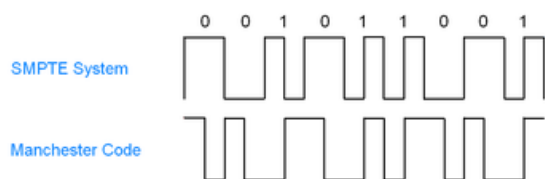


# SMPTE timecode

**SMPTE timecode** is a set of cooperating standards to label individual frames of video or film with a **time code** defined by the **Society of Motion Picture and Television Engineers** in the SMPTE 12M specification. SMPTE revised the standard in 2008, turning it into a two-part document: SMPTE 12M-1 and SMPTE 12M-2, including new explanations and clarifications.

Timecodes are added to **film**, **video** or audio material, and have also been adapted to synchronize **music**. They provide a time reference for editing, **synchronization** and identification. Timecode is a form of media **metadata**. The invention of timecode made modern **videotape editing** possible, and led eventually to the creation of **non-linear editing systems**.

## 1 Basic concepts



*SMPTE timecode signal (A 1 is expressed by a transition at the midpoint of a period. A 0 is expressed by the absence of such a transition.) compared to the outwardly-similar Manchester code (A 0 is expressed by a high-to-low transition, a 1 by low-to-high transition at the midpoint of a period).*

SMPTE timecodes (/ˈsɪmpti:/ or /ˈsɪmti:/) contain **binary coded decimal hour:minute:second:frame** identification and 32 bits for use by users. There are also drop-frame and color framing flags and three extra 'binary group flag' bits used for defining the use of the user bits. The formats of other varieties of SMPTE time codes are derived from that of the longitudinal timecode.

Time codes may use a number of **frame rates**. Common ones are:

- 24 frame/sec (film, **ATSC**, 2k, 4k, 6k)
- 25 frame/sec (**PAL** (Europe, Uruguay, Argentina, Australia), **SECAM**, **DVB**, **ATSC**)
- 29.97 (30 ÷ 1.001) frame/sec (**NTSC** American System (US, Canada, Mexico, Colombia, etc.), **ATSC**, **PAL-M** (Brazil))

- 30 frame/sec (**ATSC**)

In general, SMPTE timecode frame rate information is implicit, known from the rate of arrival of the timecode from the medium, or other metadata encoded in the medium. The interpretation of several bits, including the "colour framing" and "drop frame" bits, depends on the underlying data rate. In particular, the drop frame bit is only valid for a nominal frame rate of 30 frame/s: see below for details.

More complex timecodes such as **vertical interval timecode** can also include extra information in a variety of encodings.

## 2 Discontinuous timecode, and fly-wheel processing

Timecodes are generated as a continuous stream of sequential data values. In some applications "wall clock" time is used, in others the time encoded is a notional time. After making a series of recordings, or after crude editing, recorded timecodes may consist of discontinuous segments.

In general it is not possible to know the linear timecode (**LTC**) of the current frame until the frame has already gone by, by which time it is too late to make an edit. Practical systems watch the ascending sequence of the timecode, and infer the time of the current frame from that.

As timecodes in analog systems are prone to bit-errors and drop-outs, most timecode processing devices check for internal consistency in the sequence of timecode values, and use simple error correction schemes to correct for short error bursts. Thus, a boundary between discontinuous timecode ranges cannot be determined exactly until several subsequent frames or discontinuous sequences of them have passed. For this reason, most videotape editing attempts to keep the timecode of the recorded material continuous, so that multiple edits may be repeatedly over-recorded onto the same piece of videotape.

Although it would be possible in all-digital systems to eliminate the need for the flywheel algorithm by adding a frame delay to allow the timecode to be decoded prior to the processing of the frame, this is not done in most practical systems because it would introduce an unnecessary frame delay in the signal processing path, and there

would still be a need to compensate for timecode errors in signals derived from analog video or audio systems.

