# Using Light Emitting Diode Arrays as Touch-Sensitive Input and Output Devices

*Scott E. Hudson*
Human-Computer Interaction Institute
Carnegie Mellon University, Pittsburgh, PA 15213
E-Mail: scott.hudson@cs.cmu.edu

## ABSTRACT

Light Emitting Diodes (LEDs) offer long life, low cost, efficiency, brightness, and a full range of colors. Because of these properties, they are widely used for simple displays in electronic devices. A previously characterized, but little known property of LEDs allows them to be used as photo sensors. In this paper, we show how this capability can be used to turn unmodified, off the shelf, LED arrays into touch sensitive input devices (while still remaining capable of producing output). The technique is simple and requires little or no extra hardware – in some cases operating with the same micro-controller based circuitry normally used to produce output, requiring only software changes. We will describe a simple hybrid input/output device prototype implemented with this technique, and discuss the design opportunities that this type of device opens up.

### Categories and Subject Descriptors:

H.5.2 [**User Interfaces**]: Input devices and strategies, B.4.2 [**Input/Output Devices**]

### Additional Keywords:

Input Devices, Display Devices, Touch Sensors.

## INTRODUCTION

Light Emitting Diodes (LEDs) have become nearly ubiquitous as simple displays in electronic devices of all sorts because they are inexpensive, bright, highly efficient, long lasting, and now support a full spectrum of output colors. In addition the expected emergence of Organic LED arrays as a new and inexpensive display technology may soon greatly expand the use of LEDs as a display medium. This paper considers how a long known, but little noted property of LEDs can be exploited to allow them to act as input as well as output devices when they appear in properly wired pairs or arrays. This technique is simple, and in the case of microcontroller-based designs, often requires no new parts and minimal changes to circuits (being implemented primarily in software). As a result, this technique has widespread practical applicability, allowing a range of simple devices to be interactively enhanced in a cost-effective manner. In addition, because the LED arrays retain their original ability to display information, these techniques open up new possibilities for previously passive input devices such as buttons to include small dynamic displays. This can allow, for example, a multi-function physical button to visually indicate its current effect in ways that were previously only practical for simulated buttons in graphical interfaces. Figure 1 shows a simple prototype LED array device which can serve as both a display and an input device.

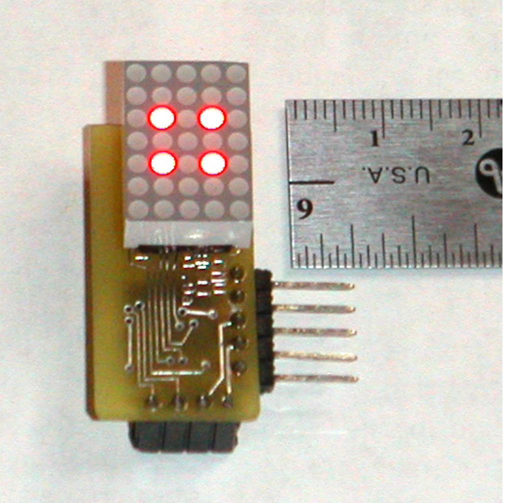
Figure 1. Prototype LED input/output device

In the next section we will consider how LEDs can be used as light sensors. We will then show how this capability can be used to create touch sensors, and describe a prototype implementation, along with some simple interaction techniques.

## USING LEDS AS LIGHT SENSORS

It has long been known that LEDs may be used as light sensors as well as light emitters. This property was recently highlighted in [3] where it was used to create bi-directional communication devices from ordinary and ubiquitous display LEDs.

As illustrated in Figure 2a, in its normal operation, light is emitted when current flows across the junction of an LED from its anode to its cathode (top to bottom in Figure 2). On the other hand, because it is a diode, it does not nominally conduct current in the opposite direction, when it is *reverse biased* by placing a positive charge on the

*UIST '04*, October 24–27, 2004, Santa Fe, New Mexico, USA.

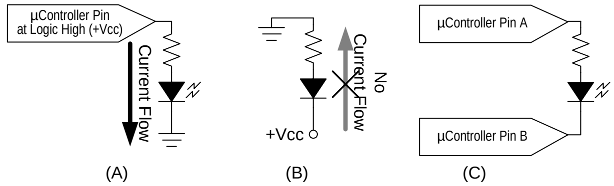

**Figure 2**. a) Normal use of an LED, b) Reverse biased LED, c) Circuit for light sensing.

cathode and a negative charge on the anode (Figure 2b). However, small amounts of current do leak across the diode junction when it is reverse biased. The amount of such leakage is related to the incident light striking the LED, with higher light levels producing substantially larger leakage across the junction.

