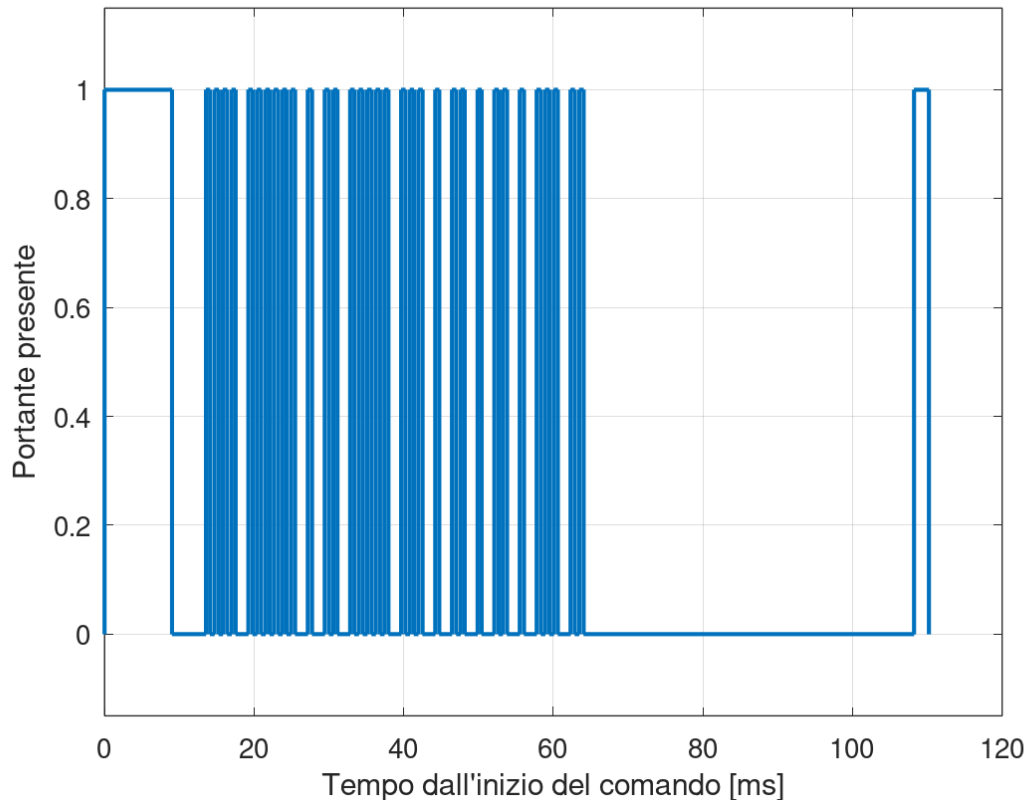


GX-TXT-v3

Manuale d'uso dello script GNU Octave per l'analisi del nodo VIR e dei protocolli IR

GXTXT_analisi_VIR_protocollo_v2_2.m

GX-TXT-v3 - Ricostruzione temporale MARK / SPACE



Versione del manuale: 1.0 - Script: 2.2 - Agosto 2026

Scopo: Analizzare acquisizioni CSV Rigol a un solo canale del nodo VIR, misurare la portante IR, ricostruire MARK e SPACE, contare i cicli nei burst e produrre dati e grafici utili alla successiva decodifica del protocollo.

1. A cosa serve lo script

Lo script GXTXT_analisi_VIR_protocollo_v2_2.m e' stato sviluppato per elaborare acquisizioni profonde del nodo VIR della GX-TXT-v3. L'ingresso previsto e' un CSV esportato da un oscilloscopio Rigol con un solo canale analogico.

L'obiettivo non e' assegnare immediatamente un nome al protocollo del telecomando, ma ricostruire prima i dati sperimentali: livello di riposo, polarita' degli impulsi, frequenza della portante, durata dei burst, durata delle pause e numero di cicli contenuti nei burst.

La versione 2.2 introduce un pretest obbligatorio, output verboso con heartbeat, lettura del CSV a blocchi, controlli sul parser e una cache binaria per rendere piu' rapidi i rilanci sullo stesso file.

Principio di lavoro: MARK = portante presente. SPACE = portante assente. Lo script misura prima i singoli cicli della portante e poi raggruppa cicli consecutivi in MARK separati dagli SPACE.

