iTar

USER MANUAL

Performance controls, MIDI implementation, and SysEx reference

REVISION	DATE	SOURCE BASIS
1.1	2026-08-31	Active August Release firmware: main-2026-0818-release-clean.c, fingerboard-2026-0813-poly-velocity.c, itar-config-2026-0813-performance.c

Product and company naming are taken from the firmware USB descriptors and source headers.

=== Codex Edit ===
Date : 2026-08-31
Changes : Correct Store/Recall for full Preset, global settings, and Guitar/iTar keymap persistence.
==========

DOCUMENT CONTROL

FIRMWARE SCOPE This revision documents the active August Release source set: main-2026-0818-release-clean.c (wrapping main-2026-0814-text-viewport.c), fingerboard-2026-0813-poly-velocity.c, itar-config-2026-0813-performance.c, and SlideExpression-2026-0729.c. Store/Recall behavior includes the 2026-08-31 full-Preset and keymap persistence correction.

See the IRIE manual for use of the iTar Remote Interactive Editor.

ITEM	VALUE
Document	iTar Firmware User Manual
Revision	1.1
Publication date	August 31 2026
USB identity	Manufacturer StarrLabs; product iTar-MIDI
BLE identity	iTar-BLE (handoff documentation; module firmware is outside this repository)
Board definition	nStrings=6, nFRETS=20

How to read this manual

Front-panel names are shown in capitals. MIDI channel numbers are shown as musicians see them (1–16); hexadecimal protocol values use the 0x prefix. “Encoded channel” means the transmitted nibble 0–15. Source uncertainties are labeled rather than inferred.

SAFETY AND HARDWARE The repository does not establish connector type, battery chemistry, charging limits, power-switch procedure, environmental ratings, or electrical safety limits. Use only the hardware documentation and power accessories supplied for the instrument.

CONTENTS

FAST START Connect by USB MIDI or pair the iTar-BLE MIDI port, select a receiving instrument, press frets to play, and tap TRIGGER to cycle P → S → T → SL. Start with octave value 3 (neutral).

1 PRODUCT OVERVIEW

The iTar firmware turns a pressure-sensitive, illuminated six-string by twenty-fret surface into a MIDI performance controller. It can send musical data over class-compliant USB MIDI and over a 31.25-kbit/s UART path used by the BLE MIDI subsystem and described in source comments as a MIDI-jack path.

Core performance features

- Four selectable play modes: Poly (P), Strum (S), Tapping (T), and Slide (SL).
- Velocity derived from the fret sensor's analog peak, transmitted as 1–127 for Note On.
- Six configurable logical string channels; defaults are MIDI channels 1–6.
- Guitar-note mapping with standard E3–A3–D4–G4–B4–E5 open-string bases.
- iTar grid mapping on MIDI channel 16 for software that treats each fret as a unique pad.
- KeyPressure output assignable to Pitch Bend, Channel Pressure, CC11, CC74, or Pitch Bend + CC74.
- Versioned SysEx for modes, octave, expression, per-string channels, and explicit Store/Recall/Factory Reset.

Signal path

SOURCE	FIRMWARE ACTION	DESTINATION
Fret and KeyPressure sensors	Scan, debounce, scale, route by mode	USB MIDI and UART/BLE when available
TRIGGER / EDIT / octave controls	Change state; send status SysEx	Local engine, LED controller, connected editor
USB MIDI input	Parse iTar SysEx; forward MIDI bytes to LED controller	Configuration engine and LPC1114 LED controller
UART/BLE input	Parse iTar SysEx; forward bytes to LED controller	Configuration engine and LPC1114 LED controller

2 SETUP AND CONNECTIONS

USB MIDI

1. Connect the iTar's USB port to a USB host using the correct cable for the hardware.
2. Select iTar-MIDI in the host's MIDI input list. The firmware declares USB 2.0, MIDI streaming, 100 mA, and both bus-powered and self-powered attributes.
3. The firmware delays USB initialization by approximately 200 ms after internal setup; enumeration then proceeds asynchronously.
4. USB is optional. Scanning and BLE/UART service continue when no USB host is configured.

Bluetooth MIDI

1. Power the instrument with the On/Off switch
2. Press the Bluttooth pairing button on the iTar.
3. Open the host's operating system's Bluetooth MIDI connection interface and select iTar-BLE. Iit will say "Connected" when paired.
4. Pairing does not require the iTar app or a SysEx handshake, but pairing to the iTar app does place the iTar into iTarMode Tuning [see Page 8]. Select the resulting MIDI port in the receiving app or instrument.

TRANSPORT BEHAVIOR Performance events are sent to USB when the endpoint is configured and mirrored to the UART/BLE path when its transmit buffer has room. Under congestion, the UART/BLE copy can be skipped; USB has priority for Slide timing.

First sound

1. Choose a MIDI sound that receives the iTar's channels. The compiled default uses channels 1–6, one logical string per channel.
2. Start in P mode and octave 3. Press a fret; pressure controls Note On velocity.
3. If using SL or KeyPressure Pitch Bend, configure the receiving sound to accept the RPN bend range sent by the iTar (default 20 semitones in Slide).
4. The iTar app has several sound sources: the embedded Sound Font Player, or any installed plugins or external apps in the same or remote devices.

