

LOREM IPSUM

tinyCCM

Manual

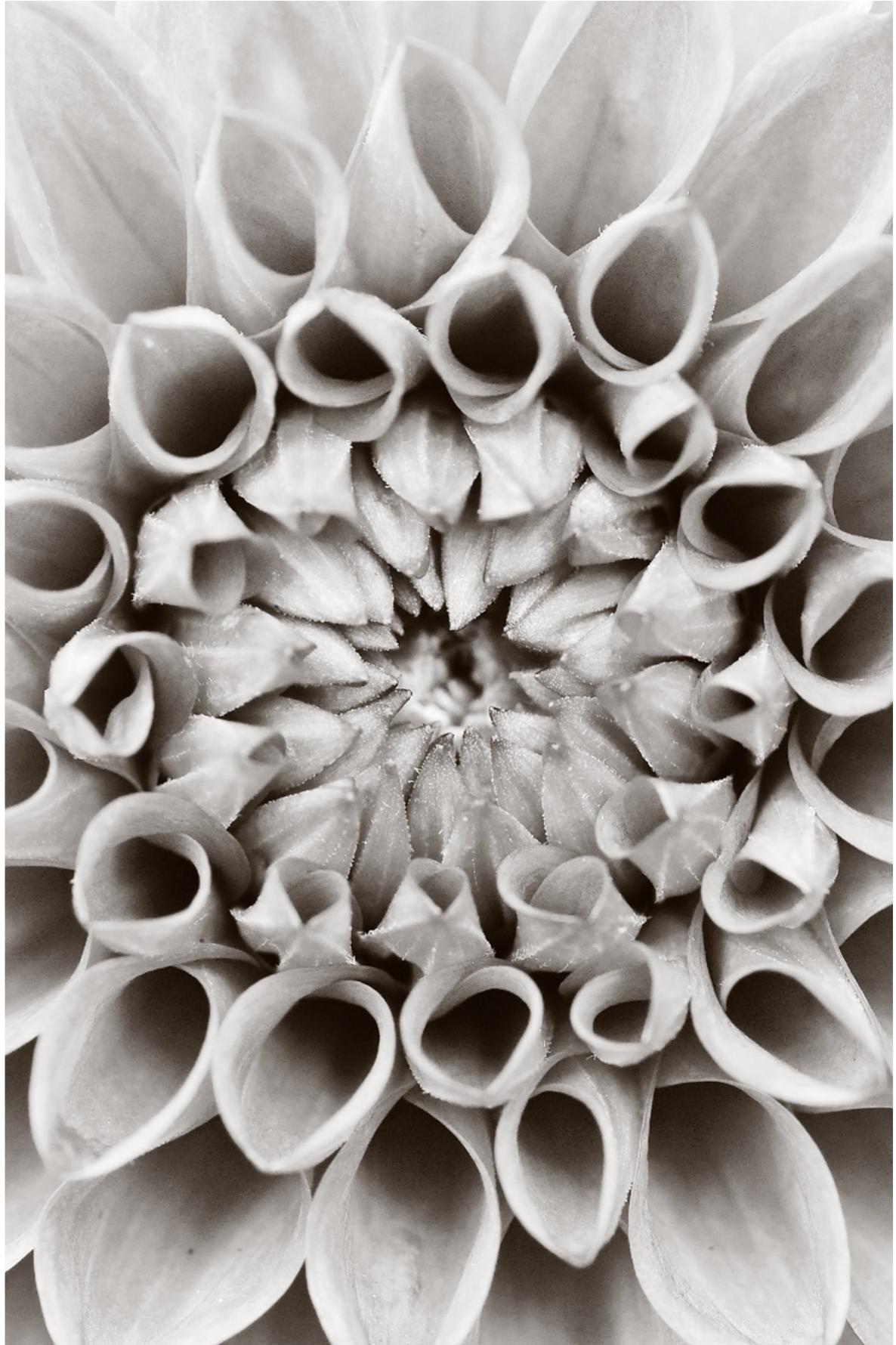


DOLOR SET AMET

Introduction

Let me first thank you for buying this product - it's been more than a year of reverse-engineering and talking to Cadac employees when I got stuck, coding the application, the firmware, drawing the PCB layout and components, enclosure design, sourcing parts, finding the right cables and packaging material and tons of hours of testing and improving. This may sound like it's a final product and while it is fully functional I'm sure there will be good feedbacks from you as a user to improve it further.

It's my first time creating a product of such complexity. Let me start with the introduction of what the tinyCCM is and does and what is needed for installation.



Why & How

Cadac J-type owners love the desk for well-known reasons. Most of them use it for mixing or stem-summing in a hybrid setup. The DAW is a tape machine that will be used for persistence and editing of audio tracks in addition of some plugins, mixing and possibly some EQing is done on the desk. For such a workflow the tinyCCM is exactly what users has been waiting for. Even when the Cadac CCM as a 2-slot module offered PC integration using SAM software it was almost impossible to have the desk's data in your DAW project. There exist also fully-programmable modules but very few people have their full desk loaded with them. This is different to the motorized channel faders that are more popular and I know some users having a J-type with 32 motorized channel faders. The non-motorized faders do indeed use a VCA and one might think that they can be remotely controlled apart from the movement but they lack the Fader Comms bus. This is a serial hi-speed async bus carrying the Fader Comms protocol that controls every state of a motorized fader, from the simple fader position up to the button states such as ISO, AXU VCA, Groups, Mute etc. The idea of the tinyCCM is to replace the Cadac CCM to drive the Fader Comms bus. Because every signal is available on the rear on the DSUB connectors, an external unit can drive the signal in the same way as the Cadac CCM does and in fact, the Cadac CCM can be removed, offering two extra slots that can be used for audio processing. You will loose the control of the

„upper“ modules, Events, Midi, Cues, Seance connection, PSU indicator and other things but you gain what we think is more valuable these days: DAW integration for the faders.

There exist also motorized DC faders that are connected to the Fader Comms bus - they are really rare, I managed to get some of them, and the tinyCCM recognizes them and drive the scribble display for status information. However there is no clear solution about what to do with the DC groups in a DAW context - so they are simply ignored. Maybe a future software update might find some use for them.

Another thing that had to be done on the Cadac CCM was mapping. Mapping is the process of knowing which module is in which slot and frame position. This allowed for non-consecutive order of motorized channel fader and to assign fader 1 to be somewhere in the middle of the frame. The tinyCCM automatically maps one frame (up to three frames can build one logical desk) you cannot cascade more than one frame and „continue“ with the fader count on the next frame. Auto-mapping discovers all motorized(!) faders in the connected frame and will simply assign the first (usually left-most) fader to be „fader 1“, then continues „to the right“ until all faders on the Fader Comm bus are found. In a hybrid mixing/summing environment this should cause no issues because one expects the order of the DAW faders to be in the same fashion on the J-type. The number of found faders will be sent to the application. We will see later why this matters.

SECTION 2

What it does and what not

KEY FEATURES

1. Reads and writes up to 32 channel fader movement data to/from your DAW in 10bit resolution
2. Consists of a tinyCCM Application and the tinyCCM hardware unit.
3. DAW's that support OSC can drive the tinyCCM directly without the application.
4. Using the application this emulates up to 4 HUI instances (4x8ch) so your DAW thinks it's connected to a HUI control desk
5. The application can save and recall static snapshots of your desk similar to the Cadac CCM. However only the motorized fader modules are supported.
6. By getting rid of the original CCM you have another 2 extra slots for audio module
7. is never meant to be a full replacement of the Cadac CCM - hence the name tinyCCM.

SECTION 3

The tinyCCM Application

You may have a basic understand what the tinyCCM hardware unit does but why is there a tinyCCM Application that goes along with it? Well, in theory it is not needed. The tinyCCM hardware unit accepts OSC command according to the schema in a later chapter. If your DAW natively speaks OSC and if you can customize what shall be sent / received for each DAW event then you could probably drive the tinyCCM without the tinyCCM Application. In theory I said. But we are in a real world where the common denominator of DAW control of a physical box goes years back to MIDI. 1997 to be exact.

Mackie came out with the HUI remote controller that I still love today and the “language” to slave the DAW to whatever you did on the control unit and vice versa was the Mackie HUI protocol that never died. And it should never die. While we are waiting for a new pseudo-standard across multiple DAW manufactures with a contemporary take of what we know as API (Application Programmable Interface, not the API brand) - it’s simply not there. Come on dear pro audio industry and

work together - make your product “integrate-able” the same way as digital audio protocols. We are grateful that Mackie HUI integration never disappeared from the major DAW’s because it defines a language (a schema) of what do to if we want the DAW channel 3 to be muted. Due to its age and the limited microprocessor performance back then, it is so much low level that it is very cumbersome to implement it: If you miss an event, there is no way to ask the control of whether mute for channel 3 is on or off - you have to get it in the first time or you are out-of-sync. Contrary to many critics, HUI did infact account for 14bit fader positional data - even when MIDI only knows 7bit values (besides SYSEX). Yes, OSC helps but it more a transport layer of message exchange over Ethernet (UDP) - OSC never standardized the implementation of what function exist and how it should be named. We call that schema-less. Every take on OSC is different so while Reaper uses „/track/3/mute 1“ to mute the third track, RME uses „mute/1/3“ to do the same. A schema is missing here and it would lead to implement every DAW integration for each own. OSC is excellent in such that it specifies how a message (any message) shall be sent but not what message. The tinyCCM Application is a gateway, a converter that translates Mackie HUI command from a compatible DAW and sends the OSC message in a „tinyCCM language“. The tinyCCM hardware listens to this message, and since both ends are under my design it will put this data to the Fader Comms bus. Once there, the fader picks it up, will read the data and follows the action (if not in ISO). This also works the other way around so when you move a fader on the desk, the motorized channel

fader will wait for the next query of the tinyCCM (or CCM) every 40 milliseconds (40ms? this should ring a bell for the older reader - that's $1/25$ of a seconds or 25 frames/s. SMPTE anybody?) and writes the fader position as new data. The tinyCCM evaluates the new position, recognizes a change and sends this „event“ via OSC to the tinyCCM Application where the OSC message is converted back to Mackie HUI language. Since the tinyCCM Application is in the middle and knows about all events and data it can also do more stuff than the DAW can do. In fact, the tinyCCM Application can be run in standalone mode without any MIDI/HUI activity. This allows you to use the user interface to control the desk, save a cue or load a cue and find support for Cadac related things such as Groups, Aux VCA, ISO, BYP that cannot be mapped to Mackie HUI. Other than such „operational“ data, communication of configuration data is done. This includes number of faders found, a ping mechanism to assure that the tinyCCM is still connected (OSC mostly uses UDP which is connectionless - we have to ping!), exchange of a license and serial number to check that the firmware and software versions are matching etc.