2. Requisiti e formato dei dati

- GNU Octave. Lo script non richiede il package parallel per l'analisi di un singolo file; se disponibile, ne viene comunque verificata la presenza.
- CSV Rigol a un solo canale, esportato con le informazioni Start e Increment nell'intestazione.
- Acquisizione in DC coupling consigliata per conservare il livello reale del nodo VIR e consentire una stima corretta della baseline.
- Pretrigger sufficiente a fornire un tratto di baseline prima dell'inizio del comando.

Esempio di intestazione supportata:

```
X,CH1,Start,Increment,
Sequence,Volt,-9.800002e-03,1.000000e-07
0,8.84e+00,
1,8.84e+00,
2,8.88e+00,
```

L'asse temporale viene ricostruito come: $t = \text{Start} + \text{Sequence} \times \text{Increment}$

3. Avvio rapido

Prima dell'uso modificare nella sezione CONFIGURAZIONE UTENTE almeno queste due righe:

```
cfg.csvFile = 'VIR_10x_DC_10ms_1.2M.csv';
cfg.outputDir = 'analisi_VIR_protocollo_v2_2';
```

3.1 Pretest consigliato

Per un nuovo CSV e' consigliabile eseguire prima soltanto il pretest:

```
GXTXT_analisi_VIR_protocollo_v2_2(3,'pretest')
```

Il pretest termina prima dell'analisi completa e verifica rapidamente che il file sia leggibile e che nel tratto iniziale esista un segnale compatibile con la portante attesa.

3.2 Analisi completa

```
GXTXT_analisi_VIR_protocollo_v2_2(2,'full')
```

La modalita' full esegue comunque di nuovo il pretest prima di procedere. Se il controllo fallisce, l'analisi pesante viene interrotta.

3.3 Esecuzione con impostazione server

```
GXTXT_analisi_VIR_protocollo_v2_2(1, 'full', 8)
```

Il terzo parametro indica il numero di worker richiesti. Nella versione 2.2 il percorso di calcolo del singolo CSV resta principalmente vettorializzato e seriale: il parametro viene verificato e registrato, ma non forza la suddivisione del singolo vettore tra worker. E' predisposto per estensioni future e analisi multi-file.

4. Livelli di verbosita'

Livello	Uso tipico	Output in console
0	Automazione	Solo errori e risultato finale
1	Server	Fasi principali
2	PC - default	Fasi, avanzamento e risultati intermedi
3	Debug	Diagnostica dettagliata, chunk, soglie e controlli

Il file di log viene comunque scritto con livello diagnostico completo, indipendentemente dalla verbosita' scelta per la console.

5. Cosa controlla il pretest

- Validita' dell'header e dei campi Start/Increment.
- Lettura corretta delle righe Rigol nel formato Sequence, Volt, e scarto di eventuali righe NaN/Inf.
- Regolarita' dell'indice Sequence.
- Stima della baseline usando il pretrigger, quando disponibile.
- Riconoscimento automatico della polarita': impulsi verso il basso o verso l'alto.
- Escursione minima del segnale rispetto alla baseline.
- Stima robusta del rumore sul pretrigger e scelta della soglia di analisi.
- Presenza di un numero sufficiente di fronti.
- Presenza di intervalli compatibili con una portante vicina alla frequenza nominale, di default 38 kHz.

Perche' e' importante: I CSV profondi possono richiedere molto tempo per essere letti. Il pretest evita di attendere l'analisi completa per scoprire errori di formato, soglia o assenza del segnale.

6. Parametri principali modificabili

Parametro	Default	Significato
cfg.nominalCarrier_Hz	38000	Frequenza di riferimento usata dal pretest per cercare la portante.
cfg.pretestSamples	300000	Numero di campioni iniziali usati dal pretest.
cfg.readChunkRows	100000	Numero di righe lette per blocco durante l'importazione del CSV.

cfg.progressInterval_s	3	Intervallo minimo tra i messaggi di heartbeat.
cfg.minSignalAmplitude_V	0.050	Escursione minima richiesta per considerare plausibile il segnale.
cfg.thresholdFraction	0.30	Frazione dell'escursione usata come soglia logica della portante.
cfg.noiseSigmaMultiplier	8	Soglia minima espressa in multipli del rumore stimato.
cfg.burstGapPeriods	3.0	Una pausa superiore a questo numero di periodi separa due MARK.
cfg.clusterRelativeTolerance	0.20	Tolleranza relativa per raggruppare durate simili in classi.
cfg.useCache	true	Abilita la cache binaria .mat del CSV già interpretato.

