

GX-TXT

Analisi e identificazione sperimentale dei protocolli IR con GNU Octave

Manuale operativo del pacchetto software

Software di analisi	GXTXT_analisi_VIR_protocollo_v2_4.m
Software di identificazione	GXTXT_identifica_protocollo_IR_v1.m
Versione del manuale	1.0 - 27 agosto 2026
Licenza del software	GNU GPL v3 o successiva (GPL-3.0-or-later)

Scopo: acquisizione -> analisi temporale -> MARK/SPACE -> identificazione del protocollo

Il software nasce nel progetto GX-TXT, ma il metodo è utilizzabile su qualunque segnale nel quale la portante IR sia ancora osservabile.

1. Scopo e filosofia d'uso

Questo manuale descrive l'uso pratico di due programmi GNU Octave pensati per essere eseguiti in sequenza. **Non è necessario acquisire il segnale in un punto specifico del GX-TXT:** il requisito fondamentale è che nel punto scelto siano ancora distinguibili i singoli cicli della portante e gli intervalli nei quali la portante è presente o assente.

Principio generale

Il primo programma trasforma una forma d'onda campionata in una sequenza temporale di MARK e SPACE. Il secondo utilizza tale sequenza, insieme alla frequenza di portante quando disponibile, per confrontare il comando con alcune famiglie di protocolli IR consumer.

Programma	Funzione principale	Output chiave
GXTXT_analisi_VIR_protocollo_v2_4.m	Importa il CSV, esegue il pretest, misura la portante, ricostruisce MARK/SPACE e produce grafici e riepiloghi.	eventi_mark_space.csv
GXTXT_identifica_protocollo_IR_v1.m	Confronta temporizzazioni e struttura del frame con protocolli noti e tenta la ricostruzione di bit e byte.	identificazione_protocollo_IR.txt

Il software di identificazione è volutamente definito **sperimentale**: protocolli differenti possono condividere portante e temporizzazioni simili. Lo score prodotto dal programma è un indice euristico di compatibilità, non una probabilità statistica.

2. Flusso di lavoro

Fase	Cosa avviene	Elemento da verificare
1. Acquisizione	L'oscilloscopio registra il segnale grezzo.	Comando completo, alcuni campioni a riposo, portante visibile.
2. Esportazione	La forma d'onda viene salvata in CSV.	Passo temporale noto e riferimento al trigger ricostruibile.
3. Pretest	Il primo programma cerca automaticamente una finestra valida attorno al trigger.	Segnale sufficiente, polarità e portante riconoscibili.
4. Analisi completa	Vengono individuati impulsi, burst, MARK, SPACE e classi di durata.	eventi_mark_space.csv coerente con la forma d'onda.
5. Identificazione	Il secondo programma confronta i timing con famiglie note.	Classifica, bit/byte ricostruiti e controlli di coerenza.

Cosa NON serve

Non è richiesta una polarità prefissata, un livello DC specifico o un valore assoluto particolare della tensione. La polarità viene riconosciuta automaticamente e la soglia di analisi è ricavata in modo relativo all'escursione del segnale e al rumore. La versione 2.4 mantiene tuttavia una soglia minima di plausibilità pari a 50 mV, modificabile in configurazione.

3. Requisiti minimi del segnale

Il punto di misura può essere scelto liberamente, purché rispetti i requisiti seguenti:

- i singoli cicli della portante devono essere ancora riconoscibili; un segnale già demodulato, nel quale la portante è stata rimossa, non è adatto alla versione corrente del primo programma;
- il rapporto segnale/rumore deve consentire di separare con affidabilità gli impulsi dal livello di riposo;
- il segnale non deve essere fortemente clippato o deformato al punto da fondere cicli consecutivi;
- la registrazione deve contenere almeno un comando completo e, preferibilmente, un breve tratto di quiete prima dell'evento;
- il passo temporale deve essere noto e sufficientemente piccolo rispetto al periodo della portante.

Esempi di punti utilizzabili: uscita del fotodiodo, collettore di uno stadio di amplificazione, nodo intermedio analogico o qualunque altro punto nel quale la modulazione a portante sia ancora presente. Il nodo VIR è semplicemente il punto utilizzato durante lo sviluppo iniziale del software.

