

# THAMWAY PROT series hardware server software

## TCP/IP port to be used

TCP/IP PORT:5025 It uses to communicate with the PG32 PULSER.

## Delimiter of command

Please take always delimiter character in the command using the TCP/IP. Delimiter, 'CR', 'CR + LF', ';' is one of the three out of.

## About numerical value to be used

### Integer

If you put a "0x" at the beginning it is regarded as hexadecimal.

If you put a "0" at the beginning it is regarded as octal.

If you put a "0b" at the beginning it is regarded as binary.

Other than the above it will consider the decimal.

Example: 0x123 → 291

0b1100 → 12

0123 → 83

### Real

If you put a "u" in the back of the numeric value is regarded as 1e-6(micro).

If you put a "m" in the back of the numeric value is regarded as 1e-3(milli).

If you put a "k" in the back of the numeric value is regarded as 1e+3(kilo).

例: 123k → 123000

1.2u → 0.0000012

### QPSK

At the same time I will output a QPSK pulse and transmitted pulse. There are two 'QPSK1' and 'QPSK2'.

### TIME

Unit of time is sec.

### FREQUENCY

Unit of frequency is Hertz(Hz).

### VOLTAGE

Unit of voltage is volt.

## About 'Lua' command

This software can also be operated using an internal Lua commands without using the TCP/IP connection. Operate and use the keyboard from the command line.

TCP / IP PORT: 5025 command, you can substitute call **PG**( "*pulsercommand*" ).

TCP / IP PORT: 5026 command, you can substitute call **AD**( "*AD command*" ).

TCP / IP PORT: 5027 command, you can substitute call **RF**( "*AD command*" ).

### Example

call **PG**( "start 10" )

call **AD**( "startad 256,1,1,0" )

call **RF**( "RFSW1" )

call **RF**( "RFSW0" )

# PULSER Command List

The use the TCP / IP PORT: 5025 or callPG ()

## Common commands (standard, extended mode Common)

### View device information

---

[Format]

**\*idn?**

[Description]

Make the display of the hardware and software of information connected.

[Result]

THAMWAY, N210-1026T PULSER, Version 2.00, PG32U40 , 071024, CLK=40MHZ, BIT=32, RAM=262144,

### Set the mode

---

[Format]

**setmode** <value>

[Description]

value 0:Standard mode

1:Extended mode

[Example]

setmode 1

### Start the pulse sequence

---

[Format]

**start** <Repeat count>

[Description]

Execute the repetition number of times the pulse sequence.

Repeat count 0:Keep repeating

1..65535:Specified number of times repeatedly.

### Stop the pulse sequence

---

[Format]

**stop**

[Description]

Stop the pulse sequence

## Run the Lua script file.

---

### [Format]

**run\_lua** <file name>

### [Description]

Run a lua script file on your PC.

### [Example]

run\_lua c:/myprograms/test1.lua

## Write data to the I/O port

---

### [Format]

**outb** <I/Oaddress>, <byte data(8bit)>

**outw** <I/Oaddress>, <word data(16bit)>

### [Description]

It is used when operating the register of each device directly.

## Read data from the I/O port

---

### [Format]

**inb** <I/O address>

**inw** <I/O address>

### [Description]

It is used when operating the register of each device directly.

inb: read 8bit data.

inw: read 16bit data.

## Display PULSER memory

---

### [Format]

**dispmem** <start address>, <length>

### [Description]

It will make the display of the current PULSER memory.

start address: Please specify the memory address. Start address is 0.

length: Specifying the number of memory to be displayed.

## STANDARD MODE COMMAND

Specifies the width of the first pulse

[Format]

**fpw** <length>

[Description]

Set the width of the 1'st PULSE.

Specifies the QPSK of the first pulse

[Format]

**fpq** <qpskvalue>

[Description]

Set the QPSK of the 1'st PULSE.

