

[enhancement] Gate SEQ 64 #13

[Edit](#)
[New issue](#)


Eurikon opened this issue on Aug 21, 2018 · 55 comments



Eurikon commented on Aug 21, 2018

Hi there, i would like to propose an enhancement for GS64, if possible.
The following conversation with Marc has already started on Facebook direct message, so i will continue it here:

OP:

If some if the new gate technique (PS16/32) could get to gate-seq 64 , then this would be so great for drums as well. Obviously not legato, but those triplets, duo, dotted stuff. That would be awesome. What do you think? Or save it for a new dedicated drum sequencer, one that could have also the polyrhythm/polymeter different length/clock per sequence track?

Marc:

Interesting idea! I'll take a look for GS64, and if there is an easy way to add it, it would make sense. Thanks for suggesting it! For the different lengths per track, the problem with this is that it seems incompatible with song mode. When a short sequence finishes, it goes to the next phrase of the song, but the longer one would not follow. Perhaps that would require a whole new design, without song mode?

OP:

hmmm ok i see your point, but how did Yamaha do it in their RS7000 , i used to make phrases with different lengths , and it had song mode in it.

And actually i want to push it a little further, where for example there would be also a optional 2nd clock input for when in 2x32 steps mode.

If the GUI doesn't have the space for that, then it could be a builtin DIV/MULT for the 2nd clock based on the 1st incoming clock.

That could be selected through the existing rotation knob.

Marc:

Perhaps with another internal run mode (like FWD, REV, PPG, ... FW3, FW4), we could have one that divides the clock for one of the 2x32 channels, and plays the other channel twice, such that we stay in the same phrase. So same interface, single clock, and it could be a first step.

OP:

upon thinking a bit, there seem to be 2 things that jump into mind. There is in this last decade or 2 something that has gotten the name: scenes. Which i don't recall in old school sequencers. there i remember pattern chaining (Roland), which later got added the term: phrase linking by Yamaha.

Maybe a solution to this "problem" lies in : scenes, where a scene is a snapshot, and a song is built out of scenes.

Marc:

Interesting!

Assignees



No one—assign yourself

Labels



feature request

Projects



None yet

Milestone



No milestone

3 participants



★ Pin issue ⓘ

  **MarcBoule** added the `feature request` label on Jan 22



MarcBoule commented on Feb 25

Owner

To follow up this thread, I have planned a new GateSequencer in the same lineup as Foundry, so it would be more powerful, and with separate clocking of its 4 tracks, it will be more flexible. In fact, I will likely derive the new gate sequencer from the code of Foundry. To be continued :-)

 1



joopvdl commented on Feb 25

Hi, can I jump in in this discussion? Apologies if it feels like breaking in.
Separate clocking, separate lengths, some random function like the Aepelzen has, that would be wonderful!
I am not a big user of song mode, I prefer the sequencers to be switchable individually (so seq A can switch from 1 to 2 etc with a pulse or a voltage, B can do the same, etc.) (of course, they would need to be have separate switching inputs, and if no input is detected, all normalize to A).
The reason I don't use song mode much is because it's much less flexible.



MarcBoule commented on Feb 25

Owner

All good Joop, this is exactly what is needed: for us to talk about the tools, so the more feedback the better!



joopvdl commented on Feb 25

Great!
Could you describe the general initial idea of what you think the sequencer/module would do?
I understand you are thinking about Gateseq's big brother, is that correct?
Are there features that you already decided on?



MarcBoule commented on Feb 25

Owner

Yes, GateSeq64's big brother, and Foundry's cousin, so to speak. To get the general idea, start with Foundry, remove the keypad and octave lights, remove the CV outputs, but keep CV2 (which would be called just CV) since we want to be able to modulate drums and/or filters with our percussions. With the space left we would have room for the same type of led/button array we have in GS64 for multitrack display. So we would have 4 independant tracks that can be separately clocked. Not fully settled on this, but it's the plan at this time.

