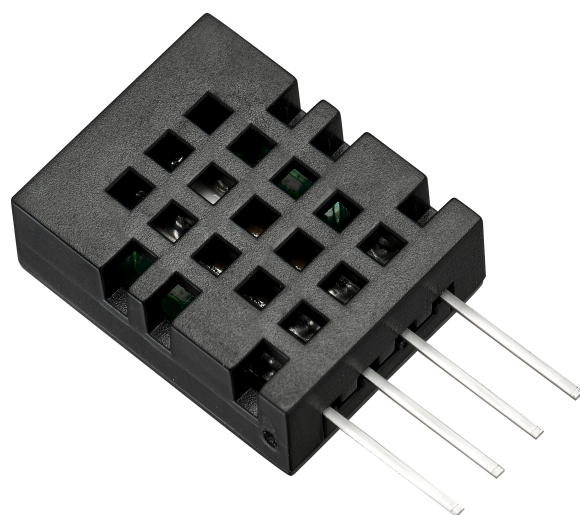




温湿度模块

DHT21 产品手册

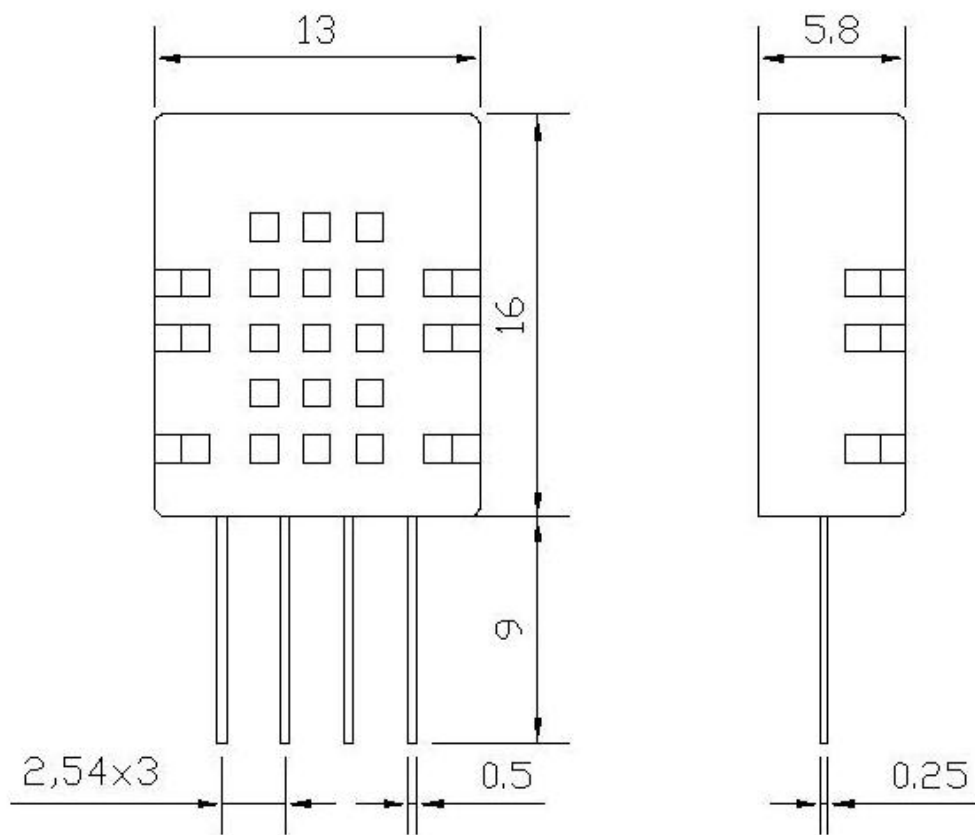


一、产品概述

本产品是采用高稳定性电容式感湿元件作为传感元件，经过微处理器采集处理转化成数字信号输出。每一个传感器都经过标定校准和测试。具有长期稳定、可靠性高、精度高、低功耗等特点。

二、尺寸图

单位: mm(± 0.5)



三、产品特点

DHT21 数字温湿度模块具有以下特点：

- 1、数字输出，单总线协议，通信距离支持 100 米；
- 2、超低功耗；
- 3、0-100%相对湿度测量范围；
- 4、全标定、温漂校准。
- 5、使用独立感湿元器件，稳定性好，抗污染能力强

