

Table of contents

1 Ordered tape to CBM content.....	2
1.1 Header data block.....	3
1.1.1 Synchronization.....	3
1.1.2 Header data.....	3
1.2 Content data block(-s).....	3
1.2.1 Synchronization.....	3
1.2.2 Content data.....	3
2 Repeatedly used.....	4
2.1 Synchronization.....	4
2.2 Data following synchronization.....	4
2.2.1 1st data transmit.....	4
2.2.1.1 1st countdown sequence.....	4
2.2.1.2 Data.....	4
2.2.1.3 Transmit block gap.....	4
2.2.2 2nd data transmit.....	4
2.2.2.1 2nd countdown sequence.....	4
2.2.2.2 Data.....	4
2.2.2.3 Transmit block gap.....	4
2.3 Data.....	5
2.3.1 Payload.....	5
2.3.2 Checksum.....	5
2.3.3 Optional end-of-data marker.....	5
2.4 Transmit block gap.....	5
2.5 Header payload.....	5
2.5.1 File type.....	5
2.5.2 Destination infos.....	6
2.5.3 File name.....	6
2.5.4 Additional bytes.....	6
2.6 Content payload.....	6
2.7 Byte.....	6
2.7.1 Marker.....	6
2.7.2 Payload bits.....	6
2.7.3 Parity bit.....	6
2.8 Symbols.....	7
2.9 Pulses.....	7
3 Sources.....	7

1 Ordered tape to CBM content

This document contains the ordered tape / datasette to Commodore computer transfer content.

Explanations about all the stuff sent to the Commodore computer is also included.

I created and used this document to successfully implement a (partial) datasette simulator running on a Raspberry Pi that is connected via its GPIO pins to a Commodore computer (e.g a CBM / PET, C64, VIC20, etc.).

Find out more about the project at my website:

<http://rhinodevel.com/>

Or directly at my Github account:

<https://github.com/RhinoDevel>

1.1 Header data block

1.1.1 Synchronization

See 2.1.

1.1.2 Header data

See 2.2.

The header's data payload (also see 2.3.1 for general informations) description is found at 2.5.

1.2 Content data block(-s)

1.2.1 Synchronization

See 2.1.

1.2.2 Content data

See 2.2.

The content's data payload (also see 2.3.1 for general informations) description is found at 2.6.

2 Repeatedly used

2.1 Synchronization

Sequence of short pulses (~1500 should be OK).

2.2 Data following synchronization

2.2.1 1st data transmit

2.2.1.1 *1st countdown sequence*

9 bytes, \$89 to \$81.

2.2.1.2 *Data*

See 2.3.

2.2.1.3 *Transmit block gap*

See 2.4.

2.2.2 2nd data transmit

2.2.2.1 *2nd countdown sequence*

9 bytes, 9 to 1.

2.2.2.2 *Data*

See 2.3.

2.2.2.3 *Transmit block gap*

See 2.4.

2.3 Data

2.3.1 Payload

2.3.2 Checksum

1 byte, holding XOR of 0 and all payload bytes (0 XOR byte0 XOR byte1 XOR byte2 ...).

2.3.3 Optional end-of-data marker

One long pulse, followed by one short pulse.

More recent Kernals seem to add this.

2.4 Transmit block gap

One long pulse,

followed by a sequence of short pulses (60 should be OK).

2.5 Header payload

At least the C64 always loads the additional bytes (see 2.5.4) into the tape buffer at \$033C.

The C64 will ignore the start address, if the file type is a relocatable program and will always use \$0801 as start address (other machines may also do this with their equivalent fixed destination address).

See 2.3.1 for its use.

2.5.1 File type

\$01	Relocatable program.
\$02	Data block for SEQ file.
\$03	Non-relocatable program.
\$04	SEQ file header.
\$05	End-of-tape marker.

2.5.2 Destination infos

None, if file type (see 2.5.1) is \$02, otherwise:

2 bytes	Start address (least significant byte first).
2 bytes	End address +1 (least significant byte first).

2.5.3 File name

None, if file type (see 2.5.1) is \$02, otherwise:

16 bytes	File name (PETSCII, padded with blanks / \$20).
----------	---

2.5.4 Additional bytes

171 bytes	By default a Kernal fills these with blanks (\$20) during SAVE. Also see 2.5.
-----------	---

2.6 Content payload

See 2.3.1 for its use.

2.7 Byte

2.7.1 Marker

The new-data marker represented by one long, followed by one medium pulse.

2.7.2 Payload bits

The eight payload bits of the byte, least significant bit first (0 to 7).

2.7.3 Parity bit

1 bit, holding XOR of 1 and all payload bits (1 XOR bit0 XOR bit1 XOR bit2 ...).

2.8 Symbols

Symbols always use two different pulses (see 2.9).

Short, followed by medium pulse	0 bit (see 2.7.2 for its use).
Medium, followed by short pulse	1 bit (see 2.7.2 for its use).
Long, followed by medium pulse	New-data marker (see 2.7.1 for its use).
Long, followed by short pulse	(Optional) end-of-data marker (see 2.3.3 for its use).

2.9 Pulses

A “pulse” is defined here as a single period of a square wave with a 50% duty cycle. For the LOAD, this period has a transition from low to high in the middle. For the SAVE, the transition in the middle goes from high to low.

During LOAD, the pulse length detection triggers on descending (negative) edges.

During LOAD, “pulse length” is the time distance between two consecutive high-low transitions on the READ line of the tape port.

During SAVE, “pulse length” means the time distance between two consecutive low-high transitions on the WRITE line of the tape port.

Pulse type	Square wave frequency
Long	1488 Hz
Medium	1953 Hz
Short	2840 Hz

3 Sources

http://c64tapes.org/dokuwiki/doku.php?id=loaders:rom_loader

<https://github.com/ikorb/tapecart/blob/master/src-firmware/main.c>

http://c64tapes.org/dokuwiki/doku.php?id=analyzing_loaders