



Control Units
MCU, ECU, HCU, Logger Dock

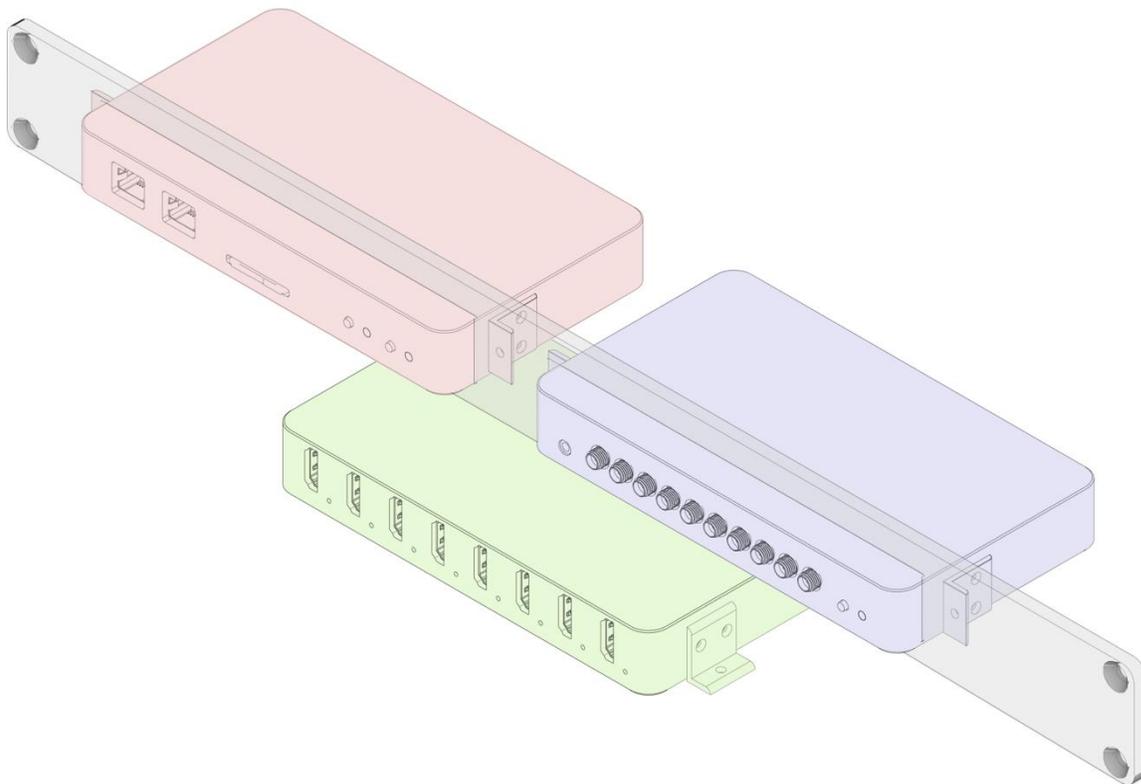


Table of Contents

Introduction	3
Hardware Setup	4
Installing Trodes.....	5
Connection Bandwidth: USB vs. Ethernet.....	6
USB Setup	7
Ethernet Setup	8
Main Control Unit (MCU).....	10
Introduction	10
MCU Specifications.....	11
Control and Indicators.....	11
Inputs/Outputs.....	12
Recording with the MCU	13
Environmental Control Unit (ECU).....	14
Introduction	14
ECU Specifications	14
Unit Front: Control and Indicator	15
Unit Rear: Inputs and Outputs.....	16
Headstage Combiner Unit (HCU)	18
Introduction	18
HCU Specifications	18
Control and Indicators.....	19
Logger Dock	21
Introduction	21
Logger Dock Primary Functions.....	21
Logger Dock Specifications.....	22
Control and Indicators.....	22
Docking and Charging	24
LED Indicators	25
Initiating Untethered Recordings.....	25
Exporting headstage Data.....	25
Additional Resources	26



Introduction

SpikeGadgets' modular control units for electrophysiology and environmental recording and control offer modular and flexible options to meet the needs of cutting-edge neuroscience experiments, offering high channel count capabilities (up to 1024 channels) and real-time and low-latency environmental control.

Main Control Unit (MCU): The MCU is the central processor and controller of the SpikeGadgets recording system. The MCU offers high channel count neural recording capabilities (up to 1024 channels), 6 digital inputs for environmental and experimental event signals, and wireless radio frequency (RF) control for untethered neural recording with SpikeGadgets data logger headstages.

Environmental Control Unit (ECU): The ECU works in conjunction with the MCU and Trodes StateScript scripting application to control the experimental environment while also recording flags and/or signals from both analog and digital devices. Inputs to the ECU are sampled at the same rate and synchronized with neural data. The ECU also offers low-latency closed loop environmental control with up to 64 digital inputs and outputs (32 of each) and 16 analog inputs and outputs (8 of each).

Headstage Combiner Unit (HCU): The HCU streamlines tethered experiments where high-channel count from multiple headstages is key. The HCU can connect to 8 headstages simultaneously, relaying up to 1024 channels of neural data to the MCU. The HCU is optimized for head-fixed applications.

Logger Dock: The Logger Dock acts as a charging and data transfer dock for the ML32 data logging headstage and supports untethered recording with any SpikeGadgets headstage. The Dock is available in 2 configurations: One configuration supports the ML32's internal battery and data storage, while the other supports headstages with external batteries and storage. Both configurations charge batteries and transfer data to the local computer while also acting as a control unit capable of initiating and terminating untethered experiments.

Hardware Setup

The most basic hardware configuration uses the Main Control Unit (MCU) paired with one data-logging headstage. With this simple setup, up to 1024 channels of neural data can be streamed to a local computer with 6 additional channels dedicated to receiving digital signals/TTLs from external devices via the MCU's digital input channels.

For experiments requiring environmental control, analog inputs and outputs (I/Os) or a large number of digital I/Os, the Environmental Control Unit (ECU) can be added for an additional 80 I/Os and low-latency control in hardware.

For head-fixed applications requiring multiple headstage, the HCU is added; allowing for simultaneously recording 1024 channels of neural data from up to 8 headstages at a time.

Power, Ground & Data: All control units function using 5V power and should only be powered using the power adaptors shipped with the unit. This is because some adaptors can introduce significant recording noise.

