Teabox

High-Speed Sensor Interface

User Guide

Copyright Notices

The user guide, and all content not otherwise noted, is Copyright © 2004, 2005 by Electrotap, L.L.C.

The teabox~, teabox.bits~, and teabox.count~ objects and the Teabox Bridge Application are Copyright © 2004, 2005 by Electrotap, L.L.C.

Max/MSP and Jitter are trademarks of Cycling '74. Reaktor is a trademark of Native Instruments. AudioMedia III is a trademark of Digidesign.

Credits & Acknowledgments

Documentation: Timothy Place and Jesse Allison.

Consulting, feedback, testing, and assistance: Richard Place, MMT Technology, Darwin Grosse, Dan Trueman, David Stevens, Ben Nevile.

Encouragement and support: Leah Place and Noelle Allison.

Table of Contents

Introduction

History

Once upon a time, two graduate students at the University of Missouri - Kansas City were working on interactive music systems, installations, and gesturally-controlled electronics. After encountering a myriad of frustrations and setbacks using existing products and technologies, these two graduate students embarked on a journey to create a new device that would resolve the problems they encountered and set a new standard for gestural sensor interfaces.

The original prototype design was housed in a wooden citrus tea box. While the design and construction no longer bears any resemblance to the original prototype, it seems apropos that the Teabox bears the name that represents its heritage. One thing they do have in common - the Teabox sheds the limitations of MIDI.

What is the Teabox?

The Teabox is an innovative high-speed interface for connecting sensors to your computer using a digital audio connection as the transport. The Teabox offers high-resolution sampling at extremely low latencies, and provides a reliable interface for your real-time sensor needs. This provides the best possible performance for artists working with sensors in installations, performances, music therapy, dance, education, and theater.

In real-time applications, such as a musical performance, live video performance, theater or dance, it is critical to get accurate sensor readings without delay. Too often sensor interfaces are designed to work with the aging MIDI standard - a communications protocol that lacks not only resolution, but also an ability to relay meaningful information in a prompt and timely manner. Some MIDI interfaces also introduce an additional problem: the delay (or latency) in how long it takes to send a message to the computer varies - meaning that you can't even compensate for the delay.

The Teabox scraps this archaic conduit to give you the fastest sensor transport available - a digital audio feed. Most musicians and artists already have sound cards or audio interfaces supporting the S/PDIF standard. Most rarely use

Timeline of Development for the Teabox sensor interface

July 2002
Original Teabox prototype generates square waves

December 2002
Wireless high frequency sine waves used to transmit sensor data via a microphone. Presented at Electronic Music Midwest, South of Chicago.

May 2003
The so-called SensorBox is developed using mixed sine waves of high frequency. Presented at NIME in Montreal.

October 2003
Initial research and idea for using S/PDIF presented at the Ultima Festival in Oslo, Norway.

these connections in performance. The Teabox connects to an available S/PDIF input (optical or coaxial) and transmits the data at lightning speed to the computer as a channel of audio.

The Teabox uses 12-bit analog-to-digital converters on its 8 continuous inputs, significantly improving the expressive range of your sensing devices. This is the equivalent of an integer range of 0 to 4095, far exceeding the 0 to 127 range of most MIDI data. The data for each sensor is then transported over S/PDIF as a 16-bit word.

Connecting Sensors to a TeaBox

The Teabox is packaged in a sturdy and portable half-rack chassis. It uses top-of-the-line Neutrik Combo™ jacks, which accept either 1/4" phone connectors or hardy XLR connectors. The 8 continuous inputs can be connected through the XLR or 1/4" TRS jacks just mentioned (as used on Le Toaster and EOBody systems), or you can use a semi-standard 3 pin header (as used on the I-Cube system), you can also bundle 4 sensors together on one telephone cable using the 6 conductor telephone (RJ11) jacks provided. The 16 toggle inputs can be accessed through 4 separate telephone jacks allowing you to group them together and send them to remote locations.

December 2003
Development of S/PDIF solution begins.

April 2004
Alpha testing begins with testers in the U.S., Canada, and Europe.

July 2004
Beta testing begins with additional testers Australia, the U.S., and the U.K.

August 2004
Teabox units go into production. Circuits manufactured by MMT Technology in Pleasant Hill, Missouri. Final assembly done by Electrotap.

November 2004
The Teabox is presented in production form at the ICMC in Miami, Florida. The Teabox begins shipping, along with assorted sensors and accessories.

May 2005
Teabox is presented at NIME05 in Vancouver, BC.

Summer 2005
A series of new sensors and accessories for the Teabox are released.

What's Included

Upon receipt of your Teabox, be sure to inspect the contents of your package and ensure that you have received the following:

1. Teabox Sensor Interface
2. Power Supply [5V, 500mA or greater, 2.1 mm plug (polarity doesn't matter)]
3. 2 meter TOSLINK optical cable
4. Teabox Software CD:
   - Teabox Bridge Application (for use with MIDI or OpenSoundControl)
   - teabox~ Max/MSP object, help patches, & source code
   - teabox.bits~ Max/MSP object, help patches, & source code
   - teabox.count~ Max/MSP object.
5. Teabox Users Manual
6. Optional High Quality Sensors, sold separately or in the Teabox Starter Kit [please visit http://electrotap.com/sensors/ for a complete listing]

If you believe that there is something missing, please contact support@electrotap.com for assistance.

Getting Started

Attaching the Teabox to the Computer

Take the Teabox out of the packaging and plug in the power supply. When you turn the power switch to the on position, the green LED on the front panel will light up indicating that it has power. After turning on the Teabox, it may take up to 30 seconds before the device actually begins transmitting data.

Next, connect the S/PDIF cable (optical TOSLINK or coaxial RCA) to the Teabox output and to your audio interface input.

Turn on your computer and you're ready to go. It is not necessary to connect or power up the Teabox in any particular sequence. If you plug it into your running computer and then turn the power on it will work just the same.

Power Cycling

Cycling the power on and off on the Teabox (by unplugging and plugging back in, or by switching the power switch) rapidly may inhibit the device from booting up properly. This is a function of the short-circuit protection that is built into the Teabox. No harm will be done; but the Teabox will not be sending any data out.

To solve this problem, turn off the Teabox and wait a minute or two before turning it back on.

