Software development for IUCAA SIDECAR Drive Electronic Controller (ISDEC) and development of scripts to automate ISDEC and detector tests

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Ketan Rikame

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

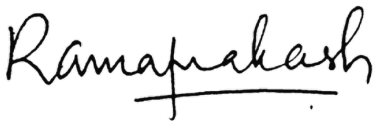
April, 2017

Supervisor: **Dr. A. N. Ramaprakash**

Certificate

This is to certify that this dissertation entitled Software development for IUCAA SIDECAR Drive Electronic Controller (ISDEC) and development of scripts to automate ISDEC and detector tests towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by **Ketan Rikame** at Inter-University Centre for Astronomy and Astrophysics, Pune, India under the supervision of **Dr. A. N. Ramaprakash**, Professor, IUCAA, Pune , during the academic year 2016-2017.

Ketan Rikame



Dr. A. N. Ramaprakash

Committee:

Dr. A. N. Ramaprakash

Dr. Prasad Subramanian

Dedicated to Mom and Dad,
who kept believing in me

Declaration

I hereby declare that the matter embodied in the report entitled Software development for IUCAA SIDECAR Drive Electronic Controller (ISDEC) and development of scripts to automate ISDEC and detector tests are the results of the work carried out by me at the IUCAA, Pune, under the supervision of **Dr. A. N. Ramaprakash** and the same has not been submitted elsewhere for any other degree.

Ketan Rikame

Dr. A. N. Ramaprakash

Acknowledgements

This thesis would not have been possible without the inspiration and support of so many wonderful individuals, it is a humbling experience to acknowledge all of them.
I would like to thank Dr. A. N. Ramaprakash for giving me an opportunity to work in the IUCAA instrumentation lab. I would also like to thank him and Mr. Mahesh Burse for his constant guidance and invaluable support. I take this opportunity to thank my project partner, Mr. Sakya Sinha and all other lab members for their encouragement. I would also like to thank Dr. Prasad Subramanian for his guidance and support.
I would like to extend my thanks to all my IISER and IUCAA colleagues for being there when I needed them the most.
Last but not the least, to mom and dad who supported me throughout my life and kept believing in me. Thank you.

Abstract

The development in the field of astronomy and astrophysics heavily relies on quality of the data obtained by various observatories. The quality of the data obtained by these observatories depend upon the quality of the imaging sensor used for observation as well as electronics involved during data acquisition from these sensors. Teledyne make HAWAII-2RG detectors are used by many observatories. IUCAA SIDECAR Drive Electronic Controller (ISDEC) is an electronic card developed to configure SIDECAR ASIC and to acquire data form HAWAII-2RG detectors. ISDEC is designed with keeping general astronomical requirements in mind like Linux compatibility and multiextension FITS output. The current version of ISDEC can not support fast mode readout for H2RG detectors. The advanced version of ISDEC (i.e. ISDEC 3.0) is under development which will support fast mode readout using USB 3.0 interface. This version has been successfully checked to support data throughput up-to 160 MBPS for original ASIC data and up-to 320 MBPS for simulation data. The work is currently in progress to improve the throughput for original ASIC data up-to 320 MBPS which is necessary for fast mode readout. This thesis is about application software development for ISDEC 3.0 and about automation of ISDEC based controller tests necessary to ensure the functionality of the detector and it's optimum performance. Application software is written in C++ and automated tests are written in Python. Both scripts are developed in Linux environment and have command line interface.

Keywords - ISDEC, HAWAII-2RG, SIDECAR ASIC, Fast Mode Readout, USB 3.0

Contents

Chapter 1

Introduction

Imaging sensors and imaging techniques have played very essential role in the development of astronomy and astrophysics. Better imaging techniques provide better quality data to the astronomical community and hence help in better understanding of the sky.
Teledyne's HAWAII-2RG detectors are used by many ground as well as space based observatories for observations in visible and infrared domains. This detector kit consists of SIDECAR ASIC which is single chip controller for the detector array and can be programmed to get the readout from the detector in variety of modes.
H2RG detectors are placed in cryogenic kits to maintain low temperature which is essential for detector operation. SIDECAR ASIC is also placed inside the cryogenics to avoid use of long cables between detector output and ASIC which may dampen or corrupt the signal.
SIDECAR ASIC needs to be programmed in order to get readout from the detector, reset the detector and to configure the detector. However, SIDECAR ASIC cannot communicate directly with host PC and Therefore external hardware is necessary to program SIDECAR ASIC, and to setup communication between PC and ASIC to obtain detector readout.
IUCAA developed an electronic board i.e. ISDEC few years ago to serve the purpose. It offers detector operation in variety of modes. This board is already being used by different observatories for their instruments.
This board uses USB 2.0 interface for communication between ASIC and host PC. This board is able to acquire data from form the detector in slow mode (i.e. up-to 500KS/sec., 1,4,32 output mode). However due to intrinsic limitation of USB 2.0 bandwidth, readout acquisition in fast mode i.e. at 5MHz frequency with 32 output mode is not possible.

ISDEC 2.0 system has been installed in various observatories such as, RSS-NIR, a prime focus facility instrument for the 11 meter SALT, RoboAO at IGO(IUCAA), CIRCE and MIRADAS at GTC and LMIRcam on LBT.
To enable fast readout mode, a new version of ISDEC with USB 3.0 interface with added functionality is currently being developed at IUCAA.
This thesis is about application software development for ISDEC 3.0 and development of automated ISDEC based controller tests to ensure functionality and performance of the device in various modes and configurations.
It is necessary to get an overview of an ISDEC system in order to understand it's software and automated scripts. Following sections explain about ISDEC architecture, it's features and abilities, performance results of ISDEC 2.0 and expectations from ISDEC 3.0.

Figure 1.1: Front view of ISDEC 2.0

Image Ref: *ISDEC-2 and ISDEC-3 controllers for HAWAII detectors*, SPIE 9915

1.1 ISDEC - an overview

Figure 1.1 and and figure 1.2 show actual images of ISDEC 2.0 setup. It can be seen from the images that board is built around Xilinx spartan-3E FPGA. All the voltages necessary to drive SIDECAR ASIC are produced on board itself and then transferred to to ASIC via flex cable. The entire board can be operated using single external 5 V power supply. The board can operate operate at -40° Celsius and hence can be put with cryogenics of the detector.

Figure 1.2: Back view of ISDEC 2.0

Image Ref: *ISDEC-2 and ISDEC-3 controllers for HAWAII detectors*, SPIE 9915

1.2 System architecture

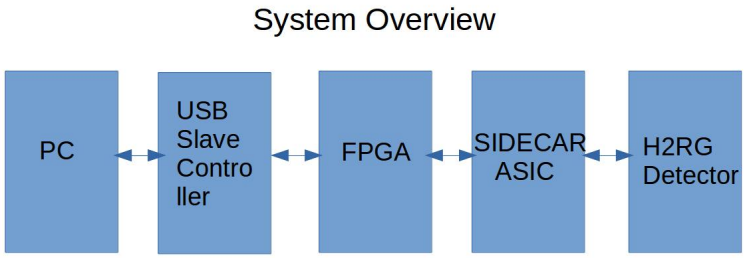


Figure 1.3: Architecture of an ISDEC system

Figure 1.3 shows schematic architecture of an ISDEC system. The top most part of the architecture is a host PC which contains application software of the device. The application software does the job of configuring individual components, providing the readout data to the user as per input instructions given. USB slave controller provides USB interface between FPGA and PC. It acts as a slave controller and allows direct communication between FPGA and PC. The middle layer architecture contains FPGA. It is programmed to receive input commands from host PC via USB slave micro-controller and communicate them to SIDECAR

ASIC. FPGA firmware is also programmed to read ASIC registers and readout data and communicate it to PC. SIDECAR ASIC is connected to the actual detector and collects the readout from the detector as per inputs provided by the application software.
The application software stands at the top of the architecture and allows user to operate the detector and get the readout via this hierarchical flow of information.

1.3 ISDEC 2.0 - Capabilities and features

ISDEC 2.0 has large FIFO buffers which allows on chip data buffering. Moreover using techniques like asynchronous USB readout, ISDEC 2.0 achieves sustained throughput of around 30 MBytes/sec. This throughput ability allows it to be able to read data with sampling rates ranging from 50 KHz to 500 KHz for all detector output modes.
According to H2RG datasheet, partial frame readout can only be done for single output mode. In ISDEC, user can read strip of image (of fixed width i.e. 2048 pixels but variable height) using 32 output mode, which is much faster compared to single mode operation.
ISDEC also offers fast reset mode using which full array or part of it can be reset very quickly.
Apart from usual full and partial frame readouts, ISDEC also supports multiple frame readout with customizable inputs. This ability to read multiple frames is especially useful in wavefront sensing applications.
Appendix B contains actual readout images for various detector mode operations.

1.4 ISDEC 3.0

ISDEC 2.0 is designed by keeping general astronomy requirements in mind. However due to limitations of USB 2.0 interface itself, it can not be used for higher data sampling rates. The fast mode of H2RG detectors samples data at 5 MHz sampling rate. (5 M pixel/sec, 32 outputs) For ISDEC to be able to sample data at such rate, throughput of the order of 300 MBytes/sec is necessary. ISDEC 3.0 is being developed to meet these requirements.
ISDEC 3.0 is built around Xilinx make artix-7 series FPGA. Ideally data rates using USB 3.0 interface are sufficient to operate it on fast mode readout. ISDEC 3.0 is capable of driving

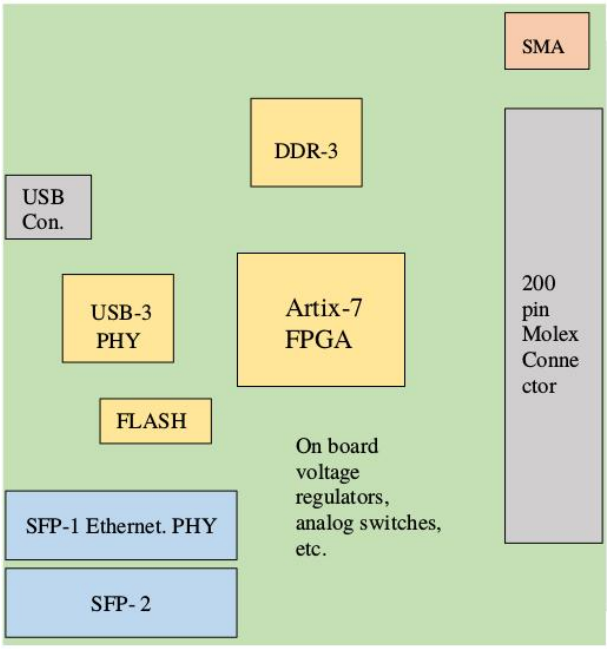


Figure 1.4: ISDEC 3.0 : main component layout

Image Ref: *ISDEC-2 and ISDEC-3 controllers for HAWAII detectors*, SPIE 9915

two SIDECAR ASICs and hence two detectors simultaneously. It also has provision of using Ethernet interface instead of USB interface for slow mode operation. It has on-chip DDR-3 memory which allows temporary storage of data on board. There are two SFP's on board, one for Ethernet interfacing and other for communication between two ISDEC chips. Similar to ISDEC 2.0, this board is also powered by external 5 V DC supply.

