

PAD1616

**100 kHz 16 channel 16 bit transient recorder
300 kHz 16 channel 16 bit transient recorder
for ISA bus**

**Hardware Manual
Driver Manual**

PAD1616a/b 100/300 kHz 16 channel - 16 bit transient recorder

- * 16 channels
- * resolution 16 bit
- * accuracy 15 bit
- * posttrigger up to 2^{21} Samples
- * sampling rate 300 kHz/100 kHz -1 kHz
- * 2 versions: 300 kHz/100 kHz
- * memory depth up to 8 MSamples.
- * number of channels and gain individually programmable
- * triggerlevel- and edges programmable
- * input range from ± 500 mV to ± 5 V
- * external TTL-trigger input/output
- * interrupt after sequence stopped
- * statusbits for polling applications

General information:

The PAD1616 is a 16 channel A/D converter board with a resolution of 16 bits. 16 channels are connected to the ADC by using a multiplexer. The sampling sequence can be programmed in a 1024 entry deep ring memory. Sampling and gain of each channel is individually programmable. It is not necessary to use a fixed order when the channels are selected. In this memory the trigger channel is also selected. The PAD1616 samples the inputs (within a sequence) with its maximum conversion rate even if a lower sampling rate is programmed. This minimizes the phase error when more than one channel should be sampled. It is possible to read out the sampled values while the board is running.

The signals TRIG-IN, TRIG-OUT, VCC, GND, +5 V, GNDA and Ch0 - Ch15 are connected to a 50 pole connector. The PAD1616 uses a 16 bit ISA slot.

Software programmable parameters

sampling rate (100 kHz):	0.6 Hz to 100 kHz
sampling rate (300 kHz):	2.3 Hz to 300 kHz
input range:	± 500 mV, ± 1 V, ± 2 V, ± 5 V
channel:	n of 16
sequence memory:	up to 1024 parameter sets
trigger:	\pm edge, extern or software
memory depth:	1 Sample up to 2 MSamples, 8 MSamples
posttrigger:	2 to 2^{21} Samples

Software

Driver for DOS, Windows 3.x, Windows 95/98 and Windows NT are delivered with the board. Comfortable programming, initializing and data display is performed by the Windows program SBench. Software drivers for National Instrument's 'LabVIEW for Windows', for Hewlett Packard's 'HP VEE' and for The Mathworks Inc. 'MATLAB' are available.

Options

sampling rate:	300 kHz, 100 kHz
memory:	2 MSample, 8 MSample

Programming the PAD1616

Programming and initialising of the PAD1616 is carried out by using eight 16 bit wide register. (Addresses are relatively to the board address.)

0x00 memory low	- write	0x00 read data	- read
0x02 memory high	- write	0x02 read status	- read
0x04 posttrigger low	- write		
0x06 posttrigger high	- write		
0x08 channel	- write		
0x0A trigger & command	- write		
0x0C sequence low	- write		
0x0E sequence high	- write		

Memory depth address: 0x00, 0x02

The required memory depth is programmed in two 16 bit words. The register at address 0x00 holds the lower value and 0x02 holds the upper value of the memory depth.

Minimum value is 3, resulting in 4 samples, maximum is 0x1FFFFFF (2 MSamples), 0x3FFFFFF (4 MSamples) or 0x7FFFFFF (8 MSamples).

$$\text{memory depth} = \text{programmed value} + 1$$

Example:

1.000.000 Samples => F4240 hex
 F4240 - 1 = F423F

=> 423F to 0x00

=> 000F to 0x02

Bit d15 of address 0x02 is the NoPreTrig bit. If this bit is set to 1 the internal state machine enables the trigger machine immediately after start. No pretrigger is available in this case. This bit must not be set if posttrigger is equal or less to memsize.

