Adrienne Electronics Corporation

AEC-μBOX-2 Instruction Manual

TABLE OF CONTENTS

INTRODUCTION

Adrienne Electronics Corporation (AEC) developed the AEC-µBOX-2 (pronounced "a-e-c-micro-box-two") in order to provide a low cost way of reading Longitudinal Time Code (LTC) into a variety of RS232 and RS422 serial devices. For example, the AEC-µBOX-2 can be configured to emulate a Sony VTR (costing thousands of dollars) for time code capture purposes. This is just one of several serial interface protocols which are available.

This booklet has been prepared to assist you with installing and successfully using an AEC-µBOX-2 in your system. The INSTALLATION section starting on page 6 will help get you up and running. The TROUBLESHOOTING and LED OPERATIONS sections should be consulted if any problems arise. More advanced users who need to set up the box or write interface software will find the necessary documentation here also.

GETTING STARTED QUICKLY

In most cases the AEC-µBOX-2 will have been properly set up for you at the factory. If you want to use your AEC-µBOX-2 right away, without reading the whole manual, just do the following:

1) Use the USB power adapter cable provided to connect the "+5VDC" power jack on the box to any available (powered) USB port. This step may in some cases be omitted if the box will be receiving its power via the RS422 serial data lines (see #3 below).
2) Connect a known good source of LTC to the "LTC IN" connector.
3) Connect the 9-pin "D" connector to the equipment which will be reading the time code via the AEC-µBOX-2. You may use the custom RS232 cable provided to connect to standard 9-pin PC COM ports. If your box is configured to be a Sony VTR emulator, connect the RS422 cable which would normally go to a Sony VTR to the AEC-µBOX-2 instead.

The box should now be powered up and fully functional. If the LTC input is bad or missing, the "STATUS" LED will blink off every second. If something doesn't work, you will have to carefully read the INSTALLATION section of this manual.

WHAT IS TIME CODE?

Time Code assigns to each video frame (picture) a unique number, having the format Hours:Minutes:Seconds:Frames. This number may then be used for editing and/or control purposes. Time code standards have been around for many years, and are sponsored by both SMPTE (for NTSC) and the EBU (for PAL and SECAM). Time code is also used in the 24fps FILM environment, and in some audio only applications.

There are actually two different kinds of time code commonly used in the television business. Vertical Interval Time Code (VITC) is encoded in the vertical interval (non visible portion) of a video signal. Longitudinal Time Code (LTC) is a specialized audio signal which is usually recorded on an audio track of a video tape, next to its associated video signal.

Use of VITC frees up one audio track for other purposes, such as stereo audio. It also allows time code to be read at very low (including still) tape speeds, where fine positioning is important. However VITC cannot usually be read at tape speeds much above play speed.

LTC is commonly used because it is usually lower cost than VITC, can be added to a video tape after the initial video recording, and it can be read during high shuttle speeds (fast forward and rewind). However, due to record/playback limitations of tape machines, LTC cannot be read reliably at very low tape speeds.

Our AEC-µBOX-2 is an LTC reader. Our older AEC-BOX-10 is a VITC reader. And our AEC-BOX-20 can read both LTC and VITC simultaneously.

LEGAL NOTICES

SPECIFICATIONS

LTC INPUT:

Connector Type:	Isolated BNC (female contact)
Impedance:	20kohms typical (differential)
Input Level:	100mVpp to 10Vpp (single ended) 100mVpp to 20Vpp (differential)
DC on Input:	±1.0V maximum
CMRR:	>34dB @ 50Hz and 60Hz
Speed Range:	1/10x to 2.2x play speed (signal quality dependent)
Direction:	Forward or Reverse
Bits Read:	All (time, user, and embedded)
Standards:	SMPTE and EBU and FILM (automatically selected)

Note that in the professional audio world, the units "dBm" are often used to indicate an audio level or signal amplitude. The 0dBm reference level is defined as the voltage required to deliver 1mW of power into a 600ohm load (a standard impedance in television audio). This corresponds to a voltage of 0.77Vrms (1.5Vpp for a square wave, 2.2Vpp for a sine wave).

SERIAL PORT:

Connector Type:	9-Pin D-Subminiature (female contacts) (with custom pinout)
RS232 Pins:	Meet all RS232 electrical specifications.
RS422 Pins:	Meet all RS422 electrical specifications.

SPECIFICATIONS (continued)

POWER REQUIREMENTS:

Supply Voltage:	+3.6V to +8.5V (+5.0V typical)
Supply Current: (1)	20mA maximum
Total Power: (2)	100mW maximum

Notes:

(1) The supply current above does not include external RS232 and RS422 pin load currents, which are not under our control. Allow up to 20mA additional for a shorted RS232 output. Allow up to 70mA additional for a shorted RS422 output.

(2) The actual power consumption depends upon the voltage supplied and external load currents which are not under our control.

MISCELLANEOUS:

Operating Temperature:	0-50 degrees Celsius (32-122 degrees Fahrenheit)
Operating Humidity:	0-95%, noncondensing
Dimensions:	70mm wide x 31mm high x 130mm deep (2.76" wide x 1.22" high x 5.12" deep)
Weight:	186 grams (0.41 pounds)

DISCLAIMER:

All specifications are subject to change without notice as the product evolves over time. Our legal advisor made us say this too.

INSTALLATION

POWER CONNECTION:
The “+5VDC” connector is a standard 5.5mm by 2.1mm DC power jack. The AEC-µBOX-2 by itself requires less than 20mA at any voltage between +3.6V and +8.5V, but external loading of the RS232 and/or RS422 output lines may increase the total supply current required to approximately 100mA. Thus to be safe, we recommend using a +5V power source that can supply at least 100mA.

By coincidence, a standard USB port supplies at least 100mA at +5V, which is why we include a “USB power cable” with each AEC-µBOX-2. Simply plug the USB end of the cable into any (powered) USB port, plug the DC plug end of the cable into the AEC-µBOX-2, and hand tighten the threaded ring to ensure that the power cable doesn’t accidentally get pulled out of the AEC-µBOX-2 at an important moment. Note that the USB connector is used only to provide +5V power to the box. No USB signalling takes place, no USB drivers need to be installed, etc.. If no spare USB port is available, you may purchase from your favorite electronics retailer a “USB power adapter”, which is a small item which plugs into an AC power outlet and generates +5V in a USB connector for powering miscellaneous small electronic devices (like the AEC-µBOX-2).

Another option for powering the box is to use an external “wall wart” DC power supply. Make sure it is a “DC” supply, somewhere between +4V and +6V, can provide at least 100mA, and has a 5.5mm by 2.1mm DC plug on the end. These supplies are usually poorly regulated, and because the AEC-µBOX-2 draws so little current, a “+4V” supply might actually be supplying +8V or higher at that low current level. Be careful not to exceed the +8.5V maximum allowed DC voltage into the AEC-µBOX-2.

If too little voltage is supplied, the box will not function properly. If too much voltage is supplied, the big zener diode which protects the “+5VDC” input may overheat and fail. Powering the AEC-µBOX-2 with +9V or +12V batteries or power supplies, for example, would be a bad idea.

Lastly, because of its low current drain, in some systems the AEC-µBOX-2 may be fully powered by the incoming RS422 data lines. For example, when emulating a Sony VTR, no external power source may be needed. This feature only works if the RS422 source can supply at least +3.6V at 20mA and if its RS422 inputs are high impedance. If this configuration does not work reliably for you, use a separate power source.

INSTALLATION (continued)

LTC INPUT CONNECTION:
The LTC input connector is an isolated BNC connector with a female contact, marked “LTC IN”. Connect your LTC signal source here.

This LTC input is high impedance (20kohms typical). The differential input circuitry will reject at least 34dB of 50Hz or 60Hz hum. The input is designed to handle signals which have no more than ±1V of DC offset, and which have an amplitude of 100mVpp to 10Vpp. The maximum readable tape speed is 2.2x play speed. The minimum readable tape speed is tape machine dependent.

