Multi-subject hardware for experiment control and precise reaction time measurement

Henning Reetz and Achim Kleinmann
Department of Linguistics, University of Konstanz, Germany
E-mail: Henning.Reetz@uni-konstanz.de, Achim.Kleinmann@uni-konstanz.de

ABSTRACT

Presenting auditory or visual stimuli and recording push-button reactions with millisecond-precision is a common task in behaviouristic experiments. Programming dedicated software on a computer system or using dedicated hardware is usually time-consuming and depends usually on specific computer hardware and/or software versions. Furthermore, precise timing on a general purpose computer is problematic due to the unknown behaviour of the operating system, even with nowadays very fast computers.

The concept of a Hard-Software-System is presented that is build around a single-chip 8051-type micro-controller programmed in C, with a serial or USB interface to a 'host'-computer, inputs for up to 16 3-buttons panels with headphone outputs, a switched VGA-channel, and an audio-interface for a CD/DAT/MD-player that controls the system with sample-precise timing instructions. The presentation here is not a complete description of a hard- and software but rather the main concepts are presented to allow the self-assembly by a technician.

1. INTRODUCTION

The market for reaction-time measurement devices is a very specific niche, with very specialized devices, or it requires expensive and large industrial oriented systems whose prices are far beyond the budget of an average research project. Additionally, these devices are either not very flexible and/or require considerable technical knowledge for configuration and usage. Very few research institutions have developed affordable marketable products for this niche market, but sometimes these products rely on a very specific operating system version or are simple not maintained anymore.

We present here a rather simple system that is build around a μ-controller with a general purpose interface for audio- and video-controlling, reaction-time measurements, and an interface to a 'host'-computer. The unit is not a complete off-the-shelf product, but requires some self assembly to be configured for a specific experimental environment. The system in the present form is not meant for binaural signals because one audio-channel is used for the time-synchronous controlling of the measurements. The unit is in use since more than 10 years in various in- and outside our institute.

2. OPERATING PRINCIPLE

The main idea behind the system is to perform in synchrony with an audio-stream other activities in an experiment, like switching video-signals on or off, and measuring reaction times of push-buttons. To guarantee synchrony between the audio-stream and the stream of events, one possibility is the construction of a dedicated hardware whose timing behaviour is well known and that provides the audio- and video-information. A different approach was taken here. The central 'timing' unit is a standard audio-DAT, -CD or -MD player. (We did not test our system with MP3 or similar coding schemes, but expect that it would work with proper compression parameters selected equally well.) The tape (or disc) stores on one channel the audio-signal and on the second audio-channel controlling information for a small dedicated hardware that performs the event handling. Reaction time data is eventually transferred to a 'host'-computer which actually does *not* control the experiment but is rather controlled via the micro-controller to collect data or present visual stimuli. The interface to the controlling audio-stream is a 'normal' audio-signal and the interface to the computer is a standard RS422/232 serial cable or a USB connection. The 'host'-computer can be of any make and can run under any operating system and should only be able to receive the data from the micro-controller. Software on this computer has essentially read the data from the serial or USB port and store it on its disk. If visual presentations are required, the software on the 'host'-computer has to provide the visual stimuli. Additional hardware in the μ-controller reads push buttons, provides an audio-amplifier and a VGA-switch. The configuration of this additional hardware might vary from application to application and depends usually on the needs of the type of experiments running in a lab. Additional functionality can be added to the system (e.g. a voice-key software) but is not described here.

Figure 1 on the next page gives an overview of the configuration of the system. Chapter 3 describes the individual units in more detail and Chapter 4 gives a general description of the software. In Chapter 5 some practical considerations (problems with specific CD players we encountered) and Chapter 6 points out some additional possibilities not realised presently in our lab.

