Biometric Keypoint Tracking Solution

Application Note

based upon AISC110C-AI-DB

Document Revision 1.0

28 May 2025

Table of Contents

1 General Description

Many embedded systems today are evolving to include biometric capabilities, such as facial detection, gesture tracking, or head pose estimation-for access control, smart monitoring, or human-machine interfaces. These applications often need to work reliably in real-time, even in low-light conditions, with minimal power and resource usage. Cloud-based solutions are impractical in such settings due to latency, privacy, and connectivity issues. The challenge is clear: deliver responsive and accurate biometric tracking directly on device, with minimal hardware requirements.

AIStorm's Biometric Keypoint Tracking Solution is a complete image acquisition and processing system consisting of the AISC110C "Cheetah" IC and the NXP MCX N947 Microcontroller with integrated NPU for Edge AI applications as well as advanced Biometric Keypoint Tracking model and GUI.

Cheetah is a high-speed imager, being able to produce good quality low-resolution images with very low latency. Cheetah Evaluation Kit uses high processing speeds to perform complex tasks on a microcontroller. A big advantage of Cheetah is that due to the lower resolution it is possible to use bigger pixel sizes for the same sensor size. This means more photons reach an individual pixel at a given exposure time, making this imager ideal to be used in low light scenarios.

Table 1: Biometric Keypoint Tracking Model Specification

Total inference time	26 ms
Memory usage	141 kB
Layers	42

The NXP MCU runs an advanced AI model for biometric keypoint tracking. This model is used to detect one or multiple faces and their unique keypoints within each captured frame. Each detected face is tracked by a black rectangular boundary surrounding it; the keypoints of a detected face are tracked using white point markers at the respective positions in the live stream. An overlay of both described tracking visualizations is shown exemplary in Figure 1.

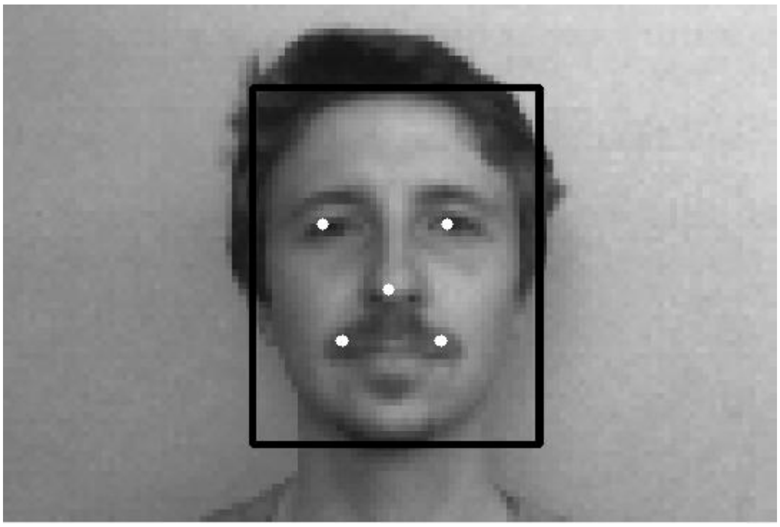

Figure 1: Visualization of Face Detection and Biometric Keypoint Tracking

The application signal processing chain consists of 4 blocks: The Cheetah imager, a physical neuron first layer of the model, digitizer and MCU. These are also shown in Figure 2. The first layer is then coupled to the NXP microcontroller, where the rest of the model is run. The outputs are coordinates that can be used for head pose estimation or face recognition.

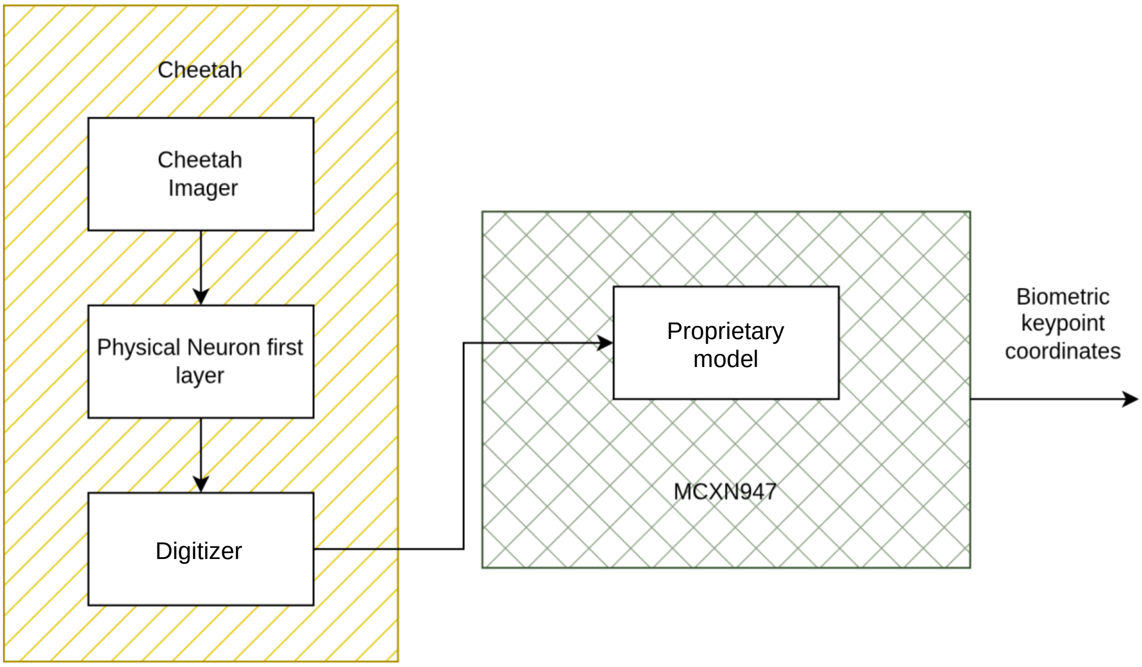

Figure 2: Signal processing chain of Biometric Keypoint Tracking application

Figure 3 shows details about the model accuracy in this application:

Parameter	Value
Data latency [ms]	10
Sampling Rate [fps]	30
Horizontal Field of View [deg]	40
Key point accuracy x at 1m	1 cm
Key point accuracy y at 1m	1 cm
Rotation range Y (roll)	$\pm 25°$
Rotation range X (pitch)	$\pm 25°$
Rotation range Z (yaw)	$\pm 40°$
Detection range [m]	0.4-2.5
Ambient light (w. infrared LED) [lux]	0

Figure 3: Model accuracy

In order to use this application, order the AISC110C-AI-DB Cheetah Evaluation Kit online under AISC110C-AI-DB AIStorm, Inc | Development Boards, Kits, Programmers | DigiKey.

Contact AIStorm under Contact Us – AIStorm for further documentation and software resources.

1.1 Scope of the document

This document provides instructions on how to install and run Biometric Keypoint Tracking on the Cheetah Evaluation Kit. A detailed description of the installation process is given, as well as a description of the firmware implemented on the MCU. The document provides the information in the following order:

- Overview of the Cheetah Evaluation Kit hardware
- Environment Installation (for Windows OS)
- Biometric keypoint tracking using the Cheetah EVK Application
- Cheetah MCU firmware description

1.2 Features

- **80x120 Ultra High-Speed Imager**
  - Power down mode
  - High IR sensitivity (>70%QE@850nm)
- **Charge-Sensor Pixel Imager**
  - Direct pixel charge processing
  - Charge to digital conversion
- **SPI Interface**
- **MCX N947 Microcontroller for image readout and processing**
  - 32-bit Arm® Dual Cortex-M33 TrustZone® microcontroller
  - eIQ® Neutron N1-16 Neural Processing Unit (NPU)
  - 512 KB of on-chip SRAM memory
  - 2 MB Flash memory
- **128 MByte on-board PSRAM for image recording**
- **USB-C user interface for PC connection**