There is a need of handle over complete device which can control and configure individual components, get input parameters from the user, communicate them to appropriate components, get the detector readout as per given input, and communicate it back to the user. Application software provides that handle to the user. The challenge in developing application software is that it should provide sufficient freedom of choosing the parameters to the user but also it should be user-friendly. Also it should have ability to manage very high data transfer rates and should be able to process that data simultaneously. This thesis is about application software development for ISDEC 3.0.
Once the device is commissioned, it is necessary to check it's functionality in all parameter space which is almost impossible manually due to very large parameter space of ISDEC 3.0. There is a need for an automated program, which can go through sufficiently large parameter

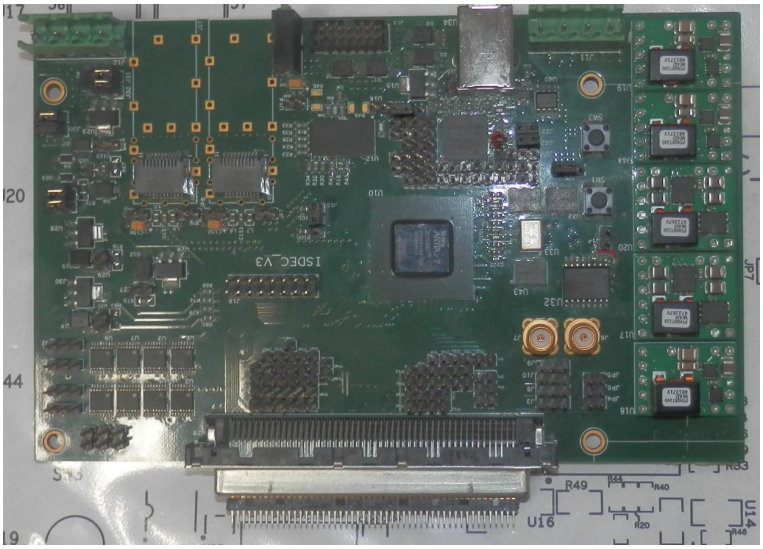


Figure 1.5: ISDEC 3.0 : front view

space and check the readouts obtained for these parameters and ensure that the device is working fine. To get the best results from the device, one should use optimized values of parameters. A scripts that determines optimum values of parameters (such as bias voltages, exposure time etc.) to get best performance for a particular detector mode and condition can also be developed.

Scripts for automated testing and to find such optimized values of parameters were developed during the duration of the thesis. They proved very useful in characterization process of the detector. Also these scripts help user to save large amount of time in checking device functionality and get most efficient readout using optimized parameters.

Chapter 2

ISDEC components

ISDEC system is composed of various components. This chapter provides overview of each of these components which is essential for the better understanding of ISDEC system as a whole and it's application.

2.1 Infrared Detector(HAWAII-2RG)

HAWAII-2RG is the infrared detector on which fully commissioned ISDEC setup will be installed. HAWAII-2RG stands for HgCdTe Astronomy Wide Area Infrared Imager with 2K x 2K resolution, Reference pixels and Guide mode.
This multiplexer can be operated at frequencies ranging from 100KHz (slow mode) till 5MHz (fast mode). It features various output modes such as single output mode with very less power consumption (less than 0.5 mW) and 32 output mode which allows very high frame rate.
Figure 2.1 shows block structure of the multiplexer. It contains 2048 x 2048 array of pixels along with vertical scanner to the left of array and horizontal scanner and column buffers on the top. All the I/O pads are on very top of the multiplexer providing close buttability along sides. The 2k x 2k array of pixels consists of 4 rows and 4 columns of reference pixels on the edges. Therefore effective pixels array for actual image formation is of the size 2040 x 2040. Provision reference pixels is extremely useful in determining electronic noise of the detector.

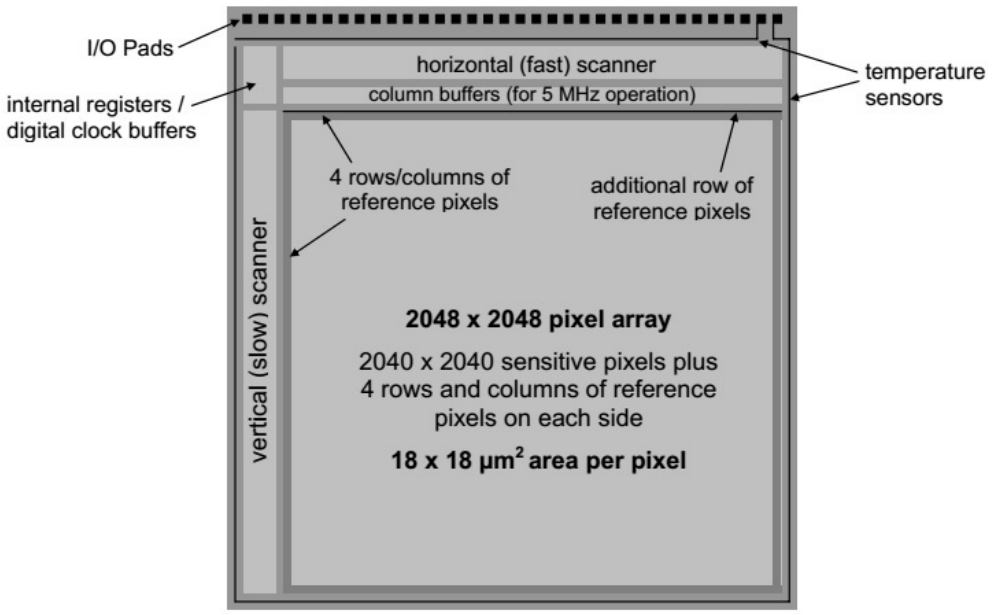


Figure 2.1: block diagram of HAWAII-2RG

Image Ref: HAWAII-2RG technical manual

Figure 2.1 shows architecture of unit cell (pixel) of the detector. This unit cell is optimised for low voltage and high dynamic range readout. It enables single pixels reset. From the diagram it can be seen that both control lines (horizontal and vertical) have control over reset switch and it closes only when both lines have reset signal. This allows pixels to reset without affecting neighbouring pixels. The actual photo detector is bump bonded to the unit cell. It produces voltage corresponding to the amount electromagnetic radiation incident on it. This voltage is read by I/O pads via read control lines.

Mercury cadmium telluride (HgCdTe) is the photo diode material used. It is an alloy of cadmium telluride (CdTe) and mercury telluride (HgTe). CdTe is an semiconductor material with band gap of 1.5 eV. HgTe has no band gap. Thus by changing concentration of both materials, one can obtain tunable band between 0.0 eV to 1.5 eV.

There are two possible atmospheric windows for infrared detection of wavelength ranging from 3-5 μm and 8-12 μm. HgCdTe is the only alloy capable of detecting in both windows by changing concentrations of it's constituents.

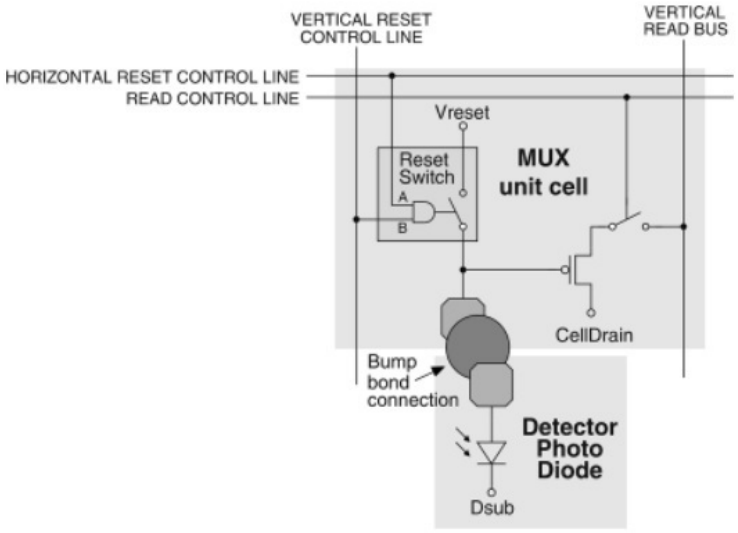


Figure 2.2: Architecture of unit HAWAII-2RG cell

Image Ref: HAWAII-2RG technical manual

2.2 SIDECAR ASIC

SIDECAR ASIC stands for System Image, Digitizing, Enhancing, Controlling, and Retrieving Application Specific Integrated Circuit. It manages all the aspects of detector operation and output digitization.
Main communication interface of the SIDECAR ASIC is made of two operation modes, one for configuration of the device and other for science data acquisition. To check the data integrity for large packets of data, ASIC implements checks such as checksum and CRC.

2.3 Cypress EZ-USB FX3 micro-controller

For USB interfacing of ISDEC, cypress Ez-USB FX3 micro-controller is used. It is highly integrated and highly flexible peripheral controller which allows USB 3.0 functionality to any system. It has fully programmable interface which can connect to any hardware such as FPGA, ASIC etc. Because of it's integrated inter-port DMA architecture, data transfer speeds greater than 400 MB/sec can be obtained.

Figure 2.3 shows image of micro-controller explorer kit used for understanding the work-

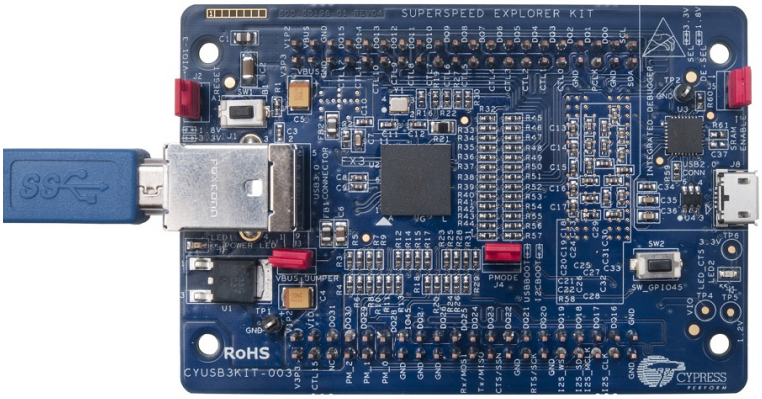


Figure 2.3: Super Speed Explorer kit

Image Ref: Getting Started with FX3 SDK, Cypress documentation

ing of micro-controller and to know the possible data throughput rates with a Linux OS . In ISDEC, this micro-controller acts as a slave device and allows direct communication between host computer and FPGA.

2.4 Field-Programmable Gate Array (FPGA)

FPGA's are fully configurable semiconductor devices. They are made of configurable logic blocks which are connected by programmable interconnects. Unlike application specific integrated circuits (ASIC), FPGAs can be configured after manufacturing. This programmable nature of FPGA makes it suitable for wide range of application.
FPGA forms the middle layer in ISDEC system architecture. (refer section 1.2) It's job is to drive SIDECAR ASIC as per given inputs and communicate the readout data to PC.
ISDEC 2.0 consists of Xilinx spartan-3E FPGA while ISDEC 3.0 is built around Xilinx artix-7 series FPGA.

Chapter 3

Tools and techniques

Many tools and techniques are necessary for the development of ISDEC. Many hardware skills such as operating logic analyzers, probing circuit boards to analyze data transmission and software skills such as C programming, programming micro-controllers and other hardware, USB communication etc. were learned during the course of this project.
This chapter describes some of these important tools used while developing the application software for ISDEC 3.0.