When the Teabox is running properly the red light in the TOSLINK optical output will be lighted. There is also a rapidly blinking LED – the Teabox s heartbeat – inside of the unit. You shouldn t need to be concerned with it, but if you are curious, you should be able to see the LED blinking by peeking through the ventilation slots in the cover.

Attaching Sensors to the Teabox

The Teabox accepts sensors that have 3 conductors: 5 Volts, ground, and a sensor connection to send back a voltage between 0.25 and 4.75 volts (the industry standard for many types of sensors).

If you purchased your sensors from Electrotap, simply plug them into the Teabox using the XLR, 1/4" TRS, the 3-pin socket on the front, or the RJ-11 (telephone-style) jacks on the back. NOTE: you can only connect one sensor to any given input, i.e. sensor #1 can only be connected via XLR or 3-pin or RJ11.

You can also connect sensors purchased from other companies. Your success may vary depending on several variables and the quality of data is limited to the parameters of the sensor. In other words, you may not get 12 bit data from a sensor that is designed for different specs, or is not conditioned. In general, you can look at the spec sheets on the manufacturer's website and get a good idea of the usable range.

Conditioning the sensor inputs to protect the Teabox and provide optimal signal is another strength of the Teabox. However, this causes any sensor that has a high-impedance output to work poorly, if at all. Using a sensor with a high-impedance output is not considered good practice anyway, as it won't drive a very powerful signal through the cable and is susceptible to noise and interference. This is particularly true of some accelerometers and infrared sensors (please contact us if you are curious about a specific sensor and we'll tell you what we know: support@electrotap.com). All of Electrotap's sensors have low impedance outputs, which will work anywhere. You can also use our Expander boards (T201), Scalers (T200), or Impedance Converter (T303F) to change a sensor with high-impedance output to output a low-impedance signal.

IMPORTANT: Shorting Sensors!

Sensors can be attached and removed at any time. If you insert a sensor and the device stops running, you have a short in the sensor. **Remove the sensor immediately**, *and check the troubleshooting section in this user s guide.*

Installing the Teabox Software

The CD included with the Teabox contains 1) the Teabox Bridge application, and 2) decoder objects for MaxMSP, along with source code and help files. The most recent versions of the software can be found at www.electrotap.com/teabox/.

To use the Teabox Bridge application, copy it to a location on your hard drive and double-click it. This application provides a conduit between the Teabox and both MIDI and OpenSound-Control outputs.

To use the MaxMSP objects, copy the teabox~ and teabox.bits~ objects into:

- OS X: boot_drive -> Library -> Application Support -> Cycling '74
- OS 9: Your Max externals folder
- PC: C drive -> Common Files -> Cycling'74

Copy the teabox~.help and Teabox.bits~.help files into:

- OS X: Applications -> MaxMSP4.5 (recommended)
- OS 9: Your Max help folder
- PC: C_drive -> Program Files -> Cycling'74 -> MaxMSP4.5

The Teabox and Your Audio Interface

The Teabox is designed to work seamlessly with your audio interface. There are a few things you should know to insure proper operation.

First, your audio interface must be slaved to the Teabox. If it is not synced to the Teabox the data will contain spurious jumps and anomalies. To set this up, you will need to use the hardware setup software that was provided with your audio interface. (Most likely you'll find it stashed away in your Applications or Program Files folders.) Make sure that the Clock Source is set to S/PDIF or External – not Internal.

When using the built in S/PDIF port on Apple's Macintosh G5, you must open the AudioMIDI Setup application to access these options. The Sound Control Panel in the System Preferences does not allow you to adjust these settings.

When synced to the Teabox, the interface will be running at 44.1kHz – CD quality. Customized Teaboxes can be made that run at 48kHz for special applications. Just contact us for info at support@electrotap.com

Many audio interfaces give you the option of routing the S/PDIF channels to various inputs depending on your configuration. While you're setting up the sync, try to ascertain which inputs will be used. It is common for the digital inputs to be appended to the end of the analog input channels. So an 8 analog input interface would have the S/PDIF on channels 9 & 10, or a 16 channel interface on 17 & 18, etc. This can be figured out fairly simply in MaxMSP or the Teabox Bridge application, so it's not imperative that you figure it out here.

Testing Teabox Communications

To test that a Teabox is communicating with your computer, start by plugging in and turning on the Teabox. Then start the Teabox Bridge application.

To test that data is coming in:

- Select your audio interface from the drop down menu on the right.
- Then select the audio channel that is receiving S/PDIF.
- If you have things set up properly, sensor data will be coming in and any unused inputs on the Teabox will rest at halfway.
- If you are not receiving the appropriate data (if the sensors are resting at 0, or are jumping around)
- Make sure that your audio interface is configured properly.

- Make sure that the S/PDIF inputs are enabled and your audio interface is set to use an external or S/PDIF clock source.
- Check your Audio interface's documentation for details.

If you are still having problems, check the Troubleshooting later in this document.

How it Works, Ranges, and Specifics

The Teabox works with sensors by providing power to them (5 Volts and Ground) and receiving a voltage back. In the case of the analog inputs, this voltage is sampled by an analog to digital converter at a resolution of 12-bits. The Teabox then takes the digital representation of the voltage levels and sends them out over a S/PDIF digital audio cable to a computer.

Analog Inputs

Most sensors that are used with the Teabox produce a continuous range of voltages - in other words, they are analog. These sensors can be connected to the Teabox using a variety of input connectors. The table below lists the available methods.

Connector Type	+5V	sensor	ground
XLR	Pin 2	Pin 3	Pin 1
TRS - 1/4"	Tip	Ring	Sleeve
RJ11	Pin 1	Pins 2-5	Pin 6
3-Pin	Pin 1	Pin 2	Pin 3

XLR is the standard for microphone cable. It provides wide availability, robust quality, shielded cable, extendability, and locking connectors. This is the preferred method for connecting sensors to a Teabox system.

TRS is a 1/4 inch phone jack that uses three conductors to carry balanced signals. Most guitar/instrument cables use 1/4" phone jacks, however, they are two conductor (TS) connectors, not 3 conductor connectors. The Teabox requires 3 conductors. This type of connector is used on systems such as the EOBody and Le Toaster sensor systems. It is not ideal due to the fact that any insertion or removal of 1/4" jacks causes very brief short circuits in the device into which the connector is inserted.