2 Cheetah Evaluation Kit

2.1 Hardware Overview

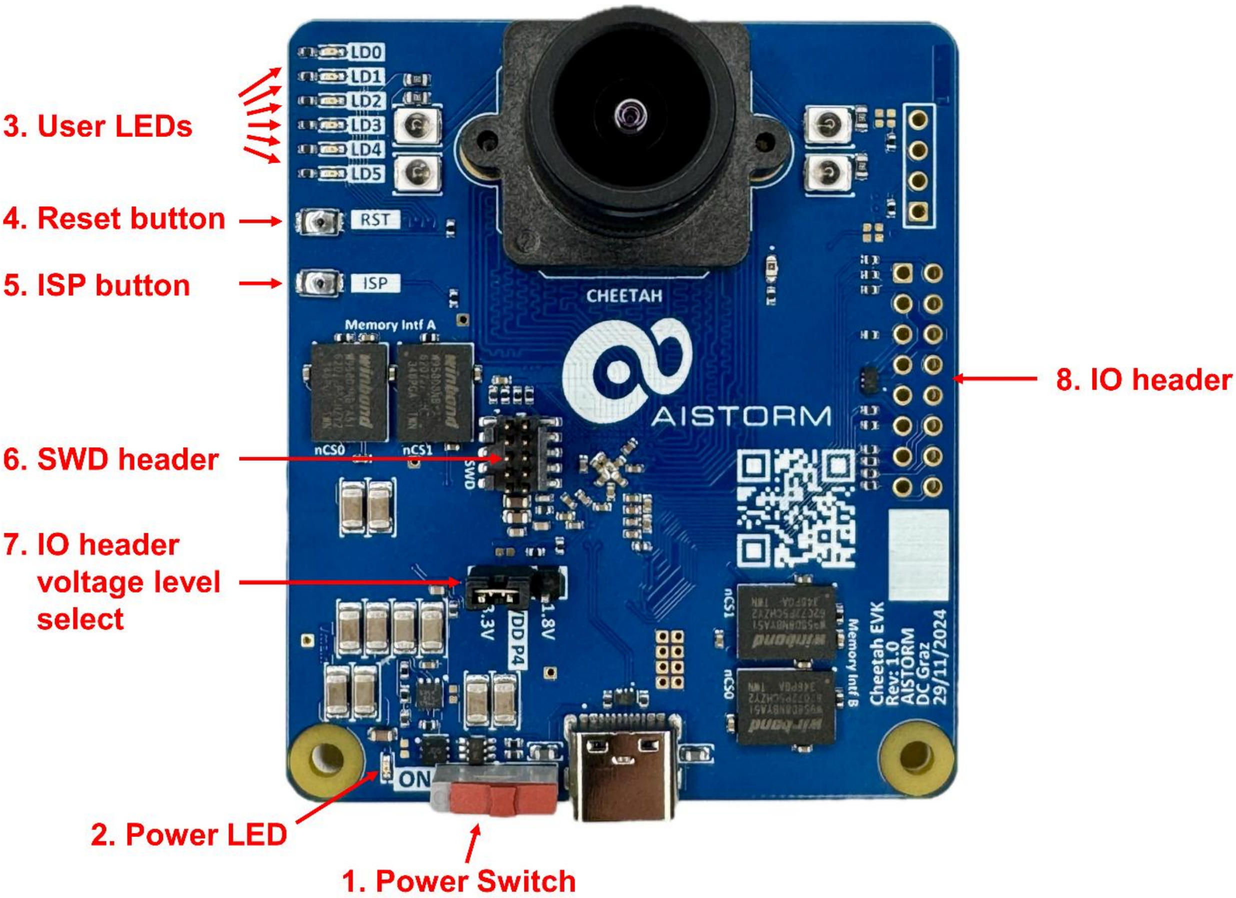

Figure 4: Cheetah Evaluation Board

Description of the numbered Items from Figure 4:

1. **Power Switch**: Turns the power for the entire Evaluation Board on or off
2. **Power LED**: Indicates that all power rails are active and in their required range
3. **User LEDs**: MCU-driven LEDs controlled by the firmware. The PC application modifies the signals shown by the LEDs upon its initial connection after power-up. They indicate the following:

Table 2: LED configuration in the firmware (v1.0.1)

	Power up	Connection with GUI
LD0	Heartbeat 1 sec period	Heartbeat 1 sec period
LD1	Startup successful	Frame captured (toggle)
LD2	Cheetah detected	Frame streamed (toggle)
LD3	Cheetah configured	AI inference run (toggle)
LD4	AI initialization failed	AI has detected a Face
LD5	Firmware error	AEC active

4. **Reset button**: When pressed, the MCU will be reset.

5. **ISP button**: Used to set the MCU into ISP mode when pressed during power up or reset. To flash the MCU via the USB connection.
6. **SWD header**: Header for debugging and programming the MCU via SWD
7. **IO header voltage level select**: Jumper for selecting the voltage Level for the IO header.
8. **IO header**: Header for connecting to external hardware.

2.2 Board dimensions

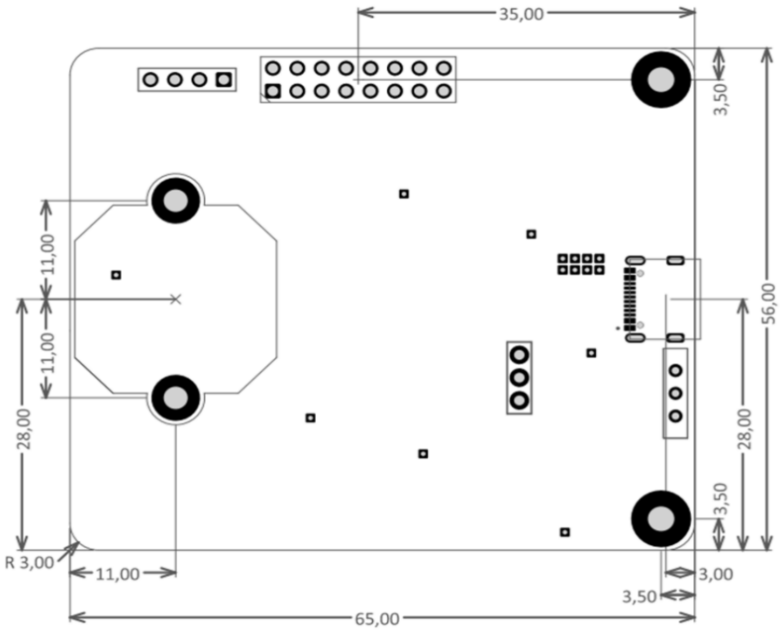

Figure 5: Dimensions of Cheetah Evaluation Board in mm

2.3 Block diagram

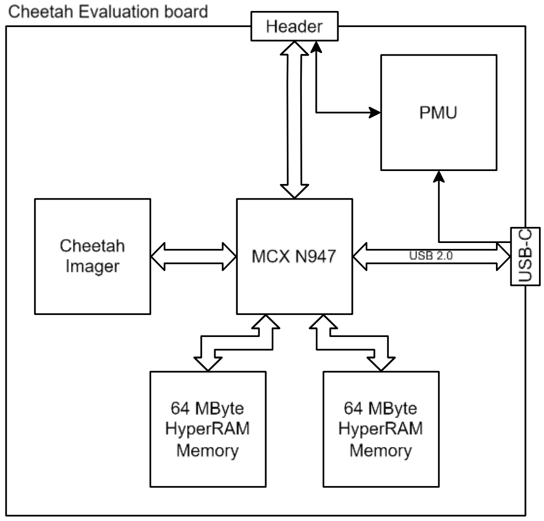

Figure 6: Hardware block diagram of Cheetah Evaluation Board

2.4 IO Header

The board has a 16-pin 2-row header with a 2.54 mm pitch for general purpose and communication with external attachments.

Table 3: IO Header pinout

Pin Number	Signal Name	Signal Type	Signal Level	Description
1	TXD/SCL	Output	VDD P4	UART serial data transmit
2	RXD/SDA	Input	VDD P4	UART serial data receive
3	p3V3out	Power Output	3.3 V	3.3 V power output from PMU
4	GND	Power	NA	Ground Reference
5	GPIO0	IO	VDD P4	General-purpose input/output 0
6	p5V	Power	5.0 V	External 5.0 V power input is not compatible with Raspberry Pi
7	GPIO1	IO	VDD P4	General-purpose input/output 1, with Raspberry Pi must be used as input only
8	p5V	Power	5.0 V	External 5.0 V power input is not compatible with Raspberry Pi
9	MOSI	Input	VDD P4	SPI serial data input
10	GND	Power	NA	Ground Reference
11	MISO	Output	VDD P4	SPI serial data output
12	GPIO2	IO	VDD P4	General-purpose input/output 2
13	SCLK	Input	VDD P4	SPI serial clock input
14	nCS	Input	VDD P4	SPI low-active chip select input
15	GND	Power	NA	Ground Reference
16	GPIO3	IO	VDD P4	General-purpose input/output 3

The Header is partially compatible with the Raspberry Pi GPIO Header. The jumper VDD_P4 allows you to choose between 3.3 V or 1.8 V as signal level for the IO Header.

2.5 Lens description

The Cheetah Evaluation Kit is shipped with a ready-to-use S-mount (M12) lens installed as shown in Figure 4. The focus of the lens is kept in position upon delivery with a locknut and can be adjusted if needed. As displayed in Figure 5 the distance between the two lens holder mounting screw holes is 22 mm. The 3.6 mm effective focal length (EFL) of the installed lens in combination with the physical imager size of Cheetah (2.4 mm by 1.6 mm) results in an angle field of view (AFOV) of 25.0 ° vertically and 36.8 ° horizontally (43.4 ° diagonally). This lens can be replaced with any other suited S-mount (M12) lens to capture an image with e.g. a different AFOV for a specific use-case. Figure 7 shows the installed lens and the dimensions of the used lens holder.

