LAPSNAPPER SOCKET SERVER SPECIFICATION v.5.20

Last updated 29th of Oct, 2024

Note: LapSnapper Socket Server version 5.20 is supported by LapSnapper software version 5.20 and newer.

E-mail: info@lapsnapper.com
Internet: www.lapsnapper.com

LapSnapper
FINLAND
Email: info@lapsnapper.com
Website address: www.lapsnapper.com

Information in this document is subject to change without notice. LapSnapper reserves the right to change or improve its products and to make changes in the content without obligation to notify any person or organization of such changes or improvements. Visit the LapSnapper website (www.lapsnapper.com) for current updates and supplemental information concerning the use and operation of this and other LapSnapper products.

CONTENTS

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

1. VERSION HISTORY

Version	Date	Description
5.20	29th of Oct,2024	Changes: - 2 notes added to the 5.2 Time Events Message - Editorial changes of the 6.3 Start Session Message
5.20	3rd of Feb,2021	ID for Volare is added
5.20	9th of Dec,2020	More examples of the binary coded messages are added
5.20	7th of Dec,2020	Editorial changes
5.20	20th of Nov,2020	Message header is object and productID is renamed to lapSnapperProductID
5.20	16th of Nov,2020	Examples are added
5.20	14th of Feb,2020	Session parameters data field is updated
5.20	10th of Feb, 2020	Megatiming product ID is added
5.20	3rd of Feb, 2020	Remote control interface is added
2.00	19th of Oct, 2018	JSON based message coding is introduced
1.00	10th of Jan, 2017	LapSnapper Socket Server Interface is introduced

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

2. INTRODUCTION

The document descripes the LapSnapper Socket Server API for the third party software developers.

LapSnapper Socket Server version 5.20 is supported by the LapSnapper software version 5.20 and newer.

Please see below the LapSnapper System Diagram:

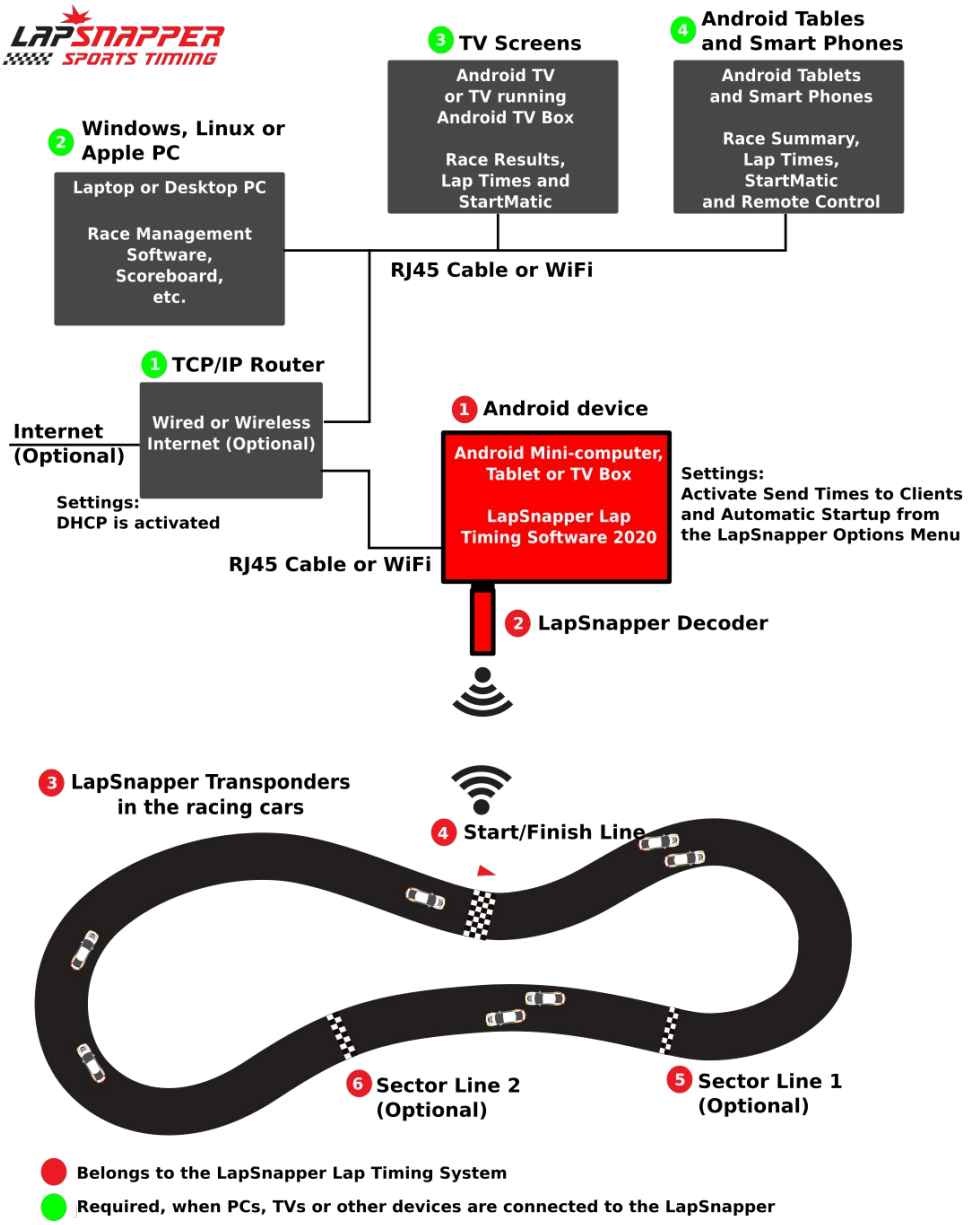

Table 1: LapSnapper System Diagram

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

3. SOCKET MESSAGE

3.1. SOCKET MESSAGE FRAME

Messages are encoded by using JSON. JSON encoded messages are packed to the socket message frame:

byte 1		byte 6 + n
2 bytes - Message ID	4 bytes - Message length (= n)	n bytes - Message decoded by using JSON

Message ID and message length are encoded in Little-endian format.

3.2. EXAMPLE CODE ABOUT SENDING THE SOCKET MESSAGES

Step 1. Create JSON

```
// Create JSON object
JSONObject message = new JSONObject();

try
{
	////////////////////////////////////////////////////////////////////////////////
	// Build message header data field
	buildMessageHeader(message,
			MSG_LAPSNAPPER_SERVER_API_GET_SERVER_INFO,
			productID,
			supportedServerVersion);

	////////////////////////////////////////////////////////////////////////////////
	// Serialize password data field
	JSONArray passwords = new JSONArray();
	for (int i = 0; i < password.length; i++)
	{
		passwords.put(password[i]);
	}
	message.put("password", passwords);
}
catch (JSONException e)
{
	CError.exception(e);
}
```

Step 2. Build Socket Message

```
////////////////////////////////////////////////////////////////////////////////
// Get payload and payload length
final byte[] payload = message.toString().getBytes());

// Calculate message length and byte array for the socket message
int messageLength = 6 + payload.length;
byte[] serializedMessage = new byte[messageLength];

// Serialize socket message frame header
int index = 0;
index = CDataSerializer.serializeSocketMessageFrameHeader(
                        serializedMessage,
                        index,
                        messageID,
                        payloadLength);

// Serialize socket message payload
CDataSerializer.serializeByteArray(serializedMessage, index, payload);
```

Step 3. Post Socket Message

```
// Post the socket message
writeData(serializedMessage);
```

3.3. EXAMPLE ABOUT THE BINARY CODED SOCKET MESSAGES

---------- Information ----------

Handshaking (Used client ID 1000000)

11:04:59.680 <Tag> Client <I> GET SERVER INFORMATION
70 17 6E 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 30 30 30 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 61 73 73 77 6F 72 64 22 3A 5B 31 2C 32 2C 33 2C 34 5D 7D

