



介绍

VL53L1X是一款长距离飞行时间传感器。

本用户手册的目的是描述使用VL53L1X驱动程序调用以获取测距数据的一组功能。 请参考VL53L1X数据表。

图1. VL53L1X测距传感器模块



内容

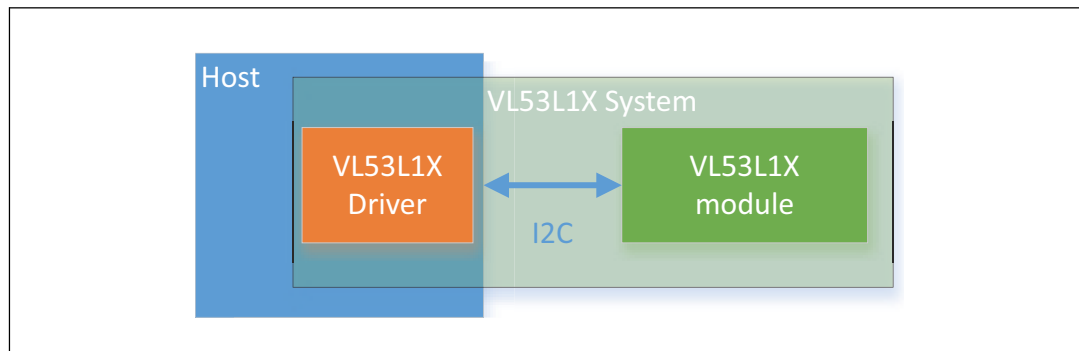
1	VL53L1X系统概述.....	4
2	测距API函数描述.....	5
2.1	自主测距说明.....	5
2.2	时机考虑.....	5
2.3	API函数调用流程.....	6
2.3.1	校准流程.....	6
2.3.2	流量.....	7
2.4	强制测距功能.....	8
2.4.1	数据初始化.....	8
2.4.2	静态初始化.....	8
2.4.3	开始测量.....	8
2.4.4	等待结果：轮询或中断.....	8
2.4.5	获取测量.....	8
2.4.6	清楚中断的来源.....	9
2.4.7	停止测量.....	9
2.5	可选的驱动功能.....	9
2.5.1	等待启动.....	9
2.5.2	时间预算和测量间期.....	9
2.5.3	距离模式.....	10
2.5.4	限制检查设置.....	10
2.5.5	阈值.....	12
2.5.6	感兴趣区域（ROI）设置.....	13
2.5.7	Spad数组坐标与场景.....	15
2.5.8	光学中心坐标.....	15
2.5.9	VDDIO配置.....	16
2.6	RangingMeasurementData结构.....	16
3	校准功能.....	18
3.1	RefSPAD校准.....	18
3.1.1	RefSPAD校准功能.....	18
3.1.2	RefSPAD校准程序.....	19
3.1.3	获得RefSPAD校准结果.....	19

3.1.4	设置RefSPAD校准数据	19
3.2	偏移校准	20
3.2.1	偏移校准功能.....	20
3.2.2	偏移校准程序.....	20
3.2.3	获得偏移校准结果	20
3.2.4	设置偏移校准数据	20
3.3	串扰校准	21
3.3.1	串扰校准功能.....	21
3.3.2	串扰校准程序.....	21
3.3.3	串扰校准距离表征	21
3.3.4	获得串扰校准结果	22
3.3.5	设置串扰校准数据	22
3.3.6	启用/禁用串扰补偿.....	23
4	驱动程序错误和警告	24
5	缩略语.....	26
6	修订记录	27

1 VL53L1X系统概述

VL53L1X系统由VL53L1X模块和在主机上运行的驱动程序组成。

图2. VL53L1X系统



ST提供了一个软件驱动程序，在本文档中称为“驱动程序”。

本文档介绍主机可以访问的驱动程序功能，以控制设备并获取测距数据。

该驱动程序是使用VL53L1X器件的一组功能的实现。它对OS集成和服务做了最少的假设。因此，动作的排序，执行/线程模型，平台适配和设备结构分配不是驱动程序实现的一部分，而是对软件集成商保持开放。

