用户手册 – XL5300

飞行时间（ToF）测距传感器

前言

XL5300 是一款单模块封装 ToF 传感器，采用了 SPAD、TDC 和直方图技术，可实现最大 4000 mm 的精确距离测量。本用户手册旨在介绍该传感器可能的系统集成模式，以及使用 XL5300 驱动程序获取测距数据所需要调用的功能。

版本信息：V2.1，2021.11.4

目录

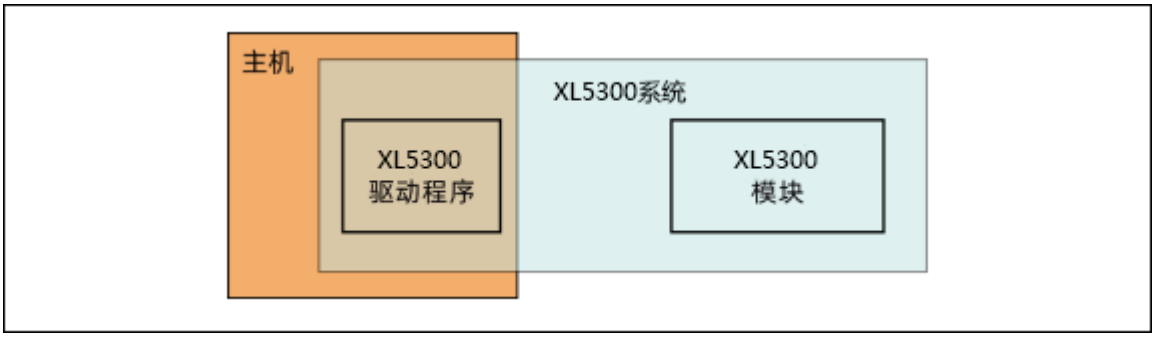
图目录

表目录

1 XL5300 系统概述

XL5300 系统由 XL5300 模块和运行在主机上的驱动程序组成。

图 1：XL5300 系统概览



根据主机是否是 Linux 主机，XL5300 模块有两种可能的集成模式，并且提供相应的驱动程序：

- 非 Linux 主机：提供裸机驱动程序。
- Linux 主机：提供 Linux 驱动程序。

这两种驱动程序都允许主机控制模块，访问测距数据并执行处理和数据结构输出。

本手册描述了 Linux 主机集成模式下主机可使用的驱动程序功能，从而实现对模块的控制并且获取测距数据。

说明
本文档仅描述已实现且已经过验证的功能。 驱动程序中可用的任何其他功能，如果未在本文档中描述，则不应使用。

2 测距功能说明

2.1 测距预设模式

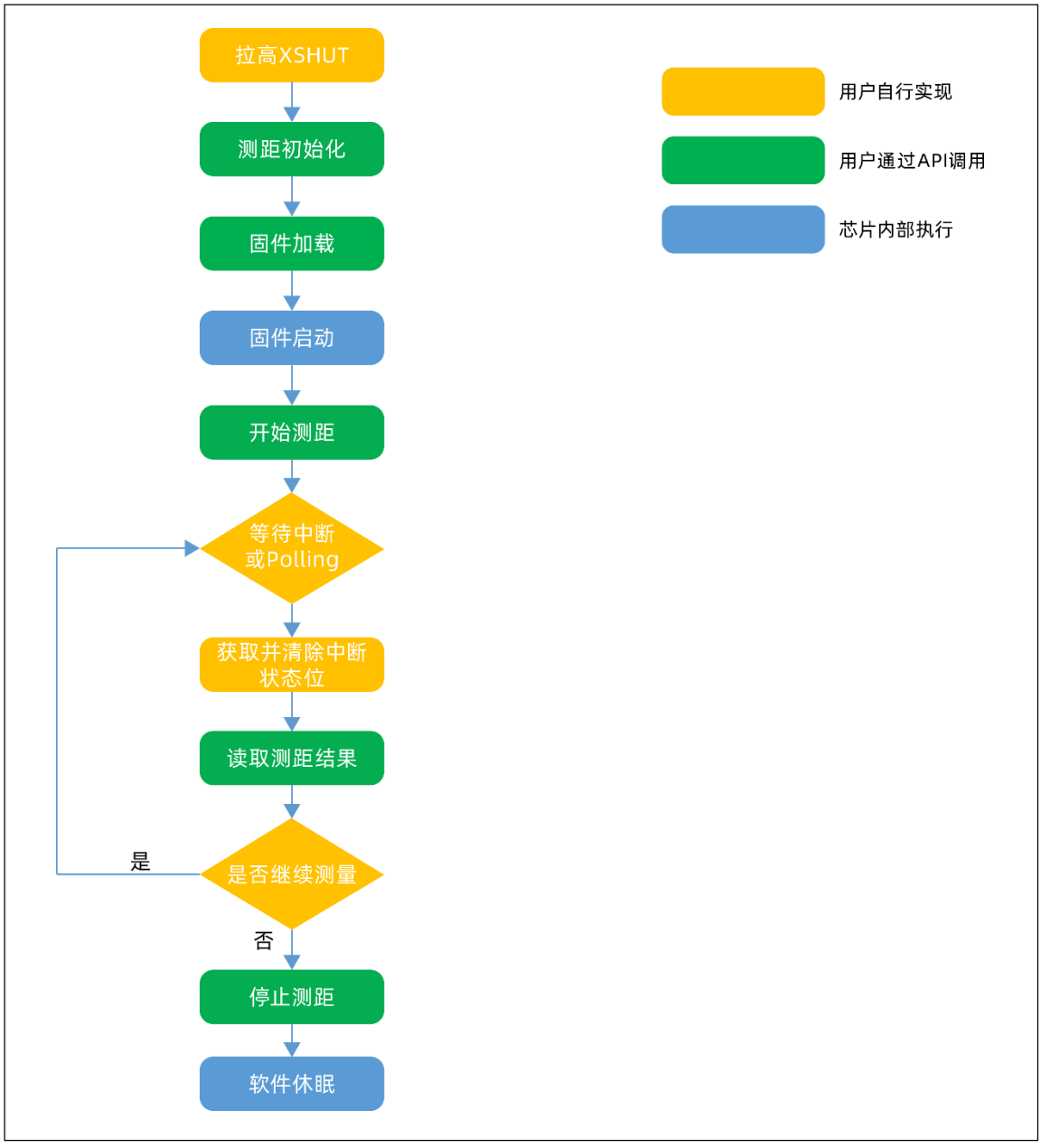
该驱动程序支持单目标测距，旨在基于固件默认模式获取单个目标的探测距离。

2.2 工作流程

该器件是基于标准的中断管理模式运行的。

每次测距之后，系统产生中断，在连续模式下，上一帧的数据不及时取出会被下一帧数据覆盖，单次测距数据完成后数据保存在缓存区，执行下一次单次测距命令才会覆盖上一次的数据。

图 2：XL5300 连续测距模式下工作流程概览



工作流程说明：

1. 用户拉高 XSHUT 引脚，使芯片上电。
2. 用户调用 API 函数实现测距初始化（等待内部 CPU 就绪、使能芯片中断）。
3. 用户下载固件到芯片内部。
4. 芯片固件开始启动，大约持续 10 ms，可以通过确认 0x08 寄存器是否为 0x66 来判断是否启动成功。固件启动后，芯片会自动进入软件休眠模式。
5. 读取 Xtalk 标定参数，并写入芯片。
6. 用户调用 API 函数开始测距，此时芯片内部执行测距命令。
7. 用户等待硬件中断(GPIO pin)发生，或轮询中断寄存器(0x03)。
8. 用户读取硬件中断(GPIO pin)状态寄存器，将中断状态清除。
9. 用户通过 API 读取测距结果。
10. 如果想要继续获取下一次测距结果，则重复上述步骤 7 ~ 9。
11. 如不需要继续测量，则调用 API 函数停止测距。停止测距后如果需要继续测距，可以再次调用“开始测距”API 来进行下一次循环。

