



SENSEMOTION

Dongle Interface

Document Title:	QSense - Dongle Interface
Company:	2M Engineering Ltd – John F Kennedylaan 3, 5555XC, Valkenswaard, The Netherlands
Project:	QSense
Author:	Jose Lopez Hidalgo



Content

1	Introduction.....	3
2	Getting started.....	4
3	Serial Interface.....	5
	3.1 Transmit.....	5
	3.2 Receive.....	5
	3.3 Connect.....	6
4	Disconnect	6
5	Stop Scan	6
6	Status	6



1 Introduction

The QSense IMU Motion Sensing Platform is a wearable system intended to monitor the 3D orientation of human body segments from linear acceleration, angular velocity, and magnetometer measurements. It provides easy data collection for a wide range of applications, including research, sports and exercise science, ergonomics and (virtual reality) games.

One of the components of this system is the QSense BLE USB Dongle. With the dongle the connection and communication with up to 13 QSense sensors is reduced to a single serial interface. Thus, it makes the integration of the QSense sensors into your application much simpler and faster.

This interface will provide you with the necessary tools and resources to access all the functionalities of the QSense sensors through the QSense Dongle via its Serial Interface.

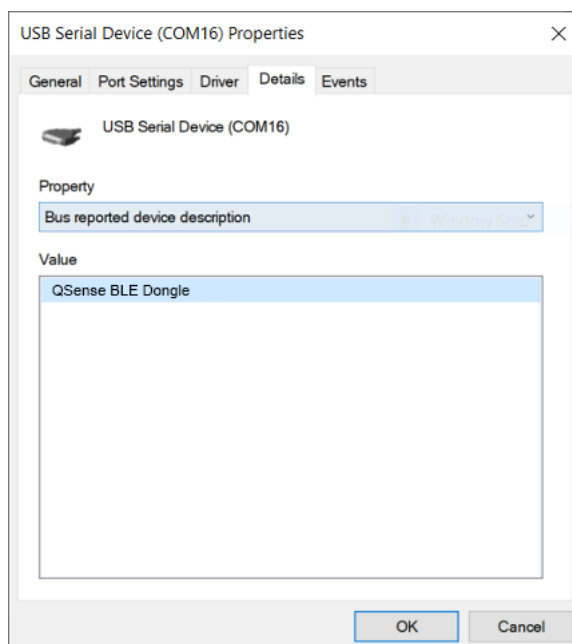
Code examples are available in [QSense-Examples](#).

NOTE: You should read and understand the “QSense Sensor Interfaces” document before going through this guide.



2 Getting started

To begin using the QSense Dongle and interact with QSense sensors, the first thing you need to do is insert the dongle into one of the USB ports or your laptop/computer. The dongle will appear in the list of devices with the name “QSense BLE Dongle” as shown in the picture below.



For optimal communication, you should configure the serial port with the following settings:

- Baud rate: 460800
- Parity: None
- Stop bits: One

When opening the serial port, make sure to enable Data Terminal Ready (DTR) to properly initialize communication.

For more details on serial communication, please refer to the external documentation linked [here](#).



3 Serial Interface

The dongle Serial Interface provides access to the Core Interface of the QSense sensors over a single serial communication and wraps the Core Interface packet in an ASCII encoded message. You can find a detailed description of the device's Core Interface in the 'QSense Motion Sensor Interfaces' document.

Besides the Core Interface packet, additional packets can be transmitted and received. The Serial Interface messages follow the following structure:

- Prefix: A Dollar character '\$' (hex value 0x24) that marks the start of the packet.
- Opcode: A single character indicating the packet type.
- Data: Optional field containing a hex encoded byte array. Every byte consists of two characters between '0..9' or 'A..F', representing hexadecimal notation. The first 2 characters of the Data part in the message represent the 1st byte of the hex encoded byte array. The 3rd and 4th character represent the 2nd byte of the hex encoded byte array, etcetera.
- Postfix: A Line Feed character '\n' (hex value 0x0A) marks the end of the packet.

Byte Address	0	1	2	...	N-1	N
Field	Prefix	Opcode	Data (optional)			Postfix
Character	\$	T/R/C/D/I/S				\n

Table 1. Packet fields and byte order of the Serial Interface packets.

