Price: $29.95

THE HB-232 SCANNER INTERFACE UNIT
~
- DEVELOPER INFORMATION -

Brought to you by

Bill Cheek
COMMtronics Engineering
PO Box 262478
San Diego, CA 92196-2478 USA
BBS & FAX: (619) 578-9247 6pm-1pm, PST
FidoNet Address: 1:202/731
RadioNet Address: 500:10/12
GEnie Address: WCHEEKSR.1

Date

November 20, 1992

The HB-232 Scanner Interface Unit - Developer Information

- TABLE OF CONTENTS -

~ FORWARD ~

Recognizing the unqualified success of the "open architecture" IBM/PC and compatible personal computers, we at ***COMMtronics Engineering*** are striving to emulate the mechanics of that awesome success. Like the IBM-PC, our **HB-232 Scanner/Computer Interface** represents, not so much a the first of its kind, but clearly a major break-through in automated and computer-controlled hobby radio! The HB-232 is the first to offer a wide spectrum of features and benefits, including automated programming of 400-memory channels at a time from a plain ASCII text file; automatic logging of "events" or transmissions to a plain ASCII file; intelligence to prevent the scanner from locking up on "birdies" and virtually all other undesirable signals; automated LookUp of frequency data from a specified database for scanner stops & pauses; AND a Script language from which all the above **and more** features can be implemented, **hands-off**! The HB-232 is the first of its kind to implement these powerful features **and more** without impacting the factory-specified performance of the scanner!

The HB-232 is a system of hardware, software and firmware, which when retrofitted to the Realistic PRO-2004, PRO-2005 and PRO-2006 scanners, offers an unparalleled measure of computer control and automation of virtually all factory-provided features and controls. Other amenities of the HB-232 take up where factory features leave off to allow the User virtually unlimited variations of operating conditions and control of after-market or retrofit enhancements. In a word, the HB-232, operated with most any IBM-PC/compatible computer, not only reads and displays whatever the scanner reads and displays, but also controls any function that can be manually controlled at the Keyboard of the scanner! Many other user-added functions can also be controlled and monitored by the HB-232! Perhaps little of this would be so remarkable except that a complete system, **including computer**, can easily cost less than $1000.00. Here is a real-life example: PRO-2006 @ $400; HB-232 Kit @ $170 and an XT/clone w/40-Mb hard drive @ $200, for a total of $770. Even with incidentals and hidden costs, the total investment is well under a grand! I know, because I just described my developmental system! Frankly, it is virtually impossible to accomplish what can be offered by the HB-232 with any other system for under $2000, and THAT might not include the cost of the computer! On top of that, other systems with comparable power will include a radio that can be termed only as "world class", which by definition, is in a very limited market! It is conservatively estimated that between 50,000 and 125,000 PRO─2004/5/6's have been sold since the Fall of 1986. It is not likely that the sum-total of all "world class" radios can approach a fraction of this, and we're talking about the likes of the ICOM R─7000, R-7100; R-9000, R-100; Yaesu FRG-9600; AOR's AR-3000 and AR-2515.

At this point, I have not so much extoled the virtues of the HB-232 as I have iterated a list of motivations for **3rd Party Developers** to take a long, close look at it and the potential market which it represents. You are cordially invited to review the enclosed documentation on both the software and the hardware with an eye to developing either a successor to the HB-232 Program or, at the very least, supporting software and utilities. You could find such a venture to be very worthwhile.

The ***HB-232 Scanner/Computer Interface*** has been undergoing Beta & Charley testing since early summer, 1992, and has now been released for sale to the public. The bugs have been worked out and resolved and a solid User Base is in place. The learning curve for documentation and technical support has been experienced and overtaken. Long-term advertising commitments have been made with "***POPULAR COMMUNICATIONS***" and "***MONITORING TIMES***" magazines. Extensive reviews have been made in "***MONITORING TIMES***" (October, 1992) and others are planned for well into 1993, in other rags. The HB-232 is quite the topic of conversation on computer BBS networks and information services, including GEnie, ComPuServe, Prodigy, America-On-Line and FidoNet. Its impact on the Hobby Radio Scene and Market seems solid and assured for a healthy lifespan. The recently passed FCC Authorization Act, which forbids the manufacture and importation of receivers that are capable of detecting cellular telephone signals and those receivers which can be easily modified for same, may very well indirectly stimulate a lengthened market life of the HB-232 by virtue of enhancing the perceived value of the PRO-2004/5/6 scanners. All in all, conditions are ripe for the success of the HB-232 and any supporting markets. As a developer and a radio enthusiast, no doubt you will be interested in exploring this opportunity.