Lens weblink:
https://www.baslerweb.com/en-us/shop/evetar-lens-m13b03618ir-f1-8-f3-6mm-1-3/

Lens holder weblink:
https://www.vision-dimension.com/en/lenses/lens-accessories/lensholder/objektivhalterung-lensholder-m12x0-5-22/14-1/666

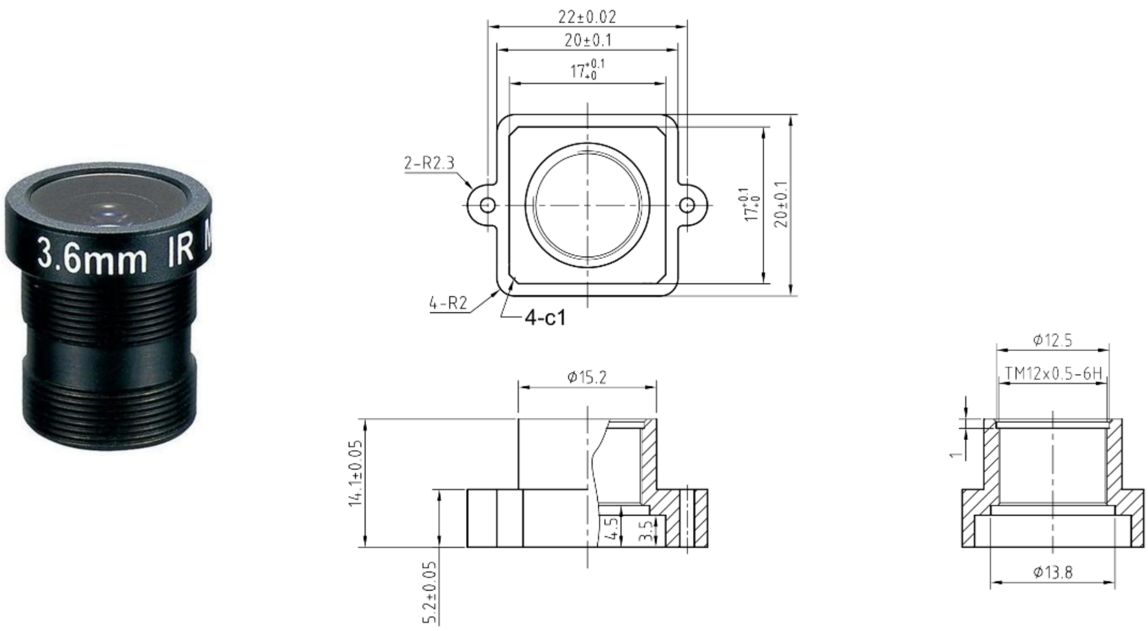

Figure 7: Lens (left) and lens holder (right)

3 Getting Started

This chapter provides instructions on how to install the required software tools.
Please note that while the Cheetah Evaluation Kit can be used with multiple OS hosts, the instructions are only provided for the Windows operating system.

To set up the system for the Evaluation Kit, the following hardware is required:

- Cheetah Evaluation Kit (see Figure 4)
- USB-C cable to connect the Cheetah Evaluation Kit to a host device

Installation of the Cheetah EVK Application (for Windows):

1. If you do not have the CheetahEVKApplication.exe saved locally already, the AISC110C-AI-DB folder can be downloaded on request

2. Inside the folder you can find the executable CheetahEVKApplication.exe

3. Continue with Chapter 4

4 Biometric Keypoint Tracking with the Cheetah EVK Application

4.1 Start-up

1. After connecting the board via USB, switch ON the board (if not done so already) and wait until the LEDs LD1 to LD3 light up (and stay) green, no red LED should light up.
2. Start the application CheetahEVKApplication.exe from the extracted repository
3. The GUI of Cheetah EVK Application v1.0 right after start-up is shown in Figure 8:

4.2 Main Control Panel

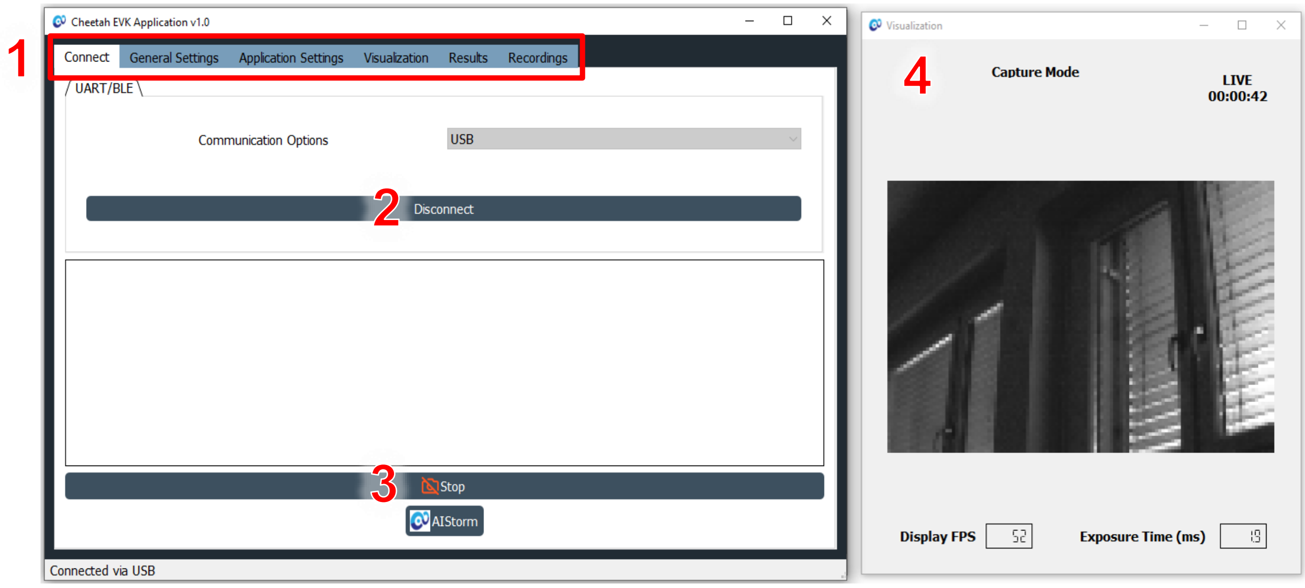

Figure 8: Main Control Panel of Cheetah EVK Application

Here is an explanation of the main control elements from Figure 8:

1. Switch between control tabs
2. START/STOP the connection with Cheetah
3. START/STOP the capturing of Cheetah
4. Detached Visualization window

4.3 Device Settings and Information

4.3.1 Device Information

Figure 9 shows version and device information for the Evaluation Board, and by pressing "Read All" you can request all this information.

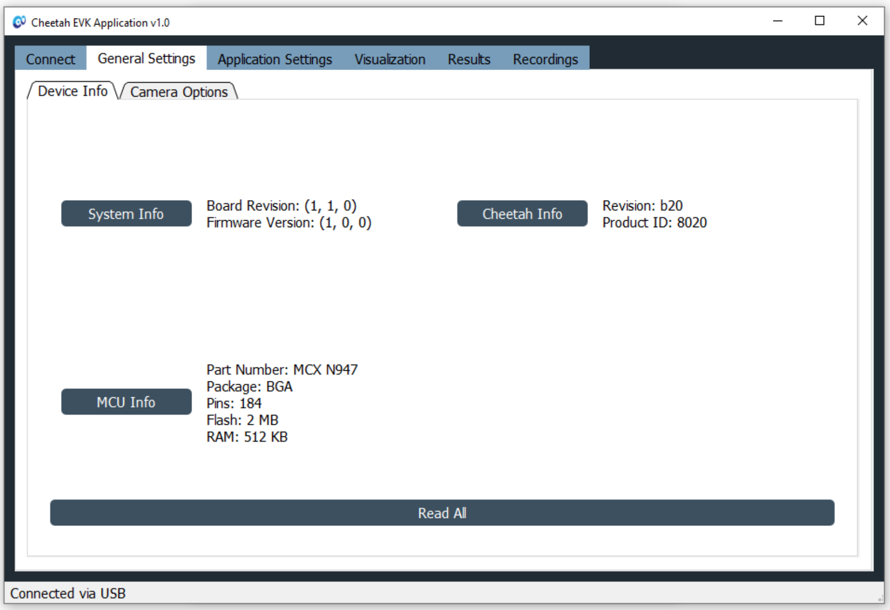

Figure 9: Device Info Tab of Cheetah EVK Application

4.3.2 Camera Options

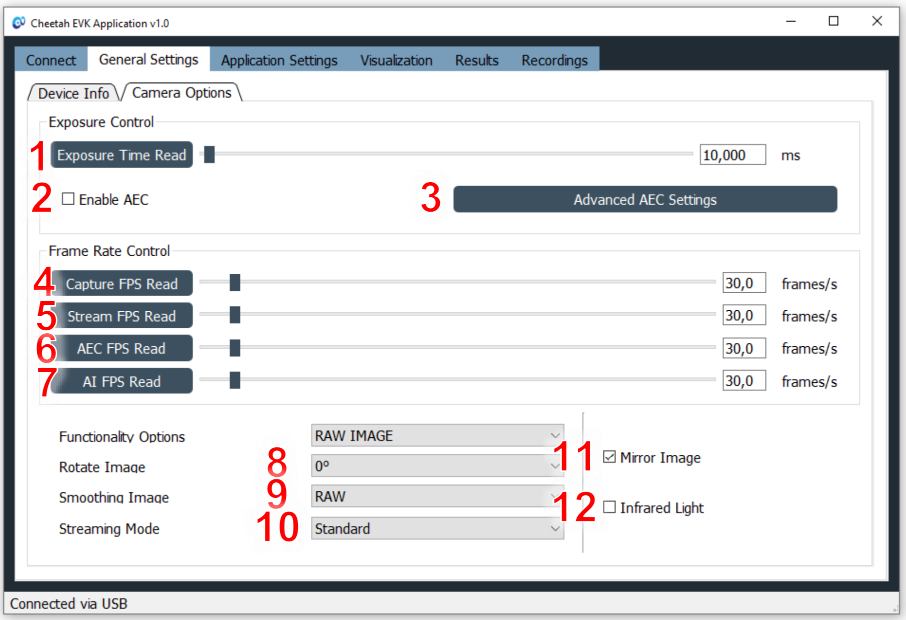

Figure 10: Camera Options Tab of Cheetah EVK Application

Here is an explanation of the control elements from Figure 10:

1. Read actual exposure time in milliseconds (value is limited by the capture FPS); Move the slider to change the value or write the value in the textbox (small adjustments can be done by either clicking on the left/right of the slider or pressing the buttons left/right on the keyboard)
2. Enable/Disable auto exposure control
3. Open advanced AEC settings (see Section 4.8)
4. Read capturing frame rate (defines how many frames are captured per second); Move the slider to change the value or write the value in the textbox (small adjustments can be done by either clicking on the left/right of the slider or pressing the buttons left/right on the keyboard)
5. Read maximum streaming frame rate (defines maximal number of frames that can be streamed in one second; the actual display FPS is limited by PC and could be lower); Move the slider to change the value or write the value in the textbox (values are restricted by the caption frame rate)
6. Read AEC frame rate (defines how often the exposure time is evaluated); Move the slider to change the value or write the value in the textbox (values are restricted by the caption frame rate)
7. Read AI frame rate (defines how many frames per second are evaluated by the detection model); Move the slider to change the value or write the value in the textbox (values are restricted by the caption frame rate and the evaluation time)
8. Rotate stream (0°, 90°, 180° or 270°)
9. Smoothing stream with different filters
10. Switch between streaming modes (switch streaming ON/OFF, capturing is always ON)
11. Mirror stream ON/OFF
12. Switch the infrared lights ON/OFF

4.4 Application Settings - Using Biometric Keypoint Tracking

The inference of the Biometric Keypoint Tracking model is done in the eIQ® Neutron N1-16 Neural Processing Unit (NPU) on the MCU. The model has a face detector and additionally detects 5 facial keypoints. The tracking result visualization is shown exemplary in Figure 12. Best performance can be achieved when the face is fully visible in the image, which corresponds to a distance of 1 to 4 meters from the face to the supplied lens. The imager has to stay in the default orientation for this application (General Settings > Camera Options > Rotate Image: set to 0°).

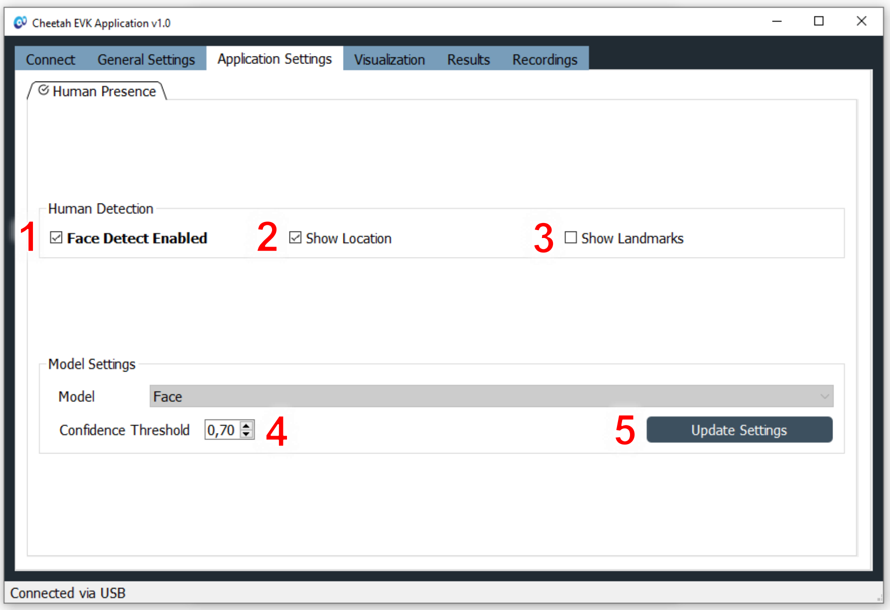

Figure 11: Application Settings Tab of Cheetah EVK Application

Here is an explanation of the control elements from Figure 11:

1. Enable/Disable the detection
2. Switch ON/OFF the visualization of the bounding boxes surrounding detected faces
3. Switch ON/OFF the visualization of the white point markers on biometric keypoints
4. Confidence threshold (values between 0 and 1); High values suppress detections with low confidence
5. Send the changed model settings to the Evaluation Board, which needs to be clicked when any model settings are changed

In Figure 12 the four possible visualization options of Biometric Keypoint Tracking are shown:

- top-left image: No visual feedback in the live stream is activated
- bottom-left image: Only Show Location is enabled
- top-right image: Only Show Landmarks enabled
- bottom-right image: Both Show Location and Show Landmarks are enabled

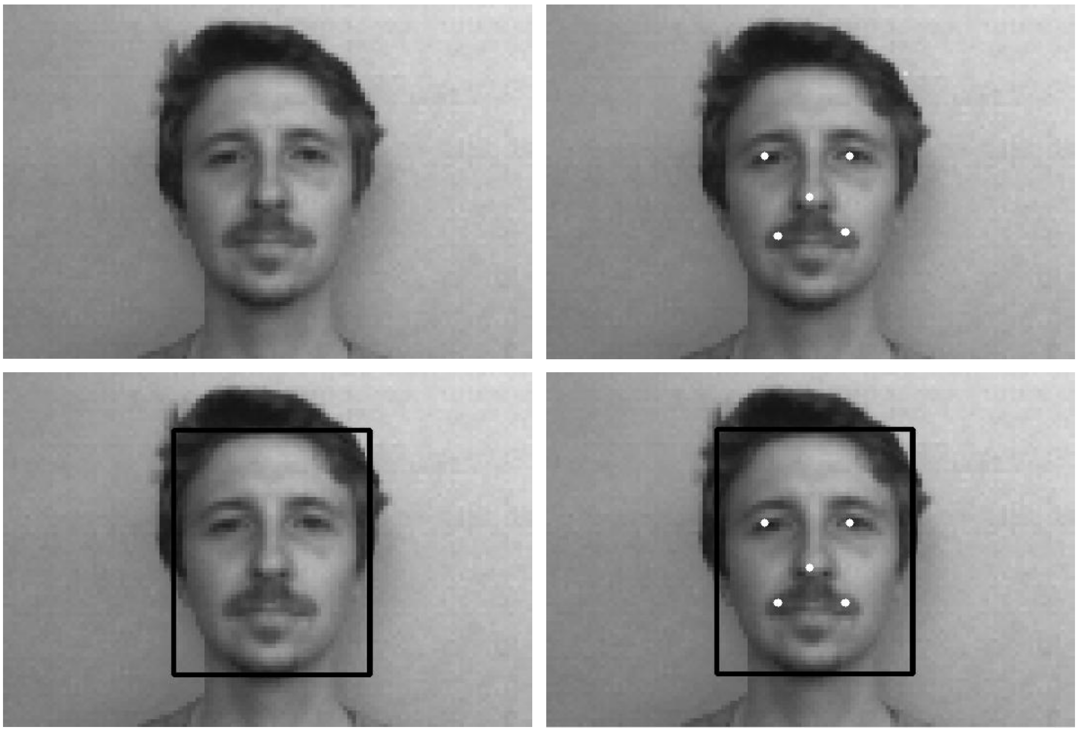

Figure 12: Biometric Keypoint Tracking visualization options

These images are captured using the Biometric Keypoint Tracking live results inside Cheetah EVK Application's Visualization window. For more details on the Visualization window see the following chapter 4.5.

For all four of the above shown cases (as long as Face Detect is enabled) the Results tab shows whether a person is detected in the current frames and how many people are currently detected, as described in more detail in chapter 4.6.

