

---

# Making 3D Printed Objects Interactive Using Wireless Accelerometers

**Jonathan Hook**

Culture Lab  
School of Computing Science  
Newcastle University, UK  
jonathan.hook@ncl.ac.uk

**Thomas Nappey**

Culture Lab  
School of Computing Science  
Newcastle University, UK  
thomas.nappey@ncl.ac.uk

**Steve Hodges**

Microsoft Research  
Cambridge, UK  
shodges@microsoft.com

**Peter Wright**

Culture Lab  
School of Computing Science  
Newcastle University, UK  
p.c.wright@ncl.ac.uk

**Patrick Olivier**

Culture Lab  
School of Computing Science  
Newcastle University, UK  
patrick.olivier@ncl.ac.uk

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- **ACM copyright:** ACM holds the copyright on the work. This is the historical approach.
- **License:** The author(s) retain copyright, but ACM receives an exclusive publication license.
- **Open Access:** The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Verdana 7 point font. Please do not change the size of this text box.

**Abstract**

We present an approach that allows designers and others to quickly and easily make 3D printed objects interactive, without the need for hardware or software expertise and with little modification to an object's physical design. With our approach, a designer simply attaches or embeds small three-axis wireless accelerometer modules into the moving parts of a 3D printed object. A simple graphical user interface is then used to configure the system to interpret the movements of these accelerometers as if they were common physical controls such as buttons or dials. The designer can then associate events generated by these controls with a range of interactive behavior, including web browser and media player control.

**Author Keywords**

3D printing; accelerometers; physical computing; rapid prototyping; solder-less electronics; tangible interfaces.

**ACM Classification Keywords**

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

**Introduction**

3D printers have revolutionized the way that physical objects are designed and developed. They allow designers to quickly and easily create detailed and

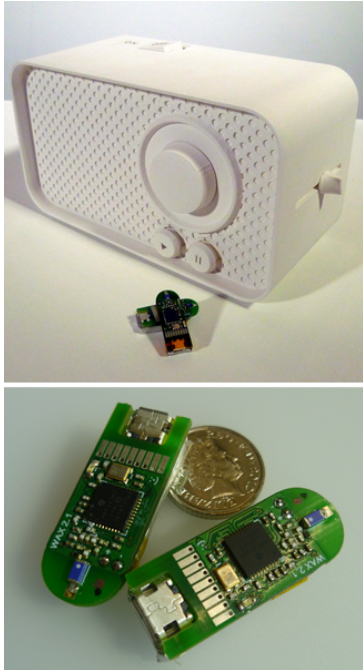


Figure 1: Interactive prototype of a radio (top) and two wireless accelerometer modules (bottom).

accurate physical prototypes, often simply by selecting 'print' in computer aided design (CAD) software. As a result, the need for designers to have significant craft skills or to rely on prolonged industrial fabrication processes to create physical prototypes of their designs has been eliminated in many situations.

3D printed prototypes can be used to experiment with form as part of an iterative design cycle, to present nascent design ideas to colleagues and clients, and to gain initial feedback about a design from potential end-users. Moreover, objects with simple moving parts can be developed by combining components in a single assembly. However, such prototypes only support exploration and evaluation of the form and mechanical characteristics of a design, and not its envisioned digital interactive behavior.

We present an approach that addresses this problem by allowing designers to quickly and easily make 3D printed objects interactive. Our approach works by tracking the movements of multiple wireless three-axis accelerometers, which are embedded within a 3D printed prototype (Figure 1). A simple graphical user interface (GUI) is used to configure the system to interpret these movements as the behavior of common physical controls like buttons and dials, which can then be mapped onto a range of digital interactive behaviors on a host PC. Our approach obviates the need for soldering, electronics or programming skills and results in minimal modification of an object's physical design. As physical prototyping tools become increasingly accessible, we believe the ability to augment the mechanical nature of a prototype with digital interactivity will not only be useful to designers, but will also appeal to other user communities.

In this work in progress paper, we present a brief discussion of related work, a detailed description of our approach and details of our initial functional implementation. We have not yet formally evaluated our approach, but rather we provide an informal validation of our ideas and our current implementation by way of an example application that we have built, a prototype radio. We conclude with a discussion of the advantages and disadvantages of our technique and our plans for extending this ongoing research.