You might ask where are the MIDI IN and MIDI OUT ports of the tinyCCM Application. And how does the DAW know about the tinyCCM Application. Under OSX, there is a virtual MIDI system called „IAC“ - you can create as many ports you want because they are virtual. The data sent into a IAC port MIDI In will arrive at the MIDI Out port of the same IAC port - so they are „pipes“ - what you put into will appear at the other

end. This happens at a much faster speed compared to hardware MIDI which is bandwidth limited to around 30kbit/s. Some folks use IAC for ReWire type of application integration when you want to overcome system breaks (i.e. Logic shall speak with Ableton in realtime while both are running). The IAC ports that must be created must follow a special naming convention in order for the tinyCCM Application to know which are the correct ones. See the Installation chapter on how to do this. At the DAW end you simply declare 1, 2, 3 or 4 instances of HUI each having their IAC port pair (In/Out). Each instance carries 8 channels worth of data. So if you have 16 motorized channel faders 2 instances of HUI and 2 IAC port pairs are enough. Since ProTools supports a maximum of 4 HUI instances we stopped testing for more than 32 faders. While it should still work to create more instances for DAW that support this, we doubt that there are many users that have more than 32 motorized faders in their desk. Let me know if you have more, it will be interesting to test it. Neither the software nor the firmware limits the numbers of faders (despite the graphical user interface only showing 32 faders)

Supported HUI commands

- Fader position

HUI doesn't tell us the dB value so we modeled the fader curve for each DAW and scaled it to match the Cadac scale. It's interesting to know that every DAW has a completely different curve. Note that HUI as a protocol supports 14 bit fader data despite the actual Mackie HUI unit only used 9 bit (still better than 7bit of first breed MIDI controllers). We can benefit from this and use all 14 bits and in fact the DAW uses all 14 bits. So the number range here is 0..16384. But Fader Comms has a narrower range of 0..964 which is 17times smaller. This

equals to around 10bit worth of data - not too bad and twice as precise than the Mackie HUI product from 1997. Thanks Cadac for realizing that in the 90s!

- Mute

Is it really needed? Glad you asked: Since the DAW is under control of which channel will send audio and we typically route the Cadac pre-fader one could argue that muting in the DAW is sufficient. When you write automation and your fingers are on your desk do you write mute automation? In favor of the little Mute Led that exist on each fader module, muting is supported in both directions.

- Solo

Solo is a complex topic. Soloing a channel in your DAW might or might not send implicit Mute commands of the non-soloed tracks. Since the Cadac doesn't have a solo system (do not confuse it with „CHECK“ that can't be automated) and because the way most users will do the audio routing of the DAW into the Cadac - we can ignore the solo but must keep track of it to make sure that channels that are explicitly muted will remain muted. There is another disaster of Solo in HUI. Some DAW let the Mute LED blink of the original Mackie HUI. Without preventing blinking, this would lead in alternating mutes on your Cadac with the rhythm of the blinking speed (like a tremolo effect). Since the user must select the DAW type in use we also know what solo behavior will be and make it right. This does include even more complex scenarios including muting channels during solo mode (who does this and why? well, you can!)

Limitations of HUI

- Banking

When a DAW knows about more than one HUI instance the question arises whether the DAW mirrors the channels 1-8 to all instances or if it auto-banks each instance to make all instances a consecutive control desk. What we want is the IAC port 2 shall become channel 9-16. Some DAW do that, some don't. For the ones that do not, auto-banking is performed in a clever way to make sure that they are adjacent to each other.

- Session open / Session close

There is no way for HUI to know if a session (a DAW project) has just been opened or closed. And we don't know how many channels a session has - when a session closes and as some DAW still show one channel (Cubase for example

shows the master channel - we do not know if this is session really had only 1 channel or if it has been closed). This in combination with auto-banking could lead into situation that manual auto-banking must be performed on the tinyCCM Application (see menu) for a second session after the DAW has been opened. Just keep that in mind, particularly for Cubase.

SECTION 4

The tinyCCM Gateway

As explained earlier the tinyCCM runs a firmware that receives OSC commands and creates the Fader Comms protocol. It then drives the bus much like the Cadac CCM did. At the same time it works the other way around: It asks every fader to send its status which includes the fader position, and sends changes over OSC to either the tinyCCM Application or any other OSC client. Connection between the computer and the tinyCCM (the OSC path) is using your LAN. So it connects to your switch or router via a standard Ethernet RJ45 cable. Please use the cable we supplied. It's black, flexible and purposely unshielded which is correct in this application. Every device in your network needs an IP address and you might be familiar with setting a static IP address or using DHCP. The tinyCCM does use DHCP so you don't have to configure anything (there are no buttons anyway). In order for the tinyCCM Application to know the ip address of the tinyCCM Gateway something called mDNS or better known as Bonjour is running to advertise services in the LAN without

knowing the ip address. That's pretty neat. When you don't use the tinyApplication and want to know the ip address you can find it by typing „ping tinyccm.local“ in your macOS terminal application. The little color indicator in the tinyCCM Application does the same. When it's green, you know that the tinyCCM is reachable. Then the ERR LED is constantly lit something is wrong with your network (most likely no DHCP possible). If you don't have a router, a simple entry-level router will do the job as it will run a simple DHCP server.

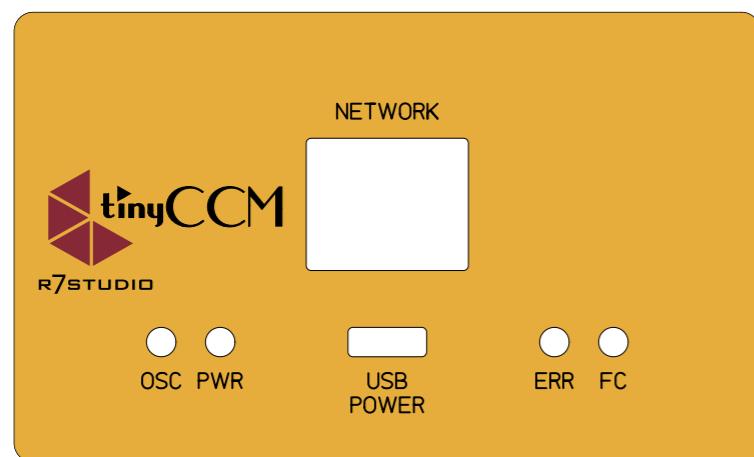
The tinyCCM is powered over USB. The connector is a micro USB type B. We do not deliver a USB power supply it's almost impossible to have one for every mains socket type and it adds weight - every household should have one anyway. Use anything that you can find to power USB devices, chargers or a cheap IKEA wall power supply. Please use the USB cable we supplied. We do not recommend powering it from a computer the reason being that the same USB port is used for firmware upgrades and the tinyCCM boots differently when it sees a computer. Note the tinyCCM needs around 500mA to 1A - almost every power supply will deliver that. For those that fear that the „noisy“ USB power can impact the audio performance of your J-type - let me say it clearly that this will not be the case. All data communication is optically isolated and „there is no power line“ between the tinyCCM and your desk.

Please also use the Fader Comms cable we supplied. It's black, it's flexible and it's shielded and it's not too long. Do not try to extend it even when you find a cable that would physically

mate. Fader Comms is a high speed bus that is sensitive to the electrical „quality“. Shipped from factory the tinyCCM is ground-lifted and the bus is intentionally unterminated. You can jumper that when you open the tinyCCM but please do that only when you face issue and you have talked with us.

Every tinyCCM has a burnt-in serial number that is unique worldwide. This is used to create a basic form of licensing since we want to make sure that the tinyCCM Application sees the correct version of the tinyCCM Gateway. If you must change the tinyCCM or the computer email us we will be happy to send you the new license key. If you sell the bundle please also deliver to license file along with the tinyCCM - they match together.

