CTR2-MIDI

Flex Mode

Manual

v3.00.00



Last Revision: November 3, 2026

Copyright 2025-26: Lynovation, LLC
All rights reserved

Updated to firmware v3.00.00

Revised sections for this version are highlighted in yellow

Contents

Introduction 4
MIDI Mode 4
Flex Mode 4
Legal Notice 4
How to use this manual 5
Change Log 5
v3.00.00: November 3, 2026 5
Quick Start Guide 6
Basic Concepts 6
Initial Startup 7
Configure WiFi and Radio Address 7
Connecting to WiFi 7
Configure AetherSDR's CTR2 Relay Mode 8
Change Knob and Input Jack Mode 9
Configure the Keyer 9
Configure the *MIDI's* Controls 9
Keywords and Definitions 10
Hardware 11
Operating Modes 11
Switching Modes 12
Design Philosophy 12
Morse Prompts 13
Boot Sequence 13
Configuring WiFi and Radio IP Settings 13
Flex Mode Options 14
Extended BTN Mode 14
Paddle Input Jack Mode 15
CW Macros 15
BTN#+Knob Options 15
Beep Mode (BTN1+Knob) 16
Speed Tuning Mode (BTN2+Knob) 16
Future (BTN3+Knob) 16
Run CW Report (BTN4+Knob* when Paddle Input = Paddles) 16
PTT Mode (BTN4+Knob* when Paddle Input = Key/PTT) 17
Switch Modes (BTN5+Knob*) 17
Connect/Disconnect WiFi (BTN6+Knob*) 18
Connecting to WiFi 18
Disconnecting from WiFi 18
Flex Terminal Menus 18
Home Menu 19
WiFi Menu 20
Keyer Menu 20
Edit CW Macros and Call Sign Menu 21
Control Menu 21
Knob Mode Settings 21
Available Knob Functions 22

Button Settings 22
Available Button Functions 22
Extended BTN Mode 23
Maintenance Menu 23
Export Settings 23
Import Settings 24
Recalibrate Buttons 25
Reset to Factory Settings 25
Appendix A: Installing or Updating CTR2-MIDI Firmware 26
Installing Firmware using ESPConnect (New Method) 26
Installing Firmware using EspressIF Flash Download Tool (Old Method) 29
Installing Firmware using Linux or Mac 31
Appendix B: Running CTR2-MIDI on CTR2-Micro Hardware 33
Flashing the Firmware 34
Operating the *Micro* with *MIDI* Firmware 34
LED Indications 34
Modified BTN#+Knob Functions 35
Appendix C: Configuring Tera Term 35
Appendix D: Configuring Putty 36
Serial Session 36
Appendix E: Apple or Linux Terminal Programs 37
Appendix F: Default Knob and Button Assignments 39
Appendix G: Flex Mode Button Assignment Worksheet 40
Appendix H: Adding a Volume Control Pot 43
Appendix I: Change Log 44

Introduction



CTR2-MIDI (called the ***MIDI*** in this document) is a small radio controller designed for ham operators. It utilizes dual-boot firmware that allows you to boot in either **MIDI** controller firmware, or **Flex** controller firmware.

MIDI Mode

Booting into **MIDI** mode allows you to control MIDI enabled radio control apps using Bluetooth or USB MIDI control.

Flex Mode

Beginning with v2.00.00 firmware, you can choose to boot **CTR2-MIDI** into **Flex** mode. In this mode, the controller connects directly to the network server in your Flex 6xxx/8xxx/Aurora radio allowing you to control it using the Flex API over WiFi. This mode works with any Flex GUI client such as SmartSDR, AetherSDR, RimuSDR, etc., including Window versions. It does not, however, support SmartLink for remote control. The radio and ***MIDI*** must be on the same network.

Beginning with v3.00.00 firmware, **Flex** mode supports the new **CTR2 Relay** feature that was added to AetherSDR v26.10.3 and above. Just plug your ***MIDI*** into a USB port on the computer running AetherSDR and **CTR2 Relay** creates a virtual connection between ***MIDI*** and your radio, with no WiFi required!

For clarity, this manual only covers **CTR2-MIDI's Flex** mode. A separate manual covers **CTR2-MIDI's MIDI** mode. That manual can be downloaded here.

CTR2-MIDI firmware can run on **CTR2-Micro** hardware using Bluetooth MIDI mode or **Flex** mode when connecting via WiFi. **USB** connections for MIDI and AetherSDR's **CTR2 Relay** are not supported on **CTR2-Micro**. Instructions and limitations can be found in Appendix B.

Legal Notice

What would a manual be without a legal notice? Here goes…

- This is a hobby endeavor. Nothing is guaranteed! Use this device at your own risk!
- I will do my best to make sure you receive functioning hardware if you buy a CTR2-MIDI and will work with you if there is a problem with your unit on arrival.
- I cannot guarantee or warranty the hardware supplied in the kit (kits have been discontinued).
- I make no warranty that the firmware provided for CTR2-MIDI will perform up to your expectations or be suitable for your application. Software bugs are a fact of life and I try to find and correct all bug reports to the best of my ability ASAP.

How to use this manual

This manual should be used as a reference manual. An expanded Help system if you will. Items in the Table of Contents link to their write up in the manual. The main categories have short write ups describing the functions available in that section. I've tried to group things logically and have added hyperlinks so you can quickly jump to other sections.

As this document evolves, sections that have changed since the last update will be highlighted in yellow.

The version number of this manual will follow the latest released version number of the firmware.

Feel free to contact me if you have question about a certain feature or have ideas for future improvements. I love to get feedback on my work. My email address is good on QRZ.com.

Change Log

This manual covers **CTR2-MIDI's Flex** mode operation. For information on **CTR2-MIDI's MIDI** mode, download the CTR2-MIDI MIDI Mode Manual.

v3.00.00: November 3, 2026

- Added support for AetherSDR's new **CTR2 Relay** USB mode
- Removed "WiFi" from **Flex** mode references since **CTR2 Relay** mode allows a USB connection to your radio in AetherSDR

v2.01.03: October 29, 2026

- No changes to **Flex** mode.

v2.01.02: September 29, 2026

- Renamed this manual from **CTR2-MIDI Flex Operation Manual to CTR2-MIDI Flex Mode Manual**
- Fixed bugs causing erratic tuning when in RTTY mode, or when a frequency offset has been enabled for repeater operation using a transverter.
- Added RTTY, NFM, and DFM modes and default filters
- Added an UNKN mode for when the radio sends a mode that **CTR2-MIDI** doesn't have configured.

v2.01.01b: May 25, 2026

- Added a note in the Keyer menu about disabling the CTR2 keyer so you can use the radio's keyer

v2.01.01a: April 15, 2026

- Added additional information and a warning about the Paddle/Key/PTT jack's function

Changes to previous versions can be found in Appendix I.

Quick Start Guide

If you're like most hams, when you receive a new tool for your shack you just want to plug it in and try it out.

CTR2-MIDI is not one of those tools!

v3.xx.xx ***MIDIs*** have two operating modes, MIDI mode, and Flex mode. If you are planning on using your ***MIDI*** in **MIDI mode**, download that manual here and follow its **Quick Start Guide**.

This manual has more information than you probably want to know. The goal of this section is to give you a basic understanding of the ***MIDI*** and get you connected so you can start learning about what the ***MIDI*** is all about.

Basic Concepts

Let's start with what the ***MIDI*** is and how it works in **Flex** mode with the Flex radio.

CTR2-MIDI Controller

CTR2-MIDI (***MIDI***), when operating in **Flex** mode, can be used a standalone radio controller that has been optimized for controlling a Flex 6000/8000 series radio. It can also work with AetherSDR's **CTR2 Relay** feature to provide a physical knob and buttons for that app. ***MIDI*** provides a tuning knob, six dual-function buttons, and a dual-mode 3.5mm stereo jack that supports paddle or straight key/PTT inputs. It can connect directly to the Flex radio over WiFi using the Flex API or through AetherSDR's **CTR2 Relay** which also uses the Flex API. Since the ***MIDI*** doesn't have a built-in display (like the M5Dial based CTR2 devices and CTR2-Max), the operator must memorize the knob and button modes or connect ***MIDI*** to a dumb terminal to see it's control functions. There are worksheets in Appendix F that you can use as cheat-sheets.

Flex 6000/8000/Aurora Series Radios

Flex radios are basically computers with an RF interface. The computer-side has a built-in IP server that supports multiple IP client connections. Information about the RF interface is shared with all connected clients and each client can control the RF interface. Some clients, like SmartSDR, AetherSDR, and RimuSDR are classified as "GUI" clients (Graphical User Interface). The radio only supports one GUI client at a time and this client is the primary control interface. Other clients, such as the ***MIDI***, connect as "non-GUI" clients. Non-GUI clients are "bound" to a GUI client, and basically work in parallel with the GUI client.

3rd-Party App

Since the ***MIDI*** doesn't have a display, if your radio isn't an "M model, you'll need to run a radio control app (GUI client) such as SmartSDR to see the control changes you make with the ***MIDI***. When running in WiFi mode, the ***MIDI*** will work with any GUI client. You can also connect it to a computer running AetherSDR and use its **CTR2 Relay** to connect to your radio using USB.

Initial Startup

The first order of business is to power up your ***MIDI***. This is generally done by plugging the ***MIDI*** into a USB port on your computer. It can also be powered by a 5-volt phone charger or USB battery. Do not plug it into an unpowered USB hub. Unpowered hubs cannot provide the power the ***MIDI*** requires when it's using the Bluetooth or WiFi radio.

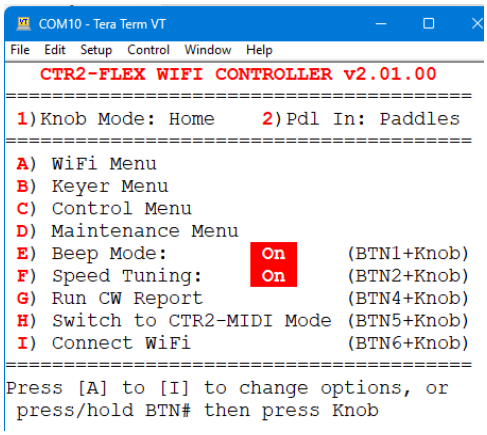
> **NOTE:** When using ***MIDI*** with AetherSDR's **CTR2 Relay**, it must be plugged into a USB on the computer running AetherSDR.

When the ***MIDI*** boots in **Flex** mode, it will flash all of the LEDs, then sound an 'F' in Morse to indicate that is in **Flex** mode. A Windows computer will also "ding" when it registers the ***MIDI***. If the ***MIDI*** sounds an 'M1' or 'M2', it's telling you that it booted into MIDI mode. You'll want to switch it to **Flex** mode for this guide. To do this, press-and-hold BTN5, then press-and-release the knob switch, followed by releasing BTN5. The unit will sound 'F?' in Morse. Press the knob switch to accept the mode change, or any button to cancel.

There are two ways to connect ***MIDI*** **Flex** mode to your radio. The first is through WiFi. The second is through AetherSDR's **CTR2 Relay**. We'll cover WiFi first.

Configure WiFi and Radio Address

To use WiFi to connect to your radio, enter your WiFi credentials and your radio's IP address into the unit. This must be done using a dumb terminal such as Putty or Tera Term.



Once connected to the terminal, press any key to start ***MIDI's*** terminal server. You will be presented with the menu on the right.

Press the **A** key to open the WiFi Menu. Enter your router's SSID and password, and your radio's IP address in this menu. You shouldn't need to change the radio's IP port # unless you've changed it. 4992 is the default.

On the **WiFi menu** you can also choose the slice to connect to, enable WiFi auto-connect, and connect WiFi. Auto-connect is handy if you want the ***MIDI*** to automatically connect to your radio when you power it up.

