# Library People Detection

For MLX90642 infrared sensor array

Datasheet

## General description

### Features & benefits

- Algorithm-based real-time monitoring
- Lightweight and optimized for low CPU and memory usage
- Privacy-friendly: no visual image
- Hardware agnostic
- No training needed on site
- Works in complete darkness
- Reliable even when people aren't moving

### Available metrics

- Humans detected / not detected
- Number of people in the room
- Coordinates and location history

### Applications examples

- Smart building management
- Elevator safety
- Building security
- Smart home

### Available support & tools

- The latest firmware for the Evaluation Kit can be found on the evaluation kit webpage: melexis.com/DVK90642
- For any information about the library please visit: melexis.com/library-people-detection
- Compiled library can be downloaded through the MyMelexis platform: www.melexis.com/mymelexis
- To gain access, please contact your local sales representative.
- If you have any questions, please contact Melexis via www.melexis.com/technical-inquiry

### Description

This software library is designed to detect room occupancy using data from the MLX90642 infrared sensor array. The MLX90642 is a compact thermal sensor that captures heat patterns, making it suitable for occupancy detection even in complete darkness.

The library uses efficient image analysis algorithms to process the thermal data and reliably determine how many people are present in a given space. It is robust and adaptable, supporting a wide range of confined indoor environments.

Easy to integrate and test, the library is well-suited for both early-stage development and deployment in demanding production environments. It offers a scalable and practical solution for adding intelligent occupancy detection to your products and systems.

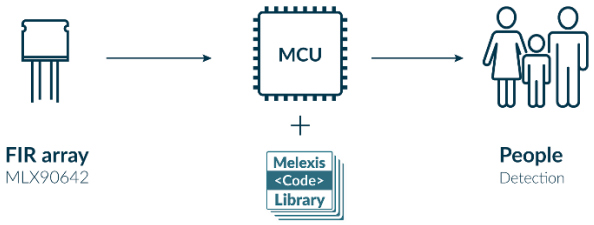


*Figure 1. Library People Detection*

Melexis
INNOVATION WITH HEART

## Contents

# 1 Architecture and Design

## 1.1 High-level architecture diagram

The people-detection library for MLX90642 follows a modular, layered architecture. Although multiple algorithm variants are provided (Simplified and Full), all variants share the same overarching processing pipeline.

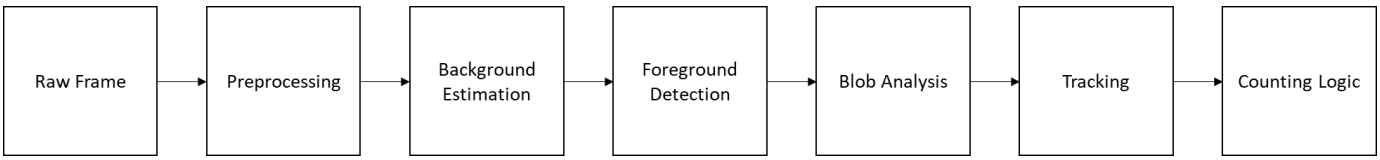


*Figure 2. High-level architecture diagram*

The library is composed of several high-level functional modules:

- **Preprocessing:** Basic conditioning of the incoming MLX90642 data.
- **Background Modeling**: Establishes a reference thermal background for comparison.
- **Foreground Extraction**: Identifies regions that differ significantly from the estimated background.
- **Blob Processing:** Groups foreground pixels into regions of interest ("blobs") and extracts simple shape information.
- **Tracking:** Maintains blob continuity across frames for stable person detection.
- **Counting Logic:** Converts detected regions into a robust estimate of the number of people.

All algorithm variants implement this structure, but differ in:

- complexity of background estimation
- blob segmentation parameters
- tracking heuristics
- RAM and CPU usage

## 1.2 Control flow chart

A typical integration follows this generic sequence:

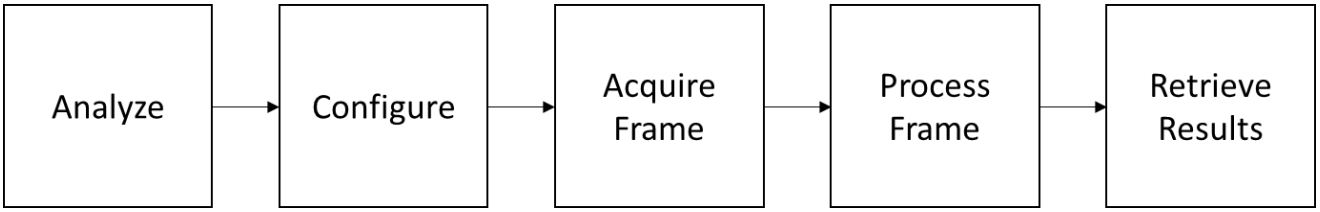


*Figure 3. Control flow chart*

This flow applies to all supported platforms and algorithm modes.

## 1.3 Dependencies

**Internal Dependencies**

- Core processing modules listed above
- Built-in MLX90642 sensor driver (initialization, configuration, frame acquisition)
- Fixed-point math utilities (no FPU needed)
- Internal buffers sized based on:
    - algorithm mod
    - output format
    - *max_blobs* configuration

**External Dependencies**

- No external MLX90642 driver required
- No additional libraries required at build time (precompiled binary distribution)
- Application may need to provide:
    - platform-specific I2C low-level functions
    - periodic frame polling or interrupt triggering based on system design

**Hardware Dependencies**

- MLX90642 sensor connected via I2C
- MCU capable of supporting required RAM/Flash:
    - Simplified mode: ~8 KB RAM, ~90 KB Flash
    - Full mode: up to 40 KB RAM, ~90 KB Flash
- Execution time depends on MCU performance

# 2 Integration Guide

This section describes the steps required to integrate and install the people-detection library on supported hardware and development environments. The library is provided as precompiled binaries and requires only basic configuration and I2C integration to operate.