3 驱动程序基本功能

本节描述了使用 XL5300 进行测距时需遵循的驱动程序函数调用流程。

XL5300 驱动程序主要用于以下两类应用：

- 用于器件校准的工厂应用，通常用于最终产品生产测试（标定流程）
- 现场应用，指使用了 XL5300 的所有终端用户应用（测距流程）

3.1 驱动程序调用流程

驱动程序的工厂标定流程如下图：

图 3：XL5300 API XTalk 标定流程

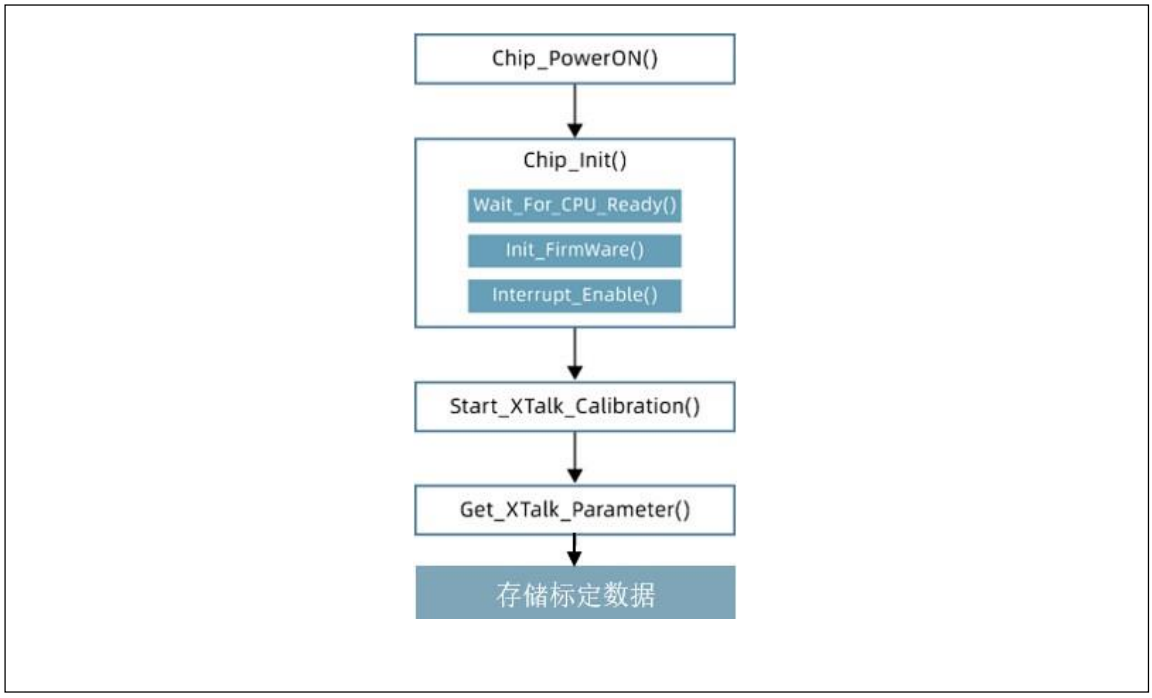
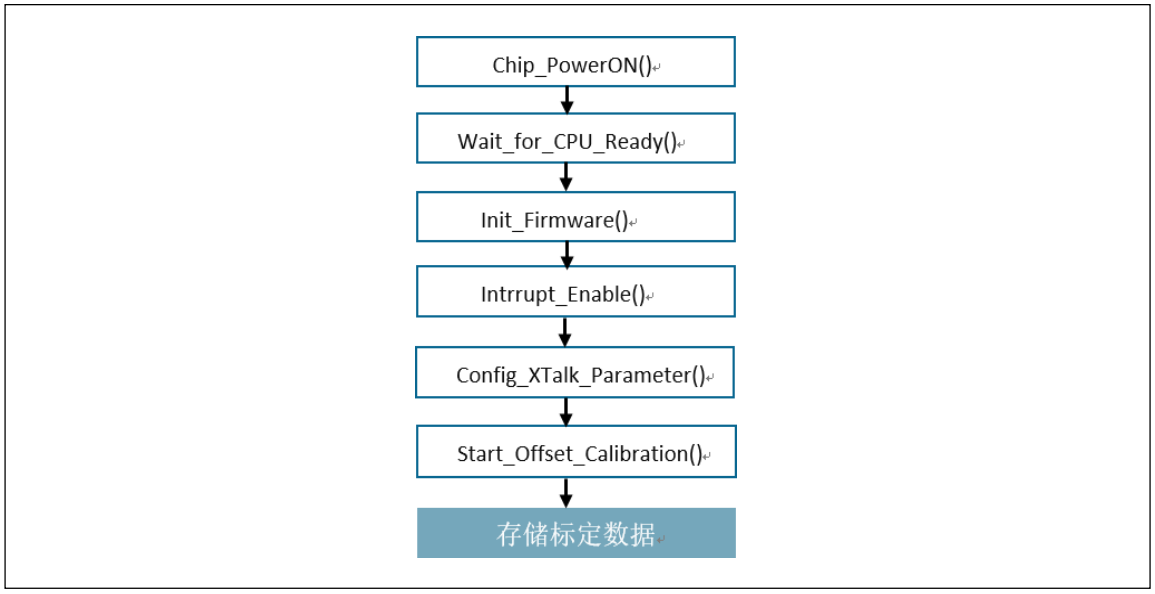
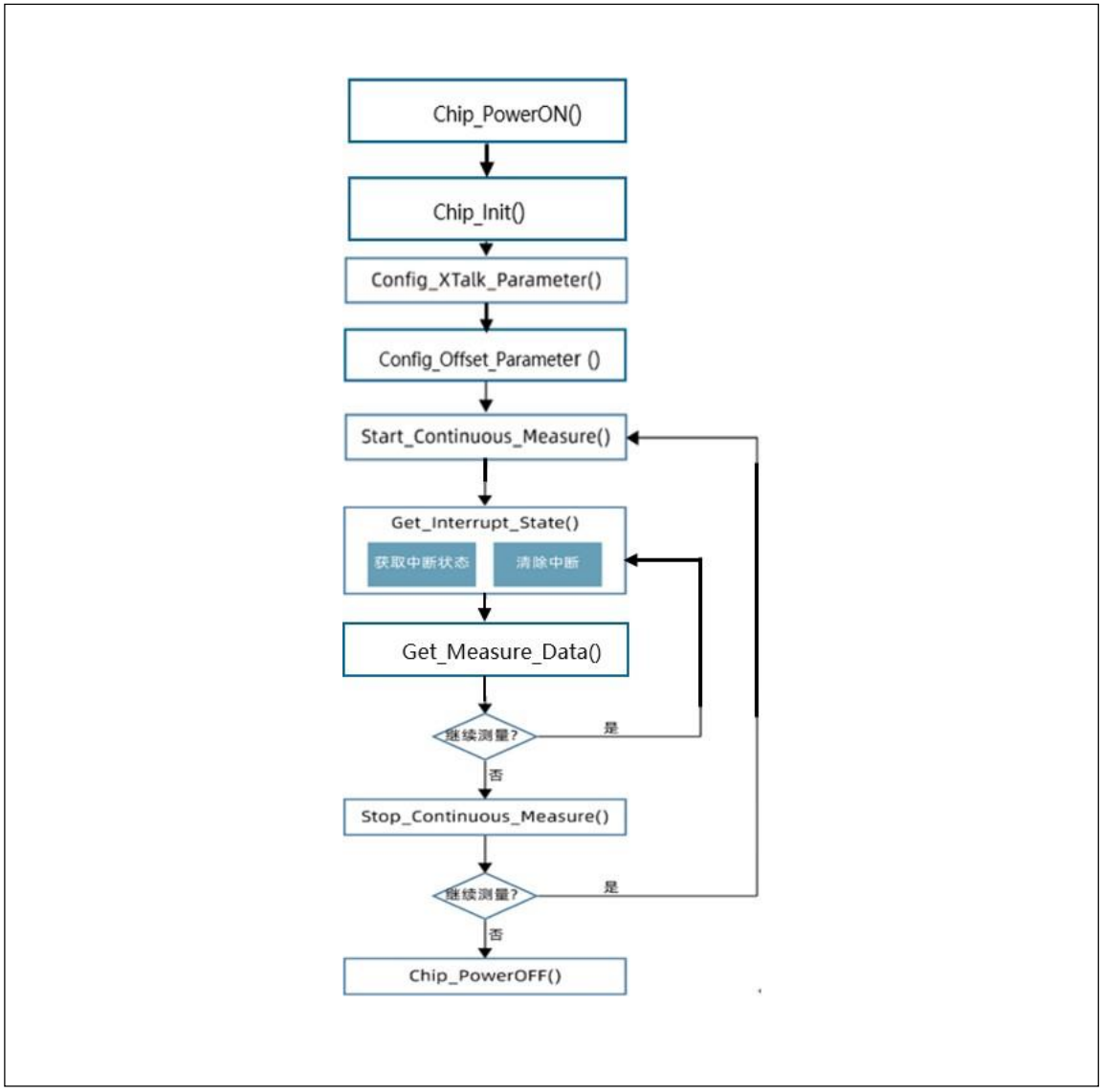


