

An ephys-compliant lick detector based on measured change in capacitance.

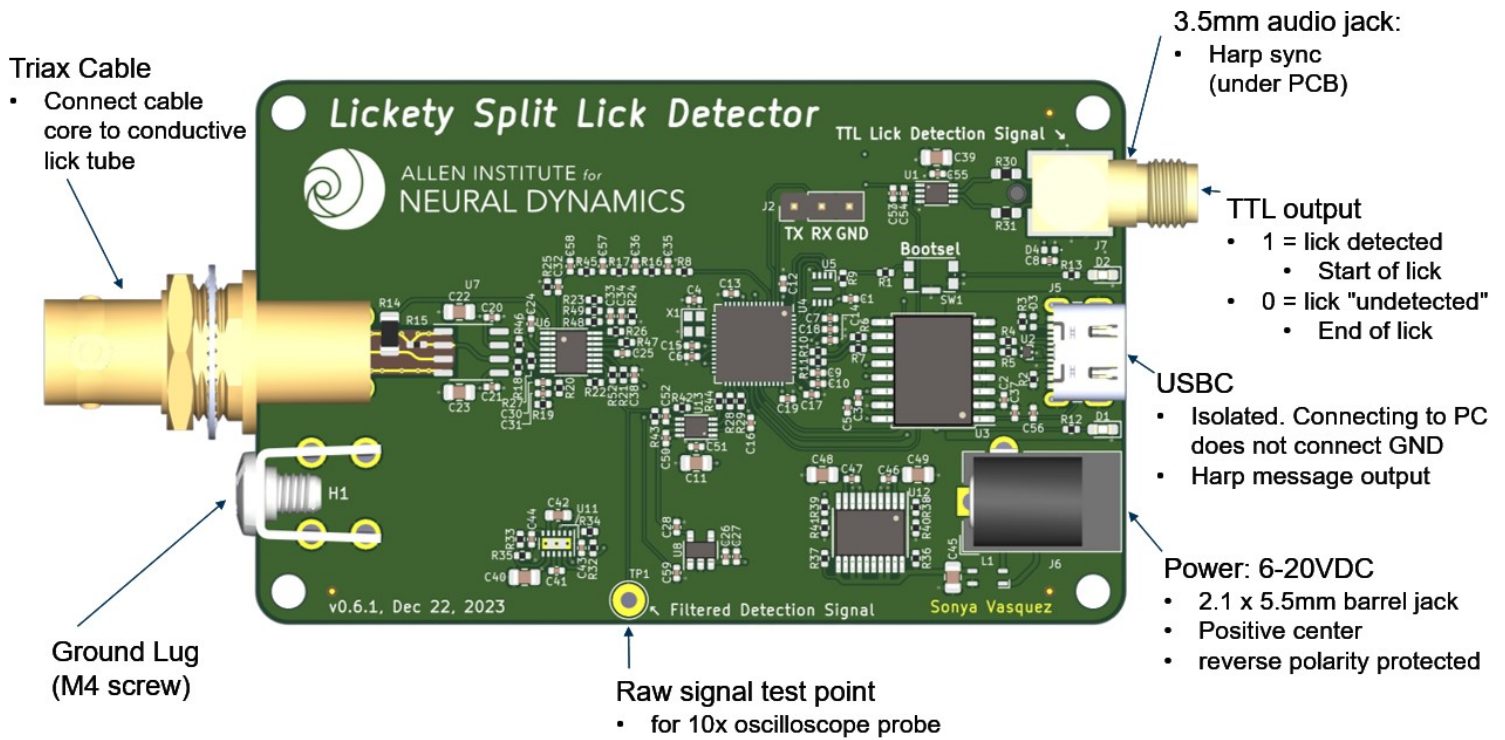
Features

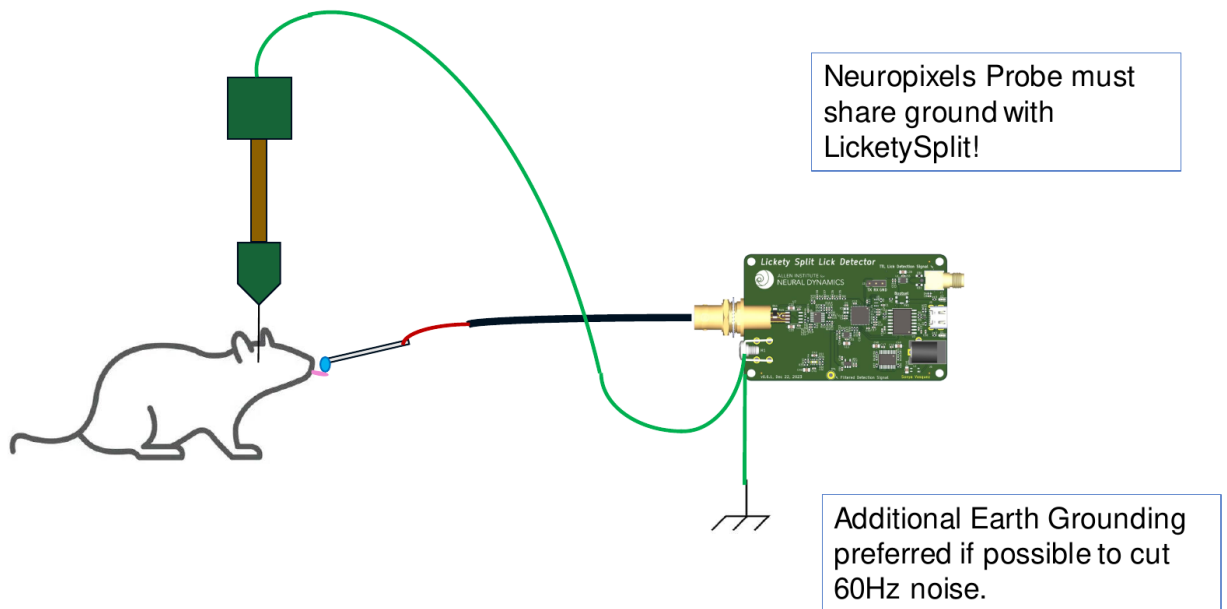
- High Frequency (100 or 125 [KHz]), low current (200[nA]) excitation signal makes this device invisible to [Neuropixel Probes](#) used in electrophysiology recordings.
- Fast. < 1[ms] response time.
- Contact-based. Device triggers when mouse tongue contacts either the dispensing tube *or* dangling reward liquid.
- TTL output triggers when a lick is detected.
- Harp-protocol compliant (serial num: 0x0578). Also dispatches timestamped Harp messages when lick state change has changed.