Ground from all connected SpikeGadgets hardware is routed through the MCU, which has a separate ground line (2mm banana plug) for connecting to the central ground shared by all components of the recording and behavioral setup. For best results, all devices in the experimental setup should be connected to this central ground.

The MCU connects to the local computer via either USB 2.0 or gigabit Ethernet. For Windows users that 1) are recording fewer than 128 channels, and 2) do not have sub-millisecond control latency requirements, USB is recommended for easy setup. For all other applications, connecting via gigabit Ethernet is recommended; detailed setup instructions are included below.

Computer Setup

All SpikeGadgets hardware is controlled using the Trodes acquisition software suite running on your local computer, with versions available for Windows, Mac, and Linux (Ubuntu). As mentioned above, the MCU communicates with your local machine using either USB or Ethernet, with setup dependent on your operating system.

Please visit the [Trodes Wiki](#) for more information about using Trodes and [getting started with recording](#).

To download the latest version of Trodes, please visit:
<https://bitbucket.org/mkarlsson/trodes>

Installing Trodes

When using USB to connect the MCU to your local machine, the FTD2XX Driver must be installed for Trodes to function properly. The FTD2XX Driver can be downloaded [here](#). Additional details can be found below about [USB](#) and [Ethernet](#) setup.

Windows

1. Install the FTD2XX Driver. For easy installation, use the setup executable provided on the FTD2XX Driver page. A link to the executable can be found [here](#) as well as in the Comments column of the Windows entry on the driver page.
2. Download the latest version of Trodes [here](#).
3. Uncompress the downloaded Trodes folder and move it to the desired location on your machine.
4. Open the downloaded and uncompressed Trodes folder and run the included vc_redist.x64.exe program. This will install the Microsoft plug-ins that Trodes requires.
5. Open Trodes by double-clicking Trodes.exe.

Linux (Ubuntu)

1. Install the FTD2XX Driver. Follow the instructions in the first section of the README doc. A link to this doc can be found [here](#), as well as in the Comments column of the Linux entry on the driver page.
2. Trodes requires your graphics card to support OpenGL. To install the correct driver for your card, type “sudo ubuntu-drivers autoinstall” into the terminal.
3. Download the latest version of Trodes [here](#).
4. Uncompress the downloaded Trodes folder and move it to the desired location on your machine.
5. If you are running a version of Ubuntu other than 20.04, you may need to install additional libraries using sudo apt-get commands. For this reason, we recommend using Ubuntu 20.04.
6. Confirm that Trodes is granted the correct permissions by right-clicking Trodes, selecting ‘Properties’ and navigating to the ‘Permissions’ tab. Make sure that ‘Allow executing file as program’ is checked.
7. Open Trodes by double-clicking or run from the terminal by typing ./Trodes from the install folder.

MacOS

Due to changing system architecture and publishing requirements, MacOS support for Trodes is currently limited. Trodes is compatible with Intel-based Mac systems, however, the installation process is not always straightforward. For this reason, if you would like to run the newest release of Trodes on your Intel-based Mac, please reaching out to us directly via support@SpikeGadgets.com for more information. Below are instructions for installing Trodes version 2.2.1 on older Intel-based mac systems.

1. Install the FTD2XX Driver. Run the D2xxHelper program to prevent macOS from claiming the device as a serial port. A link to the helper program can be found [here](#) as well as in the Comments column of the Mac OS X entry on the driver page.
2. Download the latest version of Trodes [here](#).
3. Uncompress the downloaded Trodes folder and move it to the desired location on your machine.
4. Because Trodes is not officially registered with Apple, you may need to let the OS know that the programs contained in the Trodes suite are safe to run. To do this, right-click the Trodes application you would like to run and select “Open.” For MacOS 10.15 or later, a dialog will appear when you do this, warning you that the program is not registered and cannot be opened. Typically, the first time you do this, you will not be given the option to override this. If you open the program a second time, a new button should appear that will allow you to open the program.

Once you have opened Trodes, you will see a message stating that Trodes needs access to the installation program; you must allow this for Trodes to function. This should only need to be done once per installation.

NOTE: for earlier versions of MacOS, you can simply navigate to Settings and change your “Allow apps downloaded from” security setting from “Mac App Store and identified developers” to “Anywhere.”

5. Repeat the procedure detailed in step 4 for the following applications within the Trodes Installation folder: cameraModule.app, DataLoggerGUI.app, stateScript.app and trodesexport.app

Connection Bandwidth: USB vs. Ethernet

The MCU can be connected to your computer using either USB or ethernet. When using USB to connect between the MCU and computer, the data rate is typically limited to 28 - 30 MB/s. If recording bandwidth exceeds this range the number of dropped packets will increase. For experiments requiring high channel count ethernet is recommended as it can handle > 100 MB/s when connected to a sufficiently powerful computer with a good ethernet card.

For reference, the data throughput when recording 384 channels of data from 1 Neuropixels probe @ 30Khz is ~25 MB/s. This means USB is sufficient for recording from one probe, but ethernet is required for multiprobe Neuropixels recordings.

USB Setup

USB connections to the MCU are often convenient but have greater bandwidth limitations than Ethernet. USB is recommended when sub-millisecond control latency is required while recording fewer than 128 channels. Ethernet is recommended for experiments requiring higher channel counts and sub-millisecond control latency.

Windows

Once the FDT2XX Driver has been installed, no additional setup is not required for Windows USB connections.

Linux (Ubuntu 20.04)

To connect to the MCU and ECU via USB, Ubuntu must be told how to handle the SpikeGadgets hardware that is plugged in. This is done by copying the SpikeGadgets udev rules to your machine, and only needs to be done once. To do this, run the following command (all on one line) in the terminal:

```
sudo wget https://bitbucket.org/mkarlsson/trodes/downloads/spikegadgets.rules -O /etc/udev/rules.d/spikegadgets.rules
```

The udev rules will be applied the next time the hardware is connected. If the hardware is already connected, simply unplug and replug the USB connection. The .rules file is also distributed with the precompiled binary (in Resources/SetupHelp), and in the source repository (in Resources/Linux), providing an alternative to wget if desired.

