



HPS-3D160

HPS3D160 简单通信协议

传感器每个命令包包含 2 个报头字节、1 个报文长度字节、1 个命令字节、1 个设备地址字节、参数域、2 个 CRC16-CCITT 校验字节；每个返回数据包包含 2 个报头字节、2 个报文长度字节、1 个设备地址字节、一个 RID 字节（Return ID，一般与命令字节一致）、数据域、2 个 CRC16-CCITT 校验字节、两个报尾字节。命令包和返回数据包的字节序为小端字节序，即低内存地址存放低字节数据。

命令数据包为不定长度格式，定义如下表：

字节号	描述
0	0xF5，报头字节 1
1	0x0A，报头字节 2
2	长度字节，指示字节号 3 开始的数据字节数
3	命令字节
4	设备地址，指定应答的设备，出厂地址默认 0x00，广播地址 0xFF
N	参数字段
5+N	CRC16 低字节
5+N+1	CRC16 高字节

注：命令数据包中参与 CRC 校验的字节为字节号 3~N。

返回数据包为可变长格式，定义如下表：

字节号	描述
0	0xF5，报头字节 1
1	0x5F，报头字节 2
2	剩余的有效数据长度低字节
3	剩余的有效数据长度高字节
4	设备地址
5	RID，返回包类型 ID
N	数据字段
6+N	CRC16 低字节
6+N+1	CRC16 高字节
6+N+2	0x5F，报尾字节 1
6+N+3	0xF5，报尾字节 2

注：返回数据包中参与 CRC 校验的字节为字节号 4~N。

命令#1 设定传感器的工作模式，此命令可设定传感器的工作模式。

字节号	值	描述
0	0xF5	报头字节 1
1	0x0A	报头字节 2
2	0x06	报文长度
3	0xA3	命令字节
4	目标设备地址	目标设备地址，0x00 ~ 0xFE
5	0x01	固定参数
6	0x00: 待机模式，0x01: 单次测量模式，0x02: 连续测量模式
7	CRC16 低字节
8	CRC16 高字节

返回数据:

字节号	名称	值	描述
0	报头字节 1	0xF5	报头字节 1
1	报头字节 2	0x5F	报头字节 2
2	报文长度低字节	0x07	报文长度低字节
3	报文长度高字节	0x00	报文长度高字节
4	设备地址	当前响应的设备地址
5	RID	0xA3	RID，返回包类型 ID
6	确认字节	0x01: 成功，0x00: 失败
7	CRC16 LSB	CRC16 低字节
8	CRC16 MSB	CRC16 高字节
9	报尾字节 1	0x5F	报尾字节 1
10	报尾字节 2	0xF5	报尾字节 2

示例：连续测量模式命令

字节号	值	描述
0	0xF5	报头字节 1
1	0x0A	报头字节 2
2	0x06	报文长度
3	0xA3	命令字节
4	0x00	目标设备地址，0x00 ~ 0xFE,默认地址为 0x00
5	0x01	固定参数
6	0x02	0x00: 待机模式，0x01: 单次测量模式，0x02: 连续测量模式
7	0x19	CRC16 低字节，根据设备地址为 0x00 时得到的 CRC16 校验值
8	0xe6	CRC16 高字节，根据设备地址为 0x00 时得到的 CRC16 校验值

单次测量模式命令

字节号	值	描述
0	0xF5	报头字节 1
1	0x0A	报头字节 2
2	0x06	报文长度
3	0xA3	命令字节
4	0x00	目标设备地址，0x00 ~ 0xFE,默认地址为 0x00

5	0x01	固定参数
6	0x01	0x00: 待机模式, 0x01: 单次测量模式, 0x02: 连续测量模式
7	0x7a	CRC16 低字节, 根据当前参数计算得到的 CRC16 校验值
8	0xd6	CRC16 高字节, 根据当前参数计算得到的 CRC16 校验值