3.1 USB communication and Libusb library

Libusb is an open source C library used by developers for USB communication with USB hardware. The chapter provides information about USB communication and how Libusb library operates to serve the purpose. There are two ways in which USB communication can be done.
1. **Synchronous transfer**
2. **Asynchronous transfer**
Every USB device has two types of buffers called input endpoint and output endpoint. Any data written on input endpoint buffer of the USB is taken as input command by the USB device. Output of theUSB device is communicated to PC via output endpoint.
Libusb library provides very efficient application programming interface (API) which contains functions to communicate with these endpoints.

There can be many USB devices attached to PC at once. To ensure the communication with desired device, each device has it's own ID. Libusb functions allow user to establish communication with appropriate device using appropriate ID. This process involves various steps such as opening USB device, claiming the device interface, setting the alternate interface, setting the priority of the process etc.
Once this initial communication is established, one can proceed to actual data transfer.
Data transfer happens in two steps. while reading the data, request needs to be placed at the out endpoint of the device and when the data come to the endpoint, it is read as per request. Similarly while writing the data, first it is written on in endpoint of the device and when the device is ready, it acknowledges and receives the data.
The process can subdivided further as following.
1. **Allocation of the transfer:** Transfer is defined at this step.
2. **Filling the transfer:** This step determines the parameters of the transfer request such as amount of data requested, timeout for the transfer, endpoint ID, calling of callback function etc.
3. **Submission of the transfer:** Once transfer request and it's properties are defined, transfer request is submitted. After submission, data is obtained when it reaches endpoint as per transfer request.
4. **Freeing the transfer:** After transfer is complete, memory allocated for the transfer is freed.

All the steps mentioned above can combined together in the form of just single transfer request. This type of transfer where all the sub steps in the transfer happen altogether is called **synchronous transfer**. Synchronous transfers are very easy to implement as one does not need to worry about individual sub steps.
On the other hand, in **asynchronous transfer** all these sub steps are defined separately. This makes the program very complicated but this added complexity improves the efficiency the data transfer. Previously used transfers can be resubmitted and hence memory is reused unlike synchronous transfer where memory buffer is defined at once and cannot be reused until the transfer is complete. Transfers can be submitted way before the data arrival which reduces chance of data loss. Once the data is received, separate callback function is used to process data. This function runs in another thread of the processor, and hence next data can read simultaneously while previous data is being processed. This ability to multi-thread makes asynchronous transfer very fast and efficient compared to synchronous transfer.

Libusb library in C provides application program interface for both type of transfers. USB communication part in application software of ISDEC is written using Libusb. Configuration part of the software uses synchronous data transfer because it does not requires very high data transfer speeds. Also the code for synchronous transfer is less complicated to understand. However in case of actual image data transfer from the device, ASIC sends large chunks of data at very high speed towards USB hardware. Hence actual image acquisition part of the software uses asynchronous transfer.

3.2 cFITSIO

This library is used for FITS (Flexible Image Transport System) image creation in exposure program.
It was initially developed by high energy astrophysics science archive research center at NASA for visualizing and analysing astronomical data in image format. This library is developed in ANSI C and provides simple yet powerful interface for FITS image creation which insulates user from complicated details of actual FITS format.

3.3 Cyclic redundancy check (CRC)

Cyclic redundancy check is an algorithm implemented in digital systems to detect accidental changes in data while it is being transmitted through the system. This prevents data corruption which is essential for system to work.
ISDEC implements CRC algorithm while transferring large data packets.

3.3.1 Theory

CRC's are particularly useful in detecting burst errors in data transmission. CRC code is specified by a generator polynomial. Mathematically this check is based on the division of

polynomials in the finite galois field of two elements, either zero or one.

$$M(x).x^n = Q(x).G(x) + R(x)$$

Here, M(x) is the polynomial representing data and G(x) is generator polynomial of the order n. Q(x) is the quotient polynomial and R(x) is the remiander polynomial which forms CRC. $M(x).x^n$ is nothing but n zeros added to data bits.
Now using modulo operation in $GF(2)$ it can be stated that,

$$R(x) = M(x).x^n mod G(x)$$

While transmitting the data, it is transmitted along with CRC which is n bits of R(x). Receiver only checks for bits of R(x) and assumes data to be correct if CRC matches.
Error detection ability of CRC depends on factors such as degree of generator polynomial and structure of the polynomial. There are certain limitations to this technique such as not being able to detect missing leading zeros in the data or prepending zeros in the data. This is because CRC is based on division of polynomials.

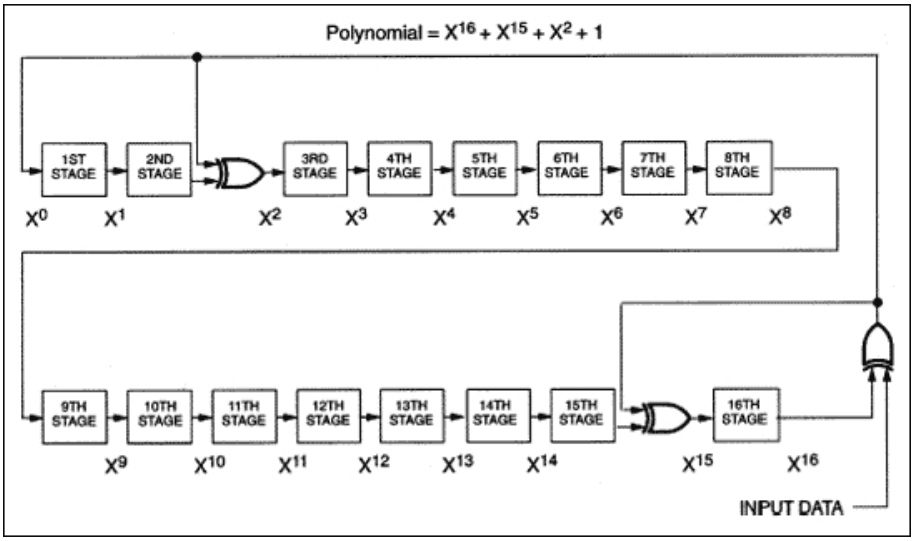


Figure 3.1: CRC-16 Hardware Implementation

Image Ref : https://www.maximintegrated.com/en/images/appnotes/27/27Fig04.gif

3.3.2 CRC-16

ISDEC uses 16 bit CRC for error detection in data transmission. $x^{16} + x^{15} + x^2 + 1$ is the generator polynomial i.e. $G(x)$ used for CRC calculation. The process of division of polynomials is achieved using Exclusive-OR logic gate and shift registers in the hardware. Figure 3.3.1 demonstrates the hardware implementation of CRC algorithm.

3.3.3 CRC implementation in ISDEC

In ISDEC 2.0, for a given packet of data, CRC used to be calculated in FPGA. After a packet transfer from FPGA to ASIC, ASIC would generate CRC and match it with the CRC calculated by FPGA. Both CRC values should match for transfer to be successful.
In ISDEC 3.0, CRC calculation algorithm is implemented in software itself. Calculated CRC is then transferred along with the packet of data. This is done to reduce load of processes happening at FPGA. CRC is implemented only for large packets of data. (for packet size more than 32-bit)

Chapter 4

Automation of ISDEC based controller tests

Parameter space for operation of ISDEC is very large. Once any device is installed, it is necessary to check it's functionality in all possibles modes and scenarios. It is also useful to optimize the device to obtain best possible results. Parameter space of ISDEC being very large, it is almost impossible to check it's complete functionality manually. This chapter is about ISDEC based controller tests which were implemented to ensure the functionality of the device and it's optimized performance.
It is essential to understand the nature of ISDEC 2.0 application software in order to understand functionality of these tests. Following sections gives overview of ISDEC 2.0 application software.

4.1 ISDEC 2.0 : Application software overview

Application software does the job of communicating input from the user to the ASIC and readout from the detector to the user. User provides input to the software via parameter files. Before getting ready for the exposure, all the components need to be configured. One part of the software does the job of configuring the system. Other part does the actual job of image acquisition as per inputs provided by parameter files. The structure of the software can be broken down to following three parts.

4.1.1 Parameter files

As already mentioned, parameter space for ISDEC is very large. There are five different parameter files which contain information regarding various aspects of the detector configuration. These are text files which contain values of parameters that user can modify in order to change the input conditions. (refer appendix C for parameter file images.)

Information related to image exposure such as number of frames per exposure, frame size, exposure time etc. is stored in exposure parameter file. This file also contains information such as file name and path to other parameter files.
Engineering parameter file contains parameters such as detector readout mode, system gain values, detector temperature etc.
Configuration parameter file has information about various bias voltage values such as main reference voltage, reset voltage and other bias voltages.
Reset parameter file has parameters regarding reset-scheme of the detector.
And auxiliary parameter file stores additional observation and telescope specific information about image.

4.1.2 Configure program

ISDEC system consists of USB slave micro-controller ,FPGA and SIDECAR ASIC. All three hardware components need to be programmed before exposure. Also system has to be turned on with appropriate voltages to each components. Configure program turns on the system and installs the firmwares of each components in order to make it ready for the exposure.

4.1.3 Exposure program

Once the system is turned on and is configured, it is ready for exposure and collect readout data. This program takes the parameters from parameter file as input and provides appropriate detector readout as output. This program does the actual job of image acquisition and produces FITS images of detector readout as output.

4.2 Automated tests

Following are the tests designed to ensure functionality and optimized performance of the detector. These tests are already implemented and tested for ISDEC 2.0 setup. Since ISDEC 3.0 setup is still under development, they are yet to be implemented in ISDEC 3.0. The test results discussed in following sections are for ISDEC 2.0 readout. Since the software structure of the both systems (ISDEC 2.0 and ISDEC 3.0) is very similar, same tests can be implemented on ISDEC 3.0 once it is fully developed and commissioned.
Parameters regarding image acquisition are stored in parameter files. General approach to design these tests is to shortlist the parameters to be tested and collect the readout data for all the possible values of these parameters keeping all other parameters at default value. Then check through readout data to detect errors.

4.2.1 Test module that checks parameters related to frame size and frame number

ISDEC provides large range of customizable options when it comes to exposure related parameters. Arbitrary frame size can be selected from detector array for image acquisition. User also has choice of deciding number of frames per exposure and sequence of read and reset exposures.
It is necessary to check that image obtained from the detector is in accordance with inputs given. In case of partial frame readout, it is necessary to check if frame size is as expected and also frame corresponds to appropriate window on the detector.
These parameters are stored in exposure parameter file. This program obtains readout images by changing parameters from exposure parameter file and matches it with the expected template. It also creates the log of the results and produces a report if detector output is not in accordance with expected template.
Following image is the snapshot of the result log while testing partial frame readout. x and y in the image represent the co-ordinates of the pixels on the detector array. As it can be seen, script goes through different frame configurations, checks the readout obtained and produces status. All other parameters are set to default value during this process. Other exposure related parameters are also checked in similar way.

