

An Infrared Link for Low-Cost Calculators and Printers

by Steven L. Harper, Robert S. Worsley, and Bruce A. Stephens

MANY OF HP'S HANDHELD CALCULATOR customers have told us of their need to create a permanent record of their calculations easily. The need seems to be particularly acute for users of financial calculators. A banker or real-estate agent often wants to give a client a printed record of loan information or perhaps an amortization table showing interest paid and remaining balance. Because of this customer feedback, printing capability was high on the list of design priorities for the HP-18C Business Consultant.¹

There are two approaches to providing this feature. One method is to design another model of the calculator with a built-in printer. The earlier HP-91 and HP-97 Calculators² are examples of this approach. Although the portability of the unit suffers somewhat, this is a good solution for the customer who uses the printer a lot and knows of the need for this capability before buying the calculator. It is less ideal for the person who really wants full portability or didn't realize the need for a printing capability until after the calculator was purchased. For this individual the other approach, that of designing a separate accessory printer which connects to the calculator in some way, is the better solution. This approach also yields somewhat lower development cost and a shorter design schedule. The HP-41C Calculator uses this method,^{3,4} and the decision was made to design an accessory printer for the HP-18C also.

Interconnection Method

One very critical design area for such a printer is the means of interconnection to the calculator. The HP-IL interface⁵ is used in the HP-41C for connection to a number of peripherals, including an accessory printer. This would have provided much more capability than was needed for a simple printer-only interface, and would have cost more. In addition, we had received a number of complaints from customers about the inconvenience of the cables for the interface and battery recharger with our portable products. With this in mind, we began to investigate the possibility of a wireless interface for a printer powered by disposable batteries.

Infrared transmission seemed to be the only wireless technology that allowed the use of low-cost, low-power, and readily available components. Infrared remote-control units for television sets and videocassette recorders have been in use for years. Their transmitting element usually consists of one or more infrared light-emitting diodes driven so as to produce short bursts, each containing a few pulses of invisible infrared light at a wavelength of 940 nm. The pulses within each burst have a repetition frequency of about 40 kHz and their intensity is proportional to the current through the diode(s). Information is encoded by varying the time between bursts.

The receiver uses a photodiode which acts as a weak current source with an amplitude proportional to the incident light intensity. A sensitive gain-controlled preamplifier provides frequency selectivity, demodulation, and conversion of the photodiode current to logic voltage levels. Since the HP-18C Calculator already has a 32.768-kHz oscillator for its time and alarm functions, our infrared link uses 32.768 kHz as the infrared modulation frequency, rather than the slightly higher value used by conventional remote-control units.

Silicon photodiodes respond to a broad range of wavelengths, including most of the visible spectrum. To reduce unwanted signals, the photodiode in the receiver is often encapsulated in a material that is opaque to visible light, but transparent to the longer infrared wavelengths. Various kinds of optical filter materials are also used as windows in front of the photodiode to reduce extraneous light further by moving the cutoff wavelength still closer to 940 nm. The infrared receiver in the printer for the HP-18C uses both of these techniques. However, incandescent lights and sunlight still have strong components in the infrared range. The modulation of the bursts allows some frequency selectivity that reduces this dc and low-frequency optical interference still further to provide a higher signal-to-noise ratio and an improvement in the distance over which the link will operate reliably.

Implementing the Link

The cost of adding the infrared transmitting circuitry in the calculator is minimal. This is especially important since it allows us to include that part of the link in every calculator without penalizing the customer who does not want the printing capability. Otherwise it would be necessary to resort to a more expensive and complex plug-in module system, which would likely add as much or more cost to the calculator, even without the module.

Unfortunately, the receiver end of the link is substantially more expensive. The infrared preamplifier chip requires several discrete components and a regulated supply. The additional printed circuit board area needed was much more than could be accommodated in the calculator. Because of this, it was necessary to make the infrared link a one-way-only interface—the calculator is the transmitter and the printer is the receiver.