MacOS

The D2xxHelper program may be needed to prevent macOS from claiming the device as a serial port. No additional setup is required once the Driver has been installed.

Ethernet Setup

Connecting to the MCU via Ethernet is recommended for applications requiring high channel count and/or sub-millisecond control latency. Below are the initial configuration changes required before Ethernet can be used to connect to the MCU.

1. A dedicated Ethernet port is needed. This port should be configured for IPv4 with the following address and subnet:

Address: 192.168.0.2

Subnet: 255.255.255.0

2. The maximum transmission unit (MTU) should also be set to 9000 or greater. A guide for how to change this value on Windows and Linux systems can be found [here](#).

3. To prevent excessive packet drops, Ethernet performance must be maximized by increasing the operating systems UDP buffers. How this is done depends on the operating system:

Windows:

To adjust buffer size, edit (or create) the following DWORD registry keys using the registry editor, then restart Windows. This can be found using the Windows Start Menu search (run) function.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Afd\Parameters]
DefaultReceiveWindow (REG_DWORD) = 8384512 (decimal)
DefaultSendWindow (REG_DWORD) = 8384512 (decimal)
```

Linux (Ubuntu 20.04):

To adjust buffer size, edit /etc/sysctl.conf (as root) and add the following lines:

```
net.core.rmem_max=8388607
net.core.rmem_default=8388607
net.core.wmem_max=8388607
net.core.wmem_default=8388607
```

MacOS:

To adjust buffer size, edit /etc/sysctl.conf (as root) and add the following lines:

```
net.inet.udp.recvspace = 8388607
net.inet.udp.sendspace = 8388607
```

4. Firewalls may block all or portions of the traffic between the computer and MCU. As such, they must be disabled or adjusted to allow unhindered Ethernet traffic between the computer and MCU.

In situations where a firewall is required, public network access must be given to the connection to Trodes. This can be done by giving full permissions to Trodes, or by selectively allowing traffic on the ports used by Trodes (8100 and 8200). For Windows we recommend using the procedure outlined below.

Windows Firewall reset:

First, completely reset your Windows firewall settings, then allow Windows to automatically “allow” Trodes access to the MCU when connecting for the first time. This can be done as follows:

1. Open Windows “Firewall & network protection,” and select “restore firewalls to defaults.” A pop-up window will open, where “Restore defaults” can be selected. When a warning pops up, select “yes.”
2. Run Trodes as Administrator, open a workspace, connect to ethernet by selecting Source > SpikeGadgets > Ethernet from the Connection Dropdown menu. Now, make sure that STATUS (in the upper right corner of your workspace) is “Connected to source.” Once you have confirmed this, proceed to step 3.
3. Select “Stream from source” from the Connections dropdown menu. Trodes STATUS will blink red with the message “No data coming from source,” and a message will pop up saying Windows Defender is blocking the app. Select “public networks,” then “Allow access.” This is crucial as the ethernet adaptor used for connecting to the MCU will show up as a public network in Windows.

Advanced information about enabling core dumps in Linux can be found in the Trodes Wiki, [here](#).

Additional Considerations:

Other connected network devices can block access to the MCU. For example, some USB Wi-Fi interfaces are known to block access to the MCU when using Linux (Ubuntu). To determine if this is disrupting your connection, try disabling/unplugging all other network devices and/or changing the network address from 192.168.0.2 to 192.168.0.x, where $2 < x < 255$ (some values may be blocked by your machine).

To test your connection, connect Trodes to the MCU under the Connection dropdown menu, selecting Source > SpikeGadgets > Ethernet. If the connection is successful, Trodes STATUS in the upper right will change to “Connected to Source.”

Main Control Unit (MCU)



Introduction

The MCU is the compact and rack-mountable central processor and controller of the SpikeGadgets recording system. It makes cutting-edge electrophysiology experiments easy to execute while also being powerful enough to handle the most complex applications.

In its most basic operation, the MCU communicates with a SpikeGadgets headstage to control data acquisition; streaming data to a computer via USB 2.0 or Ethernet (up to 1024 channels via Ethernet or 384 channels via USB). It can also be connected to other SpikeGadgets devices such as the Environmental Control Unit (ECU) or Headstage Combiner Unit (HCU) for experiments where environmental control or multiple simultaneous headstages are needed.

The MCU is used to control both tethered and untethered experiments, utilizing SpikeGadgets data logging headstages. The MCU controls tethered recording via a HDMI tether, and untethered recordings via RF signals transmitted to the headstage. Data is recorded directly to headstage onboard micro-SD storage when recording untethered.

For more information about the [ECU](#) and [HCU](#), please see the relevant sections later in this manual.

MCU Specifications

Maximum channel count	1024
Connection to computer	Ethernet or USB
Logging options	Computer or SD card
Digital Inputs & outputs (I/Os)	6 digital inputs by default
Neural inputs	Single headstage or output of HCU
Sampling clock rate	20 and 30 kHz supported
Closed- loop hardware-computer-hardware delay	Less than 1ms (median)



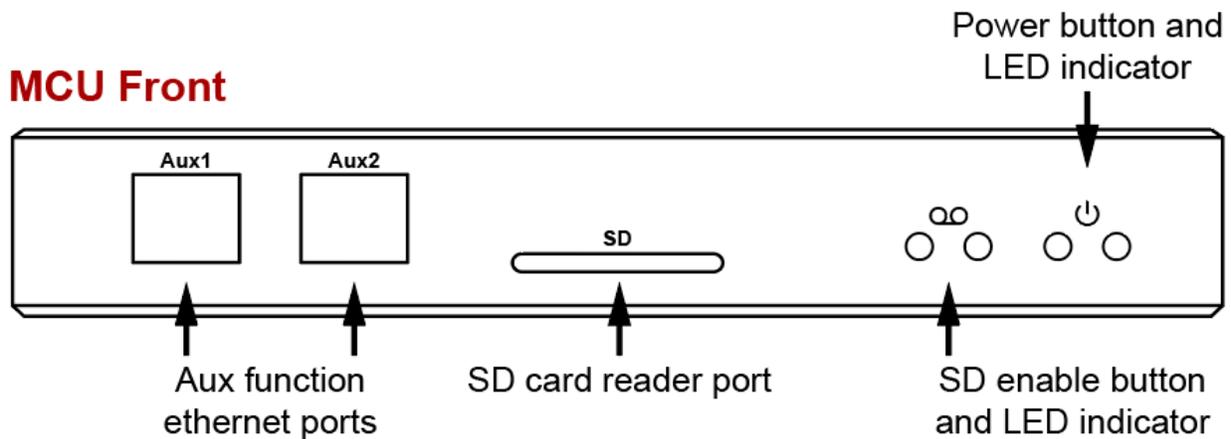
Figure 1 | The MCU front (top) and rear (bottom) panels.

Control and Indicators

Power Button and Indicator: The rightmost button powers the MCU on and off. Simply push the button to power the system on. Press and hold to power the system down. When the system is powered on, the paired LED indicator will blink orange when no headstage is detected, or solid green when a headstage is connected via HDMI or streaming untethered.

SD Card Enable Button and Indicator: SD cards can be enabled for untethered recording using the MCU. To do this, use a SD card adaptor to insert your micro-SD card into the SD reader slot, then press and hold the button down to enable the mounted SD card. The paired LED indicator will rapidly flash red while enabling the SD card, turning solid green when the card has been successfully enabled. The button can be released once the LED turns green. Requires Trodes 3.19 or later.