函数调用的顺序必须遵循本文档中定义的一组规则。

2测距API函数描述

本节给出了测距的功能描述，并描述了使用VL53L1X执行测距测量时应遵循的API呼叫流程。

2.1 自主测距说明

传感器通过可编程的测量间隔周期连续自动执行测距。

测距是在没有主机参与的情况下完成的，这允许主机处于低功耗状态。当测距可用时，主机仅在测量中断时唤醒。

可以设置距离和/或信号检测标准的阈值，然后在满足标准时引发中断。

2.2 时机考虑

定时预算定义为传感器执行和报告测距测量数据所需的编程时间。在此期间，VCSEL脉动。在计时预算结束时引发中断或更新日期就绪寄存器。

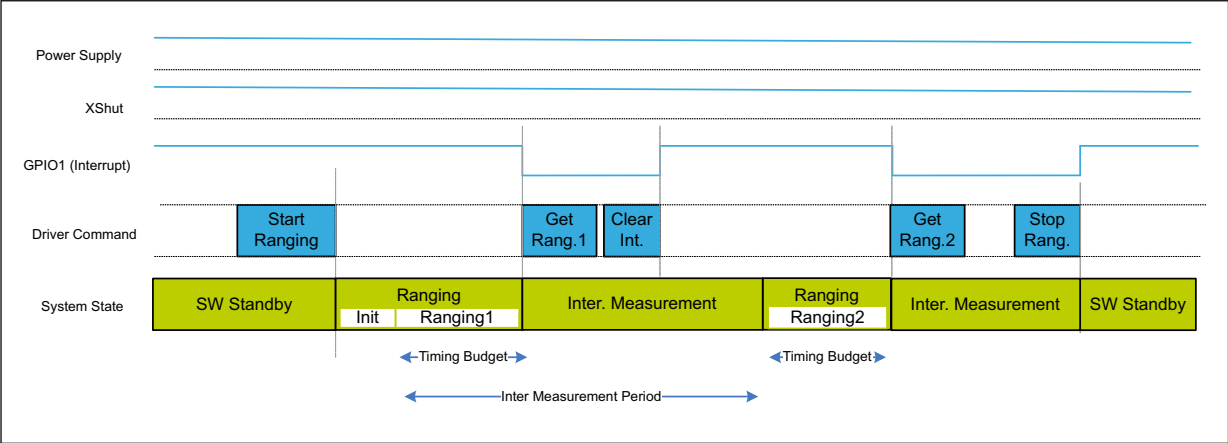
测量间期被定义为两次连续测量之间的编程时间。

[图3](#) 显示时间预算和测量间期。

主机可以通过使用专用驱动程序功能来更改默认的时间预算和测量间隔时间 [第2.5.2节：时间预算和相互测量期](#)。

主持人可以决定更改时间预算以提高测距精度或最大距离限制。

图3. VL53L1X自动量程序列和时序



2.3 API函数调用流程

VL53L1X驱动程序用于两种使用情况：

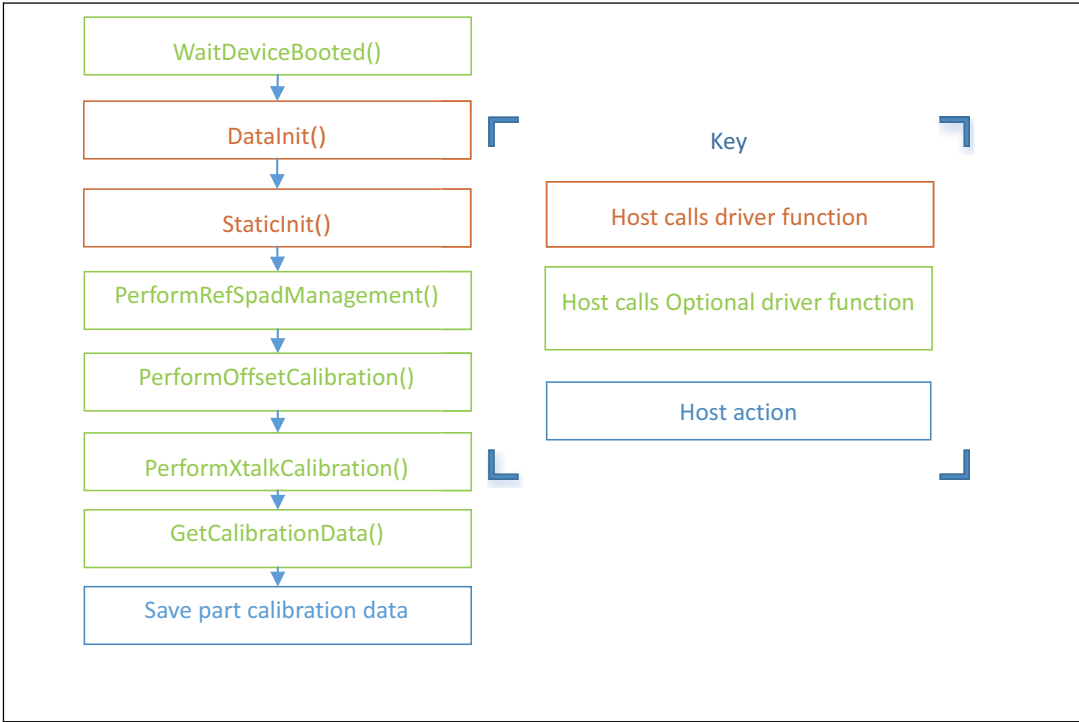
- 设备校准的校准流程
- 在用户应用程序级别使用测距流程

2.3.1 校准流程

校准流程在中描述 [图4](#).

所有用于校准的API函数都在中描述 [第3部分：校准功能](#).

