From the Communications Research Laboratory of
Bill Cheek
Publisher/Editor of "*The World Scanner Report*"
Author of *The Scanner Modification Handbooks* and *The Ultimate Scanner*
(619) 578-9247 **:Fax**
bcheek@cts.com **:E-mail**

CE-232 & HB-232 NEWS AND UPDATES; #1

Price $5.00

THE FIRST HB/CE-232 NEWSGRAM!

....and probably the ***last***. Several years ago, I started tucking a note inside each CE-232 kit advising that I could start a newsletter if there were enough interest. I got only a couple of responses over the years. Obviously, people aren't interested in newsletters. Maybe because of the Internet.....

This ***first*** and probably last issue of news and updates on the CE-232/HB-232 Scanner/Computer Interface really sums it all up anyway. There isn't much that can be called "new", but there are a few late-breakers, and some inside scoop for advanced users.

HISTORY OF THE HB/CE-232

The original HB-232 went into beta test in mid 1992. It was finalized and released for public sale in the Fall of 1992.

In late 1993, the HB-232 pc-board was redesigned and renamed to the CE-232. There is no functional difference between the two, and the software did not change. The board was redesigned for easier access to some of the "hidden" and ***still overlooked*** features that will be discussed in this newsgram.

If you have an HB-232, you might have to work a little harder to get to all the neat, but subtle feature points.

Elsewhere in this newsgram are side-by-side views of the two boards with annotations of what's located where. Still, they're the same for all practical purposes.

If you have an HB-232, you might be wise to cough up for the latest CE-232 Manual. It's more detailed and accurate than the early manuals for the HB-232. Cost is **$25, ppd**, if you mention this newsgram. $25 + S&H if not.

Let's begin with a discussion of the HB/CE-232's known bugs, errors, and oddities.

BUGS, ERRORS, ODDITIES

BUGS: Frankly, just one minor bug has come to light over all these years, and I discovered it myself, ***long*** after release. The bug occurs in **HB232V13.EXE** or **CE232V10.EXE**

The **MODE** *setting of the scanner will not be correctly AutoProgrammed* ***if*** *the APF file's* **Mode** *text is in CAPS! It must be* ***lower*** ***case,*** *not upper!*

Apparently the ***AutoProgrammer*** function ignores *upper case* settings of AM, NFM, and WFM. Here is an example of what I mean in an APF file:

```
,161,0148.2575,AM,D,L,.....
```

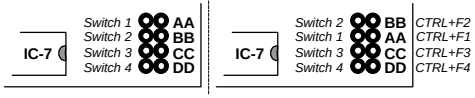
In the above, Chan-161 will be programmed with 148.2575 MHz, but will revert to the default **NFM** mode, ignoring the specified **AM** code. However, the **DELAY** and **LOCKOUT** codes (**D**) and (**L**) are not case-sensitive, and can go either way. Any of the below styles are acceptable to the HB/CE-232's ***AutoProgrammer***:

```
,161,0148.2575,am,D,L,.....
,162, 148.2575,am,d,L,.....
,163, 148.2575,nfm,D,l,....
,164,0148.2575,wfm,d,l,....
```

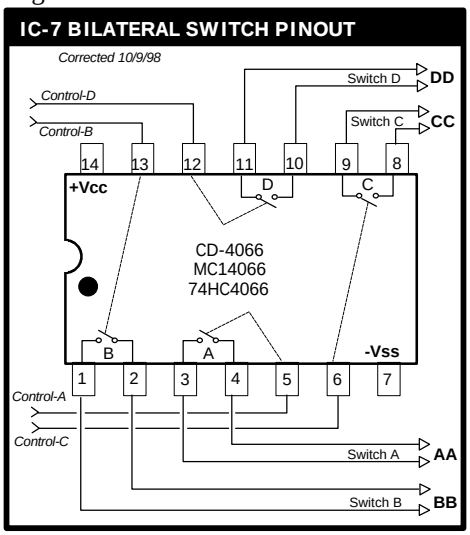
Make sure all your APF files have the **MODE** settings in ***lower case***, and all will be well. I can't imagine how this one slipped through the cracks, but it did, and not a soul noticed over the first 3-yr run of the HB/CE-232. *Maybe everyone routinely used lower-case????* Anyway, just this one *modest bug* has been found in the MS-DOS programs for the HB/CE-232.

ERRORS: Well, there aren't too many errors either; only one that could cause some grief. There are lots of insignificant typo errors in the main manual, but the big one I am speaking of is the labeling of the ***User Switch*** points, AA, BB, CC, and DD. We are only now beginning to catch up with this error in the manual and schematics, so it needs to be shown here.

The error and correction of the physical location of the ***User Switch*** outputs is depicted as follows:



Nothing more to it than that, other than I should depict the correct wiring and labeling of IC-7, the 74HC4066 chip that provides the awesome ***User Switch*** capability. So check out the schematic diagram in the next column.



ODDITIES: Not bugs, and not errors, there are oddities for which there are no real good explanations.

1. When setting up for the first time and selecting the "PRO-2006" from the <F6> config file, it may not correctly AutoProgram; that is, on a PRO-2006, a first AutoProgram session might be erratic with lots of errors.

   This can be caused by variations from one computer to the next, or even by differences between one person's construction style and the next. Frankly, we don't know for sure, but the fix is easy.

   A. You can try "playing with the Time Constant (TC) in the <F6> configuration screen, or in the **HB232.CFG** file, by raising it to a higher number.

   B. The more effective approach may be to select the "**PRO-2004/5**" scanner from the <F6> configuration screen, or in the **HB232.CFG** file. It won't matter if yours is a PRO-2006 as there isn't much difference between these scanners anyway.

   Note 6 on page 44 and another note on pg 53 of the manual explains this in detail.

   C. This oddity does not apply to the PRO-2004/5 scanners, but it may be possible to select a "PRO-2006" even for one of these, and get a faster AutoProgrammer.

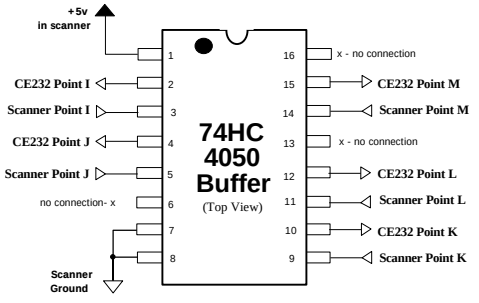
2. After the early HB-232 was released, we heard about certain other odd problems.

From the Communications Research Laboratory of
Bill Cheek
Publisher/Editor of "*The World Scanner Report*"
Author of *The Scanner Modification Handbooks* and *The Ultimate Scanner*
(619) 578-9247 **:Fax**
bcheek@cts.com **:E-mail**

CE-232 & HB-232 NEWS AND UPDATES; #1

Price $5.00

The sure-fired fix is the *now-included* **74HC4050** buffer chip (**IC9**) that should be details than this, then send $5 and ask for the "*4050 Buffer Instructions*". Instructions have been included with all CE-232's and later HB-232's. You'll need 'em only if you're one of the early HB-232 owners, or if you bought an older one second-hand or one without a manual.



Install this circuit as close as possible to Points I, J, K, L, & M on the scanner's Logic/CPU Board. (Scanner "Points" are on the Logic/CPU Board. CE-232 "Points" are on the HB/CE-232 Board.)

Words to the wise: The 74HC4050 buffer serves the same function as a ***condom***, it protects your scanner's CPU from the ravages and havoc of the outside world. Use it, even if your interface is installed inside the scanner.

