MODEL AM8e PCM/VF CALL ANALYZER WITH FEATURES 97

PROTOCOL TABLE DEVELOPMENT GUIDE

November 1996

Technical Data Subject to
Change without Notice

Ameritec
CORPORATION

760 Arrow Grand Circle
Covina, CA 91722 USA
TEL (626) 915-5441
FAX (626) 915-7181
www.ameritec.com
info@ameritec.com

We made our name with
American Technology

ISO 9001 Certified Company

For extra copies
of this manual,
order P/N 18-0040

RECORD OF REVISIONS

May 1, 1992	Last Preliminary Issue
June 1, 1992	Original Release
October 22, 1996	Minor Changes
October 29, 1996	Minor Changes
November 4, 1996	Minor Changes
November 6, 1996	Minor Changes
November 22, 1996	Minor Changes
December 6, 1996	Minor Changes
February 11, 1998	Updated Addendum (#18-0126)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1 The Audience

This document is intended for use by people who wish to develop custom protocol tables for the Ameritec Model AM8e 2.0 Mb PCM/VF Call Analyzer. Up to eight protocols may be stored and identified to the users requirements, providing that the 8K of memory allotted for protocol storage is not exceeded. The person undertaking this project should be familiar with as many of the following concepts as possible prior to starting customization procedures:

- Overall familiarity with the Ameritec AM8e product, its uses and applications.
- A good understanding of the function of telecommunications protocols as related to the AM8e.
- Basic knowledge of IBM compatible Personal Computer (PC) operation and use.
- A background in software programming with emphasis on state machine design and implementation.

Access to the following equipment is also required:

- Three Ameritec AM8e PCM/VF Call Analyzers.
- An IBM compatible (PC) computer with MS/PC-DOS operating system.
- Software required to compile AM8e protocol tables (see Appendix B).
- Software and a serial cable to download data from the PC to the AM8e.

1.2 The Ameritec PCM/VF Call Analyzer Model AM8e

The Ameritec PCM/VF Call Analyzer Model AM8e is a test set that emulates, monitors and analyzes signaling activity at an interface point on an established connection. The AM8e can emulate or monitor the near end or far end of either of two 2.048Mbps PCM spans. Tests (including "drop and insert") can be performed on any of 30 voice channels on the spans and may operate with up to eight different user specified protocols.

1.3 Summary

- This manual is a guide to programming AM8e protocol state tables in order to reconfigure the unit for specific user protocols.
- Programming protocol state tables requires some experience in telecommunications and programming.
- A personal computer is required to develop custom protocol state tables for the AM8e.

2. PROGRAMMING TEST SETUP PARAMETERS IN THE AM8E

Anyone who desires to create custom protocol state tables for the AM8e should first have a good working knowledge of the AM8e. The AM8e comes with an Instruction Manual (part #18-0035) which should be reviewed thoroughly prior to working with AM8e protocol state tables.

2.1 Programming AM8e Setup Parameters

The AM8e must be setup according to the tests to be performed. This is accomplished using three sets of parameters. These parameters are described in detail in the AM8e Instruction Manual. The following is a brief description of each set of parameters.

- **Configuration parameters:** These parameters configure the AM8e for the type of tests and reports desired, as well as controlling the RS232 configuration and unit Security. Selection of the active protocol is also made in the configuration parameters.
- **Emulation parameters:** These parameters define the values and timing of the events initiated by the AM8e. These parameters are only used when the AM8e configuration MODE is set to "emulate".
- **Analysis parameters:** These parameters set the values which the AM8e uses for analyzing the events that occur during a call sequence. Lower and upper ranges or thresholds are provided for event recognition and for flagging failures of specific events.

2.2 Interaction Between Setup Parameters and Protocols

The AM8e Configuration parameters determine whether the AM8e is emulating or monitoring either a switch or te/line, circuit type, display modes, RS232 options, security and print configurations.

One configuration parameter (PCM CONFIG 1 PROTOCOL) selects which protocol will be used. In the AM8e, protocols are stored based on previously-defined protocol state tables. The protocol state tables achieve compatibility with various protocols on PCM lines. The protocol state machine uses these tables to define which PCM digital signals are sent for various circuit conditions as well as which are expected.

Emulate parameters control the characteristics of the signals sent out from the AM8e. Analyze parameters provide the AM8e with data for distinguishing between incoming signals with acceptable characteristics and those which are out of range

2.3 Summary

- Anyone who desires to create protocol state tables for the AM8e should first have a good working knowledge of the AM8e.
- The AM8e uses Configuration, Emulation and Analysis parameters to control the unit configuration, test reports, RS232 configuration, unit security and the characteristics of signals sent out from the AM8e, as well as the characteristics of acceptable signals coming into the AM8e.
- Specific information on setup parameters as well as general unit information and specifications may be found in the AM8e Instruction Manual (P/N 18-0035)
- Additional copies of this manual may be ordered by contacting:

Ameritec
CORPORATION

760 Arrow Grand Circle
Covina, CA 91722 USA
TEL (626) 915-5441
FAX (626) 915-7181
www.ameritec.com
info@ameritec.com

We made our name with
American Technology

ISO 9001 Certified Company

3. LINE PROTOCOLS

Around the world many telecommunications signaling protocols have evolved to control telecommunications networks and functions. While these protocols are generally similar in concept, specific protocols vary by manufacture, region and nation. These variations lead to implementation problems for manufacturers and installers of equipment for the international telecommunications market.

The AM8e provides tremendous protocol flexibility through a completely programmable channel protocol state machine. State tables are used to configure this general purpose state machine, which is capable of interfacing the AM8e to virtually any line protocol.

This section deals with the development of state tables for the AM8e. It includes subsections on Channel Protocol State Machines (CPSM), gathering and diagramming required protocol information. It also includes writing, coding and debugging a protocol state table.

3.1 The Channel Protocol State Machine (CPSM)

A channel protocol state machine is a processing device which transitions between states, output actions or both, in reaction to events.

The state machine in the AM8e is a "finite-state" state machine, which can provide a finite number of outputs for a given finite number of inputs. For this reason, every possible output must be listed along with the required inputs to generate each output. In addition, the next state of the machine should be defined for each input.

The channel protocol state machine is implemented on a microprocessor dedicated to the channel interface. It is configured by loading protocol state tables into the AM8e from a PC.

The protocol state tables fully define the inputs (**Events**), outputs (**Actions** and **Signals**) and the **Next States**. The AM8e is capable of holding up to ten different protocol state tables in memory, subject to not exceeding the available 32K of memory dedicated to protocol storage. The desired protocol is selected from these ten, using the configuration parameter PCM1: PROTOCOL*.

The components of a Channel Protocol State Machine include the **States**, inputs (**Events**), outputs (**Actions** and **Signals**) and the **Next State** to which the state machine will transition. This subsection will describe these components as they relate to the AM8e.

* AM8e units with version 20C firmware or later may load ten different protocol state tables in memory, subject to not exceeding the available 32K of memory dedicated to protocol storage.

3.1.1 The Concept of State

The concept of state as it relates to telecommunications might be defined *as one of a set of various conditions having definite characteristics*. ONHOOK and OFFHOOK are states.

In the AM8e, the present State defines the current conditions at the channel interface and helps to determine the Actions to be performed, and the Next State to which the state machine will transition.

State names are defined by the user. e.g. "State INITIALIZE"

3.1.2 Events

Events are predefined by Ameritec. Event names always start with "**EV_**". Appendix A (Table A-1) lists the Event names.

Events may originate in several different places, including:

- The keypad or keyboard as input by the user (Examples are SEIZE, RING, REVerse, START and DIALing .)
- The channel interface (Example: RxSB (Receive signaling Bits))
- State machine internal facilities (examples include TIMER, COUNTER, etc.)

3.1.3 Actions

Actions are predefined by Ameritec. **Action** names always start with "**AC_**". Appendix A (Table A-2) lists the Action names.

Actions may be output in reaction to an **Event**. Some **Actions** use data values (**Action Data**) which vary the operation of the action routines. Because the CPSM has a rigidly defined structure each action has a data value, although it may not be used.

- **Actions** change the PCM output stream (transmit signaling bits or send dialing tones).
- **Actions** send information to the AM8e display (**Signals**, digits, etc.)
- **Actions** control internal state machine functions (**Timer, Counter, etc.**).

3.1.4 Signals

Signals are predefined by Ameritec. Signal names always start with "**SIG_**". Signals are sent to the AM8e display from the CPSM. "SIG_". Appendix A (Table A-3) lists the Signal names.

3.1.5 Next State

Next State is user defined and matches one of the **State** names defined in the State Table. **Next State** is the **State** which will be transitioned-to from the present **State** (if any). It will follow an Event.

If no **State** transition is desired, including cases when more Actions will be executed before the state transition, "NIL" (no transition will take place) should be the Next State name used.

3.2 Protocol Development Information

To develop an AM8e state table, an accurate and complete description of the line signaling protocol is required. These are usually available from telecommunications equipment manufacturers and service companies, or from telecommunications standards bodies worldwide.

This subsection describes the information required to develop a protocol state table and makes some suggestions for diagramming and manipulating this information which should make writing, coding and debugging the protocol state table easier.

3.2.1 Common Signaling Procedures

Some signaling procedures are common to many different protocols. Many of these procedures are automatically understood by the AM8e. The CPSM programmer doesn't need to do any extra work to accommodate the following procedures:

- Generate Pulse Dialing Digits
- Generate DTMF Dialing
- Generate MF(R1) Dialing
- Generate MF(R2) Dialing
- Analyze Pulse Dialing
- Analyze DTMF Dialing
- Analyze MF(R1) Dialing
- Analyze MF(R2) Dialing
- Display signaling Bits

NOTE: The CPSM must “enable” the analyzers. Some restrictions exist as to which analyzers may be simultaneously enabled.

3.2.2 Symmetrical and Asymmetrical Protocols

Line signaling protocols can be divided into two groups; symmetrical and asymmetrical.

Symmetrical protocols are those for which the procedures to originate and terminate a call are the same regardless of call direction. In other words, the signaling from near-end to far-end uses the same procedures as signaling from far-end to near-end. An example of a symmetrical protocol is signaling System R2 CCITT.

If a protocol is symmetrical, three state tables are required; one each to:

- Emulate the equipment connected to the line
- Monitor the near end
- Monitor the far end

These state tables are combined into one source file to generate the protocol.

Asymmetrical protocols require different procedures for each direction of signaling. An example of an Asymmetrical protocol is Loop Disconnect between a public exchange and a PBX. Another example is the TE-E&MK1-IN protocol for an AM8e in the outgoing Terminal mode connected to a Switch incoming line. Another protocol (not described) would be required for the incoming Terminal connected to the switch outgoing line.

In the case of Loop Disconnect, calls from the exchange to the PBX are typically initiated by an **ALERT** signal; but calls from the PBX to the exchange are initiated by a **SEIZE** (or **OFFHOOK**) signal.

If a protocol is asymmetrical, six state tables will be required, one each to:

- Emulate end A
- Monitor near end from end A
- Monitor far end from end A
- Emulate end B
- Monitor near end from end B
- Monitor far end from end B

NOTE: Emulate and Monitor State Tables should be combined into a source file for end A and a source file for end B. Each source file will be used to generate a protocol.

3.2.6 State Machine Transition Diagram (STD)

The next step in visualizing the state transition process is to draw a state transition diagram (STD). An STD provides a" bubble" for each state which is labeled with the name of the state. The bubble also should include the current status of the receive and transmit signaling bits.

One or more arrows are then drawn from the bubble to represent each transition to another (next) state. The arrow is labeled with the **EVent** which causes the state transition which is underlined to signify "input". **ACtions** and **SIGnals** which are associated with the event are also shown.

Usually a STD begins with a state labeled INITIALIZATION. The AM8e generates an initialization event (EV_INIT) and resets the state pointer to 0 at the moment the unit is placed "on-line" (this is done by activating the INPUT key). Depending on the particular signaling protocol, one or more steps may follow "EV_INIT". A list of possible initialization steps might include:

- Enable DTMF (Touch-Tone) Analyzer (AC_RXTT)
- Enable MF(R1) Analyzer (AC_RXMFR1)
- Enable MF(R2) Analyzer for forward or backward digits (AC_RXMFR2_F or AC_RxMFR2_B)
- Set bit patterns for DP Break (AC_BKBITS_...), Make (AC_MKBITS_...) and Interdigit time (AC_IDTBITS_...)

After the initialization **ACtions**, the CPSM should transition to the **IDLE State** (the state which exists when both ends of the circuit are Idle).