Memory depth should be the result of: Number of channels * mem.depth/channel

Programming in C :

```
output (boardaddress+0x00,0x423F);
output (boardaddress+0x02,0x000F);
```

Programming in C (no pretrigger)

```
output (boardaddress+0x00,0x423F);
output (boardaddress+0x02,0x000F | 0x8000);
```

Posttrigger address: 0x04, 0x06

D0 - D15 binary coded value of posttrigger

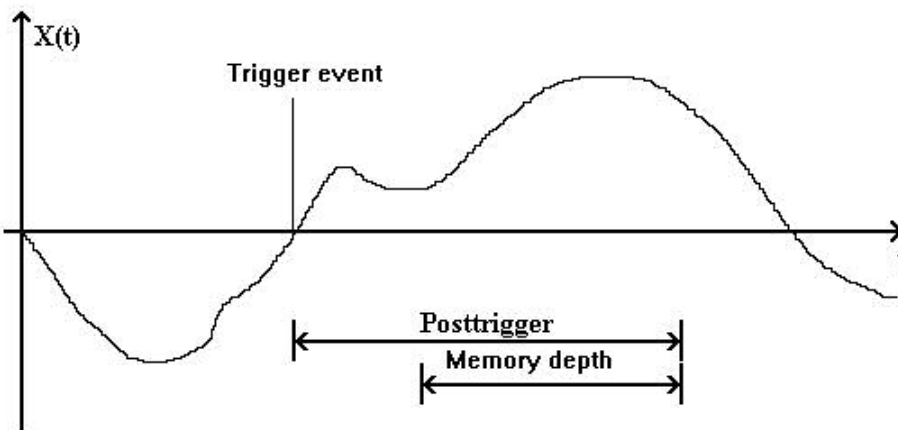
The posttrigger defines the number of samples stored **after** a trigger event is to be found.

The programmed value is calculated by the formula:

$$\text{value} = (\text{number of channels}) * (\text{required posttrigger for each channel} - 1) + 1$$

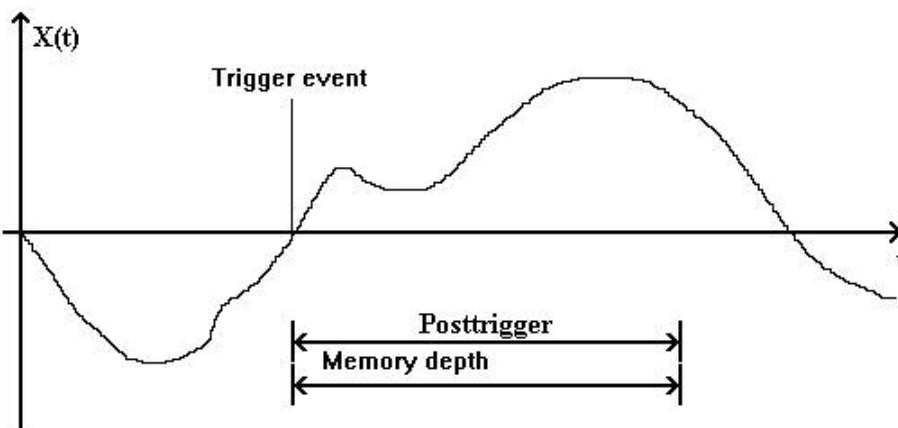
Minimum is 0x02!

Example: memory depth < posttrigger



This figure shows the situation when the posttrigger is deeper than the programmed memory. Only data after the trigger event are sampled. The trigger event is not visible.

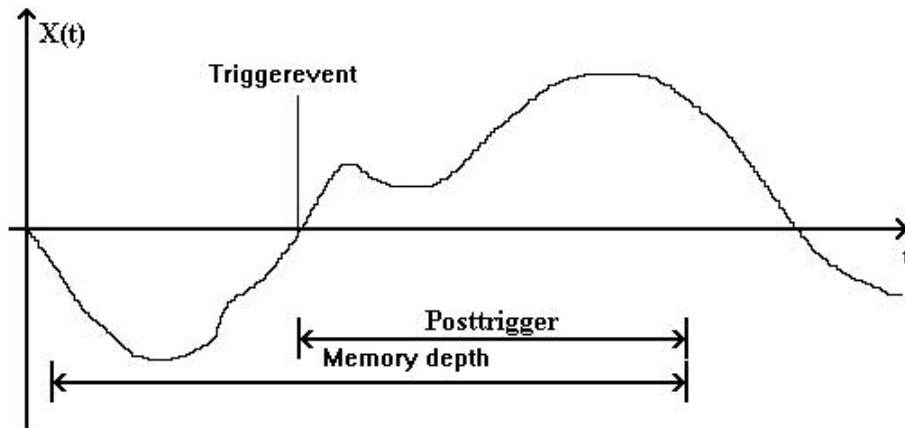
Example: memory depth = posttrigger



All signals are sampled starting at the trigger event.