Just as described here it's not a huge improvement, but there is room for some good extras, like Latif's suggestions for offering builtin envelope types for the CV out (not the same thing as the gate types of the advanced gate mode). I'll need to get started with 1.0 porting soon I think, so it would likely be after that.



joopvdl commented on Feb 25

numbers.



Eurikon commented on Feb 25

Author

..builtin envelope types for the CV out (not the same thing as the gate types of the advanced gate mode). I'll need to get started with 1.0 porting soon I think, so it would likely be after that. <

Not necessarily builtin envelope types Marc, (don't know how that could be achieved and used per 1 step anyway) plain CV value out per step would suffice.

The user can then control velocity (vca), vcf cut[off, or what ever with that for their drums/percussion voice.

:)



joopvdl commented on Feb 25

And the user can randomise this value, is my suggestion (so, a random button) (-5 to +5, I would say)



MarcBoule commented on Feb 25

Owner

@NRG70 Oh, sorry for misinterpreting, if plain CV is ok, then no problem :-)

@joopvdl CV randomized value would be interesting perhaps, but in your view, the -5 to +5 would control what aspect of the randomized CV? In other words, can you elaborate on what the button would do? In my view, I would think that two buttons would be good, a low bound and a high bound, and the random would choose a random voltage within these bounds, but there could be other ways to do it.



joopvdl commented on Feb 25

Yes, that would even be better, but it would need 3 buttons (rnd, low bound, high bound). I thought just generate a random voltage, that you can always control with an attenuverter. So you need just 1 button :)

 1



MarcBoule commented on Feb 25

Owner

in seq mode I would love to be able to switch/choose sequence numbers

I am starting to at least consider this, but in order for the idea to make some progress, I need to know how you would see an answer to the concern I raised in the other thread:

For the inability to directly play a sequence, it was a design choice given the multitrack nature of Foundry. If we were to have the previous behavior of the PhraseSeq, when we would move to a give sequence in one track, **the same sequence number may not even be defined in another track, so those would play garbage**. There is only one edit head that controls editing on all trakcs. I tried it with four editing heads, and it was hell!



joopvdl commented on Feb 25

If a sequence number is not defined, just have it output nothing. Is that hard to implement?

The feature that Joop is proposing, sounds like pattern-chaining.

Which is actually a live play mode,
that is neither: manual on the fly sequence arranging,
nor song mode.
It is a play mode on its own.

I remember bugging Yamaha about this to add it to the RS7000, back in the days. As phrase chaining.

Unfortunately they didn't see the need for it, and finally dropped support for the machine all together.
All of its users, forums, and user base.
Abandoning groove-box style sequencers, up till today.

 joopvdl commented on Feb 25

Actually I meant the manual arranging, but this sounds interesting too ...
...

 MarcBoule commented on Feb 25 • edited ▼

Owner

If a sequence number is not defined, just have it output nothing. Is that hard to implement?

Not a problem really, but that would be just one of the consequences of using the original PhraseSequencer approach in Foundry. I'm using this extreme example, but even if sequences are defined in all tracks, then when there is no correlation between sequence numbers in the tracks, they would always be forced to play together in SEQ mode. If we have chords, then no problem, but that is not always the case.

Another consequence would be that in Foundry we would loose the ability to modify sequences while the sequencer is running its song (as it stands now, Foundry always runs the song, whether we are in SEQ mode or in SONG mode). This loss may not seem like much, but since in SEQ mode we could only run one sequence number across all tracks, if we want to hear a melody in track A that is on sequence 10 simultaneously with a repeating bass line in track D that has only 1 sequence, we would be forced to go to song mode. But in song mode we would not be able to edit the sequences, so we loose that.

All in all, I would say it's a balancing of compromises, but I'm open to do what is most useful for users. If the advantages of making the change outweigh the drawbacks, I would perhaps consider modifying Foundry to make it work like the other Phrase Sequencers.