Connecting to WiFi

With the terminal connected, press **I** to connect. When the terminal is not connected, press-and-hold **BTN6**, then press-and-release the knob switch, then release **BTN6** to connect. The ***MIDI*** will sound 'Sx?' in Morse, where 'x' = the slice to connect to (i.e. 'SA?' for 'connect to Slice A?'). Press-and-release **BTN6** to step through the slices (A to H) or press-and-release the knob switch to connect. Press any other button to cancel.

NOTE: To successfully connect to your radio, your WiFi credentials must be correct (they are case-sensitive), your radio's IP address and port # must be correct, and your radio must be powered up and on the same physical network (LAN) as ***MIDI***.

WARNING: Before you connect the ***MIDI*** to your radio, make sure the GUI client is running and connected first. If it's not running, the ***MIDI****'s* connection will block the GUI client from connecting if you're running SmartSDR v2.xx. Likewise, it will block the connection if you're running SmartSDR v3.xx or v4.xx with MultiFlex disabled. If v3.xx or v4.xx has MultiFlex is enabled, the ***MIDI*** will not be bound to the same slice as the GUI client.

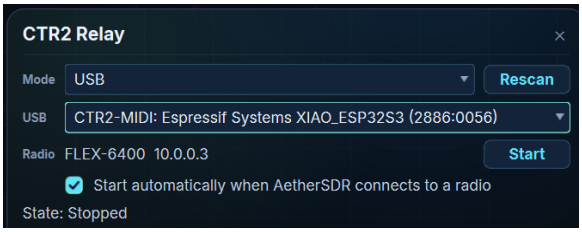
Configure AetherSDR's CTR2 Relay Mode

NOTE: CTR2 Relay is not available when CTR2-MIDI firmware is run on CTR2-Micro hardware.

AetherSDR's CTR2 Relay mode is a USB-to-network bridge that connects the ***MIDI*** directly to the radio that AetherSDR is controlling using USB. The radio can be either local or remote from AetherSDR. No router settings or radio IP address required. The only requirement is that ***MIDI*** must be connected to a USB port on the computer running AetherSDR. The drawing below shows how this works.



To initiate a connection, select **Settings -> Controllers -> CTR2…** to open the **CTR2 Relay** panel in AetherSDR. If **CTR2-MIDI** isn't shown in the **USB** list box, click **Rescan** and select it from the list of devices in the list box.



Click the **Start automatically when AetherSDR connects to a radio** checkbox if you want AetherSDR to automatically connect to ***MIDI*** when it starts.

Click the **Start** button to connect to ***MIDI***. Once the connection sequence finishes, ***MIDI*** will sound "C" followed by the slice it's connected to in Morse. ***MIDI*** can now control the radio as if it were connected with WiFi.

Change Knob and Input Jack Mode

On the terminal's **Home** menu, press the **1** key to step through the knob modes. This can also be done by simply pressing-and-releasing the knob. There are four knob modes, Home, 1, 2, and 3. The yellow LEDs on the unit indicate the selected knob mode. Each knob mode has a turn and a push-and-turn control function.

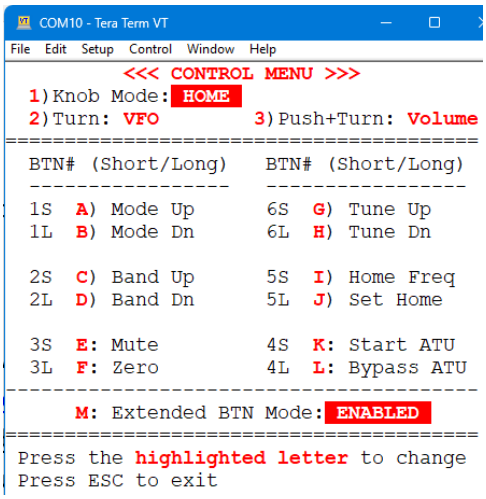
Press the **2** key on the terminal to toggle the input jack on the ***MIDI*** between **Paddles** and **Key/PTT**. This can also be done by long-pressing the knob switch when in the **Home** knob mode (all yellow LEDs are OFF). When **Key/PTT** mode is active the green LED on the ***MIDI*** will be ON.

Configure the Keyer

The ***MIDI*** supplies the CW keyer and sidetone oscillator in **Flex** mode. You can configure the keyer's settings and CW macros in the **Keyer Menu.**

Configure the *MIDI's* Controls

The ***MIDI*** ships with the knob and button controls pre-defined and **Extended BTN Mode** *enabled*. Settings can be edited in the **Control Menu** (press **C** on the terminal while in the **Home** menu).



Each knob mode has a turn and a push-and-turn function for the knob. Press the knob switch to step through each **Knob Mode**. The four knob modes are indicated by the yellow LEDs on the unit.

You can select the **Turn** and **Push+Turn** functions for the knob by pressing keys **2** or **3** on the terminal. Once selected, turn the knob or use the arrow keys on the terminal to select a new function.

In **Extended BTN Mode**, button functions change based on the selected knob mode. These can be customized in the Button settings section in the **Control** menu. Press 1 on the terminal to step through the knob modes, then press **A** through **L** to select a button (#S=short-press, #L=long-press) within the selected knob mode. Once selected, turn the knob or use the terminal's arrow keys to select a new function.

If you *disable* **Extended BTN Mode**, the buttons will always follow the button assignments in the **Home** knob mode.

This completes the basic setup of the ***MIDI's*** **Flex** mode.

Please explore this manual for in-depth descriptions of the features in the ***MIDI*** and feel free to change the knob and button settings to fit your operational requirements.

Keywords and Definitions

As many are new to the concept of controlling radios with an external controller, it will help to define many of the keywords used in this document.

Bluetooth	A wireless technology used to connect two devices together
Bluetooth MIDI	A method that sends MIDI commands to an app running on another computer using Bluetooth
BLE	Bluetooth Low Power – An advanced type of Bluetooth communication that uses less power than conventional Bluetooth
CTR2	A product line from Lynovation.com that specializes in control devices for amateur radio
CTR2-MIDI	A miniature radio controller based on the ESP32-S3 microcontroller. It uses Bluetooth or USB MIDI to connect 3rd party radio control apps that support MIDI control of their functions
Flex Mode	A mode available in **CTR2-MIDI** firmware that uses WiFi and the Flex API, instead of Bluetooth or USB MIDI, to connect directly to, and control a Flex radio without a 3rd party radio control app
MIDI Mode	This is the original CTR2-MIDI firmware that uses *Bluetooth* or *USB MIDI* to control a 3rd party radio control app
Extended BTN Mode	When *enabled*, the 12 button functions on the unit change with each *Knob Mode*. When *disabled*, the 12 button functions in the Knob Home mode are used in all knob modes. Both yellow LEDs flash when changing knob modes when *enabled*.
Paddle Input Jack Mode	The Paddle Input jack has two operating modes: **Paddle**: The Paddle Input jack connects your paddles to the internal keyer **Key/PTT**: A straight key (or external keyer) and/or external PTT switch can be connected to the Paddle Input jack to control CW carrier and MOX directly
Knob Mode	Refers to one of four modes available on the knob control – each mode has a *Push* and a *Push & Turn* function – Knob modes are indicated by the yellow LEDs
LAN	Local Area Network – generally a wired network contained within small, localized area such as a home, office, or your radio shack – It usually incorporates WiFi to connect wireless devices to the network
MIDI	A protocol originally designed to control musical instruments - It is simple, small, and fast
MIDI Button Command	A MIDI command that sends an On or Off command to control a parameter in a radio control app – also referred to as a MIDI Note command since it was originally used to send musical note command to instruments
MIDI Control Command	A MIDI command that allows sending a value to the radio control app to change a variable parameter such as tuning, volume, etc.
Radio Control App	A 3rd party program or application running on a computer, tablet, or phone that is used to control a radio remotely
Serial Port	A connection that sends data to, or receives data from, a remote device serially – the serial communications can travel over a physical wire (RS-232) or be embedded in a virtual connection using *USB*
Terminal	A program such as Tera Term or Putty running on a PC that connects to a device using a serial port allowing you to interact with that device

Terminal Server	An application running on a device that allow it respond to key strokes sent from a *terminal* – generally used to create a basic user interface to control or configure a remote device
USB	Universal Serial Bus – a common interface available on all computers that enables communications between the computer and external devices
USB MIDI	A method used to send MIDI commands to a radio control app using *USB*
WiFi	A wireless technology that allows remote devices to connect to a *LAN* network
XModem	A file transfer protocol that is used to import and export settings files

Hardware



The ***MIDI*** hardware uses a 60mm x 60mm x 20mm enclosure. This form factor is the perfect size for a busy operating desk or for portable operation.

As shown in the photo, the knob is the predominant feature on the face of the unit. When oriented with the USB-C connector and paddle jack as shown the two LEDs in the upper left corner show the knob mode with the lower LED having a value of 1 and the upper LED having a value of 2. Both LEDs will be off when the knob is in its **Home** mode. Short-pressing the knob steps you through **Knob modes 1**, **2**, **3**, and back to **Home**. In the photo above the knob is in **mode 1**.

The green LED flashes once every 5 seconds if WiFi or **CTR2 Relay** is not connected. It rapidly flashes twice each second when the unit is connected to WiFi or **CTR2 Relay** and the radio. It is on continuously (and flashes off) when Paddle Input Jack mode is set to **Key/PTT.**

The dual-function buttons (BTNs) are laid out around the circumference of the knob. **BTN1** is on the upper left, just below the yellow knob mode LEDs. The buttons are numbered counter-clockwise around the knob ending with **BTN6** on the upper right. Short and long-press operation allow you to control two **BTN** functions with each button. In Extended BTN mode, button functions follow the selected knob mode.

You'll find the USB-C and 3.5mm (1/8") stereo paddle input jack on the top edge in the photo.

CAUTION: The 3.5mm jack is for paddle, key, and PTT inputs only. It is not designed for output. Do not connect this jack to your radio as this may damage the unit.

Operating Modes

When operating in **MIDI** mode, the ***MIDI*** uses Bluetooth or USB MIDI to control MIDI enabled 3rd party radio control software such as SmartSDR for iOS/MacOS, AetherSDR and RimuSDR for Windows/Mac/Linux, or Thetis and SDR-Console for Windows. SmartSDR for iOS/MacOS, AetherSDR,

and RimuSDR support SmartLink so this is the mode you'll want to use to operate your radio when using the ***MIDI*** remotely.

When operating in **Flex** mode, the ***MIDI*** hardware connects directly to the network server in your Flex 6xxx/8xxx/Aurora radio using your station's WiFi network. This allows the ***MIDI*** to control your radio directly while using any of the Flex GUI client apps available (SmartSDR for Windows, iOS, and MacOS, RimuSDR, AetherSDR, SDRoxide, etc.) MIDI and HID control (for RHR, RemoteTx, or OnlineRemoteRadio) and SmartLink operation are not available in this mode. This mode is referred to as **Flex** mode in this manual.

When running AetherSDR, ***MIDI*** can also connect to your radio through **CTR2 Relay**, using USB while in **Flex** mode.

CTR2-MIDI MIDI control mode is covered in a separate manual. It can be downloaded here.

Switching Modes

To switch between **MIDI** and **Flex** modes, press-and-hold **BTN5** (middle-right button), then press-and-release the knob switch. The unit send either ***F?*** (if switching to **Flex** mode) or ***M?*** (if switching to **MIDI** mode) in Morse. Press the knob to accept or any other button to exit. If accepted, the unit will power down and restart in the new mode and sound either a Morse ***M*** *(plus map #)* or ***F*** to indicate the new mode.

Design Philosophy

Flex mode was ported from **CTR2-Flex** firmware. However, unlike **CTR2-Flex** hardware (which has a display and allows you to control your radio without SmartSDR), you must run a GUI client with the ***MIDI***, regardless of whether it's in **MIDI** mode or **WiFi** mode. My goal with the **Flex** implementation was to provide a simple physical interface for Flex radios that works with every Flex radio control app (including SmartSDR for Windows), doesn't require a separate display to control it, and runs on hardware you may already own.

