Configuring the VL53L1X Time-of-Flight sensor to range up to 100 Hz

Introduction

In some use cases, such as robot toys and vacuum cleaning robots, the VL53L1X system needs to range rapidly at very short distances (< 1 m). For example, vacuum cleaning robots need to range as fast as possible to detect stairs or steps to avoid falling down the stairs.

In Fast ranging mode and Short distance mode, the sensor ranges at 100 Hz up to 1.3 m.

This document shows how to configure the VL53L1X Time-of-Flight (ToF) sensor to range at its fastest ranging speed of 100 Hz, using the VL53L1X_API functions.

The user should be familiar with using the VL53L1X API and should be aware of the X-CUBE-53L1A1 development environment. For more details on VL53L1X API usage, please refer to the VL53L1X API user manual (UM2356) on st.com. For the original example source code, please refer to the software package X-CUBE-53L1A1 (long distance ranging Time-of-Flight sensor software expansion of STM32Cube) on st.com. The example code used in the current document has been extracted from the fast ranging example in the X-CUBE-53L1A1 software.

1 Acronyms and abbreviations

The main acronyms and abbreviations used in this document are listed below.

Table 1. Acronyms and abbreviations

API	application programming interface
Interrupt	In this document, interrupt designates both a software interrupt and a physical interrupt. The software interrupt corresponds to a register value toggle while the physical hardware interrupt presents at the GPIO1 pin of the sensor.
SNR	signal to noise ratio
ToF	Time-of-Flight

2 Fast ranging code extract

Below is an extract of the fast ranging code from the X-CUBE-53L1A1.

```
void FastRangingTest(void)
{
     uint8_t firstTimeInterrupt = 1;
     static VL53L1_RangingMeasurementData_t RangingData;
     status = VL53L1_WaitDeviceBooted(Dev);
     status = VL53L1_DataInit(Dev);
     status = VL53L1_StaticInit(Dev);
     /* (1) Three important functions to use to configure the sensor in fast mode */
     status = VL53L1_SetPresetMode(Dev, VL53L1_PRESETMODE_LITE_RANGING);
     status = VL53L1_SetDistanceMode(Dev, VL53L1_DISTANCEMODE_SHORT);
     status = VL53L1_SetMeasurementTimingBudgetMicroSeconds(Dev, 10000);
     status = VL53L1_StartMeasurement(Dev);
     do {
          status = VL53L1_WaitMeasurementDataReady(Dev)
          if(firstTimeInterrupt == 0){
               status = VL53L1_GetRangingMeasurementData(Dev,&RangingData);
               status = VL53L1_ClearInterruptAndStartMeasurement(Dev);
          }
          /* (2) If first interrupt, do not get data, clear interrupt and start */
          else{
              status = VL53L1_ClearInterruptAndStartMeasurement(Dev);
              firstTimeInterrupt = 0;
          }
     }
     while (1);
}
```

3 Fast ranging code description

In the fast ranging code extract, the function "FastRangingTest () does the following:

- Waits for the sensor to boot
- Initializes the sensor with the following default parameters
- Configures the sensor in Fast ranging mode
- Starts the measurement
- Gets the ranging data loop
- Ranges infinitely in the while loop: when the data are ready, an interrupt is raised, the application reads the data, clears the interrupt, and enables the next start ranging.

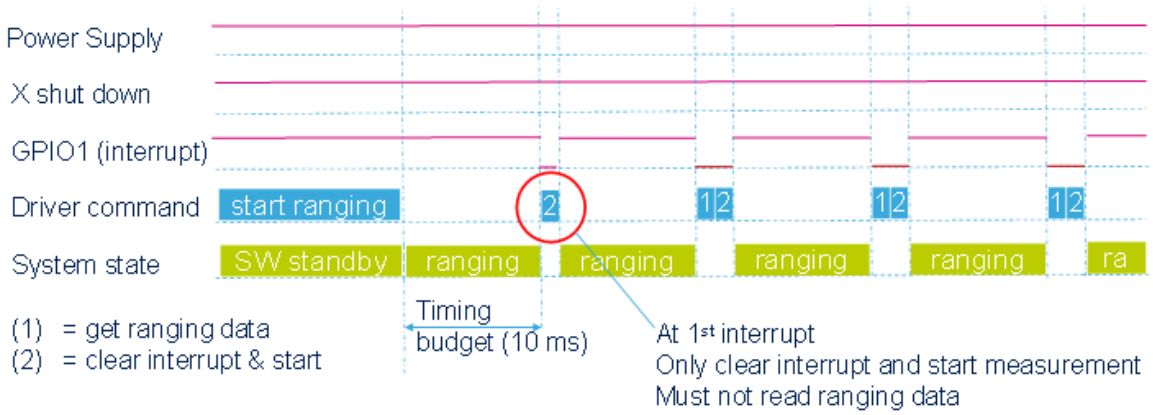
Note: (1) The function call order is important. VL53L1_SetPresetMode must to be called first followed by VL53L1_SetDistanceMode, and VL53L1_SetMeasurementTimingBudgetMicroSeconds.

*Note: (2) At the first interrupt and after a stop/restart ranging, the **data must not be read**. A clear interrupt and start measurement command must be issued to enable the next ranging and interrupt event.*

4 Function descriptions

- **VL53L1_WaitDeviceBooted(Dev)** ensures that the device is booted and ready
- **VL53L1_DataInit(Dev)** performs device initialization
- **VL53L1_StaticInit(Dev)** loads the default setting
- **VL53L1_SetPresetMode (Dev, VL53L1_PRESETMODE_LITE_RANGING)** programs the sensor when it is running in Lite ranging mode. The driver supports Autonomous and Lite ranging mode. However, Lite ranging mode only works when the CPU is available to read the result and clear the interrupt. With the preset mode VL53L1_PRESETMODE_LITE_RANGING, the start command enables the next ranging. A new ranging starts immediately after the start command is issued. By default, the inter-measurement period is set to equal the timing budget (see figure below).
- **VL53L1_SetDistanceMode (Dev, VL53L1_DISTANCEMODE_SHORT)** programs the sensor when it is running in Short ranging mode. The sensor can work with Long distance mode while it is in Fast ranging mode. However, the 10 ms of timing budget available is not long enough to get an accurate and consistent measurement. When the signal to noise ratio (SNR) becomes marginal because the target is too far away, is too small, or the ambient light is too high, the sensor needs more data. To get consistent ranging data, it is mandatory to set the distance mode to “Short” when the sensor is running in Fast ranging mode.
- **VL53L1_SetMeasurementTimingBudgetMicroSeconds (Dev, 10000)** programs the timing budget to 10 ms. This is the time resource allocated to the sensor to make one ranging. The longer the timing budget, the more time the sensor has to collect the photons required to compute the ranging distance accurately.
- **VL53L1_StartMeasurement(Dev)** starts a measurement
- **VL53L1_WaitMeasurementDataReady(Dev)** polls on the device interrupt status until the ranging data are ready.
- **VL53L1_GetRangingMeasurementData(Dev,&RangingData**) gets the ranging data
- **VL53L1_ClearInterruptAndStartMeasurement(Dev**) clears the interrupt and enables the next measurement.

Figure 1. Ranging sequence and timings in Fast ranging mode



Revision history

Table 2. **Document revision history**

Date	Version	Changes
21-Feb-2019	1	Initial release

Contents