1 What is AM4PM?

AM4PM is a simple answering machine program for OS/2 2.x PM and ZyXEL 1496. I wrote it for personal use because I could not find a suitable program. When I started to write it I had the following objectives:

- To be able to distinguish between voice/fax/modem.
- To be able to use external programs for modem and fax handling. I did not want to write these.
- To be able to listen to recorded messages from a remote phone.
- To make a program that could easily be modified without having to recompile my code.
- To make the program give up the COM port if another program needs it.

Other things on the wish list were: call my personal pager or a special phone number if an important message is recorded, change the greeting message from a remote phone or at a specified time, use some external program such as ZIP to compress the voice files.

Not all these features have been implemented yet, but be patient; some day they will. I must also admit the documentation might need some further work.

PLEASE note that this documentation does not describe how the ZyXEL modem works. To be able to modify the program a good knowledge of the voice AT commands is required.

AM4PM is a public domain program. It is free to use and free to distribute as long as unmodified versions of all the original programs and this document are included. Please report any grammatical or spelling errors in this text to me. How else can I learn to write correct English?

I take no responsibility whatsoever for any damage AM4PM can cause. Neither do I guarantee the functionality of the program.

I might not be able to fix all bugs and answer all questions about AM4PM, but I would like to be informed about problems and ideas on further development.

The program has been tested on a 486 clone with a 16550A running OS/2 2.1 XR06200B and a ZyXEL 1496+ ROM v6.11a.

Thomas Olsson
Göteborg, Sweden
email d8olsth@dtek.chalmers.se

2 What is included in this version?

Most of the functionality is programmed in REXX. AM4PM adds some functions to the REXX language that make it possible to control the modem. The REXX scripts that comes with AM4PM gives the following functionality:

- The program can be set to answer after a predefined number of rings. It will answer after the programmed number of times minus one if there are messages waiting. It will then play a greeting message. A short beep indicates the end of the greeting message and the beginning of the recording. If the user presses ‘9’ a more elaborate greeting message will be played.
- If a voice is heard during recording the message will be saved. The end of the message is determined either by 7 s of silence or after a specific maximum time.
- If a fax calling tone or a DTMF ‘5’ is heard during playback or recording AM4PM will try to see if it is a fax or a modem. This will also happen if no voice has been heard for the first 7 s of recording.
- If a DTMF ‘*’ is heard during playback or recording AM4PM will enter command mode. A command consists of a DTMF ‘*’ and a number followed by a DTMF ‘#’. The only available command will make it possible to listen to recorded messages from a remote phone.
- The COM port can be released by starting *AM4PMCMD* with the command ‘R’. The port can also be released by pressing the ‘Release’ button in the PM window. AM4PM will try to open the port again after 30 s and thereafter every ten seconds.

3 The functions of the program

The behaviour of AM4PM is very much determined by REXX scripts. These can be modified by the cunning user. The following description applies to the original scripts that are generated by the setup program.

The main window presents a list of recorded messages. The date of the recording is displayed together with the length of the message and optionally CID information.

To display the popup menu press the right mouse button or sh-F10. The manu alternatives available are:

Play

By selecting one or more messages in the list and then selecting *Play* the messages will be played in the modem's internal speaker.

Delete

By selecting one or more messages in the list and then selecting *Delete* the messages will be deleted. If *Delete* is pressed during playback the messages will be played first.

File | Play...

A file dialog will be displayed when *File | Play...* is selected. When the *Play* button is pressed the selected file will be played in the speaker of the modem.

File | Record...

A file dialog will be displayed when *File | Record...* is selected. When the *Record* button is pressed the recording of a message from the microphone connected to the modem will start.

Abort

By selecting *Abort* the current playback or recording is aborted. If a delete operation is delayed due to a playback the delete operation will also be aborted.

Stop

The current playback or recording will be stopped when *Stop* is selected.

Release

Selecting *Release* will release the COM port for 30 s. The program will then try to reopen the COM port every ten seconds.

This can also be accomplished by starting the program *AM4PMCMD* with the parameter ‘R’.

4 How to get started

- Put at least all the following files in one directory:

  AM4PM.EXE

  AM4PM.MSG

  *.AMC

- Start AM4PM.EXE. The status bar at the bottom of the screen will display ”Initializing...” and then ”Waiting for call”. If any other message is written instead check the settings by selecting *Settings...* from the menu.