图 4：XL5300 API Offset 标定流程



驱动程序的测距流程如下图：

图 5：XL5300 API 测距流程



3.2 系统初始化

本节描述了在开始测量前执行系统初始化所需的 API 函数调用。

3.2.1 芯片上电

XL5300_Chip_PowerON()函数用于将芯片上电，使芯片处于工作状态。

3.2.2 芯片初始化

芯片初始化

Chip_Init()函数主要包含以下内容
XL5300_Chip_Register_Init()
XL5300_Wait_For_CPU_Ready()
XL5300_Interrupt_Enable()
XL5300_Init_Firmware()
XL5300_Set_Integralcounts_Frame()
以及相关标定值(包含 Xtalk/offset/reftof)的配置

寄存器初始化

XL5300_Chip_Register_Init()函数功能用于芯片上电后对寄存器写值，初始化部分寄存器。

等待启动

XL5300_Wait_For_CPU_Ready()函数功能用于确保器件已启动并准备就绪。

说明
该功能会阻止当前的其他操作并进行内部轮询。在 I^2C 速率为 400 kHz 且每事务处理延迟为 2 ms 的条件下，该功能的阻止时间不会超过 5ms。

使能中断

XL5300_Interrupt_Enable()函数使能中断，防止因中断被屏蔽而接收不到测距完成之后的中断输出，从而导致无法读取测量数据。

固件初始化

XL5300_Init_Firmware()函数被调用一次以执行器件初始化。
该函数在器件复位后需要再次调用。芯片复位，firmware 从芯片存储区丢失。

配置积分次数和帧率

XL5300_Set_Integralcounts_Frame(XL5300_DEV dev, uint8_t fps, uint32_t intecounts)函数用于设置积分次数和帧率，对于不同的应用需求可能会需要积分次数、帧率可配置，此函数正是用于

满足这类需求。帧率 fps 是指每秒测距次数或是测距频率，帧率的倒数即为测距周期，即每两次测距之间时间间隔。积分次数 intecounts 是指每个测距周期内的积分次数，也是每个测距周期发射的光脉冲数。单次积分时间为 146.3ns

测距周期 = 积分时间 + 数据处理时间（1.6ms）+延时

其中测距周期由配置的帧率决定，积分时间由配置的积分次数决定，数据处理时间为固定值，延时是可变的，具体值根据上述公式由测距周期和积分时间反算出来（驱动代码自动实现），再由驱动配置到固件。

