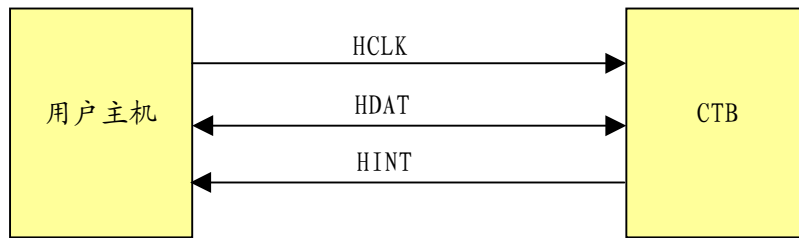


CTB 通讯总线应用注意

CTB 总线是龙珠科技应用于 OTG 系列数码板或其它模块中所使用的内部总线。采用 HCLK、HDAT、HINT 三条线作为通讯端。分为主方与从方两部分，其中用户的单片机为主方。所有的读写都是由主方发起的，主方可以使用普通的 IO 端口作为通讯端口，可以简单地将通讯函数移入程序之中。用户主机作为主动控制方。可随时写入控制指令给 CTB。两者之间通讯方向见下图：



其中 HCLK 为时钟同步信号，用户主机使用输出端口；HDAT 为数据信号，高位先出。当用户主机写指令时为输出端口，读指令时为输入端口；HINT 为 CTB 的中断申请端，用户主机在读取 CTB 的指令时必须先检测到 HINT 变为低才可执行相应的读指令。当发送或接收数据长度标志时，用户主机输出 HINT 为高，发送指令时用户主机输出 HINT 为低。

1. 主方写指令方式

在 CTB 写指令时 HCLK、HDAT、HINT 三个端口在用户主机都为输出端，当发送完毕，用户主机须检测最后一位为 ACK 应答信号，必须检测到 HDAT 为低。用户主机可随时写指令给 CTB。

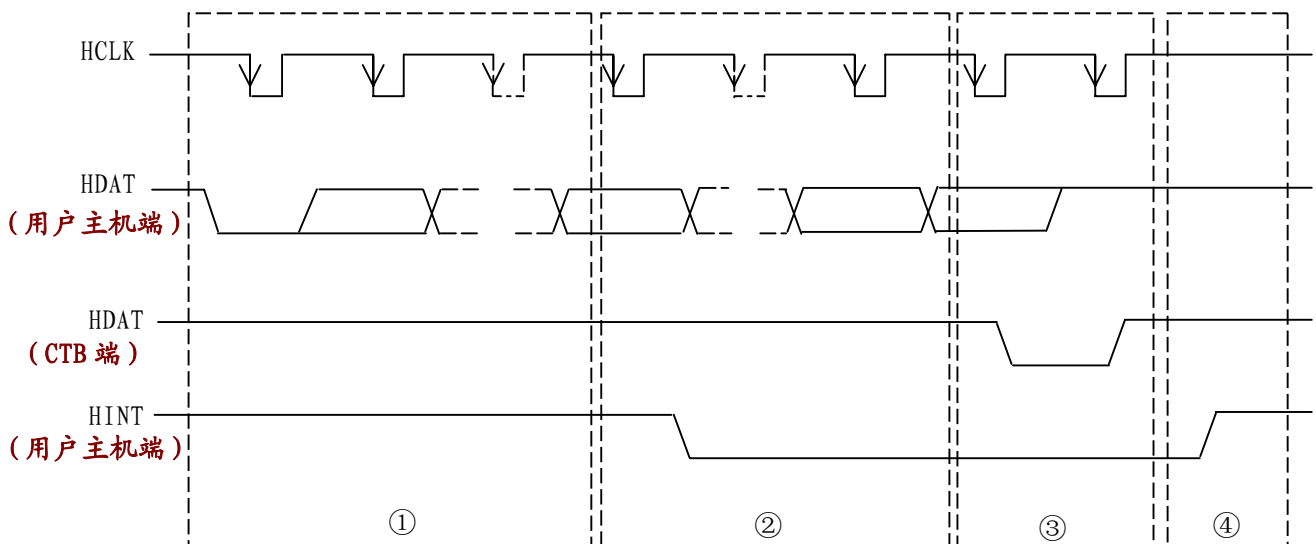


图 2. 用户主机写指令时序图



深圳市龙珠科技有限公司

Hard & Soft Technology Co., LTD.