3. Not so much an oddity as a **warning**, but Points I, J, K, L, and M on the HB/CE-232 board must not be in a "floating" condition when the interface is powered up. If you must power up the interface without these points connected to the scanner, then pull them high to +5v thru a 10-kΩ series resistor to each of the points, fed from the board's +5v supply.
4. In worst cases where AutoProgram and keyboard control of the scanner are erratic or problematical, you can experiment with lower values of R4-R7. We supply 47-kΩ, but values as low as 22-kΩ might prove helpful. Before you do this, though, ensure that D1-8 are good silicon switching diodes with a forward voltage drop greater than 0.5v. We've seen a bad or incorrect diode from time to time, especially from Radio Shack.
5. We have seen computers make a difference, too, especially since the Pentium came out. Newer problems are related to how the Pentium motherboards implement their serial ports. One simple solution is to get a cheap add-on serial I/O card and install it in an available 8-bit ISA slot. Set it up for 03F8/COM1/IRQ4 and 02F8/COM2/IRQ3. Reboot and go into your BIOS and reconfig the motherboard's COM1 & COM2 comports to: 03E8/COM3/IRQ4 and 02E8/COM4/IRQ3. This works most of the time. Remember that the HB/CE-232's stock DOS program works only on COM1 or COM2; not COM3/4.

installed at or near the scanner's Logic/CPU board. If you have an old HB-232, you may not

Whenever in doubt and using a modern Pentium computer or higher and Windows 95 or higher, try your HB/CE-232 in the pure MS-DOS mode first. Debug it under DOS before going on to operate under Windows.

6. Some Windows computers run the HB/CE-232 perfectly the first time, and then not again until the computer is rebooted. The fault is in the motherboard and there's nothing we can do about it, but there is a workaround:

   Do a cold reboot; fire up the HB/CE-232 and then shut it down. Click **START** followed by **SHUT DOWN**. Click **RESTART** once; press and hold the **SHIFT** key, and click **OK**. *Problem solved* until you have to reboot or turn the computer off, in which case, you have to repeat this procedure, next time the computer comes up.

LATEST NOTES: We may not stock parts and CE-232 Kits after the end of 1998, or if we do, our business process will be greatly streamlined. The radio business is going no where so we're bailing out…..maybe.

No matter what, ***Tech Support*** for the HB/CE-232 will remain available for the near future, but strictly by *e-mail*; ***not*** fax, ***not*** phone, and ***not*** postal mail.

Speaking of which, my e-mail address has been changed to: **bcheek@cts.com** where it will remain permanently. Our Web and FTP servers are down, and we do not expect to bring them back up; not for radio support anyway. The cost was ridiculous and the payback was abysmal. If you need files related to the HB/CE-232, contact me by e-mail to see what I can offer.

There is a new **Windows program** out now for the HB/CE-232, that overcomes some of the oddities and bugs discussed above, but it may introduce new ones. In any event, we do ***not*** support that program – *it's not ours*. We do encourage you to check it out, though. Read on……

NEW WINDOWS PROGRAM AVAILABLE FOR THE HB/CE-232

The HB/CE-232 has run just fine since 1992 on the original MS-DOS program that still comes supplied on the Kit's floppy disk. When you get your CE-232 built, installed, and proved up, you may prefer a modern Windows program.

know of this solution. The following schematic says it all, but if you need more

We don't sell or support this exciting new Windows program, but you can get a demo of it from the following Web site:

http://www.iaw.on.ca/~jabba/pro-turbo.htm

Read on for one reason why you may want ***ProTurbo*** for your HB/CE-232 Interface!

PRO-TURBO AND THE HB/CE-232'S A/D CONVERTERS

Pro-Turbo is a hot data acquisition and control program for the HB/CE-232. **Pro-Turbo** runs under Windows 95/98 and NT4.0/up and it does everything the old MS-DOS program did and more!

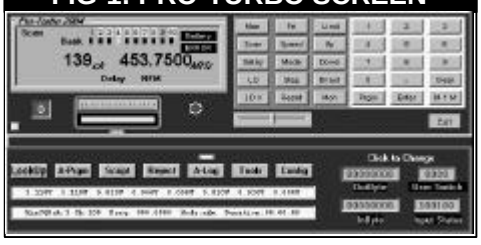
Now I can tell you about a long latent capability of the HB/CE-232 that was never exploited or announced: an ***8-channel, 8-bit Analog-to-Digital (A/D) converter***!

Indeed, that A/D has *always* been on board the HB/CE-232 but our first programmer didn't know how to program for it. He's long gone, but the ***ProTurbo*** guy rose to the challenge. The A/D is one wicked feature for those in search of the better things in life…and ***ProTurbo*** supports A/D. And it's designed for Pentiums and Windows 95/98/NT!

What's an "A/D"? It's a chip-based digital voltmeter! Since the HB/CE-232's **A/D** has 8 channels, that's like 8 voltmeters, each of which reads with 8-bit resolution. One application that comes to mind is S-metering! Just feed the output of my MOD-25 analog S-meter into one of the A/D channels for a useful HB/CE-232 output, on screen and/or to an ***AutoLog*** file!

Another application is for a Center-Tune frequency meter. In this case, a bit of extra circuitry is required. Just feed the output of a PRO-2004/5/6 discriminator to a simple "signal conditioner" circuit that outputs to one of the HB/CE-232 A/D channels. This gives a convenient real-time display of the relative accuracy of the frequency of incoming signals. The output of the WFM discriminator can feed another A/D channel, and you'll still have 5 A/D channels left.

FIG-1: PRO-TURBO SCREEN



An example of **ProTurbo**'s A/D display is shown in Fig-1. The lower of the two white strip

displays is the most recent *AutoLogged* event; just above it are the eight A/D outputs in volts. The A/D outputs are *AutoLoggable*, too! If you want to be prepared to deploy the HB/CE-232 A/D converter, here is what you need to know:

WHERE IS THAT AWESOME A/D?

IC-3 Pins 59-66 on the HB/CE-232 board are open-ended; not connected to anything. These 8 pins ARE the *input ports* to the A/D's 8 chans.

TABLE 1: A/D CONVERTER PINS

IC-3 Pin	A/D Chan	CE-232 Point	IC-3 Pin	A/D Chan	CE-232 Point
59	0	AAA	60	4	EEE
61	1	BBB	62	5	FFF
63	2	CCC	64	6	GGG
65	3	DDD	66	7	HHH

As you can see, the odd pins of IC-3 are the first four A/D channels while the even pins are the last four A/D channels.

We assigned "*point designations*" for the eight A/D channels, AAA-HHH, as shown in Table 1. For example, if I speak of "Point CCC", I'm referring to IC-3, Pin 63 or A/D-2. The HB/CE-232 uses this point labeling system throughout, to make it easier to discuss and to locate important spots. The HB/CE-232 docs presently use Points A-ZZ, so we had to assign triple-character point assignments to the A/D.

The first four channels of the A/D can be made accessible from the ***top*** of the **CE-232** board (*not HB-232*). Underneath, Pins 59, 61, 63, and 65 go to handy undrilled pads. You can drill them and insert four pinline sockets from the top into the holes; solder them to the pads beneath, and voile: convenient access to A/D 0-3.

Pins 60, 62, 64, and 66 of IC-3 are just dead-ended at the IC-3 socket and don't have handy pads. To access A/D 4-7, you'll need to either solder pinline sockets direct to these pins or solder four wires to them and bring the wires out to the top side of the board where they can connect to pinline sockets for convenient access.

Since my CE-232 is in an external metal box, I installed a second DB-25 jack and brought all eight A/D channels to selected pins on that jack. (*See Fig-7A & Item 8, pg 3*)

FIG-2: CE-232 COMPONENT SIDE

Points AAA-DDD are on solder side of board, but can be drilled as shown below IC-3 pins 59-66 are shown below circled. See text for point designations.

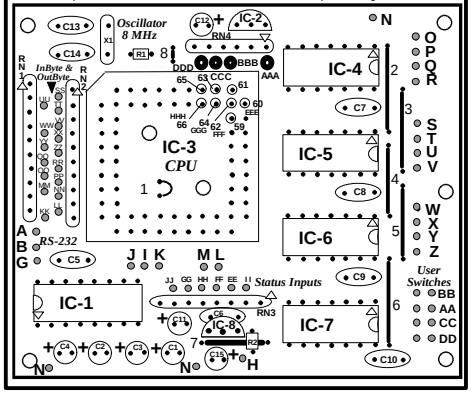


Figure 2 shows where to access the HB/CE-232's A/D ports. Other graphics in this newsgram might also be helpful. The eight A/D inputs are located on the bottom side of the board, though the topside is shown for clarity. **For CE-232's only**, you can drill the four pads for AAA-DDD as shown. You may just want to connect eight wires to IC-3, pins 59-66, and run them to a handy pinline socket strip elsewhere rather than drilling the board. Just be careful with your soldering to pins 60, 62, 64, and 66 where there isn't a lot of room for error.