The GUI client is used to view most of the changes you make in **Flex** mode. The ***MIDI*** has a built-in terminal server that you can access with a terminal program for advanced configuration. The terminal server allows you to use a terminal while you learn the knob and button settings. Once you've developed the required "muscle memory" for each knob mode and button function you'll be able to operate the unit without a terminal. You can print the default button setting page for reference while learning the control functions. A blank button configuration sheet is provided in Appendix G if you change the button configurations to fit you operating preferences.

> **NOTE:** The terminal server is disabled when ***MIDI*** is connected to **CTR2 Relay**.

Morse Prompts

Since the ***MIDI*** doesn't have a display, **Flex** mode uses Morse prompts to inform the operator of various operating changes that aren't visible in the GUI client or on the unit's LEDs, like **Set Home Freq** and **VFO Lock**. The speed at which these prompts are sent can be adjusted with a terminal in the **Keyer Menu -> Report Speed** menu item. A **CW report** is available that gives you the basic operating parameters of the unit. To start the report, press BTN4+Knob. You can stop the report at any time by pressing any of the buttons on the unit.

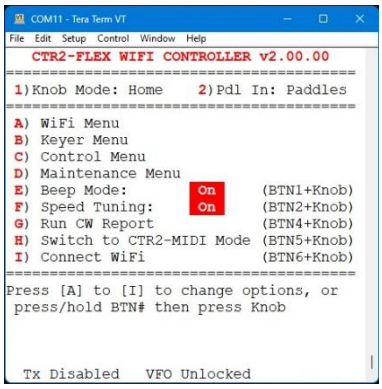
Boot Sequence

When your ***MIDI*** boots into **Flex** mode, an ***F*** will sound in Morse. This will be your only indication that your unit is in **Flex** mode. You can check the current mode if you have a terminal connected to your ***MIDI***, or press **BTN5+Knob** to check the mode. The ***MIDI*** sends ***M?*** if it's in **Flex** mode, or ***F?*** if it's in **MIDI** mode.

With a terminal connected, press any key to start the terminal server and display the **Flex Home Menu.**

See Switching Modes for information about switching between **Flex** and **MIDI** modes.

Configuring WiFi and Radio IP Settings

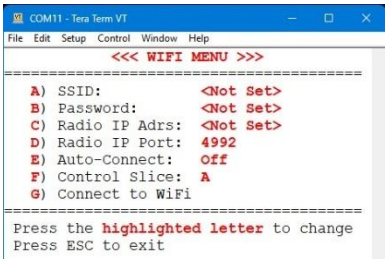


If you plan on using **Flex** mode with your WiFi network, you must first configure the WiFi credentials and radio IP address. This must be done with a terminal program. Several are available such as Tera Term and Putty for Windows. If you're a Mac or Linux user you can use the built-in Terminal program.

No WiFi configuration is required if you will be connecting to your radio using AetherSDR's **CTR2 Relay** feature. You can skip this section.

Once the ***MIDI*** boots, press any key on the terminal to open the **Home** menu shown here. From this menu, you have several options. Let's start with configuring the WiFi and radio IP settings.

Press **A** to open the **WiFi Menu**.



In the **WiFi Menu**, press **A** to enter the router's **SSID**. The SSID is case sensitive.

Next, press **B** to enter the routers' **Password**. The password will show as "********" on this menu when it has been set. The password is also case sensitive.

Finally, press **C** to enter your radio's IP address. This can be found in SmartSDR.

If you've changed the radio's IP port #, press **D** to change from the default value of 4992 to the IP port # you've set in your radio.

Set the **Auto-Connect** option (**E**) to *On* if you want your ***MIDI*** to automatically connect to your WiFi network and radio when it boots.

> **WARNING:** You should not enable **Auto-Connect** until you have verified that the unit can connect to your WiFi network and radio using the credentials entered above. If the unit fails to connect during boot, it will keep trying. To bypass **Auto-Connect**, press-and-hold **BTN1** during the boot process.
>
> **NOTE:** Because the ***MIDI*** connects to your radio as a *non-GUI client* (attached to a GUI client like SmartSDR) you must connect the ***MIDI*** after you start SmartSDR.

Press **F** to change the slice the ***MIDI*** will control when it connects to the radio. Select any slice your radio supports, from **A** to **H**.

Press **G** to connect to your WiFi network and then to your radio. You can also press-and-hold **BTN6** (top-right button) then press-and-release the knob switch to connect or disconnect WiFi. See the **BTN6+Knob** section for a complete description of the WiFi connect/disconnect process.

Other menus are described in the CTR2-**Flex** Terminal Menus section.

Flex Mode Options

Flex mode includes many options. Similar to the **MIDI** mode, it supports 4 knob modes. Each knob mode has a *Turn* and a *Push & Turn* function, for a total of 8 knob functions. Unlike **MIDI** mode, the knob and button functions in **Flex** mode can be set using a terminal program. (Setting knob and button functions for **MIDI** mode must be done in the 3rd party radio control app's MIDI map).

You can select from 29 functions for the 8 knob functions (VFO, volume, RIT, etc.) and from 63 button functions for the dual-function buttons (Mode Up/Down, Band Up/Down, Modes, Bands, CW Macros, etc.).

Once you have your unit configured for your operating style you should back up your settings by exporting them to a file on your computer. This allows you to import the settings back in if needed.

Extended BTN Mode

By default, **Extended BTN mode** is ***enabled***. In this mode, 12 button functions can be defined for each knob mode, giving you a total of 48 button functions. Knob and button functions are set in the Control Menu when a terminal is connected to the unit.

To make button functions easier to remember, I've grouped the buttons assigned to the four knob modes into groups. **Knob Home** mode has general button controls (mode, band, frequency step, etc.) **Knob 1** mode has mode control buttons, **Knob 2** mode has band control buttons, and **Knob 3** mode has CW macros.

If different button functions for each knob mode are too overwhelming, you can ***disable*** the extended buttons in the Control Menu so that only the 12 button functions on the **Home** knob mode are available regardless of the knob mode.

Paddle Input Jack Mode

The **Paddle Input Jack** has two modes, **Paddle** and **Key/PTT**. In **Paddle** mode (default), this jack is used to connect your paddles to the internal keyer. The keyer controls the CW carrier of your radio over the WiFi network. The ***MIDI*** provides a sidetone for the internal keyer.

To use a straight key (or external keyer) and/or an external PTT switch, switch to **Key/PTT** mode by long-pressing the knob while it is in the **Home** mode (all yellow LEDs off). In **Key/PTT** mode you can connect a straight-key to the left paddle input (TIP/Shield) and control the CW carrier with your straight key or an external keyer. You can also connect a remote PTT switch to the right paddle input (RING/Shield) to control the PTT on the radio.

NOTE: There are three PTT modes. **Off**, **Momentary**, and **Latching**. The PTT mode is controlled with BTN4 when in the paddle input jack is in **Key/PTT** mode.

The green LED indicates the **Paddle Jack Input** mode**.** It is *Off* in **Paddle** mode and *On* in **Key/PTT** mode.

CAUTION: The 3.5mm jack is for paddle, key, and PTT inputs only. It is not designed for output. Do not connect this jack to your radio as this may damage the unit.

CW Macros

There are 9 CW macros available. Information on editing macros and using special prosigns can be found in the **Edit CW Macros and Call Sign** section.

Keyer type, sidetone pitch, CW speed, CW Report speed, and the paddle wiring configuration can also be edited in the **Keyer Menu**.

BTN#+Knob Options

Like **MIDI** mode, **Flex** mode supports changing some operating modes by pressing and holding one of the buttons then pressing and releasing the knob switch.

Beep Mode (BTN1+Knob)

You can control when the ***MIDI*** beeps. Normally, it beeps whenever you press a button or the knob. This lets you know that you've changed something on the ***MIDI***. You may prefer the ***MIDI*** remains quiet all the time or only beeps when you long-press a button or the knob.

> **NOTE:** Turning **Beep mode** *Off* does not affect the keyer sidetone. You can adjust the sidetone and beep volume with a volume control if it has been installed on your unit. You can also turn sidetone off by setting the *Keyer Pitch* to 0 Hz in the **Keyer** menu. If you have an early version of ***MIDI*** hardware, you can add a volume control pot. See Appendix H for instructions on doing this.

To change the **Beep mode** press-and-hold **BTN1** then press-and-release the knob. The new **Beep mode** will be announced in Morse code as follows:

- "B0" = **Beep** mode is off. All LEDs will briefly flash when you long-press a button or the knob.
- "B1" = **Beep** mode is normal. The ***MIDI*** will beep with every button or knob press. A second beep will sound when you long-press a button or the knob.
- "B2" = **Beep** mode is long-press. The ***MIDI*** will beep only when you long-press a button or the knob.

Speed Tuning Mode (BTN2+Knob)

Speed (proportional) tuning is supported when **VFO** is selected for the knob control. You can select from **On**, **Fast**, or turn speed tuning **Off**.

To change the Speed mode press-and-hold **BTN2** then press-and-release the knob. The new speed mode will be announced in Morse code as follows:

- ***S0*** = speed tuning is off
- ***S1*** = speed tuning is set to normal speed
- ***S2*** = speed tuning is set to fast speed

Future (BTN3+Knob)

This function is reserved for future use.

Run CW Report (BTN4+Knob* when Paddle Input = Paddles)

When the Paddle Input Jack mode is set to **Paddle** (green LED is Off), press-and-hold **BTN4** (bottom-right button) then press-and-release the knob switch to start the **CW Report**.

The report sends ***F= WIFI OFF*** if WiFi is not connected. ***F*** indicates the unit is in **Flex** mode.

When the unit is connected to WiFi and the radio, the **CW Report** sends

F=

S{active slice}= Sends the selected slice A, B, C, D, E, F, G, or H

{Freq and Mode} Freq is in Hz – if last three digits are 0, they are not sent
PTT {0, M, or L}= 0= PTT mode is off, M= Momentary mode, L= Latch mode

NOTE: To speed up the report, frequency digits that are **0** are sent as a "long-T" and the last three digits of the frequency are not sent if they = **000** (Example: 7142000 would be sent as 7142).

NOTE: You can change the speed of the CW sending the report in the **Keyer Menu** or by turning the knob *while the report is running.*

***** When ***MIDI*** firmware is running on *Micro* hardware, *long-press* **BTN1** then press the knob.

PTT Mode (BTN4+Knob* when Paddle Input = Key/PTT)

When Paddle Input Jack mode is in **Key/PTT** mode (green LED is *On*), you can select from three **PTT modes**:

- **Off:** PTT is *Off* – Sends ***PTT O*** in Morse
- **Momentary:** PTT follows the PTT switch contacts – Sends ***PTT M*** in Morse
- **Latching:** PTT keys up on the first contact and drops off on the second contact – Sends ***PTT L*** in Morse

To change the **PTT mode** press-and-hold **BTN4** then press-and-release the knob switch.

When **Momentary** PTT is enabled, the unit sounds a short (10 mS) beep when PTT goes active or inactive. In **Latching** mode, it sounds two short beeps when PTT goes active. This can be deactivated by turning **Beep Mode** *Off.*

NOTE: PTT is only active when the ***MIDI*** is connected to your radio

***** When ***MIDI*** firmware is running on *Micro* hardware, *long-press* **BTN1** then press the knob.

Switch Modes (BTN5+Knob*)

To switch between **Flex** and **MIDI** modes, press-and-hold **BTN5** (middle-right button) then press-and-release the knob switch. If the unit is in **Flex** mode, it sends ***M?*** in Morse (do you want to switch to **MIDI** mode?). If it's in **MIDI** mode, it sends ***F?*** (do you want to switch to **Flex** mode?).

To accept the change, press-and-release the knob. The unit will send ***AS*** (wait in Morse) then power down and reboot into the new mode. When it boots into **MIDI** mode, a Morse ***M*** (plus the current map #) will be sounded. When it boots into **Flex** mode, a Morse ***F*** will be sounded. If you have a terminal connected to the unit, press any key to open the **Home** menu.
***** When ***MIDI*** firmware is running on *Micro* hardware, *long-press* **BTN2** then press the knob.