11:04:59.705 <Tag> Server <I> SERVER INFORMATION
71 17 D8 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 30 30 31 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 36 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30 7D 2C 22 64 65
63 6F 64 65 72 49 6E 66 6F 22 3A 7B 22 70 72 6F 64 75 63 74 49 44 22 3A 31 2C 22 76 65 72 73
69 6F 6E 4E 75 6D 62 65 72 22 3A 35 31 30 2C 22 64 65 76 69 63 65 54 79 70 65 22 3A 32 2C 22
64 65 76 69 63 65 49 44 22 3A 31 30 30 32 32 2C 22 64 65 63 6F 64 65 72 4D 6F 64 65 22 3A 31
2C 22 64 65 63 6F 64 65 72 53 74 61 74 75 73 22 3A 31 7D 2C 22 73 79 73 74 65 6D 53 74 61 74
65 22 3A 30 7D

Normal operation, when the connection is active

11:04:59.917 <Tag> Server <I> HEATBEAT
73 17 54 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 30 30 33 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 36 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30 7D 7D

11:05:01.918 <Tag> Server <I> HEATBEAT
73 17 54 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 30 30 33 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 36 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30 7D 7D

11:05:03.921 <Tag> Server <I> HEATBEAT
73 17 54 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 30 30 33 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 36 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30 7D 7D

11:05:08.290 <Tag> Client <I> START SESSION (Remote control password: “1234”)
02 19 33 02 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 34 30 32 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 69 6E 43 6F 64 65 22 3A 22 31 32 33 34 22 2C 22 73 65 73 73 69 6F 6E 50 61 72 61
6D 65 74 65 72 73 22 3A 7B 22 73 65 73 73 69 6F 6E 54 79 70 65 22 3A 30 2C 22 74 72 69 67 67
65 72 4C 69 6E 65 43 6F 75 6E 74 22 3A 31 2C 22 73 74 61 72 74 4C 69 6E 65 22 3A 30 2C 22 73
74 6F 70 54 79 70 65 22 3A 31 2C 22 73 74 6F 70 56 61 6C 75 65 22 3A 2D 31 2C 22 63 6F 6E 74
69 6E 75 6F 75 73 53 65 73 73 69 6F 6E 22 3A 66 61 6C 73 65 2C 22 70 65 72 73 6F 6E 61 6C 53
74 6F 70 70 69 6E 67 43 72 69 74 65 72 69 61 22 3A 66 61 6C 73 65 2C 22 72 65 73 75 6C 74 73
41 72 65 4F 72 64 65 72 65 64 41 67 61 69 6E 73 74 54 6F 74 61 6C 54 69 6D 65 22 3A 66 61 6C
73 65 2

