CCD Readout Electronics for Prototype Dragonfly Telescope

M.Tech. Thesis

By

SHUBHI TIWARI

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2025

CCD Readout Electronics for Prototype Dragonfly Telescope

A THESIS

submitted in partial fulfillment of the requirements for the award of the degree

of

Master of Technology

by

SHUBHI TIWARI

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2025

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **CCD Readout Electronics for Prototype Dragonfly Telescope**, in partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DEPARTMENT OF MECHANICAL ENGINEERING**, Indian Institute of Technology Indore, is an authentic record of my own work carried out during the time period from **June 2024** to **May 2025**, under the supervision of **Shri Ishant Dave, SOG, Dr. Yogesh Verma, SOG, ALOD, RRCAT Indore and Dr. Manoneeta Chakraborty, Associate Professor, DAASE, IIT Indore.**

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Shubhi Tiwari 22-05-2025

Signature of the student with date
(SHUBHI TIWARI)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

22/05/2025
Signature of the Supervisor of M.Tech. thesis #1 (with date)
(Shri Ishant Dave)

Manoneeta Chakraborty
22/05/2025
Signature of the Supervisor of M.Tech. thesis #2 (with date)
(Dr. Manoneeta Chakraborty)

SHUBHI TIWARI has successfully given his **M.Tech. Oral Examination** held on **09 May 2025.**

Yogesh Verma Manoneeta Chakraborty

Signature(s) of Supervisor(s) of M.Tech. thesis
Date: 25/05/2025

Convener, DPGC
Date:23-05-25

Signature of PSPC Member #1
Date:

Signature of PSPC Member #2
Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors, Shri Ishant Dave, Dr. Yogesh Verma, and Dr. Manoneeta Chakraborty, for their invaluable guidance, encouragement, and consistent support throughout the duration of this project. Their expertise and insights have been instrumental in shaping the direction and outcomes of my research work.

I am also thankful to Dr. Sendhil Raja for his constructive feedback and valuable discussions, which significantly contributed to the development of the project.

I extend my appreciation to the Advanced lasers and Optics Division (ALOD) at RRCAT, Indore, for providing the essential resources and a productive research environment. I am especially grateful to Shri Sivananda Reddy and Shri Sunil Kumar for their continued support and technical guidance during the course of this work. I would also like to acknowledge Shri S. Shyam Sundar for his valuable input and assistance.

I wish to thank the GET-GO collaboration for the opportunity to pursue part of this research at the University of Glasgow and the University of Warwick. I am particularly grateful to Prof. Giles Hammond for facilitating this opportunity. I express my sincere thanks to Dr. Jack Callaghan for his continuous assistance and support during my project at Glasgow. I am also deeply thankful to Dr. Kendall Ackley for her consistent guidance, support, and encouragement.

Finally, I am profoundly grateful to my family for their unconditional love, patience, and unwavering moral support throughout my academic journey.

Abstract

The purpose of this project is to develop readout electronics for an image sensor. This will be used for the development of low cost telescope suitable for detection of kilonova events, in the Dragonfly project. The Dragonfly Project is an initiative aimed at developing a cost-effective, scalable telescope array to detect electromagnetic counterparts of kilonovae, a transient astronomical events linked to neutron star mergers. By deploying an array of 100 low-cost telescopes strategically along the arms of the LIGO-India gravitational wave observatory, the project seeks to enhance sky coverage and improve the probability of capturing faint optical signals associated with gravitational wave events.

To achieve this, two core objectives are pursued: the development of custom FPGA-based readout electronics and the characterization of CMOS image sensors. The readout electronics were prototyped using the Digilent Basys-3 development board with a Xilinx Artix-7 FPGA, incorporating essential digital components and IP cores, and achieving initial success with linear image sensor readout. Simultaneously, CMOS sensors from QHY600, ATIK, and Moravian C1x cameras (based on the Sony IMX455) were analyzed to evaluate key performance metrics such as noise behavior, dynamic range, and photon response uniformity. These efforts collectively lay the groundwork for building a reliable, affordable astronomical imaging system capable of supporting kilonova detection.

Contents

List of Tables

List of Figures

Acronyms

FPGA Field Programmable Gate Array

LIGO Laser Interferometer for Gravitational-wave Observation.

CMOS Complementary Metal Oxide Semiconductor.

CCD Charge-Coupled Device.

IP Intellectual Property Cores.

VHDL Very High-Speed Integrated Circuit Hardware Description Language.

CDS Correlated Double Sampling

ASIC Application Specific Integrated Circuit

ADC Analog to Digital Converters

UART Universal Asynchronous Receiver/Transmitter

SPI Serial Peripheral Interface

CLB Configurable Logic Block

LUT Look-Up Table

PIP Programmable Interconnect Point

RTN Random Telegraph Noise

DSNU Dark Signal Non-Uniformity

DCNU Dark Current Non-Uniformity

PRNU Photo Response Non-Uniformity

Chapter 1

Introduction

The primary motivation for this project is the development of a cost-effective telescope system designed to detect the electromagnetic counterpart of kilonovae, a type of astronomical event associated with neutron star mergers. This initiative, known as the Dragonfly Project, proposes the deployment of an array of 100 telescopes, each contributing to a unified observational effort. These telescopes are intended to be arranged strategically along the arms of the LIGO-India gravitational wave observatory. The distributed configuration aims to maximize sky coverage and enhance the likelihood of capturing transient optical signals associated with kilonovae, thus complementing gravitational wave detections[2].

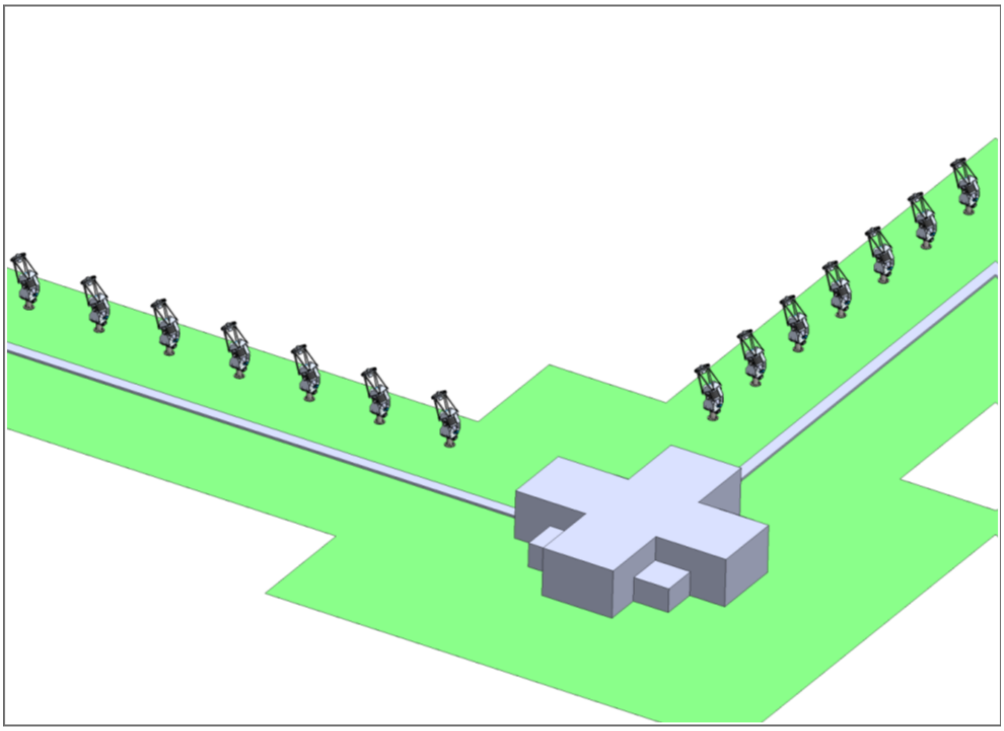


Figure 1.1: Dragonfly Project

A significant component of this effort involves reducing the overall cost of the system, particularly by designing custom readout electronics for image sensors. Commercial astronomical cameras are often expensive because of high-performance electronics and advanced noise suppression techniques. By developing optimized and cost-efficient electronics in-house, the Dragonfly Project seeks to democratize access to astronomical observations and build a scalable system without compromising performance.

1.1 Readout Electronics Development

The design of readout electronics for image sensors requires a strong foundation in hardware description languages, particularly VHDL (VHSIC Hardware Description Language). To acquire and apply this knowledge, the project uses the Digilent Basys-3 development board, which incorporates a Xilinx Artix-7 FPGA. This board serves as an ideal platform for prototyping and experimentation due to its rich feature set and community support [3].

Initial development work began with the implementation of basic digital logic circuits, such as combinational gates, latches, flip-flops, and finite state machines (FSMs). These fundamental building blocks are essential for designing more complex digital systems that manage timing, data acquisition, and signal processing within the readout chain.

As the project progressed, the MicroBlaze soft-core processor—a configurable microprocessor implemented on the FPGA, was introduced. This enabled the integration of intellectual property (IP) cores, which significantly accelerate development by offering pre-verified functional blocks such as UART communication, SPI/I2C interfaces, and memory controllers. These IP cores were critical in establishing communication between the FPGA and a host computer, as well as in enabling interfacing between the image sensor and the FPGA [4]. One major milestone was the successful readout of a linear image sensor, which served as a preliminary step toward handling more complex two-dimensional sensor arrays.

1.2 Image Sensor Characterization

The second major focus of the project involves the characterization of CMOS image sensors used in astronomical imaging. This phase aims to assess whether specific commercial sensors can meet the sensitivity, noise, and dynamic range requirements for detecting faint astronomical transients like kilonovae.

Three different astronomical cameras were selected for analysis: the QHY600, ATIK, and Moravian C1x, all of which utilize the Sony IMX455 CMOS image sensor. These cameras are representative of current high-performance astrophotography systems and offer a valuable benchmark for custom sensor development.

The characterization process included the measurement and analysis of various noise components, such as read noise, dark noise, dark current non-uniformity, dark signal non-uniformity, photon response non-uniformity, random telegraph signal, etc.

Chapter 2

Introduction to CCDs and CMOS

2.1 Introduction

High-performance imaging systems are essential to the field of astronomy, where the ability to detect, capture, and analyze faint celestial objects directly impacts scientific discovery. From deep-sky observations to planetary imaging and space-based telescopes, the sensitivity, resolution, and noise performance of image sensors are critical. Two primary technologies dominate the design of digital image sensors: Charge Coupled Devices (CCDs) and Complementary Metal Oxide Semiconductor (CMOS) sensors. While both perform the task of converting incoming photons into electrical signals, their underlying architectures and operational characteristics lead to distinct advantages and trade-offs, particularly in the context of astronomical imaging.

Historically, CCDs have played a central role in astronomy due to their high quantum efficiency, low dark current, and excellent uniformity across the sensor array. These qualities make them especially suited for long-exposure, low-light applications, such as observing distant galaxies or faint stars. Ground-based and space telescopes such as the Hubble Space Telescope have relied heavily on CCDs to produce high-fidelity scientific images [5].

However, in recent years, CMOS technology has rapidly advanced, offering benefits such as faster readout, lower power consumption, and on-chip processing [6]. These features are increasingly relevant in applications requiring high-speed imaging, wide dynamic range, or miniaturized instrumentation areas gaining traction in time-domain astronomy, adaptive optics, and satellite-based observation systems.

This chapter provides an overview of CCD and CMOS image sensors with a focus on their relevance to astronomical imaging. It introduces their historical development, core operating principles, and key performance differences that influence their selection for various observational goals. By grounding the discussion in astronomical requirements, this chapter lays the foundation for a more detailed analysis of sensor suitability in specific astrophysical contexts.

2.2 Basic Principles of Operation

Both CCD and CMOS image sensors rely on the photoelectric effect, wherein incident photons striking a semiconductor material generate electron-hole pairs [7]. These photogenerated charges are collected and measured to reconstruct an image. The distinction lies in how these charges are read out and processed.

In CCD sensors, the image area consists of an array of light-sensitive elements (pixels), where each pixel accumulates charge proportional to the intensity of incident light during the exposure period. These charges are then transferred across the chip in a coordinated fashion like a bucket brigade to a common output node where the charge is converted to a voltage, amplified, and digitized [7]. The uniformity and simplicity of this process contribute to the high image quality and low fixed-pattern noise of CCDs. However, the sequential readout imposes limitations on speed and introduces complexity in system design.

CMOS sensors, on the other hand, employ active pixel technology where each pixel contains not only a photodetector but also active transistors that amplify and buffer the signal locally [8]. This architecture enables random access readout, faster frame rates, and the possibility of integrating additional processing functions directly onto the sensor chip. However, because each pixel contains more components, CMOS sensors historically suffered from reduced fill factor, higher noise, and lower image quality issues that have been largely mitigated by modern design techniques such as pinned photodiodes, correlated double sampling, and advanced noise-reduction circuits [8].

A key conceptual difference is that in CCDs, charge is transferred and then converted to voltage, while in CMOS sensors, charge is converted to voltage at the pixel level and then read out. This fundamental divergence leads to different noise characteristics, power consumption profiles, and architectural flexibility .

2.3 Charge-Coupled Devices

CCD works on the three basic principles viz charge storage, charge transfer and the conversion of charge into a measurable voltage. In order to understand how the CCDs work we need to first understand the architecture of a CCD. When a photon whose energy is greater than the energy gap is incident on semiconductor, then an electron-hole pair is produced, now in case of an imaging device we do not want this photon generated electron to migrate away, so in order to store this electron we use a Metal-Insulator-Semiconductor (MIS) structure which is a basic building block of any solid-state image sensor. If we take silicon oxide sandwiched between a metal and highly doped p-type semiconductor, here silicon oxide acts as an insulator, then the resulting structure behaves as a capacitor and acts as a charge storage site. When a positive voltage is applied to the metal electrode, the holes in the p-type semiconductor gets repelled creating a depletion region. Now when the photon is incident on this depletion region and an electron hole pair is produced, the hole will move away while the electron is attracted towards the electrode, hence storing the electrons produced by the photon [9] [8].

If several such electrodes are placed on a single silicon chip and the zones of very high p-type doping insulate the depletion regions from each other. So, the charge under each electrode is proportional to the illuminating intensity. To retrieve these charges a method called charge coupling is used, where if the voltage applied to the electrodes is changed sequentially, the stored charge can be moved from one electrode to other, eventually the charge can be moved to an output electrode for readout [9].

Therefore, the charge transfer has to be nearly perfect, and if any defect occurs at any transport site, all other pixels before that site would be blocked. Also, the voltage applied is relatively high, so as to create a depletion region under each electrode and these voltages are rapidly turned from high to low and vice versa to transport the charges. This leads to high power consumption.

2.4 Complementry Metal Oxide Semiconductor

In CMOS arrays the charge to voltage conversion function occurs inside every pixel. A single pixel consists of photodiode, the pixel electrometer, and an output circuitry. The pixel signals are converted from electron to volts, buffered by a source follower transistor and then transferred to the buses through a MOS transistor switches located inside each pixel [10].

Transistors integrated into each pixel perform essential functions such as resetting the photodiode, amplifying the signal, and enabling pixel selection. The most common pixel configurations include the 3-transistor (3T) and 4-transistor (4T) designs. The 3T pixel consists of a reset transistor, a source follower amplifier, and a row select transistor. The 4T architecture introduces an additional transfer gate, which facilitates correlated double sampling (CDS), thereby reducing kTC noise and fixed-pattern noise (FPN) [8].

Passive pixel arrays has photodiodes without internal amplification. They suffer from many limitations, such as high noise, slow readout, and lack of scalability. Active pixel sensor on the other hand implement a buffer per pixel, which improves the performance as amplifier is only activated during readout [11].

2.5 Sensor Performance Parameters

Understanding the performance parameters of an image sensor is crucial when evaluating its suitability for specific applications such as astronomical imaging, scientific observation, or high dynamic range photography. Below is a detailed explanation of key sensor metrics that define image quality, sensitivity, and accuracy:

1. Dynamic Range

Dynamic range refers to the ratio between the maximum and minimum detectable light intensities that a sensor can capture without distortion [9]. It determines how well the sensor can represent details in both very bright and very dark regions of an image simultaneously. This range is typically expressed in decibels (dB) or stops, where a higher dynamic range allows for greater detail preservation in scenes with extreme lighting contrasts.

2. Linearity:

Linearity indicates how proportionally the sensor's output signal corresponds to the incoming light intensity. An ideal sensor produces a linear response that doubles the light, should double the output signal. High linearity ensures accurate representation of brightness levels, which is essential in scientific and calibrated imaging applications.

3. Quantum Efficiency (QE):

Quantum Efficiency is the percentage of incident photons that are successfully converted into electrons (photoelectrons) by the sensor. It is a critical measure of light sensitivity. A higher QE means the sensor can produce a stronger signal from less light, which is particularly advantageous in low-light conditions such as astronomy or fluorescence microscopy.

4. Signal-to-Noise Ratio (SNR):

SNR quantifies the ratio of the useful signal (light-induced charge) to the unwanted noise in the output. A high SNR indicates cleaner, higher-quality images with less visible grain or distortion. SNR is commonly expressed in dB, and improving it is a primary goal in sensor design.

5. Full Well Capacity (FWC):

FWC defines the maximum number of electrons a pixel can store before becoming saturated. It effectively sets the upper limit of a sensor's dynamic range. Sensors with higher FWC can handle brighter light intensities without losing detail due to saturation.

6. Read Noise:

Read noise is the electronic noise introduced during the process of reading out the charge from each pixel, even when no light is present. It is a key limitation in low-light imaging. Lower read noise values result in better sensitivity, enabling detection of faint signals.

