life.augmented

FlightSense
ST time-of-flight ranging sensors
Smart Presence Detection

STMicroelectronics
남경태 부장

Agenda

1 **About dTOF**

- Intro ST TOF
- TOF family
- VL53L7CX demo GUI

2 **Smart Presence Detection (SPD)**

- Introduction
- SPD package content

3 **SPD Demo**

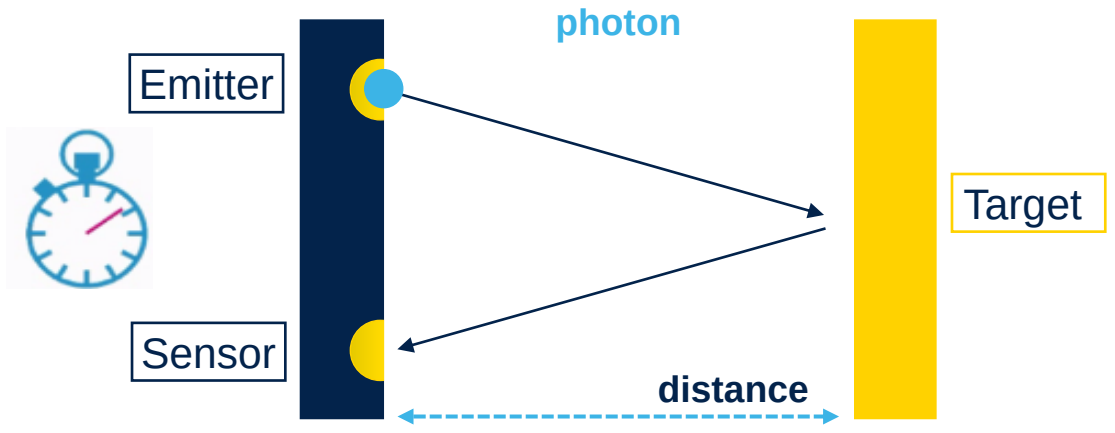
- Running SPD GUI for your demo
- Running SPD in terminal

4 **SPD in details**

- SPD in details
- SPD Programming Guide

FlightSense*
… Making Light work

Time-of-Flight Principle



Measured distance = Photon travel time / 2 x Speed of light

1cm round-trip takes 67ps

- ST proprietary FlightSense technology
- True distance measurement
  Independent of target size, color & reflectance
- Fast and low power
- Truly invisible 940nm illumination

life.augmented

FlightSense typical module overview

All-in-One (illumination & sensor) Time of Flight System
→ Optimized Size/ Perf/ Cost mix

Advanced optics with integrated IR filter

State-of-art assembly & testing manufacturing ST line in SHZ

High power VCSEL
Full Class 1 safety

Monolithic ToF SoC, SPAD Array, RAM/ROM & powerful Class1 VCSEL driver

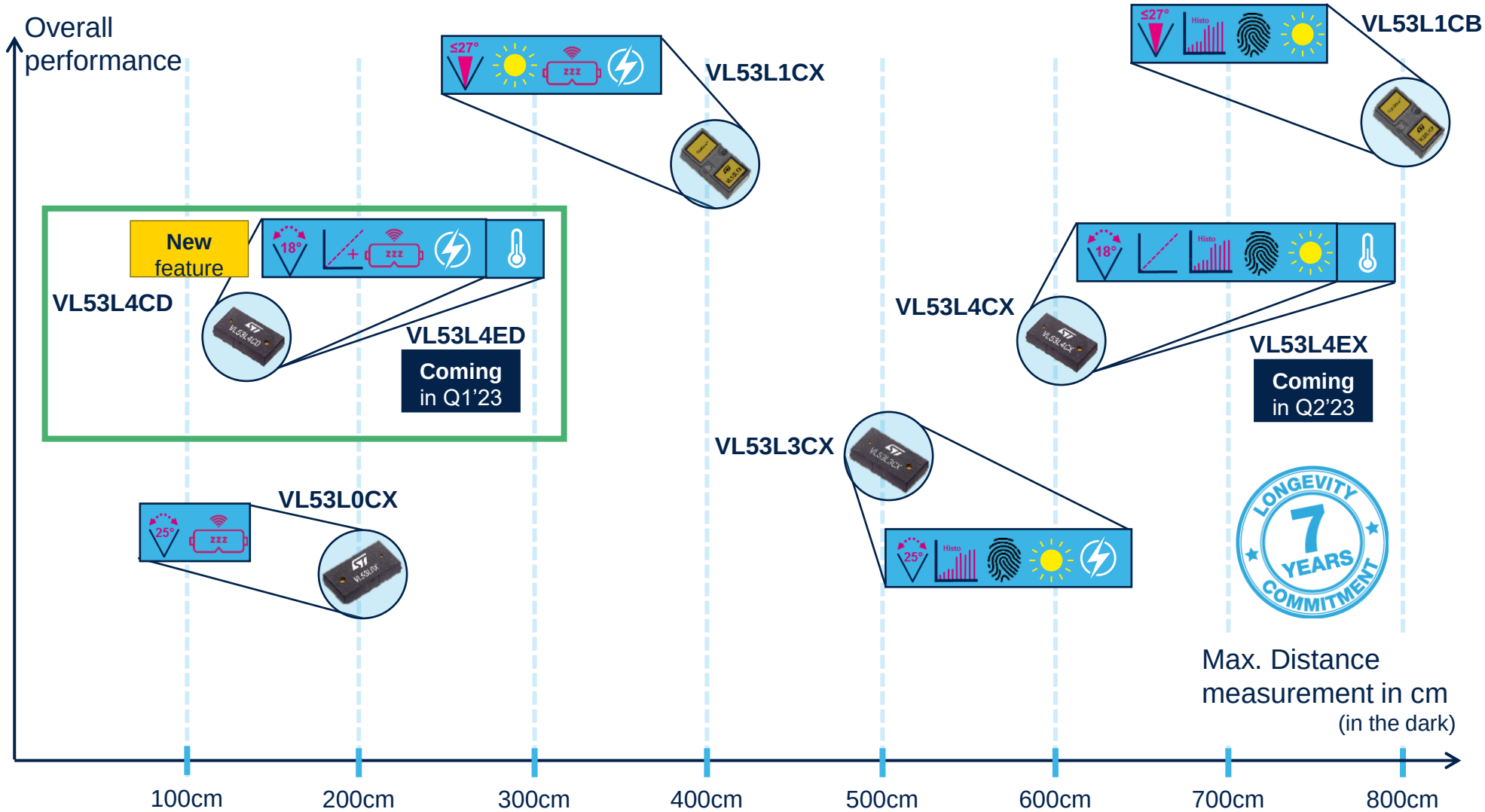
life.augmented

FlightSense ST time-of-flight ranging sensors / mass-market roadmap

life.augmented

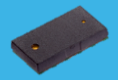
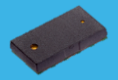
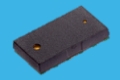
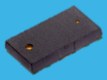
FlightSense™ Single-Zone portfolio

- XX° FoV
- ≤XX° Programmable FoV
- Histogram MultiObject detection
- Smudge correction
- Perf. Under Ambient (>100cm)
- Autonomous mode
- Ultralow Power Mode
- Close distance Linearity
  + : High Performance in short distance
- Extended Temperature range
- Ambient Light Sensing



life.augmented

FlightSense™ Single-Zone portfolio

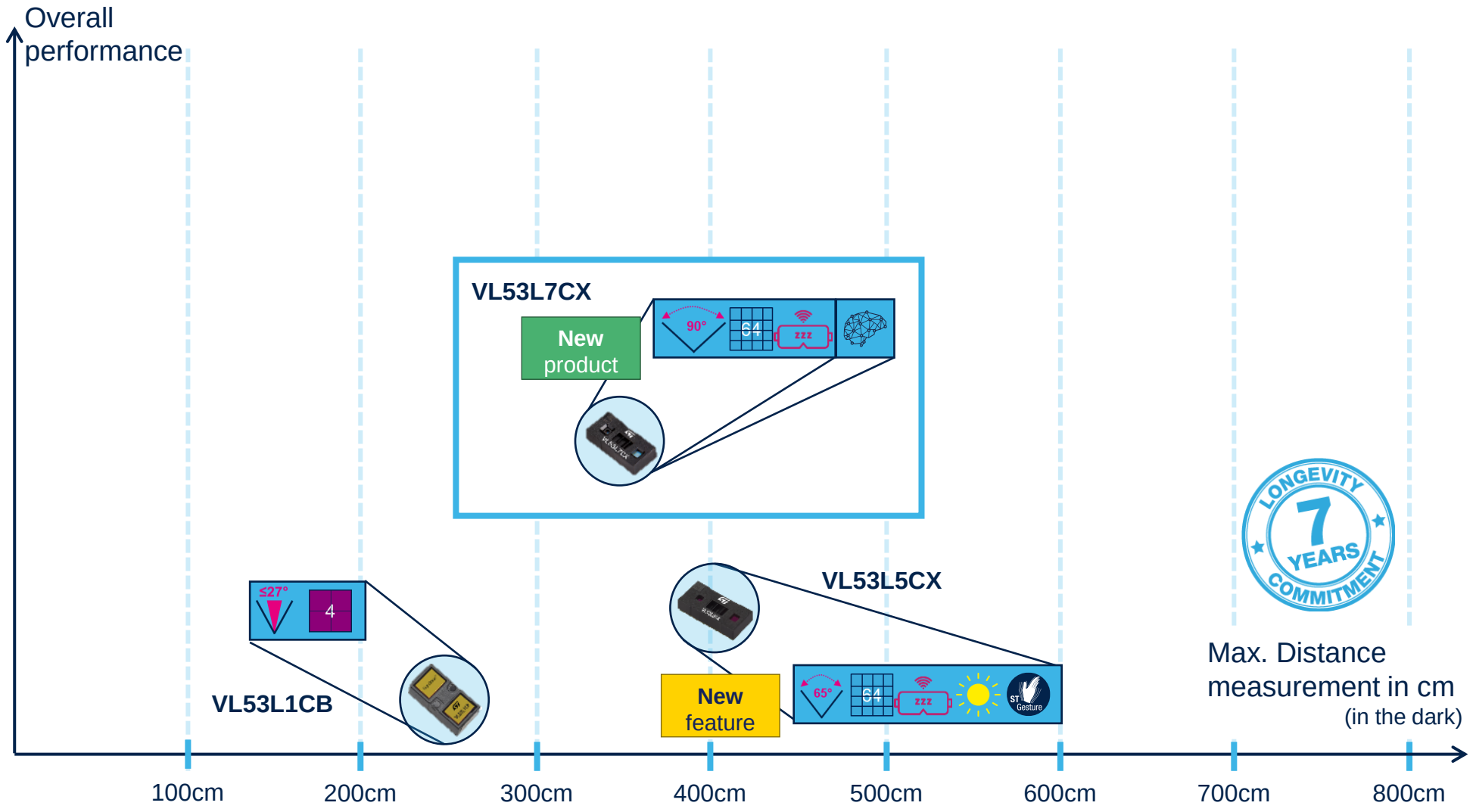
	VL6180V1	VL53L4CD	VL53L0CX	VL53L1CX	VL53L3CX	VL53L4CX	VL53L1CB
	 Proximity sensor with low power consumption	 Proximity sensor with high accuracy	 Ranging sensor	 Ranging sensor programmable FoV	 Ranging and multi-target sensor	 Short to long ranging, multi-target sensor	 Long Distance and multi-target sensor
Max distance measurement	**60 cm**	**130 cm**	**200 cm**	**400 cm**	**500 cm**	**600 cm**	**800 cm**
Field-of-View	25° No Lens	18° No Lens	25° No Lens	27° **(SW config)** Lens on Rx	25° No Lens	18° No Lens	27° **(SW config)** Lens on Rx
Perf. under ambient light (along windows with strong outside light)	10cm	60cm	80cm	**++** **(135cm)**	**++** **(140cm)**	**++++** **(180cm)**	**+++** **(160cm)**
Close distance linearity	**+++** **>1.0cm**	**+++++** **>0.1cm**	++ >2.5cm	+ >2.5cm	++ >2.5cm	**+++** **>1.0cm**	++ >2.5cm
Min. distance detection	**0cm**	**0cm**	<1.5cm	**1.0cm**	**1.0cm**	**0cm**	**1.0cm**
Multi-target detection (Histogram)	No	No	No	No	**Yes**	**Yes**	**Yes**
Crosstalk / Smudge immunity	Crosstalk compensation	Crosstalk compensation	Crosstalk compensation	Crosstalk compensation	**Immunity >80cm** <80cm: Smudge comp.	**Immunity >80cm** <80cm: Smudge comp.	**Immunity >80cm** <80cm: Smudge comp.
Power conso — Typical (act. ranging)	1.7mA	22mA	19mA	16mA	16mA	19mA	16mA
Power conso — Autonomous mode (1Hz)	170µA	240µA	340µA	395µA	-	-	-
Power conso — ULP (Ultra Low Power)	-	**55µA**	-	**65µA**	**55µA**	-	-
Memory footprint	**ROM 4.5kB** **RAM 0.44kB**	**ROM 2.7kB** **RAM >0.1kB**	ROM 23.4kB RAM 1.2kB	**ROM 3.3kB** **RAM 0.03kB**	ROM 36kB RAM 11.2kB	ROM 36kB RAM 11.2kB	ROM 47.6kB RAM 9.8kB
Product variants Same HW, different features	VL6180X ambient light sensing	VL53L4ED extended temp. range (-40°/+105°)	-	-	-	VL53L4EX extended temp. range (-40°/+105°)	-
All-in-one module size	4.8 x 2.8 x 1.0 mm	4.4 x 2.4 x 1.0 mm	4.4 x 2.4 x 1.0 mm	4.9 x 2.5 x 1.56 mm	4.4 x 2.4 x 1.0 mm	4.4 x 2.4 x 1.0 mm	4.9 x 2.5 x 1.56 mm

FlightSense™ Multi-Zones portfolio

- XX° FoV
- Number of zones
  - x In sequential mode
- Perf. Under Ambient (>100cm)
- Autonomous mode
- Ultralow Power Mode
- Gesture recognition
- Suitable for AI

Common Features

- Histogram MultiObject detection



life.augmented

FlightSense™ Multi-Zone portfolio

	VL53L1CB	VL53L5CX	VL53L7CX
	 Long Distance and multi-target sensor	 Multizone ranging sensor with wide FoV	 Multizone ranging sensor with 90° FoV
Resolution — Max zones	**Up to 4 zones**	**Up to 64 zones**	**Up to 64 zones**
Resolution — Modes	**-**	**8x8 & 4x4**	**8x8 & 4x4**
Max distance measurement	250cm	400cm	350cm
Field-of-View	27° **(SW config)** Lens on Rx	**45° x 45° (Ø65°)** Lens on Rx + Tx	**60° x 60° (Ø90°)** Lens on Rx + Tx
Perf. under ambient light (along windows with strong outside light)	tbc	**+++** **(170cm)**	+ (65cm)
Close distance linearity	+ >2.5cm	**++** **>2.0cm**	**++** **>2.0cm**
Min. distance detection	**1.0cm**	**1.0cm**	**1.0cm**
Multi-target detection (Histogram)	**Yes**	**Yes**	**Yes**
Crosstalk / Smudge immunity	**Immunity >80cm** <80cm: Smudge comp.	**Immunity >60cm** <60cm: Crosstalk comp.	**Immunity >60cm** <60cm: Crosstalk comp.
Power conso — Typical (act. ranging)	16mA	216mW	216mW
Power conso — Autonomous mode (1Hz)	-	2.8mW / 8mW	2.8mW / 8mW
Memory footprint	ROM 47.6kB RAM 9.8kB	ROM 103.372KB RAM 4.096KB	ROM 103.372KB RAM 4.096KB
All-in-one module size	4.9 x 2.5 x 1.56 mm	6.4 x 3.0 x 1.5 mm	6.4 x 3.0 x 1.6 mm

