MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Very Low-Cost Sensing and Communication Using Bidirectional LEDs

Paul Dietz, William Yerazunis, Darren Leigh

TR2003-35 July 2003

Abstract

A novel microprocessor interface circuit is described which can alternately emit and detect light using only an LED, two digital I/O pins and a single current limiting resistor. This technique is first applied to create a smart illumination system that uses a single LED as both light source and sensor. We then present several devices that use an LED as a generic wireless serial data port. An important implication of this work is that every LED connected to a microprocessor can be thought of as a wireless two-way communication port. We present this technology as a solution to the "last centimeter problem", because it permits disparate devices to communicate with each other simply and cheaply with minimal design modification.

To appear in UbiComp 2003, Seattle, Washington, October 12-15, 2003

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2003
201 Broadway, Cambridge, Massachusetts 02139

1. Version submitted to Ubicomp 2003.

Very Low-Cost Sensing and Communication Using Bidirectional LEDs

Paul Dietz, William Yerazunis, and Darren Leigh

Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, Massachusetts 02139 USA
{dietz,wsy,leigh}@merl.com

Abstract. A novel microprocessor interface circuit is described which can alternately emit and detect light using only an LED, two digital I/O pins and a single current limiting resistor. This technique is first applied to create a smart illumination system that uses a single LED as both light source and sensor. We then present several devices that use an LED as a generic wireless serial data port. An important implication of this work is that every LED connected to a microprocessor can be thought of as a wireless two-way communication port. We present this technology as a solution to the "last centimeter problem", because it permits disparate devices to communicate with each other simply and cheaply with minimal design modification.

1 Introduction

Light Emitting Diodes, or LEDs, are one of the most common types of interface components. Their diverse applications include numeric displays, flashlights, liquid crystal backlights, vehicle brake lights, traffic signals and the ubiquitous power-on indicator light.

Because LEDs are so commonly used as light emitters it is easy to forget that they are fundamentally photodiodes, and as such, are light detectors as well. Although LEDs are not optimized for light detection, they are very effective at it. This interchangeability between solid-state light emission and detection was widely publicized in the 1970's by Forrest W. Mims [1][2], but has been largely forgotten by LED users.

1.1 Ambient Illumination Sensing with LEDs

Recently, we have been investigating improvements for infrared remote controls of the type commonly used with consumer audio/video equipment. An area of immediate interest was the pushbutton illumination used on many remote controls. To activate the backlight, you must press a button that is nearly impossible to locate in the dark! We resolved to rectify this situation.

Our first solution was to use a capacitive proximity sensor (similar to the one described in [4]) to activate the remote control backlight during active handling. Unfortunately, turning on the backlight *every* time the remote is handled substantially decreases battery life, not only because the user often holds onto the remote continuously but also because the remote is sometimes used under good lighting conditions when the backlight is not needed. While a mode switch could be added, this would be little better than the original situation.

The obvious step was to add a light sensor to turn on the backlight only when needed. CdS photocells are inexpensive, but providing an optical path to the cell would add significant cost and complexity to the mechanical design. Recalling the photosensitive nature of LEDs, we decided to investigate using the backlight LED itself as the light detector. We developed a simple circuit for this purpose that requires one additional microcontroller I/O pin, but no other additional components compared to a traditional LED driver.

The success of the simple LED emitter/receiver circuit inspired us to consider other applications. For example, by quickly switching between the forward-biased (light-emitting) and back-biased (light-sensing) modes, it is possible to build an LED-based light source that appears to be constantly on, but is in fact periodically measuring the ambient lighting level and using this information to automatically adjust the brightness level of the LED. Our demonstration device, shown in Figure 6, has a capacitance sensor to determine that the device is being manipulated and an LED sensor/emitter to provide the backlight function.

1.2 LEDComm: Bidirectional LED Communication

While the measurement of ambient light levels has many applications, a more intriguing use of this technology is to transmit data back and forth between LEDs pointed at each other. We call this "LEDComm". We have developed simple prototypes that allow two-way serial data communication between LEDs over a distance of several centimeters.

One possible application of LEDComm is to replace Radio Frequency Identification (RFID) systems (e.g. [5]) for payment authorization and access control. To test this concept, we have created an inexpensive keychain-size device called an **iDropper** that can receive, store, and transmit data. Unlike RFID systems, iDroppers support true peer-to-peer communication, allowing new functionality such as directly transferring authority between devices without need of a special reader device.

The implications of LED-based data communication are significant. *Every LED connected to a microprocessor can be thought of as a wireless communication port.* Compared with other short-range wireless technologies such as IrDA [7] and Bluetooth [8], LEDComm has a far more limited range, and a much slower data rate. But LEDComm can be implemented at a fraction of the cost, and in many cases, may even be free. This is because LEDComm is essentially a software interface technique using existing hardware with minimal modification.

LEDComm allows us to implement communication functions in places where traditional techniques are too expensive. The power light on many consumer

appliances can now become a maintenance port for reading service information or uploading new firmware. Cell phones can transfer contact information to other phones by holding their displays next to each other. For automobiles, the standard expensive service connector can be bypassed, and all data transferred through the "Check Engine" light. (An automobile owner could even use an iDropper to capture the car's fault log and transmit it to the service center before a service appointment, insuring that the proper tools and spare parts will be immediately available when the vehicle is brought in.) There are many possible applications.