MCU Front



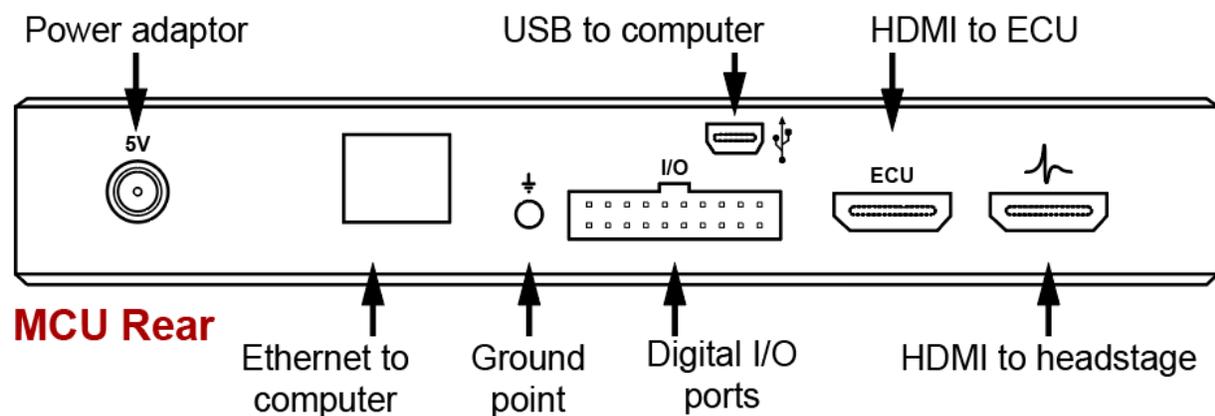
Inputs/Outputs

Auxiliary Input 1: For untethered recording, the RF transceiver is connected to Aux 1 for wireless control of the headstage.

Auxiliary Input 2: By default, Aux 2 receives camera frame sync pulses from compatible TTL-enabled cameras (Allied Vision GigE cameras is recommended).

SD Card Reader Slot: The micro SD reader slot is used to extract recorded data via the Data Logger GUI, or enable the SD card for untethered recording using the SD card enable button mentioned previously.

NOTE: Data streaming in Trodes cannot be initiated while a SD card is mounted in the MCU or connected headstage.



Power Adaptor: The 5V jack for powering the MCU. **Note:** The MCU should only be powered using the 5V power supply provided with the MCU. Other power supplies can damage the unit or produce unwanted electrical noise during recording.

Ethernet (to computer): The MCU connects to your local computer via gigabit Ethernet using a CAT5e cable. This provides a high bandwidth connection for streaming high channel count neural data.

Ground Point: When system grounding is necessary, all experimental hardware and any shielding equipment should share a common ground to reduce electrical noise.

Digital I/O: By default, 6 digital inputs (Din 1 - 6) are enabled for receiving signals from other hardware devices. Digital inputs 7 & 8 as well as digital outputs are turned off by default. Digital inputs operate using 3.3V Transistor-transistor logic (TTL) with 5V max input voltage. **Inputs greater than 5V will damage the system.**

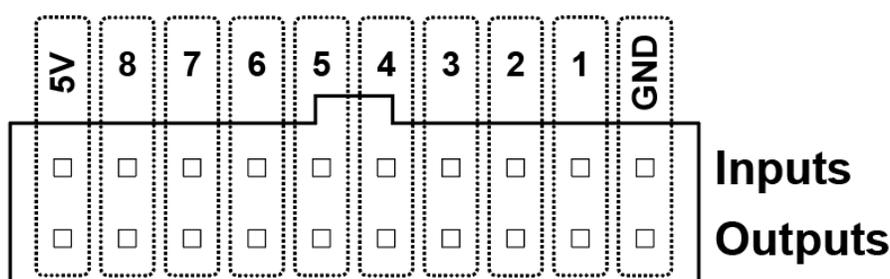


Figure 2 | MCU Digital I/O pin assignment. Inputs 1-6 are enabled by default; inputs 7-8 and outputs are turned off.

USB (to computer): The MCU connects to your local computer via the mini USB 2.0 port. Streaming neural data via USB should be limited to 384 channels, as bandwidth limitations can cause dropped packets when streaming >384 channels.

HDMI (to ECU): The number of digital and analog inputs and outputs (I/Os) can be increased substantially via HDMI connection with the [Environmental Control Unit \(ECU\)](#).

HDMI (to headstage): The MCU can receive HDMI input from either a single headstage connected directly, or up to 8 headstages connected via the [Headstage Combiner Unit \(HCU\)](#). Headstages that do not contain an integrated processor ([ML32](#) base, [HH128 np](#), [UH32](#) & [MM32](#)) must either be connected to the MCU via the HCU or the [Headstage Signal Converter](#).

Recording with the MCU

The MCU can be used in many different recording configurations depending on experimental needs and hardware configuration. Please see the Trodes wiki for more information about [getting started with tethered recordings](#) and [untethered recordings](#).



Environmental Control Unit (ECU)

Introduction

The ECU is a powerful tool for controlling and recording input and output (I/O) signals from peripheral devices such as behavioral apparatuses (beam breaks, lickometers, etc.), environmental stimulus generators (tone generators, etc.) and perturbation devices (lasers, pumps, etc.). These I/Os can be integrated with [StateScript](#), the powerful SpikeGadgets scripting language, to control input and output event timing with high temporal resolution in hardware.

The ECU is equipped with 32 digital inputs (Din), 32 digital outputs (Dout), 8 analog inputs (Ain) and 4 analog outputs (Aout). All I/Os are streamed to the MCU at a sampling rate of 30KHz and are combined with neural data before being sent to the computer.

ECU Specifications

Digital I/O	32 inputs (0 to 5V) and 32 outputs (0 to 3.3V)
Analog Inputs	8 inputs (-10 to 10V range, 13-bit digitization)
Analog Outputs	4 outputs (0 to 5V range, 16-bit DAC)
Breakout connectors	10 channels routed to front SMA connectors (1 Ain, 1 Aout, 4 Din, 4 Dout)
Connection to computer	USB
Run modes	Standalone or stream to MCU
Sampling clock rate	30kHz
Hardware scripting	StateScript



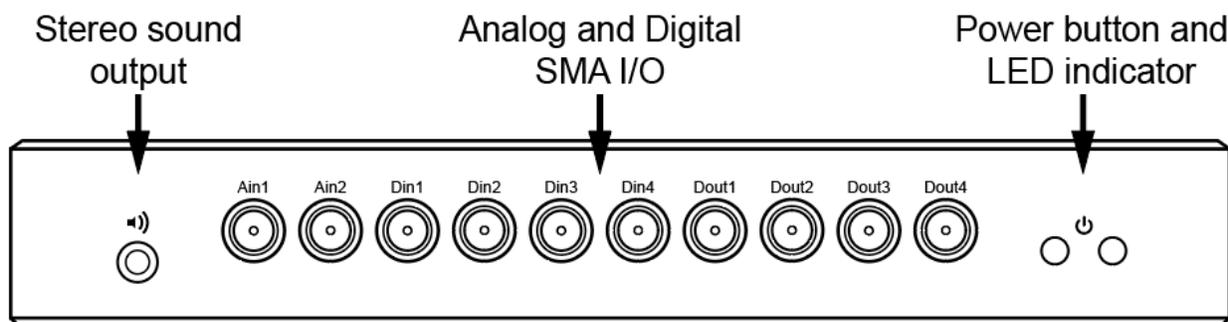
Figure 3 | The ECU front (top) and rear (bottom) panels. Controls, Indicators and I/Os detailed below.

Unit Front: Control and Indicator

The front of the ECU is equipped with a stereo output jack, 10 SMA connectors for digital I/Os (Dio) and analog inputs (Ain) and a power button and LED indicator.

Power Button and LED Indicator: The rightmost button powers the ECU on and off. Simply push the button to power the system on. Press and hold to power the system down. The LED indicator will show solid green when the ECU is powered on. This indicates that StateScript control and data streaming to the MCU are enabled.

