

ESP32-Lepton Component Documentation

Daniel Kampert

Version 1.0.0, 2026-01-27

Table of Contents

1. Abstract	1
2. Overview	2
2.1. Key Features	2
2.2. Supported Hardware	2
3. Architecture	3
3.1. Core Module	3
3.2. Communication Interfaces	3
3.3. Data Acquisition	3
4. Module Documentation	4
4.1. Main Driver (lepton)	4
4.2. CCI Commands (lepton_cci)	4
4.3. CCI Low-Level (cci)	4
4.4. VoSPI Interface (vospi)	4
4.5. Frame Capture (lepton_capture)	4
5. Getting Started	5
5.1. Prerequisites	5
5.2. Basic Usage	5
6. Error Handling	7
7. API Reference	8
8. License	9
9. Contact	10

Chapter 1. Abstract

This documentation provides a comprehensive guide to the ESP32-Lepton component, a driver library for FLIR Lepton thermal imaging sensors on ESP32 platforms.

Chapter 2. Overview

The ESP32-Lepton component is a complete driver implementation for FLIR Lepton 3.0 and 3.5 thermal imaging sensors. It provides:

- **CCI (Command and Control Interface):** I2C-based communication for sensor configuration and control
- **VoSPI (Video over SPI):** High-speed SPI interface for thermal image data acquisition
- **Frame Capture:** Efficient frame synchronization and data handling
- **Image Processing:** Built-in color palette support and thermal data processing

2.1. Key Features

- Support for Lepton 3.0 and 3.5 sensors
- Radiometric and non-radiometric modes
- Hardware reset and power management
- Telemetry data extraction
- Multiple color palettes (Iron, Rainbow, Grayscale, etc.)
- DMA-based SPI transfers for optimal performance
- FreeRTOS integration

2.2. Supported Hardware

Sensor Model	Part Number	Radiometric
Lepton 3.0	500-0726-01	No
Lepton 3.5	500-0771-01	Yes

Chapter 3. Architecture

The component is organized into several modules:

3.1. Core Module

- **lepton**: Main driver interface and initialization

3.2. Communication Interfaces

- **lepton_cci**: High-level CCI command implementation
- **CCI**: Low-level CCI (Command and Control Interface) over I2C
- **VoSPI**: Low-level VoSPI (Video over SPI) interface

3.3. Data Acquisition

- **lepton_capture**: Frame capture and synchronization

Chapter 4. Module Documentation

4.1. Main Driver (lepton)

Core driver implementation providing initialization, configuration, and control functions.

4.2. CCI Commands (lepton_cci)

High-level CCI command interface for sensor configuration and status queries.

4.3. CCI Low-Level (cci)

Low-level I2C communication layer implementing the Lepton CCI protocol.

4.4. VoSPI Interface (vospi)

SPI-based video interface for high-speed thermal data acquisition.

4.5. Frame Capture (lepton_capture)

Frame synchronization and data capture implementation.

Chapter 5. Getting Started

5.1. Prerequisites

- ESP-IDF v5.0 or later
- FLIR Lepton 3.0 or 3.5 sensor module
- SPI and I2C connections to ESP32

5.2. Basic Usage

```
#include "lepton.h"

// Initialize Lepton configuration
Lepton_Conf_t config = {
    .CCI = {
        .I2C_Read = my_i2c_read_func,
        .I2C_Write = my_i2c_write_func,
        .Address = LEPTON_I2C_ADDRESS
    },
    .VoSPI = {
        .Host = SPI2_HOST,
        .Interface = {
            .clock_speed_hz = 20000000,
            .spics_io_num = GPIO_NUM_10
        },
        .Master = {
            .sclk_io_num = GPIO_NUM_12,
            .miso_io_num = GPIO_NUM_13
        },
        .DMA = SPI_DMA_CH_AUTO
    },
    .VSync = GPIO_NUM_5,
    .Reset = &my_reset_func,
    .PowerDown = &my_powerdown_func
};

Lepton_t device;
Lepton_Result_t status;

// Initialize the device
if (Lepton_Init(&device, &config, &status) != LEPTON_ERR_OK) {
    // Handle error
}

// Start capturing
if (Lepton_StartCapture(&device, my_frame_callback, &status) != LEPTON_ERR_OK) {
    // Handle error
}
```

}

Chapter 6. Error Handling

All API functions return a `Lepton_Error_t` error code. The `Lepton_Result_t` structure provides detailed status information from the sensor.

```
Lepton_Result_t status;
Lepton_Error_t error = Lepton_SomeFunction(&device, &status);

if (error != LEPTON_ERR_OK) {
    ESP_LOGE(TAG, "Operation failed with error: %d", error);
    ESP_LOGE(TAG, "CCI Status Code: 0x%04X", status.StatusCode);
}
```

Chapter 7. API Reference

For detailed API documentation, please refer to the individual module documentation:

- [Main Driver API](#)
- [CCI Commands API](#)
- [CCI Low-Level API](#)
- [VoSPI API](#)
- [Frame Capture API](#)

Chapter 8. License

Copyright © Daniel Kampert, 2026

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Website: <https://www.kampis-elektroecke.de>

Chapter 9. Contact

For errors and commissions, please contact: DanielKampert@kampus-elektroecke.de