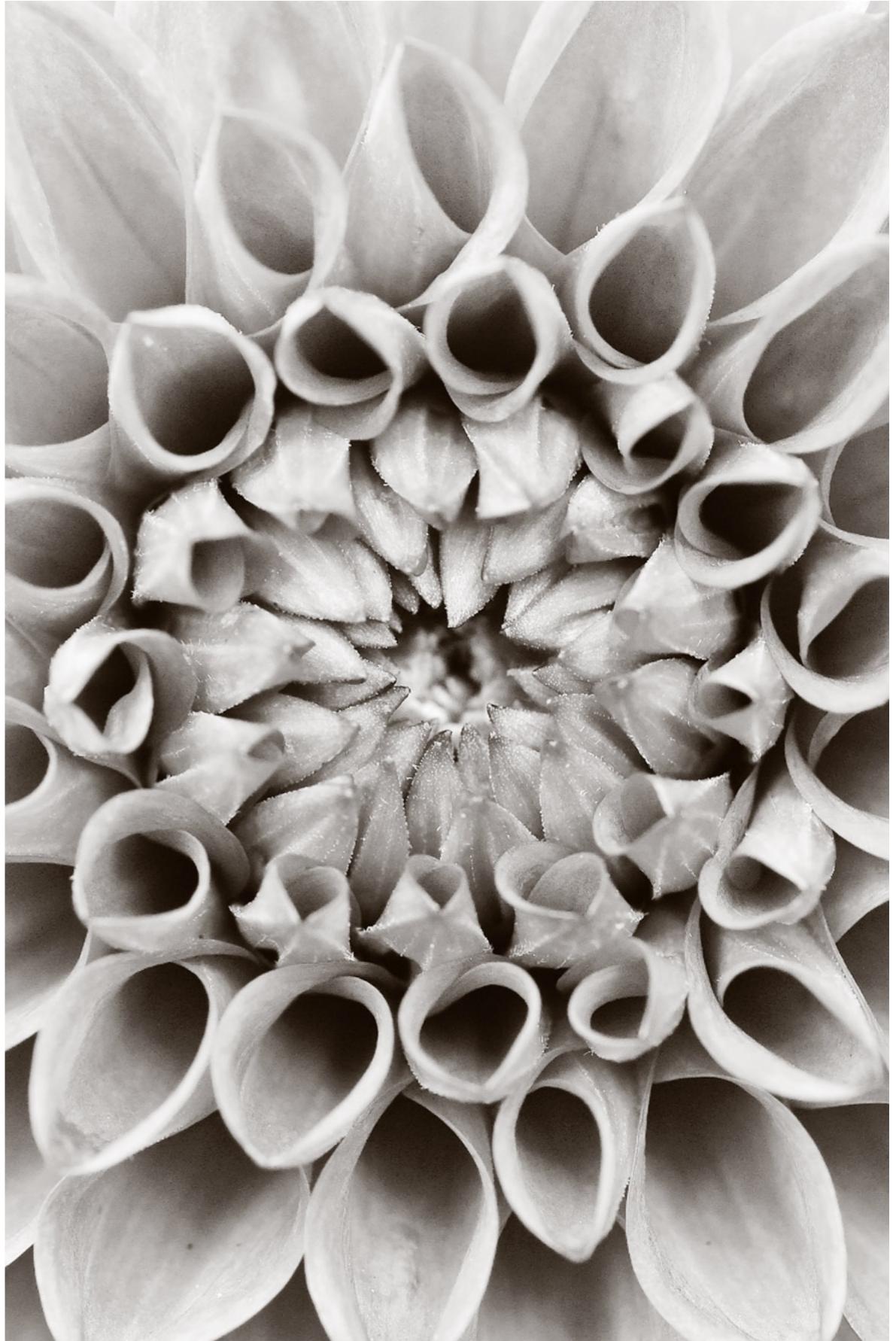
THE FRONTPANEL



LED	COLOR	FUNCTION
OSC	blue	OSC activity. Blinks when there is an incoming or outgoing OSC message. The brighter it lights the more OSC load If it's constantly lit very brightly then your tinyCCM crashed Reboot it (by powering it down)
PWR	green	When lit the tinyCCM is powered, when not your power supply is faulty. This works in all situations also while the tinyCCM is booting. This is your power on light and should be constantly on when using it
ERR	red	This is your error indicator: solid: no network found, no DHCP found or mDNS not possible 1 blink: Couldn't find the tinyApplication or the ping / keep-alive message has timed-out. 2 blinks: Cadac desk is connected meaning the tinyCCM sees at least one motorized fader 3 blinks: Both is missing the tinyApplication and the Cadac
FC	yellow	Whenever the tinyCCM speaks with the desk it will blink If it's constantly lit then your tinyCCM crashed. Reboot it (by powering it down)
OSC+ERR	blue/red	blinks in alternating fashion. This indicated that a firmware update has successfully been programmed. Your unit is ready after the next reboot.

Installation

Everything you need to make the system
up and running



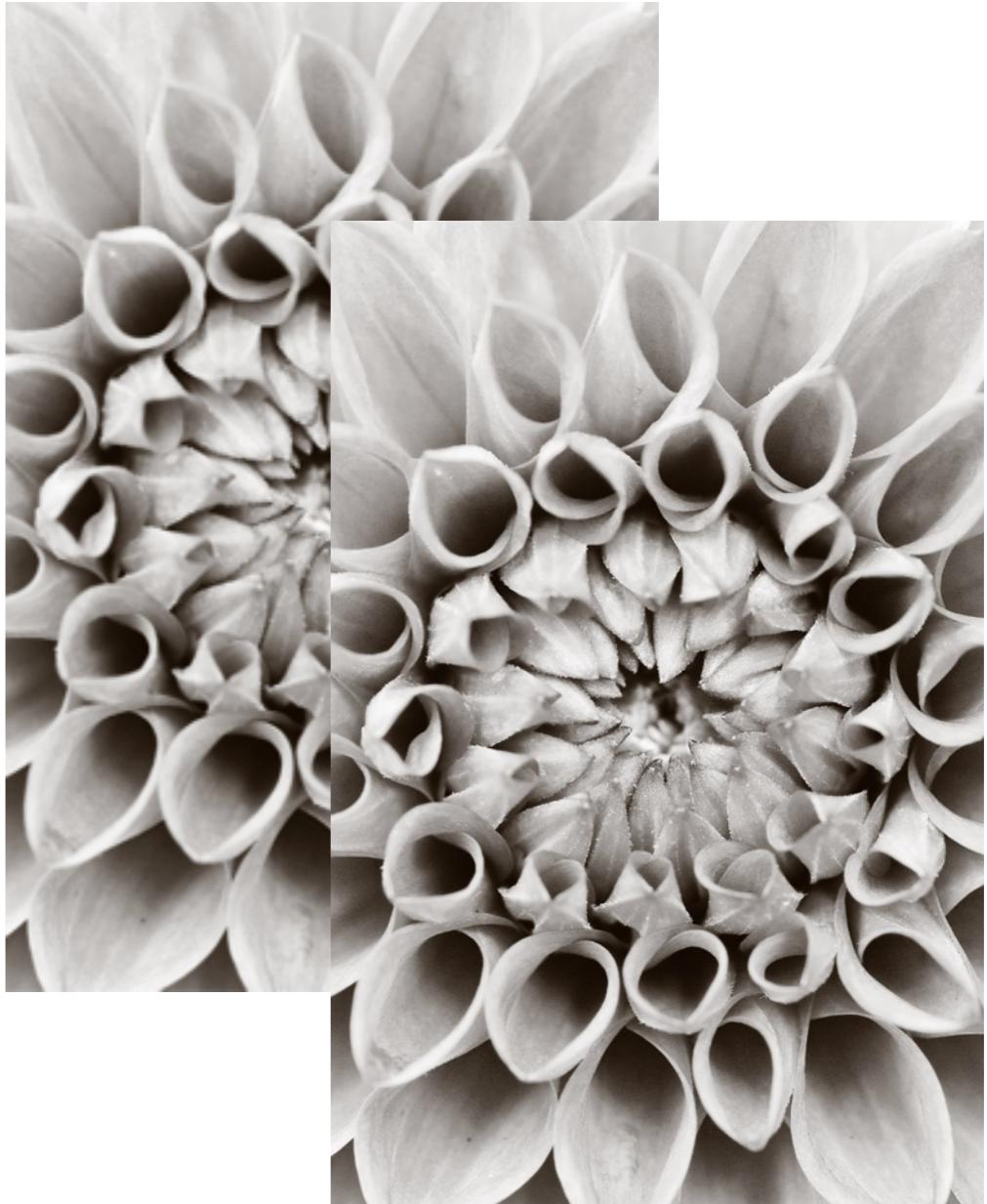
SECTION 1

Hardware Installation

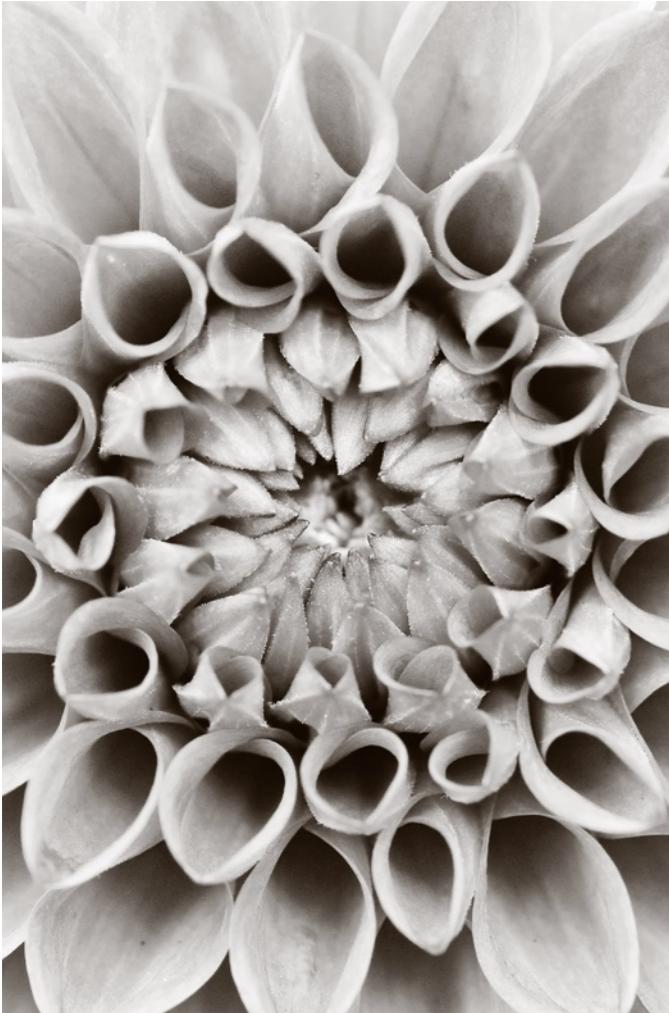
BILL OF MATERIAL CHECKLIST

1. tinyCCM Gateway
2. Fader Comms cable
3. Network cable
4. USB cable
5. USB memory stick
6. Four rubber feet
7. some tinyCCM M&M's

- Power off everything
- Connect the Fader Comms cable on both ends



- Connect the ethernet cable to your LAN switch or router. It has to be the same subnetwork as your Mac.
- Connect the usb cable to a usb power supply



- The tinyCCM indicates power (green LED) and after a short while the red ERR LED will start to blink (if not then consult the troubleshooting guide at the end)
- Power on your cadac desk. After about 20 seconds the red ERR LED will blink once every interval. This means the tinyCCM has found your cadac desk and did perform the mapping.

- If you want to place the tinyCCM on a sensitive surface, attach the four ribbon feet for protection. The enclosure also has been designed to have L-shape faceplate so you can mount it near the desk (that's why the Fader Comms cable is extra short)
- Take some tinyCCM M&Ms

SECTION 2

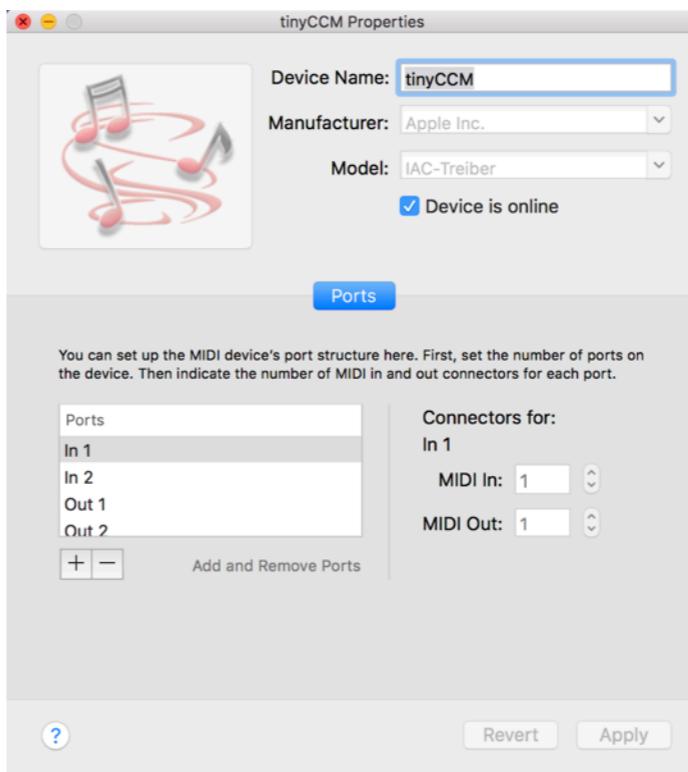
Software Installation

THINGS TO KEEP READY

1. A mac running macOS or OSX
2. your tinyCCM USB thumb drive
3. the license file that has been emailed to you
4. some tinyCCM M&M's