3-Pin connectors are small plastic connector with 3 metal conductors pointing out that can be placed into a terminal such as the one on the front panel of the Teabox. They are used in systems such as the I-Cube and Kroonde (the Kroonde is not compatible, however!). While small and light, these connectors are not ideal because they are not locking, and tend to be very fragile.

RJ11 jacks are used for telephones in North America and some parts of Europe. They provide a solution for bundling multiple sensors (up to 4) on a single line. They are also cheap, readily available, light, and have locking connectors.

To use an analog sensor, simply plug it into the Teabox using one of the aforementioned methods.

Digital Inputs

The digital inputs are useful for a variety of applications. The most direct use of them is for simply switches and buttons, but they can also be used for continuous (analog) sensors if the signal is in the appropriate range. Because the Teabox runs at such a high internal frequency, the digital inputs can also be used to receive signals that carry their information as a frequency (such as a sine wave ranging from 0 to 1000 Hertz) or as pulse-width modulation (PWM).

	min	nom	max
Voltage Range	0V (V_{SS})		5V (V_{CC})
Trigger High Threshold (V_{IH})		3V	
Trigger Low Threshold (V_{IL})		2V	

The range of acceptable voltages for the digital inputs is the same as it is for the analog inputs on the Teabox, 0 to 5 Volts. What happens is that the input is "toggled" on (assuming it was previously off) if the incoming signal goes above 3 Volts. The input can then be toggled off by reducing the input voltage to under 2 Volts. The 1 Volt range between 2 Volts and 3 Volts provides hysteresis to prevent false triggers.


Hysteresis

Hysteresis is a property of a system where its current behavior is dependent upon its history. This is useful in a variety of applications. In the case of the Teabox s digital inputs, it prevents false triggering. For example, if a signal is hovering around 2.5 Volts, it will not be constantly flipping the toggle.

For more information on hysteresis, check the following URL in a web browser:

http://en.wikipedia.org/wiki/Hysteresis

Teabox User Guide

The Teabox's digital inputs are accessed via the RJ11 jacks on the rear panel of the unit (see the picture below). It should be noted that there are 3 varieties of the RJ11 standard: 2 wire, 4 wire, and 6 wire. The Teabox uses the 6 wire (or 6 conductor, or RJ11 6-6) standard.
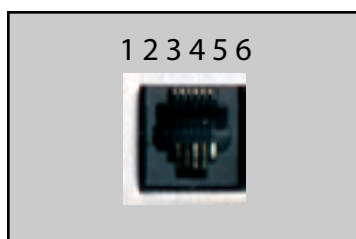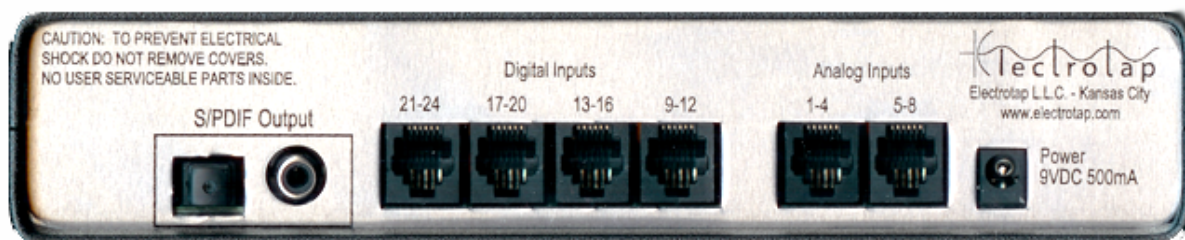
The RJ11 jacks' pins are numbered from left to right (1-6). On the Teabox, the chart below shows the pin configuration. *Note that the pin configuration on the sensor end will be reversed.*

Pin	Function
1	V_{CC} (5 Volts)
2	Sensor Return 1
3	Sensor Return 2
4	Sensor Return 3
5	Sensor Return 4
6	V_{SS} (Ground)

Additional resources for working with the Teabox's digital inputs are constantly being developed. Please check the sensors and accessories section of the Electrotap web site for the most up-to-date information.

http://electrotap.com/sensors/

Example Teabox Configurations

The Teabox is a flexible and open-ended system. What follows are some example configurations to give you an idea of how you might connect a different sensing technologies to your Teabox, leveraging the available accessories and options.
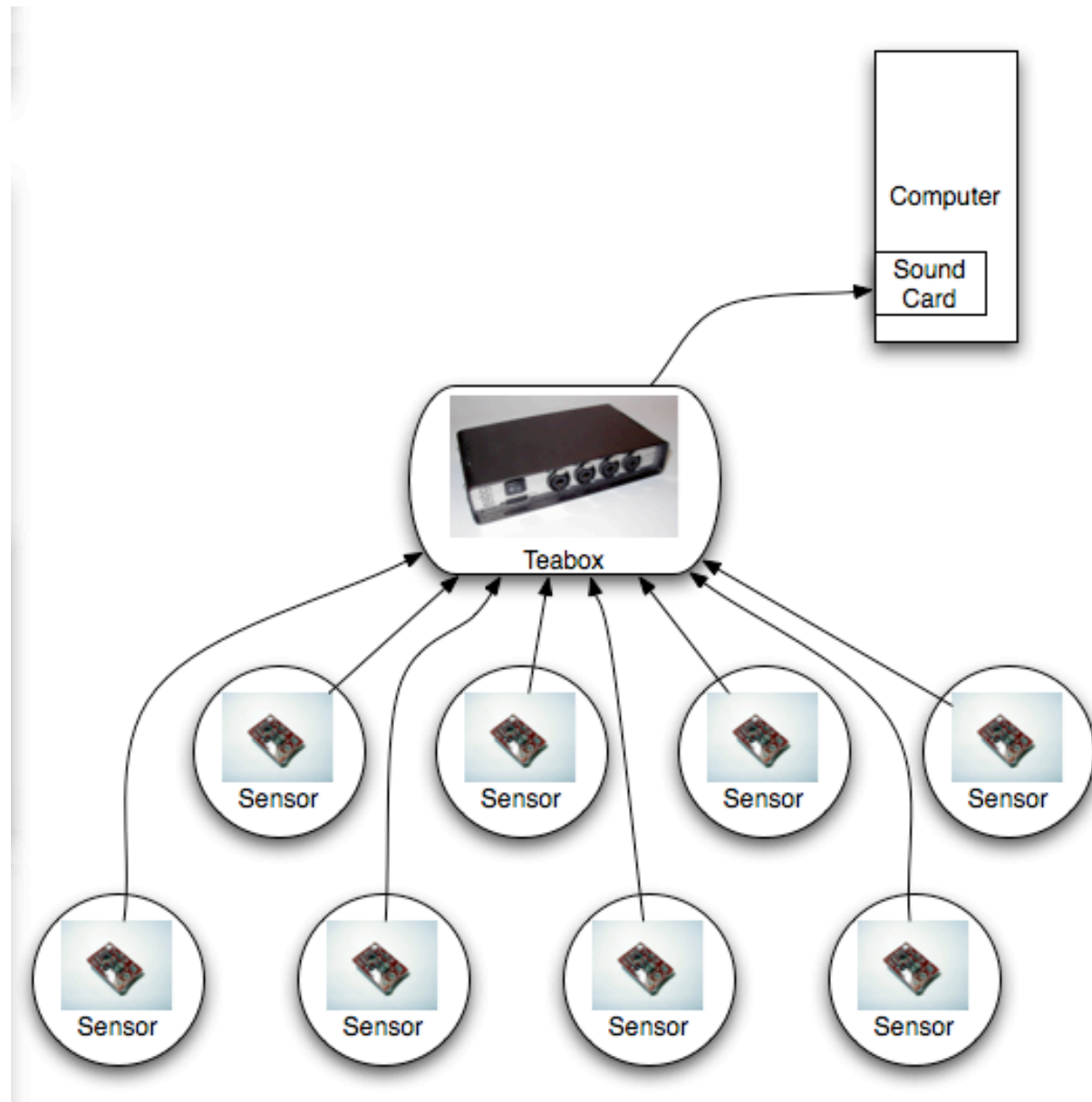
Basic Setup

