mono Mode On	-
127	Poly Mode On	-

In summary the next table shows the supported commands by SmartWAV2 and how it handles those:

Command Number(Type)	Function	Command Value
7	Channel Volume(Master Gain)	0-127
8	Balance(Same as Panning)	0-127
10	Panning(Same as Balance)	0-127
20	Tracks Loop Enable/Disable	<=63 Off, >=64 On
21	Layer Velocity Enable/Disable	<=63 Off, >=64 On
22	Note OFF Enable/Disable	<=63 Off, >=64 On
30	Audio Error Messages Enable/Disable	<=63 Off, >=64 On
64	Damper Pedal On/Off (Sustain)	<=63 Off, >=64 On
72	Release Time	0-127
120	All Sound Off	-
121	Reset All Controllers(System Reset)	-
123	All Notes Off	-
124	Channel Omni Mode Off	-
125	Channel Omni Mode On	-

-Channel Volume: this command modifies the internal **Master Gain** volume of the smartWAV2.

-Balance: this command modifies the internal panning of all the playing tracks inside smartWAV2.

-Panning: this command modifies the internal panning of all the playing tracks inside smartWAV2.

-Tracks Loop Enable/Disable: this command enables or disables the looping functionality of all the playing tracks inside smartWAV2, this command is useful for drum machines, to loop a rhythm track constantly.

-Layer Velocity Enable/Disable: this command enables or disables the note velocity layering functionality (Refer to MIDIConfig.txt file section).

-Note OFF Enable/Disable: when enabled, the SmartWAV2 ignores all incoming "Note OFF" commands (Useful for percussion sounds). (Refer to MIDIConfig.txt file section).

-Audio Error Messages Enable/Disable: this command permanently enables or disables the throwing of audible messages "No Memory Card Detected" when SmartWAV2 is powered and no microSD card is detected, by default those messages are enabled.

-Damper Pedal On/On (Sustain): This command activates/deactivates the standard sustain functionality of all playing notes.

-Release Time: this parameter is used to set the standard release time, each unit of this received parameter increases/multiplies by 40ms of time (Release Time = received parameter * 40ms), that means that smartWAV2 can go from 0ms (received value 0) up to 5,080ms (received value 127).

-All Sound Off: this command stops all playing sounds/notes.

-Reset All Controllers: this command performs an internal system reset of SmartWAV2, same effect as a hardware reset.

-All Notes Off: this command stops all playing sounds/notes.

-Channel Omni Mode Off: this command disables the channel omni mode and performs a reading on the hardware MIDI channel pins of SmartWAV2 to set the MIDI channel, after this command is called, only commands from 1 of the 16 MIDI channels will be processed.

-Channel Omni Mode On: this command enables the channel omni mode and overrides any hardware MIDI channel pins configuration, after this command is called, commands from all 16 MIDI channels will be processed.

5.5 Program Change

Program Change commands modify the patch, instrument, voice or sound type of a MIDI system, SmartWAV2 uses this command to change/Go-in Go-Out folders named from “000” to “127” based on the received program number parameter, if the MIDI host that will control SmartWAV2 does not support program change commands, then the “.wav” files must be placed on the root path (Refer to Section 5.3- MIDI Program Change/Folders Structure).

5.6 Channel Pressure

This command type is not supported by SmartWAV2 and all incoming Channel Pressure commands are ignored.

5.7 Pitch Bend Change

This command modifies the sound of the playing notes/tracks by changing the pitch from -2 to +2 half tones depending on the received pitch parameter.

5.8 System Common Messages

System common messages are not musical commands; those refer to basic control/configurations for a MIDI system. The next table shows all the MIDI system common messages, the ones in bold black are the ones that SmartWAV2 support, all other commands are ignored:

Command(hex)	Function
0xF0	Start Of System Exclusive Message
0xF1	MIDI Time code Quarter Frame
0xF2	Song Position Pointer
0xF3	Song Select
0xF4	Undefined
0xF5	Undefined
0xF6	Tune Request
0xF7	End Of System Exclusive Message
0xF8	Timing Clock
0xF9	Undefined
0xFA	Start
0xFB	Continue
0xFC	**Stop(Stops All Sounds)**
0xFD	Undefined
0xFE	Active Sensing
0xFF	**System Reset**

-Stop (Stops All Sounds): this command stops all playing sounds/notes.

-System Reset: this command performs an internal system reset of SmartWAV2, same effect as a hardware reset.

