31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

A Framework for Mouse and Keyboard Emulation in a Tongue Control System.

Morten Enemark Lund *Student Member IEEE*, Héctor A. Caltenco *Student Member IEEE*, Eugen Romulus Lontis, Henrik Vie Christiensen, Bo Bentsen, Lotte N. S. Andreasen Struijk.

***Abstract*—Effective human input devices for computer control are very important to quadriplegics and others with severe disabilities. This paper describes a framework for computer control without need for special PC software or drivers. The framework is based on a tongue control system recently developed at Center for Sensory-Motor Interaction (SMI), Aalborg University. The framework provides emulation of a standard USB keyboard and mouse, and allows tongue control of any computer using standard USB drivers available in all modern operating systems.**

I. Introduction

QUALITY OF LIFE for people with high level spinal cord injuries may be substantially improved if better and more effective computer control could be developed. Better control of and easier access to computers would allow quadriplegics to take advantage of the rapid developments within mainstream information and communication technologies. Special input devices for people with severe sensory-motor impairment are often limited in the degrees of freedom or do not provide fast and precise control of mouse and keyboard [1]. The most widely used input devices for this group of people are based on methods such as chin control, suck/puff switches, and head control [2]. Other methods such as voice recognition, eye-control, brain-computer interfaces, and tongue control systems are not as widely commercialized. All input devices have drawbacks depending on the tasks. Moreover, they usually require complex software running on the computer being controlled, which is a practical problem users often face when using any other computer than their own.

A tongue control system with 18 intra-oral sensors, activated by a tongue piercing, was recently developed at Center for Sensory-Motor Interaction (SMI) [3-5]. The high number of sensors and their special layout allows the system to be used for both typing, through use of each sensor as a key, and for directional control, by combining sensor signals to get joystick functionality.

One of the objectives of the tongue control project at SMI is to create a special input device that could be used with any computer without custom PC software. This presents some challenges because the number of possible commands (keyboard keys, mouse controls) exceeds the number of sensors in the intra-oral device. Some logic is therefore necessary to interpret the output from the intra-oral sensors and generate commands, for typing or controlling the mouse cursor.

This paper presents a framework which provides tongue control of keyboard/mouse with any computer using the USB interface. This is achieved without custom software or drivers besides the standard USB HID drivers built into modern operating system (Linux, OSX, Windows).

Manuscript received April 20, 2009. This work is supported in part by the Danish Technical Research Council, TKS A/S.

H. V. Christensen is with CISS, Department of Electronic Systems, Aalborg University, DK-9220 Aalborg, Denmark.

M. E. Lund, H. A. Caltenco, E. R. Lontis, B. Bentsen, and L. N. S. Andreasen Struijk are with Center for Sensory Motor Interaction (SMI), Department of Health Science and Technology, Aalborg University, DK-9220 Aalborg, Denmark. (Corresponding author M. E. Lund. Phone: +45 61 71 17 90; e-mail: melund@gmail.com).

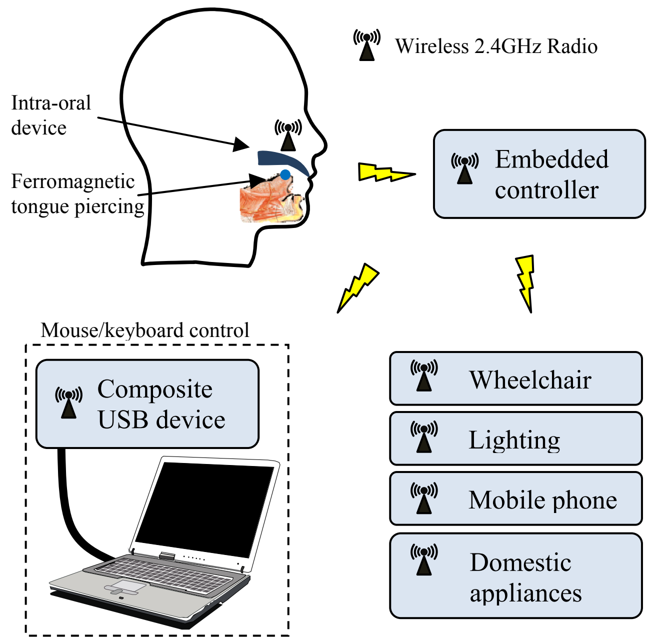


Fig. 1. Overview of the Tongue Control System. The intra-oral device transmits raw sensor readings to the external embedded controller every 33 ms. The embedded controller act as a coordinator and generates control signals for the external devices.

II. Tongue Control System

The tongue control system is based on an intra-oral device with sensors that register the movement of a ferromagnetic tongue piercing. Raw sensor data is transmitted to an embedded controller, which acts as a wireless coordinator in the system. The embedded controller processes the information and relays control commands to external devices, Fig. 1.

