用户手册	

SD16W

微型化人体测温模组

Rev. 1.4

浙江大立科技股份有限公司

杭州大立微电子有限公司

版本更新

版本	描述	作者	检查	检查	批准	日期
Rev1.1		LX	PF	YML	JLJ	2020-2-17
Rev1.2	修改了部分描述	LX	PF	YML	JLJ	2020-5-20
Rev1.3	修改了在线升级指令	LX	PF	YML	JLJ	2020-6-10
Rev1.4	文字勘误，补充说明功能	LX	PF	YML	JLJ	2020-8-20

浙江大立科技股份有限公司/杭州大立微电子有限公司 发布

地址：浙江省杭州市滨江区滨康路639号

目录

1. 产品介绍

SD16W 微型化人体测温模组，是基于本公司自主非制冷红外阵列传感器研发生产的微型化红外成像人体测温模组产品。

模组每秒输出不高于 15 帧（15Hz）全幅红外温度数据（灰度值），通过简单转换即可得到每个像素的温度值（℃），易于操作。

模组具备 8Mbps 高速串口与 SPI 两种输出接口，采用通用串口进行控制和通信，便于模组与其他主控设备的集成。

模组工作在高速串口输出模式时，通信控制与数据输出均通过 8Mbps 的高速串口完成；模组工作在 SPI 输出模式时，通信控制通过 115200bps 的串口完成，数据输出通过 SPI 完成。

模组具有测温精度高、重量轻、体积小、功耗低、性价比高等优势，适合集成应用于人体测温仪、AI 人脸识别测温、测温闸机、测温安检门、测温考勤机、测温门禁对讲等设备。

图 1：SD16W 模组外观图

SD16W 模组特点优势：

- 全幅人体测温：每帧图像提供 160×120 个温度点数据，便于用户后处理
- 高帧频：≤15Hz 帧频，肉眼观察接近实时成像
- 应用灵活：用户可进行后端图像拉伸、伪彩渲染等个性化算法处理
- 使用方便：体积小型化，单根 FFC 线输入输出

2. 技术指标

SD16W 微型化人体测温模组的详细技术指标参数描述如下：

表 1：模组技术指标

探测器类型	非制冷红外阵列传感器
分辨率	160×120
像元间距	17um
主要工作波段	8～14um
探测器 NETD	≤60mK（F/1，300K，50Hz）
帧频	≤15Hz
图像校正	出厂前完成单点、两点、死点替换等，开机自动校正
输出接口	同时具备高速串口与 SPI 接口，可选配串口-USB 转接板
控制接口	RS232(3.3V)
测温范围	20℃～50℃
测温稳定性	开机稳定后±1℃
测温精度（室温）	用户算法标定后±0.5℃，配合黑体±0.3℃
测温数据输出	全幅温度输出
工作电压	DC 5V±0.1V
功耗	≤0.5W
工作温度	10℃～45℃
整体尺寸	W24.5mm×H32.5mm×D20.7mm
安装方式	通过 PCB 上对角 2 个 φ1.2mm 通孔，用 ST1.4 螺丝安装
标配镜头	无热化 f3.85mm/F1.0 镜头
视场角（计算值）	40° × 30°
PC 演示转接板	可选配一块 USB 转接板连接电脑，演示图像、测温等功能
PC 软件界面	提供 PC 软件界面快速查看图像、测温等

3. 机械参数

模组外形尺寸如图 2 所示(单位：mm，图示尺寸包括 3.85mm 镜头)。

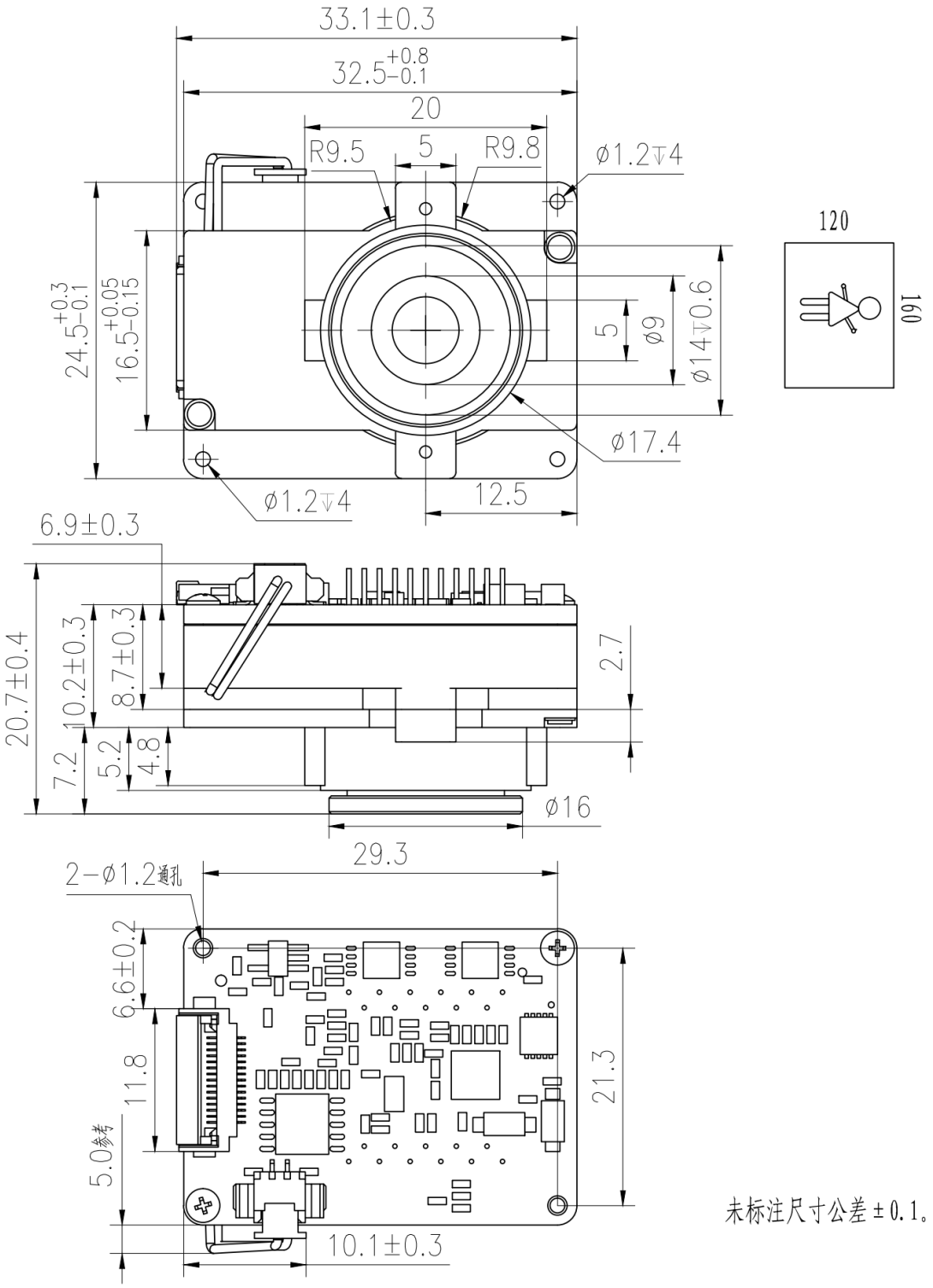
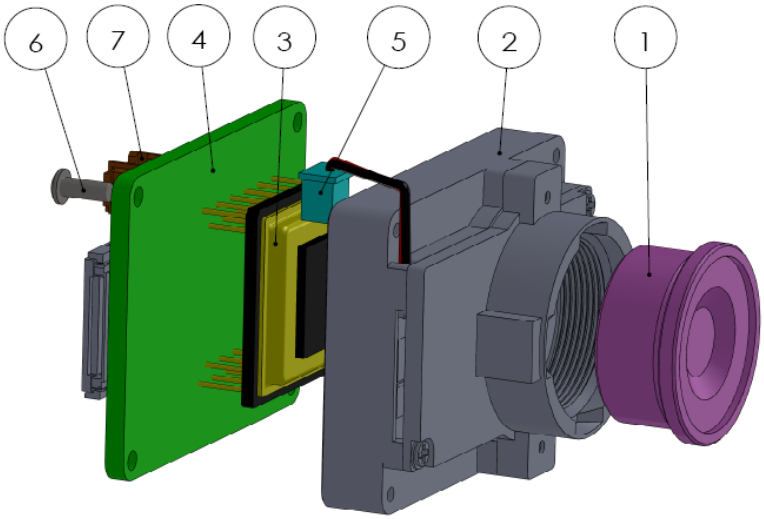