Connect/Disconnect WiFi (BTN6+Knob*)

To toggle the WiFi connection state, press-and-hold **BTN6** (top-right button) then press-and-release the knob switch.

***** When ***MIDI*** firmware is running in **Flex Mode** on *Micro* hardware, *long-press* **BTN3** then press the knob.

Connecting to WiFi

When connecting to WiFi, the unit will send ***S*** (for Slice) plus the currently selected slice followed by a "?" in Morse… ***SA?*** for slice A, ***SB?*** for slice B, and so on. To connect to the indicated slice, press-and-release the knob switch. To change the slice, press-and-release **BTN6** to step through each slice (**A** to **H**). The new slice will be sent in Morse after each press. When you get to the slice you want to connect to, press-and-release the knob switch. If you don't press the knob switch within 5 seconds, the connect process will abort and a Morse error (8-DITS) will be sent.

When the connection process starts, "wait" (***AS***) will be sent in Morse. Once the connection process is complete, ***C*** plus the slice (**A**, **B**, **C**, etc.) will be sent in Morse. When connected to WiFi, or through AetherSDR's **CTR2 Relay**, the green LED will flash twice rapidly once a second. If there's a problem connecting, a Morse error (8-DITS) will be sent. You'll need to connect a terminal program to the unit to find out why the connection is failing.

> **NOTE:** WiFi credentials and the radio's IP address and IP port must be entered correctly, and the radio must be online before the unit will connect to your WiFi network.
>
> **NOTE:** If ***MIDI*** is connected to WiFi and you start **CTR2 Relay**, WiFi will automatically disconnect.

Disconnecting from WiFi

When you are connected to WiFi, pressing **BTN6+Knob** will disconnect WiFi. A ***D*** will be sent in Morse to indicate a successful disconnection. When WiFi is off the green LED will flash once every 5 seconds.

Flex Terminal Menus

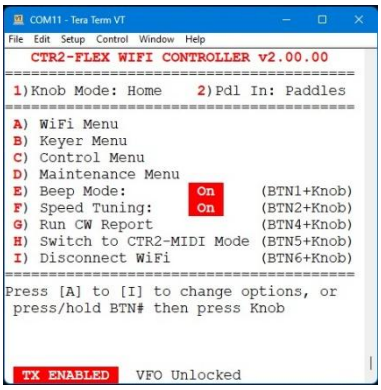
Configuration must be done with a terminal program such as Tera Term or Putty connected to the ***MIDI***. Once configured and you are familiar with the knob and button settings you can run the ***MIDI*** in standalone mode using the GUI client to show when changes are made with the ***MIDI***.

Information on connecting various terminals can be found in Appendix C, D, and E. I recommend using Tera Term because it supports the XModem file transfer protocol that is used to import and export settings files.

The following sections will describe the options in each menu.

Home Menu

When you first connect to the unit you must “wake up” its terminal server by pressing any key. The **Home** menu will appear as shown here.



> **NOTE:** The terminal server is disabled when ***MIDI*** is connected to **CTR2 Relay.**

To select an item, press the highlighted key.

At the bottom of the **Home** menu, you’ll find **Tx Enable** and **VFO Lock** status. By default, **TX** is **enabled** when you connect to a slice on the radio and **VFO Lock** is disabled. These settings are assigned to buttons.

Press **1** or press-and-release the knob switch to step through the four knob modes.

Press **2** or *long-press* the knob switch while in **Home** knob mode to toggle the Paddle Input Jack mode between **Paddles** and **Key/PTT**.

Several of the menu functions can be controlled by pressing and holding the appropriate button then pressing the knob’s switch. This allows you to control these functions without the terminal being connected. These functions have **BTN#+Knob** displayed on the right of the menu option.

Press **A** to open the WiFi Menu where you can configure your WiFi credentials and your radio’s IP address and port #

Press **B** to open the Keyer Menu where you can adjust the parameters of the internal keyer.

Press **C** to open the Control Menu where you can review and change the knob and button function assignments. You can also ***enable*** or ***disable*** the Extended BTN mode here.

Press **D** to open the Maintenance Menu where you can import and export settings files, recalibrate the BTNs, and **Reset** the unit to **Factory Settings**.

Press **E** to select the button beep mode. Options are **Off**, **On**, and **Long-Press Only**.
Shortcut: BTN1+Knob

Press **F** to select the VFO Speed Tuning mode. Options are **Off**, **On**, and **Fast**.
Shortcut: BTN2+Knob

The **G** key selects from two options, depending on the Paddle Input Jack mode.
Shortcut: BTN4+Knob

- When set for ***Paddles*** (green LED Off), press **G** to run the CW report.
- When set for **Key/PTT** (green LED On), press **G** to switch between **Off**, **Momentary**, and **Latching PTT** control.

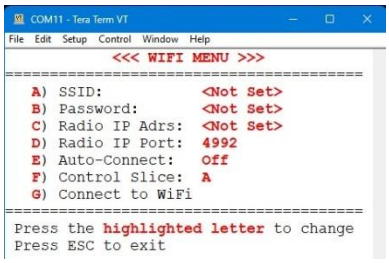
NOTE: To use **Key/PTT** mode, connect your straight key or external keyer to the TIP/SHIELD and your external PTT switch to the RING/SHIELD on the Paddle Input jack.

Press **H** to power down the unit and reboot into **MIDI** mode. The unit will sound ***M*** plus the current map # in Morse after it reboots to indicate that it is in **MIDI** mode.
Shortcut: BTN5+Knob

Press **I** to connect or disconnect from the WiFi network. You must be connected to WiFi or to AetherSDR's **CTR2 Relay** in order to control the radio. This function is also available in the WiFi Menu.
Shortcut: BTN6+Knob

WiFi Menu

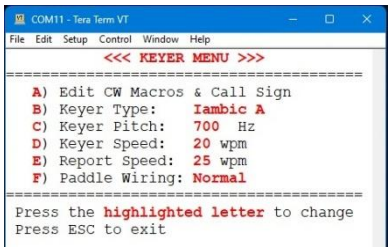
The **WiFi Menu** is accessed by pressing **A** on the **Home** page. The WiFi and Radio settings on this menu have already been described in the Configuring WiFi and Radio IP Settings section above.



> **NOTE:** If you use **CTR2 Relay**, there is no need to configure WiFi.

Keyer Menu

The **Keyer Menu** allows you to change various parameters on the internal keyer.



Press **A** to open the Edit CW Macros & Call Sign Menu. This menu is described in the next section.

Press **B** to step through the available keyer types. Choose from **Off**, **Iambic A**, **Iambic B**, **Ultimatic**, and **Bug**.

Press **C** to set the keyer's sidetone **Pitch** in 50 Hz steps. This also sets the radio's sidetone pitch if the radio is online. There's a **Pitch** knob control you can assign to a knob function if you want to set the frequency in 1 Hz steps.

Press **D** to adjust keyer's speed. This also changes the radio's keyer's speed if it's online. By default, there's a **Speed** control mapped to the *Turn* function of knob mode **3** so you can set this while sending CW.

Press **E** to adjust the CW Report code speed. Once you learn the code prompts you can increase this value to higher than you would normally copy because you'll know what to listen for as you change parameters.

Press **F** to swap the wiring on the Paddle Input Jack between **normal** and **reverse**.

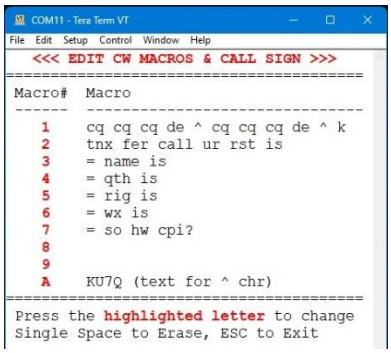
NOTE: The radio's keyer is disabled when a CTR2 keyer is selected. To release this override so you can use your radio's keyer, set the keyer **Type** to **OFF** in this menu.

Edit CW Macros and Call Sign Menu

Pressing the **A** key on the **Keyer** menu opens the **Edit CW Macros and Call Sign** menu. Here you can edit the 9 CW macros available for the internal keyer. On the terminal, press the highlighted **#** of the macro you want to edit.

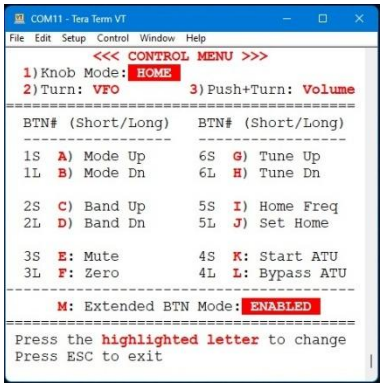
Press the **A** key to enter the text (generally your call), that you would like to assign to the **^** character. When this character is used in a macro it will be replaced with the text you enter.

You can also insert the "*" character in a macro. This character increases the key speed 50% until either another "*" is entered in the macro, or the macro ends.



Control Menu

The **Control** menu allows you to assign functions to the four knob modes and to each of the buttons on the ***MIDI***. Each knob mode has a *Turn* and a *Push and Turn* setting. Buttons have short-press and long-press functions.



A button *short-press* is a press-and-release of less than 1 second. Release the button after you hear the first beep to execute the *short-press* function.

A button *long-press* is a press-and-release of more than 1 second and less than 4 seconds. You will hear one beep when you first press a button. Once the *long-press* threshold is reached you'll hear another, higher frequency beep. If you release the button after the second beep the *long-press* button action will be executed.

If you hold a button down longer than 4 seconds, you'll hear a low frequency beep and the button will be ignored.

Knob Mode Settings

The top section allows you select a knob mode. This changes the *Turn* and *Press+Turn* functions of the knob and the button functions if Extended BTN Mode is enabled.

Press **1** to change the knob mode.

Press **2** to select the *Turn* function of the knob for this knob mode. The current value will be displayed in inverted text. Use the cursor control keys on the terminal, or turn the knob on the unit, to scroll through the available knob functions.

Press **3** to select the *Push+Turn* function of the knob for this knob mode. The current value will be displayed in inverted text. Use the cursor control keys on the terminal, or turn the knob on the unit, to scroll through the available knob functions.

Available Knob Functions

The table below lists the available functions that can be assigned to each knob mode *Turn* or *Push+Turn* function. Select **2** or **3** then turn the knob to select a function.

VFO Freq	Filter Hi-Cutoff	APF/ANF Level	Mic Gain	Panadapter Max
Volume	Filter Lo-Cutoff	RIT Freq	Line-Out Gain	Panadapter Min
Squelch	Filter Offset	XIT Freq	Headphone Gain	Panadapter Avrg
AGC-T	WNB Level	Power Out	Keyer Speed	Panadapter Gain
Monitor	NB Level	Tune Power	Key Hold Delay	Panadapter Black
Balance	NR Level	VOX Level	Panadapter Zoom	

Button Settings

The next section displays the button assignments for the current knob mode if Extended BTN Mode is ***enabled***. If it is ***disabled***, the button assignments for the **Home** knob mode will be used for all knob modes.

The buttons on the menu are arranged in the same order as they appear on the ***MIDI****'s* hardware, with buttons 1, 2, and 3 top-to-bottom on the left and buttons 6, 5, and 4 top-to-bottom on the right. Button numbers followed by "S" are short-press functions and numbers followed by "L" are long-press functions.

Press the highlighted key (**A** through **L**) to select a button to edit. Once selected, its function will be displayed in inverted text. Use the left & right cursor control keys, or the knob, to change the selected button function. Press another highlighted key, press the up & down cursor control keys, or press the Tab key to move to another button. Changes are automatically saved. When you're done, press **Esc** or **Enter** to exit this menu.

Available Button Functions

The following table lists the button functions available. They can be assigned to any button short-press or long-press function.