6- SmartWAV2 MIDI Internal Handling:

6.1- MIDI Notes/Tracks Structure:

SmartWAV2 can handle **Note ON** and **Note OFF** commands, those are intended to trigger notes or sounds, basically those commands contain the next parameters:

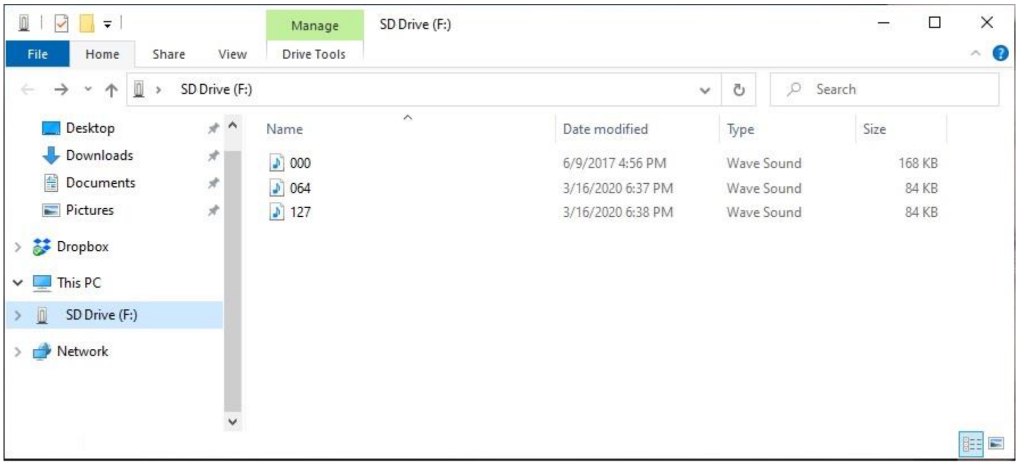
-Command Type (Note ON/Note OFF)
-MIDI Channel (1-16)
-Note Number (0-127)
-Velocity (Volume for the track)

SmartWAV2 can trigger notes/tracks based on the received **Note Number** if the ".wav" files are stored/named in the next manners:

-Note 0 -must be stored/named as "000.wav".
-Note 64 -must be stored/named as "064.wav".
-Note 127 -must be stored/named as "127.wav".

Please check that the note number/track name is 3 characters long (filled with zeros) plus the ".wav" extension.

SmartWAV2 will trigger ".wav" note number/tracks inside the current selected Folder Path/Program/Patch Number. The next image shows an example of the root path containing the note numbers 0, 64 and 127 in the correct number naming structure:



Name	Date modified	Type	Size
000	6/9/2017 4:56 PM	Wave Sound	168 KB
064	3/16/2020 6:37 PM	Wave Sound	84 KB
127	3/16/2020 6:38 PM	Wave Sound	84 KB

6.2- MIDI Program Change/Folders Structure:

SmartWAV2 can handle **Program Change** commands, in a MIDI system those are intended to change the patch, instrument, voice or sound type (Change the notes sound to guitar, piano, etc.). Inside SmartWAV2, those commands will change the current selected folder.

The next is a semi-standardized table showing the corresponding program numbers with the instrument / voice type that most manufacturers implement in their MIDI systems, therefore is not mandatory and guaranteed to work with all MIDI systems:

Table of semi-standardized program change numbers and instruments.