图 2：模组外形尺寸

4. 结构装配

模组的结构装配如下图所示。



序号	名称
1	红外镜头
2	组件外壳
3	探测器
4	主控电路板
5	挡片接口公座
6	电路板锁紧螺丝
7	挡片接口母座

图 3：模组结构装配示意图

5. 电气参数

模组主控电路板（图 3 中的部件④）的接口位置如图 4 所示。

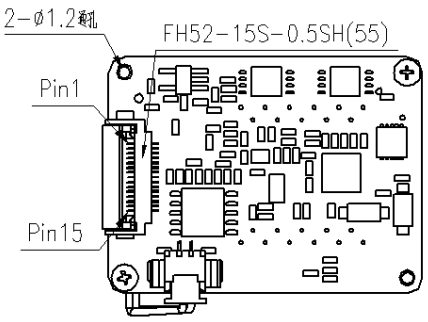


图 4：（a）主控板接口位置　　（b）主控板实物图

主控板接口采用 15Pin FFC 引脚插座，规格为 Hirose 0.5S-1-15PB（型号 FH52-15S-0.5SH(55)），可对接的插座为 0.5S-1-15PB（型号 FH52-15S-0.5SH(55)）。15Pin 的定义如表 2 所示。

电源要求输入范围 5V±0.1V，RMS 噪声小于 50mV，供电电流＞100mA。

表 2：SD16W 组件接口定义

序号	功能	I/O	序号	功能	I/O
1	DC 5V	I	2	DC 5V	I
3	GND	I	4	GND	I
5	USART_TX	O	6	USART_RX	I
7	SPI_NSS	I	8	SPI_MISO	O
9	SPI_INT	O	10	SPI_SCK	I
11	NC	-	12	NC	-
13	NC	-	14	NC	-
15	NC	-			

模组内部 PIN1 与 PIN2 两脚短接，PIN3 与 PIN4 两脚短接。

模组工作在高速串口输出模式或 SPI 输出模式的电气配置方式，见第 7 节和第 8 节。

⚠**主控电路板 FFC 插座使用注意事项：**

连接使用模组及转接板之前，务必仔细阅读附录 2 的注意事项。

6. 软件协议

数据输入输出的基本构成形式为一个数据包。每个数据包包含 8 个部分，每一个部分及其格式如表 3 所示，其中，多字节数据构成中，高字节在前，低字节在后。

表 3：数据包格式

序号	含义	字节数	格式
1	起始字节	2	固定数：0xAA55
2	状态字节	1	0x00
3	指令	1	如表 4 所示
4	数据长度	2	表示序号 6 数据的长度，见表 4“数据长度”列
5	CRC1	2	校验
6	数据	可变	实际数据，其长度可变
7	CRC2	2	校验
8	结束字节	2	固定数：0x0D0A