4.5 Visualization

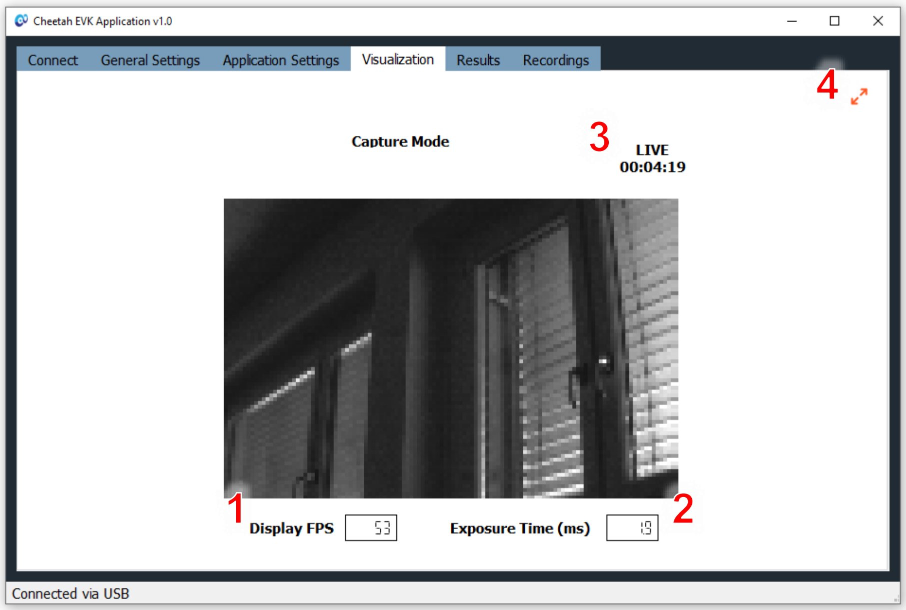

Figure 13: Attached Visualization Tab of Cheetah EVK Application

Here is an explanation of the control elements from Figure 13:

1. Display frame rate (defines how many frames are displayed per second)
2. Actual exposure time
3. Duration of streaming while capturing is ON, else it shows the actual time
4. Detach/Attach the visualization (closing the visualization window is also attaching the window)

Figure 14 shows the empty Visualization and the detached window that now displays the live video after pressing the button marked with 4 above.

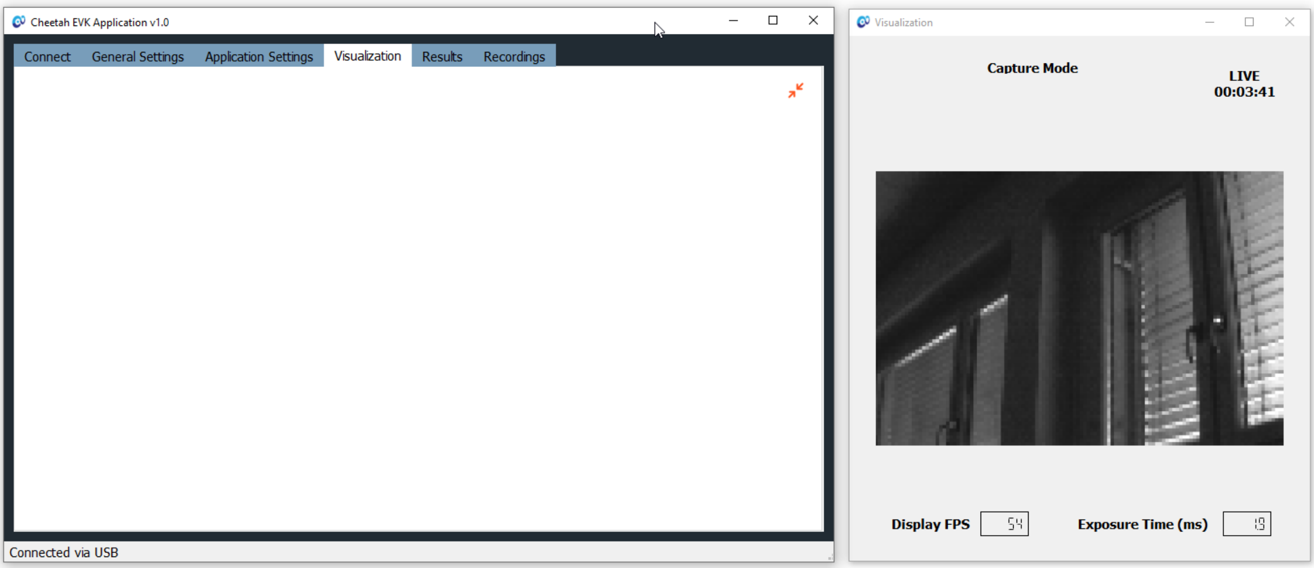

Figure 14: Detached Visualization Tab of Cheetah EVK Application

4.6 Results

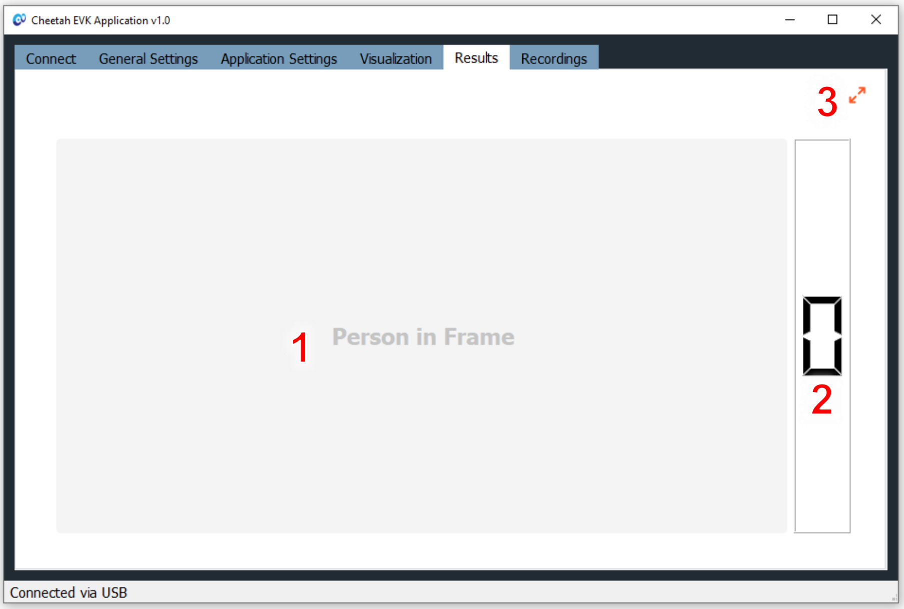

Figure 15: Attached Result Tab of Cheetah EVK Application

Here is an explanation of the marked control elements from Figure 15:

1. Indicator for detection (gray if nothing is detected, blue if a Person/Face is detected)
2. Amount of detections
3. Detach/Attach the results (closing the results window is also attaching the window)

Figure 16 shows how the Results window can be detached from the main GUI.

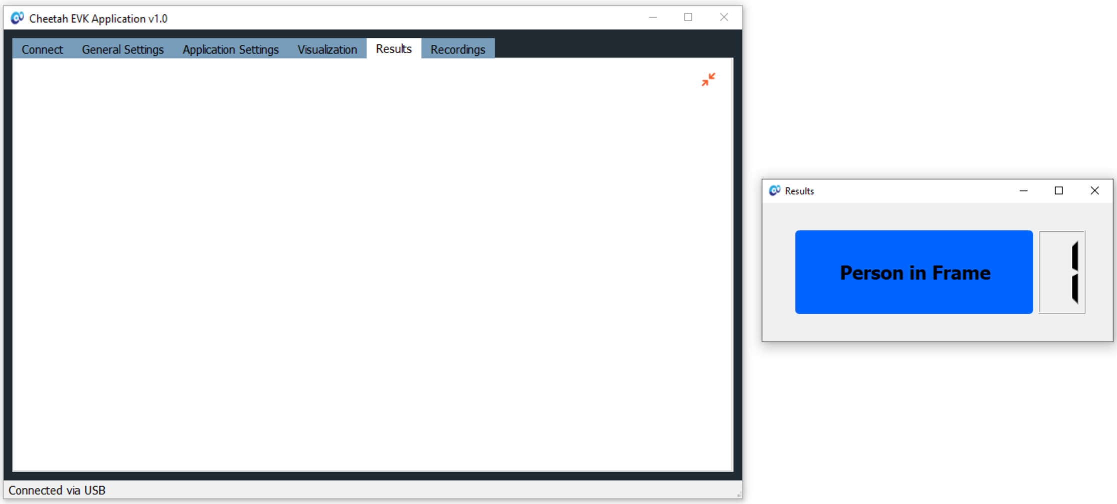

Figure 16: Detached Result Tab of Cheetah EVK Application

4.7 Recordings

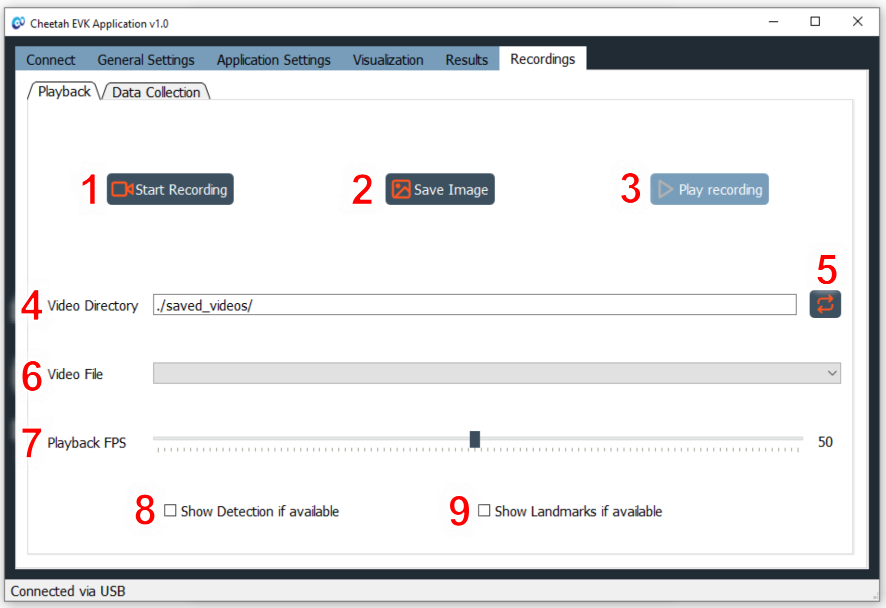

Figure 17: Recordings Tab of Cheetah EVK Application

Here is an explanation of the control elements from Figure 17:

1. Start/Stop recording the stream (only possible if streaming is ON) (save location is "./saved_videos")
2. Save the actual frame as an image (save location is "./saved_images")
3. Play saved recording (only possible if capturing is OFF)
4. Choose location to load the stream
5. Refresh the file list
6. File list from the chosen folder
7. Frame rate for playback (must be chosen before starting the playback)
8. Enable/Disable plotting of the bounding boxes if they were saved during recording (must be chosen before starting the playback)
9. Enable/Disable plotting of the landmarks if they were saved during recording (must be chosen before starting the playback)

4.8 Auto Exposure Tab

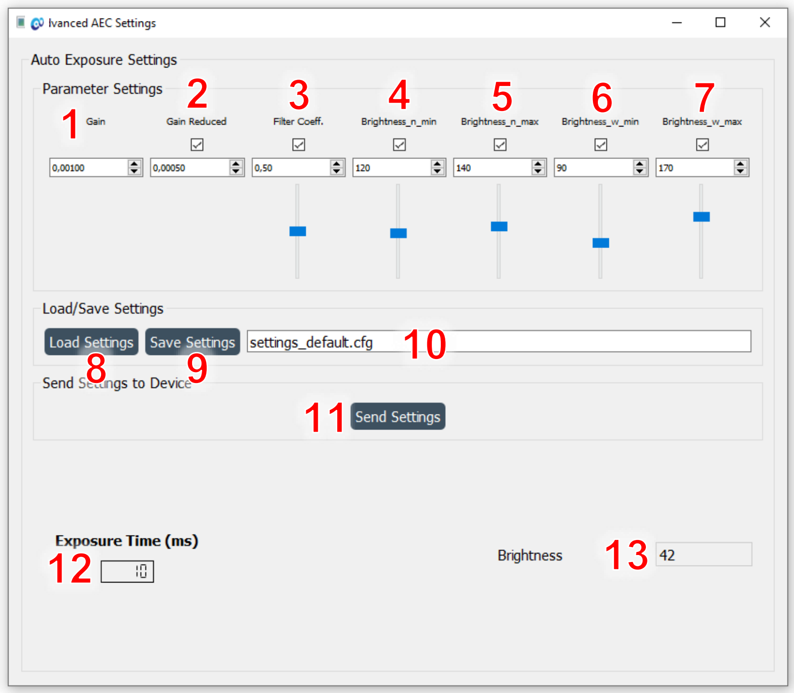

Figure 18: Advanced Auto Exposure Settings Tab of Cheetah EVK Application

Here is an explanation of the control elements from Figure 18:

1. Change the gain to control the step size and speed of the exposure change
2. Reduce the gain if restricted values are reached; switch ON/OFF
3. Filter coefficient (default value: 0.5); switch ON/OFF
4. Set minimum value of the target brightness (narrow band)
5. Set maximum value of the target brightness (narrow band)
6. Set minimum value of the tolerated brightness (narrow band with additional tolerance)
7. Set maximum value of the tolerated brightness (narrow band with additional tolerance)
8. Load saved settings from CFG file
9. Save current settings to CFG file
10. Set path to save the settings
11. Send settings to device, must be clicked when changes should be activated; Eventually the values from the sliders change because of restrictions
12. Current exposure time
13. Current brightness value

The current brightness display is tracking the target brightness when AEC mode is enabled. The maximum possible exposure time depends on the sampling frame rate. If AEC is enabled and no change in exposure time is observed, although the minimum brightness value is way higher than the current brightness value, the maximum possible exposure time is already reached. The image can now only be brightened by lowering the sampling frame rate.

4.9 Troubleshooting

- If the device gets disconnected, go to “Connect” tab and click on “Connect” to connect again. If the connection is still not available, unplug the device and restart the application.
- If the streaming is not correct (e.g. parts of the image are shifted) or the connection fails, disconnect the device and reconnect again.

5 Cheetah MCU Firmware

5.1 Overview

The firmware on the MCU is designed in a modular approach to be able to add applications as desired, as long as they can fit into the given memory space.

5.1.1 Features

- Capture frames into on-chip memory at a settable frame rate
- Live stream frames to PC with a settable frame rate
- Face and Landmark detection on MCU, results are transmitted to PC
- Adjustable auto-exposure controller
- Based on FreeRTOS and MCUXpresso SDK v24.12.00
- Infrared light
- USB 2.0 communication
- Remotely controllable via PC

5.1.2 Firmware Update

For details on how to flash new firmware please refer to the document:

“AISC110C-AI-DB_Firmware_Update_Guide.pdf”

5.1.3 Firmware Structure

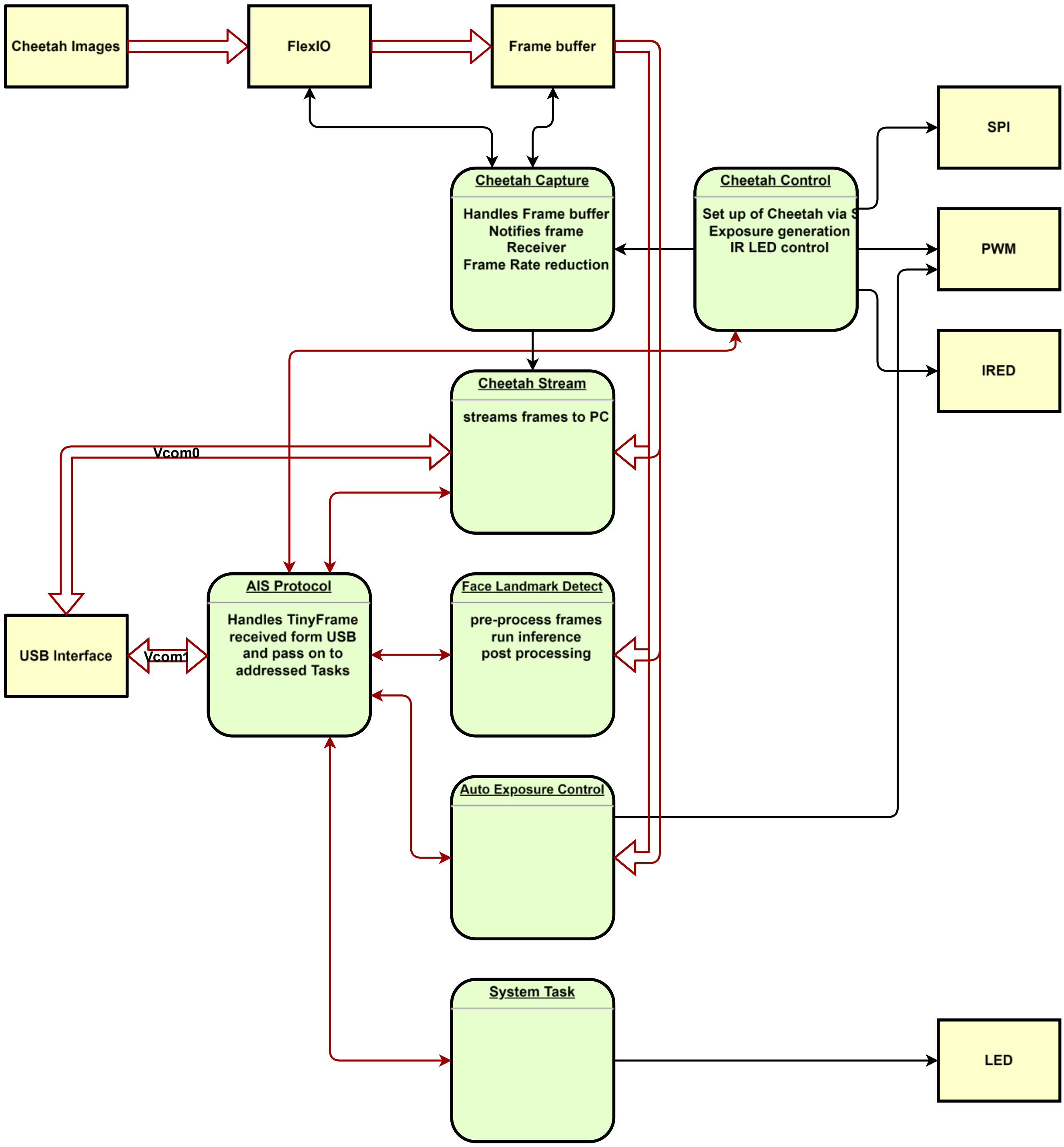

Figure 19: Overview of the Cheetah MCU firmware

5.2 AIS Protocol

The message protocol for communication between the PC and the MCU is based on the TinyFrame library (MightyPork/TinyFrame: A simple library for building and parsing data frames for serial interfaces (like UART / RS232)). This protocol is applied on top of the USB CDC device class protocol.