4. Frequenza di campionamento e profondità di memoria

L'acquisizione usata durante lo sviluppo a 10 MSa/s e molti milioni di punti era volutamente ad alta risoluzione. **Non è un requisito per il reverse engineering del protocollo.**

Per una portante vicina a 38 kHz, il numero medio di campioni per periodo è $F_s / 38 \text{ kHz}$. La tabella seguente fornisce indicazioni pratiche, non limiti rigidi.

Sample rate	Passo temporale	Campioni/periodo a 38 kHz	Campioni in 560 us	Valutazione pratica
100 kSa/s	10 us	2.6	56	Troppo basso per una misura robusta della portante; evitare se possibile.
250 kSa/s	4 us	6.6	140	Utilizzabile con segnale pulito; precisione sulla portante limitata.
500 kSa/s	2 us	13.2	280	Buono per ricostruzione MARK/SPACE e portante consumer.
1 MSa/s	1 us	26.3	560	Ottimo compromesso pratico per la maggior parte delle prove.
2 MSa/s	0.5 us	52.6	1120	Ampio margine per timing e dettaglio della portante.
10 MSa/s	0.1 us	263.2	5600	Alta risoluzione; utile per studiare la forma d'onda, non necessaria per il solo protocollo.

Regola pratica

Prima assicurarsi che l'acquisizione contenga l'intero comando; solo dopo aumentare il sample rate fino al massimo compatibile con la memoria disponibile. Per il protocollo è molto più grave tagliare il frame che rinunciare a una parte della risoluzione sulla forma della portante.

La memoria richiesta è approssimativamente $N = F_s \times T$. Per una finestra di 300 ms:

Sample rate	Punti per 300 ms
250 kSa/s	75 000
500 kSa/s	150 000
1 MSa/s	300 000
2 MSa/s	600 000
10 MSa/s	3 000 000

Se si vuole osservare anche una pressione prolungata del tasto e gli eventuali frame di repeat, conviene aumentare la finestra temporale mantenendo un sample rate ragionevole invece di usare necessariamente la massima profondità disponibile.

5. Trigger e pretest automatico

Il comando non deve iniziare esattamente sul trigger. Il primo programma utilizza il riferimento temporale $t=0$ ricavato dal CSV e, prima dell'analisi completa, esegue sempre un pretest su una finestra limitata attorno al trigger.

Non è quindi necessario conoscere con precisione il campione nel quale inizia il comando: il pretest cerca automaticamente attività utile nelle finestre attorno al trigger e l'analisi completa assume poi come inizio del comando il primo burst MARK riconosciuto. In modalità automatica la versione 2.4 prova progressivamente le seguenti finestre:

Tentativo	Prima del trigger	Dopo il trigger
1	5 ms	30 ms
2	10 ms	60 ms
3	20 ms	120 ms
4	40 ms	250 ms

Per ogni tentativo il pretest verifica che vi siano dati numerici validi, stima il livello di riposo, valuta il rumore, riconosce automaticamente la polarità, ricava una soglia di analisi e controlla che emerga una portante plausibile. Se il tentativo non è sufficiente, la finestra viene allargata automaticamente.

Importante

Il pretest non scandisce arbitrariamente un file di durata illimitata alla ricerca di un comando. Lavora attorno al riferimento $t=0$ e amplia progressivamente la finestra fino ai limiti configurati. Se l'evento è molto lontano dal trigger, conviene spostare il trigger oppure specificare una finestra manuale più ampia.

Esempi:

```
GXTXT_analisi_VIR_protocollo_v2_4(2, 'pretest')
GXTXT_analisi_VIR_protocollo_v2_4(3, 'pretest', 0, [5 40])
GXTXT_analisi_VIR_protocollo_v2_4(3, 'pretest', 0, 100)
```

Nel terzo esempio un singolo valore 100 significa 5 ms prima del trigger e 100 ms dopo.

6. Primo programma: analisi della forma d'onda

Il programma GXTXT_analisi_VIR_protocollo_v2_4.m esegue l'intera trasformazione dalla forma d'onda campionata alla sequenza MARK/SPACE.