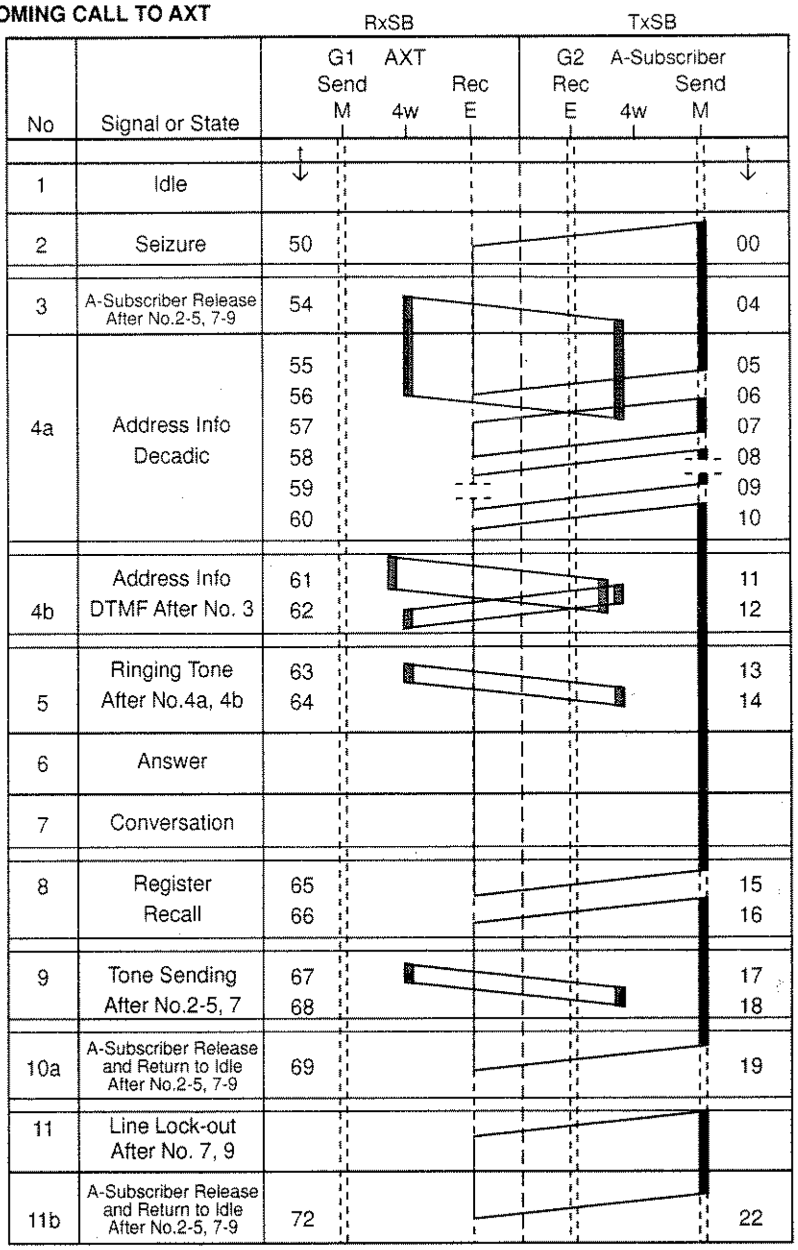
By consulting the Timing diagram, the programmer should now determine which **EVents** can occur while in the Idle State and which ACtions the AM8e should execute in response to each possible EVent.

Usually, there will be a list of **ACtions** to be performed (including any **SIGnals** that will be reported by the AM8e) followed by a required state transition. By continuing to follow the timing diagram, the programmer should be able to define a series of states that will be reached by state transitions based on **EVents**.

Once the normal sequence of states has been determined for each of the expected call possibilities, the programmer should go back and decide how the AM8e should respond to unexpected events in each state. This will usually result in extra error treatment states being defined.

CODES AND SEQUENCE DIAGRAM

INCOMING CALL TO AXT



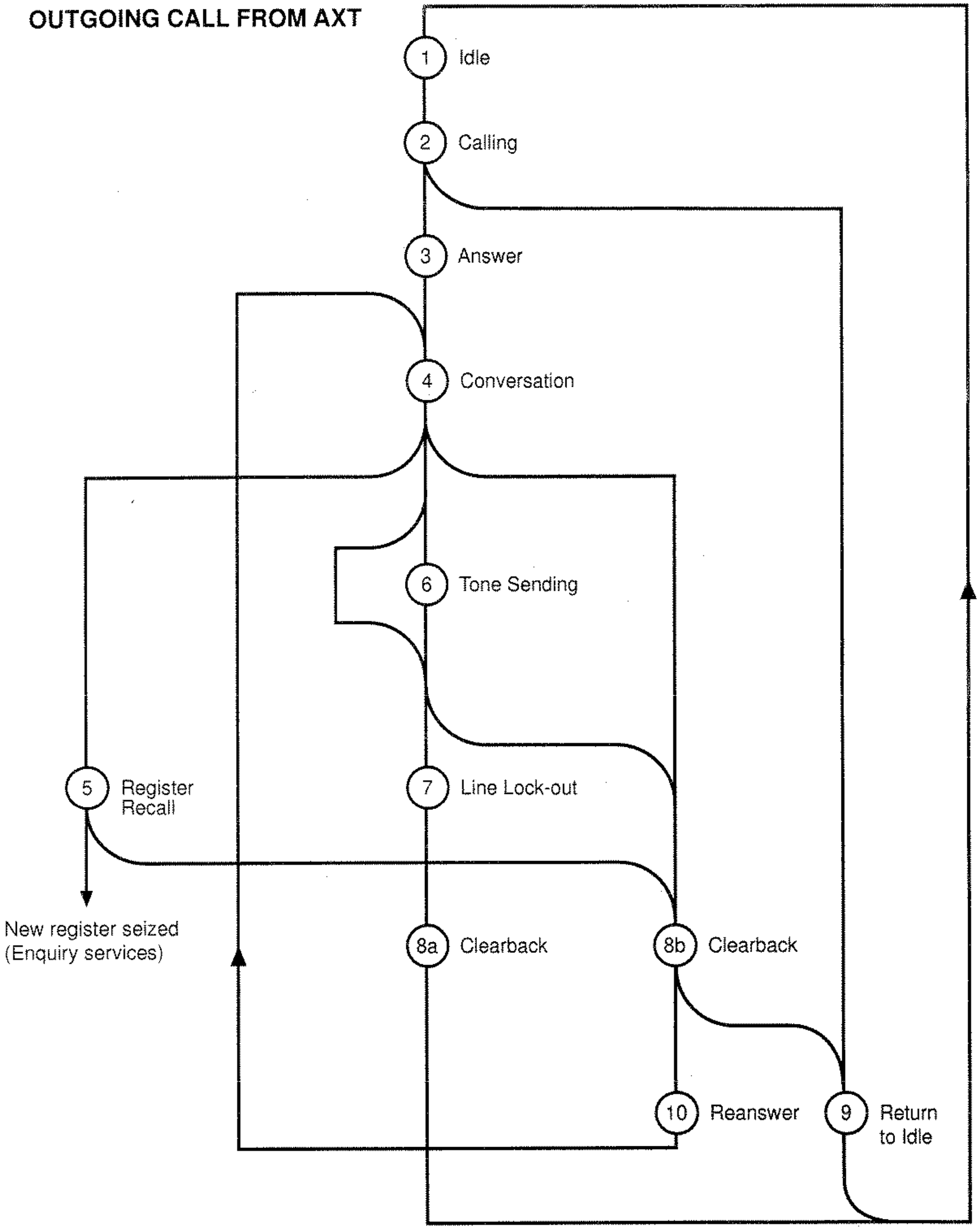
No	Signal or State	RxSB G1 AXT Send M 4w Rec E	TxSB G2 A-Subscriber Rec E 4w Send M
1	Idle		
2	Seizure	50	00
3	A-Subscriber Release After No.2-5, 7-9	54	04
4a	Address Info Decadic	55 56 57 58 59 60	05 06 07 08 09 10
4b	Address Info DTMF After No. 3	61 62	11 12
5	Ringing Tone After No.4a, 4b	63 64	13 14
6	Answer		
7	Conversation		
8	Register Recall	65 66	15 16
9	Tone Sending After No.2-5, 7	67 68	17 18
10a	A-Subscriber Release and Return to Idle After No.2-5, 7-9	69	19
11	Line Lock-out After No. 7, 9		
11b	A-Subscriber Release and Return to Idle After No.2-5, 7-9	72	22

Signal No. 5
The ringing tone is transmitted from the B-subscriber local exchange. Tone sending is intermittent and goes on until answer.

Signal No. 9
If the call set-up attempt fails, busy tone, congestion tone, etc will be transmitted from the originating local exchange (if possible). These tones are sent until the A-subscriber clears (state 10a), service digit is dialed or time out of tone sending occurs.

Figure 1 — Example Ladder and Sequence Diagrams (Sheet 1 of 4: Diagram A)

OUTGOING CALL FROM AXT



State No. 9 will occur when time supervision for reanswer signal has elapsed or after A-subscriber release.

Figure 1 — Example Ladder and Sequence Diagrams (Sheet 4 of 4: Diagram D)

