Simplified comunication protocol of BCE generic devices

Socket type TCP (First string “#BCE#/r/n”) or UDP

Device Send Stack (History from flash)

FROM MDL to SRV

IMEI [64 bits] 8 bytes	LEN [16 bits] 2 bytes	Service ID [8 bits] 1 byte 0xA5	ConfirmationKeyA [8 bits] 1 byte	Dara Bytes N bytes	CS [8bits] 1 byte
Device number	Count of bytes without IMEI,LEN and CS	**AsyncStack**	Each try of sending Stack Key is incremented	Data structures	Sum of all packet bytes except CS

example packet (af b7 ac 79 f0 3f 01 00 09 00 a5 81 06 d6 7a 76 51 00 f0 f7)

IMEI (351777042773935)	LEN (9 bytes)	Data (AsynsStack)	CS
af b7 ac 79 f0 3f 01 00	09 00	a5 81 06 d6 7a 76 51 00 f0	f7
		Service: a5 | Key: 81 | Data struct: 06 d6 7a 76 51 00 f0	Sum of all packet bytes except CS

FROM SRV to MDL

IMEI [64 bits] 8 bytes	LEN [16 bits] 2 bytes	Service ID [8 bits] 1 byte 0x19	ConfirmationKeyB[8 bits] 1 byte	CS [8bits] 1 byte
Device number	Count of bytes without IMEI,LEN and CS	**StackConfirm**	ConfirmationKeyB = ConfirmationKeyA & 0x7F	Sum of all packet bytes except CS

example packet (af b7 ac 79 f0 3f 01 00 02 00 19 01 d7)

<table>
<tr><td>IMEI (351777042773935)</td><td>LEN (2bytes)</td><td>Data (StackConfirm)</td><td>CS</td></tr>
<tr><td>af b7 ac 79 f0 3f 01 00</td><td>02 00</td><td>19 01</td><td>d7</td></tr>
<tr><td></td><td></td><td><table><tr><td>Service</td><td>Key</td></tr><tr><td>19</td><td>01</td></tr></table></td><td>Sum of all packet bytes except CS</td></tr>
</table>

Time Trigered Packet Without Confirmation

ROM MDL to SRV

IMEI [64 bits] 8 bytes	LEN [16 bits] 2 bytes	Service ID [8 bits] 1 byte 0xA0	PAD byte[8 bits] 1 byte	Dara Bytes N bytes	CS [8bits] 1 byte
Device number	Count of bytes without IMEI,LEN and CS	**Time Trigered Packet**	PAD byte always 0.	Data structures	Sum of all packet bytes except CS

example packet (af b7 ac 79 f0 3f 01 00 09 00 a0 81 06 d6 7a 76 51 00 f0 f2)

<table>
<tr><td>IMEI (351777042773935)</td><td>LEN (9 bytes)</td><td colspan="3">Data (AsynsStack)</td><td>CS</td></tr>
<tr><td>af b7 ac 79 f0 3f 01 00</td><td>09 00</td><td colspan="3">a0 81 06 d6 7a 76 51 00 f0</td><td>f2</td></tr>
<tr><td></td><td></td><td colspan="3"><table><tr><td>Service</td><td>PAD</td><td>Data struct</td></tr><tr><td>a0</td><td>00</td><td>06 d6 7a 76 51 00 f0</td></tr></table></td><td>Sum of all packet bytes except CS</td></tr>
</table>

Output control

FROM SRV to MDL

IMEI [64 bits] 8 bytes	LEN [16 bits] 2 bytes	Service ID [8 bits] 1 byte 0x41	OutputID [8 bits] 1 byte	UniqueID [8 bits] 1 byte	Output Data N bytes	CS [8bits] 1 byte
Device number	Count of bytes without IMEI,LEN and CS	**OutputControl**	0x00 - OUT1 0x01 - OUT2 0x02 - OUT3 0x0A - Virtual signal S1 0x0B - Virtual signal S2 0x1E - Data over RS232	Unique ID generated y server each time sending output control service	Output data Structures	Sum of all packet bytes except CS