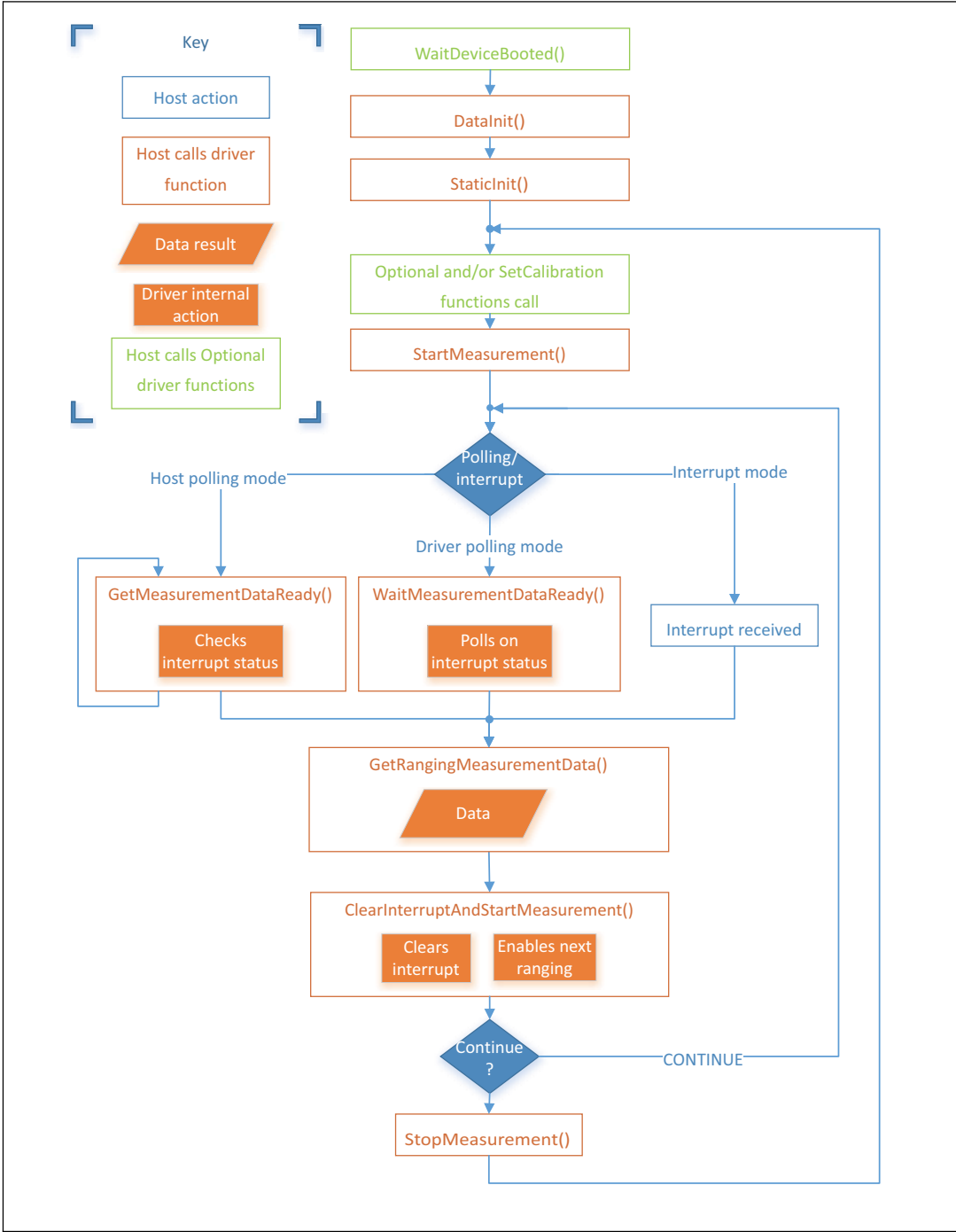
图4. VL53L1X校准流程



2.3.2 流量

测距流程在中描述 [图5](#).

图5. VL53L1X测距流程



2.4 强制测距功能

以下部分显示在开始测量之前执行系统初始化所需的API函数。

2.4.1 数据初始化

VL53L1_DataInit () 函数被调用一次。 它执行设备初始化。 它在器件退出复位后被调用一次且仅调用一次。

2.4.2 静态初始化

VL53L1_StaticInit () 函数允许加载特定用例的设备设置。

2.4.3 开始测量

必须调用VL53L1_StartMeasurement () 函数才能开始测量。

2.4.4 等待结果：轮询或中断

有三种方法可以知道测距数据可用：

1. 主机可以调用轮询功能，等待测距数据可用，

函数VL53L1_WaitMeasurementDataReady () 查询设备中断状态，直到测距数据就绪。

只要功能未完成，该功能就会阻止主机上的所有其他操作，因为执行了内部轮询。

2. 主机可以轮询功能以询问测距数据是否可用，

主机可以轮询功能VL53L1_GetMeasurementDataReady () 以了解新的测距数据是否准备就绪。

该功能不会阻止其他操作。 如果传感器在轮询模式下使用，则是首选和推荐的方法。

3. 主机可以等待物理中断

获得测距状态的另一种首选方法是使用物理中断输出：默认情况下，当新的测距数据准备就绪时，GPIO1被拉低。

该引脚只是输出引脚，该器件没有输入中断引脚。

2.4.5 获取测量

VL53L1_GetRangingMeasurementData () 可用于获取测距数据。 当调用该函数获取设备测距结果时，会调用一个结构

返回VL53L1_RangingMeasurementData_t。

这个结构描述在 [第2.6节: RangingMeasurementData结构](#)。

2.4.6 清楚中断的来源

中断必须通过调用驱动程序功能来清除：

读取测距数据后的VL53L1_ClearInterruptAndStartMeasurement ()

为了获得一致的结果，在进行测距测量后必须调用该函数。

如果此功能未被调用，则下一个测距将开始并且结果将被更新。但是，数据就绪状态标志不会被更新，物理中断引脚也不会被清除。

2.4.7 停止测量

主持人可以决定通过拨打电话来停止测量

VL53L1_StopMeasurement () 函数。

如果在量程测量期间发生停止请求，则立即中止测量。

2.5 可选的驱动功能

2.5.1 等待启动

VL53L1_WaitDeviceBooted () 函数确保设备已启动并准备就绪。

该功能是可选的。这是一个阻塞函数，因为有一个内部轮询。假设400 kHz I2C和每个事务的延迟时间为2毫秒，此功能不应阻塞超过4 ms。

2.5.2 时间预算和测量间期

定时预算是传感器执行一次量程测量所需的时间。

VL53L1_SetMeasurementTimingBudgetMicroSeconds () 是要使用的函数。

最小和最大时序预算为[20 ms, 1000 ms]示例：

Status = VL53L1_SetMeasurementTimingBudgetMicroSeconds (&VL53L1Dev, 66000) ; 将时间预算设置为66毫秒。

函数VL53L1_GetMeasurementTimingBudgetMicroSeconds () 获取编程的时序预算。

测量间期是两个测距操作之间的延迟。

测量间期可以编程。当测距完成时，设备等待编程的测量间隔期结束，然后再继续下一个测距。在测量间隔期间，传感器处于低功耗状态。

VL53L1_SetInterMeasurementPeriodMilliseconds () 是要使用的函数。

例：

Status = VL53L1_SetInterMeasurementPeriodMilliseconds (&VL53L1Dev, 1000) ; 将测量间隔周期设置为1秒。

函数VL53L1_GetInterMeasurementPeriodMilliseconds () 获取编程的测量间隔周期。

注意： 如果测量间隔时间短于定时预算，则一旦设备完成测距，则下一个测距立即开始。

注意： 传感器测距时不应调用计时预算和测量间期。 用户必须停止测距，更改这些参数并重新启动测距。

2.5.3 距离模式

距离模式是一个参数，用于优化内部设置和调谐，以根据应用所需的测距距离和环境光线条件获得最佳测距性能。

改变距离模式的好处详见 [表1: 距离模式](#)