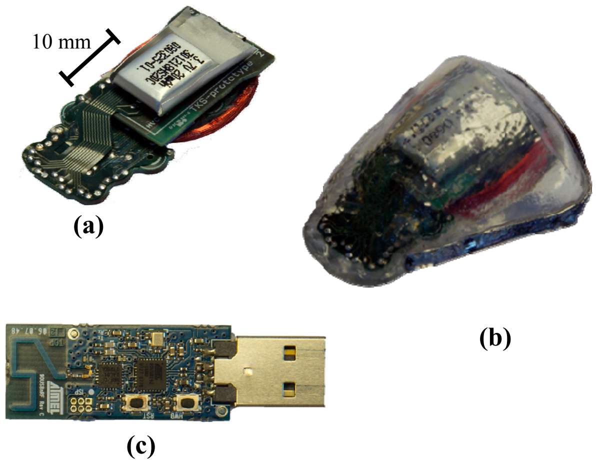


Fig. 2. Prototypes of the tongue control system. (a) Sensor plates, electronics, battery, and charging coil. (b) Encapsulated intra-oral device. Arrangement for securing the device to the teeth is not shown on the picture. (c) Board with micro controller and radio chip used to implement the composite USB device.

A. Radio communication

Radio communication between devices in the system is performed on the 2.4 GHz Industrial Scientific and Medical (ISM) band. A proprietary radio protocol is used to keep complexity and power consumption as low as possible. The embedded controller receives radio packets, containing raw sensor readings, from the intra-oral device every 33 ms (30 Hz).

B. Intra-oral device

The intra-oral device consists of 18 inductive sensors connected to a Printed circuit Board which contains a micro controller, a radio chip, and a battery, Fig. 2a. The sensors are placed in two separate areas, a key-pad area in front of the mouth which houses 10 sensors and a mouse-pad area in the back with 8 sensors, Fig. 5. The system is encapsulated in a dental retainer made of standard dental materials, Fig. 2b. Details on the intra-oral device and its use as a tongue interface is presented in [5].

III. System Framework

The framework for computer control is outlined in Fig. 3. The intra-oral device transmits raw sensor readings, and the computer only receives standard USB mouse and USB keyboard commands. All functionality is therefore defined by the embedded controller and the composite USB device attached directly to the computer.

A. Functionality in the Framework

1) Embedded controller

The embedded controller performs signal processing common to all external devices. Raw sensor readings are first filtered to remove drift and then normalized with respect to the range of the sensor in order to obtain comparable sensor readings. Furthermore, signals from sensors in the mouse-area (see Fig. 5) are combined using a Fuzzy Logic Inference System [6] to get a directional vector, similar to the output of a joystick. The Fuzzy system maps sensor signals to a two dimensional set of coordinates, which corresponds to the physical position of the tongue piercing in relation to the sensors.

The embedded controller also co-ordinates which of the external devices the user is controlling and relays processed sensor information to the appropriate device.

2) Composite USB device

The USB device is responsible for mapping sensor activation to specific USB commands such as keyboard key codes, mouse button status, and mouse movement. Details on USB implementation are not elaborated in this paper; more information can be found in [7].

Control logic responsible for the mapping is contained in two finite state machines [8]. One for keyboard emulation and one for mouse emulation (Fig. 3). The composite USB device is either in mouse mode or in keyboard mode, and commands from the embedded controller dictates which of the two state machines is used.

B. Emulating USB keyboard functionality

The keyboard functionality is implemented by assigning multiple characters to each sensor. This is somewhat similar to the multi-tap disambiguation method used in mobile phones. This poses no problem when fingers are used to feel the layout of the keys. However the same principle cannot be transferred directly to our tongue control system as it is not possible to accurately feel at which sensor the tongue piercing is located. This is analogue to typing on a mobile phone without being able to see or feel the keys.