6.1 Parametri da configurare

Per l'uso normale sono sufficienti soprattutto i primi due parametri nella sezione CONFIGURAZIONE UTENTE del sorgente:

```
cfg.csvFile = 'nome_acquisizione.csv';
cfg.outputDir = 'analisi_output';
```

Altri parametri utili in casi particolari:

Parametro	Default v2.4	Quando modificarlo
cfg.nominalCarrier_Hz	38000	Se si lavora con portanti molto lontane dalla normale banda consumer 36-40 kHz.
cfg.minSignalAmplitude_V	0.050	Ridurre se il segnale utile è inferiore a 50 mV ma resta nettamente sopra il rumore.
cfg.thresholdFraction	0.30	Modificare se la forma d'onda richiede una soglia relativa diversa.
cfg.noiseSigmaMultiplier	8	Modificare solo in presenza di rumore anomalo o falsi fronti.
cfg.burstGapPeriods	3.0	Determina quando una pausa tra impulsi separa due burst.
cfg.useCache	true	Disabilitare se non si desidera creare il file cache Octave.

Nota sulla soglia

La soglia del 30% è una soglia di riconoscimento applicata alla forma d'onda elettrica. Non rappresenta il duty-cycle ottico del trasmettitore e non richiede che il segnale abbia una specifica ampiezza assoluta.

6.2 Modalità e verbosity

Chiamata	Significato
GXTXT_analisi_VIR_protocollo_v2_4()	Analisi completa, verbosity 2, worker non forzati.
GXTXT_analisi_VIR_protocollo_v2_4(3)	Analisi completa con diagnostica molto dettagliata.
GXTXT_analisi_VIR_protocollo_v2_4(2,'pretest')	Solo pretest rapido, poi termina.
GXTXT_analisi_VIR_protocollo_v2_4(1,'full',8)	Richiede 8 worker; nella v2.4 il percorso principale resta soprattutto vettoriale e il parallelismo non garantisce accelerazioni.

Livelli di verbosity: 0 = solo errori e risultato finale; 1 = fasi principali; 2 = avanzamento e risultati intermedi; 3 = diagnostica dettagliata. Durante letture lunghe il programma emette heartbeat periodici e legge il CSV a blocchi.

6.3 Output generati

File	Contenuto
eventi_mark_space.csv	Sequenza ordinata degli eventi MARK/SPACE con inizio, fine, durata e cicli di portante.
burst.csv	Dettaglio dei burst riconosciuti e dei cicli contenuti.
riepilogo_VIR.txt	Riepilogo di acquisizione, portante, comando e classi temporali.
analisi_VIR_v2_4.log	Log diagnostico completo.
01_vir_comando_completo.png	Vista globale dell'acquisizione.
02_vir_portante_zoom.png	Dettaglio di alcuni periodi della portante.
03_mark_space_ricostruito.png	Ricostruzione logica della presenza/assenza della portante.
04_durate_mark_space.png	Andamento delle durate di MARK e SPACE.
*_cache_octave.mat	Cache opzionale per rilanci più rapidi sullo stesso CSV.

7. Adattare l'importazione CSV a un altro oscilloscopio

La parte di analisi è indipendente dal modello di oscilloscopio; il punto specifico dello strumento è il parser CSV. La versione 2.4 è stata scritta per un formato Rigol nel quale l'header contiene i campi Start e Increment e le righe dati hanno la forma Sequence,Volt,.

Per supportare un altro formato, l'obiettivo è fornire al resto del programma quattro informazioni coerenti:

Variabile	Significato	Requisito
seq	Indice del campione.	Vettore numerico finito e crescente; idealmente 0,1,2,...
v	Valore della forma d'onda.	Un campione di tensione per ogni elemento di seq.
tStart	Tempo del campione seq=0 rispetto al trigger.	Secondi; negativo se il primo campione precede il trigger.
dt	Intervallo tra campioni.	Secondi/campione, positivo e costante.

