EventBox

Audio detection and EEG

For measuring event-related potentials (ERPs) with EEG, it's important that the output file contains markers when a sound was started: many sound parts have to be averaged (and correctly aligned) to remove the (background) noise. The averaging will reduce the noise with a factor that equals the square root of the number of parts that are averaged. To create the markers in the software, triggers have to be sent externally to the hardware of an EEG device. This can be done immediately from a computer, but unfortunately the latency of the sound cards should be taken into account (> 50 ms). By using ASIO drivers the latency will probably still be 50 ms for correct playing, but because this driver is communicating directly with the hardware of the sound card, the jitter can be as low as a couple of ms.

A better approach is by using a sinusoidal train [1] on the second channel of an existing sound file which marks the beginning of the sound part. We have made an EventBox which can detect this sinusoidal train and send the stored trigger value immediately to an EEG interface when a sinusoidal train has been detected. Because the detection operates on the audio signal that has been outputted by the sound card, the latency is independent of the hardware of the computer and used operating system: the accuracy will be better then 30 µs! Idea of using a sinusoidal train by Ton Wempe.
The EventBox can communicate with a computer by using the Ethernet protocol (RJ45 network connection, peer- to-peer) so that round-trip communication (for feedback to computer) can be executed within 1 ms. The device can be powered by using the USB of the computer, so that no power adapters are needed.
The trigger value can be sent from the computer to the EventBox where it will be stored, waiting for a trigger pulse. When an sinusoidal train has been detected, " immediately" the value will be sent to an EEG interface (< 30 µs) and feedback will be sent to the host computer (< 1 ms). Because of the received feedback the host is notified that it can send the next trigger value to the EventBox.

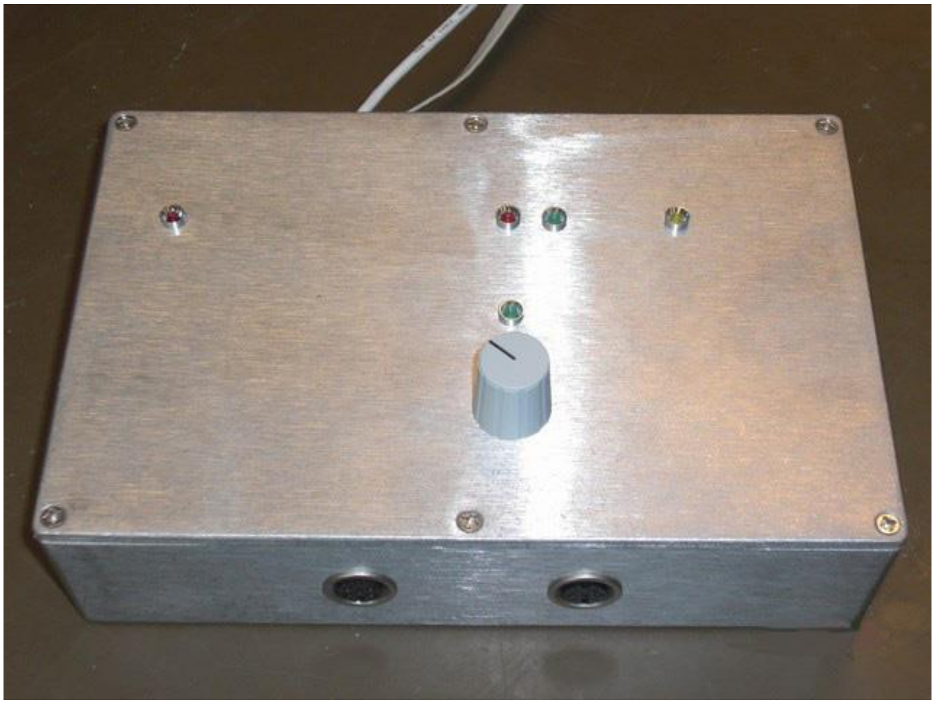

Figure 1: EventBox

Button box
Besides detecting/sending triggers the EventBox can also be used as a button box. The box will send an acknowledgment (latency within 1 ms) to the computer when a button response has been detected. The acknowledgement incorporates the internal clock of the EventBox which is independent of the hardware of the computer. This can be useful when a button press follows on an audio signal that is detected by using the sinusoidal train. When the audio signal has been detected, an acknowledgment will be sent to the host with the internal clock of the EventBox. By subtracting the internal clocks from the button response and the audio triggering, a very reliable reaction time can be accomplished, which is better then 30 µs.