11:05:23.166 <Tag> Client <I> START SESSION (Remote control password: “1234”)
02 19 33 02 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 34 30 32 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 69 6E 43 6F 64 65 22 3A 22 31 32 33 34 22 2C 22 73 65 73 73 69 6F 6E 50 61 72 61
6D 65 74 65 72 73 22 3A 7B 22 73 65 73 73 69 6F 6E 54 79 70 65 22 3A 30 2C 22 74 72 69 67 67
65 72 4C 69 6E 65 43 6F 75 6E 74 22 3A 31 2C 22 73 74 61 72 74 4C 69 6E 65 22 3A 30 2C 22 73
74 6F 70 54 79 70 65 22 3A 31 2C 22 73 74 6F 70 56 61 6C 75 65 22 3A 2D 31 2C 22 63 6F 6E 74
69 6E 75 6F 75 73 53 65 73 73 69 6F 6E 22 3A 66 61 6C 73 65 2C 22 70 65 72 73 6F 6E 61 6C 53
74 6F 70 70 69 6E 67 43 72 69 74 65 72 69 61 22 3A 66 61 6C 73 65 2C 22 72 65 73 75 6C 74 73
41 72 65 4F 72 64 65 72 65 64 41 67 61 69 6E 73 74 54 6F 74 61 6C 54 69 6D 65 22 3A 66 61 6C
73 65 2C 22 73 65 73 73 69 6F 6E 53 74 6F 70 53 74 61 72 74 73 57 68 65 6E 4C 65 61 64 65 72
52 65 61 63 68 65 73 53 74 6F 70 56 61 6C 75 65 22 3A 66 61 6C 73 65 7D 2C 22 74 72 61 63 6B
53 65 74 74 69 6E 67 73 22 3A 7B 22 6D 69 6E 69 6D 75 6D 4C 61 70 54 69 6D 65 45 6E 61 62 6C
65 64 22 3A 66 61 6C 73 65 2C 22 6D 69 6E 69 6D 75 6D 4C 61 70 54 69 6D 65 22 3A 31 35 30 30
30 2C 22 6D 69 6E 69 6D 75 6D 53 65 63 74 6F 72 31 54 69 6D 65 22 3A 35 30 30 30 2C 22 6D 69
6E 69 6D 75 6D 53 65 63 74 6F 72 32 54 69 6D 65 22 3A 35 30 30 30 2C 22 6D 69 6E 69 6D 75 6D
53 65 63 74 6F 72 33 54 69 6D 65 22 3A 35 30 30 30 2C 22 6D 61 78 69 6D 75 6D 4C 61 70 54 69
6D 65 45 6E 61 62 6C 65 64 22 3A 66 61 6C 73 65 2C 22 6D 61 78 69 6D 75 6D 4C 61 70 54 69 6D
65 22 3A 32 34 30 30 30 30 7D 7D

11:05:25.707 <Tag> Server <I> STARTMATIC STATUS (Remote control password: “1234”)
03 19 A7 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 34 30 33 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 69 6E 43 6F 64 65 22 3A 22 31 32 33 34 22 2C 22 73 74 61 72 74 4D 61 74 69 63 53
74 61 74 75 73 22 3A 31 2C 22 73 74 61 72 74 4D 61 74 69 63 44 61 74 61 22 3A 36 30 2C 22 73
74 61 72 74 4D 61 74 69 63 44 61 74 61 32 22 3A 31 7D

11:05:27.341 <Tag> Server <I> STARTMATIC STATUS (Remote control password: “1234”)
03 19 A8 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 34 30 33 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 69 6E 43 6F 64 65 22 3A 22 31 32 33 34 22 2C 22 73 74 61 72 74 4D 61 74 69 63 53
74 61 74 75 73 22 3A 34 2C 22 73 74 61 72 74 4D 61 74 69 63 44 61 74 61 22 3A 31 35 2C 22 73
74 61 72 74 4D 61 74 69 63 44 61 74 61 32 22 3A 2D 31 7D

11:05:30.266 <Tag> Server <I> STARTMATIC STATUS (Remote control password: “1234”)
03 19 A7 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 34 30 33 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 69 6E 43 6F 64 65 22 3A 22 31 32 33 34 22 2C 22 73 74 61 72 74 4D 61 74 69 63 53
74 61 74 75 73 22 3A 36 2C 22 73 74 61 72 74 4D 61 74 69 63 44 61 74 61 22 3A 31 2C 22 73 74
61 72 74 4D 61 74 69 63 44 61 74 61 32 22 3A 2D 31 7D

11:06:46.413 <Tag> Server <I> TIME EVENT
DE 17 E7 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 31 31 30 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 36 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30 7D 2C 22 74 72
61 6E 73 70 6F 6E 64 65 72 49 44 22 3A 31 30 30 30 33 2C 22 74 72 61 6E 73 70 6F 6E 64 65 72
49 6E 73 74 61 6E 63 65 22 3A 31 2C 22 74 72 61 6E 73 70 6F 6E 64 65 72 4E 61 6D 65 22 3A 22
4D 61 72 6B 22 2C 22 61 62 73 6F 6C 75 74 65 54 69 6D 65 22 3A 7B 22 74 69 6D 65 53 74 61 6D
70 49 44 22 3A 31 35 37 2C 22 61 62 73 6F 6C 75 74 65 54 69 6D 65 22 3A 31 36 30 37 35 30 34
38 30 33 33 36 31 7D 2C 22 6C 61 70 4C 69 6E 65 22 3A 31 7D