3 CONTROLS AND GESTURES

CONTROL / GESTURE	ACTION	FEEDBACK
TRIG {The TriggerMode Button] : Short press and release	Cycles TapPoly → Strum → TapMono → SlideMode→ Button acts on the release edge.	Red P, S, T, or SL glyph scrolls across the fretboard. The selected Mode's SysEx is sent via USB/BLE
TRIG long press (at least 1 second), release	Toggles hammer-velocity scaling on/off without changing the selected play mode.	IRIE receives the updated hammer-scaling state; there is no dedicated local glyph.
EDIT: short press and release	Toggles between Guitar Tuning/6-channels and iTar Tuning [Notes 0-126/Channel15] *iTarTuning is used with the iTar App and Ableton Live*	Sends subtype 08 and a full runtime preset packet; no dedicated local glyph.
EDIT, hold at least 1 second, then release	Toggles KeyPressure on/off.	Two power-LED blinks = on; one = off. Each phase is 300 ms.
Octave UP short press and release	Decrements every runtime zone's encoded octave, minimum 0.	String 6 LEDs show the Octave #
Octave DOWN short press and release	Increments every runtime zone's encoded octave, maximum 5.	String 6 LEDs show the Octave #
Oct+ and Oct- combination Press and release	MIDI Panic! . Sends All Notes Off, ResetControllers	Power LED is driven during panic; MIDI reset sequence is sent.
String1&String6 on Fret 1 + EDIT combo acts on release	Toggles Right-hand/ Left-hand after first releasing sounding notes.	Two blinks = left-handed; one = right-handed.
String1&String6 on Fret 20 + EDIT combo acts on release	Stores the full Preset, global Expression/performance settings, and Guitar/iTar keymap selection to EEPROM.	Cyan SAVE for 2 s; 3 blinks success, 5 failure.
String1&String6 on Fret 19 + EDIT combo acts on release	Validates and recalls the full Preset, global settings, and keymap.	Restore: Cyan "REST "then "ORE", 1 s each; 3 blinks success, 5 failure.
Fret Keys	Generate velocity-sensitive notes and channel afterpressure if enabled. Also with Portamento if enabled.	Key lighting has several sources available : direct key-velocity, Ableton Live Session View, Clip controll Zone, Lighting scripts, text, Chord/Scale chart color maps., images.

PANEL-LABEL CONFIRMATION The source symbols OCTAVE_UP and OCTAVE_DOWN perform the opposite encoded step: UP decrements and DOWN increments. Confirm the physical silkscreen-to-mux wiring before publishing a consumer-facing control diagram.

4 OPERATING MODES

DISPLAY	MODE	PLAYING BEHAVIOR	MODE ENTRY / EXIT
P	Full Poly	Every active fret in a configured zone can sound independently. Guitar mapping uses tuning/octave/channel; iTar mapping uses unique grid notes on channel 16.	On entry, sends CC127 value 0 (Poly Mode On / All Notes Off) once per configured channel.
S	Strumming	Frets 1–16 select pitch. Frets 17–20 are the strum zone; pressing a strum-zone fret triggers the highest selected fret on that string with strum velocity. State is maintained per string.	Entry initializes strum state. Exit sends Note On velocity 0 for ringing notes.
T	Tapping	Monophonic per string. The highest held fret sounds; a higher press replaces the old note, and release falls back to the next-highest held fret.	Entry initializes per-string state. Exit sends all active notes off.
SL	Slide	The first held highest fret sends one Note On. Later highest-fret changes retarget Pitch Bend instead of retriggering while the range permits. Pull-offs can retarget lower held frets.	Entry sends RPN 0 bend range (default 20 semitones) per configured channel. Exit stops notes and restores the default 2-semitone bend range.

Slide timing

Portamento value 0 selects the adaptive algorithm. The ramp duration is based on the time since the previous target, 50% by default, with a 12 ms minimum, a 12 ms-per-crossed-semitone floor, and a 250 ms maximum. Values 1–2000 select a fixed ramp duration. The initial Note On is immediate; only the bend trajectory is timed.

CHANNEL INDEPENDENCE Slide is polyphonic across strings only when each active string has its own MIDI channel. If channels are shared, pressing a new string stops the other string states so conflicting Pitch Bend cannot detune multiple notes.

5 PLAYING, TUNING, AND MAPPING

Default Guitar mapping

LOGICAL STRING	OPEN BASE	DEFAULT CHANNEL	FRET 1…20 NOTE RANGE
1	E3 / 40	1	41…60
2	A3 / 45	2	46…65
3	D4 / 50	3	51…70
4	G4 / 55	4	56…75
5	B4 / 59	5	60…79
6	E5 / 64	6	65…84

The global octave is encoded 0–5, with 3 neutral. Each step changes pitch by 12 semitones. The firmware clamps MIDI note values to 127 but does not provide a separate underflow warning.

iTar grid mapping for the iTar App or Ableton Live

In P mode with iTar Tuninng selected, every physical fret becomes a unique note on MIDI channel 16. Logical string n uses note = (n − 1) × 21 + fret. The unplayed open positions are gaps at notes 0, 21, 42, 63, 84, and 105.

LOGICAL STRING	FRET 1	FRET 20
1	1	20
2	22	41
3	43	62
4	64	83
5	85	104
6	106	125

Handedness

When switching Right/Left hand tuning, the physical-to-logical string lookup used for tuning, channels, and iTar grid notes changes. The Store/Recall chords remain tied to the physical outer strings and therefore do not move. Handedness is part of the full Preset saved by Store and restored by Recall.

Velocity and note-off

Fret velocity is derived from the fret-key's captured analog peak and scaled to 0–127. Use the Curves and Gain settings to adjust the response as needed. Note-releases are transmitted as Note On with velocity 0; a distinct 0x8n Note Off status is not generated by the performance paths. This is done to ease MIDI Running Status when the synth allows it.