While cost and size considerations made this decision obvious, it nevertheless represented a trade-off in link performance. The printer cannot send handshake signals to the calculator, indicating readiness to receive more data. Because of this, after enough characters are transmitted to fill the printer's buffer, the calculator must carefully pace subsequent transmissions so that even the slowest printer can keep up. When the printer has fresh batteries, it can

print at a rate of slightly more than one line per second. When the batteries are near the end of their useful life, the rate is reduced to just over one half line per second. It would have been nice to be able to run the printer from a regulated supply rather than directly from the batteries so that the speed would be more constant. Unfortunately, the current requirements are quite high (up to 1.5A average, 3A peak while printing) and the inclusion of such a supply would have escalated the cost of the printer far beyond its design objectives. In any case, the user will seldom notice this trade-off since most printing with the calculator is done in segments smaller than the size of the 200-character print buffer. Under these conditions, the printer performs as fast as it is able, given the current state of the batteries. For longer print segments, the first eight lines or so print as fast as the printer can go, and then the system slows to a rate of slightly more than one half line per second.

One of the most important specifications for any interface is the maximum data rate. For this infrared link, the rate is slightly less than 80 characters per second. Other important parameters are distance and directional sensitivity. Commercial infrared remote-control units typically have a range of thirty feet or more. This is neither necessary nor desirable for calculator-to-printer communication. The infrared link obeys a square-law response with respect to distance, that is, doubling the range requires four times the transmit drive current for a given receiver sensitivity. This quickly becomes a problem for the calculator's power supply. In addition, a long-range capability in this application creates a potential problem where one individual's calculator might interfere with someone else's printer nearby. The minimum distance is about 18 inches to allow comfortable desktop operation, but the maximum range is no more than a few feet to avoid such problems.

The infrared light-emitting diode in the calculator has an integral molded plastic lens which forms a somewhat directional radiation pattern. At an angle of slightly less

than thirty degrees from the directional axis of the pattern, the intensity falls to half its maximum value along the axis. The photodiode in the printer has no lens and its sensitivity is much less directional. Its response is proportional to the perpendicular area facing the source and is therefore a cosine function. At sixty degrees the response will be half the maximum. If the calculator's batteries are fresh, and the calculator and printer are lined up facing each other, the range is about four feet. On the other hand, if the batteries are about ready for replacement and the calculator faces thirty degrees away from the printer, the maximum distance over which the link will operate is about 18 inches. Naturally, obstructions in the path and reflections can significantly change these values.

How the Interface Works

Fig. 1 diagrams the circuitry used for the infrared link. A combination of hardware and firmware in the calculator generates the proper gated pulse waveform to drive the infrared light-emitting diode. A resistor connects the microprocessor port to the base of a bipolar transistor that acts as the driver device to handle the relatively high current peaks required. In the HP-18C, the series resistor limits the light-emitting diode current to about 80 to 160 milliamperes, depending on the state of the batteries. While these currents are fairly high, it may be possible in the future to integrate the transistor driver on the microprocessor chip, and thus reduce the size and cost of the transmit end of the link in the calculator even further.

The transmitted infrared signal is converted into a current by the photodiode in the printer, and then into a very small differential voltage by the pull-up and pull-down resistors. Two coupling capacitors feed this signal into the preamplifier IC. In the preamplifier are two externally compensated gain stages with automatic gain control. The pulse bursts then pass through a tuned synchronous demodulator stage and an output integrator and pulse shaper before