CRC1 校验：采用 CCITT-16 校验，CRC-CCITT（0xFFFF），即 CRC-16/CCITT-FALSE。

a) 该校验对“序号 1”～“序号 4”这 4 个部分进行校验运算。

b) 该校验在后端形成，在组件端进行检验，如果校验没有通过，则需要重传。

c) 组件向后端发送的数据，不做 CRC1 校验，CRC1=0x0000。

CRC2 校验：采用 CCITT-16 校验，CRC-CCITT（0xFFFF），即 CRC-16/CCITT-FALSE。

a) 该校验对“序号 1”～“序号 6”这 6 个部分进行校验运算。

b) 该校验在后端形成，在组件端进行检验，如校验没有通过，则需要重传。

c) 组件向后端发送的数据，不做 CRC2 校验，该 CRC2=0x0000。

CRC1、CRC2 校验采用的表格如附录 1 所示。

表 4：指令

指令代码	方向	数据	数据长度	说明
0x00	串口下行	0x00：通知组件开始传输数据 示例：AA 55 00 00 00 01 4F 7E 00 00 00 0D 0A 0xFF：通知组件停止传输数据 示例：AA 55 00 00 00 01 4F 7E FF 1E F0 0D 0A	1	开始/停止，开机只需发送一次
	串口上行	0-9：保留 10-19：识别编号（**加黑部分**，以 ASCII 码解析） 20-39：保留 40-49：固件版本（蓝色部分，以 ASCII 码解析） 示例：AA 55 00 00 00 32 00 00 44 4D 43 31 32 30 00 00 00 00 **50 31 35 31 30 2D 30 39 00 00** 56 31 2E 30 00 00 00 00 00 00 4D 43 31 32 30 41 00 00 00 00 41 31 36 31 31 31 34 30 31 00 42 30 31 2E 30 32 00 00 00 00 00 00 0D 0A	60	组件版本信息 0x00 的应答。注意应答数据只在帧结尾处才上传，且发送一次 0x00 串口下行，只上传应答一次
0x03	串口下行	0x00:打开单点校正（手动校正） 示例：AA 55 00 03 00 01 16 2E 00 00 00 0D 0A 0xFF:关闭单点校正（默认值） 示例：AA 55 00 03 00 01 16 2E FF 1E F0 0D 0A	1	手动单点校正，见注 1
0x04	串口下行	0x00:自动控制快门（默认值） 示例：AA 55 00 04 00 01 93 BE 00 00 00 0D 0A 0xFF:关闭自动控制快门 示例：AA 55 00 04 00 01 93 BE FF 1E F0 0D 0A	1	自动控制快门，见注 1
0x05	串口下行	0x00：打开两点校正（默认值） 示例：AA 55 00 05 00 01 A4 8E 00 00 00 0D 0A 0xFF：关闭两点校正 示例：AA 55 00 05 00 01 A4 8E FF 1E F0 0D 0A	1	两点校正，见注 1
0x09	串口下行	0x00：模组通过 0x25 上传全幅温度数据（SPI 时默认） 温度数据示例：AA 55 00 09 00 01 D1 EF 00 00 00 0D 0A 0xFF：模组通过 0x24 上传全幅灰度数据（高速串口时默认） 灰度数据示例：AA 55 00 09 00 01 D1 EF FF 1E	1	

		F0 0D 0A		
0x0D	SPI/串口上行	0-1：探测器内部温度的灰度值（**加黑部分**） 2-7：保留 示例：AA 55 00 0D 00 08 00 00 **12 34** 56 78 9A BC DE F0 00 00 0D 0A	8	组件在每帧的帧尾返回探测器内部温度，见注 2
0x0E	SPI/串口上行	0：最高温横坐标 1：最高温纵坐标 2-3：最高温温度值 4：中心点横坐标 5：中心点纵坐标 6-7：中心点温度值 8：最低温横坐标 9：最低温纵坐标 10-11：最低温温度值 12：任意点横坐标 13：任意点纵坐标 14-15：任意点温度值 示例：AA 55 00 0E 00 10 00 00 0E 49 05 8A 3C 3C 05 46 03 38 05 18 03 38 05 18 00 00 0D 0A	16	组件在每帧的帧尾返回四个点的坐标和温度值，见注 2
0x10	串口下行	0x00：模组测温数据开启人体补偿模式。开启后固件版本会自动增加字符“B”，需重发 0x00 指令刷新 示例：AA 55 00 10 00 01 0C 1D 00 00 00 0D 0A 0xFF：模组测温数据关闭人体补偿模式 示例：AA 55 00 10 00 01 0C 1D FF 1E F0 0D 0A	1	开启/关闭状态，在下一次打快门后起效，会自动保存在模组里。
0x20	串口下行	0x0E 中任意点坐标的设定 0：任意点的横坐标，示例中为 19H=25 1：任意点的纵坐标，示例中为 64H=100 示例：AA 55 00 20 00 02 F9 DB 19 64 95 C9 0D 0A	2	坐标范围为（0,0）～（159,119）
0x21	串口下行	温度偏移量校正：配置值（4 字节浮点型） 35℃校正示例：AA 55 00 21 00 04 AE 2D 00 00 0C 42 2D EB 0D 0A	4	温度偏移量校正，方法见注 3
0x22	串口下行	0：死点的横坐标 1：死点的纵坐标 示例：AA 55 00 22 00 02 97 BB 5F 3D F9 7F 0D 0A	2	标记死点 坐标范围为（0,0）～（159,119）
0x23	串口下行	0:0xFF 死点保存到 Flash。 示例：AA 55 00 23 00 01 90 E8 FF 1E F0 0D 0A	1	保存标记的死点表
0x24	SPI/串口上行	当前行号（1 字节）+像素灰度值（2 字节×160）。 行号值为 0～119 每个像素点的灰度值为 2 个字节，高字节在前，低字节在后。有效数据 14bit，最高两位为 0 示例：AA 55 00 24 01 41 00 00 00 14 3B 14 36 14 38 … … … … 14 41 00 00 0D 0A	321	图像灰度数据传输

0x25	SPI/串口上行	当前行号（1 字节）+像素温度值（2 字节×160）。 行号值为 0～119 每个像素点的温度值为 2 个字节，高字节在前，低字节在后。有效数据 14bit，最高两位为 0 示例：AA 55 00 25 01 41 00 00 00 0B E3 0B E6 0B E3 … … … … 0B EE 00 00 0D 0A	321	温度数据传输 见注 4
0xF1	串口下行	0x00:组件复位重启 示例：AA 55 00 F1 00 01 9B 1C 00 00 00 0D 0A	1	
0xA0	串口下行	0：0xFF:进入固件升级状态 1～4：固件大小（Byte） 示例：AA 55 00 A0 00 05 B2 66 FF 00 00 57 4C 44 CF 0D 0A 示例中 00 00 57 4C 对应为 22348Byte		注 5
0xA1	串口下行、上行	上行，收到 A0 的指令或 A1 的数据后，会上行返回 A1 指令 示例：AA 55 00 A1 00 00 00 00 00 00 0D 0A 下行：发送固件数据 示例：AA 55 00 A1 00 80 44 7B XX XX … … XX XX YY YY 0D 0A 每个数据包（80H）128Byte 下发，XX 长 128Byte，YY 是 CRC2 校验码。按顺序依次把固件文件按字节拆分成多个包发完，最后一个包长度不足 128Byte 时，按实际的包长度设置和发送数据。每下发一个 A1 数据包，模组都会上行一个 A1，若未上行 A1，重发当前包数据。所有 A1 数据发完后，会返回 A2 指令。		
0xA2	串口上行	固件数据发送完成后，模组返回 A2 值 示例：AA 55 00 A2 00 00 00 00 00 00 0D 0A		

