A121 Touchless Button Reference Application User Guide

User Guide

A121 Touchless Button 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 - Understanding RSS API functions
	User guides (PDF)	
A121 Assembly Test	Describes the Acconeer assembly test functionality.	- Bring-up of HW/SW - 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 - Use case evaluation
	Readme (txt)	
README	Various target specific information and links	- After SDK download

2 Touchless button

The purpose of the touchless button reference application is to employ the A121 sensor as a contactless button. This algorithm proves useful in scenarios where registering a button press is needed without physically touching a surface. A prime example would be in public spaces.

Moreover, the algorithm is designed to automatically recalibrate itself when static objects obstruct the sensor, preventing prolonged and erroneous detections. An initial calibration of the background is performed before starting to detect button presses. As long as no presses are detected the background will be continuously updated. If a button press is detected for a long period of time (*calibration_interval_s* seconds), the background will be reset since this is considered a change in the environment. If the initial calibration is done with something in front of the sensor, this will most likely cause false detects when removed, but the calibration will be reset and a new calibration will be performed. The algorithm is very dynamic to be able to adapt to changes in the environment.

2.1 Measurement Range and Presets

The reference application is designed to have two different detection ranges: one in close proximity to the sensor and another farther away. Users can choose which range, or even both, to activate simultaneously.

The close range is defined as the zone from the sensor up to 5 centimeters, while the far range spans from the sensor to roughly 24 cenitmeters. These ranges are presented in three preset configurations within the Exploration Tool: one for exclusive close range detection, another for exclusive far range detection, and a third for detecting in both ranges.

It's important to note that the algorithm can also accommodate extended ranges, provided the sensor settings are appropriately adjusted to encompass a larger distance.

2.2 Configuration

Each detection range is composed of a single subsweep, allowing for independent configuration and adjustment of each range.

The *measurement_type* processing parameter is used to activate a specific range and set which patience and sensitivity processing parameters to apply during the processing.

The sensitivity parameters (*sensitivity_close* and *sensitivity_far*) establish the detection threshold for each range. Meanwhile, the patience parameters (*patience_close* and *patience_far*) dictate the consecutive frame count above the threshold required to recognize a button press, as well as the consecutive frames below the threshold to signify the end of a button press action.

2.3 Calibration

Each subsweep is comprised of multiple points (`num_points`), with each point corresponding to a distance in space where reflected pulses are measured. These points undergo continuous individual calibration. The threshold is normalized for each point by evaluating the standard deviation in the number of sweeps received during the *calibration_duration_s* period. This threshold normalization remains dynamically updated as long as no detections occur within any range. Consequently, there might be slight variations in the processor result when both ranges are simultaneously active compared to their separate activation on the same data.

To change the calibration to include more or fewer frames, the configuration parameter *calibration_duration_s* should be increased or decreased respectively. Note that during the calibration no button press actions should be performed since the purpose of the calibration is to record the background noise.

The *calibration_interval_s* configuration parameter establishes the maximum time interval in seconds between successive calibrations. When a consecutive detection reaches the time limit set by *calibration_interval_s*, a fresh calibration is initiated. The purpose of the parameter is to adjust the normalization of the detection threshold to effectively respond to significant environmental changes, such as the introduction of static objects within the detection range. Therefore, *calibration_interval_s* should not be set lower than the estimated duration of the longest continuous detection event.

2.4 Processing

For every frame, the processor evaluates whether the sweeps significantly surpass the threshold or not. In the selected range or ranges, a frame is recorded as significant when a minimum of two sweeps at the same distance surpass the threshold within the same frame. The patience settings ($patience\_close$ and $patience\_far$) dictate the number of consecutive significant frames needed for the event to be deemed as a valid detection (button press). Similarly, it specifies the number of consecutive frames required to be nonsignificant to signal the end of a detection event. Increasing the patience setting results in the button detecting prolonged presence in front of the sensor, consequently reducing its responsiveness. However, it decreases the risk of false detections if short sporadic noise appears.

Since the data from the A121 sensor is complex, each data point includes both phase and amplitude information. The threshold takes advantage of both the real and imaginary part of the data and can be seen as an circular boundary in the complex plane. A data point can pass the threshold by either a shift in phase (which would be caused by movement), a shift in amplitude (which would be caused by a more reflecting object) or both at the same time. Which in turn will trigger a detection. The placement of the circular boundary in the complex plane is determined by the mean and standard deviation of the calibration frames measured during the time set by $calibration\_duration\_s$ and the radius of the boundary is set by the sensitivity parameters ($sensitivity\_close$ and $sensitivity\_far$). Opting for a high sensitivity setting results in a smaller radius, leading to a lower threshold. Conversely, a low sensitivity setting produces a larger radius, subsequently yielding a higher threshold.