7. Photo-Response Non-Uniformity (PRNU):

PRNU measures the variation in pixel response when the entire sensor is uniformly illuminated. It arises due to slight manufacturing inconsistencies among pixels. High PRNU results in visible pattern noise, while low PRNU indicates better uniformity in image capture.

8. Dark Signal Non-Uniformity (DSNU):

DSNU refers to pixel-to-pixel variation in dark current, the noise generated in the absence of light. Like PRNU, DSNU contributes to fixed pattern noise, which can degrade image quality in long exposures or low-light conditions.

9. Temporal Noise:

Temporal noise encompasses fluctuations in pixel output over time under constant illumination. This includes both random noise (thermal, shot noise) and correlated noise (due to fixed structures). Lower temporal noise leads to more consistent and reliable image data.

10. Responsivity:

Responsivity is the output signal (usually in volts or digital counts) per unit of incident optical power (watts). It represents how effectively the sensor converts light into a measurable signal and can vary across wavelengths, making it a key parameter for multispectral or color imaging systems.

11. Modulation Transfer Function (MTF):

MTF assesses a sensor's ability to preserve image contrast at different spatial frequencies. It effectively describes the sharpness or detail-resolving power of the imaging system. Higher MTF values at high frequencies indicate better reproduction of fine image features.

12. Shutter Efficiency:

Shutter efficiency refers to how precisely the sensor's electronic or mechanical shutter controls the exposure time. Inefficiencies can lead to motion blur, uneven exposure, or image artifacts, especially during short exposures or high-speed imaging.

2.6 Comparison of CCDs and CMOS

Feature	CCDs	CMOS
Signal Readout	Sequential, pixel-by-pixel through a single output node.	Parallel readout each pixel has its own amplifier.
Power Consumption	High – due to charge transfer process and external circuitry.	Low – uses less power; integrated amplifiers reduce need for external components.
Image Quality (SNR, DR)	Typically better SNR and dynamic range (especially in older sensors).	Has improved significantly; now competitive or better in many applications.
Speed	Slower readout speeds due to sequential charge transfer.	Faster due to parallel readout paths.
Cost	More expensive – requires specialized fabrication processes.	Less expensive – uses standard CMOS fabrication, compatible with most foundries.
Integration Capability	Limited – mainly image sensing functions only.	High – logic, timing, A/D converters, and more can be integrated on-chip.
Noise Performance	Lower read noise in older designs.	Previously noisier, but modern CMOS designs can rival or surpass CCD in low noise (e.g., BSI, stacked sensors).
Blooming Effect	More susceptible – charge spills over if full-well capacity exceeded.	Less prone – pixel saturation is more localized.
Global Shutter Availability	Difficult to implement.	Easier to implement in specialized CMOS designs.
Applications	Scientific imaging, astronomy, broadcast cameras.	Consumer cameras (phones, DSLRs), automotive, industrial, surveillance, and medical imaging.

Table 2.1: Comparison between CCDs and CMOS

Chapter 3

Digital System Design and Implementation

3.1 Hardware Description Language

In the design and development of digital systems on Field-Programmable Gate Arrays (FPGAs), a Hardware Description Language (HDL) is essential. HDLs allow engineers to describe the behavior, structure, and timing of digital circuits in a textual format, which is then synthesized into hardware logic. Among the most widely adopted HDLs are VHDL (Very High-Speed Integrated Circuit Hardware Description Language) and Verilog. Each offers different syntactic styles and design methodologies, but both serve the same fundamental purpose: enabling designers to implement complex digital logic systems with precision and reusability [12].

3.2 Field Programmable Gate Array

Here I have used Basys-3 board, which has FPGA XC7A35TCPG236-1. An FPGA basically consists of Configurable Logic Blocks, input/output blocks, interconnect resources (programmable interconnect point (PIP) blocks and programmable interconnects), block RAM and DSP slices. The input/output pins can be used as input, output, and both [3]. Configurable logic blocks are the basic elements used to implement a digital system on a FPGA. CLBs consists of look-up tables (LUTs), flip-flops, and multiplexers.

A multiplexer functions as a selector, utilizing N select lines to choose from 2N input lines, producing a single output. In FPGAs, flip-flops serve as basic memory units, each capable of storing a single bit. A look-up table (LUT) can be visualized as a group of flip-flops whose outputs are routed through a multiplexer. The select lines of the multiplexer act as address lines, determining which flip-flop's value is accessed. This configuration allows for the implementation of any combinational logic function, where the number of input variables corresponds to the number of select lines [13]. These LUTs, flip-flops and multiplexers are grouped as slices in the CLB. The interconnect also called routing channels, is a collection of wires and programmable switches which are responsible for connecting CLBs and other building blocks within the FPGA.

3.3 Microcontrollers versus FPGA

Digital systems can be built using individual logic components, a traditional design approach that offers the advantage of using only the specific gates or elements required. This method is relatively straightforward and requires only a basic understanding of logic. However, designing with discrete elements presents several limitations. It often demands substantial physical space, can result in complex and impractical wiring, and leads to a fixed, unchangeable design once implemented.

FPGAs address these challenges effectively. The size of an FPGA remains constant regardless of the logic complexity it handles, and the internal wiring is managed by the chip itself, eliminating concerns about physical interconnections. Most importantly, FPGAs offer the ability to reconfigure the design with ease. By simply modifying the corresponding section of HDL code, designers can update or adjust the logic without altering the hardware. The main requirement is proficiency in hardware description languages, which is a small trade-off for the flexibility FPGAs provide [13].

Application-Specific Integrated Circuits (ASICs) offer a practical alternative to using discrete components, effectively addressing issues related to space and complex wiring. When produced in large quantities, ASICs become cost-effective. Additionally, because ASICs are tailored to a specific application, they utilize only the necessary digital logic elements, making them highly efficient.

Although FPGAs can technically be considered a type of ASIC, in this context, ASIC refers specifically to digital circuits designed for a fixed purpose. Once an ASIC is fabricated, its configuration cannot be changed, which is a key limitation. One of the major challenges with ASICs is the long fabrication time. This is where FPGAs have a significant advantage. In fact, it's common practice to prototype and test ASIC designs on FPGAs before moving forward with mass production, ensuring functionality and reducing risk.

In many situations, a microcontroller can serve as an alternative to an FPGA, as both offer benefits like reconfigurability, compact size, and affordability. However, there are key differences between the two. One major distinction lies in their approach to operation: microcontrollers rely on a fixed instruction set to perform tasks, meaning the design must conform to predefined commands. In contrast, FPGAs offer greater design freedom, allowing users to implement custom logic without being bound by a specific instruction set making them more versatile.

That said, microcontrollers are generally easier to program, especially for beginners, whereas working with FPGAs often requires deeper knowledge of hardware description languages. Another notable difference is power efficiency; FPGAs tend to consume less power in many applications. Additionally, FPGAs excel in parallel processing capabilities. While microcontrollers execute instructions sequentially, FPGAs can perform multiple operations simultaneously, greatly increasing performance for certain tasks. Interestingly, an FPGA can also be configured to emulate a microcontroller when needed.

3.4 Basys3

The Basys3 board is a complete, ready to use digital circuit development platform based on the latest Artix-7 FPGA from Xilinx. With it high-capacity FPGA, low overall cost, and collection of USB, VGA, and other ports, the Basys-3 can host designs ranging from introductory combinational circuits to complex sequential circuits like embedded processors and controllers. It includes enough switches, LEDs and other I/O devices to allow a large number designs to be completed without the need for any additional hardware, and enough uncommitted FPGA I/O pins to allow designs to be expanded using Digilent Pmods or other custom boards and circuits [3].

3.5 IP cores

IP (Intellectual Property) cores are pre-designed and pre-verified functional blocks that can be integrated into a larger hardware design. These cores encapsulate commonly used components such as memory controllers, communication interfaces (UART, SPI, I2C), processors (e.g., MicroBlaze), and more complex subsystems like video encoders or signal processors.

Using IP cores significantly reduces development time and effort because designers don't need to build these blocks from scratch. Instead, they can focus on system integration and customization. IP cores are typically offered in either soft form (HDL-based, flexible) or hard form (fixed, optimized for silicon), and many can be configured through a graphical interface within FPGA design tools like Xilinx Vivado.

In FPGA-based projects, IP cores play a crucial role by enabling modular design, ensuring reliability through pre-validation, and supporting efficient hardware-software co-design especially when integrating processors, peripherals, or high-speed interfaces.

Attach a screenshot of custom IP Attach a flow chart describing method to create a custom IP include example

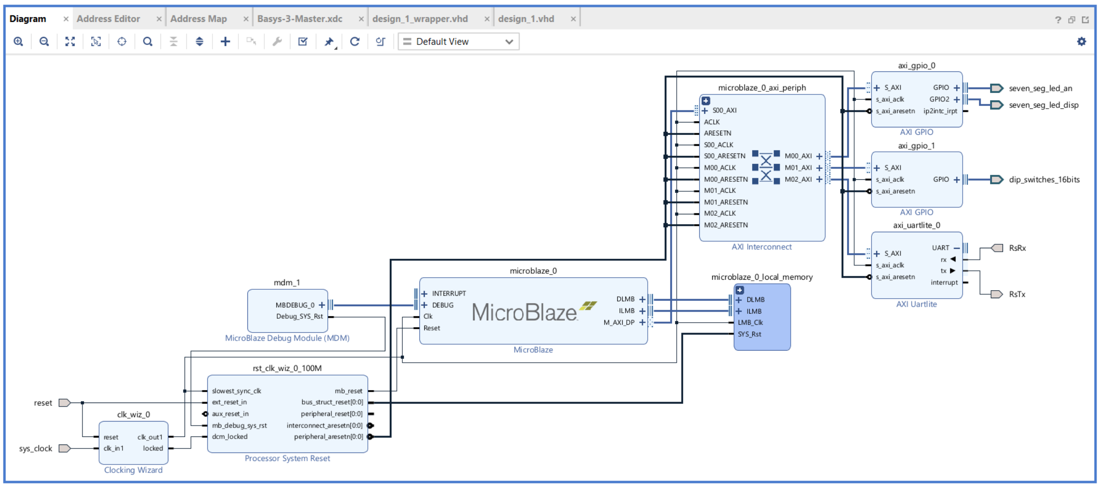


Figure 3.1: An example of UART and switch to seven segment display communication using MicroBlaze

3.6 Soft-core processor

FPGA design has a soft-core processor, which is described in a hardware description language, and can be directly instantiated on the FPGA. The Xilinx FPGA provides MicroBlaze as the soft-core processor, whose bit size of 32-bit and 64-bit, which can be custom selected. In order to use the MicroBlaze, first create a design in the Vivado and after creating the desired design and including all the required IPs, we need to run the synthesis and implementation, and then generate the bit stream. After this we can export our design, and program it in Vitis platform in C language.

3.7 FPGA and PC interface

To explain the MicroBlaze and PC interfacing and communications, I have used an example of UART and switch to Seven segment display communication (figure 3.1):

```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "xgpio.h"
#include "xparameters.h"
#include "xil_types.h"
#include "xuartlite.h"

#define AN_CHANNEL 1
#define DSP_CHANNEL 2
#define SWITCH_CHANNEL 1
```

```
#define SWITCH_MASK 0xff0f
#define AN_MASK 0b0000
#define DSP_MASK 0b00000000

int main()
{
    init_platform();

    int i ;
    print("Enter Input: \n\r");
    i = getchar() - '0';
    xil_printf("Value of i is: %d \n\r", i);

    //UARTLite parameters

    XUartLite_Config* ucfg_ptr;
    XUartLite in_data;

    // GPIO parameters
    XGpio_Config *gpio_cfg_ptr;
    XGpio switch_device, an_device, disp_device;
    u32 data_2, data_3;


    // //uart configurations
    ucfg_ptr = XUartLite_LookupConfig(XPAR_AXI_UARTLITE_0_BASEADDR);

    XUartLite_CfgInitialize(&in_data, ucfg_ptr, ucfg_ptr
                            >RegBaseAddr);
    // XUartLite_CfgInitialize(&out_data, ucfg_ptr, ucfg_ptr
                            >RegBaseAddr);

    //switch configuration
    gpio_cfg_ptr = XGpio_LookupConfig(XPAR_XGPIO_1_BASEADDR);
    XGpio_CfgInitialize(&switch_device, gpio_cfg_ptr,
                            gpio_cfg_ptr->BaseAddress);

    //seven segment display configurations
```

```
    gpio_cfg_ptr = XGpio_LookupConfig(XPAR_XGPIO_0_BASEADDR);
    XGpio_CfgInitialize(&an_device, gpio_cfg_ptr,
                        gpio_cfg_ptr->BaseAddress);
    gpio_cfg_ptr = XGpio_LookupConfig(XPAR_XGPIO_0_BASEADDR);
    XGpio_CfgInitialize(&disp_device, gpio_cfg_ptr,
                        gpio_cfg_ptr->BaseAddress);

    //setting switch data direction
    XGpio_SetDataDirection(&switch_device, SWITCH_CHANNEL,
    SWITCH_MASK);

    //setting display data direction
    XGpio_SetDataDirection(&an_device, AN_CHANNEL, 0);
    XGpio_SetDataDirection(&disp_device, DSP_CHANNEL, 0);

    char SegDsp[] ={0x40, 0x79, 0x24, 0x30, 0x19, 0x12, 0x02, 0x78,
    0x00, 0x10};


    // XUartLite_Send(&in_data, data_1, 8);
    while(1){
        data_2 = XGpio_DiscreteRead(&switch_device, SWITCH_CHANNEL);
        data_2 &= SWITCH_MASK;
        data_2 += AN_MASK;
        // XUartLite_Recv(&in_data, &data_1, 32);
        // data_1 = i;
        data_3 = SegDsp[i];
        // for (int i = 0; i <= 9; i++){
        //  data_3 = SegDsp[i];
        // }
        XGpio_DiscreteWrite(&an_device, AN_CHANNEL, data_2);
        XGpio_DiscreteWrite(&disp_device, DSP_CHANNEL, data_3);
    }

    cleanup_platform();
    return 0;
}
```

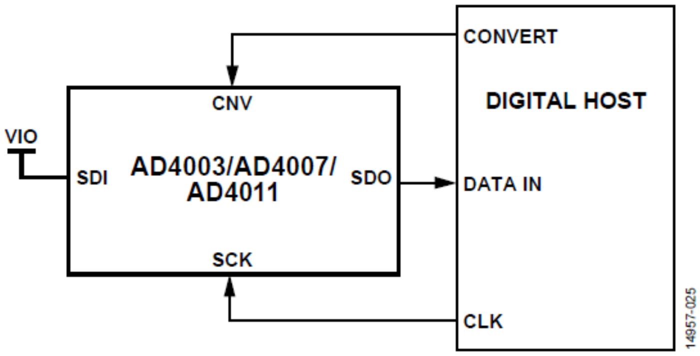


Figure 3.2: AD4003 interface with FPGA via SPI communication[1]

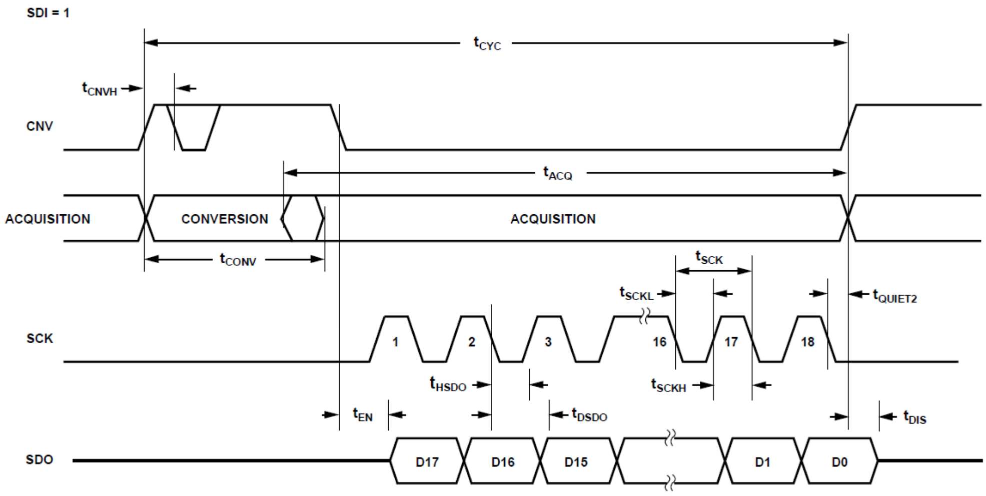


Figure 3.3: Timing diagram for SPI Interface of AD4003 and FPGA[1]