So in light of all this Joop, if you still feel strongly about the modification, please let me know. And Latif, if you have a preference also, please let me know too. Thanks guys!

 joopvdl commented on Feb 26

I see.

What about this:

in Song mode, you can define when a track (say, track A) is playing a sequence (say, number 6) (I am not a number!) (that's a Maiden reference, couldn't resist).

So, in a way, Foundry picks the sequence number it will playback. And that's independent of the other sequencers.

Couldn't there be a way to implement a sort of manual override to this?

something. If that makes any sense :)



Eurikon commented on Feb 26

Author

Well Marc for me the mode SEQ/SONG work fine as they are.
But I need to understand 1st what Joop is asking.

Joop do you mean to say that you wish in SEQ mode. That while seq 1 is running, that you can rotate to seq 5, but the device then waits till the end of the sequence, and then switches to 5?



joopvdl commented on Feb 26

That's a great idea!

I usually send out a trigger or a voltage at the end of the sequence to switch. But in Foundry, unless I am mistaken, you can only select a sequence to edit it, not to play it.



Eurikon commented on Feb 26 • edited ▾

Author

If that is the only issue. then there needs nothing to change in the structure of any of the sequencers for this.

It looks more like an option to be added to the edit mode,
so they wait till end of current sequence before to jump to a newly chosen sequence (this should be optional).

This way certain users can play with numerous combinations of sequence number variations to see and hear what they want to do, before they go next to program them in song mode.

However this does not apply to Foundry, which has a different way of doing things.



joopvdl commented on Feb 26

So you will be able to switch sequences while in play mode, and to edit them?



MarcBoule commented on Feb 26

Owner

This is getting confusing guys! :-) I'll need to digest all this, as the potential changes are really non-trivial.



Eurikon commented on Feb 26

Author

:D

from what i understand, Joop means that he would like the sequence to play out till the end before jumping to another chosen sequence, in edit mode.

ofcourse you need to decide whether this is feasible for you or not Marc.

 1



joopvdl commented on Feb 26

confusion.



MarcBoule commented on Feb 26

Owner

All good! It will surely not be in the next update, as I now have two bugs that I want to stomp, so the next update should be coming pretty soon.



Eurikon commented on Feb 26

Author

Ah ok, then i misunderstood Joop, and not actually getting what he suggested.
Have a good one ironing out bugs Marc :)



joopvdl commented on Feb 26

Yes. But I am suggesting that now, too :)
Love to be able to contribute, Marc.



2



MarcBoule commented on Feb 26

Owner

The most appreciated contributions are that you (both) take the time and have the patience to keep the discussion going :-)
(Unfortunately the Maiden reference was above head, as the most metal I ever went was early Metallica :-))

Regarding the following proposition:

in Song mode, you can define when a track (say, track A) is playing a sequence (say, number 6) So, in a way, Foundry picks the sequence number it will playback. And that's independent of the other sequencers.

Two lines of thought are hitting me:

A) If you move the BEG/END pointers around in song mode, we can already do this. But I agree it is not as simple as moving just sequence numbers for each track, and when we want to play our full song again we also have to reset those two pointers for each track.

B) This could be somewhat equivalent to having separate sequence numbers in SEQ mode (4 edit heads, one for each track). Thus with the original PS behavior of the SEQ/SONG switch, if we had four edit heads, then in SEQ mode we would be able to do what is stated above, but without messing with song mode. So when I said I had tried 4 edit heads and it was confusing, perhaps I discounted it too fast, and with a good use of the ALL tracks button, we could easily move those 4 edit heads together when wanted.

