Watch communication protocol v1

Connection procedure

Server MUST open a TCP server on a fixed port (TBD). The watch can connect to the server at any moment.

After the socket is successfully opened, the watch MUST send the version number as an 8 bit unsigned number. If the version is unknown to the server, the server SHOULD close the connection. Otherwise, if the version number is known, the server MUST continue to communicate according to that protocol version.

After the server has accepted the version number, it MUST send an ACK (00000110_2).

Now the watch and host are free to communicate in any order, per the rest of this document.

Note: A server is not required to support older version numbers, they are free to do so. But it's up to the implementator to choose the version range to support.

Universal message structure

command ID	number of parameters	parameters...
unsigned 8 bit	unsigned 8 bit	

Parameter

length of value in bytes	value
unsigned 16 bit	

Value types

Specified by every command, possible are:

- Binary data string;
- Null terminated ASCII string;
- Double precision IEEE float.

Message types

There are several message types, some bidirectional but most unidirectional.

Structure examples are in ASCII, the conversion to the real binary format should be trivial. The command ID is stated after every command name.

Bidirectional

PING (0)

This message can be send by either side in a self chosen interval. The other side should respond with a PONG within a maximum set amount of missed PINGs. Generally this is set to 3, so the other side can miss two PINGs and only then respond with a PONG to reset the miss counter.

Structure: `PING`

PONG (1)

The message to reply with a PING, should be sent in response to a PING.

Structure: `PONG`

Watch to host

INCREMENT (2)

An update to the live view on the host. Should be send every LIVE_INTERVAL.

The time delta is since the previous sent INCREMENT message in milliseconds.

Structure: `INCREMENT <sensor (ASCII string)> <time delta (double)> <data... (double)>`

PLAYBACK (3)

A data point for the full recorded session.

The time delta is since the start of the recording session in milliseconds.

Structure: `PLAYBACK <sensor (ASCII string)> <time delta (double)> <data... (double)>`

Host to watch

SENSOR_INTERVAL (4)

Set the interval for a given sensor in milliseconds.

Structure: `SENSOR_INTERVAL <sensor (ASCII string)> <interval (double)>`

SENSOR_SETTING (5)

Set a specific sensor dependent setting for the watch.

Structure: `SENSOR_SETTING <sensor (ASCII string)> <setting name (ASCII string)> <setting value (binary string)>`

Sensor settings

The settings available per sensor should be listed here.

LIVE_INTERVAL (6)

Set the interval in milliseconds for sending aggregated updates for the host's live dashboard.

Structure: `LIVE_INTERVAL <interval (double)>`

Example

Accelerometer live view update, 100ms after the previous one. Values are in order of x,y,z acceleration.

In ASCII: `INCREMENT 5 accel 100 5.43894 3.47392 1.32419`

In binary: `00000010000001010000000000000011001100001011000110110001101100101011011000000000000`