INTRODUCTION

The EXOS cassette device allows bulk storage of data on cassette tapes. It operates within the EXOS channel scheme, supporting the normal character and block operations. This is accomplished by using a large (4Kbyte) RAM area to buffer characters between tape accesses. Data on cassette tape are organised into chunks of up to 4K with no restriction on the total length of data files.

The data on tape are stored as single cycles of variable frequency. The input circuitry digitises the incoming system with a zero-crossing detection network, allowing accurate timing of cycle periods. A leader tone of constant frequency precedes each 4K chunk to provide timing information. By taking its timing from the leader signal the cassette software can translate incoming data over a wide range of transmission rates, limited by the processor clock speed at high data rates and the low frequency roll-off of the input circuitry at low data rates. The optimum data rate is 2400 'baud' with an 8MHz processor clock.

LOGICAL DATA FORMAT

The standard data chunk is organised into a string of 256 byte blocks, each block preceeded by a byte count and followed by a 2 byte cyclic redundancy check (CRC) value. The first block is preceded by a count of the number of blocks in the chunk. A chunk may contain from 0 to 16 256-byte blocks.

CHUNK format:

```
| | BLOCK | BLOCK | BLOCK | BLOCK | ... | BLOCK |
  |
  |
  |______Block count (1 Byte)
```

BLOCK format:

```
| |           DATA (up to 256 bytes)         | | |
  |                                          | |
  |                                          |_|_ CRC
  |_______________________________________________ Byte count
```

A data file comprises a sequence of data chunks preceded by a header chunk containing information identifying the file. The header contains exactly one block and is itself identified by a value of 0FFH in the block count byte. The header chunk also contains a one byte protection flag. A non-zero value for this flag inhibits the operation of BASIC's COPY function, thereby allowing a measure of data security for cassette files.

HEADER format:

```
| FF |  |  |  |      FILENAME      |  |  |
     |  |  |  |                    |  |
     |  |  |  |                    |__|___ CRC
     |  |  |  |_______________________|___ Filename length
     |  |  |__________________________|___ Protection flag
     |  |_____________________________|___ Byte count
     |________________________________|___ Header identifier
```

The end of the data file (EOF) is sensed by three possible events:

1) Block count = 0
2) 0 < Block count < 16
3) Byte count < 256

Case (1) applies to a file that is an integral multiple of 4096 bytes long. In this case there would be a chunk of zero length following the last data-containing chunk. Case (2) is the normal situation where a chunk is more than 256 bytes below its 4K capacity. If a chunk contains 16 blocks the last of which is incomplete case (3) will recognise this as the EOF.

PHYSICAL DATA FORMAT

As mentioned above, data are stored as single cycles of variable frequency, the frequency relative to the leader frequency determining the logical bit value. A low frequency cycle corresponds to a logical 0, while a high frequency cycle represents a logical 1. The values of these frequencies are chosen to allow +/- 20% wow tolerance.

Each data chunk is preceeded by 2560 cycles of leader tone followed by a single low frequency cycle used to 1) identify the beginning of the data stream and 2) determine to phase of the incoming signal. There then follows a dummy byte to allow the capacitive filtering to recover from the frequency change between the leader and the synchronisation cycle. The data stream is followed by a footer of another 5 cycles of leader tone.

CHUNK format (physical):

LEADER	SYNC CYCLE	DUMMY BYTE	DATA	FOOTER

There are two available speeds for writing data to tape, corresponding roughly to 2400 and 1000 'baud'. The frequencies for these two speeds are listed below:

FAST RATES:	Leader	2.45 KHz
	Sync	0.99 KHz
	1	2.91 KHz
	0	1.99 KHz
SLOW RATES:	Leader	1.03 KHz

Sync	0.49 KHz
1	1.29 KHz
0	0.83 KHz

HARDWARE

The EXOS cassette driver makes extensive use of the variable rate interrupt sources provided by the DAVE chip. These are programmed to provide the time base for reading and writing. Only the lower 8 bits (out of the 12 available) are used.

Output signals are also generated from the DAVE chip using the left channel DAC (port A8). Values of 0 and 1 in this register produce an output square wave signal of +/- 21mV. The right channel is not used.

On input the cassette signal passes through a filtering network designed to remove low frequency noise from DC biased audio cassette recorders. This network tends to interfere with the incoming signal in such a way as to alter duty cycles when the frequency of the input signal changes drastically, as it does in the vicinity of the synchronisation pulse. It is to counter-act this effect that a dummy byte is inserted immediately after the synchronisation pulse.

The filtered input signal then passes through a Schmidt trigger (with 25mV hysteresis) to produce a TTL output that is latched to b7 of port B6. The input circuitry provides accurate zero-crossing detection for input signals ranging from 15V peak-to-peak down to the hysteresis level of 25mV.

