

# LaVoixski the Digital Theremin



LaVoixski is a digital musical instrument based on the Theremin. It has the RF oscillation circuits "The heart of the Theremin", but does not have the analog demodulators for produce the audio signals.

The system is programmed by Arduino and is running on the Micro Controller Unit Teensy4.1.

LaVoixski has 5 wavetable oscillators. The 32 wavetables are contained as the text file on the microSD card. The player can modify them by using the math tool like "GNU Octave".

The pitch and the volume values are sensed from the real RF oscillators by "General Purpose Timer = GPT" inputs on the Micro Controller. The detected values control the pitch of wavetable oscillators and its envelopes.

LaVoixski has the several method for mixing the oscillators.

The first method is named monaural / additive synthesis, the oscillators are driven in the same pitch. The player can select any fundamental and harmonic waves for design the total shape of the waveform.

The second method is polyphonic output. The oscillators are driven in the different pitches and any waveforms. The player can select the pitch combinations from the preset data on the microSD or edit the microtonal pitches in The Chord Edit mode.

The third method is the Transition. The mixed audio output has 2 modes, "Normal " and "Transition". In the Transition mode, the output signal levels of the oscillators are controlled by the envelopes in the different phase. (see the section "Transition")

The system has 3 edit modes: Chord, WaveMix and Exciter.

In the Chord Edit mode, the player can edit 4 microtonal pitches in a chord. The 16 edited pitch groups are stored by CSV formatted text files into the microSD card. (see the section "Microtonal tuning / Chord Edit Mode")

In the WaveMix mode, the player can edit the combination of the 5 wavetables with levels. The edited combinations are also stored into the microSD card. (see the section "WaveMix Mode")

**The parameter named "Output 3" selects a signal line from any oscillators for the 3rd audio output. There are 3 modes for the outputs, Transition, Normal and Chebyshev Transfer = Exciter mode.**

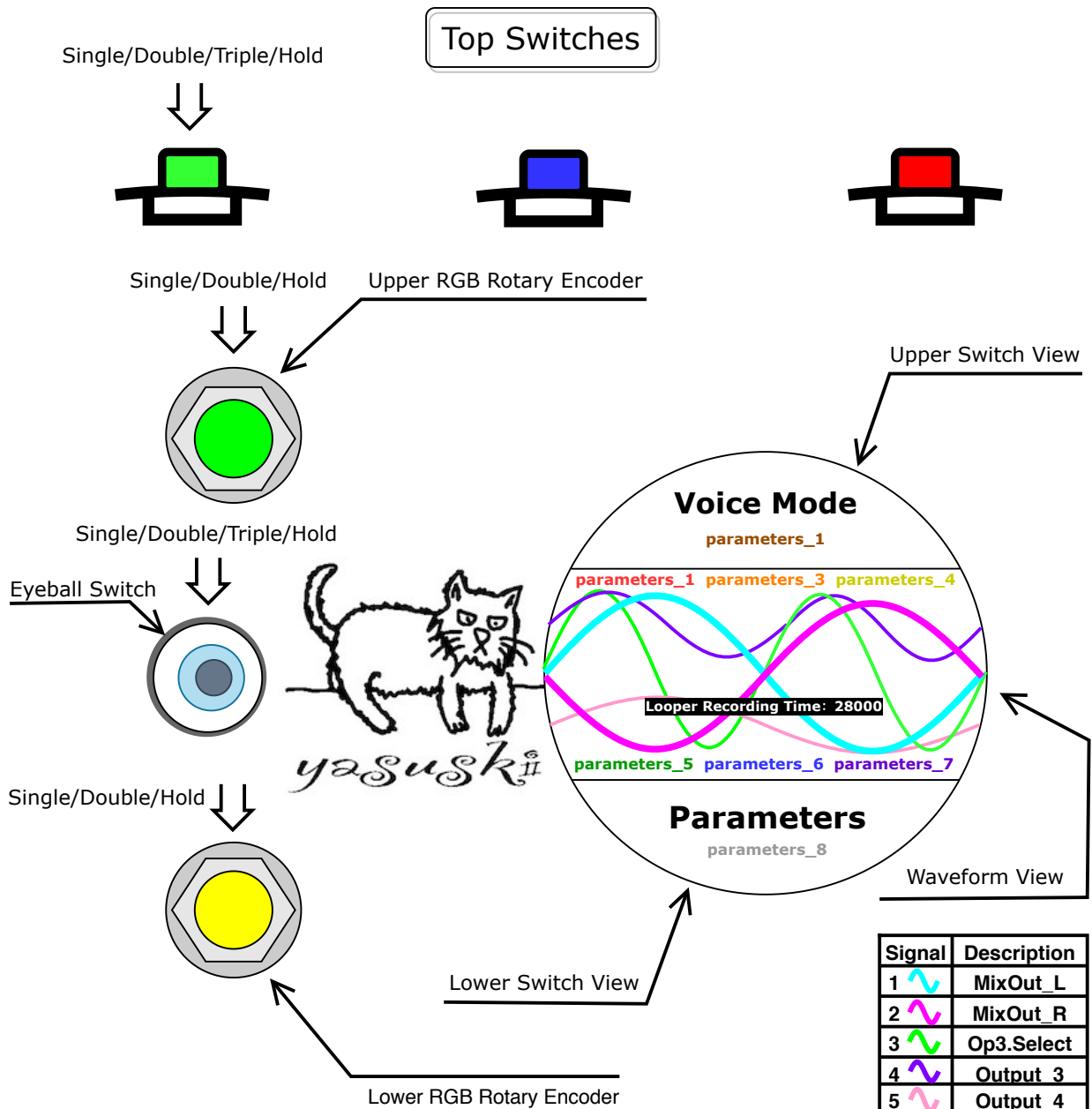
**The Chebyshev Transfer mode is specially developed for the Output 3. The Exciter transforms the balance of the harmonics of the input source by using the wavetable for the transformation.**

**The level editable parameters (fundamental plus 4 harmonics) are available. (see the section "Audio Excitation")**

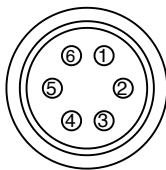
## The description about the system

|                        |                                                                    |                                                                                             |
|------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| Oscillator             | 5 individual Oscillators                                           | Reading the waveforms from the files on microSD card                                        |
| Waveforms              | 32 wavetables                                                      | Written in the 12 * 12 bit CSV format on microSD card                                       |
| Voicing                | 2 / 3 / 4 / 5 voice preset and Micro Tonal (user editable)         | The every mode has the Arpeggiator on the last address#                                     |
| Arpeggiator            | 19 preset pattern with 13 user editable notes (limited in 4 notes) | Up / Down / Alternate / Random pattern selector with Speed                                  |
| Sequencer              | 8 sequence                                                         | the 1st addressed sequencer controls the Arpeggiator                                        |
| Waveform Mix           | 14 preset with 18 editable waveform mix                            | Fixed and Transition mode is selectable The last 4 address# are tuned for the overload mode |
| PitchBend / Phaser     | Bend Up / Down / Phase Shifter                                     | Phaser is activated when the parameter AtkEnv is set to zero                                |
| LFO / AM Modulation    | *1 / *2 and High speed                                             | The depth control switches the speed rate across the value zero                             |
| Chromatic Tuner        | Chromatic or programed scale                                       | 32 scale presets                                                                            |
| VCF/LPF/HPF            | 5 VCFs on the Oscillator outputs                                   | with LFO / Envelope modulation                                                              |
| Delay                  | 6 delays on the audio lines                                        | The feedback loop accept the AM modulation                                                  |
| Comb Filter            | Flanger / Chorus / Dub selectable                                  | LFO rate / center frequency / signal feedback                                               |
| Looper                 | 30 sec Looper with rev speed                                       | *1 / *2 /* 0.5 with reverse direction                                                       |
| Audio Outputs          | L/R mix outputs with individual 3rd / 4th outputs                  | The 3rd output is selectable The 4th is fixed to OSC3                                       |
| Output Three selection | solo selection from the oscillator                                 | 3 modes are selectable Transition / Normal / Chebyshev Transfer                             |
| Chord Sequencer        | set the chord sequence in 16 address from 48 patterns              | 16 Microtonal (editable on local) +16 from microSD (editable) 16 preset patterns            |

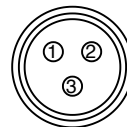
# The User Interface



| Signal | Description |
|--------|-------------|
| 1      | MixOut_L    |
| 2      | MixOut_R    |
| 3      | Op3.Select  |
| 4      | Output_3    |
| 5      | Output_4    |

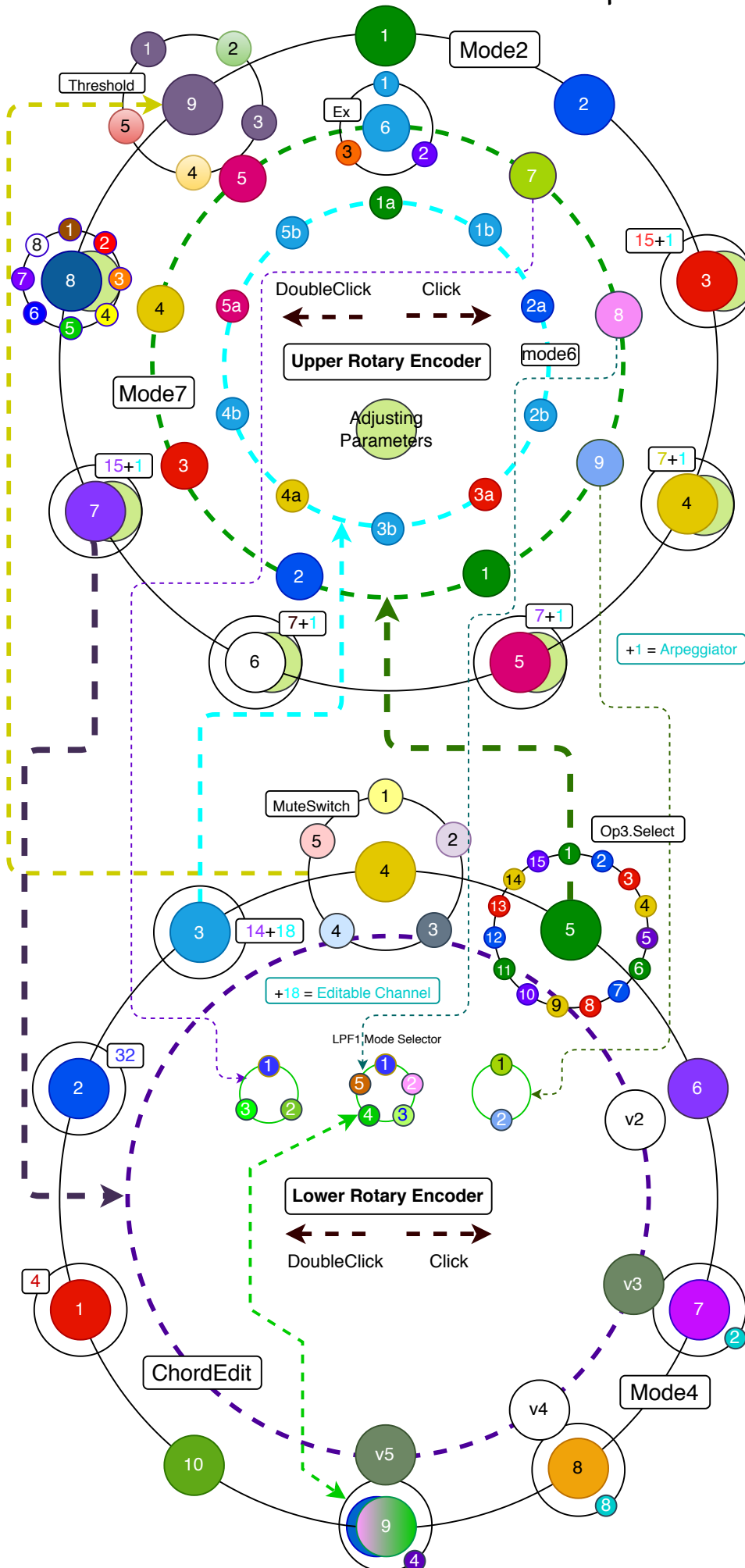


| pin# | Description |
|------|-------------|
| 1    | V+          |
| 2    | GND         |
| 3    | DAC_OUT_4   |
| 4    | DAC_OUT_3   |
| 5    | DAC_OUT_1   |
| 6    | DAC_OUT_2   |



| pin# | Description |
|------|-------------|
| 1    | Ext.SW_#01  |
| 2    | Ext.SW_#02  |
| 3    | GND         |

# Encoder Map



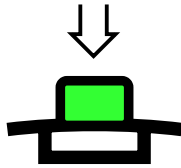
| Click                    | Long Push         |
|--------------------------|-------------------|
| 1 Pitch Tune             | Pitch Offset      |
| 2 Volume Tune            | Volume Offset     |
| 3 2 Voice Mode           | Scene Mem.2       |
| 4 3 Voice Mode           | Scene Mem.3       |
| 5 4 Voice Mode           | Scene Mem.4       |
| 6 5 Voice Mode           | Scene Mem.5       |
| 7 Chord Edit Mode        | #57 Scene /       |
| 8 Sequencer Mode         | #42-56 Chord Mem. |
| 9 Threshold Setting Mode | Scene Memory      |

|   |                                  |
|---|----------------------------------|
| 6 | Chebyshev Transfer Mem. Selector |
| 7 | Dly Fdbk / DlyLevel / CombFdbk   |
| 8 | LPF1 CutOffFreq. Fix / Mod.Spd   |
| 9 | LPF Mod Wave shape /             |
| 1 | Chebyshev Transfer 1st order     |
| 2 | Chebyshev Transfer 2nd order     |
| 3 | Chebyshev Transfer 3rd order     |
| 4 | Chebyshev Transfer 4th order     |
| 5 | Chebyshev Transfer 5th order     |

|    |                                       |
|----|---------------------------------------|
| 1* | Detector / Looper Output Selector     |
| 2* | Looper Playback Speed Selector        |
| 3* | Setting Rit Value / Pitch Drift Value |
| 4* | Starting Address for Chord Selector   |
| 5* | Setting Delay Scale                   |

|                   |                                          |
|-------------------|------------------------------------------|
| 4                 | Output Mode Select                       |
| 1*                | Normal Out                               |
| 2*                | Phase Shift Out                          |
| 3*                | Mute                                     |
| 4*                | LFO Out                                  |
| 5*                | Chromatic Out                            |
| 5                 |                                          |
| 1 2 3 4 5         | Op3 Selector Normal                      |
| 6 7 8 9 10        | Op3 Selector Transition                  |
| 11 12 13 14 15    | Op3 Selector Chebyshev Out               |
| 1 2 3             | Delay Param Selector Fdbk / Lev /        |
| 4 5               | CombFdbk                                 |
| 6 7 8 9           | LPF1 Mode Selector                       |
| 10 11 12 13 14 15 | Fix / Mod / Flanger / Chorus / Dub       |
| 1 2               | Modulation Waveform Selector 1/2         |
| 6                 | Transition Peak Distance                 |
| 7                 | Transition Peak Shape                    |
| 8                 | LPF : HPF : Distortion : USB :           |
| 9                 | Data Inputs Selector                     |
| 10                | LPF1 Mode Selector (Lower LED color)     |
| 11                | Mod Ctr(1) / Spd(2,3) / Dpt(4) / Fdbk(5) |
| 12                | Sequence Speed (When mode2 = 8)          |
| 10                | Arpeggiator Speed                        |
| 1                 | Arpeggiator Pattern Select               |
| 2                 | Arpeggiator Phrase Select                |
| 3                 | Waveform Mix Mode                        |
| 4                 | Long Push -> Mem.Waveform Mix            |
| v2                | Tuning 2nd voice                         |
| v3                | Tuning 3rd voice                         |
| v4                | Tuning 4th voice                         |
| v5                | Tuning 5th voice                         |

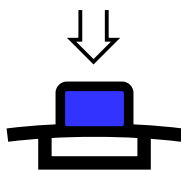
# The Switches



The top switch on the left is working as follows:

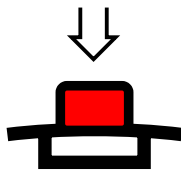
- Single click: Start/Stop Arp/Seq/Looper/AutoFade  
\*Root pitch + 1nn
- Double click: Root pitch - 1nn
- Triple click: \*Set Scale selector in fundamental position
- Hold: Skipping to the the THRES mode and back  
\*Set Root pitch in fundamental position

\* When the Chromatic mode is activated



The top switch on the center is working as follows:

- Single click: Freez Pitch  
\*Scale selector--
- Double click: PitchDrift--
- Triple click: Looper Playback speed--
- Hold: Recalling the Scene Memories



The top switch on the right is working as follows:

- Single click: Looper Start Rec / inactivate  
\*Scale selector++
- Double click: PitchDrift++
- Triple click: Looper Playback speed++
- Hold: Start/Stop the Auto Fade mode



The eyeball switch on tne center is working as follows:

- Single click: Pitch Tune Normal / -1 oct / -2 oct / +1 oct
- Double click: Phase on Ch2 Normal / Negative
- Triple click: Reserved
- Hold: Start Oscillator Tuning mode.



The signal phase of left output:  
Red : Normal  
Blue : Inverted



Start flashing when the looper is now on recording .



Pitch Drift in Octave:  
Red: Normal  
Yellow: -1 oct  
Magenta: -2 oct  
Green: +1 oct



Fixed frequency on the sub oscillators.  
White : Normal  
Cyan : Activated

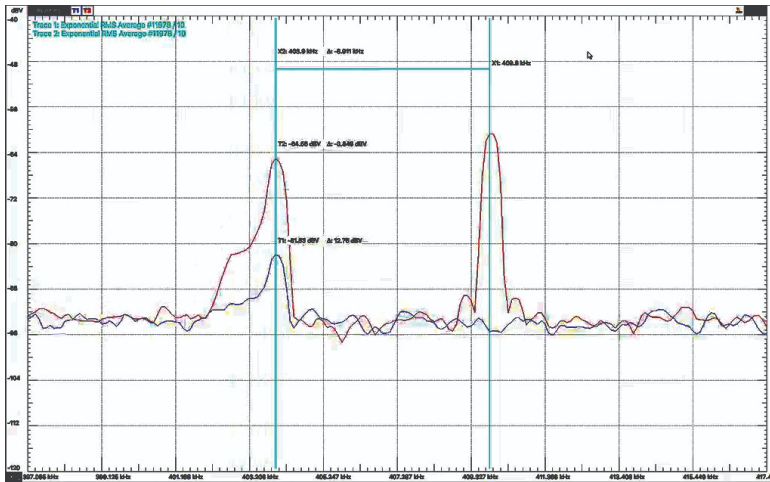


Start flashing when the recording time left 8 counts.



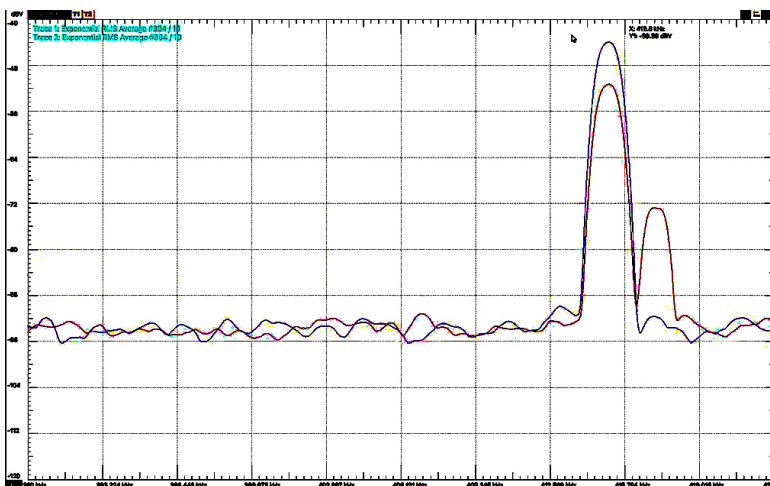
Start flashing when the AutoFade mode is activated.

# The Oscillator and the demodulator



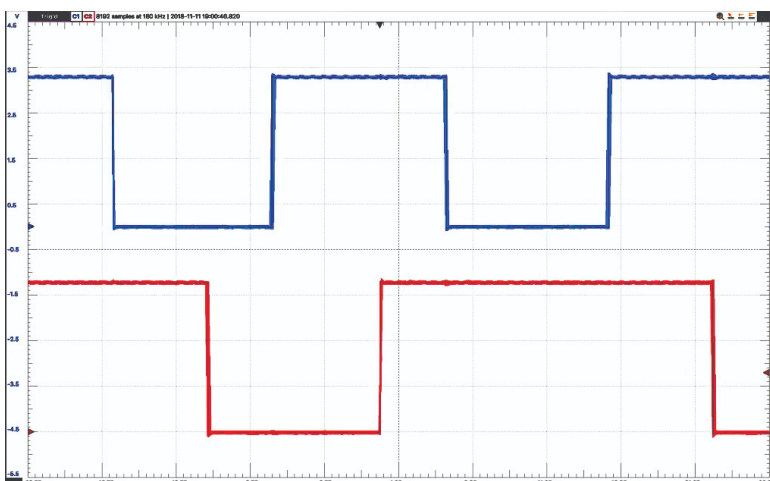
The picture shows the spectrum waveforms of the oscillator pair.