6 LED INDICATIONS

INDICATION	COLOR / PATTERN	TIMING	MEANING
Startup sign-on	Cyan ITAR, centered	2.0 s	Firmware completed initialization and started USB attach.
Mode label	Red P, S, T, or SL moving toward the bridge	24 shifts × 100 ms = 2.4 s	New TRIGGER-selected play mode.
Octave 1–5	White count on one fretboard LED string beginning at physical fret 1	900 ms	Encoded octave value; 3 is neutral.
Octave 0	One red marker at physical fret 1	900 ms	Encoded octave value zero.
KeyPressure	Power LED: 2 blinks on; 1 blink off	300 ms off/on phases	Long-EDIT result.
Handedness	Power LED: 2 blinks left; 1 blink right	300 ms off/on phases	Fret-1 outer-string chord result.
Store	Cyan SAVE; power LED 3 or 5 blinks	SAVE 2 s; 300 ms phases	3 success; 5 EEPROM write/read-back failure.
Recall	Cyan REST then ORE; power LED 3 or 5 blinks	1 s each page; 300 ms phases	3 success; 5 invalid/missing record.
USB stage diagnostic (Debug only)	Power LED: 1–4 blinks in a repeating window	Starts after 3 s; 200 ms on / 200 ms off; repeats every 2.4 s	Diagnostic only; compiled out of the active Release build.
Fret activity	LED-controller Note command, value 20	On each configured-zone fret event	Exact visible color/persistence is controlled by the separate LPC1114 firmware, not present here.

RELEASE BEHAVIOR The active Release disables the USB-stage power-LED diagnostic. The repeating 1–4 blink pattern is available only in Debug troubleshooting builds.

Internal color indices

The LPC1347 source uses LED-controller palette indices Off 0, Red 1, Green 43, Cyan 64, Blue 86, and White 127. These are palette references, not RGB triplets. The onbooard selectable Color map has 127 color locations, each defined by a 15-bit color representation.

7 MIDI IMPLEMENTATION

Implementation chart

FUNCTION	TRANSMITTED	RECOGNIZED	DETAILS
Basic channel	1–16	1–16 in configuration	Defaults: logical strings 1–6 use channels 1–6; iTar grid uses channel 16.
Mode	Mode messages through SysEx	SysEx subtypes 06 and 11	No MIDI Program Change is used to select play modes.
Note number	0–127	Not used as musical input	Guitar, Live Session, Strum, Tapping, and Slide mappings as described.
Velocity	1–127; Note On 0 for release	Not used as musical input	Pressure-derived Note On velocity.
Aftertouch—poly	No	No	Unsupported.
Aftertouch—channel	Optional	No	KeyPressure assignment 0 sends Channel Pressure/ CC11/ CC74/ PitchBend.
Pitch Bend	Yes, 14-bit	No	Slide trajectories; optional additive KeyPressure. Center 8192.
Program Change	No	No	Preset field patch exists but no active send path; unsupported.
Bank Select MSB/LSB	No	No	CC0/CC32 unsupported.
System real-time	No	Ignored on SysEx input	Bytes 0xF8–0xFF are discarded by the configuration parser.
MIDI Clock / transport	No	No	Unsupported.
Active Sensing	No	No	Unsupported.
System Reset	No	No	MIDI 0xFF is ignored by the SysEx parser.
System Exclusive	Yes	Yes	Starr Labs manufacturer ID 00 01 42; implementation in Section 8.

Control Change and RPN messages

CC / RPN	VALUE	WHEN	CHANNELS
CC11 Expression	Configured continuous value; default held-note value 127; Panic recovery 127	Optional legacy expression; KeyPressure assignment; after Panic	Performance string channel or all 16 during recovery
CC74 Timbre	0–127	KeyPressure assignment 1 or 4	Performance string channel
CC112	127 default	Optional Panic preamble	All 16
CC121 All Controllers Off	0	Panic	All 16
CC123 All Notes Off	0	Panic	All 16
CC127 Poly Mode On	0	Entering P mode; also implies All Notes Off	Each configured channel once
RPN 0 Pitch Bend Sensitivity	MSB = semitones; LSB = 0	Entering Slide / preparing channel; restored on exit	Each prepared Slide channel
CC101 / CC100	0,0 then 127,127	Select and null RPN 0	Each prepared Slide channel
CC6 / CC38	Range semitones / 0	Data Entry for RPN 0	Each prepared Slide channel

Panic sequence

1. If enabled, send configured Panic CC/value on channels 1–16 (default CC112/127).
2. If in SL mode, stop Slide notes and restore prepared channels' bend ranges.
3. For channels 1–16, send Pitch Bend center, CC123/0, and CC121/0.
4. Send CC11/127 on channels 1–16 because All Controllers Off commonly resets expression to zero.
5. Set the internal mode to Panic. The next TRIGGER press returns through the normal mode list starting from P.

8 SYSTEM EXCLUSIVE IMPLEMENTATION

Envelope

All documented configuration messages use the following byte sequence:

WIRE FORMAT F0 00 01 42 tt cc ss mm ll [payload…] F7

BYTE	NAME	MEANING
F0	Start	System Exclusive start
00 01 42	Manufacturer	Starr Labs three-byte manufacturer ID
tt	Type	01 configuration; 40 LED framebuffer family
cc	Checksum	Compact messages use 00; legacy preset uses additive 7-bit checksum
ss	Subtype	Command / response selector
mm ll	Size	14-bit payload length, MSB then LSB
payload	Data	Expected to be 7-bit clean
F7	End	End of Exclusive

Validation actually performed

- The parser validates manufacturer 00 01 42 and type 01 after all eight header bytes have been accumulated.
- Compact versioned payloads validate exact version/length pairs in their command handlers. Values are bounded or masked when applied.
- The parser does not compare the header's 14-bit size against received payload length for every legacy command.
- The parser does not validate the header checksum on received packets. The legacy preset checksum is calculated for storage/transmission but is not enforced on ingress.
- There is no general NAK or error frame. Unsupported versions, invalid sizes, and invalid zone indices commonly produce no response.
- The CRC-8 described later protects the internal EEPROM global record; the full Preset uses its established additive 7-bit checksum. Neither value is a generic checksum for compact SysEx commands.