0	Grand piano	32	Acoustic bass	64	Soprano sax	96	Ice rain
1	Bright piano	33	Fingered bass	65	Alto sax	97	Soundtrack
2	Electric piano	34	Picked bass	66	Tenor sax	98	Crystal
3	Honky tonk piano	35	Fretless bass	67	Baritone sax	99	Atmosphere
4	Electric piano 1	36	Slap bass 1	68	Oboe	100	Brightness
5	Electric piano 2	37	Slap bass 2	69	English horn	101	Goblins
6	Harpsicord	38	Synth bass 1	70	Bassoon	102	Echo drops
7	Clavinet	39	Synth bass 2	71	Clarinet	103	Sci fi
8	Celesta	40	Violin	72	Piccolo	104	Sitar
9	Glockenspiel	41	Viola	73	Flute	105	Banjo
10	Music box	42	Cello	74	Recorder	106	Shamisen
11	Vibraphone	43	Contrabass	75	Pan flute	107	Koto
12	Marimba	44	Tremolo strings	76	Bottle blow	108	Kalimba
13	Xylophone	45	Pizzicato strings	77	Shakuhachi	109	Bag pipe
14	Tubular bell	46	Orchestral strings	78	Whistle	110	Fiddle
15	Dulcimer	47	Timpani	79	Ocarina	111	Shanai
16	Hammond organ	48	String ensemble 1	80	Synth square wave	112	Tinkle bell
17	Percussive organ	49	String ensemble 2	81	Synth saw wave	113	Agogo
18	Rock organ	50	Synth strings 1	82	Synth calliope	114	Steel drums
19	Church organ	51	Synth strings 2	83	Synth chiff	115	Woodblock
20	Reed organ	52	Choir aahs	84	Synth charang	116	Taiko drum
21	Accordion	53	Voice oohs	85	Synth voice	117	Melodic tom
22	Harmonica	54	Synth choir / voice	86	Synth fifths saw	118	Synth drum
23	Tango accordion	55	Orchestra hit	87	Synth brass and lead	119	Reverse cymbal
24	Nylon string guitar	56	Trumpet	88	Fantasia / new age	120	Guitar fret noise
25	Steel string guitar	57	Trombone	89	Warm pad	121	Breath noise
26	Jazz electric guitar	58	Tuba	90	Polysynth	122	Seashore
27	Clean electric guitar	59	Muted trumpet	91	Space vox / choir	123	Bird tweet
28	Muted electric guitar	60	French horn	92	Bowed glass	124	Telephone ring
29	Overdriven guitar	61	Brass ensemble	93	Metal pad	125	Helicopter
30	Distortion guitar	62	Synth brass 1	94	Halo pad	126	Applause
31	Guitar harmonics	63	Synth brass 2	95	Sweep pad	127	Gunshot

Based on the table: folder “000” will contain Grand Piano sounds, “064” folder Soprano Sax and “127” will contain Gunshot sounds.

Program Change command is the second most important in a MIDI system, basically those contain the next parameters:

-Command Type (Program Change)
-MIDI Channel (1-16)
-Program/Patch Number (0-127)

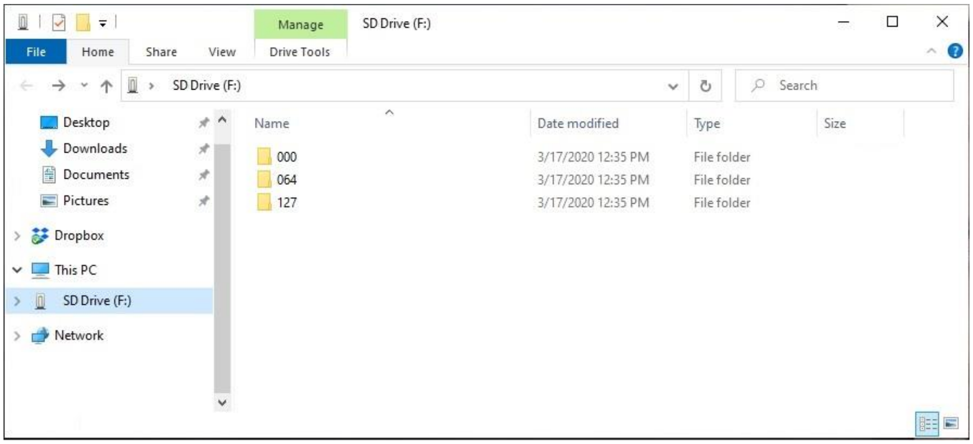
SmartWAV2 can change folders based on the received **Program / Patch Number** if the folder names are created in the next manners:

-Program/Patch Number 0 refers to a folder named "000".
-Program/Patch Number 64 refers to a folder named "064".
-Program/Patch Number 127 refers to a folder named "127".

Please check that the folder name is 3 chars long (filled with zeros).

Once SmartWAV2 starts and Program Change commands are received, it will change or jump (go inside, go outside) folders that are inside the **master folder**, this master folder path is fixed during system run and normally/by default it is the root path (Refer to master folder in section SmartWAV2 MIDIConfig.txt file).

The next image shows an example of a microSD card "root path" contents, the patch/folder numbers 0, 64 and 127 in the correct number naming structure:



Name	Date modified	Type	Size
000	3/17/2020 12:35 PM	File folder	
064	3/17/2020 12:35 PM	File folder	
127	3/17/2020 12:35 PM	File folder	

Based on the above microSD card contents, once a **Program Change** command with Program/Patch Number 64 is received, the system will go inside folder “064” and consecutive **Note ON** commands will try to trigger “.wav” files that are inside “064” folder. If a second **Program Change** command with Program/Patch Number 0 is received, the system will now go out of folder “064” and enter inside folder “000”, then consecutive **Note ON** commands will try to trigger “.wav” files that are inside “000” folder. In the case a folder does not exist or can’t be opened, the system will return to the **master folder** or root path.