## 2.1 Supported hardware

The library is designed to work exclusively with the Melexis MLX90642 thermal infrared sensor.

The library is fully C99-compliant, uses fixed-point arithmetic (no floating-point hardware required), and has no external dependencies. As a result, it can be compiled for virtually any 32-bit or higher embedded or desktop architecture, including:

- **ARM Cortex-M** — from ultra-low-power M0/M0+ to high-performance M4, M7, M33, M55, and M85
- **ARM Cortex-A** — suitable for embedded Linux/Android platforms (e.g., Raspberry Pi, NVIDIA Jetson, NXP i.MX) using standard aarch64-linux-gnu toolchains
- **RISC-V** — compatible with RV32IM and RV64GC-class microcontrollers and SoCs, including ESP32-C3/C6
- **Xtensa** — compatible with ESP32 and ESP32-S3 families
- **x86 / x86_64** — for desktop development, testing, and simulation on Windows, Linux, and macOS

Standard distribution package:

1. **ARM Cortex-M:** Precompiled static libraries for M0, M3, and M4 (M4 build is forward-compatible with M7/M33/M85)
2. **ARM Cortex-A:** Universal static library targeting 64-bit Linux (aarch64)
3. **x86_64 Linux:** Static library for desktop simulation and integration testing

Builds for additional architectures are available upon request.

## 2.2 Hardware requirements:

- Functional I2C interface connected to the MLX90642
- Sufficient Flash memory to store the library binary
  - Typical requirement: up to 90 KB FLASH
- Sufficient RAM, depending on the selected algorithm mode:
  - Full mode: 40 KB RAM
  - Simple mode: 14 KB RAM
- Minimum clock frequency:
  - Mode: 8 fps
    - Full: ~82 MHz
    - Simple: ~3 MHz
  - Mode: 16 fps
    - Full: ~164 MHz

- Simple: ~5 MHz.

Those are theoretical minimums at 100% CPU. In practice, we may recommend "At least 16 MHz for Simple mode, 100 MHz for Default mode @ 8 fps, or 200 MHz for Default mode @ 16 fps. This provides headroom for application code alongside the library.

- Reference benchmark (Cortex-M4 @ 180 MHz):
  - Full: ~57 ms per frame
  - Simple: ~1 ms per frame
- Stable power supply and recommended MLX90642 decoupling/filtering practices
- Standard integer-based error handling via mlx9064x_isp_result_t

Output format options:

- Configurable output detail level:
  - MINIMAL: Person count only
  - EXTENDED: Blob masks with associated metadata
  - Intermediate levels available in between

## 2.3 Supported environments

The library is platform-agnostic and compatible with a wide range of embedded development environments.

Supported operating environments:

- Bare-metal (no OS) — recommended for lowest latency
- RTOS-based systems
- Other embedded OSes

Supported compilers:

- GCC
- ARM Keil / ArmClang
- IAR Embedded Workbench
- RISC-V GCC toolchains

Language standard:

- Compiled for C99
- Compatible with C/C++ integration environments

# 3 Configuration & Initialization

The library requires configuration information to be provided before initialization.

All parameters are stored in the mlx9064x_isp_people_detection_config_t struct and allow the user to tune detection sensitivity, temperature assumptions, and blob-tracking behavior for MLX90642-based people detection.

During initialization, the user must:

1. Fill the configuration struct (either with defaults or custom values).
2. Pass the struct to the library initialization function.

## 3.1 Required configuration parameters

Below is a summary of all parameters that must be configured (or left at default).
Defaults ensure stable operation for most indoor environments.

**Algorithm mode** — *algo_mode*

- Modes: FULL (0) or SIMPLIFIED (1)
- Default: **1 (Simplified)**
  Defines the complexity of blob tracking. Use FULL for multi-person tracking and more stable trajectories; SIMPLIFIED for faster, low-power operation.

**Maximum blobs** — *max_blobs*

- Range: 1–6, Default: **4**
  Defines the internal memory allocation for blob tracking. Increase for higher occupancy scenes.

**Standard footprint** — *area_std_footprint*

- Range: 35–55 px, Default: **45**
  Used to convert blob area into estimated person count.

**Minimum expected temperature** — *temp_range_min*

- Format: °C × 50 (1000 = 20.0°C)
- Range: 0 or 500–1500
- Default: **0 → auto (20°C)**
  Sets the minimum temperature used for normalization.

**Maximum expected temperature** — *temp_range_max*

- Format: °C × 50 (2000 = 40.0°C)
- Range: 0 or 1500–2500
- Default: **0 → auto (40°C)**
  Sets the maximum temperature used for normalization.

**Background threshold (lower)** — *fact_std_raw_l*

- Fixed-point Q8.8 multiplier
- Range: 128–1024 (0.5× to 4.0×)

- Default: **256 (1.0×)**
  Defines how aggressively the background is suppressed below the peak.