VL53L7CX at a glance

First Multi-zone (64 zones) Time-of-Flight ranging sensor

Highlights

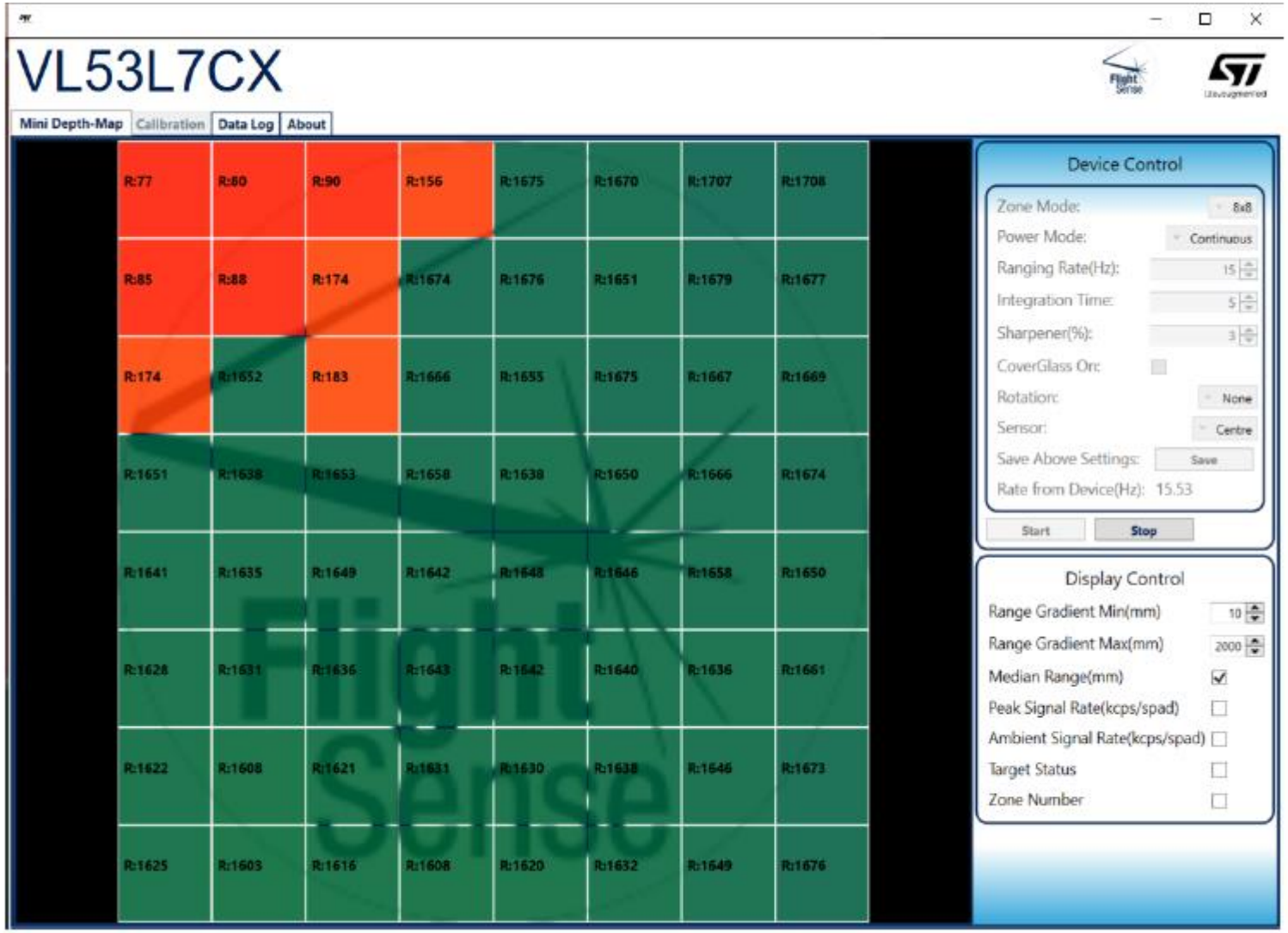
- **Parallel multizone ranging** output:
  - **4x4 or 8x8 zones** separate regions of interest
- **Ultra-wide FoV**: 60° x 60° (**90° diagonal**)
- Up to **350 cm ranging**
- **Multi-target detection** and distance measurement in each zone
- **Motion indicator** indicating if the target has moved
- **60 Hz** (4x4 zones) frame rate capability
- **Immunity to cover glass cross-talk** beyond 60cm
- **Pin-to-pin and driver compatible** with VL53L5CX

Package size : 6.4 x 3.0 x 1.6 mm
Square **FoV** : 60° x 60° (**90° diagonal**)
Multi-zone (8x8)

Max Distance measurement	**3.5 meters**
Close distance linearity	>2cm
Performance under ambient light (5klux)	**65cm**
Multi-target detection (Histogram)	**Yes, in each zone**
Crosstalk / Smudge immunity	Immunity >60cm <60cm: Smudge compensation
Power Consumption	95mA (continuous ranging) 1.3mA in Low power mode

life.augmented

VL53L7CX GUI demo



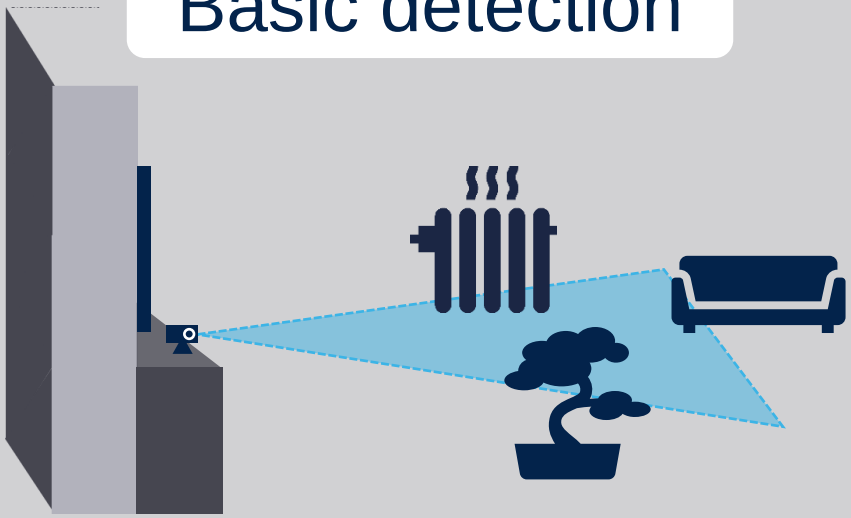
Smart Presence Detection

life.augmented

Why Smart Presence Detection?

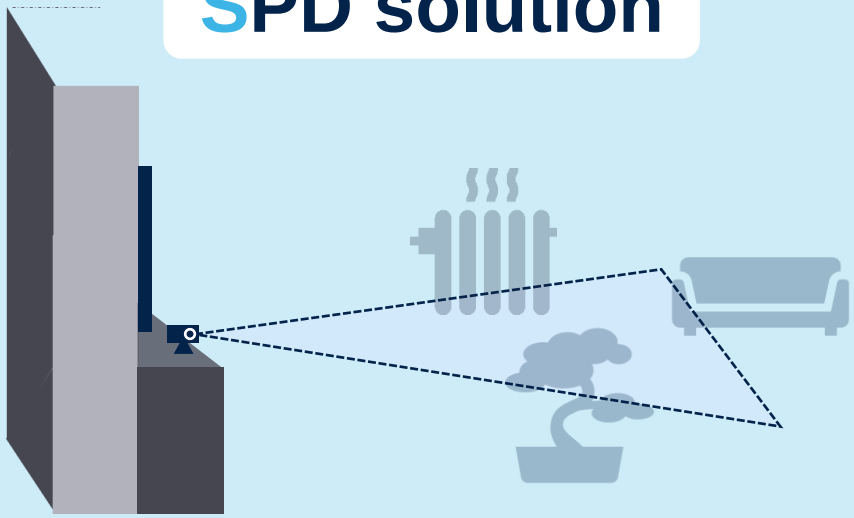
1 **Immune to the environment :** Wall / Plants / Furniture / Temperature

Basic detection



- Something Detected
- Presence = True

SPD solution



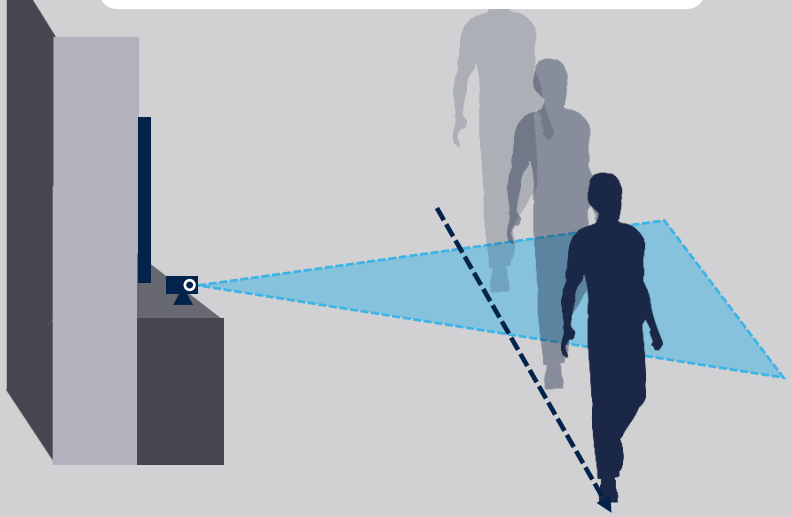
- Object Detected
- Presence = False

life.augmented

Why Smart Presence Detection?

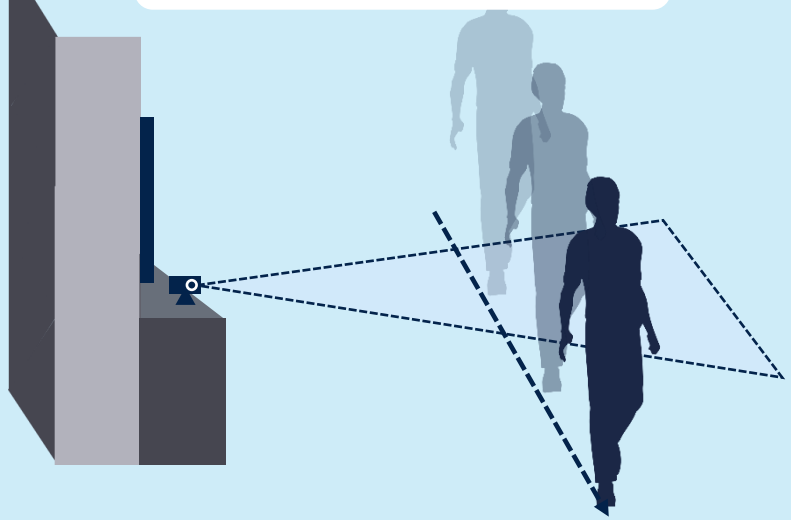
2 Filters Passing-By Users : Corridor / Lobby / Kitchen

Basic detection



- User entering the FoV
- Presence = True

SPD solution

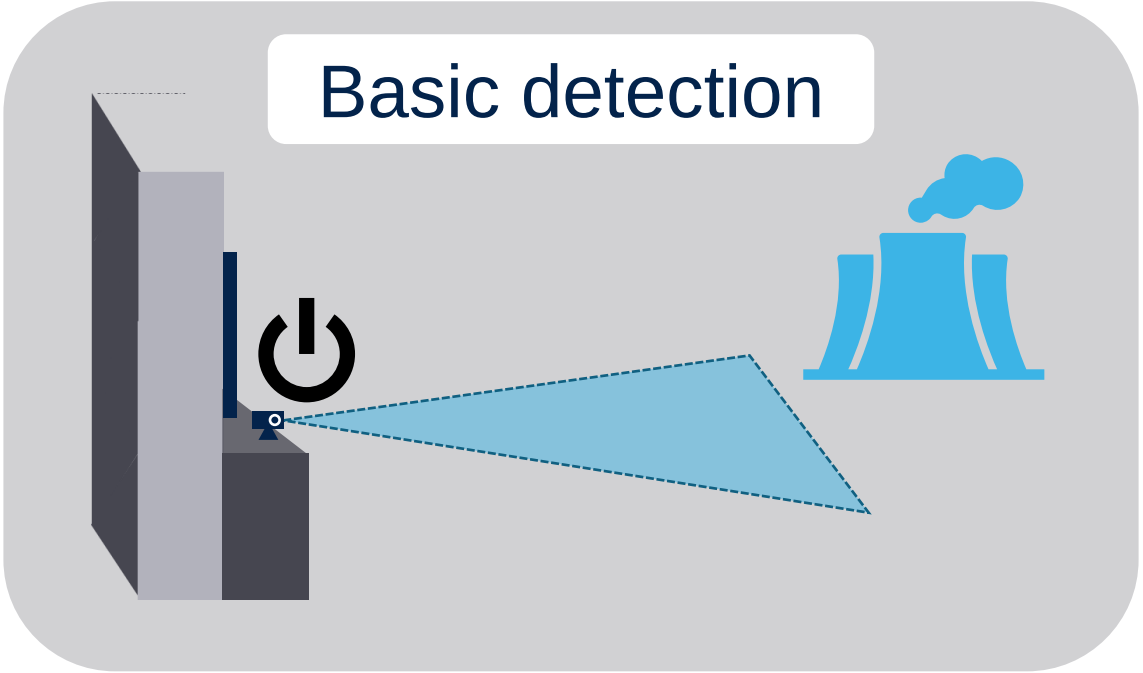


- User crossing the FoV
- Presence = False

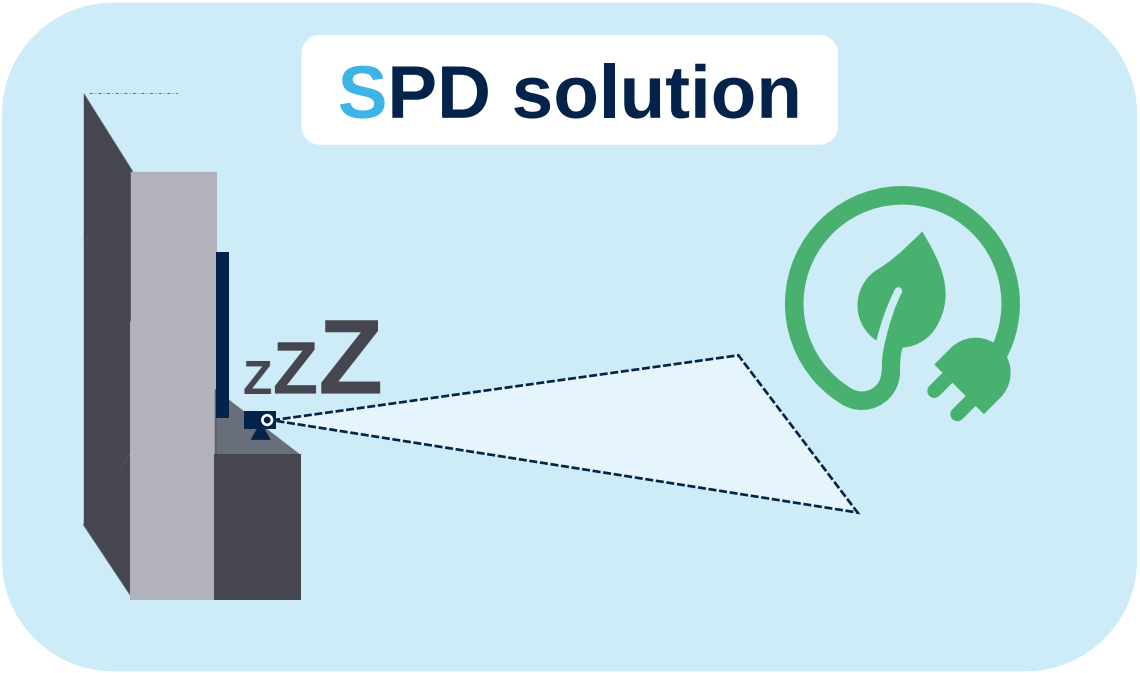
The FoV becomes a “Field Of Interest”

Why Smart Presence Detection?

