HARDWARE HACKER

New ideas in piezoelectricity and optics, the ASCII code, doing tech research

By Don Lancaster

Let's start out with several updates on previous topics. I have just found a very good resource for pressure-transducer information. Its called the *Pressure Transducer Handbook* and is published by *SenSym,* the people who took over the old *National Semiconductors* transducer line.

The handbook has lots of hacker-type ap-notes in it. Also included is a handy slide rule that does both pressure and temperature conversions for you. Both are free on a phone or letterhead request.

Turning to shaft encoders, check out the *Optical Encoder Design Guide* from *Motion Systems,* a short but helpful tutorial. *Hewlett-Packard* also has lots of new shaft-encoder products at (for them) extremely reasonable prices. Ready to go encoders are available for $24.

They also have an even cheaper HEDS-7000 panel-mount digital potentiometer that produces 256 pulses per revolution. Unlike an ordinary pot, this one goes round and round continuously. H-P intends this jewel for hand operation and strongly recommends that you do *not* motor drive it. But it should be useful for "limited-motion" encoders such as you might need on a hacker's robotics arm. Cost is as little as $12.

Finally, H-P also has a new integrated circuit that does all the conditioning and encoding of absolute shaft encoders. Contact the company directly for more details.

Don't forget that I have some free shaft-encoder software for you, along with some other freebies that include a laser printing demo pack, an RS-232C handout, and copies of this month's ASCII code tables. I've also just freshly reprinted my *Micro Cookbook,* Volume I, so this once hard-to-get volume is now back in stock.

What is new in piezoelectricity?

Piezoelectricity is the property of certain substances that produce a voltage when they are stressed and, conversely, to stretch when they have voltage applied to them. Two very old examples of piezoelectric components are the quartz crystals used for frequency standards and the Rochelle salt crystals used long ago for phonograph cartridges and microphones.

A few years back, new piezoelectric materials came along involving man-made crystals called PZT and PZBT. These are used in such things as gas pilot ignitors and sonar transducers.

One useful resource on all this is the *Piezo Technology: Data for Designers,* 40-page manual available from *Vernitron.*

But, there's now a brand new piezoelectric ball game. A magic thin-film plastic is now available from *Penwalt* called Kynar PiezoFilm. Many new applications can now be found for this new piezoelectric film.

For instance, you can now build piezoelectric fans that have no moving parts except for their vibrating plastic blades. These can be made small enough to deliver the cooling air exactly where it is needed, reliably and quietly.

The piezo film apparently is not as sensitive as older piezoelectric materials, but its form factor more than makes up for this. The film is also pyroelectric, which means it can also be used for such things as infrared receivers and detectors. One possible application would be a low cost "hot-spot" detector for firemen looking inside walls on a post-fire overhaul.

You can get a tiny free sample by a letterhead request to Penwalt. They also have a $45 evaluation kit that includes full technical data, connectors, and samples of the various film types.

Let us know what you can come up with on this, for there are literally hundreds of possible construction projects that could be worked up with this material.

What is ASCII?

ASCII is short for American Standard Code for Information Interchange. It is *the* standard way that text messages are handled by virtually all personal computers and their peripherals.

Microcomputers speak ASCII. Printers speak ASCII. Modems speak ASCII. Disks store ASCII in their text files. An understanding of what ASCII is and how it is put to use is essential for just about anyone doing anything at all with a microcomputer.

The ASCII code was originally a 7-bit code, meaning that it could represent 128 different symbols associated with the gathering and printing of text. Because most of today's micros handle text in 8-bit bytes, there are really *two* different ASCII codes in use today.

These are called the "ASCII Code" and the "High ASCII Code." The only difference between the two is whether the eighth, or most significant bit (MSB) is a logic 0 or a 1.