The output or "readout" of the A/D goes straight to **Pro-Turbo** and the PC from the serial port via IC-1, so it takes software to interpret, scale, calculate, and present the readout. This is one of the powerful features of **Pro-Turbo** for Windows95/98/NT4. It displays, and at your option, *AutoLogs* the A/D data for you! *AutoLogged* data is cool!

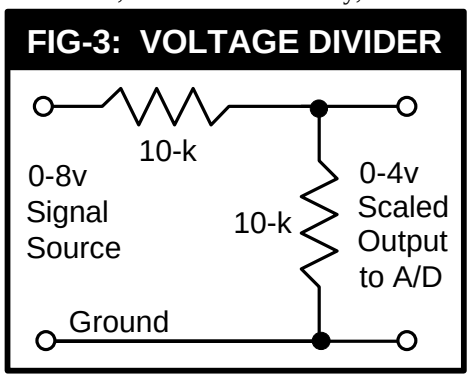
LIMITATIONS OF THE A/D

You can apply any analog or DC voltage between 0 and +5v to the A/D ports. You cannot apply negative DC voltages, AC voltages, or voltages greater than +5v. This doesn't mean you can't monitor signals larger than 5v; you'll just have to use a voltage divider (resistor) network to drop source voltages greater than +5v down to a safe level. The drop factor can later be reapplied to validate the reading. Say you want to monitor a 0 to +8v signal. Use a voltage divider to scale it in half, 0 to +4v. Figure 3 shows an example.

The 0 to +4v analog output in Fig-3 can be directly applied to any of the A/D channels on the HB/CE-232. The source 0-8v would probably damage IC-3.

A word about **resolution**: the HB/CE-232 A/D function is rated at 8-bits resolution. 8-bits is good for 0-255 points over 0 to +5v. This means one bit equals 0.0196 volt. Since the above example outputs 0-4v, then the A/D output will range from 0 to (255 x 4/5) or a raw number, 0 to 204. Not to worry, **Pro-Turbo** recalculates it back to 4 volts. You need only remember the scaling factor, in this case "2". So a 4v readout means 8v at the source. Most of the time, you won't need to scale.

FIG-3: VOLTAGE DIVIDER



Quite the opposite, for weaker signals like the 0 to +1v output of my MOD-25 S-meter circuit. You will want to amplify the signal in order to take advantage of the maximum 255-point (8-bit) resolution. We have already seen where the 255 points are spread over the full range of 5 volts. If your signal source is 0-1v, then the max bits that can be measured is just 51. So a 0-1v signal, amplified to the full 0-5v range will offer the maximum number of measurement points.

SIGNAL CONDITIONING FOR THE A/D

I suppose you can make up a regular transistor amplifier for this purpose, but that would be as much or more trouble and expense than a 4-channel LM324 op-amp.

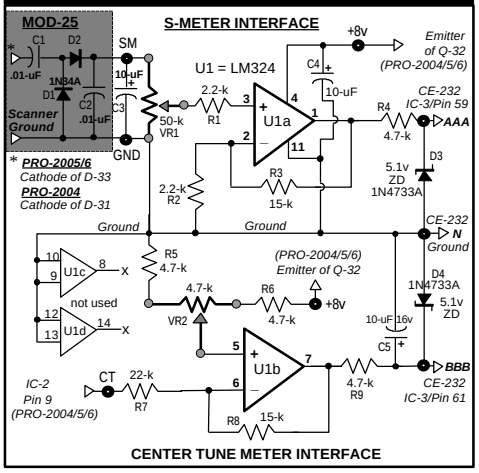
An LM324 op-amp is safer for the HB/CE-232 in the long run, and still gives four separate channels of amplification to the A/D. Assuming you need amplified inputs for all 8 A/D channels, just two LM324 chips will do the job versus eight transistors and three dozen resistors.

Figure 4 is a proved design for "conditioning" two prime signals for the HB/CE-232's A/D converter: an **S-meter** and a **Center-Tune-Meter**. It's pretty simple, and if you already have my S-meter (MOD-25) circuit, then it's even simpler. If not, then build MOD-25 as shown in the shaded part of Figure 4 on a perfboard. The only difference from my original MOD-25 is that the fixed resistor is replaced by VR-1. If your MOD-25 has the fixed resistor, just remove it, and all will be well.

The S-meter *output* at the top of C2 and cathode of D2 is an *analog* DC signal that varies from about 0v to maybe +1v on the strongest of signals. The max might be less, but it doesn't matter because U1a amplifies the signal by a gain factor set by R2 and R3. The gain formula for the Fig-4 parts is:

$$\text{Gain} = 1 + (R3/R2) \quad \text{or} \quad 1 + 15000/2200 = \underline{6.8}$$

FIG-4: A/D SIGNAL CONDITIONER



You can increase R3 or decrease R2 for more gain, as needed. If R2 is increased to 3.3-k, the gain drops to **4.5**; 4.7-k drops the gain even more to **3.2** . A lot depends on the output of the S-meter circuit at max signal input. If a walkie talkie held near the scanner results in an output at the top of VR1 of 0.6v, then to get the full 5v limit out of U1a, you'd need a gain of ($^{5}/_{0.6}$) = **8.3** in which case R2 should be about 1.8-k.

Most MOD-25's will do pretty close to +1v output with strong signals in, so a value of 2.2-k to 4.7-k for R2 will suffice in most cases. See "Calibration" ahead.

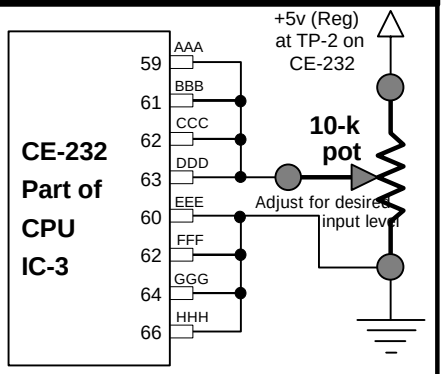
The purpose of optional 5.1v zener diodes, D3 and D4, is to protect the HB/CE-232's A/D inputs. Zener diodes do absolutely nothing until their rating is reached (5.1v) and then they conduct to keep any overvoltages clamped to that rating. D3-D4 aren't mandatory, but they're cheap insurance for IC3 on the HB/CE-232. A smart move is to put one at the input of each A/D channel that you plan to use.

Just in case you don't want to mess with the S-Meter and Center-Tune Meter functions with the A/D right now, but still want to test the A/D capabilities, Figure 5 shows an easy test circuit to whip up. Used with **ProTurbo**, you can see how A/D works, and perhaps whet your appetite for more.

Figure 5 (*next page*) shows how to wire four of the A/D channels for an adjustable input voltage to *simulate* analog signals. The other four A/D channels are grounded for a constant 0v low input. **ProTurbo** processes all 8 ch, displays, and *AutoLogs* the results.

Figure 5 is just a test circuit, but it will serve to safely acquaint you with the power of A/D in the HB/CE-232. Frankly, I don't know how I ever got along without mine, but I never knew how to implement it, not being a programmer. All I know of programming is that it comes on radio and TV. ☺

FIG-5: CE-232 A/D TEST CIRCUIT



Notes: This test circuit applies adjustble 0 to +5v to Pins 59, 61, 63, and 65 and a 0v ground to Pins 60, 62, 64, and 66. It makes a good test circuit for all 8 A/D channels and how the data is displayed, reported and AutoLogged by Pro-Turbo.

BUILDING THE HB/CE-232 A/D SIGNAL CONDITIONERS

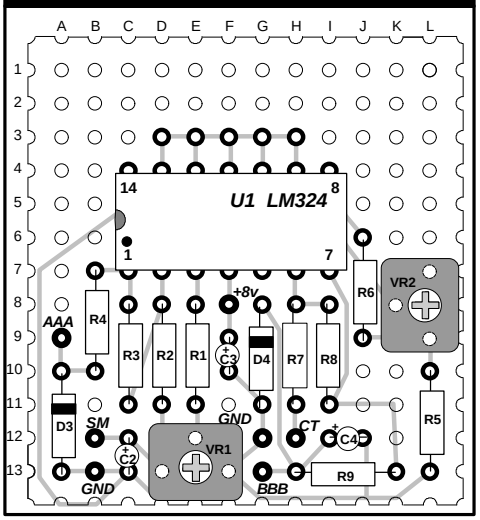
Figures 6 and 7 are tested examples of a perfboard layout and wiring plan for the A/D signal conditioners shown in Figure 4.