Firstly, a user should set the frequency distance between the antenna and reference oscillators around 4 to 5 kHz, then tuning the frequency of the reference oscillator side getting closer to the antenna side.



The difference of the frequencies are converted to the beat by the demodulator made from the D-FF.

The micro controller measures the time between the up-edges of the demodulator outputs.



The frequency drifts by the moving hand toward the antenna.

The distance decrease, the demodulated tone is down.

When the frequency is almost same, the beat will stop.

# The setup method after turning on the LaVoixski

After turn on the LaVoixski, player must tune the oscillators. The tuning work is a little bit difficult for the first time.

Firstly player have to understand the construction of the Theremin itself.

- The system has 2 oscillator pairs in different working style, fixed and variable.
- The 2 antennas connected to the oscillators are working in variable frequency.
- The antenna senses the magnetic field drifting by hands.
- The demodulator outputs the beat made from the difference of the frequency on the oscillators by running in the variable / fixed frequency.
- The up-edges between the waveform of the input signal is measured the time duration by the counter in the micro controller and converted to the pitch / volume values.
- The oscillators must be tuned to almost the same frequency.
- The tuning is done by the combination of 2 methods, adjusting the frequency on hardware oscillator then adding the offset value on software.

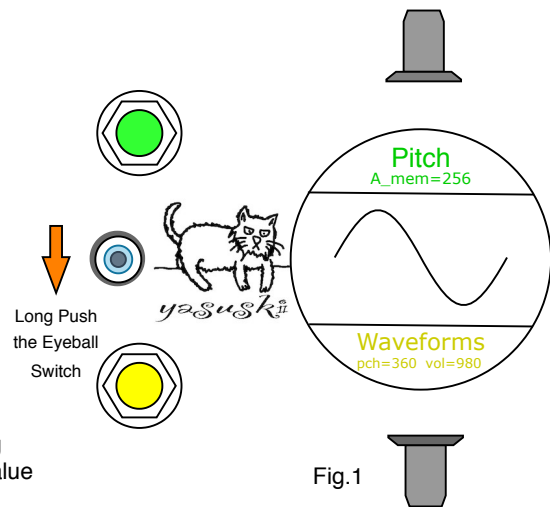


Fig.1

- Firstly, player has to search for the best tuning point by the trim pot. Player can hear the demodulated sound from the oscillator on the pitch side while long holding the eyeball LED switch. Then, player finds out where the beat appears and can check the direction of appearing in the difference.

- Tune the oscillators where the pitch turns higher while moving a hand toward the antenna. The upper knob color turns in **RED** when the frequency of the pitch oscillator has been tuned to the lower limit. (Fig.2) pch < 350

Tune the frequency just above the knob color turns in **GREEN**. (Fig.1) pch > 350

- The top switch #01 (on the left side) must be tuning "OFF" while the tuning work has been completed.

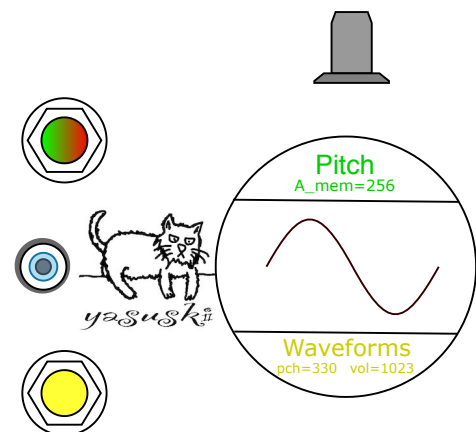


Fig.2

- The tuning method for the **volume** side is almost the same (click the top switch and holding the eyeball switch again over 1 second), but the setting is in the different manner. Player has to find out the best position for getting the widest working area while the left hand is moving toward near by the volume antenna. (Fig.3) pch = 35

- Pushing the eyeball switch again over 1 second, then tuning mode is moved to **Tuner for setting the fundamental frequency (LAVENDER)**.

- Finishing the tuning mode with holding the Eyeball switch then the top LED turns back to **BLUE**.

- Selecting the upper knob in **GREEN** again, then tune the offset of the pitch side. And pushing the upper knob over 1 second for memory the offset value to the EEPROM. (The LED color turns on 0.9s in **Magenta**) After recording the value, player do not have to do this step again.

- Selecting the upper knob in **BLUE**, then tune the offset of the volume side. And pushing the upper knob over 1 second for memory the offset value to the EEPROM. (The LED color turns on 0.9s in **Orange**)

After recording, player do not have to do this step again, too.

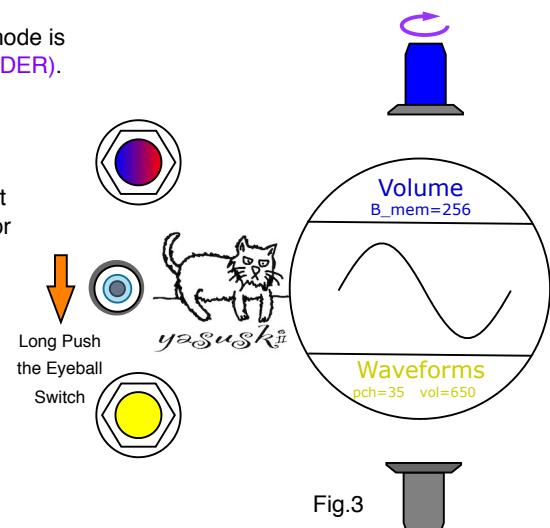
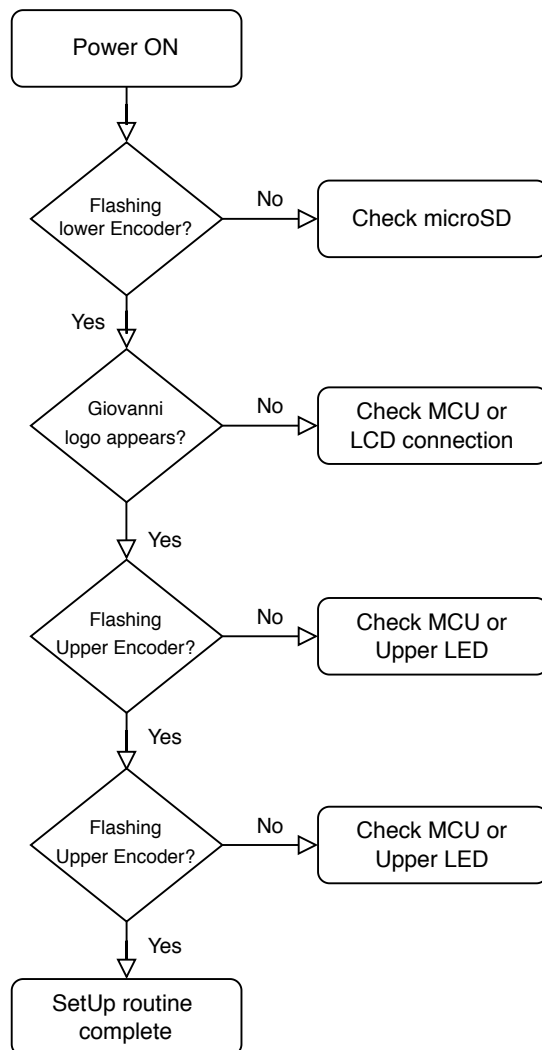
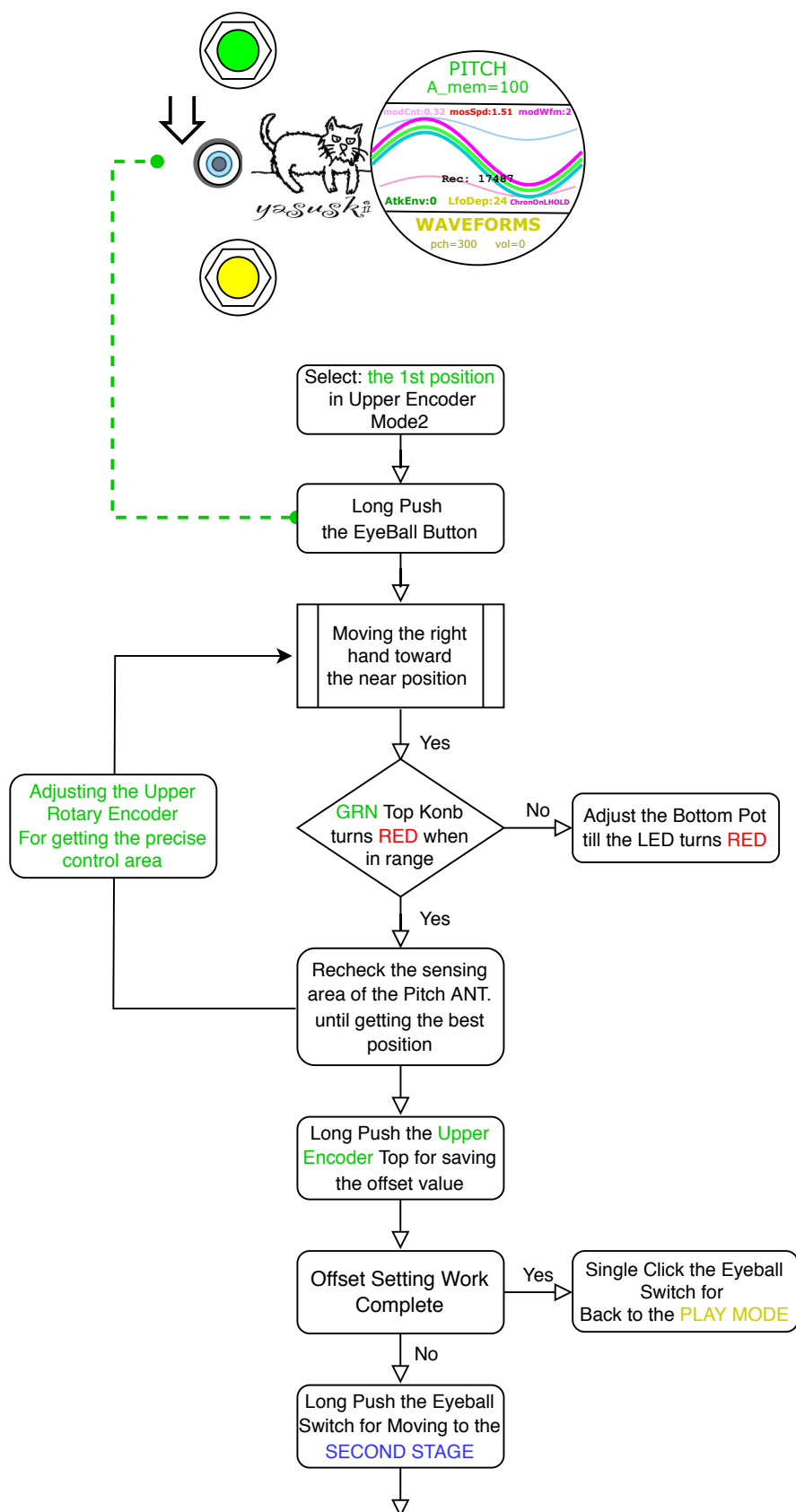
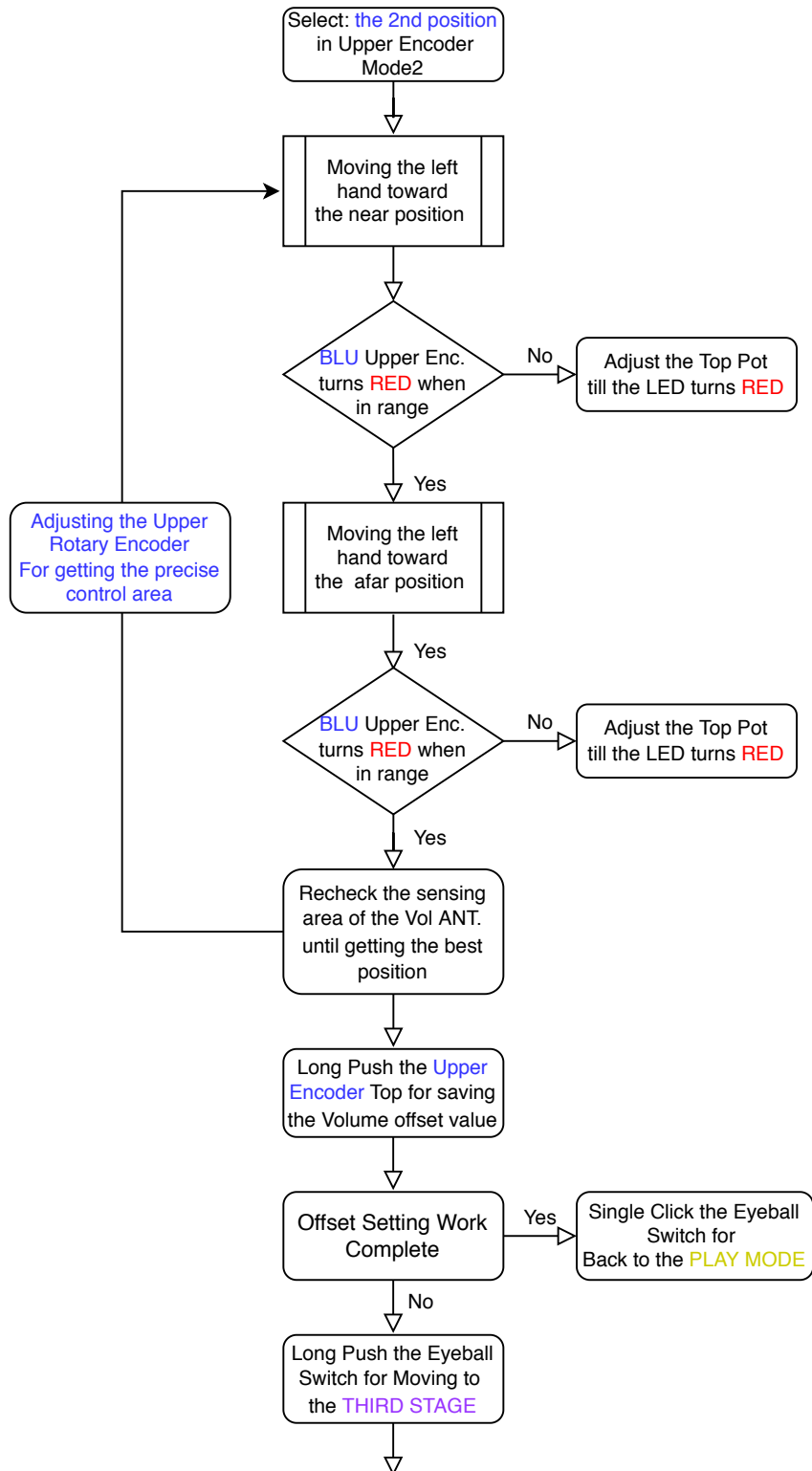
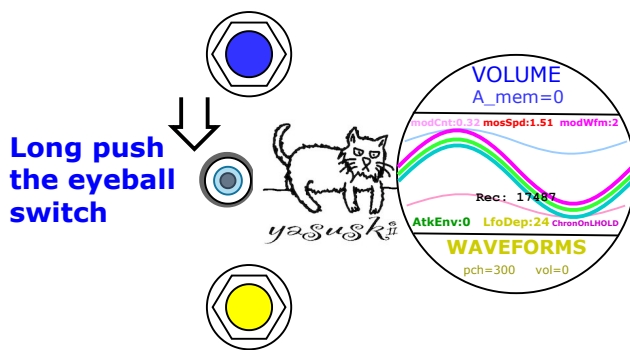
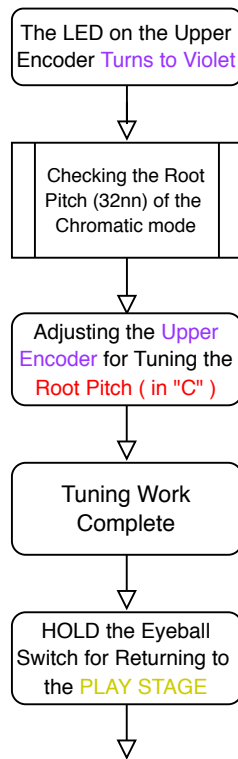
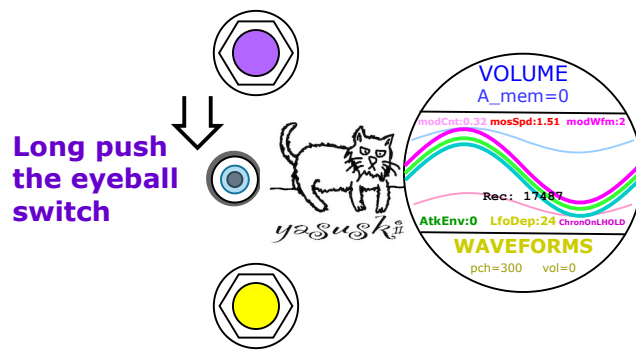


Fig.3

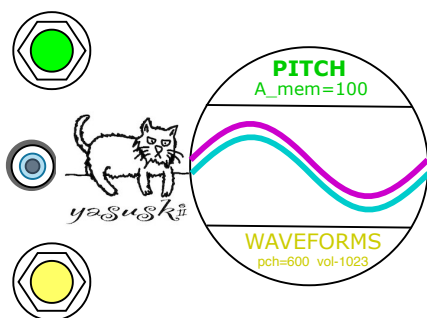






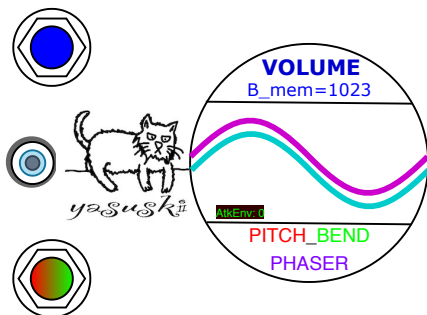


# The Voice & Playback Mode



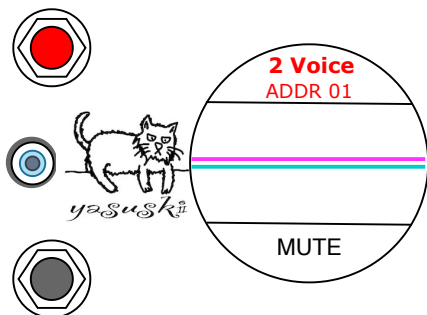
The default position is **PITCH** offset mode.

The lower knob shows the PB mode is in **NORMAL** mode.



Click the upper knob then move to **VOLUME** offset mode.

Turn Right the lower knob then the PB mode is move to **PitchBend** / **Phaser** mode. When the voice modes are selected, the upper knob accepts the **parameter input**.

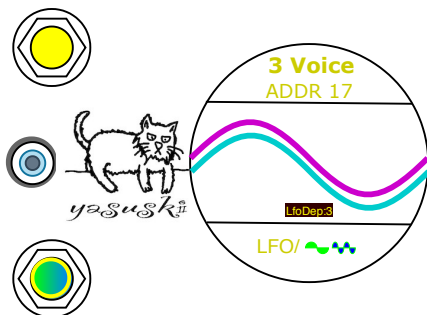


Click the upper knob then move to **2 voice mode**.

The 15 pitch combination is available in this mode.

The last one is assigned to the **Arpeggiator**.

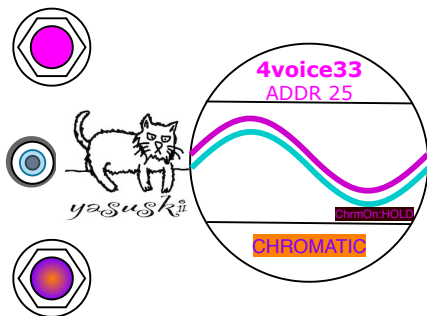
Turn Right the lower knob then the PB mode is move to **MUTE** mode.



Click the upper knob then move to **3 Voice mode**.

This mode has 7 presets with the **Arpeggiator**.

Turn Right the lower knob then the PB mode is move to **LFO** mode. When the voice modes are selected, the upper knob accepts the **parameter input**.

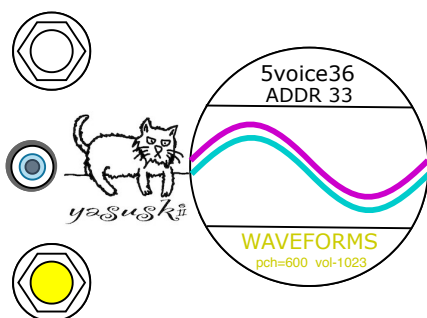


Click the upper knob then move to **4 Voice mode**.

This mode has 7 presets with the **Arpeggiator**.