In the following sections, we describe the basic bidirectional LED microprocessor interface circuit and its use in the smart backlight. We then give a full description of LEDComm, iDroppers and various applications.

2 The Bidirectional LED Interface

Light emitting diodes emit light in a fairly narrow frequency band when a small current is applied in the correct direction. Because the current-voltage characteristic is exponential, it is difficult to control a voltage applied directly across an LED accurately enough to attain a desired current; some means must be used to limit the current. In discrete systems, this is typically done by placing a resistor in series as shown in Figure 1. Since most microprocessor I/O pins can sink more current than they can source, the configuration shown in the figure is the most common way of driving an LED from a microprocessor.

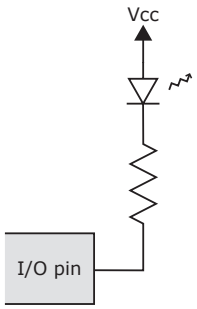

Fig. 1. Schematic of a typical LED driver

The LED is a photodiode that is sensitive to light at and above the wavelength at which it emits (barring any filtering effects of a colored plastic package). Under reverse bias conditions, a simple model for the LED is a capacitor in parallel with a current source which models the optically induced photocurrent. (see Figure 2, [3]). It is this photocurrent that we would like to measure.

An inexpensive way to make a photodetector out of an LED is to tie the anode to ground and connect the cathode to a CMOS I/O pin driven high.

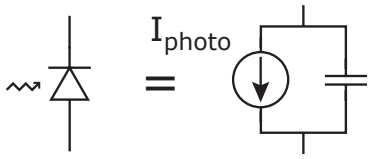

Fig. 2. Reverse-biasing an LED for photosensing

This reverse biases the diode, and charges the capacitance. Next switch the I/O pin to input mode, which allows the photocurrent to discharge the capacitance down to the digital input threshold. By timing how long this takes, we get a measurement of the photocurrent and thus the amount of incident light. This sequence is shown in Figure 3.

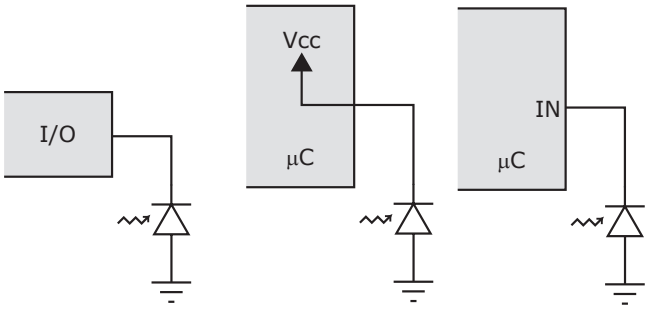

Fig. 3. LED used as a photosensor

The circuits of Figures 1 and 2 can be combined to create a general bidirectional microprocessor interface to an LED as shown in Figure 4. This is identical to the circuit of Figure 1, except that now the resistor/LED combination is placed between two I/O pins.

Figure 5 shows how the pins are driven for the two modes. Figure 5a shows the "Emitting" mode where current is driven in the forward direction, lighting the LED. Figure 5b shows "Reverse Bias" mode, which charges the capacitance and prepares the system for measurement. The actual measurement is made in "Discharge" mode shown in Figure 5c. Since the current flowing into a CMOS input is extremely small, the low value current limiting resistor has little impact on the voltage seen at the input pin. As before, we simply time how long it takes for the photocurrent to discharge the capacitance to the pin's digital input threshold. The result is a simple circuit that can switch between emitting and receiving light.

Because the circuit changes required to provide this bidirectional communication feature consist of only one additional I/O pin and printed circuit board trace (which can be provided at design time for zero additional hardware cost)

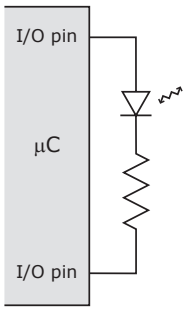

Fig. 4. Schematic of a bidirectional LED interface

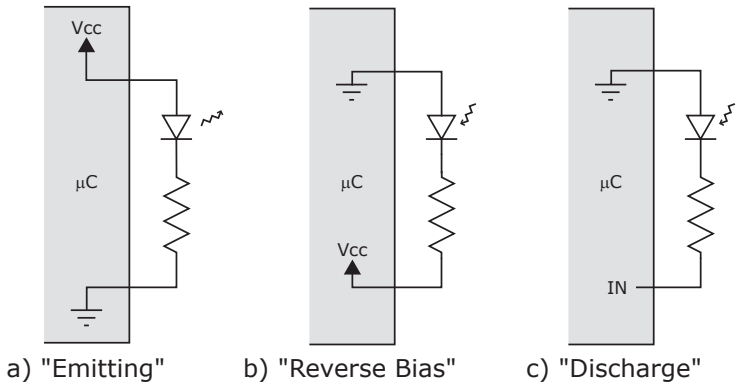

Fig. 5. Emitting and sensing light with an LED

we claim that adding this hardware feature to a device is essentially free. Of course, software and CPU runtime are also necessary to make this work.