This should make it easier for couch potatoes and non-techies to make the A/D signal conditioners for the S-Meter and Center-Tune Meters discussed herein.

Notes On Figures 6-7

1. Figures 6 and 7 are proved-up layouts, but you might find better ways. Check the schematic (Fig-4) periodically as you go - it is accurate. Report problems to me ASAP.
2. Fig-6 shows the component side layout. The backside wiring is shown in light gray. Active (**used**) perfboard holes are shown in bolder circles than **unused** ones. Input and Output (I/O) holes are shown in circles darker than the component holes. I/O is also labeled for clarity.
3. Fig-7 is a view of the wiring side layout. It is the same view as Fig-6 except that it is flipped over, left to right. Components are shown in the background with emphasis on the wiring in black. The component and I/O holes are as described in (2) above.
4. There is a lot of unused space on the board as shown. You can make your board smaller, but since there are two unused channels on the LM324 chip, U1, some extra space might be needed later for necessary additional parts, if and when you want to use those channels.
5. If there is ever a chance you will later add to this circuit, I suggest you start with a slightly larger perfboard than shown in Figs 6-7, perhaps 14 or even 15 holes long (instead of 13), and 13 or 14 holes wide (instead of 12). Use the same pattern as shown in Figs 6-7, allowing an extra column on each side and an extra row, top & bottom.
6. VR1 and VR2 are the most difficult parts to locate and fit on the board. Radio Shack's trim pots, #271-281 (4.7-k) and #271-283 (10-k) will probably fit and work fine.
7. Figs 6-7 don't account for my MOD-25. It is assumed that you already have MOD-25 built in the scanner. To use it, remove any fixed load resistor that may be present at the cathode of

FIG-6: A/D COMPONENT SIDE



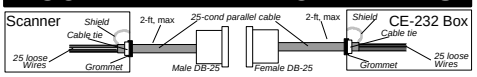
D2 to ground. (The 0.01-mF capacitors should be left on the board.) Route a wire from the cathode of D2 on the MOD-25 board to the "SM" point at C2 on the A/D conditioner board.

A/D INPUT BOARD PARTS LIST

```
C1,2    Used on MOD-25 board (0.01-uF)
D1-2    Used on MOD-25 board (1N34A)
D3-4    1N4733A; 5.1v zener diode;
RS#276-565
C3-5    10-uF/16v tantalum; RS# 272-1436
            (Tantalums are best &
smallest)
R1,2    2.2-k RS# 271-1325
R4-6,9  4.7-k RS# 271-1330
R3,8    15-k; RS# 271-1337
R7      22-k; RS# 271-1339
R9      4.7-k; RS#271-1330
U1      LM324 quad op-amp; RS# 276-1711
VR1     47-k trimmer; RS# 271-283
          (if 50-k not available)
VR2     4.7-k trimmer; RS# 271-281
Perfbd RS# 276-1395, 276-1394, or 276-
1396
```

8. Mount and install the A/D conditioner board in or near the HB/CE-232, if it is external to the scanner. You might not have spare wires in the cable that goes from the scanner to the HB/CE-232 to carry the S-Meter output signal. See Fig 7a.

FIGURE 7A: NEED MORE WIRES?



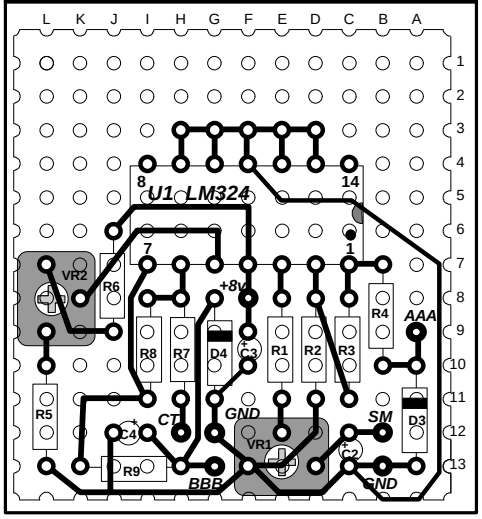
Use other spare wires for the +8v and the Center-Tune (CT) meter function. If you are out of spares, get a 25-cond parallel cable with a male DB-25 on one end and a female DB-25 on the other end. Cut the cable in the middle and install it as shown below. *Cheap and easy 25 additional wires*!

There are plenty of uses for more wires. As it is, the *scanner-to-HB/CE-232* connection requires a minimum of 19 or 20 wires, leaving only 5 or 6 spares in a 25-cond cable. These spares are easily used up with the **User Switches** and **Status Inputs**, and yet there are still eight **OutBytes** and eight **InBytes** begging for

attention. And then we have the eight **A/D channels**, so a new wire bundle isn't a luxury.

Even you experts haven't even come close to the full potential of the HB/CE-232 yet, not even counting the new A/D function. So, give yourself plenty of new wiring connectivity if you know what's good for you.

FIG-7: A/D WIRING SIDE



9. Sections C and D (U1c and U1d) of the LM324 aren't used. It is good practice to ground those inputs so that noise and oscillations don't cause problems. See Figs 4 and 6-7. Later, if you see a use for these two channels, you can remove the grounds and add whatever components needed to complete the circuit.

Calibration of VR1 and VR2

The purpose of VR1 and VR2 is to adjust the inputs to U1a and U1b so that the outputs can't exceed 5.0v. To set VR1, key a transmitter close to the scanner so that you know no stronger signals will ever be encountered. Measure the output of U1a and set VR1 so that this output is 5.0v, max. *Don't connect this circuit to the HB/CE-232 until VR1/2 have been properly adjusted.*

VR2 is similarly calibrated, with the output of U1b connected to a voltmeter (*don't connect to the HB/CE-232 yet*!), but it does require some explanation first.

The Center-Tune-Meter function is similar to the S-meter except that it sources from the scanner's NFM discriminator output, IC-2, Pin 9 for all three PRO-2004/5/6 scanners. (PRO-2035/2042, too!)

The U1b circuit processes the NFM discriminator signal, which is proportional to any error in the incoming frequency. When this signal is exactly 3.80 volts, the incoming frequency is right on with the frequency in the LCD display. Therefore, it is important that the scanner be properly aligned and tuned for this to work properly. For details, please refer to my *Scanner Modification Handbook, Vol-2*, (MOD-27) pages 123-135. For a quick and dirty calibration of this signal, tune the scanner to a known accurate transmitter, like a NOAA weather station.

Attach an accurate digital voltmeter, black to ground and red to IC-2, Pin-9 (PRO-2004/5/6) and read the voltage. If not exactly 3.80v DC, then adjust (***T8** PRO-2005/6*) or (***T13** PRO-2004*) for exactly 3.80v. These adjustments are sensitive, so you may have to "rock" them in by tiny sequential adjustments.

Now you can test off-frequency signals. Say you have a NOAA weather station on 162.400 MHz. Program the scanner with 162.390, 162.395, 162.400, 162.405, and 162.410 in that order. As you MANual step through these channels, you'll see the Center-Tune Meter function at it's best. And since you know what the voltages are for ±5 kHz and ±10 kHz errors, you can interpolate across this range to determine other errors,. You'll get the hang of it. Here are the results from one of my PRO-2004's:

TABLE 2
162.400 MHz CENTER TUNE TESTS

TUNED FREQ	VOLTS AT IC2 PIN 9	Offset kHz
162.385	ambiguous	-15
162.390	6.06	-10
162.395	5.03	- 5
162.400	3.80	0
162.405	2.48	+ 5
162.410	1.39	+10
162.415	ambiguous	+15

The Center-Tune function of the PRO-2004/5/6 scanners is great for up to ±10 kHz errors. The IF selectivity curve is set by a ceramic IF filter that renders meaningless any measurements made at ±15 kHz and beyond in the NFM and AM modes. I tried to measure the voltage at IC2-Pin 9 at the ±15 kHz offsets, but there was no way a stable reading could be taken. So ±10 kHz is the limit for accurate offsets or transmitter frequency errors on NFM and AM.