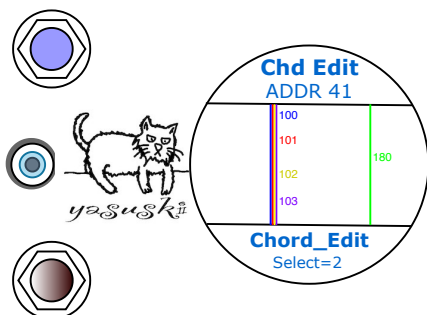
Turn Right the lower knob then the PB mode is move to **Chromatic** mode. When the voice modes are selected, the upper knob accepts the **parameter input**.

## The Voice & Playback Mode (contin'd)



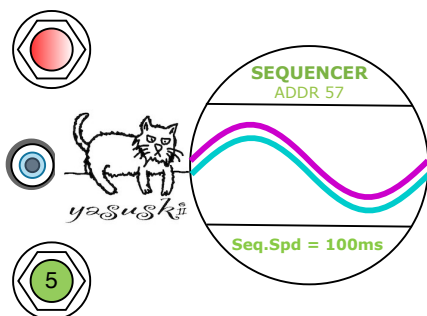
Click the upper knob then move to **5 Voice mode**.  
This mode has 7 presets with the **Arpeggiator**.

The automatic chord changer is assigned to; ADDR 36 to 39 in **5 Voice mode**.



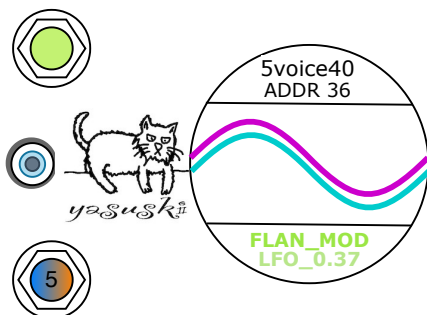
Click the upper knob then move to **chord edit mode**.  
This mode has 16 presets with the **Arpeggiator**.

Click the lower knob 2~5 times from the fundamental position then the **tuning knobs** are activated.



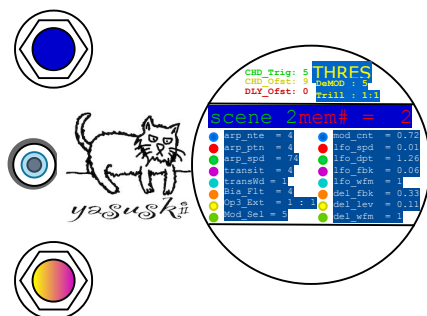
Click the upper knob then move to **SEQUENCER mode**.  
This mode has 8 sequence files.

Click the lower knob 5 times from fundamental position then the **SEQUENCER SPEED** parameter is activated.



When the Modulation Controller mode (Click the lower knob 5 times from the fundamental) is selected in the VOICE/ChordEdit modes,

The upper knob turns to the selector of the Modulation, (VCF frequency/VCF Mod Rate/FLANGER Mod Rate/Chorus Mod Rate/Dub Mod Rate) then the lower knob is working as VCF COF/LFO SPD/DPT/FBK controller.



Click the upper knob (or long push the top switch in the voice modes) then mode2 is moving to **THRES mode**. This mode set the several parameters as the threshold point of the effects/Looper Playback Speed/Looper Output Switcher/Demodulator setup/Starting address of the Chord Sequencer.

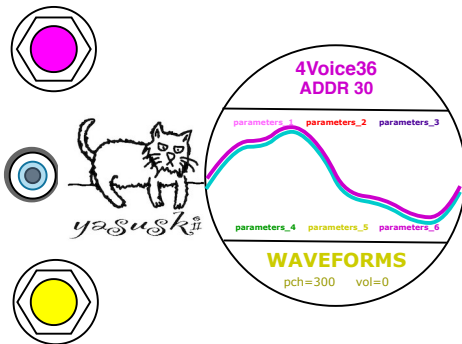
Turn right the lower knob then switching the PB mode then changing the threshold point and The parameter of the Looper/Chord Switcher.

## WAVEFORMS

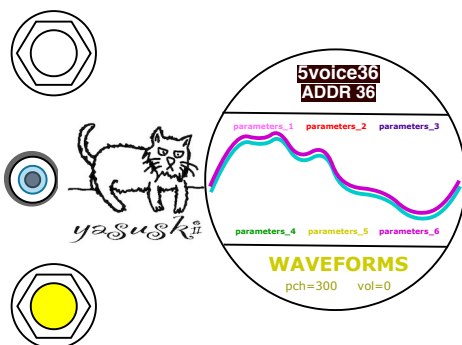
pch=600 vol-1023

# The Chord Sequencer

The Chord Sequencer is a 16 step Cyclic Harmony Generator driven by a left hand or BPM clock.



In **4 voice mode (Magenta)**, ADDR #30 and #31 are reserved for the Chord Sequencer. The Chord Sequencer switches 16 chord address selected from the chord library set of 48. The assignment of the starting address of the chord group (of 16) is set in the THRES mode. The address assignments are switched in #30 forward or #31 random by entering the threshold point on the volume value.

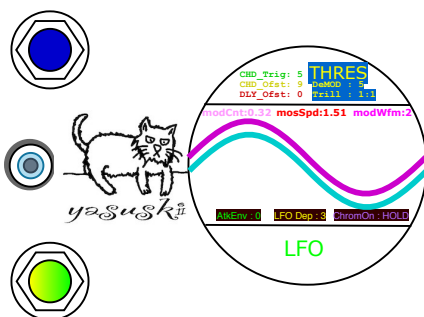


In **5 voice mode (White)**, ADDR #36 to #39 are reserved for the Chord Sequencer. The Chord Sequencer switches 16 chord address selected from the chord library set of 48. #36 (forward) and #37 (random) works as same as 4voice mode, and #38(forward) and #39(random) switching the chord works like a 16 step Sequencer.

| The Chord Library from address #01 to #48 |                  |                  |                  |                  |                  |                  |                   |
|-------------------------------------------|------------------|------------------|------------------|------------------|------------------|------------------|-------------------|
| ChordEdit42                               | ChordEdit43      | ChordEdit44      | ChordEdit45      | ChordEdit46      | ChordEdit47      | ChordEdit48      | ChordEdit49       |
| ChordEdit50                               | ChordEdit51      | ChordEdit52      | ChordEdit53      | ChordEdit54      | ChordEdit55      | ChordEdit56      | ChordEdit57       |
| Chd06 = D#D#G#G#                          | Chd07 = G,G,B,B, | Chd08 = E,G,B,E, | Chd09 = E,G,A,B, | Chd10 = E,G,2,E, | Chd11 = D#F#A,0, | Chd12 = D,E,G,2, | Chd13 = F,G,A#,3, |
| Chd14 = F,G,B,D,                          | Chd15 = D,G,A,B, | Chd16 = D#G,B,D, | Chd17 = a,C,E,G# | Chd18 = C#D,F,A, | Chd19 = D,F,G,A, | Chd20 = C,D#F#A, | Chd21 = A#C,E,G#  |
| C,As,Ds2,G2                               | C,Ds,As,D2       | C,As,E2,G2       | C,F,As,Ds        | C,F,As,D         | C,D2,F2,AA2      | C,D2,E2,G        | C2,F2,G2,AA2      |
| B,D2,E2,G2                                | C,E,F,A          | C,F,A,As         | C,E,F,A          | C,E2,G2,B        | C,Ds2,Gs2,As2    | C,D2,G2,AA2      | C,Cs,Fs2,Gs2      |

\*The C#SV chords are read from the text file on the micro SD card.

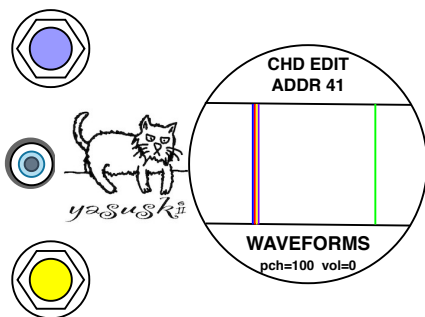
\*the C#SV chords are stored on the ROM.



Click the upper knob 8 times (or double click once} from the fundamental position or "HOLD" the top switch on the left side at the Playback mode position on the lower knob, then the voice mode is moved to the THRES mode.

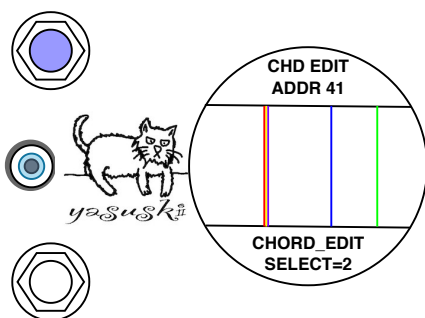
Turn the lower knob clockwise to enter LFO mode. At this time, the upper knob is assigned to a parameter that sets the starting address for the Chord Sequencer, CHD\_Ofst = pot0.

# Microtonal Tuning / Chord Edit Mode



In the Chord Edit mode, the user can tune 4 individual pitches. The tuning rules are not depend on the Equal temperament. Firstly, the user start the edit work by accessing to the neutral position of the Chord Edit mode. At the neutral position, the user can hear the fixed pitch (Green line) which is reserved for the marker for start tuning.

There are 16 editable channels. (ADDR #42 to #57)  
The example shows the first step for editing the pitches.

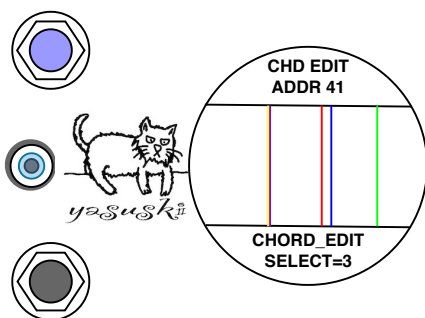


Access to the Chord Edit mode on the lower knob by clicking (forward) 2 times from the neutral position (Yellow turns White).

After selecting the neutral position of the Pitch Edit mode (ADDR #41), moving to the parameters for controlling the pitch of the oscillators.

When the user access to the tuner for 2nd OSC, LED on the lower knob turns from Yellow to White. For accessing to the next pitch (Red line), click the lower knob once. (White turns to Black)

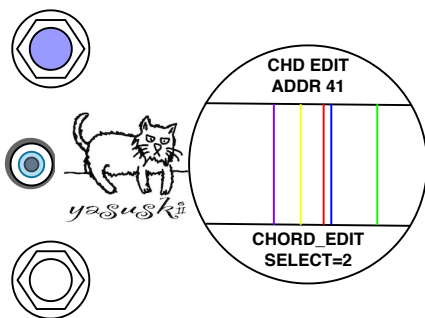
**\*\*If the mixout with transition mode is selected as the audio source, the user can not able to hear the desired pitch for tuning.**



For checking the sound, move the left hand position towards the antenna for searching the peak level or re-select the mixed signal with equal audio output levels without the transitions.

Click the top of the lower knob and Moving to the next parameter (Yellow). The LED color on the lower knob changes from Black to White.

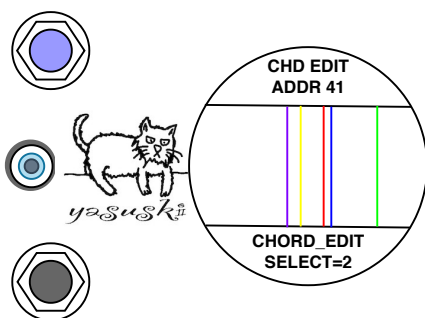
The combinations of the pitches in Chord Edit mode also be the resources of the chord book for the arpeggiator. (see the chapter "Arpeggiator")



Turn the lower knob for tuning.

After complete the tuning work, click the knob for moving to the next pitch (Magenta). The LED color changes from White to Black.

The next stage is the final tuning point.



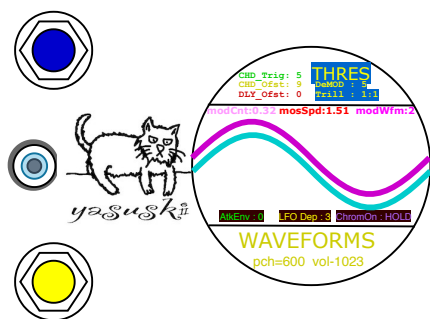
After the tuning work is completed, select the desired recording position on the upper knob.

If you want to re-check the sound, turn back to the neutral position by rotating the knob.

After deciding the recording position, push the top of the lower knob over 1 second for storing the data onto the microSD card.

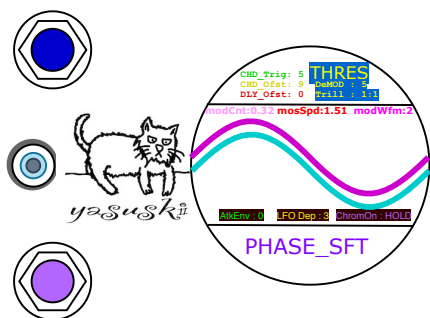
Move to the desired position you want to edit.

# THRES Mode



Click the upper knob 8 times (or double click once) from the fundamental position or "HOLD" the top switch on the left side at the Playback mode position on the lower knob, then the voice mode is moved to the THRES mode.

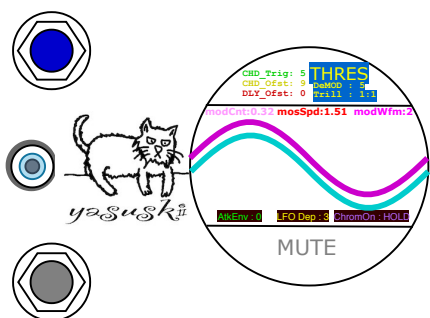
At this time, the upper knob is assigned to the DeMOD parameter for switching the FVC absorber / Looper output selector, DeMOD = pot00z.



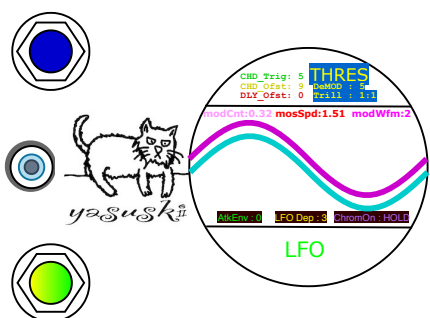
Turning the lower knob clockwise will enter pitch bend/phase shift mode.

If the PHASE\_SFT is selected, the upper knob is assigned to a parameter that sets the looper playback speed = pot00q.

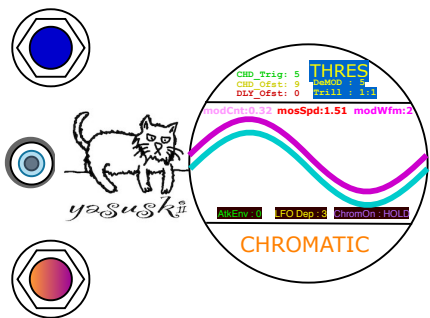
If the PITCHBEND mode is selected, the upper knob is assigned to a parameter CHD\_Trig = pot00a.



Turn the lower knob clockwise to enter MUTE mode. At this time, the upper knob is assigned to the trill note pattern selector pitch\_drift = pot00v.

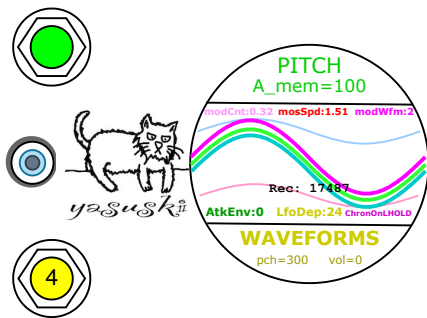


Turn the lower knob clockwise to enter LFO mode. At this time, the upper knob is assigned to a parameter that sets the starting address for the Chord Sequencer, CHD\_Ofst = pot0.



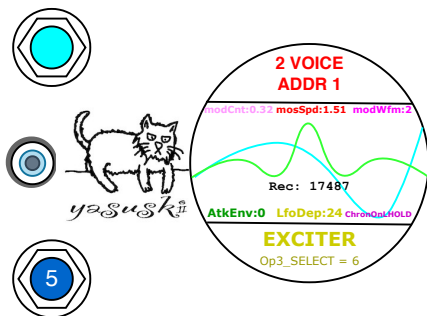
Turn the lower knob clockwise to enter CHROMATIC mode. At this time, the upper knob is assigned to a parameter that sets the delay time magnifier, DLY\_Ofst = pot00g.

# The Parameters on the lower knob



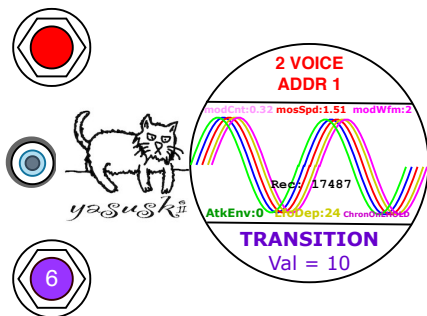
The lower knob is mainly assigned to the selectors for the audio source or the parameters for the sound effect controllers.

The fundamental position is the normal playback mode.



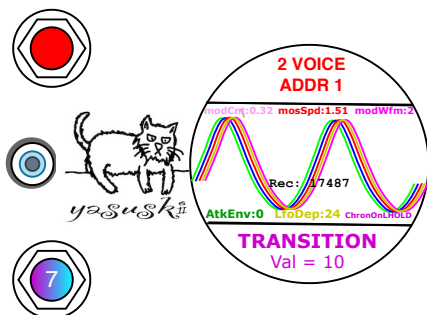
Click the lower knob then the parameter is moved to : Op3 / Exciter mode.

The function is switched by clicking the upper knob.



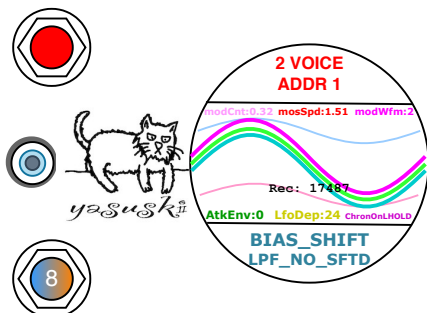
Click the lower knob then the parameter is moved to : the Transition Distance controller.

The value is changed by turning the lower knob.



Click the lower knob then the parameter is moved to : the Transition Waveform switcher.

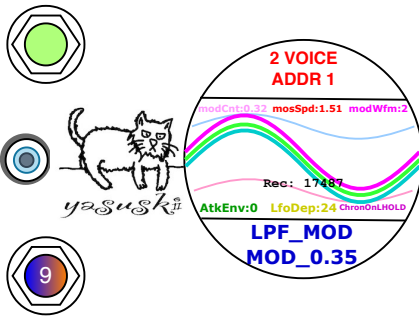
The waveform is switched by turning the lower knob.



Click the lower knob then the parameter is moved to : the LPF / HPF / Distortion / USB\_MIDI selector.

The status is switched by turning the lower knob.

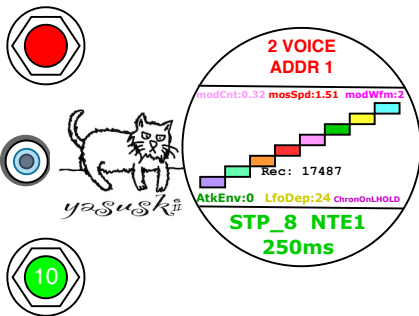
# The Parameters on the Lower Knob (contin'd)



Click the lower knob then the parameter is moved to :  
LPF Frequency / LPF Modulation / Flanger / Chorus / Dub  
Rate controller mode.

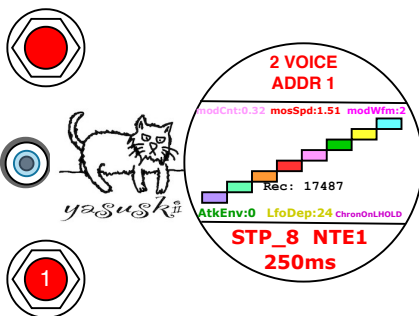
The function is switched by turning the upper knob.

\*When the Sequencer mode is selected, this stage is  
assigned to the parameter "Sequence Speed".



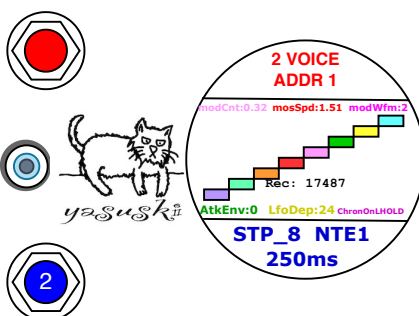
Click the lower knob then the parameter is moved to :  
speed control for the Arpeggiator.

The arp speed is controlled by turning the lower knob.



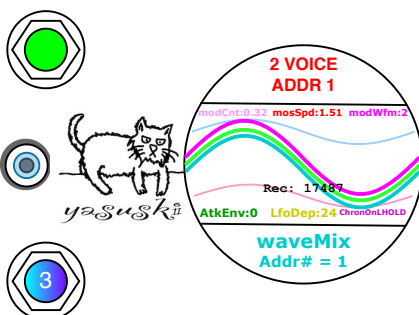
Click the lower knob then the parameter is moved to :  
Pattern selector (Up / Alternate / Down / Random)  
for the Arpeggiator.

The arp pattern is switched by turning the lower knob.



