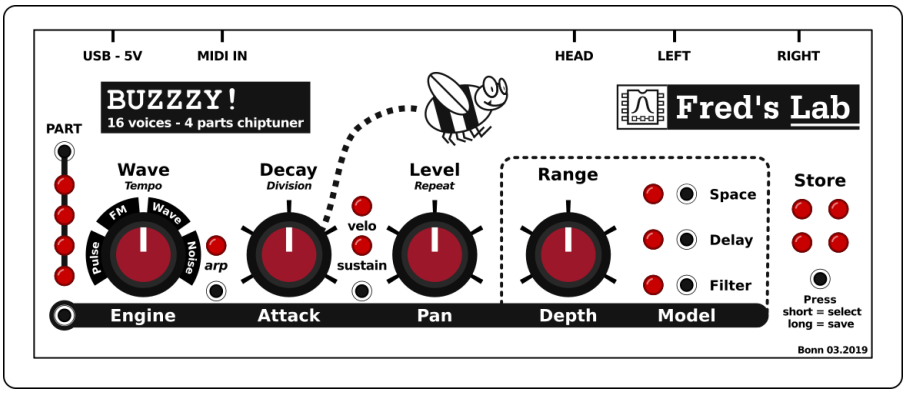
Buzzzy! : User Manual
Revision 0.1 - 18/03/2019
!NOT FOR RELEASE!



Frédéric Meslin

March 19, 2019

Introduction

Thank you for purchasing the Buzzzy!

Be sure I have put all my talent, skills and efforts to design this music instrument! Hopefully, you will like its shape, love its personality ... and come up with many great ideas and sounds for your music.

Love from Germany!

Fred, from Fred's Lab

Legal notices

Fred's Lab cannot be liable for any erroneous information contained in this manual. The contents of this manual may be updated at any time without prior notice. We put every effort to ensure the information provided here is useful and accurate. Fred's Lab extends no liabilities in regard to this manual other than those required by the local laws.

Fred's Lab - Frédéric Meslin
Herwarthstraße, 20
53115 Bonn, Germany
`info@fredslab.net`
`http://fredslab.net`

Support requests
For support requests, you can reach me per e-mail at:

`support@fredslab.net`

or per post using the address:

Fred's Lab - Frédéric Meslin
Product support
Herwarthstraße, 20
53115 Bonn, Germany

Please include the unit model, serial number and a description of the problem encountered with a maximum of details and supporting elements for a quick resolution.

Copyright information

Warranty

Fred's Lab warranty this product 2 years from its date of purchase.

This warranty covers product from unknown defects, damages & failures and the repair work needed when the product is used in normal operating conditions. However, the warranty does not cover damages due to misuse, mishandling, modifications of the product or obvious lack of care. All shipping costs involved in returning a product are at the expense of the customer.

Please read this manual carefully, especially the **Precautions** section.

Special thanks

I would like to personally thank all friends, relatives and colleagues, for their support, advises and reviews during the development of the **Buzzzy!**.

- **Fanny Homberger**
- **Jaques & Marie-Francoise Meslin**
- **Oliver Rockstedt**
- **Emilie Gillet**
- **Serge Beauchamp**
- **Benoit Ruelle**
- **Simba**

Precautions

Before plugging in the **Buzzzy!** and go **Rocking the World**, please have a sit and read this precautions through:

- Always use the device in a dry and warm environment
- Never drop the device or apply to much pressure on it
- Never spill liquid or bath the device in beer
- Never clean the device with agressive solvant
- Only use quality and certified power adapters to power the device
- Never wiggle the plugs to unconnect the cords
- Never connect the line outputs to the power outputs of an amplifier
- Never connect the headphone output to a line output of any device
- Modify the unit at your own risk!

The Buzzzy! used in conjunction with headphones and / or powerful speaker systems can produce very loud sounds in a wide range of frequencies.

Human hearing is a **very sensitive system** and can be damaged quickly and definitively. So watch out **your hears** and those of **your audience**!

Contents

1 Presentation **6**

2 Requirements **6**

3 Instrument setup **7**

4 Instrument operation **8**
4.1 Connections . . . 8
4.2 Potentiometers . . . 9
4.3 Switches & indicators . . . 10

5 The synthesis **12**

6 Engines **12**
6.1 Pulse . . . 12
6.2 FM . . . 13
6.3 Wave . . . 14
6.4 Noise . . . 15

7 Envelope **16**
7.1 Attack and Decay . . . 16
7.1.1 AD envelope . . . 16
7.2 Sustain, Velocity & Looping . . . 17
7.2.1 ASD envelope . . . 17
7.2.2 Looping envelope . . . 18
7.2.3 Velocity . . . 18

8 Mixer **19**
8.1 Volume & Pan . . . 19

9 FXs **20**
9.1 Selection & configuration . . . 20
9.2 Space FXs . . . 21
9.2.1 Fort . . . 21
9.2.2 Reservoir . . . 21
9.2.3 Abyss . . . 21
9.3 Delay FXs . . . 22
9.3.1 Mono . . . 22
9.3.2 Ping-pong . . . 22
9.3.3 Chorus / Flanger . . . 22
9.4 Filter FXs . . . 23
9.4.1 Low-pass . . . 23
9.4.2 Band-pass . . . 23
9.4.3 High-pass . . . 24
9.4.4 Distorsion . . . 24
9.4.5 Editing . . . 24

10 Arpeggiator **25**
10.1 On & Hold . . . 25

10.2 Tempo & Clocking 25
10.3 Division, Repetition & Gate length 25

11 Parts **26**
11.1 Initializing a part 26
11.2 Randomizing a part 26

12 Store **26**
12.1 Loading a multi 26
12.2 Saving a multi 26

13 Technical details **27**

14 Schematics **28**

15 MIDI implementation **32**
15.1 Channel Voice Messages 32
15.2 System Real Time Messages 33
15.3 Controller Changes / CCs 34

16 Norms **35**
16.1 Europe: CE 35
16.2 Canada 36
16.3 USA: FCC Information 36

1 Presentation

The **Buzzzy!** is a 16-voices polyphonic*, 4-parts multi-timbral, digital MIDI module.

- **Polyphonic** means it plays several notes at the same time.
- **Multi-timbral** implies it plays several instruments at the same time.
- **MIDI Module** means it needs a MIDI controller to generate notes.