表1. 距离模式

可能的距离模式	最大距离	收益/评论
短	高达1.3米	更好的环境免疫力
中	最多3米	
长（默认）	最多4米	最大距离

要使用的函数是VL53L1_SetDistanceMode () 。

用户可以调用VL53L1_GetDistanceMode () 来获得编程的距离模式。

2.5.4 限制检查设置

该驱动程序使用两个参数来限定测距测量：信号和西格马。

如果信号或西格马超出极限，测距将被标记为无效（请注意RangeStatus不等于零）。

适用的限制是：

- Sigma: VL53L1X_CHECKENABLE_SIGMA_FINAL_RANGE

西格玛用毫米表示，是对测量标准偏差的估计。

- 信号: VL53L1X_CHECKENABLE_SIGNAL_RATE_FINAL_RANGE

信号速率测量（以MCPS表示）表示从目标反射并由设备检测到的信号的幅度。



表2 给出了默认的极限状态和值。

表2. 默认限制状态和值

限制ID	默认限制状态	默认限制值	关联的RangeStatus
适马	启用	15毫米	1
信号	启用	1 Mcps	2

如果用户禁用极限检查，测距值将不再被过滤，传感器可能会返回不正确的测量结果。在这种情况下，RangeStatus 1和2将不会被报告。

更改限制默认设置应小心谨慎，因为副作用可能很重要。

标准偏差和最大测距的限制变化影响显示在 表3。

表3. 信号和西格玛限制变化影响

限制ID	行动	对标准偏差的影响	对最大测距距离的影响
	增加限制	-	+
	减少限制	+	-
	增加限制	+	-
	减少限制	-	+

VL53L1X_SetLimitCheckEnable () 和VL53L1X_GetLimitCheckEnable () 用于启用/禁用限制。

极限值使用VL53L1X_SetLimitCheckValue () 和VL53L1X_GetLimitCheckValue () 。

信号限制设置的一个示例是启用信号检查并将限制设置为0.4Mcps：

状态= VL53L1X_SetLimitCheckEnable (&VL53L1Dev, VL53L1X_CHECKENABLE_SIGNAL_RATE_FINAL_RANGE, 1) ;

*状态= VL53L1X_SetLimitCheckValue (&VL53L1Dev, VL53L1X_CHECKENABLE_SIGNAL_RATE_FINAL_RANGE, 0.40 * 65536) ;*

2.5.5 阈值

该设备可以配置为在距离和/或信号阈值检测模式下运行。当预先配置的标准匹配时，将测距数据报告给主机。

检测模式

检测模式允许选择过滤条件：

- 0：无滤波器（默认值，标准测距模式）
- 1：仅在距离标准上过滤

距离检测模式，基于阈值：

距离检测模式（通过CrossMode参数）定义距离标准：

- 0：低于一定距离：“阈值低”
 - 如果对象距离>距离较远或没有找到对象：无报告
 - 如果对象距离<距离低并且找到对象：报告
- 1：超出一定的距离：“阈值高”
 - 如果物距<距离高或没有找到物体：无报告
 - 如果对象距离>距离高并找到对象：报告
- 2：超出距离范围（最小/最大），“超出窗口”
 - 距离低<检测距离<距离高：无报告
 - 距离低>检测距离>距离高：报告
- 3：在距离范围内（最小/最大），“内窗”
 - 距离低>检测距离>距离高：无报告
 - 距离低<检测距离<距离高：报告距离低是以毫米为单位

的最小配置距离。

距离高是以毫米为单位的最大配置距离。

没有目标

这是一种替代检测模式。在标准用例中，如果未检测到目标，则不报告测距。不使用目标检测模式（将IntrNoTarget设置为1）允许在没有目标存在时产生中断。

API函数

VL53L1_SetThresholdConfig () 是要使用的函数。

VL53L1_DetectionConfig_t结构包含要设置的所有参数。

例：

```
Detectionconfig.DetectionMode = 1
```

```
Detectionconfig.Distance.CrossMode = 3
```

```
Detectionconfig.IntrNoTarget = 0
```

```
Detectionconfig.Distance.High = 1000
```

```
Detectionconfig.Distance.Low = 100
```

状态= `VL53L1_SetThresholdConfig (&VL53L1Dev, &detectionConfig)` ;

