Sentinel Workbench

Signal and bit stream analysis software

The analyst selects a signal in a wideband recording, measures it, demodulates it and analyses the resulting bit stream. At each step the tool proposes parameters with the evidence for them, and the analyst decides. The steps form an analysis flow that is saved and run again on other recordings, one at a time or a whole folder at once, on Windows and Linux, by the analyst or by a software agent.

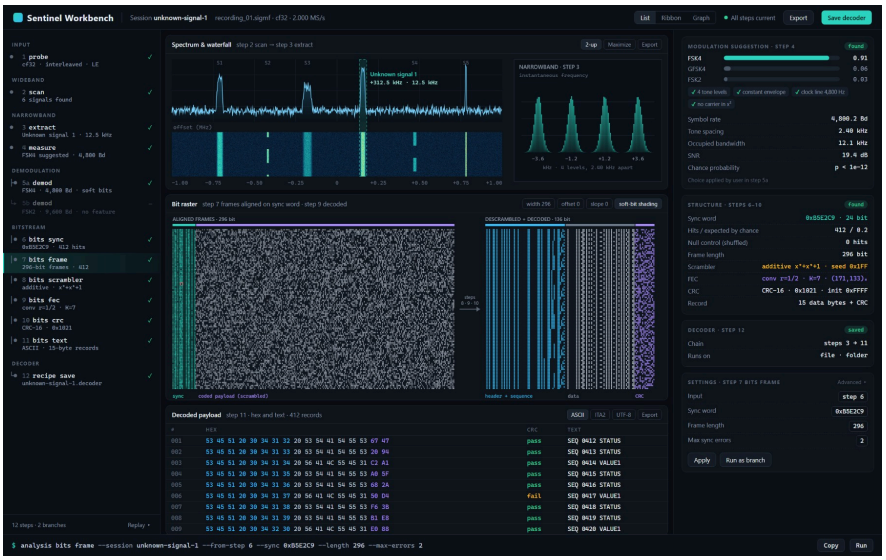

Figure 1. Example analysis of an unknown data signal with synthetic data: the signal selected in a wideband recording and its modulation measured, frames aligned on the sync word, the scrambler and error-correction code recovered and the payload decoded, with the analysis steps listed on the left

Key Features

- Signal selection and measurement
- Demodulation and bit input
- Framing and frame structure
- De-interleaving, descrambling and error correction
- Bit stream analysis
- Evidence for every finding
- Reusable decoders
- Custom steps
- Built for agent workflows

AT A GLANCE

PLATFORM
Windows and Linux; user interface and command line

AUTOMATION
Machine-readable result for every command; skills for software agents

CUSTOM STEPS
Python, inserted anywhere in the analysis flow

INPUTS
IQ and audio recordings (SigMF, WAV; sample type, byte order, real or complex, sample rate and centre frequency set by the user); soft-bit, hard-bit and binary files

OUTPUTS
SigMF, WAV, JSON, CSV, bit files, binary, text

APPLICATIONS

- Analysis of unknown data links found in monitoring
- Recovering the payload of recorded data transmissions
- Reusable decoders for newly identified formats
- Signal analysis exercises in EW operator training
- Automated analysis by software agents on closed networks

Capabilities

Signal selection and measurement

Selects a signal in a wideband recording, shifts it to baseband and measures it. Modulation proposals come with the measurements behind them; the analyst makes the decision.

Demodulation and bit input

Demodulates the selected signal in the same tool. Soft-bit, hard-bit and binary files from other demodulators are accepted as input too.

Framing and frame structure

Symbol-to-bit mapping including Gray coding, sync word search, unknown frame length, take/skip, demultiplexing, majority vote and destuffing (HDLC, DLE, SLIP, PPP). Identifies frame fields such as constants, counters, addresses and lengths, and edits frames in a frame editor.

De-interleaving, descrambling and error correction

Block, diagonal, convolutional and permutation de-interleavers with size search. Additive and self-synchronising descramblers with polynomial and initial-state search. Hamming, Golay, BCH, Reed-Solomon, convolutional and LDPC decoding, with a search for the parameters of an unknown code.

Bit stream analysis

Bit statistics, periodic and non-periodic pattern search, and alignment of two streams; bit error rate against a known sequence measures a link or a decoder.

Evidence for every finding

Every finding shows its confidence and the probability that it arose by chance, so a result is checked before the next step builds on it.

Reusable decoders

The analysis flow is replayed exactly and edited as a list or graph. A finished flow runs on every recording in a folder, and every step also runs from the command line.

Custom steps

Where the method needs a step the tool does not have, the analyst writes it in Python and inserts it anywhere in the flow. It runs like a built-in step (in the user interface, from the command line, in folder batches and inside saved decoders), and its output opens in the same viewers as any other step.

Built for agent workflows

Every step runs from the command line and returns a machine-readable result with its findings, confidence and chance probability. With skills supplied for the analysis method, the workflow is designed so that a software agent, even one running a small language model on a closed network, follows the same method and checks as the analyst.

Specifications

General	
Platform	Windows and Linux; user interface and command line
Automation	Machine-readable result for every command; skills for software agents
Custom steps	Python, inserted anywhere in the analysis flow
Inputs	IQ and audio recordings (SigMF, WAV; sample type, byte order, real or complex, sample rate and centre frequency set by the user); soft-bit, hard-bit and binary files
Outputs	SigMF, WAV, JSON, CSV, bit files, binary, text
Framing	Sync word search; unknown frame length; take/skip; demultiplexing; majority vote; destuffing (HDLC, DLE, SLIP, PPP)
De-interleaving	Block, diagonal, convolutional, permutation; size search
Descrambling	Additive and self-synchronising; polynomial and initial-state search
Error correction	Hamming, Golay, BCH, Reed-Solomon, convolutional (Viterbi), LDPC; search for block length, rate, generator and puncturing pattern
Integrity checks	CRCs and checksums, including unknown initial value and final XOR
Analysis	Bit statistics; periodic and non-periodic pattern search; alignment and comparison of two streams; bit error rate against a known sequence
Views	Bit raster, frame matrix, frame editor, hex
Text	ASCII, UTF-8, ITA2/Baudot and other code pages

Warranty

All hardware and software procured from Zeta is under warranty against manufacturing or design failures for 2 years. Customer-specific warranty terms may be arranged.

Disclaimer

Due to ongoing research and development, Zeta reserves the right to change product specifications without notice.

About Zeta

Zeta Defence develops electronic intelligence (ELINT) and communication intelligence (COMINT) equipment and radar / electronic warfare simulators. Its capabilities include RF transceiver system design, digital design and large-scale software development.

Contact

ZETA SAVUNMA ELEKTRONİK YAZILIM VE DANIŞMANLIK SANAYİ TİCARET A.Ş.
Teknopark İstanbul, Sanayi Mahallesi Teknopark Bulvarı No: 1/2C - 2108, 34906 Pendik / İSTANBUL
Tel +90 216 510 22 55 · zeta@zetasavunma.com · www.zetasavunma.com