What is MIDI?
MIDI stands for **Musical Instrument Digital Interface**. It is a technical standard allowing music instruments to communicate and be used in concerto. It defines a protocol, messages, a digital interface and cables / connectors formats.
A MIDI connection carries notes, controller changes, clocks and various configuration data.

The **Buzzzy!** is a digital instrument, it computes the sounds using mathematical models and algorithms, executed by a powerful ARM micro-processor.

Remarks on polyphony
The 16-voices polyphony is the **absolute maximal** number of notes that can be played at the same time. If you try to play more notes simultaneously, the older voices will be muted and immediately reused for newer notes. This mechanism is called **"voice-stealing"** and will result in audible "clicks" or parts of your music not being played. In order to avoid it, compose your music with the module limitations in mind.

The **Buzzzy!** uses a dynamic allocation mechanism for the processor resources. When using multiple audio FXs and more processing hungry voices, the polyphony decrease. Always disable the FXs and mute the voices you are not actively using to get the maximum polyphony out of the instrument.

2 Requirements

To play sounds with the **Buzzzy!** you need, at least:

- **An audio system** with a line-level input or a pair of headphones
- **A MIDI controller**, a sequencer or a computer
- **A USB power source**

An audio system can be a HiFi stereo system, a mixing desk connected to speakers or a pair of active speakers.

A MIDI controller can be any master-keyboard, wind-controller, surface ... that sends MIDI notes. You can also attach the instrument to a hardware or a software MIDI sequencer. It can run on a computer, a laptop or a tablet / smartphone. You may need a dedicated external MIDI interface or just use the USB connection.

A USB power source can be a USB host, a power bank or a USB wall adapter / charger. Never use cheap USB chargers as they can be very noisy power sources.

If you continuously hear irritating hum or noise sounds when **Buzzzy!** is connected via

USB, you may experience a **USB ground loop** issue. Don't worry, they are several solutions to overcome this problem.

- Use an external USB power source
- Connect the instrument with a MIDI cable

Or

- Use a USB galvanic isolator

MIDI DIN connections are, by design, prevented from ground loop problems. It is the preffered solution to avoid these annoying issues.

3 Instrument setup

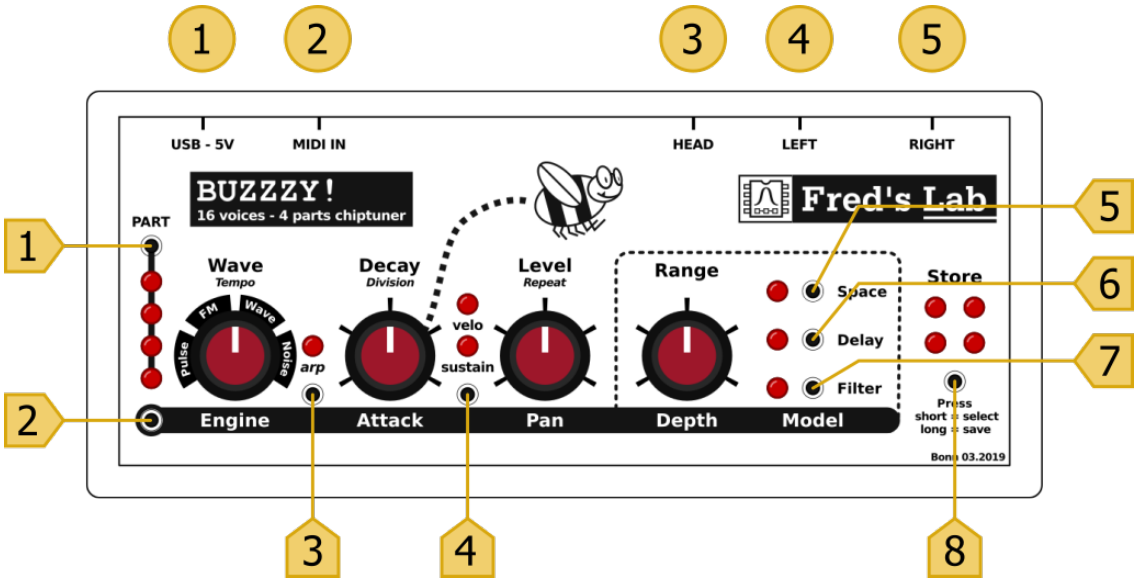
Step 1: Audio
Connect the **Buzzzy!** line-level outputs (left & right) to your audio system inputs using two 6.35mm unbalanced jacks.
Or
Connect the headphones output (head) to your headphones pair using the 3.5mm stereo jack cable. An adapter is required if your headphones have a 6.35mm jack cable.

Step 2: MIDI
Connect the **Buzzzy!** MIDI DIN input (MIDI IN) to your MIDI compatible controller or sequencer / computer using a MIDI DIN cable.
Or
Connect the USB port (USB - 5V) to your computer, laptop, tablet or smarphone.

Step 3: Power
Connect the **Buzzzy!** power input (USB - 5V) to your power source of choice.
Switch on you audio system, adjust the volume and play some notes!

4 Instrument operation

Once the instrument is properly installed and you can hear some sounds when playing notes, let's have a good look at its front panel:



4.1 Connections

(1) USB - 5V

This is the USB-B connector for both power and MIDI over USB fonction. It should be attached to a source to power up the **Buzzzy!**.

(2) MIDI IN

This is the MIDI Input DIN connector for a MIDI compatible controller or sequencer. It should be attached to a MIDI Output DIN connector of an other device.

(3) HEAD

This is the headphones output connector to attach an headphones pair. The amplifier is designed for low to medium impedance headphones. It can also be used as an additional line-level stereo audio output.

(4) (5) LEFT & RIGHT

This is the headphones output connector to attach an headphones pair. The amplifier is designed for low to medium impedance headphones. It can also be used as an additional line-level stereo audio output.

4.2 Potentiometers

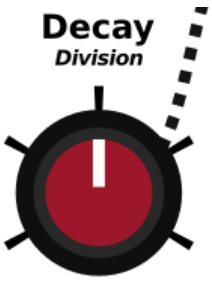
By default, turning a potentiometer adjust the *principal sound parameter*, labeled above it, with the bigger font. This parameter is only modified for the currently selected part.

