CTR2-Max

MIDI Mode Manual

V2.00.01



Last Revision: October 28, 2026

Copyright 2026 by Lynovation.com
All rights reserved

Updated to firmware v2.00.01

Revised sections for this version are highlighted in yellow

Contents

How to use this manual 4
Legal Notice 4
Change Log 5
v2.00.01 – October 28, 2026 5
Introduction 6
MIDI Mode 6
Flex Mode 7
AetherCTR2 7
System Overview 8
Quick Start Guide 9
Basic Concepts 9
Initial Startup 10
Connect to your App 10
Configure the MIDI Mapping in your App 10
Editing Labels in CTR2-Max 11
Importing Configurations into CTR2-Max 11
Time to Play! 11
Hardware 12
3.5mm Jacks on Max 12
Internal Battery Option 12
How MIDI Mode Works with Your Control App 13
Multiple Controllers / Two Modes 14
Switching Operating Modes 15
CTR2-Max's MIDI Mode Features 15
MIDI Control Types 16
MIDI Button Pairs for Dial Button Mode 17
Home Page 18
Dial Control 18
Dial Graphics 19
Tuning Lock 20
Dial Menu 20
Controls Menu 20
Control States 21
Editing Knob and Button Assignments 21
Buttons Menu 22
Assigning a Knob Switch or Pushbutton to a Button Function 22
PTT Page 23
Using a Hardware PTT Switch and Straight Key 24
Settings Menu 24
Label and Settings Maps 27
Editing Labels and Control Types 27
Dial Editor 28
Button Editor 28
Editing With a Terminal 29
Backing Up and Restoring your Settings 30
Export Settings 31

Import Settings 32
Editing 3rd-Party App Maps 33
Remote Ham Radio Maps 33
RemoteTx Mapping 34
PTT Input 35
iOS and MacOS App Maps 35
Thetis Maps 36
RimuSDR Maps 38
AetherSDR Maps 39
New CTR2-Max Features in AetherSDR 39
SDRoxide Maps 40
Advanced Options 40
Operation with N1MM Logger+ 40
Troubleshooting 41
Appendix A: Connecting CTR2-Max to Your App 43
Remote Ham Radio 43
RemoteTx 43
AetherSDR, RimuSDR, and SDRoxide 43
Marcus' Apps for iOS or MacOS 44
Other Windows Applications 45
Thetis 45
SparkSDR 46
Appendix B: Installing or Updating CTR2-Max Firmware 47
Installing Firmware using ESPConnect 47
Enter your Call and Registration Key 49
Appendix C: Configuring Tera Term 50
Appendix D: Configuring Putty 51
Serial Session 51
Appendix E: Apple or Linux Terminal Programs 53
Appendix F – Power Requirements 55
Appendix G: Using CTR2-Max with Non-MIDI Programs 56
Configuring CoyoteMIDI 56
Configure SmartSDR for Windows Frequency Control 56
Configure wfView Frequency Control 57
CoyoteMIDI Startup Options 58
Appendix H: Change Log 59

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. 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.

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. I will work with you if there is a problem with your unit on arrival.
- Each **Max** is hand-built and uses a 3D printed enclosure. This means that there may be small cosmetic flaws on your unit.
- Units are not "plug and play". Be prepared for a learning curve, especially when mapping MIDI commands to your radio control app for the first time. There are videos on my YouTube channel that can help with this.
- I make no warranty that the firmware provided will perform up to your expectations or be suitable for your application. A lot of compromises had to be made to fit the small display so review this manual to be sure you're comfortable with the user interface.
- Software bugs are a fact of life and I try to find and correct all bug reports ASAP to the best of my ability.
- **Max** requires a license key to be fully functional. License keys can be purchased separately if you build your own unit. A license key is included with your purchase of a **Max** controller from Lynovation.com.
- License keys are tied to your call sign. This allows you to use the same registration key on as many CTR2 devices as you own. You are not allowed to include your license key on units you build to sell or give to others. They will need to purchase their own key.
- Contact me if you sell your **CTR2** unit. I will create a new key from call sign of the new owner.

Change Log

v2.00.01 – October 28, 2026

- Added the Battery icon back in. The internal battery is useful when operating with a battery-operated device such an iPhone or iPad. It powers Max without draining the device's battery.

v2.00.00 – October 26, 2026

- FCC rules forced the removal of the RF features in **CTR2-Max**. This was due to the display/ESC32-S3 used in **Max** is not FCC accepted for RF emissions. Because of this, **Max** only supports USB connectivity to radio control apps. Direct connection to Flex radios over WiFi, or Bluetooth connections are no longer supported.

v1.01.03 – September 30, 2026

- Fixed a bug I injected into v1.01.01 that erased the map name in **MIDI** mode

v1.01.01/.02 – September 28/29, 2026

- No changes to **MIDI** mode firmware – just updated this manual's version to keep up with the firmware release.

v1.01.00 – September 14, 2026

- Replaced the **Knobs** menu with a new **Controls** menu
- Changed the recommended file extension for **CTR2** configuration files from **.txt** to **.ctr2**. They are still text files; this just makes the easier to identify.

v1.00.02 – August 15, 2026

- Fixed bug that displayed the wrong button for editing in a dumb terminal
- Clear all MIDI dial and button labels when importing a new file
- Added information on MIDI mapping for RimuSDR and AetherSDR Flex control apps

v1.00.01 – August 9, 2026

- Initial product release

Changes to previous versions can be found in Appendix H

Introduction

CTR2-Max (referred to as simply **Max** in this document) is the descendant of a long line of microcontroller-based radio controllers from Lynovation. If you're interested in its background, you can read the QEX and QST articles about the **CTR** series here.

Recent changes in FCC certification rules for RF emitters means that **Max**, unlike other **CTR2** controllers, does not support **WiFi** or **Bluetooth** RF modes. It only supports USB connections to the radio control apps you use. I know this is disappointing for many potential users, but I felt that it was best to comply with the regulations.

Max supports both USB MIDI control of your radio control app, and direct control of your Flex radio over USB using new **CTR2 Relay** mode built into AetherSDR. In addition, the development team at **AetherSDR** offers another firmware called **AetherCTR2** that replaces the factory **Max** firmware and is supported by the **AetherSDR** developers. More about **AetherSDR** and **AetherCTR2** can be found in the CTR2-Max Flex Mode Manual.

You can easily switch between the two modes in the **Settings** menu to fit your operating needs. You should refer to the CTR2-Max Flex Mode Manual for information about **Flex** mode operation.

Max and **AetherCTR2** firmwares only run on **Max** hardware with the ESP32-S3 2.8" touchscreen.

Max is the most capable of all **CTR2** controllers. It has five dual-function knobs, four dual-function pushbuttons, a paddle jack, a Key/PTT jack, and an optional Sidetone audio interface board that allows you to patch the keyer's sidetone in **Flex** mode into your headphones, powered speaker, or PC sound card.

MIDI Mode

When operating in **MIDI** mode, each knob, knob switch, and button function are hardwired to specific MIDI commands, *not radio control functions*. For instance, the first virtual dial control sends MIDI Change Command (CC) #100. The second dial control sends CC #101, etc. You can change the label assigned to each control and you can change the physical knob or switch each control is assigned to in the **Controls** menu.



The real magic happens when you connect **Max** to a radio control app. Each MIDI command is first labeled in **Max** and then mapped in the control app to execute a that function. For instance, CC #100 is typically labeled "VFO" in **Max** and mapped to the VFO control in the app. This control is usually assigned to knob E (large knob) on the controller.

MIDI is a one-way protocol, so when using this mode, the radio control app provides the radio information.

Many apps are capable of receiving MIDI commands. These include Marcus' (DL8MRE) excellent SmartSDR/SDR-Control/FT-Control, TS-Control, and K4-Control for iOS and MacOS, AetherSDR, RimuSDR, SDRoxide, SDR-Console, and SparkSDR, to name a few. **MIDI** mode is also supported by several browser control apps like Remote Ham Radio, Online Remote Radio, and RemoteTx.

Flex Mode



Flex mode creates a custom Flex controller specifically designed for the Flex 6000, 8000, and Aurora SDR radios using the Flex API. It works with the **CTR2 Relay** in **AetherSDR**. You can find this feature under **AetherSDR's Settings -> Controllers –> CTR2…** menu. To use this mode, connect **Max** to a USB port on the computer running **AetherSDR**, then start the **CTR2 Relay**. **AetherSDR** creates a non-GUI connection to the radio it's currently controlling and then routes Flex API messages between **Max** and the radio over the USB connection**.** This mode provides real-time CW keying and PTT control.

Unlike **MIDI** mode, the Flex API is a two-way protocol, so **Flex** mode is aware of the radio's operating state (frequency, mode, band, etc.) A **Dashboard** display provides easy access to many of the radio's functions.

Max provides a dedicated CW keyer and sidetone in **Flex** mode because the latency from operating the paddles to keying the radio over the network is enough to disrupt your fist.

AetherCTR2



AetherSDR has graciously provided an alternative firmware program for **Max** called **AetherCTR2**. This firmware can be installed on **Max** over the **AetherCTR2** USB link to **AetherSDR**. Once installed, you can run either **Max's** stock firmware, or **AetherCTR2**.

AetherCTR2 provides additional features not included in the factory **Max** firmware, such as S and power meters, CW decoding, and an audio spectrum display.

The developers at **AetherSDR** created **AetherCTR2** and provide support for it.

More information about **Flex** mode and **AetherCTR2** can be found in the Flex firmware manual.

System Overview

CTR2-Max is a custom-built radio controller that is designed specifically for the needs of amateur radio operators. It includes a large, 2.8" color touchscreen with up to 5 knobs and 4 buttons that are dual-mode (turn and push-and-turn knob functions and short-press and long-press button actions) for a total of 18 physical controls. These can be assigned to any of the virtual dial and button controls.

Knob and switch functions are displayed on the **Controls** menu. This page graphically represents the controls on the unit. The function assigned to each control can be easily changed to fit your operating situation.

Unlike generic MIDI controllers, all **Max** controllers include a 3.5mm (1/8") CW paddle input jack and a 3.5mm key/PTT input jack.

For **Flex** mode users, a transformer-coupled sidetone audio interface board is available that allows you to patch **Max's** keyer sidetone into the receive audio from your radio. The sidetone from the keyer is sinusoidal and is generated by a codec. A 1.2-watt amplifier drives a large speaker, giving you plenty of sidetone volume.

Information on the sidetone audio interface and internal battery options can be found in the CTR2-Max Hardware manual.

NOTE: The CW keyer, sidetone, and **Sidetone Audio Interface** board are not used in **MIDI** mode.

The unit requires a clean USB power source. **DO NOT** power the unit from an unpowered USB hub. A 3.7-volt Lithium-Polymer battery can be user-installed, but since the unit requires a USB connection to the control the radio control app, the internal battery only helps run the unit when connected to a laptop, iPhone, or iPad that's running on battery. Batteries up to 10mm x 35mm x 55mm can be accommodated. A 2000 mAH battery will power just **Max** for about 12-hours. The built-in charge controller automatically ramps down the charge current as it approaches full-charge. A battery icon can be displayed that shows the estimated state of charge.

MIDI mode doesn't "talk" to your radio directly. It uses MIDI commands to tell a radio control app what to do. The MIDI protocol was developed to control musical instruments. It's a very small, simple, and fast protocol. Many radio control apps are adopting MIDI to allow other devices to control their functions instead of using proprietary CAT protocols.

Since **MIDI** mode doesn't receive updates from the app, there is no **Dashboard** like the one in **Flex** mode. The **Controls** screen gives you easy access to the physical knob and switch functions.

The main advantage of using a MIDI controller like **Max** over using a mouse is that the radio control app doesn't have to be "in focus" on your computer for **Max** to control it. MIDI commands from **Max** go directly to the app that it is registered to. You can change frequency or other parameters using **Max** while you are entering log data or working in another program.

Like a mouse, **Max** only sends commands. It does not receive information back from the app or the radio. It has no idea what is going on in the "real world". The 3rd-party app you're using manages the user interface and the control communications with the radio.

The next section will help you configure **Max** and get it setup up with your 3rd-party app.

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.

*Your **CTR2** controller is not one of those tools! Be prepared for a learning curve as you discover the features in your controller.*

MIDI mode uses USB MIDI, or USB HID, to control a 3rd-party app. That app in turn controls your radio. The radio you can control, and the functions you have available, are determined by the 3rd-party app.

Pre-configured maps and spreadsheets for a variety of radio control apps can be downloaded from the CTR2-Max Config Files web page.

The goal of this section is to give you a basic understanding of how **MIDI** mode works so you can start using your **Max**.

Basic Concepts

Let's start with how **MIDI** mode works with your 3rd-party app.

CTR2-Max Controller

All of the controls (Dials and Buttons) in **MIDI** mode are "hardwired" to MIDI command numbers. You can only edit the label associated with each control. What each dial or button does is determined by the MIDI mapping in the 3rd-party app.

Unlike **Flex** mode, which monitors and displays the status of the functions it controls, **MIDI** mode is a one-way controller. It only sends MIDI commands to the 3rd-party app. It has no awareness of what the radio or control app are actually doing, therefore you won't find a **Dashboard** in **MIDI** mode.

3rd-Party Control App

The key to controlling your radio with **Max's MIDI** mode is the 3rd-party app that it controls. The app's MIDI map translates the incoming MIDI commands from **Max** to radio functions such as tuning the frequency or changing the volume. There are many 3rd-party apps available. They are generally designed for a specific radio manufacturer, and sometimes a specific radio model. Examples for Flex radios include several iOS/MacOS apps by Marcus, DL8MRE, AetherSDR and RimuSDR (which run in Windows, Mac, and Linux), Thetis (Windows) for the Apache and Hermes Lite 2 radios, and SDR-Console (Windows) by Simon Brown, among others. If your app supports MIDI control, **Max's MIDI** mode should work with it.

As an added bonus, **Max** works with several browser-based control apps such as RemoteHamRadio, RemoteTx, OnlineRemoteRadio, and Hams.Live. Many of these apps use USB HID instead of MIDI to control their functions. When USB HID is used, the control functions in the app are "hardwired" to controls in **Max**. Connecting to these apps is as easy as clicking on their **CTR2** controller button and selecting the **ESP32-S3_DEV** device. Mapping can be found in your app's documentation.

Initial Startup

The first order of business is to power up your **Max**. This must be done by plugging it into a USB port on the computer running your radio control app. An optional internal 3.7-volt Lithium Polymer battery can be installed if you plan on plugging **Max** into an iPhone or iPad, so **Max** doesn't draw power from the iOS device. Do not plug **Max** into an unpowered USB hub. Unpowered hubs cannot provide the power the unit requires.

When the unit boots, the flash screen will show **CTR2-Max**, the current operating mode (**Flex** or **MIDI** mode), and the firmware version number. If your unit boots into **Flex** mode, open the **Settings** menu, navigate to the last page, and select **Run MIDI Mode**. The unit will reboot into **MIDI** mode.

Connect to your App

Max's firmware only supports USB MIDI. Generally, you'll need to go to your app's MIDI menu and select **Max** from the list of available devices. **Max** will appear as **ESP32S3_DEV** on most devices. You can find specific information on connecting to different devices in Appendix A.