**Background threshold (upper) —** *fact_std_raw_h*

- Fixed-point Q8.8 multiplier
- Range: 128–1024 (0.5× to 4.0×)
- Default: **256 (1.0×)**
  Defines how aggressively foreground is separated from the background above the peak.

**Minimum blob area —** *min_blob_area*

- Range: 5–100 px
- Default: **10 px**
  Filters out blobs too small to be considered valid detections.

## 3.2 Functional Safety

This library does not implement a dedicated Functional Safety (FuSa) concept. However, the following guidelines can help system integrators build safe and reliable applications.

Key notes:

- The library performs self-diagnostics on the data coming from the MLX90642 sensor.
- Temperature frames data is checked after user I2C readout.
- Configuration parameters are validated at startup.
- No ASIL, SIL, or safety certification is claimed.

Recommended integrator safety measures:

- Reset the algorithm state on sudden frame corruption events.
- Implement watchdog supervision.

# 4 Dependencies and Compatibility

The library is hardware-agnostic and runs on any microcontroller platform that provides a configurable I2C interface and supports C99-compatible toolchains.
Supported environments:

- 32-bit and 64-bit microcontroller platforms (e.g., ARM Cortex-M, RISC-V, Xtensa)
- Bare-metal or RTOS-based systems
- Toolchains compliant with C99 or newer

Hardware interface requirements:

- Functional I2C bus connection to the MLX90642

- May require low level user-provided I2C functions for bus access

### 4.1 Runtime dependencies

The library has minimal runtime requirements. To operate correctly, the following conditions must be met:

- Availability of an I2C on the host microcontroller
- User-provided implementation of the MLX90642 driver (or equivalent I2C functions for sensor communication)
- Sufficient memory to store internal processing buffers

No external third-party libraries are required at runtime.

### 4.2 Changes across versions

This is the initial release (**v0.5.0**) of the library.

- First stable API
- Hardware-agnostic architecture
- Pre-compiled binaries for multiple MCU targets
- MLX90642 driver implemented
- Full documentation and usage examples included

## 5 Error codes

The library uses a standard integer-based error enumeration (*mlx9064x_isp_result_t*) to report function outcomes. All API functions return one of the defined status codes, allowing the application to detect operational errors, interface issues, or invalid inputs.

Return value convention

- 0 — Operation completed successfully
- Non-zero — An error condition occurred (see error categories below)

Error code categories:

The *mlx9064x_isp_result_t* enum includes (but is not limited to) the following groups:

1. I2C Communication Errors
   - Bus not initialized or unavailable
   - NACK received from MLX90642
   - Read/write timeouts
   - User-provided I2C function returned an error

2. Input Validation Errors
   - Null or uninitialized data structures

- Insufficient buffer size for selected output format
- Unsupported configuration or parameter out of range

3. Algorithm / Processing Errors

- Not enough RAM available for the selected algorithm mode
    - Full mode requires 40 KB RAM
    - Simple mode requiers ~14 KB RAM
- Internal processing constraints exceeded

4. Sensor Data Errors

- Invalid EEPROM data
- Inconsistent or corrupted frame data
- Temperature calculation domain errors

**Error reporting behavior**

- All API calls return a value from *mlx9064x_isp_result_t*.
- In case of fatal errors (e.g., corrupted frame data), the library ensures no partial or invalid output is generated.
- Detailed per-function error descriptions are provided in the API reference.

Notes for integration:
Output formats range from MINIMAL (person count only) to FULL (blob masks + metadata).
Ensure that the application allocates adequate buffers based on the selected output level.
The error codes are stable across versions unless explicitly documented in future release notes.

# 6 Application

## 6.1 Application description

The Melexis People Detection Library enables real-time detection and estimation of human presence using raw thermal data from the MLX90642 infrared array sensor.
It is designed for indoor environments where privacy-preserving, non-visual sensing is required.

The library runs entirely on an external microcontroller, using pre-compiled algorithm modules that process the sensor's thermal frame data. Depending on the selected algorithm variant (Simplified or Full), the library performs:

- Basic preprocessing of thermal frames
- Background and foreground separation
- Blob segmentation and filtering
- Multi-frame tracking of thermal objects
- People counting using configurable footprint models
- Optional metadata extraction (centroid, area, blob mask)

The solution is optimized for embedded systems that require:

- No camera integration
- Privacy-by-design sensing
- Deterministic processing time
- No external memory or FPU

Typical use cases include:

- Occupancy sensing in rooms, meeting spaces, offices
- Smart building automation
- People flow monitoring
- Energy-efficient lighting control
- Asset/space utilization analytics

The library does not require image reconstruction and operates purely on low-resolution thermal patterns, making it robust to lighting conditions, shadows, or visual obstacles.