Indeed, more thinking is needed :-)



1



joopvdl commented on Feb 26 • edited ▾

Sounds good :)

Now for the 6 million dollar question: In the event we were to have 4 edit heads (item B in my previous comment above), am I likely to see a request go by for having 4 separate SEQ CV inputs (instead of the one that is there now)? ;-)



joopvdl commented on Feb 26

As we're discussing a trig/gate/pulse sequencer, I would not find that too important. For chords it's very useful, but that's foundry territory. Although maybe someone would like to program a drum sequencer by live playing a midi drum kit, and recording the dynamics. But that wouldn't be me :)



MarcBoule commented on Feb 26

Owner

Since Foundry and the new gate sequencer would be based on the same core, I think it may apply to both. I'm thinking that in Foundry, given what you do with Janneker, if ever you have non-chords in Foundry, I imagine you would find having 4 separate SEQ CV inputs useful (as opposed to the single one that we have now), correct? That way you could have external control on all tracks separately, and control which one plays which sequence.



Eurikon commented on Feb 26

Author

For a phrase sequencer it might be useful for the upcoming polyphonic input capabilities, for incoming chords with after-touch or expression values. If i am understanding what you are talking about correctly.



joopvdl commented on Feb 26

To be honest, Marc, I'm not sure I understand the question.



MarcBoule commented on Feb 26

Owner

Oh, sorry. I meant that in Foundry if we had the same SEQ/SONG switch behavior as the one used in the PhraseSeqs, and if there were to be four edit heads (one per track), then the single SEQ CV input we have now would probably not be enough for those that like to externally control which sequences are playing in the sequencer (like I believe you do with the PhraseSeqs). If I understood correctly, you leave the PhraseSeqs in SEQ mode, and using the SEQ# cv input, you control the playing of the sequences yourself. If there are 4 tracks, then there would likely need to be 4 separate SEQ# cv inputs. Hope that came out better! :-)



joopvdl commented on Feb 27

Yes, I would like that for Christmas!



1



Eurikon commented on Feb 27

Author



joopvdl commented on Feb 27

And programming, too.

Lets think about what functions there are.

1. programming. Input gates, cv's. Creating sequences, songs.
2. playback. Selecting sequences (in sequence mode). Song mode.
3. on the fly programming while playback is active.

considerations:

visual feedback

randomization options

sequence length

gate length/ratcheting/other gate options

sequence clock

options to have each (of 4) track have individual settings/controls/feedback/playback settings/selection, or the 4 of them together.

This is probably not all, so feel free to add stuff.



joopvdl commented on Feb 28

To add some confusion to the confusion, here's what I like about the Aepelzen gate seq.

- it's a bit smaller than VCV pulse matrix
- but it's still very clear to see what's going on
- (I like to see all tracks and what's going on in them)
- individual clocks
- individual probability
- individual length
- pattern selection by vc (unfortunately just bank 1 is accessible) (and no "next pattern" trig input)

What vcv pulse matrix does better:

- play modes
 - reset (you know, that thing where sequencers, after reset, forget the first beat)
 - rnd and individual rnd
 - move pattern forward and backward (but that's a general setting, not for individual tracks)
 - I don't use the rows (pulse outputs on the bottom, but that's there too)

just some considerations, maybe not applicable for this discussion, maybe for the next pulse seq...



MarcBoule commented on Feb 28

Owner

Indeed, those are good points to raise Joop, and I think that the new GateSeq will have most of what you wrote. I think I will go ahead and change Foundry to make it behave like the other PS sequencers, and will have an option for 4 separate SEQ# cv inputs, so you can externally control the playing of the sequences in each track.

(Foundry is quite relevant to the discussion here, since I plan on making the new GateSeq starting with Foundry's code and its general approach)

The new gate seq will be a 4 track sequencer, with separate clocking + length + run modes + rnd/individual_rnd, just like Foundry has now, and by not needing the keyboard and oct lights we have on Foundry, the new gate seq will have room for the 4x16 matrix of led buttons.

