LIVID Block MIDI Step Sequencer

32-Step Monophonic MIDI Sequencer Built in MAX/MSP

Austin Stockwell
June 2019

Acknowledgments

Various resources were used to help create this step sequencer. The end result was a device that was suited to my specific needs in music production. I would like to thank all of those who helped create the final product! [1] [2]

Contents

List of Figures

1 Introduction

This project started as a desire to create a customizable MIDI step-sequencer that could be used with the LIVID Block MIDI controller in both a studio and live environment. Due to the power and flexibility of MAX/MSP, the program was chosen as the platform to build the sequencer.

The result was a monophonic 32-step MIDI sequencer that had all functionality accessed from the LIVID Block as well as a GUI built into the MAX patch itself. The sequencer can control external MIDI instruments as well as receive a master MIDI clock that allows it to by synchronized to external devices.

The sequencer is a fantastic live tool as it can store multiple sequences that can be recalled at any point in time with the push of a button – essentially allowing the user to combine multiple 32-step sequences to create larger chains of patterns in real time.

The sequencer can also serve as a compositional tool in the studio as it can create a variety of sequences that can be stored in external MIDI files that can be processed and altered for a variety of compositional needs.

The above descriptions of the device are just highlighted features. The pages that follow serve as a way to explain the build philosophy, document the sequencer's schematics and build, and explain its uses in full detail while suggesting further improvements that can be made in the next iteration.

2 Features

The following features are integrated into the sequencer:

- MASTER ON/OFF
- MIDI TO
- MIDI IN (external clock)
- 15 storable presets
- Clear presets button
- Number of active sequences (up to 16 sequences per preset)
- Active sequence editing selection
- Pitch/velocity/duration editing
- Active sequence ON/OFF
- Sequence playback (forward/backward/rotate)
- START/STOP step selection
- Timing division
- LIVID Block integration (faders/knobs/buttons/lights)

3 Overview

3.1 GUI

The Graphical User Interface (shown in Figure 1) allows the user to access all of the features of the device.

To begin, the user first turns on the device with the 'MASTER ON/OFF' button. The user then selects which device to send MIDI information to by selecting the appropriate device from the 'MIDI TO' dropdown menu. Next, the user selects which parameter will be edited by selecting the 'EDIT' dropdown menu. The user can choose from the following options:

- Pitch
- Velocity
- Duration
- Extra1 (Assignable)
- Extra2 (Assignable)

It is possible to alter the chosen sequence using a mouse and the graphical keyboard sequencer as shown. To do so, simply click the selected step (yellow / blue boxes) and drag them to the desired value shown. The blue squares represent the "downbeat" of each sequence (1, 5, 9, etc). When changing a step's pitch, the note value is given on the far left of the sequencer's window (shown with a virtual piano keyboard). When changing a step's duration or velocity, the value is shown with a series of vertical lines (each corresponding to an individual step). After programming a series of notes and velocities, the user can select the sequence playback direction using the 'DIRECTION' dropdown menu.

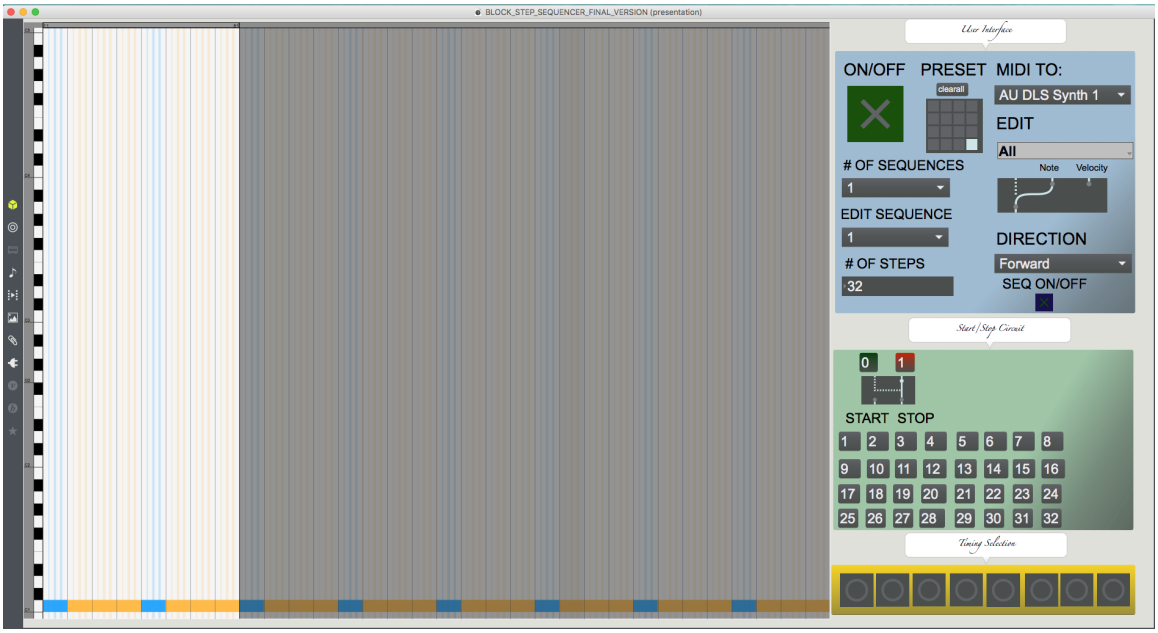

Figure 1: GUI (Shown editing note values)

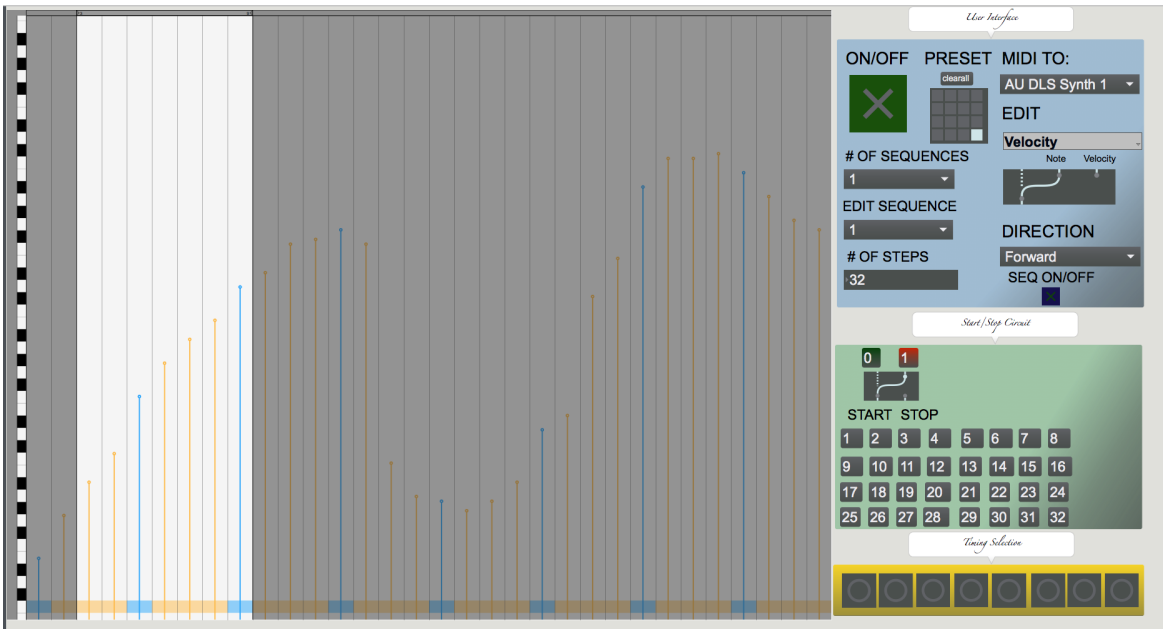

Figure 2: GUI (Shown editing velocity values)

It is also possible to start and stop the sequences from any step. The 'active' region of the sequence is shown in the graphical keyboard sequencer as the WHITE REGION. The GRAY REGION of the graphical keyboard sequencer is the 'unactive' region of the sequence. Therefore, Figure 1 shows the sequence starting on step 1 and ending on step 7. Figure 2 shows the sequence starting on step 3 and ending on step 9. Where the sequence starts and stops is chosen by the user.