注 1 0x03、0x04、0x05 指令的说明

0x03、0x04 单点校正，单点校正为输出图像非均匀性校正，过程为快门自动开关一次，模组自动做图像非均匀性校正和温度修正。正常情况下，模组是自动做单点校正的，快门响一声就进行一次单点校正。

0x05，两点校正，为图像灵敏度差异归一化处理。

没有特殊要求时，不建议用户控制 0x03、0x04、0x05 指令。

模组快门的用户控制及单点校正

默认状态下，快门由模组自动控制触发，快门每开关一次完成一次单点校正。如果用户想干预控制快门及单点校正，可按以下方法配置指令。

1）在默认状态下，发送 0x03 配置 0x00：触发快门开关一次，并完成一次图像单点校正；完成该次校正后模组恢复默认状态下快门自动控制。

2）在默认状态下，发送 0x04 配置 0xFF，关闭模组快门自动控制，再发送 0x03 配置 0x00，触发快门开关一次，并完成一次图像单点校正；每 0x03 配置一次 0x00，模组都开关一次快门并完成一次图像单点校正，实现用户对快门及单点校正的控制。在此状态下，发送 0x04 配置 0x00，模组恢复默认状态。

注 2 探测器内部温度和四点目标温度

探测器内部温度 Ts（℃）,与组件 0x0D 返回的温度量 Vs（灰度）的换算关系为：

$$Ts = 190.64 - 0.02164 \times Vs$$

探测目标的温度测量值 Tc（℃），与组件 0x0E 返回的温度量 Vc（灰度）的换算关系为：

$$Tc = Vc/10 - 100;$$

注 3 温度偏移量校正

组件中心点，在 1.2 米左右对准黑体（30℃~40℃），将黑体的温度值以浮点型通过 0x21 发送给模组。例如组件中心点对准 35℃黑体，需将 35.0 作为浮点型写入 0x21 指令的数据位发送。

0x21 的温度校正功能是永久有效的，修改后保存在模组里，断电不会消失，并且无法恢复到出厂设置值，请慎重使用。一般只有在测温值有较大幅度偏离时才用此功能来修正，用户的算法处理等建议在后端开发的上位机上进行。

此处功能只能对应中心点，不能设置任意点。

注 4 温度数据流

每个像素由两个字节组成，高字节在前，低字节在后。

探测目标的温度测量值 Tc（℃），与组件 0x25 返回的温度量 Vc（灰度）的换算关系为：

$$Tc = Vc/10 - 273;$$

注 5 固件升级指令

通过串口指令下发固件数据进行升级。过程为：

A、 先发 A0 下行指令，通知模组进入固件升级状态，并告知固件文件的大小（Byte），模组应答 A1 指令反馈已准备好；

B、 将固件大小按每次 128Byte,拆分成多个包,A1 指令依次下发所有的固件数据,注意 CRC2 需计算正确，模组每收到一个 A1 的固件数据包，都会返回一个 A1 的应答指令，若无 A1 应答需重发当前包。最后一个数据包可能长度不足 128Byte，按实际数据长度发送。注意，A0 指令发完后，延后 1 秒钟采开始发 A1 的指令包。

C、 最后一个包发送完成后，模组会自动返回 A2 指令告知已完成。模组会根据 A0 的文件长

度，自动计算包的个数并核对。

输出数据速率

每秒上传 15 帧数据，每帧数据的大小，至少为：

39974 Bytes = ((8+1+160×2+4)×120+14)Bytes

上述长度计算公式中：

- 8 为数据包头
- 1 为行号
- 160 为每行 160 个像素点
- ×2 为每个像素点 2Byte（高位在前，低位在后）
- 4 为包尾
- 120 为 120 行
- 14 为第 121 行的字节长度，最小 14Byte

每帧都是先上传 120 行图像灰度或温度数据，再上传 121 行的指令包。

一帧图像灰度/温度数据(120 行)，数据如下表：

Head		Status	CMD	Length		CRC1		Line	160*2Bytes	CRC2		End		数据长度	行号
AA	55	00	24/25	01	41	00	00	00	…	00	00	0D	0A	333Bytes	0
AA	55	00	24/25	01	41	00	00	01	…	00	00	0D	0A	333Bytes	1
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
AA	55	00	24	01	41	00	00	77	…	00	00	0D	0A	333Bytes	119

第 121 行包括下表一条或多条指令，至少有一条指令，最短的长度是 14Byte。

Head		Status	CMD	Length		CRC1		Data	CRC2		End		数据长度
AA	55	00	00	00	32	00	00	…	43	31	0D	0A	72Bytes
AA	55	00	0D	00	0C	00	00	…	00	00	0D	0A	20Bytes
AA	55	00	0E	00	0C	00	00	…	00	00	0D	0A	28Bytes
AA	55	00	FE	00	02	00	00	…	00	00	0D	0A	14Bytes

为避免数据传输等异常导致丢包或部分数据包损坏，建议集成开发时按行（数据包）缓存，检测到丢包或包损坏时，舍弃丢失或损坏的包，采用前一帧的该行数据。

0x24 或 0x25 获得的 14 位数据，转换为热像图时，需先做自动拉伸增强（比如直方图

均衡等算法），然后映射到 8bit（比如舍弃掉低 6 位），再查伪彩映射表，一个灰度对应一种颜色，显示出来。图像缩放时，一般是插值把像素增大，再做边缘的抗锯齿处理。

7. 高速串口输出模式

通过将 SPI_NSS 接地，可设置 SD16W 工作在高速串口输出模式。

1）电气配置

模组工作在高速串口输出时,需 SPI_NSS 接地,接入电源和地,接入 USART_RX 和 USART_TX 串口，串口波特率 8000000bps，8bit 数据位，1bit 停止位，无奇偶校验位，RS232 模式，逻辑电平 TTL3.3V，模组其余所有脚均悬空。

高速串口模式时，串口波特率只能是 8000000bps，不支持其他波特率。高速串口是全双工的。

2）传输协议

传输协议按第 6 节内容。

高速串口输出模式下，表 4 中所有数据上传，都通过高速串口上传。

高速串口模式下，默认通过 0x24 上传全幅 15Hz 的图像灰度数据，若需要上传 15Hz 的全幅温度数据（通过 0x25 上传），需发送 0x09 指令切换。