It is our intent to make the HB-232 as much of an "open architecture" product as possible. Already, at least two 3rd Party-utilities have been developed for the HB-232, and we hear of more that may be on the way!: You are cordially invited to review the enclosed documentation and consider the opportunity to develop 3rd Party Software for the HB-232. If this manual does not answer all of your questions, or if it leaves anything up in the air, feel free to contact us for a more intimate and personal discussion. Initial contacts are best by mail, FAX or on our BBS, ***The Hertzian Intercept***, after 5:30pm and before 1:30pm. (619) 578-9247. It's voice only between 1:30pm-5:30pm, but I'm rarely available for cold-calls because of my arduous work schedule. Still, if you are a serious developer, and if you need support, there will be ways we can get together. We're on an adventure and welcome you to join.........if you dare.............

Bill Cheek .

THE HB232 SCANNER INTERFACE UNIT
- DEVELOPER INFORMATION -

-[The information contained in this document is provided for third party developers to assist in the preparation of programs that take advantage of the HB232 Scanner Interface Unit. This data is subject to change with out notice. We accept no liability for the impact of future changes. The developer assumes all risks associated with the use of this data. At this time it appears that any changes will be in the nature of additions to, rather than modification of, existing data; however, we will not guarantee this to be the case.]-

- 1.0 INTRODUCTION -

The HB232 Scanner Interface System has three parts: the scanner, the Interface Unit, and a program in a host computer. Opportunities for the developer are in the area of the host computer's program. This document provides information needed to access the services provided by the Interface Unit.

The HB232 Interface Unit consists of a MC68HC11F1 microcomputer, RS232 level converter, four quad bilateral switches, and miscellaneous support components. The Interface is connected to the target scanner in such a way as to be able to monitor the data stream between the scanner's CPU and LCD driver chips. The data (representing the state of the scanner) is reformatted and sent out when requested by the host computer. Input to the scanner is accomplished by activating (via commands from the host) the bilateral switches to simulate pressing the scanner's keys.

1.1 THE 'HC11 MICROCOMPUTER

The MC68HC11F1 is a single chip microcomputer made by Motorola and intended for embedded controller applications. The chip has many capabilities that are not used (yet) in the HB232 (such as an 8 channel A/D converter). On board the chip are 512 bytes of EEPROM and 1024 bytes of RAM. The chip is configured for the "bootstrap" mode of operation in the Interface Unit. This chip was chosen because of its bootstrap capability (avoids the need for burning the program into ROM) and it's relatively large internal memory (avoids the need for external memory). Changes to the system software are very easily accomplished. The user is provided with a utility (EEPUTER.EXE) which will install a portion of the program into the EEPROM. The balance of the program is downloaded into RAM upon start up.

1.2 THE INTERFACE UNIT'S PROGRAM

The program for the Interface Unit is in two parts. The EEPROM is programmed one time and contains the portion of the code that emulates the scanner's LCD driver as well as code to receive commands from the host computer. The RAM, when properly loaded, contains the routines which format the scanner's data for transmission to the host, timing values for the type of scanner being used, and a unique identification byte (SigByte).

- 2.0 FILES AND UTILITIES -

Each HB232 Interface Kit is shipped with various files. The developer may access (but not distribute) these files as required. A description of these files follows:

HB232Vxx.EXE	The main PC program.
EEPUTER.EXE	A utility for the user to install the core routines into EEPROM. Needs to be run at least once by user.
HBOOTVxx.EXE	A utility (hereafter referred to as HBoot) which initializes the Interface Unit. If required, it will download the second part of the Interface Unit's program into the 'HC11 RAM. The main HB232 program shells to HBoot to perform this task. The developer is free to call this program for the same purpose. Enter a ? command line switch and HBoot will show the command line switches available. Should the developer desire to "roll his own" a brief flow description is provided in section A5 (HBOOT).
PERVxx.IDX	An index to the personality files. This file is a plain ASCII text file. A description of the personality file is followed by it's filename within brackets ({}) spaced to the right such that the user does not see it. In the main program and the HBoot utility, the filename is parsed out when a line is selected.
HELPVxx.IDX	An index to the help files for the main program (HB232Vxx.EXE) in the same format as the personality file index. (Unlikely to be of much use to the developer)
*.PER	Various personality files. These files are plain text files from which the main program and HBoot extract information needed to customize their operation to the type of scanner. There is data from which arrays are constructed to decode the bytes returned from update requests. The Binary data required for download to the 'HC11 RAM is also contained within this file. See section A4 (PERSONALITY FILE) for a breakdown of this data. Viewing the personality files with a text editor will also help understand the format used.
*.HLP	Various help files for the main program. (Unlikely to be of much use for the developer)

NOTE: "xx" represents the current version number (eg: HBOOTV11.EXE). We intend to keep this convention. In order for your application to remain current, you could do wild card searches and choose the highest numeric version. An alternate method could be the use of a config file. The HB232 main program creates a config file named "HB232.CFG" so stay away from this name.

- 3.0 INITIALIZATION -

Two utilities are provided with the HB232, EEPUTER.EXE and HBOOTV11.EXE. The names of these programs may vary slightly with updates but their purpose will remain the same. EEPUTER.EXE installs the core program into EEPROM. HBOOTV11.EXE is a utility which will initialize the program and load the HC11's RAM with the appropriate code from a personality file.

The HBoot program is provided so that the developer does not have to deal with the intricacies of initializing the Interface Unit. It may be called from a batch file prior to running you program or from within your program. Once initialized the Interface Unit remains running regardless of the state of the host program. It stays current with the state of the scanner on it's own. The HB232's PC program makes use of a "/R" switch to bypass calling HBoot and save a little time when the user knows that the Interface has been previously initialized. Once the Interface Unit is turned off, the initialization process must be repeated.

A limitation of the HBoot utility is lack of support for Comm Ports 3 and 4. If you want to support these ports you will have to handle the initialization of the Interface Unit. Section A5 (HBOOT) is a guide.

- 4.0 COMMUNICATING WITH THE INTERFACE -

The Interface is setup by the initialization process to communicate at 9600 baud (N,8,1). The Command List included with this document shows the available commands and their actions. When an update is requested, the data is sent back in the format described in the Update Formats section of this document. When receiving updates, keep in mind that there may be some delays during the update transmission. This occurs because the Interface's LCD driver emulation is interrupt driven. The scanner's status is given priority and must be handled immediately. Once the data is sent from the scanner's CPU to it's LCD driver, there is no way of retrieving it. It must be monitored as it is sent, otherwise the Interface Unit will not have current information. We recommend use of a time-out scheme for each byte to avoid the possibility of your program hanging. Experimentation will provide the best value (we use approximately 50 milliseconds).

Another timing consideration is the transmission rate of data to the Interface. One or two byte commands are not usually a problem since the HC11 chip will buffer one byte while processing the previous byte. In general, some form of repeatback or response is provided with each command. We recommend waiting for the appropriate response before sending the next byte (again with a time-out to avoid hanging). Scanner key press commands have a longer delay associated with them. The scanner uses a keypad polling technique. A key must be held closed for a definite time interval and held open for the same interval for the scanner to recognize it. The scanner can be easily overran, resulting in missed keystrokes. The timing interval is a fixed value in the portion of the program downloaded to RAM. A default Time Constant is provided as part of the personality file for the use of the host program. This constant represents the total on time and off time of a key press plus a little margin. Adjustment of the Time Constant does not affect the fixed internal time value. The purpose of the Time Constant is to pace the transmission of key presses when attempting to send multiple key closures. We have seen some model differences with PRO2006 scanners in which the fixed internal timing is too quick to properly handle rapid key presses. In these cases we recommend the user utilize the personality file for the PRO2004/5 scanners. The major difference between these files is the fixed timing value.

<u>- 5.0 HINTS -</u>

The HB232 software for the PC was developed using QuickBasic 4.5 and third party tool boxes. Some of the following hints may not apply to other languages. These hints do not represent the only way to do something but rather are based on the way we did it. Experimentation will provide the best solutions to your situation.

