This is my attempt to document the file formats stored on cassette tape for the Salora Manager and Laser 2001 microcomputers.

The data is written to the tape interface by different cycle durations to represent logic 0 or 1.

The loop counters are set to write logic high for count loops, then logic low for count loops. A longer complete cycle indicates a 1 bit.

	0	1
Laser 2001	30	90
Salora Manager	40	80

The loop count table shows that the output counter for the machines vary by a little. I would assume that this has something to do with the hardware used to detect changes, as they both load using the same algorithm.

To begin with, here's an MSBASIC hello world program, as it is sent to the cassette interface.

```
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000200   01 A5 99 5A  01 50 15 00  16 50 14 50  0A 00 BA 22  ...Z.P...P.P..."
00000210   48 65 6C 6C  6F 20 57 6F  72 6C 64 22  00 00 00 54  Hello World"...T
00000220   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000230   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
```

The CSAVE function is

1) Write sync bits, 4096 x 0 bits (512 bytes)
2) Write sync marker ($01, $A5, $99, $5A)
3) Write memory address ($01, $50)
4) Write length in bytes ($15, $00)
5) Write end of program memory ($16, $50) == (Step 2 + Step 3)
6) Write program data
7) Write checksum ($54)
8) Write sync bits, 4096 x 0 bits (512 bytes)

In both machines, MSBASIC starts at $5001. The location $5000 is always $00 and is unused.

The checksum is calculated by adding the written bytes, after the signature bytes, modulus 256.

```
00000200   01 A5 99 5A  01 50 15 00  16 50 14 50  0A 00 BA 22  ...Z.P...P.P..."
00000210   48 65 6C 6C  6F 20 57 6F  72 6C 64 22  00 00 00 54  Hello World"...T
```

This totals to $654 % 256 == $54

The BSAVE function is similar

This is a BSAVE of ROM address from $C0A0 for 48 bytes.

```
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00000200   01 A5 99 5A  A0 C0 30 00  C0 AD 00 12  C9 A2 D0 03  ...Z..0.........
00000210   6C FE 2F 4C  3A E2 8D 42  41 53 49 43  20 56 45 52  l./L:..BASIC VER
00000220   53 49 4F 4E  20 32 2E 31  8D 28 43 29  20 31 39 38  SION 2.1.(C) 198
00000230   33 20 56 54  4C 8D 00 1C  09 01 A5 99  5A
```

1) Write sync bits (4096 x 0 bits)
2) Write sync marker ($01, $A5, $99, $5A)
3) Write load/save address ($A0, $C0 or $C0A0 LE)
4) Write length in bytes ($30, $00)
5) Write data
6) Write checksum ($09)
7) Write sync marker ($01, $A5, $99, $5A)

The checksum is calculated as before, so $204 - $237 gives, $1109 % 256 == $09.

The STORE function saves the value using the same method as BSAVE. The difference being the the load/save address is always $0200 and the length is always 255.