5.2.1 Frame Structure

The frame structure configuration for the TinyFrame library is as follows.

	HEAD					DATA	
	START	ID	LENGTH	TYPE	HEAD-CKSUM	PAYLOAD	DATA-CHKSUM
SIZE (BYTE)	1	1	2	1	2	0~	2
VALUE	0xAA		sizeof(PAYLOAD)		crc16((START,ID,LENGTH,TYPE))		crc16(PAYLOAD)

The payload data is transmitted in little-endian format for easier handling on the MCU side. The Type field is used to define the kind of message transmitted in the payload.

5.2.2 Message Types

The protocol is response-based; the majority of the set commands respond with their actual values after executing the set operation. When the payload of a message for a set command is left empty, the MCU responds with the current active values. The following table gives an overview of the messages implemented in the current firmware version 1.0.0. Send Parameters are the parameters transmitted from the PC to the MCU. Response parameters are transmitted from the MCU to the PC as a response to a message from the PC.

Table 4: List of implemented Messages

Command	Type	Description	Parameters	
			Send	Response
SYS_INFO	0x01	Get System Information	None	2 * UINT32
MCU_INFO	0x02	Get MCU Information	None	3 * UINT32
CLK_INFO	0x03	Get Clock Information	None	3 * UINT32
TASK_STAT	0x04	Get Task Statistics	None	STRING
TASK_LIST	0x05	Get Task List	None	STRING
LED_SIG	0x06	Set User LED signals	6 * UINT8	6 * UINT8
CHT_INFO	0x10	Get Cheetah Information	None	2 * UINT16
CHT_EXPEN	0x18	Enable Cheetah Exposure	UINT8	None
CHT_FPS	0x19	Set Cheetah Frame Rate	2 * FLOAT32	2 * FLOAT32
CHT_EXP	0x1A	Set Cheetah Exposure Time	FLOAT32	FLOAT32
CHT_IRED	0x1B	Set IR Light state	2 *UINT8	None
STR_FPS	0x21	Set Stream Frame Rate	FLOAT32	FLOAT32
CPT_FRCNT	0x22	Get the number of captured frames	None	UINT32
CPT_FPS	0x23	Get capture frame rate	None	FLOAT32
LDM_EN	0x30	Enable Face and Landmark detection	UINT8	UINT8
LDM_FPS	0x31	Set Landmark detection Frame Rate	FLOAT32	FLOAT32
LDM_NUM	0x32	Get Number of Detections	None	UINT32
LDM_SDET	0x33	Get single Detection	UINT32	14 * FLOAT32
LDM_ADET	0x34	Get all Detections	UINT8	X * FLOAT32
LDM_PAR	0x35	Set parameters for Landmark detection	UINT8, FLOAT32	UINT8, FLOAT32
LDM_RT	0x38	Get Face Landmark detection runtime	None	FLOAT32
LDM_MEM	0x39	Get Landmark model memory size	None	UINT32
LDM_VER	0x3A	Get Landmark model version	None	UINT32
AEC_EN	0x40	Enable Auto Exposure Control	UINT8	UINT8

AEC_FPS	0x41	Set Auto Exposure Control Frame Rate	FLOAT32	FLOAT32
AEC_PAR	0x42	Set AEC parameter	7 * FLOAT32	7 * FLOAT32
AEC_BRGHT	0x43	Get Image Brightness	None	FLOAT32
AEC_RT	0x48	Get AEC runtime	None	FLOAT32

5.2.3 Description of the Messages

5.2.3.1 Get System Information

SYS_INFO	Payload		
0x01	Send	Response	
Data type		UINT32	UINT32
Data		Board Revision	Firmware Version

Board Revision and Firmware Version Encoding:

Bits	31	24	23	16	15	8	7	0
Data	Unused		Major Version		Minor Version		Patch	

5.2.3.2 Get MCU Information

MCU_INFO	Payload			
0x02	Send	Response		
Data type		UINT32	UINT32	UINT32
Data		Device Type	Device ID	Chip Revision

See the datasheet of the MCU for the meaning of the values of Device Type, Device ID and Chip Revision.

5.2.3.3 Get Clock Information

CLK_INFO	Payload			
0x03	Send	Response		
Data type		UINT32	UINT32	UINT32
Data		Core Clock	FlexIO Clock	Exposure Timer Clock

5.2.3.4 Get Task Statistics

TASK_STAT	Payload	
0x04	Send	Response
Data type		STRING
Data		See: Run Time Statistics - FreeRTOS™

5.2.3.5 Get Task List

TASK_LIST	Payload	
0x05	Send	Response
Data type		STRING
Data		See: Task Utilities - FreeRTOS™

5.2.3.6 Set User LED signals

LED_SIG	Payload					
0x06	Send					
Data type	UINT8	UINT8	UINT8	UINT8	UINT8	UINT8
Data	LD0 config	LD1 config	LD2 config	LD3 config	LD4 config	LD5 config
	Response					
Data type	UINT8	UINT8	UINT8	UINT8	UINT8	
Data	LD0 config	LD1 config	LD2 config	LD3 config	LD4 config	LD5 config

Table 5: LED configuration options

	LDx config	Signal type
Off	0	state
Startup	1	state
Heart beat	2	state
Firmware error	3	state
AIS Protocol Message received	4	event
Cheetah ASIC detected at Startup	5	state
Cheetah detection error	6	state
Cheetah configured successfully	7	state
Frame captured from Cheetah	8	event
Frame is ready to stream	9	event
Frame streamed	10	event
Landmark detection is processing a frame	11	event
Face and Landmark detected	12	event
Landmark model initialization failed	13	state
AEC is active controlling the exposure	14	event
AEC is processing a frame	15	event
AEC is limited to zero exposure time	16	event

In the current firmware version (1.0.0), state signals are only partially supported, state changes are not stored internally. This means the LED needs to be configured for a state signal before the state change appears, to be visible on the LED.

5.2.3.7 Get Cheetah Information

CHT_INFO	Payload		
0x10	Send	Response	
Data type		UINT16	UINT16
Data		Revision	Product ID

See the datasheet of the Cheetah ASIC for the values of Revision and Product ID.

5.2.3.8 Enable Cheetah Exposure

CHT_EXPEN	Payload	
0x18	Send	Response
Data type	UINT8	
Data	Exposure State	

Exposure State = 0 → The exposure timer is switched off.
Exposure State > 0 → the exposure timer is switched on.

5.2.3.9 Set Cheetah Frame Rate

CHT_FPS	Payload			
0x19	Send		Response	
Data type	FLOAT32	FLOAT32	FLOAT32	FLOAT32
Data	Frame rate in fps	Exposure in seconds	Actual Frame rate in fps	Actual Exposure in seconds

This command changes the capture frame rate, thereby temporarily disabling the exposure timer while the frame rate is updated. It also updates the exposure time simultaneously. The response is the actual set frame rate due to the resolution of the exposure timer and the actual exposure time, which could be limited due to the frame rate. The minimum exposure time is 1 µs. The frame rate imposes a limit on the maximum exposure time of

$$exposure\ time_{max} = \frac{1}{capture\ frame\ rate} - 1\mu s$$

5.2.3.10 Set Cheetah Exposure

CHT_EXP	Payload	
0x1A	Send	Response
Data type	FLOAT32	FLOAT32
Data	Exposure time in seconds	Actual Exposure time in seconds

5.2.3.11 Set Infrared Light state

CHT_EXPEN	Payload		
0x18	Send		Response
Data type	UINT8	UINT8	
Data	Top row IR light state	Bottom row IR light state	

IR light state = 0 → Infrared Light is OFF
IR light state > 0 → Infrared Light is ON

5.2.3.12 Set Stream Frame Rate

STR_FPS	Payload	
0x21	Send	Response
Data type	FLOAT32	FLOAT32
Data	Stream frame rate in fps	Actual stream frame rate in fps

5.2.3.13 Get the number of captured frames

CPT_FRCNT	Payload	
0x22	Send	Response
Data type		UINT32
Data		Number of frames captured since startup

5.2.3.14 Enable Landmark Detection

LDM_EN	Payload	
0x30	Send	Response
Data type	UINT8	UINT8
Data	Landmark Detection state	Current Landmark Detection state

Landmark Detection state = 0 → Landmark Detection is OFF
Landmark Detection state > 0 → Landmark Detection is ON

5.2.3.15 Set Landmark Detection Frame Rate

LDM_FPS	Payload	
0x31	Send	Response
Data type	FLOAT32	FLOAT32
Data	Landmark detection frame rate in fps	Actual landmark detection frame rate in fps