## 6.2 High-Level application implementation diagram

Below is the conceptual data flow for integrating MLX90642 with the People Detection Library:

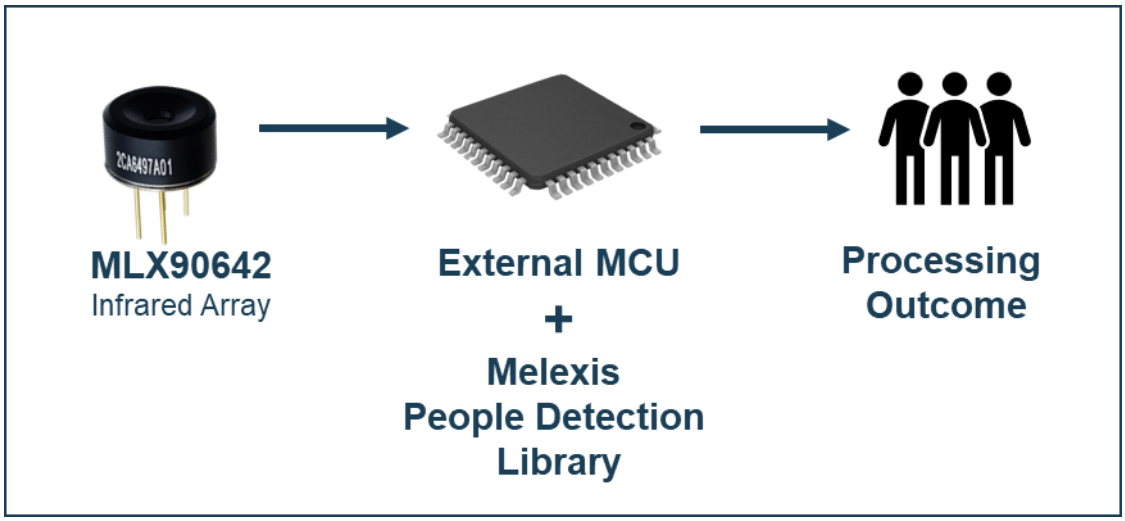


*Figure 4. High-level application implementation diagram*

# 7 Testing

## 7.1 Hardware testing

Hardware testing verifies basic integration between the MLX90642 sensor, the target MCU, and the people-detection library. The goal is to confirm correct data acquisition, stable I²C communication, and functional end-to-end processing.

### 7.1.1 How to run tests

1. Connect MLX90642 to the MCU using a functional I²C interface.
2. Integrate the pre-compiled people detection library into the MCU firmware.
3. Configure the MLX90642 for continuous frame acquisition using the preferred refresh rate.
4. Capture incoming frames and pass them to the library's frame-processing function.
5. Verify that:
   - The library initializes successfully.
   - Frames are processed without error codes.
   - Output values are provided consistently.
6. (Optional) Log raw frames and results for offline analysis.

This procedure validates correct hardware–firmware integration before performance testing.

## 7.2 Performance testing

Performance testing evaluates the behavior of the detection algorithm under real indoor conditions.

### 7.2.1 Test setup

To evaluate people-detection accuracy, tests should be performed in a controlled indoor environment.

1. Use a MLX90642 mounted on a suitable MCU platform or the People Detection Evaluation Kit (melexis.com/DVK90642).
2. Integrate the provided people-detection library into the firmware and enable continuous processing.
3. Mount the unit on the ceiling of an indoor, enclosed environment at a height between 2 m and 5 m. The recommended height for optimal performance is 2.5 m.

Ensure that subjects' heads remain within the sensor's Field of View (FOV), defined as 110° (H) × 75° (V).

For highest accuracy, install the unit at 90° (downward-facing). Operation is also supported at mounting angles between 90° and 45°. The rear side of the evaluation kit's case provides 90° and 45° reference surfaces for alignment.

This setup allows representative measurement of detection stability, tracking continuity, and counting accuracy.

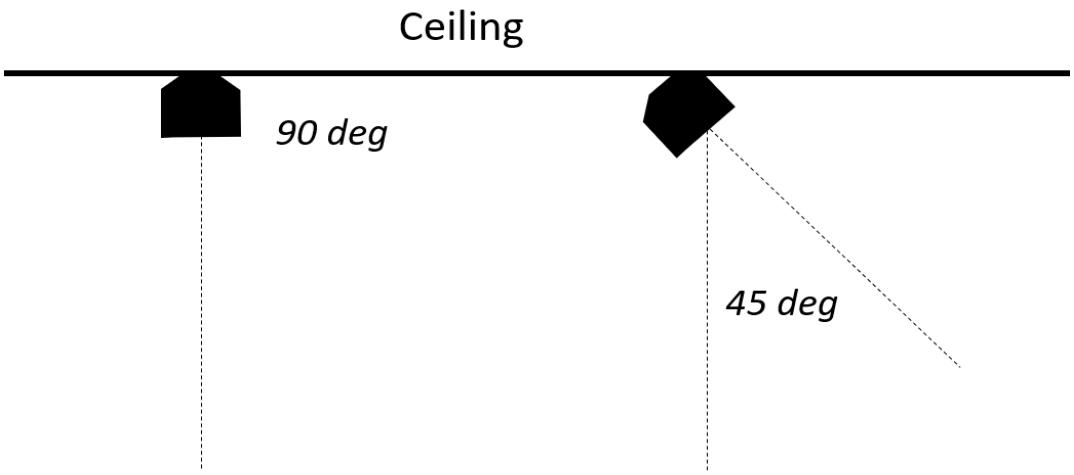


*Figure 5. Device mounting angles*

### 7.2.2 Performance comparison

Performance can be evaluated by comparing:

- Algorithm modes (Simplified vs Full)
- Different mounting heights
- Various occupancy scenarios (single person, multi-person clusters, standing, walking)
- Environmental variations (background temperature changes, airflow, HVAC operation)

Metrics commonly assessed include:

- Detection reliability
- Tracking continuity across frames
- Counting accuracy in dynamic scenes
- Sensitivity to mounting angle or distance
- Robustness against noise or background changes

Results may vary depending on MCU performance, room layout, and installation conditions.

### 7.2.3 Example images

This section provides visual examples illustrating typical operating conditions of the system.