 ISBN 1-876346-48-5

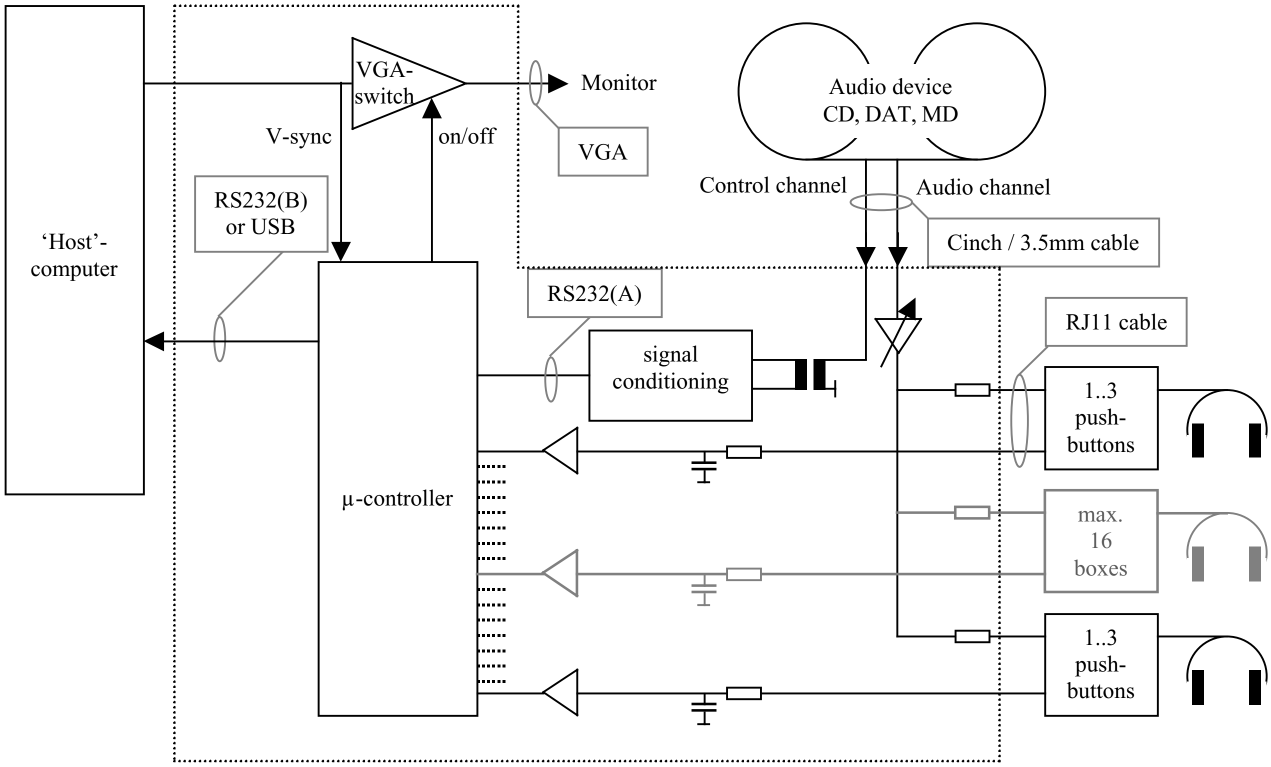

Figure 1: System overview

3. HARDWARE UNIT

The central idea behind the hardware is to use standard components, interfaces and cables. The hardware of the system is build around a industry-standard 8051-compatible μ-controller chip and an additional EPROM. This controller can be programmer with a cross-compiler under C and exists in different varieties (RS232 or USB-interfaces, nr. of parallel ports, etc.). The μ-controller receives via an signal conditioning unit (§3.1) data bytes on one of its serial ports (RS232-A) from an audio-device. These data bytes can start reaction time measurements, switch a video-amplifier on or off, or initiate a data transfer to the 'host'-computer. The controller can read up to 16 push button boxes with maximally 3 buttons each via standard RJ11 telephone cables (§3.2). The audio-signal from the audio-player is amplified and decoupled via resistors for each headphone and transmitted via the same RJ11 telephone cable (§3.3). Special care has been taken to avoid cross-talk between audio and push-button lines. The push button signals are also conditioned to suppress spurious glitches induced into the unshielded cables. A separate video amplifier can be switched on or off in synchrony with the vertical sync of the video signal (§3.4). Finally, a standard serial or USB connection to the 'host'-computer is used to transmit the acquired reaction time data. The low power consumption of the system allows battery-powered operation and care has been taken to keep analog and digital grounds separate.

3.1. CD/DAT/MD CONTROLLING INTERFACE

The controlling data stream is actually a sequence of RS232 compatible bytes that are stored on one channel of the audio device. Some care has to be taken so that the amplitude of this data is high enough to trigger the hardware and on the other hand does not cross-talk into the audio channel. Which channels to use as the audio and controlling data channel might depend in the specific DAT-, CD-, or MD-player (Oddly enough, the cross-talk from left-to-right can differ from the cross-talk right-to-left for certain players. This cross talk has to be determined for high frequencies above 10 kHz and not with 1 kHz signals as it is usually displayed in the data sheets of the players.) To handle a variety of devices we have developed two different coding schemes for the data bytes: one scheme copies the high and low bits directly on the audio device with a data rate of 9600 Baud. An alternative scheme codes the high bits as 11 kHz bursts and the low bits as silence. The former allows higher data rates but expects an audio device that handles this signal properly. (At a data rate of 9600 bits per second the signal is a high-frequency audio signal: 9600 bits/sec are equivalent to a 4.8 kHz square-wave signal with frequency components in the range of an audio recorder at 4.8 and 14.4 kHz. Even the longest possible sequence of 9 bits with the same polarity would be equivalent to an 533 Hz square wave signal. In other words, there is no danger for a pulse tilt.) The other coding scheme, the modulation with tone bursts, allows maximally 1200 baud, but we have not found a CD-, DAT-, or MD-player that did not transmit such a signal properly. In both schemes, an important point is to use a low-amplitude random noise of +/-1 sample for pauses: same audio players switch their output off if there is no signal on the audio track. Unfortunately, the logic of these players takes several milliseconds to switch the audio channel on again which leads to data loss for system described here.