Analog and Digital SMA I/O: The ECU front includes 10 SMA connectors, providing easy access to a duplicated subset of the ECU I/Os which are also accessible via the header ports on the back of the unit.

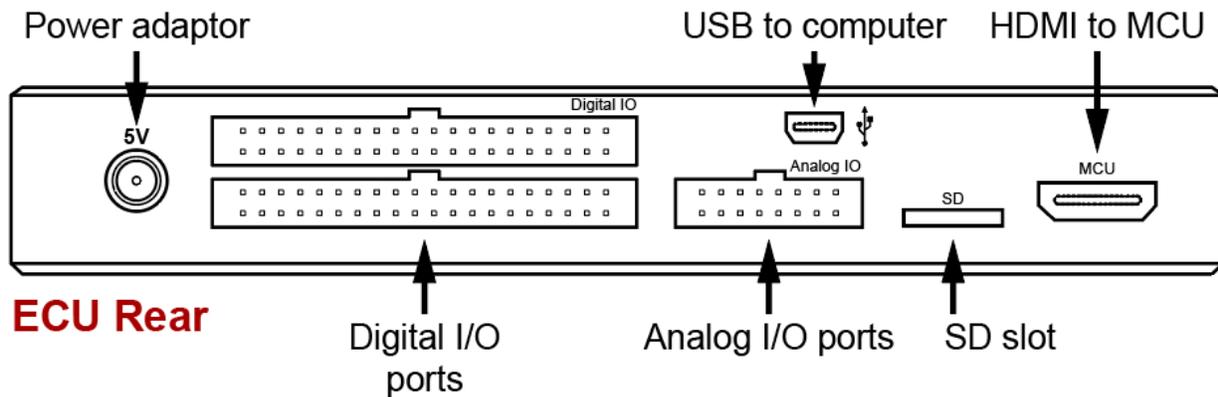


ECU Front

Unit Rear: Inputs and Outputs

The rear of the ECU is equipped with a power jack, Digital and Analog I/O channels accessible via 0.1" pitch headers, mini-USB port, SD slot for audio playback and HDMI for connecting to the MCU.

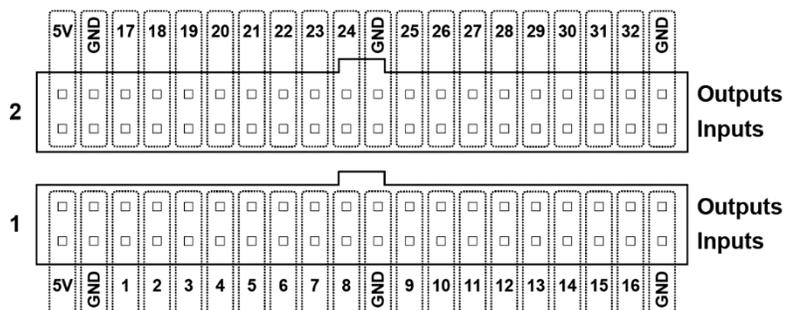
Power Adaptor: The 5V jack for powering the ECU. **Note:** The ECU should only be powered using the 5V power supply provided with the unit. Other power supplies can damage the unit or produce unwanted electrical noise during recording.



Digital I/O ports: Digital I/O pins are divided into 2 40-channel banks. Each bank containing 16 Dins, 16 Douts, 3 grounds and 1 5V out for powering a peripheral device (see Fig. 4 below). Digital inputs can safely receive 5V inputs (above 5.5V is not recommended). Digital I/Os operate using TTL (transistor-transistor logic) gate logic, where inputs 0-0.8V are “low” and inputs 2-5V are “high.” Inputs between 0.8-2V should be avoided as they can trigger either state. Floating inputs (inputs connected to nothing) are pulled down to the low state (47kOhm pull-down resistor).

Digital outputs output a 3.3V signal when in the high state and are designed for logic output only. For applications requiring larger current outputs, the SpikeGadgets [ECU \(SMA\) breakout](#) panel is recommended as it uses driver circuits for delivering larger current outputs.

Figure 4 | ECU Digital I/O pin mapping. I/Os are divided between 2 ports. Each port contains 16 digital inputs, 16 digital outputs, 3 grounds, and 1 5V power output.



Analog I/O ports: The analog I/O ports are accessible in a 3rd smaller bank of pins in the rear of the ECU. The bank contains 8 analog inputs, 4 analog outputs and 4 grounds. Analog inputs are digitized between -10V and 10V using a 12-bit ADC. Analog outputs can deliver voltages between 0-5V.

USB (to computer): The ECU connects to the computer via mini-USB. This allows the ECU to control external devices and record environmental event flag/stimuli via the StateScript environment. The StateScript environment can either be accessed via Trodes when using the MCU or opened standalone using the StateScript application in the Trodes install folder. When using the ECU in standalone mode, the ECU must be connected to your computer via mini USB. Using the StateScript application will then allow for control of external devices and recording of events without the MCU or Trodes.

More information about StateScript and using the ECU in standalone mode can be found on the [SpikeGadgets website](https://spikegadgets.com) and in the Trodes wiki [here](#).

SD Slot: The SD card reader slot is not currently supported.

HDMI to MCU: The ECU connects to the MCU via HDMI for streaming and recording environmental data in Trodes. This can be done alongside neural data, or independent of neural data. This is useful when a precise behavioral/environmental record is needed in the absence of neural recording, such as in behavioral control experiments or training sessions.

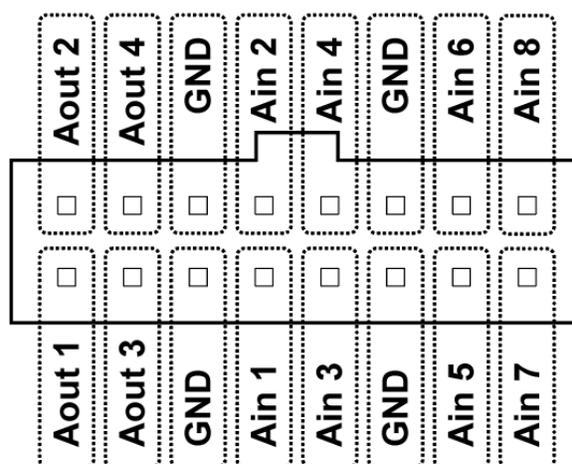


Figure 5 | ECU Digital I/O pin mapping. I/Os are divided between 2 ports. Each port contains 16 digital inputs, 16 digital outputs, 3 grounds, and 1 5V output.



Headstage Combiner Unit (HCU)

Introduction

The HCU is used to combine data streams from multiple headstages into a single data stream before it is sent to the MCU for further processing. The HCU allows for up to 8 headstage streams to be combined for a maximum channel count of 1024 channels of neural data.

The HCU also shares the same dimensions as the MCU and ECU, making it compatible with the SpikeGadgets 19" rackmount plate.

HCU Specifications

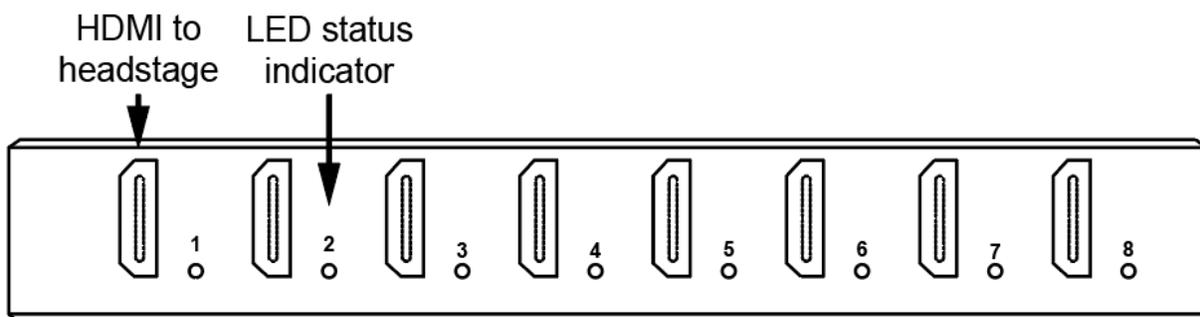
Maximum channel count	1024 or 256, when using 128 channel or 32 channel headstages, respectively.
Connection to MCU or another HCU	HDMI
Supported headstages	UH32 & HH128 without processor



Control and Indicators

The front face of the HCU is equipped with 8 HDMI ports with corresponding LED status indicators. One headstage can be connected to each HDMI port. When a headstage is connected to a port on the HCU, the corresponding LED indicator will light up in a color corresponding to the headstage channel count as follows:

- **Green:** 32-channel headstage
- **Yellow:** 64-channel headstage
- **Red:** 128-channel headstage

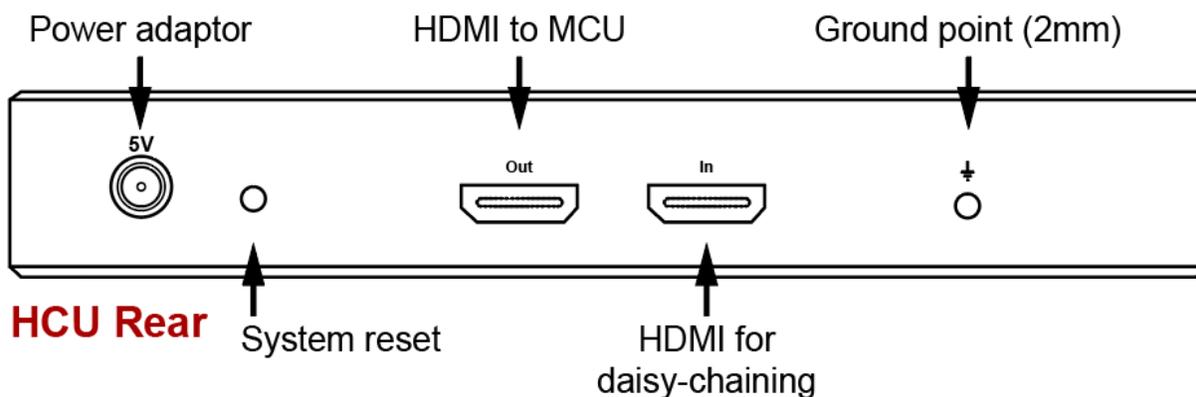


HCU Front