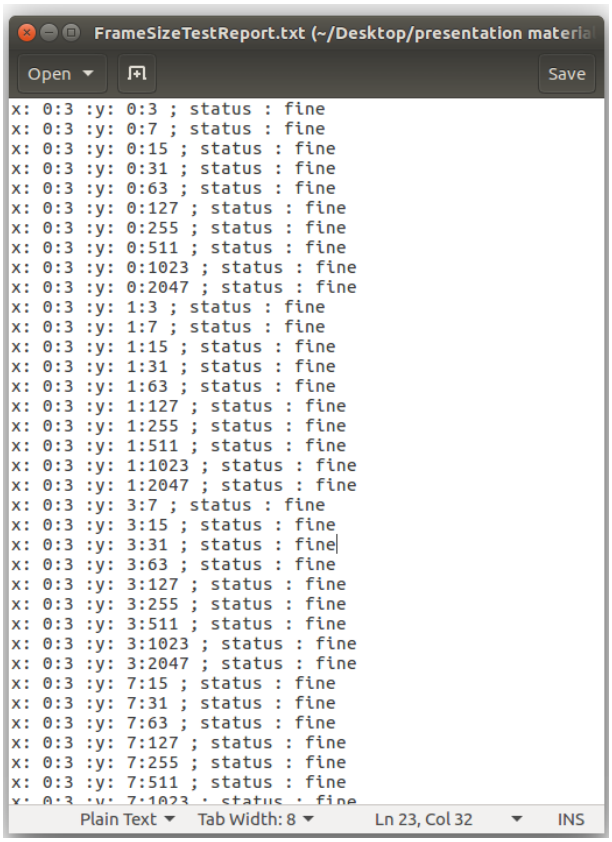


Figure 4.1: Snapshot of result log while testing partial frame readout

4.2.2 Test module to check parameters related to detector output mode and type

Engineering parameter file contains parameters related to detector output mode and output type. H2RG detectors can provide readout in three possible modes. 1-output, 4-output and 32-output mode. In 32 output mode, 32 pixels are simultaneously read by 32 voltage sensors. Similarly for 4-output mode and 1-output mode, 4 pixels and 1 pixels is read at a time respectively. This means that 32-output mode 32 times faster that 1-output mode. However using 32-output mode and 4-output mode, only strip of image or full frame image can be read unlike 1-output mode where partial frame with arbitrary co-ordinates can be read. Also in these modes, more than one voltage sensors read the image. Each voltage sensor has it's own electronic noise. Hence while processing the image, user has to deal with noise of all the sensors involved during readout. All three modes have their own advantages and disadvantages, and it is necessary that appropriate mode is selected during the readout.

The output from each pixel can be manipulated according to the input conditions, such as single ended inputs, differential inputs, grounded inputs. In grounded input mode, as the inputs being grounded, readout obtained represent the electronic noise of the detector. Appropriate output type needs to selected before readout, such as single ended or differential type for actual image readout and grounded input mode for analysing system noise.
This test module checks that all detector output modes as well as output types are working and appropriate readout is obtained for inputs provided from engineering parameter file.

4.2.3 Tests to find optimum system gain parameters

The quality of the detector readout image depends upon values of bias voltages, such as main reference voltage, reset voltage etc. These bias voltage parameters are stored in configuration parameter file.
Unlike test modules discussed above which ensure functionality of the detector, this module is about optimizing readout quality.
The output obtained at each pixel is basically voltage produced due to interaction of electromagnetic radiation with detector material. This voltage is then amplified via op-amp before it passes through detector. Hence final amplified voltage (which is then reflected as intensity of pixel) depends on reference voltage and reset voltage of the op-amp.
The voltage sensors have range of detection in which they can detect pixel output voltages. Also the analog to digital converters (ADC) which convert analog voltage signal to digital signal also have range of operation.
The pixel count obtained during readout can get saturated either because saturation of photo detector material itself or because the voltage produced is outside the range of voltage sensors and ADCs. If the pixel values in the image are saturated then one can't distinguish between two objects of different luminosity due to lack of contrast. To improve the quality of the image, there should no saturation of pixel value at any level.
As already mentioned, the output voltage of the pixel depends on the op-amp reference voltage. Greater the difference between voltage produced at the photo diode and reference voltage, greater is the amplification of the output voltage. The reference voltage should be such that for no value photo diode voltage, output voltage becomes saturated. Also difference should be large enough to produce image with sufficient contrast.

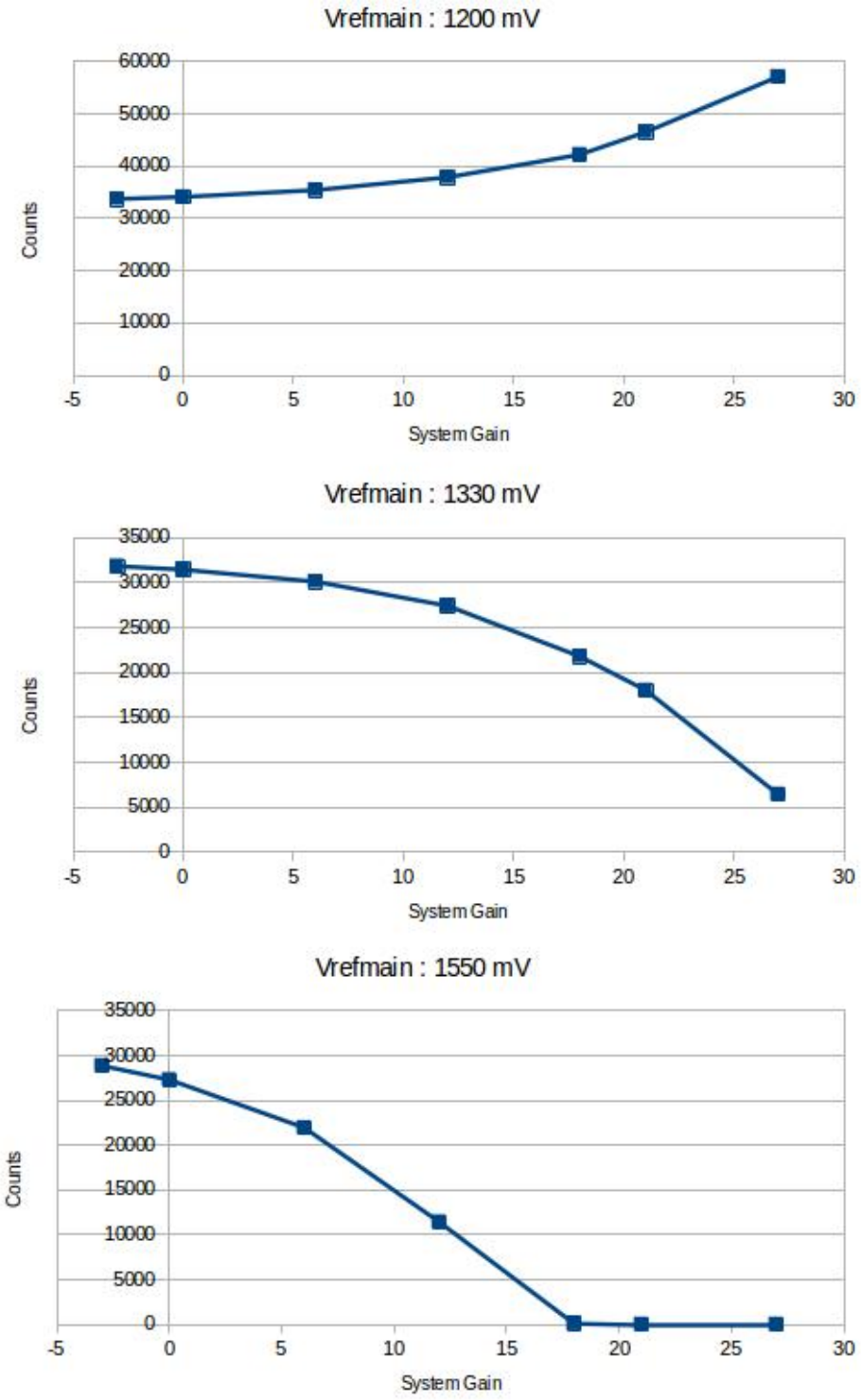


Figure 4.2: Variation of average detector output count with system gain and reference voltage

Op-amp at each pixel just amplifies the difference between reference voltage and pixel voltage. Hence for the case where pixel voltage is smaller than the reference voltage, this difference is negative and in turn pixel output value is below the mean value. Similarly in the case where pixel voltage is greater than the reference voltage, pixel output value is above the mean value.

This program obtains detector readout for various reference voltage values and for various system gain values. It then analyzes the readout and finds out parameter values for which, detector readout is not saturated. Image 4.3 shows variation of pixel count with system gain for three different reference voltage values. It can be seen from the image that for the reference voltage value 1550 mV, the average pixel count value saturates after gain value 18 dB. This clearly indicates that keeping reference voltage value as 1550 mV or above that is not advisable. It can also be seen that pixel count values do not saturate when reference voltage values are 1200 mV and 1330 mV. Hence these reference voltage values can be used while obtaining detector readout. This program provides the range of reference voltage values and corresponding system gain values for which image does not saturate and hence good quality image is obtained.

Voltage produced at photo diode can be replaced with known fixed voltage value called reset voltage. Image obtained in this configuration is called reset frame. Reset frames are very useful in determining detector noise. Usually read frames and reset frames are obtained altogether. Actual image i.e. image without readout noise can be obtained by comparing read frame and reset frame. It is necessary that reset frames do not saturate as well in order to get the correct estimate of readout noise. First program provides optimum range of reference voltage values. For those reference voltages, optimum range of reset voltage values can be determined.

Image 4.3 shows variation of average detector output count with system gain(in dB) and reset voltage for fixed reference voltage and other parameters. This program provides range of reset voltage values and corresponding system gain values for which reset frame count does not saturate.

These programs altogether provide range of parameter space in which system should be operated to get best possible readout from the detector.