3）数据通信流程

将 SPI_NSS 始终接地，通电后必须通过串口下发 0x00 通知模组开始工作，否则模组无任何输出。

串口下发 0x00 后，模组开始通过高速串口一直连续上传数据。

（1）串口 0x00 指令下发后，先上传 0x24 指令包（图像灰度），共 120 个包（120 行），紧接着从第 121 个包上传 0x00 指令应答包。

（2）接下来每帧均先上传 0x24 指令包（图像灰度），共 120 个包，紧接着从第 121 个包开始上传 0x0D 和 0x0E 指令包。后续一直重复该过程。

（3）发送 0x09 指令配置 0x00，切换到 0x25 上传全幅温度数据，则每帧均先上传 0x25 指令包（温度数据），共 120 个包，紧接着从第 121 个包上传 0x0D 和 0x0E 指令包。后续一直重复该过程。

8. SPI 输出模式

通过将 SPI_NSS 接高电平 3.3V，可设置 SD16W 工作在 SPI 输出模式。

1）电气配置

模组工作在 SPI 输出时，需将 SPI_NSS 始终接高电平 3.3V。

表 2 中所有非 NC 脚都需连接，其中，SPI_NSS，SPI_MISO、SPI_INT、SPI_SCK 为 SPI 接口信号，逻辑电平为 TTL3.3V。SPI_INT 为模组发出的中断信号，SPI 发送有效数据时，SPI_INT 输出高电平；数据发送完成后 SPI_INT 输出低电平。

USART_RX 和 USART_TX 为串口，RS232 模式，逻辑电平 TTL3.3V，串口波特率为 115200，8bit 数据位，1bit 停止位，无奇偶校验位。

模组设定为从设备，SPI 传输速度最大为 45Mbps，最小不低于 12Mbps。数据宽度为 8bit，空闲极性为低电平，上升沿有效，高位在前。

2）传输协议

传输协议按第 6 节内容。

SPI 输出模式下，表 4 中所有数据上传，除了 0x00 是通过串口上传的外，其他所有上行数据，都通过 SPI 上传。

SPI 输出模式下，默认通过 0x25 上传全幅 15Hz 温度数据，若需要上传 15Hz 的全幅图像灰度数据（通过 0x24 上传），需发送 0x09 指令切换。

3）数据通信流程

SD16W 模组与系统之间的红外图像及测温数据传输通过 SPI 接口进行，模组与系统之间的控制、通信通过 RS232 端口进行。

由于 SD16W 通过 SPI 作为从设备传输数据，存在上电时 SPI_SCK 被干扰而导致数据错位的风险。为了避免此风险，设备的启动流程如下：

（1） 系统上电；

（2） 组件上电；

（3） 拉高 SPI_NSS 信号，可与（2）同步操作；

（4） 发送 0xF1 指令重启设备，消除因干扰而可能导致的数据错位问题；

（5） 等待 1s 后发送 0x00 指令；

（6） 等待 SPI_INT 上升沿；

（7） 启动 SPI_SCK 接收数据；

流程图如图 5 所示

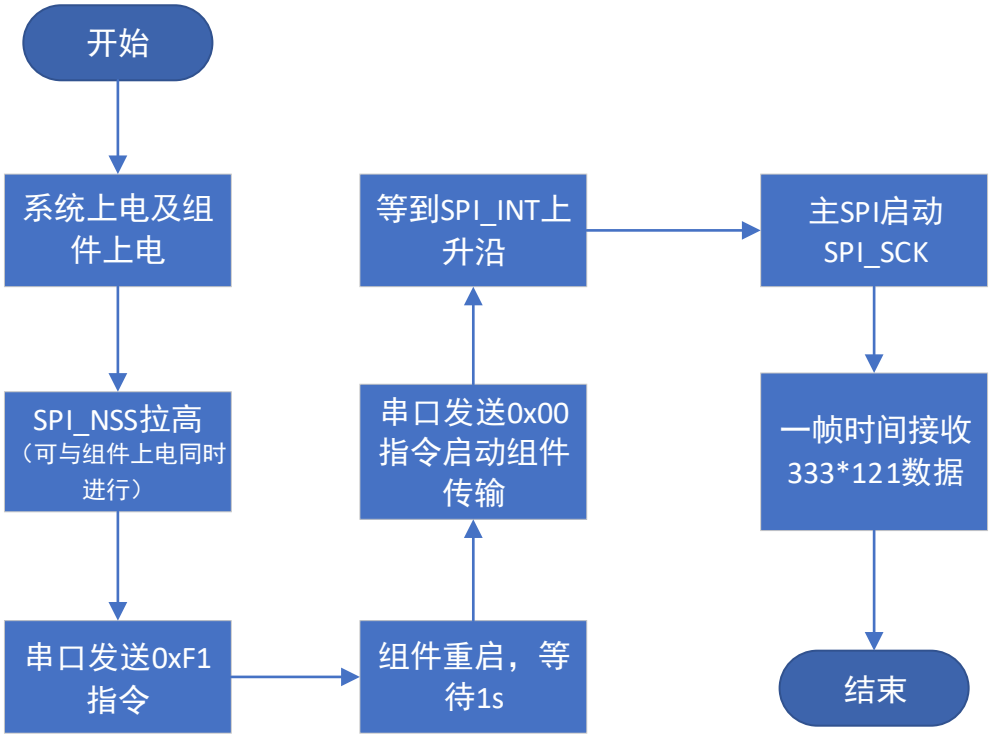


图 5 启动流程

注：一帧的时间（66.7ms/帧）需要接收一行（333 个字节）×行数（120 行）+返回指令，因此建议一帧的时间接收数据的字节数需大于或等于 333×121 个字节。