Since only the bottom seven bits are needed for ASCII symbols, the use of high

	$-0	$-1	$-2	$-3	$-4	$-5	$-6	$-7	$-8	$-9	$-A	$-B	$-C	$-D	$-E	$-F
$0-	[@] NUL 0	[A] SOH 1	[B] STX 2	[C] ETX 3	[D] EOT 4	[E] ENQ 5	[F] ACK 6	[G] BEL 7	[H] BS 8	[I] HT 9	[J] LF 10	[K] VT 11	[L] FF 12	[M] CR 13	[N] SO 14	[O] SI 15
$1-	[P] DLE 16	[Q] DC1 17	[R] DC2 18	[S] DC3 19	[T] DC4 20	[U] NAK 21	[V] SYN 22	[W] ETB 23	[X] CAN 24	[Y] EM 25	[Z] SUB 26	[[] ESC 27	[\] FS 28	[]] GS 29	[^] RS 30	[_] US 31
$2-	SPC 32	! 33	" 34	# 354	$ 36	% 37	& 38	' 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
$3-	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
$4-	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
$5-	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	\ 92	] 93	^ 94	_ 95
$6-	` 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
$7-	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	| 124	} 125	~ 126	DEL 127

Table 1. Low ASCII character code ranges from hexadecimal $00 to $7F or decimal 0 to 127.

ASCII or low ASCII depends on the system and the system designer. For instance, low ASCII might be used for normal text, and high ASCII might be used for inverse text on a video screen. Or low ASCII might be used for a "normal" font, while high ASCII might be used for a "bold" font by a printer. Or, you might mix low ASCII text characters with parsed commands, all of which have their MSB set to 1. This is done in BASIC and similar languages.

Some machine-language programs may use low ASCII for all but the last character of a text string. The last character is coded in high ASCII. This is a "free" way to tell when one message ends and the next message begins.

As specific examples, the Apple II+ and IIe screen uses high ASCII for normal text, as does the older DOS 3.3e operating system. The newer ProDos operating system uses low ASCII, as do most modems and most text routines on most other personal computers.

Table 1 shows the low ASCII code, while Table 2 shows the high ASCII code. I have purposely separated these two, since it is much easier to use one or the other separately when "tearing apart" a computer program.

As you can see, there are 128 different symbols available in low or high ASCII. There are 32 values for upper-case letters and certain punctuation; 32 values for lower-case letters and some more obscure punctuation; 32 values for numbers and common punctuation; and, finally, 32 values used for control commands.

A control command is something that never appears in print, but instead causes something to happen at some end of the communicating process. As examples, the BEL command may ring a bell or sound a tone, while the CR command may issue a carriage return, and so on down the list.

Many control commands are standardized and are always used for their intended purpose. Others are rather specialized and can be "liberated" for your custom uses or special applications.

There are several ways to read these tables. The ASCII character is the big center character in each box. The small number immediately below each ASCII character is the *decimal* value of that ASCII character. Decimal values are often used in BASIC and other high-level programming languages, sometimes as a CHR$() command.

Ferinstance, a CHR$(107) represents a lower-case "k" in low ASCII. The same symbol in high ASCII would be a CHR$(235). Note that you can alway get between high and low ASCII in decimal by adding or subtracting 128, the value of the MSB in the 8-bit word.

The table row in which an ASCII character appears gives you the leftmost, or more significant, hex byte for that ASCII character, while the column gives you the rightmost, or less significant hex byte of the code.

Ferinstance, a hexadecimal $47 is an upper-case "G" in low ASCII. The same "G" in high ASCII would be a hexadecimal $C7. Note that you can always get between high and low ASCII in hex by adding or subtracting $80, once again the value of the MSB in the 8-bit word.

The bracketed letter above the ASCII control characters in these tables shows how you would enter such a control character from the Apple keyboard. Thus, to issue an FF formfeed, use a [L] from the keyboard. This [L] means to enter a "control-L", or Press down and hold the CONTROL key. Then press and release the L key. Finally, release CONTROL.

How do you handle things that are *not* in the ASCII code? The official way to do this is with escape sequences. An escape sequence consists of an escape control command, followed by enough characters to convey the task to be done in whatever needed the escaping in the first place. For instance, the Diablo 630 daisywheel printer uses an "[esc]-M" to enter its full fill justification mode.