7. Flusso dell'analisi completa

- 1. Pretest e lettura CSV** - Il file viene letto a blocchi. La console mostra campioni letti, percentuale del file e tempo trascorso.
- 2. Preparazione globale** - Vengono ricostruiti asse temporale, sample rate e durata totale.
- 3. Riconoscimento portante** - Il segnale viene confrontato con la soglia; i fronti permettono di ricavare periodo, frequenza, pulse width e duty-cycle.
- 4. MARK e SPACE** - Gli impulsi consecutivi vengono raggruppati in MARK. Le pause superiori alla soglia burstGapPeriods separano i burst e costituiscono gli SPACE.
- 5. Classi temporali** - Le durate simili vengono raggruppate per evidenziare leader, MARK normali, SPACE corti/lunghi e altre strutture ricorrenti.
- 6. Salvataggio numerico** - Vengono creati CSV e riepilogo TXT.
- 7. Grafici** - Vengono generati overview, zoom della portante, ricostruzione MARK/SPACE e grafico delle durate.

8. File prodotti

File	Contenuto
analisi_VIR_v2_2.log	Log completo con timestamp, tempi trascorsi e diagnostica.
riepilogo_VIR.txt	Riepilogo sintetico di acquisizione, VIR, portante, comando e classi MARK/SPACE.
eventi_mark_space.csv	Sequenza completa degli eventi MARK e SPACE con start, end, durata e cicli.
burst.csv	Una riga per ogni MARK: durata, cicli rilevati, cicli stimati dalla durata e pulse width media.
*_cache_octave.mat	Cache binaria dei campioni importati; viene riutilizzata se più recente del CSV.
01_vir_comando_completo.png	Visione complessiva del comando con riduzione min/max.
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	Durate dei MARK e degli SPACE in ordine di apparizione.

9. Esempio di risultato reale

Su una acquisizione di prova della GX-TXT-v3 lo script ha elaborato 1.200.000 campioni a 10 MSa/s, per circa 120 ms di registrazione. Il risultato ha mostrato una portante di 38,0228 kHz, periodo mediano di 26,3 us e duty-cycle stimato del 57,4%.

Nella stessa acquisizione sono stati riconosciuti 35 MARK. La classe principale dei MARK era centrata attorno a 568 us; gli SPACE piu' frequenti erano attorno a 591 us e 1.697 ms. Questi numeri sono un esempio di come lo script prepari i dati necessari alla successiva ricostruzione del protocollo.