- Fully supported in Bonsai with a dedicated [Bonsai package](#)

Extra Features

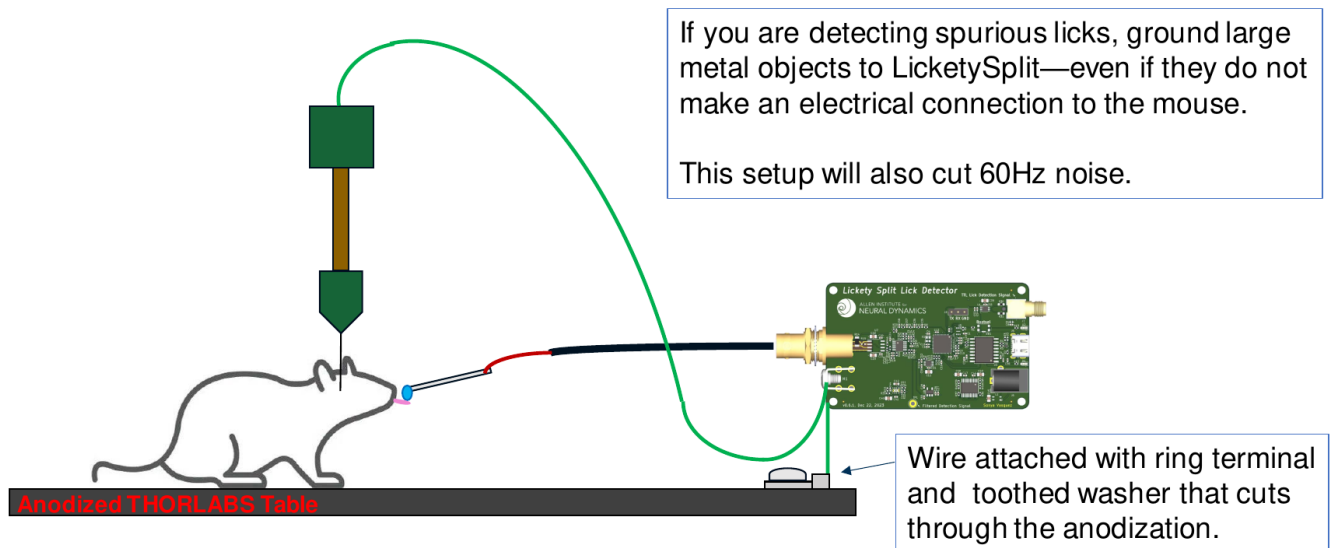
- 6-20VDC input (2.1 x 5.5mm barrel jack, positive center)
- reverse-polarity protected
- isolated USB to prevent ground loops with the PC.
- Two frequency options: 100 and 125[KHz] signal frequency to eliminate crosstalk between two closely-spaced lick tubes.
- Two amplitude options:
 - The 0.02[Vpp] option with proper grounding is intended for Ephys recordings as it introduces negligible noise artifacts.
 - The 2[Vpp] option can be used outside of an Ephys context without a ground connection