<http://www.HSAV.com>

地址:深圳市西乡龙珠路 99 号 2 楼
电话/传真:0755-27951479 27950879

技术支持: support@HSAV.com
业务联系: sales@HSAV.com

Hsavn305.pdf
2004 年 07 月 27 日



步骤①，为用户主机写指令共发送字节数的长度，有效范围为 2-18。(2 表示在步骤①之后，会再跟随有 1 个字节的指令/数据)。所有传输的数据都为高位先出，先变化 HDAT 数据位，再变低 HCLK 位，升高 HCLK 后，1 位数据传输完毕。

- ① 步骤②用户主机变低 HINT，表示跟随发送的是指令/数据字节。长度为步骤①发送的数值减 1。字节是以 8 位表示的，故必须发送共发送步骤①x8 位的数据。
- ② 步骤③用户主机升高 HDAT 及将 HDAT 置为输入端口，在 HCLK 下降后，CTB 会变低 HDAT，表示接收正确，用户可在 HCLK 的上升沿读取 HDAT 的值作为 ACK 应答位。用户主机可根据 ACK 应答位判断刚才的传输是否正确，出错则需重新发送。CTB 在 HCLK 下降后升高 HDAT。
- ③ 步骤④，用户主机升高 HINT 后，整个写指令结束。CTB 总线进入空闲状态。用户主机的 HCLK、HDAT 为高输出，HINT 为输入。

2. 从方读指令方式

CTB 读指令时，必须先检测到 HINT 变低方可进行。否则可能出现不可预测的错误。

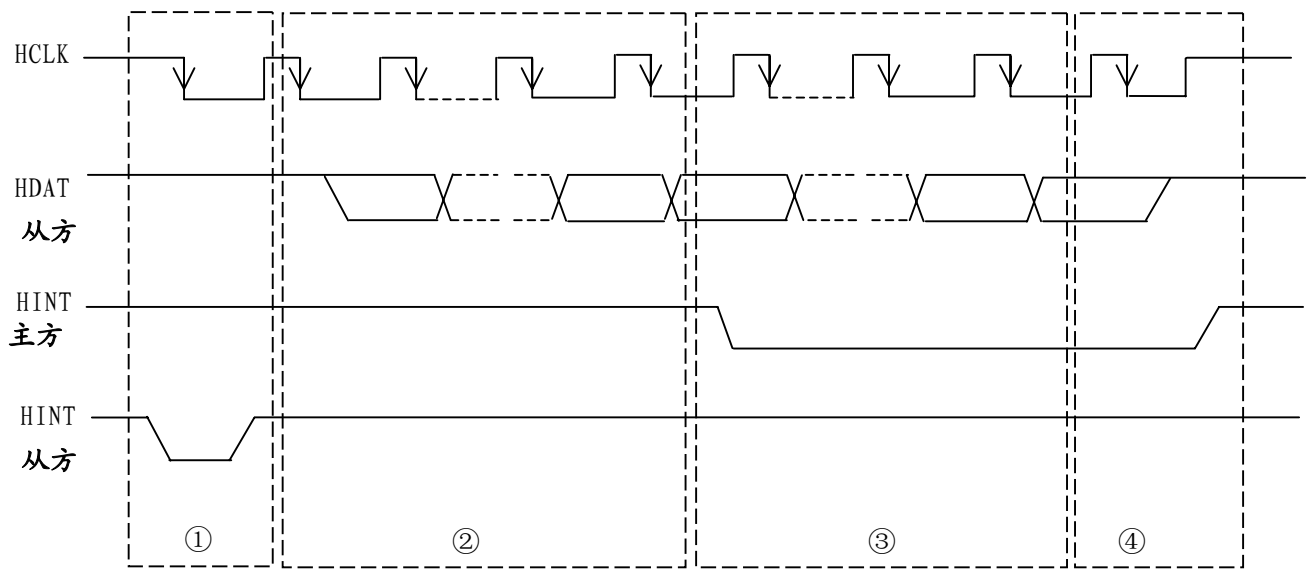


图 3. 用户主机读指令时序图

- ① 步骤①，用户主机读/写控制位，在主方读指令时为 HDAT 高，HINT 只是从方变高，主方不需理会。
- ② 步骤②，在用户主机 HCLK 变低之后，CTB 会变化 HDAT 的值，高位先出，用户主机可在 HCLK 的上升沿读取 HDAT 的值。在步骤②中，HINT 为高，表示接收的是读指令共接收字节数的长度。(2 表示在步骤①之后，会再跟随有 1 个字节的指令/数据)
- ③ 步骤③，用户主机变低 HINT 表示接收指令/数据字节。长度为步骤②接收的数值减 1。
- ④ 用户主机升高 HCLK 及变低 HCLK 后 CTB 会升高 HDAT，表示所发送的字节已完成，用户主机在升高 HCLK 及 HINT 后总线进入空闲状态。
- ⑤ 请注意，第②步固定为 8 位，用户主机必须判断接收到的数必须为 2-18 之间的值，如超出则退出，等待下一次接收