### 3 Drop frame timecode

Drop frame timecode originates from a compromise invented when color NTSC video was invented. The NTSC designers wanted to retain compatibility with existing monochrome televisions. To minimise subcarrier visibility on a monochrome receiver it was necessary to make the color subcarrier an odd multiple of half the line scan frequency; the multiple originally chosen was 495. With a 30 Hz frame rate the line scan frequency is  $(30 \times 525) = 15750$  Hz. So the subcarrier frequency then became  $(495/2 \times 15750) = 3.898125$  MHz. This was the subcarrier frequency originally chosen, but tests showed that on some monochrome receivers an interference pattern caused by the beat between the color subcarrier and the 4.5 MHz sound intercarrier could be seen. The visibility of this pattern could be greatly reduced by lowering the subcarrier frequency multiple to 455 (thus increasing the beat frequency from approx 600 kHz to approx 920 kHz) and by making the beat frequency also equal to an odd multiple of half the line scan frequency. This latter change could have been achieved by raising the sound intercarrier by 0.1% to 4.5045 MHz, but the designers, concerned that this might cause problems with some existing receivers, decided instead to reduce the color subcarrier frequency, and thus both the line scan frequency and the frame rate, by 0.1% instead. Thus the NTSC color subcarrier ended up as 3.57954545 MHz (actually 315/88 MHz), the line scan frequency as 15734.27 Hz and the frame rate 29.97 Hz (exactly  $30/1.001$  Hz).<sup>[1]</sup>

This meant that an “hour of timecode” at a nominal frame rate of 29.97 frame/s was longer than an hour of wall-clock time by 3.59 seconds, leading to an error of almost a minute and a half over a day, as the timecode was calculated in a manner that assumed the frame rate was exactly 30 frame/s.

To correct this, drop frame SMPTE timecode was invented. In spite of what the name implies, *no* video frames are dropped (skipped) using drop-frame timecode. Rather, some of the timecode “labels” are dropped. In order to make an hour of timecode match an hour on the clock, drop-frame timecode drops frame numbers 0 and 1 of the first second of every minute, except when the number of minutes is divisible by ten (i.e. when minutes mod 10 equals zero). This achieves an “easy-to-track” drop frame rate of 18 frames each ten minutes (18,000 frames @ 30frame/s) and almost perfectly compensates for the difference in rate, leaving a residual timing error of roughly 86.4 milliseconds per day, an error of only 1.0 ppm.

That is, drop frame TC drops 2 frame counts every minute, except every tenth minute, achieving  $30 \times 0.999 =$

29.97 frame/s. The error is the difference between 0.999 and  $1/1.001 = 0.999000999000999\dots$

For example, the sequence when frame counts are dropped:

```
01:08:59:28
01:08:59:29
01:09:00:02
01:09:00:03
```

For each tenth minute

```
01:09:59:28
01:09:59:29
01:10:00:00
01:10:00:01
```

While non-drop timecode is displayed with colons separating the digit pairs—“HH:MM:SS:FF”—drop frame is usually represented with a semi-colon (;) or period (.) as the divider between all the digit pairs—“HH;MM;SS;FF”, “HH.MM.SS.FF”—or just between the seconds and frames—“HH:MM:SS;FF” or “HH:MM:SS.FF”. The period is usually used on VTRs and other devices that don't have the ability to display a semi-colon.

Drop frame timecode is typically abbreviated as DF and non-drop as NDF.

### 4 Color framing and timecode

A **color framing** bit is often used to indicate field 1 of the colour frame, so that editing equipment can make sure to edit only on appropriate field boundaries in order to prevent picture corruption.

## 5 Studio operations and master clocks

In **television studio** operations, longitudinal timecode is generated by the studio master sync generator, and distributed from a central point. Central sync generators usually derive their timing from an **atomic clock**, either using network time or **GPS**. Studios usually maintain two or three clocks, and automatically switch over if one fails.

A recent development is to mount small GPS-synchronized SMPTE timecode generators on each camera, which eliminates the distribution network for portable set-ups and shooting on location.

Longitudinal SMPTE timecode is widely used to synchronise music. A frame rate of 30 frame/s is often used for audio in America, Japan, and other countries which rely on a 60 Hz mains frequency and use the **NTSC** television standard. The **EBU** (European Broadcasting Union) standard frame rate of 25 frame/s is used throughout Europe, Australia and wherever the mains frequency is 50 Hz, and the **PAL** or **SECAM** television standards are

used.<sup>[2]</sup>

## 6 SMPTE timecode media

1. **Linear timecode**, a.k.a. “longitudinal timecode” and “LTC”: suitable to be recorded on an audio channel, or by audio wires. This is how it is distributed within a studio to synchronize recorders and cameras. To read LTC, the recording must be moving, meaning that LTC is useless when the recording is stationary or nearly stationary. This shortcoming led to the development of VITC.
2. **Vertical interval timecode**, a.k.a. **VITC** (pronounced “vit-see”): recorded directly into the **VBI** (vertical blanking interval) of the video signal on each frame of video. The advantage of VITC is that, since it is a part of the playback video, it can be read when the tape is stationary.
3. **CTL timecode** (control track longitudinal): SMPTE timecode embedded in the control track of a video tape.
4. **Visible time code**, a.k.a. **burnt-in timecode** and **BITC** (pronounced “bit-see”) - the numbers are burnt into the video image so that humans can easily read the time code. Videotapes that are duplicated with these time code numbers “burnt-in” to the video are known as *window dubs*.
5. Film labels, such as **Keycode**.

