

8 The Cassette Manager.

The Cassette Manager deals with reading files from and writing files to tape. These operations can either be performed on a character by character basis or on a whole file at once. There is no hardware support for the cassette, even the timing for reading and writing bits is performed by software.

The format of data on the tape is described in great detail. This will only be of academic interest to most users. More general information can be found in sections 8.4 onwards. In the case of V1.1 machines or the CPC464 with a DDI-1 fitted, a |TAPE command will have to be used to access the Tape Operating System.

8.1 File Format.

A file on tape is split into blocks each with a header record and a data record containing up to 2K (2048) bytes of data. The cassette motor which is under software control is turned off between each file block to allow time to process the data read or to generate the data to be written. The motor start-up gap also serves to separate the blocks from each other.

The general format of a block is as follows:

Motor Start-up	File header record	File data record
-------------------	-----------------------	---------------------

However, the first and last blocks of a file have an extra pause before and after them respectively, to separate files on the tape. Their formats are thus:

First block:

Motor Start-up	Pre-file gap	File header record	File data record
-------------------	-----------------	-----------------------	---------------------

Last block:

Motor Start-up	File header record	File data record	Post-file gap
-------------------	-----------------------	---------------------	------------------

There is a strong distinction between the file header record and the file data record. The header record is written using one synchronisation character (#2C) and the data record with another (#16). This means that when the Cassette Manager is searching for a file header it is impossible for it to find a file data record by mistake, and vice versa. See 8.2 below for the use of the synchronisation characters.

8.2 Record Format.

A record can contain any number of data bytes from 1 to 65536. The data is split into segments each of which is 256 bytes long. The last segment is padded out to 256 bytes with zeros when writing if necessary. When reading a record any extra bytes are ignored although they are accumulated into the CRC.

The layout of a record is as follows:

Leader	Segment 1	Segment N	Trailer
--------	-----------	-------	-----------	---------

There are N segments where $256 * N$ is the length of data (plus padding) to be written.

A file header record always contains one segment; a file data record contains from one to eight segments (usually 8 segments).

a. Leader

At the start of all records a leader is written which has the following layout:

Pre-record gap	2048 one bits	Zero bit	Sync byte
----------------	---------------	----------	-----------

The leading gap is there to ensure the failure of any attempt to synchronise on the end of a preceding record or on data that was on the tape and that has been over-recorded.

The long sequence of one bits is used to calculate the speed at which the data was written and hence to calculate the threshold value used to distinguish one bits from zero bits.

The single zero bit is used to mark the impending end of the leader and is also used to determine whether the recording has been inverted (see section 8.3).

The synchronisation byte is there to help prevent spurious synchronisation on sequences of bits such as might be found in a record. If an incorrect value for the sync byte is found then an attempt has been made to synchronise on the middle of a record or on the wrong type of record. This byte is used to distinguish header records from data records in a file block (header records use #2C while data records use #16).

b. Segments

Each segment contains 256 data bytes and has the following format:

Byte 1	Byte 2	Byte 256	CRC 1	CRC 2
--------	--------	-------	----------	-------	-------

'CRC 1' is the more significant byte and 'CRC 2' the less significant byte of the logical NOT of the CRC calculated for the 256 bytes in the segment. (The CRC polynomial used is ' $X^{15} + X^{12} + X^5 + 1$ ' with an initial seed of #FFFF).

c. Trailer

The trailer is simply an extra 32 one bits written to the end of the record.

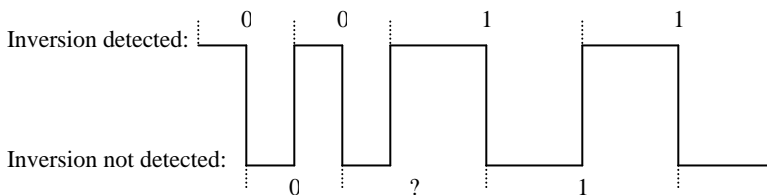
8.3 Bit Format.

A bit is written to the tape as a period of low level followed by an equal period of high level. A one is written to the tape with these periods twice as long as those for a zero. The length of the period for a zero can be set by the user (see CAS SET SPEED).