To access the *secondary sound parameter*, labeled below the potentiometer, the **shift** switch must be pressed or the shift function locked.

Arpeggiator parameters, in italic font, can be adjusted by turning the potentiometer while maintaining the **Arp** switch pressed.

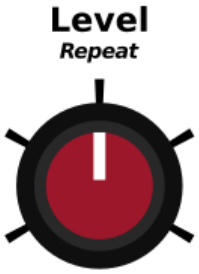
WAVE
this potentiometer
1. selects which variation of sound algorithm
2. selects which engine (Pulse, FM, Waves, Noise)
for the next note played.

With the **Arp** switch pressed, the Arpeggiator tempo can be adjusted.



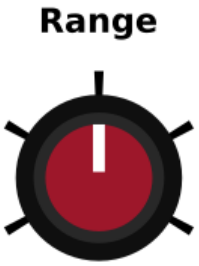
DECAY
This potentiometer
1. adjusts the envelope decay time
2. adjusts the envelope attack time
of all notes currently played.

With the **Arp** switch pressed, the Arpeggiator clock division can be adjusted.



LEVEL
This potentiometer
1. adjusts the volume
2. adjusts the panning
of all notes currently played.

With the **Arp** switch pressed, the Arpeggiator note repeat count can be adjusted.



RANGE
This potentiometer
1. adjusts the FX range parameter
2. adjusts the FX depth parameter
of the FX currently selected.

With the **Arp** switch pressed, the Arpeggiator gate length can be adjusted.

4.3 Switches & indicators

(1) PART

This is the **Part** selection switch.
By pressing it multiple times, the edited part can be chosen. The LEDs below the switch display which part is currently selected.

(2) SHIFT

This is the **Shift** switch.
Pressing it down allows access to the *secondary sound parameters*. The shift state can be **locked** by performing a double-click (two rapid presses) on the switch. The state can be **unlocked** by pressing the shift switch once more.
When the shift state is locked, the part LEDs are all blinking.

(3) ARP

This is the **Arp** enable switch.
By pressing it, the *Arpeggiator* is either enabled or disabled. The LED above the switch display the current state of the Arpeggiator.
By pressing the **Arp** switch long enough, the *Arpeggiator* **hold** function is enabled.
Keeping the **Arp** switch pressed while turning potentiometers allows the editing of the Arpeggiator specific parameters.
More information about the *Arpeggiator* can be found in the *Arpeggiator* section.

(4) SUSTAIN

This is the **Sustain** enable switch.
By pressing it multiple times, the mode of the *Envelope* is selected.
When the **Sustain** LED is lit, the *Envelope* an ASD envelope.
When the **Velocity** LED is lit, the controller transmitted **velocity** affects the volume of the notes played.
By pressing the **Sustain** switch long enough, the *Envelope* becomes a looping envelope and the **Sustain** LED blinks.
More information about the *Envelope* can be found in the *Envelope* section.

(5) SPACE

This is the space switch.
By pressing it, one of the *Space FX* is either enabled or disabled. The FX is enabled for the edited part. If a *Delay FX* was previously enabled, it is immediately disabled as *Space* and *Delay* FXs are mutually exclusive.
The *Space* LED, at the switch left, indicates the state of the FX.
Using the **Shift** function, one of the 3 available *Space FX* can be selected.

(6) DELAY

This is the delay switch.
By pressing it, one of the *Delay FX* is either enabled or disabled. The FX is enabled for the edited part. If a *Space FX* was previously enabled, it is immediately disabled as *Space* and *Delay* FXs are mutually exclusive.
The *Delay* LED, at the switch left, indicates the state of the FX.
Using the **Shift** function, one of the 3 available *Delay FX* can be selected.

(7) FILTER

This is the filter switch.
By pressing it, one of the *Filter FX* is either enabled or disabled. The FX is enabled for the edited part. The *Filter* LED, at the switch left, indicates the state of the FX.
Using the **Shift** function, one of the 3 available *Filter FX* can be selected.

(8) STORE

This is the store switch.
By pressing it shortly, the selected *multi* slot is recalled. A *multi* consist in the state of all parameters from the 4 parts.
By pressing it long enough, the current edited *multi*, is saved one the selected *multi* slot.

5 The synthesis

6 Engines

The **Buzzzy!** is equipped with 4 distinct sound engines:

Pulse, **FM**, **Wave** and **Noise**.

Each engine offer a set of sound variations.
The selected engine make the core of a note sound. Every engine offers a different timbre palette. Let's explore the different facets of these engines.

6.1 Pulse

Pulse is an engine made to produce waveforms with sharp edges. It is generates the square, the pulses, the alternative pulses (with richer bass) waves, all with or without pulse-width modulation and vibrato.

List of variations available:

N°	Name	Description
1	Square	Square wave
2	Pulse25	Pulse with 25.0% PW ratio
3	Pulse12	Pulse with 12.5% PW ratio
4	Pulse6	Pulse with 25% PW ratio
5	Alt38	Alternate pulse with 37.5% PW ratio
6	Alt25	Alternate pulse with 25.0% PW ratio
7	Alt12	Alternate pulse with 12.5% PW ratio
8	Alt6	Alternate pulse with 6.25% PW ratio
9	SquareSlow	Square with slow PW modulation
10	PulseSlow	Pulse with slow PW modulation
11	SquareFast	Square with fast PW modulation
12	PulseFast	Pulse with fast PW modulation
13	Alt50Slow	Alternate pulse 50% slow PW modulation
14	Alt25Slow	Alternate pulse 25% slow PW modulation
15	Alt50Fast	Alternate pulse 50% fast PW modulation
16	Alt25Fast	Alternate pulse 25% fast PW modulation

6.2 FM

FM is an engine made to produce rounder waveforms based on traditionnal linear phase modulation. It generates classic and distorted FM tones, using a sinus, triangle or sawtooth carrier and a sinus modulator. Vibrato is also available for this engine.