For example, if the user wants the sequence to start on the third step and stop on the ninth step (shown in Figure 2 and Figure 3), the user would:

- PRESS GREEN 0 BUTTON (START)
- PRESS 3 ON NUMBER PAD
- PRESS RED 1 BUTTON (STOP)
- PRESS 9 ON NUMBER PAD

Each preset and its corresponding sequences can be stored using the 4X4 grid (16 presets total). To save a preset hold down the SHIFT button while clicking the desired preset square (SHIFT + CLICK).

NOTE: Only 15 presets are available when using the sequencer with the LIVID Block controller!

Figure 3: GUI (Starting sequence on step 3 and ending on step 9)

If it is desired to have more than one sequence playing at any given time, the 'NUMBER OF SEQUENCES' dropdown menu should be changed from 1 to the desired number of sequences (up to 16). The 'EDIT' dropdown menu is then used to select which of the sequences are to be edited. The graphical keyboard sequencer will only show the values of the current selected sequence. If the user wishes to deactivate any of the programmed sequences, the 'SEQ ON/OFF' button is used.

Note: When using the LIVID Block MIDI controller, only one active sequence is available for any preset at a given time!

3.2 LIVID Block MIDI Controller

Nearly all features but the ability to select and program multiple sequences per preset are accessible from the LIVID Block MIDI controller. The functionality of the different components are as follows:

- Knobs K1 - K8: Step Parameter Selection (Pitch/Velocity/Duration...)
- Button F1: Knob edit (STEPS 1 - 8)
- Button F2: Sequence START POINT
- Button F3: Knob edit (STEPS 9 - 16)
- Button F4: Sequence END POINT
- Button F5: Knob edit (STEPS 17 - 24)
- Button F6: Knob edit (STEPS 25 - 32)
- BPM Button: Knob edit (note / velocity)
- Slider S1: MASTER CLOCK START/STOP
- Slider S2: Sequence playback (forward/backward/rotate)
- Grid G1 - G32: Step selection (backlit buttons to show active step)
- Grid G33 - G60: RECALL (Presets 1-15)
- G37 - G63: SAVE (Presets 1-15)
- G64: CLEAR ALL PRESETS

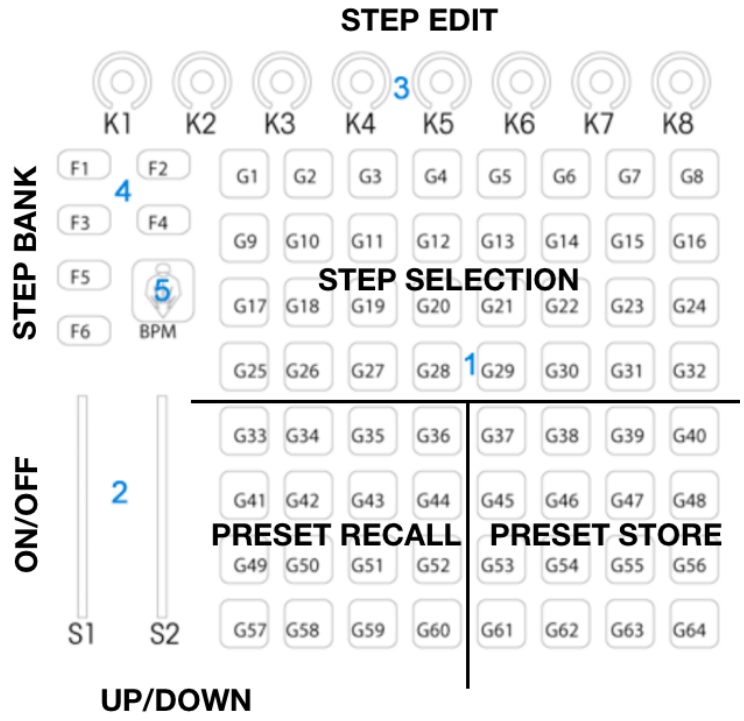

Figure 4: LIVID Block layout

4 Design

Figure 5 shows the completed 'sky-view' design broken into separate main modules. Each main module has seperate sub-modules that will be explored in more detail. However, the main modules shown in Figure 5 are as follows:

- Internal clocking module
- Step START/STOP module
- Timing selection module
- Livid Block controller module
- Step sequencer / external clock module
- GUI module

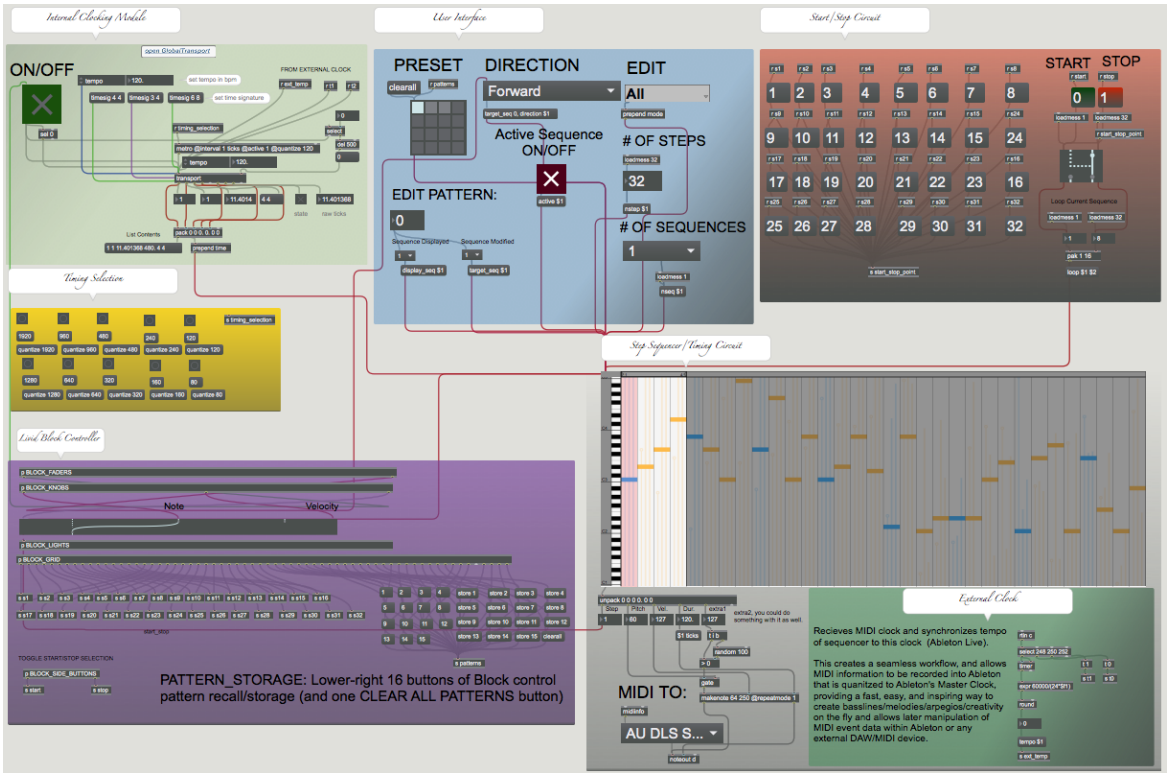

Figure 5: Design overview

4.1 Internal Clocking Module

The internal clocking module provides the necessary timing circuit to advance the sequencer by a specified time interval. A closer look of the module is given in Figure 6. The main components of this module are the following objects:

- metro
- transport
- pack
- prepend

The module uses the 'metro' object to act as a metronome which outputs 'bangs' at a regular, specified interval. This object interfaces with MAX/MSPs 'Global Transport' master clock via the 'transport' object. The 'transport' object converts the bangs coming from the 'metro' object into musical BARS, BEATS, RESOLUTION (ticks per beat), and TEMPO values to be used by the 'live.step' object. These values are then packaged via the 'pack' object and attached via 'prepend' object to a message delivered to the 'live.step' object (the main graphical step sequencer of the GUI). This information serves to advance the sequencer step-by-step according to the selected timing interval.

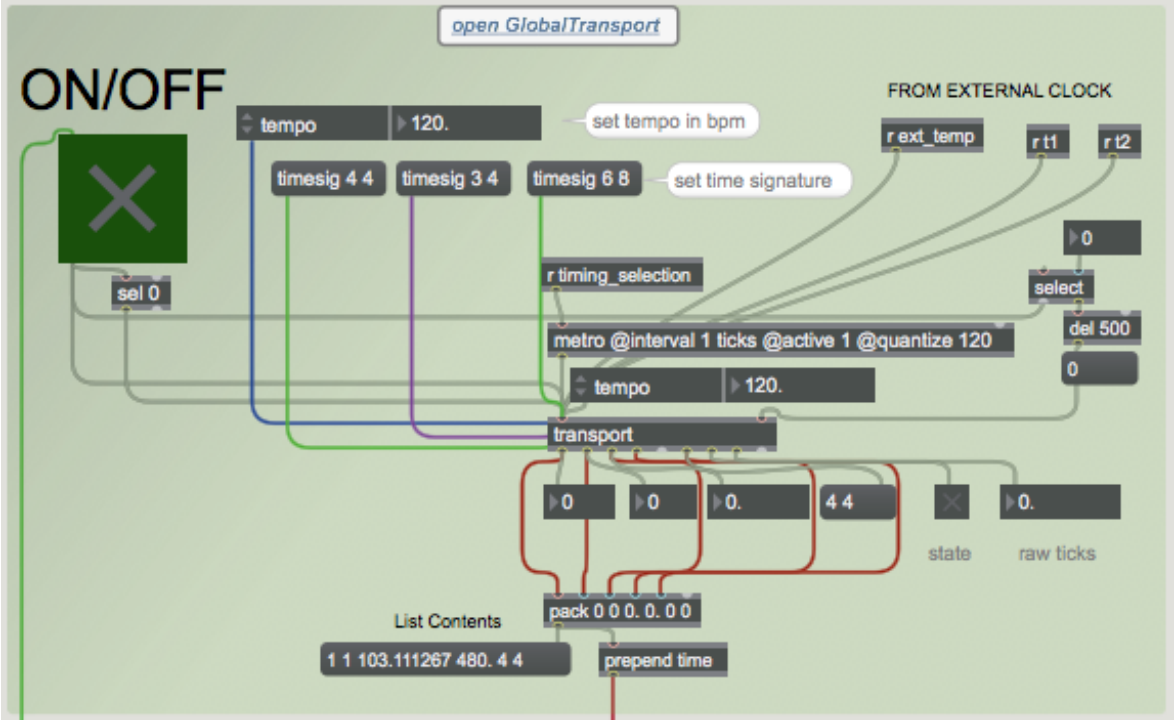

Figure 6: Internal Clocking Module

4.2 Step START/STOP Module

This module allows the sequencer to begin and end at any point within a sequence (such as in Figure 3). A closer look at the module is given below in Figure 7. The main components of this module are the following objects:

- gswitch
- messages
- return objects (r)
- send objects (s)

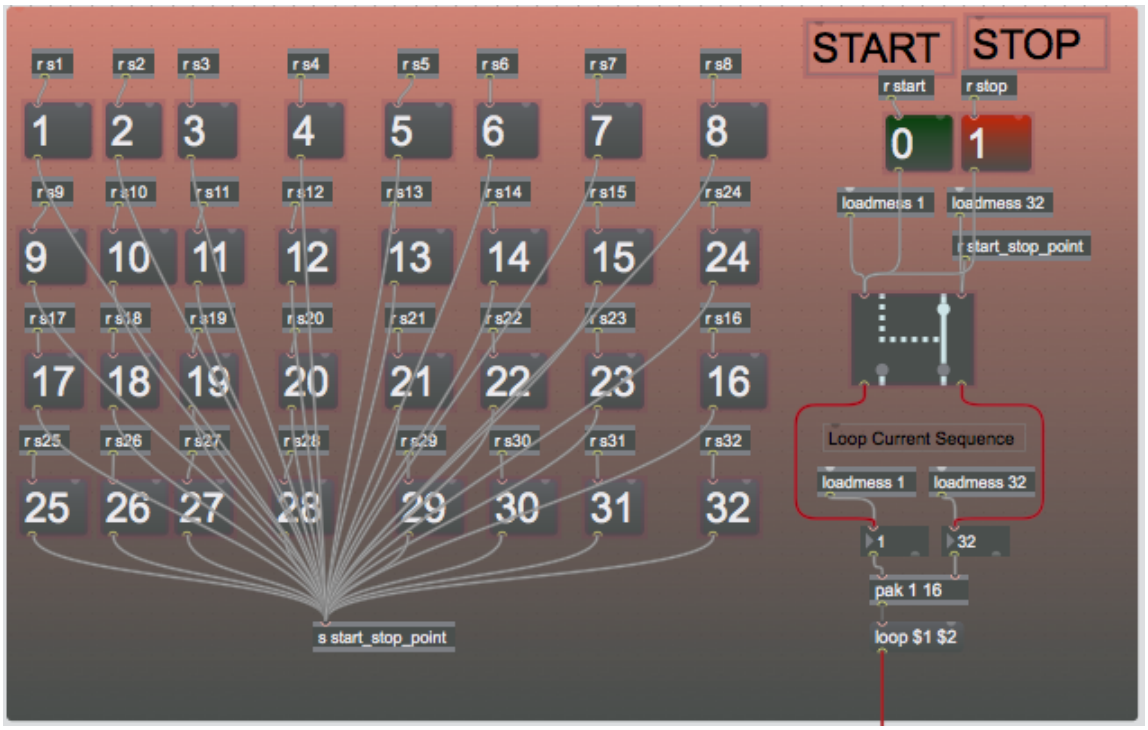

Figure 7: Step START/STOP Module

Each individual message number (1 - 32) is connected to its own 'r' object (r s1 - r s32). Each 'r' object receives a message from the LIVID Block Controller module (MIDI data from the controller) and activates the corresponding step number message if that button is pressed on the LIVID Block. Therefore, LIVID Block buttons B1-B32 correspond with the number messages 1-32 of this sub-module. This allows the LIVID Block to choose which step is selected for the individual START/STOP points of the sequence being edited.

