# A121 Tank Level Reference Application User Guide

User Guide

A121 Tank Level Reference Application User Guide

User Guide

Author: Acconeer AB

Version:a121-v1.5.0

Acconeer AB January 19, 2024

## Contents

## 1 Acconeer SDK Documentation Overview

To better understand what SDK document to use, a summary of the documents are shown in the table below.

Table 1: SDK document overview.

| Name | Description | When to use |
|---|---|---|
| | ***RSS API documentation (html)*** | |
| rss_api | The complete C API documentation. | - RSS application implementation<br>- Understanding RSS API functions |
| | ***User guides (PDF)*** | |
| A121 Assembly Test | Describes the Acconeer assembly test functionality. | - Bring-up of HW/SW<br>- Production test implementation |
| A121 Breathing Reference Application | Describes the functionality of the Breathing Reference Application. | - Working with the Breathing Reference Application |
| A121 Distance Detector | Describes usage and algorithms of the Distance Detector. | - Working with the Distance Detector |
| A121 SW Integration | Describes how to implement each integration function needed to use the Acconeer sensor. | - SW implementation of custom HW integration |
| A121 Presence Detector | Describes usage and algorithms of the Presence Detector. | - Working with the Presence Detector |
| A121 Smart Presence Reference Application | Describes the functionality of the Smart Presence Reference Application. | - Working with the Smart Presence Reference Application |
| A121 Sparse IQ Service | Describes usage of the Sparse IQ Service. | - Working with the Sparse IQ Service |
| A121 Tank Level Reference Application | Describes the functionality of the Tank Level Reference Application. | - Working with the Tank Level Reference Application |
| A121 Touchless Button Reference Application | Describes the functionality of the Touchless Button Reference Application. | - Working with the Touchless Button Reference Application |
| A121 STM32CubeIDE | Describes the flow of taking an Acconeer SDK and integrate into STM32CubeIDE. | - Using STM32CubeIDE |
| A121 Raspberry Pi Software | Describes how to develop for Raspberry Pi. | - Working with Raspberry Pi |
| A121 Ripple | Describes how to develop for Ripple. | - Working with Ripple on Raspberry Pi |
| XM125 Software | Describes how to develop for XM125. | - Working with XM125 |
| XM126 Software | Describes how to develop for XM126. | - Working with XM126 |
| I2C Distance Detector | Describes the functionality of the I2C Distance Detector Application. | - Working with the I2C Distance Detector Application |
| I2C Presence Detector | Describes the functionality of the I2C Presence Detector Application. | - Working with the I2C Presence Detector Application |
| I2C Breathing Reference Application | Describes the functionality of the I2C Breathing Reference Application. | - Working with the I2C Breathing Reference Application |
| | ***Handbook (PDF)*** | |
| Handbook | Describes different aspects of the Acconeer offer, for example radar principles and how to configure | - To understand the Acconeer sensor<br>- Use case evaluation |
| | ***Readme (txt)*** | |
| README | Various target specific information and links | - After SDK download |

## 2 Tank level

The tank level reference application shows the liquid level in a tank with an A121 sensor mounted at the top. This reference application is built on top of the distance detector (see Distance detector) with some additional configurations specific to the tank level application.

**Measurement range** The liquid level in the tank can be measured from a minimum distance of 3 cm from the sensor to a maximum distance of 20 m.

**Presets** The application includes three pre-defined configurations optimized for tanks of varying sizes: small, medium, and large, corresponding to depths of 50 cm, 6.0 m, and 20.0 m, respectively.

**Configuration** The configuration parameter *start_m* defines the distance from the sensor to the surface of the liquid when the tank is full. Similarly, *end_m* defines the distance from the sensor to the tank base, i.e., the liquid level when the tank is empty.

Multiple peaks can be detected in the distance domain by the detector due to various factors such as sensor installation, tank geometry, etc. The peak sorting method in the detector parameters can be used to ensure that the correct peak is chosen as the first peak for calculating the liquid level. Refer to Distance detector for a detailed description of the detector parameters.