Configure the MIDI Mapping in your App

Now that you have **Max** connected to your device, open the MIDI mapping menu in your 3rd-party app and check the labels assigned in **Max's Dial** and **Buttons** menus against the mapping in the app. **Max** ships with labels for SmartSDR control in Map 1 and labels for Thetis in Map 2, unless you asked for a different configuration when you ordered your unit. The **Ctrl Map** can be selected in the **Settings** menu on page 1. Neither set of labels match the default mapping in these apps. In fact, Thetis does not come with a MIDI map pre-installed.

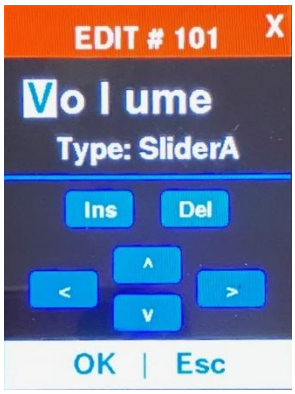
Luckily, most apps have an **Import** feature that will allow you to import a MIDI map into the app. I have maps available for a variety of 3rd-party apps on the CTR2-Max Configuration Files page on Lynovation.com.

Apps use different name extensions for their map files. [*.map*] is used with SmartSDR, SDR-Control, etc. [*.xml*] is used with AetherSDR and SDR-Console, [.json] is used with RimuSDR, and [*.m2c*] is used with Thetis. Instructions for loading these map file can be found app's user manual. I have some information in the Editing 3rd-Party App Maps section below.

To find out how each control in **Max** is mapped in your app, open its map editor and turn the dial control or touch a button on **Max**. Most apps highlight the MIDI command received in their map, along with the radio function mapped to that MIDI command. Your app will provide a list of functions to choose from for each MIDI command.

The last item of business is setting the label on the control in **Max** to match the functions mapped in the app.

Editing Labels in CTR2-Max



There are two ways to edit the labels associated with the controls in **MIDI** mode.

1. Use the simple text editors built into **Max**. These can be found on page 4 in the **Settings** menu. These editors allow you to change each character in the label using the [^} and [v] buttons on the screen. You can change the selected character using [<] and [>] buttons, insert new characters, or delete the current character using the onscreen buttons. When editing a **Dial** setting, you also have the option of changing the MIDI control type for that dial. Touch the **Type:** label to step through the available types. For more than one or two labels, this can be tedious.
2. Connect a dumb terminal to **Max** and edit labels using the terminal's keyboard. Information on doing this can be found in the Editing Labels and Control Types section.

The goal is to match up the label for each control in **MIDI** mode with the actual function controlled by the 3rd-party app.

There are spreadsheets in the CTR2-Dial Configuration Files zip file that provide examples of how **Max** and 3rd-party apps are mapped. You can use these sheets as a basis for setting up your own configuration.

Importing Configurations into CTR2-Max

Max includes Import and Export tools that you can use to load a configuration or save an existing configuration. The CTR2-Max Configuration Files web page includes **CTR2** configuration files that match the 3rd-party app maps that you can import directly into **Max**. These are text files with the extension [*.ctr2*]. The Backing Up and Restoring your Settings section has Information on using these tools.

NOTE: Both Map1 and Map2 are exported and imported using these tools.

Time to Play!

With the basic configuration done, the best way to learn about **MIDI** mode is to use it. Feel free to edit the labels in **Max** and the functions in your 3rd-party app to fit you're operating style.

The following sections provide an in-depth look at the hardware and how **MIDI** mode works with **Max** and your 3rd-party app.

Hardware

Max's hardware, including the sidetone interface board and internal battery options are covered in a separate manual. The hardware manual can be found here.

3.5mm Jacks on Max

CAUTION: The 3.5mm jack(s) on Max are for paddle, key, and PTT inputs only. They are not designed for output. Do not connect these jacks to your radio as this may damage the unit.

Internal Battery Option

Max is designed to support an internal 3.7-volt Lithium Polymer battery. Since **Max** must be connected to a USB port on the computer running the radio control app, the internal battery is only useful if you plan on using **Max** for an extended time with your iPhone or iPad running on batteries. Running **Max** on its own battery increases the time your iPhone or iPad can run on its battery. I do not ship batteries with any **CTR2** unit, primarily due to postal regulations. You can purchase them from many suppliers. The CTR2-Max Hardware manual provides instructions for installing it.

Battery Operation

The ESP32-S3 2.8" display used in **Max** was not designed for ultra-low power operation. The display and the screen touch sensor remain powered up, even when the processor is put into deep-sleep mode. Because of this, the unit draws 35 mA from the USB power or internal battery while powered down. This will discharge a 1100 mA battery in about 30 hours.

The workaround to this problem is to add a battery isolation switch. This switch is mounted on the side of the unit and is in series with the internal battery (but not the USB power).

With the **Battery** switch In the ON position, the internal battery is connected to the display board. In this state, the unit will run on the battery, and the battery will be charged if the unit is plugged into a USB power source.

NOTE: If you power down the unit from the **Settings** menu, and leave the battery switch ON, the unit will discharge the battery if the unit is not connected to USB power.

With the **Battery** in the OFF position, the internal battery is isolated from the unit and the unit will turn completely off when USB power is removed. The battery will not charge and no current is drawn from the battery. With the **Battery** switch OFF you can store **Max** for long periods of time.

How MIDI Mode Works with Your Control App

Before getting into the details of **Max's MIDI** mode, it's essential that you have a basic understanding of what **Max** does, and how it works with your radio control app.

MIDI mode creates a MIDI controller with a touchscreen display. **MIDI** mode supports 18 virtual dial controls and up to 60 virtual buttons. The number of buttons displayed in the **Buttons** menu can be changed on page 3 in the **Settings** menu. Each of these controls are "hardwired" to MIDI command numbers. In other words, the MIDI command # for each control never changes.

You can edit the label displayed for each control and change the type of MIDI command used for wheel and slider controls in the Dial. The label is just a convenient way to let you know what a control is *supposed* to be changing in your 3rd-party app. The app's MIDI mapping determines what it *actually* controled.

> **Example:** MIDI wheel control #100 is always set to the first Dial control in **Max**. This control is labeled "VFO" in **Max's Dial Control** page when it is open. In order for **Max** to change the frequency of your radio, MIDI control #100 must be "mapped" to the VFO frequency control object in your 3rd-party app.
>
> **NOTE:** If the VFO control doesn't work properly in your app, make sure it is set to either **WheelA** or **WheelB**. Most apps required **WheelA**, but some, like Thetis use **WheelB**. If the frequency control operates in reverse, it's probably set to **WheelB**.

Your app will have a method to edit the function assigned to each MIDI control number.

Controls in **Max** can be executed by touching the control on the display, or they can be assigned to the physical knobs and switches on the unit.

When you execute a control in **Max**, a 3-byte MIDI command is sent to your app. The MIDI command includes the MIDI control #, and a modifier. The modifier can be a direction value for a wheel control, a value for a slider control, or an On/Off value for a button control.

IMPORTANT: The default labels for the controls for Map 1 in **CTR2-Max** are setup for Marcus' SmartSDR for iOS/MacOS apps, but they do not match the default settings he ships in SmartSDR. You'll need to edit the mapping in SmartSDR to match **Max's** default settings. Map 2 is set up for use with **Thetis** but you'll need to map Thetis to match the labels in **CTR2-Max**.

You can download the configuration and map files for various radio control apps from the CTR2-Max Configuration Files web page.

> ### *Your App Executes the Wrong Control*
> If you execute a control on **Max** and the app executes the wrong function (or no function at all) you can either edit the label in **Max** to match the function mapped to that MIDI control # in the app, or edit the app's map to execute the function shown by the label in **Max**. It doesn't matter which setting you change, but they must match.

The 18 **Dial** controls are assigned MIDI control #100 through #117. The MIDI control # is shown on the display just above the label. The Dial Editor on page 4 of **Max's Settings** menu allows you to edit the dial's label and MIDI control type.

The 60 button controls are assigned to MIDI control #1 through #60. Buttons are displayed in the Buttons menu. MIDI button numbers are not displayed due to size constraints. You can edit the label for each button in the Button Editor on page 4 in **Max's Setting** menu. If you find that you don't need 60 buttons, you can change the # of buttons displayed in the **Buttons** menu with the # Buttons option on page 1 of the Settings menu.

The table below lists the MIDI button numbers on each page of the Buttons menu, in the order they're displayed.

Page	Buttons		Page	Buttons		Page	Buttons		Page	Buttons		Page	Buttons	
1	1	4	**2**	7	10	**3**	13	16	**4**	19	22	**5**	25	28
	2	5		8	11		14	17		20	23		26	29
	3	6		9	12		15	18		21	24		27	30
Page	**Buttons**		**Page**	**Buttons**		**Page**	**Buttons**		**Page**	**Buttons**		**Page**	**Buttons**	
6	31	34	**7**	37	40	**8**	43	46	**9**	49	52	**10**	55	58
	32	35		38	41		44	47		50	53		56	59
	33	36		39	42		45	48		51	54		57	60*

* Button 60 is shared with Dial 100's CCW action when using the Button MIDI type for the dial.

I have posted a video on YouTube describing the mapping process with **CTR2-Quad**. **Max** uses the same process. You can find it here.

NOTE: You can export **Max's** settings to your computer to back them up, and import them back in as needed. To do this, you must use a terminal program that supports XModem file transfer. Tera Term is the easiest terminal program to use and it supports XModem. Putty and Apple/Linux Terminal programs do not support XModem. More information on importing and exporting settings in **CTR2-Max** can be found in the Backing Up and Restoring Your Settings section.

Multiple Controllers / Two Modes

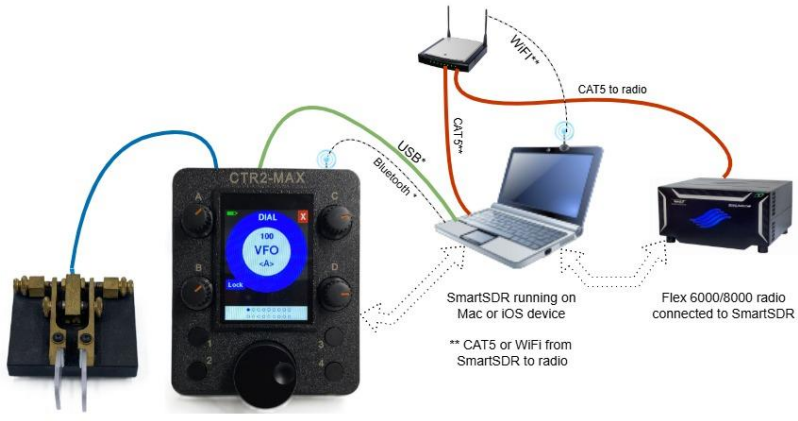
Max can be used in parallel with any of the other **CTR2** controllers. If you have an older **CTR2** unit, you can continue to use it with **Max**. You can use multiple **CTR2** controllers in **Flex** mode, and if your app allows MIDI control, you can have one controller in **Flex** mode and the other in **MIDI** mode. Marcus's apps (SmartSDR, SDR-Control, etc.) allow you to connect multiple MIDI devices to it. Just select the **MIDI**

Controller, **CTR2 Controller**, or **CTR2 Dial Controllers** tools in the **Tools** menu to connect your **CTR2** controller to the app. You can never have too many knobs!

Switching Operating Modes

To switch operating modes, open the **Settings** menu and navigate to the last page and select **Run Flex Mode** or **Run MIDI Mode**, depending on which mode is enabled.

In **MIDI** mode, the controller uses USB MIDI to connect to, and control, the 3rd party app that is controlling the radio.



CTR2-Max
Controlling SmartSDR using MIDI

CTR2-Max's MIDI Mode Features

Max runs on a 2.8" capacitive touchscreen display with an integrated ESP32-S3 processor. The firmware includes the following features for **MIDI** mode:

- Supports USB MIDI and USB HID for browser-based apps
- Up to 60 virtual button controls for modes/bands/filters selection
- 18 virtual dial (wheel/slider) controls for tuning, volume, RIT, XIT, etc.
- Five physical encoders (referred to as **Knobs**)
    - Knobs and switches can be programmed in the Controls menu
    - Selecting **<Dial>** for a knob function in the **Controls** menu allows that knob to be used to control the selected dial function in the **Dial control**
    - Knob push switches can be mapped to two virtual button functions using short and long-press
- Four pushbuttons are provided – each of these can be programmed for any two button functions (using short and long-press) or be dedicated to control PTT.
- VFO Lock control
- Two 3.5mm (1/8") stereo input jacks allow you to use your paddles to control your app's keyer, straight key, or PTT
- A virtual PTT switch
- Four beep volume levels, Off, Low, Medium, and High

- Automatic speed sensitive tuning & control adjustments
- Two maps allow **Max** to be used with separate programs without requiring configuration changes
- Seven color themes and backlight control
- Sleep and power down timers to save battery power
- Power Down button in the **Settings** menu
    - **NOTE**: **Max** will draw 35 mA even when powered down – use the **Battery** disconnect switch to turn off the unit if using an internal battery.
- User programmable labels for each button and dial control
- Backup **CTR2-Max's** settings using the **Import** and **Export Settings** option
- Firmware updating is done through the USB-C port
- Run **MIDI** or **Flex** mode without re-flashing the unit

MIDI Control Types

Before discussing **MIDI** mode's functions, it's important to understand the types of MIDI controls **Max** supports. The table below summarizes each function.

Buttons	Application
Button controls	Buttons are implemented using MIDI Note controls. Each button in **MIDI** mode is assigned to a MIDI Note # starting at 1 and continuing to 60. When you execute a button function the program sends a MIDI Note ON command followed by a MIDI Note OFF command to the app for the assigned button ***when you release the button***.
Wheels & Sliders	**Application**
WheelA	This control is used with *Marcus' iOS/MacOS programs, AetherSDR, RimuSDR, RHR*, *RemoteTx,* and *Online Remote Radio* for frequency control. This control is continuously adjustable and is centered at 64 counts. Frequency up/down control is initiated by sending a modifier value above or below 64 depending on the direction the dial is turned. The offset from 64 counts determines the turning speed.
WheelB	This control is similar to WheelA and is used with *Thetis* and most Windows control programs. This wheel sends 126 counts to decrement the frequency by one step and 1 to increment the frequency by one step. When *Speed Tuning* is enabled the *Dial* or *Knob* is quickly turned the firmware sends 10 commands for each step instead of one to speed up tuning.
WheelB-r	This control is the same as Wheel B except that 1 is sent to decrement frequency and 126 is sent to increment frequency. This control type was added to support the *PI HPSDR* control program.
SliderA	This control is similar to potentiometer. Its range is 0 to 127 counts. Turning the *Dial* control CCW reduces the count until 0 is reached. Turning it clockwise increases the count until 127 is reached.
SliderB	This control is similar to SliderA except that it is used specifically for center tuned controls like RIT and XIT. This control is centered at 64 counts. Decrementing below 64 counts applies a negative offset and

SliderB (cont)	incrementing above 64 counts sends a positive offset. When the value of this control reaches 64 when turning from either direction, **Max** beeps and pauses for 750 milliseconds. This allows you to find zero on the control. **HINT:** Touching the control's label in the center of the *Dial* display when the control is "off zero" resets the control to its zero position (64 counts).
Buttons Wheel Type	This control assigns a pair of MIDI buttons controls to send up and down button controls to the control program using the *Dial* or *Knob*. The lower control # is sent when the control is turned CCW and the upper control # is set when the control is turned CW. This is normally used for frequency or frequency tuning step up/down control. The control pair is determined by the Dial # as shown in the next table.

