WAX Wireless Accelerometer

WAX Software

Software: waxrec

The waxrec command-line software. Windows is the only supported platform, but Linux and Mac may also work (unsupported).

The only required parameter is the port name (On Windows, use the syntax “\\.\COM*n*” to support serial ports with numbers greater than 10, or specify “!” to auto-discover a device). The optional command-line switches are as follows:

Parameter	**Notes**
-log	Log of data received to stdout (or and output file if specified).
-tee	Duplicate all log data to stderr.
-osc *host:port*	Transmit OSC data to the specified host IP address and port.
-timetag	Include a time-tag in OSC packets.
-stomphost *host:port*	Transmit data using the STOMP protocol to a message queue.
-init *string*	Send an initializing string to the connected device.

An example to log to the file log.csv and also echo the output to stderr:

```
    waxrec ! -log -tee > log.csv
```

See Wax Data / Log file format below for details on the format of the log file.

Software: WaxGUI

The WaxGUI software is Windows-only, and is used to provide a very basic, real-time view of the data received. Simply choose the port for the receiver and select “Connect and Log”. Any received accelerometer data will be displayed on a simple graph.

WAX Transmitter Commands

Connect the device to a computer and establish a serial connection. The connection can be verified using the command `AT`, a valid response will be `OK`. The following commands can then be used to configure the device operation. These settings can be used to configure the accelerometer behaviour and transmitter settings on the WAX device.

Sample Frequency and Range

This is the sample frequency the accelerometer is sampled at (not to be confused with the **transmit interval**). The transmit interval must also be tuned to ensure packets are transmitted often enough to ensure all the sampled data gets through.

Command	**Range/units**	**Default**	**Setting**
RATE=*code*	*(see table)*	RATE=25	50 Hz ±16g low-power

Other available rate codes are as follows:

	Power consumption and sensitivity/range							
	Lower power				Normal power			
Sample rate	**±2g**	**±4g**	**±8g**	**±16g**	**±2g**	**±4g**	**±8g**	**±16g**
12.5 Hz	215	151	87	23	199	135	71	7
25 Hz	216	152	88	24	200	136	72	8
50 Hz	217	153	89	25	201	137	73	9
100 Hz	218	154	90	26	202	138	74	10
200 Hz	219	155	91	27	203	139	75	11

'Normal' power modes have slightly less noise at the expense of consuming slightly more power.

Activity Threshold

The level of acceleration needed to wake up the device from sleep.

Command	**Range/units**	**Default**	**Setting**
ACTTHRES=*level*	0-254, 62.5mg	ACTTHRES=4	250mg threshold

Inactivity Threshold

The level of movement in units of 62.5 mg considered as no activity. If the device falls below this level for **active time**, the device will go into low-power sleep mode and stop transmitting until moved again. This value should be less than the activity threshold.

Command	**Range/units**	**Default**	**Setting**
INACTTHRES=*level*	0-254, 62.5mg	INACTTHRES=2	125mg threshold

Active Time

The time in seconds that sustained inactivity below the **inactivity threshold**, will cause the device to enter low-power sleep mode.

Command	**Range/units**	**Default**	**Setting**
ACTTIME=*time*	0-254, seconds	ACTTIME=4	4 second timeout

Channel

The digital channel used to transmit messages on (16 available). Must be matched with the receiver. The channel has a bandwidth of 192kbps, care must be taken to ensure the **transmit interval** and the **sample frequency** do not exceed this limit

Command	**Range/units**	**Default**	**Setting**
CHANNEL=*channel*	11-26, channel	CHANNEL=15	Channel 15

Transmit Interval

The interval in units of 4 µs between each packet transmission. If a higher **sample frequency** is used, this interval should be decreased.

Command	Range/units	Default	Setting
TRANINT=*time*	1-65534, 4 µs	TRANINT=31250	125 ms

Device ID

The ID adopted by the transmitter. This will be appended to the packets and used to identify the device with the receiver. Device IDs must be unique on the network. Device 0 cannot be assigned to a transmitter.

Command	Range/units	Default	Setting
DEVICE=*id*	1-254	DEVICE=*(varies)*	Unique number

Shipping

For shipping, the command SHIPPING may be issued which turns off the radio and the device enters a low-power state. The device must be connected once to exit shipping mode.

Advanced

For advanced users only:

- WATERMARK: The number of samples stored on the accelerometer before waking the processor (1-31 samples, default WATERMARK=10 samples).
- JITTERMASK: An additional randomized time delay to add to the transmission interval to reduce the chances of packet collision with other sensors on the network (0-65534, units of 4 µs, must be a value of 2^n-1, default JITTERMASK=8191 which is 33 ms).
- TARGET: The device to send the packets to on the current channel. Default TARGET=0.