3.8 Sensor and FPGA interface

For interfacing the sensor and the FPGA, a sensor module usually comes with an integrated ADC, where we need to interface the FPGA with the ADC to read the digital data. An example of such interface is via SPI protocol. SPI is a high-speed, synchronous serial communication protocol. Here we have considered an example of AD4003 SPI interface with FPGA (from the datasheet of the ADC [1]) using SPI communication.

```
--AD4003 Controller
-- CS mode, 3-wire without busy indicator
-- Page 30 of 38 of datasheet


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
```

```
entity ad4003 is

  port(
    reset    : in  std_logic;
    clk      : in  std_logic;
    sck      : out std_logic;
    sdo_in   : in  std_logic;
    sdi_out  : out std_logic;
    cnv      : out std_logic;
    data_out : out signed(17 downto 0)
    );
end ad4003;


architecture ad4003 of ad4003 is
  signal data_reg : signed(17 downto 0);
begin

  --For CS mode, 3-wire without busy indicator, Page 32 of AD4003 datasheet.
  sdi_out <= '1';

  process(clk,reset)
    variable cnt : integer := 0;

  begin

    if reset = '1' then
      cnv <= '0';
      sck <= '0';
      cnt := 0;
    else
      if rising_edge(clk) then
 cnt := cnt + 1;
 if cnt = 1 then
   cnv <= '1';
 end if;
 if cnt = 36 then
   cnv <= '0';
 end if;
```

```
if cnt = 40 then
  sck <= '1';

end if;
if cnt = 41 then
  sck        <= '0';
  data_reg(17) <= sdo_in;
end if;
if cnt = 42 then
  sck <= '1';

end if;
if cnt = 43 then
  sck        <= '0';
  data_reg(16) <= sdo_in;
end if;
if cnt = 44 then
  sck <= '1';

end if;
if cnt = 45 then
  sck        <= '0';
  data_reg(15) <= sdo_in;
end if;
if cnt = 46 then
  sck <= '1';

end if;
if cnt = 47 then
  sck        <= '0';
  data_reg(14) <= sdo_in;
end if;
if cnt = 48 then
  sck <= '1';

end if;
if cnt = 49 then
  sck        <= '0';
  data_reg(13) <= sdo_in;
```

```
end if;
if cnt = 50 then
  sck <= '1';

end if;
if cnt = 51 then
  sck        <= '0';
  data_reg(12) <= sdo_in;
end if;
if cnt = 52 then
  sck <= '1';

end if;
if cnt = 53 then
  sck        <= '0';
  data_reg(11) <= sdo_in;
end if;
if cnt = 54 then
  sck <= '1';

end if;
if cnt = 55 then
  sck        <= '0';
  data_reg(10) <= sdo_in;
end if;
if cnt = 56 then
  sck <= '1';

end if;
if cnt = 57 then
  sck       <= '0';
  data_reg(9) <= sdo_in;
end if;
if cnt = 58 then
  sck <= '1';

end if;
if cnt = 59 then
  sck       <= '0';
```

```
  data_reg(8) <= sdo_in;
end if;
if cnt = 60 then
  sck <= '1';

end if;
if cnt = 61 then
  sck       <= '0';
  data_reg(7) <= sdo_in;
end if;
if cnt = 62 then
  sck <= '1';

end if;
if cnt = 63 then
  sck       <= '0';
  data_reg(6) <= sdo_in;
end if;
if cnt = 64 then
  sck <= '1';

end if;
if cnt = 65 then
  sck       <= '0';
  data_reg(5) <= sdo_in;
end if;
if cnt = 66 then
  sck <= '1';

end if;
if cnt = 67 then
  sck       <= '0';
  data_reg(4) <= sdo_in;
end if;
if cnt = 68 then
  sck <= '1';

end if;
if cnt = 69 then
```

```
    sck       <= '0';
    data_reg(3) <= sdo_in;
  end if;
  if cnt = 70 then
    sck <= '1';

  end if;
  if cnt = 71 then
    sck       <= '0';
    data_reg(2) <= sdo_in;
  end if;
  if cnt = 72 then
    sck <= '1';

  end if;
  if cnt = 73 then
    sck       <= '0';
    data_reg(1) <= sdo_in;
  end if;
  if cnt = 74 then
    sck <= '1';

  end if;
  if cnt = 75 then
    sck       <= '0';
    data_reg(0) <= sdo_in;
  end if;
  if cnt = 120 then
     data_out <= data_reg;
    cnt    := 0;
  end if;
       end if;
     end if;
   end process;

end ad4003;
```

Chapter 4

Linear Image Sensor Readout

4.1 Basic image sensor readout mechanism

For the readout of an image sensor, requires three modules, first is a driver module which will supply driver clock pulses to drive the shift registers of the sensors, these driver clocks can be supplied to sensor from the FPGA (VHDL code and see the input pulses in figure). Then the output the sensor comes out as an analog signal, this analog signal needs to be converted to digital output, which can be done either by using an additional ADC, or one can also use the XADC provided in the FPGA board (here in this project Xilinx Basys3 board is used). Then interfacing of the FPGA with sensor is required to read the output of the sensor, this can be done using SPI, I2C, or UART communications. After reading from the sensor we need to save or view digital output, which requires interfacing between FPGA and the computer, this can be done using USB 3.0, UART, interfacing techniques.

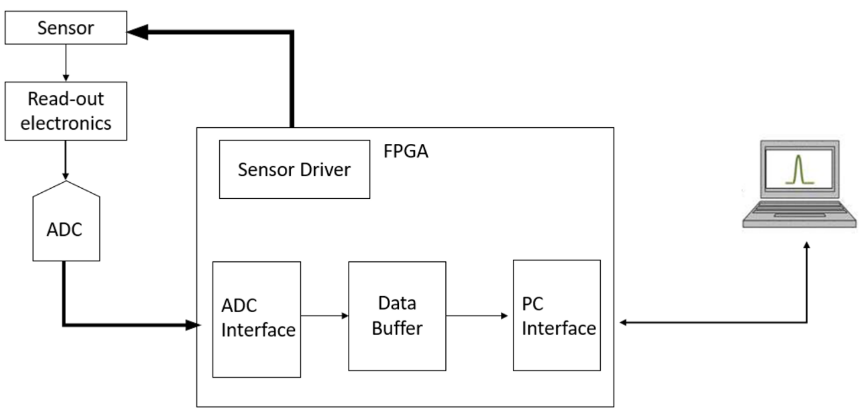


Figure 4.1: image sensor readout mechanism

4.2 NMOS Sensor

The image sensor used in this project is a S3902/S3903 series NMOS linear image sensor (S3902-1024Q/448), it has 1024 pixels and 12 pins. It requires two input clock pulses ($\phi1$ and $\phi2$), which are out of phase, these pulses are used to operate the MOS shift register. The video data rate is equal to the clock pulse frequency since the video output signal is obtained synchronously with the rise of the second pulse ($\phi2$). A start pulse is also required for starting the MOS shift register operation. The time interval between the start pulses is the exposure time of the sensor.

The output of this image sensor is an ACTIVE VIDEO and TRIGGER signal. After all 1024 pixels are read, we get a down pulse of EOS (end of scan) signal. Figure 4.3 and Figure 4.4 shows the equivalent circuitry of the image sensor and the timing diagram of the image sensor.

Figure 4.2: Linear NMOS image sensor and it's driver module

4.3 Driver Module

The image sensor driver module consists of a Programmable Array Logic which is programmed to generate the required input signals for the image sensor i.e.$\phi1$ and $\phi2$, by using the input from the FPGA i.e. the clock and the start pulse. Inn this project the we have used NMOS multichannel detector head(C8892) which is an integrated low-noise driver/amplifier circuit for NMOS linear image sensor. It requires a master start pulse, a master clock pulse, a ±15 V supply, (and an addition 5 V is also given).

The Figure 4.5 and 4.6 shows the bock diagram of the integrated module and it's timing chart respectively.

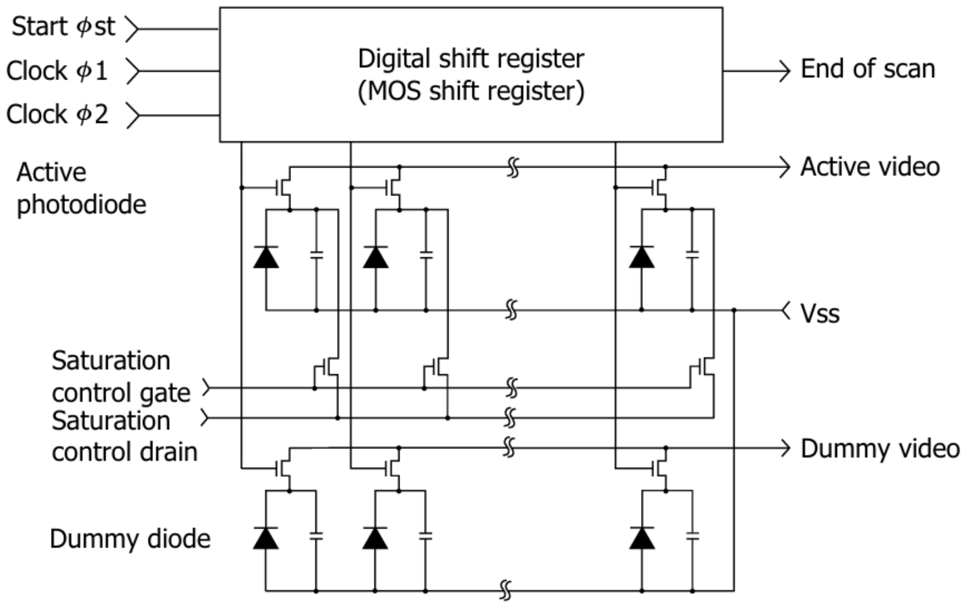


Figure 4.3: Equivalent circuit of the image sensor

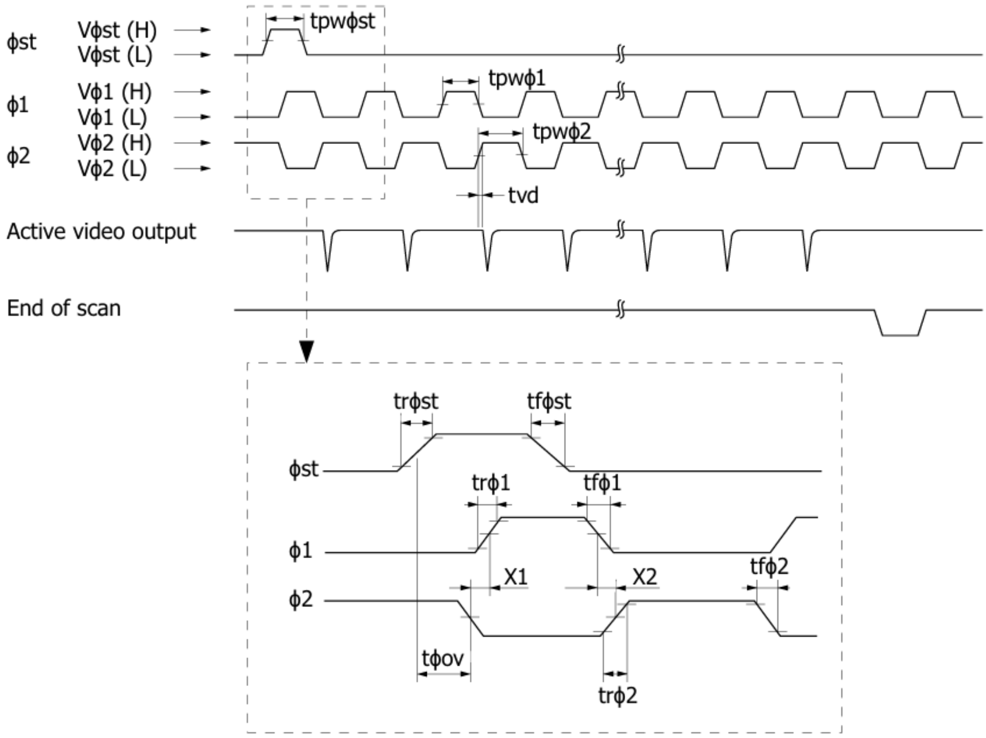


Figure 4.4: Timing diagram of the image sensor

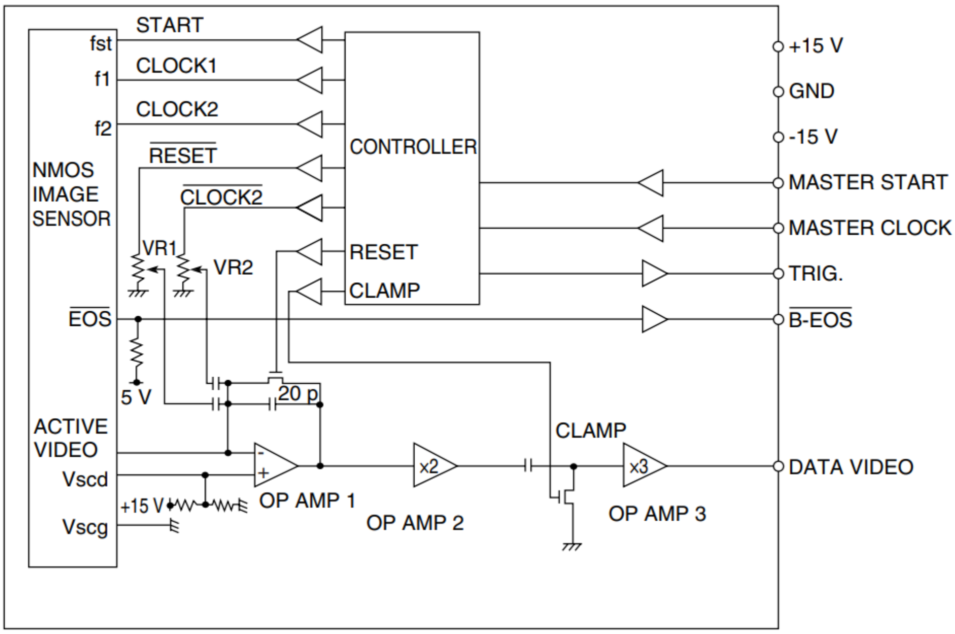


Figure 4.5: Block diagram of the image sensor integrated driver module

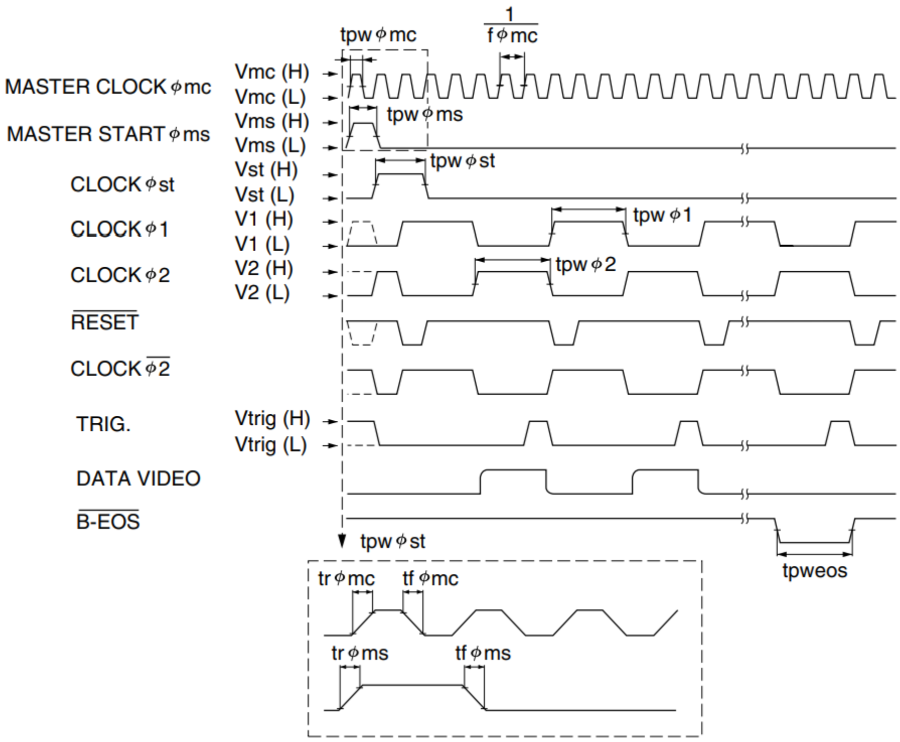


Figure 4.6: Timing diagram of the image sensor integrated driver module

4.4 Results and Discussion

4.4.1 Experimental Setup

The experimental setup for the NMOS linear image sensor readout consist of an integrated drive/amplifier circuit for NMOS linear image sensor, Xilinx Basys3 FPGA kit, ±15 V power supply, additional 5 V supply and an oscilloscope. All the connections were made according to the connections mentioned in the reference module of both the sensor and driver module.

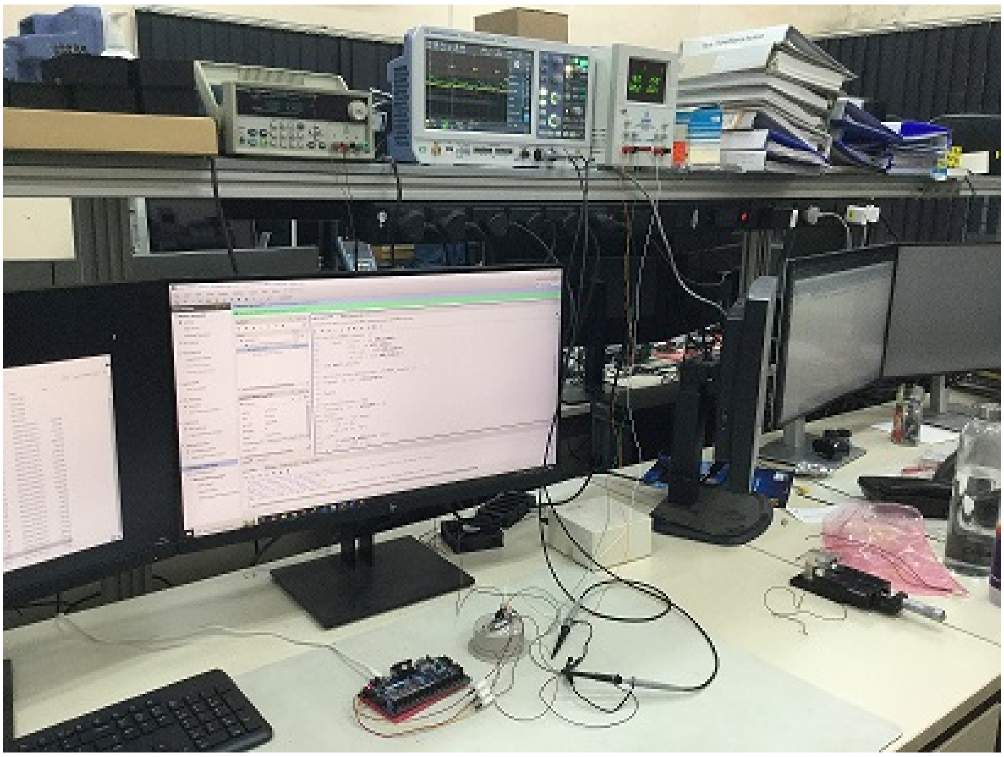


Figure 4.7: Set-up for the readout

4.4.2 Driver Signal

To drive the image sensor a start pulse and driving clock signal is given from the FPGA, the clock signal generated has a time period of 1 s and and start pulse is high for 1 s and repeats after 40 ms. The VHDL code for this has been mentioned below. The Figure 4.9 and Figure 4.10 show the generated clock pulse and the start pulse. These pulses act as the master clock and master start for the sensor integrated module.