Device Pinout





Option C works well if your rig *does* have large sources of 60Hz noise, but you cannot earth ground it.



Cables

This system uses a Triax cable and will work with cable lengths up to .8 meters (2.5 ft) long. Triax cables can be purchased mostly-assembled from this eBay seller [here](#) and then modified slightly to connect to your lick port.

Warnings

Electrical Setup

It is critical that (1) both the device and mouse under test are grounded to a common ground and (2) the rig is earth-grounded (i.e: any exposed metal components are connected to earth ground). Otherwise,

the device may introduce noise on the Neuropixel probes or produce spurious licks from outside electromagnetic interference.

Multiple-Lick-Detector Crosstalk

Multiple lick detectors can be used in the same setup provided that (1) they are grounded correctly (see *Electrical Setup*), (2) the lick tubes are electrically insulated relative to each other, and (3) the spacing between exposed lick tubes exceeds 4mm. If the spacing between exposed metal lick tubes is less than 4mm, then the antenna noise from one lick detector may affect the other. This phenomenon is called *crosstalk*, and it can appear in 2 ways.

1. Lick signals that appear on one detector may show up on both detectors. The system is useable in this case, but the data may need to be cleaned up in post-processing where the shorter "simultaneously" detected lick is thrown out.
2. The amplitudes of the two AC signals will constructively interfere with each other, causing the excitation signals to saturate to their maximum values. The system cannot be used to detect licks this way.

To fix this issue,

1. Put each lick detector on a different frequency (100KHz and 125KHz).
2. Space the lick detection tubes farther apart.
3. Reduce the length of exposed metal on the lick tube.

Theory of Operation

This device detects a threshold change in capacitance. A 100KHz, 10mVpp AC sine wave is played on the tip of a conductive lick spout, (such as McMaster-Carr pn: [89875K271](#)), and the amplitude of the returned waveform is measured every period.

A mouse sharing a ground with this device presents itself as a series resistive and capacitive load. When the mouse's tongue touches the metal tube (or water on the tip of the metal tube), the AC signal passes through the mouse to ground, lowering the amplitude of the returned sine wave below its trigger value, which triggers a lick to be detected. This detection signal is bandpassed in hardware (4th order Butterworth) and the resulting detection signal is lowpassed and compared against a threshold value in firmware to produce a lick/no-lick external trigger output. Overall propagation time through the entire signal chain is less than 1 millisecond.

The detection signal is AC such that repeated contact with the lick tube does not slowly charge the mouse.

Tuning Parameters

Warning: changing the values of these tuning parameters may produce a detection signal that exceeds the <1 [ms] detection time guarantee. We suggest ensuring that your setup is electrically grounded correctly first before changing these values.