Compare this to the cost of adding IrDA [7] (about $7) or Bluetooth [8] (more than $10) to a product. Using even a simple mechanical connector can cost several dollars because of the required level-shifting and electrostatic discharge (ESD) protection circuitry. Using an existing LED for communication can also save manufacturing costs because expensive plastic molds for the housing need not be altered to accommodate a dedicated infrared transceiver, antenna or physical connector.

3 The Smart Backlight

The Smart Backlight is one application of the bidirectional LED circuit. As noted previously, the idea of the smart remote control backlight is to turn on the backlighting *before* the user has to press a button. Also, to conserve power, we wish to turn on this backlight only when it is actually dark enough to need it.

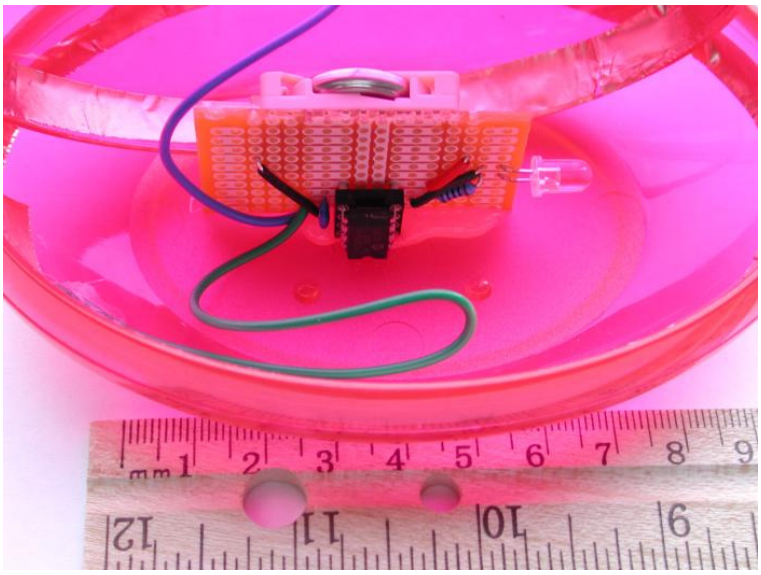

Fig. 6. The automatic backlight prototype

To demonstrate this function we created the prototype shown in Figure 6, with the complete schematic shown in Figure 7. This circuit uses a capacitive proximity detector to determine handling state. Although the basic capacitance measurement circuit is identical to that used in the buffer phone [4], we process the data to look for active handling (changes in capacitance) rather than simple

presence (increased capacitance). Many users will continue to hold a remote even when they are not actively using it, so the detection of active handling is critical for extending battery life. Of course, as soon as the user wishes to actively use the remote again, any significant motion turns the light back on.

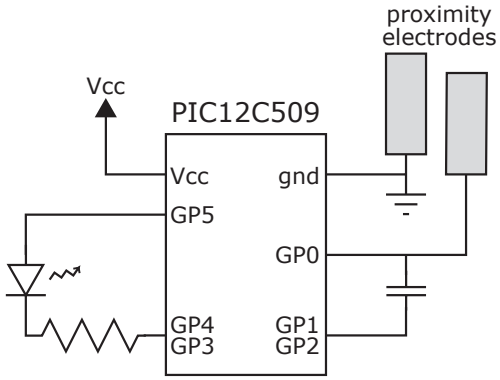

Fig. 7. Schematic for the automatic backlight

The smart backlight functions as follows: periodically, the microprocessor wakes from sleep, and measures the capacitance. If no active handling is detected, the processor goes back to sleep. Otherwise, a light measurement is made with the LED. If the room is dark, it turns on the backlight for at least two seconds. While the backlight is on, it continues to check for active handling. Each time handling is detected, the backlight timer is reset to stay on for another two seconds.

Since remote controls already contain low-end microprocessors, adding this functionality costs very little. The proximity electrodes can be part of the printed circuit board, eliminating the need for special tooling. If there are spare I/O pins available, the only additional component is a single, inexpensive capacitor for the capacitance sensor.

One might wonder if the constantly running proximity detector adversely impacts battery life. In fact, the circuit draws microwatts of power; the prototype ran continuously for 6 months on a single type CR2032 coin-cell "watch" battery. Remote controls typically use AAA or AA batteries with a storage capacity an order of magnitude higher than the coin-cell, so the power draw would be insignificant compared to the batteries' self-discharge characteristics.

4 Bidirectional Communication Protocols

In our initial experimentation with the smart backlight, we often used LED-based flashlights to test the light detecting circuit. This suggested to us that LED-to-LED communication was feasible. We constructed a simple test setup

using two identical, generic PIC microcontroller boards with RS-232 interfaces as shown in Figure 8.