停止测量模式命令

字节号	值	描述
0	0xF5	报头字节 1
1	0x0A	报头字节 2
2	0x06	报文长度
3	0xA3	命令字节
4	0x00	目标设备地址, 0x00 ~ 0Xfe, 默认地址为 0x00
5	0x01	固定参数
6	0x00	0x00: 待机模式, 0x01: 单次测量模式, 0x02: 连续测量模式
7	0x5b	CRC16 低字节, 根据当前参数计算得到的 CRC16 校验值
8	0xc6	CRC16 高字节, 根据当前参数计算得到的 CRC16 校验值

测量数据包数据解析

测量数据包分为以下四种:

- 完整数据包: 包含关键的测量数据和全视角的深度数据, 适用于需要全视角的深度数据做二次开发的应用场合, 但对对终端设备的数据处理能力要求较高。数据包数据域定义如下:

报头	报文长度	设备地址	RID	测量数据	CRC16 值	报尾
----	------	------	-----	------	---------	----

其中, 测量数据域详细格式如下:

字节号	名称	值	描述
0 ~ 1	Dummy	任意值	该值可忽略
2	平均距离	低字节	全视角的平均距离, 单位: 毫米 (mm)
3		高字节	
4	有效信号强度	低字节	有效信号强度, 该值无单位
5		高字节	
6	平均信号强度	低字节	全视角的平均信号强度, 该值无单位, 数值越大反射信号强度越高, 具体定义如下: 平均信号强度 < 150: 信号强度弱 150 <= 平均信号强度 <= 800: 信号强度良好 平均信号强度 > 800: 信号过强
7		高字节	
8	低信号像素个数	低字节	信号强度弱的像素个数
9		高字节	
10	饱和像素个数	低字节	饱和像素个数
11		高字节	
12	最大距离值	低字节	视角范围内的最大距离值, 如果该值为 0, 则代表无效数据
13		高字节	
14	最小距离值	低字节	视角范围内的最小距离值, 如果该值为 65535, 则代表无效数据
15		高字节	
16	数据帧计数器	最低字节	测量数据帧的计数器数据, 方便调试数据传输, 用来检查是否丢帧
17		次低字节	
18		次高字节	

19		最高字节	
20	保留字节	
21			
22			
23			
19200 字节	深度数据	每个像素点的 距离数据由 2 个字节表示， 低字节号存放 低字节数据	数据排列为： 像素点 1...像素点 160 像素点 161...像素点 320 像素点 9440...像素点 9600

2. 精简测量数据包：仅包含关键的测量数据，适用于仅需要全视角的关键测量数据的应用场合，对终端设备的数据处理能力和通讯速率要求较低。数据包数据域定义如下：

字节号	名称	值	描述
0 ~ 1	Dummy	任意值	该值可忽略
2	平均距离	低字节	全视角的平均距离，单位：毫米（mm）
3		高字节	
4	有效信号强度	低字节	有效信号强度，该值无单位
5		高字节	
6	平均信号强度	低字节	全视角的平均信号强度，该值无单位，数值越大反射信号强度越高，具体定义如下： 平均信号强度 < 150：信号强度弱 150 ≤ 平均信号强度 ≤ 800：信号强度良好 平均信号强度 > 800：信号过强
7		高字节	
8	低信号像素点数	低字节	信号强度弱的像素点个数
9		高字节	
10	饱和像素点数	低字节	饱和像素点个数
11		高字节	
12	最大距离值	低字节	视角范围内的最大距离值，如果该值为 0，则代表无效数据
13		高字节	
14	最小距离值	低字节	视角范围内的最小距离值，如果该值为 65535，则代表无效数据
15		高字节	
16	数据帧计数器	最低字节	测量数据帧的计数器数据，方便调试数据传输，用来检查是否丢帧
17		次低字节	
18		次高字节	
19		最高字节	
20 ~ 23	保留字节	保留字节