四、性能特征

相对湿度

参数	条件	最小	典型	最大	单位
分辨率			0.1		%RH
量程范围		0		99.9	%RH
精度	25℃		±3		%RH
重复性			±0.1		
响应时间	1/e (63%)		<8		S
迟滞			±0.5		
漂移	典型值		<3		%RH/r

表 1

温度

参数	条件	最小	典型	最大	单位
分辨率			0.1		℃
工作范围		-40		80	℃
精度			±0.5		℃
重复性			±0.2		℃
响应时间	1/e (63%)	1			S
迟滞			±0.1		
漂移	典型值		<0.2		℃/r

表 2

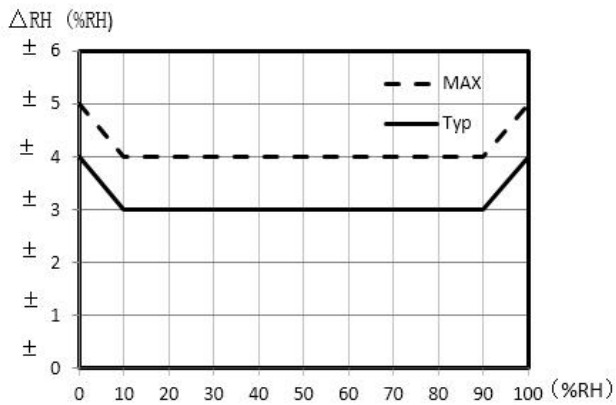


图 1

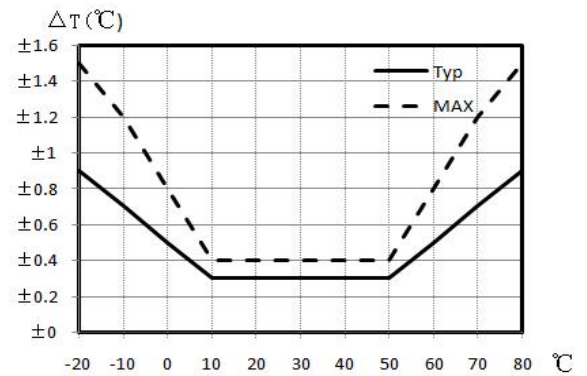


图 2

五、电气特性

参数	条件	最小	典型	最大	单位
供电电压 VDD		2.8	3.3	5.5	V
供电电流	休眠模式		0.2		μA
	测量模式		500		μA
采样周期			2.0		S
低电平输出电压	$I_o < 4\text{mA}$	0		250	mV
高电平输出电压	$R_p < 25\text{k}\Omega$	80%		100%	VDD
低电平输入电压	下降沿	0%		20%	VDD
高电平输入电压	上升沿	80%		100%	VDD
输出电流	0n			4	mA
	三态门 (Off)		10	20	μA

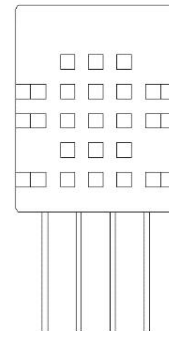
表 3 I_o 表示低电平输出电流 R_p 代表上拉电阻

六、用户指南

1、引脚分配

引脚	名称	描述
1	VDD	电源 2.8~5.5V
2	SDA	串行数据，双向口
3	NC	NC
4	GND	地

表4：引脚分配



1. 2. 3. 4.
图 3

1.1、电源引脚（VDD GND）

本产品的供电电压为2.8~5.5V，建议供电电压为3.3V。

1.2、串行数据（SDA）

SDA 为数据口，三态结构，SDA 在 SCK 时钟下降沿之后改变状态，并仅在 SCK 上升沿有效

2、通信协议

DHT21 为了精确测量气体的湿度，减少温度对测量的影响，DHT21 传感器在非工作期间，自动转为休眠模式，以降低传感器自身的发热对周围气体湿度的影响。DHT21 采用被动式工作模式，即主机通过指令唤醒传感器后，传感器才开始测量、应答等动作。通讯结束后，传感器进入休眠状态。

