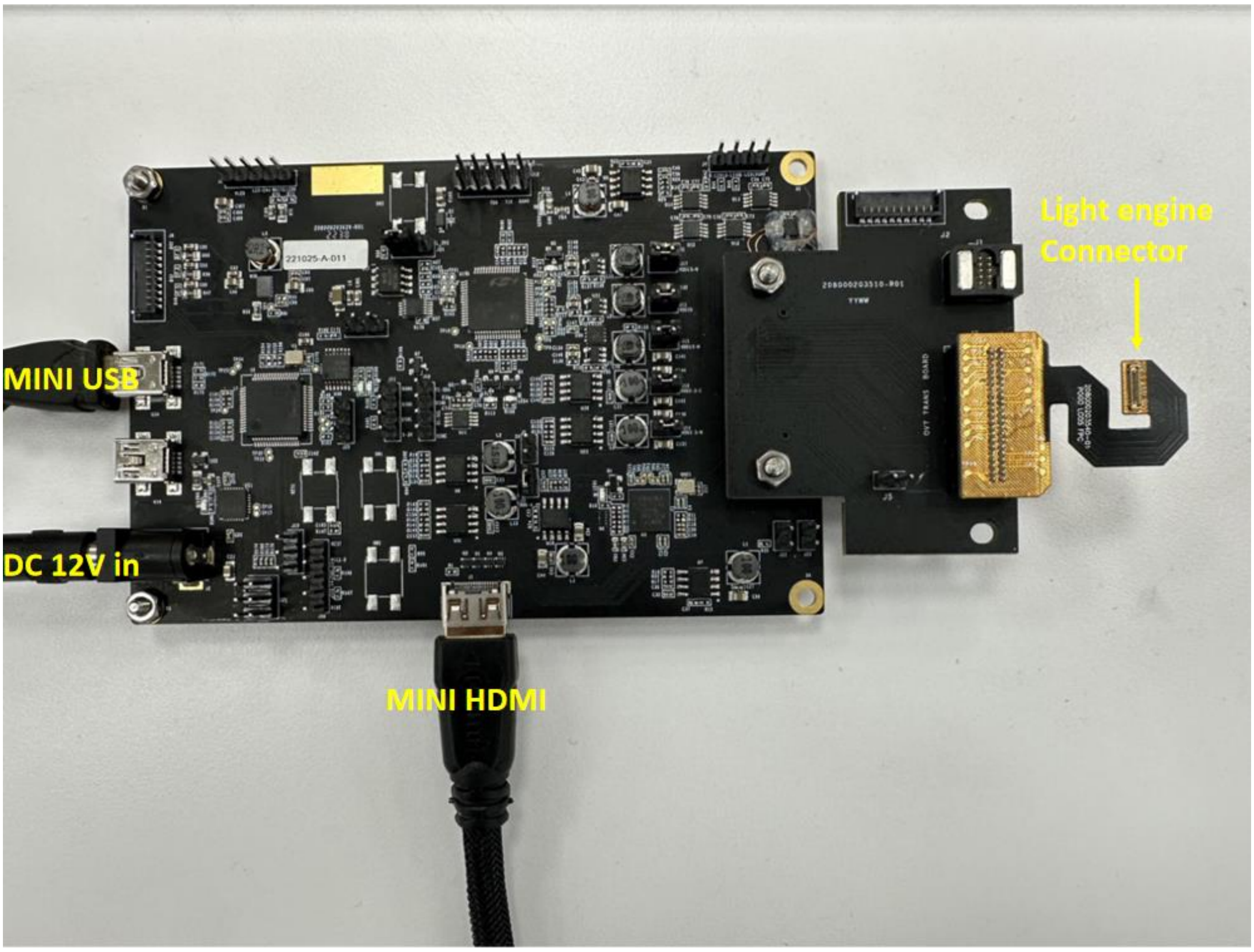
# testBD Communication protocol-v2.1

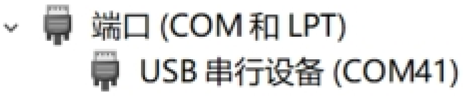
| Revision | Change Description |
|---|---|
| v0.1 | ● Create |
| v0.2 | ● Add commands：get e2-data，setd e2-data，setd lc-data，set l-white，set l-black，set l-red，set l-green，set l-blue，set l-grid，get temp-r，get temp-g，get temp-b，get temp-lc，get temp-db，set lc-cal |
| v0.3 | ● Add Part 3 transparent transmission<br>● Add commands：set d-pwmx，get version |
| v0.4 | ● Add Part 2.1-6 note<br>● Change Part 3.1 eeprom device address 0x40->0xA0 |
| v0.5 | ● Change ″set r=0\r\n″ is led-red-disable<br>″set r=1\r\n″ is led-red-enable<br>● Change ″set g=0\r\n″ is led-green-disable<br>″set g=1\r\n″ is led-green-enable<br>● Change ″set b=0\r\n″ is led-blue-disable<br>″set b=1\r\n″ is led-blue-enable |
| v0.6 | ● Add picture in Part1.1<br>● Add commads：get chip-id， set temp-bit，set t-ron，set t-gon，set t-bon，set t-blk<br>● Change command：set d-pwmx |
| v0.7 | ● Add Part 2.1-6.b.iii note |
| v0.8 | ● Add commands：get ri，get gi，get bi，get d-pwmx，set l-grv，set l-grh，set l-barv，set l-barh，set mode，set r-l，set g-l，set b-l，set a-l<br>● Change Part 2.1-6.b note<br>● Delete Part 2.1-6.b.iii note |
| v0.9 | ● Add commands：set l-gridx，set l-mipi，set lc-init |
