Task Specific Paper Controller that Can Be Created by Users for a Specific Computer Operation

Daisuke Komoriya(✉), Buntarou Shizuki, and Jiro Tanaka

University of Tsukuba, Tsukuba, Japan
{komoriya,shizuki,jiro}@iplab.cs.tsukuba.ac.jp

Abstract. We describe Paper Controller, a paper based controller that allows users to design and create their own task specific controllers with touch-sensing capability for controlling a desktop computer. Casual users of computers can design and create a task specific Paper Controller by printing and/or drawing buttons freely with conductive ink and by drawing annotations including text and figures with regular ink. We implemented a prototype system for the Paper Controller. The system consists of the Paper Controller, a Clipboard for the Paper Controller, and a parameterization software running on a computer. We conducted an experiment to examine whether users can create a Paper Controller. The results show that the users can create and use their own Paper Controllers.

Keywords: Paper controller · Prototyping · Conductive ink · Capacitive sensing

1 Introduction

While buttons on the controllers of consumer electronics, including desktop computers, are necessary to execute all the available functions of the electronics, the controllers suffer from a lack of usability in some cases; for example, there are too many buttons on the controller, making it difficult for certain users to perform the user's specific daily tasks.

One supporting example is shown in Fig. 2. This task specific controller is designed for casual users of consumer electronics, such as for a grandmother to control her television. Moreover, although this photograph might seem comical, examining this controller suggests two designs principles:

- A limited number of buttons is necessary for a task specific controller.
- Annotation improves the usability of a task specific controller.

N. Streitz and P. Markopoulos (Eds.): DAPI 2015, LNCS 9189, pp. 418–428, 2015.
DOI: 10.1007/978-3-319-20804-6_38

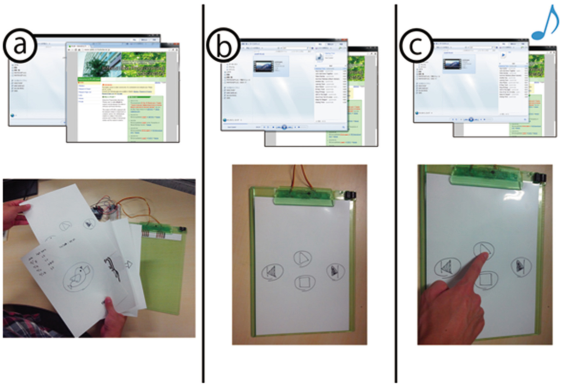

Fig. 1. How to use a Paper Controller. (a) The user selects a Paper Controller that suits the specific task that she wants to perform among her own Paper Controllers. (b) She then clips the Paper Controller on a Clipboard for Paper Controllers. (c) Now, she can use the Paper Controller by touching the buttons on it.

A possible solution to the above problem would be a controller designed to be suitable for a specific user to perform the user's specific task; if one can design and create such a controller easily, the problem will be solved.

In this paper, we present a novel approach to solve the above problem, which involves these two design principles. This approach uses paper with touch-sensing capability, which we call *Paper Controller*. Casual users of computers can design and create a task specific Paper Controller by printing and/or drawing buttons freely with conductive ink and drawing annotations including text and figures with regular ink. It is low cost and easy to duplicate and distribute because a Paper Controller consists of a sheet of paper and conductive ink. Additionally, a

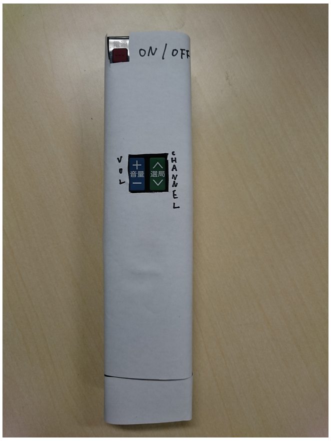

Fig. 2. Most basic television remote controller for casual users of consumer electronics.

Paper Controller can be printed out if the user has its data. Moreover, the user can easily store many Paper Controllers with a clipboard or in a folder. As the first step to explore the feasibility of the Paper Controller, we focused on Paper Controllers for controlling a desktop computer.