模组工作时序图如下，

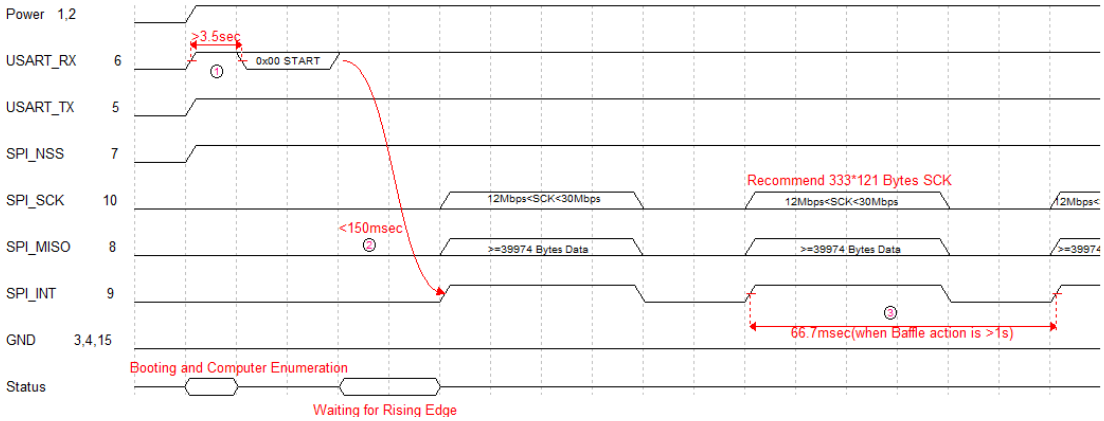


图 6 SD16W 启动工作时序图

注意事项：

（1）SPI_NSS 必须为高电平。

（2）SPI_INT 为高电平时，模组从 SPI 接口输出数据，数据输出完成后 SPI_INT 变为低电平。串口 UART_Rx 发送初始化指令 0x00 后小于 150ms 会出现 SPI_INT 上升沿；

（3）SPI_INT 上升沿间隔(周期)约 66.7ms，但快门自动校正时，SPI_INT 的周期可能

会大于 1s；

（4）SPI_SCK 推荐发送 333×121 Bytes 的时钟个数；

（5）SPI_MISO 包含至少 39974 Bytes 数据（说明见第 6 节中输出数据速率部分）；

（6）USART 速率为 115200bps；

（7）① 是模组输出正常图像时间，如果通过电脑发送起始指令(如 USB)，需要注意电脑枚举时间；

（8）0x00 起始指令如果和第一帧传输有冲突，可能会存在第一帧数据不传输的情况，如下面时序。

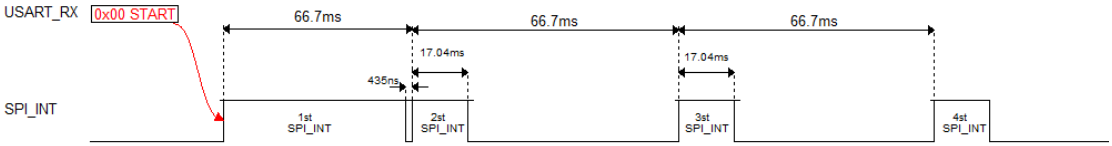


图 7 第一帧冲突时时序图

只要串口下发 0x00 通知模组开始传输数据，SPI 就开始一直连续上传数据。

（1）串口 0x00 指令下发后，SPI 先上传 0x25 指令包（温度数据），共 120 个包（120 行），紧接着从第 121 个包上传 0x00 指令应答包。

（2）接下来每帧均先上传 0x25 指令包（温度数据），共 120 个包，紧接着从第 121 个包上传 0x0D 和 0x0E 指令包。后续一直重复该过程。

（3）发送 0x09 指令配置 0xFF，切换到 0x24 上传图像灰度数据，则每帧均先上传 0x24 指令包（图像灰度），共 120 个包，紧接着从第 121 个包上传 0x0D 和 0x0E 指令包。后续一直重复该过程。

9. 模组使用注意事项

1） 严格按照产品规定的使用条件使用，系统集成使用时，需加强散热和保证模组周围温度均匀，建议以金属材料包裹模组四周，并与其他热源隔离开。如

果系统发热温升高，会影响测温精度。

2） 注意保护镜头表面，避免硬物划伤、油渍沾污。

3） 操作软排线的接口座时应按照使用手册附录中的提示，采用正确的方式，不要用力过大，防止造成盖板掉落或卡扣断裂。FFC 插座损坏不属于质保范围。

4） 模组不要直视太阳等高温目标，防止灼烧探测器。

（以下页为附录 1、附录 2）

附录 1：

CRC 校验表

序号	CRC 值	序号	CRC 值	序号	CRC 值	序号	CRC 值
0x00	0x0000	0x40	0x48c4	0x80	0x9188	0xc0	0xd94c
0x01	0x1021	0x41	0x58e5	0x81	0x81a9	0xc1	0xc96d
0x02	0x2042	0x42	0x6886	0x82	0xb1ca	0xc2	0xf90e
0x03	0x3063	0x43	0x78a7	0x83	0xa1eb	0xc3	0xe92f
0x04	0x4084	0x44	0x0840	0x84	0xd10c	0xc4	0x99c8
0x05	0x50a5	0x45	0x1861	0x85	0xc12d	0xc5	0x89e9
0x06	0x60c6	0x46	0x2802	0x86	0xf14e	0xc6	0xb98a
0x07	0x70e7	0x47	0x3823	0x87	0xe16f	0xc7	0xa9ab
0x08	0x8108	0x48	0xc9cc	0x88	0x1080	0xc8	0x5844
0x09	0x9129	0x49	0xd9ed	0x89	0x00a1	0xc9	0x4865
0x0a	0xa14a	0x4a	0xe98e	0x8a	0x30c2	0xca	0x7806
0x0b	0xb16b	0x4b	0xf9af	0x8b	0x20e3	0xcb	0x6827
0x0c	0xc18c	0x4c	0x8948	0x8c	0x5004	0xcc	0x18c0
0x0d	0xd1ad	0x4d	0x9969	0x8d	0x4025	0xcd	0x08e1
0x0e	0xe1ce	0x4e	0xa90a	0x8e	0x7046	0xce	0x3882
0x0f	0xf1ef	0x4f	0xb92b	0x8f	0x6067	0xcf	0x28a3
0x10	0x1231	0x50	0x5af5	0x90	0x83b9	0xd0	0xcb7d
0x11	0x0210	0x51	0x4ad4	0x91	0x9398	0xd1	0xdb5c
0x12	0x3273	0x52	0x7ab7	0x92	0xa3fb	0xd2	0xeb3f
0x13	0x2252	0x53	0x6a96	0x93	0xb3da	0xd3	0xfb1e
0x14	0x52b5	0x54	0x1a71	0x94	0xc33d	0xd4	0x8bf9
0x15	0x4294	0x55	0x0a50	0x95	0xd31c	0xd5	0x9bd8
0x16	0x72f7	0x56	0x3a33	0x96	0xe37f	0xd6	0xabbb
0x17	0x62d6	0x57	0x2a12	0x97	0xf35e	0xd7	0xbb9a
0x18	0x9339	0x58	0xdbfd	0x98	0x02b1	0xd8	0x4a75
0x19	0x8318	0x59	0xcbdc	0x99	0x1290	0xd9	0x5a54
0x1a	0xb37b	0x5a	0xfbbf	0x9a	0x22f3	0xda	0x6a37
0x1b	0xa35a	0x5b	0xeb9e	0x9b	0x32d2	0xdb	0x7a16
0x1c	0xd3bd	0x5c	0x9b79	0x9c	0x4235	0xdc	0x0af1
0x1d	0xc39c	0x5d	0x8b58	0x9d	0x5214	0xdd	0x1ad0
0x1e	0xf3ff	0x5e	0xbb3b	0x9e	0x6277	0xde	0x2ab3
0x1f	0xe3de	0x5f	0xab1a	0x9f	0x7256	0xdf	0x3a92
0x20	0x2462	0x60	0x6ca6	0xa0	0xb5ea	0xe0	0xfd2e
0x21	0x3443	0x61	0x7c87	0xa1	0xa5cb	0xe1	0xed0f
0x22	0x0420	0x62	0x4ce4	0xa2	0x95a8	0xe2	0xdd6c
0x23	0x1401	0x63	0x5cc5	0xa3	0x8589	0xe3	0xcd4d
0x24	0x64e6	0x64	0x2c22	0xa4	0xf56e	0xe4	0xbdaa