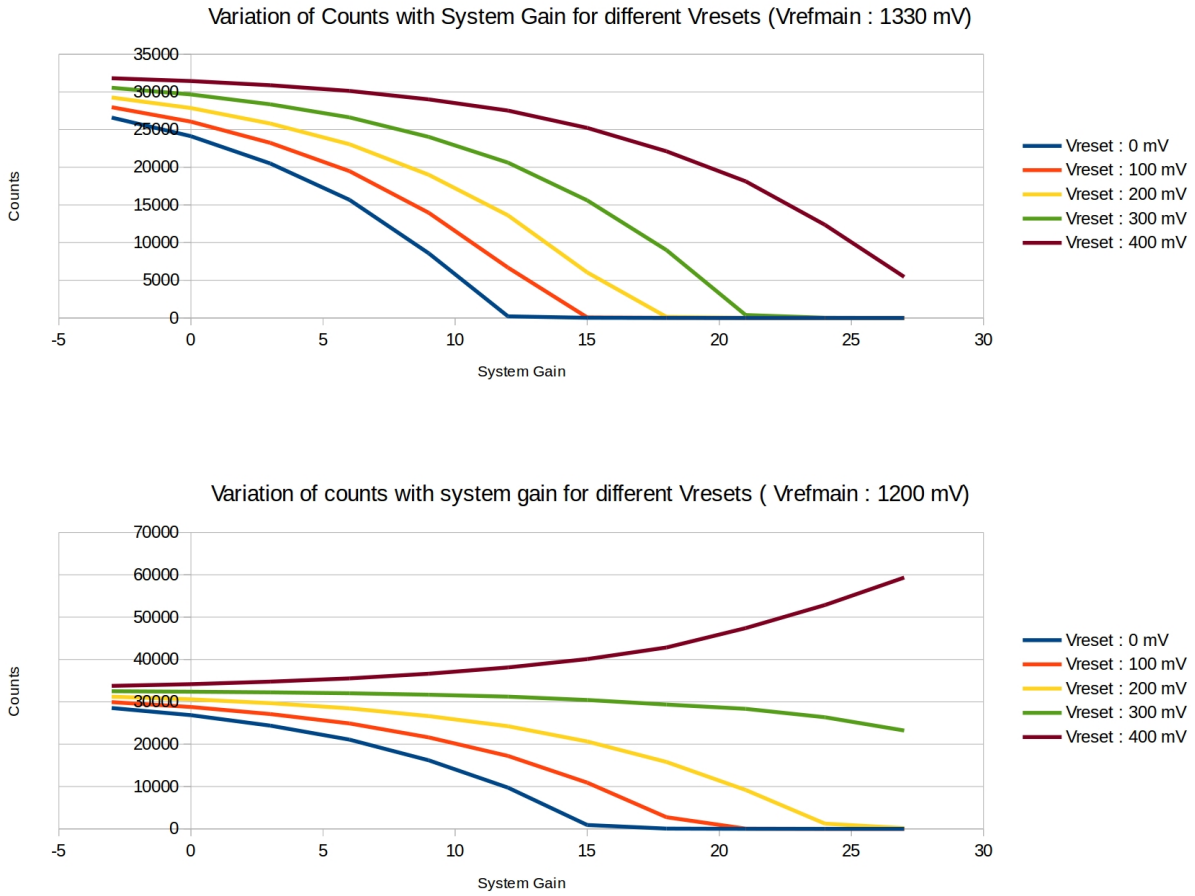


Figure 4.3: Variation of average detector output count with system gain and reset voltage

4.2.4 Persistence test

After exposing detector array to the input radiation, voltage is produced at photo diode. This voltage produced at each pixel should be drained before next exposure otherwise it will add up in the next exposure ruining the image. This can avoided by resetting detector array each time before exposure or by giving sufficient time between two exposures for voltage to drain completely. Larger the intensity of the exposure, larger will be the time for which voltage produced will remain at the detector. By performing multiple reset operations, this voltage can be drained quickly.

Persistence test provides information about number of resets it takes two nullify the effect of previous exposure before new one. This test is performed by getting exposure readout followed by multiple reset operations. Analyzing the readout obtained, minimum number of reset operations to sufficiently drain the effect of exposure is determined.

As already mentioned, these tests are implemented and tested on ISDEC 2.0 setup and will be implemented in ISDEC 3.0 as well once it is commissioned. There are other possible tests which can be automated such as inter pixel capacitance test, tests for finding system gain, multiplexer gain etc. They will also be implemented in ISDEC 3.0.

Chapter 5

Application software for ISDEC 3.0

Application software stands at the top of ISDEC architecture. It provides handle to the user using which ISDEC system can be controlled. The software structure for ISDEC 3.0 is very similar to ISDEC 2.0. The structure can divided into three parts, parameter files, configure program and exposure program just like ISDEC 2.0 system. This chapter describes process of the development of this software, challenges in designing the code and current status of the software development.

Parameter files for ISDEC 2.0 and ISDEC 3.0 are same. The way configure program and exposure program get input from these files and communicate with hardware is different form ISDEC 2.0 because hardware components and there firmware are different in both systems. Please refer section 4.1 to understand parameter file structure. Following sections describe the process of development of other parts of application software i.e. configure program and exposure program.

5.1 Configuring ISDEC 3.0 system

ISDEC 3.0 system consists of USB slave micro-controller, FPGA and SIDECAR ASIC.
Job of USB micro-controller is to provide USB interface for the system and allow direct communication between FPGA and PC. USB micro-controller has to be programmed to do

this job.
ISDEC 3.0 uses cypress FX3 micro-controller. The firmware of this micro-controller is written in C and it's image can be downloaded to the controller via application programming interface (API) provided by Cypress. This firmware can be modified depending upon the application of the micro-controller. Cypress provides few default firmwares for various applications such as loop-back application or for getting constant or incremental data at output endpoint. One can also define input and output endpoints using this firmware.
ISDEC uses slave-FIFO mode firmware wherein direct memory access (DMA) data transfer takes place in between FPGA and FIFO memory of USB. This completely bypasses micro-controller.
First job of the configure program is to download the appropriate USB firmware such that USB micro-controller is ready for application. Image 5.1 shows the snapshot of the process of firmware download.

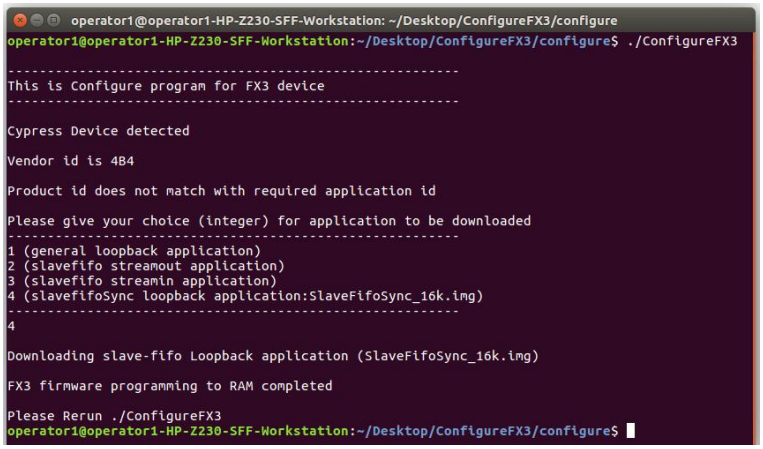


Figure 5.1: Configure program installing USB firmware

USB device is identified by product ID and vendor ID. Vendor ID of the device can be configured in firmware and hence each firmware can be identified with unique vendor ID. Configure program checks if the vendor ID matches appropriate firmware and if not, downloads the firmware to the device.
Once USB firmware is downloaded, micro-controller is ready for it's application. Next job of the configure program is to initialize all other ISDEC components. Initialization process involves turning on ISDEC board by supplying appropriate voltages to all hardware component. This process also involves setting ASIC operation speed. (setting ASIC frequency)
Before proceeding further into configure program, it is necessary to understand general in-

put command structure and about communication of the input command to the ISDEC hardware.

Input command structure and USB communication

All the data transfers in both configure and exposure programs are carried out using libusb library. Please refer section 3.1 to understand principles of USB communication and application of libusb library in detail.

Configure program uses synchronous transfers for communication of input commands to FPGA and FPGA response to PC. Input commands are in hex data format. For particular operation to be performed, there exists a unique command. There is only one input endpoint where all the input commands are given. FPGA then analyzes the command and executes it. FPGA also produces response which indicates the status of command execution (i.e. success or failure). This response is read at out endpoint. This is how user gets know about status of command execution.

If the command is executed successfully then configure program proceeds to the next command and so on. Otherwise it terminates so that problem can be solved.

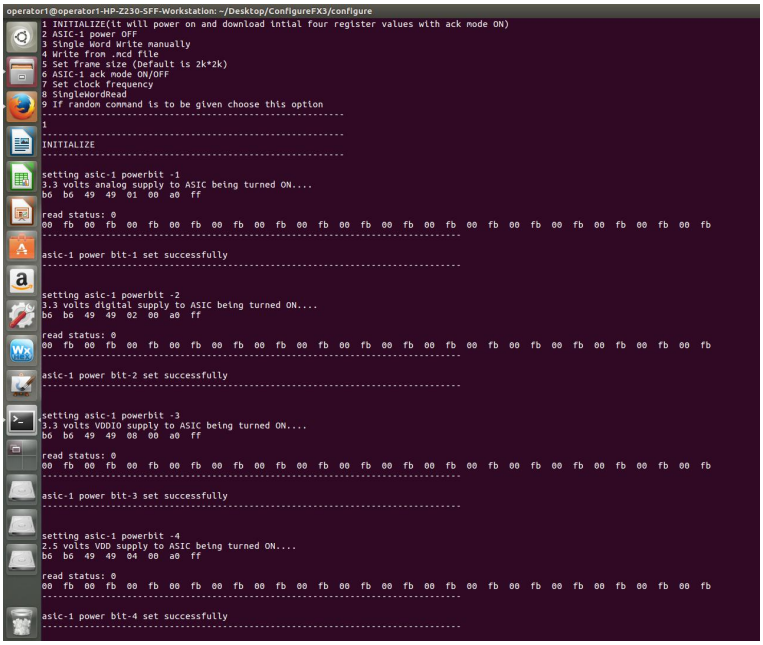


Figure 5.2: Configure program initializing ISDEC device

Now in the Initialization process, commands for setting ASIC frequency and for turning

on ISDEC power are executed one after the other. After this step in configure program, USB micro-controller is ready and ISDEC setup is powered on. FPGA program is already hard-coded. All that remains is configuring SIDECAR ASIC.
SIDECAR ASIC consists of many control registers and values written to those registers determine the functionality of ASIC. Each register address corresponds to certain parameter in ASIC configuration. Firmware of ASIC contains nothing but register addresses and their corresponding values. By writing these values to appropriate registers, ASIC can be configured. ASIC provides specific command structure for writing particular value to particular register. Using this command structure, configure program can write commands to USB endpoint and communicate them to ASIC via FPGA. This is the way ASIC configuration is done.
One can write to single register at a time or to multiple registers at a time using single-word write and block-word write structures provided by ASIC manual. In case of writing multiple registers at a time (i.e. block-word write operation), CRC check (refer section 3.3)) is added at the end of the command structure to prevent data corruption and incorrect configuration of ASIC.
ASIC also has the provision for single-word read and block-word read operations which can be used to read ASIC register values. One can ensure that values written to ASIC control registers using this provision.

After the configuration of ASIC, ISDEC device becomes ready for image acquisition and hence the job of the configure program is done.

5.2 Image acquisition from ISDEC 3.0 system (Exposure program)

Exposure program is the last part of the application software which obtains readout data from the detector. This program communicates all the parameters from parameter files to the ASIC in same way as configure program. It then executes command to obtain the readout from the ASIC and reads the ASIC readout. This program also produces images from the obtained data. Images are formed with the help of CFITSIO library. (see section 3.2)
Application of exposure program is very simple. It only has to read detector readout. The

actual challenge in exposure program is to read the detector output at high data transfer speeds without loss and corruption of data. Exposure program uses asynchronous transfer operation for reading the detector output as it is more suited to read continuous stream of data at high speeds. (refer section 3.1)

H2RG detector provides slow option and fast option for readout sampling. Sampling rates for slow mode are around 50 KHz to 500 KHz. The fast mode of H2RG detectors samples data at 5 MHz sampling rate. ISDEC 2.0 could support slow mode readout but not the fast mode due to limitations of USB 2.0 interface. To be able to sample data in fast mode, USB throughput should be around 300 MBPS which is possible in ISDEC 3.0.

Along with asynchronous USB communication, this program also uses dynamic memory allocation to increase the performance and efficiency of the readout.