### Related Work

A number of approaches have the potential to support prototyping interactive 3D printed objects. Simple electronic components such as buttons and potentiometers can be built into 3D objects and interfaced with electronics platforms like Arduino [2] to make them interactive. However, this approach requires a basic knowledge of electronics, the ability to solder and microcontroller programming skills. Several 'plug and play' electronics prototyping platforms [e.g. 6] allow designers to work with electronic components without the need for such knowledge or skills. However, significant changes to a design are typically still required to accommodate the components and associated wiring. Therefore, these platforms are not ideal for the rapid and low-effort design prototyping context explored in this paper. We do, however, recognize that this is an active area of research and new ideas exploring how to embed digital circuitry into physical devices [e.g. 5, 7] are actively being explored.

As an alternative to embedding digital circuitry, a number of previous research projects have explored how computer vision might be used to make 3D printed objects interactive. Display Objects combine a projector

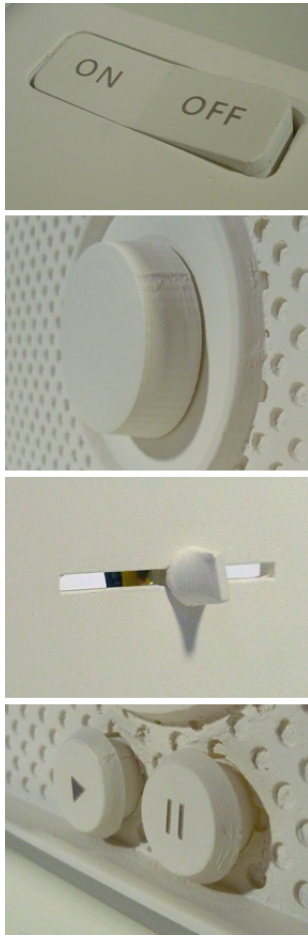


Figure 2: 3D printed physical controls (from top to bottom) rocker switch, push buttons, slider and rotary encoder.

with a Vicon motion tracking system to augment 3D prototypes with GUI elements and touch interaction [1]. This approach allows a wide variety of complex and functional interfaces to be prototyped without the need for any electronics, wiring or modification to a design. However, the tracking and projection technology enforces a number of constraints regarding the interactions supported and it only works in a lab with an expensive motion capture system installed.

Tracking the position and orientation of fiducial markers that are attached to the rear of physical controls is a cheaper alternative [3]. This approach is, however, likely to require significant alterations to the physical form of a prototype to ensure that markers attached to controls are within the line of sight of a camera. Recent work has sought to address the limitations of this camera-based approach by providing CAD software that automates and assists the user in the placement of the controls, camera and a set of mirrors [4]. Additionally, the use of 3D printed light guides can reduce how much a 3D printed object has to be modified to track users' interactions with its moving parts [8]. However, this approach might also be out of reach of designers at the present, due to its reliance on 3D printers that are able to print optically clear material.

### **Our Approach**

The approach presented in this paper builds on this previous work by offering an alternative means to make 3D printed objects interactive. We embed a number of small three-axis wireless accelerometer modules into the moving parts of a 3D printed object in order to track any movement. We associate these movements with the behavior of a number of common physical controls, including: push, rocker and trigger buttons;

levers and pedals; continuous value dials and discrete rotary encoders (Figure 2). The behavior of these controls can then be mapped to a range of different behaviors on a host PC using a simple GUI.

For example, in a prototype radio that we developed to test our approach, an interactive rotary encoder simulates scanning through stations by changing tracks on a media player. To create such a rotary encoder using our approach, a 3D model is developed that includes a rotating dial. A small cavity is made within this dial to house a wireless accelerometer module, which is fitted once the model has been printed. A drop down menu on the GUI is used to specify that the movements of this accelerometer should be interpreted as a rotary encoder. The user then follows instructions on the GUI to *configure* the control, a procedure that involves rotating the dial to a number of set positions. Accelerometer readings taken at these positions are fed into an algorithm that translates movements into clockwise and anti-clockwise rotation events. Finally, drop-down menus in the GUI are used to set the rotary encoder to skip forward and backwards between tracks on a media player when clockwise and anticlockwise rotation events are fired.

Designers can prototype a range of other common physical controls using a similar workflow. For each different type of control, the GUI presents a custom dialog that guides the designer through the configuration phase (Figure 3). Additionally, the GUI is automatically configured to only offer interactive behaviors that can be meaningfully mapped on the events generated by a particular type of control. We have implemented a number of different interactive behaviors, which include generating key and mouse

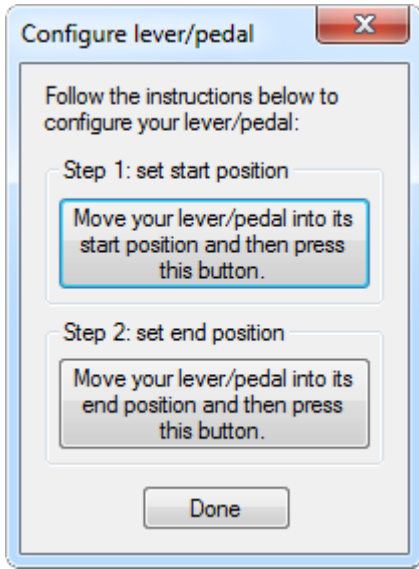


Figure 3: The dialog used to configure lever and pedal type controls.

events, setting the cursor position, and controlling a media player and a web browser.