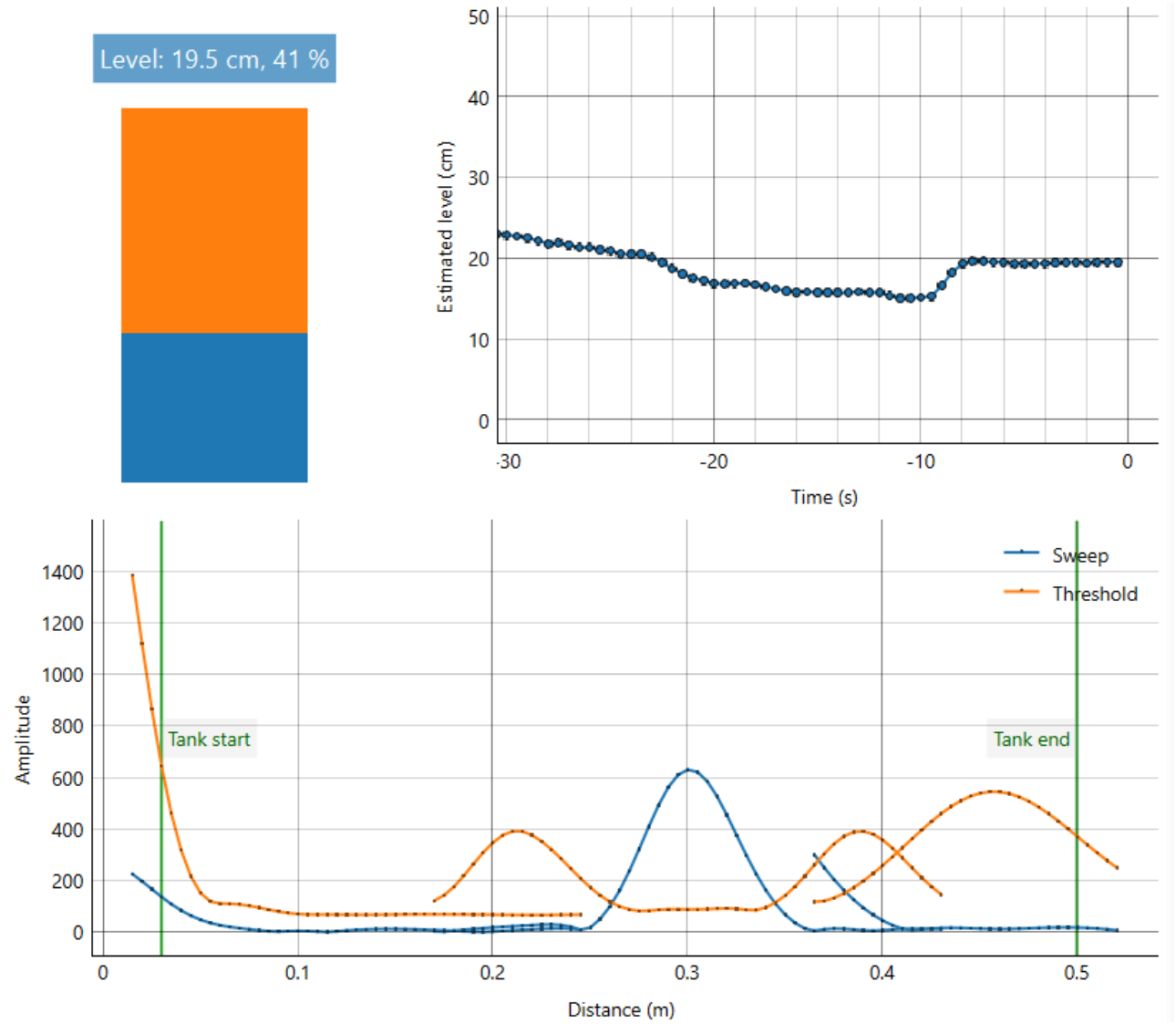
**Calibration** The distance detector calibration process performs noise level estimation and offset compensation. The close range measurement calibration is also performed in case the close range measurement is active, which depends on the starting distance. In addition, the recorded threshold is also computed if the detector is configured to use the recorded threshold or if the close range measurement is active.

Before starting level measurements, the detector needs to be calibrated. For close range measurements, no object must be present in the close range when the calibration is started.

**Processing** The liquid level is given as the distance of the surface of the liquid from the tank base, and is calculated using the distance to the first peak in the distance detector results. Due to movement in the surface of the liquid, the level measurements may fluctuate. A median filter is employed to counter the fluctuation in the level results by calculating the median of *median_filter_length* results. Averaging *num_medians_to_average* median filter results can further improve the confidence in the level result.