Each example shows an RGB image of the scene together with the corresponding infrared frame captured from the same viewpoint. These examples demonstrate how the system interprets the thermal data and identifies the number of subjects present.

Example 1— One subject in an elevator (90° mounting)

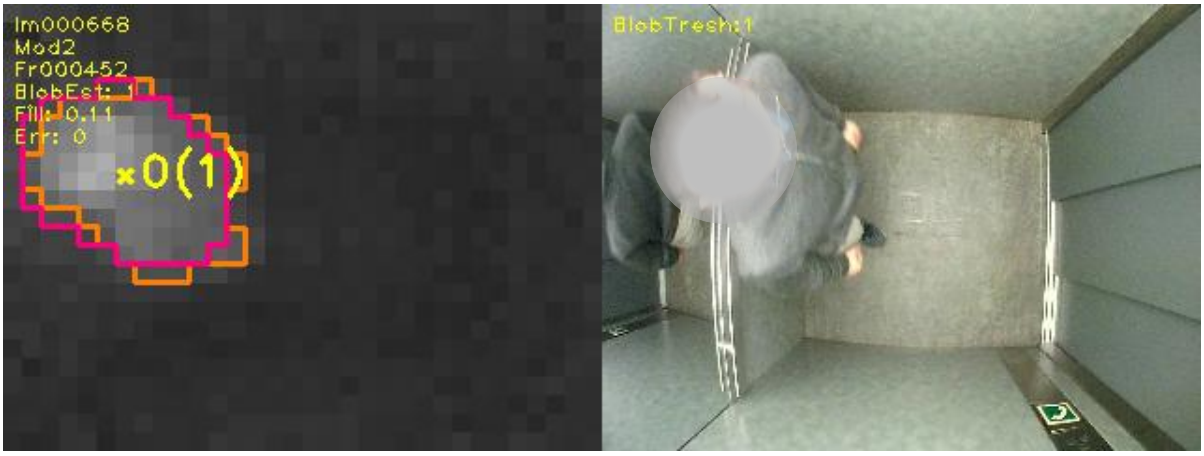


*Figure 6. One subject in an elevator (90° mounting)*

Example 2— Four subjects in an elevator (90° mounting)

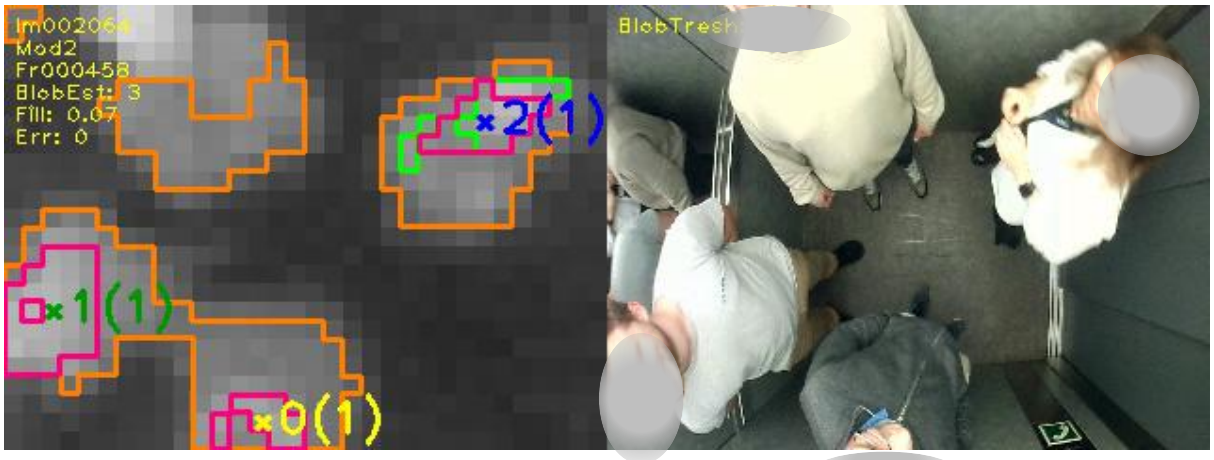


*Figure 7. Four subjects in an elevator (90° mounting)*

Example 3 — Two subjects in an elevator (45° mounting)

