Cheetah Connect PCB

General Description

Cheetah Connect PCB is a complete image acquisition and processing system consisting of Cheetah IC and Cheetah Receiver FPGA.

Scope of the document

This document provides instructions on installing and using the Cheetah Connect PCB. A detailed description of the installation process is given, as well as a description of the Cheetah Receiver FPGA system and its operation. The document provides the information in the following order:

1. How to use the Cheetah Connect PCB
2. Environment Installation (for Windows OS)
3. Cheetah Receiver FPGA description
4. Application code structure
5. Example of image acquisition

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
- Spartan 7 FPGA for image readout and processing
    - on-chip 1620 kbit of SRAM memory
    - 60 MHz system clock for processing
    - scalable architecture for arbitrary image algorithms
- Onboard 128MByte PSRAM for image recording
- USB-C user interface for PC connection

Applications

- Gamer mouse frontend
- High-speed tracking
- High-speed inspection

Subject to change without notice.

Table of Contents

List of Figures

List of Tables

List of Listings

1 Getting Started

To set up the system for Cheetah Connect PCB, the following hardware and software is required:

- Cheetah Connect PCB (see Figure 1-1)
- USB-C cable to connect the host and Cheetah Connect PCB
- Python version 3.10

1.1 Hardware Overview

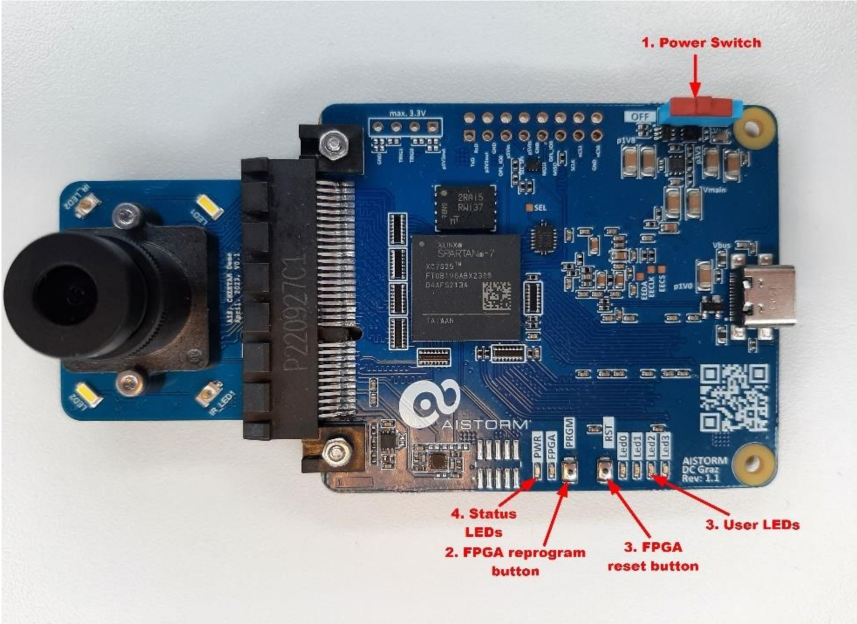


Figure 1-1: Picture of the Cheetah Connect PCB

1. Power Switch: turns the board on or off

2. FPGA Reprogram Button: when pressed, the FPGA is reprogrammed from the FLASH

3. FPGA Reset Button: when pressed, the reset signal to the FPGA is asserted

4. Status LEDs: indicate the status of the device
   - "PWR" LED indicates that all power rails are active and in their required range
   - "FPGA" indicates the FPGA has successfully booted from the configuration flash

5. User LEDs: FPGA-driven LEDs, in the current configuration indicate the following:
   - "LED1": record mode is active
   - "LED2": replay mode is active
   - "LED3": frame stream active
   - "LED4": frame readout ready