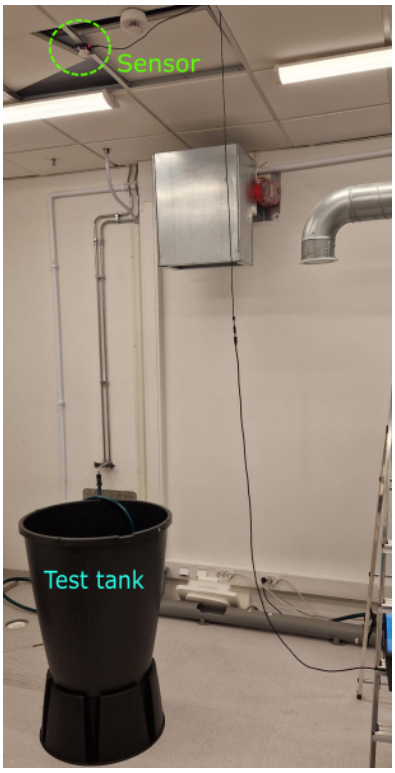
### 2.1 GUI

The GUI includes three plots. The top left plot indicates the fluid level, and the top right plot shows the level history. The bottom plot shows the tank size, the subsweeps, and the threshold used by the distance detector to detect amplitude peaks in the subsweeps. The subsweeps and different threshold types are decribed in Distance detector.




## 2.2 Testing

**Test setup** The level estimation performance of the reference application is tested using the three different setups shown below, which correspond to small (left), medium (middle), and large (right) tanks.


**Test equipment**

- A121 EVK + XR121
- FZP lens ( for medium and large tanks)
- Small tank (height = 30 cm)
- Test tank (height = 1.0 m)
- Exploration tool with Tank level refrence application

A simple workaround is used to estimate the performance for the medium and the large tank, where the sensor is mounted at a height to have the water level in at a longer distance than the actual test tank size.

**Test case** Fill tank x cm and verfiy that the actual distance is equal to the measured distance.

**Configurations**

Table 2: Application parameter configurations

| Parameter | Small | Medium | Large |
|---|---|---|---|
| Median filter length | 5 | 3 | 3 |
| Num measurements averaged | 5 | 3 | 1 |
| Tank start | 0.03 m | 0.05 m | 0.5 m |
| Tank end | 0.3 m | 2.7 m | 7.8 m |
| Max step length | 1 | 2 | 8 |
| Max profile | 1 | 3 | 5 |
| Threshold method | CFAR | CFAR | CFAR |
| Reflector shape | Planar | Planar | Planar |
| Peak sorting method | Closest | Strongest | Strongest |
| Threshold sensitivity | 0.0 | 0.0 | 0.0 |
| Signal quality | 20 | 20 | 20 |

**Results** Few results obtained using the above configurations are listed below.

Table 3: Test results

| Tank | Actual level (m) | Measured level (m) |
|---|---|---|
| Small | 0.106 | 0.104 |
| Medium | 0.398 | 0.401 |
| Large | 0.100 | 0.085 |

# 3 Memory

## 3.1 Flash

The reference application compiled from ref_app_smart_presence.c on the XM125 module requires around 95 kB.

## 3.2 RAM

The RAM can be divided into three categories, static RAM, heap, and stack. Below is a table for approximate RAM for an application compiled from ref_app_smart_presence.c.

| **RAM** | **Size (kB)** | | |
|---|---|---|---|
| *Preset* | *Small* | *Medium* | *Large* |
| Static | 1 | 1 | 1 |
| Heap | 17 | 16 | 16 |
| Stack | 4 | 4 | 4 |
| Total | 22 | 21 | 21 |

# 4 Power Consumption

| **Average current** | **Current (mA)** | | |
|---|---|---|---|
| *Preset* | *Small* | *Medium* | *Large* |
| 0.1 Hz | 2.8 | 0.96 | 0.50 |
| 1.0 Hz | 25.2 | 9.4 | 4.7 |

## 5 Disclaimer

The information herein is believed to be correct as of the date issued. Acconeer AB (“Acconeer”) will not be responsible for damages of any nature resulting from the use or reliance upon the information contained herein. Acconeer makes no warranties, expressed or implied, of merchantability or fitness for a particular purpose or course of performance or usage of trade. Therefore, it is the user’s responsibility to thoroughly test the product in their particular application to determine its performance, efficacy and safety. Users should obtain the latest relevant information before placing orders.

Unless Acconeer has explicitly designated an individual Acconeer product as meeting the requirement of a particular industry standard, Acconeer is not responsible for any failure to meet such industry standard requirements.

Unless explicitly stated herein this document Acconeer has not performed any regulatory conformity test. It is the user’s responsibility to assure that necessary regulatory conditions are met and approvals have been obtained when using the product. Regardless of whether the product has passed any conformity test, this document does not constitute any regulatory approval of the user’s product or application using Acconeer’s product.

Nothing contained herein is to be considered as permission or a recommendation to infringe any patent or any other intellectual property right. No license, express or implied, to any intellectual property right is granted by Acconeer herein.

Acconeer reserves the right to at any time correct, change, amend, enhance, modify, and improve this document and/or Acconeer products without notice.

This document supersedes and replaces all information supplied prior to the publication hereof.