3. CTB 与用户主机通信的注意事项

- ① 用户主机在发送 HCLK 的每个脉冲之间，包括由低变高及由高变低，必须保证间隔时间大于 15us，以确保与 CTB 之间的正确通信。
- ② CTB 总线的空闲状态为 HCLK、HDAT、HINT 都为高，在用户主机端 HINT 为输入，随时检测 HINT 的变化，如果变低则进行相应的写指令程序。
- ③ CTB 内置 8 组的接收 FIFO 接收寄存器，可连续接收多组用户主机的指令。多于连续 8 组的指令，则必须延时 300 毫秒左右再发送。
- ④ CTB 内置 8 组的发送 FIFO 寄存器，可保存 8 组的指令等待用户主机读取。用户主机在接收到 HINT 变低后，在不多于 300 毫秒的时间内接收相应的指令组。以防出现 FIFO 溢出的现象而漏收到 CTB 发出的指令。
- ⑤ CTB 总线只有最短时间，没有最大时间限制，只需保证 CTB 的 FIFO 不溢出即可。
- ⑥ 采用 CTB 总线的产品内置的 FIFO 不一定全部相同，以各自用户手册为准。
- ⑦ 采用 CTB 总线的产品最大传输长度不一定全部相同，以各自用户手册为准。

4. 用户主机发送指令解说

用户主机发送及接收到指令字节 A 为指令长度，字节 B 为指令类型，发送及接收时必须注意指令长度。

关于目录及子目录，请参阅《格式》获得更详细的说明：

用户主机作为主动方，随时写指令给 OTG12X。用户主机可发送从 2 字节到最多 18 字节的指令/数据给 OTG12X，也可以从 OTG12X 接收 2 到 18 字节的指令/数据。发送的格式如下：

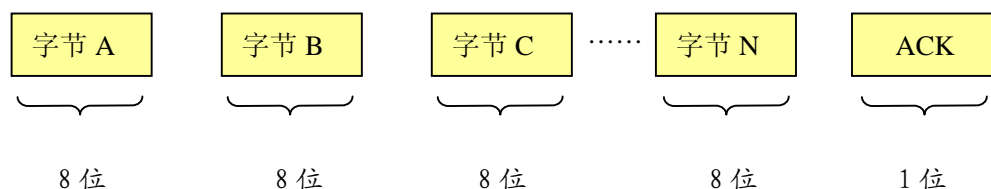
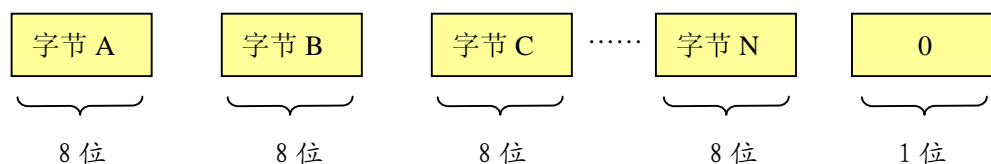


图 1 用户主机发送指令示意图

用户主机发送的指令/数据以 8 位为一个字节单位，分别叫字节 A、字节 B.....字节 N 等，字节 A 为描述本次发送的总字节数，例如为 2 则表示发送 16 位的指令/数据。

必须注意，当用户主机发送完指令/数据后，需回收一位的应答信号，本位固定为零，表示 OTG12X 正确接收到本次指令，否则无效须重发。详细请参阅《OTB 通讯总线应用注意》

用户主机发送的指令/数据分为 2 字节、3 字节、4 字节、多字节等几种，以下分别说明各种指令/数据：





3. CTB 与用户主机通信的 C 语言例程

```

#define WORD                unsigned int
#define BYTE                unsigned char
sfr P1                      = 0x90;

sbit pHCLK                 = P1 ^ 0;
sbit pHDAT                 = P1 ^ 1;
sbit pHINT                 = P1 ^ 1;

#define cCTBRetry           3                /* 发送失败后重复发送次数 */
#define cCTBMaxByte        18              /* CTB发送、接收最大字节数 */
#define cCTBDelay10us      10             /* 延时15微秒左右 */

#define cCTB_PLAY_PAUSE    0x31
#define cCTB_STOP          0x38
#define cCTB_SKIP_UP       0x42
#define cCTB_SKIP_DOWN     0x43

BYTE gCTBCommandBuffer[cCTBMaxByte];
BYTE GPKeyData;

void MDIPUpdateScreen();
void MRmKeyProcess();
BYTE MCTBCommandWrite();
BYTE MCTBCommandRead();
void mCTB_01_Pulse();

void main(){
    pHCLK = 1;                /* CTB总线初始化端口 */
    pHDAT = 1;
    pHINT = 1;
    /* 其他初始化 */
    while (1){
        if (!pHINT){
            if (!MCTBCommandRead()){
                MDIPUpdateScreen();
            }
        }
        MRmKeyProcess();      /* 调用遥控、按键接收键值处理 */
        /* 其他处理程序 */
    }
}

void MDIPUpdateScreen(){
    /* 显示程序 */
    return;
}

void MRmKeyProcess(){
    /* 遥控、按键接收键值处理 */
    if (GPKeyData != 0){
        gCTBCommandBuffer[0] = 0x02;
        gCTBCommandBuffer[1] = 0xff;
        switch (GPKeyData){
            case 0xcf :
                gCTBCommandBuffer[1] = cCTB_PLAY_PAUSE;
                break;
            case 0xfa :
                gCTBCommandBuffer[1] = cCTB_SKIP_UP;
                break;
            case 0xfc :
                gCTBCommandBuffer[1] = cCTB_SKIP_DOWN;
                break;
            case 0xb7 :
                gCTBCommandBuffer[1] = cCTB_STOP;
                break;
            default : break;
        }
        if (MCTBCommandWrite()){
            /* 没有找到DR53X, 显示出错或作其他处理 */
        }
        GPKeyData = 0;
    }
    return;
}

