Smart Buttoning: Assisting the dressing activities of Alzheimer patients

Andrei Popleteev, Oscar Mayora
CREATE-NET
Trento, Italy
firstname.lastname@create-net.org

***Abstract* – This paper presents a wearable computing approach to assisting dressing activities of elderly and patients with cognitive disorders, who experience difficulties with buttoning their clothes correctly.**

Keywords – ambient assisted living; tangible interaction; wearable computing.

I. Introduction

According to occupational therapists who work with Alzheimer patients, people with mild cognitive decline experience significant difficulties with "dressing" [1] and, more specifically, "buttoning" activities [2]. The main problem states in the fact that these patients cannot button properly while dressing and this results in problems of loosing self-confidence by the patients, with direct consequences on inhibiting their will to go out to public places and interact with other people, creating a state of depression.

This project targets the problem of incorrect buttoning of clothes typical for people with cognitive decline. The incorrect buttoning happens mainly by mismatching the snapping of buttons. The proposed "Smart Buttoning" system solves this problem by detecting if a button is locked with its correct counterpart. In case the button is not locked or locked with a wrong counterpart an event is triggered and the system can provide an alert (sound, vibration, light, message, etc.) and/or record the event details in an embedded memory for further analysis by caregivers.

The originality of the proposed approach is that the only current solution relies on the direct observation of the therapists on how patients perform the buttoning operation. The proposed system, in contrast, performs the monitoring in a discreet and automatic way. The main advantage of the system is that cognitive-decline patients do not require external supervision to button clothes correctly.

The paper is organized as follows. The next section provides a review of the related work. Then, we introduce the smart buttoning approach and its prototype implementation. The final summarizing section concludes the paper.

II. Smart buttoning approach

This section describes three possible varieties of smart buttoning: connectivity-based, resistance based, and the one utilizing standalone smart buttons. The following sections describe these solutions in detail.

A. Connectivity-based approach

This approach comprises a garment with an embedded electronic device. The buttons are represented by a pair of metal parts which become electrically connected when fastened, and not connected otherwise. Both contact pairs of each button are connected to the processing unit via wires hidden in the cloth's textile (see Figure 1). When the patient dresses himself/herself, he/she puts the buttons into a contact. If the buttons are mismatched, – that is, a wrong pair

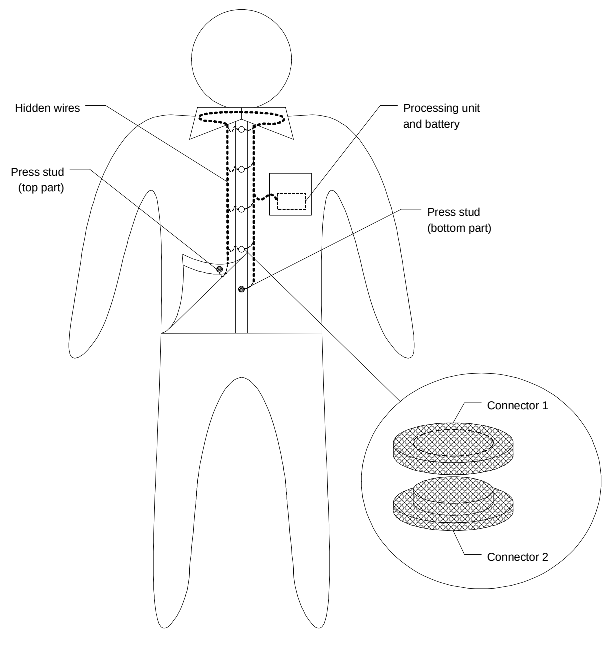

Figure 1. General view.

of contacts is connected – the processing unit detects the mismatch and produces an audible (visible, tactile) alert, prompting the patient about the issue. If, on contrary, the dress is buttoned correctly, no feedback is provided.

A schematic representation of the device's general architecture is presented in Figure 2. The device consists of a microcontroller unit (MCU), a loudspeaker, and a battery. As the input, the MCU has contact pairs of the fasteners. In our prototype implementation (Figure 3), the MCU is an Atmel ATMEGA8 [3], clocked at 1 MHz by its internal RC-generator. Due to the limited number of available input/output (IO) ports, the device is capable of monitoring

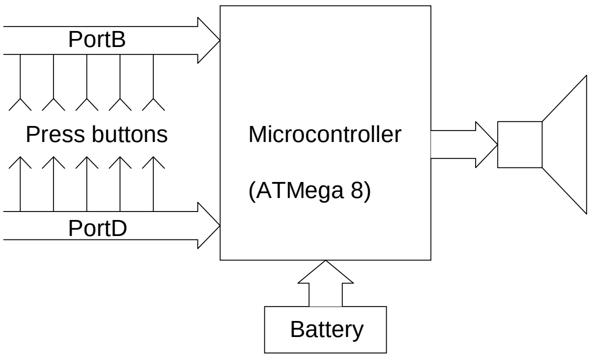

Figure 2. Device architecture.