Click the lower knob then the parameter is moved to :  
Note pattern selector (17 preset with 15 editable notes)  
for the Arpeggiator. The preset notes are written in the file  
by the C#SV format on the microSD.

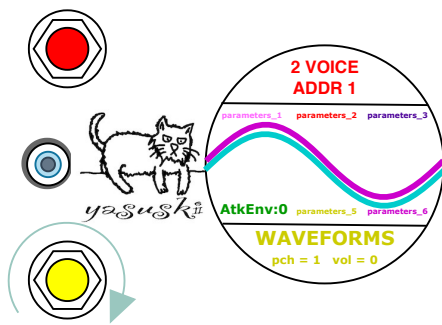
The arp notes is switched by turning the lower knob.



Click the lower knob then the parameter is moved to :  
Waveform Mix Selector / Edit mode. (the 14 preset with 18  
editable waveform mix) The parameters (32 wavetables with  
5 levels) controlled by the upper knob is changed by clicking  
the upper  
knob, then turning the knob for selecting / changing the values.

The waveform mix memory address is switched by turning the  
lower knob.

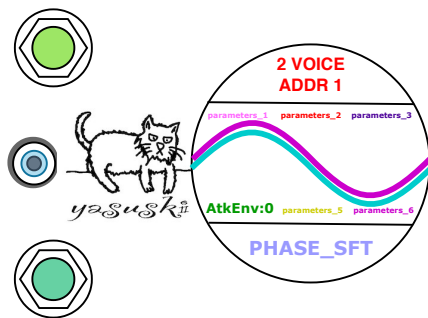
# Pitch Bend and Phase Shifter



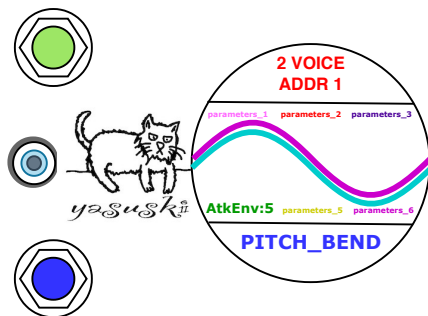
The PitchBend mode is available on all voicing modes. The Up/Down direction is selectable by the border point of zero value on the pot.

Attention: “Bend Down” means “up bended pitch turns to equal to the un-bended pitch” in decided time.

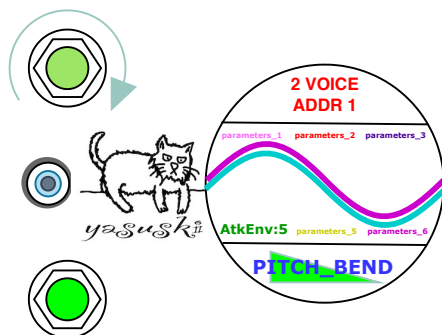
“Bend Up” do the alternative work. Please imagine the sound of the guitar playing method named as “Harmonized choking”.



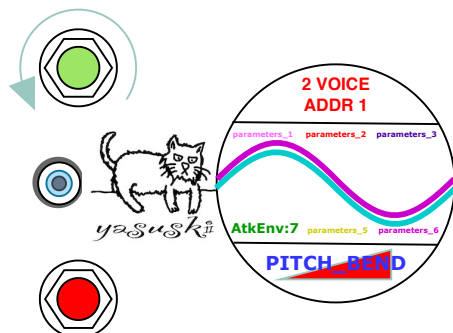
Access to the parameter “MUTE Switch” on the lower knob as the power-on neutral position(Yellow). Rotate the lower knob clockwise then the LED color on the lower knob turns Indigo showing at the zero point. Then the LED on the upper knob turns a color in LEAF.



In the PitchBend mode, the upper knob accepts the value for controlling the bending direction and depth. The gating point is related with the Volume value and is set by the parameter in the Threshold mode. At this time, the effect mode is set as the "PHASER". When the zero value is varied, the LED turns in BLUE.



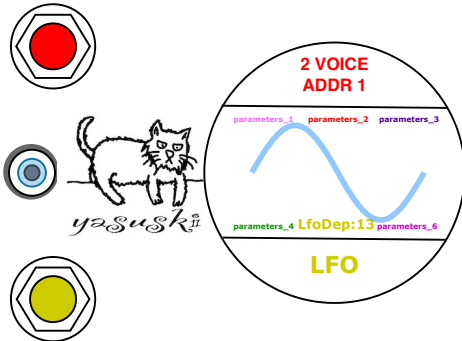
The value of the bending direction is switched by the border of the “zero” value on the upper knob. Turning clockwise, the LED turns Green when the bend down action is activated and increasing the value of the bending time.



Turning counter clockwise select the bend up mode and increasing the value of the bend depth. The LED turns Red when the bend up action is activated.

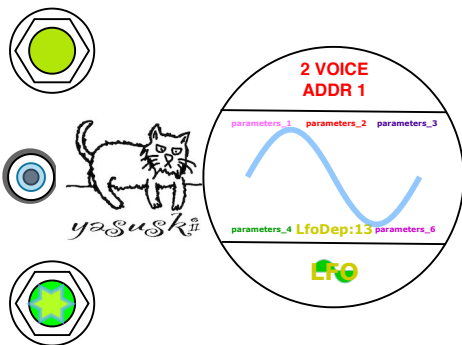
After complete the editing work, push the top of the knob over 1 second for recording the values of the parameters on the scene memory on the Voice Mode.

# The Tremolo Mode

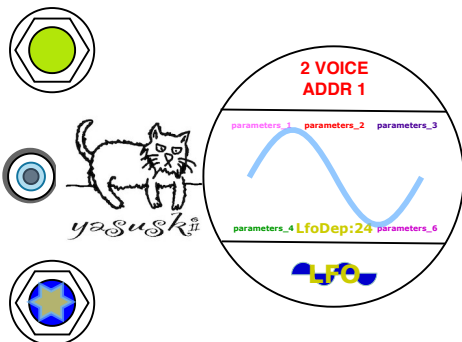


The Tremolo mode is available on all voicing modes. The shape of the LFO signal is editable by the value on the pot.

Access to the parameter "Mute" on the lower knob as the power-on neutral position (Yellow). Rotate the lower knob clockwise in 2 steps then the LED color turns **Green**/**Blue**.



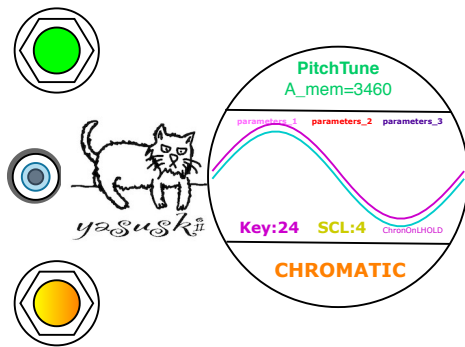
In the Tremolo mode, the upper knob accepts the value for controlling the shape of the LFO signal and speed magnification (**x1** / **x2**). The LED turns **Green** when the upper knob is rotated in clockwise.



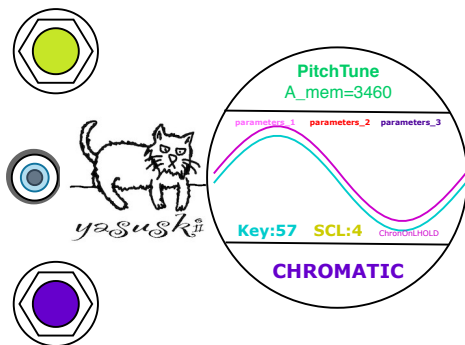
The LED turns **Blue** when the upper knob rotates in clockwise. and the LFO is working on **x2 speed**.

Rotating action increases the value of the depth. After completing the edit work, push the top of the knob over 1 second for recording the values of the LFO depth parameter.

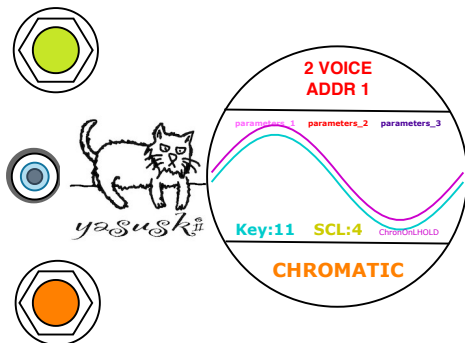
# The Chromatic Mode



The Chromatic mode is available on all voicing modes. Access to the parameter "MUTE Switch" on the lower knob as the power-on neutral position(Yellow). Rotate the lower knob clockwise 3 notches then the LED color on the upper knob turns **Leaf** and lower knob turns **Orange**.

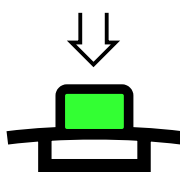


When the **PitchTune** mode is selected with the upper knob, the upper knob is assigned to the parameter that sets the activation time of the chromatic mode. Then the LED color on the upper knob turns to **Leaf**. When the Chromatic mode is inactivated, the LED on the lower knob turns **Purple**.



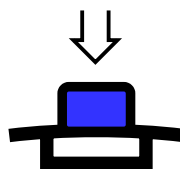
Click the upper knob twice, then moving to the **2 voice mode**. The upper knob has been working for the scale selector at this time. The color of the upper knob is still colored in **Leaf**. The 32 preset scales are available.

The 3 top switches are working in the special tasks when the Chromatic mode is selected.



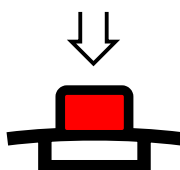
The top switch on the left is working as follows:

Single click: noteNumber++  
Double click: noteNumber--  
Triple click: Scales is in the neutral position  
Hold: noteNumber is in the neutral position.



The top switch on the center is working as follows:

Single click: Scales--  
Double click: PitchDrift--  
Triple click: Playback speed on the Looper--  
Hold: Recalling the Scene Memories



The top switch on the right is working as follows:

Single click: Scales++  
Double click: PitchDrift++  
Triple click: Playback speed on the Looper--  
Hold: Activating the Auto Fade mode

# The Chromatic Mode (Continue'd)

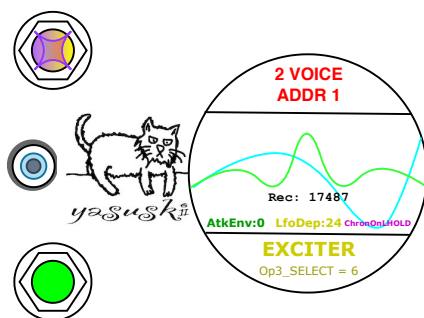
| SCALES |   |    |   |    |   |   |    |   |    |   |    |   |
|--------|---|----|---|----|---|---|----|---|----|---|----|---|
|        | C | C# | D | D# | E | F | F# | G | G# | A | A# | B |
| 1      | X | X  | X | X  | X | X | X  | X | X  | X | X  | X |
| 2      | X |    | X |    |   |   | X  |   | X  |   |    |   |
| 3      | X |    | X |    |   | X |    | X | X  | X |    |   |
| 4      | X |    | X |    | X |   |    | X |    | X |    |   |
| 5      | X |    | X |    |   | X |    | X |    | X |    |   |
| 6      | X |    | X | X  |   |   |    | X |    | X |    |   |
| 7      | X |    |   | X  |   |   | X  |   |    | X |    |   |
| 8      | X |    | X |    |   | X |    |   | X  | X |    |   |
| 9      | X |    | X |    | X |   | X  |   | X  |   | X  |   |
| 10     | X | X  |   |    | X | X |    | X | X  |   |    | X |
| 11     | X |    | X |    |   |   | X  | X |    | X |    |   |
| 12     | X | X  |   |    | X |   | X  |   | X  |   | X  |   |
| 13     | X |    | X | X  |   |   | X  | X |    | X | X  |   |
| 14     | X |    | X |    | X | X |    |   | X  | X |    |   |
| 15     | X |    | X | X  |   |   | X  | X |    | X |    | X |
| 16     | X |    | X | X  |   |   | X  | X |    | X |    |   |
| 17     | X |    | X |    | X | X |    | X |    | X |    | X |
| 18     | X |    | X |    | X |   | X  | X |    | X |    | X |
| 19     | X |    | X |    | X | X |    | X |    | X | X  |   |
| 20     | X |    | X |    |   | X |    |   |    | X |    |   |
| 21     | X |    |   |    |   | X |    | X |    |   |    | X |
| 22     | X |    | X |    | X |   |    | X |    |   |    |   |
| 23     | X |    | X | X  |   | X |    | X |    | X | X  |   |
| 24     | X |    | X | X  |   |   | X  |   | X  | X |    | X |
| 25     | X | X  |   | X  |   | X |    | X | X  |   | X  |   |
| 26     | X |    |   | X  |   | X | X  |   | X  |   | X  |   |
| 27     | X |    | X | X  |   | X | X  |   | X  |   | X  |   |
| 28     | X |    | X | X  |   |   | X  | X |    | X | X  |   |
| 29     | X | X  | X | X  |   |   | X  |   | X  | X | X  |   |
| 30     | X |    | X | X  |   |   | X  |   | X  |   | X  |   |
| 31     | X |    |   | X  | X |   | X  |   | X  | X |    | X |
| 32     | C | C# |   |    | E | F | F# |   | G# | A | A# |   |

# The Exciter

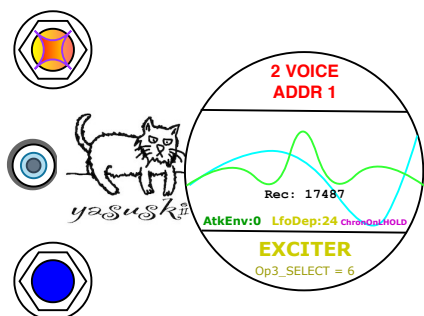
The "Op3" has the selector for the output source, Transition / Normal envelope and Exciter.

In Exciter mode, the input signal is reading out the wavetable which is generated for trans- forming from by Chebysheve's polinomial expressions.

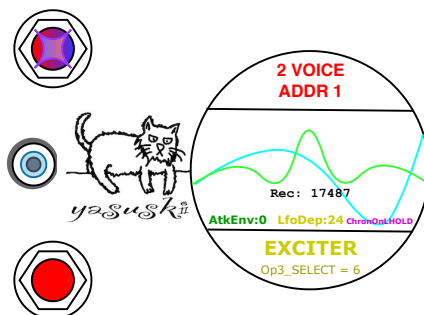
The 5 parameters as the fundamental and 4 harmonics levels are available for generating the wavetable. Fig.6 shows the generated wavetables by different parameter settings. The user can edit the mixing level of the harmonics by the following steps.



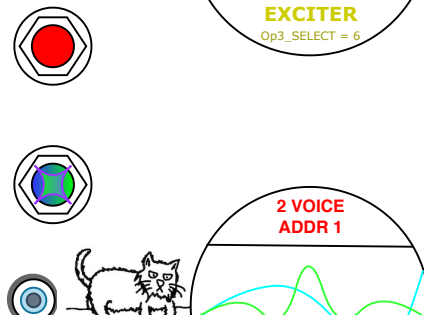
- 1) Double click the upper knob then select the Harmonic Level 5 (Magenta)
- 2) Turn the upper knob for setting the Harmonic Level 5.
- 3) The Blinking speed of the upper LED show the values of the parameters.



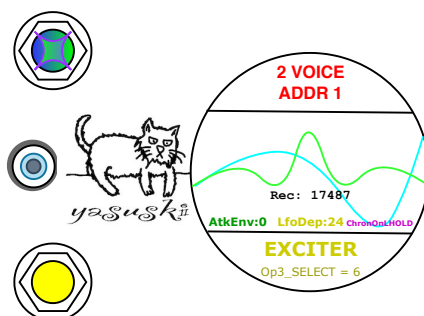
- 1) Double click the upper knob then select the Harmonic Level 4 (Yellow)
- 2) Turn the upper knob for setting the Harmonic Level 4.
- 3) The Blinking speed of the upper LED show the values of the parameters.



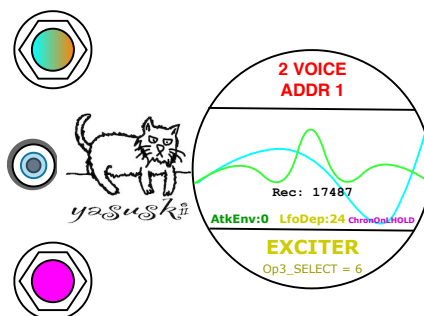
- 1) Double click the upper knob then select the Harmonic Level 3 (Red)
- 2) Turn the upper knob for setting the Harmonic Level 3.
- 3) The Blinking speed of the upper LED show the values of the parameters.



- 1) Double click the upper knob then select the Harmonic Level 2 (Blue)
- 2) Turn the upper knob for setting the Harmonic Level 2
- 3) The Blinking speed of the upper LED show the values of the parameters.



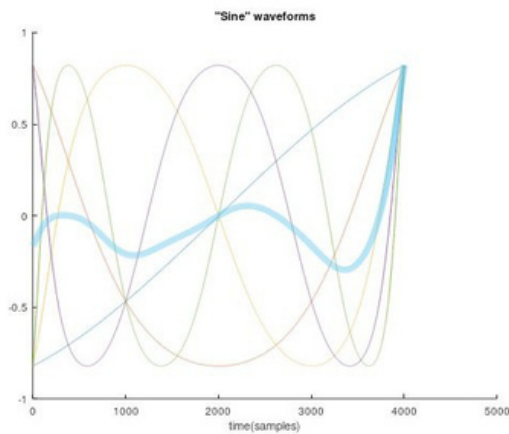
- 1) Double click the upper knob then select the Harmonic Level 1 (Green)
- 2) Turn the upper knob for setting the Harmonic Level 1.
- 3) The Blinking speed of the upper LED show the values of the parameters.



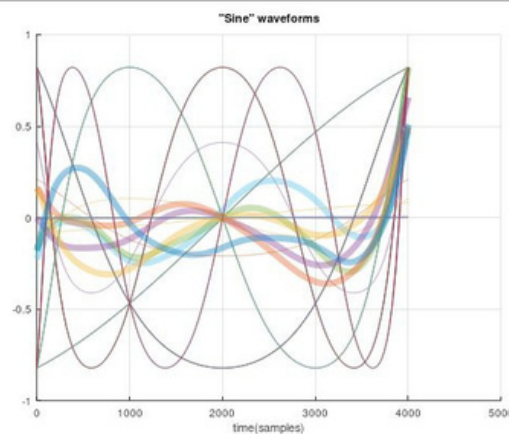
- 1) Double click the upper knob 4 times then return to the starting point.
- 2) Turn the upper knob for selecting the Memory Bank (3) of the Exciter.
- 3) Hold the top knob for recording the parameters to the Memory Bank. then the LCD shows the details of the parameters.

The lower knob is assigned to the output selector for CH.3. Turn the knob for selecting a signal from 5 oscillator output. The first group of 5 set the output level with transition. The second group of 5 set the output level with normal envelope. The last group of 5 set the transfer output with normal envelope. The LED on the lower knob shows which oscillator is assigned from.

# Audio Excitation



**Fig.5 Transformation example 1**



**Fig.6 Transformation example 2**

The player can select 3 output mode, Transition / Normal envelope and Exciter on Op3.

In Exciter mode, the input signal is reading out the wavetable which is generated for transforming from by Chebyshev's polynomial expressions. ( see Fig.5 )

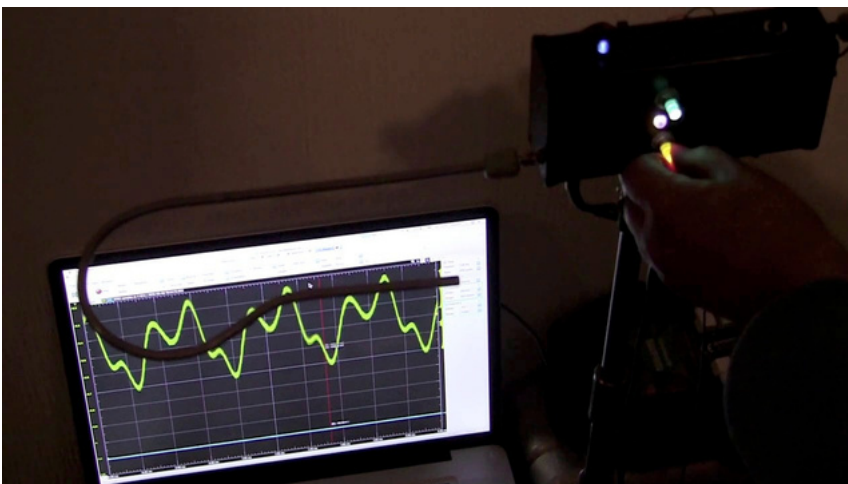
The 5 parameters as the fundamental and 4 harmonics levels are available for generating the wavetable.

Fig.6 shows the generated wavetables by different parameter settings. The user can edit the mixing level of the harmonics by the following steps.

- 1) Select the Exciter mode on the lower knob.
- 2) Select the OSC levels (Green, Blue, Red, Yellow and Purple) on the upper knob and edit the harmonic levels.
- 3) Push the upper knob on the memory bank for saving the parameter setup.
- 4) The Blinking speed of the LED shows the values of the parameters.