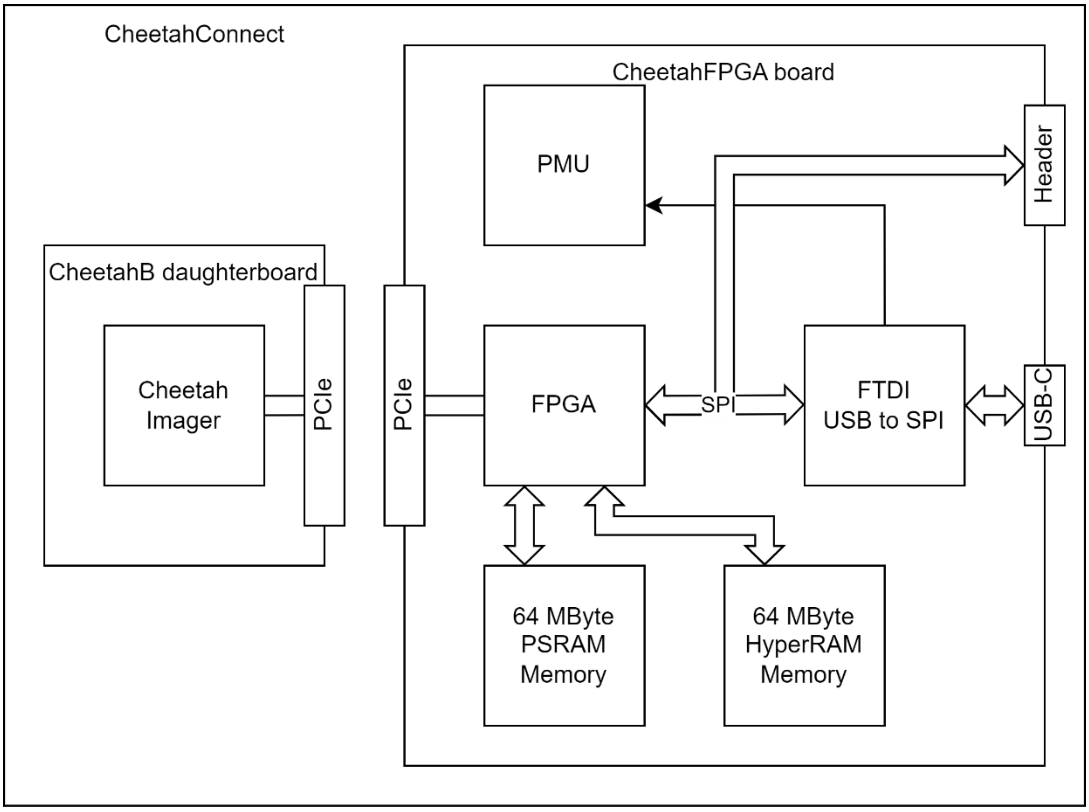


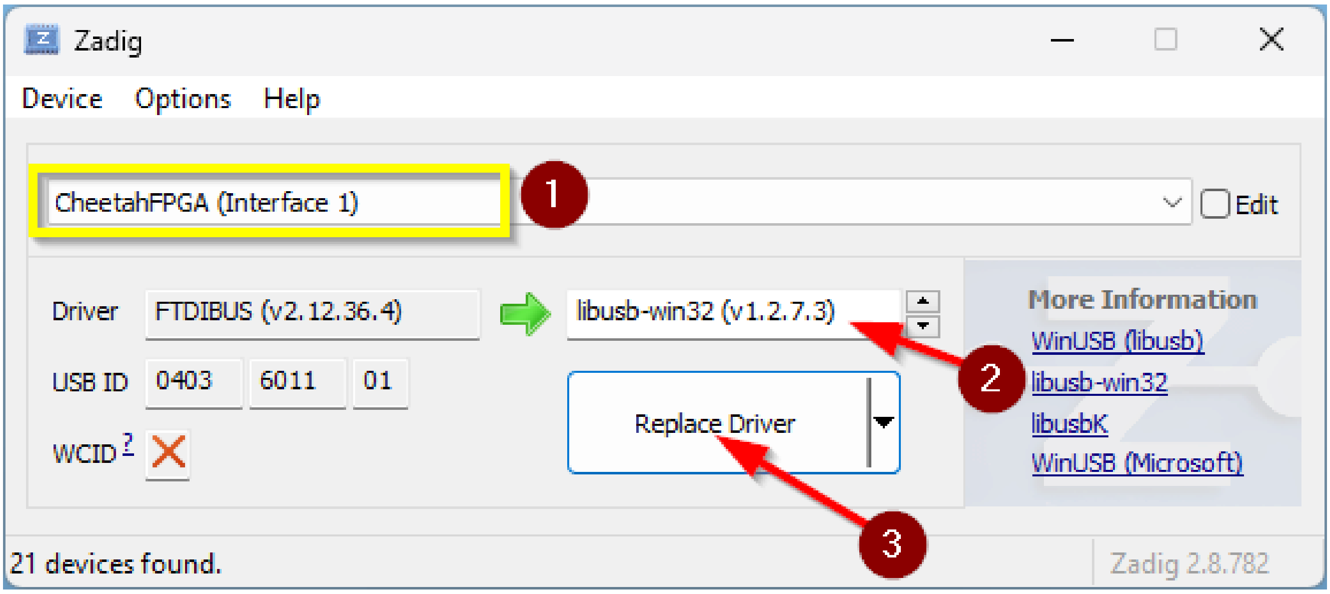
Figure 1-2: Cheetah Connect PCB hardware block diagram

2 Environment Installation

This chapter provides instructions on how to install the required software tools.
Please note that while the Cheetah Connect PCB system can run in multiple OS hosts, the instructions are provided only for the Windows operating system.

2.1 Installing the Windows USB driver

1. Connect the board to your computer with the appropriate cable and power it up
2. Run the zadig executable
3. Choose Options→ List All Devices
4. Select “CheetahFPGA (Interface 1)” (tag 1 in the image below)
5. Select libusb-win32 in the Driver drop-down list (tag 2 in the image below)
6. Click “Replace Driver” (tag 3 in the image below)
7. Close the zadig tool



CheetahFPGA (Interface 1)

Driver FTDIBUS (v2.12.36.4) → libusb-win32 (v1.2.7.3)

USB ID 0403 6011 01

WCID ☒

21 devices found.