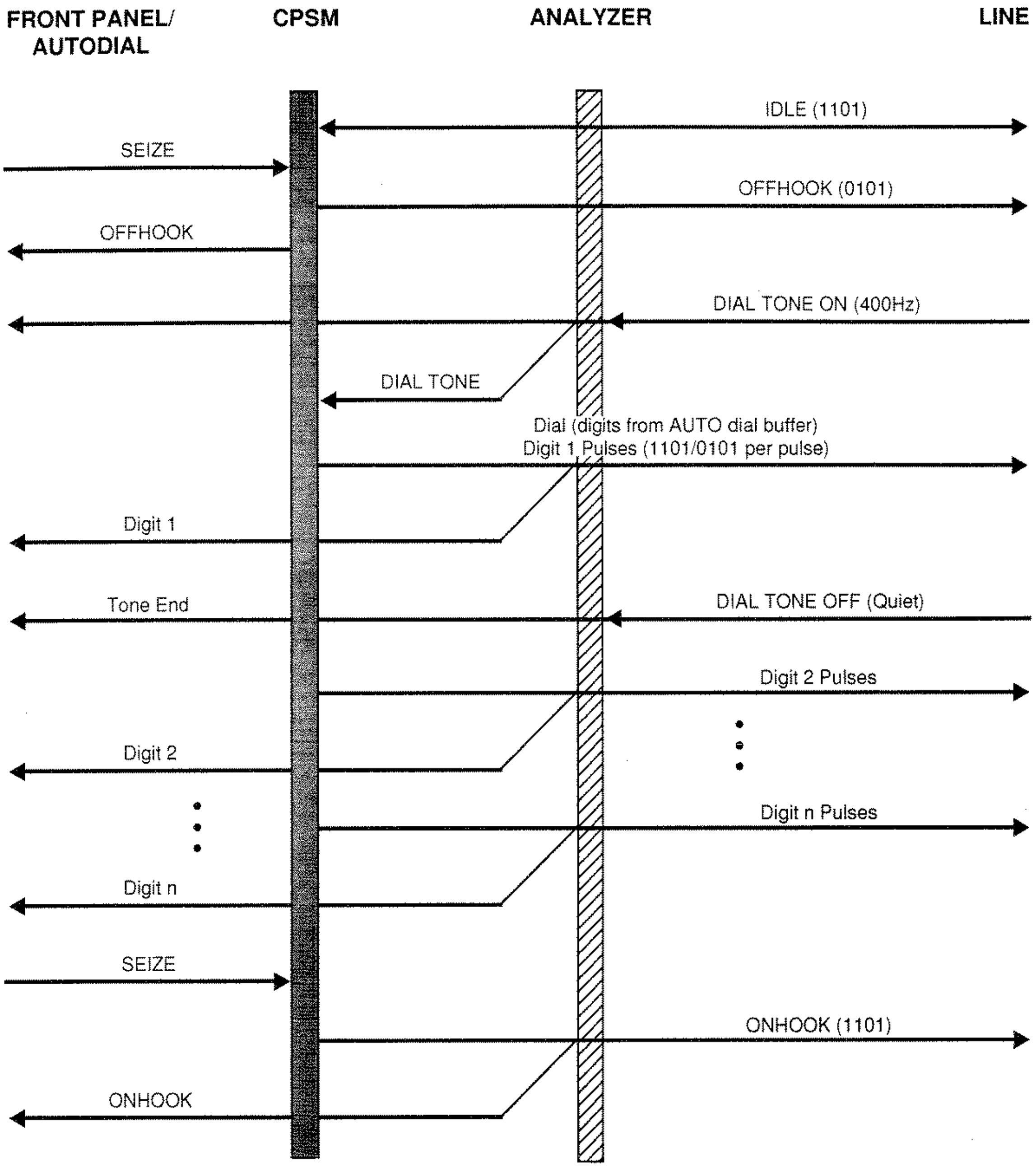


Figure 2 — Typical User-Developed Timing Diagram

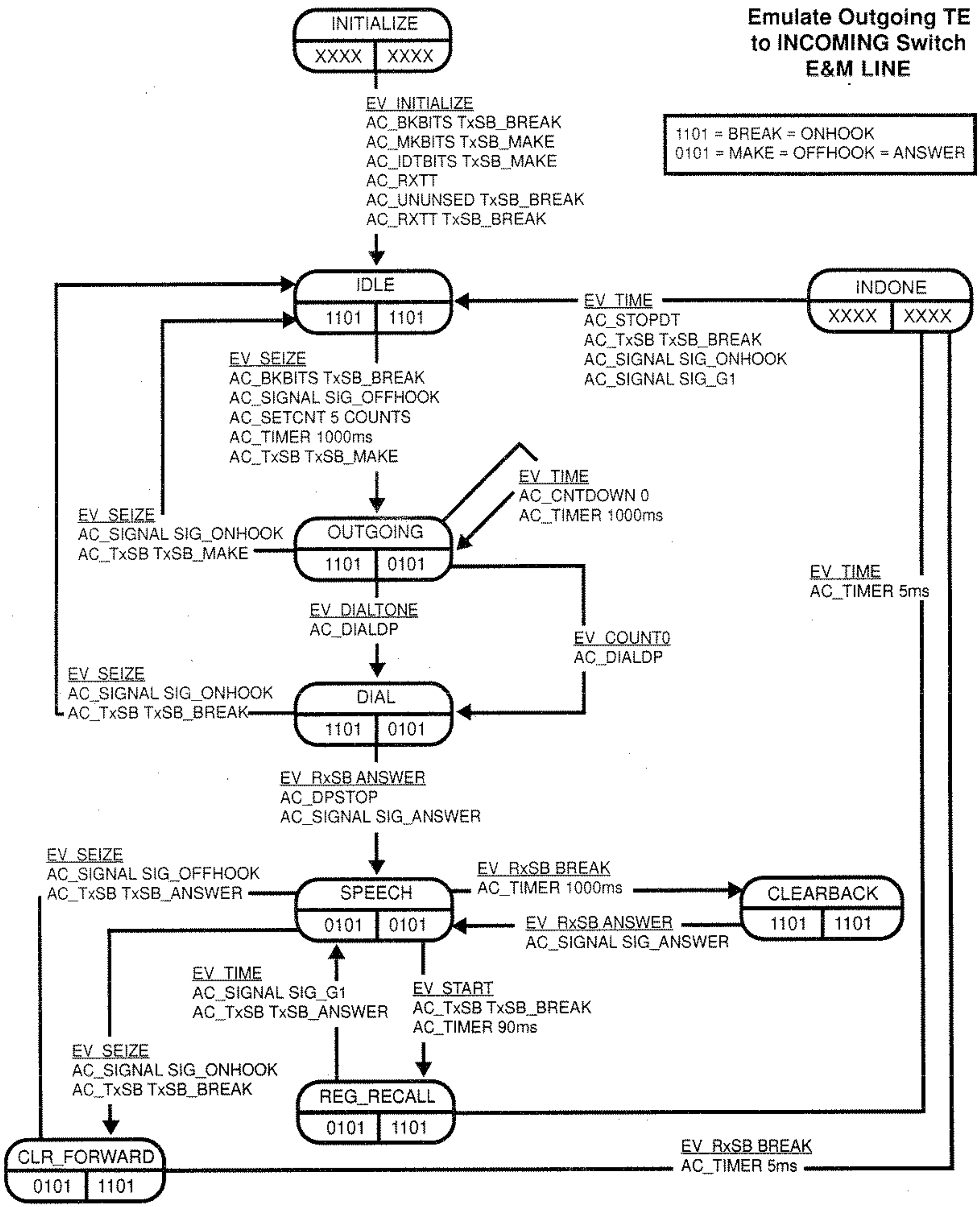


Figure 3 — State Transition Diagrams for Outgoing TE Protocol (Sheet 2 of 2)

3.3 Writing, Coding and Debugging a Protocol State Table

Timing and State Diagrams should be prepared with all events, Actions, States and State transitions defined. Once the required information is gathered and formatted correctly, the process of writing, coding and debugging a user defined custom protocol may proceed. In this subsection methods are presented to ease this task.

3.3.1 Writing a Protocol State Table

The first thing to do is develop tables which contain the information about each state in a sequence that approximates the way you will code the statements into the PC. Figure 4 provides a framework for supplying the basic parts of a single state definition and also provides code examples.

State Name

The user selected name of the state. Precede state name with the word "State";
e.g.: `State INITIALIZE`

Events	Next State	Actions	Action Data
Events which will cause a change of state. Precede the event name with the work "Event" and another space. (A comma must follow the event, next state and actions." Event names are listed in Appendix A.	The name of the next state must match another name in the state table. This entry may cause a transition to another state (loopint). If the event causes no transition (continuing with more actions for the same event) enter "NIL."	Actions to be performed when each event is recognized. The action occurs before the state transition. If no action, enter "AC_NIL." Action names are listed in Appendix B.	Data for the action specified. The data may be carried over to the next state. If no data is appropriate for the action, enter "0". A ";" and comment may follow. Action Data names are listed in Appendix C.
Examples:	Examples:	Examples:	Examples:
Event EV_SEIZE,	INITIALIZE,	AC_SIGNAL,	SIG_OFFHOOK
Event EV_START,	IDLE,	AC_SIGNAL,	SIG_G1
Event EV_REVERSE,	OUTGOING,	AC_TIMER,	10; x 5ms=50ms
Event EV_RING,	SPEECH,	AC_TxSB,	TxSB_MAKE
Event EV_DIALTONE,	INDONE,	AC_COUNTDOWN,	0; COUNT DOWN
Event RxSB_ANSWER,	NIL.	AC_FAREND,	FARIDLE
		AC_NEAREND,	NEARIDLE

State End

A standard statement which ends the State Definition;
e.g.: `StateEnd`

Figure 4 — State Definition Summary

3.3.2 Coding the Total Protocol State Table

Once tables defining each state are constructed, coding of the total AM8e protocol state table is straight-forward. All state definitions are combined into one text file. The text file has a simple syntax and layout, some of which has been described in Figure 4.

Prepare the files to build the protocol state tables with a plain ASCII text editor or a word processor, in non-document or ASCII mode, on an IBM PC-Compatible Personal Computer. After the text file has been written and checked, it is then passed through a Macro Assembler program to produce a Protocol State Table data file which may be downloaded to the AM8e.

A listing of a protocol state table, using the example protocol (Emulate mode for call origination), is shown in Figure 5. Observe the following list of guidelines and rules for writing the statements in the state table. (Indentation and tabulation may be used, as shown in examples, to improve readability, but are not required.)

1. **Comments** may be placed in the text by placing a semi-colon (;) at the start of the line or after the Protocol State Table text. Example:

```
;Outgoing Protocol
```

2. The **NT_EMULATE** statement defines the maximum number of transitions per state in the Emulate Protocol. Example:

```
NT_EMULATE     EQ    7; max number of transitions for Emulate
```

3. The **NT_MONITOR** statement defines the maximum number of transitions per state in the Monitor Protocol. Example:

```
NT_MONITOR     EQ    7; max number of transitions for Monitor
```

NT_EMULATE and NT_MONITOR must be placed in the state table before the INCLUDE statement.

4. The **INCLUDE** statement is required to merge Ameritec supplied files with the use Protocol State Table file. Example:

```
INCLUDE AM8e.INC
```

5. The **Headern** (n=0 for emulate or =4 for monitor) statements allow the user to name and number the Protocol State Table. This name will be visible on the PROTOCOL display after the table is downloaded to the AM8e. The name may contain any printable characters, must be enclosed in angle brackets ("<...>") and must be the same for both Headers. The maximum length of the name is 11 characters.

   One or two digits must follow the ","after the ">" and are used to provide table id numbers. **Neither the name nor the number may be duplicated in state tables for different protocols that are loaded into the same AM8e**, Examples:

   ```
   Header0 <TE-E&MK1-IN>,0;this header precedes emulate state table
   Header4 <TE-E&MK1-IN>,0;this header precedes monitor state table
   ```

6. The **Constant Definition** statements are "EQUates" which permit the programmer to use descriptive statements instead of the binary data, when required. Examples:

   ```
   ;CONSTANT DEFINITION (introductory comment BREAK event or data shown)
   TxSB_BREAK      EQU   00001101B   ;BREAK  =  ONHOOK SIGNAL
                                       • • •
   RxSB_BREAK      EQU   TxSB_BREAK
   ```

   NOTE: Refer to Figure 4 for summary of items 7 through 10. An example for one state is shown following item 10 and in Figure 5 (Sheet 2).

7. The **State** statement introduces each new state definition and assigns the name given to the state. This name can be referenced by other states in the table. The name must not have any embedded blanks or spaces and may be of any length, although the first 8 characters should be unique to each state name.

8. The **Event** statement defines a combination of input EVent, Next State, ACtion and Action Data. EVent names are the Ameritec defined "EV_" codes (See Appendix A.1). Next State is user defined, however it must match the name in a State statement somewhere in the Protocol State Table file. Indentation and tabulation may be used to improve readability. If an event is to trigger multiple Actions, multiple event lines may be used, with the Next State value set to NIL; this will cause the following event line to be executed.

9. The **ACtion** names are the Ameritec defined "AC_" codes (See Appendix A.2). Action Data values may be **SIGnal** names (see Appendix A.3). Action Data values may also be binary or numeric constants that are appropriate to the ACtion. These constants may have descriptive names through a preceding list of CONSTANT DEFINITIONS (see item 6). An Action Data value must always be provided. In the case where the Action Name does not require a data value a dummy-value of "0" must be coded.

10. The **StateEnd** statement closes a state definition. The StateEnd statement may be followed by the State name for clarity, but it is not required. The example below shows the definition for one state.

```
State DIAL
   Event EV_SEIZE,      NIL,          AC_SIGNAL,    SIG_ONHOOK
   Event EV_SEIZE,      IDLE,         AC_TxSB,      TxSB_BREAK
   Event RxSB_ANSWER    NIL,          AC_DPSTOP,    0; STOP DP ANALYZER
   Event RxSB_ANSWER    SPEECH,       AC_SIGNAL,    SIG_ANSWER
StateEnd DIAL; repeating the state name is optional
```

11. The group of statements, beginning with **State** and ending with **StateEnd** is repeated for each Protocol State. No particular order is required by the State Machine but the order of the states in the Protocol State Table file should follow the State Transition Diagram for clarity.

12. The **TableEnd** statement closes a Protocol State Table file and must be the statement in the file.

The text in a Protocol State Table definition file is not case sensitive. Both UPPER and lower case may be used to make long names easier to read.The Protocol State Table may be given any filename permitted by DOS; however, the file extension (after the period in the filename) must be .STB.

```
;Example State Table
NT_EMULATE   EQU    7; Max number of Transitions for Emulate
NT_MONITOR   EQU    7; Max number of Transitions for Monitor
INCLUDE      AM8e.INC
;
;CONSTANT DEFINITION
TxSB_BREAK   EQU    00001101B     ;BREAK = ONHOOK SIGNAL
TxSB_MAKE    EQU    00000101B     ;MAKE = OFFHOOK SIGNAL
TxSB_ANSWER  EQU    00000101B     ;ANSWER SIGNAL
RxSB_BREAK   EQU    TxSB_BREAK
RxSB_MAKE    EQU    TxSB_MAKE
RxSB_ANSWER  EQU    TxSB_ANSWER
;
Header0 <TE-E&MK1-IN>,0
;
State INITIALIZE
 Event       EV_INIT,      NIL,          AC_BKBITS,    TxSB_BREAK
 Event       EV_INIT,      NIL,          AC_MKBITS,    TxSB_MAKE
 Event       EV_INIT,      NIL,          AC_IDTBITS,   TxSB_MAKE
 Event       EV_INIT,      NIL,          AC_RXTT,      0
 Event       EV_INIT,      NIL,          AC_UNUSED,    TxSB_BREAK
 Event       EV_INIT,      IDLE,         AC_TxSB,      TxSB_BREAK
StateEnd
;
State IDLE
 Event       EV_SEIZE,     NIL,          AC_BKBITS,    TxSB_BREAK
 Event       EV_SEIZE,     NIL,          AC_SIGNAL,    SIG_OFFHOOK
 Event       EV_SEIZE,     NIL,          AC_SETCNT,    5;5 COUNTS
```