2 Related Work

2.1 Enhanced Paper

Similar to our research that enhanced paper to create interactive controllers, some researches enhanced paper to create user interfaces. Koike et al. [1] proposed EnhancedDesk, which is a desk that integrates information on a sheet of paper and a computer. The desk allows a user to touch the paper to interact with it by computer vision. A paper control panel [2] is a camera based paper controller. It allows the user to operate a computer by touching shapes drawn on normal paper. In contrast to these researches that enhanced paper to allow interaction with paper through cameras, we use touch sensors instead of cameras, which allows for little setup with occlusion-free touch interactions.

Some researches enhanced paper by using conductive ink and electronic components put on the paper. Qi et al. [3] enhanced paper by using traditional pop-up mechanisms and paper-based electronics. Lo et al. [4] proposed electronic circuits that allow sketching and shrinking. Untoolkit [5] combined micro controllers with a craft drawing circuit on paper with conductive ink. Saul et al. [6] proposed interactive paper devices that users can create and build their own designs. In contrast to the above research, we use conductive ink to make touch-sensitive paper. Moreover, in our research, we do not use electronic components on the paper. Therefore, a Paper Controller is flat, thin, and easy to duplicate and distribute.

Conductive ink has been focused on and used in some researches as a material for building interactive paper-like functional objects. Karagozler et al. [7] proposed a technology for harvesting energy from users' interactions with paper-like materials. PrintScreen [8] is a technology for digital fabrication of customized flexible displays. Both are composed of various materials including conductive ink. Jacoby et al. [9] proposed interactive paper for storytelling. When a user touches a part of a paper, the system detects it and plays the story. Olberding et al. [10] proposed a cuttable multi-touch sensor. Both use conductive ink for building touch sensors on paper. Similarly, we use conductive ink to allow users to design touch-sensitive paper.

2.2 GUI as Freehand Drawings

Similar to our research, some researches used freehand drawings to design GUIs. For example, Coyette et al. [11] proposed a converter that converts freehand drawings into XML code. This allows users to create GUIs by drawing figures. UISKEI [12] is a sketch based GUI prototyping tool for designers. It converts sketches into GUI components such as buttons, checkboxes, and textboxes. In contrast, our research uses freehand drawings to create touch interfaces.

2.3 Customizable Controller

Various methods to fabricate customizable controllers have been proposed. Villar et al. [13] proposed a malleable structure on which users can freely arrange controls, such as buttons, sliders, dials, and joysticks, and assign functionality to each control. Holman et al. [14] used conductive tape to customize everyday objects into touch-sensitive controllers. Corsten et al. [15] changed everyday objects into controllers by using cameras. Reference [16–18] allow users to design a physical controller by assigning an operation such as a shortcut key to a physical controller. Klemmer et al. [19] proposed an everyday object input interface that uses a RGB camera and RFID. In contrast, our research uses touch sensors drawn/printed by using conductive ink on a sheet of paper to fabricate customizable controllers, whose sizes and shapes can be designed freely, providing users with a large degree of freedom in designing.

3 Paper Controller

The Paper Controller is a paper-based controller. Each Paper Controller is designed to be suitable for a specific user to perform the user's specific task on a desktop computer.

3.1 Using

To use a Paper Controller, the user selects a Paper Controller that is suitable to the specific task that he or she wants to perform among many of their own Paper Controllers, and clips it on a Clipboard made for Paper Controllers that is a specialized clipboard for our system. Then, the system detects the clipped Paper Controller by reading the ID printed on it and makes the windows necessary for the task active. Now, the user can use the Paper Controller by touching the buttons on it. Every time a button is touched, the system sends the assigned commands to the windows. The user can easily perform another task by simply changing the Paper Controller on the clipboard.

Figure 1 shows an example use case. Here, the user is going to play songs from their favorite musician. Assume that the user is currently performing another task (Fig. 1a). When the user clips the Paper Controller assigned to playing songs, the system detects the clipped Paper Controller and makes the media player active. After that, the system opens the assigned playlist (Fig. 1b). When the user touches the play button on the Paper Controller, the media player plays the songs from the play list (Fig. 1c).

3.2 Designing and Creating

To allow a user to design and create a Paper Controller, we provide the parameterization software that will be used on a desktop computer. After the user prints and/or draws buttons freely by using conductive ink on a Paper Controller, the user can easily assign an operation (e.g., opening a specific file, opening a specific play list, or jumping to a web site with a specific URL) to each button on the Paper Controller by using the GUI of this parameterization software.