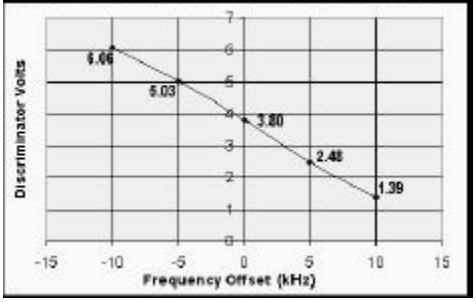
When my data set was plotted into a graph (see Figs 8-9), the ±10 kHz offsets (and points between) certainly are meaningful and the data-set is linear.

This means a simple **Y = mX + b** formula can convert the raw data into meaningful frequency errors. **m** is the slope; **b** is the Y axis intercept. This can be done in a spreadsheet or perhaps in the **Pro-Turbo** program if you can talk the author into supporting it! For now we'll interpolate.

From Figure 8, we see at least two other bits of meaningful information, the first of which is that IC2-Pin 9 definitely rises above 5 volts, actually over 6v. We dare not input this into the HB/CE-232's A/D so we have to do something to limit it.

Secondly, we see a negative slope. As the frequency offset goes above center, the IC2-Pin 9 voltage decreases; kind of inverse to what you'd think. So we use U1b to limit the output to 5v max, and at the same time invert the function to a logical slope where the voltage rises as the offset rises.

FIG-8: CENTER-TUNE FUNCTION



Connect the "**CT**" point on the A/D Signal Conditioner board to IC-2/Pin-9 in the scanner and have a voltmeter connected to Point BBB. Repeat the above test for off-frequency signals and take your own data like I did. You'll get similar results.

Now, with the scanner tuned 10 kHz ***above*** a known accurate signal, the output of U1b will be maximum. Adjust VR2 so that this maximum comes in under +5v. I set mine for 4.6v.

Next, tune the scanner back to center frequency and note the output of U1b. Now tweak VR2 again so that center freq produces a meaningful voltage. I trimmed mine to 3.0v - easy to remember for center frequency, you see! You can pick another reference center, if you like, but for safety of IC3, don't go above 3.0v.

TABLE 3
162.400 MHz CENTER A/D TESTS

TUNED FREQ	VOLTS AT A/D (BBB)	Offset kHz
162.390	1.55	-10
162.395	2.20	- 5
162.400	3.00	0
162.405	3.85	+ 5
162.410	4.60	+10

Fig-9 compares the response of the Center-Tune functions, one from the Discriminator at IC-2/Pin-9 and the other as fed into the HB/CE-232's A/D after "conditioning" by U1b. You can see where the 5v limit is not threatened and where the offset produces a logical "rising" slope.

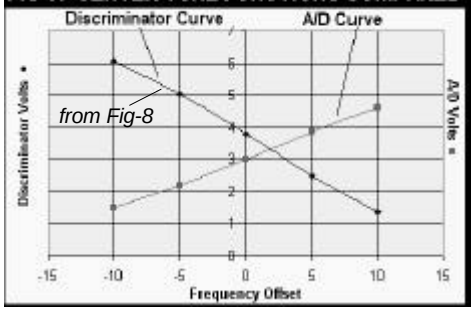
INTERPOLATING THE DATA

Easy. Let's suppose the A/D reports 3.15v at BBB with a certain signal. We use simple linear interpolation to determine the offset:

$$\frac{(5000)(3.15 - 3.00)}{(3.85 - 3.00)} = +882 \text{ Hz}$$

....or, that transmitter is about 1-kHz high of center frequency. Rather than take a page to explain what I just did, you can figure it out since all the numbers are plainly visible. In a few words, $^{0.15}/_{0.85}$ is the voltage offset, and when multiplied by the frequency increment (5 kHz), equals the probable freq offset. Next is an example for 2.6v.

FIG-9: CENTER-TUNE FUNCTIONS COMPARED



$$\frac{(-5000)(2.60 - 2.20)}{(3.00 - 2.20)} = -2500 \text{ Hz}$$

This transmitter is 2.5-kHz low of center!

One neat thing about *AutoLoggable* Center-Tune and S-Meter data is the "fingerprinting" power. You can I.D. transmitters by signal strength and by how much they're off freq; very useful when a number of transmitters are on the same assignment. *AutoLogged* S-meter and Center-Tune data can help you know at a glance, who is who and who has been using what frequencies! Too cool!

For example, you start an *AutoLog* session and go to bed. Next morning, you find a number of hits on 158.970 MHz. Are they all the same transmitter, or a variety? *You **don't know** until you check the S-Meter and Center-Tune Data*!

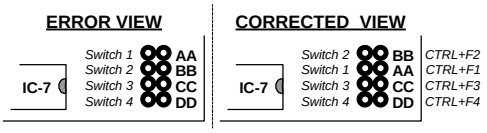
MORE HOT STUFF

Not exactly new on the HB/CE-232, but long overlooked by the user community are four features that call for ingenuity and creativity in order to implement and use.

These four features are best described as *user-controllable INPUT and OUTPUT ports*. Let's start with the best known of the four that we call the ***User Switches***.

IC7 on the HB/CE-232's board is just like IC4-6, a 74HC4066 CMOS quad bilateral switch. IC-7 doesn't do anything until *YOU* tell it what to do and when to do it. Basically, IC7 is just a set of four independent SPST toggle switches. You can use them to turn external circuits on and off, or to control most anything that can be conjured up by your imagination *(25-mA max)*

The four ***User Switches*** can be controlled from the keyboard when the HB/CE-232 is in operation by *holding down* the **CONTROL** key and tapping **F1** thru **F4** (corresponding to Switches 1-4). Due to a documentation error, the physical location of these switch outputs is shown again as follows:



You can operate any ***User Switch*** separately and independently from the rest. Any one, two, three, or all four switches can be on, off, or in any combination.

The ***User Switches*** can also be controlled from a ***Script*** with the command:

OUTSW [=] nnnn *(The "=" sign is optional)*

*(where **nnnn** is a 4-bit binary number, 0000-1111, and where "X" can also be positioned.)* Examples:

```
OUTSW 1101  (SW4, 3, & 1 are on and SW2 is off.)
OUTSW 0001  (SW4, 3, & 2 are off and SW1 is on.)
OUTSW 0XX1  (SW4 off; SW3-2 left alone; SW1 is on.)
```

That's right, if you put an "X" in one of the positions, the ***Script*** will leave that switch alone in whatever position it was in, on or off, before the ***Script*** was run!

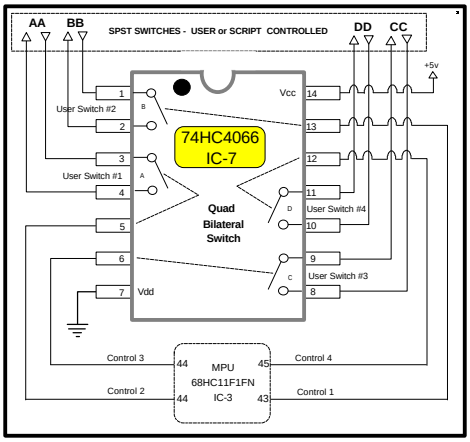
There is a visual indicator of the ***User Switch*** status in the upper-left corner of the DOS control program screen. It's labeled "*Output Status*" and can't be missed.

A schematic diagram of IC7 and how it feeds Points AA-DD is shown ahead.

Okay, so now you know everything there is to know about the ***User Switches***. So make plans to use them. And once you do, a measly four of 'em might not be enough. So how's about eight more of the little buggars? If that idea appeals to you, see next page for the discussion of the ***OutByte*** feature, though the ***OutBytes*** are good for a lot more!