此功能用于编程设备，以便仅在10 cm和1 m内检测到物体时报告测距（如本例中）。

VL53L1_GetThresholdConfig () 函数允许编程的报告阈值配置被禁用。

2.5.6 感兴趣区域 (ROI) 设置

传感器的接收SPAD阵列包括覆盖整个视场 (FoV) 的16x16 SPAD。 可以使用较少数量的SPAD编程较小的感兴趣区域 (ROI)，以减少FoV。

要设置不同于默认16x16的投资回报率，用户可以拨打电话

`VL53L1_SetUserROI ()` 函数。

ROI是由两个角落定义的正方形或矩形：左上角和右下角。 四个坐标用于定位整个

SPAD阵列上的这两个角点：

- TopLeftX
- TopLeftY
- BotRightX
- BotRightY

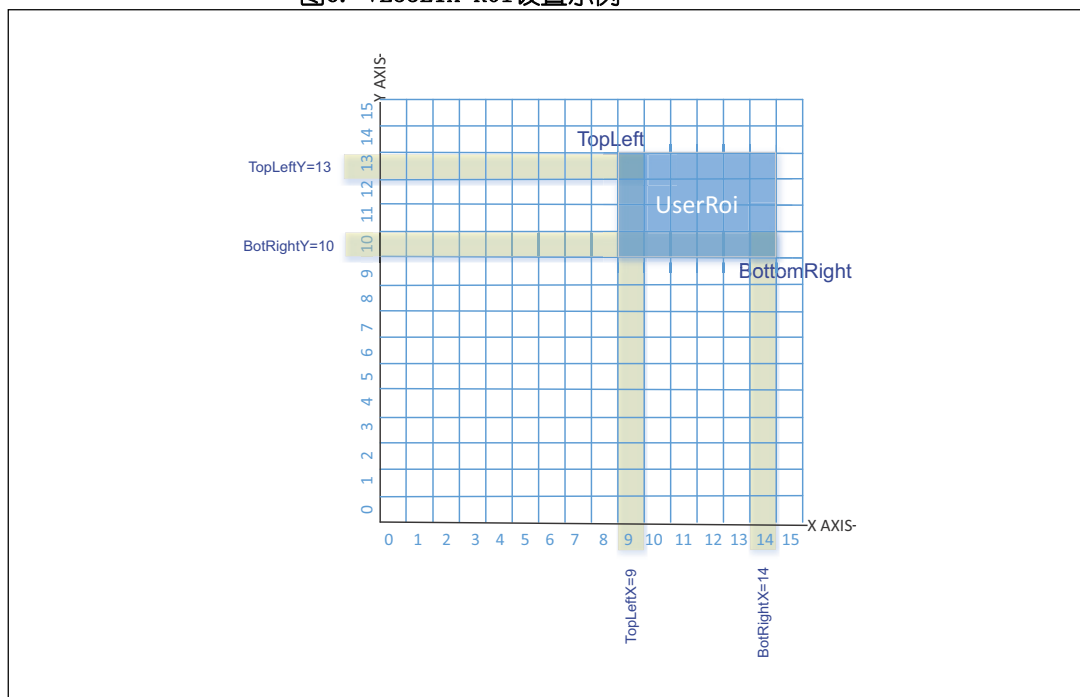
这些坐标是VL53L1_UserRoi_t结构的一部分。

用户必须定义结构中的ROI坐标值，并调用驱动程序函数来应用ROI更改。

最小ROI大小是4x4。

ROI设置的示例在中给出 [图6](#)。

图6. VL53L1X ROI设置示例



VL53L1_UserRoi_t结构包含ROI的坐标：

- TopLeftX: 8位整数，给出左上角的x坐标[0; 15]
- TopLeftY: 8位整数，给出左上角y坐标[0; 15]
- BotRightX: 8位整数，给出右下角的x坐标[0; 15]
- BotRightY: 8位整数，给出右下角的x坐标[0; 15]

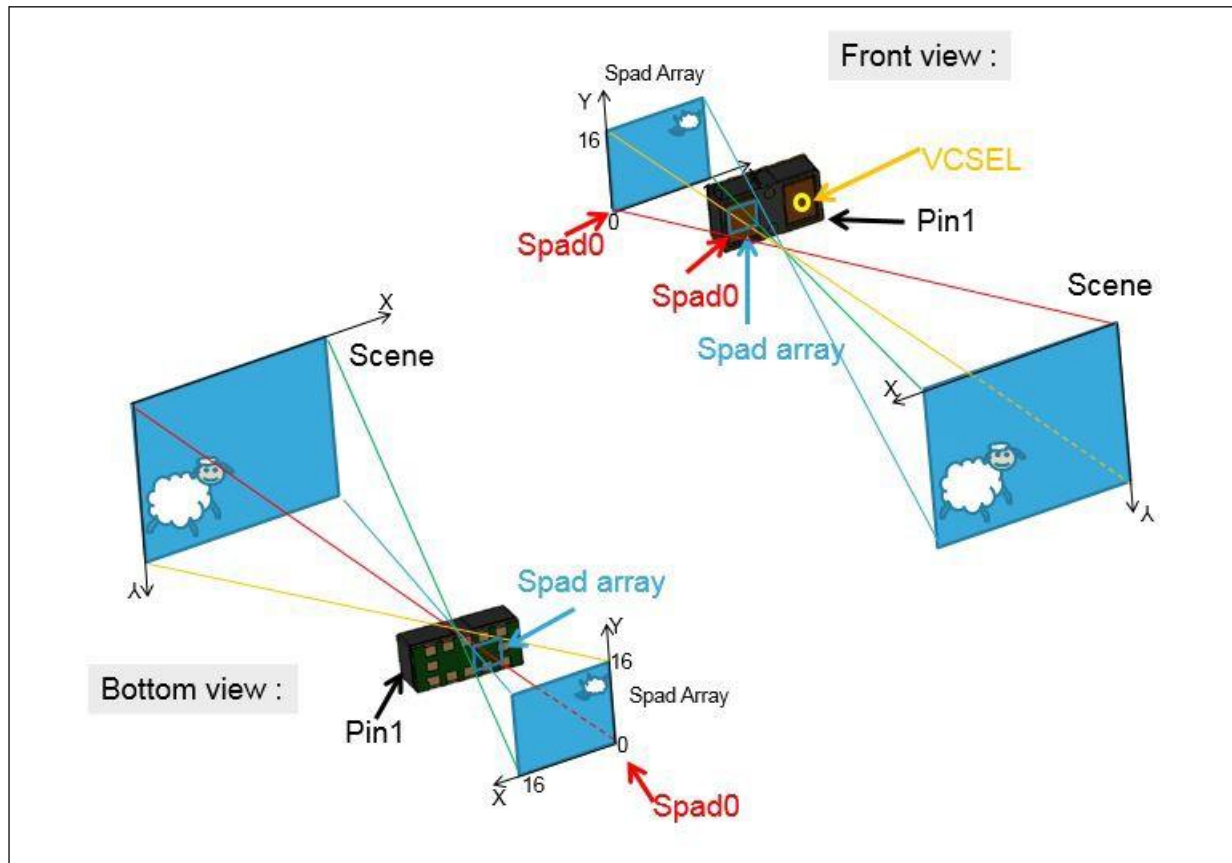
示例设置一个投资回报率（基于 图6）：

```
VL53L1_UserRoi_t roiConfig;
roiConfig.TopLeftX      = 9;
roiConfig.TopLeftY      = 13;
roiConfig.BotRightX     = 14;
roiConfig.BotRightY     = 10;
status = VL53L1_SetUserROI (&VL53L1Dev, &roiConfig) ;
```