Example: memory depth > posttrigger

This figure shows a typical pretrigger configuration.



Example:
3 channels used, with 7000(dec.) => 5208(hex) samples posttrigger.

prog. value = ((7000-1) * 3) + 1 = 20998 (dez) => 5206 (hex)

Programming in C:

```
outport(board address+0x04,0x5206);
outport(board address+0x06,0x0000);
```

Channel and gain sequence address: 0x08

This register holds up to 1024 channel parameters.

This register must be completely programmed every time the PAD1616 is restarted.

It is 9 bits wide and holds the following information:

D2	D1	D0	input range
0	0	0	±10 V (not used)
0	0	1	±5 V
0	1	0	±2 V
0	1	1	±1 V
1	0	0	±500 mV

Channel selection

D6	D5	D4	D3	channel
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

D7 = 1 = trigger channel selects this channel to be the trigger.

This bit has to be set only once in the whole sequence.

D8 = 1 = sequence-bit. This bit must be set in the first sequencer word.

Example:

Ch4 and ch1 should be sampled. Ch4 uses the input range 5 V and ch1 the input range 1 V. Ch1 is the trigger channel.

```
ch 4 :                0x0020
sequence start :     OR   0x0100
input range :        OR   0x0001
                    =>   0x0121
```

```
ch 1:                0x0008
triggerchannel:     OR   0x0080
input range :        OR   0x0003
                    =>   0x008B
```

programming in C:

```
output (board address+0x0A,0x1000); /* FIFO-Reset */
output (board address+0x08,0x0121);
output (board address+0x08,0x008B);
```

Command address: 0x0A

This register holds the trigger and starts parameter.

D0-D6 Triggerlevel, compared to the 7 MSBs of the ADC (2th complement).

D7 Rising/falling edge of the trigger edge. (rising edge \leftrightarrow HIGH).

D8,D9	Triggermode	D9	D8
	softwaretrigger	0	0
	signaltrigger	0	1
	external TTL-trigger	1	0
	signaltrigger	1	1

D11 n.u.

D12 FIFO-reset reset = 1

D13 n.u.

D14 n.u.

D15 board START start = 1

Triggerlevel

The triggerlevel is a 7 bit 2th complement value. It depends on the input range.

(not used)

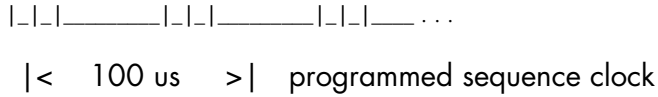
$\pm 500 \text{ mV}$	$\pm 1 \text{ V}$	$\pm 2 \text{ V}$	$\pm 5 \text{ V}$	$\pm 10 \text{ V}$	value
-0,492	-0,984	-1,969	-4,922	-9,844	0x41
-0,484	-0,969	-1,938	-4,844	-9,688	0x42
					..
-0,0078	-0,0156	-0,031	-0,078	-0,156	0x7F
0,0	0,0	0,0	0,0	0,0	0x00
0,0078	0,0156	0,031	0,078	0,156	0x01
					..
0,477	0,953	1,906	4,766	9,533	0x3C
0,484	0,969	1,938	4,844	9,688	0x3E
0,492	0,984	1,969	4,922	9,844	0x3F

Resolution is: $2 * \text{input range} / 128$

Sequence clock address: 0x0C, 0x0E

This value defines the sequence repetition rate. It is equal to the sampling rate for each channel. The PAD1616 samples the selected input channels with its maximum speed even when a lower frequency is programmed.

Example: 3 channels selected with 10 kHz sampling each. Using the PAD1616a with 100 kHz.



This minimizes phase errors when sampling more than one input.