MIDI Button Pairs for Dial Button Mode

When the **Button** control type is selected for a **Dial** control, the MIDI button pair is determined by the Dial control #. Turning the knob CCW sends the lower MIDI command #, and turning it CW sends the upper MIDI command # from the table below.

Dial Ctrl #	Button Pair	Dial Ctrl #	Button Pair	Dial Ctrl #	Button Pair
100	**60*/61**	**106**	**72/73**	**112**	**84/85**
101	**62/63**	**107**	**74/75**	**113**	**86/87**
102	**64/65**	**108**	**76/77**	**114**	**88/89**
103	**66/67**	**109**	**78/79**	**115**	**90/91**
104	**68/69**	**110**	**80/81**	**116**	**92/93**
105	**70/71**	**111**	**82/83**	**117**	**94/95**

* **Button 60** is shared with the last button in the **Button** menu.

EXAMPLE

To assign a knob control to increment or decrement the frequency tuning step in SmartSDR, do the following:

1. Determine which Dial control you want to use. We'll use #103 in this example.
2. Using the **Edit Dial Setting** option on page 8 of the **Settings** menu, select the fourth button on the first menu (#103). Edit its label as needed, then touch the **Type** option to step to **Button**.

3. Open the **Edit Mapping** menu in the **CTR2 Dial Controller Tool** in SmartSDR
4. Select control 103 on the **Dial control** then turn the dial CCW. MIDI button 56 should be highlighted in the **Edit Mapping** menu in SmartSDR.
5. Select the *Tune Step Previous* function from the function list.
6. Turn the dial CW. MIDI button 57 should highlight in the **Edit Mapping** menu in SmartSDR.
7. Select the *Tune Step (1,20,100,1000)* function from the function list.

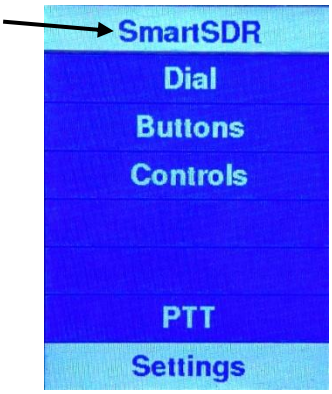
Now, turning the knob for Dial control 103 will increment or decrement the frequency tuning step in SmartSDR.

NOTE: If you use the **Dial Button** mode for **Dial** control 100, you cannot use **Button 60** on the **Button** menu for another function.

Home Page

The user interacts with **Max** using the touch screen to select from a series of menus and control pages. The **Home** page is the default display and is titled **SmartSDR** (this can be changed in the **Settings** menu). From here you can select the various control pages and the **Settings** menu.

The map name is used to show the radio control app the current map is configured for. It can be edited in the Settings menu.



From the **Home** page you can quickly access the **Dial**, **Buttons**, **Controls**, and virtual **PTT** pages or open the **Settings** menu.

Dial Control

The virtual *Dial* control supports 18 MIDI wheel or slider controls. The dial touch pad is the outer ring around the round center label. Swipe your finger around the outer ring to adjust the control.



Each dial control is assigned a MIDI control number starting at 100 and ending at 117. The number of the selected dial control is shown on the top of the center label. This # indicates the control # you need to map in your app's MIDI function menu to execute this control.

The user supplied label for the selected control is shown below the MIDI control #. The label lets you know which function the app is *supposed to* execute for this control. If the label doesn't match the MIDI function mapped in the app, you can edit it to match.

To select another dial control, touch the center of the label or touch the dotted button at the bottom of the display.

Touch the red [**X**] in the top-right corner to return to the **Home** page.

NOTE: Assigning the **<NAV>** function to one of the knob functions in the **Knobs** menu allows you to use that knob to select a dial control on this page.

Dial Graphics

The **Dial's** background depends on the MIDI *control type* selected for the control.

Control types can be set using a dumb terminal program, or you can use the **Edit Dial Setting** on page 4 in the Settings menu.

Wheel Controls

The lead photo above shows a wheel control. Wheels are continuously variable and used for tuning. The graphic on this control is not animated. The MIDI control # (100) is shown on the top line of the center circle. The label in the middle (in this case **VFO-A**) indicates the control function and can be edited by the user. **<A>** at the bottom of the label indicates that this is a *WheelA* control. The same display is used for *WheelB* (**<B>**) and *WheelB-r* (**<B-r>**) controls.

In addition to wheel controls, you can select from two slider control types.

SliderA



SliderA is the most common. The outer ring on the **Dial** control represents a gauge with the level indicated by the filled arc that starts on the bottom left. In the photo to the right the volume is set at 50% and the level gage is 50% full. The MIDI control # (101) is displayed at the top of the center label. Swipe your finger around this ring to change the value.

SliderB



SliderB is used for RIT, XIT, and other +/- offset functions. In this display the offset is represented by an arc that starts at the top center of the display. For negative offsets it grows to the left and for positive offsets it grows to the right from center. The MIDI control # (107) is displayed at the top of the center label.

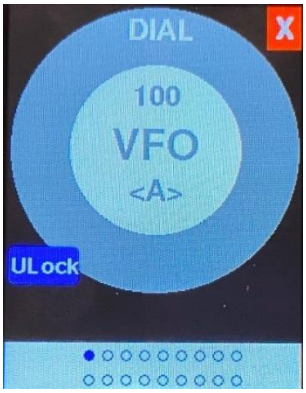
When the RIT/XIT control is offset as shown, touching the center label (**RIT>**) resets the offset back to 0. You can enable/disable RIT or XIT from the **Buttons** menu. You can also assign the enable/disable RIT or XIT buttons to one of the knob switches or pushbuttons.

NOTE: The RIT/XIT display shows the percent of offset. The actual frequency offset is determined by the app you're controlling.

Buttons

You can select buttons instead of wheels to be used with a *Dial* control. These are typically used in Marcus' apps that support button up/down controls for frequency control. Generally, you won't use buttons with the *Dial* but the option is there if you want to try it.

Tuning Lock



You can lock tuning to prevent inadvertent changes simply by touching the **[Lock]** button on a control that is set for a Wheel MIDI type (typically VFO controls). When tuning is locked, the *Dial control* will be "grayed out" as shown here. Adjusting the control, or turning a knob assigned to this control will post a message with instructions on how to reset the lock.

To unlock the control, touch the **[ULock]** button.
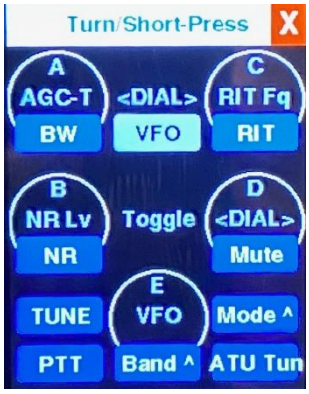
Dial Menu



If a knob has been set to **<NAV>** in the **Knobs** menu, it can be used to change the virtual *Dial control's* function. Otherwise, to change the virtual dial's function you'll need to open the **Dial** menu. To do this, touch the MIDI control# on the *Dial control*. This menu allows you to select from a two-page menu of dial functions. The current dial setting will be highlighted (**VFO** in this example). Touch the bottom button or twist the knob assigned to **<NAV>** to move to the next page.

Touch a button to select that control and return to the virtual dial control, or touch the page's caption to return to the virtual *Dial control* without changing the selection.

Controls Menu



Max supports dual-function knobs and switches. Each knob has a *turn* and a *push-an-turn* function and each switch (knob switches and pushbuttons) have a short-press and a long-press function.

The **Controls** menu provides a graphical representation of the physical controls on **Max**. The function assigned to each knob and switch are displayed.

The caption indicates which mode is displayed. The **Turn/Short-Press** mode is displayed here. Touch the caption or the **Toggle** button to switch to the **P&T/Long-Press** (Press & Turn) mode.

Adjusting any control will bring up a temporary display showing that control's operation.

Labels in the circles show the function of each knob. The switches are displayed as labeled buttons.

The exception is the button under the **<DIAL>** label. This button displays the dial currently selected in the **Dial Control** page. You can edit this setting on this page.

Control States

Each control has two states. For knobs, it's either **Turn** or **Push & Turn**. For knob switches and pushbuttons, it's either **short-press** or **long-press**.

To switch which state is displayed on the **Controls** menu, touch the caption or the **Toggle** button.

NOTE: While the **Controls** menu is displayed, the controls respond based on the selected page. If the **Turn/Short-press** page is displayed, the controls work as normal. If the **P&T/Long-Press** page is displayed, the control state is inverted… i.e. turning a knob executes its **P&T** function, and pressing and turning a knob executes its **Turn** function. The same is true for the switches. Short-pressing a switch executes its **Long-Press** function, and long-pressing a switch executes its **Short-Press** function.

Editing Knob and Button Assignments

You can easily edit the control assigned to each knob or switch.



To do this, simply touch the control you want to change. The control's background will turn red and **DELETE** will be displayed in the center of the display. This example shows **Knob A** selected for editing.

To change the assignment, simply turn any of the knobs on the unit to change the control function to what you want. Touch the control again to accept the change. **Edit** mode will time out after 6-seconds and revert to its original setting.

PTT Assignment

If a pushbutton is assigned to PTT (pushbutton 2 in this example), you cannot change or delete its assignment on this page. This must be done on page 3 in the **Settings** menu.

Deleting an Assignment

To delete the assignment on the highlighted control, touch **DELETE**. The label on a knob will go to **<DIAL>**, meaning it will control the selected **Dial Control**. Buttons will go blank, indicating this control is no longer active. To re-assign a function to this control, touch it again to enter **Edit** mode, and turn any knob.

NOTE: The **DELETE** function is not available for the **<DIAL>** function.

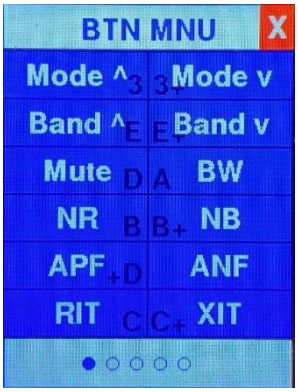
<DIAL> Function

The **<DIAL>** function can be selected for any knob. When a knob is set to **<DIAL>**, it will control whatever dial function is selected in the **Dial Control**, regardless of the page that is being displayed. In this example, **Knob D** is set to **<DIAL>.**

<NAV> Function

The **<NAV>** function can also be assigned to any knob. When a knob is set to **<NAV>** it acts as a navigation control, and can be used to switch pages in the **Settings** menu or select another dial control in the **Dial Control** page.

Buttons Menu



Open the **Buttons** menu from the **Home** menu to select from 60 button actions. Touching a button immediately sends the MIDI control for that action ***when you release the button***.

There are five pages of buttons with 12 buttons on each page. If a knob is assigned to the **<NAV>** function, turning it will change the **Button** page, otherwise, touching one of the dots on the bottom button will open that page.

Press the red [**X**] in the top-right corner to return to the **Home** page.

Assigning a Knob Switch or Pushbutton to a Button Function

The **Controls** menu provides the easiest way to assign knob switches or pushbuttons to button functions. You and also change these assignments in the **Buttons** menu.

Knob switches and pushbuttons can be set to two actions: a short-press (less than 1 second) and a long-press (greater than 1 second).

Switch Short-Press Action

To assign a short-press button action to a knob's push switch or pushbutton, do the following:

- Open the **Buttons** menu and navigate to the page with the action you want to assign
- Touch and Hold the knob switch or pushbutton you want to use
- Touch and Release the action button you want the switch to execute

Once you have assigned a function to a knob switch or pushbutton, the knob's letter or pushbutton's # will appear in that button's label. That action will be sent to the control app anytime you press that knob or pushbutton.

Switch Long-Press Action

To assign a long-press button action to a knob's push switch or pushbutton, follow the steps above for setting the short-press action then while still holding the knob or pushbutton down, touch the desired function a second time. **Buttons** set to long-press actions will have a **+** appended to the knob or pushbutton # displayed on the button.

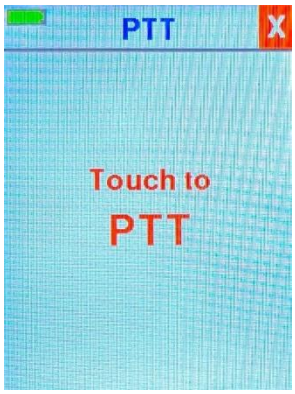
The photo above shows the **Buttons** menu**.** *Knob B's* short-press action is set to the **NR** action and its long-press action is set to the **NB** action. *Knob C's* short-press action is set to toggle **RIT** and its long-press action is set to toggle **XIT**.

To clear a knob switch or pushbutton's assignment, hold the knob switch or pushbutton down and touch the selected action again.

NOTE: You must clear both switch actions from the button before you can assign it to another knob switch or pushbutton.

NOTE: Changing the PTT assignment for any pushbutton in the **Settings** menu automatically cancels all switch actions assigned to all pushbuttons.

PTT Page



The **PTT** page offers a virtual PTT button. Simply touch the display to key your radio and lift your finger to unkey. The MIDI control # sent to the app from this page is determined by the **PTT->MIDI Ctrl #** selected on page 3 in the **Settings** menu.

If your radio control app supports latching PTT, map that function in your app to the MIDI control # selected for **PTT->MIDI** in the **Settings** menu. In latching mode, touching and releasing the display keys PTT. Touch and release the display again to unkey PTT.

NOTE: If your app supports latched PTT, **Max's** display will not remain red when you lift your finger after PTT is keyed.

The default MIDI control # for PTT is 99. You can choose from 96, 97, 98, or 99 in the **Settings** menu under **PTT Ctrl**.

NOTE: Controls 20, 21, 30, and 31 are also available in **PTT->MIDI** to allow compatibility with apps such as RHR and RemoteTx that are configured for **CTR2-MIDI**. Selecting one of these controls overrides the **Button** menu action associated with it.

NOTE: If your radio keys up when you press the PTT display but your microphone doesn't produce power output, check the microphone settings in the app.

Using a Hardware PTT Switch and Straight Key

You can use a straight key (or external keyer) and/or an external PTT switch connected to the **Key/PTT** jack on **Max**.

1. Connect your straight key (or external keyer) to the TIP and SHIELD (ground) terminals on a 3.5mm (1/8") stereo plug. Connect your PTT switch between the RING and SHIELD on the same plug.
2. Plug the stereo plug into the *Key/PTT* jack.
3. In the **Settings** menu, set **PTT->MIDI: 99** (or **31** if using CTR2-Max with Remote Ham Radio)

Once configured, pressing the straight key will send MIDI command 98 (or 30) and pressing the PTT switch will send the MIDI control defined in **PTT->MIDI** to your app to key PTT. Make sure these control # are mapped to straight key and MOX (or PTT) in your app.

CAUTION: The 3.5mm jack(s) on Max are for paddle, key, and PTT inputs only. They are not designed for output. Do not connect these jacks to your radio as this may damage Max.

Settings Menu



To enter the **Settings** menu, press the **Settings** button at the bottom of the **Home** page.

The **Settings** menu is divided up into 5 pages with six buttons on each page. You must press a button twice to change its setting.