NOTE: THE 's' OBJECT IS ONLY USED TO REMOTELY 'SEND' THE STEP NUMBER VALUES TO THE 'gswitch' OBJECT TO HELP ELIMINATE PATCH CABLE CLUTTER!)

If the green '0' button (START) is pushed on the GUI (or button F2 of the LIVID Block), the 'live.step' object will START on the following number selected. If the red '1' button (STOP) is pushed on the GUI (or button F4 of the LIVID Block), the 'live.step' object will STOP on the following number selected.

For example, if the user wants the sequence to start on the third step and stop on the ninth step (shown in Figure 3), the user would:

- PRESS GREEN 0 BUTTON (F2 on LIVID Block)
- PRESS 3 ON NUMBER PAD (G3 on LIVID Block)
- PRESS RED 1 BUTTON (F4 on LIVID Block)
- PRESS 9 ON NUMBER PAD (G9 on LIVID Block)

4.3 Timing Selection Module

The Timing Selection Module allows the user to subdivide the clock by a predetermined set of values. This module is used for more artistic and creative approaches rather than a purely scientific approach as behavior can be hard to detect. A closer look at the module is given below in Figure 8. The main components of this module are the following objects:

- button
- message
- send object (s)

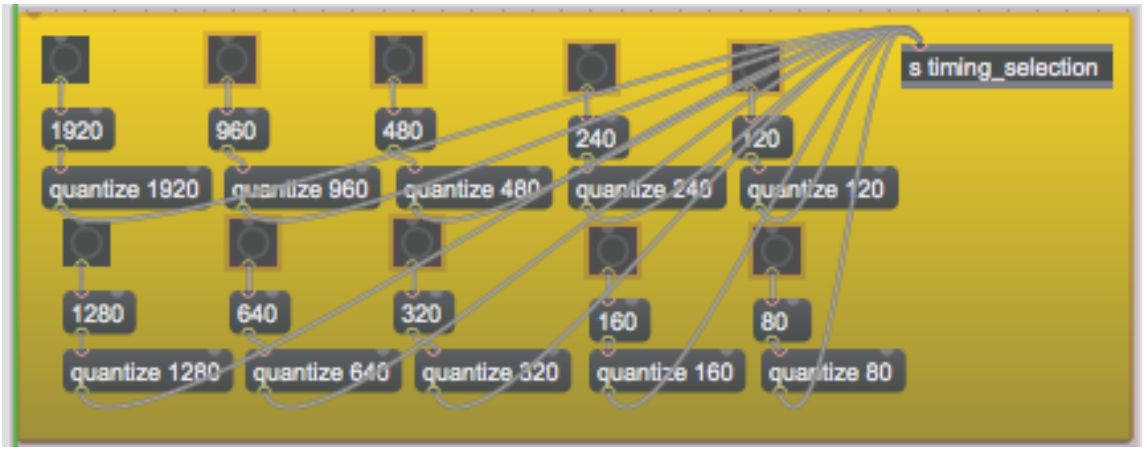

Figure 8: Timing Selection Module

When the user clicks on any of the buttons of this module, the various 'quantize' messages are activated. The messages are then sent to the 'metro' object of the Internal Clocking Module (as shown in Figure 9) via the 's' (send) object. Each 'quantize' message contains a different argument number that is used to determine 'metro' behavior. The following list loosely explains the behavior of each button value (for creative efforts):

- 1920: Plays only the 1st and 5th step at half-speed
- 1280: Plays 1st and 9th step at half-speed
- 960: Plays 1st and 9th step
- 640: Plays 1st and 5th step at half-speed

- 480: Plays 1st and 5th step
- 320: Plays 1st and 9th step
- 240: Skips every other step
- 160: Plays 1st and 5th step
- 120: Plays every step
- 80: Plays every other step

NOTE: There are four buttons that have redundant functionality (160/480 and 320/960). For this reason, only eight buttons appear on the GUI rather than ten.

NOTE: The timing selection module is currently not supported by the LIVID Block MIDI Controller!

Figure 9: Sending timing division internal clock ('metro')

4.4 LIVID Block Controller Module

The LIVID Block Controller Module is used to incorporate the hardware successfully into the patch. This module is broken down into 5 'sub-modules' as follows:

- BLOCK FADERS
- BLOCK KNOBS
- BLOCK LIGHTS
- BLOCK GRID
- BLOCK SIDE BUTTONS

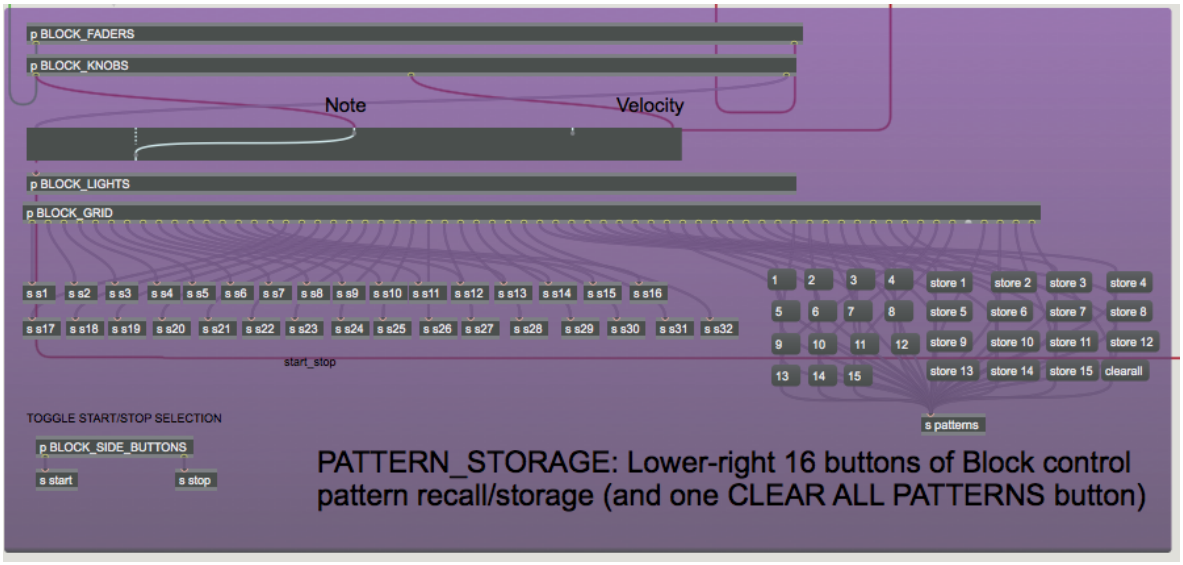

Figure 10: LIVID Block Controller Module

4.4.1 BLOCK FADERS

This sub-module is used to accept CC values from the LIVID Block's two faders (S1 and S2). Two CC values are used when receiving information from the controller's faders to control the patch:

- CC value assigned to that particular fader (9 for S1 / 8 for S2)
- CC value representing range of motion within each fader (0 -127)

Slider S1 turns ON/OFF the sequencer master clock. Slider S1 corresponds to CC value 9. This value is received via the 'ctlin 9' object. The 'scale' object then takes the values (0 -127) received from the LIVID Block that represent the position of the fader and outputs either a 0 or a 1. Thus, when the object receives the highest value from the fader (127) a 1 is output, and when the object receives the lowest value from the fader (0) a 0 is output. This is then routed to the MASTER ON/OFF button of the Internal Clocking Module of the patch to turn the master clock ON (1) or OFF (0).