The method of the audio excitation is inspired from the "Arduino Music and Audio Projects" by Mike Cook.

The Exciter generates the extra harmonics from the pure sine wave input. This is a helpful effects for enhancing the audio character from the single waveform output.



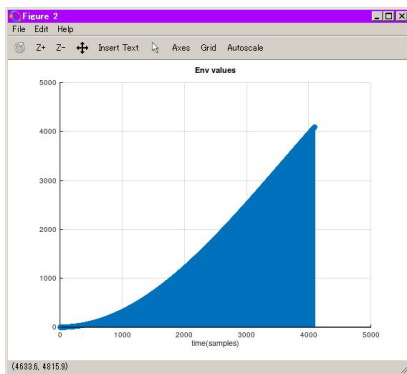
**Fig.7 Setting the wavetable for transformation**

The picture (Fig.7) shows a transformed sine input in Exciter mode.

Unfortunately, the parameters cannot change in the real time because it takes over 2 second for generating the wavetable.

The parameters are stored into microSD card when the edit work has been completed.

# The Transition



The audio output levels from the oscillators are controlled by the envelope wavetables read out from the text file in the microSD card.

There are 2 types of the control envelopes available, for total volume control and transition control. The data format of the total volume is 12 bits depth with 11 bits length.

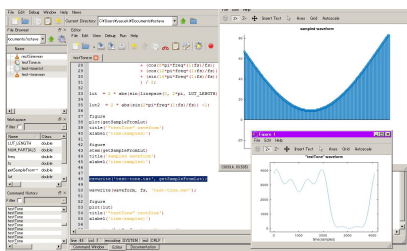
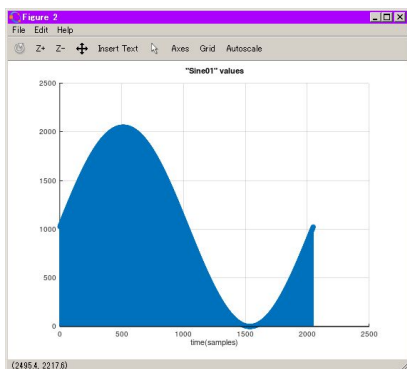
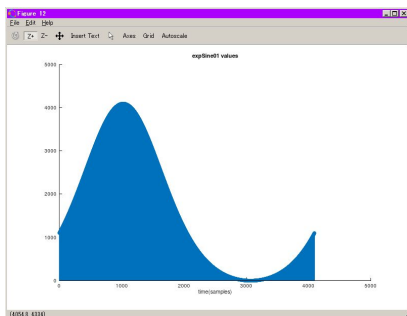
In the Transition mode, the audio output level from the oscillators are controlled by the envelopes which are in the different phase. The phase of the each envelopes are set by the parameter named "Transition" which adds the offsets of the readout points to the each waveforms.

There are 2 control waveforms available. The 1st one is Exp.Sine curved waveform. It has narrower shape on the peak side than the normal sinewave.

The 2nd is Sine curved waveform. It has the wider peak shape than the Exp.Sine curve. It means that the Sine wave has the wider mixing area around the peak level than Exp.Sine wave. On the other hands, wider mixing area makes the distorted signals when the peak points are getting closer to closer.

The narrow peaked envelope has the better separation with less distortion.

The file format of both waveforms is 11 x 11 bits.

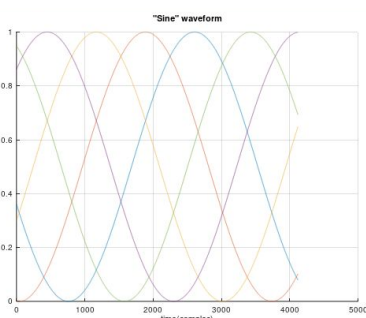


The waveforms are generated by the GNU Octave in CSV format. GNU Octave is a free math software which is downloadable from: <https://www.gnu.org/software/octave/>

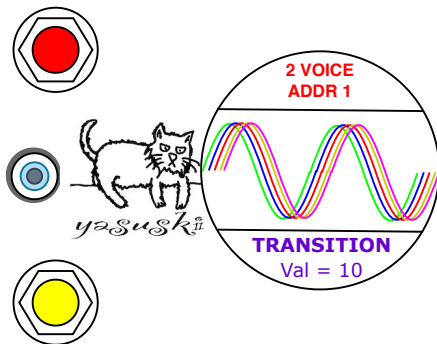
The software works on the Windows and Linux. Unfortunately, the Mac version needs a little bit complex works for the installation.

The reading point of the envelopes which controls the audio output levels between the oscillator outputs are varied by the position of a left hand towards the Volume antenna.

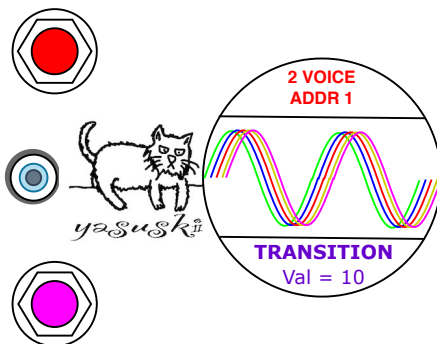
The parameter "Transition" controls the distance / phase between the envelopes. The parameter "Transition Selector" switches the wavetables for the envelopes.



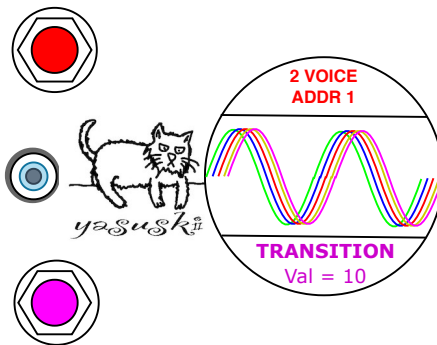
# The Transition (continue'd)



Click the lower knob twice from the neutral position (yellow) to enter TRANSITION mode. The LED on the upper knob lights up in Magenta color, which means the knob is accessing to the parameter of the TRANSITION.

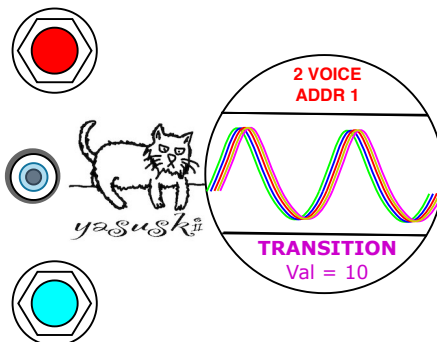


Turn the clockwise lower knob for setting the Transition. Then the distance between the waveforms are changed.



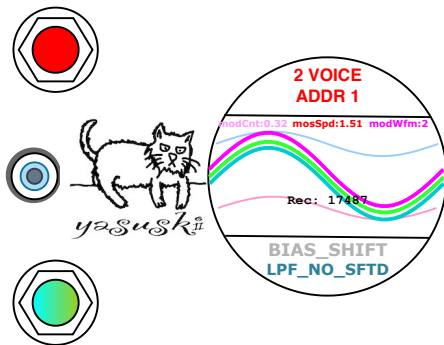
Click the lower knob then the parameter is moved to : the Transition Waveform switcher.

The waveform is switched by turning the lower knob.

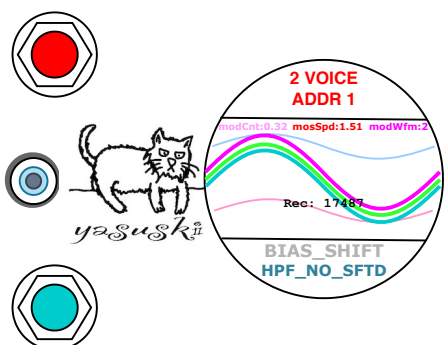


Turn the lower knob for switching the Transition Waveform. Then the LED color on the lower knob lights up in Cyan .

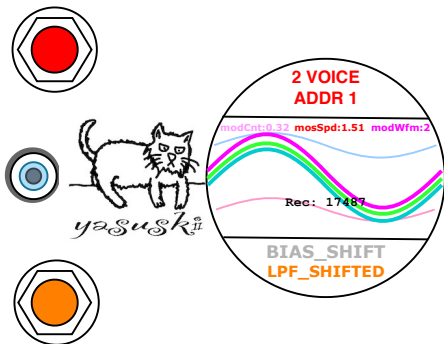
# The Filter Selector / BIAS\_SHIFT # 1



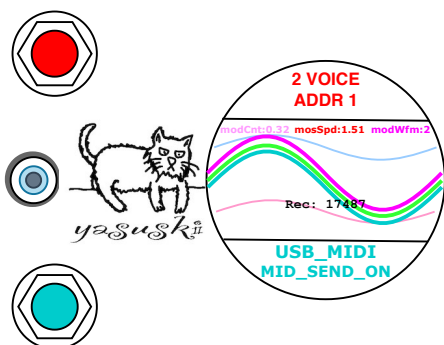
Click the lower knob 4 times from the neutral position (yellow) to enter BIAS\_SHIFT mode. The LED on the lower knob lights up in Cyan, which means the knob is accessing to the mode selector of the LPF / HPF / BIAS\_SHIFT / USB\_MIDI / Input Selectors.



Turn the lower knob clockwise from the neutral position "LPF with NO\_BIAS\_SHIFT". The LED on the lower knob still lights up in Cyan, which means the knob is activating the mode "HPF with NO\_BIAS\_SHIFT".

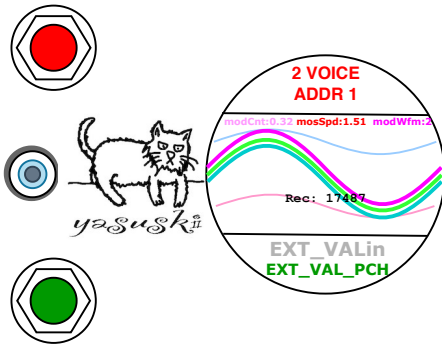


Turn the lower knob clockwise then the LED on the lower knob lights up in Orange, which means the knob is activating the mode "Distorted signals with BIAS\_SHIFT". The BIAS\_SHIFT is analog circuit not coding on the software, which is activated by the analog switches.

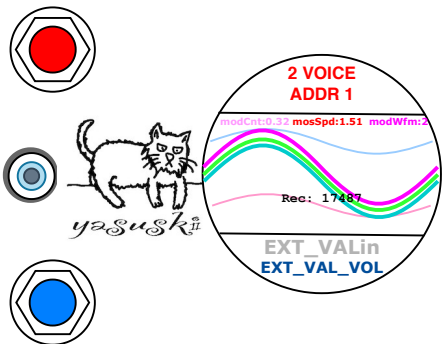


Turn the lower knob clockwise then the LED on the lower knob lights up in DarkCyan, which means the knob is activating the mode "USB\_MIDI\_OUT with NO\_BIAS\_SHIFT".

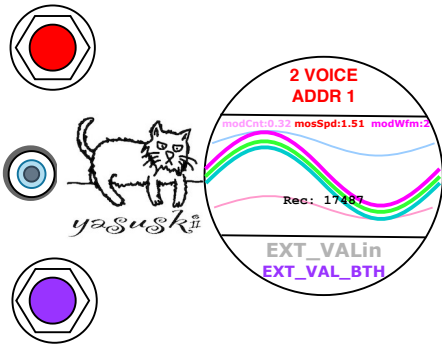
# The Filter Selector / BIAS\_SHIFT continue'd



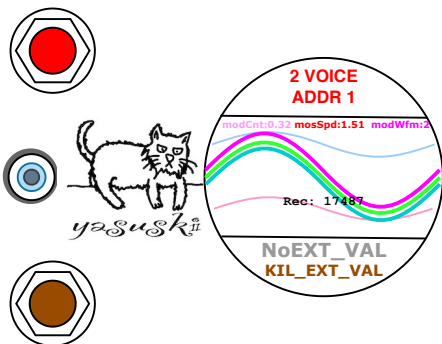
Turn the lower knob clockwise then the LED on the lower knob lights up in DarkGreen, which means the knob is activating the mode "EXT\_VALin with EXT\_VAL\_PCH".



Turn the lower knob clockwise then the LED on the lower knob lights up in Indigo, which means the knob is activating the mode "EXT\_VALin with EXT\_VAL\_VOL".

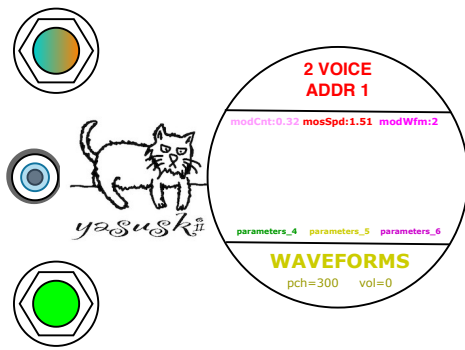


Turn the lower knob clockwise then the LED on the lower knob lights up in Violet, which means the knob is activating the mode "EXT\_VALin with EXT\_VAL\_BTH".

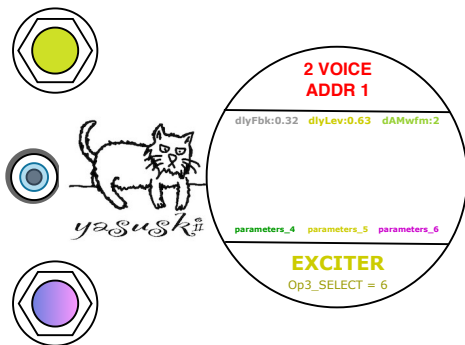


Turn the lower knob clockwise then the LED on the lower knob lights up in Brown, which means the knob is activating the mode "NoEXT\_VAL with KIL\_EXT\_VAL".

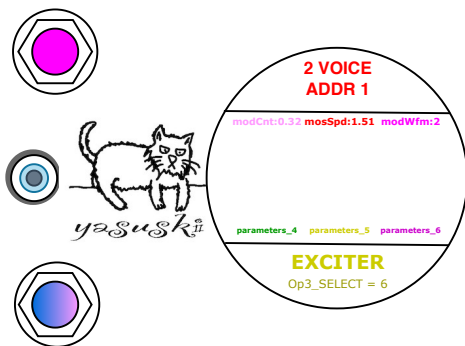
# The Delay & LPF controller



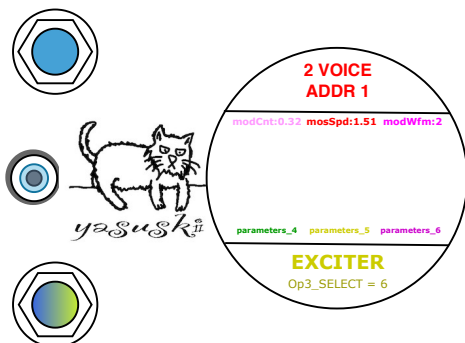
Click the lower knob once from the neutral position (yellow) to enter EXCITER mode. The LED on the upper knob lights up in SkyBlue / DarkCyan / Orange color, which means the knob is accessing to the preset selector of the EXCITER.



Click the upper knob to move to the Delay controller (color changes to Leaf). The lower knob changes to the parameter selector. The fundamental is switching to the Delay feedback (Indigo). Then the upper knob accepts to set the values. The next, turn the lower knob until the LED turns to Leaf to select the level controller for the Delay units. Then the upper knob accepts to Delay Level .

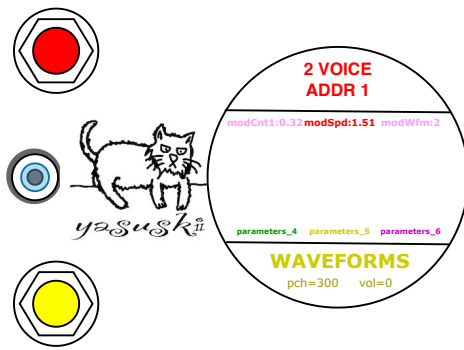


Click the upper knob to move to the LPF controller (color changes to Magenta). The lower knob changes to the effect mode selector. The fundamental is switching to the LPF cut off frequency (Indigo). The next, turn the lower knob until the LED turns to Pink to select the modulator for the LPF. Then the upper knob accepts to LFO speed.

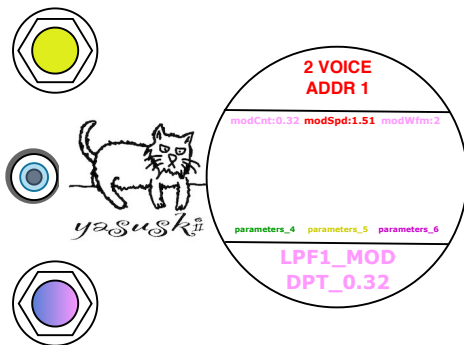


Click the upper knob to move to the LFO1/2 controller (color changes to DarkCyan). The lower knob changes to the selector for LFO1/2. The fundamental is switching to the LFO1 waveform selector (Indigo). The next, turn the lower knob until the LED turns to Leaf to select the waveform selector for the LFO2. The upper knob is reserved for the LFO waveform selectors (4 types).

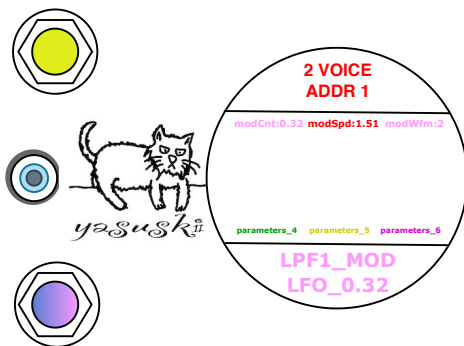
# The LPF



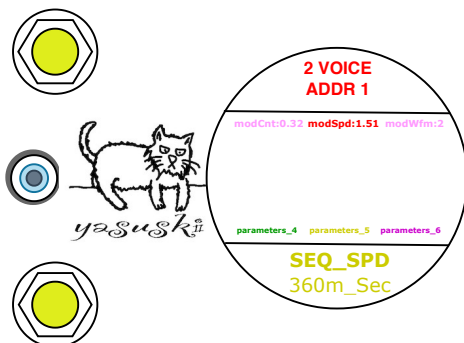
For easy accessing, the LPF has the another controllers. Click 5 times the lower knob from the neutral position (Yellow) to enter LPF\_MOD mode. The LED on the upper knob lights up in Leaf, which means the knob is accessing to the parameters for the FILTERS.



The fundamental position of the upper knob is switching to the LPF cut off frequency (Indigo). The lower knob accept the parameter of the cut off frequency for the LPF. The next, turn the upper knob until the LED turns to Pink to select the modulator for the LPF. Then the lower knob accepts to LFO1 speed.

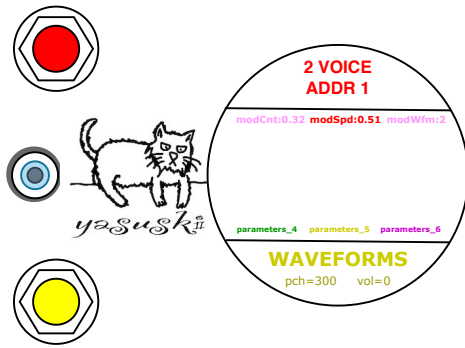


In this mode, the upper knob do not accept the address assignment and switching the voice mode.



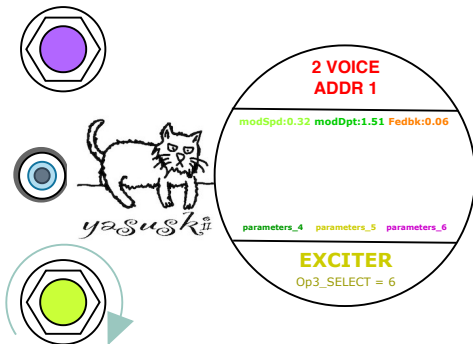
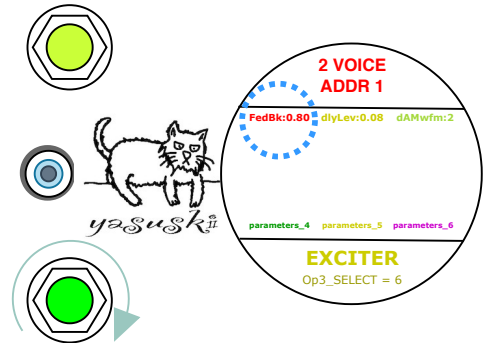
If the SEQUENCER mode is selected, this page is assigned to SEQ\_SPD. Move to EXCITER page for access to the filter controllers.

# The CombFilter

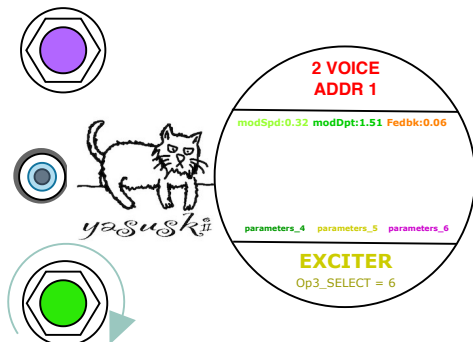


Click the lower knob from the neutral position (yellow) to enter EXCITER mode. The LED on the upper knob lights up in SkyBlue / DarkCyan / Orange color, which means the knob is accessing to the preset selector of the EXCITER.