Figure 5 – Protocol State Table Example (1 of 3)

```
Event  EV_SEIZE,        NIL,            AC_SETCNT,        5;5 COUNTS
 Event EV_SEIZE,        NIL,            AC_TIMER,         200;1 SEC EXPIRE
 Event EV_SEIZE,        OUTGOING,       AC_TxSB,          TxSB_MAKE
StateEnd
;
State OUTGOING
 Event EV_DIALTONE,     DIAL,           AC_DIALDP,        0;SEND STRING
DIGIT
 Event EV_TIME,         NIL,            AC_CNTDWN,        0;COUNT DOWN
 Event EV_TIME,         OUTGOING,       AC_TIMER,         200;
 Event EV_COUNT0,       DIAL,           AC_DIALDP,        0;IF TIMER EXPIRE
 Event EV_SEIZE,        NIL,            AC_SIGNAL,        SIG_ONHOOK
 Event EV_SEIZE,        IDLE,           AC_TxSB,          TxSB_BREAK
StateEnd
;
State DIAL
 Event EV_SEIZE,        NIL,            AC_SIGNAL,        SIG_ONHOOK
 Event EV_SEIZE,        IDLE,           AC_TxSB,          TxSB_BREAK
 Event RxSB_ANSWER,     NIL,            AC_DPSTOP,        0;STOP DP ANALYZER
 Event RxSB_ANSWER,     SPEECH,         AC_SIGNAL,        SIG_ANSWER
StateEnd
;
State SPEECH
 Event EV_SEIZE,        NIL,            AC_SIGNAL,        SIG_ONHOOK
 Event EV_SEIZE,        CLR_FORWARD,    AC_TxSB,          TxSB_BREAK
 Event EV_START,        NIL,            AC_TxSB,          TxSB_BREAK
 Event EV_START,        REG_RECALL,     AC_TIMER,         18;90 mS DURATION
 Event RxSB_BREAK,      CLEAR_BACK,     AC_TIMER,         200;CLEAR FORWARD
StateEnd
;
State CLEAR_BACK
 Event EV_TIME,         INDONE,         AC_TIMER,         1;ONHOOK
 Event RxSB_ANSWER,     SPEECH,         AC_SIGNAL,        SIG_ANSWER
StateEnd
;
State REG_RECALL
 Event EV_TIME,         NIL,            AC_SIGNAL,        SIG_G1;REG-RECALL
 Event EV_TIME,         SPEECH,         AC_TxSB,          TxSB_ANSWER
 Event EV_SEIZE,        NIL,                              AC_SIGNAL,
       SIG_ONHOOK
 Event EV_SEIZE,        CLEAR_BACK,     AC_TxSB,          TxSB_BREAK
StateEnd
;
State CLR_FORWARD
 Event RxSB_BREAK,      INDONE,         AC_TIMER,         1;GOTO IDLE
 Event EV_SEIZE,        NIL,            AC_TxSB,          TxSB_ANSWER
 Event EV_SEIZE,        SPEECH,         AC_SIGNAL,        SIG_OFFHOOK
StateEnd
;
State INDONE
 Event EV_TIME,         NIL,            AC_STOPDT,        0;STOP DIAL TONE
 Event EV_TIME,         NIL,            AC_TxSB,          TxSB_BREAK
 Event EV_TIME,         NIL,            AC_SIGNAL,        SIG_ONHOOK
 Event EV_TIME,         NIL,            AC_PRSTOP,        0
 Event EV_TIME,         IDLE,           AC_SIGNAL,        SIG_RELEASE
StateEnd
```

Figure 5 – Protocol State Table Example (2 of 3)

```
;
;MONITOR START HERE "EXCHANGE HOOKS UP TO TX AND SUBSCRIBER HOOKS TO RX"
;
Header4 <TE-E&MK1-IN>,0
;
State INITIALIZE
;THESE INITIALIZATIONS MUST DEFINE THE IDLE STATE FOR FAR & NEAR END
MONITOR
STATE MACHINES
 Event  EV_INIT,         NIL,             AC_BKBITS,        TxSB_BREAK
 Event  EV_INIT,         NIL,             AC_MKBITS,        TxSB_MAKE
 Event EV_INIT,          NIL,             AC_IDTBITS,       TxSB_MAKE
 Event  EV_INIT,         NIL,             AC_RXTT,          0
 Event  EV_INIT,         NIL,             AC_FAREND,        FARIDLE
 Event  EV_INIT,         NIL,             AC_NEAREND,       NEARIDLE
StateEnd
;FAR END MESSAGE HERE
State FARIDLE
 Event  RxSB_ANSWER,     NIL,             AC_DPSTOP,        0;STOP DP ANALYZER
 Event  RxSB_ANSWER,     F_ANSWER,        AC_SIGNAL,        SIG_ANSWER
StateEnd
;
State F_ANSWER
 Event  RxSB_BREAK,      FARIDLE,         AC_SIGNAL,        SIG_RELEASE
StateEnd
;
State NEARIDLE
 Event  RxSB_MAKE,       N_OUTGOING,      AC_SIGNAL,        SIG_OFFHOOK
StateEnd
;

State N_OUTGOING
 Event  RxSB_BREAK,      N_ONHK1?,        AC_TIMER,         30
StateEnd
;
State N_ANSWER
 Event  RxSB_BREAK,      N_ONHK1?,        AC_TIMER,         30
 Event  EV_TIME,         N_CONV,          AC_NIL,           0
StateEnd
;
State N_CONV
 Event  RxSB_BREAK,      N_REG_RECALL,    AC_TIMER,         26;130 ms MAX
StateEnd
;
State N_REG_RECALL
 Event  EV_TIME,         NEARIDLE,        AC_SIGNAL,        SIG_ONHOOK
 Event  RxSB_MAKE,       N_CONV,          AC_SIGNAL,        SIG_G1
StateEnd
;
State N_ONHK1?
 Event  RxSB_MAKE,       N_ANSWER,        AC_TIMER,         200;1SEC
 Event  EV_TIME,         NEARIDLE,        AC_SIGNAL,        SIG_ONHOOK
StateEnd
;
TableEnd
```

Figure 5 — Protocol State Table Example (3 of 3)

3.3.3 Assembling and Linking

Once Protocol State table statements have been entered with the editor or word processor, they must saved as a file with a Protocol State Table filename, followed by the extension name ".STB" (example: E&MK1.STB). This file must then be converted to a downloadable Protocol State Table file. The third party Macro Assembler package is used to accomplish the assembly and linkage that produces the downloadable file (refer to Appendix B.3 and documentation that accompanies the third party package).

If the assembler and linker find an error during assembling and linking, the line containing the error is highlighted and an error message appears. The error messages and their meanings follow:

```
*************************************************************
Protocol table error -- # of Transitions in Emulate too large
*************************************************************
```

This message results from having at least one state in the Emulate State table which has a number of transitions greater than the maximum number defined by the NT_EMULATE statement.

```
************************************************************
Protocol table error -- # of Transitions in Monitor too large
************************************************************
```

This message results from having at least one state in the Emulate State table which has a number of transitions greater than the maximum number defined by the NT_MONITOR statement.

```
************************************************************
Protocol table error -- Header Name over 11 characters
************************************************************
```

This message results from having a Header name longer than the maximum allowable length (11).

After a successful assembly and linkage, a new file is produced with the same Protocol State Table filename and an extension name of ".HEX" (in the sample the file is E&MK1.HEX). This file may then be downloaded into the AM8e as described in the next sub-section. A debugging file with an extension name of ".LST" (e.g. E&MK1.LST is also produced. It will be similar in appearance to the .STB file; however, it will contain additional information generated by the assembler.

NOTE: *The assembly and linking procedure is repeated for each Protocol State Table ".STB" file that has been created.*

3.3.4 Loading Protocol State Tables into AM8e

Following successful assembly and linking, each Protocol State Table file (such as E&MK1.HEX) is ready to download to the AM8e. To download, a third party Communication program such as PROCOMM (Appendix B.2) must be available in the PC.

Protocols must be down-loaded into the three AM8e units that are required for testing the protocols. Directions for downloading to each AM8e are as follows:

1. Verify that the PC is connected to the RS232 port on the AM8e to be loaded.
2. Set the AM8e and PC to the same baud rate (9600 baud is recommended), parity, and word length.
3. At the PC load and execute the communications program (PROCOMM).
4. At the AM8e, press the keys SHIFT, RECALL, A, 9
5. At the PC, press PAGE-up key and select the ASCII option from PROCOMM menu (7=ASCII).
6. When PROCOMM requests a filename, enter the Protocol State Table downloadable file name (Example: E&MK1.HEX).
7. Wait until there is a message that indicates that the loading is complete.
8. Repeat steps 4 through 7 for the next file.

3.3.5 Listing and (optionally) Deleting a Protocol

After all Protocol State Table files have been loaded into an AM8e, the AM8e should be commanded to list the files. As part of the file listing, the AM8e will also provide prompts that will permit the deletion of any unwanted files. (Messages will be displayed at the PC by PROCOMM; which should still be in control of the communications.)

1. At the AM8e, press the keys SHIFT, RECALL, A, 0.
2. The PC will display a list of all files loaded. This list will consist of all Protocol State Table filenames in the AM8e memory (as defined by header statements).
3. If any files are to be deleted, follow the instructions given by prompts that are displayed after the listing is completed.
4. At the PC, exit PROCOMM by pressing ALT-X keys.

NOTE: Similar instructions, including "(DNLD)" and "(EDIT)" remote commands via RS232, are contained in sub-section 7.1.9 of the AM8e Instruction Manual.

3.4 Debugging a Protocol State Table

Debugging a Protocol State Table requires the use of three (3) AM8e units, each of which has the protocol to be debugged in memory.

One unit must be configured to Emulate the Terminal end of the line the second unit must be configured to Emulate the Switch end of the line and the third unit must be configured to Monitor the signals at both ends of the line.

> *NOTE: At the monitoring AM8e, be certain that the Configuration and the RX and TX connections are correct for the end to be monitored. Refer to the small diagram at the bottom of page 16 .*

1. Store some test (AUTO DIAL) numbers in the emulating unit(s) which will be dialing and enable AUTO on both emulating units.
2. Enable EVENTS display on all three units.
3. Enable INPUT on all three units.
4. SEIZE the line with the appropriate unit.
5. Observe the EVENTS displays on all units and verify that the correct sequence of protocol events are displayed for the SEIZE event.
6. Verify also that the event display is appropriate for the emulate unit at the other end.
7. Refer to the timing diagrams and state diagrams you developed and the listingof the state table for the following steps.
8. Use the SIGNALING keys and/or DIALed digit keys to step through each event as appropriate for the outgoing or incoming end.
9. Observe that the appropriate signals are sent to the line and indicated by the events display at each end.
10. Verify also that the appropriate state transition occurs (or does not occur) as shown by changes to the a, b, c and d LED indicators.
11. Enter some invalid events to try to make the protocol fail.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A— DEFINITION TABLES

A.1 EVENT NAMES

Table A-1 provides an alphabetical list of EVent names which have been pre-defined by Ameritec, and should be used when you code events. These EVent names are EQUates contained in the EVENTDEF.INC file.

> ***Note:*** *A listing of this file may show EVent names which are not included in Table A-1. These EVents are for Ameritec use only.*

A.2 ACTION NAMES

Table A-2 provides an alphabetical list of ACtion names which have been pre-defined by Ameritec, and should be used when you code Actions. These ACtion names are EQUates contained in the ACTNDEF.INC file.

> ***Note:*** *A listing of this file may show ACtion names which are not included in Table A-2. These ACtions are for Ameritec use only.*

A.3 SIGNAL NAMES

Table A-3 provides an alphabetical list of SIGnal names which have been pre-defined by Ameritec, and should be used when you Action Data for the AC_SIGNAL actions. These SIGnal names are EQUates contained in the SIGDEF.INC file.

> ***Note:*** *A listing of this file may show SIGnal names which are not included in Table A-3. These SIGnals are for Ameritec use only.*

TABLE A-1 — EVENT DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
EV_ALERTACK	Alert signal from far and received
EV_C5SIG_f1f	2400Hz tone off
EV_C5SIG_f1n	2400Hz tone on
EV_C5SIG_f2f	2600Hz tone off
EV_C5SIG_f2n	2600Hz tone on
EV_C5SIG_f1f2f	2400Hz and 2600Hz tones off
EV_C5SIG_f1f2n	2400Hz and 2600Hz tones on
EV_CALLPG_BEG	Beginning of Call Progress Tone
EV_CALLPG_END	End of Call Progress Tone
EV_COUNT0	Internal decremental counter has reached 0
EV_DIALDP	Indicates dialing Pulse digits has started
EV_DIALMFR1	Indicates dialing MF(R1) digits has started
EV_DIALMFR2	Indicates dialing MF(R2) digits has started
EV_DIALTONE	Received Dial Tone signal
EV_DIALTT	Indicates dialing DTMF digits has started
EV_DIGIT_0	Get digit action has read a 0 from dial buffer
EV_DIGIT_1	Get digit action has read a 1 from dial buffer
EV_DIGIT_2	Get digit action has read a 2 from dial buffer
EV_DIGIT_3	Get digit action has read a 3 from dial buffer
EV_DIGIT_4	Get digit action has read a 4 from dial buffer
EV_DIGIT_5	Get digit action has read a 5 from dial buffer
EV_DIGIT_6	Get digit action has read a 6 from dial buffer
EV_DIGIT_7	Get digit action has read a 7 from dial buffer
EV_DIGIT_8	Get digit action has read a 8 from dial buffer
EV_DIGIT_9	Get digit action has read a 9 from dial buffer
EV_DIGIT_A	Get digit action has read a A from dial buffer
EV_DIGIT_B	Get digit action has read a B from dial buffer
EV_DIGIT_C	Get digit action has read a C from dial buffer
EV_DIGIT_D	Get digit action has read a D from dial buffer
EV_DIGIT_E	Get digit action has read a E from dial buffer
EV_DIGIT_F	Get digit action has read a F from dial buffer
EV_DIGIT_KP	Get digit action has read a * from dial buffer

