Video V-PIPE Pipeline for Computer Vision

sddec24-06

Advisor: Dr. Jones

Client: JR Spidell

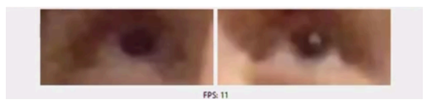
Team: Liam Janda, Taylor Johnson
Ritwesh Kumar, Deniz Tazegul

Problem Statement

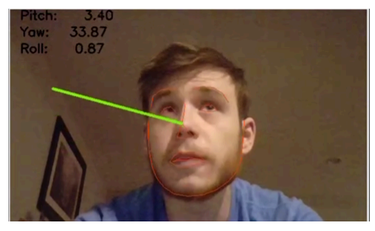
The goal of this project is to improve the navigational abilities of individuals restricted to motorized wheelchairs by creating a video pipeline capable of high-speed video transmission. This video pipeline will be able to take in data from a camera facing the user and run a machine learning algorithm that will be created by another design team to track the orientation and movement of the eyes.

Users

The primary users for this project will be persons confined to motorized wheelchairs with very limited mobility who require assistance in general locomotion.



Example of camera facing user with eye tracking



Requirements

Functional

- Output video data through DisplayPort monitor
- Linux image
- Ultra96-v2 FPGA board and daughter card
- Sony IMX219QH5-C image sensor

Non-Functional

- Minimum of 640x480p resolution at 15fps

Assumptions

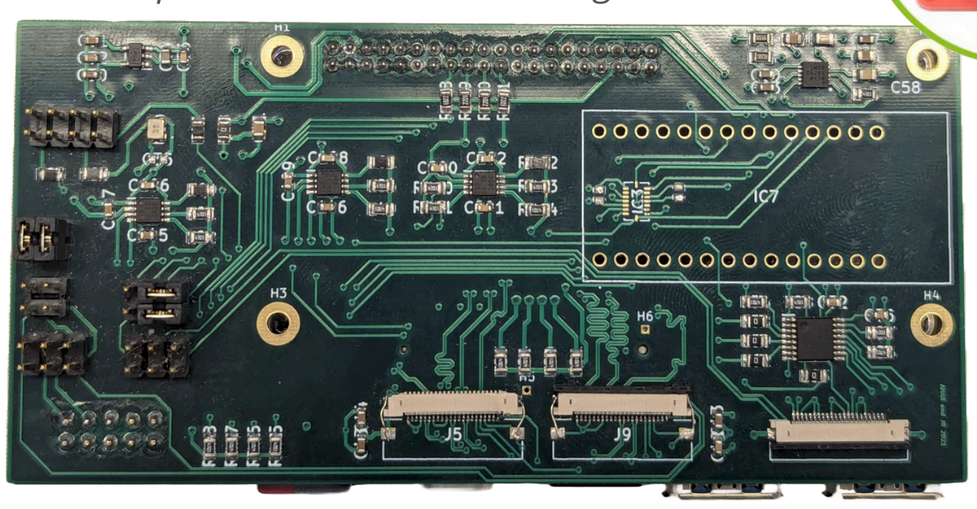
- User's face is well lit by a light source

Standards

- Python and C for initialization and testing
- AXI-4 Lite to control register configurations
- AXI-4 Stream carries data between IP blocks
- MIPI D-PHY physical Communication Layer
- MIPI CSI-2 high speed camera serial interface
- I2C to configure the camera

Ultra96-v2

Top view ultra96-V2 with daughter card



IMX219 (camera)

