

ObjectDump

1. Installazione

Per installare il programma è sufficiente eseguire il Makefile con il comando “make”.

Per la compilazione sono necessarie le stesse librerie di Wavedump: se nel computer è già installato Wavedump, anche objectDump viene compilato senza problemi.

Dopo la compilazione vengono prodotti tre eseguibili: “objectdump”, “objectdumpclient” e “decode”. I primi due vengono posti nella cartella principale, il terzo nella sottocartella RawData.

2. Avvio dei programmi

2.1 Lato server

Il lato server del programma può essere lanciato con il seguente comando:

```
./objectdump [-m {user | tcp | all}] [-f configurationfilepath] [-d rawdatapath] [-l logfilepath]
```

L'opzione -m permette di scegliere la modalità di inserimento dell'input:

- m user: il programma accetta input solo da tastiera

- m tcp: il programma accetta input solo via TCP/IP. In questo caso è lanciato in modalità demone.

- m all: il programma accetta input sia da tastiera sia via TCP/IP.

-m all è l'opzione di default.

L'opzione -f permette di specificare il path del file di configurazione. Di default esso è “ConfigurationFile”, posto nella cartella principale del programma.

L'opzione -d permette di specificare il path del file dove salvare i dati non decodificati. Di default esso è “data.txt” nella sottocartella RawData.

L'opzione -l permette di specificare il path del logfile. Di default esso è “LogFile”, posto nella cartella principale del programma.

2.2 Lato client

Il lato client del programma può essere lanciato con il comando:

```
./objectdumpclient -i serveripaddress
```

L'opzione -i permette di inserire l'indirizzo ip del server. Per testare il programma in locale è sufficiente digitare:

```
./objectdumpclient -i 127.0.0.1
```

2.3 Programma di decoding

Una delle idee su cui è fondato il progetto è quella di salvare su disco i dati acquisiti senza decodificarli. Il programma “decode” permette di decodificarli.

Attualmente, a causa della politica di gestione delle librerie di decoding dell'azienda CAEN, per lanciare il programma decode è necessario comunque interfacciarsi col digitizer.

Il programma decode è presente nella sottocartella RawData. Nella stessa sottocartella è presente il file “DigitizerConfig.conf”.

Per lanciare decode è necessario impostare il file DigitizerConfig.conf modificando il parametro OPEN (v. File di configurazione di objectDump) per permettere l'apertura del digitizer.

Il programma può essere eseguito con:

```
./decode -r rawdatapath -s rawdatasize -o decodeeventpath
```

Tutte le opzioni sono necessarie.

L'opzione -r specifica il path del file con i dati non decodificati.

L'opzione -s specifica il path del file con le dimensioni dei dati non decodificati

N.B. Quando objectDump salva su disco i dati non decodificati, crea automaticamente questo file accanto a quello dove sono salvati i rawdata aggiungendo i caratteri “sz” alla fine di questo.

Esempio: di default objectDump salva i dati non decodificati nel file data.txt della sottocartella rawdata e le dimensioni nel file data.txtsz della sottocartella rawdata.

L'opzione -o specifica il path del file contenente i dati decodificati.

Esempio:

```
./decode -r data.txt -s data.txtsz -o events.txt
```

3. Comandi

- **init**: apre il digitizer.
- **setup**: imposta il digitizer.
- **start**: inizia la DAQ.
- **stop**: interrompe la DAQ.
- **prestart**: inizia il preprocessing.
- **prestop**: interrompe il preprocessing.
- **vistart [channelnumber]**: visualizza i dati ricevuti nel canale indicato da channelnumber.
- **vistop**: ferma la visualizzazione.

- **rawstart**: inizia a salvare su disco i dati acquisiti.
- **rawstop**: interrompe il salvataggio su disco dei dati ricevuti.
- **close**: chiude il digitizer.
- **send**: invia un software trigger.
- **help**: visualizza la lista dei comandi disponibili.
- **check**: stampa il contenuto delle impostazioni lette nel file di configurazione e controlla la presenza e la correttezza dei parametri fondamentali.
- **chkconf**: stampa il contenuto delle impostazioni lette nel file di configurazione.
- **write register 0x[register] 0x[data]**: scrive nel registro indicato da register i dati indicati da data.
- **read register 0x[register]**: legge il registro indicato da register.
- **-f [conf file path]**: imposta il path del file di configurazione.
- **-d [data file path]**: imposta il path del file dove vengono salvati i dati non decodificati.
- **-l [log file path]**: cambio il path del logfile.
- **print**: stampa il contenuto della configurazione interna del programma.
- **print files**: stampa il path del file di configurazione, del file dove vengono salvati i dati non decodificati e del log file.
- **status**: stampa lo status dei thread del programma (cioè se sono attivi o spenti).
- **more**: stampa il contenuto del log file.
- **exit/quit**: esci dal programma.