4 Implementation

Our prototype system consists of the Paper Controllers, a Clipboard for the Paper Controller, and the parameterization software running on a computer.

4.1 Paper Controller

A Paper Controller has buttons and pins printed by using conductive ink (we use AgIC silver nano-particle ink [20] in our current implementation). Figure 3 shows both sides of a Paper Controller. The user draws freehand drawings including figures and text on the front side by using regular ink. The buttons drawn on the back side correspond to the figures on the front side. There are two kinds of pins printed on the back side. One kind is pins for the ID (the four pins in the top left in Fig. 3 right). The ID is binary encoded by using the pins (the short-circuited pins mean 0s; the other mean 1s; the ID is 010 in Fig. 3 right). The others kind is pins for buttons (the five pins in the top right in Fig. 3 right). Each button is connected to a pin.

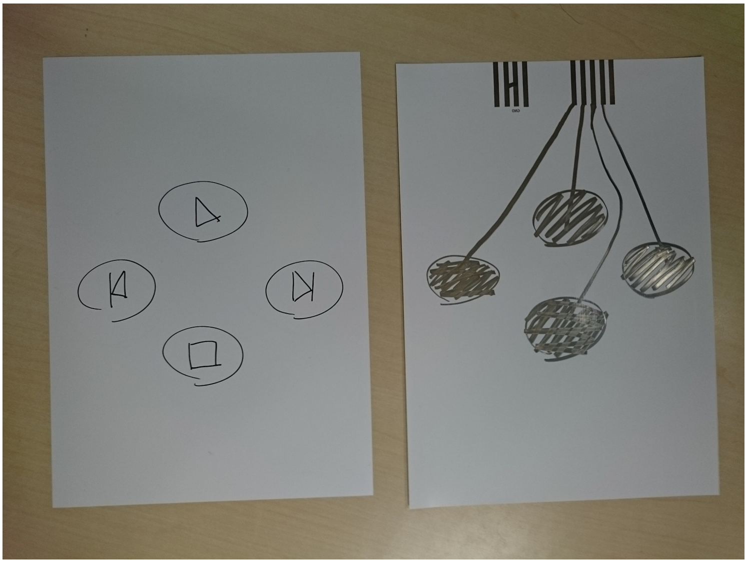

Fig. 3. Both sides of a Paper Controller (left: front side; right: back side).

4.2 Clipboard for Paper Controllers

The clip of the Clipboard for the Paper Controller also has pins that are connected to the pins on the back of a Paper Controller (Fig. 4). All the pins of the clip are connected to a micro controller (Arduino) to read the ID and detect button touches. The clipboard sends an event to the computer when a Paper Controller is changed or a touch is detected. When a Paper Controller is changed, the clipboard sends the "change Paper Controller" event along with the Paper Controller ID. We use capacitive touch sensing [21] to detect the touches. When the micro controller detects a touch, it sends a touch event to the computer with the Paper Controller ID and the button ID.

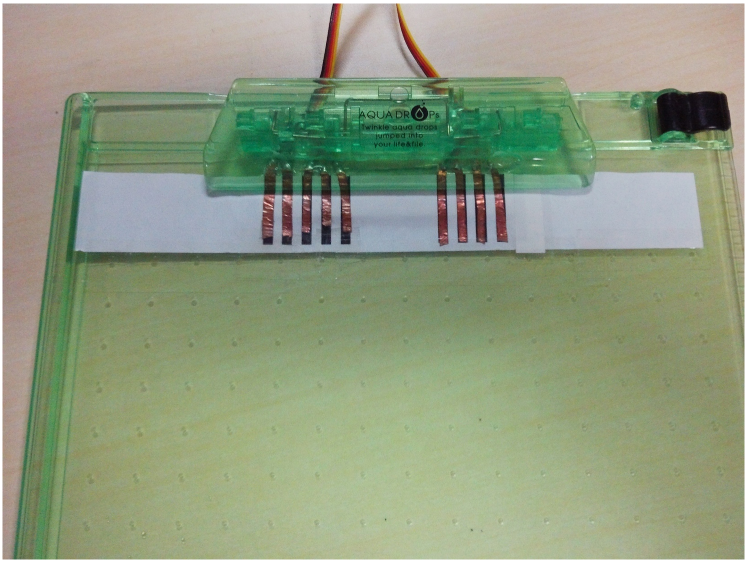

Fig. 4. The clip of the Clipboard for the Paper Controller.