Nella versione corrente le due aree da adattare sono:

- la funzione `parse_header_rigol(headerNames, headerValues)`, che ricava `tStart` e `dt`;
- il blocco di lettura dentro `leggi_csv_con_pretest(cfg)`, che usa `textscan` per leggere Sequence e Volt a blocchi.

```
% Formato Rigol attuale, concettualmente:
[tStart, dt] = parse_header_rigol(headerNames, headerValues);
% ...
dati = textscan(fid, '%f,%f*[\n]', cfg.readChunkRows, ...);
% prima colonna -> seq, seconda colonna -> v
```

Se il CSV del proprio oscilloscopio contiene direttamente una colonna tempo e una colonna tensione, un adattatore semplice può ricavare `dt` come mediana delle differenze temporali, porre `tStart` uguale al tempo del primo campione e generare `seq = (0:N-1)`. Prima di farlo è necessario verificare che il campionamento sia sostanzialmente uniforme.

Trigger nei CSV di altri strumenti

Il pretest presume che $t=0$ rappresenti il trigger o comunque un riferimento temporale vicino al comando. Se il CSV non conserva questa informazione, il parser deve ricostruirla oppure assegnare un $tStart$ coerente in modo che l'evento cada nella finestra del pretest. Non è necessario che il comando inizi a $t=0$.

Per file molto grandi è consigliabile conservare la lettura a blocchi della versione originale, perché permette di eseguire il pretest prima di completare operazioni pesanti e di mostrare l'avanzamento durante l'importazione.

8. Secondo programma: identificazione del protocollo

GXTXT_identifica_protocollo_IR_v1.m legge il file eventi_mark_space.csv prodotto dal primo programma e confronta la sequenza con alcune famiglie temporali note.

Uso tipico:

```
GXTXT_identifica_protocollo_IR_v1('analisi_output/eventi_mark_space.csv')

% Portante fornita esplicitamente e verbosity 3:
GXTXT_identifica_protocollo_IR_v1('analisi_output/eventi_mark_space.csv', 38.0, 3)
```

Se la frequenza di portante non viene passata come secondo argomento, il programma prova a leggerla automaticamente da riepilogo_VIR.txt nella stessa directory. Se non è disponibile, la portante viene esclusa dal punteggio e l'identificazione prosegue usando i timing disponibili.

Famiglia supportata nella v1	Tipo di codifica
NEC family	Pulse distance
Samsung	Pulse distance
Panasonic / Kaseikyo	Pulse distance
Sony SIRC	Pulse width
Philips RC5	Bifase
Philips RC6	Bifase con header

Il risultato contiene una classifica dei candidati ordinata per score. Per i protocolli pulse-distance e pulse-width il programma tenta anche la ricostruzione dei bit. Per frame NEC-family da 32 bit raggruppa inoltre i bit in byte LSB-first ed esegue controlli di complemento quando applicabili.

Interpretazione dello score

Uno score basso indica una migliore corrispondenza tra i timing osservati e quelli nominali usati dal classificatore. Le etichette di confidenza sono euristiche. Non vanno interpretate come probabilità e non eliminano la necessità di verificare struttura del frame, numero di bit, repeat e coerenza del payload.

9. Procedura rapida

1. Scegliere un punto del circuito nel quale siano ancora visibili i cicli della portante IR.
2. Impostare l'oscilloscopio in modo da acquisire un comando completo e un breve tratto a riposo prima dell'evento.

3. Usare un sample rate adeguato: circa 1 MSa/s è un buon punto di partenza per portanti consumer intorno a 36-40 kHz; aumentarlo solo se serve più dettaglio.
4. Esportare il canale in CSV.
5. Se il formato non è Rigol, adattare esclusivamente il parser in modo da ottenere seq, v, tStart e dt coerenti.
6. Impostare `cfg.csvFile` e `cfg.outputDir` nel primo programma.
7. Eseguire `GXTXT_analisi_VIR_protocollo_v2_4(2,'pretest')` e verificare che il pretest riconosca segnale e portante.
8. Eseguire l'analisi completa.
9. Controllare almeno `riepilogo_VIR.txt` e il grafico MARK/SPACE.
10. Eseguire `GXTXT_identifica_protocollo_IR_v1` sul file `eventi_mark_space.csv` e valutare il candidato migliore insieme alla struttura ricostruita.