There are three parameters that can be adjusted related to tuning the lick detection sensitivity. While the starting values should be sufficient for almost all setups, it is (unlikely but) possible that these knobs may need to be adjusted for a particular setup. The four parameters are:

- "fast moving average" window size
 - the last N raw signal amplitude values are averaged and sent to the consensus filter.
 - This filter smooths out high frequency amplitude changes by averaging the last few values together.
 - constraint: N must be a power of 2 (i.e: 2, 4, 8, 16, ...).
- "consensus" filter window size
 - the last N values must all be below the *lick-detection start* threshold to trigger a detected lick.
 - This filter increases the system's ability to reject intermittent (most likely inductive) noise sources that shrink the signal amplitude for a small period of time and may appear as a lick. Noise sources include fans, valves, and other lick detection tubes placed closer than 4mm apart.
 - Increasing this value will improve the system's ability to reject outside inductive noise, but will slow down the detection time.
 - constraint: N must be a power of 2.
- lick-detection "start" threshold
 - a signal value below this threshold will trigger a detected lick (i.e: the output signal will be set to 5V.)
 - constraint: "start" < "stop" threshold
- lick-detection "stop" threshold
 - a signal value above threshold will untrigger a detected lick (i.e: the output signal will be set to 0V.)
 - constraint: "start" < "stop" threshold

| LicketySplit | |
|-----------------|------|
| whoAml | 1400 |
| firmwareVersion | 0.0 |
| hardwareTargets | 0.5 |

Registers

| name | address | type | length | access | description | range | interfaceType |
|---|---------|------|--------|--------|--|-------|--|
| LickState  | 32 | U8 | | Event | Emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise. | | LickChannels  |
| Channel0TriggerThreshold  | 33 | U8 | | Write | Threshold value to detect the lick. Values below this threshold will be considered a detected lick. | | |
| Channel0UntriggerThreshold  | 34 | U8 | | Write | Threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick. | | |

Analog Front-End for an implanted 22KOhm thermistor.

Features:

- 1Hz highpass filter
- 1KHz lowpass filter
- $V_{out} = 1.65 + 100 \cdot V_{in}$
- Common mode thermistor input filtering
- Common mode supply voltage filtering

Wiring Diagram

TODO

Additional Parts


These parts aren't part of the PCBA, but they are required for a complete sensing solution:

- 2-pin Mill Max Connector pins PN: 850-10-002-20-001000 [digikey link](#)
- 2-socket Mill Max Connector socket PN: 801-47-002-10-012000 [digikey link](#)
- 22kOhm thermistor PN: GAG22K7MCD419 [digikey link](#)
- Flexible 2-conductor 36AWG Shielded TAU cable PN: TAU-3607.02 [link](#)

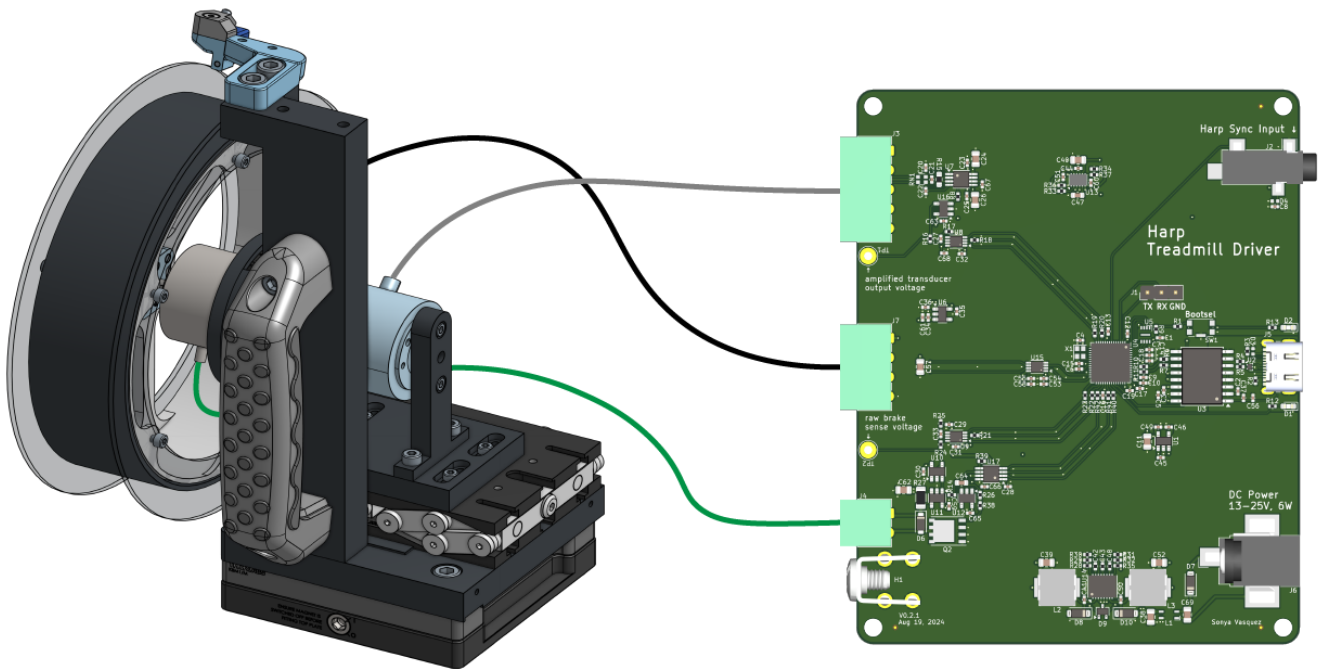
| SniffDetector | |
|-----------------|------|
| whoAml | 1401 |
| firmwareVersion | 0.1 |
| hardwareTargets | 0.1 |