4.3 Parameterization Software

We implemented parameterization software to assign a button of a Paper Controller to a window and its commands (we use shortcut keys in our current implementation). This parameterization software is built on the .NET Framework 4.0. When the parameterization software receives a touch event from the clipboard, it sends the assigned shortcut keys to the assigned window.

5 Experiment

We conducted an experiment to examine whether and how users can create a Paper Controller.

5.1 Participants

Four Participants (3 males and 1 female) ranging in age from 22 to 23 took part in this experiment as volunteers. All of them are majoring in Computer Science. Three of them were graduate school students; one was an undergraduate.

5.2 Apparatus

We used a personal computer (Lenovo ThinkPad X201i; Intel Core i3 processor and 4 GB of RAM) running the parameterization software, sheets of paper (Mitsubishi Paper Mills NB-WF-3GF100) on which we printed pins to connect to the clipboard (Fig. 5), and a conductive ink pen (AgIC Circuit Marker) which is a pen containing AgIC silver nano-particle ink.

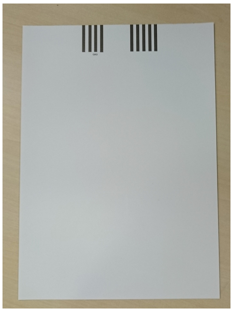

Fig. 5. Paper Controller with pins already printed to connect with the clipboard.

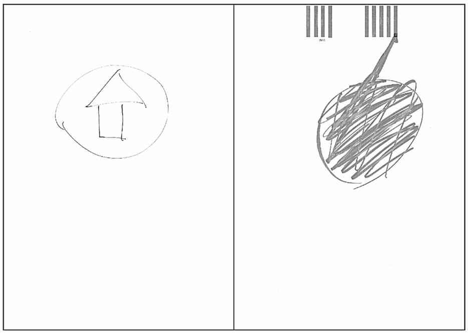

Fig. 6. Paper Controller designed as an example (left: front side; right: back side).

5.3 Procedure

We explained the concept of the Paper Controller and handed them a Paper Controller that we had already designed as an example (Fig. 6). After that, the participants used the Paper Controller for approximately 5 min to play Flappy Bird[1], a one-button game running inside a web browser, to familiarize themselves with the concept of the Paper Controllers. We used Internet Explorer 11 as the web browser. Then, we explained how to create a Paper Controller. First, draw figures on the front side. Second, draw buttons and lines from each button to a pin to connect them with the conductive ink pen. Third, clip the Paper Controller to the clipboard. We note that, in this experiment, the shortcut key was already assigned by us by using the parameterization software. After that, the participants created their own Paper Controllers. During the experiment, we answered questions from the participants. After that, we asked them to complete the questionnaires. Each participant took about 15 min to complete this experiment.

[1] http://www.freeflappybird.org/.

5.4 Results

Figure 7 shows the Paper Controllers that were created by the participants. All figures were drawn on the upper half of the sheet. This may be caused by the pins being placed at the top of the Paper Controller.

P1 drew a character that represented a bird and a triangle as the button for flapping up. The task in this experiment is a game. In this game, users tap a button shaped like a bird to make the character flap upward. Therefore, P1 drew a triangle to flap up. P2 drew five circles that represent the places of the fingers. In P4's experiment, the line that connects the pin to the button was too thin at first, therefore the touch sensor did not work. We asked him to draw a thicker one. After that, the touch sensor worked.

5.5 Comments

We collected the following comments from the questionnaires:

It is difficult to draw a circuit on the back side to make it match the button on the front. (P1)
Since the paper we used in this experiment was thick, the participants could not see through to the front side when the paper was placed on the table. We observed that some of the participants held the Paper Controller up to look through to the front side.