### Implementation

Our current implementation uses Axivity WAX wireless accelerometers. These rechargeable devices are 34.5mm × 16mm × 15mm in dimension, have a 4mg resolution and an 8-hour continuous use battery life when transmitting data values up to 25m at 50Hz. This data is received and processed by a C# application on a host PC, which includes the GUI that is used to configure how the data is interpreted and mapped to interactive behaviors. The following sections describe how we emulate a range of common physical controls by tracking the moving parts of 3D printed objects using accelerometers.

#### Levers and Pedals

To create a lever or pedal, an accelerometer is embedded within a moving part of a model, such that its orientation with respect to gravity varies on at least one axis as the control is moved. When creating a lever the accelerometer might be positioned to rotate around the pivot point, or put inside the handle that extends from it. To configure the control, the designer moves it to its farthest extent in each direction of travel. The angle between these two positions cannot exceed 180°. Readings from the accelerometer are low-pass filtered and recorded at these positions. The position of the control between these extents is calculated using Equation 1, where  $\hat{a}$  and  $\hat{b}$  are unit vectors of the recorded accelerometer values and  $\hat{c}$  is a unit vector of the current value with a low-pass filter applied.

$$value = \frac{\cos^{-1} \hat{a} \cdot \hat{c}}{\cos^{-1} \hat{a} \cdot \hat{b}} \quad (1)$$

#### Triggers and Rocker Switches

Triggers and rocker switches are created in a similar manner to levers and pedals. An accelerometer is placed within the trigger or rocker switch, such that its orientation with respect to gravity varies when it is pressed and released. The control is configured by recording low-pass filtered values in its pressed and released states. The current position of the control between these extents is calculated using Equation 1. A 'pressed' event is fired when this position exceeds a threshold, which the designer is able to set from within the GUI, and a subsequent 'released' event is fired when it drops back below this threshold. We use a rocker switch as the on/off control in our radio prototype.

#### Dials

Dials can be created by placing an accelerometer within a moving part that rotates around a central point. To configure a dial, the designer initially places it in its 'zero' position, then at 'one-quarter' rotation and finally at 'one-half' rotation. Low-pass filtered accelerometer readings are recorded at each position. The position of the dial is calculated in two stages. Firstly, the position of the accelerometer between the recorded zero and one-half rotation positions is calculated using Equation 1. Then a second step is conducted to resolve the sign of this value and, therefore, determine whether the dial is in the first or second half of its range. The sign is resolved using Equation 2, where  $\mathbf{a}$  is the reading recorded at the zero position,  $\mathbf{c}$  is the current reading with a low-pass filter applied and  $\mathbf{d}$  is the reading at the quarter rotation position.

$$sign = ((\mathbf{a} \times \mathbf{d}) \times \mathbf{c}) \cdot \mathbf{a} \quad (2)$$

### *Rotary Encoders*

Rotary encoders are created in a similar manner to dials. To create a rotary encoder the designer places an accelerometer in a dial-like moving part of a model. The control is configured in an identical manner to a dial. However, rather than returning a continuous value reflecting its position, a rotary encoder fires 'clockwise' and 'anti-clockwise' events when its position changes by an amount that exceeds a configurable threshold, which is set using the GUI. We use a rotary encoder to scan between stations in our radio prototype.

### *Push Buttons*

Push buttons that move along in a linear path when pressed and released can also be created by placing an accelerometer within the part of the button that moves when pressed. To configure a push button, the designer presses and releases the button a number of times. A window of high-pass filtered accelerometer readings around these press and release movements is recorded and principal component analysis is used to determine the unit vector along which the button travels when pressed and released. By projecting high-pass filtered accelerometer values onto this unit vector, we are able to filter out movements that are not closely aligned with the button's direction of travel, such as those caused by moving or reorienting a prototype. When the value returned by this projection is above a configurable threshold, a 'pressed' event is generated. We use push buttons to start and pause playback in our radio prototype.