| v1.0 | ● Add commands：set lc-cali，set lc-lm，set save-v，set temp-am，setd lc-id，get lc-id，get temp-am，get save-v |
| v1.1 | ● Add commands：set lc-x0，set lc-y0，set lc-xyen |
| v1.2 | ● Change command：set l-grid，set l-gridx |
| v1.3 | ● Add commands：set lc-lowc，get l-gpio0 |
| v1.4 | ● Change command：set l-grid |
| v1.5 | ● Add command：get anf-name |
| v1.6 | ● Add commands：get lcos-id，get pmic-otp，set ah |
| v1.7 | ● Change command：set e32，set e51 |
| v1.8 | ● Change command：setd lc-id，get lc-id |
| v1.9 | ● Add commands：get rv-ro，get gv-ro，get bv-ro |
| v2.0 | ● Add commands：get e2-dev，set e2-dev |
| v2.1 | ● Add commands：get ri-ad，get gi-ad，get bi-ad |

# 1 Prepare

## 1.1 Connect （U34）usb of testBD to computer



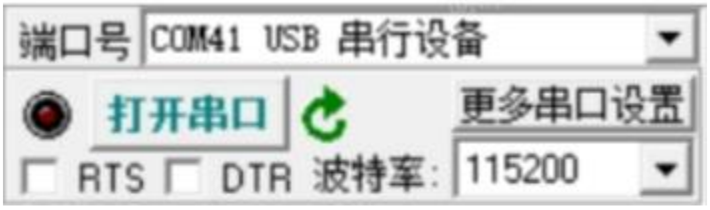
### 1.2 Find USB device from **Device Manager**， for example：



端口 (COM和 LPT)

USB 串行设备 (COM41)

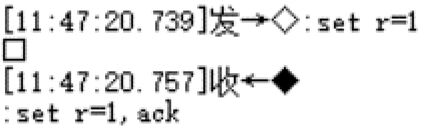
### 1.3 Open Serial port tool and select USB device， for example：



端口号 COM41 USB 串行设备

□ RTS □ DTR 波特率: 115200

### 1.4 Now，you can send cmd and receive info from testBD，for example：



```
[11:47:20.739]发→◇:set r=1
□
[11:47:20.757]收←◆
:set r=1, ack
```

# 2 Protocol Format

## 2.1 Format

| Protocol head | Left value | Operate | Right value | Protocol end |
|---|---|---|---|---|