FROM MDL to SRV

IMEI [64 bits] 8 bytes	LEN [16 bits] 2 bytes	Service ID [8 bits] 1 byte 0xC1	OutputID [8 bits] 1 byte	CS [8bits] 1 byte
Device number	Count of bytes without IMEI,LEN and CS	**OutputControlAck**	0x00 - OUT1 0x01 - OUT2 0x02 - OUT3	Sum of all packet bytes except CS

Examples

RAW DATA PACKET (hex)
B0, 8, 48, D9, 4, 41, 1, 0, 23, 0, A5, E4, 20, A7, B7, E3, 54, 3, 80, 48, 0, F, 7C, AB, 41, 9B, F8, 27, 42, 0, 27, 59, E, 1, 0, 0, 0, 0, 80, 0, 7, 10, 50, 6C, 1, A1

PARSING
B0, 8, 48, D9, 4, 41, 1, 0, (imei)
23, 0, (len)
A5, (AsynckStack)
E4, (Confimration Key)
20, (len)
A7, B7, E3, 54, (Time and strucutre type)

3, 80, (Mask1)
48, 0, (Mask2)

F, 7C, AB, 41, 9B, F8, 27, 42, 0, 27, 59, E, 1, 0, 0, 0, 0, (*Coord. group1*)

80, 0, (*Digital inputs status*)

7, (*J1939 Fuel level*)
10, 50, 6C, 1, (J1939 *Total vehicle distance*)

A1 (CS)

Output 1 control example (1 second pulse to ground):

B0, 8, 48, D9, 4, 41, 1, 0,

06, 00, (len)

41, (Output control service)

00, (OUT1)

cc, (Unique ID - non repeatable, on each output cotrol new value - maybe always incrementing)

01, (Form Id - pulse type)

64, 00, (1 sec long pulse)

xx CS

Output 2 control example (5 second pulse to ground):

B0, 8, 48, D9, 4, 41, 1, 0,

06, 00, (len)

41, (Output control service)

01, (OUT1)

cc, (Unique ID - non repeatable, on each output cotrol new value - maybe always incrementing)

01, (Form Id - pulse type)

F4, 01, (5 sec long pulse)

xx CS

Static signal S1 ON control example :

B0, 8, 48, D9, 4, 41, 1, 0,

05, 00, (len)

41, (Output control service)

0A, (S1)

cc, (Unique ID - non repeatable, on each output cotrol new value - maybe always incrementing)

00, (FormID padByte)

55, (ON)

xx CS

Static signal S1 OFF control example :

B0, 8, 48, D9, 4, 41, 1, 0,

05, 00, (len)

41, (Output control service)

0A, (S1)

cc, (Unique ID - non repeatable, on each output cotrol new value - maybe always incrementing)

00, (FormID padByte)

00, (OFF)

xx CS

packed arrived through UDP or TCP
"085a629c834601008600a59b41c760ed5c0bc0d39204201909de41f95a5742001c72e400000000008000ed030101010903570b00300000004011
0583040d001200001f0c1a03c605004d690b0000416766ed5c0bc0d39204201909de41f95a5742001b72e400000000008000ee030101010903570
b00300000004011 0583040d001200001f0c1a03c605004d690b0000a0"

Packet parsing

08 5a 62 9c 83 46 01 00 (IMEI 0x146839c625a08 = 359006055062024)

86 00 (packet len 0x86 = 134 bytes)
a5 (Service FlashStack)
9b (Service 0xA5 confirmation key)

