ActPad– A Smart Desk Platform to Enable User Interaction with IoT Devices

Sarah Delgado Rodriguez
sarah.delgado@unibw.de
Bundeswehr University Munich
LMU Munich
Germany

Sarah Prange
sarah.prange@unibw.de
Bundeswehr University Munich
LMU Munich
Germany

Lukas Mecke
lukas.mecke@unibw.de
Bundeswehr University Munich
LMU Munich
Germany

Florian Alt
florian.alt@unibw.de
Bundeswehr University Munich
Germany

Figure 1: We present *ActPad*, a smart desk platform to enable user interaction with IoT devices. In particular, *ActPad* enables interaction via *touch points*, attached to arbitrary objects via connectors (a). Touch points may be on stationary (e.g., Android decoration (b)) and movable (e.g., a coffee cup (c)) objects as well as on the pad itself (d). With *ActPad*, various use cases can be realized, such as turning on lights (d), switching to a next song (e) or triggering the coffee machine to heat up (f).

ABSTRACT

ActPad is a desk pad, capable of sensing capacitive touch input in desk setups. Our prototype can sense touches on both, its electrodes and on connected objects. *ActPad*'s interaction-space is customizable, allowing easy integration and extension of existing desk environments. In smart environments, users may interact with more than one device at the same time. This generates the need for new interaction mechanisms that bundle the control of multiple ubiquitous devices. We support this need through a platform that extends interaction with IoT devices. *ActPad* accounts for different ways of controlling IoT devices by enabling various modes of interaction – in particular simultaneous, sequential, implicit and explicit – and, hence, a rich input space. As a proof of concept, we illustrate several use cases, including, but not limited to, controlling the browser on a PC, turning lights on/off, switching songs, or preparing coffee.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile devices**; *Interaction techniques*; Interface design prototyping.

KEYWORDS

Smart Desk, IoT, Capacitive Sensing, Interaction Platform

ACM Reference Format:
Sarah Delgado Rodriguez, Sarah Prange, Lukas Mecke, and Florian Alt. 2021. *ActPad*– A Smart Desk Platform to Enable User Interaction with IoT Devices. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '21 Extended Abstracts), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3411763.3451825

CHI '21 Extended Abstracts, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8095-9/21/05…$15.00
https://doi.org/10.1145/3411763.3451825

1 INTRODUCTION

In the era of ubiquitous computing, interaction is no longer a one user to one device relationship, but one user may be in control of many Internet of Things (IoT) devices. This creates several challenges, which become apparent as we take a closer look at one specific setting, that is desks. A desk setup typically allows many (IoT) devices to be controlled, such as a computer, lights, speakers, and potentially a telephone. Beyond the desk, further devices come into play such as printers, projectors, or a coffee machine. Each device comes with its own interaction modalities and metaphors.

As of now, interaction with these devices is mostly handled separately. Smart home hubs and applications are first approaches towards centralizing interaction within the home and controlling IoT devices for purposes such as automation and energy savings. However, they require extensive additional hardware and software to provide interaction techniques and computational power – means that are usually available at desks by default.

In this paper, we describe *ActPad*, a touch sensing desk pad, which opens novel opportunities for interaction in smart environments. *ActPad*'s interaction space is customizable. This not only enables configurations for particular needs, but it also enables researchers/designers to prototype new means for smart environment interaction. Furthermore, our platform enables interactions using physical metaphors. For example *ActPad* would allow touching an eraser to delete a document or touching the left/right speaker to switch back and forth between songs (cf. Figure 1–e).

Our work focuses on the implementation of *ActPad* and the exploration of different use cases. Our research approach is as follows: First, we describe the overall concept and the modes of inputs – in particular simultaneous, sequential, implicit and explicit – it supports. Second, we present the design decisions that led to the final implementation of *ActPad*. Finally, we present a number of use cases as proof of concept, including interactions within and beyond the desk, such as controlling applications on a PC, turning lights on and off, switching songs, or preparing coffee.