To be able to obtain fast mode readout, it is necessary to make sure that all the components involved in the data transfer process are able to support this throughput. In case of loss of data, it is essential to understand where exactly loss happens in order to troubleshoot the issue. This step by step approach is followed while developing exposure program.

First step is to ensure that USB chip is able to read and write data at high speed. To test this, USB loop-back application was used which loop-backs data written at the input endpoint to the output endpoint.

Next step is to ensure that there is no loss of data between PC and FPGA communication at high speed. This process involves, writing input data at USB input endpoint which is then communicated to FIFO memory. FPGA reads this command and produces response and writes it on the FIFO memory. USB micro-controller then transfers this response to USB output endpoint where received data is read.

FPGA program has counter application which serves the purpose. Counter application command has provision to give counter number. Once the FPGA receives the command, it produces incremental data starting from zero until the counter number. Hence larger the counter number, larger is the data size. The speed at which FPGA writes this data can be controlled. Also as the data being incremental it is easy to check for data loss if any.

Full frame image readout from the detector is of the size around 8 MB. Hence the system should be able to read at least 8 MB data with the speed of around 300 MBPS. Using this counter code application, data chunks of size 128 MB were successfully read at 320 MBPS FPGA writing speed. This clearly indicates that PC to FPGA interface is capable of reading fast mode readout.

In this counter code application, FPGA produces counter response but there is no flow

of data though FPGA architecture. In actual readout, data from the detector will flow though FPGA architecture according to FPGA program structure. FPGA program also has provision to simulate ASIC readout.

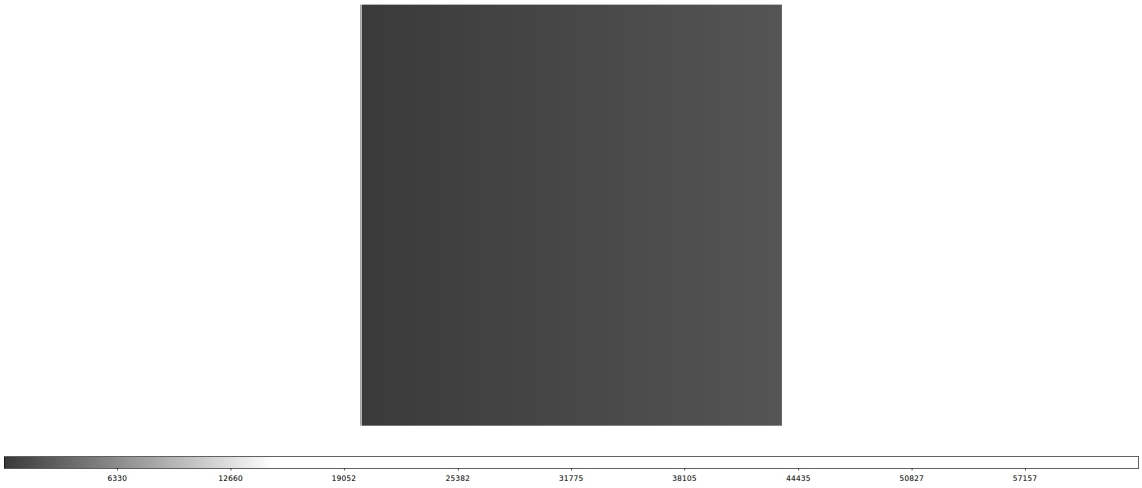


Figure 5.3: Image obtained by the exposure program for simulated ASIC data by FPGA

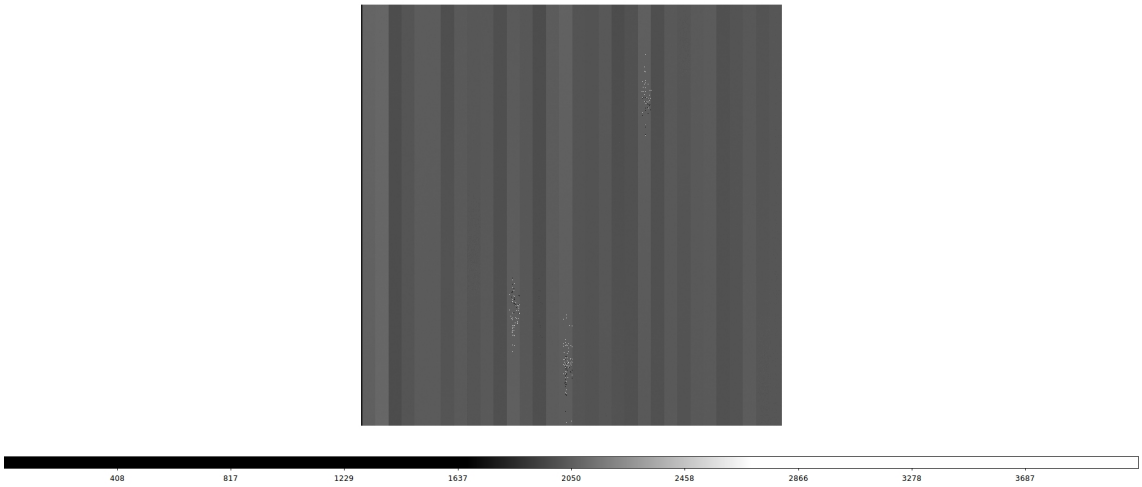


Figure 5.4: Image obtained by the exposure program for the original ASIC data

Figure 5.3 shows the image acquired by the exposure program at ASIC operation frequency of 40 MHz. No data loss in this mode implies that FPGA architecture can support fast mode data acquisition. (upto 40 MHz operation frequency) Once all the components and in between links are checked for their ability for high speed data transmission, actual ASIC can be connected to the setup to obtain the readout.

Figure 5.3 shows the original ASIC readout at operation frequency of 40 MHz. Different stripes are seen because electronic noise of each voltage sensor in 32-output mode is different.

5.3 Current status of the development

Readout speed of 200 MBPS is achieved for reading the ASIC simulation data. This means that ISDEC architecture except actual ASIC connection can support readout upto 200 MBps. For original ASIC data readout however, data transfer speed of up-to 160 MBPS is achieved. Data loss is observed at higher operation speeds.
Final aim of the project is to support data transfer speeds of upto 300 MBPS in order to support fast mode readout. Work is in progress to achieve that target.

Chapter 6

Discussion and Conclusion

This thesis is about development of application software for high speed data acquisition system and development of automated scripts to ensure the performance of such system. This chapter includes discussion about results obtained so far and things necessary to achieve the final goal.

Automated tests described in section 4 were implemented and tested for ISDEC 2.0 setup. It is observed that implementation of these tests saves large amounts of time required in testing detector setup manually. These tests also successfully provided bias voltage parameters for which detector readout did not saturate. These tests were carried out in the lab without actually exposing the detector in the sky. However same tests can be implemented for the actual telescope to check it's functioning and to obtain best set of detector parameters to obtain best image. After commissioning of ISDEC 3.0, these tests will also be implemented for that system.

The application software development is still in progress. The configuration part of the software is successfully completed. (see section 5) Configure program successfully configures the system and makes it ready from data acquisition. As described in section 5, the performance of the exposure program can be analyzed using FPGA produced ASIC simulation data. It is observed that exposure program can read ASIC simulation data for data transfer speeds of up-to 200 MBPS. When original ASIC data is read, it is seen that data is successfully read at speeds of 160 MBPS without any loss. This throughput is quite remarkable compared to ISDEC 2.0 throughput which was only up-to 30 MBPS. Final aim however is to support fast mode readout which requires throughput of around 300 MBPS. Development is still in

progress to achieve this target.

The ultimate goal of the project is to commission fully functional ISDEC-3.0 system. Expertise in many areas of science and technology is necessary to build such system. These areas of expertise include digital circuit design, circuit assembly, micro-contoller and FPGA programming, software development and so on. Apart from software development, this project provided an opportunity to understand these areas in brief and to understand the process of building an instrument as a whole.

Appendix A

List of Abbreviations

1. ANSI : American National Standards Institute
2. API : Application Programming Interface
3. CIRCE : Canarias InfraRed Camera Experiment
4. DDR : Double Data Rate
5. FIFO : First In First Out
6. FITS : Flexible Image Transport System
7. FPGA : Field Programmable Gate Array
8. GTC : Gran Telescopio CANARIAS
9. HAWAII-xRG detector : HgCdTe Astronomy Wide Area Infrared Imager with xK x xK resolution, Reference pixels and Guide mode
10. IGO : IUCAA Girawali Observatory
11. ISDEC : IUCAA SIDECAR Drive Electronics Controller
12. LBT : Large Binocular Telescope
13. LMIRcam : Large Mid Infrared Camera
14. Op-amp : Operational amplifier
15. RoboAO : Robotic Adaptive Optics for small telescopes
16. RSS-NIR : Robert Strobie Spectrograph Near Infrared Instrument
17. SALT : South African Large Telescope
18. SFP transceiver : Small form-factor pluggable transceiver
19. SIDECAR ASIC : System Image, Digitizing, Enhancing, Controlling, and Retrieving Application Specific Integrated Circuit

20. USB : Universal Serial Bus

Appendix B

ISDEC-2.0 readout images

This section contains FITS images obtained from ISDEC 2.0 in various modes. These readouts were used to develop automated ISDEC based controller tests. The readout images shown here are obtained at IUCAA instrumentation lab without actually exposing the detector to the sky.

Read and reset frames

Following are the images of the read frame and reset frame obtained from the detector while keeping all the parameters in parameter file at default value. Reset frame provides estimate of the noise in the system. Noise in read image can be removed by comparing it with the reset frame.

Effect of bias voltage variation (main reference voltage) on the readout

This section contains readout images showing effect of variation of main reference voltage on the readout. All other parameters are kept at default value. It can be seen that, readout values begin to saturate for reference voltage values above and below the optimum value.

Effect of system gain variation on the readout

These images show variation of detector readout with system gain parameter. For this readout all other parameters are at default value and main reference voltage is at it's optimum value i.e. 1150 mV. It can be seen that, detector readout never really saturates even at the extreme gain values because of the optimum choice of reference voltage.
Similar readout is obtained for reset frames as well to obtain optimum reset voltage value. It can also be seen that the detector output does not saturate for optimum reset voltage value.