2.5.7 Spad数组坐标与场景

图7 显示与FoV中的位置相比，SPAD阵列中的对象的坐标。

图7. VL53L1X坐标与场景



2.5.8 光学中心坐标

由于装配公差，设备的光学中心可能会有所不同。设备的光学中心针对每个部件进行测量。光学中心坐标存储在设备NVM中。

用户可通过调用VL53L1_GetCalibrationData () 访问光学中心坐标。返回的结构VL53L1_CalibrationData_t包含一个包含两个坐标（以SPAD编号表示）的子结构VL53L1_optical_centre_t:

- x_centre
- y_centre

主机可以使用这两个坐标来更好地将ROI与光学中心对齐。

2.5.9 VDDIO配置

如数据表中所述，用户可以为1V8或2V8模式的VDDIO值选择两种模式。

通过调用编译密钥直接在代码中选择模式
USE_I2C_2V8k。

如果定义了该编译密钥，系统将进入2V8模式，否则将保持默认的1V8模式。

2.6 RangingMeasurementData结构

VL53L1_RangingMeasurementData_t结构由以下部分组成：

- *TimeStamp*：未实施，请忽略它。
- *StreamCount*：这个8位整数计数器在每个范围内递增。该值从0开始，增加到255，然后从128增加到255。
- *RangingQualityLevel*：未执行，请忽略它。
- *SignalRateRtnMegaCps*：此值是MegaCountPer Second (MCPS) 中的返回信号速率。这是一个16.16的固定点值。要获得实际值，应该除以65536。
- *AmbientRateRtnMegaCps*：此值是返回环境速率（以MCPS计）。它是一个16.16固定点值，这是有效测量红外光的指标。要获得实际价值，应该除以65536。
- *EffectiveSpadRtnCount*：这个16位整数返回当前测距的有效SPAD计数。要获得实际值，应该除以256。
- *SigmaMilliMeter*：这个16.16定点值是电流测量的标准偏差的估算，以毫米表示。要获得实际价值，应该除以65536。
- *RangeMilliMeter*：这个16位整数给出了以毫米为单位的范围距离。
- *RangeFractionalPart*：未执行，请忽略它。
- *RangeStatus*：这个8位整数给出当前测量的范围状态。值为0表示测距有效（请参阅[表4](#)）。

表4. 范围状态

值	RangeStatus字符串	评论
0	VL53L1_RANGESTATUS_RANGE_VALID	测距测量是有效的
1	VL53L1_RANGESTATUS_SIGMA_FAIL	如果西格玛估计器检查高于内部定义的阈值，则会引发此问题
2	VL53L1_RANGESTATUS_SIGNAL_FAIL	如果信号值低于内部定义的阈值则产生
4	VL53L1_RANGESTATUS_OUTOFBOUNDS_FAIL	阶段超出界限时引发
5	VL53L1_RANGESTATUS_HARDWARE_FAIL	在硬件或VCSEL故障的情况下引发
7	VL53L1_RANGESTATUS_WRAP_TARGET_FAIL	包裹的目标，不匹配阶段
8	VL53L1_RANGESTATUS_PROCESSING_FAIL	内部算法下溢或溢出
14	VL53L1_RANGESTATUS_RANGE_INVALID	报告的范围无效



3 校准功能

为了充分发挥设备的性能，VL53L1X驱动器包含校准功能，应在客户生产线上运行一次。

必须运行校准程序以补偿可能影响器件性能的部件间部件参数和玻璃盖的存在。

存储在主机中的校准数据必须在每次启动时使用专用的驱动程序功能加载到VL53L1X中。

需要三次校准：RefSPAD，失调和串扰。

校准函数的调用顺序非常重要：首先调用RefSPAD，然后偏移第二，串扰第三。

三种校准功能可以依次进行，也可以单独进行。当单独运行时，必须在运行当前校准之前加载先前的步骤数据。

3.1 RefSPAD校准

ST在最后的模块测试期间校准SPAD的数量。该部分对零件的值存储在NVM中，并在引导期间自动加载到设备中。

此校准允许调整SPAD的数量以优化器件动态。

但是，在模块顶部添加护盖玻璃可能会影响此校准。我们建议客户在最终产品应用中再次执行此校准。

在调用此函数时应用在FMT中运行的相同算法。该算法搜索三种可能的SPAD类型：

1. 非衰减SPAD
2. SPAD衰减5倍
3. SPAD衰减10倍

选择SPAD的数量和类型以避免内部信号饱和。

3.1.1 RefSPAD校准功能

该操作有专用功能：

VL53L1_PerformRefSpadManagement (&VL53L1Dev)

注意： 必须在校准过程中首先调用该功能。

3.1.2 RefSPAD校准程序

用户必须确保在校准过程中距离传感器5厘米处没有距离较近的目标。
在低红外光条件下（室内）进行校准会更好。执行此校准的时间仅为几毫秒。

必须在调用VL53L1_DataInit () 和VL53L1_StaticInit () 函数后调用VL53L1_PerformRefSpadManagement函数。参考 [图4: VL53L1X校准流程](#)。

当调用校准功能时，执行RefSPAD校准，并在最后应用新的RefSPAD参数。

3.1.3 获得RefSPAD校准结果

函数VL53L1_GetCalibrationData () 允许获取所有校准数据。返回的结构VL53L1_CalibrationData_t还包含一个名为VL53L1_customer_nv_managed_t的子结构，其中包含八个RefSPAD校准参数：