2.1、DHT21 连接图

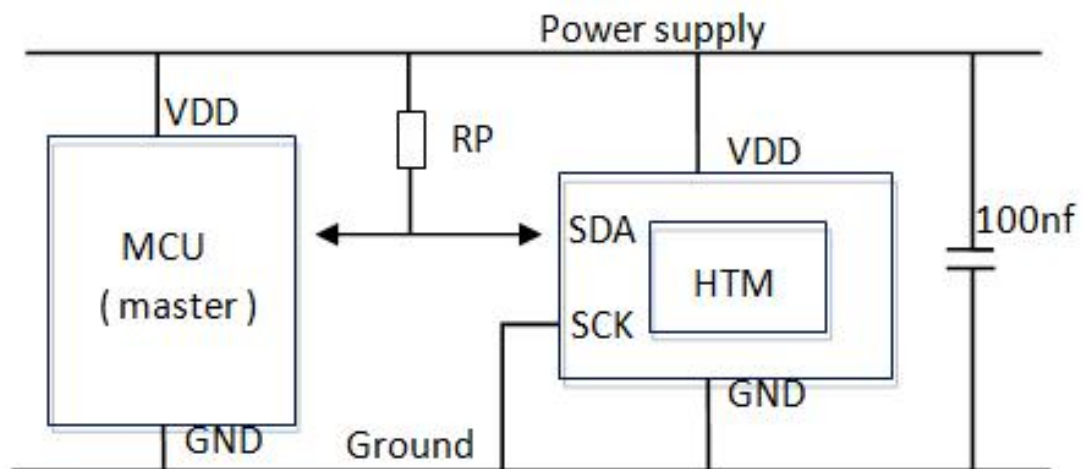
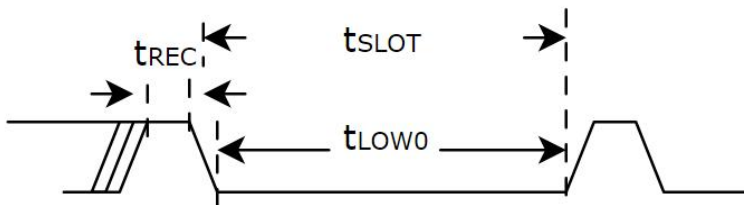


图 4 单总线通讯连接图

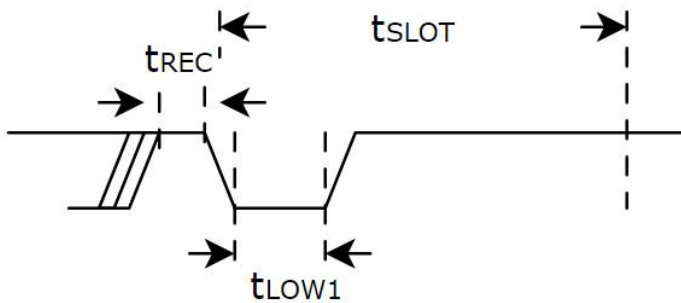
2.2、单总线接口时序

参数	符号	最小	典型	最大	单位
时序长度	t_{SLOT}	60		120	us
恢复时间	t_{REC}	1	5		us
写 0 低电平时间	t_{LOW0}	60	60	120	us
写 1 低电平时间	t_{LOW1}	1	5	15	us
读数据有效时间	t_{RDV}		5	15	us
复位低电平时间	t_{RSTL}	480	960		us
应答高电平时间	t_{PDHIGH}	15	30	60	us
应答低电平时间	t_{PDLOW}	60	115	240	us