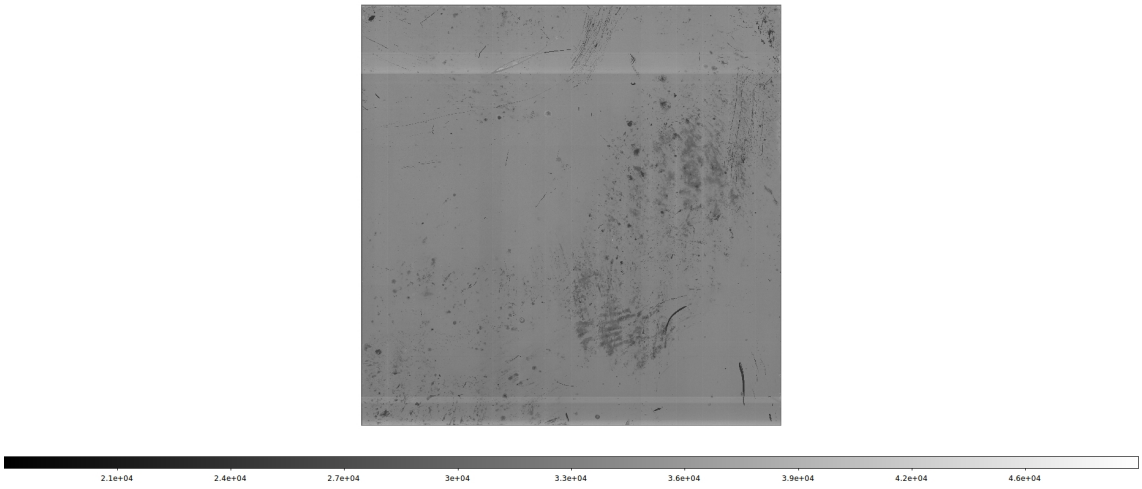


Figure B.1: Default read frame

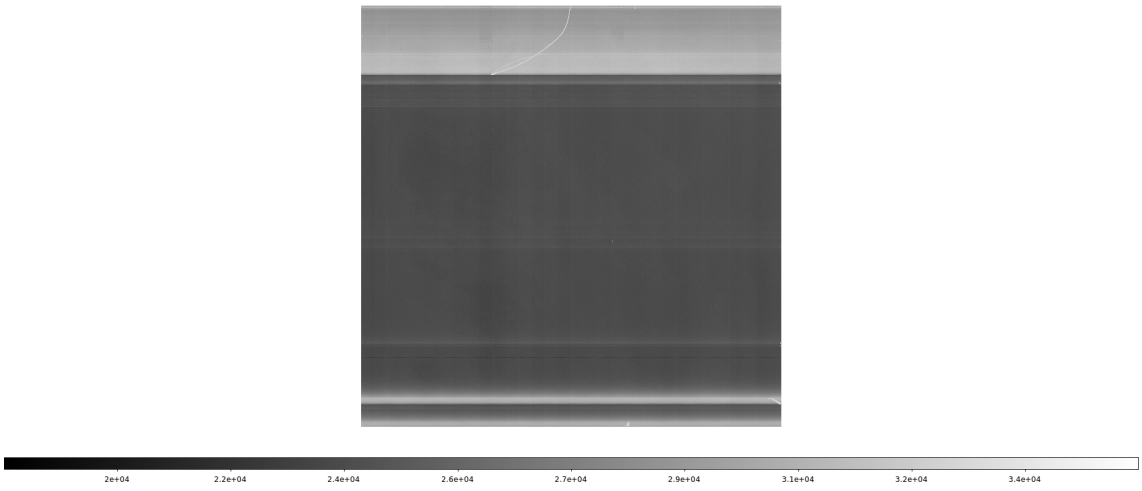


Figure B.2: Default reset frame

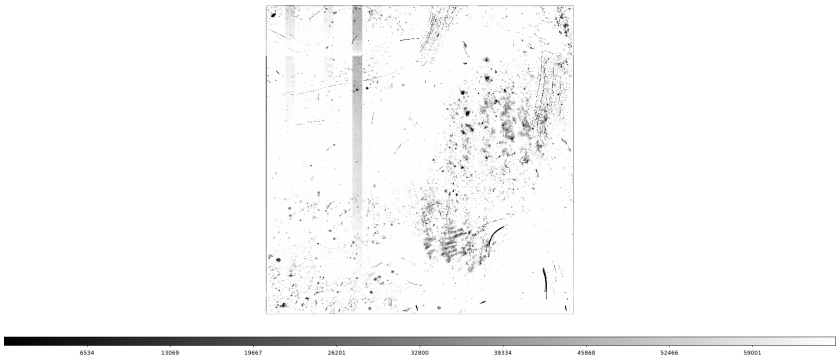


Figure B.3: Main reference voltage : 1000 mV

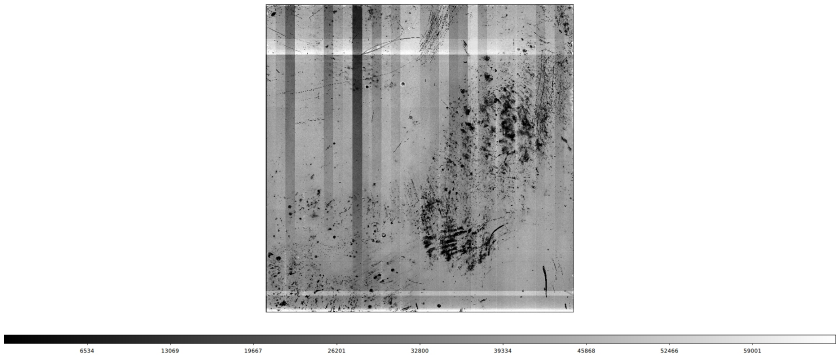


Figure B.4: Main reference voltage : 1150 mV

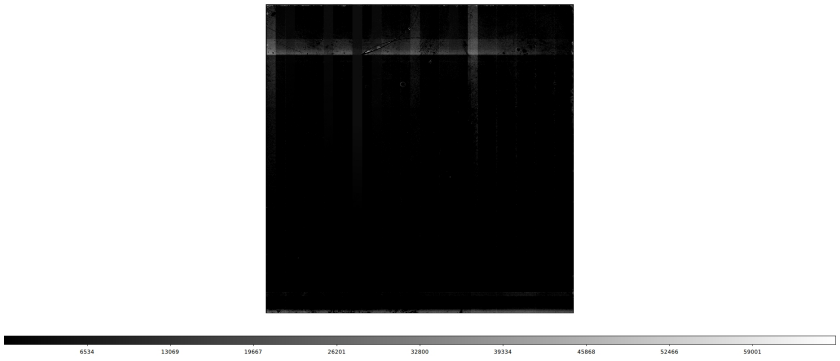


Figure B.5: Main reference voltage : 1300 mV

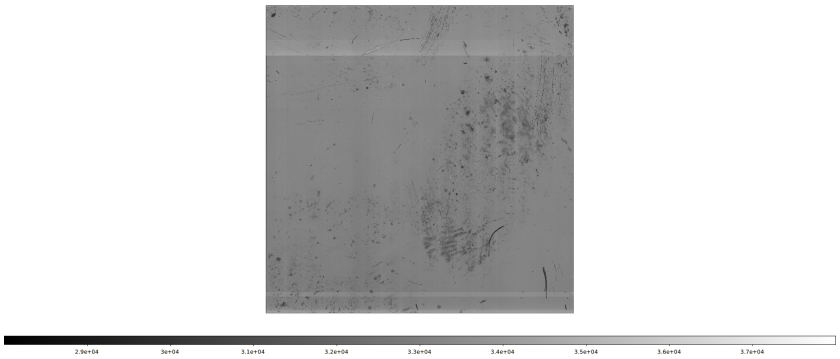


Figure B.6: System gain value : -3 dB (Vref : 1150 mV)

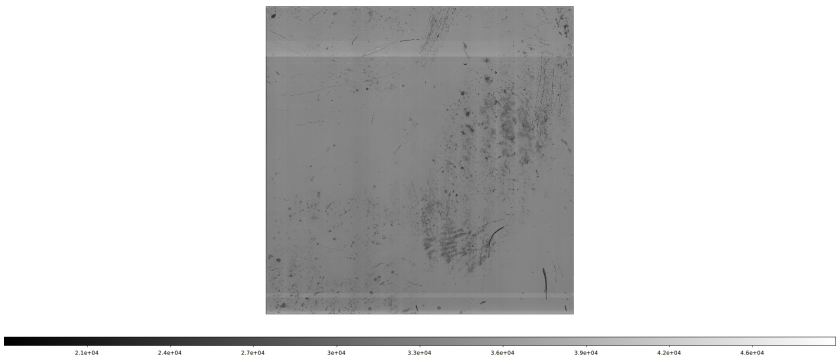


Figure B.7: System gain value : 6 dB (Vref : 1150 mV)

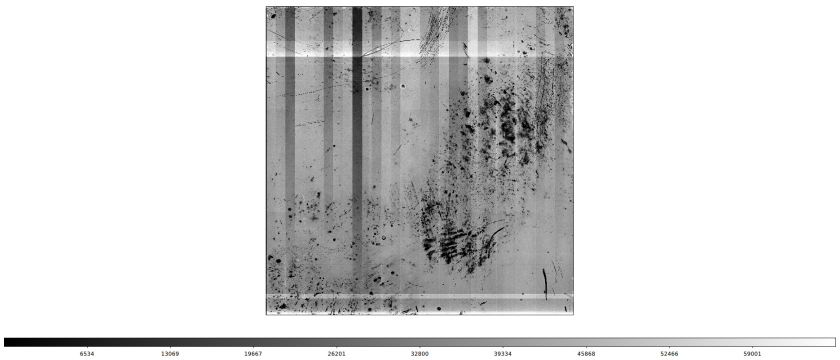


Figure B.8: System gain value : 27 dB (Vref : 1150 mV)

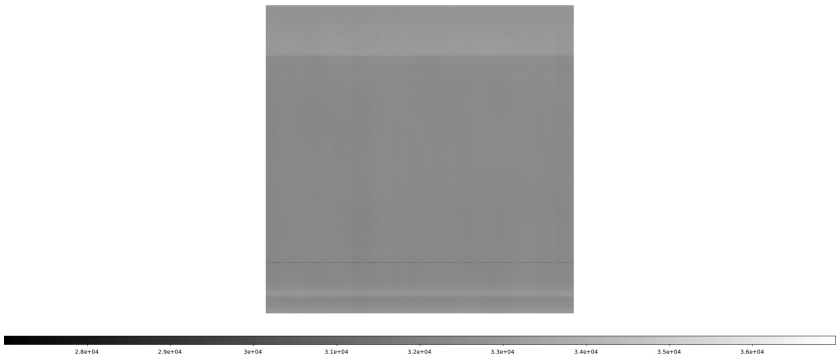


Figure B.9: System gain value : -3 dB (Vrst : 350 mV)

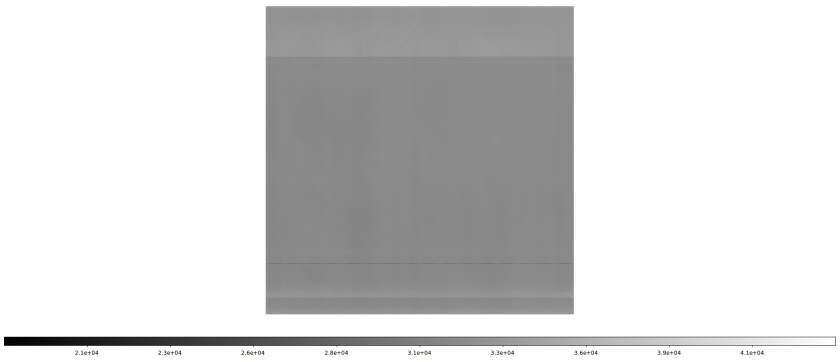


Figure B.10: System gain value : 6 dB (Vrst : 350 mV)

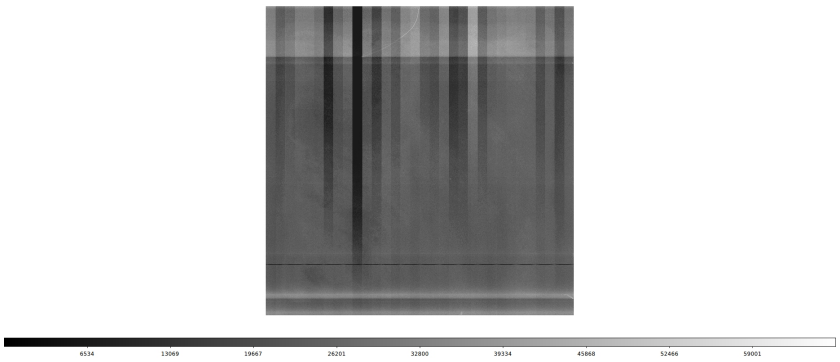


Figure B.11: System gain value : 27 dB (Vrst : 350 mV)

Appendix C

ISDEC parameter files

Application software for ISDEC has all it's parameters stored in parameter files. There are five different parameter files. This appendix contains images of all the parameters file.