Registers

| name | address | type | length | access | description | range | interfaceType |
|----------------------------|---------|------|--------|--------|---|-------|---------------|
| RawVoltage | 32 | U16 | | Event | Emits a periodic event containing the raw voltage read of the | | |

| name | address | type | length | access | description | range | interfaceType |
|--|---------|------|--------|--------|---|-------|---------------|
| | | | | | thermistor sensor. | | |
| RawVoltageDispatchRate  | 33 | U16 | | Write | Sets the rate at which the RawVoltage event is emitted. | | |

A Harp Device for controlling an instrumented variable-friction mouse treadmill



NOTE

This repository contains design files for the electronics assembly. For the mechanical assembly, design files, and bill-of-materials, see the [OnShape project](#).

Features

- linear (no switching!) current control of 12V magnetic brake, Placid Industries B5ZD, [datasheet](#).
- reads measurements from an [RTS-10 torque transducer](#)
- wheatstone bridge input for torque transducer
- quadrature encoder input for position/speed measurements
- Harp-protocol compliant over full-speed USB
- accepts Harp time synchronization input.
- reverse-polarity protected power supply.

Register Map

[List of Registers](#)




Credits

This board design was created in partnership with the team at [Second Order Effects](#).

| Treadmill | |
|-----------------|------|
| whoAml | 1402 |
| firmwareVersion | 0.0 |
| hardwareTargets | 0.0 |

Registers

| name | address | type | length | access | description | range | intei |
|--|---------|------|--------|--------|---|-----------|----------------------|
| Encoder | 32 | S32 | | Read | Contains the current accumulated number of ticks. | | |
| Torque | 33 | S16 | | Read | Contains the current torque value. | | |
| TorqueLoadCurrent | 34 | S16 | | Read | Contains the current output current applied to the variable torque load. | | |
| SensorData | 35 | S32 | 3 | Event | Emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent] | | Sens |
| SensorDataDispatchRate | 36 | U16 | | Write | Value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s). | [0:1000] | |
| BrakeCurrentSetPoint | 37 | U16 | | Write | Sets the raw value of the torque set-point to be applied to the | [0:65535] | |

| name | address | type | length | access | description | range | inter |
|--|---------|------|--------|--------------|---|-------|----------------------|
| | | | | | treadmill. This value is cleared to 0 if torque_limiting is enabled and triggered. Further writes in this condition return a WRITE_ERROR. | | |
| TareSensors  | 38 | U8 | | Write | Tares the specified sensors. | | Sens |
| ResetTareSensors  | 39 | U8 | | Write | Removes the tare from the specified sensors. | | Sens |
| EnableTorqueLimit  | 40 | U8 | | Write | Enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default. | | Enab |
| TorqueLimitState  | 41 | U8 | | Event, Write | A value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake. | | |

An RP2040-based 16-output [Harp](#)-compliant Clock Synchronizer.

