

隐智科技 Caps 协议 (V3.0)

具体协议手册

一般是上位机主动向单片机发送命令，发送 ASCII 码（文本模式），串口波特率为 115200。每条命令都是以~开头，都是以#符结尾。每个指令发送完毕后需要间隔 80 毫秒才能发送下一条指令。

（一）、下位机在棋盘盘面有变化时，或者有按键被按下时，会主动发送消息给上位机。

1. ~SDA0000000XXXXXXXXXXXXXXXXX.....XXXXX#

~ 命令开头

SDA 标识符，表示发出的是全盘数据

361 个数据，代表位置 1 到位置 361 的棋盘状态。0 表示位置上无子，1 表示位置上黑子，2 表示位置上白子。

#表示命令结束符。

下位机检测到棋盘盘面有位置变化时，下位机会主动发送此命令。

2. ~WKY# 白方按钮单击，

3. ~WTK# 白方按钮长按

4. ~BKY# 黑方按钮单击

5. ~BTK# 黑方按钮长按

注意：检测按键的 4 个状态是下位机实时检测，检测到有事件发生就主动往上位机发送指令。

(二) 上位机主动发命令给下位机

1. \sim SHPxxx, rxxgxxbxxx, l#

此命令表示上位机要点亮棋盘上某个位置（1~361 固定为 3 位），r 等级为（0~255），g（0~255），b（0~255）的彩色 LED。

位置和等级必须固定为 3 位，因为下位机程序写死是固定的。位置是 001 到 361，等级大小为 0 到 255，等级为 0 表示熄灭。用这个指令点亮一个灯以后，此条指令会把自己点亮的上一个灯熄灭。指令执行成功会返回 \sim HCS#。后面的, l, 没有做任何处理。为预留。

执行成功返回指令 \sim TLS#

2. \sim STA#，表示上位机主动请求要求下位机发送一次全盘数据，下位机返回一个全盘数据状态（ \sim SDA111……#）。

3. \sim RGC#

此命令的含义是让下位机把 361 个三色 LED 灯全部关掉。同时此命令会把存储到单片机内要显示的彩色 LED 的数据全部清零。底层关闭所有彩色指示灯以后会返回指令 \sim ALC#。

4. \sim AWO#

此命令的含义是让蜂鸣器产生滴滴报警声。蜂鸣器滴滴两声后会自动关闭。主要应用场景就是下棋过程中棋盘棋子的实际情况跟盘面情况下棋子情况下不符等其他需要提醒对弈者的报警方式。

5. \sim LEDxx#

第一个 x 如果是 1 代表的是黑方亮灯，如果是 2 代表是白方亮灯。如果是其他数字，则不返回指令。

第二个 x 如果是 1，代表让行棋指示灯亮，如果是 0，代表是关闭黑白双方行棋指示灯。

如果发送的指令符合要求，下位机会返回指令 \sim LOS#

6. ~BOD09#

控制棋盘为 9 路棋盘，下位机返回指令~GBS09#，同时棋盘只会检测以天元为中心点的 9 路棋盘。会在 9 路棋盘的外围亮一圈灯。下位机返回的数据也是 81 个数据。上位机给下位机发指示灯亮灯的数据也是 81 位。发单个亮灯的指令的时候，是以小棋盘重新计算坐标。

7. ~BOD13#

控制棋盘为 13 路棋盘，下位机返回指令~GBS13#，其余参照 9 路小棋盘。

8. ~BOD15#

控制棋盘为 15 路棋盘，下位机返回指令~GBS15#，其余参照 9 路小棋盘。

9. ~BOD19#

控制棋盘为 19 路棋盘。下位机返回指令~GBS19#。同时需要发送~RGC#指令关闭小棋盘边界灯。

10. ~RGF#

表示关闭小棋盘内所有指示灯，同时不关闭小棋盘的边界灯。

11. ~SALXXXXXXXXX.....#

xxx 总共有 361 个，只能是 0、1 或者 2，该条指令可以点亮盘面上多个位置的 LED 灯，0 并不能够关闭。1 代表是绿色，2 代表是白。亮灯的位置由顺序决定。XXX 的个数必须是 361 个。亮度不可调。指示灯为高亮度。返回指令为~ALS#

12. ~SAMXXXXXXXXX.....#

xxx 总共有 361 个，只能是 0、1 或者 2，该条指令可以点亮盘面上多个位置的 LED 灯，0 并不能够关闭。1 代表是绿色，2 代表是白色。亮灯的位置由顺序决定。XXX 的个数必须是 361 个。亮度不可调。指示灯为中等亮度，返回指令为~ALS#。