Refer to QPSK VALUE (Table1)

Set SINGLE PULSE MODE

[Format]

**single**

[Description]

Set to SINGLE PULSE MODE

Set DOUBLE PULSE MODE

[Format]

**double**

[Description]

Set to DOUBLE PULSE MODE

Use COMB PULSE

[Format]

**usecomb** <value>

[Description]

value 0: not use

1: use

Set the length of the COMB PULSE

[Format]

**cpw** <length>

[Description]

Set the length of the COMB PULSE

### Specify the interval of COMB PULSE and COMB PULSE

---

[Format]

**cp**i      <length>

[Description]

Specify the interval of COMB PULSE and COMB PULSE.

### Specify the number of COMB PULSES.

---

[Format]

**cp**n      <number>

[Description]

Specify the number of COMB PULSES.

### Specifies the QPSK of the COMB PULSE.

---

[Format]

**cp**q      <qpskvalue>

[Description]

Specifies the QPSK of the COMB PULSE.

Refer QPSK VALUE(Table1)

### Specify the interval of up to 1st pulse from COMB PULSE.

---

[Format]

**t**j <length>

[Description]

Specify the interval of up to 1st pulse from COMB PULSE.

### Specifies the width of the second pulse.

---

[Format]

**sp**w <length>

[Description]

Specifies the width of the second pulse.

### Specifies the QPSK of the second pulse.

---

[Format]

**sp**q <qpskvalue>

[Description]

Set the QPSK of second pulse.

Refer to QPSK VALUE(Table1)

## Specifies interval until the first pulse and second pulse

---

[Format]

**t2** <time>

[Description]

Specify the interval of 1, st PULSE and 2' nd PULSE.

## Specifies A/D trigger offset

---

[Format]

**adoff** <time>

[Description]

You set the relative time of the A / D converter for the trigger to be output from the AD. TRG terminal. Pulse width is 1 $\mu$ S fixed. You can set the value of both positive and negative. Please set so as not to be outside the range.

[Example]

adoff -1.3uS

adoff 2uS

## Setting of A/D trigger

---

[Format]

**adtrg** <value>

[Description]

Make the pulse setting of the output from the AD. TRG terminal.

Refer to figure1

value            0:Spin echo position  
                  1:FreeInductionDecay position

## Setting of TRG. OUT terminal

---

[Format]

**trgout** <value>

[Description]

You set the position of the output pulses from TRG. OUT.

The length of the pulse is 1us.

**Table 1: TRG.OUT output**

<b>value</b>	<b>position</b>
0	Start of COMB PULSE
1	End of COMB PULSE
2	1'st pulse rise edge
3	2'nd pulse raise edge
4	
5	Same as A/D trigger pulse.



## Setting of external trigger

---

[Format]

**exttrg** <value>

[Description]

set use of 'PG. TRG. IN'.

value 0: not use 'PG. TRG. IN'.

1: use 'PG. TRG. IN'.

## Setting the blanking time

---

[Format]

**blank** <time>

[Description]

The unit is seconds. Please set a positive value.

## Setting of the kind of the blanking time

---

[Format]

**waitmode**

**loopmode**

[Description]

**waitmode**: The software uses 'blank' as time for interval.

**loopmode**: The software uses 'blank' as time for repetition.

Please usually use it in waitmode. When I measure the sample which is sensitive to temperature such as the cryogenic temperature, I use loopmode.

The follows are commands to coordinate to the characteristic of the transmitter.

#### Setting of the quantity of transmission gate delay

---

[Format]

**tx2predelay**      <time>

**tx2postdelay**    <time>

[Description]

tx2predelay: This appoints the width for the front part of TX1. The initial value is 10uS.

tx2postdelay: This appoints the width for the rear of TX1. The initial value is 0.

**Attention: It becomes invalid with the extended mode.**

#### QPSK ディレイ量設定

---

[Format]

**qpskdelay** <time>

[Description]

This sets quantity of delay for QPSK pulse and the TX pulse. And I depend on the QPSK modulator.

The initial value is 1.5uSec.

**Attention: It becomes invalid with the extended mode.**

## EXTENDED MODE

The following commands are the commands that are effective in an expansion mode. Because it is a fixed form mode, at the time of the software start, the expansion mode, please using 'setmode'.