5.1 - Timing

Use a time-out scheme when waiting for data to be returned from the Interface. When doing any action (such as a keypress) that causes the scanner's display to change, a delay in response from the Interface Unit will occur while it monitors the scanner's display data. This will also happen routinely when the scanner is in a scan or search mode of operation. A combination of waiting for the expected repeatback/response and a time-out scheme appears to work best. Flushing the receive buffer before each command will get rid of responses received after a previous time-out. Timing problems show up most often when trying to rapidly press the scanner's keys (AutoPrograming).

5.2 - Test on a variety of scanners

The biggest differences between the PRO2004/5/6 series of scanners is in internal timing.

5.3 - Don't change the Interface Unit's speed

The software for the HC11 assumes an 8 Mhz crystal. Using a different crystal will result in different baud rates as well as different internal time values for key press control. Don't try to initialize the HC11 with the 0 byte (as in HBoot) at 9600 baud. This is faster then the default bootstrap baud rate and will not be recognized as a 0. 1200 baud is recommended here. Once initialized, the Interface Unit sets itself up for 9600 baud. NOTE: A continuous break longer then one byte time at 1200 baud should work.

5.4 - Make sure correct Binary is in the HC11 RAM

Use the commands for checking the sigbyte and the checksum of RAM. If the RAM is not loaded with the proper program you will lose control of the Interface when you ask for an update! An update request transfers control to RAM. Internal keypress timing also depends on finding a timing value in RAM.

5.5 - Initialize as soon as possible

When the scanner is turned on, it's CPU will initialize the LCD driver. If the Interface Unit is not initialized yet, it will miss some of this data. The ERROR & BATT indications will be set and blinking will not be enabled. The original HB232 system used the bootstrap features to download the entire program upon startup which took about 10 seconds. The only way to catch the scanner's initialization stream was to perform a back panel reset. The program was split up and the LCD driver emulation moved to EEPROM. The program in EEPROM can be started by sending only one byte (0). HBoot essentially waits for something to show up on the serial port (a clue that the HC11 is powering up) and after a short pause (to give a chance to finish powering up) will send a 0 byte. This causes the EEPROM's program to start in time to catch that first initialization stream. The HC11 powers up quicker than the scanner and is ready to go when the scanner starts.

5.6 - Checksum and data validity

The RS232 driver in the Interface Unit does not provide full 12 volt swings. Normally this is not a problem, but we have seen one serial card that picked up a lot of noise. The checksum was added to help with this situation. An invalid checksum indicates that the data was corrupted between the Interface Unit and the PC. Data corruption between the scanner and the Interface Unit is not detectable. One particular type of error pops up intermittently, a bogus frequency digit due to the scanner's display data changing during an update request. It usually appears as an asterisk. When the scanner locks up in response to a squelch break, the data stabilizes quickly. It wouldn't hurt to make sure it remains stable for several updates before trusting the value.

5.7 - Update rate

If you continuously request updates keep in mind that you may not see every change in scanner status. On a slow computer you will miss some data because the scanner is scanning faster than you can get updates. This is not always obvious. One implication is that you should not look for exact values during a scan or search. For instance, if your program waits for the scanner to reach 150.0000 Mhz then performs some action it will not happen if the update occurs just before and just after that frequency occurs. You will get better performance by looking for an equal to or greater match. You may be using a 386/33 but your customer may be using a 4.77 MHz XT.