Another way to handle ASCII exceptions is to put the main code in low ASCII and the exceptions in high ASCII. For instance, a full typesetting font might need a trademark symbol, or a letter with those european dots on top of it that might be coded in high ASCII. Then some convention is used to switch you between the two.

In the Postscript language used in the Laserwriter, a reverse slash, followed by a three-digit octal number will print any of 256 different characters or control commands, but does so only while receiving ordinary printable low ASCII characters.

More information on tearing apart ASCII text files appears in my book *Enhancing your Apple II and IIe,* Volume I (SAMS #21822).

You might like to copy Tables 1 and 2, place them back to back, and laminate them in plastic.

	$-0	$-1	$-2	$-3	$-4	$-5	$-6	$-7	$-8	$-9	$-A	$-B	$-C	$-D	$-E	$-F
$8-	[@] NUL 128	[A] SOH 129	[B] STX 130	[C] ETX 131	[D] EOT 132	[E] ENQ 133	[F] ACK 134	[G] BEL 135	[H] BS 136	[I] HT 137	[J] LF 138	[K] VT 139	[L] FF 140	[M] CR 141	[N] SO 142	[O] SI 143
$9-	[P] DLE 144	[Q] DC1 145	[R] DC2 146	[S] DC3 147	[T] DC4 148	[U] NAK 149	[V] SYN 150	[W] ETB 151	[X] CAN 152	[Y] EM 153	[Z] SUB 154	[[] ESC 155	[\] FS 156	[]] GS 157	[^] RS 158	[_] US 159
$A-	SPC 160	! 161	" 162	# 163	$ 164	% 165	& 166	' 167	(168	) 169	* 170	+ 171	, 172	- 173	. 174	/ 175
$B-	0 176	1 177	2 178	3 179	4 180	5 181	6 182	7 183	8 184	9 185	: 186	; 187	< 188	= 189	> 190	? 191
$C-	@ 192	A 193	B 194	C 195	D 196	E 197	F 198	G 199	H 200	I 201	J 202	K 203	L 204	M 205	N 206	O 207
$D-	P 208	Q 209	R 210	S 211	T 212	U 213	V 214	W 215	X 216	Y 217	Z 218	[219	\ 220	] 221	^ 222	_ 223
$E-	` 224	a 225	b 226	c 227	d 228	e 229	f 230	g 231	h 232	i 233	j 234	k 235	l 236	m 237	n 238	o 239
$F-	p 240	q 241	r 242	s 243	t 244	u 245	v 246	w 247	x 248	y 249	z 250	{ 251	| 252	} 253	~ 254	DEL 255

Table 2. High ASCII character code ranges from hexadecimal $80 to $FF or decimal 128 to 255.

What's new in optics?

A mind-boggling assortment of new and

exciting optics concepts have recently shown up in many different magazines. All have outstanding hacker potential and are crying for your use. Here's a quick rundown:

From the April 86 *Popular Science,* a magic new plastic from *Mobay Chemical* that carries total internal reflection to extremes. Much of the light that goes in the faces comes out the ends. Besides "gee whiz" neon-looking business cards, this stuff should be ideal for "sidelit" dials and panels. More importantly, it should be a very-low-cost solar electric concentrator.

From the March 6, 1986 issue of *Machine Design,* a brand new, simple, and cheap way of creating just about anything at all. You pour a liquid polymer onto a sheet or into a clear tank. Then you shine light at those portions of the polymer you want to harden into a product. Then you wash away the remaining unexposed polymer. Such things as "drilling" blind square holes are now trivally done by a hacker on his kitchen table!

The future of light-hardened polymers is even better. Take a tank of liquid polymer and scan it with three lasers on the X, Y and Z axes. Where they cross, the liquid polymer hardens. And, yes, there is a definite threshold effect where low light levels will do no hardening at all. Software routines can then describe any shape from an artist's sculpture to a precision machinery part. Which means that repair and replacement machinery parts can eventually be "shipped" anywhere in the world over a telephone!