HOST REQUIREMENT Send one complete, 7-bit-clean message at a time and wait for the expected response. Do not treat silence as success.

Command and response map

SS	NAME	DIRECTION	PAYLOAD / RESPONSE
01	Legacy full preset	Both	Native Preset bytes; receive applies, stores, returns subtype 04 and also echoes the full config.
02	Config reset	Host → iTar	Reserved/incomplete. Creates a zeroed temporary object only; no reset or confirmation.
03	Request	Host → iTar	Legacy handshake; returns subtype 04 firmware version.
04	Response	iTar → host	One byte; firmware version 01.
05	Octave	iTar → host	One byte 00…05; 03 neutral. No dedicated receive handler.
06	Legacy Trigger Boolean	Both	00=P request / triggers off state; 01=T request / triggers on state. S and SL report 00.
07	LED config	Host → iTar	Handler present but LED initialization is disabled; no functional configuration action.
08	iTar mapping	Both	One byte: 00 Guitar mapping, nonzero iTar grid mapping.
09	Expression settings	Both	Versioned 6/18/24/26-byte set or canonical v5 response.
0A	Expression request	Host → iTar	Zero payload; returns 06, 11, 05, then 09.
0B	String channels	Both	8 bytes: version, zone, six encoded channels.
0C	String channels request	Host → iTar	2 bytes: version, zone; returns 0B if valid.
0D	Store	Host → iTar	Zero payload; stores and verifies the full Preset plus global v5 settings/keymap, then returns subtype 10 status and current compact state.
0E	Recall	Host → iTar	Zero payload; validates both records before applying either, echoes the restored full Preset, then returns subtype 10 status and current compact state.
0F	Factory Reset globals	Host → iTar	Zero payload; resets/stores global mode, octave, Expression; same response sequence.
10	Persistence status	iTar → host	3 bytes: version 01, operation 0D/0E/0F, result 00/01.
11	Exact play mode	Both	One byte: 00=P, 01=S, 02=T, 03=SL.

Expression settings, subtype 09 version 5

BYTE(S)	FIELD	RANGE / DEFAULT
0	Version	05
1	Flags	bit0 legacy expression; bit1 Panic CC; bit2 KeyPressure enabled; bit3 editor owns enable
2	Legacy expression CC	0–127 / 11
3	Legacy held-note value	0–127 / 127
4–5	Portamento ms, 14-bit	0 adaptive; 1–2000 fixed / 0
6	Pitch Bend range semitones	1–127 / 20
7	Bend points per semitone	1–16 / 5
8	Gesture time used	1–100% / 50
9–10	Minimum glide ms	0–2000 / 12
11–12	Minimum ms per semitone	0–2000 / 12
13–14	Maximum adaptive glide ms	minimum…2000 / 250
15	Panic CC	0–127 / 112
16	Panic value	0–127 / 127
17	Bend range restored on exit	1–127 / 2 semitones
18	KeyPressure assignment	0 Channel Pressure; 1 CC74; 2 Pitch Bend; 3 CC11; 4 Pitch Bend + CC74; default 2
19	Pressure curve	1 linear; 2 quadratic / 2
20–21	Soft threshold, 14-bit	0–2046 / 600
22–23	Hard threshold, 14-bit	soft+1…2047 / 2047
24–25	KeyPressure Pitch Bend depth cents, 14-bit	0–2400 / 100

Versions 1–4 remain accepted only at their exact defined lengths: v1 6 bytes, v2 18 bytes, and v3/v4 24 bytes. Missing newer fields preserve their current values. A successful set returns canonical version 5.

Examples

Request current global state: F0 00 01 42 01 00 0A 00 00 F7

Factory Reset globals: F0 00 01 42 01 00 0F 00 00 F7

Successful Store status: F0 00 01 42 01 00 10 00 03 01 0D 01 F7

Per-zone string channels

Subtype 0B payload: version 01, zone 00–04, then six channel values encoded 00–0F. Display them as MIDI channels 1–16. Handedness is applied after this logical-string array.

Set Zone 0 to channels 1–6: F0 00 01 42 01 00 0B 00 08 01 00 00 01 02 03 04 05 F7

Request Zone 0: F0 00 01 42 01 00 0C 00 02 01 00 F7

Legacy full preset, subtype 01

FIELD	ENCODING
presetId	4 native bytes
left-handed	1 native bool
velocity mode	1 byte
KeyPressure enabled	1 native bool
Strum enabled	1 native bool
sensitivity gain	1 byte
zone count	1 byte, maximum 5
each Zone	start/end fret, low/high string, patch, octave, LED flags, six tuning bytes, six channel bytes
checksum	Additive sum of active preset bytes, masked to 7 bits; placed in header for transmission/storage

LEGACY LIMITATION This packet mirrors a native C structure and is therefore not a stable long-term interchange format. The send routine and header-size calculation also differ in how they derive the active byte count. New software should use versioned compact messages and treat the legacy packet as compatibility-only.