- A1. COMMAND LIST -

Cmnd (hex)	**Description**
00	Initialize or check comm's. The Interface will respond with FF (hex) if the program in EEPROM is running.
0E	The Interface will download the following 768 bytes to it's RAM (this is the personality's binary data and must be 768 bytes exactly). The command (0E) and each of the download bytes are echoed back to the PC.
1E	Check signature byte (sigbyte). This command looks at ram and reports back the contents of the sigbyte memory location. The sigbyte is a unique id for each personality file and can be used to verify that the proper binary has been downloaded. Only the sigbyte is returned (the command is not echoed back).
2E	RAM Checksum. This command adds up the bytes in the 768 byte personality binary area in RAM and reports back the lowest eight bits (low byte) of that sum to the PC. Used to verify the integrity of the downloaded data. Only the checksum is returned (the command is not echoed back).
6E	Output Byte (outbyte) follows. This command signals the Interface that the next byte sent is to be placed on the hardware output port. The command (6E) and the outbyte are echoed back to the PC.
7E	Output User Switch (outsw) follows. This command is similar to the previous except that the data is placed in a buffer which is OR'd with each keyboard command in order to open, close or maintain a User switch position. The user switch setting should be in the upper nibble of the data byte. The command (7E) and the outsw byte are echoed back to the PC. NOTE: Unlike the Output Byte command, the user switches are not changed directly by this command but are activated with the next keypad command. Activation can be forced by sending a keypad command which does nothing such as 01 00. Sending 00 00 will not work since this is an Interface command (actually command 00 twice).
8E,9E,AE BE,CE,DE EE,FE	These commands provide different levels of scanner data (updates) to the PC. The last byte sent to the PC is always a checksum (low byte) of the data provided in response to the command. The command is not echoed back.
Key Cmds	Any other command in which the lower nibble (of the first byte sent) is not hex E or hex F (ie; xE or xF) is treated as a key pad command. The byte is passed to the keypad matrix and the interface waits for a second byte which is passed to the matrix to complete the command. After a preset time the interface clears the command from the matrix (releases the key). Refer to section A3 (CODES) for valid Key Pad commands. Both bytes are echoed back.

<u>- A2. UPDATE FORMATS -</u>

Byte 1	-4 bit opmode code- , Squelch(not), Pri, Delay, Lockout
	<8E (hex) returns Byte 1 and a checksum>
Byte 2	Freq (LSD)
thru	Decimal point, -7 bit ASCII code-
Byte 9	Freq (MSD)
	<9E (hex) returns Bytes 1 thru 9 and a checksum>
Byte 10	Channel (LSD)
thru	Decimal point, -7 bit ASCII code-
Byte 12	Channel (MSD)
	<AE (hex) returns Bytes 1 thru 12 and a checksum>
Byte 13	Blink On, Blink Step, Blink Rcvmode, P, Ch, Mhz, Err, Batt
	<BE (hex) returns Bytes 1 thru 13 and a checksum>
Byte 14	- 4 bit Step code - , - 4 bit Rcvmode code -
	<CE (hex) returns Bytes 1 thru 14 and a checksum>
Byte 15	Input Switches (insw)- lower six bits valid.
Byte 16	Input Byte (inbyte)- 8 bit hardware input.
	<DE (hex) returns Bytes 1 thru 16 and a checksum>
Byte 17	0, Monitor, m1, m2, 0, Bank, b1, b2
Byte 18	m3, m4, m5, m6, m7, m8, m9, m10
Byte 19	b3, b4, b5, b6, b7, b8, b9, b10
	<EE (hex) returns Bytes 1 thru 19 and a checksum>
Byte 20	
thru	Same as Byte 17 - 19 except indicates blinking
Byte 22	
Byte 23	Checksum (low byte)
	<FE (hex) returns Bytes 1 thru 23 - full update>

- A3. CODES -

	Opmode Codes	Step Codes	Rcvmode Codes
0000	-		
0001	- Scan	5	nfm
0010	- Search Up	12.5	wfm
0011	- Search Down	30	am
0100	- Manual	50	
0101	- Program		

KeyPad Codes

1st byte (hex)	2nd byte (hex)	Scanner Key
01	01	7
01	02	9
01	04	Enter
01	08	8
01	10	Delay
01	20	(down arrow)
01	40	Reset
01	80	Mode
02	01	1
02	02	3
02	04	**not used**
02	08	2
02	10	Manual
02	20	Limit
02	40	**not used**
02	80	Pri
04	01	0
04	02	not used
04	04	Clear
04	08	**.** (decimal pt)
04	10	Lock-out
04	20	Direct
04	40	Monitor
04	80	Step
08	01	4
08	02	6
08	04	Program
08	08	5
08	10	Scan
08	20	(up arrow)
08	40	Lock-out Review
08	80	Speed