2 BACKGROUND AND RELATED WORK

We draw from several strands of related work. On one hand, capacitive sensing and toolkits provide the foundation on which we built our prototype. On the other hand, interaction in smart spaces in general and interactive tabletops in particular influenced our work.

2.1 Capacitive Touch Sensing

Capacitive sensors constantly measure the capacitance of connected electrodes, which can be influenced by a close-by human body. This is used to detect touches or hovering gestures [8]. This technology has often been applied to research prototypes, as it is easy to integrate [3, 8]. *Capacitive toolkits* enable prototyping touch sensitive objects [10, 16] and allow touch recognition to be added to existing objects [7, 14] and surfaces [2, 6].

For example, Valkyrie et al.'s toolkit *Midas* supports enhancing existing objects with capacitive touch sensors. They provide software to facilitate the design of electrodes and wiring, which is subsequently cut out of a copper foil sheet. The resulting shapes can then be stuck to an existing object or connected to capacitive touch sensors [15].

2.2 Interaction in Smart Spaces

Interaction in smart spaces has gathered considerable attention in HCI research. In such scenarios, the user is facing myriads of devices in a ubiquitous computing environment. Novel interaction modes for smart spaces have been suggested. Examples include, but are not limited to, Beigl's *point & click* approach [1] and Wilson and Shafer's *XWand* [20], both using additional hardware in the form of a remote control to point at and control devices. Related work also compared different user interfaces, suggesting that a PC could act as central control unit, while a mobile device was more preferred for instant control anywhere within an apartment [11]. Kühnel et al. suggested using a mobile phone as interaction device as well as a three-dimensional gesture space. Metaphoric gestures are most intuitive and memorable (e.g., for increasing volume, turning on or off devices) [12]. Another approach for interacting with smart devices is the use of simple trigger-action programming [17].

Tabletop Interaction. As a more specific example within smart spaces, interactive tabletops (interactive surfaces that combine the physical world (i.e., a table) with the digital world) and respective mechanisms have been suggested in prior work. This opens opportunities for novel input modalities, such as the use of gestures and/or tangible objects [9]. An early example for this is Wellner's 'digital desk', using physical metaphors to interact with digital objects such as documents [19]. Also gestures and methods to define gesture sets have been suggested for tabletop interaction [21].

As an example for tabletop interaction that is *portable*, Villar et al. suggested *Zanzibar*, a flexible mat that allows for interaction using tangible objects equipped with NFC tags. The mat can track tagged objects and sense touch or hover gestures, while preserving a portable form factor to be used on arbitrary surfaces [18].

2.3 Summary

To summarize, interaction in smart spaces is still challenging and recent approaches either require additional hardware (e.g., a special form of remote control [20]) or switching to another device that might be out of reach (e.g., the smartphone as a proxy). This creates a need for novel, scalable interaction mechanisms that are easy to implement and customizable. We envision *ActPad* as an interaction device for desk setups, that is easy to customize and integrate into existing setups. Furthermore, we see *ActPad* as a potential prototyping tool for touch interaction in desk setups, contrary to other toolkits that focus on general enhancement of objects [7, 10, 15]. Thus, *ActPad* can bundle smart desk approaches and rapid prototyping of stationary and movable touch sensitive objects.

3 THE ACTPAD PLATFORM

We aim at centralizing interaction with ubiquitous devices in a desk pad. Hence, we built *ActPad*, a platform enabling capacitive touch input on arbitrary objects to control local and remote IoT devices.

3.1 Requirements

The presented prior literature and conceptual considerations led to a set of conceptual and technical requirements, which we briefly summarize in the following:

Support Different Interaction Commands *ActPad* should enable interaction with a large number of IoT devices and their respective control options. The required interaction commands should be customizable and adapt to different setups, user behaviors, and preferences.

Enable Natural Integration into the Desk Environment It should be easy to integrate *ActPad* in a desk environment, specifically without requiring much space or influencing the regular desk setup. Touch interactions with existing objects (e.g., lamp or speakers) should be detectable.

Customizable Object-to-Action Mapping Due to very different habits of users, another requirement is flexibility with regards to which objects should be mapped to which actions. Hence, we envision using technology that not only allows objects to be easily connected but also the setup to be freely changed at any time.