List of variations available:

N°	Name	Description
1	HalfSoft	Square wave
2	HalfMedium	Pulse with 25.0% PW ratio
3	HalfHard	Pulse with 12.5% PW ratio
4	SineSoft	Pulse with 25% PW ratio
5	SineMedium	Pulse with 25% PW ratio
6	2xSoft	Pulse with 25% PW ratio
5	3xSoft	Pulse with 25% PW ratio
6	4xSoft	Pulse with 25% PW ratio
7	HalfTriSoft	Pulse with 25% PW ratio
8	SineMedium	Pulse with 25% PW ratio
8	SineHard	Pulse with 25% PW ratio

6.3 Wave

Wave is an engine made to produce low resolution waveforms, as they would be played from a ROM, like in some early sound generators. Vibrato is also available for this engine.

List of variations available:

N°	Name	Description
1	Sinus	A regular sine wave
2	FlatSine	A slightly clipped sine wave
3	SlantedSine	A phase distorted sine wave
4	DoubleSine	Sum of two sine waves (octaves)
5	Triangle	A regular triangle wave
6	FlatTri	A slightly clipped triangle wave
7	SlantedTri	A phase distorted triangle wave
8	DoubleTri	Sum of two triangle waves (octaves)
9	Smooth	A smoothed random wave
10	Random	A harsh random wave
11	SawSoft	A soft fake saw wave
12	SawHard	A hard fake saw wave
13	Sin2Tri	A wave evolving from sine to triangle
14	Tri2DSin	A wave evolving from triangle to double sine
15	Smo2Rnd	A wave evolving from smooth to random wave
16	Soft2Hard	A wave evolving from soft to hard wave

6.4 Noise

Noise is an engine made to produce noises with different kind of filtering (low pass and band pass), as well as at different sampling rates. Pitched noises are also available.

List of variations available:

N°	Name	Description
1	NoiseLP	Low passed colored noise
2	NoiseBP	Band passed colored noise
3	HalfLP	Under-sampled low passed noise
4	HalfBP	Under-sampled band passed noise
5	CrapLP	Crappy low passed noise
6	CrapBP	Crappy band passed noise
7	Pitch1	Pitched noise 1
8	Pitch2	Pitched noise 2

7 Envelope

The *envelope* controls the dynamic level of a note. It also acts on several other sound parameters (depending on selected engine and variation). The **Buzzzy!** offers one envelope per voice. While this is limited, the envelopes can work in different modes.

7.1 Attack and Decay

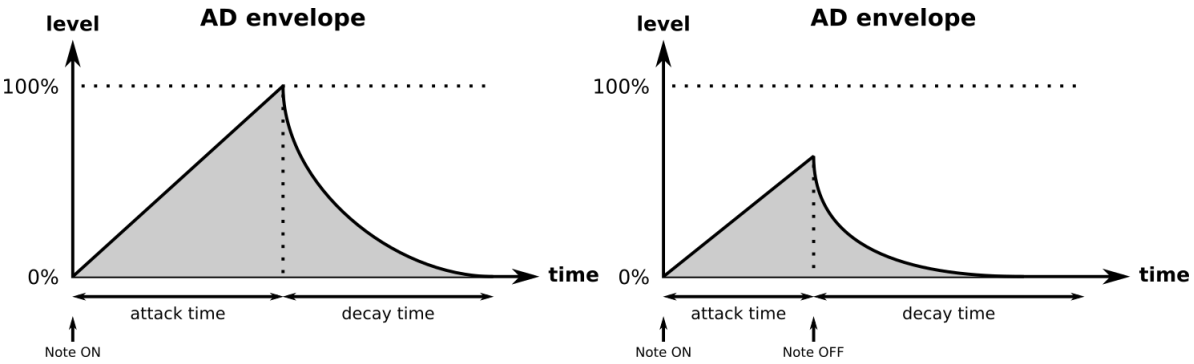
The basic attributes for a simple envelope are its **attack time** and **decay time**.

The **attack time** defines the time taken by the volume of a note to rise from minimum to its maximum.

The **decay time** defines the time taken by the volume of a note to fall from its release level to the minimum.

The times specified range from very short (< 1ms) to extended periods (18s).

7.1.1 AD envelope



This is the default behavior of the *envelope*.

When a note is triggered, its associated voice AD *envelope* rise from the minimum to the maximum level. When the maximum is reached, the *envelope* immediately falls from the maximum back to the minimum level. After this cycle, the note is considered to be terminated and its related voice is consequently released.

The *envelope* can be interrupted in its **attack phase**, if the note is released before the maximum level is reached. In this situation, the *envelope* will fall from its release level to the minimum, at the specified **decay** time.

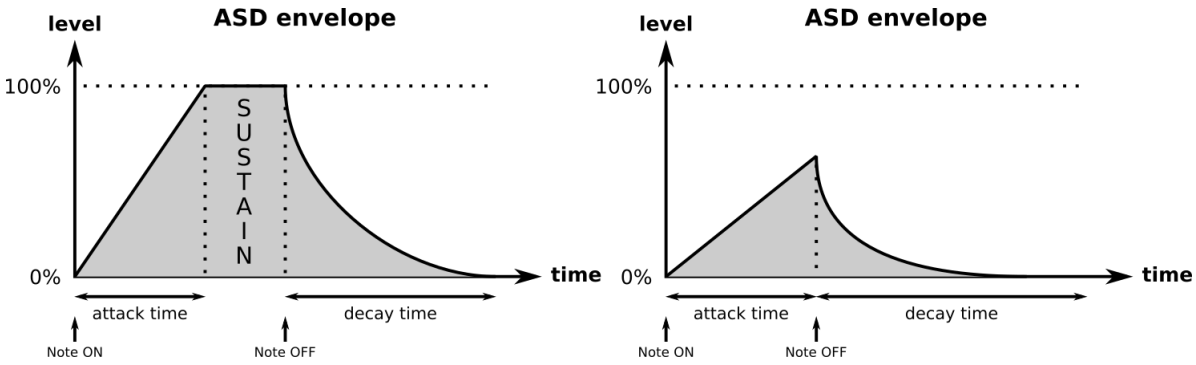
Remark
AD *envelopes* are very useful for percussive sounds, short melodic bleeps and plucks but also for arpeggiated patterns.

7.2 Sustain, Velocity & Looping

For longer notes, an additionnal **sustain** stage is often required.
By pressing the **sustain** siwtch several times, the ASD mode can be enabled or disabled.
To select the *envelope* **looping** mode, press the **sustain** switch long enough.