The tape circuitry has a tendency to move the positions of edges (transitions from high to low or low to high) so as to balance out the difference between ones and zeros written to tape. Precompensation is used - which adds to the period of a one bit and subtracts from the period of a zero bit to make the waveform closer to the ideal when it is read.

When reading, the speed at which the recording was made is determined by timing the one bits in the record leader. As this is a long sequence of the same bit the edges are not shifted and no precompensation is applied. Since the speed is established independently for each record this automatically takes into account most tape speed variations.

Data is written low-high but may be inverted when read (i.e. high-low). It is important for the firmware to determine whether the waveform being read is inverted or not. If this is not achieved then the bits will not be read properly as the following example shows:



The zero bit in the record leader is used to determine whether the recording has been inverted.

Bytes written to the tape are written with the most significant bit first and the least significant bit last.

8.4 The Header Record.

The header record in a file block contains information about the file and about the data in the following data record. Some of the entries in the header are used by the system for various purposes. The remaining entries are available for the user to set when writing a file, and to read when reading a file. These entries are the file type (byte 18) and all the user fields (bytes 24..63) including the logical length (bytes 24..25) and the entry address (bytes 26..27). The user fields will all be set to zero if they are not used.

The header is laid out as follows:

System fields

Bytes 0..15	Filename	Padded to 16 bytes with nulls.
Byte 16	Block number	The first block is normally block 1 and block numbers increase by 1 on successive blocks.
Byte 17	Last block	A non-zero value means that this is the last block of a file.
Byte 18	File type	A value recording the type of the file (see below).
Bytes 19..20	Data length	The number of data bytes in the data record.
Bytes 21..22	Data location	Where the data was written from originally.
Byte 23	First block	A non-zero value means that this is the first block of a file.

User fields

Bytes 24..25	Logical length	This is the total length of the file in bytes.
Bytes 26..27	Entry address	The execution address for machine code programs.
Bytes 28..63	Unallocated	These are unallocated and may be used as required.

The file type (byte 18) is split into a number of fields:

Bit 0	Protection	If this bit is set the file is protected in some way.
Bits 1..3	File contents	0 = Internal BASIC. 1 = Binary. 2 = Screen image. 3 = ASCII. 4..7 are unallocated.
Bits 4..7	Version	ASCII files should be version 1, all other files should be version 0.

8.5 Read and Write Speeds.

The Cassette Manager is capable of reading and writing data at speeds ranging from 700 baud to 2500 baud. There are two speeds commonly used in this range, 1000 baud (the default speed selected at EMS) and 2000 baud. The default speed is chosen to be near the slowest speed to give maximum reliability. The reliability at 2000 baud is still good, however, particularly when playing back on the same machine that was used to record a tape.

Bits are written to the tape as a single cycle of a tone. The tone for a one always has half the frequency of the tone for a zero. Thus ones are twice as long as zeros on the tape. This means that the baud rates given above are only averages and vary according to the actual data written.

Even with the built in cassette mechanism the Cassette Manager has to precompensate the waveform written to the tape to achieve the speeds quoted. This means that the lengths of bits written are altered (ones lengthened, zeros shortened) to try to make the waveform read closer to the ideal after the edges of the waveform have been shifted by the cassette circuitry.

It is only necessary to set the Cassette Manager's write speed. When reading a record from tape the record leader is used to calculate the speed at which it was written. This also allows for tape speed variations between different machines.

8.6 Cataloguing.

To generate a catalogue from the tape the Cassette Manager reads a sequence of file blocks and prints information from them. The file blocks may come from any file, in any order. Cataloguing continues until the user hits the escape key.

The information is reported as follows:

```
FILENAME      block N L Ok
```

FILENAME is either the name of the file or 'Unnamed file' if the filename starts with a null.

The block number, N, indicates which block of the file it is. Normally block 1 is the first block of a file.

L is a character representing the file type and protection status of the file. It is formed by adding #24 (character '\$') to the file type from the header masked with #0F. This gives the following characters:

\$	an unprotected BASIC program.
%	a protected BASIC program.
&	a binary file.
'	a protected binary file.
*	an ASCII file.