3. 敏感区域完整数据包：包含每个敏感区域的关键测量数据和敏感区域的深度数据，适用于仅需要视角中某特定敏感区域信息的应用，对终端设备的数据处理能力要求适中。数据包数据域定义如下：

报头	报文长度	设备地址	RID	敏感区域信息	敏感区域测量数据 1	敏感区域测量数据 N	CRC16	报尾
----	------	------	-----	--------	------------	-------	------------	-------	----

其中，敏感区域信息的数据域定义如下：

字节号	名称	值	描述
0	使能的敏感区域个数	0x00~0x07	根据敏感区域使能个数而定
1	敏感区域组 ID	0x00~0x0F	当前使用的敏感区域组的 ID 号
2	数据帧计数器	最低字节	测量数据帧的计数器数据，方便调试数据传输，用来检查是否丢帧
3		次低字节	
4		次高字节	
5		最高字节	
6 ~ 23	保留字节	保留字节

敏感区域测量数据域定义如下：

字节号	名称	值	描述
0	敏感区域 ID	低字节	当前敏感区域的 ID 号
1		高字节	
2	敏感区域左上角 X 坐标	低字节	当前敏感区域左上角 X 坐标
3		高字节	
4	敏感区域左上角 Y 坐标	低字节	当前敏感区域左上角 Y 坐标
5		高字节	
6	敏感区域右下角 X 坐标	低字节	当前敏感区域右下角 X 坐标
7		高字节	
8	敏感区域右下角 Y 坐标	低字节	当前敏感区域右下角 Y 坐标
9		高字节	
10	平均信号强度	低字节	敏感区域的平均信号强度，该值无单位，数值越大反射信号强度越高，具体定义如下： 平均信号强度 < 150：信号强度弱 150 ≤ 平均信号强度 ≤ 800：信号强度良好 平均信号强度 > 800：信号过强
11		高字节	
12	有效信号强度	低字节	敏感区域区域内除去信号弱和信号过强像素点后的有效信号强度，该值无单位，数值越大反射信号强度越高
13		高字节	
14	平均距离	低字节	敏感区域内像素的平均测量距离
15		高字节	
16	最大距离值	低字节	敏感区域内的最大距离值，如果该值为 0，则代表无效数据
17		高字节	
18	最小距离值	低字节	敏感区域内的最小距离值，如果该值为 65535，则代表无效数据
19		高字节	
20	饱和像素点数	低字节	饱和像素点个数
21		高字节	
22	阈值比较结果	低字节	Bit0 ~ Bit2: 阈值 0, 阈值 1, 阈值 2, 触发报警或警报解除时相应的阈值位自动置 1 或置 0 Bit3~Bit15:保留
23		高字节	
24	最大距离值 X 坐标	低字节	敏感区域中最大距离值像素相对于全视角的 X 坐标
25		高字节	
26	最大距离值 Y 坐标	低字节	敏感区域中最大距离值像素相对于全视角的 Y 坐标
27		高字节	
28	最小距离值 X 坐标	低字节	敏感区域中最小距离值像素相对于全视角的 X 坐标

29		高字节	
30	最小距离值 Y 坐标	低字节	敏感区域中最小距离值像素相对于全视角的 Y 坐标
31		高字节	
.....	敏感区域深度数据	每个像素点的距离数据由 2 个字节表示，低字节号存放低字节数据	起始数据为左上角第一个像素，其余数据按行顺序输出

4. 敏感区域简单数据包：仅包含每个敏感区域的关键测量数据，对终端设备的数据处理能力和通讯速率要求较低。数据包数据域定义如下：

报头	报文长度	设备地址	RID	敏感区域信息	敏感区域测量数据 1	敏感区域测量数据 N	CRC16	报尾
----	------	------	-----	--------	------------	-------	------------	-------	----