The AEC-µBOX-2 automatically adapts (within 4 seconds) to whatever standard of LTC you are providing, whether SMPTE (30fps), EBU (25fps), or FILM (24fps). No setup work is required. Once the box learns what kind of LTC you are providing, it stores this information in its internal EEPROM memory, which remains valid even when the box power is turned off. When power to the box is restored, it starts reading that same type of time code immediately (no waiting is required).

Be sure that the “LTC IN” BNC connector shield (outer conductor) does not contact any metal (conductive) objects. This is a differential audio input, and you normally do not want to ground the shield at the receiving end. You may choose to wrap black electrical tape around the connector body several times to make sure that such a short does not occur.

Because LTC is a specialized type of audio signal, it may be present on any audio track. Some VTR’s have a separate track dedicated to LTC, in which case you should connect the AEC-µBOX-2’s LTC input to the VTR’s LTC output connector. If LTC is present on a normal audio track, keep in mind that audio channels are often muted (turned off) when the tape is operating at anything other than play speed, thus making your LTC track temporarily unreadable.

INSTALLATION
(continued)

RS232/RS422 SERIAL PORT CONNECTIONS:
The 9-pin "D" connector on the AEC-µBOX-2 has socket (female) contacts and contains all of the RS232 and RS422 data lines. Whenever the box is transmitting, it drives both the RS232 and RS422 data lines simultaneously. The RS232 receive line is the same as the RS422 negative receive line, so only one standard can be used at any given time. The serial data cable you connect to the AEC-µBOX-2 determines whether RS232 or RS422 voltage levels will be used.

For historical reasons the connector pinout was chosen so that it looks like an RS422 ESbus tributary (and a Sony VTR). Thus if you are using the box as a Sony VTR emulator, just connect the RS422 cable that you would normally plug into a Sony tape machine directly to the AEC-µBOX-2. Nothing else needs to be done.

If you are using RS232 voltage levels, note that this 9-pin "D" connector has a nonstandard RS232 pinout. We provide a nonstandard RS232 cable with each AEC-µBOX-2 so that you can easily connect your box to a standard 9-pin PC "COM" port if needed. This is especially useful if you wish to boot to the CDROM which came with your box, run the test/demo program supplied, and verify that your AEC-µBOX-2 and your time code source are both working properly. Refer to the TROUBLESHOOTING section on page 30 of this manual for further details.

The pinout of this connector is as follows:

- Pin 1 => Chassis Ground
- Pin 2 => TX422-
- Pin 3 => RX422+
- Pin 4 => Receive Ground
- Pin 5 => TX232
- Pin 6 => Transmit Ground
- Pin 7 => TX422+
- Pin 8 => RX422- and RX232
- Pin 9 => Chassis Ground

Note the tiny numbers molded onto the connector face next to each pin, and be careful not to be "off by one" if wiring your own cable. Connect the transmit pin(s) on our box to the corresponding receive pin(s) on the other piece of equipment, and vice-versa.

SERIAL INTERFACE STANDARDS

RS232 Standard:
Interface signals are inverted versions of the UART (TXD and RXD) signals. A valid "1" is -5V to -15V. A valid "0" is +5V to +15V. Since RS232 drivers are always on, you can't bus them together, which makes RS232 strictly a point-to-point communication link. Before the advent of USB and other high speed serial data buses, RS232 was by far the most commonly used interface in the computer industry, and was usually seen as a 9-pin or 25-pin "D" connector on modems, terminals, serial ports (like the IBM PC COM1 and COM2 ports), and just about any kind of peripheral you could think of. Cables should be limited to 30 meters maximum, and the data rate should be limited to 19,200 baud maximum, in accordance with the RS232 standard. This AEC-µBOX-2 product does not have (and does not need) any of the handshake lines which are used by many other RS232 devices.

RS422 Standard:
The RS422 transmission standard allows for cables up to 1,200 meters long, and data rates up to 10Megabaud. It uses differential (2 complementary line) transmitters and receivers, which greatly reduces sensitivity to common mode noise. In addition, the transmitters can be set to a high impedance (Hi-Z) state, which allows several transmitters to share a pair of data lines. Thus many pieces of equipment can share the same data bus. For the "+" output, typical output low voltages are about 0V, and typical output high voltages are about +4V. For the "-" output, the signal polarity is reversed.

In AEC-µBOX-2 BROADCAST modes, the RS422 transmitter is always enabled. In POLLED modes, the RS422 transmitter is enabled only while messages are being sent. This enables multiple boxes (and other devices) to be controlled from one RS422 port.

ESbus Standard:
The ESbus (EBU/SMPTE Machine Control Bus) is used in the television industry to control VTR's, routers, switchers, mixers, and other equipment. The pinout of the AEC-µBOX-2's 9-pin "D" connector is that of an ESbus tributary. Experienced users will also note that the BINARY POLLED mode message protocol conforms with ESbus standards for a tributary device.

LTC READER OPERATIONS

The AEC-μBOX-2 uses proprietary hardware and software to read LTC from the signal present at the "LTC IN" BNC connector. Depending on the LTC signal quality, which varies from one tape machine to the next, the AEC-μBOX-2 can read LTC at tape speeds from 1/30x to 2.2x play speed. With most analog VTR outputs, reading LTC at tape speeds which are below 1/5x play speed is doubtful. Even a single bit error out of the 80 in each frame is enough to invalidate the entire frame. In the case where LTC read errors are detected, the AEC-μBOX-2 will always "hold on" to the last valid data read. Thus if the LTC signal is of poor quality, the time data will appear to momentarily "freeze", then will skip several frames. Also, if the LTC input signal disappears altogether, the last valid count will be held indefinitely.

The LTC input signal is first passed through a differential amplifier and a band pass filter to remove DC offsets, high frequency noise, and common mode noise. The cleaned up LTC input signal can be viewed at the "ALTC" test point on the board. This signal is then passed through a window comparator to convert the analog waveform into a digital signal, which can be viewed at the "DLTC" test point on the board. The window comparator has a threshold which increases as the LTC signal amplitude increases, thus providing very good noise rejection for normal amplitude (1Vpp to 2Vpp) LTC input signals. The digitized LTC input signal is then analyzed using a combination of hardware and software to extract the desired time bits, user bits, and embedded bits information.

The LTC output data by default is always "on-time", wherein our software automatically adds (forward direction) or subtracts (reverse direction) one LTC frame count. This ensures that the LTC frame count always matches that of the current video frame. The only potential problems with this scheme are that 1) the associated user bits will appear to lag behind by one frame, and 2) discontinuities (jumps) in the LTC data will be delayed by one frame count. This "on-time" feature can be disabled via a simple EEPROM modification (see page 15) or via the BINARY POLLED mode (using a "SET BOX MODE" command).

Because the SMPTE and EBU and FILM longitudinal time codes are so similar to each other, the AEC-μBOX-2 will read any of these LTC formats interchangeably. No setup work needs to be done.

LTC READER OPERATIONS (continued)

If you are having trouble reading LTC, first try the bootable test/demo program on the small CDROM which came with your box. Be sure the cable and connectors you are using are of good quality and are wired correctly. Use a continuity tester if necessary to check them out. Connect one end of the cable to the "LTC IN" connector on the AEC-μBOX-2, as described in the INSTALLATION section of this manual. Connect the other end of the cable to the "LTC OUTPUT" of the signal source. Because LTC is a specialized type of audio signal, and can thus be recorded onto any audio channel, you may have to experiment around to find out which channel has LTC on it.

AUTOMATIC MODE SWITCHING

The AEC-μBOX-2 automatically switches between SMPTE (30fps), EBU (25fps), and FILM (24fps) time codes as needed. It can take up to a second to switch from EBU or FILM to SMPTE. It takes about four seconds to switch to either EBU or FILM. These delays ensure that the box will not erroneously respond to jumps in the time code data (such as edit points) or to bad time code values. The box remembers the current time code standard in its internal EEPROM memory, which remains valid even when the box power is turned off. The box should thus always power up using the proper time code standard, and (unless you keep changing time code standards) you should never see any mode switching delays. It is possible to disable this automatic mode switching via an EEPROM change (see page 15) or via the BINARY POLLED mode (see page 26), but doing so can create other problems and is thus not recommended.