If the MIDI host that will control SmartWAV2 does not support Program Change commands, then the “.wav” files must be placed on the **master folder** or root path instead of “000” to “127” folders.

7- SmartWAV2 MIDIConfig.txt File:

At system boot/start SmartWAV2 always look for a file called "MIDIConfig.txt" in the root path, this file is used to provide SmartWAV2 with some configurations and parameters, this file is optional as are its parameters, if any of the next configuration parameters are not present, the defaults will be taken.

-Master Gain: If the file contains the word **MASTERGAIN:** followed by a number between **0-100**, SmartWAV2 will take this and set the gain to the system, this can be used as a "start volume". Default master gain volume is 100.

-Release Time: If the file contains the word **RELEASETIME:** followed by a number between **0-127**, SmartWAV2 will take this and set the system release time. Default release time is 400ms.

-Hi-Hat Close Note: If the file contains the word **HHCLOSENOTE**: followed by a number between **0-127**, SmartWAV2 will take this note and save during system execution.

-Hi-Hat Open Note: If the file contains the word **HHOPENNOTE:** followed by a number between **0-127**, SmartWAV2 will take this note and if during system execution, a "Note On" command with note number same as the saved **HHCLOSENOTE** is received, the system will automatically stop playing the track number **HHOPENNOTE** (open hi-hat note sound), this will simulate a hi-hat close pedal action. This functionality is intended for drum modules / percussion applications.

-Sample Rate: If the file contains the word **SAMPLERATE:** followed by a number between **8000-44100**, SmartWAV2 will take this and set the system main sample rate. Default main sample rate is 44100.

-Loop: If the file contains the word **LOOP**, SmartWAV2 will enable the looping functionality of the system. Default loop is disabled.

-Note OFF Disable: If the file contains the word **NOTEOFFDISABLE**, SmartWAV2 will disable/ignore all Note OFF commands during system run, this functionality is great percussion sounds, where there is no need to receive Note OFF commands. Default Note OFF is enabled.

-Layer Velocity: If the file contains the word **LAYERVEL**, SmartWAV2 will enable the velocity layering functionality of the system, this means that during system execution, received notes will trigger tracks based not only on the note number, but also on the **velocity** parameter.

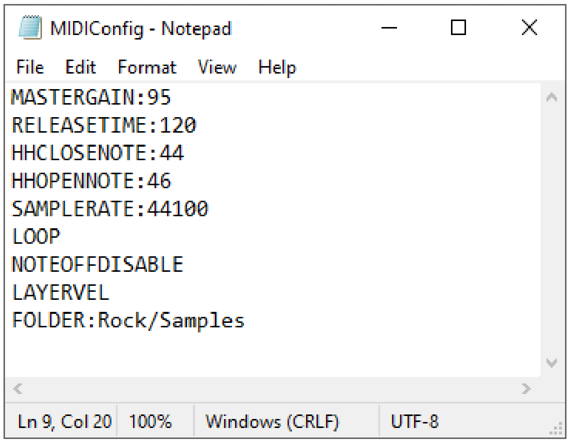
SmartWAV2 support 4 velocity layers, so once this functionality is enabled, the system will trigger note names with a suffix 'a', 'b', 'c' or 'd', the next table illustrates how the ".wav" files inside the microSD card must be named:

Note #	Layer Velocity Disabled(default)	Layer Velocity Enabled, each note has 4 ".wav" files.
0	"000.wav"	"000a.wav", "000b.wav", "000c.wav", "000d.wav"
64	"064.wav"	"064a.wav", "064b.wav", "064c.wav", "064d.wav"
127	"127.wav"	"127a.wav", "127b.wav", "127c.wav", "127d.wav"

This functionality is useful for percussion sounds as for example a snare drum does not sound equal being hit at slow(a), medium low(b), medium high(c) or fast(d) speed, user can store different sounding ".wav" files for those 4 layers. *By default layer velocity is disabled.

-Master Folder: If the file contains the word **FOLDER:** followed by a folder name, SmartWAV2 will try to open and change the folder path from the root to the defined **master folder**, this means that up from here the system will perform the normal functionality inside this defined folder instead of from the microSD card root path. The defined master folder can contain the character '/' to denote folder nesting. Default master folder is the root path.