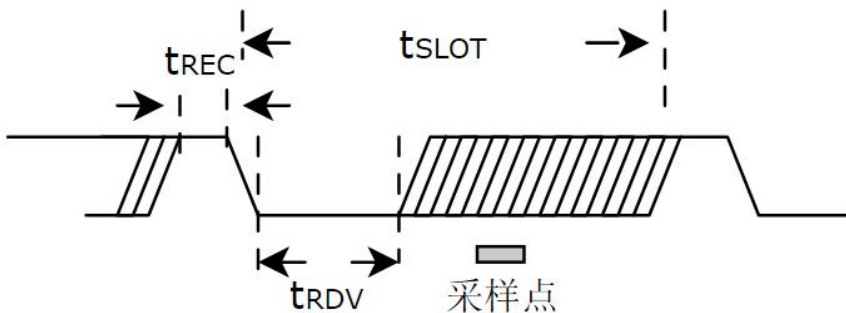
单总线写 0 时序



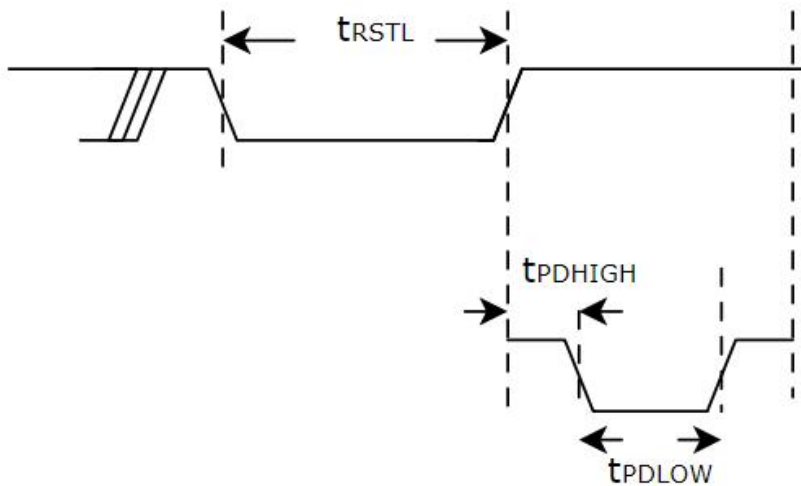
单总线写 1 时序



单总线读取时序



单总线复位时序



应答

读取湿度校准数 A、B	CCDDH
触发温湿度测量	CC10H
读取转化结果	CCBDH

时序说明：（详细请参考程序例程附件）

每一帧时序长度为 60us。每次数据通信前使用单总线复位时序，同步时序。

紧接着监测应答信号。

发送指令 0xCCDD 获取湿度校准数据 A、B

发送指令 0xCC10 触发温湿度转化。耗时约 30ms

发送指令 0xCCBD 获取温湿度转化结果。

客户同样可以无需等待读取，则此时读取的为上次触发转化的温湿度数据。

温度、相对湿度转换

通过上面两种方式可以获得 16bit 温度原始数据 S_t 、16bit 湿度原始数据 S_h

最终带入下面公式

温度 (°C):

$$T = 40 + \frac{S_t}{256}$$

湿度 (%RH):

$$RH = 30 + \frac{(S_h - \text{HumB}) * 60}{\text{HumA} - \text{HumB}}$$

$$RH_{rel} = RH + 0.25 * (T - 25.0) \quad (\text{注: 温漂 } 1^\circ\text{C 为 } -0.25\%RH)$$

注：RH 限定在 0~100, if(RH>100) RH=100;
Else if(RH<0) RH=0;

5) 循环冗余校验 (CRC) 计算

```
u8 CRC8MHT_Cal(u8 *serial, u8 length)
{
    u8 result = 0x00;
    u8 pDataBuf;
    u8 i;

    while(length--) {
        pDataBuf = *serial++;
        for(i=0; i<8; i++) {
            if((result^(pDataBuf))&0x01) {
                result ^= 0x18;
                result >>= 1;
                result |= 0x80;
            }
            else {
                result >>= 1;
            }
            pDataBuf >>= 1;
        }
    }
    return result;
}
```

七、应用信息

1、工作条件

确保传感器性能正常稳定的工作，建议使用温度范围-40℃-80℃，湿度范围 0-99.9%RH。超出建议的范围可能导致测量结果暂时性漂移。

2、存储条件与恢复

湿度传感器为环境敏感型电子元器件，需要仔细防护。长期暴露在高浓度的化学蒸汽中将会致使传感器的测量产生漂移。因此建议将传感器存放于原包装内，并符合存储条件：温度范围 10℃

-50℃；湿度范围 20-60%RH。在生产和运输过程中，要保证传感器远离高浓度的化学溶剂。要避免使用挥发性胶水、粘性胶带、不干胶贴纸，或者具有挥发性的包装材料，如发泡塑料袋、泡沫塑料等。

3 温度影响

相对湿度，很大程度上依赖于温度。产品在出厂前都做了温度的校准补偿，测量湿度时，应尽可能的保证传感器在同一温度下工作，安装在产品上时要尽可能的远离热源。否则将无法准确的测试到气体的相对湿度。

八、特别说明

许可协议

以上内容由广州海谷电子科技有限公司提供，版权所有，未经本公司之书面许可，此手册中任何段落，章节内容均不得被摘抄、拷贝或以任何形式复制、传播，否则一切后果由违者自负，本公司保留一切法律权利。