DIP SWITCH OPERATIONS

The AEC-μBOX-2 has a number of different software protocols available. The software protocol and the serial communication parameters to be used are determined by an 8-position DIP switch inside the box. In most cases the box will have been properly set up for you at the factory, and no DIP switch changes are necessary. Just connect the box to your equipment, see if it all works, and change the DIP switches only if something isn't working correctly.

CAUTION - Do not open up the AEC-μBOX-2 without observing proper ESD (electrostatic discharge) handling precautions. This product is protected against ESD damage only when fully and properly assembled in its enclosure. It is easy to damage the board inside the AEC-μBOX-2 if you are careless. It may become intermittent or fail outright, and you would then have to send the entire unit back to the factory for repairs. Once the board has been removed from the enclosure, do not walk up and touch it, or allow anybody else to do so. Before picking up or touching the board, first touch the table or whatever surface it is laying upon for about a second, then touch the 9-pin "D" connector shell (board ground).

Board Removal Procedure:

1) Remove all external cables.
2) Read and understand the ESD handling procedures above.
3) Use a small (#1) Phillips screwdriver to remove the two small black screws which are on the 9-pin "D" connector end of the enclosure.
4) Gently push the "LTC IN" BNC connector into the box. This will slide the board out of the other end of the enclosure a bit.
5) Pull the board out just far enough so that you can see the big red colored DIP switch assembly. Keep the big square microcomputer chip inside the enclosure for ESD safety purposes.

Changing DIP Switch Settings:

Note that the switches are numbered 1 through 8. We refer to these as "SW1" through "SW8". Also note the small "1" and "0" numbers down on the PCB next to both ends of the DIP switch assembly. To set a switch to be a "1", simply press down on the "1" (OPEN) end of that switch. Conversely, to set a switch to be a "0", simply press down on the "0" end of that switch. Those of you who are familiar with our older AEC-BOX-1 and AEC-BOX-2 products will note that this DIP switch layout is exactly the same as on those older products, for backwards compatibility reasons.

DIP SWITCH OPERATIONS (continued)

DIP Switch Functionality: (see also "Sony Protocol" on the next page)

Switch	Function
1	Baud Code #1
2	Baud Code #2
3	Odd(1) or Even(0) Parity
4	Parity Enabled(1) or Disabled(0)
5	Seven Bit ASCII(1) or Eight Bit Binary(0)
6	Box Address Bit #2 (see page 14)
7	Box Address Bit #1 (ditto)
8	Broadcast(1) or Polled(0) Operating Mode

The two "Baud Code" switches function as follows:

SW1	SW2		
1	1	=>	38400 baud
1	0	=>	19200 baud
0	1	=>	9600 baud
0	0	=>	1200 baud

Note that at 1200 baud, it takes longer than one LTC frame to transmit a message, so some frame numbers will be skipped.

Factory Default Setting:

Unless you requested otherwise, the factory default setting is BROADCAST mode, 9600 baud, 8-bit BINARY message protocol, and ODD parity, so SW1-SW8 will normally be 01110001 respectively. There is no way to alter the factory default of 1 stop bit.

Message Protocol Notes:

It does not make sense to send 8-bit time and user bits data in binary form over a 7-bit data link, since the top bit will be lost. Thus you will find that DIP switch #5 (SW5) controls the message protocol as well as the number of data bits being sent per serial character. If you select 8 data bits, one of the BINARY protocols on pages 18-26 will be used. If you select 7 data bits, one of the ASCII protocols on pages 28-29 will be used.

DIP SWITCH OPERATIONS (continued)

Special "Sony Protocol" Switch Setting:
DIP switches SW1-SW8 can be set to 11110111 respectively (only SW5 is low) to make the box look like a serially controlled Sony broadcast protocol tape machine. The serial interface is then 38400 baud, 8 bits, odd parity. This mode can be used to read time code into nonlinear editors and other equipment which expects to be connected to a real Sony VTR (at a fraction of the price).

Box Address Bit Usage:
In BROADCAST modes, and in Sony VTR Emulator mode, the two "Box Address" bits (SW6 and SW7) are not used. In POLLED modes of operation, SW8 is always zero, and SW6 and SW7 form an address offset of 0(000), 2(010), 4(100), or 6(110) with respect to the base address. For example, if SW6 is a "1", and SW7 is a "0", the address offset is 4. In this case the BINARY POLLED mode SELECT address will become AEC4h, and the ASCII POLLED mode box address will become "04". Normally SW6 and SW7 are both zero, giving an offset of zero, and the two box addresses are AEC0h and "00" respectively. The two Box Address bits allow you to have up to four boxes on the same (RS422) data line, each with a different address, without having to do any EEPROM modifications.

Box Reassembly Procedure:
1) If you accidentally pulled the board out too far (shame on you), you will have to be very careful putting it back together to avoid possible damage. Note the big metallic "spring clip" (J3) near the lower left corner of the board. This clip will have to be pressed inwards slightly when you slide the board back into the enclosure. Position the board in the second set of slots (counting from the bottom of the enclosure), slide it in gently until contact is made with spring clip J3, press J3 inwards just enough to clear the side of the enclosure, then slide the board in the rest of the way.
2) Carefully reinstall the two small black screws which you removed earlier. They should go in very easily. Be careful not to strip the threads in the Aluminum side extrusions!

EEPROM MODIFICATIONS

Almost all customers will be able to use their AEC-µBOX-2 without any DIP switch changes or EEPROM modifications, in which case you don't need to read this section. For more advanced users, if you are using the BINARY POLLED protocol, several aspects of box operations can be modified by sending the SET BOX MODE command (see page 26). Otherwise the only way to modify box operations is to use a special sequence of DIP switch settings to change data in the microcomputer chip's internal EEPROM memory. This EEPROM memory retains its data even when power to the box is turned off, so any changes you make to the EEPROM data should be considered permanent (unless you change them again, or reset the entire EEPROM).

If you really really need to make an EEPROM change, first read the "DIP SWITCH OPERATIONS" section of the manual, especially the sections regarding ESD handling precautions, and the board removal procedure, and the box reassembly procedure.

General EEPROM Data Change Procedure:
1) Make a working copy of the "EEPROM Change Worksheet" on page 17.
2) Determine (from the documentation which follows) the address of the EEPROM byte which needs to be changed, and enter that address onto the worksheet. Use the "NUMBER CONVERSION TABLE" on page 35 if you need help translating hex digits into their binary equivalents.
3) Determine also the data pattern which you wish to program into the selected EEPROM byte, and enter that data onto the worksheet.
4) Follow the board removal procedure on page 12, being careful to follow all of the ESD handling precautions so that you don't accidentally "zap" and disable your board.
5) Note the existing DIP switch settings, and write them down on the worksheet. This is very important, so that you can put the box back into its original operating mode when you are finished programming.
6) Follow the instructions on the worksheet to program the EEPROM. If you need to modify more than one EEPROM byte, just repeat the entire process as many times as needed until all EEPROM bytes are properly programmed (only one byte can be programmed at a time).

EEPROM MODIFICATIONS (continued)

EEPROM Register Map:
Following is a public map of the EEPROM registers which you can modify if needed. Do not attempt to modify any other EEPROM registers (not listed here) without our express permission.