Press the page caption or the red [**X**] to return to the **Home** page**.**

To change menu page, touch one of the dots on the bottom button. If a knob has been set to **<NAV>** in the **Knobs** menu, rotating that knob will also change the page.

The table below summarizes each setting option.

Page	Btn	Function	Description
1	1	**Beep Mode**	Changes the volume of control beeps: Off, Low, Medium, and High
	2	**Speed Tune**	When enabled, touch and hold the outer ring on the **Dial** for > 1 second to enter speed tuning mode. Valid for *WheelA* and *WheelB*.
	3	**Dial Sens**	Changes the sensitivity of the virtual dial. Range: Low, Medium, High. Default is Medium.
	4	**Ctrl Map**	Selects control Map 1 or 2. Each map can be configured for a specific radio control app. Use a different color theme for each map. Default is *Map 1*.
	5	**Info**	Displays the firmware version #, you call, registration key, and the unit's unique Bluetooth ID
	6	**Future**	

2	1	**Backlight**	Adjusts the backlight. Range: Low, Medium1, Medium2, and High. Lower settings reduce current draw on the unit.
	2	**Theme**	Select the color theme for the selected Map. Options include Dark, Light, Blue, Orange, Green, Red, and Violet. Default is Blue for Map 1 and Violet for Map 2.
	3	**Rotate Screen**	Rotates the screen 180 degrees. This allows the screen to be aligned for various mounting configurations
	4	**Future**	
	5	**Future**	
	6	**Battery Icon**	Toggles the battery icon on/off. The internal battery is useful when operating with a battery-operated device such a laptop, iPhone, or iPad. It powers Max without draining the device's battery.
3	1	**Pdl->MIDI**	Selects the MIDI control numbers mapped to the *Paddle Input* jack. Default is 96/97 and these are usually mapped to the *Left* and *Right* paddle controls in the app. Option 98/99 can be selected and mapped to the Straight Key and PTT options in the app if you want to use your paddle for these functions (**Max** provides a separate jack for key and PTT inputs). For compatibility with Remote Ham Radio and RemoteTx's mapping for **CTR2-MIDI** you can also select 20/21 and 30/31 pairing. **NOTE:** By default, controls 20/21 and 30/31 are mapped to buttons. Remove these mappings in your app when controlling them with the paddle input.
	2	**PTT->MIDI**	Selects which MIDI control # is mapped to the virtual **PTT** button page and the pushbutton switch selected in the **SWx set to PTT** option below. The default of 99 is typically mapped to PTT or MOX in the app. Options are 96, 97, 98, and 99. MIDI controls 20, 21, 30, and 31 are also available for compatibility with Remote Ham Radio and RemoteTx's **CTR2-MIDI** mapping. To use an external PTT switch, wire it to the RING and SHIELD of the **Key/PTT** input jack, select **Pdl->MIDI: 96/97** above, **PTT->MIDI: 99** here, and **SW2 set to PTT**. If using Remote Ham Radio or RemoteTx, select set **PTT Ctrl** to **31**. See Using a Hardware PTT Switch and Straight key.
	3	**SWx set to PTT**	This option allows you to select which pushbutton is assigned to PTT. To use an external PTT switch on the 5-knob **Max**, the **Key/PTT** input jack, this must be set to **SW2**. On the one-knob **Max**, only **SW1** is available for PTT. On the two-knob **Max** there are no switches available for PTT.
	4	**PDLS & KEY**	This option swaps the TIP and RING on either the *Paddle* input jack and/or the *Key/PTT* input jack. Steps through all possible combinations.
	5	**# Knobs**	Allows you to select the # of knob available on your **Max** unit. The current hardware is fixed at 5 knobs.
	6	**# Buttons**	Changes the number of buttons displayed in the **Buttons** menu. This allows you to limit the # of buttons to the buttons you actually use. Range is 6 to 60.

4	1	**Edit Map Name**	Allows you to edit the current map's name. This name is displayed in the caption of the **Home** page. You can enter the radio control app's name here to identify the app this map is used with.
	2	**Edit Dial Setting**	Opens the Dial control label and MIDI control type editor. See Editing CTR2-Max's Labels and Control Types for more information on dial labels and MIDI control #s.
	3	**Edit Btn Label**	Opens the Button label editor. See Editing CTR2-Max's Labels and Control Types for more information on button labels. Opens the Button label editor. See Editing CTR2-Max's Labels and Control Types for more information on button labels.
	4	**Future**	
	5	**Export Settings**	Backup **CTR2-Max's** configuration by exporting them to a file on your computer. To do this, in Tera Term select *File->Transfer->XMODEM->Receive* then enter the name of the file you wish to save. Next, touch **Export Settings** to start the transfer. Once this file has been saved you can recover from a unintended memory erase when flashing new firmware by using the **Import Settings** option above. **NOTE:** You can also use the import/export options to create setting files for different radio control apps.
	6	**Import Settings**	Import settings from a previously exported setting file using XModem. In Tera Term select *File->Transfer->XMODEM->Send* and select a previously saved configuration file on your computer. Next, touch **Import Settings** to start the transfer.
5	1	**Sleep**	Sets a timer that turns off the display backlight after no activity for the set number of minutes to reduce current draw. Sleep mode does not close USB or Bluetooth connections. Range: Never, 1, 5, 10, 30, and 60 minutes.
	2	**Pwr Off**	Sets a timer to turn off the unit after no activity for the set number of minutes. Pwr Off mode reduces current draw to 14 microamps and closes USB and Bluetooth connections. Range: Never, 10, 30, 60, 120, and 180 minutes.
	3	**Power Down**	Immediately closes the Bluetooth and USB connection and puts the unit to sleep, reducing current draw to 14 microamps. Touch the display to wake up the unit. **NOTE:** If you are powering your **CTR2** controller with a 3.7-volt battery you will need to press and hold the **M5** button on the orange ring for > 10 seconds to wake the controller up from sleep mode.
	4	**Run Flex Mode**	Shuts down **MIDI** mode and restarts the unit into **Flex** mode
	5	**Restore to Start Up**	This option allows you to reset your settings to what they were when you first booted **CTR2-Max**, *if you didn't choose to reset your call and registration key in* ***Reset to Factory***. This is handy in case you want to "roll back" changes you made during the current session.

5	6	**Demo Mode:** **Edit Call & Key** ---------- **Normal Mode:** **Reset to Factory**	**In Demo Mode:** This option allows you to enter your call and registration key without using a terminal program. To enter text, use the [^] and [v] buttons to select a character and [<] and [>] to move to the next character. Press **OK** to accept. You can also connect a terminal program and use the terminal keyboard to enter your call and registration key. **In Normal Mode:** This option resets **Max's** configuration back to the original factory settings. **NOTE:** You will be given the option to reset your call sign and firmware registration key. **NOTE:** If you reset the configuration by mistake, and didn't choose to reset your call and registration key, execute the **Restore to Start Up** option to restore your settings before powering the unit down.

Label and Settings Maps

MIDI mode supports two maps for control settings. Select the map you want to use on page 1 of the **Settings** menu. These maps store control labels and MIDI control types for the dial controls. They do not change the MIDI command #s assigned to the controls in **Max**. You must *edit your apps' map* to change the *function* the app executes for MIDI command #'s that **Max** sends.

The name of each map is displayed in the caption of the **Home** page. You can edit this name to reflect which radio control app **Max** is configured to control. Map names are limited to 8 characters.

You can edit the labels and control types in each map to fit your needs. For instance, you can configure Map 1 to contain the labels and control types for SmartSDR and Map 2 for labels and control types for SDR-Control. Simply switch maps to match the app you're using.

HINT: I recommend using different color themes on each map so it's easy to determine which map you're using.

Editing Labels and Control Types

Two methods are provided for you to edit the control labels. The first is to connect a dumb terminal to **Max** and use the keyboard. Just select which control you want to edit (select the dial on the **Dial** menu or touch a button on the **Button** menu) to display that controls label options.

The second method is to use the built-in editors that allow you to edit the properties associated with **Max's** *Dial* and *Button* controls without using a terminal.

The Dial editor allows you to edit the label and the MIDI control type assigned to each *Dial control*.

The Button editor allows you to edit the label associated with each *Button* control.

These editors can be found on page 4 in the **Settings** menu. For more information on how **Max** controls your radio control app, see How CTR2-Max Works with Your Control App.

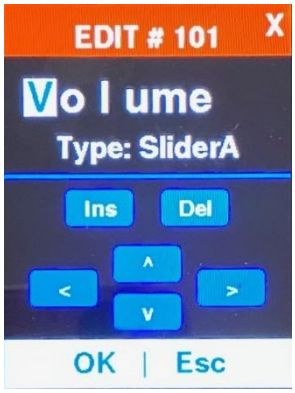
Dial Editor



When you select the **Edit Dial Settings** option the **EDIT DIAL** menu opens. This menu is similar to the **Dial menu** in that it has 2 pages of dial controls to choose from. The currently selected dial control will be highlighted.

Touch the open dot on the bottom button, or, if a knob has been assigned to **<NAV>** in the **Knobs** menu, rotate that knob to change pages.

Touch the label of the dial control you want to edit. The edit page will open. Touch the **EDIT DIAL** caption or red [**X**] to cancel this mode and exit back to the **Settings** menu.



The MIDI control # for the selected dial control will be displayed in the caption of the edit page. MIDI control # is hardcoded and cannot be edited.
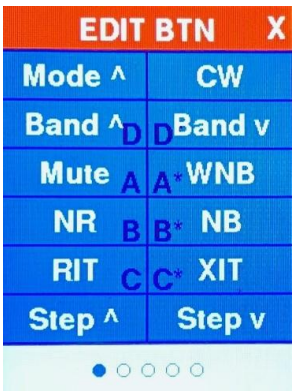
The edit cursor will highlight the character to be edited ("V" in this example). Use the [^] and [v] buttons to change the character. Touch [<] or [>] to move the edit cursor to the next position. You can also touch another character in the label to move the edit cursor to that position. There are eight characters available for labels.

Touch **Ins** to insert a character at the cursor position and **Del** to delete the character at that position.

To change the MIDI control type associated with this control, touch **Type:** on the display. The MIDI control type will step to the next control type. Refer to the MIDI Control Types section for a description of each control type.

When you're done editing, touch **OK**. To exit without saving your changes, touch **Esc** or the red [**X**].

Button Editor

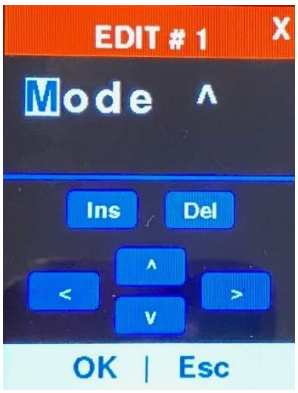


When you touch the **Edit Button Labels** the **EDIT BTN** menu opens. This menu has 5 pages of buttons with 12 buttons on each page, similar to the **Buttons** menu.

Touch an open dot on the bottom button, or, if a knob has been assigned to **<NAV>** in the **Knobs** menu, rotate that knob to change pages.

Touch the button you want to edit. Touch the caption or the red [**X**] to exit.

Once you select a button the **Button Editor** opens. The MIDI control # is shown in the caption – it cannot be changed.



The **Button Editor** has the same features as the **Dial Editor** but it doesn't have a MIDI **Type** setting. This is because buttons always send MIDI button On/Off commands.

Editing With a Terminal

Onscreen editors are useful for small editing changes but if you have several controls to edit, it's easier to use a dumb terminal.

Start your terminal program and connect it to the virtual USB serial port assigned to **Max** by your operating system.

When the terminal first starts you will be presented with a blank screen because the terminal server is waiting for you to wake it up. This is done by pressing any key (usually [Space]) or by selecting a control on the display.

NOTE: Dial controls without labels will not show up on the **Dial Control** unless a terminal is connected to **CTR2-Max**. To add a label to a blank control, select it in the **Dial menu**.

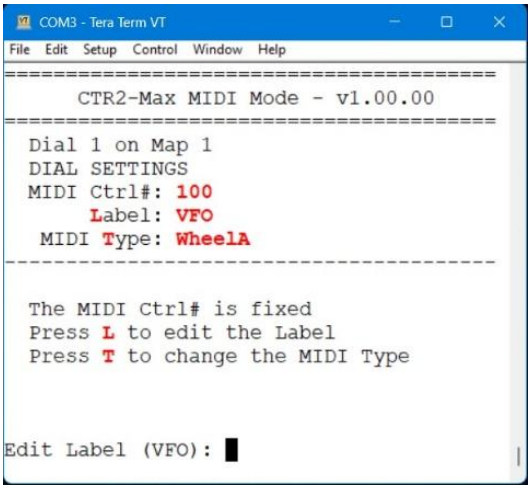


The header of each page contains the firmware version #.

To edit a control, simply select it on **Max**. The terminal page will update with that control's settings. In the screenshot at the right, **Max** is on page 1 of the *Button menu*. The first button on the selected page is always displayed first. To edit the label on this button, press the **L** key on the terminal. The **Edit Label:** line will appear with the current label in parenthesis. Enter the new label at the prompt.

Wheel and slider controls on the **Dial** page include additional settings.

Dial controls are assigned to MIDI control #s 100 through 117. You cannot change these.



Press **T** to step through the MIDI control type assigned to the control or press **L** to edit the label.

Once you have **Max's** map configured the way you want, edit the map in your radio control app to match.

You can export your settings to your computer so you have a backup copy. This process will be described next. You can also import settings from other configurations. You can find configurations on the Lynovation web site on the CTR2-Max Configuration Files page.

Backing Up and Restoring your Settings



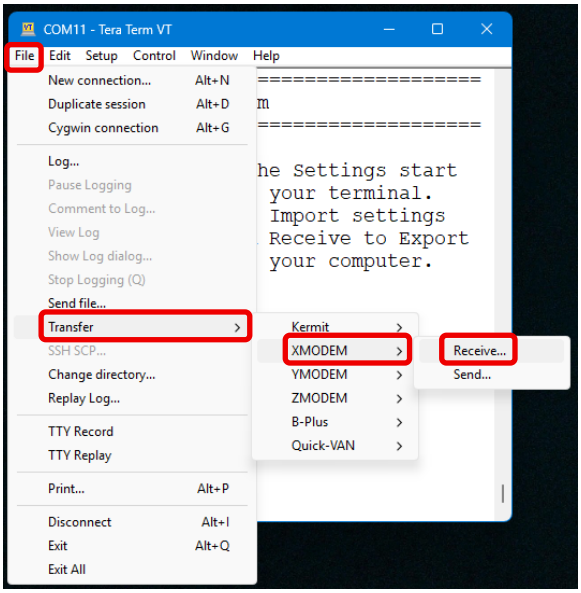
Once you have your **CTR2-Max** setup the way you want it, go to page 4 on the Settings menu and **Export** your settings to back up the file on your computer.

Max uses the XModem file transfer protocol to transport settings to and from backup files on your computer. XModem is an older protocol and not many terminal programs support it nowadays. Tera Term does and I'll use it in the document. Putty does not. To get XModem functionality with Putty you need to download and install ExtraPutty. This program is a branch off of Putty. You can download it from here.

Before you can use the **Import Settings** option you must export the settings from **Max** first to save its configuration files. If you want to just return to factory settings, select the **Reset to FACTORY** option on the last page of the **Settings** menu.