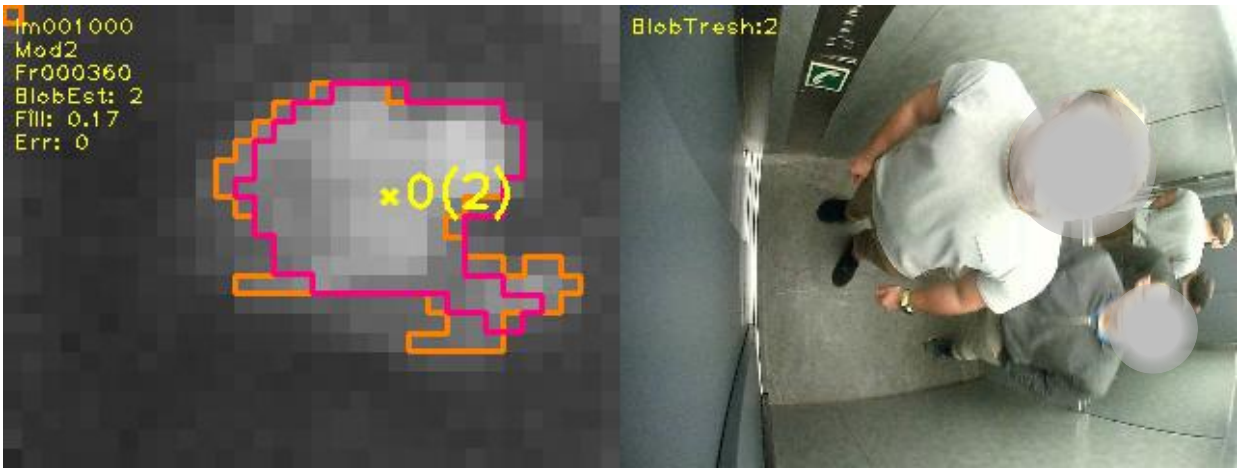


*Figure 8. Two subjects in an elevator (45° mounting)*

Example 4 — Three subjects in an elevator (45° mounting)

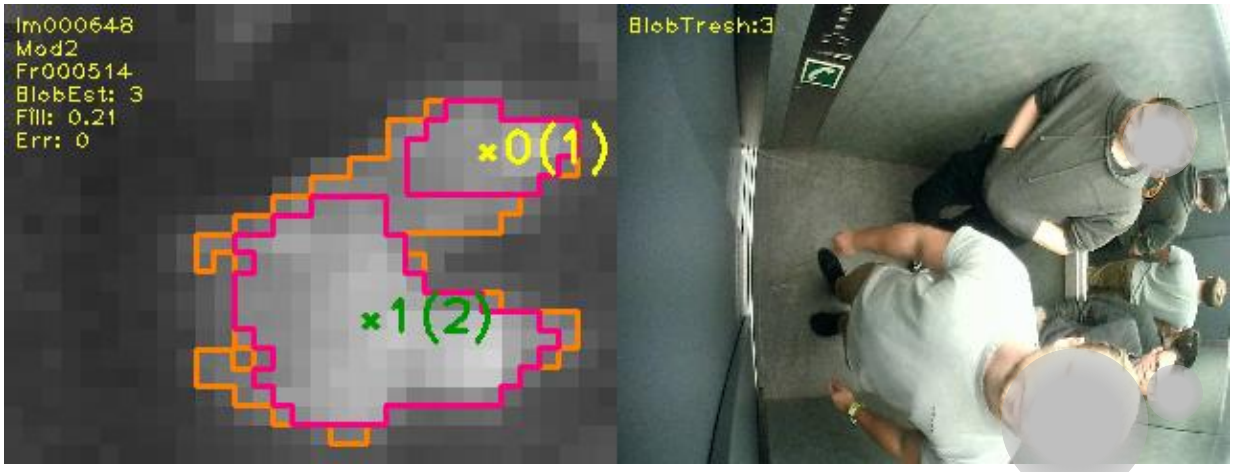


*Figure 9. Three subjects in an elevator (45° mounting)*

Example 5 — Two subjects in a corridor (45° mounting)

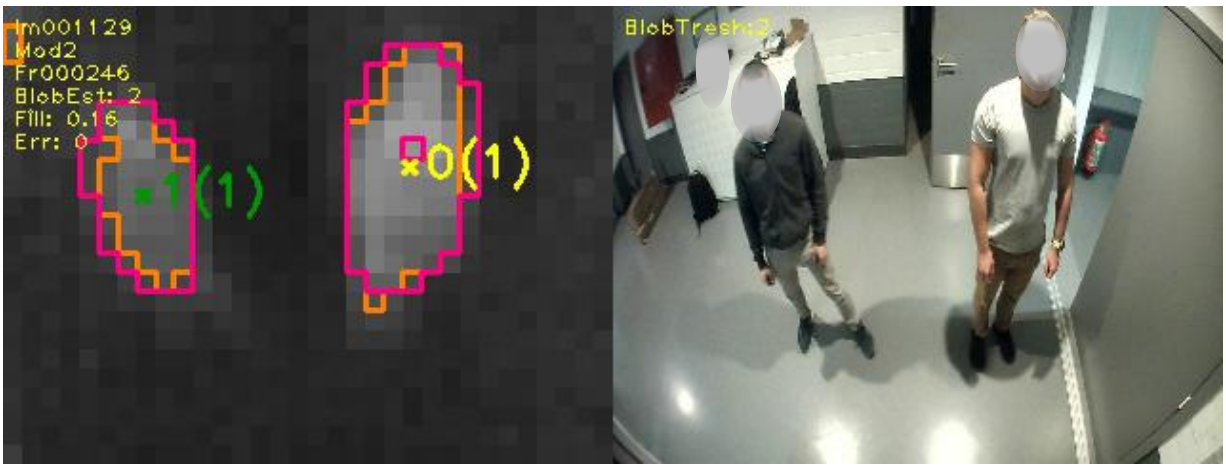


*Figure 10. Two subjects in a corridor (45° mounting)*

### 7.2.4 Disclaimer

Performance results depend on the specific installation, environmental conditions, sensor placement, and host MCU performance.
Melexis does not guarantee that test results obtained under one setup will generalize to all deployment conditions.
Users are responsible for validating system performance in their intended application environment.

# 8 Performance metrics

This section summarizes the key performance characteristics of the people-detection library running on typical embedded microcontroller platforms. Actual results depend on the target MCU architecture, clock frequency, memory configuration, and compiler optimization settings. Integrators should evaluate performance on their specific hardware.
Memory footprint and performance characteristics may change between library releases due to optimizations or feature additions. Consult the documentation included with your release package for version-specific specifications.