LED packet type 40

Type 40 is reserved for the separate LED-fretboard processor. Source constants define subtypes 00 color map, 08 tuning, and 09 zone. The iTar main CPU forwards inbound USB/UART MIDI bytes toward the separate LED controller where Clipper command messages are implemented on the Fretboard lighting.. Direct third-party use is unsupported by this manual but nonetheless encouraged. Refer to the Clipper command chart for details.

9 STORE, RECALL, AND RESET

What Store includes

- The full 112-byte Preset: handedness; velocity mode; Sidebend and Strum flags; sensitivity; zone count; and each active zone's bounds, patch, octave, poly/LED flags, six tunings, and six MIDI channels.
- The versioned global record: Expression version 5 settings, exact P/S/T/SL mode, global octave, Hammer Up/Down, and hammer-scaling enable/percentage.
- The current note-keymap selection: Guitar mapping or the unique iTar grid mapping.

What it does not include

- The live IRIE per-zone runtime sidecar (zone type, route, LED owner, and related transient metadata), LED-controller color maps/images, or other lighting-processor RAM.
- USB/BLE connection state, receiving-synth configuration, or other host-only state.

EEPROM record validation

Store writes two EEPROM records. The 112-byte Preset begins at address 0x0004 and uses the established additive 7-bit checksum over its active fields. The version 5 global record begins at 0x0080 and is 39 bytes: ASCII IGLB, record version 05, payload length 20, the 26-byte Expression payload, mode, octave, performance flags, hammer-scaling enable/percentage, Guitar/iTar keymap selection, and CRC-8. The CRC uses polynomial 0x07 with initial value 0. Recall validates both records before changing live state; either invalid record makes the Recall fail without applying either one. Global record versions 1–4 remain readable and default an absent keymap field to Guitar mapping.

OPERATION	REMOTE COMMAND	ONBOARD GESTURE	FAILURE BEHAVIOR
Store	Subtype 0D, zero payload	Outer strings Fret 20 + EDIT release	Status result 00; five blinks. A write/read-back failure in either record makes the combined Store fail.
Recall	Subtype 0E, zero payload	Outer strings Fret 19 + EDIT release	Current live state remains unchanged; status result 00; five blinks. Either the Preset checksum/zone count or the global magic/version/length/CRC failed validation.
UI SAVE button	Legacy Preset subtype 01, or combined Store subtype 0D, depending on the IRIE action	Not applicable in the current hardware	Subtype 01 saves the full Preset only. Subtype 0D saves and verifies the full Preset plus global settings and the current Guitar/iTar keymap.
Factory Reset globals	Subtype 0F, zero payload	No onboard gesture	Compiled defaults are applied, then storage result is reported.

Factory defaults

P mode, octave 3 neutral, legacy fixed expression off, Panic CC enabled at CC112/value 127, KeyPressure off but preselected for Pitch Bend, 20-semitone Slide range, 2-semitone restore range, adaptive timing, quadratic pressure curve, thresholds 600/2047, and 100-cent KeyPressure depth.

POWER-CYCLE POLICY Release automatically restores the versioned global record (including keymap), but the current build policy does not automatically restore the full Preset. Explicit Recall restores and validates both records. Debug starts from compiled global defaults unless explicitly configured otherwise.

Firmware update behavior

No user-facing bootloader, firmware-update gesture, update transport, or image-validation routine is implemented in the inspected iTar project. Firmware installation currently requires the development/programming workflow outside this manual. Do not advertise USB, BLE, or MIDI firmware updating from this source.

10 TROUBLESHOOTING

SYMPTOM	CHECK	SOURCE-BASED ACTION
No USB MIDI device	Power LED USB diagnostic after 3 s	1 blink: host reset not seen; 2: reset only; 3: SETUP traffic; 4: configured. Try a known data cable/port and reconnect after boot.
BLE pairs but no notes	Host selected iTar-BLE as MIDI input; receiving channels	Use P mode and listen on channels 1–6. BLE pairing itself does not require an app handshake.
One string bends other strings	Channel assignment	Give each active Slide string an independent MIDI channel using subtype 0B.
Slide interval is wrong	Receiver bend range	Allow the iTar RPN 0 sequence or manually set the receiver to the firmware's configured range (default 20 semitones).
KeyPressure does nothing	Enable state and assignment	Hold EDIT at least 1 s and release; two blinks means on. Confirm synth supports the selected Pitch Bend/Channel Pressure/CC destination.
Pitch remains raised	KeyPressure pressure and Panic	Release pressure. If needed, hold both octave controls for Panic; it sends center bend and controller resets on all channels.
Unexpected notes after short EDIT	Mapping mode	Short EDIT toggles Guitar/iTar mapping. In iTar mapping P mode uses channel 16 and grid note numbers.
Store/Recall fails	Five power-LED blinks; combined Store/Recall status reports failure	Repeat while musically idle. Recall requires both a valid Preset and a valid global record.
Settings do not return after power cycle	Preset restore policy	Use explicit Recall after power-up to restore the full Preset. Release restores global settings/keymap automatically; Debug does not by default.
Mode label or note LED looks wrong	Separate LED controller	The LPC1347 sends commands to an LPC1114 whose source is not in this project. Verify matching LED-controller firmware.

Recovery sequence

1. Return the receiving synth to a known patch and listen on channels 1–6.
2. Select P mode and octave 3.
3. Perform Panic with both octave controls.
4. Short-press EDIT if the controller is accidentally in iTar grid mapping.
5. If a valid stored Preset and global record exist, perform manual Recall; otherwise use remote Factory Reset for global defaults and resend/save the desired Preset from IRIE.

Clipper™ Command charts and appendices Starr Labs (c) 2012-2026

LED Fingerboard Command Specification

24-fret instruments use MIDI Channels 15 and part of 16 for addressing the 144 LEDs on the fretboard.