VHDL code for driver signal

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

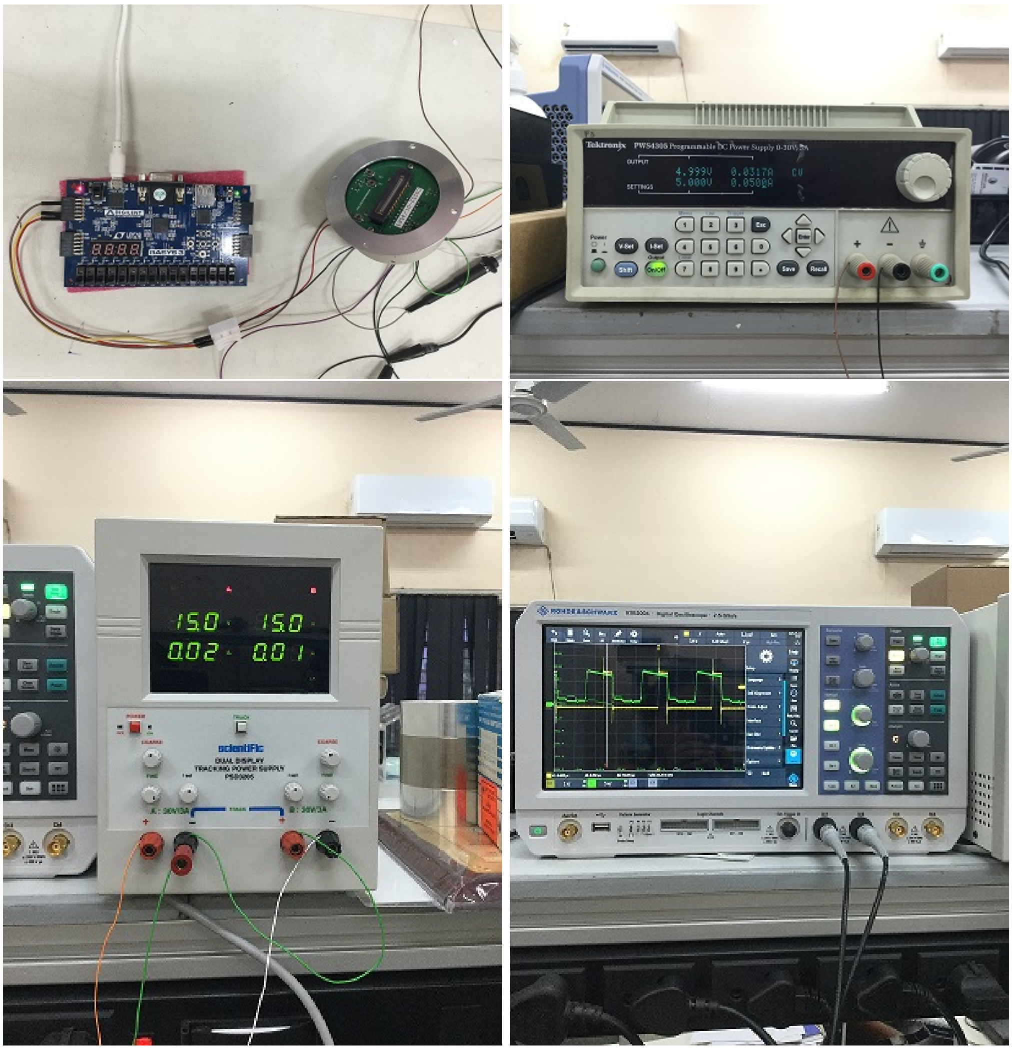


Figure 4.8: Xilinx Basys3 (Artrix7)FPGA board connected to NMOS linear image sensor, 5 volt supply, 15 volts supply, Output Data Video on the oscilloscope

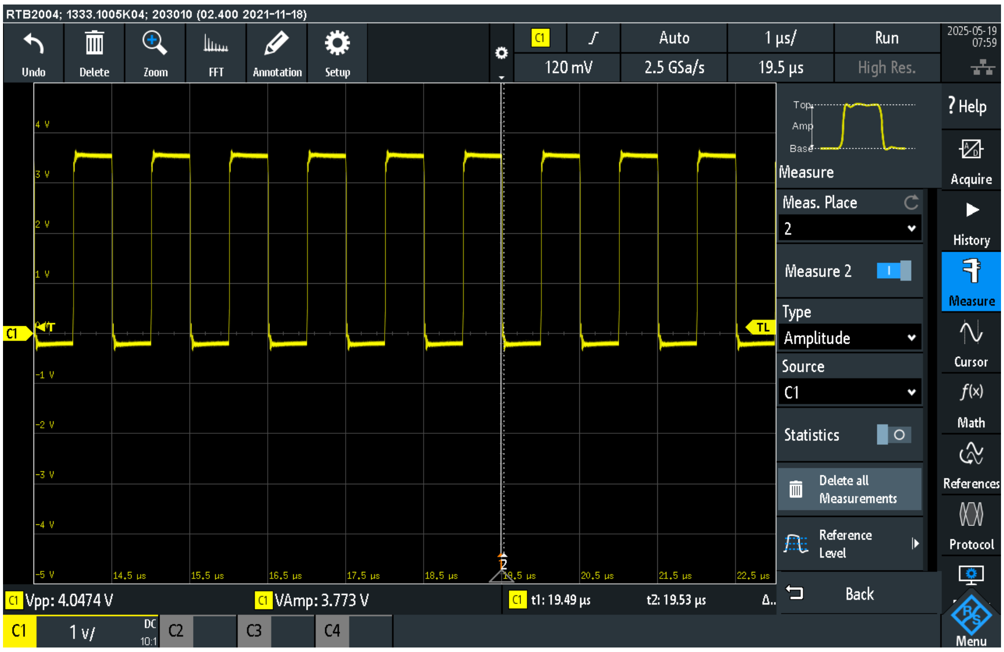


Figure 4.9: Input CLOCK (generated from FPGA)

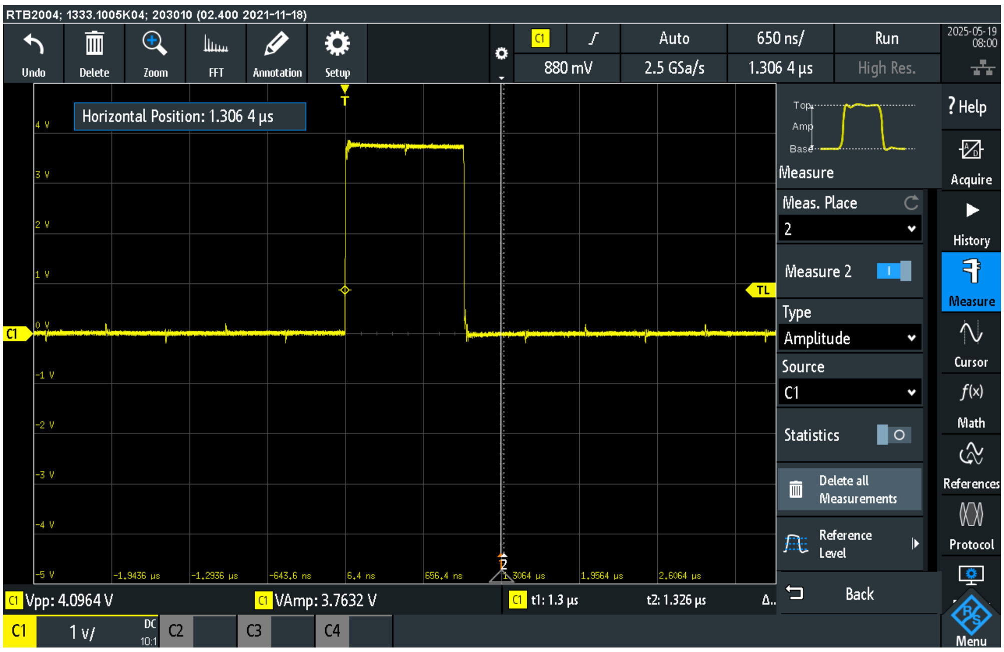


Figure 4.10: Input START Pulse (generated from FPGA)

```
entity NMOS_sensor is
    Port ( reset : in STD_LOGIC;
           clk : in STD_LOGIC;
           clk_out : out STD_LOGIC;
           st_out : out STD_LOGIC);
end NMOS_sensor;

architecture Behavioral of NMOS_sensor is
signal rd_clk : std_logic;

begin

process(reset, clk)
variable cnt : integer:= 0;
begin
if reset = '1' then
    cnt := 0;
    st_out <= '0';
else
    if rising_edge(clk) then
    cnt := cnt + 1;
        if cnt = 1 then
        st_out <= '1';
        end if;
        if cnt = 100 then
        st_out <= '0';
        end if;
        if cnt = 4000000 then
        cnt := 0;
        end if;
   end if;
end if;
end process;

process (reset, clk)
variable cnt_2 : integer:= 0;
begin
if reset = '1' then
```

```
    cnt_2 := 0;
    clk_out <= '0';
else
    if rising_edge(clk) then
    cnt_2 := cnt_2 + 1;
        if cnt_2 = 1 then
        clk_out <= '1';
        end if;
        if cnt_2 = 51 then
        clk_out <= '0';
        end if;
        if cnt_2 = 100 then
        cnt_2 := 0;
        end if;
   end if;
end if;
end process;
end Behavioral;
```

4.4.3 $\phi 1$ and $\phi 2$

The driver module using the input signals, generates two clock pulses $\phi 1$ and $\phi 2$ (Figure4.11). These pulses are out of phase and are used to operate the MOS shift register. The analog output of the sensor has the data rate equal to that of ϕ 1 and ϕ 2, but the output video rate from the module is 1/6 of the master clock (supplied from the FPGA).

4.4.4 Analog Output

The resulting output is an analog signal(Data Video), along with a Trigger Signal as an additional output. Figure-4.12 shows the result of the NMOS readout on an oscilloscope. As already discussed the output video signal is synchronous with the rise of the $\phi 2$, in Figure-4.13 we observe output data video and $\phi 2$. Additionally, validating the given timing diagram of the sensor module, we can see in the Figure-4.14 that the EOS (end of scan) goes low after the last photodiode is read.

Figure 4.11: $\phi 1$ and $\phi 2$

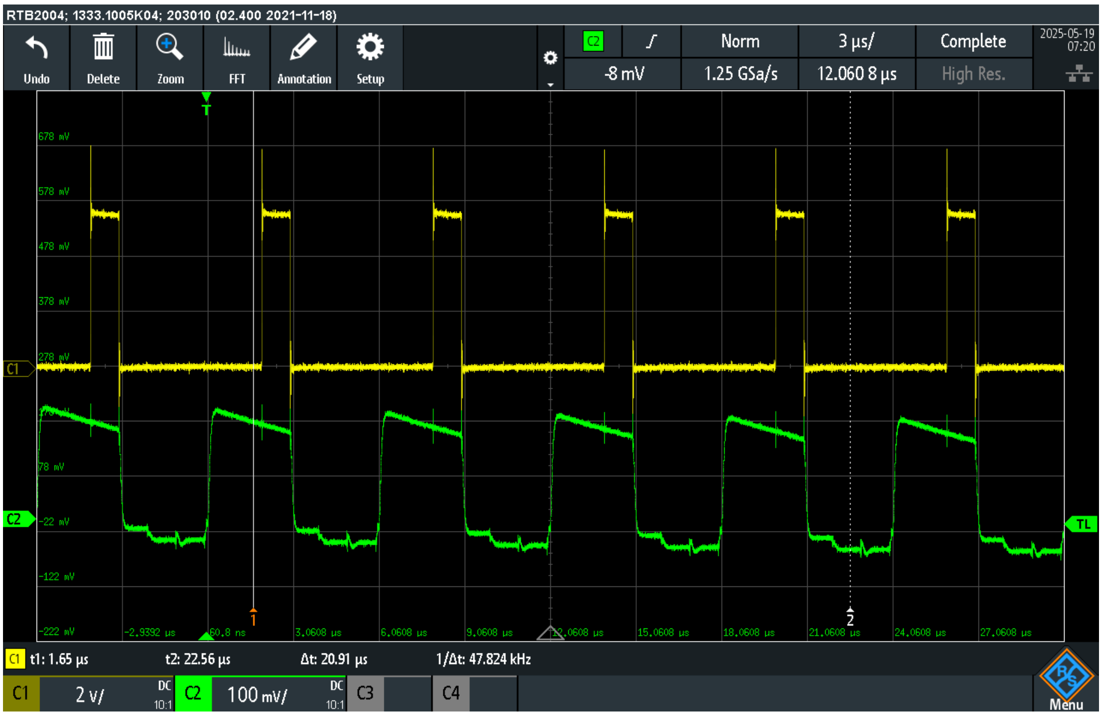


Figure 4.12: Output data video and trigger signal

Figure 4.13: Output Data Video and $\phi 2$

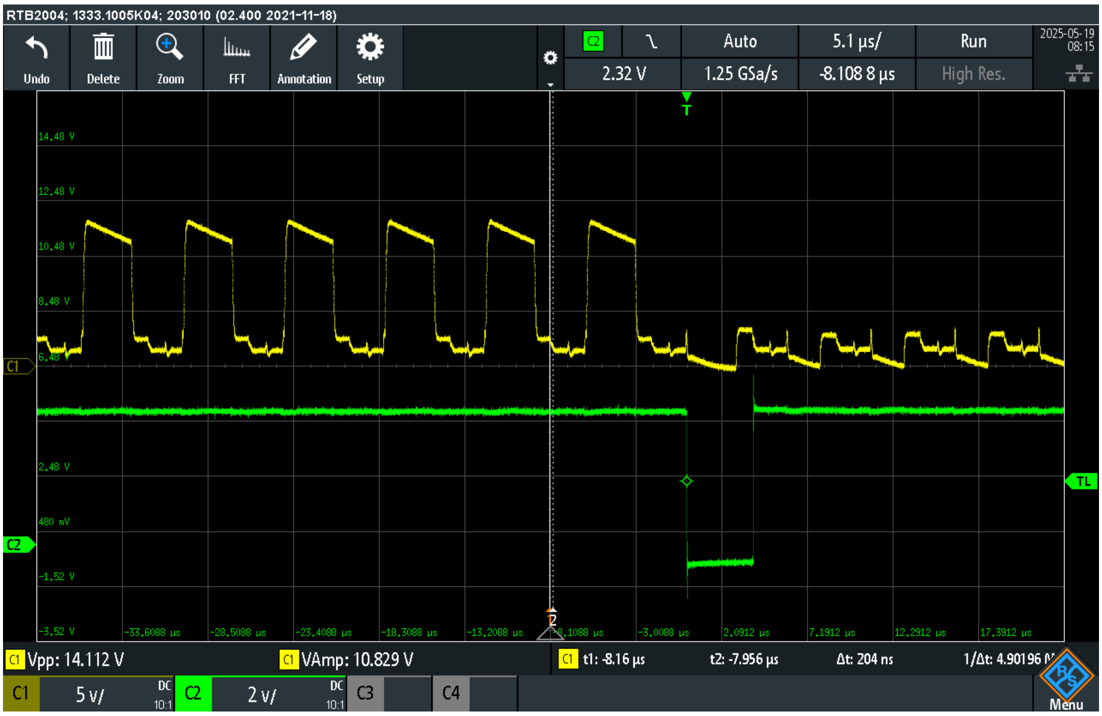


Figure 4.14: Output Data Video and EOS

Chapter 5

Image Sensor Characterization

In the design and deployment of modern imaging systems, particularly in scientific, industrial, and astronomical applications, the characterization of image sensors plays a vital role in ensuring performance reliability, precision, and suitability for specific use-cases. Characterization involves a comprehensive analysis of various performance parameters such as dynamic range, quantum efficiency, read noise, linearity, full well capacity, and dark current, among others. These parameters directly influence the quality and accuracy of the captured images and, consequently, the outcomes of the systems that rely on them.

For applications such as astronomical imaging, where the detection of faint celestial objects or electromagnetic counterparts of transient phenomena (e.g., Kilonovae) is required, the performance of the image sensor becomes critically important. Even subtle inaccuracies in sensor response—such as non-uniformities in dark signal, elevated read noise, or poor linearity—can result in loss of vital information or misinterpretation of data. Therefore, rigorous sensor characterization is necessary to understand the limitations and optimize the system configuration accordingly.

Moreover, the process of characterization assists in evaluating sensor suitability for a given task before full system integration. For instance, comparing different CMOS sensors (such as the Sony IMX455 used in scientific cameras like QHY600, ATIK, and Moravian C1x series) requires a detailed understanding of their quantum efficiency, temporal noise behavior, photo-response uniformity, and other intrinsic features. Such insight enables designers and researchers to make informed decisions about sensor selection and system calibration.

Characterization is also indispensable during the development of readout electronics and interfacing strategies, particularly when custom FPGA-based systems are used. Understanding the electrical and optical behavior of the sensor ensures that signal processing pipelines can be correctly designed to preserve image fidelity. This becomes increasingly relevant in projects aiming to minimize system costs while maximizing sensor performance, such as the Dragonfly project aimed at kilonova detection using an array of low-cost telescopes.