5.2.3.16 Get Number of Detections

LDM_NUM	Payload	
0x32	Send	Response
Data type		UINT32
Data		Number of detected Faces and Landmarks in the last inference run

5.2.3.17 Get single Detection

LDM_SDET	Payload													
0x33	Send													
Data type	UINT32													
Data	Index of the detected landmark requested													
	Response													
Data type	10 * FLOAT32										4 * FLOAT32			
	Keypoints										Bounding box			
Data	k0x	k0y	k1x	k1y	k2x	k2y	k3x	k3y	k4x	k4y	tlx	tly	brx	bry

5.2.3.18 Get all Detection

LDM_ADET	Payload													
0x34	Send													
Data type	UINT8													
Data	Enable the automatic transmission of all landmarks													
	Response													
	Amount of detected landmarks times:													
Data type	10 * FLOAT32										4 * FLOAT32			
	Keypoints										Bounding box			
Data	k0x	k0y	k1x	k1y	k2x	k2y	k3x	k3y	k4x	k4y	tlx	tly	brx	bry

This command is an exception to the rule that the MCU responds to every get command with the current values. In terms that as long as enabled, the results are automatically transmitted when the post processing is finished, and there is at least one face detected. The number of detected faces is always transmitted, regardless of whether a face was detected or not. This means the number of detections is transmitted with the same frame rate as set for landmark detection.

5.2.3.19 Set Parameters for Landmark Detection

LDM_PAR	Payload			
0x35	Send		Response	
Data type	UINT8	FLOAT32	UINT8	FLOAT32
Data	Confidence Threshold	NMS Threshold	Confidence Threshold	NMS Threshold

5.2.3.20 Get Landmark Detection Runtime

LDM_RT	Payload	
0x38	Send	Response
Data type		FLOAT32
Data		Runtime of the Landmark Detection Algorithm in seconds

5.2.3.21 Get Landmark model memory size

LDM_MEM	Payload	
0x39	Send	Response
Data type		UINT32
Data		Tensor arena memory size in bytes

5.2.3.22 Get Landmark model schema version

LDM_VER	Payload	
0x3A	Send	Response
Data type		UINT32
Data		Landmark model schema version

5.2.3.23 Enable Auto Exposure Controller

AEC_EN	Payload	
0x40	Send	Response
Data type	UINT8	UINT8
Data	Auto Exposure Controller state	Current Auto Exposure Controller state

Auto Exposure Controller state = 0 → Auto Exposure Controller is OFF
Auto Exposure Controller state > 0 → Auto Exposure Controller is ON

5.2.3.24 Set Auto Exposure Controller Frame Rate

AEC_FPS	Payload	
0x41	Send	Response
Data type	FLOAT32	FLOAT32
Data	Auto Exposure Controller frame rate in fps	Actual Auto Exposure Controller frame rate in fps

5.2.3.25 Set AEC parameter

AEC_PAR	Payload						
0x42	Send						
Data type	FLOAT32	FLOAT32	FLOAT32	FLOAT32	FLOAT32	FLOAT32	FLOAT32
Data	exp_k	exp_k_reduced	filt_k	n_min	n_max	w_min	w_max
	Response						
Data type	FLOAT32	FLOAT32	FLOAT32	FLOAT32	FLOAT32	FLOAT32	FLOAT32
Data	exp_k	exp_k_reduced	filt_k	n_min	n_max	w_min	w_max

Table 6: AEC parameter description

AEC parameter		Unit	default value
exp_k	AEC Gain	seconds/Brightness	1e-6
exp_k_reduced	Reduced AEC Gain when the controller clips at zero exposure	seconds/Brightness	5e-8
filt_k	Brightness filter coefficient	one	0.5
n_min	minimum target brightness	Brightness	90
n_max	maximum target brightness	Brightness	110
w_min	minimum tolerated brightness	Brightness	70
w_max	maximum tolerated brightness	Brightness	130

5.2.3.26 Get Image Brightness

AEC_BRGHT	Payload	
0x43	Send	Response
Data type		FLOAT32
Data		Current brightness of the last frame processed by the AEC

$$Brightness = \frac{\sum pixel}{\#pixel}$$

5.2.3.27 Get AEC Runtime

LDM_RT	Payload	
0x48	Send	Response
Data type		FLOAT32
Data		Runtime of the Auto Exposure Controller in seconds

5.3 Core Tasks

The Core Tasks provide services that are required by most of the application tasks.

5.3.1 Cheetah Control

Configures the Cheetah ASIC after startup as required for image capturing and verifies if communication via SPI is possible. Generates the exposure signals for the Cheetah ASIC via a PWM peripheral module. Handles all the messages starting with 0x1_ regarding the Cheetah frame rate and exposure time. Further takes care of the infrared light.

5.3.2 Cheetah Capture

Manages the frame buffer and distributes the frames to the subscribed application tasks. Takes care of the frame rate reduction for the subscribed application tasks. Distributing the frames is based on a publisher-subscriber pattern. Application tasks that want to receive and process frames register as receivers at the Cheetah Capture Task with their desired receive frame rate at their start-up. After registering the receiving application tasks blocking wait in their main loop for new frames.

5.3.3 AIS Protocol

This task is responsible for receiving data from the USB virtual COM interface 1 and passing it to the TinyFrame library for decoding. Then the decoded messages are passed to the subscribed application task. The distribution of messages is also based on a publisher-subscriber pattern. Tasks that require messages from a certain type or a range of types register at startup as a message receiver for the desired type or type range. The messages are distributed via FreeRTOS message queues that are generated by the receiving application task. Therefore, the application task needs to make sure the message queue is large enough for all the expected messages. Decoding the message payload is the sole responsibility of the receiving task.

5.4 Application Tasks

The following applications are implemented in the current firmware (1.0.1).

5.4.1 Cheetah Stream

Streams the frames to the PC via the USB virtual COM interface 0. Implements a straightforward protocol for streaming frames, where frames are transmitted only after the PC indicates readiness to receive them. The PC sends a 16-bit magic number as a ready signal whenever it is prepared to receive a new frame. Handles all messages regarding streaming and the streaming frame rate, and some state information about the cheetah capture task.

5.4.2 Biometric Keypoint Tracking

Utilizes the NPU in the MCU to run a Biometric Keypoint Tracking model on the received frame. Performs preprocessing of the image to transform the frame into the shape required by the AI model. Runs the model inference and performs the post-processing on the results. When enabled, the post-processed results of the Face and Biometric Keypoint Detection model are automatically transmitted to the PC. Further, all messages regarding the Detection are handled in this task.

5.4.3 Auto Exposure Control

Controls the exposure time to keep the image brightness in a desired range. The control algorithm implements a hysteresis to determine when the controller is engaged and disengaged. Meaning, when the brightness is outside the wide brightness range, the controller is enabled; as soon as the brightness falls inside the narrow brightness range, the controller is disabled. Additionally, all the messages regarding the AEC are handled, which allows the configuration of the AEC as desired.

6 List of Abbreviations

Table 7: List of Abbreviations

Name	Description
AEC	Auto Exposure Control
AFOV	Angle Field Of View
EFL	Effective Focal Length
GPIO	General Purpose Input Output
IRED	Infrared Light Emitting Diode
ISP	In System Programming
LED	Light Emitting Diode
MCU	Microcontroller Unit
NPU	Neural Processing Unit
PMU	Power Management Unit
SPI	Serial Peripheral Interface
TBA	To Be Added

7 Revision History

Table 8: Revision History

Revision	Date	Description	Author
0.1	2025-04-10	Initial Version	Martin Jungwirth, Verena Fink
1.0	2025-05-28	Document release	Maximilian Heindel

8 List of Figures

9 List of Tables

10 Important Notice and Disclaimer

AIStorm provides technical and reliability data, design resources, application advice, web tools, safety information, and other resources in their original state, with all faults. AIStorm explicitly disclaims all warranties, both express and implied, including but not limited to warranties of merchantability, fitness for a particular purpose, and non-infringement of third-party intellectual property rights.

These resources are intended for skilled developers who are designing applications using AIStorm products. It is your sole responsibility to:

1. Select the appropriate AIStorm products for your application.
2. Design, validate, and test your application.
3. Ensure that your application complies with applicable standards and meets all safety, security, regulatory, and other requirements.

Please note that these resources are subject to change without notice. AIStorm grants you permission to use these resources solely for the development of an application that utilizes AIStorm products as described in the resource. Reproduction and display of these resources for any other purposes are strictly prohibited. No license is granted for any other AIStorm intellectual property right or any third-party intellectual property right. AIStorm disclaims any responsibility, and you agree to indemnify AIStorm and its representatives fully, for any claims, damages, costs, losses, and liabilities arising from your use of these resources.

AIStorm's products are provided under AIStorm's Terms of Sale or other applicable terms, available on aistorm.ai or provided in conjunction with the respective AIStorm products. The provision of these resources by AIStorm does not modify or expand AIStorm's applicable warranties or warranty disclaimers for AIStorm products.

AIStorm explicitly rejects any additional or different terms proposed by you.