3 **Runs in Low Power Autonomous Mode:** System going to sleep mode…



- Sensor Continuous Scanning
- MCU running



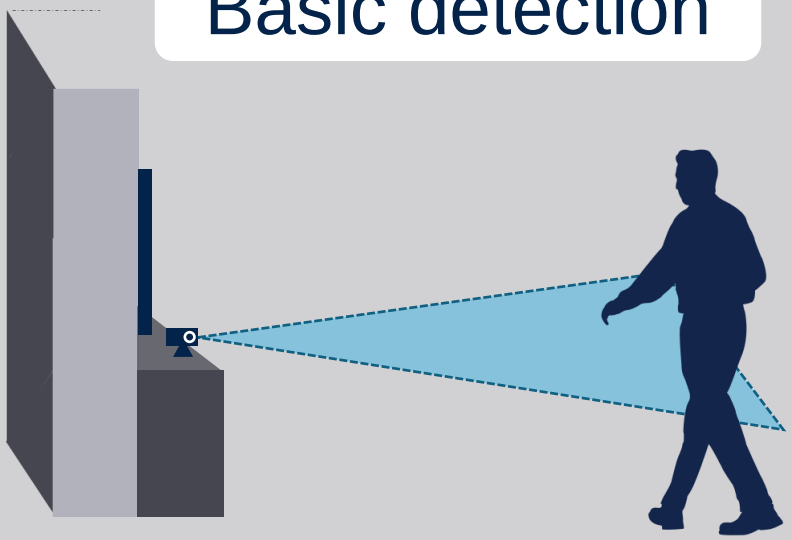
- Sensor in Autonomous Mode
- MCU sleeping
- Below 1mW

life.augmented

Why Smart Presence Detection?