Slider S2 allows the user to change sequence playback. The left S2 slider of the LIVID Block corresponds to CC value 8, received by the object 'ctlin 8'. The 'select' objects select discrete values throughout the range of the fader (0 -127). If any of the four pre-defined values are received from the fader (0, 45, 90, 127) the 'DIRECTION' drop down menu object of the User Interface Module will change from Forward/Backward/Back and Forth/Rotate.

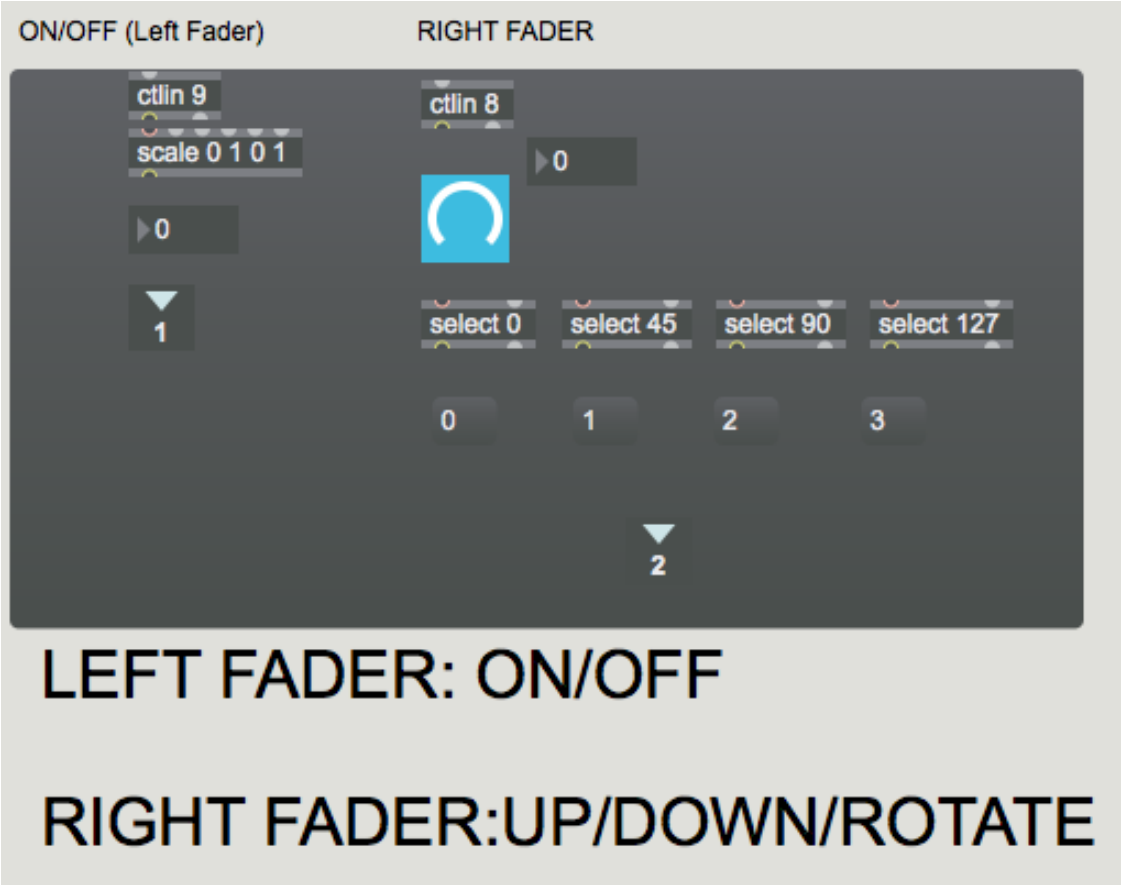

Figure 11: Block Faders Submodule

4.4.2 BLOCK KNOBS

This sub-module allows the user to change all 32 individual step parameters (pitch and velocity) using the 8 knobs of the LIVID Block controller. This is done by dividing the 32 steps into four banks of 8 steps per bank. Which steps are controlled with the knobs is determined by the current bank as follows:

- Button F1 (STEPS 1 - 8)
- Button F3 (STEPS 9 -16)
- Button F5 (STEPS 17 -24)
- Button F6 (STEPS 25 - 32)

The 'midiin' object receives MIDI data from the LIVID Block controller. That data is then sent to the 'select' objects by parsing and unpacking the MIDI information via the 'midiparse' and 'unpack' objects. Because F1, F3, F5, and F6 are momentary switches, the 'stripnote' object is used to "latch" the value of the buttons to 1 when pressed. Without the 'stripnote' object, the button press would always fall to 0 once it is released rather than remaining at 1 (which is needed to activate the buttons that activate the 'radiogroup' object at the bottom of Figure 12). This 'radiogroup' object is then connected to the 4-channel 'gswitch' objects of the Pitch and Velocity sections of the sub-module that determine which step group is sent out to the Graphical Step Sequencer.

Figure 12: Bank Selection Section

The pitch and velocity of the individual steps are controlled from the 8 knobs of the LIVID Block controller via the Pitch and Velocity sections shown in Figure 13 and Figure 14. The separate 'ctlin' objects of each of the section receive data from the 8 knobs of the controller (CC3 corresponds to K1 and CC2 corresponds to K2, etc)

The 'scale' objects of the Pitch Submodule scale the input from the knobs (0 -127) to values between 36 (MIDI value for LOW C) and 84 (MIDI value for HI C). This essentially gives the user a total of 4 octaves that can be reached with a single control knob of the LIVID Block.

There are 32 'prepend' objects in each of these two sections as well as a four-channel 'gswitch' object. 'Prepend' objects 1 - 8 correspond to steps 1 - 8 and are therefore connected to the first "channel" of the 'gswitch' object (blue wires of figure 13).

The 'gswitch' object is controlled by pressing either of the four buttons on the LIVID Block (F1/F3/F5/F6). Therefore, when button F1 is pressed on the Block, only the data sent by the 'prepend' objects 1 - 8 get routed to the output to control the 'live.step' object (Graphical Step Sequencer) of the GUI.

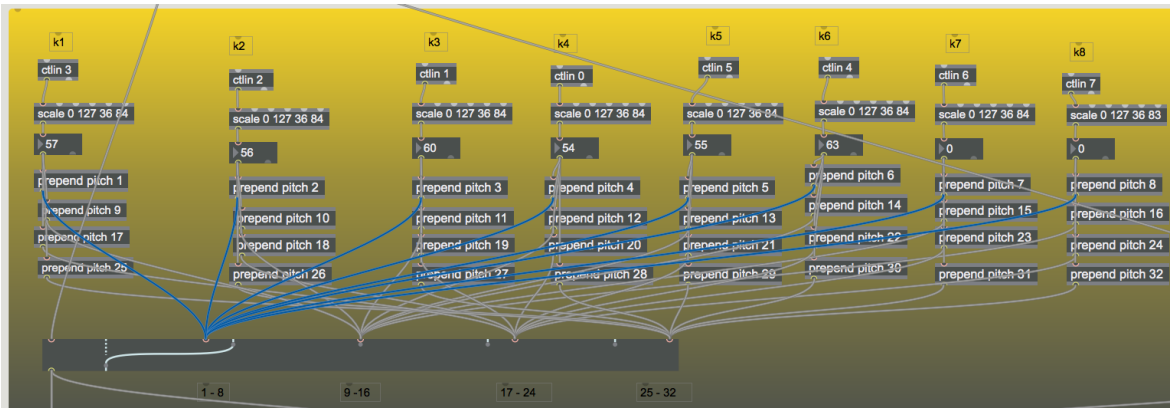

Figure 13: Pitch Section

Figure 14: Velocity Section

The final component of this submodule is the Pitch/Velocity Select. This section receives input from the BPM button of the LIVID Block (large square button) and uses the button press to toggle between having the K1-K8 knobs control either pitch or velocity.