The optimum input level is +/- 3V. The strength of the input signal relative to this optimum is detected by a biased comparator buffered through a 10 microF capacitor to b6 of port B6.

OVERALL STRATEGY

In order to maintain compatability with the EXOS channel-based system, channel RAM is used to buffer characters on reading and writing operations. Also, since tape is not a random access device there is a necessary asymmetry between reading and writ-ing. Therefore, the OPEN and CREATE EXOS commands, rather than doing roughly similar operations, are used for reading and writ-ing, respectively. These operations correspond to the BASIC LOAD and SAVE commands.

OPEN => READ from cassette => LOAD
CREATE => WRITE to cassette => SAVE

All reading and writing operations are organised to use two large subroutines that read or write the data chunks to or from channel RAM. When a channel is CREATEd to the cassette driver the cassette driver sets up the header chunk in channel RAM and then writes it to tape. The channel RAM is then filled with characters using either single or block write calls. When the buffer contents reach 4K the next chunk is written to tape. This continues until the channel is DESTROYed, at which time the remaining contents of the channel RAM buffer are written to tape.

Similarly, when a channel is OPENed to the cassette driver, the driver enters a searching routine to find the header corresponding to the desired file. If no filename is specified the first file encountered is used. The user then uses single or block character reads to access the contents of the channel RAM buffer. Whenever the user tries to read from the an empty buffer the cassette driver goes off to read the next chunk. Should that chunk contain one of the three EOF conditions, a flag is set in device RAM which returns an EOF error to the user when he next attempts to read from an empty RAM buffer (or immediatly in the first EOF case).

LOW LEVEL STRATEGY

Both reading and writing depend on the use of the programmable DAVE interrupt generators. The cassette driver, on entering the low level reading and writing routines, reconfigures the EXOS interrupt jump vector at 038H, having stored the previous contents in device RAM.

1) WRITING

Three parameters are used by the tape writing routine: the location of the channel RAM buffer, the number of characters in the buffer, and a flag to determine whether a header chunk is to be written. Having configured the interrupts and the DAVE interrupt timers, the data are written to tape, LSB to MSB, in 256 byte blocks. CRC values are calculated on the data bits and the two byte CRC is sent after each block, low byte first.

The CRC is calculated using the following algorithm:

1) XOR the new bit with b15 of the CRC value
2) Copy the new value of b15 into the CF
3) If CF = 1 then XOR the CRC with 0810H
4) Rotate the CRC one bit to the left, putting the CF into b0.

Having sent the data stream a footer of 5 leader cycles is tacked on the end.

2) READING

The reading routine revolves around two subroutines that start DAVE interrupt timers when transitions in the squared cassette output are detected in b7 of port B6. Both of these routines detect transitions relative to a reference level, the one looking for transitions to the reference state, the other looking for transitions away from the reference. In both cases, when the required transition is found, the DAVE timer is enabled. Computations are done 'underneath' the delay, ending in a HALT instruction. When the DAVE interrupt occurs one of the polling loops is entered, timing the delay until the next appropriate transition, then setting up the next interrupt delay when the transition is detected.

Even though data is written to the cassette with a 50/50 duty cycle, there can be no guarantees that the duty cycle will be preserved on playback. Therefore all computations are done on full cycle timings. Even so, there can be local problems (such as the one described above for the synchronisation pulse) that affect the full cycle timing for a few cycles. The ratio of logical 1 to logical 0 cycle period is not great enough to cause such problems, so data reliablility remains high.

On entry the chunk reading subroutine goes through a process of noise exclusion by looking for 50 cycles of approximately the same frequency, which it then assumes to be the leader tone. Having found the leader (which can be assumed to have a constant duty cycle), a value is calibrated for the DAVE chip corresponding to the half-cycle time plus 25%. This value remains unchanged through the rest of the routine.

The routine then determines the time from the occurrence of the DAVE interrupt to the next transition. This value is the leader cycle reference relative to which the logical 1 and 0 times will be compared. This time is measured and averaged 256 times to ensure consistency.

Having done its calibration from the leader, the synchronisation pulse is detected by comparing cycle times until a very long cycle is found. (In this context 'very long' means more than 25% longer than the leader cycle period.) The phase of the incoming signal is determined by looking at the level of the first half-cycle of the synchronisation pulse and the phase references readjusted accordingly.

The data stream is then read by repeated applications of the same routine: interrupt delay started at leading edge; while waiting for the interrupt to occur compare the last interrupt-to-leading-edge time with that determined from the leader; if the observed time is greater then the bit is a 0, otherwise it is a 1; perform necessary byte manipulations and bufferings; HALT in expectation of the DAVE interrupt; when the interrupt occurs poll to the next leading edge and continue.

