
PaperPulse: An Integrated Approach to Fabricating Interactive Paper

Raf Ramakers **Kashyap Todi** **Kris Luyten**
Hasselt University - tUL - iMinds
Expertise Centre for Digital Media
Diepenbeek, Belgium
firstname.lastname@uhasselt.be

*Note: This extended abstract is for CHI 2015 'Interactivity'.
Please refer to the proceedings of CHI 2015 for the full paper.*



Figure 1: Artifacts made using *PaperPulse*: (a) An interactive game (b) A birthday invitation card with a code-slot to reveal the date (c) A diet-card to keep track of daily eating habits.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).

CHI'15 Extended Abstracts, Apr 18-23, 2015, Seoul, Republic of Korea
ACM 978-1-4503-3146-3/15/04.

<http://dx.doi.org/10.1145/2702613.2725430>

Abstract

We present *PaperPulse*, a design and fabrication approach that enables designers to produce standalone interactive paper artifacts by augmenting them with electronics. With *PaperPulse*, users overlay visual designs with widgets provided in the design tool. *PaperPulse* provides three families of widgets, designed for smooth integration with paper, for a total of 20 different interactive components. We also contribute a demonstration and recording approach, *Pulsation*, that allows specifying interaction logic. Using the final design and the recorded *Pulsation* logic, *PaperPulse* generates layered electronic circuit designs, and code that can be deployed on a microcontroller. By following automatically generated assembly instructions, designers can seamlessly integrate the microcontroller and widgets in the final paper artifact.

In our presentation, we will demonstrate the process of fabricating interactive paper using *PaperPulse*, and techniques for constructing paper widgets.

Author Keywords

Paper electronics; Paper-crafts; Fabrication; Design Tools; PBD; Tangible Interfaces.

ACM Classification Keywords

H.5.2: User Interfaces.

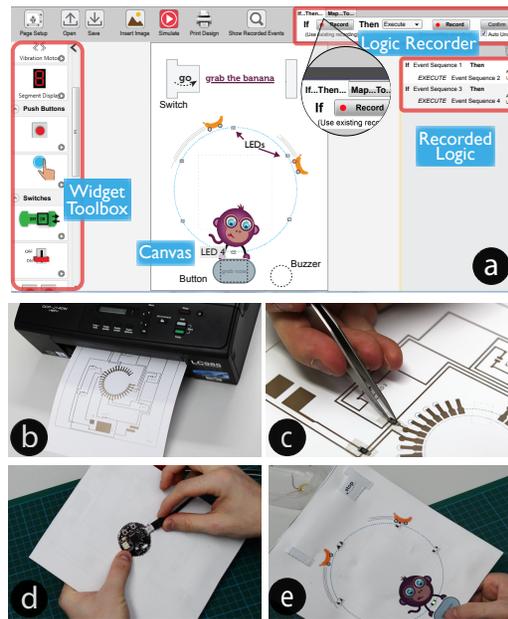


Figure 2: The *PaperPulse* workflow. (a) Design and specify logic; (b) print sheets; (c) assemble; (d) upload generated program to microcontroller; (e) final paper artifact.

Introduction

There has been a growing interest in different fields and communities (e.g. HCI research, maker movement, engineering and marketing) in making paper interactive by augmenting it with electronics. It has become possible to produce low-cost paper versions of PCBs in lab environments [1], and bring liveness to paper artifacts such as books [2] and posters [4]. Although advancements in fabrication tools for electronic circuits, such as conductive pens, threads, inkjet printers [1] and vinyl cutters [3], make building paper circuits accessible to

people, the requirement of possessing elementary knowledge of electronics and programming can make creating interactive paper circuits a challenging task.

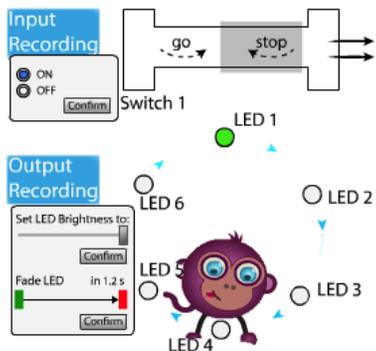
To enable designers to augment paper designs with electronics, we present *PaperPulse*, a design tool that assists users and automates parts of the design, programming, and fabrication process for electronic paper circuits. With *PaperPulse*, users make stand-alone interactive paper artifacts, in which electronic components are seamlessly integrated in visual designs, without requiring skills in electronics or programming.