Other characters are possible but the above are the standard file types that are written by the on board ROM.

The above information is printed when the header record is read correctly.

Ok is printed after the data record has been read correctly.

8.7 Reading Files.

Before a file can be read from it must be opened (by calling CAS IN OPEN). This sets up the filename (see 8. 10 below) and reads the first block of the file so that the header can be inspected.

The file may either be opened for character input or for direct input, but not both. The mode of input is set by the first access to the file and not when it is opened. As soon as one mode is selected it becomes impossible to use the other mode to access the file.

Character input (calling CAS IN CHAR) allows the user to read the file sequentially one character at a time. Blocks of the file are read from tape into the buffer as needed. This is intended for reading text files and similar applications.

Direct input (calling CAS IN DIRECT) reads the whole of the file into memory in one go. This is intended for loading machine code programs or screen dumps and similar applications.

Interrupts are disabled whilst reading from tape because this has serious timing constraints. Disabling interrupts will prevent the various timer interrupts (as described in section 10. 1) from occurring. In particular this might leave the sound chip making a noise for a long period of time and so the Sound Manager is shut down (see SOUND RESET).

In V1.1 firmware the cassette manager routines for reading files return error codes to indicate the cause of the errors (for compatibility with AMSDOS):

- #00: The user hit escape
- #0E: The stream is already/not in use
- #0F: Have reached the end of the file

8.8 Writing Files.

Before a file can be written to it must be opened (by calling CAS OUT OPEN). This sets up the filename (see 8. 10 below) and the rest of the header that will be written in each file block.

The file may either be opened for character output or for direct output, but not both. The mode of output is set by the first write to the file and not when it is opened. As soon as one mode is selected it becomes impossible to use the other mode to write to the file.

Character output (calling CAS OUT CHAR) allows the user to write to the file one character at a time. The characters are buffered until a complete block (2048 characters) is ready to be written whereupon a file block is written to the tape.

Direct output (calling CAS OUT DIRECT) writes the whole of the file from memory in one go. The data written is still packaged into 2048 byte blocks.

Whichever output mode is used, it is important to close the output file properly (using CAS OUT CLOSE) otherwise the last block of the file will not be written.

Interrupts are disabled whilst writing to tape because this has serious timing constraints. Disabling interrupts will prevent the various timer interrupts (as described in section 10. 1) from occurring. In particular this might leave the sound chip making a noise for a long period of time and so the Sound Manager is shut down (see SOUND RESET).

In V1.1 firmware the cassette manager routines for writing files return error codes to indicate the of the errors (for compatibility with AMSDOS):

#00: The user hit escape
#0E: The stream is already/not in use

8.9 Reading and Writing Simultaneously.

The Cassette Manager allows two files to be open simultaneously. One must be open for reading and the other for writing. Thus it is possible to read from one file and write to another file at the same time.

When the Cassette Manager is about to read a block it asks the user to press PLAY and this implies that the tape with the file for reading should be loaded. Similarly, when it is about to write a block it asks the user to press REC and PLAY and this implies that the tape to which the file is to be written should be loaded. The Cassette Manager assumes that the tape is not changed and that the appropriate cassette controls remain pressed as requested until a prompt is issued. It also assumes that pressing a key means that the prompt has been obeyed.

It is unwise to attempt to read and write simultaneously with the Cassette Manager messages turned off. The only notification given of which tape should be loaded is in the prompt messages.

8. 10 Filenames.

When the user opens a file for reading or writing the name of the file to be read or written is specified. The filename is a string of any 16 characters (#00..#FF). If the file name specified is longer than 16 characters then it is truncated and if it is shorter than 16 characters it is padded to 16 characters with nulls (character # 00).

When opening a file for reading a zero length filename or one that starts with a null has a special meaning - read the next file on the tape. The Cassette Manager searches the tape until it finds the first block of a file and it reads this file. Once the first block of a file has been found the Cassette Manager will only read from that file and no other.