Figure 2-1: zadig tool screenshot with marks.

2.2 Verify the driver installation, (optional)

1. Open “Device Manager”
2. There should be a “libusb-win32 devices” category
3. The category should contain at least one device: “CheetahFPGA (interface 1)”

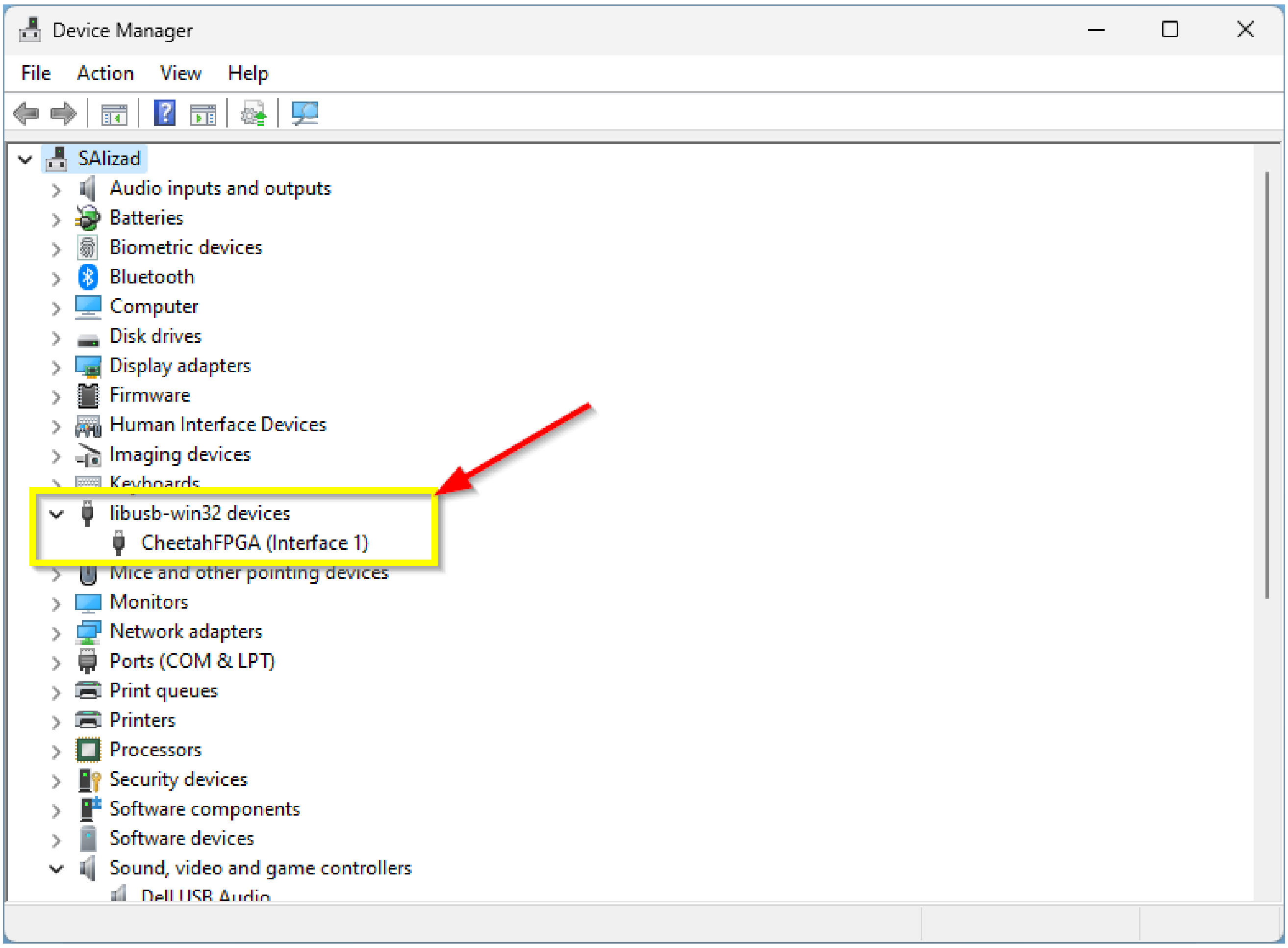
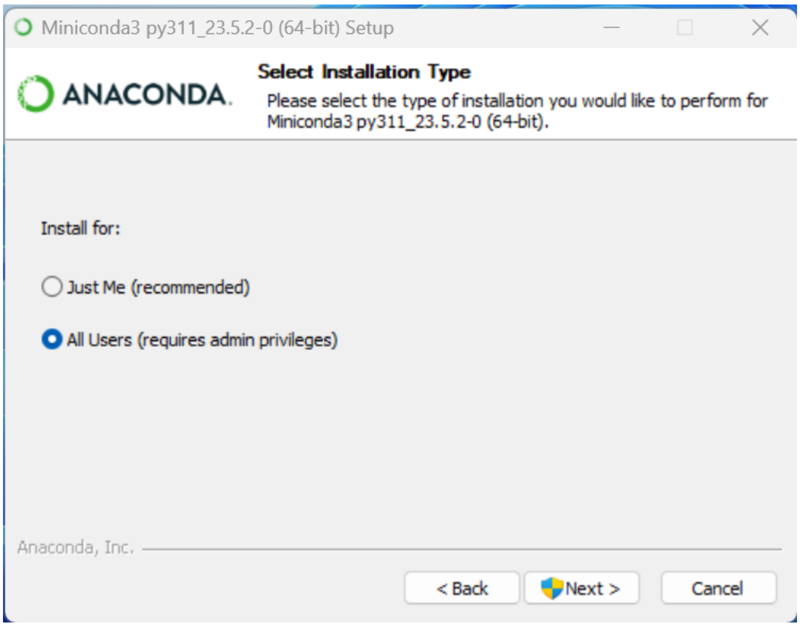