EEPROM Address	Register Description
80h	ASCII Mode Upper Address Byte (default is "0" (30h))
81h	ASCII Mode Lower Address Byte (default is "0" (30h)) The software assumes that these two base address bytes will both be restricted to the "0-9" range (30h-39h).
82h	Binary Mode Upper SELECT Address Byte (default is AEh)
83h	Binary Mode Lower SELECT Address Byte (default is C0h) Note that bit 7 of both bytes must be "1", and bit 0 of the lower byte must be "0".
84h	Sony VTR Emulator Mode Upper ID Byte (default is 10h)
85h	Sony VTR Emulator Mode Lower ID Byte (default is 00h) The default 1000h indicates a Sony BVU-800 VTR. Contact Sony if you really need a different VTR ID code.
89h	Serial Interface Option Bits (default is 00h) Bits 7-2 = 0 (reserved) Bit 1 = 1 if want to use 00h instead of break characters Bit 0 = 1 if want the RS422 output lines always active
90h	Box Mode Control Bits (default is 21h) (see bit map on page 26)

EEPROM CHANGE WORKSHEET

Make a working copy of this worksheet, then enter the necessary ADDRESS and DATA as described on the previous page, then follow the procedure below. A special sequence of DIP switch settings and power-on resets is required. It is assumed that you have already read the EEPROM MODIFICATIONS instructions on page 15 and are following the ESD handling procedures on page 12. We suggest that you rotate the board 180 degrees so that SW8 is on the left and SW1 is on the right (makes data entry much easier).

If you accidentally change the wrong EEPROM byte, or program in the wrong data, or aren't happy with the results, you can always reset the entire EEPROM back to its factory default state by following this same procedure but with ADDRESS = 01010101 and DATA = 01010101.

EEPROM Change Procedure:	(MSB) (LSB)
1) Note the small "1" and "0" numbers down on the PCB next to both ends of the DIP switch assembly.	8 7 6 5 4 3 2 1
2) Write down the current DIP switch settings............ (will be used later to restore your setup)	_ _ _ _ _ _ _ _
3) Set all DIP switch segments to "1".........................	1 1 1 1 1 1 1 1
4) Power up the board, wait 1 second, power down.	
5) Set EEPROM byte ADDRESS................................	_ _ _ _ _ _ _ _
6) Power up the board, wait 1 second, power down.	
7) Set EEPROM byte DATA..	_ _ _ _ _ _ _ _
8) Power up the board, wait 1 second, power down.	
9) Set DIP switch segments to special pattern............	0 1 0 1 0 1 0 1
A) Power up the board. You should see a fast toggle pattern on the STATUS LED for about 10 seconds if the programming procedure went OK.	
B) Turn off power to the board, then restore DIP switch settings from step 2..................	_ _ _ _ _ _ _ _
C) Put the box back together per the instructions on page 14.	

Concluding Remarks:
The programming operation for this single EEPROM byte should now be complete, and the box should now be ready to resume normal operations. Power up the box and try it out. Please be sure that you are following this procedure exactly (no shortcuts) before calling us for assistance.

BINARY BROADCAST MODE

Assuming that the LTC input is OK, the AEC-µBOX-2 will use 8 data bits per character to transmit the following message every LTC frame:

Byte 0 = BREAK (20 bits low, then 2 bits high)
Byte 1 = XBh Status and length byte:
- Bit 7 = 1 (indicates that LTC data follows)
- Bit 5 = 1 if the LTC input is OK
- Bit 4 = 1 if skipped one or more LTC frames
- Bits 3-0 = # of bytes which follow (including checksum)

Byte 2 =	00h-29h	Time bits frames (packed BCD) (EBU=24h max)
Byte 3 =	00h-59h	Time bits seconds (packed BCD)
Byte 4 =	00h-59h	Time bits minutes (packed BCD)
Byte 5 =	00h-23h	Time bits hours (packed BCD)
Byte 6 =	00h-FFh	User bits frames
Byte 7 =	00h-FFh	User bits seconds
Byte 8 =	00h-FFh	User bits minutes
Byte 9 =	00h-FFh	User bits hours

Byte 10 = XXh Embedded bits plus miscellaneous flags:

	SMPTE	EBU
Bit 7 = 1 if an LTC parity error was detected		
Bit 6 = 1 if reading EBU (25fps) LTC		
Bit 5 = LTC bit 59	bin flag	phase bit
Bit 4 = LTC bit 58	reserved	reserved
Bit 3 = LTC bit 43	bin flag	bin flag
Bit 2 = LTC bit 27	phase bit	bin flag
Bit 1 = LTC bit 11	color frame flag	color lock flag
Bit 0 = LTC bit 10	drop frame flag	reserved

Byte 11 = XXh Status byte:
- Bit 6 = 1 if time code indicates FORWARD tape direction
- Bit 5 = 1 if time code is PLAYING (+0.8x to +1.2x)
- Bit 4 = 1 if time code is FAST (greater than 1.2x)
- Bit 3 = 1 if time code is SLOW (0.1x to 0.8x)
- Bit 2 = 1 if time code is STOPPED (less than 0.1x)

Byte 12 = XXh Checksum (8-bit sum of bytes 1-12 should be 0)

Notes:
1) All embedded bits have been removed from the time bits data, and appear instead in byte 10.
2) In the reverse tape direction, the PLAY flag is not used. Instead, the SLOW flag will cover the -0.1x to -1.2x range.
3) The AEC-µBOX-2 actually has the ability to read LTC down to 1/30x if the signal quality is good at that speed. The "STOPPED" flag may be set even though the tape is still moving slowly.
4) If all of the speed indicator flags are "0", the box is unable to detect the speed, usually because of poor signal quality.
5) This protocol was originally designed to handle only the SMPTE and EBU time code cases. If FILM (24fps) LTC is present, the "EBU" bit will be low (because it is not EBU time code), the frame count will vary from 00h to 23h, and the speed indicator bits will operate correctly. For backwards compatibility reasons, there is no "FILM" flag. The lack of frame numbers 24h through 29h indicates that FILM LTC is present.

If the LTC input is unreadable or missing, the STATUS LED will blink off every second, and the following short message will be transmitted at the same time:

Byte 0 =	BREAK	(20 bits low, then 2 bits high)
Byte 1 =	11h	(indicates only 1 byte follows)
Byte 2 =	EFh	Checksum (8-bit sum of bytes 1-2 should be 0)

For more information on transmitting and receiving BREAK characters, including the option of using 00h characters instead of BREAK characters, refer to page 27.

BINARY POLLED MODE

The box does not send any data until requested to do so. The protocol conforms closely with the ESbus machine control standard (when 38,400 baud and EVEN parity are selected) which is widely used in the television equipment industry. If you are not familiar with the SMPTE machine control standard, the state diagram on the next page will be very helpful. Eight data bits are used for all messages in this mode.

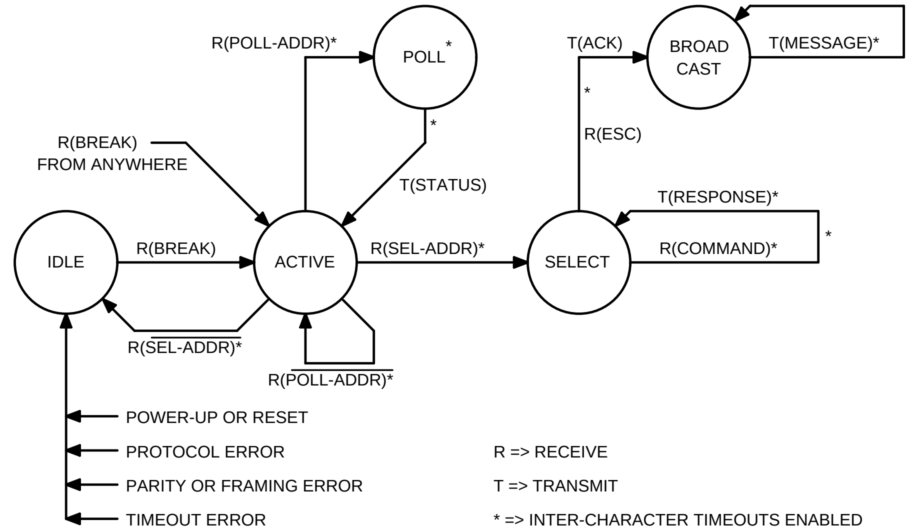
Controllers and Tributaries:
In the ESbus standard, there is one “controller” (typically your computer) which controls one or more “tributaries”. In this case, each AEC-μBOX-2 is a “tributary”, and must have its own unique 16-bit SELECT address and its own unique (but related) 16-bit POLL address.