Hence, image sensor characterization is a foundational step in both research and product development. It provides the quantitative data required to benchmark sensor performance, identify artifacts and limitations, and tailor system-level optimizations to meet demanding application needs. Without such analysis, imaging systems risk underperforming or generating unreliable data, especially in precision-critical environments.

5.1 Noise Characterization

In image sensors, noise generally falls into two main categories: signal-related noise, which comes from the light itself and isn't influenced by the sensor, and sensor-related noise, which originates within the sensor hardware. Sensor noise includes things like fixed pattern noise, shot noise (caused by fluctuations in light), and read noise (generated during signal processing).

Some types of noise called temporal noise, change over time, while others known as spatial noise, stay consistent but differ from pixel to pixel. Spatial noise can usually be corrected using standard image-processing techniques. Temporal noise, however, such as electronic interference, is much harder to deal with and often cannot be fully removed.

CMOS sensors are especially vulnerable to a particular type of temporal noise called Random Telegraph Signal (RTS) or Salt and Pepper noise, which appears as sudden flickers or black-and-white specks in images. It is important to be aware of such noise as in case of transient observations where a source may be a point source which takes only a pixel's space, so it's hard to know if the pixel is showing RTS or giving off the actual signal, although not completely, but this noise can be corrected by some measures such as masking [Alarcon, et al.].

5.2 Cameras

For this project the cameras used where the QHY600, ATIK APX60 and Moravian c1x. All the cameras had Sony's IMX455 CMOS image sensor, which is a scientific CMOS camera, made for the purpose of astronomical imaging. This project shows a study of all these cameras, with respect to their noise characteristics.

The following figures show the cameras and their basic details.

Figure 5.1: QHY600, ATIK APX60 and Moravian c1x Cameras

Model	QHY600	ATIK	Moravian c1x
Image Sensor	IMX455	IMX455	IMX455
Pixel Size	3.76 × 3.76 µm	3.76 × 3.76 µm	3.76 × 3.76 µm
Pixel	9576 × 6388	9576 × 6388	9576 × 6388
Read noise	1.0e- to 3.7e- (Standard Mode)	1.2 e-	1.46 e- (at Gain no. = 2750) (conversion factor = 0.26 e-/ADU)
Full well capacity	Standard Mode >51ke-	51,000 e-	16,900 e-
Dark current	0.0022e-/p/s @ -20°C 0.0046e-/p/s @ -10°C	0.005 e-/p/s	Very low

Figure 5.2: Cameras basic information

5.3 Telescope, Resolution and Plate scale

The telescope to be used in the Dragonfly project is a 20 inch F/9 Ritchey Chretien Telescope. It has an effective focal length of 4.529 meters and an entrance pupil diameter of 0.5 meters. The wavelength at which it will be used is 0.5875 µm. Therefore, the resolution turns out to be 0.29 arc sec. The IMX455 CMOS image sensor has a pixel size of 3.75 µm, hence the resulting plate scale is 0.17 arc sec. Figure 5.3 shows the Ritchey Chretien Telescope.

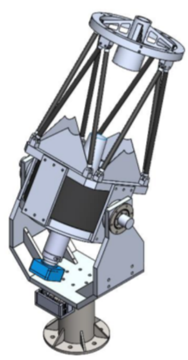


Figure 5.3: 20 inch F/9 Ritchey Chretien Telescope

5.4 Methodology

I initially started by reading the frames from the camera in FITS format. Reading the header of the frames and took some dark frames. The dark frames were taken by varying the exposure times, temperature, and gain settings and studying the effect of these parameters on the digital counts as well as the dark noise (note: here the readout noise is not elliminated; therefore the noise is contributed by both read noise and dark noise). These frames were taken via QHY600 camera, installed at University of Glasgow's Acre Road Observatory. The steps followed were, take the frames as FITS format, read the FITS files, crop the frames, and stack the frames taken at same parameters, and then plot these frames and perform the necessary data analysis.

5.4.1 Dark Frames at different exposure (QHY600)

Dark frames taken at 10s, 30s, 60s, 300s exposure times (see in Figure-5.4, Figure-5.5 and Figure-5.6).

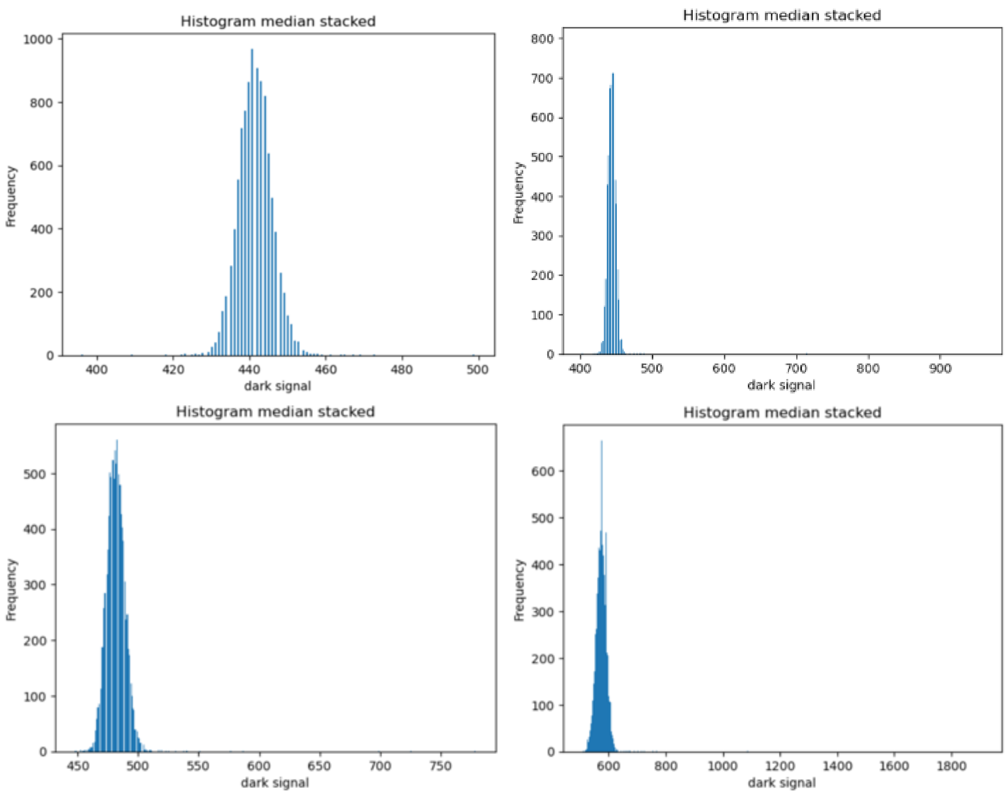


Figure 5.4: Histograms signal(digital counts) distribution, of the dark frames at exposures 10s, 30s, 60s and 300s(QHY600 camera)

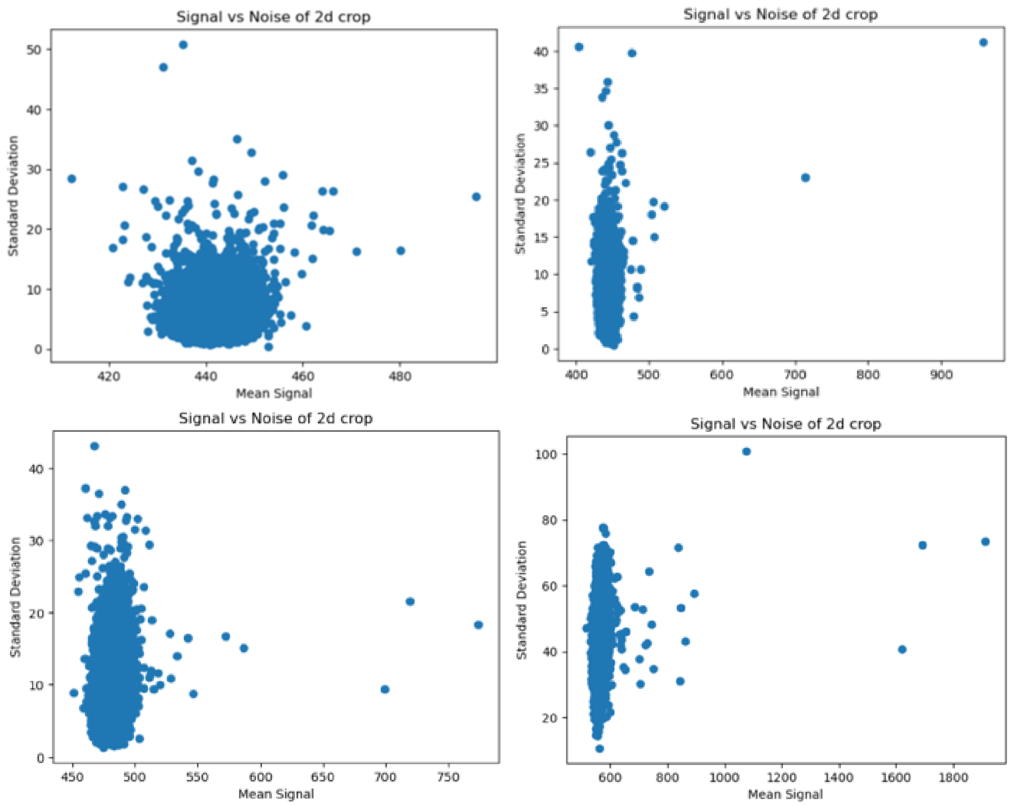
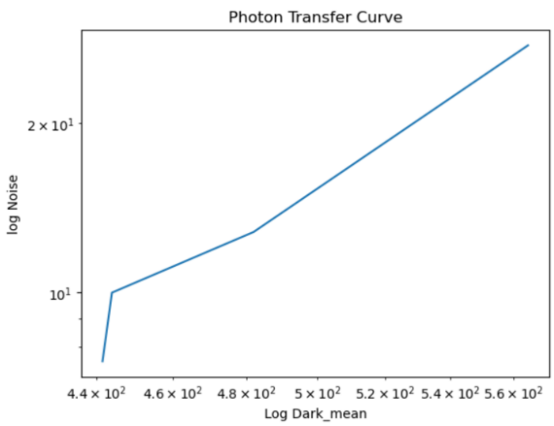


Figure 5.5: Standard deviation vs Mean signal(ADU)(digital counts) of each pixel, for the dark frames at exposures 10s, 30s, 60s and 300s(QHY600 camera)



At gain 30	Temp. -14 ^oC	
Exposure(seconds)	Mean	Standard Deviation
10	442	7.545456
30	444	9.994847
60	482	12.81717
300	565	27.530414

Figure 5.6: Comparison of dark frames at all exposure times

5.4.2 Dark Frames at different temperature (QHY600)

Dark frames taken at temperatures -5, -10, -15, -20 deg C (see in Figure-5.7, Figure-5.8 and Figure-5.9).

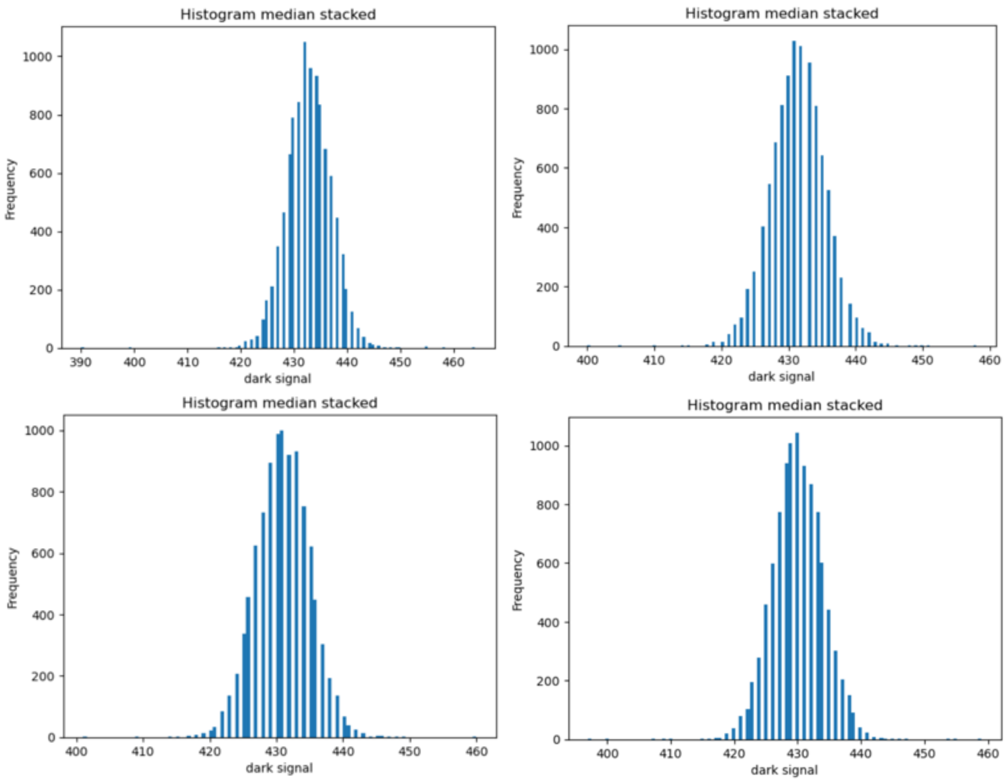


Figure 5.7: Histograms signal(digital counts) distribution, of the dark frames at temperatures -5, -10, -15 and -20 degC (QHY600 camera)

5.4.3 Dark Frames at different gain setting (QHY600)

Dark frames were taken at different gain setting. The gain setting at which observations were made are 10, 30, 50 and 70 (see in Figure-5.10, Figure-5.11 and Figure-5.12).

5.5 Results and Disscussions

5.5.1 Bias level and Read Noise

QHY600

To determine the read noise, 10 bias frames were taken and these frames were then median-stacked (unclipped) and then standard deviation versus mean for each pixel was plotted, the result shows a kite-shape distribution.

According to the figure 5.4, the Read noise is equal to 5.744 ADU and mean signal is equal to 425.7 ADU. Also the small blob shape, can be due to some memory issue

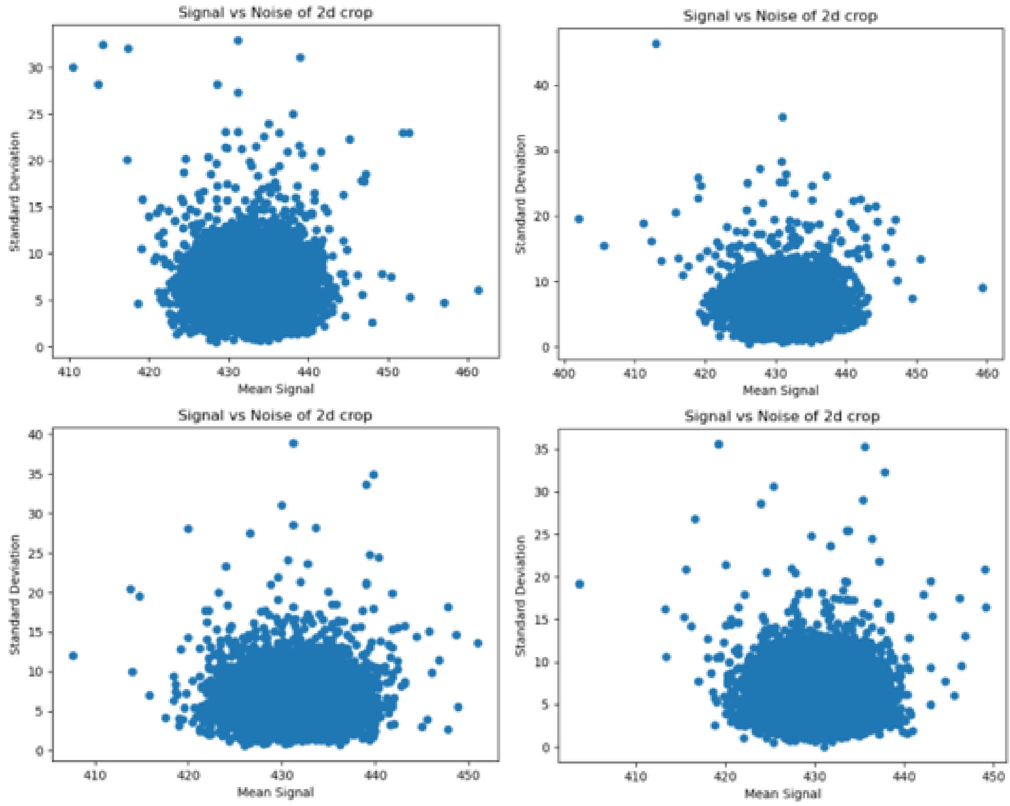


Figure 5.8: Standard deviation vs Mean signal(ADU)(digital counts) of each pixel, of the dark frames at temperatures -5, -10, -15 and -20 degC (QHY600 camera)