20-fret instruments like the iTar have 126 LEDs and are addressed full on MIDI channel 15

Special MIDI-Note Functions on Channel 16 :

[Notice that the minimum velocity value available for your use is the number 1. Velocity = 0 is reserved for the Note-Off function.]

Command	#	Comments
CLEAR FRAME	127	Ignore velocity, clears to 0 unless clear to background mode set
INIT ZONES	126	ignore velocity
REFRESH	125	ignore velocity
ENABLE ZONE	124	velocity = ZONE 1 - 31, bit6 1 = enable 1 = disable (Displays Zone lighting. Bring to foreground)
LEFTRIGHT	123	velocity 2 = LEFT, 1 = RIGHT, mirror about the Y-axis
SHIFT RIGHT	122	velocity offset 0:no effect, =>1:Shift Right one position Bit6=0: Rotate (Wrap) image, Bit6=1: Do not wrap image. Rotate Right: vel<64; Shift Right: vel =>64D
SHIFT LEFT	121	velocity offset 0:no effect, =>1:Shift Left one position Bit6=0: Rotate (Wrap) image, Bit6=1: Do not wrap image. Rotate Left: vel<64; ShiftLeft: vel =>64D
SCROLL UP	120	velocity offset: 0:no effect, =>1:Shift Up one position Bit6=0: Rotate (Wrap) image, Bit6=1: Do not wrap image. Rotate Up: vel<64; Shift Up: vel =>64D
SCROLL DN	119	velocity offset: 0:no effect, =>1:Shift Down one position Bit6=0: Rotate (Wrap) image, Bit6=1: Do not wrap image. Rotate Down: vel<64; Shift Down: vel =>64D
BLINK ON(upper 16)	118	velocity + 128 = led#
BLINK ON(lower 128)	117	velocity = led#
BLINK OFF(upper 16)	116	velocity + 128 = led#

BLINK OFF(lower 128)	115	velocity = led#
BRIGHT(upper 16)	114	velocity + 128 = led#
BRIGHT(lower 128)	113	velocity = led#
DIM(upper 16)	112	velocity + 128 = led#
DIM(lower 128)	111	velocity = led#
RedBox Scroll Up	110	Velocity: Ignore - not used by ledfb firmware, output to Ableton
RedBox Scroll Down	109	Velocity: Ignore - not used by ledfb firmware, output to Ableton
RedBox Scroll Right	108	Velocity: Ignore - not used by ledfb firmware, output to Ableton
RedBox Scroll Left	107	Velocity: Ignore - not used by ledfb firmware, output to Ableton
• Unused -	106	• unused -
REVERSE LEDS	105	Velocity: 0-64 = Normal, 65-127 = Reverse
TABLE TOP GUITAR	104	Mirror about X-axis. Velocity 7F=Tabletop: body is low LED number Velocity 00=Guitar: head is low LED number
Set Char (lower 128)	103	velocity = text character in lower 128 of ASCII table
Set Char (upper 128)	102	velocity + 128 = text character in upper 128 of ASCII
SET CHAR DIRECTION	101	velocity sets the direction Chars scroll from, 0 from head, >0 from body
SELECT SCRIPT [Available scripts are listed in a table below.]	100	Velocity selects a Script, 0 aborts. A Script is an embedded macro that can display images, text, and/or lighting effects. The opening splash screen is a Script. Any script is aborted by asserting Select-Script with a value=0.
ENABLE BACKGROUND	99	Enable Background: velocity 1 = clear to black, velocity 2 = Enable background. When an LED is turned on, then turned off according to some dynamic function, the off-state may be either an image saved to the background or simply black.
SET COLOR MAP	98	velocity is color map number; Max = 4. *See Pg.18, Maps*
SELECT IMAGE	97	velocity is the number of the image to display. Image# 0-31 are RAM, images 32-128 are ROM based images
SET STRING 6	96	Set all LEDs on a string to the same color. Velocity = color index
SET STRING 5	95	Velocity = color index
SET STRING 4	94	Velocity = color index
SET STRING 3	93	Velocity = color index
SET STRING 2	92	Velocity = color index

SET STRING 1	91	Velocity = color index
Display Tuning Map	90	Velocity = zone number*
Send To Background	89	Send the foreground image to the background. Velocity is ignored.
Set Tuning	88	Pre-defined Tuning Maps are stored in ROM. The full list is provided below. Velocity = Tuning map number. Guitar, Piano, etc. 1=ClearTuning, which clears the background. The full list is described on Page 11.
SCROLL-IMAGE	87	A selected image is scrolled over the short dimension of the fingerboard. Velocity = step timing for the Scroll delay. Velocity = color map index# of the foreground color.
SET FOREGROUND COLOR SET BACKGROUND COLOR	86	Velocity = color map index# of the background color.
	85	
Rotate RedBox 90°	84	Velocity: Ignore - not used by ledfb firmware, this talks to Ableton Live and Rotates the Ableton RedBox
SYNC	83	Reset the Blink timer and other internal step-timers.

MIDI-Note Messages on Channel 16 continued:

OFFSET FOREGOUND COLOR INDEX	82	Velocity is added to all color selections for indexed-color pixels, i.e., foreground colors. If the resulting color-index exceeds the length of the color map, the value wraps around to the beginning of the table.
SET FRETMARKS OVERLAY COLOR	81	Velocity = color for the Fretmarks
ENABLE FRETMARKS OVERLAY	80	Velocity = 1 disables the Fretmarks. Velocity = 2 enables the Fretmarks

MIDI-CC Messages on Channel 16 (Channel 15- Unused):

Latch LEDs	64	Off < 65, On > 64; Once set, all LEDs will remain lit until the command is reset.