SELECT Address:
The default SELECT address of AEC0h is stored in the EEPROM, and may be changed if needed (see page 15). DIP switches 6 and 7 form an address offset of 0(0000), 2(0010), 4(0100), or 6(0110) (switches 5 and 8 are always zero in this mode). This DIP switch address offset is then added to the address stored in the EEPROM to form the SELECT address for this particular box. For instance, if switch 6 is a “1”, and switch 7 is a “0”, the box SELECT address will become AEC4h. This allows you to have up to four boxes on the same data line, each with a different address, without having to do any EEPROM modifications. The MSB of both SELECT address bytes must be “1”, and the LSB of the lower SELECT address byte must be “0”.

POLL Address:
The POLL address is equal to the SELECT address plus 1.

Inter-Character Timeouts:
In all cases where two or more bytes are being sent in a message, including two-byte addresses, if the space between any two of these bytes exceeds six character periods, a timeout error exists, and the box will set its internal NAK flag and revert to the IDLE state.

Response Timeouts:
Whenever the controller expects a reply from the AEC-μBOX-2, the first character in the reply message must arrive within six character periods of the end of the previous message. Otherwise the controller can assume that the box is not responding.



BINARY POLLED MODE STATE DIAGRAM

IDLE State:
The IDLE state is entered:
1) When power is first applied or the box resets itself.
2) Whenever any protocol errors are detected.
3) Whenever any parity or framing errors are detected.
4) Whenever a timeout error is detected.
The IDLE state is exited whenever a BREAK character is received.

ACTIVE State:
The ACTIVE state is entered whenever a BREAK character is received, no matter what else the box was doing. The box immediately ceases all transmissions, then waits for a two-byte address (with inter-character timeout):
1) If the box’s POLL address is received, it immediately jumps to the POLL state.
2) If some other POLL address is received, it reverts to the ACTIVE state.
3) If the box’s SELECT address is received, it immediately jumps to the SELECT state.
4) If some other SELECT address is received, it goes back to the IDLE state and waits for a BREAK character.

BINARY POLLED MODE (continued)

POLL State:
This state is used to quickly determine which tributary devices (such as our box) have information ready for the controller. The AEC-µBOX-2 immediately transmits a single byte response, then reverts to the ACTIVE state. The POLL response byte may be:

1) 07h (RST) - Indicates that the box has powered up or reset.
2) 05h (NAK) - Indicates that a communication or protocol error of some kind has been detected.
3) 08h (SVC) - Indicates that the box needs to be serviced. For the AEC-µBOX-2, this means that a new frame of LTC data has been read.
4) 04h (ACK) - Indicates that nothing has changed.

The above responses are listed in order of priority (RST is highest).

A typical POLL sequence from the controller might be:

Byte 0 = BREAK	(20 bits low, then 2 bits high)
Byte 1 = AEh	Upper POLL address byte.
Byte 2 = C1h	Lower POLL address byte (LSB = "1").

The POLL response from the AEC-µBOX-2 might then be:

Byte 1 = 04h	ACK response byte indicating all is OK.

SELECT State:
This state is used for sending messages back and forth between the AEC-µBOX-2 and the controller. It includes the ability to place the box into the BINARY BROADCAST mode. If any errors are detected, the box sets the NAK flag and jumps to the IDLE state.

A typical SELECT sequence from the controller might be:

Byte 0 = BREAK	(20 bits low, then 2 bits high)
Byte 1 = AEh	Upper SELECT address byte.
Byte 2 = C0h	Lower SELECT address byte (LSB = "0").

Once in the SELECT state, the box will remain there unless errors are detected or a BREAK character is received. The SELECT state commands (listed beginning on the next page) may be sent to the AEC-µBOX-2 whenever it is in the SELECT state.

BINARY POLLED MODE (continued)

SELECT State Commands:

1) GET BOX ID command format (to AEC-µBOX-2) is:

Byte 1 = 3Fh	Question Mark ("?")

The response to this command (from AEC-µBOX-2) is:

Byte 1 = 02h	AEC-µBOX Number (2d)
Byte 2 = 41h-5Ah	Software Revision Letter (major) (A-Z)
Byte 3 = 31h-39h	Software Revision Number (minor) (1-9)
Byte 4 = XXh	Checksum (8-bit sum of bytes 1-4 should be 0)

2) GO TO BROADCAST MODE command format (to AEC-µBOX-2) is:

Byte 1 = 1Bh (ESC)	Escape to BINARY BROADCAST mode.

The response to this command (from AEC-µBOX-2) is:

Byte 1 = 04h (ACK)	Acknowledges receipt of this command.

The box will now be in BINARY BROADCAST mode. You can force it back into the ACTIVE state with a BREAK character.

3) READ FULL MESSAGE command format (to AEC-µBOX-2) is:

Byte 1 = 09h (TEN)	Transmit Enable

The response to this command is a single message identical to what you would expect from the BINARY BROADCAST mode. The only difference is that the data returned is the last valid time code data read.

(SELECT state commands are continued on the next page)

SELECT State Commands (continued):

4) READ TIME BITS command format (to AEC-µBOX-2) is:

Byte 1 = 40h	Requests time bits (with no embedded bits).

The response to this command (from AEC-µBOX-2) is:

Byte 1 = 00h-29h	Time bits frames (packed BCD) (EBU = 24h max)
Byte 2 = 00h-59h	Time bits seconds (packed BCD)
Byte 3 = 00h-59h	Time bits minutes (packed BCD)
Byte 4 = 00h-23h	Time bits hours (packed BCD)
Byte 5 = XXh	Checksum (8-bit sum of bytes 1-5 should be 0)

The data returned is from the last valid time code data read.

5) READ USER BITS command format (to AEC-µBOX-2) is:

Byte 1 = 42h	Requests user bits.

The response to this command (from AEC-µBOX-2) is:

Byte 1 = 00h-FFh	User bits frames
Byte 2 = 00h-FFh	User bits seconds
Byte 3 = 00h-FFh	User bits minutes
Byte 4 = 00h-FFh	User bits hours
Byte 5 = XXh	Checksum (8-bit sum of bytes 1-5 should be 0)

The data returned is from the last valid time code data read.

(SELECT state commands are continued on the next page)

SELECT State Commands (continued):

6) READ EMBEDDED BITS command format (to AEC-µBOX-2) is:

Byte 1 = 44h	Command Code

The response to this command (from AEC-µBOX-2) is:

Byte 1 = XXh Embedded bits plus miscellaneous flags:

	SMPTE	EBU
Bit 7 = 1 if an LTC parity error was detected		
Bit 6 = 1 if reading EBU (25fps) LTC		
Bit 5 = LTC bit 59	bin flag	phase bit
Bit 4 = LTC bit 58	reserved	reserved
Bit 3 = LTC bit 43	bin flag	bin flag
Bit 2 = LTC bit 27	phase bit	bin flag
Bit 1 = LTC bit 11	color frame flag	color lock flag
Bit 0 = LTC bit 10	drop frame flag	reserved

Byte 2 = XXh Checksum (8-bit sum of bytes 1-2 should be 0)

The data returned is from the last valid time code data read.