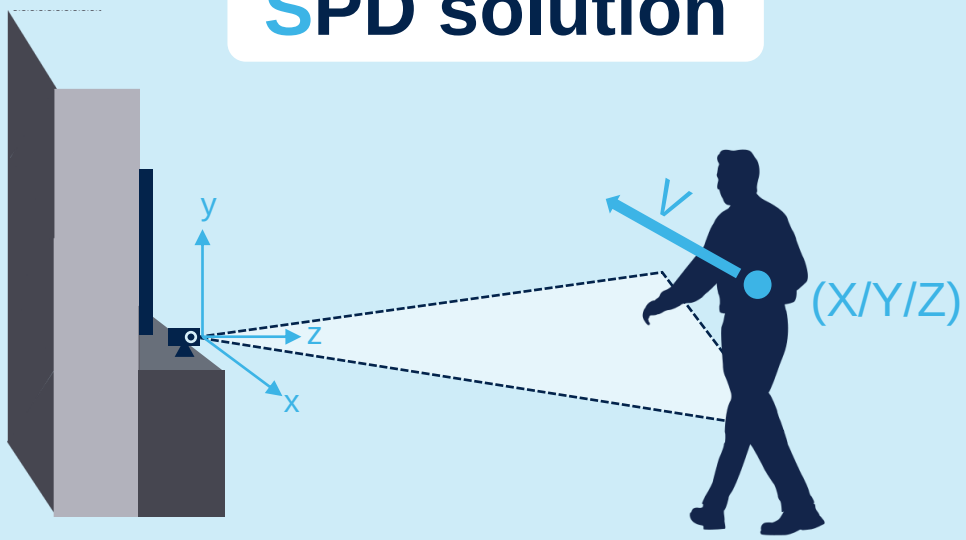
4 **Tracks users :** position and speed

Basic detection



➢ A user entering the FoV

SPD solution



➢ A user is entering the FoV at the position X/Y/Z and moving at V km/h

Why Smart Presence Detection ?
No camera / No image

1. **Immune to the environment**

2. **Runs in Low Power Autonomous Mode**

3. **Filters Passing-By Users**

4. **Tracks users**

5. **Identifies the number of users**

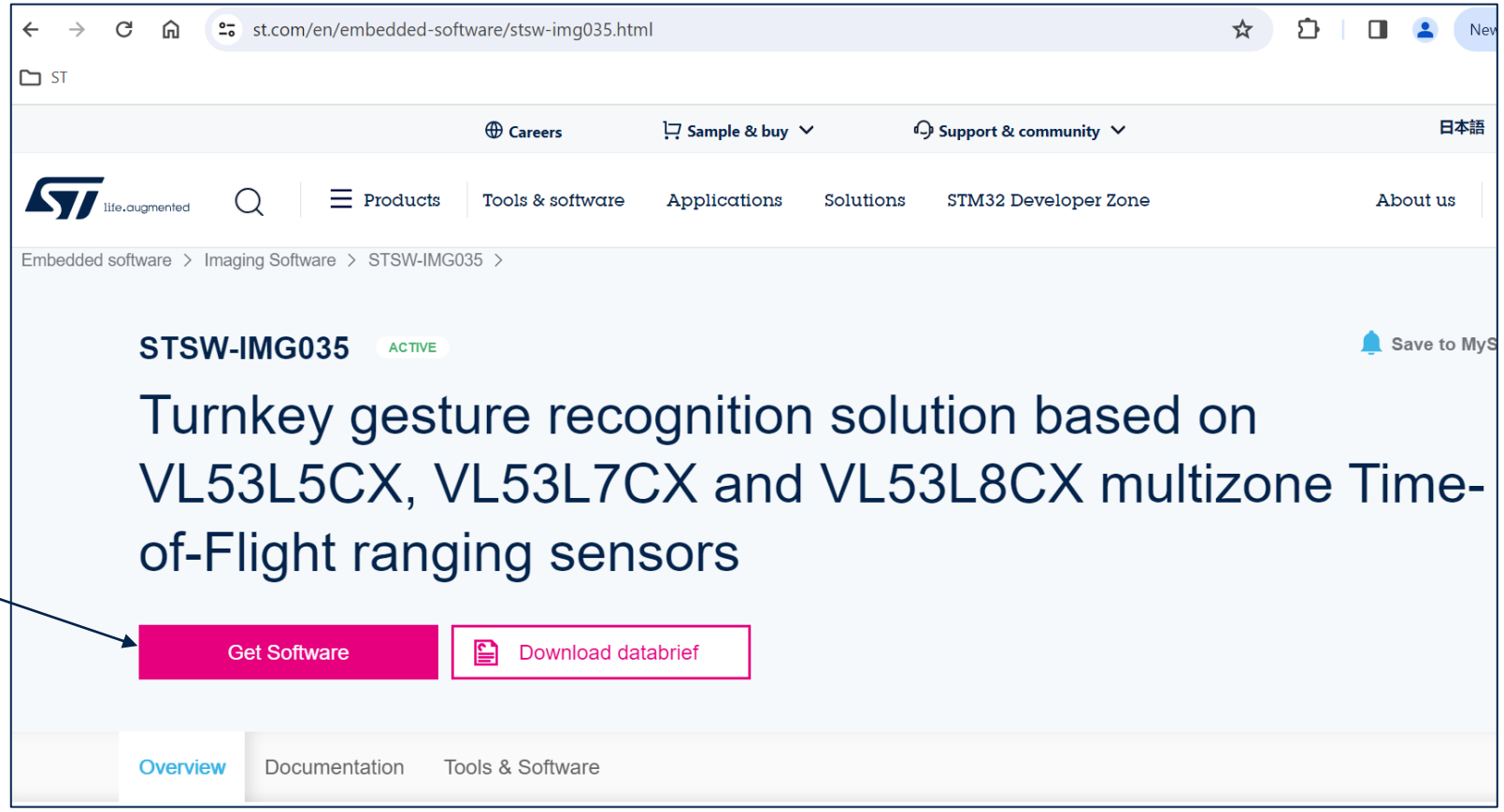
100% Privacy

SPD package content

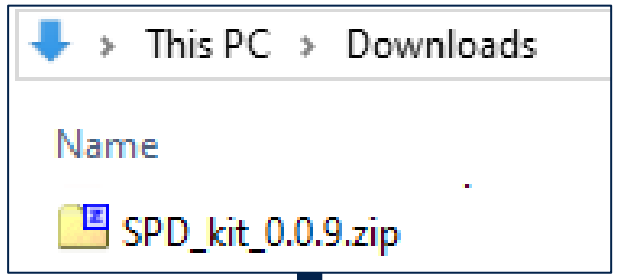
life.augmented

ST.com

SPD is a
all-in-one package

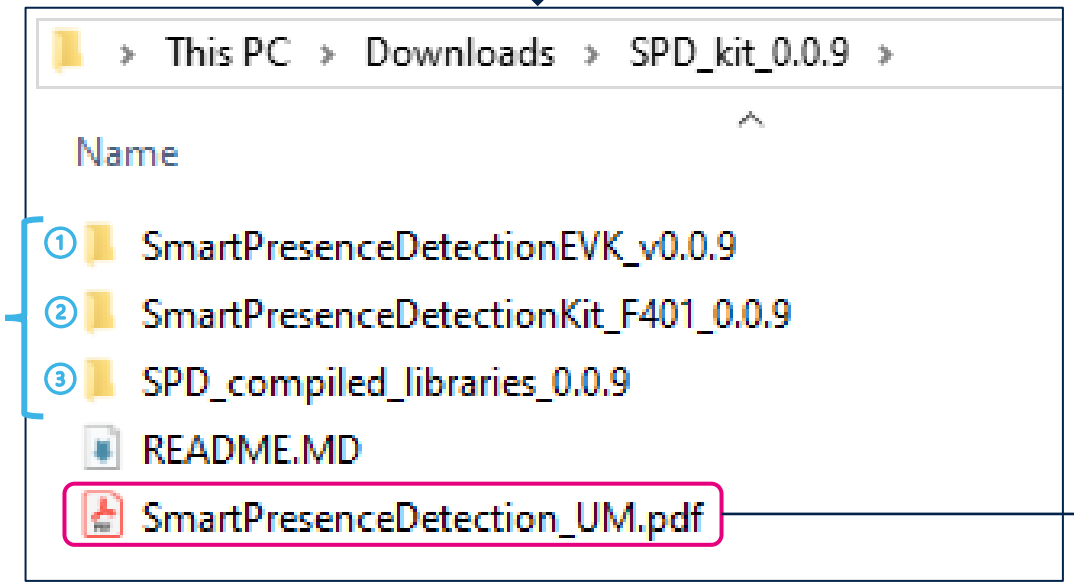


Package content

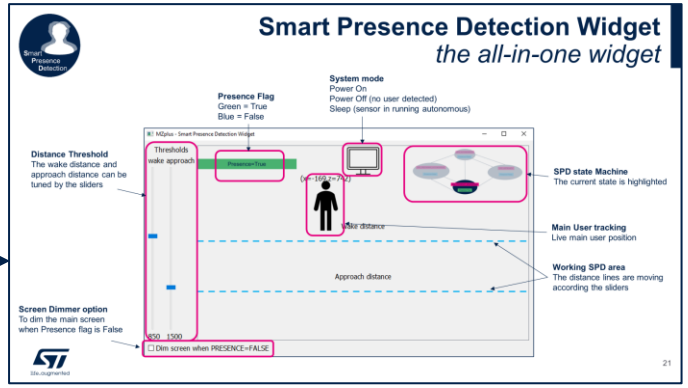


SPD is a
all-in-one package

Unzip it

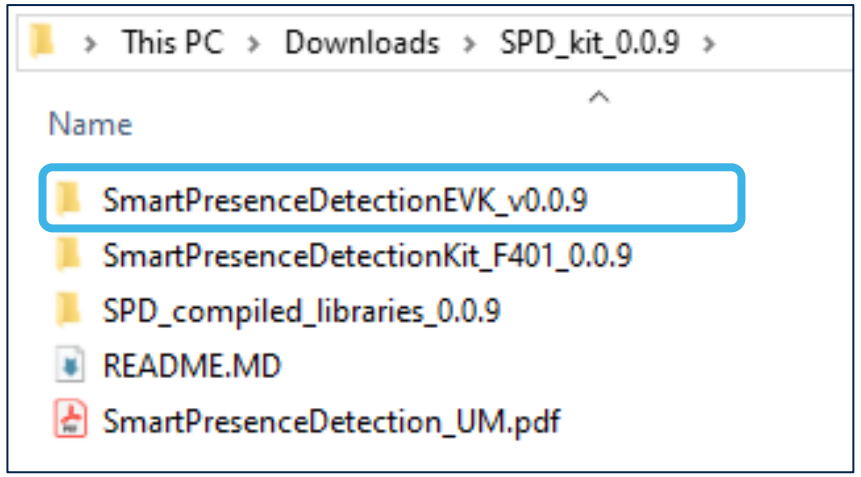


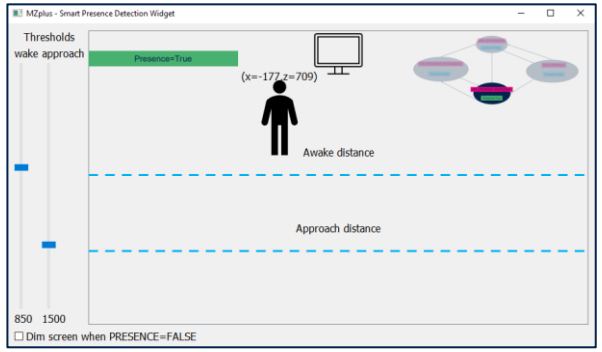
3 deliveries

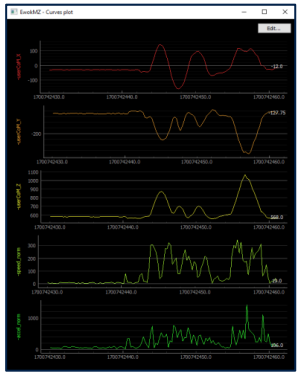


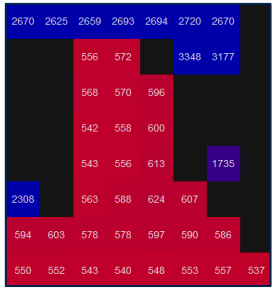
Package content

1 SmartPresenceDetection EVK GUI







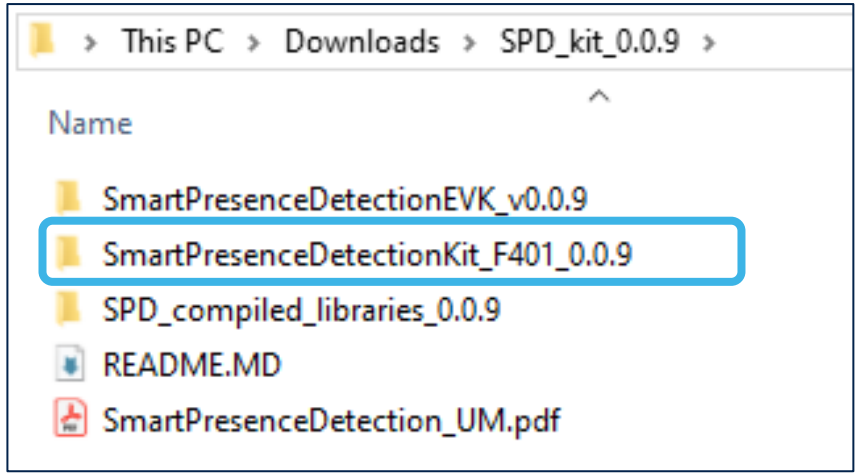


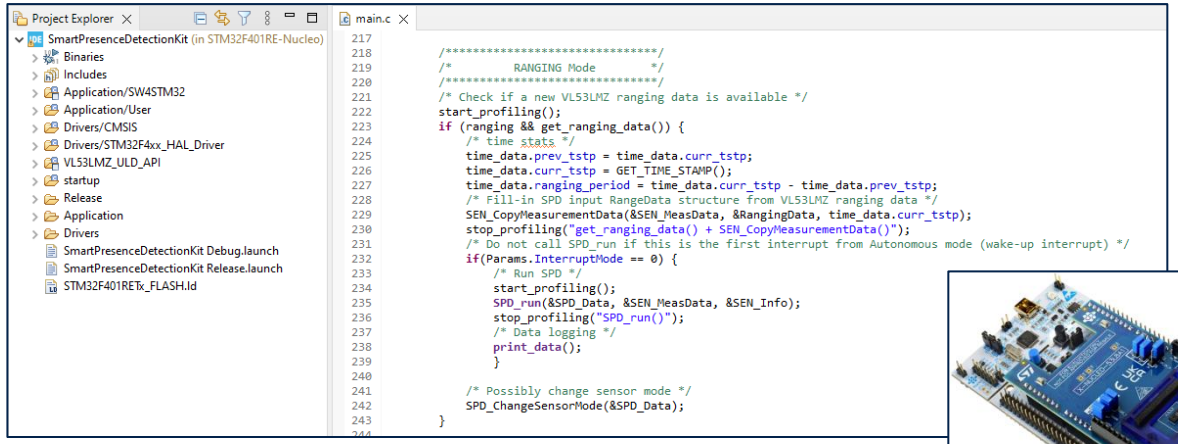
- Software User Interface for SPD
- Sensor data visualization
- SPD algo output monitoring
- SPD applicative demo
- Powerful datalogger

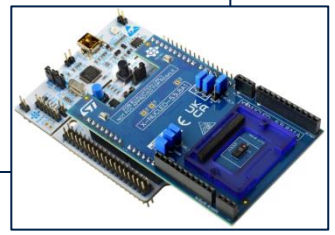
↪ User Manual: “Running in the EVK GUI”

Package content

2 SmartPresenceDetection Kit F401





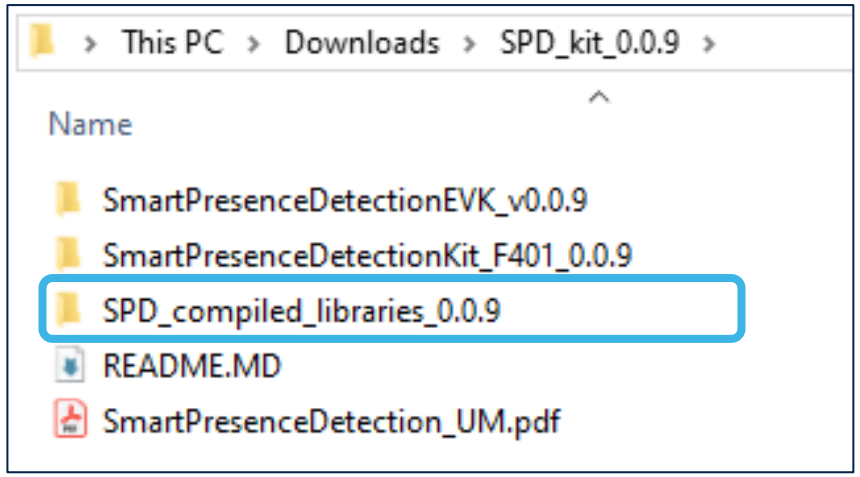


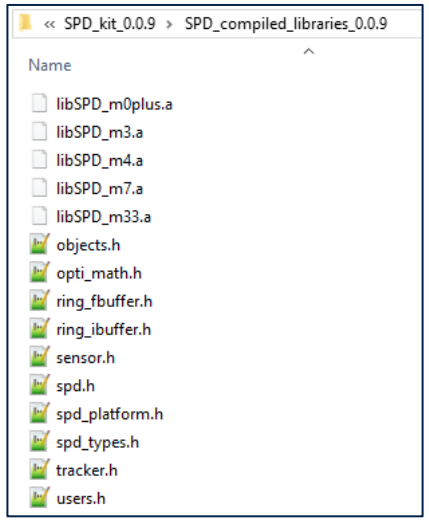
- Complete **C project** example
- P-NUCLEO-53L7/8A1 - STM32**F401**RE
- ST ToF MZ **generic ULD**
- **SPD algo integration**
- **Sensor configuration**

↪ User Manual: “SPD Library: Programming Guide”

Package content

3 SPD compiled libraries





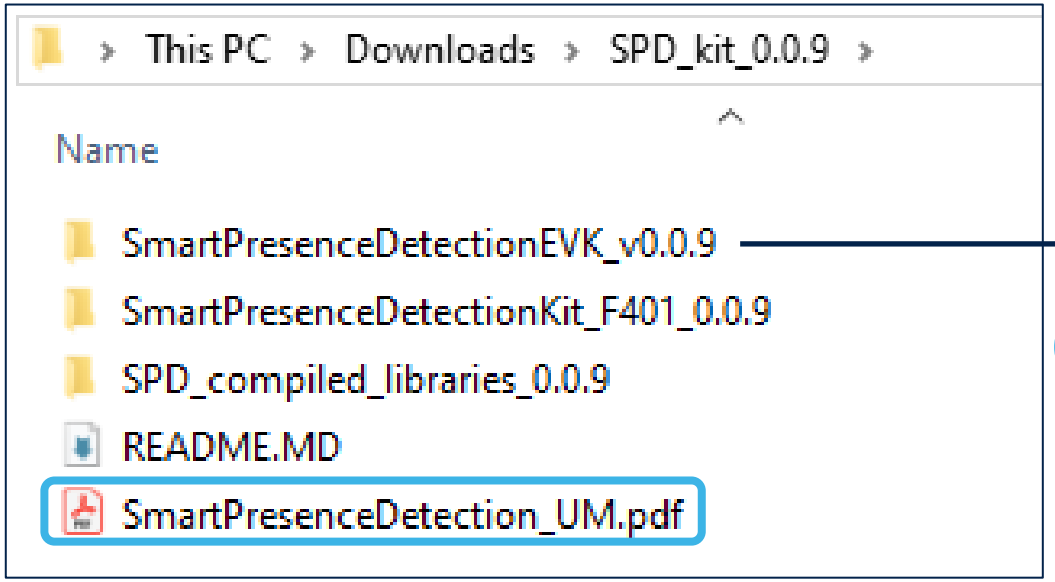
arm CORTEX-M

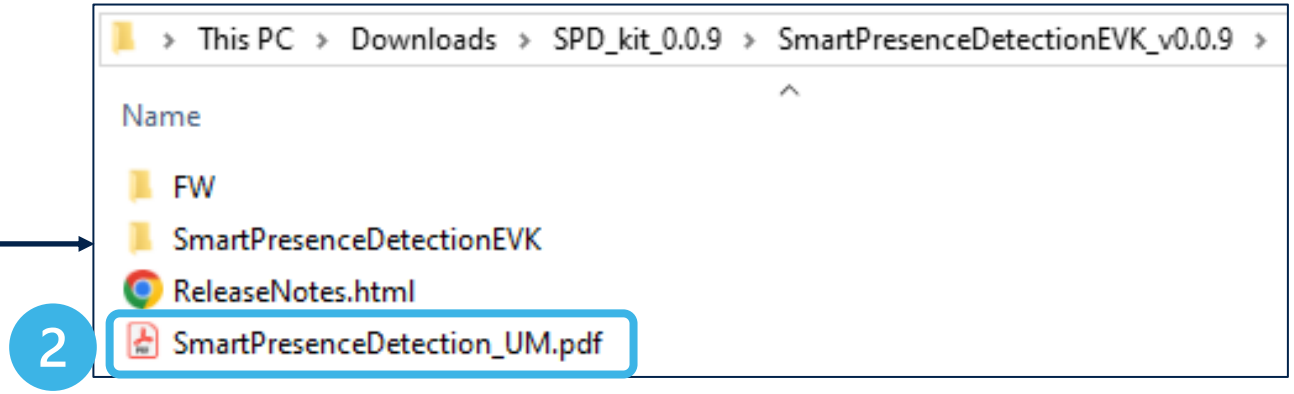
STM32

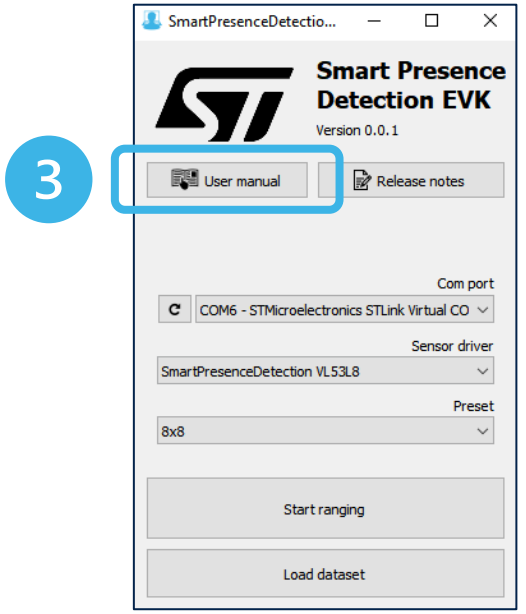
- ARM Cortex-M4
- ARM Cortex-M33
- ARM Cortex-M0+
- ARM Cortex-M7
- ARM Cortex-M3