TABLE A-1 — EVENT DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
EV_DIGIT_ST	Get digit action has read a # from dial buffer
EV_DIGIT_UNK	Get digit action has read an unknown digit
EV_DONE	General Done
EV_DTMF_d	Received DTMF digit d; d = 1 to 15 (see note a)
EV_END	General End; generated by "+" in autodial strings
EV_EQUAL	Register A or B compare is true
EV_FAIL	General Fail
EV_FLASH	Received Hook Flash
EV_GREATER_THAN	Register A or B compare is greater than Reg. A' or B'
EV_INIT	Initialize State Machine
EV_LESS_THAN	Register A or B compare is less than Reg. A'or B'
EV_MFR1_d	Received MFR1 digit d; d = 1 to 15 (see note b)
EV_MFR1_OFF	Indicates MFR1 digits are disabled
EV_MFR2_BWD_OFF	Indicates MFR2 backward digits are disabled
EV_MFR2_FWD_NAK	Indicates no handshaking signal received on MFR2 forward digit
EV_MFR2_FWD_OFF	Indicates MFR2 forward digits are disabled
EV_MFR2BWD_d	Received MFR2 backward digit d; d = 1 to 15 (see note c)
EV_MFR2FWD_d	Received MFR2 forward digit d; d = 1 to 15 (see note c)
EV_NEWDIALSTR	Indicates a "/" was read from the dial buffer
EV_NODIGITS	All avialable digits sent
EV_NOT_EQUAL	Mar Reg. A Register A compare not equal to Mar Reg. B
EV_OFFHOOK	Received OFF HOOK signal
EV_ONHOOK	Received ON HOOK signal
EV_PERROR	Parameter is outside of allowed range.
EV_REVERSE	REV key or Reverse key pressed
EV_RING	RING key pressed
EV_RxSB_abcd	Received signaling bits; abcd are bit values for protocol states (see note d)
EV_SEIZE	Seize key pressed
EV_SETDSP	Initialize the Digital Signal Processors
EV_SZACK	Send SEIZE acknowledgement
EV_TIME	An internal timer has expired (from AC_TIMER, _TIMER1,or PrTimer)

TABLE A-1 — EVENT DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
EV_TOG_D	Indicates a "d" was read out of the dial buffer
EV_TOG_M	Indicates a "m" was read out of the dial buffer
EV_TOG_P	Indicates a "p" was read out of the dial buffer
EV_TOG_R	Indicates a "r" was read out of the dial buffer
EV_TONELN_BEG	Beginning of Single Tone line side
EV_TONELN_END	End of Siin;le Tone line side
EV_TONESW_BEG	Beginning of Single Tone with side
EV_TONESW_END	End of Single Tone switch side.
EV_TSTDIG_FALSE	Compare of digit and parameter are not equal
EV_TSTDIG_TRUE	Compare of digit and parameter are equal
EV_UNBALANCE	Received line unbalanced signal
EV_UNDEFTONE	Indicates an unknown tone was detected (ie, voice)
EV_WAIT	Indicates a "=" was read out of the dial buffer
EV_WAITANS	Indicates a "N" was read out of the dial buffer
EV_WAITCP	Indicates a "X" was read out of the dial buffer
EV_WAITDT	Indicates a "P" was read out of the dial buffer
EV_WAITST	Indicates a "T" was read out of the dial buffer
EV_WAITUT	Indicates a "?" was read out of the dial buffer
EV_WAITWINK	Indicates a "W" was read out of the dial buffer
EV_WAITY	Indicates a "Y" was read out of the dial buffer
EV_WAITZ	Indicates a "Z" was read out of the dial buffer

Notes:

a. 1 to 9 = digits 1 to 9; 10 = 0 (operator), 11 = *, 12 = #, 13 = A, 14 = B, 15 = C, 0 = D (see 4.9.3 in AM8e Instruction Manual).

b. 1 to 9 = digits 1 to 9; 10 = 0 (operator), 11 = KP, 12 = ST, 13 = ST3P, 14 = STP, 15 = ST2P (see 4.9.4 in AM8e Instruction Manual).

c. See figure A-8 in AM8e Instruction Manual for MF(R2) backward and forward digits that 1-15 represent.

d. "abcd" are the FAR end (Rx) bits, and = 0000 - 1111 depending on protocol state received. EV_RxSB_abcd may also be coded EV_SB_abcd. "abcd" may also be code as a statename (ie, ANSWER, MAKE, BREAK, etc.) provided the statenames have been defined in the state table by CONSTANT DEFINITION equates (EQU). Reference section 3.3.2 item 6.

TABLE A-2 — ACTION DEFINITIONS FOR AM8e

ACTION CODE	ACTION DATA	EXPLANATION
AC_ACKMFR2	1 - 15 (notes a & b)	Send a single MFR2 backward (ACK) digit
AC_ADAPTER	0 = no adapter 1 = E&M adapter	Select 300056 E&M Adapter
AC_BKBITS	xxxx0000B - xxxx1111B (notes c & g)	Define DP BREAK bits
AC_C5SIG	0=quiet, 1=2400Hz 2=2600Hz, 3=2400Hz & 2600Hz	Send C5 signaling tones to line (SF tones)
AC_CNTDWN	0 (none)	Decrement countdown register
AC_DIALDP	0 (none)	Transmit dial buffer using dial PULSE
AC_DIALMFR1	0 (none)	Transmit dial buffer using MF(R1)
AC_DIALMFR2	0 (none)	Transmit dial buffer using MF(R2)
AC_DIALSTR	0 (none)	Copy autodial string to dial buffer
AC_DIALTT	0 (none)	Transmit dial buffer using DTMF
AC_DIGDP	1 - 15	Transmit digit using dial PULSE
AC_DIGMFR1	1 - 15	Transmit digit using MF(R1)
AC_DIGMFR2	1 - 15	Transmit digit using MF(R2)
AC_DIGTT	0 - 15	Transmit digit using DTMF
AC_DPSTOP	0 (none)	Stop analyzing dial PULSE dialing
AC_DWNCNT	-128 to 127	Decrement countdown register
AC_ERROR	0 (none)	Report error to host
AC_FAIL	0 (none)	Send fail signal to cp, send report to host
AC_FAREND	FARIDLE	Load far end (Rx) idle state for monitor
AC_FASTA	0 - 15	Enable fast-a bit for Italy
AC_GETDIGIT	0 (none)	Read digit from dial buffer
AC_IDTBITS	xxxx0000B - xxxx1111B (see note g)	Define dial PULSE inter digit time bits
AC_MKBITS	xxxx0000B - xxxx1111B (see note g)	Define dial PULSE MAKE bits
AC_MOVDIGPTR	-60 to 60	Dynamically moves digit pointer in dial buffer to left (-) or right (+). Limits are the first and last digit in the menu
AC_NEAREND	NEARIDLE	Loads NEAR-end (Tx) idle state for monitor
AC_NIL	0 (none)	No Action

TABLE A-2 — ACTION DEFINITIONS FOR AM8e

ACTION CODE	ACTION DATA	EXPLANATION
AC_RECLCNT	0 - 15	Recall count from register
AC_RXC5	0 (none)	Set analyzer for C5 signaling
AC_RXMFR1	0 (none)	Set analyzer for MF(R1)
AC_RXMFR2_B	0 (none)	Set analyzer for MF(R2) backward digits
AC_RXMFR2_F	0 (none)	Set analyzer for MF(R2) forward digits
AC_RXTT	0 (none)	Set analyzer for DTMF
AC_SENDDIGTT	0 - 15	Send DTMF digit until turned off
AC_SENDTONE	0, 1, 2	Transmits single tone, 0 - turns off, 1 - turns on, 2 - pulses
AC_SETCNT	0 - 255	Set down counter register to starting count
AC_SELDIALSTR	0 - 9	Copy auto dial string to dial buffer
AC_SETDIGPTR	1 - 255	Set digit pointer
AC_SIGNAL	see SIG_ definitions	Report signal to unit
AC_SPCMF	n	Special commands
AC_SSB	xxxx0000B - xxxx1111B	Transmit specified signaling bits
AC_STARTDT	0 (none) (see note h)	Start dial tone
AC_STOPDT	0 (none)	Stop dial tone
AC_STOPTIMER	0 (none)	Stop general purpose timer
AC_STOPTIMER1	0 (none)	Stop extra timer
AC_STORCNT	0 - 15	Save count in register
AC_TESTDIGIT	0 - 15	Compare digit in dial buffer to parameter
AC_TESTEVNT	0 - 255	Generate the equivalent event
AC_TIMER	1 - 255 5mS time tiks	Start general purpose timer
AC_TIMER1	1 - 255 5mS time tiks	Start extra timer
AC_TxSB	xxxx0000B - xxxx1111B	Transmit specified signaling bits
AC_UNUSED	xxxx0000B - xxxx1111B	Define signaling bits for unused channel
AC_WAIT	0 (none)	Wait 3000mS and generate event EV_DONE
AC_WRITE	xxxx0000B - xxxx1111B	Transmit specified signaling bits

Notes:

a. Numeric data is entered directly, ie, AC_ACKMFR2, 5.

b. See figure A-8 in AM8e Instruction Manual for MF(R2) backward and forward digits that 1-15 represent.

c. xxxx0000B - xxxx1111B binary data is often coded by equivalent state names defined by CONSTANT DEFINITION equate (EQU) statements at the beginning of the state table. This data represents signalling bits and is defined by the protocol requirements, ie, AC_BKBITS, TxSB_BREAK (reference section 3.3.2 item 6).

d. (none)" indicates no action data; however, 0 must be coded into state table; ie, AC_DIALDP, 0.

e. - 9 = digits 1 - 9; 10 = 0 (operator), 11 = KP, 12 = ST, 13 = ST3P, 14 = STP, 15 = ST2P (see table 4.9.4 in AM8e Instruction Manual).

f. - 9 = digits 1 - 9; 10 = 0 (operator), 11 = *, 12 = #, 13 = A, 14 = B, 15 = C, 0 = D (see table 4.9.3 in AM8e Instruction Manual).

g. Must precede use of AC_DIALDP or AC_DIGDP.

h. The SIG signal definitions communicate signal values back to the AM8e. Most will cause an event display and/or some message to be displayed/printed by the unit, ie, AC_SIGNAL, SIG_OFFHOOK causes an "H" to be displayed.

TABLE A-3 — SIGNAL DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
SIG_ABANDON	Ringing abandoned.
SIG_ALERT	ALERT (incoming Call).
SIG_ANSWER	FAR-end answer (OFFHOOK) detected (H blinking).
SIG_BADACK	Bad (unexpected) backward MFR2 digit.
SIG_BDELAY	Begin delay dial (X event).
SIG_BDELAYb	Begin delay dial (X event blinking).
SIG_BPROG	Begin call progress tone (X event).
SIG_BREV	Begin Polarity reversal (P event).
SIG_BREVb	Begin Polarity reversal (P event blinking).
SIG_BRING	Begin Ring (B event).
SIG_BRINGb	Begin Ring (R event blinking).
SIG_BTONE	Begin Tone (T event).
SIG_BTONEb	Begin Tone (T event blinking).
SIG_BUSY	Called subscriber is BUSY (B-3).
SIG_DISCONNECT	Caller disconnected.
SIG_DONE	General done.
SIG_DP	Dial Pulse digit.
SIG_DPb	Dial pulse digit blinking.
SIG_EDELAY	End Delay Dial (x event).
SIG_EDELAYb	End Delay Dial (x event blinking).
SIG_EPROG	End Progress tone (x event)
SIG_EREV	End Polarity reversal (p event)
SIG_EREVb	End Polarity reversal (p event blinking)
SIG_ERING	End ring (r event)
SIG_ERINGb	End Ring (r event blinking).
SIG_ERR	General error signal
SIG_ETONE	End Tone (t event)
SIG_ETONEb	End Tone (x event blinking)
SIG_FAIL	FAIL (often due to Timeout).
SIG_FLASH	FLASH (F event)
SIG_FLASHb	FLASH (F event blinking)
SIG_FLASHd	Duplication of FLASH
SIG_Gn	Generic Signal n (where n=1 - 10) = !, $, %, (,), +, =, @, [,] events)
SIG_Gnb	Generic Signal n (where n=1-10 = !, $, %, (,), +, =, @, [,] events blinking)

TABLE A-3 — SIGNAL DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
SIG_MF	MFR1 Analyzer
SIG_MFR2B	MFR2 dial backward analyzer
SIG_MFR2F	MFR2 dial forward analyzer
SIG_MTRPULSE	Meter-pulse (M event if MFR2)
SIG_MTRPULSEb	Meter-pulse (M event blinking if MFR2)
SIG_OFFHOOK	OFFHOOK (H event)
SIG_OFFHOOKd	Duplication of OFFHOOK
SIG_ONHOOK	ONHOOK (h event).
SIG_ONHOOKd	Duplication of ONHOOK.
SIG_RELEASE	Line has been released (h event blinking).
SIG_SIEZE	SEIZE from network
SIG_STARTDIAL	START DIAL (Dial tone, wink, etc.)
SIG_STARTDIALb	START DIAL (blinking).
SIG_SZAK	SEIZE acknowledged from MFR2 originate channel.
SIG_TT	Touch Tone (DTMF) analyzer
SIG_UNBAL	Unbalanced line
SIG_WINK	WINK (W event)
SIG_WINKb	WINK (W event blinking)

Notes:

a. The Signal definitions are Action data used with an AC_SIGNAL action and must be written in the State Table as AC_SIGNAL, SIG_... (e.g.: AC_SIGNAL, SIG_BADACK). In drawing State Diagrams, the "," and "SIG_" may be omitted for brevity (e.g.: AC_SIGNAL BADACK.