7) READ TIME CODE STATUS BITS command (to AEC-µBOX-2) is:

Byte 1 = 46h	Command Code

The response to this command (from AEC-µBOX-2) is:

Byte 1 = XXh Time code STATUS byte #1:
- Bit 7 = 1 if the LTC input is currently OK

Byte 2 = XXh Time code STATUS byte #2:
- Bit 6 = 1 if time code indicates FORWARD tape direction
- Bit 5 = 1 if time code is PLAYING (+0.8x to +1.2x)
- Bit 4 = 1 if time code is FAST (greater than 1.2x)
- Bit 3 = 1 if time code is SLOW (0.1x to 0.8x)
- Bit 2 = 1 if time code is STOPPED (less than 0.1x)

Byte 3 = XXh Checksum (8-bit sum of bytes 1-3 should be 0)

BINARY POLLED MODE (continued)

SELECT State Commands (continued):

8) SET BOX MODE command format (to AEC-µBOX-2) is:

Byte 1 = 63h		Command Code
Byte 2 = XXh		Box mode control bits:
	Bit 7 = 0	
	Bit 6 = 0	
	Bit 5 = 1 enables "on-time" LTC counting	
	Bit 4 = 1 if want to use EBU/PAL inputs and outputs	
	Bit 3 = 1 disables automatic SMPTE/EBU/FILM mode switching	
	Bit 2 = 0	
	Bit 1 = 0	
	Bit 0 = 1 enables the LTC reader	
Byte 3 = XXh		Checksum (8-bit sum of bytes 1-3 should be 0)

The response to this command (from AEC-µBOX-2) is:
Byte 1 = 04h (ACK) Acknowledges receipt of this command.

The power-on default box mode control bits are 21h ("on-time" counting, automatic mode switching, and LTC reader all enabled). Because the AEC-µBOX-2 automatically adapts to the incoming time code standard (SMPTE, EBU, or FILM), bit 4 is no longer needed, but is included here for backwards compatibility reasons.

9) READ BOX MODE command format (to AEC-µBOX-2) is:
Byte 1 = 64h Command Code

The response to this command (from AEC-µBOX-2) is:

Byte 1 = XXh	Current box mode control bits.
Byte 2 = XXh	Checksum (8-bit sum of bytes 1-2 should be 0)

The data returned in byte 1 have the same format as byte 2 of the SET BOX MODE command above, plus bit 6 should be a "1", indicating that LTC (not VITC) data is currently selected.

BREAK CHARACTER OPERATIONS

The transmit data pin of the microcomputer chip inside the box is normally high (inactive). When a normal serial data character is transmitted, the transmit pin first goes low for 1 bit period (the START bit), followed by 7 or 8 data bits, then a PARITY bit (if enabled), then finally goes high for 1 bit period (the STOP bit). The start bit for the next serial data character may start immediately thereafter.

Break characters are very different. A "break" character is defined as a special pulse which goes low for 20 bit periods, then goes back high for at least 2 bit periods.

Note that the RS232 output signal on the 9-pin "D" connector has a polarity opposite to that described above. The TX232 line, which is normally low (about -5.5V), pulses high (to about +5.5V) for 20 bit periods when the break character is transmitted, then goes back low.

The break character guarantees that the receiving UART will be properly locked to the serial data stream, even under worst case conditions. Since the binary mode message string includes user bits data, which can assume any value from 00h to FFh, the break character is also necessary to unambiguously define the start of a message string.

The break character can be detected in several ways:
a) Some UART's have a break character flag and/or interrupt, which makes your job real easy.
b) A break character will be received as 00h data together with a framing error.
c) If odd parity is being used, a break character will cause reception of 00h data with a parity error.

It has come to our attention that some communications software does not support the use of break characters. The AEC-µBOX-2 thus includes the option of using 00h characters instead of break characters. A simple EEPROM programming sequence enables this option (see page 15). Once programmed in this manner, your software is responsible for discriminating between the 00h at the start of a BINARY BROADCAST message, and possible 00h data (such as user bits) elsewhere in the message. Testing received data for validity, using inter-character timeouts, and performing a checksum calculation can prevent the reception of bogus data.

ASCII BROADCAST MODE

This protocol is much simpler than the BINARY BROADCAST protocol, but it is also much more limited in its capabilities. Assuming that the LTC input is OK, the AEC-µBOX-2 will use 7 data bits per character to transmit the following message every LTC frame:

Byte 1 = 30h-32h	0-2	Time bits hours, tens digit.
Byte 2 = 30h-39h	0-9	Time bits hours, units digit.
Byte 3 = 3Ah	:	Separator
Byte 4 = 30h-35h	0-5	Time bits minutes, tens digit.
Byte 5 = 30h-39h	0-9	Time bits minutes, units digit.
Byte 6 = 3Ah	:	Separator
Byte 7 = 30h-35h	0-5	Time bits seconds, tens digit.
Byte 8 = 30h-39h	0-9	Time bits seconds, units digit.
Byte 9 = 3Ah/2Eh	:/.	Separator (see note 1 below)
Byte 10 = 30h-32h	0-2	Time bits frames, tens digit.
Byte 11 = 30h-39h	0-9	Time bits frames, units digit.
Byte 12 = 20h/3Fh	/?	Status character (see note 2 below)
Byte 13 = 0Dh	CR	Carriage Return

Notes:
1) Byte 9 will be an ASCII period (2Eh) if SMPTE drop frame counting is being used. Otherwise it will be a normal ":" separator (3Ah).
2) In broadcast mode, byte 12 is always an ASCII space (20h). When in polled mode, if any read errors have been found, byte 12 will instead be an ASCII question mark (3Fh).
3) The old AEC-BOX-1 protocol is slightly different in that it omits the status character (byte 12 above) and thus sends the 0Dh carriage return immediately after byte 11. By setting all DIP switch segments to their normal values, but changing DIP switch segment SW7 to "1", the AEC-µBOX-2 can be forced to emulate the AEC-BOX-1 protocol exactly.

If the LTC input is unreadable or missing, the STATUS LED will blink off every second, and the following short message will be transmitted at the same time:

Byte 1 = 3Fh	?	Question Mark
Byte 2 = 0Dh	CR	Carriage Return

ASCII POLLED MODE

This protocol is much simpler than the BINARY POLLED protocol, but it is also much more limited in its capabilities. The box does not send any data until requested to do so. Seven data bits are used for all messages in this mode.

1) TIME REQUEST command format (to AEC-µBOX-2) is:

Byte 1 = 02h	STX	Start Transmission
Byte 2 = 30h-39h	0-9	Upper Address Byte (from EEPROM)
Byte 3 = 30h-39h	0-9	Lower Address Byte (from EEPROM) (3)
Byte 4 = 05h	ENQ	Enquiry

2) TIME REQUEST response format (from AEC-µBOX-2) is identical to that listed for the ASCII BROADCAST mode on the previous page.

3) BOX ID REQUEST command format (to AEC-µBOX-2) is:

Byte 1 = 02h	STX	Start Transmission
Byte 2 = 30h-39h	0-9	Upper Address Byte (from EEPROM)
Byte 3 = 30h-39h	0-9	Lower Address Byte (from EEPROM) (3)
Byte 4 = 3Fh	?	Question Mark

4) BOX ID REQUEST response format (from AEC-µBOX-2) is:

Byte 1 = 02h	---	AEC-µBOX Number (2d)
Byte 2 = 41h-5Ah	A-Z	Software Revision Letter (major)
Byte 3 = 31h-39h	1-9	Software Revision Number (minor)
Byte 4 = 0Dh	CR	Carriage Return

5) BROADCAST MODE command format (to AEC-µBOX-2) is:

Byte 1 = 02h	STX	Start Transmission
Byte 2 = 30h-39h	0-9	Upper Address Byte (from EEPROM)
Byte 3 = 30h-39h	0-9	Lower Address Byte (from EEPROM) (3)
Byte 4 = 1Bh	ESC	Escape

At this point, the box will enter the ASCII BROADCAST mode.
It will exit this mode upon receipt of an STX (02h) byte.

Notes:
1) If any transmission or protocol errors are detected, the STATUS LED will blink off for about 500ms, and the software will wait for the next 02h STX byte.
2) There are no timeout periods between bytes.
3) See box addressing information on page 14.

TROUBLESHOOTING

Please check everything out very carefully here, because historically over 90% of products returned to us for repair have absolutely nothing wrong with them.

The first step to take if you think that your AEC-µBOX-2 is not working properly is to make sure that it has adequate power. If the "STATUS" LED isn't coming on at all, then you have a power supply problem. Review the INSTALLATION section for power connection requirements.

