# Bome MIDI Translator Pro Setup for Akai APC Mini MK2 LED Feedback

This guide provides step-by-step instructions to configure Bome MIDI Translator Pro for controlling the Akai APC Mini MK2's LEDs based on clip states in Ableton Live 12. This setup ensures:

\* All pads turn green on startup.

\* Launched clips turn their pad red.

\* Stopping a clip turns its pad green.

\* Launching a new clip in a track turns the previously playing clip's pad (in the same track) green.

\*\*Important Note:\*\* This guide assumes Ableton Live is configured to send MIDI Note On feedback (for notes 0-63) back to the Bome Virtual Port when clips are launched or stopped. If this feedback is missing, the main pad lighting (Steps 6 & 7) will not work, and you need to troubleshoot Ableton's MIDI output/feedback configuration first.

## Prerequisites

Before starting, ensure you have completed the initial setup:

1. \*\*Bome MIDI Translator Pro:\*\*

 \* MIDI Thru Paths configured in Project Properties -> MIDI Router:

 \* `1 - APC - MINI - MK2` (or your controller alias) -> `Ableton1` (or your Bome Virtual Port alias)

 \* `Ableton1` (or your Bome Virtual Port alias) -> `1 - APC - MINI - MK2` (or your controller alias)

 \* Aliases created (Recommended):

 \* `Ableton1` for `Bome MIDI Translator Virtual 1`

 \* `1-APC-MINI-MK2` (or `Controller`) for your `APC MINI MK2` physical port.

2. \*\*Ableton Live 12:\*\*

 \* Preferences -> Link/MIDI:

 \* Control Surface row for APC Mini MK2: Set to `None`.

 \* Input row for `BMT 1` (Bome Virtual Port 1): Set Control Surface to `None`, Track [On], Sync [On], Remote [On].

 \* Output row for `BMT 1` (Bome Virtual Port 1): Set Control Surface to `None`, Track [On], Sync [On], Remote [On].

## Bome MIDI Translator Pro Configuration

Follow these steps within Bome MIDI Translator Pro:

\*\*Step 1: Create the Preset\*\*

1. Create a new preset named `APC Mini MK2 Lights`.

2. Select the preset.

3. In \*\*Preset Properties -> Preset Default MIDI Ports\*\*:

 \* \*\*MIDI INPUT:\*\* Select `Specific Ports...` and check \*\*only\*\* `Ableton1`.

 \* \*\*MIDI OUTPUT:\*\* Select `Specific Ports...` and check \*\*only\*\* `1-APC-MINI-MK2` (or your controller alias).

\*\*Step 2: Translator - Initialize Track Variables\*\*

\*Purpose: Resets the memory of which clip is active in each track when the project loads.\*

1. Add a translator named `Initialize Track Variables`.

2. \*\*Incoming Action:\*\* `Project`, `The Project file is opened`.

3. \*\*Rules:\*\*

 ```text

 // Initialize all track active clip variables to "no clip" (127)

 g0 = 127

 g1 = 127

 g2 = 127

 g3 = 127

 g4 = 127

 g5 = 127

 g6 = 127

 g7 = 127

 Exit Rules, Skip Outgoing Action

 ```

4. \*\*Outgoing Action:\*\* `None`.

\*\*Step 3: Translator - Start Set Lights Loop\*\*

\*Purpose: Initiates the process of turning all pads green on startup.\*

1. Add a translator named `Start Set Lights Loop`.

2. \*\*Incoming Action:\*\* `Project`, `The Project file is opened`.

3. \*\*Rules:\*\*

 ```text

 // Start the loop at pad 0

 nn = 0

 perform "SetNextLight", nn

 Exit Rules, Skip Outgoing Action

 ```

4. \*\*Outgoing Action:\*\* `None`.

\*\*Step 4: Translator - Set Next Light Green\*\*

\*Purpose: Turns the specified pad green and triggers the next iteration.\*

1. Add a translator named `Set Next Light Green`.

2. \*\*Incoming Action:\*\* `Perform`, Name: `SetNextLight`, Parameters: `nn` (Assign value to variable `nn`).

