

US1D 二次开发手册 v1.1.4

固件版本 00.01.17

一、获取设备信息

- 发送指令:

长度: 24 字节

数据包(HEX): 01 00 10 00 00 00 BE F0 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00

- 接收指令(tDeviceInfo):

长度: 84 字节

包头: 0xFBFB

数据包(HEX): FB FB 4C 00 00 00 ED 36 53 45 49 49 43 48 49 20 54 65 63 68 6E 6F
6C 6F 67 69 65 73 00 55 53 31 44 2D 36 30 30 50 00 00 00 00 33 34 36 45 43 38 32
31 45 32 42 34 30 39 32 37 33 39 38 36 32 33 37 38 43 43 33 36 34 38 35 43 00 30
30 2E 30 31 2E 31 36 00

- 数据结构定义:

```
#pragma pack(1) //强制 1 字节对齐
typedef struct
{
    char    manufacturer[21]; //制造商名称[21]
    char    model[13];        //型号[13]
    char    serial[33];        //序列号[33]
    char    version[9];        //固件版本[9]
}tInfo;          //设备信息[76]

#pragma pack(1) //强制 1 字节对齐
typedef struct
{
    uint16_t head;    //包头    *固定 0xFBFB
    uint32_t len;      //包长度    *SizeOf(tInfo)= 76
    uint16_t crc;      //校验    *CRC16-MOUDBUS 校验
    tInfo info;        //设备信息*76byte
}tDeviceInfo;        //接收包    *设备信息数据包[84]
```

- 解析示例:

FB FB //head 包头

4C 00 00 00 //长度

ED 36 //CRC16-Modbus 校验值

53 45 49 49 43 48 49 20 54 65 63 68 6E 6F 6C 6F 67 69 65 73 00 //manufacturer
SEIICHI Technologies 制造商名称

55 53 31 44 2D 36 30 30 50 00 00 00 00 //model **US1D-600P** 型号

33 34 36 45 43 38 32 31 45 32 42 34 30 39 32 37 33 39 38 36 32 33 37 38 43 43 33
36 34 38 35 43 00 //serial **346EC821E2B4092739862378CC36485C** 序列号

30 30 2E 30 31 2E 31 36 00 //version **00.01.16** 固件版本号

试用水印

二、获取测量帧数据

- 发送指令：

长度：24 字节

数据包 (HEX)：05 00 10 00 00 00 7F 60 01 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00

- 接收指令：

长度：1036 字节

包头：0xFDFD

数据包 (HEX)：FD FD 04 04 00 00 4F 77 90 D0 03 00 17 00 0D 00 07 00 05 00 05 00
06 00 06 00 06 00 >此处省略 1008 字节帧数据

- 数据结构定义：

```
#pragma pack(2) //强制 2 字节对齐
typedef struct
{
    uint16_t head;           //包头      *固定 0xFDFD
    uint32_t len;            //包长度   *SizeOf(tDeviceInfo)= 76
    uint16_t crc;            //校验     *CRC16-MOUBUS 校验
    int32_t temp;            //温度     *真实温度= temperature/10000
    uint16_t data[1024];     //测量帧   *1024 点包络波形数据
} tEnvelopeFrame;          //接收包   *测量帧数据包 1036 字节
```

- 解析示例：

```
FD FD          //head 包头
04 04 00 00    //长度
4F 77          //CRC16-Modbus 校验值，该校验起始位为&temp 到包结束一共 1028 个字节
90 D0 03 00    //传感器温度值 250000/10000 =25 度

17 00 0D 00 07 00 05 00 05 00 06 00 ..... //1024 点包络帧数据
```

三、获取测量距离

- 发送指令：

长度：24 字节

数据包(HEX)： 06 00 10 00 00 00 25 3A 01 00 5A 00 00 00 00 00
00 00 00 00 00 00 00 00

- 接收指令：

长度：16 字节

包头：0xFCFC

数据包(HEX)： FC FC 08 00 00 00 4D 2D 90 D0 03 00 40 42 00 00

- 数据结构定义：

```
#pragma pack(2) //强制 2 字节对齐
typedef struct
{
    uint16_t head;           //包头          *固定 0x06
    uint32_t len;            //包长度        *固定 0x10
    uint16_t crc;            //校验          *CRC16-MOUDBUS 校验
    uint16_t mode;           //配置模式      *必须填写 1
    uint16_t blind;          //配置测量盲区 *单位厘米 cm 范围(10~200)
} tValueFrame;             //发送包        *测量距离数值包 24 字节
```

```
#pragma pack(2) //强制 2 字节对齐
typedef struct
{
    uint16_t head;           //包头          *固定 0xFCFC
    uint32_t len;            //包长度        *固定 0x8
    uint16_t crc;            //校验          *CRC16-MOUDBUS 校验
    int32_t temp;            //温度          *真实温度= temperature/10000
    uint32_t Distmm;         //距离          *测量距离单位毫米 mm
} tValueFrame;             //接收包        *测量距离数值包 16 字节
```

- 解析示例：

FC FC //head 包头
08 00 00 00 //长度
4D 2D //CRC16-Modbus 校验值，该校验起始位为&temp 到包结束一共 8 个字节
90 D0 03 00 //传感器温度值 250000/10000 =25 度
CA 0F 00 00 //测量距离数值单位 mm 4240/1000= 4.24 米

四、运行参数设置

- 发送指令:

长度: 24 字节

数据包(HEX): 04 00 10 00 00 00 67 5A 02 03 01 F0 49 02 00 32
00 00 00 00 00 00 00 00