Where to find the User Manual

~~User Manual~~ ➡ User Guide



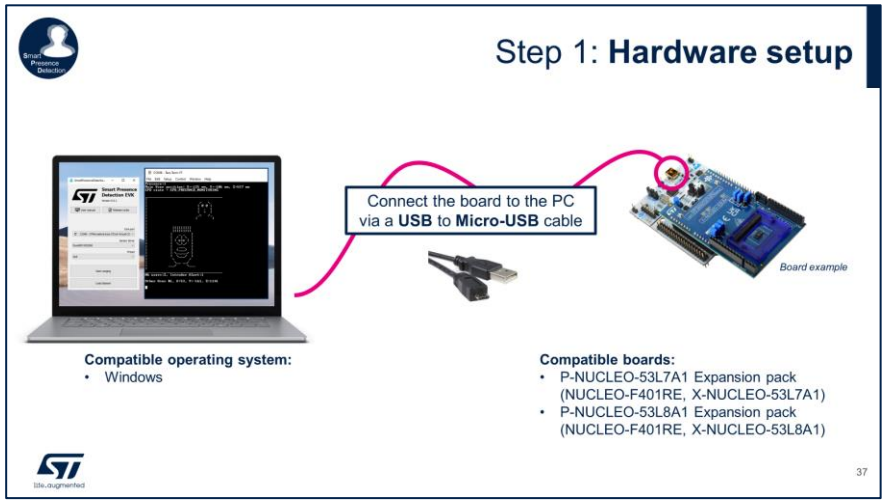


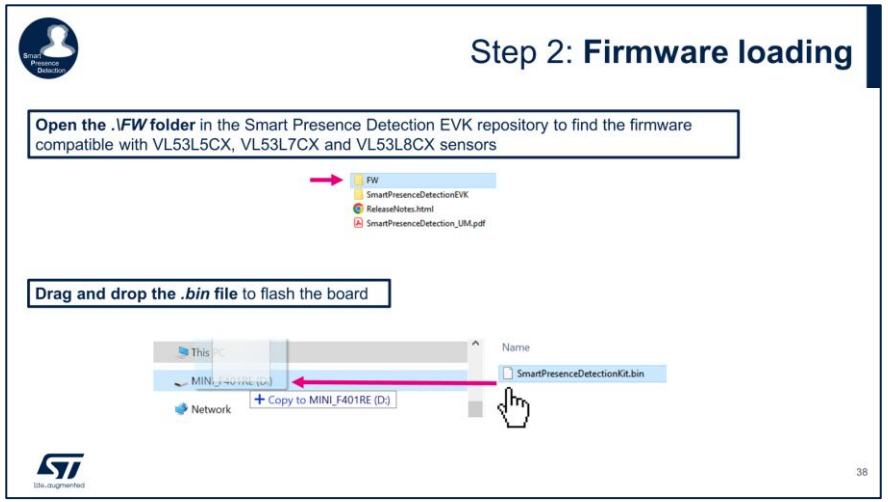


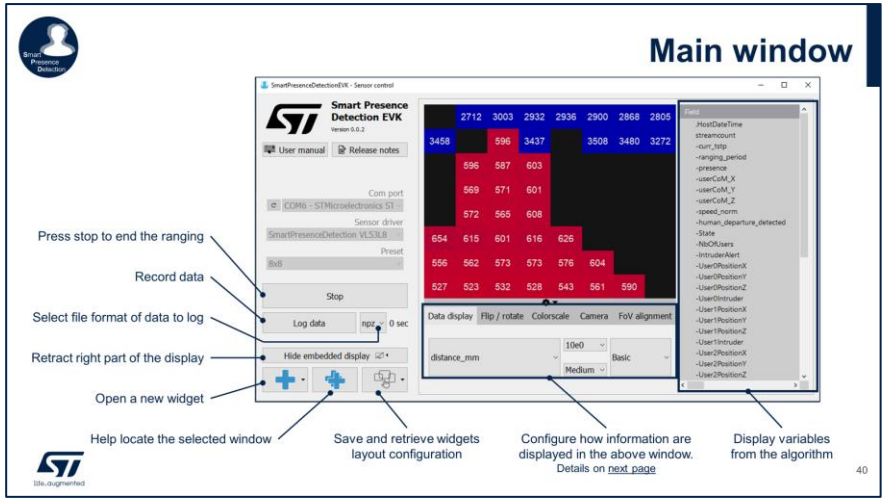
life.augmented

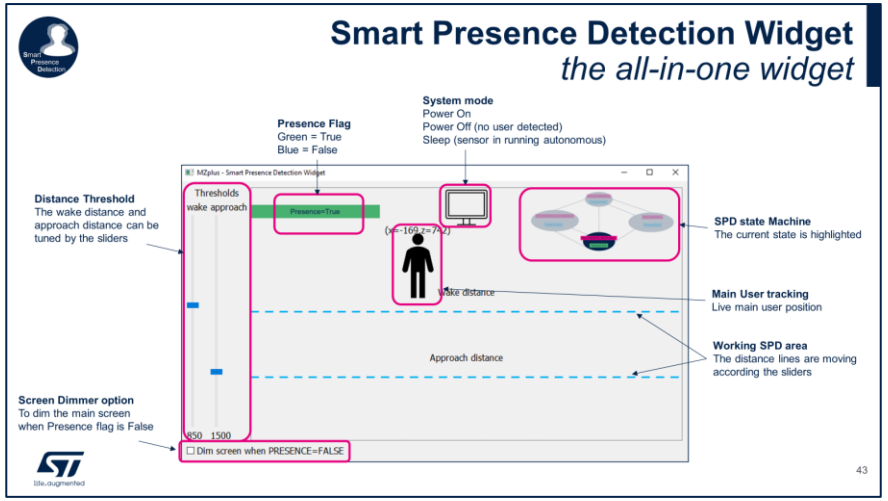
Running SPD GUI for your demo

User Guide: EVK Getting Started
Running in the EVK GUI

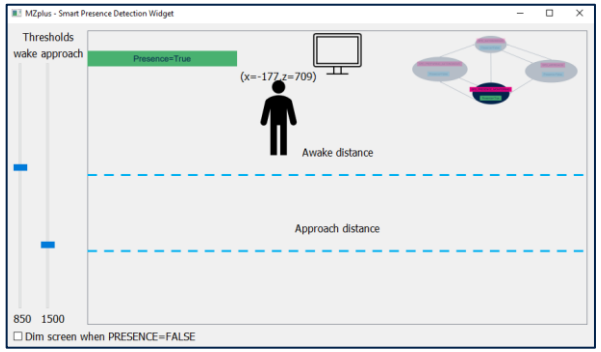






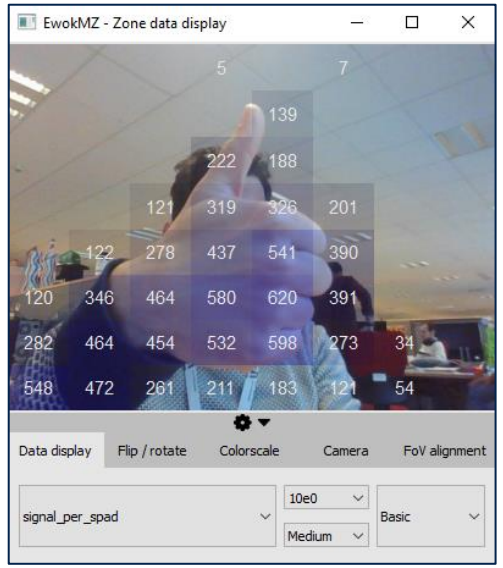


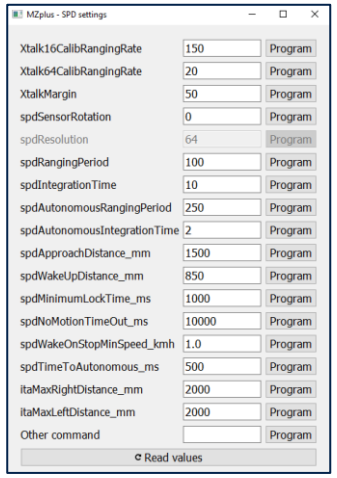
Smart System Detection Widgets



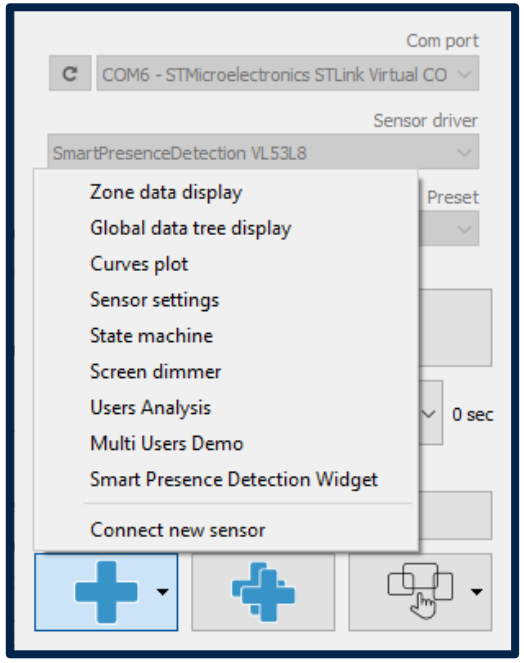
SPD widget
All in one widget:
- User tracking
- Distance thresholds tuning
- Presence Monitoring
- State Machine
- Screen dimmer

Zones display data
Open a display window (similar to the main one), connected to the same sensor.
This is usefull to display other data types in parallel

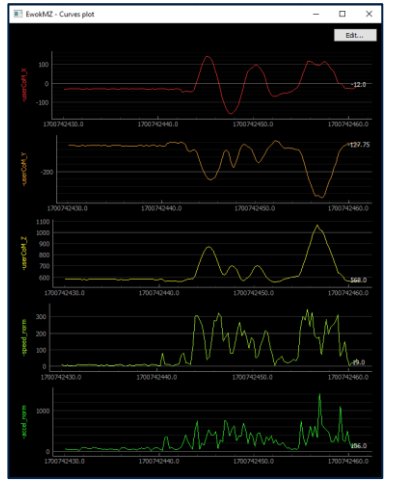


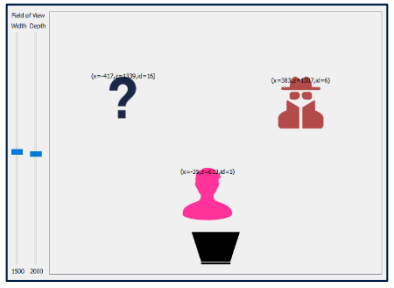


SPD settings
Live SPD parameters tuning



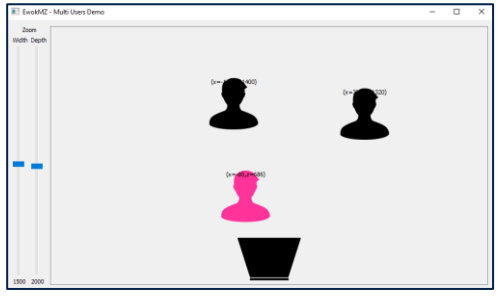
Curves plot
Complete tool to plot curves among all the data outputted by the device.





User Analysis
Allows to detect multiple objects and identify if they are intruders

Multi Users Demo
Allows to detect multiple humans and track their positions

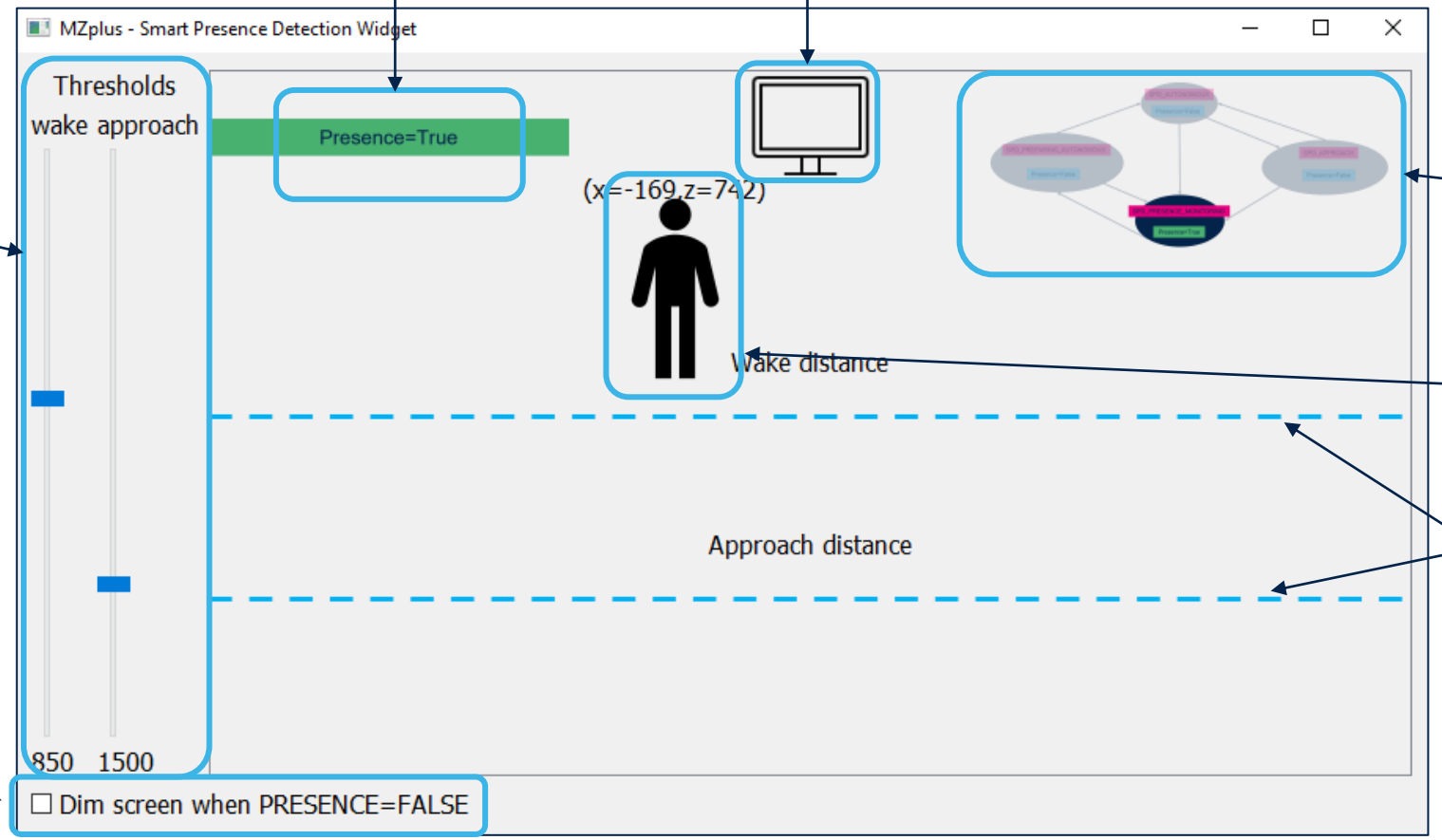


Smart Presence Detection Widget
the all-in-one widget

System mode
Power On
Power Off (no user detected)
Sleep (sensor in running autonomous)

Presence Flag
Green = True
Blue = False

Distance Threshold
The wake distance and approach distance can be tuned by the sliders



SPD state Machine
The current state is highlighted

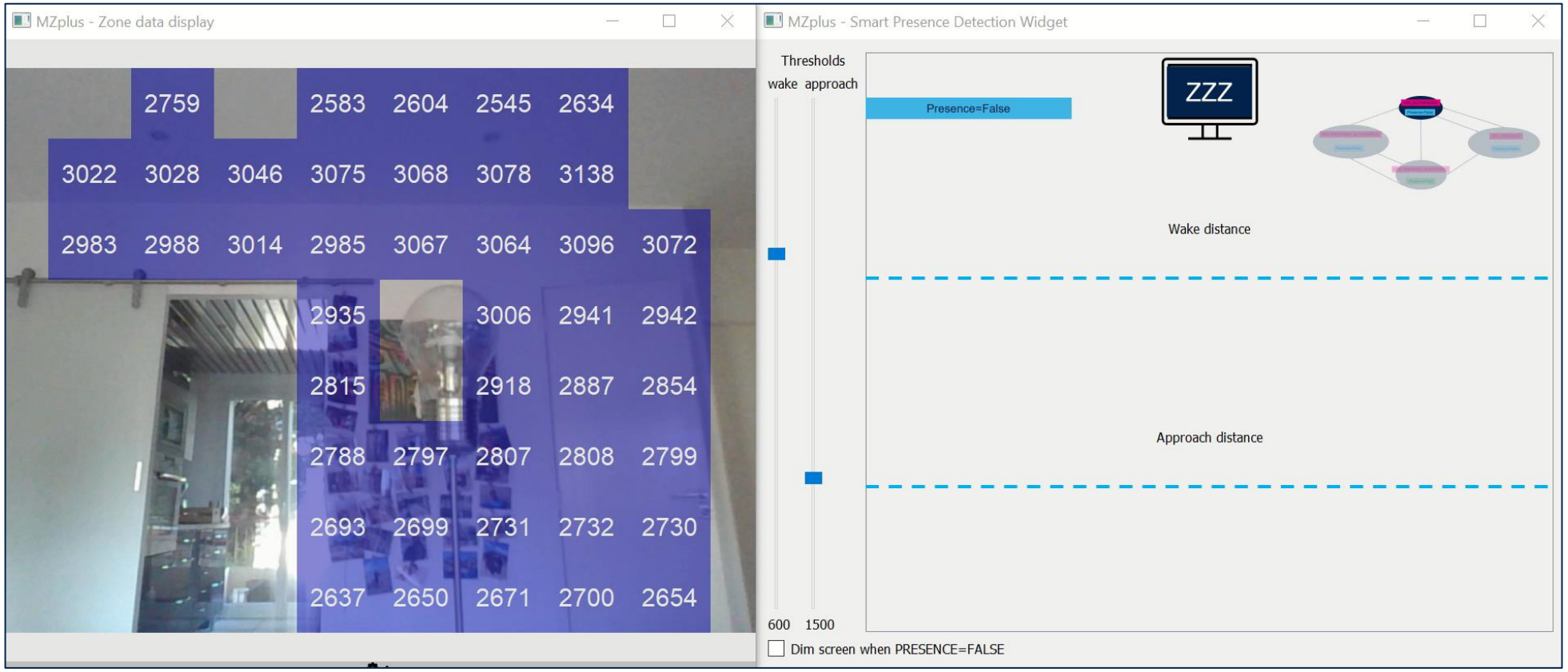
Main User tracking
Live main user position