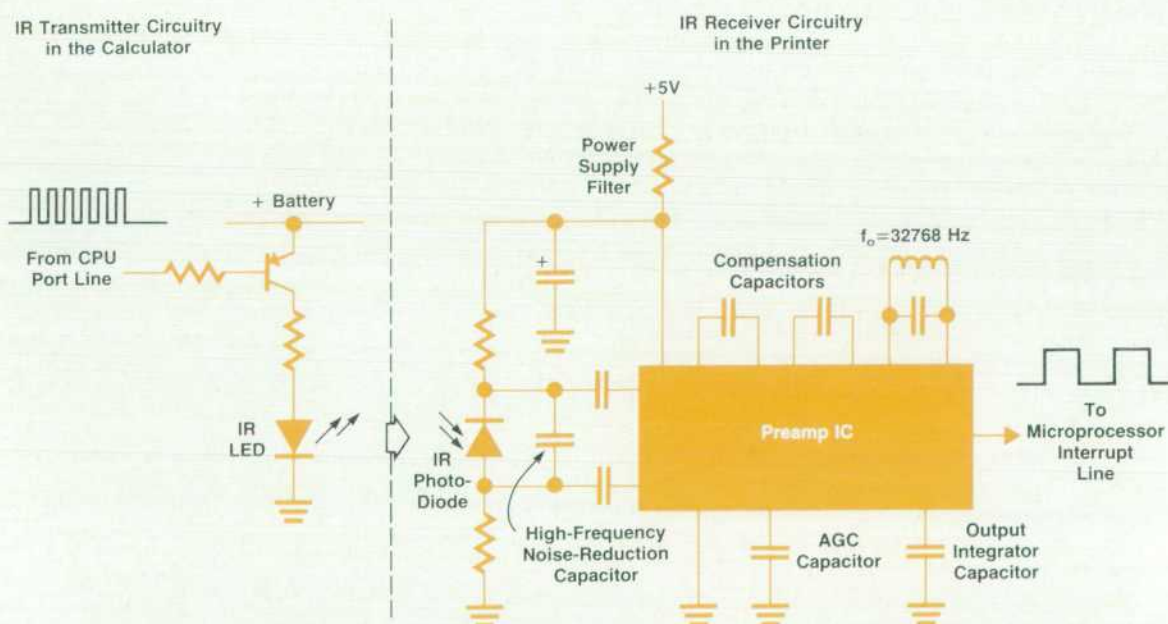


Fig. 1. Infrared link schematic diagram.

reaching the logic signal levels required to drive the interrupt line of the printer's microprocessor.

Electrical interference is a special problem for the extremely high-gain preamplifier. Careful printed circuit trace routing for power supply and ground lines and power supply filtering are essential. The typical solution to this problem is shielding, but this is particularly difficult for a small portable product with a nonmetallic case and no connection to earth ground. Several circuit configurations were tried before an acceptable solution was found. A capacitor is added in parallel with the photodiode, its value carefully chosen such that high-frequency noise is shorted to ground and prevented from getting into the first stage of the preamplifier without attenuating the 32.768-kHz signal frequency appreciably. This increases the link range substantially.

The calculator sends bursts of infrared light that are modulated with a 32.768-kHz square wave to produce 6 to 8 pulses of light in each burst. Fig. 2 shows two typical bursts. Each 32.768-kHz cycle consists of a 15.26- μ s pulse of light followed by a 15.26- μ s pause.

Coding

The data is encoded in bit times which are subdivided into half-bit times. A bit time is defined as 28 periods of a 32.768-kHz waveform (approximately 854.5 μ s). Time intervals are measured from the leading edge of the bursts. There are three kinds of bits (see Fig. 3 for an example):

- One bit. A one bit is defined as a burst at the beginning of the first half-bit time with no burst in the second half-bit time.
- Zero bit. A zero bit is defined as no burst in the first half-bit time and a burst at the beginning of the second half-bit time.
- Start bit. A start bit is defined as a burst at the beginning of three consecutive half bit times, an otherwise illegal sequence. Start-bit bursts can have six to nine 32.768-kHz pulses of infrared light.

Each frame consists of a start bit followed by 12 data bits. The first four data bits are the error correction bits and the remaining eight bits are the byte being transmitted. There must be a delay of at least three half-bit times between frames, measured from the end of the last bit time of a frame to the leading edge of the start bit of the next frame. This gives a maximum data rate of about 78 bytes/s. An example of a complete frame is shown in Fig. 3.

Error Correction

Two kinds of errors are expected to be the most likely:

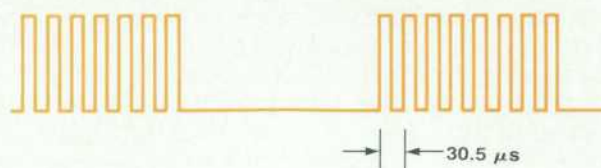


Fig. 2. Data is transmitted as a series of bursts of infrared light, each burst formed by six to eight pulses of energy at a pulse repetition rate of 32.768 kHz.

bits that are missed entirely and noise bursts introduced in addition to the correct data bursts. The bit decoding routine treats bit times with extra bursts (noise) as missed bits since it does not know which burst was data and which burst was noise. Therefore, the error correction code only has to correct one kind of error—missed bits. Flipped bits (1 $\rightarrow$ 0 or 0 $\rightarrow$ 1) are much less likely since this requires a noise burst to occur in the opposite half-bit time of a missed burst, so these errors are not corrected.