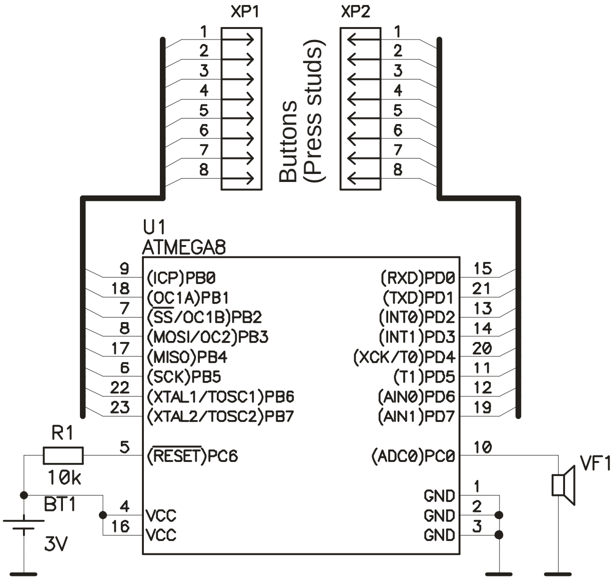

Figure 3. Electric scheme of the prototype device.

up to 8 button-pairs. While this number is sufficient for most cases, it can also be easily extended by a employing a MCU with more IO ports, if necessary.

The device is programmed to perform the algorithm presented in Figure 4. Upon startup, the microcontroller initializes its IO ports, in particular PortB and PortD, that are used to detect connections between buttons. All pins of PortB are configured as outputs, all pins of PortD – as inputs with internal pull-up resistors. The latter fact ensures high logical level on disconnected pins of PortD. Then the device produces a short beep to indicate the start of the working phase. The main loop of the program consists of three general steps: checking whether buttons' connections match, producing an alarm if appropriate, and ultimately putting the MCU into a sleep mode to preserve power. An internal timer periodically wakes the MCU up to check buttons' state.

The buttons check is performed one-by-one for each pair by "running zero" scan on PortB (see Figure 4). With this approach, one and only one pin of PortB is set to low logical level ('0') at any moment, while other pins are set to high

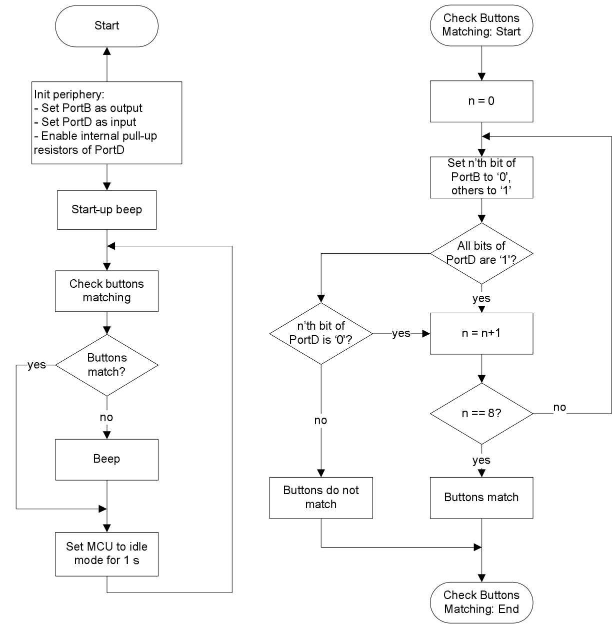

Figure 4. Detection algorithm.

logical level ('1'). In the case when no buttons are connected, all pins of PortD remain '1' due to MCU's internal pull-up resistors. When N'th button pair is properly connected, setting N'th pin of PortB to '0' will set N'th pin of PortD to '0', too. However, if the N'th button is connected to a wrong counter-part (e.g. M'th), setting N'th pin of PortB will result in '0' on M'th pin of PortD. This enables the program to detect buttons' mismatch and to appropriately inform the user.

The presented algorithm checks only if the fastened buttons match correctly, but the fastening itself is optional. However, the algorithm can be easily updated to detect the cases of unfinished fastening and suggest the user to complete the process. Some button pairs might still remain optional (e.g. top button can be unfastened for hot weather).

B. Resistance-based approach

Another variety of the smart buttoning solution is presented in Figure 5. There, every combinations of connected buttons results in a unique value of the total resistance Rx (see Table 1). This resistance is measured by MCU's internal analog-to-digital converter (ADC). When no buttons are fastened, the Rx value is very high and most of the time MCU spends in idle mode. When some buttons are connected, and the total resistance Rx becomes finite, the MCU checks if Rx value corresponds to a correct connection pattern (e.g. by searching through a predefined table of valid values). If Rx does not correspond to any of valid connection patters, the user is informed about the buttons mismatch.

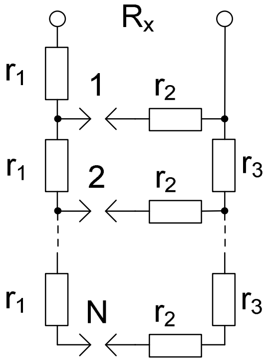

Figure 5. Electric scheme of the resistance-based method.

In comparison to the connectivity-based approach, this method does not require extensive wiring of the patient's garment, but has a higher computational complexity and is more sensitive to the buttons' contact quality.