11:06:57.625 <Tag> Server <I> TIME EVENT
DE 17 EA 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 31 31 30 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 36 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30 7D 2C 22 74 72
61 6E 73 70 6F 6E 64 65 72 49 44 22 3A 31 30 30 30 34 2C 22 74 72 61 6E 73 70 6F 6E 64 65 72
49 6E 73 74 61 6E 63 65 22 3A 31 2C 22 74 72 61 6E 73 70 6F 6E 64 65 72 4E 61 6D 65 22 3A 22
4D 69 63 68 61 65 6C 22 2C 22 61 62 73 6F 6C 75 74 65 54 69 6D 65 22 3A 7B 22 74 69 6D 65 53
74 61 6D 70 49 44 22 3A 31 36 30 2C 22 61 62 73 6F 6C 75 74 65 54 69 6D 65 22 3A 31 36 30 37
35 30 34 38 31 34 35 39 30 7D 2C 22 6C 61 70 4C 69 6E 65 22 3A 31 7D

11:07:06.773 <Tag> Client <I> STOP SESSION (Remote control password: “1234”)
00 19 6A 00 00 00 7B 22 6D 65 73 73 61 67 65 48 65 61 64 65 72 22 3A 7B 22 6D 65 73 73 61 67
65 49 44 22 3A 36 34 30 30 2C 22 6C 61 70 53 6E 61 70 70 65 72 50 72 6F 64 75 63 74 49 44 22
3A 31 30 30 30 30 30 30 2C 22 73 65 72 76 65 72 56 65 72 73 69 6F 6E 22 3A 35 32 30 30 30 30
7D 2C 22 70 69 6E 43 6F 64 65 22 3A 22 31 32 33 34 22 7D

4. CONNECTION MANAGEMENT INTERFACE

4.1. CONNECT TO THE LAPSNAPPER SOCKET SERVER

Create socket connection to the LapSnapper Socket Server IP address (Port 9001) and send Get LapSnapper Server Information message to the LapSnapper Socket Server.

Step 1: Send Get LapSnapper Server Information message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6000,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"password": ["1", "2", "3", "4"]
}
	
	See coding 7.1
	Socket socket server version supported by client (5.20.000)

Table 2: Encoding of Get Server Information message

Step 2a: LapSnapper Socket Server responds to Get LapSnapper Server Information message by sending LapSnapper Server Information message when successful.

JSON	Notes

{
"messageHeader": {
"messageID": 6001,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"decoderInfo": {
"productID": Integer,
"versionNumber": Integer,
"deviceType": Integer,
"deviceID": Integer,
"decoderMode": Integer,
"decoderStatus": Integer,
},
"systemState": Integer
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	Example 520000 = 5.20.000
	Types: 1 or 2
	Decoder device ID
	1 = host and 2 = slave
	
	System state (See coding 4.3)

Table 3: Encoding of LapSnapper Server Information message

Step 2b: LapSnapper Socket Server responds to Get LapSnapper Server Information message by sending LapSnapper Server Information Failed message when LapSnapper Server version number or server type fails.

JSON	Notes

{
"messageHeader": {
"messageID": 6002,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"status": 1
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	Unsupported LapSnapper Socket Server version

Table 4: Enoding of LapSnapper Server Information Failed message

Step 2c: LapSnapper Socket Server disconnects the client when LapSnapper Server password fails or incorrect message has been received.

4.2. HEARTBEATS

LapSnapper Socket Server sends the heartbeats to all clients every 2 seconds for keeping the connection in alive.

JSON	Notes

{
"messageHeader": {
"messageID": 6003,
"lapSnapperProductID": Integer,
"serverVersion": 520000
}
}
	
	See coding 7.1
	Socket socket server version supported by client (5.20.000)

Table 5: Encoding of Heartbeat message

4.3. DISCONNECT THE CLIENT

Disconnect socket connection.