1. Protocol head:

   a) “:set ”， 4Btyte：‘:’，‘s’，‘e’，‘t’，‘ ’（space:0x20）

   b) “:get ”， 4Btyte：‘:’，‘g’，‘e’，‘t’，‘ ’（space:0x20）

   c) “:setd ”， 5Btyte：‘:’，‘s’，‘e’， ‘t’， ‘d’，‘ ’（space:0x20）

2. Left value: 1~8Byte（Refer to Command table）

3. Option: 1Byte，‘=’

4. Right value:　1~8Byte（Refer to Command table）

5. Protocol end: 2Byte：‘\r’（0x0D），‘\n’（0x0A）

6. Note：

    a) Do not miss out string "\r\n"，when sending command.

    b) LED driver switch：

        i. If you want to use testBD drive LED light，you must send the follow commands first：":set mode=1\r\n" or ":set mode=3\r\n".

            1. mode=1，using lcos RGB-enable drive LED，and short circuit J26(pin2 and pin3)

            2. mode=3，using stm32 RGB-enable drive LED，and open circuit J26(pin2 and pin3)

        ii. If you want to use DDB drive LED light，you must send the follow commands first：":set mode=2\r\n" or ":set mode=4\r\n".

            1. mode=2，using lcos RGB-enable drive LED，and short circuit J26(pin2 and pin3)

            2. mode=4，using stm32 RGB-enable drive LED，and open circuit J26(pin2 and pin3)

## 2.2 Command table1 for “set ”

| Left value | Right value | Description | Example |
|---|---|---|---|
| ri | 0~300 | led-R Currents,mA (Rsense=500mΩ) | Set Red 10mA，Green 20mA，Blue 30mA：<br>″:set ri=10:set gi=20:set bi=30:set ai=1\r\n″ |
| gi | 0~300 | led-G Currents,mA (Rsense=500mΩ) | |
| bi | 0~300 | led-B Currents,mA (Rsense=500mΩ) | |
| ai | 1 | enable ri,gi,bi setting | |
| rgbi | 0~300 | led-RGB Currents,mA (Rsense=500mΩ) | Set Red/Green/Blue 30mA：<br>″:set rgbi=30\r\n″ |
| rgbi-ad | 0~1023 | led-RGB LED Optical Gain | Set Red/Green/Blue min Gain：(Currents)<br>″:set rgbi-ad=0\r\n″<br>Set Red/Green/Blue max Gain：(Currents)<br>″:set rgbi-ad=1023\r\n″ |
| rv | 1.0~5.5 | led-R Voltage,V | Set Vled-Red 3.0V,Vled-Green 3.7V,Vled-Blue 3.6V,Vblank 3.3V<br>″:set rv=3.0:set gv=3.7:set bv=3.6:set vv=3.3:set av=1\r\n″ |
| gv | 1.0~5.5 | led-G Voltage,V | |
| bv | 1.0~5.5 | led-B Voltage,V | |
| vv | 1.0~5.5 | led-vblk Voltage,V | |
| av | 1 | enable rv,gv,bv,vv setting | |
| en-ld | 0~1 | 0：enable testBD-led-driver disable DDB-led-driver<br>1：disable testBD-led-driver enable DDB-led-driver | ″:set en-ld=0\r\n″ |
| en-lcos | 0~1 | 0：disable DDB-PMIC<br>1：enable DDB-PMIC | ″:set en-lcos=0\r\n″ |
| panel | 0~1 | 0：panel low<br>1：panel high | ″:set panel=0\r\n″ |
| iic-sw | 0~1 | 0：c8051<->testBD,Stm32<->DDB<br>1：Stm32<->testBD,C8051<->NULL | ″:set iic-sw=0\r\n″ |
| e51 | 1~4 | Change NVRAM for 8051 on baseboard<br>1：eeprom test<br>write 0xaa full,Readback check<br>write 0xff full,Readback check<br>2：write op02220 default FW<br>3：write op03010 default FW-60hz<br>4：write op03010 default FW-30hz | ″:set iic-sw=1\r\n″<br>″:set e51=1\r\n″<br>″:set e51=2\r\n″<br>″:set e51=3\r\n″ |
| e32 | 1~4 | Change NVRAM for stm32 on DDB<br>1：eeprom test<br>write 0xaa full,Readback check<br>write 0xff full,Readback check<br>2：write op02220 default FW<br>3：write op03010 default FW-60hz | ″:set iic-sw=0\r\n″<br>″:set e32=1\r\n″<br>″:set e32=2\r\n″<br>″:set e32=3\r\n″ |