Each error correction bit encodes the parity of a subset of the data bits, allowing correction of up to two missed bits by checking the parity of separate sets of bits. The correction bits (H1 to H4) are set as the even parity of the data byte ANDed with the following masks:

Bit	Mask
H1	01111000
H2	11100110
H3	11010101
H4	10001011

Unidirectional Communication

The unidirectional infrared interface permits the calculator to talk to the printer, but the printer cannot communicate back. This eliminates the need for any receiver circuitry in the calculator. Conversely, the printer is saved the circuitry needed to transmit back to the calculator.

As a result of the unidirectional nature of the communication, the calculator has no direct information on the status (or even the existence) of the printer. It can merely transmit bursts of infrared light in such a manner that the printer can handle them. The most critical piece of information that is missing is whether the printer's buffer is full. The printer has a 200-character buffer. The calculator must make sure that this buffer never overflows. The printer is capable of emptying the buffer at a rate of so many lines per second, but the number of lines in the buffer depends on how many characters there are per line.

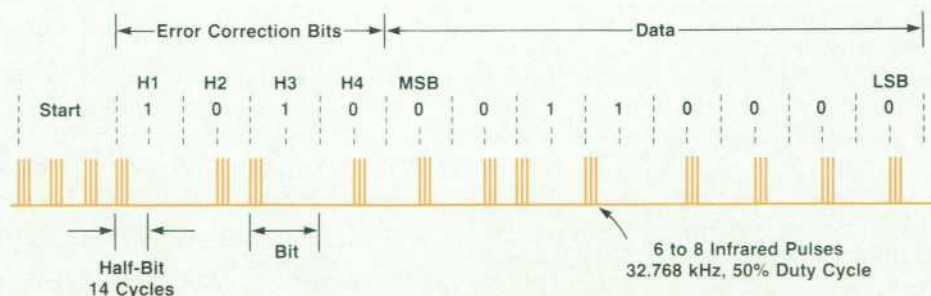


Fig. 3. Example of infrared message frame.

A simple solution would be simply to wait long enough following each transmission of a line to be sure that the printer has finished. This doesn't take much advantage of the printer's buffer and causes the calculator to remain inactive for prolonged periods of time. What we would really like is to have the calculator strive to keep the printer's buffer full. After a long pause without printing the printer's buffer is certain to be empty, so several lines can be transmitted immediately and the printer should be able to buffer them and print them as fast as possible. Only when the calculator has sent enough lines to fill the printer's buffer should the calculator have to wait to transmit more data.

A truly perfect algorithm would have the calculator keep track of when each of the lines was transmitted and how many bytes were in each line. However, this requires too much memory. Instead, some simplifying assumptions are made. All lines are assumed to contain 25 bytes (24 characters plus an end-of-line byte), which means that the printer's buffer will hold up to eight lines. Rather than keeping track of when each line was sent, a simple line counter and a record of the time when the last line was sent are used. The counter is incremented each time a line is sent and the time of that transmission is saved. Before sending a line, the calculator checks how much time has elapsed since the last line was transmitted. It then calculates how many lines the printer should have printed during that time and subtracts that number from the line count. If the count indicates that the buffer can now hold another full line, the calculator sends the line and saves the current time as the last transmission time and updates the line count to include this newly transmitted line. If the buffer cannot hold another line yet, the calculator waits and repeats the above process until the buffer empties enough to accept another line.

Assuming that the printer can print at the specified rate, this algorithm is foolproof. Since not all printers print at the same rate and they tend to slow down as their batteries deteriorate, the calculator must never send lines faster than the slowest printer can print while running at the lowest acceptable battery level. Hence, since most printers empty their buffers more quickly than the calculator is allowed to transmit, they will often pause while the calculator is waiting to send the next line.

Critical Timing in the Link

Another challenge in the printer transmission firmware was the timing of the bursts of infrared energy needed to transmit a character. As the development of the printer progressed along with the calculator, the need for accuracy of the burst timing became quite apparent. Each character is transmitted as a series of bursts of infrared energy modulated at 32.768 kHz. Each transmitted byte (frame) is divided into 27 subparts which include the start bits, redundantly encoded data, and error correction information. This scheme is designed to maximize the printer's ability to recover garbled frames.