其中，敏感区域信息的数据域定义如下：

字节号	名称	值	描述
0	使能的敏感区域个数	0x00~0x07	根据敏感区域使能个数而定
1	敏感区域组 ID	0x00~0x0F	当前使用的敏感区域组的 ID 号
2	数据帧计数器	最低字节	测量数据帧的计数器数据，方便调试数据传输，用来检查是否丢帧
3		次低字节	
4		次高字节	
5		最高字节	
6 ~ 23	保留字节	保留字节

敏感区域测量数据域：

字节号	名称	值	描述
0	敏感区域 ID	低字节	当前敏感区域的 ID 号
1		高字节	
2	平均信号强度	低字节	敏感区域的平均信号强度，该值无单位，数值越大反射信号强度越高，具体定义如下： 平均信号强度 < 150：信号强度弱 150 ≤ 平均信号强度 ≤ 800：信号强度良好 平均信号强度 > 800：信号过强
3		高字节	
4	有效信号强度	低字节	敏感区域区域内除去信号弱和信号过强像素点后的有效信号强度，该值无单位，数值越大反射信号强度越高
5		高字节	
6	平均距离	低字节	敏感区域内像素的平均测量距离
7		高字节	
8	最大距离值	低字节	敏感区域内的最大距离值，如果该值为 0，则代表无效数据
9		高字节	
10	最小距离值	低字节	敏感区域内的最小距离值，如果该值为 65535，则代表无效数据
11		高字节	
12	饱和像素点数	低字节	饱和像素点个数
13		高字节	
14	阈值比较结果	低字节	Bit0 ~ Bit2：阈值 0，阈值 1，阈值 2，触发报警或

15		高字节	警报解除时相应的阈值位自动置 1 或置 0 Bit3~Bit15:保留
16	最大距离值 X 坐标	低字节	敏感区域中最大距离值像素相对于全视角的 X 坐标
17		高字节	
18	最大距离值 Y 坐标	低字节	敏感区域中最大距离值像素相对于全视角的 Y 坐标
19		高字节	
20	最小距离值 X 坐标	低字节	敏感区域中最小距离值像素相对于全视角的 X 坐标
21		高字节	
22	最小距离值 Y 坐标	低字节	敏感区域中最小距离值像素相对于全视角的 Y 坐标
23		高字节	
24 ~ 32	保留字节	保留字节

修订历史记录

Date	Revision	Description
2021/01/11	1.0	初始版本。

附录

CRC16 的 C 语言实现

```
static const USIGN16 crc16_tab[] = {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
    0xdbfd, 0xcbbc, 0xfbff, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
    0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
    0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
    0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
    0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
    0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
    0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
    0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
    0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
    0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
    0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
    0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0,
};

/*-----*/
// @USIGN16 Calc_CRC16(const USIGN8 *buf, const int len)
// @brief 计算2字节16bit的CRC校验值
// @param buf-待计算的数据缓冲区指针
// @param len-待计算的数据长度
// @return 16bit的CRC校验值
/*-----*/
USIGN16 Calc_CRC16(const USIGN8 *buf, const USIGN32 len)
```



```
{  
    USIGN32 i;  
    USIGN16 cksum;  
  
    cksum = 0xffff;  
    for (i = 0; i < len; i++) {  
        cksum = crc16_tab[((cksum>>8) ^ *buf++) & 0xFF] ^ (cksum << 8);  
    }  
    return cksum;  
}
```

```
/*-----The End of File-----*/
```

Note:

The SDK is available, please contact sales@hypersen.com for more information.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Hypersen Technologies Co., Ltd. reserve the right to make changes, corrections, enhancements, modifications, and improvements to Hypersen products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on Hypersen products before placing orders. Hypersen products are sold pursuant to Hypersen's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of Hypersen products and Hypersen assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by Hypersen herein.

Resale of Hypersen products with provisions different from the information set forth herein shall void any warranty granted by Hypersen for such product.

Hypersen and the Hypersen logo are trademarks of Hypersen. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.