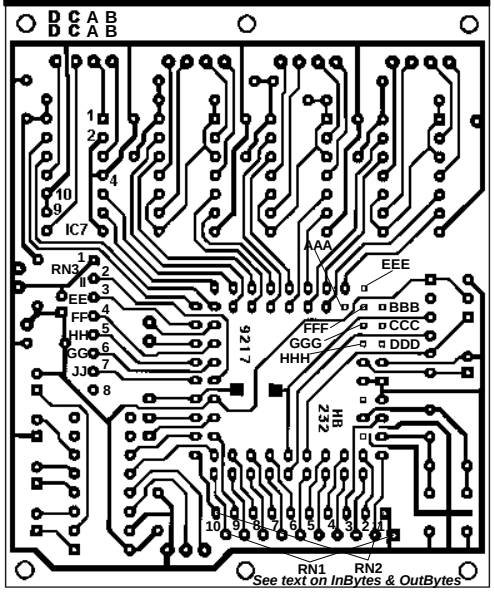
GET ACQUAINTED WITH THE STATUS INPUTS

The HB/CE-232 is the only scanner interface around with 4 software *switches* that can be controlled by computer keys ***or*** the powerful automated Script function. But there's a lot more than mere ***User Switches*** to make the HB/CE-232 unique among scanner interfaces! *Ohhhhh, so much, much more….*

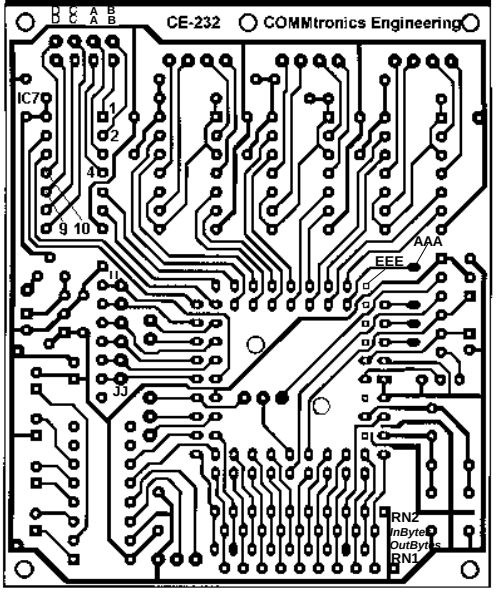


Another of the HB/CE-232's claims to fame is the 6-bits of **read-only** ports that we call the ***Status Inputs***. In a word, when we designed the HB/CE-232, we couldn't figure out what to do with the spare input ports of the powerful microprocessor that runs the interface. So we allocated six of

HB-232 BOARD



CE-232 BOARD



those logic input ports to *status indicators* that can be used for any +5v logic display purpose.

Confused? Think of these six ports as "idiot lights" in a car, except that they can be used for any purpose that you might imagine, so long as that purpose has a 5-volt logical "high" and a 0-volt logical low.

The visible location of the ***Status Inputs*** is in the upper right corner of the DOS control program for the HB/CE-232. It's labeled "*Input Status*" and has six zeros just under the label. These digits will always read "0" until you use them.

The ***Status Input*** ports are available as Points EE, FF, GG, HH, II, and JJ on the CE/HB-232 board. Physically, these points are at Pins 2-7 of resistor network, RN3. On the CE-232 board, these six points are padded, drilled, and made available on the top side of the board. The HB-232 boards do not have these points padded, drilled, or conveniently available, but you can access them from the bottom side of the board at Pins 2-7 of RN-3. A strip of six pin-line sockets is suggested as the best access.

To use the ***Status Inputs***, you need only apply a +5v logic signal directly to the one of six pins. You can use any one or any combination of these ports for a total of six individual and independent bits of status display. Caution must be exercised to not exceed about 5.1v or else the expensive CPU chip, IC-3, could be destroyed. Fortunately, +5v logic signals don't go higher than 5 volts.

Two immediate applications come to mind: **Squelch** and **Mute**. The +5v **Squelch** gate signal in the PRO-2005/6 can be found at Pin 5 of CN3 and in the PRO-2004 at Pin 8 of CN504. The logic of the **Squelch** gate at this point is 0v (low) when **Squelch** breaks and +5v when **Squelch** is set. If you were to run a wire from the **Squelch** gate to, say Point JJ on the Interface board, then the left-most digit of the ***Status Inputs*** will indicate a "0" when **Squelch** breaks and a "1" when **Squelch** is set.

If you ran a wire from the **MUTE** gate, CN3, Pin 12 in the PRO-2005/6 and CN504, Pin 8 in the PRO-2004, to Point GG on the Interface board, then when audio is present, the next-to-left-most ***Status Input*** bit will show a "1", and a "0" when there is no audio. You could also tie the **SOUND SQUELCH** gate to one of the ***Status Inputs***, if you wanted, as well as any other 5v logic signal.

One of the more important aspects of the ***Status Inputs*** is not the visual indicator on the screen; rather the AutoLog option for these ports! That's right, the ***AutoLogger*** will log these bits every time there is a "signal hit".

It's possible to connect an S-Meter to 4-bits of the ***Status Inputs*** for a binary log of 0-10. For that circuit and relevant details, I will refer you to back issues of the ***World Scanner Report***, Vol-3 No-6 & Vol-3, No-8 for the gory details. An AutoLoggable S-meter function is quite a power-house!

The ***Status Inputs*** can be used in logical operations by the ***Script*** feature, too. The script name for the ***Status Inputs*** is **INSW** and can be used with **IF** : **THEN** : **ELSE** operations, as well as stored in a **VALUE@X**. I'll refer you to the ***Script*** documentation in the main CE/HB-232 manual for the details. But in a word, you can write a simple script to test for the values in the ***Status Inputs*** and then make a logical **IF** : **THEN** : **ELSE** decision.

In summary, the ***Status Inputs*** allow the CE/HB-232 owner to connect up to six logical inputs for *visual*, *loggable*, and *scriptable* scanning. Value is apparent when the operator is away from the shack and returns to find meaningful data awaiting!

GET ACQUAINTED WITH THE INBYTES

The ***InBytes*** are exactly like the ***Status Inputs*** in all respects, except that there is no visual indication of them in the main operating screen of the DOS control program. The ***InBytes*** can be *AutoLogged* and operated on within a ***Script*** and you can connect 5-volt logic signals to them for monitoring via the *AutoLogge*r, or from within a ***Script***. The script name is **INBYTE**.

The eight ***InByte*** ports are physically located at RN1, pins 7-10 and RN2, Pins 6-8 and 10. Point labels are KK-RR as follows:

RN2, Pin 10, Point KK = InByte 0
RN1, Pin 10, Point LL = InByte 1
RN2, Pin 8, Point MM = InByte 2
RN1, Pin 9, Point NN = InByte 3
RN2, Pin 7, Point OO = InByte 4
RN1, Pin 8, Point PP = InByte 5
RN2, Pin 6, Point QQ = InByte 6
RN1, Pin 7, Point RR = InByte 7

CE-232 board have the ***InBytes*** padded, drilled, and accessible from the top side of the board where pin-line sockets will fit nicely. Older HB-232 boards do not have these points padded or drilled, so access has to be from the bottom of the board, directly to the pins of resistor networks, RN1-2, as shown above. ***Do not under any circumstances connect anything to RN2, Pins 2 & 9***.

Concluding, there are 14 bits or ports for +5v logic-level input that can be applied by the user. Six of these bits are visible on the main operating screen and are called the ***Status Inputs*** while the other eight called the ***InByte***s, are not visible, but all are *AutoLoggable* and *scriptable*. These 14 bits are limited only by your imagination!

GET ACQUAINTED WITH THE OUTBYTES

The eight ***OutBytes*** are similar to the ***InBytes*** except that they're user-controllable logic *outputs*! Less useful than the ***User Switches***, the ***OutBytes*** are not switches and cannot be controlled from the keyboard.

The ***OutBytes*** are controllable only from the *Script* feature where the name is **OUTBYTE**. Like the ***User Switches***, control is done as:

OUTBYTE [=] nnnnnnnn *("=" sign is optional)*

(where **nnnnnnnn** *is an 8-bit binary number, 00000000-11111111, and where "X" can also be positioned as a bit.)* See examples next:

OUTSW 00001101 *(OB7-4, & 2 are low and OB3, 2 and 0 are on off.)*
OUTSW 11110001 *(OB7-4 and 0 are on and OB3-1 are off.)*
OUTSW 1XX10XX1 *(OB7, 4, and 0 are on; OB6,5,2 and 1 are left alone and OB3 is off.)*

Like the ***User Switches***, an "X" can be used in the binary string to leave the settings of the ***OutBytes*** unchanged. (X = "don't care")

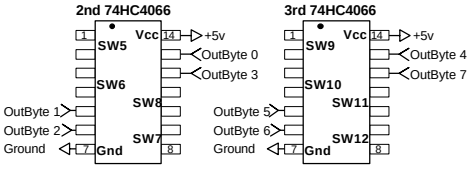
I said the ***OutBytes*** weren't switches, and they're not, but in certain cases, they can be used as switches. One of those cases is my **Extended Memory Mods** where the Block addressing is switched by grounding or raising high the address pins. If you have a 6400-ch or 25,600-ch Extended Memory mod, you can do away with the toggle or DIP switches and use the ***OutBytes*** to control the Blocks instead. You can write a ***Script*** to run any combination of Blocks, including sequential, if you like.