| | | | |
|---|---|---|---|
| | | 4：write op03010 default FW-30hz | |
| exx-r | 1~x | 1~x：read x Byte from current eeprom from address 0x0 | ″:set exx-r=16\r\n″ |
| r | 0~1 | 0：led-red-disable<br>1：led-red-enable<br>Control by STM32 pwm | ″:set r=0\r\n″<br>″:set r=1\r\n″ |
| g | 0~1 | 0：led-green-disable<br>1：led-green-enable<br>Control by STM32 pwm | ″:set g=0\r\n″<br>″:set g=1\r\n″ |
| b | 0~1 | 0：led-blue-disable<br>1：led-blue-enable<br>Control by STM32 pwm | ″:set b=0\r\n″<br>″:set b=1\r\n″ |
| a | 0~1 | 0：led-rgb-disable<br>1：led-rgb-enable<br>Control by STM32 pwm | ″:set a=0\r\n″<br>″:set a=1\r\n″ |
| r-l | 0~1 | 0：led-red-disable<br>1：led-red-enable<br>Control by writing the lcos register | ″:set r-l=0\r\n″<br>″:set r-l=1\r\n″ |
| g-l | 0~1 | 0：led-green-disable<br>1：led-green-enable<br>Control by writing the lcos register | ″:set g-l=0\r\n″<br>″:set g-l=1\r\n″ |
| b-l | 0~1 | 0：led-blue-disable<br>1：led-blue-enable<br>Control by writing the lcos register | ″:set b-l=0\r\n″<br>″:set b-l=1\r\n″ |
| a-l | 0~1 | 0：led-rgb-disable<br>1：led-rgb-enable<br>Control by writing the lcos register | ″:set a-l=0\r\n″<br>″:set a-l=1\r\n″ |
| mode | 0~4 | 0：default，8051 drive testBD<br>1：stm32 drive testBD with lcos-rgb-enable<br>2：stm32 drive DDB with lcos-rgb-enable<br>3：stm32 drive testBD with stm32-rgb-enable<br>4：stm32 drive DDB with stm32-rgb-enable | ″:set mode=1\r\n″ |
| sense | X | X is Rsense value(mΩ) | ″:set sense=500\r\n″ |
| check | 0~1 | 0：disable info upload<br>1：enable info upload | ″:set check=0\r\n″<br>″:set check=1\r\n″ |

| | | | |
|---|---|---|---|
| l-white | 1 | set white pattern | ":set l-white=1\r\n" |
| l-black | 1 | set black pattern | ":set l-black=1\r\n" |
| l-red | 1 | set red pattern | ":set l-red=1\r\n" |
| l-green | 1 | set green pattern | ":set l-green=1\r\n" |
| l-blue | 1 | set blue pattern | ":set l-blue=1\r\n" |
| l-grid | 0~8 | 0：set grid pattern0<br>1：set grid pattern1<br>2：set CheckerBoard pattern<br>3~7：set grid pattern3~7<br>8：set crosshair | ":set l-grid=1\r\n" |
| l-grv | 1 | set vertical gray ramp pattern | ":set l-grv=1\r\n" |
| l-grh | 1 | set horizontal gray ramp pattern | ":set l-grh=1\r\n" |
| l-barv | 1 | set vertical color bar pattern | ":set l-barv=1\r\n" |
| l-barh | 1 | set horizontal color bar pattern | ":set l-barh=1\r\n" |
| lc-cal | 1~21 | 1~21：index of Calibration Data in NVRAM<br>Set the current value of the current index(1~21) | ":set lc-cal=1\r\n" |
| lc-cali | 1~21 | 1~21：index of Calibration Data in NVRAM<br>Set testBD only. What is the current index(1~21).<br>It needs to be used with other commands<br>Refer command"get save-v" | ":set lc-cali=1\r\n" |
| lc-lm | x | Save temporary lumen value.<br>It needs to be used with other commands<br>Refer command"set save-v=1" | ":set lc-lm=1.5\r\n" |
| lc-x0 | 0~647 | Start coordinate of image(lcos)：x | ":set lc-x0=24:set lc-y0=24:set lc-xyen=1\r\n" |
| lc-y0 | 0~647 | Start coordinate of image(lcos)：y | |
| lc-xyen | 1 | Enable x&y coordinate setting | |