Working SPD area
The distance lines are moving according the sliders

Screen Dimmer option
To dim the main screen when Presence flag is False

How to Showcase the SPD features

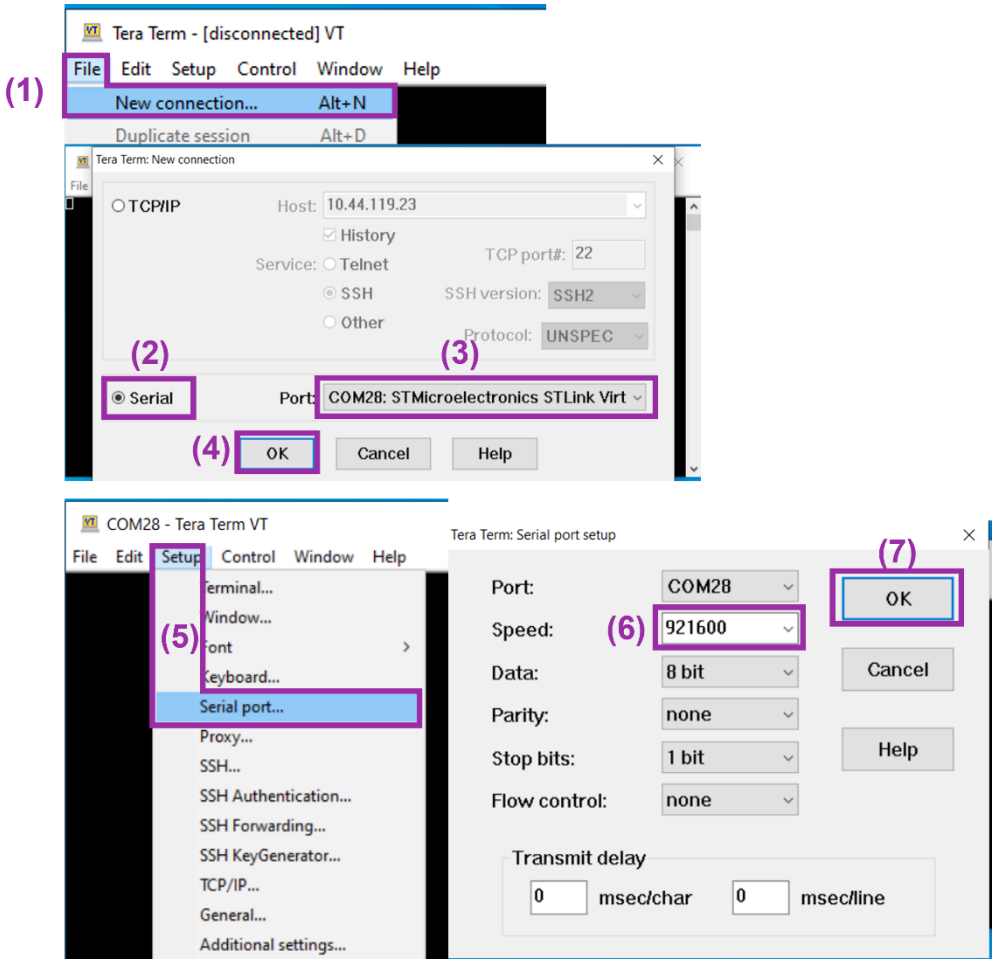
- Wake-on Approach
- Human VS Object
- Passing-By Filter
- Wake-on Stop
- Walk-away Lock



Running SPD in terminal

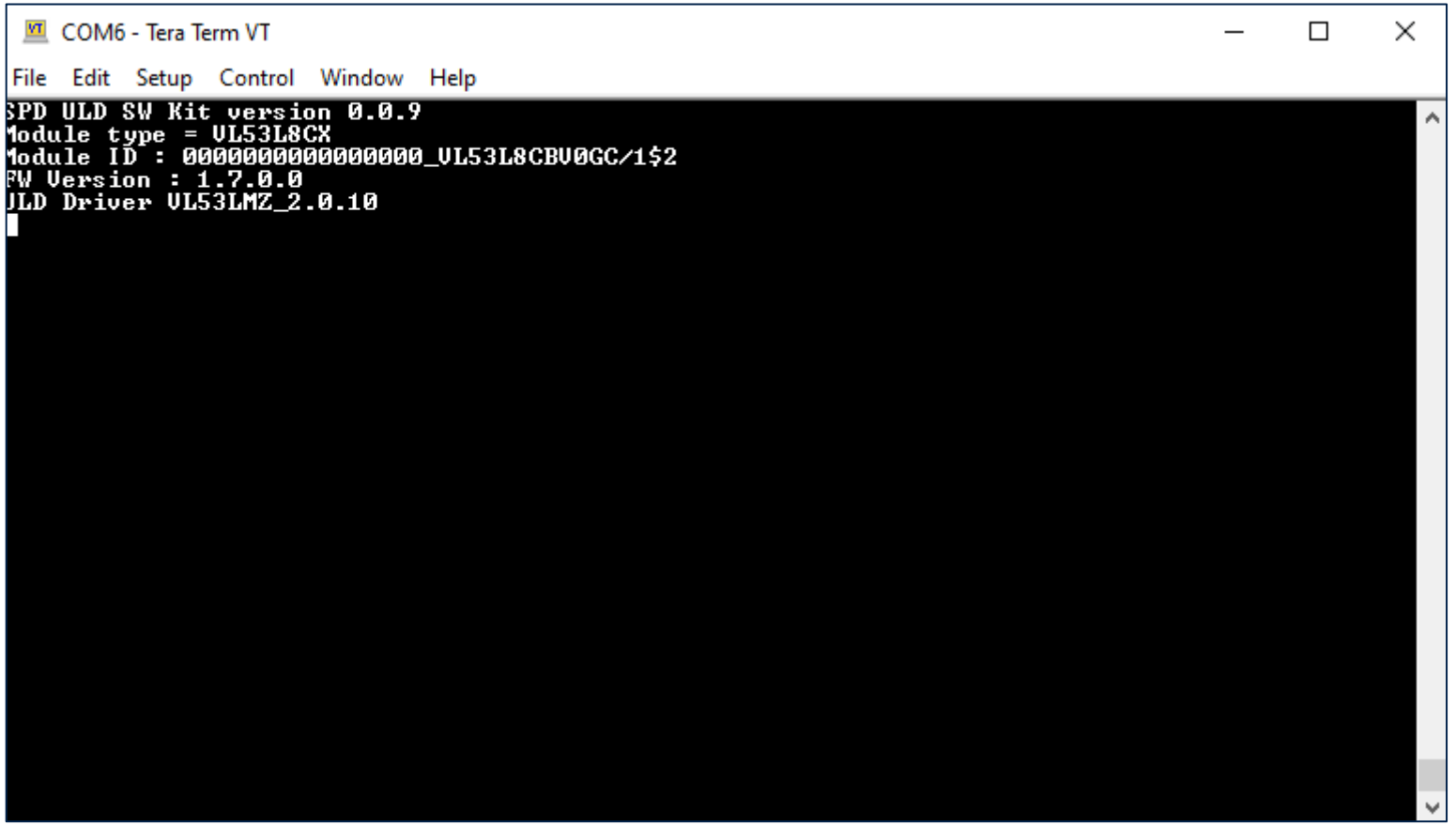
Running in a serial terminal (1/3)

- To connect the serial port please follow the steps shown in the figure below:



Running in a serial terminal (2/3)

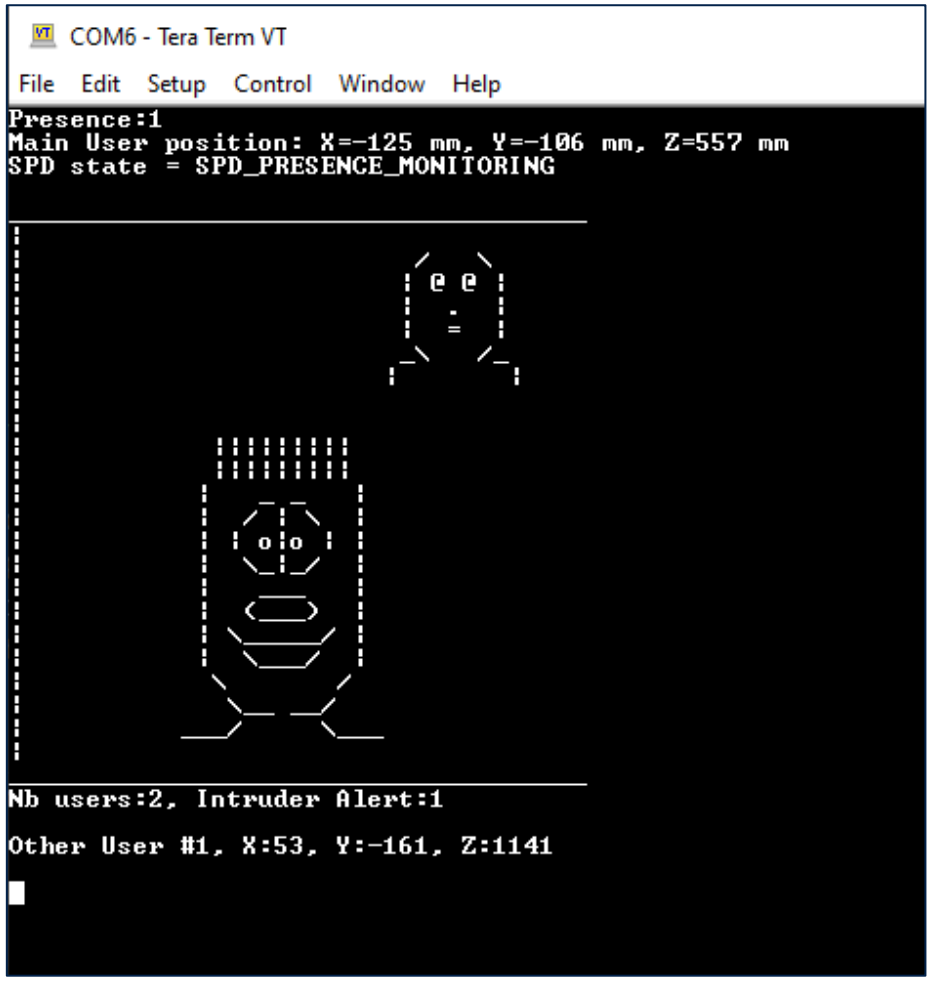
- To run the application, use the command line “enable” and press enter:



* For more information about the available commands, see the chapter: command manager

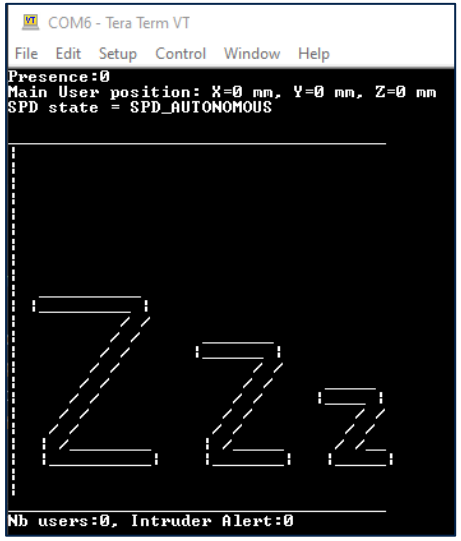
Running in a serial terminal (3/3)

- The serial terminal shows the output of the ST SPD solution:



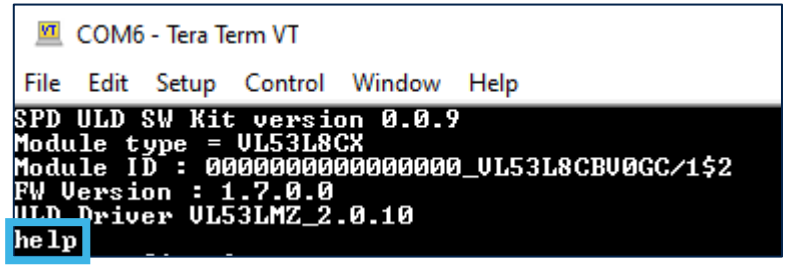
- SPD metrics:
  - Presence flag
  - Main User position
  - Number of Users

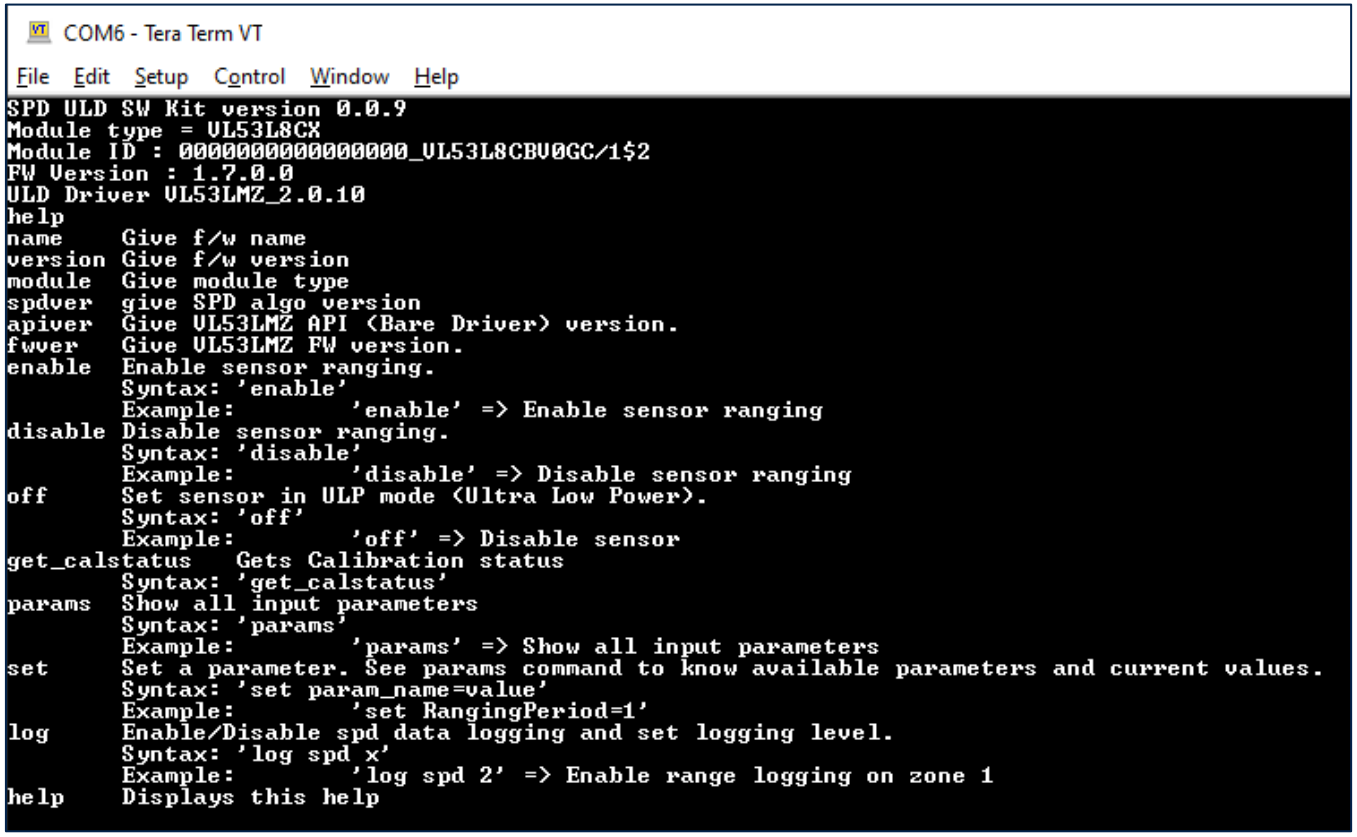
- State Machine
- Multi human detection viewer
- Users' information



Command manager

- All the commands are available in the help:





- The principal commands:
  - help
  - enable
  - disable
  - params
  - set

SPD in details

SPD key outputs

SPD outputs:

- **Presence Flag**
- Main User X/Y/Z positions
- Main User moving speed
- SPD state