Mode Up	RIT On/Off	Tx Ant	USB	160m	10m	Set Home Freq	CW Macro9
Mode Down	XIT On/Off	Start ATU	AM	80m	6m	CW Macro1	------ (No action)
Band Up	WNB On/Off	Bypass ATU	SAM	60m	General	CW Macro2	
Band Down	NB On/Off	Tx Proc On/Off	RTTY	40m	2m	CW Macro3	
Tune Step Up	NR On/Off	Tx Enable On/Off	FM	30m	1.25m	CW Macro4	

Tune Step Down	APF/ANF On/Off	Zero Digits Below TStep	NFM	20m	70cm	CW Macro5	
Breakin	RF Gain	VFO Lock On/Off	DFM	17m	33cm	CW Macro6	
Mute	AGC	CW	DIGL	15m	23cm	CW Macro7	
Bandwidth	Rx Ant	LSB	DIGU	12m	Home Freq Recall	CW Macro8	

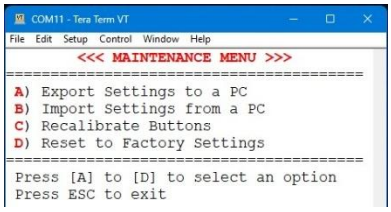
Extended BTN Mode

By default, **Extended BTN mode** is ***enabled*** and the button assignments follow the knob mode. Each mode supports 12 button functions, giving you a total of 48 button functions. You can disable this feature and have the **Home** buttons applied to all knob modes by pressing the **M** key to toggle **Extended BTN Mode** *On* and *Off*.

> **NOTE:** When this mode is ***enabled***, both yellow LEDs flash twice when changing the knob mode.

Maintenance Menu

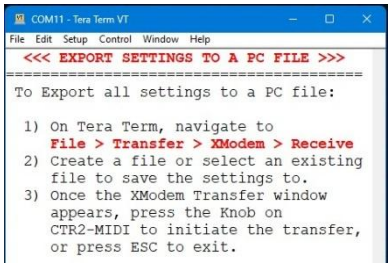
The **Maintenance** menu is where you can backup and restore the settings on your unit, recalibrate the BTNs, and reset your unit to factory settings.



NOTE: You need to use a terminal program that supports XModem such as Tera Term to **Export** and **Import** settings files. It will be used here to explain the process.

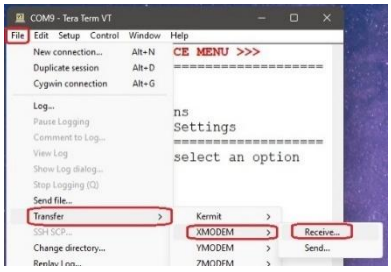
Export Settings

Press **A** to export the settings on your unit to a file on your computer. The terminal will display instructions on how to export your settings.



NOTE: This screen will time-out after 60 seconds if a transfer hasn't been started. You can also press any BTN on the unit to exit this screen.

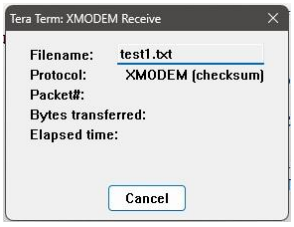
Next, setup Tera Term to receive your data. Select **File->Transfer->XModem->Receive…** and enter the name of the settings file you wish to create. If you enter an existing file name, you'll be asked if you want to overwrite it.



I'll create a file named *test1.txt* for this example. You can use any valid file name and extension for the file name.

NOTE: I recommend using the .txt extension since the file to be created is a text file and can be edited with a text editor.

Once you've entered the file name, the **XModem Receive** dialog box will appear. At this point, Tera Term is waiting for the unit to send its settings.



>>> Press-and-release the knob on the *MIDI* to start the transfer.

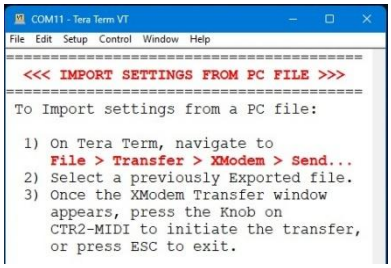
Once the transfer completes, you'll be returned to the **Maintenance** menu.

NOTE: Your WiFi and Radio IP settings are encrypted in the settings file for security.

Import Settings

Press **B** to import a previously exported settings file from your computer. This option uses **XModem Send** method to send the file.

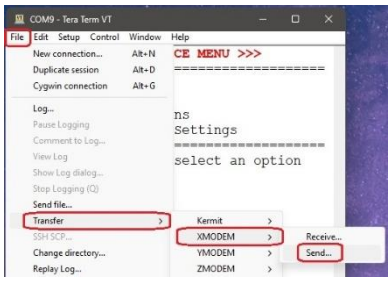
To initiate the **Import**, press **B** to start the process. The screen at the right will appear giving you instructions on how to proceed.



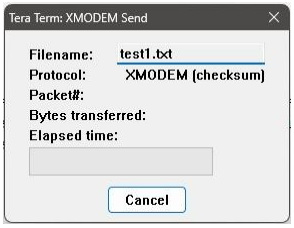
NOTE: This screen will time-out after 60 seconds if a transfer hasn't been started. You can also press any BTN on the unit to exit this screen.

Next, select **File->Transfer->XModem->Send…** in Tera Term, then select a previously exported file from the file list to send to the ***MIDI***.

In this example, I've selected a file named *test1.txt*.

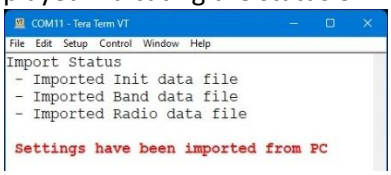


The **XModem Send** dialog box will appear indicating that Tera Term is waiting for the unit to indicate that it is ready to receive the file.



>>> Press-and-release the knob on the *MIDI* to initiate the transfer.

Once the transfer is complete, an **Import Status** screen will be briefly displayed indicating the status of each file imported. If any of the files indicate the import failed, try the **Import** function again.



The program will return to the **Maintenance Menu** once the transfer has completed.

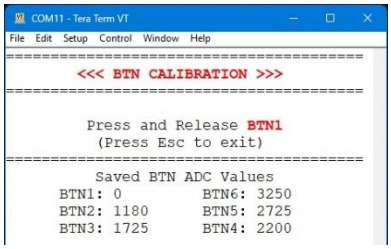
Recalibrate Buttons

Each button on the ***MIDI*** presents a specific voltage to the ADC (analog/digital converter) on the processor. This allows the processor to determine which button is pressed.

To ensure the highest repeatability, the buttons should be calibrated to compensate for manufacturing tolerances.

Press **C** on the **Maintenance** menu to start the calibration process.

Once the **BTN CALIBRATION** menu appears, press each button, one at a time, in sequence, starting with **BTN1** and working your way around to **BTN6**. This records the ADC values for each button. Once you've finished, the calibration screen will update and then return back to the **Maintenance** menu.



Reset to Factory Settings

Press **D** on the **Maintenance** menu to reset the unit back to factory settings. You will need to recalibrate the buttons, and re-enter your WiFi credentials, radio IP address, and call sign.

NOTE: If you don't want to lose your settings, Export them to your PC before resetting back to factory. After the reset, import them back into the ***MIDI*** using the Import function. You'll also want to export your settings if you plan on erasing the flash memory when updating the firmware.

Appendix A: Installing or Updating CTR2-MIDI Firmware

As of v2.00.00, the ***MIDI's*** firmware is distributed as a single BIN file. This simplifies the installation process and reduces the possibilities of entering the wrong offset address for individual BIN files. The address of the single BIN file always starts at **0x0**.

Kits and assembled **CTR2-MIDIs** have the firmware already installed on them but as you might expect, changes will be made to the program over time. To install the latest MIDI firmware on your device, follow these steps:

1. Determine which version of the firmware you need for your device. If your device is a **CTR2-MIDI** (with six BTNs) you'll need the **S3** firmware (for the ESP32-S3). If you are running **CTR2-MIDI** on a **CTR2-Micro** (with three BTNs) you'll need the **C3** firmware (for the ESP32-C3).

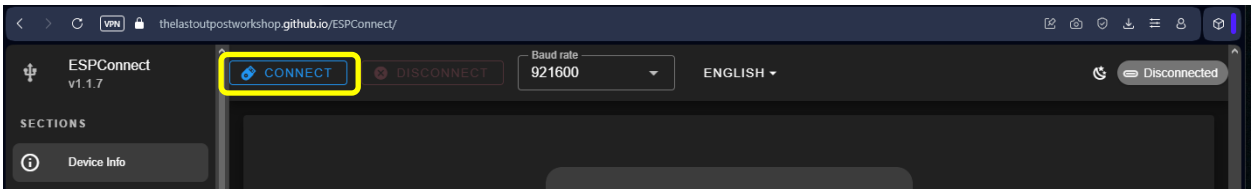
2. Download and unzip the appropriate **CTR2-MIDI** firmware from my web site or the CTR2 Group. Unzip it into a different folder than where you store other CTR2 firmware update files.

Installing Firmware using ESPConnect (New Method)

By far, the easiest method to install or update firmware on an ESP32 process is by using ESPConnect, an open-source ESP32 management project.

ESPConnect is a browser-based tool that must be opened on a Chromium-based browser that supports Web Serial, such as Chrome, Edge, or Opera. It will run on a Windows PC or an Apple Mac. Unfortunately, Linux doesn't support Web Serial in any browser.

To start, open **ESPConnect** here: https://thelastoutpostworkshop.github.io/ESPConnect/



Next, plug your **CTR2-MIDI** unit into a USB port on your computer.

NOTE: Do not connect it to an unpowered USB hub.

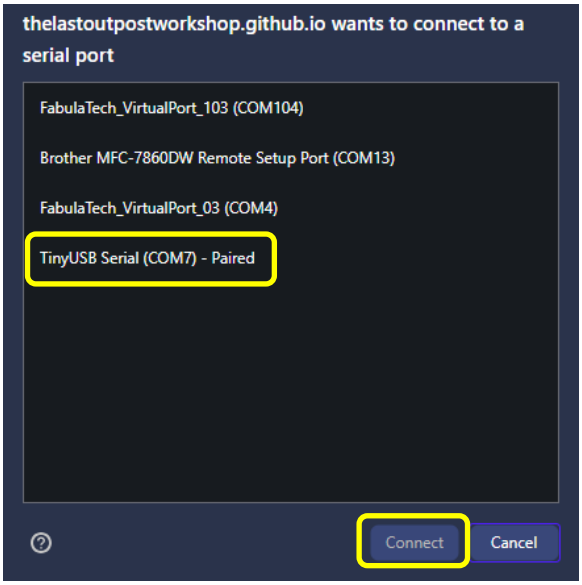
Click the **Connect** button.

A device list will pop up showing the available USB devices on your computer.

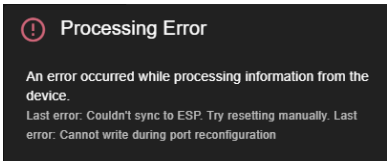
On a Windows PCs, **CTR2-MIDI** will be shown as a **TinyUSB Serial** device, as shown here.

On Macs, **CTR2-MIDI** units will be listed as a **XIAO_ESP32S3** device.

Select the **CTR2** device and click the **Connect** button.



A popup warning will appear telling you that an error occurred. Some errors may tell you to press the BOOT button on the processor to put it into bootloader mode. <u>You do not need to do this.</u>



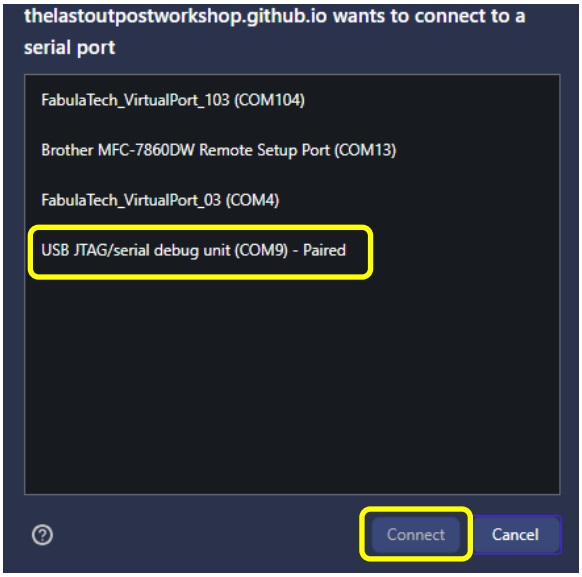
This is the normal response when trying to connect to a **Tiny USB serial** device.