Figure 2-2: Driver installation verification in the Device Manager

2.3 Install the Conda Environment

1. If you don't have the Conda environment already installed, download the executable from the following location:
   https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe

2. Install Miniconda with the settings shown in the images below



Miniconda3 py311_23.5.2-0 (64-bit) Setup

Select Installation Type

Please select the type of installation you would like to perform for Miniconda3 py311_23.5.2-0 (64-bit).

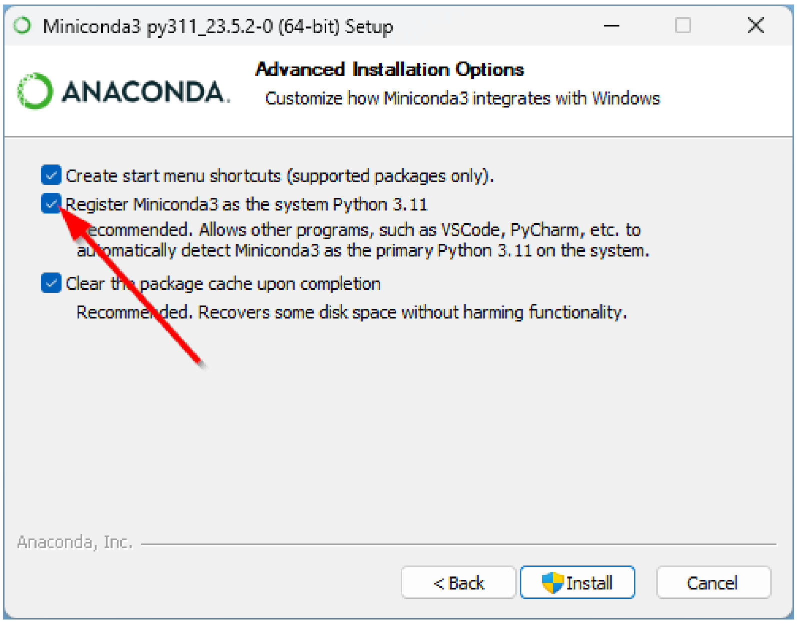
Install for:

○ Just Me (recommended)

● All Users (requires admin privileges)

Anaconda, Inc.

Figure 2-3. Install Miniconda for Users.



Miniconda3 py311_23.5.2-0 (64-bit) Setup

Advanced Installation Options

Customize how Miniconda3 integrates with Windows

☑ Create start menu shortcuts (supported packages only).

☑ Register Miniconda3 as the system Python 3.11
Recommended. Allows other programs, such as VSCode, PyCharm, etc. to automatically detect Miniconda3 as the primary Python 3.11 on the system.

☑ Clear the package cache upon completion
Recommended. Recovers some disk space without harming functionality.

Anaconda, Inc.

Figure 2-4: Register Miniconda3 as system Python.

2.4 Install Application Software

1. Download the Cheetah Connect package from the following location:
   git hub link will be provided on request

2. Unzip the file and go to the folder

3. Open the “_Setup” folder and run the “_install.bat” to install the required Python packages

4. To start the application, run the file “_run.bat”

3 Cheetah Receiver FPGA

3.1 Overview

The following digital circuit implemented in the FPGA on the Cheetah Connect PCB is a custom design to facilitate capturing, storing in onboard RAM, and readout of frames from our Cheetah Imager ASIC. It is intended to evaluate the capabilities of the Cheetah Imager ASIC. Therefore, the data handled inside the FPGA is divided into frames as the smallest logical units.