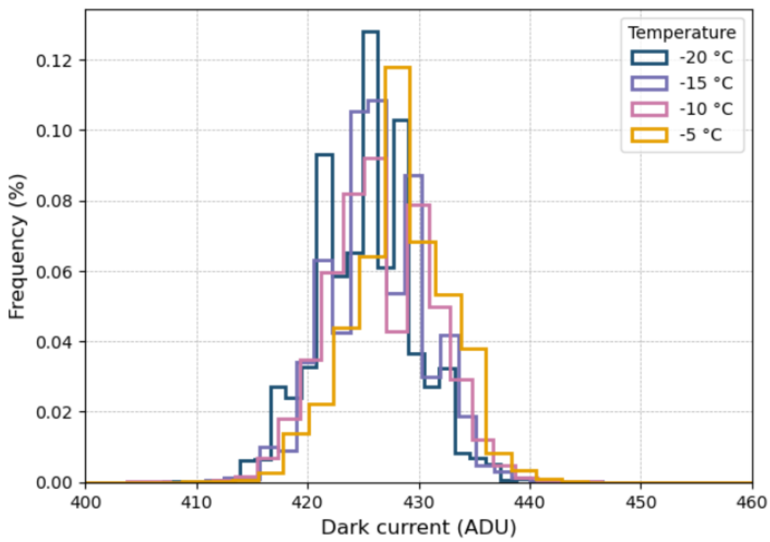


Figure 5.9: Comparison of dark frames at different temperatures

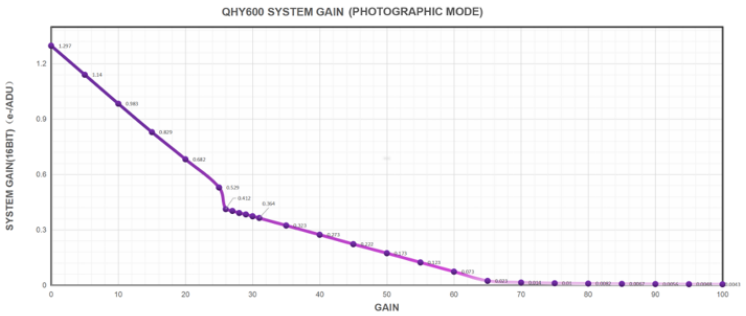


Figure 5.10: From the datasheet of QHY600, the gain and sensitivity graph

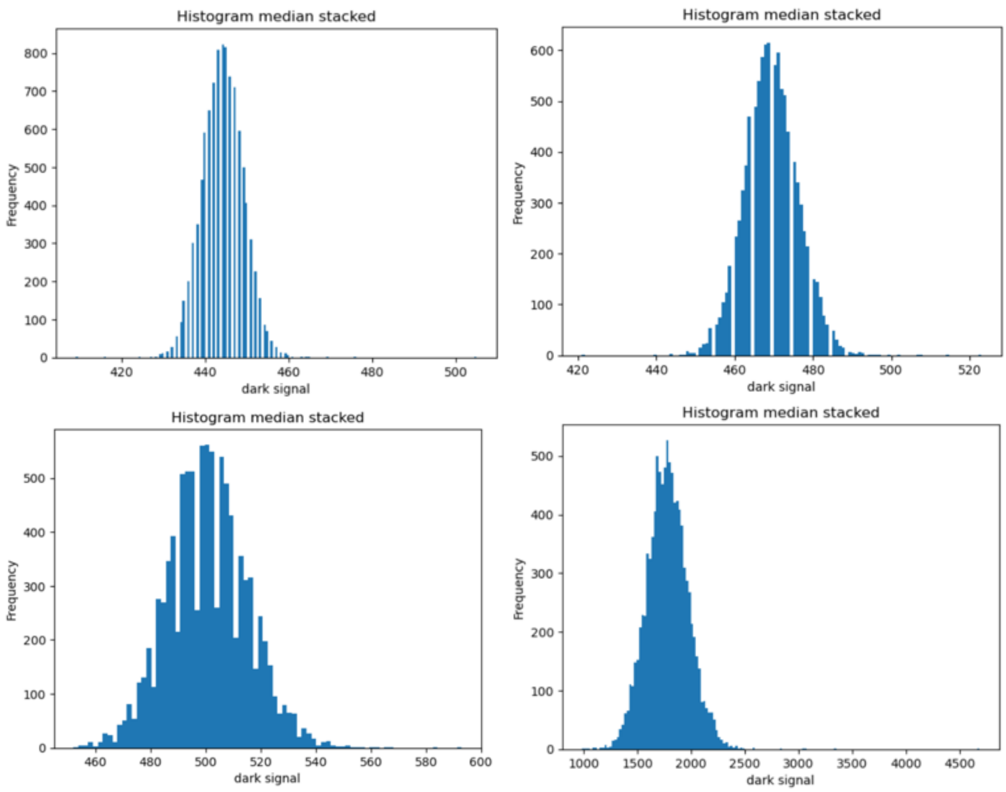


Figure 5.11: Histograms signal(digital counts) distribution, of the dark frames at gains 10, 30, 50 and 70 (QHY600 camera)

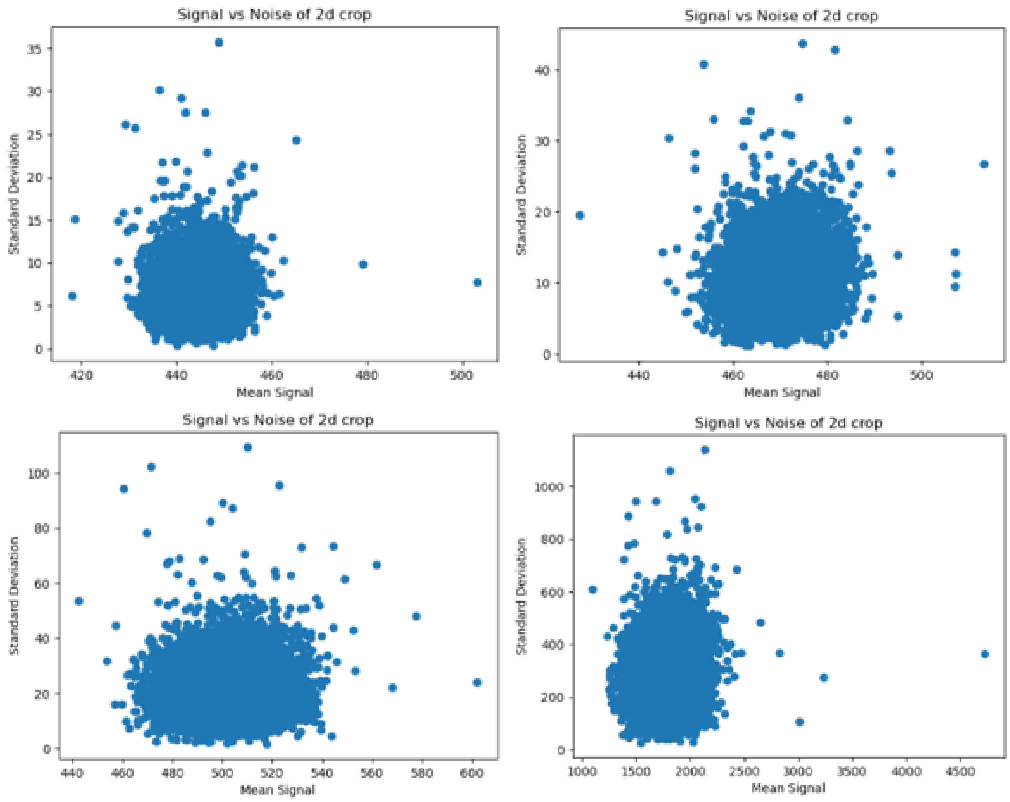


Figure 5.12: Standard deviation vs Mean signal(ADU)(digital counts) of each pixel, of the dark frames at gains 10, 30, 50 and 70 (QHY600 camera)

with the camera, and since the number of frames taken over here is very less, we cannot observe large distribution, to observe the better read noise one must use large number of frames. This issue is corrected with the other two cameras.

ATIK APX60

100 bias frames were taken at 0 degC, then median stacked. The figure shows per pixel mean vs standard deviation of ATIK CMOS Camera at 0 degC.

Readout Noise (Standard deviation) (ATIK) = 3.928 ADU Mean (ATIK) = 500.647 ADU.

Moravian c1x 61000

The process was repeated for the Moravian c1x camera, but here we took 100 bias frames at different temperatures, the frames were taken at gain setting 2750 (0.26 e-/ADU), and then the read noise was determined using the standard deviation versus mean signal plot(figure). The read noise at 0 degC, was 4.21 ADU (1.09 e-). A comparison of read noise and mean signal at different temperatures is shown in figure 5.15.

The read noise at 0 degC, was 4.21 ADU (1.09 e-). A comparison of read noise and

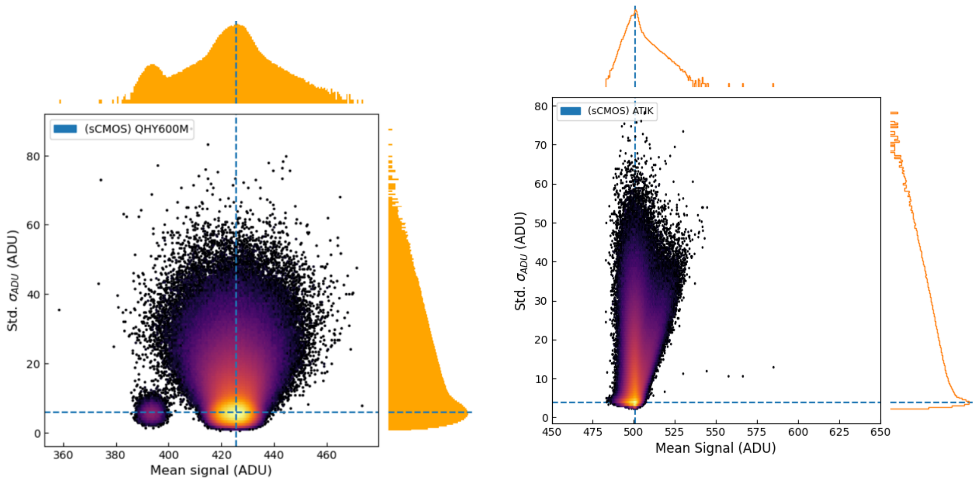


Figure 5.13: Standard deviation vs. mean signal of each QHY600M pixel (warm colors) in 10 consecutive bias frames and ATIK APX60 100 bias frames respectively. The histogram of the distribution of these two variables is included at top and to the right-hand side.

mean signal at different temperatures is shown in figure 5.15. Here we do observe the increase in standard deviation and mean signal, with increase in temperature, but the change is not significant.

5.5.2 Dark Signal Non-Uniformity(DSNU)

The dark signal varying from pixel to pixel is called dark signal non-uniformity.[EMVA Standard] A master bias obtained by median stacking of 100 bias frames, Figure 5.17 show the master bias and signal distribution of 5-sigma clip of per pixel mean obtained from 100 bias frames stack (Moravian c1x).

5.5.3 Random Telegraph Signal / Salt and Pepper Noise

The methodology presented in (Alarcon et. al. 2023) was studied to gain a deeper understanding of image sensor calibration techniques. Figure 5.18 shows temporal mean signal (left-hand) and standard deviation n terms of readout noise (right-hand) obtained in a 20 × 20 pixel central region of 1000 bias frames taken consecutively with the QHY600M Pro, presented by Alarcon, et.al. Jumps between above and below average signal values were observed at different frames, regardless of their exposure time and temperature. A pattern of bright and dark pixels is observed that appears and disappears from one frame to the next. This effect, is random telegraph noise (RTN) and is sometimes referred as Salt and Pepper, [6]. RTN is the fluctuation of the signal between discrete levels as a consequence of the capture and emission of charges by defects or

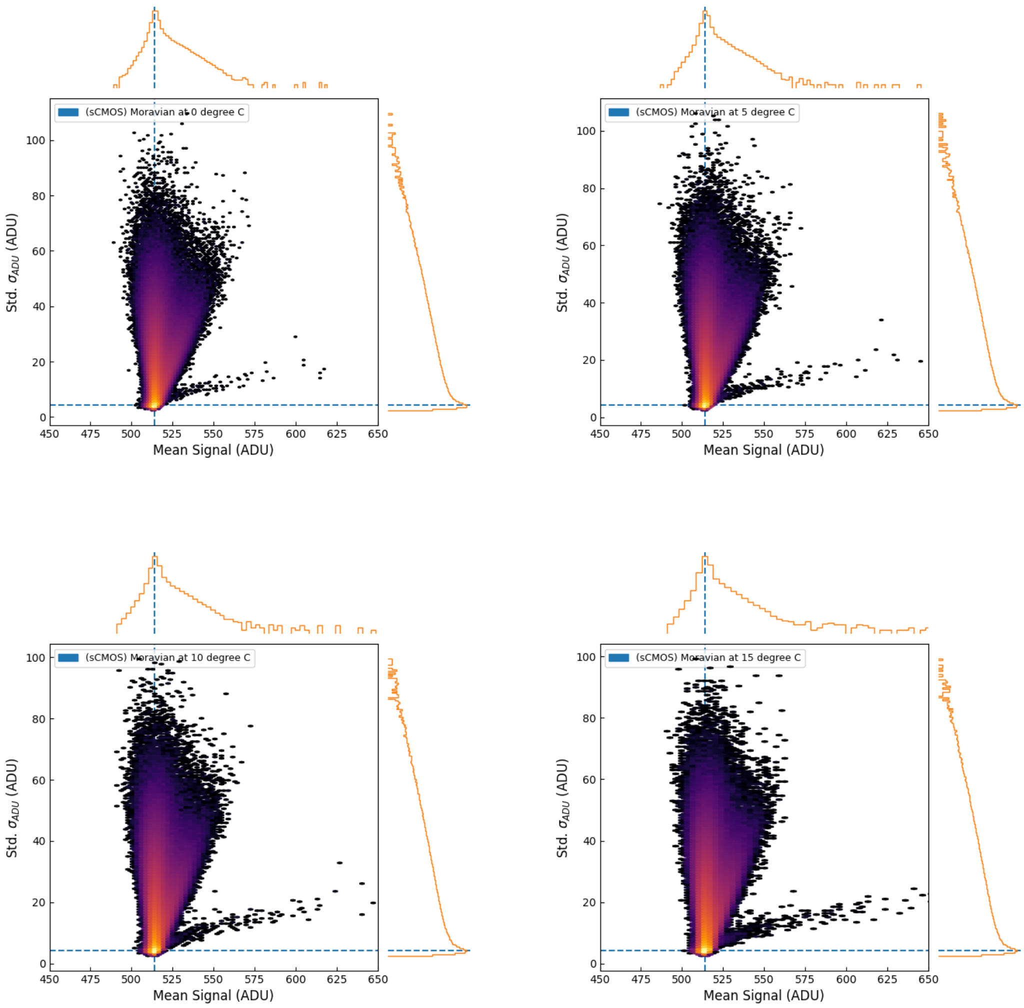


Figure 5.14: Standard deviation vs. mean signal of each Moravian c1x camera pixel, at temperatures, 0, 5, 10, 15 deg C

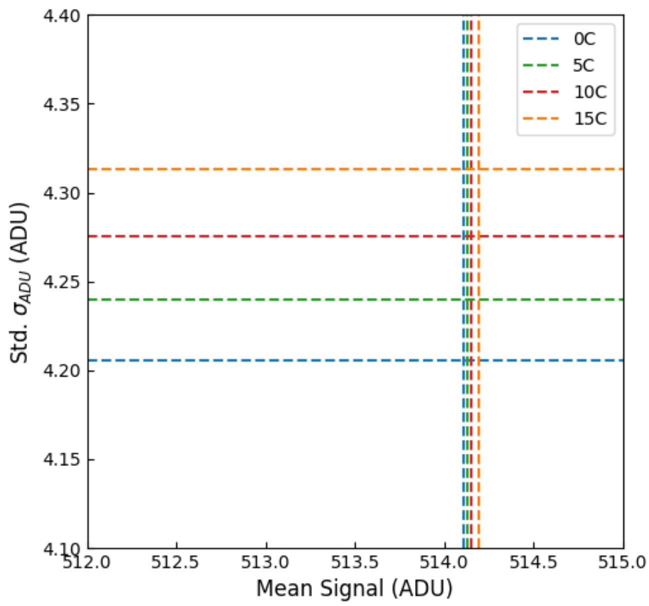


Figure 5.15: Standard deviation vs. mean signal at different temperatures.

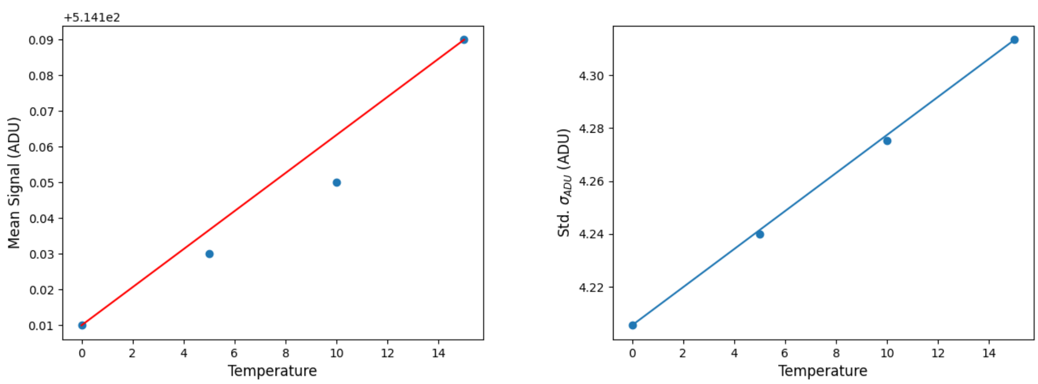


Figure 5.16: Standard deviation and mean signal at different temperature

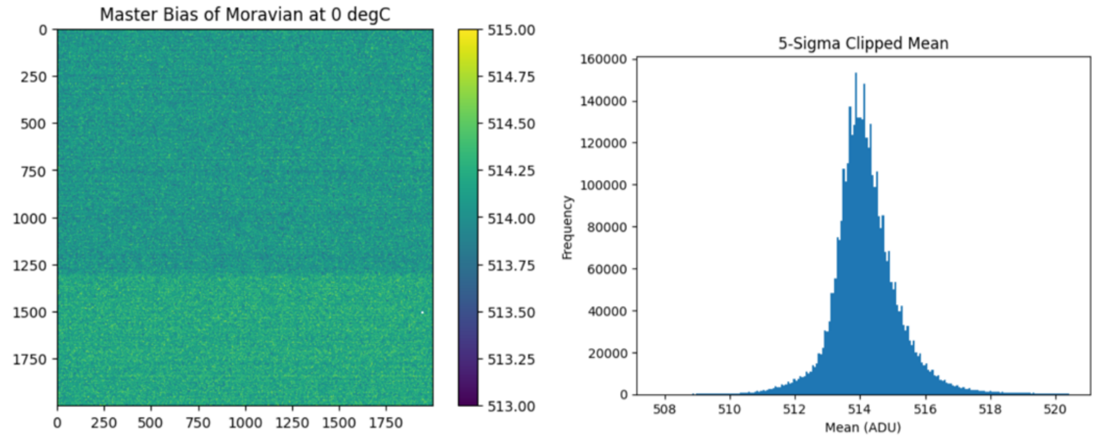


Figure 5.17: Master Bias obtained by median stacking of 100 bias frames and 5-sigma clip of per pixel mean from 100 images stack respectively.

traps located very close to the Si–SiO2 interface (Uren et al. 1985).