Export Settings

To export the current settings from **Max**, connect Tera Term to **Max** then open its XModem transfer utility. This can be found at *Files->Transfer->XMODEM->Receive…* option shown here.



When you select this option, Tera Term will ask you to enter the file name for the file you want to create. Navigate to the folder you want to save it in then give it a name. *I prefer to use the .**ctr2** extension but you can use any extension you want*.

NOTE: If you don't enter an extension, your file will be created without one.

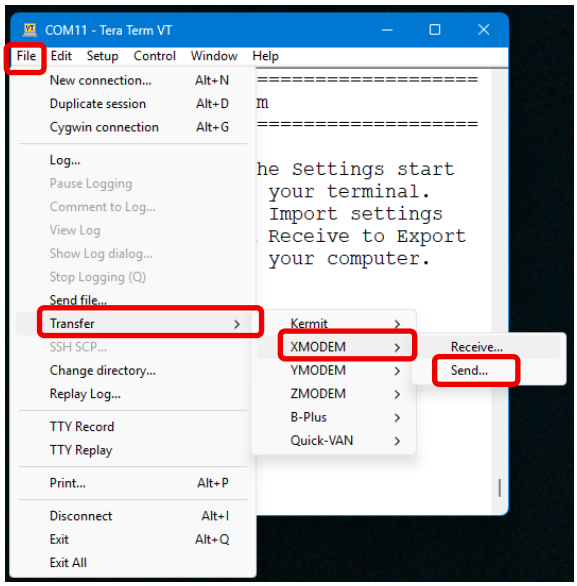
When you click OK, Tera Term will open the XModem transfer window and wait for the transfer to start. At this point, touch the **Export Settings** option in the **Settings** menu.

You can select whether to send just the basic configuration of you unit, or a full system backup that includes all of the radio and frequency information (including your WiFi credentials and radio address). <u>If you plan on sharing your configuration with others, I recommend saving just the basic configuration.</u>

The transfer should complete in a second or two and you should have a new file in your backup folder. You can view this file with a text editor. I don't recommend editing the values in this file because invalid values may cause problems with **Max's** boot process.

Import Settings

Now that you have a backup file of your settings (or have downloaded a configuration file from the Lynovation web site) you can import it using the **Import Settings** option.

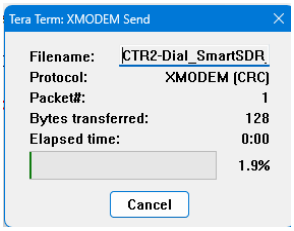


To start an import, touch the **Import Setting** button in the **Settings** menu. Next, in Tera Term, navigate to the *File->Transfer->XModem* and select the *Send* option. Tera Term will ask you to select a file. Select one of the files you previously exported, or a file from the CTR2-Dial Configuration File zip file. Tera Term will then open the file transfer window and transfer the file.

HINT: You can use the export and import settings functions to save and reload multiple versions of your settings. Perhaps you use **Max** on more than two apps and have each app configured differently. You can export the settings for each of your apps then import them back in when you want to change apps. This is also handy if you have more than one **CTR2** controller, or if you want to share your configuration with other users.

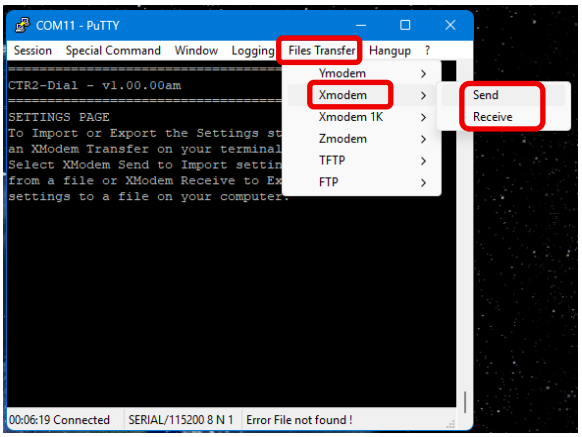
NOTE: Max always saves **Map1** and **Map2** to the export file.

NOTE: If the file transfer stalls (as shown here), cancel the transfer in Tera Term and run it again. This seems to be a random bug in Tera Term as it doesn't happen all the time. The import function in **CTR2-Max** will timeout after 30 seconds if the transfer doesn't complete.



Using ExtraPutty

If you use the XModem transfer option in ExtraPutty as shown here, use the *Xmodem* option, not the *Xmodem 1K* option.



Editing 3rd-Party App Maps

The following sections describe how to map several apps that you may use. It's not an inclusive list of all the apps that support MIDI control. These instructions are included in the configuration files available on the Lynovation web site under Download->CTR2-Max Config Files.

NOTE: Max is compatible with the original **CTR2-Dial** configuration file format.

Before editing the settings in **Max**, you may want to export the settings to your computer in case you need to restore them in the future.

Remote Ham Radio Maps

The current release of Remote Ham Radio does not include mapping for **Max**. It is optimized for **CTR2-MIDI**. The instructions below allow you to change the map in **Max** to fit the current MIDI mapping in RHR.

In the RHR console, open the **Tools** menu and select **CTR2-MIDI**.

Connect a terminal to **Max** and change the labels on the following controls:

Keying Input

Navigate to page 3 in the **Settings** menu and change the **Pdl->MIDI** to **20/21** to use your paddles to control the keyer in RHR. Change **Pdl->MIDI** to **30/31** to use the paddle input jack to control the straight key and PTT functions instead.

Dial Controls

The **Dial** labels should be edited in a terminal or with the **Edit Dial Setting** option in the **Settings** menu to match the function assigned in RHR.

DIAL ACTIONS		
Dial #	**MIDI Control #**	**Action Taken**
1	100	Change VFO
2	101	Slice Volume
3	102	NR Level
4	103	NB Level
5	104	VOX Level
7	106	Keyer Speed

The other **Dial** controls are not currently used. To avoid confusion set their labels to a single space. This will hide them on the *Dial control* page (but not in the **Dial** menu) until you connect a terminal to **Max**.

Button Controls

The number of buttons available in **Max** greatly exceeds those available on the **CTR2-MIDI**. This means that there will be a lot of holes in the **Buttons** menu in **Max** when set to match the current mapping used in RHR for **CTR2-MIDI**.

BUTTON ACTIONS			
Button and MIDI Ctrl #	**Original *MIDI* Button**	**Action Taken**	**Dial Label**
1	BTN1	CW Mode	CW
11	BTN1 (long-press)	Toggle NR	NR
2	BTN2	SSB Mode	SSB
Button and MIDI Ctrl #	**Original *MIDI* Button**	**Action Taken**	**Dial Label**
12	BTN2 (long-press)	Toggle NB	NB
3	BTN3	FT8	FT8
13	BTN3 (long-press)	FT4	FT4
4	BTN4	Toggle PTT	PTT
14	BTN4 (long-press)	Toggle VOX	VOX
5	BTN5	Band Down	Band v
6	BTN6	Band Up	Band ^

All other button labels on **CTR2-Max** can be changed to a single space.

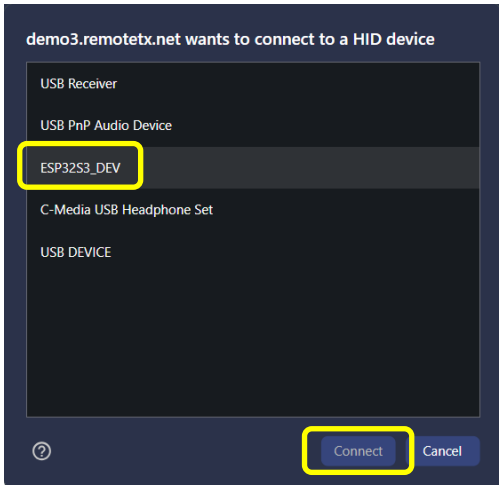
To hide the buttons not used in the **Buttons** menu, go to page 3 of the **Settings** menu and change **# Buttons:** to **18**.

RemoteTx Mapping

To use **Max** with RemoteTx, click **VFO** in **RemoteTx's** *Radio* page, before you connect to the radio.



A list of USB devices on your computer will open. Select the **ESP32S3_DEV** device, then press **Connect**. The **VFO** button should turn red, indicating that the **Max** is connected to this browser session.



To use **Max** with **RemoteTx**, you'll want to edit the labels and knob control types to match the HID controls **RemoteTx** is programmed for. A spreadsheet showing the required mapping can be found at

CTR2-Max Config Files on Lynovation.com. Download the zip file and navigate to the **RemoteTx** folder. This folder also contains the configuration file, *CTR2-Dial_RemoteTx.ctr2*, that you can import into **Max** to change all of the settings for **RemoteTx**.

PTT Input

RemoteTx only supports PTT control. It does not currently support a CW keyer. Plug a straight key or PTT switch into the 3.5mm **Paddle** jack on **Max**. Wire your switch to TIP or RING and SHIELD of a 3.5mm stereo plug.

NOTE: RemoteTx is looking for MIDI control 20 or 21 for PTT. To set your unit up for this, open the **Settings** menu and navigate to page 3. Touch the **Pdl->MIDI** button until it shows **Pdl->MIDI: 20/21**. If you've wired your switch to the TIP and SHIELD of the paddle plug, touch the **PTT->MIDI** button until it shows **PTT->MIDI: 20**. If you wired your switch to the RING and SHIELD of the paddle plug, select **PTT->MIDI: 21**.

iOS and MacOS App Maps

Maps for Marcus' apps are available from the CTR2 group or from the CTR2-Max Config Files page on Lynovation.com.

To change the default map that comes with you iOS/MacOS app, first, make sure your app has permission to use your iCloud drive.

Load the app map you want to use into your iCloud folder. Next, open the **CTR2 Dial Controller** object in the app's **Tools** menu then go to the **Edit Mapping** page. On this page, click or touch the **File** icon at the top of the page then select *Import Mapping from iCloud Drive*. This will open a list of files available on your iCloud drive. Select the appropriate map from the list.

<<< IMPORTANT >>>
Once you have loaded the map you must change one of the control assignments before the map will be saved to your app. To do this, select one of the controls, change it to something else, then change it back to its original setting. If you don't do this the map will revert back to the original map.

The default configuration for **Max's Map 1** is setup for SmartSDR for iOS/MacOS, *but it does not match the default map shipped with SmartSDR*. You will need edit this map manually or import the correct map from the CTR2-Max Config Files page for your app. If you use other apps from Marcus (SDR-Control, FT-Control, TS-Control, or K4-Control) you'll also need to import the correct map to your app *and import the correct **Max** configuration file, or edit the labels in **Max** manually*. I have a video on YouTube explaining the mapping process.

Using Paddles with the App's Keyer

Marcus' iOS/MacOS apps include a keyer that you can control with the paddles connected to **Max**.

Follow these steps to configure the app and **Max** for paddle keying:

- In the app, map **Button 96** to **Trigger CW Left Paddle** and **Button 97** to **Trigger CW Right Panel**. You can swap this mapping if your paddles are wired backwards.
- Plug your paddles into the **Paddle In** jack on **Max**
- Set the radio's mode to CW
- In SmartSDR, press the **View** menu in the Panadapter display and select **CWX Panel**. You can adjust speed and other setting here.
- In SDR-Control, select the **CW Keyer** option in the **Tools** menu to configure the keyer. DO NOT enable the MIDI device in this panel, this is done in the **CTR2 Dial Controller Tool** panel.
- Pressing either paddle will now activate the keyer and key the radio.

Using a Straight Key or an External PTT Switch

Straight Key Wiring

If you want to use a straight key with **Max**, map **Button 98** to **Trigger CW Straight Key** in the app then wire your straight key to the TIP and SHIELD of a 3.5mm (1/8") stereo plug and insert it into the **Key/PTT** jack on **Max**. Do not use a mono jack.

PTT Switch Wiring

If you want to use an external PTT switch with **Max**, map **Button 99** to **PTT Push** in the app and wire your PTT switch to the RING and SHIELD of a 3.5mm (1/8") stereo plug and insert it into the **Key/PTT** jack on **Max**.

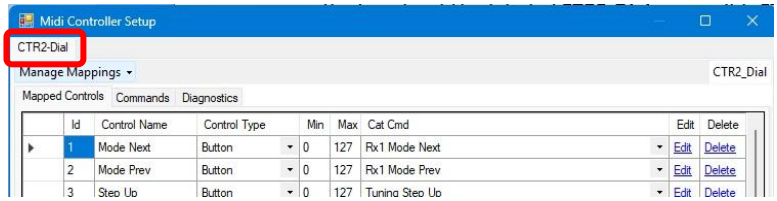
NOTE: You can wire your straight key and external PTT switch to the same stereo plug.

CAUTION: The 3.5mm jack(s) on all CTR2 units are for paddle, key, and PTT inputs only. They are not designed for output. Do not connect these jacks to your radio as this may damage the unit.

Thetis Maps

Thetis is a little different than RHR and iOS/MacOS apps in that it uses **Max's WheelB** control instead of **WheelA** for frequency tuning. **Map 2** is pre-configured with **WheelB** for tuning. You can select **Ctrl Map: 2** on page 2 in the **Settings** menu.

With the **Setup -> Setup -> Serial/Network/Midi CAT** window open in Thetis, select the **MIDI** tab and click the **Configure MIDI** button. The program will find and initialize **CTR2-Max** using USB.

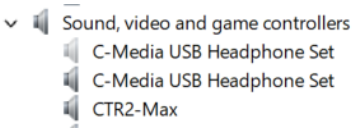


Windows doesn't support Bluetooth LE, so you must use USB with Thetis.

NOTE: If the program doesn't initialize **Max**, check to make sure you are using a USB-C data cable, not a USB-C charge cable.

It's a Software Device

If you can't get Thetis to recognize **Max**, verify that **CTR2-Max** appears in the **Device Manager** under **Sound, Video, and game controllers**. Try right clicking on the entry and uninstall it. Cycle the power on your PC and try connecting **CTR2-Max** again.

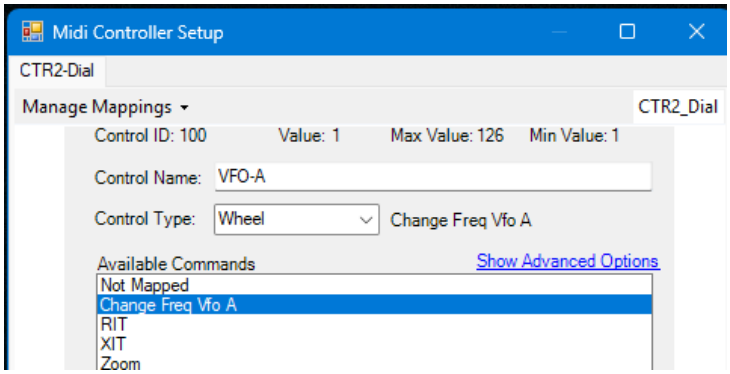


Mapping Controls

You can download a default Thetis map from the CTR2-Max Config Files web site. Once you have unzipped the Thetis file go to the **Midi Controller Setup** page in the app select **Manage Mappings -> Import Mappings** and select the Thetis map you unzipped.

To manually map your **Max** to Thetis, or change the default mapping, use the following process:

Touch a button or change a *dial* or *knob* setting on **Max** and the window at the left will open.