某一帧率下可配置的最大积分次数为延时=0 时对应的积分次数

积分次数=(1000/帧率 - 数据处理时间（1.6ms）- 延时)/146.3ns x1000000

例如帧率为 30fps 时，可以配置的最大积分次数如下：

软件延时为 0；

积分次数 =（33.3ms – 1.6ms）/ 146.3ms x 1000000= 216600。

3.2.3 Xtalk 标定与配置参数

XTalk 标定值用于盖板玻璃脏污或串扰校正。XL5300 在每次测距时都会执行实时脏污检测并且应用新的串扰校正值。当满足 60 cm 以内无目标、且环境光无 IR 光时，用户可以调用接口函数 XL5300_Start_XTalk_Calibration（ ）执行 XTalk 标定。在标定过程结束后存储相关参数。芯片在上电 XSHUT 拉高后下载标定参数，随后在测距过程中采集直方图结束后利用标定参数执行串扰校正。在芯片初始化完成后从flash 或者文件里面获取之前Xtalk 标定的标定值(xtalk_datas. xtalk_cal），调用 API 函数将 XTalk 标定值写入芯片内部，

```
struct XL5300_XTALK_Calib_Data
    {int8_t xtalk_cal;
    uint16_t xtalk_peak;
    int16_t xtalk_tof;
} xtalk_datas;
```

xtalk_datas. xtalk_cal 就是需要通过驱动下发给芯片的 XTalk 标定值。

通过调用以下函数可实现 Xtalk 标定以及获取和配置 Xtalk 标定值：

- XL5300_Start_XTalk_Calibration()
- XL5300_Get_XTalk_Parameter()
- XL5300_Config_XTalk_Parameter()

3.2.4 OFFSET 标定与配置参数

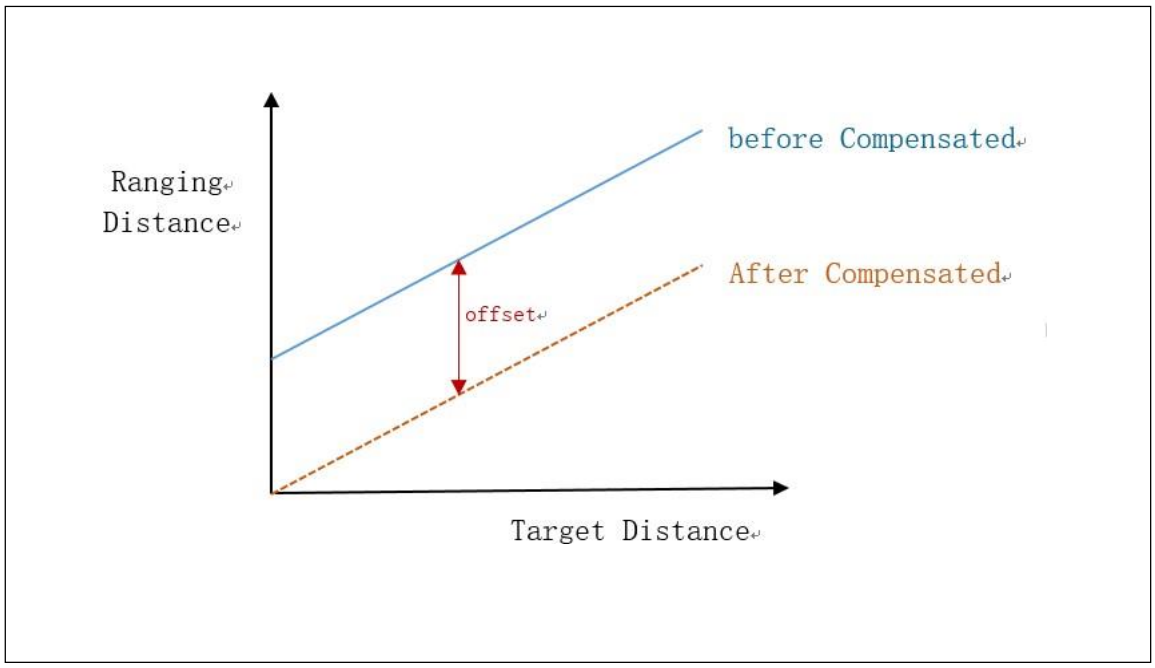
在用户板上贴片或者组装的过程会引入一些误差，导致在测距时存在固定偏移，所以需要做整机的 offset 标定。在测距前方 10cm 位置设置白卡或者灰卡，在芯片完成初始化和 Xtalk 参数配置后，调用 XL5300_Start_offset_Calibration()开始标定 offset，标定值会保存到 flash 或者文件里面。标定后需要在初始化以及 Xtalk 参数配置完成后，从 flash 或者文件里面获取之前 OFFSET 标定的标定值(offset_datas. offset_cal），直接赋值给内部全局变量，用于 offset 校正。其结构体如下

```
struct XL5300_OFFSET_Calib_Data {
```

int16_t offset_cal;
int16_t ref_tof;
} offset_datas;