The **sustain** and **velocity** LEDs, located above the switch, indicates the current *envelope* mode and its sensitivity to **velocity**. When the **sustain** LED blinks, the *envelope* is in looping mode.

7.2.1 ASD envelope



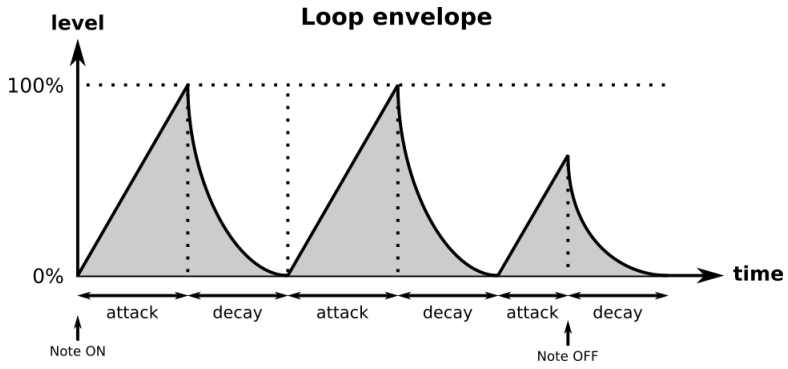
This mode is similar to the AD *envelope*, excepting it has a **sustain phase**.

When a note is triggered, its ASD *envelope* rise from the minimum to the maximum level. When the maximum is reached, the *envelope* **stays at this level** until the note is released. When this happens, the *envelope* level falls from the maximum back to the minimum level. After this cycle, the note is terminated and its related voice is released.

As in AD mode, the ASD *envelope* can also be interrupted in its **attack phase**. The ASD behavior is then strictly identical to the AD *envelope* one.

Remark
ASD *envelopes* are perfect for chords or drone sounds.

7.2.2 Looping envelope



This mode is similar to the AD *envelope*, excepting the *envelope* immediately restarts when it reaches the minimum level.

When a note is triggered, the looping *envelope* rise from the minimum to the maximum level. When the maximum is reached, the *envelope* immediately falls from the maximum back to the minimum level. When the minimum level is reached, the cycle restarts. This loop repeats until the note is released. When this happens, the *envelope* level falls from the release level to the minimum level and no longer loops. The note is then terminated and its related voice is released.

Remark
Cycling *envelopes* have great use for sound effects, percussion noises and pulsating chords.

7.2.3 Velocity

If the *envelope* **velocity** is enabled, notes velocity information sent by the MIDI controller or the sequencer to the **Buzzzy!** affects the volume of the notes played on the edited part. This gives more expressivity to an instrument by letting the musician control the notes loudness individually.

Remark
Velocity variations come in handy to accentuate or de-emphasize elements in a rythmic pattern. When playing complex chords, different **velocity** levels can help to finely distribute the predominance of the chord degrees or notes.

8 Mixer

The *mixer* sets the **level** and the **panning** of a part. It is meant to adjust the music instrument loudnesses and positions in the stereo space. The mixer uses a sinus / cosinus panning law.

8.1 Volume & Pan

Only two parameters are available: **level** and **pan**.
To adjust the **level**, turn the **Level** potentiometer.
To adjust the **pan**, maintain the **shift** switch pressed (or **lock** the **shift** state) and turn the **Level** potentiometer.

Remark
To avoid sound distorsion or even hard-clipping when mixing together several parts or instruments playing a lot of notes, try to leave some safe **level** margin.

The **Buzzzy!** has no **master level** control. The overall volume is the result of the sum of each part sound volume.

9 FXs

The **Buzzzy!** incorporates a wide selection of quality sound effects, with several **delays**, **reverbs** and **filters**. These effects can be chained and set differently on every part.

Important
Sound effects require **processing power**. Some effects as *filters* or *delays* not that much, some others like *flanger* and *reverbs* much more. To maximize the polyphony, disable the unused effects and limit the number of *reverbs* used simultaneously. Otherwise, the number of voices available will be very limited.

Both *Delays FXs* and *Space FXs* (special *reverbs*) require a significant amount of memory. This is the reason why *Delays FXs* and *Space FXs* are exclusive. Put differently, a *delay* and a *space* cannot be chained on a *part*.

9.1 Selection & configuration

By pressing one of the 3 **effect** switches, the respective *effect* is enabled or disabled on the edited *part*. If the *effect* enabled is one of the *Space FXs* and there was a *Delay FX* already engaged, the later is disabled and vice versa. A *Filter FX* can always be added to a time time-domain *effect*.

Effects have four parameters: **range**, **depth**, **model**, **mix** (not for *filters*).

Their exact influence of these paramters on the sound depends on the combinaison of effect and model selected. These parameters will be detailed in the following sections.

Selecting a model
Maintaining the **shift** switch pressed (or having the **shift** state **locked**) while pressing any **effect** switch changes the related *effect* **model**. Models can be selected whether the *effect* is enabled or not.

Adjusting the mix
The **mix** parameter sets the amount of effect "wet" signal blent with the original "dry" signal. *Filters* don't have a **mix** parameter. Maintaining the desired **effect** switch pressed while turning the **Range** potentiometer modifies the **mix** parameter.

Let's now have a look at all the various *effects*.

9.2 Space FXs

Space FXs belong to the **reverberation** type of effects. These are time-domain effects. They are made to simulate the influence of a certain "space" or "room" on the sound. Here, they are deliberately not called *reverbs*, as this term refers to a specific category of effects, the algorithmic **reverberations**.

Models used in the **Buzzzy!** are very simplified and were designed empirically. The goal was to obtain the best sound character and musicality using a quite limited computing power.

9.2.1 Fort

Fort is a large empty room with wall / ceiling / floor kind of reflections model. The reflections are relatively dry, bright and little modulated. There is no tail sound with this algorithm. The effect produces stereo filtered reflections.

The **decay** of the reflections is set with the **decay** (= **Range pot.**) parameter and the room **size** using the **size** (= **Depth pot.**) parameter.

9.2.2 Reservoir

Reservoir is very similar to *Fort*. The difference is the modulation of the reflections. The room size is fixed but the speed and amount of the modulations are both adjustable.

The **decay** of the reflections is set with the **decay** (= **Range pot.**) parameter and the **modulation** speed & amount using the **modulation** (= **Depth pot.**) parameter.