### **Discussion and Future Work**

The aim of our work is to complement the growing array of accessible physical prototyping tools with an approach that readily supports the integration of digital

sensing and, therefore, interaction with digital content. By simply embedding small wireless accelerometers into the moving parts of a 3D printed object, we can enable non-experts to prototype interactive devices and applications. We believe that our approach offers a number of advantages over previous work. It does not require electronics knowledge, the ability to solder or any programming, and only minor modifications to a design to house the small wireless accelerometer modules are necessary. However, our approach does have a number of limitations, which we discuss here.

Movement of a prototype can cause controls to report unexpected values. In our experiments so far we have found that prototypes are quite robust to being picked up and moved. Therefore, we think that this limitation will not be problematic when prototyping devices that are mainly used in-situ. However, we acknowledge that our current implementation may not be suitable for prototyping devices that will be continually moved during use, like games controllers or mobile phones. To make our approach more robust to movements in the future, we plan to explore whether an additional *ground truth* sensor placed within a prototype could be used to filter out readings caused by movements.

It is not possible to reliably track or infer the position of an accelerometer in space. As a result, our approach cannot be used directly to prototype the linear sliders sometimes used on interactive devices. However, when making our radio prototype, we found that the absolute position of the linear slider used to control volume could be determined by tracking the orientation change of an accelerometer attached to a simple lever mechanism connected to the slider (Figure 4).

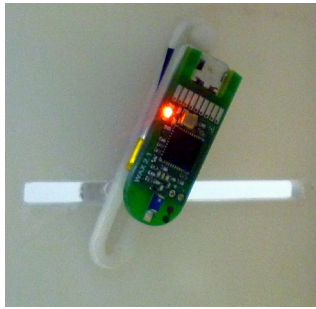


Figure 4: Module attached to a lever to create a linear slider.

All of the controls that we have implemented so far, except for the push button, work by detecting changes in an accelerometer's orientation with respect to gravity. This rules out rotation around an accelerometer's z-axis, as no orientation change is detectable in such cases. However, the use of an inertial measurement unit (IMU) would rectify this limitation. Since small and low-power IMUs are now available we intend to explore this possibility in the future development of our research.

In the future, we plan to extend our approach in a number of ways. We intend to explore whether a wider range of physical controls can be implemented using our approach. For instance, we plan to extend our implementation of levers and pedals into two axes to facilitate the creation of joystick-like controls. In order to help designers learn to use our approach more easily, we plan to develop the GUI to include tutorials that explain the functionality and limitations of each control type. We believe that the inclusion of links to short videos showing examples of 3D printed controls that would and wouldn't work would be particularly helpful for designers using our approach. We would also like to develop a plug-in for CAD software that would guide designers when creating models for use with our approach. This plug-in might include pre-set models for control assemblies, such as the slider in Figure 4.

We are also interested in exploring how small wireless modules embedded in 3D printed objects can be used to provide feedback to the user, in addition to detecting interaction. We have already begun to explore this idea, by adding a remotely controlled vibration motor and LED to an accelerometer module. Using this simple setup we have been able to prototype a range of simple

forms of output, such as giving bursts of vibro-tactile feedback to simulate notches on a rotary encoder.

Ultimately, we believe that it will be critical to evaluate our approach with larger numbers of designers. However, as we undertake this work and continue to develop our ideas, we hope that others working in this exciting field will find the initial research presented in this paper useful for informing their work.

## References

1. Akaoka, E., Ginn, T. & Vertegaal, R. DisplayObjects: prototyping functional physical interfaces on 3D Styrofoam, paper or cardboard models. In *Proc. of TEI '10*, 49-56.
2. Arduino. <http://www.arduino.cc>.
3. Döring, T., Pfleging, B., Kray, C. & Schmidt, A. Design by physical composition for complex tangible user interfaces. In *Ext. Abs. of CHI '10*, 3541-3546.
4. Savage, V., Chang, C. & Hartmann, B. Sauron: Embedded Single-Camera Sensing of Printed Physical User Interfaces. In *Proc. of UIST '13*, 447-456.
5. Savage, V., Zhang, X. & Hartmann, B. Midas: Fabricating custom capacitive touch sensors to prototype interactive objects. In *Proc. of UIST '12*, 579-588.
6. Villar, N. et al. .NET Gadgeteer: A Platform for Custom Devices. In *Proc. of Pervasive '12*, 216-233.
7. Weichel, C., Lau, M., & Gellersen, H. Enclosed: a component-centric interface for designing prototype enclosures. In *Proc. of TEI '13*, ACM, 215-218.
8. Willis, K., Brockmeyer, E., Hudson, S. & Poupyrev, I. Printed optics: 3d printing of embedded optical elements for interactive devices. In *Proc. of UIST '12*, 589-598.