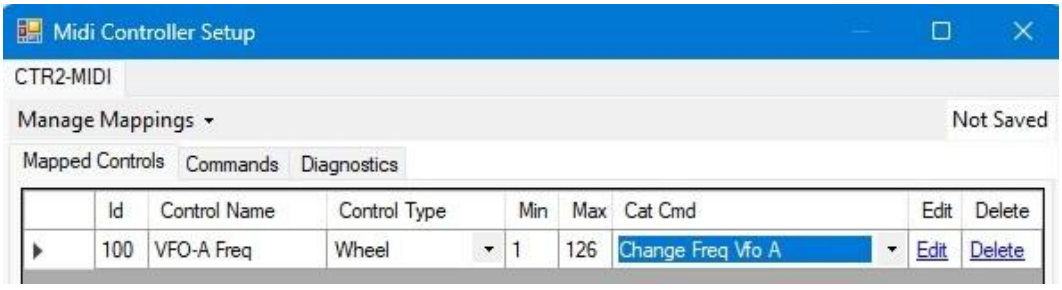


In this example, I changed the dial control when it was set to the first control (VFO-A, MIDI control 100).

NOTE: When mapping *dial controls* you need to run the dial fully CCW until you get to 1 and then fully CW until you get to 127 to set the Min and Max range values.

Next, give the control a name. I'll call it **VFO-A Freq** and select **Change Freq VFO A** from the **Available Commands** list.

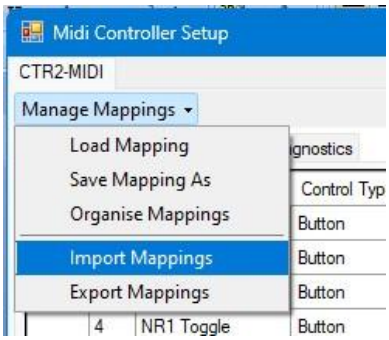
Click the **Done** and then the **Save** button to return to the **Manage Mappings** page.



Now follow the same procedure to the other *dials*, *knobs*, and *buttons* in **Max**.

NOTE: When using the Thetis **Wheel** control type, you must set the encoder on **Max** to **WheelB**.

When you're finished mapping controls make sure to click the **Save** button in the **Midi Controller Setup** page and **Apply** before you leave the **Setup** page so these functions will be restored the next time you run Thetis.



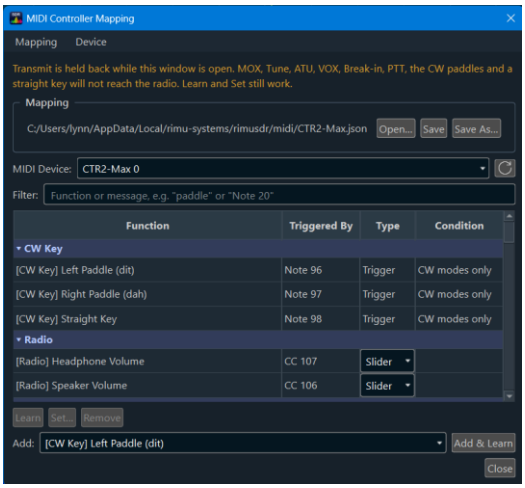
Managing Thetis Maps

You can save or load your map setting from the Thetis database using the **Load Mapping** and **Save Mapping As** options on the **Manage Mappings** tab. To load a map from others, use the **Import Mappings** option. To save your map to share with others, use the **Export Mappings** option.

NOTE: You need to open the **Configure MIDI** page in Thetis every time you start it in order for it to find **CTR2-Max.**

RimuSDR Maps

RimuSDR is a new Flex radio control app designed for Linux. It also has limited functionality for Windows PCs.

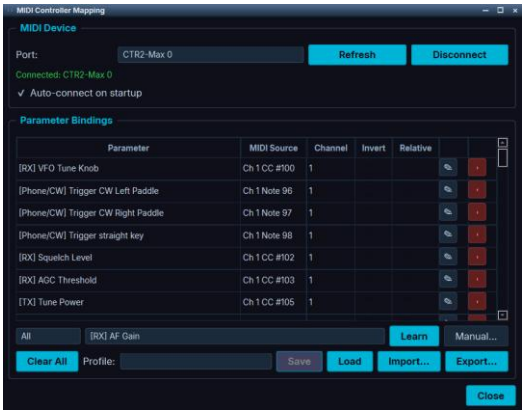


The CTR2-Max Config Files page contains a **RimuSDR** file set. Included is a pre-defined .JSON file that you can import directly into **RimuSDR** using the **Settings -> MIDI Controller…** menu option. Click the **Open…** button to load the .JSON file.

A **CTR2-Max_RimuSDR_MIDI_Map_v1.xx.ctr2** file is also available that you can import directly into **Max** using the **Import Settings** option in **Max's Settings** menu.

AetherSDR Maps

AetherSDR is another newcomer to the radio control scene. It supports several radios including Flex and Hermes Light. native Windows, Apple, and Linux builds.



To access the MIDI map in **AetherSDR**, select **Settings -> MIDI Mapping…**

Max should be automatically detected and connected when you start the program. If it's not, click the **Refresh** button on this page and select it from the list.

To assign a MIDI control to one of the functions in AetherSDR, simply select the function from the dropdown list then click the **Learn** button. Next, execute the knob or switch on **Max** that you want to assign to that function. The function and MIDI assignment will be added to the list of parameter bindings.

You can **Load**, **Import**, and **Export** MIDI configurations.

AetherSDR MIDI setting files are [.XLM] files and they are stored in the C:\Users\{user name}\AppData\Local\AetherSDR\midi folder in Windows. Consult the app's documentation for file locations for Apple and Linux distributions.

New CTR2-Max Features in AetherSDR

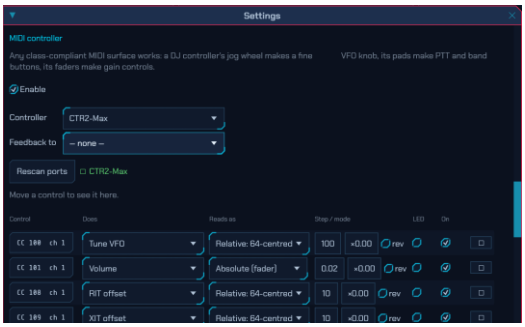
AetherSDR has added a lot of support for Flex radio users using **Max**. A new **CTR2 Relay** option allows you to connect **Max's Flex** mode to AetherSDR over USB and operate your radio just as if you were using WiFi on the other **CTR2** controllers.

The developers at AetherSDR have also created a new app that runs exclusively on **Max's** hardware, called **AetherCTR2**, that controls your Flex radio. This app replaces the factory **CTR2-Max** firmware with a new, modern user interface. You can choose between **Max's** original firmware or **AetherCTR2** with the touch of a button.



See the CTR2-Max Flex Manual for more information about these features.

SDRoxide Maps



SDRoxide is another new radio control app that supports multiple radios and MIDI (and thereby **Max**).

To enable MIDI control, select **SYS->SETTINGS->Controls** then scroll down to the **MIDI Controls** section.

The app is currently under active development and doesn't yet support loading or saving MIDI configurations. You'll need to manually enter the configuration you want to use with **Max**.

I have a spreadsheet and a **Max** configuration file in the **SDRoxide** folder in CTR2-Max Config Files with my suggested mapping.

Advanced Options

Operation with N1MM Logger+

You can use **Max** as an interface to use N1MM's CW keyer with SmartSDR or SDR-Control for iOS. To do this, follow the instructions on the N1MM Interfacing Basics webpage to build a key interface circuit. This will require a USB serial adapter with a keying interface circuit wired to its RTS output. Connect the output of the keying interface circuit to the TIP and SHIELD of **Max's Key/PTT** input jack. Next, map MIDI button command **98** to **Trigger CW Straight Key**. With this setup you can control your radio using the N1MM Logger+ or **Max** and use keyboard keying in N1MM Logger+.

The biggest advantage of using **Max** over using a mouse to control your radio is that commands from **Max** go directly to your radio control app whereas your control app must be "in focus" to use a mouse. This is especially handy when running logging programs like N1MM Logger.

Unfortunately, this configuration doesn't allow you to use a set of paddles with **Max** at the same time nor does it allow N1MM Logger+ to control your radio remotely.

If you're looking to have N1MM Logger control your radio remotely you might consider using **CTR2-Micro** to remotely control and key your radio with SmartSDR or SDR-Control providing the user interface. Instructions for this can be found in the CTR2-Micro Operation Manual under the **Advanced Options** section.

Troubleshooting

Issue	Solution
I can't connect the app to **Max** using USB MIDI	1) Make sure you are using a USB-C data cable. Many USB-C cables are power only. These will not work with USB MIDI. 2) If using an older Apple device (iPhone or iPad) with the Lightning connector you must use an OTG adapter for your device. This adapter provides a USB connection for an external device. Newer iPhones and iPads with USB-C connectors do not require an OTG adapter.
I have to connect **Max** every time I start the app	You may need to need to open the **MIDI Configuration** window in Thetis every time you open that app it so it will find **Max**.
My Flex radio starts sending a string of Dits or Dahs when I select CW mode	This seems to be a bug SmartSDR for iOS/MacOS v6.9.15. If you press the paddles when the app is not in CW mode but the CWX panel is open. The paddle press is buffered in the app. When you change to CW mode from another mode, CWX triggers the last paddle press you made and keeps sending it until you press the paddle again, turn off CW mode, or close the CWX panel. This should be fixed in the latest update of SmartSDR.
Max connects to my device but does not control anything	1) Verify that you have mapped each control on **Max** that you are using to a function in your app.
Frequency tuning is not working	1) Make sure you have the correct map selected in the **Settings** menu. 2) Use **WheelA** for RHR and Marcus' iOS and MacOS apps. Use **WheelB** for Thetis and other Windows apps. 3) Make sure you have the correct MIDI command mapped to the frequency control function in your app.
How can I tell when I'm at the center (0 Hz) when controlling RIT or XIT?	Use the **SliderB** control for **RIT** and **XIT** controls. The display will show a black bar indicating the offset and **Max** will beep and pause tuning for about ½ second when the control reaches its center position (64 counts). **SliderA** does not beep at center or show the +/- offset.
Speed tuning is not working	Make sure **Speed Tune** is set to **On** in page 1 of the **Settings** menu.
Paddles don't key the radio	1) Verify the radio is in CW mode 2) Verify the keyer is in **Breakin** mode and **Sidetone** in **On** in the P/CW panel 3) Verify **Pdl->MIDI** in the **Settings** menu is set the same MIDI control #s that you have mapped in the app for left and right paddle control. 4) Open the CWX Panel in the app's **View** menu 5) If you're using a Flex radio, it might be in CW Zombie mode.
Paddles are reversed	Remap the MIDI control #s to swap Left and Right paddle assignments
No sidetone in SmartSDR when keying	Open the left pop-out window in the Panadapter display on the app and click the **Audio** menu. Set the **Local Audio Monitor** slider to 0.
Frequency does not change correctly when I tune the *Dial* or *Knob*	First, make sure you have mapped the MIDI control # you are changing to your app's frequency control function. Second, there are two MIDI control types used to tune frequency, **WheelA** and **WheelB**. You must select the correct type for your app. • **WheelA** is used in Marcus' iOS/MacOS apps and in RHR. • **WheelB** is used in Thetis and other Windows programs If the wheel is the wrong type change it by opening the **Edit Dial Setting** option in the **Settings** menu. Touch the **Type:** label to step through the types available.

I can't get the ESP32-S3 in **Max** l into programming mode	There are two buttons that must be pressed in sequence to force the ESP32-S3 into programming mode. On the back of the unit, next to the USB-C connection, use a toothpick or small screwdriver to press and hold the **Boot** button then press and release the **Reset** button, then release the **Boot** button. **NOTE: The buttons are on the display PCB so they are upside-down. To operate them, push up, toward the face of the enclosure.**
The tuning controls don't work	1) Make sure you have selected the correct MIDI Wheel Type for the app you're using. Most Apple apps (SmartSDR, SDR-Control, etc. use type **WheelA**. Windows apps (Thetis, SDR-Console, etc.) tend to use type **WheelB**. PiHPSDR uses **WheelB-r**. 2) If the tuning control has been locked it will be "grayed out". To unlock the control, long-press on the control's label.
How do I know which app **Max** is configured for?	You can to enter a name for each map on page 4 in the **Settings** menu. This name is displayed in the caption of the **Home** page. Set the name to the name of the app this map is configured for (i.e. SmartSDR, SDR-Control, Thetis, etc.)
The Key and PTT jack does not control my radio	The 3.5mm jack(s) on **Max** are designed for input only. Connecting these jacks directly to your radio may damage **Max**. To use a straight key, external keyer, or external PTT switch to control your radio, refer to this section.

Appendix A: Connecting CTR2-Max to Your App

This section describes how to connect **Max** to various apps. Once you are connected to the app see the How CTR2-Max Works with Your Control App section to understand how **Max** controls work with the app.

Unlike the other **CTR2** controllers that support Bluetooth, **Max** only supports a direct USB connection to the computer running your radio control app. Do not plug **Max** into an unpowered USB hub. These hubs cannot supply the current **Max** needs to operate reliably.

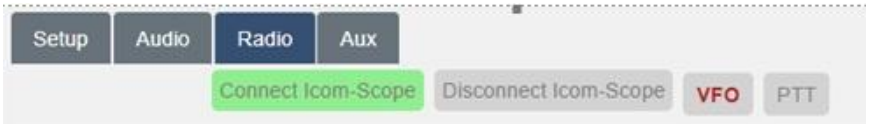
Remote Ham Radio

To connect **Max** to Remote Ham Radio simply plug **Max** into your one of your computer's USB ports. **Do not use an unpowered hub**. Set **Max** to run in **MIDI** mode. Next, select the **CTR2-MIDI** option in the RHR console's **Tool** menu. The **CTR2-MIDI** option is pre-configured for **CTR2-MIDI** mapping so you'll need to change the configuration of **Max** to match RHR. You can see how **Max's** controls should be mapped to RHR in the Remote Ham Radio Configuration section.

RemoteTx

RemoteTx is a browser based remote control system. Instead of connecting to the browser using MIDI commands, **Max** connects to **RemoteTx** as a HID (Human Interface Device). This eliminates most of the problems users experience with MIDI control in browser, especially when refreshing a page.

NOTE: RemoteTx is a subscription service and requires a Chrome, Edge, or Opera based browser.



To connect **Max** to RemoteTx, plug **Max** running in **MIDI** mode, into a USB port on your computer and start the **RemoteTx** web page. Select the *Radio* button at the top of the page click the **Connect Icom-Scope** button. Next, click the **VFO** button. Next select the **ESP32S3_DEV** device from the dropdown list.

For information about setting **Max** up to work with **RemoteTx**, see the RemoteTx Mapping section

AetherSDR, RimuSDR, and SDRoxide

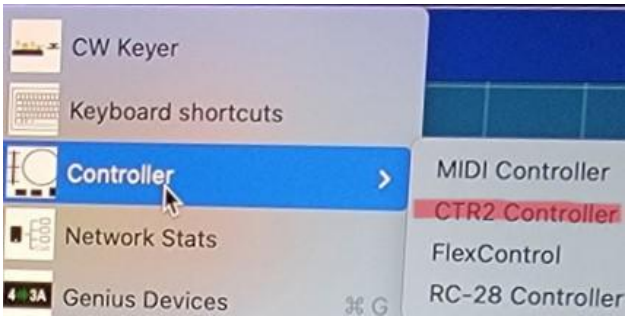
Plug **Max** into a USB port on your computer. They will automatically detect **Max** once you have loaded the settings file for the app from my CTR2-Max Config Files page.