9.2.3 Abyss

Abyss share some similarities to large *hall* **reverb** algorithms. The focus of this model is on the sound tail quality and length.

The **decay** of the tail is set with the **decay** (= **Range pot.**) parameter and the room **size** using the **size** (= **Depth pot.**) parameter.

Remark
Abyss is the most demanding effect model for computing power ... and by far!
Use it sporadically to preserve the instrument polyphony.

9.3 Delay FXs

Delays FXs belong to the **echo / delay** type of effects. These are also time-domain effects and they are based on the repetition of the original signal at different intervals, with playback modulations and filtering.

Remark
In the **Buzzzy!** sound engine, the **delay lengths** are tuned. They respect an exponential law and therefore the *Delays FX* can also be used as resonators or comb filters.

9.3.1 Mono

The *mono* delay repeats the original signal a defined number of times. The repetitions are progressively degraded, they lose their loudness, dynamic precision and some of their frequency content.

The frequency of the repetitions is set with the **length** (= **Range pot.**) parameter and the number of the repetitions using the **feedback** (= **Depth pot.**) parameter.

Feedback parameter extend to infinite repeat, allowing drones and ambient lushy sounds.

9.3.2 Ping-pong

The *ping-pong* delay is very similar to the *mono* delay. The essential difference is the repetitions come first from the left channel and then come from the right channel and so on. It gives a nice stereo image to the sound.

The frequency of the repetitions is set with the **length** (= **Range pot.**) parameter and the number of the repetitions using the **feedback** (= **Depth pot.**) parameter.

Feedback parameter also extend to infinite repeat.

9.3.3 Chorus / Flanger

Although this effect does not feel like a delay, it uses the same ground principes. The repetitions are closer and modulated, the perceived effect is a constant variation of the sound spectrum giving it a thicker character. This stereo effect has multiple uses, try to experiment with different settings!

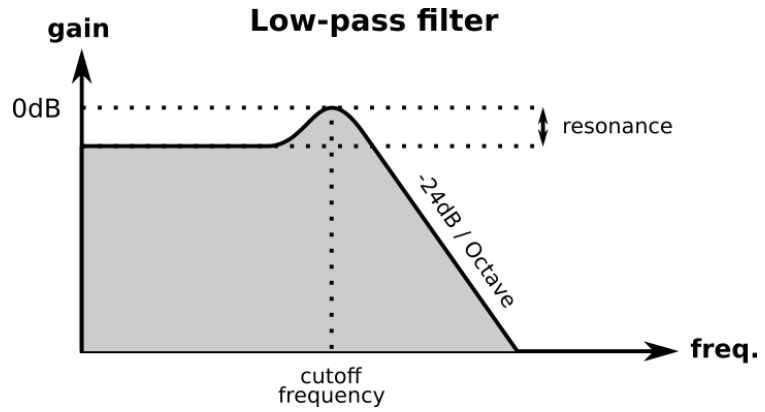
The modulation rate is set with the **speed** (= **Range pot.**) parameter and the amount of feedback using the **feedback** (= **Depth pot.**) parameter.

9.4 Filter FXs

The *filter* is an effect shaping the frequency content (audio spectrum) of the sound of a part. It can attenuate or amplify full bands of frequencies. The *filter* can always be activated and is applied prior to any other effect (*Space FX* or a *Delay FX*).

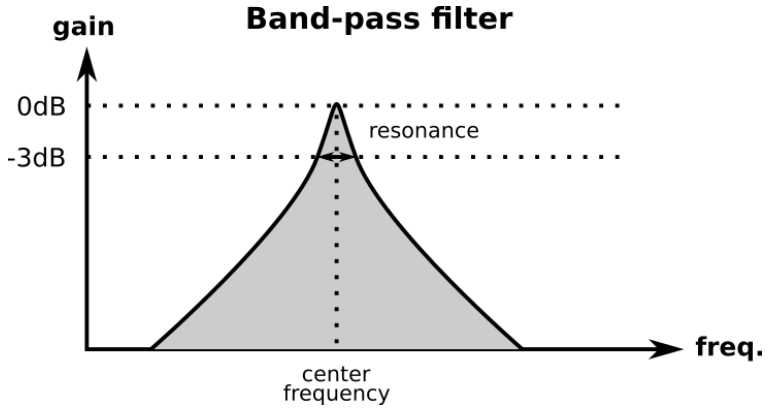
The *filter* exists in 3 modes: **Low-pass** (LP), **Band-pass** (BP) and **High-pass** (HP). The attenuation slope / filter order is fixed and is 24dB per octave or 4th order.

9.4.1 Low-pass

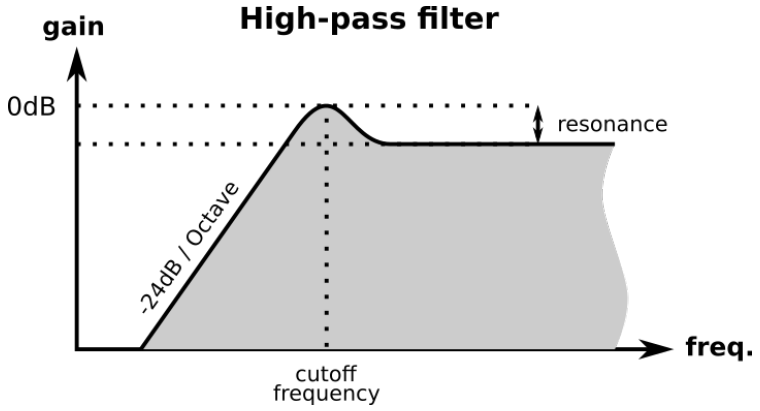


The **low-pass** *filter* only keeps frequencies below its **cutoff frequency**. The frequencies above this limit are progressively attenuated. **Resonance** parameter creates an adjustable "bump" that accentuates the frequencies right before the **cutoff frequency**.

9.4.2 Band-pass



The **band-pass** *filter* keeps a band of frequencies around its **center frequency**. The frequencies outside of this band are progressively attenuated. **Resonance** parameter adjust the size of the band (or the filter **selectivity**) around the **center frequency**.