MIDI-CC Messages used by the LED fingerboard. Only on channel 16

BRIGHTNESS	7	Scale global brightness from Master Volume, Channel 16
SCALE RED	20	Adjust the brightness for all Red LEDs on the
SCALE GREEN	21	Adjust the brightness for all Green LEDs on the
SCALE BLUE	22	Adjust the brightness for all Blue LEDs on the
SCALE SATURATION	23	0-127; Scales the intensity of all colors. Applies to images, tuning maps, and foreground colors.
VU String 1 *	24	Fires all of the LEDs on String #1 in sequence (Data value= length of the lit-string. All LEDs are lit from Fret1 to the scaled data-level) CC-data values of 0-127 scale from no lights to all 24 columns lit. All LEDs will light from Fret1 up to the scaled data-value. Each Fret (column) corresponds to a data increment of a little more than 5.
VU String 2	25	Fires all of the LEDs on String #2 (in sequence, value= length) lights sequential number of frets according to MIDI data 0-127
VU String 3	26	Fires all of the LEDs on String #3 (in sequence, value= length) lights sequential number of frets according to 0-127
VU String 4	27	Fires all of the LEDs on String #4 (in sequence, value= length) lights sequential number of frets according to 0-127
VU String 5	28	Fires all of the LEDs on String #5 (in sequence, value= length) lights sequential number of frets according to 0-127
VU String 6	29	Fires all of the LEDs on String #6 (in sequence, value= length) lights sequential number of frets according to 0-127
VU All Strings	30	Fires all of the LEDs on all Strings (in sequence, value= length) lights sequential number of frets according to 0-127
VU Fret 1 **	31	Lights LEDs sequentially up fret 1 like a VU meter. CC-data values of 0-127 scale from no lights to all 6 rows lit in 12 increments. From bottom to top: Rows 1-4= green, Row5=yellow, Row6=Red. Each key lights at half-brightness for the lower half of its own range. Any given data value will light all keys from the bottom to the scaled level. Each lighting increment corresponds to a data increment of about 10 and a half. For example, a datum110 (decimal) lights Row6 (Red) at half-brightness, Row5 (Yellow) full-brightness, and Rows4-1 (Green) full-brightness.
VU Fret 2	32	Lights LEDs sequentially up fret 2 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 3	33	Lights LEDs sequentially up fret 3 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.

VU Fret 4	34	Lights LEDs sequentially up fret 4 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 5	35	Lights LEDs sequentially up fret 5 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 6	36	Lights LEDs sequentially up fret 6 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 7	37	Lights LEDs sequentially up fret 7 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 8	38	Lights LEDs sequentially up fret 8 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 9	39	Lights LEDs sequentially up fret 9 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 10	40	Lights LEDs sequentially up fret 10 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 11	41	Lights LEDs sequentially up fret 11 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 12	42	Lights LEDs sequentially up fret 12 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 13	43	Lights LEDs sequentially up fret 13 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 14	44	Lights LEDs sequentially up fret 14 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 15	45	Lights LEDs sequentially up fret 15 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 16	46	Lights LEDs sequentially up fret 16 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 17	47	Lights LEDs sequentially up fret 17 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 18	48	Lights LEDs sequentially up fret 18 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 19	49	Lights LEDs sequentially up fret 19 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 20	50	Lights LEDs sequentially up fret 20 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 21	51	Lights LEDs sequentially up fret 21 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 22	52	Lights LEDs sequentially up fret 22 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
VU Fret 23	53	Lights LEDs sequentially up fret 23 like a VU meter. Value of 0-127 scales from

		no lights to all lit, fading in brightness and color.
VU Fret 24	54	Lights LEDs sequentially up fret 24 like a VU meter. Value of 0-127 scales from no lights to all lit, fading in brightness and color.
LATCH MODE	64	(MIDI Sustain) When Latch Mode is On, incoming Note-Off messages will be ignored rather than turning off their intended LEDs. Velocity < 65 = Off, velocity > 64 = On.
FADE TIME	72	CC-data sets the duration the LEDs remain illuminated after being lit from a Note-On event. Similar to a synthesizser's ADSR Release time.
SET-SCRIPT-CLOCK-COARSE	75	Sets the coarse timing delay for clocked script-looping functions like the Swirl effect in 10 millisecond increments. Delay = velocity x .01 seconds.
SET-SCRIPT-CLOCK-FINE	76	Sets the fine timing for clocked script-looping functions like the Swirl effect in 0.1 millisecond increments. Delay = velocity x .0001 seconds. The coarse and fine values are additive.
STORE-RAM-IMAGE	77	The dynamic LED data frame buffer is copied to a RAM image location. Velocity = Image number (v = 0-9) The duration of the lights from pressed keys; similar to a synth "release time".

Clipper Setup Charts

Embedded lighting Scripts:

[Set-script command: Note 100]

0: Null script
1: Abort Script. Aborts any active script.
2: Grand Demo. Combines several scripts and the Starr Labs sign-on screen.
3: Product_ID. Current Clipper OS version#.
4: Null
5: Spiral Effect. Single shot event.
6: Spiral Script. Loops the spiral effect.
7: Swirl Effect. Single shot event.
8: Swirl Script. Loops the swirl effect.
9: Boustrofedon. Serpentine effect.
10: Boustrofedon Script. Loops the single shot event.
11: Show current color map.
12: Scroll RAM Images.
13: Scroll ROM images.
14: Cylon script

Embedded Color Maps:

[One of these Color Maps is loaded into the default 127-location Color Map table. When you apply a MIDI note to light up a location on the fingerboard, the velocity selects a color from the default table.]