The value to be written to the register is calculated by:

- PAD1616a (100 kHz) value = (100 kHz/(intended sampling rate)) - 1
- PAD1616b (300 kHz) value = (300 kHz/(intended sampling rate)) - 1

Maximum sampling frequency for multiple channels:

$$\text{PAD1616x} \quad \text{max.F} = (100 \text{ kHz}, 300 \text{ kHz})/n \quad (n = \text{number of used channels})$$

Example: PAD1616a (100 kHz), intended sampling ck = 25 kHz

$$\text{value} = (100000/25000) - 1 = 3$$

Reading data address: 0x00

Data are read by selecting the boards address 0x00 with an I/O-read strobe.

$$\text{in} = \text{inport}(\text{boardaddress})$$

When the PAD1616 has stopped (D0 in the statusregister = 0), the read data comes into the following order:

1st read gives the oldest data for the input channel that is programmed first in the sequencer memory,

2nd read the oldest data for the 2nd programmed channel,

. . . and so on until all programmed channels are read.

The next read cycle gives the next data for the 1st programmed channel . . .

Example: channel 0, 1, 7 are programmed. After the board has been stopped one read:

$\text{ch0}(t_n), \text{ch1}(t_n), \text{ch7}(t_n), \text{ch0}(t_{n+1}), \text{ch1}(t_{n+1}), \text{ch7}(t_{n+1}) \dots \text{ch0}(t_0), \text{ch1}(t_0), \text{ch7}(t_0)$

(t_n = most past sampling time)

When the PAD1616 is active a read-access to the address 0x00 results in getting the value of the actually sampled channel.

Status address: 0x02

The board has 2 status bits giving the user information about the state of the PAD1616.

D0 = 1 PAD1616 active

D1 = 1 trigger found, posttrigger active

All other bits are undefined.

Addresses

The address is selectable in increments of 16 in the 512 Byte I/O addressspace. Selection is done by setting the jumpers J1 - J8. The

J1	corresponds with address signal	A4
J2	"	A5
J3	"	A6
J4	"	A7
J5	"	A8
J6	"	A9
J7	"	A14 (option, jumper must be fitted)
J8	"	A15 (option, jumper must be fitted)

A fitted jumper selects the address signal to be "0".

J6	J5	J4	J3	J2	J1	address
0	0	0	0	0	0	0x0
0	0	0	0	0	1	0x10
0	0	0	0	1	0	0x20
.1	1	0	1	0	0	0x340
1	1	0	1	1	0	0x360
.
1	1	1	1	1	1	0x3F0

The PAD1616 is delivered with the address 0x340 jumpered.