9.4.3 High-pass

The **high-pass** *filter* only keeps frequencies above its **cutoff frequency**. The frequencies below this limit are progressively attenuated. **Resonance** parameter creates an adjustable "bump" that accentuates the frequencies right after the **cutoff frequency**.

9.4.4 Distorsion

The filters are coupled with a *soft-clippling distorsion* which adds extra character to the sound and round off the harsh resonance peaks. To adjust the *distorsion* effect, set the *part* volume to an appropriate level.

9.4.5 Editing

To adjust the *filter* **cutoff frequency** or **center frequency**, turn the **Range** potentiometer, while the *Filter FX* is activated.
To adjust the *filter* **resonance** or **selectivity**, maintain the **shift** switch pressed (or **lock** the **shift** state) and turn the **Range** potentiometer, while the *Filter FX* is activated.

Remark
For editing, the *filter*, when actived, always has the priority. In other words, you cannot edit the parameters from a *Space FX* or a *Delay FX* while the *Filter FX* is enabled. This is a limitation of the device.

10 Arpeggiator

The *Arpeggiator* is a handy function allowing the creation of complete melodic patterns from chords or single notes. **Buzzzy!** *Arpeggiator* always plays notes respecting the order (in time) in which they have been entered. Each part has its own *Arpeggiator* that can be used independently.

10.1 On & Hold

To enable the *Arpeggiator* on the edited part, press the **Arp** switch. The **Arp** LED located above indicates the state (enable or disable) the *Arpeggiator* is.

10.2 Tempo & Clocking

There are two ways to set the base **tempo** for the *Arpeggiator*:

- **Free-running**, the **tempo** is set manually, using the tempo parameter.
- **Synchronized**, the **tempo** is set automatically, with MIDI clock ticks (USB or DIN).

The **Buzzzy!** automatically switch to **synchronized** mode if an external clock is received and revert back to the **free running** mode when no other clock source is available.

Tempo (for **free-running** mode only) is set by holding the **Arp** switch down while turning the **Wave** potentiometer.

10.3 Division, Repetition & Gate length

In addition to the **tempo**, the step length of the notes sequence can be set using the **division** parameter. The **division** rate ranges from 1/4 (corresponding to a step length of a sixteenth note) to 2 (corresponding to a step length of a half note) by increments of 1/4.

These are the available divisions:

N°	Ratio	Note
1	0.25	A sixteenth note
2	0.50	An eighth note
3	0.75	A dotted eighth note
4	1.00	A quarter note
5	1.25	...
6	1.50	A dotted quarter note
7	1.75	A double-dotted quarter note
8	2.00	A half note

Division is set by holding the **Arp** switch down while turning the **Decay** potentiometer.

The *Arpeggiator* offers to repeat the notes of the sequence. The **repeat** parameter ranges from 1 (no repeat) to 16 times.

Repetition is set by holding the **Arp** switch down while turning the **Level** potentiometer.

The notes duration can also be adjusted using the **gate length** parameter. The duration ranges from 0% to 100% of the selected step length.

Gate length is set by holding the **Arp** switch down while turning the **Range** potentiometer.

11 Parts

A *part* is associated with an instrument and corresponds to a dedicated MIDI channel. To simplify the configuration, **part1** is mapped to MIDI channel 1, **part2** is mapped to MIDI channel 2 and so on.
To play a note from a specific *part*, the MIDI controller or software sequencer must send its MIDI note events to the appropriate MIDI channel.

The **part** switch allows to select which *part* is currently edited using the **Buzzzy!** interface. The **part** LEDs below display which *part* is currently selected.

11.1 Initializing a part

To initialize a *part* to its default configuration, select the desired *part* using the **part** switch and then press long enough this very same **part** switch. All sound parameters from this edited *part* will be set back to their default value.

11.2 Randomizing a part

To randomize a *part* configuration, select the desired *part* using the **part** switch and then, while the **shift** button is pressed (or the **shift** state is **locked**), press long enough the **part** switch. All sound parameters from this edited *part* will be set to random values.

12 Store

The store is used as the user configuration memory and allows to save up to 4 *multis* for a later recall. A *multi* consist in the state of all parameters from the 4 parts. The *multis* are stored in the machine internal **Flash** memory. Therefore, the configurations are preserved even when the machine is powered off.

Important
Never switch the machine off while saving a *multi*!
This can result in the corruption of the **Flash** memory and all user data stored will be definitely lost!

12.1 Loading a multi

Press the **store** switch multiple times to load the desired *multi*. Be careful, all unsaved edited parameters will be discarded without notification! The store **store** LEDs above the switch indicate which *multi* has been recalled.

12.2 Saving a multi

Press the **store** switch long enough and all the configurations of the 4 parts edited *multi* will be saved in the selected *multi* slot.

13 Technical details

Here you will find all technical data concerning the **Buzzzy!**.

• **Number of parts & maximal polyphony:**
4 parts / 16 voices

• **Number of effects:**
3 Space effects: Fort, Reservoir & Abyss
3 Delay effects: Echo, Ping-pong & Flanger
3 Filter effects: LP, BP & HP, 24dB / Octave

• **User Memory:**
4 multi slots

• **Device dimensions & weight:**
Unit 148 (width) x 70 (depth) x 68 (height) mm
Unit 200 grams (approximately)

• **USB classification:**
USB 1.1 fullspeed (12 Mbit/s)
USB MIDI Class-Compliant device

• **Power requirements:**
+5V +/- 5%, 75mA (maximum)
375mW (maximum)

• **Line output characteristics:**
+8.5dBu / 2.1V_{RMS} / 5.8Vpkpk peak level
Impedance DC 470Ω, AC 0Ω (f < 24kHz)

• **Headphones output chracteristics:**
90mW average / 16Ω headphones
64mW average / 32Ω headphones
Impedance DC / AC 10Ω

• **Connectors:**
USB type B socket
MIDI Input DIN5 socket
Headphones 3.5mm female stereo jack socket
Line outputs 2x 6.35mm female jack sockets

• **Master clocks:**
MCU @ 81.92MHz, USB @ 48MHz, I2S @ 12.288MHz

• **Audio resolution:**
48.00kHz, 24Bit stereo audio
100dB SNR & dynamic range

14 Schematics

15 MIDI implementation

15.1 Channel Voice Messages

Channel Voice Messages are used to send basic music events to one specific channel or part. They consists in 3 bytes: the **status** byte, followed by two **data** bytes. Here follows the list of supported messages.