Preparing the IAC/MIDI system

- Open Audio MIDI Setup in Applications > Utilities
- Switch to MIDI view (Menu Window > Show MIDI Studio)
- Double-click the red IAC component



1) change the device name to „tinyCCM“

2) create as many IAC port pairs as needed and name them accordingly:

1-8ch: In 1, Out 1

9-16ch: In 2, Out 2

17-32ch: In 3, Out 3

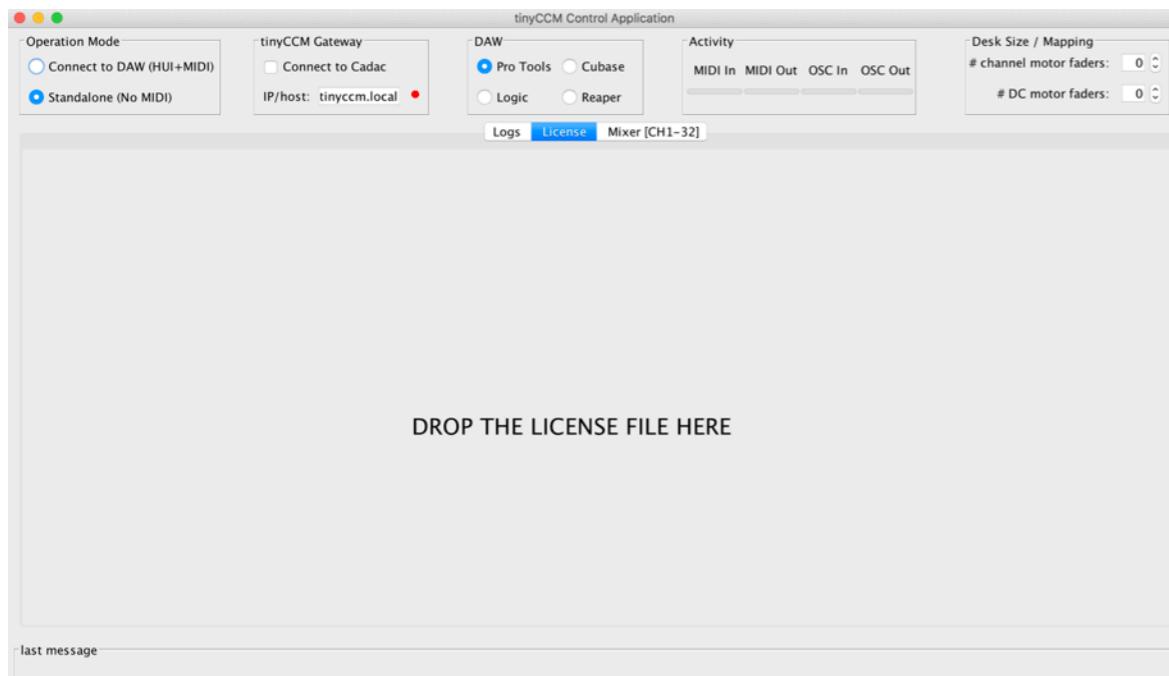
This picture shows a setup with 2 IAC port pairs for 16 channels this equals to two HUI instances.

Installing the tinyCCM Application

Plug in your tinyCCM USB thumb drive; it will be mounted automatically and appear on your Desktop. Open the volume and double click on the installer. Follow the steps. You will find the tinyCCM Application in your Applications folder. Create a shortcut to your Dock if you want.

Installing the tinyCCM Application license

- Open the tinyCCM Application so you can see the graphic user interface.
- Open the mail or the tinyCCM USB stick with the license file and drag it to a temporary folder (Desktop, Download etc...). From there drag the file into the tinyCCM Application window and let it drop. It will automatically verify and install the license. Everytime you do this the license will be overwritten.



That's it your one-time setup is completed

SECTION 3

Logic Pro X Setup

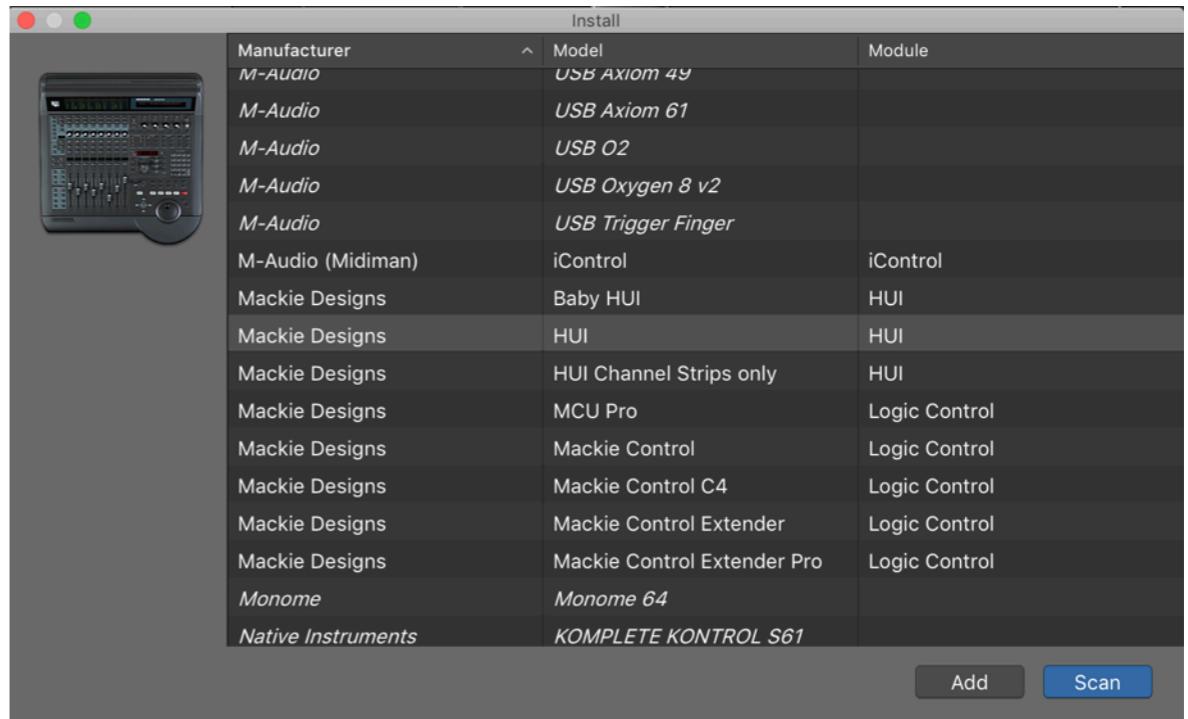
QUICK GUIDE

1. Create as many HUI instances you have motorized faders for and connect them to their corresponding IAC MIDI port
2. The first instance shall be of HUI type, all further instances shall be HUI Channel Strips only.
3. Make sure you did uncheck „Flash Mute and Solo buttons“
4. Reduce MIDI bandwidth to 60%
5. Enjoy nice HUI experience - ups I meant: Enjoy nice tinyCCM experience.

Open Logic X and goto Logic Pro X menu > Control Surfaces > Setup.

If you don't see this menu item enable it by checking Preferences > Advanced > Additional Options > Control Surfaces

Click on „New“ then „Install“ and find Mackie Design HUI



Add this device. Make sure there is only one instance of this HUI.

You now have 1 HUI instance that makes it for 8 faders. If you want more repeat these steps but this time add a HUI Channel Strips only. For a full 32 fader system you have to create three additional HUI Channel Strips only

This picture shows a system of one main HUI and a 2nd instance for a total of 16 channels



Select the main HUI and configure it so that Output Port is connected to „tinyCCM In 1“ and Input Port is connected to „tinyCCM Out 1“ - Note that the name logic crosses similar to the cross over when installing physical MIDI cables.

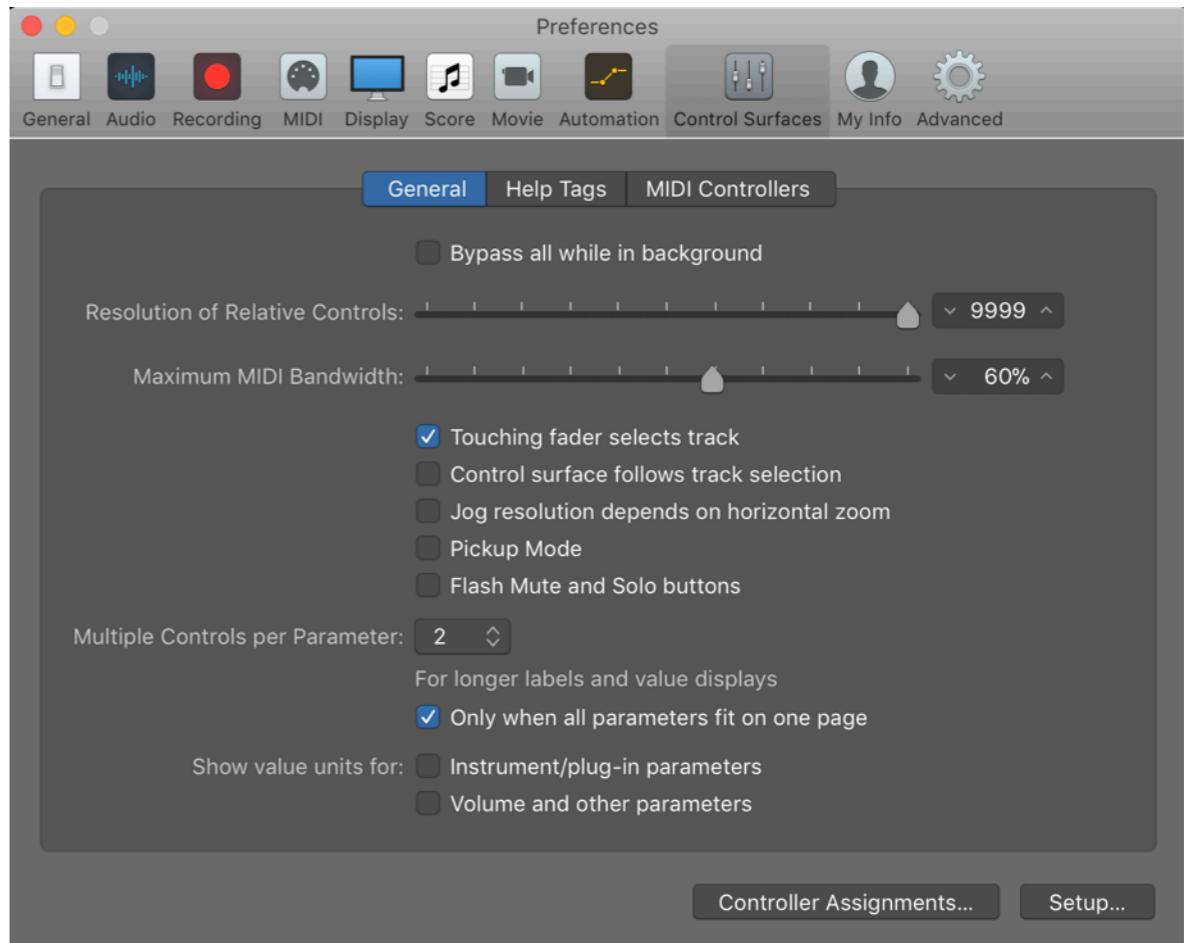
Do the same for the 2nd HUI instance which must be connected to tinyCCM In 2 and tinyCCM Out 2. Continue to do this until all your IAC MIDI pairs are used.