Assuming that the LED is coming on at least some of the time, the next step is to make sure that the box is reading LTC properly. Disconnect the serial data cable (if any) from the 9-pin "D" connector to be sure that serial errors aren't causing the LED to blink (you may need to omit this step if the RS422 data lines are providing power to the box). If the LTC input is OK (and if there are no serial data errors), the LED will be on solid (no blinking). If the LED blinks off for a very short time every second, the box is indicating that the LTC input signal is missing or bad, perhaps due to a bad cable or a bad source signal. See the INSTALLATION section for LTC input connection requirements.

Assuming that the power is OK and the LTC input is OK, the only other thing that could go wrong is the serial communications path. Make sure that the proper serial data cable is connected (see the INSTALLATION section for details). If the box is in a POLLED mode and the LED is blinking off for 500ms periodically (or continuously), then some sort of serial communications error is being detected. This is usually caused by the DIP switches being set incorrectly for your system (incorrect baud rate, parity, etc.). There are two different tests that you can perform to help diagnose the problem.

If the box is in a BROADCAST mode (SW8 = "1") (test if needed), and if you connect the transmit data lines to the receive data lines, the STATUS LED will start blinking on and off at a 1Hz rate (with 50% duty cycle) if the transmit/receive loopback test is OK. For example, connect TX232 (pin 5) to RX232 (pin 8) and watch the LED. If the LED starts toggling at 1Hz, then the box's RS232 transmit and receive circuits are OK. This test can also be performed using the RS422 pins if you connect TX422+ (pin 7) to RX422+ (pin 3), and connect TX422- (pin 2) to RX422- (pin 8). By making these loopback connections at the far end of a serial data cable which is plugged into the AEC-µBOX-2, the cable can be tested also.

TROUBLESHOOTING (continued)

The other good serial communication path test is to connect your AEC-µBOX-2 to a COM port on an "IBM PC" type computer using the RS232 cable provided, then boot to the small CDROM which came with your box and run the "BOX20TST" program. If your CDROM has been misplaced, or if for some reason you never received one in the first place, or if you cannot boot to a CDROM, let us know and we will e-mail you whatever software is needed to make this work. The BOX20TST program cycles through all possible baud rates, parity settings, etc., until it finds the box, at which time it will display the current DIP switch settings and show you running time code if a good LTC source is connected. You can then change the DIP switches (see page 12) to whatever settings are needed by your application software.

If you are having trouble getting good results with one or more aspects of the demo program, refer to the corresponding section(s) of this manual. For example, if you are having trouble getting the LTC reader to work, refer to the "INSTALLATION" and "LTC READER OPERATIONS" sections of this manual. In this particular case, the LTC cable may be connected to the wrong place on a VCR, or the cable may be broken/intermittent, or there may be no LTC on the tape, or the tape speed may be too slow. We can sometimes help you with such problems over the phone, but more than 90% of the time the problems are external to our box, and we cannot see what's going on via a telephone link. Please triple check everything on your end before giving us a call.

If everything is working fine with the test/demo program (on the small CDROM), but you cannot get your application program to work properly, at least you know by now that the box itself is OK, and that the external signal sources and cables are OK.

Our AEC-µBOX-2 is used in many different applications, and we have no control over and probably no experience with your particular application. If you have tried everything mentioned above, and your box is working fine, but your application still doesn't work properly, please contact the people who sold you the application program for assistance.

AEC-μBOX-2 LED OPERATIONS

The "STATUS" LED on the box behaves in a variety of ways so that you can have some clues as to what is (or is not) going on inside the box.

If the STATUS LED fails to come on at all, either there is a power supply issue or the box is damaged and will have to be returned for repairs. The AEC-μBOX-2 can in some cases be powered by the incoming RS422 data lines (see page 6). If this is not working, or if it is not working reliably, then a separate power supply will be required (see also page 6).

If the LED blinks ON every 2 seconds in a "heartbeat" pattern, there must be some kind of internal hardware/software problem, and the box will have to be returned for repairs.

If the LED blinks OFF occasionally (or constantly), the box is working OK, but it is indicating that there is something unusual with the time code and/or serial data signals which are coming into the box.

The most serious problems, which blink the LED off for about 500ms per occurrence, are any serial data reception errors. These may include setup errors (wrong baud rate, parity, etc.), random communication errors (like parity or framing errors), or protocol errors. If there is no connection to the serial interface, no serial errors will be indicated.

The other possible problems are LTC read errors. If the LTC input is missing or unreadable for one full second, the LED will blink off for about 50ms (very short). The LED will then continue to blink once per second until the LTC input is restored.

The LED will blink ON and OFF at a 1Hz rate (with 50% duty cycle) if the transmit/receive loopback test is OK. This test is performed by making sure that the box is in one of the BROADCAST modes (switch 8 = "1"), then shorting the transmit and receive lines to each other (see page 30).

Note that no matter how many errors are detected, the LED will always come on at least once per second. This way you will know that the power supply is OK.

If the "STATUS" LED stays on all the time, everything must be running perfectly, and you can go read something else.

REPAIR PROCEDURES

Please contact us before returning any AEC-μBOX-2 for repair purposes, because historically over 90% of products returned have absolutely nothing wrong with them, and a lot of time and money gets wasted in the process. We will attempt to get you back on line via telephone or e-mail support, because that is the least expensive and fastest method. If for some reason that does not work, we will let you know where to return the product to.

There is essentially nothing inside an AEC-μBOX-2 which can be repaired in the field. Very special parts, knowledge, repair equipment, and test equipment are required. We do not make any money on repair jobs. If the box is still under warranty, and hasn't been abused, the repair and retest will cost you nothing except one-way shipping. If the box is out of warranty, we will let you know ahead of time what the repair cost will be.

You are always responsible for the full cost of shipping the box back to us, well packaged and insured. We will evaluate, repair if necessary, test, and will usually return the box within 1-2 days of receipt. We pay for shipping back to you, via a method comparable to what you used to get the box to us in the first place. If you send us a box via overnight courier, we will normally return it via overnight courier. If you send us a box via ground service, we will return the box via ground service.

It is very important that you include enough documentation with the box so that we know who you are, what's wrong with the box, where to return it to, a phone number in case we have any questions, and so forth.

WARRANTY

For the first two(2) years following the shipment of an AEC product, we will repair or replace, at our option, any such product which is found to be inoperative due to defects in materials or workmanship. Not covered is damage due to unusual electrical and/or physical abuse. Altered hardware, software, labels, or other identifying marks may also void the warranty.

AEC-µBOX-2 vs. AEC-BOX-2

The AEC-µBOX-2 was designed to replace the older AEC-BOX-1 and AEC-BOX-2 products for almost all applications. This page describes the main differences between these three products.

The AEC-BOX-1 “Play Speed LTC Reader with RS232/RS422 Output” was introduced in 1990. It only reads LTC at or near play speed. It is a transmit-only device, so it does not support Sony VTR emulator mode or any of the polled modes of operation.

The AEC-BOX-2 “Wideband LTC Reader with RS232/RS422 I/O” was introduced in 1991. It can read LTC up to 80x play speed, and supports five different software protocols, including Sony VTR emulation.

The AEC-µBOX-2 “Play Speed LTC Reader with RS232/RS422 I/O” was introduced in 2007 for a newer market, and as a smaller, cheaper, and easier to manufacture replacement for the other two products.

The AEC-µBOX-2 offers these advantages:
a) Smaller size, lower power consumption, less heat, lower cost.
b) Full DIP switch and software compatibility.
c) Full I/O connector compatibility.
d) CE-Mark acceptance for the European market.
e) Some operational characteristics can be changed in the field via DIP switch programming of the internal EEPROM (no EPROM editing is required).
f) Is more tolerant of ESD (sparks) and other electrical disruptions.
g) In many cases can be powered directly from the RS422 data lines, such as when placed into Sony VTR emulator mode.

The AEC-BOX-2 should be ordered instead of the AEC-µBOX-2 if:
a) A wideband (80x play speed) LTC reader is required.
b) The comparator software or the relay option is required.
c) CE-Mark acceptance is not required.
d) The higher price for the AEC-BOX-2 is not an issue.