| | | | |
|---|---|---|---|
| temp-am | x | Save ambient temperature | ″:set temp-am=36.5\r\n″ |
| save-v | 1~21 | Save the current lumen, current, voltage and temperature to group x(1~21)<br>It needs to be used with other commands | ″:set lc-lm=1.5:set save-v=1\r\n″ |
| lc-init | 1 | Initialize lcos manually | ″:set lc-init=1\r\n″ |
| l-gridx | 3~255 | set grid pattern type<br>default value：40 | If you want to set grid 4x4, and the current resolution is 600x600, then sending commands(600/4=150)：<br>″:set l-gridx=150:set l-grid=2\r\n″ |
| l-mipi | 1 | set HDMI mode from PATTERN mode | ″:set l-mipi=1\r\n″ |
| d-pwmx | 0~2 | 0：r/g/b-duty ratio is 3:5:2（90Hz）<br>1：r/g/b-duty ratio is 3:3:3（90Hz）<br>2：custom，reference：<br>set t-ron<br>set t-gon<br>set t-bon<br>set t-blk | ″:set d-pwmx=1\r\n″ |
| temp-bit | 10,12 | 10（default）：set resolution to 10 bits for Temperature sensor（DDB/R/G/B）, 0.25° C per step<br>12：set resolution to 12 bits for Temperature sensor（DDB/R/G/B）, 0.0625° C per step | ″:set temp-bit=12\r\n″ |
| lc-lowc | 0~1 | 0：set normal current mode for LED-driver<br>1：set low current mode for LED-driver | ″:set lc-lowc=1\r\n″ |
| ah | 0~1 | 0：Auto Voltage Adjust disable for LED driver<br>1：Auto Voltage Adjust enable for LED driver | ″:set ah=1\r\n″ |
| e2-dev | 0~255 | Set nvram device address | ″:set e2-dev=80\r\n″ |
| t-ron | 0~65535 | Set the enabling time of the red light | ″:set t-ron=3180:set t-gon=5110:set t-bon=1930:set t-blk=300:set d-pwmx=2\r\n″<br>Refer to Figure 1 |
| t-gon | 0~65535 | Set the enabling time of the green light | |

| | | | |
|---|---|---|---|
| t-bon | 0~65535 | Set the enabling time of the blue light | Refer to Table 1 |
| t-blk | 0~65535 | Set the enabling time of the blank | |