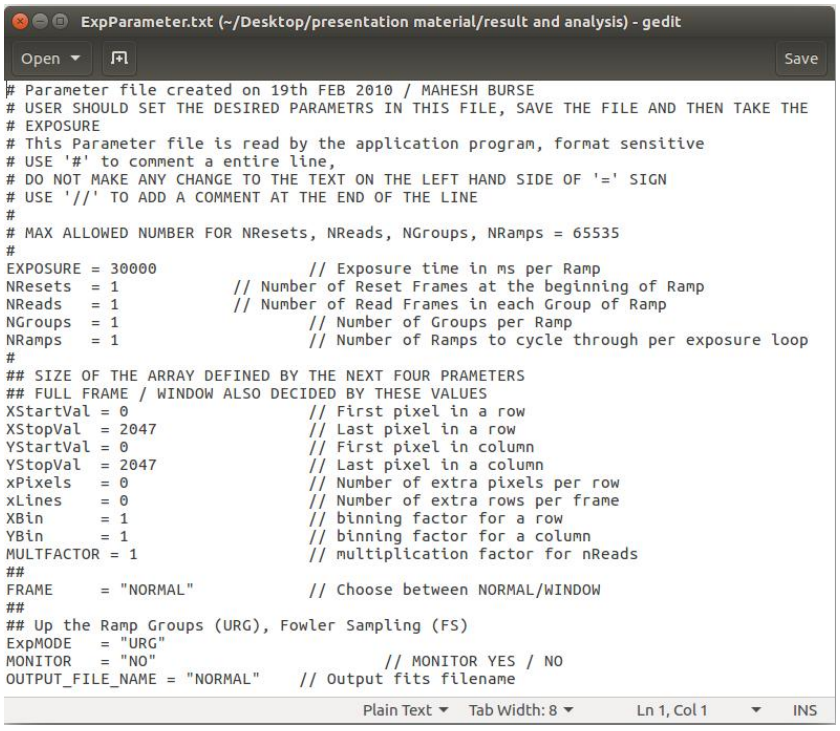


Figure C.1: Exposure parameter file

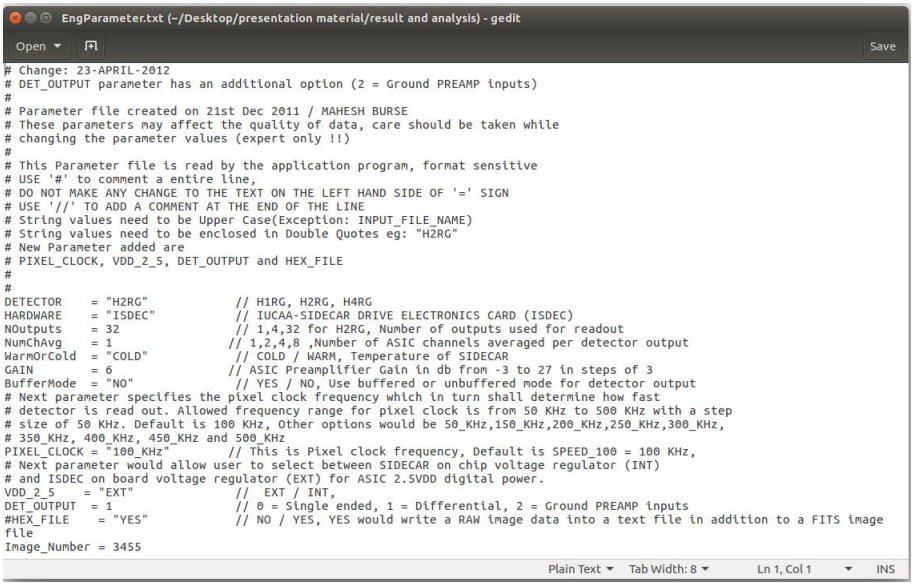


Figure C.2: Engineering parameter file

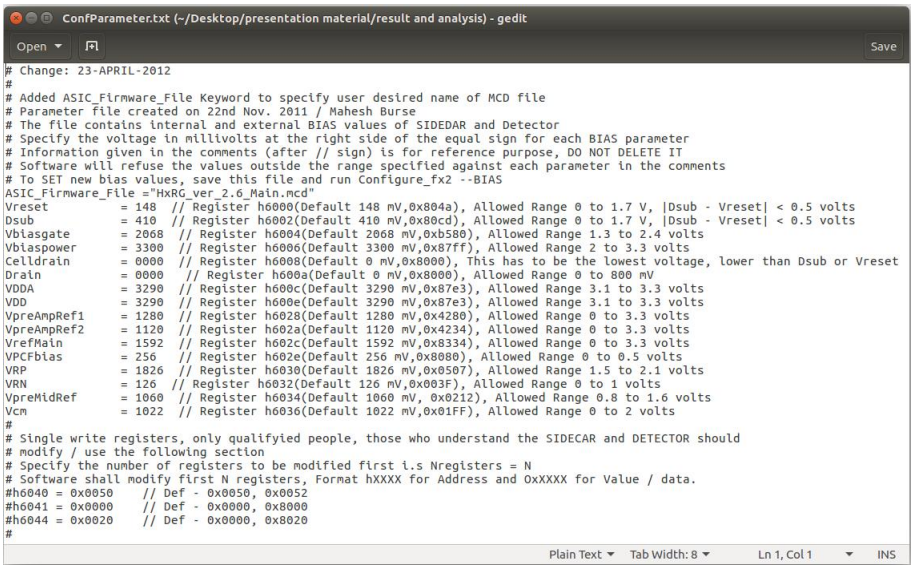


Figure C.3: Configuration parameter file

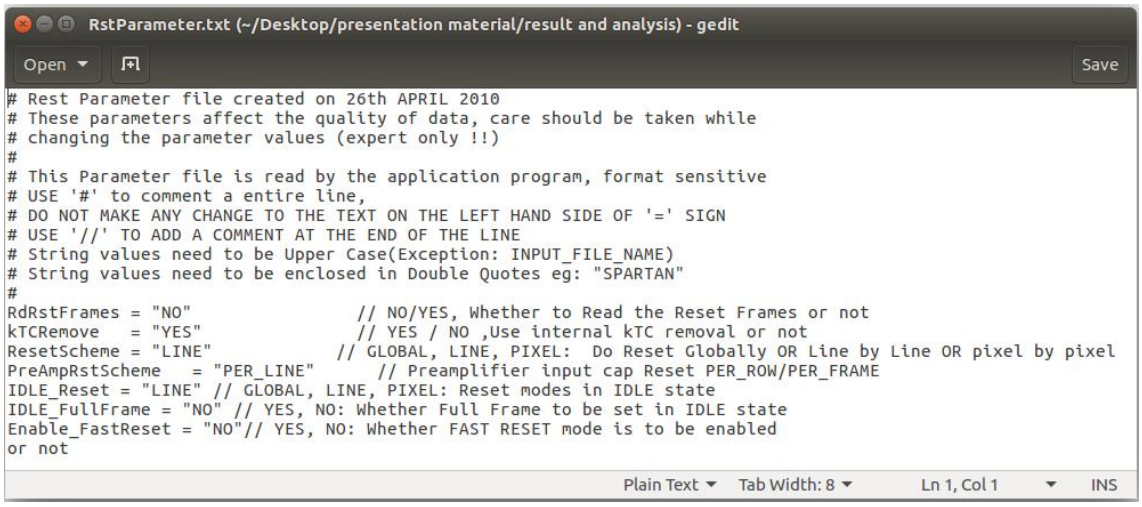


Figure C.4: Reset parameter file

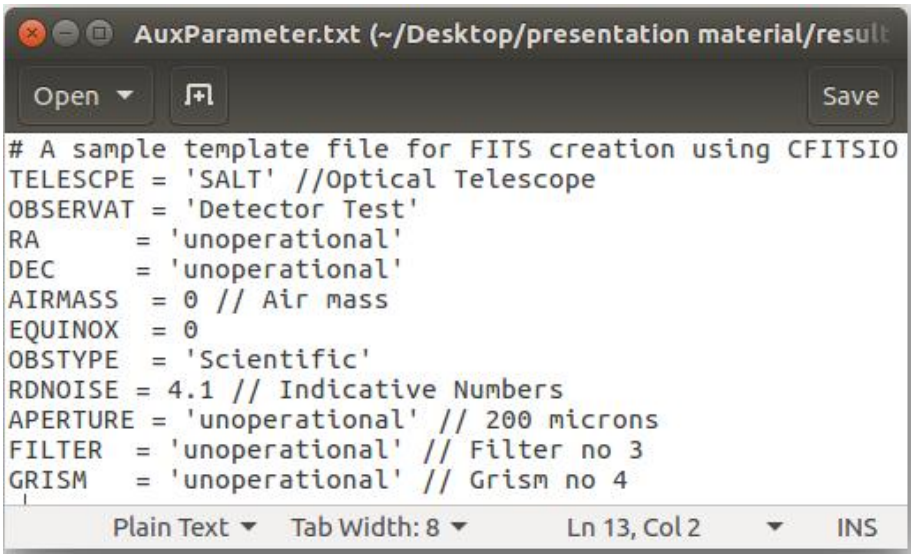


Figure C.5: Auxiliary parameter file

Bibliography

[1] Ramaprakash A., Burse M., Chordia P., Chillal K., Kohok A., Mestry V., Punnadi S., Sinha S. *Controller and data acquisition system for SIDECAR ASIC driven HAWAII detectors*, Proc. of SPIE Vol 7742-93 (2010)

[2] Mahesh Burse et al.; *ISDEC-2 and ISDEC-3 controllers for HAWAII detectors*, SPIE 9915, High Energy, Optical, and Infrared Detectors for Astronomy VII, 991526

[3] P. Norton; *HgCdTe infrared detectors*, OPTO-ELECTRONICS REVIEW 10 (3), 159174 (2002)

[4] McLean, Ian S.; *Electronic Imaging in Astronomy - Detectors and Instrumentation*, Springer

[5] Cypress Semiconductor; *Getting Started with FX3 SDK*, Cypress documentation

[6] Cypress Semiconductor; *CyUSB Suite for Linux Version 1.0.4 Programmer's Guide*, Cypress documentation

[7] Clive Maxfield; *Fundamentals of FPGAs*, EETimes, http://m.eet.com/content/courses/course1919/player.html

[8] libusb API; *http://libusb.sourceforge.net/api-1.0/*

[9] Cyclic redundancy check; *https://en.wikipedia.org/wiki/Cyclic-redundancy-check*