This device can serve as a drop-in replacement for the [Harp-Tech Clock Synchronizer](#) with added capabilities.



Features

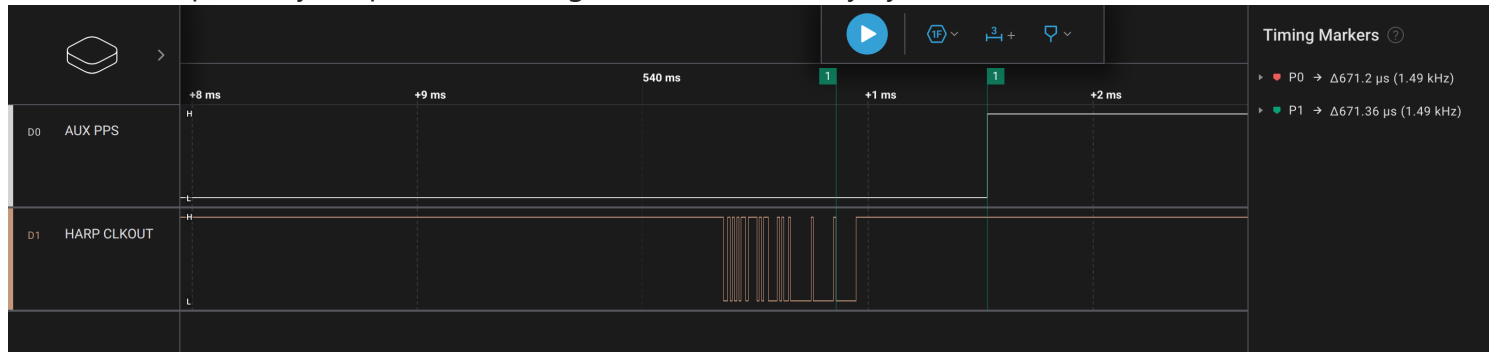
- 16 output channels for distributing clocks to other devices
- 1 input channel for receiving and synchronizing to another clock source
- PPS output
- serial output of the time at user-specifiable baud-rate
- "Used channel detection." Device can identify which channels are in use.

Auxiliary Output

This device features an auxiliary output that can either produce a *pulse-per-second* (PPS) or UART message at the start of the whole second. This external signal enables Harp devices to additionally further synchronize with *non-Harp* devices.

PPS Output

This device optionally outputs a 1[Hz] signal with a 50% duty cycle on the whole second.

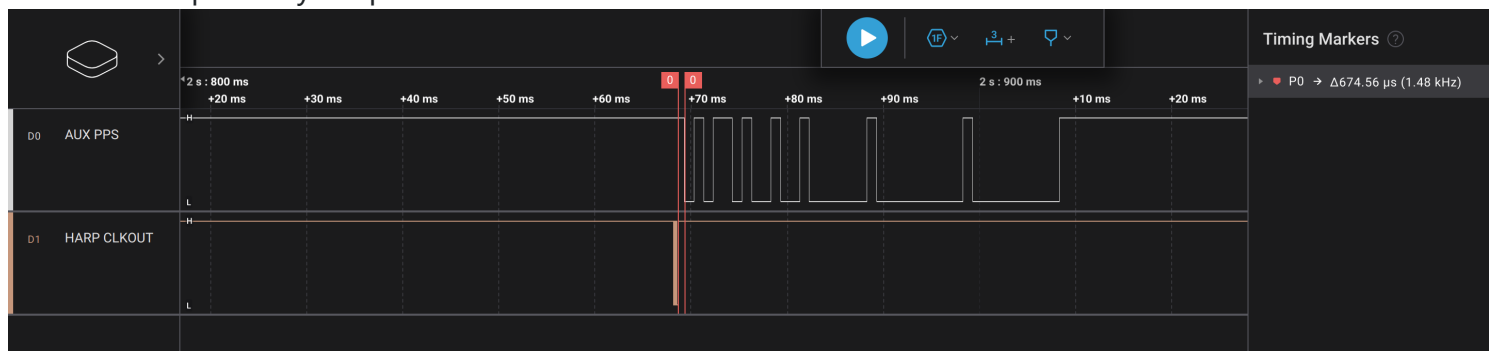


Error from the nominal Harp time is < 1[us].