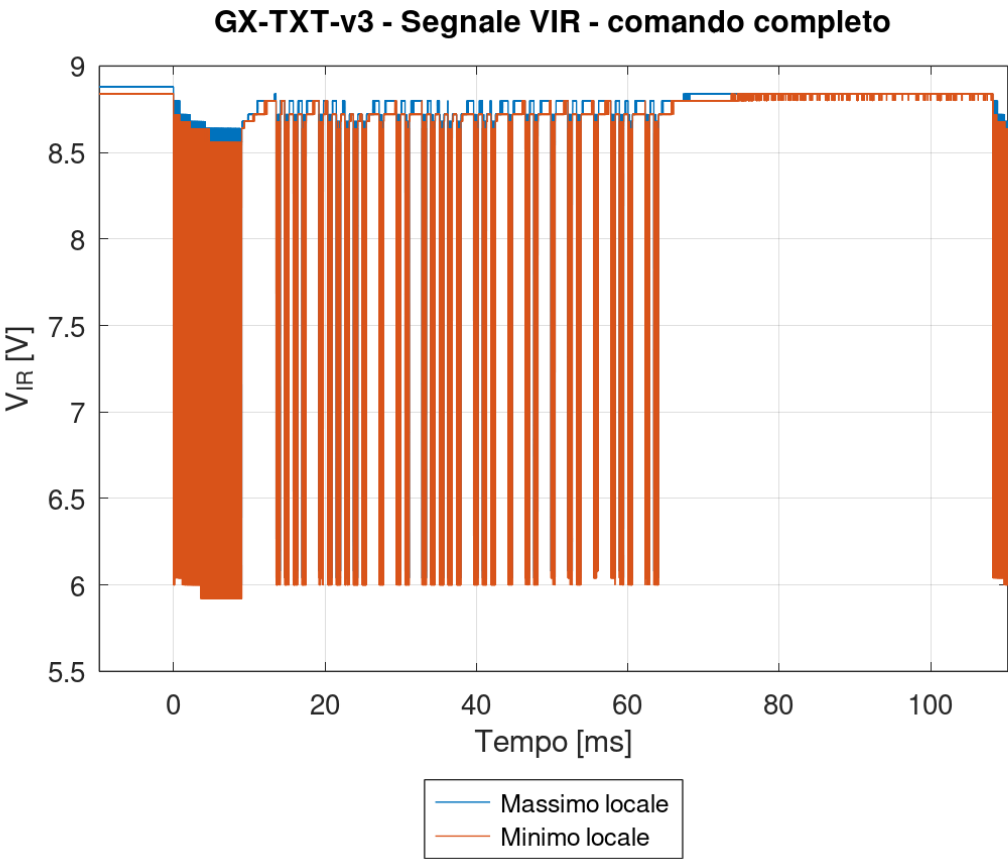


Figura 1 - Visione complessiva del segnale VIR acquisito.

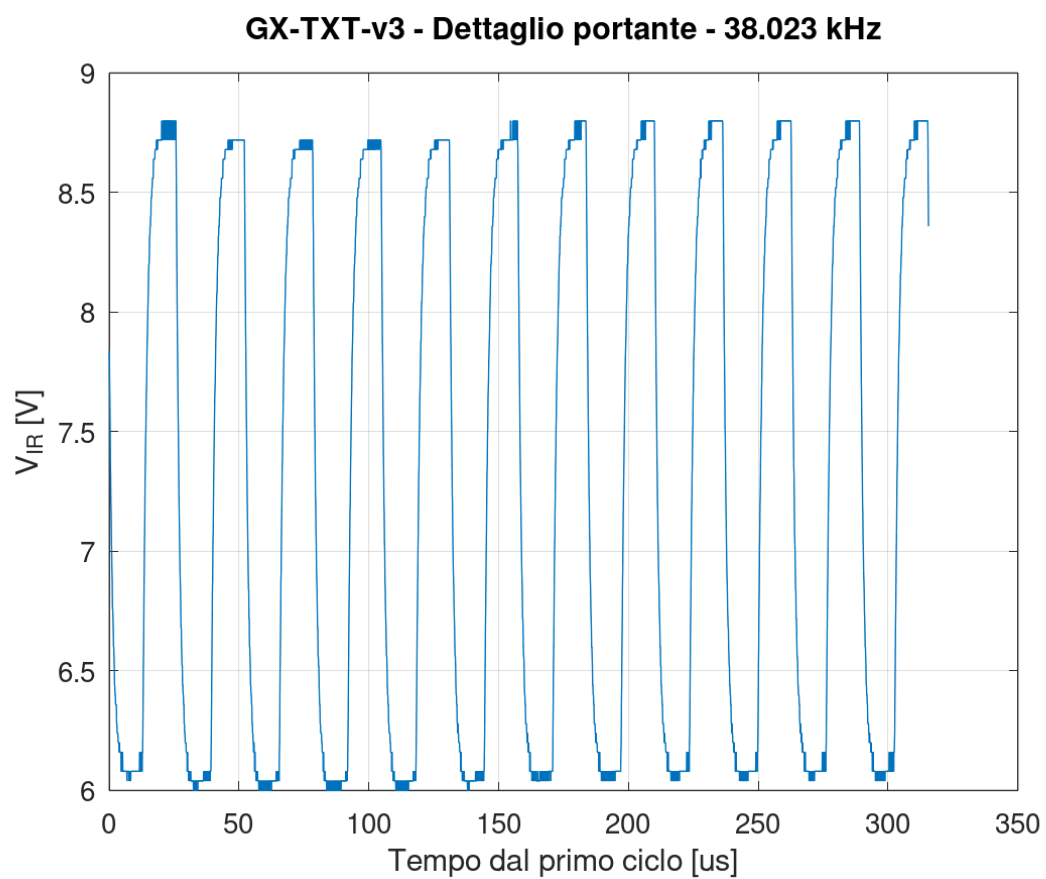


Figura 2 - Dettaglio della portante misurata sul nodo VIR.