### Clear memory

---

[Format]

**memclr**

[Description]

This command initializes the memory for the pulse making. Please carry it out by all means before making a pulse.

### Append pulse

---

[Format]

**makepulse** <BitNumber>, <start time>, <pulse width>

[Description]

This command insert in a 1-bit wide pulse at a storage device.

BitNumber: It is a bit position. The range is 0-31. (Refer to Table4)

start time: This is time from the top. It is absolute value. Please input a positive value.

Pulse width: This is pulse output time. Please input a positive value.

### Erase pulse

---

[Format]

**erasepulse** <BitNumber>, <start time>, <pulse width>

[Description]

This removes a 1-bit wide pulse.

BitNumber: It is a bit position. The range is 0-31. (Refer to Table4)

start time: This is time from the top. It is absolute value. Please input a positive value.

Pulse width This is width of the pulses to remove. Please input a positive value.

## Write board control command

### [Format]

**makecommand** <write position>, <command>, <data>

### [Description]

**write position:** This appoints the note position of the command in time from the top.  
(positive value)

**command:** appoint a jump command and a stop command during a sequence.

**data:** Appoint necessary numerical value. In the case of the command that numerical value is unnecessary, please appoint 0.

### [Example]

This example writes in jump at the position of 1.35 seconds from the top.

makecommand 1.35, 0xff200000, 0

Table 2: Board control command

Name	Command	Data	Meaning	
GOTO	0xFF200000+<memory address>	Unnecessary (0)	jump to memory address. use for looping.	
STOP	0xFF400000	Unnecessary (0)	Stop a pulse sequence.	
TRG	0xFF800000	Unnecessary (0)	Wait for trigger.	

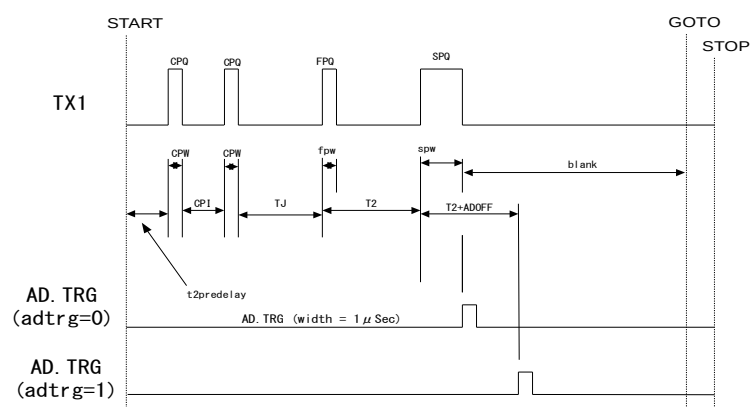
## Example of Lua script in STANDARD MODE

The following text perform the pulse output in Lua.

```
function wait_pulser()  
    repeat until "STOP" == callPG("ISRUN?")  
end  
  
print("START PULSER")  
callPG("stop")  
callPG("setmode 0")  
callPG("loopmode")  
callPG("double")  
callPG("t2 35u")  
callPG("blank 1m")  
callPG("fpw 10u")  
callPG("spw 20u")  
callPG("start 1000")  
wait_pulser()  
print("END")
```

