IRIG Corruptor Script Parser

Shockely Project

Table of Contents

General

IRIG Corruptor accepts scripts in the form of a text file. The script file is split on newline (\n) where each line can be blank, a comment, or a single IRIG Corruptor command.

Comments

A comment is considered to be any line where the first non-whitespace character(s) are either '#' or '//'. Everything following will be treated as a comment and ignored by the command parser. Comments must be on their own line, they suffix a command.

Valid Comment Example

```
# This is a valid comment using hash tag.
// This is a valid comment using backslashes.
```

Invalid Comment Example

```
WAIT 1000 # Invalid comment
PPS_JIT 500 // Another invalid comment
```

Previous Example Corrected

```
# Valid Comment
WAIT 1000
// Another valid comment
PPS_JIT 500
```

Commands

Commands follow a Linux shell style where a line is tokenized on whitespace. The first token is considered the command with all following tokens ton be considered parameters to that command. Commands are case insensitive.

WAIT

Description: Halts script execution until the specified number of milliseconds has passed.
Parameters: The WAIT command takes one argument, the time in milliseconds for which to wait.
Usage: WAIT [0-9]+
Example:

```
# wait for 1 second
WAIT 1000
```

OUT_OFF

Description: Turn off IRIG output.
Parameters: None

OUT_ON
Description: Turn on IRIG output.
Parameters: None

RESET
Description: Reset's the IRIG corrupt to initial values.
Parameters: None

TIME
Description: Sets the time of the IRIG output stream.
Parameters: The TIME command accepts a single argument, a time value in 24 hours format hh:mm:ss.
Usage: `TIME hh:mm:ss`
Example:

```
# Set time to 1:30pm
TIME 13:30:00
# Set time to 1:30am
TIME 1:30:00
```

DATE
Description: Sets the date of the IRIG output stream.
Parameters: The TIME command accepts a single argument, a date value in mm/dd/yyyy or mm/dd/yy format.
Usage: `DATE [1-12]/[1-31]/([0-9]{4}|[0-9]{2})`
Example:

```
# Set date to March 26th, 2013
DATE 3/26/13
# Another way to achieve the same result
DATE 03/26/2013
```

UTC
Description: Sets the UTC offset for IRIG Corruptor date handling.
Parameters: The UTC command accepts a single argument, the UTC offset in the format (+-)nn.m where n is an integer in the range [0,9] and m is value from the set {0, 25, 5, 75}.
Usage: `UTC [+-][0-9]{2}.(0|25|5|75)`
Example:

```
# Set time zone to pacific
UTC -07.0
# Another way to achieve the same result
UTC -7.0
```

TQUAL

Description: Sets the time quality of the IRIG Corruptor

Parameters: TQUAL accepts a single integer parameter n in the range [0,15]. Each value represents a quality value.

- 0 – 0ms
- 1 – 1ns
- 2 – 10ns
- 3 – 100ns
- 4 – 1us
- 5 – 10us
- 6 – 100us
- 7 – 1ms
- 8 – 10ms
- 9 – 100ms
- 10 – 1s
- 11 – 10s
- 15 – FAULT

Usage: `TQUAL [0-15]`

Example:

```
# Set quality bit to 100us
TQUAL 6
```

DSTP

Description: Sets the DSTP bit.

Parameters: DSTP accepts a single integer value of 0 or 1 representing off or on respectively.

Usage: `DSTP [0-1]`

Example:

```
# Set DSTP bit
DSTP 1
```

DST

Description: Sets the DST bit.

Parameters: DST accepts a single integer value of 0 or 1 representing off or on respectively.

Usage: `DSTP [0-1]`

Example:

```
# Set DST bit
DST 1
```

LS

Description: Sets the LS (leap second) bit.

Parameters: LS accepts a single integer value of 0 or 1 representing off or on respectively.

Usage: `LS [0-1]`

Example:

```
# Unset LS bit
LS 0
```

LSP

Description: Sets the LSP (leap second pending) bit.

Parameters: LSP accepts a single integer value of 0 or 1 representing off or on respectively.

Usage: `LSP [0-1]`

Example:

```
# Set LSP bit
LSP 1
```

PPS_JIT

Description: Sets jitter on pps pulses.

Parameter: PPS_JIT accepts a single integer value in the range (1ns,10ms) in nanoseconds.

Usage: `PPS_JIT [0-10000000]`

Example:

```
# Set PPS_JIT to 500ns
PPS_JIT 500
```

10MS_JIT

Description: Sets jitter on 10ms pulses.

Parameter: 10MS_JIT accepts a single integer value in the range (1ns,10ms) in nanoseconds.

Usage: `10MS_JIT [0-10000000]`

Example:

```
# Set 10MS_JIT to 1000ns
10MS_JIT 1000
```

JITTER

Description: Enables or disables output jitter.

Parameters: Jitter accepts a single integer value or 0 or 1 representing off or on respectively.

Usage: `JITTER [0-1]`

Example:

```
# Turn output jitter off
JITTER 0
```

FORCE_TRIG

Description: Sets the trigger that will begin the force output sequence.

Parameters: FORCE_TRIG accepts a single integer parameter in the range [0,2]. Each value represents a different trigger.

- 0 – Trigger on FORCE_START
- 1 – Trigger on next PPS after FORCE_START command
- 2 – Trigger on next 10ms rising edge after FORCE_START command

Usage: `FORCE_TRIG [0-2]`

Example:

```
# Set trigger to start on 10ms rising edge
FORCE_TRIG 2
```

FORCE_MOD

Description: Sets the modification that will be performed by the force output sequence.

Parameters: FORCE_MOD accepts a single integer parameter in the range [0,3]. Each value represents a different output modification to be performed.

- 0 – No Modification.
- 1 – Force IRIG output to 1
- 2 – Force IRIG output to 0
- 3 – Force IRIG output to the inverse of the input IRIG

Usage: `FORCE_MOD [0-3]`

Example:

```
# Force IRIG output to 1
FORCE_MOD 1
```

FORCE_DELAY

Description: Sets the amount of time to wait after the trigger before modifying the output IRIG signal.

Parameters: FORCE_DELAY accepts a single integer value in the range [0ns-0.214s] in nanoseconds.

Usage: `FORCE_DELAY [0-214000000]`

Example:

```
# Set delay to 1ms
FORCE_DELAY 1000000
```

FORCE_DUR

Description: Sets the amount of time to spend modifying the output IRIG signal.

Parameters: FORCE_DUR accepts a single integer value in the range [0ns-0.214s] in nanoseconds.

Usage: `FORCE_DUR [0-214000000]`

Example:

```
# Set duration to 10ms
FORCE_DUR 10000000
```

FORCE_START

Description: Begin waiting for trigger event.

Parameters: None

FORCE_STOP

Description: Stop everything and return to passing input IRIG to the output unmodified.

Parameters: None

Example Scripts

If possible I would like to put maybe 2 or tree short example scripts that actually achieve something useful.