BASIC uses a slightly extended form of the filename. If the first character of a BASIC filename is an exclamation mark (character #21) the BASIC turns the prompt messages off (see 8. 11 below) and removes the exclamation mark from the name. This facility is not provided at the Cassette Manager level.

8.11 Cassette Manager Messages.

The Cassette Manager issues a number of messages to prompt and inform the user and to warn when errors have occurred. The messages that prompt or inform the user may be turned on or off as desired (see CAS NOISY). Messages that inform the user of errors cannot be turned off by this mechanism.

a. Prompt messages.

Press PLAY then any key:

This message is issued when the Cassette Manager is about to read the first block of a file from tape or when it is about to read a block after having written to tape (see section 8.9). It indicates that the tape containing the file to be read should be loaded and that the PLAY button on the recorder should be pressed. The Cassette Manager does not issue this message at other times since it assumes that the correct tape is still loaded and that the PLAY button is still pressed.

Press REC and PLAY then any key:

This message is issued when the Cassette Manager is about to write the first block of a file to tape or when it is about to write a block after having read from tape. It indicates that the tape on which the file is to be written should be loaded and that the REC and PLAY buttons on the recorder should be pressed. The Cassette Manager does not issue this message at other times since it assumes that the correct tape is still loaded and that the REC and PLAY buttons are still pressed.

b. Information messages.

Found FILENAME block N

This message is printed when reading from the tape if a header record is found that for any reason does not match the record that was expected. This may indicate that the tape is positioned incorrectly (too early or too late) or that the wrong tape is being played.

Loading FILENAME block N

A block of the file has been found and is being read from tape.

Saving FILENAME block N

A block of the file is being written to tape.

FILENAME in the above messages is the name of the file or 'Unnamed file' if the filename starts with a null.

The block number, N, indicates which block of the file is being read or written. The first block of a file is normally block 1, the second block 2 etc.

c. Error messages.

Rewind tape

While searching for a block of the file being read, a higher numbered block than that required has been found. The required block has been missed. This message is often produced after a read error in the required block when the next block is found.

Read error X

An error of some kind occurred whilst reading from the tape. The tape should be rewound and the block played again. The X is a single letter indicating what kind of read error occurred:

'a'	Bit too long	An impossibly long one or zero has been measured. This often indicates reading past the end of the record.
'b'	CRC error	Data was read from tape incorrectly.
'd'	Block too long	The data record contains more than the expected 2048 bytes of data.

Write error a

An error occurred whilst writing to the tape. There is only one possible write error. This indicates that the Cassette Manager was unable to write a bit as fast as was requested. This error will never occur unless the user has set the write speed beyond the maximum possible.

8.12 Escape Key.

The escape key on the keyboard may be used to abandon cassette operations at certain times.

When the Cassette Manager issues one of the prompt messages it calls KM READ CHAR repeatedly to empty the key buffer out. Then it calls KM WAIT KEY to wait until the user presses a key to acknowledge the prompt. If the value generated from the key the user presses is #FC, which is the value normally generated by the escape key, then the Cassette Manager will abandon the read or write and will return an error condition to the caller.

When reading from or writing to the cassette interrupts are disabled and the normal key scanning mechanism is not active. While reading or writing the record leader the Cassette Manager itself scans the keyboard to test whether key 66, the escape key, is pressed. If this key is found to be pressed then the Cassette Manager abandons the read or write and returns to the caller (with an appropriate error condition). While reading or writing the data in the record there is no way to interrupt the Cassette Manager, thus pressing ESC may not be detected for several seconds.

8.13 Low Level Cassette Driving.

To allow the user to produce a new filing system the record read and write routines, CAS READ and CAS WRITE, are in the firmware jumpblock. There is a third routine at this level, CAS CHECK, whose facilities are not used by the higher levels of the Cassette Manager. It allows the data that has been written to tape to be compared with the data in store. This could be used to perform a read after write check if so desired.

Also available in the firmware jumpblock are routines to turn the cassette motor on and off (CAS START MOTOR and CAS STOP MOTOR). It is not necessary to turn the motor on and off around a call of CAS READ, CAS WRITE or CAS CHECK as these routines automatically turn the motor on and off.