NOTE: There is a known bug in Windows where **Max** will not be enumerated properly if it is plugged in and powered up before you boot Windows. If your app doesn't find **Max** automatically, simply cycle the power on **Max** after you have started the app.

Marcus' Apps for iOS or MacOS

To use **Max** running **MIDI** mode with a USB connection on Mac's and newer iOS devices, connect **Max** to your device using a USB data cable. Next, select the **CTR2 Dial Controller** from the apps **Tool** menu.





This will open the **CTR2 Dial Controller** page.

From here, select **M5STAMP** from the **Device** list (this is the name Marcus' apps use to identify all **CTR2** controllers with displays), and click the **Enabled** button. The **Status** should change to **Connected**.



Press the **Edit Mapping** button and press a button or turn the encoder to verify the app is receiving MIDI commands.

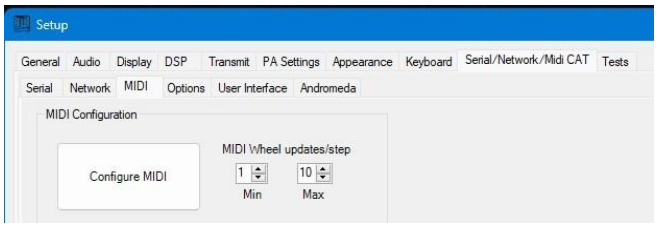
NOTE: To use a USB connection with an older iOS device (with a Lightning connector) you'll need to purchase a Lightning OTG (*On The Go*) adapter such as the one shown here. These are also referred to as "Camera Adapters" on Amazon. The adapter I use has two USB-A connectors on it. It also has a Lightning port so you can connect a wall charger or battery to power your phone **Max**.



Other Windows Applications

Thetis

When you connect **Max** to your Windows computer's USB port it will automatically register as a MIDI device and be assigned a virtual COM port.



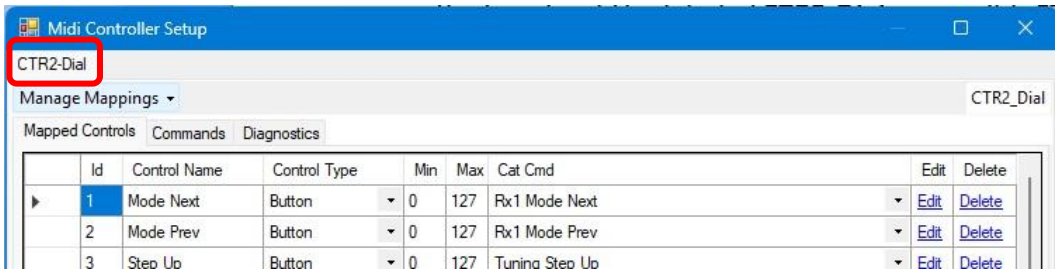
NOTE: USB MIDI requires a USB-C data cable. One is provided with your **Max**. If you need a longer cable, don't buy a USB-C power cable, it will not work!

Most Windows apps will need to be configured (i.e. mapped). **Thetis** is used as an example here but connecting to and mapping controls in other apps should be similar.

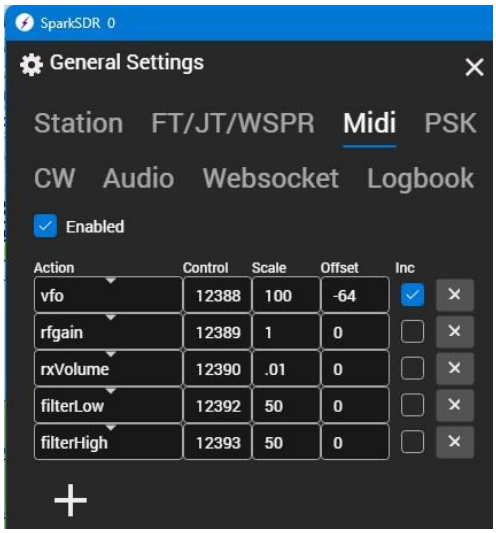
Additional information on mapping MIDI controls in Thetis can be found here.

Start by selecting the **Settings** menu on the main page in Thetis then select the **Serial/Network/MIDI CAT** tab. Click the **MIDI** tab as shown above.

Next, click the **Configure MIDI** button to open the **Midi Controller Setup** page. The program should go through an initialization sequence and end up on this page if it finds **CTR2-Max** on the USB port. The tab at the top should be labeled **CTR2-Max** or possibly **ESP32S3_DEV**. If the initialization fails, double-check that you are using a USB-C data cable, not a charge cable and try again.



SparkSDR



SparkSDR (https://www.sparksdr.com/) has limited support for MIDI control. It doesn't support buttons, only the *dial* and *knob* controls. To map **Max's** knob functions to this program, first plug **Max** into a USB port on your computer and open SparkSDR. Once you have the program running, click on the **General Settings** menu and select **Midi**. You should see **CTR2-Max 0** as a MIDI controller and turning the encoder should display MIDI messages from the controller. To map these messages to a control in the app simply press the **+** icon and enter the **Control** number shown in the **Last midi message:** window in the **Control** column. Then select an **Action** and set the **Scale** and **Offset** required for that control. You may need to play with these values to get the control action you want. Check the **Inc** box if you want the value from CTR2-Max to be added to the current setting of the control. This is only used with the **vfo** control. This screenshot shows the settings I use.

Appendix B: Installing or Updating CTR2-Max Firmware

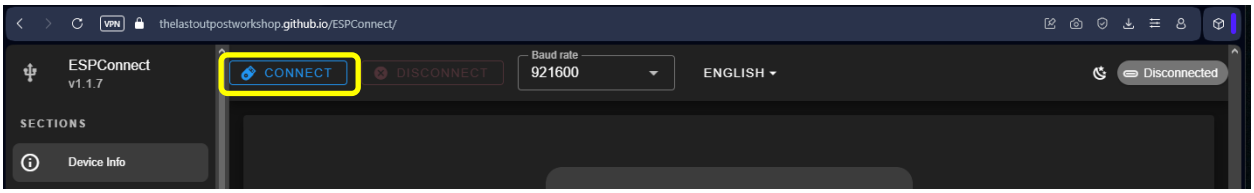
To update the firmware on **Max**, download the latest **CTR2-Max** firmware from my web site. Unzip the firmware distribution zip file into a folder.

Installing Firmware using ESPConnect

By far, the easiest, and recommended method to install or update firmware on an ESP32 processor is with ESPConnect, an open-source ESP32 management project. You can use the EspressIF **Flash Download Tool**, but it is geared for developers and many find it difficult to use.

ESPConnect is a browser-based tool that must be opened on a Chromium-based browser that supports Web Serial, such as Chrome, Edge, Brave, or Opera. It will run on a Windows PC or an Apple Mac. Unfortunately, Firefox, Safari, and all Linux browsers don't support Web Serial.

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



Next, plug **Max** into a USB port on your computer.

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

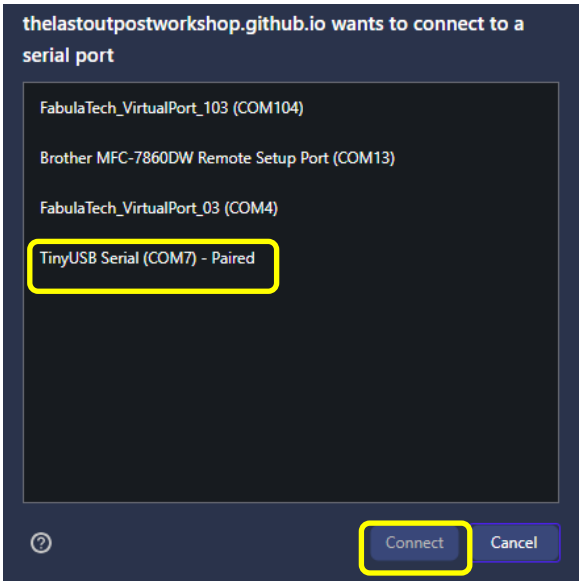
Click the **Connect** button.

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

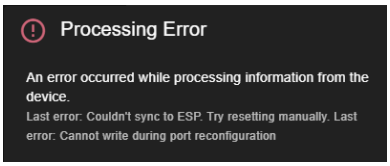
On a Windows PCs, **Max** will be shown as a **TinyUSB Serial** device, as shown here.

On Macs, **Max** will be listed as an **ESP32S3_DEV** device.

Select the **Max** 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. You do not need to do this.



This is the normal response when trying to connect to a **Tiny USB serial** device. You'll notice that **Max** is frozen at this point because it acutally did enter boot mode!

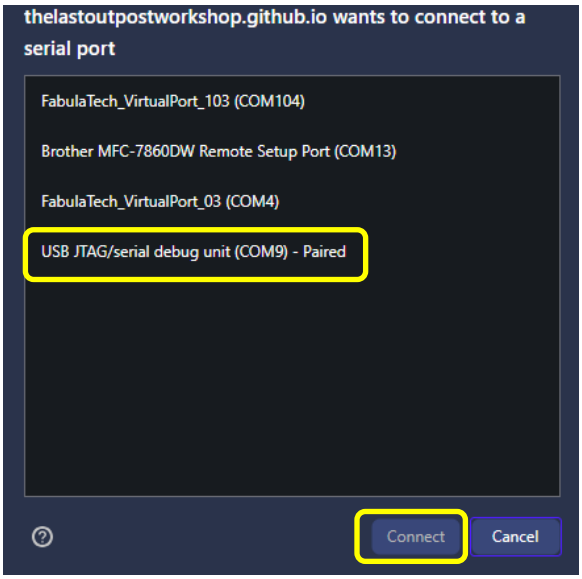
Click the **Connect** button on the **Home** page again.

This time, the **TinyUSB serial** (or **ESP32S3_DEV** 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**.

NOTE: Apparently, not all ESP32-S3 processors will enable the **USB JTAG** port automatically. If your's doesn't, you'll need to force it into boot mode. Do this by pressing and holding the **Boot** button on the back of **Max**, then press and release the **Reset** button.



Once you're connected to **Max** in Boot mode, 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 **Max's** 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.

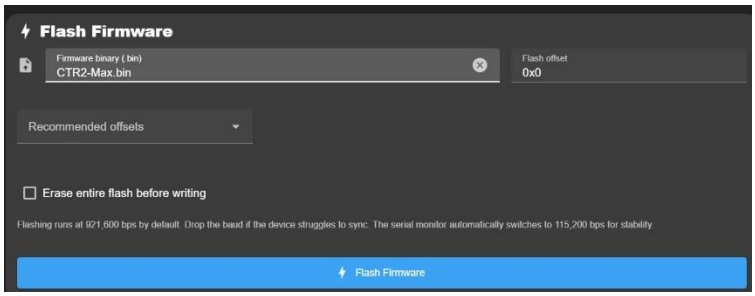


A file navigation screen will appear. Navigate to where you unzipped the firmware zip file from the firmware update page and click on the **CTR2-Max.bin** file.

Your browser's display should now look like this:

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

WARNING: If you check the erase box, you'll need to re-enter your call and registration key and reconfigure your unit.



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

Once the process completes, cycle the power on **Max**. If your unit has an internal battery, turn the **Battery** switch OFF then disconnect **Max** from the computer's USB port. Next, plug it back into the computer. The version # will be displayed when the unit boots. The version # should match the version # from the firmware zip file.

This completes the firmware update process.

Enter your Call and Registration Key

If you happen to erase the flash memory when you update the firmware, your unit will start in **Demo** mode and you'll need to re-enter you call and registration key.

Your call and registration key are written on the label on the bottom of the unit. They are also in the confirmation email I sent you when you purchased your unit.

NOTE: The registration key is an 8-digit Hex number. It only uses characters 0-9 and A-F. If your unit refuses to register your key, verify you are entering the correct digits. Sometimes my handwriting can be questionable!

There are two ways to enter your registration credentials.

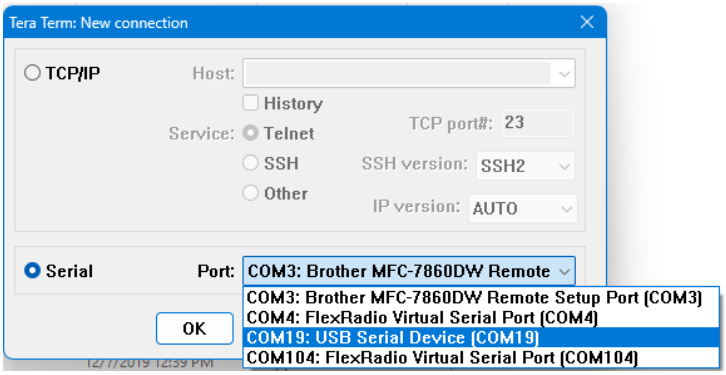
1. Connect your unit to a terminal program such as Tera Term or Putty (see Appendix C or D, or to use the terminal in your Mac, Appendix E). Once connected, press any key to start the terminal server on the unit. Next, press the **Del** or **Backspace** key on the keyboard. Enter the call sign you registered and the registration key you received from me.
2. Use the on-screen editor to enter your registration credentials. Select **Enter Call & Key** on the last page in the **Settings** menu. Use the [**^**] and [**v**] buttons to change the letter and the [**<**] and [**>**] buttons to advance to the next letter position. [**Ins**] and [**Del**] allow you to add or delete characters. Press the [**OK**] button to accept. The input sequence starts with your call and advances to your registration key once your call has been entered.

Appendix C: Configuring Tera Term

Tera Term is the simplest terminal program to get running for a serial connection.

If you search for Tera Term you 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 IO files 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 **Max**. Press any key to start **Max's** terminal server.

You can change the terminal size in the **Setup** menu. Select **Terminal…** Set the **Terminal Size** to at least *41 x 20.* **Max'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 **Max**.

Appendix D: Configuring Putty

Putty is a terminal program that can be configured for a variety of needs. **CTR2** controllers only support serial connections. This section describes how to configure the program to interface with **Max**.

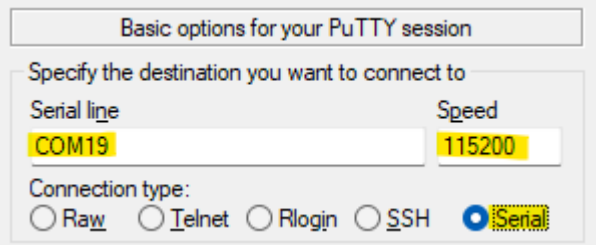
NOTE: Putty does not have XModem transfer capability. If you want to export and import backup files for **Max's** settings consider using Tera Term or install ExtraPutty.

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 **Max** using its USB serial port in order to configure it.