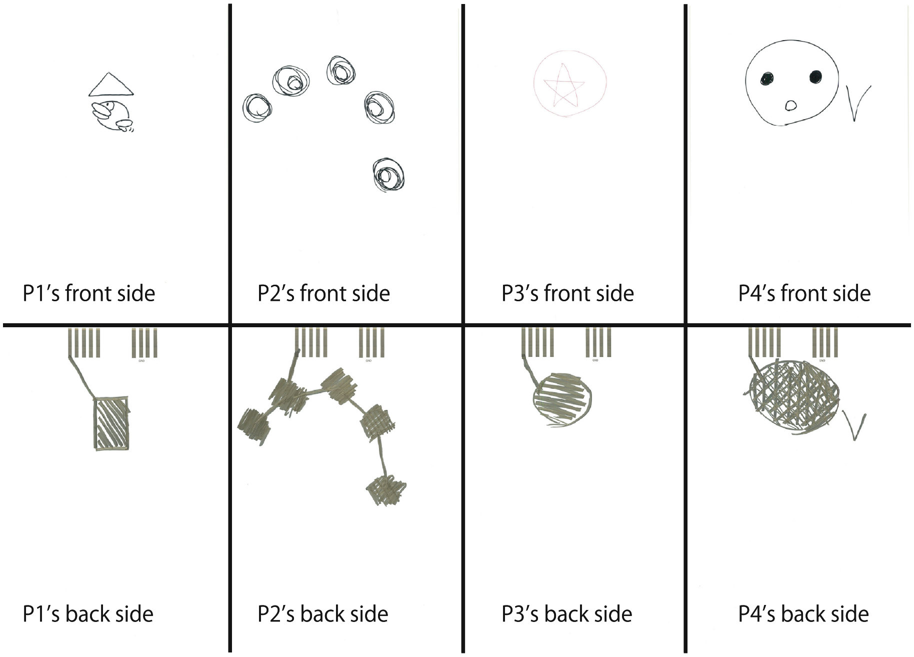

Fig. 7. Participants' Paper Controllers. The back side is mirrored to improve visibility.

It is good that I can draw something that I like. (P1, P2, and P3)
Paper Controllers created by participants affirmed this comment as the four Paper Controllers had quite different designs from each other in spite of the fact that all of them had the same functionality.

A user doesn't need to remember shortcut keys. (P3)
Since shortcut keys are assigned to buttons, only the user of the parameterization software needs to know the shortcut keys.

6 Discussion and Future Work

6.1 Experiment

As the first step to explore the feasibility of the Paper Controller, we conducted an experiment where we asked participants whose major is Computer Science, i.e., having satisfactory knowledge of computers, to create a Paper Controller after we show an example. Therefore, to further examine the feasibility of the Paper Controller, we plan to conduct experiments where we ask participants not majoring in Computer Science to freely design their own Paper Controllers.

6.2 Authoring Editor

As we described above, P4 had a problem where the line was too thin during the experiment. To solve this problem, we plan to implement a authoring editor for the Paper Controller that helps users lay buttons out more easily. Ideally, the user would scan the front side of a Paper Controller. After that, the editor would create the circuits. After that, the circuits can be printed out by using printer with conductive ink.

6.3 Realizing Other Touch Gestures

In our prototype, we used conductive ink to create capacitive touch sensors. This makes it possible for users to draw touchable buttons and to use printers to print them. However, we consider that conductive ink provides us with many chances to design and implement various touch gestures and GUIs, thus improving the functionality and usability of the Paper Controller. For one approach, we will consider providing not only simple touch but also other touch gestures, such as tap and flick; this can be realized by printing specialized patterns of conductive ink [22]. For another approach, we also plan to implement sliders and knobs by also printing specialized patterns. Moreover, we plan to reduce the number of connectors by using the technique described in [23] and to investigate other touch sensing techniques such as TempTouch [24] to enrich touch gestures.

6.4 Target

In this paper, the Paper Controller was used for controlling a window based GUI as the first step to explore the feasibility of the Paper Controller because window based GUIs are popular and complicated. As the next step, we plan to improve the system of the Paper Controller to apply it to consumer electronics including TV, air conditioners, and microwaves, therefore we plan to add an IR remote control function to the clipboard.

7 Conclusion

We presented Paper Controller, a paper based controller that allows users to design and create their own task specific controllers with touch-sensing capability for controlling a desktop computer. We conducted an experiment to examine whether and how users can create a Paper Controller. The result shows that users can create and use their own Paper Controllers.