## 8.1 Execution time benchmarks

The library is optimized for real-time operation on resource-constrained systems.
Typical processing includes:

- Thermal frame preprocessing
- Background thresholding
- Blob extraction and filtering
- Person-count estimation

Execution time per frame: up to 0.0625 seconds per frame
Frame rate supported: up to 16 frames per second.

## 8.2 Memory footprint

The library uses static memory allocation only. No dynamic (malloc) or OS-dependent memory functions are used.

Memory components include:

- Internal working buffers
- Blob tracking structures
- Detection metadata
- Optional temporary buffers for filtering or noise reduction

Total RAM usage, depending on the selected algorithm mode:

- Full mode: 40 KB RAM
- Simple mode: 14 KB RAM

Flash footprint of compiled binary: 90 KB

## 8.3 CPU utilization

The algorithm runs entirely on the MCU's CPU. GPU or hardware acceleration is not required.

*Table 1. CPU utilization*

| Metric | Simple Mode | Full Mode |
|---|---|---|
| Processing time per frame | 1.5-2.0 ms | 57 ms |
| CPU @ 8 fps (180 MHz) | 1.2-1.6% | 45.6% |
| CPU @ 16 fps (180 MHz) | 2.4-3.2% | 91.2% |

Note 1. The algorithm executes as a single linear processing sequence without internal idle or sleep states. As a result, the peak processing load is effectively equal to the average load for each frame. No separate processing spike exists for individual stages such as blob extraction.

Note 2. All algorithm stages are fully preemptible. The library does not use long critical sections or disabled-interrupt regions, making it safe to interrupt execution with ISRs or higher-priority tasks in real-time systems.

# 9 Security Considerations

This section describes the security aspects relevant to the integration of the library.

The library operates entirely on device-local data coming from the MLX90642 sensor over I2C and does not include any networking, cloud communication, or remote interfaces. For this reason, the security attack surface is minimal and depends primarily on the integrator's system architecture.

## 9.1 Known vulnerabilities

At the time of release, no security vulnerabilities are known.

- The library does not perform dynamic memory allocation and therefore avoids common heap-related vulnerabilities.
- All external interactions are limited to user-supplied I2C read/write callbacks, reducing exposure to external attack vectors.

Because the library is shipped in pre-compiled binary form, the internal implementation cannot be altered by malicious third parties unless the overall system is compromised.

## 9.2 Secure coding practices used

The following secure coding principles are applied during implementation:

- Static memory allocation
- Avoids buffer overflows related to dynamic allocation.
- Range checks on configuration parameters
- All parameters in mlx9064x_isp_people_detection_config_t are validated before use.
- No use of global writable state that can be accidentally shared between threads (except for controlled internal state machines).
- No external input parsing
- The library operates on numeric sensor data only, eliminating risks from malformed strings or external user input.
- Internal bounds-checking on pixel operations
- Ensures loops and indexing remain within the 32×24 pixel array.

The library avoids undefined behaviour and complies with standard C99 restrictions relevant to embedded environments.

## 9.3 Input validation

The main inputs to the library are:

1. Temperature data from the MLX90642
2. Configuration parameters

The library enforces input constraints as follows:

**Configuration validation:**

- Ensures max_blobs, footprint, threshold multipliers, and temperature ranges fall within supported limits.
- Defaults are applied for out-of-range parameters.

**Frame data validation:**

- Basic structural checks (expected frame size, expected alignment).
- Rejection of frames with invalid temperature values (e.g., extreme outliers outside physically meaningful limits).

**Callback validation:**

- Initialization ensures the user provided valid non-null I2C functions.
- These validations help prevent algorithm misbehaviour, false detections, or crashes due to invalid inputs.

## 9.4 Encryption or data protection methods

This library does not implement its own encryption or data protection layer.

- All interaction occurs over the microcontroller's local I2C bus, which typically operates within a trusted hardware boundary.
- No network communication, cloud transmission, or wireless connectivity is included or required.

- If end applications exchange detection results externally (e.g., UART, BLE, Wi-Fi), it is the responsibility of the integrator to use secure protocols (TLS, authenticated BLE, etc.).

### 9.5 No liability disclaimer

This library is provided “as is”, without any explicit or implied warranty.
The authors and distributors do not accept liability for:

- Incorrect system integration
- Data loss or corrupted sensor readings
- Malfunction caused by improper electrical or thermal operating conditions
- Any damages arising from the use of this library in commercial, industrial, or safety-critical systems

Use of this library is at the user’s own risk.
It is the responsibility of the integrator to ensure appropriate safety, security, and validation measures in their final application.

## 10 Packaging & Distribution

This section describes how the library is packaged, versioned, and delivered to customers. The library is distributed free of charge for use exclusively with the MLX90642 infrared temperature sensor. Access to downloads requires a registered MyMelexis account. Testing is available using the People Detection Evaluation Kit.

### 10.1 Versioning scheme

The library uses Semantic Versioning (SemVer):

MAJOR.MINOR.PATCH

MAJOR — Increased when breaking API changes are introduced
MINOR — Increased when new features or capabilities are added
PATCH — Increased for bug fixes, performance improvements, and non-breaking internal changes

Example: v0.5.0 (initial release)

### 10.2 Where to download

The library is distributed exclusively through MyMelexis, the official Melexis customer portal.
Access requires a registered MyMelexis account.
Registration can be arranged by your Melexis sales representative.
Only customers working with the MLX90642 sensor are granted access.