Text 1: Example of Lua script in STANDARD MODE

figure 1: STANDARD MODE pulse output



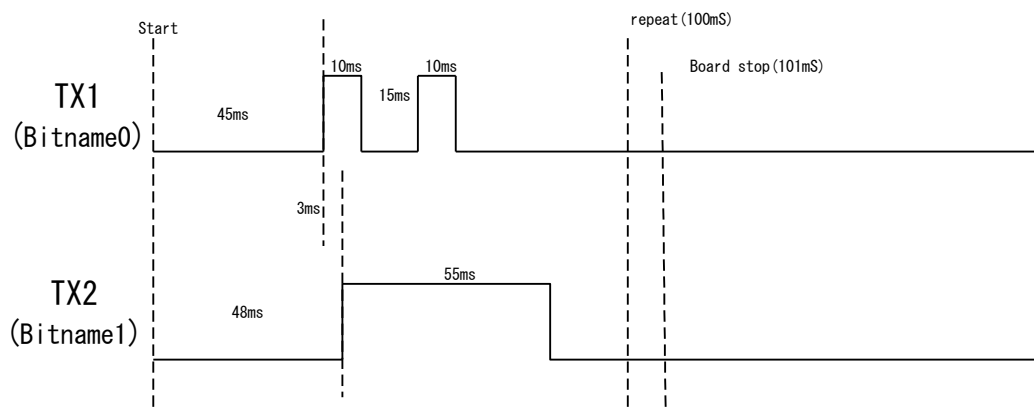
## Example of Lua script in EXTENDED MODE

The following text perform the pulse output in Lua.

```
function wait_pulser()  
    repeat until "STOP" == callPG("ISRUN?")  
end  
  
print("START PULSER EXTEND MODE")  
callPG("stop")  
callPG("setmode 1")  
callPG("memclr")  
callPG("makepulse 0, 45m, 10m")  
callPG("makepulse 1, 48m, 55m")  
callPG("makepulse 0, 70m, 10m")  
callPG("start 1000")  
wait_pulser()  
print("END")
```

**Text 2: Example of Lua script in EXTENDED MODE**

figure 2: The pulse output when I executed the script above



**Table 3: QPSK VALUE**

qpskvalue	TX phase	QPSK2 output level	QPSK1 output level
0	+X (0 degree)	LOW	LOW
1	+Y (90 degree)	LOW	HIGH
2	-X (180 degree)	HIGH	LOW
3	-Y (270 degree)	HIGH	HIGH

**Table 4: PULSER BIT NUMBER**

BitNumber	Name	Remarks
0	TX1	TX gate signal
1	TX2	TX biasgate signal
2	AUX1	RX gate signal
3	QPSK1	QPSK LSB
4	QPSK2	QPSK MSB
5	AUX2	
6	TRG. OUT	Trigger for Oscilloscope
7	METER	Trigger for PowerMeter
8	METER. RESET/AUX9	Reset for PowerMeter
9	AD. TRG	Trigger for A/D
10	AUX3	
11	COMB/DA. TRG	COMB PULSE
12	1' st PULSE/AUX6	1' st PULSE
13	2' nd PULSE/AUX7	2' nd PULSE
14	AUX4	
15	AUX10	

**Table 5: I/O PORT (low address)**

address map ( low-address 8bit )	port name
0x00-0x1f	PULSER BOARD
0x20-0x3f	AD BOARD
0x40-0x5f	reserved
0x60-0x7f	reserved
0x80-0x9f	----
0xa0-0xbf	RF LOW LEVEL
0xc0-0xcf	----
0xe0-0xff	----

**Table 6: I/O port (high address)**

RANGE	DIP SWITCH			
	3	2	2	1
0x0000-0x00ff	ON	ON	ON	ON
0x0100-0x01ff	ON	ON	ON	OFF
0x0200-0x02ff	ON	ON	OFF	ON
0x0300-0x03ff	ON	ON	OFF	OFF
0x0400-0x04ff	ON	OFF	ON	ON
0x0500-0x05ff	ON	OFF	ON	OFF
0x0600-0x06ff	ON	OFF	OFF	ON
0x0700-0x07ff	ON	OFF	OFF	OFF
0x0800-0x08ff	OFF	ON	ON	ON
0x0900-0x09ff	OFF	ON	ON	OFF
0x0a00-0x0aff	OFF	ON	OFF	ON
0x0b00-0x0bff	OFF	ON	OFF	OFF
0x0c00-0x0cff	OFF	OFF	ON	ON
0x0d00-0x0dff	OFF	OFF	ON	OFF
0x0e00-0x0eff	OFF	OFF	OFF	ON
0x0f00-0x0fff	OFF	OFF	OFF	OFF