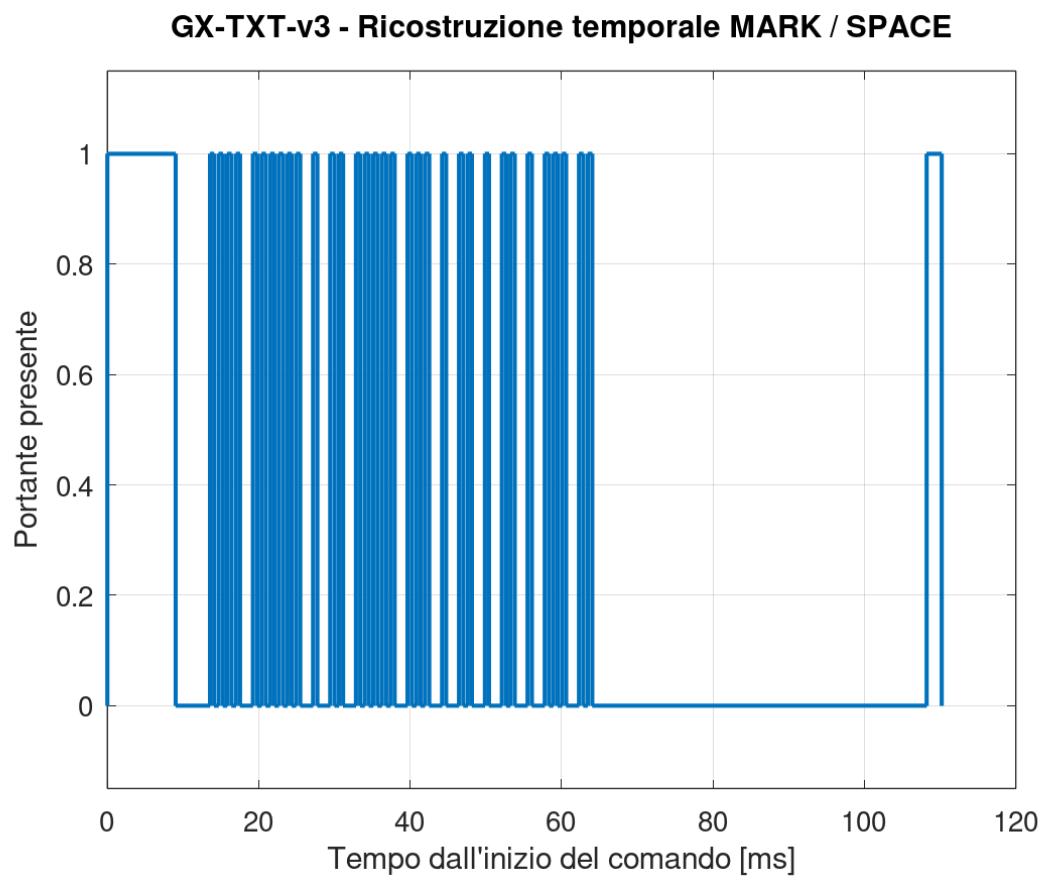


Figura 3 - Ricostruzione temporale MARK/SPACE prodotta dallo script.