- 接收指令(tDeviceConfig):

长度: 24 字节

包头: 0xFEFE

数据包(HEX): FE FE 10 00 00 00 AB B4 02 03 01 F0 49 02 00 32
00 08 0E 03 00 00 00 00

- 数据结构定义:

```
#pragma pack(1) //强制 2 字节对齐
typedef struct
{
    uint8_t DistIdx; //范围 indx
    uint8_t Gian; //增益
    uint8_t Reserved1; //保留 1 发送配置指令此字节必须为 0
    uint32_t SoundSpeed; //声速
    uint16_t Pluse; //脉宽
    uint32_t Reserved2; //保留 2 发送配置指令此字节必须为 0
    uint8_t Reserved3[3]; //保留 3 发送配置指令此字节必须为 0
}tConfig; //16

#pragma pack(2) //强制 2 字节对齐
typedef struct
{
    uint16_t head; //包头 *配置发送 0x04, 返回为固定 0xFEFE
    uint32_t len; //包长度 *SizeOf(tConfig)= 16
    uint16_t crc; //校验 *CRC16-MOUBUS 校验
    tConfig config; //配置参数 *16 字节
} tDeviceConfig; //发送和接收包 *24 字节
```

• 参数定义:

tConfig 结构中的每个字段都表示设备运行的不同参数，具体含义如下:

1. DistIdx: 表示测量距离范围索引有以下选择:

DistIdx	含义
1	测量范围 0~3.2 米
2	测量范围 0~6.4 米
3	测量范围 0~12.8 米
4	测量范围 0~25.6 米
5	测量范围 0~51.2 米
6	测量范围 0~102.4 米
7	测量范围 0~204.8 米

默认值为:2 测量范围 0~6.4 米, 如果不配置请设置为 0。

2. Gian: 表示前级接收信号增益范围有以下选择:

DistIdx	含义
1	放大 4 倍(12dB)
2	放大 8 倍(18dB)
3	放大 16 倍(24dB)
4	放大 32 倍(30dB)
5	放大 64 倍(36dB)
6	放大 128 倍(42dB)
7	放大 256 倍(48dB)

默认值为:2 放大 8 倍(18dB) , 如果不配置请设置为 0。

3. SoundSpeed: 表示声音在水中的速度单位是(厘米/秒)

一般声音在水中的传播速度是 150000 厘米/秒,但实际情况要根据水的温度和水的成分而定, 如果需要精确可以自行设置该值, 默认值为 150000, 如果不配置请设置为 0。

4. Pluse: 表示测量脉冲的宽度单位是微秒 us

测量范围越小此值应该越小一般选择范围在 30~100us 默认值为 50, 如果不配置请设置为 0。

5. Reserved*: 此类保留字段在发送配置指令时应设置为 0。

- 参数设置示例：

设置测量范围 0~6.4 米，放大倍数为 16 倍：

Head: 04 00 //固定为 4
 Len: 10 00 00 00 //固定为 16
 CRC16: 7C 90 //对以下字段数据进行 CRC16 校验值

DistIdx: 02 //距离范围 0~6.4 米取值为：2

Gain: 03 //放大 16 倍取值为：3

Reserved1: 00 //系统保留字段 1 取值为：0

SoundSpeed: 00 00 00 00 //不修改

Pluse: 00 00 //不修改

Reserved2: 00 00 00 00 //不修改

Reserved3: 00 00 00 //不修改

以上数据拼接起来为：

04 00 10 00 00 00 7C 90 02 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00

当发送以上拼接的数据后，设备将返回当前运行值：

FE FE 10 00 00 00 AB B4 02 03 01 F0 49 02 00 32 00 08 0E 03 00 00 00 00

对数据进行解析得到：

FE FE head
 10 00 00 00 len
 AB B4 CRC16
 02 DistIdx //设置的范围 0~6.4 米已经生效
 03 Gain //设置的放大 16 倍已经生效
 01 Reserved1
 F0 49 02 00 SoundSpeed
 32 00 Pluse
 08 0E 03 00 Reserved2
 00 00 00 Reserved3

五、CRC16-Modbus 校验算法

所有的接收或发送的数据包中请使用 CRC16-Modbus 校验算法来进行数据校验，以确保通讯数据正确性。对于固定的指令可以直接使用示例值，对于较长的包络帧数据接收最好使用 CRC16 的校验。

```

/*
  CRC16-Modbus function

  多项式:  $X^{16}+X^{15}+X^2+X+1$ 
  POLY: 0x8005
  INIT: 0xFFFF
  XOROut: 0
  输入数据反转: 是
  输出数据反转: 是
*/

const unsigned int CRC_16_Tab[16] =
{
    0x0000, 0xCC01, 0xD801, 0x1400, 0xF001, 0x3C00, 0x2800, 0xE401,
    0xA001, 0x6C00, 0x7800, 0xB401, 0x5000, 0x9C01, 0x8801, 0x4400
};

unsigned int ModBusCRC16(unsigned char *data, unsigned char len)
{
    unsigned char i,temp;
    unsigned int crc_16 = 0xffff;
    for(i=0;i<len;i++)
    {
        temp = ((unsigned char)(crc_16&0x000F))^(*data&0x0F);
        crc_16 >>= 4;
        crc_16 ^= CRC_16_Tab[temp];
        temp = ((unsigned char)(crc_16&0x000F))^(*data>>4);
        crc_16 >>= 4;
        crc_16 ^= CRC_16_Tab[temp];
    }
    return crc_16;
}

```

以下无正文。