3.2 Modes of Input

To support a wide range of user commands, *ActPad* enables unlimited input lengths, sequences that include repeated inputs and both, simultaneous and sequential touch entries. This results in a theoretically unlimited input space.

3.2.1 Sequential vs Simultaneous Input. Each input can include one or multiple touches that may be performed sequentially or simultaneously. Generating multiple touch entries *simultaneously* is recognized as a single input. This is similar to shortcuts known from keyboards such as, e.g. ctrl + C for copy on Windows PCs. Such inputs could be used for simple commands (e.g. turning on lights or switching songs) with a reduced risk of being performed unintentionally. In addition, the length of *sequential* input sequences can be used to reflect different user considerations like frequency of use (frequently used actions could be triggered through 'easier'/shorter combinations) or security requirements (e.g. access to sensitive data could require more complex combinations.).

3.2.2 Explicit vs Implicit Input. As illustrated in the previous section, *ActPad* supports explicit input, i.e. users deliberately hit their custom sequence to trigger a desired action. At the same time, *ActPad* can also support implicit input. For example, a user could be logged into the system in the morning based on their routines (e.g., the order in which they touch different objects on their desk). Also more sophisticated features, such as the duration for which users touch an object or the time between touching two objects could serve as input (cf. behavioral biometrics systems). While we do not explore this further in this work, possible applications are security as well as adaptive user interfaces.

3.3 Design Decisions and Implementation

With *ActPad*, we aim at providing customizable and tangible interaction. Our platform should be easy to integrate in existing desk setups and allow for custom interaction commands to be defined. In this section, we explain our design decisions and implementation.

3.3.1 Capacitive Desk Pad. To enable touch interactivity of *ActPad*, we used a one-layered self-capacitive sensing approach [8, 16]. By using a desk pad, we avoid modifying existing furniture and achieve mobility of our system. We further designed *ActPad* thin and with a smooth surface in order not to interfere with regular user interaction. The resulting desk pad is 90 cm wide, 60 cm deep and 5 mm high and consists of two layers of flexible polyvinyl chloride (pvc). It has 40 electrodes on its top surface, all of which are individually wired to a central circuit board (cf. Figure 2a).

3.3.2 Connectors for Object-To-Sensor Mapping. To facilitate flexible user-created object-to-sensor mappings, we designed reusable connectors that can provide a wired connection between any electrode and desired object. They consist of a 3D-printed, flexible casing, reusable double-sided tape and a copper foil electrode connected to a wire (Figure 2b). The connectors can be attached to conductive surfaces and can be rearranged at any time.

3.3.3 Controlling Unit: Sensing and Computation. We used a Raspberry Pi 4[1], as it is a standalone computer which can easily produce a graphical user interface. It also provides GPIO (general-purpose input/output) pins that enable the parallel connection of multiple capacitive sensors (MPR121[2]). We added a 3D-printed case and a fan to protect the setup and provide the necessary ventilation. In summary, our controlling unit bundles all required sensing and computation capabilities. It can easily be integrated with existing setups and is mobile due to its reduced size and the provided casing.

3.3.4 Desk Setup. We deployed *ActPad* in a desk setting, connecting several objects usually found on desks (cf. Figure 1, center). Possible items include stationary (e.g., PC, monitor) and movable objects (e.g., cup). Instead of using a connector we added a conductive surface to the cup's bottom and sides so that it can be used to transfer touches to the desk pad. It thus provides an additional modality of sequential input, as the object can be grabbed and moved from one electrode to another. Note that this is only an example setup demonstrating that *ActPad* can be easily used on most regular desks.

3.4 Customization of Interaction Commands

To explore user interaction capabilities with *ActPad*, we implemented a Python-based graphical user interface using the tkinter gui-toolkit[3]. This *exploration interface* includes a visual representation of each touch point and its current state (touched or not). It also visualizes touch interaction with the objects connected to *ActPad* using a previously defined object-to-sensor mapping. Furthermore, the current user input, sequential or simultaneous, can be recorded (Figure 3). The GUI allows to subsequently select an action, which should be executed when the recorded input has been detected and save these settings.