The rear of the HCU is equipped with a power jack, recessed system reset button, HDMI ports for connecting to the MCU or daisy-chaining to other HCUs (*This feature will be included in a future release*) and a 2mm ground point.

Power Adaptor: The 5V jack for powering the HCU. **Note:** The HCU should only be powered using the 5V power supply provided with the unit. Other power supplies can damage the unit or produce unwanted electrical noise during recording.

System Reset: When a headstage is added or removed from the HCU while the unit is powered on, the system reset button can be pressed to update to the new configuration without power-cycling the HCU. The system reset button is recessed inside the enclosure to prevent it from being pressed accidentally, and must be accessed using a thin screwdriver or paperclip.



When the HCU is powered on, any connected headstages are automatically detected, and the device configuration is set. The HCU's current configuration can be read out using the color-coded LED indicators on the front of the unit.

HDMI to MCU: The port labelled "Out" connects to the MCU or another HCU via HDMI.

HDMI for Daisy-chaining: *This feature will be included in a future release.* When using more than one HCU, the port labelled "In" receives input from the next HCU down the line as shown below:

HCU 1 "Out" → "In" HCU 2 "Out" → ... → "MCU"

Ground Point (2mm): When system grounding is necessary, the HCU ground point should be connected to the ground point on the MCU. In general, only one ground point or uninterruptable power source should be used at a time to prevent ground loops. In addition to this, all experimental hardware or any shielding equipment should share this common ground to reduce electrical noise.



Logger Dock

Introduction

The Logger Dock is a powerful asset for streamlining untethered recording experiments. Not only can the Logger Dock be used to start and stop experiments synchronously for one or many headstages, but it can also be used to export neural data, charge batteries, and enable SD cards; all while maintaining a compact size and affordable price.

Logger Dock Primary Functions

Acquisition: The Logger Dock is used in conjunction with Trodes to wirelessly start and stop untethered recording experiments and is also capable of recording environmental events via the Dock's 4 SPI inputs (3 digital and 1 analog).

Data Transfer: The Logger Dock's micro SD slot and headstage cradle can be used to seamlessly transfer neural data to the local computer. Data can be transferred from micro SD cards, or headstages with integrated storage such as the [MiniLogger32 \(ML32\)](#). Currently the ML32 is the only headstage that can be docked directly to the Logger Dock.

Battery Charging: The Logger Dock can also be used to charge batteries for data-logging headstages. When using the ML32 headstage, this is as simple as docking the ML32 in the cradle on the top of the Dock. For all other headstage types, the ML32 cradle is swapped for a charging port compatible with externally mounted headstage batteries.

The Logger Dock can also be used in conjunction with the SpikeGadgets Environmental Control Unit (ECU) for experiments requiring additional digital and analog I/Os.

Logger Dock Specifications

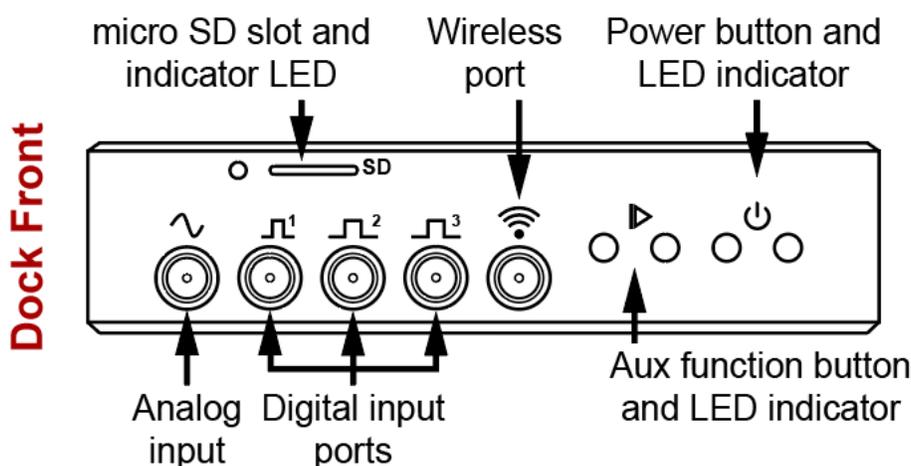
Dimensions	97 x 80 x 25 mm
Connection to computer	USB
Event input connection	SPI
Input Type	3 Digital, 1 Analog
Data upload speed	25 MBps
Power	5V linear connector

Control and Indicators

Power Button and LED Indicator: The rightmost button powers the Logger Dock on and off. Simply push the button to power the system on. Press and hold to power the system down.

Analog and Digital Inputs: The Logger Dock is equipped with 4 SPI inputs for acquiring signals from external devices to be synchronized with the recorded neural data (1 analog input, 3 digital inputs).

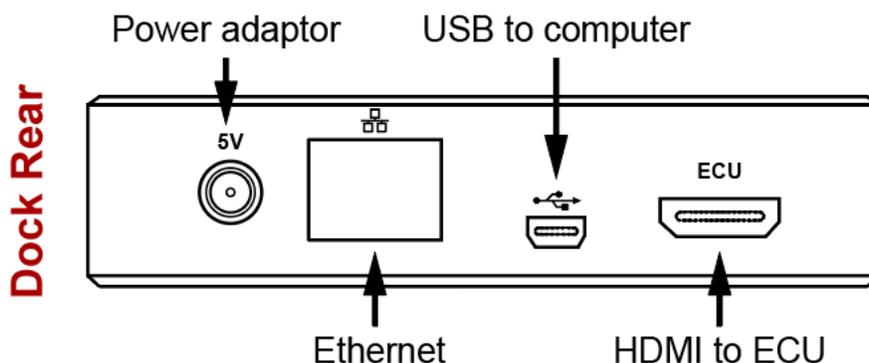
Analog input: 0-5V, 12-bit; **Digital input:** 0-5V (low = 0-1.5V; high = 3.5-5V).



Micro SD Slot and LED Indicator: The micro SD card slot is used to enable SD cards for untethered recording experiments as well as import recorded data to the local computer. This is done using the [Data Logger GUI](#) application packaged with Trodes.

Wireless Port: The antenna included with the Logger Dock connects to the wireless port and transmits start and stop commands for untethered recording experiments. The antenna is needed to ensure maximum wireless range and should always be used when the dock is controlling data logger headstages operating in untethered mode.

Auxiliary Function Button: By default, the auxiliary function button is disabled, but can be reprogrammed to support custom applications using custom firmware. For more information about this, please contact support@SpikeGadgets.com.