The most common problem with infrared transmission is dropout of the infrared signal. Some of the transmitted bursts may not be detected by the printer if the calculator and printer are at the limits of their range or if something

momentarily blocks the transmission path between them. The encoding scheme used works well for recovering from lost bursts, but only if the bursts are timed accurately enough so that the printer does not get completely out of synchronization with the calculator. This would cause the rest of the bits in the frame to be incorrect.

To achieve the required timing accuracy, the calculator's microprocessor needs to start each infrared burst within a 4- μ s window. Since the nominal clock rate is 617 kHz, the microprocessor must not be more than one cycle away from the perfect time. The clock is generated by an LC oscillator whose frequency varies slightly from unit to unit and with changes in temperature and battery voltage. The calculator also contains a 32.768-kHz crystal oscillator that the firmware can use to calibrate the loops used to time infrared bursts. The exact number of processor cycles required between bursts is calculated, and by using variable cycle count instructions, this exact number of cycles is achieved. This calculation is performed before each line is sent to guarantee that the clock has not drifted significantly. The flowchart in Fig. 4 shows what is done.

The printer receiver decodes the incoming infrared bursts by measuring the time intervals between successive bursts. This time-interval measurement is subject to four sources of error:

1. After being detected by the photodiode, the incoming bursts go through a preamplifier, which introduces additional timing error. The input to the preamplifier consists of a series of 15.26- μ s-wide pulses of current separated by pauses. The preamplifier may not respond to the bursts of pulses in exactly the same way every time. Its output might go true on the n th pulse in one burst and on the $(n+1)$ th pulse in the next burst, which would make the interval between bursts at the preamplifier output look 30.5 μ s too long. Similarly, the output could go true on the $(n+1)$ th pulse in one burst and on the n th pulse in the next burst, which would make the interval look 30.5 μ s too short. Therefore the error from the preamplifier is approximately $\pm 30.5 \mu$ s.
2. The output of the preamplifier goes to the interrupt pin of the printer's microprocessor. Since there are places in the printer's firmware where the interrupt must be temporarily disabled, the interrupt can be delayed up to 13 processor cycles (about 34 μ s). Depending on whether this happens on the first or last pulse of an interval, the interval can look longer or shorter by 13 cycles.
3. Once the interrupt occurs, the interval between interrupts is measured using a timer in the CPU. Since the timer ticks only every 32 cycles (about 84 μ s), additional error is introduced by this granularity. For example, an interval that is really 11.3 timer ticks long will be measured as either 11 or 12 ticks long depending on when it occurs relative to the timer ticks.
4. The final error comes from the printer's oscillator speed variations. Since an LC oscillator is used, the frequency varies somewhat from printer to printer. Therefore, the actual time interval represented by the measured number of timer ticks varies with processor timing. A fast processor will show more timer ticks for a given interval than a slow processor.

The receiver code uses the time interval since the last

interrupt as an index into a fixed timing table to determine whether the current interrupt is for a one bit or a zero bit. The time interval must be adjusted by five ticks if the last bit was a zero since the interrupt for a zero occurs at the midpoint of the bit time rather than at its beginning. The ability to correct up to two missed bits means that the table must be seven half-bit times long to cover the bit sequence 1XX0 where X indicates a missed bit. Table I shows the range of timer ticks calculated for each bit-to-bit interval, taking into account the above errors as well as the number of ticks actually used in the timing table.