4 PROOF OF CONCEPT

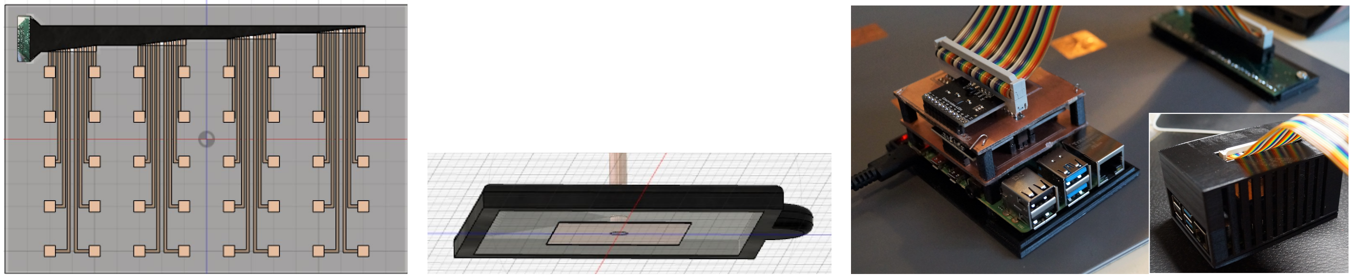
We implemented the following examples to showcase application opportunities of *ActPad*. They demonstrate the wide range of possible interactions with IoT devices our platform supports, due to the use of different interaction types (cf. Section 3.2) and the large number of possible interaction sequences.

---

[1] https://www.raspberrypi.org/products/raspberry-pi-4-model-b, last accessed December 18, 2020

[2] https://learn.adafruit.com/adafruit-mpr121-12-key-capacitive-touch-sensor-breakout-tutorial, last accessed December 18, 2020

[3] https://docs.python.org/3/library/tkinter.html, last accessed December 18, 2020

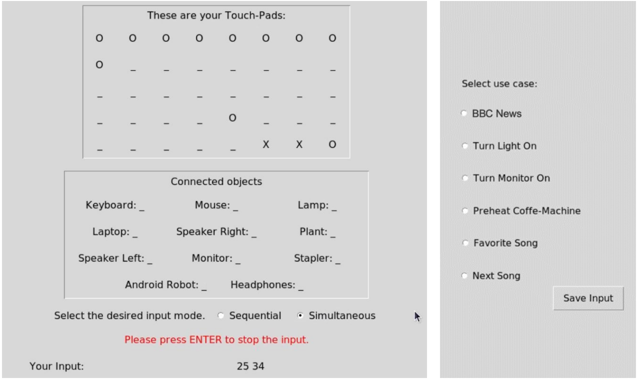


(a) CAD design of *ActPad* with top layer set invisible, to show the inner wiring.

(b) Our connectors consist of a 3D printed casing, a reusable double-sided tape, copper foil and a wire.

(c) *ActPad*'s controlling unit contains one Raspberry Pi 4 Model B and four MPR121 capacitive touch shields.

Figure 2: Details on the construction of our *ActPad* prototype: (a) desk pad, (b) connectors and (c) central computing unit.



These are your Touch-Pads:

```
O  O  O  O  O  O  O  O
O  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _
_  _  _  _  O  _  _  _
_  _  _  _  _  X  X  O
```

Connected objects

Keyboard: _ Mouse: _ Lamp: _
Laptop: _ Speaker Right: _ Plant: _
Speaker Left: _ Monitor: _ Stapler: _
Android Robot: _ Headphones: _

Select the desired input mode. ○ Sequential ● Simultaneous

Please press ENTER to stop the input.

Your Input: 25 34

Select use case:
○ BBC News
○ Turn Light On
○ Turn Monitor On
○ Preheat Coffe-Machine
○ Favorite Song
○ Next Song

Figure 3: The *exploration interface* visualizes the current state of all touch points and connected objects (O = connected to an object, _= not touched, X = touched). Users' sequential or simultaneous input can be recorded and assigned to an action, e.g. controlling the light, turning on the monitor or the coffee-machine, open a webpage, play a song or switch to the next song.