0: Null Map, blank
1: Ableton Live color map. 60 colors. Duplicated to (mostly) fill the 127 location map.
2: Spectral color map. Full color in the traditional rainbow sequence.
3: Tertiary color map. The basic 12 colors resulting from the simple combinations of the R,G, and B LEDs.
4: Velocity color map. A four-color map to clearly distinguish MIDI velocity levels.

TUNING MAPS: [maps of 7/7/17]

- Clear: No background lighting.
- Fret Marks: shows guitar fret marks at the standard fret locations.
- Spectral fourths: 7 colors are assigned to the natural-notes, the white keys of the piano. The piano's black keys are defaulted to OFF. Straight Fourths tuning.
- Spectral fourths/Minus-6 : 7 colors are assigned to the natural-notes, the white keys of the piano. The piano's black keys are defaulted to OFF. Straight Fourths tuning. Expands the range of the 12-string Clipper.
- WhiteBlack Fourths: Two colors as on the Piano. Straight Fourths tuning starting at a low F natural.
- WhiteBlack Fourths/Minus-6: Two colors as on the Piano. Straight Fourths tuning starting at a low B natural.

Using the MIDI command Note 88/velocityN, select the desired Tuning Map and then use the SetToBackground command. DisplayTuningMap Note- 90 (only needed one time)

Here is an example to set up a 12-sttring Clipper with a Spectral Tuning (rainbow colors for the natural notes.).

```
Refer to the manual to see the command chart where these commands reside.

Low 6 strings:

Set the Tuning map to Spectral 4ths/ -6 (Low B):
Go to an unused Sensor-Pad.
Set the Pad to Notes-Basic or -All.
In Event #1 set it to Channel 14, Note 88 and for the velocity, instead of
"PAD" set it to 4.
In the Output screen, set it to MIDI: R
Press the Pad.
If it doesn't take at first try setting the pad to Note 90 and press the Pad..
Repeat if necessary.

For the high 6 strings:

Do the exact same thing as for the Lower 6 strings, but for two changes:
1)change the MIDI channel to 16.
2)change the TuningMap selection to Note88-Velocity 3.
```

Your Clipper may have an alternative Map table like this due to different firmware revisions:

- Fret Marks: shows guitar fret marks at the standard fret locations.
- Tertiary guitar: 12 colors, 1 per chromatic scale note-name. Guitar tuning.
- WhiteBlack (2 color): The fingerboard is tuned to standard Guitar tuning and the keys are coded according to the Black and White leys of the piano
- Tertiary fourths: 12 color, 1 per chromatic scale note-name. Straight Fourths tuning.
- WhiteBlack Fourths: Two colors as on the Piano. Straight Fourths tuning starting at a low Open E natural.

11 SPECIFICATIONS

ITEM	SOURCE-VERIFIED VALUE
Controller MCU	NXP LPC1347, Cortex-M3; 64 KiB flash / 8 KiB local RAM target definition
Playing surface	6 strings × 20 compiled frets
Controls	TRIGGER, EDIT, two octave-control mux inputs, fret sensors, per-string KeyPressure sensors
Play modes	P Full Poly; S Strumming; T Tapping; SL Slide
USB	USB 2.0 class-compliant MIDI streaming; VID 0x1FC9; PID 0x1003; endpoint size 16; cable 0
USB name	Manufacturer StarrLabs; product iTar-MIDI; ports iTar Out / iTar In
Serial MIDI path	UART 31,250 bit/s; used by BLE subsystem and source-described MIDI-jack path
BLE MIDI name	iTar-BLE in handoff documentation
Default channels	Logical strings 1–6 → MIDI channels 1–6; iTar grid → channel 16
Default tuning	E3, A3, D4, G4, B4, E5 (MIDI 40,45,50,55,59,64)
Octave	Encoded 0–5; 3 neutral; ±12 semitones per step
Velocity	Fret pressure-derived Note On velocity; Note On velocity 0 release
Pitch Bend	14-bit; Slide default sensitivity ±20 semitones; restores ±2 semitones
Global EEPROM	Address 0x0080; 39-byte v5 record; CRC-8 polynomial 0x07; record versions 1–4 remain readable
Preset EEPROM	Address 0x0004; 112-byte native Preset; additive 7-bit checksum
LED record	Address 0x0100; initialization disabled in active config path
Power / dimensions / weight / operating environment	Not specified by the inspected repository

APPENDIX A FIRMWARE TRACEABILITY

The separate source-map file in the output folder contains the detailed file/function matrix.

APPENDIX B PUBLICATION CONFIRMATIONS

CONFIRMATION NEEDED	WHY IT MATTERS
Confirm the exact production-flash image and external version label.	The active Release source now builds successfully, but production flashing and acceptance on target hardware remain owner-controlled.
Which physical octave label maps to mux 13 and mux 14?	Symbolic UP decrements octave and symbolic DOWN increments it in active source.
Should TRIG double-press be implemented as a quick SlideMode switch?	No double-press detection exists in the active source: short press cycles modes; long press toggles hammer-velocity scaling.
Is a physical 5-pin DIN MIDI connector fitted?	Not currently
Confirm power, charging, connector, environmental, size, and weight specifications.	These are not defined by firmware and were intentionally not guessed.
Confirm the separate LED-controller firmware revision.	Exact note-lighting colors, persistence, and type-40 behavior depend on that missing source.
Confirm full Preset Store/Recall on target hardware and with the shipping IRIE build.	The source and Release binary now save/verify both EEPROM records, but hardware EEPROM retention and IRIE interoperability still require an end-to-end test.