Here is a small table outlining the configuration:

FADER	HUI TYPE	IAC/MIDI PORT
1-8	HUI	Output Port => tinyCCM In 1 Input Port => tinyCCM Out 1
9-16	HUI Channel Strips only	Output Port => tinyCCM In 2 Input Port => tinyCCM Out 2
17-24	HUI Channel Strips only	Output Port => tinyCCM In 3 Input Port => tinyCCM Out 3
25-32	HUI Channel Strips only	Output Port => tinyCCM In 4 Input Port => tinyCCM Out 4

Back in the Control Surface Preferences Window set the Maximum MIDI Bandwidth to 60% and make sure „Flash

Mute and Solo buttons“ is unchecked. The other settings can be set according to your liking.



A note to „MIDI bandwidth“: Logic is the only DAW that has such a throttling mechanism and is a nice feature to balance accuracy vs latency. While IAC doesn't suffer from the slow MIDI speed and OSC sending is also very fast in theory 100% is the best. However if you have 32 tracks and they move all at the same time in a fast fashion the load is significant and messages might get buffered along the line. Buffering means delay. We have found a setting of 60% to be the best balance in case something is congested (which can also be your LAN).

Settings lower than 20% reduce this even further but fader movements are not smooth and will move in small steps. You can set this freely as long as you understand that this might become a balancing act.

Load the tinyCCM Test project which consists of a ramp figure on all tracks simultaneously. Let it play and check that all faders are at the same position at all times even when you randomly jog along the timeline as seen in the demonstration video. Small „correction“ of faders that are not at the absolute position is performed by the faders firmware most of the time a dirty driving systems can cause this. It is important to understand that even when the fader position is hindered the VCA level of the fader module is very precise - so the deviation is only on your eyesight not on your „ear sight“.

Move the first fader on your desk by hand and check that Logic Pro X does follow the movement in the mixer view. You might see that it sometimes stalls in this direction but it seems to be a Logic Pro X bug since we don't see this behavior on any other DAW. It can also be that we have to fine-tune the Logic Pro X fader curve better which we will do in a future update. Any jumps or steps in a smoothly performed fader movement on your written automation track can be manually be smoothed by using the DAW automation smoothing feature.

Do the same test with your ninth fader. Make sure it is indeed the ninth fader in Logic Pro X. Continue this test for all your HUI instances to check if they are banked correctly.

Logic automatically clusters multiple HUI instances and banks them correctly. Also nothing must be done when opening another session/project as Logic Pro X HUI implementation is well done and stable and remembers its settings across different sessions and launches of the DAW. Nice job Apple!

SECTION 4

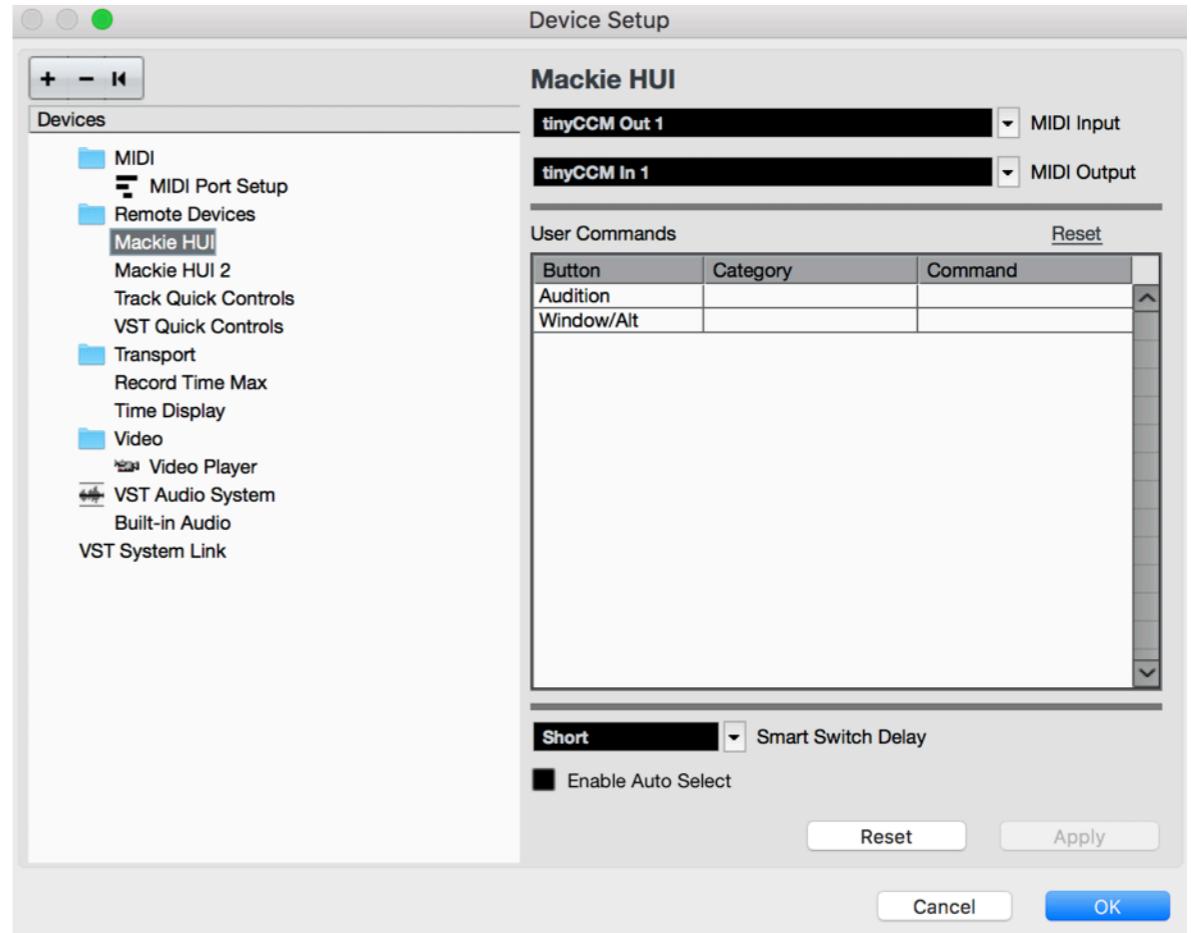
Cubase Setup

might work for Nuendo too

QUICK GUIDE

1. Create as many HUI instances you have motorized faders for and connect them to their corresponding IAC MIDI port
2. All instances shall be of HUI type
3. Remember: Steinberg didn't cluster the HUI instances so rebanking might be needed.
4. Enjoy nice HUI experience - ups I meant:
Enjoy nice tinyCCM experience.

Open Cubase and goto Devices > Device Setup > Setup under „Remote Devices“ click the Add (plus) button and add as many Mackie HUI instance you need. They'll get incrementally numbered.



This picture shows a system of two HUI instances for a total of 16 channels

After you added all HUI instances assign the correct IAC/MIDI port to each instance.

Select the first HUI (it's the one without a number and at the top of the list) and configure it so that MIDI Output is

connected to „tinyCCM In 1“ and MIDI Input is connected to „tinyCCM Out 1“ - Note that the name logic crosses similar to the cross over when installing physical MIDI cables.

Do the same for the 2nd HUI instance which must be connected to tinyCCM In 2 and tinyCCM Out 2. Continue to do this until all your IAC MIDI pairs are used.

FADER	DEVICE	IAC/MIDI PORT
1-8	Mackie HUI	Output Port => tinyCCM In 1 Input Port => tinyCCM Out 1
9-16	Mackie HUI 2	Output Port => tinyCCM In 2 Input Port => tinyCCM Out 2
17-24	Mackie HUI 3	Output Port => tinyCCM In 3 Input Port => tinyCCM Out 3
25-32	Mackie HUI 4	Output Port => tinyCCM In 4 Input Port => tinyCCM Out 4

Here is a small table outlining the configuration:

no additional configuration is needed.

Steinberg decided to not cluster multiple HUI instances to make it a one cascaded surface. Instead every instance get the same 8 faders and this means whenever you see that groups of 8 faders mirror each other rebanking is needed. On the original Mackie HUI console there are BANK UP and BANK DOWN buttons and users that had more than one HUI in a cascaded clustered array had to bank the second instance one

times up (to the right) in order to match fader 9-16 to this instance.

Because the tinyCCM Application knows which IAC MIDI port should be which fader group it can perform the banking for you. We call this auto-banking. It will move each instance to its correct place. Autobanking is done whenever the MIDI system of the tinyCCM Application is opened (or reopened).

As nice as this may sound there is still a problem with this. If you close the session and open another Cubase project the tinyCCM Application doesn't know about this and the banking of the first project will not be applied to the second project. Should this happen you have to rebank manually and there is a menu command for this in the tinyCCM Application.