13. ~SAWXXXXXXXXX.....#

xxx 总共有 361 个，只能是 0、1 或者 2，该条指令可以点亮盘面上多个位置的 LED 灯，0 并不能够关闭。1 代表是绿色，2 代表是白色。亮灯的

位置由顺序决定。XXX 的个数必须是 361 个。亮度不可调。指示灯为低等亮度，返回指令为~ALS#。

14. ~SARXXXXXX.....# xxx 总共有 361 个， 可以是 0、1、2、3、4、5、6、7、8、9，该条指令可以点亮盘面上多个位置的 LED 灯，亮灯的位置由顺序决定。XXX 的个数必须是 361 个。亮度不可调，返回指令为~ALS#。

0、表示关闭这个位置的指示灯

1、绿色

2、白色

3、亮度低一些的绿色

4、亮度低一些的白色

5、红色

6、粉色

7、粉白色

8、蓝色

9、米白色

具体以实际看到的颜色为准。

特别注意： 因为三色指示灯为耗能器件，如果红绿蓝都点到最亮的话，一颗灯珠消耗的电流为 45 毫安。如果点亮太多的灯会导致电源头供电不够，从而导致棋盘亏电重启或者直接关机。建议亮灯时上位机控制好亮灯的数量，如果有亮灯较多的情况，需要选择调用低亮度的指令。另外因为 11、12、13、14 指令比较长，发出指令后，需要等待返回指令，如果规定时间没有收到返回指令，需要重新发指令。

15. ~RLT# 棋盘亮灯显示一个对勾

16. ~RLW# 棋盘亮灯显示一把叉

17. ~RLO# 棋盘显示 一个“OK”

18. ~GVE# 获取版本底层程序的版本号，下位机会返回~VERXXX#，xxx 会是 100 以上的数字。目前版本 114，以后有更新就会增加数字值。

19.

**~HOTAxrrxxxgxxxbxxxBxxrrxxxgxxxbxxxCxxrrxxxgxxx
bxxxDxxrrxxxgxxxbxxxNx#**

~: 指令头

HOT: 标识符

A、B、C、D: 后面必须是三位数 大小是 1~361。 是点亮灯的位置。

r、g、b: 后面必须是三位数, 0~255 代表亮度等级, 等于 0 并不能关闭指示灯, 必须调用~RGC#指令, ~RGF#(小棋盘)指令关闭。r 后面表示红色 g 后面表示绿色 b 后面表示蓝色。

N 后面的叉叉, 只能是 1、2、3、4。 1 表示只亮前面第一个灯, 2、表示只亮前面两个灯, 3 表示只亮前面三个灯, 4 表示亮四个灯。

指令执行成功后返回~HSM#

20.

~TL0xxx, rxxxgxxxbxxx, X#

此命令表示上位机要点亮棋盘上某个位置的三色灯的同时, 控制行棋指示灯。

~TL0xxx, rxxxgxxxbxxx, X#指令最后那个大写的 X 如果是 1, 就点亮黑方行棋指示灯时关闭白方指示灯; 如果等于 2, 就是开启白方指示灯, 关闭黑方指示灯; 如果等于 3, 那就黑方行棋指示灯和白方行棋指示灯都亮; 如果是 0, 那就都关闭; 如果是其他数字整个指令无效。

TL0 后面的 xxx, 表示三色灯的某个位置 (1~361 固定为 3 位), r 后面的 xxx 为红色等级为 r (0~255), g 后面的 xxx 为绿色等级 (0~255), b 后面的 xxx 为蓝色等级 (0~255)。

位置和等级必须固定为 3 位, 因为下位机程序写死是固定的。位置是 1 到 361, 等级大小为 0~255, 等级为 0 表示熄灭。

执行成功返回指令~TLS#

21.

~CTS0#

~CTS1#

如果上层向底层发送~CTS1# 表示下位机在盘面有变化时可以主动往上位机发送棋盘状态数据。如果底层收到指令成功后，会返回指令 ~UDS1#，

如果上层向底层发送~CTS0# ，表示下位机在盘面有变化时不会主动往上位机发送棋盘状态数据，上位机想获取盘面状态只能发送~STA#，主动获取盘面。如果底层收到指令成功后，会返回指令~UDS0#

22.

~GLCxRxxxGxxxBxxx#

~: 命令头

GLC: 标示符

GLC 后的第一个 x 只能等于 1 或者 2

如果是 1 ，表示控制的是黑方的行棋指示灯

如果是 2 ，表示控制的是白方的行棋指示灯

R: 仅仅是分隔符 没有意义

Rxxx 后面的三个 xxx 只能是 0 到 255 包含 255，标示红色灯的亮度

Gxxx 后面的三个 xxx 只能是 0 到 255 包含 255，标示绿色灯的亮度

Bxxx 后面的三个 xxx 只能是 0 到 255 包含 255，标示蓝色灯的亮度

行棋指示灯点亮后不会自己熄灭，熄灭的方式有两种，第一种 ~LED10#或者~LED20# 黑白全灭。第二种 RGB 的数值全部为 0。前面是 1 就灭黑方行棋指示灯，是 2 就灭白方行棋指示灯。指令执行成功以后返回~LOS#。