NOTICE

The following Actions and Events are only available in AM8e's equipped with software Revision 20F or higher. Older machines can be upgraded. Please contact your Ameritec Sales office for price and availability.

NEW EVENT DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
EV_CALLPG_BEG	Beginning of Call Progress Tone
EV_CALLPG_END	End of Call Progress Tone
EV_DIGIT_0	Get digit action has read a 0 from dial buffer
EV_DIGIT_1	Get digit action has read a 1 from dial buffer
EV_DIGIT_2	Get digit action has read a 2 from dial buffer
EV_DIGIT_3	Get digit action has read a 3 from dial buffer
EV_DIGIT_4	Get digit action has read a 4 from dial buffer
EV_DIGIT_5	Get digit action has read a 5 from dial buffer
EV_DIGIT_6	Get digit action has read a 6 from dial buffer
EV_DIGIT_7	Get digit action has read a 7 from dial buffer
EV_DIGIT_8	Get digit action has read a 8 from dial buffer
EV_DIGIT_9	Get digit action has read a 9 from dial buffer
EV_END	General End; generated by "+" in autodial strings
EV_EQUAL	Register A or B compare is true
EV_GREATER_THAN	Register A or B compare is greater than Reg. A' or B'
EV_LESS_THAN	Register A or B compare is less than Reg. A'or B'
EV_NEWDIALSTR	Indicates a "/" was read from the dial buffer
EV_NOT_EQUAL	Mar Reg. A Register A compare not equal to Mar Reg. B
EV_PERROR	Parameter is outside of allowed range.
EV_TOG_D	Indicates a "d" was read out of the dial buffer
EV_TOG_M	Indicates a "m" was read out of the dial buffer
EV_TOG_P	Indicates a "p" was read out of the dial buffer
EV_TOG_R	Indicates a "r" was read out of the dial buffer
EV_TONELN_BEG	Beginning of Single Tone line side
EV_TONELN_END	End of Siin;le Tone line side
EV_TONESW_BEG	Beginning of Single Tone with side
EV_TONESW_END	End of Single Tone switch side.

NEW EVENT DEFINITIONS FOR AM8e

EVENT CODE	EXPLANATION
EV_TSTDIG_FALSE	Compare of digit and parameter are not equal
EV_TSTDIG_TRUE	Compare of digit and parameter are equal
EV_UNDEFTONE	Indicates an unknown tone was detected (ie, voice)
EV_WAIT	Indicates a "=" was read out of the dial buffer
EV_WAITANS	Indicates a "N" was read out of the dial buffer
EV_WAITCP	Indicates a "X" was read out of the dial buffer
EV_WAITDT	Indicates a "P" was read out of the dial buffer
EV_WAITST	Indicates a "T" was read out of the dial buffer
EV_WAITUT	Indicates a "?" was read out of the dial buffer
EV_WAITWINK	Indicates a "W" was read out of the dial buffer
EV_WAITY	Indicates a "Y" was read out of the dial buffer
EV_WAITZ	Indicates a "Z" was read out of the dial buffer

NEW ACTION DEFINITIONS FOR AM8e

ACTION CODE	ACTION DATA	EXPLANATION
AC_DIALSTR	0 (none)	Copy autodial string to dial buffer
AC_DWNCNT	-128 to 127	Decrement countdown register
AC_GETDIGIT	0 (none)	Read digit from dial buffer
AC_RECLCNT	0 - 15	Recall count from register
AC_SELDIALSTR	0 - 9	Copy auto dial string to dial buffer
AC_SENDDIGTT	0 - 15	Send DTMF digit until turned off
AC_SPCMD	n	Special commands
AC_STORCNT	0 - 15	Save count in register
AC_TESTDIGIT	0 - 15	Compare digit in dial buffer to parameter
AC_TESTEVNT	0 - 255	Generate the equivalent event
AC_WAIT	0 (none)	Wait 3000mS and generate event EV_DONE
AC_WRITE	xxxx0000B - xxxx1111B	Transmit specified signaling bits

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B — REQUIRED AM8e SOFTWARE AND INSTALLATION

B.1 AMERITEC SUPPLIED FILES

There are several files supplied on the PC-DOS formatted Protocol Table Development disk (Part Number 9480080B) supplied with this document.

All supplied files on the disk should be copied into the working directory of the PC, then the disk should be stored in a safe place.

The working directory also must contain the required third-party Communications (see D.2) and Assembler and Linker software (see D.3).

Following is a list and description of files supplied by Ameritec

File Name	Description
AM8e.BAT	BATCH file to prepare Protocol Table file (assemble/link).
AM8e.INC	
EVENTDEF.INC	.INC Files are definition files that are automatically used during Table preparation. Refer to Tables A-1, A-2 and
ACTNDEF.INC	A-3 for the EVents, Actions and SIGnals for AM8e that are contained in these files.
SIGDEF.INC	
E&MK1.STB	Source Table for Demonstration Protocol (see Figure 5).
E&MK1.LST	EDiagnostic Source Table Listing produced by Assembler/Linker.
E&MK1.HEX	Assembled and Linked Protocol which may be loaded into the AM8e by the Communications Program.

B.2 REQUIRED THIRD-PARTY COMMUNICATION PROGRAM

The PROCOMM® Communications Program is recommended, and examples in this manual and the Instruction Manual are based on its use. This program may be obtained from:

DATASTORM TECHNOLOGIES, INC.
P.O. Box 1471
Columbia, MO 65205 U.S.A.
Phone 314-443-3282

B.3 REQUIRED THIRD PARTY ASSEMBLER/LINKER TOOLS

To prepare Protocol Tables, third party Assembler/Liner software tools must be used. The recommended tools are SASM and SLINK assembler/linker from Softools, Inc.

The Softools programs should be installed in accordance with their installation procedures on the working disk. The Ameritec supplied files (see D.1) should then be copied into the SASM directory.

Order these tools directly from **Softools, Inc.** or their distributors as follows:.

SASM, SREF, SLINK and SLIB Toolset
Softools, Inc.
9200 Route 108, Suite 120
Columbia, Maryland 21045 USA
(410) 964-9903 Voice
(410) 964-9905 FAX/BBS

i SYSTEM GmbH
Einsteinstrasse 5
W-8060 Dachau
Germany
48-081-311686
49-081-3114024 FAX

THAU Computer
Lowenstrasse 21
8953 Dietikon
Switzerland
41-1740-4105
41-1740-1567 FAX

Electro Optics Pty.
P.O. Box 67
Kenthurst, NSW 2156
Australia
61-2654-1873
61-2654-1539 FAX

APPENDIX C — USER DEFINED TEXT FEATURE (P/N 948001100r)

The user-defined display of single-digit event characters on the first line of the display and detail event text on the second line of the display is a new feature provided by the User-Defined Text Feature for the protocol development package.

This feature is provided on the Protocol Table Development disk, part number 948001100r (where r=revision letter, subject to latest update). This disk is loaded into the user's PC instead of Part Number 9480080r referred to in Appendix B.1.

The following applies to Protocols developed (or adapted from existing protocols) for the User Defined Text.

1. The NT_EMULATE and NT_MONITOR statements which define the maximum number of transitions per state in the Emulate and Monitor protocols are not required. (Refer to 3.3.2, steps 2 and 3 and lines 2 and 3 oZf Figure 5). **Before re-compiling existing protocol state tables, these statements should be removed.**

   The number of transitions for each state are automatically calculated. This will result in protocols which occupy a smaller amount of memory in the AM8e. It should also eliminate the possibility of the protocol table error messages on state transitions described in 3.3.3.

2. Two new Action Definitions are added to those shown in table A-2.

Action Code	Action Data	Explanation
AC_SIGTXT	see step 3, below	Send the event display to the AM8e LCD from the NEAR end.
AC_SIGTXTb	see step 3, below	Send the event display to the AM8e LCD from the FAR end. Display is blinking.

Example statements using AC_SIGTXT and AC_SIGTXTb:

Within EMULATE table:

```
Event EV_SEIZE, NEXT_STATE, AC_SIGTXT, TEXT_SEIZE
```

Within MONITOR table:

```
Event RxSB_0001, NEXT_STATE, AC_SIGTXTb, TEXT_MSEIZE
```

CAUTION: The statements that define action-data for emulate states must be different from those that define monitor states.

3. The Action Data for the AC_SIGTXT and AC_SIGTXTb Action Codes should be defined at the end of each Emulate and Monitor state table that includes them. They cannot be shared. The Action data must be defined as a STATE in the following format.

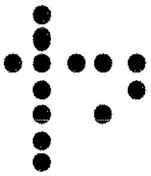
```
        State user_display
              DB char, 'text', 255;
        StateEnd
```

Where:

`user_display` = name for the Action Data (example: TEXT_SEIZE)

`char` = A single alphanumeric character enclosed in single quotes (' ') which will be displayed as an event. Character set may be any displayable/printable numeric, alphabetic or special character (such as . , - + = , etc.) (? and> should not be used).
Example: 'S'

= The binary equivalent of any of the AM8e characters (including those that are non-alphanumeric) shown on Page 37.
Example: 11010100B for the following character:
Trailing B must be included.



= The hexadecimal equivalent of any character.
Example: 0D4H for the preceding binary number.
Leading zero and trailing H must be included.

= The decimal equivalent of any character.
Example: 212 instead of 11010100B or 0D4H.

`'text'` = Event detail text. This text must be less than 30 characters
Example: 'Seize'. The single-quotes (') must be included.
Any displayable/printable text may be used.

`255;` = Terminator and end-command. These must be included.

4. To avoid confusion, when selecting a single character for a protocol event, the User should refer to Page 1-6 in the AM8e Instruction Manual for event characters and event detail texts which are already defined within the AM8e software.

   The protocol writer may include any character. However, this could result in the same character display for two different events. IT IS IMPORTANT FOR THE writer of the protocol to clearly explain the meaning of all displayed characters; both through the event detail text, and in written documentation which accompanies the protocol.

5. EXAMPLE of statements added to Emulate and Monitor portions of State Table coding to provide User-defined display events are on Figure C-1. A complete example protocol table is provided in Figure 5 of this manual.

```
;Example partial State Table to illustrate new event display statements.
;
;note that NT_EMULATE and NT_MONITOR statements are not used)
INCLUDE AM8e.INC
              •
              •
              •
Header0 <as required>, 0
              •
              •
              •
State IDLE
              •
              •
              •
 Event EV_SEIZE, NEXT_STATE, AC_SIGTXT, TEXT_SEIZE
StateEnd
              •
              •
              •
State TEXT_SEIZE
      DB 'S', 'Seize', 255;
; or DB 01010011B, 'Seize', 255;
; or DB 053H, 'Seize', 255;
; or DB  83, 'Seize', 255;
StateEnd
;Begin Monitor Table
;
Header 4 <same as for 0>, 0
              •
              •
              •
State FARIDLE
              •
              •
              •
 Event RxSB_0001, NEXT_STATE, AC_SIGTXTb, TEXT_MSEIZE
StateEnd
              •
              •
              •
State TEXT_MSEIZE
      DB 'S','Seize', 255;
StateEnd
;
TableEnd
```

Event included in state table (statemen could be added to example in Figure 5.)

State for Action Data for AC_SIGTXT, above, at end of table.

Alphanumeric S surrounded by ' '
Binary equivalent of 'S'
Hexadecimal equivalent of 'S'
Decimal equivalent of 'S'

Statements for Monitor Table. Note that Action Data must have a different name.

Figure C-1 — Protocol State Table Statements to Provide User Defined Event Display

The following figure shows 5 x 7 dot-matrix characters which may be displayed by the AM8e. For the single digit event, characters may be programmed as Action Data as described on page 35 (3.) and examples in Figure C-1. Eight-bit character codes in the first column and those that generate characters that are greater than seven dots high should not be used. The way that characters are represented in the printed reports depends on the characteristics of the device connected to the RS232 port.

High-Order 4 bit Low-Order 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
××××0000	CG RAM (1)		0	@	P	`	p		ー	タ	ミ	α	p
××××0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
××××0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
××××0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
××××0100	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
××××0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
××××0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
××××0111	(8)	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
××××1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
××××1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
××××1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
××××1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
××××1100	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
××××1101	(6)	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
××××1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
××××1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

AM8e Display Characters

APPENDIX D— PROTOCOL SIZE

In AM8e units with firmware version 20C or later, it is possible to create an emulate protocol larger than the maximum 4080 size limit that currently exists with firmware version prior to 20C.

To make a larger emulate protocol the file AM8e.INC needs to have the start of the monitor changed. Change the Header4macro line DS 4000H-(End0-BEGIN) to another value such as DS 6000H-(End0-BEGIN). The value selected needs to be a multiple of 1000. (Example: Acceptable values are 5000H, 6000H or 7000H)

The absolute maximum size of a protocol is 7936 (1F00H). Any protocol larger than this maximum size will not run correctly.