NOTE: The dongle can only handle single operations at a time. It is advised to either wait until all devices are connected or to stop the scanning manually before starting to stream data from the QSense sensors.

3.1 Transmit

This packet allows you to send a Core Interface packet to the recorder via the serial interface. It uses opcode 'T' (hex value 0x54). The data field of this packet represents the encapsulated Core Interface packet as a hex encoded byte array., preceded by the handle of the QSense sensor we want to transmit the packet to.

If a response is necessary, the recorder will reply with a Receive packet.

Byte Address	0	1	2	...	N-1	N
Field	Prefix	Opcode	Data			Postfix
Character	\$	T	Sensor Handle + Core Interface packet			\n

Table 2. Transmit packet.

3.2 Receive

This packet type is sent by the recorder in response to a Transmit packet. Its opcode 'R' (hex value 0x52). The data field of this packet represents the encapsulated Core Interface packet as a hex encoded byte array, preceded by the handle of the QSense sensor that sent the Core Interface packet.

Byte Address	0	1	2	...	N-1	N
Field	Prefix	Opcode	Data			Postfix
Character	\$	R	Sensor Handle + Core Interface packet			\n

Table 3. Receive packet.



3.3 Connect

This packet is used to make the dongle scan for BLE devices until all possible connections are achieved. The data field of this packet contains the maximum number of connections the dongle is going to maintain, followed by the name of the QSense sensors in hexadecimal, which is "5153656E7365". The absolute maximum number of connections that the dongle can maintain is 13 (hex value 0x0D). Its opcode 'C' (hex value 0x43).

Byte Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Field	Prefix	Opcode	Data														Postfix
Character	\$	C	0	D	5	1	5	3	6	5	6	E	7	3	6	5	\n

Table 4. Connect packet.

3.4 Connect Whitelist

This packet is used to make the dongle scan for specific BLE devices. The data field of this packet contains the number of devices in the whitelist, followed by 6-byte long MAC address of every device. The whitelist can contain up to 13 devices. Its opcode 'W' (hex value 0x 57).

Byte Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Field	Prefix	Opcode	Data														Postfix
Character	\$	W	0	1	F	F	F	F	F	F	F	F	F	F	F	F	\n

Table 5. Example of a Connect Whitelist packet, where there is only one device in the whitelist, and its MAC address is 0xFFFFFFFFFFFF.

3.5 Disconnect

A Disconnect packet is used to disconnect from the QSense sensors connected to dongle. Before disconnecting from the devices, the dongle rewrites the pin register in the devices to disable the memory access. Unlike the previous packets, this one only contains the prefix, the opcode which is 'D' (hex value 0x44), and the postfix.

Byte Address	0	1	3
Field	Prefix	Opcode	Postfix
Character	\$	D	\n

Table 6. Disconnect packet.

3.6 Stop Scan

The Stop scan packet allows the user to stop the scan for devices even if the dongle has not connected to the maximum number of devices set in the Connect packet. The opcode of this command is 'I' (hex value 0x49).

Byte Address	0	1	3
Field	Prefix	Opcode	0x0A
Character	\$	I	\n

Table 7. Stop Scan packet.

3.7 Status

Status packets can be sent by either the user or the dongle. When the user sends this packet, it only contains the prefix, the opcode 'S' (hex value 0x53) and the postfix. The dongle will respond by sending another Status packet whose data field contains the maximum data size of the packets that it can send to the QSense sensors, and the connection status of each of the handles. The maximum data size is typically 244 bytes (hex value 0xF4). This status will be 0 if the dongle is in idle mode, 1 if it is scanning and 2 if it is connected to a QSense sensor on that handle position.



Byte Address	0	1	2
Field	Prefix	Opcode	Postfix
Character	\$	S	\n

Table 8. Status packet request.

Byte Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Field	Prefix	Opcode	Data													
Character	\$	S	F	4	0	2	0	1	0	1	0	1	0	1	0	1
Byte Address	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Field															Postfix	
Character	0	1	0	1	0	1	0	1	0	1	0	1	0	1	\n	

Table 9. Example of Status packet response, showing that the dongle is connected to one device and is scanning for the remaining ones.