Tutti i comandi possono essere inviati anche con objectdumpclient via TCP/IP con la differenza che exit interrompe objectdumpclient, non objectdump.

4. File di configurazione

Il file di configurazione di objectdump “ConfigurationFile” è posizionato di default nella cartella del programma.

L'utente può modificarne il path o lanciando il programma con l'opzione *-f configurationfilepath* oppure inserendo il comando *-f configurationfilepath* durante l'esecuzione del programma.

4.1 Impostazioni comuni a tutti i canali

OPEN usb|pci LinkNumber NodeNumber BaseAddress

Il parametro open consente di specificare le informazioni necessarie per aprire il digitizer. Se una di queste informazioni non è necessaria (ad esempio il BaseAddress), occorre settarla con il parametro 0.

Esempio: OPEN PCI 0 0 0

MAX_NUM_EVENTS_BLT maximum_number_of_events

Il parametro MAX_NUM_EVENTS_BLT imposta il numero di eventi massimo che può essere trasferito in un block transfer.

Esempio: MAX_NUM_EVENTS_BLT 2

RECORD_LENGTH number_of_samples

Il parametro RECORD_LENGTH indica il numero di campioni da acquisire ad ogni trigger.

Esempio: RECORD_LENGTH 1024

POST_TRIGGER value

Il parametro POST_TRIGGER indica la dimensione del post-trigger in percentuale della grandezza di record_length. Nel caso dei digitizer x742, c'è un ulteriore delay di 35 ns.

Esempio: POST_TRIGGER 10

TEST_PATTERN yes|no

Il parametro TEST_PATTERN permette di sostituire alla ADC un'onda triangolare di test con un range da 0 al massimo acquisibile.

Esempio: TEST_PATTERN yes

FPIO_LEVEL ttl|nim

Il parametro FPIO_LEVEL indica il tipo dell'input/output dei front panel LEMO connectors.

Esempio: FPIO_LEVEL nim

DECIMATION_FACTOR number_of_samples

Il parametro DECIMATION_FACTOR, significativo solo per i digitizers della famiglia X740, specifica il decimation factor dell'acquisizione.

Esempio: DECIMATION_FACTOR 1

ENABLED_FAST_TRIGGER_DIGITIZING yes|no

Il parametro ENABLED_FAST_TRIGGER_DIGITIZING, significativo solo per i digitizers della famiglia x742, indica se digitalizzare e rendere disponibili nel readout i segnali acquisiti dai canali di fast triggering.

Esempio: ENABLED_FAST_TRIGGER_DIGITIZING yes

FAST_TRIGGER acquisition_only|disabled

Il parametro FAST_TRIGGER permette di usare l'input proveniente dai canali di fast triggering come segnale di trigger per, rispettivamente, i gruppi 0-1 e 2-3.

Esempio: FAST_TRIGGER acquisition_only

EXTERNAL_TRIGGER acquisition_only|acquisition_and_trgout|disabled

Il parametro EXTERNAL_TRIGGER permette di impostare il modo con cui usare il segnale di trigger.

Esempio: EXTERNAL_TRIGGER acquisition_only

ENABLE_DES_MODE yes|no

Il parametro ENABLE_DES_MODE permette di abilitare la Dual Edge Sampling (DES) mode per i digitizers delle serie 731 e 751. Quando la DES mode è attiva, solo metà dei canali è abilitata (pari per la serie 731, dispari per la serie 751).

Esempio: ENABLE_DES_MODE yes

GNUPLOT_PATH gnuplotcommand|gnuplotprogrampath

Il parametro GNUPLOT_PATH indica il comando che objectDump utilizzerà per lanciare gnuplot. Quindi, il parametro deve essere impostato o con il comando utilizzato nella shell per lanciare gnuplot o con il path assoluto del programma gnuplot.

Esempio: GNUPLOT_PATH gnuplot

DRS4_FREQUENCY 0|1|2

Il parametro DRS4_FREQUENCY, significativo solo per i digitizers della famiglia x742, permette di impostare la frequenza di campionamento.