If noted at all, this property has in the past been considered a minor annoyance. However, with the clever technique described in [3] it is possible to use a microcontroller (which in many cases is already controlling the LED) to exploit this property to measure incident light.

This is done as follows. First, both ends of the LED and current limiting resistor pair are wired to separate I/O pins on the microcontroller (as shown in Figure 2c). This requires no new components, but in this configuration requires the use of one extra pin on the microcontroller which is already present in most electronic devices (as described in the next section, for some array configurations no additional parts, pins, or other changes will be required over the normal display circuitry). To sense light, the microcontroller briefly reverse-biases the LED by setting pin A to logic 0 (ground or 0v) and pin B to logic 1 (typically +5v). This charges the small intrinsic capacitance found in the wire and diode. Pin B is then switched to high impedance input mode. At that point, the input value at the pin will read logic 1. Over a short period of time, the charge on the wire will leak past the reversed biased LED junction to the ground provided by pin A, with the input at pin B subsequently dropping far enough to register as a logic 0. By measuring the time it takes for this charge to drop below the logic 1 level, we can determine the rate of reverse bias leakage, and hence the level of incident light (again with higher light levels causing more leakage, and hence shorter times). This process is illustrated in the code shown in Figure 3.

```
int16 sense_light( )
  {
    // reverse bias LED
    set_mode_pin_a(OUTPUT);
    set_mode_pin_b(OUTPUT);
    output_pin_a(0);   output_pin_b(1);

    // make pin b an input
    set_mode_pin_b(INPUT);

    // count off time until pin b drops
    int16 result_tm = 0;
    while (input_pin_b() && result_tm<LIMIT)
      result_tm++;

    // lower return values mean more light
    return result_tm;
  }
```

**Figure 3.** LED light sensing routine

## USING LEDS AS TOUCH SENSORS

While it would be possible to use simple changes in incident light as a source of input from a single LED, we found it difficult to get reliable operation for this type of touch sensor under changing lighting conditions. To produce a more reliable touch sensor, as well as expand the kinds of inputs that can be supported, we have turned to the use of pairs (and then larger arrays) of LEDs. In this configuration we use at least one LED to provide controlled illumination, and another nearby to sense with. To detect touch we first sense with the illuminating LED turned off (what we will call the *non-illuminated* measurement phase), we then quickly sense again with the illuminating LED turned on (what we will called the *illuminated* measurement phase). When no objects are near, most of the illuminating light goes straight away from the emitter and does not return to the sensing LED. However, when an object is nearby, light is reflected from the object back to sensing LED. By subtracting the illuminated measurement from the non-illuminated measurement we get a value approximating the amount of reflected light alone. Note that opaque objects may actually decrease reflected light when they touch the display surface. However, partially translucent objects such as a finger seem to provide the highest light transfer between LEDs when pressed against the display surface.

When array configurations of LEDs are used, sensing behavior can be more complicated than when simple pairs are used. For example, in the matrix configuration used in our prototype below, coupling through shared wiring and parasitic capacitance effects, as well as possible capacitive coupling to the user's finger, have small effects on the sensor result[†]. However, as is demonstrated by the measurements presented below, the basic sensing paradigm still works well, and the device remains very well behaved.

## A PROTOTYPE DEVICE

The technique described above can be used to create a robust touch sensor from two or more appropriately wired LEDs. To explore the use of this technique for richer combined display and input devices we constructed a simple prototype using a small off-the-shelf LED array, as shown in Figure 1. In this case we used a 0.7 inch high 5x7 dot matrix display (specifically Lite-On Electronics Inc. part number LTP-747KR). This display, while just a

---

[†] For example, the voltage drop across lit LEDs in adjacent rows and columns causes a small part of the light reading at a pixel to come from the unlit pixels next to it in the illuminated case, but not in the non-illuminated case.