- Make recordings of the following messages:

Name	Description
GREETING.ZVD	Played when a call is answered.
GREET2.ZVD	Played when someone has pressed a correct code.
NOMSG.ZVD	Played to tell that there are no recorded messages.
DELMSG.ZVD	Played to tell that a message has been deleted.
ENDMSGS.ZVD	Played to tell that are no more waiting messages.
INFO.ZVD	Played when the caller presses ‘9’. This is a longer version of the greeting message.
SOMEMSGS.ZVD	Played when the caller asks for recorded message to tell that there are messages waiting.
CHGMSG.ZVD	Played when the caller is about to record a temporary greeting message.
RESTMSG.ZVD	Played when the caller is restoring the original greeting message.
GREETORG.ZVD	The original greeting message.
0-19, 20, 30, 40, 50, 60, 70, 80, 90.ZVD	Played to tell the date and time of the recorded messages. The algorithm used will not work with all languages. It will work with English and Swedish, but some other languages, such as German and Danish, have a different word order. French will almost work, but some numbers will fail (31 (trente et un) can not be told by just appending 30 and 1).
NM1-NM12.ZVD	The name of the months.

- Change the remote codes and the password in RING.AMC.

- If you want to use OS2YOU make sure it is in your path. If FaxWorks is to be used to receive faxes see to that FxRcv is in the path (see special note for how to use FaxWorks).

To play recorded messages from a remote phone press ‘*’ on the phone during the greeting message. This will be acknowledged by a beep. Enter your code withing five seconds (the default code is ‘123’) and press ‘#’. The message GREET2.ZVD will be played. If there are no messages NOMSG.ZVD will be played. If there are messages SOMEMSGS.ZVD is played and then all the messages preceeded by date and time. After each message a short beep will be heard. During playback of a message or within five seconds after the beep you can press any of the following keys:

1 Skip to the next message

2 Delete the current message

3 Play the message from the start again

If none of these keys are pressed the next message will be played automatically. If there are no more messages ENDMSGS.ZVD will be played.

5 The REXX scripts

AM4PM uses REXX scripts for many tasks. This makes it easy to modify and costomize the operation of AM4PM. The scripts called normally have the file extension *AMC*. Some functions are added to the REXX language to control the modem.

RING.AMC

This script is called when AM4PM receives ‘VCON’ or a string starting with ‘RING’ from the modem.

The first parameter is the string that was recieved before this script was started. If it is ‘VCON’ the hook is already off. If it is a string starting with ‘RING’ the second parameter tells the number of rings that has been detected so far for this call. When the time between two successive rings are greater than five seconds they are considered to belong to different calls.

The script normally checks if it should answer the phone. When it answers a greeting message is played and then the recording of a message is started. The script can also start a fax or modem communiction program if it detects a fax or a modem. Features for listening to recorded messages from a remote phone might also be included.

To save hard disk space this script can call a compression program such as *LHARC* to compress the recorded messages.

If CID is supported this script can attach a suitable text as an EA with the name .SUBJECT to the recorded file. The text will be displayed in the list of messages.

The first parameter can be used together with the distinctive ring feature of some modems to make AM4PM answer in different ways

PLAY.AMC

This script is called when the user have selected one or more messages from the list in the main window in AM4PM and then selected *Play* from the menu.

Input parameter is the name of the file to play.

Normally this script just plays the message through the speaker in the modem, but if the message is stored in a compressed format it is necessay to uncompress it first.

Other functionality in this script can be to play the message on another device such as a MMPM/2.

RECORD.AMC

This script is called when the user selects *File | Record...* from the menu.

Input parameter is the name of the file to record.

Normally this script just starts a recording from the microphone connected to the modem. The recording continues until *Stop* is selected.

Other functionality in this script can be to record the message from another device such as MMPM/2.

DELETE.AMC

This script is called when the user have selected one or more messages from the list in the main window in AM4PM and then selected *Delete* from the menu.

Input parameter is the name of the file to delete.

This script just deletes the file.

CID.AMC

This script is called when AM4PM receives CID information from the modem. The CID messages are identified by the string ‘TIME’ at the beginning.

Input parameter is the the string starting with ‘TIME’.

The CID information can be displayed in the *last event* field of AM4PM.

INIT.AMC

This script is called every time AM4PM want to reninitialize the modem.

Normally this script resets the modem (ATZ), disables echo (ATE0) and sets answer mode (AT+FCLASS=8).

USER.AMC

This script is called when a user action is sent to AM4PM. This can be done by running AM4PMCMD with the parameter ‘u’.

Input parameter is the name of the user action.

Normally this script does not exist.

This script can be used handling special switchboard hardware. See note below.

Other scripts

Other scripts can be called by using the program AM4PMCMD (see separate description below).

This can be used when another program wants to access to functions of AM4PM. For example if a voice message should be delivered at a specific time the alarm application can call AM4PMCMD to start a script. That script can then call the number and play the message.