Note Off

Stops the playback of a note.

```
Byte: 0     1     2
Data: 0x8c  note  velo
      c    = MIDI channel (0 - 15)
      note = MIDI note (0 - 127)
      velo = note velocity (0 - 127)
```

Note On

Starts the playback of a note.

```
Byte: 0     1     2
Data: 0x9c  note  velo
      c    = MIDI channel (0 - 15)
      note = MIDI note (0 - 127)
      velo = note velocity (1 - 127)*
* Note: a velocity of 0 corresponds to a Note off event.
```

Channel Aftertouch

Controls the tremolo level of the given channel (= part) notes.

```
Byte: 0     1     2
Data: 0xDc  least most
      c     = MIDI channel (0 - 15)
      least = least significant 7bit of 14bit bend value*
      most  = most significant 7bit of 14bit bend value*
* Note: bend value is centered around 0x4000 = 0 cents.
```

Pitch Bend

Increases or decreases the pitch of the given channel (= part) notes.

```
Byte: 0     1     2
Data: 0xEc  least most
      c     = MIDI channel (0 - 15)
      least = least significant 7bit of 14bit bend value*
      most  = most significant 7bit of 14bit bend value*
* Note: bend value is centered around 0x4000 = 0 cents.
```

15.2 System Real Time Messages

System Real Time Messages are used to send global and high-priority messages relative to timing and sequences.

Clock start
Start all parts arpeggiators if enabled.

```
Byte: 0
Data: 0xFA
```

Clock continue
Re-start all parts arpeggiators if enabled.

```
Byte: 0
Data: 0xFB
```

Clock stop
Stop all parts arpeggiator.

```
Byte: 0
Data: 0xFC
```

Clock tick
Synchronizes the MIDI master clock of the sound module.
A tick corresponds to 1/24th of a quarter note.

```
Byte: 0
Data: 0xF8
```

Active sensing
Ensure the MIDI connection is working.

Once first sent, **active sensing** messages must be sentto the module periodically. If the module stops receiving **active sensing** messages for an extended period of time, it will automatically perform a (all sounds off) action. The module will immediately stop emitting sound.

```
Byte: 0
Data: 0xFE
```

System Reset
Reset the sound module.

```
Byte: 0
Data: 0xFF
```

15.3 Controller Changes / CCs

Controller Changes Messages are used to alterate sound parameters of a given channel. They allow more musical expressivity by adjusting the sound character while playing or allow the editing of the part parameters from an extended / remote MIDI interface or a computer software (sequencer or preset editor).

The messages consist in 3 bytes: the **status** byte, followed by the **control number** byte and then terminated by the **control value** byte. Here follows the list of supported messages.

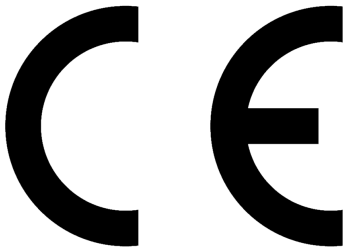
Controller Change
Modifies a specific sound parameter value of the given channel (= part).

```
Byte: 0     1     2
Data: 0xBc  least most
      c     = MIDI channel (0 - 15)
      number = controller number (see the list)
      value  = controller value* (see the list)
* Note: internal parameter values are represented with 10bit.
        Since MIDI CCs can only transmit 7bit values, values transmitted
        are interpolated and smoothed in time.
```

Parameter	CC(Dec)	CC(Hex)	Description
Modwheel	1	0x01	Modulation wheel (vibrato level)
Volume	7	0x07	Part mixer level
Pan	10	0x0A	Part mixer pan
Tempo	16	0x10	Part arpeggiator tempo (when not clocked)
Div	17	0x11	Part arpeggiator division
Repeat	18	0x12	Part arpeggiator repeat
Gate	19	0x13	Part arpeggiator gate length
Engine	70	0x46	Part oscillator engine
Wave	71	0x47	Part oscillator wave
Decay	72	0x48	Part envelope decay time
Attack	73	0x49	Part envelope attack time
Cutoff	74	0x4A	Part filter cutoff / center frequency
Resonance	75	0x4B	Part filter resonance
Space model	85	0x55	Part space effect model
Delay model	86	0x56	Part delay effect model
Filter type	87	0x57	Part filter effect type
Space mix	89	0x59	Part space effect mix level
Delay mix	90	0x5A	Part delay effect mix level
Space decay	91	0x5B	Part space effect decay time
Space diffuse	92	0x5C	Part space effect diffusion factor
Delay feedback	93	0x5D	Part delay effect feedback level
Delay time	94	0x5E	Part delay effect delay time
All sounds off	120	0x78	Mute immediately all sound from the part
Reset all controllers	121	0x79	Set all controllers to their initial value
All notes off	123	0x7B	Release immediately all notes from the part

16 Norms

16.1 Europe: CE



EC DECLARATION OF CONFORMITY

1. Product unique identification:
   Buzzzy! digital sound module

2. Address of the manufacturer and his authorised representative:
   Fred's Lab - Frédéric Meslin
   Herwarthstraße, 20
   53115 Bonn, Germany

3. This declaration of conformity is issued under the sole responsibility of:
   Fred's Lab - Frédéric Meslin

4. Object of the declaration:
   This equipment conforms to the requirements:

5. Signed for and on behalf of Fred's Lab
   Bonn (Germany), the 16/03/2019

   Frédéric Meslin,
   Founder and Lead engineer of **Fred's Lab**

16.2 Canada

This device does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the radio interference regulation of the Canadian Department of Communications.

Cet équipement n'émet pas de bruits radiofréquence dépassant les limites applicables aux appareils numériques de la Classe B prescrites dans le règlement sur les interférences radio-électriques édicté par le Ministère Des Communications du Canada.

16.3 USA: FCC Information

This equipment has been verified to comply with the limits for a class B computing device, pursuant to FCC Rules. In order to maintain compliance with FCC regulations, shielded cables must be used with this equipment. Operation with non-approved equipment or unshielded cables is likely to result in interference to radio and TV reception.

Important: Changes and modifications made to the equipment without the approval of the manufacturer can void your authority to operate this equipment.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.

However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help