Serial Session

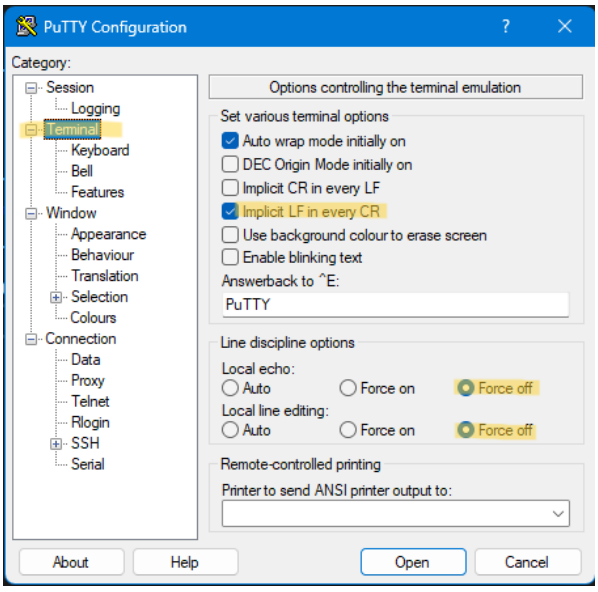
Select **Serial** then set the **Serial Line** to the COM port you found in the Device Manager and set **Speed** (Baud Rate) to 115200.



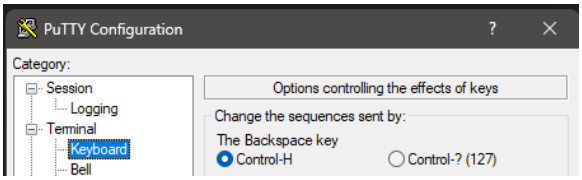
NOTE: Since this is a USB serial port the **Speed** (baud rate) doesn't matter. Data will be sent at USB speeds regardless of the **Speed** setting.

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*.



Next, select the **Keyboard** menu option and note the setting for the **Backspace key.** If **Control-H** is selected you will need to press **Ctrl+Backspace** to send the **Del** key (ASCII 127). If **Control-? (127)** is selected, press the **Backspace** key to send the **Del** key code.



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 **Max** to a terminal session, use the following process.

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

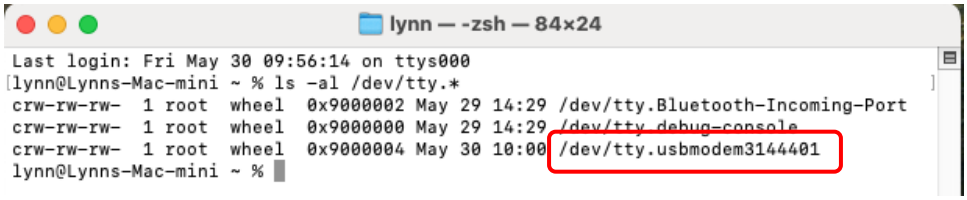
- On the Mac open **Applications/Utilities/Terminal.app**. On Linux open the terminal program supplied by your distro.
- On the Mac, enter ***ls -l /dev/tty.usb****, on Linux, enter ***ls /dev/tty**** This will return a list of all known serial ports.
- Next, plug the *Flex* into the computer's serial 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 **Max** into your PC, make sure you are using a USB-C cable that supports data. Many USB-C cables that come with consumer electronics only provide power to the device.

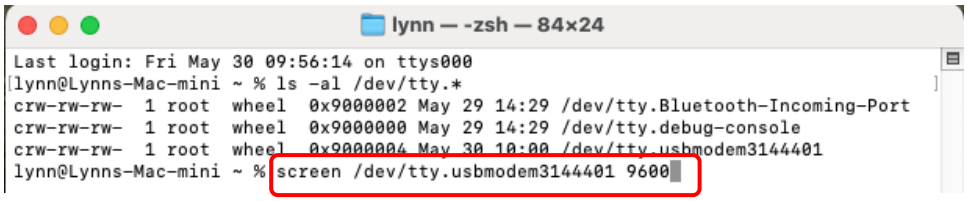
Once you know **Max's** USB serial port ID, write it on the label on the bottom of the unit using a fine-tipped permanent marker for future reference. Put a piece of transparent tape over the label to seal the ink so it doesn't rub off (it's not as permanent as you think). You can always remove the tape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

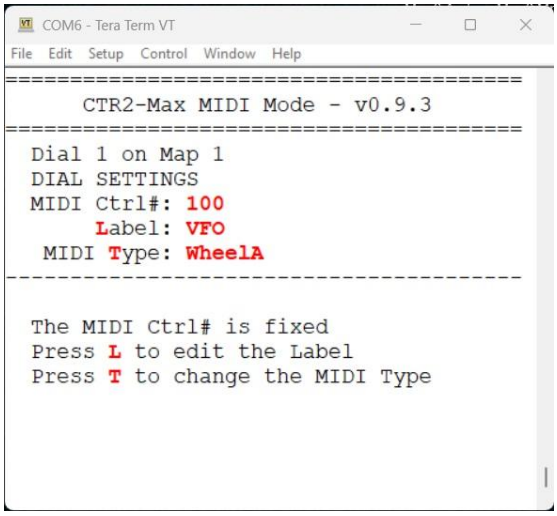
1. Get the list of serial devices on your computer. We're looking for the **usb** device.



2. Open the **screen** utility using the **usbmodem31444-1** device. Your USB device will have a different numerical #.



3. This is **Max's** terminal display. Select a dial control or button to edit that control's label.



Appendix F – Power Requirements

Max is designed to support an internal 3.7-volt Lithium-Polymer battery. The display board has a charge controller that will charge at approximately 300 mA until the battery reaches 4.2 volts, then it will reduce to 0 mA. The unit can monitor the approximate battery voltage. This is used to provide a relative state-of-charge via a battery icon that can be enabled on page 5 of the **Settings** menu.

The battery mounts in the base of the unit and has a strap to secure it. The enclosure a battery up to 55mm L x 35mm W x 10mm H. This allows a 2000 mAH battery to be installed. This battery will power the unit for about 12 hours.

Battery installation is covered in the **CTR2-Max Hardware Manual** available on my website.

NOTE: Since **Max** only supports USB MIDI, it must be connected to the USB port of the computer or Apple device running the radio control software. Therefore, the internal battery in **Max** is only useful when the computer or Apple device you're connected to is operating on its own batteries. **Max's** internal battery will power **Max** so it doesn't draw power from your device's battery.

Appendix G: Using CTR2-Max with Non-MIDI Programs

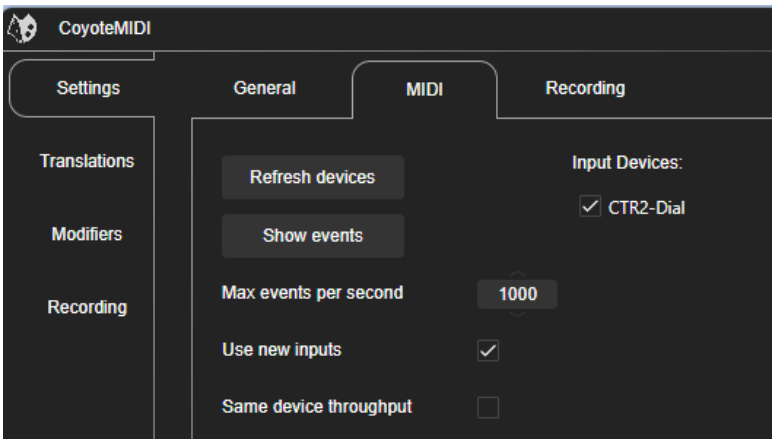
MIDI mode is not limited to programs that support MIDI controls. Third-party MIDI translator apps such as CoyoteMIDI can receive MIDI commands from **Max** and convert them to other actions such as mouse scroll or keystroke actions. MIDI translator programs are available for Windows, Mac, and Linux.

These apps allow **Max** to be used as a physical tuning knob with radio control programs such as SmartSDR for Windows, wfView, and WebSDR that don't normally support external tuning knobs. And it doesn't stop there. These apps also allow **CTR2-Max** to be used with any program that accepts mouse scrolls or keystroke for control. For instance, it makes a great jog wheel for Audacity.

There are a couple of drawbacks to using a translation app instead of native MIDI control. First, like your mouse or keyboard, the app you want to control must be "in focus" to receive the mouse scrolls or keystrokes from the translation app. Second, most programs don't have a lot of control functions tied to mouse scroll or keystroke actions. For instance, SmartSDR for Windows only accepts mouse scroll actions for frequency control and wfView only supports the J and K keys for frequency control.

Configuring CoyoteMIDI

This section will cover configuring CoyoteMIDI to interface with SmartSDR for Windows and wfView. The translator you choose will have a different setup procedure but they're all basically the same.



Before you start, plug **Max** (running in **MIDI** mode) into a USB port on your computer. It will automatically register as a MIDI device. Open the **Dial** control and turn the gray ring to select the first control, typically named **VFO**. There

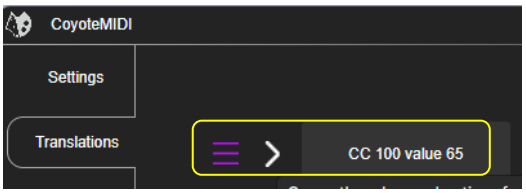
Once you've installed CoyoteMIDI open the app and navigate to the **Settings -> MIDI** tab. **CTR2-Max** should show up on the **Input Devices:** list. Click the checkbox to enable it.

Configure SmartSDR for Windows Frequency Control

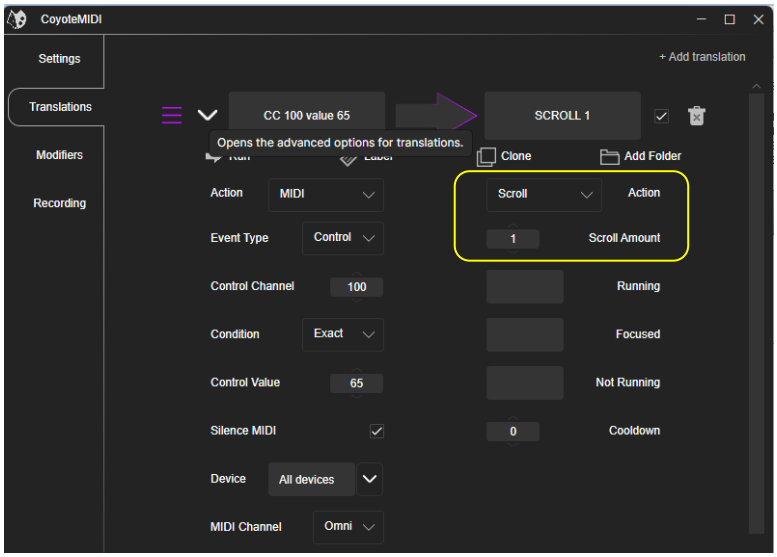
Next, select the **Translations** tab and click the **+ Add Translation** button in the top-right corner.

The first example uses the **VFO** dial control (MIDI control #100). This mode uses the **WheelA** MIDI control where a value of 65 represents a clockwise turn and 63 represents a counter-clockwise turn of the knob.

Click the box with **Click to record** then turn the knob on **Max** *slowly* clockwise. You should see **CC 100 value 65** appear in the left box. If you don't see this, make sure the selected control on **Max** shows control 100 above the control's label. Hover the mouse over this entry to show the violet menu and **>** icons.



Click the **>** icon to view the details of this command. I've already set this command to be translated to the **SCROLL 1** mouse command.



The options on the left should be as shown. If not, edit them to match what's shown here.

To configure the **SCROLL** 1 output, click the **Action** dropdown box and select **Scroll**. Next, set the **Scroll Amount** to **1** so each tick of the clockwise turn of the encoder produces on scroll tick.

That's it. You configured your first translation. Click the **+Add Translation** button again and follow the same process to program the counter-clockwise knob turn. In this case, set the **Scroll Amount** to **-1**. This will send the reverse mouse scroll.

Now open SmartSDR for Windows and the frequency should change as you change the knob on **Max**. You can edit the label on **Max** if you want to change it.

Configure wfView Frequency Control

You can set up translations for many other programs the same way. Next, we'll go over setting up keystroke "J" and "K" to control the frequency in wfView.

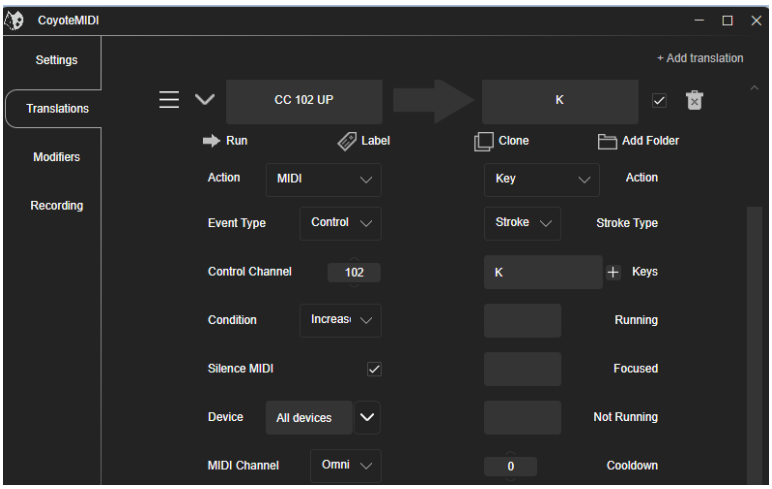
On **Max,** to switch to control 102 on the **Dial control**. This is normally shown as **AGC-T**. You can change this label with a terminal program. You can also use any other knob control.

NOTE: This encoder uses the **SliderA** control type. This type operates like a potentiometer and has a range of 0 to 127 counts. To use this control type, we'll look for the value increasing or decreasing to determine the direction the knob is turning. You can edit the MIDI control type in a terminal if you want to change it to **WheelA**. If you do this, follow the setting in the first example.

Click the **+Add Translation** button to add another translation.

Click the box that says **Click to record** then turn the knob on **Max** clockwise. You should see **CC 102 value xx**, where xx is a number between 1 and 127.

Hover the mouse next to this box then click the **>** icon to show the details.



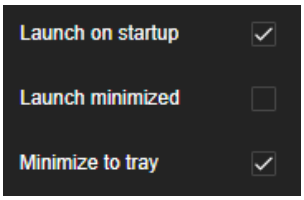
On the left column, click the **Condition** dropdown box and select **Increase**. This action will trigger when the MIDI value is increasing. The title in the top box will change to **CC 102 UP**.

In the **Action** dropdown box in the right column select **Key** and in the **Stroke Type** select **Stroke**. Finally, in the **+ Keys** box, enter **K**. This is the key used to increment the frequency in wfView.

Next, click the **+ Add Translation** button again, click the new box showing **Click to record** and turn CTR2-Max's knob counter-clockwise. Now, follow the same process as outlined above except select **Decrease** in the **Condition** box and enter the letter **J** in the **+ Keys** box. wfView uses the **J** key to decrement the frequency. I'm not aware of any other keystrokes wfView supports. If there are more they can easily be added.

CoyoteMIDI Startup Options

There are a couple of options in the **Settings -> General** menu that you might want to set if you regularly use the *MIDI* and CoyoteMIDI with your apps. If you select **Launch on startup** and **Minimize to tray** options CoyoteMIDI will automatically start when you boot your computer and will be available in the Windows tray if you need to change anything.



Appendix H: Change Log

Older change log entries will be saved here.