So here's where I'm at now. I will make the change to Foundry to make it work like the PhraseSequencers. But since we have 4 tracks in Foundry, we will have 4 separate sequence numbers to manage in SEQ mode, such that we can make the sequencer play any sequence we want in any track. Since there's no room in the expansion panel to add 3 more SEQ# cv inputs (there's only one now), I will combine the functionality of the four CV2 IN ports and have a switch such that they can alternately function as SEQ# cv for each track. I will assume that with those 4 dual-purpose cv inputs, if we want to simultaneously control the seq# *and* write CV2 inputs, then it can't be done and we must do things separately.

Christmas will come early I think! ;-)
Cheers Joop!



1



Eurikon commented on Feb 28

Author

awesome guys!



joopvdl commented on Feb 28

Already looking forward to testing those babies :)



joopvdl commented on Mar 1

More considerations :)

I have a monome grid (64) that's great for programming drum sequencers (But I wish I had the bigger version)

When I connect it to one of the Monome modules, it's a wonderful way to program those.

Have you ever tried them out? There's a virtual grid, so you don't have to have hardware.

Anyway, it can also connect to regular modules, through midi notes (you need some extra software, but not much).

Here's the thing (and I think I suggested it before):

if you can give all buttons on all 4 grids a midi note value, you can program the sequencer with hardware, on the fly. (It needs to be all 4, because you don't want to have to switch between rows). You can "play" the drums! (much like on a real 909 or 808 or whatever.)



MarcBoule commented on Mar 1

Owner

Ok, I'll think about this one, but in order to have a note for each step (64), users with 61 key keyboards would not be able to control the whole array, unless I am misunderstanding something? I'll also see if there's a way to do this with the new midi mapping capabilities in Rack 1.0 when I do the code modifications.



joopvdl commented on Mar 2

That's true, but most keyboard controllers can be transposed up or down.

And the most practical use is when connected to a grid-like controller, like the monome or the ableton stuff. So the midi note transmitted/received is not too important, because we're not talking about actual notes here, just a means of communication between hardware and software.

As long as there's 64 different note values, it's ok.

yes most practical for grid based controllers, but very welcome!
Push / Monome / Maschine / Akai controllers.



joopvdl commented on Mar 3

most practical, but I don't know how hard that is to implement, would be a midi learn function.



Eurikon commented on Mar 4

Author

I think trowasoft modules work like that "out of the box",
so maybe a hint of how it should be done can be found there?
i can remember having used the Push controller with those modules when i still had the Push.
Just can not remember if i had to download anything extra for it to work.



MarcBoule commented on Mar 4

Owner

Thanks, I'll check that out.