SECTION 5

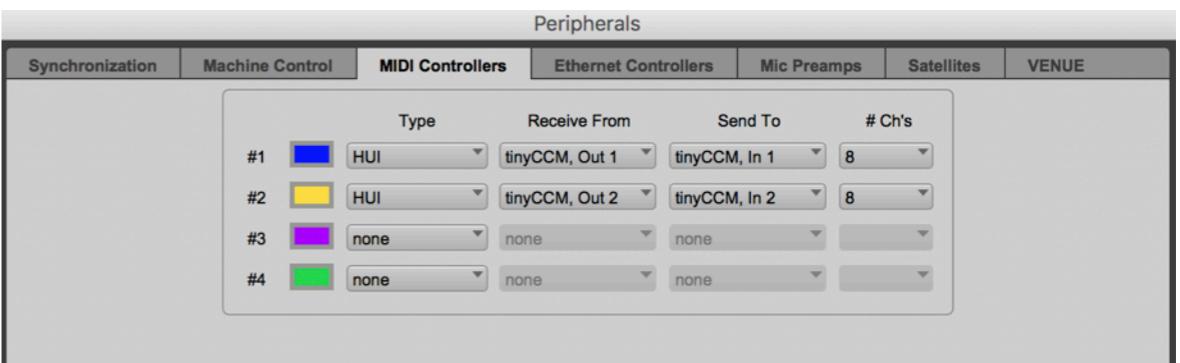
Pro Tools Setup

doesn't work for Pro Tools First

QUICK GUIDE

1. Create as many HUI instances you have motorized faders for and connect them to their corresponding IAC MIDI port.
2. The maximum is four instances (32 faders)
3. All instances shall be of HUI type

Open Pro Tools and goto Setup > Peripherals > MIDI Controllers. You will see a maximum of 4 available slots.



This picture shows a system of two HUI instances for a total of 16 channels

For the slot #1 choose Type > HUI and configure it so that Receive From is connected to „tinyCCM, Out 1“ and Send To is connected to „tinyCCM, In 1“

Do the same for the 2nd HUI instance which must be connected to tinyCCM In 2 and tinyCCM Out 2.

Here is a small table outlining the configuration:

FADER	TYPE	IAC/MIDI PORT
1-8	HUI	Receive From => tinyCCM, Out 1 Send To => tinyCCM, In 1
9-16	HUI	Receive From => tinyCCM, Out 2 Send To => tinyCCM, In 2
17-24	HUI	Receive From => tinyCCM, Out 3 Send To => tinyCCM, In 3
25-32	HUI	Receive From => tinyCCM, Out 4 Send To => tinyCCM, In 4

Continue to do this until all your IAC MIDI pairs are used.

no additional configuration is needed.

Pro Tools clusters all HUI instances to make them into a continuous desk up to a maximum of 32 channels. No auto-banking is needed.

Pro Tools let the Mute LEDs flash when in solo mode in order to prevent chopped audio the tinyCCM Application has to suppress this. This is automatically done when you select Pro Tools as your DAW type in the tinyCCM Application user interface.

SECTION 6

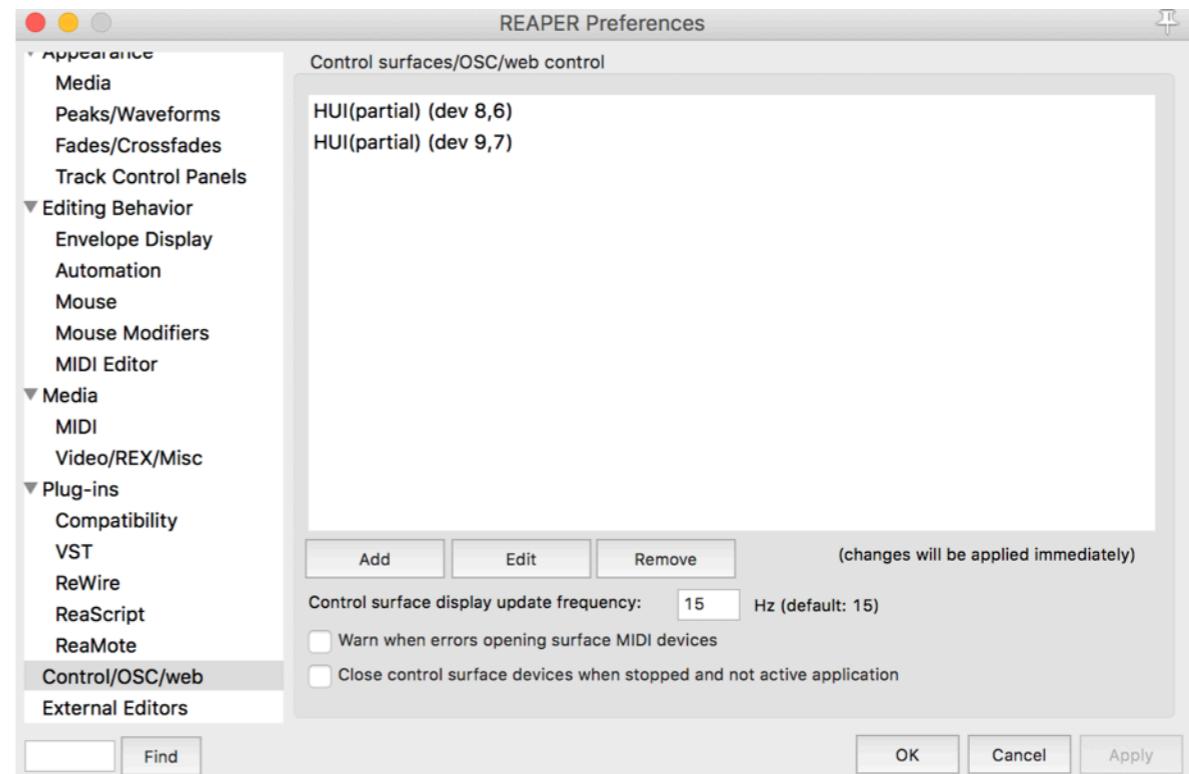
Reaper Setup

QUICK GUIDE

1. Create as many HUI instances you have motorized faders for and connect them to their corresponding IAC MIDI port.
2. Set the offset correctly

All instances shall be of HUI (partial) type

Open Reaper and goto Reaper Preferences scroll down the list to see Control/OSC/web:

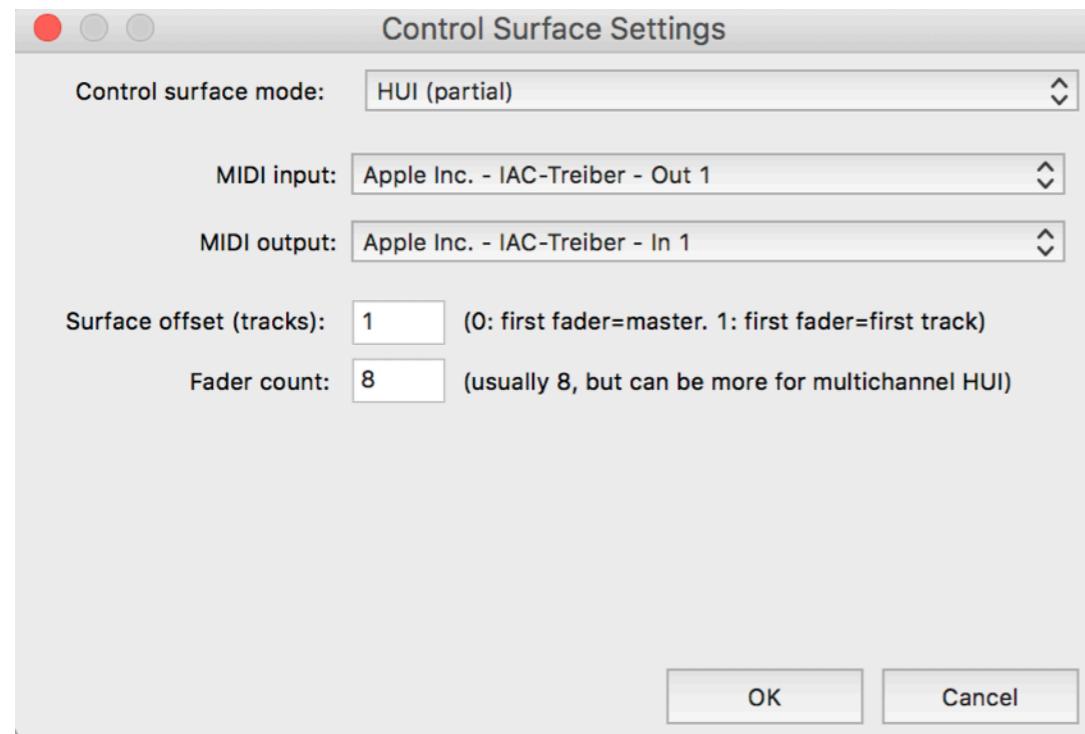


This picture shows a system of two HUI instances for a total of 16 channels

To add the first HUI instance click „Add“ and choose „HUI (partial)“.

Continue to do this until all your instances are used.

Click „Edit“ for each of the instances and configure it according to the following table:



FADER	TYPE	IAC/MIDI PORT
1-8	HUI (partial)	MIDI input => Apple Inc - IAC-Driver- Out 1 MIDI output => Apple Inc - IAC-Driver- In 1 Surface offset (tracks): 1 Fader count: 8
9-16	HUI (partial)	MIDI input => Apple Inc - IAC-Driver- Out 2 MIDI output => Apple Inc - IAC-Driver- In 2 Surface offset (tracks): 9 Fader count: 8
17-24	HUI (partial)	MIDI input => Apple Inc - IAC-Driver- Out 3 MIDI output => Apple Inc - IAC-Driver- In 3 Surface offset (tracks): 17 Fader count: 8
25-32	HUI (partial)	MIDI input => Apple Inc - IAC-Driver- Out 4 MIDI output => Apple Inc - IAC-Driver- In 4 Surface offset (tracks): 25 Fader count: 8

no additional configuration is needed.

Using the correct offset settings Reaper clusters all HUI instances to make them into a continuous desk up to a maximum of 32 channels. No auto-banking is needed.

Testing The Setup

QUICK GUIDE

1. Check the reported number of found faders on your desk
2. Open the supplied test session file for your DAW under test.
3. Let it play and validate that each fader moves according to the automation curve
4. Test the other direction by moving any fader on your desk and check that only the corresponding DAW fader moves according to your movement

Power on everything and connect both sides of the tinyCCM Gateway and the tinyCCM Application. Check that your tinyCCM has found the correct number of fader channels

Load the tinyCCM Test project which consists of a ramp figure on all tracks simultaneously. Let it play and check that all faders are at the same position at all times even when you randomly jog along the timeline as seen in the demonstration video. Small „correction“ of faders that are not at the absolute position is performed by the faders firmware most of the time a dirty driving systems can cause this. It is important to understand that even when the fader position is hindered the VCA level of the fader module is very precise - so the deviation is only on your eyesight not on your „ear sight“.