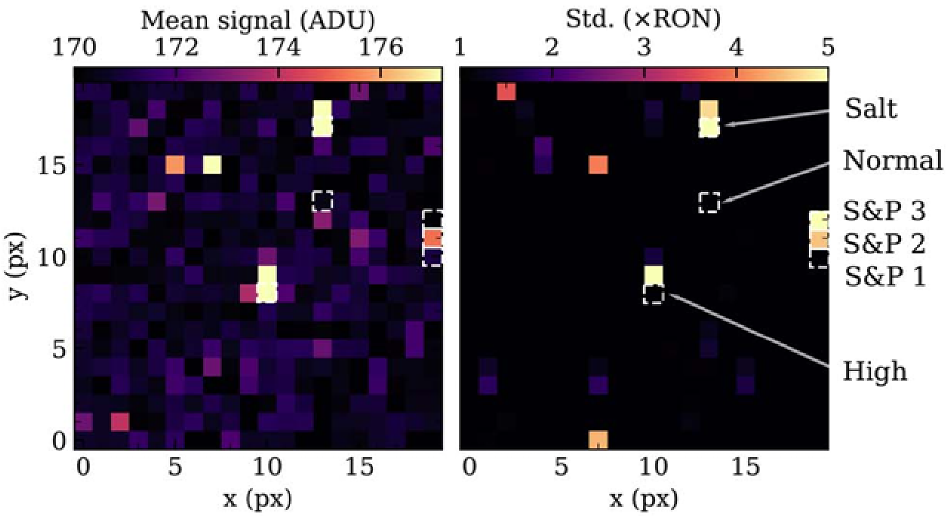


Figure 5.18: Alarcon et. al. - RTN observations in QHY600 pro camera

To test the presence of RTN in the Moravian c1x camera, 500 bias frames were taken, and then randomly selected pixel's count was plotted against the frame number, and we observed the presence of random telegraph noise Figure 5.19. The experiment was repeated for 100 bias frames, and we still observed RTN, Figure 5.20.

5.5.4 Dark Current Analysis

When we take frames without any light entering onto the camera sensor, we still observe some output at the pixels, these pixel counts are due to noise in the electronics which may have been generated because of the thermal generations of charges. This noise is called dark current. Dark current is dependent on both temperature and exposure time.

To study dark signal, dark frames were taken at 100 s and 300 s respectively at 0

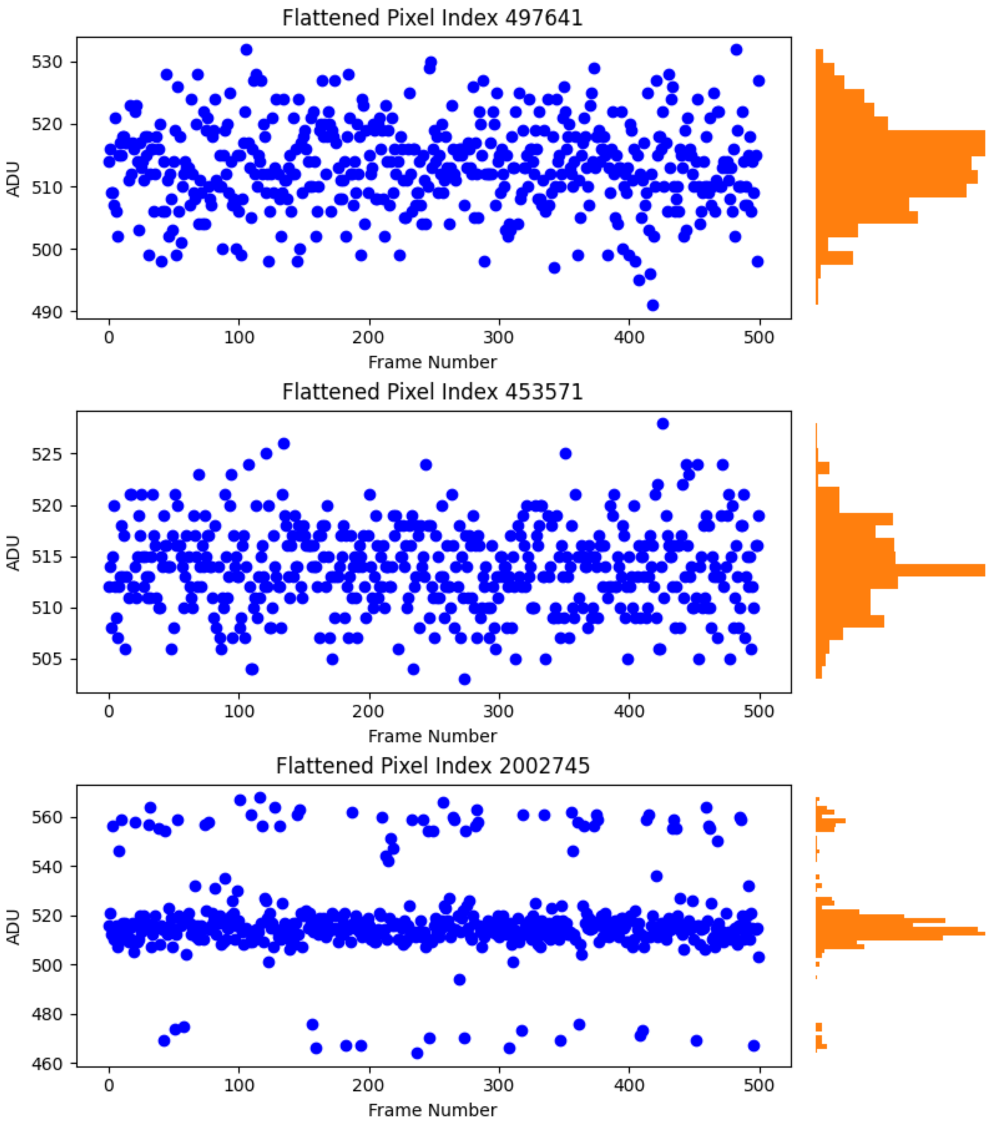


Figure 5.19: Signal Count vs frame number, for 500 bias frame taken at 0 degC (Moravian c1x camera)

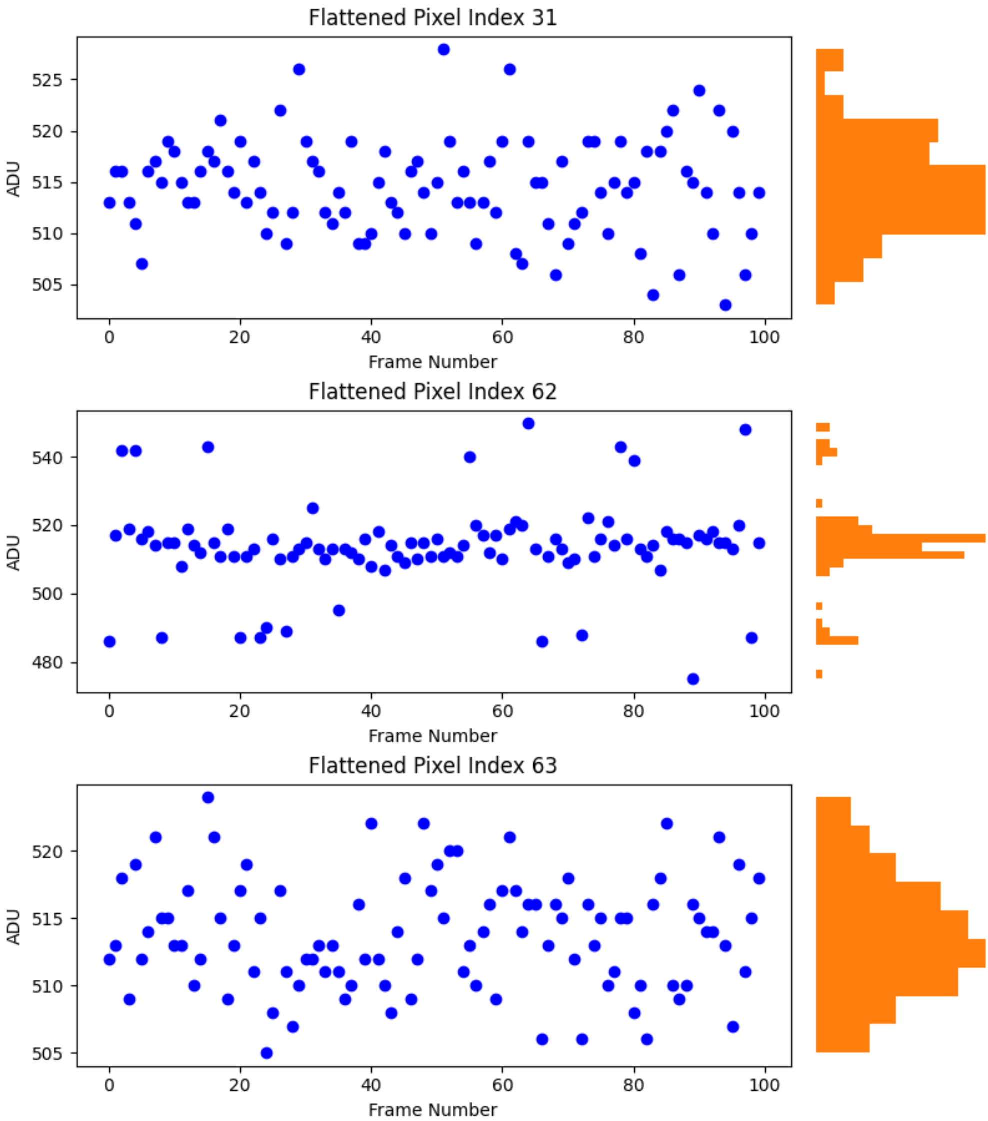


Figure 5.20: Signal Count vs frame number, for 100 bias frame taken at 0 degC (Moravian c1x camera)

degree Celsius and their distribution was plotted shown in figure 5.21. The median value from the distribution came out to be 0.240 ADU and 0.700 ADU at 100 second and 300 second respectively.

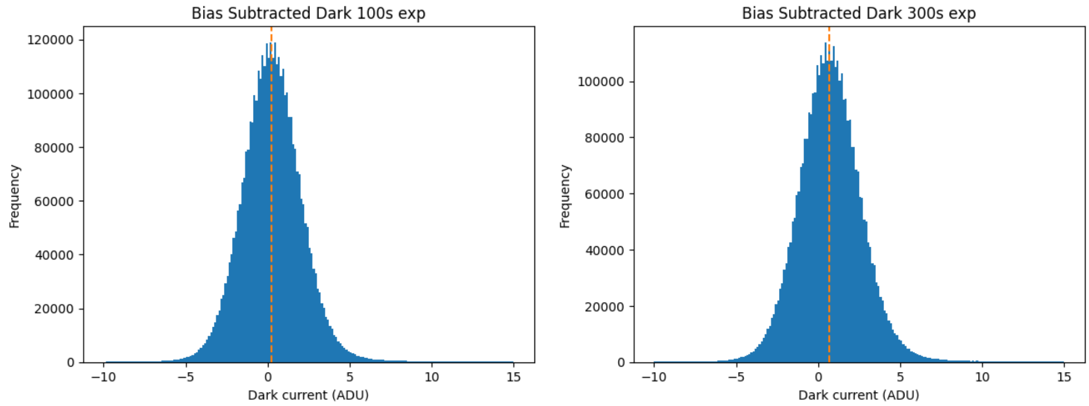


Figure 5.21: Dark signal distribution at 0 deg C and 100 s and 300 s exposure time respectively (Moravian c1x camera)

Dark Current Non-Uniformity

To study the dark current variation with respect to exposure time, one dark frame was taken at exposure time 100s, 200s, 300s, till 1000s at 0 degree Celsius with the Moravian c1x camera. These frames were then stacked, and every pixel, a count versus exposure time plot was taken. From the plot we could determine slope and it's intercept value, where the slope gives dark current and intercept denotes bias level. This way dark current of each pixel can be determined [13]. In Figure 5.22, the 2000*2000 central crop of the frame shows a master dark, which was obtained from the slope value for each pixel, it is to be noted that the values are stretched to certain limits to view the pixel-to-pixel variation, and the histogram shows the dark current distribution. The median value of the resulting dark current was 0.0025 ADU.

Using the intercept from the plots of each pixel, we can determine the bias level, and the resulting distribution of these intercept values for every pixel, results in a master bias, Figure 5.23. Here also the values are stretched and the histogram show the distribution of the bias values across the pixels, from where we can determine the median value of the bias level, which is 514 ADU.

For more analysis of the variation of dark current with respect to exposure time, single pixel-wise count was plotted with exposure time till a specified number of counts, and the variation the mean of the pixel data obtained from the crop was also plotted with respect to exposure time, which is shown in Figure 5.24

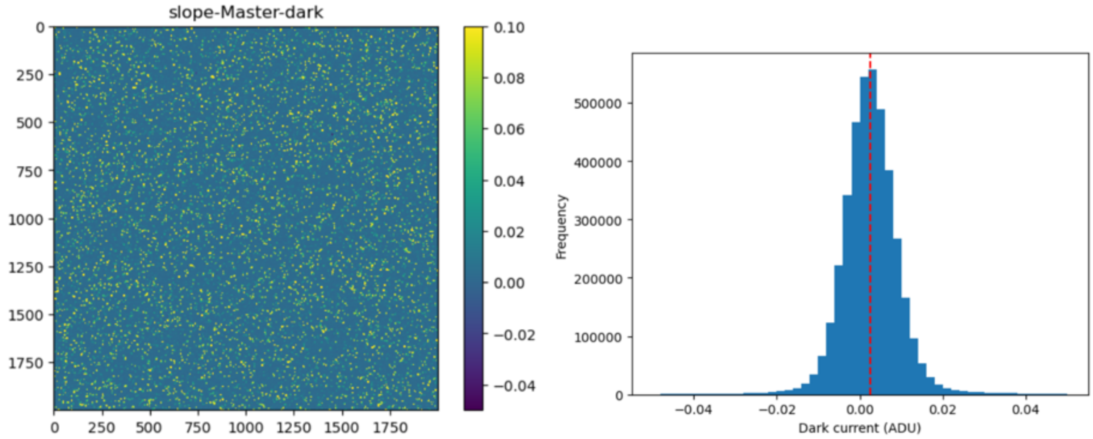


Figure 5.22: Dark current non-uniformity, and it's distribution (Moravian c1x camera)

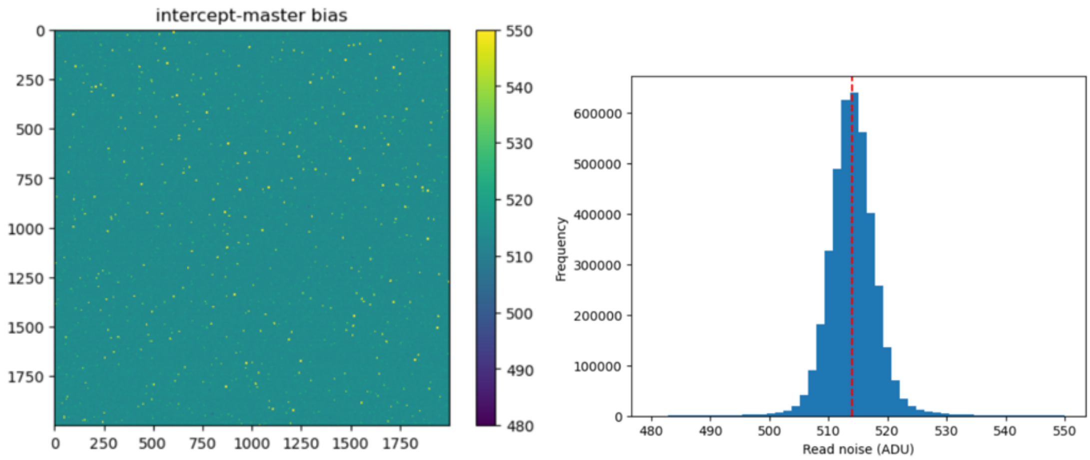


Figure 5.23: Master bias determined from the intercept values, and it's distribution (Moravian c1x camera)