3.1.1 Features

- Record frames at high frame rates into the onboard RAM
- Live stream capability during record with a reduced frame rate
- Replay the stored frames at a low frame rate for readout
- Control the LEDs and IR LEDs state and intensity

3.1.2 Block Diagram

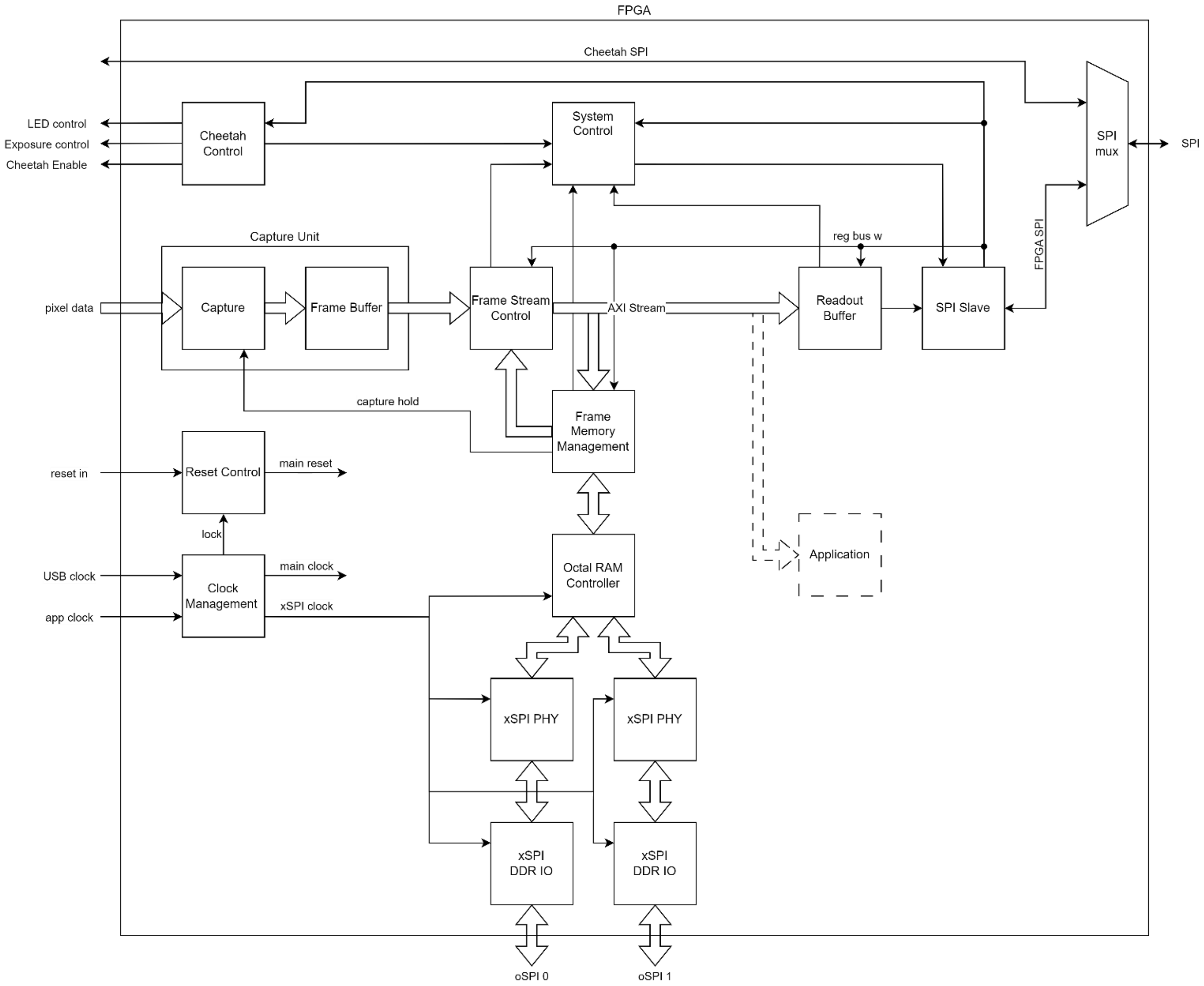


Figure 3-1: Block diagram for the Cheetah Receiver

3.1.3 Capture Unit

The capture unit samples the pixel data from the cheetah ASIC synchronized to the pixel clock. The sampled pixels are packed into 16-pixel packets and placed into an internal frame buffer for further processing.

3.1.4 Frame Stream Control Unit

The frame stream control unit acts like a switch between the frames from the Cheetah ASIC and those from the onboard memory. In stream mode, it will repeatedly check if frames are in the internal frame buffer of the capture unit. When there is a frame in the buffer it will output the frame as AXI Stream. In replay mode, it will periodically initiate a read of one frame from the onboard memory with the set frame rate.

3.1.5 Frame Memory Management Unit

The frame memory management unit manages the content of the four onboard PSRAM chips. Read and Write operations are organized as bursts with the size of one frame with the lowest resolution, which means one burst has a size of 1200 bytes. Therefore, it generates the addresses required for each burst read and write operation.
Linear address spaces

3.1.6 Octal RAM Controller

The Octal RAM controller translates the AXI read and write requests into a command and address transaction for the Octal RAM. The linear address from the Frame Memory management unit is mapped into row address and column address. The AXI input data is split into data words for the two xSPI PHY units.