## 7 History

Longitudinal and vertical-interval timecodes were developed in 1967 by EECO, an electronics company that developed video recorders, and later video production systems. EECO assigned its intellectual property to permit public use.

## 8 See also

- Linear timecode
- Vertical interval timecode
- Burnt-in timecode
- MIDI timecode
- AES-EBU embedded timecode
- Rewritable consumer timecode
- 2-3 pulldown
- Field dominance

## 9 References

- John Ratcliff (1999). *Timecode: A user's guide, second edition* (Third ed.). Focal Press. ISBN 978-0-240-51539-7.
- Charles Poynton (1996). *A Technical Introduction to Digital Video*. John Wiley & Sons. ISBN 0-471-12253-X.

[1] “Color Television Standards - Selected papers and records of the NTSC” edited by Donald Fink, McGraw Hill 1955

[2] <http://www.philrees.co.uk/articles/timecode.htm>

## 10 External links

- Introduction to the Basic Principles of the SMPTE/EBU Time Code
- *museum.tv* article about timecode and video editing
- Technical Introduction to Timecode by Charles Poynton
- Article on timecode by Chris Pirazzi
- Synchronisation and SMPTE TimeCodes.
- Peter Utz. “SMPTE Time Code Explained”. Archived from the original on 2009-02-10.
- Conversion between SMPTE hh:mm:ss:ff Time Code and Frames with c source code by Brooks Harris

## 11 Text and image sources, contributors, and licenses

### 11.1 Text

- **SMPTE timecode** *Source:* [https://en.wikipedia.org/wiki/SMPTE\\_timecode?oldid=724610487](https://en.wikipedia.org/wiki/SMPTE_timecode?oldid=724610487) *Contributors:* Damian Yerrick, The Anome, Aldie, JohnOwens, Nixdorf, CesarB, Julesd, Mulad, Mrand, Robbot, Rsduhamel, Unfree, Rchandra, AlistairMcMillan, Bobblewik, Christopherlin, Gary D, Jh51681, DmitryKo, Rich Farmbrough, Pjacobi, Kwamikagami, Mtruch, Polluks, Courtarro, Sketchee, Drbreznjev, NilsPhilippsen, Tabletop, Marudubshinki, Misternuvistor, Forage, MMuzammils, EngineerScotty, Długosz, Deville, SmackBot, Mauls, Andyweb, Daniel.Cardenas, Ckatz, Loadmaster, The Mekon, Kvng, Robin Kerrison, Paulshanks, Sgw1009, Deflective, Mark Rizo, Magioladitis, KTo288, BigslyE5, Jeremyh3, Is Mise, Lightmouse, ClueBot, Mild Bill Hiccup, Xtophyr Wright, DumZiBoT, Kozmefulanito, XLinkBot, Spitfire, Addbot, Non-dropframe, Sschulte, OIEnglish, FriendlyRobotOverlord, AnomieBOT, JimVC3, Patrick Waddell, FrescoBot, Wfplb, LittleWink, Trappist the monk, A p3rson, Ammodramus, Chronulator, John of Reading, Neatonk, Scatterbrained, ClueBot NG, Davidlooser, JordoCo, Helpful Pixie Bot, Audioinfo, BattyBot, Bacon76, Rwwrowley, Timecode76, Brooks E Harris and Anonymous: 66

### 11.2 Images

- **File:EmbeddedSynchronousSignalling.png** *Source:* <https://upload.wikimedia.org/wikipedia/commons/b/b5/EmbeddedSynchronousSignalling.png> *License:* CC-BY-SA-3.0 *Contributors:* Createted in MS Paint by the uploader *Original artist:* Bob DuHamel
- **File:Text\_document\_with\_red\_question\_mark.svg** *Source:* [https://upload.wikimedia.org/wikipedia/commons/a/a4/Text\\_document\\_with\\_red\\_question\\_mark.svg](https://upload.wikimedia.org/wikipedia/commons/a/a4/Text_document_with_red_question_mark.svg) *License:* Public domain *Contributors:* Created by bdesham with Inkscape; based upon Text-x-generic.svg from the Tango project. *Original artist:* Benjamin D. Esham (bdesham)

### 11.3 Content license

- Creative Commons Attribution-Share Alike 3.0