2.5 Results

The algorithm will provide six different results, three for each range: detection, threshold and detection score.

The result parameters `detection_close` and `detection_far` will simply declare if there is detection within the range. The parameters will be `True` for detection, `False` for no detection and `None` if the range is not activated.

The detection scores parameters will provide the detection score for each point in each subsweep. The output shape will therefore be (`sweeps_per_frame`, number of points in current subsweep). The detection score will be `None` if the range is not activated.

The parameters `threshold_close` and `threshold_far` gives the threshold to which the detection scores are compared against. The thresholds are inversely proportional to the sensitivity parameters where `threshold_close = 10 /` $sensitivity\_close$ and `threshold_far = 10 /` $sensitivity\_far$.

2.6 GUI

In the GUI two plots are displayed, see Figure 1. The top plot shows the detection duration for the close and far range. The displayed range can be changed by switching the *Range* parameter under *Processor parameters* in the GUI or switch preset under *Preset Configurations*.

The bottom plot displays the detection score for each distance point in each range. To hide thresholds or points click on the corresponding symbol in the legend. The points represent the 2nd highest detection score per frame for each distance. The 2nd highest score is chosen for plotting since a frame counts as significant after two points at the same distance pass the threshold during the same frame. A detection in the activated range(s) will be shown in the top plot when the consecutive number of points above the threshold is greater or equal to the patience parameter for the activated range(s). The purpose of the bottom plot is to demonstrate the effect of the sensitivity settings and to give the user an idea of which points are most important for the user's detection scenario. The sensitivity settings might need to be adjusted depending on integration and purpose of the application. The sensitivity will always result in a trade-off between missed detections and false detections.