3.1.7 xSPI PHY Unit

The xSPI PHY unit generates the signals for the xSPI lanes required by the protocol for the onboard PSRAM. It takes care of the required timing, the command, row, and column address cycles and the data transmission cycles.

3.1.8 Readout Buffer

The Readout buffer is used to buffer up to 5 frames for transmission over SPI to the onboard FTDI chip or other SPI master. A flag in the status register is provided to signal to an SPI master that frames are waiting in the buffer for readout.

3.1.9 SPI Slave

The SPI Slaves receive the commands and data sent by an external SPI master and translate it to a 16-bit data and address bus according to the implemented communication protocol. This 16-bit data and address bus is used to access all the control and status registers of all the circuit blocks.

3.1.10 Cheetah Control

The Cheetah control unit is responsible for generating the control signals like enabling signals for the Cheetah ASIC control inputs. Four PWM channels with settable pulse counts are available, which can be used for exposure impulse generation or controlling the intensity of the LEDs on the daughterboard. In the current configuration, PWM channel 0 is used for exposure control.

3.1.11 Reset Control

The Reset control unit generates the main reset signal for all the circuit blocks after powering up or when an external reset signal is applied.

3.1.12 System Control

The system control unit holds the read-only register for the board version and the current FPGA configuration (gateware).

3.2 Operation

The Cheetah receiver circuit in the FPGA has two main operating modes. The first one is the record mode to store the frames coming from Cheetah ASIC in the onboard RAM chips. The second is the replay mode to read the previously recorded frames from the onboard RAM with a different frame rate then the where recorded, usually a much lower rate for readout via SPI.

3.2.1 Record Mode

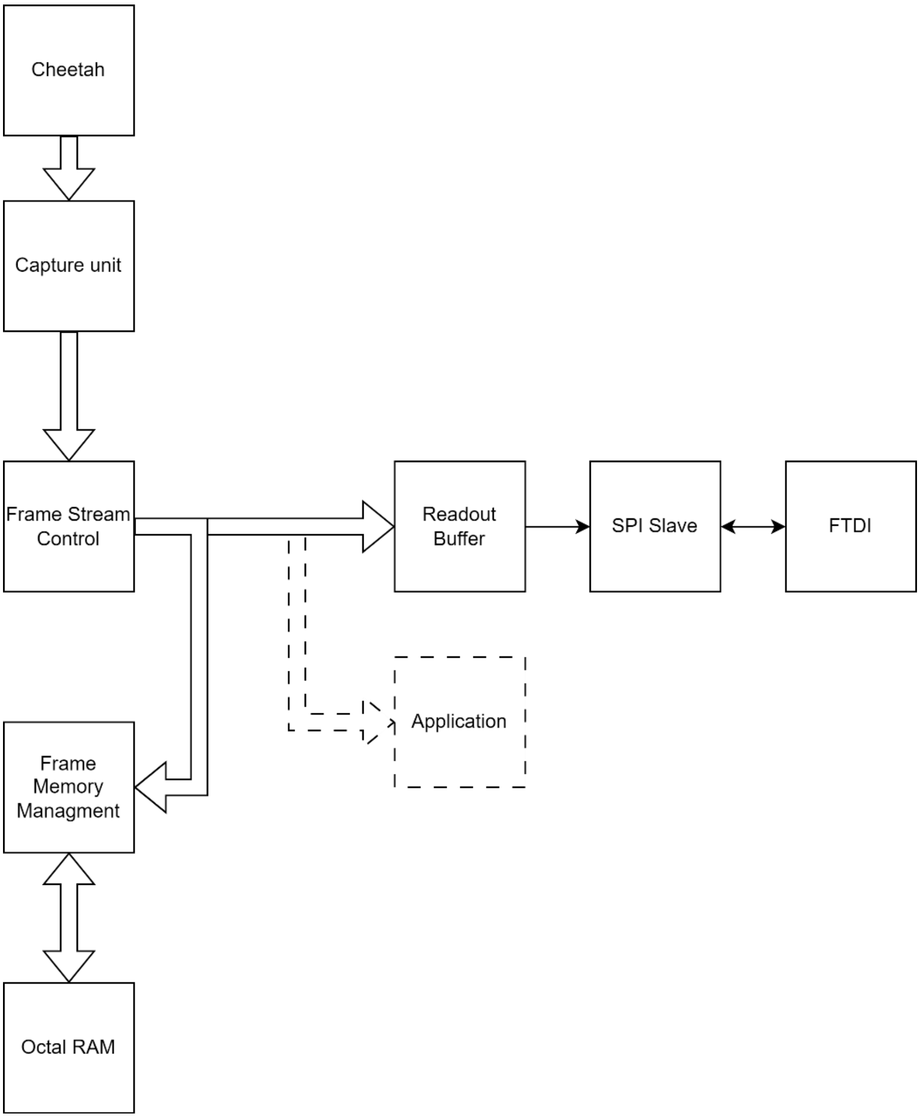


Figure 3-2: Dataflow in Record Mode

The Record mode is used to store frames captured by the Cheetah ASIC at high frame rates e.g., 10,000 fps. The Frame Stream Control unit is set into stream mode, to read frames from the frame buffer inside the Capture unit and stream them as an AXI4 stream output port. Adding a frame start and stop signal. Additionally, the Frame Stream Control unit can provide a second frame start signal at every x frame, where x can be configured via register. Thereby providing a stream with a reduced frame rate. This additional frame start signal is used by the readout buffer for a live stream readout during recording.