The eight ***OutByte*** ports are physically located at RN1, pins 2-6 and RN2, Pins 2, 4, and 5. Point labels are SS-ZZ, as follows:

RN1, Pin 2, Point SS = OutByte 0
RN1, Pin 3, Point TT = OutByte 1
RN2, Pin 2, Point UU = OutByte 2
RN1, Pin 4, Point VV = OutByte 3
RN2, Pin 4, Point WW = OutByte 4
RN1, Pin 5, Point XX = OutByte 5
RN2, Pin 5, Point YY = OutByte 6
RN1, Pin 6, Point ZZ = OutByte 7

OUTBYTES AS SWITCHES ?

Yes, the ***OutBytes*** can perform switch *functions* whereever 5v logic highs and lows can suffice. You can't use them as physical switches, however. You can use the ***OutBytes*** to control a second (or a third) set of ***User Switches***, if you need more than the four stock ones. Just use the ***OutBytes*** as control signals for one or two more 4066 CMOS switches!



In conclusion, there are 8 bits or ports for +5v logic-level output that can be applied by the user. These ***OutBytes*** are not visible on the screen and can't be keyboard controlled like the ***User Switches***, but they are controllable by a ***Script*** and their status or positions can be *AutoLogged*!

OTHER HB/CE-232 FEATURES

Hey, **all** but the A/D are in the Manual! But in the offhand chance you'll often refer to this newsgram, I'll include an overview of the rest of the HB/CE-232 features, just so you have a quick reference. Just don't be afraid to read the Manual.

AutoLogger: User and script-controllable, logs a record of all pertinent scanner activity and settings to a plain ASCII text file with the *.LOG extension. This AutoLog file is comma-delimited to make it easy to drop the file into a spreadsheet or database manager for some monstrous data processing power. An AutoLog file is closed and a new one created every hour on the hour, when it's active. Obviously, 24 files a day can be generated and if that's cumbersome, you can copy them sequentially into a single file. The file format of the AutoLogger is explained in the Manual and will not be treated here.

AutoProgrammer: Programs the scanner from a plain ASCII comma-delimited text file, very similar to AutoLogger files. In fact, an AutoLog file can be renamed and used as an *.APF file, if needed. The key thing about the AutoProgrammer is the required format, which is explained in the Manual and will only be summarized here.

- Six fields, each separated by a comma are required; but everything after the 6th comma is ignored by the AutoProgrammer.
- APF file format is as follows:

Filte	Chan	Freq	Mode	Delay	LockOu	etc…

If a **Filter** character or code isn't used, then each line of an APF file *must begin with a comma*. A typical line in an APF file w-w/o **Filter** looks like:

Police,110,158.970,nfm,D,L,*a description can go here*

,110,158.970,nfm,D,L,*a description can go here*

The AutoProgrammer would select **ch-110**, and program **158.9700** in **nfm** mode, set the **Delay** and then **LockOut** the channel. The **Police** filter is ignored unless selected at the <F4> menu.

- 255-chars, max, per line, for *.APF files.
- No real limit to the number of lines in an APF.
- Don't use commas in the description fields; instead use semicolons, colons, or dashes.
- You can have as many filters as you want, but their length is limited to 12-chars (no commas)
- Filters are useful for large APF files. Suppose your APF file had 2400 lines; this could pose a problem for 400-ch PRO-2004/5/6's, right? But suppose 400 had a "**Police**" filter, 400 more had a "**Fire**" filter; and 400 each for "**County**", "**City**", "**FedGov**", and "**Aero**". The AutoProgram function allows a filter to be specified for each session, so you can keep all your freq records in one file, if you like, and program by filter!.
- I recommend the database manager in ***Microsoft Works For Windows*** as a great tool for preparing and maintaining APF files. ***Works*** imports and exports comma-delimited ASCII files with ease, and it's very low cost. For more info, check out:

www.microsoft.com/products/prodref/568_ov.htm

Or call your favorite computer store.

- Reid Drummond's **APFTools** program is an awesome tool for making APF files out of raw lists of frequencies…..like "magic", no less!

AutoReject: Any frequency placed in a *.REJ file will be automatically rejected during SCAN and SEARCH operations, when the AutoReject function is selected and active. An AutoReject file is a plain ASCII text file that consists strictly of a column of frequencies without spaces, punctuation, or other characters other than the numbers and a decimal, one frequency per line. Up to 32,000 frequencies can be in one *.REJ file. Performance can be improved if the reject list is sorted ascending, and if "four-dot-four" format is used. Leading and trailing zeros are okay. A typical AutoReject list will contain all your scanner's "birdies" as well as all the non-voice signals that your scanner encounters. You can use a Script (**BIRDFIND.SCR**) supplied with your Kit to AutoLog all the birdies so you can drop them into an AutoReject file.

AutoLookUp: The AutoLookUp function can search and display from a file of your choice, the data that exists for each frequency the scanner stops on. Typically, the AutoLookUp file will be the same APF file that was used to program the scanner, and since a description can be used in the filter field as well as after the 6th comma, it follows that you can see the data behind the "hit" almost instantly after the scanner stops!

Scripts: The HB/CE-232 Interface comes with a very simple and elementary programming language that we call the Script Feature. It may be simple, but oh is it powerful. There are about 40 script *commands* and maybe that many again *operators* and *conditions*, that can be used to automate the scanning process, and by "automate", I mean to take away the drudgery without affecting the fun! Just consider how much fun it would be to manually find and log all the "birdies" in your scanner – a hundred or more. The **BIRDFIND.SCR** script can do it for you while you're out sipping suds and munching pizza.

Much of the power of the Script is in doing things that you wouldn't think of doing manually. More of that power is underscored by the data acquisition and processing power it offers. More still comes from the logical decision-making power behind the **IF** : **THEN** : **ELSE** : **AND** : **OR** logic that can be woven into a script.

If you are not a programmer (as I am ***not***), relax! With less than a hundred "words" to learn, you can have your first script up and running within minutes. Inside a week or two, you can become an expert script writer. The Manual is pretty good at the introductory level, and coupled with the few dozen sample scripts provided on Disk with your Kit, there is no reason not to dig in and get your feet wet.

While I do not directly support the Script feature I can suggest it's easy to get started. Just use any text editor, like NOTEPAD.EXE, WORDPAD.EXE, etc, and open a text file for editing. The first line is easy: it should begin with the name of the script, preceded by a "rem" symbol, the "/" forward slash and a space, like this: / **BIRDFILE.SCR** The second line is even easier: **KEYCLR** Here is an example:

```
/ TEST.SCR
KEYCLR
Chan = 17
Freq = 158.970
man
end
```

This script selects channel 17 and programs it with 158.970 MHz, and then returns to the MANual mode. Not bad, eh? I'll leave the rest up to you, but there's no reason not to get started cranking out some scripts of your own!

Here's another line of thought for your script possibilities. Suppose you have a bank of fire freqs with all locked out except the dispatch. You could have a script that tests for IF the dispatch freq gets active, then unlock the rest of the batch and scan only those to the exclusion of the rest!

In other words, you can write scripts to control your scanner the same way YOU would control it manually, complete with logical analyses and decision-making. Scripts can change banks; lockout some freqs and unlock others, turn things on and off; and virtually anything you can do manually, except the script does it faster and more accurately. Possibilities are limitless. So explore!

OPERATING SCREENS

There are four operating screens in the DOS program for the HB/CE-232? You can view them by pressing the **right-arrow** cursor key. The most important could be "Screen 2", shown next.

Screen 1 is the main operating screen, while Screen 3 is for ***AutoLookUp*** and Screen 4 is for ***Scripts***. Screens 3 & 4 auto-appear at times, but Screen 2 has to be selected or you'll miss it. It shows ***AutoLog*** data and the *active files* for the session. Check it out. Just tap your **right-arrow** key a few times when the program is up and running.