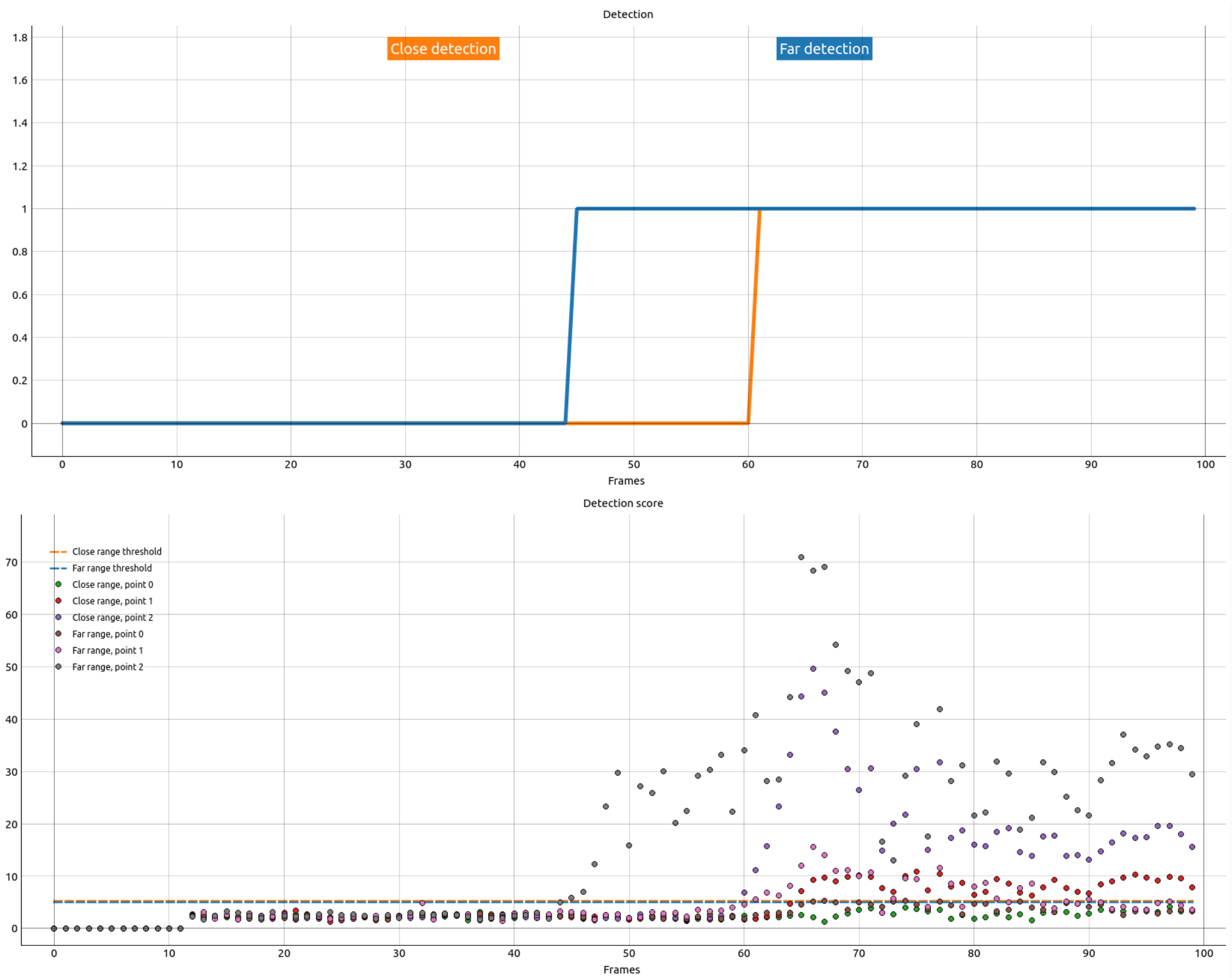

Figure 1: Example of the touchless button GUI. Detection is found in both the close and far range. The threshold for each range and the detection score for each distance is shown in the lower plot.

2.7 Tests

The following section presents the results from various tests on the algorithm.

Presets range test

The purpose of this test was to check that no detections are made outside the appointed range for each preset. Close range should not have detections outside 0.05 m and far range should not have detections outside 0.24 m.

Test setup

For this test an A121 EVK (XC120 + XE121) was used. To test the presets a corner reflector was moved just outside the edge of the appointed ranges (0.05 m and 0.24 m), see Figure 2.