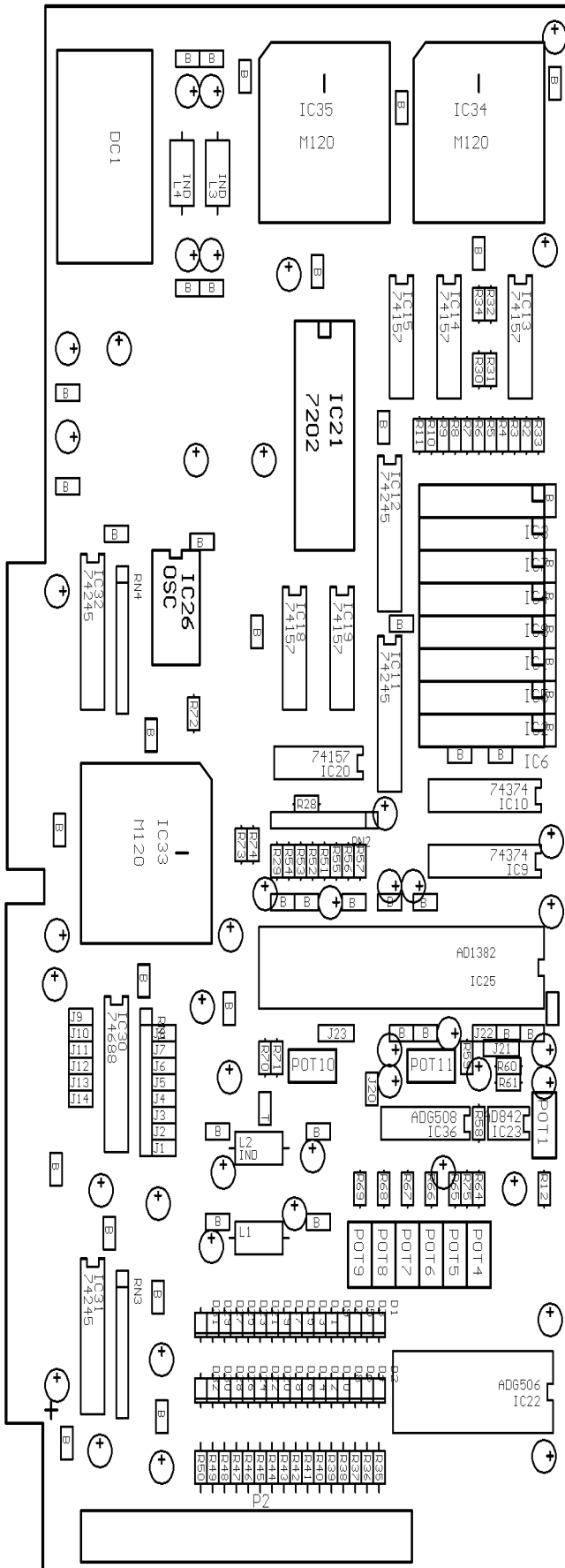
Interrupt

Selection of the interrupt channel is carried out by using the jumpers J9 - J14.

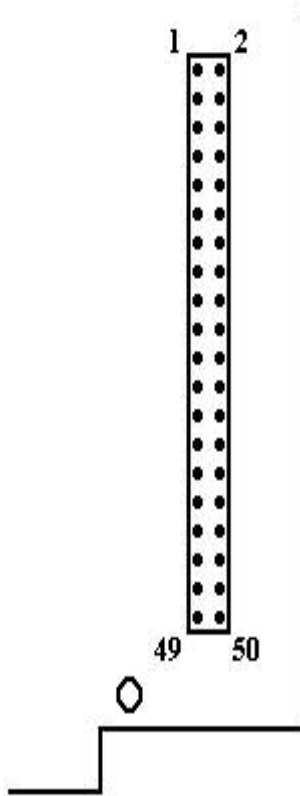
Only one jumper should be fitted!

jumper	interrupt channel
J9	IRQ3
J10	IRQ4
J11	IRQ5
J12	IRQ6
J13	IRQ7
J14	IRQ9

Placement



Connector



PIN	name
1	channel 00
3	channel 01
5	channel 02
7	channel 03
9	channel 04
11	channel 05
13	channel 06
15	channel 07
17	channel 08
19	channel 09
21	channel 10
23	channel 11
25	channel 12
27	channel 13
29	channel 14
31	channel 15
2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32	GND signal
43, 44, 45, 47	GND digital
49, 50	+5 V digital
46	trigger OUT
48	trigger IN (ext. trigger)

Technical data

resolution		16 bit
accuracy		15 bit
data format		2nd complement
channels		16
crosstalk	2 V-5 V	better -86 dB
crosstalk	500 mV-1 V	better -78 dB
memory depth		8 MSamples (16 MBytes). 4 MSamples (8 MBytes), 2 MSamples (4 MBytes).
sampling rate		100 kHz 300 kHz
posttrigger		21 bit
PC- bus		16 bit ISA
power supply		
	+5 V	1 A
	-5 V	0.2 A,
	+12 V	0.25 A
	-12 V	0.25 A
warm up time		min. 6 Minutes
maximum operating temp.		10° - 50°C
maximale store temp.		10° - +75°C
maximum humidity		0 - 80% (not condensing)
dimension		282 mm x 112 mm

order number	version
PAD1616a	100 kHz
PAD1616b	300 kHz

Software Description

Software Register

These software register are to be used for the functions *SpcSetParam* and *SpcGetParam* of the software driver. All constants are found in the header file REGS.H.

The PAD1616 must be initialised first with the function *SpcInitBoard*. The functions *SpcSetParam*, *SpcGetParam* and *SpcGetData* are implemented. The function *SpcGetData* will ignore the params *ch*, *start* and *len*. The function always returns the complete recorded data as described in the hardware part of the manual. Have a look at the programming examples for further information.