###(Start of structure #1 in packet)
41 (structure len 65 bytes)
c7 60 ed 5c (DT=7 Time=0x5ced60c*0x02+0x47798280=0x53172E98=1394028184= "GMT: Wed, 05 Mar 2014 14:03:04 GMT")
##(start of parsing detected DT=7 structure Masks)
0b c0 (Mask1 0xC00B = 0b1100000000001011)
d3 92 (Mask2,becasue most significant bit in Mask1 set, Mask2 present 0x92d3 = 1001001011010011)
04 20 (Mask3,becasue most significant bit in Mask2 set, Mask3 present 0x2004 = 0010000000000100)
##(End od Mask, because most significant byte is not set in last mask)
##MASK1 BEGIN
(Mask1 bit0=1 -> "Coord. group1" 17 bytes)
19 09 de 41 f9 5a 57 42 00 1c 72 e4 00 00 00 00 00
(Mask1 bit1=1 -> "Digital inputs status" 2 bytes)
80 00
(Mask1 bit3=1 -> "ADC2" 2 bytes)
ed 03
(Mask1 bit14=1 -> "GSM Info Group" 9 bytes)

01 01 01 09 03 57 0b 00 30
##MASK1 END
##MASK2 BEGIN
(Mask2 bit0=1 -> "J1939 Wheel speed" 2 bytes)
00 00
(Mask2 bit1=1 -> "J1939 Acceleration pedal position" 1 bytes)
00
(Mask2 bit4=1 -> "J1939 Engine Speed" 2 bytes)
40 11
(Mask2 bit6=1 -> "J1939 Total vehicle distance" 4 bytes)
05 83 04 0d
(Mask2 bit7=1 -> "1939 Engine Coolant temperature" 1 bytes)
00
(Mask2 bit9=1 -> "1939 J1939 Engine load" 1 bytes)
12
(Mask2 bit12=1 -> "J1939 Ambient Air Temperature" 2 bytes)
00 00
##MASK2 END
##MASK3 BEGIN
(Mask3 bit2=1 -> "J1939 Axle weight" 3 bytes)
1f 0c 1a
(Mask3 bit13=1 -> "J1708 Group1" 9 bytes)
03 c6 05 00 4d 69 0b 00 00
##MASK3 END
###(End of structure #1 in packet)

###(Start of structure #2 in packet)
41 (structure len 65 bytes)
67 66 ed 5c (DT=7 Time=0x5ced666*0x02+0x47798280=0x53172F4C=1394028364= "GMT: Wed, 05 Mar 2014 14:06:04 GMT")

0b c0 (Mask1 0xC00B = 0b1100000000001011)
d3 92 (Mask2,becasue most significant bit in Mask1 set Mask2 present 0x92d3 = 1001001011010011)
04 20 (Mask3,becasue most significant bit in Mask2 set Mask3 present 0x2004 = 0010000000000100)
##(End od Mask, because most significant byte is not set in last mask)
##MASK1 BEGIN
(Mask1 bit0=1 -> "Coord. group1" 17 bytes)
19 09 de 41 f9 5a 57 42 00 1b 72 e4 00 00 00 00 00
(Mask1 bit1=1 -> "Digital inputs status" 2 bytes)
80 00
(Mask1 bit3=1 -> "ADC2" 2 bytes)
ee 03
(Mask1 bit14=1 -> "GSM Info Group" 9 bytes)
01 01 01 09 03 57 0b 00 30
##MASK1 END
##MASK2 BEGIN
(Mask2 bit0=1 -> "J1939 Wheel speed" 2 bytes)
00 00
(Mask2 bit1=1 -> "J1939 Acceleration pedal position" 1 bytes)
00
(Mask2 bit4=1 -> "J1939 Engine Speed" 2 bytes)
40 11
(Mask2 bit6=1 -> "J1939 Total vehicle distance" 4 bytes)
05 83 04 0d
(Mask2 bit7=1 -> "1939 Engine Coolant temperature" 1 bytes)
00
(Mask2 bit9=1 -> "1939 J1939 Engine load" 1 bytes)
12
(Mask2 bit12=1 -> "J1939 Ambient Air Temperature" 2 bytes)
00 00
##MASK2 END

##MASK3 BEGIN
(Mask3 bit2=1 -> "J1939 Axle weight" 3 bytes)
1f 0c 1a
(Mask3 bit13=1 -> "J1708 Group1" 9 bytes)
03 c6 05 00 4d 69 0b 00 00
##MASK3 END
###(End of structure #2 in packet)
--
(CS)
a0