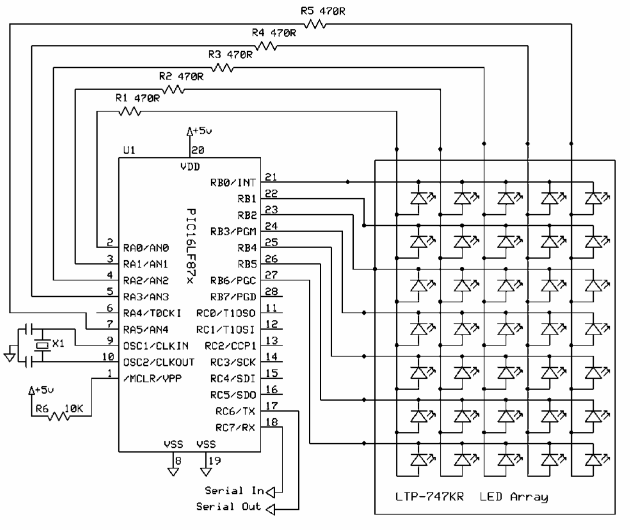

**Figure 4.** Prototype schematic

bit larger than a typical user's thumb and only supporting 35 pixels, is still capable of showing small animated displays, including for example, simple scrolling text. As a result it is fairly capable and can perform actions such as providing dynamic prompts. We chose this format because it allows several different interaction techniques to be explored in the same physical device. For example, the device can be used as a simple button by sensing a single point in the center, or as an incrementing and decrementing valuator by sensing two points that the finger can slide between.

Figure 4 shows the schematic diagram for the circuit used in our prototype. Here we use 13 of the I/O pins from a PIC16LF876 micro-controller to directly drive the seven rows and five columns of the LED array. For example, to light the top left LED in the array, micro-controller pins RA0 and RB0 would be configured as outputs and driven to logic high (+5v) and low (0v) respectively, while pins RA1…RA5 and RB1…RB6 would be configured as high impedance inputs. Note that this circuit requires a minimum of components – using the typical resonator and reset pull-up resistor associated with the microcontroller, then adding just the LED array and associated current limiting resistors. This is the same circuit that would be used for driving the LED array in an output only fashion; so in this case, the addition of an input capability can be performed entirely in software.

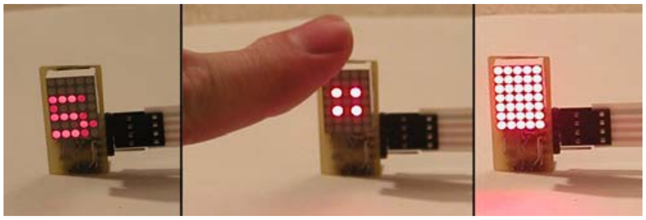

**Figure 5**. Touch sensor prototype in use
Left: part of "Press here" scrolling text prompt,
Middle: sensing, Right: after press feedback

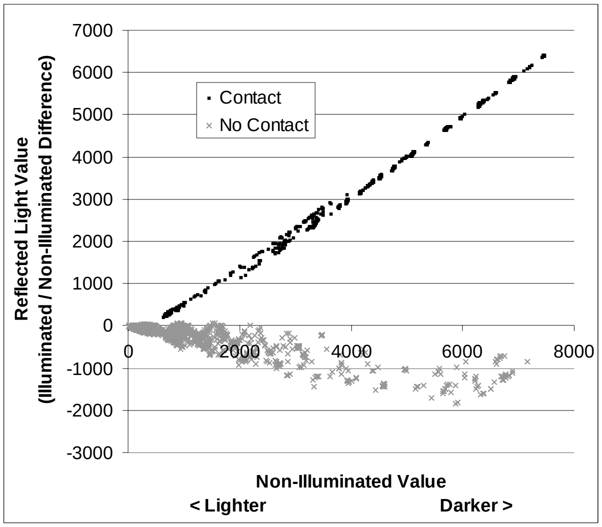

**Figure 6.** Sensor response under differing ambient light

In this design, sensing can be done at any of the 35 pixel positions. For interior sensing points illumination is provided by lighting the four corner pixels surrounding the sense point as shown in Figure 5 (where the center pixel is being sensed). When sensing corner or edge pixels, only one or two of the adjacent corner pixels, respectively, is lit.

Figure 6 shows the values returned by the prototype sensor under a variety of ambient light conditions ranging from being held ~3cm from a 100 watt incandescent lamp (left), to near total darkness (right). Two sets of measurements are shown, a *non-contact* set with the device not being touched, and a *contact* set where a finger is pressed onto the surface of the device. This data was gathered in a dark room with the device in proximity to a 100 watt incandescent bulb controlled by a dimmer. As illustrated in Figure 5, the sensing was done at the center pixel of the array. Values in Figure 6 are expressed in raw sensor units. One sensor unit corresponds to eight iterations of an inner sensing loop very much like the one illustrated in Figure 3, and corresponds to about 50 µsec of elapsed time (although timing is not exact). The x axis of Figure 6 indicates a value returned from the non-illuminated sensing phase and hence reflects ambient light incident on the device, either directly, or through the body of the user's finger. The y axis of Figure 6 indicates the value returned by the full sensor computation which is the difference between the illuminated and non-illuminated sensing phases, and hence corresponds to reflected light (and device leakage).

