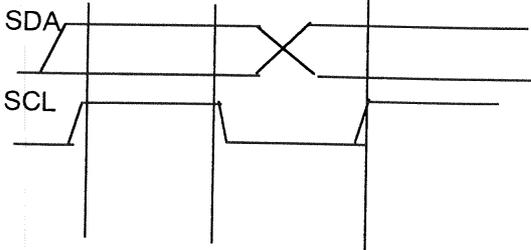


Trouble Example of SMBus smart battery

Clock stretch

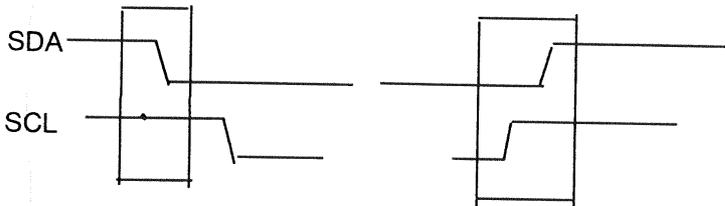
Host PC can't communicate with BP when it stretch clock line.

SMBus



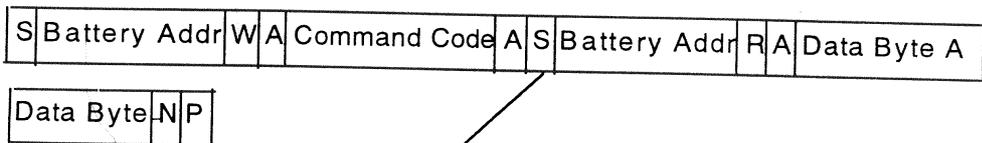
while clock is High , data line stable (data valid)

while clock Low is Low , change of data allowed



start and stop condition are made while clock line is high to make falling edge and rising edge

SMBus Read Word Protocol



BP(Slave) stretch clock line to prepare reply data , but HOST PC(Master)make start condition in spite of clock is stretched

can't charge

can't charge while HOST PC is power off or go into suspend mode

Smart Battery can charge while it communicate with HOST PC
This is in order to make battery safety while charging.

This case is that when HOST PC is power off or go into suspend mode, KBC go into the same status.

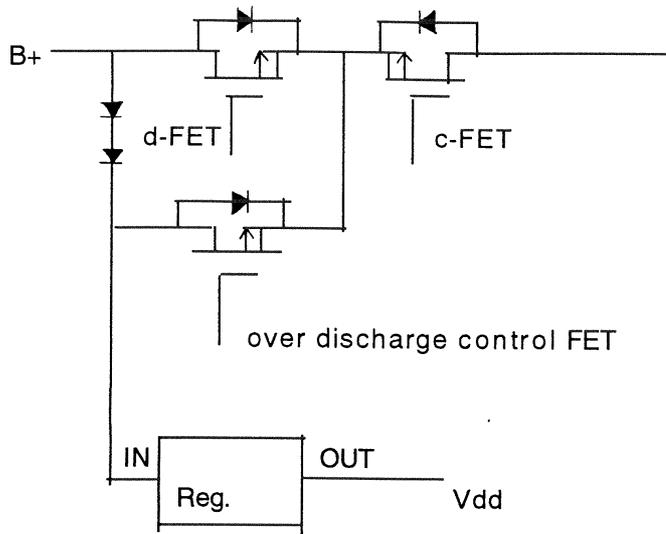
Communication between PC and SB is missing , then SB stop to charge.

On the other hand , there are big demand to charge by exclusive charger witch haven't SMbus..

How to answer for these request?

can't wake up from Low Battery Status

Low Battery can't wake up by Smart Charger



Smart charger need both command charge current(0x14) and charge voltage(0x15) from SB
But SB is shut down and can't issue both command.

Smart Battery Master Function

System freeze when SB send command as a Master

Charging Current (0x14)

Changing Voltage (0x15)

Alarm Warning (0x16)

these functions are permitted as master function from SB

This case , HOST side use polling style .

So it can't receive command from SB.

Proposal

0x14 and 0x15 Master function can be inhibited by Battery Mode(0x03)from HOST PC

How about Alarm function also can be inhibited by 0x03 command using a reserved bit?

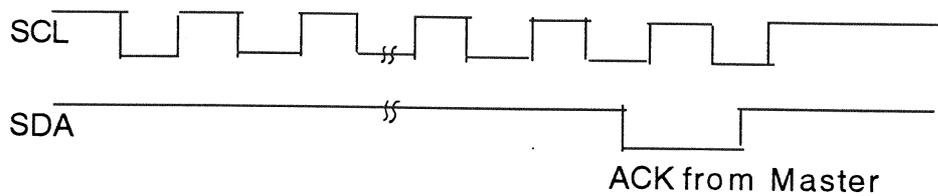
Data Format confusion

Master Device can't decide 0xFFFF data is true or not.

SBD define fully 16bits binary data from 0 to 65536 ($2^{16}-1$)

While SB do not response the command from master device, it start to get data synchronizing a clock.

Finally , it get 0xFFFF data.



Proposal

inhibit 0xFFFF format in response data