Click the **Connect** button on the home page again.

This time, the **TinyUSB serial** (or **XIAO_ESP32S3** on Macs) will not be shown on the list. Instead, a new device, **USB JTAG/serial debug unit** will be shown (on both PCs and the Mac).

This is the bootloader port and the unit is now ready to program.

Select the **USB JTAG/serial debug unit** device then click **Connect**.



Once you're connected to your **CTR2** unit, the left menu items on the page will be enabled.

<<< WARNING >>>

ESPConnect is a powerful device editor. Many of the functions can cause problems with **CTR2** firmware if you change them. If you don't know what a functions does, DON'T CHANGE IT!

Click on the **Flash Tools** menu item and scroll down to the **Flash Firmware** section.

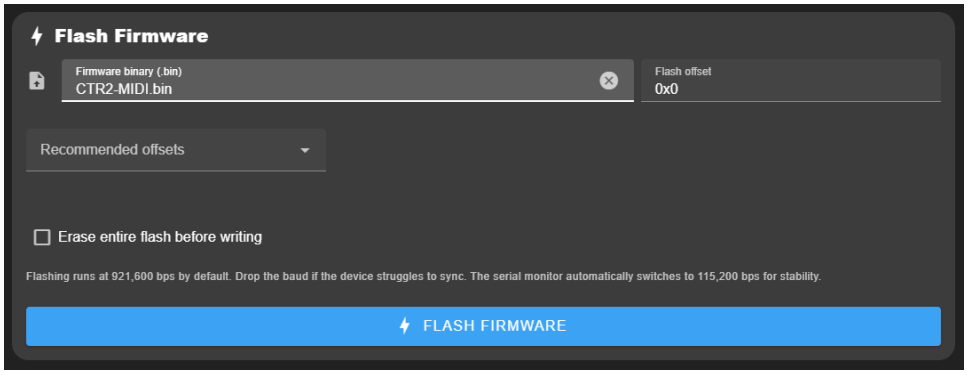
Click the **Firmware binary (.bin)** text box.



A file navigation screen will appear. Navigate to where you unzipped the firmware zip file from the firmware update page and click on the **CTR2-Flex.bin** file.

Your display should now look like this:

Leave the **Flash offset** set to **0x0** and uncheck the **Erase entire flash before writing** checkbox.



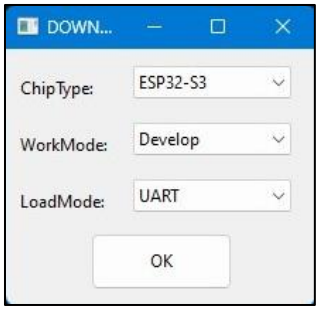
Click the **FLASH FIRMWARE** button to start the flash process.

Once the process completes, cycle the power on your **CTR2-MIDI** unit. The version # will be displayed in a dumb terminal when the unit boots. The version # should match the version # from the firmware zip file.

This completes the firmware update process.

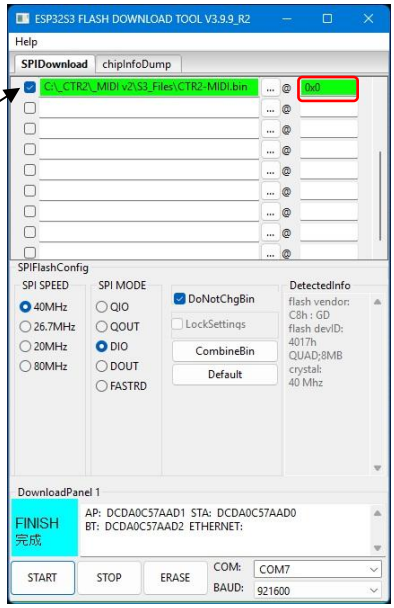
Installing Firmware using EspressIF Flash Download Tool (Old Method)

If you choose to use the EspressIF Flash Download tool, proceed with these steps after you have downloaded the firmware from https://ctr2.lynovation.com/download-ctr2-midi-firmware/



1. Download and open the **EspressIF Flash Downloader Tool**. When it starts, select the **Chip Type** your unit is using (*ESP32-S3* for CTR2-MIDI or *ESP32-C3* for CTR2-Micro hardware). Leave **WorkMode** set to *Develop* and **LoadMode** set to *UART.*

   NOTE: A browser-based flash download tool is also available. See this page on my web site for more information.



2. Map the **CTR2-MIDI.BIN** file unzipped from the **CTR2-MIDI_v3.xx.xx** firmware distribution file into the downloader tool.

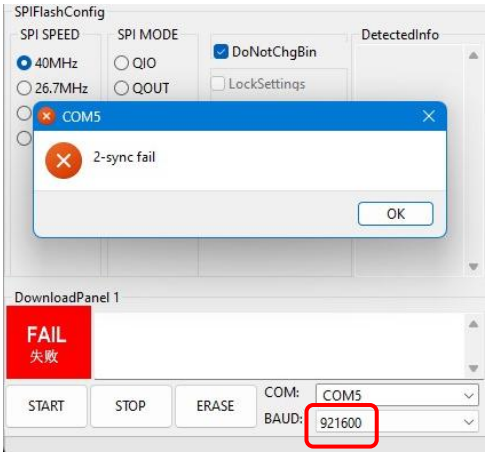
3. Check the checkbox on the left of the **CTR2-MIDI.BIN** filename as shown. This tells the program to flash this file.

   Set the **COM:** port to the port assigned to **CTR2-MIDI** by your computer and set the **Baud** to **921600**.

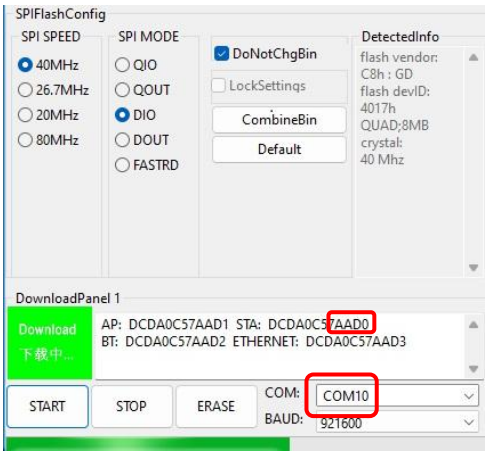
   NOTE: You must use a USB-C data cable. USB-C charge-only cables (supplied with may devices) will not work.

IMPORTANT NOTE FOR ESP32-S3

The ESP32–S3 processor in CTR2-MIDI has two USB UARTs. The first UART (COM 5 in this example, yours will be different) is active when you initially power up the unit and is the COM port you connect to with a terminal program. ***When you select this port the Flash Download Tool will fail and pop up a 2-sync fail window when you press either Start or Erase.*** Press **OK** to clear the alarm window.



Next, drop down the **COM:** list again. You will notice that the initial COM port (COM 5 in this example) is not on the list. You'll also notice a new COM port has been added to the list. On my computer it shows up as COM 10 (your computer will show a different port #). This is the second USB UART on the processor and is used for programming. Select this new COM port.



NOTE: If you don't see the second COM port you will need to press the **PROGRAM** button while you power up the ***MIDI*** to force it into program mode. See the Manually Switch the Processor to Program Mode section for more information on this button.

NOTE: The ESP32-C3 processor in **CTR2-Micro** only has one UART so the above steps are not required. Just select the COM port for the *Micro* when flashing CTR2-MIDI firmware to it.

4. Click the **START** button to start the download.

   NOTE: The unit's Bluetooth ID is based on the last four digits of the Station MAC address that is displayed in the top line of the info box once the flash process starts. In this case it is AADO (circled in the screenshot above). The Bluetooth ID of this unit will be **CTR2_AADO**. Your unit will have a different ID#.

5. Once the download is complete, cycle the power on the unit to start the new MIDI firmware.

   NOTE: The ESP32-S3 processor will revert back to its initial COM port (COM 5 in this example) after the reboot.

NOTE: Clicking the **ERASE** button will erase the entire flash memory including the setup file. This will reset any custom MIDI control types you have selected for **MIDI** mode, and all custom settings in **Flex** mode. It also resets the BTN ADC values back to default values. You will need to recalibrate the buttons when you start the program and edit the maps on the ***MIDI*** to restore your custom settings. MIDI Mapping in your app is not affected.

NOTE: If you are flashing the ***MIDI*** firmware to a **CTR2-Micro** (-C3 processor) erasing the flash will also delete the configuration files used with the normal *Micro* firmware.

Installing Firmware using Linux or Mac

Mac users should use the new method using ESPConnect, described above. Linux users have no other option, other than to follow the procedure below.

A script file is also supplied in the firmware update zip file. This script file can be used in a Linux or Mac environment if you don't have access to a Windows computer.

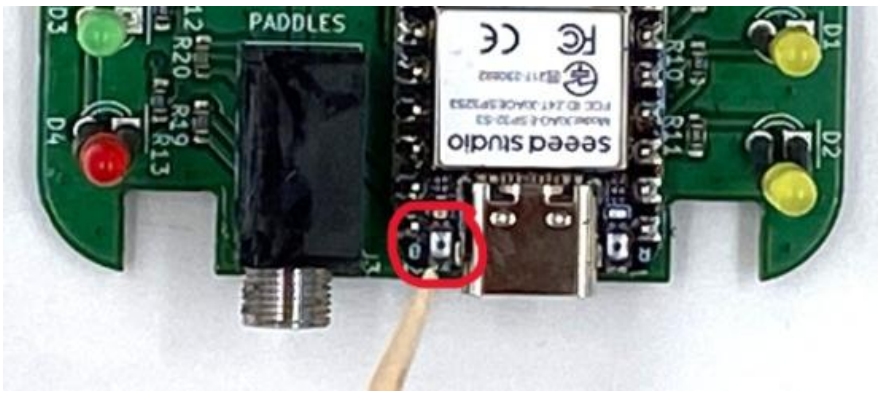
Instructions for using this script file are include in the CTR2-Micro Operation Manual in **Appendix B**.

> **NOTE:** A browser-based flash download tool is also available. See this page on my web site for more information.

The firmware that allows USB MIDI control, changes the way the virtual COM port works on the ESP32-S3. One COM port is used in the normal operating mode and another COM port is used for flashing the firmware. In order to flash new firmware to the unit you must force the ESP32-S3 to switch from the normal operating mode to programming mode.

Manually Switch the Processor to Program Mode

1) Power the unit off.
2) Press-and-hold the **PROGRAM** button down on the processor board then apply power to the board. Yes, it's that little black dot on the square silver pad the toothpick is pointing to in this photo!



A notch has been cut in the enclosure next to the USB-C connector to allow you to access this button without opening the enclosure.

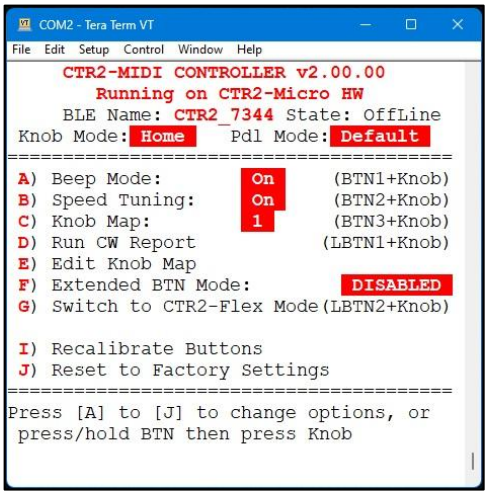
It's still hard to do especially if your eyes aren't as good as they use to be. Insert the toothpick only about 6mm (1/4") and gently press down. You will feel a light click. If you insert the toothpick too far it will rest on the chip LED in back of the switch and will not press the button. Once you have the button pressed power up the unit. If it's in programming mode the normal LED boot sequence will not run. Use a terminal program to find the Program UARTs COM port assignment.