4.1 Local Applications

We first implemented interactions with applications on the Raspberry Pi itself, such as an Internet browser or the music player cmus[4]. Corresponding python and terminal commands were bound to inputs using the previously described UI. As an example for metaphoric interactions, after touching the left speaker, cmus started and played a predefined song. To switch to the next song, the right speaker could be touched. We also tested simultaneous touch inputs, serving as shortcut to show BBC's web page in a browser.

4.2 Remote Appliances

Furthermore, we connected *ActPad* to remote appliances. We established the communication with a smart power socket switch (Delock 11826[5]) via WLAN. This power socket switch comes with the open-source firmware Tasmota[6] and can be controlled by HTTP GET requests. This way we enabled (a) controlling the lamp of our desk setup, (b) turning on and preheating a coffee machine, which was situated in another room, and (c) powering up the monitor. This could be done by implicitly or explicitly entering the configured sequential or simultaneous input.

To give some examples for utilized interaction sequences: we activated the coffee machine by moving the cup from one touch point to another, controlled the lamp by explicitly entering a specific touch sequence and powered the monitor on through the implicit movement of the mouse after sitting down in front of the desk.

5 DISCUSSION AND FUTURE WORK

In this section, we discuss the limitations of our prototype implementation, as well as opportunities *ActPad* provides for future research on user experience, multi-user scenarios, for safety, security and privacy, potential output and feedback and further use cases in desk setups and beyond.

5.1 Limitations of Our Implementation

With *ActPad*, we introduce a platform for quick and easy control of smart home appliances. However, our prototype is still reliant on physical connections of all objects that should be used for input. Hence, *ActPad* in its actual state relies on cables and connectors, which might impact usability. This can be sub-optimal, in particular for movable objects. Nevertheless, we trust in the vision that future smart devices can recognize being touched by default, which eliminates the need for our workaround with copper foil and cables.

Another limitation of our work is that it is – as of now – a standalone platform without tight integration into existing commercial smart home solutions, other than the open source protocol Tasmota. However, our system's controlling unit includes most commonly used communication technologies, such as Bluetooth and WiFi. Hence it provides all necessary functionalities to interact with arbitrary IoT devices, as long as these provide corresponding interfaces.

---

[4] https://cmus.github.io/, last accessed December 18, 2020

[5] https://www.delock.de/produkte/G_11826/merkmale.html?setLanguage=en, last accessed December 18, 2020

[6] https://tasmota.github.io/docs/#/, last accessed December 18, 2020

5.2 User Experience and Usability

User experience and usability of *ActPad* are an important aspect for future work. Interesting research questions are, e.g.: How does *ActPad* perform compared to traditional IoT controlling systems, such as keyboards, smartphone apps or remote controls? How could the system distinguish between unintentional touches and intentional inputs? And how could *ActPad* support implicit interaction? The latter clearly reduces interaction effort, but might be affected by unintentional inputs. Hence, we see particular potential for intelligent decision making, based on environmental information (e.g. did the user just arrive at the desk?).

5.3 Multi-User Scenarios

Open questions evolve around the use of *ActPad* in multi-user environments. In particular, if users are sharing devices or services it might be necessary to implement suitable control features. Prior work has shown that multi-user households arrange with the shared use of smart devices by transferring existing roles within the household, indicating that the initial owner of the device has power over it [5]. However, in case the interaction is decoupled from the actual device, what happens if multiple users want to control the light in parallel? In such cases, mechanisms are required that manage control, by, e.g., assigning roles to multiple instances of *ActPad*.

5.4 Security, Safety & Privacy of ActPad

ActPad enables novel approaches to implement *security* mechanisms. For example, means for implicitly authenticating at the desktop can be embedded in the users’ routine as they arrive at their workplace in the morning. The system could, for example, treat the order in which users turn on the monitor, put the laptop in the docking station and position the keyboard and how they do it as a password that would subsequently log users into the system. However, our current implementation relies on physical connections to enhanced objects, affecting its resistance against specific attacks (e.g. shoulder-surfing [4]). Future work should evaluate *ActPad*’s overall resistance against these types of attacks. As for now, this limitation could be mitigated by connecting more objects than necessary. Nevertheless, we envision a future application of our concept through ubiquitous, touch-sensitive IoT devices, making cables and copper-tape unnecessary.