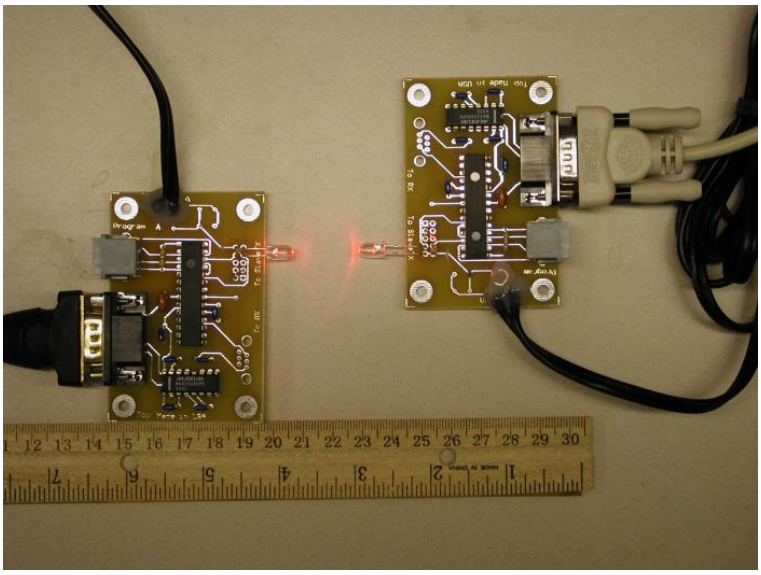

Fig. 8. Bidirectional communication with LEDs

These test boards use a simple protocol for data transfer which allows two unsynchronized devices to phase-lock to each other and exchange pulse-width-modulated data bidirectionally. A basic explanation of the protocol is that the two devices take turns flashing their LEDs at each other. A short flash indicates a 0 or SPACE state, and a long flash indicates a 1 or MARK state.

The protocol starts out on powerup with the device performing an idling cycle, transmitting a 1 millisecond light pulse followed by a 4 millisecond receive period. During the receive period, the device executes 40 light measurements, each one taking 100 microseconds. These light measurements provide only one bit of resolution, i.e. whether the incoming light flux is above or below the digital I/O pin's threshold (nominally about 1.5 volts). With only normal room light incident upon the LED there is insufficient photocurrent to discharge the capacitance below the threshold during the 100 microsecond receive period.

The oscilloscope trace in Figure 9 shows the voltage at the LED cathode during several light measurements with normal illumination. The vertical scale is 1 volt/division and the horizontal is 100 microseconds/division. The capacitance is initially charged to about 5 volts and then allowed to discharge. Notice that the voltage never drops below the threshold and so the microcontroller will always read the pin as a 1.

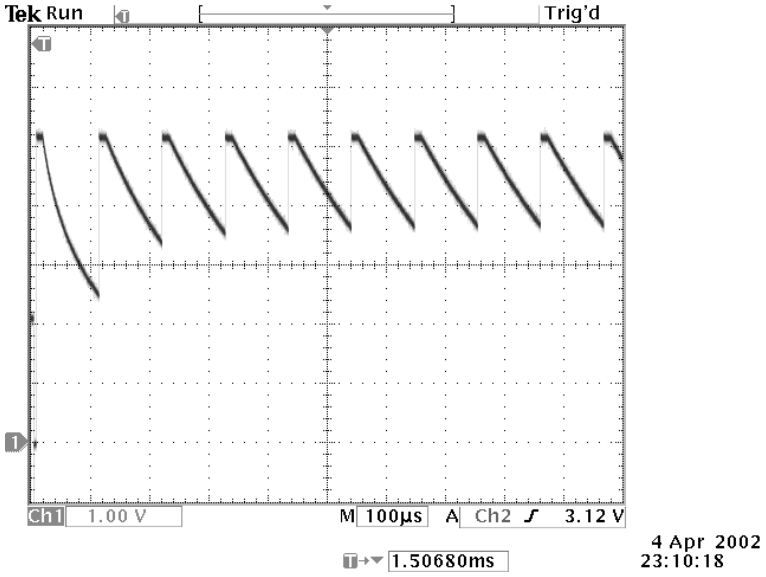

Fig. 9. A series of light measurements under normal room illumination

Figure 10 is an oscilloscope trace of the same setup, but with the LED being illuminated by another LED. The capacitance discharges completely during the measurement period, bringing the I/O pin voltage well below threshold and causing the pin to read as a 0. The idling cycle continues until at least two measurement times in succession indicate "light seen". At this point, the device assumes an incoming pulse of light from a similar device has been detected, and shifts from the idling loop of 1 millisecond ON then 4 milliseconds OFF to a slightly faster synchronizing loop, described next.

During the synchronizing loop, the transmitted light pulse is still 1 millisecond ON, but followed by a variable number of 100 microsecond light measurements. When in the synchronizing loop, the microcontroller will terminate the measurement set after either 40 are performed, or when the trailing edge of a light pulse is detected. A trailing edge is considered to be found when a pair of back-to-back measurements both indicate "light seen" followed by ten measurements without "light seen".