5. TIME MEASUREMENT INTERFACE

5.1. GET DECODER TIME

Step 1: Send Get Decoder Time message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6100,
"lapSnapperProductID": Integer,
"serverVersion": 520000
}
}
	
	See coding 7.1
	Socket socket server version supported by client (5.20.000)

Table 6: Encoding of Get Decoder Time message

Step 2a: LapSnapper Socket Server responds to Get Decoder Time message by sending Decoder Time message when successful.

JSON	Notes

{
"messageHeader": {
"messageID": 6101,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"decoderTime": Long
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	Decoder time in milliseconds

Table 7: Encoding of Decoder Time message

Step 2a: LapSnapper Socket Server responds to Get Decoder Time message by sending Decoder Time Failed message, when decoder time wasn't available.

JSON	Notes

{
"messageHeader": {
"messageID": 6102,
"lapSnapperProductID": Integer,
"serverVersion": 520000
}
}
	
	See coding 7.1
	Socket Server version (5.20.000)

Table 8: Encoding of Decoder Time Failed message

5.2. TIME EVENTS

LapSnapper Socket Server sends Time Event message to the clients, when new time is available for the transponder.

JSON	Notes

{
"messageHeader": {
"messageID": 6110,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"transponderID": Integer,
"transponderInstance": Integer,
"transponderName": String,
"absoluteTime": {
"timeStampID": Integer,
"absoluteTime": Long
},
"lapLine": Integer
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	Time stamp ID
	Time stamp in milliseconds
	
	Available only, when session is running and sectors are used:
	1 if lap line and 0 if sector line

Table 9: Encoding of Time Event message

Note: More than one a time event messages may come at once. This may happen, if sector times are used or transponder wasn't able to send previous lap time for the decoder. It means client software has to have capability to handle more than one a time event messages in a short period.

Note 2: Order of a time event messages are not guaranteed. In special cases older time stamp may come later. This may happen, if decoder USB connection temporary disconnects.

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

6. REMOTE CONTROL INTERFACE

The Remote Control Interface is controlled from the options menu by selecting "Remote control settings".

6.1. STOP SESSION

Send Stop Session message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6400,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"pinCode": String
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	PIN code

Table 10: Encoding of Stop Session message

6.2. EXIT SESSION

Send Exit Session message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6401,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"pinCode": String
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	PIN code

Table 11: Encoding of Exit Session message

6.3. START SESSION

Send Start Session message to LapSnapper Socket Server.

JSON	Notes
{	
"messageHeader": {	
"messageID": 6402,	
"lapSnapperProductID": Integer,	See coding 7.1
"serverVersion": 520000	Socket Server version (5.20.000)
},	
"pinCode": String,	PIN code
"sessionParameters": {	See coding 7.3
"sessionType": Integer,	Session type
"triggerLineCount": Integer,	Sector count
"startLine": Integer,	Start line
"stopType": Integer,	Stop type
"stopValue": Integer,	Stop value
"continuousSession": Boolean,	Continuous session
"personalStoppingCriteria": Boolean,	Personal stopping criteria
"resultsAreOrderedAgainstTotalTime": Boolean,	Results of the qualifying session are ordered against total time
"sessionStopStartsWhenLeader ReachesStopValue": Boolean	Session is stopped when the leader has finished
},	
"trackSettings": {	
"minimumLapTimeEnabled": Boolean,	Minimum lap time is enabled
"minimumLapTime": Integer,	Minimum lap time
"minimumSector1Time": Integer,	Minimum sector 1 time
"minimumSector2Time": Integer,	Minimum sector 2 time
"minimumSector3Time": Integer,	Minimum sector 3 time
"maximumLapTimeEnabled": Boolean,	Maximum lap time is enabled
"maximumLapTime": Integer	Maximum lap time
}	
}	

Table 12: Encoding of Start Session message

6.4. STARTMATIC STATUS

Send StartMatic Status message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6403,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"pinCode": String,
"startMaticStatus": Integer,
"startMaticData": Long,
"startMaticData2": Long
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	PIN code
	See StartMatic Statuses from table 14