本公司保留对手册所描述之产品规格进行修改的权利，恕不另行通知。订货前，请垂询当地代理商以获悉本产品的最新规格。

警告

使用及人身伤害

勿将本产品用于安全保护装置或急停设备上，以及由于本产品故障可能导致人身受到伤害的任何应用中；在使用本产品前，请仔细阅读本说明书中的内容；

禁止在易燃气体附近使用

禁止在易燃、易爆气体的场所使用；

严禁直接接触及传感器

为防止污染感湿膜，避免手指直接触摸元件表面；汗液会污染感湿膜会导致性能漂移，接触传感器请戴防静电手指套；

工作环境

建议使用温度范围-40℃-80℃，湿度范围 0-100%RH。超出建议的范围可能导致测量结果暂时性漂移；本产品对光线不敏感，但长时间暴露在太阳光或则紫外线辐射中，同样加速老化。

广州海谷电子科技有限公司

广州科学城香山路 17 号优宝科技园 A503

电话：020-62863270

传真：020-62863271

邮箱：2482423075@qq.com

网址：www.gzhaigu.com

附件 1:

协议说明:

- 1、上电后数据口 SDA 高电平。拉低>480us 后，让数据口 SDA 转换为输入状态对电平检测，等待该模块低电平响应信号。发现低电平响应信号后等待模块释放该总线（即变为高电平）。
- 2、写入字节，（低位先入），SDA 转为输出口，发送 1 即：拉低电平>1us <15us 然后拉高电平>60us
发送 0 即：拉低电平>60us 然后拉高电平(该时间要求不高典型值 5us)
- 3、读取字节，（低位先出），SDA 转为输出口，拉低电平 (>1us 典型 5us <15us) 触发模块输出，释放总线延时 5us 左右后采样，当总线为高电平，即代表当前接收位为 1，否则为 0。

```
static u16 OwHumA, OwHumB;
```

```
//初始化，获取校准数据 A B 值
```

```
void DHT21Init_OW(void)
```

```
{
```

```
    uint8_t i, crc;
```

```
    u8 ResDat[13];
```

```
Timer4Stop(); //暂停有可能打断通信的中断
```

```
    DQ_Rst();
```

```
    DQ_Presence();
```

```
    DQ_Write_Byte(0xcc);
```

```
    DQ_Write_Byte(0xdd);
```

```
    OwHumA = DQ_Read_Byte();
```

```
    OwHumA = (OwHumA<<8) | DQ_Read_Byte();
```

```
    OwHumB = DQ_Read_Byte();
```

```
    OwHumB = (OwHumB<<8) | DQ_Read_Byte();
```

```
Timer4Start();
```

```
//当读取满一组数据即 13 个字节后，才有 CRC 值
```

```
//无需要时可以按上面读取前面 4 个即可
```

```
/*
```

```

for(i=0;i<13;i++)
{
    ResDat[i] = DQ_Read_Byte();
}
crc = CRC8MHT_Cal(ResDat, 13);
if(crc == 0)
{
    OwHumA = ResDat[0];
    OwHumA = (OwHumA<<8) | ResDat[1];
    OwHumB = ResDat[2];
    OwHumB = (OwHumB<<8) | ResDat[3];
}
*/
}

```

//读取温湿度函数

```

u8 ReadDHT21_onewire(s16 *tem, u16 *hum)
{
    u8 ResDat[5], crc=0, ReBit;
    u16 i;
    s16 TemBuf;
    Long CapBuf;//s32

```

Timer4Stop();//单总线通讯 暂停中断

```

    DQ_Rst();
    DQ_Presence();
    DQ_Write_Byte(0xcc);
    DQ_Write_Byte(0x10);

```

Timer4Start();

//可以不延时直接读取，但读取到的是上次转化的数据
 delay(7075);//2ms*15 35ms 改时间可以去处理其他任务回来读取

Timer4Stop();

```

    DQ_Rst();
    DQ_Presence();
    DQ_Write_Byte(0xcc);
    DQ_Write_Byte(0xbd);
    ResDat[0] = DQ_Read_Byte();
    ResDat[1] = DQ_Read_Byte();
    ResDat[2] = DQ_Read_Byte();
    ResDat[3] = DQ_Read_Byte();
    ResDat[4] = DQ_Read_Byte();