The execution pattern inside the synchronize loop is therefore composed of one device's LED on for 1 millisecond, then a 1 millisecond period with both LEDs off, followed by the other device's LED on for 1 millisecond, and finally both LEDs off for 1 millisecond. Even if the devices have clock frequency errors of up to 25%, they will still be able to synchronize. The nominal synchronize loop pulse rate is 250 Hz, with a 25% duty cycle. Figure 11 shows an oscilloscope trace of two devices in the synchronize loop, firing pulses of light at each other.

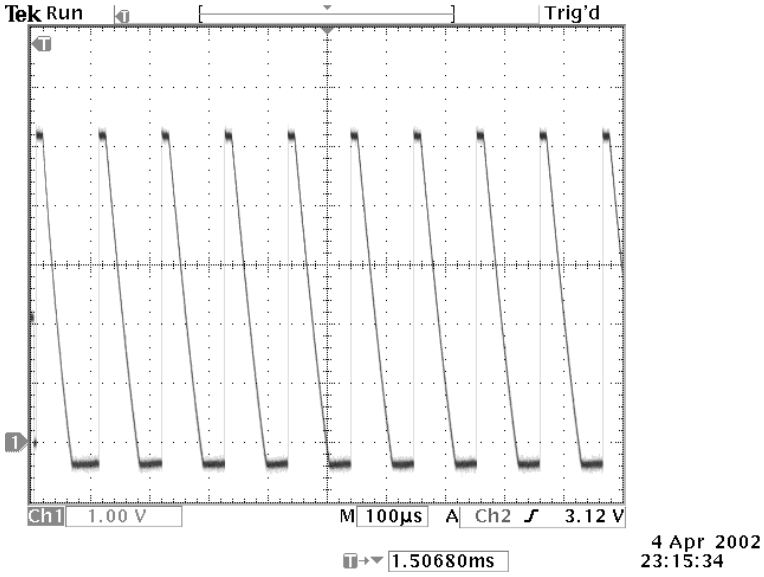

Fig. 10. A series of light measurements under LED illumination

Note that their clocks are completely independent and that all synchronization is occurring via the LEDs and the base protocol.

During communication, data bits are transmitted in asynchronous form. A 1 millisecond light pulse indicates a MARK and a 0.5 millisecond light pulse indicates a SPACE. The system normally idles with MARK bits being transmitted (the data transfer loop is the same software as the synchronize loop). During data transmission, the format starts with a single SPACE as a start bit, followed by eight bits of data, followed by one MARK as a stop bit. This is similar to the common 8-N-1 RS-232 format. The top trace of Figure 11 shows the data pulse train of a device that is idling, sending all MARK pulses. The bottom trace shows a device sending both narrow SPACE and wide MARK pulses.

To decode the light pulses, the receiving device keeps a count of "light seen" measurements for each execution of the synchronize loop. If seven or fewer light-seen measurements are tallied, a SPACE is recorded; if eight or more are seen, a MARK is recorded. The usual asynchronous deframing (dropping the leading SPACE start bit and the trailing MARK stop bit) is performed. The resulting 8-bit data word is then available to the application-level program. A simple higher-level protocol allows for error detection and correction.

The LEDComm test setup works very well. The underlying protocol transmits data at a rate of approximately 250 bits/sec in each direction. The microprocessors buffer the data and connect to a host at 38400 bps. Data transfer is robust up to a range of approximately three centimeters. Because the LEDs we

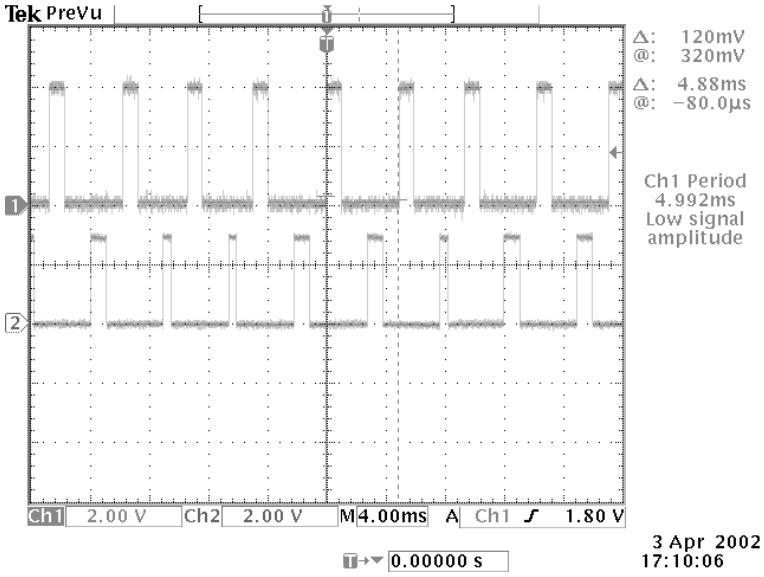

Fig. 11. Oscilloscope trace of two devices in synchronized operation

used have a fairly narrow beam angle, they permit a pointing error of only about 20 degrees.

Unlike many other protocols, this system is highly resistant to clock speed errors. Not all of our devices have precise crystal oscillators; some use the inaccurate, internal RC oscillator of the PIC microcontrollers. Even with errors in clock speed of up to 25%, communication is not disrupted. In contrast, errors over 5% in an RS-232 data link will often cause problems.