This example shows a simple, but typical, configuration. The computer uses an external interface, such as a MOTU 2408 or RME Hammerfall. It might be connected using Firewire, USB, or a proprietary cable. The audio interface is connected to the Teabox using a S/PDIF connection (optical or coaxial). A few sensors are connected to the front panel of the Teabox.

Teabox User Guide
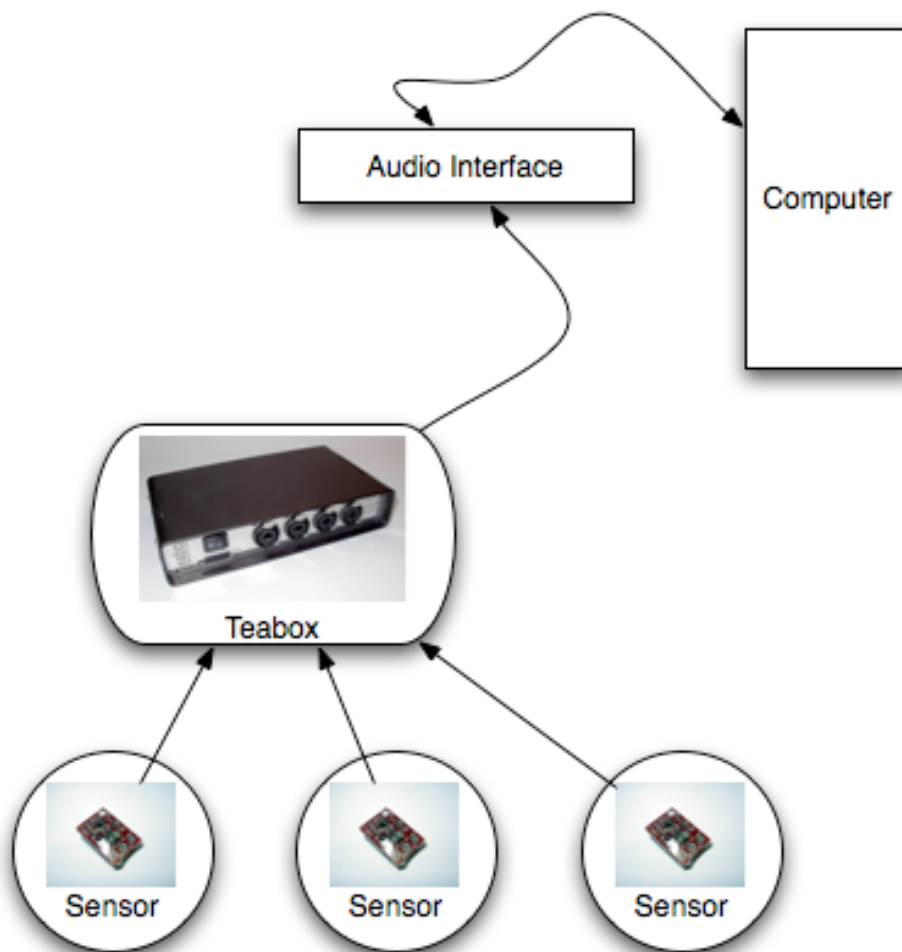
Basic Setup with more sensors

The example below shows 8 sensors connected to the analog inputs on the front of the Tea-box. To achieve this, at least 4 of the inputs will be connected using the I-Cube compatible 3-pin connector block. It also shows an example where the sound card is incorporated into the body of the computer. This is how the setup would look if the built in S/PDIF on a Macintosh G5 were used, or a Digidesign AudioMediaIII card in PC was used.