At the same time, enabling remote input (e.g., to turn on the stove or coffee machine) creates a need for new *safety* mechanisms. The ability to remotely turn on devices may circumvent existing safety mechanisms: For example, the buttons to turn on a stove may be designed in a way such that children cannot turn them on (as a result of high resistance). Furthermore, turning on the stove while not being co-located to it might pose a risk as well. In this case, providing users the ability to verify if a device is (still) on or to explicitly notify users about this might be useful.

Finally, the ability to control devices remotely has potential to protect users’ *privacy* as it is more difficult to observe their actions while they are at their desk compared to shared spaces.

5.5 Output and Feedback

An important principle in HCI is to provide immediate feedback [13]. It is well-known that this creates a challenge in environments where input and output modalities are decoupled. While at a desktop computer (input: mouse, keyboard; output: display) user interfaces can leverage the fact that users mostly direct their attention towards the screen, this becomes difficult if the device/service being controlled is not visible from the users’ position. Here, users might want to receive feedback on (a) whether the command was triggered successfully and (b) when the device/service is ready to be used. In both cases, it is yet unclear how to best provide feedback and through which modality. Also, the user’s current context and task need to be considered. One approach may be to enhance future versions of *ActPad* with a feedback channel that would, for example, allow a vibration motor to be triggered or an LED to light up. For desk setups, another approach is to integrate feedback on the PC, e.g., in the task bar.

5.6 Further Application Areas

We demonstrated how *ActPad* can be a useful means considering a desk setup. At the same time, we see further use cases beyond desk scenarios that could benefit from our setup. One example is the *kitchen*. Tasks secondary to the main cooking task, such as looking up a recipe, pre-heating the oven, turning on the radio, or receiving a phone call are frequent, but often difficult to achieve due to dirty hands or the need to stay in close proximity to, e.g., the stove. With our approach, interactive controls could be integrated with the kitchen worktop in close proximity to the user. Generally, we believe that use cases where users are in (physically) *fixed settings* could benefit from our approach – in particular, when it comes to controlling remote objects or services. Hence, we plan to explore more application areas through user studies with our prototype.

5.7 ActPad as a Prototyping Tool

We also see *ActPad* as a rapid prototyping tool for research on novel touch interaction devices in desk setups. *ActPad* allows for easy and fast enhancement of stationary and movable objects and also enables capacitive touch sensing on the tabletop, which are both useful features for the exploration of new interaction interfaces. In this regard, we see *ActPad* as a tool for early design and evaluation phases in the development process of such interfaces. The application of *ActPad* as a prototyping tool facilitates “by design” integration of interaction capabilities into novel devices. Hence, *ActPad* bundles opportunities for rapid prototyping of touch sensing objects [7, 15] and smart tabletop interactions [18].

6 CONCLUSION

In this paper, we presented the rich design opportunities of *ActPad*. We described in detail conceptual as well as technical requirements for the platform and implementation, and a first prototype. Furthermore, we provided several examples for applications that we implemented with *ActPad*. We envision the platform to serve as a basis for further applications and extensions to shape interaction within smart homes.

PROJECT RESOURCES

To enable researchers and practitioners, to rebuild *ActPad*, we provide the following material[7]: CAD designs of the desk pad and 3D printed parts, templates for circuits, listings of used materials and build instructions.

ACKNOWLEDGMENTS

This work has been supported in part by the DFG under grant agreement no. 425869382 as well as by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr (Voice of Wisdom).

REFERENCES

[1] Michael Beigl. 1999. Point & Click-Interaction in Smart Environments. In *Handheld and Ubiquitous Computing*, Hans-W. Gellersen (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 311–313.