Power Adaptor: The 5V jack for powering the Logger Dock. **Note:** The Logger Dock should only be powered using the 5V power supply provided with the unit. Other power supplies can damage the unit or produce unwanted electrical noise during recording.

Ethernet: The Logger Dock ethernet port of deactivated by default but retained for future applications.

USB (to computer): The Logger Dock connects to the local computer via the mini USB 2.0 port. Once connected, the dock can be used to control untethered recording experiments via Trodes, and import untethered data recorded to SD via the Data Logger GUI.

HDMI (to ECU): The Logger Dock can be connected to a SpikeGadgets ECU via HDMI. This allows additional I/O channels to be used for untethered recording experiments controlled by the dock.

Docking and Charging

The Logger Dock supports untethered data logging experiments and is available with 2 different hardware configurations depending on the headstage(s) being used.

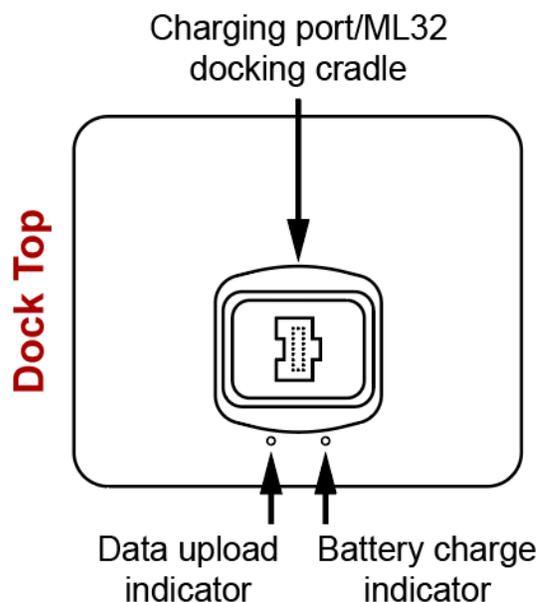
MiniLogger32 docking cradle: When using the MiniLogger32 (ML32), the Logger Dock is configured with a docking cradle for charging, data import and headstage control. Data recorded to the ML32 is imported using the DataLoggerGUI and can be merged with the environmental record recorded by Trodes. When docked, the ML32 begin charging, and headstage settings can be changed via Trodes or the DataLoggerGUI.

Charge port: Batteries used to power headstages other than the ML32 (HH128, Modular Stacking headstage, Neuropixels Probe headstage, etc.) are charged via this port. When using a dock in this configuration, the front-facing micro SD slot is used to import untethered data recorded by the data-logging headstages listed above.

NOTE: Dock configuration is set during manufacture and is not readily changeable by the end-user. This is because internal hardware modifications are needed when changing docking configuration. Please contact support@SpikeGadgets.com for more information if your dock requires a change in configuration.

Data Upload Indicator: When docking the ML32, the data upload indicator turns blue to indicate the SD card is mounted. When exporting data, the indicator turns green.

Battery Charge Indicator: The battery charge indicator provides an easy readout of charge status. Red indicates the battery is charging; green indicates the battery is fully charged.



ML32 Docking Cradle



Charge Port



LED Indicators

Indicator	Location	LED Pulse/Color	Meaning
Power	Front	Solid Green	Dock is powered on
Aux function	Front	Solid Red	Start command received from Trodes; Recording active
Battery	Top	Solid Red	Battery is charging
		Solid Green	Battery is fully charged
SD/Data Upload	Top/Front	Solid Blue	Mounted micro-SD card is detected/idle
		Flashing Green	SD data uploading to computer
		Flashing Red	Headstage/micro-SD card mounting/reading error (try unplugging and replugging)

Initiating Untethered Recordings

Similar to the MCU, the Logger Dock can be used to control untethered recording experiments using SpikeGadgets data-logging headstages. For more information about setting up untethered recording experiments, please see the Trodes wiki [here](#).

Exporting headstage Data

The Logger Dock can be used to upload data to your local computer from any SpikeGadgets data-logging headstage. When using the ML32, this is done by connecting the headstage to the Logger Dock via the ML32 docking cradle. For all other SpikeGadgets data-logging headstages, this is done via the SD card reader slot on the face of the dock.

Once the ML32 headstage or SD card has been mounted to the Logger Dock, data is exported using the [DataLoggerGUI](#) application, or from the command line using the [export utility scripts](#); both of which are packaged with the Trodes application suite.

Additional Resources

For information about getting started with your SpikeGadgets recording system, please visit the [Trodes Wiki](#), which contains additional information and resources covering many basic and advanced topics about using SpikeGadgets recording systems. For new users, a good place to start is the [Quickstart Guide](#).

Additional information and [documentation](#) about all our [products](#) and [software](#) can be found on our [website](#).

About SpikeGadgets

SpikeGadgets is trying something new. Our hybrid approach is to design and sell powerful hardware that interfaces with an open-source software platform supported by a large community of scientists and developers. Our goal is to support the efforts of the open-source community in a commercially-sustainable way.

Technical Support

If you would like technical support, please email us at support@spikegadgets.com.