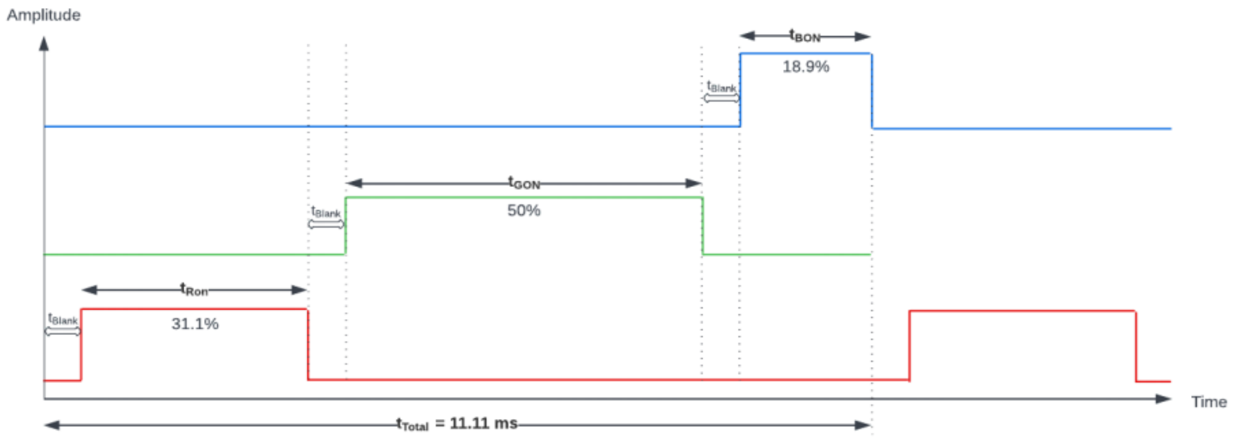


Figure 1: R/G/B Enable signal timings driving the inputs of the LED Driver

| Parameter | Spec | Units |
|---|---|---|
| *tBlank* | 300 | µs |
| *tRon* | 3.18 | ms |
| *tGon* | 5.11 | |
| *tBon* | 1.93 | |

Table 1: LED Timings used for functional testing of the LED driver

## 2.3 Command table2 for “set ”-reply

| | | | |
|---|---|---|---|
| recv | / | fomat: "\n(cmd),(result)\r\n" | If send one cmd:<br>send: ″:set check=0\r\n″<br>recv: ok/fail/iic-fail<br>″\n:set check=0,ack\r\n″<br>″\n:set check=0,error\r\n″<br>″\n:set check=0,iic-error\r\n″<br><br>If send two or more cmd,and its ok<br>send: ″:set ri=10:set gi=20:set bi=30:set ai=1\r\n″<br>It will receive info Sequential:<br>″\n:set ai=1,ack\r\n″<br>″\n:set bi=30,ack\r\n″<br>″\n:set gi=20,ack\r\n″<br>″\n:set ri=10,ack\r\n″ |

## 2.4 Command table3 for “get ”

| Left value | Right value | Description | Example |
|---|---|---|---|
| rv-ro | / | Get led-R Voltage from led driver,V | Get led-red Voltage(V)：<br>send：″:get rv-ro\r\n″<br>recv：″3.041\r\n″ |
| gv-ro | / | Get led-G Voltage from led driver,V | |
| bv-ro | / | Get led-B Voltage from led driver,V | |
| rv | / | Get led-R Voltage from stm32-ADC,V | Get led-red Voltage(V)：<br>send：″:get rv\r\n″<br>recv：″3.001,0.842\r\n″ (+,-) |
| gv | / | Get led-G Voltage from stm32-ADC,V | |
| bv | / | Get led-B Voltage from stm32-ADC,V | |
| ri | / | Get led-R Currents,mA | Get led-R Currents(mA)：<br>send：″:get ri\r\n″<br>recv：″100.00\r\n″ |
| gi | / | Get led-G Currents,mA | |
| bi | / | Get led-B Currents,mA | |
| ri-ad | / | Get led-R Currents,0~1023 | Get led-R Currents(dac value)：<br>send：″:get ri-ad\r\n″<br>recv：″1023\r\n″ |
| gi-ad | / | Get led-R Currents,0~1023 | |
| bi-ad | / | Get led-R Currents,0~1023 | |
| e2-data | / | Get eeprom data at address x,<br>Return hex value by string,<br>Max:4byte | Get eeprom data at address 0x0011<br>send：<br>″:set e2-addr=17\r\n″<br>recv：<br>″\n:set e2-addr=17,ack\r\n″<br>send：<br>″:get e2-data\r\n″<br>recv：<br>″11 22 ff ee″ |
| temp-r | / | Get led-R Temperature,℃ | Get led-red Temperature (℃)：<br>send：″:get temp-r\r\n″<br>recv：″25.75\r\n″ |
| temp-g | / | Get led-G Temperature,℃ | |
| temp-b | / | Get led-B Temperature,℃ | |
| temp-db | / | Get DDB Temperature,℃ | |
| temp-lc | / | Get lcos Temperature,℃ | |
| temp-am | / | Read ambient Temperature,℃ | |
| version | / | Get Firmware version | send：″:get version\r\n″<br>recv：″0.0.3\r\n″ |
| chip-id | / | Get PMIC CID | send：″:get chip-id\r\n″<br>recv：″0x09\r\n″ |
| d-pwmx | / | Get the enabling time of the red/green/blue light | send：″:get d-pwmx\r\n″<br>recv：″ 3180,5110,1930,300\r\n″ |