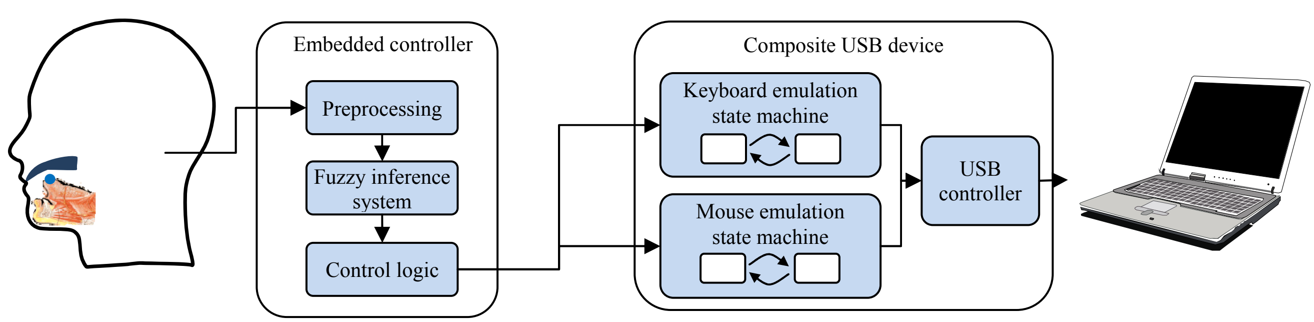


Fig. 3. Overview of framework for mouse and keyboard control. General functionality used by all external devices is performed by the embedded controller. The Composite USB device contains the logic to control the computer by sending standard USB mouse and keyboard packets. Arrows denote the flow of information from the intra-oral device to the PC.

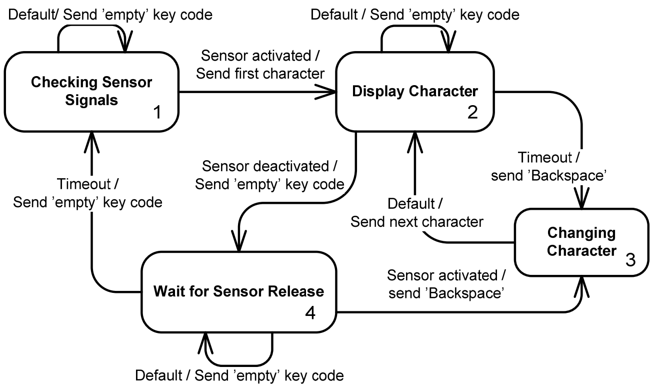


Fig. 4. Finite State Machine for keyboard emulation. A transition in the state machine is performed every time a keyboard report is sent to the PC.

1) *Visual feedback*

A mechanism is needed that allows the user to search for the correct sensor by sliding the tongue piercing from one sensor to another. This requires some kind of sensory feedback to the user (e.g. visual feedback). Mobile phones also provide visual feedback to the user. Tapping the key of a mobile phone cycles through the associated characters, allowing the user to see the alternating character before it is selected. This cannot be accomplished through the USB connection without special software running on the PC.

However, the same effect can simulated by sending a key code representing backspace when a character on screen should be changed. This method is used by the keyboard state machine to alternate characters.

2) *Keyboard state machine*

The visual feedback is handled by a finite state machine (FSM) implemented in software in the USB device. The state machine is invoked before a USB keyboard packet is sent to the PC. It adds characters to the outgoing report based on its' current state and the activated sensor (Fig. 4).

Activating a sensor causes the device to send USB keyboard packets containing the first key code associated with that particular sensor (Fig. 4: 1 ↷ 2). If the sensor is still activated after 0.5 seconds the device sends a 'backspace' key code followed by the next character (Fig. 4: 2 ⇄ 3). This continues as long as the sensor is activated, and the user sees the characters alternating on the PC. If the tongue piercing is moved directly from one sensor to another the system will start alternating characters associated with that sensor instead (Fig. 4: 2 ↷ 4 ↷ 3 ↷ 2 ⇄ 3). The user can therefore slide the tongue piercing over the different sensor to search for a certain character. When the wanted character is displayed the user deactivates the sensor, and the character stays on screen (Fig. 4: 2 ↷ 4 ↷ 1). The state machine is then ready to type the next character.

Associating sets of characters with sensors is done using a lookup table holding an array of USB keyboard key codes for each sensor. The assignment of characters to each sensor is illustrated in Fig. 5.

C. *Emulating USB mouse functionality*

Emulating mouse functionality is simpler because there are enough sensors to provide directional information and two mouse buttons and a scroll function. The mouse cursor is moved based on information from sensors 11-18 (Fig. 5) combined in a Fuzzy inference system. Moving the tongue piercing away from the center of the sensor plate moves the cursor in that direction, while the distance from center determines the speed of the cursor.

Cursor movement on screen provides visual feedback in itself thereby allowing the user to easily control the mouse. The same is not the case when emulating the mouse buttons. To ensure that the user can find the correct sensors only a few of the sensors in the key area are used, Fig. 5.

Mouse functionality is also controlled by a finite state machine, which tracks the state of the mouse buttons. Activating the sensor for the left or right mouse button for more than 3 seconds makes the mouse button stick, and allows operations such as drag and drop. Fig. 5 shows sensors associated with the mouse functionality.

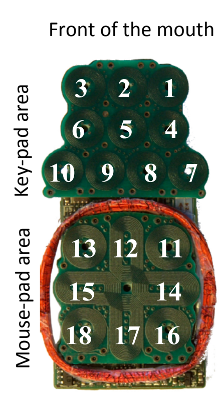
IV. Prototypes and Testing

The software prototype for the USB device was implemented on a USB dongle containing a 8-bit micro controller and 2.4GHz radio (Fig. 2c, [9].) The embedded controller was implemented on a development board also containing an 8-bit micro controller from ATMEL.