3. \*\*Rules:\*\*

 ```text

 // Set light for pad nn, then trigger next iteration

 // Check if done (nn > 63) - Stop before processing pad 64

 IF nn > 63 THEN Exit Rules, Skip Outgoing Action

 // Set pad note and GREEN color for MIDI Outgoing Action

 pp = nn

 rr = 21 // Green

 // Execute the Outgoing Action for this pad FIRST

 Exit Rules, Execute Outgoing Action

 ```

4. \*\*Outgoing Action:\*\*

 \* Type: `MIDI Message`

 \* Message Type: `Note On`

 \* Channel: `6` (Solid 100%)

 \* Note: `value of variable 'pp'`

 \* Velocity: `value of variable 'rr'` (Will be 21)

 \* Select MIDI ports: `Project/Preset Default Ports`

\*\*Step 5: Translator - Continue Set Lights Loop\*\*

\*Purpose: Increments the pad counter and triggers the next light setting action.\*

1. Add a translator named `Continue Set Lights Loop`.

2. \*\*Incoming Action:\*\* `Perform`, Name: `SetNextLight`, Parameters: `nn` (Assign value to variable `nn`).

3. \*\*Rules:\*\*

 ```text

 // Increment counter and trigger next iteration

 // Check if done \*before\* incrementing to avoid going past 63

 IF nn >= 63 THEN Exit Rules, Skip Outgoing Action

 // Increment counter

 nn = nn + 1

 // Trigger the next light setting action

 perform "SetNextLight", nn

 Exit Rules, Skip Outgoing Action

 ```

4. \*\*Outgoing Action:\*\* `None`.

\*\*Step 6: Translator - Per-Track Clip State Tracker (Revised Rules)\*\*

\*Purpose: Main logic - Sets incoming pad Red (playing) or Green (stopped). Uses IF statements instead of GOTO for initial track handling.\*

1. Add a translator named `Per-Track Clip State Tracker`.

2. \*\*Incoming Action:\*\*

 \* Type: `MIDI Message`, Message Type: `Note On`

 \* Select MIDI Ports: `Project/Preset Default Ports` (Listens to `Ableton1`)