```

Timer4Start();

```

    crc = CRC8MHT_Cal(ResDat, 5);
    if(crc == 0)

```

```

    {
        TemBuf = (u16)ResDat[1]<<8|(ResDat[0]);
        TemBuf = 400+TemBuf/25.6;//*10 结果*10 倍 286 即 28.6℃
        *tem = TemBuf;

        CapBuf = (u16)ResDat[3]<<8|(ResDat[2]);
        CapBuf = (CapBuf-0wHumB)*600/(0wHumA-0wHumB)+300;//同样结果*10

        //20℃为 5 个湿度点 即 1℃为 0.25 个湿度点 0.1℃ 为 0.025
        CapBuf = CapBuf+ 25*(TemBuf-250)/100;

        if(CapBuf>999)CapBuf = 999;
        else if(CapBuf<0)CapBuf=0;
        *hum = (u16)CapBuf;
    }

    return crc;
}

//子函数-----
void DQ_Rst(void)
{
    HDCSDA_Output();
    DelaySus(2); //5us 无需严格要求
    HDCSDA_CLR();
    delay(114); //>480us 典型值 960us 规格书: tRSTL
    HDCSDA_SET();
    DelaySus(7); //8us 无需严格要求
}

//应答 DQ_Rst 低电平释放后, 模块会有一个 tPDLOW 的应答信号
u8 DQ_Presence(void)
{
    u8 pulse_time = 0;
    HDCSDA_Input();
    DelaySus(2); //5us 无需严格要求
    while( (HDCGet_SDA()) && pulse_time<100 ) //存在检测高电平 15~60us 模块响应高电
    平时间 tPDHIGH 规格书:
    {
        pulse_time++;
        DelaySus(5);//>6us
    }
    if( pulse_time >=20 )
        return 0x01;
    else
        pulse_time = 0;//应答正常
}

```

```

while((HDCGet_SDA()==0) && pulse_time<240 ) //存在检测低电平时间 60~240us  tPDLOW
{
    pulse_time++;
    DelaySus(2); //1~5us
}
if( pulse_time >=10 )// 应答正常
{
    return 0x01;
}
else
    return 0x0;
}

```

//拉低最少 1us 后延时一小段时间如 6us, 后再 15us 内读取完。后延时 45us 以上 即一个周期是 60us

```
//tINIT+tRC+tSample < 15us
```

```
//tINIT+tRC+tSample+tDelay>60us
```

```
u8 DQ_Read_Bit(void)
```

```

{
    u8 dat;

    HDCSDA_Output();

    HDCSDA_CLR();
    DelaySus(2); //tINIT>1us 典型 5us <15us
//    HDCSDA_SET();
    HDCSDA_Input();
    DelaySus(5); //tRC 典型 5us
    if(HDCGet_SDA())//tSample
        dat = 1;
    else
        dat = 0;
    DelaySus(33); //tDelay >60us 确保一帧数据传输完毕
    return dat;
}

```

```
u8 DQ_Read_Byte(void)
```

```

{
    u8 i, j, dat = 0;

    for(i=0; i<8; i++)
    {
        j = DQ_Read_Bit();
        dat = (dat) | (j<<i);
    }
    return dat;
}