- `ref_spad_man num_requested_ref_spads`: 此值介于5和44之间。它给出了所选SPAD的数量。
- `ref_spad_man ref_location`: 这个值可以
 - 非减毒SPAD
 - SPAD衰减5倍
 - SPAD衰减10倍
- 六个附加参数给出了所选位置的良好SPAD图。

出厂校准后，这些校准数据必须存储在主机存储器中，并在每个设备启动时加载，以避免重新校准。用户可以存储整个结构VL53L1_CalibrationData_t或存储八个参数（以节省存储空间）。

3.1.4 设置RefSPAD校准数据

在每次启动时，硬重置后，用户可以从主机内存中加载RefSPAD校准数据。在调用VL53L1_DataInit () 和VL53L1_StaticInit () 函数之后，必须调用VL53L1_SetCalibrationData ()。参考 [图5: VL53L1X测距流程](#)。

如果用户在校准过程中优化了校准数据存储，建议通过调用VL53L1_GetCalibrationData () 来获得整个校准结构，修改 [第3.1.3节: 获取RefSPAD校准结果](#)，并调用VL53L1_SetCalibrationData ()。

3.2 偏移校准

在客户板上焊接器件或添加玻璃罩可能会在测距范围内产生偏移。 在零点校准过程中，必须测量和补偿这种零件间偏移。

3.2.1 偏移校准功能

该操作有专用功能：

```
VL53L1_PerformOffsetSimpleCalibration (&VL53L1Dev, CalDistanceMilliMeter)
```

该函数的参数是以毫米为单位的偏移校准距离。

注意：在串扰校准和RefSPAD优化完成后（校准完成或RefSPAD参数加载），必须执行偏移校准。

3.2.2 偏移校准程序

客户必须使用校准图表，放置在给定的距离（CalDistanceMilliMeter）来执行偏移校准。推荐设置的详细信息在中给出 [表5](#)。

表5. 设置偏移校准

图表	图表距离 (CalDistanceMilliMeter)	环境条件
灰色目标 (在940nm下17%的反射率)	建议值：140毫米	黑暗 (没有投入红利)

当调用校准功能时，执行偏移校准，并在最后应用偏移校正。

3.2.3 获得偏移校准结果

函数VL53L1_GetCalibrationData（）允许获取所有校准数据。 返回的结构VL53L1_CalibrationData_t还包含一个名为VL53L1_customer_nvmm_managed_t的子结构，其中包含主要的偏移校准结果：
算法part_to_part_range_offset_mm。

3.2.4 设置偏移校准数据

客户可以在调用VL53L1_DataInit（）和VL53L1_StaticInit（）函数后，使用VL53L1_SetCalibrationData（）加载偏移校准数据。
但是，最好调用VL53L1_GetCalibrationData（），修改参数中描述的参数 [第3.2.3节：获得偏移校准结果](#)
（algo part_to_part_range_offset_mm）并调用VL53L1_SetCalibrationData（）。

3.3 串扰校准

串扰 (xtalk) 定义为由于审美和保护性原因，在模块顶部添加的保护窗（保护玻璃）内由于VCSEL光反射而在感应阵列上接收的返回信号量。

根据玻璃罩的质量，返回信号的数量可能很大，并可能影响传感器的性能。 VL53L1X具有内置校正功能，可以补偿这种串扰现象。

串扰校准用于估计补偿模块顶部添加的玻璃罩所产生的影响所需的校正量。

3.3.1 串扰校准功能

一个专用功能可用于此操作: *VL53L1_PerformSingleTargetXTalkCalibration (&VL53L1Dev, XtalkCalDistance)* ;

该功能的一个参数是以毫米为单位的串扰校准距离。

注意: 在完成偏移补偿（校准完成或偏移参数加载）之后，此功能必须在校准流程中第三位调用。

3.3.2 串扰校准程序

串扰校准应该在黑暗的环境中进行，没有红外线贡献。

当调用校准功能时，将执行串扰校准，并在最后应用串扰校正。

表6. 设置串扰校准

图表	图表距离 (<i>XtalkCalDistance</i>)	环境条件
灰色目标 (在940nm下17%的反射率)	如定义 第3.3.3节: 串扰 校准距离表征	黑暗 (没有投入红利)