References

1. Hideki, K., Yoichi, S., Yoshinori, K.: Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system. ACM Trans. Comput.-Hum. Interact. **8**(4), 307–322 (2001)
2. Kaneko, M., Tanaka, J.: Paper control panel: making paper-based touch interface. Proc. Interact. **2014**, 562–567 (2014). (In Japanese)
3. Qi, J., Buechley, L.: Electronic popables: exploring paper-based computing through an interactive pop-up book. In: Proceedings of TEI 2010, pp. 121–128 (2010)
4. Lo, J., Paulos, E.: ShrinkyCircuits : sketching, shrinking, and formgiving for electronic circuits. In: Proceedings of UIST 2014, pp. 291–299 (2014)
5. Mellis, D.A., Jacoby, S., Buechley, L., Perner-Wilson, H., Qi, J.: Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an "untoolkit". In: Proceedings of TEI 2013, pp. 83–90 (2013)
6. Saul, G., Xu, C., Gross, M.D.: Interactive paper devices: end-user design and fabrication. In: Proceedings of TEI 2010, pp. 205–212 (2010)
7. Karagozler, M.E., Poupyrev, I., Fedder, G.K., Suzuki, Y.: Paper generators: harvesting energy from touching, rubbing and sliding. In: Proceedings of UIST 2013, pp. 23–30 (2013)
8. Olberding, S., Wessely, M., Steimle, J.: PrintScreen: fabricating highly customizable thin-film touch-displays. In: Proceedings of UIST 2014, pp. 281–290 (2014)
9. Jacoby, S., Buechley, L.: Drawing the electric: storytelling with conductive ink. In: Proceedings of IDC 2013, pp. 265–268 (2013)
10. Olberding, S., Gong, N.-W., Tiab, J., Paradiso, J.A., Steimle, J.: A cuttable multitouch sensor. In Proceedings of UIST 2013, pp. 245–254 (2013)
11. Coyette, A., Faulkner, S., Kolp, M., Limbourg, Q., Vanderdonckt, J.: SketchiXML: towards a multi-agent design tool for sketching user interfaces based on USIXML. In: Proceedings of TAMODIA 2004, pp. 75–82 (2004)
12. Segura, V.C.V.B., Barbosa, S.D.J., Simões, F.P.: UISKEI: a sketch-based prototyping tool for defining and evaluating user interface behavior. In: Proceedings of AVI 2012, pp. 18–25 (2012)

13. Villar, N., Gellersen, H.: A malleable control structure for softwired user interfaces. In: Proceedings of TEI 2007, pp. 49–56 (2007)
14. Holman, D., Vertegaal, R.: TactileTape: low-cost touch sensing on curved surfaces. In: Proceedings of UIST 2011 Adjunct, pp. 17–18 (2011)
15. Corsten, C., Avellino, I., Möllers, M., Borchers, J.: Instant user interfaces: repurposing everyday objects as input devices. In: Proceedings of ITS 2013, pp. 71–80 (2013)
16. Hudson, S.E., Mankoff, J.: Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. In: Proceedings of UIST 2006, pp. 289–298 (2006)
17. Greenberg, S., Boyle, M.: Customizable physical interfaces for interacting with conventional applications. In: Proceedings of UIST 2002, pp. 31–40 (2002)
18. Spiessl, W., Villar, N., Gellersen, H., Schmidt, A.: VoodooFlash: authoring across physical and digital form. In: Proceedings of TEI 2007, pp. 97–100 (2007)
19. Klemmer, S.R., Li, J., Lin, J., Landay, J.A.: Papier-Mâché: toolkit support for tangible input. In: Proceedings of CHI 2004, pp. 399–406 (2004)
20. Kawahara, Y., Hodges, S., Cook, B.S., Zhang, C., Abowd, G.D.: Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of ubicomp devices. In: Proceedings of UbiComp 2013, pp. 363–372 (2013)
21. Arduino Playground - CapacitiveSensor. http://playground.arduino.cc/main/capacitivesensor. Last accessed: February 2015
22. Manabe, H., Inamura, H.: Single capacitive touch sensor that detects multi-touch gestures. In: Proceedings of ISWC 2014, pp. 137–138 (2014)
23. Wimmer, R., Baudisch, P.: Modular and deformable touch-sensitive surfaces based on time domain reflectometry. In: Proceedings of UIST 2011, pp. 517–526 (2011)
24. Peiris, R.L., Nakatsu, R.: TempTouch: a novel touch sensor using temperature controllers for surface based textile displays. In: Proceedings of ITS 2013, pp. 105–114 (2013)