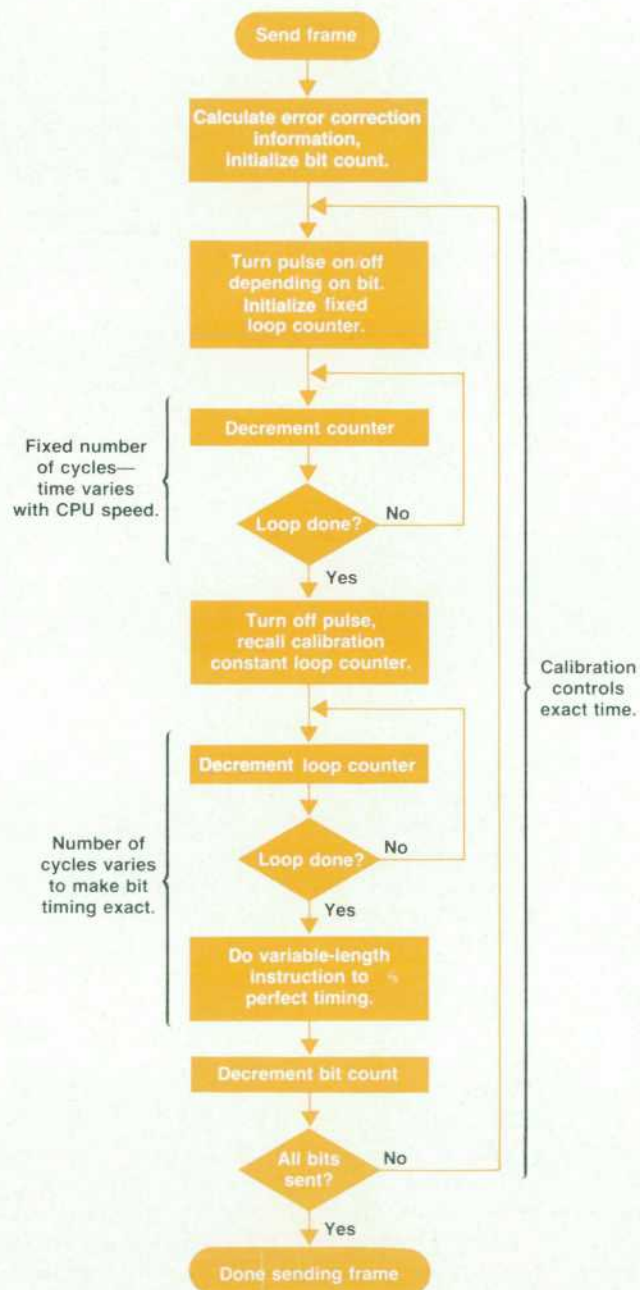


Fig. 4. Flowchart of algorithm used by calculator to adjust data transmission timing.

Table I
Timing Table

Processor Ticks	Table Ticks	Bit Received	Comments
0-2			Too short, ignore it
3-7	3-7	X	Two pulses in same bit time, treat as missed bit
8-12	8-12	1	
13-17	13-17	0	
18-23	18-22	X-1	Missed bit followed by a one bit
23-28	23-28	X-0	Missed bit followed by a zero bit
28-34	29-33	X-X-1	Two missed bits, then a one bit
33-39	34-39	X-X-0	Two missed bits, then a zero bit

The burst timing resyncs on each received burst since the time of each interrupt is saved as the starting point of the time interval between it and the next interrupt. Start-bit timing is handled as a special case. The start bit must be received correctly before the code looks for the 12 data bits. After the frame is received, the byte is checked using the error correction bits and corrected if necessary.

Looking at an earlier version of the calculated timing table, it became apparent that the calculator must transmit the infrared bursts very accurately to get reliable decoding of received frames since the intervals overlap when bits are missed. For example, if 34 processor ticks occur between bursts, the cause is two missed bits followed by either a one or zero bit, but the printer would interpret it as the latter. Table I includes the maximum allowed error in the calculator transmission as well as the printer errors. The overlap of intervals leaves a possibility of improper bit decoding, but a statistical simulation showed that the probability of this occurring is very low.

Acknowledgments

A number of people deserve special mention for the contributions to the infrared printer project. Dave Rabinowitz was the project manager. Mechanical design was handled by Dave Smith and Jack Muranami, who explain some of the mechanical design decisions in the article on page 21. Gary Podwalny did the industrial design. Theresa Gibney's efforts in manufacturing were especially important. Grant Salmonson did the production test tooling. With the help of Ng Say Ban, David Shum, and Tan Zing Chiou at Singapore, a difficult transfer to production went very smoothly. Herbert Ting provided strong support from the QA department.

References

1. S.L. Wechsler, "A Handheld Business Consultant," *Hewlett-Packard Journal*, Vol. 38, no. 8, August 1987.
2. B.E. Musch and R.B. Taggart, "Portable Scientific Calculator Has Built-In Printer," *Hewlett-Packard Journal*, Vol. 28, no. 3, November 1976.
3. B.E. Musch, "Powerful Personal Calculator System Sets New Standards," *Hewlett-Packard Journal*, Vol. 31, no. 3, March 1980.