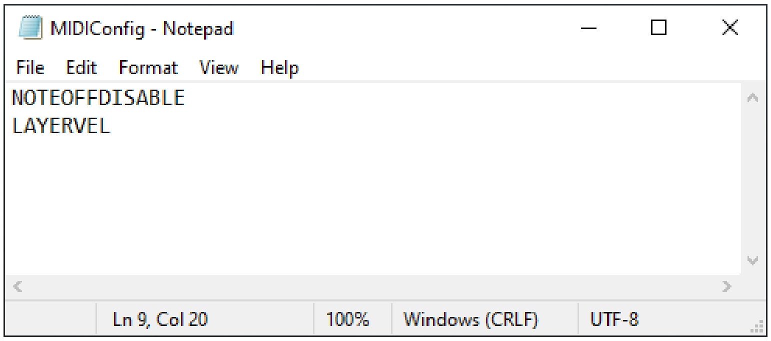
The next image shows an example of a "MIDIConfig.txt" file:



MIDIConfig - Notepad
File Edit Format View Help
Ln 9, Col 20 100% Windows (CRLF) UTF-8

```
MASTERGAIN:95
RELEASETIME:120
HHCLOSENOTE:44
HHOPENNOTE:46
SAMPLERATE:44100
LOOP
NOTEOFFDISABLE
LAYERVEL
FOLDER:Rock/Samples
```

*Note that it is not strictly necessary that the file contains all of the "words", but only the required ones, a file can contain 1, 2 or any words, example:



MIDIConfig - Notepad
File Edit Format View Help
Ln 9, Col 20 100% Windows (CRLF) UTF-8

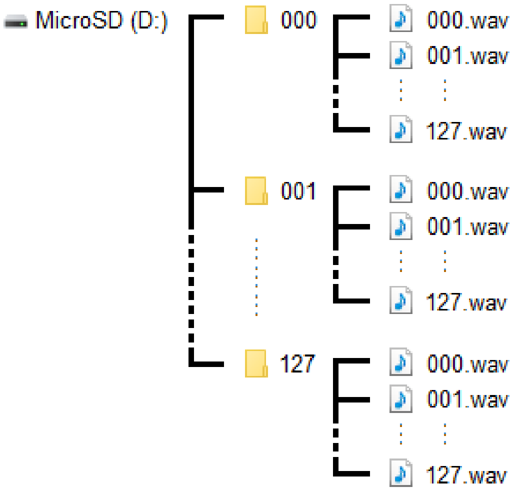
```
NOTEOFFDISABLE
LAYERVEL
```

8-MicroSD File/Folder Organization:

The SmartWAV2 is capable of managing folders via the **Program Change** commands:

The next image shows the folder and files structure of a full MIDI system:

MicroSD Card File Contents



The Image shows how folders from 000 to 127 are inside the root path, those denote **Program Change** possible values, inside each of the folders there are files from 0 to 127 showing the name of the actual notes/tracks to be triggered via **Note ON** commands

In this case is the master folder is the root path (Refer to master folder in section SmartWAV2 MIDIConfig.txt file).

*It is not strictly necessary that the microSD card contains all the 000 to 127 folders, neither all the 000 to 127 ".wav" notes for each folder, the microSD card can be loaded with only the patches/sounds and ".wav" notes that will be required by the system.

*If the MIDI host that will control SmartWAV2 does not support Program Change commands, then the ".wav" files must be placed on the root path and not inside any folder.

9- Proprietary Information:

The information contained in this document is the property of Vizic Technologies and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

Vizic Tech endeavors to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development tools of Vizic products and services are continuous and published information may not be up to date. It is important to check the current position with Vizic Technologies at the web site.

All trademarks belong to their respective owners and are recognized and acknowledged.

10- Disclaimer of Warranties & Limitation of Liability:

Vizic Technologies makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall Vizic be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by Vizic, or the use or inability to use the same, even if Vizic has been advised of the possibility of such damages.

Use of Vizic devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Vizic Technologies from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Vizic Technologies intellectual property rights.

THE DATASHEETS AND SOFTWARE ARE PROVIDED "AS IS." VIZIC EXPRESSLY DISCLAIM ANY WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT.
IN NO EVENT SHALL VIZIC BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, HARM TO YOUR EQUIPMENT, COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES, ANY CLAIMS BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENCE THEREOF), ANY CLAIMS FOR INDEMNITY OR CONTRIBUTION, OR OTHER SIMILAR COSTS.

www.VIZICTECHNOLOGIES.COM