The Frame Memory Management unit is set into record mode acting as a sink for the frame stream from the Frame Stream Control unit and putting the frames into the onboard RAM sequentially. The number of frames stored in the onboard RAM is settable via register.

3.2.2 Replay Mode

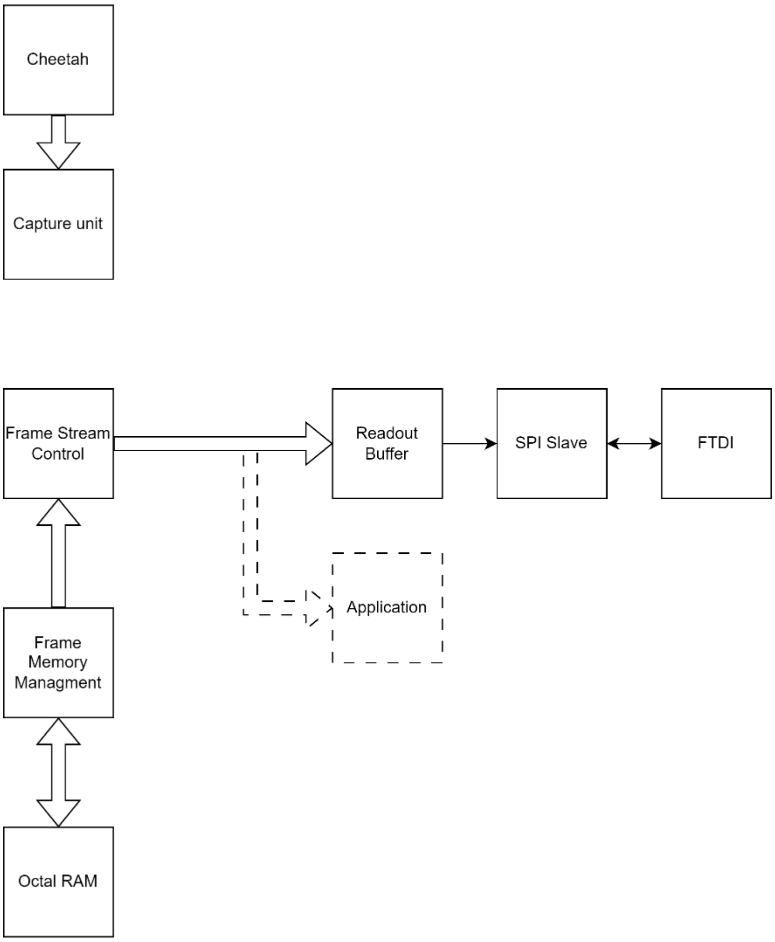


Figure 3-3: Dataflow in replay mode

In replay mode, the Frame Memory Management unit takes the place of Cheetah ASIC and streams the previously recorded frames from the onboard RAM. Therefore, the Frame Stream Control unit is set into replay mode acting like a switch and passing the frames from the Frame Memory Management unit to its AXI4 stream output port. The Frame Stream Control unit controls the rate at which frames are read and streamed, adjustable by register. The number of frames replayed is configured in the Frame Memory Management unit.

3.3 Register Map

Address	Name	Reset	Description
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]

3.4 System

3.4.1 SYS_STATUS

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.4.2 BOARD

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.4.3 GATEWARE

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.4.4 FRAME_RES

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.4.5 FRAMES_TOT_L

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.4.6 FRAMES_TOT_H

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5 Cheetah Control

3.5.1 CH_STATUS

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5.2 CH_CTRL

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5.3 CH_GPIO

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5.4 CH_PWM_CTRL

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5.5 CH_PWM0_PER_L

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5.6 CH_PWM0_PER_H

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

PWM channel 0 is used for exposure time and frame rate (FR) generation, the frame rate can be set as follows:

$$FR_{capture} = \frac{f_{SYS\_CLOCK}}{(\mathrm{CH\_PWM0\_PER\_H} \cdot 2^{16} + \mathrm{CH\_PWM0\_PER\_L} + 1)}$$

3.5.7 CH_PWM0_CMP_L

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.5.8 CH_PWM0_CMP_H

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

The exposure time can be set as follows, care must be taken that care must be taken that exposure time is less than the reciprocal $FR_{capture}$:

$$t_{exposure} = \frac{(\mathrm{CH\_PWM0\_CMP\_H} \cdot 2^{16} + \mathrm{CH\_PWM0\_CMP\_L} + 1)}{f_{SYS\_CLOCK}}$$

3.5.9 CH_PWM0_PULSES

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.6 Frame Stream Control

3.6.1 FSTREAM_STATUS

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.6.2 FSTREAM_CTRL

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Stream mode has preference over Replay mode.