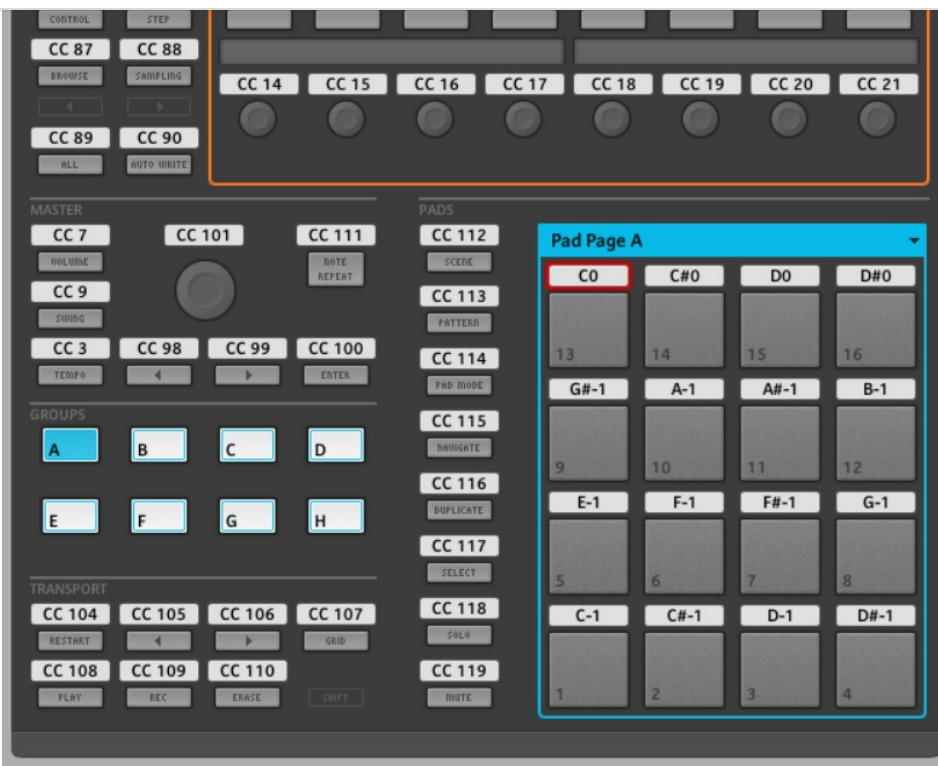


Eurikon commented on Mar 4 • edited ▾

Author

I also checked for you, i remember now, i used it with a Maschine controller from Native instruments at that time.
Also did away with that controller by now.
I have downloaded the control surfaces layout file, and taken a look at it.
Nothing much special done there related to the pads, only some controls added to resemble his module functions.
Most of the unseen under the hood work was done in a special pure data patch that also had to be downloaded,
that would translate all the midi information coming from an external controller in to OSC and pass it through to the trowasoft TrigSeq 64 module.

For now all i can say is that a default mapping of for example the Maschine controller (which i think will be the same on many controllers out there) with matrix of 4x4 pads is ranged from: C-1 to C0 on bank A
For Gate Seq 64 there would be 4 banks of midi notes assigned to all the steps in the case of 1x64 steps having the last step end at C3



hope any of this makes sense, cause i don't see yet how a VCV module could be locked to a note range coming in via midi. while still receiving other CV information from that same midi input.

edit:

i remember one other thing, i think the pure date patch was also necessary to provide visual feedback by "telling" the controller what steps were active. So the controller would have the corresponding lights on in the pads.



MarcBoule commented on Mar 9

Owner

Hi guys, just to update this thread since we were also talking about Foundry above, I have changed Foundry's main switch (SEQ/SONG) to make it behave the same as in the Phrase Sequencers. The expansion panel was also enlarged such that we now have 4 separate SEQ# cv inputs to externally control the playing of the sequencer. Nigel had a great idea to also add an option to make those sequence changes happen only when the sequence gets to the end. This can be pretty neat: when we choose the Notes mode for those inputs and we use a keyboard to control which sequence plays, we don't have to have perfect timing! If ever you want to try it out, I put a build in the releases section (0.6.16b).



Eurikon commented on Mar 9

Author

Thanks Marc! Exciting changes 👍👍👍



joopvdl commented on Mar 10

downloading now! :)

Op zo 10 mrt. 2019 om 01:40 schreef NRG70 <notifications@github.com>:

...

About the switching at the end of a sequence (which is a great idea!)
Have you thought about what happens if, when seq switching is set to trig increase, the sequencer receives a trigger at the beginning of the sequence? Will it switch immediately, or will it wait until the end?

Op zo 10 mrt. 2019 om 10:35 schreef Joop van der Linden <joopvanderlinden@gmail.com>:

...



MarcBoule commented on Mar 10

Owner

The trig-increment mode should work like the two other modes, that is, if the sync switch is activated, it will register the trigger internally but the sequence change will actually only happen at the end of the sequence. I would say if you are using Foundry with Janneker you probably don't need to activate sync mode though.



joopvdl commented on Mar 10

Ah, it's a separate mode. Hadn't seen the switch, I thought it was implemented. Nice!

Op zo 10 mrt. 2019 om 15:13 schreef Marc Boulé <notifications@github.com>:

...