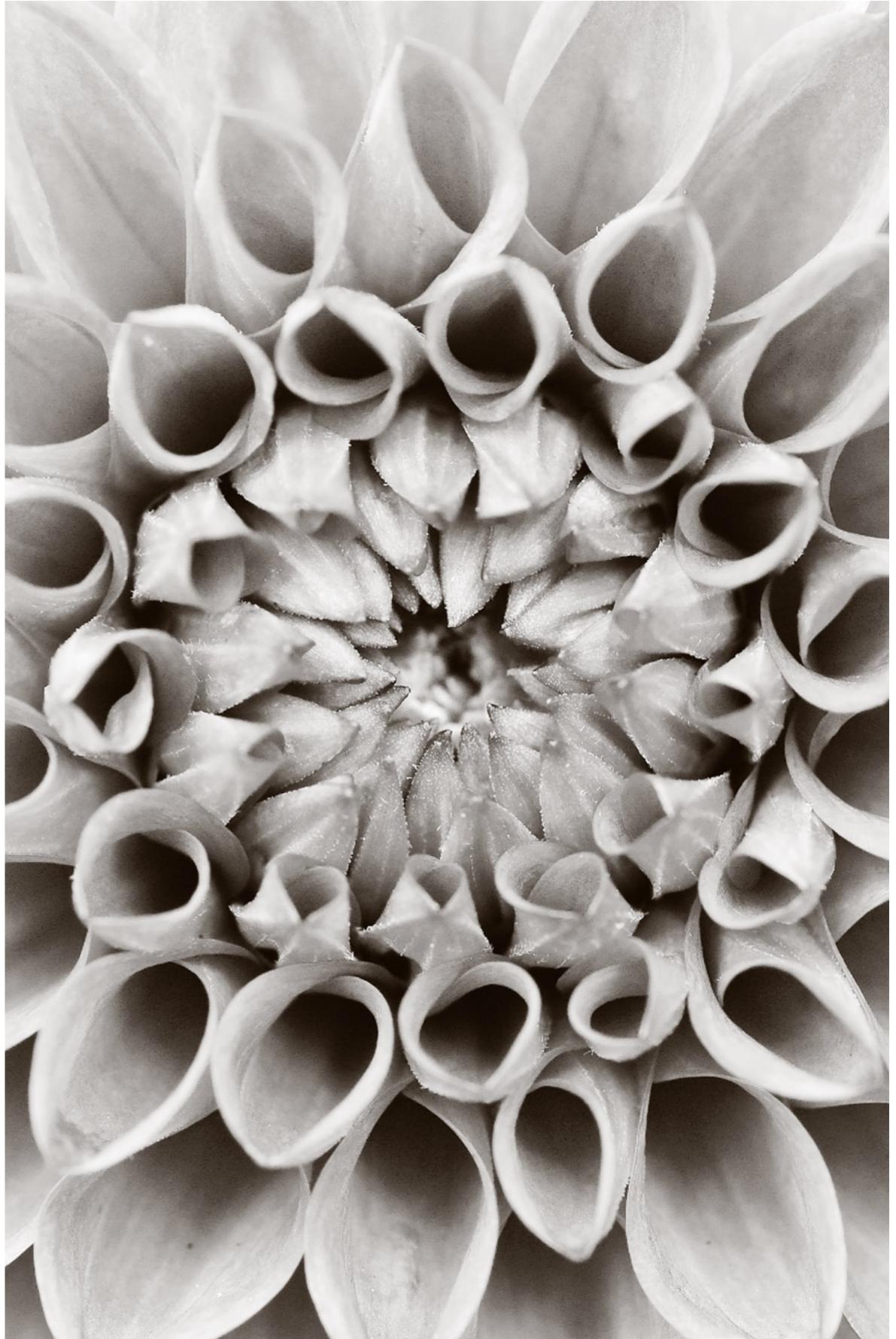
Move the first fader on your desk by hand and check that your DAW does follow the movement in the mixer view. You might see that it sometimes stalls in this direction but it seems to be a Logic Pro X bug since we don't see this behavior on any other DAW. It can also be that we have to fine-tune the Logic Pro X fader curve better which we will do in a future update. Any jumps or steps in a smoothly performed fader movement on your written automation track can be manually be smoothed by using the DAW automation smoothing feature.

Do the same test with the ninth fader. Make sure it is indeed the ninth fader in Logic Pro X. Continue this test for all your HUI instances to check if they are banked correctly.

Don't forget the mutes and maybe do a solo.

Operation

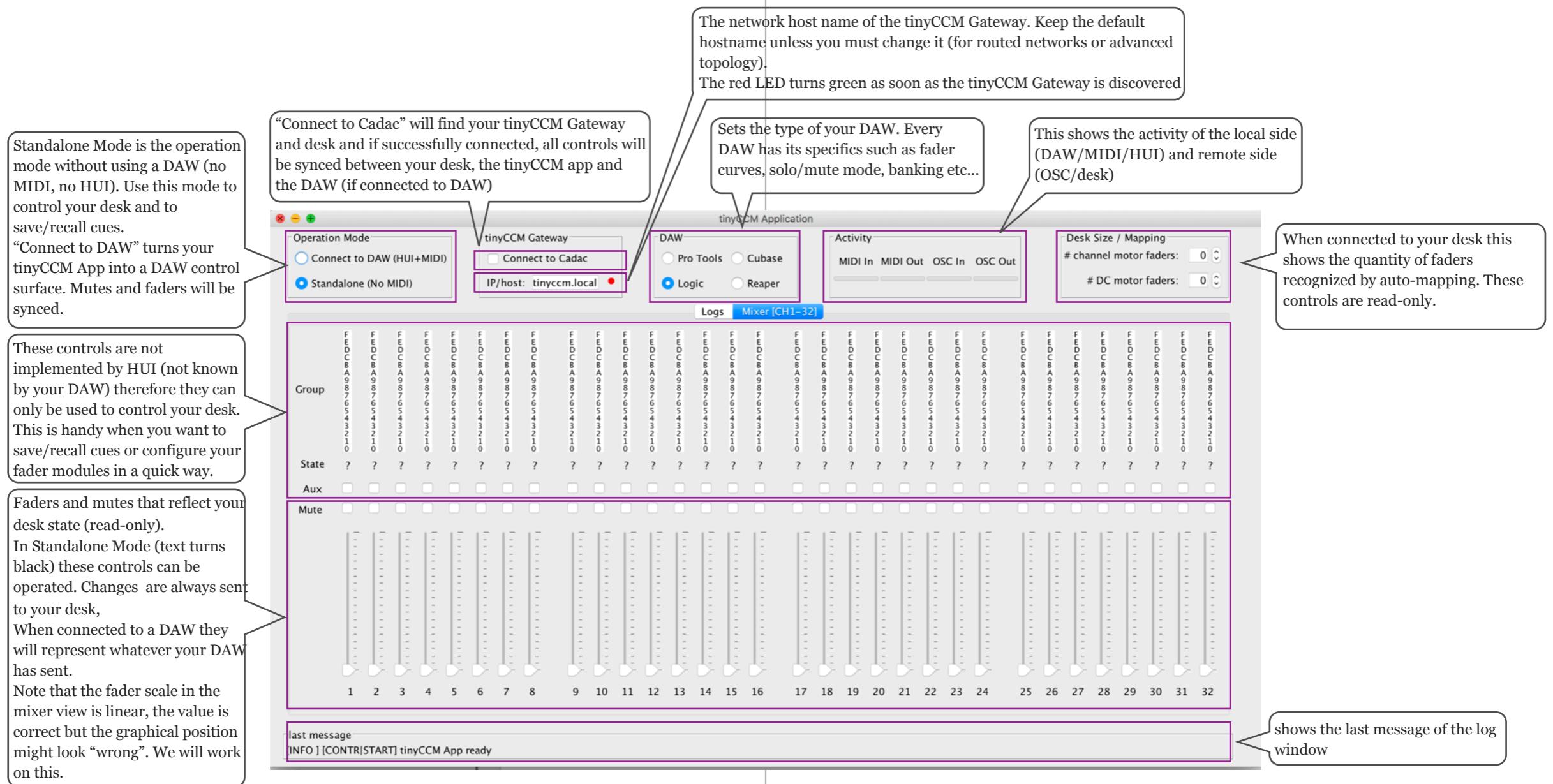
This chapter describes what you can do after the installation



SECTION 1

Using The tinyCCM Application (Main View)

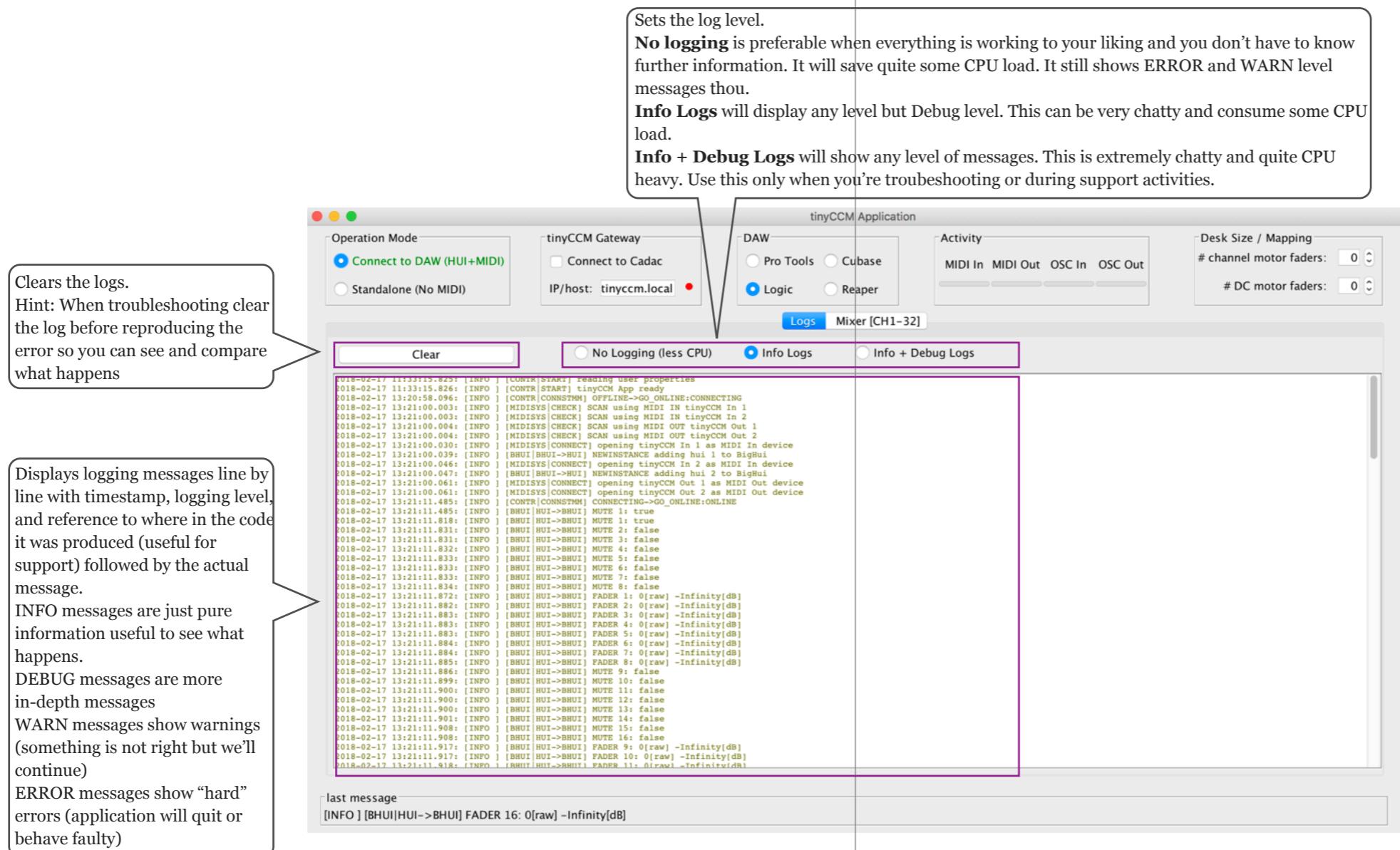
The graphic user interface and its components