Thus, the main **contributions** of this work are:

- (1) Three families of interactive widgets, suitable for paper. Each family consists of standard controls, such as push buttons, switches, sliders and radio buttons.
- (2) *Pulsation*, a demonstration technique to enable non-programmers to specify logic between basic electronic sensors.
- (3) A design tool to overlay widgets onto visual designs, specify logic using *Pulsation*, and simulate this before deployment. The tool also simplifies the fabrication process by automatically generating circuits, microcontroller code, and instructions to help assemble final paper artifacts.

Walkthrough: The Hungry Monkey Game

The following walkthrough illustrates the process of designing and fabricating a paper game with *PaperPulse* (Figure 2). The game consists of a loop of six LEDs that consecutively turn on and off. The objective of the game is to “grab the banana” by pressing a button at the moment when a particular LED lights up. A buzzer rings for a short duration each time the player succeeds.



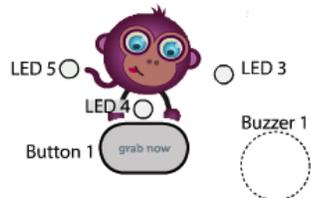
IF Switch 1 ON

THEN

Action	Time (s)
Brightness LED 1 to 100	0.3
Brightness LED 1 to 0	0.6
Brightness LED 6 to 100	3.3
Brightness LED 6 to 0	3.6

Number of Loops: **Infinite**

Figure 3: Recording an *if-then* rule for the LED sequence when a switch is turned on. (Animated figures best viewed with Acrobat Reader.)



IF Brightness LED 4 equals 100

Button 1 Press

THEN

Action	Time (s)
Volume Buzzer 1 to 100	0.0
Volume Buzzer 1 to 0	2.0

Random Simultaneous Sequential

Figure 4: Recording another rule to ring the buzzer if the button is pressed at the moment “LED 4” turns on.

Step 1: Designing the Interactive Paper Layout

The designer starts by specifying the dimensions of the paper design, importing pre-designed images, and arranging them onto the canvas (Figure 2). Next, the designer overlays this design with interactive components (six LEDs, a buzzer, a pull switch, and a button), available in the widget toolbox (Figure 2).

Step 2: Defining and Verifying Logic Iteratively

The designer first links the “stop-go” switch to the circle of the LEDs. When the switch is turned on, the LEDs blink in a looping pattern. Figure 3 illustrates how recordings are created by demonstrating actions specific to widgets, and the specified logic for the desired behavior.

To verify the recorded rule, the designer uses the *simulator* to interact with the widgets and observes the corresponding output.

Next, the designer records the logic for the “grab now” button. If pressed at the moment when the LED under the monkey (“LED 4”) lights up, the buzzer should ring to indicate that the game is completed. Figure 4 illustrates the specified logic to obtain this behavior.

Step 3: Printing and Assembly

The printing process starts by generating: (1) An electronic circuit that connects widgets to pins on the Threadneedle microcontroller¹. PaperPulse limits the number of intersecting circuit traces. (2) PDF files consisting of the electronic circuits, widget-specific assembly lines (e.g. cut lines, fold lines), and visual designs. (3) Microcontroller code. (4) A customized tutorial to guide the designer through the printing, deployment, and assembly process.

¹modlab.co.uk

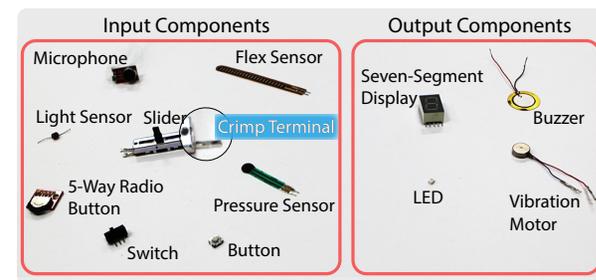
Following the tutorial, the designer prints the PDF files, mounts the electronic components, assembles the widgets, and uploads the code to the microcontroller. As shown in Figure 2e, the resulting end-product, powered by a battery, can now be used as a standalone paper game.

PaperPulse Widgets

We present three families of standard widgets to realize basic controls such as push buttons, switches, sliders, and radio buttons. These widgets have been selected and designed in such a way that they smoothly integrate and blend with paper designs.

Off-the-Shelf Widgets

PaperPulse currently supports eight off-the-shelf input sensors and four output actuators. Flat components are attached using ECATT-tape²; through-hole components are extended with female crimp terminals and attached using conductive paint.



Although off-the-shelf widgets require little manual assembly, they have fixed designs and often protrude from the surface. In contrast, *Paper-membrane* and *pull-chain* widgets are resizable and integrate seamlessly with visual elements on paper.