```

```

}

void DQ_Write_Byte(uint8_t dat)
{
    uint8_t i, testb;
    HDCSDA_Output();
    for( i=0; i<8; i++ )
    {
        testb = dat&0x01;
        dat = dat>>1;
        if(testb)//写 1
        {
            HDCSDA_CLR();
            DelaySus(3);//>1us <15us
            HDCSDA_SET();
            DelaySus(33);//>=60us
        }
        else//写 0
        {
            HDCSDA_CLR();
            DelaySus(38);//MY_DELAY_US(70);>60us
            HDCSDA_SET();
            DelaySus(3);//典型 5us
        }
    }
}

u8 CRC8MHT_Cal(u8 *serial, u8 length)
{
    u8 result = 0x00;
    u8 pDataBuf;
    u8 i;

    while(length-->0) {
        pDataBuf = *serial++;
        for(i=0; i<8; i++) {
            if((result^(pDataBuf))&0x01) {
                result ^= 0x18;
                result >>= 1;
                result |= 0x80;
            }
            else {
                result >>= 1;
            }
            pDataBuf >>= 1;
        }
    }
    return result;
}

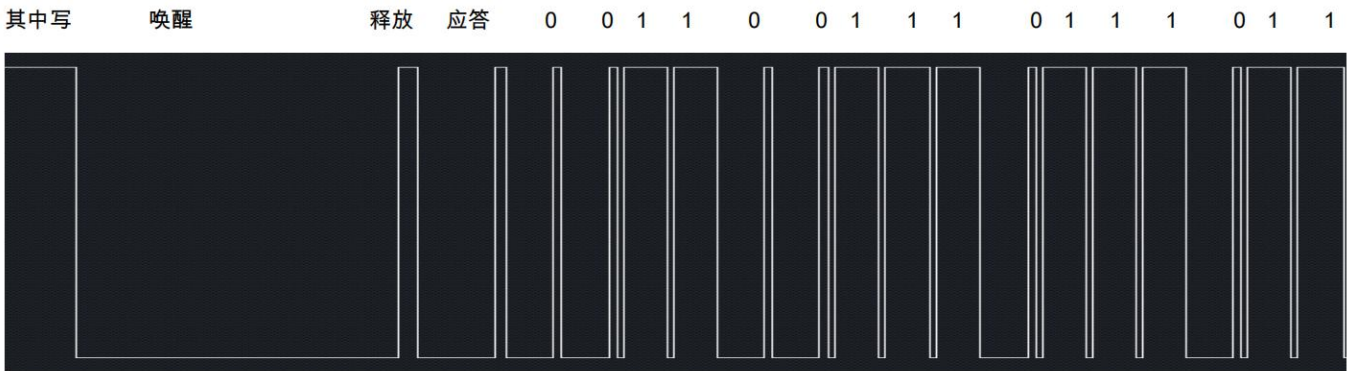
```

时序例程:

这是重置总线，然后写入 0xccdd 命令，获取校准值 A、B 的时序

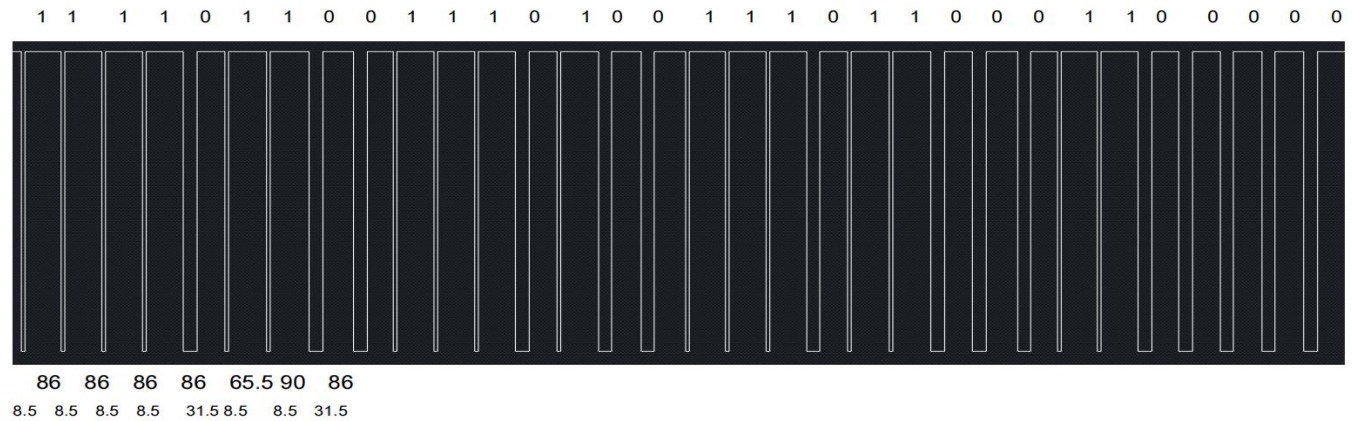


R A 0011 00 111 0111011



高电平		31	16.5	13.5	13	66	66	13.5								
低电平	495us		118	71	71	10	10	71								

即按低位先出，写入的数据是：11001100 即 0xcc
11011101 即 0xdd



即得到 4 个字节依次是（低位先出）：01101111 即 0x6f
00101110 即 0x2e 即 HumA 为 0x6f2e 即 28462
00110111 即 0x37
00000110 即 0x06 即 HumB 为 0x3706 即 14086