```



```

/* 龙珠科技CTB总线用户主机端程序V1.02版本(2003年5月20日) 技术支持 support@HSAV.com */
BYTE MCTBCommandWrite(){ /* 从gCTBCommandBuffer[]中发送 */
BYTE gLocal_1; /* 发送字节暂存器 */
BYTE gLocal_2; /* 发送位长度暂存器 */
BYTE gLocal_3; /* 发送字节长度暂存器 */
BYTE gLocal_4; /* 发送失败重试暂存器 */

gLocal_4 = cCTBRetry; /* 设置重复发送初值 */
do { /* 发送字节长度计数器 */
gLocal_3 = 0; /* 发送字节长度计数器 */
do{ /* 位长度计数器 */
gLocal_1 = gCTBCommandBuffer[gLocal_3];
gLocal_2 = 8;
do{
if (gLocal_1 & 0x80) pHDAT = 1;
else pHDAT = 0;
gLocal_1 <<= 1; /* 数据高位先出 */
mCTB_01_Pulse(); /* HCLK 0 -> 1 脉冲 */
} while (--gLocal_2 != 0);
pHINT = 0; /* 变低HINT发送指令/数据字节 */
} while (++gLocal_3 < gCTBCommandBuffer[0]);
mCTB_01_Pulse();
pHDAT = 1;
if (!pHDAT){
mCTB_01_Pulse(); /* HCLK 0 -> 1 脉冲 */
break; /* 接收到ACK位正确退出 */
}
mCTB_01_Pulse();
} while (--gLocal_4 != 0); /* ACK接收不正确则重试 */
pHINT = 1;
pHDAT = 1;
if (gLocal_4 == 0) return 1; /* 正常发送回传0,否则为1 */
return 0;
}

BYTE MCTBCommandRead(){ /* 接收到gCTBCommandBuffer[]中 */
BYTE gLocal_1; /* 接收字节暂存器 */
BYTE gLocal_2; /* 接收位长度暂存器 */
BYTE gLocal_3; /* 接收字节长度暂存器 */
BYTE gLocal_4; /* 接收返回标志暂存器 */
BYTE gLocal_5; /* 接收缓存指针暂存器 */
pHDAT = 1; /* 用户主机读指令,读/写位为1 */
mCTB_01_Pulse();
mCTB_01_Pulse();
gLocal_5 = 0;
gLocal_3 = 0; /* 接收字节长度计数器 */
do{ /* 接收8位数据 */
gLocal_2 = 8;
do{
gLocal_1 <<= 1;
if (pHDAT) gLocal_1 |= 0x01;
mCTB_01_Pulse(); /* HCLK 0 -> 1 脉冲 */
} while (--gLocal_2 != 0);
pHINT = 0; /* 变低HINT接收指令/数据字节 */
gCTBCommandBuffer[gLocal_5++] = gLocal_1; /* 将数据移入接收缓存 */
if (gLocal_3 == 0){
gLocal_3 = 1;
gLocal_4 = gLocal_1;

if (gLocal_4 < 2 || gLocal_4 > cCTBMaxByte){
gLocal_4 = 1; /* 如果数据长度超出范围 */
mCTB_01_Pulse(); /* 出错退出 */
break;
}
}
} while (--gLocal_4 != 0);
pHINT = 1;
return gLocal_4; /* 正常接收回传0,否则为1 */
}
}

```



```
void mCTB_01_Pulse(){ /* 发送HCLK变低 -> 变高脉冲 */
    BYTE gLocal_1;
    pHCLK = 0;
    gLocal_1 = cCTBDelay10us; /* 延时15微秒左右 */
    while (--gLocal_1 != 0);
    pHCLK = 1;
    gLocal_1 = cCTBDelay10us;
    while (--gLocal_1 != 0);
    return;
}
```