Because there is a band (75 to 190 units) which fully separates the contact data points from the non-contact data points, a simple threshold can be employed to separate contacts from non-contacts. The value sets are clearly approaching each other when extremely bright lighting (~3cm from a 100 watt bulb) is used. However, such extreme situations are not likely to occur in most practical uses, since the output of the device is impossible to see. Our prototype exhibits robust behavior beyond light levels allowing one to read the display using a conservative threshold of 500 units. If desired, it is also possible to operate under more extreme conditions. Since sensing only takes a few milliseconds to complete in the high lighting case, when the device detects extremely bright conditions, it can take multiple samples and use a median filter or other voting scheme to control its action with greater assurance of accuracy.

Another issue with the device is false positives due to lighting which changes during a sensing cycle. If very fast moving shadows are present in the environment, significant changes in ambient lighting can occur between the illuminated and non-illuminated phases of a single sensing cycle, resulting in false readings when they are subtracted. To mitigate this issue a technique analogous to debouncing a physical switch is employed – requiring the triggering condition to be present for two full sensing cycles before it is acted upon.

## SAMPLE INTERACTION TECHNIQUES

To provide some early exploration of the design opportunities presented by this device, we have programmed it to implement two simple interaction techniques: a text prompted push button and a simple incrementing and decrementing valuator. The button, illustrated in Figure 5, works by scrolling or flashing a short textual prompt, then sensing the center pixel of the display. If a touch is detected a confirmation flash is produced (and the associated action fired). If no touch occurs within a specified timeout period, the prompt is repeated.

The valuator is implemented by sensing two points, one near the top center of the array, and one near the bottom center. If the finger is placed over the top sensing point alone the value is incremented at a programmable rate. Similarly if the finger is placed over lower location it is decremented. Feedback is optionally given by a display filling bar which rises or falls (wrapping around as needed), and which is shown alternating with sensing rounds.

## EXPERIENCE AND LIMITATIONS

In design and informal use of both of these interactions we found that there are significant advantages to being able to display dynamic prompts within the confines of the small device. However, these advantages are muted for actuation feedback because the user's finger will typically block some of the display area. Through iteration, this led us to consider the large and prominent actuation displays that remain in our current designs.

Since this technique relies on properties not specifically engineered into the device, sensing with some LEDs may not be as robust. For example, we were unable to sense using an RGB LED because the casing was specifically designed to mix the output colors, and so there was little difference between light returned in touched and non-touched states.

Another important limitation of this technique is its comparatively slow sensing rates in low ambient light conditions – reaching as high as several 100 msec. In those cases it is advisable to sense in the illuminated condition first, then sense the non-illuminated condition only to the threshold point. A related issue that emerged was a pronounced flashing from the alternating illuminated and non-illuminated rounds in low light settings. If this is undesirable for a given application, sensing can begin using a repetition of the illuminated phase alone so that the illuminating pixels are always on. If a substantial change in the value for this phase is discovered, then a full illuminated / non-illuminated round can be performed to remove false positives from changing ambient light alone.

## CONCLUSIONS

The technique described here offers interesting advantages in its ability to provide both a small display and an input device in the same small package, allowing itself to be dynamically configured to operate in several different ways as needed by the context or interaction flow. It is potentially inexpensive to deploy, especially in situations where many or all of the requisite components are already part of the design for display purposes. Although if does not provide direct tactile feedback, this technique has the advantage of no moving parts and an ability to work while sealed within a case, enclosure, or control panel.

While only a few of the potential uses of this technique have been described here, we hope that it can join other touch sensing techniques such as [1, 2, 4, 5] to expand interaction possibilities in devices large and small.

## ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation under Grants IIS-01215603 and IIS-0325351. The author would like to thank Dale Iffrig for sparking an interest in electronics in him many years ago.

## REFERENCES

[1] Buxton, W., Hill, R. and Rowley, P., "Issues and Techniques in Touch-Sensitive Tablet Input", in ACM Computer Graphics, v19 n3, July 1985, pp.215-224.

[2] Dietz, P, and Leigh D., "DiamondTouch: A Multi-user Touch Technology", in Proceedings of the 14th annual ACM symposium on User Interface Software and Technology, Orlando Florida, November 2001, pp. 219-226.

[3] Dietz, P, Yerazunis, W., and Leigh D., "Very Low-Cost Sensing and Communication Using Bidirectional LEDs", in Proceedings of UbiComp 2003, Seattle, WA, October 2003, pp. 175-191.

[4] Hinckley, K. and Sinclair, M., "Touch-sensing input devices", in Proceedings of the SIGCHI conference on Human factors in computing systems, Pittsburgh , May 1999, pp.223-230.

[5] Rekimoto, J., "SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces", in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Minneapolis, April 2002, pp. 113-120.