The two inputs to the 'gswitch' object are the two separate Velocity and Pitch sections (Figure 13 and Figure 14) explained in the previous section.

The 'mididselect' module accepts input from the LIVID Block BPM button (MIDI note value 64) and uses the 'unpack' object to send either a 0 (OFF) or a 64 (ON) to the 'select' object, which in turn activates a button and the 2-channel 'gswitch' object.

- BPM Button press 1: Pitch data is output to the Graphical Step Sequencer
- BPM Button press 2: Velocity data is output to the Graphical Step Sequencer

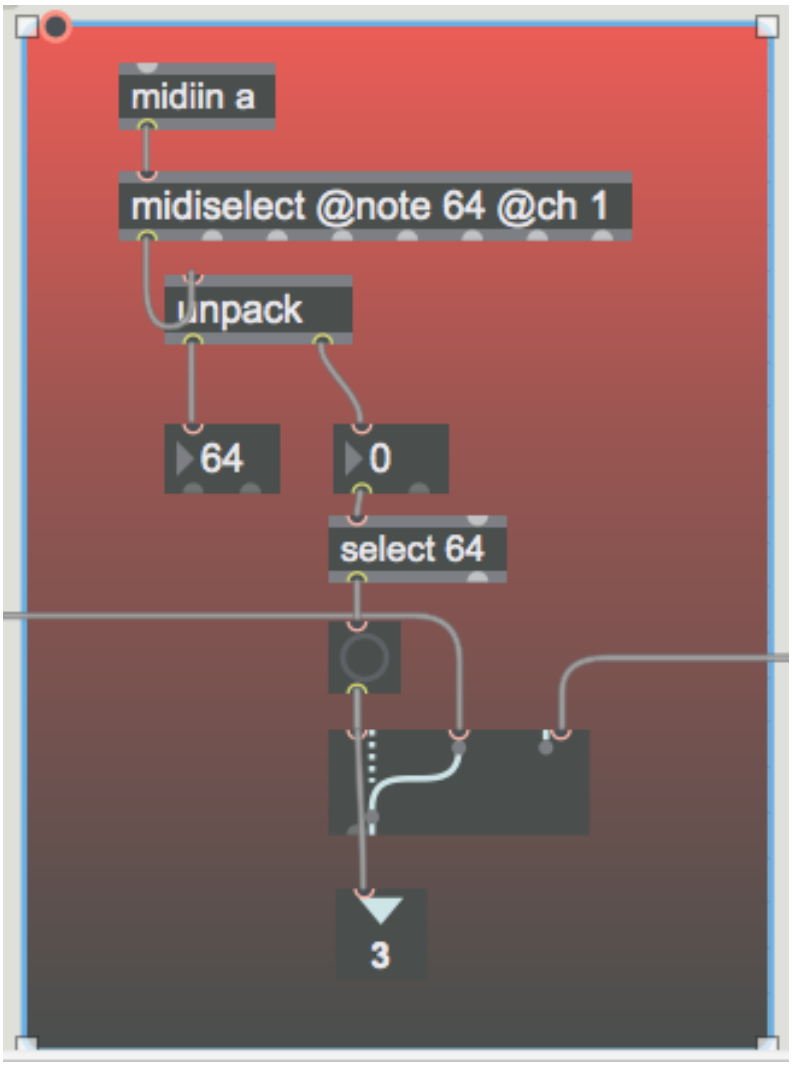

Figure 15: Pitch/Velocity Select Section

4.4.3 BLOCK LIGHTS

This module controls the grid lights of the LIVID Block controller. The system allows the user to view which step is active while also interpreting tempo information via the 32 LEDs of the LIVID Block. When a step is active, the light corresponding to that step value is turned ON via the brown message corresponding to that light.

For example, the top left light (button G1) is turned ON using the following message: (144, 0, 60) and is turned OFF using the following message: (144, 0, 0).

The light is turned off using the 'delay object' (set to an interval of 50 ms) immediately after the light is turned on. This prevents the lights from 'sticking'.