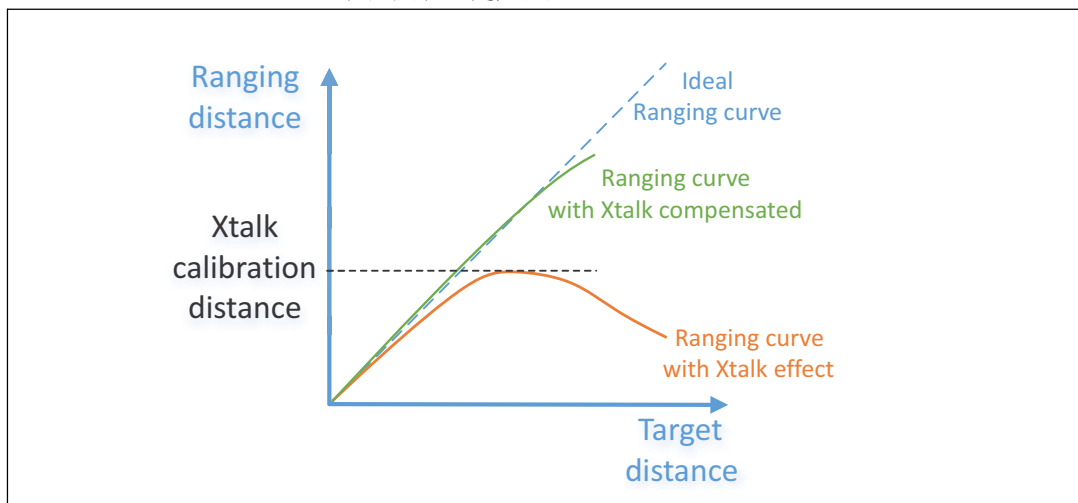
3.3.3 串扰校准距离表征

串扰校准距离需要由用户来表征，因为它取决于系统环境，主要取决于覆盖玻璃材料和光学属性，以及气隙值（传感器和玻璃盖之间的距离）。

[图8: VL53L1X串扰校准距离定义](#) 显示测距曲线上的串扰效应。 从给定的距离，串扰的影响是主要的，传感器开始低于范围。



图8. VL53L1X串扰校准距离定义



串扰校准距离对应于存在盖玻片时传感器报告的最大测距距离（参见图2）[图8](#)。

该最大测距距离是串扰校准驱动器功能的一个参数。

串扰校正的测距曲线是应用串扰补偿时（串扰校准完成时或串扰校准数据加载后）的测距结果。

3.3.4 获得串扰校准结果

函数VL53L1_GetCalibrationData（）允许获取所有校准数据。返回的结构VL53L1_CalibrationData_t还包含一个名为VL53L1_customer_nvmm_managed_t的子结构，其中包含串扰校准结果：
算法crosstalk_compensation_plane_offset_kcps。

3.3.5 设置串扰校准数据

客户可以在调用VL53L1_DataInit（）和VL53L1_StaticInit（）函数后，使用VL53L1_SetCalibrationData（）函数加载串扰校准数据，

建议调用VL53L1_GetCalibrationData（），修改VL53L1_customer_nvmm_managed_t子结构中的算法crosstalk_compensation_plane_offset_kcps参数，然后调用VL53L1_SetCalibrationData（）应用串扰补偿。

3.3.6 启用/禁用串扰补偿

调用以启用或禁用串扰补偿的功能是：

VL53L1_SetXTalkCompensationEnable () 。

VL53L1_SetXTalkCompensationEnable (&VL53L1Dev, 0) ; //禁用串扰补偿。

VL53L1_SetXTalkCompensationEnable (&VL53L1Dev, 1) ; //启用串扰补偿。

注意： 此功能不执行任何串扰校准或数据加载，它只是启用串扰补偿。 必须在启动测距之前调用可选的函数调用。

4 驱动程序错误和警告

当调用任何驱动程序函数时会报告驱动程序错误。 驱动程序错误的可能值在 [表7](#)。

请注意，存在警告以通知用户某些参数未优化。 警告不是阻止主机的点。

表7. 裸机驱动程序错误和警告说明

错误值	API错误字符串	发生
0	VL53L1_ERROR_NONE	没有错误
-1	VL53L1_ERROR_CALIBRATION_WARNING	校准数据无效
-4	VL53L1_ERROR_INVALID_PARAMS	无效的参数在函数中设置
-5	VL53L1_ERROR_NOT_SUPPORTED	编程的配置不支持所请求的参数
-6	VL53L1_ERROR_RANGE_ERROR	中断状态不正确
-7	VL53L1_ERROR_TIME_OUT	测距由于超时而中止
-8	VL53L1_ERROR_MODE_NOT_SUPPORTED	请求模式不受支持
-10	VL53L1_ERROR_CALIBRATION_WARNING	提供的缓冲区大于I2C支持
-14	VL53L1_ERROR_INVALID_COMMAND	命令在当前模式下无效
-16	VL53L1_ERROR_REF_SPAD_INIT	参考SPAD校准期间发生错误
-22	VL53L1_ERROR_XTALK_EXTRACTION_FAIL	当串扰校准功能没有成功的采样来计算串扰时投射。在这种情况下，没有足够的信息来生成新的串扰参数信息。 该功能将退出并保持当前的串扰参数不变。
-23	VL53L1_ERROR_XTALK_EXTRACTION_SIGMA_LIMIT_FAIL	当串扰校准函数发现 Σ 估计超出允许的最大限制时抛出。在这种情况下，串扰样本的噪声太大，无法进行测量。 该功能将退出并保持当前的串扰参数不变。
-24	VL53L1_ERROR_OFFSET_CAL_NO_SAMPLE_FAIL	偏移校准功能找不到有效测距时抛出。
-28	VL53L1_WARNING_REF_SPAD_CHAR_NOT_ENOUGH_SPADS	如果有少于5个可用的SPAD，则投掷。 确保校准设置符合ST的建议。
-29	VL53L1_WARNING_REF_SPAD_CHAR_RATE_TOO_HIGH	如果最终参考速率大于参考速率上限，则抛出 - 默认值为40 Mcps。 确保校准设置符合ST的建议。
-30	VL53L1_WARNING_REF_SPAD_CHAR_RATE_TOO_LOW	如果最终参考速率低于参考速率下限，则抛出 - 默认值为10 Mcps。 确保校准设置符合ST的建议。

表7. 裸机驱动程序错误和警告说明（续）

错误值	API错误字符串	发生
-31	VL53L1_WARNING_OFFSET_CAL_MISSING_SAMPLES	如果有少于所要求的有效样本数量，则抛出。 确保偏移校准设置符合ST建议。
-32	VL53L1_WARNING_OFFSET_CAL_SIGMA_TOO_HIGH	如果偏移校准范围西格玛估计过高，则抛出。 确保偏移校准设置符合ST建议。
-33	VL53L1_WARNING_OFFSET_CAL_RATE_TOO_HIGH	信号速率大于极限且传感器饱和时投射。 确保偏移校准设置符合ST建议。
-34	VL53L1_WARNING_OFFSET_CAL_SPAD_COUNT_TOO_LOW	投掷不足时可以使用SPADS。 确保偏移校准设置符合ST建议。
-41	VL53L1_ERROR_NOT_IMPLEMENTED	所调用的函数未实现

5 缩略语

表8. 缩略语和缩写

首字母缩略词/缩写	定义
I2C	内部集成电路（串行总线）
NVM	非易失性存储器
SPAD	单光子雪崩二极管
VCSEL	垂直腔面发射激光器
FMT	最终模块测试
API	应用程序接口



6 修订记录

表9. 文档修订历史记录

日期	调整	变化
07-Mar-2018	1	初始发行