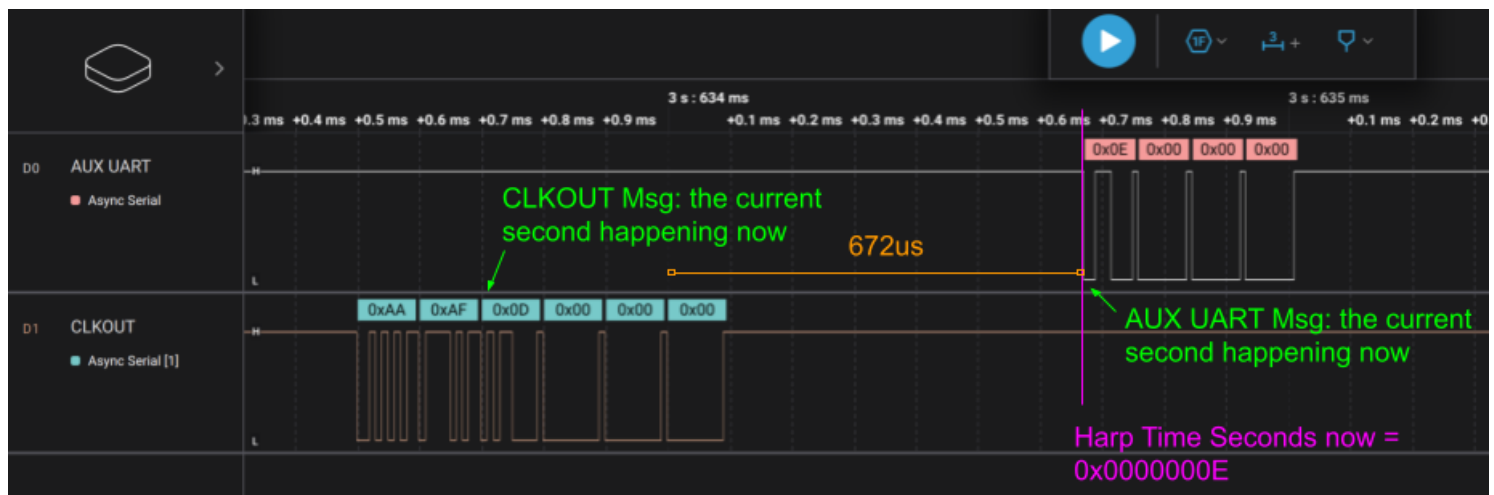
This feature is available on the AUX Port (3-pin terminal block).

AUX UART Output

This device optionally outputs a the current time in seconds at the start of the whole second.



Error from the nominal Harp time is < 3[us].



This feature is available on the AUX Port (3-pin terminal block), and the baud rate is configurable via Harp Protocol (U32 in Register 36).

Developer Notes




In the firmware's **CMakeLists.txt**, adding (or uncommenting) the debug message below:




```
add_definitions(-DDEBUG)
```

will *override* the Auxiliary port behavior to use the auxiliary uart for `printf`-style debug messages. **PPS** and **AUX UART** features will not be available in debug mode.

| WhiteRabbit | |
|-----------------|------|
| whoAml | 1404 |
| firmwareVersion | 0.1 |
| hardwareTargets | 0.1 |

Registers

| name | address | type | length | access | description | range | inter |
|--|---------|------|--------|--------------|--|-----------|-----------------------|
| ConnectedDevices  | 32 | U16 | | Event | The currently connected output channels. An event will be generated when any of the channels are connected or disconnected. | | Clock |
| Counter  | 33 | U32 | | Write, Event | The counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value. | | |
| CounterFrequencyHz  | 34 | U16 | | Write | The frequency at which the counter is incremented. A value of 0 disables the counter. | 0 [0:500] | |

| name | address | type | length | access | description | range | inter |
|---|---------|------|--------|--------|--|----------------------|--|
| AuxPortMode  | 35 | U8 | | Write | The function of the auxiliary port. | | AuxP  |
| AuxPortBaudRate  | 36 | U32 | | Write | The baud rate, in bps, of the auxiliary port when in HarpClock mode. | 1000 [40:1000000] | |

An octal TTL input-or-output board with programmable square wave output

Features

- 8x TTL output or input, configurable.
- Programmable square wave generation.
- Harp-protocol compliant (serial num: 0x057B).
- Bonus: "passthrough buffer mode." External 3.3V and 5V CMOS devices can use this device as an octal buffer with external pins.