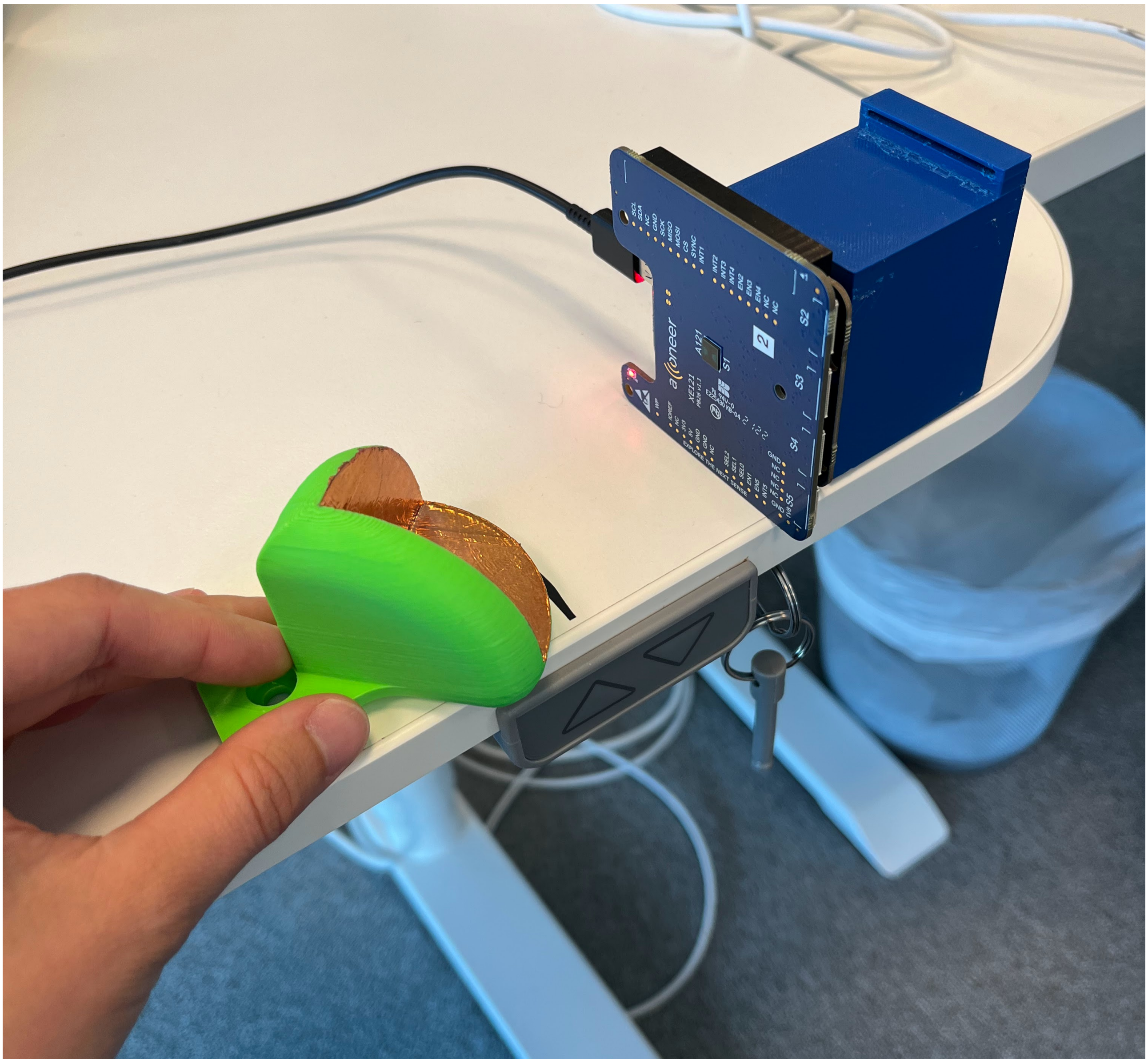

Figure 2: Test setup to test preset ranges, to ensure no detection outside the range.

Configurations

The configurations below corresponds to the presets in Exploration Tool, see Table 2.

Table 2: Touchless button configurations.

Parameter	Close range	Far range
Sensitivity	1.9	2.0
Patience	2	2
Calibration duration	0.6 s	0.6 s
Calibration interval	20.0 s	20.0 s
Sweeps per frame	16	16
Sweep rate	320 Hz	320 Hz
Inter sweep idle state	Ready	Ready
Inter frame idle state	Ready	Ready
Continous sweep mode	True	True
Double buffering	True	True
Start point	0	0
Number of points	3	3
Step length	6	24
HWAAS	40	60
Profile	1	3

Results

No detection outside any of the ranges.

Persons test

The algorithm was tested on 10 different people to evaluate the functionality of the algorithm.

Test setup

For this test an A121 EVK (XC120 + XE121) was used integrated with a blinkstick to give the user direct response on their action. The setup was encapsulated in a 3D-printed cover. No lens was used. See Figure 3.

10 different people were asked to perform a tap with the back of their hand/fingers towards the sensor, as if they were tapping a button to open a door on for example a buss or a train. The action counted as detected if either or both of the ranges (close range and far range) detected the action.

Figure 3: Shows the setup and action used in the persons test.

Configurations

This test utilized the “Close and far range” preset in Exploration Tool. This preset uses two subsweeps, the subsweep configurations can be seen in Table 2.

Results

Table 3: Results from persons test.

	Number of detections	Number of actions
Person 1	10	10
Person 2	10	10
Person 3	10	10
Person 4	10	10
Person 5	10	10
Person 6	10	10
Person 7	10	10
Person 8	10	10
Person 9	10	10
Person 10	10	10

Comments regarding changes in temperature

Changing temperature will affect the SNR of the signal. At lower temperatures the SNR is increased and at higher temperatures the SNR is decreased. Some actions will therefore trigger detection easier at lower temperatures and the sensitivity can threrefore be set lower at these termperatures. It is therefore favorable to set the sensitivity according to the desired responsiveness at the highest estimated temperature for the intended integration. The sensitivities selected for the presets were chosen to minimize missed detections and false detections in the range -10°C and 50°C. The evaluation was made using 8 different sensors at three different temperatures: -10°C, 25°C and 50°C.

3 Memory

3.1 Flash

The reference application compiled from ref_app_touchless_button.c on the XM125 module requires around 73 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_touchless_button.c.

RAM	Size (kB)		
Preset	*Close*	*Far*	*Close and Far*
Static	1	1	1
Heap	7	7	10
Stack	3	3	3
Total	11	11	14

4 Power Consumption

Average current	Current (mA)		
Preset	*Close*	*Far*	*Close and Far*
	68	68	70

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.