在 offset 标定时，如果有标定 reftof 功能，则会同时标定。s

图 6：XL5300 offset 标定对比图



3.2.5 REFTOF 标定与配置参数

如果需要 REFTOF 标定，在芯片完成初始化和 Xtalk 以及 offset 参数配置后，从 flash 或者文件里面获取之前 OFFSET 标定的标定值(offset_datas. ref_tof)，reftof 会随温度变化，标定后需要在初始化阶段通过 API 写入 reftof 标定值到芯片内部。

其结构体如下

struct XL5300_OFFSET_Calib_Data
{int16_t offset_cal;
int16_t ref_tof;
} offset_datas;

4 XL5300 测距过程

用户根据实际应用需要、平台能力以及驱动程序的调用序列规则对驱动程序的调用进行顺序分配。

4.1 连续模式开始测量

必须调用 XL5300_Start_Contiunous_Measure()函数来启动测量。

4.2 等待结果：轮询或中断

通过以下方法可以得知是否有可用的测距数据：

- 主机调用轮询功能
- 主机等待硬件中断

4.2.1 驱动程序轮询获取结果状态

通过调用 XL5300_Get_Interrupt_State()函数轮询内部中断状态寄存器标志位获取结果状态，待该标志位置 1 即可读取测距数据。

4.2.2 使用硬件中断

获取测距状态更推荐使用硬件中断输出。默认情况下，当新的测距数据准备就绪时，GPIO1 被拉低。可以调用 XL5300_Get_Interrupt_State()函数清除中断标志位，这一步非必须。

4.3 获取测距数据

硬件会在每个测距回路中采集一组测量数据。通过调用 XL5300_Get_Measure_Data()函数可获取测量数据，同时将返回一个如 4.6 章节所述的数据结构。

4.4 连续模式停止测量

在连续测距模式下，主机可通过调用 XL5300_Stop_Continuous_Measure()函数来停止测量。如果在测距过程中发出停止请求，则立即中止测量。

4.5 单次测距

调用 XL5300_Single_Measure()函数开始测距，单次测距测量一次后自动停止，无需发送停止测距命令。发送命令后等待中断或者轮询中断状态寄存器读取测量数据。

4.6 测距数据结构

数据结构由以下数据组成，每次测距时更新：

- **milimeter**：16 位整数值，给出以毫米为单位的测量距离。
- **confidence**：32 位整数值，表示当前 TOF 值的可信度。
- **status**：8 位整数值，表示当前的测量状态。

4.7 测量结果状态

每次测距过程中的状态以 XL5300_Status 数据结构来表示。

表 1：测距状态说明

值	XL5300_Status 字符串	含义说明
0	XL5300_ERROR_NONE	测距有效
1	XL5300_ERROR_POWER_ON	芯片上电错误
2	XL5300_ERROR_POWER_OFF	芯片下电错误
3	XL5300_ERROR_CPU_BUSY	芯片内部 CPU BUSY 错误
4	XL5300_ERROR_GPIO_ERROR	芯片 GPIO 操作错误
5	XL5300_ERROR_FW_FAILURE	芯片下载 Firmware 失败
6	XL5300_ERROR_INIT_ERROR	芯片初始化失败
7	XL5300_ERROR_XTALK_CALIB	芯片 Xtalk 标定失败
8	XL5300_ERROR_OFFSET_CALIB	芯片 Offset 标定失败
9	XL5300_ERROR_XTALK_CONFIG	配置 Xtalk 标定参数失败
10	XL5300_ERROR_SINGLE_CMD	单次测距命令发送失败
11	XL5300_ERROR_CONTINUOUS_CMD	连续测距命令发送失败
12	XL5300_ERROR_GET_DATA	获取测量数据失败
13	XL5300_ERROR_STOP_CMD	发送停止连续测距命令失败
14	XL5300_ERROR_IRQ_STATE	获取中断标志位失败
15	XL5300_ERROR_ENABLE_INTR	中断使能失败