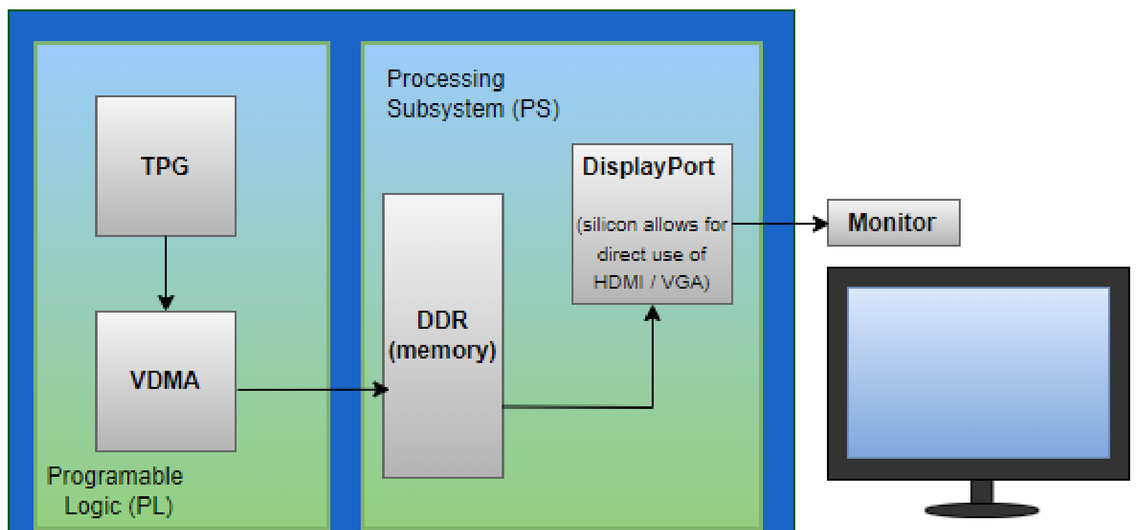
Design

Full video pipeline block diagram

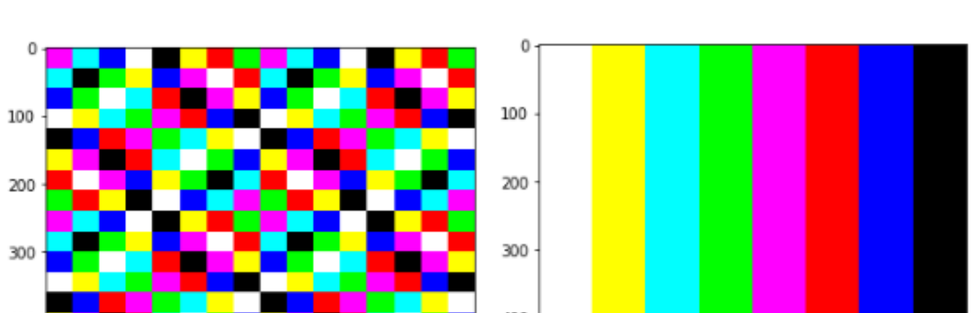
Testing

First pipeline implementation (TPG to VDMA to monitor)

- Configure the TPG and VDMA IP blocks
- Static image from TPG plotted
- Monitor status registers from VDMA
- Visual verification with expected patterns



TPG to VDMA video pipeline block diagram



Results: test patterns from TPG IP block 640x480

Technical Details

Technical Vocabulary

- MIPI - Mobile Industry Processor Interface
- VDMA - Video Direct Memory Access
- TPG - Test Pattern Generator

Full pipeline implementation (IMX219 to monitor)

- Configure IMX219 registers and verify by reading from known register values
- Configure MIPI Controller registers
- Monitor MIPI CSI-2 and D-PHY status registers
- Configure VDMA and monitor VDMA status register
- Visual verification with known output
- Measure frame rate (frames per second or fps)

Software

- Python and C programing
- Jupyter Notebook environment
- PYNQ (python for embedded systems) is one of the main libraries used
- Open CV for video display
- Vivado hardware overlays

Data Format

- Bayer RAW8/RGB color format

Limitation

- Frame rate (by the hardware)

640x480p at 20 fps: test image (left), frame from IMX219 (right)

IOWA STATE UNIVERSITY

REC