THIS PAGE INTENTIONALLY LEFT BLANK

PROTOCOLS CURRENTLY AVAILABLE for the AM8e PCM/VF CALL ANALYZER

INTRODUCTION

A single Protocol Package may be ordered from Ameritec by protocol number suffixed by "-PP", e.g. 9280001-PP. A protocol package consists of a Protocol Manual and a software disk. Some protocol packages may be restricted in their availability. As shown in the list, some protocols are always provided in groups. If any one protocol is ordered, the others listed are provided at no additional charge.

Protocols ordered with an AM8e will be installed and ready for selection, however, the protocol packages will be supplied also. Protocol packages that are ordered later must be loaded into the AM8e from the software disk.

When the protocol is being used, the Protocol Manual must be consulted for use of the AM8e Signaling Keys and Displays (beyond those described in the AM8e Instruction Manual).

Protocol manuals 9280001-MN, 9280002-MN and 9280033-MN are supplied with the AM8e Protocol Development Kit (P/N 19-0004). The Protocol Development Kit also includes the Protocol Table Development Guide, this Addendum, the Protocol Table Development disk (P/N 94800110x) and a three-ring binder.

Revisions to this Addendum will be issued as more protocols are added. Revisions to the protocol software or documentation will be made when necessary to assure proper operation and description. Refer to the protocol package for the revision letter that applies to the supplied protocol.

Note: This Addendum to the AM8e PCM/VF Call Analyzer Protocol Development Guide should be inserted following the Appendices in the Guide. It is also available separately as P/N 18-0126.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Oct 25, 95	9280001	R2-CCITT-1D (9280002)	**R2 CCITT -** CCITT R2 signaling with backward digits generated automatically. Emulate or monitor both ends. Not country or switch vendor specific.
Oct 25, 95	9280002	R2-CCITT-2D (9280001)	**R2 CCITT Autodial -** CCITT R2 signaling with backward digits taken from the autodial memory. The operator must enter into one of the autodial menus the exact sequence of backward digits to be sent and select this menu into one of the autodial memories. Emulate or monitor both ends. Not country or switch vendor specific.
Mar 11, 94	9280003 *Note*	LP-SUB-DP-B (9280004, 9280005, 9280006)	**Loop Start DP -** Loop Start, 2-wire, DP signaling carried over a 2Mbps multiplexor system. Emulate or monitor the subscriber end. Based on Telcom New Zealand document TCTA 0056 which was supplied to Ameritec under a nondisclosure agreement.
Mar 11, 94	9280004 *Note*	LP-SUB-TT-B (9280003, 9280005, 9280006)	**Loop Start TT -** Loop Start, 2-wire, TT (DTMF) signaling carried over a 2Mbps multiplexor system. Emulate or monitor the subscriber end. Based on Telcom New Zealand document TCTA 0056 which was supplied to Ameritec under a nondisclosure agreement.
Mar 11, 94	9280005 *Note*	LP-SUB-MF-B (9280003, 9280004, 9280006)	**Loop Start MFR1 -** Loop Start, subscriber end, 2-wire, MFR1 signaling carried over a 2Mbps multiplexor system. Emulate or monitor the subscriber end. Based on Telcom New Zealand document TCTA 0056 which was supplied to Ameritec under a nondisclosure agreement.
11 Mar 94	9280006 *Note*	LP-2W-EXG-B (9280003, 9280004, 9280005)	**Loop Start DP/TT/MFR1 -** Loop Start, exchange end, 2-wire, DP/TT(DTMF)/MFR1, signaling carried over a 2Mbps multiplexor system. Emulate or monitor the exchange end. Based on Telcom New Zealand document TCTA 0056 which was supplied to Ameritec under a nondisclosure agreement.

Note: Because the information used to prepare this protocol was provided under a nondisclosure agreement, this protocol may only be supplied to the entity providing the information. For example: 9280003 may only be supplied to Telcom New Zealand.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Mar 11, 94	9280007 *Note*	NZ-PSTN-1B (9280008, 9280014, 9280015)	**R2 New Zealand PSTN -** R2 signaling used between exchanges in the New Zealand Public Switched Telephone Network with backward digits generated automatically. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on Telcom New Zealand Engineering Report ER/TN 0072(AN) which was supplied to Ameritec under a nondisclosure agreement.
Mar 11, 94	9280008 *Note*	NZ-PSTN-2B (9280007, 9280014, 9280015)	**R2 New Zealand PSTN Autodial -** R2 signaling used between exchanges in the New Zealand Public Switched Telephone Network with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on Telcom New Zealand Engineering Report ER/TN 0072(AN) which was supplied to Ameritec under a nondisclosure agreement.
May 4, 94	9280009	R2-FIN-1B (9280010)	**R2 Finland -** R2 signaling used between exchanges in the Finland National Telephone Network with backward digits generated automatically. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on R2ENGLISH 4.10.1990 specification.
May 4, 94	9280010	R2-FIN-2B (9280009)	**R2 Finland Autodial -** R2 signaling used between exchanges in the Finland National Telephone Network with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on R2ENGLISH 4.10.1990 specification.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Mar 11, 94	9280011	AUST-01-B (9280012)	**R2 Australian -** R2 signaling used between exchanges in the Australian Public Switched Telephone Network with backward digits generated automatically. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on Telcom Australia document TP 00536 S.
Mar 11, 94	9280012	AUST-02-B (9280011)	**R2 Australian Autodial -** R2 signaling used between exchanges in the Australian Public Switched Telephone Network with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on Telcom Australia document TP 00536S.
Dec 7, 93	9280013	R2-TOGLE-1C	**R2 CCITT Toggle -** CCITT R2 signaling with backward digits generated automatically. Emulate or monitor both ends. Allows dialing to be toggled to DTMF during the conversation time. This allows testing of voice mail systems that accept R2 network signaling followed by DTMF user signaling.
Mar 11, 94	9280014 *Note*	NZ-PSTN-3A (9280007, 9280008, 9280015)	**R2 New Zealand -** R2 signaling used between exchanges in the New Zealand Public Switched Telephone Network with backward digits generated automatically. Emulate or monitor both ends. Terminate accepts dialing before seize acknowledge is sent. Based on Telcom New Zealand Engineering Report ER/TN 0072(AN) which was supplied to Ameritec under a nondisclosure agreement.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Mar 11, 94	9280015 *Note*	NZ-PSTN-4A (9280007, 9280008, 9280014)	**R2 New Zealand Autodial -** R2 signaling used between exchanges in the New Zealand Public Switched Telephone Network with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Terminate accepts dialing before seize acknowledge is sent. Based on Telcom New Zealand Engineering Report ER/TN 0072(AN) which was supplied to Ameritec under a nondisclosure agreement.
Dec 12, 93	9280016	P7-INC PBX (9280018)	**Sweden SS P7 PBX IN -** Signaling system P7 used in Sweden to connect a PBX to a public exchange. Emulates or monitors the PBX end for calls directed from the public exchange to the PBX. Based on SVENSK Standard SS 63 63 32.
Dec 12, 93	9280017	P7-OUT PBX (9280019)	**Sweden SS P7 PBX OUT -** Signaling system P7 used in Sweden to connect a PBX to a public exchange. Emulates or monitors the PBX end for calls directed from the PBX to the public exchange. Based on SVENSK Standard SS 63 63 31.
Dec 12, 93	9280018	P7-INC EXG (9280016)	**Sweden SS P7 EXG IN -** Signaling system P7 used in Sweden to connect a PBX to a public exchange. Emulates or monitors the public exchange end for calls directed from the public exchange to the PBX. Based on SVENSK Standard SS 63 63 32.
Dec 12, 93	9280019	P7-OUT EXG (9280017)	**Sweden SS P7 EXG OUT -** Signaling system P7 used in Sweden to connect a PBX to a public exchange. Emulates or monitors the public exchange end for calls directed from the PBX to the public exchange. Based on SVENSK Standard SS 63 63 31.
Dec 12, 93	9280020	P8-INC PBX (9280021)	**Sweden SS P8 PBX IN DDI -** Signaling system P8 used on Sweden to connect a PBX to a public exchange providing direct dialing in (DDI) capability. Emulates or monitors the PBX end. Based on SVENSK Standard SS 63 63 33.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Dec 12, 93	9280021	P8-INC EXG (9280020)	**Sweden SS P8 EXC IN DDI -** Signaling system P8 used on Sweden to connect a PBX to a public exchange providing direct dialing in (DDI) capability. Emulates or monitors the public exchange end. Based on SVENSK Standard SS 63 63 33.
Dec 12, 93	9280022	KUWAIT-01A (9280023)	**R2 Kuwait PSTN -** R2 signaling used between exchanges in the Kuwait Public Switched Telephone Network with backward digits always sent as 1. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on REGISTER SIGNALING LME MFC, Kuwait (1914-ANS 53102).
Dec 12, 93	9280023	KUWAIT-02A (9280022)	**R2 Kuwait PSTN Autotdial -** R2 signaling used between exchanges in the Kuwait Public Switched Telephone Network with backward digits backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on REGISTER SIGNALING LME MFC, Kuwait (1914-ANS 53102).
Dec 12, 93	9280024	GERMANY-01A (9280025)	**R2 German PSTN -** R2 signaling used between exchanges in the German Public Switched Telephone Network with backward digits always sent as 1. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on LINE SIGNALING System based on CCITT R2 PCM Traffic Type I (1/1914-ANS 43401) and National R2 MFC (1914-ANS 53201).