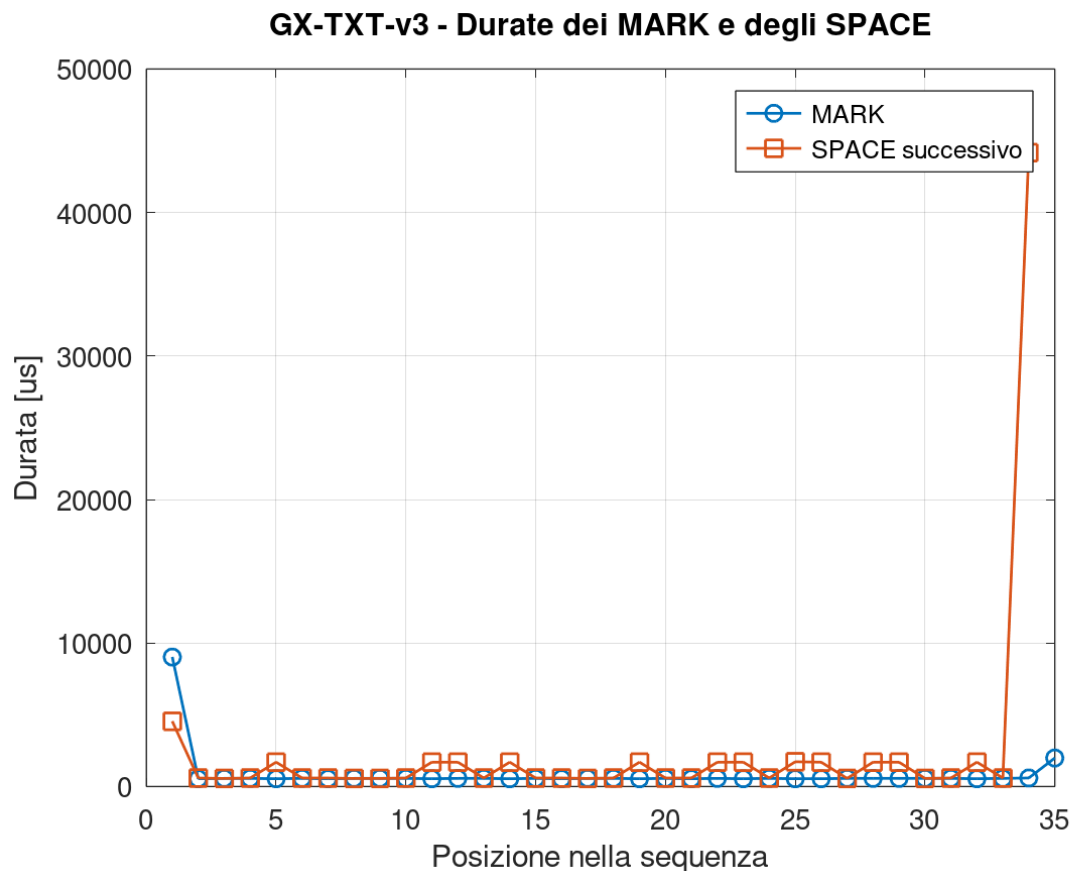


Figura 4 - Durate dei MARK e degli SPACE in sequenza.

10. Risoluzione dei problemi

CSV non trovato: Controllare `cfg.csvFile` e la cartella di lavoro corrente di Octave.

PRETEST FALLITO: escursione insufficiente: Verificare che il tratto iniziale contenga davvero il comando, che il canale sia corretto e che `minSignalAmplitude_V` non sia troppo alto.

PRETEST: pochi fronti: Controllare polarità, soglia, campionamento e presenza reale della portante.

Portante non vicina a 38 kHz: Se il telecomando usa una portante differente, modificare `cfg.nominalCarrier_Hz`. Prima verificare comunque il segnale con l'oscilloscopio.

Lettura molto lenta: Usare il pretest prima del full. Dal secondo lancio sullo stesso CSV la cache `.mat` evita il parsing completo del file.

Risultati MARK/SPACE poco plausibili: Valutare `thresholdFraction` e `burstGapPeriods`. Modificare un parametro alla volta e confrontare con l'oscillogramma.

Cache non desiderata: Impostare `cfg.useCache = false` oppure cancellare il file `*_cache_octave.mat`.

Output troppo rumoroso: Usare `verbosity 1` sul server o 0 in automazione. Il log su file resta dettagliato.

11. Limiti della versione 2.2

- Il formato atteso è quello Rigol a un solo canale con colonne Sequence e Volt e campi Start/Increment nell'header.
- Lo script ricostruisce la struttura temporale ma non assegna automaticamente il nome del protocollo e non decodifica ancora i bit in byte/comandi.
- Il parametro `nWorkers` è predisposto per il lavoro futuro, ma l'analisi del singolo CSV resta principalmente vettorializzata e non viene distribuita tra più worker.
- La classificazione delle durate è euristica. I risultati devono essere verificati rispetto alla forma d'onda e al contesto sperimentale.

- Le soglie di default sono state pensate per le acquisizioni VIR della GX-TXT-v3 e possono richiedere adattamento su segnali molto diversi.

12. Flusso consigliato per una nuova acquisizione

1. Esportare il CSV Rigol a un solo canale.
2. Impostare `cfg.csvFile`.
3. Eseguire: `GXTXT_analisi_VIR_protocollo_v2_2(3,'pretest')`
4. Controllare baseline, min/max, polarità e portante preliminare.
5. Se il pretest è coerente, eseguire: `GXTXT_analisi_VIR_protocollo_v2_2(2,'full')`
6. Aprire `riepilogo_VIR.txt` e i quattro grafici.
7. Verificare `eventi_mark_space.csv` prima di interpretare il protocollo.
8. Conservare CSV, cache, risultati e parametri usati insieme alla misura.

Regola pratica: Prima si ricostruiscono i tempi dal segnale reale; solo dopo si confrontano MARK e SPACE con le specifiche di protocolli noti. In questo modo l'identificazione del protocollo deriva dalla misura e non da un'ipotesi iniziale.

Fine del manuale