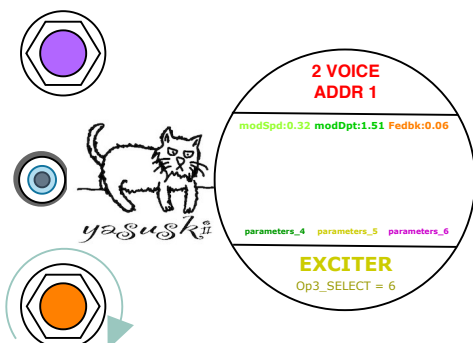
Click the upper knob to move to the Delay controller (color changes to Leaf). The lower knob changes to the effect mode selector. Turn the knob until the LED turns to Green to select the Feedback control for the comb filter. Then the upper knob accepts to feedback level for the comb filter.



Click the upper knob to move to the LPF controller (color changes to Magenta). The lower knob changes to the effect mode selector. Turn the lower knob until the LED turns to Leaf to select the flanger. Then the upper knob accepts to LFO speed.

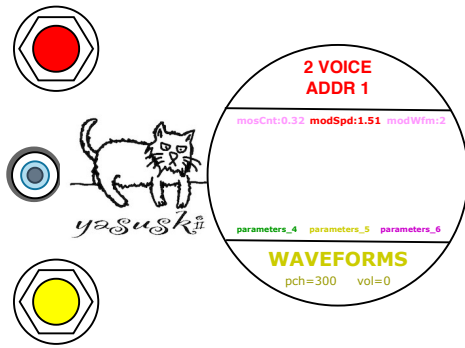


Turn the lower knob until the LED turns to Green to select the chorus. Then the upper knob accepts to LFO depth.

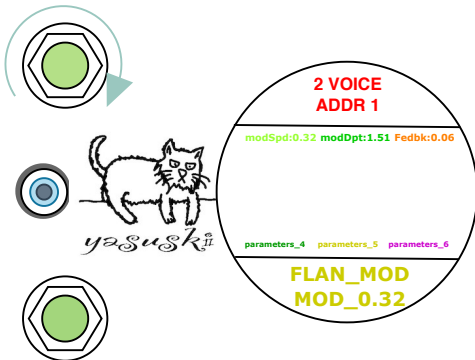


Turn the lower knob until the LED turns to Orange to select the chorus. Then the upper knob accepts to feedback of the combFilter.

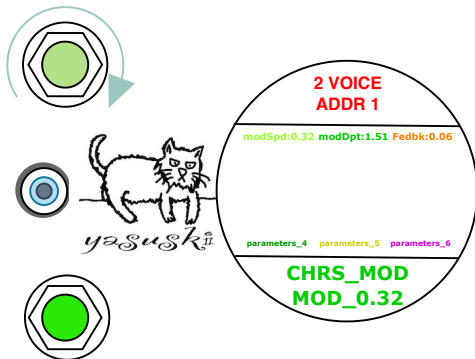
# The CombFilter (contin'd)



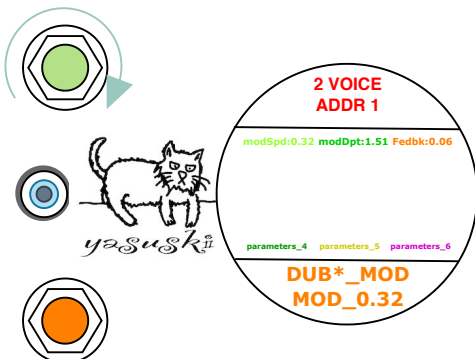
For easy accessing, the comb filter has another controllers. Click 5 times the lower knob from the neutral position (Yellow) to enter LPF\_MOD mode. The LED on the upper knob lights up in Leaf, which means the knob is accessing to the parameters for the FILTERS.



Turn the upper knob until the LED on the lower knob turns to Leaf to select the flanger. Then the lower knob accepts to LFO speed.

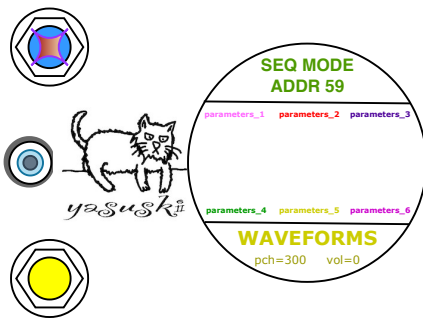


Turn the upper knob until the LED on the lower knob turns to Green to select the Chorus. Then the lower knob accepts to LFO depth.



Turn the upper knob until the LED turns to Orange to select the Dub Mode. Then the lower knob accepts to feedback of the comb filter.

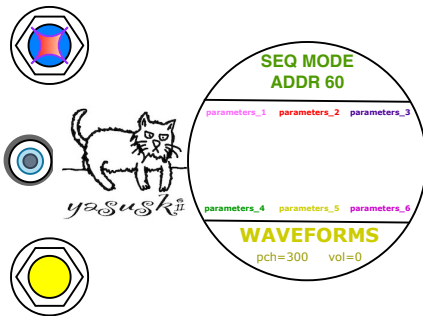
# The Sequencer



Click the upper knob 7 times from the fundamental position then the voice mode is moved to **SEQUENCER mode**.

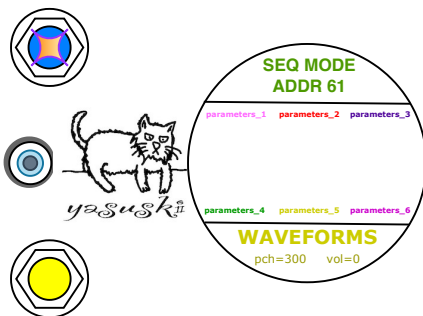
This mode has 8 sequence files.

Click the lower knob 5 times from fundamental position then the **SEQUENCE SPEED** parameter is activated.



The sequencer has eight readout channels. Each channel stores 1024 to 4096 steps of notes. The first three sequence channels have sequential arpeggiators.

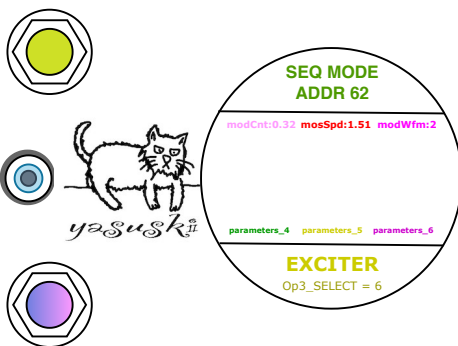
The sequence channel ADDR# 59 has a 1024-step sequential arpeggiator. The root note of the arpeggiator is driven by the pitch oscillator value.



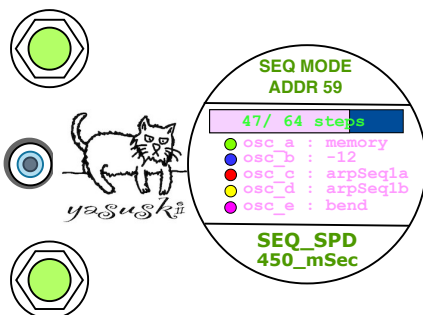
The ADDR# 60 & 61 have a 4096-step sequential arpeggiator with a different structure. The root note of the arpeggiator is driven by the pitch note described in the sequence file.

The ADDR# 63 is a simple 4096-step sequencer. The ADDR# 64 & 65 is 4096-step 3 voice chord sequencer.

The ADDR# 65 & 66 is 4096-step 3 voice chord sequencer with random note selection. \*See the programming method of the sequencer.



The controller for the filters are placed in the Exciter mode. Click the lower knob to move to Exciter mode and move to the filter controllers by clicking the upper knob (color of the LED on the top knob changes to Leaf or Magenta). Then the lower knob changes to the effect mode selector.



Click 5 times on the lower knob from the neutral position (Yellow). The LED on the lower knob lights up in Leaf, it means the knob is accessing to the parameter for the Sequence Speed.

After complete the editing work, push the top of the knob over 1 second for recording the values of the parameters.

## C#SV File Format

LaVoixski uses the original data format named the "C#SV" comma and sharp separated values.

Normally the system uses the CSV for storing the values of the parameter settings except for memory of the musical notes and HEX code.

| <u>Comma Sharp Separated Value</u> |     |     |     |
|------------------------------------|-----|-----|-----|
| c ,                                | C , | 1 , | 8 , |
| c#                                 | C#  | 1#  | 8#  |
| d ,                                | D , | 2 , | 9 , |
| d#                                 | D#  | 2#  | 9#  |
| e ,                                | E , | 3 , | 0 , |
| f ,                                | F , | 4 , | h , |
| f#                                 | F#  | 4#  | h#  |
| g ,                                | G , | 5 , | i , |
| g#                                 | G#  | 5#  | i#  |
| a ,                                | A , | 6 , | j , |
| a#                                 | A#  | 6#  | j#  |
| b ,                                | B , | 7 , | k , |

| <u>char to HEX</u> |                          |                          |
|--------------------|--------------------------|--------------------------|
| 0~f = 0x00~0x0F    |                          |                          |
| g~p = 0x10~0x19    | A~F =<br>0x1A~0x1F       |                          |
| G~P = 0x20~0x29    | v~z & "-" =<br>0x2A~0x2F |                          |
| q~u = 0x30~0x34    | Q~U = 0x35~0x39          | V~Z & "+" =<br>0x3A~0x3F |

# Sequence File HEX Format

The first 3 address on the sequencer mode has the sequential arpeggiator.

The characters on the sequence file "SequenceXXA" are addressed to the value of the controller for the Arpeggiator.

The characters are converted to the HEX numbers.

The system reads the HEX numbers then select the status of the Arpeggiator.

| <u>HEX to Arpeggiator control</u> |                  |
|-----------------------------------|------------------|
| 0xX0 ~ 0xFF                       | ARP_NOTE #1 ~ 16 |
| 0x0X                              | UP count         |
| 0x1X                              | UP/DOWN count    |
| 0x2X                              | DOWN count       |
| 0x3X                              | RANDOM           |

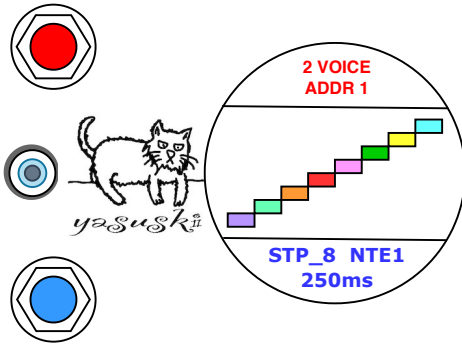
# The Arpeggiator

LaVoixski has the Arpeggiator with 32 readout channels.

The first 5 address have 32 memory steps, the second 5 have 64 steps and the next 9 have 128 steps for the arpeggio notes, and they are read from the microSD card on the Teensy .

The other notes from address #20 to 31 have 4steps and they are picked up from the memorized chord edit #1 to #12.

The last address #32 is reserved for the random notes.

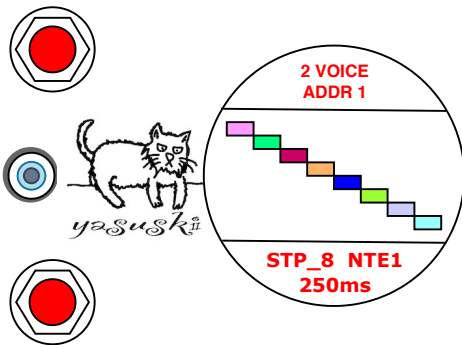


Access to the parameter “ArpNotes” on the lower knob by double clicking (Backward) 2 times from the neutral position (Yellow).

The LED on the lower knob turns Blue and ready for access to 32 note patterns. (19 preset with 13 editable notes from Chord Edit mode)

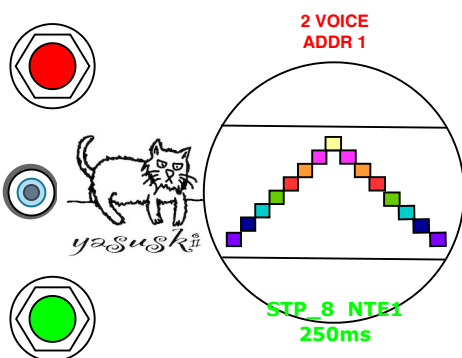
Moving to (Backward) the next parameter “ArpPtn” for selecting the playback pattern (Rise / Down / Switchback / Random) of the Arpeggiator.

The LED turns Red and ready for access to the ArpSpd controller.

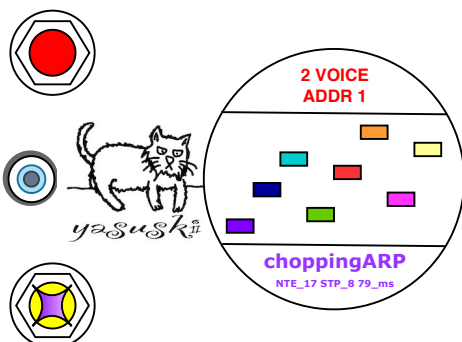


After selecting the ArpPtn for the Arpeggiator, moving to the next parameter “ArpSpd” for controlling the playback speed of the Arpeggiator.

The LED turns Green and ready for access to the speed controller.

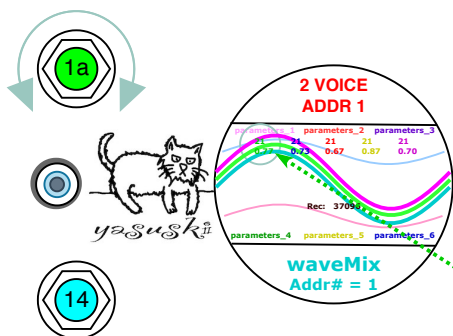


After complete the editing work, select the desired position of the scene memory of any voice mode and push the top of the knob over 1 second for recording the values of the parameters.



The Chopping Arpeggiator mode is available on address # 17 to #20. The mute actions are shown by the flick of the LEDs in violet.

# The WaveMix Mode



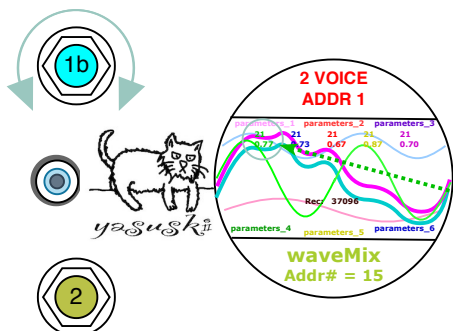
The Mix Out is a selector for the memory set which contains the combinations of the parameters, waveform selector and output level controller for the Oscillators.

The Mix Out has 32 address. The first 14 is the "Preset" group (LED = SkyBlue). Access to the Mix Out Selector on the lower knob by double clicking (backward) from the neutral position (Yellow) .

There are 17 editable channels (LED color = 2 YellowGreen / 3 Olive / 14 Purple / 4 Lavender) after the "Preset" group. The example shows the first step for editing the Oscillator group.

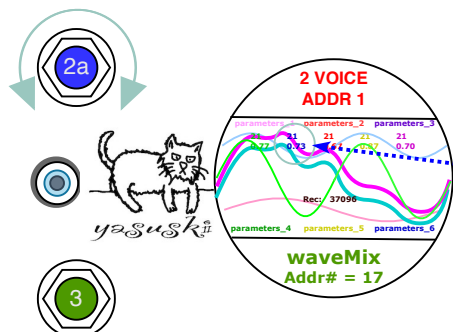
After selecting the writable address on Mix Out by the lower knob, moving to the next parameters for controlling the oscillator outputs.

Access to the parameter "OSC1 output level" appears on the upper knob (Green) then turn the knob for getting the desired volume level.

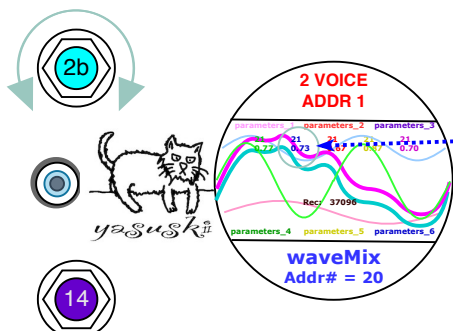


Access to the parameter "OSC1 waveform selector" by clicking the upper knob (forward) once. The LED turns to SkyBlue.

After selecting the waveform, forward the next step (Blue) by clicking the upper knob.

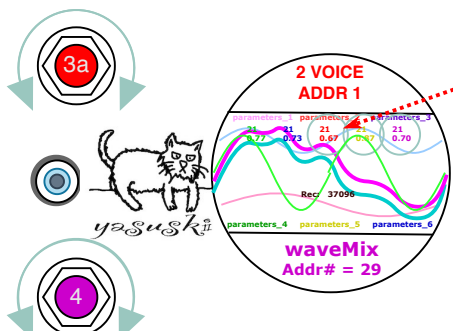


Access to the parameter "OSC2 output level" appears on the upper knob (Blue) then turn the knob for getting the desired volume level.



Access to the parameter "OSC2 waveform selector" by clicking the upper knob (forward) once. The LED turns to SkyBlue.

After selecting the waveform, forward the next step (Red) by clicking the upper knob.



Access to the parameter "OSC3 output level" appears on the upper knob (Red) then turn the knob for getting the desired volume level, and move to the next.

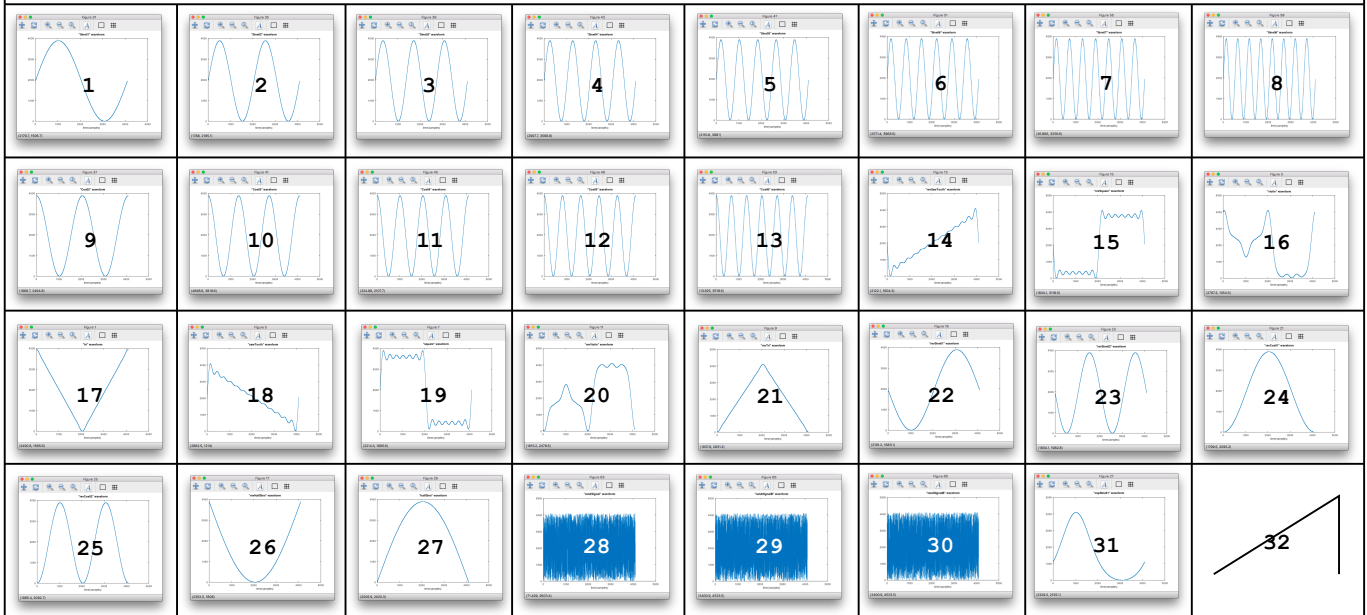
After the editing work on the other oscillators (#4 and #5) is complete, then push the lower knob over 1 second for storing the parameters.

Then move to the desired position you want to edit.

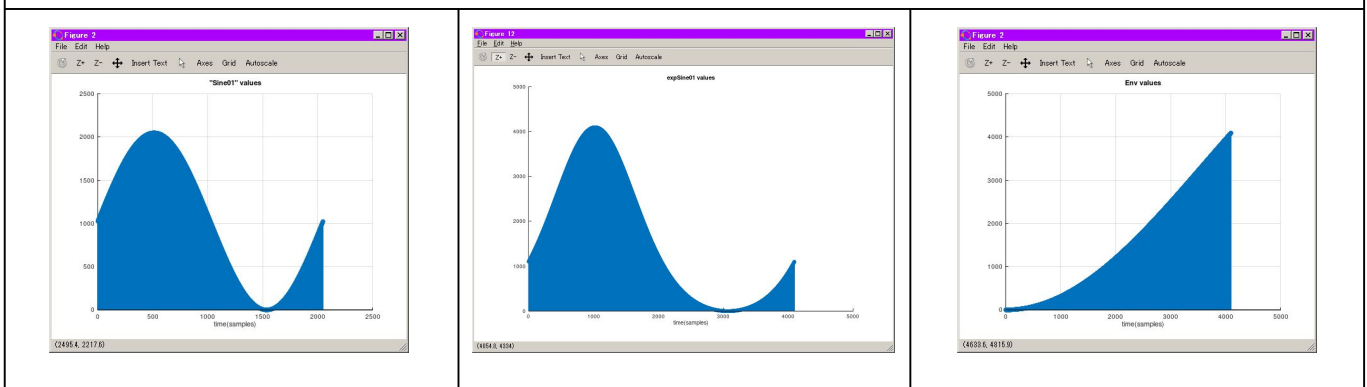
Although the "Presets" cannot edit the combinations in a realtime, the parameters are editable by rewriting the CSV in the text file.