6 The REXX functions

handle = AMGetHotComm()

AM4PM stops using the COM port until AMReleaseHotComm is called. The port is not closed. A file handle to the open port is returned and can be used as a parameter to other programs such as REFAX/2 and OS2YOU.

The settings for data format and handshaking are restored to the same values they had when AM4PM was started.

CALL AMReleaseHotComm

AM4PM reinitializes the port and starts listeningr input again.

port = AMCloseComm()

AM4PM closes the COM port to make it available to other programs. The port can be reopened by calling AMOpenComm. The port will also be reopened if the script is ended.

CALL AMOpenComm

AM4PM opens the port and sets the port mode settings. If the port can not be opened this function will retry to open it every ten seconds.

CALL AMLog *text*

Writes *text* to the log file.

filename = AMOpenRecFile(*mode)*

Creates a unique file to be used for storing messages. The filename is returned. The file is created in the current directory and is called M*nnnn*.ZVD where *nnnn* is a unique number.

The *mode* parameter is used for setting the voice mode in the modem by sending ‘AT+VSM’. The following modes are supported on a ZyXEL 1496E+ with ROM v6.00:

1.	CELP	9600 baud. High compression. Fairly good quality.
2.	bit ADPCM	19200 baud. Bad quality.
3.	bit ADPCM	28800 baud. Good quality.

Reception of a message is started by calling AMStartRec and ended by calling AMEndRec.

The file must be closed by calling AMCloseRecFile.

CALL AMCloseRecFile

Closes the message file opened by calling AMOpenRecFile.

This function also attaches an EA with the name AM4PM.LEN to the file that tells the length of the message in seconds. This EA should be of type BINARY and stored as a 32-bit unsigned integer. The length information will be displayed in the list of messages. If this EA is not found the length will be calculated as the file size divided by 2000.

dlecode = AMPlayFile(*filename [, dlecodes]*)

Plays the file *filename*. If a DLE code is returned from the modem the playback is aborted and the code is returned. A string of acceptable DLE codes can be specified as the second parameter. If no DLE code is detected a blank is returned. At least the following codes can be returned:

‘0’-‘9’, ‘*’, ‘#’	A DTMF code was received.
c	A fax calling tone was received.
b	A busy tone was received.
e	Modem calling tone (ROM >= 6.11)
d	dial tone (ROM >= 6.11)

This function cal also return a user action sent by AM4PMCMD.

CALL AMStartRec

Starts recording. The message is saved in the current message file opened by calling AMOpenRecFile. The recording is stopped by calling AMEndRec and the file is closed by calling AmCloseRecFile.

The function AMWaitDLECode is normally called between this call and AMEndRec to see if any DLE codes are heard.

CALL AMEndRec

Ends the recording. The file must be closed by calling AMCloseRecFile.

CALL AMStartListenDLECode *[mode]*

Starts recording but does not save the message to a file. This is used to listen for DLE codes. The listening must be stopped by calling AMEndListenDLECode. If no *mode* parameter is given mode 2 is used. See AMOpenRecFile for a description of the modes.

The function AMWaitDLECode is normally called between this call and AMEndListenDLECode to see if any DLE codes are heard.

Some modems reports DLE codes as soon they voice mode is selected. (ZyXEL 1496 ROM >= 6.10 S39.6=1) In that case this function and AMEndListenDLECode is not needed.

CALL AMEndListenDLECode

Ends the listening for DLE codes.

dlecode = AMWaitDLECode(*[timeout [,dlecodes]]*)

Listen for DLE codes during recording started either by AMStartRec or AMStartListenDLECode. The optional time out parameter is given in ms. Default is 2 min. The optional second parameter is a set acceptable DLE codes. At least the following codes can be returned:

‘0’-‘9’, ‘*’, ‘#’	A DTMF code was received.
c	A fax calling tone was received.
b	A busy tone was received.
q	Quiet detected. Some voice was heard at the beginning, but now there is silence.
s	Silence detected. No voice has been heard at all.
e	Modem calling tone (ROM >= 6.11)
d	dial tone (ROM >= 6.11)

This function cal also return a user action sent by AM4PMCMD.

dceresp = AMSendW(*text*)

Sends the text *text* followed by CR/LF to the modem and then waits for a response. The response is returned. Note that this functions waits for a LF to be received from the modem.

CALL AMSendAT *text [, timeout, [retries]]*

The *text* consists of pairs of commands and expected result codes. The result codes are enclosed in curly brackets. Example: ‘ATZ{OK}ATE0{OK}’

If the expected result code is not received before the time out error message 20 will be written to the log file. The command is then resent. This is repeated the number of times given in the *retries* parameter. The default timeout is 1 minute and the default number of retries is 1.

dceresp = AMGetMoreDCEResp(*[timeout]*)

If more then one string is returned from the modem as a response to AMSendW or AMSendAT this function can the be used to retrieve them.