C. *Standalone smart button approach*

The major limitation of the previous approaches is the need for additional wiring connecting buttons with the processing unit. The standalone button approach solves this issue by integrating the processing unit into the button itself. In this case, each button consists of two parts: top and bottom (Figure 6). These parts have a pair of contact pads which become connected when the button is fastened. The bottom part comprises a processing device module which, in turn, includes a battery, a piezosounder and a microcontroller (MCU) with an internal analog-to-digital converter (ADC). The top part contains an internal (hidden) resistor Rn attached to the contact pads. The resistance value of the matching top part is stored in the MCU's firmware and is unique for each button pair.

The device operates as shown in Figure 7. Firstly, the microcontroller (MCU) initializes its internal ADC and uses it to measure the resistance Rx between the contact pads of the bottom part. When the button is not fastened, Rx has a large value (over 1MOhm). When the button is fastened, the measured Rx value is equal to the Rn value of the attached top part. If the measured resistance does not correspond to the Rn value stored in MCU's firmware, – this means that a wrong button pair is fastened, and the user is warned by a sound alarm. Finally, to preserve the battery power, the MCU switches to the idle mode for one second; after that the cycle repeats.

Besides the contact pads connecting the top and bottom

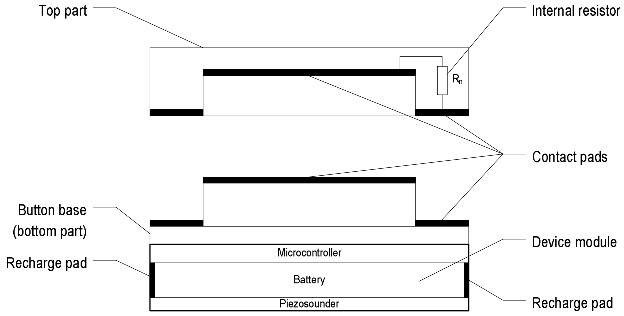

Figure 6. Standalone smart button structure (side view).

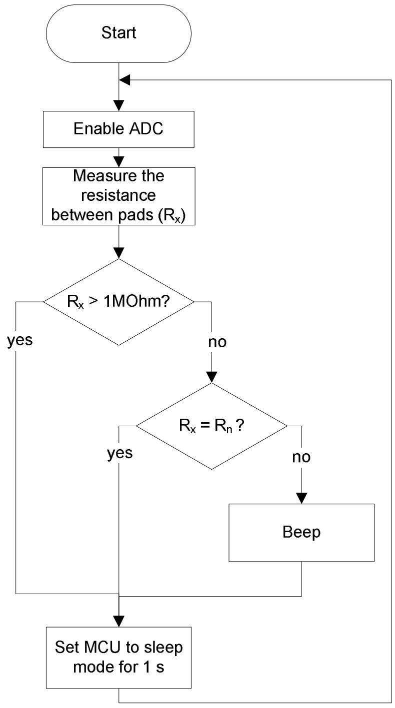

Figure 7. Standalone smart button algorithm.

TABLE I. TOTAL RESISTANCE FOR VARIOUS CONNECTION PATTERNS.

Connection pattern	Corresponding R_x value
Buttons 1 and 2 are not connected	∞
Button 1 connected, Button 2 not connected	r_1+r_2
Button 1 not connected, Button 2 connected	$2r_1+r_2+r_3$
Buttons 1 and 2 both correctly connected	$r_1+\frac{r_2(r_1+r_2+r_3)}{r_1+2r_2+r_3}$
Left part of Button 1 erroneously connected to the right part of Button 2	$r_1+r_2+r_3$
Left part of Button 2 erroneously connected to the right part of Button 1	$2r_1+r_2$
Both buttons connected erroneously (combination of the two above cases)	$r_1+\frac{(r_1+r_2)(r_2+r_3)}{r_1+2r_2+r_3}$

parts, the smart button also includes a pair of pads for battery recharging. Except the pads, the bottom part is completely covered by a waterproof compound, so that the cloths can be washed without the risk of damaging the smart buttons attached to it.

III. Conclusion

In this paper we presented a system for supporting buttoning operations of people with mild cognitive decline. The system includes a wearable device integrated with patient's dress; in case of mismatch in buttoning, the device provides alarms of different modalities (audio, vibration, voice messages, visual feedback). In contrast to the current practice of direct observation by caregivers, the proposed solution is automatic and provides the patients with higher independence, while preserving their privacy.

The future plans include minimization of the device module dimensions and performing a user experience study of the smart buttoning system.

References

[1] A. Matic, P. Mehta, J. M. Rehg, V. Osmani, and O. Mayora, "AID-ME: Automatic Identification of Dressing failures through Monitoring of patients and activity Evaluation", Proc. Int. Conf. Pervasive Computing Technologies for Healthcare (Pervasive Health 2010), Munich, Germany, 2010.

[2] G. Miceli and R. Capasso, discussion of occupational therapy needs and issues. Trento, Italy, 2008.

[3] Atmel ATMEGA 8 datasheet.
Online: www.atmel.com/atmel/acrobat/doc2486.pdf