- Given the audio file in WAV format (contained in the ZIP archive together with this instructions), decode the binary data encoded in it.
- The data is encoded using Audio Frequency Shift-Keying (AFSK) in its simplest form
  - A single bit is the waveform between two zero-crossings
  - A one signal is a rectangle signal of t = 320 microseconds
  - A zero signal is a rectangle signal of t = 640 microseconds
  - The real-life data might no longer be an ideal rectangle, since it has undergone storage on physical media (e.g. a tape drive)
- The bit-stream that can be extracted from the decoded audio signal can be converted into bytes
  - The signal starts with a lead tone of roughly 2.5 seconds (all 1-bits, or 0xff bytes), and ends with an end block of about 0.5 seconds (all 1-bits).
  - 11 bits are used to encode a single byte – 8 bits for the byte plus one start bit (valued 0) and two stop bits (valued 1)
  - The data is encoded with least-significant bit first
- The byte-stream has the following form:
  - The first two bytes are 0x42 and 0x03
  - After that, construct 64 messages of 30 bytes each, with the 31st byte being the checksum of the 30 bytes before that (you need 1984 bytes = 64 * 31 for that) •

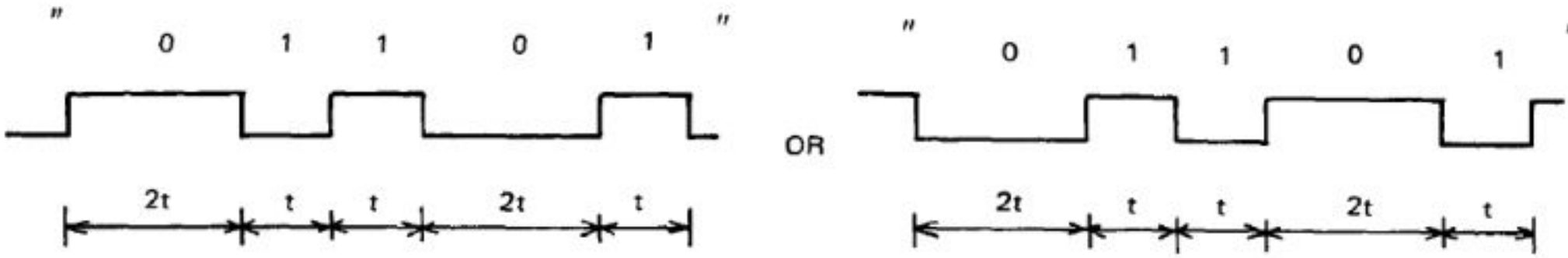
    The last byte before the end block is a 0x00 byte.
- The checksums will help you detect that your encoding works
- The data in this real-life file will have no meaning to you, unless you figure out which machine it was created by – this could be near impossible (don’t try).

Explanation : Binary Encoding

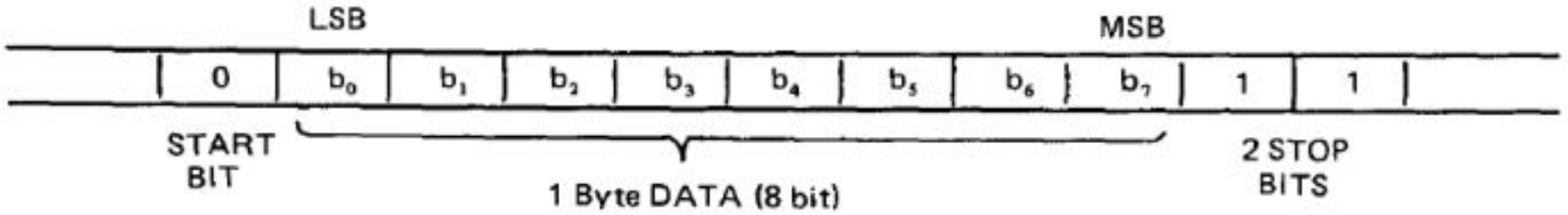
1. Modulation system

"1" . . . . . t = 320 μs
"0" . . . . . 2t = 640 μs

Example:



Explanation: Bit-stream encoding, single byte in the bit -



Explanation: Message format, overall structure and checksum positions in byte - stream