| | | | |
|---|---|---|---|
| | | | The enabling time(μs) of the red/green/blue light is：<br>3180,5100,1930<br>And duty cycle of the red/green/blue light is：<br>3180/(3180+5110+1930+300*3)*100%<br>5110/(3180+5110+1930+300*3)*100%<br>1930/(3180+5110+1930+300*3)*100% |
| lc-id | / | Get DDB SN | send：″:get lc-id\r\n″<br>recv：″1122ffee\r\n″ |
| save-v | / | Get the current index data<br>It needs to be used with other commands | ″:set lc-cali=1:get save-v″<br>recv：<br>″36.50,112233ee,1.50,2.5,2.5,2.5,2.5,2.5,2.5,2.5,2.5,2.5,2.5,0.00,0.00,0.00\r\n″<br>CORRESPOND：<br>AmbienTemp,ID,Lm,Ired,Igreen,Iblue,Vfred,Vfgreen,Vfblue,TempDDB,TempRedBlue,TempGreen,TempLcos,IredSet,IgreenSet,IblueSet |
| l-gpio0 | / | Get low/normal current mode:<br>1：normal current mode for LED-driver<br>0：low current mode for LED-driver | send：″:get l-gpio0\r\n″<br>recv：″1\r\n″ |
| anf-name | / | Get anf name in NVRAM of DDB | send：″:get anf-name\r\n″<br>recv：<br>″OP03010_600_9cf_60hz_v0\r\n″ |
| lcos-id | / | Get lcos id from lcos-register | send：″:get lcos-id\r\n″<br>recv：″0x03 0xa2 0x0a\r\n″ |
| pmic-otp | / | Get PMIC OTP | send：″:get pmic-otp\r\n″<br>recv：″3c 00 00 ff f1 00 09 00 70 29 8c 28 28 2b 34 2a\r\n″ |
| e2-dev | / | Get nvram device address | send：″:get e2-dev\r\n″<br>recv：″0x50\r\n″ |

## 2.5 Command table3 for “setd ”

| Left value | Right value | Description | Example |
|---|---|---|---|
| e2-data | / | Hex value by string,<br>Max:8byte | Wirte 0x11 0x22 0xff 0xee to eeprom/lcos at address 0x0011：<br>send：<br>″:set e2-addr=17\r\n″<br>Or<br>″:set lc-addr=17\r\n″<br>recv：<br>″\n:set e2-addr=17,ack\r\n″<br>Or<br>″\n:set lc-addr=17,ack\r\n″<br>send：<br>″:setd e2-data=1122ffee\r\n″<br>Or<br>″:setd lc-data=1122ffee\r\n″<br>recv：<br>″\n:setd e2-data=1122ffee,ack\r\n ″<br>Or<br>″\n:setd lc-data=1122ffee,ack\r\n ″ |
| lc-data | / | Hex value by string,<br>Max:8byte | |
| lc-id | / | Hex value by string,<br>Max:8byte | Set DDB SN：″1122ffee″<br>send：<br>″:setd lc-id=1122ffee\r\n″<br>recv：<br>″\n:setd lc-id=1122ffee,ack\r\n ″ |

# 3 Transparent Transmission

## 3.1 TestBD support transparent transmission，from PC to TestBD：

For Example：

Write or read eeprom，eeprom device address is 0xA0

- Write 0x11 0x22 0x33 0x44 to eeprom at reg-address 0x0012：

  Send：0xA0 0x00 0x12 0x11 0x22 0x33 0x44

Recv(ok)：0x00

Recv(fail)：0xff

- Read 4 byte from eerpom at reg-address 0x0012：

Send：0xA1 0x00 0x12 0x04

Recv(ok)：0x11 0x22 0x33 0x44

Recv(fail)：no-reply，3ms time out

# The End