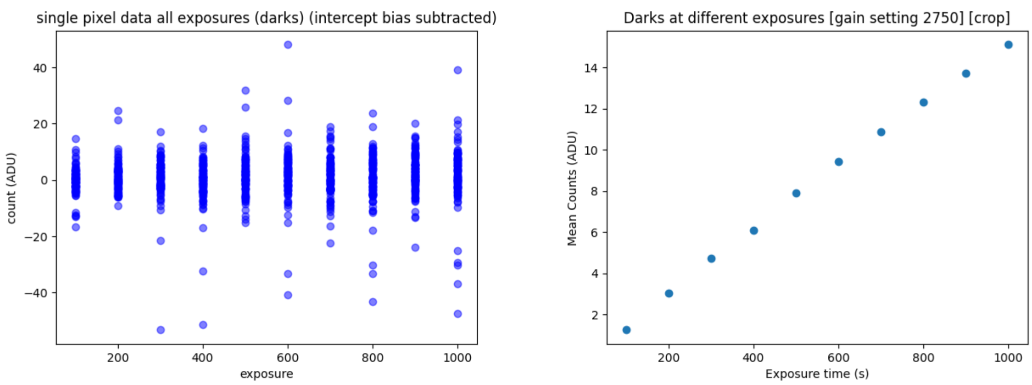


Figure 5.24: Single pixel-wise count with exposure time till a specified count and Variation of mean of cropped frame data with exposure time (Moravian c1x camera)

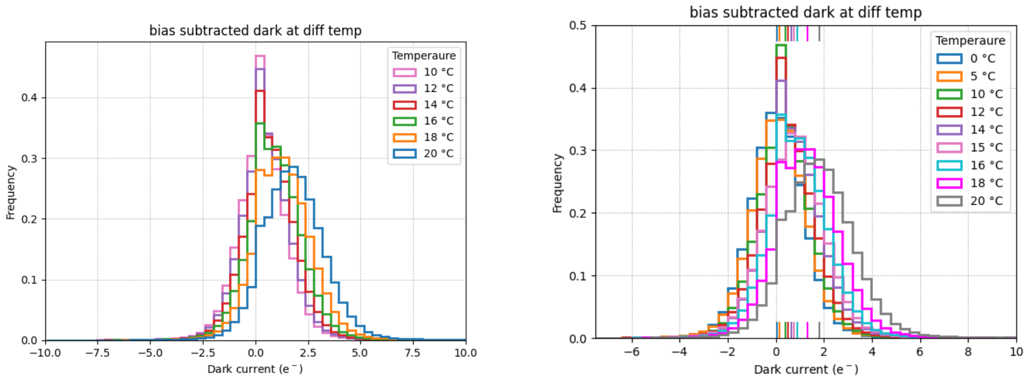


Figure 5.25: Darks versus temperature (Moravian c1x camera)

Dark current variation with temperature

Dark current depends on temperature, because as the temperature increases, thermal noise increases. To study this, five dark frames were taken at different temperatures (0°C, 5°C, 10°C, and 15°C) with 100-second exposures. A master bias (obtained from the intercept) was subtracted, the frames were median-stacked, and the dark current distribution was plotted against frequency. However, since the bias was taken at 0°C, the comparison may not be accurate. Additionally, a sudden shift in the distribution is observed in the comparative plot. Therefore, five dark frames were taken again at temperatures of 20°C, 18°C, 16°C, 14°C, 12°C, and 10°C, each with a 100-second exposure time. This time, two bias frames were taken before each exposure, and their average was subtracted from the corresponding dark frames (with respect to each temperature set). The resulting frames were then median-stacked to plot the dark current distribution for each temperature, as shown in the figure 5.25.

5.5.5 Shot Noise

Photon shot noise, is a temporal noise, which depends on the incoming photons, and not the image sensor. It occurs due to the particle nature of light[14]. To look for the presence of the photon shot noise, 100 flat frames at 0 °C and 1 second exposure time were taken. Figure 5.26 shows the observation of shot noise in Moravian c1x camera.

5.5.6 Photon Transfer Curve

To characterize an image sensor, the photon transfer curve (PTC) is a standard method used to study the sensor's response to uniform illumination. The PTC helps determine various characteristics such as the camera's conversion gain, digital response, and fixed pattern noise. It can also be used to identify the contributions of different noise sources.

A set of two uniformly illuminated frames was captured at increasing exposure

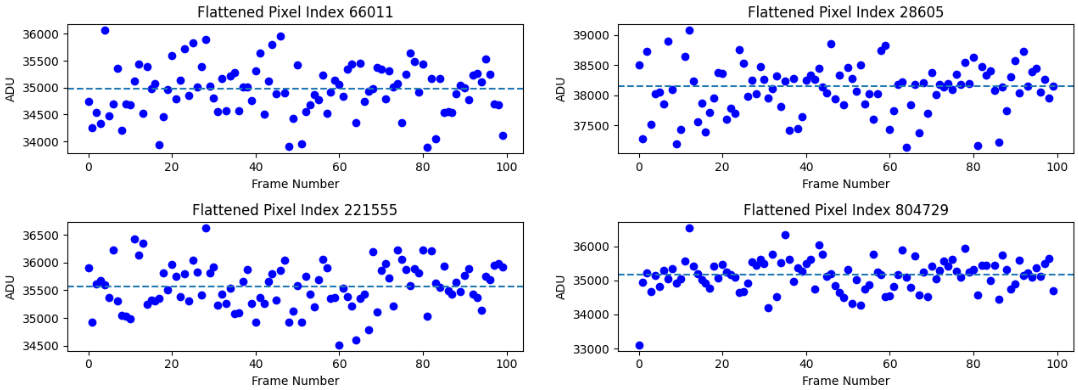


Figure 5.26: Shot noise (Moravian c1x camera)

times, starting from 0.05 seconds, then 0.1 seconds, 0.2 seconds, and so on, until saturation (set-up is shown in figure 5.27). Then these set of images were then cropped (central 2000 * 2000 crop) and then averaged. The master bias calculated initially for the Moravian c1x camera was then subtracted from averages flat frame. Then the variance was calculated and plotted against the pixel counts. The resulting plot is the photon transfer curve, where we observe that with the increase in the pixel counts, the variance also increases, but after reaching the full well capacity variance drops to zero, figure 5.28 .

From this PTC, we can determine, the ADU to electron conversion gain and the full well capacity of a pixel. It is to be noted that the frames were taken at the gain number of 2750. The gain calculated was 0.261 e-/ADU and the full well capacity was 15933.6 e-.

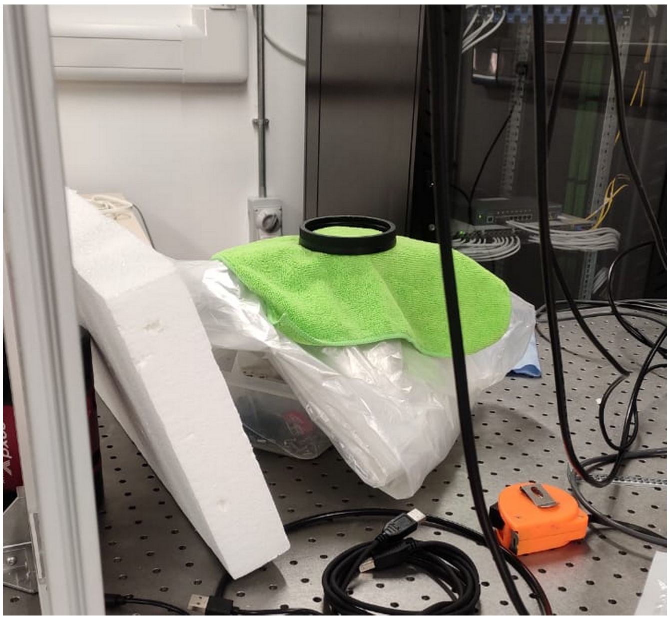


Figure 5.27: Photon Transfer Curve Set-up (Moravian c1x camera)

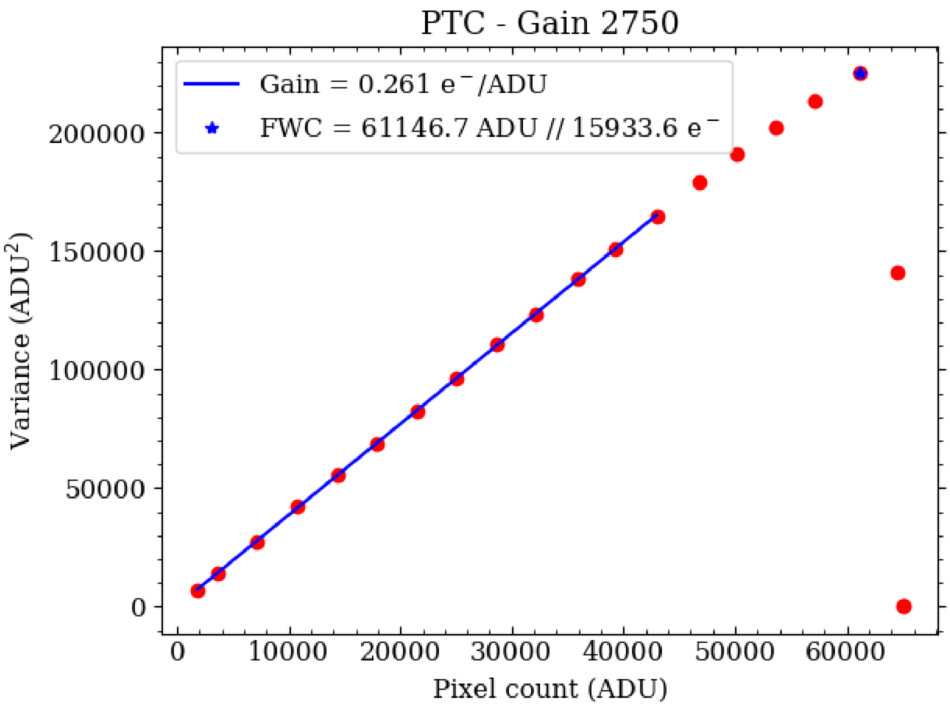


Figure 5.28: Photon Transfer Curve (Moravian c1x camera)

5.5.7 Linearity

Using the data used to plot the photon transfer curve, if we fit a linear curve at 5 percent to 95 percent of the full well capacity in signal versus exposure time plot, then the percentage variation from the linear fit determines the linearity error of the image sensor. The resulting linear error for the Moravian c1x camera was ± 0.315 percent.

Exposure	Average	Variance	Residuals
0.05	1813.05	7116.32	0.016
0.1	3566.38	13812.89	-0.049
0.2	7140.83	27520.16	-0.057
0.3	10820.74	42166.85	0.126
0.4	14341.18	55334.32	0.021
0.5	17860.77	68822.10	-0.086
0.6	21494.45	82772.87	0.013
0.7	25053.47	96469.39	-0.022
0.8	28686.15	110796.28	0.075
0.9	32143.93	123588.44	-0.144
1.0	35845.40	138184.95	0.078
1.1	39305.82	150937.19	-0.136
1.2	42972.62	164821.88	0.023
1.3	46798.13	179366.15	0.472
1.4	50084.75	190743.99	-0.059
1.5	53643.456	202267.09	-0.095
1.6	57187.889	213638.58	-0.157
1.7	61146.66	224891.72	0.532
1.8	64552.92	140915.36	0.219
1.9	65020.80	21.03	-5.433
2.0	65020.87	0.0	-11.9363

Table 5.1: Results from linearity test

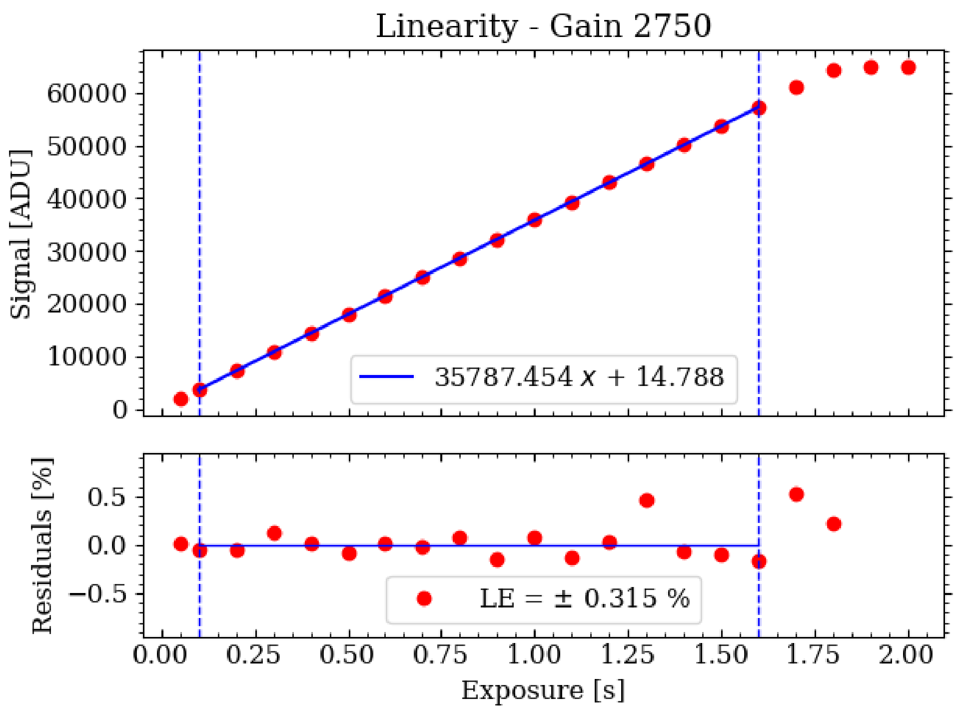


Figure 5.29: Linearity (Moravian c1x camera)

Chapter 6

Conclusion

This thesis has presented the development and evaluation of a cost-effective, scalable imaging system designed to support the detection of electromagnetic counterparts of astronomical events, particularly kilonovae. Through the Dragonfly Project, the concept of deploying an array of small telescopes integrated with custom-designed electronics offers a promising complement to large-scale observatories like LIGO-India [2]. The distributed configuration not only increases sky coverage but also enables responsive and affordable observation of transient celestial phenomena.

A major focus of the project involved the development of FPGA-based readout electronics using the Basys-3 development board [3]. Fundamental digital logic designs were implemented and later extended using the MicroBlaze soft-core processor and IP cores to enable real-time communication and data handling. Successful readout of linear image sensors marked a critical step toward full 2D sensor interfacing, demonstrating the system's potential for future expansion.

In parallel, extensive characterization of CMOS image sensors was conducted to assess their suitability for low-light astronomical imaging. Cameras employing the Sony IMX455 sensor such as QHY600, ATIK, and Moravian C1x, were analyzed for noise behavior, dynamic range, and sensitivity [15][16][17]. This analysis provided key insights into optimizing sensor performance and selecting appropriate components for future telescope nodes.

In conclusion, the thesis has laid a strong foundation for building a modular, efficient, and economically viable imaging platform. Future work will involve integrating full 2D sensor arrays, refining system calibration procedures, and deploying prototype units for field testing. Additionally, the scope can be expanded by exploring and performing detailed CCD sensor characterization to compare their performance with CMOS counterparts and assess their feasibility for specific astronomical applications. With continued development, this project has the potential to make meaningful contributions to multi-messenger astronomy and democratize access to deep-sky observation tools.

Bibliography

[1] A. Devices, "Ad4003/ad4007/ad4001: 16-bit, 8-channel, adc with spi interface datasheet," https://www.analog.com/media/en/technical-documentation/data-sheets/AD4003-4007-4001.pdf, 2023, revision C.

[2] B. P. Abbott, R. Abbott, T. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. Adya *et al.*, "Multi-messenger observations of a binary neutron star merger," *Astrophys. J. Lett*, vol. 848, no. 2, p. L12, 2017.

[3] D. Inc., "Basys 3 reference manual," https://digilent.com/reference/programmable-logic/basys-3/start, 2025, accessed: 20-May-2025.

[4] C. Chengtao, W. Mingyan, and L. Yanhua, "Fpga-based array ccd sensor drive system design and implementation," *Sensors & Transducers*, vol. 176, no. 8, pp. 49–57, 2014.

[5] J. Janesick, "Scientific charge-coupled devices, 83 spie press," *Bellingham, Washington*, 2001.

[6] M. R. Alarcon, J. Licandro, M. Serra-Ricart, E. Joven, V. Gaitan, and R. de Sousa, "Scientific cmos sensors in astronomy: Imx455 and imx411," *Publications of the Astronomical Society of the Pacific*, vol. 135, no. 1047, p. 055001, 2023.

[7] J. R. Janesick, *Photon transfer*. SPIE press, 2007, no. PUBDB-2021-04195.

[8] T. Holst and G. Lomheim, *CMOS/CCD Sensors*. JCD publishing, 2007.

[9] I. S. McLean *et al.*, *Electronic imaging in astronomy: detectors and instrumentation*. Springer, 2008, vol. 552.

[10] H. Photonics, "Advances in cmos image sensors," https://hub.hamamatsu.com/us/en/technical-notes/image-sensors/advances-in-cmos-image-sensors.html, 2025, accessed: 20-May-2025.

[11] M. Bigas, E. Cabruja, J. Forest, and J. Salvi, "Review of cmos image sensors," *Microelectronics journal*, vol. 37, no. 5, pp. 433–451, 2006.

[12] J. F. Wakerly, *Digital design: principles and practices, 4/E.* Pearson Education India, 2008.

[13] M. E. Borek, "Implementation of emva 1288 standard release 4.0 for characterization of image sensors," *Electronic Imaging*, vol. 35, pp. 1–6, 2023.

[14] J. Johnson, "Understanding noise and noise reduction in cmos imaging sensors," 2024.

[15] QHYCCD, "Qhy600: Full frame cmos camera datasheet," https://www.qhyccd.com/qhy600.html, 2023, accessed: May 20, 2025.

[16] M. Instruments, "C1x 61000: Cmos camera sensor datasheet," https://www.moravian.cz/en/c1x-61000, 2023, accessed: May 20, 2025.

[17] A. Cameras, "Apx60: Cmos camera sensor datasheet," https://www.atik-cameras.com/atik-apx60, 2023, accessed: May 20, 2025.