The used coding schemes are easy to implement: sample values have to be set to ‘high’ or ‘low’ to generate the waveform of the audio tract for the controlling channel. The signal conditioning unit converts these into ‘low’ and ‘high’ bits for the RS232 input of the µ-controller. The timing of the data bits can directly computed from the baud rate. The duration of a high bit immediately before a low bit should be shortened by one period to compensate oscillations of the audio hardware of some players.

3.2. PUSH-BUTTON INTERFACE

The push button interface was designed to fit four constraints: (1) the connection to the subjects should be simple with standard and reliable cables, (2) the audio channels should allow mixed sets of headphones, (3) there should be no cross-talk between data-lines and audio-lines in the connection between push-button boxes and controlling system, and (4) the push-buttons cables should be resistant to inductive currents.

To fulfill these requirements, each subject receives the audio signal via a resistor from an audio-amplifier and the audio-lines and push-button lines are transmitted via a single 6-wire RJ11 telephone cable. The audio-ground and data-grounds in the cable (and the system) are kept separate to reduce cross-talk. With two data lines for the push-buttons, four conditions can be transmitted: no, left, right, or middle button pressed. Hence the reaction box can have one to three buttons. (For experiments with more alternatives we use more than one reaction time box per subject and join the data later; reaction times for so many alternatives are usually not evaluated because they reflect the time to find and press a button rather than a specific delay to make a certain decision.) The logic for the push buttons is made so that the data lines are normally on the ground level and only a pushed button leads to a ‘high’. This is achieved by using an active driver in the push-button box. This polarity scheme reduces cross talk between data- and audio-lines and reduces the effect of inductive currents because of the low impedance of this set-up. Furthermore, an additional RC-network filters short spikes and an additional TTL-buffer prevents the input of the µ-controller against high voltage spikes.

3.3. AUDIO INTERFACE

The audio-signal form the player is amplified by an integrated circuit and the output of the amplifier is decoupled by resistors for each subject to ensure that broken headphones, cables, or differences between the impedance in the headphones do not influence other headphones. The audio amplifier has its own power-supply to avoid interference with the digital signals of the systems.

3.4. VIDEO INTERFACE

For some experimental paradigms it is necessary to present visual information during the run of an experiment and collect the subjects’ reaction associated with it. The most accurate timing can be achieved by showing a complete picture at once, like presenting a diapositive through a shutter. Nevertheless, visual presentation is usually performed with cathode-ray-tubes of computer monitors. Here the picture is not displayed as a whole but the picture is assembled by a fast-moving electron-beam across a phosphorescent surface. The human eye seems to perceive a steady picture but in reality the picture is made by short glows of the activated phosphor between 60 to 120 times a second. Presenting visual and auditory information in parallel would require to know the exact position of the electron-beam on the CRT and the presentation of the auditory stimuli in relation to the beam position. The presented system is obviously not able to perform this task because both the audio-stream from the player and the video-stream from the computer are not synchronized.

Nevertheless, we choose a solution where the audio signal is only roughly related to the vertical synchronisation of the video signal. With a vertical rate of 120 Hz of modern video cards and monitors this leads to an uncertainty of +/– 4.17 ms. This uncertainty is uncorrelated with the presented stimuli and, hence, it is a sort or random noise added to the reaction times. Keeping in mind that speech signals are cut often only with the precision of +/- one glottal pulse, which results in roughly +/- 10ms precision, the additional jitter of 4.17 ms seems tolerable in comparison to the hardware overhead needed to achieve the precision which is possible with CRT displays.

It should be mentioned that flat panel displays and video-projectors (beamers) have a variety of schemes of how the picture is build up. These devices have usually a delay of one or two video frames (i.e. can have an delay of more than 30 ms) between receiving and displaying the data. With these devices there is always a (random) jitter that cannot be controlled by external hardware. It is however possible for our system to measure the actual presentation of a visual stimuli by such a unit by using a light-sensitive transistor and feeding its output as a sort of ‘push-button’ reaction into the µ-controller. Reaction times can then be related to the actual high-lighting of the display or projector.

The process of presenting visual stimuli is as follows: the µ-controller blanks the video signal with the VGA-switch and informs the ‘host’-computer about this activity. The ‘host’ computer can output now the next picture. The µ-controller switches than the video on dependent on a signal from the audio player. This procedure works well if there is enough time (around 100ms) between presenting video material. An additional handshake between ‘host’-computer and µ-controller is required if more reliability is needed at this point.