Table 13: Encoding of StartMatic Status message

StartMatic Statuses:

Value	Notes
1 startMaticData startMaticData2	StartMatic is opened - StartMatic Delay - StartMatic Mode
2	StartMatic is closed
3	StartMatic is canceled
4 startMaticdata	StartMatic Delay - StartMatic Delay
5 startMaticData	StarMatic Mode - StartMatic Mode
6	StartMatic countdown is started

Table 14: Encoding of StartMatic Statuses

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

6.5. SESSION INFORMATION

Send Session Information message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6404,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"pinCode": String,
"sessionInformation": {
"sessionName": String,
"allTranspondersActive": Boolean,
"driverInformation": {
[{
"driverNumber": Integer,
"driverName": String,
"transponderID": {
"transponderID": Integer,
"transponderInstance": Integer
},
"transponderEnabled": Boolean
}
]
}
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	PIN code
	
	Session name
	All transponders are active
	
	Driver number
	Driver name
	
	Transponder ID
	Transponder Instance (Default = 1)
	
	Transponder is enabled

Table 15: Encoding of Session Information message

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

6.6. RACE TRACK SETTINGS

Send Race Track Settings message to LapSnapper Socket Server.

JSON	Notes

{
"messageHeader": {
"messageID": 6405,
"lapSnapperProductID": Integer,
"serverVersion": 520000
},
"pinCode": String,
"trackSettings": {
"minimumLapTimeEnabled": Boolean,
"minimumlapTime": Integer,
"minimumSector1Time": Integer,
"minimumSector2Time": Integer,
"minimumSector3Time": Integer,
"maximumLapTimeEnabled": Boolean,
"maximumLapTime": Integer
}
}
	
	See coding 7.1
	Socket Server version (5.20.000)
	
	PIN code
	
	Minimum lap time is enabled
	Minimum lap time
	Minimum sector 1 time
	Minimum sector 2 time
	Minimum sector 3 time
	Maximum lap time is enabled
	Maximum lap time

Table 16: Encoding of Race Track Settings message

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

7. INFORMATION ELEMENTS

7.1. LAPSNAPPER PRODUCT ID DATA FIELD

Value	Description
50 200 10000 10001 10002 10003 10004 > 1000000	LapSnapper Product IDs: - LapSnapper Pro - LapSnapper Client - Kart timer (www.karttimer.fi) - RaceFacer (www.racefacer.com) - RC Timing (www.rc-timing.ch) - Megatiming (www.megatiming.se) - Volare (www.volarehq.com) - Free for all applications

Table 17: Encoding of product ID data field

7.2. SYSTEM STATE DATA FIELD

Value	Description
0 1 2 3 4 5 6 7 8 9 10 -1	System state - Main menu state - Session options dialog state - Timing state - Timing recovery dialog state - Results dialog state - Results browsing state - Results browsing save results dialog state - Settings dialog state - Save results restart dialog state - Save results dialog state - Automatic session startup state - Unknown state

Table 18: Encoding of System State data field

7.3. SESSION PARAMETERS DATA FIELD

Value	Description
0 1 2	Session type - Practice - Qualification - Race
1, 2 or 3	Sector count - 1, 2 or 3 sectors
0, 1 or 2	Start line - Start/Finish line is 1st, 2nd or 3rd sector line
1 2 3	Stop type - Manual - Session time - Lap count
N/A X seconds X laps	Stop value - Manual stop: N/A - Session time stop: Session time in seconds - Lap count stop: lap count
False True	Continuous session (Practice and qualification sessions) - Disabled - Enabled
False True	Personal stopping criteria (Practice and qualification sessions) - Disabled - Enabled
False True	Time order of the qualification session - Results are ordered against the best lap time - Results are ordered against the total time
False True	Session is stopped when the leader has finished - Session is stopped when the session time has ended or lap count is reached - Session is stopped when the leader has finished

Table 19: Encoding of Session Parameters data field

Email: info@lapsnapper.com
Internet: www.lapsnapper.com

APPENDIX A: TABLES