[2] Paul Dietz and Darren Leigh. 2001. DiamondTouch: A Multi-User Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) *(UIST '01)*. Association for Computing Machinery, New York, NY, USA, 219–226. https://doi.org/10.1145/502348.502389

[3] David Dobbelstein, Philipp Hock, and Enrico Rukzio. 2015. Belt: An Unobtrusive Touch Input Device for Head-Worn Displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2135–2138. https://doi.org/10.1145/2702123.2702450

[4] Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. 2017. Understanding Shoulder Surfing in the Wild: Stories from Users and Observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4254–4265. https://doi.org/10.1145/3025453.3025636

[5] Christine Geeng and Franziska Roesner. 2019. Who's In Control? Interactions In Multi-User Smart Homes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 268, 13 pages. https://doi.org/10.1145/3290605.3300498

[6] Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. 2014. PrintSense: A Versatile Sensing Technique to Support Multimodal Flexible Surface Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1407–1410. https://doi.org/10.1145/2556288.2557173

[7] T. Grosse-Puppendahl, Y. Berghoefer, A. Braun, R. Wimmer, and A. Kuijper. 2013. OpenCapSense: A rapid prototyping toolkit for pervasive interaction using capacitive sensing. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, New York, NY, USA, 152–159. https://doi.org/10.1109/PerCom.2013.6526726

[8] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017. Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3293–3315. https://doi.org/10.1145/3025453.3025808

[9] Otmar Hilliges, Andreas Butz, Shahram Izadi, and Andrew D. Wilson. 2010. *Interaction on the Tabletop: Bringing the Physical to the Digital.* Springer London, London, 189–221. https://doi.org/10.1007/978-1-84996-113-4_9

[10] Scott E. Hudson and Jennifer Mankoff. 2006. Rapid Construction of Functioning Physical Interfaces from Cardboard, Thumbtacks, Tin Foil and Masking Tape. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 289–298. https://doi.org/10.1145/1166253.1166299

[11] Tiiu Koskela and Kaisa Väänänen-Vainio-Mattila. 2004. Evolution towards Smart Home Environments: Empirical Evaluation of Three User Interfaces. *Personal Ubiquitous Comput.* 8, 3–4 (July 2004), 234–240. https://doi.org/10.1007/s00779-004-0283-x

[12] Christine Kühnel, Tilo Westermann, Fabian Hemmert, Sven Kratz, Alexander Müller, and Sebastian Möller. 2011. I'm home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human-Computer Studies* 69, 11 (2011), 693 – 704. https://doi.org/10.1016/j.ijhcs.2011.04.005

[13] Jakob Nielsen. 1995. 10 usability heuristics for user interface design. https://www.nngroup.com/articles/ten-usability-heuristics/

[14] Narjes Pourjafarian, Anusha Withana, Joseph A. Paradiso, and Jürgen Steimle. 2019. Multi-Touch Kit: A Do-It-Yourself Technique for Capacitive Multi-Touch Sensing Using a Commodity Microcontroller. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1071–1083. https://doi.org/10.1145/3332165.3347895

[15] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: Fabricating Custom Capacitive Touch Sensors to Prototype Interactive Objects. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 579–588. https://doi.org/10.1145/2380116.2380189

[16] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 253–258. https://doi.org/10.1145/2807442.2807503

[17] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical Trigger-Action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 803–812. https://doi.org/10.1145/2556288.2557420

[18] Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, William Field, and Haiyan Zhang. 2018. Project Zanzibar: A Portable and Flexible Tangible Interaction Platform. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 515, 13 pages. https://doi.org/10.1145/3173574.3174089

[19] Pierre Wellner. 1991. The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology* (Hilton Head, South Carolina, USA) *(UIST '91)*. Association for Computing Machinery, New York, NY, USA, 27–33. https://doi.org/10.1145/120782.120785

[20] Andrew Wilson and Steven Shafer. 2003. XWand: UI for Intelligent Spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 545–552. https://doi.org/10.1145/642611.642706

[21] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-Defined Gestures for Surface Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1083–1092. https://doi.org/10.1145/1518701.1518866

---

[7] www.unibw.de/usable-security-and-privacy-en/research/projects/actpad