Once logged in, users can download:

- Precompiled binaries
- Documentation (datasheet, integration guides)
- Example code
- Release notes

### 10.3 Compiled binaries

Each release includes a set of precompiled binaries optimized for a range of MCU architectures. The following packages are typically provided:

1. **ARM Cortex-M**: Precompiled static libraries for M0, M3, and M4 (M4 build is forward-compatible with M7/M33/M85)
2. **ARM Cortex-A**: Universal static library targeting 64-bit Linux (aarch64)
3. **x86_64 Linux**: Static library for desktop simulation and integration testing

Builds for additional architectures are available upon request.

All binaries are:

- Built with standard C99
- Hardware-agnostic except for I2C integration
- Provided as static libraries (.a) or precompiled object files (.o) depending on target
- No source code is included.

## 11 Support & Maintenance

This section describes how customers can obtain support, request new features, and report issues related to the library. Support is provided to registered Melexis customers working with the MLX90642.

### 11.1 Support channels

MyMelexis Portal provides access to documentation, downloads, release notes, and updates.

Technical support is available to customers and potential customers either through their regional sales representative or via the contact form on the Melexis website, which routes inquiries directly to the application engineering team.

Customers can reach out for:

- Integration assistance
- Questions regarding functionality
- Registration requests for new team members

No public forum or community support is provided for this library.

### 11.2 Maintenance policy

Major functional extensions may be introduced in new MINOR or MAJOR releases.

Melexis reserves the right to discontinue maintenance for older versions after a reasonable support window.

The library is provided in binary form only, and no custom builds are offered except through explicit agreements with Melexis.

### 11.3 Bug report & Feature request

Customers can report bugs or request new features through sales representatives of their region or via the contact form on the Melexis website.

## 12 List of tables



## 13 List of figures



## 14 Revision history

*Table 2. Revision History*

| Revision | Date | Change history |
|---|---|---|
| 001 | 04-12-2025 | Creation |

## 15 Disclaimer

*The content of this document is believed to be correct and accurate. However, the content of this document is furnished "as is" for informational use only and no representation, nor warranty is provided by Melexis about its accuracy, nor about the results of its implementation. Melexis assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. Customer will follow the practices contained in this document under its sole responsibility. This documentation is in fact provided without warranty, term, or condition of any kind, either implied or expressed, including but not limited to warranties of merchantability, satisfactory quality, non-infringement, and fitness for purpose. Melexis, its employees and agents and its affiliates' and their employees and agents will not be responsible for any loss, however arising, from the use of, or reliance on this document. Notwithstanding the foregoing, contractual obligations expressly undertaken in writing by Melexis prevail over this disclaimer.*

*This document is subject to change without notice, and should not be construed as a commitment by Melexis. Therefore, before placing orders or prior to designing the product into a system, users or any third party should obtain the latest version of the relevant information.*
*Users or any third party must determine the suitability of the product described in this document for its application, including the level of reliability required and determine whether it is fit for a particular purpose.*

*This document as well as the product here described may be subject to export control regulations. Be aware that export might require a prior authorization from competent authorities. The product is not designed, authorized or warranted to be suitable in applications requiring extended temperature range and/or unusual environmental requirements. High reliability applications, such as medical life-support or life-sustaining equipment or avionics application are specifically excluded by Melexis. The product may not be used for the following applications subject to export control regulations: the development, production, processing, operation, maintenance, storage, recognition or proliferation of:*
*1. chemical, biological or nuclear weapons, or for the development, production, maintenance or storage of missiles for such weapons;*
*2. civil firearms, including spare parts or ammunition for such arms;*
*3. defense related products, or other material for military use or for law enforcement;*
*4. any applications that, alone or in combination with other goods, substances or organisms could cause serious harm to persons or goods and that can be used as a means of violence in an armed conflict or any similar violent situation.*

*No license nor any other right or interest is granted to any of Melexis' or third party's intellectual property rights.*

*If this document is marked “restricted” or with similar words, or if in any case the content of this document is to be reasonably understood as being confidential, the recipient of this document shall not communicate, nor disclose to any third party, any part of the document without Melexis’ express written consent. The recipient shall take all necessary measures to apply and preserve the confidential character of the document. In particular, the recipient shall (i) hold document in confidence with at least the same degree of care by which it maintains the confidentiality of its own proprietary and confidential information, but no less than reasonable care; (ii) restrict the disclosure of the document solely to its employees for the purpose for which this document was received, on a strictly need to know basis and providing that such persons to whom the document is disclosed are bound by confidentiality terms substantially similar to those in this disclaimer; (iii) use the document only in connection with the purpose for which this document was received, and reproduce document only to the extent necessary for such purposes; (iv) not use the document for commercial purposes or to the detriment of Melexis or its customers. The confidentiality obligations set forth in this disclaimer will have indefinite duration and in any case they will be effective for no less than 10 years from the receipt of this document.*

*This disclaimer will be governed by and construed in accordance with Belgian law and any disputes relating to this disclaimer will be subject to the exclusive jurisdiction of the courts of Brussels, Belgium.*

*The invalidity or ineffectiveness of any of the provisions of this disclaimer does not affect the validity or effectiveness of the other provisions.*
*The previous versions of this document are repealed.*



*IATF 16949 and ISO 14001 Certified*

For the latest revision of this document, visit www.melexis.com/library-people-detection
Happy to help you! www.melexis.com/contact