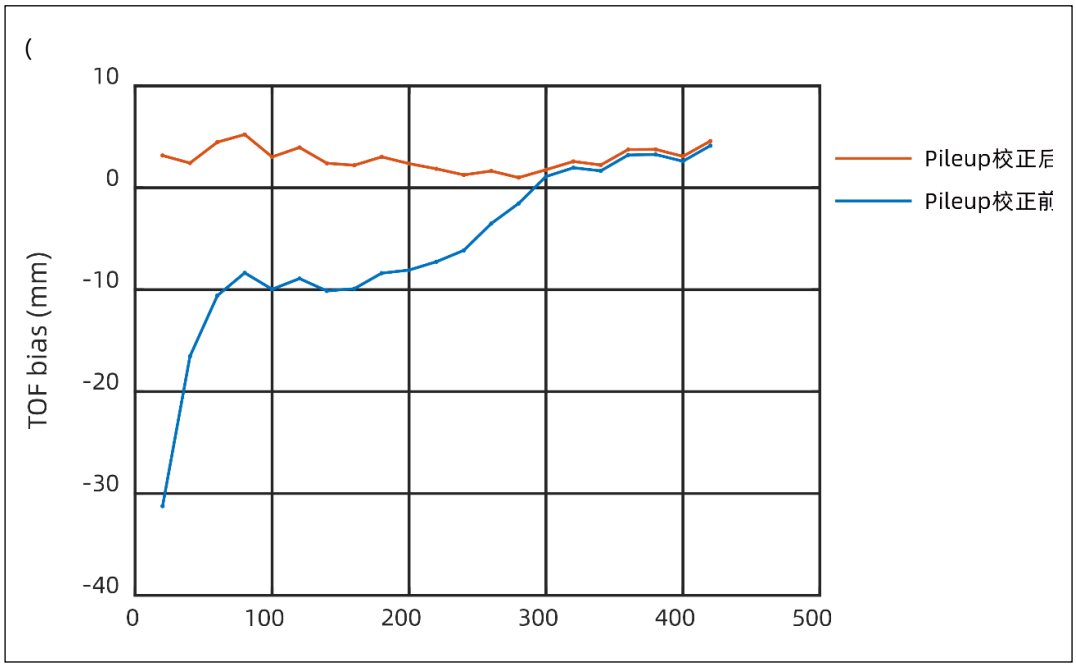
5 驱动程序其他功能

5.1 Pileup 校正

由于 SPAD 器件存在死区时间（Dead time），SPAD 响应首先到达的光子后，对死区时间内到达的第二个光子无法再次响应。因此，对于一定宽度的光脉冲，脉宽时间内到达的光子超过 2 时，只有第一个光子能够触发 SPAD，后续的光子将被屏蔽，从而导致 SPAD 累计得到的信号峰值比光脉冲峰值在时间上更靠前，这个现象称为 pileup 现象。

基于 SPAD 死区时间内光子先到达先触发的原则，测得的 ToF 会随着 Peak 值的增大而偏小，需要通过调整 Peak 值的大小来进行修正。校正 ToF 值的方式：TOF_cal = TOF + OTOF。