Screen sensor
It's also possible to connect a light senor which can be connected to the surface of a LCD screen. Because this sensor is color sensitive, it can detect when a little square with a certain color is painted. Red/blue is the color pair that makes the best distinction since they are on both sides of the light spectrum. Using colors is convenient since the backlight of most LCDs are using pulse-width modulation for controlling the brightness. This means that the backlight is flashing around 200 times a second with a certain on/off-time. A "normal" sensitive light sensor would also react to this flashing which is unwanted.
With the screen sensor there can be detected when new info has been painted to the screen after a vertical blank. Because the box is communicating via Ethernet the latency could be better then 1 ms. Due to the fact that the response time of the monitor influences the accuracy of the light sensor, this parameter sets probably the latency.

Hardware/Software of audio detection

A Sinusoidal train of 5 kHz (trigger), with a duration of 10 ms is used on the second channel of a Wav file to mark the beginning of a sound part. The sinusoid train is starting with a negative half which is used by the hardware for reliable detection. The zero-crossing after the first negative half should then be lined up with the start of the target sound (see fig. 2). It's preferable that the sound starts with some silence, so that startup effects of the sound card are avoided. There is no latency problem due to adding blanks, because it will mark precisely when the sounds start, independently of the blank sound part. The position and writing of the sinusoidal train can be be done in Praat with some scripting. First the beginning of the sound has to be marked by using annotation (TextGrid). Scripts that we have made will create the sinusoidal train on the position that has been marked by the TextGrid in Praat.
Of course the second channel will not be heard by the participant, it's only used by the EventBox for detection. The EventBox will reroute the first channel to both audio output channels so that there is audio signal on both speakers/headphone.

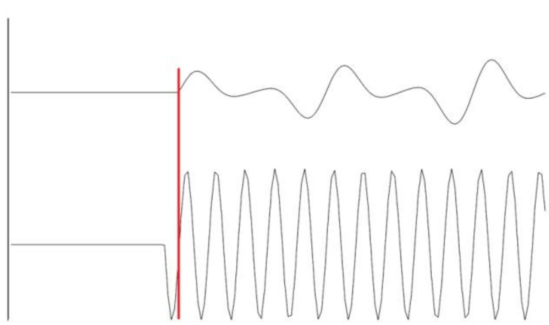

Figure 2: Alignment trigger

For a reliable detection the amplitude of this trigger should be significantly above noise level and clipping should be avoided, otherwise saturation effects will cause delays. To make sure that the incoming trigger signal is within bounds, it can be adjusted in amplitude with a potentiometer: so that the green LED is on and the red LED is off (see fig. 3).

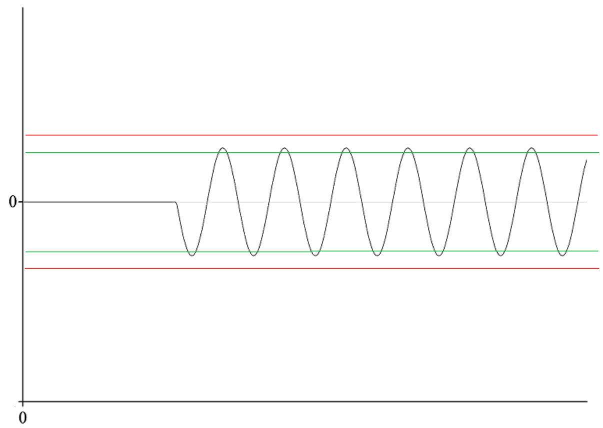

Figure 3: Amplitude adjustment

The signal will be passed through a simple 1st order filter. A higher order will give more phase shifting which we want to avoid.