OTHER SCANNER SUPPORT ?

The HB/CE-232 can support only the legendary PRO-2004/5/6 scanners with the full feature set that I call "*two-way mode*". F irst, let me describe those "*two-ways*" as **READ** and **CONTROL**.

Control consists of four functions: the ***AutoProgrammer***; the ability to ***remote control*** the scanner from the keyboard of the PC; the four ***User Switches***; and to a limited extent, the ***Script***.

Read is *data acquisition*, a mode that makes all the other features possible, including ***AutoLog***, ***AutoLookUp***; ***AutoReject***; ***Status Inputs***, ***InBytes***; ***OutBytes*** and most of the ***Script*** feature.

The **Control** mode uses Points O-Z, and AA-DD, while the **Read** mode uses Points I-M. Once you understand this, then you may see where *only* the PRO-2004/5/6 can be connected to all these points, and therefore gain the *full* feature set.

For the technically inclined, Points I, J, K, and L of the HB/CE-232 go to key points in the scanner where data flows between the CPU and the LCD Display. The HB/CE-232 taps that data and **reads** it for the program and the PC. Only the PRO-2004/5/6 scanners offer these key pickoff points and scanner data in the right format that allows this **Read** mode.

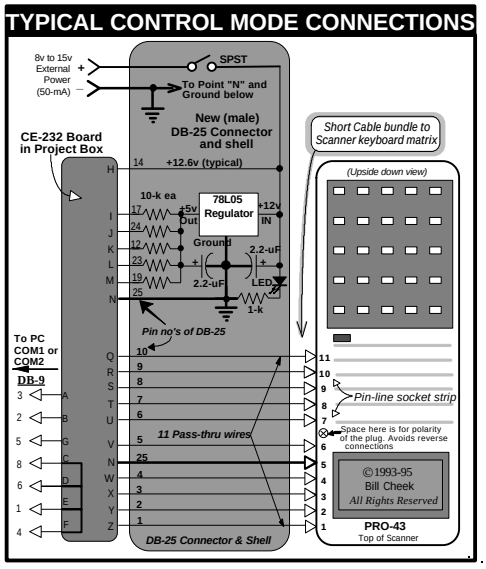
As a sidenote, however, I have determined that three other scanners have the *same* display and data path circuitry as the PRO-2004/5/6, but I have not experimented with them: BC-235XLT, BC-2500XLT, and the PRO-90. The common thread for all six scanners is a standalone LCD Display driver chip, the uPD7225. The thing is, these latter three scanners are made by Uniden, while the PRO-2004/5/6 are made by GRE-Japan, so it is not likely that the ***format*** of the data that flows between the scanner CPU and the uPD7225 Display driver chip is compatible with the HB/CE-232's requirements. (*like the PRO-2004/5/6*). I probably won't ever experiment with these latter three scanners, but it could be an interesting project for one of you adventuresome techie types….and if you want to try, I'll support your efforts to a limited extent by e-mail.

It is possible, however, to connect the HB/CE-232 to *many* other scanners with the **Control** mode only. *All scanners* have a keyboard matrix, which is where Points O-Z are terminated. The following scanners are known to work with the HB/CE-232 in **Control** mode only:

PRO-2042	**PRO-2022**	**PRO-26**	**PRO-39**
PRO-2035	**PRO-2021**	**PRO-34**	**PRO-43**

I am sure that many more can be added to this list, but I haven't actually done the work, you see. There are several issues associated with making the HB/CE-232 **Control** other scanners:

1. Points O-Z have to be identified and located in the scanner.
2. Not all Points O-Z have to be used, but since O-V are matrix columns and W-Z are rows, then we are limited to those scanners with 8 or fewer keyboard matrix columns and 4 or fewer rows. In other words, the HB/CE-232 probably can work with scanners that have a keyboard matrix of 7 x 4, 8 x 3, or any combination of 8 x 4 or less.
3. The foregoing being satisfied, you will need a 'personality file' (*.PER) for the scanner. You can always *try one* that comes with your Kit on disk, but if it doesn't work, I will have to make one for you. The cost is $10, forwarded by e-mail, or $15, ppd, on floppy disk.
4. You need a means of connecting the HB/CE-232 to that scanner. A row of pin-line sockets on the side of a handheld scanner, or a DB-25 connector on the rear of a base scanner, and the Interface in an external metal box is ideal. Either one is internally wired to the matrix. The HB/CE can be used with many scanners, that way, see?
5. Since Points I-M can't be used in **Control** mode only applications, you must devise a way to feed +5v via a 10-kΩ resistor to each of these five points on the HB/CE-232 pcboard ***when*** the Interface is connected to that scanner. We have detailed Instruction Sets for some of the above scanners ($14.95, ppd), that might suffice for non-listed scanners. Our method of feeding +5v to Points I-M is 'universal' and could be just fine for you. See below for a PRO-43 example, but all are similar, matrix lines being the main variable:

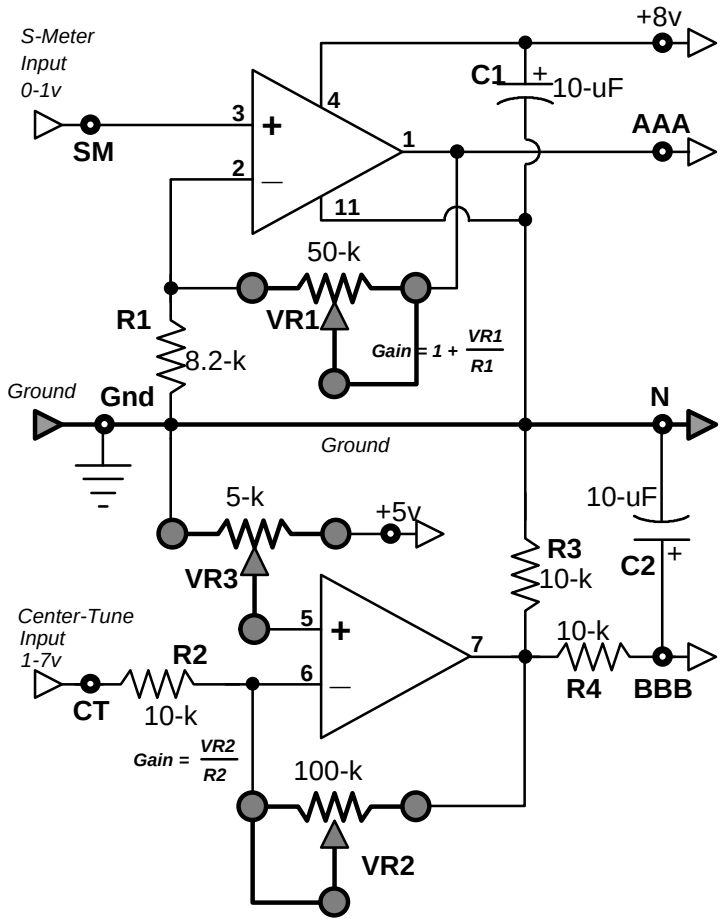


6. In the above, the HB/CE-232 works normally with your PRO-2004/5/6. For other scanners, disconnect the parallel cable between PRO-2004/5/6 and Interface, and connect the special DB-25 connector to the Interface and connect the short matrix wire bundle to the scanner. Everything is contingent on your use of our standard wiring plan for external Interfaces with the PRO-2004/5/6 as shown in the Manual.

WRAP-UP

The HB/CE-232 is ***not*** being phased out, but the PRO-2004/5/6 scanners have long been off the market now, and interest in the CE-232 is waning. We can't afford to continue inventory and new research for it.

If you want more, it's going to be up to YOU *and* your fellow Users of the HB/CE-232 to form a "**user group**" that will represent a voice and buying power. I will remain supportive and helpful for as long as I can, but the rest is up to you as a community. ***So what now?***


S-Meter
Input
0-1v
SM
3
2
4
1
11
+8v
C1
10-uF
AAA
50-k
R1
8.2-k
VR1
Gain = 1 + VR1/R1
Ground
Gnd
Ground
N
5-k
+5v
VR3
10-uF
C2
R3
10-k
Center-Tune
Input
1-7v
CT
R2
10-k
5
6
7
10-k
R4
BBB
Gain = VR2/R2
100-k
VR2