0x25	0x74c7	0x65	0x3c03	0xa5	0xe54f	0xe5	0xad8b
0x26	0x44a4	0x66	0x0c60	0xa6	0xd52c	0xe6	0x9de8
0x27	0x5485	0x67	0x1c41	0xa7	0xc50d	0xe7	0x8dc9
0x28	0xa56a	0x68	0xedae	0xa8	0x34e2	0xe8	0x7c26
0x29	0xb54b	0x69	0xfd8f	0xa9	0x24c3	0xe9	0x6c07
0x2a	0x8528	0x6a	0xcdec	0xaa	0x14a0	0xea	0x5c64
0x2b	0x9509	0x6b	0xddcd	0xab	0x0481	0xeb	0x4c45
0x2c	0xe5ee	0x6c	0xad2a	0xac	0x7466	0xec	0x3ca2
0x2d	0xf5cf	0x6d	0xbd0b	0xad	0x6447	0xed	0x2c83
0x2e	0xc5ac	0x6e	0x8d68	0xae	0x5424	0xee	0x1ce0
0x2f	0xd58d	0x6f	0x9d49	0xaf	0x4405	0xef	0x0cc1
0x30	0x3653	0x70	0x7e97	0xb0	0xa7db	0xf0	0xef1f
0x31	0x2672	0x71	0x6eb6	0xb1	0xb7fa	0xf1	0xff3e
0x32	0x1611	0x72	0x5ed5	0xb2	0x8799	0xf2	0xcf5d
0x33	0x0630	0x73	0x4ef4	0xb3	0x97b8	0xf3	0xdf7c
0x34	0x76d7	0x74	0x3e13	0xb4	0xe75f	0xf4	0xaf9b
0x35	0x66f6	0x75	0x2e32	0xb5	0xf77e	0xf5	0xbfba
0x36	0x5695	0x76	0x1e51	0xb6	0xc71d	0xf6	0x8fd9
0x37	0x46b4	0x77	0x0e70	0xb7	0xd73c	0xf7	0x9ff8
0x38	0xb75b	0x78	0xff9f	0xb8	0x26d3	0xf8	0x6e17
0x39	0xa77a	0x79	0xefbe	0xb9	0x36f2	0xf9	0x7e36
0x3a	0x9719	0x7a	0xdfdd	0xba	0x0691	0xfa	0x4e55
0x3b	0x8738	0x7b	0xcffc	0xbb	0x16b0	0xfb	0x5e74
0x3c	0xf7df	0x7c	0xbf1b	0xbc	0x6657	0xfc	0x2e93
0x3d	0xe7fe	0x7d	0xaf3a	0xbd	0x7676	0xfd	0x3eb2
0x3e	0xd79d	0x7e	0x9f59	0xbe	0x4615	0xfe	0x0ed1
0x3f	0xc7bc	0x7f	0x8f78	0xbf	0x5634	0xff	0x1ef0