²Electrically Conductive Adhesive Transfer Tape

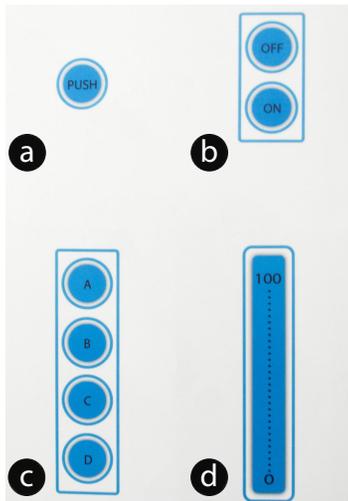


Figure 5: Paper-Membrane Widgets: Push button; switch; radio buttons; and slider.

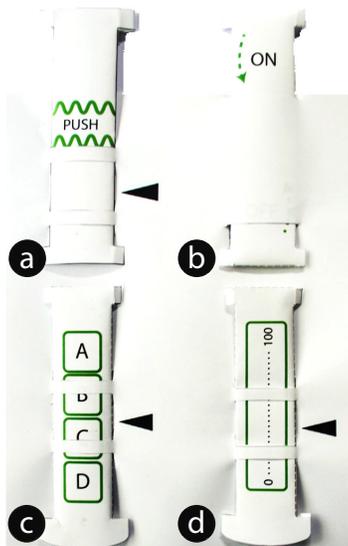


Figure 6: Pull-Chain Widgets.

Paper-Membrane Widgets

Figure 5 illustrates the different paper-membrane widgets. A circuit is created between the base layer and the top layer; these two layers are separated by a thin air gap, using a paper-frame (membrane). Pressing on the top layer connects it to the bottom, thus closing the circuit and generating an input signal.

In contrast to off-the-shelf widgets, paper-membrane widgets are customizable. On the other hand, they do not offer tangibility. This is the essence of pull-chain widgets.

Pull-Chain Widgets

These widgets use a pull-chain mechanism to provide tangibility. Figure 6 shows a push button (using a crossing interaction technique), a pull-chain switch, radio buttons and a slider.

The mechanical design of every pull-chain widget consist of (a) a folded tube structure (widget-specific layer), with a hollow center to provide strength and rigidity during pulling and pushing motions, (b) slots to guide the pull-strip, (c) a wing tab to lock the pull-strip in place and (d) a pull-tab that functions as handle.

Pulsation: Specifying Sensor Logic By Demonstration

Pulsation allows users to specify logic by demonstrating and recording actions directly in the context of the visual design elements. To provide a higher ceiling than is possible with demonstration alone, *Pulsation* augments widgets and the demonstrated actions with dialogs that allow fine-tuning specific properties.

Logic is defined by specifying relationships between *input sets* (conditions that have to be fulfilled) and *output sets* (consist of one or more output actions). These

relationships can be specified using either *if-then* or *map-to* rules. For *if-then* rules, the output set is executed when all conditions in the input set have been satisfied. The walkthrough (Figure 3, 4) shows two examples of such rules. For *map-to* rules, parameters of *input set* (e.g. raw value, repetitions, progress within the set) are continuously mapped to parameters of the *output set* (e.g. value, progress of actions, speed with which the actions are executed). Example usages of *map-to* rules are mapping the value of a slider to the brightness (value) of LEDs, or mapping the number of times (repetitions) a button is pressed to the number of LEDs (progress) that are lit up.

Conclusion

Using these rules, and the aforementioned widgets, users can interactively realise a broad range of applications, such as shown in Figure 1. As illustrated in the walkthrough, *PaperPulse* streamlines the entire design and fabrication process, assisting and supporting designers during each step of creating interactive paper.

References

- [1] Kawahara, Y., Hodges, S., Cook, B. S., Zhang, C., and Abowd, G. D. Instant inkjet circuits: Lab-based inkjet printing to support rapid prototyping of ubicomp devices. In *Proc. UbiComp '13*,
- [2] Qi, J., and Buechley, L. Electronic popables: Exploring paper-based computing through an interactive pop-up book. In *Proc. TEI '10*,
- [3] Savage, V., Zhang, X., and Hartmann, B. Midas: Fabricating custom capacitive touch sensors to prototype interactive objects. In *Proc. UIST '12*,
- [4] Shorter, M., Rogers, J., and McGhee, J. Enhancing everyday paper interactions with paper circuits. In *Proc. DIS '14*,