## Basic Setup with Expander

The downside to the above setup is that you have to use the 3-pin connectors, which are infe-rior to XLR in terms or robustness, extensibility, etc. Another way to connect 8 sensors, but use XLR connections, is to use an Expander accessory. The Expander provides 4 XLR inputs,

which it then sends to the Teabox on an RJ11 cable that gets plugged into the back of the Teabox. The Expander can be connected to analog or digital inputs, depending on your requirements. The Expanders come in a variety of configurations, including some with custom scaling and additional external power.
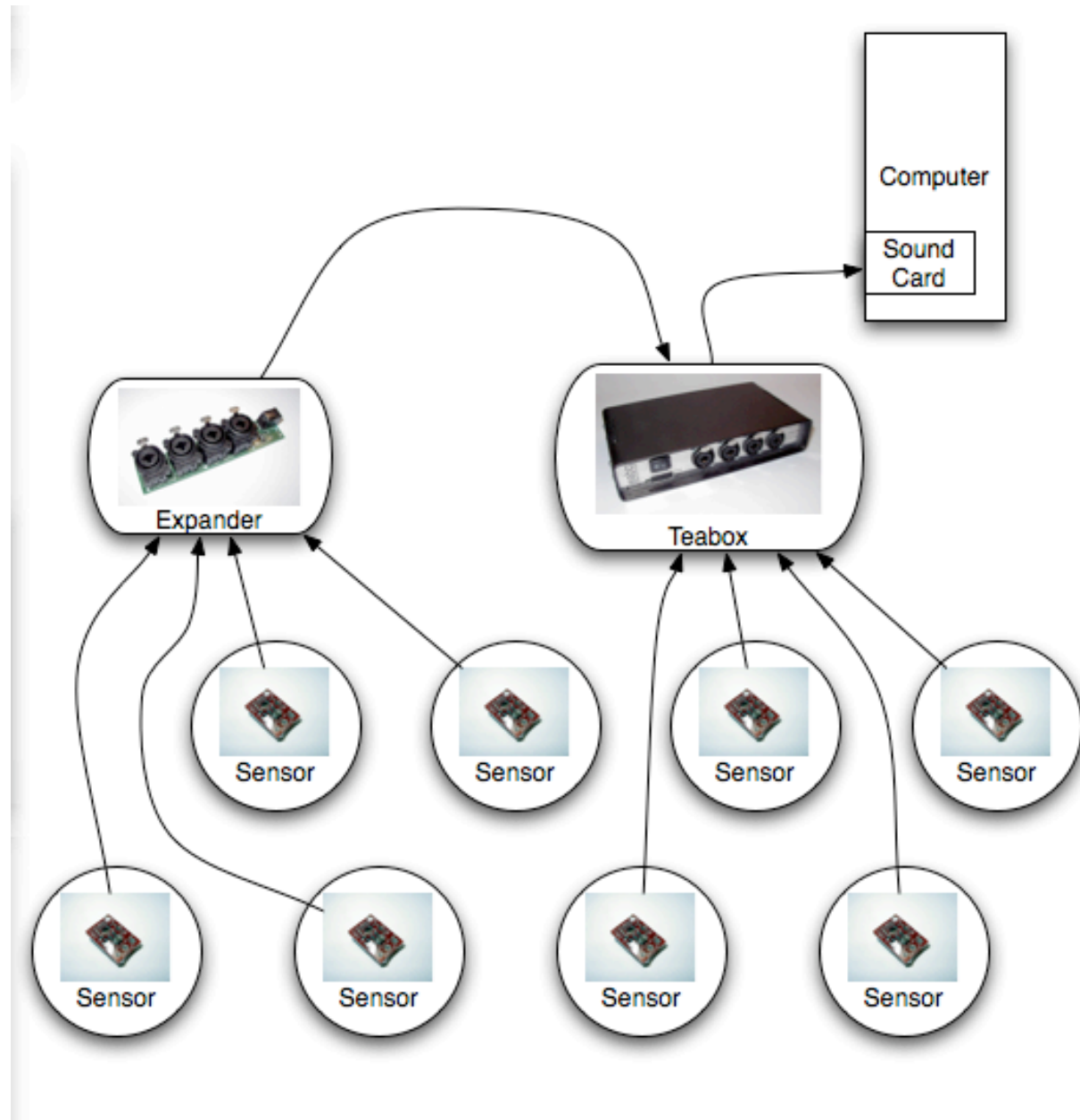
Mixed Setup

In this setup both analog and digital sensors are used with the Teabox. Three analog sensors are connected to the front panel, as in the first example. Then an accelerometer is connected directly to an analog RJ11 input on the back of the Teabox. Finally, a button array is connected to one of the digital RJ11 inputs on the back of the Teabox.
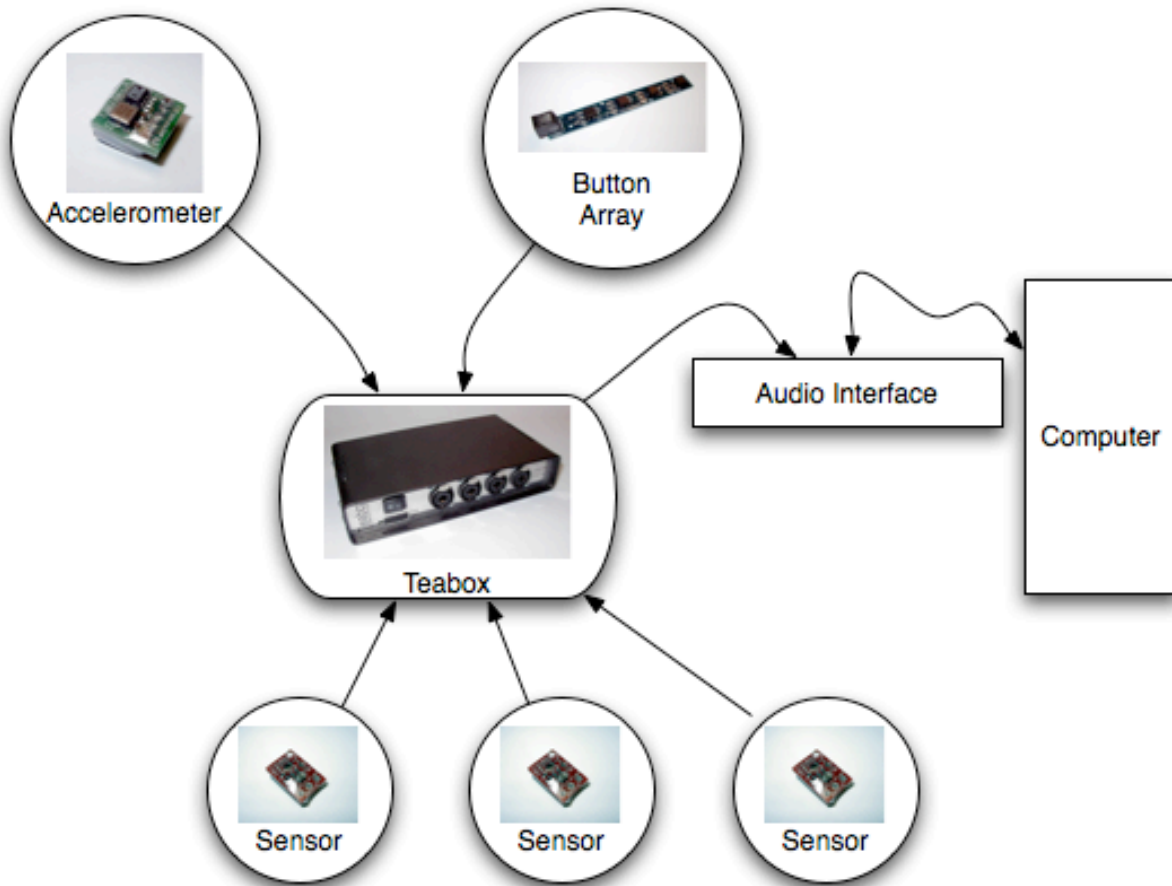
Full Setup