A. *Proof of concept*

An uncontrolled test showed that the system could be used to control both the keyboard and mouse in both Windows XP, Ubuntu (Linux), and OS X (Mac) using only the built-in drivers.

It was possible to type simple sentences and delete characters by activating the sensors with the help of an "activation pin", a small stick glued to the ferromagnetic part of the tongue piercing. The system was manipulated by



#	Keyboard mode	Mouse mode
1	. , ? ! ' " 1	Left click
2	a b c 2 A B C	-
3	d e f 3 D E F	Right click
4	g h I 4 G H I	-
5	j k l 5 J K L	-
6	m n o 6 M N O	-
7	p q r s 7 P Q R S	Scroll up
8	t u v 8 T U V	-
9	w x y z 9 W X Y Z	-
10	'space' 0 ↲	Scroll down
11	'Home'	Cursor
12	'Up arrow'	Cursor
13	'End'	Cursor
14	'Left arrow'	Cursor
15	'Right arrow'	Cursor
16	'Backspace'	Cursor
17	'Down arrow'	Cursor
18	'Delete'	Cursor

Fig. 5 The figure and table illustrate the mapping between the intra-oral sensors and mouse/keyboard functions

hand (outside the mouth). The mouse cursor was able to be controlled with a speed proportional to the placement of the activation pin from the center over the mouse-pad area.

V. Discussion

The framework presented in this paper provides basic control of any computer with the integrated tongue computer interface recently developed at Center for Sensory-Motor Interaction (SMI), Aalborg University. The framework was implemented as a proof of concept using standard development boards commercially available. The limited processing power of an 8-bit micro controller was sufficient for this framework but a more powerful platform may be needed. This could be used to allow the embedded controller to perform more advanced signal processing or control an LCD display which would give the tongue control system a standalone user interface.

Initial experiments with the new fully integrated and wireless intra-oral device shows promising result [5]. The system provides good typing rates with a short training period. Similar experiments needs to be performed to estimate the typing rate for the proposed framework.

Better usability, more advanced functions, customizable parameters, and faster typing rates, may be achieved using custom PC software that integrates directly with the PC's operating system. Work is currently being done on developing such a solution, which could also add word prediction and provide better visual feedback to the user. Such a solution would most likely provide the best usability for everyday use of computers and other electronic equipment in the user's home. However, the basic functionality presented in this paper provides a huge advantage in mobility and universal accessibility which is not provided by any other commercial computer-interfaces for people with severe physical disabilities.

Many factors are important when researching new assistive devices and it is important to involve actual users if outcome are to be commercially successful. The request for a system that functions without custom software comes from the user group of quadriplegics who collaborate with our research group. This paper demonstrates that such a feature can be implemented while still having the option of using a more advanced system integrated with the operating system.

References

[1] A. Craig, Y. Tran, P. McIsaac and P. Boord, "The efficacy and benefits of environmental control systems for the severely disabled," Med.Sci.Monit., vol. 11, pp. RA32-9, Jan. 2005.

[2] R.G. Platts and M.H. Fraser, "Assistive technology in the rehabilitation of patients with high spinal cord lesions," Paraplegia, vol. 31, pp. 280-287, May. 1993.

[3] E.R. Lontis, Struijk, Lotte N. S. Andreasen, H.A. Caltenco, H.V. Christiensen , B. Bentsen and M.E. Lund, "Inductive Pointing Device for Tongue Control System for Computers and Assistive Devices," in EMBC 2009. 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2009.

[4] H.A. Caltenco, Struijk, Lotte N. S. Andreasen, E.R. Lontis, H.V. Christiensen , B. Bentsen and M.E. Lund, "Character Activation Time Prediction Model for Tongue-Typing: Adaptation of Fitts's Law," in EMBC 2009. 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2009.

[5] Struijk, Lotte N. S. Andreasen, H.A. Caltenco, E.R. Lontis, H.V. Christiensen , B. Bentsen and M.E. Lund, "Fully Integrated Wireless Inductive Tongue Computer Interface for Disabled People (Submitted for publication)," in EMBC 2009. 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2009.

[6] MathWorks. "Simulating Fuzzy Inference Systems using the Fuzzy Inference Engine," April 2009; http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/fp571.html.

[7] USB Implementers' Forum, "Device Class Definition for Human Interface Devices (HID)," usb.org., Tech. Rep. Version 1.11, 2001.

[8] P. Van der Linden, Expert C programming: deep C secrets, Englewood Cliffs, N.J. : SunSoft Press, 1994, pp. 353.

[9] ATMEL. "ATAVRUSBRF01 USB 2.4 GHz Reference Design," April 2009; http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4322.