 \* Channel: `any channel, set 'oo' to channel`

 \* Note: `any note, set 'pp' to note`

 \* Velocity: `any velocity, set 'qq' to velocity`

3. \*\*Rules:\*\* (Copy-paste the revised rules below)

 ```text

 // Only process pad notes (0-63)

 IF pp > 63 THEN Exit Rules, Skip Outgoing Action

 // Determine track (column)

 tt = pp & 7

 // Store previous clip state for this track directly using tt

 IF tt == 0 THEN ga = g0

 IF tt == 1 THEN ga = g1

 IF tt == 2 THEN ga = g2

 IF tt == 3 THEN ga = g3

 IF tt == 4 THEN ga = g4

 IF tt == 5 THEN ga = g5

 IF tt == 6 THEN ga = g6

 IF tt == 7 THEN ga = g7

 // Update active clip for this track

 IF tt == 0 THEN g0 = pp

 IF tt == 1 THEN g1 = pp

 IF tt == 2 THEN g2 = pp

 IF tt == 3 THEN g3 = pp

 IF tt == 4 THEN g4 = pp

 IF tt == 5 THEN g5 = pp

 IF tt == 6 THEN g6 = pp

 IF tt == 7 THEN g7 = pp

 // Determine color based on incoming velocity (qq)

 IF qq > 0 THEN GOTO "ClipPlaying"

 GOTO "ClipStopped"

 Label "ClipPlaying"

 rr = 5 // Red

 Exit Rules, Execute Outgoing Action

 Label "ClipStopped"

 rr = 21 // Green

 // Clear this track's active clip state if it's the one that stopped

 IF tt == 0 THEN GOTO "CheckTrack0"

 IF tt == 1 THEN GOTO "CheckTrack1"

 IF tt == 2 THEN GOTO "CheckTrack2"

 IF tt == 3 THEN GOTO "CheckTrack3"

 IF tt == 4 THEN GOTO "CheckTrack4"

 IF tt == 5 THEN GOTO "CheckTrack5"

 IF tt == 6 THEN GOTO "CheckTrack6"

 IF tt == 7 THEN GOTO "CheckTrack7"

 GOTO "ExitRule" // Go directly to exit if track doesn't match

 Label "CheckTrack0"

 IF pp == g0 THEN g0 = 127

 GOTO "ExitRule"

 Label "CheckTrack1"

 IF pp == g1 THEN g1 = 127

 GOTO "ExitRule"

 Label "CheckTrack2"

 IF pp == g2 THEN g2 = 127

 GOTO "ExitRule"

 Label "CheckTrack3"

 IF pp == g3 THEN g3 = 127

 GOTO "ExitRule"

 Label "CheckTrack4"

 IF pp == g4 THEN g4 = 127

 GOTO "ExitRule"

 Label "CheckTrack5"

 IF pp == g5 THEN g5 = 127

 GOTO "ExitRule"

 Label "CheckTrack6"

 IF pp == g6 THEN g6 = 127

 GOTO "ExitRule"

 Label "CheckTrack7"

 IF pp == g7 THEN g7 = 127

 GOTO "ExitRule"

 Label "ExitRule"

 // Execute Outgoing Action (which uses rr set above)

 Exit Rules, Execute Outgoing Action

 ```

4. \*\*Outgoing Action:\*\*

 \* Type: `MIDI Message`, Message Type: `Note On`

 \* Channel: `6` (Solid 100%)

 \* Note: `value of variable 'pp'`

 \* Velocity: `value of variable 'rr'`

 \* Select MIDI ports: `Project/Preset Default Ports` (Sends to `1-APC-MINI-MK2`)

\*\*Step 7: Translator - Reset Previous Clip in Track\*\*

\*Purpose: Turns the \*previously\* playing clip pad (in the same track) green when a \*new\* clip starts playing.\*

1. Add a translator named `Reset Previous Clip in Track`.

2. \*\*Incoming Action:\*\*

 \* Type: `MIDI Message`, Message Type: `Note On`

 \* Select MIDI Ports: `Project/Preset Default Ports` (Listens to `Ableton1`)

 \* Channel: `any channel, set 'oo' to channel`

 \* Note: `any note, set 'pp' to note`

 \* Velocity: `any velocity, set 'qq' to velocity`

3. \*\*Rules:\*\*

 ```text

 // Only process pad notes (0-63) with velocity > 0 (playing clips)

 IF pp > 63 THEN Exit Rules, Skip Outgoing Action

 IF qq == 0 THEN Exit Rules, Skip Outgoing Action

 // Make sure previous clip isn't the same as current clip

 IF ga == pp THEN Exit Rules, Skip Outgoing Action

 // Skip if no previous clip (value 127 is our "no clip" marker)

 IF ga == 127 THEN Exit Rules, Skip Outgoing Action

 // Set note (ss) and color (tt) for previous clip

 ss = ga

 tt = 21 // Green color

 Exit Rules, Execute Outgoing Action

 ```

4. \*\*Outgoing Action:\*\*

 \* Type: `MIDI Message`, Message Type: `Note On`

 \* Channel: `6` (Solid 100%)

 \* Note: `value of variable 'ss'` (The previous pad note stored in `ga`)

 \* Velocity: `value of variable 'tt'` (Green color = 21)

 \* Select MIDI ports: `Project/Preset Default Ports` (Sends to `1-APC-MINI-MK2`)

---

Save your Bome project. Test the startup sequence first. If all pads turn green, proceed to test clip launching/stopping, focusing on whether Ableton sends feedback for notes 0-63 to the Bome Virtual Port.