Racelogic Serial Data Stream

The Racelogic Serial data stream uses the following protocol:
115200 Baud, no parity, 8 data bits, 1 stop bit
The Serial message is made up of two parts, message 1 and message 2. Message 1 contains the VBOX GPS standard channels and Message 2 contains any VBOX CAN module channels that have been selected to be included in the VBOX serial data stream.
Note: On the VBOX20SL and SL3 the internal slip, pitch and roll angle channels are included Message 2.

Message 1 format: $VBOXII,nnnn0000,stttaaaaoooovvhheeezzmmmggcc

The first part of Message 1 ($VBOXII) indicates which VBOX the serial data is coming from.
VBOXIIDCF = $VBOXII
VBOXIIS/VBOX2SX = $VB2SX$
VBOX2SX-10 = $VBSX10
VBOX20SL = $VB2SL$
The $VBOXII and comma part of the message are in ASCII, the rest is in binary.
The first four bytes after the header are reserved, nnnn0000 bytes indicate the presence of the channels in the serial data stream. For example if only Sats and Velocity are present the corresponding bit masks 0x00000001 and 0x00000010 would equate to the nnnn0000 bytes equaling 0x0000001100000000.

	Bytes	Description	**nnnn,** bit mask.
nnnn0000	8	Reserved to indicate channel presence	
s	1	**Satellites** Number of satellites	**0x00000001**
ttt	3	**Time** Number of 10ms ticks since midnight UTC	**0x00000002**
aaaa	4	**Latitude** (DDMM.MMMMM * 100,000) Highest bit indicates east / west hemisphere and must be masked from latitude value 0=North,1=South	**0x00000004**
oooo	4	**Longitude** (DDDMM.MMMMM * 100,000) Highest bit indicates north / south hemisphere and must be masked from longitude value 0=West, 1=East	**0x00000008**
vv	2	**Velocity** Velocity in Knots * 100	**0x00000010**
hh	2	**Heading** Degrees from true north * 100	**0x00000020**
eee	3	**Height** Altitude. In metres WGS84 * 100 True signed 24 bit number	**0x00000040**

mmm	3	**Internal RAM pointer** Current position of internal RAM pointer	**0x08000000**
gg	2	**Event time** Number of clock ticks since Trigger input (11570 = 50ms)	**0x10000000**
zz	2	**Vertical Velocity** Vertical velocity in m/s	**0x00000080**
cc	2	**Checksum** CRC of message, see Note 1*	

Below is an example of the Message 1 serial message from a VBOX2SX

Message 1

24 56 42 32 53 58 24 2C 18 00 00 7F 00 00 00 00 2C 00 00 1F C2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15 00 00 07 E5

Header
24 56 42 32 53 58 24 2C
$VB2SX$,

Available Channels
18 00 00 7F = (0001 1000 0000 0000 0000 0000 0111 1111)
00 00 00 00 = (0000 0000 0000 0000 0000 0000 0000 0000)

LSB Byte (18 00 00 7F)
00 01 = Satellites 1 byte
00 10 = Time 3 bytes
01 00 = Latitude 4 bytes
10 00 = Longitude 4 bytes

2nd Byte (18 00 00 7F)
00 01 = Velocity 2 bytes
00 10 = Heading 2 bytes
01 00 = Height 3 bytes

7th Byte (18 00 00 7F)
00 10 = Memory Used 3 bytes

8th Byte (18 00 00 7F)
01 00 = Trigger event time 2 bytes (4 for vb3)

Separator
2C
,

Data
00 00 1F C2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 15 00 00

Satellites = 00
Time = 00 1F C2
Latitude = 00 00 00 00
Longitude = 00 00 00 00
Velocity = 00 00
Heading = 00 00
Height = 00 00 00
Memory Used = 00 00 15
Trigger event time = 00 00

CheckSum
07 E5

Racelogic CAN modules

When CAN channels are also selected to be present in the Serial data stream the data stream will look like the following:
$VBOXII$,nnnn0000,stttaaaaoooovvhheeezzmmmggcc $NEWCAN,XXXX,YYYY.....YYYYZZ

$NEWCAN,XXXX,YYYY.....YYYYZZ this is the additional CAN channel information to the serial data stream.
The ZZ is a two byte checksum. - This is the same format as before see NOTE 1 below

The format for the serial string is $NEWCAN,XXXX,YYYY.....YYYYZZ

Where XXXX is a 32-bit value.

The length of YYYY depends upon the value of XXXX. For each bit set in XXXX the length of YYYY is incremented by four. i.e. if XXXX = $0001 then YYYY will be four bytes long. The order in which the data will come across depends upon the order in which you select the channels from the set-up screen. For example if the first channel you choose to log is FIM Channel 1 then the first 4 bytes will contain the data from the FIM. If the second channel was ADC02 Channel4 then the next 4 bytes will contain data for the ADC02. If you then remove (choose not to log) FIM Channel 1 and select ADC02 Channel 1 then the first 4 bytes will now contain the data for ADC02 Channel 1.

The data transmitted for each channel (YYYY) is in a Racelogic format.
If you take each YYYY you can split it into four bytes the first byte is a signed exponent the next three are a signed mantissa.

e.g. if YYYY = 0x03 0x12 0x11 0x00 this equates to exponent = 0x03
mantissa = 0x121100 or 1184000 decimal,

Below is the example of the NEW CAN message in the VBOX serial data stream.

Message 2

24 4E 45 57 43 41 4E 2C 00 00 00 00 2C 25 41

Header
24 4E 45 57 43 41 4E 2C
$NEWCAN,

Available Channels
00 00 00 00 = (0000 0000 0000 0000 0000 0000 0000 0000)

Separator
2C
,

Data
None available (no channels are selected, note each channel in this message is 4 bytes long)

CheckSum
25 41

***Note 1**

CRC Calculation example :
s[n] is a string containing the message (s[1] = $ from the beginning of the entire string)

```
Polynomial:= 4129
CRC:=0;
for Loop:=1 to Length(s) do
begin
        Temp:=s[Loop];
        CRC:= CRC xor (integer(Temp) * 256);
        CRC:= CRC mod 65536;
        for i:=7 downto 0 do
        begin
                if ( (CRC and 32768)=32768) then
                begin
                        CRC:= CRC *2 ;
                        CRC:= CRC xor Polynomial;
                end
                else
                begin
                        CRC:= CRC *2 ;
                end;
                CRC:=CRC mod 65536;
        end;
end;
result:=CRC;
```