0---> 5 Ghz (valore di default)

1---> 2.5 Ghz
2---> 1 Ghz.

Esempio: DRS4_FREQUENCY 1 (cioè viene impostata la frequenza di campionamento a 2.5 Ghz).

GROUP_ENABLE_MASK groupenablemask

Il parametro GROUP_ENABLE_MASK consente di impostare quali gruppi di canali saranno presenti nell'acquisizione. Questo parametro ha senso per le famiglie x740, x742 e x743.

Esempio: GROUP_ENABLE_MASK 0x9. In questo caso saranno presenti solo il gruppo 0 e il gruppo 3 (0x9 = 1001 in base 2).

CHANNEL_ENABLE_MASK channelenablemask

Il parametro CHANNEL_ENABLE_MASK consente di impostare quali canali saranno presenti nell'acquisizione. Questo parametro non ha senso per le famiglie x740, x742 e x743.

Esempio: CHANNEL_ENABLE_MASK 0x3. In questo caso saranno presenti solo il canale 0 e il canale 2 (0x3 = 11 in base 2).

ALL_DC_OFFSET dc_offset

Il parametro ALL_DC_OFFSET consente di eseguire lo shift dell'input di tutti i canali disponibili della dimensione indicata in dc_offset. Per avere maggiori informazioni sul significato di tale dimensione, consultare la documentazione tecnica del digitizer.

Esempio: ALL_DC_OFFSET 0x3fff

ALL_TRIGGER_THRESHOLD triggerthreshold

Il parametro ALL_TRIGGER_THRESHOLD consente di impostare su tutti i canali disponibili la soglia di self triggering indicata da triggerthreshold.

Esempio: ALL_TRIGGER_THRESHOLD 0x0100

SELF_TRIGGER_ENABLE_MASK selftriggerenablemask acquisition_only|acquisition_and_trgout|disabled

Il parametro TRIGGER_ENABLE_MASK consente di impostare quali canali generano un segnale di trigger nel caso in cui il loro input superi la TRIGGER_THRESHOLD impostata.

Esempio (nel caso in cui il digitizer abbia 4 canali): SELF_TRIGGER_ENABLE_MASK 0x9 acquisition_only. In questo modo i canali che possono generare il trigger sono lo 0 e il 3 (0x9 = 1001 in base 2).

4.2 Impostazioni per singolo canale o gruppo

CH channelnumber TRIGGER_THRESHOLD triggerthreshold

Esempio: CH 2 TRIGGER_THRESHOLD 0x100

Imposta a 0x100 la soglia di auto triggering del canale 2.

GR groupnumber TRIGGER_THRESHOLD triggerthreshold

Esempio: GR 2 TRIGGER_THRESHOLD 0x100

Imposta a 0x100 la soglia di auto triggering del gruppo 2.

FAST fastnumber TRIGGER_THRESHOLD triggerthreshold

Esempio: FAST 1 TRIGGER_THRESHOLD 0x100

Imposta a 0x100 la soglia di auto triggering del canale di fast triggering 1. L'impostazione ha senso solo per i digitizers della famiglia x742.

CH channelnumber DC_OFFSET dcoffset

Esempio: CH 2 DC_OFFSET 0x3fff

Imposta a 0x3fff il dc offset del canale 2.

GR groupnumber DC_OFFSET dcoffset

Esempio: GR 2 DC_OFFSET 0x3fff

Imposta a 0x3fff il dc offset del gruppo 2.

FAST groupnumber DC_OFFSET dcoffset

Esempio: FAST 1 DC_OFFSET 0x3fff

Imposta a 0x3fff il dc offset del canale di fast triggering 1. L'impostazione ha senso solo per i digitizers della famiglia x742.

5. Note per la compilazione

Il Makefile mette a disposizione i seguenti target:

all: produce gli eseguibili “objectdump”, “objectdumpclient” e “decode” (vedi sezione installazione). Il codice oggetto prodotto dalla compilazione è posto nella cartella objectcode.

remove: rimuove gli eseguibili “objectdump”, “objectdumpclient”, “decode” e il contenuto della cartella objectcode.

flex: partendo dal file AnalizzatoreLessicale.flex, produce il file Analizzatore.c.

N.B. Se si modifica il file AnalizzatoreLessicale.flex, occorre eseguire “make flex” per produrre un nuovo file Analizzatore.c e rendere quindi effettive le modifiche alla successiva compilazione (eseguibile semplicemente con il comando “make”).