

# 产品技术规格书

Product standards document

客户名称 Customer Name: \_\_\_\_\_

客户编号 Customer NO.: \_\_\_\_\_

产品名称 Our Name: \_\_\_\_\_ 声光报警器

产品型号 Product model: \_\_\_\_\_ GD-S001

客户确认回复 Customer Approval Status:

签核 Signature:

## 声光报警器产品规格书

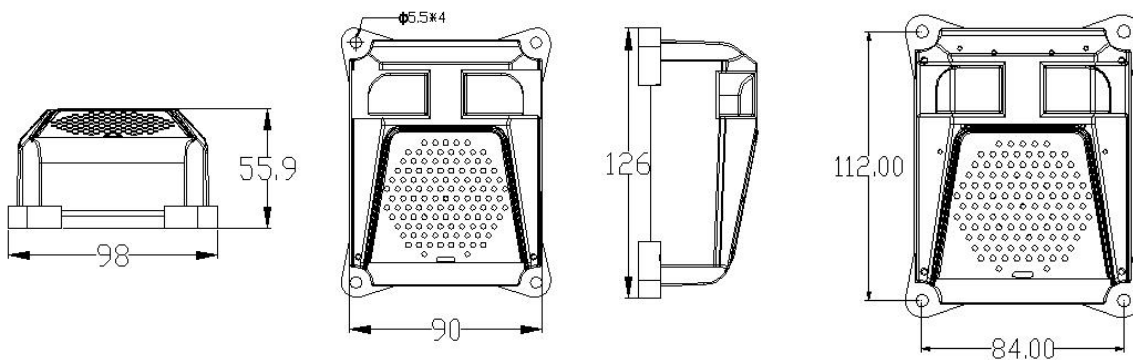
### 一：产品简介

1. **GD-S001** 是我司自主研发设计的一款声光报警器,此产品应用于多个领域。
2. 电压范围：**9-36V**.
3. 内置防水喇叭：**8W**
4. 语音播报。
5. 防水防尘：**IP67**

二：外观新颖美观，采取铝合金材料压铸，使用寿命长。

三：工作温度-30° ~+70° 。

四：外观尺寸图(单位)：mm



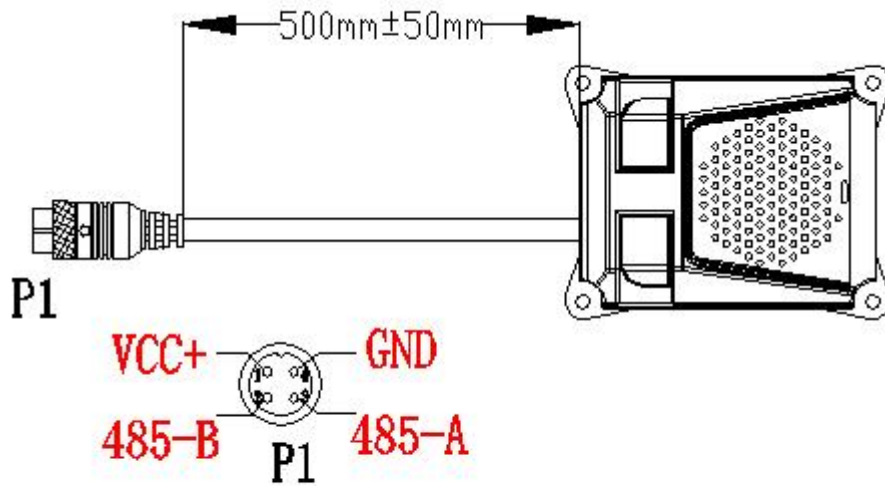
**深圳国鼎科技有限公司**  
**Shenzhen Guoding Technology Co.,LTD**

型号	GD-S001
工作电压	DC:9-36V
通讯接口	RS485
工作电流	静态电流: 35mA ,最大电流: 0.9A
通讯波特率	9600 bps
报警模式	声音/发光 语音提示
喇叭	2.5" 大口径防水喇叭
功率	D类功放 MAX:8W
LED 发光	红/绿 双色
接口	航空头 GX12-4 母头 0.6 米
防水等级	IP67
外壳材料	铝合金
重量	0.5KG
颜色	哑黑色

产品图片：



接口定义:



注: 以上相关规格、参数如有改变、恕不另行通知

## 通讯协议:

包头域	地址域	功能域	数据长度	数据域	CRC 校验	包尾域
2 个字节	1 个字节	1 个字节	2 个字节	N 字节	2 个字节	2 个字节

一: 通讯协议基本格式

注: 波特率 9600,n,8,1

设备默认地址 0x01

包头 0xaa55

包尾 0x55aa

数据格式模式, 例如数据长度为 1 个字节 数据序列为 0x00 0x01 高字节在前,  
低字节在后, 低字节在后。

二：返回命令

将发送的命令字或上 0x80,按通讯协议基本格式返回。

功能域	读/写	数据个数	说明
01	读	1	读取设备地址
02	写	1	更改设备地址
03	读	1	读设备波特率
04	写	1	波特率类型 1 个字节
			00-1200
			01-2400
			02-4800
			03-9600
			04-19200
			05-38400
			06-57600
05	写	1	报警功能，报警状态 1 个字节：
			00-全关闭
			01-声音 1+闪光
			02-闪光
			03-声音 1
			04-声音 2+闪光
			05-声音 2

三：命令格式

1、CRC 校验算法（C 语言）

UInt8\_t auchCRCHI[]=

{

```
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
```

```
0X40
```

};



深圳国鼎科技有限公司  
Shenzhen Guoding Technology Co.,LTD

---

Unit8\_t auchCRCLo[] =

```
{  
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,  
    0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,  
    0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,  
    0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,  
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,  
    0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F 0x3E, 0xFE, 0xFA, 0x3A,  
    0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,  
    0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,  
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,  
    0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,  
    0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,  
    0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,  
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,  
    0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,  
    0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,  
    0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,  
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,  
    0X40  
};
```

//校验算法函数

```
uint16_t N_CRC16(uint8_t*updata, uint16_t len)
{
    uint8_t uchCRCHi=0xff;
    uint8_t uchCRCLo=0xff;
    Uint16_t uindex;

    Whlie(len--)
    {
        uindex = uchCRCHi ^ *updata++;
        uchCRCHi = uchCRCLo ^  uchCRCHi[uindex];
        uchCRCLo = auchCRCLo[uindex];
    }
    return(uchCRCHi << 8 | uchCRCLo);
}
```