On the other hand, Wavetable itself can edit by the GNU Octave, too. (See the chapter The Wavetables)

## Wavetable Library



## Envelope Library



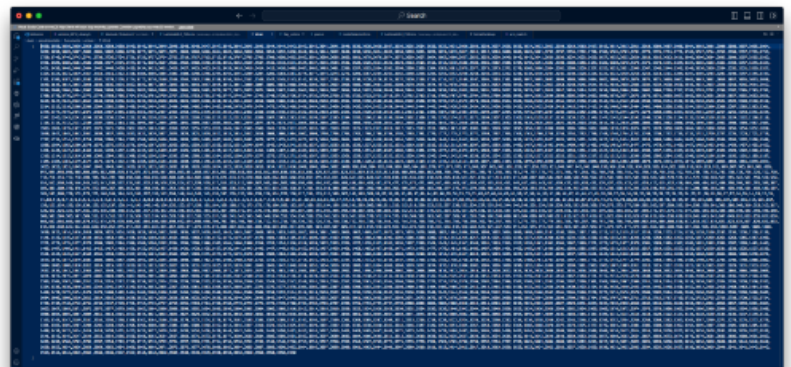
```

1 LUT_LENGTH = 2512;
2 NUM_PARTIALS = 16;
3 fs = 44100; % Hz
4 freq = 440; % Hz
5
6 tri = round((((
7     ((cos(2*pi* linspace(0,1, LUT_LENGTH)+1) * 0.95)
8     % + ((sin(4*pi* linspace(0,1, LUT_LENGTH)+1) * 0.94)
9     % + ((cos(6*pi* linspace(0,1, LUT_LENGTH)+1) * 0.111)
10    % + ((sin(8*pi* linspace(0,1, LUT_LENGTH)+1) * 0.5)
11    % + ((cos(10*pi* linspace(0,1, LUT_LENGTH)+1) * 0.04)
12    % + ((sin(12*pi* linspace(0,1, LUT_LENGTH)+1) * 0.4)
13    % + ((cos(14*pi* linspace(0,1, LUT_LENGTH)+1) * 0.02)
14    % + ((sin(16*pi* linspace(0,1, LUT_LENGTH)+1) * 0.4)
15    % + ((cos(18*pi* linspace(0,1, LUT_LENGTH)+1) * 0.012)
16    % + ((sin(20*pi* linspace(0,1, LUT_LENGTH)+1) * 0.3)
17    % + ((cos(22*pi* linspace(0,1, LUT_LENGTH)+1) * 0.008)
18    % + ((sin(24*pi* linspace(0,1, LUT_LENGTH)+1) * 0.2)
19    % + ((cos(26*pi* linspace(0,1, LUT_LENGTH)+1) * 0.004)
20    ) / 2) * 3450));
21
22 csvwrite('tri.txt', tri);
23
24 figure
25 plot(tri)
26 title('tri waveforms')
27 xlabel('time(samples)')
28
29 figure
30 stem(tri)
31 title('tri values')
32 xlabel('time(samples)')
33

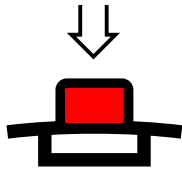
```

The waveforms are generated by the GNU Octave in CSV format. GNU Octave is a free math software which is downloadable from: <https://www.gnu.org/software/octave/>

The software works on the Windows and Linux. Unfortunately, the Mac version needs a little bit complex works for the installation.

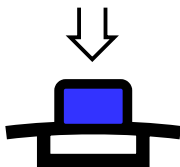


# The Looper



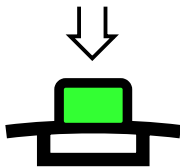
The top switch on the right is working as follows:

**Single click:** Looper Start Recording / inactivate  
**Double click:** PitchDrift--  
**Triple click:** Set Playback speed on the Looper++  
**Hold:** Start/Stop the Auto Fade mode



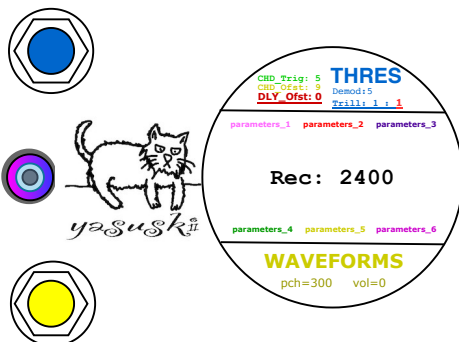
The top switch on the center is working as follows:

**Single click:** Freez Pitch  
**Double click:** PitchDrift--  
**Triple click:** Playback speed on the Looper--  
**Hold:** Recalling the Scene Memories



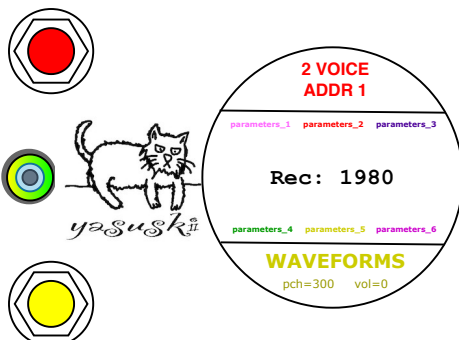
The top switch on the left is working as follows:

**Single click:** Start/Stop Arp/Seq/Looper  
**Double click:** noteNumber --  
**Triple click:** Set Playback speed on the Looper --  
**Hold:** Skipping to the the THRES mode and back



The looper capture time follows the tempo and steps of the Arpeggiator. For example, 25ms \* 4steps \* 24(magnification) = 2400ms. The magnification is set by **DLY\_Ofst** in **TRES** / **CHROMATIC** mode. The recording duration is shown at the center of the LCD, then start count down after the Looper start recording.

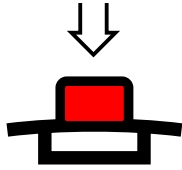
The Looper can change the 6 playback speeds by the parameter on **TRES** / **PHASE\_SHIFT** mode :  
 Normal Norm\*2 Norm\*1/2 Reverse Revs\*2 Revs\*1/2.



The top switches have the "jumping" actions in this mode. For example, jump to the **THRES** mode when Holding the **left side switch** or jump back to the **Voice mode**.

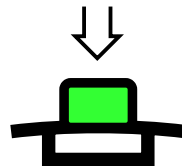
After sound recording begins, the LED on the centerpiece flashes **Magenta** and **Blue** in tempo. On the last 8 counts, the LED changes color to **Yellow** & **Green**.

# AutoFade Mode



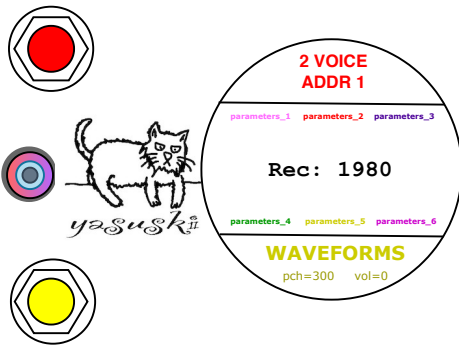
The top switch on the right is working as follows:

Hold: Start/Stop the Auto Fade mode



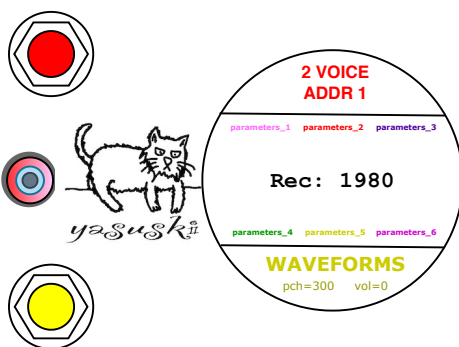
The top switch on the left is working as follows:

Single click: Start/Stop AM modulation



The AutoFade mode is activated by **HOLDING** the top switch on the right side. The fundamental status of the AutoFade is Fixed output level. Then click the top switch on the left side start AM modulation. The modulation speed is varies from the distance between the "left hand position" and the Volume antenna.

**In the Chromatic mode, the top switch on the left side does not working as start/stop switch for AM modulation.**



The LED on the eyeball switch start flashing when the AM modulation is activated.

The AM modulation is stopped when the top switch on the left side is clicked. The output level is fixed as the value of the AM modulation in the moment.

# i2c Address list

**LaVoixski receives the i2c data stream via WIFI in ESP\_NOW format or direct connection to the i2c port of the MCU.**

**The data stream is the group of the bite data packed in 2, 3, 4, 7, 29 bites.**

**The mapped index numbers are changed by how many bites are grouped with.**

**The 2 bites group is reserved for the simple switch, selector or values in low resolution.**

**The 3 bites group is reserved for handling the values in high resolution.**

**The 4 bites group is reserved for handling the mode/parameter selector with the control values.**

**The 7 bites group is reserved for the global controller which changes several parameters at the same time.**

**The 29 bites group is reserved for driving the scene memory.**

| 2 bites               |                       |                   |            |
|-----------------------|-----------------------|-------------------|------------|
| Description           | Parameter             | 1st bite          | 2nd bite   |
| Arp Patern            | arp2a to arp2e        | 0x00, 0x10 ~ 0x50 | value % 4  |
| Arp Notes ADDR        | arpNotes_aa to mode3b | 0x01, 0x11 ~ 0x51 | value % 32 |
| Waveform Selector     | addr1aa to a1         | 0x02 ~ 0x52       | value % 32 |
| muteSwitch Selector   | muteSwitchBa to Bf    | 0x03 ~ 0x53       | value % 5  |
| output#3 Selector     | op3SelectBa to Bf     | 0x04 ~ 0x54       | value % 15 |
| output level switch   | vcs                   | 0x06 ~ 0x56       | value % 2  |
| BIAS_SHIFT switch     | dst                   | 0x07 ~ 0x57       | value % 2  |
| LPF1 Mod Speed        | pot00na to nf1        | 0x08 ~ 0x58       | value      |
| Arp Speed             | arpSpdBa to Bf        | 0x09 ~ 0x59       | value      |
| Waveform Edit ADDR    | addr2b,pot19a~22a,26a | 0x0A ~ 0x5A       | value % 17 |
| Mod Feedback          | pot00ra to f1         | 0x0B ~ 0x5B       | value      |
| LPF2 Mod Speed        | pot00oa to f1         | 0x0C ~ 0x5C       | value      |
| Looper Playback Spd   | pot00q                | 0x80              | value % 6  |
| Octave Tuning         | octDown               | 0x81              | value % 3  |
| Chord Group ADDR      | chordSeq              | 0x82              | value % 48 |
| Chromatic Root Tuning | pot00x                | 0x83              | value % 12 |
| Pitch Drift Selector  | pitch_drift           | 0x84              | value % 12 |
| Mode2 Selector        | mode2                 | 0x85              | value % 9  |
| Scale Selector        | pot00y                | 0x86              | value % 32 |
| Lch Out Neg "ON"      | neg_Flag              | 0x87              | bool       |
| Start SEQ/ARP/Looper  | state4                | 0x88              | bool       |
| Parameter Selector    | mode4                 | 0x89              | value % 10 |

## i2c Address list (continue'd)

| 2 bites                                     |            |          |              |
|---------------------------------------------|------------|----------|--------------|
| Description                                 | Parameter  | 1st bite | 2nd bite     |
| Voicing Selector                            | addr2q     | 0x8A     | value % 66   |
| Auto Fade mode                              | flag_Fade  | 0x8B     | bool         |
| LFO activation / Offset ADDR on Chord group | pot0       | 0x8C     | value (% 35) |
| Freez Mode Activation                       | flag_Freez | 0x8D     | bool         |
| Looper Start REC                            | looperSW   | 0x8E     | bool         |
| Waveform Edit Parameter                     | mode6      | 0x8F     | value % 10   |
| Exciter Parameter Select                    | mode7      | 0x90     | value % 9    |
| Pitch Shift Switch1                         | up_shift   | 0x91     | bool         |
| Pitch Shift Switch2                         | up_shift2  | 0x92     | bool         |
| Pitch Shift Switch3                         | up_shift3  | 0x93     | bool         |
| Pitch Shift Switch4                         | up_shift4  | 0x94     | bool         |

| 3 bites                 |                     |           |                |
|-------------------------|---------------------|-----------|----------------|
| Description             | Parameter           | 1st bite  | 2nd & 3rd bite |
| Transition Control      | transition_aa to ae | 0x05~0x45 | value          |
| Transition Control      | transition_a1 to a8 | 0x55      | value          |
| LPF1/2 Modulation Depth | pot00ha to hf1      | 0x0B~5B   | value & 0x03FF |
| LPF1/2 Modulation Speed | pot00ia to if1      | 0x0A~5A   | value & 0x03FF |
| LPF3/4 Modulation Depth | pot00j              | 0x8B      | value & 0x03FF |
| LPF3/4 Modulation Depth | pot00k              | 0x8C      | value & 0x03FF |
| LPF1/2 Mod Waveform     | pot00l              | 0x8D      | value (% 4)    |
| LPF3/4 Mod Waveform     | pot00m              | 0x8E      | value (% 4)    |
| CombFilter Feedback     | pot00r              | 0x8F      | value & 0x007F |
| Pitch Value             | add_val_f           | 0x80      | value & 0xFFFF |
| Volume Value            | vol_16a             | 0x81      | value & 0xFFFF |

### \*Global Controller

| Transmit 2 bites                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>uint8_t data[2]{}; data[0] = 0x00; // transmit data data[1] = arp2a; // transmit data esp_now_send(slaveAddress, &amp;*data, sizeof(data));</pre> |

| Transmit 3 bites                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>uint8_t data[3]{}; data[0] = 0x05; // transmit data data[1] = (transition_3 &amp; 0x0700) &gt;&gt; 8; data[2] = transition_3 &amp; 0x00ff; esp_now_send(slaveAddress, &amp;*data, sizeof(data));</pre> |

## i2c Address list (continue'd)

| 4 bites / mode2 & 2 bite Parameters Address Direct |           |          |                               |                                                                                                                                                                    |                                                                                                                 |
|----------------------------------------------------|-----------|----------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Description                                        | Parameter | 1st bite | 2nd bite<br>select Voice mode | 3rd bite<br>select 2bite parameter                                                                                                                                 | 4th bite                                                                                                        |
| Voice mode Direct                                  | mode2     | 0x85     | value2 % 9                    | 0x03 = addr1q<br>0x04 = muteSwitchB<br>0x05 = op3SelectB<br>0x07 = transitionSelB<br>0x08 = BIAS_SHIFT<br>0x09 = flag_LPF_mod<br>0x01 = arpPtnB<br>0x02 = arpNoteB | Value4 % 32<br>Value4 % 5<br>Value4 % 15<br>Value4 % 2<br>Value4 % 4<br>Value4 % 5<br>Value4 % 4<br>Value4 % 32 |
|                                                    |           |          |                               |                                                                                                                                                                    |                                                                                                                 |

| 5 bites / mode2 & 3 bite Parameters Address Direct |                      |          |                               |                                      |                      |
|----------------------------------------------------|----------------------|----------|-------------------------------|--------------------------------------|----------------------|
| Description                                        | Parameter            | 1st bite | 2nd bite<br>select Voice mode | 3rd bite<br>select 2bite parameter   | 4/5th bite           |
| Voice mode Direct                                  | mode2                | 0x85     | value2 % 9                    | 0x06 = transitionB<br>0x0A = arpSpdB | Value4/5<br>Value4/5 |
| Pitch Volume Value set                             | add_val_f<br>vol_16a | 0x80     | add_val_f MSB                 | add_val_f LSB                        | vol_16a<br>MSB+LSB   |

| 7 bites / Global Controller |                                                                                              |          |                 |
|-----------------------------|----------------------------------------------------------------------------------------------|----------|-----------------|
| Description                 | Parameter                                                                                    | 1st bite | 2nd to 7th bite |
| Delay Feedback GC           | pot000a<br>pot000b<br>pot000c<br>pot000d<br>pot000e<br>pot000f1                              | 0xFC     | bool            |
| Mute Switch GC              | muteSwitchBa<br>muteSwitchBb<br>muteSwitchBc<br>muteSwitchBd<br>muteSwitchBe<br>muteSwitchBf | 0xFA     | Values + 1      |

### \*Global Controller

| Transmit 5 bites                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>uint8_t data[5]{}; data[0] = 0x85;      // transmit data data[1] = mode2; data[2] = 0x06; data[3] =(transition_4 &amp; 0x0700) &gt;&gt; 8; data[4] = transition_4 &amp; 0x00ff; esp_now_send(slaveAddress, &amp;*data, sizeof(data));</pre> |

| Transmit 7 bites                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>uint8_t data[7]{}; data[0] = 0xFC;    // transmit data data[1] = HIGH; data[2] = HIGH; data[3] = HIGH; data[4] = HIGH; data[5] = HIGH; data[6] = HIGH; esp_now_send(slaveAddress, &amp;*data, sizeof(data));</pre> |

## i2c Address list (continue'd)

Receive 29 bites for Recalling the Scene Memory

```
case 0xF0: // 2 voice mode
    transition_aa = (value2 << 8) + value3; // tra_a      = transition_aa;
    attack1       = (value4 << 8) + value5; // atk1        = attackB1;
    LFO1          = (value6 << 8) + value7; // lfom1       = LFOB1;
    pot00ha       = (value8 << 8) + value9; // lpf_a       = pot00ha;
    pot00ia       = (value10 << 8) + value11; // lpfSpd_a    = pot00ia;
    pot00ja       = (value12 << 8) + value13; // lpf2_a      = pot00ja;
    pot00ka       = (value14 << 8) + value15; // lpf2Spd_a   = pot00ka;
    pot00la       = (value16 << 8) + value17; // lpfWf_a     = pot00la;
    pot00ma       = (value18 << 8) + value19; // lpf2Wf_a    = pot00ma;
    pot00ra       = value20;                // fdbk_a      = pot00ra;
    pot00sa       = value21;                // cdpt_a      = pot00sa;
    pot00na       = value22;                // lpfSW_a     = pot00na;
    pot00oa       = value23;                // lpf2SW_a    = pot00oa;
    exB1          = value24;                // ex1         = exB1;
    op3SelectBa   = value25;                // op3_a       = op3SelectBa;
    arp2a         = value26;                // arptn_a     = arp2a;
    arpSpdBa      = value27;                // spd_a       = arpSpdBa;
    arpNotes_aa   = value28;                // apnote_a    = arpNotes_aa;
    addr1aa       = value29;                // waves_a     = addr1aa;
    break;

case 0xF1: // 3 voice mode
    transition_ab = (value2 << 8) + value3; // tra_b      = transition_ab;
    attack2       = (value4 << 8) + value5; // atk2       = attackB2;
    LFO2          = (value6 << 8) + value7; // lfom2      = LFOB2;
    pot00hb       = (value8 << 8) + value9; // lpf_b      = pot00hb;
    pot00ib       = (value10 << 8) + value11; // lpfSpd_b   = pot00ib;
    pot00jb       = (value12 << 8) + value13; // lpf2_b     = pot00jb;
    pot00kb       = (value14 << 8) + value15; // lpf2Spd_b  = pot00kb;
    pot00lb       = (value16 << 8) + value17; // lpfWf_b    = pot00lb;
    pot00mb       = (value18 << 8) + value19; // lpf2Wf_b   = pot00mb;
    pot00rb       = value20;                // fdbk_b     = pot00ra;
    pot00sb       = value21;                // cdpt_b     = pot00sa;
    pot00nb       = value22;                // lpfSW_b    = pot00nb;
    pot00ob       = value23;                // lpf2SW_b   = pot00ob;
    exB2          = value24;                // ex1        = exB2;
    op3SelectBb   = value25;                // op3_b      = op3SelectBb;
    arp2b         = value26;                // arptn_b    = arp2b;
    arpSpdBb      = value27;                // spd_b      = arpSpdBb;
    arpNotes_ab   = value28;                // apnote_b   = arpNotes_ab;
    addr1ab       = value29;                // waves_b    = addr1ab;
    break;
```