A third optic development came out of the recent *Microsoft* CD Disk conference and is outlined in the March 17, 1986 *Infoworld.* There's a new CD disk operating system called CD-I, where the I means *interactive.* This means that DC disk systems can now handle pictures, animation, and graphics, as well as words, and that you can quickly random access any portion of the disk. It uses the Tandy Color Computer operating system.

Turning to the March 17, 1986 issue of *Electronics,* a Swedish firm named ASEA now has some very interesting fiber-optic sensors. One is a remote reading thermometer. You shine light of one frequency down the fiber, and the frequency of the light coming back is proportional to the sensor temperature. It works remotely from over a quarter of a mile away, with total electrical and safety isolation. A somewhat similar product has a cantilevered mirror at the end that forms a remote accelerometer.

A final look at the April '86 *Scientific American* reveals two fascinating optical areas. One is the optical gyroscope, which has few or no moving parts. The first hacker to develop a $25 optical gyroscope will run away with a very large bag of nickels. The time is ripe for a dramatic drop in the cost of gyroscopes, which would open them up to an incredible array of new applications, even toys.

The same issue has some construction details on creating a magic phase conjugating mirror that automatically removes distortion from optical images. Under certain circumstances, these magic mirrors even exhibit the concept of negative time!

Yes, all of this stuff is for real. And here and now.

NAMES AND NUMBERS

Hewlett-Packard
3155 Porter Oaks Drive
Palo Alto, CA 94304
(415) 857-8000

Motion Systems
7230 Hollister Avenue
Goleta, CA 93117
(805) 968-0782

NTIC
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4929

Penwalt Piezo
Box C
Prussia, PA 19406
(215) 337-6710

SAMS BOOKS
4300 West 62nd Street
Indianpolis, IN 46206
(800) 428-SAMS

SenSym
1255 Reamwood Avenue
Sunnyvale, CA 94089
(408) 744-1500

Vernitron
232 Forbes Road
Bedford, OH 44146
(216) 232-8600

How do you do technical research?

First and foremost, read as many possible trade journals involving your field of interest. So many times, you find useful and profitable information in areas where you least expect to find it.

Secondly, get thoroughly familiar with a large technical library. I'm purposely not going to give you any names or numbers to those optics references above, because I want you to go to any old library and check out a key reference book called *Uhlrichts*

Periodicals Dictionary. This will show you all the trade journals available in all fields and how to get them.

One obscure technical library tip: Practically all of the reference materials will send you *backward* through time, and thus aren't that useful in fast breaking fields. But there is a reference called the *Science Citations Index* that, believe it or not, moves you *forward* through time.

How does it work? Every time a paper is referenced, that newer paper is listed. Just about any field has its early "horses-mouth" source documents. For instance, anything competetent written on active filters *must* reference Sallen and Key. Anything competent written on unfocused solar collectors *must* reference Winston, and so on.

This can start a tree, for as soon as you find a newer author making the reference more than once, you can also chase him *up* through time.

The third way to research is with electronic data bases. Things have gotten far too complicated for any library to keep up, let alone any individual. As an availability example, *NTIS*, among zillions of other published searches, can send you a 208-entry annotated bibliography on laser optical gyroscopes if you order No. PB84-852987CAI. Cost is around $50, which is a real bargin, considering what you get.

How about another contest?

All right, a *Modern Electronics* Hardware Hacker has a problem that you may be able to help solve. I'll award the usual SAMS book to the best ten entries, and an all expense paid tinaja quest for two (FOB Thatcher, AZ) to the best entry.

The problem is to design a robotic cow. Rodeo is a very big sport in many parts of this country. To train cowboys and roping horses for certain events, a robotic cow is needed.

The pseudocow has to travel along an 80-foot-long linear track and must be able to rapidly change speed and direction, both under programmable and remote operator control. As much as possible, "found" or adapted materials are to be used. The budget is tight, say $250 maximum, not including stock personal computers or standard R/C radio controllers.

ME

NEED HELP?

Phone or write your **Hardware Hacker** questions directly to:

Don Lancaster
Synergetics
Box 809
Thatcher, AZ 85552
(602) 428-4073