In this example the full capabilities of the Teabox are used. We have 3 analog sensors still connected to the front panel of the Teabox.

We then have 2 dual-axis accelerometers connected. The accelerometer acts as two sensors (one for each axis), so only 2 signals are carried on the RJ11 cable that is connected to the the sensors. The Teabox RJ11 inputs can use this, but they are capable of taking upto 4 inputs. To be more efficient, we can use Merger accessory to connect both accelerometers to a single RJ11 input on the back of the Teabox.

At this point, we have used 7 of the 8 analog inputs. For the final sensor, we will use some kind of funky custom sensor that is then connected to a Project Splitter accessory, which provides a cheap and convenient way to solder things onto a board that connects via an RJ11 jack. So this now maxes out our analog inputs.

But let's assume that 8 analog inputs really isn't enough for this application. In this case we can use the Digitizer accessory. The Digitizer takes up to 4 analog inputs and converts them into square-waves with a variable frequency (roughly 100 to 1000 Hz) that you can then connect to the Teabox's digital RJ11 inputs. That means we can now have 4 more analog sensors connected. The timing and resolution won't be as good as the built-in analog connections, but they will work just fine for this example.

Teabox User Guide
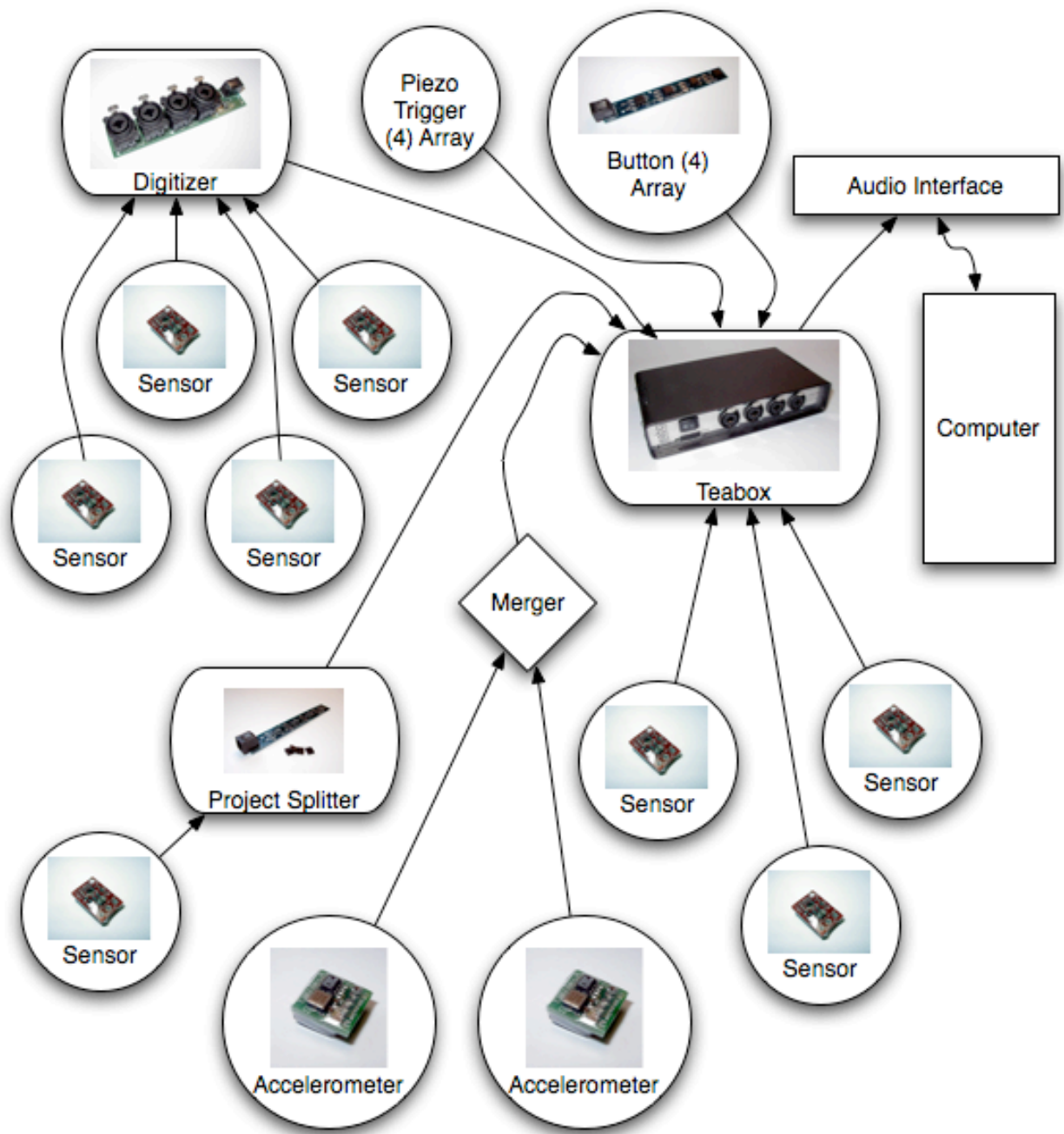
This is followed by an array of 4 piezos used as triggers. This means if the piezo is struck (or something like a drum head on which a piezo is mounted is struck) then it will transmit a trigger signal to the computer via the Teabox.