## i2c Address list (continue'd)

Receive 29 bites for Recalling the Scene Memory

```
case 0xF2: // 4 voice mode
    transition_ac = (value2 << 8) + value3; // tra_c      = transition_ac;
    attack3       = (value4 << 8) + value5; // atk3       = attackB3;
    LFO3          = (value6 << 8) + value7; // lfom3      = LFOB3;
    pot00hc       = (value8 << 8) + value9; // lpf_c      = pot00hc;
    pot00ic       = (value10 << 8) + value11; // lpfSpd_c   = pot00ic;
    pot00jc       = (value12 << 8) + value13; // lpf2_c     = pot00jc;
    pot00kc       = (value14 << 8) + value15; // lpf2Spd_c  = pot00kc;
    pot00lc       = (value16 << 8) + value17; // lpfWf_c    = pot00lc;
    pot00mc       = (value18 << 8) + value19; // lpf2Wf_c   = pot00mc;
    pot00rc       = value20;                // fdbk_c     = pot00ra;
    pot00sc       = value21;                // cdpt_c     = pot00sa;
    pot00nc       = value22;                // lpfSW_c    = pot00nc;
    pot00oc       = value23;                // lpf2SW_c   = pot00oc;
    exB3          = value24;                // ex3        = exB3;
    op3SelectBc   = value25;                // op3_c      = op3SelectBc;
    arp2c         = value26;                // arptn_c    = arp2c;
    arpSpdBc      = value27;                // spd_c      = arpSpdBc;
    arpNotes_ac   = value28;                // apnote_c   = arpNotes_ac;
    addr1ac       = value29;                // waves_c    = addr1ac;
    break;

case 0xF3: // 5 voice mode
    transition_ad = (value2 << 8) + value3; // tra_d      = transition_ad;
    attack4       = (value4 << 8) + value5; // atk4       = attackB4;
    LFO4          = (value6 << 8) + value7; // lfom4      = LFOB4;
    pot00hd       = (value8 << 8) + value9; // lpf_d      = pot00hd;
    pot00id       = (value10 << 8) + value11; // lpfSpd_d   = pot00id;
    pot00jd       = (value12 << 8) + value13; // lpf2_d     = pot00jd;
    pot00kb       = (value14 << 8) + value15; // lpf2Spd_d  = pot00kd;
    pot00ld       = (value16 << 8) + value17; // lpfWf_d    = pot00ld;
    pot00md       = (value18 << 8) + value19; // lpf2Wf_d   = pot00md;
    pot00rd       = value20;                // fdbk_d     = pot00ra;
    pot00sd       = value21;                // cdpt_d     = pot00sa;
    pot00nd       = value22;                // lpfSW_d    = pot00nd;
    pot00od       = value23;                // lpf2SW_d   = pot00od;
    exB4          = value24;                // ex4        = exB4;
    op3SelectBd   = value25;                // op3_d      = op3SelectBd;
    arp2d         = value26;                // arptn_d    = arp2d;
    arpSpdBd      = value27;                // spd_d      = arpSpdBd;
    arpNotes_ad   = value28;                // apnote_d   = arpNotes_ad;
    addr1ad       = value29;                // waves_d    = addr1ad;
    break;
```

## i2c Address list (continue'd)

Receive 29 bites for Recalling the Scene Memory

```
case 0xF4: // Chord Edit mode
    transition_ae = (value2 << 8) + value3; // tra_e      = transition_ae;
    attack5       = (value4 << 8) + value5; // atk5        = attackB5;
    LFO5          = (value6 << 8) + value7; // lfom5        = LFOB5;
    pot00he       = (value8 << 8) + value9; // lpf_e        = pot00he;
    pot00ie       = (value10 << 8) + value11; // lpfSpd_e     = pot00ie;
    pot00je       = (value12 << 8) + value13; // lpf2_e       = pot00je;
    pot00ke       = (value14 << 8) + value15; // lpf2Spd_e    = pot00ke;
    pot00le       = (value16 << 8) + value17; // lpfWf_e      = pot00le;
    pot00me       = (value18 << 8) + value19; // lpf2Wf_e     = pot00me;
    pot00re       = value20;                // fdbk_e       = pot00ra;
    pot00se       = value21;                // cdpt_e       = pot00sa;
    pot00ne       = value22;                // lpfSW_e      = pot00ne;
    pot00oe       = value23;                // lpf2SW_e     = pot00oe;
    exB5          = value24;                // ex5          = exB5;
    op3SelectBe   = value25;                // op3_e        = op3SelectBe;
    arp2e         = value26;                // arptn_e      = arp2e;
    arpSpdBe      = value27;                // spd_e        = arpSpdBe;
    arpNotes_ae   = value28;                // apnote_e     = arpNotes_ae;
    addr1ae       = value29;                // waves_e      = addr1ae;
    break;

case 0xF5: // Sequencer mode
    transition_a1 = (value2 << 8) + value3; // tra_a1      = transition_af1;
    attack6       = (value4 << 8) + value5; // atk6        = attackB6;
    LFO6          = (value6 << 8) + value7; // lfom6        = LFOB6;
    pot00hf1      = (value8 << 8) + value9; // lpf_f1       = pot00hf1;
    pot00if1      = (value10 << 8) + value11; // lpfSpd_f1    = pot00if1;
    pot00jf1      = (value12 << 8) + value13; // lpf2_f1      = pot00jf1;
    pot00kf1      = (value14 << 8) + value15; // lpf2Spd_f1   = pot00kf1;
    pot00lf1      = (value16 << 8) + value17; // lpfWf_f1     = pot00lf1;
    pot00mf1      = (value18 << 8) + value19; // lpf2Wf_f1    = pot00mf1;
    pot00rf1      = value20;                // fdbk_f1      = pot00ra;
    pot00sf1      = value21;                // cdpt_f1      = pot00sa;
    pot00nf1      = value22;                // lpfSW_f1     = pot00nf1;
    pot00of1      = value23;                // lpf2SW_f1    = pot00of1;
    exB6          = value24;                // ex6          = exB6;
    op3SelectBf1  = value25;                // op3_f1       = op3SelectBf1;
    arp2f1        = value26;                // arptn_f1     = arp2f1;
    arpSpdBf1     = value27;                // spd_f1       = arpSpdBf1;
    arpNotes_af1  = value28;                // apnote_f1    = arpNotes_af1;
    addr1af1      = value29;                // waves_f1     = addr1af1;
    break;
```

## i2c Address list (continue'd)

## ESP\_NOW Transmitter

```
#include <esp_now.h>
#include <WiFi.h>
esp_now_peer_info_t peerInfo;

void setup()
{
  WiFi.mode(WIFI_STA);
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  memcpy(peerInfo.peer_addr, slaveAddress, 6);
  peerInfo.channel = 0;
  peerInfo.encrypt = false;

  if (esp_now_add_peer(&peerInfo) != ESP_OK){    // Add peer
    Serial.println("Failed to add peer");
    return;
  }
}
```

Additional 2 bite address

[illegible]

## i2c Address list (continue'd)

### ESP\_NOW Receiver / i2c Transponder

```
#include <esp_now.h>
#include <WiFi.h>
#include // Include the Wire library for I2C communication
#define WIFI_CHANNEL 0

volatile uint8_t addr01[128] = {};
volatile uint8_t addr02[128] = {};
uint8_t buffer[128] = {};
uint8_t bites = 0;

void onReceive(const uint8_t * mac, const uint8_t *data, int len) {

    Wire.beginTransaction(0x8);

    memcpy(&buffer[0], data, len);

    for (int i = 0; i < len; i++) {
        Wire.write(data[i]);
        addr01[i] = data[i];
        addr02[i] = addr01[i];
    }

    Wire.endTransmission();

    delay(1);
}

void setup()
{
    Wire.begin(PIN_SDA, PIN_SCL);

    Wire.setClock(1000000); // fSCL = 1MHz

    WiFi.mode(WIFI_STA); // Set ESP32 as a Wi-Fi Station

    if (esp_now_init() != ESP_OK) { // Initilize ESP-NOW
        return;
    }

    esp_now_register_recv_cb(onReceive); // Register callback function

    delay(1);
}
```

**\*Program works only with Teensyduino 1.57 and ESP tools 2.0.5.**

# i2c Wireless Remote Controller

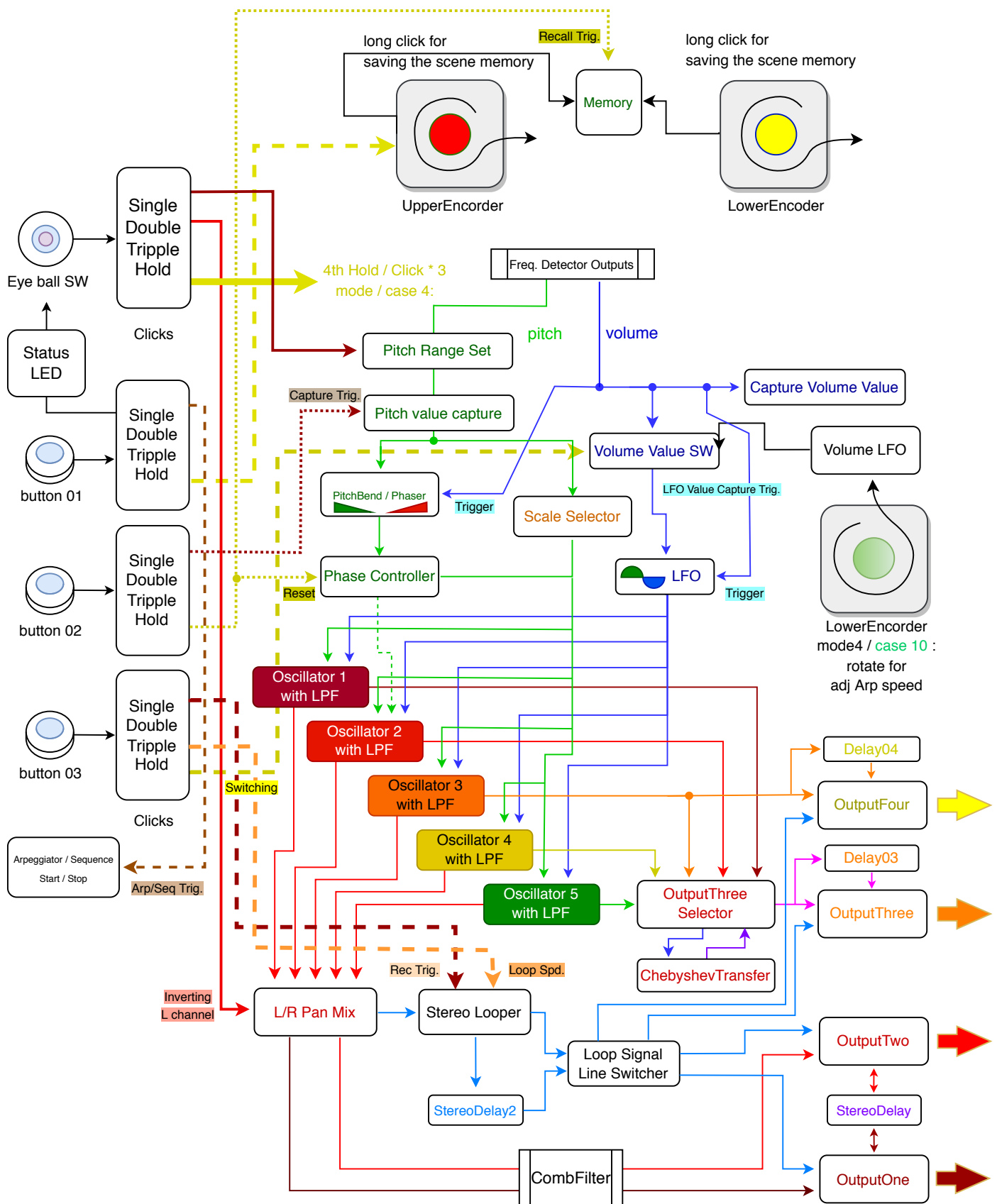
| parameter switch 7                                                 |                   |                               |             |
|--------------------------------------------------------------------|-------------------|-------------------------------|-------------|
| Description                                                        | i2c address       | HOLD                          | i2c address |
| 1st: Chromatic Mode %32                                            | 0x86              | <b>NORM/CHRM</b>              | <b>0xFA</b> |
| 2nd: Root Note Set %12                                             | 0x83              | Set to "0"                    | 0x83 = 0;   |
| 3rd: Mode 2 Selector %9                                            | 0x85              | AutoFade                      | 0x8B = bool |
| 4th ReadOffset for ChordSeq %16                                    | 0x82              | Bank Change<br>with no offset | +0/+16/+32; |
| 5th: Loop Record / PbSpd %6                                        | 0x8E/0x80         | PbSpd =<br>Normal             | 0x80 = 0;   |
| 6th: Arp/Seq/Lpr: Start/Stop<br>/ <b>delayFeedbackLoop: ON/OFF</b> | 0x88/ <b>0xFB</b> | LchNegative                   | 0x87 = bool |
| 7th: octDown %5                                                    | 0x81              | PitchFreez                    | 0x8D = bool |
| 8th: pitchDrift(Bender) %12                                        | 0x84              | set to "0"                    | 0x84 = 0;   |

| voice mode switch 9 |             |                  |       |
|---------------------|-------------|------------------|-------|
| Mode 2              | i2c address | 2nd bite (value) | Range |
| #1 Pitch            | none        | none             | none  |
| #2 Volume           | none        | none             | none  |
| #3 2 voice          | 0x0A        | addr2b           | 1~16  |
| #4 3 voice          | 0x1A        | pot19a           | 17~24 |
| #5 4 voice          | 0x2A        | pot20a           | 25~32 |
| #6 5 voice          | 0x3A        | pot21a           | 33~40 |
| #7 Chord Edit       | 0x4A        | pot22a           | 41~58 |
| #8 Sequencer        | 0x5A        | pot26a           | 59~66 |
| #9 THRES mode       | none        | none             | none  |

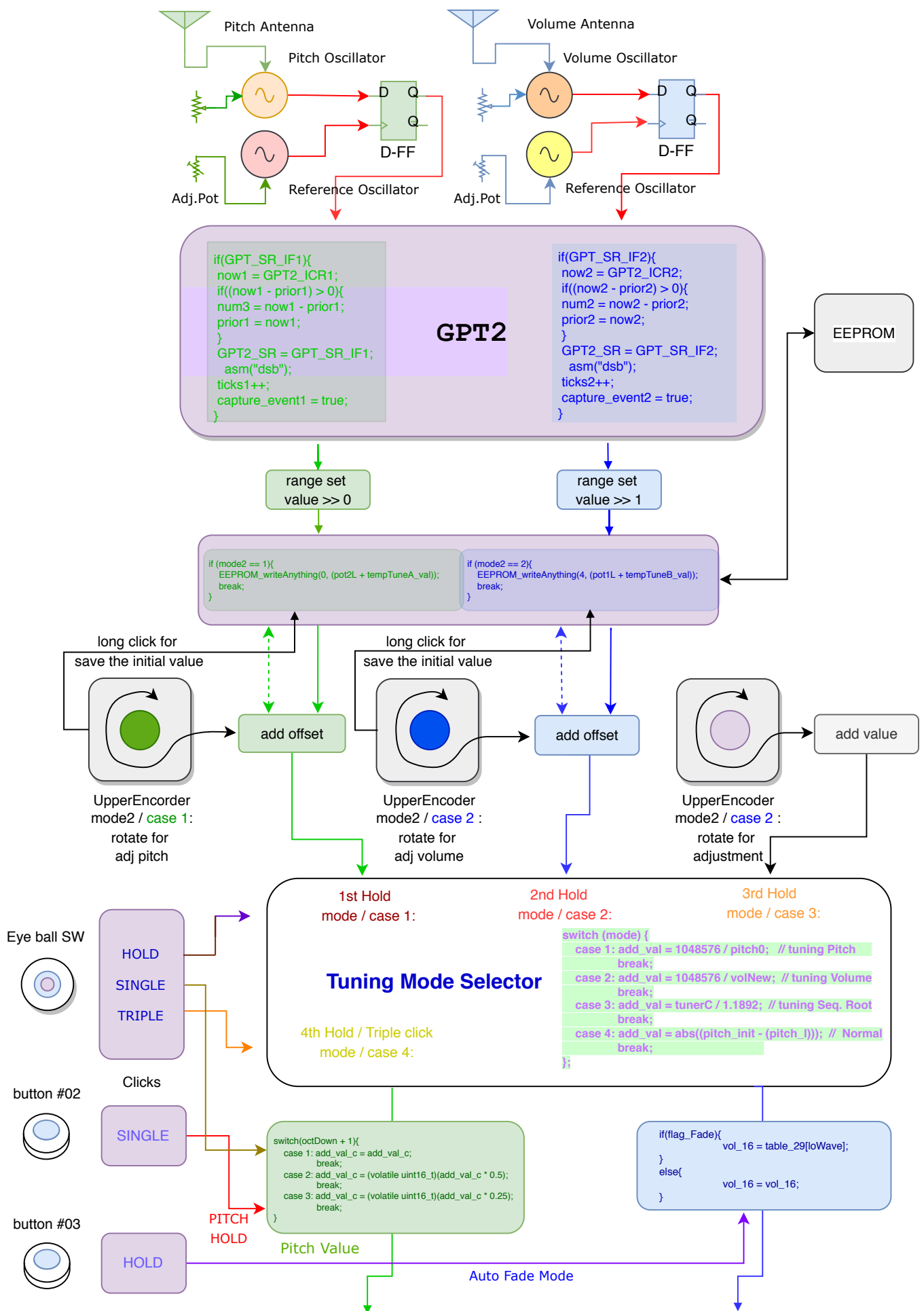
## \*Global Controller

| Transmit 2 bites                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>uint8_t data[2]{}; data[0] = 0x83; // transmit data data[1] = pot00x; // transmit data esp_now_send(slaveAddress, &amp;*data, sizeof(data));</pre> |

| Transmit 7 bites                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>uint8_t data[7]{}; data[0] = 0xFA; // transmit data data[1] = 5; data[2] = 5; data[3] = 5; data[4] = 5; data[5] = 5; data[6] = 5; esp_now_send(slaveAddress, &amp;*data, sizeof(data));</pre> |

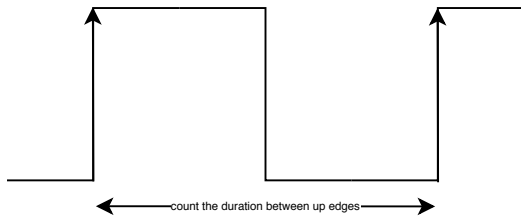


The signal flow around the audio & parameters (1st Layer)



The signal flow around the Pitch/Volume Tuning System

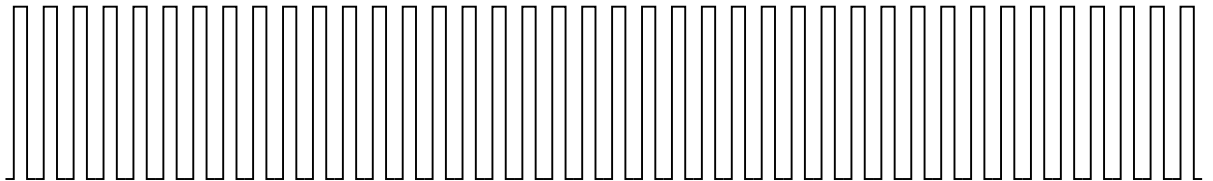
# Linear interpolation



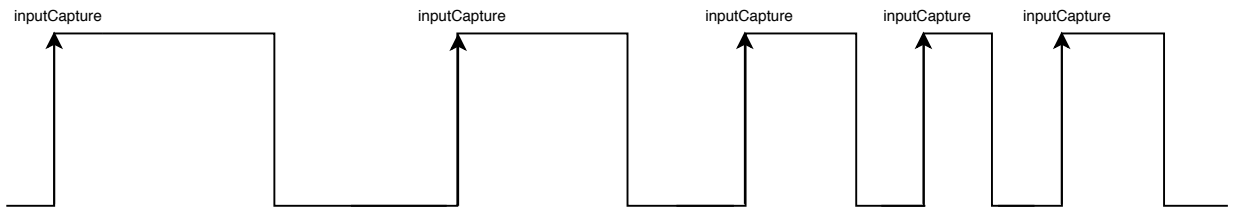
```

if(capture_event1){
    cli();
    pitch0 = num3;
    pitchNew = ((pitch0 >> b_shift_p) & 0xffff) ; // divide here
    average += pitchNew;
    pitchNew = (int32_t) average();
    flag_pitch2 = true;
    flag_pitch();
    asm("dsb");
    capture_event1 = false;
    sei();
}
    
```

Sampling Clock  
(LRCK)



Demodulated  
signal source

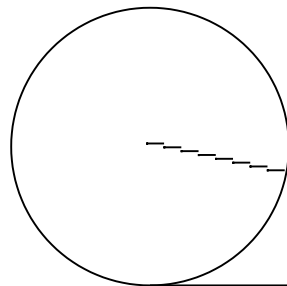


Results from  
inputCapture

```

if(step_pc > step_p){
    add_val_c = add_val_c;
}
else{
    interp_px = abs(interp_p) + interp_px;
    step_pc++;
}
if(pMag == HIGH){
    add_val_c = add_val_b + (int16_t)interp_px;
}
else{
    add_val_c = add_val_b - (int16_t)interp_px;
}
    
```

Linearized  
signal



```

interp_p = ((int16_t(add_val_b - add_val_a) * 1000) / step_p) * 0.001;
if(add_val_b < add_val_a){
    pMag = HIGH;
}
else{
    pMag = LOW;
}
    
```