ISBN 1-876346-48-5 © 2003 UAB

3.5. COMPUTER INTERFACE

The interface to the 'host'-computer is a standard RS232/ 422 line. The µ-controller transmits at the end of each reaction time measurement the data to this computer, where the subjects' reactions can be displayed online. We use a data rate of 57.6 KBaud on this line which results in a data transfer from 16 subjects (2 byte reaction time in ms plus 1 byte reaction = 48 bytes) in 10 ms.

4. SOFTWARE

The system relies on the presentation of the auditory stimuli together with the controlling signals from an audio device. Two audio streams (usually in .WAV or .AIFF format) are generated: one with the audio signal, one with the time-aligned controlling signals. These two data streams are merged into one 'stereo' file and copied to the medium of the audio player (CD or DAT) or played through the analogue output of the computer onto the media. In the latter case, the analogue output should be able to handle the controlling signals without severe distortions. If visual stimuli are presented from the 'host'-computer, additional information must be stored and presented on the 'host' during the experiment (cf. §3.4).

4.1. EXPERIMENT PREPARATION

The audio and controlling data are generated in our Lab with a simple splicing program, that generates a stream of audio data on one channel and, related to it, the stream of controlling data on the second channel. A byte code is used for each activity in the controlling channel. The values 1 to 6 code the specific activity ('end reaction time measurement', 'switch video on', 'switch video off', 'end of experiment', 'beginning of experiment'), the values 8 to 254 signal a start of a reaction time measurement with 246 different codes to identify the items. Values 0 and 255 are unused for compatibility reasons with other hardware in our Lab. The value 7 is used as a header for multi-byte codes not used yet (e.g. to extend the number of reaction time codes or to transmit alphanumeric data via the audio device to the 'host'-computer for visual presentation). The RS232 compatible bytes are generated with the start- and stop-bits as 'low' or 'high' levels or as tone bursts, depending on the coding scheme (cf. §3.1).

All data on the controlling channel is pre-poned so that the µ-controller can start its activity at the end of a data byte in synchrony with the audio signal. For example, with the 1200 Baud coding scheme, the 'start reaction time measurement' byte is pre-poned by 8.33 ms to the related position of the audio signal. The µ-controller requires only few µ-seconds to actually start the reaction time measurement so that the end of the data-byte aligns with the beginning of the measurement.

Additionally, the audio- and controlling-signal get +/- 1 bit noise for silent signal parts to prevent players to switch off their audio lines (cf. §3.1).

4.2. EXPERIMENT RUN

During the run of an experiment, the 'host'-computer waits for data from the µ-controller, presents them on its screen and stores them on its hard disk. For the presentation of visual stimuli the 'host' displays the appropriate items on its video-output, which is usually the monitor. In this case, the experimentater sees the items and not the reaction time data, whereas the subjects see only the items when the µ-controller switches the video line on.

4.3. EVALUATION

The subjects' reactions (buttons and reaction times) are stored together with the code delivered from the 'start reaction time' command delivered from the audio-player. These data are written as normal text that can be imported into standard statistic software for subsequent evaluation.

5. PRACTICAL CONSIDERATIONS

The described system has been in use for more than 10 years in our institute in a variety of experiments. Several portable and battery powered units have been build and used in different locations around the world. (The biggest problem there were airport security and customs declaration for a small silvery box, many black push-button boxes, and long telephone cables.) The unit had never failed and our biggest concern are the audio players. We have not encountered a DAT player that did not reproduce the controlling signals correctly. CD players on the other hand have more audio-distortions, swap the polarity of the outputs, or switch of in 'silent' parts. Only informally we tested MD players and we were surprised that the hardware works also with the devices we tested despite the fact that the signals of the controlling channel look rather distorted. Using a computer (probably the 'host'-computer) is also a possibility, but our experience with the quality of most audio cards specially in laptops was rather disappointing.

6. FUTURE DEVELOPMENTS

With the disappearance of the serial RS232/422 lines we investigate the usage of USB-ports for communication to the 'host'-computer. Variants of 8051 µ-controllers have such a port build in so that we expect an easy transfer to this new technology. Furthermore, the 'host'-computer could than be replaced by a single solid state disk ('flash/pen/USB'-drive) which makes the 'host'-computer obsolete. Consequently, we investigate the replacement of the audio player by a solid-state disk and adding a high quality D/A converter to the system. In that case, the experimental set-up would consist only of the µ-controlled system. The question here is, why use an dedicated computer to run experiments and not a laptop. The answer was given in the beginning of this paper: the system is rather simple and all activities are under control of the program; no intervening and changing operating systems will influence the reliability of the acquired data.