PACKED BCD NUMBERS

A “packed BCD” byte contains two(2) BCD digits in an 8-bit byte. Bits 7-4 (the upper nibble) contain the upper BCD digit, and bits 3-0 (the lower nibble) contain the lower BCD digit.

For example, incrementing BINARY 09h leaves you with 0Ah, but incrementing PACKED BCD 09h leaves you with 10h. A packed BCD number such as 0Ah would be invalid, because “A” is not a valid BCD digit.

Here is one more example, showing the packed BCD format as used for time bits I/O. The 30 second (half minute) mark would be read (or written) as a 30h byte, even though 30 decimal is the same as binary 1Eh.

NUMBER CONVERSION TABLE

This chart will help you make conversions between the various numbering systems which are used in this manual.

Hexadecimal	(MSB) Binary (LSB)	Decimal	BCD
0	0 0 0 0	0	0
1	0 0 0 1	1	1
2	0 0 1 0	2	2
3	0 0 1 1	3	3
4	0 1 0 0	4	4
5	0 1 0 1	5	5
6	0 1 1 0	6	6
7	0 1 1 1	7	7
8	1 0 0 0	8	8
9	1 0 0 1	9	9
A	1 0 1 0	10	invalid
B	1 0 1 1	11	invalid
C	1 1 0 0	12	invalid
D	1 1 0 1	13	invalid
E	1 1 1 0	14	invalid
F	1 1 1 1	15	invalid
(base 16)	(base 2)	(base 10)	

(BCD is an abbreviation for “Binary Coded Decimal”)

7-BIT ASCII CODE CHART

ASCII is an abbreviation for “the American Standard Code for Information Interchange”. The keystrokes indicated below are those typically encountered on IBM PC type computers, and may be somewhat different for your particular computer model:

Dec	Hex	Key	Char	Description	Dec	Hex	Key	Char	Description
0	00h	^2	NUL	null char.	32	20h	SPC		space
1	01h	^A	SOH	start heading	33	21h	!	!	exclamation
2	02h	^B	STX	start of text	34	22h	“	“	double quote
3	03h	^C	ETX	end of text	35	23h	#	#	number sign
4	04h	^D	EOT	end of trans.	36	24h	$	$	dollar sign
5	05h	^E	ENQ	enquiry	37	25h	%	%	percent sign
6	06h	^F	ACK	acknowledge	38	26h	&	&	ampersand
7	07h	^G	BEL	ring bell	39	27h	‘	‘	apostrophe
8	08h	^H	BS	backspace	40	28h	(	(	left parenth
9	09h	^I	HT	horizontal tab	41	29h	)	)	right parenth
10	0Ah	^J	LF	line feed	42	2Ah	*	*	asterisk
11	0Bh	^K	VT	vertical tab	43	2Bh	+	+	plus sign
12	0Ch	^L	FF	form feed	44	2Ch	,	,	comma
13	0Dh	^M	CR	carriage ret.	45	2Dh	-	-	dash/minus
14	0Eh	^N	SO	shift out	46	2Eh	.	.	period
15	0Fh	^O	SI	shift in	47	2Fh	/	/	forward slash
16	10h	^P	DLE	data link esc.	48	30h	0	0	
17	11h	^Q	DC1	device ctrl 1	49	31h	1	1	
18	12h	^R	DC2	device ctrl 2	50	32h	2	2	
19	13h	^S	DC3	device ctrl 3	51	33h	3	3	
20	14h	^T	DC4	device ctrl 4	52	34h	4	4	
21	15h	^U	NAK	not ACK	53	35h	5	5	
22	16h	^V	SYN	synch. idle	54	36h	6	6	
23	17h	^W	ETB	end TX block	55	37h	7	7	
24	18h	^X	CAN	cancel	56	38h	8	8	
25	19h	^Y	EM	end medium	57	39h	9	9	
26	1Ah	^Z	SUB	substitute	58	3Ah	:	:	colon
27	1Bh	^[	ESC	escape	59	3Bh	;	;	semicolon
28	1Ch	^\	FS	file separator	60	3Ch	<	<	less than
29	1Dh	^]	GS	group sep.	61	3Dh	=	=	equal sign
30	1Eh	^6	RS	record sep.	62	3Eh	>	>	greater than
31	1Fh	^-	US	unit separator	63	3Fh	?	?	question mark

7-BIT ASCII CODE CHART (continued)

Dec	Hex	Key	Char	Description	Dec	Hex	Key	Char	Description
64	40h	@	@	at sign	96	60h	`	`	
65	41h	A	A		97	61h	a	a	
66	42h	B	B		98	62h	b	b	
67	43h	C	C		99	63h	c	c	
68	44h	D	D		100	64h	d	d	
69	45h	E	E		101	65h	e	e	
70	46h	F	F		102	66h	f	f	
71	47h	G	G		103	67h	g	g	
72	48h	H	H		104	68h	h	h	
73	49h	I	I		105	69h	i	i	
74	4Ah	J	J		106	6Ah	j	j	
75	4Bh	K	K		107	6Bh	k	k	
76	4Ch	L	L		108	6Ch	l	l	
77	4Dh	M	M		109	6Dh	m	m	
78	4Eh	N	N		110	6Eh	n	n	
79	4Fh	O	O		111	6Fh	o	o	
80	50h	P	P		112	70h	p	p	
81	51h	Q	Q		113	71h	q	q	
82	52h	R	R		114	72h	r	r	
83	53h	S	S		115	73h	s	s	
84	54h	T	T		116	74h	t	t	
85	55h	U	U		117	75h	u	u	
86	56h	V	V		118	76h	v	v	
87	57h	W	W		119	77h	w	w	
88	58h	X	X		120	78h	x	x	
89	59h	Y	Y		121	79h	y	y	
90	5Ah	Z	Z		122	7Ah	z	z	
91	5Bh	[	[	left bracket	123	7Bh	{	{	left brace
92	5Ch	\	\	back slash	124	7Ch	|	|	
93	5Dh	]	]	right bracket	125	7Dh	}	}	right brace
94	5Eh	^	^	caret	126	7Eh	~	~	tilde
95	5Fh	_	_	underline	127	7Fh	DEL		delete

EMI COMPATIBILITY

The AEC-μBOX-2 has been tested by an independent certified testing agency and found to meet all FCC and CE-Mark requirements, including electromagnetic emissions, electromagnetic immunity, ESD protection, fast transient protection, and so forth. Do not attempt to shock the AEC-μBOX-2 to see if it survives, because these tests could shorten the useful lifetime of your AEC-μBOX-2, and such deliberate abuse is not covered by the warranty.

FCC NOTICE

This equipment has been tested and found to comply with the limits for a "Class A" digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a COMMERCIAL environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed in accordance with this manual, could cause harmful interference to radio communications. Operation of this equipment in a residential area could cause harmful interference, in which case the user will be required to correct the interference at his or her own expense.

DECLARATION OF CONFORMITY

Here is all of the "CE-Mark" information, which is required for equipment being used in European Union (EU) countries. Application of Council Directive 2004/108/EEC. Standards to which conformity is declared are EN55022 Class A; EN61000-3-2; EN61000-3-3; EN55024:1998; EN61000-4-2; EN61000-4-3; EN61000-4-4; EN61000-4-5; EN61000-4-6; EN61000-4-8; and EN61000-4-11. Equipment description is "LTC Reader with RS232/RS422". Equipment class is "ITE Class A". Model number is "AEC-μBOX-2". Manufacturer is Adrienne Electronics Corporation, whose address and other contact information is on the inside back cover of this manual. This product is RoHS compliant, and is thus Lead free.

Adrienne Electronics Corporation
7225 Bermuda Road, Unit G
Las Vegas, NV 89119 U.S.A.
tel: +1-702-896-1858
fax: +1-702-896-3034
www.adrielec.com
<info@adrielec.com>