- A4. PERSONALITY FILE -

```
/ Header - title, copyright, description, etc.
/ Sections of the per file are identified with <section name>
/ Additional data follow the section name.
/ "C" means coded.  C4 = 4 bit coded data (ie; 16 possible values)
/ "D" means discrete.  D4 = 4 bits discrete data (ie; 4 possible values)
/ "ASCII7" represents a 7 bit ascii code (ie; 128 possible ascii values)
/ Other types of data are described below following section name.
/ The term "offset" refers to where the data is displayed on the screen of the
/ HB232 main program.  An offset value of 0 means that the data is not displayed.
/ NOTE: All values are DECIMAL values in the per file

<BEGINPER>                  /marks start of per file
<BYTE1> C4,D4               /data below section name in C4,D4 order (msb - lsb)
   :
 (sixteen entries -->  data, offset)
   :
 (followed immediately by)
   :
 (four entries -->  data, offset)
   :
<BYTE2─9> D1,ASCII7,281         /D1 = decimal point, frequency, offset
<BYTE10─12> D1,ASCII7,265       /D1 = decimal point, channel, offset
<BYTE13> D8
   :
 (eight entries --> data, offset)
   :
<BYTE14> C4,C4
   :
 (sixteen entries --> data, offset)
   :
 (followed immediately by)
   :
 (sixteen entries --> data, offset)
   :
<BYTE15> INSW
<BYTE16> INBYTE
<BYTE17> D8
   :
 (eight entries --> data, offset)
   :
<BYTE18> D8
   :
 (eight entries --> data, offset)
   :
```

- A4. PERSONALITY FILE -(continued) -

```
<BYTE19> D8
   :
 (eight entries --> data, offset)
   :
<BYTE20─22> BYTE17─19BLINK  /these bytes indicate blinking for bytes 17-19

<BYTE23> CHKSUM     /indicates that this is a checksum byte
<SCANCODE> 41       /indicates there are 41 entries in the scancode table
8,4,4
 :
(41 entries --> 7 bit ascii, 1st keypad code , 2nd keypad code)
 :
 :
118,8,64
<BINARY>  17, 156, 105      / SigByte, Checksum, Time Constant
    :
    :
 768 bytes of data in decimal organized as
 48 rows of 16 bytes each separated by commas
    :
    :
<ENDPER>
```

NOTE: The SigByte value following the <BINARY> section name is provided so that comparisons can be made to values returned by the check sigbyte command. The sigbyte that is loaded into the HC11's RAM is contained within the 768 bytes of binary that follows. The Checksum is the low byte of the sum of the 768 bytes that follow and should equal the value returned by the Checksum command if this per file binary is resident in RAM. The Time Constant is a reference value for the PC program to pace it's transmission rate to the Interface Unit. The internal timing value for the Interface Unit is contained in the binary data.

- A5. HBOOT -

The following is essentially a description of the internal workings of the HBoot utility provided with the HB232 Kit ($=HEX):

```
START
Setup for 9600 baud
GOSUB CommCheck
IF Comm's bad THEN
    PROMPT user to turn on Interface   (or, if on, turn off and back on)
    WAIT for anything to show up on serial port   (break from 'HC11)
    PAUSE for about 100 ms
    SEND $0 at 1200 baud    (only place 1200 baud used)
    switch back to 9600 baud
    GOSUB CommCheck
    IF Comm's bad THEN
        END with error message
    ENDIF
ENDIF
GET SigByte, Checksum, and Binary from Per file
GOSUB CheckDownLoad
IF Download bad THEN
    SEND $0E + 768 bytes of Binary from Per file
ENDIF
GOSUB CheckDownLoad
IF Download bad THEN
    END with error message
ENDIF
END

**********************************************************
SUB CommCheck
    SEND $0 at 9600 baud
    IF $FF returned THEN
        Comm's good
    ELSE
        Comm's bad
    ENDIF
    RETURN

SUB CheckDownLoad
    SEND $1E at 9600 baud and get back RAMsigbyte
    IF RAMsigbyte = SigByte form Per file THEN
        SEND $2E at 9600 baud and get back RAMchecksum
        IF RAMchecksum = Checksum form Per file THEN
            Download good
        ELSE
            Download bad
        ENDIF
    ELSE
        Download bad
    ENDIF
    RETURN
```

PROGRAMMER NOTES & PUNCHLIST

PROGRAMMER NOTES & PUNCHLIST