10. Risoluzione dei problemi

Sintomo	Cause probabili	Intervento
Campi Start/Increment non trovati	CSV di un altro oscilloscopio o header differente.	Adattare <code>parse_header_rigol</code> oppure sostituirla con un parser specifico.
PRETEST: escursione inferiore al minimo	Segnale piccolo oppure soglia minima di 50 mV troppo alta.	Migliorare la scala/acquisizione o ridurre <code>cfg.minSignalAmplitude_V</code> se il SNR è comunque buono.
Non emerge una portante plausibile	Sample rate troppo basso, punto già demodulato, clipping, trigger lontano o portante insolita.	Aumentare Fs, cambiare nodo, spostare trigger o modificare <code>cfg.nominalCarrier_Hz</code> .
Troppi pochi impulsi	Comando tagliato o soglia non adatta.	Aumentare la finestra temporale; controllare <code>thresholdFraction</code> e rumore.
MARK/SPACE frammentati	Rumore o ringing generano falsi fronti.	Controllare forma d'onda; valutare <code>thresholdFraction</code> , <code>noiseSigmaMultiplier</code> e <code>burstGapPeriods</code> .
Frame incompleto	Acquisizione troppo corta.	Aumentare la finestra, anche riducendo il sample rate per contenere la memoria.
Protocollo ambiguo	Famiglie diverse condividono timing simili.	Valutare bit count, header, repeat, payload e controlli di complemento; non affidarsi solo allo score.
File molto pesante	Sample rate o durata eccessivi rispetto allo scopo.	Ridurre Fs o durata; mantenere solo la risoluzione necessaria per distinguere portante e burst.

11. Limiti attuali

- Il primo programma richiede che i singoli cicli della portante siano ancora presenti nella forma d'onda.
- Il parser CSV fornito non è universale e può richiedere un adattamento per strumenti differenti.
- Il classificatore copre un insieme limitato di protocolli consumer e varianti temporali.
- Protocolli appartenenti alla stessa famiglia possono non essere distinguibili dal solo timing di un singolo frame.
- I valori di soglia e i parametri di raggruppamento sono configurabili e possono richiedere taratura su segnali molto diversi da quelli tipici IR consumer.

Questi limiti sono intenzionali: il pacchetto è pensato come strumento di misura e reverse engineering, non come sostituto di un ricevitore IR integrato o di un decoder universale.

12. Licenza del software

Licenza scelta: GNU General Public License, versione 3 o successiva (GPL-3.0-or-later).

La GPL consente di usare, studiare, modificare e ridistribuire il software secondo i termini della licenza. Quando il software o una versione modificata viene distribuito, devono essere rispettati gli obblighi previsti dalla GPL, inclusa la disponibilità del codice sorgente corrispondente nei casi previsti e la conservazione delle informazioni di licenza.

Identificatore SPDX consigliato da inserire nei sorgenti:

```
% SPDX-License-Identifier: GPL-3.0-or-later
```

Esempio di intestazione sintetica:

```
% GX-TXT - software di analisi IR
% Copyright (C) 2026 Giux-Lab
% SPDX-License-Identifier: GPL-3.0-or-later
% Questo programma e' software libero: puo' essere ridistribuito e/o modificato
% secondo i termini della GNU General Public License v3 o successiva.
```

Testo ufficiale della licenza: <https://www.gnu.org/licenses/gpl-3.0.html>

Appendice A - Convenzioni temporali usate dai file intermedi

Campo	Unità	Significato
start_s	s	Inizio dell'evento rispetto al riferimento temporale dell'acquisizione.
end_s	s	Fine dell'evento.
durata_us	us	Durata dell'evento in microsecondi.
cicli_portante	conteggio	Numero di cicli rilevati nei MARK; zero per gli SPACE.

Il secondo programma usa soprattutto la sequenza dei tipi MARK/SPACE e la colonna durata_us. Gli altri campi restano utili per controlli, grafici e sviluppi futuri.

Appendice B - Checklist prima di condividere un'acquisizione

- Il CSV contiene un solo canale utile oppure il parser sa quale colonna leggere.
- Il sample interval è noto e corretto.
- Il riferimento al trigger è noto o ricostruito.
- Il frame è completamente contenuto nell'acquisizione.
- La portante presenta almeno diversi campioni per periodo.
- Il segnale non è fortemente clippato.
- Il pretest viene superato prima di avviare l'analisi completa.
- Il grafico MARK/SPACE è coerente a vista con il segnale originale.

Fine del manuale