SECTION 2

Using The tinyCCM Application (Logs View)

The graphic user interface and its components



SECTION 3

Using The tinyCCM Application (Menu)

MENU	MENU ITEM	SHORTCUT	DESCRIPTION
File	Load Cue...	⌘L	Loads a previously creates cue from a selected file.
	Save Cue As...	⌘S	Saves the current mixer (all controls in the mixer view) as a cue file.
	Quit	⌘Q	Quit the application
Functions	Rebank HUI Instances	⌘B	Some DAW do not rebank the HUI instances automatically, if needs be you have to rebank manually.
	Reset HUI (experimental)	⌘R	Tells the DAW to reset the HUI control surface. So far most DAW just ignores this command. ;-(
	Cadac -> App Upload	⌘↑	Makes the mixer window in the tinyCCM app reflect the desk's fader state
	App -> Cadac Download	⌘↓	Makes the desk's fader state reflect the mixer window state in the tinyCCM app (if connected to DAW this is identical with the DAW mixer state)
	Reset Mixer	0 (zero)	very handy shortcut to nullify every fader module. Note that this is being sent to the desk in slow speed
Settings	Auto Connect		If selected the tinyCCM app tries to automatically connect to the tinyCCM Gateway hardware whenever discovered in the network
	Mute Roundtrip Time		As for the time of writing this manual keep the settings to "Mid" unless instructed to change this.
Help	About	⌘A	Useful to see the exact tinyCCM app version and credits

SECTION 4

Using The tinyCCM Application (tray icon)

When the tinyCCM app has started you will see a little “tray icon” in the menu bar. It shows the connection state as explained below. If you close the application window (press the little red x-button) you can restore the window by pressing on the tray icon. This is to reduce screen clutter.



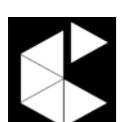
Both sides (local and remote) are disconnected.
This means the DAW and the Cadac is not connected.



Local side (DAW) is disconnected.
tinyCCM app runs in Standalone Mode.



Remote side (Cadac) is disconnected.
Press “Connect to Cadac” to reconnect.



Both sides (local and remote) are connected.
This is your target and typical connection state



Something is running in the background - the tinyCCM app is occupied with itself. You should never see this.

SECTION 5

Typical everyday startup

YOUR EVERDAY STARTUP PROCEDURE IS

1. Make sure all hardware is powered and connected
2. Start the tinyCCM app
3. Press “Connect to Cadac” - after about 10 seconds your desk is “online”.
Bonus tip: not needed when “Auto connect” is selected in the Functions menu
4. Verify reported / found fader modules
5. Verify that the DAW Type is correct
6. Press “Connect to DAW”
7. Load your session file in the DAW
8. If everything is well, close the window of the tinyCCM app by pressing the little red x (close window).

Specification

SOFTWARE REQUIREMENTS

macOS/OSX that runs Java 1.7.

Note that the installer is self-containing it brings its own JRE (java version 1.8). The installed java version managed by OSX updates should have no impact to this and it not used.

Recommended system configuration is:

OSX Mavericks or higher, 4 GB RAM or more, 4 reported cores (2 hyperthreaded cpus) or more, 300 MB disk space and a Ethernet network interface.

We do not recommend Wifi. The reason is latency and more importantly OSC is UDP which means packet can arrive at any arbitrary order and the tinyCCM Gateway hardware.

The tinyCCM app needs 200-400 MB RAM.

NETWORK REQUIREMENTS

Local Area Network with a DHCP server (typically included in your broadband router)

Connect the tinyCCM Gateway hardware to a network switch or router. Do not connect the tinyCCM directly to your computer without a DHCP server.

We do not recommend Wifi. The reason is latency and more importantly OSC uses UDP which means packet can arrive at any arbitrary order and the tinyCCM Gateway hardware.

Both the computer and the tinyCCM Gateway hardware must be on the same subnet and broadcast domain. Otherwise you have to assign a static IP to your tinyCCM (a reserved IP address in your router) and enter this IP in the hostname field of your tinyCCM app instead of the default hostname.

TINYCCM HARDWARE REQUIREMENTS

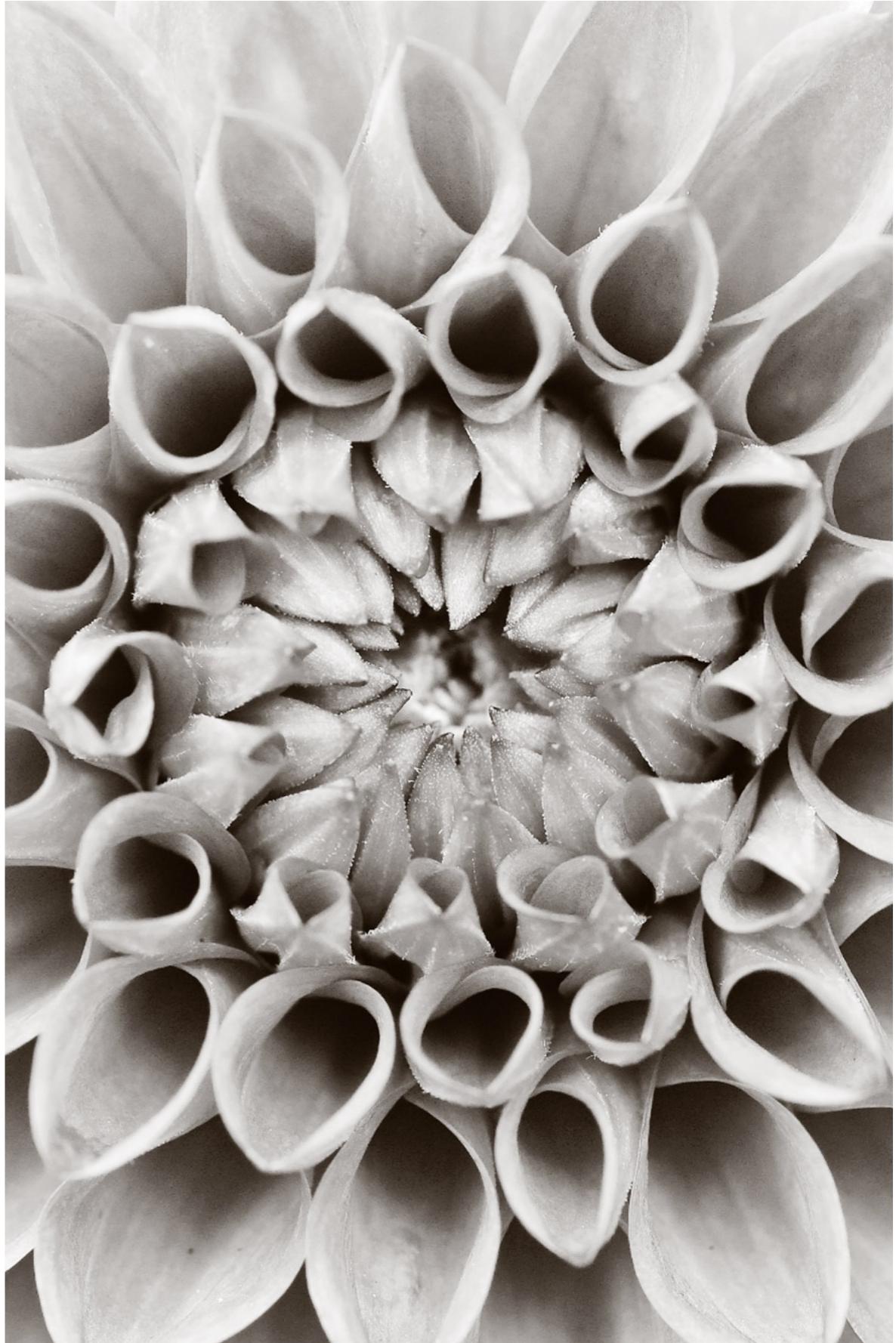
USB (micro type B) power supply or using the supplied USB cable a USB Type A connector that powers with 5V 0.5A. Higher current rating is supported. Do not power the tinyCCM hardware directly from your computer (with the exception of firmware updates).

Only use the supplied DSUB-type cable to fadercomms. Do not extend this cable. If you have multiple Cadac frames consult us before making your own cable.

- Power supply: USB bus power
- Typical power consumption: 2 Watts
- Dimensions (WxHxD): 112 x 27 x 83 mm (4.4" x 1.1" x 3.3")
- Weight: 273 g (0.6 lbs)
- Temperature range: +5° up to +50° Celsius (41° F up to 122°F)
- Relative humidity: < 75%, non condensing

Software Updates

Both



Firmware Updates

The tinyCCM Gateway hardware is internally based on programmable logic. By re-programming both function and behaviour of the unit can be changed at any time.

At the time of writing this manual the tinyCCM Gateway hardware is shipped with firmware 11. The firmware version is displayed in the logging window (INFO) upon successful connection as well as the Firmware Update Tool.

Firmware 10: Initial release.

Firmware 11: boot robustness and performance improvements

UPDATING THE FIRMWARE

There is a tinyCCMFW Tool on the USB stick that was shipped together with the tinyCCM. Install this tool and follow the instructions. In addition to this tool you have to keep a firmware file ready which has a *.hex file extension. The original firmware at the time of shipping is also supplied on the USB stick. Further updates will be sent by email or dropbox link to known customers