The cheap, unstable oscillators internal to the PIC devices are actually advantageous in this application. Even if two devices are powered up at the same time in the exact same phase relationship, they will quickly drift out of synchronization enough that a trailing edge will be detected and the devices will synchronize quickly into alternating flashes. In our devices, this usually happens in under 50 milliseconds. If two LEDComm devices were to both have highly stable timebases (or if both devices derived their clocks from the same source), it would be necessary to insert a jitter source (perhaps based on a hash of the device serial number) into the idle loop to assure that the two pulse trains would drift out of phase enough for a pulse trailing edge to be detected.

An additional feature of this base protocol over a balanced pulse protocol, such as Manchester coding, is that the LED gives a visible indication of idle state vs. synchronized state vs. the data-transfer state. The perceived light brightens when ready to transfer data (due to the faster pulse repetition rate) and darkens during data transfer (due to the short 0.5 millisecond SPACE pulses versus the no-data 1 millisecond MARK pulses).

5 iDroppers

To act as tangible, portable repositories of information or authorization, we have designed and constructed a device that we call an **iDropper** (for Information Dropper). Like an eyedropper, an iDropper can suck up a small amount of information, hold the information, and then expel the information on demand. Unlike an eyedropper, the iDropper can repeatedly expel the same information nondestructively.

The iDropper is meant to be used in situations where the user wishes to transfer data between devices that, for economic or practical reasons, do not have a viable user interface. This may be because the data transfer happens too infrequently to justify adding a display and keypad to the device, i.e. for diagnostic and initial setup information. An iDropper can be used to shuttle the data between the device and another which does have a user interface.

The tangible information appliance aspect of the iDropper is similar in effect to mediaBlocks [9]. The major difference is that mediaBlocks do not hold any information; they act as tokens for information that is passed along a network. The iDropper itself does hold information and can itself be used as part of the network.

The iDropper hardware is composed of a tiny printed circuit board, a single pushbutton switch (the sole user input), a Microchip PIC16LF628 microcontroller, an LED (which performs data input, data output, and user output), a 3 volt lithium "coin-cell" battery, a capacitor, and two resistors. There are an additional five solder pads so that extra components can be added for experimentation purposes. The entire assembly is smaller and cheaper than most car-alarm keychain remote controls and contains fewer components. A mass produced version should cost less than a dollar more than a similar LED keychain flashlight.

The prototype iDroppers are also equipped with an in-circuit programming connector which allows us to download code into the microcontroller and to change the personality of the device. We have also devised a small adapter board to convert this connector to Microchip's standard RJ-11 in-circuit debugging module. A pair of iDroppers is shown in Figure 12. The lower one has the adapter board attached. The large plastic part visible on the bottom of the lower iDropper is the battery holder – the largest component of the device. We have left over a centimeter of empty printed circuit board material at the back end of the iDropper so that a hole can be drilled and it can be attached to a keychain.

The default iDropper personality program is that of an information eyedropper. To suck information into the iDropper, the user presses the button twice and holds it in; the iDropper will then suck in any data stream presented to it and store it internally. Distinctive flash patterns indicate when the mode has been entered and when recording has finished. Releasing the button early will abort the process. To squeeze information out of the iDropper, the user presses and holds the button; the data is then emitted repeatedly, about once a second. This

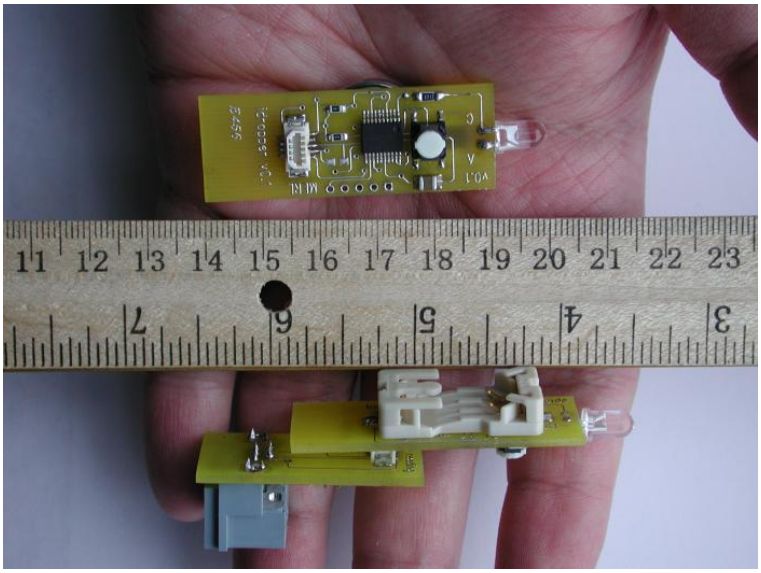

Fig. 12. A pair of iDroppers, one with programming adapter

mode appears to the eye like a simple flashlight. This is by design: an iDropper is perfectly useful as a small keychain flashlight.