Finally, we have a 4 button array connected to the digital inputs. In fact, the full capabilities of the Teabox are not used in this example - there is still room for 4 more digital sensors!

Using the Teabox with Max/MSP

The quality of data coming from the Teabox, paired with the flexibility of Cycling '74's Max/MSP & Jitter (http://cycling74.com/), makes creating interactive art an invigorating experience. We provide a few objects and help files to get you going. More examples and ideas can be found at http://electrotap.com/teabox/ . This software is licensed as open source under the GNU LGPL, and is freely re-distributable.

Contribute

If you come up with some useful patches, please let us know. We may be able exhibit and share your work from our website.

This discussion assumes that you are familiar with Max/MSP, and have at least a rudimentary knowledge of creating patches using audio signals. If you do not, please take a look at the reference and tutorials for MSP to get your feet wet.

Setting up DSP and Getting S/PDIF into Max

Max/MSP will bring the sensor data in on whatever input channel your audio interface is using for S/PDIF. All you'll need to do is route it into the teabox~ object and it does the rest.

The Teabox Object

The teabox~ object takes the sensor audio signal in its left inlet. It then outputs 8 signals that correspond to the 8 continuous sensor inputs, one more signal that contains all 16 toggles as a single 16-bit signal, and a final outlet to get information about the object.

The best way to get to know the object is to place it into a patch, and then open the help file by alt-clicking it (or control-clicking it and selecting help from the drop down menu).

The help patch demonstrates a number of ways of using the signal outputs to control things.

Using snapshot~ for control messages

If you don't need to maintain the high speed of the data as audio, you can sample the audio signal at marked intervals with the snapshot~ object. You can then use its output (a floating-point number) to control things. This is especially useful for triggering and controlling regular

max objects. If you need to scale and shift the data to a new range, you can either multiply the number to scale it and add to it for an offset, or use the scale object.

Using the audio outputs

If you need tight synchronization, low latency, and the most responsiveness from your sensors, you'll need to keep the sensor signals in the signal domain. If you need to scale and shift the data you can use the *~ and +~ objects as before, or download and use the tap.scale~ object from www.electrotap.com/taptools/ .

The thresh~ object can be used to trigger events such as grain production in granular synthesis. Routing it into Nathan Wolek's grain.pulse~ object is but one example. You can download that as a part of his Granular Tool Kit at www.nathanwolek.com.

For amplitude modulation simply route the sensor signal into the *~ object and route an audio signal into the other side of the *~ object.

Basically, anything that can take a signal input can be used. This includes filters, FFT's, pitch shifters, etc. Take a stroll through the MSP Tutorials and see what trouble you can get into.

Connecting the toggles

The toggles are also easy to use. The toggle outlet is the second outlet from the right, and it sends out a stream of 16 bit samples. Each bit corresponds to one toggle on the Teabox. 1 is high, 0 is low. To get the data into a more usable format, route the toggle outlet into the inlet of the teabox.bits~ object. This object splits the inputs into 16 individual signal outlets that can be freely utilized.

The edge~ object is very useful for converting the individual toggle signals into regular control data. It detects changes from 0 to 1 (and vice versa), which can then be used to trigger sfplay~'s, gates, switches, etc.
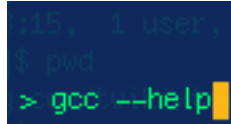
Using Other Applications:
The Teabox Bridge

To facilitate the use of the Teabox with existing systems and software, Electrotap has created an open-source standalone application that translates the Teabox sensor data into MIDI. This MIDI can then be sent out of the computer through a MIDI interface or forwarded on to other applications via an interapplication MIDI bus. This application is called the Teabox Bridge, and it is available for both the Windows and Macintosh platforms.

The Teabox Bridge also has the ability to communicate with other programs, computers, or networks by using the OpenSoundControl network protocol. Unlike MIDI, this provides high-speed, high-resolution, scalable solution to sending data from a Teabox to an OpenSound-Control ready application like Native Instruments' *Reaktor*.

The Teabox Bridge application is freely available on the web at the following URL:
   http://electrotap.com/teabox/

For further information, refer to the documentation available in the Teabox Bridge download.

Writing Your Own Teabox Decoder

If you are using an application other than Max/MSP, and are not satisfied with the Teabox Bridge (or cannot use it), then you may wish to write your own Teabox decoder in a programming language. For example, if you are using PureData on the Linux operating system, it should be easy to write a PD external to decode the sensor data.

This discussion assumes some familiarity with a suitable programming language, and also knowledge of how to obtain samples of audio from the API of your host environment. This example also assumes a floating-point signal representation.

Explanation of the Task

The Teabox samples each sensor at 12-bit resolution and assigns it to a 16 bit value. These values are then multiplexed (cycled through) and sent as S/PDIF digital audio. The order in which samples are received is presented below.

0	1	2	3	4	5	6	7	8	9	10
Start Flag	Firm-ware Version	Analog Input #1	Analog Input #2	Analog Input #3	Analog Input #4	Analog Input #5	Analog Input #6	Analog Input #7	Analog Input #8	Digital Inputs

This process is mostly straight-forward. The one exception is the handling of the Start Flag. The Start Flag is defined as all 16-bits being high (a value of 0xFFFF). It is possible, however, for the Digital Inputs to also produce the value 0xFFFF if all of the inputs are toggled high. This is rare situation for most users, but it must be handled properly by the decoding algorithm.

Example Code (in C)

```
// DEFINE VARIABLES
float      input_previous = 0;   // used for comparison
float      sensors[9];           // 8 analog + 1 digital
int        bitmask;              // 16 digitals represented as an int
int        counter = 0;          // location in the block of samples
float      hw_version = 0;       // firmware version

// DECODER FUNCTION
void teabox_decoder(float input)
{
      // If the sample is the start flag...
      if(input < 0.0 || x->counter > 9){
```

Teabox User Guide

```
        // look at last input in case it was digital inputs
        if(input_previous < 0.0){
              sensors[8] = input_previous;
        }
        counter = 0;
}
else if(x->counter == 0){
        hw_version = value * 8.0;
        counter++;
}
else{
        sensors[x->counter - 1] = value * 8.0;// Normalize range
        counter++;
}

// POST-PROCESS TOGGLE INPUT BITMASK
if(sensors[8] < 0){
        // 4096 = 32768 / 8 (we already multiplied by 8)
        bitmask = sensors[8] * 32768;
        bitmask ^= -32768;
        bitmask = 32768 + (bitmask);       // 2^3
}
else{
        // 4096 = 32768 / 8 (we already multiplied by 8)
        bitmask = sensors[8] * 4096;
}
sensors[8] = (float)bitmask;

/* All sensor data is stored in the sensors[] array,
 * which may be accessed to get the values
 */

input_previous = input;
}
```
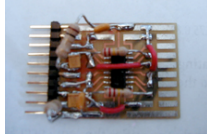
Building Your Own Sensors

For do-it-yourselfers, creating your own sensors to use with the Teabox can be fun and relatively easy.