Extra Features

- 6-20VDC input (2.1 x 5.5mm barrel jack, positive center)
- isolated USB to prevent forming ground loops with the PC.

Wiring Diagram

TODO

Theory of Operation

TODO: awesome diagram








Harp Register Map









TODO

| Cuttlefish | |
|-----------------|------|
| whoAml | 1403 |
| firmwareVersion | 0.1 |
| hardwareTargets | 0.0 |

Registers

| name | address | type | length | access | description | range | interfaceType |
|---------------------------------|---------|------|--------|--------|--------------------------------|-------|-------------------------|
| PortDirection ↗ | 32 | U8 | | Write | Set the direction of the ports | | Ports ↗ |

| name | address | type | length | access | description | range | interfaceType |
|---|---------|------|--------|-----------------|--|-------|---|
| PortState  | 33 | U8 | | Event, Write | Read or write the state of the ports. An event will be triggered when the state changes without a write command. | | Ports  |
| PwmTask  | 34 | U8 | 18 | Write | Struct to configure the PWM task. offset_us (U32), start_time_us (U32), stop_time_us (U32), port_mask (U8), cycles (U32),invert (U8) | | |
| ArmExternalStartTrigger  | 35 | U8 | | Write | If set to 1, the device will execute the PMW task using the selected pins. | | Ports  |
| ExternalStartTriggerEdge  | 36 | U8 | | Write | Set the edge of the external | | Ports  |

| name | address | type | length | access | description | range | interfaceType |
|---|---------|------|--------|--------|---|-------|---|
| | | | | | trigger. 0: Rising, 1: Falling | | |
| ArmExternalStopTrigger  | 37 | U8 | | Write | If set to 1, the device will stop the PMW task using the selected pins. | | Ports  |
| ExternalStopTriggerEdge  | 38 | U8 | | Write | Set the edge of the external trigger. 0: Rising, 1: Falling | | Ports  |
| SoftwareStartTrigger  | 39 | U8 | | Write | Writing a non-0 value to this register will trigger the PWM task. | | |
| SoftwareStopTrigger  | 40 | U8 | | Write | Writing a non-0 value to this register will stop the PWM task. | | |
| TaskControl  | 41 | U8 | | Write | | | TaskControlPaylo  |

A self-contained, harp-compliant environment sensor

Features

- pressure sensing
- temperature sensing
- humidity sensing
- USB powered
- Harp synchronization input

Wiring Diagram

TODO

Device Pinout

TODO

Wiring Diagram

TODO

Theory of Operation

TODO: awesome diagram

Harp Register Map

[List of Registers](#)

| EnvironmentSensor | |
|-------------------|------|
| whoAml | 1405 |
| firmwareVersion | 0.1 |
| hardwareTargets | 0.1 |

Registers

| name | address | type | length | access | description | range | interfaceType |
|---------------------------------------|---------|-------|--------|--------|-----------------|-------|---------------|
| Pressure [↗] | 32 | U32 | | Read | Pressure, in Pa | | |
| Temperature | 33 | Float | | Read | Temperature in | | |

| name | address | type | length | access | description | range | interfaceType |
|--|---------|-------|--------|--------|---|-------|---|
| ↗ | | | | | degrees C | | |
| Humidity ↗ | 34 | Float | | Read | Humidity, in %RH | | |
| SensorData ↗ | 35 | Float | 3 | Event | A periodic event will be emitted with aggregated data from all sensors. | | SensorDataPayload ↗ |
| EnableEvents ↗ | 36 | U8 | | Write | Enables (~2Hz) or disables the SensorData events | | Events ↗ |