The registers SPC_ISAADR and SPC_INSTMEM must be set before writing or reading any of the other registers.

Registers

name	value (dec)	r/w	
SPC_COMMAND	0	w	Command register, allowed values listed below
SPC_STATUS	10	r	Status register, possible values listed below
SPC_ISAADR	1010	r/w	Installation address: 0x200 up to 0x3F0 in step of 0x10
SPC_INSTMEM	1020	r/w	Installed memory: 2M, 4M, 8M
SPC_MEMSIZE	10000	r/w	Memory size for recording : 4 samples up to installed memory. If posttrigger is larger than memsize the pretrigger area will automatically be disabled resulting in a large speed up of the recording.
SPC_POSTTRIGGER	10100	r/w	Posttrigger: 4 samples up to 2 Msamples.
SPC_SAMPLERATE	20000	r/w	Samplerate per channel for recording: "maximum samplerate" / "number of recorded channels" down to "maximum samplerate"/128k.
SPC_TRIGGERMODE	40000	r/w	Trigger mode for board: one of the below listed values
SPC_TRIGGERLEVEL	42000	r/w	Trigger level for channel trigger: value from -64 to 63
SPC_SEQUENCERESET	320000	w	Resets the programmed recording sequence
SPC_SEQUENCEADD	320010	w	Adds a sequence step for recording. A maximum of 1024 sequence steps could be used for the board. The parameters for this command are listed below. The sequence steps are performed with maximum conversion speed. The value "samplerate" defines the repetition rate of the sequence.

Values for command register

name	value (dec)	
SPC_START	10	Starts the board with the actual setup
SPC_STOP	20	Stops the board and resets the logic

Values for status register

name	value (dec)	
SPC_RUN	0	Hardware is running: the board is waiting for trigger or recording data.
SPC_READY	20	Recording has stopped, data are available

Values for trigger

name	value (dec)	
TM_SOFTWARE	0	trigger immediately with software
TM_CHXPOS	10000	trigger on marked channel with positive edge
TM_CHXNEG	10010	trigger on marked channel with negative edge
TM_TTLPOS	20000	external trigger TTL positive edge
TM_TTLNEG	20010	external trigger TTL negative edge

Values for sequence register

name	value (dec)	
SEQ_IR_10000MV	0	This sequence step uses ± 10 V input range.
SEQ_IR_5000MV	1	This sequence step uses ± 5 V input range.
SEQ_IR_2000MV	2	This sequence step uses ± 2 V input range.
SEQ_IR_1000MV	3	This sequence step uses ± 1 V input range.
SEQ_IR_500MV	4	This sequence step uses ± 500 mV input range.
SEQ_CH0	0	This sequence step records on channel 0.
SEQ_CH1	8	This sequence step records on channel 1.
SEQ_CH2	16	This sequence step records on channel 2.
SEQ_CH3	24	This sequence step records on channel 3.
SEQ_CH4	32	This sequence step records on channel 4.
SEQ_CH5	40	This sequence step records on channel 5.
SEQ_CH6	48	This sequence step records on channel 6.
SEQ_CH7	56	This sequence step records on channel 7.
SEQ_CH8	64	This sequence step records on channel 8.
SEQ_CH9	72	This sequence step records on channel 9.
SEQ_CH10	80	This sequence step records on channel 10.
SEQ_CH11	88	This sequence step records on channel 11.
SEQ_CH12	96	This sequence step records on channel 12.
SEQ_CH13	104	This sequence step records on channel 13.
SEQ_CH14	112	This sequence step records on channel 14.
SEQ_CH15	120	This sequence step records on channel 15.
SEQ_TRIGGER	128	This sequence step is the trigger source. Only one sequence step should be defined as trigger source.

All values must be combined with a logic OR. The SEQ_TRIGGER is only allowed to be set for one sequence step.

Examples:

Channel 0, ± 1 V, no trigger:	SEQ_CH0 or SEQ_IR_1000MV	3
Channel 8, ± 500 mV, trigger channel	SEQ_CH8 or SEQ_IR_500MV or SEQ_TRIGGER	196
Channel 15, ± 10 V, no trigger	SEQ_CH15 or SEQ_IR_10000MV	120