Once the unit is in Programming mode, edit and run the script to flash the firmware.

NOTE: To use the script, you must set it as an executable file and grant permission in you OS to access the **DIALOUT** group. You'll also need to edit the file to add the serial port ID and the path to the CTR2-MIDI.BIN file.

Appendix B: Running CTR2-MIDI on CTR2-Micro Hardware



The CTR2-MIDI firmware can be flashed to a CTR2-Micro. This gives current ***Micro*** users access to the ***MIDI*** features without having to buy new hardware.

NOTE: CTR2 Relay is not available on CTR2-Micro hardware.

NOTE: Because of the way the buttons are sampled on **CTR2-Micro** hardware (with the ESP32-C3 processor), you must recalibrate the buttons when switching between **CTR2-Micro** firmware and **CTR2-MIDI** firmware.

The terminal display for ***MIDI*** firmware running on the *Micro* is similar to the normal ***MIDI*** display except the **USB Name** and the Bluetooth Power option is not shown because these options are not available on the *Micro.*

There are several limitations to using the ***MIDI*** firmware on the *Micro*:

- **CTR2-Micro** hardware only supports Bluetooth-LE MIDI and WiFi. The ESP32-C3 processor in the *Micro* doesn't have the hardware to support USB MIDI or AetherSDR's **CTR2 Relay**.
- Once you have flashed the ***MIDI****'s* firmware on your *Micro* most of the features you use on the *Micro* will no longer be available. This includes the following:
    - WiFi is not supported in **MIDI** mode. To control a Flex radio over WiFi, switch to **Flex** mode. **Flex** mode does not support CTR2-Voice
    - Serial CAT is not supported so you can't control other radios with the *Micro*
    - The *Micro's* keyer is not supported so you can't use a terminal as a keyboard keyer or use any CW message buffers you had set up in the *Micro*. **Flex** mode does have a built-in keyer and has its own set of CW message buffers.
    - Physical Key and PTT Output are not supported; however, **Flex** mode supports Key and PTT over WiFi
    - Favorite Frequency and Previous Frequency lists are not supported
    - The web browser interface is not supported
    - You must reflash the **CTR2-Micro** firmware on your *Micro* to restore its normal features
- The *Micro* only has three buttons so you can only map 6 MIDI **Buttons** in the app (or 24 if you enable Extended Button Mode).
- The *Micro* only has one LED. This LED flashes to indicate the current mode of the unit. It does not indicate long-press mode if **Beep** mode is turned *Off*.
- CW Report and Switch Mode options (**BTN4+Knob** and **BTN5+Knob**) are accessible using *long-press* **BTN1+Knob** and **BTN2+Knob** sequences.

Flashing the Firmware

To flash the ***MIDI*** firmware to your *Micro,* unzip the **CTR2-MIDI.BIN** binary file from the ***MIDI ESP32-C3*** firmware distribution zip file into a unique folder then follow the instructions in **Appendix A**.

You will need to complete the **BTN Calibration** on the *Micro* to save the BTN ADC counts in the ***MIDI****'s* initialization file. This only needs to be done once.

> **NOTE:** The ***MIDI*** firmware maintains its own file structure therefore the settings you had for the *Micro* firmware are not lost. Just reflash the *Micro* firmware to your *Micro* to restore your *Micro* to its normal operation – **do not Erase the flash memory** in the Flash Download tool.

Operating the *Micro* with *MIDI* Firmware

MIDI firmware can run on *Micro* hardware, with a few differences. You will need to go through the same steps to connect the *Micro* to your iPad and you will need to map the controls on the *Micro* to the app. The main difference is that you will only have 6 MIDI **Button** functions (two on each BTN) and all of the knob modes (24 if you enable Extended BTN Mode).

LED Indications

The big difference between the ***MIDI*** and the *Micro* is with the LED indicators (or lack thereof). The *Micro's* status is indicated by the flash sequence of the status LED.

The table below summarizes these flash sequences and the beeps associated with them.

Knob Mode	Flash Sequence	Buzzer Tone when entering this mode	Description
Home	One long flash	One high frequency beep	Primary knob sends **Control Change (CC) 100** Push and turn sends **CC 101**
Mode 1	One short flash	One low frequency beep	Primary knob sends **CC 102** Push and turn sends **CC 103**
Mode 2	Two short flashes	Two low frequency beeps	Primary knob sends **CC 104** Push and turn sends **CC 105**
Mode 3	Three short flashes	Three low frequency beeps	Primary knob sends **CC 106** Push and turn sends **CC 107**

The **Paddle Mode** determines the rate of the LED flash.

- When **Paddle Mode** is in **Normal** mode (controlling MIDI **Buttons 20** and **21)** the LED flashes the status at 2 second intervals
- When **Paddle Mode** is in **Extended** mode (controlling MIDI **Buttons 30** and **31**) the LED flashes the status at 1 second intervals

Modified BTN#+Knob Functions

Since the *Micro* hardware only has three physical buttons the BTN#+Knob options are slightly modified.

The BTN4+Knob function to play the report or change PTT modes is accessed by *long-pressing* **BTN1+Knob**.

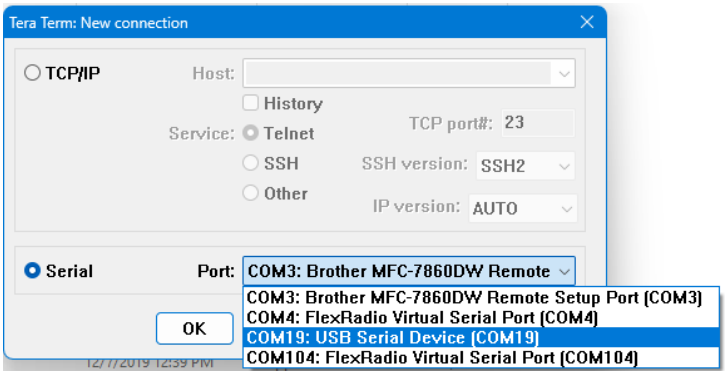
The BTN5+Knob function to switch modes is accessed by *long-pressing* **BTN2+Knob**.

The BTN6+Knob function to toggle WiFi (in Flex mode) is accessed by *long-pressing* **BTN3+Knob**.

Appendix C: Configuring Tera Term

If you search for Tera Term you will find a lot of garbage with malware attached to it. I've downloaded a clean copy of Tera Term v4.106 and posted it in the CTR2 Group Tera Term Install File section. You can download it here. As far as I know, Tera Term is only available for Windows.

When you first open Tera Term you'll be presented with the **Tera Term New connection** window. Simply select the **Serial** radio button, select the COM port Window's assigned to your Micro when you plugged it in, and click the **OK** button.



Since you are connecting to a USB serial port there is no need to set the baud rate. It will run at USB speed regardless of the baud setting.

That's it! Tera Term will connect to the ***MIDI***. Press any key to open the configuration display.

You can change the terminal size in the **Setup** menu. Select **Terminal…** Set the **Terminal Size** to *41 x 20.* The ***MIDI****'s* terminal interface was designed for this size.

While in the **Terminal…** settings verify the **New-line** options are set to *CR* for both **Transmit** and **Receive** and the **Terminal ID** is set to *VT100*.

You'll probably want to change the font size and colors. These are also changed in Tera Term's **Setup** menu. Select **Display** to change the font and background colors to your liking. Select **Font** to change the font and font size. I like *Courier New*, *Regular*, and *14-point size*. Your preferences may differ.

Once you have the program configured the way you like, select the **Setup->Save Setup…** menu and save your configuration. If you use the default file name, TERATERM.INI the program will automatically start a Telnet session using the COM port you selected above when it opens. This provides one-click access to your ***MIDI***.

Appendix D: Configuring Putty

Putty is a terminal program that can be configured for a variety of needs. The ***MIDI*** only supports serial connections. This section describes how to configure the program to interface with the ***MIDI***.

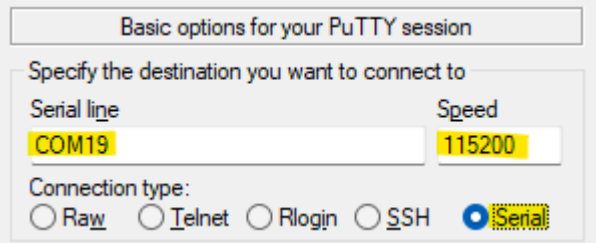
Download Putty for Windows from https://www.putty.org/. It's also available for Linux at https://www.ssh.com/academy/ssh/putty/linux and for Mac at https://www.ssh.com/academy/ssh/putty/mac.

You'll need to connect to the ***MIDI*** using its USB serial port in order to configure it.

Serial Session

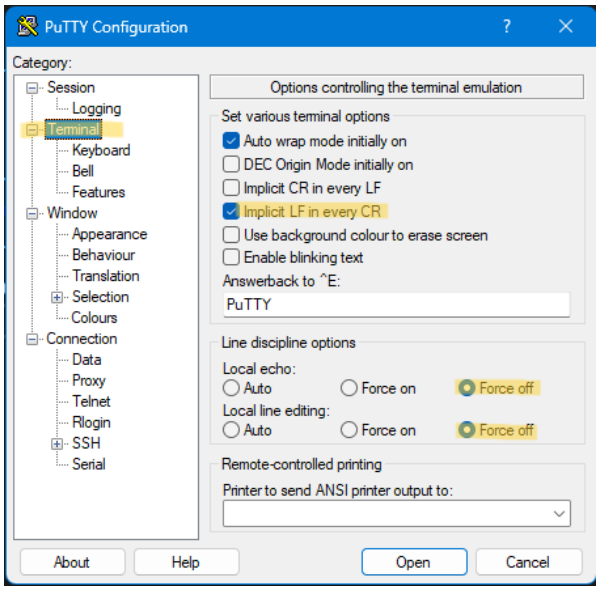
Select **Serial** then set the **Serial Line** to the COM port you found in the Device Manager and set **Speed** (Baud Rate) to 115200.

NOTE: Since this is a USB serial port the **Speed** (baud rate) doesn't matter. Data will be sent at USB speeds regardless of the **Speed** setting.

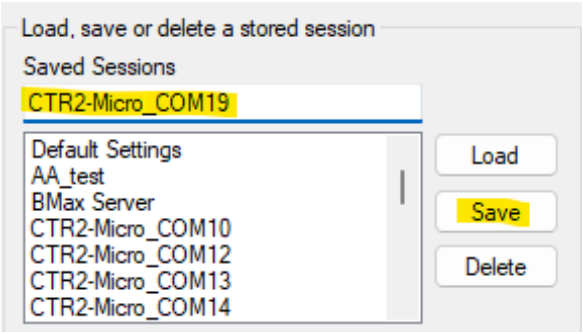


Next, select the **Terminal** item and set the **Implicit LF in Every CR** to on, and **Local Echo,** and **Local Line Editing** to **Force Off**.

You can change the window size under the **Window** item. Set the **Columns** to *41* and the **Rows** to *20*.



Once this has been done, return to the **Session** menu item, enter a name for this session and click the **Save** button. This allows you to easily re-open this session with just a couple of clicks.



If you right-click on the Putty icon in the Windows toolbar the last few sessions you had open will be displayed. Just select the one you want to open it.

You can adjust the display colors on the **Windows->Colours** menu item. The Micro uses the **Bold** attribute to highlight the ***hotkeys*** and other items. I like to set the **Background** color to blue and the **Bold** color to yellow but you can find the colors that work for you. After you get a color combination you like return to the **Session** menu and **Save** the session.

Appendix E: Apple or Linux Terminal Programs

The Apple Mac and Linux have built-in terminal programs so there is no need to install a separate app. To connect your ***MIDI*** to a terminal session, use the following process.

First, list your current serial ports without the ***MIDI*** plugged in.