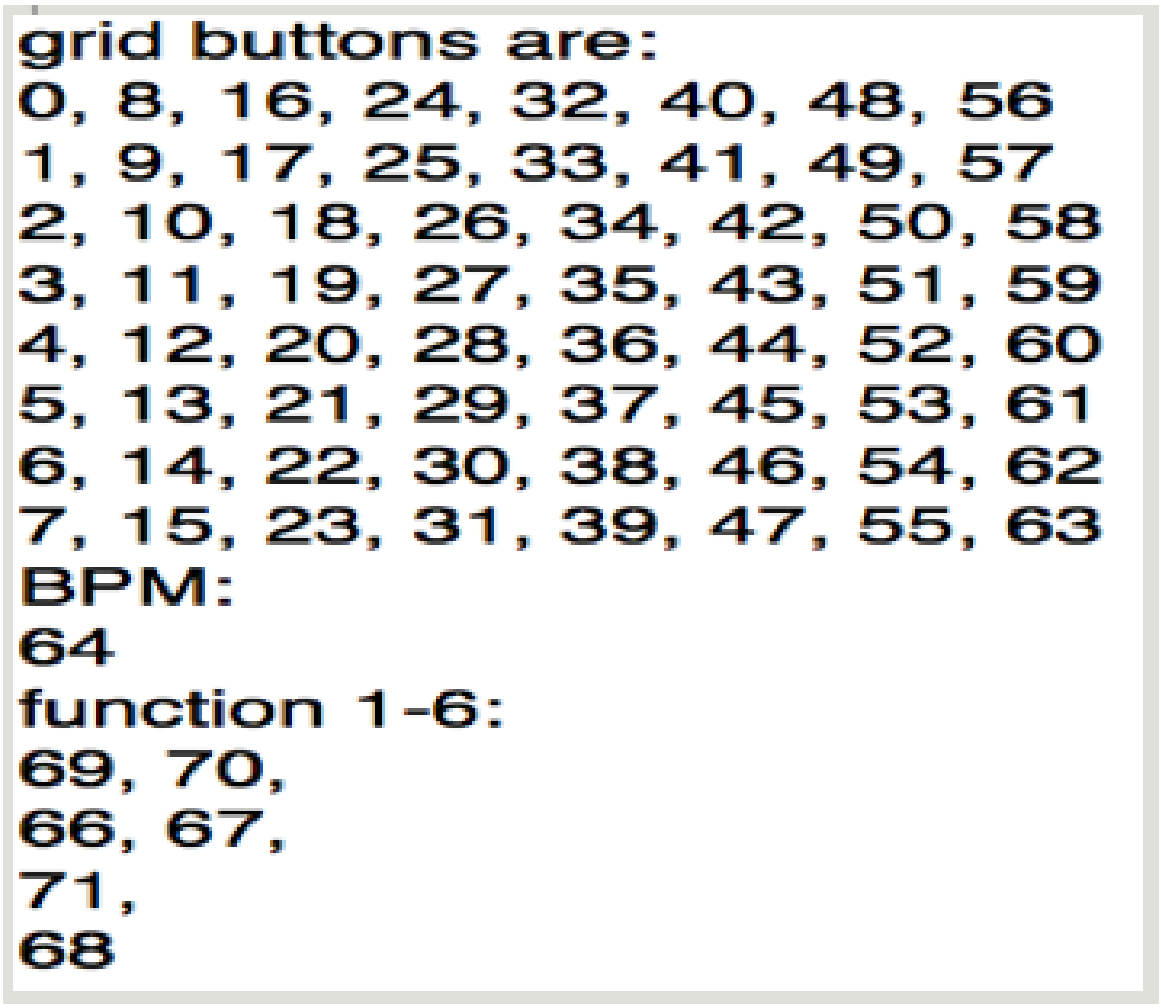

Figure 16: Block lighting system CC values

The Block Lights Module receives individual step value input from the output of the 'live.step' object (the Graphical Step Sequencer). This information comes into the Block Lights Module via the large '1' object (a return object). This data then increments a 'counter' object which in turn increments the 'select' object. When the various stages of the 'select' object are activated, each individual LIGHT ON message is activated (turning on the active step's LED), followed by activation of the LIGHT OFF message. The control structure is as follows:

- SELECT 1 ACTIVATED
- OUTPUT TURNS ON G1 LIGHT (activates message (144, 0, 60))
- SELECT 1 output also activates a 'delay 50' object
- OUTPUT of 'delay 50' TURNS OFF G1 LIGHT (activates message (144, 0, 0))

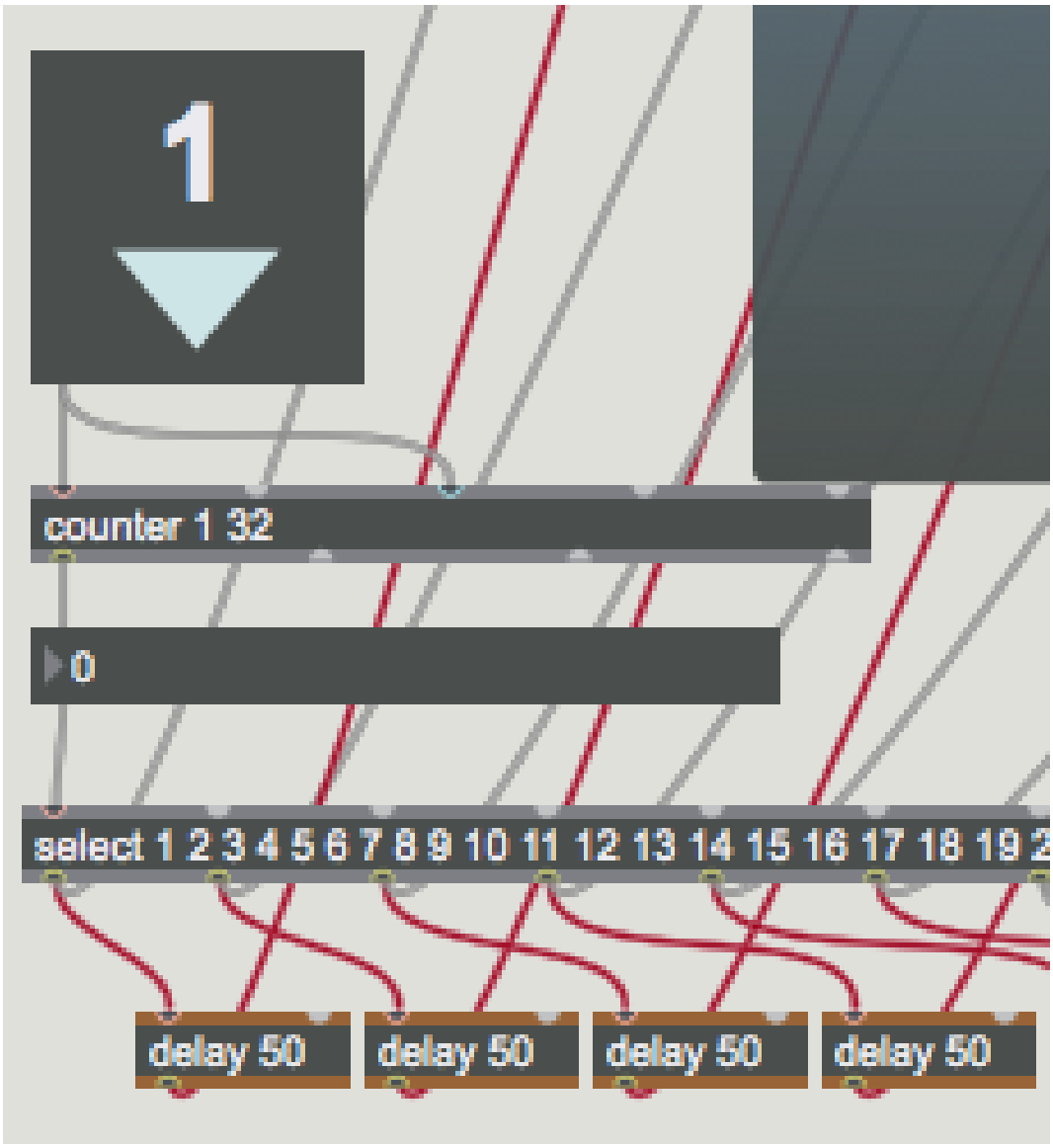

Figure 17: Block lighting clock input / control structure

Figure 18: Block light grid

The output of the 'message' objects that control the lighting are routed to four unique 's' objects (sends) that correspond to the four rows of eight LEDs of the controller. These 's' objects are connected to the 'r' objects wirelessly (shown in Figure 19) which then send the message data to the LIVID Block controller via the 'midiout' object.

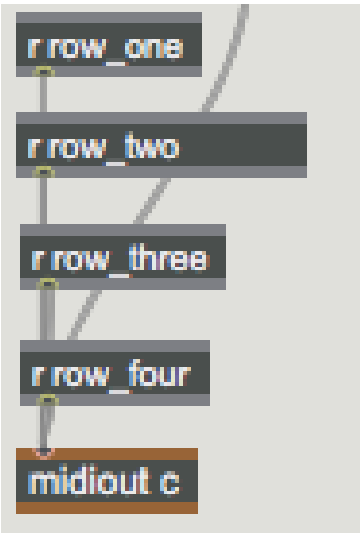

Figure 19: Light ON/OFF data sent to LIVID Block

4.4.4 BLOCK GRID

This patch receives input from each of the LIVID Block's 64 grid buttons and sends the button push data to multiple points of the patch for use in the START/STOP functions as well as PRESET SAVE/RECALL functions of the sequencer. The 'midiin' objects receive the MIDI information from the LIVID Block's individual grid buttons and the data is then routed to the corresponding 'midiselect' objects. Each 'midiselect' object will only pass data that corresponds to the arguments given in the object.

For example, the left-most 'midiselect' object has arguments @note0 and @ ch1. This means that that particular object will only pass data through if the note value received from the LIVID Block is 0 on MIDI Channel 1. A quick reference to Figure 9 shows that the note value for the top left grid button is in fact 0.

Because there are 64 grid buttons on the LIVID Block controller, this patch actually contains 64 different sub-circuits shown in Figure 20 (which shows 4 of the 64 sub-circuits). This object then sends the data to be unpacked via the 'unpack' object.