A zero-crossing detector will detect the first zero-crossing after the first negative half. A mono-stable will create a pulse with a width of 30 ms. The rising edge of this pulse is used as a trigger for enabling the parallel port output buffers (see fig. 4).

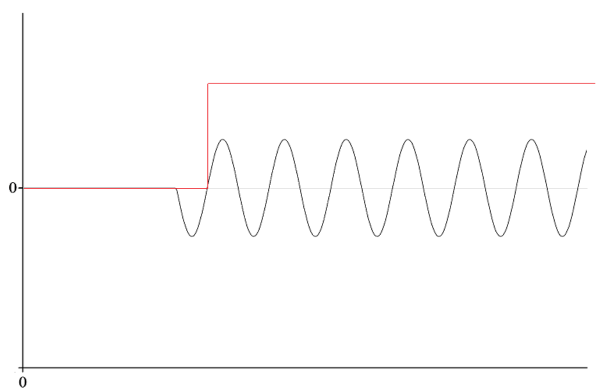

Figure 4: Created trigger pulse

The output buffers are already loaded with the correct trigger value (that have been received by the host) that have to be sent to the EEG interface. So the only substantial delay is caused by the phase shift of the filter at the input.
An acknowledgement will be sent to the host, so that the host is notified that the next trigger value can be sent. This value can be read from a file which has info on every line for the name of the sound file and the trigger value. This takes care of the synchronization between the trigger value and the sound file that has to be played.

Because the triggers that are sent to an EEG interface are very accurate, we know precisely when the sound was started, but we cannot control for the exact time between events. This means that the time between audio events could jitter.

Most often this isn't an issue, but if it is, it can be solved by creating one big audio file with all the audio events concatenated. Care should be taken that the synchronization between EventBox and host is keeps correct, otherwise trigger values can shift (trigger value and sound file won't match). Because care is taken in the electronic design of the detection of the sinusoidal train, the only wrong synchronization could take place in the beginning. So to be sure that the system starts correctly, a couple of dummies of sinusoidal trains could be made. If the signal starts e.g. with 5 sinusoidal trains, the triggering could be checked within the EEG-software if they are detected before important real stimuli is issued. If those 5 are correctly detected, no communication issues would be expected.

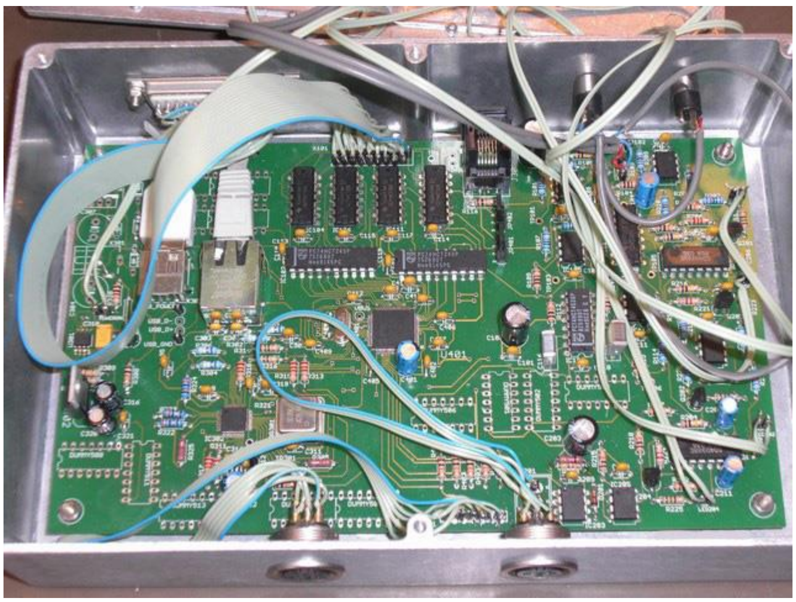

Figure 5: EventBox with internal electronics

References

[1] Idea of using sinusoidal train on second channel by Ton Wempe, http://www.fon.hum.uva.nl/wempe