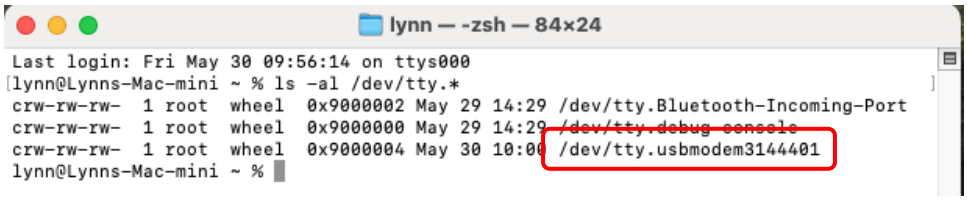
- On the Mac open **Applications/Utilities/Terminal.app**. On Linux open the terminal program supplied by your distro.
- On the Mac, enter ***ls -l /dev/tty.usb****, on Linux, enter ***ls /dev/tty**** This will return a list of all known serial ports.
- Next, plug the ***MIDI*** into the computer's serial port and execute the command above again. This is easily done by pressing the *Up*-arrow key.
- Compare the new list with the old list. The *Flex's* serial port ID will appear on just the new list. For Mac users the serial port ID format will be **/dev/tty.usbserialxxxxx** where **xxxxx** is a unique device ID #. Linux users will see something like **/dev/ttyACMx** or **/dev/ttyUSBx**, where **x** is a unique # for that port.

NOTE: If a new virtual serial port is not created when you plug your Micro into your PC make sure you are using a USB-C cable that supports data. Many USB-C cables only provide power to the remote device.

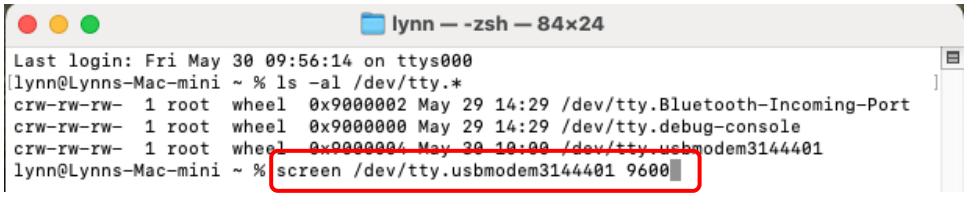
Once you know the ***MIDI****'s* USB serial port ID, write it on the label on the bottom of the unit using a fine-tipped permanent marker for future reference. Put a piece of transparent tape over the label to seal the ink so it doesn't rub off (it's not as permanent as you think). You can always remove the tape if you want to change what's written on the label.

Once you have the serial port ID, enter the following: **screen {serial port ID} 9600**. Include the complete device description (i.e. **/dev/ttyxxxxxxx**) for the **serial port ID**. This will open the serial port using 9600 baud in a terminal session. The following screenshots demonstrate these steps.

1. Get the list of serial devices on your computer. We're looking for the **usb** device.

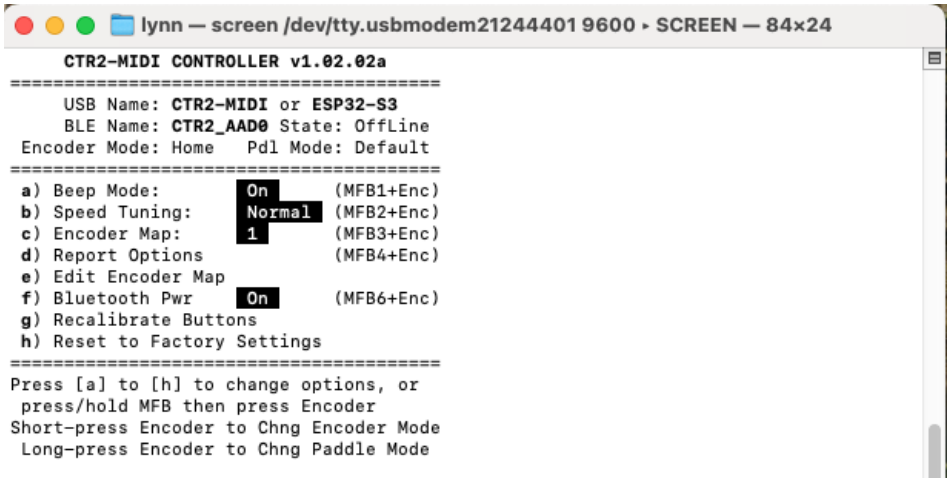


2. Open the **screen** utility using the **usbmodem3144401** device. Your USB device will have a different #.



3. The Apple and Linux terminals don't offer a lot of customization, and they don't support XModem transfers for exporting and importing setting file backups.

   This is the ***MIDI*** terminal display. Press the indicated key to select an option.



Appendix F: Default Knob and Button Assignments

The factory default button assignments are shown on the spreadsheet below. I've tried to group them by function (General, Mode, Band, and CW) so they're easier to remember.

You can change these assignments in the Control Menu. An empty chart is included in Appendix G that you can print and fill in with your button assignments.

You can print the worksheet for reference.

You can download the Excel file for this spreadsheet here.

CTR2-MIDI Flex Mode Extended Button Worksheet
Factory Default Settings

This sheet documents the factory default setting of the Knob modes and Button functions

Excel files can be fouind here

Knob Mode P&T = Push & Turn

Firmware v2.00.00

Button Modes: S=Short-press, L=Long-press

Knob Home

Knob Assignments	
Mode	Function
Turn	VFO
P&T	Volume

Button Assignments	
BTN	Function
1S	Mode Up
1L	Mode Down
2S	Band Up
2L	Band Down
3S	Mute
3L	Zero Low Digits



Button Assignments	
Function	BTN
Tune Step Inc	6S
Tune Step Dec	6L
Home Freq	5S
Set Home Freq	5L
Start ATU	4S
Bypass ATU	4L

Knob #2

Knob Assignments	
Mode	Function
Turn	NR Level
P&T	Mic Gain

Button Assignments	
BTN	Function
1S	80 m
1L	160 m
2S	40 m
2L	60 m
3S	20 m
3L	30 m



Button Assignments	
Function	BTN
6 m	6S
VFO Lock	6L
10 m	5S
12 m	5L
15 m	4S
17 m	4L

Knob #1

Knob Assignments	
Mode	Function
Turn	AGC-T
P&T	Panadapter Zoom

Button Assignments	
BTN	Function
1S	CW
1L	DIGL
2S	LSB
2L	DIGH
3S	USB
3L	RTTY



Button Assignments	
Function	BTN
AGC	6S
NR On/Off	6L
FM	5S
Tx Processor	5L
AM	4S
SAM	4L

Knob #3

Knob Assignments	
Mode	Function
Turn	CW Speed
P&T	RIT Freq

Button Assignments	
BTN	Function
1S	Macro 1
1L	Macro 5
2S	Macro 2
2L	Macro 6
3S	Macro 3
3L	Macro 7



Button Assignments	
Function	BTN
RF Gain	6S
Tx Enable	6L
Bandwidth	5S
RIT On/Off	5L
Macro 4	4S
Breakin	4L

Appendix G: Flex Mode Button Assignment Worksheet

Use the worksheet on the next page to record Knob and Button assignments if you change them from the default settings.

You can print the worksheet for reference.

You can download the Excel file for this spreadsheet here.

CTR2-MIDI Extended Button Worksheet

Use this sheet to document your custom settings

Knob Modes: P&T = Push & Turn

Button Modes: S=Short-press, L=Long-press

Excel files can be fouind here

Firmware v2.00.00

Knob Home

Knob Assignments	
Mode	Function
Turn	
P&T	

Button Assignments	
BTN	Function
1S	
1L	
2S	
2L	
3S	
3L	



Button Assignments	
Function	BTN
	6S
	6L
	5S
	5L
	4S
	4L

Knob #2

Knob Assignments	
Mode	Function
Turn	
P&T	

Button Assignments	
BTN	Function
1S	
1L	
2S	
2L	
3S	
3L	



Button Assignments	
Function	BTN
	6S
	6L
	5S
	5L
	4S
	4L

Knob #1

Knob Assignments	
Mode	Function
Turn	
P&T	

Button Assignments	
BTN	Function
1S	
1L	
2S	
2L	
3S	
3L	



Button Assignments	
Function	BTN
	6S
	6L
	5S
	5L
	4S
	4L

Knob #3

Knob Assignments	
Mode	Function
Turn	
P&T	

Button Assignments	
BTN	Function
1S	
1L	
2S	
2L	
3S	
3L	

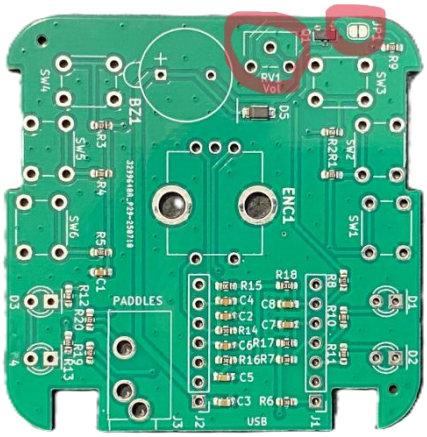


Button Assignments	
Function	BTN
	6S
	6L
	5S
	5L
	4S
	4L

Appendix H: Adding a Volume Control Pot

The original ***MIDI*** hardware does not include a volume control pot. There is provision for one, but a fixed resistor was used to set the beep volume.

To add a volume control to your unit, you'll need to purchase the potentiometer. The PCB is designed for a [Bourns 3306W-1-503,](#) 50K ohm unit, available from Mouser or other sources. You'll need to cut the small trace between the two pads on JP1. The locations for RV1 and JP1 are shown in the photo at the right.



Next, drill a 3.2mm (1/8") hole in the encloser on the edge opposite the USB-C connector and paddle jack. This is the bottom edge. BTN3 and BTN4 are closest to this edge.



Appendix I: Change Log

v2.01.01: March 24, 2026

- Fixed PTT operation so it works properly in Flex mode
    - Added short beeps when PTT is keyed and unkeyed
- No changes to MIDI mode

v2.01.00b: March 7, 2026

- Added a new Quick Start Guide to help new users get their **CTR2-MIDI** up and running quickly

v2.01.00a: February 12, 2026

- Updated Appendix A: Installing or Updating CTR2-Flex Firmware to include ESPConnect as the primary method to flash firmware to the **CTR2-MIDI** on PCs and Macs (sorry, not Linux).

v2.01.00: February 3, 2026

- Simplified the firmware descriptions in this manual. **MIDI** mode refers to the firmware using the MIDI protocol to control a 3rd-party radio control app. **Flex** mode refers to the firmware using the Flex API protocol and WiFi to connect to, and control, a Flex radio without a 3rd-party app.
- Added HID support for RemoteTx to the **MIDI** mode firmware
    - **NOTE: HID** and **RemoteTx** support only works in **MIDI** mode

v2.00.02: December 26, 2025

- Additional refinements to paddle debounce code to prevent random code elements
- Fixed bug in WiFi menu that blocked connecting to WiFi (option G)

v2.00.01: November 10, 2025

- Added paddle release debounce to fix issues with cheaper paddles adding extra code elements
- Improved keyer Bug mode so it works properly now
- The **CTR2-MIDI firmware is now distributed as a single BIN file** named **CTR2-MIDI.bin**. The instructions for flashing the single BIN file have been updated in Appendix C.

v2.00.00: November 3, 2025

- Added a new **Flex** mode to **CTR2-MIDI** firmware. This mode is similar to the dual-boot mode in **CTR2-Flex/CTR2-Dial** firmware
    - **Flex** mode was ported from CTR2-Flex firmware. It allows you to connect CTR2-MIDI directly to your Flex radio over your local WiFi network. This mode supports all versions of SmartSDR, including Windows versions. Radio control is done directly with the radio using the Flex API, not MIDI, so no 3rd party apps are required for this mode. **CTR2-MIDI's Flex** mode does not support SmartLink.
- Added a new Keywords and Definitions section
- Replaced all references to the term "encoder" in this manual and in the firmware with the term "knob".
- Replaced all references to the term "MFB" (i.e. Multi-Function Button) in this manual and in the firmware with the term "BTN".