The Teabox provides both 5 volts and ground to the sensor. The sensor then uses that (or not) and sends back a voltage somewhere between 0.25 and 4.75 volts. Anything beyond those extremes (within reason) is clipped.

There are three basic types of sensors (there are other types, but they are less common):

1. Variable Resistors
2. Variable Capacitors
3. Voltage Generators

The easiest to work with, by far, is a variable resistor. They can measure light, pressure, flex, and implement turn potentiometers, sliders, and all sorts of other things. For more information on creating sensors and working with their signals, there are several articles available online at http://electrotap.com/articles/ .

Impedance and the Teabox

You should also be advised that the Teabox is optimized for use with low-impedance sensors. Most sensors will have no problem working with the Teabox, but if your sensor is a high-impedance sensor then you should use an op-amp as a unity-gain voltage follower.

For additional resources, please check Appendix A in this document.

Troubleshooting Guide

The chart below lists a series of problems that you may encounter with the Teabox, and at least one likely solution.

Problem	Likely Cause(s)	Solution
Teabox's sensor data has occasional spikes or jumps in it.	This usually indicates that the computer is not slaved to the Teabox's S/PDIF signal. This may be due to an incorrect software setting, or due to the use of a poor quality cable.	Make sure that you are using a high quality S/PDIF cable (such as those provided by Electrotap). Find the audio setup utility for you audio card or interface. In that software set the clock source to external (or S/PDIF).
Sensor data seems jittery or jumpy.	If no sensor is plugged in, the input floats. This is expected behavior.	No action required.
	If a sensor is connected, the jumps are most likely caused by either the sensor itself or by the cable connected to it.	Ensure that you are using a high quality cable. Ensure that your sensor signal is properly conditioned prior to being sent into the Teabox (all sensors produced by Electrotap are fully conditioned). If the sensor is still jittery (some types of sensors are naturally this way - particularly accelerometers and flex sensors), then condition the signal in software. For example, you can run a demultiplexed signal through a lowpass filter.

Problem	Likely Cause(s)	Solution
The Teabox has stopped sending data.	A sensor has created a short circuit between V_{CC} (power) and Ground.	Disconnect the offending sensor IMMEDIATELY. Try to locate the source of the short.
	Too much power has been drawn by the sensors and a brown-out has occurred.	Disconnect sensors one-by-one, waiting 30 seconds after each one to see if the Teabox starts back up. If it does start back up, then the sensors were drawing too much current. The Teabox can provide up to 250mA to its sensors, which far exceeds typical usage. To fix this you will need to cut down on the power consumption of the sensors. It may be as simple as adding a resister on the power supply to the sensor to limit current flow.
The Teabox doesn't seem to start up properly. When it is turned on the power light comes on, but the interface never locks onto the S/PDIF signal. If the optical cable is unplugged, the little red LED never lights up.	The internal power management system (used for short circuit protection) needs to be reset. This condition can occasionally be caused by cycling the power on and off too rapidly.	Turn the Teabox off and wait 5 minutes. Then try turning the Teabox on.

Problem	Likely Cause(s)	Solution
Teabox Bridge, or teabox~ Max object is sending out nothing but zeros.	The S/PDIF channel with sensor data on it has not been properly routed to the application or teabox~ object.	Check to make sure that S/PDIF is enabled on the audio interface and that the correct channel is being used.
	The Teabox has been shorted by a sensor.	Check to see that the optical LED is lit and green internal LED is blinking rapidly. This indicates proper functioning. Unplug the offending sensor and restart the Teabox. Make sure to give it a little time (10-20 seconds) before turning it back on.
	The audio interface is not in sync. It is not receiving its clock source from the Teabox.	Set the clock source to S/PDIF or external in your audio interface's setup software.
	The Teabox is not turned on.	Make sure the Teabox is plugged in and turned on. This can be verified by the green light on the front panel of the Teabox.

Contact and Support

The most efficient way to reach us is either by email or via the support form on our website. When sending us a support question or request, please remember to include the following information, and be as thorough as possible:

- Type of computer you are using
- Version of your Operating System
- Type of audio interface or sound card
- Type of cable you are using to connect the Teabox to your sound card
- Version (and name) of the software you are using with the Teabox
- A concrete list of steps to reproduce your problem, from scratch
- What sensors you are using with the system
- Confirm that you've looked through our Troubleshooting Guide in this Document.

When you have compiled, send it to us and we will look into your problem. We know that problems, even if they are misunderstandings, can be frustrating, and we will do our best respond in a timely manner.

Electrotap, L.L.C.
606-A East Street
Parkville, MO 64152

support@electrotap.com
http:/electrotap.com/

Appendix A: Additional Resources

Articles @ Electrotap
http://electrotap.com/articles/

Our website contains a section for articles and application notes with information that is pertinent to the Teabox, sensor design, and interactive systems.

Physical Computing
http://fargo.itp.tsoa.nyu.edu/~dano/physical/physical.html

A Hands On Guide for Artists, Scientists, Hobbyists (and Engineers in a Hurry).

Parallax
http://www.parallax.com/

These are the makers of the Basic Stamp II microcontroller, a popular do-it-yourself solution for interfacing sensors to a computer. It contains a variety of application notes and sensor recommendations which may also be applicable to the Teabox.

New Interfaces for Musical Expression
http://www.nime.org/

NIME is an annual gathering for the presentation of the latest innovative solutions for working with interactive systems and gestural controllers. The proceedings are published online and offer a wealth of information regarding sensors.

Circuit Cellar
http://www.circuitcellar.com/

More interesting articles on electronics and design.