Each button press is separated into two 'number' object boxes. When each button is pressed, it sends a number 64 for ON and 0 for OFF. Therefore the two number boxes for each button represent:

- LEFT NUMBER: Control Value (Button Value)
- RIGHT NUMBER: 64(ON) or 0 (OFF)

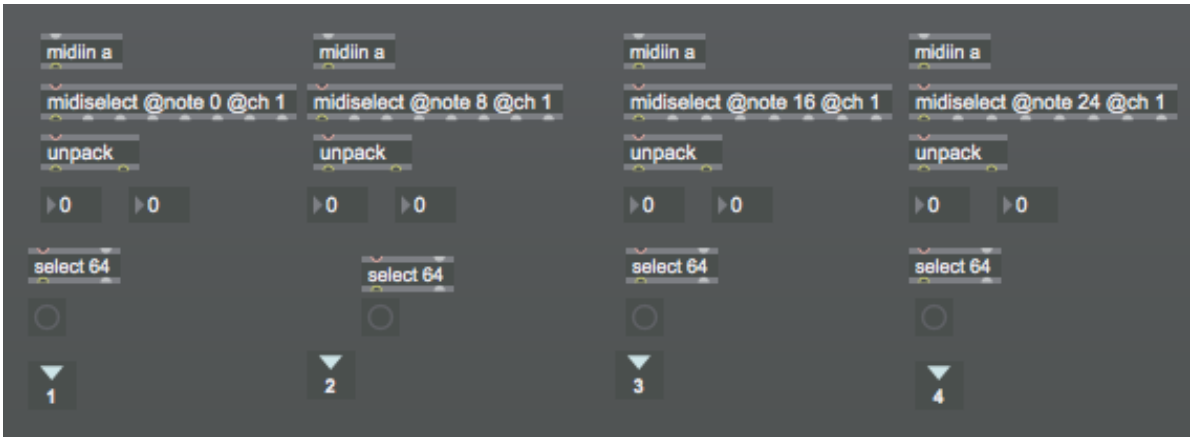

Figure 20: Block Grid Submodule

4.4.5 BLOCK SIDE BUTTONS

The final sub-patch of the LIVID Block Controller Module is the Block Side Button sub-module. This patch receives input from the LIVID Block's F2/F4 control buttons and routes them to the START and STOP feature (allowing the user to choose the beginning and end of each sequence). As shown in Figure 21, it functions exactly like the Block Grid sub-module discussed in the previous section.

Figure 21: Block Side Buttons Submodule

4.5 Step Sequencer / External Clock Module

The Step Sequencer External Clock Module is the final module of this sequencer. Its purpose is to interface with the 'live.step' object (Figure 23) as well as receive external MIDI clock from an external device (such as Ableton Live). This allows the sequencer to play at the same rate specified by the external piece of equipment.

The patch is composed of the 'live.step' object which is then sent to an 'unpack' object that splits the data coming from the 'live.step' object into five discrete numbers (Step/Pitch/Velocity/Duration/Extra1). These values are then sent to a 'noteout' object that creates and sends a MIDI message out of the patch to an external MIDI device.

The external clock submodule utilizes a 'rtin' object that receives incoming MIDI real time messages transmitted from a specified MIDI input device. The next 'timer' object reports elapsed time between two events. The timer starts keeping time when a bang is sent to the left inlet. The right outlet reports elapsed time in milliseconds. This timing value is then rounded and converted into a number that is sent to the 'transport' object within the Internal Clocking Module (which controls the master clock of the sequencer).

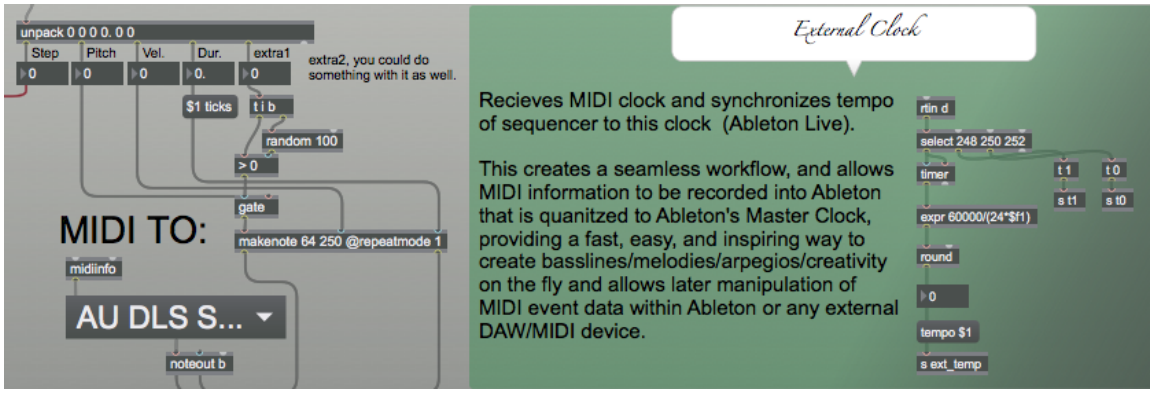

Figure 22: Step Sequencer/External Clock Module

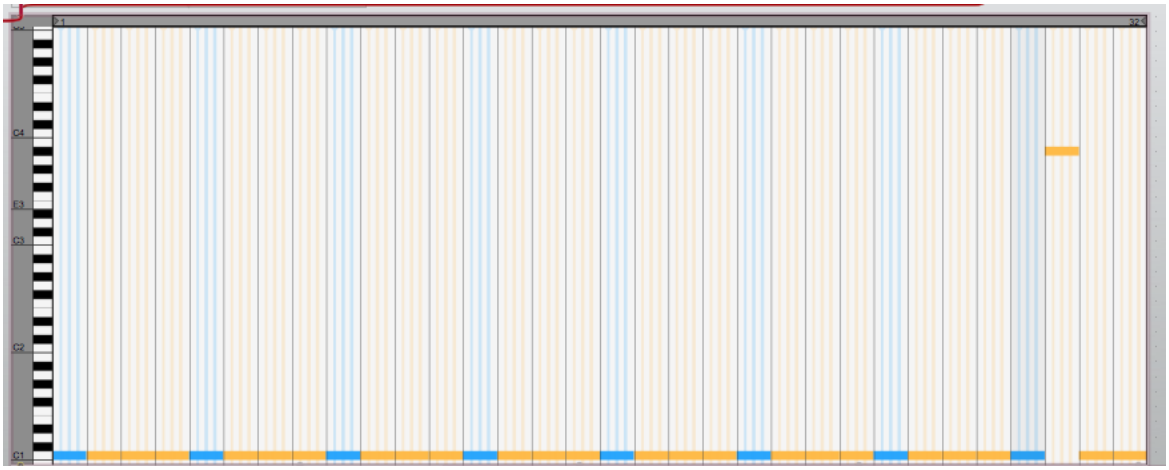

Figure 23: 'live.step' object

5 Future Improvements

Although the LIVID Block Step Sequencer works wonderfully for its intended design there are many features that will be added in the future. The improvements are as follows:

- Polyphony
- Addition of 32 steps (64 total)
- Swing
- Time signature selection
- Ability to clear individual presets
- Timing division using LIVID Block controller
- Multiple sequence programming using LIVID Block controller

References

[1] M. G. Alessandro Cipriani, *Electronic Music and Sound Design: Theory and Practice with MAX/MSP*. ConTempoNet, 2014, vol. 2.

[2] ——, *Electronic Music and Sound Design: Theory and Practice with MAX/MSP*. ConTempoNet, 2009, vol. 1.