图 7：XL5300 pileup 校正前后对比

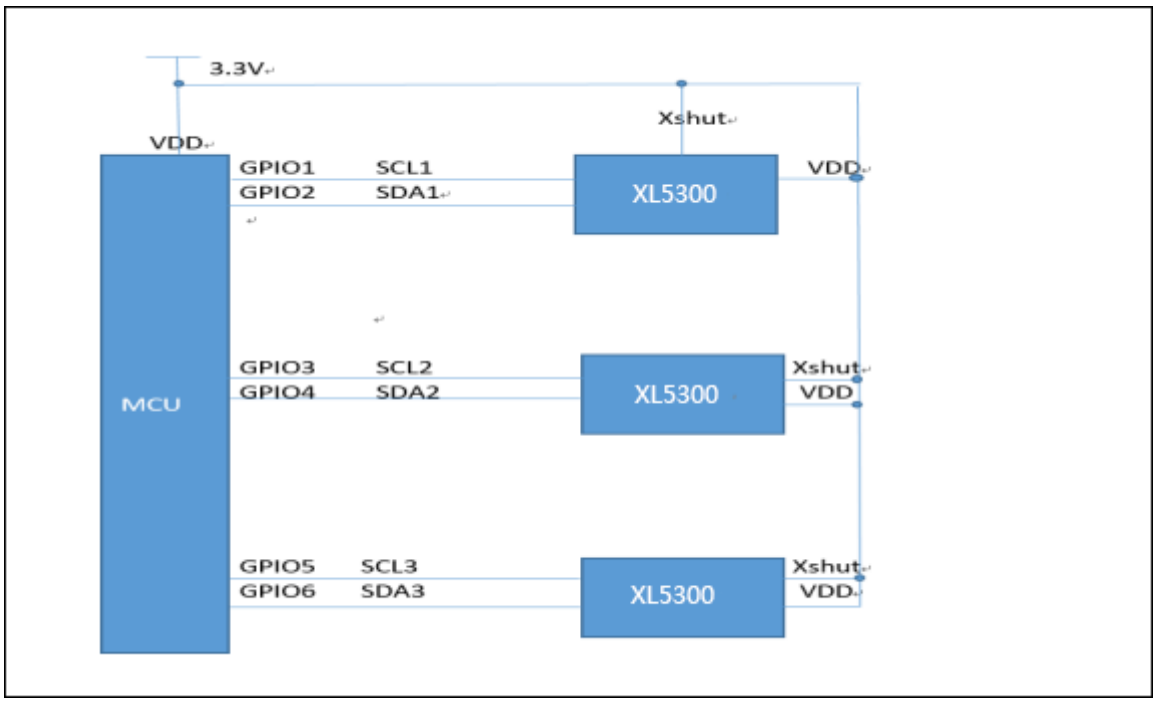


5.2 同一个主机下挂载多颗 XL5300 的方案

第一种方案：

将多颗分别挂在不同的 I2C 总线，或者使用 GPIO 模拟 I2C，挂载 XL5300。

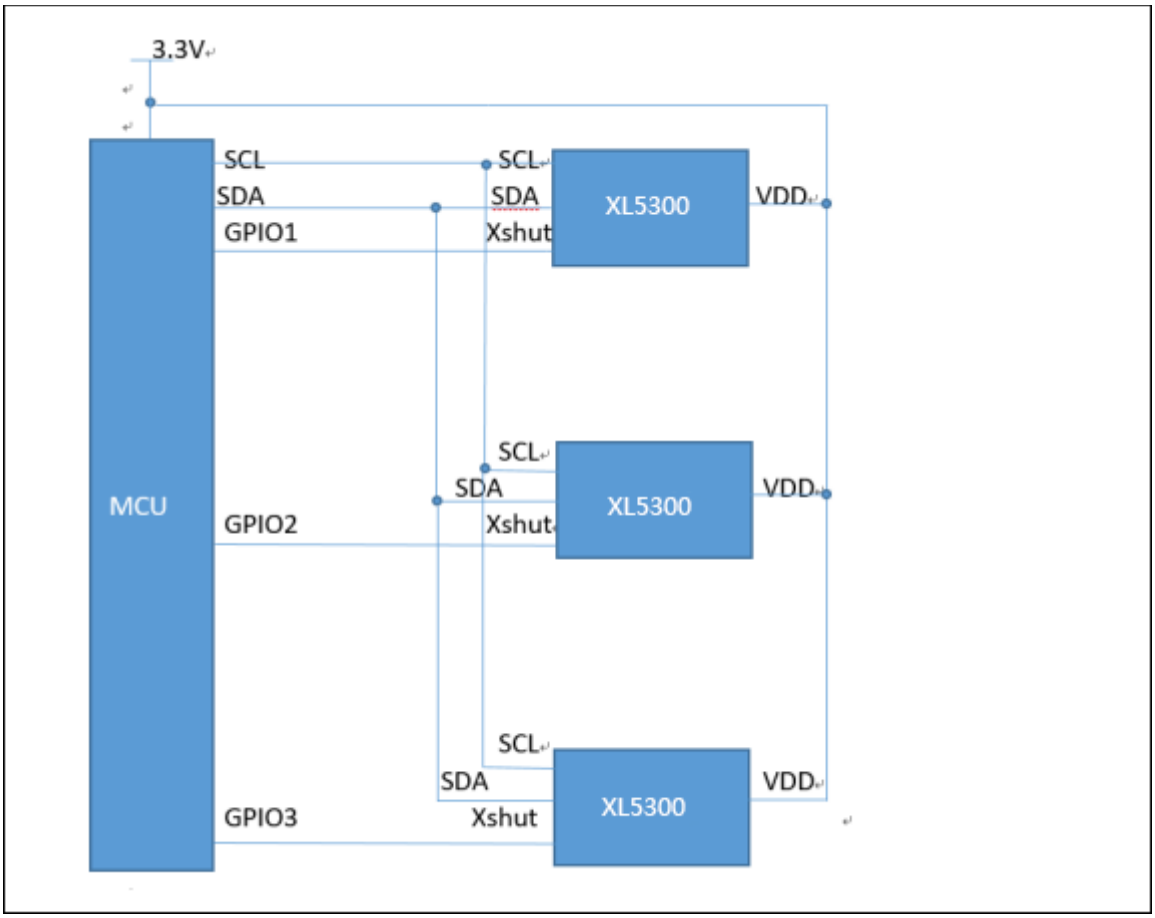
图 8：XL5300 多路 IIC 总线示意图



第二种方案：

XL5300 支持修改 I2C 地址，芯片上电 XSHUT 拉高后修改 I2C 地址，逐个修改，逐个初始化。I2C 地址可通过修改 0x06 寄存器值来更改。芯片掉电后 I2C 地址恢复出厂值，再次上电需要重新修改。

图 9：XL5300 单路 IIC 总线示意图



6 缩略语

表 2：缩略语

缩写	英文全称	含义
ESD	Electrostatic discharge	静电放电
I^2C	Inter-integrated circuit (serial bus)	集成电路总线
SPAD	Single photon avalanche diode	单光子雪崩二极管
SPI	Serial Peripheral Interface	串行外设接口
VCSEL	Vertical cavity surface emitting laser	垂直腔面发射激光器
ToF	Time of Flight	飞行时间
dToF	Direct Time of Flight	直接飞行时间
FoV	Field of view	视场角

修订记录

表 3：修订记录

版本	日期	更新说明
1.0	2020.09.02	初始发布版本。
1.1	2020.09.24	增加 XTalk 出厂标定流程； 更新测距流程； 增加 Pileup 校正说明。
1.2	2021.03.25	增加 offset 标定以及流程 更新函数名称
2.0	2021.05.24	增加帧率和积分次数配置说明 增加挂载多颗 XL5300 使用方法说明