EXOS CALLS

1) OPEN CHANNEL

The cassette driver is configured to allow one read channel and one write channel to be open simultaneously. If the user attempts to OPEN a second read channel the driver will return with a .2NDCH error. If there is not enough RAM to hold the 4K buffer a .NORAM error is returned. Finally, if the searching routine is interrupted by the STOP key, a .STOP error will be returned.

If any of these errors occurs, the cassette device will believe that the channel has not been opened succesfully and will allow another read channel to be OPENed. However, the .STOP error condition will leave the 4K channel RAM buffer allocated to the cassette device. It is the user's responsibility to decide whether to de-allocate that RAM by CLOSEing the channel, or to proceed in some other way.

2) CREATE CHANNEL

This function responds in the same way as the OPEN function with the addition that if an OPENed file has had a non-zero protection flag, the CREATE routine returns with a .NOFN error. This condition is like the .STOP error in that the RAM is allocated but the cassette driver does not recognize the channel as being open.

3) CLOSE/DESTROY CHANNEL

CLOSE and DESTROY are identical functions. If a write channel is CLOSE/DESTROYed the remaining buffer contents are written out to tape. CLOSE/DESTROY also clears the status line and reconfigures the status line parameter table to the palette values present when channels were OPEN/CREATEed.

4) CHARACTER/BLOCK READ

Characters are read from the channel RAM buffer. If the buffer is empty and one of the three EOF conditions has been met, a .EOF error is returned to the user. If the EOF condition has not been met, the next tape chunk is read. The operation of the CRC error condition is described below in the section on status line messages.

5) CHARACTER/BLOCK WRITE

Characters are read into the channel RAM buffer. When the buffer size reaches 4K it is automatically written to tape.

> N.B. Character read, block read, character write, and block write will only work on the appropriate read or write channel. If a read operation is attempted on a write channel, or vice versa, a .2NDCH error is returned.

6) CHANNEL STATUS

Returns a flag indicating whether the chunk currently in RAM is the final chunk of the file. This function should not be confused with the .EOF error flag returned by the read operations. The latter error flags are returned in case of an attempt to read a character beyond the last character in the file. Channel status reports on the status of the chunk currently residing in RAM.

C = 0FFH => End of file chunk
= 0 => Not EOF

7) SET CHANNEL STATUS and SPECIAL FUNCTIONS

Not supported by the cassette driver. Attempts to access these functions will result in a .NOFN error.

OTHER FUNCTIONS

1) REMOTE CONTROL

The two remote control relays on the Enterprise are controlled by code in the cassette driver. EXOS maintains two variables, REM1 and REM2, which can be changed to toggle the remotes.

```
REM1/2 = 0 => OFF
        <>0 => ON
```

In the case of a direct call, the entry conditions are:

```
A = 0 => REM1
  <>0 => REM2

D = FF => ON
  <>FF => OFF
```

When reading or writing from tape, the cassette driver uses the remote control relays to switch the tape machine on and off between chunks. If a single read/write channel is open to the cassette driver, both relays are switched simultaneously. If a read and a write channel are both open, as would be the case for copying a cassette, REM1 is used for READ channel and REM2 is used for the write channel, allowing separate control of the two machines.

2) LEVEL INDICATION

A display on the status line reflects the status of the input signal level by reading b7 of port B7. If the level is higher than the optimum, a red square is displayed, otherwise a green square is displayed. The optimum tape machine output level can be found by adjusting until the display flashes rapidly between red and green.

The level indicator only functions properly when the data part of the chunk is being read, where the interrupt timing is not so critical. During the leader, a constant red or green square is displayed, from which coarse adjustments can be made.

3) MESSAGES

Various messages are displayed on the status line during operation of the cassette device.

When a file is saved the legend 'SAVING' appears with the filename (truncated to 15 characters).

On reading a file from tape, the legend 'SEARCHING' appears while the driver looks for a header chunk. When a header is found, the legend 'LOADING <filename>' appears if the correct header has been found, or 'FOUND <filename>' if the correct file header has yet to be found. In the latter case the searching routine continues until the correct file header is found.

If a CRC error is detected in the course of reading a chunk, a flag is set in device RAM. When the user attempts to read the from the channel buffer, a .CCRC error is returned and the message 'LOADING ERROR' appears on the status line.

The status line is blanked when channels are CLOSE/DESTROYed.

4) STOP KEY

The STOP key is tested on every input level test and every output interrupt. A .STOP error is returned and the software interrupt request flag set.

5) WRITING SPEED

The output data rate can be adjusted by setting the EXOS variable SP_TAPE.

```
SP_TAPE  = 0 => 2400 'baud' (default)
          <>0 => 1000 'baud'
```