DATE	PROTOCOL#	TITLE	DESCRIPTION
Dec 12, 93	9280025	GERMANY-02A (9280024)	**R2 Germany PSTN Autodial -** R2 signaling used between exchanges in the German Public Switched Telephone Network with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on LINE SIGNALING System based on CCITT R2 PCM Traffic Type I (1/1914-ANS 43401) and National R2 MFC (1914-ANS 53201).
Dec 12, 93	9280026	NOR-R2-01A (9280027)	**R2 Norwegian PSTN -** R2 signaling used between exchanges in the Norwegian Public Switched Telephone Network with backward digits always sent as 1. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on TECHNICAL SPECIFICATIONS for Public Paging System based on RDS published by the Norwegian Telecommunications Administration.
Dec 12, 93	9280027	NOR-R2-02A (9280026)	**R2 Norwegian PSTN Autodial -** R2 signaling used between exchanges in the Norwegian Public Switched Telephone Network with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Seize acknowledge is required before dialing begins. Based on TECHNICAL SPECIFICATIONS for Public Paging System based on RDS published by the Norwegian Telecommunications Administration.
Mar 11, 94	9280028	R1-DTMF-E&M (9280029, 9280030, 9280031, 9280032)	**R1 DTMF Dialing with E&M Signaling -** DTMF dialing with E&M signaling using the A signaling bit. Used on a PCM line between a DMS-100 and a Smith-Jones multiplexor (DSC1). The R1 designation does not refer to R1 dialing. Based on Clear Communications DAL Technical Specification-Issue 1.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Dec 12, 93	9280029	EX-LOOP-OUT (9280028, 9280030, 9280031, 9280032)	**LOOP Start for PBX to CO -** DTMF dialing with signaling bits similar to R2. Used on a PCM line between a DMS-100 and a Smith-Jones multiplexor (DSC1). The application is for LOOP START for PBX to CO calls. Based on Clear Communications DAL Technical Specification-Issue 1.
Dec 12, 93	9280030	R2-DTMF-DDI (9280028, 9280029, 9280031, 9280032)	**R2 DTMF Direct Dialing -** DTMF dialing with signaling bits similar to R2. Used on a PCM line between a DMS-100 and a Smith-Jones multiplexor (DSC1). The application is for Direct Dialing In. Based on Clear Communications DAL Technical Specification-Issue 1.
Dec 12, 93	9280031	EX1-LOOP-IN (9280028, 9280029, 9280030, 9280032)	**LOOP Start for DO to PBX (R1) -** DTMF dialing with signaling bits similar to R1. Used on a PCM line between a DMS-100 and a Smith-Jones multiplexor (DSC1). The application is for LOOP START for CO to PBX calls. Based on Clear Communications DAL Technical Specification-Issue 1.
Dec 12, 93	9280032	EX2-LOOP-IN (9280028, 9280029, 9280030, 9280031)	**LOOP Start for CO to PBX (R2) -** DTMF dialing with signaling bits similar to R2. Used on a PCM line between a DMS-100 and a Smith-Jones multiplexor (DSC1). The application is for LOOP START for CO to PBX calls. Based on Clear Communications DAL Technical Specification-Issue 1.
Aug 19, 93	9280033	CCITT#5	**CCITT Signaling #5 -** CCITT Signaling System Number 5. Tones of 2400/2600 Hz are used for line signaling. Dual-tone signals similar to MFR1 are used for register signaling. Signaling bits are not used. Based on CCITT Q.140–Q.164.
Dec 12, 93	9280034	TURKEY-DDI	**R1 Turkey DDI -** MFR1 dialing with signaling bits similar to E & M wink start. Used in Turkey for DDI operation for CO to PBX calls.
Dec 12, 93	9280035	MS-PTT-OUT (280036, 9280037, 9280038)	**Malaysian PTT Pulse E&M LC to GSC -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the Malaysian PTT protocol. The signaling is Pulse E&Mm LC to GSC.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Jan 30, 96	9280036	PTT-SW (9280035, 9280037, 9280038)	**Malaysian PTT Pulse E&M GSC to LC -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the Malaysian PTT protocol as described in documents provided by Pernas. The signaling is Pulse E&M from GSC to LC.
Dec 21, 95	9280037	PTT-OUT-PCM (9280035, 9280036, 9280038)	**Malaysian PTT Non-Pulse E&M LC to GSC -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the Malaysian PTT protocol as described in documents provided by Pernas. The signaling is Non-Pulse E&M in PCM interface from LC to LC, and LC to GSC.
Dec 12, 93	9280038	PTT-SW-PCM (9280035, 9280036, 9280037)	**Malaysian PTT Non-Pulse E&M GSC to LC -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the Malaysian PTT protocol as described in documents provided by Pernas. The signaling in Non-Pulse E&M from LC to GSC.
Feb 15, 96	9280039C	TELMEX	**TELMEX -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to emulate or monitor the signaling on a 2.0MB CAS trunk circuit which conforms to the Telmex protocol used in Mexico.
Jan 10, 94	9280040	E&M-ADAPT	**E&M with Signaling Adapter -** This protocol is designed for use with the AM8e E&M Signaling Adapter Model 300056. The protocol uses either delay dial or wink start and accomodates all E&M & connection types shown on the 300056 for basic E&M signaling. Additionally, it incorporates MFR1 dialing.
Dec 12, 93	9280041	C5-ANALOG	**C5 Signaling Analog -** Designed for CCITT Signaling System number 5 on an anlog line. This protocol requires the 300056 AM8e E&M signaling adapter. Tones of 2400/2600 Hz are used for line signaling. Dual-tone signals similiar to MFR1 are used for register signaling. Signaling bits are not used. Based on CCITT >Q.140-Q.164.
Dec 12, 93	9280042	QUALCOM-1	**Australian TP202 -** DTMF or Pulse dialing with signaling bits and call progress as specified by Qualcom in Australia document TP 00202.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Dec 25, 95	9280043	SIGBIT	**Manual ABCD Bit** - Designed to give the user Real Time Information as to the current signaling bit pattern of the selected channel. With each allowable bit transaction, the AM8e will display detailed information as to the incomming bit pattern and the relative time stamp. All possible "abcd" bit patterns are shown in the events display of the AM8e. With the use of this protocol, the user can receive and record any and all signaling bit patterns, view the real-time pattern on the front panel's LED's, capture a running record with the use of a PC and a Terminal Emulation Software, and view the events on the AM8e's Events Display. The protocol also allows the user to send any desired signaling pattern when in Emulate Mode.
Apr 15, 93	9280044	MERC-MCL (9280045)	**Mercury Signaling SS5002 between PBX and MCL** - This protocol is designed to allow the user to emulate or monitor the signaling on a line conforming to Mercury Signaling Document SS 5002 for call origination between a PABX and MCL. It is an asymmetrical protocol and should be accompanied by 9280045 (MERC-RING) which is used for the ring-down side.
Apr 15, 93	9280045	MERC-RING (9280044)	**Mercury Signaling SS5002 between PBX and MCL Call Ring-Down** - This protocol is designed to allow the user to emulate or monitor the signaling on a line conforming to Mercury Signaling Document SS5002 for call ring-down from MCL to PABX. It is an asymmetrical protocol and should be accompanied by 9280044 (MERC-MCL) which is used for the call origination side.
Aug 30, 93	9280046	STO-8000	**STO-8000** - This protocol is designed to allow the user to emulate or monitor the signaling on a line conforming to STO-8000 as described by documents supported by AT&T. It incorporates semi-compelled MFR2 register signaling.
Dec 7, 93	9280047	BRAZIL-AA	**Brazil Double Attendant** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the Brazilian Double Attendant Protocol. It incorporates Manual DTMF Dialing after the Answer signal.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Dec 7, 93	9280048	R2-TT-MN	**R2 CCITT Toggle to TT** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using R2 dialing for China then toggling to TT after Answer Protocol. It incorporates Manual DTMF Dialing after the Answer signal.
Dec 7, 93	9280049	R2-TT-CH	**R2 CCITT CH Toggle to TT** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using R2 dialing then toggling to TT after Answer Protocol as described by documents supplied by Octel Communications. It incorporates Manual DTMF Dialing after the Answer signal. See special note on page two concerning Manual DTMF.
Aug 20, 94	9280050	INDONESIA	**E&M T1 Adapter DP or MFR1 -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the Indonesian R2 Signaling Protocol. The MFR2 FWD and BWD digits must be programmed into the AUTODIAL buffer prior to the beginning of the call.
Dec 7, 93	9280051	E1-E&M-51	**E1/R1 E&M with ANI** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the E1/R1 E&M with ANI Protocol. It incorporates Manual DTMF Dialing after the Answer signal.
Dec 7, 93	9280052	E&M-ADP-DP	**E&M Adapter DP or MFR1 -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the 30-0056 E&M Signaling Adapter using E&M signaling with either MFR1 or Dial Pulse Dialing.
Nov 23, 93	9280053	R2_BOLIVIA	**R2 Bolivia -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the R2-BOLIVIA Protocol. This protocol incorporated user defined text
Dec 25, 95	9280054	R2-T1-01	**R2 CCITT T1 Adapter -** CCITT R2 signaling with backward digits generated automatically. Emulate or monitor both ends. Not country or switch vendor specific. This protocol has been adapted from 9280001 for use with the 30-0056 AM8e T1 Adaptor. It is functionally the same as 9280001, the only changes made were to accommodate the T1 Interface.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Feb 29, 96	9280055	R2_T1-02	**R2 CCITT T1 Adapter Autodial -** CCITT R2 signaling with backward digits taken from the autodial memory. The operator must enter the exact sequence of backward digits to be sent into one of the autodial memories and select this memory. Emulate or monitor both ends. Not country or switch vendor specific. This protocol has been adapted from 9280002 for use with the 30-0068 AM8e T1 Adaptor. It is functionally the same as 9280002, the only changes made were to accommodate the T1 Interface.
Feb 29, 96	9280056	E&M-T1-DP	**E&M T1 Adapter DP or MFR1** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the 300068 AM8e T1 Adaptor using E&M signaling with either MFR1 or Dial Pulse Dialing.
Sep 15, 94	9280057	INDONESIA-2	**R2 INDONESIA (Indosat) -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the R2 protocol used by Indosat in Indonesia.
Sep 15, 94	9280058	Fetex-150	**China Fetex-150 -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to emulate or monitor the signaling on a line connected to the Fetex-150 exchange in China.
Mar 10, 95	9280059	ARGENTINA-1	**R2 ARGENTINA -** CCITT R2 signaling with backward digits generated automatically. Emulate or monitor both ends. Designed for signaling in Argentina. Conforms to NTP-297-3401-1802.
Sep 15, 95	9280060	ADP-IMM-TT	**ADP-IMM-TT -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the E&M Signaling Adaptor, P/N 30-0056, using E&M signaling with either DTMF or Dial Pulse Dialing and Immediate Start.
Sep 15, 95	9280061	CENTIGRAM-1	**Centigram Pulse E&M -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the protocol as described in documents provided by Centigram. The signaling is Pulse E&M.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Sep 15, 95	9280062	COLISEE-ST	**Colisee with State Changes -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using the COLISEE-STATE (Colisee with state changes) protocol and DTMF Signaling.
Mar 10, 95	9280063	P8-INCX-63	**Sweden SS P8 Exchange -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using signaling system P8 in Sweden which connects a PBX to a public Exchange. It has direct inward dialing (DID) capability. This protocol Emulates or Monitors the EXG end. DTMF digits are used instead of Dial Pulse. This is an "assymmetrical" protocol which is complemented by protocol 9280064 (P8-INCP-64), which Emulates or Monitors the PBX end.
Mar 10, 95	9280064	P8-INCP-64	**Sweden SS P8 PBX -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with equipment using signaling system P8 in Sweden which connects a PBX to a public Exchange. It has direct inward dialing (DID) capability. This protocol Emulates or Monitors the EXG end. DTMF digits are used instead of Dial Pulse. This is an "assymmetrical" protocol which is complemented by protocol 9280064 (P8-INCP-64), which Emulates or Monitors the Exchange end.
Mar 10, 95	9280065	LDLZ	**Argentina PSTN Loop Start, DP, LMO Signaling MP -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Loop Start, Dial pulse, LMO signaling with Meter Pulse.
Aug 10, 95	9280066	LDBZ	**Argentina PSTN Loop Start, DP, BPO Signaling MP -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Loop Start, Dial pulse, BPO signaling with Meter Pulse.
Aug 10, 95	9280067	LMBZ	**Argentina PSTN Loop Start, MFR2, BPO Signaling MP -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Loop Start, MFR2, BPO signaling with Meter Pulse.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Aug 10, 95	9280068	EMMZ	**Argentina PSTN E&M, MFR2, MP -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using E&M signaling, MFR2 dialing with Meter Pulse.
Aug 10, 95	9280069	EMDZ	**Argentina PSTN E&M, DP, MP -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using E&M signaling, Dial Pulse dialing with Meter Pulse.
Aug 10, 95	9280070	LMLN-LMBN	**Argentina PSTN Loop Start, MFR2, LMO or BPO** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Loop Start, MFR2, LMO or BPO signaling without Meter Pulse.
Aug 10, 95	9280071	LDLN-LDBN	**Argentina PSTN Loop Start, MFR2, LMO or BPO** - This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Loop Start, Dial Pulse, LMO or BPO signaling without Meter Pulse.
Aug 10, 95	9280072	EMMANA	**Argentina PSTN Analog E&M, MFR2 -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Analog E&M signaling with MFR2 dialing.
Aug 10, 95	9280073	EMDANA	**Argentina PSTN Analog E&M, DP -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Argentina PSTN on lines using Analog E&M signaling without MFR2 dialing.
Aug 10, 95	9280074	T1-Korea	**Korea T1 Adapter -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the 30-0068 AM8e T1 adapter using E&M signaling with MFR2 dialing. It allows for DTMF after answer and is designed for use by Korea Mobile Telecom Paging.
Feb 28, 95	9280075	E1-Korea	**Korea PSTN -** This protocol is designed to allow the AM8e PCM/VF Call Analyzer to interface with the Korean PSTN. It uses MFR2 signaling with DTMF after answer. It is designed for use by Korea Mobile Telecom Paging.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Feb 28, 95	9280076	SSDC-5A	**SSDC-5A Signaling -** AM8e T1 adapter using E&M signaling with equipment using SSDC-5A signaling. It provides for immediate start or optional wink start signaling. It provides for MFR1 or Dial Pulse dialing. It conforms to POR-1182.
Mar 25, 96	9280082	TES_R2C_TBA	**TES_R2C_TBA -** This protocol was written for Hughes Network Systems Telephony Earth Station R2MF Compelled Type B Signaling Interface.
Mar 25, 96	9280083	E&M-T1-TOG	**E&M-T1-TOG -** This protocol is designed to allow the AM8e to interface with the 30-0071AM8e T1 Adapter using E&M signaling. Dialing can toggle to DTMF from Dial Pulse or MFR1 type dialing.
May 31, 96	9280084	CHINA #1-L	**CHINA NO. 1 LOCAL -** This protocol will allow the AM8e to emulate or monitor the interexchange signaling required for Local Calls as either the local exchange or the distant exchange.
May 31, 96	9280085	CHINA NO. 1 TOLL	**CHINA NO. 1 TOLL -** This protocol will allow the AM8e to emulate or monitor the interexchange signaling required for Toll Calls, national or international which employ semi-compelled (pulsed) MFC signaling.
Dec 1, 97	9280088	PNG_R2-8e	**PNG_R2-8e -** This protocol will allow the AM8e to emulate the local or distant exchange signaling required for Local calls as well as inter-exchange signaling required for national and international calls which employ semi-compelled MFC signaling.
Dec 1, 97	9280089	NG-E&M-R2_A	**NG-E&M-R2_A -** This protocol will allow the AM8e to emulate the local or distant exchange signaling required for Local calls as well as inter-exchange signaling required for national and international calls which employ semi-compelled MFC signaling.
Nov 18, 97	9280092	R2EM_ADPT_2	**R2EM_ADPT_2 -** This protocol will allow the AM8e to interface with equipment on Analog 2/4 wire E&M trunk circuits using E&M signaling protocol with MFR2 Compelled dialing emulation.
Nov 18, 97	9280093	R2EM_ADPT_2	**R2EM_ADPT_2 -** This protocol will allow the AM8e to interface with equipment on Analog 2/4 wire E&M trunk circuits using E&M signaling protocol with MFR2 Compelled dialing emulation.

DATE	PROTOCOL#	TITLE	DESCRIPTION
Dec. 16, 97	9280096	GR303	**GR303 -** This protocol will allow the user to monitor the Line Signaling (ABCD bits) and the Address Signaling (DTMF/MF/Dial Pulse digits) between a Remote Digital Terminal (RDT) and an Integrated Digital Terminal (IDT) on the selcted channel.