The lithium battery employed in the device will allow over ten hours of continuous use. When an iDropper is not transmitting or receiving data, the PIC microcontroller goes into sleep mode. This lowers the power requirement for the entire system to below the leakage current of the battery, giving a shelf life of several years.

6 iDroppers as Intelligent Keys

One of our goals for the iDropper is to use it as an intelligent, programmable key. Although many other technologies are used in intelligent keys (RF and RFID, card-keys, etc.), LEDComm has some distinct advantages. First, it requires no physical contact so there is no mechanical wear unlike in some card-key systems. Second, unlike RF systems, it is directional and short range so the user has complete control over what is being unlocked. This allows a single key to be used for many different locks without the possibility of unlocking the wrong one just because it is nearby. Third, LEDComm is fundamentally bidirectional allowing the use of challenge/response and encryption protocols which can make the key very difficult to copy or spoof. Fourth, the visible nature of the LED allows for some user interface. At the very least, the user can easily tell whether the device is operating or if the battery is dead. Fifth, LEDComm readers are much easier and less expensive to implement than keycard or RFID readers. This

could important in situations where the number of locks is on the same order as the number of keys.

The sixth, and perhaps most interesting, advantage is that LEDComm is capable of peer-to-peer communication. Any LEDComm device can pass information or authorization to another LEDComm device (assuming the application software allows it). In this case, an iDropper with the standard "suck/squeeze" personality program can learn the unlock code, and pass that to yet more iDroppers. This ability to delegate authority is completely unique and not a capability of smart cards or RFID tags.

To demonstrate this use of the iDropper as an intelligent key, we added a reed relay and an external power supply to one device, and wired it into the security system that locks and unlocks our site's front door. The iDropper's LED was aimed out through the glass windows of the lobby. The iDropper personality program was altered to do nothing until it received the proper (secret) command, and when it received that command, to activate the relay to unlock the door for five seconds. Figure 13 shows the test setup.

We then programmed one iDropper with the correct code, and (as expected) used it to unlock the door. Taking advantage of the LEDComm peer-to-peer ability, we passed the unlock code to several other iDroppers which were also used to unlock the door.

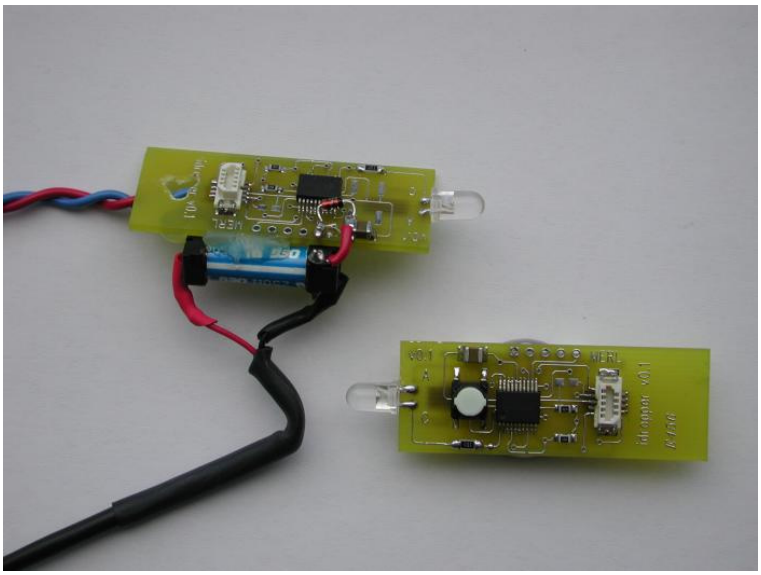

Fig. 13. iDroppers being used as a door lock (left) and key (right)

7 iDroppers, Authentication and Security

In some applications, the peer-to-peer ability to transfer information or authorization is desirable. In other applications, such as financial and other secure transactions, authentication is as important as transfer, and the uncontrolled passing of authority must be prevented. An unfortunate side effect of the programmable nature of the iDropper is that there is no guarantee that another device will respect any "do not forward" data tags that may be inserted by an application. Non-transferable authorization and unforgeable proof-of-identity are difficult problems with many subtleties. A solution for even a highly constrained scenario would involve hardware, software and cryptographic techniques beyond the scope of this paper.

However, simple cryptography is quite possible and can be used to keep iDropper transactions secure from eavesdropping and spoofing. The microcontroller used has sufficient power to implement common symmetric cryptographic algorithms. These require the sender and recipient to share a secret key so communication between any two devices must be configured in advance. The iDropper has enough memory to hold many symmetric encryption keys and can therefore be set up to talk to a number of other devices.

Zero-knowledge proofs and public-key (or asymmetric) cryptography [6] would enable an iDropper to securely prove its identity and communicate with any device that had access to published information; no shared secrets would be necessary. Unfortunately, all published algorithms for these require extensive calculations and data communication which, although available on modern fast workstations, are not possible in the extremely limited computational environment of an iDropper. The small lithium battery contains only enough energy to run the 4 MHz processor for a few hours, so a calculation that a workstation could complete in seconds would consume an iDropper's entire battery life. We are currently investigating algorithms for zero-knowledge proofs that require only the limited processing we have available.

