MQTT protocol for F1 EEG Cap

(as obtained by re-engineering, May 2024, Christoph Reichert)

Overview of topics

Send-topics:

Topic	**Argument**	**Comment**
action/sampling/start	Sampling Parameter (JSON format)	starts data transfer
action/sampling/stop	-	stops data transfer
action/impedance/start	TBA	starts impedance measurement

Recieve-topics:

Topic	**Data format (exempl. output)**	**Comment**
mqtt	string („accepted“)	
state/battery/amp/charge	int32	% battery
state/battery/amp/voltage	float32	battery voltage
state/battery/amp/critical	int32? (0)	
state/battery/mrk/charge	boolean?	
state/battery/mrk/critical	int32? (0)	
state/led/ready	string („on“)	LED (ready) Status
state/led/battery	string („off“)	LED (battery) Status
state/recording/active	boolean	
state/device/info	JSON ({string/value pairs})	recording parameters
state/impedance/acceptable	boolean (false)	
state/impedance/active	JSON or false	
state/sampling/active	JSON or false	
state/sampling/short_circuit	boolean	
state/storage/critical	int32? (0)	
state/storage/free	float32	
state/pairing	JSON ({string/value pairs})	
data/event	JSON	
data/samples	int32 buffer	EEG Data chunks
error	JSON	

Communicate with the F1 module

- First, establish a mqtt connection to IP address 172.31.1.1
- read mqtt log until topic "state/device/info" is received
- extract float value “scale_to_uV”
- send topic “action/sampling/start” and parameter in JSON format, e.g.

  ```
  {"channel_label":["Fp1“,“Fpz“,“Fp2“,“F7“,
               „F3“,“Fz“,“F4“,“F8“,
               „T3“,“C3“,“Cz“,“C4“,“T4“,
               „T5“,“P3“,“Pz“,“P4“,“T6“,
               „O1“,“Oz“,“O2“,“A1“,“A2“],
               „data_format“:0.0,“gain“:12.0,
               „impedance_interval“:0.0,
               „layout“:1.0,
               „marker_id“:“„,
               „output_rate“:20.0,
               „radio_bandw“:13.0,
               „radio_chan“:1.0,"
               „reference“:[„Fpz“],
               „sampling_rate“:500.0}
  ```
- continuously read mqtt logs
- interpret the topics and use the data where necessary
- most interesting part is topic "data/samples" which is a buffer of 32bit values, where the first two elements have format uint32 and give the start and end sample position of the chunk (so the first two values serve as a kind of header). The number of samples can therefore be determined by subtracting the first from the second value and the number of channels by dividing the number of values in the buffer (without the two header values) by the number of samples.
- the integer values must be multiplied with the scale_to_uV value to achieve the exact EEG signal
- when your program finishes, send topic “action/sampling/stop” and wait a second for further log messages
- finally, close the mqtt connection