CRC 的初值为 0xFFFF。

注：CRC 不是直接查附录 1 的表，是按算法算出来的，附录 1 的表格，是计算过程中调用的表，CRC 的算法源代码中要用到这个表格。

模组的 CRC 校验算法是 CRC-CCITT（0xFFFF），或叫 CRC-16/CCITT-FALSE。

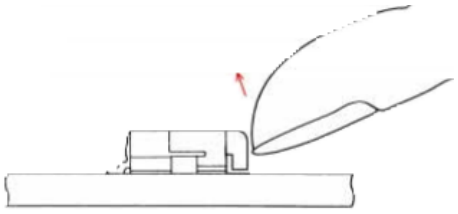
附录 2：

⚠ 主板 FFC 插座使用注意事项：

请连接使用模组及转接板之前，务必仔细阅读以下注意事项和建议：

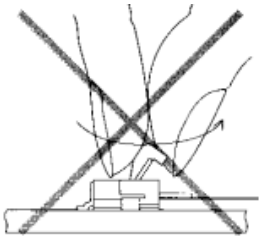
1、安装 FFC

1）打开插座盖板：正确方法用拇指或食指提起盖板，打开模组主控板上的 FFC 连接器。



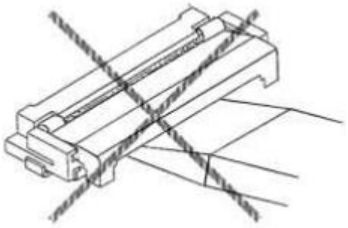
附图 1（a）：提起盖板

注意：避免用两个手指抓住盖板或用指甲提起盖板。



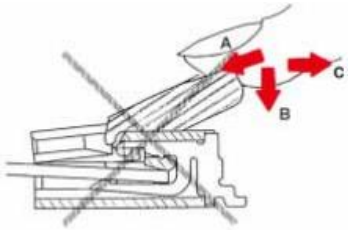
附图 1（b）：错误操作

注意：不要使用薄工具（如螺丝刀头）打开盖板，这样做会导致触点变形。



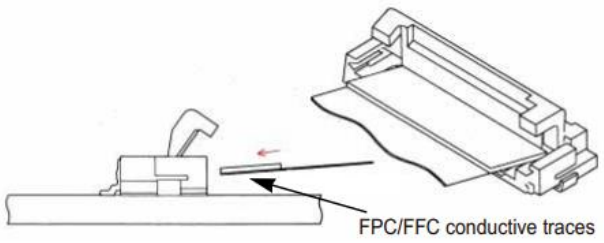
附图 1（c）：错误操作

注意：当盖板打开时，不要向箭头 A、B 或 C 的方向施力，这样做将导致盖板分离或损坏接头处钩子部分。



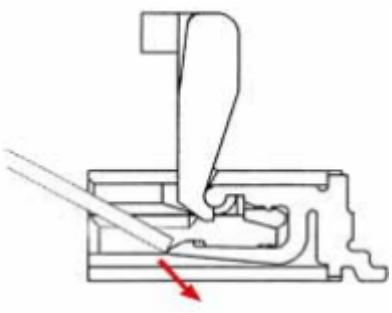
附图 1（d）：错误操作

2）完全插入与安装表面平行的 FFC，注意导电迹线朝下。



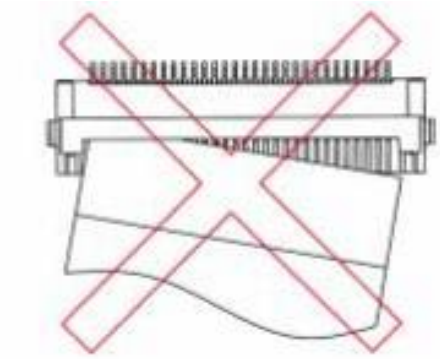
附图 2（a）：将 FFC 插入插座

注意：当插入 FFC 时，不要用力摩擦连接器插入槽下面的表面。这样做将导致柔性线路板用力撞击触点，从而导致接触变形、柔性线路板导线剥落和其他不规则现象。



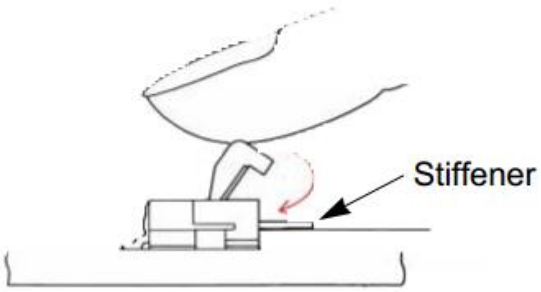
附图 2（b）：错误操作

注意：将 FFC 插入连接器插入槽的适当位置，并相对于连接器垂直插入。不正确的插入将导致连续性问题、盖板松脱和接头处钩子损坏。



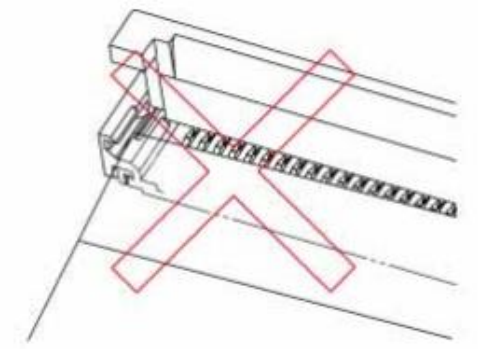
附图 2（c）：错误方式

3）向下旋转盖板，直到牢固关闭。插入的 FFC 不能松动并保证完全插入。如果 FFC 松动，打开盖板并从步骤 1 开始。



附图 3：固定 FFC

注意：FFC 部分插入、偏移插入和对角插入将导致盖板接合不良。若要更正此问题，请先卸下 FFC，然后重新插入。用力锁会损坏接口。

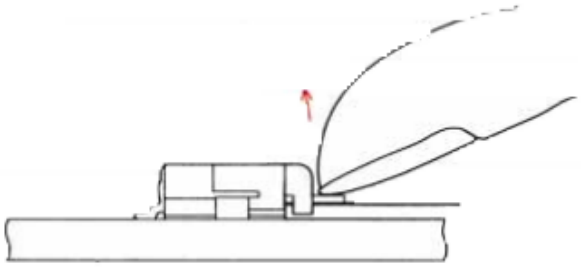


附图 4：错误操作

2、FFC 移除

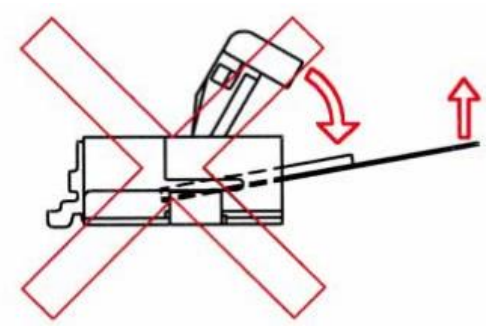
1）按照正确的方法用拇指或食指提起盖板。

2）小心地拆下 FFC。



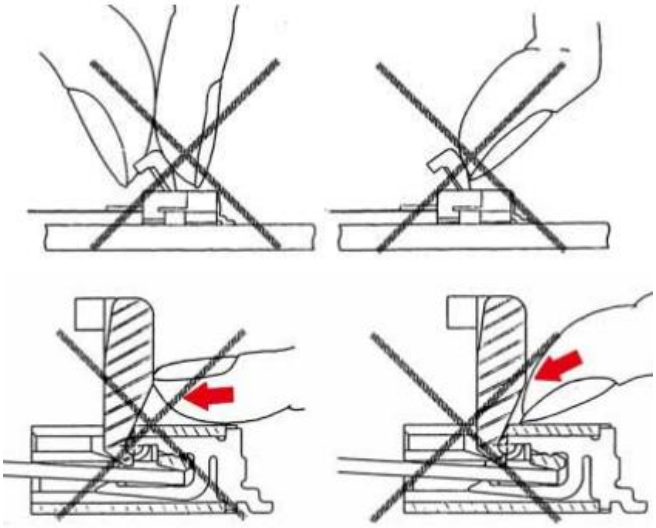
附图 5：FFC 移除

注意：锁定时，移动 FFC，若施加力的方向与盖板的旋转运动方向相反时，将施加过大的负载，这将导致盖板连接器损坏。



附图 6：错误方式

注意：在锁定 FFC 时，施加如下图 7 中箭头所示的力将导致盖板脱落，当盖板脱落时，请更换整个接头。



附图 7：错误方式