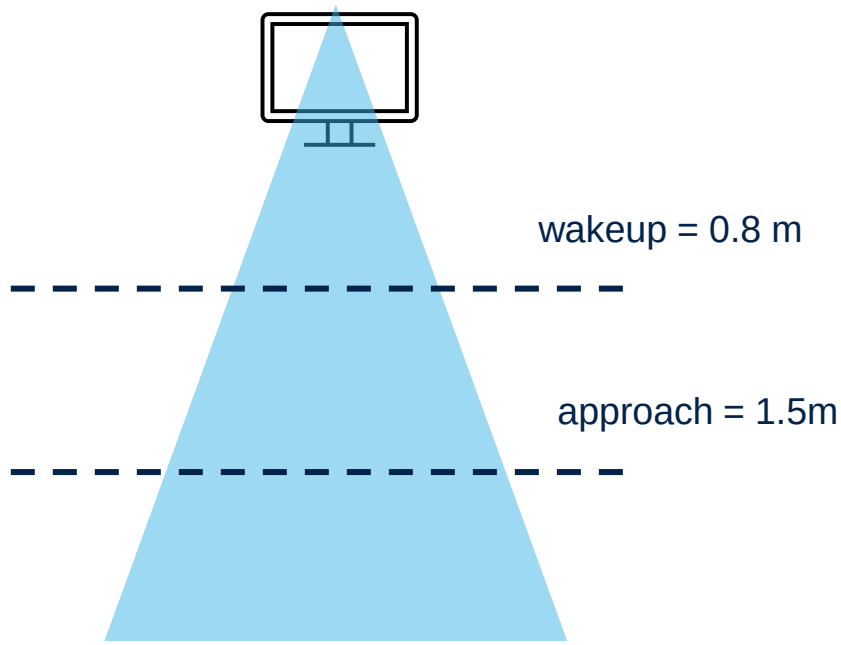
Multi-Human tracking outputs:

- **Number of Users**
- Each user X/Y/Z positions and speed

Key distance parameters

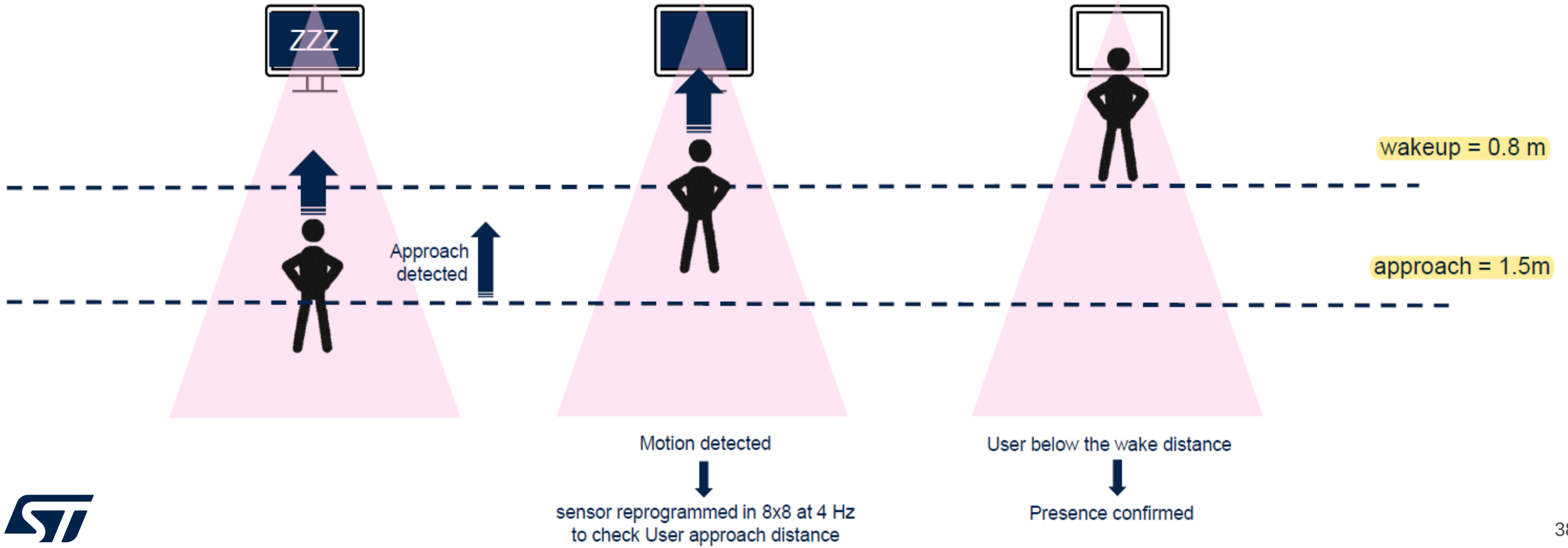
- WAKEUP:
  - the **Presence Flag** is always **True** when a User is detected inside this area

- APPROACH:
  - The SPD algorithm is tracking the identified User and try to determine if:
    - Passing-By: the user is just crossing the sensor FoV and doesn't have interest in the application
      - Presence flag = False
    - Wake-On Stop: the user stops his movement in front of the application and potentially he is interested by
      - Presence flag = True



Wake On Approach

FSM	SPD_AUTONOMOUS	SPD_APPROACH	SPD_PRESENCE_MONITORING
Sensor	Sleep, 4x4, 1 Hz	Ranging, 8x8, 4 Hz	Ranging, 8x8, 4 Hz
Presence	False	False	True



SPD ranging mode

Autonomous mode

Sensor is ranging autonomously and wake the host up only if a motion is detected

Example
Sensor configuration 1:
- **Ranging Frequency = 1 Hz**
- **Integration Time = 2 ms**

Example
Sensor configuration 2:
- **Ranging Frequency = 4 Hz**
- **Integration Time = 2 ms**

Default SPD settings

Low Power Consumption

Tracking mode

Sensor is ranging continuously and sends the ranging data to the host and the SPD algorithm

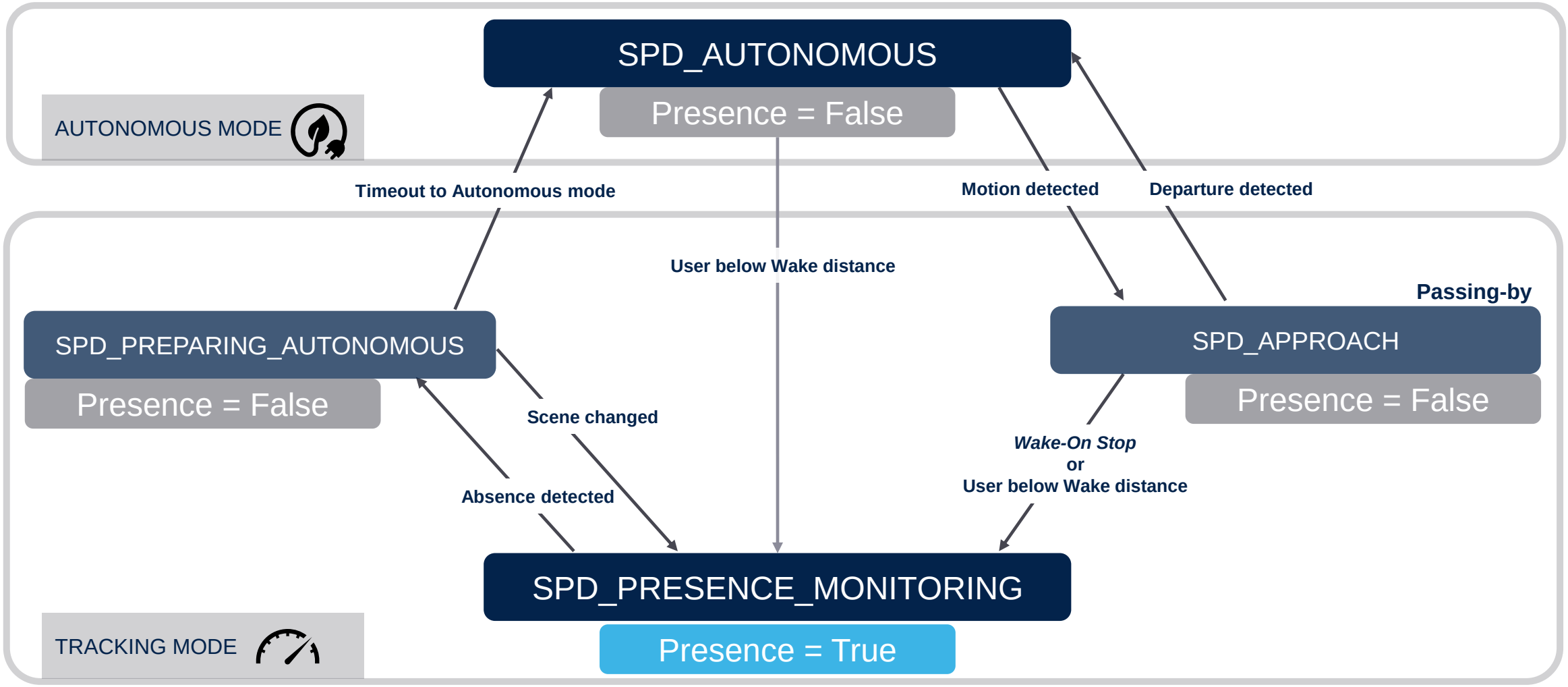
Example
Sensor configuration 1:
- **Ranging Frequency = 4 Hz**
- **Integration Time = 5 ms**

Example
Sensor configuration 2:
- **Ranging Frequency = 10 Hz**
- **Integration Time = 10 ms**

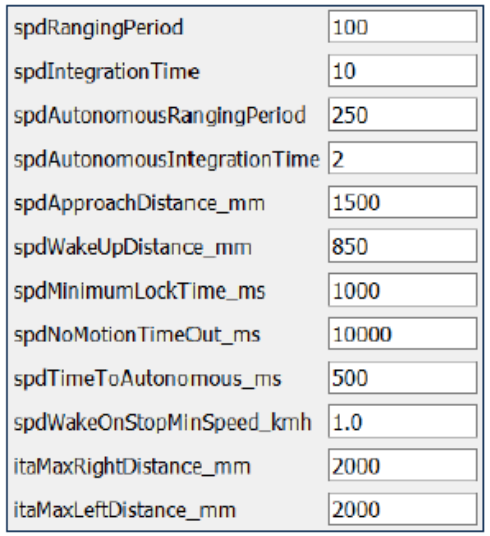
Default SPD settings

Full performances

SPD State Machine



SPD Params



SPD Settings widget

Parameters	Default value	Value range	Description
spdSensorRotation	0		Change the sensor orientation used in the application and adapt the data indexes
spdRangingPeriod (ms)	100	50 – 1000 (20Hz – 1Hz)(*)	Sensor ranging period when the sensor is in tracking mode (SPD_APPROACH / SPD_PRESENCE_MONITORING / SPD_PREPARING_AUTONOMOUS)
spdIntegrationTime (ms)	10	1 – 20	Sensor integration time when the sensor is in tracking mode (SPD_APPROACH / SPD_PRESENCE_MONITORING / SPD_PREPARING_AUTONOMOUS)
spdAutonomousRangingPeriod (ms)		(20Hz – 1Hz)	Sensor ranging period when the sensor is in autonomous mode (SPD_AUTONOMOUS)
spdAutonomousIntegrationTime (ms)	5	1 – 20	Sensor integration time when the sensor is in autonomous mode (SPD_AUTONOMOUS)
spdWakeUpDistance_mm (mm)	850	0 – 2000	"Wake Up distance" threshold -> Key distance parameters
spdApproachDistance_mm (mm)			"Approach distance" threshold -> Key distance parameters It has to be higher than the "Wake Up distance"
spdMinimumLockTime_ms (ms)	1000	0 – 2147483647 (**)	Timeout after departure detection: FSM: SPD_PRESENCE_MONITORING -> SPD_PREPARING_AUTONOMOUS
spdNoMotionTimeOut_ms (ms)			Timeout if no motion detected in the scene: FSM: SPD_PRESENCE_MONITORING -> SPD_PREPARING_AUTONOMOUS
spdTimeToAutonomous_ms (ms)			Time to switch to autonomous mode when the Presence Flag if False FSM: SPD_PREPARING_AUTONOMOUS -> SPD_AUTONOMOUS
spdWakeOnStopMinSpeed_kmh (km/h)	1	0.2 – 10 (***)	Threshold speed to detect the user stopped FSM: SPD_APPROACH -> SPD_PRESENCE_MONITORING

Sensor Settings

Distances

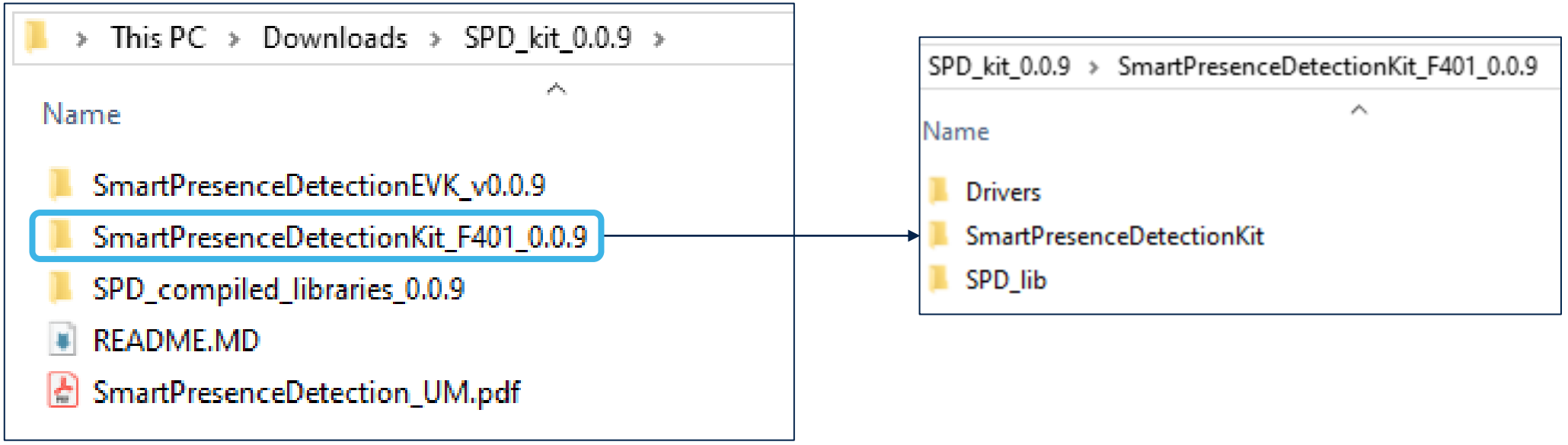
Timings

(*) The max ranging frequency @20Hz can only be reached with the maxI2C baudrate 1MHz
(**) 2147483647 is the max value supported by the algo variable, but has probably no application interest
(***) The algo can support more than the max value but 10Km/h is already more than the average walking human speed