Default timeout is 1 minute.

rc = AMStartProg(*command*)

Starts the program found at the beginning of the string *command*. The program name must be given without path and extension. The environment variable *path* will be used when searching for the file. All text following the program name will be used as a parameter to the program.

The return code of the program is returned.

The program will be started minimized and closed when the session ends unless AM4PM is in debug mode.

CALL AMSetStateText *text*

Sets the text in the AM4PM window under the heading ‘*State’* to *text*.

The text is reset to *‘Waiting for call’* when the rexx program terminates.

CALL AMSetLastEventText *text*

Sets the text in the AM4PM window under the heading ‘*Last event*’ to *text*.

The text is set to *‘Abort’* when *Abort* is pressed.

value = AMReadIni(*key*, *default*)

Retreives a value from the file AM4PM.INI.

CALL AMWriteIni *key*, *value*

Sets a value in the file AM4PM.INI.

CALL AMOutP *port, value*

Send the *value* to the *port.* AM4PM must have IOPL.

value = AMReadIni(*port*)

Retreives a value from *port*. AM4PM must have IOPL.

CALL AMTrimFileEnd(*bytes*)

Removes *bytes* from the end of the current message file.

7 The REXX variables

AM4PM adds some variables to the REXX script language.

vAMRings

The number of rings that should be heard before AM4PM answers. This value is stored in the AM4PM.INI file and can be changed by the script and the new value will be stored.

vAMMsgs

The number of messages recorded.

vAMMsgPat

The file search pattern used for finding message files. This value is stored in the AM4PM.INI file.

8 AM4PMCMD

This program is used for controlling AM4PM from other programs. It sends messages to AM4PM. All messages are queued.

Command	Description
R	Release COM port for 30 s. AM4PM will then try to reopen the port every 10 s. AM4PMCMD waits until the port is released.
C *command*	Tells AM4PM to run the script *command*. AM4PMCMD does not wait for the script to execute. All text after the first space in *command* will be sent as a parameter to the script. For example the command `AM4PMCMD C ”play.amc greeting.zvd”` will play the message *greeting.zvd*.
U *event*	The if a script is running when this call is made the *event* string can be retrieved by using AMPlayFile or AMWaitDLECode. Otherwise the script USER.AMC is started.

A return code of zero indicates a successful call.

9 The file AM4PM.INI

Key	Description
Rings	The number of rings before answering. This value is stored in the Rexx variable vAMRings.
Com	The name of the COM port to open.
Baud	The baud rate to use when communicating with the modem.
Titles	Set to ‘Y’ if the field titles should be displayed in the list of messages.
ATHangUp	String sent to modem to hang up. Default: ”AT+VLS=0{OK}”
ATVoiceMode	String sent to modem to set voice mode. Default: ”AT+VSM=%u{OK}”
ATVoiceTX	String sent to modem to start voice playback. Default: ”AT+VTX{CONNECT}”
ATVoiceRX	String sent to modem to start recording. Default: ”AT+VRX{CONNECT}”
ATVoiceEndTX	String sent to modem to end playback. Default: ”\x10\x03{VCON}”
ATVoiceCancelTX	String sent to modem to abort playback. Default: ”\x10\x14{VCON}”
MessagePattern	Search pattern used for searching for message files. This value is stored in the Rexx variable vAMMsgPat. Default: ”M*.ZVD”
FileExtension	File extension used for message files. Default: ”ZVD”
FileMode	Mode for recorded files. ”Z” for ZyXEL files and ”G” for generic AM4PM files. Default: ”Z”

If any of the values above is changed by an AM4PM script AM4PM should be restarted.

The AM4PM.INI file can also be used to store values of other keys used by scripts written for AM4PM. No version of AM4PM will ever use a key starting with an ‘x’. This can be used to make sure that there is no conflict with future versions.

10 Debugging the Rexx scripts

Output from the Rexx *SAY* and *TRACE* statements is redirected to the queue *rxsio*. The program *QWRITE* can be used to monitor the output. For good performance remove the *SAY* and *TRACE* statements when debugging is finished.

11 Functions I will add some day

- Better error handling.
- Keep track of messages that have never been listened to and mark them in some way in the list.
- The icon displayed when the program is minimized should be changed when there are messages waiting.
- Make it possible to use AM4PM with Rockwell chip set modems.
- An installation program that generates the AMC script files.
- Support for MMPM/2.

SOM objects for WPS.