WAX Receiver Commands

Connect the device to a computer and establish a serial connection. The connection can be verified using the `AT` command. A valid response will be `OK`. The following commands can then be used to configure the device operation:

Channel

The digital channel used to receive messages on (16 available). Must be matched with the transmitter.

Command	Range/units	Default	Setting
CHANNEL=*channel*	11-26, channel	CHANNEL=15	Channel 15

Shipping

For shipping, the command SHIPPING may be issued which turns off the radio and the device enters a low-power state. The device must be connected once to exit shipping mode.

Advanced

For advanced users only:

- DEVICE: The device identifier. Receivers should normally be DEVICE=0.
- TARGET: The device to send the packets to on the current channel. Default TARGET=0.

- MODE: The current receiver mode (0=none, 1=SLIP-encoded binary, 2=text, 3=SLIP-encoded OSC). Default MODE=1. Changes from the default mode are not recommended or supported (some data loss may occur).
- STARTMODE: The default mode used at start up (this does not change the current mode). STARTMODE=1 (SLIP).

WAX Data

Log file format

Log file format for *waxrec* and *WaxGui* software:

"ACCEL",*timestamp*,*deviceId*,*sequence*,*x*,*y*,*z*

The *timestamp* is in the format: *YYYY-MM-DD hh*:*mm*:*ss.fff.*

The *x*, *y*, and *z* values are in 1/256 g units.

Device stop/starts, duplicate transmitted packets and missing packets (as described above) can easily be spotted in the log files by observing non-sequential *sequence* values for a given device.

Raw serial line data (advanced users only)

Each packet is sent over the serial port in SLIP-encoding as a delimiter (defined in RFC 1055). Packets are delimited with an END character (where END=0xC0), any occurrence of END in the packet is replaced by ESC/ESC_END (where ESC=0xDB, ESC_END=0xDC), and any occurrence of ESC with ESC/ESC_ESC (where ESC_ESC=0xDD). The following code can be used to separate each packet (as modified from RFC 1055):

```
#define END 0xC0
#define ESC 0xDB
#define ESC_END 0xDC
#define ESC_ESC 0xDD

int recv_packet(char *p, size_t len)
{
    char c;
    int received = 0;
    for (;;)
    {
        c = recv_char();
        switch(c)
        {
        case END:
            if (received) return received;
            break;
        case ESC:
            c = recv_char();
            switch(c)
            {
                case ESC_END:
                    c = END;
                    break;
                case ESC_ESC:
                    c = ESC;
                    break;
            }
            // Fall-through to default
        default:
            if (received < len) p[received++] = c;
        }
    }
}
```

The packets have the following format:

Offset (bytes)	Type	Name	Description
0	uint8	reportType	0x12 (user)
1	uint8	reportId	0x78 ASCII ‘x’
2	uint16	deviceId	Device Identifier
4	uint8	status	Reserved (bit-0 battery warning).
5	uint16	sample	Reserved (analogue sample).
7	uint8	format	Accelerometer data format. Bits: rreeffff. *rr* = Range code: $\pm 2^{rr}$ (g) *ee* = Encoding format: 0=3 × 10-bit + 2-bit; 2=signed 16-bit. *ffff* = Frequency code: 32000 / ($2^{15\text{-}ffff}$)
8	uint16	sequenceId	Sequence number of first accelerometer reading in this packet (will wrap, or reset if device sleeps and resumes).
10	uint8	outstanding	Number of samples remaining on the device after this packet (can be used at the receiver to estimate the sampling timestamp)
11	uint8	sampleCount	Number of samples in this packet
12	*(varies)*	sampleData	Sample data (depends on format).

Where *format*:*ee* is 2, the *sampleData* is in a 6-byte per sample format (older firmware only):

Offset (bytes)	Type	Name	Description
0	int16	xAxis	Signed X-axis value in 1/256 g units.
2	int16	yAxis	Signed Y-axis value in 1/256 g units.
4	int16	zAxis	Signed Z-axis value in 1/256 g units.

Where *format*:*ee* is 0, the *sampleData* is in a compressed 4-byte per sample format. With the sample encoded as a little-endian uint32 data type, the bit field is as follows (the little-endian encoding means the first byte sent is on the right):

```
eezzzzzz zzzzyyyy yyyyyyxx xxxxxxxx
```

Each 10-bit axis value must have the top bit sign-extended to the rest of the word, then bit-shifted left by *ee* (0-3) bits, result in 1/256 g units.

The sequence number for a given device restarts when the device has become idle and moved again. Some packets could be received twice (if the transmitter misses an acknowledgement) or could become lost due to radio conditions. These can be identified by observing non-sequential sequence ids for a given device.