3.6.3 FSTREAM_RPLY_FRPER_L

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.6.4 FSTREAM_RPLY_FRPER_H

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

$$FR_{replay} = \frac{f_{SYS\_CLOCK}}{(\mathrm{CH\_PWM0\_CMP\_H} \cdot 2^{16} + \mathrm{CH\_PWM0\_CMP\_L} + 1)}$$

3.7 Frame Memory Management

3.7.1 FMEM_STATUS

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.7.2 FMEM_CTRL

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Record mode has preference over Replay mode.

3.7.3 FMEM_FRM_CNT_L

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.7.4 FMEM_FRM_CNT_H

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.7.5 FMEM_FRM_NR_L

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.7.6 FMEM_FRM_NR_H

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.8 Readout Buffer

3.8.1 RDOUT_STATUS

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

3.8.2 RDOUT_CTRL

Bit	Type	Reset	Field	Description
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

4 Application Code Structure

The following figure provides an overview of the test and evaluation software modules written in Python and their interaction.

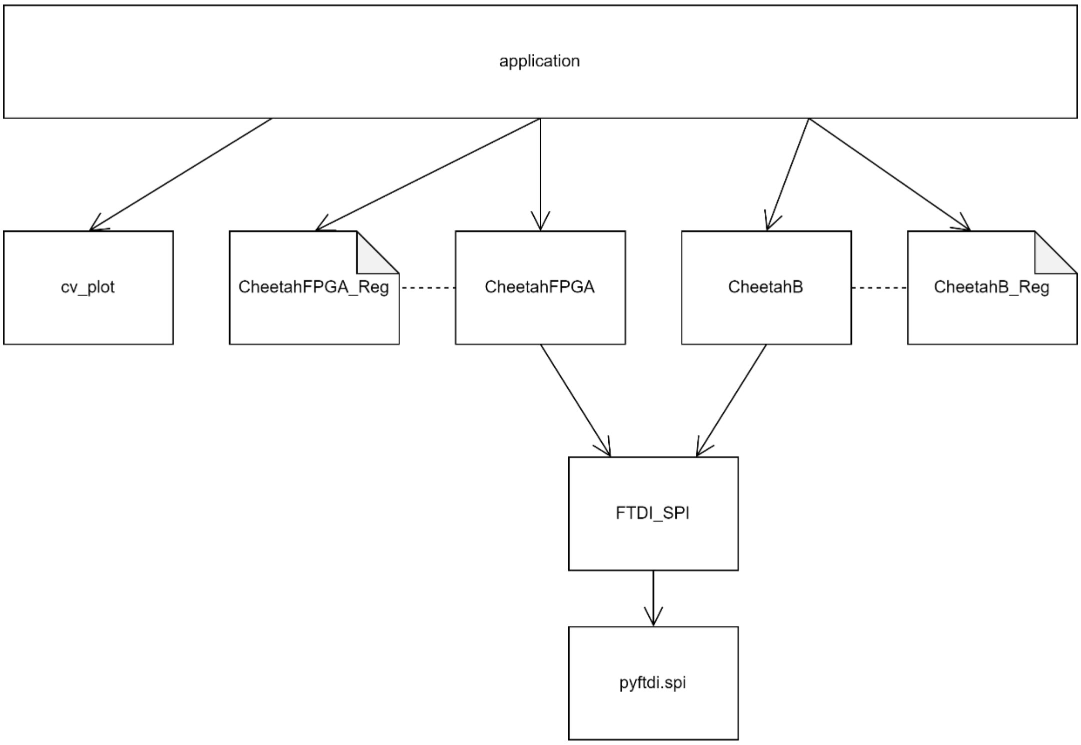


Figure 4-1: Structure of the provided Python code

The provided software modules use the library pyftdi for SPI communication with the Cheetah Receiver FPGA. The module FTDI_SPI.py configures pyftdi in the required mode and selects the right USB port. The modules CheetahFPGA.py and CheetahB.py encapsulate the required register settings in class methods.

4.1 Example Usage

4.1.1 Setup the Cheetah Receiver

[redacted]

[redacted]

[redacted]

[redacted]

Listing 4-1: Setup script for Cheetah Receiver FPGA

4.1.2 Record Mode

[redacted]

[redacted]

[redacted]

Listing 4-2: Minimal code to record frames into onboard RAM

4.1.3 Replay Mode

[redacted]

[redacted]

Listing 4-3: Basic script to replay Frames from onboard RAM

5 List of Abbreviations

Table 5-1: List of Abbreviations

Name	Description
TBA	To Be Added
FR	Frame Rate
GPIO	General Purpose Input Output
SPI	Serial Peripheral Interface

6 Revision History

Table 6-1: Revision History

Revision	Date	Description	Author
0.1	2023-10-27	Initial revision	Martin Jungwirth

7 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.