SPD Programming Guide

life.augmented

Project architecture



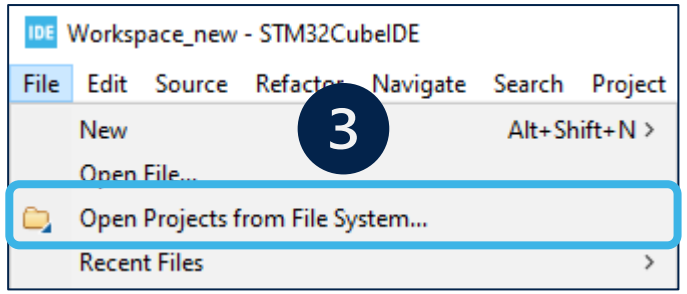
Drivers: STM32F401 libraries

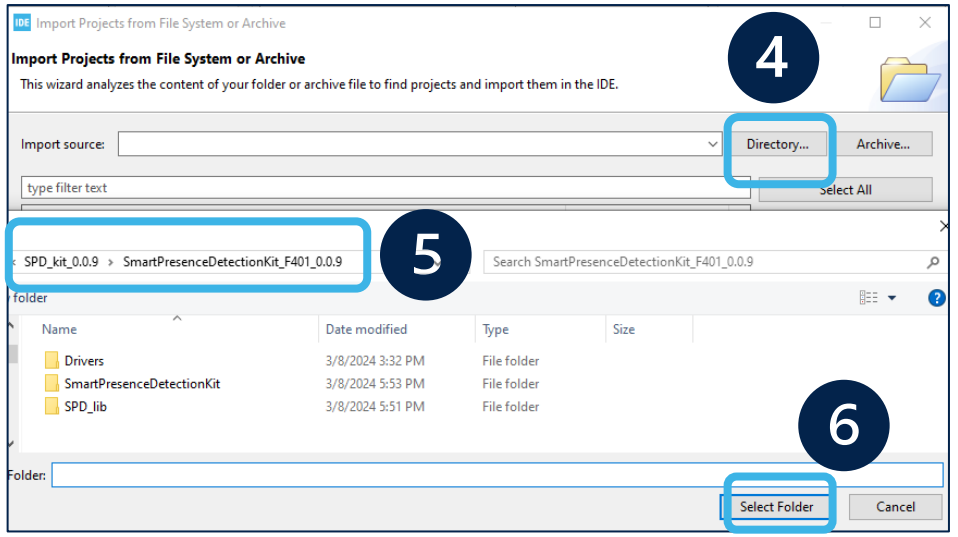
SmartPresenceDetectionKit: Application code

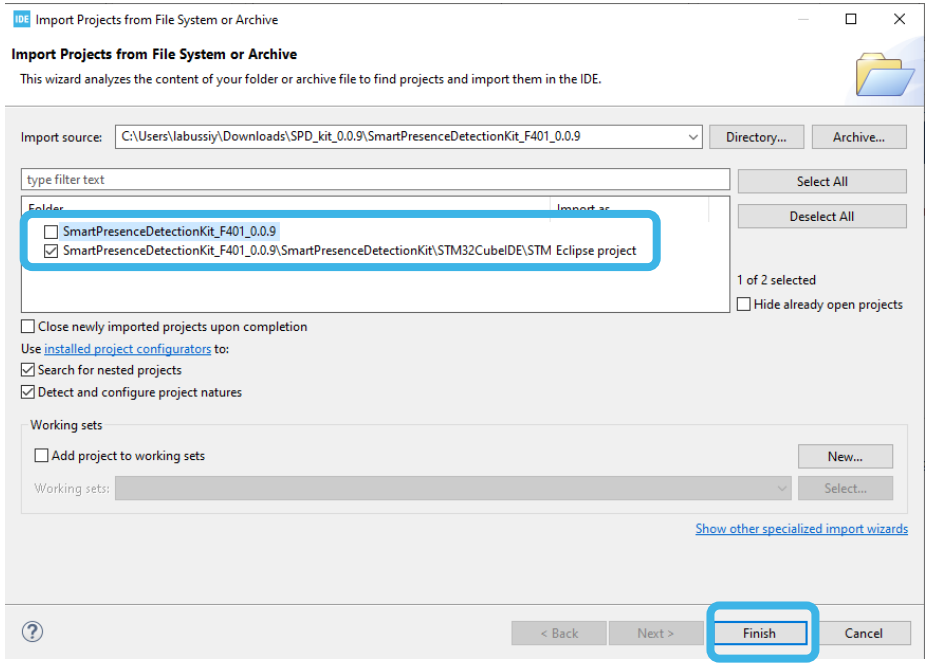
SPD_lib: SPD library + header files

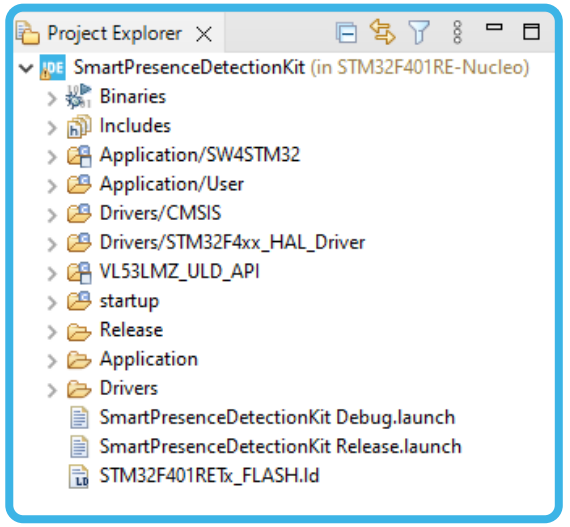
Open the project in CubeIDE

1. Open CubeIDE
2. New workspace










```
Build Finished. 0 errors, 3 warnings.
```

Implementation flow

Open main.c

1. Peripheral init
2. ToF sensor Init
3. SPD algo Init
4. Set Params in Tracking Mode
5. Configure the sensor
6. Start the Ranging & get frame

SPD Loop

7. SPD algo Run
8. Change Sensor Mode according SPD FSM

1
```
/* Reset of all peripherals, Initializes the
HAL_Init();

/* Configure the system clock */
SystemClock_Config();

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_DMA_Init();
MX_USART2_UART_Init();
MX_I2C1_Init();
UartComm_Start();
```

2
```
/* Initialize VL53LMZ I2C address */
LMZDev.platform.address = 0x52;

/* Initialize sensor */
status = init_vl53lmz_sensor();
```

3
```
/* Initialize SPD algo */
SPD_Init(1);
```

4
```
// Reset sensor settings in Tracking mode (after SPD init)
SPD_SetParamsForTracking();
sci_motion_map_stored = 0;
```

5
```
// Configure VL53LMZ
status = vl53lmz_Configure();
```

6
```
// Start ranging
status = vl53lmz_start_ranging(&LMZDev);
```

```
get_ranging_data()
```

7
```
/* Run SPD */
SPD_run(&SPD_Data, &SEN_MeasData, &SEN_Info);
```

8
```
/* Possibly change sensor mode */
SPD_ChangeSensorMode(&SPD_Data);
```

What is SPD_ChangeSensorMode doing?

- This function changes the sensor mode according the SPD FSM.
- There are 2 FSM transitions requiring to change the sensor mode

- Going to SPD_AUTONOMOUS
  - **SEN_SetParamsForAutonomous()**
    - Resolution = 4x4 (mandatory)
    - spdAutonomousRangingPeriod
    - spdAutonomousIntegrationTime
    - disablePipe
    - sciDetectionThreshold
    - auto_stop enabled

- Leaving SPD_AUTONOMOUS
  - **SEN_SetParamsForTracking()**
    - Resolution = spdResolution
    - spdRangingPeriod
    - spdIntegrationTime
    - enablePipe
    - sciDetectionThreshold
    - auto_stop disabled

Algo Data structures

* only useful variables are described

- Data structure:
  - **SPD_Data_t** (spd.h)
    - state : SPD FSM current state
    - presence : User Presence detection flag
    - userID : User # of the main user
    - TRK_Data.tkd_Obj[SPD_Data.userID].CoM_pos.x/y/z : Main User x/y/z position
    - TRK_Data.tkd_Obj[SPD_Data.userID].speed_norm : Main User x/y/z motion speed
    - Nb users
    - positions

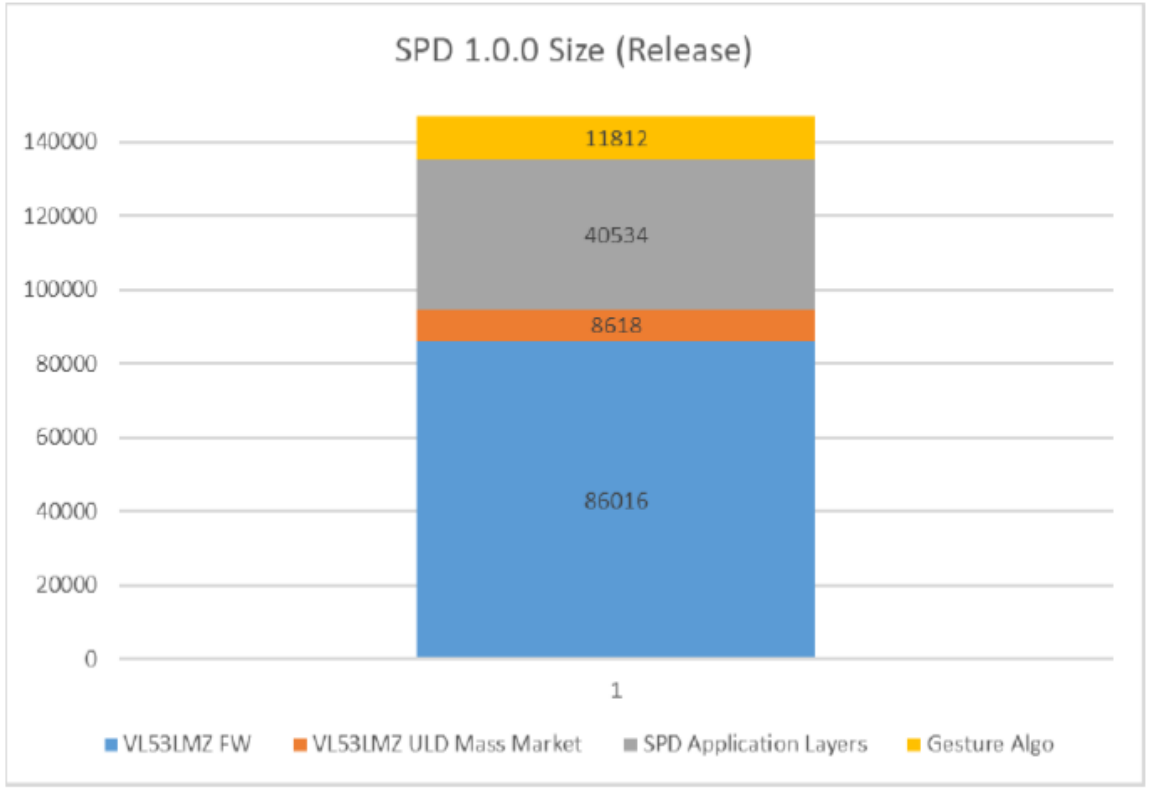
- Measurement structure:
  - **SEN_Measurement_Data_t** (sensor.h)
    - Data sensor formatted for the SPD algo
    - timestamp_ms / RangeMilliMeter / SignalRatePerSpad / …

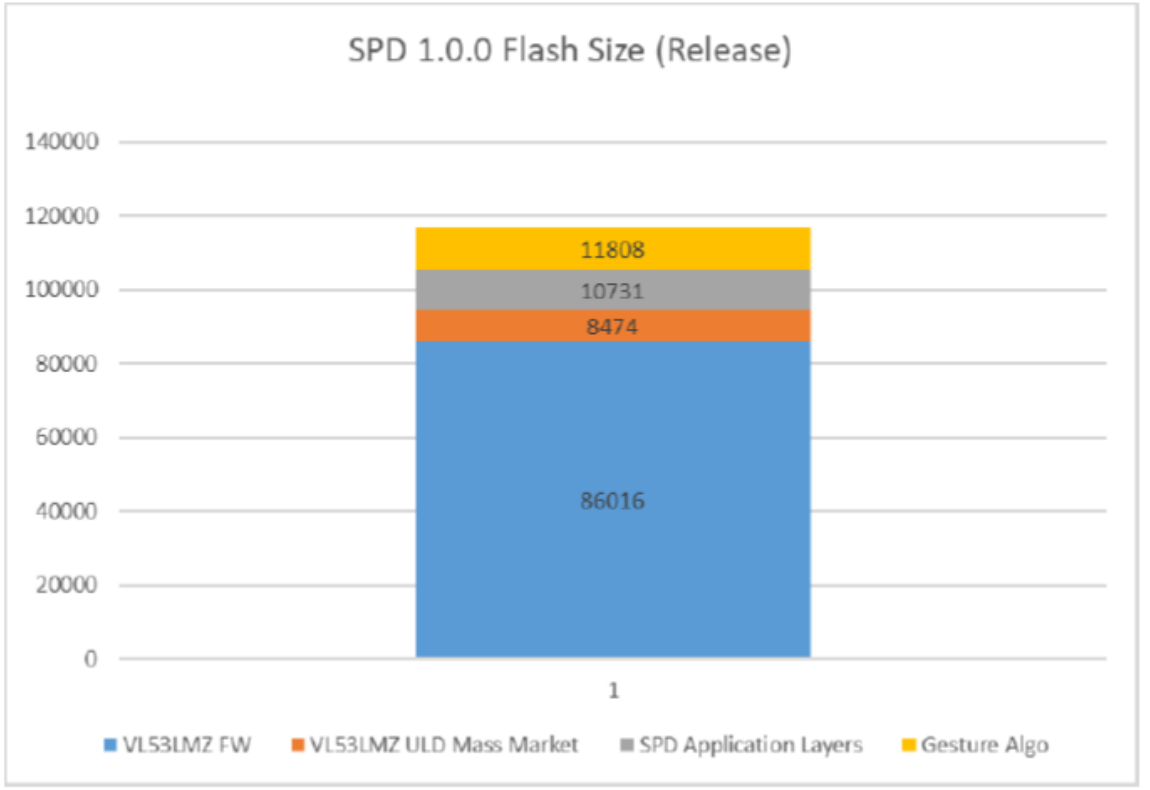
- Sensor info structure
  - **SEN_Info_t** (sensor.h)
    - General information about the sensor settings to tune the SPD algo
    - nb_zones : Sensor resolution
    - Freq : Sensor ranging frequency

- Input parameters
  - **Params_t** (sensor_command.h)
    - Main parameters described in SPD Params

Memory Footprint

SmartPresenceDetectionKit_1.0.0 (spd v1.0.0)			
	FLASH	RAM	memory
VL53LMZ FW	86016	0	86016
VL53LMZ ULD Mass Market	8474	144	8618
SPD Application Layers	10731	29803	40534
Gesture Algo	11808	4	11812
TOTAL	**117029**	**29951**	**146980**





SPD algo profiling

Algo Rev	I2C @ 400KHz get_ranging_data() + SEN_CopyMeasurementData()		I2C @ 1MHz get_ranging_data() + SEN_CopyMeasurementData()		SPD Init execution time SPD_init()		Maximum SPD execution time SPD_run()	
	CPU cycles	ms	CPU cycles	ms	CPU cycles	ms	CPU cycles	ms
V1.0.0	*2676450*	*3.2*	*118890*	*1.4*	*131398*	*1.5*	*470000*	*5.6*

Performance on STM32F401 @ 84MHz

Our technology starts with You

Find out more at www.st.com

life.augmented