One side effect of the limited data rate and battery lifetime of the iDropper is that the total amount of data it can communicate in its battery lifetime (about 10^6 bytes) could be held in an inexpensive non-volatile memory. This would allow the use of one-time pad encryption, either for extreme security or for use with a very limited microprocessor.

8 Every LED is a Communications Port

Although almost every electronic device made today contains a microcontroller and (theoretically) has sufficient capability to communicate with similar devices, the cost of including a communication link often precludes two devices that are sitting side-by-side from talking to each other. This is the "last centimeter problem".

With LEDComm technology, every LED becomes a potential communication path. This has broad implications because LEDs are widely used as power-on

indicators in microcontroller-based devices. The indicator is usually not wired directly to the power supply, but is connected through the microcontroller so that a minimal user interface (some blinking) is available. With the proper modifications, the indicator can be used to communicate with an iDropper or other LEDComm-enabled device.

The ability to cheaply and easily transfer data between a device with a user interface and one without will permit designers to add more capability to inexpensive products. Small, portable products can be carried to the user interface machine for interaction there, while larger ones may require that the user carry data back and forth with an iDropper-like device. Here are some applications of LEDComm-enabled devices that we have been considering:

1. The power indicator LED of a modern CRT monitor is connected to its CPU so that it can blink to indicate a low-power "standby" state. Newer models are equipped with USB, both to control monitor settings and to provide easy access for mice and keyboards. Adding LEDComm can provide a complete data path from the power LED to the host computer, allowing an iDropper or similar device to be used as a key. This could be used instead of or in addition to a password to log in to the computer, or could be used as a cryptographic authentication device for electronic commerce. A similar technique could be used with keyboard indicator lights.
2. A homeowner could copy the full diagnostic state of his or her malfunctioning washing machine by iDroppering the data from the power-on LED and carrying it to their PC for upload to a service site on the web. No special display or connector on the washer is needed, nor is it necessary to run a data cable to the computer.
3. Exchange of telephone numbers and electronic business cards with a new acquaintance could be performed by holding two mobile phones together, display to display, while the backlight LEDs exchange the relevant data.
4. A programmable, electronic doorbell might need to have its tune changed seasonally. This could easily be accomplished by composing or downloading a new tune on a computer and transferring that to the doorbell with an iDropper. There is no need to remove and carry the doorbell itself (with the attendant wiring difficulties) or to implement an expensive wireless data link.
5. LEDComm could be used for mobile phone-based electronic payment. A purchaser could use the user interface and wireless data connection of the telephone to set up an electronic payment transaction with their bank. They could then point the phone's LEDComm-enabled power LED at the vending machine of interest, completing the transaction. The LED's directionality and short range are an advantage here because they allow the user to specifically and naturally indicate for which machine the payment is intended.
6. Inexpensive toys using LEDComm could communicate with each other to synchronize their actions or provide emergent behavior among a group of related toys. They could also communicate with the family computer to interact with programs running there or to download new functionality.

9 Future Work

The range of communication for an LEDComm device is currently quite short – a few centimeters at best. The data rate is also fixed at 250 bits/second in each direction. These two values (range and data rate) are inversely related: altering the base protocol to use longer integration times can yield a longer range, while increasing the data rate decreases the total integrated light captured by the LED, which lowers the signal-to-noise ratio and so limits the maximum distance between the LEDs.

We are currently perfecting improvements to the LEDComm hardware and software to permit operation at substantially longer ranges (greater than one meter) with somewhat slower data rates, as well as operation at over 1000 bits/second at the existing maximum range. The operating conditions can be detected by the system so that the data rate can be automatically raised or lowered as conditions permit.

LEDComm-enabled devices will have to become widespread to be useful. This will require a standardization process for the several layers of communication protocol, optical characteristics, etc.

References

1. Mims, Forrest M., III, *Siliconnections: Coming of Age in the Electronic Era*, McGraw-Hill, New York, NY, 1986.
2. Mims, Forrest M., III, *LED Circuits and Projects*, Howard W. Sams and Co., Inc., New York, NY, pp. 60-61, 76-77, 122-123.
3. Graeme, Jerald, *Photodiode Amplifiers: Op-amp Solutions*, McGraw-Hill, 1996, pp. 4-7.
4. Dietz, P. and Yerazunis, W., *Real-Time Audio Buffering for Telephone Applications*, in Proceedings of UIST 2001, Orlando, FL, Nov. 11-14, 2001, pp. 193-4.
5. Mobil Speedpass is an RFID payment system. Information is available at `http://speedpass.com`
6. Schneier, B., *Applied Cryptography*, second edition, John Wiley and Sons, New York, NY, 1996, pp. 101-111.
7. Infrared Data Associates (IrDA) information and specifications can be obtained at the association's web site: `http://www.irda.org`
8. Bluetooth information and specifications can be obtained from the Bluetooth SIG, Inc. website: `http://www.bluetooth.org`
9. Ullmer, B., Ishii, H. and Glas, D. *mediaBlocks: Physical Containers, Transports, and Controls for Online Media*, in Computer Graphics Proceedings (SIGGRAPH '98), July 19-24, 1998.